Preface

We are remembering a great scientist, a colleague, a friend in this special issue: Tony (Antal) Bejczy physicist, lead researcher of the National Aeronautics and Space Administration, NASA's Jet Propulsion Laboratory, lecturer of the California Institute of Technology and honorary doctor of Óbuda University, was an internationally renowned expert in space research and robotics.

Remembering his achievements and great personality, Óbuda University Antal Bejczy Center for Intelligent Robotics established the International Bejczy Day, falling on January 16th of each year, the birthday of Tony. This special issue was called upon as a joint work of many of his former colleagues, friends and robotics experts inspired by his work. It spans across various domains of robotics and control engineering from space robotics to underwater vehicles, representing the diversity of the field that grew tremendously in the recent years, partially thanks to the pioneering work of Tony. The Introduction to this volume is given by two of his closest friends and collaborators, world renowned roboticists, Prof. Khatib from Stanford University and Prof. Fiorini from University of Verona.

Acknowledgements. We would like to thank all authors and coauthors for their great contributions to this special issue. Many thanks go to the reviewers and the guest editors.

I hope you will enjoy reading this issue.

Prof. Dr. Imre J. Rudas

University Research and Innovation Center

Óbuda University

Budapest, January 16, 2016

Introduction

This special issue is dedicated to the memory of Tony Bejczy, in celebration of his life and in recognition of his scientific achievements in robotics. In the early years of the field, when the concept of a robot was still forming, the creation of an intelligent machine in the appearance of a human was an exciting aspiration for many researchers and engineers.

For the two of us, the journey into the emerging field auspiciously began with a privileged encounter with Tony Bejczy. Our first meetings with Tony took place long before we actually started to interact with him professionally, encounters which have made lasting impacts on us both.

For Paolo, the first meeting with Tony came during his visit to NASA Jet Propulsion Laboratory in 1981, during the Saturn encounter with Voyager 2. On that occasion, the Laboratory was open to the public, and Tony was presenting his work on teleoperated robots. Paolo, as a consequence of that meeting, decided that robotics would be the topic of his Master's Thesis. Several years later, in 1985, when Paolo applied for a position at JPL, Tony recalled to have seen Paolo's thesis work at one of the earliest robotics conferences in 1983. In the years after, Tony and Paolo became colleagues at JPL, and their collaboration and interaction continued later, when Paolo moved back to Verona.

The encounter of Oussama with Tony took place in Toulouse in 1978. Tony was invited together with Dan Whitney by Georges Giralt for a special Colloquium at LAAS/CNRS. Oussama, who at the time was a PhD student at SupAero, attended what was actually his very first Colloquium in robotics, which turned out to be his most memorable. Several years later, Oussama moved from Toulouse to Stanford. In the mid-eighties came the birth of the IEEE-RA Council, which later became the IEEE-RAS Society. These developments in the field afforded Oussama and Tony the opportunity to work together both on technical and professional activities. They also eventually collaborated on the organization of conferences, notably ICAR 1997 in Monterey, California, and have jointly participated as speaker or panelist in numerous events.

It has been a great privilege to have known Tony throughout our time in the field. For me, Tony was an inspiring colleague, and a wonderful friend. This special issue contains a compilation of fourteen contributions that explore the interaction and impact of Tony's work on the development of robotics, from its early years to today's most recent achievements.

The first four papers address Tony's specific contribution to robotic surgery, sensorized robotic hands and robotic intelligence. The paper by Takács, Nagy, Rudas and Haidegger describes the history of robotic surgery, tracing its origins to the initial NASA desire of giving astronauts remote medical assistance during long duration missions. Tony had the key role in the development of NASA-JPL's RAMS system, the first surgical robot with haptic feedback used in animal

experiments, and in building a relation between NASA and Computer Motion Co., the developer of the Zeus robotic surgical system. Some of the key advances in robotic surgery are also the subject of the paper by Kronreif that addresses some challenges of developing new applications in surgical robotics. In spite of the difficulties, new systems were developed by Austrian companies for very simple but widespread procedures, such as biopsies. The paper by Rudas, Gáti, Szakál and Némethy summarizes the career of Tony Bejczy, as the developer of "Smart Hands" and of dynamical models of robots. These two technologies were combined in the advanced teleoperation systems at JPL, demonstrating their applicability of these technologies to space exploration. The paper by Kovács, Petunin, Ivanko, and Yusupova addresses the relation that exists between Robotics and Artificial Intelligence, as demonstrated in the chess playing machine "the Turk", the computer chess Deep Blue and the Mars rover Pathfinder. They all shared the common aim of reasoning about the data from the environment and acting upon this reasoning.

The impact of Tony's research on control theory is very widespread and is addressed in the following six papers. The paper by Roman, Radac, Precup and Petriu proposes a new tuning approach by which the parameters of data-driven Model-Free Adaptive Control (MFAC) algorithm are automatically determined using nonlinear Virtual Reference Feedback Tuning (VRFT) algorithm. This algorithm uses a model free approach that is computationally simple and could replace the knowledge of the dynamical model of the controlled systems, as proposed in Tony's research. The paper by Tar, Bitó and Rudas introduces a control method that can be used as an alternative to the computation expensive Model Predictive Approach. The method described in this paper is based on a fixed point transformation that changes the problem of computing the control signal into the task of finding an appropriate fixed point of a contractive map. The paper by Kowalczyk and Kozlowsky addresses the issues related to the motion planning of non-holonomic mobile robots, such as, cars of planetary rovers. The approach presented relies on a potential field, which is adapted to account for the constraints imposed by the turning radius of the robots. The paper by Lantos and Max describe an algorithm for the hierarchical formation control of a group of unmanned vehicles. The approach presented consists of the generalization of the multi-body method for underactuated car-like vehicles, developed originally for fully-actuated surface ships. The control system consists of the high level centralized formation control of the UGVs and the low level decentralized PID type suspension, speed and steering control of the different vehicles. The paper by Takács, Dóczi, Sütő, Kalló, Várkonyi, Haidegger and Kozlovszky describes the challenges of developing an autonomous underwater vehicle suitable for search and rescue missions. Standard tests have been developed by NIST to validate the capabilities of such robotic devices and the paper describes the enhancements made to a research robot to perform the NIST tests. The paper by Eigner, Tar, Rudas and Kovács describes a novel approach to the modeling of pathological situation, such as diabetes. The authors propose to use the Linear Parameter

Varying (LPV) methodology, which is based on the parameter vectors and is a satisfactory model of the disease.

The next three papers address the human–robot interaction in the context of teleoperation and of collaborative robots. The paper by Muradore and Fiorini describes some of the algorithms that are currently used to control a remote robot by a human located in a safe place. These algorithms can trace their origin to the seminal work of Tony's in his Advanced Teleoperation Laboratory at JPL in the 1980s, where force feedback was proven to be essential to guarantee a safe interaction with the remote environment. The paper by Kinugawa, Sugahara and Kosuge presents a new concept of robotic co-worker, called PaDY. This prototype is an example of Physical Human Robot Interaction that is one of the most important achievements of today's robotics: i.e., developing robots that will not hurt humans. This concept is also related to haptics and robot operator interaction, and it nicely extends the work on telerobotics and robotic teleoperation with strict safety constraints. These concepts are also addressed by the paper by Ficuciello, Villani and Siciliano that deals with safety in human–robot interaction, where contacts between the human and his robotic co-worker should be always injury free. The paper presents solutions for the cases of accidental contacts and co-manipulation, which is a very peculiar case of robotic teleoperation.

The last paper of this special issue addresses the research area that summarizes all the technologies described above, which was the main focus of Tony's work, i.e., space telerobotics and force feedback teleoperation. The paper by Artigas and Hirzinger summarizes the development of these two technologies at the German Space Agency DLR, starting with ROTEX, the first space telerobotics mission in history, and continuing with demonstrations of on-orbit servicing space force-reflection teleoperation.

The contributions presented in this special issue cover a wide range of robotics areas. These contributions are a testimony to the impact Tony's work has made in robotics. We would like to extend our appreciations and thanks to all the authors who contributed to this memorial special issue.

Tony's vision of robotics, the impact his work has made in many research areas, the energy and enthusiasm he always demonstrated in pursuing his research and communicating it to the robotics community, all showed him as an influential leader and a major contributor to the field. This special issue represents a tribute to a dear friend and a great colleague to many in our community.

Prof. Paolo Fiorini, University of Verona (Italy)

Prof. Oussama Khatib, Stanford University (USA)

Origins of Surgical Robotics: From Space to the Operating Room

Árpád Takács, Dénes Á. Nagy, Imre J. Rudas, Tamás Haidegger

Antal Bejczy Center for Intelligent Robotics,
Óbuda University, Bécsi út 96/b, H-1034 Budapest, Hungary
{arpad.takacs, denes.nagy, haidegger, rudas}@irob.uni-obuda.hu

Abstract: The rapid development of telerobotic systems led to novel applications beyond the nuclear and industrial domains. Medical telerobotics enabled surgeons to perform medical operations from remote places, far from their patient. Telesurgery systems allow great flexibility, improved performance in general, and support the creation of ideal surgical conditions. The first attempts to develop telesurgical systems borrowed the idea from space research, where the need of novel robots emerged for invasive treatment, even under extreme situations, such as weightlessness. Telesurgical instruments on Earth appeared following the same concept, aiming first for military, then onward for civilian applications. Today, more than 1.5 million patients are receiving telerobotic treatment annually, worldwide. As the surgical robotics domain grew from the initial concepts, it developed along three major concepts: telesurgery, cooperatively controlled robots and automated (image-guided) applications. These domains continue to develop into application specific systems with the goal of reaching the specificity and versatility of conventional surgical instruments.

Keywords: surgical robotics; space robotics; teleoperation

1 Introduction

In recent years, a large number of surgical robots and robotic surgery related research projects have been introduced and conducted. As a result, useful software and hardware tools appeared on the market, which accelerated the pursuit for new research results in modern robotic surgery and telesurgery [1, 2]. Computer-Integrated-Surgery (CIS) and telemedicine are becoming popular in the world's developed countries, improving the quality of medical treatment and patient care. The development of these systems requires a strict and effective cooperation of surgeons, IT experts, engineers and scientists from the various fields of natural and human sciences, creating the possibility of remote or even transcontinental surgery. The concept of these systems often originate from specific extreme applications, thus their

testing also requires extreme environments, such as weightlessness or extremely high pressure [3].

2 The Birth of Surgical Robotics

There is no consensus about the title of ”the first surgical robot“, since it is hard to define, what criteria should be used to claim a robots role in a surgical procedure fundamental. Generally speaking, the first systems, which appeared in the 1970s were used for different purposes, primarily as assisting devices and supporting manipulators [4]. The concept of telerobotics for surgery appeared in the early 1970s, initiated by the National Aeronautics and Space Administration (NASA) [5]. The goal of the original project was to provide medical assistance for astronauts during their remote mission. For this purpose, remotely controlled robots would have been used, operated from the Earth, as shown in Fig. 1. At that time the proposal was not funded, only limited documentation remained accessible. Although the idea was concluded in a short period of time, another 15 years passed until the first prototypes appeared, mostly backed by the US military. During the development phase, it became apparent that controlling telesurgical robots is very challenging, e.g. in space, due to the effects of time delay caused by the large distances. The attention from telesurgery in space shifted to shorter distance telesurgery solutions, leading to the introduction of the first commercial oriented surgical robots into the market by the end of the 1980s.

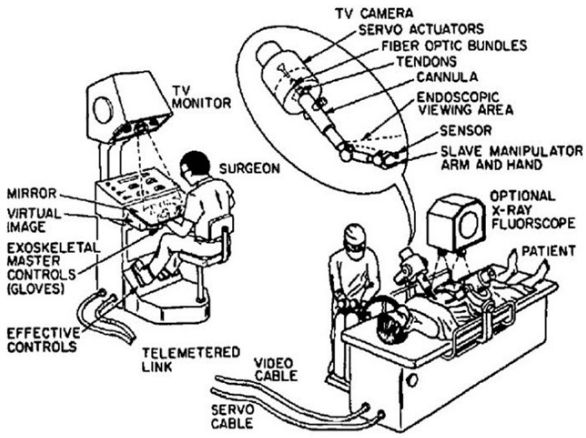


Figure 1
The first concept of a telesurgical robotic system from a USA project proposal (1971) [6].

The first telesurgical systems were designed to improve the dexterity of the surgeons, to increase the reliability of the surgical interventions and to improve the accuracy of the manipulations. Academic centers started to get involved in the development of new systems around the world in the 1980s. The first robot that was explicitly used and designed for patient treatment assistance was the Arthrobot of

Hearthrob, in 1983, together with a scrub nurse robot. The development was led by J. McEwen, G. Auchinlec and B. Day at the University of British Columbia, Canada [7]. The first procedure, assisted by that robot was carried out in 1984 at the UBC Hospital, organized by the same academic institution. A year later, more than 60 arthroscopic procedures were documented [8], however, the submitted patent referred it to as an active supporting device rather than a treatment device [9]. The developer group from UBC submitted another patent on their newly developed robot the same year, but the submission was withdrawn six years later [10]. Nevertheless, they introduced an arm-holder version of the robot, intended for in-surgery use [11].

Thirty years ago, a robot was used on a human patient for the first time, providing direct surgical support. It was proven that accurate stereotaxis could be achieved by the use of robotic surgical systems, in a CT-guided brain biopsy setup shown in Fig. 2.

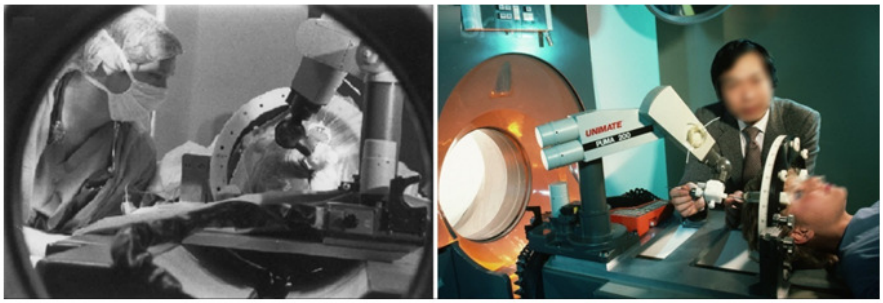


Figure 2
Puma 200, the first robot used for assisting human neurosurgery (1985) [12].

The Puma 200 (Programmable Universal Machine for Assembly) was used for holding and manipulating a biopsy cannulae, navigated by a stereotactic frame mounted on the base of the robot. The intervention was carried out under the supervision of Y. S. Kwoh, at the Memorial Medical Center in Long Beach, CA, USA [12]. The Puma 200 provided a development platform for many novel instruments in the later years. It was used for image processing, performing complete neurosurgical operations, all of which was achieved by manipulating probes during the positioning of the robotic arm.

The U.S. Army has invested a lot of resources into robotic surgery ever since the beginning. It was evident that robotic telesurgery would allow medical assistance and treatment to the soldiers on the battlefield, increasing safety and decreasing treatment costs. Early prototypes of remote healthcare systems were introduced by the Telemedicine & Advanced Technology Research Center (TATRC), based on the concept of telesurgery proposed by NASA. The Green telepresence system is considered to be the first system where doctors could apply treatments to patients from a distance, introduced in 1991 [13]. This system can be considered as the predecessor of the later most successful da Vinci surgical system. The Green telepresence

system was a prototype for telesurgery, but from the structural and functional design point of view, it already contained the fundamental concept for today's teleoperation systems. Even more, it incorporated force feedback, which is not yet part of the commercial systems today. The very first system used for long distance experiments was created by the Jet Propulsion Laboratory (Pasadena, CA) and used for experiments reaching Milan, Italy (1993) [14]. Several years later, the U.S. Department of Defense launched it's long distance medical assistance project, the Trauma Pod. The project goal was to demonstrate the feasibility of an integrated, robot-driven emergency care unit, planned to be used by combat surgeons in order to perform remote operations by 2025 [15, 16].

3 Early Prototypes and System Concepts

The real growth of surgical robotics happened in the 1990s, in which Computer Motion Co. (Goleta, CA) played and important role. Computer Motion was founded in 1989, and they were specialized in developing systems for laparoscopic surgery. Their first product, the AESOP (Automated Endoscopic System for Optimal Positioning) was an endoscopic camera holder, used for camera assistance [23]. Nonetheless, this was the first tool of its kind to be approved by the Food and Drug Administration (FDA). The second generation of AESOP was capable of voice control and was introduced in 1996, while the third generation, released two years later was improved by adding an additional Degree(s) of Freedom to the robotic arm, and further developed for networking with other devices in the operating room. The AESOP arms were later extended, making them capable of controlling a wide range of surgical instruments. The new system was named Zeus Robotic Surgical System, which included three arms [25].

In the year of 1995, the first prototype was created, while the first animal trials were conducted in 1996 (tubal re-anastomosis). Two years later, the first coronary artery bypass graft (CABG) was carried out, and the system was improved with new components through the early years of the 2000s, such as the Micro-Wrist and the Micro-Joint. The latter was improved to hold and manipulate more than two dozens of different surgical instruments, including scissors, dissectors, scalpels and hooks. FDA approval of the Zeus system came in 2001, which opened up new frontiers in clinical use. The high-end Zeus systems were capable of filtering hand tremor, increasing the reliability of the interventions. The system was designed for Minimally Invasive Surgery (MIS), making it an appropriate tool for beating heart surgery, mitral valve surgery or endoscopic CABG.

Imperial College (London, UK) has been a European flagship institute of surgical robotics since the mid 1980s, becoming a lead research center for prostate surgery [28]. They developed the system called PROBOT for transurethral resection of the prostate (TURP) procedures. This procedure is usually carried out manually with instruments inserted into the urethra and reaching the prostate gland from the inside. Then a wire cutter is inserted, and a conic shape is excised from the prostate. This is often performed manually, however the procedure is long and the absorp-

Figure 3
The AESOP surgical robotic platform for camera handling.

tion of hypo-osmolar irrigation fluid can cause further complications. The first five human patients were treated between 1991 and 1993. The successful beginning led to the establishment of the a new Science Engineering Research Council funding initiative. However, the activities at Imperial College were not limited to urologic procedures. They also developed the ACROBOT system to precisely shape the bone surface of the knee for perfect alignment of implants. This system was used to register the surgeon's movements and restrict them to safe regions (active constraints).

Integrated Surgical Systems (ISS, Santa Monica, CA) created the first orthopedic surgical system in 1992, after six years of development at the IBM T. J. Watson Center and U.C. Davis [25]. ROBODOC became quite popular, being the first robot to be used in robot-assisted human hip replacement. It was rebranded to Curexo Robodoc in 2007 and to THINK Surgical in 2014. ISS used to lead the research and development of neurosurgical robotic devices as well. Their robot called NeuroMate—a stereotaxic targeting device for neurosurgery— was the first neurosurgical robot to get FDA clearance. Neuromate was originally designed based on an industrial robot, and it was developed market-ready by Innovative Medical Machines International, Lyon, France. It successfully used image guidance based on preoperative 3D medical images such as Computed Tomography (CT) or Magnetic Resonance Imaging (MR). In the beginning of each operation the robot held a calibration cage and X-ray imaging was used to register the patient's coordinate system to the robot. After the registration, a PC controlled the robot based on the preoperative plan. The robot could take advantage of movements with 5 DoF. When the robot reached position, the neurosurgeon inserted the tool through the robot's guiding channel. With ± 1 mm accuracy, the NeuroMate robot enabled more accurate targeting in neurosurgery and therefore the surgeons could reach areas where

operation had been impossible before.

The first surgical robot for cardiac procedures was the ARTEMIS system. It was designed as a teleoperation and telepresence system for cardiac procedures. Since it was used for training, planning and executing different minimally invasive procedures was part of the design. ARTEMIS had two master–slave units and an endoscope guiding system. The slave side was designed in the way that the manipulators were pivoting around a predefined point aligned with the thoracic insertion point. After the initial setup, the surgeon was able to operate with 6 DoF instruments from the master console. The system also incorporated a 3D endoscopic vision system, which was operated from an additional joystick, separately from the other master manipulators [21].

Figure 4
The Artemis surgical robot [22].

After the groundbreaking work was done by these robots, other systems started to appear on the market, targeting specific fields undiscovered by previous solutions. Over 500 research projects were submitted leading to dosens of system prototypes [1]. Many of these systems were discontinued, and today more than 80% of the market is owned by Intuitive Surgical's da Vinci.

3.1 The Concept of Robotic Assisted Microsurgery

The concept of Robotic Assisted Microsurgery (RAMS) plays an important role in the design and development of robotic surgical devices. Originally coming from the NASA laboratories, it appeared as a side-innovation for telerobotics and microsurgery. It was S. Charles who first proposed the idea of using telerobotic systems as a tool in microsurgical procedures [17]. The project was initiated at the NASA JPL, more precisely within the Advanced Teleoperation Lab (ATOP), which, at that

time, was directed by A. Bejczy. RAMS was developed around 1994, inheriting the basic concepts of the NASA telesurgical technologies (Fig. 5).

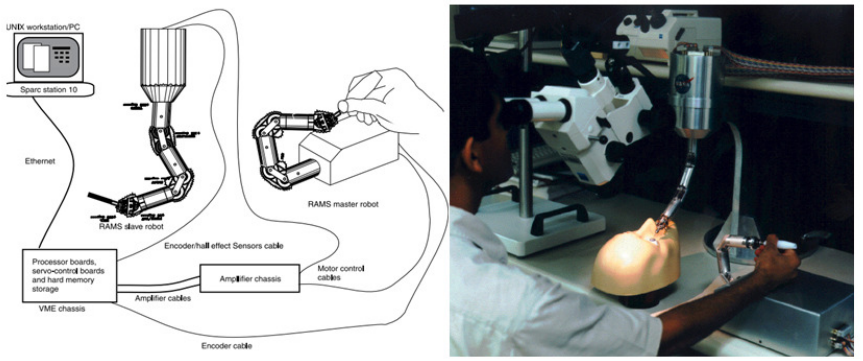


Figure 5
The concept and realization of the RAMS system at NASA JPL ATOP [18, 23].

The developed robot was built up from a 3-axis wrist that was mounted on a torso–shoulder–elbow assembly, capable of 6 DoF motions. The robot included a manipulator, whereas the engineers were aiming to create similar kinematic layout for both master and slave sides of the instrument. The use of RAMS instruments was later extended for carrying out coronary artery anastomoses [18], and soon successful animal trials were documented. Other studies pointed out that robotic devices could also be used for endoscopic cardiac surgical procedures, including endoscopic anastomosis [19, 20].

4 Basic concepts in Computer-Integrated-Surgery

With the spread of surgical robotics initiatives, a new domain was established. CIS was born based on the understanding that engineering can provide surgeons with tools, making procedures safer, less invasive and eventually more efficient. Arguably, the appearance of digital medical imaging brought the biggest impact, when it finally replaced the previously used analog technology. Not only did it provide the capability to do fast intra-operative scans, but also made pre-operative imaging more informative. One great example of this is the appearance of CT devices. Before CT was used, surgeons only had access to projective X-ray images; with CT, a clear 2D slice of the human body could be obtained, and even the 3D reconstructions of the anatomy became achievable. From computer-based imaging and other fields such as genetics, blood testing, etc., we can generate a database of patient specific information, which was previously inaccessible. This patient specific information made it possible for CIS to improve, and fundamentally change treatment planning. Without accurate imaging, it was only possible to guess the underlying anatomy with physical examination, and often surgeons needed to face unexpected

situations after the operation started. Today, procedures can be planned on accurate, patient specific data gained from imaging (Fluoroscope, US, or even MR) and other modalities. This enables surgeons to decide where and when to operate, and to provide the best possible quality of life for the patient as a result of the surgery. As devices became more portable, it was possible to move them into the Operating Room (OR), where they could be used for intra-operative imaging, or to provide assistance to the procedure. Data collected from these devices can be used to adjust or re-plan the procedure during the execution. Finally, after the procedure, the data collected could be used for statistical analysis, which could provide important information on performance and become a basis of further development. This loop of Computer-Integrated-Surgery is shown on Fig.6. Pretty much all of the concept robots listed above choose one or another approach to optimize its use within the frames of CIS. Over the years three, distinct strategies prevailed, introduced in the following.

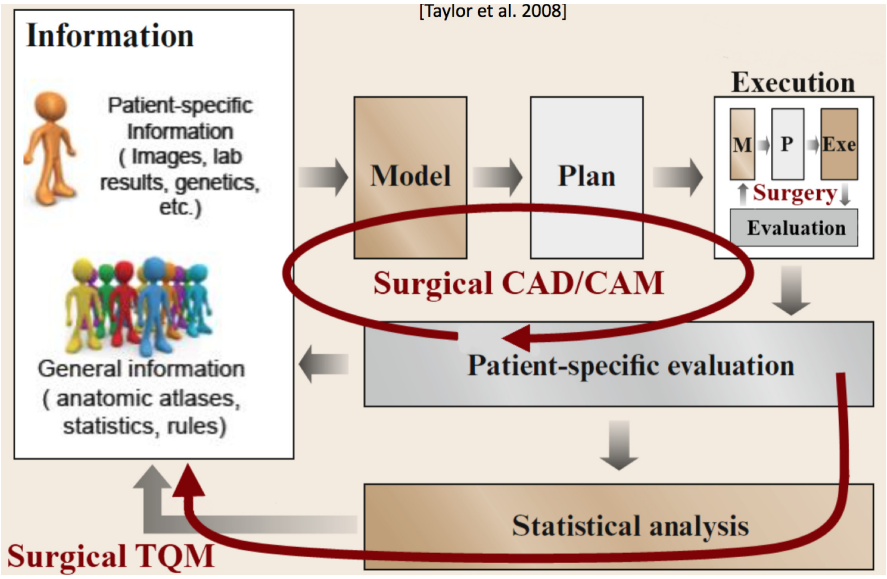


Figure 6
Basic concept of Computer-Integrated-Surgery [27].

4.1 Types of Robotic Assisted Surgery

Robotics can assist surgeons in various ways, and it is probably an impossible task to find a uniform characteristic for all cases besides that engineering work enhances the performance of the surgeon. In the following, we present the three main approaches surgical robotics follows nowadays giving an example for each one trough a robotic platform used in surgical practice.

4.1.1 Robotic telesurgery (Human-in-the-loop control)

Robotic telesurgery devices are applications where the surgeon remotely controls the robot performing the operation (teleoperation). The da Vinci system—which is definitely the most successful robotic platform in existence—follows this approach. The surgeon works at a console, his/her hand movements are perceived by the master manipulator, and transmitted onto the patient side manipulator, holding the actual tool, where the fingers (thumb and index) are mapped to the tip of the tools. This has a major ergonomic benefit, since statistics have shown that the incidence of chronic back pain increased significantly among surgeons performing conventional laparoscopic procedures, which is due to the fact that during these procedures, the surgeon stands next to the patient and forced to maintain an anatomically inadequate position. Letting the surgeon work at a specially designed workstation provides a great opportunity for engineering an ergonomic environment, lowering the physical and mental stress on the surgeon. Applications based on teleoperation methodology can also incorporate functions to cancel out the surgeons tremor, or to provide magnification to the view of the field the surgeon is interested in. However, these platforms do not incorporate much automation, as the feedback loop of the control requires the continuous involvement of the human operator. This is also a benefit for the manufacturer from the regulatory point of view, since it was easier to convince the authorities about the systems' safety, while the human surgeon is always in charge of the robot's motion. Teleoperation would be a preferred way to do interventions in remote areas where specialists are not available, or could not reach the patient in time. Extreme environment research such as a space expeditions or military applications are driving this field, as they face the same difficulties. In these situations teleoperation would be required to operate with large physical distances between the master and slave sides, and time delay would become a significant issue, probably compromising the success of the surgery. Several algorithms were proposed for the control of these systems, but there is no clear solution for the problem yet [26]. The main reason behind this is that when time delay becomes noticeable, even if the system stays stable, the operation becomes harder and takes significant processing time to complete. Time delay can be shortened by using dedicated channels, however these are costly, and therefore not applicable beyond some special cases. Another solution is to model the behavior of the tissue on the master side, and later correct the model based on the data received from the patient side sensors. Unfortunately the complexity of the operating field can not be sufficiently modelled today.

The da Vinci Surgical System

Intuitive Surgical Inc. is the manufacturer of the da Vinci Surgical System, which was created for MIS procedures. The da Vinci was the first telerobotic system for human use that was cleared by the Food and Drug Administration (FDA) in 2000. Approval of the system included urological, thoracoscopic, gynecological and laparoscopic procedures, both for adult and pediatric use. The prototype Mona was involved in the first trials in 1997, while the first closed-chest, multi-vessel cardiac bypass procedure was performed using this robot in 1999. Originally, the system was designed for carrying out cardiac procedures, such as beating heart surgery, yet the real popularity of the system was brought by the rapid adaptation in radi-

Figure 7
Generations of the da Vinci Surgical System from 1999 to 2014. Da Vinci Classic, S, Si and Xi systems. Image credit. Intuitive Surgical Inc.

cal prostatectomy and hysterectomy procedures. The recent growth of trans-oral robotic surgery has also brought new prospects to the development of the da Vinci system.

The da Vinci is a typical master–slave system, consisting of a master side console, a patient side set of robotic arms and a visualizing system. The basic architecture

has remained the same, following the initial concepts from the first generation. The four generations of the da Vinci are the Classic, S, Si and the Xi, as shown in Fig. 7. The fundamental concept has not changed since the NASA's '70s plans, while the hardware and software capabilities have improved dramatically.

The master side manipulators are serving as the interface for the surgeon, allowing him/her to manipulate the tools virtually. A 3D display system is also integrated at the master side, showing the surgical field recorded by the stereo camera endoscope. The connection of two master consoles is also possible enabling efficient training and teaching. In all generations, the maximum number of patient side manipulators is 3, extended with an additional robotic arm holding the endoscopic camera. The arms are copying the movements of the surgeon's arms, following them in real-time. This can only be achieved if the surgeon is in the same room with the patient in order to keep the latency (time delay) low. To achieve this, Intuitive used the technology of pre-existing systems, which is supported by the fact that after its incorporation, it closely observed the RAMS patents for force reflection surgery from the California Institute of Technology. Later one of the first da Vinci was installed at one of the NASA facilities. In the same year, the first closed-chest beating heart cardiac bypass surgery was performed using the Zeus robot, the great competitor at that time.

4.1.2 Image-Guided Surgical Robotics

Medical imaging went trough tremendous improvements when it moved from analog technologies to information based methods. Having accurate digital images of the patients opened up new possibilities in image guidance, allowing more accurate targeting in fields like oncology or neurosurgery. Unlike telesurgery, the image–guided approach typically incorporates autonomous execution of the whole or part of the surgical plan, for which the machine does not require a human operator in the control loop. Real-time intra-operative tracking of the tools and the patient, achieved with either 3D cameras, electromagnetic trackers or other modalities [29]. Accurate navigation can only be achieved if the reference image is registered to the physical world and the transformations between the different coordinate frames are known. The most commonly used approach is the rigid frame registration, registration for distorted images is achieved by numeric approximation. The ROBODOC system is (cleared by the FDA for total knee and hip replacement surgeries) a typical example. It uses pre-surgical 3D planning based on the patients CT scan. Given the surgical plan, the ROBODOC is able to execute the bone milling task with submillimeter accuracy. In this application the robot is actively milling and the surgeon only provides supervision to the procedure.

4.1.3 Cooperative or Hands-on Surgical Robotics

The human hand is a remarkable multi-purpose grasping tool with amazing capabilities in terms of dexterity. This is why surgeons tend to prefer using it during the surgery, rather than just relying on some limited human–machine interfaces. Another argument against the telesurgical approach is typically the loss of certain

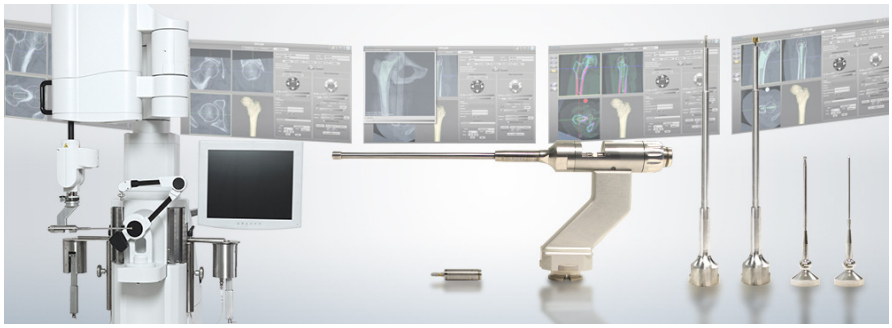


Figure 8
The THINK Surgical system from Curexo Technology Corp. Photo courtesy by Curexo Technology Corp.

sensory feedback from the operational site. In the meanwhile, for some special surgical tasks, more sensitivity or a wider range of motions are required, beyond that of our physical construction can provide. Cooperative control offers a plausible alternative, enabling the surgeon to use a robotic tool as a slave device right at the spot of the intervention, directly controlling it with its hands. This is typically achieved with force/torque sensors and control [32]. Eye surgeons face this problem in their everyday work. During a procedure it is required to access areas only a couple of cell layers thick, with a resistance impossible to feel, when the procedure is performed manually. During these tasks, ophthalmologists are usually left to use visual feedback only. The Steady Hand device (Fig. 9.), developed at the Johns Hopkins University (Baltimore MD, USA) is intended to tackle this problem. It is a cooperatively controlled robot, where the surgeon moves the tool attached to the robotic platform by exerting forces on it. The Steady Hand robot is able to cancel out some of the tremor and unwanted motions from the hand movement. It also incorporates sensitive pressure sensors for measuring the small changes in tissue resistance. By amplifying these forces, it can give a sensible feeling to microsurgery which did not exist during traditional procedures. It can also provide visual guidance and audible sounds to better assist the surgeon.

Bordering image-guided and cooperative control paradigm, the RIO system from MAKO (acquired by Stryker in 2015). It uses image guidance to control the milling tool mounted at the end of a robotic arm, which is eventually mounted on a movable cart. Unlike the ROBODOC, the RIO system does not carry out the actual milling, but instead it creates resistance on predefined boundaries—also called virtual fixtures—allowing the surgeon to access only a restricted area of the operation field. With this mechanism, the RIO system can protect from unwanted errors, but, in the case of a registration, planning or system error, the surgeon can interfere as he/she is still in complete control of the instrument.

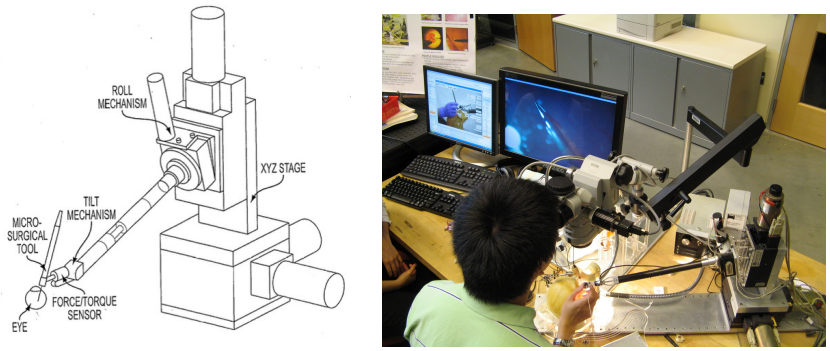


Figure 9
The schematic view of the Steady Hand device, and the experimental setup at the Johns Hopkins University.

5 Surgical Robotics in Extreme Environments

DARPA and NASA remained deeply interested in telerobotic surgery throughout the past decades. Surgical procedures in space, as it was pointed out in the early trials by NASA, carry many difficulties and challenges, and the signal latency is one of the many problems to be solved [33]. In order to investigate the possibilities of operation in weightlessness, a Soviet team carried out the first animal surgery of this kind during a parabolic flight in 1967. Thirty years later, an experiment was performed on a rat model during the STS 90 Neurolab Mission, however, the abdominal cavities were not treated in this procedure. The first zero gravity surgical experiment by NASA was a suturing task performed by an M7 robot (developed by SRI, Sacramento, CA), during hyperbolic flight. The task was a classical, teleoperated knot tying, where the slave and master sides were equipped with acceleration compensation. The inconclusive results showed that human surgeons were better at adaptation than the robots, in the case of extreme conditions.

The European Space Agency (ESA) took part in the first European initiative for investigating surgical procedures in weightlessness on board of a Zero G plane in 2003. In 2006, a cyst was removed from a patient's arm during parabolic flight. The further planned projects for telesurgery experiments during parabolic flights have been postponed ever since.

When a procedure in space is taken place beyond Earth's orbit, numerous difficulties arise with telesurgical robotics. Theoretically, communication between the master and the slave side is propagating at the speed of light. However, this speed already causes a latency of more than 1 second on the Earth–Moon distance. Inter-planetary distances, such as the Earth–Mars relation increases the delay to 44 seconds, which can become even more because of the significant computational efforts in compressing and decompressing video data.

Extreme telesurgery experiments under terrestrial conditions have long been investigated by NASA. Experiments were primarily carried out during the NASA Extreme

Environment Mission Operations (NEEMO). This is the world's only permanent undersea laboratory, near Florida Keys, 19 meters below the water surface. State-of-the-art computers and equipments are used by astronauts, biologists and engineers, with the intention of finding the consequences of operating in extreme environments, simulating the tasks of space missions.

The Zeus robot was involved in the 2004 series of experiments, during the 7th NEEMO project, focusing on telementoring and teleoperation. There were five test conditions investigated: ultrasound-guided abscess damage, ultrasonic examination of abdominal organs, cystoscopy, laparoscopic cholecystectomy and renal stone removal. Two astronauts, a physician and a surgeon were involved in this undersea experiment, while the robot was controlled remotely from the Centre for Minimal Access Surgery, London, UK. The distance was 2500 kilometers, and the latency was manually increased to a value between 100 and 2000 milliseconds. Results showed that the crew members with less or no expertise were quite successful in performing the interventions based on the guidance of the telementors. The effectiveness of teleoperation and telementoring was compared, and an important deduction was made: teleoperation procedures were more accurate, but took significantly more time and effort than telementoring.

In 2006, during the 9th NEEMO project, an assembly task was given to the crew for installing an M7 surgical robot for performing abdominal surgery in a virtual environment. The latency was set to 3 seconds. As the intention was to replicate a Lunar environment, the robot was intended for manipulating rock samples from the bottom of the ocean. Telementoring tasks were also included, focusing on fatigue and other stressor effects on the performance of the crew.

The 12th NEEMO project investigated the feasibility of telesurgery using the Raven and M7 robots, as shown in Fig. 10. The structure of the crew was the same as in the 7th project, performing suturing tasks, measuring the effectiveness of teleoperating surgeons located in Seattle, Washington. Commercial internet connection was used, with a relatively short latency time.

The NEEMO projects conclusion stated that the high quality tactile and visual feedback are equally important in order to create reliable teleoperation environments. High quality video streaming, however, is difficult to achieve due to the limited bandwidth. The bandwidth of 50 Mbps is already available on the International Space Station, although Mars missions are still calculated with the half of this amount by 2025.

Discussion & Conclusion

Surgical robotics originated from the need to operate over long distances, where a medical doctor could not get physical access to the patient. This requirement emerged first at NASA missions for space expeditions, but was quickly picked up by the military as well, where DARPA funded the key research projects on telesurgical robotics. The most successful robotic system for surgery, the da Vinci grew out of these early results in 1998, successfully combining the advantages of various prototypes. Since then, technology continued to improve, and instead of the military applications, the private sector has become the driving force of the surgical

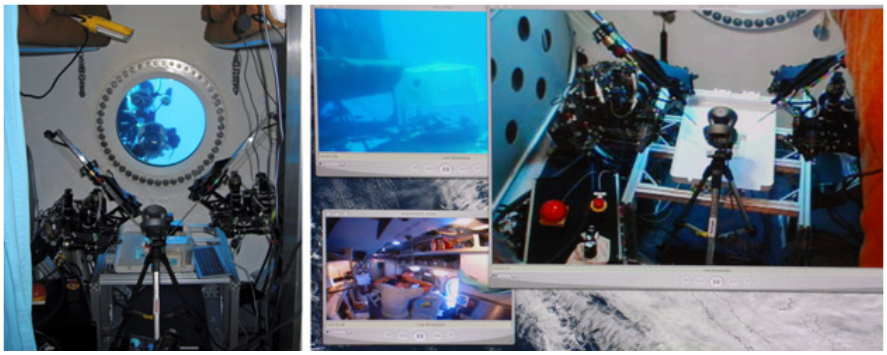


Figure 10
Underwater experiments at NASA Aquarius station during the NEEMO 12 project: setup and remote view of the Raven robots, developed at the UNiversity of Washington (Seattle, WA). Photo credit: NASA.

robotics industry, which is estimated to be around $5 billion per year. Along with the constant improvement in telesurgery, other robotic devices appeared for enhancing surgical capabilities. Three major approaches can be defined within the field of Computer-Integrated Surgery, these are telesurgery, image-guided surgery and cooperatively controlled surgical robotics. In all of these three fields, an important trend is that pre- and intra-operative information—in the form of imaging, physiological data collection, etc.—are playing an increasingly significant role during the procedure, and are enabling robotic systems to gain more autonomy. The field is rapidly changing thanks to the hundreds of research teams focusing on relevant projects, and nowadays, major players, like Google are also entering the arena. The computing capabilities of modern ICT devices are to allow the usage of more sophisticated systems, however, overarching regulations and standards in the field are still missing. Surgical robotic specific standards currently under development will make it possible for the industrial players to better design their systems, to be able to prove their safety and accuracy to the authorities. As for today, most applications keep the human operator active in the control loop, enabling the robot only to enhance the surgeons' capabilities. Autonomous task execution on Earth and in space remains a topic of the future.

Acknowledgement

Tamás Haidegger is a Bolyai Fellow of the Hungarian Academy of Sciences. This work has been supported by ACMIT (Austrian Center for Medical Innovation and Technology), which is funded within the scope of the COMET (Competence Centers for Excellent Technologies) program of the Austrian Government.

References

[1] M. Hoeckelmann, I. J. Rudas, P. Fiorini, F. Kirchner and T. Haidegger, "Current Capabilities and Development Potential in Surgical Robotics" *International Journal of Advanced Robotic Systems*, vol.12(61) pp. 1–39., 2015

[2] Á. Takács, L. Kovács, I. J. Rudas, R. E. Precup and T. Haidegger, "Models for Force Control in Telesurgical Robot Systems" *Acta Polytechnica Hungarica*, vol.12, issue 8, pp. 95–114., 2016

[3] C. R. Doarn, A. E. Nicogossian and R. C. Merrell, "Applications of telemedicine in the United States space program" *Telemed J*, vol. 4, no. 1, pp. 19–30, 1998.

[4] K. Corker, J. H. Lyman and S. Sheredos, "A preliminary evaluation of remote medical manipulators" *Bull Prosthet Res*, issue 16, vol 2. pp. 107–134, 1979.

[5] A. D. Alexander, "Impacts of Telemation on Modern Society" in *On Theory and Practice of Robots and Manipulators*, Springer Berlin Heidelberg, 1972, pp. 121–136.

[6] A. D. Alexander, "Impacts of telemation on modern society," Jan. 1973.

[7] Wikipedia: Robot-assisted Surgery

[8] O. Lechky, "Worlds first surgical robot in B.C.," *The Medical Post*, vol. 21, no. 23, pp. 92–93, Nov. 1985. Available: `http://www.brianday.ca/imagez/1051_28738.pdf`

[9] Powered surgical retractor. United States Patent 5271384. Available: `http://www.freepatentsonline.com/5271384.html` Accessed: Nov. 4. 2015

[10] Medizinischer Roboter, EP Patent App. EP19,860,106,362, `https://www.google.com/patents/EP0201883A2?cl=de`, 1986, Accessed: Nov. 5. 2015

[11] Patient limb positioning apparatus, US Patent US4807618. Available: `http://www.google.com/patents/US4807618` Accessed: Nov. 5. 2015

[12] Y. S. Kwoh, J. Hou, E. A. Jonckheere and S. Hayati, "A robot with improved absolute positioning accuracy for CT guided stereotactic brain surgery," *IEEE Trans. on Biomedical Engineering* vol. 35, no. 2, pp. 153–160, Feb. 1988.

[13] J. C. Bowersox, A. Shah, J. Jensen, J. Hill, P. R. Cordts and P. S. Green, "Vascular applications of telepresence surgery: initial feasibility studies in swine," *Journal of Vascular Surgery*, vol. 23, no. 2, pp. 281–287, Feb. 1996.

[14] A. Rovetta, R. Sala, F. Cosmi, X. Wen, S. Milanesi, D. Sabbadini, A. Togno, L. Angelini and A. K. Bejczy, "A New Telerobotic Application: Remote Laparoscopic Surgery Using Satellites and Optical Fiber Networks for Data Exchange" *The Intl. J. of Robotics Research* vol. 15, no. 3, pp. 267–279, Jun. 1996.

[15] R. M. Satava, "Virtual reality, telesurgery, and the new world order of medicine," *Journal of Image Guided Surgery*, vol. 1, no. 1, pp. 12–16, 1995.

[16] P. Garcia, J. Rosen, C. Kapoor, M. Noakes, G. Elbert, M. Treat, T. Ganous, M. Hanson, J. Manak, C. Hasser, D. Rohler and R. Satava, "Trauma Pod: a semi-automated telerobotic surgical system," *Int. J. Med. Robotics Comput. Assist. Surg.*, vol. 5, no. 2, pp. 136–146, 2009.

[17] H. Das, T. Ohm, C. Boswell, R. Steele and G. Rodriguez, "Robot-Assisted Microsurgery Development at JPL," in *Information Technologies in Medicine*, M. Akay and Andyrsh, Eds. John Wiley & Sons Inc., 2001, pp. 85–99.

[18] E. R. Stephenson, S. Sankholkar, C. T. Ducko and R. J. Damiano, "Robotically assisted microsurgery for endoscopic coronary artery bypass grafting," *The Annals of Thoracic Surgery*, vol. 66, no. 3, pp. 1064–1067, Sep. 1998.

[19] P. D. Le Roux, H. Das, S. Esquenazi and P. J. Kelly, "Robot-assisted microsurgery: a feasibility study in the rat," *Neurosurgery*, vol. 48, no. 3, pp. 584–589, Mar. 2001.

[20] J. Schiff, P. S. Li and M. Goldstein, "Robotic microsurgical vasovasostomy and vasoepididymostomy: a prospective randomized study in a rat model," *Journal of Urology*, vol. 171, no. 4, pp. 1720–1725, Apr. 2004.

[21] M.O. Schurr, A. Arezzo, G.F. Buess, "Robotics and systems technology for advanced endoscopic procedures: experiences in general surgery," *European Journal Cardio-Thoracic Surgery*, vol. 16, Suppl 2, pp. 97–105, 1999.

[22] H. Rininsland, "ARTEMIS. A telemanipulator for cardiac surgery," *European Journal Cardio-Thoracic Surgery*, vol. 16, Suppl 2, pp. S106–S111, 1999.

[23] S. Saraf, "Role of robot assisted microsurgery in Plastic Surgery," *Indian Journal of Plastic Surgery*, vol. 39, no. 1, pp. 57–61, 2006.

[24] S. E. Butner and M. Ghodoussi, "Transforming a Surgical Robot for Human Telesurgery," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 5, pp. 818–825, 2003.

[25] J. Rassweilera, J. Binderc and T. Frede, "Robotic and telesurgery: will they change our future?," *Current Opinion in Urology*, vol. 11, pp. 309–320, 2001.

[26] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[27] R. H. Taylor, A. Menciassi, G. Fichtinger and P. Dario, *Medical robots and systems*. In: B. Siciliano, O. Khatib, eds., *Springer Handbook of Robotics*. ch. 52 Berlin: Springer–Verlag, 2008.

[28] S.J. Harris, F. Arambula-Cosio, Q. Mei, R.D. Hibberd, B.L. Davies, J.E. Wickham, M.S. Nathan and B. Kundu, "The Probot—an active robot for prostate resection.", *Proc. Inst. Mech. Eng. H.*, vol. 211(4), pp. 317–325., 1997.

[29] Y. Lu, C. Li, M. Liu, J. Fritz, J. A. Carrino, L. Wu and B. Zhao, "MRI-guided stereotactic aspiration of brain abscesses by use of an optical tracking navigation system," *Acta Radiol*, vol. 55, no. 1, pp. 121–128, 2014.

[30] A. P. Schulz, K. Seide, C. Queitsch, A. von Haugwitz, J. Meiners, B. Kienast, M. Tarabolsi, M. Kammal and C. Jrgens, "Results of total hip replacement using the Robodoc surgical assistant system: clinical outcome and evaluation of complications for 97 procedures," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 3, no. 4, pp. 301–306, 2007.

[31] A. N. Supe, G. V. Kulkarni, P. A. Supe, "Ergonomics in laparoscopic surgery," *Journal of Minimal Access Surgery*; vol.6(2) pp. 31–36, 2010

[32] T. Haidegger, B. Benyó, L. Kovács and Z. Benyó, "Force Sensing and Force Control for Surgical Robots," *Proc. of the 7th IFAC Symposium on Modelling and Control in Biomedical Systems*, Aalborg, Denmark, Aug. 12–14, pp. 413–418, 2009

[33] T. Haidegger, J. Sándor and Z. Benyó, "Surgery in space: The future of robotic telesurgery," *Surgical Endoscopy* vol. 25(3) pp. 681–690, 2011

Mechatronic Assistance for Surgical Applications

Gernot Kronreif

Austrian Center for Medical Innovation and Technology
ACMIT Gmbh
Viktor Kaplan-Strasse 2, A-2700 Wiener Neustadt, Austria
e-mail: Gernot.Kronreif@acmit.at
www.acmit.at

Abstract: Medical Robotics is an interdisciplinary field that focuses on developing electromechanical devices for diagnosis and therapy. The long-term goal of this application area is to enable new medical techniques by providing new capabilities to the physician or by providing assistance during surgical procedures. The field has tremendous potential for improving the precision and capabilities of physicians when performing surgical procedures, and thus it is believed that the field will continue to grow. On the other hand – and unlike the area of factory robotics, which grew rapidly during the 1970s and 1980s – Medical Robotics has not yet gained widespread acceptance. There are still many technological challenges and research topics – including both the development of system components and the development of systems as a whole. The paper aims to give a short overview on the state-of-the-art of Medical Robotics in selected application areas, to highlight some open problems e.g., system architecture, user interfaces and safety issues, and finally to show some examples for medical robotic developments from Austria.

Keywords: Medical Robotics; Computer-Aided Surgery; Surgical Mechatronic Assistance

1 Introduction

The field of Medical Robotics is relatively new, with the first recorded medical application of a robot – i.e., image-guided orientation of a needle guide for biopsy of the brain – occurring in 1985 [1]. About the same time, research groups in Asia, Europe, and the USA began investigating other medical applications of robotics [2]. Currently, there are a large number of research laboratories, but only a few commercial ventures active in the field of Medical Robotics. After a short period of recovery from first negative experience, the topic once again is growing rapidly and dedicated sessions on Medical Robotics can now be found at almost any medical related conference.

Compared to many other fields of medical technology, medical robots are still in their infant state and many research questions are still open for investigation and improvement of the technology. Based on the experience of previous research and clinical practice, an appropriate operating concept is essential for the success of such a system. In the context of minimally invasive procedures, current robots quite often are under fully control of surgeons by using a tele-operation setup in which the human operator manipulates a master input device and patient-side robot follows the input [9]. With such a setup, the robot allows the surgeon to have dexterity inside the body of the patient, scale down motions from normal human dimensions down to very small distances, and filter out tremor. The next generation of such robots will provide with a more intuitive connection between the operator and the instrument tips, with physical enhancements such as "no-fly" zones around delicate anatomical structures as well as better recognition of surgical motions and patient state to evaluate performance and predict health outcomes. In parallel, more autonomy will be implemented in such surgical assistants – e.g., in order to take over routine tasks or to allow the surgeon to concentrate to key actions during surgery rather than be burdened with operation of an additional device. For medical procedures which can be planned ahead of time and executed in a reasonable predictable manner, robot operation follows a concept which is often referred to as "Surgical CAD/CAM" (in analogy to Computer-Aided Design and Computer-Aided Manufacturing known from industrial manufacturing systems). One example is in the area of interventional radiology, where robots have the potential to position a needle guide, based on pre-operative planning in order to support the physician to push the needle through this guide. As imaging, tissue modeling, and needle steering technology improve, future systems are likely to become more highly integrated and will actively place the needles and therapy devices along paths that cannot be achieved by simply aiming a needle guide.

1.1 State-of-the-Art in Medical Robotics

Literature shows a couple of developed setups – and products – which have also been used clinically with moderate success. There is only a small number of commercial companies selling medical robots and thus the total number installed is relatively small (and dominated by one particular setup) [9]. Unlike the area of factory robotics, which grew rapidly during the 1970s and 1980s, Medical Robotics has not yet reached a critical mass. However, it is believed that the benefits of medical robots will become increasingly clear, which finally will lead to a continued rise of their use in clinical routines.

Some robots function as surgical assistants in orthopedics, and others can be used as a surgeon's "third hand" for moving the camera during minimally invasive procedures. Others exist to perform or facilitate tele-surgery, tele-monitoring, tele-mentoring, or true tele-presence instruction. Still other robotic devices perform or

assist with image guided interventions. Further possible classifications of surgical robots can be based on their level of autonomy as in:

- *Passive* - In which the robot serves as a guide to the surgeon's hands;
- *Hands-on* - In which the surgeon operates the device in the surgical field, but where the robot controller supports the surgeon and/or helps to avoid wrong actions ("shared control");
- *Tele-operated* - In which the robot is in the surgical field but is explicitly controlled by a surgeon some distance (usually a few feet, i.e., within the operating room) away;
- *Automated* - In which the robot executes pre-operatively planned trajectories;

or based on surgical application fields (Neurosurgery, Orthopaedics, Urology, Maxillofacial, Radiosurgery, Ophthalmology, Cardiac Surgery, Interventional Radiology) [3].

The area of image guided robots has been one of the starting points for medical robot systems, with applications in orthopedic surgery (e.g., ROBODOC, CASPAR, SPINEASSIST, RIO) and for neurosurgical applications (e.g., MINERVA, NEUROMATE, ROSA). Very close to the aforementioned systems are robot setups for interventional radiology (e.g., PAKY+RCM, B-ROB, INNOMOTION, MAXIO, ISYS 1) and external radiotherapy (e.g., CYBERKNIFE). Another group of robots is the one of "hands-on" and/or remotely controlled ("master/slave") setups like RAMS, MICRON for micro-surgery, DAVINCI, ALF-X or camera holding assistants like AESOP, FREEHAND, SOLOASSIST, VICKY for laparoscopic surgery, and MAGELLAN for robot-guided catheter placement. A third category includes intracorporeal robot systems but only a few prototypes are described in literature at present (e.g., HEARTLANDER, EMUL).

Several databases are available via the Internet and provide a good overview about past and present medical robotic systems, like the "Surgical Robotics Blogspot" (http://surgrob.blogspot.com/).

2 Open Research Questions

It should be mentioned at this point, that only very few of the systems developed so far are being used in clinical routines. The main reason for this lack of success is not related to technical functioning, but to other factors, such as:

- Cumbersome use of robots (complexity, size, missing integration into clinical workflow) which hinders application in clinical routine;

- High cost for a robot system and operational cost, i.e., cost/benefit ratio is not satisfactory, in most cases;
- High setup time and effort (e.g., need for an additional person to operate the robot system);
- Limitation in portability and/or mobility;
- Unsolved safety issues.

To demonstrate the potential benefits of medical robotic systems future research is required in both areas development of system components and the development of systems as a whole. In terms of system components, research topics can be outlined as follows.

System Architecture - For Medical Robotics to evolve as an independent field and to decrease the cost and difficulty of developing prototype systems, the establishment of a suitable system architecture is an enabling step. Development in systems architecture should emphasize modularity in mechanical design, control system electronics, and software. Middleware concepts such as ROS [10], OROCOS [11] for robot systems and IGSTK, 3DSLICER/PLUS [12] and others for image-guided surgical setups in broader context help for the prototyping phase – extension of such concepts toward development of medical products still needs to be investigated in more detail.

Software Design - The development of a software environment for Medical Robotics is a significant challenge. Research should be based – along with the system architecture mentioned above – on software frameworks geared to the medical environment and especially considering safety and robustness. While this software environment would still need to be customized for different surgical procedures, research on new applications would at least have a better starting point for the development work.

Mechanical Design - Novel mechanical designs are needed to improve the utility of robotics in medical procedures. It is the author's belief that special purpose mechanical designs are more appropriate for most applications. In particular, these designs should be safer, as they can be designed specifically for the medical environment and customized for different medical procedures. However, it should be noted that special purpose designs will not enjoy the same economies of scale as more general designs, which should be addressed by developing a set of reusable and configurable modules for dedicated designs.

Compatibility to Imaging Devices - With the increasing popularity of image-guided interventions, robotic systems need to be able to work within the constraints of various clinical imaging modalities. While these systems currently are still under the direct control of the physician most of the time, in the future they will increasingly be linked to these imaging modalities.

Safety Issues - Safety is a paramount concern in the application of a medical device and must be addressed appropriately to move the field forward. Safety issues have been discussed by many authors (e.g., [4] and [5]) – all of them coming to the conclusion that Medical Robotics is a completely different application from industrial robots, e.g., because medical robots must operate in cooperation with, and even more important directly *on* humans to be fully effective. Therefore suitable safety levels must be defined and discussed by the community at large. This opens the question about what guidance is available for robotic developers in order to support the development process. Are there any best-practice examples for basic orientation? As a matter of fact, there are some standards in place, which define the safety requirements for medical devices. But interpretation of these requirements and transfer into concrete actions is rather complicated, especially for developers lacking long-term experience in the field. A dedicated standard for robots in personal care, in particular for medical applications, is in preparatory stage and should give more guidance for development of such devices.

User Interface - Known surgical robotic setups are mainly dominated by rigid interaction modes such as tele-operation (co-manipulation) or 'autonomous execution under supervision'. The gap between both procedures is large and the choice quickly is towards tele-operation when a certain level of complexity is reached by the application (e.g., tissue movement during intervention). Occasionally some hybrid forms – mostly featuring pre-operatively determined additional guidance or constraints - can be encountered. But the many questions remain: What is a suitable user interface for a medical robot? The answer will certainly vary depending on the medical task for which the robot is being designed. It seems that medical robots will initially at least be more accepted by physicians if the physicians feel to be still in control of the entire procedure.

3 Selected Examples for Medical Robotic Systems

In the following some selected examples for medical robotic systems, in particular for needle-based procedures, are briefly described. All described systems have been developed under the supervision of the author and have already demonstrated their potential in practical use.

3.1 Robot-Assisted Biopsy: B-Rob I and B-Rob II

Basic problem for percutaneous interventions is to target a needle-based instrument through the skin and thus without direct sight exactly to the region of interest (e.g., a tumor). Different types of intra-operative imaging modalities, like ultra-sound imaging (US), computed-tomography (CT) or magnetic resonance

tomography (MR) are being used in order to support the procedure. Compared to a traditional (freehand) technique, robotics assistance potentially helps to improve the procedure. Such a robot could serve as the physician's "third hand", exactly maintaining the previous planned trajectory and (sometimes; especially in case of deeply situated target areas) helping to shorten the time associated with definition of the desired insertion point and angulation.

Prototype B-Rob I. Starting in 2001, a custom made robotic targeting device for interventional radiology – called B-Rob I – was developed by ARC Seibersdorf Research GmbH (Seibersdorf, Austria) together with the Departments of Diagnostic Radiology and of Biomedical Engineering and Physics, Vienna Medical University (Vienna, Austria). The main goal of the system was to improve diagnostic accuracy as well as to reduce the risk of possible complications while planning of an oblique trajectory. Planned clinical applications for this first prototype have been biopsies in the abdominal area using US and CT as intra-operative imaging modalities.

The planned use of the system with different CT-scanners as well as with US-systems led to a tailored 7 DOF (degrees-of-freedom) kinematic structure of the robot system. Delivery to the skin entry point was performed by a 4 DOF gross positioning system consisting of three linear axes in Cartesian configuration together with one additional rotational link. Final orientation of the needle was done by means of a dedicated "Needle Positioning Unit" (NPU) with its two linear DOFs. Another linear DOF with a limited stroke of 50 mm was moving the entire NPU to the patient's skin in a secure approach movement, i.e., with minimized velocity and force. Setting the needle orientation was strictly decoupled from movement of any axis of the gross positioning system. Remote center of motion ("pivot point") for angulation of the needle was maintained by the kinematic structure of the NPU in order to achieve the maximum of safety during the intervention. Registration of the robot system to the imaging space, as well as, measurement of the current patient position was performed by an optical tracker system (Polaris, NDI, Canada).

Performance of the complete system was evaluated in a series of in-vitro tests using a needle-penetrable phantom. Peas (diameter = 9.4±0.7 mm) were embedded within a custom-made gel-phantom, as targets. Biopsy needles were inserted according to a previous planning procedure and distance between actual needle tract and center of the target was evaluated. During this in-vitro study, the robot system could demonstrate highly sufficient accuracy for the medical application in question. Each of the biopsy targets (n = 40 for US-guidance; n = 20 for CT-guidance) could be reached straightaway. Radial deviation between the center of the target and the needle tract was measured with 2.18 mm ± 1.25 mm for US-guided interventions and 1.48 mm ± 0.79 mm for CT-guidance. [6]

Prototype B-Rob II. Apart from the very promising results concerning accuracy, in-vitro trials with B-Rob I have also shown that size and complexity of the setup would be a major barrier for use in clinical routine. A new prototype – called B-Rob II – thus should transfer the proofed concepts from B-Rob I into a practical clinical setup. Development started in 2004 in a cooperation between PROFACTOR Research and Solutions GmbH (Seibersdorf, Austria) and Medical Intelligence GmbH (Schwabmünchen, Germany) with the major goals to create a modular setup of a robot assistant for a broad variety of clinical applications as well as to significantly reduce the technical complexity of the system in order to reach an acceptable cost/benefit ratio for clinical application.

The new B-Rob II setup consisted of two 2DOF positioning modules with small size (dimensions of one 2DOF module: WxLxH = 100 mm x 150 mm x 30 mm). Connection of the two modules with two carbon "fingers" – forming a parallelogram structure - allowed needle angulation of ± 30° and 2DOF positioning of ± 20 mm with high accuracy (see Figure 1, left). For efficient gross positioning of the robot system, the modules were mounted on a passive 7DOF multi-functional holding arm (ATLAS Arm, Medical Intelligence GmbH).

A series of in-vitro studies was performed with B-Rob II, with a similar setup as described above. 50 biopsy procedures were successfully performed showing a calculated average targeting error of 1.05 mm ± 0.35 mm [7]. The design goal of providing with a modular setup could be achieved and demonstrated in different application setups. In cooperation with Johns Hopkins University (Baltimore, USA) the robot, after slight modification to extend the reach of motion from ± 2 cm in x-y-plane to ± 4 cm, has been configured for TRUS-guided brachytherapy of the prostate with very promising results, including a clinical study with 10 patients (see Figure 1, right). [13]

Figure 1
Robot system B-Rob II during in-vitro trials; Modified B-Rob II with extended range of motion in clinical trial for TRUS-guided prostate biopsy (courtesy of ACMIT Gmbh and Johns Hopkins University)

Together with the ENT department of University of Erlangen (Erlangen, Germany) the robot, after slight modification to extend the robot's payload, has been evaluated as camera holder for trans-nasal endoscopic applications (see Figure 2, left). [14] Together with two clinical partners – neurosurgical department of Wagner-Jauregg hospital in Linz (Linz, Austria) as well as neurosurgical department of Krankenhaus Karlsruhe (Karlsruhe, Germany) – the robot system also has been successfully evaluated for brain-biopsy applications (see Figure 2, right). [15]

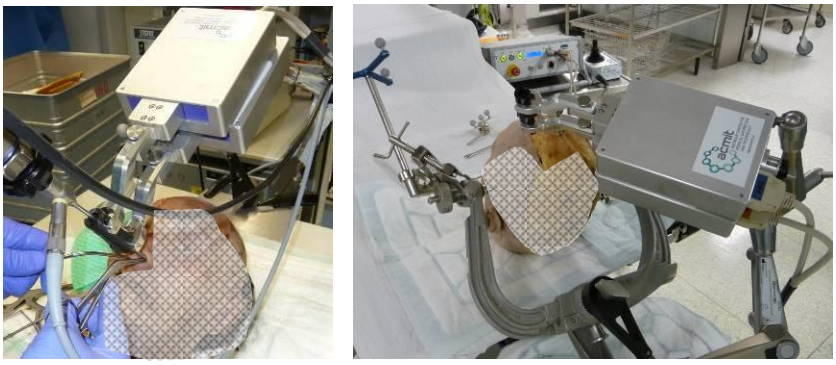


Figure 2
Modified B-Rob II with extended payload in cadaver trials for the use as tool holder in ENT and neurosurgery (courtesy of ACMIT Gmbh, Austria)

3.2 Robot System iSYS 1

In close cooperation with the iSYS Medizintechnik GmbH (Kitzbühel, Austria) the B-Rob II robot system has been redesigned for use as remote-controlled needle holder for interventional radiology. Dedicated workflow and setups for two imaging modalities, cone-beam CT (CBCT) and classical computed tomography imaging (CT), have been developed and have successfully passed CE certification.

Similar to the B-Rob II concept, this robot setup consists of a 7 DOF passive "Multi-Functional Arm" (MFA) for pre-positioning of the 2x2 DOF active "Robot Positioning Unit" (RPU). The general kinematic concept for the active robot is based on a parallelogram mechanism, with two parallel "Needle Guide Extensions" (NGE) showing relative linear movement and being connected to each other by means of a dedicated "Needle Guide Adapter" (NGA). The applied highly integrated design results in a small footprint of the robot (WxLxH: 120 mm x 200 mm x 70 mm) which allows the use of the system also inside the CT-gantry without kinematic restrictions. The range of motion (ROM) for the active robot results in +/- 20 mm for translation in a x-y-plane (for fine adjustment of the needle entry point) and +/- 30° for needle angulation. The use of high precision

components for the mechanical system (anti-backlash gear stage, high-precision ball screw system and monorail guides, etc.) results in a positioning resolution of about 5 µm and/or an angulation resolution of about 1/100 degree. For guidance of the tool, i.e., the needle, a set of "Needle Guide Inserts" (NGI) is available for the NGA. With these NGIs it is possible to have precise guidance of the needle during insertion, and release it from the robot's NGA, by pulling out the NGI and opening it. Beside of the RPU and the MFA, the robot setup consists of a set of adapters in order to securely mount the MFA to the patient supporting table ("Table Top Adapter" – TTA). A "Control Unit" (CU) and a dedicated "Human Control Unit" (HCU), both of which prepared for attachment to the side rail of the table (see Figure 3, left), complete the setup.

"Registration" of the robot to the acquired images in general requires a structure which can be easily and stably identified in the pre-/intra-operative images as well as in robot coordinates. For the CBCT-workflow, a "Needle Guide Marker Insert" (NGMI) has been designed which includes two small metal rings in line with the needle insertion axis. By moving the imaging plane of the CBCT into a "bull's eye view" direction (i.e., into direction of the trajectory) the NGI can be exactly aligned according to the planned needle trajectory by remotely bringing the aforementioned rings into concentric position. For the available CT-workflow, two dedicated marker structures, equipped with 10 mm glass balls in exactly known positions, are mounted on the robot. After acquisition of a 3D data set (which is also used to plan the needle trajectory), a fully automatic registration process of the dedicated planning software "RoboNav" (iSYS Medizintechnik GmbH, Austria) segments the registration marker and exactly calculates the robot position relative to the patient.

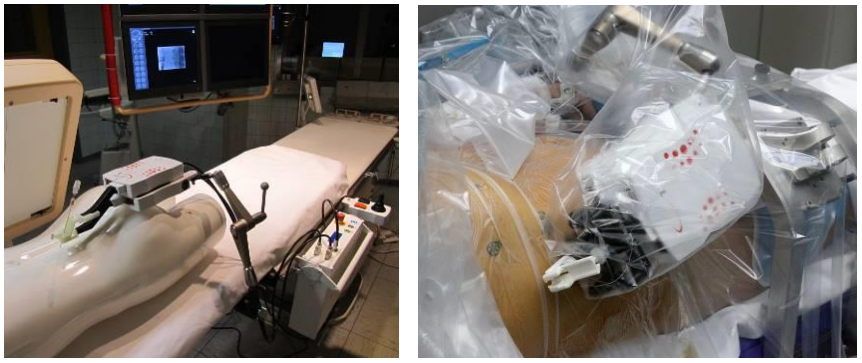


Figure 3
Robot system "iSYS 1" in CBCT configuration and in CT configuration in clinical setup (courtesy of iSYS Medizintechnik GmbH, Austria)

The iSYS 1 robot system now is in routine clinical use in several hospitals around the world with applications ranging from classical biopsy procedures over tumor ablation with different procedures (RFA, IRE, microwave, cryo-ablation) to

orthopedic procedures (see Figure 3, right). In addition, a patient study in cooperation with the Institute for Neurosurgery of the Medical University Vienna (Vienna, Austria) is currently in progress in order to evaluate the suitability of the iSYS 1 robot system for a set of neurosurgical procedures [16].

3.3 VIRTOBOT – Robot System for VIRTOPSY Procedures

The concept of Virtopsy was born from the desire to implement new techniques in digital imaging (CT/MR), three dimensional (3D) surface scanning, photogrammetry, and post-mortem minimally invasive biopsy for the benefit of forensic science. The aim is to establish an observer independent, objective and reproducible forensic assessment method with digitally storable results. In order to support such procedures a robotic system, called VIRTOBOT, was developed in cooperation with the Institute for Forensic Medicine in Zurich (Zurich, Switzerland). The entire system consists of a Multi-Slice CT scanner (Somatom Emotion 6, Siemens, Germany), a modified heart-lung machine for post mortem angiography, as well as, the VIRTOBOT robotic system which operates different system tools. Additionally, a high precision optical tracking system (Polaris, NDI, Canada) is integrated to increase the placement precision of automatic tissue sampling.

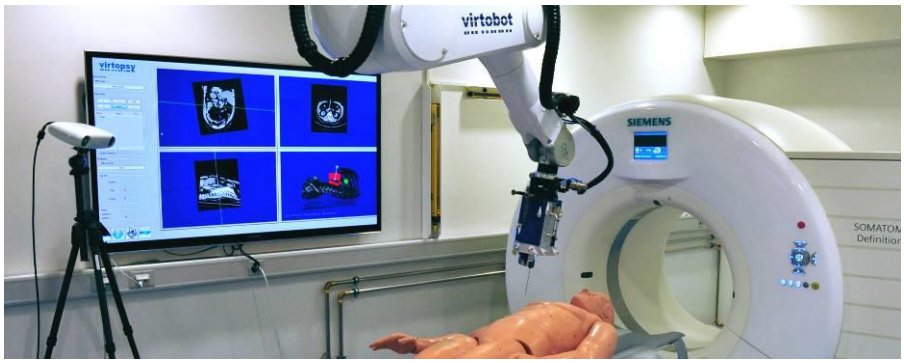


Figure 4
VIRTOBOT system on biopsy configuration (courtesy of IRM Zurich, Switzerland)

The VIRTOBOT comprises a Stäubli TX90L robotic arm and a 4 m external linear axis mounted on the ceiling in parallel to the CT table axis. Several safety precautions have been implemented into the VIRTOBOT system cell beside the typical emergency stop switches. Safety light curtains (Safety Class 4) ensure redundant safety during robot operation by triggering an immediate emergency stop of the complete system when being interrupted. Two further safety light curtains (Safety Class 2) are mounted between the robot tool and the CT-scanner in order to avoid collisions with the imaging system.

Complete workflow of the system and coordination of the application software packages (navigation and planning software, surface scan and photogrammetry software, robot control) is organized by the VCC software (Virtopsy Control Center). The VCC links diagnostic information and following process data of photogrammetry and surface scanning with the acquired CT scan dataset and stores the information in a "session". The session manager collects all data and documents for the whole process of a particular Virtopsy procedure.

The VIRTOBOT system has been in routine use since 2014. Use of the system helps in minimizing errors and leads to better and more reproducible results. Compared to the manual procedure, overall scanning times can be reduced significantly [8].

Conclusions

Among medical robotic systems, mechatronic assistants for percutaneous procedures show a high potential. Such robots help guide the needle into the most promising region of the lesion with a very high accuracy. As a consequence, more efficacious characterization and treatment, particularly for lesions that are difficult to target, can be anticipated.

Acknowledgements

The author thanks the development team at ACMIT (Martin Fürst, Martin Kornfeld, Wolfgang Ptacek, Nikolaus Hocke, Peter Mainer) as well as our cooperation partners Johns Hopkins University (Peter Kazanzides), Queens University (Gabor Fichtinger), Krankenhaus Karlsruhe (Uwe Spetzger), IRM Zurich (Michael Thali), and iSYS Medizintechnik GmbH (Michael Vogele).

References

[1] Davies B: A Review of Robotics in Surgery, Proceedings of Inst. Mechanical Engineering, 214, 129-140 (2000)

[2] Howe RD, Matsuaka Y: Robotics for Surgery, Annu. Rev. Biomed. Eng., 1, 211-240 (1999)

[3] Moore JE, Zouridakis G: Biomedical Technology and Devices Handbook, The Mechanical Engineering Handbook Series, 32-1-32-24, CRC Press (2004)

[4] Davies B: A Discussion of Safety Issues for Medical Robotics, Journal Computer-Integrated Surgery, Taylor. R. H. et al. (Eds) 287-296, MIT Press, Cambridge (1996)

[5] Elder MC, Knight JC: Specifying User Interfaces for Safety-Critical Medical Systems, Journal Medical Robotics and Computer Assisted Surgery, 148-155, Wiley (1995)

[6] Kettenbach J, Kronreif G, Figl M, Fürst M, Birkfellner W, Hanel R, Ptacek W, Bergmann H: Robot-assisted Biopsy Using CT-Guidance: Initial

Results from In-Vitro Tests, Journal Investigative Radiology, Volume 40, Issue 4, 219-228, Lippincott Williams & Wilkins (2005)

[7] Kronreif G, Fürst M, Ptacek W, Kornfeld M, Kettenbach J: Robotic Platform B Rob II: In-vitro Tests and Results, In: 4th International Conference on Computer Aided Surgery around the Head, Innsbruck (2007)

[8] Ebert L, Ptacek W, Naether S, Fürst M, Ross S, Buck U, Weber S, Thali M: Virtobot -- a Multi-Functional Robotic System for 3D Surface Scanning and Automatic Post Mortem Biopsy, Journal Medical Robotics and Computer Assisted Surgery, 6(1), Wiley, 18-27 (2010)

[9] Hoeckelmann M, Rudas IJ, Fiorini P, Kirchner K, Haidegger T: Current Capabilities and Development Potential in Surgical Robotics, International Journal of Advanced Robotic Systems, Vol. 12:61, ISSN 1729-8806 (2015)

[10] ROS – Robot Operating System: http://www.ros.org/ (last visit: Dec 2015)

[11] Open Robot Control Software – OROCOS: http://www.orocos.org/ (last visit: Dec 2015)

[12] 3DSLICER: https://www.slicer.org/ (last visit: Dec 2015)

[13] Song DY, Burdette EC, Fiene J, Armour E, Kronreif G, Deguet A, Iordachita I, Fichtinger G, Kazanzides P: Robotic Needle Guide for Prostate Brachytherapy: Clinical Testing of Feasibility and Performance, Brachytherapy, 10(1):57-63 (2011)

[14] Bohr C, Milojcic R, Kronreif G, Fuerst M, Stegman A, Vogele M, Wurm J, Iro H: Feasibility Study for a New Robot Design, B-Rob II, in Lateral Skull Base and Transsphenoidal Surgery, Proc. 5th World Congress for Endoscopic Surgery of the Brain, Skull Base & Spine (2012)

[15] Kronreif G, Ptacek W, Kornfeld M, Fürst M: Evaluation of Robot Assistance in Neuro-Surgical Applications, Journal of Robotic Surgery, Volume 6, Issue 1, pp. 33-39 (2012)

[16] Kronreif G, Ptacek W, Fürst M, Vogele M: iSYS 1 Robot System for Neurosurgical Applications, Abstract book, DGNC 2015 (2015)

From the Smart Hands to Tele-Operations

Imre J. Rudas, József Gáti, Anikó Szakál, Krisztina Némethy

Óbuda University, Bécsi út 96/b, H-1034 Budapest, Hungary
{rudas,gati,szakal}@uni-obuda.hu, nemethy.krisztina@rh.uni-obuda.hu

Abstract: Professor Dr. Antal K. Bejczy, the Hungarian-born scientist innovator worked for the NASA Jet Propulsion Laboratory (JPL) in California from 1969, where he developed, with his colleagues, the first dynamic model of robotic manipulators based on the Lagrangian formulation. In the field of teleoperation, he developed, and sponsored through NASA Telerobotics program, a number of prototypes aimed at achieving full Telepresence in space and undersea. He pioneered robot dynamics development and published one of the first papers in this field, describing "smart hands" with multi-fingered, effector equipped sensors. One of his research papers named "The Robot Arm Dynamics and Control" was published by JPL in 1974. This was one of the most important papers in the history of robotics.

Keywords: Antal K. Bejczy; smart hands;exoskeleton; Antal Bejczy Center for Intelligent Robotics; dynamic model of robotic manipulators; teleoperation

1 Prof. Dr. Antal K. Bejczy at NASA JPL

Antal K. Bejczy was born in Hungary in 1930. After high school graduation with "excellence" in Kalocsa at Jesuits, he was working for an electrical motor factory for some years. Due to political reasons he could only start his studies in 1953, at the Technical University of Budapest, in the field of electrical engineering. He left Hungary with one of his friends after the revolution in 1956. He went to Norway, where he completed his University studies, in the field of applied physics, at the University of Oslo in Norway in 1963. His thesis work concerned the geometric problems of atomic reactors. After that, he taught statistical thermodynamics at the University, until 1966. He met in Norway, the world famous Hungarian born scientists Tódor Kármán, who invited him to the JPL NASA Laboratory in the USA. (The NASA Jet Propulsion Laboratory was established by Tódor Kármán). Because of his meeting Tódor Kármán, Bejczy became interested in the new field of study. He applied at four American Universities and he received a NATO/Fulbright scholarship from the California Institute of Technology (Caltech) in Pasadena, as a Senior Research Fellow in 1966.

RESEARCH FELLOWS

Nadine Alleaume,* D.Sc. ..... *Applied Science*
Johann Arbocz, Ph.D. ..... *Aeronautics*
Antal K. Bejczy, Cand. Real. ..... *Engineering*
Halka Bilinski,* Ph.D. ..... *Environmental Health Engineering*
George W. Bluman,* Ph.D. ..... *Applied Mathematics*

Figure 1

Information for students: California Institute of Technology 1968

He worked on optimal control and nonlinear filtering problems. He joined the Jet Propulsion Laboratory (JPL) as a Member of the Technical Staff in 1969. He became a Senior Research Scientist (a professor rank at JPL-Caltech) in 1985 and the Technical Manager of the robotics program at JPL.

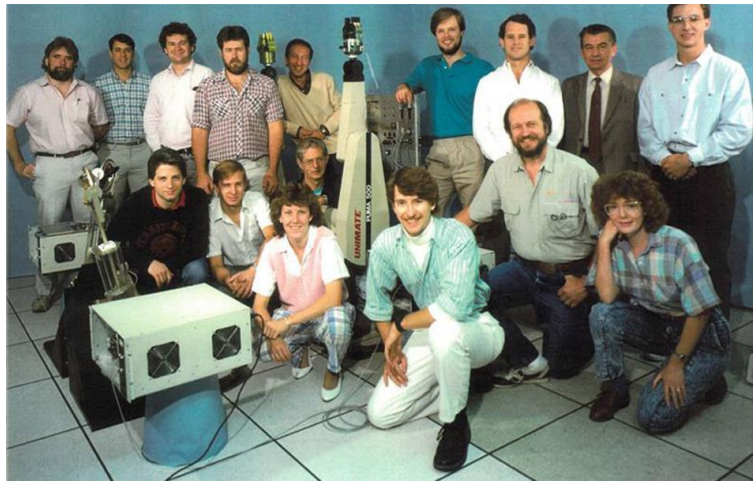


Figure 2

The Advanced Teleoperation Group with Antal K. Bejczy at JPL in 1986

He was also an Affiliate (non-resident) Professor in System Science and Mathematics at Washington University in St. Louis, MO, from 1983, with the duty of establishing and maintaining the Graduate Studies in Robotics. There he supervised – together with local professors – eleven successful Ph.D. students in their thesis work, in the field of robotics. He served as a member and chair of various governmental or private technical committees for creating and evaluating robotic and automation proposals or establishing state-of-the-art in robotics and automation.

He was the successful Principal Investigator of the robot arm, force-moment sensor, flight experiment for the Space Shuttle in 1994.

1.1 Publications of Antal K Bejczy

He initiated work on modeling and sensing based intelligence, in robot control, published more than 160 technical papers, 11 book chapters on topic sensing, dynamic modeling, control, Telepresence, virtual environments, human-machine interaction in robotics and on the application of robots in space research.

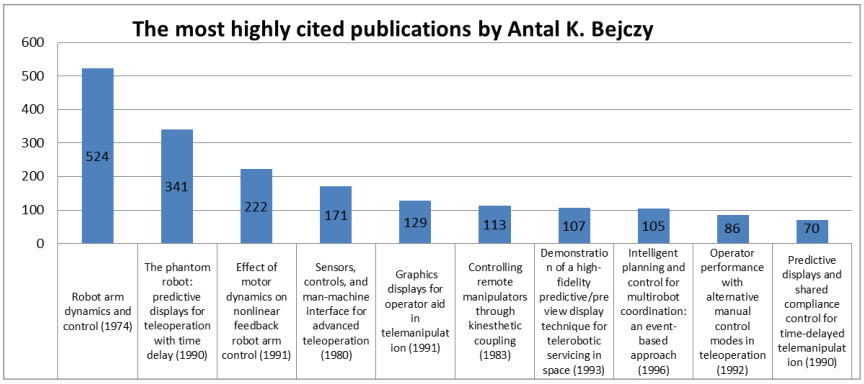


Figure 3

Most highly cited publications, Source: Google Scholar 2015

The book titled Robot Arm Dynamics and Control (Figure 3) by Antal K. Bejczy (1974) is the sixth highest accessed publication according to the Google Scholar in the field of Robot Arm Research. The two main topics of his book relate to the dynamic aspects and central problems of the 6 degrees of freedom, JPL Robot Research Project manipulator:

- Variations in total inertia and gravidity loads as the joint outputs
- The relative importance of gravity and acceleration-generated reaction torques of forces versus inertia torques of forces. Bejczy [1]

The core idea of the control is that the inputs should be computed from the state. The main result is the construction of a set of greatly simplified state equations which describe total inertia and gravity load variations at the output of the six joints with an average error of less than five percent. The dynamic model of the six degrees of freedom JPL Robot Research Project manipulator can be obtained from the known laws of the Newtonian mechanics and physical measurements. The algorithm is based on a special representation of link coordinate frames in jointed mechanism and formalism of the Langrian mechanics. The dynamic model is described by a set of six coupled nonlinear differential equations which contains a large amount of torque or force terms classified into four groups. The above general algorithm (Figure4) describe: the manipulator equations of motion is given by following expression for the torque of force F_i acting at joint "i" Lewis at all [11]

$$\sum_{j=1}^{n}\left\{\sum_{k=1}^{j}\left[\mathrm{Trace}\left(U_{jk}J_jU_{ji}^T\right)\ddot{q}_k\right]+\sum_{k=1}^{j}\sum_{p=1}^{j}\left[\mathrm{Trace}\left(U_{jkp}J_jU_{ji}^T\right)\dot{q}_k\dot{q}_p\right]-m_jGU_{ji}\bar{r}_j\right\}=F_i\ ,\ i=1,\ 2,\ \ldots,\ n \qquad (1)$$

where superscript T denotes the transpose of the matrix U_{ji}, and

Figure 4

Lewis, R.A., Bejczy A. K., RRP Manipulator Conventions, Coordinate Systems, and Trajectory Consideration. 1972

He co-edited a book on Parallel Computation Systems in Robotics [5]. He was also on the editorial board of four robotics and automation journals like the International Journal of Robotics and Automation.

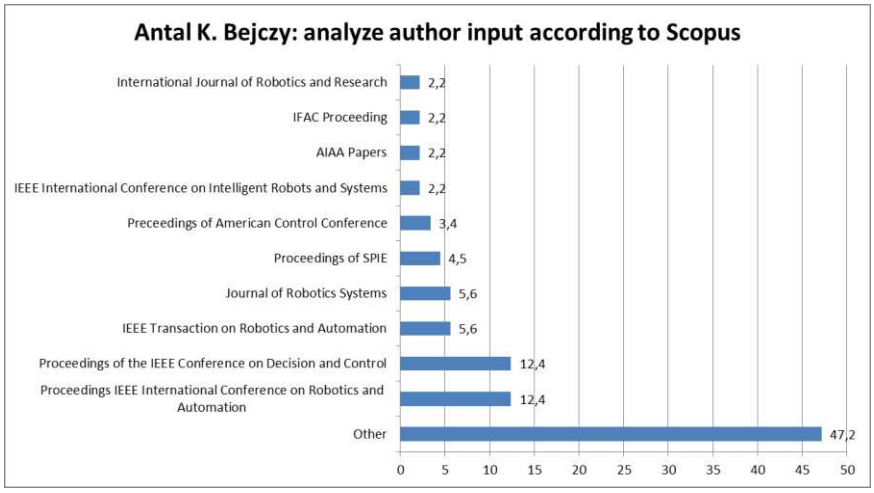


Figure 5

Author input: Source: Scopus 2015

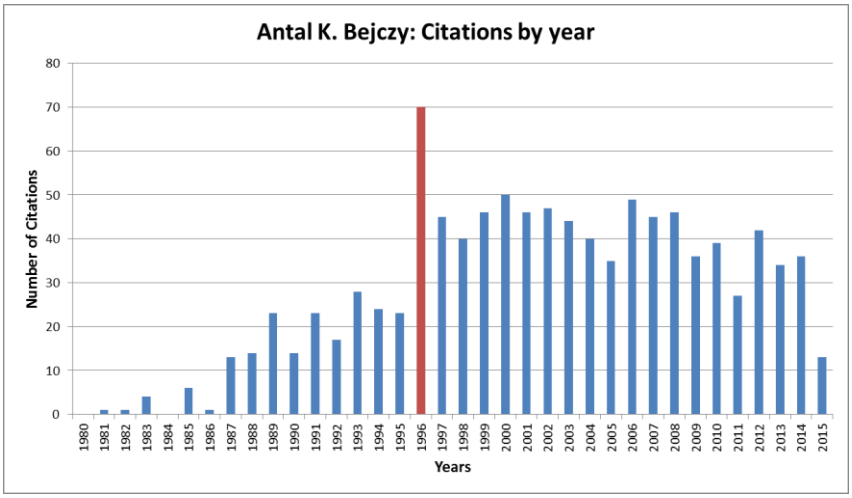


Figure 6
Citations by year, Source: Scopus 2015

When the Mars Pathfinder spacecraft inaugurated a return to the Red Planet in 1996, the citations of Antal K. Bejczy's papers were significantly higher level like before.

1.2 Pathfinder

"*JPL initiated the Pathfinder project, a shuttle-based acquisition and tracking experiment, for the Strategic Defense Initiative Organization. The project's two major goals are to demonstrate technologies associated with tracking a missile-booster plume and acquiring data on the plume from which its signature and relationship to the booster can be derived. Progress was made in designing the mission, conducting the preliminary design, and starting to build test and protoflight hardware. The Pathfinder project was to fly as part of a shuttle mission in late 1987, but late in 1986 the Pathfinder project was told to adapt the existing design of a free-flying satellite for launch on an expendable vehicle. The launch date was expected to slip to 1990.*" [2] (JPL report 1986, NASA JPL Pasadena)

"*The Mars Global Surveyor and Mars Pathfinder spacecraft inaugurated a return to the Red Planet in 1996 with their launches atop McDonnell Douglas Delta II. rockets-two decades after the landmark Viking missions. Surveyor was launched 10-month cruise to the planet, where it will map the Martian globe in polar orbit for at least two years. Pathfinder, the first in NASA's Discovery series of low-cost, highly focused missions, was launched December 4, with the goal of placing a lander and the small (11.S-kilogram) Sojourner rover on the Martian surface on July 4,1997.*"[3] (JPL report, NASA JPL Pasadena 1996)

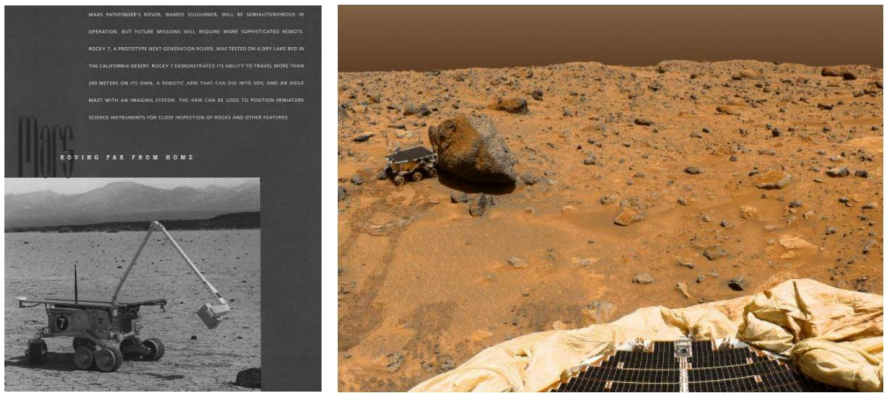


Figure 7, 8
Pathfinder: JPL report 1986, 1997, NASA JPL CA, Pasadena [2, 4]

The remote control methods of the Pathfinder were prepared and developed by JPL researchers. During the project, they needed to confirm the "remote controlled sensing", the operation of the robot arms equipped with sensors. *"On a robotic arm there is no hand. Instead of a hand there are spectroscopic instruments (alpha particle X-ray spectrometer for determining the components of the soil), therefore, the movement of these had to be designed so that they did not collide with the material being investigated. It was only data for spectral analysis that was sent and received…"*—said professor Bejczy in 2012.

2 President of the IEEE Council of Robotics and Automation and IEEE Fellow

Dr. Bejczy was a frequent organizer of tutorials, workshops and sessions on robotics at the conferences for the Institution of Electrical and Electronics Engineers (IEEE). He gave lectures and seminars on robotics topics at twenty four universities in nine countries.

He was Chairman of the IEEE Control System Society Technical Committee on Robotics and Automation from 1983 to 1985, General Chairman of the IEEE International Conference on Robotics and Automation in San Francisco, California in 1986, and President of the IEEE Council of Robotics and Automation in 1987 when he helped to transform the Council of the current IEEE Robotics and Automation Society (RAS), with about seven thousand members.

He served as a member of the Governing Board of the IEEE RAS during 1989-1991, and reelected for 1994-1999. He was the General Chair of the 8. International Conference on Advanced Robotics (ICAR) in Monterey. CA in 1997 and the

General Co-Chair of the 10th ICAR in Budapest, Hungary in 2001, Program Co-Chair for the 11th ICAR in Coimbra, Portugal, in 2003.

Figure 9
IEEE RAS Past-President Dinner at his last ICRA in St. Paul (2012)

He received the IEEE Fellow honor, in 1987, for "*Contributions to the theory and applications of robotics*", and the Jean Vertut Award of the Society of the Manufacturing Engineering&Robotics International (SME&RI) for the "*Remote application of robotic technology*," in 1991 the NASA Exceptional Service Medal for "*Sustained contributions to innovative technology and NASA leadership in advanced teleoperation,*" in 1994 the NASA Flight Experiment Achievement Certificate for "O*utstanding contribution on the Dexterous End Effector Flight Demonstration*," in 2000 the "*IEEE Third Millennium Medal for outstanding achievements and contributions,*" in 2004 the IEEE RAS Pioneer Award "*For seminal technical contribution to robotics and teleoperation and pioneer research in space robotics and human-robot interfaces,*" in 2007 the IEEE RAS Distinguished Service Award "*For outstanding contribution to Robotics and Automation Society professional activities and his leadership as the IEEE Robotics and Automation Council Chair who fostered the transition from a Council to a Society.*"

In 2007 he received the American Institute of Aeronautics and Astronautics (AIAA) Space Automation and Robotics Award "*For establishing the foundation of robotics for space applications and pioneering the development of key enabling techniques to overcome the special challenges of communication delay, operation feedback, and unstructured environment.*" That award was augmented in 2007 by the International Technology Institute's (ITI) "*Diploma for Hall of Fame,*" together with the ITI highest award, "*The Willard F. Rockwell, Jr. Medal.*"

He was also the recipient of the 2009 IEEE Robotics and Automation "*Technical Field Award*" (TFA) "*For leadership and sustained contribution to a broad set of innovative robotic and automation techniques applicable to space research and on Earth*."

3 Prof. Dr. Antal K. Bejczy: 34 NASA Innovation Awards and 7 U.S. Patents.

He received forty-three NASA innovation awards and holds seven U.S. patents. He pioneered the development of innovative robot components such as smart hands with smart sensors and a novel telerobotic system using a general-purpose force-reflecting hand controller for remote robot arm control, which contributed to the birth of haptic research[7]. The results of Antal K. Bejczy's research projects were collected in seven U.S, patents: (source: NASA Technical Reports (NTRS) [7]

1. **Terminal Guidance Sensor System**: No. 4260187 (1979)

"*Abstract: A system is described for guiding a claw to the proper distance and into the proper orientation in yaw and pitch, to engage a grappling fixture. The system includes four proximity sensors on the claw, that are arranged at the corners of an imaginary square, which sense the distance to the top surface of the grappling fixture. If a pair of sensors at opposite corners of the square sense a different distance to the top surface of the grappling fixture, then it is known that the claw is rotated about a corresponding axis with respect to the plane of the grappling fixture."* (Inventors: Robert A. Administrator of the National Aeronautics and Space Administration, with respect to an invention of Frosch, Antal K. Bejczy.

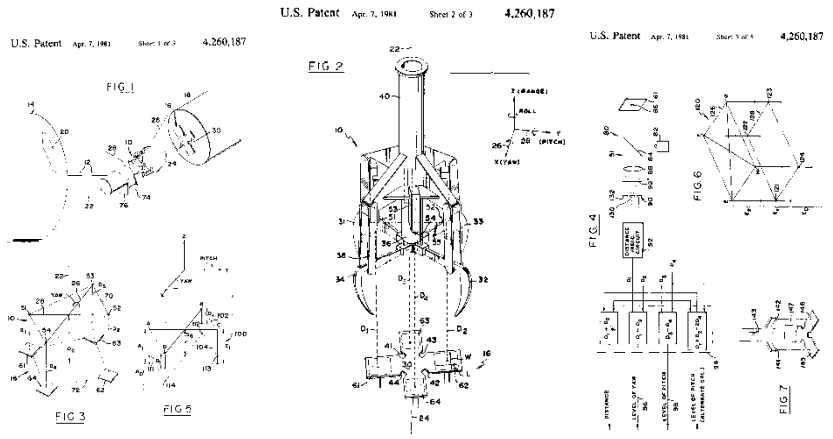


Figure 10
Terminal Guidance Sensor System: No. 4260187 (1979) US Patents

2. **Optical fiber tactile sensor:** No. 4405197 (1983)

"Abstract: A tactile sensor comprises an array of cells, which are covered by an elastic membrane, having an exposed surface which is adapted to come in contact with an object. Light is conducted to each cell from a light source by an optical fiber which terminates at the cell. Reflected light from the cell is conducted by an

optical fiber to a light processor, which senses changes in the light received thereby from an ambient level whenever an object comes in contact with the membrane surface above the cell."(Inventor: Antal K. Bejczy)

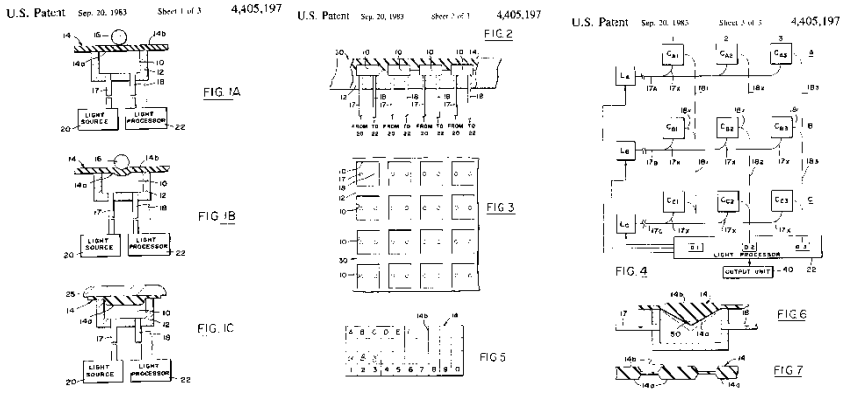


Figure 11
Optical fiber tactile sensor: No. 4405197 (1983) US Patents

3. **Linear Array Optical Edge Sensor***:* No. 4694153 (1987)

"Abstract: A series of independent parallel pairs of light emitting and detecting diodes for a linear pixel array, which is laterally positioned over an edge-like discontinuity in a work piece to be scanned, is disclosed. These independent pairs of light emitters and detectors sense along intersecting pairs of separate optical axes. A discontinuity, such as an edge of the sensed work piece, reflects a detectable difference in the amount of light from that discontinuity in comparison to the amount of light that is reflected on either side of the discontinuity. A sequentially synchronized clamping and sampling circuit detects that difference as an electrical signal which is recovered by circuitry that exhibits an improved signal-to-noise capability for the system." (Inventors: Antal K. Bejczy, Howard C. Primus)

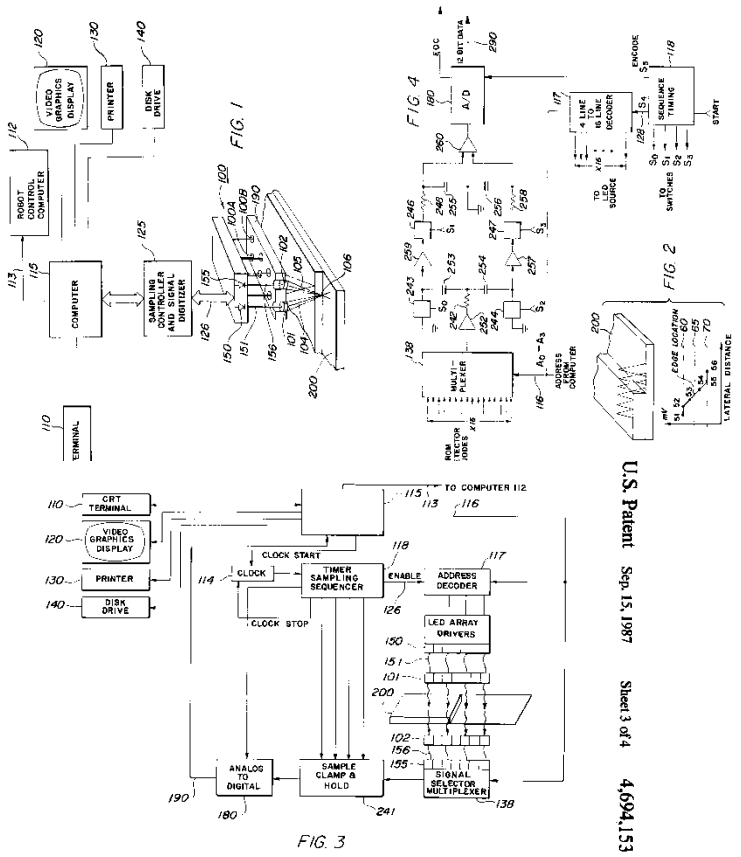


Figure 12
Linear Array Optical Edge Sensor: No. 4694153 (1987) US Patents

4. **Grasp Force Sensor for Robotic Hands***:* No. 4819978 (1989)

"Abstract: A grasp force sensor for robotic hands is disclosed. A flexible block is located in the base of each claw through which the grasp force is exerted. The block yields minute parallelogram deflection when the claws are subjected to grasping forces. A parallelogram deflection closely resembles pure translational deflection, whereby the claws remain in substantial alignment with each other during grasping. Strain gauge transducer supply signals which provide precise knowledge of and control over grasp forces."(Inventors: Victor D. Scheinman, Antal K. Bejczy, Howard C. Primus)

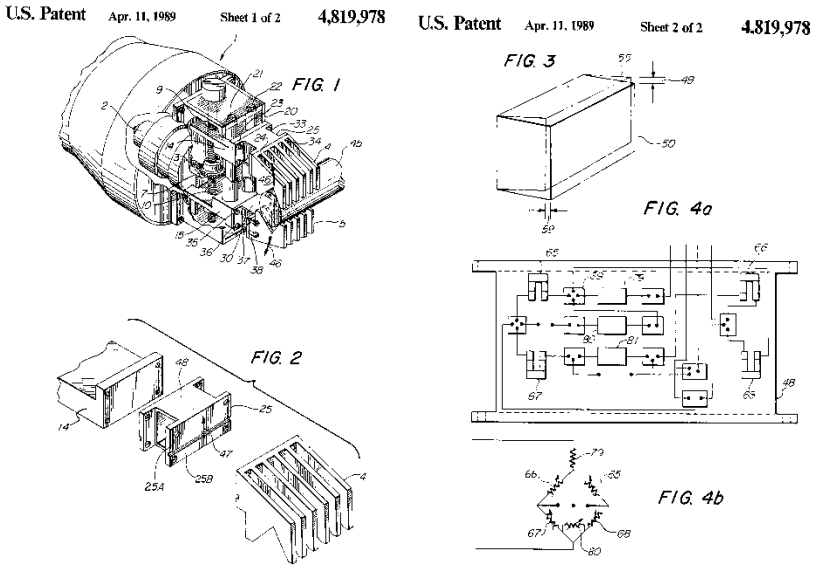


Figure 13
Grasp Force Sensor for Robotic Hands: No. 4819978 (1989) U.S. Patents

The research of Antal Bejczy was focused around a basic concept, that "*without the elaboration of the robot–human interaction there is no such thing as successful robotics*". By Amir Fijany, Sherman Oaks and Antal Bejczy [8] created two patents based on the results of the experiments:

5. **Special purpose parallel computer architecture for real-time control and simulation in robotic applications:** No 5218709 (1993)

"Abstract: A Real-time Robotic Controller and Simulator (RRCS) with an MIMD-SIMD parallel architecture for interfacing with an external host computer provides a high degree of parallelism in computation for robotic control and simulation. A host processor receives instructions from, and transmits answers to, the external host computer. A plurality of SIMD microprocessors, each SIMD processor being an SIMD parallel processor, is capable of exploiting fine-grain parallelism and is able to operate asynchronously to form an MIMD architecture. Each SIMD processor comprises an SIMD architecture capable of performing two matrix-vector operations in parallel while fully exploiting parallelism in each operation. A system bus connects the host processor to the plurality of SIMD microprocessors and a common clock provides a continuous sequence of clock pulses." (Inventors: Amir Fijany, Antal K. Bejczy.)

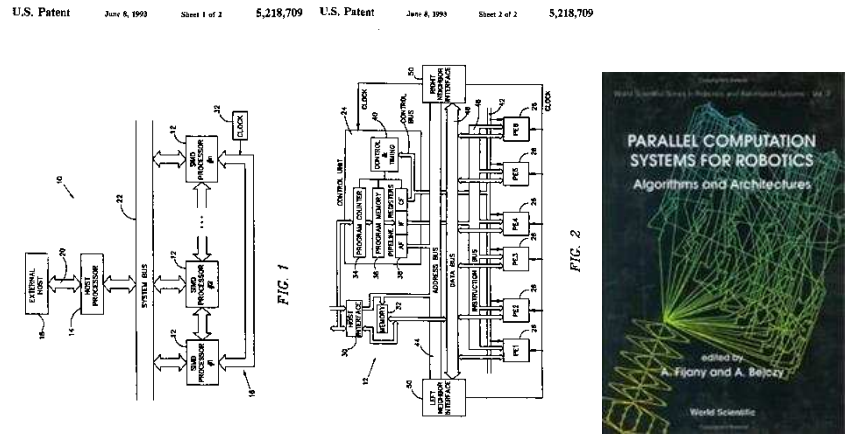


Figure 14, 15

14:Special purpose parallel computer architecture for real-time control and simulation in robotic applications: No 5218709 (1993) US Patents

15: Parallel Computation Systems for Robotics Algorithms and Architectures Edited by: A Fijany (JPL/Caltech), A Bejczy (JPL/Caltech) ISBN: 978-981-02-0663-5 [8]

In 1993, NASA scientists at JPL, were using a new remote control technology and during the test, a new graphically oriented program, to remotely control a robot arm at the NASA' Goddard Space Flight Center. The experiment's technical manager was Dr. Bejczy at JPL. The technology was licensed by a private firm and the main target of the free-flying robots was the service of the existing orbiting satellites, for many possible uses on Earth, like the clean-up of toxic or nuclear waste sites or for special medical procedures. The main target of the robotic arm was to insert a screwdriver into an 18 inch deep hole, to reach the latching mechanism while in Space, but the operator controlled the robotic arm from Earth. A special software, that allows the remote operator to superimpose high-fidelity computer graphics models of the robot arm and satellite module, on the monitor in live-scene and make visible, but hidden critical motion events.

"The operator is able to generate, predict or preview motions without commanding the actual hardware and can see the consequences in real time"-said Professor Bejczy in 1993. One of the most important projects of Professor Bejczy, was to find a solution for time delays and to use the robotic arm in real time.

6. **Highly parallel reconfigurable computer architecture for robotic computation having plural processor cells, each having right and left ensembles of plural processors:** No**:**5361367 (1994)

"Abstract: In a computer having a large number of single-instruction multiple data (SIMD) processors, each of the SIMD processors has two sets of three individual processor elements controlled by a master control unit and interconnected among a plurality of register file units where data is stored. The register files input and output data in synchronism with a minor cycle clock under the control of two slave control units controlling the register file units connected to respective ones of the two sets of processor elements. Depending upon which ones of the register file units are enabled to store or transmit data during a particular minor clock cycle, the processor elements within an SIMD processor are connected with rings or in pipeline arrays, and may exchange data with the internal bus or with neighboring SIMD processors through interface units controlled by respective ones of the two slave control units. (Inventors: Amir Fijany, Antal K. Bejczy.)

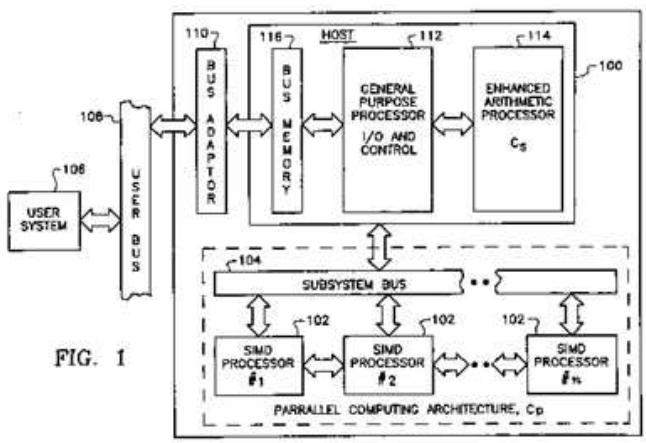


FIG. 1

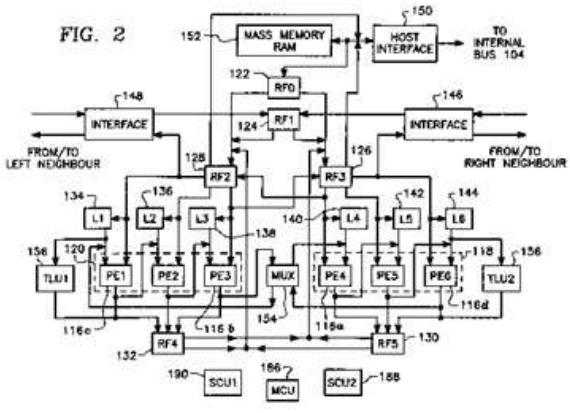


FIG. 2

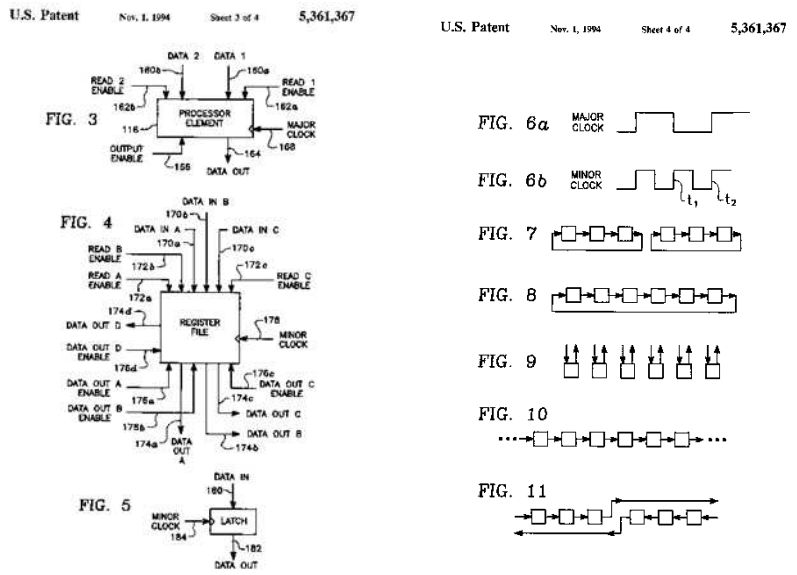


Figure 12

Highly parallel reconfigurable computer architecture for robotic computation having plural processor cells, each having right and left ensembles of plural processors: No:5361367 (1994) US Patents

Many new technologies for space exploration were later, very important and significant applications within the field of human medical applications. Antal Bejczy, the senior research scientist and lead engineer was also working on several projects at JPL like the robotic stepper device. The scientists were working together with therapists to support the rehabilitation of patients with lower stepping capacity, for example, after a stroke, neuromotor disorders or spinal injuries. The highly sensitive sensors were able to collect twenty four different data about the patients movement on the computer screen. Also an important result of this method that his system had been replaced the work of four therapists.

7. **Method, apparatus and system for automation of body weight support training (BWST) of biped locomotion over a treadmill using a programmable stepper device (PSD) operating like an exoskeleton drive system from a fixed base.** No: 6666831 (2003)

"Abstract: A robotic exoskeleton and a control system for driving the robotic exoskeleton, including a method for making and using the robotic exoskeleton and its control system. The robotic exoskeleton has sensors embedded in it which provide feedback to the control system. Feedback is used to the motion of the legs themselves, as they deviate from a normal gait, to provide corrective pressure and guidance. The position versus time is sensed and compared to a normal gait profile. Various normal profiles are obtained based on studies of the population for age, weight, height and other variables".(Inventors: V. Reggie Edgerton, M. Kathleen Day, Susan Harkema, Antal K. Bejczy, James R. Weiss)

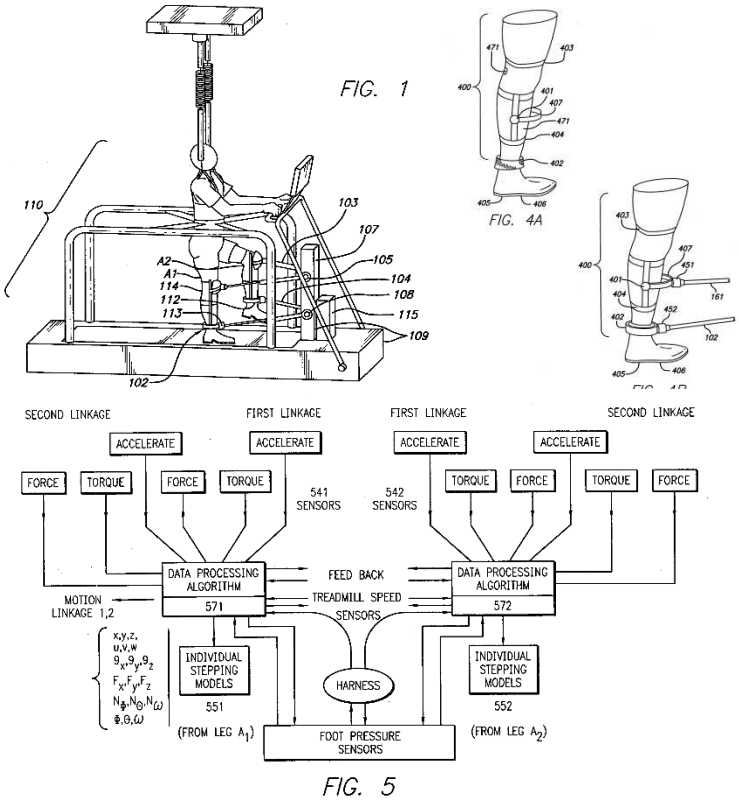


Figure 12

Method, apparatus and system for automation of body weight support training (BWST) of biped locomotion over a treadmill using a programmable stepper device (PSD) operating like an exoskeleton drive system from a fixed base. No: 6666831 (2003) US Patents

Already in 1980 was published by Antal K. Bejczy and J. K. Salisbury the basic mechanism of the force reflecting hand controller where the band controller able to measure three positions (x, y, z) and also three orientations like the pitch, yaw and roll. Because of the forces and torques in three positional and three rotational axes of the hand controller, the operator able to feel the task he is controlling. Bejczy at al 1980 [12]

3.1 Anthropomorphic Telemanipulation System

The theory described the prototype anthropomorphic kinesthetic Telepresence system. Antal K. Bejczy and his colleagues at the JPL laboratory (B. M. Jau and M. A. Lewis) presented the results in the book titled: Theory and Practice of Robots and Manipulators. *"The newly developed master-glove is integrated with*

our previously developed non-anthropomorphic six degree-of freedom universal force reflecting hand controller. The system is controlled by high performance distributed controller. Control electronics and computing architecture were custom developed for this telemanipulaton system. The system is currently being evaluated, focusing on tool handling and astronaut equivalent task executions. The evaluation revealed the system's potential for tool handling but it also became evident that hand tool manipulations and space operations require a dual arm robot. " [9]

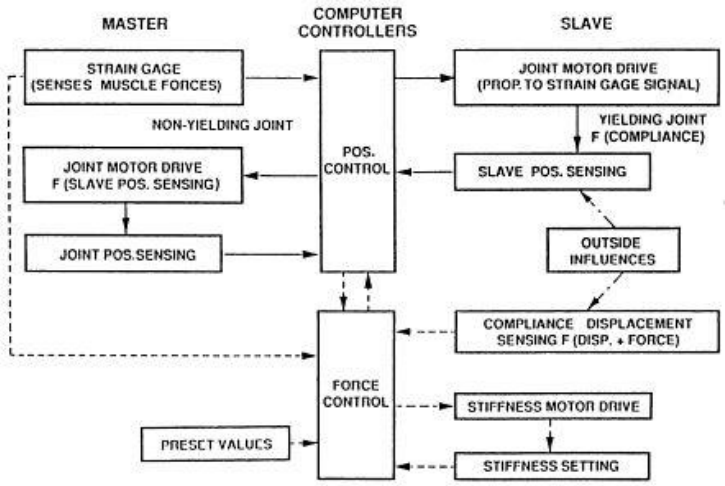


Figure 13
Anthropomorphic Telemanipulation System in Terminus Control Mode [9]

Conclusions

During his life and career, Prof. Dr. Antal K. Bejczy was open to a wide-range of knowledge and science, from Theoretical Mathematics to Practical Physics. Wherever he worked, he received many jobs and always performed in an excellent manor. Mainly, his innovative thinking and personality, raised him to be among the best scientists.

The oeuvre of Prof. Dr. Antal K. Bejczy (1930-2015) is a prime example for all Researchers in Hungary and Overseas. The Antal Bejczy Center for Intelligent Robotics, opened by the Óbuda University, in 2013 in Budapest. The Antal Bejczy Award (2015) and the Antal Bejczy Museum was also established by Obuda University, Hungary - University Research and Innovation Center. The Opening Ceremony will be held on the 18th of January, in 2016.

Acknowledgement

This work was supported by Óbuda University, University Research and Innovation Center, Antal Bejczy Center for Intelligent Robotics

Figure 14
Prof. Dr. Antal K. Bejczy, honorary doctor of Óbuda University (2012)

References

[1] A. K Bejczy Robot Arm Dynamics and Control (1974) NASA JPL Technical Memorandum 33-669

[2] NASA JPL 1986 Annual Report, California Institute of Technology Pasadena, California

[3] NASA JPL 1996 Annual Report, California Institute of Technology Pasadena, California

[4] NASA JPL 1997 Annual Report, California Institute of Technology Pasadena, California

[5] Parallel Computation Systems for Robotics Algorithms and Architectures Edited by: A Fijany (JPL/Caltech), A Bejczy (JPL/Caltech) ISBN: 978-981-02-0663-5

[6] T. Haidegger, The Rich Heritage of Antal Bejczy, in the proceedings of INES 2015 • IEEE 19th International Conference on Intelligent Engineering Systems • September 3-5, 2015, Bratislava, Slovakia, Plenary talk

[7] NASA Technical Reports (NTRS) Antal K. Bejczy, US Patents

[8] Parallel Computation Systems for Robotics Algorithms and Architectures Edited by: A Fijany (JPL/Caltech), A Bejczy (JPL/Caltech) ISBN: 978-981-02-0663-5

[9] B. M. Jau, M. A. Lewis, A. K. Bejczy, Theory and Practice of Robots and Manipulators, Volume 361 of the series International Centre for Mechanical Sciences, pp. 101-111, Anthropomorphic Telemanipulation System in Terminus Control Model, ISBN 978-3-7091-2698-1 (1995)

[10] The IEEE Robotics and Automation Society History, IEEE Robotics and Automation Award: about Antal K. Bejczy

[11] Lewis, R. A., Bejczy A. K., RRP Manipulator Conventions, Coordinate Systems, and Trajectory Consideration. JPL Technical Memo 343-174, 1972

[12] Bejczy, A. K. and Salisbury, J. K., Jr., “Kinesthetic Coupling Between Operator and Remote Manipulator,” .Proc. of ASME Int’l Computer Technology Conference, Vol. 1, San Francisco, CA, Augus 12-15, 1980, pp. 197-2 1 1

From the First Chess-Automaton to the Mars Pathfinder

George Kovács

Computer and Automation Research Institute, Hungarian Academy of Sciences
Kende u. 13-17, H-1111 Budapest, Hungary; kovacs.gyorgy@sztaki.mta.hu

Alexander Petunin*, Jevgenij Ivanko**

*,**Ural Federal University, ul. Mira 19, 620002 Ekaterinburg, Russia
a.a.petunin@urfu.ru; ivanko@uran.ru
**Institute of Mathematics and Mechanics, Ural Branch, Russian Academy of Sciences, 16, S. Kovalevskoi, 620990 Ekaterinburg, Russia

Nafissa Yusupova

Ufa State Aviation Technical University, 12 K. Marx Str., 450000 Ufa, Russia
yussupova@ugatu.ac.ru

Abstract: This paper aims to highlight the relationship of Artificial Intelligence, the first Chessautomaton (The Turk), Computer Chess (Deep Blue), Mars Mission (Pathfinder Sojourner), Intelligent Robotics and Industrial Robots Biographical and technical data is presented in order to evaluate and laudate the extraordinary achievements of extreme talents, starting with two Hungarian world class innovators: Farkas Kempelen and Antal Bejczy. This paper gives an overview of their lives and contributions, pointing out some interesting connections. A novel evaluation and classification method of robots is suggested.

Keywords: Mars Rover; Pathfinder; Chess automaton

1 Introduction

There are many Hungarians, who had a major contribution in the most important inventions and scientific milestones of mankind. John von Neumann, József Galamb, Ányos Jedlik, Tódor Kármán, Leó Szilárd, Miksa Déri are only a few of the many famous Hungarian scientists and engineers, who played a major role in

shaping our world's technology over the past two centuries. Robotics is a relatively narrow and young field of engineering and computer science. Still, it is inevitable to find the names of two Hungarian scientists in the fundamental work in this field: Antal Bejczy and Farkas Kempelen.

Farkas Kempelen was the first in the world, who constructed and demonstrated a chess playing machine that was human made, supposedly independent and automatic. However, while he claimed that he designed and made an intelligent robot, several decades passed until it was revealed that it was neither intelligent, nor a robot. It only acted like one and looked like one. The "robot" was called The Turk, a chess automaton, which won most of its games for about 85 years, between 1770 and 1854. It was only 16 years after the death of its creator, in 1820, when the truth about the machine: it was an illusion as there was no automaton, no thinking machine, but a small human in the box of The Turk. The human operator played chess and moved the figures and other parts of The Turk using magnets, mirrors and mechanical structures. Looking back, we can all agree that it wasn't the robot but Kempelen, who had the intelligence, and who made the world believe the unbelievable. Although today we know that cheating is forbidden, whether we discuss sports, exams, games etc., the concept of cheating may have been differently accepted 250 years ago among certain given circumstances.

Prof. Antal K. Bejczy passed away recently, after a 35-year-long career in the American space industry. He was best known for being one of the major contributors to the Sojourner, which was the first rover to land on the Mars, conducting experiments for 85 days in 1997 as part of the Pathfinder mission [1]. As a leader of the Advanced Teleoperation Laboratory at NASA Jet Propulsion Laboratory (JPL), his team provided the remote operation capabilities and control of the robot arm.

Robotics and automation have gone under a lot of development between the creation of the first chess "automaton" and the landing of the first rover on Mars. Definitions, properties and abilities of robots have developed over time. The goal of this paper is to evaluate, how these two machines fit in the concept of robotics that mankind has developed over the centuries. The steps of development will be illustrated on 5 more distinct, but corresponding robotic objects, which will be also implemented in the evaluation.

Antal (Tony) Bejczy was always interested in such discussions. During his career, he aimed to research the most modern problems, at the same time, he was always supporting new ideas. He had a good sense of humor to comprehend funny research ideas as well, such as the comparisons and evaluations we present in this paper. Honoring his memory in this issue, we believe that he would support our efforts and would await the results of this research with curiosity and sympathy.

2 Comparison Methods

In our, research, we focus on 4 mechanical, intelligent and robotic systems:

- The Turk (TAI)
- Pathfinder (PF)
- Deep Blue (DB)
- Industrial robots (IR)

In addition, three virtual robots well be investigated, derived from the first 3 cases, while an industrial welding robot will be taken as a reference. In the following listing, some of the most relevant properties of the investigated robots are listed, primary from the Artificial Intelligence (AI) point of view.

1) **TAI**: "The Turk" was considered a thinking, intelligent machine in 1770 [2]. Considering it as a machine, it had restricted communication capabilities with its environment, had knowledge of playing chess it a high level, and finally, it was equipped with tools and was capable of moving its head and hands, in order to grasp a chess figure with its fingers and to place it where it had to be placed. TAI was also capable of producing voice, even spoke words. Taking aside the fact that it was an illusion, a kind of cheating, it worked and won for about 85 years, with the help of a hidden human operator.
2) **TM**: Let us consider the mechanism of The Turk separately. It was a mechanical construction, without any intelligence, or actively actuated mechanism, thus it was not capable of moving any of its parts by itself. In its form, it can be referred to as a simple manipulator. It is important to note that as of our knowledge, it was never on the stage "as is", since the hidden person, the human operator was necessary to control its actions.
3) **PF**: The Pathfinder was a robotic spacecraft, carrying the first Mars rover, Sojourner to the Red Planet, in which Tony Bejczy's team played a major role. It was capable of moving to any direction, avoided obstacles, took pictures, exchanged information with the control room, picked up and analyzed Martian terrain, etc.
4) **PFx**: Let us consider a virtual robot, which would be similar to the Pathfinder, as it could have been extended and equipped with more sensors and actuators, capable of carrying out further tasks, such as state-of-the-art visual system, robotic arms, data analyzers, etc.
5) **DB**: Deep Blue is one of the first successful chess computers, which defeated Garry Kasparov, the chess world champion, in 1997 [3]. 20 years ago, this was an important milestone of artificial intelligence research. Its early success lie in the appropriately increased speed and memory of the computers, aided with sophisticated AI programs with multiple optimizations.

6) **DBx**: Let us consider the virtual expansion of the DB, which would resemble on a real robot, just like TAI. DB would be equipped with cameras, wheels, actuators, robotic arms, etc. It would be capable of walking, swimming, but most importantly, it would play chess without human assistance. In this case, DBx would become an intelligent autonomously moving robot. Deep Blue did not need to smile or to move its hands, but one can assume that being capable of implementing this level of knowledge in chess, all these functions could be easily implemented by today's technologies.
7) **IR**: Let us take a welding robot in an assembly line as reference. We composed all its knowledge from 10 different robot definitions and 50 random characteristics of robots collected from the literature.

2.1 The Relationship of the Selected Robots

The Pathfinder (PF) was a live demonstration of human knowledge of artificial intelligence, robotics and telecommunications, measurement techniques and numerous other disciplines. It is for the sake of the game that we added a virtual extension to it to get a more advanced (virtual) robot, PFx. In this study, we use the name TAI instead of Turk to denote the intelligent illusion.

The chess-winner machine from 1770 naturally calls for a real chess-winner computer of our era, which points to the chess-computer Deep Blue, created by IBM (DB). DB could have been extended or upgraded to a more complex, more general, actuated robot from its chess machine status, technically in a relatively easy way. This virtual machine will be called DBx.

We decided to compare the above mentioned devices as robots, therefore it is a logical step to involve a real, "reference" robot (IR) as well [4]. This reference robot was chosen to be a general industrial welding robot, one of the most widely used manufacturing robotic system in today's industry.

The list contains 7 objects. 2 of them (PFx and DBx) are virtual extensions created by our imagination, TAI is an illusion, a hoax, and TM is derived from that illusion. Let us consider the case when we remove the illusion, the derived machine (TAI and TM) and the two extended, virtual machines (DBx and PFx) from the comparison. The 3 remaining robots would be PF, DB and IR.

One can see that only the industrial robot (IR) is a "Traditional" one, if we consider the concept of robotics that is commonly used today. DB and PF are rather specific and goal-oriented, in some aspects "perfect" for their task. DB performed the highest level of AI, using many software resources to play chess, and PF was perfect in tracing, moving (at the speed of max. 40 cm/min), obstacle avoiding, collecting materials, performing measurements and taking photos on the Mars surface, processing and sending/receiving information.

It is expected that the two corresponding upgrades (DBx and PFx) would get the highest marks in any evaluation and comparison. Before we draw any conclusions, let us not forget that these robots were, at a theoretical level, removed from the competition; so to say, they could only become virtual champions. The real competitors are only IR, PF and DB, and in spite of the high performance of TAI, it should be excluded because of cheating. However, for the sake of completeness, all robots will be included in the evaluation.

It is an interesting coincidence that Deep Blue won against the active word champion, Garry Kasparov the same year, when the successful Mars Mission of the Pathfinder was completed. In the past 20 years, more powerful chess computers and better Mars rovers were developed, and this development goes on, which is not part of discussion of this work.

3 The Game of Chess

3.1 Human Versus Human

"Chess is a two-player board game played on a chessboard, a checkered game-board with 64 squares arranged in an eight-by-eight grid. Each player begins the game with 16 pieces: one king, one queen, two rooks, two knights, two bishops, and eight pawns. Each of the six piece types moves differently.

Chess is believed to have originated in India, sometime before the 7th Century; the Indian game of *chaturanga* is also the likely ancestor of *xiangqi* and *shogi*. The pieces took on their current powers in Spain in the late 15th Century and the rules were finally standardized in the 19th Century. The first generally recognized World Chess Champion, Wilhelm Steinitz, claimed his title in 1886. The current World Champion is the Norwegian Magnus Carlsen.

The game structure and nature of chess is related to several combinatorial and topological problems. In 1913, Ernst Zermelo used chess as a basis for his theory of game strategies, which is considered one of the predecessors of game theory.

The number of legal positions in chess is estimated to be between 10^{43} and 10^{47} (a provable upper bound), with a game-tree complexity of approximately 10^{123}. This was first calculated by Claude Shannon as 10^{120}, a number known as the Shannon number. An average position has thirty to forty possible moves, but there may be as few as zero (in the case of checkmate or stalemate) or as many as 218.

The most important challenge of chess is the development of algorithms that can play chess. The idea of creating a chess-playing machine dates to the 18th Century; this was the time, when the chess-playing automaton called The Turk became famous before being exposed as a hoax. Before the development of digital

computing, serious trials based on automata such as El Ajedrecista of 1912, were too complex and limited to be useful for playing full games of chess.

Since the advent of the digital computer in the 1950s, chess enthusiasts, computer engineers and computer scientists have built, with increasing degrees of seriousness and success, chess-playing machines and computer programs. Since the 1990s, computer analysis has contributed significantly to chess theory, particularly in the endgame. The challenges were magnified by Shannon and others with the huge numbers, and Shannon's paper of 1950: "Programming a Computer for Playing Chess". He wrote: "the discrete structure of chess fits well into the digital nature of modern computers" [5]."[1]

3.2 Computer versus Human

"Most players agree that looking at least five moves ahead (five plies) is required to play well. Normal tournament rules give each player an average of three minutes per move. On average there are more than 30 legal moves per chess position, so a computer must examine a quadrillion (10^{15}) possibilities to look ahead ten plies (five full moves). Examining a million positions a second would require more than 30 years. After the discovering refutation screening – the application of alpha-beta pruning to optimizing move evaluation – in 1957, some experts predicted that a computer would defeat the world human champion by 1967 [6].

In the late 1970s chess programs suddenly began defeating top human players. The real breakthrough was in 1980, when Belle (Bell Lab.) began defeating masters. By 1982, two programs played at master level and three were slightly weaker. The sudden improvement without a theoretical breakthrough surprised humans, who did not expect that Belle's ability to examine 100,000 positions a second – about eight plies – would be sufficient. By 1982, microcomputer chess programs could evaluate up to 1,500 moves a second. However, in 1989, Garry Kasparov demonstrated that Deep Thought was still considerably below World Championship Level."[2]

4 The Protagonists of the Game

In our performance study, there are three types of equally important aspects: description of things of interest, robots and people around the robots. We concentrate on facts and technical data, which are interesting enough, sometimes hard to collect, but worthy of study.

1 https://en.wikipedia.org/wiki/Chess
2 https://en.wikipedia.org/wiki/Computer_chess

4.1 An Average Industrial Robot (IR)

"An industrial robot is defined by ISO 8373 as "automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes" [7]. There are 10 other definitions listed in this paper, although it would be hard to collect all existing descriptions from the literature. As these definitions are quite similar in terms of technical details, we could use almost any of them. Typical applications of robots include welding, painting, assembly, pick and place (such as packaging, palletizing and SMT), product inspection, and testing; all accomplished with high endurance, speed, and precision.

Figure 1
Different types of welding robots

Commonly used robot configurations are articulated robots, SCARA robots, delta robots and Cartesian coordinate robots, (gantry robots or x-y-z robots). In the context of general robotics, most types of robot would fall into the category of robotic arms."[3] The most important parameters or simply information and data worth to know about a given robot, or a class of given robots:

- Degree of autonomy
- Intelligence, adaptivity, flexibility
- Number of axes, Degrees of Freedom — these are usually the same
- Kinematics
- Carrying capacity or payload
- Speed — Acceleration — Accuracy — Repeatability
- Power source: electric motors and/or hydraulic actuators
- Motion control
- Drive (gears, direct drive, harmonic drive)
- Compliance
- Etc.tc.

If we did not have the 10 to 50 aspects, listed in the Appendix, we would have to match all other robots to the actual real one and in strength of the above given qualities.

[3] https://en.wikipedia.org/wiki/Industrial_robot

4.2 Short CV of Deep Blue (Chess Computer)

Figure 2
Deep blue playing Gary Kasparov in 1997

"Development for Deep Blue of IBM began in 1985 at Carnegie Mellon University. After some name changes (for example Deep Thought), in 1989 it became Deep Blue again, and in 1995 held the name "Deep Blue prototype". It won a second place on the 8Th World Computer Chess Championship with this name in 1995.

Deep Blue's evaluation function was initially written in a generalized form, with many to-be-determined parameters. The optimal values for these parameters were then determined by the system itself, by analyzing thousands of master games. The evaluation function had been split into 8,000 parts, many of them designed for special positions. In the opening book there were over 4,000 positions and 700,000 grandmaster games. The endgame database contained many six piece endgames and five or fewer piece positions."[4]

Deep Blue was not the first "Deep" chess computer, which Kasparov met. For example, in 1989, he proved in two strong wins that Deep Thought was still far below World Championship Level. The Deep Blue era started in February 1996 and ended in May 1997, almost 20 years ago.

Deep Blue won its first game against a world champion on February 10, 1996, when it defeated Garry Kasparov in game one of a six-game match. However, Kasparov won three and drew two of the following five games, defeating Deep Blue by a score of 4–2.

[4] https://en.wikipedia.org/wiki/Deep_Blue_(chess_computer)

“Deep Blue was then heavily upgraded, the chess knowledge of the program was fine-tuned (unofficially nicknamed "Deeper Blue"), and played Kasparov again in May 1997. Deep Blue won the rematch 3½–2½ by winning the deciding game six after Kasparov made a mistake in the opening. Deep Blue became the first computer system to defeat a reigning world champion in a match under standard chess tournament time controls.

The system derived its playing strength mainly out of brute force computing power. It was a massively parallel, RS/6000 SP Thin P2SC-based system with 30 nodes, with each node containing a 120 MHz P2SC microprocessor, enhanced with 480 special purpose VLSI chess chips. Its chess playing program was written in C and ran under the AIX operating system. It was capable of evaluating 200 million positions per second, twice as fast as the 1996 version. In June 1997, Deep Blue was the 259th most powerful supercomputer according to the TOP500 list, achieving 11.38 GFLOPS on the High-Performance LINPACK benchmark.

Kasparov accused IBM of cheating and demanded a rematch. IBM refused and retired Deep Blue. Writer Nate Silver suggests that a bug in Deep Blue's software led to a seemingly random move (the 44th in the first game) which Kasparov misattributed to "superior intelligence". Subsequently, Kasparov experienced a drop in performance due to anxiety in the following game.” [5]

The numbers defining computer capacities and speeds are steadily increasing as well as computation methodologies as distributed, using grid and cloud computing, etc. We have the feeling that algorithmic changes have to happen soon in computer-chess.

4.3 The Turk, the Chess-Automaton of Kempelen

The idea of creating a chess-playing machine dates back to the eighteenth century [8]. Around 1769, the chess playing automaton called The Turk became famous before being exposed as a hoax (Farkas Kempelen).

“The Turk, also known as the Mechanical Turk or Automaton Chess Player (German: *Schachtürke*, "chess Turk" Hungarian: *A Török*), was a fake chess-playing machine constructed in the late 18th Century. From 1770 until its destruction by fire in 1854 it was exhibited by various owners as an automaton, though it was exposed in the early 1820s as an elaborate hoax. Constructed and unveiled in 1770 by Wolfgang von Kempelen (Hungarian: Kempelen Farkas; 1734-1804) to impress the Empress Maria Theresa of Austria, the mechanism appeared to be able to play a strong game of chess against a human opponent.

The Turk was in fact a mechanical illusion that allowed a human chess master hiding inside to operate the machine. With a skilled operator, The Turk won most of the games played during its demonstrations around Europe and the Americas

[5] http://www.chess.com/blog/ramin18/deep-blue-chess-computer

for nearly 84 years, playing and defeating many challengers including statesmen such as Napoleon Bonaparte and Benjamin Franklin."[6]

According to certain resources, Kempelen found the first very small (dwarf) chess-genie in a dirty pub somewhere in Italy, where he was playing chess for making money. He was escaped from the jail where he was imprisoned due to the major crimes he had committed. Kempelen went to the jail and "purchased freedom" for the mini-champion. When Kempelen was travelling with The Turk, the small man had to be either in the machine or in his hotel room. Which, resulted in tension and several relationship problems between them.

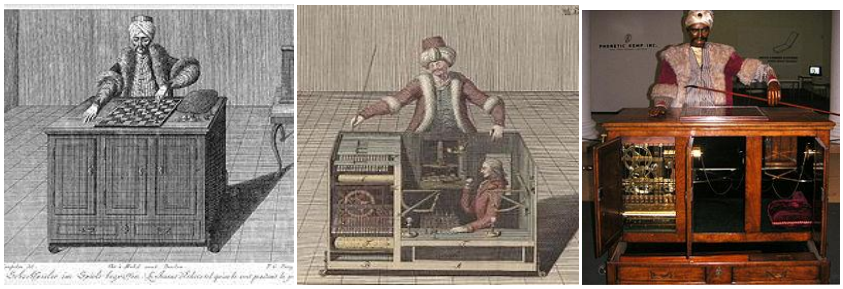


Figure 3
Left to right: Schematic representation of The Turk form the 18th Century, a fantasy image of its interior and the reconstructed machine

"The Turk made its debut in 1770 at Schönbrunn Palace. The machine consisted of a life-sized model of a human head and torso, with a black beard and grey eyes, and dressed in Turkish robes and a turban. Its left arm held a long Turkish smoking pipe while at rest, while its right lay on the top of a large cabinet that measured about three-and-a-half feet (110 cm) long, two feet (60 cm) wide, and two-and-a-half feet (75 cm) high. Placed on the top of the cabinet was a chessboard, which measured eighteen inches square (~11x11 cm). The front of the cabinet consisted of three doors, an opening, and a drawer, which could be opened to reveal a red and white ivory chess set.

The interior of the machine was very complicated and designed to mislead those who observed it. When opened on the left, the front doors of the cabinet exposed a number of gears and cogs similar to clockwork. The section was designed so that if the back doors of the cabinet were open at the same time one could see through the machine. The other side of the cabinet did not house machinery; instead it contained a red cushion and some removable parts, as well as brass structures. This area was also designed to provide a clear line of vision through the machine. Underneath the robes of The Turkish model, two other doors were hidden. These also exposed clockwork machinery and provided a similarly unobstructed view

6 https://en.wikipedia.org/wiki/The_Turk

through the machine. The design allowed the presenter of the machine to open every available door to the public, to maintain the illusion.

Neither the clockwork visible to the left side of the machine nor the drawer that housed the chess set extended fully to the rear of the cabinet; they instead went only one third of the way. A sliding seat was also installed, allowing the director inside to slide from place to place and thus evade observation as the presenter opened various doors. The sliding of the seat caused dummy machinery to slide into its place to further conceal the person inside the cabinet.

The chessboard on the top of the cabinet was thin enough to allow for a magnetic linkage. Each piece in the chess set had a small, strong magnet attached to its base, and when they were placed on the board the pieces would attract a magnet attached to a string under their specific places on the board. This allowed the director inside the machine to see which pieces moved where on the chess board. The bottom of the chessboard had corresponding numbers, 1-64, allowing the director to see which places on the board were affected by a player's move. The internal magnets were positioned in a way that outside magnetic forces did not influence them, and Kempelen would often allow a large magnet to sit at the side of the board in an attempt to show that the machine was not influenced by magnetism. The authors of this paper suppose that some mirrors were involved as well to inform the operator about the board positions and about the people around the machine.

The interior also contained a pegboard chess board connected to a pantograph-style series of levers that controlled the model's left arm. The metal pointer on the pantograph moved over the interior chessboard, and would simultaneously move the arm of The Turk over the chessboard on the cabinet. The range of motion allowed the director to move The Turk's arm up and down, and turning the lever would open and close The Turk's hand, allowing it to grasp the pieces on the board.

The pantograph was one of the secret parts of The Turk, and as long as people believed in the automatic behavior of the chess machine it could have been taken as an intelligent robot with chess playing knowledge, with arm, drives, joints, gripper and a controller. Even when it became clear that no automatic behavior is present, the pantograph-like system could still be taken at least as a manipulator from 1770. As such, it is a robot for us, worth dealing with.

All of this was made visible to the director by using a simple candle, which had a ventilation system through the model. Other parts of the machinery allowed for a clockwork-type sound to be played when The Turk made a move, further adding to the machinery illusion, and for The Turk to make various facial expressions. A voice box was added some years later, allowing the machine to say "Échec!" (French for "check") during matches.

An operator inside the machine also had tools to assist in communicating with the presenter outside. Two brass discs equipped with numbers were positioned opposite each other on the inside and outside of the cabinet. A rod could rotate the discs to the desired number, which acted as a code between the two.

The Turk could nod twice if it threatened its opponent's queen, and three times upon placing the king in check. If an opponent made an illegal move, The Turk would shake its head, move the piece back and make its own move, thus forcing a forfeit of its opponent's move. The Turk also had the ability to converse with spectators using a letter board. The director, whose identity during the period when Kempelen presented the machine at Schönbrunn Palace is unknown, was able to do this in English, French, and German."[7]

4.4 The Short Story of the Pathfinder

"Sojourner was the Mars Pathfinder robotic Mars rover that landed on July 4, 1997 and explored Mars for around three months. It had front and rear cameras and hardware to conduct several scientific experiments. Designed for a mission lasting 7 sols (7x24 hours), with possible extension to 30 sols, it was in fact active for 83 sols. The base station had its last communication session with Earth at 3: a.m. Pacific Daylight Time on September 27, 1997. The rover needed the base station to communicate with Earth, despite still functioning at the time communications ended.

Sojourner traveled a distance of just over 100 meters (330 ft.) by the time communication was lost. It was instructed to stay stationary until October 5, 1997 (sol 91) and then drive around the lander.

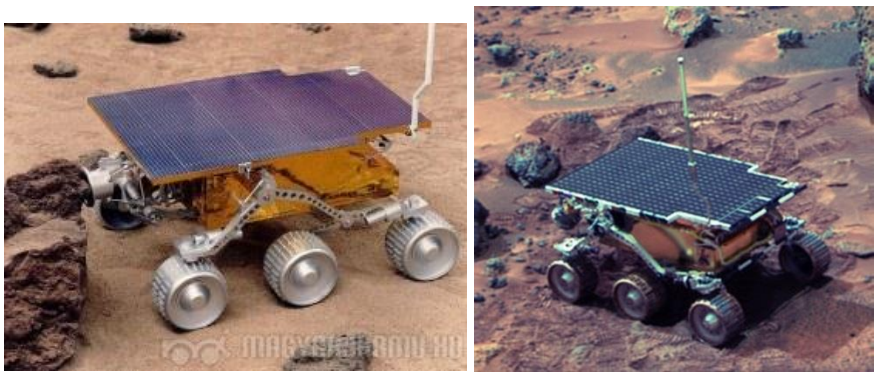


Figure 4
The Sojourner microrover

[7] https://en.wikipedia.org/wiki/The_Turk

The rover's name, *Sojourner*, means "traveler", however its name was selected after the name of a famous abolitionist and women's rights activist Sojourner Truth. The rover was also known as *Microrover Flight Experiment* abbreviated MFEX.

Sojourner has solar panels and a non-rechargeable battery, which allowed limited nocturnal operations. Once the batteries were depleted, it could only operate during the day. The batteries are lithium-thionyl chloride (LiSOCl2) and could provide 150 watt-hours. The batteries also allowed the health of the rover to be checked while enclosed in the cruise stage while en route to Mars.

0.22 square meters of solar cells could produce a maximum of about 15 watts on Mars, depending on the conditions. The cells were GaAs/Ge (Gallium Arsenide/Germanium) and capable of about 18 percent efficiency. They could survive down to about -140° Celsius (-220 °F).

Its central processing unit (CPU) is an 80C85 with a 2 MHz clock, addressing 64 Kbytes of memory. It has four memory stores; the previously mentioned 64 Kbytes of RAM (made by IBM) for the main processor, 16 Kbytes of radiation-hardened PROM (made by Harris), 176 Kbytes of non-volatile storage (made by Seeq Technology), and 512 Kbytes of temporary data storage (made by Micron). The electronics were housed inside the *Warm Electronics Box* inside the rover.

It communicated with the base station with 9,600 baud radio modems. The practical rate was closer to 2,600 baud with a theoretical range of about half a kilometer. The rover could travel out of range of the lander, but its software would need to be changed to that mode. Under normal driving, it would periodically send a "heartbeat" message to the lander. The UHF radio modems worked similar to walkie-talkies, but sent data, not voice. It could send or receive, but not both at same time, which is known as half-duplex. The data was communicated in bursts of 2 kilobytes.

The Alpha Proton X-ray Spectrometer (APXS) is nearly identical to the one on Mars 96, and was a collaboration between the Max Planck Institute for Solar System Research in Lindau, Germany (formally known as the Max Planck Institute For Aeronomy) and the University of Chicago in the United States. APXS could determine elemental composition of Mars rocks and dust, except for hydrogen. It works by exposing a sample to alpha particles, then measuring the energies of emitted protons, X-rays, and backscattered alpha particles.

The rover had three cameras: 2 monochrome cameras in front, and a color camera in the rear.[12] Each front camera had an array 484 pixels high by 768 wide. The optics consisted of a window, lens, and field flattener. The window was made of sapphire, while the lens objective and flattener were made of zinc selenide. The rover was imaged on Mars by the base station's IMP camera system, which also helped determine where the rover should go.

Sojourner operation was supported by *Rover Control Software*, which ran on a Silicon Graphics Onyx2 computer back on Earth, and allowed command sequences to be generated using a graphical interface. The rover driver would wear 3D goggles supplied with imagery from the base station and move a virtual model with the spaceball controller, a specialized joystick. The control software allowed the rover and surrounding terrain to be viewed from any angle or position, supporting the study of terrain features, placing waypoints, or doing virtual flyovers.

The rover had a mass of 11.5 kg (weighing about 25 pounds on Earth), which equates to a weight of 4.5 kg (10 pounds) on Mars."[8]

5 Evaluation and Comparison Methods

In the previous Section, all the machines/robots of concern were explained in detail. In order to compare their performance, several definitions, properties and abilities have been collected in Table III. Table III, contains important information of 7 objects, which represent real robots and virtual robots.

D stands for robot definitions (1-10), P means properties (1-30) and A denotes abilities (1-20). $W_i, i = 1\ldots10$ are weights corresponding to the D values (1-10), while $V_j, j = 1\ldots50$ are the weights corresponding to P and A values.

In order to fine-tune the evaluations, marks were attached to every line according to their importance. Definitions were marked as very important in this approach, therefore the corresponding weights were assigned the maximum value of 10. Other secondary features, such as properties and abilities were given the weight of 5. Those features, which have little importance in the evaluation, have been assigned the value of 1. The evaluation was done by adding all numbers of the weights W for D1-D10, V for P1-P30 and for A1-A20, where there is a *yes* in the object's column. These sums will define the ranking of the robots to be compared.

5.2 Numerical Results

The creation of the Table III. was done by collecting data from different sources. However, it is not a trivial task to find properties that would match with all the 7 objects due to their diversity. And after several attempts to find appropriate weight values and proper *yes* and *no* answers (y, n) in Table III., several calculations have been carried out, then the weighting factors were adjusted in order to match a real ranking.

[8] https://en.wikipedia.org/wiki/Sojourner_(rover)

5.1.1 The Question of Yes/No Answers

If all weight factors W_i would be 10 and V_j would be 5, the theoretical maximum values would be the same for each object.

$$M_{\max} = 10 \times 10 + (30 + 20) \times 5 = 100 + 250 = 350 \quad (1)$$

Keeping the values of W_i and V_j according to this setting, the weighted score for competitors was collected in Table I.

Table I
Ranking of the competitors using maximum the maximum weight values

	TAI	TM	DB	DBx	IR	PF	PFx
D:W1-W10	8	0	7	8	7	5	8
F: V1-V50	25	9	22	35	20	37	40
10xD+5xF	205	45	180	250	170	235	280

If there are only *yes* answers in the boxes of a column in Table III, the object in the column cannot be beaten, as it gets the maximum evaluation value. Consequently, the number of *yes* answers in a column has decisive role. However, let us suppose that there are only *yes* answers in the D1-D10 positions and all others in P and A are *no*. Let the W_i values be (10,10,10,1,1,1,1,1,1,1), respectively. If an object gets 10 points only 3 times (e.g. D1-D3), and another one gets 1 points 7 times (e.g. D4-D10), the one with 3 *yes* values (30) will beat the other one with 7 yes values (7). If Table III would get arbitrary W_i and V_j values, other conclusions could be drawn, but normally there is technical content that defines these weights.

In order to address the question of determining the final weights, the following problem can be formulated:

Given the weight factors W_i for D1-D10, V_j for P1-P30 and A1-A20, i.e. V1-V50, it is required to find values W_i and V_j, for which the final ranking is the same as the one resulting from Table I, for the case of $W_i = 10$ and $V_j = 5$.

In order to solve this problem, there is no need to use both W_i and V_j, as it is only their relation to each other that has a direct effect on the results. Let this relation be $x = W/V$, assuming that $W_i = W$ and $V_i = V$.

Let us take the original results for $W = 10$ and $V = 5$:

$$M(PFx) \geq M(DBx) \geq M(PF) \geq M(TAI) \geq M(DB) \geq M(IR) \geq M(TM)$$

We use the following program written in Wolfram Mathematica:
wmin = 0
wmax = 10

Reduce [8 x + 40 >= 8 x + 35 >= 5 x + 37 >= 8 x + 25 >=7 x + 22 >= 7 x + 20 >= 0 x + 9 && Wmin <= x <= Wmax, x]

The result of the calculation is:

$$\frac{2}{3} \le x \le 4 \tag{1}$$

This means that if $2/3V \le W \le 4V$, then the given ranking survives. Similar calculations can be done for different weight values and for different other problems as well.

5.1.2 Ordering as Function of the Weight Parameters

The maximum points reachable in theory, if we assign 10 points to each W_i, is 350. Let us consider the following robot types:

$\mathrm{R} = \{\mathrm{PFx}, \mathrm{DBx}, \mathrm{PF}, \mathrm{TAI}, \mathrm{IR}, \mathrm{DB}, \mathrm{TM}\}$.

Let W be the weight for definitions (rows 3-13 of Table III) and V be the weight for properties and abilities (rows 17-71 of Table III). The score of each robot type is $\mathrm{M(r)} = \mathrm{n_1(r)W} + \mathrm{n_2(r)V}$, where $\mathrm{n_1(r)}$ is the number of *yes* answers for robot type r among definitions and $\mathrm{n_2(r)}$ is the number of *yes* answers among properties and abilities of robot type r. Each pair of weights (W, V) results in some descending order among robot types according to their weights $\mathrm{M(r)}$, e.g. if $\mathrm{W} = 10$ and $\mathrm{V} = 5$, the order is $\mathrm{M(PFx)} \geq \mathrm{F(DBx)} \geq \mathrm{B(PF)} \geq \mathrm{Fx(TAI)} \geq \mathrm{A(IR)} \geq \mathrm{R(DB)} \geq \mathrm{BIhe}$.

There are $7! = 5040$ possible orders of 7 elements and for each rearrangement. In the following, we study the values of W and V that result in a predefined ranking. In other words for each permutation: $\phi: \overline{1,7} \leftrightarrow \mathrm{R}$ we find the constraints on the values of W and V such as

$$\forall i \in \overline{1,6} \quad \mathrm{n_1}(\phi(i))\mathrm{W} + \mathrm{n_2}(\phi(i))\mathrm{V} \geq \mathrm{n_1}(\phi(i+1))\mathrm{W} + \mathrm{n_2}(\phi(i+1))\mathrm{V}. \tag{3}$$

It is easy to see, that if V is not zero, and then each inequality from (3) may be divided by V and reduced to one-variable inequality

$$\forall i \in \overline{1,6} \quad \mathrm{n_1}(\phi(i))x + \mathrm{n_2}(\phi(i)) \geq \mathrm{n_1}(\phi(i+1))x + \mathrm{n_2}(\phi(i+1)), \tag{4}$$

where $x = W/V$. We suppose that $x \in [0, \infty)$ and find the values of x corresponding to all possible permutations; the permutations that cannot be attained by any values of x (vast majority) are omitted (see Table II).

Table II
Different orders of robot types according to their weights $M(r)$ depending on "relative definitions-properties weight" $x = W/V$

Order of robot types	Corresponding interval of $x = W/V$
[PFx, PF, DBx, TAI, DB, IR, TM]	[0, 2/3]
[PFx, DBx, PF, TAI, DB, IR, TM]	[2/3, 2]
[PFx, DBx, PF, TAI, IR, DB, TM]	[2, 10/3]
[PFx, DBx, TAI, PF, IR, DB, TM]	[10/3, 17/2]
[PFx, DBx, TAI, IR, PF, DB, TM]	[17/2, 15]
[PFx, DBx, TAI, IR, DB, PF, TM]	[15, ∞)

According to Table II, it is clear that the gold medal holders from the initial ranking (PFx and DBx) are on the first positions, regardless the values of the weights, while TM is taking the last place. In the mid-field however there are some interesting positions of the others, that would be worthy of further study.

Conclusions

In this paper, we began to look for famous robot experts of Hungarian ancestry, finding a connection between Farkas Kempelen and Tony Bejczy, the two most important of them. The chess automaton of Kempelen, The Turk directly led us to Kasparov and Deep Blue. This simple path gave us the chance to introduce the game of chess, and the miraculous machines Deep Blue, The Turk and Pathfinder. Virtual machines based on these systems were introduced.

The detailed explanation of the machines started with Pathfinder (PF) and The Turk (TAI). A general industrial welding robot (IR) was added to the list, serving a as a reference for further evaluation. The two virtual extensions were made using the two most successful items: the Pathfinder (PF → PFx) and the Deep Blue (DB → DBx), which resulted in two unbeatable robots in the proposed competition.

The final score: Virtual robots were disqualified from the competition for obvious reasons, while The Turk has been dropped out for cheating, as it were in any competition. The final score reads: Deep Blue: 9.5, Pathfinder: 9 ½. It is a tie, but if there were only one gold medal, in the authors' opinion, it would go to Bejczy and the Pathfinder, since DB is too specialized, and the PF to PFx transition would be much easier to achieve than the DB to DBx transition.

It is important to note that arbitrarily chosen weight values may strongly influence the results. The method presented in this article is suitable for solving this problem. The experiments with fine-tuning, which differentiate between qualities could completely change the final scores, led us to the conclusion that according to the scoring table, ranking possibilities can be well differentiated from each other, as well.

Acknowledgements

The efforts of Y. Ivanko were supported by the Russian Scientific Foundation grant 14-11-00109. The work was supported by Act of the Government of the Russian Federation, contract № 02.A03.21.0006

References

[1] NASA - Mars Pathfinder and Sojourner. [Online]. Available: http://www.nasa.gov/mission_pages/mars-pathfinder/ [Accessed: 12-Dec-2015]

[2] K. G. von Windisch, *Inanimate Reason, Or a Circumstantial Account of that Astonishing Piece of Mechanism*, Bladon, 1784

[3] F. Hsu, *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*. Princeton University Press, 2002

[4] IFR International Federation of Robotics. [Online] Available: http://www.ifr.org/ [Accessed: 12-Dec-2015]

[5] C. E. Shannon, "XXII. Programming a Computer for Playing Chess," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Vol. 41, No. 314, pp. 256-275, Mar. 1950

[6] D. Levy and M. Newborn, *How Computers Play Chess*. New York, NY, USA: Computer Science Press, Inc., 1991

[7] C. Leroux, "A Green Paper on Legal Issues in Robotics euRobotics Coordination Action," *International Workshop on Autonomics and Legal Implications*, Berlin, 2012

[8] R. Szalatnai, *Kempelen a varázsló* (Kempelen, the Magician), Móra Ferenc, Budapest, 2004

[9] NASA - Mars Pathfinder and Sojourner. [Online]. Available: http://www.nasa.gov/mission_pages/mars-pathfinder/ [Accessed: 12-Dec-2015]

Appendix

A1. Robot definitions

D1. The simplest correct definition: „A re-programmable Manipulator – the same machine can be used to solve different tasks, by simply changing its control program"

D2.Wikipedia "A robot is a mechanical or virtual intelligent agent which can perform tasks on its own, or with guidance. In practice a robot is usually an electro-mechanical machine which is guided by computer and electronic programming".

D3. Encyclopaedia Britannica, a sociological definition: "any automatically operated machine that replaces human effort, though it may not resemble human beings in appearance or perform functions in a humanlike manner".

D4. Webopedia 2: "A program that runs automatically without human intervention. Typically, a robot is endowed with some artificial intelligence so that it can react to different situations it may encounter. Two common types of robots are *agents* and *spiders*."

D5. Oxford 1: a machine that can perform a complicated series of tasks automatically

D6.Oxford 2 (especially in stories): „a machine that is made to look like a human and that can do some things that a human can do"

D7.Merriam-Webster a) „a machine that looks like a human being and performs various complex acts of a human being (as walking or talking)";

D8. Merriam-Webster b):"a device that automatically performs complicated often repeti*tive tasks";*

D9.Merriam-Webster c) „ a mechanism guided by automatic controls".

D10.ISO 8373, "an actuated mechanism programmable in two or more axes (*directions used to specify the robot motion in a linear or rotary mode*) with a degree of autonomy, moving within its environment, to perform intended tasks"

A2. List of properties for comparison

Communication with the external world, programmable manipulator, activities similar to men, independent agent in the world, completely human made, autonomous, able to move with 3-7 degrees of freedom, complex, as works in the real world, hardware and sensors really work, AI tools, teleoperator CNC based, Generation 1 -moves, Generation 2 –sensors, Generation 3 -complex signal processing, Intelligence 0, Intelligence 1, Intelligence 2, mobile, collects and evaluates sensory input, solves complex problems, has legs, has wheels, obstacle avoidance, moving instructions what to recognize, autonomous, on the ground, energy, solar cells, fixed, extra robots, nano.

A3. A list of robot abilities for comparison

See, act, localize, compute, navigate, transport, manipulate, talk learn, observe, smell, cooperate, work, dialog, play, stimulate, fly, move, create, make reasoning.

A4. The Table containing all data to compare the 7 objects

Table III

Important data of the 7 competing objects against reference values

No.	W	**D No.xx (D1-D10)**	TAI	TM	DB	DB x	IR	PF	PFx
1		D1 see above	y	n	n	y	y	n	n
2		D2 see above	n	n	y	y	n	y	y
3		D3 see above	y	n	y	y	y	y	y
4		D4 see above	n	n	y	y	n	n	y
5		D5 see above	y	n	y	y	y	y	y
6		D6 see above	y	n	y	y	n	n	y
7		D7 see above	y	n	n	n	y	n	y
8		D8 see above	y	n	y	y	y	y	y
9		D9 see above	y	n	y	y	y	y	y
10		D10 see above	y	n	n	n	y	n	n
	P	**PROPERTIES (P1-P30)**							
1		activities similar to men	y	n	y	y	y	n	n
2		independent agent in the world	**n**	n	y	y	n	y	y
3		communication with the world	y	y	y	y	y	y	y
4		programmable manipulator	y	n	n	y	y	y	y
5		completely human made	y	y	y	y	y	y	y
6		autonomous	n	n	y	y	n	y	y
7		able to move with 3-7 DoF	n	n	n	y	y	y	y
8		works in the REAL world	y	y	y	y	y	y	y
9		hardware REALLY works	y	y	n	y	n	y	y
10		AI applications	n	n	y	y	y	y	y
11		teleoperator CNC based	n	n	n	y	y	y	y
12		generation 1 -moves	n	n	n	y	n	y	y
13		generation 2 -sensors	y	n	n	y	y	y	y
14		generation 3- complex signal proc.	y	n	y	y	y	y	y
15		intelligence 0	y	n	y	y	y	y	y
16		intelligence 1	y	n	y	y	n	y	y
17		intelligence 2	y	n	y	y	n	y	y
18		mobile	n	n	n	y	n	y	y
19		collects and evaluates sensory inp.	n	n	n	n	y	y	y
20		solves complex problems	y	n	y	y	y	y	y
21		has	n	n	n	n	n	n	n
22		has wheels	n	n	n	y	n	y	y
23		obstacle avoidance	n	n	n	n	n	y	y
24		moving instructions	n	n	n	n	n	y	y
25		on the ground	y	y	y	y	y	y	y
26		autonomous	y	n	n	y	n	y	y
27		energy, solar cells	n	n	n	n	n	y	y
28		fixed	y	y	y	n	y	n	n
29		extra robots	n	n	n	n	n	n	y
30		nano	n	n	n	n	n	n	n

	A	ABILITIES (A1-A20)							
1		see	y	n	n	y	n	y	y
2		act	y	y	y	y	n	y	y
3		localize	n	n	n	n	n	n	y
4		compute	y	n	y	y	y	y	y
5		navigate	n	n	n	y	n	y	y
6		transport	n	n	n	n	n	y	y
7		manipulate	y	y	n	n	y	y	y
8		talk	y	y	n	y	n	n	n
9		learn	y	n	y	y	n	y	y
10		observe	y	n	n	y	n	y	y
11		smell	n	n	n	n	n	n	n
12		cooperate	n	n	n	n	n	n	y
13		work	n	n	n	n	y	y	y
14		dialog	y	n	y	y	y	y	y
15		play	y	n	y	y	n	n	n
16		stimulate	n	n	y	y	n	n	n
17		fly	n	n	n	n	n	n	n
18		move	n	n	n	y	n	y	y
19		create	n	n	y	y	n	n	n
20		make reasoning	y	n	y	y	y	y	y

Data-driven Model-Free Adaptive Control Tuned by Virtual Reference Feedback Tuning

Raul-Cristian Roman[1], Mircea-Bogdan Radac[1], Radu-Emil Precup[1], Emil M. Petriu[2]

[1]Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, 300223 Timisoara, Romania
E-mail: raul-cristian.roman@student.upt.ro, mircea.radac@upt.ro, radu.precup@upt.ro

[2]School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward, Ottawa, Ontario, Canada, K1N 6N5
E-mail: petriu@uottawa.ca

Abstract: This paper proposes a new tuning approach, by which, all parameters of a data-driven Model-Free Adaptive Control (MFAC) algorithm are automatically determined using a nonlinear Virtual Reference Feedback Tuning (VRFT) algorithm. The approach is referred to as mixed MFAC-VFRT control and it leads to mixed MFAC-VFRT algorithms. An advantage of mixed MFAC-VFRT control, is that it combines systematically, the features of VRFT (it computes the controller parameters using only the input/output data) with those of MFAC. This is especially illustrated by comparison with the classical MFAC algorithms, the initial values of the parameters, which are obtained through a process that involves solving an optimization problem. The application that validates the mixed MFAC-VFRT algorithms, by experiment, is a nonlinear twin rotor aerodynamic system laboratory equipment position control system, that represents a tribute, to Prof. Antal (Tony) K. Bejczy for his excellent results in space robotics, robot dynamics and control, haptics and force perception/control.

Keywords: Model-Free Adaptive Control; twin rotor aerodynamic system; optimal control; state-space model; Virtual Reference Feedback Tuning

1 Introduction

Virtual Reference Feedback Tuning (VRFT) is a technique used for data-driven controllers. VRFT was first proposed and applied in [1] to Single Input-Single Output (SISO) systems, then in [2] Multi Input-Multi Output (MIMO) systems and next extended in [3, 4] a nonlinear version. The main process of VRFT consists of collecting the input/output (I/O) data from an unknown open-loop

process, and with this data, computes the controller parameters. A disadvantage of this technique is that it does not guarantee the closed loop control system (CS) stability.

As presented in [5, 6], the main features of Model-Free Adaptive Control (MFAC) is that MFAC algorithms make use of only the online I/O data of the process, and they ensure CS stability through reset conditions related to a so-called Pseudo-Partial-Derivatives (PPD) matrix.

Using the complementary features of MFAC and VRFT, this paper proposes a mixed MFAC-VFRT control approach. This mixed algorithm is also successfully applied in [7] to a class of nonlinear MIMO systems. The approach aims to control the azimuth and pitch motions of the Twin Rotor Aerodynamic System (TRAS), i.e., a representative process for nonlinear robotics, space and automotive applications [8]-[12] with focus on the seminal contributions of Prof. Antal (Tony) K. Bejczy, to whose memory, this paper is dedicated.

As proven in [13] for TRAS, the MFAC algorithms behave practically, like classical integral controllers, because the PPD matrix is almost constant, and this motivates the need for combination with other data-driven techniques. The mixed MFAC-VFRT control approach is time saving in finding the optimal parameters of the classical MFAC algorithm, which has five parameters in the SISO scenario and eight parameters in the MIMO scenario, for the TRAS laboratory equipment considered in this paper. This is especially important, as a basis for other combinations of data-driven control approaches [14, 15].

The paper is organized as follows. Section 2 describes the TRAS laboratory equipment. An overview on MFAC and nonlinear VRFT is presented in Section 3. The MFAC-VFRT control approach is shown in Section 4. The experimental validation is done in Section 5 and conclusions are outlined in the final section.

2 Twin Rotor Aerodynamic System

The nonlinear state-space model that describes the MIMO TRAS process is [16]:

$$\begin{aligned}
&\dot{\Omega}_h = [l_t F_h(\omega_t)\cos\alpha_v + \Omega_h f_h + u_2 k_{vh}]/J_h,\\
&\dot{\Omega}_v = \{l_m F_v(\omega_m) + \Omega_v f_v + g[(A-B)\cos\alpha_v - C\sin\alpha_v]\\
&-(\Omega_h^2/2)(A+B+C)\sin 2\alpha_v\}/J_v,\\
&\dot{\alpha}_h = \Omega_h,\ \dot{\alpha}_v = \Omega_v,\ \dot{\omega}_h = (u_1 - \omega_h/k_{Hh})/I_h,\ \dot{\omega}_v = (u_2 - \omega_v/k_{Hv})/I_v,\\
&y_1 = \alpha_h, y_2 = \alpha_v,
\end{aligned} \tag{1}$$

where: $u_1[\%]$ – the first control input, i.e., the PWM duty cycle of the horizontal (main) direct current (DC) motor, $u_2[\%]$ – the second control input, i.e., the

PWM duty cycle of the vertical (tail) DC motor, $\alpha_h\,[\text{rad}] = y_1$ – the first process output, i.e., the azimuth (horizontal) position of the beam that supports the main and the tail rotor, $\alpha_v\,[\text{rad}] = y_2$ – the second process output, i.e., the pitch (vertical) position of the beam. The linearization of (1) at the equilibrium point leads to the linearized state-space model of the process, which consists of the third to eighth equations plus the first two equations replaced by [17, 18]:

$$\dot{\Omega}_h = a_{14}\alpha_v + a_{15}\omega_h,\ \dot{\Omega}_v = a_{34}\alpha_v + a_{36}\omega_v, \tag{2}$$

where all variables are expressed as deviations with respect to the equilibrium point.

The typical control objective for TRAS is to ensure the regulation and tracking for vertical and horizontal motions, i.e., to control the azimuth and the pitch. This paper considers a MIMO CS that is decomposed into two SISO CSs, namely the azimuth control loop and the pitch control loop. Although the theory will be presented as follows, in the general MIMO case, the experimental results will be given in Section 5 for both SISO CSs.

3 Overview on MFAC and Nonlinear VRFT

3.1 MFAC

MFAC is developed using the MIMO nonlinear discrete-time process model:

$$\mathbf{y}(k+1) = \mathbf{f}(\mathbf{y}(k),...,\mathbf{y}(k-n_y),\mathbf{u}(k),...,\mathbf{u}(k-n_u)), \tag{3}$$

where $\mathbf{y}(k) = [y_1(k) \quad y_2(k)]^T \in \mathbf{R}^{2\times 1}$ is the controlled output vector, $\mathbf{u}(k) = [u_1(k) \quad u_2(k)]^T \in \mathbf{R}^{2\times 1}$ is the control input vector, T stands for matrix transposition, n_y and n_u are the unknown process orders and $\mathbf{f}$ is an unknown nonlinear vector-valued function, $\mathbf{f}: \mathbf{R}^{2(n_y+n_u+2)} \to \mathbf{R}^2$. The partial derivatives of $\mathbf{f}$ with respect to the elements of the vector $\mathbf{u}(k)$ (i.e., the control inputs) are assumed to be continuous.

Since the Compact Form Dynamic Linearization (CFDL) is the most popular version of MFAC [5], this paper treats only the CFDL version. The results can be extended to other versions as well. According to [5] the PPD matrix $\mathbf{\Phi}(k)$ exists such that (3) can be transformed into the following CFDL-MFAC data model:

$$\Delta\mathbf{y}(k+1) = \mathbf{\Phi}(k)\Delta\mathbf{u}(k), \tag{4}$$

where $\mathbf{\Phi}(k)=[\phi_{ij}(k)]_{i,j\in\{1,2\}}$, $\|\mathbf{\Phi}(k)\|\le b$. These conditions concerning $\mathbf{\Phi}(k)$ are met only if the model in (3) is assumed Lipschitz, i.e., $\|\Delta\mathbf{y}(k+1)\|\le b\|\Delta\mathbf{u}(k)\|$ for each fixed discrete time moment k, and $\|\Delta\mathbf{u}(k)\|\neq 0$, with $\Delta\mathbf{y}(k+1)=\mathbf{y}(k+1)-\mathbf{y}(k)$, $\Delta\mathbf{u}(k)=\mathbf{u}(k)-\mathbf{u}(k-1)$ and $b=\text{const}>0$.

The MFAC objective is to solve the optimization problem [5]:

$$\begin{aligned}&\mathbf{u}^*(k)=\arg\min_{\mathbf{u}(k)} J_{MFAC}(\mathbf{u}(k)),\\&J_{MFAC}(\mathbf{u}(k))=\|\mathbf{y}^*(k+1)-\mathbf{y}(k+1)\|^2+\lambda\|\Delta\mathbf{u}(k)\|^2,\end{aligned}\tag{5}$$

where $\mathbf{y}^*(k+1)=[y_1^*(k+1)\quad y_2^*(k+1)]^T$ is the tracking reference input vector and $\lambda\ge 0$ is a weighting parameter. The estimate of $\mathbf{\Phi}(k)$ is computed using the I/O data from the process, this matrix should be diagonally dominant and bounded:

$$|\phi_{ij}(k)|\le b_1,\ b_2\le|\phi_{ii}(k)|\le a\,b_2, i,j\in\{1,2\}, i\neq j, a\ge 1, b_2>b_1(2a+1),\tag{6}$$

where the signs of all elements of $\mathbf{\Phi}(k)$ should remain unchanged.

The estimate $\hat{\mathbf{\Phi}}(k)$ of the PPD matrix $\mathbf{\Phi}(k)$ is:

$$\hat{\mathbf{\Phi}}(k)=\hat{\mathbf{\Phi}}(k-1)+\frac{\eta[\Delta\mathbf{y}(k)-\hat{\mathbf{\Phi}}(k-1)\Delta\mathbf{u}(k-1)]\Delta\mathbf{u}^T(k-1)}{\mu+\|\Delta\mathbf{u}(k-1)\|^2},\tag{7}$$

where $0<\eta<1$ is a step size constant and $\mu>0$ is another weighting factor parameter, different to optimal control. The resetting conditions are:

$$\begin{aligned}&\hat{\phi}_{ii}(k)=\hat{\phi}_{ii}(1),\ \text{if}\ |\hat{\phi}_{ii}(k)|<b_2\ \text{or}\ |\hat{\phi}_{ii}(k)|>a\,b_2\ \text{or}\ \text{sgn}(\hat{\phi}_{ii}(k))\neq\text{sgn}(\hat{\phi}_{ii}(1)),\\&\hat{\phi}_{ij}(k)=\hat{\phi}_{ij}(1),\ \text{if}\ |\hat{\phi}_{ij}(k)|>b_1\ \text{or}\ \text{sgn}(\hat{\phi}_{ij}(k))\neq\text{sgn}(\hat{\phi}_{ij}(1)),\ i\neq j,\end{aligned}\tag{8}$$

where $\hat{\phi}_{ij}(1)$ is the initial value of $\hat{\phi}_{ij}(k)$, $i\in\{1,2\}$, $j\in\{1,2\}$. According to [5], the substitution of $\mathbf{y}(k+1)=\mathbf{y}(k)+\mathbf{\Phi}(k)\Delta\mathbf{u}(k)$ into (5) leads to the control law specific to MFAC algorithms:

$$\mathbf{u}(k)=\mathbf{u}(k-1)+\frac{\rho\hat{\mathbf{\Phi}}^T(k)[\mathbf{y}^*(k+1)-\mathbf{y}(k)]}{\lambda+\|\hat{\mathbf{\Phi}}(k)\|^2},\tag{9}$$

where $\rho>0$ is another step size constant. Finding the parameters $\hat{\mathbf{\Phi}}(1)$, ρ, η, λ, μ of the MFAC algorithm is a difficult task, without a model of the controlled process and guidelines for appropriate selection, which do not exist to the best of authors' knowledge. This procedure involving a process model is usually an optimization problem, which is solved for a specified control scenario as illustrated in [17, 18]. However, this defies the purpose of MFAC and prevents it from being a truly model-free approach. The parameters of MFAC are obtained in a nonlinear VRFT framework that will be introduced in Section 4.

3.2 Nonlinear VRFT

Nonlinear VRFT uses a linear or a nonlinear reference model, which ultimately must be tracked by the closed-loop CS. Nonlinear VRFT uses only a single open-loop experiment, where a rich spectrum frequency signal is applied as input to the stable nonlinear process, then the I/O signals are collected, and then used to compute the controller parameters [3, 4, 19].

The model reference objective function (o.f.) used in nonlinear VRFT is [19]:

$$J_{MR}(\boldsymbol{\theta}) = \sum_{k=1}^{N} \left\| \mathbf{y}_{\boldsymbol{\theta}}(k) - \mathbf{y}^{d}(k) \right\|^{2}, \tag{10}$$

where $\mathbf{y}_{\boldsymbol{\theta}}(k+1) = \mathbf{f}(\mathbf{y}(k),...,\mathbf{y}(k-n_y),\mathbf{u}_{\boldsymbol{\theta}}(k),...,\mathbf{u}_{\boldsymbol{\theta}}(k-n_u))$ is the nonlinear process output vector, $\mathbf{u}_{\boldsymbol{\theta}}(k) = C_{\boldsymbol{\theta}}(\boldsymbol{\theta},\mathbf{u}(k-1),\ldots,\mathbf{u}(k-n_{uc}),\mathbf{e}(k),\ldots,\mathbf{e}(k-n_{ec}))$ (in shorthand notation expressed as $\mathbf{u}_{\boldsymbol{\theta}}(k) = C_{\boldsymbol{\theta}}(\boldsymbol{\theta},\mathbf{u}(k-1),\mathbf{e}(k))$) is the nonlinear controller output vector, with uc and ec – the known orders of the fixed structure controller parameterized by the vector $\boldsymbol{\theta}$, $\mathbf{e}(k) = \mathbf{r}(k) - \mathbf{y}_{\boldsymbol{\theta}}(k)$ is the tracking error, $\mathbf{r}(k)$ is the reference input vector applied to the closed-loop CS, $\mathbf{y}^{d}(k) = \mathbf{m}(\mathbf{y}^{d}(k-1),\ldots,\mathbf{y}^{d}(k-n_{\mathbf{y}m}),\mathbf{r}(k-1),\ldots,\mathbf{r}(k-n_{\mathbf{r}m}))$ is the output of the user-selected nonlinear reference model $\mathbf{m}$ of orders $\mathbf{y}m$ and $\mathbf{r}m$ accepting that the input is set as $\mathbf{r}(k)$. It is assumed that $\mathbf{m}$ is non-singular.

VRFT assumes that an I/O pair of data $\{\mathbf{u}(k),\mathbf{y}(k)\}$, $k = 0...N$, are available from the open-loop stable process. Then a virtual reference input vector $\bar{\mathbf{r}}(k)$ is calculated as $\bar{\mathbf{r}}(k) = \mathbf{m}^{-1}(\mathbf{y}(k))$, such that the reference model output and the closed-loop CS output have similar trajectories. By enforcing the notation of $\mathbf{m}^{-1}(\mathbf{y}(k))$ results in $\bar{\mathbf{r}}(k)$, which set as input to $\mathbf{m}$ and gives $\mathbf{y}(k)$. The virtual reference tracking error is then $\bar{\mathbf{e}}(k) = \bar{\mathbf{r}}(k) - \mathbf{y}(k)$. The controller which achieves $\mathbf{u}(k)$ if $\bar{\mathbf{e}}(k)$ is applied to its input is the one achieving reference model tracking. The parameters of this controller are calculated by minimizing the o.f. [19]:

$$J_{VRFT}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{k=1}^{N} \left\| \mathbf{C}_{\boldsymbol{\theta}}(\boldsymbol{\theta},\bar{\mathbf{e}}(k)) - \mathbf{u}(k) \right\|^{2}, \tag{11}$$

According to [19], in MIMO VRFT there is no need for any time-varying filter to make $J_{MR}(\boldsymbol{\theta})$ and $J_{VR}(\boldsymbol{\theta})$ approximately equal, as is usually the case in classical VRFT. The two o.f.s can be made approximately equal for a rich parameterization of the controller, which can be, for example, a neural network [19, 20]. The same nonlinear VRFT theory can be used for SISO CS design as a particular case.

4 Mixed MFAC-VRFT Control Approach

This section shows that VRFT can be used to find the parameters of MFAC algorithms. First, it will be shown that a general MFAC algorithm comprised of the estimation mechanism (7) and the control law (9) can be expressed as both a state-space nonlinear model and an I/O nonlinear recurrence. Let the nonlinear state-space model of the MFAC controller be:

$$\begin{aligned}\mathbf{u}(k) &= \mathbf{u}(k-1) + \frac{\rho\hat{\mathbf{\Phi}}^T(k)[\mathbf{y}^*(k+1) - \mathbf{y}(k)]}{\lambda + \|\hat{\mathbf{\Phi}}(k)\|^2},\\ \hat{\mathbf{\Phi}}(k) &= \hat{\mathbf{\Phi}}(k-1) + \frac{\eta[\Delta\mathbf{y}(k) - \hat{\mathbf{\Phi}}(k-1)(\mathbf{u}(k-1) - \mathbf{u}(k-2))](\mathbf{u}^T(k-1) - \mathbf{u}^T(k-2))}{\mu + \|\mathbf{u}(k-1) - \mathbf{u}(k-2)\|^2},\end{aligned} \tag{12}$$

equivalent to:

$$\begin{aligned}\mathbf{u}(k) &= \mathbf{g}(\hat{\mathbf{\Phi}}(k), \mathbf{u}(k-1), \mathbf{y}^*(k+1), \mathbf{y}(k), \boldsymbol{\theta}),\\ \hat{\mathbf{\Phi}}(k) &= \mathbf{h}(\hat{\mathbf{\Phi}}(k-1), \mathbf{u}(k-1), \mathbf{u}(k-2), \mathbf{y}(k), \mathbf{y}(k-1), \boldsymbol{\theta}),\end{aligned} \tag{13}$$

where $\mathbf{g}, \mathbf{h} \in \mathbf{R}^{2\times 1}$ are nonlinear functions of their arguments. By introducing the additional state vector $\mathbf{z}(k) = \mathbf{u}(k-1)$, it can be shown that a state-space mathematical model is in the form $\boldsymbol{\chi}(k) = \mathbf{F}(\boldsymbol{\chi}(k-1), \mathbf{U}(k), \boldsymbol{\theta})$, where the state vector is $\boldsymbol{\chi}(k) = [\mathbf{u}(k)^T \ \mathbf{z}(k)^T \ \hat{\mathbf{\Phi}}(k)^T]^T$, the input vector is $\mathbf{U}(k) = [\mathbf{y}^*(k+1)^T \ \mathbf{y}(k)^T \ \mathbf{y}(k-1)^T]^T$ and the parameter vector is $\boldsymbol{\theta} = [\rho \ \eta \ \lambda \ \mu]^T$, which is considered as an additional input vector (i.e., disturbance vector).

Using the above notations and replacing $\hat{\mathbf{\Phi}}(k)$ from the first equation in (13) with the second one, the following state-space form of the MIMO MFAC algorithm is obtained:

$$\begin{aligned}\mathbf{u}(k) &= \mathbf{g}(\hat{\mathbf{\Phi}}(k-1), \mathbf{u}(k-1), \mathbf{z}(k-1), \mathbf{y}(k), \mathbf{y}(k-1), \mathbf{y}^*(k+1), \boldsymbol{\theta}),\\ \mathbf{z}(k) &= \mathbf{u}(k-1),\\ \hat{\mathbf{\Phi}}(k) &= \mathbf{h}(\hat{\mathbf{\Phi}}(k-1), \mathbf{z}(k), \mathbf{z}(k-1), \mathbf{y}(k), \mathbf{y}(k-1), \boldsymbol{\theta}).\end{aligned} \tag{14}$$

Starting with the initial conditions $\hat{\mathbf{\Phi}}(1), \mathbf{u}(1), \mathbf{z}(1) = \mathbf{u}(0)$ applied to the nonlinear state-space model given in (14), the control input vector $\mathbf{u}(k)$ is expressed recurrently:

$$
\begin{aligned}
&\hat{\boldsymbol{\Phi}}(2) = \mathbf{h}(\hat{\boldsymbol{\Phi}}(1), \mathbf{u}(1), \mathbf{u}(0), \mathbf{y}(2), \mathbf{y}(1), \boldsymbol{\theta}),\\
&\mathbf{u}(2) = \mathbf{g}(\hat{\boldsymbol{\Phi}}(1), \mathbf{u}(1), \mathbf{u}(0), \mathbf{y}(2), \mathbf{y}(1), \mathbf{y}^*(3), \boldsymbol{\theta}),\\
&\mathbf{u}(3) = \mathbf{g}(\hat{\boldsymbol{\Phi}}(2), \mathbf{u}(2), \mathbf{u}(1), \mathbf{y}(3), \mathbf{y}(2), \mathbf{y}^*(4), \boldsymbol{\theta}) = \mathbf{g}(\mathbf{h}(\hat{\boldsymbol{\Phi}}(1), \mathbf{u}(1), \mathbf{u}(0), \mathbf{y}(2),\\
&\quad \mathbf{y}(1), \boldsymbol{\theta}), \mathbf{g}(\hat{\boldsymbol{\Phi}}(1), \mathbf{u}(1), \mathbf{u}(0), \mathbf{y}(2), \mathbf{y}(1), \mathbf{y}^*(3), \boldsymbol{\theta}), \mathbf{u}(1), \mathbf{y}(3), \mathbf{y}(2), \mathbf{y}^*(4), \boldsymbol{\theta})\\
&\quad = \mathbf{g}(\hat{\boldsymbol{\Phi}}(1), \mathbf{u}(1), \mathbf{u}(0), \mathbf{y}(3), \mathbf{y}(2), \mathbf{y}(1), \mathbf{y}^*(3), \mathbf{y}^*(4), \boldsymbol{\theta}),\\
&\ldots\\
&\mathbf{u}(k) = \mathbf{g}(\hat{\boldsymbol{\Phi}}(1), \mathbf{u}(1), \mathbf{u}(0), \mathbf{y}(k), \mathbf{y}(k-1), \ldots, \mathbf{y}(2), \mathbf{y}(1), \mathbf{y}^*(k+1), \mathbf{y}^*(k), ...,\\
&\quad \mathbf{y}^*(4), \mathbf{y}^*(3), \boldsymbol{\theta}) = \mathbf{g}(\hat{\boldsymbol{\Phi}}(1), \mathbf{u}(1), \mathbf{u}(0), \mathbf{y}^*(k+1) - \mathbf{y}(k), \mathbf{y}^*(k) - \mathbf{y}(k-1), \ldots,\\
&\quad \mathbf{y}^*(3) - \mathbf{y}(2), \mathbf{y}(1), \boldsymbol{\theta}).
\end{aligned}
\tag{15}
$$

If we denote $\mathbf{e}(k) = \mathbf{y}^*(k+1) - \mathbf{y}(k)$ then $\mathbf{u}(k)$ in (15) can be considered to emerge from an input-output nonlinear recurrent description of the form $\mathbf{u}_{\boldsymbol{\theta}_e}(k) = C_{\boldsymbol{\theta}_e}(\boldsymbol{\theta}_e, \mathbf{u}(k-1), \ldots, \mathbf{u}(k-n_{uc}), \mathbf{e}(k), \ldots, \mathbf{e}(k-n_{ec}))$, with $\boldsymbol{\theta}_e = \{\hat{\boldsymbol{\Phi}}(1), \boldsymbol{\theta}^T\}$. If $\mathbf{r}(k)$ specific to VRFT is considered equivalent to $\mathbf{y}^*(k+1)$ in MFAC, then the MFAC controller structure can be considered in a closed-loop CS. Figure 1 shows the CS structure with MFAC-VRFT algorithm.

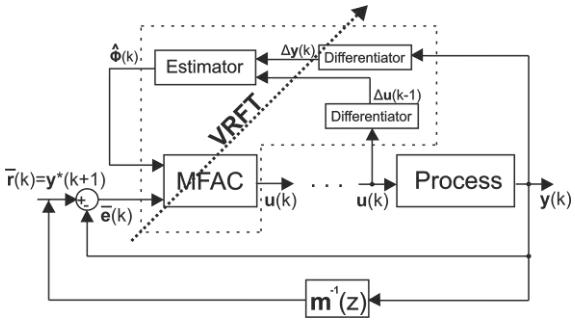


Figure 1
CS structure with mixed MFAC-VRFT algorithm [7]

Choosing the reference model $\mathbf{m} = \mathbf{I}_2$ in the nonlinear VRFT design is equivalent to trying to minimize $J_{MR}(\boldsymbol{\theta}) = \sum_{k=1}^{N} \left\| \mathbf{y}_{\boldsymbol{\theta}}(k) - \mathbf{y}^*(k) \right\|^2$, which is the batch-wise version of the adaptive one-step ahead MFAC o.f. J_{MFAC} in (5) with $\lambda = 0$. However, no causal controller that can achieve $\mathbf{m} = \mathbf{I}_2$ exists in practice. Choosing therefore $\mathbf{m} \neq \mathbf{I}_2$ in VRFT is equivalent to $\lambda \neq 0$ in MFAC. The parameter λ is crucial since it impacts the MFAC stability in the sense that an increased λ improves stability which simply means adding more weight to the control input increment. In terms of VRFT, this means choosing a reference model $\mathbf{m}$ with lower

bandwidth which increases the CS response time but also increases the overall CS robustness. Therefore, the proposed mixed MFAC-VRFT control approach translates the design of MFAC algorithm parameters (such as $\hat{\mathbf{\Phi}}(1)$ and $\boldsymbol{\theta} = [\rho \ \eta \ \lambda \ \mu]^T$) into easier to comprehend closed-loop CS characteristics described by the reference model $\mathbf{m}$.

5 SISO Experimental Validation

Two tuning strategies are described in this section in order to validate the mixed MFAC-VRFT control approach:

- an indirect one, in which the VRFT framework is used and the o.f. in (11) is minimized, this is the mixed MFAC-VRFT control approach
- a direct tuning approach, in which a process model is used and a metaheuristics Gravitational Search Algorithm (GSA) optimizer [21]-[23] is used to minimize the o.f.s:

$$\begin{aligned} \boldsymbol{\tau}^* &= \arg\min_{\boldsymbol{\tau}} J(\boldsymbol{\tau}), \ J_{\varepsilon}^{a}(\boldsymbol{\tau}) = \frac{1}{N}\sum_{k=1}^{N}((\mathbf{y}_1^*(k,\boldsymbol{\tau}) - \mathbf{y}_1(k,\boldsymbol{\tau}))^2), \\ \boldsymbol{\tau}^* &= \arg\min_{\boldsymbol{\tau}} J(\boldsymbol{\tau}), \ J_{\varepsilon}^{p}(\boldsymbol{\tau}) = \frac{1}{N}\sum_{k=1}^{N}((\mathbf{y}_2^*(k,\boldsymbol{\tau}) - \mathbf{y}_2(k,\boldsymbol{\tau}))^2), \end{aligned} \tag{16}$$

where $\boldsymbol{\tau}^*$ is the optimal parameter vector of VRFT-MFAC and MFAC algorithms, the expression of the parameter vector is $\boldsymbol{\tau} = [\hat{\Phi}_a(1) \ \rho_a \ \eta_a \ \lambda_a \ \mu_a]^T$ for azimuth control and $\boldsymbol{\tau} = [\hat{\Phi}_p(1) \ \rho_p \ \eta_p \ \lambda_p \ \mu_p]^T$ for pitch control, a indicates the azimuth control and p indicates the pitch control. Other optimization problems with adequate o.f.s used as performance indices can be used as well [24]-[26].

The bounds in (6) are set as $b_2 = \hat{\Phi}(1)/2$ and ab_2, where $a = 3$. This section will investigate if the performance of CS with mixed VRFT-MFAC algorithm is similar to the performance of CS with MFAC algorithm. The CS performance is assessed through ten experimental trials of the o.f.s J_{ε}^{a} and J_{ε}^{p}. The averages and variances of these o.f.s. are next taken for the sake of improved measurement of CS and algorithm performance to avoid random disturbances.

The experiments have shown that the performance of CS with mixed VRFT-MFAC algorithm depends on the initial signals applied to the open-loop experiment and also on the reference model $\mathbf{m}$, which, according to [1]-[4] must be chosen such that the closed-loop CS signal should be capable to track the reference model.

Extensive work shows that the choice of the reference model ensuring an overall stable CS is rather restrictive.

The MFAC algosithms are designed using the transfer function matrix:

$$m_a(z) = \frac{0.00079z^{-1} + 0.00079z^{-2}}{1 - 1.981z^{-1} + 0.982z^{-2}} \tag{17}$$

for azimuth control, and:

$$m_p(z) = \frac{0.00176z^{-1} + 0.00172z^{-2}}{1 - 1.938z^{-1} + 0.941z^{-2}} \tag{18}$$

for pitch control. VRFT is next applied to compute the controllers initial parameters using a GSA that minimizes the o.f. in (11). These parameters are: $\hat{\Phi}(1) = 513$, bounded by $\hat{\Phi}(1) \in (256.5,\ 769.5)$, $\rho = 7$, $\eta = 0.0076$, $\lambda = 704$, and $\mu = 993.05$ for azimuth control and $\hat{\Phi}(1) = 4.22$, bounded by $\hat{\Phi}(1) \in (2.11,\ 6.34)$, $\rho = 0.18$, $\eta = 0.0039$, $\lambda = 4.43$, and $\mu = 999.85$ for pitch control.

The initial parameters of the MFAC algorithms obtained by a GSA that minimizes the o.f. in (17) are: $\hat{\Phi}(1) = 110$, bounded by $\hat{\Phi}(1) \in (55,\ 165)$, $\rho = 1.55$, $\eta = 0.1$, $\lambda = 3.65$, and $\mu = 0.89$ for azimuth control, and $\hat{\Phi}(1) = 160$, bounded by $\hat{\Phi}(1) \in (80,\ 240)$, $\rho = 5.35$, $\eta = 0.31$, $\lambda = 6.21$, and $\mu = 0.54$ for pitch control.

Table 1 gives the averages and the variances of J_ε. The CS responses as control inputs and controlled outputs versus time are presented in Figure 2 for the azimuth SISO control loop and in Figure 3 for the pitch SISO control loop. Figures 2 and 3 also illustrate the tracking reference inputs, which can be slightly different for other applications [27]-[31].

Table 1
The values of the o.f.s

	Mixed VRFT-MFAC	MFAC
Average of J_ε^a	0.004	0.0036
Variance of J_ε^a	$1.8990 \cdot 10^{-7}$	$6.5343 \cdot 10^{-7}$
Average of J_ε^p	0.0034	0.0036
Variance of J_ε^p	$1.9739 \cdot 10^{-8}$	$4.5406 \cdot 10^{-9}$

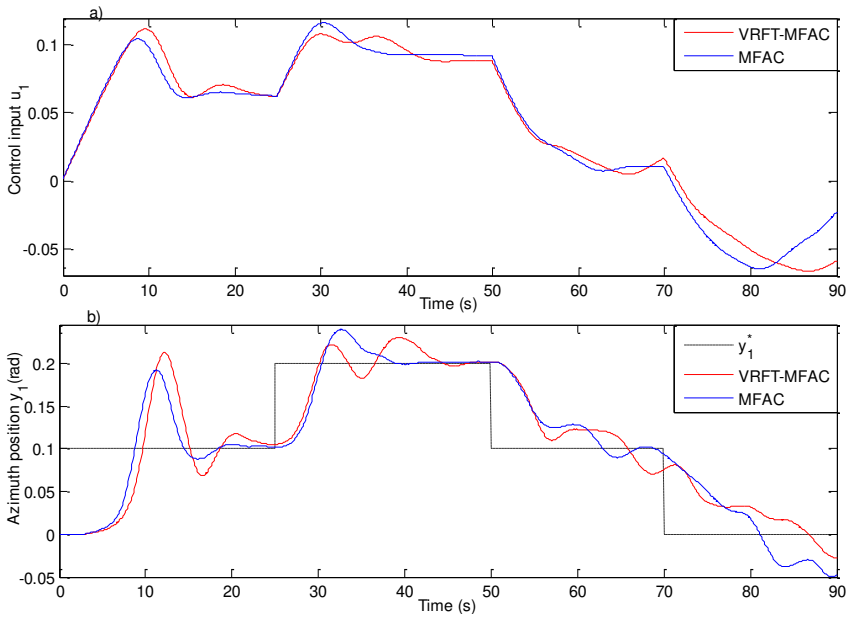


Figure 2
Experimental results related to SISO azimuth control: a) u_1 versus time, b) y_1 and y_1^* versus time

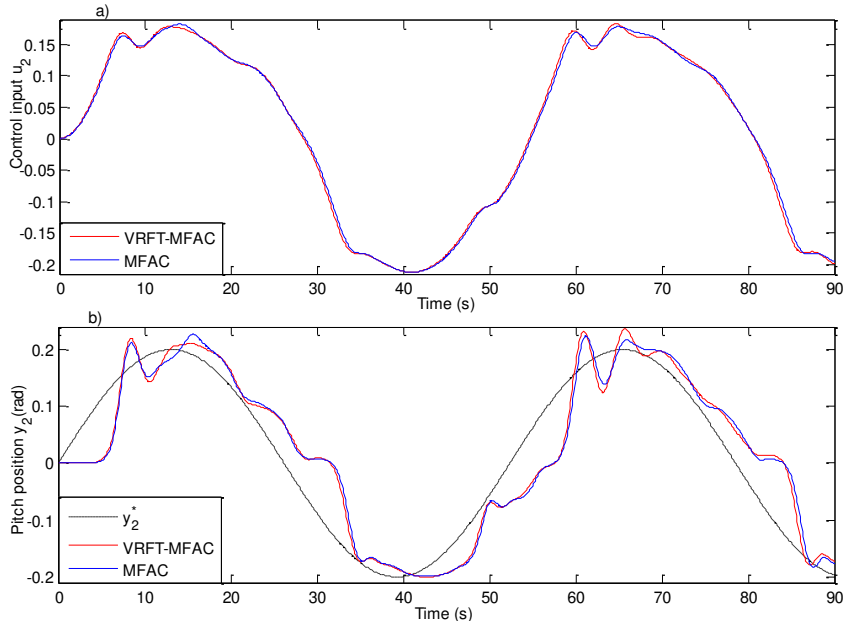


Figure 3
Experimental results related to SISO azimuth control: a) u_2 versus time, b) y_2 and y_2^* versus time

Conclusions

This paper has proposed the combination of two data-driven control approaches, which has led to the formulation of a mixed MFAC-VFRT control approach. This approach leads to mixed MFAC-VFRT algorithms that are actually MFAC algorithms tuned by VRFT.

The experimental results presented in Section 5 outline that the differences of the o.f.s. from Table 2 are insignificant for both azimuth and pitch SISO control. Figures 2 and 3 show that the control inputs and the controlled outputs almost overlap. Therefore, the mixed MFAC-VFRT control approach is a time saving solution that finds the controller optimal parameters and offers similar CS performance with that of CS with MFAC algorithm, whose initial parameters were obtained using GSA. The mixed MFAC-VFRT control approach is useful for processes whose identification is difficult or impossible.

Further research will treat the study of several constraints concerning the choice of the reference model and performance improvement, which can be achieved by the combination of artificial intelligence techniques (including fuzzy control) and neural networks [32]-[39].

Acknowledgement

This work was supported by grants of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project numbers PN-II-RU-TE-2014-4-0207 and PN-II-ID-PCE-2011-3-0109, and from the NSERC of Canada.

References

[1] M. C. Campi, A. Lecchini, S. M. Savaresi: Virtual Reference Feedback Tuning: A Direct Method for the Design of Feedback Controllers, Automatica, Vol. 38, No. 8, 2002, pp. 1337-1346

[2] S. Formentin, S. M. Savaresi, L. Del Re: Non-Iterative Direct Data-Driven Tuning for Multivariable Systems: Theory and Application, IET Control Theory & Applications, Vol. 6, No. 9, 2012, pp. 1250-1257

[3] M. C. Campi, S. M. Savaresi: Virtual Reference Feedback Tuning for Non-Linear Systems, Proceedings of 44th IEEE Conference on Decision and Control European Control Conference CDC-ECC '05, Seville, Spain, 2005, pp. 6608-6613

[4] M. C. Campi, S. M. Savaresi: Direct Nonlinear Control Design: The Virtual Reference Feedback Tuning (VRFT) Approach, IEEE Transactions on Automatic Control, Vol. 51, No. 1, 2006, pp. 14-27

[5] Z.-S. Hou, S. Jin: Data-Driven Model-Free Adaptive Control for a Class of MIMO Nonlinear Discrete-Time Systems, IEEE Transactions on Neural Networks, Vol. 22, No. 12, 2011, pp. 2173-2188

[6] Z.-S. Hou, S. Jin: A Novel Data-Driven Control Approach for a Class of Discrete-Time Nonlinear Systems, IEEE Transactions on Control Systems Technology, Vol. 19, No. 6, 2011, pp. 1549-1558

[7] R.-C. Roman, M.-B. Radac, R.-E. Precup, E. M. Petriu: Virtual Reference Feedback Tuning of MIMO Data-driven Model-Free Adaptive Control Algorithms, Proceedings of 7th Advanced Doctoral Conference on Computing, Electrical and Industrial Systems DoCEIS'16, Caparica (Lisbon), Portugal, 2016, pp. 1-8

[8] R. A. Lewis, A. K. Bejczy: Planning Considerations for a Roving Robot with Arm, Proceedings of 3rd International Joint Conference on Artificial Intelligence IJCAI 1973, Standford, CA, USA, 1973, pp. 308-316

[9] T. J. Tarn, A. K. Bejczy, S. Ganguly: Feedback Control of Robot Arm with Discrete Nonlinear Model, Journal of Field Robotics, Vol. 9, No. 4, 1992, pp. 529-559

[10] A. K. Ramadorai, T. J. Tarn, A. K. Bejczy, N. Xi: Task-driven Control of Multi-Arm systems, IEEE Transactions on Control Systems Technology, Vol. 2, No. 3, 1994, pp. 198-206

[11] R.-E. Precup, S. Preitl, M. Balas, V. Balas: Fuzzy Controllers for Tire Slip Control in Anti-Lock Braking Systems, Proceedings of IEEE International Conference on Fuzzy Systems FUZZ-IEEE 2004, Budapest, Hungary, 2004, Vol. 3, pp. 1317-1322

[12] R.-E. Precup, C.-A. Dragos, S. Preitl, M.-B. Radac, E. M. Petriu: Novel Tensor Product Models for Automatic Transmission System Control, IEEE Systems Journal, Vol. 6, No. 3, 2012, pp. 488-498

[13] R.-C. Roman, M.-B. Radac, R.-E. Precup, A.-I. Stinean: Two Data-driven Control Algorithms for a MIMO Aerodynamic System with Experimental Validation, Proceedings of 19th International Conference on System Theory, Control and Computing Joint Conference ICSTCC 2015, Cheile Gradistei, Romania, 2015, pp. 736-741

[14] A. S. Bazanella, L. Campestrini, D. Eckhard: Data-driven Controller Design: The H_2 Approach, Springer-Verlag, Berlin, Heidelberg, 2012

[15] Z.-S. Hou, Z. Wang: From Model-Based Control to Data-driven Control: Survey, Classification and Perspective, Information Sciences, Vol. 235, 2013, pp. 3-35

[16] Two Rotor Aerodynamical System, User's Manual, Inteco Ltd., Krakow, Poland, 2007

[17] M.-B. Radac, R.-C. Roman, R.-E. Precup, E. M. Petriu: Data-driven Model-Free Control of Twin Rotor Aerodynamic Systems: Algorithms and Experiments, Proceedings of 2014 IEEE International Symposium on

Intelligent Control ISIC 2014 as part of IEEE Multi-conference on Systems and Control MSC 2014, Antibes, France, 2014, pp. 1889-1894

[18] R.-C. Roman, M.-B. Radac, R.-E. Precup: Data-driven Model-Free Adaptive Control of Twin Rotor Aerodynamic Systems, Proceedings of IEEE 9th International Symposium on Applied Computational Intelligence and Informatics SACI 2014, Timisoara, Romania, 2014, pp. 25-30

[19] P. Yan, D. Liu, D. Wang, H. Ma: Data-driven Controller Design for General MIMO Nonlinear Systems via Virtual Reference Feedback Tuning and Neural Networks, Neurocomputing, 2015, Vol. 171, 2016, pp. 815-825

[20] A. Esparza, A. Sala, P. Albertos: Neural Networks in Virtual Reference Tuning, Engineering Applications of Artificial Intelligence, Vol. 24, No. 6, 2011, pp. 983-995

[21] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, A. S. Paul: Gravitational Search Algorithm-based Tuning of Fuzzy Control Systems with a Reduced Parametric Sensitivity, in: Soft Computing in Industrial Applications, A. Gaspar-Cunha, R. Takahashi, G. Schaefer, L. Costa, Eds., Springer-Verlag, Berlin, Heidelberg, Advances in Intelligent and Soft Computing, Vol. 96, 2011, pp. 141-150

[22] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, M.-B. Radac: Gravitational Search Algorithms in Fuzzy Control Systems Tuning, Proceedings of 18th IFAC World Congress, Milano, Italy, 2011, pp. 13624-13629

[23] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, M.-B. Radac: Fuzzy Logic-based Adaptive Gravitational Search Algorithm for Optimal Tuning of Fuzzy Controlled Servo Systems, IET Control Theory & Applications, Vol. 7, No. 1, 2013, pp. 99-107

[24] L. Horváth, I. J. Rudas: Active Knowledge for the Situation-Driven Control of Product Definition, Acta Polytechnica Hungarica, Vol. 10, No. 2, 2013, pp. 217-234

[25] R.-E. Precup, M. L. Tomescu, M.-B. Radac, E. M. Petriu, S. Preitl, C.-A. Dragos: Iterative Performance Improvement of Fuzzy Control Systems for Three Tank Systems, Expert Systems with Applications, Vol. 39, No. 9, 2012, pp. 8288-8299

[26] E. Osaba, E. Onieva, F. Dia, R. Carballedo, P. Lopez, A. Perallos: A Migration Strategy for Distributed Evolutionary Algorithms Based on Stopping Non-Promising Subpopulations: A Case Study on Routing Problems, International Journal of Artificial Intelligence, Vol. 13, No. 2, 2015, pp. 46-56

[27] F. G. Filip: Decision Support and Control for Large-Scale Complex Systems, Annual Reviews in Control, Vol. 32, No. 1, 2008, pp. 61-70

[28] M. Bošnak, D. Matko, S. Blažič: Quadrocopter Control Using an On-Board Video System with Off-Board Processing, Robotics and Autonomous Systems, Vol. 60, No. 4, 2012, pp. 657-667

[29] X. Zhao, S. Yin, H. Li, B. Niu: Switching Stabilization for a Class of Slowly Switched Systems, IEEE Transactions on Automatic Control, Vol. 60, No. 1, 2015, pp. 221-226

[30] H.-N. L. Teodorescu: A Retrospective Assessment of Fuzzy Logic Applications in Voice Communications and Speech Analytics, International Journal of Computers, Communications & Control, Vol. 10, No. 6, 2015, pp. 865-872

[31] P. Moallem, B. S. Mousavi, S. Sh Naghibzadeh: Fuzzy Inference System Optimized by Genetic Algorithm for Robust Face and Pose Detection, International Journal of Artificial Intelligence, Vol. 13, No. 2, 2015, pp. 73-88

[32] S. Preitl, R.-E. Precup: Introducere in Conducerea Fuzzy a Proceselor, Editura Tehnica, Bucharest, 1997

[33] Zs. Cs. Johanyák, Sz. Kovács: Distance-based Similarity Measures of Fuzzy Sets, Proceedings of 3rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence SAMI 2005, Herl'any, Slovakia, 2005, pp. 265-276

[34] R.-E. Precup, M. L. Tomescu, S. Preitl: Fuzzy Logic Control System Stability Analysis Based on Lyapunov's Direct Method, International Journal of Computers, Communications & Control, Vol. 4, No. 4, 2009, pp. 415-426

[35] R.-E. Precup, S. Preitl, M.-B. Radac, E. M. Petriu, C.-A. Dragos, J. K. Tar: Experiment-based Teaching in Advanced Control Engineering, IEEE Transactions on Education, Vol. 54, No. 3, 2011, pp. 345-355

[36] J. Vaščák: Adaptation of Fuzzy Cognitive Maps by Migration Algorithms, Kybernetes, Vol. 41, No. 3-4, 2012, pp. 429-443

[37] R.-C. David, R.-E. Precup, E. M. Petriu, M.-B. Radac, S. Preitl: Gravitational Search Algorithm-based Design of Fuzzy Control Systems with a Reduced Parametric Sensitivity, Information Sciences, Vol. 247, 2013, pp. 154-173

[38] G. Kaur, J. Dhar, R. K. Guha: Stock Market Forecasting Using ANFIS with OWA Operator, International Journal of Artificial Intelligence, Vol. 12, No. 2, 2015, pp. 102-114

[39] B. Takarics, P. Baranyi: Friction Compensation in TP Model Form – Aeroelastic Wing as an Example System, Acta Polytechnica Hungarica, Vol. 12, No. 4, 2015, pp. 127-145

Contradiction Resolution in the Adaptive Control of Underactuated Mechanical Systems Evading the Framework of Optimal Controllers

József K. Tar[†], János F. Bitó[†], Imre J. Rudas[‡]

[†]Antal Bejczy Center for Intelligent Robotics, [‡]Research and Innovation Center,
[†,‡]John von Neumann Faculty of Informatics, Institute of Applied Mathematics,
Óbuda University, H-1034 Budapest, Bécsi út 96/b, Hungary
E-mail: {tar.jozsef@nik., bito@, rudas@}uni-obuda.hu

Abstract: In the practice, precise and efficient control is needed for certain state variables of multiple variable physical systems in which the number of the independent control variables is less than that of the independent state variables. In such cases, either the propagation of certain state variables is completely abandoned or the concept of the Model Predictive Control (MPC) *is applied in which the model of the controlled system is embedded into the mathematical framework of the* Optimal Controllers*. This approach uses a* cost function *that summarizes the contributions of the frequently contradictory requirements. By minimizing this cost a kind of "compromise" is achieved. Whenever approximate and/or incomplete system models are available, the use of this controller is justified only for short time-intervals. The only way to reduce the accumulation of the effects of the modeling errors is the frequent re-design of the time horizon from the actual state as initial state that is done by the* Receding Horizon Controllers*. The more sophisticated* Adaptive Controllers *are designed by the use of Lyapunov's "Direct Method" that has a complicated mathematical framework that cannot easily be combined with that of the optimal controllers. As a potential competitor of the Lyapunov function-based adaptive controllers a* Fixed Point Transformation*-based approach was invented that in the first step transforms the the problem of computing the control signal into the task of finding an appropriate fixed point of a contractive map. The fixed point can be found by iteration in which the iterative sequence is generated by this contracting map. This method can be used for contradiction resolution without the minimization of any cost function by tracking the observable state components with time-sharing on a rotary basis.*

In the present paper a novel fixed point transformation is introduced. It is shown that this construction for monotonic response function of bounded derivative can guarantee global stability. Furthermore, the time-sharing-based method is demonstrated by the control of an underactuated 3 DoF Classical Mechanical system via numerical simulations.

Keywords: adaptive control; underactuated mechanical systems; fixed point transformations; optimal control; contradiction resolution; time-sharing;

1 Introduction

In practice, precise and efficient control is needed for certain state variables of multiple variable physical systems in which the number of the independent control variables is less than that of the independent state variables.

Typical examples are the *underactuated Classical Mechanical systems*, such as the *Translational Oscillations with an Eccentric Rotational Proof Mass Actuator (TORA)* that is a simplified model of a dual-spin spacecraft with mass imbalance. It serves as a "benchmark paradigm" for controller designers (e.g. [1]) for the control of which various controllers can be developed as a cascade and a passivity based controller in [2], a model-based controller using the *Tensor Product Form (TP)* in [3]. In [4] nine papers were published on the control of the TORA system in a special issue.

From the subject area of physiology, the illness called *Type 1 Diabetes Mellitus (T1DM)*, has various, more or less complex models. *Bergman's Minimal Model* as presented by Chee et Fernando in [5] has only three state variables. More complex models of the same phenomenon take into consideration more variables (e.g. [6]) that can be combined with digestion models as e.g. that in [7] work with 10 state variables. However, the only measurable variable is the blood glucose concentration while we have only one control signal, namely, the insulin ingress rate.

Another interesting area from the realm of nonlinear phenomena is the operation of the neurons. From the beginning of the 20th century various efforts were made to expound the spiking property of the neurons. From Lapicque's *"Integrate and Fire Neuron Model"* in 1907 to the quite sophisticated Hodgkin-Huxley model in 1952 [8] distinguishing between sodium, potassium and leakage channels. Various simplifications were also used. As examples the *Chua-Matsumoto Circuit* in 1984 [9] or the FitzHugh-Nagumo model in 1961 [10] can be mentioned. Each of these models is a multivariable system having nonlinear dynamic coupling between its variables to which only one control signal is available.

In a wider sense the above examples well represent the "underactuated systems". These systems have the important feature that makes it physically impossible to drive them through an arbitrary "trajectory" along which each state variable's precise position is prescribed in time. Controlling only one state variable and letting the other ones propagate as they want generally cannot be an acceptable option. It is more expedient to somehow "distribute" the tracking error over the various state variables that evidently may be a contradictory task. A plausible solution for contradiction resolution is the minimization of a cost functional that is constructed as a sum of the errors to be kept at bay as well as some other terms that express some limitations of the controllers or other extra requirements. In general this problem appears as the Hamilton-Jacobi-Bellman equation that can be solved by *Dynamic Programming* that consumes up a lot of computational power [11]. It uses the principle of optimality of subproblems and applies tabulation in the state space to compute recursively a feedback control. The *Indirect Methods* are related to the variational principles of Classical Mechanics with the introduction of the *Lagrange Multipliers*

as "co-state" variables. The direct methods transform the infinite optimal control problem into a finite dimensional *Nonlinear Programming Problem (NLP)*. Simply treatable problem is obtained only if the goal functional has very special form as e.g. in the case of the *Linear Quadratic Regulator* [12, 13] in which, the controlled system is *Linear Time Invariant (LTI)* and the cost function has quadratic structure. This makes the problem tractable by using Riccati differential or algebraic equations, depending on the role of the terminal conditions. The simplicity means that a separate solution becomes available for the co-state and the state variables. In more general cases the possibility for this separation ceases and *state-dependent Riccati equations* appear (e.g. [14]). These approaches normally are based on the globally linearizable form and apply state-dependent weighting matrices in the LQR form.

Whenever the available models are not precise enough, the time-horizon for the controller design cannot be too long. To evade the accumulation of the effects of the modeling errors the so calculated control signal can be applied only for a short time-horizon, the new initial conditions have to be measured and a redesign has to be initiated for the next short period (the *Receding Horizon Control* that appeared in the late seventies of the past century in relation with industrial applications [15]).

An alternative error-compensation possibility is the creation of an *Adaptive Controller*. The adaptive controllers traditionally are designed by the use of Lyapunov's "Direct Method" that he elaborated in his PhD dissertation in 1892 [16, 17] when he investigated the stability of motion. The main idea was that in spite of the fact that normally, the solution of coupled nonlinear differential equations cannot be expressed in closed analytical form, without knowing the details of the motion it became possible to determine its stability. The *Adaptive Inverse Dynamics Controller (AIDC)* and the *Adaptive Slotine-Li Controller (ASLC)* for robots in the nineties were developed by the use of this technique [18]. The method seems to be prevailing nowadays, in the design of the *Model Reference Adaptive Controllers (MRAC)*, too (e.g. [19, 20, 21]). Normally, this method can guarantee *global stability* without revealing any details of the controlled motion. In general whole set of adaptive control parameters can result in global stability with different "transients" of the controlled motion. Whenever these details are important the adaptive control parameters can be tuned to improve the transient behavior e.g. by evolutionary methods (e.g. [22, 23]). The application of Lyapunov function in adaptive controllers need very skilled control designers since for each particular control task one has to construct an individual Lyapunov function and has to prove the non-positivity of its time-derivative. One source of the mathematical difficulties may be that this method is based on *satisfactory conditions* and it may prescribe much more restrictions, than necessary. Another problem is that this approach does not seem to be easily integrable with the mathematical framework of the optimal controllers.

Another practical approach that can evade the complexity of the Lyapunov function based design is the use of *Robust Variable Structure/Sliding Mode Controllers (VS/SM)*. The very simple idea originates from the Soviet Union in the 1960s and became known to the western world later (e.g. [24, 25, 26]). Its main point was that in the possession of an approximate model only, and under the influence of unknown external disturbances, for a kinematically prescribed trajectory tracking, it is

impossible to calculate the appropriate control signals. Instead of that the concept of the *error metrics* was so introduced that driving it to and keeping it in the vicinity of zero make the trajectory tracking error converge to zero. The great advantage of this solution is its simplicity and is easily realizable. Its drawback is that the control signal can sharply vary, when the components of the error metrics cross the value zero. The so generated *chattering* can excite "not modeled" subsystems unexpectedly. Normally this effect can be eliminated by "softening" the switching rule that may reduce the precision of the trajectory tracking.

With the aim of evading the difficulties of the Lyapunov function-based techniques and maintaining the simplicity of the VS/SM controllers without their aptitude for chattering, a novel adaptive controller-design methodology was suggested in 2009 [27, 28], that directly tries to realize a purely kinematically prescribed trajectory tracking property of the controlled motion by studying the response of the controlled system in the given control situation. This approach, at first, converts the control task into a *fixed point problem* that iteratively can be solved afterwards. The use of this idea goes back to the 17th century (the *"Newton-Raphson Algorithm"* [29]) and obtains great attention even in our days (e.g. [30, 31]). In 1922 Banach generalized this fixed point method to quite abstract linear, normed, complete metric spaces [32]. This iteration-based approach is essentially different to the method of *Iterative Learning Control (ILC)* that was elaborated for robots repetitively executing the same task (e.g. [33, 34, 35, 36, 37]). The original transformation introduced in [27] was called *Robust Fixed Point Transformation (RFPT)* that contained only three adaptive control parameters. While in several control applications, fix settings of these parameters was found to be satisfactory, for other cases, complementary tuning strategies were elaborated, for tuning only, its one adaptive parameters [38, 39, 40]. Attempts were also made to modify the fixed point transformation used for transforming the control task into a fixed point problem [41, 42, 43].

The method was found to be appropriate for various control tasks, via simulations, as e.g. chemical reactions [44, 45], the FitzHugh-Nagumo neuron model [46], the Hodgkin Huxley neuron [47], Chua-Matsumoto circuit [48], and various diabetes models [49, 50].

In [46] and [48] the idea of replacing the cost-functional-based optimal control approach for the resolution of the contradictions regarding the prescriptions for the tracking precision of the various state-variables of *underactuated systems* by time-sharing on a rotary basis already arose. The aim of the present paper is to show that this idea can work in the case of an underactuated Classical Mechanical system, the TORA model. In the simulations an improved version of the fixed point transformation suggested in [43] was used.

2 The Fixed Point Transformation-based Approach

This approach assumes that we have the *approximate model* of the dynamic system to be controlled. The control actions are calculated with its help based on some kinematically expressed trajectory error reduction by comparing the *desired*

response r^{Des} and the *actually observed response* r^{Act} of the controlled system. Due to modeling imprecisions and unknown external disturbances normally $r^{Act} \neq r^{Des}$. For the given control situation a *"response function"* can be observed that sets a relationship between the desired and the observed responses as $r^{Act} = f\left(r^{Des}\right)$. The core of the fixed point transformation-based technique is the deformation of the input of the response function from r^{Des} to $r_\star$ using *Banach's Fixed Point Theorem* [32] in order to achieve the situation $r^{Act} \equiv f(r_\star) = r^{Des}$.

2.1 Antecedents

For the purpose of obtaining the appropriate deformation in [27] for SISO systems the transformation function was introduced as

$$r_{n+1} = (r_n + K_c)\left[1 + B_c \tanh\left(A_c\left\{f(r_n) - r^{Des}\right\}\right)\right] - K_c \tag{1}$$

with $K_c, B_c, A_c \in \mathbb{R}$ parameters. This construction had two fixed points for a monotonic response function $f(r)$: a trivial one at $r = -K_c$, and the solution of the control task $r = r_\star$. In [40] it was shown that for monotonic $f(r)$ and fixed K_c and B_c as $|A_c|$ was increased from zero the fixed point at $r = -K_c$ always behaved as a *monotonic repulsive* one, while the other one $r = r_\star$ at first was *monotonic attractive*, then it turned into *oscillatory attractive* before turning to an *oscillatory repulsive*. The session of the monotonic attractive behavior did not risk the stability of the controller, it was observed by *model-independent observers* and was used for tuning A_c in order to avoid the occurrence of the regime of bounded chaotic oscillations. On this reason these oscillations were called *Precursor Oscillations* in [40, 51]. The chaotic behavior was studied in [52, 53, 51] and it was shown that it was generated by the co-operation of two repulsive fixed points. It was found that in general it does not risk the precision of the trajectory tracking, and its great chattering can be reduced.

In [43] the *Sigmoid Generated Fixed Point Transformation* was suggested that was constructed of a monotonic increasing, bounded and smooth $g(x) : \mathbb{R} \mapsto \mathbb{R}$ *sigmoid* function. For some $K > 0$ and $D > 0$ positive parameters the iterative sequence $\{x_0, \ldots, g(x_n) - K = g(x_{n+1} - D), \ldots\}$ was generated that lead to the function $F(x) \stackrel{def}{=} g^{-1}(g(x) - K) + D$ where the inverse function of $g()$ is denoted by $g^{-1}()$. The *fixed point* of $F(x)$ is the solution of the equation $F(x_*) = x_*$. This function was used in [43] for the generation of the sequence of the *deformed inputs* as

$$r_{n+1} = G(r_n) \stackrel{def}{=} F\left(A_c\left[f(r_n) - r^{Des}\right] + x_*\right) + r_n - x_* \tag{2}$$

where $A_c \in \mathbb{R}$ stands for a parameter, and normally, in the control applications $r_0 \stackrel{def}{=} r_0^{Des}$, that is the *desired response* in the initial control cycle. Obviously, if $r_\star$ is the solution of the control task, i.e. $f(r_\star) - r^{Des} = 0$ then $G(r_\star) = r_\star$. Since $F(x_*) = x_*$, this solution is a fixed point of the function G. In order to guarantee the convergence of the series $\{r_n\}$ function G must be contractive, i.e. the relation

$\left|\frac{dG}{dr}\right| < 1$ must be valid. It was shown that this construction can have two fixed points as the originally suggested RFPT has, can produce the *precursor oscillations*, and can be used for adaptive control purposes.

However, this construction may have difficulties. Both constructions worked only with *bounded region of attraction* around $r_\star$ that formally cannot guarantee *global stability*. It is clear that the graph of the original bounded function $g(x)$ was *shifted down* in $g(x)-K$, and it was shifted to the right by $g(x-D)$. It was assumed that for the first element of the iteration x_0 there exists x_1 for which $g(x_0)-K = g(x_1-D)$. This may be possible for several x_0 values but not necessarily for each $x_0 \in \mathbb{R}$. For instance, if $g(x) \stackrel{def}{=} \tanh(x)$, $g(x)-K \in (-1-K, 1-K)$ and $g(x-D) \in (-1,1)$, therefore there exist x_0 to which no x_1 belongs. In the present paper this deficiency, is eliminated, by the introduction of a transformation generation technique, using *"Stretched Sigmoid Functions". It also will be shown that for SISO systems, of monotonic response functions, with bounded derivatives, this construction, can guarantee global stability*.

2.2 Stretched Sigmoids Generated Fixed Point Transformation (SSGFPT)

This fixed point transformation is generated by sigmoid functions $g(x)$, and $h(x)$ as follows:

$$g(x) \stackrel{def}{=} \tanh(x) - K_c \ , \tag{3a}$$

$$h(x) \stackrel{def}{=} \left(1 + \frac{K_c + J_c}{2}\right) \tanh\left(\frac{x - D_c}{1 + \frac{K_c + J_c}{2}}\right) - \frac{K_c + J_c}{2} \ , \tag{3b}$$

$$g(x_0) = h(x_1), \ldots, g(x_n) = h(x_{n+1}), \ldots \ , \tag{3c}$$

$$x_{n+1} = h^{-1}(g(x_n)) \equiv F(x_n) \ , \tag{3d}$$

in which $K_c, J_c,$ and $D_c > 0$, and $h^{-1}(x)$ denotes the inverse of the monotonic function $h(x)$. For the parameter settings $K_c = 0.5$, $J_c = 0.2$, and $D_c = 0.6$ this iteration is exemplified by Figs. 1, 2.

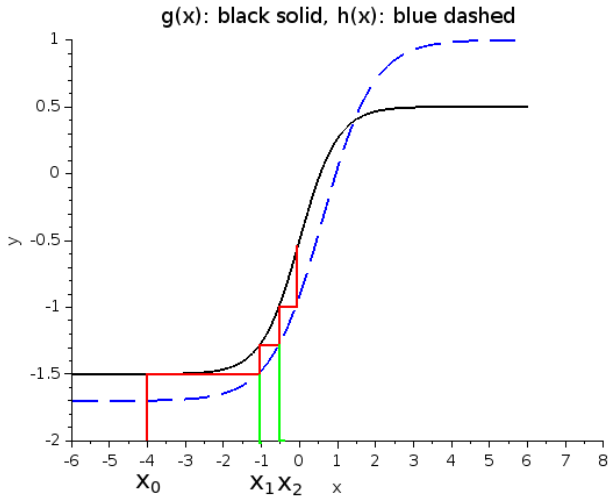


Figure 1
Generation of the fixed point transformation by the functions $g(x)$ and $h(x)$ in (3)

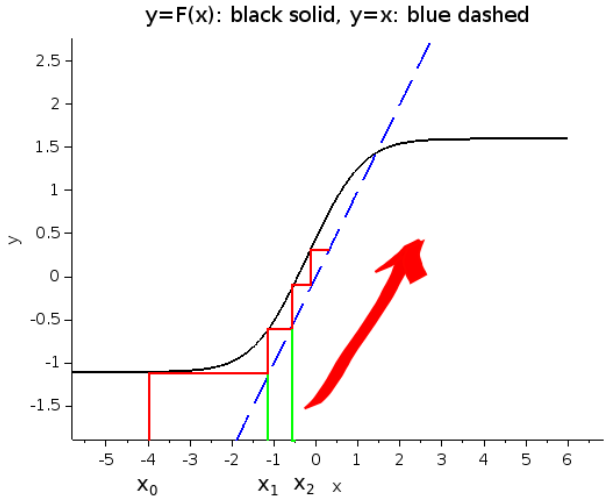


Figure 2
Generation of the fixed point transformation by the function $F(x)$ in (3)

Since $\forall x \in \mathbb{R}$ $g(x) \in (-1-K_c, 1-K_c)$, and $h(x) \in (-1-K_c-J_c, 1)$, the iteration defined in (3) always converges to $x_\star$.

The here suggested solution has considerable advantages in comparison with the RFPT transformation published in [27] or the new variants suggested in [42, 43]:

if the response function of the controlled system is monotonic and $\left|\frac{\mathrm{d}f}{\mathrm{d}r}\right|$ is bounded the function in (2) has only a single attractive fixed point over the whole real axis $\mathbb{R}$. Furthermore, due to the saturation of $g(x)$ and $h(x)$ it produces very fast convergence if the distance $|x-x_*|$ is great, and provides acceptable convergence speed in the vicinity of x_*. Due to the strongly saturated nature of $g(x)$ in (2) relatively great value for the parameter A_c can be used that also speeds up the convergence in the "beaked" structure near x_*. In the sequel the application of this fixed point transformation is exemplified in the case of an underactuated, extended TORA model.

3 Adaptive Optimal Control based on Time Sharing and SSGFPT

In this section the application of SSGFPT will be exemplified by the use of a TORA variant we already considered in [40, 43, 42]. At first the dynamic model is explained.

3.1 The Dynamic Model of the TORA System

The model consists of a cart, a pendulum (practically a beam) and a dial that can be rotated around an axle attached to the end of the beam. Its equations of motion under full actuation are are given in (4) with the dynamic parameters defined in Table 1. The generalized coordinates of the system are $q_1\,[rad]$ that describes the rotation of the beam with respect to the vertical direction, $q_2\,[rad]$ that is the rotation angle of the dial with respect to the beam, and $q_3\,[m]$ that corresponds to the translation of the cart in the horizontal direction. The generalized force components are $Q_1\,[N\cdot m]$, $Q_2\,[N\cdot m]$, and $Q_3\,[N]$. In the underactuated mode of operation it is assumed that $Q_2\equiv 0$, and $Q_3\equiv 0$, i.e. only a single control signal Q_1 can be used for controlling the motion of the axles q_1, q_2, and q_3.

$$\begin{bmatrix} (mL^2+\Theta) & \Theta & mL\cos(q_1) \\ \Theta & \Theta & 0 \\ mL\cos(q_1) & 0 & (m+M) \end{bmatrix} \begin{bmatrix} \ddot{q_1} \\ \ddot{q_2} \\ \ddot{q_3} \end{bmatrix} + \\ + \begin{bmatrix} -mLg\sin(q_1) \\ 0 \\ -mL\sin(q_1)\dot{q_1}^2 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \tag{4}$$

[In the model the mass of the beam was neglected. Furthermore, it was assumed that the dial is connected to the beam by axle q_2 at its mass center point. These facts explain certain simplifications that are present in (4).]

In the sequel the idea of the adaptive optimal control is expounded.

Table 1
The parameters of the approximate model and that of the actually controlled system's model

Parameter	Approximate	Exact
$M\,[kg]$ mass of the cart	1.5×5	5
$L\,[m]$ length of the beam	1.5×1	1
$m\,[kg]$ mass of the dial	1.5×2	2
$\Theta\,[kg \cdot m^2]$ inertia momentum of the dial	1.5×6	6
$g\,[m \cdot s^{-2}]$ gravitational acceleration	1.5×9.81	9.81

3.2 Application of Time Sharing in Adaptive Optimal Control

In our system a single active control torque Q_1 can be used for controlling the motion of q_1, q_2, and q_3. It is evidently impossible to track some *arbitrarily prescribed nominal trajectory* $[q_1^N(t), q_2^N(t), q_3^N(t)]^T$. In the practice, the *tracking imprecisions have to be distributed over the three coordinates*. In the classical solutions as in the LQR controller (e.g. [13]) a cost function is constructed of the tracking errors and the cost functional that is obtained as its integral is minimized. Some limitation for the control forces can be built in the cost functional, too. The minimization can be executed by the use of the *Riccati equations* in the simpler cases, or by *nonlinear programming* in the more general ones.

In our approach the cost function is completely eliminated according to the idea of time-sharing. The operation time of the controller is divided into disjoint time intervals in which simultaneously the motion of only one coordinate is controlled while the other ones can propagate as they "want". In the next session the controller tries to keep at bay one of the other coordinates, and so on.

When the motion of q_1 is under control by the use of the last two equations of the matrix form in (4) $\ddot{q}_2$ and $\ddot{q}_3$ can be expressed by $\ddot{q}_1$, and these expressions can be substituted into the first equation with the application of the available approximate model parameters denoted by the subscript "a":

$$Q_1 = \left(m_a L_a^2 - \frac{m_a^2 L_a^2 \cos q_1^2}{m_a + M_a} \right) \ddot{q}_1 + \frac{(m_a L_a)^2 \cos q_1 \sin q_1 \dot{q}_1^2}{m_a + M_a} - m_a L_a g_a \sin q_1 \quad . \tag{5}$$

When the coordinate q_2 is under control from the third equation $\ddot{q}_3$ can be expressed by $\ddot{q}_1$, and from the second equation $\ddot{q}_1$ can be expressed by $\ddot{q}_2$. These values can be substituted into the first equation to obtain the appropriate control torque as

$$Q_1 = -\left(m_a L_a^2 - \frac{m_a^2 L_a^2 \cos q_1^2}{m_a + M_a} \right) \ddot{q}_2 + \frac{(m_a L_a)^2 \cos q_1 \sin q_1 \dot{q}_1^2}{m_a + M_a} - m_a L_a g_a \sin q_1 \quad . \tag{6}$$

Finally, when q_3 is under control, from the third equation $\ddot{q}_1$ can be expressed with $\ddot{q}_3$. With its use from the second equation $\ddot{q}_2$ can also be expressed by $\ddot{q}_3$. These quantities have to be substituted into the 1st equation to obtain Q_1 as:

$$Q_1 = \left(-\frac{(m_a + M_a)L_a}{\cos q_1} + m_a L_a \cos q_1 \right) \ddot{q}_3 + \frac{m_a L_a^2 \sin q_1 \dot{q}_1^2}{\cos q_1} - m_a g_a L_a \sin q_1 \ . \qquad (7)$$

To each session some time-slot was allocated in the simulations detailed in the sequel.

4 Numerical Calculations

4.1 Setting the Parameters of the Numerical Simulations

The numerical simulations were made by the use of the SCILAB (ver. 5.5.2) and its graphically programmable package XCOS. The kinematically prescribed trajectory tracking contained a PD-type feedback defined by (8) using a *"time-constant of tracking"* $\Lambda > 0$:

$$e(t) \stackrel{def}{=} q^N(t) - q(t) \text{ tracking error} \ , \qquad (8a)$$

$$\left(\Lambda + \frac{\mathrm{d}}{\mathrm{d}t} \right)^2 e(t) = 0 \text{ desired behavior leading to} \qquad (8b)$$

$$\ddot{q}^{Des} = \ddot{q}^N + 2\Lambda \dot{e} + \Lambda^2 e \ . \qquad (8c)$$

The numerical values are given in Table 2. Whenever the controlled axis was changed the adaptivity was switched off for a short period defined by the parameter *"time-slot of inactive adaptivity"*. This was necessary for "clearing the memory" of the adaptive controller that had inadequate antecedents since the past data at axle switching belonged to the previously controlled axle.

Table 2
The parameters of the adaptive controller and the numerical simulator

Parameter	**Value**
$\Lambda\,[s^{-1}]$	4
$K_c\,[nondimensional]$	0.5
$D_c\,[nondimensional]$	0.6
$J_c\,[nondimensional]$	0.2
$x_*\,[nondimensional]$ (dependent)	1.435116
$A_c \in \mathbb{R}^3 [\frac{s^2}{rad}, \frac{s^2}{rad}, \frac{s^2}{m}]^T$	$[-0.75, -2, -2]^T$
$\delta t\,[s]$ Cycle time	10^{-3}
$\Delta t\,[s]$ Time-slots for q_1, q_2, and q_3	$[2, 2, 1]^T$
$\tau_{NA}\,[s]$ Time-slot of inactive adaptivity	5×10^{-3}
Numerical integrator's method	Runge-Kutta 4(5) of SCILAB ver. 5.5.2
Maximum allowed time step in integration	Automatic

4.2 Simulation Results

The comparison of the operation of the non-adaptive and adaptive controllers can be done by using Figs. 3–5.

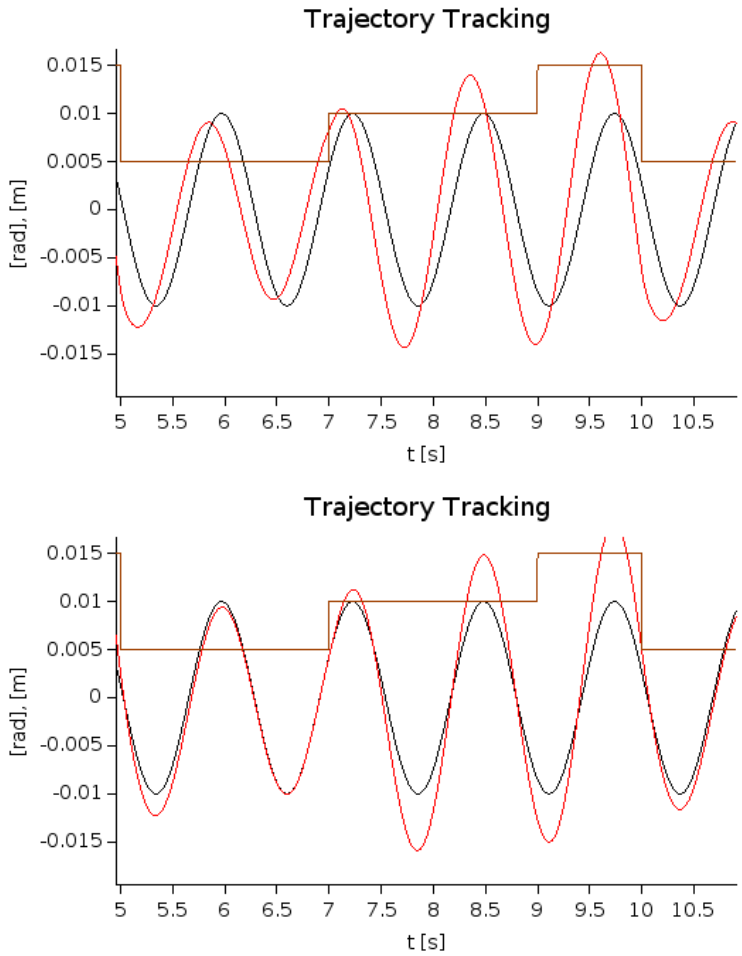


Figure 3
Tracking of q_1 in the non-adaptive (upper chart) and the adaptive (lower chart) cases: q_1^N: black, q_1: red lines, the time slots are indicated by the step function (brown line): increasing values belong to q_1, q_2, and q_3, respectively

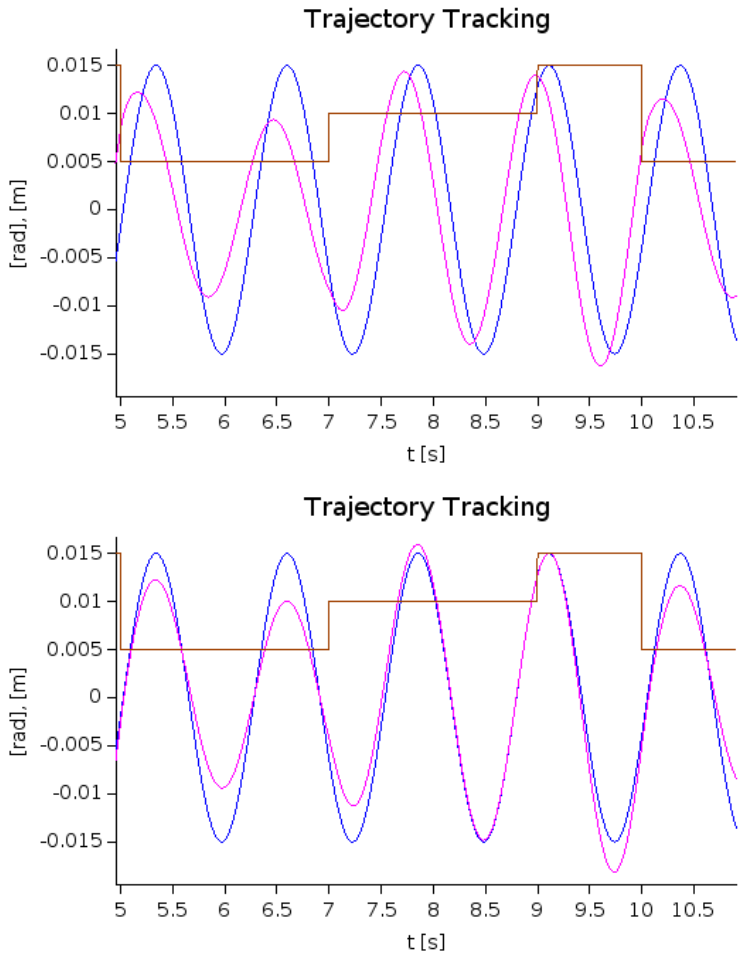


Figure 4
Tracking of q_2 in the non-adaptive (upper chart) and the adaptive (lower chart) cases: q_2^N: blue, q_2: magenta lines, the time slots are indicated by the step function (brown line): increasing values belong to q_1, q_2, and q_3, respectively

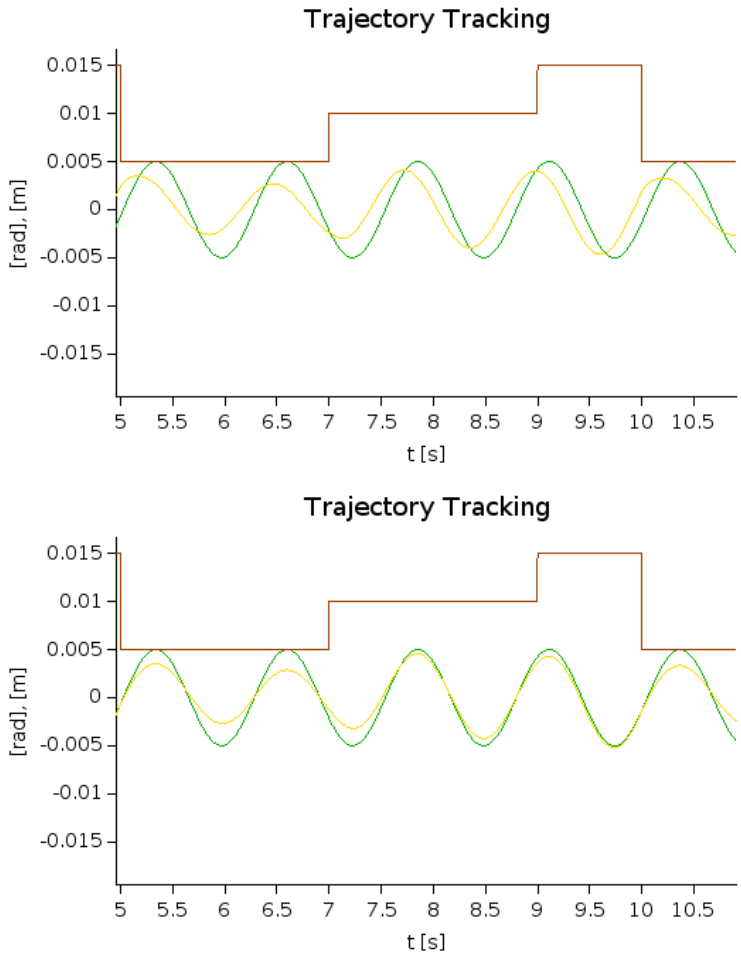


Figure 5
Tracking of q_3 in the non-adaptive (upper chart) and the adaptive (lower chart) cases: q_3^N: green, q_3: ocher lines, the time slots are indicated by the step function (brown line): increasing values belong to q_1, q_2, and q_3, respectively

More details are revealed by Figs. 6–8, displaying the tracking errors versus time. It is evident that within its own time-slot each of the adaptively controlled axle was adjusted to track the nominal motion that was prescribed to it. In the non-adaptive slots they left the nominal trajectory. However, due to the rotation of the adaptive slots the motion of each axle was kept in the vicinity of the nominal trajectory.

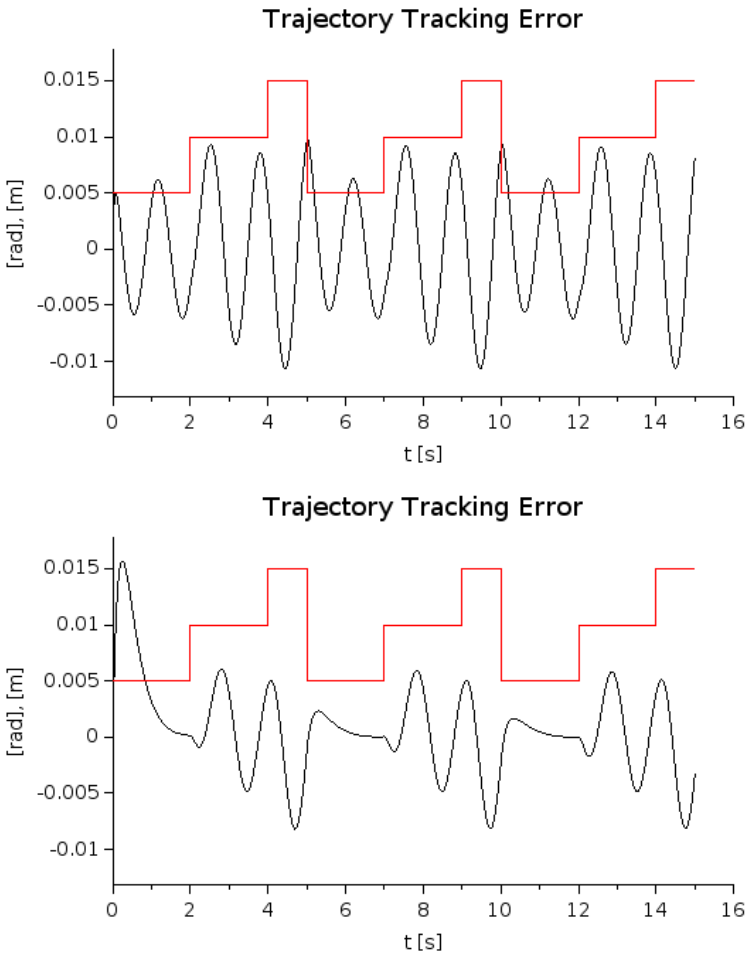


Figure 6
Tracking error of q_1 in the non-adaptive (upper chart) and the adaptive (lower chart) cases: $q_1^N - q_1$: black line, the time slots are indicated by the step function (red line): increasing values belong to q_1, q_2, and q_3, respectively

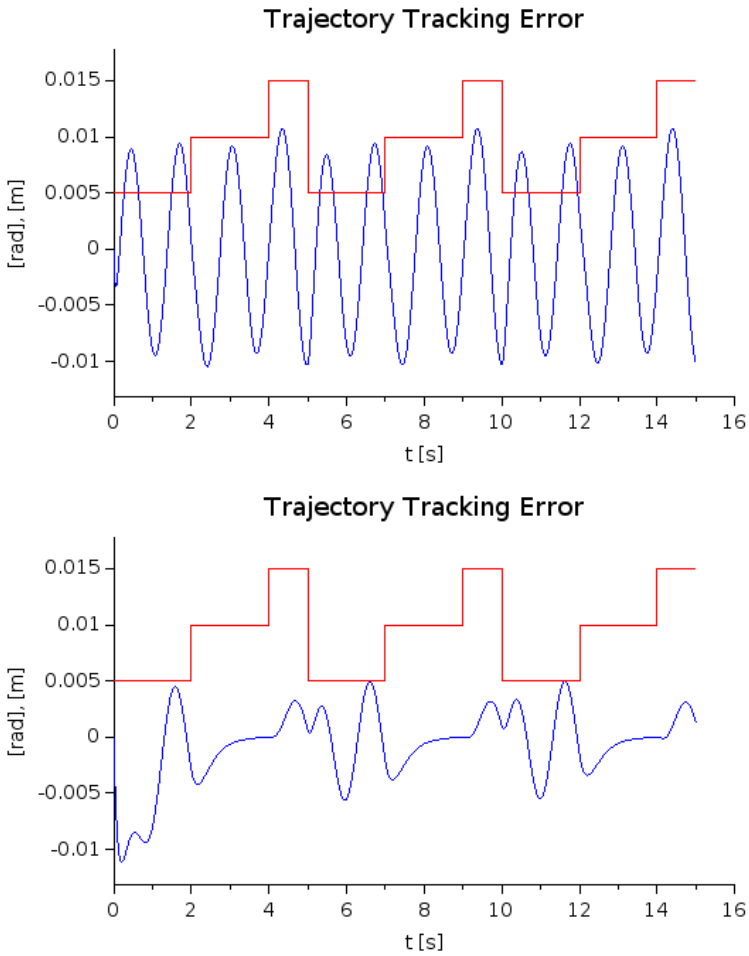


Figure 7
Tracking error of q_2 in the non-adaptive (upper chart) and the adaptive (lower chart) cases: $q_2^N - q_2$: blue line, the time slots are indicated by the step function (red line): increasing values belong to q_1, q_2, and q_3, respectively

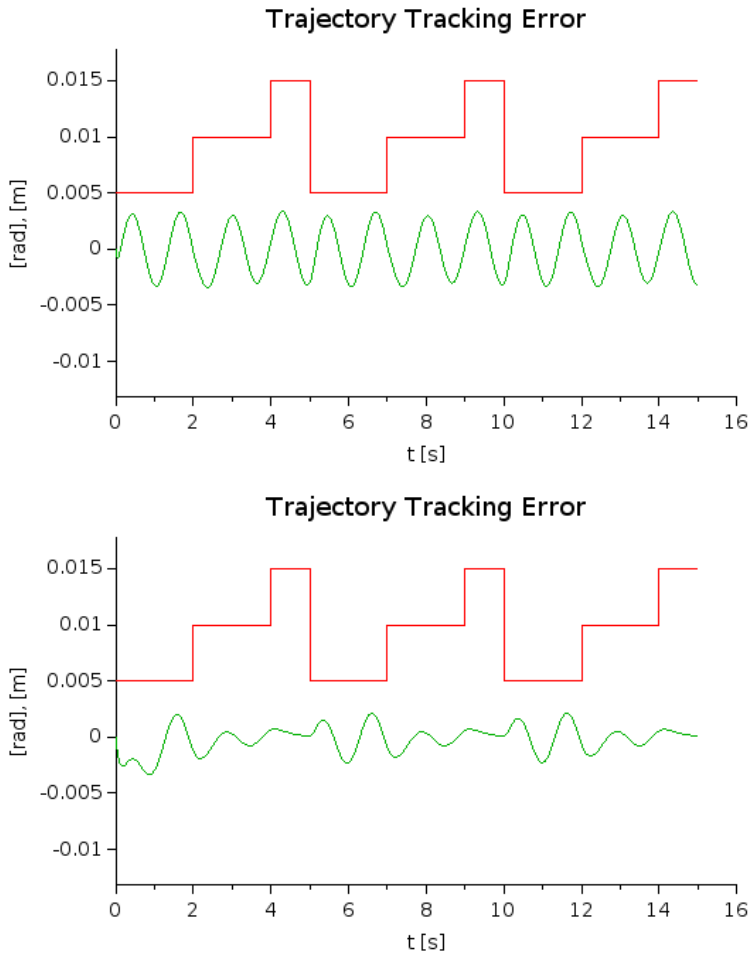


Figure 8
Tracking error of q_3 in the non-adaptive (upper chart) and the adaptive (lower chart) cases: $q_3^N - q_3$: green line, the time slots are indicated by the step function (red line): increasing values belong to q_1, q_2, and q_3, respectively

The generalized force Q_1 exerted by the controller is given in Fig. 9. The adaptive and the non-adaptive cases worked with control torques within the same order of magnitude.

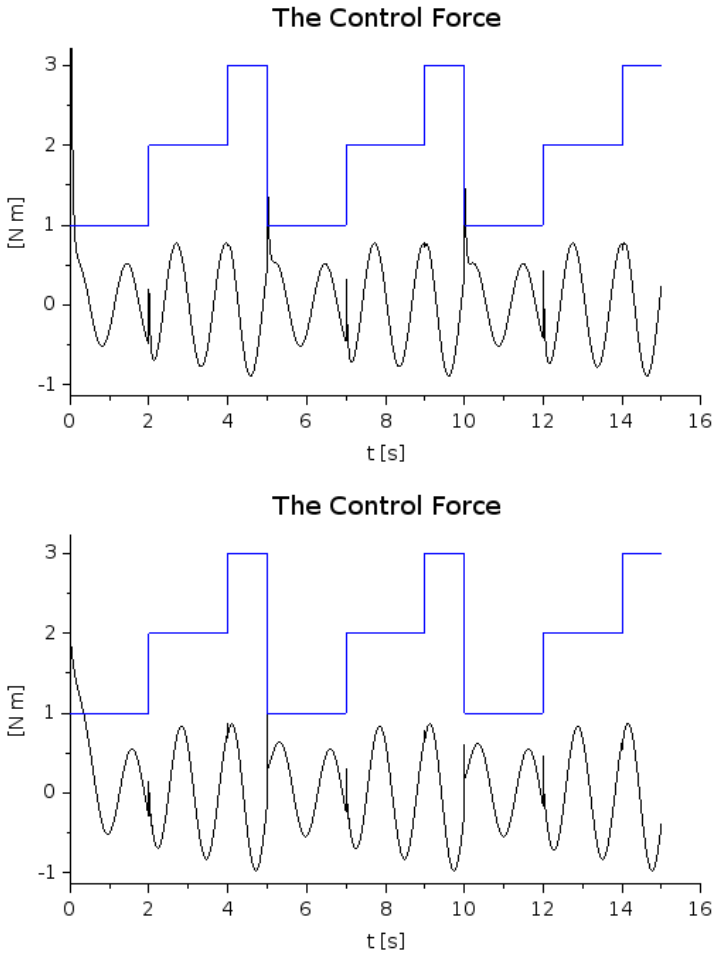


Figure 9
The control torque Q_1 in the non-adaptive (upper chart) and the adaptive (lower chart) cases; the time slots are indicated by the step function (blue line): increasing values belong to q_1, q_2, and q_3, respectively

The operation of the controller can be understood by considering the "desired" and the "realized" second time-derivatives of the generalized coordinates, as they are given in Figs. 10–12.

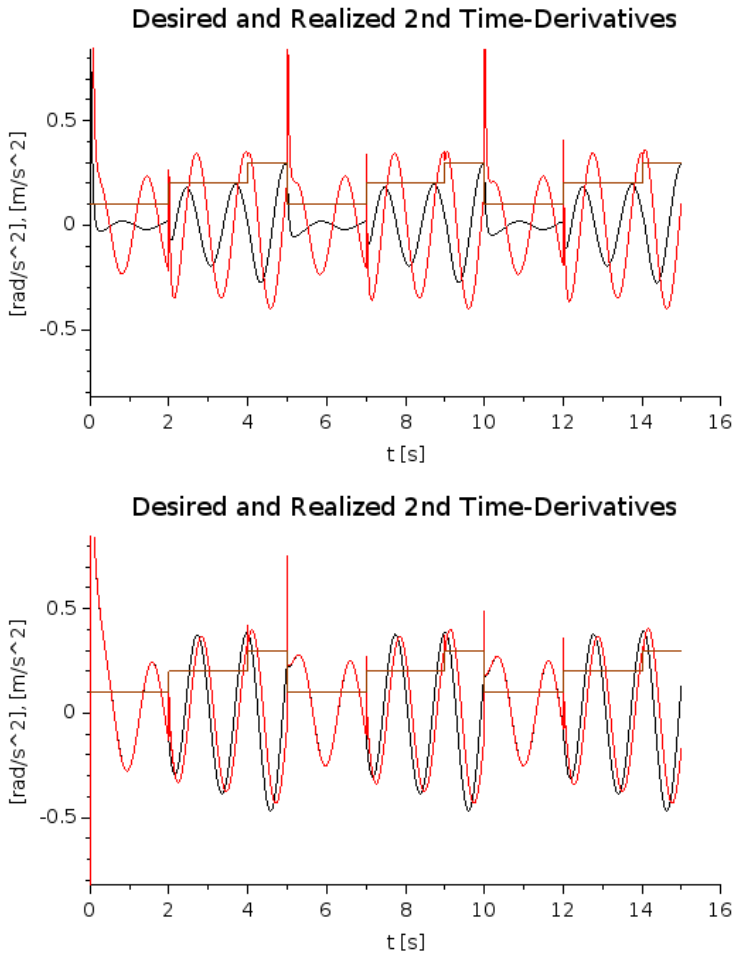


Figure 10
The "desired" $\ddot{q}_1^{Des}$ (black lines) and the realized $\ddot{q}_1$ (red lines) values in the non-adaptive (upper chart) and the adaptive (lower chart) cases; the time slots are indicated by the step function (brown line): increasing values belong to q_1, q_2, and q_3, respectively

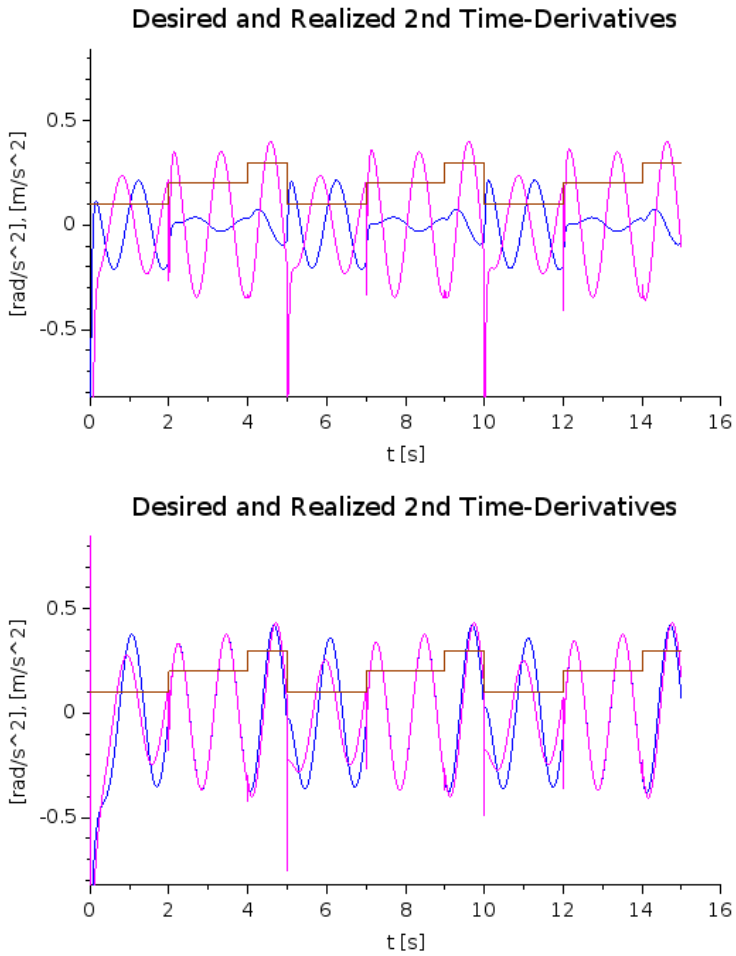


Figure 11
The "desired" $\ddot{q}_2^{Des}$ (blue lines) and the realized $\ddot{q}_2$ (magenta lines) values in the non-adaptive (upper chart) and the adaptive (lower chart) cases; the time slots are indicated by the step function (brown line): increasing values belong to q_1, q_2, and q_3, respectively

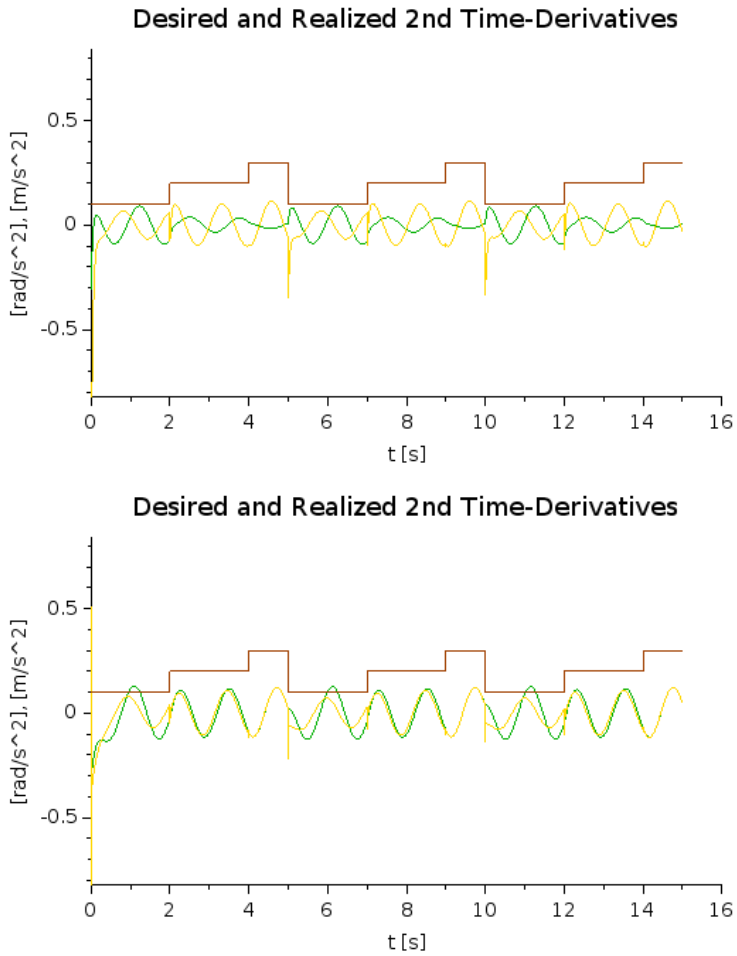


Figure 12
The "desired" $\ddot{q}_3^{Des}$ (green lines) and the realized $\ddot{q}_3$ (ocher lines) values in the non-adaptive (upper chart) and the adaptive (lower chart) cases; the time slots are indicated by the step function (brown line): increasing values belong to q_1, q_2, and q_3, respectively

Evidently, in the appropriate adaptive sessions, the suggested fixed point transformation precisely realized the kinematically prescribed trajectory tracking, for the actually controlled coordinate.

Conclusions

In this paper, a novel fixed point transformation, called *"Stretched Sigmoids Generated Fixed Point Transformation (SSGFPT)"* was suggested, for the realization of *"Adaptive Optimal Control"* for an underactuated Classical Mechanical system, a TORA model.

It was shown that for SISO systems of monotonic increasing response functions of bounded derivatives this controller can realize globally stable operation. This operation is possible because, in this case, the fixed point transformation used for transforming the computation of the necessary control force into the problem of finding the fixed point of a contractive map via iteration, has unbounded basin of attraction. Furthermore, it guarantees very fast convergence, if the actual point is far from the fixed point.

The main point of the optimization considered herein, is the idea of using cost-function free optimization, in which, the necessary compromise between the contradictory prescriptions, is found via time-sharing, realized by rotating time-slots. In contrast to the traditional optimal control, that formally is made complicated, because of the use of cost functionals, the herein applied approach allows simple combination with a non Lyapunov function-based adaptive design.

The numerical simulations well exemplified the operation of the suggested method.

In our future work, we should like to proceed in two separate directions. First, we wish to generalize the SSGFPT method from SISO to multivariable (MIMO) systems. Second, we wish to study the possibilities to include further limiting factors in the computations. Traditionally these factors appear as contributions to the cost functions.

Acknowledgement

This work was supported by the *Research and Innovation Center of Óbuda University*.

At last, but not least, the authors express their thanks to Professor Antal Bejczy with whom they were in professional connection, since the 1980s. At the beginning they met at various conferences. Later, on Professor Bejczy regularly visited Hungary and supported the authors' activities even within "official" frameworks. With this paper the authors wish to give a humble contribution to the commemoration of his great achievements in robotics and control.

References

[1] R.T. Bupp, D.S. Bernstein, and V.T. Coppola. A benchmark problem for nonlinear control design: Problem statement, experiment testbed and passive nonlinear compensation. *Proc. of the American Control Conference, Seattle, US, 1995*, pages 4363–4376, 1995.

[2] M. Jankovic, D. Fontanie, and P.V. Kokotovic. TORA example: Cascade- and passivity based control designs. *IEEE Transaction on Control System Technologies*, 4:292–297, 2006.

[3] P. Baranyi, Z. Petres, P. Várlaki, and P. Michelberger. Observer and control law design to the TORA system via TPDC framework. *WSEAS Transactions on Systems*, 1(5):156–163, 2006.

[4] Authors of a special issue. A special issue presenting 9 papers on controlling the TORA system. *International Journal of Robust and Nonlinear Control*, 8:305–457, 1998.

[5] F. Chee and T. Fernando. *Closed-Loop Control of Blood Glucose*. Springer, 2007.

[6] C. Dalla Man, R.A. Rizza, and C. Cobelli. Meal simulation model of glucose-insulin system. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 54(10):1740–1749, 2007.

[7] L. Magni, D.M. Raimondo, L. Bossi, C. Dalla Man, G. De Nicolao, B. Kovatchev, and C. Cobelli. Model Predictive Control of Type 1 Diabetes: An in silico trial. *J Diab Sci Techn*, 1:804–812, 2007.

[8] A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 1952.

[9] T. Matsumoto. A chaotic attractor from Chua's circuit. *IEEE Transactions on Circuits and Systems*, CAS-31(12):1055–1058, 1984.

[10] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophys J.*, 1(6):445–466, 1961.

[11] R. Bellman. *Dynamic Programming*. Princeton Univ. Press, Princeton, N. J., 1957.

[12] R.E. Kalman. Contribution to the theory of optimal control. *Boletin Sociedad Matematica Mexicana*, 5(1):102–119, 1960.

[13] Brian D.O. Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice – Hall International, Inc., A Division of Simon & Schuster, Englewood Cliffs, NJ 07632, 1989.

[14] Tayfun Çimen. State-dependent Riccati equation in nonlinear optimal control synthesis. *In the Proc. of the Special International Conference on Complex Systems: Synergy of Control, Communications and Computing - COSY 2011, Hotel Metropol Resort, Ohrid, Republic of Macedonia, September, 16 – 20, 2011*, pages 321–332, 2011.

[15] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978.

[16] A.M. Lyapunov. *A general task about the stability of motion. (in Russian)*. Ph.D. Thesis, University of Kazan, Tatarstan (Russia), 1892.

[17] A.M. Lyapunov. *Stability of motion*. Academic Press, New-York and London, 1966.

[18] Jean-Jacques E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall International, Inc., Englewood Cliffs, New Jersey, 1991.

[19] C.C. Nguyen, S.S. Antrazi, Zhen-Lei Zhou, and C.E. Campbell Jr. Adaptive control of a Stewart platform-based manipulator. *Journal of Robotic Systems*, 10(5):657–687, 1993.

[20] J. Somló, B. Lantos, and P.T. Cát. *Advanced Robot Control*. Akadémiai Kiadó, Budapest, 2002.

[21] K. Hosseini-Suny, H. Momeni, and F. Janabi-Sharifi. Model Reference Adaptive Control design for a teleoperation system with output prediction. *J Intell Robot Syst*, DOI 10.1007/s10846-010-9400-4:1–21, 2010.

[22] I. Sekaj and V. Veselý. Robust output feedback controller design: Genetic Algorithm approach. *IMA J Math Control Info*, 22(3):257–265, 2005.

[23] J.L. Chen and Wei-Der Chang. Feedback linearization control of a two-link robot using a Multi-Crossover Genetic Algorithm. *Expert Systems with Applications*, 2(2 Part 2):4154–4159, 2009.

[24] S.V. Emelyanov, S.K. Korovin, and L.V. Levantovsky. Higher order sliding regimes in the binary control systems. *Soviet Physics*, 31:291–293, 1986.

[25] V.I. Utkin. *Sliding Modes in Optimization and Control Problems*. Springer Verlag, New York, 1992.

[26] A. Levant. Arbitrary-order sliding modes with finite time convergence. *In Proc. of the 6th IEEE Mediterranean Conference on Control and Systems, June 9-11, 1998, Alghero, Sardinia, Italy*, 1998.

[27] J.K. Tar, J.F. Bitó, L. Nádai, and J.A. Tenreiro Machado. Robust Fixed Point Transformations in adaptive control using local basin of attraction. *Acta Polytechnica Hungarica*, 6(1):21–37, 2009.

[28] J.K. Tar. *Adaptive Control of Smooth Nonlinear Systems Based on Lucid Geometric Interpretation (DSc Dissertation)*. Hungarian Academy of Sciences, Budapest, Hungary, 2012.

[29] Tjalling J. Ypma. Historical development of the Newton-Raphson method. *SIAM Review*, 37(4):531–551, 1995.

[30] C.T. Kelley. *Solving Nonlinear Equations with Newton's Method, no 1 in Fundamentals of Algorithms*. SIAM, 2003.

[31] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms, Springer Series in Computational Mathematics, Vol. 35*. Springer, Berlin, 2004.

[32] S. Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales (About the Operations in the Abstract Sets and Their Application to Integral Equations). *Fund. Math.*, 3:133–181, 1922.

[33] S. Arimoto, S. Kawamura, and F. Miyazaki. Bettering operation of robots by learning. *Journal of Robotic Systems*, 1(2):123–140, 1984.

[34] S. Gunnarsson and M. Norrlöf. On the design of ILC algorithms using optimization. *Automatica*, 37:2011–2016, 2001.

[35] D.H. Owens and J. Hätönen. Iterative learning control – An optimization paradigm. *Annual Reviews in Control*, 29:57–70, 2005.

[36] Y. Wang, F. Gao, and F.J. Doyle III. Survey on iterative learning control, repetitive control, and run-to-run control. *Journal of Process Control*, 19:1589–1600, 2009.

[37] M. Deniša, A. Ude, and A. Gams. Adaptation of motor primitives to the environment through learning and statistical generalization. *Proc. of the 24th International Workshop on Robotics in Alpe Adria Danube Region (RAAD2015), May 27-29 2015, Bucharest, Romania*, pages 1–8, 2015.

[38] J.K. Tar. *Towards Replacing Lyapunov's "Direct" Method in Adaptive Control of Nonlinear Systems (invited plenary lecture at the Mathematical Methods in Engineering Intl. Symp. (MME), 21-24 October 2010, Coimbra, Portugal), chapter in Mathematical Methods in Engineering (Eds.: N.M. Fonseca Ferreira & J.A. Tenreiro Machado)*. Springer Dordrecht, Heidelberg, New York, London, 2014.

[39] J.K. Tar, L. Nádai, I.J. Rudas, and T.A. Várkonyi. RFPT-based adaptive control stabilized by fuzzy parameter tuning. *In Proc. of the 9^{th} European Workshop on Advanced Control and Diagnosis (ACD 2011), Budapest, Hungary*, pages 1–8, 2011.

[40] K. Kósi, J.K. Tar, and I.J. Rudas. Improvement of the stability of RFPT-based adaptive controllers by observing "precursor oscillations". *In Proc. of the 9^{th} IEEE Intl. Conf. on Computational Cybernetics, Tihany, Hungary*, pages 267–272, 2013.

[41] A. Dineva, J.K. Tar, and A.R. Várkonyi-Kóczy. Novel generation of Fixed Point Transformation for the adaptive control of a nonlinear neuron model. *In proc. of the IEEE International Conference on Systems, Man, and Cybernetics, October 10-13, 2015, Hong Kong (SMC 2015)*, pages 987–992, 2015.

[42] A. Dineva, J.K. Tar, A.R. Várkonyi-Kóczy, and V. Piuri. Generalization of a sigmoid generated fixed point transformation from SISO to MIMO systems. *Accepted for publication at the IEEE 19th International Conference on Intelligent Engineering Systems, September 3-5, 2015, Bratislava, Slovakia (INES 2015)*, page paper no. 3, 2015.

[43] A. Dineva, J.K. Tar, A.R. Várkonyi-Kóczy, and V. Piuri. Replacement of parameter tuning with simple calculation in adaptive control using "Sigmoid Generated Fixed Point Transformation". *Accepted for publication at the 13th IEEE International Symposium on Intelligent Systems and Informatics, September 17-19, Subotica, Serbia (SISY 2015)*, page paper no. 3, 2015.

[44] K. Kósi, J.F. Bitó, and J.K. Tar. On the effects of strong asymmetries on the adaptive controllers based on robust fixed point transformations. *In Proc. of*

the IEEE 10^{th} *Jubilee Intl. Symp. on Intelligent Systems and Informatics (SISY 2012), Subotica, Serbia*, pages 259–264, 2012.

[45] K. Kósi, A. Dineva, and J.K. Tar. Increased cycle time achieved by fractional derivatives in the adaptive control of the Brusselator model. *In Proc. of the* 11^{th} *IEEE Intl. Symp. on Applied Machine Intelligence and Informatics (SAMI), Herl'any, Slovakia*, pages 65–70, 2013.

[46] B. Kurtán. *Investigation and Adaptive Synchronization of Physiological Neuron Models via Numerical Simulations (MSc Thesis, Supervisor: J.K. Tar).* John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary, 2014.

[47] J.F. Bitó and J.K. Tar. Fixed point transformation-based adaptive control of the Hodgkin-Huxley neuron. *Proc. of the Workshop on Information Technology and Bionics – Symposium in Memory of Tamás Roska, 23-24 June, 2015, Pázmány University* e*Press, Budapest, Hungary*, pages 69–72, 2015.

[48] L. Nádai and I. Felde. Adaptive control by using time-sharing and Fixed Point Transformation. *In proc. of the IEEE International Conference on Systems, Man, and Cybernetics, October 10-13, 2015, Hong Kong (SMC 2015)*, pages 2492–2497, 2015.

[49] Gy. Eigner, J.K. Tar, and L. Kovács. Adaptive control solution for T1DM control. *In: Proc. of the 10th IEEE Jubilee International Symposium on Applied Computational Intellience and Informatics (SACI 2015), from May 21 to 23 2015, Timişoara, Romania*, pages 215–220, 2015.

[50] Gy. Eigner, P. Horváth, J.K. Tar, I.J. Rudas, and L. Kovács. Application of Robust Fixed Point control in case of T1DM. *In proc. of the IEEE International Conference on Systems, Man, and Cybernetics, October 10-13, 2015, Hong Kong (SMC 2015)*, pages 2459–2463, 2015.

[51] K. Kósi. *Further development and novel applications of the Robust Fixed Point Transformation-based adaptive control (PhD Thesis, supervisor: J.K. Tar).* Óbuda University, Budapest, Hungary, 2015.

[52] K. Kósi, Sz. Hajdu, J.F. Bitó, and J.K. Tar. Chaos formation and reduction in Robust Fixed Point Transformations based adaptive control. *In Proc. of the 4th IEEE Intl. Conf. on Nonlinear Science and Complexity (NSC 2012), Budapest, Hungary*, pages 211–216, 2012.

[53] K. Kósi, Á. Breier, and J.K. Tar. Chaos patterns in a 3 degree of freedom control with robust fixed point transformation. *In Proc. of the* 13^{th} *IEEE Intl. Symp. on Computational Intelligence and Informatics, Budapest, Hungary*, pages 1–5, 2012.

Control of the Differentially-driven Mobile Robot in the Environment with a Non-Convex Star-Shape Obstacle: Simulation and Experiments

Wojciech Kowalczyk, Krzysztof Kozłowski

Poznań University of Technology, Chair of Control and Systems Engineering, Piotrowo 3A, 60-965 Poznań, Poland
wojciech.kowalczyk@put.poznan.pl, krzysztof.kozlowski@put.poznan.pl

Abstract: The paper presents numerical verification and experimental results for the set-point control of the nonholonomic mobile robot. The task is to move to the goal and reach it with desired orientation avoiding collisions with static obstacles. The obstacles in the task space are modelled using analytic functions. The algorithm is investigated for both convex and non-convex star-shape obstacles.

Keywords: nonholonomic mobile robot; collision avoidance; navigation function; set-point control

1 Introduction

In the mid-1980s Khatib [1] presented a crucial idea of using repulsive and attractive interactions to avoid collision with obstacles and move to a set goal. This approach utilized local artificial potential functions (APF) associated with the obstacles and the goal. Local minima were at a significant disadvantage using this method.

In 1990 Rimon and Koditschek in series of publications [3], [4], [5], [6] proposed a navigation function that is a global artificial potential function without local minima. This method was applicable for a variety of task space configurations: sphere worlds, star worlds and trees of stars. The shapes of the obstacles and also the shape of the task space are described by analytic functions. The free space (task space) remains after all obstacles have been removed from the workspace. It must be emphasized that even in the case of the properly designed navigation function there still remain saddle points and their number is equal to the number of the obstacles.

In 2004 Urakubo [2] proposed an extension of the above approach. In this method nonholonomic constraints of the differentially driven mobile robot are taken into account. Urakubo also introduced methodology based on the time-varying functions such that the robot leaves the saddle point.

In the next years the navigation function was used to control multi-robot systems [8], [9], [10], [11]. In all of them robots are treated as obstacles. Problems of conflicts resolution between agents, limited knowledge of the environment and nonholonomic constraints of mobile platforms are addressed it these papers.

Another approach for collision-free control of nonholonomic mobile robot was proposed in [13]. The algorithm does not require the map of the environment. An iteratively generated path is optimized in free space (the curvature of the path) and replanning is done in the case of an unpredictable drift. The algorithm was verified numerically.

Algorithms that decompose the free space into subsets is another class of methods. In [14] the nonholonomic robot is driven through a sequence of triangular cells. This method was verified experimentally with process noise and both static and dynamic environments with obstacles.

In this paper simulation and experimental results for the algorithm presented in [2] are shown. According to the author's best knowledge this method has not been previously verified experimentally. In comparison to the original publication the effectiveness of the algorithm was investigated not only for a convex but also for non-convex obstacles.

It should be mentioned that local artificial potential functions can also be used to solve complex collision avoidance problems [7] in the case of circular obstacles; however, they must be designed carefully to avoid local minima.

In Section 2 the kinematic model of the robot and control algorithm introduced in [2] are presented. In Section 3 simulation results are shown. Section 4 briefly describes the experimental test-bed and experiments. Finally, the paper finishes off with concluding remarks.

2 Control Algorithm

The model of the differentially driven mobile robot is given by the following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = B \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

where x, y and θ are position and orientation coordinates of the robot respectively, $\boldsymbol{u}=[v\ \omega]^T$ is control vector with v denoting linear velocity control and ω - angular velocity control of the mobile platform and

$$B=\begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}. \tag{2}$$

Control proposed in [2] is given by the following equation:

$$\begin{bmatrix}v\\ \omega\end{bmatrix}=-\left(a\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}+b\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\right)B^T\nabla V \tag{3}$$

where

$$b=-\bar{b}\,\frac{L^T\nabla V}{h(g)} \tag{4}$$

and ∇V denotes the gradient of the artificial potential function, $L=[\sin\theta\ -\cos\theta\ 0]^T$, $h(g)=g^2+\varepsilon_g\sqrt{g}$, $g=\left\|B^T\nabla V\right\|$. In the above equations a, $\bar{b}$ and ε_g are positive parameters.

The navigation function is given by the following equation:

$$V=\frac{C}{C^{\kappa}+\beta}, \tag{5}$$

where $C=\|\hat{r}\|^2+w\theta^2$, $w=\dfrac{k_w}{k_w+\|\hat{r}\|^2}$; k_w is a positive parameter that allows tuning the weight of orientation when the robot approaches the desired position. As noted in [4] by setting a sufficiently large value of κ it is ensured that (4) is a navigation function i.e. there are no local minima in the APF.

To solve the problem of local minima for the case of star shape obstacles position coordinates of the robot $r=[x\ y]^T$ are transformed to the model sphere world as follows:

$$\hat{r}=\left(1-\sum s_i(r)\right)r+\sum s_i(r)T_i(r) \tag{5}$$

where M – number of the obstacles, $r=[x\ y]^T$,

$$s_i(r) = \frac{\|r\|^2 \prod \beta_j(r)}{\|r\|^2 \prod \beta_j(r) + \lambda_s \beta_i(r)} \tag{6}$$

and

$$T_i(r) = \frac{\rho_i(1+\beta_i(r))}{\|r - q_i\|}(r - q_i) + p_i \,, \tag{7}$$

$$T_0(r) = \frac{\rho_0(1-\beta_0(r))}{\|r - q_0\|}(r - q_0) + p_0 \,. \tag{8}$$

In the above equations p_0, p_i represent the centers of the spheres to which original obstacles are transformed ρ_0, ρ_i are their radiuses, q_0, q_i are the centers of the stars (points from which all the rays cross the boundary of the obstacle once and only once) and β_0, β_i are analytic obstacle functions.

The sphere-world obstacle function is a product of the sphere obstacle functions:

$$\hat{\beta} = \prod_{i=0}^{M} \hat{\beta}_i \tag{9}$$

where

$$\hat{\beta}_i(\hat{r}) = \|\hat{r} - p_i\|^2 - \rho_i^2 \,, \tag{10}$$

$$\hat{\beta}_0(\hat{r}) = \rho_0^2 - \|\hat{r} - p_0\|^2 \,. \tag{11}$$

3 Simulations Results

Simulation results were obtained for the task space containing one circular obstacle and one star-shape obstacle. The origin is the desired position and the desired orientation is equal to zero.

As the navigation function (4) is the function of three variables it is not possible to show its graph. In Figs. 1a and 2a the x-y plane cross-sections of V for $\theta = 0$ and $\theta = \pi/2$ are shown, respectively. It can be observed that in the first case the global minimum exists in the center of the task space. In the second one the potential grows around the origin not allowing the robot to approach it with incorrect orientation. If the robot is too close it is repelled, but as the orientation

converge to the desired value the x-y components of the V drive robot to the origin.

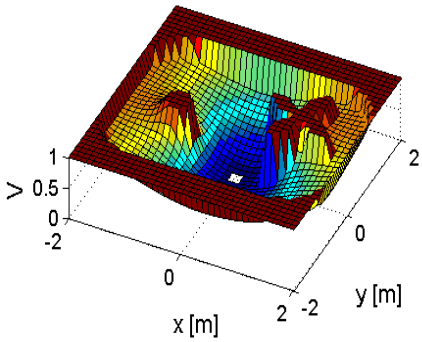


Figure 1a
APF for $\theta = 0$

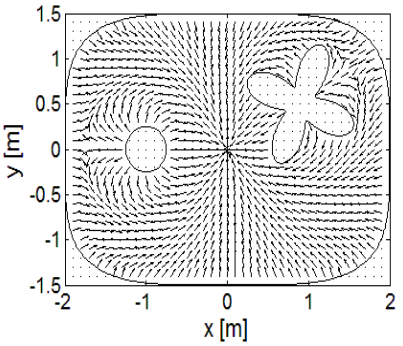


Figure 1b
Control vecor field $-\nabla V$ for $\theta = 0$

In Figs. 1b and 2b x-y plane cross-sections of the $-\nabla V$ for the same robot angles are shown (the length of the vectors was normalized).

The data were obtained for the following values of the parameters: $a = 0.25$, $\bar{b} = 1.25$, $\varepsilon_g = 10^{-6}$, $\kappa = 3$, $k_w = 0.1$, $\lambda_s = 10^6$.

In Fig. 3 the robot path in x-y plane is shown. The dashed line represents simulation results and solid line experimental data (their comparison will be discussed in the next section). The initial coordinates of the robot were as follows $[x \quad y \quad \theta]^T = [-1.01 \quad 0.92 \quad -2.92]^T$.

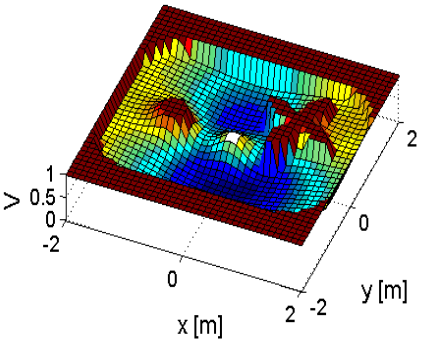


Figure 2a
APF for $\theta = \pi/2$

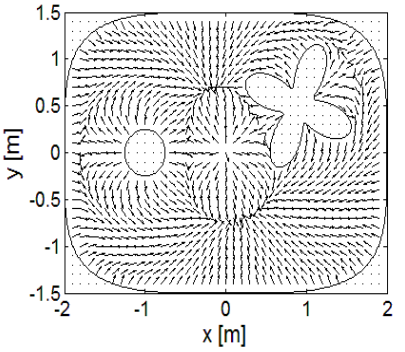


Figure 2b
Control vector field $-\nabla V$ for $\theta = \pi/2$

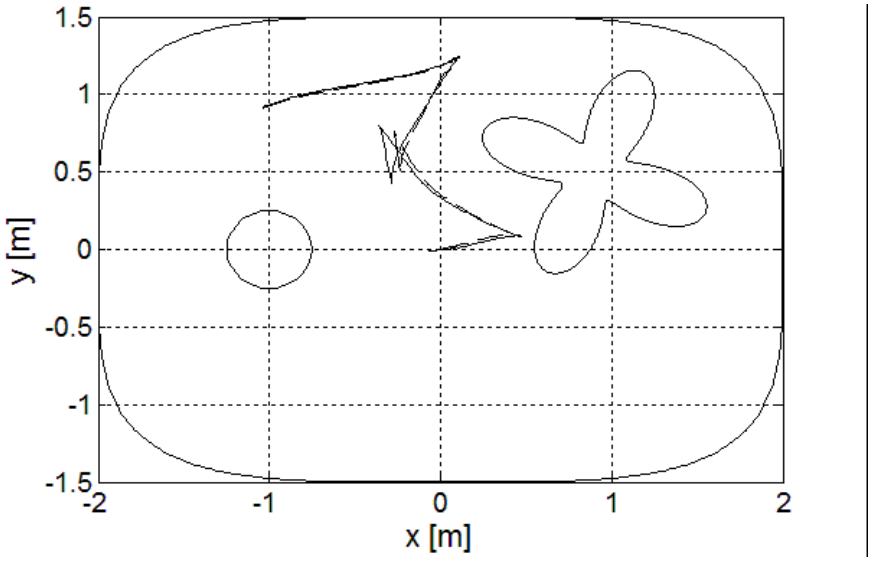


Figure 3
Robot motion in (x,y) plane; dashed line – simulation, solid line – experiment

In Fig. 4a time graphs of the position errors (e_x - solid line, e_y - dashed line) and orientation errors (e_θ - dot line) are shown. They converge to zero reaching small values in about 30 seconds.

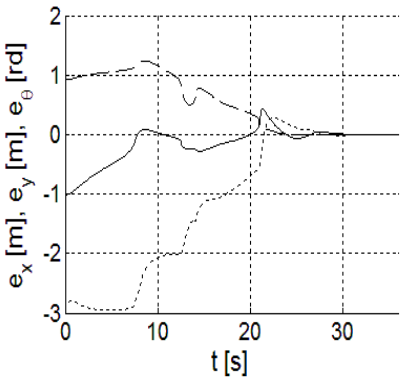


Figure 4a
State errors – simulation

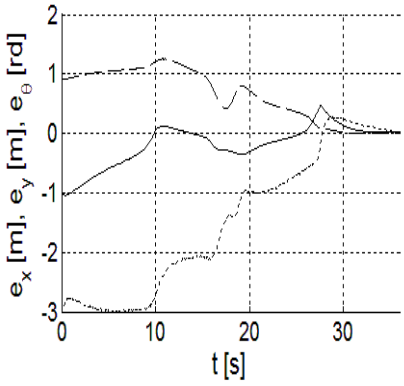


Figure 4b
State errors – experiment

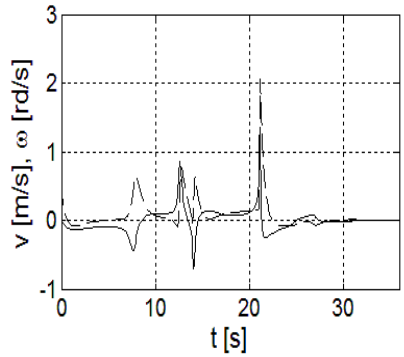


Figure 5a
Platform controls - simulation

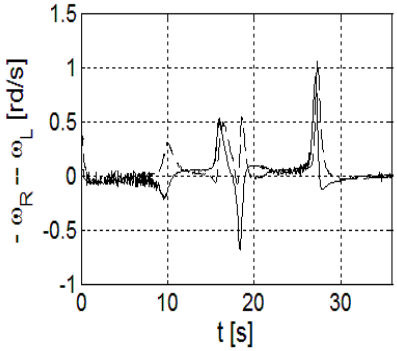


Figure 5b
Platform controls – experiment

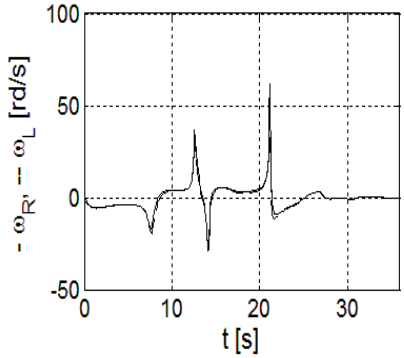


Figure 6a
Wheel controls - simulation

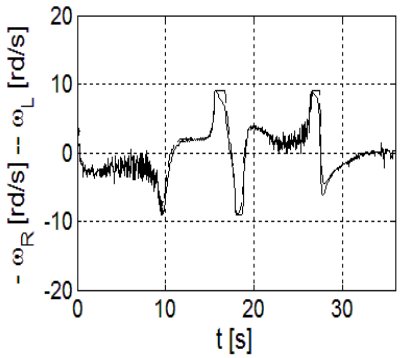


Figure 6b
Wheel controls – experiment

In Fig. 5a linear (solid line) and angular (dashed line) controls for the robot are presented.

Fig. 6a presents wheel control velocities (ω_R - solid line, ω_L - dashed line). There are peaks of the large values that cannot be achieved in the physical system. In the experiments presented in the next section these values are limited to achievable values.

In Fig. 7 robots path in x-y plane for a more complex case is shown. To reach the desired coordinates robot has to bypass the "leaf" of the obstacles. For this case local APF control algorithm usually fails trapping the robot in the local minimum. The initial coordinates of the robot were as follows $\begin{bmatrix} x & y & \theta \end{bmatrix}^T = \begin{bmatrix} 1.22 & 0.65 & -0.51 \end{bmatrix}^T$.

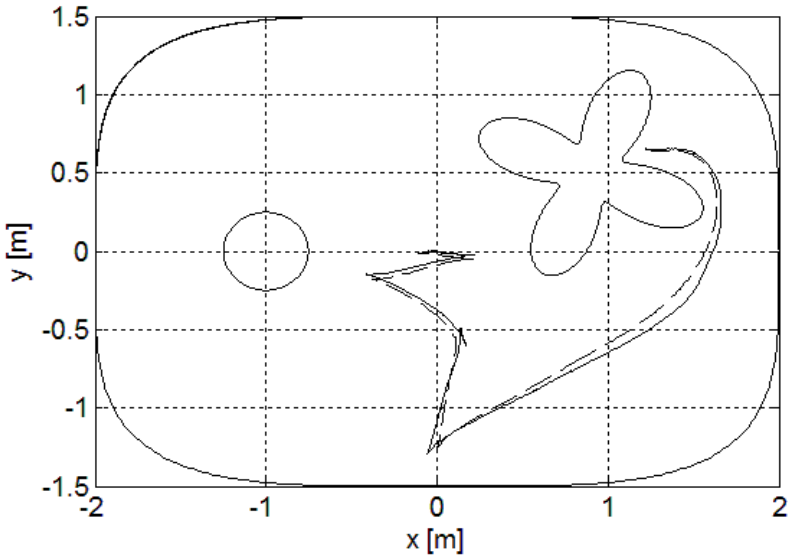


Figure 7
Robot motion in (x,y) plane; dashed line – simulation, solid line - experiment

In Fig. 8a position and orientation errors are presented, while Fig. 9a shows platform control signals. In Fig. 10a wheel control graphs are plotted.

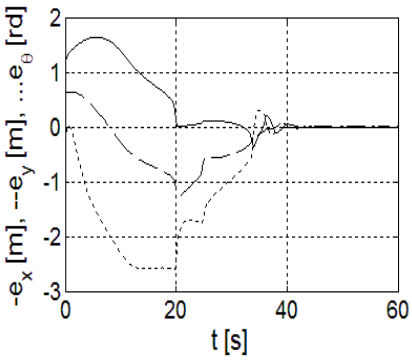


Figure 8a
State errors – simulation

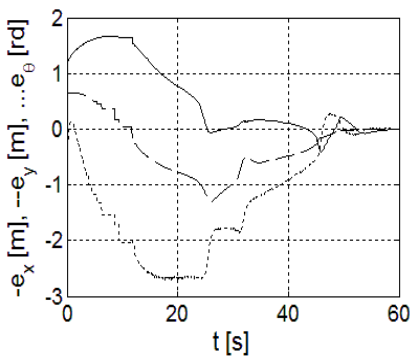


Figure 8b
State errors – experiment

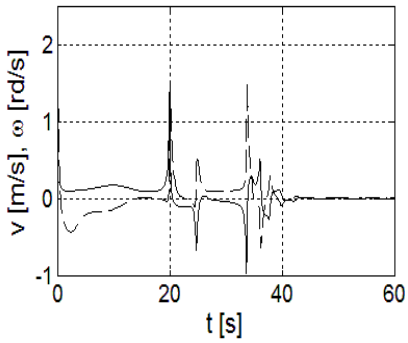


Figure 9a
Platform controls - simulation

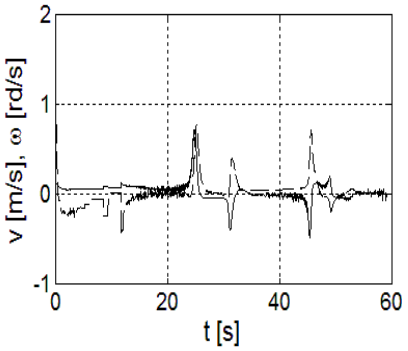


Figure 9b
Platform controls - experiment

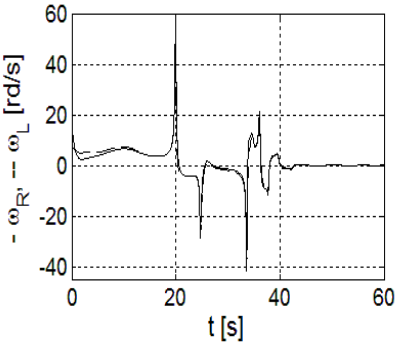


Figure 10a
Wheel controls - simulation

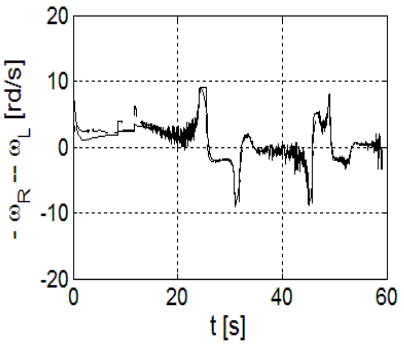


Figure 10b
Wheel controls - experiment

4 Experiments

4.1 Experimental Test-Bed

The algorithm was verified experimentally using the MTracker robot (Fig. 11). The MTracker is a differentially driven mobile robot. Its diameter is 170 mm and the height (in common configuration) – 65 mm. The linear velocity can reach the value of 1 m/s. The low level motion controller is implemented with the signal processor TMS 320F28335 150 MHz. Presented results were obtained using a robot expanded with Intel Atom 1,6 GHz PC board and Wi-Fi link used to obtain localization data from the external vision system. The robot was equipped with an LED marker for reliable and fast visual recognition. More details about the test-bed including robot configuration, communication and localization system can be found in [12].

Figure 11
MTracker robot used for experiments

4.2 Experiments

In Fig. 3 path of the robot in x-y plane is presented. The experimental data are represented with a solid line. As can be observed in the figure both simulation and experimental results are very similar. In this experiment robot moves in open areas of the environment but it is required to change the orientation. As it approaches the origin the direction of the motion is changed four times to drive the orientation according to the control rule.

In Fig. 4b position and orientation errors are shown. They reach near zero values in 35 seconds.

In Fig. 5b platform control signals are shown. There are peaks that are not achievable in the real system. In Fig. 6b wheel controls are presented. They are limited ($\omega_{RL\,max} = 9rd/s$) and scaled to keep the robot on the path. Limitation and scaling of the wheel velocities is the best solution of the large control signal problem as it is not affecting the mobile platform direction vector (it affects only its evolution in time). This approach separates a high level controller; whose primary objective is to avoid collisions, form the problems connected with the physical actuator limitations.

In Fig. 7 both simulation and experimental results for the star bypassing case are shown. The solid line represents the experimental path of the robot in the x-y plane and dashed line represents the simulation data. As can be observed in the figures in simulation the robot changed the direction of motion three times. In the experiment one more motion direction change was performed. Which was due to the disturbances and/or delays in the positioning system, however both graphs are very similar.

In Fig. 8b position and orientation errors are presented. They reach the near zero value in about 60 seconds (in simulation 40 seconds).

In Fig. 9b and 10b control signals for the platform and wheels are shown respectively. As in the previous experiment wheel velocities were limited and scaled.

Conclusions

Simulation and experimental results for navigation function control were presented. The non-convex star shape obstacle makes presented cases nontrivial. As shown in the attached figures the nonholonomic mobile robot reaches the desired position and orientation avoiding collisions with both convex and non-convex obstacles.

List of notations

x , y - position coordinates of the robot

θ - orientation of the robot

v - linear velocity control

ω - angular velocity control

B - zero space of the Pfaffian matrix

a , $\bar{b}$, ε_g , κ , k_w , λ_s - positive constant parameters

V - navigation function

∇V - gradient of the navigation function

b - is defined by Eq. (4) ; is a weighting coefficient of the skew symmetric matrix

L - Pfaffian matrix

g - norm of the product zero space matrix and gradient of the navigation function

$h(g)$ - nonlinear function of g; used as coefficient in Eq, (4)

C , w - positive coefficients used to define Eq. (5)

r - robot position

$\hat{r}$ - robot position in the auxiliary sphere world

β - product of the all obstacle functions

β_i - obstacle function associated with the *i*-th obstacle

s_i , T_i - auxiliary variable for star-to-sphere transformation

p_i - center of the *i*-th sphere

ρ_i - radius of the *i*-th sphere,

q_i - center of the *i*-th star,

M - number of the internal obstacles,

e_x , e_y , e_θ - coordinate errors,

ω_R , ω_L - wheel velocities,

$\omega_{RL\,max}$ - wheel velocity limit.

Acknowledgement

This work was supported by NCBiR grant PBS1/A3/8/2012 (RobREx).

References

[1] O. Khatib, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, The International Journal of Robotics Research, Vol. 5, No. 1, pp. 90-98, 1986

[2] T. Urakubo, K. Okuma and Y. Tada, Feedback Control of a Two Wheeled Mobile Robot with Obstacle Avoidance using Potential Functions, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vol. 3, pp. 2428-2433, 2004

[3] E. Rimon and D. E. Koditschek, Exact Robot Navigation using Cost Functions: the Case of Distinct Spherical Boundaries, IEEE International Conference on Robotics and Automation, Vol. 3, pp. 1791-1796, 1988

[4] E. Rimon and D. Koditschek, Exact Robot Navigation using Artificial Potential Functions, IEEE Transactions on Robotics and Automation, Vol. 8, No. 5, pp. 501-518, 1992

[5] E. Rimon and D. E. Koditschek, The Constructions of Analytic Diffeomorphisms for Exact Robot Navigation on Star Worlds, Transaction of the American Mathematical Society, Vol. 327, pp. 71-116, 1991

[6] E. Rimon and D. E. Koditschek, Exact Robot Navigation using Artificial Potential Fields, IEEE Transactions on Robotics and Automation, Vol. 8, No. 5, pp. 501-518, 1992

[7] W. Kowalczyk, K. Kozlowski, and J. K. Tar, Trajectory Tracking for Multiple Unicycles in the Environment with Obstacles, International Conference on Robotics in Alpe-Adria-Danube Region (RAAD), pp. 451-456, 2010

[8] I. Filippidis and K. J. Kyriakopoulos, Adjustable Navigation Functions for Unknown Sphere Worlds, IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 4276-4281, 2011

[9] D. V. Dimarogonasa, S. G. Loizoua, K. J. Kyriakopoulos and M. M. Zavlanos, A Feedback Stabilization and Collision Avoidance Scheme for Multiple Independent Non-Point Agents, Automatica, Vol. 42, No. 2, pp. 229-243, 2005

[10] G. Roussos, K. J. Kyriakopoulos, Decentralized and Prioritized Navigation and Collision Avoidance for Multiple Mobile Robots, Distributed Autonomous Robotic Systems - Springer Tracts in Advanced Robotics, Vol. 83, pp. 189-202, 2013

[11] G. Roussos, K. J. Kyriakopoulos, Completely Decentralised Navigation of Multiple Unicycle Agents with Prioritisation and Fault Tolerance, IEEE Conference on Decision and Control (CDC), pp. 1372-1377, 2010

[12] K. Kozlowski, W. Kowalczyk, B. Krysiak, M. Kielczewski, T. Jedwabny, Modular Architecture of the Multi-Robot System for Teleoperation and Formation Control Purposes," in Robot Motion and Control (RoMoCo), 2013 9th Workshop on, pp. 19-24, 3-5 July 2013

[13] M. K. Weir, M. P. Bott, High Quality Goal Connection for Nonholonomic Obstacle Navigation Allowing for Drift using Dynamic Potential Fields, in Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 3221-3226, 3-7 May 2010

[14] V. Kallem, A. T. Komoroski, V. Kumar, Sequential Composition for Navigating a Nonholonomic Cart in the Presence of Obstacles, in Robotics, IEEE Transactions on, Vol. 27, No. 6, pp. 1152-1159, Dec. 2011

Hierarchical control of unmanned ground vehicle formations using multi-body approach

Béla Lantos, György Max

Budapest University of Technology and Economics, Hungary
H-1117 Budapest, Magyar Tudósok krt. 2., Hungary
E-mail: *lantos@iit.bme.hu*, *max@iit.bme.hu*

Abstract: The paper deals with the formation control of Unmanned Ground Vehicles (UGVs) moving in horizontal plane. The control system consists of the high level centralized formation control of the UGVs and the low level decentralized PID type suspension, speed and steering control of the different vehicles. Both problems are discussed in multi-body assumptions. The paper presents the generalization of the multi-body method for underactuated car-like vehicles, developed originally for fully-actuated surface ships. In order to simplify the design and implementation on the formation level, an approximate single track dynamic model was assumed for each vehicle. At low level a more realistic two track dynamic model is used in the form of a multibody system in tree structure. This realistic nonlinear model is obtained by using Appell's method, Pacejka's magic formula for tyre-road connections and kinematic constraints expressing the nullity of vertical accelerations of the contact points. The interface between the higher and lower control levels is presented in the form of acceleration and steering angle prescriptions (output of high level). The decentralized control system of each vehicle converts the specifications in smooth reference signals and performs the desired motion. Simulation results of the high level control of UGV formations are presented for sine-shaped and circular paths.

Keywords: Formation Control, Unmanned Ground Vehicles, Multi-Body Approach, Tree Structured Vehicle, Pacejka's Magic Formula, Contact Point Constraints, Robust PID Control

1 Introduction

Formation control design and implementation is a complex and time-critical problem for which a hierarchical control system will be suggested. The high level subsystem deals with the formation control of vehicles satisfying connection constraints equivalent to the formation. The problem is a multi-body one in the sense that many vehicles take part in the formation. In order to simplify the design and realization at the formation level, approximate single track dynamic model will be assumed here for each vehicle. However, the realization needs a more realistic model for

the vehicles, hence a two track model will be considered for each vehicle at the lower control level. The interface between the higher and the lower control levels will be presented in the form of acceleration and steering angle prescriptions for the different vehicles and produced as output of the formation control. At low level, the decentralized control system of each vehicle converts the specifications in sufficiently smooth reference signals and performs the desired control based on robust PID type suspension, speed and steering control. At this level each vehicle is considered as a real multi-body system in tree structure. The motion of the formation is the result of both levels.

For stabilization of ground vehicles (robots) in formation the fusion of potential field method, passivity theory, dynamic inversion and LMI technique is a theoretically well founded approach if the inertia of the car-like vehicles has to be taken into consideration [1], [2]. Synchronized path following based on the fusion of backstepping control and passivity theory was suggested for surface ships [3]. Unfortunately this method cannot be used for UGVs because the dynamic model of the vehicles does not satisfy the strict-feedback form which is assumed for backstepping control. Another approach may be multi-body interpretation of the formation resulting in constrained control. This method was successfully applied in the formation control of full-actuated surface ships [4]. For formation flight control of constraint multi-body system [5] presents an approach where the aircraft model is of point-mass type and only position distance constraints are considered. Ground vehicles are underactuated and in many cases their inertia cannot be neglected, thus the original formulation for ships has to be generalized. One aim of the paper is to elaborate the necessary modification of the theory and illustrate its applicability for car-like UGVs.

In general, the ground vehicle can be regarded as a multi-body system whose base is the mobile chassis and the wheels are the end effectors. Several methods are available to find the kinematics and dynamic models of mobile robots [6, 7], but they mostly build on simplifying kinematic constraints that do not take into account the three-dimensional forces between the wheel and the ground. Other recent works [8, 9] use robotic description for modeling and validation of cars, but they do not deal with closed loop control and do not take into account the lateral and longitudinal offset in the vehicle's centre of gravity point (CoG). Numerical methods and symbolic software (Symoro+, OpenSYMORO) are available to find the dynamic model based on Newton-Euler method [10]. Another often used method is the Lagrange technique. In this paper, an alternative approach is introduced that uses the concept of acceleration energy and eliminates a large number of numerical steps of the Newton-Euler method. The algorithm is based on Appell's method which directly computes the dynamic model of the composite system. Although using symbolic software these methods result in equivalent dynamic models, however for only numerical computations (without the use of the symbolic results) they have different computation time. Another aim of the work is to develop a complex vehicle control system that is capable of eliminating the rolling and pitching effect through active suspension control system as well as maintaining a prescribed velocity and steering angle profile in closed loop. Such a system helps also studying the interaction of the system with the environment.

The structure of the paper is as follows. Section 2 summarizes the concept of the multi-body implementation for full-actuated systems (surface ships, robots etc.). Section 3 discusses the conversion of formation specifications into multi-body constraints. Section 4 gives the approximate dynamic model of a single ground vehicle and the generalization of the multi-body approach for UGVs. The simplified model consideres the tyre-road connections through the cornering stiffnesses which is a linear approximation. Section 5 presents the geometric and kinematic model and the tree structured topology of a single vehicle using the modified Denavit-Hartenberg form [11]. Here will be developed the vehicle's realistic two-track dynamic model using Appell's method by calculating the Gibbs functions of each segment. This section describes also the kinematic constraints and the external forces acting on the vehicle. The realistic model considers the tyre-road connections in the form of Pacejka's magic formulas hence the model is nonlinear in the state variables. In Section 6 the decentralized low-level control system will be presented in short form containing the reference signal design and the concept of PID type active suspension, driving and steering control based on the realistic nonlinear vehicle model. Section 7 shows the simulation results for formation control of UGVs using multi-body approach. Finally Section 8 summarizes the conclusions and the main directions of future research.

2 Fully actuated control of constrained multi-body systems

A single full-actuated marine vehicle moving in the horizontal plane can be modelled by

$$\begin{aligned} \dot{\eta} &= R(\psi)\nu \\ M\dot{\nu} + n(\nu, \dot{\nu}, \eta) &= \tau \end{aligned} \tag{1}$$

where $R(\psi)$is the rotation matrix from body to the (quasi) inertia frame, $\eta = (x, y, \psi)^T$ is the position and orientation, $\nu = (u, v, r)^T$ is the linear and angular velocity, M denotes the system inertia (for ships the rigid body inertia and the added mass) and n contains the centripetal, Coriolis, damping and gravity effects. Notice the similarity to robot control in 6-DOF where $H(q)\ddot{q} + h(q, \dot{q}) = \tau$ is the dynamic model of the robot in joint coordinates and the Jacobian $J(q)$ plays the role of the rotation matrix according to $\dot{x} = J(q)\dot{q}$.

If a set of constraints is given in the form of $C(\eta) = 0 \in R^p$ in the inertia system and the constraints Jacobian is denoted by $W(\eta) = \frac{\partial C(\eta)}{\partial \eta}$ then, by using the results of [12], the motion equation is modified to

$$M\dot{\nu} + n(\nu, \dot{\nu}, \eta) = \tau + \tau_c \tag{2}$$

where the constraint force τ_c has the form $\tau_c = -W(\eta)^T\lambda$ and λ is the Lagrange multiplier.

Transforming the motion equation into the inertia frame and using the fact that $R^T\tau_\eta = \tau \Rightarrow \tau_\eta = R\tau$, it yields

$$M_\eta(\eta)\ddot{\eta} + n_\eta(\nu,\dot{\nu},\eta) = \tau_\eta - R(\psi)W(\eta)^T\lambda. \tag{3}$$

It follows from $C(\eta) = 0$ that

$$\dot{C}(\eta) = \frac{\partial C}{\partial \eta}\dot{\eta} = W(\eta)\dot{\eta} = 0, \quad \ddot{C}(\eta) = W(\eta)\ddot{\eta} + \dot{W}(\eta)\dot{\eta} = 0. \tag{4}$$

Adding stabilizing terms we choose

$$\ddot{C} = -K_d\dot{C} - K_pC \tag{5}$$

with K_d, K_p diagonal and having positive elements. Then $s^2 + k_{di}s + k_{pi} = s^2 + 2\xi\omega_0 s + \omega_0^2 = 0$ is stable if $k_{di} = 2\xi\omega_0$ and $k_{pi} = \omega_0^2$ where $\xi > 0$ is the damping and ω_0 is the undamped eigenfrequency. In this case it follows

$$WM_\eta^{-1}(\tau_\eta - n_\eta - RW^T\lambda) + \dot{W}\dot{\eta} = -K_d\dot{C} - K_pC \tag{6}$$

$$\lambda = (WM_\eta^{-1}RW^T)^{-1}[WM_\eta^{-1}(\tau_\eta - n_\eta) + \dot{W}\dot{\eta} + K_d\dot{C} + K_pC] \tag{7}$$

if $WM_\eta^{-1}RW^T$ is invertible which is satisfied if W has full row rank.

If there are n vehicles then we can collect vectors into new vectors and matrices into new blockdiagonal matrices. The resulting vectors and matrices will be denoted further on by $\eta, n_\eta, \tau_\eta, \tau_c$ and M_η, respectively. The prescribed formation can be converted to the constraint $C(\eta) = 0$ having Jacobian $W(\eta)$.

The constraint force for the ith vehicle is

$$\begin{aligned}\tau_{ci} = \textstyle\sum_{k\in A_c^i}\sum_{j\in B_k} & -W_{ki}^T(W_kM_{\eta,ij}^{-1}R_{ij}^{-1T}W_k^T)^{-1}\times \\ & [W_{ki}M_{\eta,ij}(\tau_{\eta,ij} - n_{\eta,ij}) + K_{d,ki}\dot{C}_{ki} + K_{p,ki}C_{ki}]\end{aligned} \tag{8}$$

where A_c^i is the index set of vehicles staying in connection with vehicle i, B_k is the index of constraints selected by index k and $W_{ki} = 0$ for $k \notin A_c^i$.

3 Conversion of formation specifications to multi-body constraints

In the sequel the indexes p, o and d denote position, orientation and desired value, respectively, furthermore f is for fixed, tt for time dependent value and r for relative value between two vehicles. For simplicity denote here $\xi_i = (x_i, y_i)^T$ the position and ψ_i the orientation of vehicle i, and let their collected vectors be ξ and ψ, respectively.

Desired position and orientation constraints. The position of vehicle i is forced to ξ_d by the constraint

$$C_p(\xi) = \xi_i - \xi_d = 0. \tag{9}$$

For at least three times differentiable desired path $\xi_d(t)$ we have

$$C_{tt}(\xi) = \xi_i - \xi_d(t) = 0. \tag{10}$$

Similarly, for orientation constraints it yields

$$C_o(\psi) = \psi_i - \psi_d = 0, \quad C_{o,tt}(\psi) = \psi_i - \psi_d(t) = 0. \tag{11}$$

Distance constraints. If the distance r_{ij} should be satisfied between vehicle i and j then the appropriate constraint is

$$C_{rd}(\xi) = (\xi_i - \xi_j)^T(\xi_i - \xi_j) - r_{ij}^2 = 0. \tag{12}$$

Fixed relative position and orientation constraints. For prescribed relative position and orientation between two vehicles the constraints are

$$C_{fp}(\xi) = \xi_i - \xi_j - p_{ij} = 0, \quad C_{fo}(\psi) = \psi_i - \psi_j - o_{ij} = 0. \tag{13}$$

Combined constraints. If for example C_{rd} and C_p are two constraints to which the constraint forces are $W_{rd}^T\lambda_{rd}$ and $W_p^T\lambda_p$ then they can be combined to

$$W^T\lambda = [W_{rd}^T\, W_p^T] \begin{bmatrix} \lambda_{rd} \\ \lambda_p \end{bmatrix}. \tag{14}$$

Formation topology. Typical formation specifications can be converted to a resulting constraint set by using the above steps and their combinations. We shall assume that redundant constraints have already been omitted and there are no contradictions amongst the constraints which means that the resulting W has full row rank.

Master vehicle and followers. We can specify a master vehicle for which the desired path and path velocity will be designed. Specifications for the other vehicles can be derived from them if the formation type is chosen. Typical formations may be longitudinal, transversal, V-shaped and circular ones.

If $x_r(t), y_r(t)$ is the desired reference path for the master vehicle then its desired reference orientation can be determined by $\psi_r(t) = \arctan 2(\dot{y}_r(t), \dot{x}_r(t))$. Denoting the relative position of vehicle i to the master vehicle by p_{x_i}, p_{y_i} then the following constraints have to be introduced:

$$C_{m,tt}(\eta) = \begin{bmatrix} x_m - x_r(t) \\ y_m - y_r(t) \\ \psi_m - \psi_r(t) \end{bmatrix}, \quad C_{i,tt}(\eta) = \begin{bmatrix} x_i - x_r(t) - p_{x_i}(t) \\ y_i - y_r(t) - p_{y_i}(t) \\ \psi_i - \psi_r(t) \end{bmatrix} \; i \neq m. \tag{15}$$

$$C(\eta) = \begin{bmatrix} C_{m,tt}^T(\eta) & \cdots & C_{i,tt}^T(\eta) & \cdots \end{bmatrix}^T. \tag{16}$$

If W denotes the Jacobian of C then it yields

$$W^T\lambda = \begin{bmatrix} W_{m,tt}^T & \cdots & W_{i,tt}^T & \cdots \end{bmatrix} \begin{bmatrix} \lambda_{m,tt}^T & \cdots & \lambda_{i,tt}^T & \cdots \end{bmatrix}^T. \tag{17}$$

Since in each row of C appears only a single variable hence in case of the above convention $W = I$ and $\dot{W} = 0$ which simplifies the computations.

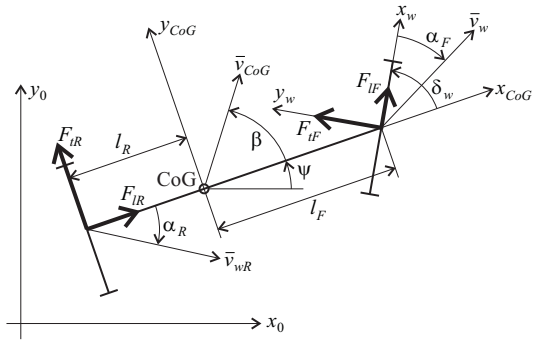


Figure 1
Simplified sketch of a single ground vehicle

4 Multi-body approach for underactuated UGVs in formation

The high level formation control design is a complex problem. Therefore, as usual in the literature, we apply the single-track vehicle model in order to make the computations easier and more efficient. The simplified sketch of a single car-like ground vehicle moving in horizontal plane is shown in Fig. 1.

4.1 Simplified dynamic model of a single vehicle

The dynamic model of a single ground vehicle can be written in the form

$$\begin{array}{lll} \dot{x} = v\cos(\phi) & \dot{\phi} = \frac{a_{11}}{v}\beta + \frac{a_{12}}{v^2}r + \frac{b_1}{v}\delta_w & \dot{r} = a_{21}\beta + \frac{a_{22}}{v}r + b_2\delta_w \\ \dot{y} = v\sin(\phi) & \dot{\beta} = \frac{a_{11}}{v}\beta + \left(\frac{a_{12}}{v^2} - 1\right)r + \frac{b_1}{v}\delta_w & \dot{v} = \alpha \end{array} \tag{18}$$

where $q_1 = (x,y)^T$ is the position, v is the absolute value of the velocity, ψ is the orintation, β is the side slip angle, $\phi = \beta + \psi$, δ_w is the steering angle and α is the longitudinal acceleration, see [2]. Here we used the notations

$$\begin{array}{lll} a_{11} = -\frac{c_F + c_R}{m_v} & a_{12} = \frac{c_R l_R - c_F l_F}{m_v} & b_1 = \frac{c_F}{m_v} \\ a_{21} = \frac{c_R l_R - c_F l_F}{I_z} & a_{22} = -\frac{c_R l_R^2 + c_F l_F^2}{I_z} & b_2 = \frac{c_F l_F}{I_z} \end{array} \tag{19}$$

where m_v is the mass, I_z is the inertia moment of the vehicle and c_F, c_R are the cornering stiffnesses assumed to be constant.

For heavy-duty cars we assumed $a_{11} = -147.1481$, $a_{12} = 0.0645$, $a_{21} = 0.0123$, $a_{22} = -147.1494$, $b_1 = 66.2026$ and $b_2 = 31.9835$, all in standard SI units.

With $\bar{x} = (x,y,\phi,\beta,r,v)^T$ and $u = (\delta_w,\alpha)^T$ the system can be brought to the parameter dependent input affine form $\dot{\bar{x}} = \bar{A}(x,\rho) + \bar{B}(x,\rho)u$ where $\rho = (v,v^2)^T$ is the parameter vector.

4.2 Generalization of multi-body interpretation for UGVs

Since each vehicle is underactuated and the number of input signals is two, we can prescribe only limited type of constraints. From physical consideration, the variables for which constraints can be specified, will be the position coordinates x, y. Hence, we omit orientation parts from the set of constraints. However, if the position follows its prescribed path and the side slip angle is zero (except for short transients), then the vehicle's velocity is parallel to the tangent of the path and its orientation is acceptable. For small side slip angle β the assumption is fulfilled.

Choosing new state variables according to $q_1 = (x,y)^T$, $q_2 = (\dot{x},\dot{y})^T$ and $q_3 = (\beta, r)^T$, and applying the usual notation in robotics $C_\phi = \cos(\phi)$ and $S_\phi = \sin(\phi)$, then we obtain

$$\dot{q}_1 = \begin{bmatrix} vC_\phi \\ vS_\phi \end{bmatrix} \tag{20}$$

$$\dot{q}_2 = \begin{bmatrix} \dot{v}C_\phi - vS_\phi\dot{\phi} \\ \dot{v}S_\phi + vC_\phi\dot{\phi} \end{bmatrix} = \begin{bmatrix} \alpha C_\phi - vS_\phi[(a_{11}/v)\beta + (a_{12}/v^2)r + (b_1/v)\delta_w] \\ \alpha S_\phi + vC_\phi[(a_{11}/v)\beta + (a_{12}/v^2)r + (b_1/v)\delta_w] \end{bmatrix} \tag{21}$$

from which follows $\ddot{q}_1 = P(\phi)z + Q(\phi)\tau$, where $z = (\beta, r/v)^T$, $\tau = (\delta_w, \alpha)^T$ and

$$P(\phi) = \begin{bmatrix} -a_{11}S_\phi & -a_{12}S_\phi \\ a_{11}C_\phi & a_{12}C_\phi \end{bmatrix}, \quad Q(\phi) = \begin{bmatrix} -b_1S_\phi & C_\phi \\ b_1C_\phi & S_\phi \end{bmatrix}. \tag{22}$$

The same is valid for each vehicle. Denoting the appropriate terms for vehicle i by q_{1i}, $\dot{q}_{1i}$, z_i, τ_i, P_i, Q_i and collecting them in the vectors q_1, $\dot{q}_1$, z, τ and in the blockdiagonal matrices P, Q, respectively, then we can generalize the original method for UGVs. Hence

$$W\ddot{q}_1 + \dot{W}\dot{q}_1 = \ddot{C} \tag{23}$$

$$W[Pz + Q\tau] + \dot{W}\dot{q}_1 = \ddot{C} \tag{24}$$

$$\tau = -W^T\lambda \tag{25}$$

$$W[Pz - QW^T\lambda] + \dot{W}\dot{q}_1 = \ddot{C} \tag{26}$$

$$WQW^T\lambda = WPz + \dot{W}\dot{q}_1 - \ddot{C} \tag{27}$$

from which it follows

$$\lambda = (WQW^T)^{-1}(WPz + \dot{W}\dot{q}_1 - \ddot{C}). \tag{28}$$

The inverse of WQW^T exists since $\det Q = -b_1 \neq 0$ and W has full row rank.

The constraint force for the ith vehicle is

$$\begin{aligned} \tau_{ci} \quad = \quad & \sum_{k\in A_c^i}\sum_{j\in B_k} -W_{ki}^T(W_kQ_{ij}W_k^T)^{-1}\times \\ & (W_{ki}P_{ij}z_{ij} + \dot{W}_{ki}\dot{q}_{1,ij} + K_{d,ki}\dot{C}_{ki} + K_{p,ki}C_{ki}) \end{aligned} \tag{29}$$

where A_c^i, B_k and W_{ki} are defined as earlier and the environmental force is assumed to be zero.

4.3 Stability considerations

The feedback loop works as follows. Each vehicle determines its state $\bar{x}_i = (x_i, y_i, \phi_i, \beta_i, r_i, v_i)^T$ and computes $\psi_i = \phi_i - \beta_i$, $z_i = (\beta_i, r_i/v_i)^T$, $P_i(\phi_i)$, $Q_i(\phi_i)$, $q_{1i} = (x_i, y_i)^T$ and $\dot{q}_{1i} = (\dot{x}_i, \dot{y}_i)^T$. The composite vectors q_1, $\dot{q}_1$, z and the blockdiagonal matrices P, Q are formed, then the constraint C, its Jacobian W and the derivative $\dot{C} = W\dot{q}_1$ will be computed. For each vehicle i the force $\tau_{c,i}$ will be determined using formation information and Eq. (29). Finally $\tau_{c,i} = (\delta_{w,i}, \alpha_i)^T$ will be applied as steering angle and acceleration for the vehicle. Thus the feedback loop is closed.

The constraints determine a manifold M_C. The choice of K_d, K_p and $\ddot{C} + K_d\dot{C} + K_pC = 0$ assures that the system moves on the manifold satisfying global exponential stability (GES).

However the system of UGVs is underactuated therfore zero dynamics is present. The stability of the zero dynamics was proved in [2].

5 Realistic dynamic modeling and control using robotic formalism

5.1 Geometric topology of 16 DoF ground vehicle

Consider a tree-structured mechanical system assembled by rigid bodies B_j for $j = 1, \ldots, n$, i.e. numbered from the base body to the terminals. A body can be virtual or real: virtual bodies are introduced to describe joints with multiple degrees of freedom such as ball joints or intermediate fixed frames.

Frame K_j, associated with body B_j, is given by its origin and an orthonormal basis (x_j, y_j, z_j). Transformation between two consecutive frames K_i and K_j is performed by the modified Denavit-Hartenberg formalism mand can be described by the homogeneous transformation, see [11]:

$$ {}^iT_j = \begin{bmatrix} {}^iA_j & {}^ip_j \\ 0_{1\times 3} & 1 \end{bmatrix} $$

where ${}^iA_j$ defines the (3×3) rotation matrix and ${}^ip_j$ is the (3×1) vector describing the position of the origin of K_j with respect to K_i. The generalized coordinate of the jth joint connecting B_{j-1} and B_j is defined as follows:

$$ q_j = \bar{\sigma}_j\theta_j + \sigma_j r_j, \quad \bar{\sigma}_j = 1 - \sigma_j \tag{30} $$

where σ_j is 0 for rotational joints and 1 for translational joints. In the case of fixed frames attached to the same body, no joint variable is used.

The vehicle is considered as a mobile robot interconnected by joints, see Fig. 2, and modeled as a multi-body system consisting of 10 actuated and 10 virtual bodies similar to [13]. In the sequel, front steered and rear axle driven vehicle is assumed. Notice that for example $x_{4,5}$ means that the axes x_4 and x_5 are equivalent. The different joint variables are denoted by q_j.

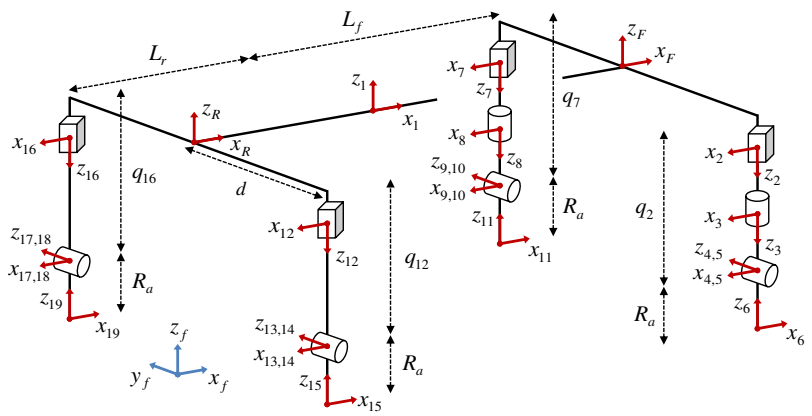


Figure 2
Model of 16 DoF ground vehicle

The 16 degree of freedom (DoF) model of the vehicle incorporates a 6 DoF $(x, y, z, \phi, \theta, \psi)$ chassis (K_1) as the base, two front steering wheels (q_3, q_8) which can be rotated about the front vertical axes, four suspensions $(q_2, q_7, q_{12}, q_{16})$ connected to the chassis by vertical translational joints and four driving wheels $(q_5, q_{10}, q_{14}, q_{18})$. Notice that for rear axle driven vehicles (q_5, q_{10}) are not actuated.

5.2 Novel Appell Formalism for Tree Structured Systems

There exist several equivalent methods for mass-point systems (Newton-Euler, Lagrange, Appell) based on the common assumption that the sum of internal forces and the sum of the moment of internal forces to any point are zero which can be extended to rigid multi-body systems, see Section A.4 in [2]. Each method tends to derive the dynamic model in vector form of

$$\tau = M(q)\ddot{q} + h(q, \dot{q}) \tag{31}$$

where $M(q)$ is the generalized inertia matrix and the effects of the centrifugal, Coriolis, gravity and external forces are contained in $h(q, \dot{q})$.

In our approach, Appell's method is chosen which uses the concept of acceleration energy or more precisely, the Gibbs function [2, 14] and eliminates a large number of numerical steps of the Newton-Euler technique. The algorithm directly computes the dynamic model of the composite system without the need of differentiation by time as in the Lagrange formalism. In case of numerical computations the complexity of the methods is different. Using symbolic computations, these differences play no more role.

The dynamic model by Appell's method reads as

$$\frac{\partial G}{\partial \ddot{q}_j} + \frac{\partial P}{\partial q_j} = \tau_j \tag{32}$$

where P is the potential energy (gravity effect) of the segment, q_j is the generalized variable in the direction of the generalized force τ_j.

In order to obtain the dynamic model in the vector form $M(q)\ddot{q}+h(q,\dot{q})=\tau$ and simplify the use of Appell's method, the acceleration and angular acceleration of the segments will be computed in the compatible form

$$a_j=\Omega_j(q)\ddot{q}+\theta_j(q,\dot{q}),\quad \varepsilon_j=\Gamma_j(q)\ddot{q}+\phi_j(q,\dot{q}). \tag{33}$$

Moving from the root to the terminal of a branch in the tree structure, the matrices and vectors of the kinematic model can be computed in forward recursion. Let the antecedent of segment j be i and the efficient dimension of q will be increased by 1 in each step, then

$$\begin{aligned}
\Gamma_j &= \left[\ {}^iA_j^T\Gamma_i \quad |\,\bar{\sigma}_j(0,0,1)^T\ \right],\ \omega_j=\Gamma_j\dot{q}\\
\phi_j &= {}^iA_j^T\phi_i+\bar{\sigma}_j\left(\omega_{y,j},-\omega_{x,j},0\right)^T\dot{q}_j\\
\Omega_j &= \left[\ {}^iA_j^T(\Omega_i-[{}^ip_j\times]\Gamma_i) \quad |\,\sigma_j(0,0,1)^T\ \right]\\
\theta_j &= {}^iA_j^T\{\theta_i+([\phi_i\times]+[\omega_i\times]^2){}^ip_j)\}+\sigma_j 2\left(\omega_{y,j},-\omega_{x,j},0\right)^T\dot{q}_j.
\end{aligned}$$

After some conversions the matrix and vector portions of a single segment of the dynamic model can be written in form of

$$\begin{aligned}
M_s(q) &= \left[\ \Omega^T \quad \Gamma^T\ \right]\begin{bmatrix} mI_3 & -[m\rho_c\times] \\ [m\rho_c\times] & J\end{bmatrix}_s\begin{bmatrix}\Omega\\ \Gamma\end{bmatrix}\\
h_s(q) &= \left[\ \Omega^T \quad \Gamma^T\ \right]\begin{bmatrix} \theta m+\phi\times m\rho_c+[\omega\times][\omega\times]m\rho_c \\ J\phi-\theta\times m\rho_c+\omega\times(J\omega)\end{bmatrix}_s
\end{aligned}$$

where the matrices Ω, Γ are already the concatenated ones, e.g. Ω is of type $3n\times n$ and Γ is of type $3n\times n$.

The dynamic model (31) is the sum of the above portions if the index s goes from the root to the terminals of the branches because the Gibbs function is additive. After these extensions both composite matrices Ω and Γ have $(6+n_q)$ columns and $3\cdot 21$ rows (see the number of frames in Fig. 2).

5.3 Kinematic constraints

Composite variables are defined to collect the parameters of the 6 DoF moving base and the generalized coordinates of the vehicle

$$q_{EL}=[x,y,z,\varphi,\theta,\psi,q_2,\ldots,q_{18}]^T. \tag{34}$$

In order to keep the vehicle in the ground, kinematic constraints are introduced that express the nullity of vertical accelerations at the contact points in the reference frame

$$({}^fa_{6z},\,{}^fa_{11z},\,{}^fa_{15z},\,{}^fa_{19z})^T=J_4\ddot{q}_{EL}+\Psi=0_{4\times 1}. \tag{35}$$

Adding the above constraint equations to the result of Eq. (31) in form of Lagrange multipliers (λ) and assuming that no external forces act on the moving base, the direct dynamic model becomes

$$\begin{pmatrix} \ddot{q}_{EL} \\ \lambda \end{pmatrix} = \begin{bmatrix} M_{16\times16} & J_4^T \\ J_4 & 0_{4\times4} \end{bmatrix}^{-1} \begin{pmatrix} \tau - h \\ -\Psi \end{pmatrix} \tag{36}$$

where λ represent the constraint forces to maintain the contact points of the wheels on the ground and the first six components of τ are zero, see [12]. System (36) can be reformulated into a set of ordinary differential equations (ODEs) with constraints by using the well known differentiation rules $a = \dot{v} + \omega \times v$ and $\varepsilon = \dot{\omega}$ in moving frames.

5.4 External forces

Normal ($F_{z,j}$), lateral ($F_{y,j}$) and longitudinal ($F_{x,j}$) forces at the wheel/ground contact points are taken into account in the right hand side of the dynamic model by projecting them to the base by the corresponding Jacobian matrix. Normal forces can be computed from the dynamic load distribution as described in [15] for $j = \{6, 11, 15, 19\}$ respectively:

$$F_{z,j} = \frac{m}{L_f + L_r}(gL^* + \eta h^1 a_x)\left(\frac{1}{2} + \nu\frac{h^1 a_y}{2gd}\right) \tag{37}$$

$$\eta = \{-1, -1, 1, 1\}, \quad \nu = \{-1, 1, -1, 1\} \tag{38}$$

where g is the gravity constant, m is the total mass of the vehicle, η and ν are wheel selectors and L^* defines the static equilibrium point along the x-axis (front and rear are different) based on $\rho_{1c,x}$ and h is the height of the center of mass above the road surface in K_f. Since this formula does not take into consideration $\rho_{1c,y}$ we have developed corrections for it similarly to [9].

The longitudinal and lateral wheel forces are described by Pacejka's model [16] and given by the same formula, with different coefficients, in function of the longitudinal and lateral slip

$$F_{x,y} = D\sin(C\text{atan}(B\alpha - E(B\alpha - \text{atan}(B\alpha))))$$

$$\alpha_{x,j} = -({}^1v_{x,j} - R_a\dot{q}_j)/(\max\{{}^1v_{x,j}, R_a\dot{q}_j\})$$

$$\alpha_{y,j} = -\kappa q_i - \arctan\left(\frac{{}^1v_{y,j}}{{}^1v_{x,j}}\right), \quad j = \{6, 11, 15, 19\}$$

where $i = \{3, 8\}$ for the right and left front wheels and κ is zero for the rear wheels. The total torques acting on the driven wheels consist of the active actuator torques and the passive longitudinal wheel forces

$$\tau_j = \tau_{aj} - R_a F_{x,j}, \quad j = \{14, 18\}. \tag{39}$$

6 Low level decentralized control system based on realistic vehicle model

6.1 Reference signals

Reference signals for vehicles can be derived from the acceleration pedal and the steering wheel changes, respectively. It will be assumed that they are variables for the desired longitudinal velocity and the derivative of the steering angle (whose integral is the steering angle). However, for control purposes in many cases a strategy has to be elaborated to find their time derivatives, i.e. the reference acceleration and the steering angle acceleration. Of course, if the reference signals and their derivatives are designed analytically, then this step can be omitted.

Denote with $\dot{x}_{\text{ref}}$ any variable to be differentiated further by the time and assume an approximating linear model for the output $\dot{x}$ in the form of $G_{\dot{x},u}(s) = \frac{1}{s(1+s\tau)}$. A fictitious feedback system can be designed with PI controller $G_{u,e}(s) = k_p + \frac{k_i}{s}$, error signal $e = \dot{x}_{\text{ref}} - \dot{x}$ and open loop $G_o(s) = (k_p s + k_i)\frac{1}{s^2(1+s\tau)}$ and closed loop transfer function

$$G_{\dot{x},\dot{x}_{\text{ref}}}(s) = \frac{G_0(s)}{1+G_0(s)} = \frac{k_p s + k_i}{\tau s^3 + s^2 + k_p s + k_i}. \tag{40}$$

Then the derivative $\ddot{x}_{\text{ref}}$ can be approximated by the output of the controller of this fictitious closed loop system, e.g.

$$\ddot{x}_{\text{ref}} \approx \frac{k_p s^2 + k_i s}{\tau s^3 + s^2 + k_p s + k_i}. \tag{41}$$

Notice, that τ is responsible for the precision of the approximation. Based on the different forms of the root locus, for high speed approximations $\tau = 0.005$sec, $k_i/k_p = 0.25$ and $k_p = 100$ were chosen for the applications.

This method was applied for finding both $\dot{v}_{\text{ref}}$ and $\ddot{\delta}_{\text{ref}}$.

6.2 Active suspension control

The vertical movement for passive suspensions can be taken into account as an elasticity model of

$$\tau_{ej} = k_j(q_j - q_{j0}) + F_{\text{offs},j} + d_j\dot{q}_j, \quad j = \{2,7,12,16\} \tag{42}$$

with stiffness k_j and damping d_j. The displacement of the suspensions is also influenced by the initial offset $F_{\text{offs},j}$.

An important problem is the determination of the resulting total mass and center of gravity from the first moments of the chassis and the four wheel branches in steady state. Using their values and (37)-(38) the load forces can be determined.

Then from them and the stiffness values of the suspension the vertical movements of the translational joints q_{j0} and their average value $q_{0,\text{avg}}$ can be determined. The prescribed height of K_1 above the road surface was chosen as $z_0 = R_a + q_{0,\text{avg}}$ and the offsets $F_{\text{offs},j}$ were computed to it. The passive suspension forces can be computed from them by (42).

Beside the passive suspension PID type *active suspension* was applied according to

$$\tau_j = K_P(z_0 - R_a - q_j) - K_D\dot{q}_j + K_I \int (z_0 - R_a - q_j)dt,$$
$$K_P = 100000, \quad K_D = 10000, \quad K_I = 300000.$$

6.3 Speed control

For each actively driven wheel yields component-wise $\tau = \theta\ddot{q}$ where θ denotes the resulting inertia moment of the axis. The usual choice is $\dot{q}_{v,\text{ref}} = v_{\text{ref}}/R_a$. Hence, the PID control law

$$\tau := \theta\left[k_{i,v}(q_{v,\text{ref}} - q) + k_{p,v}(\dot{q}_{v,\text{ref}} - \dot{q}) + \ddot{q}_{v,\text{ref}}\right] \tag{43}$$

can be suggested, from which with $e := q_{v,\text{ref}} - q$ follows the error differential equation and from it the characteristic equation

$$\ddot{e} + k_{p,v}\dot{e} + k_{i,v}e = 0 \Rightarrow s^2 + k_{p,v}s + k_{i,v} = 0. \tag{44}$$

With the choice of $(1+sT)^2 = 0$, it follows that $k_{p,v} = \frac{2}{T}$ and $k_{i,v} = \frac{1}{T^2}$ are satisfactory for the closed loop stability.

6.4 Steering control

For active steering a similar concept was chosen as for speed control, however now $\dot{q}_{D,\text{ref}} = \dot{\delta}_{w,\text{ref}}$ and its integral and derivative are $q_{D,\text{ref}}$ and $\ddot{q}_{D,\text{ref}}$, respectively. The PID control law and controller parameters were chosen similarly to active speed control.

7 Simulation results of high level control of UGV formations

Efficiency of the robust and high-speed low level control system is presented in another paper [17]. Hence, only the high level control of UGV formations is discussed here. The high level system produces reference signals for the low level system in form of acceleration (α) and steering angle (δ_w) for each vehicle of the formation. In the sequel simulation results will be presented using MATLAB/Simulink.

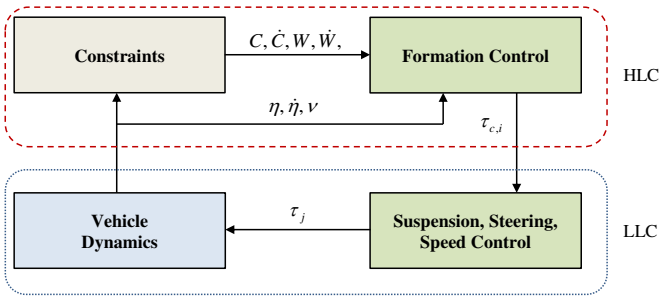


Figure 3
Structure of the hierarchical control system using multi-body interpretation

7.1 Software system

A software system was elaborated for the investigation of formation control of unmanned ground and marine vehicles using a broad field of methods [18]. The method based on potential function can be applied for UGVs and UMVs.

Synchronized path following was implemented only for surface ships in formation because this method is based on the dynamic model in strictly feedback form which is not valid for UGVs.

Formation control based on multi-body implementation was elaborated both for full-actuated ships and underactuated car-like vehicles. The methods allow the investigation of different types of formations, amongst horizontal, vertical, V-shaped and circular ones. The formation can be dynamically changed during the experiment.

The software has a graphical user interface in which the control method, the number of vehicles, their groups, the initial positions/orientations and the parameters of the paths, vehicles and controllers can be easily formulated. After the simulation all the states, control and other signals can be drawn and the motion of the formation is presented in animation.

From the simulation results we present here only the formation control of UGVs based on multi-body method in varying formations. For the constraints manifold $k_{di} = k_{pi} = 150$ was chosen. The structure of the control system based on multi-body interpretation is shown in Fig. 3.

7.2 Sine-shaped paths

The master vehicle has index 1 and its orientation is $\psi_r = \arctan 2(A\omega\cos(\omega t), 1)$, where A is the magnitude and ω is the angular velocity of the master's path. In the experiments heavy-duty cars are used therefore $A = 100$m, $\omega = 0.02$rad/sec and $D = 12.5$m are assumed.

In *horizontal formation* the reference path of the i-th vehicle is sin-shaped according to

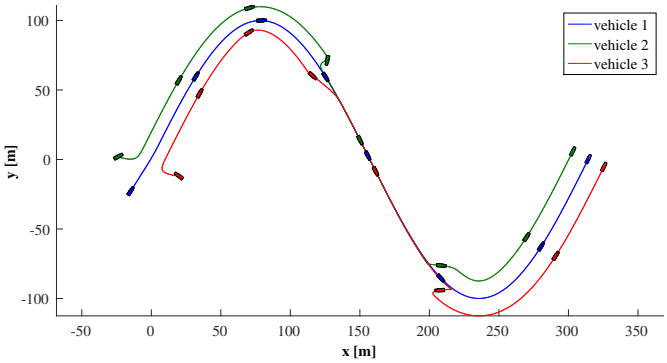


Figure 4
Realized motion of UGVs along sin-shaped path in varying formations

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \begin{pmatrix} t + d_i \cos(\psi_r) \\ A\sin(\omega t) + d_i \sin(\psi_r) \end{pmatrix}$$

where $d_i = (-1)^{i-1} \lfloor i/2 \rfloor D$ is the distance from the master and D is the relative distance between the vehicles. Vehicles having odd index are before the master, the others are behind the master. The formation is tangential to the master's path.

Vertical formation is orthogonal to the tangent of the master's path. Vehicles having odd index are to the right from the master, while vehicles having even index are to the left from the master. The path is

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \begin{pmatrix} t + d_i \sin(\psi_r) \\ A\sin(\omega t) - d_i \cos(\psi_r) \end{pmatrix}$$

where d_i is as for horizontal formation.

V-shaped formation has path

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \begin{cases} \begin{pmatrix} t - d_i \sin(\pi/4 - \psi_r) \\ A\sin(\omega t) - d_i \cos(\pi/4 - \psi_r) \end{pmatrix} & i = 2k+1 \\ \begin{pmatrix} t - d_i \cos(\pi/4 - \psi_r) \\ A\sin(\omega t) + d_i \sin(\pi/4 - \psi_r) \end{pmatrix} & i = 2k \end{cases}$$

where $d_i = \lfloor i/2 \rfloor D$ is the distance between master and follower. The wing angle of the V-shape is $\pi/4$ relative to the tangent of the master's path. Vehicles having odd index are to the right from the master, the others are to the left.

Simulation results for three UGVs along sin-formed paths in varying V-shaped, horizontal and vertical formations are shown as follows. Fig. 4 presents the realized paths for the varying formations using multi-body approach. Snap-shot of the control signals along sin-formed paths are shown in Fig. 5. The snap-shot of the side slip angles along sin-formed paths can be seen in Fig. 6 illustrating that β is small except transients belonging to larger curvatures.

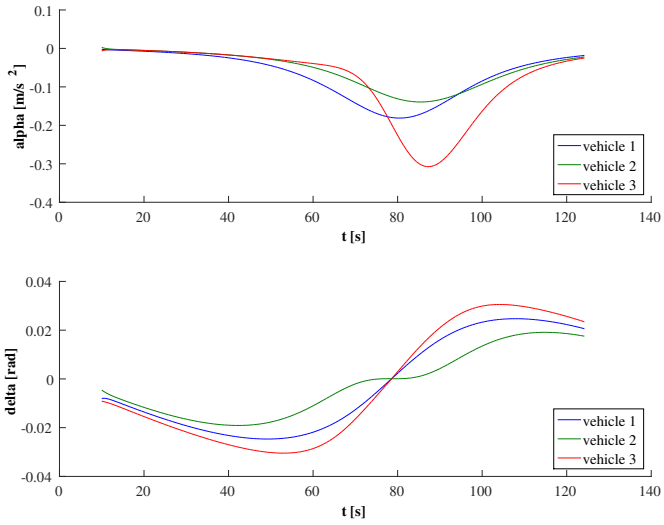


Figure 5
Snap-shot of the control signals on V-shaped section along sin-formed path

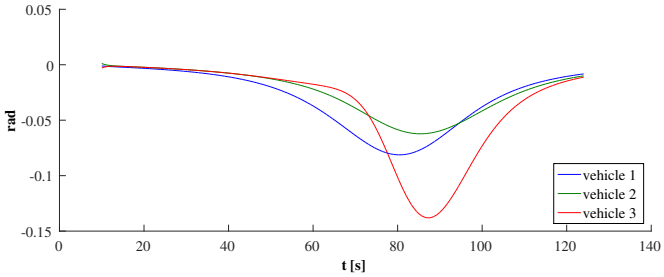


Figure 6
Snap-shot of the side-slip angles on V-shaped section along sin-formed path

7.3 Circular paths

The master vehicle has index 1 and A and ω are as for the sin-formed path, however the orientation is $\psi_r = mod(\omega t + \pi/2,\ 2\pi/\omega)$.

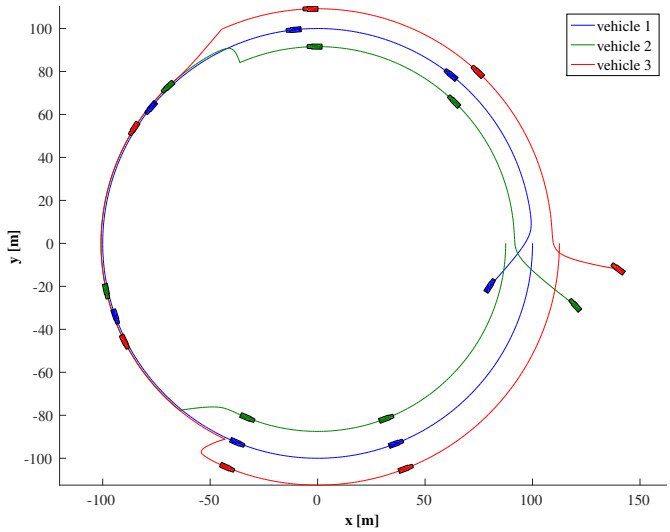


Figure 7
Realized motion of UGVs along circular paths in varying formations

In *horizontal formation* the circular path is defined by

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \begin{pmatrix} A\cos(\omega t) + d_i\cos(\psi_r) \\ A\sin(\omega t) + d_i\sin(\psi_r) \end{pmatrix}$$

where $d_i = (-1)^{i-1}\lfloor i/2 \rfloor D$ and D is the relative distance between the vehicles. The formation is tangential to the master's path. Vehicles having odd index are before the master, the others are behind the master.

In *vertical formation*

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \begin{pmatrix} A\cos(\omega t) + d_i\sin(\psi_r) \\ A\sin(\omega t) - d_i\cos(\psi_r) \end{pmatrix}$$

where $d_i = (-1)^{i-1}\lfloor i/2 \rfloor D$. The formation is orthogonal to the master's path. Vehicles having odd index are to the right from the master, the others are to the left.

In *V-shaped formation* the wing angle of the V-shape is $\pi/4$ relative to the tangent of the master's path. Vehicles having odd index are to the right from the master, the others are to the left. The position is defined by

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \begin{cases} \begin{pmatrix} A\cos(\omega t) - d_i\sin(\pi/4 - \psi_r) \\ A\sin(\omega t) - d_i\cos(\pi/4 - \psi_r) \end{pmatrix} & i = 2k+1 \\ \begin{pmatrix} A\cos(\omega t) - d_i\cos(\pi/4 - \psi_r) \\ A\sin(\omega t) + d_i\sin(\pi/4 - \psi_r) \end{pmatrix} & i = 2k \end{cases}$$

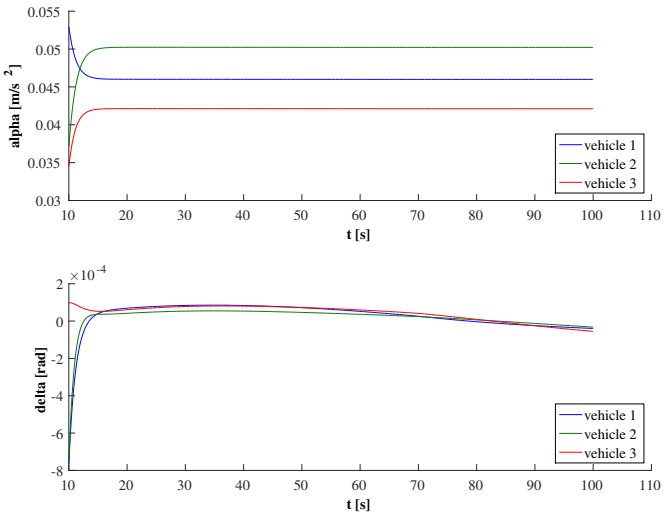


Figure 8
Snap-shot of the control signals on V-shaped section along circular path

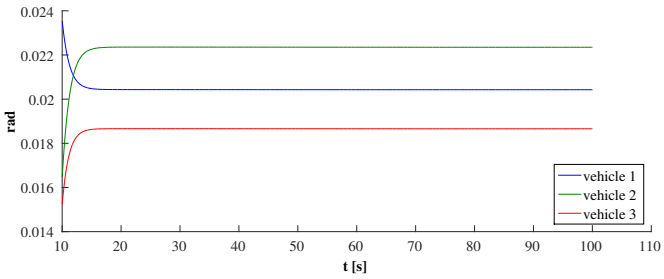


Figure 9
Snap-shot of the side-slip angles on V-shaped section along circular path

where $d_i = \lfloor i/2 \rfloor D$ is the distance between master and follower.

Simulation results for three UGVs along circular paths in varying V-shaped, horizontal and vertical formations are shown as follows. Fig. 7 presents the realized paths for the varying formations using multi-body approach. Snap-shot of the control signals along circular paths are shown in Fig. 8. The snap-shot of the side slip angles along circular paths can be seen in Fig. 9 illustrating that β is small except transients belonging to larger curvatures.

8 Conclusions

A hierarchical control system has been elaborated for the formation control of UGVs moving in horizontal plane. The control system consists of the high level centralized formation control of the UGVs and the low level decentralized PID type suspension,

speed and steering control of the different vehicles. Both problems were discussed in multi-body assumptions.

The formation control method, developed originally for fully-actuated ships, was generalized for underactuated car-like vehicles. Multi-body theory at this level used the result of Lanczos and the method of Lagrange multipliers. Formation specifications were formulated as constraints containing position, orientation and distance prescriptions. In order to simplify the design and implementation on formation level, approximate single track dynamic model was assumed for each vehicle.

At low level a more realistic two track dynamic model is used in the form of a multi-body system in tree structure. This realistic nonlinear model is obtained by using Appell's method, Pacejka's magic formula for tyre-road connections and kinematic constraints expressing the nullity of vertical accelerations of the contact points. The interface between the higher and lower control levels is presented in the form of acceleration and steering angle prescriptions (output of high level). At low level the decentralized control system of each vehicle converts the specifications in smooth reference signals and performs the desired motion.

Simulation results of the high level control of UGV formations were presented for sine-shaped and circular paths. The Simulation results demonstrate the applicability of the multi-body approach for car-like UGVs.

Detailed simulation results for low level vehicle control based on PID type suspension, speed and steering controllers can be found in another paper [17].

Further researches are in progress to check the method under real-time conditions and state estimation based on the fusion of GPS and IMU. Further directions may be the elaboration of real-time multi-body approach for unmanned indoor quadrotor helicopters.

References

[1] T. Peni, "Feedback design methods for cooperative and constrained control problems," Ph.D. dissertation, Budapest University of Technology and Economics, 2009.

[2] B. Lantos and L. Márton, *Nonlinear Control of Vehicles and Robots*. Springer, 2011.

[3] I.-A. F. Ihle, M. Arcak, and T. I. Fossen, "Passivity-based designs for synchronized path-following," *Automatica*, vol. 43, no. 9, pp. 1508 – 1518, 2007.

[4] I.-A. F. Ihle, "Coordinated control of marine craft," Ph.D. dissertation, Norwegian University of Science and Technology, 2006.

[5] Y. Zou, P. R. Pagilla, and R. T. Ratliff, "Distributed Formation control of multiple aircraft using constraint forces," 2008, pp. 644–649.

[6] D. Tilbury, O. Sordalen, L. Bushnell, and S. Sastry, "A multisteering trailer system: conversion into chained form using dynamic feedback," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 6, pp. 807–818, 1995.

[7] V. Cossalter, A. Doria, and R. Lot, "Steady turning of two-wheeled vehicles," *Vehicle System Dynamics*, vol. 31, no. 3, pp. 157–181, 1999.

[8] S. Maakaroun, W. Khalil, M. Gautier, and P. Chevrel, "Modeling and simulating a narrow tilting car using robotics formalism," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 1026–1038, June 2014.

[9] A. Rucco, G. Notarstefano, and J. Hauser, "Development and numerical validation of a reduced-order two-track car model," *European Journal of Control*, vol. 20, no. 4, pp. 163 – 171, 2014.

[10] W. Khalil, A. Vijayalingam, B. Khomutenko, I. Mukhanov, P. Lemoine, and G. Ecorchard, "OpenSYMORO: An open-source software package for symbolic modelling of robots," in *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*, July 2014, pp. 1206–1211.

[11] W. Khalil and J.-F. Kleinfinger, "Minimum operations and minimum parameters of the dynamic models of tree structure robots," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 6, pp. 517–526, 1987.

[12] C. Lanczos, *The Variational Principles of Mechanics*. Dover Publications, 1986.

[13] S. Maakaroun, "Modélisation et simulation dynamique d'un véhicule urbain innovant en utilisant le formalisme de la robotique," Ph.D. dissertation, Ecole des Mines de Nantes, 2011.

[14] J. Somlo, B. Lantos, and P. T. Cat, *Advanced Robot Control Advances in Electronics 14*. Budapest, Hungary: Akademiai Kiado, 1997.

[15] U. Kiencke and L. Nielsen, *Automotive Control Systems: For Engine, Driveline, and Vehicle*. Springer, 2005.

[16] H. B. Pacejka, *Tyre and Vehicle Dynamics*, ser. Automotive engineering. Butterworth–Heinemann, 2006.

[17] G. Max and B. Lantos, "Active Suspension, Speed and Steering Control of Vehicles Using Robotic Formalism," in *Proceedings of IEEE 16th International Syposium on Computational Intelligence and Informatics CINTI 2015*, Budapest, Hungary, 19–21 November 2015, to appear.

[18] G. Max, "Control of Vehicles Moving in Formation," Master's thesis, Budapest University of Technology and Economics, Hungary, 2011.

Extending AUV Response Robot Capabilities to Solve Standardized Test Methods

Bence Takács[1,3], Roland Dóczi[1], Balázs Sütő[2], János Kalló[1], Teréz Anna Várkonyi[1,3], Tamás Haidegger[1,3], Miklós Kozlovszky[2,4]

[1] John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary
[2] BioTech Research Center, Óbuda University, Budapest, Hungary
[3] Antal Bejczy Center for Intelligent Robotics, Óbuda University, Budapest, Hungary
[4] MTA SZTAKI/Laboratory of Parallel and Distributed Computing, Budapest, Hungary
bence.takacs@irob.uni-obuda.hu; suto.balazs@biotech.uni-obuda.hu; teri.varkonyi@irob.uni-obuda.hu; haidegger@irob.uni-obuda.hu; kozlovszky.miklos@nik.uni-obuda.hu

Abstract: Autonomous Underwater Vehicle (AUV) response robots are special multipurpose devices, capable of moving and performing various tasks in water, autonomously, or with human teleoperation. Capability assessment of such devices is hard and complex work. This paper describes our work in AUV Response Robot testing from two aspects: First, additional testing methods are proposed for AUV capability assessment and second, we describe, in detail, how an AUV can be enhanced to pass the existing underwater response robot tests, defined by National Institute of Standards and Technology (NIST). In the first part of the paper, a short overview of the existing AUV testing methods is given, followed by our proposed, new test scenarios. The second part covers a general overview about our system design and development, which enabled the custom, enhanced AUV to pass the test scenarios.

Keywords: autonomous underwater vehicles (AUV); response robotics; AUV testing; underwater manipulation; underwater teleoperation

1 Introduction

The field of Autonomous Robotics Research has increased tremendously, in popularity, over the last decade, for air, land and sea applications. Emergency response, Autonomous Underwater Vehicles (AUVs), can be equipped with a vast number of sensors and actuators, to be used for a broad range of applications.

Civilian and military use cases include, underwater mapping, structural inspection of power plants, localization of leaking underwater pipes or finding missing people after an accident. For these tasks, underwater navigation is essential. In most cases, CCD cameras are employed in the visual domain as sensors. Nevertheless, underwater conditions make it difficult to use normal camera systems with RGB color space for object detection. Additional problems arise from disturbances originated from the external environment, such as underwater lighting, reflection and ray scattering, high pressure and last, but not least, high conductivity of liquids [1].

1.1 Motivation

The motivation behind the research is twofold. First, there is the social drive: our research is mainly inspired by the need to remedy the consequences of industrial accidents (e.g., the Fukushima Daiichi accident in 2011). It is often required that underwater robots survey the scene, collect environmental data and to identify critical hazards. Such scenarios require complex task execution, realized through autonomous functions or by the means of teleoperation. Second, our team had a basic research interest in how to build up underwater response robots, working in a hazardous environment and how such robots are able to solve autonomously and effectively, complex tasks.

2 Standardization and Testing

The National Institute of Standards and Technology (NIST) has a strong reputation in standardization and testing in various domains. NIST also deals with complex cases, such as the evaluation of robotic platforms dedicated to search and hazmat operations. NIST's Robotics Test Facility – Building 207 – at the Gaithersburg campus, hosts a large number of robot test systems and artifacts (aka "props"), which are designed to be abstract representations of the targeted environments and tasks. The main mission of the facility is to foster the manufacturing and the deployment of advanced robotic systems through the development of performance testing methods (benchmarks), measurement capabilities and standards. Their work includes the assessment of joined sensors, intelligent behaviors, open-architecture controllers and high-fidelity simulation tools, summarized in the DHS NIST ASTM Robot Test Methods [2, 3].

The performance evaluation of mobile response robots has the following areas:

- Collaboration
- Autonomy
- Mapping and Planning

- Knowledge Representation
- Sensory Perception
- Human-Robot Interaction
- Locomotion.

Intelligent response robots utilize a variety of sensors that include actuators, navigation and driving systems and communication systems. Just like any other typical robot, there is a need for mission-specific devices/packages controlled by an intelligent controller/ remote teleoperation.

For underwater response robots, autonomy is a common requirement. The survivability of the robot in an ever-changing, harsh environment depends on accurate situational and environmental awareness, based on reliable sensor data acquisition, data fusion, data evaluation and behavior generation (decision support).

Our proposed addition to the existing testing methods focuses on the temporal variability of the environment. In most cases, the robots are tested only for static scenarios. Such tests can hardly grasp how a response robot is able to accommodate to a new, suddenly changing environmental condition. Static terrain mapping can be misleading, if the environment is changing over time drastically (e.g., when a building fire spreads out, parts of a building collapse, or a boat is sinking).

2.1 Proposed Additional Response Robot Testing Parameter Groups

Our proposed two parameter groups, to assess the adaptability to the dynamically changing environment (temporal awareness) of a robot are the following:

- **Temporal resolution** (sensing/sampling frequency)
  In many scenarios, the sampling frequency is an important factor. A good and simple example here is the real-time image acquisition, where fast moving objects are hardly recognizable if the frame rate is not high enough.
- **Information aging speed**
  In a rapidly changing environment, the acquired data for data fusion, data evaluation and decision support can become outdated within a short time. Old and inaccurate data cause wrong situational (environment, location, etc.) awareness, and can introduce less effective behaviors than just using pure blind guessing.

3 Underwater Response Robot Testing Methods

The test framework set up by NIST for unmanned underwater robots consists of various tasks [4]. In this paper, we are focusing on five selected domains from the aforementioned set:

1) Inspection/Station Keeping
2) Rope Cutting
3) Hooking
4) Soft Grab
5) Closing a valve

To accomplish these tasks, the manoeuvrability of a robot under trial should be precise and fine-tuned. Furthermore, smooth process controllers are needed for accurate positioning and depth tracking. The following examples are taken from tasks captured by a camera of an AUV during a NIST test execution.

3.1 Inspection/Station Keeping

This test measures the position keeping and the inspection capability of an AUV. During the task execution, there are various disruptive conditions, such as, turbidity or current. In order to compensate for these disturbances, typically, an underwater camera is installed on the robot. The objective of this task is to inspect cylinders on an underwater wall, and count the number of black lines placed in them. This translates into the thorough inspection of underwater areas. The precise position control is needed to solve this NIST task, because the cylinders are small, and the lines are only visible from a certain angle, thus better station-keeping capability is a major advantage. Fig. 1a shows the arrangement of the actual cylinders during a test round with an AUV.

3.2 Rope Cutting

In the second test case, the robots should clear an area enclosed by ropes. This method measures the cutting and targeting capability of the robot using different materials. The ropes are placed in different orientations. To solve this task, a cutting tool needs to be installed on the AUV. It has to be stable and sharp enough to cut through the thick, wet ropes. Fig. 1b shows an example structure of the ropes.

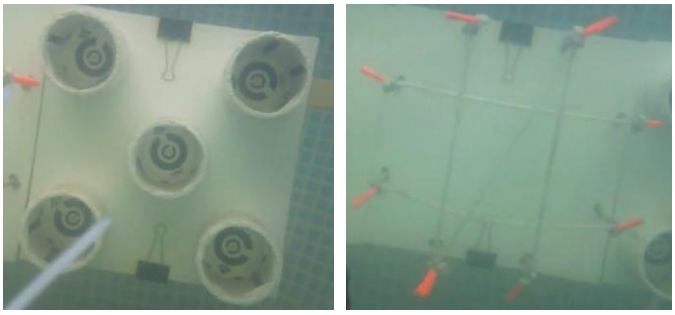


Figure 1 a-b
Examples of the NIST underwater response robot capability tests
a) The structure of the cylinders to be approached and explores
b) Rope structure to be cut by an AUV

3.3 Hooking

In the third test, the AUV should deploy a carabineer to the selected object, which is one of the loops placed in different directions. The complete object consists of 5 U-bolts, arranged in different orientations, thus the orientation of the carabineer is very important. Fig. 2a shows the structure of the carabineer holder.

3.4 Soft Grab

The fourth test is similar to the third; however, in this case, the robot should deploy an alligator clip on a soft target. The difficulty in this task is that the target keeps moving, driven by the currents and other conditions, therefore the AUV control methods must be much more sophisticated. Furthermore, precise positioning of the clip is required. Fig. 2b shows the soft target and the clip.

Figure 2 a-b
a) AUV test case 3, the structure of the carabineer holder
b) The fabric strap and the clip

3.5 Closing a Valve

The fifth task is about closing and opening a valve as presented in Fig. 3. The object is set up in a rotatable way, and the AUV has to rotate the tap of the valve 90° to the left and then to the right. To achieve this task, precise control of the robot is indispensable, because the AUV should maneuver up/down, left/right and forward/backward along a curved path.

Figure 3
The mock of an underwater valve to be closed by an AUV

4 Available Hardware and Software Components

As a solid AUV platform to pass the NIST tests with, we employed a Sparus II lightweight hovering vehicle with mission-specific payload area and efficient hydrodynamics for long autonomy in shallow water (200 meters). The Sparus was originally developed at the University of Girona [5]. The AUV is torpedo-shaped, and has a built-in computer with an Intel Core i7 processor, 4 GB RAM, a 250 GB SSD, and is equipped with a 1.5 kWh battery (providing up to 8 hours autonomous navigation[1]. The Sparus II is shown in Fig. 4. It has 3 motors for underwater locomotion: one motor is for depth control and two are for maneuvering. On the software side, the system is based on the Robot Operating System (ROS), and has an additional software package named COLA2, which enables the hardware to use the integrated complex sensor and actuator systems. We have used this basic package and created our own software packages, for autonomous navigation and teleoperation. The Sparus is a very capable platform for developing an advanced AUV.

[1] cirs.udg.edu/auvs-technology/auvs/sparus-ii-auv/

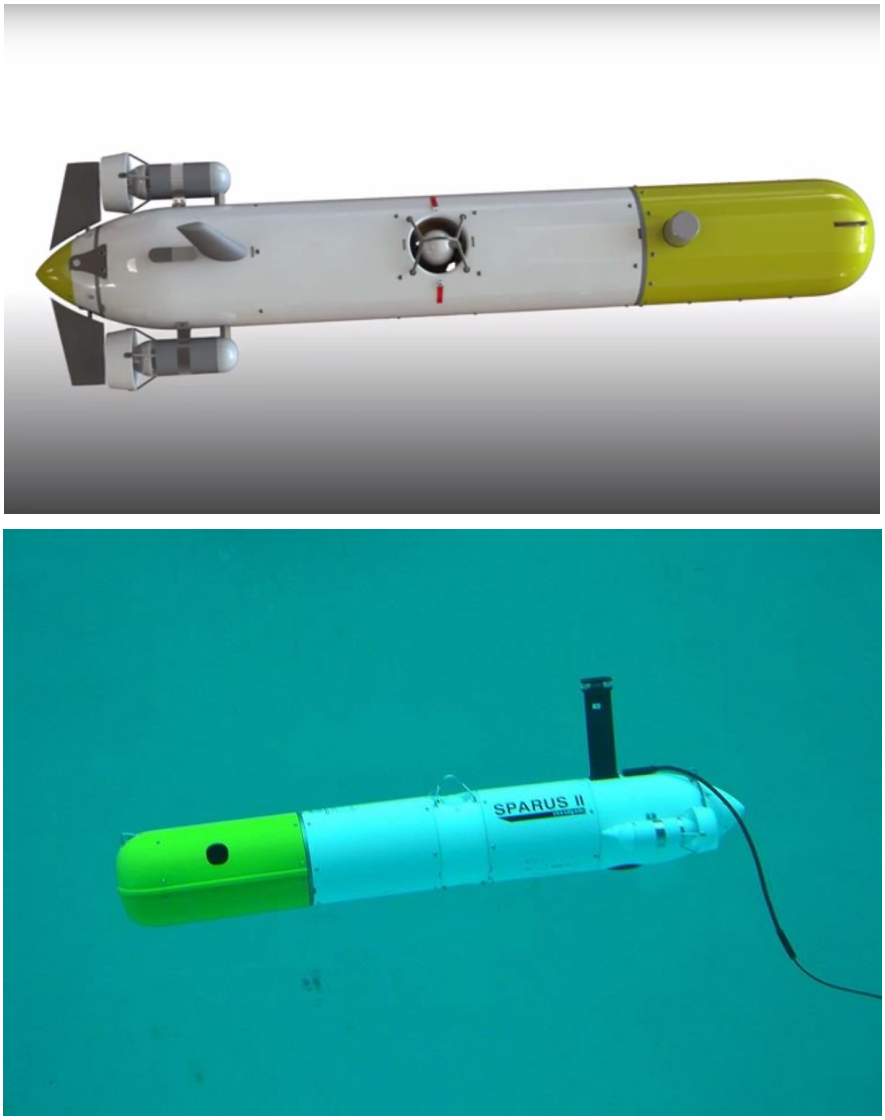


Figure 4
The Sparus II AUV platform developed at the University of Girona
(Photo credit: University of Girona)

5 Extensions to the AUV Platform

We designed two special hardware components to complete the NIST tests. The first is a waterproof cover for the CCD camera. It is indispensable for performing teleoperation tasks with the robot and the mobile manipulator later equipped on the Sparus. The other components were two waterproof covers for the servos, enabling us to build an underwater mobile 2+1 Degree of Freedom (DoF) manipulator for task execution.

5.1 Waterproof Camera Cover

We used a Microsoft LifeCam Cinema HD USB web camera (Fig. 5) to provide high quality real-time video streaming. The first step was making a waterproof cover for the web camera. The biggest challenge was presented by the external pressure, as the comparable water pressure is about 2 bars (200 kPa) at 10 meters below the surface. This means 2 kg weight on every 1 cm^2. The other difficulty was the corrosive effect of the sea water, when we used plastic materials (Plexiglas, thermosetting plastic) to manufacture the cover.

Figure 5
Microsoft LifeCam Cinema HD USB web camera

5.2 Waterproof Servo Cover

To create a mobile manipulator, we employed model RC servos, like the ones used in model boats and cars. These servos are not waterproof, therefore, we designed a custom cover for each of them. The case is compatible with all of the standard sized servos that can be found in commercial distribution. We built a 2+1 DoF robot manipulator with simple kinematics (Fig. 6) from these servos, where each DoF is providing an orientation, while another 3 DoF were derived from the AUV's ability for positioning.

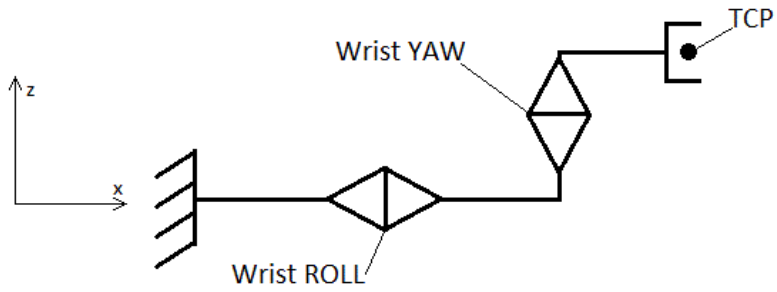


Figure 6
Kinematic structure of the underwater arm designed for the AUV

6 Hardware Implementation

6.1 Design and Manufacturing of the Waterproof Camera Cover

The components of the cover were designed using the SolidWorks modeling software. Fig. 7 shows the exploded 3D CAD model of the cover.

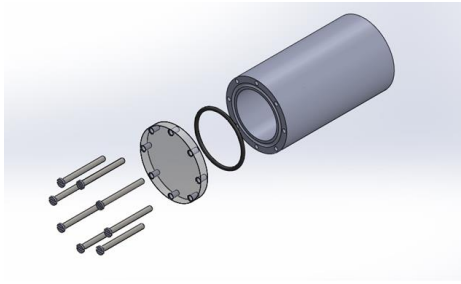


Figure 7
3D CAD model of the waterproof camera cover for the AUV

The base of the cover was made of thermosetting plastic using a lathe. In the front, there is a lid, made of water-clean Plexiglas, and the hermetic seal is provided by an O-ring. The lid is secured by eight M3 screws, and the outlet of the USB cable is insulated with epoxy glue and silicone rubber. Fig. 8 shows the cover with an installed camera.

Figure 8
The AUV camera fit into its waterproof cover

6.2 CAD/CAM of the Waterproof Servo Cover

The servo cover was also designed in SolidWorks. This cover is composed of two parts: the top contains two ball bearings for holding the drive axle stable. The hermetic seal is provided by a lip seal. The drive axle and the axle of the servo are connected by coupling. In the bottom part, there is an outlet for a cable. It is also insulated by epoxy glue and silicone rubber. Fig. 9 shows the 3D CAD model of the cover.

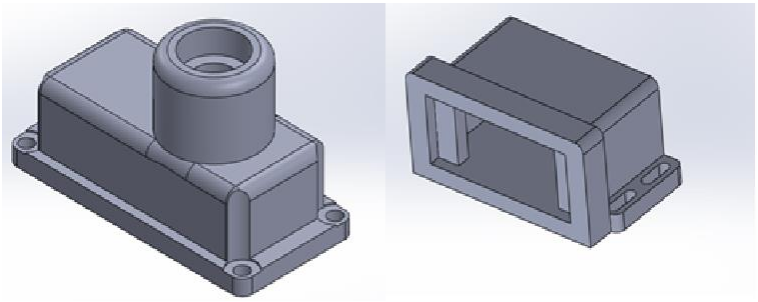


Figure 9
3D CAD model of the waterproof servo cover

7 Software for Teleoperation

The implemented teleoperation solution in the AUV is a ROS node that communicates in two directions (Fig. 10). On the one hand, it reads values from the *"/joy"* topic of the COLA2 framework in every 50 ms. The values are the states of the buttons of the integrated Xbox 360 controller. The left and right triggers and arms are used on the controller, thus the values of these control events are sent to the thrusters and the Arduino Mega microcontroller.

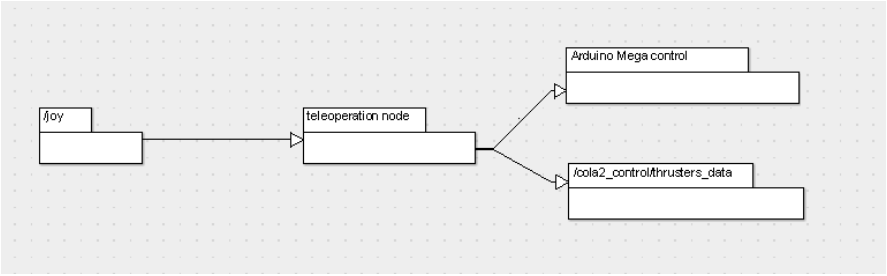


Figure 10
Teleoperation ROS node architecture for the AUV

With the vertical movements of the triggers and arms, the AUV is directly controlled. The game controller posts into the *"/joy"* COLA2 topic a value between -1 and 1 every time when some status change occurs. Fortunately, the thrusters of the AUV can be operated by values between the same intervals, so it is not justified to map the value between other intervals. These values can be forwarded directly to the "*/cola2_control/thrusters_data*" ROS topic, where the control of the thrusters is solved. Because of the noisy signals of the Xbox 360 controller, the values of the arms and triggers between 0 and 0.3 are considered as 0. A forwarded 1 means that the thruster should work with 100% performance, -1 is the opposite, -0.3–0.3 means that the thruster is stopped. This operates on a similar principle in the case of the servos. The vertical movement of the left trigger, of the joystick, results in the AUV moving forward or backward. On the other hand, the right trigger of the joystick results in the device turning left or right. If the operator wants to turn left or right, the value of the left and right trigger will be sent to the *"/cola2_control/thrusters"* data. The left thruster will receive the value, while the right thruster will receive the value with the opposite sign. The maximum performance output of the thrusters is not enabled, and automatically degraded to a safe performance output value by the software.

8 Underwater Manipulation

This low-cost underwater manipulator, with three DoF, required additional software and hardware components for the AUV to work. An Arduino Mega 2650 microcontroller was used for direct control, and a Xbox 360 game controller enables the human operator to drive the manipulator indirectly. The ROS connected both control units. During teleoperation, the game controller was re-mapped to enable the smooth 2 DoF movements, of the manipulator. The left arm of the game controller defined the vertical movements. (This first servo is attached to the AUV). The right arm of the game controller defines the movement of the second servo attached to the first one. After some tests, it was deduced that the arms of the Xbox 360 game controller returns a value between -1 and 1, so if the program is able to read this value every 50 ms, it could move a servo up to 20 degrees per minute. This simple solution was robust enough to be used for servo control. According to our tests with a polling rate of 20 per second, a smooth underwater manipulation with sufficient precision is realized.

9 Tests

Some tests have been carried out after the realization of the waterproof cases. The first test environment was a pressure chamber with 10 bars, where all tests were successful. After this, we attached it to the AUV at a temporary location. For different kind of tests, a pool was set up outside the lab, filled with fresh water. It had the dimensions of 4 x 2 x 2 meters (L x W x H). The tools created by the team were left in the pool with the AUV to decide how well they could stand up against the water. The next test environment was also a pool. It was set up in the euRathlon 2015 competition (http://eurathlon.eu/) for test purposes, yet filled up with sea water. One of our camera cases was slightly damaged by the salty water, but destroyed, during the competition, so we continued the NIST tests. The last test environment was the euRathlon 2015 competition (S1 and S2 session) where we had to use our device in five meters of depth, performing some of the NIST tests.

10 Lessons Learned

The outdoor euRathlon 2015 competition and the NIST trials were the ultimate testing environments for our AUV. Both the developers and the response robot had to cope with the real-world scenario. It was a physical challenge that brought both human and machine to their limits. The very first problem was the difference between the software based simulation environment and reality was that we were

able to realize during the firsts tests in the Lupa lake in Hungary, where a lot of time was spent balancing the AUV. Without proper balance, autonomous navigation algorithms and teleoperation solutions cannot work properly. Another serious problem was the inappropriate choice of materials of some of the 3D printed components. Certain kinds of materials can be damaged by the sea water and the team did experience this during the trials in fresh water. The first 3D models were printed with PLA instead of ABS, and salt water has an effective degenerative effect on PLA. Furthermore, it was easy to integrate the Xbox 360 joystick into the AUV system, but it was very difficult to achieve a smooth control of the thrusters with teleoperation, which was definitely needed for the NIST tests.

There was a special failure of the system that we realized during the tests. Every time we wanted to control the thrusters at a high RPM, the USB web camera was detached by the operation system that runs on the AUV, so the camera was not able to support our solutions and the system. The problem turned out to be the high pulse generated by the thrusters, the metal body of the AUV, and the inadequate insulation of the cables. Once the cables were properly insulated, this failure disappeared instantly.

Conclusions

This paper describes our work in AUV response robot testing, from two aspects: we proposed additional testing methods for AUV capability assessment. Here, two parameter groups were identified to assess the adaptability to a dynamically changing environment (temporal awareness) of the robots. These are the temporal resolution (sensing/sampling frequency) and Information aging speed. Further, we detailed how a Sparus II platform based AUV can be enhanced to pass the existing underwater response robot tests defined by National Institute of Standards and Technology (NIST).We developed software and hardware components to extend the capabilities of the AUV platform: the additional software components were indispensable for precise navigation, position holding and teleoperation. Beyond this, we integrated an Xbox 360 game controller, a self-made waterproof manipulator arm and camera. The sensor provided the required visual data for teleoperation, the game controller and the actuator enabled the smooth operation and control of the manipulator joints. The outdoor euRathlon 2015 competition and the NIST tests were the real field trials in a physical environment for our AUV system. The project provided us with massive opportunities to find and successfully resolve major, “real-world” engineering challenges.

Acknowledgement

Authors would like to thank NATO Centre for Maritime Research and Organization (CMRE) for the opportunity to access and use the Sparus II AUV during the euRathlon competition, and also the friendly support of NIST and University of Girona (UdG). The financial support of this work was from the University Research and Innovation Center, Óbuda University, Hungary (URIC/EKIK).

References

[1] W. Dirk, D. R. Edgington and C. Koch. "Detection and Tracking of Objects in Underwater Video." *In Proc. of the IEEE Computer Vision and Pattern Recognition*. CVPR 2004, Vol. 1, pp. I-544, 2004

[2] Underwater Robotics Research Centre (CIRS), "Design of SPARUS II AUV," Technical Report. cirs.udg.edu/wp-content/uploads/2014/05/ /SPARUS_II_design.pdf, 2014

[3] A. Jacoff, E. Messina, H.-M. Huang, A. Virts, A. Downs, R. Norcross, R. Sheh; "Guide for Evaluating, Purchasing, and Training with Response Robots using DHS-NIST-ASTM International Standard Test Methods"; NIST Standard Test Methods For Response Robots; ASTM International Standards Committee on Homeland Security Applications (E54.08.01). http://www.nist.gov/el/isd/ks/upload/DHS_NIST_ASTM_Robot_Test_Methods-2.pdf, 2014

[4] A. Jacoff, K. Saidi, R. von Loewenfeldt, Y. Koibuchi; "Development of Standard Test Methods for Evaluation of ROV/AUV Performance for Emergency Response Applications", in Proc. of the 16th MTS/IEEE OCEANS, Washington DC, 2015

[5] M. Carreras, C. Candela, D. Ribas, N. Palomeras, L. Magí, A. Mallios, E. Vidal, È. Vidal and P. Ridao; "Testing SPARUS II AUV, an open platform for industrial, scientific and academic applications", *6th Intl. Workshop On Marine Technology*, Martech 2015, Cartagena, pp. 106-109, ISBN: 978-84-608-1708-6, 2015

LPV-based quality interpretations on modeling and control of diabetes

György Eigner[1], József K. Tar[2], Imre Rudas[2], Levente Kovács[1]

Research and Innovation Center of Óbuda University
[1] Physiological Controls Group
[2] Antal Bejczy Center for Intelligent Robotics
Kiscelli street 82., H-1032, Budapest, Hungary
{eigner.gyorgy, tar.jozsef, kovacs.levente}@nik.uni-obuda.hu, rudas@uni-obuda.hu

Abstract: In this study we introduce different novel interpretations in the case of Linear Parameter Varying (LPV) methodology, which are directly usable in modeling and control design in diabetes research. These novel interpretations are based on the parameter vectors of the LPV parameter space. The theoretical solutions are demonstrated on a simple, known Type 1 Diabetes Model used in intensive care.

Keywords: Diabetes, LPV model, Affine LPV, qLPV

1 Introduction

We would like to dedicate this paper in honor of Prof. Antal Bejczy's life achievements who was a pioneer of space robotics. Prof. Bejczy provided continuous inspiration for us through his world leading work on the field of control and automation.
Modeling and control is extremely important in the research of Diabetes Mellitus (DM) [1]. In the light of the fact that no cure exists for the disease, the available treatment for those, who suffer from insulin dependent diabetes (Type 1 DM (T1DM), Double DM (DDM), etc.), the only chance in order to stay alive and/or maintain their condition, is the treatment with externally injected insulin hormone. The regular treatment is manual insulin administration by pre-calculated amounts of insulin. It depends on the assumed carbohydrate intake of the patient. More sophisticated solution is the Artificial Pancreas (AP) concept, in which the insulin injections are made by an insulin pump device with rapid acting insulin based on control algorithms and Continuous Glucose Monitoring (CGM). In recent years several advanced control solutions have appeared in literature, which may successfully deal with DM control. There were attempts with the fusion of modern and classical control theory, like switching Proportional-Integral-Derivative (PID) control [2, 3], with modern control theory, for example Model Predictive Control (MPC) [4, 5],

or Soft Computing-based Control (SCC) [6, 7]. Moreover, Robust Control (RC) theory also were considered by researches [8–10] in this scientific discipline. In order to test and preliminarily validate the developed algorithms in silico models can be used. The role of these mathematical models - so called patient models -, is to simulate the glucose-insulin household of a real patient as realistically as possible. Moreover, with different additional sub-models (like digestion, absorption, physical activity, sensor, noise, etc. models), highly sophisticated in silico models can be realized. Generally, one can consider large scale (macro size) models of biological processes that are relatively slowly changing [11] in time. In case of such processes Gain Scheduling Theorem (GST) can be used for modeling and control purposes. From modeling point of view, one of the most powerful tools of GST is the Linear Parameter Varying (LPV) method. LPV approaches are useful tools in control design for biological processes since their nonlinearities complicated by time delay effects. LPV techniques are frequently used in RC [12]. In order to describe a system with uncertainty, linear fractional transformation (LFT) can be used. In such cases, LPV system becomes derivable from LFT, since the nominal system depends on the uncertainties. Intuitively, this dependency can be described with function relations. Moreover, this kind of dependencies can be produced if the elements which cause the nonlinearity are getting out from the nonlinear system. In this way, a linear system is obtained which will be the function of the pulled components. Regularly, the selected elements on which the system depends are called scheduling parameters [13]. A vector can be created from them, which is called scheduling parameter vector, or shortly, parameter vector ($\rho \in \mathbb{R}^k$).

In diabetes modeling uncertainty, which comes from the varying parameters, the intra- and inter-patient variability is a crucial question. Robust control allows to handle these uncertainties in a natural way. With LPV modeling linear RC methods also can be used, besides that the properties of the original nonlinear model are still valid [8,14]. Usually, in the physiological models the nonlinearities occur within the system model and do not affect the output matrices. This is especially true in case of diabetes modeling, since the nonlinearities are usually connected to the description of different dynamical effects, insulin kinetics, etc. That means, that if a system is described with its state-space (SS) representation the state matrix, $A(t)$ will be affected by nonlinearities. In order to hide this effect, the causing component can be selected as scheduling parameter, in this way the nonlinearities in $A(\rho(t))$ can be separated.

In this paper we are going to introduce novel interpretations and considerations about the similarity of different diabetes models. Moreover, new quality definitions will be derived based on the LPV parameter vector which can be used during modeling and controller design.

2 State space representations and LPV configurations

2.1 State space representations of a dynamic system

A general, nonlinear time varying (NLTV) system can be represented with the following functions [15–17]:

$$\begin{aligned} \dot{x}(t) &= f(x(t),u(t)) \\ y(t) &= h(x(t),u(t)) \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $f(x(t),u(t))$ is a nonlinear state function, $y(t) \in \mathbb{R}^p$ is the output of the system and $h(x(t),u(t))$ is the nonlinear output function. With reformulation this can be described in SS form:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \tag{2}$$

where $A(t) \in \mathbb{R}^{n\times n}$ is the state matrix, $B(t) \in \mathbb{R}^{n\times m}$ is the input matrix, $C(t) \in \mathbb{R}^{p\times n}$ is the output matrix and $D(t) \in \mathbb{R}^{p\times m}$ is the feed-forward matrix. The state matrices in (2) can be united into a single system matrix:

$$S(t) = \begin{pmatrix} A(t) & B(t) \\ C(t) & D(t) \end{pmatrix} \tag{3}$$

where $S(t) \in \mathbb{R}^{(n+p)\times(n+m)}$. Thereby (2) in simpler form via (3):

$$\begin{pmatrix} \dot{x}(t) \\ y(t) \end{pmatrix} = S(t) \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \quad . \tag{4}$$

If the state matrices do not depend on time, Linear Time Invariant (LTI) system occurs, described with the following SS equation:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{5}$$

and can be written in the previous compact form (4), however, here S does not depend on time:

$$\begin{pmatrix} \dot{x}(t) \\ y(t) \end{pmatrix} = S \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \quad . \tag{6}$$

Basically, with GST an LTV system can be described with infinite number of LTI systems in continuous time domain and particular number of LTI systems in discrete time domain, if only the elements of the state matrices vary over time, but the structure of the SS representation is invariant. That means, the LTV systems run through a ”trajectory” during operation over time - where the trajectory consists of infinite number of LTI systems. Fixing the elements of the SS representation of a LTV system at a given moment means that the LTV system is simplified to a LTI structure. For example, $S(t)$ exactly at 10 *min* will be equal to $S(t_{10}) = S_{10}$.

2.2 LPV description and configurations

The literature distinguishes between the models according to the fact whether the selected scheduling variables are not state variables (LPV) or they are state variables also selected as scheduling parameters (quasi-LPV, qLPV). Nevertheless, there is no difference between them from notation point of view. However, the eligible interpretation of the cases is important to be noticed. In the sequel we introduce the general form of LPV systems. Assume that the parameter vector is designated with $\rho(t)$. In such a case the usual notation for an LPV model is:

$$\begin{aligned} \dot{x}(t) &= A(\rho(t))x(t) + B(\rho(t))u(t) \\ y(t) &= C(\rho(t))x(t) + D(\rho(t))u(t) \end{aligned} \quad . \tag{7}$$

Unification can be made similarly as in (3) and from (7):

$$S(\rho(t)) = \begin{pmatrix} A(\rho(t)) & B(\rho(t)) \\ C(\rho(t)) & D(\rho(t)) \end{pmatrix} \quad . \tag{8}$$

The compact form of general LPV system from (7) becomes:

$$\begin{pmatrix} \dot{x}(t) \\ y(t) \end{pmatrix} = S(\rho(t)) \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \quad . \tag{9}$$

The classical approaches that use LPV form in modeling apply affine and polytopic LPV system models [13, 16, 17]. However, in the recent years a soft-computing based LPV modeling issue arose the TP transformation-based LPV modeling possibility [18, 19]. Because our quality interpretation can be used beside affine and polytopic configurations, as well, we shortly described them below.

2.2.1 Affine LPV configuration

In this type, the LPV systems are the affine function of the parameter vector. If the system is given with its SS representation, then it consists on two main parts: a permanent, which is independent from the parameter vector and a varying, where the dependency occurs.

$$\begin{aligned} A(\rho(t)) &= A_0 + \sum_{i=1}^{k} \rho_i(t) A_i \\ B(\rho(t)) &= B_0 + \sum_{i=1}^{k} \rho_i(t) B_i \\ C(\rho(t)) &= C_0 + \sum_{i=1}^{k} \rho_i(t) C_i \\ D(\rho(t)) &= D_0 + \sum_{i=1}^{k} \rho_i(t) D_i \end{aligned} \quad . \tag{10}$$

The permanent matrices are the A_0, B_0, C_0, D_0, which represents the independent parts from the parameter vector. The permanent and varying parts can be written in

short form similarly as (3):

$$S(\rho(t)) = \begin{pmatrix} A_0 + \sum_{i=1}^{k} \rho_i(t) \cdot A_i & B_0 + \sum_{i=1}^{k} \rho_i(t) \cdot B_i \\ C_0 + \sum_{i=1}^{k} \rho_i(t) \cdot C_i & D_0 + \sum_{i=1}^{k} \rho_i(t) \cdot D_i \end{pmatrix} \tag{11}$$

and in this way the complex system matrix can be simplified as:

$$S(\rho(t)) = S_0 + \sum_{i=1}^{k} \rho_i(t) \cdot S_i \quad . \tag{12}$$

The affine LPV system can be written in compact form similarly as (9).

2.2.2 Border of validity in case of affine LPV models

The elements of the parameter vector ρ forming the so-called Parameter Space (PS) which is an abstract, arbitrary mathematical space. The dimension of it is equal to the number of the selected scheduling variable.
Assume that the dimension is equal to k in general case, $\rho \in \mathbb{R}^k$. The values of the parameter vector ρ are varying over time, however, this varying is inside a particular range determined by the minimum and maximum value of the given variable. This validity range covers the meaningful parameter range based on physiological or physical processes (e.g. positive masses). This attitude determines the "Parameter Box" (PB), which is a particular region in the PS. The affine LPV models keep their validity only in this range (inside the PB) during operation. This configuration is very advantageous from control engineering points of view, since

- The PB represents the workspace where the LTV system can be found during operation and each of the points also can represent an LTI system at a given moment;
- The control design tasks may be easier, because these regions are usually small;
- In case of RC, the borders of the PB can be the borders of parameter uncertainties;
- With affine LPV representation, nonlinearities can be hidden. Moreover, a given time stamp represents an LTI system, which can be selected as "operating system", if its properties are appropriate from control design point of view.

In Fig.1(a). we highlighted a $3D$ PS, where the $\rho \in \mathbb{R}^3$ and the values of the scheduling parameter are varying among the range which is determined by the minimum and maximum values of the parameters. Mathematically, this can be reached, if the parameter vector is:

$$\rho(t) = \begin{pmatrix} \rho_1(t) \\ \rho_2(t) \\ \rho_3(t) \end{pmatrix} = \begin{pmatrix} [\rho_1^- .. \rho_1^+] \\ [\rho_2^- .. \rho_2^+] \\ [\rho_3^- .. \rho_3^+] \end{pmatrix} \quad . \tag{13}$$

2.2.3 Polytopic LPV configuration

Affine LPV configuration means a natural way to describe or highlight different properties of a system, however, usually not directly used in controller design [20]. Nevertheless, the polytopic LPV configuration, which is directly derivable from affine configuration (and gives a basis for the TP-transformation based design, as well) is directly usable in such design methods. Practically, the polytopic LPV theory is based on the barycentric theorem of Möbius, describing the position of a point in a triangle with using the vertices of the triangle as reference points [21, 22]. Further, Warren and his colleagues have proved the possibility that in case of arbitrary convex sets it is also true that with using the vertices of a convex set as reference points the position of an arbitrary internal point can be described. This is the key property which can be used in control engineering approaches [16].
As the affine LPV system is only operating inside the parameter box, the vertices of the parameter box can be used as reference points to describe each system that can occur during operation, namely, each such internal system will be the convex combination of the vertices of the polytope, if the convexity criteria is satisfied. In the case of a control system, the convexity depends on the following considerations. An internal system S can be described with polytopic coordinates α_i, if the system representation belonging to α_i, i.e. S_i satisfy the following restrictions:

- The polytopic coordinates should be non-negative real values, $\alpha_i \geq 0$;
- The sum of the polytopic coordinates should be equal to one, $\sum_{i=1}^{q} \alpha_i = 1$;
- The internal system is the convex combination of the vertices of the polytope, $S = \sum_{i=1}^{q} \alpha_i S_i$.

or shortly [15]:

$$\left\{ S = \sum_{i=1}^{q} \alpha_i S_i : \alpha_i \geq 0, \sum_{i=1}^{q} \alpha_i = 1 \right\} \quad . \tag{14}$$

Normally, when $q > k$ we have a redundant representation that generally allows the satisfaction of the restrictions. The barycentric coordinate function can be given in the following way [16]:

$$"\alpha_i(\rho) = \frac{\Upsilon_i(\rho)}{\sum_i \Upsilon_i(\rho)} \tag{15}$$

where Υ_i is the weight function which belongs to the ith vertex for a ρ point inside the convex polytope. The weight function can be calculated as follows:

$$\Upsilon_i(\rho) = \frac{\mathrm{vol}(\Pi_i)}{\prod_{j \in \mathrm{ind}(\Pi_i)} (n_j \cdot (\Pi_i - \rho))} \tag{16}$$

where $\mathrm{vol}(\Pi_i)$ is the volume of the parallelepiped span by the normals to the facets incident on vertex i, i.e., Π_i, $\{n_j\}$ is the collection of normal vectors to the facets incident on vertex i, and $\mathrm{ind}(S_i)$ denotes the set of indices j such that the facet normal to n_j contains the vertex Π_i. The volume of the parallelepiped can be computed as:

$$\mathrm{vol}(\Pi_i) = |\det(n_{\mathrm{ind}})| \tag{17}$$

where n_{ind} is a matrix whose rows are the vectors n_j where $j \in \mathrm{ind}(V_i)$ [16]."

Fig.1(b). shows an example whereon the aforementioned theories were taken into account in case of 3 scheduling variables. In this case, the parameter space is 3 dimensional and it is visible that the parameter box is determined by the minimum and maximum values of the parameter vector. Furthermore, the vertices of this box, S_i serve as reference points and α_i are the convex coordinates at the same time. The actual system inside can be calculated with the barycentric calculus, namely, the actual system $S(\rho(t))$ will be the convex combination of the vertices, i.e.:

$$S(\rho(t)) = \sum_{i=1}^{8} \alpha_i(\rho(t)) \cdot S_i \quad . \tag{18}$$

Obviously, if the actual system reaches a vertex that means the convex coordinate of the particular vertex will be equal to one, however, the others will be equal to zero. For example, if the system reaches the vertex S_1, the $\alpha_1 = 1$, further $\sum_{i=2}^{8} \alpha_i(\rho(t)) \cdot S_i = 0$ and the actual system is:

$$S(\rho(t)) = \alpha_1(\rho(t)) \cdot S_1 + \sum_{i=2}^{8} \alpha_i(\rho(t)) \cdot S_i = \alpha_1(\rho(t)) \cdot S_1 \quad . \tag{19}$$

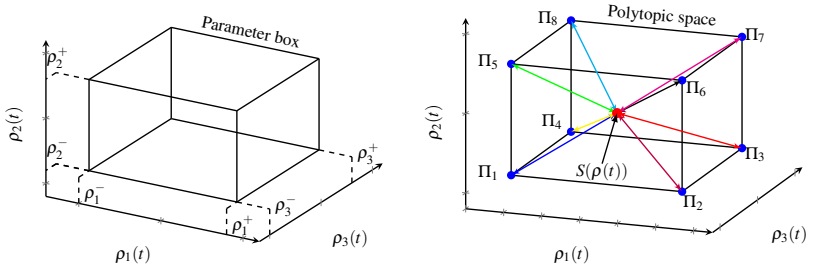


Figure 1
Affine and polytopic LPV model examples in the 3*D* parameter space

2.3 Specificities of LPV models in the field of diabetes research

According to the complex patient models the following general properties can be considered:

- Inputs are not affected by nonlinearities; they have impulse attitude; mostly consists of external insulin and glucose (or rate of appearence of glucose) intake; do not directly affect the outputs (in state space representation this means that the D vector contains only zero elements and it does not depend on the parameter vector ρ);

- Output(s) are connected to the blood glucose level (or the blood glucose level is the output itself); not affected by nonlinearities;

- Since the nonlinearities do not affect the inputs and the outputs, it is not necessary to select their elements as scheduling parameters, which means that B and C are independent from the parameter vector ρ; moreover, these usually do not depend on time;

- The nonlinearities occur in the state matrix (A) regarding to glucose-insulin dynamics, glucose and/or insulin absorption, effect and dynamics of insulin; the intra- and inter-patient variabilities are represented in the elements of A and usually these are time dependent; scheduling variables should be selected from the elements of A in order to hide the nonlinearities and make the handling of A convenient from control point of view.

From the aforementioned consideration it can be derived that the LPV-type diabetes models have the following form which can be observed in several studies [8, 9, 14]:

$$\begin{aligned} \dot{x}(t) &= A(\rho(t))x(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{20}$$

where the system matrix S is the following:

$$S(\rho(t)) = \begin{pmatrix} A(\rho(t)) & B \\ C & D \end{pmatrix} \tag{21}$$

and the state space representation in compact form is:

$$\begin{pmatrix} \dot{x}(t) \\ y(t) \end{pmatrix} = S(\rho(t)) \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \quad . \tag{22}$$

Equation (21) shows that in this form the LPV-type diabetes models only contain dependency from the parameter vector ρ in the state matrix A and all time dependent components are selected as scheduling variable.

3 Different interpretations of quality based on LPV configurations

3.1 Norm-based "difference" definition in the Parameter Space

Each dimensions of the PS correspond to an element of the parameter vector ($\rho \in \mathbb{R}^k$). Inside this abstract space each point can be determined by the corresponding parameter vector. Furthermore, this abstract PS is an Euclidean vector space and vector L_p norms can be interpreted. Assume two vectors ρ_a, $\rho_b \in \mathbb{R}^k$ in the PS. The L_2 norm based distance between the vectors can be described as follows:

$$\|\rho_a - \rho_b\|_2 = d_2 \ . \tag{23}$$

The defined d_2 can be used in various way depending on the interpretation of the PS which are presented in the next section.

3.2 Possible interpretations of the defined norm-based difference in the Parameter Space

The points of the PS which are determined by the parameter vector can be interpreted on their own as vectors, which elements consist of the parts of the system. However, the parameter vectors can unequivocally determine an underlying LTI system in the affine LPV case and a well-characterized LTI system in the polytopic case. The following statements are general LPV model properties regardless it is the affine or polytopic type LPV model.
When the goal is to emphasize particular properties of a system, each of the parts representing these properties have to be selected as scheduling variables. For example, if we investigate only the insulinemia (I) and blood glucose (G) from (A-1), these have to be selected as scheduling variables and the PS will be 2 dimensional. In this case, the introduced d_2 is appropriate to define a "difference" between the I and G which belong to different points in the PS. Namely, a permanent reference parameter vector ρ_{ref} can be defined with given constant I and G. The actual parameter vector ρ_{actual} varies over time. In this case $d_2 = \|\rho_{ref} - \rho_{actual}\|_2$ determines the 2-norm based difference of them and this can be interpreted as an "error" or "quality" signal, if ρ_{ref} and ρ_{actual} are different during operation. Generally, this interpretation can be extended for any k dimensional parameter vector.
If the question is to design a controller the key point is what the selected scheduling variables are. At this point several approaches and interpretations can be distinguished. The main ones are the following:

1. The selected scheduling variables are those properties which have to be monitored during the operation. In this case, the 2-norm based difference can be used as "quality" signal and the performance of different controllers can be compared with this quality signal in the PS.

2. The selected scheduling variable are those properties which have to be controlled and monitored during operation. In this case, we can interpret the

2-norm based signal as "error" signal. This type of error can be used during the controller design.

3. The selected scheduling variables are those components which are time dependent in LTV case. The parameter vector unequivocally determines the underlying LTI systems which can occur from the general LTV system during operation at given time moments. In this case, the defined 2-norm difference can be used as "metric" in order to compare LTI systems in the PS.
4. The selected scheduling variables are those components which are causing nonlinearities in NLTV case. The general purpose of this approach is that the linear controller design techniques become usable beside polytopic LPV systems. If each nonlinearity causing and time dependent parameters are selected as scheduling variables, the parameter vector unequivocally (affine LPV case) or satisfactorily (polytopic LPV case) determines an LTI system during operation at given time moments. In this case the defined 2-norm difference can be used as metric in order to compare LTI systems in the PS.

In the following sections we detail the key aspects from the above mentioned points.

3.2.1 The 2-norm based difference as quality and error signal

The most important issue in these cases is the way how the parameters are selected from the original model and the interpretation of them.
If the scheduling parameters are selected each-by-each and not grouped, then each dimensions of the PS will be an individual variable with physiological or physical meaning (e.g. the I and G parameter from (A-1)).
However, the scheduling variables can be grouped (e.g.,(25)). In this case the scheduling variables can loose their original meaning and cannot be interpreted individually.
Nevertheless, the grouped scheduling parameters allow to interpret the 2-norm based difference sophisticatedly. If the goal is to monitor how the specific properties of the system vary over time and compare this vary with predefined requirements, the 2-norm based difference can be interpreted as quality signal. This approach can be important in such application, where the different parts of the system cannot change to drastically relative to each other. Naturally, in this case, these specific parts have to be selected as scheduling variables.
It has to be noted that the observability should be considered in this case, since, the selected parts have to be observable or estimable. Fig. 2(a). shows a 2 dimensional parameter space. For example, a possible goal (beside other goals) of the applied controller can be to hold permanently two system properties during operation. If these properties (which are represented with different parts of the system) are selected as scheduling variables, the performance of the controller can be assessed based on the $d_2(t)$ signal.
During controller design the 2-norm based difference can be used as an "error" signal in the classical meaning. The appropriate selection and interpretation of the scheduling variables are necessary. The observability and controllability of the scheduling variables are important issues, as well.

The first step is the selection of parts of the models as scheduling variables which have to be controlled. However, in case of grouped multiply out (e.g. $S_I Q(t)/(1+\alpha_G Q(t))$ of (25)) should be reasonable. The error signal ought to be known at every time moment as the basis of control.
If the elements of the parameter vector are not observable, they have to be estimated or approximated. The control signal affects the scheduling variables directly or through coupling. Without connection, the scheduling variables cannot be influenced by the control signal.
Beside these constraints, controller design is possible. For example, the recently developed Robust Fixed Point Transformation (RFPT)-based controller design methodology can be used [23]. The first stage is the investigation of the effect chain of the control action, namely, how the control signal affects the controlled variables which are here the scheduling variables $\rho(t) = f(\rho(t)^-, u(t))$, where $\rho(t)^-$ is the a-priori knowledge about the $\rho(t)$ and $u(t)$ is the control signal. With approximate inverse kinematic description ($\tilde{u}(t) = f(\rho(t), \rho(t)^-)$) and appropriate control laws a RFPT-based controller can be designed. In this case the error signal can be the developed by 2-norm based difference (d_2), which arises when the nominal prescriptions of the controlled variables (the scheduling variables) are not equal with the actual values of them. Geometrically, the nominal prescriptions of the controlled variables can be a permanent point of the PS (ρ_{ref}) and the actual values $\rho_{act}(t)$ are varying in time during operation. Based on the arised error signal $d_2(t)$ a RFPT-based controller can be designed [23].
In Fig.2(a). a $2D$ example can be seen, where $d_2(t)$ can be interpreted as the mentioned error signal. The comparability of the order of magnitudes of the scheduling variables represent a significant point. The nature of the Euclidean norms determine the particular difference signals affect mostly on the 2-norm based difference which has the highest magnitudes (e.g. the $\rho_{reference,1} - \rho_{actual,1} \gg \rho_{reference,k} - \rho_{actual,k}$, $\rho \in \mathbb{R}^k$, $k \neq 1$ determines that the 2-norm based difference will have strong connection with the variation of $\rho_{reference,1} - \rho_{actual,1}$). If the scheduling variables have to be considered with the same "weight" (they have the same importance), different normalization and weighting techniques can be used [20].

3.2.2 The 2-norm based difference as comparison of systems

Generally, LPV techniques are used in order to embed the uncertainties into a system model or hide the system model nonlinearities by making the application of linear controller design techniques possible. LPV models can be used in the classical control design solutions, however, the use of such models according to Linear Matrix Inequality (LMI) based controller design methodologies are possible, as well. LMIs are powerful mathematical opportunities of controller design techniques.
Almost each control design method can be formulated as a LMI problem and can be solved via iterative numerical processes [17]. In recent years parallel with numerical computational evolution a wide range of LMI applications were discovered and used in control engineering [16, 24].
However, the basic concept behind the LPV-LMI based modeling and control approaches consist in guaranteeing and exploiting the convexity properties. Basically,

this means that it is enough to design such sub-controllers which can deal with the LTI systems in the vertices of the convex polytope, and the convex combination of such controllers can handle each occurring LTI system during operation, if the basic LPV model was appropriate.
In order to use the developed 2-norm based difference as a "metric" on the underlying systems which are determined by parameter vectors inside the PS, several control and mathematical constraints have to be considered. Particular parameter vectors belong to each of the points inside the PS. Since, the parameter vector consist of elements which were multiplied out from the SS model, a parameter vector can determines an underlying system. The key questions are the type of the underlying systems regarding to the parameter vector and how can the parameter vector be used to describe differences among the underlying systems. A few scenarios can be considered dependent on the type of the original and the describing LPV system. The reasonable original system can be NLTV, LTV and LTI beside the describing LPV system (affine or polytopic).

- In LTI-LPV case, each of the points inside the PS is an LTI system and fully determined by the parameter vector.
- In LTV case, a parameter vector determines the underlying system only, if each time-dependent element is selected as scheduling variable.
- In NLTV case, if each time-dependent and nonlinearity causing element is selected as scheduling variable, the parameter vector determines the underlying system.

In all three cases, the parameter vectors determine an underlying LTI system. In NLTV-LTV-LPV case, the original models become simpler. Furthermore, during operation these get around a path inside the PS. The most typical application is when the nonlinearity causing elements are selected as scheduling variables from the NLTV system and the obtained LPV model is used in a LPV-LMI control application. However, in this case, a parameter vector does not determine equivocally the underlying system, since, the time-dependent components can cause hidden differencies, which cannot be seen through the parameter vector.
Assume that the selection of the scheduling variables was appropriate and each parameter vector determines an underlying LTI system. That means the parameter vector-based differences can be interpreted as a "metric" on the occurring LTI systems for which these vectors belong. Namely, instead of the Frobenius-norm based difference in the systems' space the Euclidean-norm based difference in the PS can be used to determine the "difference" between the occurring LTI systems.

$$\|S(\rho_a) - S(\rho_b)\|_F \to \|\rho_a - \rho_b\|_2 = d_2 \ . \tag{24}$$

In the convex polytope, every LTI systems are uniquely specified with their parameter vectors as a consequence of the aforementioned statements. However, the LTI systems in the convex polytope can be calculated as the convex combination of the vertices of the convex polytope. The key question then is the determination of barycentric coordinates ($\alpha_i \in \mathbb{R}^q$) via the uniqueness of the vertices of the given polytope.

Inside the convex polytope each occurring LTI system is over defined, since in a k dimensional parameter space a q pieces coordinate is used to define them, where $q > k$. That means, if the barycentric coordinates are arbitrarily defined, the occurring LTI system description will not be unequivocal, since with the same set of coordinates describes more then one system. In other words, because of the null space problem (differences could occur in the null spaces of the given LTI system) the parameter vector based metric cannot be used as a classic "metric" and cannot be unequivocally interpreted on the LTI systems behind.

Nevertheless, the calculation of the barycentric coordinates ($S = \sum_{i=1}^{q} \alpha_i S_i$) are connected to the selected vertices of the convex polytope and equivocally defined by 15-17. With this condition, the defined metrics can be valid on the occurring LTI systems in case of polytopic LPV systems, as well.

The developed parameter vector based metric can be used in modeling and control as a "quality marker". For example, if a given LPV model is used during identification, the identified model $S(\rho_{ident})$ can be compared to a reference model $S(\rho_{ref})$ in order to estimate the efficiency of the identification procedure. Furthermore, this can be an on-line estimation as well, when the system under identification is described with $S(\rho_{actual}(t_p))$. This procedure can be characterized by the developed $\|d_2\|_2(t)$ instead of the Frobenius-norm based difference. If the goal is to monitor the variation of the system during operation compared to a reference system, the previous solution can be used here as well. Further usabilities and interpretations will be developed in our future work.

Fig.2(b). shows a $2D$ example about the aforementioned interpretations. The PB is the rectangle which is determined by the $\rho_{min,1,2}$ and $\rho_{max,1,2}$ in the PS. Further, this is the validity border of the affine LPV model. At the same time, the rectangle forms a convex polytope. Inside the polytope each occurring LTI system can be calculated as the convex combination of the vertices of the convex polytope. Each aforementioned interpretations and issues are demonstrated on a biomedical engineering example regarding diabetes.

4 Example of the developed approach: Modeling of diabetes

4.1 Selected diabetes model and an LPV form of it

We have selected a simple diabetes model developed for the Intensive Care Unit (ICU) treatment by Wong et al. in [25, 26]. The model is a 3rd order one described by (A-1). The model equations can be handled as in (20-22), which means only the state matrix depends on the parameter vector ($A(\rho(t))$). The main aspect of the model is to describe the glucose-insulin dynamics of an inpatient who suffers from T1DM and is nurtured on the Intensive Care Unit (ICU) [25, 26]. It is expected that this simple model -after preliminary identification-, can provide the current and the future Blood Glucose (BG) level of the patient with a precision that is good enough for the realization of the tight glycemic control (TGC). Detailed description of the

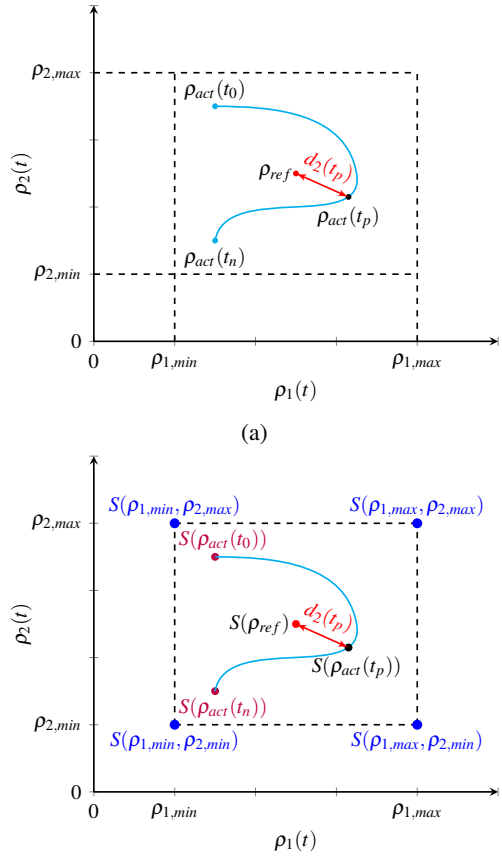


Figure 2
Examples of the possible interpretation of the 2-norm based difference

model can be found in the Appendix of the current paper.

4.1.1 qLPV model of the original Wong model

We selected the following scheduling parameters as the elements of the parameter vector based on our previous work [27]:

$$\rho(t) = \begin{bmatrix} \rho_1(t) \\ \rho_2(t) \\ \rho_3(t) \end{bmatrix} = \begin{bmatrix} \dfrac{S_I Q(t)}{1+\alpha_G Q(t)} \\ \dfrac{S_I G_E}{1+\alpha_G Q(t)} \\ \dfrac{1}{1+\alpha_I I(t)} \end{bmatrix} . \tag{25}$$

It can be seen, that the $\rho(t)$ contains grouped variables. The goal here was to use this LPV model in classical LPV-LMI controller design. Each nonlinearity causing element was selected as scheduling variable. Based on (A-1 and 25), the qLPV model of the original Wong model can be described as follows:

$$A(\rho(t)) = A_0 + A_1\rho_1(t) + A_2\rho_2(t) + A_3\rho_3(t) =$$

$$\begin{bmatrix} -p_G & 0 & 0 \\ 0 & -k & k \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rho_1(t)$$

$$+ \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rho_2(t) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n \end{bmatrix} \rho_3(t) \quad . \tag{26}$$

$$B(\rho(t)) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \dfrac{1}{V_L} & -p_4 I_b \end{bmatrix}$$

$$C(\rho(t)) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

After defining the border of the parameter box, namely the convex polytopic space, the polytopic model form of the qLPV model can be easily obtained based on the affine qLPV form of (26). We have selected tight ranges in every dimension in order to catch the dynamics as precise as possible:

$$\rho(t) = \begin{pmatrix} \rho_1(t) \\ \rho_2(t) \\ \rho_3(t) \end{pmatrix} = \begin{pmatrix} [\rho_1^- .. \rho_1^+] \\ [\rho_2^- .. \rho_2^+] \\ [\rho_3^- .. \rho_3^+] \end{pmatrix} = \begin{pmatrix} [0..5] \\ [0..5] \\ [0..5] \end{pmatrix} \quad . \tag{27}$$

4.2 Presentation of the results

The main goal was to test the usability of the developed "metric" without physiological constraints. In this demonstration we used the consideration of Sec. 3.2.2.
Fig.3. shows the changing of the elements of the parameter vector $\rho(t)$ and the developed 2-norm based difference. On every diagram, the dashed line represents the fixed value, which belongs to the $\rho_{ref} = [0.2,\ \ 0.01025,\ \ 0.98]^T$. It can be seen regarding to the input selected as a symmetrical repeating impulse ($P(t) = 100$ at every $140min$ for $7min$ long and $u_{ex}(t) = 100$ at every $130min$ for $6.5min$ long) that after the first period's decay, the parameter vector have taken the same values in each cycle, which means the same LTI systems occur over time in each cycle.
Fig.4. shows the same signals as Fig.3. on one diagram in order to compare the orders of magnitudes. It can be seen that based on Sec. 3.2.1 those signals that reflect mostly in the $\|d_2\|_2$ have the highest amplitude. Since, in this case the $\rho_1(t)$ has the highest amplitudes, the $\|d_2\|_2$ correlated mostly with $\rho_1(t)$. If each scheduling variables have to be considered with the same weight, normalization procedures

can be done [20]. However, we did not apply such methods as the goal was only demonstration.
If the parameter vectors fully determine the underlying LTI systems during operation, the parameter vector based metric can be used to compare the "difference" between these systems. Fig.5. shows this issue in case of the selected model and parameter vector, namely, instead of the Frobenius-norm based "difference" the developed metric can approximately provides similar results. Naturally, the signals are not the same, since the numerical computations are different. However, the maximum root mean square error (RMSE) was $5.4e^{-5}$.

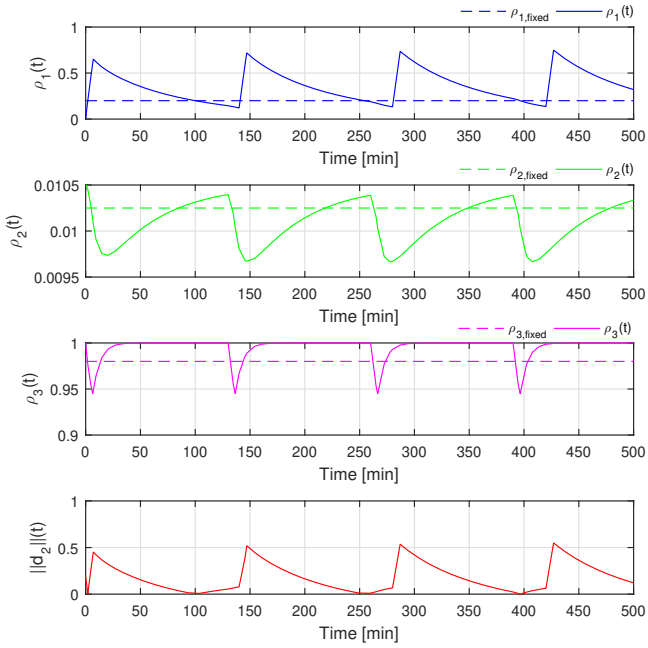


Figure 3
Varying of the scheduling variables and the norm-based error signal

Conclusions

In this study we introduced norm based "difference" interpretation regarding to LPV systems, based on the properties of the LPV parameter space. We have defined how to use these interpretations as error and quality criteria during modeling and control and demonstrated our theoretical findings by a concrete example on diabetes modeling and control of ICU patients. Our future work will focus on the investigation of how this approach can be implemented to the actual LMI based control design methods in order to realize more precise controller for the practice.

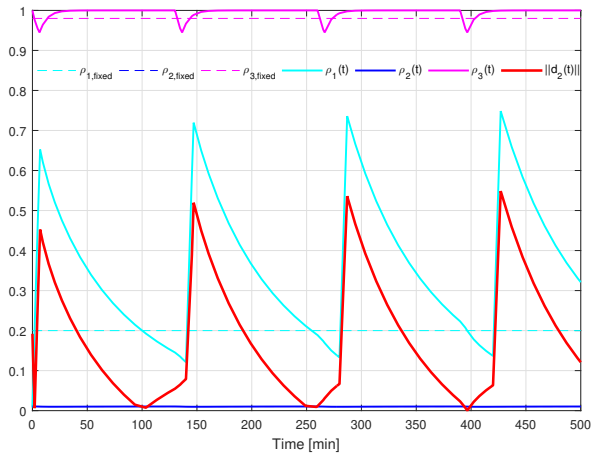


Figure 4
Comparison of the magnitudes of the scheduling variables and the norm-based error signal

Acknowledgement

Gy. Eigner thankfully acknowledge the support of the Robotics Special College of Obuda University. L. Kovács is also supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. The research was also supported by the Research and Innovation Center of Obuda University.

References

[1] V.N. Shah, A. Shoskes, B. Tawfik, and S.K. Garg. Closed-loop system in the management of diabetes: Past, present, and future. *Diabetes Technol The*, 16(8):477 – 490, 2014.

[2] S. Kamath. Model based simulation for type 1 diabetes patients. *Asian Am J Chem*, 1(1):11 – 19, 2013.

[3] G. Marchetti, M. Barolo, L. Jovanovic, H. Zisser, and D.E. Seborg. An improved PID switching control strategy for type 1 diabetes. *IEEE T Bio-Mde Eng*, 55:857 – 865, 2004.

[4] G. Schlotthauer, L.G. Gamero, M.E. Torres, and G.A. Nicolini. Modeling, identification and nonlinear model predictive control of type i diabetic patient. *Med Eng Phys*, 28:240 – 250, 2006.

[5] P. Maxime, H. Gueguen, and A. Belmiloudi. A robust receding horizon control approach to artificial glucose control for type 1 diabetes. *Nonlin Contr Sys*, 9(1):833 – 838, 2013.

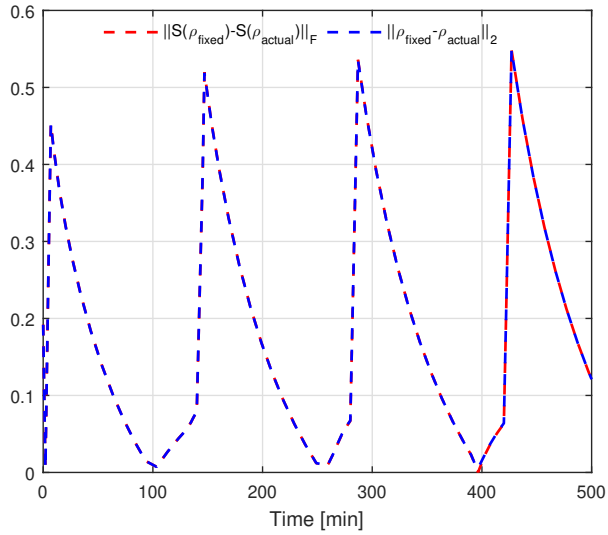


Figure 5
Different norm-based differences

[6] P. Herrero, P. Georgiou, N. Oliver, D.G. Johnston, and C. Toumazou. A bio-inspired glucose controller based on pancreatic β-cell physiology. *J Diabetes Scien Technol*, 6(3):606–616, 2012.

[7] E. Atlas, R. Nimri, S. Miller, E.A. Grunberg, and M. Phillip. Md-logic anrtificial pancreas system. a pilot study in adults with type 1 diabetes. *Diab Care*, 33:1072–1076, 2010.

[8] P. Szalay, Gy. Eigner, and L. Kovács. Linear matrix inequality-based robust controller design for type-1 diabetes model. In *IFAC 2014 – 19th World Congress of The International Federation of Automatic Control*, pages 9247 – 9252. IFAC, 2014.

[9] L. Kovács, P. Szalay, Zs. Almássy, and L. Barkai. Applicability results of a nonlinear model-based robust blood glucose control algorithm. *J Diabetes Scien Technol*, 7(3):708 – 716, 2013.

[10] P. Colmegna and S. Peña. Analysis of three T1DM simulation models for evaluating robust closed-loop controllers. *Comput Meth Prog Bio*, 113(1):371 – 382, 2014.

[11] S.S. Hacısalihzade. *Biomedical Applications of Control Engineering*. Springer-Verlag, Berlin, 1st edition, 2013.

[12] L. Kovács, B. Benyó, J. Bokor, and Z. Benyó. Induced l_2-norm minimization of glucose–insulin system for type I diabetic patients. *Comput Meth Prog Bio*, 102(2):105 – 118, 2011.

[13] C. Scherer and S. Weiland. Lecture Notes DISC Course on Linear Matrix Inequalities in Control. Delft University, 1999.

[14] P. Szalay, Gy. Eigner, M. Kozlovszky, I. Rudas, and L. Kovács. The significance of LPV modeling of a widely used T1DM model. In *EMBC 2013 – 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3531 – 3534. IEEE EMBS, 2013.

[15] D.W. Gu, P.H. Petkov, and M.M. Konstantinov. *Robust Control Design with Matlab*. Springer, London, 2nd edition, 2013.

[16] A.P. White, G. Zhu, and J. Choi. *Linear Parameter Varying Control for Engineering Applicaitons*. Springer, London, 1st edition, 2013.

[17] O. Sename, P. Gáspár, and J. Bokor. Robust control and linear parameter varying approaches, application to vehicle dynamics. volume 437 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, 2013.

[18] B. Takarics and P. Baranyi. TP model-based robust stabilization of the 3 degrees-of-freedom aeroelastic wing section. *ACTA Polytech Hung*, 12(1):209 – 228, 2015.

[19] B. Takarics and P. Baranyi. Friction compensation in TP model form - aeroelastic wing as an example system. *ACTA Polytech Hung*, 12(4):127 – 145, 2015.

[20] W.S. Levine. *The Control Engineering Handbook*. CRC Press, Taylor and Francis Group, Boca Raton, 2nd edition, 2011.

[21] F.A. Möbius. *The Barycentric Calculus (in German)*. Verlag von Johann Ambrosius Barth, Leipzig, 1st edition, 1827.

[22] J. Fauvel, F. Raymond, and R. Wilson, editors. *Möbius and his Band*. Oxford University Press, Oxford, 1st edition, 1993.

[23] J.K Tar, J.F Bitó, L. Nádai, and J.A.T Machado. Robust Fixed Point Transformations in Adaptive Control Using Local Basin of Attraction. *ACTA Pol Hung*, 6(1):21–37, 2009.

[24] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics. SIAM, Philadelphia, 1st edition, 1994.

[25] X.W. Wong, J.G. Chase, G.M. Shaw, C.E. Hanna, T. Lotz, J. Lin, I. Singh-Levett, L.J. Hollingsworth, and O.S.W. Wong. Model predictive glycaemic regulation in critical illness using insulin and nutrition input: A pilot study. *Med Eng Phys*, 28:665 – 681, 2006.

[26] X.W. Wong, J.G. Chase, G.M. Shaw, C.E. Hann, J. Lin, and T. Lotz. Comparison of adaptive and sliding-scale glycaemic control in critical care and the impact of nutritional inputs. In *12th International Conference On Biomedical Engineering*, pages 1 – 4, 2005.

[27] L. Kovács, A. György, B. Kulcsar, P. Szalay, B. Benyo, and Z. Benyo. Robust control of type 1 diabetes using μ-synthesis. In *UKACC International Conference on Control 2010*, pages 1 – 6, 2010.

Appendix

The Wong-model consist of the following equations [25, 26]:

$$\begin{aligned}\dot{G}(t) &= -p_G G(t) - S_I(G(t)+G_E)\frac{Q(t)}{1+\alpha_G Q(t)} + P(t)\\ \dot{X}(t) &= -kI(t) - kQ(t)\\ \dot{I}(t) &= -\frac{nI(t)}{1+\alpha_I I(t)} + \frac{u_{ex}(t)}{V}\end{aligned} \tag{A-1}$$

The following table contains the parameters, their descriptions and their values which were used in this study regarding to the Wong-model [25, 26].

Table A-1
Detailed desriptions and values of the parameters of the Wong-model

Notation	Unit	Description	Value
G	$mmol/L$	Plasma glucose above equilibrium level	-
Q	mU/L	Concentration of insulin bounded to interstitial sites	-
I	mU/L	Plasma insulin resulting from external input	-
P	$mmol/L/min$	Total plasma glucose input	-
u_{ex}	mU/min	External insulin input	-
G_E	$mmol/L$	Plasma equilibrium level	10.5
p_G	$1/min$	Endogenous glucose clearance	0.01
S_I	$L/mU/min$	Insulin sensitivity	0.001
V	L	Insulin distribution volume	12
k	$1/min$	Effective life of insulin in the compartment	0.0198
n	$1/min$	First order decay rate from plasma	0.16
α_I	L/mU	Plasma insulin disappearance	0.0017
α_G	L/mU	Insulin effect	0.0154

A Review of Bilateral Teleoperation Algorithms

Riccardo Muradore and Paolo Fiorini

Riccardo Muradore and Paolo Fiorini are with the Department of Computer Science, University of Verona, Strada Le Grazie 15, 37134 Verona, Italy.
riccardo.muradore@univr.it, paolo.fiorini@univr.it

Abstract: Bilateral Teleoperation is a key technology to allow humans to interact with remote environments by providing the operator with haptic feedback. Haptic feedback from the one hand improves human perception and therefore the quality of the human-robot interaction, on the other hand it can tamper with the stability of the system when the communication between the master side (where the operator is) and the slave side (where the remote robot interacts with the environment) is not instantaneous but affected by delay and packet drops. In the last 40 years many algorithms have been developed to guarantee the stability of haptic teleoperation in the presence of time delay, many of them based on passivity theory. In this paper we review and compare a few algorithms that are representative of the tools in the frequency or in the time domains that have been used to develop a safe and transparent physical human-robot interaction with unknown environments.

Keywords: Bilateral teleoperation, Passivity, Communication delay, Haptics, Force reflection.

1 Introduction

Teleoperation of mechanical arms marked the beginning of robotics development and it has been an important component of this field ever since. Teleoperation systems allow humans to interact with remote environments by providing the operator with sensory feedback similar to that s/he would experience as if they were at the remote site. To achieve full sensory feedback, teleoperation systems should communicate also use contact force/torque information, from the slave to the master side to improve human perception and understanding, improve task performance and achieve the telepresence. The specific teleoperation configuration, in which, the kinesthetic coupling between operator and environment is enhanced by dynamic coupling, is referred to as *bilateral* teleoperation, since it provides *force feedback*. This has been one of the fields pioneered by *Antal (Tony) Bejczy* and this review aims at showing the wide legacy of his initial research.

Current examples of teleoperation span from space applications [1, 2] to rehabilitation, to surgery [3] and to Unmanned Air/Ground Vehicles (UAV, UGV).

A large number of papers have been written on the various aspects of teleoperation as shown in many survey papers [4, 5, 6], thus indicating a significant interest in the field; however these survey papers present mostly the analytical aspects of the reviewed algorithms without comparison and discussion of experimental results.
To fill this gap, we have added to our research in teleoperation, a series of experiments to analyse in detail the most common teleoperation algorithms with the goal of developing a unified classification of the field and of formalising a curriculum in teleoperation studies that could be of benefit to the whole robotics community. We focus this review on *bilateral teleoperation systems* in which the master and the slave devices are connected through a packet-based communication channel which delivers the signals, such as commands from the masters and measurements from the slave. Although this architecture is not the most commonly used in real systems, because of the potential instability due to force feedback in the presence of commu-nication time delay, it surely must be analyzed and discussed because it provides the operator the full perception of the remote side [7, 8, 9].
In this paper, we present the initial results of our classification and analysis of the hundreds of teleoperation algorithms available in the scientific literature. We focus on algorithms that guarantee stability of the bilateral teleoperation system and we group them according the time delay present in the communication channel. For each algorithm we show its block diagram and its basic equations.

In section 2 we discuss one algorithm for the communication without time delay. In section 3 we present two algorithms that compensate an unknown but constant communication time delay. Finally, in Section 4, we present three algorithms that support force feedback in the presence of variable communication time delay and of data packet losses. Section 5 describes the experimental set-up on which we compare the performance of the various algorithms, some of the hardware calibration details and the results of some experiments. It must be noted that the experimental data presented, were the results of the work of the students following the course in Advanced Robotics offered at our University. Finally, Section 6 summarizes the paper and presents our plans to continue the analysis of teleoperation algorithms.

2 Bilateral teleoperation with no communication delay

Even if most teleoperation algorithms face communication delays, there are many important applications where the teleoperated system uses dedicated communication channels (e.g. Da Vinci surgical robot, ESA system), without any time delay noticeable by the user.

2.1 Four channel teleoperation architecture

Among the many algorithms that consider the bilateral teleoperation without communication delay we analyse the one proposed by Lawrence in 1993, the *Four channel architecture* [10].

The teleoperation system is completely transparent if the operator feels that s/he is directly interacting with the remote environment (equal forces $f_m = f_s$ and velocities $v_m = v_s$). Transparency requires that the transmitted impedance Z_t is equal to the environment impedance $Z_e = \frac{f_s}{v_s}$. According to the block diagram in Figure 1 we have to design the blocks $C_s, C_m, C_1, \ldots C_4$ in such a way that the hybrid matrix H

$$\begin{bmatrix} f_m(s) \\ v_m(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \begin{bmatrix} v_s(s) \\ f_s(s) \end{bmatrix} \tag{1}$$

is equal to the matrix $\begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$. This implies

$$Z_t = \frac{f_m}{v_m} = (H_{11} - H_{12}Z_e)(H_{21} - H_{22}Z_e)^{-1} = Z_e. \tag{2}$$

To achieve this goal it is necessary to have very accurate models of the master and slave robots, otherwise the transparency condition will not be satisfied. However it is worth highlighting, that good transparency is important mainly at low frequencies, i.e. where the operator is working and where, fortunately, the mathematical models are more accurate.

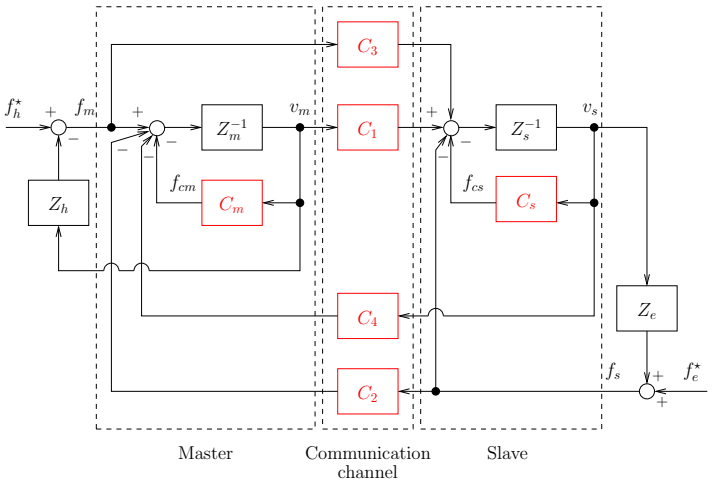


Figure 1
Four channel force-velocity architecture. Legend: Z_h operator arm impedance, Z_m^{-1} master robot admittance, Z_e environment impedance, Z_s^{-1} slave robot admittance; C_m master local controller, C_s slave local controller, $C_1, \ldots, C_4$ communication link transfer functions; $f_h^\star$ operator intentional force, f_m, v_m force and velocity at the master side, f_s, v_s force and velocity at the slave side, $f_e^\star$ exogenous force at the remote site, f_{cm}, f_{cs} force controls at the master and slave side, respectively.

2.1.1 Discussion

The main tasks when using this algorithm are:

- Analyse the impact of the discretisation on transparency

- Identify the real values of the elements in H by using time series of velocities and forces, and compare them with the ideal values

- Consider the effects of model uncertainty and static friction

3 Bilateral teleoperation with constant and unknown communication delay

In several teleoperation systems the communication delay cannot be neglected but it can be assumed constant but unknown. The following approaches exploit the constant delay hypothesis to design control architectures that guarantee the stability of the system independently of the delay and the interaction with passive environments.

3.1 Wave variables and Scattering transformation

The architecture proposed in [11] transmits wave variables $\{u_m, w_m\}, \{u_s, w_s\}$ instead of power variables $\{f_m, v_m\}, \{f_s, v_s\}$ to guarantee the passivity of the system. The wave transformation relating wave and power variables is given by

$$u_m = \frac{1}{\sqrt{2b}}(f_m + bv_m), \qquad u_s = \frac{1}{\sqrt{2b}}(f_s - bv_s), \tag{3}$$

$$w_m = \frac{1}{\sqrt{2b}}(f_m - bv_m), \qquad w_s = \frac{1}{\sqrt{2b}}(f_s + bv_s) \tag{4}$$

where b is the characteristic impedance. The value of b is a design parameter. Figure 2 shows the architecture where we exploit the fact that the wave transformation is invertible and so the power variables can be retrieved from the wave variables

$$f_m = \sqrt{\frac{b}{2}}(u_m + w_m), \qquad f_s = \sqrt{\frac{b}{2}}(u_s + w_s), \tag{5}$$

$$v_m = \frac{1}{\sqrt{2b}}(u_m - w_m), \qquad v_s = -\frac{1}{\sqrt{2b}}(u_s - w_s). \tag{6}$$

Due to the communication delay τ, we have

$$w_s(t) = u_m(t-\tau) \tag{7}$$

$$w_m(t) = u_s(t-\tau). \tag{8}$$

It is worth remarking that when $\tau = 0$, transmitting wave variables is the same to transmit power variables.

The control architecture consists also of a velocity controller at the slave side whereas the force measured f_s is sent to the haptic devices for force rendering.

The solution of the time delay problem proposed in this algorithm shows why time delay may generate energy (and so destabilise the system) and how the wave transformation solves this problem [11].

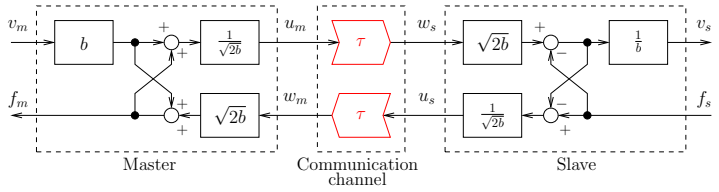


Figure 2
Wave variables. Legend: $\{u_m, w_m\}, \{u_s, w_s\}$ wave variables at master and slave side; $\{f_m, v_m\}, \{f_s, v_s\}$ power variables at master and slave side; τ constant communication delay; b characteristic impedance.

3.1.1 Discussion

The main aspects in the implementation of this algorithm are:

- The system at the master side behaves in a sluggish way also when the slave is in free motion. This behavior is function of the communication delay τ e the value of b.
- The parameter b should be tuned according to the delay τ.
- The transparency of the scheme in Figure 2 should be compared to the one having termination elements to match the impedances, which also introduce scaling in force (master side) and velocity (slave side).

3.2 PD and passivity terms

The solution proposed by Lee and Spong guarantees the stability of a position-position bilateral teleoperation by adding a dissipative term to the PD controller at each side of the architecture [12].

As shown in Figure 3 the master and slave controllers are given by

$$u_m(t) = \underbrace{-K_p\tilde{x}_m(t) - K_v\tilde{v}_m(t)}_{f_{cm}} - \underbrace{(K_{dis}+P_\varepsilon)v_m(t)}_{f_{pm},\text{dissipation}} \tag{9}$$

$$u_s(t) = \underbrace{-K_p\tilde{x}_s(t) - K_v\tilde{v}_s(t)}_{f_{cs}} - \underbrace{(K_{dis}+P_\varepsilon)v_s(t)}_{f_{ps},\text{dissipation}} \tag{10}$$

where $\tilde{x}_m(t) := x_s(t-\tau_{s2m}) - x_m(t), \tilde{v}_m(t) := v_s(t-\tau_{s2m}) - v_m(t)$ and $\tilde{x}_s(t) := x_m(t-\tau_{m2s}) - x_s(t), \tilde{v}_s(t) := v_m(t-\tau_{m2s}) - v_s(t)$ are the position/velocity tracking errors at the master and slave side, respectively. The gain K_{dis} is needed to ensure passivity of the P-control action ($K_{diss} = \frac{\bar{\tau}_{RTT}}{2}K_p$, where $\bar{\tau}_{RTT}$ is an upper bound of the round trip time $\bar{\tau}_{RTT} \geq \tau_{RTT} := \tau_{m2s} + \tau_{s2m}$) and P_ε adds additional damping to guarantee the master-slave position coordination.

3.2.1 Discussion

The main issues to be considered when using this algorithm are:

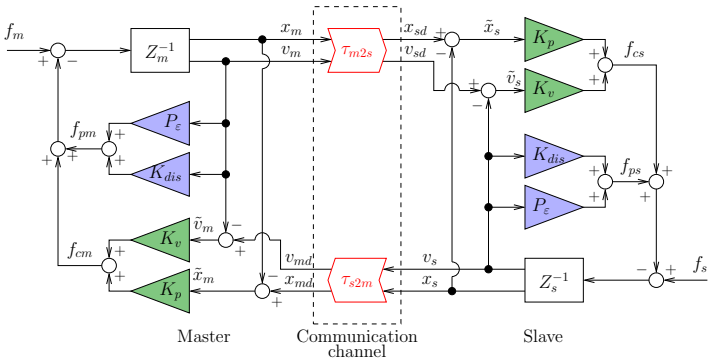


Figure 3
PD and passivity terms. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; K_p, K_v position and velocity gains of the PD controller; K_{dis}, P_ε dissipative terms; f_m, x_m, v_m force, position and velocity at the master side, f_s, x_s, v_s force, position and velocity at the slave side; f_{cm}, f_{cs} force controls at the master and slave side, f_{pm}, f_{ps} passivity-related forces.

- The effect of the discretization and of the velocity estimation.
- The speed of the system response as a function of K_{dis} (and so of K_p and τ_{RTT}).
- The effect of the values of K_p and K_v at the master and slave sides.
- The effect of P_ε on the level of transparency.
- The on-line adaptation of K_{dis} according to the measured delay.

3.3 Adaptive algorithm

The algorithm presented in this section for the constant communication delay scenario makes use of adaptive control to guarantee the passivity of the system [13]. In this algorithm, the PD controllers of the previous approach are replaced by an equivalent proportional controller on the new variables:

$$r_m(t) := v_m(t) + \lambda x_m(t) \tag{11}$$
$$r_s(t) := v_s(t) + \lambda x_s(t) \tag{12}$$

where $\lambda > 0$ is a tuning parameter. The overall architecture is shown in Figure 4 where the *Adaptive Estimation* blocks estimate the inertia and damping parameters of the DC motor at the slave and master side using an adaptive algorithm.

$$\begin{bmatrix} \dot{\hat{J}}_m(t) \\ \dot{\hat{B}}_m(t) \end{bmatrix} = \Gamma_m Y_m^T(x_m(t), r_m(t)) r_m(t) \tag{13}$$
$$\begin{bmatrix} \dot{\hat{J}}_s(t) \\ \dot{\hat{B}}_s(t) \end{bmatrix} = \Gamma_s Y_s^T(x_s(t), r_s(t)) r_s(t) \tag{14}$$

where Γ_m, Γ_s are constant positive definite matrices and Y_m, Y_s are the regression matrices.

The overall master and slave controls are computed as:

$$u_m(t) = f_{mc}(t) - \hat{f}_m(t)\lambda v_m(t) - \hat{B}_m(t)\lambda x_m(t) \quad (15)$$

$$u_s(t) = f_{sc}(t) - \hat{f}_s(t)\lambda v_s(t) - \hat{B}_s(t)\lambda x_s(t) \quad (16)$$

where $f_{mc}(t)$ and $f_{sc}(t)$ are the coordinating torques

$$f_{mc}(t) = K(r_m(t) - r_s(t - \tau_{s2m})) \quad (17)$$

$$f_{sc}(t) = K(r_s(t) - r_m(t - \tau_{m22})). \quad (18)$$

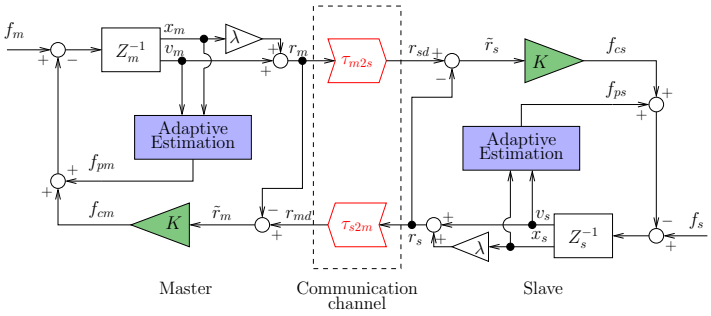


Figure 4
Adaptive-based algorithm. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; K gain of the controller; f_m, x_m, v_m force, position and velocity at the master side, f_s, x_s, v_s force, position and velocity at the slave side; λ tuning parameter; f_{cm}, f_{cs} force controls at the master and slave side, f_{pm}, f_{ps} passivity-related forces.

3.3.1 Discussion

The main steps for the practical implementation of this algorithm are:

- The effect of the discretization on the adaptive estimation.
- System transparency as a function of K, λ, Γ_m, Γ_s.
- The computation of the value of K, assuming to exactly know the dynamic parameters of the motors, and of λ to ensure the equivalence of this algorithm to the algorithm presented in the previous section.

4 Bilateral teleoperation with time-varying communication delay

In this section we address the case of bilateral teleoperation when the communication delay is time-varying and unknown. We present three solutions that share the

same idea: the passivity is guaranteed in the time domain by computing the energy balance at run time.

4.1 Time Domain Passivity Approach

The Time Domain Passivity Approach (TDPA) proposed in [14] uses two tools to evaluate the energy and to cope with energy shortage:

- The passivity observers PO compute the monotonically increasing energy leaving or entering the master and the slave side

$$E^*_{in}(k) = \begin{cases} E^*_{in}(k-1) + T_s P^*(k), & \text{if } P^*(k) > 0 \\ E^*_{in}(k-1), & \text{if } P^*(k) \leq 0 \end{cases} \tag{19}$$

$$E^*_{out}(k) = \begin{cases} E^*_{out}(k-1) - T_s P^*(k), & \text{if } P^*(k) < 0 \\ E^*_{out}(k-1), & \text{if } P^*(k) \geq 0 \end{cases} \tag{20}$$

  where $* = m, s$, $P^*(k) = f_*(k)v_*(k)$ is the actual discrete-time power computed at time $t = kT_s$, with T_s the sample time.

- The passivity controllers PC intervene any time an unstable behavior is going to be applied by activating dissipative elements

$$\beta_k = \begin{cases} \frac{\Delta E_s(k)}{T_s f_{sd}^2} & \text{if } \Delta E_s(k) > 0 \text{ and } f_{sd} \neq 0 \\ 0 & \text{if } \Delta E_s(k) \leq 0 \end{cases} \tag{21}$$

$$\alpha_k = \begin{cases} \frac{\Delta E_m(k)}{T_s v_{md}^2} & \text{if } \Delta E_m(k) > 0 \text{ and } v_{md} \neq 0 \\ 0 & \text{if } \Delta E_m(k) \leq 0 \end{cases} \tag{22}$$

  where

$$\Delta E_s(k) := E^s_{out}(k) - E^m_{in}(k - \tau_{m2s}(k)) \tag{23}$$

$$\Delta E_m(k) := E^m_{out}(k) - E^s_{in}(k - \tau_{s2m}(k)). \tag{24}$$

Figure 5 shows this architecture where also low pass filters F are implemented in a bi-directional fashion (to preserve passivity) to reduce bumps and oscillations at the operator side.

4.1.1 Discussion

The following are important considerations that a designer using this algorithm should consider:

- How conservative is this "distributed" energy control.
- The effect of the communication delay on the performance of the TDPA algorithm.
- If and how to implement a position-position teleoperation instead of the proposed position-force scheme.

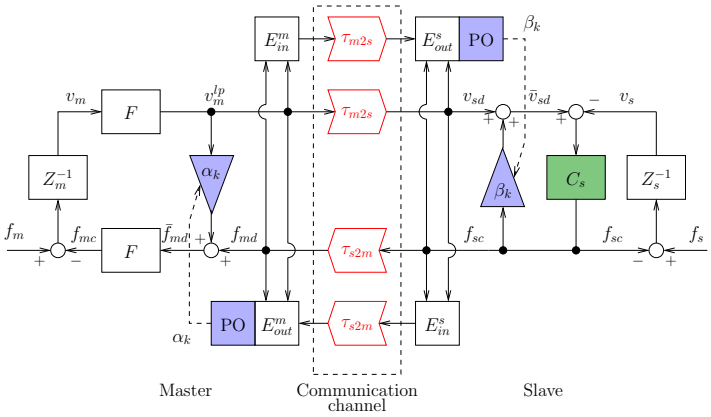


Figure 5
TDPA algorithm. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; C_s velocity controller; f_m, v_m force and velocity at the master side, f_s, v_s force and velocity at the slave side; $v_{sd}(t) = v_m^{lp}(t - \tau_{m2s})$ desired velocity at the slave side with $v_m^{lp} = F(s)v_m$, $\bar{v}_{sd}$ modified velocity reference, $f_{md}(t) = f_{sc}(t - \tau_{s2c})$ desired force, $\bar{f}_{md}$ modified force reference with $f_{md} = F(s)\bar{f}_{md}$.

4.2 Passive Set-Position Modulation

The Passive Set-Position Modulation (PSPM) algorithm is another approach to modify the value of the reference signal received from the master or the slave to comply with the passivity constraint.

The scheme proposed in [15] is a position-position teleoperation architecture where the original set-position signal (xmd or xsd) is modulated in such a way that the new value (x¯md or x¯sd) satisfies the passivity constraint when applied to a spring $K_p(x_*(t) - \bar{x}_{*d}(k))$ with damping injection $K_v v_*$, $* = m, s$.

The novelty of this approach is to explicitly consider the possibly of passivity-breaking due to spring energy jumps at the switching instances. The scheme is shown in Figure 6 where E_m is the virtual energy reservoirs at the master, and the PSPM block solves the following minimization problem any time a new reference value is received:

$$\begin{aligned} &\min_{\bar{x}_{md}(k)} && \|x_{md}(k) - \bar{x}_{md}(k)\| \\ &s.to && E_m(k-1) + \Delta E_s(k) + D_m(k-1) - \Delta \bar{P}_m(k) \geq 0 \end{aligned} \tag{25}$$

where $E_m(k-1)$ is the available energy, $\Delta E_s(k)$ is the energy received by the slave (*energy-shuffling*), $D_m(k-1)$ is the causal approximation of the damping dissipation (*energy reharvesting*, $\frac{1}{2}K_v x_m^2(k)$) and $\Delta \bar{P}_m(k)$ is the spring energy jump

$$\Delta \bar{P}(k) = \frac{1}{2}\|x_m(k) - \bar{x}_{md}(k)\|^2_{K_p} - \frac{1}{2}\|x_m(k) - \bar{x}_{md}(k-1)\|^2_{K_p} \tag{26}$$

In this algorithm, the local controllers are assumed continuous: in real implementation we can only assume that the local controllers have a sample time T_s smaller than the sample time at which the slave and master send their commands/measurements

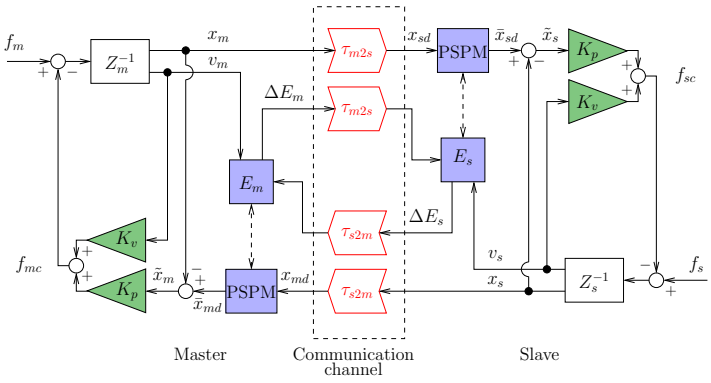


Figure 6
PSPM algorithm. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; K_p, K_v position and velocity gains; f_m, x_m, v_m force, position and velocity at the master side, f_s, x_s, v_s force, position and velocity at the slave side; $x_{md}, \bar{x}_{md}$ desired and modulated reference position at the master side; $x_{sd}, \bar{x}_{sd}$ desired and modulated reference position at the slave side; f_{mc}, f_{sc} force controls at the master and slave side.

to the other side of the bilateral architecture. The PSPM at the master side can also decide to send energy to the slave side (ΔE_m) by checking the current energy (*energy ceiling*).

Exactly the same approach works at the slave side proving that this architecture is symmetric as shown in Figure 6. The idea is to compute x_{md} (x_{sd}) in such a way that its value is as close to $\bar{x}_{md}$ ($\bar{x}_{sd}$) as possible (transparency) but without violating the passivity constraint (stability).

4.2.1 Discussion

Since the original formulation assumes that the local controllers are continuous and only the communication channel is discrete-time, in a real implementation one should consider that also the controllers are discrete-time, and re-evaluate the performance of the algorithm as a function of the ratio of the sample time of the network and of the controllers. Other important aspects to be considered are:

- The effect of the size of ΔE_m and of ΔE_s on the transparency of the algorithm.
- The use of different techniques to solve in an efficient way the minimization problem.

4.3 A two-layer approach

The last algorithm discussed implements a hierarchical two-layer approach: the top layer (transparency layer) is used to implement the control strategy that better fits the performance requirement, and the lower layer (passivity layer) ensures that no "virtual" energy is generated [16].

The scheme is reported in Figure 7 for a position-force configuration. It is interesting to see that the approach of this algorithm is similar to the one of the two previous solutions but the passivity verification is done after the computation of the commands and not before.

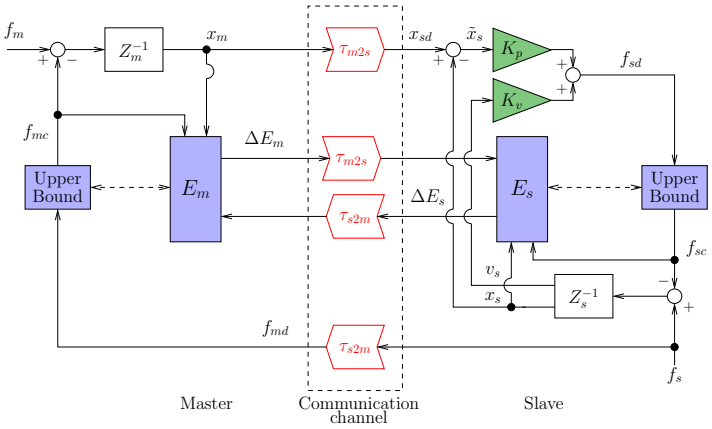


Figure 7
Hierarchical two-layer approach. Legend: Z_m^{-1} master robot admittance, Z_s^{-1} slave robot admittance; K_p, K_v position and velocity gains; f_m, x_m, v_m force, position and velocity at the master side, f_s, x_s, v_s force, position and velocity at the slave side; f_{md}, f_{mc} desired and modulated forces at the master side; f_{sd}, f_{sc} desired and modulated force at the slave side.

The level of the energy tank E_m (the same hold for E_s) consists of three terms

$$E_m(k) = E_m(k-1) + E_{s2m}(k) - E_H(k) \tag{27}$$

where $E_m(k-1)$ is the past value of the energy, $E_{s2m}(k) = \Delta E_s(k - \tau_{s2m})$ is the energy received by the slave side and $E_H(k)$ is the energy exchange between the discrete-time controller and physical world. It is computed as

$$E_H(k) = u_m(k)(x_m(k) - x_m(k-1)) \tag{28}$$

where $u_m(k)$ is the torque applied to the motor over the interval $[k-1, k]$.

This energy is used to compute an upper bound for the command to guarantee the stability of the teleoperated system, e.g. $u_m^{max} = \frac{E_m(k)}{\hat{v}_m(k)T_s}$ where T_s is the sample time of the local controller and $\hat{v}_m(k)$ is an estimation of the future velocity.

If $E_m(k)$ is large enough the master sends energy $E_{m2s}(k)$ to the slave side and updates $E_m(k)$ accordingly

$$E_m(k) \leftarrow E_m(k) - E_{m2s}(k). \tag{29}$$

The same architecture is implemented at the slave side as shown in Figure 7.

4.3.1 Discussion

Additional points to be considered are:

- The implementation of a position-position structure within the hierarchical two-layer approach.
- The use of different policies for the master-slave energy exchange.
- The amount of energy present "in the network" as a function of τ_{m2s}, τ_{m2s} and the energy transfer policy.

5 Experimental setup

The experimental setup is shown in Figure 8 and it supports the implementation of C++ different types of control architecture in a real time middleware. This teleoperation architecture has one degree of freedom: this simplifies the analysis because we can avoid the complication of nonlinear control of multi-input multi-output manipulators, and is also more suitable to educational purpose and to compare the algorithms' characteristics.

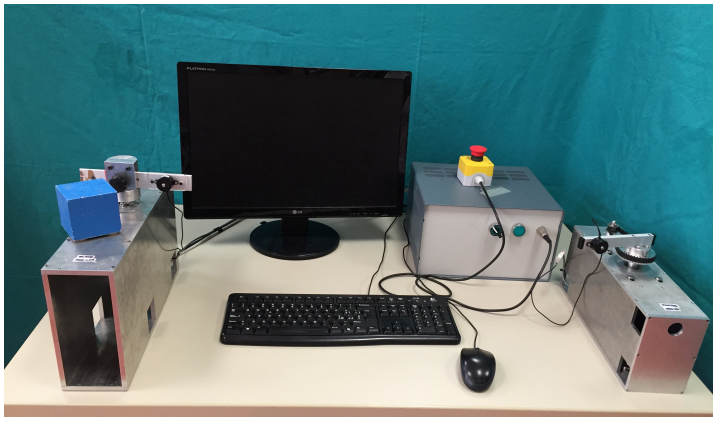


Figure 8
Bilateral teleoperation test bench.

The hardware consists of two DC motors, a gear at the master side with ratio equal to 4, two single axis force sensors mounted on rods, and a control board. The board has an interface to a Beckhoff™ EtherCAT® controller. The motor controllers are logically separated at software level sharing only the physical communication bus.

The software is developed in OROCOS [17], an open source component-based frame-work for robotic applications developed by K.U. Leuven, Belgium, in collaboration with LAAS Toulouse, France, and KTH, Sweden. It allows program flexibility (in C++), real time performance and code reusability.

The software runs onto a low latency Linux kernel patched for real time-preemption whereas the communication with the board is handled by RTnet [18]. This results in a stable computational platform capable of executing any teleoperation algorithm.

The main software component is called *E-Board* and interacts with the physical board by using the SOEM library [19]. Each side of the teleoperation architecture is represented by a logical abstraction of the control and of the DC motor and are indicated with *Master Side* and *Slave Side* in Figure 9. Both components receive the current angular positions from the board and send voltage commands to the motors' amplifiers. The master and slave sides exchange control/measurement data through two instances of a component that simulates a network connection by delaying and/or losing packets. The network component queues packets and assigns them a delay value. The delay can be constant or random; we implemented the following probability distributions: uniform (between an upper and lower bounds), Gaussian or exponential. When the current packet is received, the queue content is sorted according to the corresponding delay and the last valid packet is written to the output port. It is worth highlighting that in case of multiple packets with the same exit time (receiving time + communication delay), the queue is emptied of any packet with equal exit time and the one with the largest packet identifier (pktid) is sent out.

The state of master and slave is contained within a *struct* type called *DataStructure* (indicated as "D" in Figure 9):

```
struct DataStructure
{
    unsigned int pktid;
    char sender;
    double joint;
    double voltage;
    double speed;
    double absorbedCurrent;
    double timestamp;
    double genericField1;
    double genericField2;
    double genericField3;
    double genericField4;
};
```

The *Generic Fields* within the *DataStructure* are needed to exchange further information between the master and slave sides according to the particular solution under study (e.g. wave variables, energy quanta).

6 Results

In this section we show some plots of data collected during the test and comparison of the algorithms.

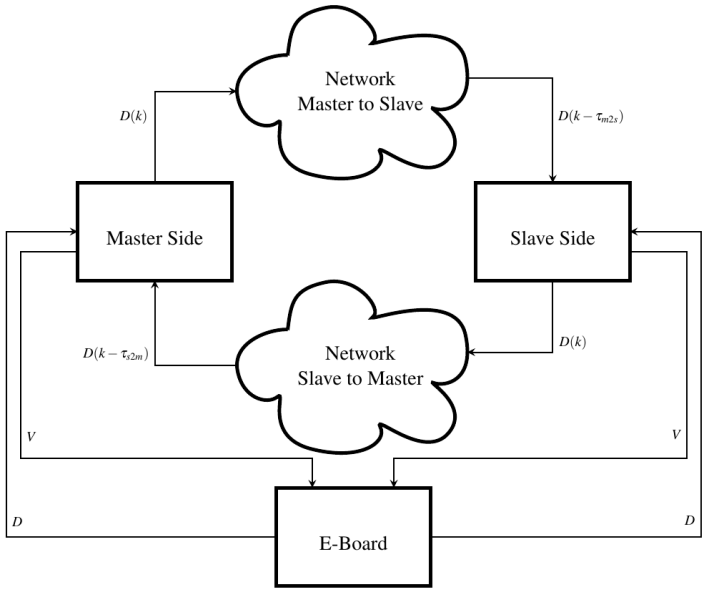


Figure 9
Components organisation.

6.1 Force sensors tuning

The test bench is equipped with two single-axis force sensors at the master and slave side. Some of the bilateral teleoperation algorithms explicitly use force measurements to provide *direct* force feedback to the user, whereas others provide an *indirect* force feedback related to the displacement between master and slave positions. The second set of algorithms requires fewer sensors but generates force feedback to the user even if the slave is not in contact with the environment.

An important step is to relate the measurement force to the motor torque and to the voltage command (ignoring the dynamics of the electrical subsystem). Figure 10 shows the linear interpolation relating voltages and force measurements. It is interesting to see that the force sensors have an important negative bias. Even though the sensors are the same, the slopes are different because of the gear ratio at the master side ($N = 4$). Moreover there is a "dead zone" around zero. This is not due to the sensor but to the high static friction of the motors. Voltages below 0.5 V do not move the motors. These force sensors have also a rather high quantization error (around 0.15 N) that will be clearly visible in Section 6.2.

6.2 Experiments

Figure 11 shows the behavior of the algorithm described in Section 3.2 when the slave gets in contact with the environment ($\sim$3.1s). The bilateral teleoperation system remains stable despite the communication delay and the hard contact. The

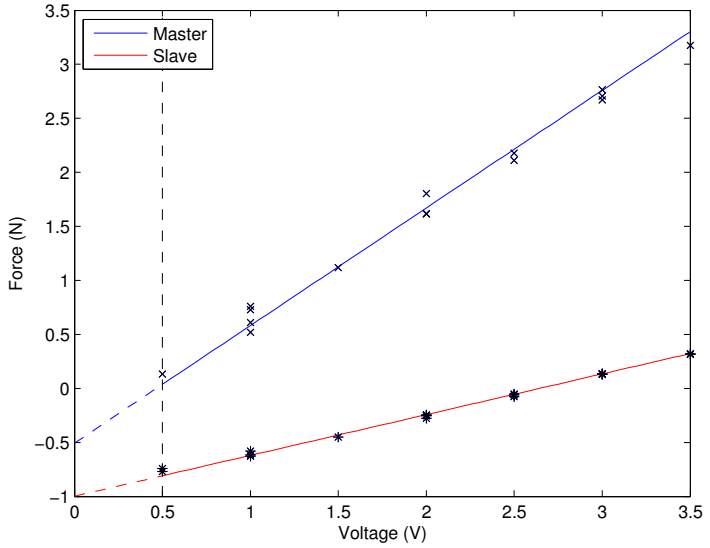


Figure 10
Calibration of the force sensors.

voltages at the master and slave side have very similar values (in absolute value but opposite signs) and are strongly correlated to the force measurements at the master side.

After the contact, the position of the slave does not change whereas the master force is proportional to the difference between the master and slave positions. The temporal misalignment that can be seen in the master and slave voltages is mainly due to the round trip time $\tau_{m2s} + \tau_{s2m}$ and partially on the haptic paddle dynamics and the control parameters (K_p, K_v and K_{dis}, P_ε). High values for τ_{m2s}, τ_{s2m} imply a high value for K_{dis}: the system behaves in a sluggish manner. This is the price to pay to have the system passive with high round trip time.

The plots in Figure 12 show how the PSPM algorithm of Section 4.2 modulates the desired reference position x_{sd} when the energy level E_s is close to zero. In this case there is no contact but the instability arises due to the discretization and the fast movement compared with the energy shuffling and reharvesting. Any time there is a lack of energy (positive energy in the bottom plot in the figure) the minimization problem (25) determines the proper modulated reference position $\bar{x}_{sd}$ to satisfy the stability constraint.

Conclusions

In this paper we reviewed classic and more recent algorithms that guarantee the stability of bilateral teleoperation systems. The crucial concept of passivity is implemented in each algorithm in different forms to safely interact with unknown environment despite communication delay from the master and slave sides. We list the assumptions behind each algorithm, their advantages and disadvantages, and we

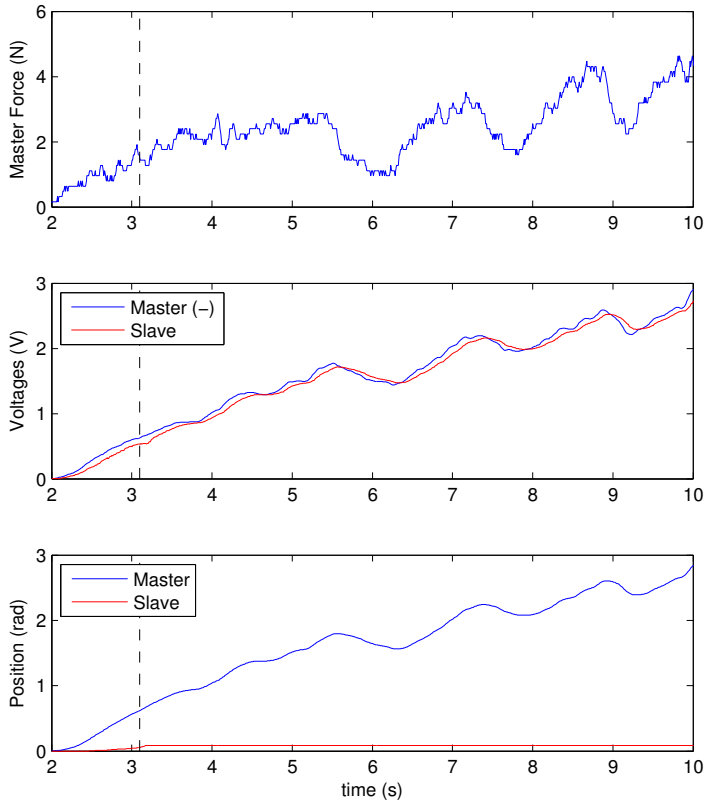


Figure 11
PD plus passivity terms. Hard Contact scenario, $\tau_{m2s} = \tau_{s2m} = 10ms$, $F_s = 100Hz$, $K_p = 1.0$, $K_v = 0.02$, $P_\varepsilon = 0.001$. The dashed vertical line around 3.1s is when the hard contact occurs.

highlight some important points that a designer should take into account to chose the best algorithm for his/her applications. Experimental results obtained by students of the Advanced Robotics course, offered at the University of Verona, are also reported to demonstrate the behavior of the different algorithms.

References

[1] ESA, http://esa-telerobotics.net/, (Access date: Dec 14, 2015).

[2] NASA, http://www.nasa.gov/, (Access date: Dec 14, 2015).

[3] Intuitive Surgical, http://www.intuitivesurgical.com, (Access date: Dec 14, 2015).

[4] T. B. Sheridan, "Space teleoperation through time delay: review and prognosis," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 5, pp. 592–606, 1993.

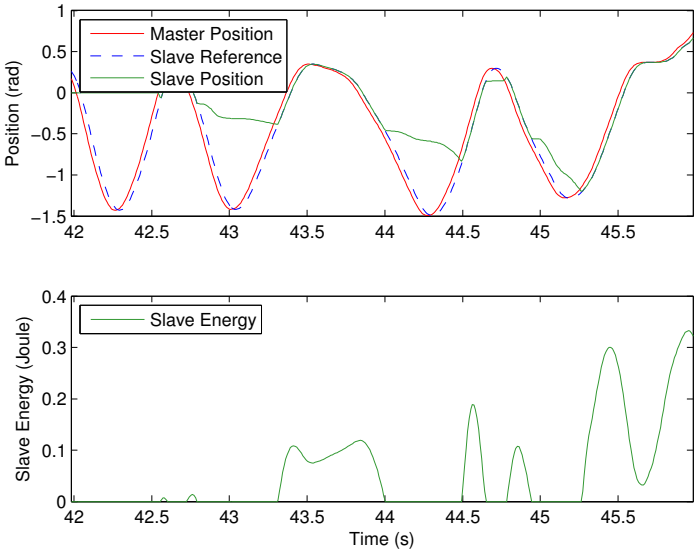


Figure 12
PSPM algorithm. $\tau_{M2S} = \tau_{S2M} = 10ms$, $F_s^{ctrl} = 500Hz$, $F_s^{network} = 100Hz$, $K_p = 1.0$, $K_v = 0.01$.

[5] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[6] E. Nuño, L. Basañez, and R. Ortega, "Passivity-based control for bilateral teleoperation: A tutorial," *Automatica*, vol. 47, no. 3, pp. 485–495, 2011.

[7] Aliaga, I., A. Rubio, and E. Sanchez, "Experimental quantitative comparison of different control architectures for master-slave teleoperation," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 1, pp. 2–11, 2014.

[8] De Gersem, G., H. Van Brussel, and F. Tendick, "Reliable and enhanced stiffness perception in soft-tissue telemanipulation," *The International Journal of Robotics Research*, vol. 24, no. 10, pp. 805–822, 2005.

[9] Malysz, P. and S. Sirouspour, "Nonlinear and filtered force/position mappings in bilateral teleoperation with application to enhanced stiffness discrimination,", *Robotics, IEEE Transactions on*, vol. 25, no. 5, pp. 1134–1149, 2009.

[10] D. Lawrence, "Stability and transparency in bilateral teleoperation," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 5, pp. 624–637, 1993.

[11] G. Niemeyer and J.-J. E. Slotine, "Stable adaptive teleoperation," *Oceanic Engineering, IEEE Journal of*, vol. 16, no. 1, pp. 152–162, 1991.

[12] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *Robotics, IEEE Transactions on*, vol. 22, no. 2, pp. 269–281, 2006.

[13] N. Chopra, M. W. Spong, and R. Lozano, "Synchronization of bilateral teleoperators with time delay," *Automatica*, vol. 44, no. 8, pp. 2142–2148, 2008.

[14] J.-H. Ryu, J. Artigas, and C. Preusche, "A passive bilateral control scheme for a teleoperator with time-varying communication delay," *Mechatronics*, vol. 20, no. 7, pp. 812–823, 2010.

[15] D. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *Robotics, IEEE Transactions on*, vol. 26, no. 2, pp. 354–369, 2010.

[16] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *Robotics, IEEE Transactions on*, vol. 27, no. 4, pp. 741–756, 2011.

[17] OROCOS, http://www.orocos.org, (Access date: Dec 14, 2015).

[18] RTnet, http://www.rtnet.org/, (Access date: Dec 14, 2015).

[19] SOEM, http://openethercatsociety.github.io/, (Access date: Dec 14, 2015).

Co-worker Robot - “PaDY”

Jun KINUGAWA†, Yusuke SUGAHARA‡, Kazuhiro KOSUGE†

†Department of Bioengineering and Robotics, Graduate School of Engineering, Tohoku University, 6-6-01 Aoba, Aramaki, Aobaku, Sendai 980-8579, JAPAN, kinugawa@irs.mech.tohoku.ac.jp, kosuge@irs.mech.toho-ku.ac.jp
‡Tokyo Institute of Technology, 2-12-1 Ookayama, Meguroku, Tokyo 152-8552, Japan, sugahara@mech.titech.ac.jp

Abstract: We present the first developed prototype of the co-worker robot “PaDY” (i.e., in-time Parts/tools Delivery to You) for automotive assembly processes and demonstrate its effectiveness in assembling vehicles. PaDY delivers the necessary part(s) and tool(s) for each task, to a worker during the assembly process, increasing efficiency, by reducing unnecessary movements by the worker; this also reduces the load of the worker and mistakes. Here we introduce the concept underlying PaDY and illustrate how PaDY can be applied to an actual assembly process based on our previous articles[3]-[6].

Keywords: Human-Robot Interaction; Partner Robot; Co-worker Robot;

1 Introduction

Industrial robots are developed to replace human workers involved in dirty, dangerous, and demanding tasks, or ”3D” tasks. Today, however, industrial robots cannot fully replace human workers in all production stages. Actually, many complex tasks are still performed by skilled human workers.

With the rapid worldwide aging of the population, replacing human workers has become a key issues for future production systems. A realistic solution is to reduce their workload and to increase efficiency.

The development of a robot system that can cooperate with human workers to reduce their load and increase efficiency and quality of work is attracting much attention in the development of "next-generation" robots, such as, YuMi, developed by ABB and Baxter, developed by Rethink Robotics.

As assembly task support system in automobile production line, Toyota Motor Corporation has developed a skill-assist system for handling a heavy instrument panel module[1] and a front and rear windows installation assist system[2]. These systems were developed for assisting a worker physically.

Figure 1
Dance Partner Robot “PBDR”

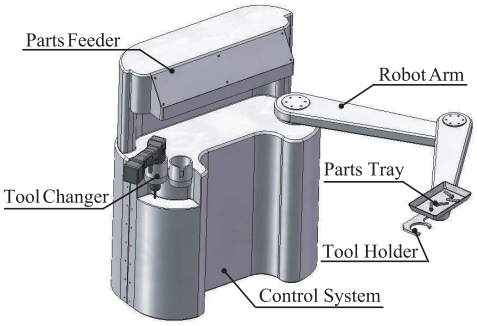


Figure 2
Concept of Co-worker Robot PaDY for Automobile Assemble Process

The co-worker robot “PaDY” was developed in our laboratory as a prototype for the assembly process in the automotive production system. In this article, we will introduce the outline of PaDY [3]-[6], which we are developing, and illustrate how they are utilized in a production system.

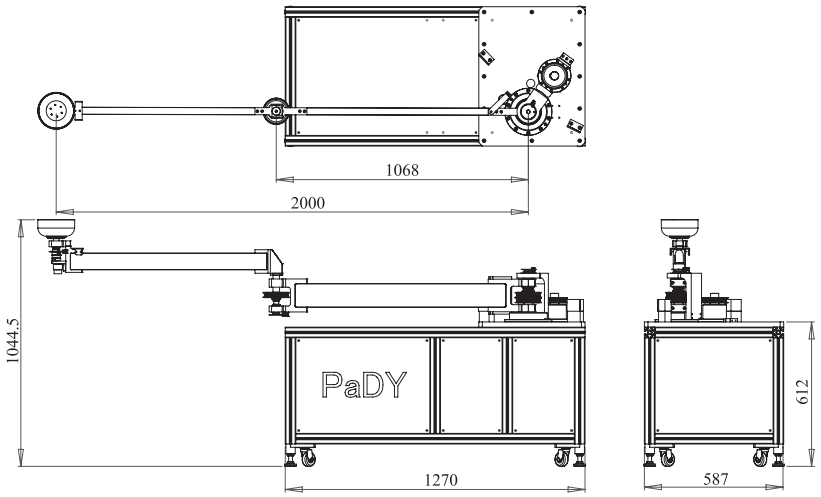
2 Co-worker Robot PaDY

2.1 Concept

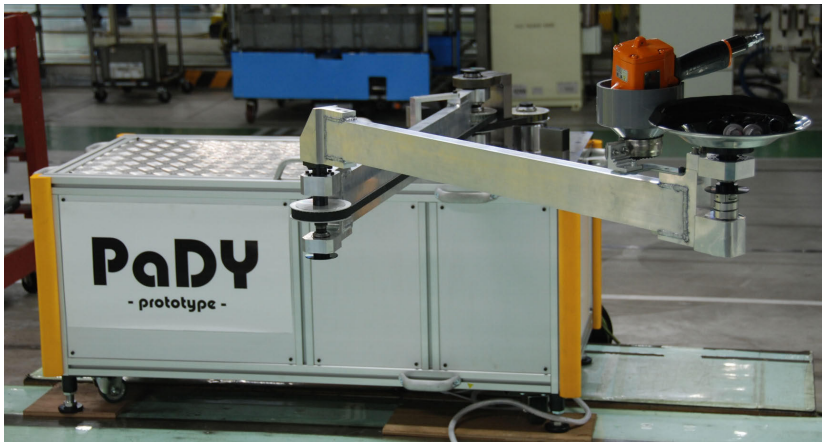
PaDY has been developed, based on the results of our previous study on the dance partner robot “PBDR” (i.e., Partner Ballroom Dance Robot) [7](Fig. 1). PBDR is a female dance partner robot, which could perform a ballroom dance along with a human male dancer. PBDR could understand its dance partner’s lead, estimate the fnext dance steps and dance in coordination with the male partner. Estimating the male dancer’s lead was the key issue in developing PBDR, along with realizing the stable physical human-robot interaction between PBDR and its partner.
When we visited an automotive assembly factory in Japan, we came across an interesting scene. An assembly process comprises a certain number of tasks, which have to be completed in a predetermined order defined by the working process schedule. The workers in certain assembly processes were noted to spend a lot of time selecting and picking up the necessary parts and tools before attaching these to the vehicle being assembled, which consumes a significant amount of time within this process.

A robot that is capable of estimating the worker’s next move or anticipating the next assembly task could probably provide the worker with the necessary part(s) and tool(s) as and when required. This is the concept of the co-worker robot ”PaDY” (i.e., in-time Parts/tools Delivery to You). PaDY is capable of delivering the necessary parts and tools to a human worker when he/she needs them for each task in the assembly process.

PaDY is expected to reduce unnecessary actions involved in assembling parts to a vehicle, such as walking to the storage (i.e., a place where parts and tools for



(a) Dimensions of PaDY P1



(b) Photo of PaDY P1

Figure 3
The Co-worker robot PaDY Prototype P1

the assembly process are stored), which is usually located close to the assembly line; selecting and picking up the necessary part(s) and tool(s) from the storage; and returning to the place where the assembly was being performed. Consequently, the assembly process can be conducted more efficiently using PaDY. In addition, possible mistakes, such as selection of incorrect part(s) and tool(s) for the next task, could be reduced as PaDY automatically selects the necessary part(s) and tool(s) according to the reference process of the assembly process.

2.2 System Configuration

PaDY consists of a manipulator system for delivering parts and tools to a human worker; a measurement system, which is used to locate the worker; part feeders, which place the necessary part(s) onto a tray attached to the end of the manipulator;

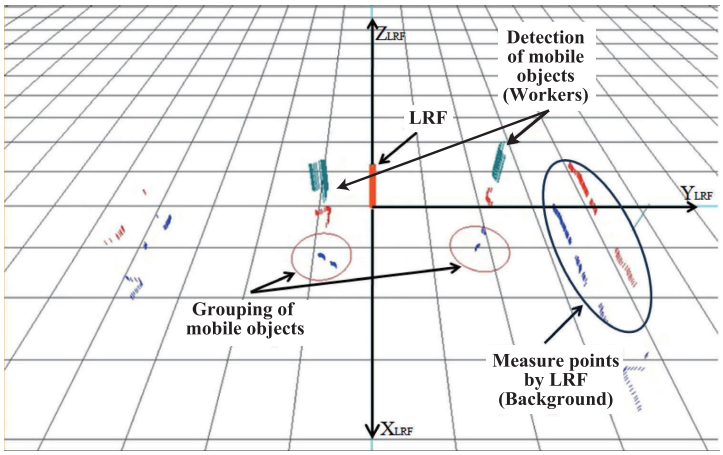


Figure 4
Worker Detection by Using Background Subtraction Technique and a Clustering Method

a tool changer for selecting the necessary tool(s); and an integrated control system.

The integrated control system includes the work progress estimation and arm motion planning systems. The work progress estimation system estimates the current state of the work process based on information from the measurement system, while referring to the working process schedule a priori. The motion planning system controls the manipulator by planning a trajectory of the arm in accordance with the work progress obtained via the work progress estimation system.

First, the worker is located using the laser range finder (LRF) system; subsequently, the integrated control system decides the appropriate delivery location and time at which the part(s) and tool(s) are required for the next task based on the history of the worker's position and the work progress model of the working process. PaDY then delivers the part(s) and tool(s) to the worker by controlling the manipulator based on the motion planned. The system architecture is shown in Fig. 2.

The co-worker robot prototype PaDY P1 is presented in Fig. 3. Note that for simplification, peripheral systems such as the motion measurement system, part feeders, and tool changers have not been included in the figures.

2.3 Manipulator Design for PaDY

PaDY coexists with the worker and operates within the worker's workspace. The manipulator was designed based on the following requirements:

- Wide range of motion: The manipulator needs to have a sufficient range of motion to cover the entire workspace.
- Lightweight: To reduce the impact force for unavoidable/possible collisions with the worker, the manipulator needs to be as lightweight as possible.
- Low power actuators: To ensure safety, low power actuators are incorporated.

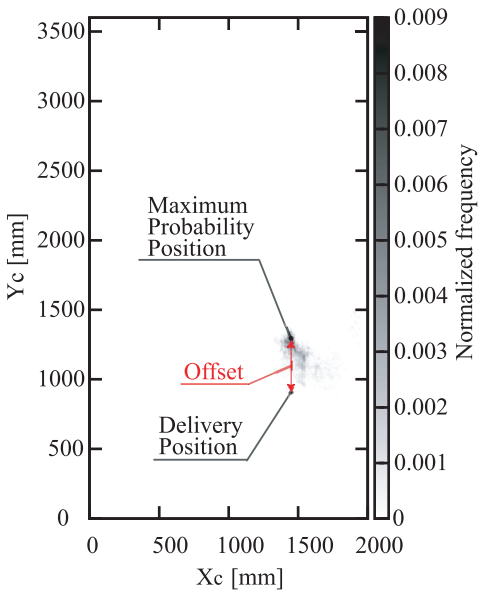


Figure 5
Delivery Position

- Load capacity: The manipulator needs to have adequate load capacity for carrying the part(s) and tool(s) required for each task of the assembly process.

We developed the Very Light Weight Wide Workspace (VLWWW) Arm to meet the requirements listed above. The VLWWW arm is driven by servomotors through mechanical torque limiters to guarantee that excessive force is not applied to its environment, including the worker in the environment.

3 Motion Planning Based on a Statistical Approach

3.1 Locating the Workers

In the prototype system, the worker is located using the LRF system. It is because the LRF system could cover wide space enough to cover the wide range workspace. Assuming that the worker is the only object that moves in the workspace being considered, the background subtraction method is used for extracting the worker from the measurement data by the LRF system. The workers in the workspace are detected by applying a clustering method to the measurement data processed by the background subtraction method[6].

In the prototype system, the nearest neighbor method[8] is used, and a set of measurement points located within a circle with a certain radius (in the following example, 500 mm) relative to a point of the set are regarded to belong to the same cluster. The number of clusters obtained by this technique is regarded as the number of the workers in the same workspace. Figure 4 shows an example of identifying and locating the workers using this method.

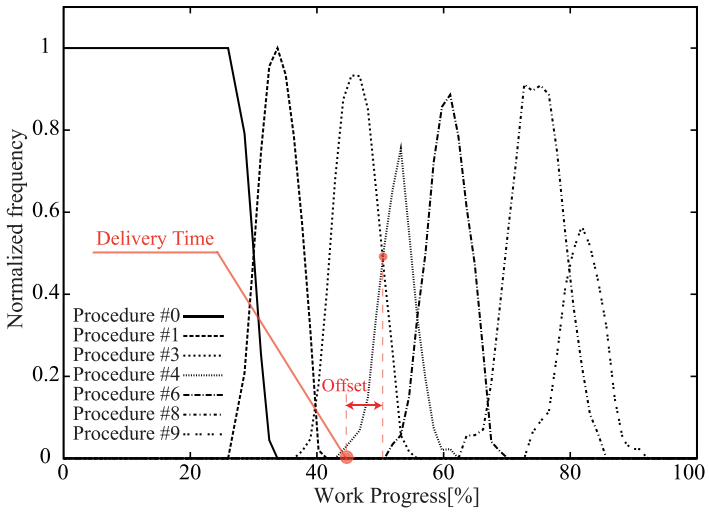


Figure 6
Delivery Time(Tasks #0,#2,#5,and #7 are not shown in this figure because these are not assembly tasks.)

3.2 Motion Planning System

Ordinal industrial robots are controlled using the teach-in and playback system; this system is quite useful for simple, repetitive operations. However, the teach-in and playback concept could not be applied to the co-worker robot because the human worker could not repeat the exact same motion in repetitive operations. To solve this problem, we developed the motion planning system, which generates robot motion based on the work progress model constructed by statistically processing the worker's motion data.

3.2.1 Statistical Analysis of the Assembly Process

To statistically analyze the assembly process and to understand how each task is performed, the worker is localized in a coordinate system attached to the body of the vehicle being assembled. The workspace described in the coordinate system attached to the body of the vehicle is divided into cells by a grid with an appropriate size.

The worker's position is periodically measured by the LRF system. Notably, the vehicle moves during any assembly process; therefore, each assembly task considered in this study was performed at a different location.

We calculate the existence probability of the worker $E_{n,i,j}$ in each cell (i,j) for n-th task of the assembly process by dividing the frequency recorded for the cell (i,j) for n-th task by the total number of data points recorded for all of the tasks involved in the assembly process for the cell (i,j). $E_{n,i,j}$ shows the probability that the worker is located in this cell (i,j) during the n-th task of the process.

The frequency $F_{n,t}$ that the worker is engaged in the n-th task at time t from the start

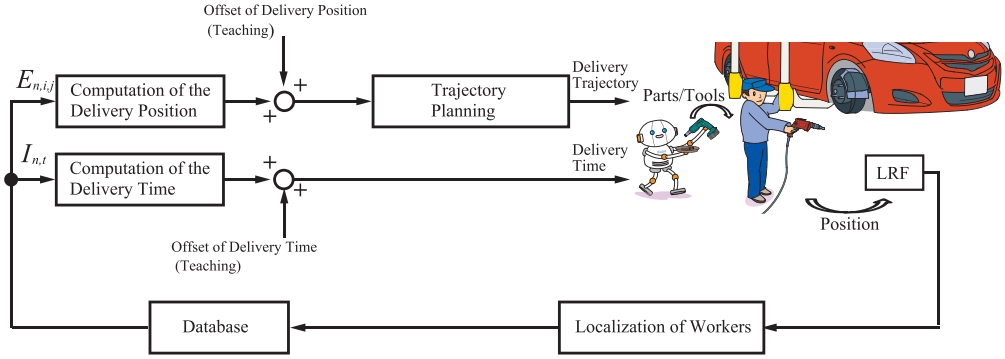


Figure 7
Control System

of each assembly process is also repeatedly recorded. The execution probability for the n-th task at time t, $I_{n,t}$, is computed by dividing $F_{n,t}$ with which the worker is engaged in the n-th task at t by the total number of the recorded repetitions of the assembly process. Therefore, we could estimate which task the worker is executing at t using $I_{n,t}$.

The frequency $G_{n,i,j}$ with which the worker is engaged in the n-th task in a particular cell (i, j) is also recorded. The execution probability of the n-th task for a given cell (i, j), $R_{n,i,j}$, is calculated by dividing $G_{n,i,j}$ by the total number of recorded frequencies for the cell (i, j) for all of the tasks and for all the repetitions of the assembly process. $R_{n,i,j}$ indicates the probability that the worker is engaged in the n-th task in the cell (i, j).

The manipulator's motion is planned based on the probabilities defined above.

3.2.2 Motion Planning

In the motion planning system, the time and location at which the part(s) and tool(s) will be delivered is determined and the manipulator's trajectory is generated. The outline of the motion planning is shown below:

1. The worker's location is estimated using the data measured by the LRF system.
2. The existence probability of the worker $E_{n,i,j}$, execution probability of n-th task at time t, $I_{n,t}$, execution probability of n-th task for a given cell (i, j), and $R_{n,i,j}$ are calculated using the recorded data for a certain number of repetitions of the assembly process.
3. The delivery position of the part(s) and tool(s) for the n-th task is determined by the existence probability $E_{n,i,j}$. That is, the cell having the highest probability is the location where the n-th task is executed, as each task in the assembly process is defined considering the location of work with respect to the vehicle body when the assembly process is segmented into tasks. The

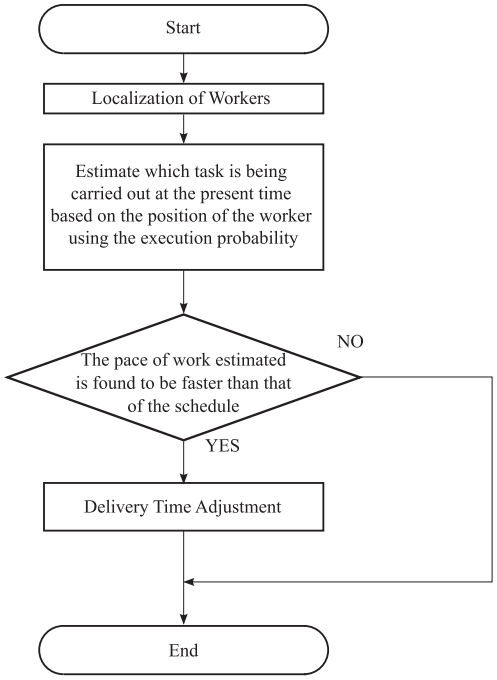


Figure 8
Delivery Time Adjustment Based on Worker's Position information

delivery position of the part(s) and tool(s) is appropriately determined for the worker based on the delivery position determined above by adding the "off-set" position to the selected location (Fig. 5).

4. The delivery time of the part(s) and tool(s) of the n-th task is determined by the execution probability $I_{n,t}$ of the n-th task at time t. That is, the delivery time is determined based on the time when the probability of the n-th task exceeds the probability of the $(n-1)$-th task. The delivery time is determined by subtracting an offset time from this time (Fig. 6).
5. The trajectory of the manipulator is generated based on the delivery position and time of the part(s) and tool(s).

Note that the aforementioned planning is conducted for each task of the assembly process being considered. Figure 7 illustrates the architecture of the control system based on the motion planning system described above.

3.2.3 Delivery Time Adjustment

In the method stated above, the start of each task is determined under the assumption that the worker is engaged in the task, which has the highest execution probability with respect to time. If some of the tasks up to the $(n-1)$-th task are performed earlier than usual, the worker needs to wait until the part(s) and tool(s) arrive at the location for the n-th task, as the start of each task is determined based on the highest execution probability of tasks with respect to time.

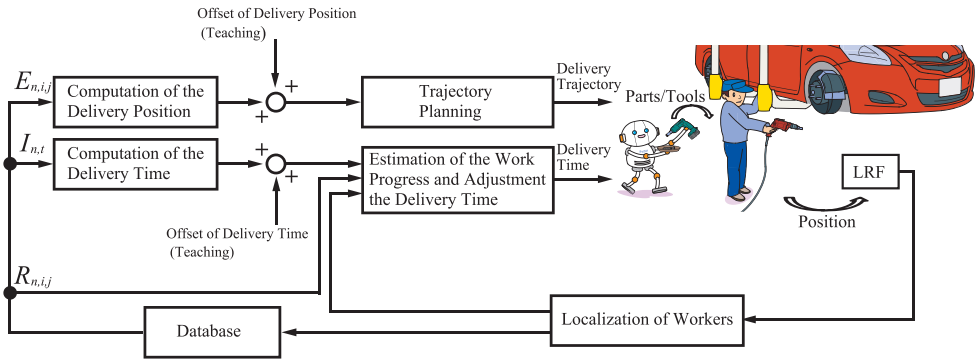


Figure 9
Control System

In the production line, an assembly process is defined by the working process schedule a priori. The working process schedule specifies when each task starts, which part(s) and tool(s) are required for each task, how long it takes to finish each task, and where each task is performed. The worker executes the assembly process according to the working process schedule.

This is mandatory in most of today's automobile production systems and the worker is not allowed to modify the schedule. Thus, it is possible to estimate which task is being performed at each time based on the position of the worker using the execution probability of the n-th task for a given cell (i,j), $R_{n,i,j}$.

Based on $R_{n,i,j}$, we can determine which task is most probably performed within each cell (i,j) [3]. That is, each cell is related to a task, which is most probably executed in that cell. Thus, the task most probably executed by the worker can be estimated from the cell, where the worker is located at each time.

Therefore, we can adjust the start of each task as and when necessary (Fig. 8). Note that only when the n-th task is completed earlier than estimated using the execution probability for the n-th task at time t, $I_{n,t}$, the start time of the $(n+1)$-th task is adjusted by making the start of the task earlier based on the difference observed in real time. The control system, which includes the adjustment of the start of each task, is shown in Fig. 9.

3.2.4 Experiments

The control system shown in Fig. 9 was implemented in PaDY P1, and experiments were performed in an actual vehicle assembly line. The robot is controlled by a PD controller. The detailed experimental results are shown in Ref. [3]. When PaDY supported the worker, the time walking to the storage and the length of the worker's path during the task have been decreased. As a result, the time required for the assembly process, which includes six assembly tasks, was reduced by $>25\%$, by introducing PaDY.

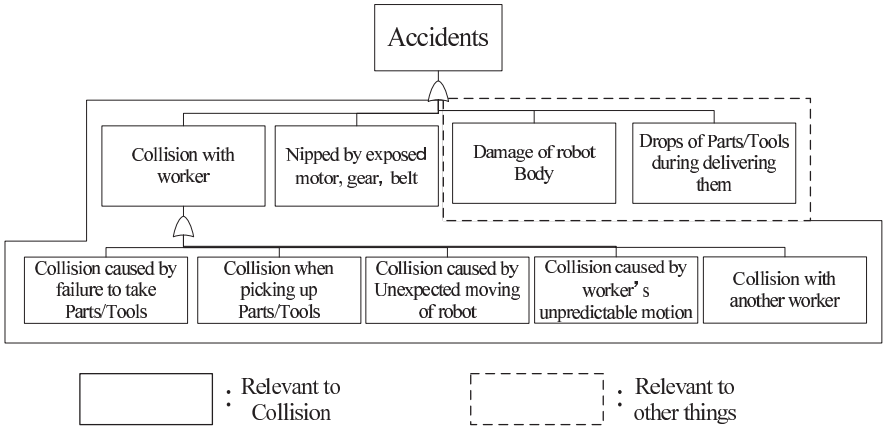


Figure 10
Fault Tree Analysis

4 Collision Risk Reduction System

PaDY autonomously operates based on the information from the assembly line. It is installed in the assembly process of the line, and it determines its behavior based on the information from the assembly line, such as when the next vehicle is coming, which types of vehicles are coming, and the worker's location detected by the LRF system.

Even if PaDY is operating without any problem, the worker engaged in the assembly process may collide with it by being in close proximity with it. The worker may not always be able to see PaDY because they are usually focused on completing the tasks within the working process schedule.

4.1 Fault Tree Analysis

Possible hazards leading to an accident in this system were analyzed using fault tree analysis (FTA). The results are shown in Fig. 10. Hazards surrounded by solid lines in Fig. 10 are related to collisions between the worker and the robot, which is the main type of hazard.

It is therefore important to implement measures to prevent collisions between PaDY and the worker, in order to reduce risks and to prevent nips by exposed mechanisms including motors, gears and belts.

4.2 Obstacle Detection System

We designed an obstacle detection system for PaDY. The system comprises multiple proximity sensors distributed on the surface of PaDY's manipulator arm (Figs. 11 and 12). The braking distance required for completely stopping PaDY depends on the attitude and speed of PaDY's manipulator at any given time. For this reason, a

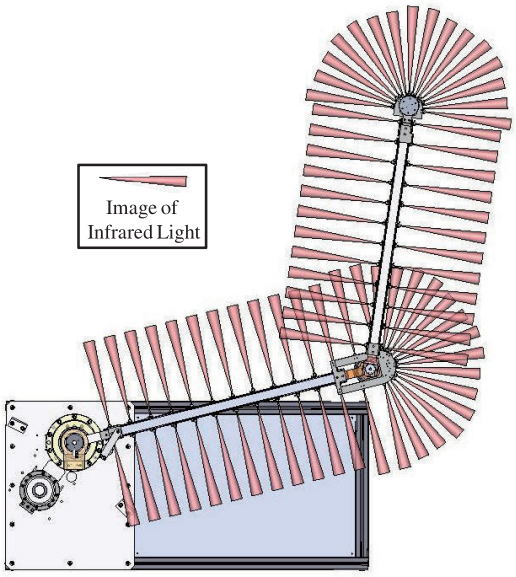
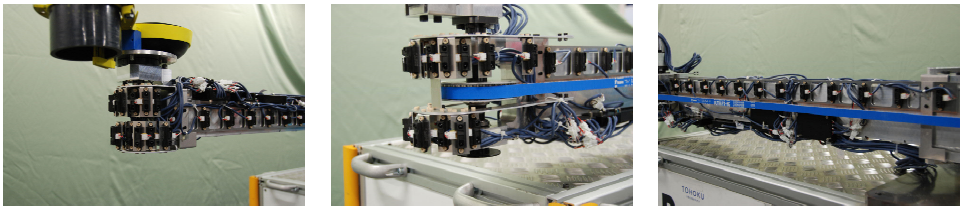


Figure 11
PSD Sensor Arrays on PaDY



(a) (b) (c)

Figure 12
Parallel Distributed Obstacle Detection System: (a) around End Effector, (b) around Joint, (c) around Links

proximity sensor with variable detection distance (GP2Y0A21YK0F, SHARP) was selected for the obstacle detection sensor.

4.3 Collision Avoidance System and Reduction of Collision Risks

The collision avoidance system performs braking control to immediately stop PaDY's manipulator when the obstacle detection system detects an obstacle. Note that each joint is driven by a motor through a mechanical torque limiter in this system.

If the manipulator suddenly stops, the mechanical torque limiters are activated by the inertial force of the manipulator, and the manipulator would be out of control. The braking control need to make the manipulator stop at the shortest possible distance without activating the torque limiters [5].

Even with the collision avoidance system, it is impossible to completely prevent

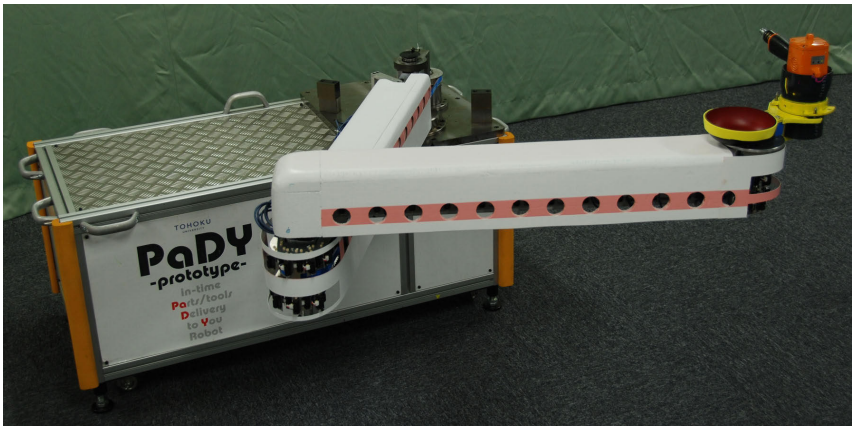


Figure 13
PaDY with Collision Risk Reduction System

collisions. When a collision occurs, it is necessary to reduce the impact and pressing forces after the collision. How much impact force is given to a human from a robot when they collide with each other depends on the magnitude of the moment of inertia. The peak of the impact force becomes large, if the manipulator arm is not covered with a buffer material. K. Suita et al., proposed a concept and a design method of covering a robot with a viscoelastic material to reduce impact force keeping within the human pain tolerance limit[9]. We adopted a similar method proposed by them.

We developed a lightweight arm for PaDY to reduce the moment of inertia and decrease the impact force and attached a buffer material on the surface of PaDY to reduce the peak of the impact force. Fig.13 shows the appearance of the robot arm with the buffer materials on the surface of PaDY. Furthermore, the mechanical torque limiters were installed between the motor outputs and joints of PaDY’s manipulator to reduce the pressing force.

PaDY was designed safely to mitigate the possible causes of risks and to meet the safety standards of the company based on Japanese Industrial Standards (JIS B 8433), where PaDY P1 is installed for the further development of PaDY.

Conclusions

In this article, we have introduced an overview of the Co-Worker Robot called “PaDY”, which has been installed in an automotive production system to improve efficiency, reducing the time required to complete the assembly process by >25%. Different types of PaDYs are under development and will be installed in other assembly processes within the assembly line. We expect that the co-worker robots will be able to assist workers and reduce their working load and thereby increase their overall productivity.

References

[1] Y. Yamada, H. Konosu, T. Morizono and Y. Umetani: Proposal of Skill-Assist: A System of Assisting Human Workers by Reflecting Their Skills in

Positioning Tasks, Proceedings IEEE International Conference on Systems, Man, and Cybernetics, IV, 1999, pp. 11-16

[2] Naoyuki Takesue, Hideyuki Murayama, Kousyun Fujiwara, Kuniyasu Matsumoto, Hitoshi Konosu, and Hideo Fujimoto: Kinesthetic Assistance for Improving Task Performance -The Case of Window Installation Assist-, International Journal of Automation Technology, Vol.3, No.6, 2009, pp.663-670

[3] Jun Kinugawa, Yuta Kawaai, Yusuke Sugahara, and Kazuhiro Kosuge: PaDY : Human-Friendly/Cooperative Working Support Robot for Production Site, Proceedings of the 2010 IEEE/RSJ International Conference on Inteligent Robots and Systems (IROS2010), Taipei 2010, pp. 5472-5479

[4] Jun Kinugawa, Yasufumi Tanaka, Yuta Kawaai, Yusuke Sugahara, and Kazuhiro Kosuge: A Path Generation Method for Collision Risk Reduction and Quantitative Evaluation of Assembly Task Partner Robot, Proceedings of the 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2011), Budapest 2011, pp. 409-415

[5] Jun Kinugawa, Yuta Kawaai, Yasufumi Tanaka, Yusuke Sugahara, and Kazuhiro Kosuge: Collision Risk Reduction System for Assembly Task Partner Robot, Proceedings of the 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2012), Kaohsiung 2012, pp. 280-285

[6] Yasufumi Tanaka, Jun Kinugawa, and Kazuhiro Kosuge: Motion Planning with Worker's Trajectory Prediction for Assembly Task Partner Robot, Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2012), Vilamoura 2012, pp. 1525-1532

[7] Takahiro Takeda, Yasuhisa Hirata, and Kazuhiro Kosuge: DanceStep Estimation Method Based on HMM for Dance Partner Robot, IEEE Transaction Industrial Electronics, vol. 54, no. 2, 2007, pp.699-706

[8] Philip J. Clark and Francis C. Evans: Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations, Ecology, Vol.35, No.4, 1954, pp.445-453

[9] K. Suita, Y. Yamada, N. Tuchida, K. Imai, H. lkeda, N. Sugimoto: Failure-to-Safety "Kyozon" System with Simple Contact Detection and Stop Capabilities for Safe Human-Autonomous Robot Coexistence, Proc. IEEE International Conference on Robotics and Automation, 1995, pp. 3089-3096

Impedance Control of Redundant Manipulators for Safe Human-Robot Collaboration

Fanny Ficuciello, Luigi Villani, Bruno Siciliano

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione,
Università di Napoli Federico II, Napoli, Italy,
{fanny.ficuciello,luigi.villani,bruno.siciliano}@unina.it

In this paper, the impedance control paradigm is used to design control algorithms for safe human-robot collaboration. In particular, the problem of controlling a redundant robot manipulator in task space, while guaranteeing a compliant behavior for the redundant degrees of freedom, is considered first. The proposed approach allows safe and dependable reaction of the robot during deliberate or accidental physical interaction with a human or the environment, thanks to null-space impedance control. Moreover, the case of control for co-manipulation is considered. In particular, the role of the kinematic redundancy and that of the impedance parameters modulation are investigated. The algorithms are verified through experiments on a 7R KUKA lightweight robot arm.

Keywords: robot manipulators; physical human-robot interaction; impedance control

1 Introduction

The need of safe and dependable robots operating in the close vicinity to humans or directly interacting with persons is growing in a wide range of application domains, ranging from domestic environments to industry. This requires a major technology shift from classical industrial robots, which are closed in cages to guarantee safety, to a new generation of robots suitable to be used in close collaboration with humans.

These robots must be designed with a high degree of compliance to reduce the interaction forces, both in the case of collision and during physical collaboration with humans. Moreover, a safe human-robot coexistence must be guaranteed combining different control strategies.

Collisions should be avoided using exteroceptive sensors, such as video cameras or depth sensors, together with fast collision avoidance control algorithms. Also, appropriate collision detection and reaction strategies must be adopted in case of collisions that cannot be avoided. The reaction strategies are aimed at immediately removing the robot from the collision area. Nevertheless, in the case of redundant robots, it is

possible to preserve as much as possible the execution of the end effector task by projecting the reaction torques into the null space of the main task.

Similarly, in the case of accidental or intentional physical interaction of a human with the robot's body, a suitable compliance control strategy can be adopted for the redundant degrees of freedom of the robot, which possibly does not interfere with the task assigned to the robot's end effector. The resulting approach, proposed in the first section of this paper, can be denoted as task space control with null space compliance.

On the other hand, a number of robotic tasks require intentional physical interaction of humans with robots. This happens, for example, in co-manipulation tasks, where the human guides the robot's end effector. For these tasks, suitable control strategies ensuring the robot's compliance at the end effector must be adopted and kinematic redundancy can be exploited to enhance intuitiveness and stability of the physical human-robot interaction. This problem is addressed in the second section of the paper.

The paradigm common to the two approaches is the use of impedance control [1] for human-robot physical interaction. They have been considered separately in the references [2], for what concerns null-space impedance control, and [3] for variable impedance control at the end effector.

2 Null-Space Impedance Control for Human-Robot Coexistence

When a robot works close to humans, interaction may occurs with the robot's body. Safety can be guaranteed by ensuring a compliant behaviour to the robot's joint. In the presence of kinematic redundancy, the compliant behaviour can be projected in the null space of the main task, so that the latter can be correctly executed. In other words, the control goal is to minimize the error of the main task and at the same time, to ensure safe interaction through active compliance in the null space of the main task.

Active compliance can be obtained using impedance control, which has been extensively studied in the literature. The compliant behavior usually is realized in the task space to control the interaction of the end effector. However, an impedance behavior can be imposed also in the null space of the main task to ensure safety.

To this purpose, two control schemes which do not require direct joint torque measurements are presented. The first scheme is based on a disturbance observer which estimates the external forces acting on the task variables on the basis of the task space error. The second scheme relies on the momentum-based observer [4]. In both cases, the overall stability of the system, with asymptotic convergence of the main task and a desired impedance behavior in the null space of the main task, can be proven. More details and the proofs are in [2].

2.1 Null-Space Impedance Control

The dynamic model of a n-link robot manipulator can be expressed by

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)+\tau_{ext}=\tau, \tag{1}$$

with standard notation. In this model, τ is the input vector torque while τ_{ext} is the torque resulting from external interaction. The well-known model-based resolved acceleration control can be adopted to compute the driving torques

$$\tau=M(q)\ddot{q}_c+C(q,\dot{q})\dot{q}+g(q), \tag{2}$$

where $\ddot{q}_c$ is the joint command acceleration to be suitably designed.

For a redundant manipulator, redundancy lets us to have some kind of joint impedance and task space control simultaneously. The so-called null-space impedance can be realized in the null space of the main task to control the interaction on the robot's body. The corresponding command joint acceleration in (2) is given by

$$\ddot{q}_c=J^{\dagger}(\ddot{x}_c-\dot{J}\dot{q})+N(\ddot{q}_d+M_d^{-1}(B_d\dot{\tilde{q}}+K_d\tilde{q})), \tag{3}$$

which produces the task space and null-space closed-loop behavior respectively as follows

$$\begin{aligned}&\ddot{x}_c-\ddot{x}=JM^{-1}\tau_{ext},\\&N(\ddot{\tilde{q}}+M_d^{-1}(B_d\dot{\tilde{q}}+K_d\tilde{q})-M^{-1}\tau_{ext})=0.\end{aligned} \tag{4}$$

Here $\ddot{x}_c$ is the command acceleration in the task space, $\tilde{q}=q_d-q$ where q_d is the desired trajectory or a rest configuration in the joint space, $J^{\dagger}$ is any (weighted) right pseudo-inverse of the task space Jacobian matrix J, $N=(I-J^{\dagger}J)$ is the null-space projection matrix, and M_d, B_d and K_d are the impedance matrices. This choice of $\ddot{q}_c$ allows the joint space impedance in the null-space of the main task to be realized, provided that the desired inertia matrix is chosen as $M_d=M(q)$. On the other hand, the main task experiences errors as a result of the external torques that are applied on the robot's body. It can be easily shown that an arbitrary M_d can be set only if the measurement of τ_{ext} is available to the controller; in this case, also the influence of τ_{ext} on the main task can be cancelled out.

2.2 Task-Based Observer

The following theorem is given for correct execution of the main task, while ensuring a compliant behaviour of the robot's body.

Theorem 1. Let us denote with $\hat{\tau}$ the estimated external torque and with $\tilde{\tau}=\tau_{ext}-\hat{\tau}$ the estimation error. Also, define the error $s=\dot{\tilde{x}}+P\tilde{x}$, where P is a positive definite diagonal matrix and $\tilde{x}=x_d-x$. Then, for selected constant diagonal positive definite matrix K and constant positive definite matrix Γ_f, the control law

$$\begin{aligned}\tau=&J^T\Lambda(\ddot{x}_d+P\dot{\tilde{x}}-\dot{J}\dot{q})+J^T\left(\frac{1}{2}\dot{\Lambda}+K\right)-J^TJ^{\#T}\hat{\tau}\\&+MN_{\#}(\ddot{q}_d+M^{-1}(B_d\dot{\tilde{q}}+K_d\tilde{q}))+C(q,\dot{q})\dot{q}+g(q),\end{aligned} \tag{5}$$

with the disturbance observer

$$\dot{\hat{\tau}} = -\Gamma_f^T J^{\#} s, \tag{6}$$

guarantee that the $\tilde{x}$ and $\dot{\tilde{x}}$ go to zero asymptotically while the null-space impedance behavior is imposed. Moreover, the estimated disturbance remains bounded and the closed-loop system is stable.

In (5), $\Lambda = (JM^{-1}J^T)^{-1}$ is task inertia matrix, $J^{\#} = M^{-1}J^T\Lambda$ is the dynamically consistent generalized inverse [5] and $N_{\#} = (I - J^{\#}J)$.

It can be easily shown that the closed-loop dynamics for the null-space is

$$N_{\#}(\ddot{\tilde{q}} + M^{-1}(B_d\dot{\tilde{q}} + K_d\tilde{q}) - M^{-1}\tau_{ext}) = 0, \tag{7}$$

corresponding to an impedance behavior in the null-space of the main task.

2.3 Momentum-Based Observer

Another method to ensure the correct execution of the main task during interaction is based on the collision detection algorithm presented by [4]. The basic concept is the computation of the n-dimensional residual vector

$$r(t) = K_I\left(p(t) - \int_0^t (\tau + C^T(q,\dot{q})\dot{q} - g(q) + r(\sigma))d\sigma\right), \tag{8}$$

where $p(t) = M(q)\dot{q}$ is the robot generalized momentum and K_I is a positive definite diagonal matrix. This vector can be computed using the measured signals q and $\dot{q}$, and the commanded torque τ, with initial conditions $r(0) = 0$ and $p(0) = 0$. The dynamics of r is

$$\dot{r} = -K_I r - K_I\tau_{ext}, \tag{9}$$

corresponding to a filtered version of the real external torques, i.e. $r(t) = \tau_{ext}$. In the absence of interaction, assuming no noise and unmodeled disturbances, $r(t) = 0$. As soon as interaction occurs, the components of r will raise exponentially and will reach to the value of $-\tau_{ext}$.

A control algorithm similar to the one that was given by (5), using r in place of $\hat{\tau}$, is adopted

$$\begin{aligned}\tau = {} & J^T\Lambda(\ddot{x}_d + K_D\dot{\tilde{x}} + K_p\tilde{x} - \dot{J}\dot{q}) - J^T J^{\#T} r \\ & + MN_{\#}[\ddot{q}_d + M_d^{-1}(B_d\dot{\tilde{q}} + K_d\tilde{q})] + C(q,\dot{q})\dot{q} + g(q),\end{aligned} \tag{10}$$

By defining the estimation error $\tilde{r} = r + \tau_{ext}$, the closed-loop task dynamics is

$$\begin{aligned}\ddot{\tilde{x}} + K_D\dot{\tilde{x}} + K_p\tilde{x} &= JM^{-1}\tilde{r} \\ \dot{\tilde{r}} &= -K_I\tilde{r} + \dot{\tau}_{ext}. \qquad ,\end{aligned} \tag{11}$$

while the closed-loop dynamics for the null-space is the same as in (7). From the stability properties of the cascade systems it can be shown that when $\|\dot{\tau}_{ext}\|$ is bounded, $\tilde{x}$ is also bounded and specifically when $\dot{\tau}_{ext} = 0$ the system is asymptotically stable and $\tilde{x}, \dot{\tilde{x}} \to 0$.

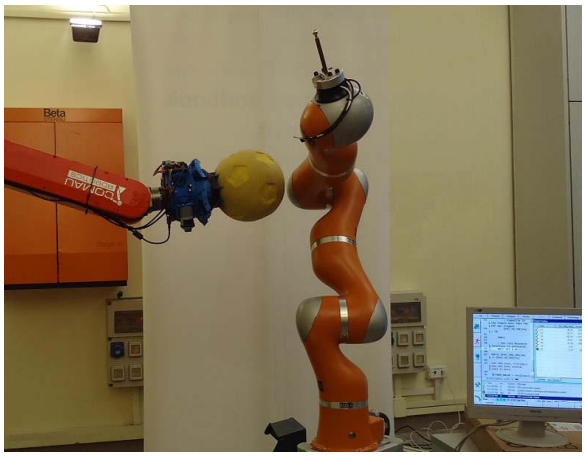


Figure 1
Elastic ball in interaction with a KUKA LWR4 arm

2.4 Experimental Results

The proposed approaches are verified experimentally on a 7 DOF KUKA LWR4 lightweight arm ($n = 7$). Control algorithms are executed through Fast Research Interface (FRI) library with a sampling rate of 2 ms.

The experiments are performed for three cases: without observer, with task error based observer and with momentum based observer. In all the cases the position of the end effector is assumed as the main task ($m = 3$). Therefore, the robot has 4 degrees of redundancy ($r = 4$).

A constant configuration $q_d = [\pi/4, -\pi/6, 0, -\pi/1.8, \pi/6, -\pi/4, 0]$ is considered, corresponding to the constant desired position of the end effector in the task space $x_d = [-0.242, -0.133, 0.968]^T$.

The interaction occurs with an elastic ball of 1200 N/m approximate stiffness at a point of the robot arm close to the fourth joint. While the end effector is commanded to be in the desired position, the sphere comes in contact with the robot, stops for 10 s and finally goes back far from the robot. In order to have the same scenario in all the experiments and guarantee repeatability, the ball is moved by a position controlled industrial robot with constant speed of 4.5 cm/s along a straight line. A snapshot of the experimental setup is depicted in Fig. 1.

Case I, Interaction control without external interaction observer: The command acceleration given by (3) was considered with the gains $K_p = 2000I$, $K_v = 90I$, $M_d = M(q)$, $B_d = 0.4I$, and $K_d = 8I$.

The corresponding main task error and the estimated external torques, obtained by the torque sensors available on LWR4 robot, together with the joints position, are shown in Fig. 2. The time interval when interaction occurs is identified by the two vertical lines. It can be observed that the task space error components are zero initially but, after the collision with the sphere, they increase and reach constant

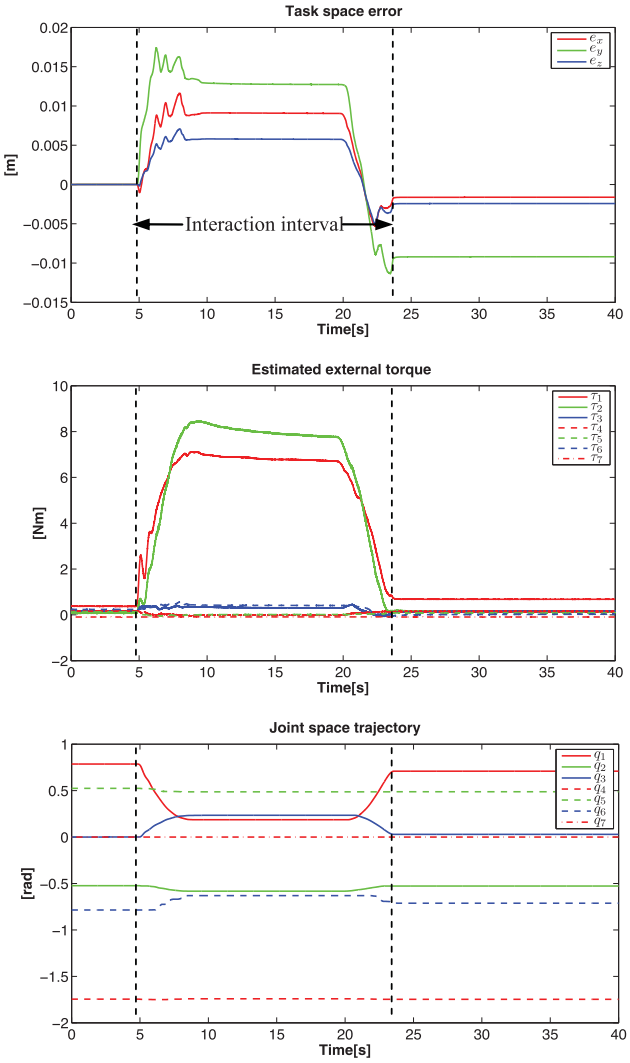


Figure 2
Task errors, external torques estimation evaluated from torque sensors and joint space trajectory, without observer

values when the sphere stops. When the sphere is retreated and contact is lost, the task error components become small but non-null, due to the presence of non negligible joint friction.

From the time histories of the joint variables in Fig. 2 it can be argued that, during the interaction, the configuration of the robot changes and the redundancy allows the manipulator to comply with the external forces. As soon as the contact is lost, the robot comes back to its desired configuration. The behavior of the arm in the null space can be set by properly choosing the control gains.

Case II, Interaction control with task error based disturbance observer: The previous experiment is repeated by using the control law (5) with disturbance observer

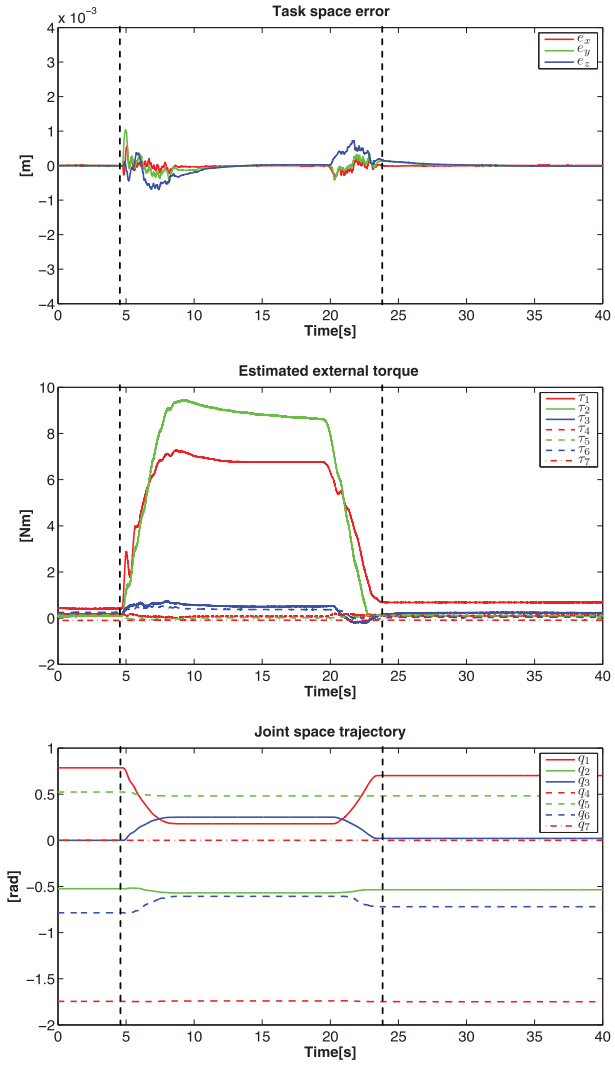


Figure 3
Task errors, external torques estimation evaluated from torque sensors and joint space trajectory, using task error based observer

(6). The parameters of the controller are tuned as $P = 25I, K = 80I$, and $\Gamma_f = 0.125I$ and the impedance matrices are selected as in the previous case.

The performance of the controller is shown in Fig. 3. Even though the external torque τ_{ext} is not constant during the first and the third phase of the interaction, namely when the sphere is approached and retreated, the controller performs very well and the task errors are more than three times lower than the previous case and the resulting interaction torques remain bounded. In the second phase of the interaction, when the sphere is at rest and a constant torque is applied, the task error converges to zero.

Comparing the plots of the time histories of the external torques and of the joint

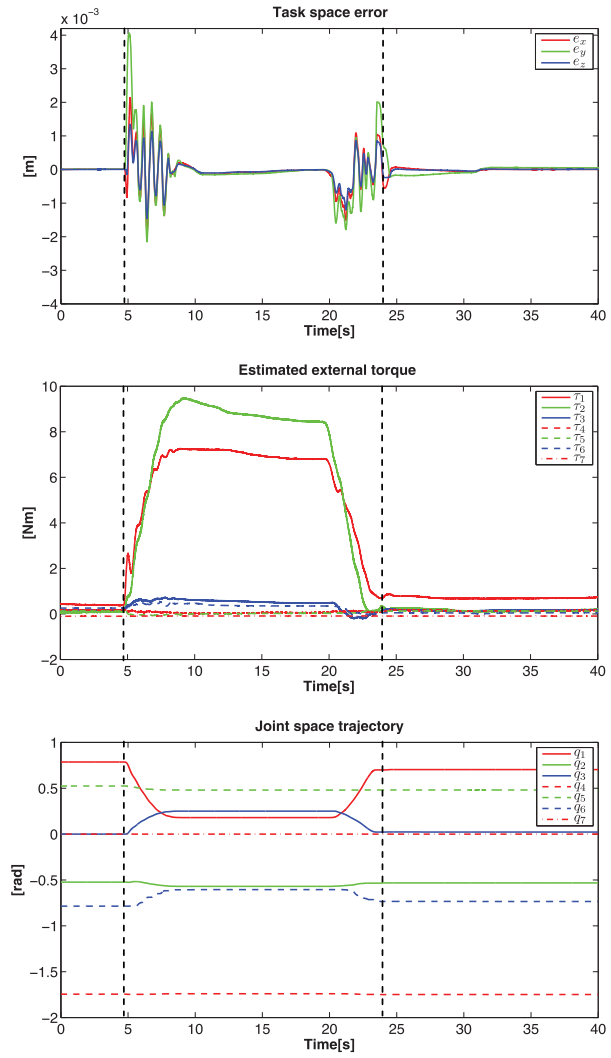


Figure 4
Task errors, external torques estimation evaluated from torque sensors and joint space trajectory, using momentum based observer

positions in Fig. 2 and Fig. 3, it can be inferred that the behavior of the robot in the null space does not change appreciably. Moreover, comparing the task space errors of Fig. 3 with those reported in Fig. 2, it can be observed that the control scheme with task error based observer not only reduces the error during the interaction, but allows also reducing the effects of friction, so that the task space error goes to zero when the contact is lost.

Case III, Interaction control with momentum based observer: The experiment is repeated using the momentum based observer (8) with $K_I = 8I$ and the control torque (10). The control gains and the impedance parameters are set as in Case I. The results are depicted in Fig. 4. It can be seen that the controller works very well during the constant phase of the interaction. However, when the external torque is

is not constant, during the first and the third phase of the interaction, the task space error shows high frequency oscillations.

3 Human-Robot Co-Manipulation

Robot manual guidance and co-manipulation tasks are of great interest in both the service and industrial fields where physical human-robot interaction constitutes an added value for the realization and simplification of many applications, such as teaching by demonstration and robot-aided manufacturing. It is widely known in literature that impedance/admittance strategies are the most effective and natural way to handle physical interaction in the face of the unpredictability of human behaviours. Indeed, impedance control is an essential paradigm to ensure reliability and safety.

To enhance intuitiveness and stability in physical human-robot interaction during co-manipulation tasks both redundancy and impedance parameters modulation play an important rule. Through an extensive experimental study on a 7-DOF KUKA LWR4 arm, we show that variable impedance is more performant with respect to constant impedance and that redundancy resolution influences not only stability but also performance. For this purpose, an impedance strategy to control a redundant manipulator is defined in the Cartesian space and different modulation laws for the impedance parameters are tested in combination with different strategies to solve redundancy.

In order to make the end effector able to follow and adapt to the force exerted by the operator at the tip, the end effector dynamics can be set as a mass-damper system of equation

$$\Lambda_d \ddot{x} + D_d \dot{x} = F_{ext}, \tag{12}$$

where Λ_d and D_d are suitable inertia and damping matrices, that are positive definite and are usually set as constant diagonal matrices.

In detail, a suitably selected variable impedance strategy [7] has been compared to constant impedance. Among possible redundancy resolution criteria, two secondary task functions inspired to the dynamic conditioning index (DCI) [13], and to the kinematic manipulability index were selected and compared [6]. For the evaluation of the results, a writing task on a horizontal plane operated by a human has been selected as a case study. In order to have significant redundant degrees of freedom that can be used for the secondary task, the orientation was not considered.

A snapshot of the co-manipulation task is reported in Fig. 5, where the operator guides a paint marker mounted on the robot's tip along a path drawn on a paper sheet.

Fig. 6 summarizes the results which will be further discussed and detailed below. The performance level increases from left to right. It can be seen that the best solution is achieved using variable damping and secondary task functions inspired to the dynamic conditioning index.

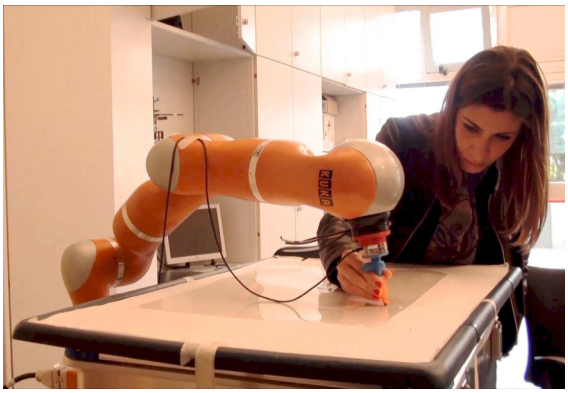


Figure 5
Snapshot of the co-manipulation task.

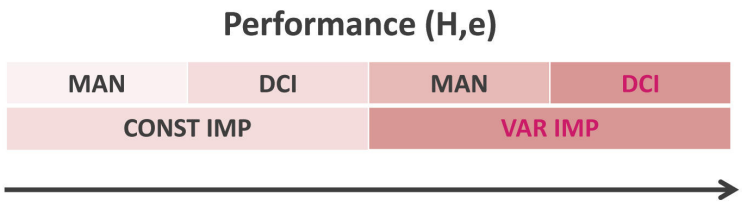


Figure 6
The results obtained with different combination of redundancy resolution and impedance strategies presented on the basis of the performance level in increasing order from left to right.

3.1 Stability Issues

Studies available in literature demonstrate that the stability of human-robot interaction, depends on the coupled dynamics of both interacting systems [12], which cannot be accurately modelled and evaluated as it involves also the estimation of human arm impedance, it depends also on the hardware namely, the robot kinematics and dynamics, the kind of transmission, the presence of friction and of structural compliance, and the kind of sensors and actuators [10, 9, 11]. Thus, in this work an experimental procedure has been set up to find the allowed range of variation of the impedance parameters where stability is preserved. In this section, only the results of the experimental procedure are reported in Fig. 7, while the whole procedure can be found in [3, 7]. The same damping and the same mass has been set along all the directions of the Cartesian space, i.e., $D_d = DI$ and $\Lambda_d = \Lambda I$, with $\Lambda = \alpha\bar{\Lambda}$, being $\bar{\Lambda} = 4.2456$kg the maximum eigenvalue of the end effector inertia in a chosen initial configuration, and $0 < \alpha \leq 1$ a scaling factor. The stability region for the parameters D and α is that included between the continuous and the dotted line.

The stability region has been evaluated experimentally by setting a value of damping D in the interval $[5, 60]$ Ns/m and reducing the value of α, starting from $\alpha = 1$, until vibrations can be felt by an operator shaking the end effector in a neighborhood

of an initial configuration. This configuration is chosen in such a way to have the effective Cartesian inertia diagonal with asymmetric distribution, i.e. one of the eigenvalues of the inertia matrix assumes a value $\bar{\Lambda}$ much bigger with respect to the others. Hence, the worst-case configuration for scaling (reducing) the end effector inertia is considered.

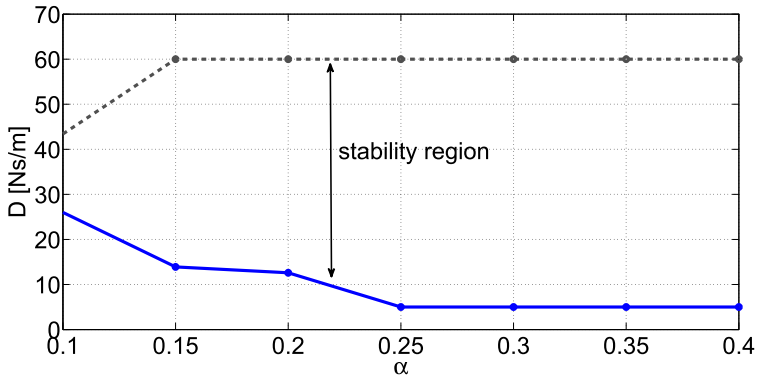


Figure 7
Range of minimum and maximum allowed damping D for a given scaling factor α of the inertia matrix.

3.2 Redundancy Resolution

The experimental studies on a redundant robot demonstrate that redundancy has an important role on both stability and performance during human-robot physical interaction. Different strategies to solve redundancy have been applied and the results compared in terms of performances and in terms of stability. The maximization of the kinematic manipulability index [6] and the minimization of the DC index [13] are the two secondary tasks that have been tested. In Fig. 8 the time histories of DC index are reported in the two cases. In the top figure, that is the case where in the impedance equation low virtual inertia has been settled at the boundary of stability region, the task can be completed only if redundancy is used to decouple the effective Cartesian inertia at the end effector of the robot. Otherwise, when manipulability index is exploited, the task is interrupted because instability occurs. In the bottom figure the virtual inertia is settled high thus the stability is always preserved. This happened since instability is likely to occur during interaction when the controller attempts to impose to the robot impedance dynamics that differs signicantly from the intrinsic hardware dynamics.

About performances related to redundancy resolution, the methods have been compared using two different impedance laws, one with constant parameters (set as $\Lambda = 1.1\,\text{kg}$, $D = 60\,\text{Ns/m}$) and one with variable damping (low constant mass, $\Lambda = 1.1\,\text{kg}$). Since the assigned task consists in pursuing a given path, a significant measure of performance is the error between the reference and the actual path, that can be defined in different ways. A very simple measure is the absolute value of the difference between the length of the path drawn in cooperation with the robot,

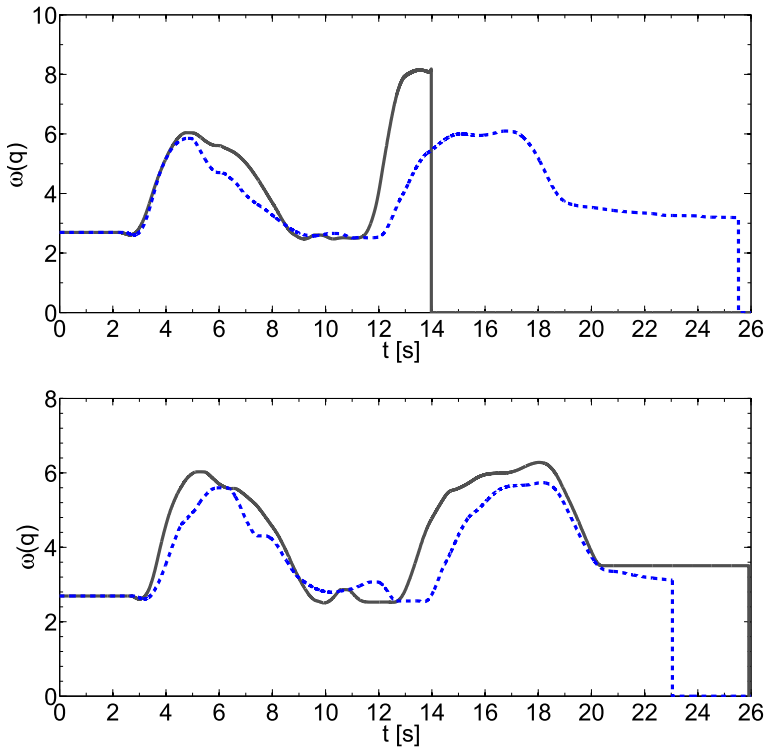


Figure 8
Time histories of the values of DC index in the case of low (top) and high (bottom) virtual inertia. The continuous lines represent the DC index when redundancy is used to increase manipulability. The dashed lines represent the DC index when redundancy is used to minimize the DC index.

l_e, and the ideal path length, l_d, namely the length error:

$$e = |l_d - l_e|. \tag{13}$$

Another performance parameter is the execution time H of the trajectory, defined as the difference between the time when the entire path is completed and the time when the drawing tool touches the paper on the desk to start writing. In order to obtain quantities that overcome the skills of the singular operator, the above parameters are evaluated as the average on the performance of more subjects.

The results of the tests are reported in Fig. 9, where the error on the length of path e versus execution time H is reported for all the subjects, as well as their mean values.

It can be observed that, for the impedance control with constant parameters, the use of DC index (DCI) ensures better performance than the use of manipulability index (Man) both in terms of execution time and error on the path. This is true also for variable impedance control even though the use of variable parameters reduces the error on the path in spite of the strategy used to solve the redundancy.

Last but not least, all the subjects involved in the experiments have confirmed that the "feeling" of the manual guidance (in terms of intuitiveness and response of the

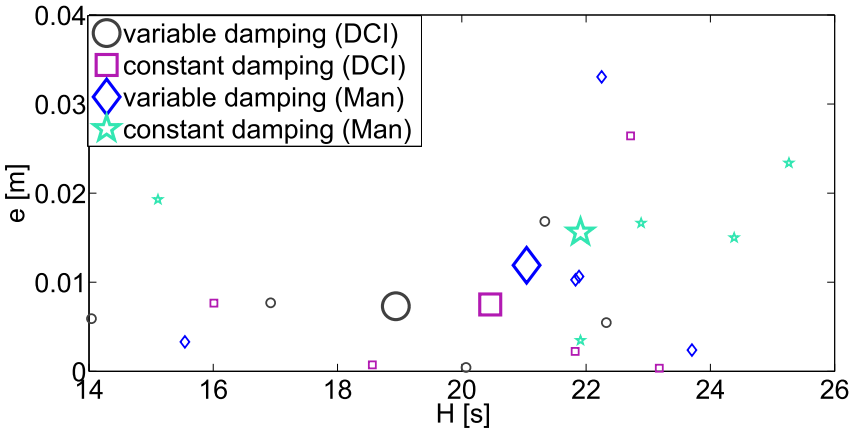


Figure 9
Values of the length error e and execution time H in the experiments on five subjects using variable and low constant impedance; both manipulability index and DCI optimisation are used as secondary tasks. The bigger markers are the mean values on the five different subjects.

robot) improves when the DC index is adopted, i.e., when redundancy is used to decouple the natural end effector dynamics along the principal directions of the task.

3.3 Impedance Modulation

In order to accommodate the human movement during physical interaction, high impedance parameters are desired when the operator performs fine movements at low velocity while lower values of the parameters should be used for large movements at high velocity [9, 10, 8]. Thus, using a variable impedance strategy it is possible to vary the damping and mass properties of the robot on-line according to human unpredictable behaviour. A crucial point is the interpretation of the human intention. Taking into account that human perception is mainly influenced by the damping parameter, while, for a given damping, the desired (virtual) mass is crucial for stability, different strategies to vary the impedance parameters have been tested and compared in [7].

Comparing the performance in terms of execution time and accuracy we finally choose to vary the damping according to the absolute value of the end effector Cartesian velocity and to set the mass as low as possible. Namely, when the velocity is high, the damping force is reduced, so that the operator can move the end effector with minimum effort and the execution time can be reduced; vice versa, at low velocity, the damping force is increased to improve accuracy. On the other hand, the virtual mass is set so as the parameters of the system remain in the stability region.

The relationships used to vary the damping for each of the Cartesian principal directions is

$$D(\dot{x}) = \min\{a e^{-b|\dot{x}|}, 5\}, \tag{14}$$

with $a = 60$ and $b = 4$. These parameters have been chosen in order to have a variation of the damping within the interval $[5, 60]$ Ns/m for the possible range of velocities in the considered task. A saturation to the minimum value of 5 Ns/m is introduced in case of high velocity.

The variable impedance control has been compared with two different sets of constant impedance gains (chosen along the curve), namely: high damping ($\Lambda = 1.1$ kg, $D = 60$ Ns/m) and low damping ($\Lambda = 1.1$ kg, $D = 20$ Ns/m).

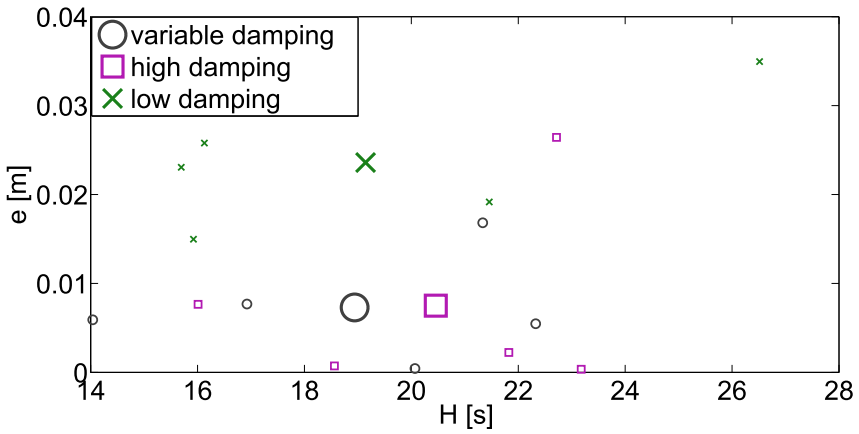


Figure 10
Values of length error e and execution time H in the experiments on five subjects using DCI optimisation, with the variable impedance control L1 and two different sets of constant parameters. The bigger markers are the mean values on the five different subjects.

The results, carried out on five different subjects, are shown in Fig. 10, where the execution time H and the error on the length of the path are reported.

The constant impedance with high damping ensures higher accuracy with respect to the constant impedance with low damping, as expected. This result, however, comes at the expenses of the execution time and of the operator effort requested for the manual guidance.

Indeed, from Fig. 11 it can be verified that higher damping requires higher forces to be exerted to the end effector. On the contrary, impedance with low damping allows the task to be performed more easily, with less effort and time, but with less accuracy.

In conclusion, the variable impedance guarantees the best compromise between accuracy, execution time and effort of the operator (Fig. 10 and Fig. 11).

Conclusions

The use of impedance control both on Cartesian or task variables and in the null space of the main task was considered in this work. The redundancy of the system was utilised to ensure safe and dependable physical interaction, as well as to enhance intuitiveness and stability of the physical human-robot interaction. In the case of co-manipulation tasks, the benefit of impedance parameters modulation was

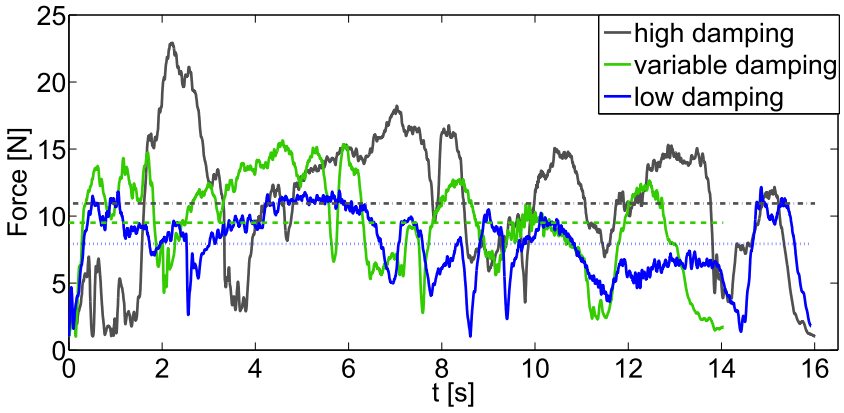


Figure 11
Norm and mean value of the contact forces for high, variable and low damping, for one subject.

investigated. The performance of the proposed algorithms has been experimentally tested on a torque controlled KUKA LWR4 robot.

Acknowledgement

This research has been partially funded by the EC Seventh Framework Programme (FP7) within SAPHARI project 287513 and RoDyMan project 320992.

References

[1] N. Hogan, "Impedance control: An approach to manipulation: Part i theory; Part ii implementation; Part iii applications," *J. Dyn. Sys., Meas., Control*, vol. 107, no. 12, pp. 1–24, 1985.

[2] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-Space Control of Robot Manipulators with Null-Space Compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2014.

[3] F. Ficuciello, L. Villani, and B. Siciliano, "Variable Impedance Control of Redundant Manipulators for Intuitive Human-Robot Physical Interaction," *IEEE Transactions on Robotics*, vol. 31, no. 12, pp.850–863, 2015.

[4] A. D. Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight robot arm," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 1623–1630.

[5] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 1115–1120, 1987.

[6] F. Ficuciello, H. Sadeghian, L. Villani, and M. Keshmiri, "Global Impedance Control of Dual-Arm Manipulation for safe Human-Robot Interaction," *10th IFAC Symposium on Robot Control*, Dubrovnik, Croatia, 2012, pp. 767–773.

[7] F. Ficuciello, A. Romano, L. Villani, and B. Siciliano, "Cartesian impedance control of redundant manipulators for human-robot co-manipulation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Chicago, USA, 2014, pp. 2120–2125.

[8] R. Ikeura, T. Moriguchi, and K. Mizutani, "Optimal variable impedance control for a robot and its application to lifting an object with a human," in *IEEE Int. Workshop on Robot and Human Interactive Communication*, Berlin, Germany, 2002, pp. 500–505.

[9] A. Lecours, B. Mayer-St-Onge, and C. Gosselin, "Variable admittance control of a four-degree-of-freedom intelligent assist device," in *IEEE Int. Conf. on Robotics and Automation*, Saint Paul, Minnesota, USA, 2012, pp. 3903–3908.

[10] V. Duchaine, B. Mayer-St-Onge, D. Gao, and C. Gosselin, "Stable and intuitive control of an intelligent assist device," *IEEE Transactions on Haptics*, vol. 5, no. 2, pp. 1939–1412, 2012.

[11] T. Insperger, L.L. Kovacs, P. Galambos, and G. Stepan, "Increasing the accuracy of digital force control process using the act-and-wait concept," *IEEE Transactions on Mechatronics*, vol. 15, no. 2, pp. 291–298, 2010.

[12] S. Buerger and N. Hogan, "Complementary stability and loop shaping for improved human-robot interaction," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 232–244, 2007.

[13] O. Ma and J. Angeles, "The concept of dynamic isotropy and its applications to inverse kinematics and trajectory planning," in *IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, 1990, pp. 10–15.

A Brief History of DLR's Space Telerobotics and Force-Feedback Teleoperation

Jordi Artigas, Gerd Hirzinger

German Aerospace Center
Institute of Robotics and Mechatronics
jordi.artigas@dlr.de, gerd.hirzinger@dlr.de

Abstract: The DLR's Institute of Robotics and Mechatronics enjoys a unique history in space telerobotics that begins in 1993 with ROTEX, the first space telerobotics mission in history [1], and has been continuing to pave the way towards space telerobotics, on-orbit servicing and planetary exploration until the present. This paper reviews DLR's major telerobotics break-throughs during the last ten years, describing in particular requirements for space telerobotics, main mission challenges and robot control methods to allow one of the oldest yet still cutting-edge DLR's robotics vision: To extend the human arm into space, that is, space telepresence.
Our work has been massively inspired by the pioneering work of Tony Bejczy and his co-workers.

Keywords: Force-feedback; Teleoperation; Space Robotics; On-Orbit Servicing

1 Space Telerobotics

Telerobotics is one of the most succesful and versatile space technologies. In the last years there have been impressive space missions that involved the use of robots, showing their effectiveness in fields as diverse as Mars exploration, on-orbit servicing (OOS) or meteorite sample and return. The ROTEX mission, back in 1993, [1] [2] has been the major breakthrough in space robotics up to present since for the first time, a space robot was controlled from Earth. Apart from fundamental work of Tom Sheridan [3] our close contact to and many discussions with Tony Bejczy [4] were decisive for inspiring our courage to successfully perform, e.g. the catching of a free-floating object and force controlled assembly of a bajonet closure with roundtrip delays between 5 and 7 seconds. Predictive simulation [5] the only proved method for compensating delays of this nature and shared compliance control [6] were the techniques we have applied in these situations. Tony Bejczy had performed impressive work in the field of bilateral force reflection handcontrollers in the late eighties. His big hope was to fly his handcontroller - kinematically different from any teleoperated robot - during this ROTEX experiment, we seek to test and find

out how force reflection works under zero gravity. Unfortunately he did not get the necessary support from NASA for this space experiment.

In general, the goal of a space telerobot is to perform some sort of telemanipulation of a spacecraft, an orbital object or a planet surface with more or less degree of autonomy. However, robotics hasn't yet shown its full potential in missions that require high dexterity levels. The ISS is still fully maintained by astronauts; four manned servicing missions on the Hubble Space Telescope raised original costs of $2.5B to $10B as of 2010 [7]. Robots are undeniable safer and more cost effective compared to on-site astronautic operations, though arguably, much more limited in terms of cognitive and manipulation abilities. Real-time teleoperation and force feedback are appealing technologies in these scenarios as they combine robotic capabilities with human intelligence and manipulation skills. Furthermore, those are technologies that have been thoroughly investigated for many years and find themselves in a rather mature developmental stage.

The issue of space debris demands a quick response to mitigate the increasing growth of debris population. Concerns are running high about the Low Earth Orbit environment, which could soon become unstable if nothing is actively done. Currently, there are different approaches under discussion to actively remove debris. Methods based on space robots, nets and harpoons are being analyzed. Space robots are arguably more complex and expensive but more flexible in terms of provided services.

On the other hand, it is expected robots will play a major role in future Mars and Moon exploration missions. Recent Mars and Moon exploration missions have already shown the potential of robotics in this field.

2 Force-feedback in Space

In 2005, ROKVISS (Robotic Component Verification on the ISS) was launched [8], [9]. The robot was a 2 degrees of freedom (d.o.f.) development with integrated torque sensors at each joint to allow compliant interaction with the environment. The robot was installed on the outer part of the Russian module of the ISS. A force feedback joystick along with the video streaming data was used to teleoperate the robot. A dedicated bidirectional point-to-point link - from the DLR to the ISS - was used which resulted in round trip delays of around 30 ms.

In 2008, the ARTEMIS telerobotics experiment was conducted [10]. In this experiment, the european geostationary satellite Artemis was used as space mirror to communicate a light-weight-robot (LWR), configured as a haptic device, and another LWR, used a teleoperator. Both systems were physically located in the DLR, separated by a distance of 5 meters. The communication link however traced a path of approximately 36,000 km: From the DLR, in Oberpfaffenhofen, to an ESA relay antenna in Redu, Belgium, through the GEO satellite Artemis in both directions, resulting in an average time delay of 620 ms.

In 2009, first experiments were conducted to investigate the effects of time delay in human performance when teleoperating such distant robots in space with force reflection. One of the results of this study revealed that a complex task whose mean execution time required 15 seconds in the absence of delay, demanded 115 seconds in the presence of 620 ms round trip delay. Clearly, time delay impairs human performance to very high levels. However, all the telemanipulation tasks were successfully fulfilled [10].

In 2011, KONTUR-1 mission was conducted. This mission was aimed at globally extending the communication link used in ROKVISS to virtually anywhere on Earth. The packages received from the ISS at the DLR-GSOC S-Band Antenna in Weilheim were rerouted to several locations in Germany and Russia through standard internet infrastructures (UDP), resulting in a hybrid link that was shaped by the delays, data losses and jitters of the space link and the internet communication.

It is well known in robotics, and in particular in the control field, that closed loops systems that include non-negligible time delays can produce negative effects on system stability. This issue is magnified in those closed loop systems that are characterized by tight couplings, that is, where high frequency control actions are required to capture a reasonable spectrum of the dynamics of the controlled system. For instance, in order to capture the interaction of a robot's end-effector while contacting a hard surface, a well established control-loop frequency is 1000Hz.

In general, these systems are governed by a bilateral controller because the control action takes place at both sides: At the master side, in order to control the haptic interface being manipulated by a human operator; and at the slave side, where the robot, or tele-operator, is located. The requirements for such a bilateral controller are remarkable as it needs to cope with a) large time delays, which furthermore are b) variable, producing therefore c) communication jitter and d) data losses. Further insight into the developed bilateral controller follows in the next sections.

In 2014 and 2015, an important milestone in on-orbit servicing has been achieved. The DLR's on-orbit servicing facility (OOS-SIM) was linked through the European communication satellite, ASTRA [11]. A haptic interface was used to control the robot manipulator mounted on the OOS-SIM's *servicer* satellite in order grasp, stabilize and dock the *target* satellite of the facility. The communication channel between haptic interface and servicer robot, shaped once again by space and internet links, resulted in an average round trip delay of 540 ms.

From August to December 2015, KONTUR-2 addressed scientific and technological questions for future planetary exploration. In this mission, ROKVISS' robot & haptic interface constellation was reversed, that is, a cosmonaut in the ISS controlled different robots located in the DLR through a force-feedback joystick. In

KONTUR-2, Earth and ISS have been used as a test-bed to evaluate and demonstrate a new technology for real-time telemanipulation from space [12].

3 Space Communication Infrastructures

Table 1 shows communication parameters registered during the main DLR telerobotics missions.

Experiment	Type	RT-Delay	P.Loss	Bandwidth
ROKVISS	ISS Link	20-30 ms	0.1%	256Kbps/4Mbps
ARTEMIS	GEO-Sat	620 ms	5.8%	4 Mbps
ASTRA	GEO-Sat	540 ms	2.6%	4 Mbps
KONTUR-2	ISS Link	20-30 ms	0.1%	256Kbps/4Mbps
K2 Training	Internet udp	65 ms	10%	10Mbps

Table 1
Main DLR space telerobotics missions communication parameters. P.Loss:Package Loss; RT:Round Trip

As it can be seen, time delay varies substantially depending on the particular communication infrastructure. Geostationary satellite communications (ASTRA, ARTEMIS) average higher than 0.5 seconds [13], [14]. The direct link used in ROKVISS on the other hand presents less delay.

In general, two communication approaches have been tested so far: n overflight point-to-point communication link and links based on geostationary satellite relays.

Table 2 compares the main features of both approaches.

Link type	Window (s)	Avg. RTD (ms)	Tested in
Direct link	8-10	20-30 ms	ROKVISS KONTUR-1/-2
Geo Relay	≥ 45	540 - 820 ms	ARTEMIS ASTRA, Haptics

Table 2
Direct link and geostationary relay based space links comparison

In KONTUR-2, two scenarios had to be considered in the design of the bilateral controller: ISS and training. The first is the nominal mission case, where the cosmonaut controls the robot from the ISS through a S-band link. The second, is a geographically distributed scenario for cosmonaut training purposes (see Fig. 1). Since the exact same system needs to operate in both, the requirements for the bilateral controller are clearly strengthened as both links are characterized by different communication parameters.

The cosmonaut training took place at the Gagarin Research and Test Cosmonaut Training Center (GCTC), located in Moscow. During the training, the cosmonaut

practiced with a joystick qualification model (QM) with identical characteristics of the ISS flight model (FM), and controlled the robot located at the DLR, in Germany, through the internet.

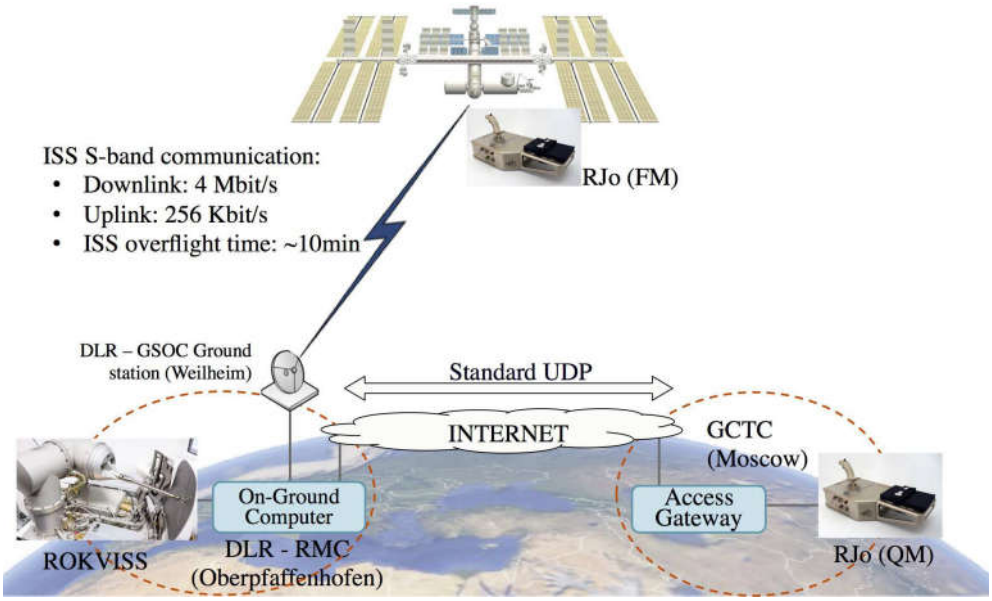


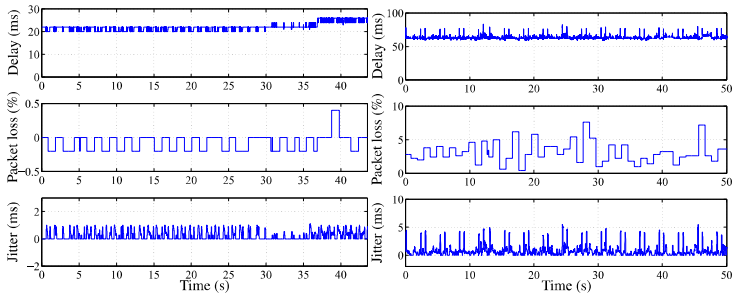
Figure 1
KONTUR-2 scenarios

Fig. 2(a) and Fig. 2(b) show the performance of ISS and internet UDP links. The nature of these two links is quite different in terms of time delay, data losses and jitter. The time delay for the ISS communication varied from 20 to 30 *ms* (corresponding to azimuth and horizon points) with mean negligible data losses. The internet training setup introduced a mean delay of 65 *ms* and highly oscillating package loss ratio, from 5% to 15% (due to the UDP protocol). Though more limited in bandwidth, the ISS link is higher in performance. However, shadowing can occur resulting in signal attenuation and in turn higher package loss ratios or even communication blackouts. On the other hand, the internet link measurements confirm a typical UDP behaviour.

On the other hand, the communication Ka-Band link used in the ASTRA experiment resulted in much larger time delays. Fig. 4 shows time delay, package loss ratio and jitter registered during one of the experiments in 2014 for a single trip. The round trip delay is therefore approximately twice as large.

4 Robotics and Space Debris

Spacecrafts are the only complex engineering systems without maintenance and repair infrastructure. Occasionally, there are have been space shuttle based servicing missions, starting with the Solar Maximum Repair Mission (SMRM) in 1984, but there are no routine procedures foreseen for individual spacecrafts. Most malfunc-



(a) Comm. Pars. ISS link

(b) Comm. Pars. GCTC internet link

Figure 2
Space and Internet links parameters registred in the KONTUR-2 mission. Very similar results were obtained in ROKVISS.

tioning spacecrafts require only a minor maintenance operation on orbit, a so-called On-Orbit Servicing (OOS) mission, to continue operational its work. Instead, they have to be replaced due to the lack of OOS opportunities. The accomplishment of OOS missions would, similar to terrestrial servicing procedures, be of great benefit for spacecraft operators, since a wide spectrum of use cases exists as, e.g. spacecraft assembly, orbit transfer, maintenance and repair, resupply, or even safe deorbiting.

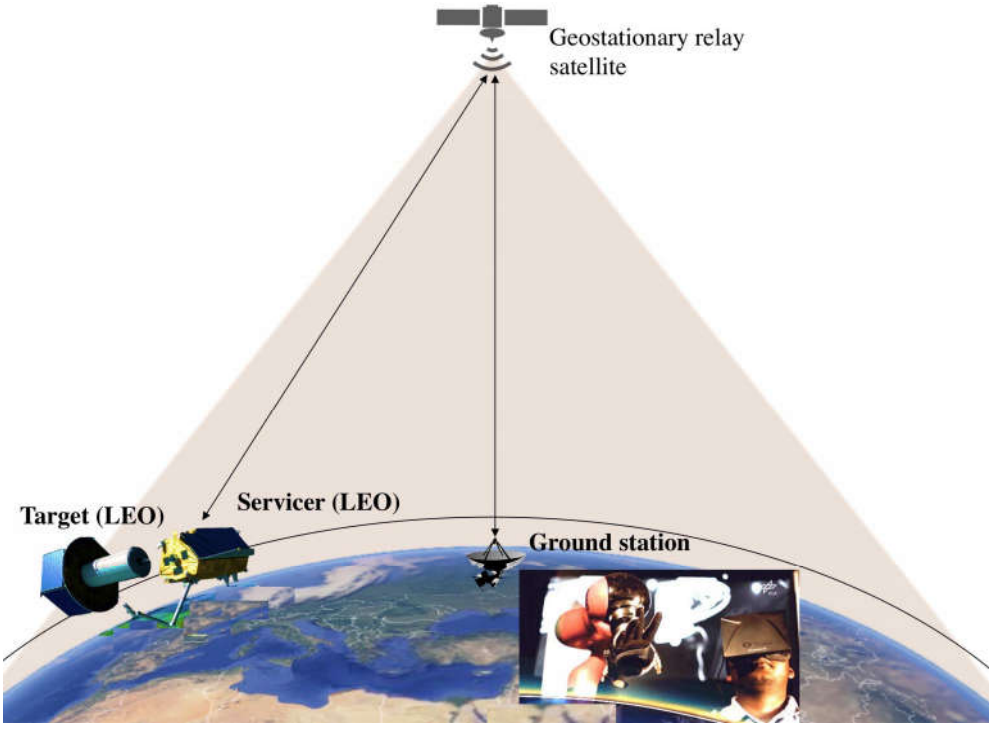


Figure 3
Targeted GEO relay based teleoperation system for on-orbit servicing

Roughly, a space robot consist of one or more manipulators mounted on a servicer spacecraft. The robot is capable of grasping a free-tumbling target and docking it onto the servicer, either to de-orbit it or for on-orbit servicing. Compared to classical robotics, the use of space robots for OOS presents some unique challenges that characterize its complexity and efficiency. Among the most remarkable are a) controlling the manipulator on a free-floating base, b) addressing impact dynamics c) reliability of visual sensors in space conditions and d) time delay between servicer and ground station.
In general, there are two strategies to control the space robot: semi-autonomy, in which the robot is controlled through visual servoing, and teleoperation, in which a human operator controls the robot from ground in real-time through a haptic device. Advantages and disadvantages of each approach are closely related to the mentioned challenges.

In general, the semi-autonomous system is higher in complexity and its effectiveness is highly dependent on the sensor capabilities to cope with extreme lighting conditions. On the other hand, teleoperation can be arguably more rudimentary but is less dependent on the reliability of the sensor and vision algorithms. In this paper we explore feasibility of the second strategy, also known as telepresence. One of the key elements in this approach is the combination of robotic feedback control capabilities with human manipulation skills. Generally speaking, the increased complexity of an autonomous system is well justified in applications characterized by systematic or repetitive tasks. OOS tasks are, however, rather unique and distinctive and might be better addressed by means of teleoperation.

In particular, On-Orbit Servicing (OOS) in LEO presents a special difficulty since direct contact between a ground station and the servicing spacecraft is only given in small time intervals, that is, when the spacecraft is flying over the terrestrial antenna(s). However, the feasibility of OOS operations is highly dependent on whether and how long a communication link between the controlling ground station and the servicer spacecraft can be established. A good example is given by the space shuttle based OOS missions of the Hubble Space Telescope (HST). Each task required several EVAs resulting in a total EVA time of more than 24 hours. An OOS mission, which is telepresently controlled from ground demands, therefore an equivalent amount of contact time. According to Table 2, using direct communication links in LEO would require several weeks or a complex ground station network of antennas globally distributed on Earth. Since the HST orbits the Earth at approximately 570 km, 4-8 orbit revolutions per day exist, in which a human operator could teleoperate a robotic servicer for maximum 10 minutes per orbit revolution.

Thus, the use of geostationary satellites a promising approach for space telerobotics as it can increase the mean acquisition time of the spacecraft in LEO up to more than 1 hour per orbit revolution. The two space missions mentioned above, ARTEMIS [10] and ASTRA [11] address these questions (see Fig. 3). It is clear that the use of geostationary data relay satellites drastically increases the round trip delay of the signal, that is, the time between operator action and spacecraft feedback. The main goal of both experiments was to prove that the utilization of geostationary (GEO) data relay satellites for OOS is reconcilable with a telepresent control of the servicer

spacecraft, that is, with force-feedback teleoperation. The real-time link given by the geostationary ARTEMIS satellite resulted in a mean delay of 620 ms; similar results were obtained using ASTRA (see registred singe trip delay plots in Fig. 4).

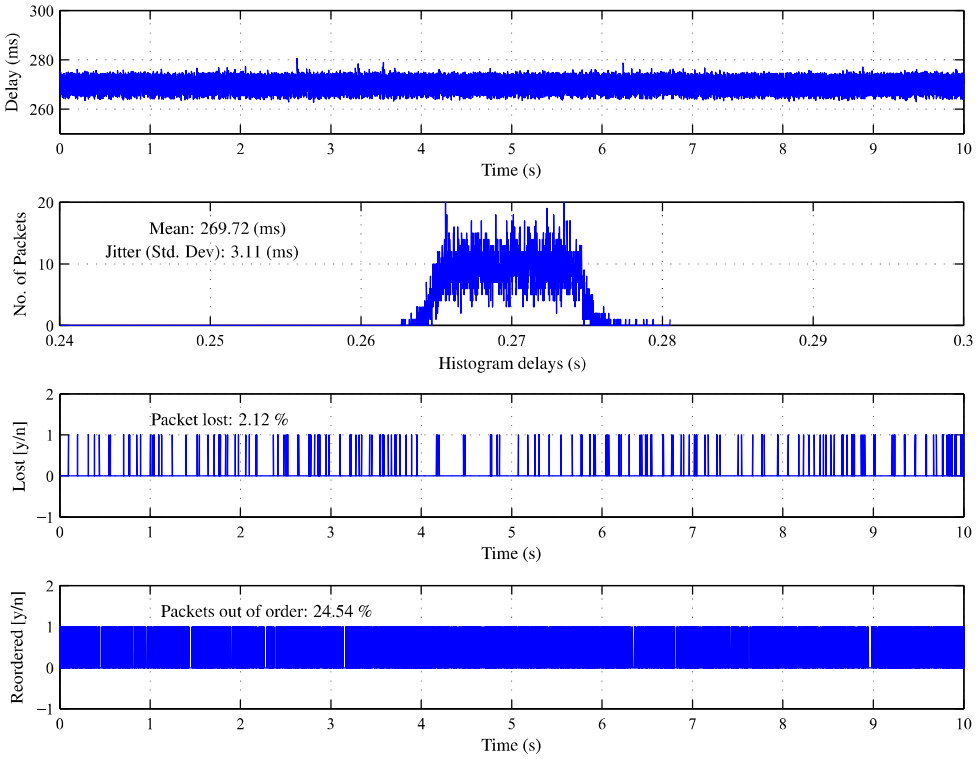


Figure 4
Communication link properties for a single channel (i.e. forward or feedback channel, see Fig. 5)

At the present time, it can be concluded that bilateral controller is an effective mechanism to deal with large and variable time delays, that is, the control structure between the space robot and the haptic interface on ground. In other words, stable and reliable force-feedback from the task space to the operator can be provided. This has been proved from a stability and control performance point of views. Efficiency in dealing with human factors was initially investigated during the ARTEMIS experiment. However, further studies are required to evaluate the efficiency of the approach for e.g. performing satellite grasping and docking maneuvers.

5 Controlling robots *in* and *from* Space: Bilateral Control

The bilateral controller enables a human operator to control the space robot using a haptic interface, through which he feels the interaction forces from the slave. In general, bilateral control is challenging as it establishes a very long closed control loop. For instance, the systems mentioned above with geostationary relays results in closed loops systems of length of approximately 36,000 Km. Furthermore, the

controller needs to be robust against data losses and jitters, i.e. variations of time delay at each sampling step. Furthermore, a special feature of this loops is that they run at high sampling rates. This is required to achieve reliable force-feedback since the bandwidth of the human haptic spectra is characterized by very frequencies, i.e. greater than 1000 Hz. This can be well seen in hard contact situation in which the force transition happens in very short time intervals. In all of the previous experiments, the Time Domain Passivity Control approach (TDPA) with the Time Delay Power Network (TDPN) concept has been successfully applied. This control approach is based on monitoring the virtual energy that results from the delayed communications and applying a variable damper - called Passivity Controller.- which is a function of the observed energy (Passivity Observer). See [15], [16] and [17] for a review on the TDPA and TDPN concepts.

A final remark follows related to the two well established methods for controlling a robot manipulator: position and torque control. The first is widely used since a position interface is available in every robot. The second requires torque sensors inside the robot structure, something that is only available in a few comercial robot manipulators due to its high costs. Nevertheless, the trend in the last years is evolving towards torque controlled robots, specially in service robotics. The main reason to that is that a torque interface allows to implement impedance controllers. With impedance control, a desired compliant behaviour can be implemented at the robot end-effector. The robot can behave as soft and flexible manipulation tool or e.g. as a heavy and stiff tool. Position control achieves higher accuracy degrees but in general it does not allow to render compliant behaviors. Therefore, it is less tolerant to uncertainties in the task execution and modelling. This is specially true in space telerobotics missions as they are characterized by harshness environments. Extreme lightning conditions result in limited sensor capabilities; limited satellite geolocation capabilities result in limited positioning accuracy; and non-realtime / low-bandwidth communication infrastructures result in delayed tasks executions. Arguably, position control can achieve higher positioning accuracies in well-structured environments, enjoys higher technical readiness levels (TRL) and is less expensive. On the other hand, impedance control can naturally handle inaccurate physical interactions, can better cope with latencies and can adapt its impedance to each task (e.g. for matching impedance between the robot manipulator and a target spacecraft). For theses reasons, torque control (i.e. impedance control) in all of the above experiments.

6 Conclusion

After our long term investigations on up to which amount of roundtrip delay force reflection works at all, we may now state that up to around 650 ms delay, time domain passivity algorithms can compensate the delays, with a satisfactory performance. Larger roundtrip delays, approximating 900 ms, are still possible, but of course with a decaying efficiency. But, interestingly enough this is more or less the resulting delay to control a robot system in lower Earth orbits via a geostationary re-

lay satellite (assuring communication coverage of typically around 45 minutes). As seen in the various experiments presented in this article, controlling a robot through a link based on a geostationary relay plus an internet connection leads to unavoidable delays of 540 to 620 ms. Today's fast internet infrastructures allow data streams between continents with roundtrips delays of less than 100 ms. Force-feedback and so telemanipulation can therefore not only be globally covered but also be used to in the earth's orbital space.

Our future work will seek to further establish force-feedback teleoperation as a technology for supporting astronautic tasks. To that end, the use of geostationary relay satellite is crucial to achieve intervention times close to one hour. It's worth mentioning that once it has been proved that internet allows real-time transmission of haptic streams (by using appropriate control methods), we expect that force-feedback teleopera-tion will be key technology for terrestrial applications too: From elderly care and tele-nursing to maintenance and repair of energy plants.

References

[1] G. Hirzinger, "Rotex the first space robot technology experiment," in *Experimental Robotics III*, ser. Lecture Notes in Control and Information Sciences, T. Yoshikawa and F. Miyazaki, Eds. Springer Berlin Heidelberg, 1994, vol. 200, pp. 579–598. [Online]. Available: http://dx.doi.org/10.1007/BFb0027622

[2] G. Hirzinger, K. Landzettel, and J. Heindl, "Rotex: space telerobotic flight experiment," in *Proceedings of SPIE International Symposium on Optical Tools for Manufacturing and Advanced Automation*, Boston, USA, 1993.

[3] T. B. Sheridan, "Space Teleoperation Through Time Delay: Review and Prognosis," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 592–606, October 1993.

[4] A. Bejczy, "Towards advanced teleoperation in space," *Teleoperation and Robotics in Space, Prog. Astronaut, Aeronaut*, vol. 161, no. 5, pp. 494–501, 1994.

[5] A. Bejczy and W. S.Kim, "Predictive displays and shared compliance control for time-delayed telemanipulation," in *Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS '90. IEEE International Workshop on*, Jul 1990, pp. 407–412 vol.1.

[6] W. Kim, B. Hannaford, and A. Bejczy, "Force-reflection and shared compliant control in operating telemanipulators with time delay," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 2, pp. 176–185, Apr 1992.

[7] J. William F. Ballhaus, *James Webb Space Telescope (JWST) Independent Comprehensive Review Panel (ICRP) Final Report*, ser. NASA contractor report. National Aeronautics and Space Administration, Scientific and Technical Information Branch, 2010, no. v. 3.

[8] G. Hirzinger, K. Landzettel, D. Reintsema, C. Preusche, A. Albu-Schäffer, B. Rebele, and M. Turk, "Rokviss-robotics component verification on iss," in *Proc. 8th Int. Symp. Artif. Intell. Robot. Autom. Space (iSAIRAS)(Munich 2005) p. Session2B*, 2005.

[9] A. Albu-Schäffer, W. Bertleff, D. Rebele, B. Schäfer, K. Landzettel, and G. Hirzinger, "Rokviss - robotics component verification on iss current experimental results on parameter identification," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 3879–3885.

[10] E. Stoll, U. Walter, J. Artigas, C. Preusche, P. Kremer, G. Hirzinger, J. Letschnik, and H. Pongrac, "Ground verification of the feasibility of telepresent on-orbit servicing," *Journal of Field Robotics*, vol. 26, no. 3, pp. 287–307, 2009.

[11] J. Artigas, Balachandran, M. De Stefano, M. Panzirsch, R. Lampariello, J. Letschnik, J. Harder, and A. Albu-Schaeffer, "Teleoperation for on-orbit servicing missions through the astra geostationary satellite," in *(to be pusblished) Aerospace Conference, 2016 IEEE*. IEEE, 2016, pp. 1–20.

[12] J. Artigas, R. Balachandran, M. Stelzer, B. Weber, J.-H. Ryu, C. Riecke, and A. Albu-Schaeffer, "KONTUR-2: Force-feedback Teleoperation from the International Space Station," in *To be published at IEEE ICRA 2016*.

[13] E. Stoll, U. Walter, J. Artigas, C. Preusche, P. Kremer, G. Hirzinger, J. Letschnik, and H. Pongrac, "Ground verification of the feasibility of telepresent on-orbit servicing," *J. Field Robot.*, vol. 26, no. 3, pp. 287–307, Mar. 2009. [Online]. Available: http://dx.doi.org/10.1002/rob.v26:3

[14] R. Lampariello, N. Oumer, J. Artigas, W. Rackl, G. Panin, R. Purschke, J. Harder, U. Walter, J. Frickel, I. Masic, K. Ravandoor, J. Scharnagl, K. Schilling, K. Landzettel, and G. Hirzinger, "Forrost: Advances in on-orbit robotic technologies," in *Aerospace Conference, 2015 IEEE*, March 2015, pp. 1–20.

[15] J. Artigas, J.-H. Ryu, C. Preusche, and G. Hirzinger, "Network representation and passivity of delayed teleoperation systems," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 177–183.

[16] J. Artigas, J.-H. Ryu, and C. Preusche, "Time domain passivity control for position-position teleoperation architectures," *Presence: Teleoperators and Virtual Environments*, vol. 19, no. 5, pp. 482–497, 2010.

[17] J.-H. Ryu, J. Artigas, and C. Preusche, "A passive bilateral control scheme for a teleoperator with time-varying communication delay," *Mechatronics*, vol. 20, no. 7, pp. 812–823, 2010.