Acta Universitatis Sapientiae

Electrical and Mechanical Engineering

Volume 10, 2018

Sapientia Hungarian University of Transylvania
Scientia Publishing House

Contents

Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 10 (2018) 5-19

DOI: 10.2478/auseme-2018-0001

An Algorithm for the Calculation of the Dwell Time Constraint for Switched $\mathcal{H}_\infty$ Filters

Zsolt HORVÁTH[1], András EDELMAYER[2,3]

[1] School of Postgraduate Studies of Multidisciplinary Technical Sciences, Faculty of Technical Sciences, Széchenyi István University, Győr, e-mail: zsolt2.horvath@audi.hu
[2] Department of Informatics Engineering, Faculty of Technical Sciences, Széchenyi István University, Győr, e-mail: edelmayer@sze.hu
[3] Systems and Control Laboratory, Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, e-mail: edelmayer@sztaki.mta.hu

Manuscript received September 28, 2018; revised November 30, 2018.

Abstract: This paper presents a numerical algorithm for determining the minimum dwell time constraint for switched linear $\mathcal{H}_\infty$ fault detection filters. When applying switched systems, ensuring the stability is a crucial target, which can be guaranteed, when we switch slowly enough between the subsystems, more precisely when the intervals between two consecutive switching instants, called dwell time, are large enough. The problem formulation is based on multiple Lyapunov functions and is expressed through a special form of linear matrix inequalities (LMIs), which include a nonlinear term of the dwell time. This represents a multivariable, time dependent optimization problem. As a result, the task cannot be treated as a simple feasibility problem involving a LMI solver as it is widely used in applications of the linear control. To solve these special LMIs, we propose a numerical algorithm, called $\boldsymbol{T_d}$-iteration, which combines the procedure of interval halving with an LMI solver. The algorithm implemented in MATLAB shows its applicability as well as suggest further benefits for the switched linear control and filter synthesis.

Keywords: Switched linear system, dwell time, switched $\mathcal{H}_\infty$ fault detection filter, MFARE

1. Introduction

Switched systems for purpose of nonlinear control have been studied extensively in the two past decades and useful results are now available, see e.g. [1], [2], [3], [4] and [5]. As it was stated by several authors e.g. (Liberzon and Morse in 1999, Hespana in 2004, Chen and Saif in 2004, Colaneri in 2008), the asymptotic stability can be ensured when we switch slowly enough between the

subsystems, more precisely when the intervals between two consecutive switching instants -called dwell time-, are large enough. This problem has been specially addressed in the synthesis of switched state estimator of Luenberger type, e.g. (Prandini in 2003, Chen and Saif in 2004) and it is also a crucial part in our objective of designing a switched linear $\mathcal{H}_\infty$ fault detection filter. In earlier researches different methods have been proposed for determining the minimum dwell time, see [4], [6], [7], [8], [9] and [10]. The most commonly used algorithms, such as e.g. the representation based on Kronecker products (Geromel and Colaneri, 2006), or Logic-Based Switching Algorithms (Hespana, 1998) are constructed using multiple Lyapunov functions and expressed in form of linear matrix inequalities (LMIs), see in [6], [7], [9], [10] and [11].

Since we deal with $\mathcal{H}_\infty$ filtering, the basic Lyapunov theorem needs to be extended to cope with performance requirements such as the root mean square (RMS) property of a switched system, which corresponds to finding an upper bound of the minimum dwell time. To this aim, in our research we consider a method used by (Geromel and Colaneri, 2008) for $\mathcal{H}_\infty$ nonlinear control and we have adopted it to the classical $\mathcal{H}_\infty$ detection filtering problem, see in [12], [13], [14], [15] and [16]. More exactly, the concept of the switched $\mathcal{H}_\infty$ control in [7] can be associated to the switched $\mathcal{H}_\infty$ filtering problem by duality and sufficient stability conditions can be derived.

LMIs are nowadays widely used powerful tools for solving complex optimization problem in the field of control engineering, see e.g. [17], [18], [19] and [20]. The commonly used advanced methods, however, refer to a LMI solver only accept formulation where the decision variables are included in linear terms. On the contrary, our problem formulated as LMIs, which include the term of matrix-exponential function with the dwell time, is consequently nonlinear. As a result, the task cannot be treated as a simple feasibility problem, see e.g. [17], [21] and [22]. Despite of the widespread referring to this special LMI formulation, however, there can't be found any solution algorithm about it in the control literature. In this paper we present an algorithm to calculate the common minimum dwell time, assuring each specified $\mathcal{H}_\infty$ level calculated separately for each single filter.

The contents of this paper are as follows. After the introduction, in Section II the dwell time condition for assuring stability of the switched linear $\mathcal{H}_\infty$ filter is presented. The main outcome is a special form of LMIs including the nonlinear term with the dwell time which represents a multivariable time dependent optimization problem. Section III presents the proposed numerical algorithm for the calculation of the common minimum dwell time assuring each specified $\mathcal{H}_\infty$ level. In Section IV the T_d-iteration algorithm is applied on an illustrative example in MATLAB. In Section V the main results are summarized and the paper is concluded with some final remarks.

2. Stability of the state estimation error dynamics involving the dwell time constraint

The synthesis technique proposed below is originated from results (Geromel and Colaneri, 2008) with focus on the application to robust nonlinear control, see in [7] and [6]. We have adopted this concept to a $\mathcal{H}_\infty$ detection filtering problem, which will be introduced in this chapter. However, in order to improve the detection's performance, we formulate our concept slightly different from theirs. That means, instead of calculation of the minimum dwell time assuring a common specified $\mathcal{H}_\infty$ level for each controller, we determine the common minimum dwell time to each specified $\mathcal{H}_\infty$ level calculated separately for each single filter. In the following we are referring to the concept in [12], which's system-description has been extended to a switched linear system.

Extending the switched linear system representation in [6] to the concept of perturbed system, see in [12], the extended switched linear system subjected to disturbance and faults, can be represented in state space form as follows:

$$\begin{aligned} \dot{x}(t) &= A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + B_{\kappa\sigma(t)}\kappa(t) + \sum_{i=1}^{k} L_{i\,\sigma(t)}\nu_i(t), \quad x(0) = \xi, \\ y(t) &= C_{\sigma(t)}x(t), \end{aligned} \tag{1}$$

where for all $t \geq 0$, $x(t) \in \mathbb{R}^n$ is the state vector, $\xi \in \mathbb{R}^n$ is the arbitrarily fixed initial condition, $u(t) \in \mathbb{R}^m$ is the input vector, $y(t) \in \mathbb{R}^p$ is the output vector, $\sigma(t)\colon [0,\infty) \to \Theta$ is the piecewise constant switching function. $A_{\sigma(t)} \in \mathbb{R}^{nxn}$, $B_{\sigma(t)} \in \mathbb{R}^{nxm}$ and $C_{\sigma(t)} \in \mathbb{R}^{pxn}$ are an appropriate matrices. Assume, that the pairs $(A_{\sigma(t)}, C_{\sigma(t)})$ are observable for all $t \geq 0$. For further consideration denote n_e the number of subsystems, $\Theta = \{1, \dots, n_e\}$ an index set and $q = 1, \dots, n_e$ the sequence number of the switchings. $B_{\kappa\,\sigma(t)} = [B_w, L_\Delta]$ denotes the worst-case input direction and $\kappa(t) \in L_2\,[0, T]$ is the input function for all $t \in \mathbb{R}_+$ representing the worst–case effects of modelling uncertainties and external disturbances. The cumulative effect of a number of k faults appearing in known directions L_i of the state space is modelled by an additive linear term $\sum L_{i\,\sigma(t)}\,\nu_i(t)$. $L_i \in \mathbb{R}^{nxs}$ and $\nu_i(t)$ are the fault signatures and failure modes respectively. $\nu_i(t)$ are arbitrary unknown time functions for $t \geq t_{ji}$, $0 \leq t \leq T$, where t_{ji} is the time instant when the i-th fault appears and $\nu_i = 0$, if $t < t_{ji}$. If $\nu_i(t) = 0$, for every i, then the plant is assumed to be fault free. Assume, however, that only one fault appears in the system at a time.

Denote t_ℓ and $t_{\ell+1}$ successive switching times satisfying $t_{\ell+1} - t_\ell \geq \tau_D$. Then the piecewise constant switching function between two consecutive switching instants as $\sigma(t)\colon [0,\infty) \to \Theta$ for all $t(t_\ell, t_{\ell+1}]$ ensures, that the equilibrium point $x = 0$ of the system in (1) is globally asymptotically stable.

The referred constant $\tau_D > 0$ is called the dwell time. Consequently, when designing a switched system one also has to make sure, that the time difference between two consecutive switching instants not smaller than τ_D, then the asymptotical stability of the switched linear system is preserved, see e.g. in [1], [3], [4] and [7].

Generally interpreted, the fault detection filtering is done by estimating the states of the subjected system. Of course, we consider now a switched linear system approach, where the q-th sub-filter is selected whenever the q-th subsystem is active. The stability of the state estimation error dynamics may be a crucial part of such a design, which can be ensured when we switch slowly enough between the subsystems, to allow the transient effects to dissipate (Chen and Saif, 2004), (Prandini, 2015).

The state estimator for the system description (1) can be represented by the switched system as follows. Let $z \epsilon \mathbb{R}^p$ denote the output signal, then the state estimate can be obtained as

$$\begin{aligned} \dot{\hat{x}}(t) &= \left(A_{\sigma(t)} - Y_{\sigma(t)} C^T{}_{\sigma(t)} C_{\sigma(t)}\right) \hat{x}(t) + B_{\sigma(t)} u(t) + Y_{\sigma(t)} C^T{}_{\sigma(t)}\, y(t), \\ \hat{y}(t) &= C_{\sigma(t)} \hat{x}(t), \\ \hat{z}(t) &= C_{z\sigma(t)}\, \hat{x}(t), \end{aligned} \tag{2}$$

where $\hat{x} \in \mathbb{R}^n$ represents the observer state, $\hat{y} \in \mathbb{R}^p$ represents the output estimate, and $\hat{z} \in \mathbb{R}^p$ is the weighted output estimate, $Y_{\sigma(t)}$ is a positive definite matrix as a solution of the optimization problem in (5) and $C_{z\sigma(t)}$ is the estimation weighting.

The equation of the state estimation error for (2) is expressed as

$$\begin{aligned} \dot{\tilde{x}}(t) &= (A_{\sigma(t)} - Y_{\sigma(t)} C^T_{\sigma(t)} C_{\sigma(t)})\, \tilde{x}(t) + B_{\kappa\sigma(t)} \kappa(t) + \sum_{i=1}^{k} L_{i\sigma(t)}\, \nu_i(t), \\ \varepsilon(t) &= C_{z\sigma(t)} \tilde{x}(t), \end{aligned} \tag{3}$$

where $\tilde{x}(t)$ and $\varepsilon(t)$ are defined as

$$\begin{aligned} \tilde{x}(t) &= x(t) - \hat{x}(t), \\ \varepsilon(t) &= z(t) - \hat{z}(t), \end{aligned} \tag{4}$$

As the switching occurs within the finite set of $q \in \Theta = \{1, \ldots, n_e\}$ subsystems, the system description in (1) and consequently in (2) and (3) can be simply represented by the matrices $(A_q,\ B_q,\ B_{\kappa q},\ C_q,\ C_{zq},\ L_{iq},\ Y_q)$, $q \in \Theta$. Assume that all matrices $A_q,\ q \in \Theta$ are Hurwitz.

By duality we can associate the $\mathcal{H}_\infty$ control problem of the switched linear system described in [12] to our switched $\mathcal{H}_\infty$ filtering task, the synthesis of

which is based on the Modified Riccati Equation (MFARE), that can be formulated for switched linear system as

$$A_q Y_q + Y_q A_q^T + Y_q \left(\frac{1}{\gamma_q^2} C_{z\,q}^T C_{z\,q} - C_q^T C_q \right) Y_q + B_{\kappa q} B_{\kappa\,q}^T = 0, \tag{5}$$

for all $q \in \Theta$. In (5) $\gamma_q > 0$ are positive rational constants and $Y_q \in R^{nxn}$ denote the decision variables which are positive definite matrices.

Following the steps of the synthesis procedure in [7], the MFARE can be factorized in form of Riccati Equation as

$$H_q Y_q + Y_q H_q^T + Q_q = 0, \ \forall q \in \Theta, \tag{6}$$

where the associated matrices are

$$W_q = Y_q \left[C_{z\,q}^T C_q^T \right] \begin{bmatrix} -\frac{1}{\gamma_q^2} I & 0 \\ 0 & I \end{bmatrix}, \tag{7}$$

$$H_q = \left(A_q - W_q \begin{bmatrix} C_{zq} \\ C_q \end{bmatrix} \right), \tag{8}$$

$$Q_q = \left(W_q \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} W_q^T + B_{\kappa q} B_{\kappa\,q}^T \right) - \gamma_q^2 W_q \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} W_q^T . \tag{9}$$

We have to note, that the optimal gain W_q is determined from the unique stabilizing solution to MAFARE and the matrix H_q is Hurwitz for each $q \in \Theta$. Since Q_q depends on the γ_{minq} value, $Q_q \geq 0$ is not guaranteed for any $q \in \Theta$. However, (6) admits a positive definite solution, since that was created by factorizing the MFARE. It is to note, that for solving the LMIs in (11) the condition $Q_q \geq 0$ is necessary, hence, if $Q_q \geq 0$ does not hold, $\gamma_q > \gamma_{minq}$ should be chosen such that $Q_q \geq 0$ holds.

For any $\sigma(t): [0, \infty) \to \Theta$ and for all $t \in (t_\ell, t_{\ell+1}]$, where $t_{\ell+1} = t_\ell + T_\ell$ with $T_\ell \geq T_d > 0$ and at $t = t_{\ell+1}$ the switching jumps to $\sigma(t) = j \in \Theta$, where the corresponding solution of the Lyapunov function along a trajectory of the switched filter state estimation error (2) is expressed by

$$\mathrm{V}\left(\tilde{\mathrm{x}}\left(t_\ell\right)\right) = \tilde{\mathrm{x}}\left(t_{\ell+1}\right)^{\mathrm{T}} Z_j \, \tilde{\mathrm{x}}\left(t_{\ell+1}\right) = \tilde{\mathrm{x}}\left(t_\ell\right)^{\mathrm{T}} e^{H_q{}^T T_\ell} Z_j e^{H_q T_\ell} \, \tilde{\mathrm{x}}\left(t_\ell\right), \tag{10}$$

where $Z_j \in R^{nxn}$ is a positive definite matrix.

The $\mathcal{H}_\infty$ control problem described in [7] can be associated to the $\mathcal{H}_\infty$ filtering problem by duality. Based on (6) and the Lyapunov function formulated along a trajectory of the state estimation error (10), one can derive time varying LMIs which can be used to obtain the common minimum dwell time constraint satisfying each $\mathcal{H}_\infty$ filter's specification.

Assume that for a given T_d there exists a collection of positive definite matrices $\{Z_1, \dots, Z_{ne}\}$ of compatible dimensions such that the LMIs

$$\begin{aligned} & H_q Z_q + Z_q H_q^{\,T} + Q_q < 0, \\ & \forall q \in \Theta, \\ & e^{H_q T_d} Z_j e^{H_q^{\,T} T_d} - Z_q + Y_q < 0, \\ & \forall q \neq j \in \Theta, \end{aligned} \tag{11}$$

hold under the worst-case input assumption in (1) for any switching signal $\sigma(t): [0, \infty) \to \Theta$ satisfying the condition $T_d = t_{\ell+1} - t_\ell \geq T_{dmin}$. Then, the equilibrium solution of the state estimation error (2) is globally asymptotically stable.

3. A numerical algorithm for determining the common minimum dwell time assuring each specified $\mathcal{H}_\infty$ level

As we have shown in the previous chapter, the problem of determining the minimum dwell time can be obtained by solving the set of LMIs (11). According to our idea, by means of combining an algorithm of interval halving for a fixed scalar T_d the LMIs can be treated as well as solved as a feasibility problem and the common minimum dwell time can be calculated. Before doing that, the MFARE in (5) was factorized and Y_q, H_q and Q_q matrices were obtained for each $q \in \Theta$. In [16] it is explained how the MFARE can be formulated and solved as a LMI. Then (11) can be represented via the following optimization problem:

$$\begin{cases} \min T_d \\ s.t.\ \ Z_q > 0 \\ \quad Z_j > 0 \\ \quad H_q Z_q + Z_q H_q^{\ T} + Q_q < 0 \\ \quad e^{H_q T_d} Z_j e^{H_q^{\ T} T_d} - Z_q + Y_q < 0. \\ \forall q \neq j \in \Theta \end{cases} \tag{12}$$

The main benefit of the LMI formulation is, that it defines a convex constraint with respect to the variable vector. For that reason, it has a convex feasible set which can be found guaranteed by means of convex optimization procedure. When using an LMI solver, however, it usually only accepts formulation where the decision variables are included in linear terms. Unfortunately the LMIs in (12), which include the term of matrix-exponential with the design scalar variable T_d, are nonlinear, consequently the task cannot be treated simply as a feasibility problem, see in [18], [19], [20] and [22]. To overcome these difficulties we implemented an algorithm called T_d-iteration, in which an interval halving method is used iteratively. The algorithm reduces gradually the value of the T_d scalar variable until the constraints of the LMIs in (12) are no longer feasible, consequently any of the Z_q matrices, has no longer positive definite solutions. The T_{dmin} which is so reached, is within the limits given by an arbitrarily small tolerance $\varepsilon > 0$ and is the minimum dwell time, such it holds that $T_{dmin} \leq \tau_D$.

The algorithm for the feasibility problem of determing the common minimum dwell time can be formulated as follows:

The inputs for the method are: Y_q, H_q and Q_q matrices for each $q \in \Theta$ which can be obtained from (5), and from (8), (9), respectively.

eps is the relative accuracy of the solution, T_{dmax} is the right limit of the interval (the left limit is zero).

The inner variables are: a, b and i. They stand for assignation of interval and counting cycle respectively. The T_{dm} variable contains the value of T_d at the end of the iteration.

The outputs are: Z_q matrices $q \in \Theta$ are positive definite decision variables, the T_d is the step size (midpoint). T_{dmin} contains the T_d value when the iteration is finished.

Each iteration performs the following steps:

1. Calculate T_d, the midpoint of the interval, which is assigned by a and b. That is $T_d = a + (b - a)/2$;

2. Calculate the matrix exponential function $e^{H_q T_d}$ for the fixed T_d value and substitute its values in (11);
3. Solve the LMIs in (11) as a feasibility problem by the MATLAB function *feasp* [22], which returns both the scalar value of t_{min} as a measure of the feasibility and the feasibility decision vector *xfeas;*
4. Call the MATLAB function *dec2mat* which returns the solutions for Z_q;
5. If the feasibility criteria with fixed T_d are not satisfied, that is $t_{min} \geq 0$, then the upper and lower bounds of interval are changed;
   Otherwise the value of T_d is saved, that is $T_{dm} = T_d$ and the iteration is continued;
6. Examine whether the new interval assigned by *b-a* reached the relative accuracy of the solution - called *epsilon*:
   - If not, the iteration is repeated;
   - If yes, the iteration is finished and the Z_q matrices are calculated based on the previous value of T_d. Additionally, $T_{dmin} = T_{dm}$.

In the following the MATLAB script for the T_d-iteration an illustrative example is presented for solving the multivariable time dependent optimization problem in (12). It was implemented for synthesis of a switched linear $\mathcal{H}_\infty$ filter, which consits of three subsystems for purpose of demonstation.

An approximate calculation of the minimum dwell time based on the L_2-norm of the state estimator system

Another, and very conservative approach is calculating the dwell time based on the L_2-norm of the H_q, see in [24]. This theorem says, that for each subsystem $q \in \{1, 2, \ldots, n_e\}$ because H_q is Hurwitz, there exist $a_q \geq 0$ and $\lambda_q > 0$ such that for all $q \geq 0$, it can be written

$$\left\| e^{H_q t} \right\| \leq e^{a_q - \lambda_q t}, \tag{13}$$

$$\text{where} \quad \|H\| = \sqrt{\lambda_{max}(H^T H)}. \tag{14}$$

Based on this the dwell time is given as

$$\tau_D > \max_{q=1,2,\ldots,\text{ne}} \left\{ \frac{a_q}{\lambda_q} \right\}. \tag{15}$$

Using the similarity transformation for matrix *H*, that is

$$H = TDT^{-1}. \tag{16}$$

The condition for the asymptotic stability expressed using T can be written as:

$$\|e^{Ht}\| \leq ||T||e^{\lambda t}||T^{-1}||. \tag{17}$$

Then using the T and D matrices, the parameter for calculating (14) can be obtained as

$$a = ln(\|T\|\|T^{-1}\|) \text{ and } \lambda = max_{1\leq i\leq n}\{\lambda_q\}. \tag{18}$$

4. An illustrative example

T_d – iteration algorithm

In the following the MATLAB script, for the T_d-iteration an illustrative example is presented for solving the multivariable time dependent optimization problem in (12). It was implemented for the synthesis of switched linear $\mathcal{H}_\infty$ filter and consists of three subsystems for purpose of demonstation.

Consider that the matrices Y_q, H_q and Q_q have been formerly calculated from (5), (8) and (9). Note, that these calculations are not presented in this paper.

```
% LMIFeaspDWT3m.m - 2018.06.25
% Calculating the minimum dwell time assuring the specified H-inf level
% Matrices derived from MFARE
% Subsystem 1
H1 = [-107.4991 15.4019 27.8936; 80.9958 -564.3523 1.9649; 0.1119 0.3748 -
8.6469];
Q1 = 1.0e+005 * [0.1810 -0.2182 -0.0002; -0.2182 2.7837 0.0006; -0.0002 0.0006
0.0000];
Y1 = [81.6978 -17.0191 -0.1127; -17.0191 244.1844 0.2370; -0.1127 0.2370
0.0103];
% Subsystem 2
H2 = [-91.9324 20.9809 28.0690; 106.1269 -611.2837 1.5668; 0.1088 0.4258 -
8.6914];
Q2 = 1.0e+005 * [0.1209 -0.2449 -0.0002; -0.2449 3.0927 0.0006; -0.0002 0.0006
0.0000];
Y2 = [61.5844 -18.0647 -0.1088; -18.0647 249.8336 0.2452; -0.1088 0.2452
0.0120];
% Subsystem 3
H3 = [-78.3760 25.7621 27.8720; 132.6617 -626.8514 1.9901; 0.1106 0.4382 -
8.5647];
Q3 = 1.0e+005 * [0.0854 -0.2658 -0.0001; -0.2658 3.1959 0.0008; -0.0001 0.0008
0.0000];
Y3 = [48.0245 -19.4905 -0.1127; -19.4905 250.7915 0.2733; -0.1127 0.2733
0.0142];

% Interval-halving
I=eye(3);
eps =1e-3;% the relative accuracy of the solution
Tdmax=3;        % the upper limit of the interval
Td=Tdmax; % the step size (midpoint)
b=Tdmax;        % the initial upper limit of the interval
a=0;       % the initial lower limit of the interval
i=0;       % initialization of the step counter
```

```
while (b-a)>eps      % examine whether the new interval reached the relative
accuracy
     Td = a+(b-a)/2;      % interval-halving
     i = i+1              % number of the iterations

% calulation of the matrix exponential functions each subsytems
     expHT1 = expm(Td*H1)
     expHT2 = expm(Td*H2)
     expHT3 = expm(Td*H3)

     setlmis([]);                % define the system of LMI-s
% specifying the matrix variable Zq of the LMI ss.1
     Z1 = lmivar(1, [size(Y2, 1) 1]);
% specifying the matrix variable Zq of the LMI ss.2
     Z2 = lmivar(1, [size(Y2, 1) 1]);
% specifying the matrix variable Zq of the LMI ss.3
     Z3 = lmivar(1, [size(Y2, 1) 1]);
        % constructing the system of the LMI-s
     % for subsystem 1
     lmiterm([1, 1, 1, Z1], H1, 1, 's');  % LMI #1: Hq*Zq + Zq*Hq'
     lmiterm([1, 1, 1, 0], Q1);           % LMI #1: Qq
        lmiterm([2, 1, 1, Z2], expHT1, expHT1');  % LMI #2: expHTq*Zj*expHTq'
     lmiterm([2, 1, 1, Z3], expHT1, expHT1');  % LMI #2: expHTq*Zj*expHTq'
        lmiterm([2, 1, 1, Z1], -1, 1);         % LMI #2: -Zj
     lmiterm([2, 1, 1, 0], Y1);           % LMI #2: Yq
        % for subsystem 2
     lmiterm([3, 1, 1, Z2], H2, 1, 's');  % LMI #3: Hq*Zq + Zq*Hq'
     lmiterm([3, 1, 1, 0], Q2);           % LMI #3: Qq
     lmiterm([4, 1, 1, Z1], expHT2, expHT2');  % LMI #4: expHTq*Zj*expHTq'
     lmiterm([4, 1, 1, Z3], expHT2, expHT2');  % LMI #4: expHTq*Zj*expHTq'
     lmiterm([4, 1, 1, Z2], -1, 1);       % LMI #4: -Zj
     lmiterm([4, 1, 1, 0], Y2);           % LMI #4: Yq
        % for subsystem 2
     lmiterm([5, 1, 1, Z3], H3, 1, 's');  % LMI #5: Hq*Zq + Zq*Hq'
     lmiterm([5, 1, 1, 0], Q3);           % LMI #5: Qq
     lmiterm([6, 1, 1, Z1], expHT3, expHT3');  % LMI #6: expHTq*Zj*expHTq'
     lmiterm([6, 1, 1, Z2], expHT3, expHT3');  % LMI #6: expHTq*Zj*expHTq'
     lmiterm([6, 1, 1, Z3], -1, 1);       % LMI #6: -Zj
     lmiterm([6, 1, 1, 0], Y3);           % LMI #6: Yq

     % positivness of Zq
     lmiterm([-7, 1, 1, Z1], 1, 1);       % LMI #7: Z1>0
     lmiterm([-8, 1, 1, Z2], 1, 1);       % LMI #8: Z2>0
     lmiterm([-9, 1, 1, Z3], 1, 1);       % LMI #9: Z3>0

     lmis = getlmis;           % obtaining the system of LMI

        [tmin,xfeas] = feasp(lmis)  % calling function of feasibiliy.

% the solution Zq of ss. 1 corresponding to the
     Zs1 = dec2mat(lmis,xfeas,Z1)
% the solution Zq ss. 2 corresponding to the feasible
     Zs2 = dec2mat(lmis,xfeas,Z2)
% the solution Zq ss. 3 corresponding to the feasible
     Zs3 = dec2mat(lmis,xfeas,Z3)
```

```
% decision vector xfeas since tmin < 0
% checking constraints of feasibility. That is that if % tmin < 0.
        if tmin >= 0

          a = Td;              % the minimum is changed to the Td
          else
          b = Td;              % iteration is continued the minimum
                                    % is changed to the Td
          Tdm = b;             % saving value of Td
          end             % the iteration is continued

end                       % the iteration is finished

Tdmin = Tdm                    % the minimum dwell-time
```

The obtained values of the feasible solutions for Z_q denoted by Z_{s1}, Z_{s2}, Z_{s3} are shown below. The corresponding eigenvalues $\mathrm{eig}(\mathrm{Z_q})$ denoted by $\mathrm{eig}(\mathrm{Z_{s1}})$, $\mathrm{eig}(\mathrm{Z_{s2}}), \mathrm{eig}(\mathrm{Z_{s3}})$ are shown in the row 10 of the Table 1. The positive eigenvalues prove the positive definiteness of $\mathrm{Z_q}$ and the feasibility as well. Note that in the rows 6, 8 and 11 we did not get a feasible solution, because the scalar t_{min} returned with a positive value, which means that the associated Z_q pencil contains eigenvalues on or very near to the imaginary axis. Of course, this resulted in infeasibility. In such cases, according to the algorithm of interval halving, in these steps the upper - and lower bounds of an interval changed, to ensure a proper distance between the eigenvalues and the imaginary axis.

$$Zs_1 = 10^4 * \begin{bmatrix} 0.4073 & 0.0542 & 0.5203 \\ 0.0542 & 0.0518 & 0.0806 \\ 0.5203 & 0.0806 & 1.8642 \end{bmatrix}, Zs_2 = 10^4 * \begin{bmatrix} 0.5122 & 0.0832 & 0.6322 \\ 0.0832 & 0.0573 & 0.1137 \\ 0.6322 & 0.1137 & 1.8605 \end{bmatrix},$$

$$Zs_3 = 10^4 * \begin{bmatrix} 0.6249 & 0.1250 & 0.7722 \\ 0.1250 & 0.0691 & 0.1679 \\ 0.7722 & 0.1679 & 1.8750 \end{bmatrix}.$$

Table 1: Values obtained for T_d , t_m and Z_q during the iteration

i	T_d	t_{min}	eig. (Zs_1)	eig.(Zs_2)	eig. (Zs_3)
1	1	-0.0854	25.9263 83.4787 246.1332	25.2094 65.1757 251.7411	24.3039 54.9130 252.8941
2	0.5000	-0.0849	25.9233 83.4782 246.1332	25.2067 65.1749 251.7411	24.3024 54.9119 252.8940
3	0.2500	-0.0529	25.7108 83.4469 246.1329	25.0205 65.1218 251.7407	24.1738 54.8301 252.8934
4	0.1250	-0.4185	63.3241 100.7714 246.4629	52.9620 95.4908 252.1020	44.0826 99.2161 253.4098
5	0.0625	-0.5410	68.0075 111.0561 246.6869	54.9368 107.9080 252.3338	45.0798 111.5681 253.6986
6	0.0313	0.0146	$10^4 *$ 0.0000 0.0219 3.5277	$10^4 *$ 0.0000 0.0115 3.0118	$10^4 *$ 0.0000 0.0073 2.1199
7	0.0469	-0.7438	65.5712 108.0716 246.8438	53.7206 104.2321 252.4862	44.1185 107.6819 253.8765
8	0.0391	0.0128	$10^4 *$ 0.0000 0.0177 3.8775	$10^4 *$ 0.0000 0.0098 3.0578	$10^4 *$ 0.0000 0.0114 3.2479
9	0.0430	-0.6534	63.1824 105.1323 246.8925	52.4740 100.2198 252.5290	43.2152 103.1864 253.9147
10	0.0410	-9.7372	$10^4 *$ 0.0437 0.2442 2.0353	$10^4 *$ 0.0426 0.2679 2.1196	$10^4 *$ 0.0423 0.2640 2.2627
11	0.0400	0.0126	$10^4 *$ 0.0000 0.0243 6.2884	$10^4 *$ 0.0000 0.0128 5.7631	$10^4 *$ 0.0000 0.0040 1.0035

The iteration ran till the new interval assigned by *b-a* reached the pre-specified relative accuracy of the solution $eps = 0.001$. Performed the T_d- iteration and repeated it 10-times the $T_{dmin} = 0.0410$s is obtained. The computational cost is primarily dependant on solving the q independent LMIs

plus the iteration. We have seen, that despite of the multivariable time dependent optimization problem, combination of an algorithm interval halving with an LMI solver to determine the common minimum dwell time could be efficiently applied. A variation of the measures for the feasibility t_{min} during the iteration is shown in *Fig. 1*.

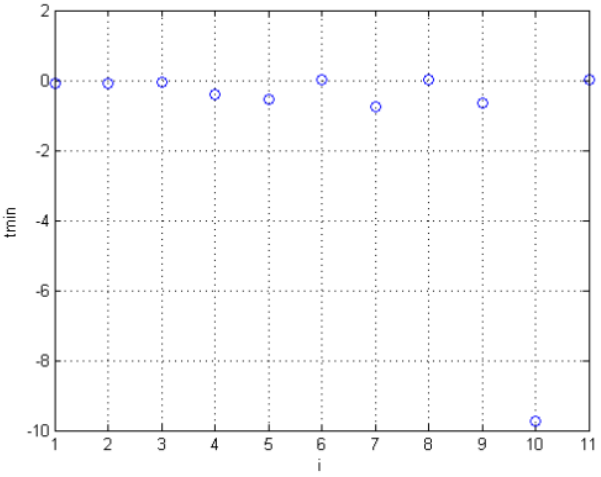


Figure 1: The variation of t_{min} during the iteration

An approximative calculation of the minimum dwell time based on the L_2-norm of the state estimator system

For purpose of comparison we applied the calculation of the minimum dwell time based on the L_2-norm of the state estimator system on the example. It is to note, that as it was introduced in the Chapter 3, this approach may lead to a conservative result.
The similarity transformation (16) has been computed by the MATLAB function $[T, D] = eig\ (H)$, see in [25], which returned the matrices T and D.
From the calculation in (15), the worst-case condition for the minimum dwell time is given by $\tau_{D*} > 0.055$ sec. The corresponding parameters in (18) were $a = 0.467$ and $\lambda = -8.474$.

5. Conclusion

This paper was concerned with a numerical algorithm for determining the minimum dwell time constraint for switched linear $\mathcal{H}_\infty$ fault detection filters.

Despite the multivariable time dependent optimization problem, by means of the T_d-iteration the common minimum dwell time assuring each specified $\mathcal{H}_\infty$ level of each single filter could be determined. The case study implemented in MATLAB resulted in positive definite solutions for Z_q and also in the corresponding minimum dwell time $T_{dmin} = 0.0410$ s. The results indicate that the frequency of changing the switching signal sequence should be lower than 24.39 Hz to ensure the robust stability of the state estimation error between the switching instants. Additionally, the dwell time was approximately calculated based on the L_2-norm of H_q. As it was shown, smaller value could be reached using the T_d-iteration, than from the approach based on the L_2-norm of H_q. Of course, the latter is only a very conservative approach. On the other hand the $\boldsymbol{T_d}$-iteration has to face with successive numerical computation of the quadratic matrix inequalities resulted in a proportional computation cost. We have found the solution after running the code in MATLAB after 0.5 second CPU time on a PC with Intel® Celeron® CPU B815 (1.60 GHz).

We think, that the technique of the T_d-iteration offers further benefits from the point of view the designer. Apart from the advantage that a variety of design specifications and constraints can be expressed through LMI-s, we assume, due to the combination with the interval halving algorithm, it gives more flexibility to examine the solution during the entire design process. For example, it is easy to analyse the impact of the T_d value on the number of iteration steps or to analyse the impact of the variation of the relative accuracy of the solution. One can easily perform experiments and get answers e.g. to the following questions: How does the iteration converge? How do the eigenvalues of the decision variable change? How close are they to the imaginary axis? Issues with such explicit conditions can be easily examined, step by step during the iterations, which can also be useful for better understanding the nature of switched systems.

References

[1] Paxman, J., “Switching Controllers: Realization, Initialization and Stability”, Ph.D Dissertation, University of Cambridge, 2003.

[2] DeCarlo, R. A., Branicky, M.S., Pettersson, S. and Lennartson, B., “Perspectives and Results on the Stability and Stabilizability of Hybrid Systems”, in *Proc. IEEE*, Vol. 88, pp. 1069–1082., 2000.

[3] Liberzon, D., Morse, A. S., “Basic Problems in Stability and Design of Switched Systems”, IEEE Contr. Syst., pp. 59–70., 1999.

[4] Hespanha, J. P., Morse, A.S., “Stability of switched systems with average dwell-time”, *Proc. 38th Conf. Decision and Control*, 1999, pp. 2655–2660.

[5] Alessandri, A., Coletta, P., “Switching observers for continuous-time and discrete-time linear systems”, in *Proceedings of the American Control Conference Arlington*, VA June 2001.

[6] Colaneri, P., "Analysis and control of linear switched systems", Lecture notes, Politecnico Di Milano, 2009.
[7] Geromel, J. C., Colaneri, P., "H_∞ and Dwell Time Specifications of Switched Linear Systems", *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Dec. 9–11.2008.
[8] Chen, W., Saif, M., "Observer design for linear switched control systems", in *American Control Conference Proceedings of the 2004*, 0-7803–8335-4, Boston, 2004.
[9] Colaneri, P., Geromel, J.C., and Astolfi, A., "Stabilization of continuous-time switched nonlinear systems", Systems Control Letter, 57-1, pp. 95–103, 2008.
[10] Geromel, J. C., Colaneri, P., "Stability and stabilization of continuous-time switched linear systems", SIAM J. on Contr. Optim., 45-5, pp.1915–1930, 2006.
[11] Karabacak, O., and Sengör, N. S., "A dwell time approach to the stability of switched linear systems based on the distance between eigenvector sets", Int. J. System Science, 40(8), pp. 845–853, 2009.
[12] Edelmayer, A., "Fault detection in dynamic systems: From state estimation to direct input reconstruction", Universitas-Győr Nonprofit Kft., Győr, 2012.
[13] Edelmayer, A., Bokor, J., Keviczky, L., "An H_∞ Filtering Approach to Robust Detection of Failures in Dynamical Systems", in *Proc. 33th Annual Decision and Control*, Conf., Buena Vista, USA, 1994, pp. 3037–3039.
[14] Edelmayer, A., Bokor, J., Keviczky, L., "An H_∞ Filter Design for Linear Systems: Comparison of two Approaches", IFAC 13th Triennial World Congress, San Francisco, USA, 1996.
[15] Horváth, Zs., Edelmayer, A., "Robust Model-Based Detection of Faults in the Air Path of Diesel Engines", Acta Universitatis Sapientiae Electrical and Mechanical Engineering, Vol. 7 pp. 5–22, 2015.
[16] Horváth, Zs., Edelmayer, A., "Solving of the Modified Filter Algebraic Riccati Equation for H-infinity fault detection filtering", Acta Universitatis Sapientiae Electrical and Mechanical Engineering, Vol. 9, pp. 57–77, 2017.
[17] Bokor, J., Gáspár, P., Szabó, Z., "Robust Control Theory", Typotex, Budapest, 2013.
[18] Boyd, S., Ghaoui, L. E., Feron, E. and Balakrishnan, V., "Linear Matrix Inequalities in System and Control Theory", SIAM, Philadelphia, 1994.
[19] Nesterow, Y., Nemirovski, A., "Interior Point Polynomial Methods in Convex Programming: Theory and Applications", SIAM, Philadelphia, 1994.
[20] Gahinet, P, Apkarian, P., "A linear matrix inequality approach to H_∞ control", International Journal of Robust and Nonlinear Control, Vol.4, pp.421–448, 1994.
[21] Duan, G., Yu, R., H., H., "LMIs in Control Systems: Analysis, Design and Applications", CRC Press, Boca Raton, 2013.
[22] https://www.mathworks.com/help/robust/ref/feasp.html
[23] Gahinet, P., Nemirovski, A., Laub, A. J. and Chilali, M. "LMI Control Toolbox for Use with Matlab", The MathWorks Inc., Natick, 1995.
[24] http://home.deib.polimi.it/prandini/file/2015_06_16%20hybrid%20systems_2.pdf
[25] https://www.mathworks.com/help/robust/ref/eig.html

Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 10 (2018) 20-41

DOI: 10.2478/auseme-2018-0002

Model Predictive Control of a Differential-Drive Mobile Robot

Samir BOUZOUALEGH[1], El-Hadi GUECHI[1] and Ridha KELAIAIA[2]

[1] Laboratoire d'Automatique de Skikda (LAS), Faculté de Technologie, Département de Génie Électrique, Université 20 Août 1955-Skikda, BP 26, Route El-Hadaeik, Skikda 21000, Algeria, e-mail: samir.bouzoualegh@gmail.com, e.guechi@univ-skikda.dz
[2] Faculty of Technology, Université de 20 Août 1955-Skikda, e-mail: r.kelaiaia@univ-skikda.dz

Manuscript received September 22, 2018; revised December 10, 2018.

Abstract: This paper presents a model predictive control (MPC) for a differential-drive mobile robot (DDMR) based on the dynamic model. The robot's mathematical model is nonlinear, which is why an input–output linearization technique is used, and, based on the obtained linear model, an MPC was developed. The predictive control law gains were acquired by minimizing a quadratic criterion. In addition, to enable better tuning of the obtained predictive controller gains, torques and settling time graphs were used. To show the efficiency of the proposed approach, some simulation results are provided.

Keywords: mobile base; dynamic model; nonlinear control; model predictive control; input-output linearization.

1. Related works

In general, the control of mobile robots poses a very strong challenge among the robot control community. Today, several studies are still ongoing, especially in aeronautical space exploration, the inspection of nuclear power plants, and automated agriculture, motivating the increasing interest in mobile robotics [1]. Such applications need to find an adequate control law for mobile robots after taking into consideration all the possible constraints. For control purposes, many approaches are available in the literature; for example, Guechi et al. [2] developed predictive control dynamics for a two-link manipulator robot. The idea consists of linearizing the nonlinear dynamic model of the robot through feedback linearization, and, based on the obtained linear model, an MPC was developed by assuming the outputs of the predictive controller as constant in the prediction horizon interval. A similar idea was exploited by Belda et al. [3] but with a different model presentation; as they designed a model predictive control (MPC) for an autonomous mobile robot system with a 5-DOF manipulator arm.

Based on the manipulator robot dynamic model, the nonlinear state space model is converted into a linear model by considering the time-varying matrices as constants within one prediction horizon. Another application of [3] was intended for mobile platforms; Kamel et al. [4] applied a combination of linear model predictive control and input–output feedback linearization for a team of wheeled mobile robots to accomplish a formation task. The time-variant matrix of the state space model was considered as a constant matrix between the prediction intervals. Guechi et al. [5] did a comparative study between a MPC and Linear Quadratic (LQ) optimal control of a two-link robot arm, and the simulation results showed that the proposed MPC gave a better system performance then the LQ optimal control approach. Elkhateeb [6] developed a novel tuning methodology of a PID controller for trajectory tracking of a manipulator robot. The optimal gains of the PID controller are obtained by using a dynamic inertia weight artificial bee colony optimization algorithm. Mendili et al. [7], [8] presented two papers, the first of which was a predictive controller of a holonomic omnidirectional mobile robot. The predictive control law was obtained by using a state space model, which is based on the robot dynamic model. The simulation results illustrated the effectiveness of the control law joined with the dynamic model. Meanwhile, the second paper was an application of the model predictive control to solve the problem of the trajectory tracking and path following of an omnidirectional mobile robot. Two different models were considered: the kinematic and the dynamic model. The results showed the effectiveness of the dynamic model in comparison with the kinematic one in tracking the trajectory and path without posture. To achieve better path tracking for a wheeled mobile robot (WMR), Maniatopoulos et al. [9] developed an MPC-based solution to the problem of navigating a non-holonomic mobile robot while maintaining visibility. The proposed approach combines the convergence properties of a dipolar vector field along with a constrained nonlinear MPC formulation using recentered barrier functions, which take into account the visibility constraints and the saturation of control inputs. The control strategy falls into the class of dual-mode MPC schemes. That is, the system trajectories are forced by the model predictive controller into a suitably defined terminal region containing the goal configuration. In this region, the trajectories resulting from tracking the dipolar vector field by construction do not violate the visibility constraints. To achieve better path tracking for a wheeled mobile robot (WMR), Sinaeefar et al. [10] developed an adaptive fuzzy nonlinear model predictive control (NMPC). The proposed controller solves the integrated kinematic and dynamic tracking problem in the presence of both parametric and non-parametric uncertainties. Furthermore, a fuzzy system, the parameters of which are updated online by a gradient descent algorithm, is employed. While this fuzzy system can provide an appropriate

model for the robot, it can also deal with any changes in robot parameters. Mazur [11] presented a general solution to the path-following problem for mobile manipulators with a non-holonomic mobile platform. New proposed control algorithms – for mobile manipulators with fully known dynamics or with parametric uncertainty in the dynamics – take into consideration the kinematics as well as the dynamics of the non-holonomic mobile manipulator. The convergence of the control algorithms was proved using LaSalle's invariance principle. Ostafew et al. [12] developed a learning-based nonlinear model predictive control algorithm for a path-repeating mobile robot negotiating large-scale, GPS-denied outdoor environments. The disturbance is modelled as a Gaussian process based on observed disturbances as a function of relevant variables, such as the system state and input. Localization for the controller is provided by an on-board visual teach and repeat mapping and navigation system. Two experiments on two significantly different robots demonstrated the system's ability to handle unmodelled terrain and robot dynamics and a speed scheduler based on previous experience to address the classic exploration vs. exploitation trade-off, balancing speed and path-tracking errors. Mitrovic et al. [13] presented a new methodology for the avoidance of one or more obstacles for the navigation of a differential-drive mobile robot. The approach is based on fuzzy logic with virtual fuzzy magnets and represents a reactive controller for navigation through an unknown environment. The relative parameters of the obstacle in the robot's way are determined at the preprocessing stage, and the algorithm is therefore applicable to obstacles of different sizes. The algorithm, designed to avoid a single stationary obstacle, was generalized and successfully applied in a multiple-obstacle navigation scenario. The efficiency of the algorithm was illustrated by computer simulations using the kinematic model of a mobile robot.

This paper proposes a novel approach to controlling a differential-drive mobile robot. The control strategy of this approach is to linearize the DDMR dynamic model by using an input–output linearization control in the first step; then, the second step is to develop an MPC for the obtained linear model by minimizing a quadratic criterion.

This paper is organized as follows. In the second section, we provide a description of the DDMR and its different models. In the third section, the control strategy for the DDMR from an initial position up to a final position using a model predictive control (MPC) is presented. The simulation results are presented in the fourth section.

2. Differential drive mobile robot models

A differential drive mobile robot can be presented as depicted in *Fig.1.*

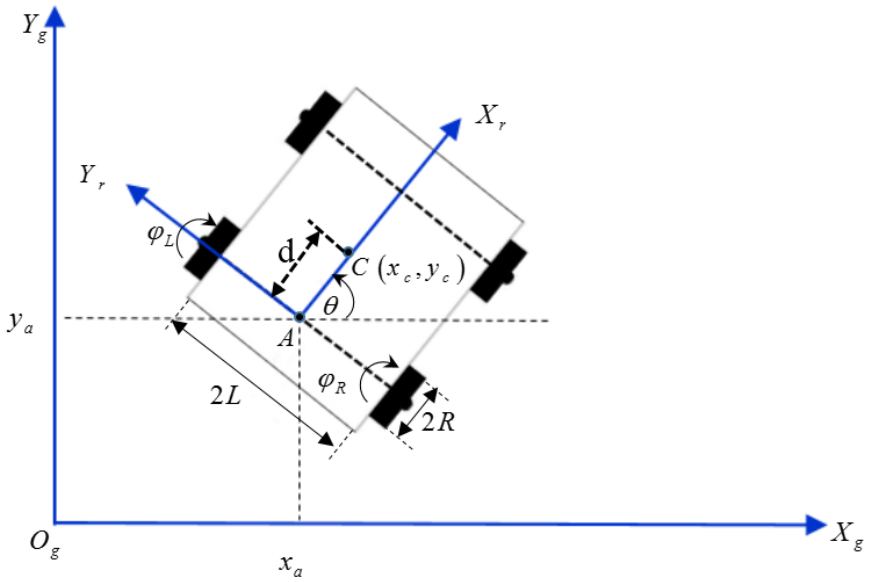


Figure 1: Schematic representation of a DDMR robot.

A. Coordinate systems

To describe the mobile robot's position, we have defined two coordinate systems:

A.1. Global coordinate system: This coordinate system is a global frame which is fixed in the environment in which the DDMR moves in. and is denoted as $\{X_g, Y_g\}$.

A.2. Robot coordinate system: This coordinate system is a local frame attached to the DDMR, and thus, moving with it. And is denoted as $\{X_r, Y_r\}$.

The two defined frames are shown in *Fig.1* the origin of the robot frame is defined to be the mid-point A on the axis between the wheels. The center of mass C of the robot is assumed to be on the axis of symmetry, at a distance d from the origin A.

The mobile robot position and orientation in the global Frame can be defined as:

$$q^g = \left| x_a \quad y_a \quad \theta \right|^T \tag{1}$$

B. Kinematic constraints of DDMR

The DDMR motion is characterized by the non-holonomic constraints equations, which are based on the following assumptions [14]:

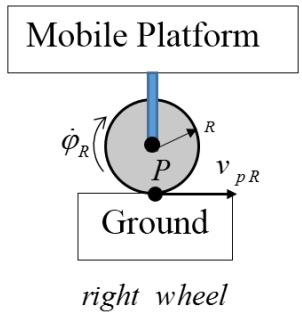


Figure 2: Rolling motion constraints.

- *No lateral slip*:

$$-x_a \cos\theta + y_a \sin\theta = 0 \tag{2}$$

- *Pure rolling*:

$$\begin{cases} \dot{x}_{PR} \cos\theta + \dot{y}_{PR} \sin\theta = R\dot{\phi}_R \\ \dot{x}_{PL} \cos\theta + \dot{y}_{PL} \sin\theta = R\dot{\phi}_L \end{cases} \tag{3}$$

$$\begin{cases} \dot{x}_{PR} = \dot{x}_{PL} = \dot{x}_a + L\dot{\theta}\cos\theta \\ \dot{y}_{PR} = \dot{y}_{PL} = \dot{y}_a + L\dot{\theta}\sin\theta \end{cases} \tag{4}$$

where: $\dot{\phi}_R$ and $\dot{\phi}_L$ are the right and left wheels angular velocities respectively, R is the wheels' radius, L is the distance between the driving wheels and the axis of symmetry and d is the distance between the points A and C.

Taking (4) into account, (3)becomes the following:

$$\begin{cases} \dot{x}_a \cos\theta + \dot{y}_a \sin\theta + L\dot{\theta} - R\dot{\phi}_R = 0 \\ \dot{x}_a \cos\theta + \dot{y}_a \sin\theta - L\dot{\theta} - R\dot{\phi}_L = 0 \end{cases} \tag{5}$$

The constraint equations (2) and (4) can be presented as follow:

$$\Lambda(q)\dot{q} = 0 \tag{6}$$

where:
$$\Lambda(q) = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 & 0 \\ \cos\theta & \sin\theta & L & -R & 0 \\ \cos\theta & \sin\theta & -L & 0 & -R \end{bmatrix} \tag{7}$$

$$\dot{q} = \begin{bmatrix} \dot{x}_a & \dot{y}_a & \dot{\theta} & \dot{\phi}_R & \dot{\phi}_L \end{bmatrix}^T \tag{8}$$

C. Kinematic model

Kinematic modelling is the study of the motion of mechanical systems without considering the forces that affect the motion.

We can calculate the linear and angular velocities for the driving wheels in the robot frame as below:

$$v = \frac{v_{pR} + v_{pL}}{2}, w = \frac{v_{pR} - v_{pL}}{2L} \tag{9}$$

where v_{pR} and v_{pL} Are the right and left linear velocities of the contact point P.

Therefore, the platform centre A velocities in the robot and global frames are as follows:

$$\dot{q}^r = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ w \end{bmatrix} = \frac{R}{2} \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix}, \tag{10}$$

$$\dot{q}^g = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ w \end{bmatrix} = \frac{R}{2} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \tag{11}$$

D. Dynamic model

Dynamics is the study of mechanical system motion taking into consideration the different forces that affect it. The dynamic model of the DDMR is essential for simulation analysis of its motion and for the design of various motion control algorithms using the Lagrange formula:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right)+\frac{\partial L}{\partial q_i}=F-\Lambda^T\left(q\right)\lambda \tag{12}$$

where L=T-V, T is the kinematic energy, V is the potential energy of the mobile platform, q_i are the generalized coordinates, F is the generalized force vector, Λ is the constraint matrix, and λ is the Lagrange multiplier vector associated with the constraints. In addition, knowing the DDMR kinetic energy, which is the sum of T_c, the kinetic energy of the robot platform without wheels, plus T_{wR},T_{wL}, the kinetic energy of the wheels, note that their formulas are as follows[14]:

$$\begin{cases} T_c=\dfrac{1}{2}m_c v_M^2+\dfrac{1}{2}I_c\dot{\theta}^2 \\ T_{wR}=\dfrac{1}{2}m_w v_{wR}^2+\dfrac{1}{2}I_m\dot{\theta}^2+\dfrac{1}{2}I_w\dot{\phi}_R^2 \\ T_{wL}=\dfrac{1}{2}m_w v_{wL}^2+\dfrac{1}{2}I_m\dot{\theta}^2+\dfrac{1}{2}I_w\dot{\phi}_L^2 \end{cases} \tag{13}$$

where m_c and I_c are the mass and the moment of inertia of the platform without the driving wheels, respectively, m_w and I_w are the mass and the moment of inertia of each driving wheel plus the rotor of its motor.

The dynamic model of the non-holonomic DDMR with n generalized coordinates $(q_1,q_2,...,q_n)$ and subject to *m* constraints can be described by the following equation of motion [15]:

$$M\left(q\right)\ddot{q}+V\left(q,\dot{q}\right)\dot{q}=B(q)\tau-\Lambda^T\left(q\right)\lambda \tag{14}$$

$M\left(q\right)$: the inertia moment matrix, symmetric positive definite matrix.

$V\left(q,\dot{q}\right)$: the Coriolis and centrifugal matrix.

$B(q)$: input matrix.

τ : input vector.

In addition:

$$M(q)=\begin{bmatrix} m_T & 0 & -m_T d\sin\theta & 0 & 0 \\ 0 & m_T & m_T d\cos\theta & 0 & 0 \\ -m_T d\sin\theta & m_T d\cos\theta & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \tag{15}$$

$$V(q,\dot{q})=\begin{bmatrix} 0 & -m_T d\dot{\theta}\cos\theta & 0 & 0 & 0 \\ 0 & -m_T d\dot{\theta}\sin\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

$$B(q)=\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}^T \tag{17}$$

Note that:

$$m_T = m_c + 2m_w, \quad I = I_c + m_c d^2 + 2m_w L^2 + 2I_m \tag{18}$$

For the purpose of control and simulation and because the Lagrange multipliers λ_i are unknown, it is more convenient to eliminate the constraint term $\Lambda(q)^T\lambda$ in (14). Furthermore, since the constrained velocity is always in the null space of $\Lambda(q)$, it is possible to define $(n-m=2)$ velocities $\eta(t)=[\eta_1\ \eta_2\ ]$, such that [15]:

$$\dot{q} = S(q)\eta(t) \tag{19}$$

We can verify that $S(q)$ is also the null space of the constraint matrix $\Lambda(q)$, which means: $S(q)\Lambda(q)=0$; then, it is easy to verify that:

$$S(q)=\begin{bmatrix} \frac{R}{2L}(L\cos\theta - d\sin\theta) & \frac{R}{2L}(L\cos\theta + d\sin\theta) \\ \frac{R}{2L}(L\sin\theta + d\cos\theta) & \frac{R}{2L}(L\sin\theta - d\cos\theta) \\ \frac{R}{2L} & -\frac{R}{2L} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{20}$$

Therefore based on $S(q)$ matrix choice, and the state variable $q=\begin{bmatrix} x_a & y_a & \theta & \phi_R & \phi_L \end{bmatrix}^T$ we have $\eta(t)=[\dot{\phi}_R \ \dot{\phi}_L\ ]$. By differentiating (19) and substituting the expression of $\ddot{q}$ into (14) and multiplying it by S^T, the dynamic model equation is as follow [14]:

$$\bar{M}\,\dot{\eta}+\bar{V}=\bar{B}\,\tau=\tau \tag{21}$$

Thus:

$$\begin{cases} \bar{M}=S^T(q)M(q)S(q) \\ \bar{V}=S^T(q)\left(M(q)\dot{S}(q)+V(q,\dot{q})S(q)\right) \\ \bar{B}=S^T(q)B(q)=\mathrm{I}_{2\times 2} \end{cases} \tag{22}$$

(21) shows that the DDMR dynamic model is a function only of the right and left wheel angular velocities $(\dot{\phi}_R,\dot{\phi}_L)$, the robot angular velocity $\dot{\theta}$ and the driving motor torques (τ_R,τ_L).

3. Control algorithm

In this part, a predictive control law for a differential-drive mobile robot is developed. First, we consider the nonlinear dynamic model given by (21) Then, we convert the nonlinear dynamic model into a completely linear model on which the linear control approach is applied. Once the linear model has been obtained, a model predictive control will be designed in the second step [15].

3.A Input-output linearization

Let consider the state space vector below:

$$x=[q^T\ \eta^T\ ]^T=[x_c,y_c,\theta,\phi_R,\phi_L,\dot{\phi}_R,\dot{\phi}_R\ ]^T \tag{23}$$

We can write the state space dynamic model as follows:

$$\dot{x}=\begin{bmatrix} S\eta \\ 0 \end{bmatrix}+\begin{bmatrix} 0 \\ \bar{M}^{-1}\left(\tau-\bar{V}\right) \end{bmatrix}=\begin{bmatrix} S\eta \\ 0 \end{bmatrix}+\begin{bmatrix} 0 \\ I_{2x2} \end{bmatrix}\bar{M}^{-1}\left(\tau-\bar{V}\right) \tag{24}$$

We can rewrite the above state space equation as follow:

$$\dot{x}=f(x)+g(x)u \tag{25}$$

where:

$$\begin{cases} u = \bar{M}^{-1}\left(\tau - \bar{V}\right) \\ f\left(x\right) = \begin{bmatrix} S\eta \\ 0 \end{bmatrix} \\ g\left(x\right) = \begin{bmatrix} 0 \\ I_{2\times 2} \end{bmatrix} \end{cases} \tag{26}$$

From (25), we could assume a new input u which could linearize(24), but the challenge now is to find another equation that links the new input u with the robot's position in such a way that we can compute the wheels' torques based on this new input.

Let us suppose $y = \Upsilon(q) = \begin{bmatrix} x_c & y_c \end{bmatrix}^T$ is the output vector that we need to control, as shown in *Fig. 2.*

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x_a \\ y_a \end{bmatrix} + \begin{bmatrix} x_c^a & -y_c^a \\ x_c^a & y_c^a \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{27}$$

x_c^a, y_c^a are the coordinates of point C in the robot frame. Therefore:

$$\dot{y} = \partial\Upsilon/\partial t = \left(\partial\Upsilon/\partial q\right)\dot{q} = J_f\left[S\left(q\right)\eta(t)\right] = \left[J_f S\left(q\right)\right]\eta(t) = \Gamma(q)\eta(t) \tag{28}$$

Γ : 2x2 decoupling matrix, such as $\Gamma = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix}$,

where:

$$\begin{bmatrix} \Gamma_{11} \\ \Gamma_{12} \\ \Gamma_{21} \\ \Gamma_{22} \end{bmatrix} = \frac{R}{2L} \begin{bmatrix} \mathrm{L} - y_c^a & -d - x_c^a \\ \mathrm{L} + y_c^a & d + x_c^a \\ \mathrm{L} - y_c^a & d + x_c^a \\ \mathrm{L} + y_c^a & -d - x_c^a \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{29}$$

The second derivate of (28), is obtained as follows:

$$\ddot{y} = \dot{\Gamma}\eta + \Gamma\dot{\eta} \tag{30}$$

As developed in [15], to make Eq. (30) linear we have to do the following substitution:

$$\begin{cases} \ddot{y} = v \\ \dot{\eta} = u \end{cases} \tag{31}$$

which is the I/O feedback linearisation, therefore the second input u will be:

$$v = \dot{\Gamma}\eta + \Gamma u \Rightarrow u = \Gamma^{-1}(v - \dot{\Gamma}\eta) \tag{32}$$

By substituting the second input u of the Eq.(32) into Eq. (21), we can compute the DDMR wheels' torques as follows:

$$\tau = \bar{M}u + \bar{\mathrm{V}} \tag{33}$$

Now we need to find out the variable v , note that:

$$\ddot{y} = \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{34}$$

where: $v = [v_1, v_2]$ is the synthetic control vector. This will be the subject of the following subsection.

3.B Model Predictive Control law

Following the I/O feedback linearization, let us apply the model predictive control to compute the first input v_1, where v_i is the synthetic control vector. Now let us develop the predictive control law for the x_c coordinate, which will be similar to the y_c coordinate.

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = v_1(t) \\ z(t) = x_1(t) \end{cases} \tag{35}$$

where: $[x_1 \quad x_2]^{\mathrm{T}} = [x_c \quad \dot{x}_c]^{\mathrm{T}}$, v_1 is the synthetic control of the x_c variable, while $z(t) = x_c(t)$ is the output signal.

Let consider that $v(t) = v_1$ is constant in the time interval [t t + h] [16-18] where h is the prediction horizon time, and by using the Eq.(35), we can formulate the prediction model as follows:

$$x_c(t+h) = \frac{1}{2} v_1 h^2 + \dot{x}_c(t) h + x_c(t) \tag{36}$$

We want to minimize not only the articulation deviation error but also the energy needed by the manipulator arm to reach the final position, which is why we choose the following cost function:

$$J = e_1^2(t+h) + \rho \int_t^{t+h} v^2 dt \tag{37}$$

where: $e_1(t+h) = x_{cd} - x_c(t+h)$ is the prediction error and x_{cd} is the desired value generated by the referenced trajectory. The horizon time *h* and the weight factor ρ are both positive parameters to be computed later.

By replacing the predicted value of $x_c(t+h)$ given by (36) into (37), therefore the criterion J become:

$$\begin{aligned} J = {} & x_{cd}^2 - x_{cd} v_1 h^2 - 2h x_{cd} \dot{x}_c(\mathrm{t}) - 2x_{cd} x_c(\mathrm{t}) + \frac{1}{4} v_1^2 h^4 + \\ & v_1 h^3 x_{cd} + v_1 h^2 x_c(\mathrm{t}) + \dot{x}_c^2(\mathrm{t}) h^2 + 2h\dot{x}_c(\mathrm{t}) x_c(\mathrm{t}) + x_c^2(\mathrm{t}) + \rho h v_1^2 \end{aligned} \tag{38}$$

Therefore: $\min\{J\}_{v_1}$ with subject to $h > 0$ and $\rho > 0$

$$\Rightarrow v_1(t) = k_1(x_{cd}(\mathrm{t}) - x_c(t)) - k_2 \dot{x}_c(t) \tag{39}$$

where the predictive control law gains are : $k_1 = \dfrac{2h}{h^3 + 4\rho}$ and $k_2 = \dfrac{2h^2}{h^3 + 4\rho}$.

The block diagram of the closed-loop system can be presented as shown in *Fig. 3.*

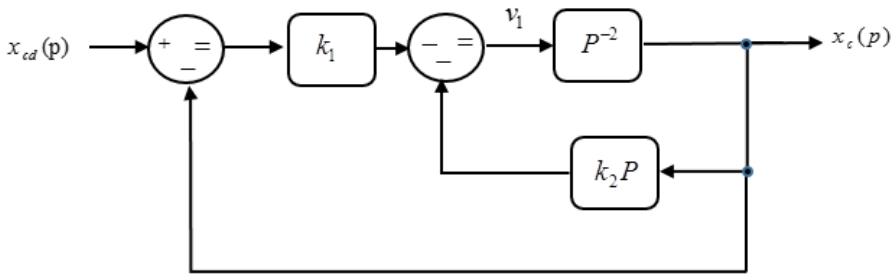


Figure 3: Closed-loop system.

Regarding the response of the equivalent system, we want it to adopt similar behavior to the second-order system: $\frac{w_0^2}{p^2 + 2\xi w_0 p + w_0^2}$, where ξ and w_0 are respectively the damping factor and the natural frequency.

The closed loop transfer function of the system shown in *Fig. 3* is given by:

$$\frac{x_c(p)}{x_{cd}(p)} = \frac{k_1}{p^2 + k_2 p + k_1} \tag{40}$$

we can verify that:

$$\begin{cases} 2\xi w_0 = \dfrac{2h^2}{h^3 + 4\rho} = hk \\ w_0^2 = \dfrac{2h}{h^3 + 4\rho} \end{cases} \tag{41}$$

Therefore:

$$h = \frac{2\xi}{w_0}, \quad \rho = \frac{\xi}{w_0^3}\left(1 - 2\xi^2\right) \tag{42}$$

The weight factor $\rho > 0$; therefore, from Eq.(42), as the damping factor ξ must be less than $1/\sqrt{2}$ to obtain good damping, we have to choose ξ as near as possible to $1/\sqrt{2}$; let us suppose $\xi = 0.999/\sqrt{2}$.

We want to find the h and ρ values in such a way that our system fulfills the following conditions:

$$\left|\tau_i\right| \le 150\,Nm\,, i = 1,2. \text{ and } \quad T_{r\pm 5\%} \le 1s \tag{43}$$

with:

$$T_{r\pm 5\%} = \frac{-\ln(0.05\sqrt{1-\xi^2})}{\xi w_0} \text{ [19]} \tag{44}$$

Hence, we calculate and draw the maximum and minimum torque values (for each driving wheel) and the settling time values for several values of w_0 (from 1 rad/s to 10 rad/s), with a Matlab script obtained from the graphs below.

From *Fig. 4*, to have the maximum wheel torque less than or equal to 150 Nm, the natural frequency w_0 should be less than or equal to 9 rad/s. From *Fig. 5*, to have the minimum wheel torque greater than or equal to 150 Nm, the natural frequency w_0 should be less than or equal to 9 rad/s. In addition, from

Fig. 6, to have a settling time $T_{r\pm5\%}$ less than or equal to 1 s, the natural frequency w_0 should be greater than or equal to 4.8 rad/s. Finally, the natural frequency w_0 should verify the following inequality:

$$4.8rd/s \le w_0 \le 9rd/s \tag{45}$$

Let us take $w_0 = 9$ rad/s, which means that $h = 0.157$ s, $\rho = 1.9.10^{-6}$ and the predictor control law gains values are $k_1 = 72.26$ and $k_2 = 12$.

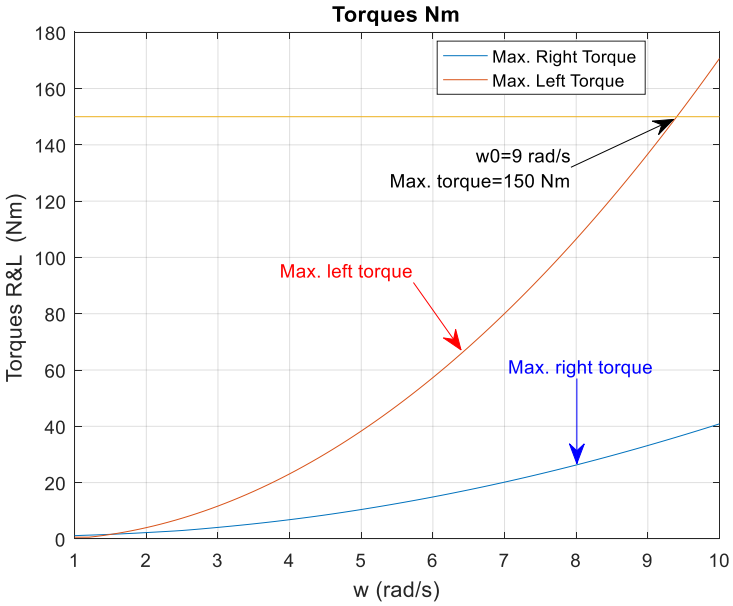


Figure 4: Max. torque $\tau_{i\max}$.

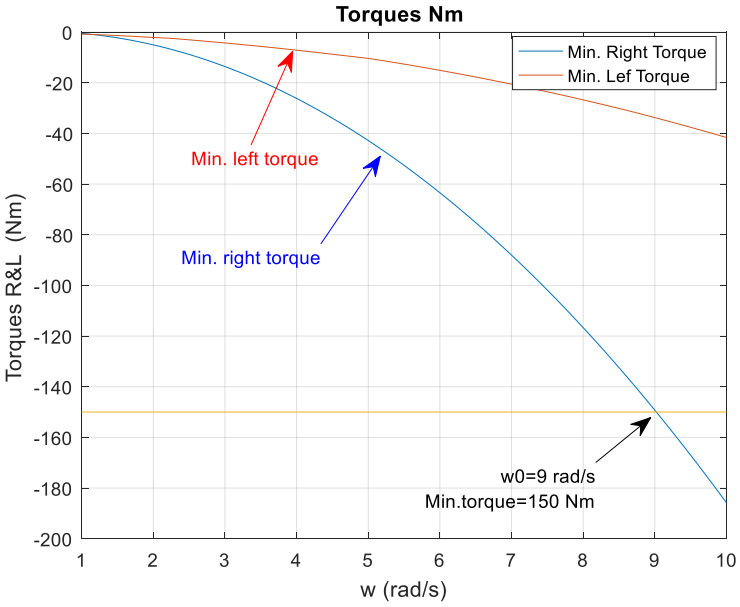


Figure 5: Max. torque $\tau_{i\,\min}$.

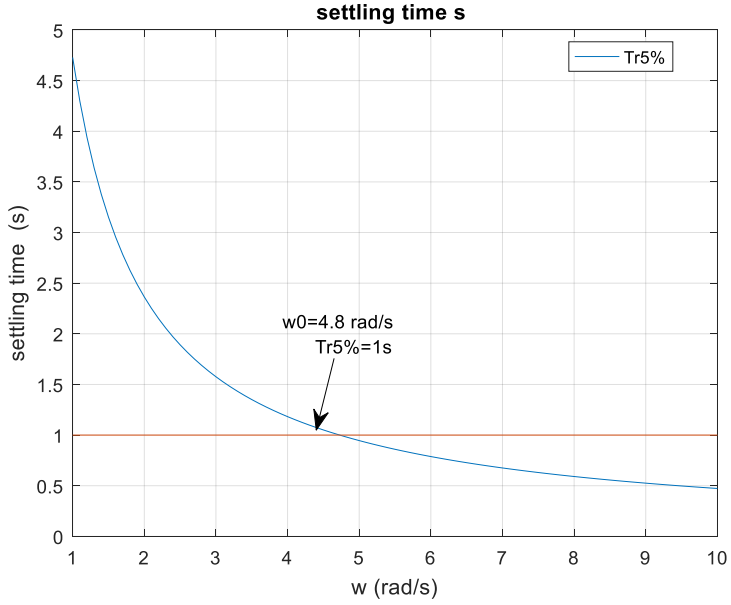


Figure 6: Max. torque $T_{r\pm5\%}$.

4. Simulation results

To show the suitability of our tinning parameters, k1 and k2, let us consider $m_c = 1(kg)$ and $I_c = 1(Nm^2)$ the mass and the moment of inertia of the platform without the driving wheels. $m_w = 0.1(kg)$ and $I_w = 0.1(Nm^2)$:the mass and the moment of inertia of each driving wheel plus the rotor of its motor. $I_m = 0.1(Nm^2)$: the moment of inertia of each wheel and the motor rotor.

For trajectory generation, we show that the DDMR moves from an initial point $P_{init} = (1,1)$ to a desired position $P_{dis} = (40,60)$ during a time period of 60 s, following a short distance, which will be a straight line; note that we suppose that there is no obstacle between P_{init} and P_{dis} .

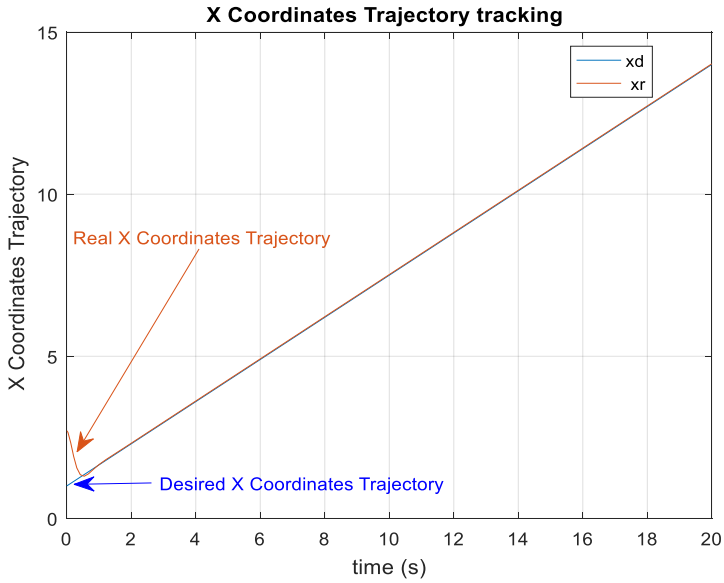


Figure 7: Desired and real Xc.

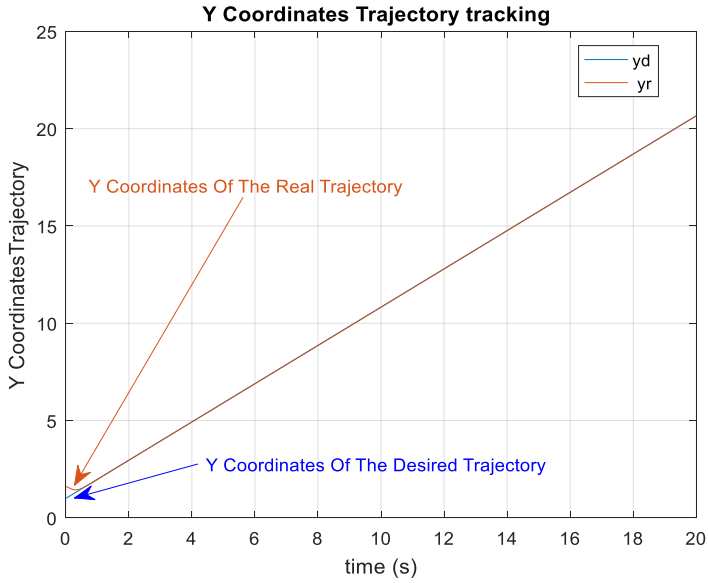


Figure 8: Desired and real Yc.

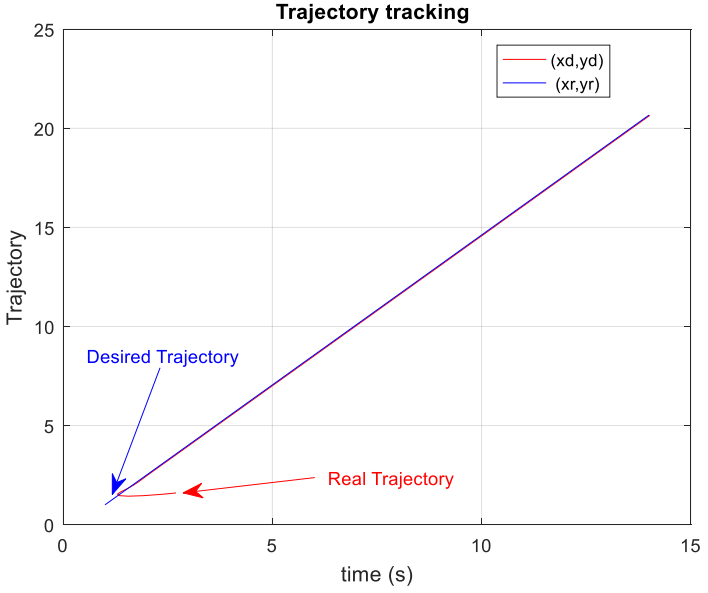


Figure 9: Desired and real trajectory.

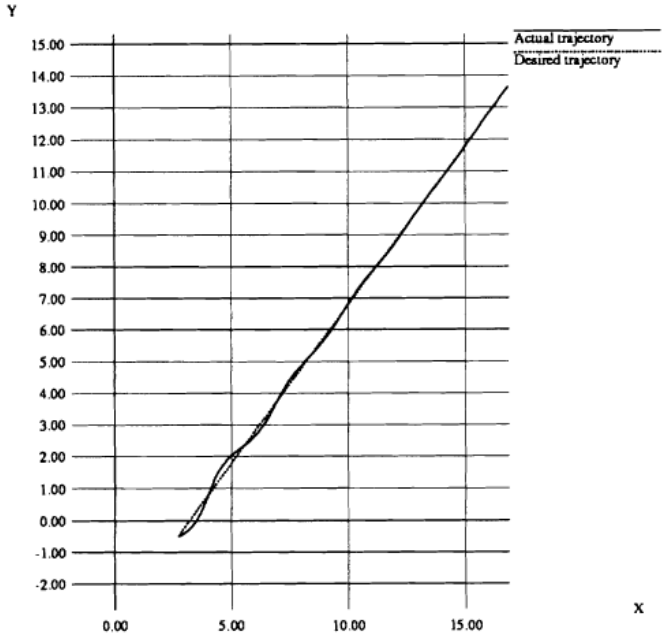


Figure 10: Desired and real trajectory (PD).

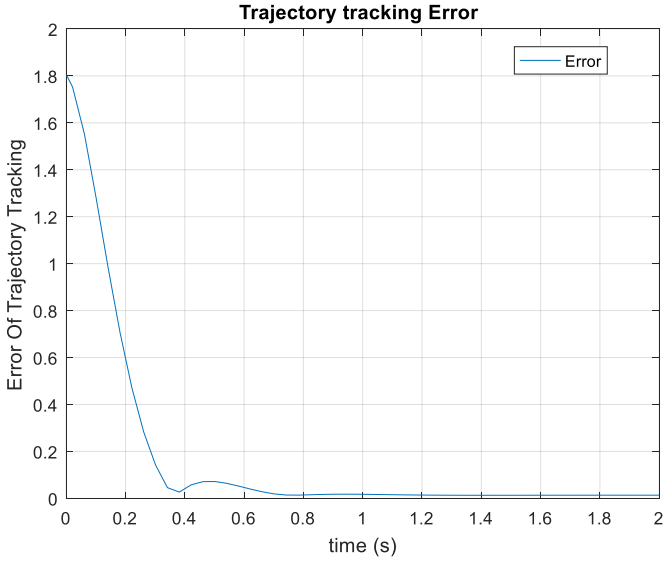


Figure 11: Trajectory tracking error.

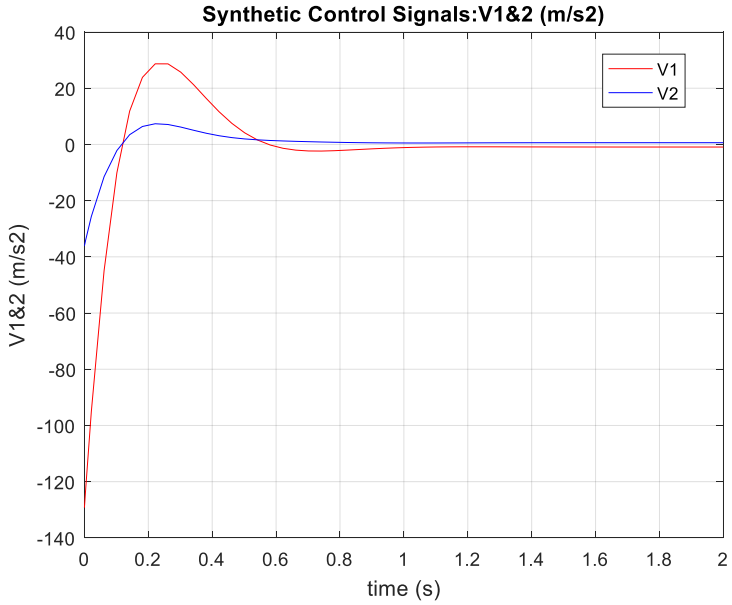


Figure 12: DDMR Synthetic control.

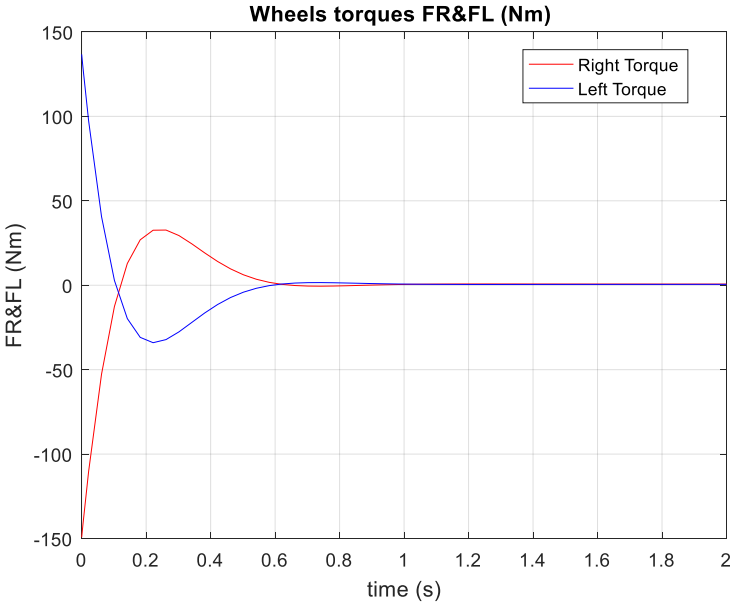


Figure 13: Wheels' driving torques.

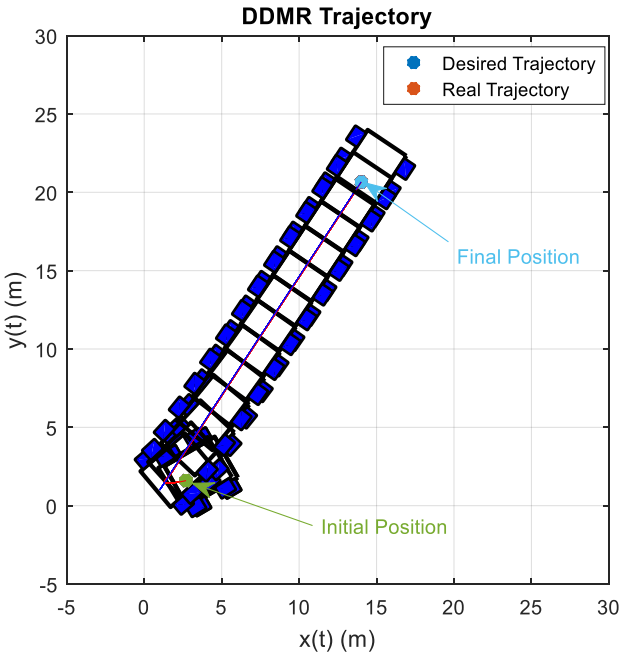


Figure 14: DDMR trajectory simulation.

Fig.7, 8 and *9* represent the trajectory of the Xc and Yc coordinates of the DDMR, from the initial to the final position; we notice fast asymptotic convergence of the two coordinates without oscillations comparing with *Fig. 10*, Yoshio and Xiaoping work [15], and the imposed settling time $T_{r\pm5\%}$ limit (less than 1s) is respected. *Fig.11* shows the convergence of the trajectory tracking error towards a very small value. using the proposed control approach. In *Fig.12* the DDMR synthetic controls v_1, v_2 given by (39) are shown. As we can notice, the synthetic controls do converge almost to zero. *Fig.13* shows that the DDMR torques τ_1, τ_2 that can be obtained from the synthetic controls using (33), respect the imposed torque limitations $\tau_{i\,\max} \leq 150$. The *Fig. 14*, shows that the DDMR follows the desired trajectory rapidly without any overshoot or oscillations.

5. Conclusion

The present article proposes a model predictive control for a DDMR. The control strategy started with an input–output linearization technique to linearize the nonlinear dynamic model of the DDMR, then, based on the obtained linear

model, a predictive control law was developed. The prediction horizon time *h* and the weight factor ρ were tuned based on torques and settling time graphics analysis; the simulation results showed that we could find a trade-off between the settling time and the energy required for the mobile robot to follow the desired trajectory. In addition, the proposed approach produces a better system performance than the PD control technique proposed by Yoshio and Xiaoping [15]. Future work aims to apply this control law in discrete form, to a mobile manipulator robot with a high degree of freedom and challenge the obtained control law to follow a more complicated predefined trajectory. After that, the validation of the proposed approach on a real robot is envisaged.

Acknowledgements

This work was supported by the Ministry of Higher Education and Scientific Research of Algeria (CNEPRU J0201620140014).

The authors would like to thank the reviewers for their valuable comments.

References

[1] Guechi, E.H., Abellard, A., and Abellard, P. "TS-Fuzzy Predictor Observer Design for TrajectoryTracking of Wheeled Mobile Robot", IECON 2011-37th annual Conf. of IEEE Industrial Electronic Society, pp. 319–324, 2011.

[2] Guechi, E. H., Bouzoualegh, S., Messikh, L. and Blažic, S. "Model Predictive Control of a two-link robot arm", 2018 Int. Conf. on Advanced Sys. and Electric Tech. (IC_ASET), Tunisia, pp. 409–414, March 2018.

[3] Belda, K. and Rovný, O. "Predictive Control of 5 DOF Robot Arm of Autonomous Mobile Robotic System", In Proc. Process Control (PC), 2017 21st International Conference on IEEE, 2017.

[4] Kamel, M. and Zhang, Y. "Linear Model Predictive Control via Feedback Linearization for Formation Control of Multiple Wheeled Mobile Robots", Information and Automation, 2015 IEEE International Conference on. IEEE, pp. 1283–1288, Aug. 2015.

[5] Guechi, E. H., Bouzoualegh, S., Zennir, Y. and Blažic, S. "MPC Control Study and LQ Optimal Control of A Two-Link Robot Arm:A Comparative Study", Machines 2018, vol. 6, no. 3, pp. 37.

[6] Elkhateeb, N. A. and Badr, R. I. "Novel PID Tracking Controller for 2DOF Robotic Manipulator System Based on Artificial Bee Colony Algorithm", Electrical, Control and Communication Engineering, vol. 13, no.1 pp. 55–62, Dec. 2017.

[7] Mendili, M. and Bouani, F. "Predictive Control Based on Dynamic Modeling of OmniDir Mobile", Engineering & MIS (ICEMIS), 2017 International Conference on IEEE, pp. 1–6, May 2017.

[8] Mendili, M. and Bouani, F. "Predictive Control of Mobile Robot Using Kinematic and Dynamic Models", Hindawi, Journal of Control Science and Eng., vol. 2017.

[9] Maniatopoulos, S., Panagou, D. and Kyriakopoulos, K. J. "Model Predictive Control for the Navigation of a Nonholonomic Vehicle with Field-of-View Constraints", American Control Conference (ACC), 2013. IEEE, pp. 3967–3972, June 2013.

[10] Sinaeefar, Z. and Farrokhi, M. "Adaptive Fuzzy Model Based Predictive Control of Nonholonomic Wheeled Mobile Robots Including Actuators Dynamics", Int. Journal of Scientific & Eng. Research, vol. 3, no. 9, Sep. 2012.

[11] Mazur, A. "Hybrid adaptive control laws solving a path following problem for Non-Holonomic mobile manipulators", International Journal of Control, vol. 77, no. 15, pp. 1297–1306, Feb. 2007.

[12] Ostafew, C. J., Schoellig, A. P. and Barfoot, T. D. "Learning-Based Nonlinear MPC to Improve Vision-Based Mobile Robot Path Tracking", Journal of Field Robotics, vol. 33, no 1, pp. 133–152, 2016.

[13] Mitrovic, S. T., & Djurovic, Z. M. "Fuzzy-Based Controller for Differential Drive Mobile Robot Obstacle Avoidance", IFAC Proceedings, vol. 43, no. 16, pp. 67–72, 2010.

[14] Dhaouadi, R., & Hatab, A. A. "Dynamic modelling of differential drive mobile robots, a unified framework", Advances in Robotics & Automation, vol. 2, no. 2, pp. 1–7, 2013.

[15] Yamamoto, Y. and Yun, X. "Coordinating Locomotion and Manipulation of a Mobile Manipulator", Decision and Control, Proceedings of the 31st IEEE Conference, pp. 2643–2648, 1992.

[16] Magni, L., Scattolini, R. and Aström, K. J. "Global stabilization of the inverted pendulum using model predictive control", Proceedings of the 15th IFAC World Congress, vol. 1554, 2002.

[17] Gawthrop, P. J. and Wang, L. "Intermittent predictive control of an inverted pendulum", Control Engineering Practice, vol. 14, no. 11, pp. 1347–1356, 2006.

[18] Mills, A., Wills, A. and Ninness, B. "Nonlinear model predictive control of an inverted pendulum", Proceedings of the American Control Conference, pp. 2335–2340, 2009.

[19] MIT Open Course Ware,"Dynamics and Control", Spring 2008.

Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 10 (2018) 42-66

DOI: 10.2478/auseme-2018-0003

GA/PSO Robust Sliding Mode Control of Aerodynamics in Gas Turbine

Abdesselam DEBBAH[1], Hamid KHERFANE[2]

[1] L2RCS Laboratory, Department of Electronics, Badji Mokhtar University, Annaba, Algeria,
Department of Petro-chemical and Process Engineering, 20 Aout 1955 University,
Skikda, Algeria, e-mail: debbeha@yahoo.fr
[2] L2RCS Laboratory, Department of Electronics, Badji Mokhtar University, Annaba, Algeria
e-mail: hamid_kherfane@yahoo.fr

Manuscript received September 15, 2018; revised December 15, 2018.

Abstract: In gas turbine process, the axial compressor is subjected to aerodynamic instabilities because of rotating stall and surge associated with bifurcation nonlinear behaviour. This paper presents a Genetic Algorithm and Particle Swarm Optimization (GA/PSO) of robust sliding mode controller in order to deal with this transaction between compressor characteristics, uncertainties and bifurcation behaviour. Firstly, robust theory based equivalent sliding mode control is developed via linear matrix inequality approach to achieve a robust sliding surface, then the GA/PSO optimization is introduced to find the optimal switching controller parameters with the aim of driving the variable speed axial compressor (VSAC) to the optimal operating point with minimum control effort. Since the impossibility of finding the model uncertainties and system characteristics, the adaptive design widely considered to be the most used strategy to deal with these problems. Simulation tests were conducted to confirm the effectiveness of the proposed controllers.

Keywords: aerodynamic instabilities, variable speed axial compressor (VSAC), sliding mode control (SMC), adaptive robust control, genetic algorithm (GA), particle swarm optimization (PSO).

1. Introduction

The increased performances are potentially achievable with modern gas turbines operating close to the maximum pressure rise, and under physical constraints [1]. This characteristic makes it very required in critical industries such as jet-engine, power generation and petrochemical. The gas turbine is however subjected to nonlinear phenomena of different nature: aerodynamic (pumping and rotating stall), aero-elasticity (the float) and combustion, that do not allow proper operation [2]. The gas turbine suffers from two types of aerodynamic instabilities, namely rotating stall and surge, which are closely

related to the limitations of their efficiency and performance [2]. Rotating stall is a non-axisymmetric perturbation that travels around the annulus of the axial compressor while surge is a large axial oscillation of flow [2].

In 1997, Gravdahl and Egeland developed a model and investigated surge and speed control. For the first time, the model developed by Gravdahl for axial compressors considered the B-parameter (proportional to the speed of the compressor) as a state and included higher harmonics of rotating stall as well [3]. Contrary to Gravdahl's variable speed model, the Moore-Greitzer original model does not imply any rotating stall development, since the working point is situated at an adequate margin from the surge line [4]. This temporary stall development and pressure drop can cause trouble for the normal turbo machines' operation. Furthermore, including model uncertainties (the precise estimation of model parameters, especially in the unstable area, being difficult) and external perturbations make the problem even more challenging [3], [4]. Finally, the squared amplitude of stall modes used as state variables are experimentally difficult to measure and full-state feedback cannot be considered in control design. In order to overcome this issue, throttle valve and Close-Coupled Valve (CCV) actuation are used to guarantee the stability, and a drive torque is applied to increase the speed of the rotor. The CCV is considered to be one of the most promising actuation methods [5]. The investigation on the model dynamics makes the Robust Sliding Mode controller (RSMC) the favorite control strategy. It is well known for its high accuracy, fast dynamic response, stability, the simplicity of implementation, and robustness for changes in uncertainties and external disturbances for dynamic systems [6]. But in real conditions, the prior knowledge on the upper bound of the disturbances and the high frequency switching known as chattering will reduce the system's robustness, and can excite unwanted dynamics that risk to damage or even destroy the system studied [6]. Also, the matched condition and affine form of the control law can't be guaranteed for all steady states especially for highly uncertain, nonlinear and complex systems. Adaptive control law could lead to a stable closed-loop system and the deviation from the sliding surface is bounded [7].

Motivation of this work comes from the fact that other past-proposed controllers usually devoted efforts on stabilizing axial compressors are based on the constant speed assumption, and even if the reported achievements [6], [7], [8], [9], [10], [11], [12], [13] investigate the variable speed model in close loop control, they propose some conservative assumptions that make the controller efficient in a very restricted operating range, as reported in [2], [5], [14]. The Gravdahl-England based models are used in order to design such controllers; however, they are idealized models for variable speed axial compressor (VSAC) systems. Therefore, they never represent the nature perfectly. From the other point of view, in real conditions, the prior knowledge of disturbance and

uncertainty becomes difficult. Thus, designing a control system in which information about disturbance, uncertainty, and dynamics of the system is used cannot be satisfactory for real applications. To accomplish the mentioned motivation, a Linear Matrix Inequality (LMI) optimization is used to design the equivalent control to guarantee the asymptotic stability regarding the speed transition behaviour, and GA/PSO optimized switching control is developed to tackle the system uncertainties, perturbations, and high-frequency behaviour of the controller with an efficient and effective constraint-handling. The proposed intelligent control system is developed and utilized to control rotating stall, surge and speed in axial compressors, without the need of prior knowledge on system uncertainties and perturbations. The outline of this paper is as follows. The variable speed axial compressor is presented in Section 2. Section 3 shows robust approach design. Section 4 describes the genetic algorithm and particle swarm optimization adaptive base on robust SMC. Section 5 shows the simulations results. Section 6 concludes with a summary and discussion.

2. The model

The compression process studied in this paper involves an intake duct, inlet guide vanes IGV, a variable speed axial compressor, the exit duct, plenum volume (turbine), varying area throttle valve, varying area close-coupled valve (*Fig. 1*). The throttle can be viewed as a streamlined model of a turbine [2].

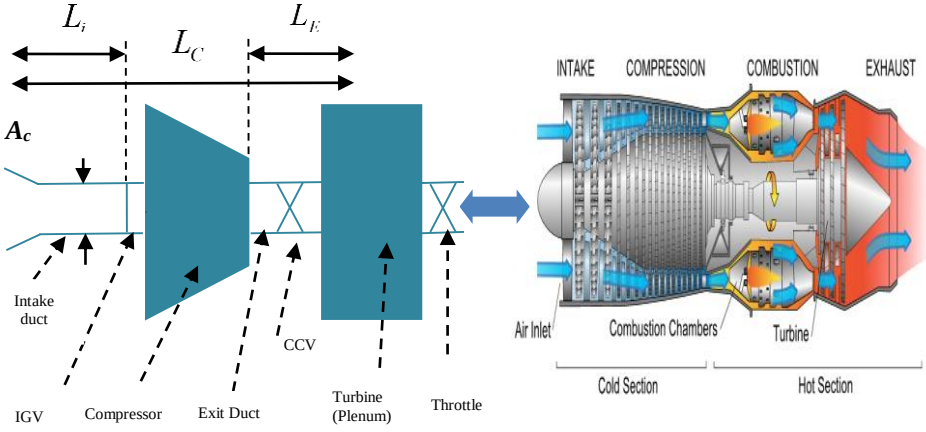


Figure 1: Schematic of the system showing non-dimensional lengths [2].

Gravdahl developed a model for the axial compressor, the exit duct, plenum volume (turbine), varying area throttle valve, varying area close-coupled valve

(*Fig. 1*). The throttle can be viewed as a streamlined model of a turbine [2]. Gravdahl developed a model for variable speed axial compressors and considered the speed of the rotor as a state variable [2]. Later, Zaiet et al. [8] modified the model to include the pressure drop over a CCV and to make it suitable for control applications. The states ϕ, ψ and U denote respectively the annulus averaged mass flow coefficient, the non-dimensional plenum pressure, and the speed of the rotor (m/s). $t = \frac{U_d \cdot t_d}{R}$ is a non-dimensional time, where t_d is the dimensional time, R is the mean compressor radius, and U_d is the desired speed. J_1 is the squared amplitude of the first harmonic of the rotating stall [5]. The actuators' forces are input variables u_1, u_2 and u_3 defined respectively as: the pressure drop over CCV, the throttle gain, and the non-dimensional drive torque being used to increase the speed. At an operating point ($\Phi_0=0.55$, $\Psi_0=0.66$, $U_0=9.617$), the dynamic model can be given in the form of state-space equations in error coordinates (see [1], [2] for more details). The model which only includes the first harmonic of the rotating stall and comprises actuator forces is given in the following equations:

$$\begin{aligned}\frac{d\phi(t)}{dt} = \frac{H}{l_c(t)}\Bigg[&-\frac{\psi(\mathrm{t}) + \Psi_0 - \psi_{c0}}{H} + 1 + \frac{3}{2}\left(\frac{\phi(t) + \Phi_0}{W} - 1\right)\left(1 - \frac{J_1(t)}{2}\right)\\ &- \frac{1}{2}\left(\frac{\phi(t) + \Phi_0}{W} - 1\right)^3 - \frac{u_1(t)}{H} - C_1 J_1(t)\\ &- G_1(\phi(t) + \Phi_0)u_3 + G_1 c(\phi(t) + \Phi_0)^3 + \Delta_\psi(t)\Bigg]\end{aligned} \tag{1}$$

$$\begin{aligned}\frac{dJ_1(t)}{dt} = J_1(t)\Bigg[&1 - \left(\frac{\phi(t) + \Phi_0}{W} - 1\right)^2 - \frac{J_1(t)}{4} - G_2 - G_3 u_3(t)\\ &+ G_3 c(\phi(t) + \Phi_0)^2 - \frac{1}{\gamma_v^2} C_2(\phi(t) + \Phi_0)\Bigg]\frac{3aH}{(1 - m_U(t)a)W}\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{d\psi(\mathrm{t})}{d\zeta} = &\frac{\Lambda_2}{U(t) + U_0}\left(\phi(t) + \Phi_0 - u_2(t)\sqrt{\psi(t)}\right)\\ &- 2\Lambda_1 \frac{U(t) + U_0}{b}\,(\psi(t) + \Psi_0)u_3(t)\\ &+ 2\Lambda_1 \frac{U(t) + U_0}{b}\,(\psi(t) + \Psi_0)c(\phi(t) + \Phi_0)^2 - \Delta_\phi(t)\end{aligned} \tag{3}$$

$$\frac{dU(t)}{d\zeta} = \Lambda_1(U(t)+U_0)^2 u_3(t) - \Lambda_1(U(t)+U_0)^2 c(\phi(t)+\Phi_0)^2 \qquad (4)$$

In the above equations:

$$I_c(t) = l_i + l_E\frac{U_d}{U(t)} + \frac{1}{a},\ m_u(t) = (1-m)\frac{U_d}{U(t)} - 1,\ C_1 = \frac{W^2}{2H\gamma_v^2},$$
$$C_2 = \frac{4W}{3H},\ G_1 = \frac{U_0\Lambda_1 l_E}{bH},\ G_2 = \frac{\mu W}{3aH},\ G_3 = \frac{2U_0\Lambda_1(m-1)W}{3Hb} \qquad (5)$$

The definition of the remaining model parameters *H, W,* $\psi_{c0}, \gamma_v, \Lambda_1, \Lambda_2, m, b, \mu, a,$ $\rho_1 \cong \frac{1}{l_c(t)}$, $\rho_2 \cong \frac{3aH}{(1-m_U(t)a)W}$ which are all positive non-zero parameters, can be found in [4], [5]. To investigate the effect of the uncertainties, we introduce Δ_Ψ and Δ_Φ in the model. Δ_Φ consists of two terms: $\Phi_d(t)$ is a time varying mass flow disturbance and introduces a constant or slow varying uncertainty in the throttle characteristic. Similarly, Δ_Ψ consists of two terms: $\Psi_d(t)$ is a time varying pressure disturbance and d_Ψ can be considered as a constant or slow varying uncertainty in the compressor map. Furthermore, it is supposed that these uncertain terms are bounded.

3. Robust design approach

Let us consider the model (1, 2, 3, 4) with (5) as a MIMO norm bounded form [15-20]:

$$\dot{\underline{x}}(t) = (\boldsymbol{A} + \Delta\boldsymbol{A}(t))\cdot\underline{x}(t) + \boldsymbol{B}\cdot(\underline{u}(t) + \underline{\Delta}_x(t))$$
$$\underline{y}(t) = \boldsymbol{C}\cdot\underline{x}(t) \qquad (6)$$

From the state variables $x=(J_1,\phi,\psi,U)\in R^4$ taking $y=(\phi,\psi,U)\in R^3$ as a smooth measurable output vector. In spite of the fact that J_1 is the fourth state variable, it cannot be measured; moreover its nature as a perturbation conveys the idea that it can be considered as an uncertain term. This approach simplifies the control design and makes the proposed control method pertinent [4]. In (6), *A* and *B* are respectively the state and control matrix of the system at the operating point considered the origin (ϕ_0, ψ_0, U_0) [18-20]. Here, $\Delta A(t)= f_1(\phi_0,\psi_0,U)$ is the uncertainty in the dynamic matrix corresponding to the variable speed behavior, $\Delta_x(t)=f_2(J_1,\phi,\psi,U)$ are the model uncertainties, external disturbance and

perturbations with unknown bound. Our objective is to stabilize the efficient operating point *(J_1=0, ϕ=0.5, ψ=0.66)* for two different speeds, low speed *(U_d=40 m/s)* and high speed *(U_d=150 m/s).*

Assumption 1: *$f_1(\phi,\psi,U)$* and *$f_2(J_1,\phi,\psi,U)$* are continuous and bounded polynomial functions of uncertainties, disturbances and *J_1*. Due to the boundedness of *J_1*, ϕ and U assumption 1 is satisfied as reported in [2], [5].

A. Reformulating the problem to equivalent sliding mode control

Consider the following linear continuous sliding function [15]:

$$\underline{\sigma_x}(t) = \boldsymbol{S}\cdot\underline{x}(t) = \boldsymbol{B}^T\cdot\boldsymbol{P}\cdot\underline{x}(t) \tag{7}$$

where $\boldsymbol{S}\in R^{3x3}$ and $\boldsymbol{P}\in R^{3x3}$ is symmetric positive definite matrix. From (6) and (7), the equivalent control law may be obtained as:

$$\underline{u_{eq}}(t) = -(\boldsymbol{S}\cdot\boldsymbol{B})^{-1}\cdot\boldsymbol{S}\cdot\boldsymbol{A}\cdot\underline{x}(t) \tag{8}$$

with $\boldsymbol{S} = \boldsymbol{B}^T\cdot\boldsymbol{P}$ and $\boldsymbol{S}\cdot\boldsymbol{B}$ is non-singular. It should be remarked that the obtained control law contains some uncertain terms, which can be deduced from the non-linear system. The non-linear part of controller called switching control, will be taken as [15]:

$$\underline{u_{nl}}(t) = \underline{u_{switching}}(t) = -(\boldsymbol{S}\cdot\boldsymbol{B})^{-1}\cdot(\left|\boldsymbol{S}\cdot\boldsymbol{B}\right|\cdot\underline{\delta_f} + \underline{\varepsilon_0})\cdot sign(\underline{\sigma_x}(t)) \tag{9}$$

where $\left|\underline{\Delta_x}(t)\right| \le \underline{\delta_f}$ and ε_0 is a positive number [15]. The Lyapunov function has been selected as [15], [16]:

$$\begin{aligned}\underline{V_x}(t) &= \frac{1}{2}\underline{\sigma_x^2}(t)\\ \dot{\underline{V}}_x(t) &= \underline{\sigma_x}(t)\cdot\dot{\underline{\sigma}}_x(t)\end{aligned} \tag{10}$$

From equations (8), (9) and (10), when *$t\ge0$* (with *t_0=0*), there exists a sliding surface $\underline{\sigma_x}(t) = \boldsymbol{S}\cdot\underline{x}(t) = 0$, i.e. $\underline{x}^T(t)\cdot\boldsymbol{S}^T(t) = 0$, the following expressions are obtained:

$$\dot{\underline{\sigma}}_x(t) = -(\left|\boldsymbol{S}\cdot\boldsymbol{B}\right|\cdot\underline{\delta_f} + \underline{\varepsilon_0})\cdot sign(\underline{\sigma_x}(t)) + \boldsymbol{S}\cdot\boldsymbol{B}\cdot\underline{\Delta_x}(t) \tag{11}$$

$$\dot{\underline{V}}_x(t) = -(\left|\boldsymbol{S}\cdot\boldsymbol{B}\right|\cdot\underline{\delta_f} + \underline{\varepsilon_0})\cdot\left|\underline{\sigma_x}(t)\right| + \underline{\sigma_x}(t)\cdot\boldsymbol{S}\cdot\boldsymbol{B}\cdot\underline{\Delta_x}(t) \le -\underline{\varepsilon_0}\cdot\left|\underline{\sigma_x}(t)\right| \tag{12}$$

As proved in [15], [23], the reachability condition is satisfied if $\varepsilon_0 > 0$ and $\Delta_x(t)$ is bounded.

B. Auxiliary feedback and stability analysis

To solve this problem, the sliding mode controller will be designed with a feedback as follows:

$$\underline{u}(t) = -\boldsymbol{K}\cdot\underline{x}(t) + \underline{\nu}(t) = \underline{u_{eq}}(t) + \underline{u_{nl}}(t) \tag{13}$$

where $\underline{\nu}(t) = \boldsymbol{K}\cdot\underline{x}(t) + \underline{u_{eq}}(t) + \underline{u_{nl}}(t)$.

$\boldsymbol{K}$ is chosen to get a $\tilde{\boldsymbol{A}} = A + \Delta\boldsymbol{A}(t) - \boldsymbol{B}\boldsymbol{K} = \overline{\boldsymbol{A}} + \Delta\boldsymbol{A}(t)$ stable in closed loop [15], [21]. Selecting the Lyapunov function as $V(t) = x^T(t)Px(t)$ [22], the time derivative of the selected function is:

$$\begin{aligned}\dot{V}(t) &= 2\cdot\underline{x}^T(t)\cdot\boldsymbol{P}\cdot\underline{\dot{x}}(t) = 2\cdot\underline{x}^T(t)\cdot\boldsymbol{P}\cdot(\tilde{\boldsymbol{A}}\cdot\underline{x}(t) + \boldsymbol{B}\cdot(\underline{\nu}(t) + \underline{\Delta_x}(t)))\\ &= 2\cdot\underline{x}^T(t)\cdot\boldsymbol{P}\cdot\overline{\boldsymbol{A}}\cdot\underline{x}(t) + 2\cdot\underline{x}^T(t)\cdot\boldsymbol{P}\cdot\Delta\boldsymbol{A}\cdot(t)\cdot\underline{x}(t) + 2\cdot\underline{x}^T(t)\cdot\boldsymbol{P}\cdot\boldsymbol{B}\cdot(\underline{\nu}(t) + \underline{\Delta_x}(t))\\ &= 2\cdot\underline{x}^T(t)\cdot\boldsymbol{P}\cdot(\overline{\boldsymbol{A}} + \Delta\boldsymbol{A}(t))\cdot\underline{x}(t) + 2\cdot\underline{x}^T(t)\cdot\boldsymbol{P}\cdot\boldsymbol{B}\cdot(\underline{\nu}(t) + \underline{\Delta_x}(t))\end{aligned} \tag{14}$$

For $t \geq t_0$, the sliding variable $\underline{\sigma_x}(t) = \boldsymbol{B}^T\cdot\boldsymbol{P}\cdot\underline{x}(t) = 0$ which implies $2\cdot\underline{x}^T(t)\cdot\mathbf{P}\cdot\mathbf{B}\cdot(\underline{\nu}(t) + \underline{\Delta_x}(t)) = 0$.

Theorem 1: The uncertain sliding dynamics in (14) is asymptotically stable in closed loop with a state feedback, for Lyapunov function candidate $V(t) = \underline{x}^T(t)\cdot\boldsymbol{P}\cdot\underline{x}(t)$, if there exists a symmetric matrix $P > 0$, satisfying the following LMI:

$$\begin{aligned}(\boldsymbol{A} + \Delta\boldsymbol{A}(t) - \boldsymbol{B}\cdot\boldsymbol{K})\cdot\boldsymbol{Q} + \boldsymbol{Q}\cdot(\boldsymbol{A} + \Delta\boldsymbol{A}(t) - \boldsymbol{B}\cdot\boldsymbol{K})^T + 2\cdot\alpha\cdot\boldsymbol{Q} < 0\\ \boldsymbol{Q} > 0\\ \alpha > 0\end{aligned} \tag{15}$$

with $\boldsymbol{Q} = \boldsymbol{P}^{-1}$. The closed loop system matrix has its eigenvalues strictly on the left hand side of the line α, in complex s-plan.

C. Robust control design

The equation (15) is non-linear matrix inequality, difficult to solve being non convex. It can be solved by increasing α in order to shift the eigenvalues of $\boldsymbol{A}+\Delta\boldsymbol{A}(t)-\boldsymbol{B}\cdot\boldsymbol{K}$ progressively toward the region that guarantees the stability.

However, in order to make the controller more robust against the model uncertainties and non-linearity, we will propose robust design [24], [25].

Consider the uncertain matrix $\Delta\boldsymbol{A}(t)=\boldsymbol{M}\cdot\boldsymbol{\Delta}(t)\cdot\boldsymbol{N}$, where $\boldsymbol{M}$ and $\boldsymbol{N}$ are known, and $\boldsymbol{\Delta}(t)$ is an unknown matrix satisfying $\boldsymbol{\Delta}(t)^T\cdot\boldsymbol{\Delta}(t)\le I$ [26]. Note that this congruence transformation does not change the definiteness of $\boldsymbol{\Delta}(t)$.

Theorem 2: The uncertain sliding dynamic in (15) can be robustly stabilized if there exists $\boldsymbol{Q}^T>0$, $\boldsymbol{K}>0$ and $\alpha>0$ satisfying the following LMI:

$$\begin{bmatrix} \boldsymbol{A}\cdot\boldsymbol{Q}+\boldsymbol{Q}\cdot\boldsymbol{A}^T-\boldsymbol{B}\cdot\boldsymbol{K}\cdot\boldsymbol{Q}-\boldsymbol{Q}^T\cdot\boldsymbol{K}^T\cdot\boldsymbol{B}^T+2\cdot\alpha\cdot\boldsymbol{Q}+\varepsilon_1\cdot\boldsymbol{M}\cdot\boldsymbol{M}^T & \boldsymbol{N}\cdot\boldsymbol{Q} \\ \boldsymbol{Q}^T\cdot\boldsymbol{N}^T & -\varepsilon_1\cdot\boldsymbol{I} \end{bmatrix}<0 \quad (16)$$

Proof: by replacing $\Delta\boldsymbol{A}$ by $\boldsymbol{M}\cdot\boldsymbol{\Delta}(t)\cdot\boldsymbol{N}$ in (15), it yields

$$\begin{aligned}&\boldsymbol{A}\cdot\boldsymbol{Q}+\boldsymbol{Q}\cdot\boldsymbol{A}^T-\boldsymbol{B}\cdot\boldsymbol{K}\cdot\boldsymbol{Q}-\boldsymbol{Q}\cdot\boldsymbol{B}\cdot\boldsymbol{K}+2\cdot\alpha\cdot\boldsymbol{Q}+\boldsymbol{M}\cdot\boldsymbol{\Delta(t)}\cdot\boldsymbol{NQ}\\&+\boldsymbol{Q}^T\cdot\boldsymbol{N}^T\cdot\boldsymbol{\Delta(t)}\cdot\boldsymbol{M}^T<0\end{aligned} \quad (17)$$

with the assumption $\boldsymbol{\Delta}(t)^T\boldsymbol{\Delta}(t)\le I\rightarrow\|\boldsymbol{\Delta}(t)\|<I$, as given in [25] and [26], it follows that:

$$\boldsymbol{M}\cdot\boldsymbol{\Delta}(t)\cdot\boldsymbol{N}\cdot\boldsymbol{Q}+\boldsymbol{Q}^T\cdot\boldsymbol{N}^T\cdot\boldsymbol{\Delta}(t)\cdot\boldsymbol{M}^T\le\varepsilon_1^{-1}\cdot\boldsymbol{M}\cdot\boldsymbol{M}^T+\varepsilon_1\cdot\boldsymbol{N}\cdot\boldsymbol{Q}\cdot\boldsymbol{Q}^T\cdot\boldsymbol{N}^T \quad (18)$$

with $\varepsilon_1>0$, the inequality (17) is satisfied if the following equation is satisfied:

$$\begin{aligned}&\boldsymbol{A}\cdot\boldsymbol{Q}+\boldsymbol{Q}\cdot\boldsymbol{A}^T-\boldsymbol{B}\cdot\boldsymbol{K}\cdot\boldsymbol{Q}-\boldsymbol{Q}\cdot\boldsymbol{B}\cdot\boldsymbol{K}+2\cdot\alpha\cdot\boldsymbol{Q}+\varepsilon_1^{-1}\cdot\boldsymbol{M}\cdot\boldsymbol{M}^T\\&+\varepsilon_1\cdot\boldsymbol{N}\cdot\boldsymbol{Q}\cdot\boldsymbol{Q}^T\cdot\boldsymbol{N}^T<0\end{aligned} \quad (19)$$

Using Schur complement, we can put (19) in the form (16) as desired. The proposed robust Sliding Mode Controller (SMC) can be constructed similar to the previous algorithm by replacing (15) by (16) [27].

D. Chattering reduction

The sliding mode control law of (13), with LMI constraints (19) guarantees the asymptotic stability of $x(t)=0$ in error coordinates, eliminating the effect of uncertainties and perturbations on system state variables. In order to restrain the chattering phenomena, a continuous function $\underline{k}\cdot\underline{\sigma_x}(t)$ can be chosen instead of

discontinuous function $sign(\underline{\sigma_x}(t))$ [23], [28], [29]. The nonlinear part of the controller can be expressed as:

$$\underline{u_{nl}}(t) = -(\boldsymbol{S}\cdot\boldsymbol{B})^{-1}\cdot(\left|\boldsymbol{S}\cdot\boldsymbol{B}\right|\cdot\underline{\delta_f} + \underline{\varepsilon_0})\cdot\underline{k}\cdot\underline{\sigma_x}(t) \tag{20}$$

with $k > 0$.

4. GA/PSO robust design approach

From theoretical point of view, the proposed controller can be considered as an optimization of a robust sliding mode controller. The optimization of the controller parameter has a vital role in the design of such a sliding mode controller. A properly optimized controller tries to minimize an appropriate objective function of the system and it assures the process output to track the desired target as well as to reduce the effect of perturbations affecting the system. To optimize a sliding mode controller, there is not any method that has been specified in the literature survey [30]. In the present work, GA and PSO are used to optimize the controller parameters.

A. The proposed Algorithm

The steps involved in the proposed GA/PSO Robust sliding mode control algorithm are:

Phase A (previous sections) - Robust sliding mode control:

Step1: Design of the dynamic control based on equivalent and switching control, as illustrated in equations (8) and (9).
Step2: Ensure the asymptotic stability of the proposed controller, as illustrated in equations (12) and (14).
Step3: After designing the robust sliding surface, the system dynamic will be driven onto the sliding surface, and remain on it. The resulting problem in (14) can be transformed to Linear Matrix Inequality (LMI) optimization in equation (16). These resulting optimization problems can be solved numerically very efficiently using developed interior-point methods implemented in MATLAB software.

Phase B (current Section) - Optimization of robust sliding mode control:

Step4: In the conventional robust sling mode control, it is primordial to have the information about the uncertainties, in order to design a control law with switching part dominating the effect of perturbations [29]. To overcome this, in this section we propose to optimize the parameters of robust sliding mode control

using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) based on using equations (12) and (20), and on the block diagram in *Fig. 2*.

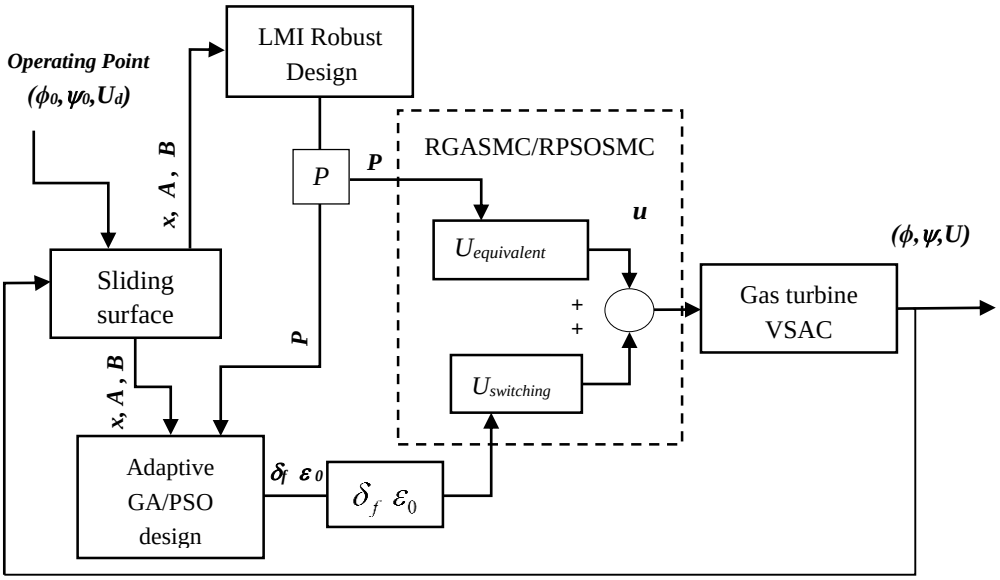


Figure 2: The block diagram of proposed control.

The objective function comprises robust performance and stability criterion which is required to optimize the use of switching control efforts. The objective function is given by equation (21).

$$\begin{aligned} &\min \quad f(\underline{\delta_f}, \underline{\varepsilon_0}, \underline{k}) = \underline{u}(\underline{\delta_f}, \underline{\varepsilon_0}, \underline{k})^T \cdot \underline{u}(\underline{\delta_f}, \underline{\varepsilon_0}, \underline{k}) \\ &\textit{Subjected to} \; -(|\mathbf{S}\cdot\mathbf{B}|\cdot\underline{\delta_f} + \underline{\varepsilon_0})\cdot\underline{k}\cdot\underline{\sigma_x}^2(t) + \sigma_x(t)\cdot\mathbf{S}\cdot\mathbf{B}\cdot\underline{\Delta_x}(t) < 0 \cdots (a) \\ &u_2 > 0 \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (b) \\ &\underline{\delta_f} > 0 \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (c) \\ &\underline{\varepsilon_0} > 0 \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (d) \end{aligned} \tag{21}$$

In equation (21), the constraint (a) is required to generate a robust stability bound for the given model specifications and amount of plant uncertainties by using compressor map curve and throttle dynamic. The reachability condition (a) is a feasible optimization $-(|\boldsymbol{S}\cdot\boldsymbol{B}|\cdot\underline{\delta_f} + \underline{\varepsilon_0})\cdot\underline{k}\cdot\underline{\sigma_x}^2(t) + \underline{\sigma_x}(t)\cdot\boldsymbol{S}\cdot\boldsymbol{B}\cdot\underline{\Delta_x}(t) \le -\underline{\varepsilon_0}\,\underline{k}\,\underline{\sigma_x}^2(t)$, for $\underline{\varepsilon_0} > 0$ and $\underline{k}\underline{\sigma_x}^2(t)$ is a positive definite function [15], [19]. Then, with the

flexibility of the GA and PSO algorithms, these numerical bounds can be used directly in an online optimization of decision variables δ_f, ε_0 and k is a positive constant as an adaptive gain.

B. Genetic Algorithm Optimization (GA)

The genetic algorithm is a heuristic approach to solving a non-linear optimization problem, which is essentially based on the theory of natural selection, the process that drives biological evolution [31]. In all global search problem, there is an optimization problem of maximizing or minimizing an objective function for a given space of arbitrary dimension [32-34]. In this paper the objective function is the equation (21), where δ_f, ε_0 are decision variables. The flowchart in *Fig. 3* explains the process in brief. The implementation of the GA is based on the following fundamental initializations: chromosome representation, selection function, the genetic operators, initialization, termination and evaluation function. A variety of constraints-handling methods for genetic algorithms have been developed in the last decades. Most of them can be classified into two main types of concepts: penalty function and multi-objective optimization concept [34], [35]. In this work, the used concept to constraints-handling is the penalty function.

C. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a derivative-free global optimum solver. It is inspired by the surprisingly organized behavior of large groups of simple animals, such as flocks of birds, schools of fish, or swarms of locusts [36]. The nonlinear optimization is illustrated in equation (21), where δ_f, ε_0 are decision variables. The flowchart in *Fig. 4* explains the process in brief. The individual creatures, or "particles", in this algorithm are primitive, knowing only four simple things, their own current location in the search space and fitness value, their previous personal best location, and the overall best location found by all the particles in the "swarm". There are no gradients or Hessians to calculate. Each particle continually adjusts its speed and trajectory in the search space based on this information, moving closer towards the global optimum with each iteration. As seen in nature, this computational swarm displays a remarkable level of coherence and coordination despite the simplicity of its individual particles. While the particles in the PSO algorithm are searching the space, each particle remembers two positions. The first is the position of the best point the particle has found (self- best), while the second is the position of the best point found among all particles (group-best). Let X and V represent the particle position and velocities in the given search space, respectively. Therefore, the i-th particle is

represented as $X_i = (x_{i1}, \ldots\ldots x_{im})$, in the m-di-mensional search space. The previous position of the i-th particle is recorded and represented as $j_{p\,best} = (j_{p\,best_{t1}}, \ldots\ldots j_{p\,best_{tm}})$. The index of the best particle among all the particles in the group is represented by j_{pbest}. The rate of the velocity for particle i is represented as $V_i = (v_{i1}, \ldots\ldots v_{in})$. The modified velocity and position of each particle can be calculated using the current velocity and distance from J_{pbest} and J_{gbest} use the following equations:

$$\underline{V_i^{t+1}} = Q_p \cdot \underline{V_i^t} + K_1 \cdot rand_1 \cdot (\underline{X_{pbest}} - \underline{X_i^t}) + K_2 \cdot rand_2 \cdot (\underline{X_{gbest}} - \underline{X_i^t}) \tag{22}$$

$$\underline{X_i^{t+1}} = \underline{X_i^t} + \gamma \cdot \underline{V_i^{t+1}} \tag{23}$$

Where K_1 and K_2 are two positive constants, $rand_1$ and $rand_2$ are random numbers in the range [0,1], and Q_p is the inertia weight. X_i^t represents the current position of the i-th particle and V_i^t is its current velocity. The positions of the particles are updated using Equation (23), where X_i^{t+1} is the new position of the i-th particle of m-dimensional search space, where "*iter*" is the iteration count [37]. Particle swarm optimization is guided by the quality of its candidate solutions. Consequently, an obvious solution to constraint handling is to penalize the fitness of infeasible methods. Penalty method (penalty functions) is easy to implement, and shows an improvement of the approximation of optima with active constraints [36], [38]. The weight Q_p is updated using the following equation:

$$Q_p = Q_{p\max} - \left[\frac{Q_{p\max} - Q_{p\min}}{iter_{\max c}} \right] \cdot iter \tag{24}$$

The parameters used for GA and PSO performed in the present study are given in *Table 1* and *Table 2*.

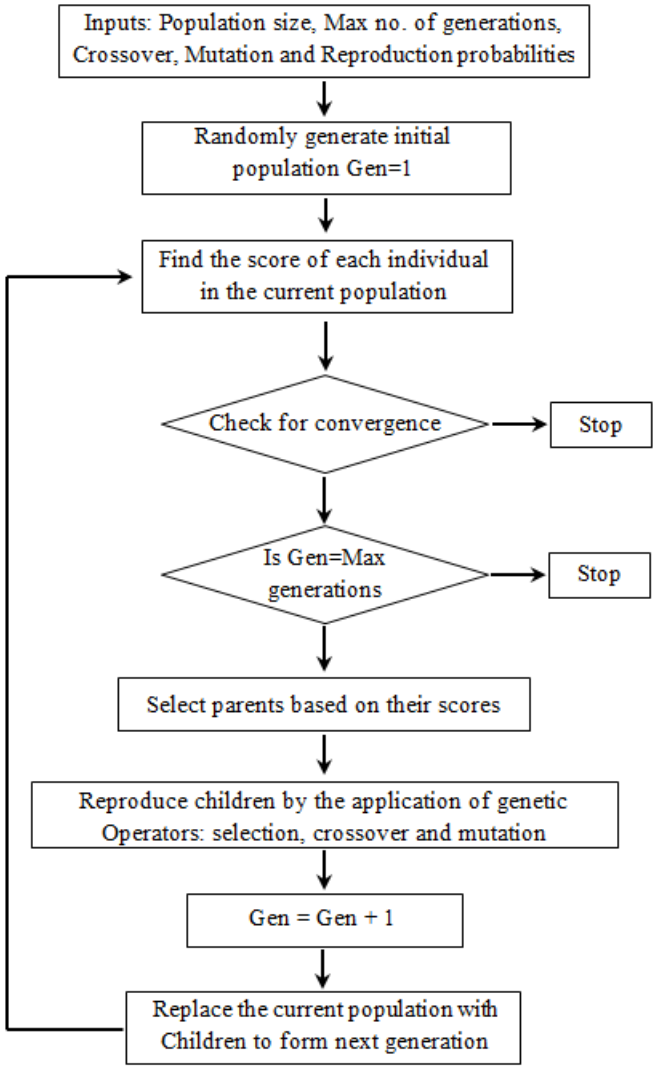


Figure 3: Flowchart of Genetic Algorithm Optimization (GA).

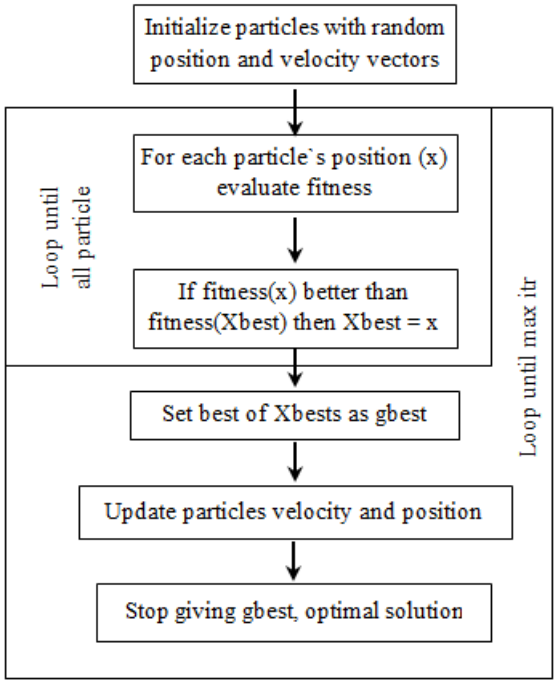


Figure 4: Flowchart of Particle Swarm Optimization (PSO).

Table 1: GA Parameters used in the simulation

Parameter	Value
Population size	40
Maximum number of generations	50
Type of selection	Roulette wheel
Type of crossover	Intermediate
Fitness function	Equation (21)
Constraints-handling methods	Penalty function
Type of mutation	adapt feasible
Crossover Ration	0.8

Table 2: PSO Parameters used in the simulation

Parameter	Value
Maximum iteration	50
Population size	40
Dimension	2
Value of K2	2.0
Maximum weight	0.90
Minimum weight	0.40
Fitness function	Equation (21)

5. Numerical simulation

For simulation purposes it is considered the compressor model of Gravdahl as given in equations (1) to (5), with numerical values are given in *Table 3* [2], [5].

Table 3: Model parameters used in simulation

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
W	*0.25*	a	*0.3*	b	*96.16*	d_Ψ	*0.02*
H	*0.18*	l_i	*1.75*	m	*1.75*	Λ_1	*2.168e-4*
μ	*0.01*	d_Φ	*−0.05*	l_E	*3*	Λ_2	*0.0189*
ρ_1	$0<\rho_1<1$	ρ_2	$0<\rho_2<1$	c	*0.7*	γ_v	*1*

The aim of this simulation is demonstrating the effectiveness and the robustness of the proposed controllers in preventing the compressor from developing temporary rotating stall ($J_1>0$), and pressure drop under the following critical operating conditions:

1- Constrained Throttle valve opening ($u_2>0$): It is interesting to note that in [2], [4], [5], [14] it is reported that the saturated effort of the throttle valves can cause a temporary rotating stall.

2- Speed Transition: As reported in [2], [4], [5], [14], when speed varies at an efficient operating point (0.5,0.66) temporary stall developments can lead to a fully developed rotating stall.

3-Pertubations: Previously reported results in [1], [2], [5] show that pressure and flow external perturbations can destroy the stability of compressors at an efficient operating point (0.5, 0.66) and lead to fully developed rotating stall or deep surge

depending on the speed of the rotor (i.e. for low speeds the system goes to rotating stall and for high speeds it develops deep surge). Two types of perturbations are applied to the system denoted by $\Phi_d(t)=\Psi_d(t)=0.01sin(0.2t)$, they are considered as mass flow and pressure disturbances respectively and, d_Φ, d_Ψ represent the uncertainty of the compressor map and throttle characteristic. At $t=1000$ a higher perturbation magnitude $\Phi_d(t)=\Psi_d(t)=0.1sin(0.2t)$ is applied to system to check the robust and adaptive behaviour of the three proposed controllers. The simulation numerical values are given in *Table 3* [2], [5].

In **TEST 1**, perturbations are applied and constrains are considered. A low desired speed U_d *=40 m/s* of the turbine is considered. In **TEST 2**, perturbations are applied and constrains are considered. A high desired speed U_d *=150 m/s* of the turbine is considered. In order to illustrate the advantage of the proposed robust sliding mode controller without optimization **(RSMC)**, Genetic Algorithm optimized robust sliding mode controller **(RGASMC)**, and particle swarm optimized robust sliding mode controller **(RPSOSMC)** a comparative simulation is carried out using Matlab software.

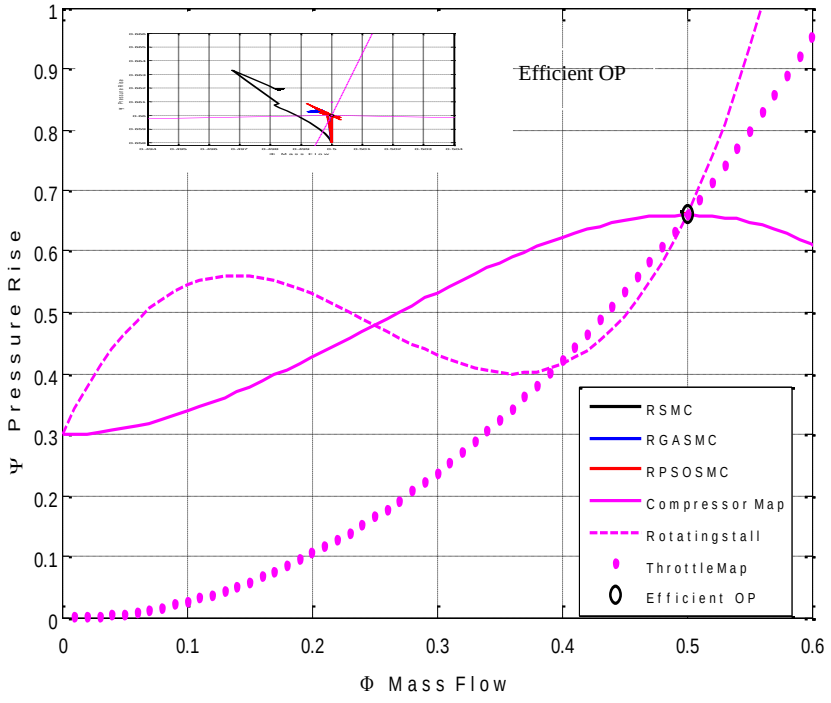


Figure 5: Closed loop system map ***TEST1.***

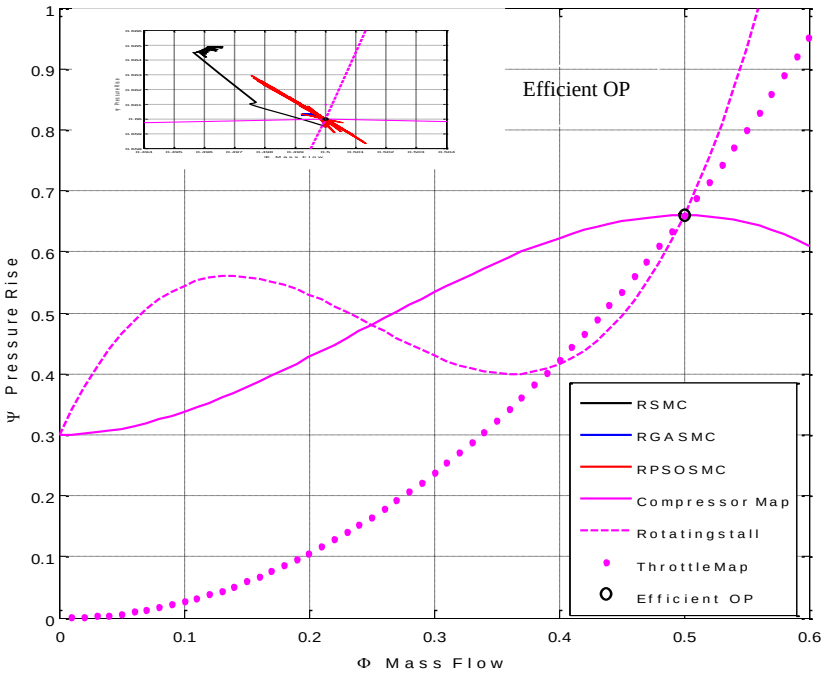


Figure 6: Closed loop system map ***TEST 2.***

For the tests **TEST1** and **TEST2**, *Fig. 5* and *Fig. 6* show the variables Φ and Ψ in the phase space along with compressor map and stall characteristic. The system starts from an effective initial operating point (OP) at the top of the compressor map. At *t=0*, the controller is activated and closes the loop. Examining *Fig.* 5 and *Fig. 6*, we found that, the proposed controllers effectively stabilize the compression system at the efficient point OP and prevent it from developing a steady rotating stall due to the speed variation, thus limiting the throttle valve opening coefficient which must always be positive·

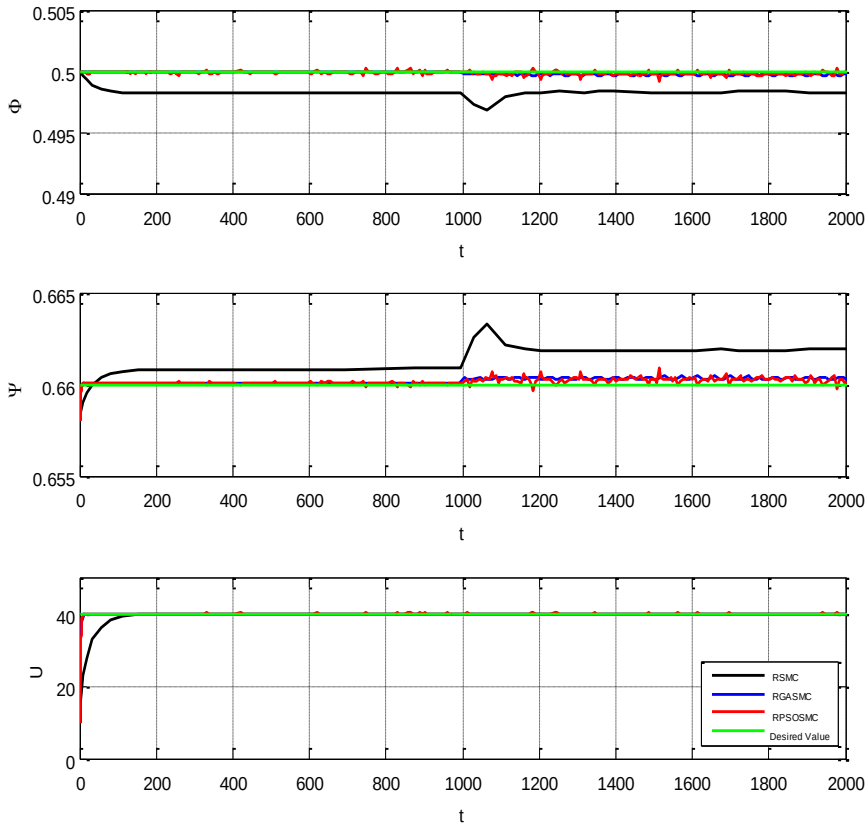


Figure 7: Output dynamic in closed loop ***TEST 1.***

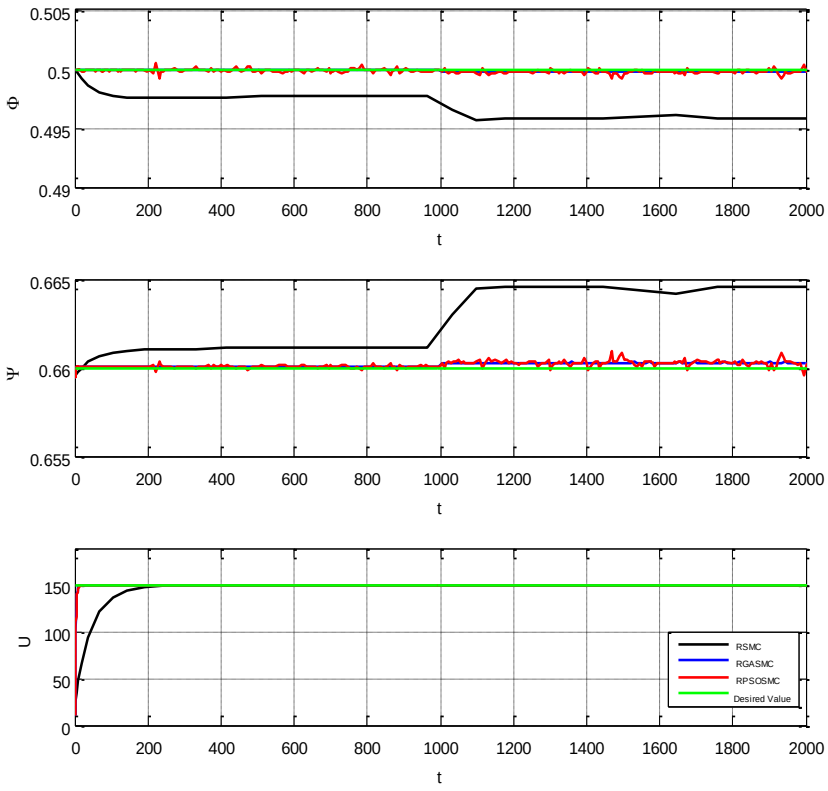


Figure 8: Output dynamic in closed loop ***TEST 2.***

In *Fig. 7* and *Fig. 8*, we found that the RGASMC and RPSOSMC controllers make the compressor operating close to his efficient OP $(\Phi,\Psi)=(0.5,0.66)$ despite the existence of uncertainties, and perturbation (negligible variation). The robust sliding mode controller (RSMC) can't reject the effect of the perturbation. This can be explained by the need of the prior knowledge of the upper bound of the perturbations and uncertainties, governed by $|\Delta_x| \leq \delta_f$. Compared to (RSMC) and many control strategies proposed in the previous literature, one advantage of the proposed GA and PSO controllers designed in this paper is their ability to be applied in real applications without a need to a prior knowledge on perturbations and uncertainties.

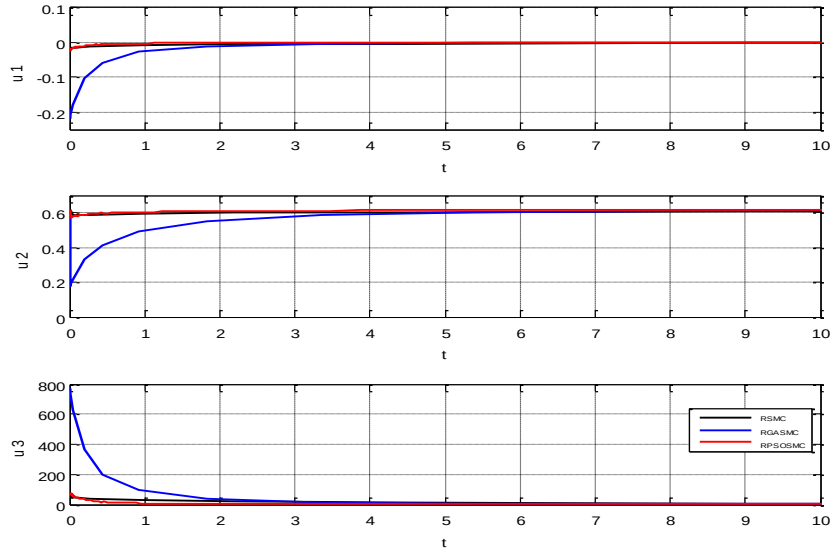


Figure 9: Control efforts dynamic in closed loop ***TEST 1.***

In *Fig. 9* and *Fig. 10*, we have noted a variation in throttle actuator, despite that the system still reaches its stable OP, where the pressure is high enough for normal operation of the gas turbine.

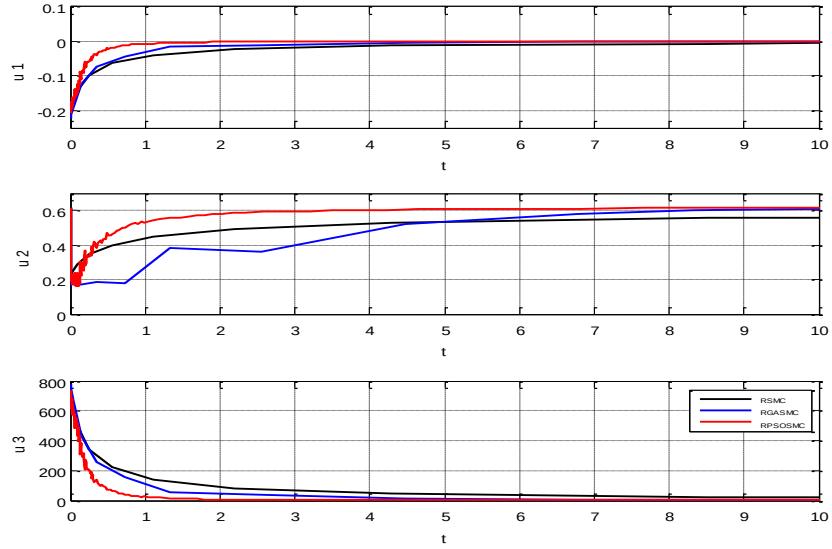


Figure 10: Control efforts' dynamic in closed loop ***TEST 2.***

The throttle gain decrease is caused by the high level of perturbation on the pressure rate, which is a consequence of the low speed of the turbine during the starting phase (9.617 m/s). The throttle valve immediately damps out rotating stall as illustrated in [10], it should be turned down in order to add some resistance to the compression system when the flow change is positive and the pressure change rises is not negative. It can be seen that the throttle gain for the three controllers is still positive and lower than one $0 < u_2 < 1$.

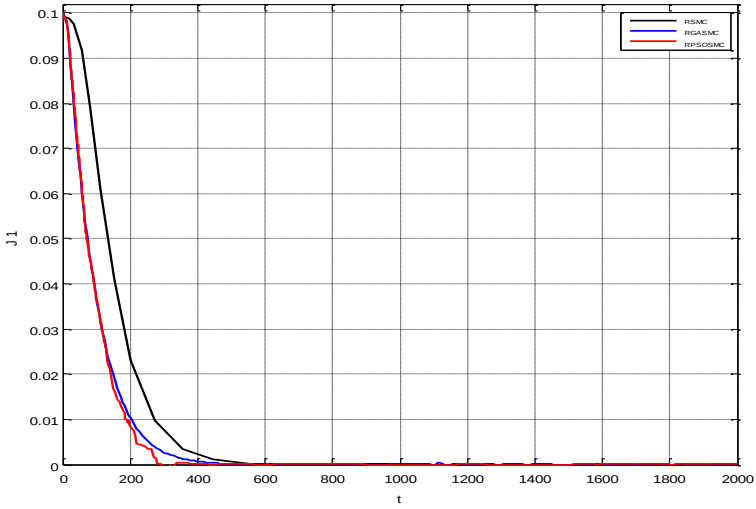


Figure 11: The first harmonic of rotating stall ***TEST 1.***

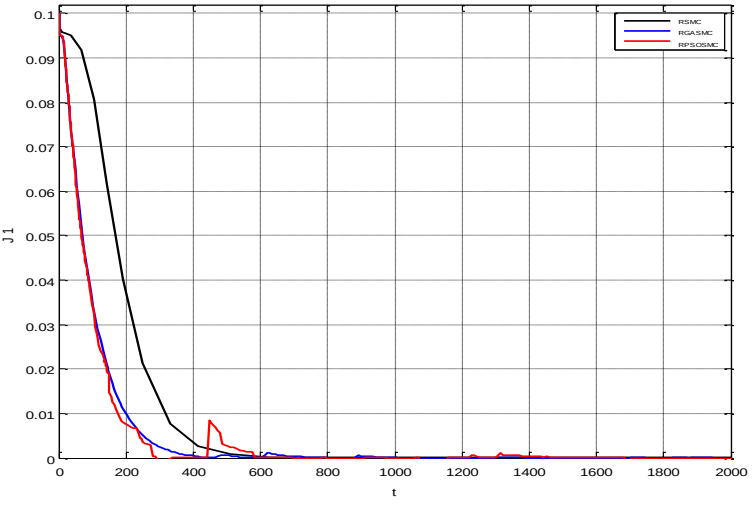


Figure 12: The first harmonic of rotating stall ***TEST 2.***

Fig. 11 and *Fig. 12* show the effectiveness of the proposed control law regarding the rotating stall control. It can be noticed that, even though the cumulative computational time increases linearly with the number of generations for both PSO and GA, the computational time for GA is low compared to the PSO optimization algorithm.

6. Conclusion

This paper has presented a GA and PSO adaptive sliding mode controller design based on linear matrix inequality. The proposed approach is applied to the gas turbine, which is a variable speed system, by its nature. This turbine suffers from temporarily developed instabilities which may lead to a steady and fully developed rotating stall or surge. The used model reveals significantly the impact of speed transitions (measurable output) and throttle gain (control effort) on the stability of the compression system. The addition of model uncertainties and external perturbations and impossibility to have a full feedback control (rotating stall is not measurable) constitute a challenging issue. The proposed controllers do not require precise knowledge of the compressor map, an upper bound of the uncertainties and perturbations, and do not use a full-state feedback. Time-domain simulations have demonstrated that RGASMC and RPSOSMC controllers are still stable, close to desired performances and are damping out system instabilities including surge and rotating stall.

Appendix: Nomenclature of the model variables:

ϕ: Annulus averaged mass flow coefficient

ψ: Plenum pressure rise coefficient

J_1: The first mode squared amplitude of rotating stall

U: Rotor tangential velocity at mean radius

U_d: Desired constant velocity

γ_T: Throttle Gain

γ_v: Close Coupled valve gain

t: Non-dimensional time

t_d: Dimensional time

U_0: compressor initial velocity

$d_\phi, d\psi$: Mass flow and pressure uncertainty

Φ_d, Ψ_d: Time varying and mass flow pressure disturbance

R: mean compressor radius

$\Psi_c(\Phi)$: Compressor characteristic

$\Psi_s(\Phi)$: Stall characteristic

H: semi-height of the compressor characteristic

W: semi-width of the compressor characteristic

ψ_{c0}: shut-off value of the compressor characteristic

l_c, l_i, l_E : Effective flow passage non-dimensional length of the compressor, Inlet duct and exit duct respectively.

m : Compressor duct flow parameter

c: Coefficient of compressor torque.

a : Reciprocal time lag parameter of the blade passage

a_s : Sonic velocity

μ : Viscosity

Λ_1, Λ_2 : Constants in Greitzer model

b : Constant in Greitzer model

References

[1] Gravdahl, J. T., and Egeland, O., "Moore-Greitzer Axial Compressor Model with Spool Dynamics", in *Proc. of 36th IEEE Conference on Decision and Control*, 1997, pp. 4714–4719.

[2] Gravdahl, J. T., and Egeland,O., "Compressor surge and rotating stall Modelling and control", Advances in Industrial Control. Springer Verlag. London Limited, 1999.

[3] Sari, G., Akhrif, O., and Saydy, L., "Qualitative Analysis of an Axial Compressor Model with Non-constant Speed", in *Proc. of ASME 2011 Power Conference*, Vol. 1, 2011, pp. 515–524.

[4] Sari, G., Akhrif, O., and Saydy, L., "The impact of speed variation on the stability of variable speed axial compressors at efficient operating points", in *Proc. of American Control Conference*, 2012, pp. 404–410.

[5] Sari.O., "Model analysis and nonlinear control of air compressors", *PhD thesis*. Montreal University, 2014.

[6] Sheng, H., Huang, W., Zhang, T., and Huang, X., "Robust Adaptive Fuzzy Control of Compressor Surge Using Back-stepping", *Arabian Journal for science and engineering*, Vol. 39, Issue.2, pp. 9301–9308, December 2014.

[7] Moghaddam, J. J., and Madani, M., "A decoupled adaptive neuro-fuzzy sliding mode control system to control rotating stall and surge in axial compressors", *Journal of Expert Systems with Applications*, Vol. 38, Issue. 4, pp. 4490–4496, April 2011.

[8] Zaiet, C., Akhrif, O., and Saydy, L., "Modeling and Non Linear Control of a Gas Turbine", International Symposium on Industrial Electronics ISIE 2006, 2006, pp. 2688–2694.

[9] Fontaine, D., Liao, S., Paduano, J., and Kokotovic, P., "Nonlinear control experiments for rotating on an axial flow compressor", in *Proc. of 39th IEEE conference on decision and control*, 2000, pp. 1329–1334.

[10] Lin, S., Yang, C., Wu, P., and Song, Z., "Fuzzy logic surge control in variable speed axial compressors", in *Proc. of 10th IEEE International Conference on Control and Automation* (*ICCA*), 2013, pp. 1178–1183.

[11] Uddin, N., and Gravdahl, J.T., "Active compressor surge control using piston actuation", in *Proc. ASME 2011 Dynamic Systems and Control Conference DSCC2011*, 2011, pp. 69–76.

[12] Moghaddam, J. J., Farahani, M. H., and Amanifard, N., "A neural network-based sliding-mode control for rotating stall and surge in axial compressors", *Journal of Applied Soft Computing*, Vol.11, Issue. 1, pp. 1036–1043, January 2011.

[13] Sheng, H., Huang, W., and Zhang, T., "Output feedback control of surge and rotating stall in axial compressors", *Asian Journal of Control*, Vol. 19, Issue 1, pp.1–7, September 2016.

[14] Debbah, A., Kharfene, H., Karboua, A., "Using Robust sliding mode controller to improving the aerodynamic performance of a gas turbine", in *Proc. of the 4th International Seminar on New and Renewable Energies*, 2016, pp. 1–6.

[15] Lui, J., "Advanced sliding mode control for mechanical systems". Springer. Tsinghua University Press. Beijing, 2012.

[16] Defoort, M., Floquet, T., Kokosy, A., and Perruquetti, W., "A novel higher order sliding mode control scheme", *Journal of Systems and Control Letters Elsevier*, Vol. 58, Issue. 2, pp. 102–108, 2009.

[17] Tsai, C., and Wu, H., "Robust passivity-based control of weakly minimum phase nonlinear uncertain systems: An application to manipulator", in *Proc. of 7th Asian Control Conference*, ASCC, 2009, pp. 919–924.

[18] Feher, A., Kutasi, D. N., "Modelling and control of bounded hybrid systems in power electronics*", Acta Universitatis Sapientiae, Electrical and Mechanical Engineering*, Vol. 9, Issue 1, pp. 33–42, December 2017.

[19] Horvath,Z., Edelmayer, A., "Robust model-based detection of faults in the air path diesel engines ", *Acta Universitatis Sapientiae, Electrical and Mechanical Engineering*, Vol. 7, Issue 1, pp. 5–22, December 2015.

[20] Fodor, D., "Experimental investigation on robust control of induction motor using H_∞ output feedback*", Acta Universitatis Sapientiae, Electrical and Mechanical Engineering*, Vol. 1, Issue 1, pp. 29–40, December 2009.

[21] Chen, P.V., Chiang, C. H., and Liu, J. C., "The input/output constrained control for V/STOL aircraft dynamics under a descriptor system formulation", *Journal of Aeronautic and aviation*, Vol. 41, Issue 1, pp. 43–52, January 2009.

[22] Kelemen, A., Kutasi, N., "Lyapunov-based frequency-shift power control of induction-heating converters with hybrid resonant load", *Acta Universitatis Sapientiae, Electrical and Mechanical Engineering*, Vol. 1, Issue 1, pp. 41–52, December 2009.

[23] Leu,V. Q., Choi, H. H., and Jung, J. H., "LMI based sliding mode speed tracking control design for surface-mounted permanent magnet synchronous Motors", *Journal of Electrical Engineering and Technology*, Vol. 7, Issue 4, pp. 513–523, July 2012.

[24] Der-Cherng, L., Ren, S. M., and Shih-Tse, S., "A Feedback Linearization Design for Compressor's Surge Control", in *Proc. of Industrial Technology ICIT*, 2008, pp. 1–6.

[25] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., "Linear Matrix Inequalities in System and Control Theory", SIAM. Philadelphia, 1994.

[26] Argha, A., Li, L., SU, S. H., and Nguyen, H., "A new LMI-Based Robust Sliding Mode control for the uncertain Discrete-Time system", in *Proc. of 53th IEEE Conference on Decision and Control*, 2014, pp. 4747–4752.

[27] Mahmoud, M. S., and Soliman, H. M., "Design of Robust Power System Stabilizer Based on Particle Swarm Optimization", *Journal of Circuits and Systems*, Vol. 3, pp. 82–89, 2012.

[28] Chen, D.Y., Liu, Y. X., Ma, X. Y., and Zhang, R. F., "No-chattering sliding mode control in a class of fractional-order chaotic systems", *Chinese Physics B*, Vol. 2, No. 20, pp. 120506-1–120506-9, December 2011.

[29] Hu, X., Wu, L., Hu, C., and Gao, H., "Adaptive sliding mode tracking control for a flexible air-breathing hypersonic vehicle", *Journal of the Franklin Institute*, Vol. 349, Issue. 2, pp. 559–577, Marsh 2012.

[30] Kumar, J., Azar, A. T., Kumar. V., and Rana K. P. S., "Design of Fractional Order Fuzzy Sliding Mode Controller for Nonlinear Complex Systems", Chapter 9, Advances in Nonlinear Dynamics and Chaos (ANDC) Elsevier, pp. 249–282, 2018.

[31] Panda, S., and Padhy, N, P., "Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design", *Elsevier Applied Soft Computing*, Vol. 8, Issue. 4, pp. 1418–1427, September 2008.

[32] Laminia, C., Benhlimab, S., and Elbekria, A., "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning", *Journal of Procedia Computer Science Elsevier*, Vol. 127, pp. 180–189, 2018.

[33] Khan, A. A., and Mir, R. N., "Optimization of Constrained Function Using Genetic Algorithm", *Journal of Computer Engineering and Intelligent System*, Vol. 8, No. 2, pp. 11–15, 2017.

[34] Deb, K., "An Efficient Constraint Handling Method for Genetic Algorithm. Computer Methods in Applied Mechanics and Engineering", *Journal of Computer Methods in Applied Mechanics and Engineering Elsevier*, Vol. 186, Issue 2–4, pp. 311–338, June 2000.

[35] Kramer, O., "A Review of Constraint-Handling Techniques for Evolution Strategies", *Journal of Applied Computational Intelligence and Soft Computing*, Vol. 2010, pp. 1–11, January 2010.

[36] Yanzhi, R., and Sanyang, L., "Modified particle swarm optimization algorithm for engineering structural optimization problem", in *Proc. of 13th International Conference on Computational Intelligence and Security*, 2017, pp. 504–507.

[37] Satpati, B., Koley,C., and Datta, S., "Robust PID controller design using particle swarm optimization-enabled automated quantitative feedback theory approach for a first-order lag system with minimal dead time", *Journal of Systems Science & Control Engineering*, Vol. 2, Issue 1, pp. 502–511, April 2014.

[38] Juneja, M., and Nagar, S.K., "Particle swarm optimization algorithm and its parameters: A review", *in Proc. of International Conference on Control. Computing Communication and Materials (ICCCCM)*, 2016, pp. 1–5.

Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 10 (2018) 67-76

DOI: 10.2478/auseme-2018-0004

A Bayesian Information System for Predicting Stator Faults in Induction Machines

Ahmed RAMDANE[1], Abdelaziz LAKEHAL[2],
Ridha KELAIAIA[1], Salah SAAD[3]

[1] Faculty of Technology, Université 20 août 1955-Skikda, PB N°26 Route Elhadaik, Skikda, 21000, Algeria, email: r.kelaiaia@univ-skikda.dz
[2] Department of Mechanical Engineering, Mohamed Chérif Messaadia University, P.O. Box 1553, Souk-Ahras, 41000, Algeria, email: lakehal21@yahoo.fr
[3] Laboratoire Systèmes Electromécaniques (LSELM), University of Badji-Mokhtar Annaba, 23000, Algeria

Manuscript received October 18, 2018; revised December 10, 2018.

Abstract: The approach adopted in this paper focuses on the faults prediction in asynchronous machines. The main goal is to explore interesting information regarding the diagnosis and prediction of electrical machines failures by the use of a Bayesian graphical model. The Bayesian forecasting model developed in this paper provides a posteriori probability for faults in each hierarchical level related to the breakdowns process. It has the advantage that it can give needed information's for maintenance planning. A real industrial case study is presented in which the maintenance staff expertise has been used to identify the structure of the Bayesian network and completed by the parameters definition of the Bayesian network using historical file data of an induction motor. The robustness of the proposed methodology has also been tested. The results showed that the Bayesian network can be used for safety, reliability and planning applications.

Keywords: Induction machine, faults, Bayesian network, Stator, Maintenance

1. Introduction

In a concept evolution, corrective maintenance has become a disadvantage with regard to preventive maintenance. This preventive maintenance must make it possible to avoid faults of equipment in use. The cost analysis must highlight a gain in relation to the faults that it avoids. To implement this type of maintenance on vital electric motors, monitoring systems have been put in place [1]. Through these systems, preventive visits make it possible to accumulate information relating to the behaviour of the motor. The majority of vital electric motors are monitored in real time. Conditional based maintenance of large

electric motors is therefore maintenance depending on experience and involving information collected in real time. Conditional based maintenance is characterized by the highlighting of the weak points of the induction motor. Depending on the case, it is desirable to put them under surveillance and, from there, to decide on an intervention when a certain threshold is reached.

Indicators are initially defined for monitoring the induction machine. These indicators depend mainly on the type of constraint. Besides electrical constraints, thermal stresses, mechanical stresses and environmental conditions, improper operation can affect the life of the induction motor. The most important parts of the motor to be inspected are: the stator windings, the rotor winding, and the mechanical part of the rotor. The vibration analysis, the analysis of the signature of the electric current, and the acoustic analysis are mainly the techniques used for the monitoring and the diagnosis of induction motors. In recent decades these techniques have been improved by the use of artificial intelligence methods. This hybridization of techniques has shown promising results and made the conditional maintenance of induction motors more efficient.

Adaptive Neuro-Fuzzy Inference System (ANFIS) have been used to analyse vibration signals of a faulty induction motor [2]. In this study, the correlation between the motor fault types and their corresponding characteristic frequency spectra using the Adaptive Neuro-Fuzzy Inference System show that the developed system is very performing and robust for fault diagnosis. In another research work [3], the technical orbits Park was implementing, strengthened by the application of the Fourier transforms to the Park vector of the stator current allowed the identification of the unbalance defect at low frequency. Other researchers have used both vibration and current signature for fault prediction in the induction motor. In [4] it was presented a contribution in which multiclass support vector machine (MSVM) algorithms have been trained at various operating conditions using the radial basis function kernel and tested for the same operating conditions. The obtained results are in the form of percentage fault prediction. Other works have been based on the acoustic signature for the isolation, diagnosis and early detection of electrical and mechanical faults [5, 6, 7].

Recent contributions have been developed in the last years based on probabilistic and statistical analysis. Fuzzy algorithm-based induction motor fault diagnosis systems have been designed in [8]. The main objective of this algorithm is to evaluate the probability of various types of motor faults. In a similar contribution [9], the authors tried to show how to construct the hierarchical fuzzy inference nets with the propagation of probabilities concerning the uncertainty of faults. A probabilistic model for induction machines was developed by [10]. Based on measuring the current of a healthy induction motor a recursive probability density estimation algorithm was

proposed for real-time fault detection. All the models developed and the techniques used are specific to certain specific situations. It also does not allow the detection and diagnosis of all faults at the same time and give information on isolated faults. The diagnosis is a complex procedure especially in the case of simultaneous faults. However, the only way to better diagnose the induction motor is to exploit feedback and information's about fault occurrence in the equipment.

This paper deals with the fault analysis of a vital induction motor. The main objective of the work was to investigate stator element problems in the high power motor. The proposed method relies on two main phases: The first phase is the definition of causal links between causes and consequences. This will allow defining the probabilities of each fault in the induction machines. The second phase of the study concerns the definition and the implementation of an appropriate corrective maintenance program to combat the issue.

The paper is organized as follows. Section 2 is dedicated to present the Bayesian approach. In section 3 we describe steps to build a Bayesian graphical model. Before ending by giving some conclusions in section 5, section 4 is devoted to the application of the developed Bayesian network in a real case study.

2. The Bayesian approach

Probabilistic inference has been widely used in recent years to solve a variety of problems. One of the best-known techniques is Bayesian inference. It is used in diagnosis and prediction initially in the field of medicine and is currently widely used in the industrial field. Bayesian reasoning is the basis for solving reasoning problems under uncertainty and in the presence of incomplete information. These tools have also shown a great interest in the field of maintenance, reliability and safety.

Bayesian networks are graphical models, called also causal networks or probabilistic networks. They combine graph theory and probability theory [11]. To solve problems related to diagnosis we will give preference to the Bayesian networks since they are simple and easily readable and understandable by a non-specialist. The second advantage is the use of new information. In the presence of new information, the Bayesian network takes over the calculation of the branch concerned by this new information only. This is something that makes Bayesian network easy to calculate and therefore minimizes errors. Another advantage of Bayesian networks is the ability to make inference for diagnosis and prediction from the same model. A Bayesian network is a compact tool that allows conducting at a time, quantitative and qualitative fault analysis.

Bayesian network consists mainly of nodes and a set of edges. Discrete or continuous random variables are modelled by the nodes. The edges define the causal relationships between the variables. Inference in a Bayesian network represents the calculation of probabilities. Two types of nodes exist in Bayesian graphical models: parent nodes and children. For child nodes without parents the a priori probabilities are defined by the modeller, while for the parent nodes the probabilities are conditional and they are calculated by the Bayes theorem given by the formula (1).

$$P(\mathrm{A/B}) = \frac{\mathrm{P(A).P(B/A)}}{\mathrm{P(B)}} \tag{1}$$

Example:

The measurement of vibration and electric current are two means of controlling the state of induction motors widely used in conditional based maintenance. In this example, and in a maintenance workshop, the expert made the following observations in case of similar induction motors, which work in the same conditions:

- If an induction motor has a current consumption higher than the rated current (electrical fault), then for 2 motors out of 5 there are reported abnormal vibrations;
- If a motor does not show electrical fault, then 4 motors out of 5 do not show any vibration.
- Half of the motors from the park present electrical fault.

Calculate the probability that, if a motor has a vibration, then it also presents electrical fault?

By applying formula (1):

$$P(B) = P(A) \times P(B/A) + P(\bar{A}) \times P(B/\bar{A}) = \frac{1}{2} \times \frac{2}{5} + \frac{1}{2} \times \frac{1}{5} = \frac{3}{10}$$

$$P(A/B) = \frac{P(A) \times P(B/A)}{P(B)} = \frac{\frac{1}{2} \times \frac{2}{5}}{\frac{3}{10}} = \frac{2}{3}$$

A posteriori probabilities of the causes: The 2/3 of the motors presenting vibration, presents also electrical fault. This result denotes a clear impact of the electrical fault on the evolution of the vibration in the engine. From uncertain to certain environment the expert can make decision that the vibration in the motor is the consequence of an electrical fault.

Bayesian network is an acyclic oriented graph. It prohibits dependencies children nodes towards parent nodes. According to the conditional independence and the chain rule, BNs represent the joint probability distribution P(V1,

V2,…, V_n) where C(Vi) are the parents of Vi or causes of Vi in the Bayesian network. In a general context, a network is called Bayesian if it realizes Markov factorization condition. So a Bayesian network is defined by:

$$P(\mathrm{V1}, \mathrm{V2}, \cdots, V_n) = \prod_{i=1}^{n} P(V_i \ / CV_i)) \quad (2)$$

3. Bayesian graphical model design

For a long time, researchers started using real-time maintenance monitoring data, collected from sensors, for predictive maintenance and health monitoring of rotating machines. However, there is a lack of prior studies that investigated the forecasting of faults probability. In this paper, we studied three phase induction machines which are widely used in industry. Some issues have not explicitly been addressed in prior studies such as: the possibility of fault predicting in induction machine using the historical file of the machine and data collected from trend curves given by the automatic monitoring system. Another issue is the opinion of the maintenance expert regarding the faults which can occur, and their relationships. The objective of the Bayesian approach is to optimize the availability and avoid accidental breakdowns. We aim to minimize the probability of fault or any incident that affects the reliability of the machine.

As discussed in the previous section, a Bayesian network consists of a structure and parameters. To construct the structure of the Bayesian network it is often necessary to call on an expert. For our case, a bibliographic investigation allowed us to construct the structure of the network given in *Fig. 1*. The logic is to look for the causal links between the basic events representing in general the causes and the intermediate events representing the consequences. It should be noted here that the hierarchical level of the network structure reflects the degree of fault analysis and fault correction means. The consequence of a cause can itself be a cause for another consequence. Increasing the hierarchy of the scenarios of the breakdown gives more precision to the diagnosis. The top event represents in general the fault of the element concerned by the study, for our case study, the stator.

Bayesian network parameters represent a priori probabilities and conditional probabilities. The exploitation of the experience feedback by the use of the information recorded in the historical files of the motor are at the base of the definition of the parameters of the Bayesian network. For conditional probabilities the causality is strict which means that the presence of a cause inevitably leads to the fault appearance. Also, it means that each conditional probability is equal to 1.

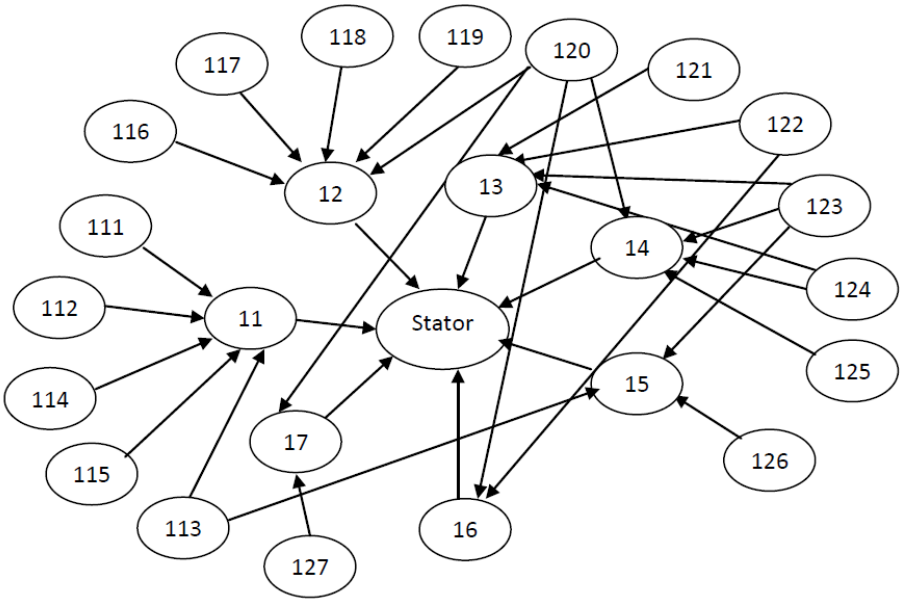


Figure 1: The Bayesian network modelling the fault prediction of stator element.

For causes which represent the child variables in the Bayesian network of *Fig. 1*, a priori probabilities are given. For the consequences which represent the parent variables the probabilities are conditional and they are calculated by inference in the Bayesian network. For the stator fault analysis of the induction motor, it is possible to define 17 probable causes, and 7 intermediate consequences that can cause faults in the stator of the induction motor. The causes are: unbalanced magnetic pull, winding motion, unbalanced power supply, overloading, rotor strike, crushing of the turn by the carcass, thermal cycling, an abrasion of the insulation, laminations slack slot wedges, shock or vibration, damage to insulation during insertion of windings, frequent starting, extreme temperature, extreme humidity, overvoltage, slacking of coils, and slack joints. The consequences are: vibration, stator carcass fault, insulation fault, stator turn-turn faults, stator phase-phase faults, displacement of conductors, and connectors failure.

4. Model application

The induction motor studied in this article is three-phase. The technical characteristics of this device are given by the nameplate of *Table 1*. The role of the induction machine is very important and its availability is essential. It is a motor used by a company involved in the field of petroleum engineering.

Table 1: The nameplate of the motor.

380 volts three-phase induction motor	
Type	M3KP280SMB4B3
Power	90 kW
Rated load current	166 A
Speed	1491 rot/min
Frequency	50Hz
COS Φ	0.87
Insulation Class	F
Tmax	40°C
3 PHASE	Delta connecte
The diameter of the end of the motor shaft	75mm.
Motor shaft length	140mm

In order to exploit the experience feedback and to obtain a permanent update of the predictive maintenance plan of the induction motor, the priorities will be defined according to the determination of the most likely faults on the stator of the machine. For reliable prediction of faults, the starting information (model inputs) must be accurate. A priori probabilities given in *Table 2*. are defined on the basis of the factual information, on the one hand, and the more or less complex measurement results recorded in the historical files, on the other hand.

Table 2: The inputs of the model or a priori probabilities.

Causes	Code	A priori probabilities
Unbalanced magnetic pull	111	0.088
Winding motion	112	0.028
Unbalanced power supply	113	0.1
Over loading	114	0.001
Rotor strike	115	0.001
Crushing of the turn by the carcass	116	0.023
Thermal cycling	117	0.001
An abrasion of the insulation	118	0.079
Laminations slack slot wedges	119	0.001
Shock or vibration	120	0.045
Damage to insulation during insertion of windings	121	0.001
Frequent starting	122	0.075
Extreme temperature	123	0.001
Extreme Humidity	124	0.045
Overvoltage	125	0.001
Slacking of coils	126	0.001
Slack joints	127	0.001

In the Bayesian approach used in this work, the relation between causes and effects is given by a conditional probability table. Example of the conditional probability table of the consequence 17 which represents Connectors Failure is given by *Table 3*.

Table 3: Conditional probability table for variable "Connectors Failure".

	120	True		False	
	127	True	False	True	False
17	True	1	1	1	0
	False	0	0	0	1

It is possible to update information by inference in the Bayesian network of *Fig. 1*. The a posteriori probabilities computed from the Bayesian network are given in *Table 4*.

An example of calculation is given for the variable "Connectors Failure" as follows:

P (17= True) =
P (17= True/120= True, 127= True) × P (120= True) × P (127= True) +
P (17= True/120= True, 127= False) × P (120= True) × P (127= False) +
P (17= True/120= False, 127= True) × P (120= False) × P (127= True) +
P (17= True/120= False, 127= False) × P (120= False) × P (127= False)

P (17= True) = (1 × 0.045 × 0.001) + (1 × 0.045 × 0.999) + (1 × 0.955× 0.001) + (0 × 0.955 × 0.999)

P (17= True) = 0.000045 + 0.044955 + 0.000955 + 0 = **0.045955**

Table 4: The outputs of the model or a posteriori probabilities.

Element	Fault	Code	A posteriori probabilities
Stator		1	0,39925971
	Vibration	11	0,20377724
	Stator carcass fault	12	0,14239256
	Insulation fault	13	0,11759662
	Stator turn-turn faults	14	0,089798138
	Stator phase-phase faults	15	0,1017991
	Displacement of conductors	16	0,116625
	Connectors Failure	17	0,045955

From these a posteriori probabilities, it is possible to make decisions on the corrective actions to be taken in a certain environment. These diagnostic results

also provide the ability to organize actions in order of priority. It should be noted that the probability of having a defect in the stator is 0.39%, which is significant and the main fault in descending order are respectively: Vibration, Stator carcass fault, Insulation fault, Displacement of conductors, Stator phase-phase faults, Stator turn-turn faults, and finally Connectors Failure.

5. Conclusion

In this work a Bayesian network model has been introduced as a mean to enable maintenance staff to assess the probability of faults and the priorities of corrective actions. By this, the company can consider a first step to incorporate an Information System into their supervision system and thus actively contribute to a reliable monitoring system. The integration of the developed Bayesian information system might have a significant impact on the existing supervision system and might thus lead to increase the availability of vital machines. Also, the integration of the Bayesian information system minimizes the costs related to accidental damages and other inconveniences caused by these damages. This paper also shows a concept evolution for predictive maintenance. In addition to decision making support, it is possible to predict and anticipate faults rather than real-time detection. It is found that faults could be identified correctly and decisions could be made with certainty.

The model structure is standard for all induction motors. The parameters are specific for each machine. However, a weighting coefficient is necessary for the definition of intervention priorities and in order to make the presented approach more reliable. This coefficient must take into consideration the criticality of the machine. A FMECA (Failure Mode Effects and Criticality Analysis) study can be used to define this coefficient.

We are going to address this approach in a future paper which will demonstrate the usefulness of the Bayesian methodology in rotor fault prediction of vital induction motors. Such an extension of this research is important for ensuring adequate assessment of the probability of each fault, whilst taking into account the influence of other monitoring parameter on the a posteriori probability evaluation.

References

[1] Gill, P. “Electrical power equipment maintenance and testing”, 2nd ed. Taylor & Francis Group, LLC., 2009.

[2] Moghadasian, M., Shakouhi, S. M., and Moosavi, S. S. “Induction motor fault diagnosis using ANFIS based on vibration signal spectrum analysis”, in *Proc. 3rd International Conference on Frontiers of Signal Processing* (*ICFSP*). *Paris, France*, 2017.

[3] Bouras, A. Bouras, S. and Kerfali, S. "Prediction of the mass unbalance of a variable speed induction motor by stator current multiple approaches", *Turkish Journal of Electrical Engineering & Computer Sciences.*, vol. 26, pp. 1056–1068, 2017.

[4] Gangsar, P., and Tiwari, R., "Comparative investigation of vibration and current monitoring for prediction of mechanical and electrical faults in induction motor based on multiclass-support vector machine algorithms", *Mechanical Systems and Signal Processing*, vol. 94, pp. 464–481, 2017.

[5] Akçay, H., Germen, E., and Turkay, S. "Induction Motor Identification from Acoustic Noise Spectrum by a Covariance Subspace Algorithm", in *Proc. IEEE 14th International Conference on Control and Automation (ICCA). Anchorage, AK, USA*, 2018.

[6] Glowacz, A. "Acoustic based fault diagnosis of three-phase induction motor", *Applied Acoustics*, vol. 137, pp. 82–89, 2018.

[7] Hassan, O.E., Amer, M., Abdelsalam, A.K., and Williams, B.W. "Induction motor broken rotor bar fault detection techniques based on fault signature analysis – a review", *IET Electric Power Applications*, vol. 12, no. 7, pp. 895–907, 2018.

[8] Chang, H., Kuo, C., Hsueh, Y., Wang, Y., and Hsieh, C. "Fuzzy-based fault diagnosis system for induction motors on smart grid structures", in *Proc. IEEE International Conference on Smart Energy Grid Engineering (SEGE). Oshawa, ON, Canada*, 2017.

[9] Dong, M., Cheang, T., Booma, D.S., and Chan, S. "Fuzzy-expert diagnostics for detecting and locating internal faults in three phase induction motors", *Tsinghua Science and Technology*, vol. 13, no. 6, pp. 817–822, 2008.

[10] Cho, H.C., Kim, K. S., Song, C.H., Lee, Y.J., and Lee. K.S. "Online fault detection and diagnosis algorithm based on probabilistic model for induction machines", in Proc. 2008 SICE Annual Conference. Tokyo, Japan, 2008.

[11] Darwiche, A. "Modeling and reasoning with Bayesian networks", Cambridge University Press, 2009.

Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 10 (2018) 77-89

DOI: 10.2478/auseme-2018-0005

The Demonstration Model of an Electromagnetic Accelerator Gun

Andras MOLNAR

John von Neumann Faculty of Informatics, Óbuda University, Budapest,
e-mail: molnar@uni-obuda.hu

Manuscript received September 19, 2018; revised December 07, 2018.

Abstract: These days a lot can be heard about special weapons which accelerate the projectile not based on the traditional, chemical energy release, but providing the muzzle velocity with the help of electromagnets. In English terminology, many descriptions can be read about these devices, referred to as "coilgun". There are so many hobbyist and amateurs who make these devices [1,2] and publish their results on the internet [3,4].

The purpose of the project is dual. On one hand, features, advantages, disadvantages and the limits of the electromagnetically accelerated weapons can be found by building an experimental tool. On the other hand, it was intended to point out the fact that anybody can build such a tool using commercially available commercial components.

Although the muzzle energy of the device presented in this paper is not more than 6.8J, but it can cause serious injury. The paper also points out that in a similar way, still not using special components, a weapon can be made with a larger (10-20J) muzzle energy.

Keywords: Coilgun, Electromagnetic accelerator

1. The theoretical structure of the device

In principle, an electromagnetic accelerator is an extremely simple construction (*Fig. 1*). The magnetizable projectile is attracted by a magnetic field created around the electrified coil. The trajectory of the projectile is ensured by a thin-walled barrel.

The electric current flowing through the coil must be timed in such manner that there must not be a magnetic field around the coil during the exit of the projectile which would slow it down. The magnitude of the magnetic induction created by the coil is proportional to the intensity of the electric current flowing through the coil (1), which ultimately determines the accelerating force affecting the projectile. In order to calculate the magnetic induction "B" generated by the

coil (solenoid) the following relation can be applied based on the 4th laws of Maxwell and Biot-Savart law (1).

Therefore, by increasing the number of turns as well as by reducing the length of the coil, magnetic induction is increased, but the creation of a short impulse demands the modification of these parameters in the opposite direction, in other words a decrease. Accordingly, the basic way to increase the magnetic induction is to increase the current through the coil. (1) The formula is correct for solenoid only, but it is enough to the analyze the effects of individual members.

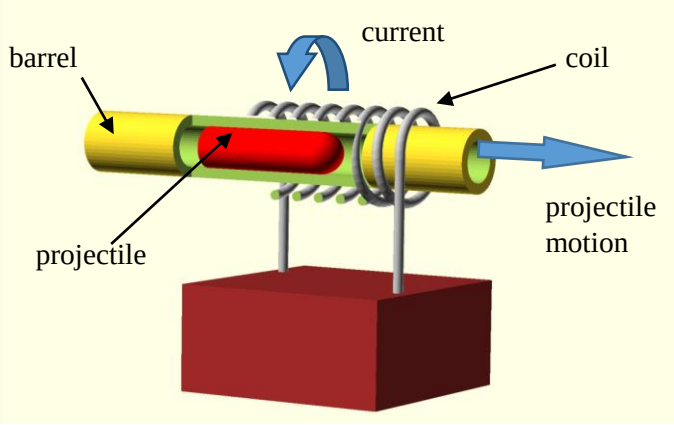


Figure 1: The structure of an electromagnetic accelerator

$$B = \mu_0 \frac{I*N}{l} \tag{1}$$

where

B is the magnitude of the magnetic field's induction generated by the coil [T]

I is the intensity of the current flowing in the coil [A]

N is the number of turns in the coil [1]

l is the length of the coil [m]

μ_0 is the magnetic constant [$4\pi * 10^{-7} \frac{H}{m}$]

In order for an even higher current to flow through the coil, firstly, it must be powered by a power source which has very low internal resistance. The most suitable option for this is the capacitor. Accordingly, the launcher is composed of a coil and a capacitor connected in parallel. In order to achieve a correct synchronization of the current impulse, a fast switching device (such is fast thyristor or spark switch [5]) must be used to connect the charged capacitor to the coil.

In this case, the length of the impulse generated during the discharge depends on capacity and inductance values (from a practical standpoint the ohmic resistance of the closed circuit must also be taken into consideration). For the sake of a fast discharge it is advisable to use some type of a semiconductor switching device instead of a mechanical switching device. Thyristor is rather widely utilized for this purpose. At dimensioning, it must be taken into consideration that during the discharge, though only for a few milliseconds, thousands (!) of amperes flow through the coil.

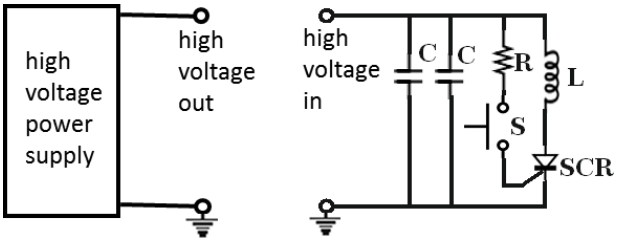


Figure 2: The theoretical circuit diagram of the electromagnetic accelerator.

The operation of the accelerator shown in *Fig. 2* is the following. The "C" capacitors are charged to working voltage using an external power source. The maximum voltage of the charging power supply affects charging time only, otherwise it has no influence on the performance of the launcher. The thyristor marked "SCR" does not conduct during recharging, thus current does not flow through the work coil marked "L". The charged capacitors discharge through the coil by pushing the snap-switch marked "S".

This is the moment when firing takes place. Resistance "R" provides the necessary starting voltage knowing the required voltage of the applied thyristor and the capacitors. When choosing the type of the thyristor, short current impulses must be taken into consideration, which in this current case can even be thousands of amps. Thyristors can typically even bear an impulse current larger by an order of magnitude compared to the statically permissible maximum current.

If the accelerator contains multiple stages, those have a structure similar to *Fig. 2* but thyristor firing is not managed by a push-button but by the signal of some type of sensor. The easiest method to start the stage is to use an optical transmitter. When the projectile crosses the light of the transmitter, the photo-sensor goes into "shadow".

This signal must be connected to the controlling electrode of the switch, which starts the stage connected to it. At the scaling of further stages, the velocity of the projectile approaching the coil must be taken into consideration. Generally speaking, the faster the projectile entering the coil is, the shorter the impulse

needs to be for acceleration. In order for acceleration to be effective the short impulse (in case the same electric power and same voltage) must be accompanied by a higher current (2).

$$W = P * t = U * I * t \tag{2}$$

where

W: work [J]
P: power [W]
U: voltage [V]
I: current [A]
t: time [s]

The design of the three-staged electromagnetic accelerator can be seen in *Fig. 3*. The first stage can be started using a switch as it was introduced previously. As a result, the projectile accelerates and proceeds inside the barrel towards the second stage. The barrel was drilled before the second stage for the light of the light source (marked with a red LED in the figure) placed below to reach the light sensor placed above it (a photo diode marked with blue color in the figure) through the two drilled holes.

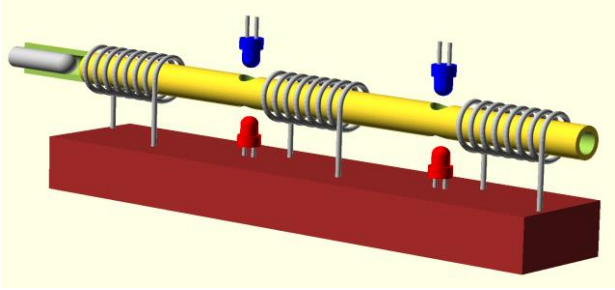


Figure 3: The design of the three-staged electronic accelerator.

The electronic impulse generated by the interruption of the way of light starts the stage connected to it which further accelerates the projectile. The distance of the optical transmitter from the coil controlled by it is a relevant parameter. When determining the distance, the switching moment of the circuit and its entry speed must be taken into consideration. In case if a too large distance, the magnetic field generated by the coil does not have a sufficient accelerating force of a substantial degree on the projectile. In case of a too short distance, it may occur that the electric current is still flowing through the coil when the projectile moves away from it, and the not yet vanished magnetic field will have a braking effect on it. The operation of the third stage is similar to the operation of the second stage that we have just shown.

Fig. 4 demonstrates the three-staged accelerator. The three stage are coupled by the projectile. The system is practically composed of three individual units. The only common point of the units is their ground which ensures the simpler connection of the measurement devices connected to it (not visible in the figure). Every unit has its own capacitor battery and the units charging them individually (not visible in the figure).

During practical implementation, the charging circuits of the flashes, utilized for photography, were used to charge the capacitors. These circuits are designed in such a way that they are protected from the transient effects of the discharge following the charging of the capacitor connected to them, thus during their use no special attention must be paid to this.

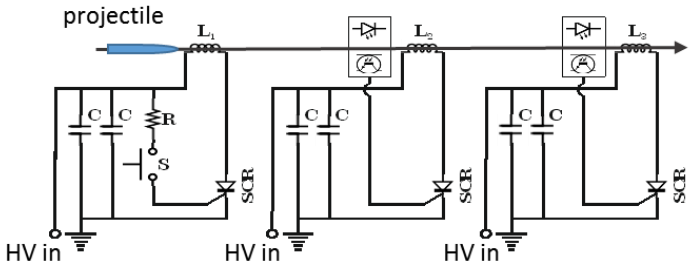


Figure 4: The theoretical circuit diagram of the three-staged electromagnetic accelerator.

2. The electrodynamic and kinetic simulation of the device

The acceleration of the projectile depends on several factors. In order to produce a relatively efficient device it is advisable to conduct calculations for the sake of determining the parameters of the energy banks (capacitors), the accelerating electromagnets (coils), the voltage supply, the size of the projectile and other devices. As the calculations are rather complex and the simultaneous effect of several variants influence the end result it is advisable to employ a simulation tool. During the project, the design of the device to be realized was made on the basis of the results of a program available free of charge [6]. The simulator has several "classical" calculation modules and a special kinetic module.

For the sake of facilitating handling it is sufficient to enter the main physical dimensions of the coil which the program uses to calculate the number of wire layers, the wire length to be used, the ohmic resistance and inductance. The second module conducts an LRC circuit simulation from the data from received the previous module, as well as from the value of the set capacity and from the

value of the electric current, which shows voltage and current flow through the coil for a few milliseconds after switch-on (the time range of the simulation depends on user settings).

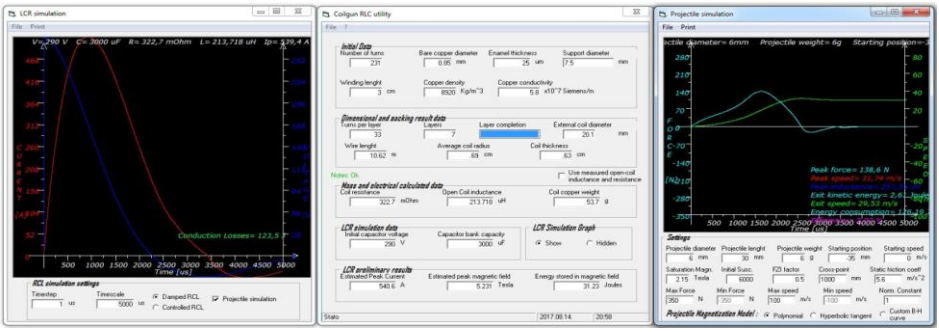


Figure 5: The simulation result of stage I.

Fig. 5 shows the simulation results of the first stage. The parameters of the simulated first stage are the following (*Table 1*).

Table 1: Initial parameters of the first stage

coil length:	30 mm,
internal diameter of the coil:	7,5 mm,
number of turns in the coil:	231,
wire diameter:	0,85 mm,
capacitor capacity:	3000 μF,
working voltage:	290 V
projectile length:	30 mm,
projectile diameter:	6,5 mm
projectile mass:	6 g
starting velocity of the projectile:	0 m/s

The third larger module of the simulator software calculates and displays (*Fig. 5* right-side screen shot) the motion data of the projectile. In this module it is possible to enter the initial velocity of the projectile, in this way simulating the kinematics of the second and the third stage. On the website of the software developer it can be read that the simulation contains various simplifications and conditions. Based on the tests of the developers the simulation results of the software closely approximate the results given by professional calculation models [7, 8], in this way the simulator, based on its recommendation, is not accurate but it provides good help during the design of an electromagnetic accelerator gun for hobby purposes.

The current- and voltage-time diagram of the coil of stage I can be seen on the left side of *Fig. 5*. The left-side figure displays the velocity of the projectile in relation to time. The muzzle velocity of the projectile can be read from the figure (as the program conducts only a one-staged simulation). In the present case this value is around 27 m/s. This velocity value will be the initial velocity of the projectile at the simulation of the next stage.

It is worth mentioning that in case of leaving all parameters unchanged the accelerator ensures only 27 m/s exit velocity in case of a projectile with 0 m/s starting speed. If the second stage matches the first one in every parameter, it will slow down the projectile arriving at 27 m/s speed instead of accelerating it (*Fig. 6*). It can be seen in *Fig. 6* that the entry speed of the projectile is 29 m/s, and at the exit of this the stage it is only 23 m/s.

The reason for this is that the discharge time of stage I is too long, and in this way the magnetic field "pulls back" the projectile. Discharge time can be reduced if the capacity of the capacitor bank is reduced, but then the energy released during the discharge will be smaller. This is the reason why it is more advisable to reduce its inductance by reducing the turn number of the coil. Another effect of reducing the turn number is that discharge current will increase, thus when choosing the thyristor suitable for the stage this must be taken into consideration.

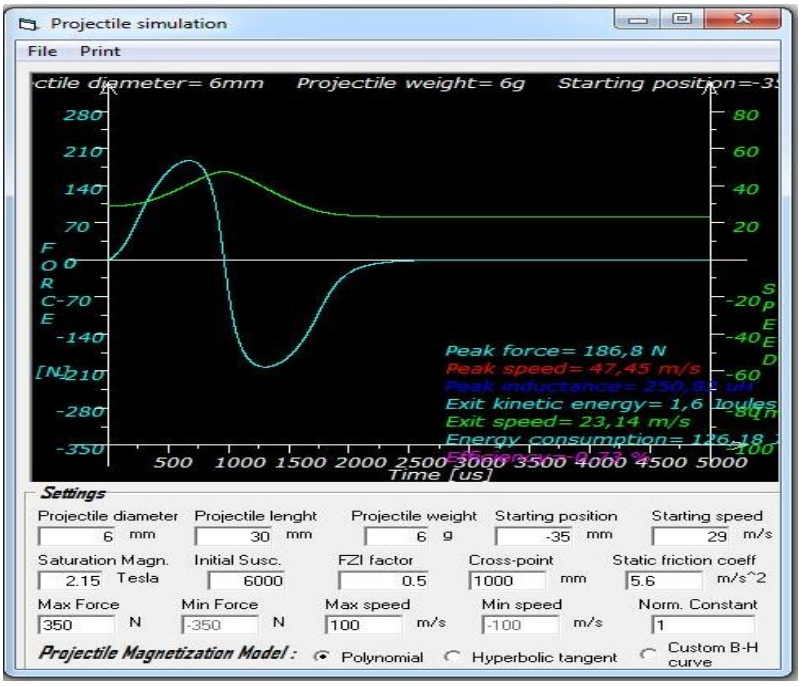


Figure 6: The slowing effect of stage I in case of the projectile with 29 m/s entry speed.

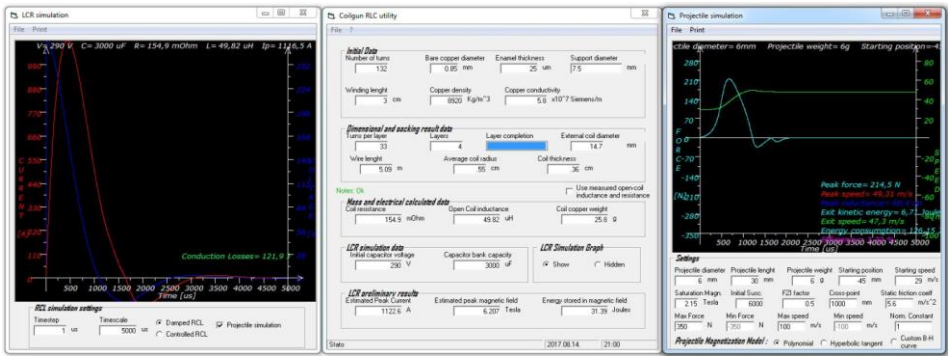


Figure 7: The simulation result of stage II.

Fig. 7 shows the simulation of stage II. The velocity of the projectile arriving to the stage is the exit velocity resulting from the previous simulation.
As the data of the projectile are obviously unchanged, except velocity, the internal diameter of the coil does not change either, these are not displayed in the following list of parameters of the second stage (*Table 2*).

Table 2: Parameters of the stage II.

coil length:	30 mm,
coil turn number:	132,
wire diameter:	0,85 mm,
capacitor capacity:	3000 μF,
working voltage:	290 V
starting velocity of the projectile:	29 m/s

Figure 8: The simulation result of stage III.

It can be seen from the simulation that whereas in case of the first stage the discharge peak current is 540 A, in case of the second stage this is now 1122 A. Leaving the stage, the velocity of the projectile reaches 47 m/s.

It can be read from *Fig. 8* that the maximum velocity of the projectile is more than 61 m/s which in case of the given projectile means more than 11 J muzzle energy. The parameters of stage III. are the following (*Table 3*).

Table 3: Initial parameters of the stage III.

coil length:	30 mm,
coil turn number:	99,
wire diameter:	0,85 mm,
capacitor capacity:	3000 μF,
working voltage:	290 V
starting velocity of the projectile:	47 m/s

A frame construction is needed to build the structure determined using simulations, as the position of the coils and their transmitters must be kept accurate. During the simulation it was clearly revealed that the accelerator is very sensitive to projectile parameters, furthermore, to synchronization of the triggering of the stages with the position of the projectile. As the failure of the mechanical construction may thwart the operation of the device, it is advised to build sufficiently stable mechanics, in other words one which keeps the position of the parts stable. Sufficiently flexible, easily plannable and executable mechanics can be built by using 3D printing. Furthermore, the procedure is sufficiently economical too.

4. Experimental calculations, experiments

Calculations were conducted on our experimental device to observe whether the previously calculated results and the nature of the temporal changes corrolate with each other. As a first step, we examined whether the variation in time of the coil voltage corresponds to the expectations. By conducting the measurements using a multi-channel oscilloscope with digital storage, we received the oscillogram corresponding to *Fig. 14*.

The minimal deviations of the initial DC voltages, seen in the figure, can be experienced due to the slightly differing settings of the independent charging circuits in certain coils. It is clearly visible that stage by stage the voltage slopes become steeper and steeper (*Fig. 9*).

The results of the kinematic measurements are demonstrated by *Fig. 9* and *Fig. 10*. The first measurements are directed towards the question whether the given stages indeed further accelerate the projectile. In case of improper settings, indeed, it occurs often that further stages, because of the wrong timing, pull back

the projectile accelerated by the previous stage and as a result the multi-staged device is weaker than the accelerator composed of one stage.

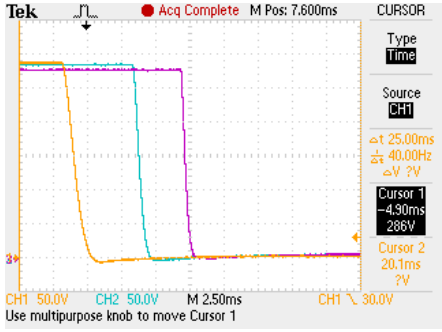


Figure 9: Oscillograms of the three coil voltages during a shot.

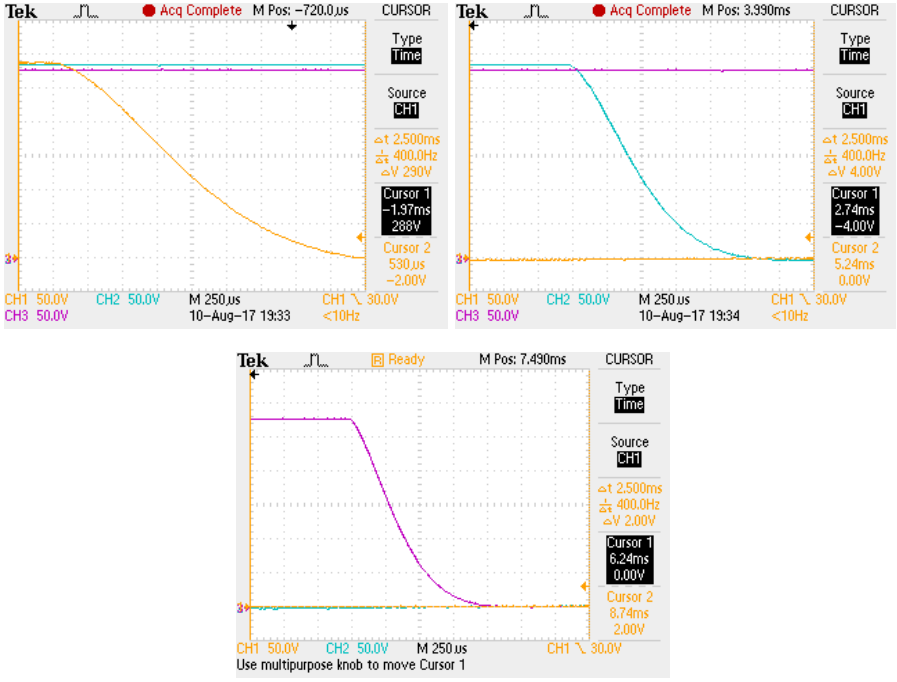


Figure 10: Oscillograms of individual stages during a shot.

Table 4 demonstrates the effect of the given stages. It is clearly visible that both the second and the third stages increase the projectile's velocity. The three stages together resulted in 4.5 J muzzle energy during the experiment which was below simulation results (a value around 11 J). Its reason may come from several

sources. The deviation is partly caused by the inaccurate design of the coils and largely by the timing of the coil magnetization.

Table 4: The variation of muzzle velocity depending on the number of accelerating stages

	Stage I.	Stage I. + II.	Stage I. + II. + III.
1st measurement muzzle velocity [m/s]	23,39	34,2	39,3
2nd measurement muzzle velocity [m/s]	23,51	33,9	39,6
3rd measurement muzzle velocity[m/s]	22,28	35,2	38,4
average muzzle velocity [m/s]	23,06	34,43	39,1
average muzzle energy [m/s]	~ 1,6	~ 3,5	~ 4,6

Table 5: The variant of muzzle energy depending on the mass of the projectile and its starting position

	6 gram projectile		5 gram projectile	
	0 mm starting position	-2 mm starting position	0 mm starting position	-2 mm starting position
1st measurement muzzle velocity [m/s]	39,3	47,0	48,2	52,5
2nd measurement muzzle velocity [m/s]	39,6	45,4	48,4	52,4
3rd measurement muzzle velocity [m/s]	38,4	45,5	48,5	52,4
average muzzle velocity [m/s]	39,1	45,96	48,36	52,43
average muzzle energy [J]	~ 4,6	~ 6,3	~ 5,8	6,8

As the initial positon of the projectile is crucial regarding performance, measurements have been conducted with different starting positions. Apart from the use of all three accelerator coils, *Table 5* demonstrates the effect of the initial position and the effect of the used projectile's mass.

Reducing the mass of the projectile resulted in a larger muzzle energy which occurred due to the improvement of the acceleration's performance. The same projectile slightly pushed inwards into the barrel gathered significant extra energy compared to the one placed at the end of the tube. During several experiments, the muzzle energy of the projectile placed more inwards from the end of the tube proved to be the largest. In this way, using the projectile with 5 g mass 6.8 J

muzzle energy was achieved which is still far behind theoretical, calculated values.

5. Conclusion

As the synchronization time of the coils' magnetization with the position of the projectile is crucial, the theoretical energy level may be approximated by correct triggering of stages II. and III. The device however is not suitable for these stages to be set. In this way it is necessary to build another experimental device to achieve maximum muzzle energy [9]. In this prototype the optical trigger unit is firmly fixed to the barrel. For the new design, the optical trigger unit position must be movable mounted to fine-tune the optimum magnetization moment. In the new gun, measuring the velocity of the projectile must be done before each unit to determine the optimal magnetization time of the coil. In the absence of the speed measurement units in the prototype, the coil magnetization time optimization is not possible.

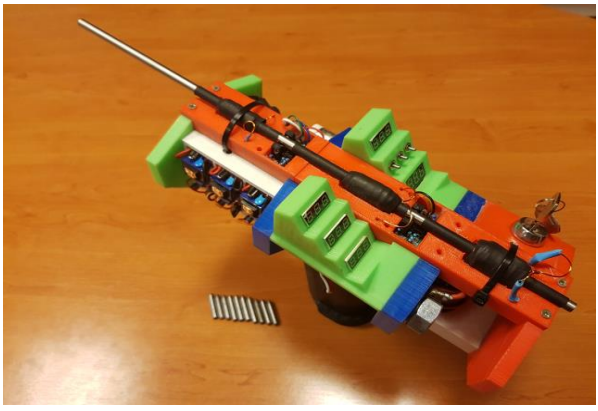


Figure 11: Photo of the created device.

References

[1] Burgess, T. J., Cnare, E. C., W. L. Oberkampf, Beard, S. G., and Cowan, M. "The Electro-magnetic Theta Gun and Tubular Projectiles", IEEE *Trans. on Magnetics*, vol. mag-18, no. 1, Jan. 1982.

[2] Andrews and, J. A., Devine, J. R. "Armature Design for Coaxial Induction Launchers", IEEE *Trans. on Magnetics*, vol. 27, no. 1, Jan. 1991.

[3] http://www.coilgun.eclipse.co.uk/

[4] https://www.coilgun.info/

[5] Kharlov, A. V. "Numerical simulation of spark channel dynamics in railgun switches" Journal of Physics Conference Series 1115:022027, Nov. 2018

[6] http://www.webalice.it/saverio.filippi/coilgun_simulator.html
[7] He, Y. et al. "Efficiency analysis of an electromagnetic railgun with a full circuit model" IEEE *Trans. Plasma Sci*., vol. 38, no. 12, pp. 3425–3428, Dec. 2010
[8] Kulkarni, A. S., Thomas, M. J. "Design of a Compulsator to Drive a Railgun" IEEE *Transactions on Plasma Science*, 2010, vol. 45 no.7, pp. 1482–1488
[9] Engel, T. G., Neri, J. M., and Veracka, M. J. "The maximum theoretical efficiency of constant inductance gradient electromagnetic launchers" IEEE *Trans. Plasma Sci*., vol. 37, no. 4, pp. 608–614, Apr. 2009

Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 10 (2018) 90-101

DOI: 10.2478/auseme-2018-0006

Simulating Solid Tumors with a Microenvironment-Coupled Agent-Based Computational Model

Dániel KISS[1], Anna LOVRICS[2]

[1] Doctoral School of Applied Informatics and Applied Mathematics,
Óbuda University, Budapest, e-mail: kiss.daniel@nik.uni-obuda.hu
[2] Institute of Enzymology, Research Centre for Natural Sciences,
Hungarian Academy of Sciences, Budapest, e-mail: anna.lovrics@gmail.com

Manuscript received July 25, 2018; revised December 15, 2018.

Abstract: In this paper, we introduce a three-dimensional lattice-based computational model in which every lattice point can be occupied by an agent of various types (e.g. cancer cell, blood vessel cell or extracellular matrix). The behavior of agents can be associated to different chemical compounds that obey mass-transfer laws such as diffusion and decay in the surrounding environment. Furthermore, agents are also able to produce and consume chemical compounds. After a detailed description, the capabilities of the model are demonstrated by presenting and discussing a simulation of a biological experiment available in the literature.

Keywords: Agent-based model, On-lattice computational model, Tumor growth characteristics, Hypoxia, Necrotic core formation

1. Introduction

Computational modeling of biological processes such as tumor growth or the onset of drug resistance is a widely researched area. These complex mechanisms are usually explained by both intracellular alterations and interactions with the local environment, however, many aspects of these events show nonlinear behavior, which makes laboratory experiments hard to design and perform. Not mentioning the cost aspects, mathematical and computational models can be efficiently used to pre-test hypotheses of different biological phenomena which cannot be carried out in standard laboratory experiments.

Solid tumors of clinically relevant sizes consist billions of cells. Simulating the evolution of such cell conglomerates is computationally challenging, especially if we wish to follow the faith of each cell one by one. Agent-based modeling is a reliable and frequently used technique to simulate the collective behavior of many particle-like individual objects like biological cells.

Nowadays, one may choose either from many *out-of-the-box* simulation frameworks [1–3] or computational models presented in the literature depending on the purpose. It was shown, that agent-based simulations can be efficiently used to study the properties (e.g growth, pattern formation or death) of a relatively low number of agents [4, 5] as well as emergent behavior of epithelial cell cultures [6], just to mention a few possibilities. However, performing simulations with a large number of cells (e.g. simulating a few mm^3 of tissue) is still computationally challenging. To overcome this problem, models can be tailored for high performance computing architectures [7, 8] or can be simplified significantly to be feasible to run on a standard workstation computer [9].

In this paper, an agent-based computational model of cells coupled to differential equations governing the mass transport of chemical compounds, especially designed for workstations and laptop computers is presented. As a trade-off between the size of the simulated space and computational performance, many features of the agents are greatly simplified. The agent-based part of the model as well as the differential equations are described in Sec. 2. The effect of choosing different parameters is presented in Sec. 3A and 3B. To demonstrate the capabilities of the coupled model, the simulation of tumor hypoxia, a process in which cells inside of a tumor spheroid perish due to the lack of oxygen is also presented in Sec. 3C.

2. Model description

A. On-lattice agent-based model

The model is defined on a three-dimensional regular lattice $L \subset \boldsymbol{Z}^3$ on which sites can be occupied by single agents in a similar way as presented in [9]. It was shown, that this approach highly reduces the computation costs of the simulation by ignoring the mechanical characteristics of the agents without losing the relevance of the model.

The basic unit of the model is an *agent* representing a biological cell. Agent i is defined by its position $\mathbf{x}_i \in L$, genotype $\theta_i \in \Theta$, phenotype $\phi_i \in \Phi$, maturity stage $m_i \in \boldsymbol{Z}_0^+$, duplication time $\tau_i \in \boldsymbol{R}^+$, motility $\nu_i \in \boldsymbol{R}_0^+$ and aggressivity $\alpha_i \in [0,1]$.

Position $\mathbf{x}_i$ defines the site which is fully occupied by i, therefore $\mathbf{x}_i(t) \neq \mathbf{x}_j(t)$ when $i \neq j$ for all $t \in [0, T_{\max}]$. Genotype set Θ defines a collection of possible cell types, such as *healthy cell*, *cancer cell* or *blood vessel cell*. Different agent types adhere to different rule sets, for instance it is possible to define rules that only apply to blood vessel cells. Phenotype ϕ_i defines the current behavior of agent i (e.g. the rate of production of a chemical compound).

Grid spacing l_L defines the fineness of the simulated space and it can be directly related to the volume of a biological cell. This parameter also influences the resolution of the microenvironment (see Sec. 2D).

B. Cell motility

Cell movement is modelled by performing motility trials on each cell (see Sect. 2E for more details). Motility ν_i characterizes the ability of agent i to move towards another site of L. A motility trial performed on agent i consists of

(i) calculating a displacement probability

$$p_i = \frac{\nu_i \cdot \Delta t_m}{l_L}, \tag{1}$$

(ii) randomly selecting a neighboring site $\mathbf{x}_i^{\text{new}}$ of i,

(iii) updating $\mathbf{x}_i = \mathbf{x}_i^{\text{new}}$ with probability p_i only if $\mathbf{x}_i^{\text{new}}$ is an empty site.

Hence agent i with no neighboring sites moves with a characteristic velocity ν_i. Conversely, if all neighboring sites of i are occupied then $\mathbf{x}_i$ does not change (i.e. agent i turns into stationary until at least one of its neighboring site becomes vacant).

C. Cell proliferation

Maturity stage m_i and duplication time τ_i is related to a process called *cell proliferation* during which a biological cell replicates itself periodically. This could be modeled as a single-step transition with probability $\Delta t_p/\tau_i$, which correctly reproduces the mean growth rate of the overall population. However, the inter-replication times of i follow an exponential distribution, which is biologically not plausible as it can be seen in Fig. 1.

To make a simple yet reasonable model of cell duplication we use the multistage concept proposed in [11] by splitting the cell cycle into k independent exponentially distributed stages. Then, the time to progress through these stages is exponentially distributed, therefore, if the transition rates between stages are identical, the cumulated progression time is Erlang distributed with scale parameter τ_i/k and shape parameter k. Fig. 1 shows that experimentally observed cell cycle times (the time difference between successful duplications of a cell) can be better described using this concept. By choosing appropriate values for τ and k various cell types can be described accurately.

In the simulations, when agent i finishes the progression through its cell cycle ($m_i = k$) it undergoes a replication trial as described in the next paragraph.

The aggressivity value α_i influences this process by defining the probability of the following scenarios.

(S1) Agent i may replicate into an empty neighboring site with probability $1 - \alpha_i$.

(S2) Agent i replicates unconditionally with probability α_i by determining the direction which belongs to the minimal number of other agents which are needed to be shifted by one site to make an empty site for replication.

During this trial process, i may be copied to a randomly selected neighboring site. If the selected site is empty or becomes empty after a shifting procedure described as (S2), then a duplicate of i is placed into that site, otherwise the trial is repeated after Δt_p simulation time.

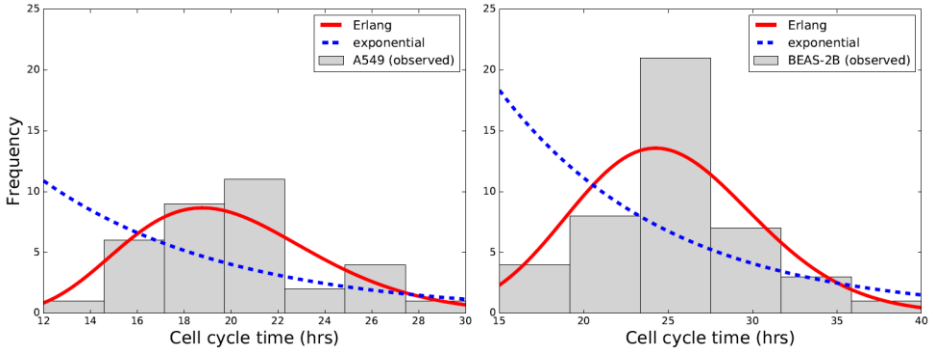


Figure 1: Distribution of experimentally observed duplication times in A549 (left) and BEAS-2B (right) cell cultures reproduced from the data available in [10]. Note the poor agreement between the best-fit exponential distribution ($\lambda_{A549} = \frac{1}{20.1}$ and $\lambda_{\text{BEAS-2B}} = \frac{1}{25.0}$, blue dashed line) and the observed data. Fitting a gamma distribution (or specifically, Erlang distribution, red solid line) describes the data more accurately.

D. Diffusion-decay model

To describe the chemical processes in the environment of biological cells (the so-called *microenvironment*) we numerically solve partial differential equations similar to [12]. Processes including diffusion, decay and concentration modifications caused by agents (e.g. vasculature or agent uptake and secretion) of chemical compound ρ_j are governed by

$$\frac{\partial \rho_j}{\partial t} = \underbrace{D_j \nabla^2 \rho_j - \lambda_j \rho_j}_{\text{diffusion and decay}} + \underbrace{\sum_{\text{agent } i} \mathbf{1}_i(\mathbf{x}) \left[S_j^i \left[\rho_j^* - \rho_j \right] - U_j^i \rho_j \right]}_{\text{agent secretions and uptakes}} \tag{2}$$

on $\Omega \subset \boldsymbol{R}^3$ with no-flux boundary conditions and $\rho_j(\mathbf{x}, t) = f_j$ in Ω. The first term is responsible for the diffusion and the natural decay of compound j, while the second term modifies ρ_j in $\mathbf{x} \in \Omega$ only if $\mathbf{x}$ is overlayed by agent i. Symbol

ρ_j^* denotes the concentration at which compound j saturates. A detailed description of how this PDE model works can be found in [12]. Note, that both coefficients S and U can be chosen arbitrary and independently from i and j, therefore it is possible to define agents that only alter some of the concentrations of compounds in the microenvironment.

E. Time scales and implementation

The simulation of the model is a sequence of calculations and transitions for each $t \in [0, T_{\max}]$. The following time units are used.

(i) Δt: defines the base time step of the simulation. Numerical calculations for mass transport discussed in Sec. 2D are based on Δt. The magnitude of this unit ranges from milliseconds to seconds.

(ii) Δt_m: defines the time step of motility trials. When $t \equiv 0 (\mathrm{mod}\, \Delta t_m)$ a motility trial is performed on all agents as detailed in Sec. 2B. The magnitude of this unit ranges from seconds to minutes.

(iii) Δt_p: defines the time step of proliferation trials. When $t \equiv 0 (\mathrm{mod}\, \Delta t_p)$ a replication trial is performed on all agents that has reached the end of the cell cycle as detailed in Sec. 2C. The magnitude of this unit ranges on a few minutes.

The model was implemented in C# using the standard framework packages. Simulation states were stored regularly on the local disk, and snapshot images were rendered using Mayavi data visualizer [13]. Numerical approximation of the solution was obtained by a finite difference discretization of (2) using a parallelized FTCS explicit method. This generally used scheme provides a good trade-off between accuracy and programming effort [14].

3. Results

A. The effect of motility on growth characteristics

In the absence of chemoattractants or other signals, isolated biological cells have been observed to perform a random walk [15]. This behavior can be easily reproduced by the model detailed in Sec. 2B. However, agent's characteristic velocity v not only influences the movement of the detached agents, but also has an effect on spheroid growth.

Multiple simulations were executed, starting from one agent placed in the center of the simulation space. No chemical compounds were defined, therefore spatially isolated agents did not influence each others' behavior. When an agent

replicated, the descendant inherited all parameter values from its ancestor, hence, the values did not change during the simulation.

Fig. 2 shows representative images of how different values of ν alter the diameter and regularity of a growing spheroid. All relevant parameters of the agents are given in the caption of the figure. As expected, low motility values ($\nu \approx 0$ μmh^{-1}) produce regularly shaped spheroids with sharp surface limits, while high motility ($\nu \gg 0$ μmh^{-1}) results in a more diffuse border and scattered shape. Furthermore, the calculated growth rate of the spheroid also becomes larger due to the increased number of empty sites inside the spheroid.

B. The effect of duplication aggressivity on growth characteristics

Most healthy cells are only able to replicate themselves if there is enough empty space in their surrounding environment. On the contrary, cancer cells are often characterized by the loss of ability to stop proliferation even in a fully occupied environment. To capture this behavior, we introduced the model parameter α which describes the "aggressivity" of an agent during duplication.

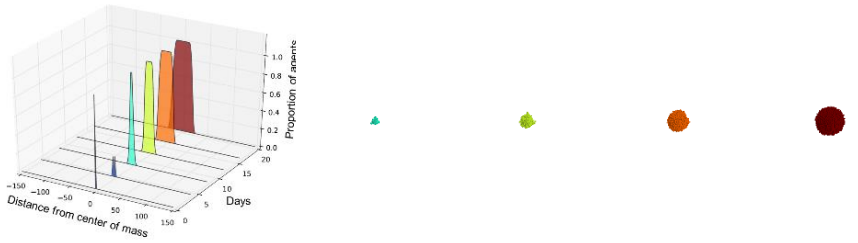
When agent i is marked for replication, it randomly selects a strategy depending on α as described in Sec. 2C. By correctly choosing the value of α, it is possible to simulate tumor spheroids growing at different rates, even if the agents have the same τ mean duplication times. Low values of α lead to a power-law-like kinetics, when the diameter of the spheroid grows linearly, hence the number of agents $N \propto t^3$. This is because only agents on the surface of the growing spheroid can replicate due to insufficient space inside.

To test the behavior of our model regarding to parameter α, we started simulations from only one agent, and their parameters were inherited without modification on duplication as described in Sec. 3A.

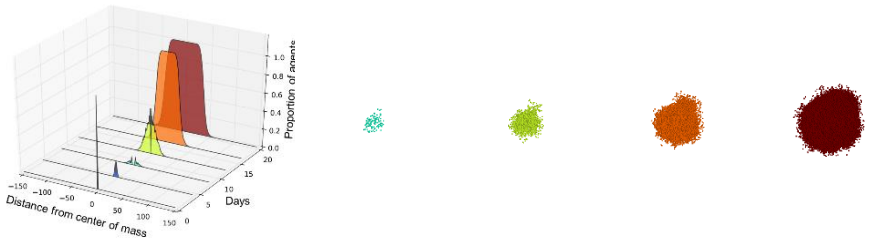
We found that using high values of the parameter ($\alpha \approx 1$) the number of agents grows exponentially, and newly created agents are distributed uniformly inside the spheroid as it can be seen in Fig. 3. Parameters of the agents as well as their approximate number can be found in the caption of the figure.

C. Microenvironment-coupled state transitions of the agents

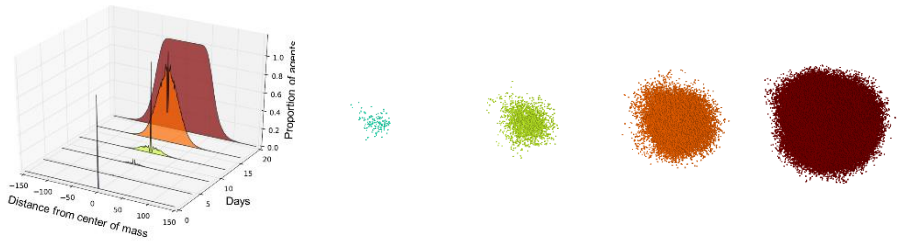
We simulated the oxygen consumption of a tumor spheroid grown in an oxygen-rich environment. As it was presented in [16], solid tumors form distinct layers of cells from their surface to inner regions. The limited availability of oxygen causes cell hypoxia followed by necrosis (a type of cell death) in the core of the spheroid. Biological experiments confirmed the relationship between the thickness of the oxygen-rich outer layer called the *viable rim* and the diameter of the tumor spheroid.



(*a*) Without motility ($\nu = 0$ μmh^{-1}) the growing spheroid forms a dense ball with a well-defined sharp boundary, but the diameter of the spheroid increases slowly because only the agents on the surface are able to duplicate.

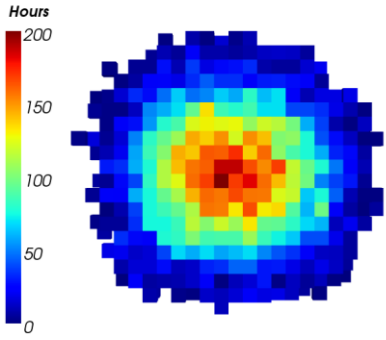


(*b*) With moderate motility ($\nu = 5$ μmh^{-1}) the growth rate of the diameter is clearly increased without the spheroid becoming significantly irregular or scattered. This is mainly because the diameter of the spheroid grows on the same magnitude as the traversable path of the agents via diffusion-like random motion.

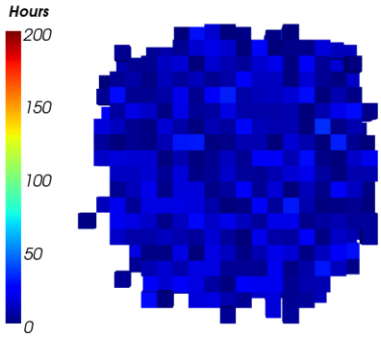


(*c*) High motility ($\nu = 20$ μmh^{-1}) results in a sparse, less regular cluster with scattered border. In this situation, agents are able to travel longer paths between two successive duplication events.

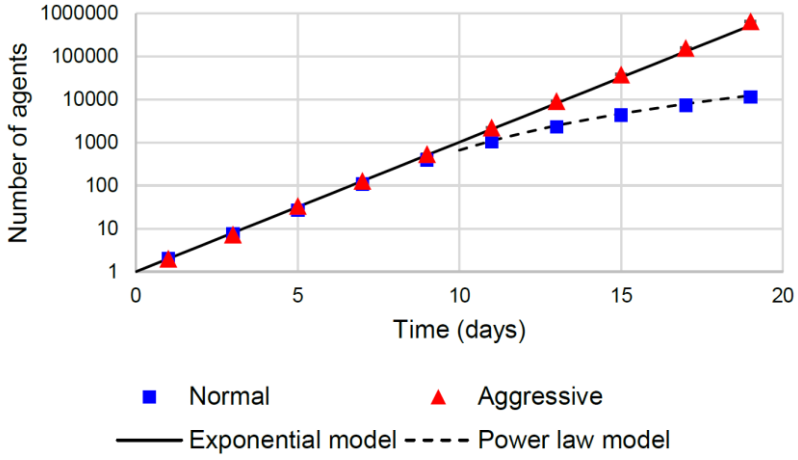
Figure 2: The effect of different values of ν on a growing spheroid started from a single agent. Density diagrams on the left show the proportion of occupied sites at a given distance from the center of mass of the cell cluster. Narrow, continuous, steep-tailed plots belong to dense, regular spheroids. Note, that colors of the spheroids on the right correspond to the density diagram. Parameters used for the simulation: $\tau = 24$ hours, $k = 12$, $\Delta t_p = 15$ min, $\alpha = 0$.



(*a*) Normal duplication strategy ($\alpha = 0$): the duplication process is executed only if a randomly selected neighbor site is empty. Only the agents on the surface of the spheroid are able to perform duplication.



(*b*) Aggressive duplication strategy ($\alpha = 1$): the duplication process is always executed, even if all neighboring sites are occupied. Agents inside the spheroid also duplicate by pushing another agent toward the surface.



(*c*) Size of simulated tumor spheroids using different duplication strategies. Using $\alpha \approx 0$ an initial exponential phase is followed by a linear phase – i.e., the diameter of the tumor spheroid grows linearly. This phase can be well-described with a power law tumor growth model. When $\alpha \approx 1$ the number of agents in the spheroid keeps growing exponentially even after the initial phase.

Figure 3: Cross-section of the simulated tumor spheroids and their growth curves using different duplication strategies. Simulations were started from one agent and stopped at $T_{\max} = 19$ days. For cross-section images we selected a time point, when approximately 4000 agents were present. Agent colors are related to the time passed since the last duplication event of the agent. Parameters used for the simulation: $\tau = 24$ h, $k = 12$, $\Delta t_p = 15$ min, $\nu = 0$ μmh^{-1}, $\alpha = 0$ and $\alpha = 1$.

First, we generated a two-dimensional circular set of *cell* agents with diameter d, as a model of a slice of the whole tumor spheroid. We assumed that the oxygen concentration in the surrounding environment is constant during the experiment, therefore we marked all non-cell nodes as oxygen sources. This means, that oxygen concentration on these outer grid points was constantly $\rho_{O_2,\text{normal}}$. When an agent is present on a certain grid point, it acts as a sink of oxygen, therefore oxygen concentration reduces according to (2), however, oxygen concentration may also increase due to diffusion from outer regions.

When the oxygen concentration on an inner grid point inside the spheroid falls below $\rho_{O_2,\text{critical}}$ the corresponding cell agent (if the site is occupied) changes its type and becomes *necrotic*. Necrotic type agents are unable to move or duplicate, and the transition is irreversible. As a consequence of these two counter-directional processes – oxygen consumption of agents and diffusion from outer regions – a necrotic region starts to grow in the core of the spheroid. As the diameter of the spheroid increases, so does the necrotic region, but the thickness of the viable rim decreases, just as it was observed in real experiments.

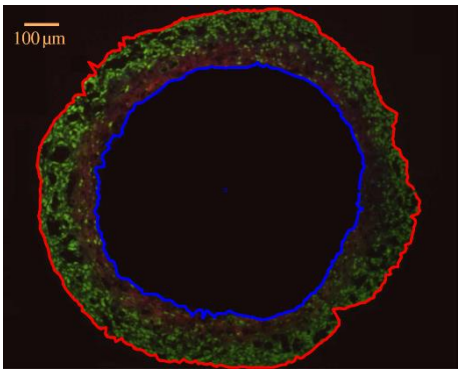
Multiple simulations were executed with the same parameter set to average stochastic effects. Results of each simulation were evaluated by

(i) determining the convex hull of both the whole spheroid (H_0) and the necrotic region (H_n) at a given time,

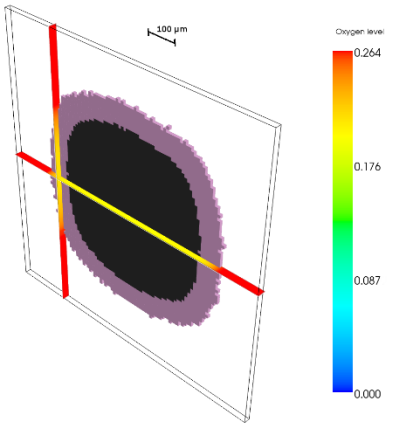
(ii) calculating the mean distance r_0 and r_n between the mass of center of the spheroid and the points in H_0 and H_n, respectively,

(iii) calculating the viable rim thickness as $r_c = r_0 - r_n$.

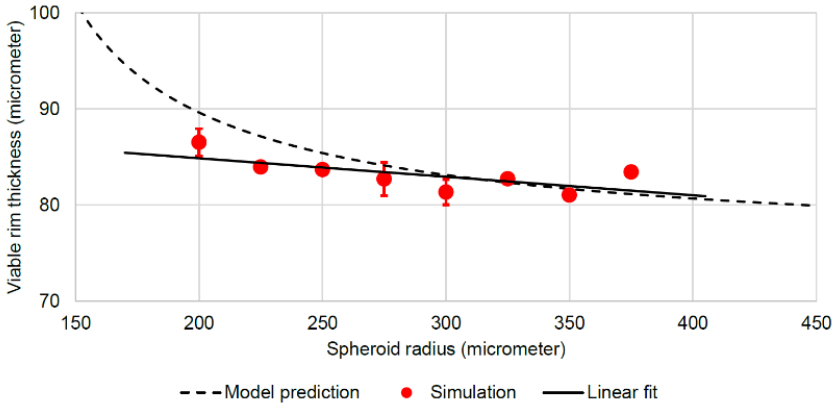
We found a qualitative accordance with the model presented in [16], meaning that the viable rim thickness decreases as the diameter of the spheroid increases (see quantitative results in Fig. 4c). The exact parameter values of the agents as well as the relevant coefficients of the mass-transport model can be found in the caption of the figure.



(*a*) Image of a tumor spheroid cross-section. Proliferating (living) cells are shown as green particles near the spheroid boundary marked by a red line. The hypoxic core boundary is shown by the blue line. (Photo by Grimes et al. [16] licensed under the terms of Creative Commons Attribution License.)



(*b*) Representative image of a simulation. Pink dots represent living agents, black dots show necrotic (dead) agents. Note the color-scaled section planes showing oxygen concentration of the microenvironment.



(*c*) The thickness of the viable rim of a tumor spheroid depends on its radius. The dashed line shows the predicted value of the viable rim using the model of Grimes et al. with $r_l = 130$ µm. In agreement with the prediction the best linear fit of simulation data shows a clear decreasing tendency as the spheroid diameter grows. Simulation data shows mean±SD.

Figure 4: The effect of oxygen consumption of cells in a tumor spheroid. Low oxygen level leads to necrosis in the core of a growing spheroid. Parameters used for the simulation: $\tau = 24$ h, $k = 12$, $\Delta t_p = 15$ min, $\nu = 0$ µmh^{-1}, $\alpha = 0$. In (2) we set $\rho_{O2,normal} = 0.264$, $\rho_{O2,critical} = 0.2$, $D_{O_2} = 6 \times 10^{-8}$m^{2}h^{-1}, $U_{O_2}^{cancercell} = 0.1$ min^{-1}, $\lambda_{O_2} = 0.01\, U_{O_2}^{cancercell}$ and $\Delta t = 500$ ms according to literature data [17].

4. Conclusion

This paper presented an on-lattice computational model of solid tumors in which individual units (so-called agents) are coupled to their microenvironment. Features of agents like motility, proliferation potential or sensitivity to different compounds from the microenvironment can be set independently. The simplicity of the model makes it possible to simulate a few mm^3 of tissue on a standard desktop workstation or laptop computer, in reasonable time.

The effects of different values of agents' motility and aggressivity parameters were tested along with their microenvironment-coupled behavior. In agreement with experimental observations we found that both the motility and the aggressive proliferation influences the size and the shape of a growing solid tumor. It is also have to be mentioned, that high motility or aggressive duplication strategy of the agents makes the simulation slightly slower, especially for large and dense clusters. This effect could be compensated by a selection heuristic we recently presented for an off-lattice agent based model [18].

Simulation results of the development of a well-oxygenated viable rim in tumor spheroids show good qualitative match to real biological observations, however, optimal parameters of the model must be determined to achieve quantitative match as well.

We think that this computational model can be a suitable tool for scientists to give valuable insights into emergent behavior of complex biological systems.

Supplementary information

Additional data including source codes, simulation data and animations are available at https://bitbucket.org/kissdanieldezso/onlatticets.

Acknowledgements

Dániel Kiss was supported by the ÚNKP-17-3/I New National Excellence Program of the Ministry of Human Capacities. Anna Lovrics was supported by OTKA PD124467 grant. The authors wish to thank Gergely Szakács and the members of Membrane Proteins Research Group for their contribution to this project.

References

[1] Starruß, J., de Back, W., Brusch, L., Deutsch, A. "Morpheus: a user-friendly modeling environment for multiscale and multicellular systems biology", *Bioinformatics* 30, 9 (2014) 1331–1332.

[2] Ghaffarizadeh, A., Heiland, R., Friedman, S. H., Mumenthaler, S. M., Macklin, P., "PhysiCell: an open source physics-based cell simulator for 3-D multicellular systems", *PLoS Comput. Biol.* 14, 2 (2018).

[3] Kang, S., Kahan, S., McDermott, J., Flann, N., Shmulevich, I., "Biocellion: accelerating computer simulation of multicellular biological system models", *Bioinformatics* 30, 21 (2014) 3101–3108.

[4] Drasdo, D., Höhme, S., "Individual-Based Approaches to Birth and Death in Avascular Tumors", *Mathematical and Computer Modelling* 37, (2003) 1163–1175.

[5] Drasdo, D., Hoehme, S., Block, M., "On the Role of Physics in the Growth and Pattern Formation of Multi-Cellular Systems: What can we Learn from Individual-Cell Based Models?", *Journal of Statistical Physics* 128, 1 (2007) 287–345.

[6] Walker, D. C., Southgate, J., Hill, G., Holcombe, M., Hosea, D. R., Wood, S. M., Mac Neil, S., Smallwood, R. H., "The epitheliome: agent-based modelling of the social behaviour of cells.", *BioSystems* 76, (2004) 89–100.

[7] Cytowski, M., Szymanska, Z., "Large-Scale Parallel Simulations of 3D Cell Colony Dynamics", *Computing in Science & Engineering* 16, 5 (2014) 86–95.

[8] Cytowski, M., Szymanska, Z., Uminski, P., Andrejczuk, G., Raszkowski K., "Implementation of an Agent-Based Parallel Tissue Modelling Framework for the Intel MIC Architecture", *Scientific Programming* 8721612, (2017).

[9] Waclaw, B., Bozic, I., Pittman, M. E., Hruban, R. H., Vogelstein, B., Nowak, M. A., "A spatial model predicts that dispersal and cell turnover limit intratumour heterogeneity", *Nature* 525, 7568 (2015) 261–264.

[10] Summers, H. D., Wills, J. W., Brown, M. R., Rees, P., "Poisson-Event-Based Analysis of Cell Proliferation", *Cytometry A.* 87, 5 (2015) 385–392.

[11] Yates, C. A., Ford, M. J., Mort, R. L., "A Multi-Stage Representation of Cell Proliferation as a Markov Process", *Bulletin of Mathematical Biology* 79, 12 (2017) 2905–2928.

[12] Ghaffarizadeh, A., Friedman, S. H., Macklin, P., "BioFVM: an efficient, parallelized diffusive transport solver for 3-D biological simulations", *Bioinformatics* 32, 8 (2016) 1256–1258.

[13] Ramachandran, P., Varoquaux, G., "Mayavi: 3D Visualization of Scientific Data", *IEEE Computing in Science & Engineering*, 13, 2 (2011), 40–51.

[14] Roberts, D. L., Selim, M. S., "Comparative study of six explicit and two implicit finite difference schemes for solving one-dimensional parabolic partial differential equations", *International Journal for Numerical Methods in Engineering* 20 (1984) 817–844.

[15] Mombach, J., Glazier, J., "Single cell motion in aggregates of embryonic cells", *Phys. Rev. Lett.* 76, 3032 (1996).

[16] Grimes, D. R., Kelly, C., Bloch, K., Partridge, M., "A method for estimating the oxygen consumption rate in multicellular tumour spheroids", *J. R. Soc. Interface* 11, 20131124 (2014).

[17] Macklin, P., Edgerton, M. E., Thompson, A. M., Cristini, V., "Patient-calibrated agent-based modelling of ductal carcinoma in situ (DCIS): From microscopic measurements to macroscopic predictions of clinical progression", *Journal of Theoretical Biology* 301 (2012) 122–140.

[18] Kiss, D., Lovrics, A., "Performance analysis of a computational off-lattice tumor growth model", *Proceedings of the IEEE 30th Jubilee Neumann Colloquium* (2017).

Acta Universitatis Sapientiae

The scientific journal of Sapientia Hungarian University of Transylvania (Cluj-Napoca, Romania) publishes original papers and surveys in several areas of sciences written in English.
Information about each series can be found at
http://www.acta.sapientia.ro.

Acta Universitatis Sapientiae
Electrical and Mechanical Engineering

Sapientia University

Scientia Publishing House

ISSN 2065-5916
http://www.acta.sapientia.ro

Information for authors

Acta Universitatis Sapientiae, Electrical and Mechanical Engineering publishes only original papers and surveys in various fields of Electrical and Mechanical Engineering. All papers are peer-reviewed.

Papers published in current and previous volumes can be found in Portable Document Format (PDF) form at the address: `http://www.acta.sapientia.ro`.

The submitted papers must not be considered to be published by other journals. The corresponding author is responsible to obtain the permission for publication of co-authors and of the authorities of institutes, if needed. The Editorial Board is disclaiming any responsibility.

The paper must be submitted both in MSWord document and PDF format. The submitted PDF document is used as reference. The camera-ready journal is prepared in PDF format by the editors. In order to reduce subsequent changes of aspect to minimum, an accurate formatting is required. The paper should be prepared on A4 paper (210×297 mm) and it must contain an abstract of 200-250 words.

The language of the journal is English. The paper must be prepared in single-column format, not exceeding 12 pages including figures, tables and references.

The template file from http://www.acta.sapientia.ro/acta-emeng/emeng-main.htm may be used for details.

Submission must be made only by e-mail (`acta-emeng@acta.sapientia.ro`).

One issue is offered to each author free of charge. No reprints are available.

Contact address and subscription:
Acta Universitatis Sapientiae, Electrical and Mechanical Engineering
Sapientiae Hungarian University of Transylvania
RO 400112 Cluj-Napoca
Str. Matei Corvin nr. 4.
E-mail: acta-emeng@acta.sapientia.ro

Printed by Digital Color Company
www.digitalcolorcompany.ro