Application of SVD in the Regularization of the Control Law for GPC-based Tracking Systems

Matheus Pelzl [*] **Thyago Estrabis** [**] **Gabriel Gentil** [***]
Raymundo Cordero [****] **Walter Suemitsu** [†]

[*] *Federal University of Mato Grosso do Sul, MS,*
(e-mail:matheus.pelzl@ieee.org)
[**] *Federal University of Rio de Janeiro, RJ*
(e-mail:thyago.estrabis@gmail.com)
[***] *Federal University of Rio de Janeiro, RJ*
(e-mail:gabrielgent@gmail.com)
[****] *Federal University of Mato Grosso do Sul, MS,*
(e-mail:raymundo.garcia@ufms.br)
[†] *Federal University of Rio de Janeiro, RJ,*
(e-mail:walter@dee.ufrj.br).

Abstract:
Generalized predictive control (GPC) has become one of the most studied and popular control approaches. The GPC control law requires the estimation of the Hessian matrix, which requires a matrix inversion procedure. However, depending on the plant model and the GPC parameters, the aforementioned procedure may be ill-conditioned: a slight variation in the parameters may generate a more significant variation in the Hessian matrix value. In that case, the noise or quantization effect reduces the GPC robustness. The process of solving ill-conditioned problems is called regularization. This paper proposes the Singular Value Decomposition (SVD) application to regularize the matrix inversion procedure used to get the Hessian matrix. SVD decomposes a matrix based on the concept of singular values. Only the most significant singular values are used in the SVD regularization technique to calculate a matrix inverse, as the smallest singular values produce ill-conditioned problems. A methodology to define the singular values used in matrix inversion is explained in this work. The proposed approach was used in a GPC-based resonant controller, using 16 bits fixed-point numbers. Simulation and experimental tests using a FPGA show that the proposed approach allows getting an accurate and robust GPC response for the tracking of sinusoidal references.

Keywords: Generalized predictive control; matrix inversion; regularization; singular value decomposition.

1. INTRODUCTION

Nowadays, generalized predictive control (GPC) is one of the most popular and studied control strategies (Camacho and Bordons, 2007; Qin and Badgwell, 2003). GPC is based on the prediction of future plant responses. That prediction is made using the current augmented model state vector and a set of future control actions. An optimization process allows the optimal future actions to minimize a cost function that measures the control performance (Wang, 2009; Cordero et al., 2021a).

The GPC control law requires calculating a Hessian matrix through the matrix inversion method. This matrix depends on the augmented model and the prediction parameters. Depending on those factors, the matrix whose inverse is the Hessian can be almost singular (with a high condition number). If so, the GPC control law estimation may become an *ill-conditioned* problem (Aidoud et al., 2019; Sanchis et al., 2002): a small error in the input data may produce a larger error in the answer. The nominal plant parameters used to get the control law are different from the actual parameters for practical applications. Besides, the digital implementation of the controller produces quantization errors. As a result, the estimation of the GPC control law may become unstable (prone to significant numerical errors) when the matrix inversion to get the Hessian matrix is *ill-conditioned.*

Regularization is the process to solve *ill-conditioned* problems. Different regularization techniques were proposed in literature (Neumaier, 1998; Belsley and Oldford, 1986). Singular value decomposition (SVD) can be used for regularization: the matrix to be inverted is decomposed into three matrix factors, using the concept of singular values, and the matrix inverse is estimated through those factors (Strang, 1988; Wang, 2009). SVD-based matrix inversion method eliminates the effect of the small singular values (which are responsible for the *ill-conditioned* inversion

[⋆] This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

ISSN: 2525-8311

DOI: 10.20906/CBA2022/3553

problem). Thus the matrix inversion is *well-conditioned* (small variations in the input data will produce small variations in the result). This matrix inversion approach is also called Truncated SVD (TSVD).

This paper proposes using SVD to get a robust GPC control law against noise and numerical errors, which are common problems in embedded systems. The SVD-based inversion method is used to get the Hessian matrix. The mathematical analysis considers a SISO (Single-Input Single-Output) plant, but it can be easily adapted for MIMO (Multiple-Input Multiple-Output) plants. Simulation and experimental results using an Field-Programmable Gate Array (FPGA) show that the proposed approach allows calculating a *well-conditioned* GPC control law.

In this paper, $M \in \Re^{n\times m}$ indicates that M is a $n \times m$ matrix, I_n denotes a $n \times n$ identity matrix, O represents a matrix of zeros with adequate dimensions, the difference operator is $\Delta a(k) = \Delta^1 a(k) = a(k) - a(k-1)$, $\Delta^0 a(k) = a(k)$ and $\Delta^m a(k) = \Delta^{m-1}a(k) - \Delta^{m-1}a(k-1)$. Besides, similarly to MATLAB software, $M(r_1 : r_2, c_1 : c_2)$ denotes a sub-matrix of M composed by the element between the rows from r_1 to r_2 and the columns c_1 to c_2.

2. THEORETICAL FOUNDATIONS

2.1 GPC

The discrete-time space state model of a n-order SISO (Single Input Single Output) plant is defined as follows:

$$x_d(k+1) = A_d x(k) + B_d u_d(k), \tag{1}$$

$$y_d(k) = C_d x(k), \tag{2}$$

where $x_d(k) \in \Re^{n\times 1}$ is the state vector, $u_d(k)$ is the plant input, $y_d(K)$ is the plant output, $A_d \in \Re^{n\times n}$, $B_d \in \Re^{n\times 1}$ and $C_d \in \Re^{1\times n}$. In GPC, an augmented model is used to predict the future plant response:

$$\underline{x}(k+1) = \underline{\mathcal{A}}\underline{x}(k) + \underline{\mathcal{B}}\mu(k), \tag{3}$$

$$\underline{y}(k) = \underline{\mathcal{C}}\underline{x}(k), \tag{4}$$

where $\underline{\mathcal{A}}$, $\underline{\mathcal{B}}$, $\underline{\mathcal{C}}$, $\underline{x}(k)$, $\underline{y}(k)$ and $\mu(k)$ are defined according to the prediction technique. The matrices $\underline{\mathcal{A}}$, $\underline{\mathcal{B}}$ and $\underline{\mathcal{C}}$ depend on A_d, B_d and C_d. Table 1 shows the matrices of the augmented model for the following approaches:

- GPC for step reference tracking defined in Wang (2009).
- Poly-GPC: GPC for the tracking of polynomial references of degree $m-1$, defined in Cordero et al. (2021b).
- RGPC: Resonant GPC for sinusoidal reference tracking defined in Cordero et al. (2022).

In GPC, the future outputs of the selected augmented model $\underline{y}(i+1)$, $\underline{y}(i+2)$, ..., $\underline{y}(i+n_p)$ are predicted, being n_p the size of the prediction window (Wang, 2009). This prediction is done using the current model state vector $\underline{x}(k)$ and the future control trajectory $\mu(k)$, $\mu(i+1)$, ..., $\mu(i+n_c-1)$, being n_c the size of the control horizon such as $n_c \leq n_p$. The vector of future responses Y and the vector of future actions U are defined as follows:

$$Y = \left[\underline{y}(k+1)\ \underline{y}(k+2)\ \ldots\ \underline{y}(k+n_p)\right] \in \Re^{n_p\times 1}, \tag{5}$$

$$U = \left[\mu(k)\ \mu(k+1)\ \ldots\ \mu(k+n_c-1)\right] \in \Re^{n_c\times 1}, \tag{6}$$

Table 1. Augmented Prediction Models

Matrix	Wang (2009)	Poly-GPC	RGPC
$\underline{\mathcal{A}}$	$\begin{bmatrix} A_d & O \\ C_dA_d & 1 \end{bmatrix}$	$\begin{bmatrix} E & -\gamma C_dA_d \\ O & A_d \end{bmatrix}$	$\begin{bmatrix} A_d & O & O \\ C_dA_d & 1 & \varsigma \\ C_dA_d & 1 & 1+\varsigma \end{bmatrix}$
$\underline{\mathcal{B}}$	$\begin{bmatrix} B_d \\ C_dB_d \end{bmatrix}$	$\begin{bmatrix} -\gamma C_dB_d \\ B_d \end{bmatrix}$	$\begin{bmatrix} B_d \\ C_dB_d \\ C_dB_d \end{bmatrix}$
$\underline{\mathcal{C}}$	$\begin{bmatrix} O & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & O & O \end{bmatrix}$	$\begin{bmatrix} O & 0 & 1 \end{bmatrix}$
$\underline{x}(k)$	$\begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix}$	$\begin{bmatrix} \varepsilon(k) \\ \Delta^m x(k) \end{bmatrix}$	$\begin{bmatrix} \varsigma x(k-1) - \Delta^2 x(k) \\ \Delta e(k) \\ e(k) \end{bmatrix}$
$\mu(k)$	$\Delta x(k)$	$\Delta^m x(k)$	$\varsigma u(k-1) - \Delta^2 u(k)$
$\underline{y}(k)$	$y(k)$	$e(k)$	$e(k)$

$\varepsilon(k) = [e(k)\ \Delta e(k)\ ...\ \Delta^{m-1}e(k)]^T$, $\gamma = [1\ 1\ ...\ 1]^T$, $\varsigma = 2\cos(\varpi) - 2$, E is an upper diagonal matrix composed by 1, while $\varpi = \omega t_s$ (ω is the sinusoidal reference frequency).

The vector of future response can be obtained through (7):

$$Y = F\underline{x}(k) + \Phi, \tag{7}$$

where

$$F = \begin{bmatrix} \underline{\mathcal{C}}\,\underline{\mathcal{A}} \\ \underline{\mathcal{C}}\,\underline{\mathcal{A}}^2 \\ \vdots \\ \underline{\mathcal{C}}\,\underline{\mathcal{A}}^{n_p} \end{bmatrix}, \tag{8}$$

$$\Phi = \begin{bmatrix} \underline{\mathcal{C}\mathcal{B}} & 0 & 0 & \ldots & 0 \\ \underline{\mathcal{C}\mathcal{A}\mathcal{B}} & \underline{\mathcal{C}\mathcal{B}} & 0 & \ldots & 0 \\ \underline{\mathcal{C}\mathcal{A}}^2\underline{\mathcal{B}} & \underline{\mathcal{C}\mathcal{A}\mathcal{B}} & \underline{\mathcal{C}\mathcal{B}} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \underline{\mathcal{C}\mathcal{A}}^{n_p-1}\underline{\mathcal{B}} & \underline{\mathcal{C}\mathcal{A}}^{n_p-2}\underline{\mathcal{B}} & \ldots & \ldots & \underline{\mathcal{C}\mathcal{A}}^{n_p-n_c}\underline{\mathcal{B}} \end{bmatrix}. \tag{9}$$

An optimization process is applied to get the optimal U to minimize future errors. Let $R \in \Re^{n_p\times 1}$ be a vector composed of the plant references during the prediction window. According to GPC, R should be constant within that window. Equation (10) defines the cost function used in the optimization process (Wang, 2009):

$$J = (Y - R)^T (Y - R) + U^T R_w U, \tag{10}$$

$$R_w = r_w I_c, \tag{11}$$

where I_c is a $n_c \times n_c$ identity matrix, while r_w is an adjustment parameter. The larger the value of r_w, the more important the minimization of the magnitude of U (Wang (2009)). Replacing (5) into (10) yields:

$$\begin{aligned} J = &\left(R - F\underline{x}(k)\right)^T \left(R - F\underline{x}(k)\right) - 2U^T\Phi^T \left(R - F\underline{x}(k)\right) \\ &+ U^T \left(\Phi^T\Phi + R_w\right) U. \end{aligned} \tag{12}$$

The optimal solution of (10), U_{op}, makes $\frac{\partial J(U_{op})}{\partial U} = 0$. Replacing (10) into the derivative of (12) allows getting U_{op}:

$$\frac{\partial J(U_{op})}{\partial U} = 0 \rightarrow U_{op} = \left(\Phi^T\Phi + R_w\right)^{-1} \Phi^T \left(R - F\underline{x}(k)\right). \tag{13}$$

The receding horizon approach states that only the first element of the optimized solution, i.e., $\mu(k)$ is used to define the control law (i.e., the plant input). Hence:

$$\mu(k) = \underbrace{[1\ 0\ \ldots\ 0]}_{n_c \text{ elements}} \underbrace{\left(\Phi^T\Phi + R_w\right)^{-1} \Phi^T \left(R - F\underline{x}(k)\right)}_{U_{op}}. \tag{14}$$

The GPC approach in Wang (2009) was designed to track a step reference. Thus, $R = [1\ 1\ \ldots\ 1]^T$. However, polynomial and sinusoidal references change over time. For that reason, the approaches in Cordero et al. (2021b), and Cordero et al. (2022) use the tracking error $e(k)$ as the prediction model output due to the desired value (i.e., the reference) of an error is always zero. Hence $R = [0\ 0\ \ldots\ 0]^T$ in Cordero et al. (2021b) and Cordero et al. (2022). In those cases:

$$\mu(k) = -[1\ 0\ \ldots\ 0]\left(\Phi^T\Phi + R_w\right)^{-1}\Phi^T F\underline{x}(k), \qquad (15)$$
$$R = [0\ 0\ \ldots\ 0]^T.$$

According to Table 1, $\mu(k)$ depends on past and present values of the input plant $u(k)$. Hence, $u(k)$ can be deduced from the definition of $\mu(k)$ and using (14).

The Hessian matrix H is defined as follows (Wang, 2009):

$$H = \left(\Phi^T\Phi + R_w\right)^{-1}. \qquad (16)$$

The Hessian exists if $\Phi^T\Phi + R_w$ is non-singular.

2.2 Singular Value Decomposition and Ill-Conditioned Problems

Let $A \in \Re^{n\times n}$. The matrix $A^TA \in \Re^{n\times n}$ is positive definite. Hence, each eigenvalue of A^TA,denoted by $\lambda_j,\ j = 1,\ \ldots,\ n$ is real and non-negative (Chen, 1999):

$$0 \le \lambda_j, \quad j = 1,\ \ldots,\ n. \qquad (17)$$

The singular values of A, $\sigma_j,\ j = 1,\ \ldots,\ n$, are the square root of the eigenvalues of A^TA (Chen, 1999):

$$\sigma_j = \lambda_j^{0.5}, \quad j = 1,\ \ldots,\ n. \qquad (18)$$

Any matrix $A \in \Re^{n\times n}$ can be decomposed as follows:

$$A = USV^T. \qquad (19)$$

where the columns of U are the orthonormalized eigenvectors of AA^T, the columns of V are the orthonormalized eigenvectors of A^TA, while S is a diagonal matrix whose diagonal is composed of the singular values of A:

$$S = \begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_n \end{bmatrix}, \quad \sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_n \ge 0. \qquad (20)$$

where the singular values in the diagonal of S are arranged in decreasing order. The inverse of A can be expressed in terms of U, V and S (Strang, 1988):

$$A^{-1} = VS^{\dagger}U^T, \qquad (21)$$

where

$$S^{\dagger} = S^{-1} = \begin{bmatrix} 1/\sigma_1 & 0 & \ldots & 0 \\ 0 & 1/\sigma_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1/\sigma_n \end{bmatrix}. \qquad (22)$$

Let's consider the following linear equation

$$y_{eq} = Ax_{eq}, \qquad (23)$$

where x_{eq} is the solution to be found. Equations (21) shows that A^{-1} exists if all the singular values of A are nonzero. In that case:

$$x_{eq} = A^{-1}y_{eq}. \qquad (24)$$

However, if a singular value σ is near zero, then $1/\sigma$ tends to be infinite. In that case, A is *ill-conditioned*: the matrix A is almost singular and small variations in the parameters of y_{eq} will produce a large variation in the solution x_{eq} (Strang, 1988). That variations can be produced by noise in the signal y_{eq}, rounding process, or working with a limited number of bits to perform arithmetical operations.

Let σ_{max} and σ_{min} be the largest and the smallest singular value of A, respectively. The condition number of A, denoted by c_A is (Strang, 1988):

$$c_A = \frac{\sigma_{max}}{\sigma_{min}} = \frac{\sigma_1}{\sigma_n}. \qquad (25)$$

The matrix A is *ill-conditioned* if its conditioning number (c_A) is too large. Equation (25) allows deducing that small singular values produce *ill-conditioned* inversion problems. A common technique to eliminate the *ill-conditioned* problem is to redefine the matrix $S^{\dagger}$ in (21):

$$S^{\dagger} = diag(z_1,\ z_2, \ldots,\ z_n), \qquad (26)$$
$$z_i = \begin{cases} 1/\sigma_i & \text{if } \sigma_i\ >\ \sigma_{thr}; \\ 0 & \text{otherwise} \end{cases},\ i = 1, \ldots, n.$$

where σ_{thr} is a threshold. Equation (26) allows eliminating the effect of the small singular values of A. A procedure used to solve *ill-conditioned* problems is called regularization. Thus, the following regularized inverse of A in (27) is *well-conditioned* (with small condition number):

$$A^{-1} \approx V diag(z_1,\ z_2, \ldots,\ z_n)U^T. \qquad (27)$$

The regularization process based on using only the largest singular values of a matrix is called Truncated SVD (Strang, 1988). Let n_s be the number of singular values greater than σ_{thr}:

$$\sigma_i > \sigma_{thr}, \quad i = 1,\ 2,\ \ldots,\ n_s. \qquad (28)$$

Considering only the most significant singular values defined in (28), the decomposition of any matrix can be done as follows:

$$A \approx A_T = U_TS_TV_T^T, \qquad (29)$$
$$U_T = U(1:n_c, 1:n_s) \in \Re^{n_c\times n_s},$$
$$S_T = S(1:n_s, 1:n_s) \in \Re^{n_s\times n_s},$$
$$V_T = V(1:n_c, 1:n_s) \in \Re^{n_c\times n_s},$$

where U_T is composed by the firsts n_s columns of U, while V_T is composed by the firsts n_s columns of V. In that case, the condition number of A_T is

$$c_{A_T} = \frac{\sigma_1}{\sigma_{n_s}}. \qquad (30)$$

As $\sigma_{n_s} > \sigma_n$ then $c_{A_T} < c_A$. Applying (21) into (29), yields:

$$A_T^{-1} = V_TS_T^{-1}U_T^T. \qquad (31)$$

The inverses of A using (27) and (31) are equal.

3. APPLICATION OF SVD IN THE REGULARIZATION OF THE GPC CONTROL LAW

Equation (13) can be rewritten according to (24):

$$\underbrace{U_{op}}_{x_{eq}} = \underbrace{\left(\Phi^T\Phi + R_w\right)^{-1}}_{A}\underbrace{\Phi^T\left(R - F\underline{x}(k)\right)}_{y_{eq}}. \qquad (32)$$

Note that $A = \left(\Phi^T\Phi + R_w\right)^{-1}$ in (32) is the Hessian matrix in (16), whose value depends on the matrices that compose the augmented model ($\underline{\mathcal{A}}$, $\underline{\mathcal{B}}$ and $\underline{\mathcal{C}}$) and the GPC parameters (n_p, n_c and r_w). Depending of those values, $\left(\Phi^T\Phi + R_w\right)$ may be *ill-conditioned.* Besides, note that $\Phi^T\left(R - F\underline{x}(k)\right)$ depends on the plant state variables (as indicated in Table 1). Hence, noise during the acquisition of the plant state variables or quantization effect may produce an undesirable variation in the estimation of U_{op}, when the GPC algorithm is implemented in digital processors.

This work proposes using SVD to estimate the Hessian matrix in (16). Equation (9) allows proving that $\Phi^T\Phi + R_w$ is a $n_c \times n_c$ matrix, so it has n_c singular values $\sigma_{1H},\ \ldots,\sigma_{n_cH}$. Thus, according to (19), the matrix $\Phi^T\Phi + R_w$ can be decomposed as follows:

$$\Phi^T\Phi + R_w = U_H S_H V_H^T, \tag{33}$$

$$S_H = \begin{bmatrix} \sigma_{1H} & 0 & \ldots & 0 \\ 0 & \sigma_{2H} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_{n_cH} \end{bmatrix}. \tag{34}$$

The Hessian matrix can be estimated applying (26) and (27) into (33):

$$H = V_H diag(z_{1H},\ z_{2H}, \ldots,\ z_{n_cH}) U_H^T, \tag{35}$$

$$z_{iH} = \begin{cases} 1/\sigma_{iH} & \text{if } \sigma_{iH} > \sigma_{thr}; \\ 0 & \text{otherwise} \end{cases},\ i = 1, \ldots, n_c.$$

Equation (30) allows the relationship between the condition number and the number of the largest singular values used to get the inverse of $\Phi^T\Phi + R_w$, as shown in Figure 1. This relationship can be used to set the adequate number of singular values for regularized matrix inversion. However, selecting the truncation range (i.e., the number of the most significant singular values) is essential in SVD-related applications. Much information about the matrix inverse (the Hessian matrix in this case) may be lost or distorted if the few most significant singular values are selected. However, selecting a large number of singular values may produce an inverse matrix that is still *ill-conditioned* (i.e., a Hessian matrix would still be unstable and sensitive to noise and parameter variations) (Gavish and Donoho, 2014). For that reason, the following subsection explains a methodology to select the adequate number of singular values for the regularization procedure.

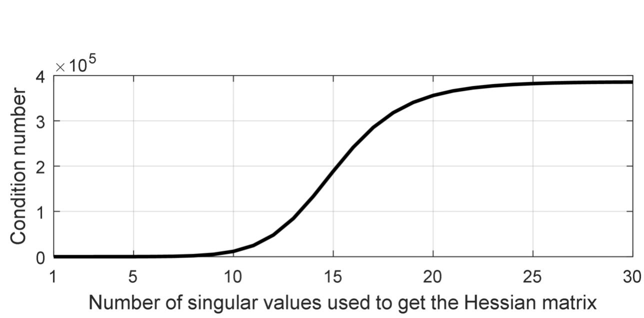


Figure 1. Condition number in function of the number of the largest singular values used in regularization.

3.1 Selection of Optimal Number of Singular Values

Many methods are proposed for selecting the optimal truncation point (i.e., selecting the adequate singular values used in regularization). Some techniques include identifying "knees" in the distribution curve of the condition numbers (see Figure 1). Such techniques, although showing some effectiveness, are very heuristic.

Gavish and Donoho (2014) propose a more analytical technique to select the singular values. This technique estimates a value τ, the optimal location for the hard thresholding of singular values. For matrices with known noise amplitude, the estimation of τ is presented in (36):

$$\tau_* = \lambda(\beta)\sqrt{n} * \sigma, \tag{36}$$

where σ is the noise amplitude, β is the relation of the matrix dimensions, i.e. $\beta = m/n$, and $\lambda(\beta)$ is the optimal hard threshold coefficient. However, the noise level of the $(\Phi^T\Phi + R_w)$ matrix, i.e., the inverse of the Hessian matrix, is unknown. Hence, for data with an unknown noise level, (37) can be applied:

$$\hat{\tau}_* = \omega(\beta) * y_{med}, \tag{37}$$

Where $\omega(\beta)$ is the optimal hard threshold coefficient for σ unknown, and y_{med} is the median value of singular values of the matrix in question. Nevertheless, as the Hessian matrix is ever squared, as a result of the term $\phi^T\phi$. Thus, $\beta = 1$, and the value referent to $\omega(1)$ is tabled and set as 2.852. Thus, according to Gavish and Donoho (2014), the optimal truncation of the inverse of the Hessian matrix used in this paper is defined as in (38):

$$\hat{\tau}_* = 2.8582 * y_{med}. \tag{38}$$

4. RESULTS

Simulation tests were performed to prove the advantages of the proposed application of SVD to get a robust GPC control law. The plant used in the test has the transfer function $G(s) = \frac{1.418\times10^6 s + 3.637\times10^8}{s^3 + 2179s^2 + 2.273\times10^6 s + 7.274\times10^8}$. This plant was discretized using a sample time of $t_s = 0.5$ ms. Thus the matrix of the discrete-time plant model are $A = \begin{bmatrix} 0.413 & 0.454 & 0 \\ -0.240 & 0.788 & 0 \\ -0.437 & 0.422 & 0.774 \end{bmatrix}$, $B = \begin{bmatrix} 0.1331 \\ 0.4528 \\ 0.1273 \end{bmatrix}$ and $C = [0\ 0\ 1]$. The proposed approach was applied in the GPC-based resonant controller described in (Cordero et al., 2022). Table 2 lists the GPC parameters, and the number of singular values used in regularization (according to (38)) applied in the tests. Figure 2 shows the sinusoidal reference (50 Hz) of the plant.

The tests were done using 16-bits fixed-point number representation, using eight fractional bits. Besides, white noise (zero mean, power spectral density of 0.00001) was added to the plant state variables. Noise and the reduced number of bits for arithmetic operations are problems in embedded systems, mainly when the controller is based on an *ill-conditioned* system. These test conditions were

Table 2. GPC and SVD Parameters

Configuration	n_p	n_c	r_w	Number of singular values
1	20	10	0.01	4
2	100	50	0.001	22
3	100	80	0.1	25

considered to show how the proposed approach allows for more robust GPC controllers.

Figures 3 to 8 show the simulation results for the tracking error, with/without using SVD and with/without adding noise to the plant state variables. The settling time increases (in the tests without noise), while the noise effect is greater when SVD is not used. These results result from using an *ill-conditioned* Hessian matrix to set the GPC control law. On the other hand, the proposed SVD-based approach allows getting a more robust control response, while the tracking errors tend to zero when SVD is applied.

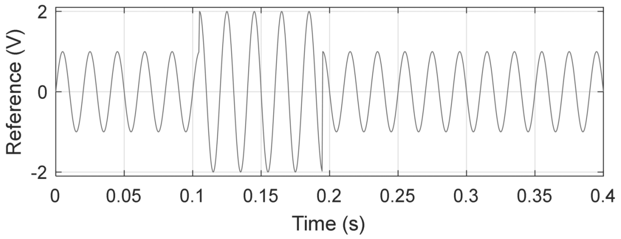


Figure 2. Control system reference.

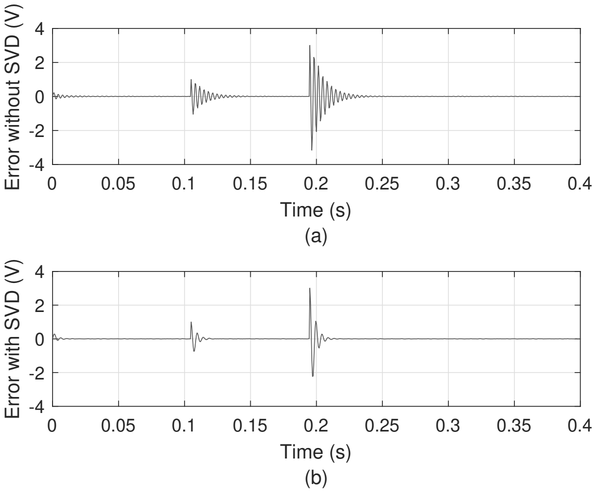


Figure 3. Simulation test: Error for $n_p = 20$, $n_c = 10$ and $r_w = 0.01$, without adding noise. (a) Without using SVD. (b) Using SVD.

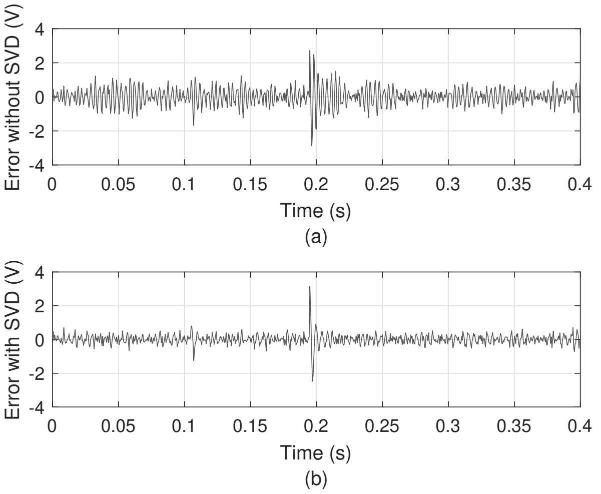


Figure 4. Simulation test: Error for $n_p = 20$, $n_c = 10$ and $r_w = 0.01$, adding noise. (a) Without using SVD. (b) Using SVD.

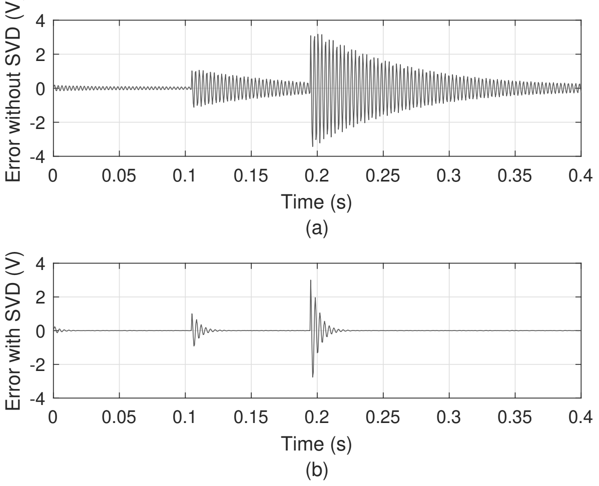


Figure 5. Simulation test: Error for $n_p = 100$, $n_c = 50$ and $r_w = 0.001$, without adding noise. (a) Without using SVD. (b) Using SVD.

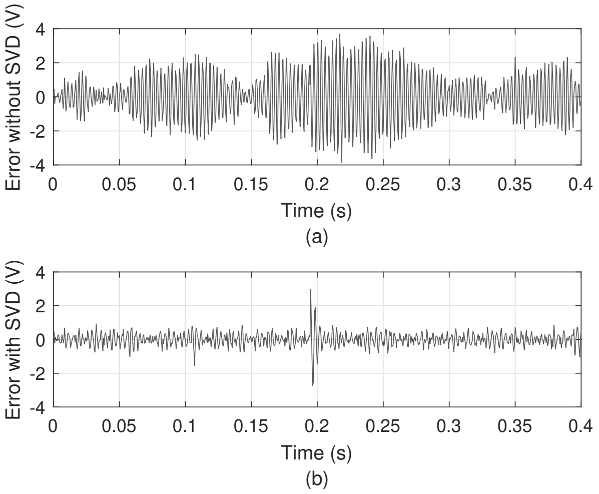


Figure 6. Simulation test: Error for $n_p = 100$, $n_c = 50$ and $r_w = 0.001$, adding noise. (a) Without using SVD. (b) Using SVD.

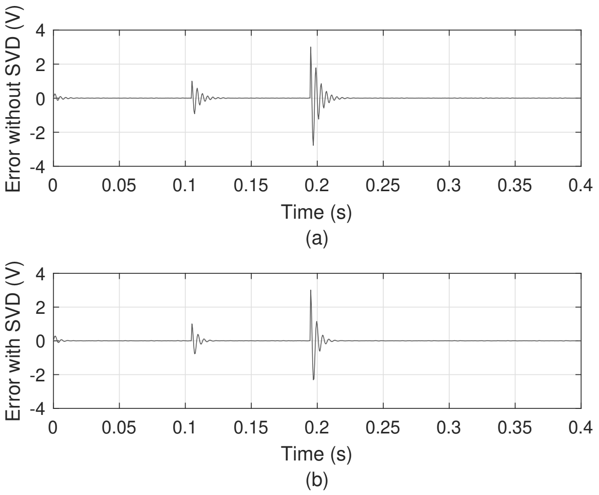


Figure 7. Simulation test: Error for $n_p = 100$, $n_c = 80$ and $r_w = 0.1$, without adding noise. (a) Without using SVD. (b) Using SVD.

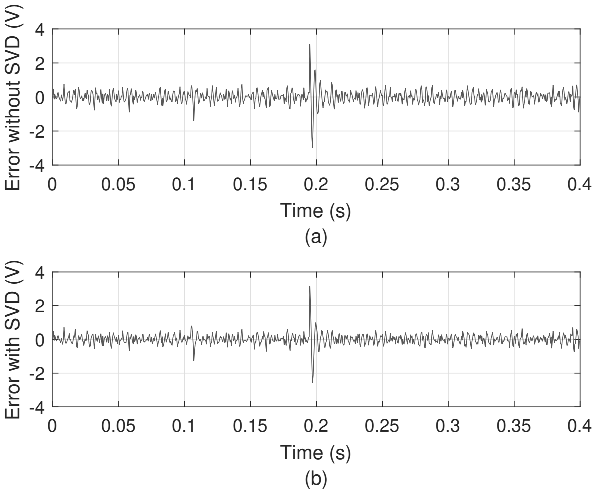


Figure 8. Simulation test: Error for $n_p = 100$, $n_c = 80$ and $r_w = 0.1$, adding noise. (a) Without using SVD. (b) Using SVD.

Table 3 shows each test's root mean square error (RMSE) and the settling time (ST). Note that the RMSE and the ST are smaller for the proposed approach.

Figures 9 to 14 show the control laws for each test. The application of SVD allows getting more stable control laws with robustness against noise and the effect of using few bits for representing numbers, as SVD allows getting a *well-conditioned* Hessian matrix. Besides, the results using SVD are similar to those obtained using 64-bit arithmetic operations.

The proposed GPC controller was implemented in the FPGA kit DE115 of ALTERA and tested through FPGA in-the-loop strategy: the controller (implemented in the FPGA) controls a plant designed in SIMULINK program executed in a PC. The FPGA and PC communication is done through an ethernet cable. Figure 15 shows the experimental setup. Figure 16 shows the experimental results for $n_p = 100$, $n_c = 50$ and $r_w = 0.001$. Note that SVD allows getting a more stable response.

Table 3. Results of The Simulation Tests

Configuration	Noise	SVD	RMSE	ST
1	No	No	0.2978	47.5 ms
1	No	Yes	0.2077	22.5 ms
1	Yes	No	0.5229	59.5 ms
1	Yes	Yes	0.3112	15.0 ms
2	No	No	0.8015	$>$ 200 ms
2	No	Yes	0.2394	16.5 ms
2	Yes	No	1.3692	$>$ 200 ms
2	Yes	Yes	0.4032	30.0 ms
3	No	No	0.2383	20.0 ms
3	No	Yes	0.2127	17.0 ms
3	Yes	No	0.3719	32.5 ms
3	Yes	Yes	0.3181	22.5 ms

- Configuration 1:$n_p = 20, n_c = 10$ And $r_w = 0.01$
- Configuration 2:$n_p = 100, n_c = 50$ And $r_w = 0.001$
- Configuration 3:$n_p = 100, n_c = 80$ And $r_w = 0.1$

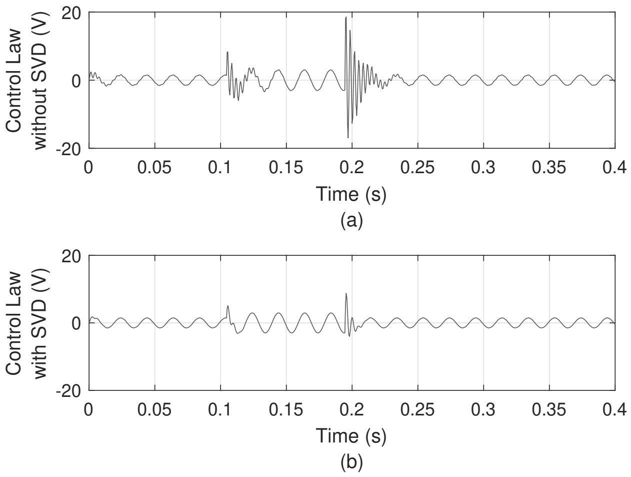


Figure 9. Simulation test: Control law for $n_p = 20$, $n_c = 10$ and $r_w = 0.01$, without adding noise. (a) Without using SVD. (b) Using SVD.

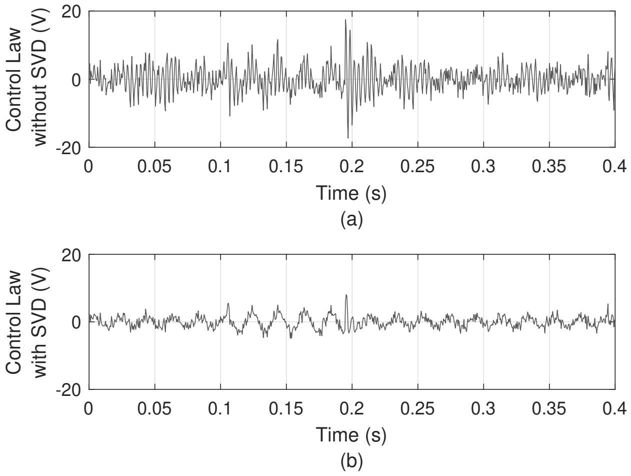


Figure 10. Simulation test: Control law for $n_p = 20$, $n_c = 10$ and $r_w = 0.01$, adding noise. (a) Without using SVD. (b) Using SVD.

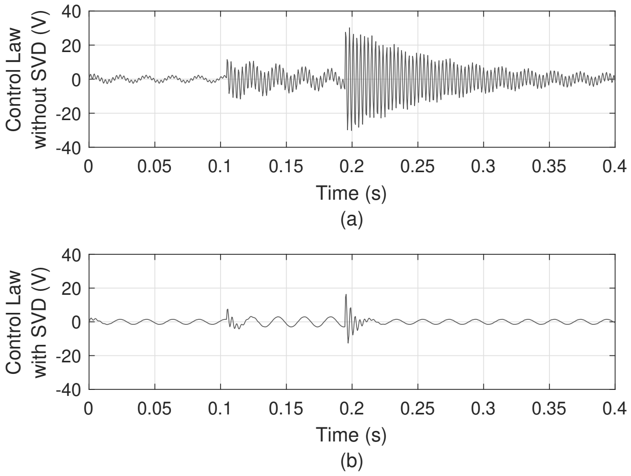
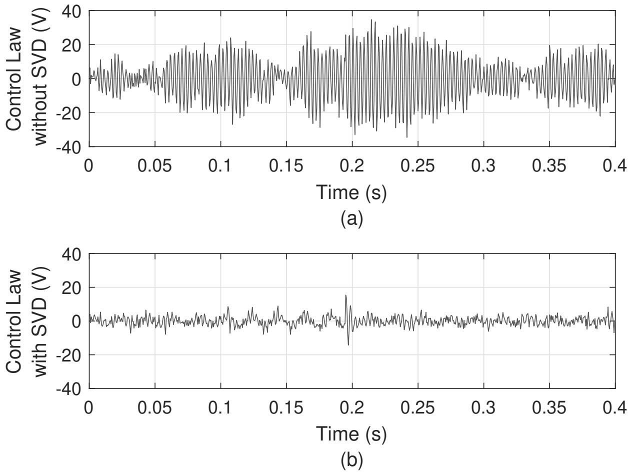


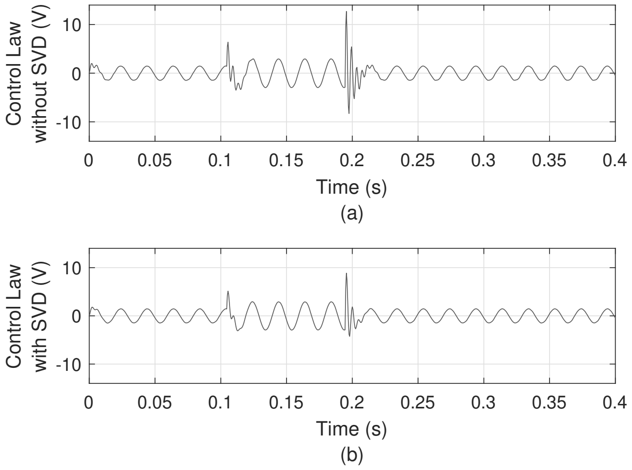
Figure 11. Simulation test: Control law for $n_p = 100$, $n_c = 50$ and $r_w = 0.001$, without adding noise. (a) Without using SVD. (b) Using SVD.



(a)

(b)

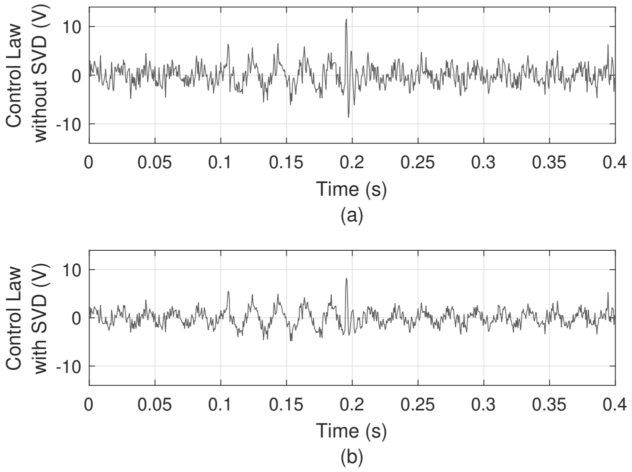
Figure 12. Simulation test: Control law for $n_p = 1$ $n_c = 50$ and $r_w = 0.001$, adding noise. (a) Witho using SVD. (b) Using SVD



(a)

(b)

Figure 13. Simulation test: Control law for $n_p = 100$, $n_c = 80$ and $r_w = 0.1$, without adding noise. (a) Without using SVD. (b) Using SVD.



(a)

(b)

Figure 14. Simulation test: Control law for $n_p = 100$, $n_c = 80$ and $r_w = 0.1$, adding noise. (a) Without using SVD. (b) Using SVD.

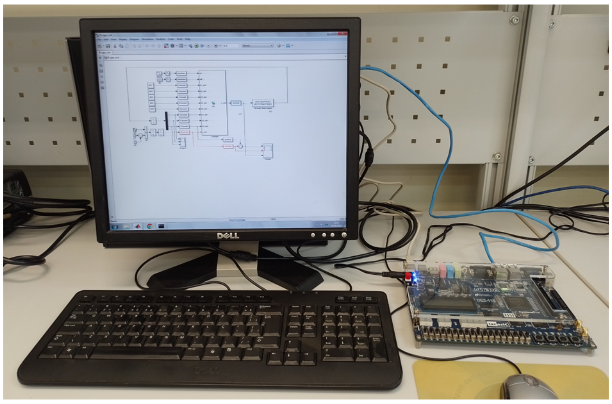
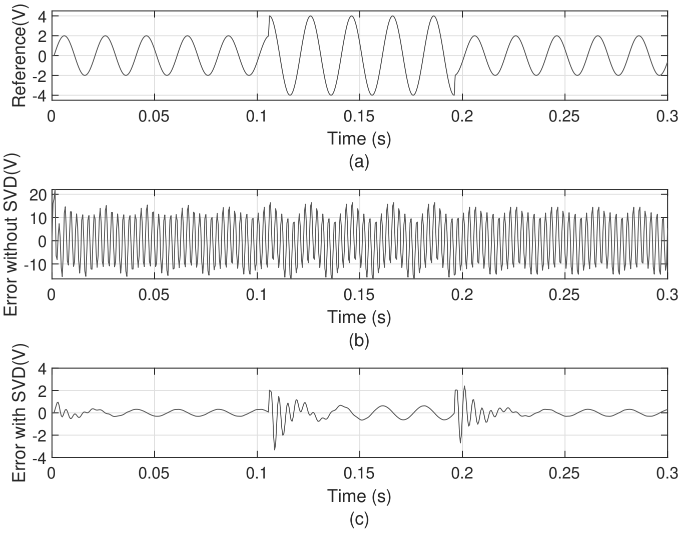


Figure 15. Experimental setup.



(a)

(b)

(c)

Figure 16. Experimental test for $n_p = 100$, $n_c = 50$ and $r_w = 0.001$. (a) Reference, (b) Without using SVD. (c) Using SVD.

5. DISCUSSION

The application of SVD to get the Hessian matrix that defines the GPC control law allows getting a more stable controller for different values of n_p, r_c, and r_w as proved through simulation and experimental tests. However, the structures of the augmented model, the cost function, the prediction formula in (7) and the GPC control law in (15) are not affected by the regularization technique (only their numerical values). Hence, applying restrictions and adapting the controller for MIMO (Multiple-Inputs Multiple-Output) plants is possible through the techniques explained in Wang (2009). Improvements in the design of restrictions in GPC isbeyond the objective of this work. However, he uses of restrictions will be done in future researches.

It is difficult to define an analytic relationship between the number of singular values used in regularization with the robustness against noise. In general, the influence of n_p, r_c, and r_w in the close-loop dynamics is difficult to analyze. As regularization affects the Hessian matrix's value, the number of singular values $\hat{\tau}_*$ should also be considered in the tuning of the predictive controller. A

Differential Evolution (DE) or a Genetic Algorithm (GA) could be used to define the values of n_p, r_c, r_w and $\hat{\tau}_*$. Many strategies exist to define the optimal SVD truncation according to the desired control performance. The fitness function used in a DE or a GA can be defined to evaluate that performance and look for the best regularization.

6. CONCLUSIONS

This paper proposes the application of Singular Value Decomposition (SVD) in estimating the GPC control law. When the Hessian matrix is *ill-conditioned*, i.e., when it has a large condition number, the GPC control law becomes sensitive to noise and quantification errors produced by using few bits for arithmetic operations. SVD regularization procedure and the proposed methodology to define the number of the largest singular values used in regularization allow a more robust (*well-conditioned*) Hessian matrix. The number of singular values depends on the applications. In many cases, a more robust control response can be obtained by reducing the response speed of the control system. However, simulations show that the proposed approach allows a faster and more robust response compared to the *ill-conditioned* GPC systems. The transient response depends on the GPC parameters (n_p, n_c and r_w), but also on the number of singular values used in regularization ($\hat{\tau}_*$). Further research should be done to select those parameters. Future works will explore the application of the proposed approach considering restrictions and MIMO systems.

ACKNOWLEDGE

Authors want to thanks Federal University of Mato Grosso do Sul, Federal University of Rio de Janeiro, CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and IEEE PELS UFMS Chapter for the support given to this research.

REFERENCES

Aidoud, M., Sedraoui, M., Feraga, C.E., and Sebbagh, A. (2019). Robustification of the generalized predictive law (gpc) by the implicit application of the h∞ method. In *Proceedings of the 9th International Conference on Information Systems and Technologies.*

Belsley, D.A. and Oldford, R. (1986). The general problem of ill conditioning and its role in statistical analysis. *Computational Statistics & Data Analysis*, 4(2), 103–120. doi:https://doi.org/10.1016/0167-9473(86)90014-9.

Camacho, E.F. and Bordons, C. (2007). *Model Predictive Control.* Springer, London, 2 edition.

Chen, T.S. (ed.) (1999). *Linear System Theory and Design.* Oxford University Press, 3^{rd} edition.

Cordero, R., Estrabis, T., Batista, E.A., Andrea, C.Q., and Gentil, G. (2021a). Ramp-tracking generalized predictive control system-based on second-order difference. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(4), 1283–1287. doi:10.1109/TCSII.2020.3019028.

Cordero, R., Estrabis, T., Brito, M.A., and Gentil, G. (2022). Development of a resonant generalized predictive controller for sinusoidal reference tracking. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(3), 1218–1222. doi:10.1109/TCSII.2021.3102535.

Cordero, R., Estrabis, T., Gentil, G., Batista, E.A., and Andrea, C.Q. (2021b). Development of a generalized predictive control system for polynomial reference tracking. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(8), 2875–2879. doi:10.1109/TCSII.2021.3058625.

Gavish, M. and Donoho, D.L. (2014). The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8), 5040–5053. doi:10.1109/TIT.2014.2323359.

Neumaier, A. (1998). Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM Review*, 40(3), 636–666. doi:10.1137/S0036144597321909.

Qin, S. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764. doi:https://doi.org/10.1016/S0967-0661(02)00186-7.

Sanchis, J., Martínez, M., Ramos, C., and Salcedo, J. (2002). Principal component gpc with terminal equality constraint. *IFAC Proceedings Volumes*, 35(1), 103–108.

Strang, G. (ed.) (1988). *Linear Algebra and its Applications.* Oxford University Press, 3^{rd} edition.

Wang, W. (ed.) (2009). *Model Predictive Control System Design and Implementation Using MATLAB.* Springer-Verlag London.