Adaptive Block Pole Placement For MIMO Systems

Belkacem Bekhiti[1], Bachir Nail[2] and Kamel Hariche[1]

[1]Institute of Aeronautics and Space Studies, (IASS Blida) BP 270 Route Soumâa ,BLIDA, Algeria
[2]Faculty of sciences and Technology, University of Ziane Achour, Djelfa. P 3117, Djelfa 17000, Algeria

Abstract
In this paper we have introduced a new digital adaptive control design algorithms based on the theory of matrix polynomials. the main contribution of this procedure is based on the so called block pole placement gathered with a MIMO RLS estimator, the advantages of this control are the non-interacted behavior, simplicity in control design and allowing to relocate not only the eigenvalues but both eigenstructure are adaptively assigned altering both stability and the rate convergence.

Keywords
Block roots, MIMO RLS, Diophantine equation, Matrix polynomials, MFD

1. Introduction

Design of controllers for multivariable systems requires an assessment of structural properties of transfer matrices and matrix polynomials see [1],[2],[3] and [4]. Unlike to the scalar cases zeros and gains in multivariable systems have directions which lead to the creation of new concepts called matrix fraction description (MFD)[5, 6, 7], Block poles, Block zeros etc... to analyze more easily and with less effort the multivariable compensator reaching a desired performances see [8] ,[9] and [10]. The combination of a pole placement control law with a parameter estimator or an adaptive law leads to an adaptive pole placement control (APPC) scheme that can be used to control a wide class of LTI plants with unknown parameters see [11] ,[12] and [13]. An extension of this idea to the more general case in MIMO systems described by either left or right matrix fraction lead to adaptive Block pole placement[14, 15, 16, 17, 18, 19].

In the present work a new MIMO adaptive compensator design procedure is proposed which offers the designer a larger degree of freedom[20, 21, 22] (more than the set of original desired eigenvalues can be assigned) and the algorithm is direct and allows assigning desired eigenstructure through block poles[23, 24]. Compared to the previous works and to the best of the authors' knowledge no body considered using adaptive block poles placement for digital systems described by matrix fractions to assign an eigenstructure using dynamic compensators[25, 26, 27].

The paper is organized as follow, the first section be an introductory to the present work. Then the second section will include some exist MIMO identification algorithms. It is followed with the section which deals with adaptive block pole placement (or adaptive matrix polynomial placement) via MIMO digital compensator design. In the fourth section we have illustrated an application example which is the discrete adaptive control of winding process via a digital compensator design based on the Block structure assignment[28]. Finally comments and a conclusion will finish the paper.

ICYRIME 2025: 10th International Conference of Yearly Reports on Informatics, Mathematics, and Engineering. Czestochowa, January 14-16, 2025
BNail@univ-djelfa.dz (B. Nail)

2. Some Exist MIMO Identification Algorithms

System identification will provide us with an approximate model which is often sufficient to achieve control goals, therefor we will introduce some exist MIMO identification algorithms [29], [30], [31] and [32].

2.1. MIMO Least Squares

A MIMO ARMAX (autoregressive moving average with exogenous excitation) mode

$$A(q^{-1})y[k] = B(q^{-1})u[k] + C(q^{-1})e[k] \tag{1}$$

can be written in LMFD form as:

$$y[k] = A(q^{-1})^{-1}B(q^{-1})u[k] + A(q^{-1})^{-1}C(q^{-1})e[k] \tag{2}$$

Where $u[k] \in R^m$ and $y[k] \in R^p$ are input and output vectors of the system respectively, while $e[k] \in R^p$ is a white-noise signal and the polynomial matrices $A(q^{-1})$, $B(q^{-1})$ and $C(q^{-1})$ have the following structure

$$A(q^{-1}) = I_p + A_1q^{-1} + \cdots + A_{na}q^{-na}$$
$$B(q^{-1}) = B_1q^{-1} + \cdots + B_{nb}q^{-nb}$$
$$C(q^{-1}) = I_p + C_1q^{-1} + \cdots + C_{nc}q^{-nc}$$

The objective is to identify the matrix coefficients $A_i \in R^{p\times p}$ and $B_i \in R^{p\times m}$ of the matrix polynomials

$A(q^{-1})$ and $B(q^{-1})$ assuming in this case $C(q^{-1}) = I_p$. Taking the transpose of equation (1) and expanding $A(q^{-1})$ and $B(q^{-1})$ yield:

$$\begin{aligned} e^T[k] = & \left(y^T[k] + y^T[k-1]A_1{}^T + \cdots + y^T[k-na]A_{na}{}^T\right) \\ & - \left(u^T[k-1]B_1{}^T + \cdots + u^T[k-nb]B_{nb}{}^T\right) \\ = & \; y^T[k] - \varphi^T[k]\theta \end{aligned}$$

Where

$$\begin{aligned} \varphi^T[k] &= \left[-y^T[k-1]\cdots - y^T[k-n_a], u^T[k-1]\cdots u^T[k-n_b]\right] \\ \theta &= \left[A_1{}^T \cdots A_{na}{}^T, B_1{}^T \cdots B_{nb}{}^T\right]^T \end{aligned}$$

so, the least squares estimate is

$$\hat{\theta_{ls}} = \Phi^T \Phi^{-1} \Phi^T Y \qquad (3)$$

Where

$$Y = \begin{bmatrix} y^T[n+1,:] \\ \vdots \\ y^T[M,:] \end{bmatrix}, \quad \Phi = \left[\Phi_y \vdots \Phi_u\right]$$

$$\Phi_y = \left[\begin{bmatrix} -y^T[n,:] \\ \vdots \\ -y^T[M-1,:] \end{bmatrix} \begin{matrix} \cdots \\ \cdots \\ \cdots \end{matrix} \begin{bmatrix} -y^T[n-n_a+1,:] \\ \vdots \\ -y^T[M-n_a,:] \end{bmatrix}\right]$$

$$\Phi_u = \left[\begin{bmatrix} u^T[n,:] \\ \vdots \\ u^T[M-1,:] \end{bmatrix} \begin{matrix} \cdots \\ \cdots \\ \cdots \end{matrix} \begin{bmatrix} u^T[n-n_b+1,:] \\ \vdots \\ u^T[M-n_b,:] \end{bmatrix}\right]$$

with $n = n_a$ and M is the number of I/O data. To avoid a large space memory and the large dimension matrix inversion taken by the simple least secure a MIMO recursive least square algorithm can be handled and elaborated to be used in digital software preserving the memory space.

2.2. MIMO Recursive Least Squares

A recursive implementation of the MIMO least squares can be written as an algorithm:

Algorithm (MIMO RLS Algorithm)
1- Initialize θ to zero
2- Let $P = c \times I$ Where c is a constant
3- For $k = n_a : N-1$,
$\psi = \left[-y^T[k,:]\cdots - y^T[k-n_a+1,:], u^T[k,:]\cdots u^T[k-n_b+1,:]\right]^T$
$G = (P\psi)(1+\psi^T P\psi)^{-1}$
$\theta = \theta + G(y^T(k+1,:) - \psi^T\theta)$
$P = (I - G\psi^T)P$
end

Example1: consider the next dynamical system with the following matrices.

$$A(q^{-1}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0.5 & -0.4 \\ 0.3 & -0.6 \end{bmatrix} q^{-1} + \begin{bmatrix} -0.1 & -0.3 \\ 0.2 & 0.3 \end{bmatrix} q^{-2}$$

$$B(q^{-1}) = \begin{bmatrix} -0.1 & -0.9 \\ 0.4 & 0.5 \end{bmatrix} q^{-1}$$

A PRBS data sequence of length $N = 1000$ is used to excite the system. A simulation experiment has been performed for signal to noise ratio equal to $20db$ for both outputs. The results of using the next algorithm:

$P = 1000 \times I$
For k=2:1000-1 do,
$\psi = \left[-y^T[k,:], -y^T[k-2+1,:], u^T[k,:]\right]^T$
$G = (P\psi)(1+\psi^T P\psi)^{-1}$
$\theta = \theta + G(y^T(k+1,:) - \psi^T\theta)$
$P = (I - G\psi^T)P$

are given below

$$\theta = \begin{bmatrix} A_1^T \\ A_2^T \\ B_1^T \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0.5015 & 0.3047 \\ -0.4006 & -0.5953 \end{bmatrix} \\ \begin{bmatrix} -0.0947 & 0.2053 \\ -0.2994 & 0.6957 \end{bmatrix} \\ \begin{bmatrix} -0.0992 & 0.4027 \\ -0.8998 & 0.5021 \end{bmatrix} \end{bmatrix}$$

Where: The dimension of the matrices are $\dim(\psi) = 6\times 1$, $\dim(G) = 6\times 1$, $\dim(P) = 6\times 6$, $\dim(\theta) = 6\times 2$ and $\dim(I) = 6\times 6$.

Comments:

▪ The MIMO Recursive least squares reduces the computation load associated with MIMO least squares by casting it in recursive form which is useful for On-line system identification.

▪ This basic RLS can be improved by introducing a forgetting factor [31] in order to give more weights to the most recent data.

2.3. MIMO Maximum Likelihood

For the previous given A MIMO ARMAX model the equation (1) can be developed to yield

$$\begin{aligned} C(q^{-1})e[k] = & \left(y[k] + A_1 y[k-1] + \cdots + A_{n_a} y[k-n_a]\right) \\ & - \left(B_1 u[k-1] + \cdots + B_{n_b} u[k-n_b]\right) \end{aligned}$$

Using the Kronecker operator we can be rewrite it as:

$$e[k] = I_p \otimes y^T[k] col(I_p) - [\eta_y, \eta_u, \eta_e]\,\theta \qquad (4)$$

Where:

$$\begin{aligned} \eta_y &= \; I_p \otimes y^T[k-1] + \cdots + I_p \otimes y^T[k-n_a] \\ \eta_u &= -I_p \otimes u^T[k-1] - \cdots - I_p \otimes u^T[k-n_b] \\ \eta_e &= -I_p \otimes e^T[k-1] - \cdots - I_p \otimes e^T[k-n_c] \end{aligned}$$

$$\begin{aligned}\theta &= [\theta_A, \theta_B, \theta_C]^T \\ \theta_A &= \left[col(A_1^T)^T \cdots col(A_{n_a}^T)^T\right] \\ \theta_B &= \left[col(B_1^T)^T \cdots col(B_{n_b}^T)^T\right] \\ \theta_C &= \left[col(C_1^T)^T \cdots col(C_{n_c}^T)^T\right]\end{aligned}$$

The best estimate of the parameter vector $\hat{\theta}$ can be obtained using a numerical minimization algorithm such as:

- Steepest descent method: $\theta_{k+1} = \theta_k - \lambda\nabla^T E$
- Gauss Newton method: $\theta_{k+1} = \theta_k - (\nabla^T\nabla)^{-1}\nabla^T E$

With

$$\nabla = \begin{bmatrix} \dfrac{\partial e[m+1]}{\partial\theta^T} \\ \vdots \\ \dfrac{\partial e[N]}{\partial\theta^T}\end{bmatrix}, \quad E = \begin{bmatrix} e[m+1] \\ \vdots \\ e[N]\end{bmatrix}$$

MIMO ML Algorithm:

Step1:

For $k = m+1$ to N

- Compute the prediction error

$\hat{e}[k] = \hat{A}(q^{-1})y[k] - \hat{B}(q^{-1})u[k] - C_1\hat{e}[k-1] \cdots - C_{n_c}\hat{e}[k-n_c]$

- Compute the partial derivatives of $e[k]$: $\dfrac{\partial e[k]}{\partial\theta^T}$

The elements of $\dfrac{\partial e[k]}{\partial\theta^T}$ can be computed through MIMO IIR (Infinite Impulse Response) digital filtering using the updated matrix coefficients estimates $\hat{C}_i$ of the matrix polynomial $\hat{C}(q^{-1})$

Step2: Estimate the parameter vector θ using

$\theta_{k+1} = \theta_k - \lambda\nabla^T E$ or $\theta_{k+1} = \theta_k - (\nabla^T\nabla)^{-1}\nabla^T E$

with $m = n_a$, $0 < \lambda < 1$ and N is the number of I/O data.

Step3: If no convergence, go to step1.

Example2: Let's consider the 2-input 2-output process (ie, p=m=2) described in LMFD by its polynomial matrices as

$$\begin{aligned} A(q^{-1}) &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + \begin{bmatrix}0.5 & -0.4\\ 0.3 & -0.6\end{bmatrix}q^{-1} + \begin{bmatrix}-0.1 & -0.3\\ 0.2 & 0.3\end{bmatrix}q^{-2} \\ B(q^{-1}) &= \begin{bmatrix}-0.1 & -0.9\\ 0.2 & 0.3\end{bmatrix}q^{-1} + \begin{bmatrix}-0.8 & -0.3\\ 0.1 & 0.7\end{bmatrix}q^{-2} \\ C(q^{-1}) &= \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + \begin{bmatrix}0.7 & 0.2\\ 0.3 & -0.9\end{bmatrix}q^{-1} + \begin{bmatrix}0.3 & 0.4\\ -0.5 & 0.7\end{bmatrix}q^{-2}\end{aligned}$$

The aim is to estimate the matrix polynomials , and $A(q^{-1})$ $B(q^{-1})$ and $C(q^{-1})$ from I/O data contaminated by white noise. A PRBS data sequence of length $N = 1000$ is used to excite the system. A simulation experiment has been performed for signal to noise ratio equal to $20db$ for both outputs.

$$\begin{aligned} \frac{\partial e[k]}{\partial(\theta_A)} &= \left[\frac{\partial e[k]}{\partial(col(A_1^T))^T}, \frac{\partial e[k]}{\partial(col(A_2^T))^T}\right] \\ &= \left[C(q^{-1})^{-1}\left[I_p \otimes y^T[k-1]\right], C(q^{-1})^{-1}\left[I_p \otimes y^T[k-2]\right]\right] \\ \frac{\partial e[k]}{\partial(\theta_B)} &= \left[\frac{\partial e[k]}{\partial(col(B_1^T))^T}, \frac{\partial e[k]}{\partial(col(B_2^T))^T}\right] \\ &= -\left[C(q^{-1})^{-1}\left[I_p \otimes u^T[k-1]\right], C(q^{-1})^{-1}\left[I_p \otimes u^T[k-2]\right]\right] \\ \frac{\partial e[k]}{\partial(\theta_C)} &= \left[\frac{\partial e[k]}{\partial(col(C_1^T))^T}, \frac{\partial e[k]}{\partial(col(C_2^T))^T}\right] \\ &= -\left[C(q^{-1})^{-1}\left[I_p \otimes e^T[k-1]\right], C(q^{-1})^{-1}\left[I_p \otimes e^T[k-2]\right]\right]\end{aligned}$$

and finally we can form $\dfrac{\partial e[k]}{\partial(\theta)^T}$ as:

$$\frac{\partial e[k]}{\partial(\theta)^T} = \left[\frac{\partial e[k]}{\partial(\theta_A)}, \frac{\partial e[k]}{\partial(\theta_B)}, \frac{\partial e[k]}{\partial(\theta_C)}\right]$$

Where it elements can be computed through MIMO IIR (Infinite Impulse Response) digital filtering using the updated matrix coefficients estimates $\hat{C}_1$,$\hat{C}_2$ of the matrix polynomial:

$$\hat{C}(q^{-1}) = I_2 + \hat{C}_1 q^{-1} + \hat{C}_2 q^{-2}$$

Then using the Gauss Newton method to update the parameter vector θ gives the results shown below:

$$\hat{\theta}_A{}^T = \begin{bmatrix}0.4932\\ -0.4055\\ 0.2963\\ -0.6029\\ -0.1022\\ -0.2968\\ 0.1975\\ 0.3040\end{bmatrix}, \hat{\theta}_B{}^T = \begin{bmatrix}-0.1056\\ -0.8996\\ 0.1954\\ 0.3014\\ -0.8017\\ -0.2929\\ 0.1018\\ 0.7053\end{bmatrix}, \hat{\theta}_C{}^T = \begin{bmatrix}0.7225\\ 0.1946\\ 0.3023\\ 0.9205\\ 0.3202\\ 0.4200\\ -0.4972\\ 0.7518\end{bmatrix}$$

Comments: The use of the Kronecker operator , block filtering using MIMO IIR (Infinite Impulse Response) digital filtering using the updated matrix coefficients estimates are the main features of the MIMO Maximum Likelihood algorithm. Better process and noise dynamics estimates can be achieved by increasing the number of samples or increasing the signal to noise ratio.

3. Adaptive compensator design

3.1. Matrix Fraction Discerption

Matrix Faction Description (MFD) is a representation of a matrix transfer function of a multivariable system as a ratio of two polynomial matrices. The MFD approach is based on the fact that the Transfer Function Matrices

$H(q^{-1})$ and $F(q^{-1})$ of a MIMO system described by the vector difference equation

$$y[k] = H(q^{-1})u[k] + F(q^{-1})e[k] \quad (5)$$

can be represented as ratio of two polynomial matrices. However, because matrices do not commute in general, we note that there are two representations for the transfer function matrix $H(q^{-1})$ (or $F(q^{-1})$) as a ratio of two polynomial matrices [8] which are:

• Right Matrix Fraction Description (RMFD)

$$H(q^{-1}) = C(q^{-1})D(q^{-1})^{-1} \quad (6)$$

• Left Matrix Fraction Description (LMFD)

$$H(q^{-1}) = A(q^{-1})^{-1}B(q^{-1}) \quad (7)$$

Where the matrix polynomials $A(q^{-1}), B(q^{-1}), C(q^{-1})$ and $D(q^{-1})$ have the following structures

$$\begin{aligned} A(q^{-1}) &= I_p + A_1 q^{-1} + ... + A_{n_a} q^{-n_a} \\ B(q^{-1}) &= B_0 + B_1 q^{-1} + ... + B_{n_b} q^{-n_b} \\ C(q^{-1}) &= C_0 + C_1 q^{-1} + ... + C_{n_c} q^{-n_c} \\ D(q^{-1}) &= I_m + D_1 q^{-1} + ... + D_{n_d} q^{-n_d} \end{aligned}$$

The matrix coefficients have the following dimensions: $A_i \in \Re^{p\times p}, B_i \in \Re^{p\times m}, C_i \in \Re^{p\times m}$ and $D_i \in \Re^{m\times m}$

Remark1: it is possible to obtain either LMFD or RMFD from the other only by solving the following matrix equation

$$A(q^{-1})C(q^{-1}) = B(q^{-1})D(q^{-1}) \quad (8)$$

This last matrix equality can be expanded and rewritten in more compact form after rearrangement into

$$S_{AB}S_{CD} = S_B$$

Where:

S_{AB} is the Silvester matrix and

$$S_{CD} = \left[{C_1}^T, {C_2}^T, \cdots, {C_{n_b}}^T, -{D_1}^T, \cdots, -{D_{n_a}}^T\right]^T$$

$$S_B = \left[{B_1}^T, {B_2}^T, \cdots, {B_{n_b}}^T, {O_{p\times m}}^T, {O_{p\times m}}^T, \cdots, {O_{p\times m}}^T\right]^T$$

and the solution vector is

$$S_{CD} = S_{AB}^{+} S_B$$

3.2. Nonadaptive Compensator Design

Consider now the unity feedback system in the next figure. The plant is described by a $p \times m$ proper rational matrix (RMFD)

$$H(q^{-1}) = C(q^{-1})D(q^{-1})^{-1} \quad (9)$$

The compensator to be designed is required to have a $m \times p$ proper rational matrix (LMFD).

$$G_c(q^{-1}) = D_c(q^{-1})^{-1}N_c(q^{-1}) \quad (10)$$

Hence the closed-loop transfer matrix is:

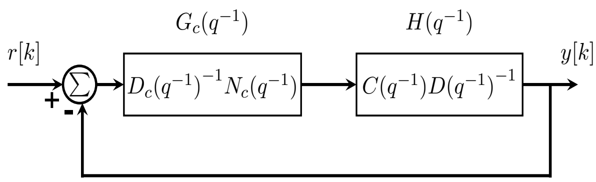

Figure 1: RMFD Compensator Structure

$$G_{cl}(q^{-1}) = \left(I + H(q^{-1})G_c(q^{-1})\right)^{-1} H(q^{-1})G_c(q^{-1}) \quad (11)$$

Using the identity:$(I + AB)^{-1}A = A(I + BA)^{-1}$ we get:

$$G_{cl}(q^{-1}) = H(q^{-1})\left(I + G_c(q^{-1})H(q^{-1})\right)^{-1} G_c(q^{-1}) \quad (12)$$

Which can be written as:

$$G_{cl}(q^{-1}) = C(q^{-1})D_f(q^{-1})^{-1}N_c(q^{-1}) \quad (13)$$

Where $D_f(q^{-1})$ is called the Diophantine matrix equation and defined by the next formula:

$$D_f(q^{-1}) = D_c(q^{-1})D(q^{-1}) + N_c(q^{-1})C(q^{-1}) \quad (14)$$

Hence the design problem becomes: Given $D(q^{-1})$ and $C(q^{-1})$ and an arbitrary $D_f(q^{-1})$, find $D_c(q^{-1})$ and $N_c(q^{-1})$ to satisfy this compensator equation. We note that the roots of $D_f(q^{-1})$ are the poles of the closed-loop transfer matrix $G_{cl}(q^{-1})$, and the solvents of $D_f(q^{-1})$ are block-poles of $G_{cl}(q^{-1})$. The compensator design, to achieve arbitrary block pole placement for the feedback configurations described previously, requires the solution of the compensator equation (14). Various numerical algorithms, for solving the Diophantine equation, have been developed and different approaches have been attempted [33] and [34]. The method proposed in this section is developed from the results obtained by Chen [8]. The idea is basically to transform the given matrices into a set of linear algebraic equations, which leads to the construction of a Sylvester matrix (or a generalized resultant matrix of $\{B(q^{-1}), A(q^{-1})\}$). The solution is obtained by applying searching algorithms for linearly dependent rows of the obtained matrix.

The Recursive Search Algorithm:

Given a set of n-dimensional rows $T_1, T_2, ..., T_p$, an $n \times n$ matrix $P(k)$ is determined recursively for $k = 1, 2, ..., p$
1. initialize $P(0) = I_n$($n \times n$ identity matrix)
2. for $k = 1, 2, ...p$ do
if $T_k P(k-1) T_k^T \neq 0$, then

$$P(k) = P(k-1) - \frac{\left[P(k-1)T_k^T\right]\left[P(k-1)T_k^T\right]^T}{T_k P(k-1) T_k^T}$$

and T_k is linearly independent of the previous rows
else $P(k) = P(k-1)$
and T_k is linearly dependent.

3.3. Adaptation Mechanism Devolvement

Classical controllers cannot solve the problem of uncertainties in dynamic systems, because the change in the process parameters cause a change in that operating conditions which leads to technical matters in the system (Instability and undesired performance) see [12] and [13]. Hence the adaptive control theory arise naturally when we surprised by this matter of uncertainties, therefore a typical problem is a parameter adjustment rule that is guaranteed to results in a stable closed loop system. Adaptive controllers can be divided into two main groups called model reference adaptive system (MRAS) and the self-tuning regulators (STRs). In this work we focus on the second category which is based on the parametric estimation, and the next figure show the overall mechanism. **Algorithm:**

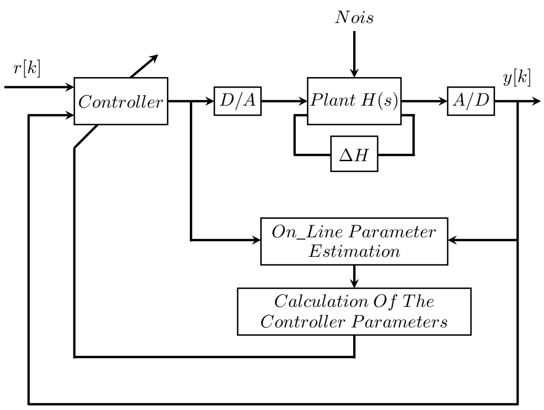

Figure 2: Indirect adaptive control Structure

Step1:

-Enter the values of: $M,\ n_a,\ n_b,\ P = c.I$
-Enter the nominal values of the $D_i \in R^{m\times m}$ and $C_i \in R^{p\times m}$
-Initiate $\hat{D}(q^{-1})$ and $\hat{C}(q^{-1})$ by the values of $D_i \in R^{m\times m}$ and $C_i \in R^{p\times m}$

For $k = n_a : M$ do

Step2:

Enter the desired Block poles $R_{id} \in R^{m\times m}$ to be placed and construct the corresponding matrix polynomial $D_f(q^{-1})$
Then compose the Diophantine equation as:

$$D_f(q^{-1}) = \left(\hat{D}_c(q^{-1})\hat{D}(q^{-1}) + \hat{N}_c(q^{-1})\hat{C}(q^{-1})\right)$$

Now solve the Diophantine equation using recursive search algorithm we obtain $\hat{D}_c(q^{-1})$ and $\hat{N}_c(q^{-1})$

Step3:

Give the desired trajectory sequence $r[k]$.
Compute the closed loop output and the control law by:

$$y[k] = \hat{C}(q^{-1})\left(\hat{D}_c(q^{-1})\hat{D}(q^{-1}) + \hat{N}_c(q^{-1})\hat{C}(q^{-1})\right)^{-1}\hat{N}_c(q^{-1})r[k]$$
$$u[k] = \hat{D}_c(q^{-1})^{-1}\hat{N}_c(q^{-1})u_c[k]$$
$$u_c[k] = r[k] - y[k]$$

Step4:

Identify the plant parametrs using MIMO-RLS:

$$\psi = \left[-y^T[k,:]\cdots - y^T[k-n_a+1,:], u^T[k,:]\cdots u^T[k-n_b+1,:]\right]^T$$

$$\begin{aligned} G &= (P\psi)(1+\psi^T P\psi)^{-1} \\ \hat{\theta}_{AB} &= \hat{\theta}_{AB} + G(y^T(k+1,:) - \psi^T\hat{\theta}_{AB}) \\ P &= (I - G\psi^T)P \end{aligned}$$

Step5:

Updating the matrix coefficients $\theta_{AB} = \hat{\theta}_{AB}$.
Convert LMFD to RMFD using Silvester Matrix equation

$$\begin{aligned} \hat{\theta}_{CD} &= f\left(A(q^{-1}), B(q^{-1})\right) \\ &= f(\theta_{AB}) \end{aligned}$$

Getting $\hat{C}_i \in R^{p\times m}$ and $\hat{D}_i \in R^{m\times m}$ and go to Step:2

4. Application to Winding process

Winding systems are encountered in a wide variety of industrial plants such as rolling mills in the steel industry, plants involving web conveyance including coating, paper-making and polymer film extrusion processes. The main role of a winding process is to control the web conveyance in order to avoid the effects of friction and sliding, as well as the problems of material distortion and can also damage the quality of the final product [35]. The illustrative example used here is modeled by identification of RMFD using constrained PEM see [36].

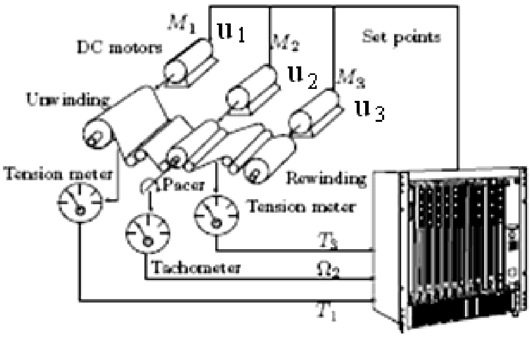

Figure 3: The winding process

The system inputs and outputs are:

u_1 : setpoint of motor current 1
u_2 : setpoint of motor angular speed 2
u_3 : setpoint of motor current 3
y_1 : web tension between motors 1 and 2 (T_1)
y_2 : web tension between motors 2 and 3 (T_3)
y_3 : motor angular speed 2 (Ω_2)

Referring to [36] the winding is described by the following RMFD model

$$y[k] = C(q^{-1})D(q^{-1})^{-1}u[k]$$

Where:

$$D(q^{-1}) = I_3 + D_1q^{-1} + D_2q^{-2}$$
$$C(q^{-1}) = C_1q^{-1} + C_2q^{-2}$$

With the following matrix coefficients:

$$D_1 = \begin{bmatrix} -2.2783 & -0.06775 & 0.55208 \\ 1.8518 & -1.6639 & -2.9525 \\ -1.1294 & -0.17974 & -0.18159 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 1.2801 & 0.067705 & -0.5535 \\ -1.8033 & 0.67218 & 2.8754 \\ 1.0791 & 0.17193 & -0.73911 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0.0042909 & 0.0063885 & -0.012831 \\ 0.066877 & 0.008738 & -0.12115 \\ 0.00077228 & 0.030226 & 0.00060105 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0.010399 & -0.0027112 & -0.015017 \\ -0.14411 & -0.020832 & 0.24284 \\ 0.048767 & -0.021682 & -0.079244 \end{bmatrix}$$

Remark2: In order to simplify the control procedure let we chose a fixed structure compensator of 1^{st} order With constant gain pre-compensator

$$u[k] = (\hat{D}_{c0} + \hat{D}_{c1}q^{-1})^{-1}(\hat{N}_{c0} + \hat{N}_{c1}q^{-1})(r[k] - y[k])$$
$$F = \lim_{q\to 1}\left[\hat{C}(q^{-1})D_f(q^{-1})^{-1}\hat{N}_c(q^{-1})\right]^{-1}$$

Then the desired matrix polynomial $D_f(q^{-1})$ is of order three, let we now chose thee Block roots (Solvents) to be placed

$$R_1 = \begin{pmatrix} 0.0000 & -0.0082 & 0.0033 \\ 0.0306 & -0.0533 & 0.0228 \\ 0.0028 & -0.0041 & -0.0045 \end{pmatrix}, \sigma_1(R_1) = \begin{pmatrix} -0.046 \\ -0.0053 \\ -0.0065 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} 0.0607 & -0.0201 & -0.0278 \\ 0.1411 & -0.0482 & -0.0718 \\ 0.0875 & -0.0267 & -0.0432 \end{pmatrix}, \sigma_2(R_2) = \begin{pmatrix} -0.033 \\ 0.0074 \\ -0.0051 \end{pmatrix}$$

$$R_3 = \begin{pmatrix} 0.0499 & -0.0223 & -0.0250 \\ 0.0156 & -0.0023 & -0.0133 \\ 0.0572 & -0.0285 & -0.0302 \end{pmatrix}, \sigma_3(R_3) = \begin{pmatrix} 0.0054 \\ -0.010 \\ 0.022 \end{pmatrix}$$

Where: $\sigma_1(R_1)$, $\sigma_2(R_2)$ and $\sigma_3(R_3)$ are the spectrum of those block roots. Hence to reconstruct the desired matrix polynomial we orient the reader to see [9] and [4].

$$D_f q^{-1} = I_3 + D_{f1}q^{-1} + D_{f2}q^{-2} + D_{f3}q^{-3}$$

Solving the Diophantine matrix equation yield to the linear system of equations:

$$\begin{bmatrix} \hat{D}_{c0}{}^T \\ \hat{D}_{c1}{}^T \\ \hat{N}_{c0}{}^T \\ \hat{N}_{c1}{}^T \end{bmatrix} = \begin{bmatrix} I_3^T & O_3^T & O_3^T & O_3^T \\ D_1{}^T & I_3^T & C_1{}^T & O_3^T \\ D_2{}^T & D_1{}^T & C_2{}^T & C_1{}^T \\ O_3^T & D_2{}^T & O_3^T & C_2{}^T \end{bmatrix}^{-1} \begin{bmatrix} I_3^T \\ D_{f1}{}^T \\ D_{f2}{}^T \\ D_{f3}{}^T \end{bmatrix}$$

The nominal values of the compensator coefficients are obtained according to this last equation.

Remark3: assuming that the system uncertainties are of 7% of the nominal one, means that $H(q^{-1}) = H_0(q^{-1}) + \Delta H(q^{-1})$
Now starting the adaptive block pole placement algorithm we obtain the next results as shown in figure It can be observed from the above simulation results that the algorithm developed in this paper able to assign block poles guaranteing the system stability even if some sudden uncertainties occurs and with smaller tracking errors. The influence of the parameter change don't affect the designed digital compensator due to the goodness of its adaptation mechanism, also simulation results show that relatively small interactions for the closed-loop system

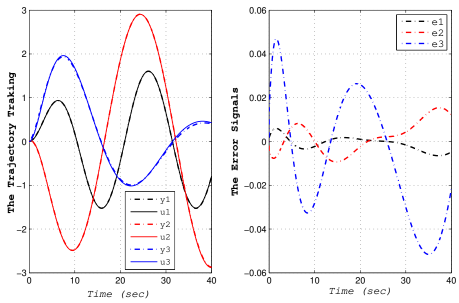

Figure 4: Adaptive Trajectory Tracking and Error Signals

when the setpoint of one of the variables is changed. Which means that the closed loop system is perfectly and complectly decoupled, This is because the control action produced by the MFD in both variables acts simultaneously on both manipulated variables as soon as a change in the reference of any of them is detected.

5. Conclusion

In this paper, a new adaptive Block pole-placement control for MIMO discrete-time systems has been considered. This control scheme includes the MIMO RLS estimation algorithm and Block pole-placement control, which is used not only to identify the unknown plant parameters but to achieve some specified performances. The proposed control scheme indeed improves both regulation and tracking error.

6. Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] K. Hariche, E. D. Denman, Interpolation theory and λ-matrices, Journal of Mathematical Analysis and Applications 143 (1989) 53–547.

[2] K. Hariche, E. D. Denman, On solvents and lagrange interpolating λ-matrices, Applied Mathematics and Computation 253 (1988) 21–332.

[3] K. Hariche, Interpolation Theory in the Structural Analysis of λ-matrices, Ph.D. thesis, University of Houston, 1987. Chapter 3.

[4] M. Yaici, K. Hariche, On solvents of matrix polynomials, International Journal of Modeling and Optimization 4 (2014).

[5] N. Brandizzi, A. Fanti, R. Gallotta, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Unsupervised pose estimation by means of an innovative vision transformer, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 13589 LNAI, 2023, p. 3 – 20. doi:10.1007/978-3-031-23480-4_1.

[6] B. Nail, B. Djaidir, I. E. Tibermacine, C. Napoli, N. Haidour, R. Abdelaziz, Gas turbine vibration monitoring based on real data and neuro-fuzzy system, Diagnostyka 25 (2024).

[7] B. Nail, M. A. Atoussi, S. Saadi, I. E. Tibermacine, C. Napoli, Real-time synchronisation of multiple fractional-order chaotic systems: an application study in secure communication, Fractal and Fractional 8 (2024) 104.

[8] C. T. Chen, Linear System Theory and Design, Holt, Reinhart and Winston, 1984.

[9] M. Yaici, K. Hariche, On eigenstructure assignment using block poles placement, European Journal of Control (2014).

[10] J. J. DiStefano, A. R. Stubberud, I. J. Williams, Theory and Problems of Feedback and Control Systems, McGraw-Hill, 1967.

[11] L. Ljung, T. Soderstrom, Theory and Practice of Recursive Identification, MIT Press, Cambridge, Massachusetts; London, England, 1987.

[12] I. D. Landau, R. Lozano, M. M'Saad, A. Karimi, Adaptive Control: Algorithms, Analysis and Applications, Springer-Verlag, London, 2011.

[13] P. A. Ioannou, J. Sun, Robust Adaptive Control: Design, Analysis and Robustness Bounds, Springer-Verlag, New York, 1991.

[14] B. Ladjal, I. E. Tibermacine, M. Bechouat, M. Sedraoui, C. Napoli, A. Rabehi, D. Lalmi, Hybrid models for direct normal irradiance forecasting: A case study of ghardaia zone (algeria), Natural Hazards 120 (2024) 14703–14725.

[15] C. Napoli, G. Pappalardo, E. Tramontana, Using modularity metrics to assist move method refactoring of large systems, in: Proceedings - 2013 7th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2013, 2013, p. 529 – 534. doi:10.1109/CISIS.2013.96.

[16] N. Brandizzi, A. Fanti, R. Gallotta, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Unsupervised pose estimation by means of an innovative vision transformer, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 13589 LNAI, 2023, p. 3 – 20. doi:10.1007/

978-3-031-23480-4_1.
[17] A. Tibermacine, D. Akrour, R. Khamar, I. E. Tibermacine, A. Rabehi, Comparative analysis of svm and cnn classifiers for eeg signal classification in response to different auditory stimuli, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–8.
[18] A. Alfarano, G. De Magistris, L. Mongelli, S. Russo, J. Starczewski, C. Napoli, A novel convmixer transformer based architecture for violent behavior detection, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 14126 LNAI, 2023, p. 3 – 16. doi:10.1007/978-3-031-42508-0_1.
[19] C. Napoli, V. Ponzi, A. Puglisi, S. Russo, I. Tibermacine, et al., Exploiting robots as healthcare resources for epidemics management and support caregivers, in: CEUR Workshop Proceedings, volume 3686, CEUR-WS, 2024, pp. 1–10.
[20] S. Russo, I. E. Tibermacine, A. Tibermacine, D. Chebana, A. Nahili, J. Starczewscki, C. Napoli, Analyzing eeg patterns in young adults exposed to different acrophobia levels: a vr study, Frontiers in Human Neuroscience 18 (2024). doi:10.3389/fnhum.2024.1348154.
[21] S. Falciglia, F. Betello, S. Russo, C. Napoli, Learning visual stimulus-evoked eeg manifold for neural image classification, Neurocomputing 588 (2024). doi:10.1016/j.neucom.2024.127654.
[22] A. Tibermacine, I. E. Tibermacine, M. Zouai, A. Rabehi, Eeg classification using contrastive learning and riemannian tangent space representations, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–7.
[23] S. Russo, S. Ahmed, I. E. Tibermacine, C. Napoli, Enhancing eeg signal reconstruction in cross-domain adaptation using cyclegan, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–8.
[24] A. TIBERMACINE, W. GUETTALA, I. E. TIBERMACINE, Efficient one-stage deep learning for text detection in scene images., Electrotehnica, Electronica, Automatica 72 (2024).
[25] S. Bouchelaghem, I. E. Tibermacine, M. Balsi, M. Moroni, C. Napoli, Cross-domain machine learning approaches using hyperspectral imaging for plastics litter detection, in: 2024 IEEE Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS), IEEE, 2024, pp. 36–40.
[26] N. Boutarfaia, S. Russo, A. Tibermacine, I. E. Tibermacine, Deep learning for eeg-based motor imagery classification: Towards enhanced human-machine interaction and assistive robotics, life 2 (2023) 4.
[27] I. Naidji, A. Tibermacine, W. Guettala, I. E. Tibermacine, et al., Semi-mind controlled robots based on reinforcement learning for indoor application., in: ICYRIME, 2023, pp. 51–59.
[28] S. eddine Boukredine, E. Mehallel, A. Boualleg, O. Baitiche, A. Rabehi, M. Guermoui, A. Douara, I. E. Tibermacine, Enhanced performance of microstrip antenna arrays through concave modifications and cut-corner techniques, ITEGAM-JETIA 11 (2025) 65–71.
[29] M. Akroum, K. Hariche, An optimal instrumental variable identification method for lmfd models, Studies in Informatics and Control 17 (2008).
[30] M. Akroum, K. Hariche, Extending the sriv algorithm to lmfd models, Journal of Electrical Engineering and Technology 4 (2009).
[31] L. Ljung, System Identification: Theory for the User, Prentice Hall, 1999.
[32] Y. Zhu, T. Backx, Identification of Multivariable Industrial Processes, Springer-Verlag, London, 1993.
[33] C. H. Fang, A simple approach to solving the matrix diophantine equation, IEEE Transactions on Automatic Control 37 (1992) 152–155.
[34] C. H. Fang, F. R. Chang, A novel approach for solving diophantine equations, IEEE Transactions on Automatic Control 37 (1990).
[35] T. Bastogne, et al., Multivariable identification of winding process by subspace methods for a tension control, Control Engineering Practice 6 (1998) 1077–1088.
[36] M. Akroum, K. Hariche, Right matrix fraction description models identification using constrained pem, Kuwait Journal of Science and Engineering 37 (2010) 69–88.