



ÉCOLE NATIONALE D'INGÉNIEURS DE TUNIS

Département Génie Industriel

Rapport projet Automatique

---

# Commande prédictive basée sur le modèle ARIMAX

---

*Elaboré par :*

Balti Chaima

Lakhzouri Roukaya

*Encadré par :*

M. Faouzi Bouani



2<sup>ème</sup> Année MINDS

Année universitaire : 2023/2024

# Remerciements

Nous tenons à exprimer notre gratitude envers toutes les personnes qui ont apporté leur soutien, leurs conseils ou leur participation, contribuant ainsi à la concrétisation de ce projet.

Nous souhaitons exprimer notre reconnaissance particulière envers M. Faouzi Bouani pour sa disponibilité et sa patience infaillibles, manifestées au moment où nous en avons le plus besoin.

# Table des matières

<b>Table des figures</b>	<b>3</b>
<b>I Recherche Bibliographique</b>	<b>5</b>
I.1 La commande prédictive : . . . . .	5
I.1.1 Définition et objectif : . . . . .	5
I.1.2 Principe : . . . . .	5
I.2 Commande prédictive généralisé : . . . . .	6
I.2.1 Principe : . . . . .	6
I.2.2 Présentation de modèle ARIMAX (CARIMA) : . . . . .	6
I.2.3 Utilisation des Équations Diophantiennes pour la Détermination de $\hat{y}$ . . . . .	7
I.2.4 Notation Matricielle : . . . . .	8
I.2.5 Détermination de l'expression de $u(k)$ : . . . . .	8
<b>II Simulation sur Matlab</b>	<b>10</b>
II.1 Système du premier ordre . . . . .	10
II.2 Système du Second Ordre . . . . .	13
<b>III Programmation sur carte Arduino</b>	<b>16</b>
III.1 Système du premier ordre . . . . .	16
III.1.1 Circuit Électrique du Système d'Ordre 1 . . . . .	16
III.1.2 Script Arduino pour le Système d'Ordre 1 . . . . .	18
III.1.3 Performance du Modèle ARIMAX Appliqué à un Système d'Ordre 1 . . . . .	19
III.2 Système du second ordre . . . . .	20
III.2.1 Circuit Électrique du Système d'Ordre 2 . . . . .	20
III.2.2 Script Arduino pour le Système d'Ordre 2 . . . . .	21
III.2.3 Performance du Modèle ARIMAX Appliqué à un Système d'Ordre 2 . . . . .	23
<b>Bibliographie</b>	<b>25</b>

# Table des figures

I.1	Diagramme temporel de la prédiction à l'horizon fini . . . . .	5
II.1	Système 1 <sup>er</sup> ordre Visualisation des Signaux Y, Yc et U dans le Temps	12
II.2	Système 2 <sup>ème</sup> ordre : Visualisation des Signaux Y, Yc et U dans le Temps . . . . .	15
III.1	Système 1 <sup>er</sup> ordre : circuit électrique . . . . .	17
III.2	Système 1 <sup>er</sup> ordre : Schéma du circuit électrique . . . . .	17
III.3	Système 1 <sup>er</sup> ordre : Visualisation des valeurs Pratique de Y, Yc et U dans le Temps . . . . .	19
III.4	Système 2 <sup>ème</sup> ordre : circuit électrique . . . . .	20
III.5	Système 2 <sup>ème</sup> ordre : Schéma du circuit électrique . . . . .	21
III.6	Système 2 <sup>ème</sup> ordre : Visualisation des valeurs Pratique de Y, Yc et U dans le Temps . . . . .	23

# Introduction

Dans le domaine de l'ingénierie et de l'automatisation, la commande prédictive émerge comme une discipline cruciale, révolutionnant la manière dont les systèmes sont gérés et optimisés.

Dans ce projet, nous visons à réaliser une exploration approfondie du thème "Commande prédictive basée sur le modèle ARIMAX" en suivant une démarche en trois étapes :

1. **Recherche bibliographique** : Par le biais d'une recherche approfondie, notre objectif sera de comprendre les concepts fondamentaux et les avancées récentes liés à ce sujet .
2. **Simulation MATLAB** : Nous mettrons en pratique nos connaissances en effectuant des simulations sous l'environnement MATLAB. Cela impliquera l'implémentation de l'algorithme de commande, ainsi que l'illustration de son fonctionnement au moyen d'un exemple de simulation concret.
3. **Transfert de l'algorithme vers la carte Arduino** : Nous réaliserons une implémentation pratique du modèle ARIMAX pour la commande prédictive en transférant l'algorithme étudié vers la carte Arduino. Cette étape vise à concrétiser les concepts théoriques dans un contexte réel.

# Chapitre I

## Recherche Bibliographique

### I.1 La commande prédictive :

#### I.1.1 Définition et objectif :

La commande prédictive consiste à prendre en compte, à l'instant présent, le comportement futur du système. Ceci en utilisant un modèle numérique du système pour prédire sa sortie dans le futur sur un horizon fini.

Le but de la stratégie prédictive est de faire coïncider la sortie du processus avec la consigne dans le futur sur un horizon fini.

Et voici un schéma simplifiant ce qui précède :

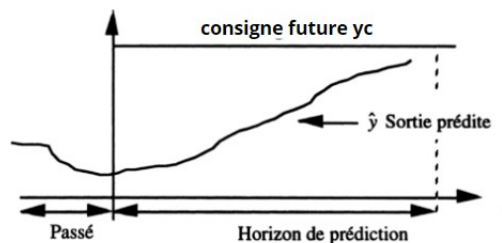


FIGURE I.1 – Diagramme temporel de la prédiction à l'horizon fini

#### I.1.2 Principe :

1. **Étape 1** : À l'instant présent, les sorties futures du système sont prédites (sur un horizon de taille  $N_y$ , dénommé horizon de prédiction sur la sortie) à l'aide d'un modèle de comportement du processus à commander. Ces prédictions

sont notées  $\hat{y}(k + j/k)$  avec  $j = 1, \dots, N_y$  pour indiquer la valeur de la sortie à l'instant  $k + j$ , calculée à l'instant  $k$ .

2. **Étape 2 :** Les futurs signaux de commande  $u(k + j/k)$ ,  $j = 0, \dots, N_u - 1$  avec  $N_u$  l'horizon de commande, sont calculés en optimisant la fonction de coût  $J$  qui représente la distance entre  $y_c(k + j)$  et  $\hat{y}(k + j)$ . (qui sera détaillée dans le paragraphe suivant).
3. **Étape 3 :** Seul le premier élément  $u(k/k)$ , du vecteur du signal de commande optimal  $u(k + j/k)$ ,  $j = 0, \dots, N_u - 1$  est appliqué au système pendant le prochain instant.

Dans ce qui suit, nous fixons l'horizon de prédiction à  $N_y = 3$  et l'horizon de commande à  $N_u = 1$ . Et voici une illustration de principe de commande prédictive : Ainsi, ce qui diffère d'une commande à une autre réside dans le modèle utilisé, la fonction coût  $J$ , et l'algorithme d'optimisation associé à cette fonction.

## I.2 Commande prédictive généralisé :

### I.2.1 Principe :

Le principe de commande reste toujours le même pour toutes les techniques de la commande prédictive, la différence principale réside dans le modèle utilisé pour la prédiction. Dans le cas de la commande prédictive généralisée (GPC), il est nécessaire d'avoir un modèle numérique de prédiction (dans notre cas ARIMAX) et la résolution récursive de l'équation diaphontienne.

### I.2.2 Présentation de modèle ARIMAX (CARIMA) :

Le modèle ARIMAX (**A**uto**R**egressive **I**ntegrated **M**oving **A**verage with **eX**ogenous variables) est une extension du modèle ARIMA (**A**uto**R**egressive **I**ntegrated **M**oving **A**verage). Ce modèle de prévision temporelle combine des composantes autorégressives (AR), des différences intégrées (I), et des moyennes mobiles (MA), en y ajoutant des variables exogènes (X) pour améliorer la précision des prédictions. Ce modèle prédit la sortie du système en se basant sur une représentation discrète donnée par l'équation :

$$a_0 y(k) + a_1 y(k-1) + \dots + a_{N_y} y(k-N_y) = b_0 u(k) + b_1 u(k-1) + \dots + b_{N_u} u(k-N_u) + \varepsilon(k),$$

où  $y(\cdot)$  représente la sortie,  $u(\cdot)$  la commande,  $\varepsilon(\cdot)$  le bruit.

avec  $a_i, b_i, i = 0, 1 \dots N_y$  étant les poids. En introduisant le facteur de retard temporel  $z^{-1}$  et l'opérateur  $\Delta = 1 - z^{-1}$ , l'équation ci-dessus peut être reformulée en :

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + \xi(k)/\Delta \quad (1.1)$$

avec

$$\begin{cases} A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_{n_a}z^{-n_a} \\ B(z^{-1}) = b_0 + b_1z^{-1} + \dots + b_{n_b}z^{-n_b} \\ \Delta = 1 - z^{-1} \end{cases}$$

### I.2.3 Utilisation des Équations Diophantiennes pour la Détermination de $\hat{y}$

Tout ce que nous voulons, c'est prédire la sortie au temps  $k$ , par exemple  $y(k)$ . Cependant, il n'est pas judicieux de diviser les deux côtés par  $A(z^{-1})$  car  $A(z^{-1})$  est un polynôme. Si nous pouvons multiplier les deux côtés par  $E_j$ , puis par certains termes  $z^{-j}F_j$ , qui satisfont

$$1 = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}), j = 1, \dots, N \quad (1.2)$$

D'où l'importance de résoudre cette équation qui constitue la première équation diophantienne, ayant comme inconnues les polynômes

$$\begin{aligned} E_j &= e_0 + e_1z^{-1} + \dots + e_{j-1}z^{-(j-1)} \\ F_j &= f_{j0} + f_{j1}z^{-1} + \dots + f_{jn_a}z^{-n_a} \end{aligned}$$

avec  $\deg(E_j) = j - 1$  et  $\deg(F_j) = \deg(A)$

Après la résolution de la première équation diophantienne et en multipliant les deux côtés de l'équation (1.1) par  $E_j(z^{-1})$ , puis en ajoutant  $z^{-j}F_j(z^{-1})$  des deux côtés de l'équation, on parvient à l'expression suivante :

$$\hat{y}(k+j | k) = E_j B \Delta u(k+j-1) + F_j y(k).$$

Le problème réside dans le fait que les valeurs  $\Delta u(i), i < k$  sont déjà attribuées au mécanisme de commande précédent et ne peuvent pas être réassignées. Par conséquent, nous sommes confrontés à la nécessité de séparer les valeurs déjà déterminées  $\Delta u(i), i < k$  de celles à concevoir  $\Delta u(i), k \leq i \leq k+j$ , d'où l'impératif de résoudre la deuxième équation diophantienne :

$$E_j B = G_j + z^{-j}H_j, \quad j = 1, \dots, N,$$

d'inconnues les polynômes :

$$\begin{aligned} H_j &= h_{j0} + h_{j1}z^{-1} + \dots + h_{jn_b-1}z^{-(n_b-1)}, \\ G_j &= g_0 + g_1z^{-1} + \dots + g_{j-1}z^{-(j-1)}. \end{aligned}$$



avec  $\deg(G_j) = j - 1$  et  $\deg(H_j) = \deg(B) - 1$   
D'où résulte :

$$\hat{y}(k + j | k) = G_j \Delta u(k + j - 1) + H_j \Delta u(k - 1) + F_j y(k),$$

### I.2.4 Notation Matricielle :

soit

$$\hat{Y}(k) = \begin{bmatrix} \hat{y}(k + 1 | k) \\ \hat{y}(k + 2 | k) \\ \dots \\ \hat{y}(k + N_y | k) \end{bmatrix}, \Delta U(k) = \begin{bmatrix} \Delta u(k) \\ \Delta u(k + 1) \\ \dots \\ \Delta u(k + N_u + 1) \end{bmatrix}, Y_0(k) = \begin{bmatrix} y_0(k + 1) \\ y_0(k + 2) \\ \dots \\ y_0(k + N_y) \end{bmatrix},$$

avec  $y_0(k + j) = H_j \Delta u(k - 1) + F_j y(k)$   
on a dans le cas ou  $N_u \leq N_y$  donc on peut ecrire

$$\hat{Y}(k) = Y_0(k) + G \Delta U(k),$$

avec

$$G = \begin{bmatrix} g_0 & & 0 \\ g_1 & \ddots & \\ \vdots & \ddots & g_0 \\ \vdots & & g_1 \\ g_{N_y-1} & \dots & g_{N_y-N_u} \end{bmatrix}_{N_y^* N_u}$$

### I.2.5 Détermination de l'expression de u(k) :

soit la fonction de cout  $J = \sum_{j=1}^N [\hat{y}(k + j | k) - y_c(k + j)]^2 + \lambda \sum_{j=1}^N [\Delta u(k + j - 1)]^2$   
soit

$$Y_c(k) = \begin{bmatrix} y_c(k + 1) \\ y_c(k + 2) \\ \dots \\ y_c(k + N_y) \end{bmatrix},$$

On peut donc exprimer J sous forme matricielle

$$J = [(\hat{Y}(k) - Y_c(k))]^T [\hat{Y}(k) - Y_c(k)] + \lambda [\Delta U(k)]^T [\Delta U(k)]$$

pour déterminer  $\Delta U_{opt}$  il faut  $\frac{\partial J}{\partial \Delta u(k)} = 0$  donc on obtient :

$$\Delta U(k)_{opt} = (G^T G + \lambda I)^{-1} G^T [Y d^{\wedge}(k) - Y_0(k)]$$

Prendre la première composante de  $\Delta U(k)$  comme  $\Delta u(k)$  qui représente la commande en temps  $k$ . Ainsi, la commande  $u(k)$  est :

$$u(k) = u(k-1) + \Delta u(k)$$

# Chapitre II

## Simulation sur Matlab

Le deuxième chapitre de ce rapport se concentre sur la partie de simulation réalisée à l'aide de MATLAB. L'objectif principal de cette section est de fournir une exploration détaillée de la commande prédictive basée sur le modèle ARIMAX.

### II.1 Système du premier ordre

Dans cette section, notre objectif est de réaliser une simulation MATLAB de la prédiction d'un système du premier ordre en utilisant le modèle ARIMAX.

$$\begin{aligned} H(p) &= \frac{1}{1 + 10p} \\ \Rightarrow H(z^{-1}) &= \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_0}{1 + a_1 z^{-1}}, \quad \text{avec} \quad \begin{cases} b_0 = 0.1 \\ a_1 = -e^{-0.1} \approx -0.905 \end{cases} \\ \Rightarrow H(z^{-1}) &= \frac{0.1}{1 - 0.905z^{-1}} \end{aligned}$$

Les solutions des équations diophantiennes :

$$\begin{aligned} F &= \begin{bmatrix} 1.905 & -0.905 \\ 2.724 & -1.724 \\ 3.465 & -2.465 \end{bmatrix} & E &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1.905 & 0 \\ 1 & 1.905 & 2.724 \end{bmatrix} & G &= \begin{bmatrix} 0.1 \\ 1.905 \\ 0.2724 \end{bmatrix} \\ H &= 0 \end{aligned}$$

```

1 clear all;
2 close all;
3 Ny      = 3;           %Longueur de Y
4 Nu      = 1;           %Longueur de U
5 %order 1
6 A       = [1, -0.905];
7 B       = [0.1];
8 na      = length(A) - 1; %degree de A
9 nb      = length(B) - 1; %degree de B
10 lambda = 2;
11 N = 100;
12 Yc=ones(N,1);
13 Yc(51:N,1)*=2;
14 Y=zeros(N,1);
15
16 %initialisation de F
17 F=zeros(Ny,na+1);
18 F(1,:)= [1-A(2), A(2)];
19 F(2,:)= [(A(2)-1)^2+A(2), A(2)*(1-A(2))];
20 F(3,:)= [(1-A(2))^3-2*A(2)*(A(2)-1), A(2)*((1-A(2))^2+A(2))];
21
22 %initialisation de E
23 E=zeros(Ny,Ny);
24 E(1,:)= [1,0,0];
25 E(2,:)= [1,(1-A(2)),0];
26 E(3,:)= [1,(1-A(2)),(1-A(2))^2+A(2)];
27
28 g=zeros(Ny,Nu);
29 g(:,1)= [B(1), (1-A(2))*B(1), B(1)*(A(2)+(-A(2)+1)^2)];
30
31 H=0
32
33 G1=zeros(Ny,Ny);
34 G1(:,1)=g(:,1);
35 G1(2:3,2)=g(1:2,1);
36 G1(3,3)=g(1,1);
37
38 G=G1(:,1:Nu);
39 u=zeros(N, 1);
40
41 du=1;
42
43 for k=3:N-3
44     Y(k)=-A(2)*Y(k-1)+B(1)*u(k-1);
45     for i=1:Ny
46         Y0(i) = F(i,1)*Y(k)+F(i,2)*Y(k-1)+H*du;
47     end
48 L=inverse(transpose(G)*G+lambda*eye(Ny-Nu));
49 V=L*(transpose(G)*(Yc(k+1:k+3)-transpose(Y0)));

```

```

50 du=V(1);
51 u(k)=du+u(k-1);
52 end
53
54 figure(1)
55 plot(1:97,Yc(1:97));
56 hold on
57 plot(1:97,Y(1:97));
58 hold on
59 plot(1:97,u(1:97));
60 xlabel("temps ")
61 ylabel("sortie")
62 legend('Yc(k)', 'Y(k)', 'U(k)');

```

Listing II.1 – Script MATLAB pour le modèle prédictif ARIMAX : système du premier ordre

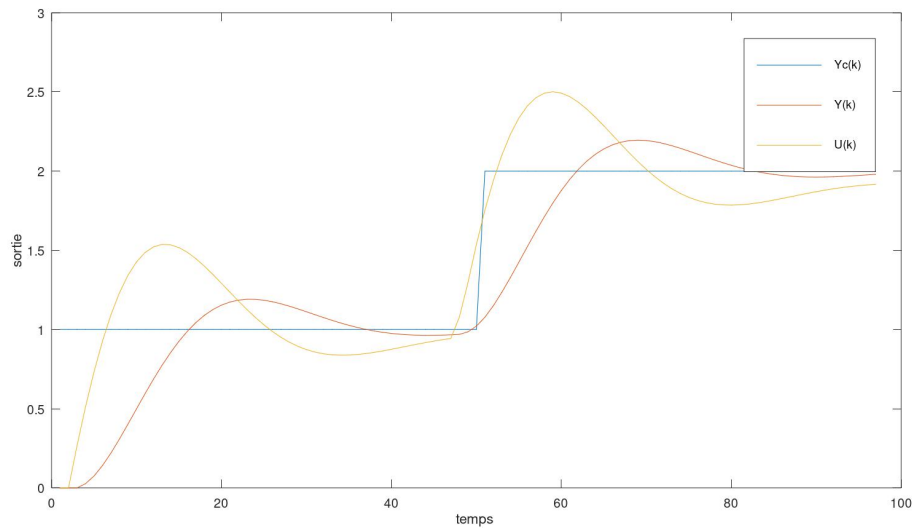


FIGURE II.1 – Système 1<sup>er</sup> ordre Visualisation des Signaux Y, Yc et U dans le Temps

## II.2 Système du Second Ordre

L'objectif principal de cette section réside dans la modélisation et la prédiction du comportement d'un système du second ordre en utilisant l'approche ARIMAX.

$$H(p) = \frac{1}{(1 + 10p)(1 + 2p)}$$

$$\Rightarrow H(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} , \quad \text{avec} \quad \begin{cases} b_0 = 0 \\ b_1 = 0.106 \\ a_1 = -1.511 \\ a_2 = 0.549 \end{cases}$$

$$\Rightarrow H(z^{-1}) = \frac{0.106 z^{-1}}{1 - 1.511 z^{-1} + 0.549 z^{-2}}$$

Les solutions des équations diophantiennes :

$$F = \begin{bmatrix} 2.511 & -2.06 & 0.549 \\ 4.2451 & -4.6237 & 1.3785 \\ 6.0358 & -7.3664 & 2.3306 \end{bmatrix} \quad E = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2.511 & 0 \\ 1 & 2.511 & 4.2451 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0.106 \\ 0.26617 \end{bmatrix}$$

$$H = \begin{bmatrix} 0.106 & 0.26617 & 0.44998 \end{bmatrix}$$

```

1 clear all;
2 close all;
3 Ny      = 3;           % Longueur de Y
4 Nu      = 1;           % Longueur de U
5 A       = [1, -1.511, 0.549];
6 B       = [0, 0.106];
7 na      = length(A) - 1; % degree de A
8 nb      = length(B) - 1; % degree de B
9 lambda  = 2;
10 N       = 400;
11 Yc=ones(N,1);
12 Yc(101:200,1)*=2;
13 Yc(301:N,1)*=2;
14 Y=zeros(N,1);
15
16 % initialisation de F
17 F=zeros(3,3);
18 F(1,:)= [1-A(2), A(2)-A(3), A(3)];
19 F(2,:)= [(A(2)-1)^2+A(2)-A(3), A(3)-(A(3)-A(2))*(1-A(2)), A(3)*(1-A(2))];
20 F(3,:)= [A(3)+(1-A(2))^3+2*(A(3)-A(2))*(A(2)-1), (1-A(2))*A(3)-(A(3)-A(2))*((1-A(2))^2-(A(3)-A(2))), A(3)*((1-A(2))^2-(A(3)-A(2)))]];
21
22 % initialisation de E
23 E=zeros(3,3);
24 E(1,:)= [1, 0, 0];
25 E(2,:)= [1, (1-A(2)), 0];
26 E(3,:)= [1, (1-A(2)), (1-A(2))^2-(A(3)-A(2))];
27
28 g=zeros(3,1);
29 g(:,1)= [B(1), ((1-A(2))*B(1))+B(2), B(2)*(1-A(2))+B(1)*(A(2)-A(3))+(-A(2)+1)^2)];
30
31 H=B(2)*[1, (1-A(2)), (1-A(2))^2-(A(3)-A(2))];
32
33 G1=zeros(3,3);
34 G1(:,1)=g(:,1);
35 G1(2:3,2)=g(1:2,1);
36 G1(3,3)=g(1,1);
37
38 G=G1(:,1:Nu);
39 u=zeros(N, 1);
40
41 du=1;
42
43 for k=3:N-3
44     Y(k)=-A(2)*Y(k-1)-A(3)*Y(k-2)+B(1)*u(k-1)+B(2)*u(k-2);
45     for i=1:Ny

```

```

46     Y0(i) = F(i,1)*Y(k)+F(i,2)*Y(k-1)+F(i,3)*Y(k-2)+H(i)*du;
47     end
48     L=inverse(transpose(G)*G+lambda*eye(Ny-Nu));
49     V=L*(transpose(G)*(Yc(k+1:k+3)-Y0'));
50     du=V(1);
51     u(k)=du+u(k-1);
52     end
53
54     figure(1)
55     plot(1:397,Yc(1:397));
56     hold on
57     plot(1:397,Y(1:397));
58     hold on
59     plot(1:397,u(1:397));
60     xlabel("temps ")
61     ylabel("sortie")
62     legend('Yc(k)', 'Y(k)', 'U(k)');

```

Listing II.2 – Script MATLAB pour le modèle prédictif ARIMAX : système du premier ordre

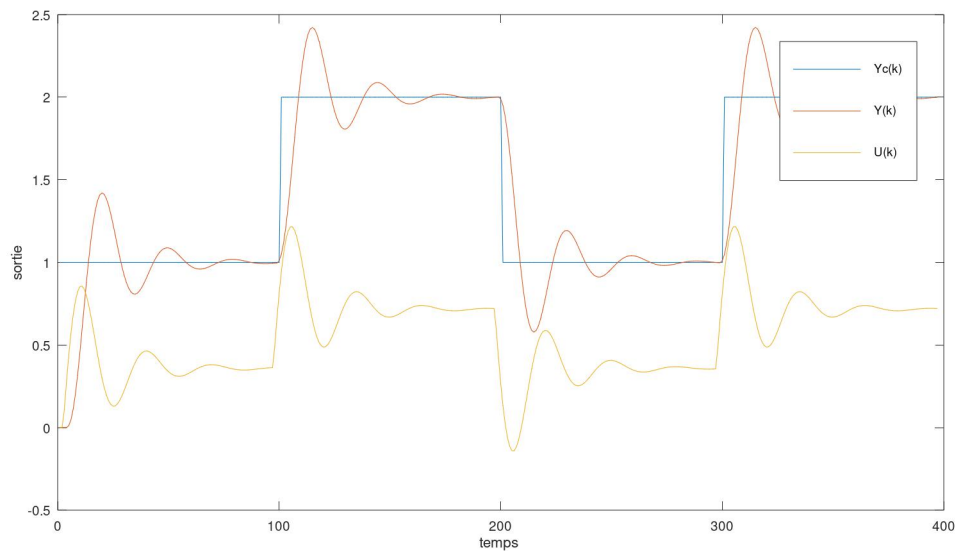


FIGURE II.2 – Système 2<sup>ème</sup> ordre : Visualisation des Signaux Y, Yc et U dans le Temps



# Chapitre III

## Programmation sur carte Arduino

### Introduction

Dans ce chapitre, nous abordons la transition de la simulation MATLAB à la programmation réelle sur une carte Arduino MEGA 2560. L'objectif est de mettre en œuvre le code de contrôle issu de notre modèle MATLAB sur une plateforme matérielle.

### III.1 Système du premier ordre

#### III.1.1 Circuit Électrique du Système d'Ordre 1

En explorant le système du premier ordre, nous avons introduit un filtre dans le circuit. De plus, pour permettre une interaction plus avancée et la collecte de données en temps réel, nous avons connecté le circuit à la carte Arduino Mega 2560.

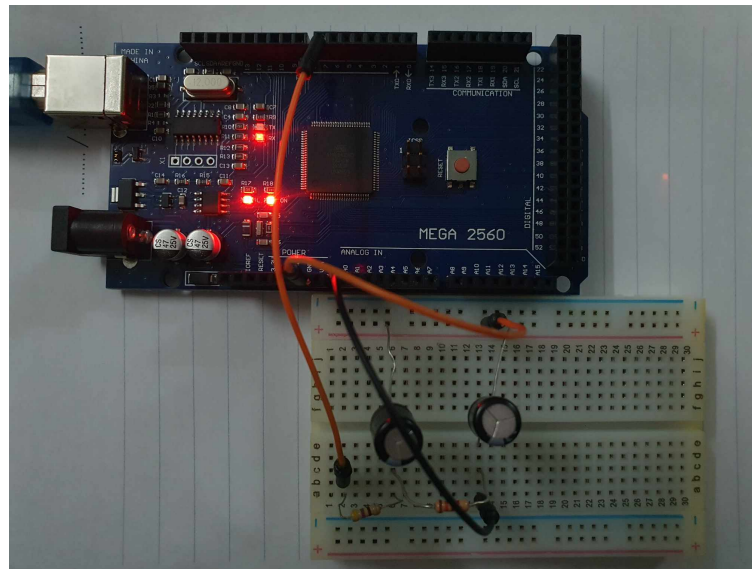


FIGURE III.1 – Système 1<sup>er</sup> ordre : circuit électrique

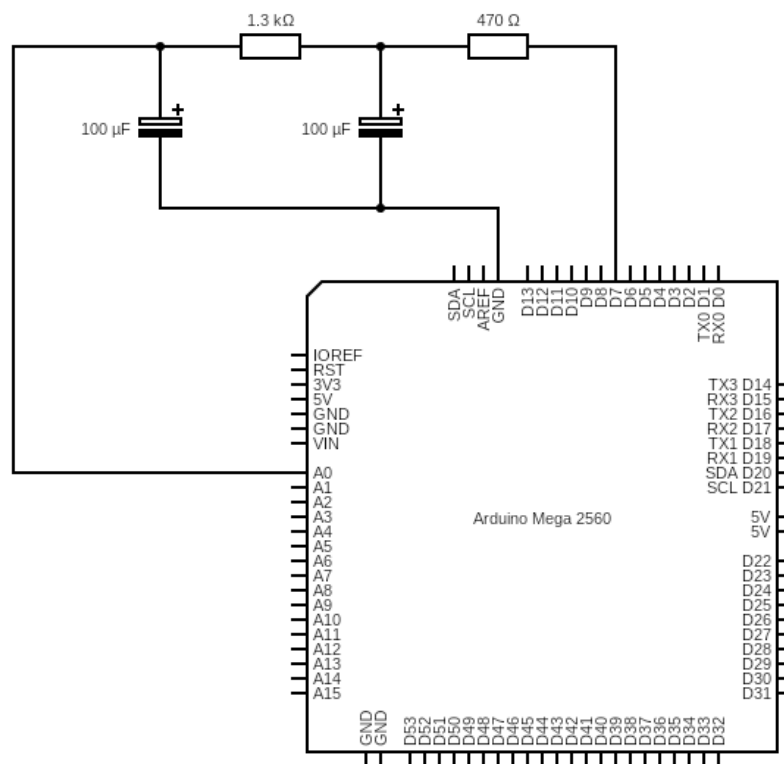


FIGURE III.2 – Système 1<sup>er</sup> ordre : Schéma du circuit électrique

### III.1.2 Script Arduino pour le Système d'Ordre 1

```
1  const int Ny = 3; // Length of Y
2  const int Nu = 1; // Length of U
3  const int analogInPin = A0; // Analog input pin
4  const int pwmOutPin = 7;
5  int N = 200;
6  double Yc[200];
7  double Y[200];
8  double u[200];
9  void setup()
10 {
11     Serial.begin(9600);
12
13     for (int i = 0; i < N; i=i+1)
14     {
15         Yc[i] = 1.0;
16         if (i > 100 && i <= 200)
17         {
18             Yc[i] = 2.0;
19         }
20     }
21
22     for (int i = 0; i < 200; i++)
23     {
24         u[i] = 0.0;
25     }
26     pinMode(analogInPin, INPUT);
27     pinMode(pwmOutPin, OUTPUT);
28 }
29
30 void loop()
31 {
32     double du;
33
34     for (int k = 2; k < 200 - 2; k++)
35     {
36         Y[k] = analogRead(analogInPin) * (5 / 1023.0);
37
38         double Y0[3];
39         Y0[0] = 1.905 * Y[k] - 0.905 * Y[k - 1];
40         Y0[1] = 2.724 * Y[k] - 1.724 * Y[k - 1];
41         Y0[2] = 3.4652 * Y[k] - 2.4652 * Y[k - 1];
42
43         double V;
44         V = 0.1 * (Yc[k + 0] - Y0[0]) + 0.1905 * (Yc[k + 1] - Y0[1]) +
45             0.2724 * (Yc[k + 2] - Y0[2]);
46         du = 0.47312 * V;
47         u[k] = du + u[k - 1];
```

```

47     analogWrite(pwmOutPin, u[k]*255/5);
48     Serial.println(Y[k]);
49     Serial.print("\t");
50     Serial.print(Yc[k]);
51     Serial.print("\t");
52     Serial.print(u[k]);
53     Serial.print("\t");
54     delay(50);
55 }
56 }

```

### III.1.3 Performance du Modèle ARIMAX Appliqué à un Système d'Ordre 1



FIGURE III.3 – Système 1<sup>er</sup> ordre : Visualisation des valeurs Pratique de Y, Yc et U dans le Temps

L'analyse des résultats du modèle prédictif démontre une adhérence marquée aux consignes établies. En effet, le suivi précis des prédictions dans le contexte d'un système d'ordre 1 renforce la fiabilité et l'efficacité du modèle ARIMAX.

## III.2 Système du second ordre

### III.2.1 Circuit Électrique du Système d'Ordre 2

De manière analogue à notre approche pour le système du premier ordre, nous avons intégré un filtre dans le circuit, le circuit du système du second ordre a également été connecté à la carte Arduino Mega 2560.

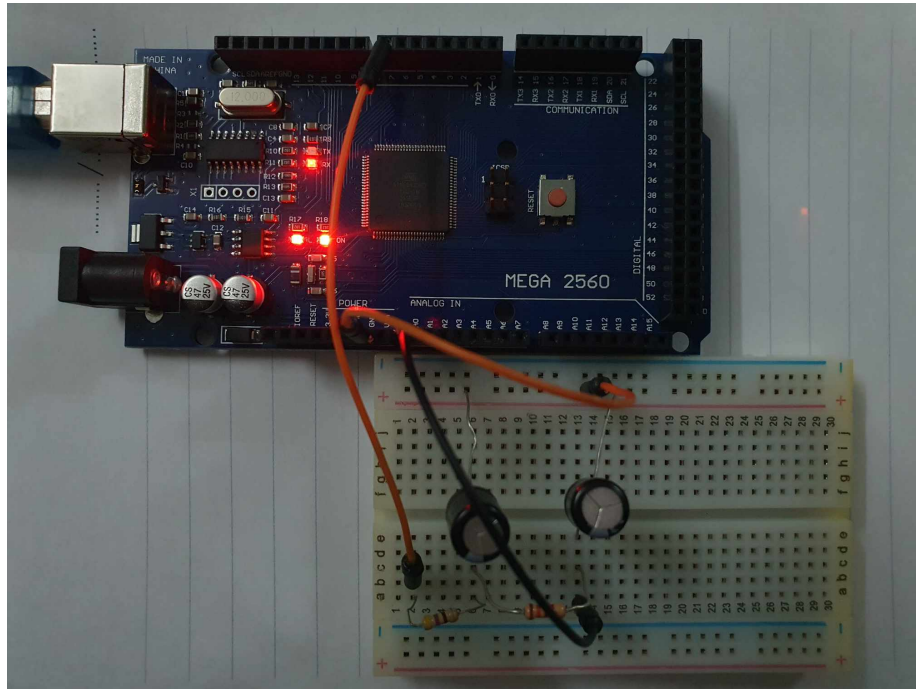


FIGURE III.4 – Système 2<sup>ème</sup> ordre : circuit électrique

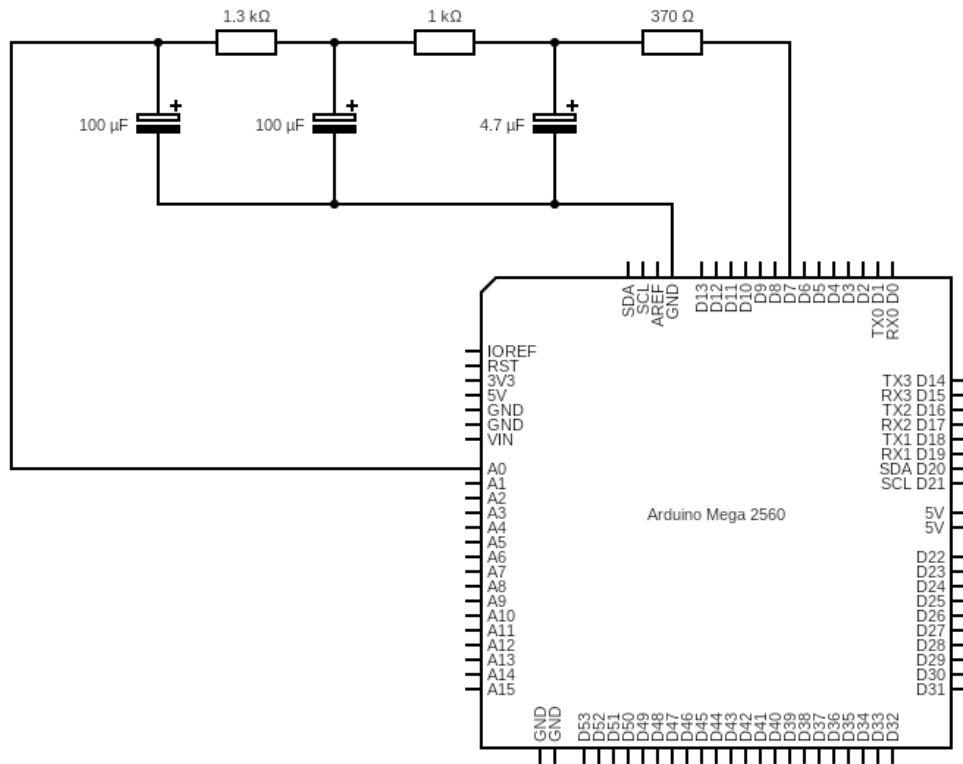


FIGURE III.5 – Système 2<sup>ème</sup> ordre : Schéma du circuit électrique

### III.2.2 Script Arduino pour le Système d'Ordre 2

```

1  const int Ny = 3; // Length of Y
2  const int Nu = 1; // Length of U
3  const int analogInPin = A0; // Analog input pin
4  const int pwmOutPin = 7;
5  int N = 600;
6  double Yc[600];
7  double Y[600];
8  double u[600];
9  float F[3][3] = {
10     {2.511, -2.06, 0.549},
11     {4.2451, -4.6237, 1.3785},
12     {6.0358, -7.3664, 2.3306}
13 };
14 float H[] = {0.106, 0.26617, 0.44998};
15 void setup()
16 {

```

```

17 Serial.begin(9600);
18
19 for (int i = 0; i < N; i=i+1)
20 {
21     Yc[i] = 1.0;
22     if (i > 100 && i <= 200)
23     {
24         Yc[i] = 2.0;
25     }
26     if (i > 300 && i <= 400)
27     {
28         Yc[i] = 2.0;
29     }
30     if (i > 500 && i <= 600)
31     {
32         Yc[i] = 2.0;
33     }
34 }
35
36 for (int i = 0; i < N; i++)
37 {
38     u[i] = 0.0;
39 }
40 pinMode(analogInPin, INPUT);
41 pinMode(pwmOutPin, OUTPUT);
42 }
43
44 void loop()
45 {
46     double du;
47
48     for (int k = 2; k < N - 2; k++)
49     {
50         Y[k] = analogRead(analogInPin) * (5 / 1023.0);
51
52         float Y0[Ny];
53         for (int i = 0; i < Ny; i++) {
54             Y0[i] = F[i][0] * Y[k] + F[i][1] * Y[k - 1] + F[i][2] * Y[k
- 2] + H[i] * du;
55         }
56
57         double V;
58         V = 0.106 * (Yc[k + 1] - Y0[1]) + 0.26617 * (Yc[k + 2] - Y0
[2]);
59         du = 0.48104 * V;
60         u[k] = du + u[k - 1];
61         analogWrite(pwmOutPin, u[k]*255/5);
62         Serial.println(Y[k]);
63         Serial.print("\t");

```

```

64     Serial.print(Yc[k]);
65     Serial.print("\t");
66     Serial.print(u[k]);
67     Serial.print("\t");
68     delay(50);
69 }
70 }

```

### III.2.3 Performance du Modèle ARIMAX Appliqué à un Système d'Ordre 2

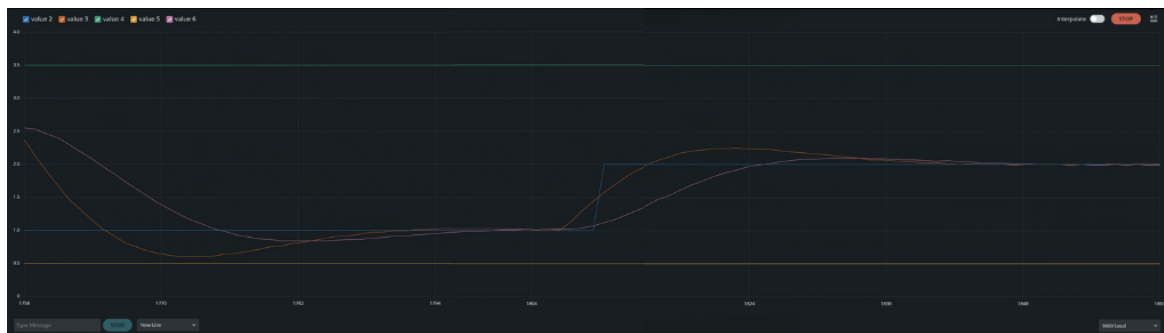


FIGURE III.6 – Système 2<sup>ème</sup> ordre : Visualisation des valeurs Pratique de Y, Yc et U dans le Temps

L'évaluation des résultats issus du modèle prédictif révèle une cohérence remarquable avec les consignes définies, démontrant ainsi la pertinence du modèle ARIMAX dans le contexte d'un système d'ordre 2. La précision et la fidélité du suivi des prédictions attestent de la robustesse et de l'efficacité de ce modèle, soulignant sa capacité à anticiper de manière précise et adaptative les dynamiques du système d'ordre 2. Ces observations positives renforcent la crédibilité du modèle ARIMAX dans la modélisation et la prédiction des systèmes.



# Conclusion

En résumé, ce projet a été dédié à la création et à la mise en œuvre d'un contrôleur prédictif novateur, s'appuyant sur le modèle ARIMAX pour la commande proactive de systèmes dynamiques. À travers des efforts consciencieux, nous avons synthétisé un régulateur prédictif généralisé, puis procédé à son implémentation sur Matlab pour simuler son efficacité dans le contrôle d'un système du 1er et 2ème ordre.

Les simulations sur Matlab ont confirmé la validité et la robustesse de notre régulateur prédictif. Forts de ces résultats, nous avons franchi le pas de l'application pratique en implantant le régulateur sur une carte Arduino. Cette transition vers un système réel a été couronnée de succès, démontrant la capacité de la commande prédictive basée sur le modèle ARIMAX à diriger un système vers la référence échelon souhaitée.

Cette expérience enrichissante souligne le potentiel significatif de la commande prédictive dans la résolution de problèmes complexes et dans l'amélioration des performances des systèmes en temps réel. Notre démarche, de la conception à la mise en œuvre sur une plateforme physique, offre une contribution tangible à l'avancement de la commande prédictive, ouvrant ainsi la voie à des applications futures dans divers domaines de l'ingénierie et de l'automatisation.

En conclusion, la réussite de ce projet confirme que la commande prédictive basée sur le modèle ARIMAX représente une approche prometteuse pour optimiser la régulation de systèmes dynamiques, marquant ainsi une étape significative vers des solutions plus intelligentes et réactives dans le domaine du contrôle automatique.

# Bibliographie

- [1] TroyDanielFZ. Generalised-predictive-control-gpc. <https://github.com/TroyDanielFZ/Generalised-Predictive-Control-GPC/blob/master/GPC-C%2B%2B-Implementation/Readme.md>, 2019.
- [2] Lin-Ya Chiu, Dan Jeric Arcega Rustia, Chen-Yi Lu, and Ta-Te Lin. Modelling and forecasting of greenhouse whitefly incidence using time-series and arimax analysis. *IFAC-PapersOnLine*, 52(30) :196–201, 2019.
- [3] Ranjit Kumar Paul and Kanchan Sinha. Forecasting crop yield : a comparative assessment of arimax and narx model. pages 78–80, 2016.
- [4] Ammar Ramdani and Said Grouni. Dynamic matrix control and generalized predictive control, comparison study with imc-pid. *International Journal of Hydrogen Energy*, 42(28) :17561–17570, 2017.
- [5] Boutaina Laredj and Ahsene Encadreur Boubakir. *Commandes prédictives appliquées à un système hydraulique : Etude expérimentale*. PhD thesis, Université de Jijel, 2021.