

ECOLE NATIONALE POLYTECHNIQUE

IDENTIFICATION DES PROCESSUS

Mini Projet

MÉTHODE DES MOMENTS GÉNÉRALISÉS

NECHAT Ghiles

`ghiles.nechat@g.enp.edu.dz`

MEGHNOUDJ Houssem

`houssem.meghnoudj@g.enp.edu.dz`

Département d'Automatique

19 janvier 2019

Table des matières

1	Introduction	2
2	Théorie	2
2.1	Rappels et définitions	2
2.2	Principe	3
2.3	Calcul des paramètres	3
2.3.1	Forme polynomiale	3
2.3.2	Forme de STREJ avec retard	4
3	Réalisation	4
3.1	Script MATLAB	5
4	Interface graphique	12
4.1	Lecture Tab	13
4.2	Simulation Tab	14
4.3	Identification Tab	15
4.4	Code MATLAB	16
5	Exemples	34
5.1	Forme de STREJ avec retard	34
5.2	Forme polynomiale, avec variation de (a)	36
5.3	Système instable	40
6	Conclusion	41

Table des figures

1	Interface de l'App Designer en DesignView	12
2	Interface de l'App Designer en CodeView	13
3	Interface Graphique en <i>LECTURE</i>	14
4	Interface Graphique en <i>SIMULATION</i>	15
5	Interface Graphique en <i>IDENTIFICATION</i>	16
6	Forme de STREJ <i>LECTURE</i>	34
7	Forme de STREJ <i>SIMULATION</i> sans pondération	35
8	Forme de STREJ <i>SIMULATION</i> avec pondération	35
9	Forme de STREJ <i>IDENTIFICATION</i>	36
10	Forme polynomiale <i>LECTURE</i>	36
11	Forme polynomiale <i>SIMULATION</i>	37
12	Forme polynomiale <i>IDENTIFICATION</i> pour $a = 0.7$	37
13	Forme polynomiale, $a = 0.1$	38
14	Forme polynomiale, $a = 0.2$	38
15	Forme polynomiale, $a = 0.3$	39
16	Système instable <i>LECTURE</i>	40
17	Système instable <i>SIMULATION</i>	40
18	Système instable <i>IDENTIFICATION</i>	41

1 Introduction

L'identification par la méthode des moments nécessite comme toute identification une durée d'observation supérieur au temps de réponse du système ainsi qu'à d'autres règles sur le signal d'entrée, d'autres contraintes spécifique à la méthode des moments s'ajoutent, une entrée impulsionnelle ainsi que la stabilité du système est nécessaire.

Sachant qu'une entrée impulsion $\delta(t)$ n'est pas réalisable physiquement même si on peut l'approximer par un rectangle petit et limité dans le temps ça reste insuffisant, ajouté à cela certains systèmes ne supporte pas une entrée brusque.

La méthode des moments généralisés viens compléter la méthode des moments du fait qu'elle ne présente pas ces inconvénients.

2 Théorie

2.1 Rappels et définitions

Pour la méthode des moments généralisés, on pondère la fonction à l'intégrer avec une fonction de pondération :

Moments temporels d'une fonction

$$M_k(p) = \int_0^{+\infty} t^k f(t) p(t) dt$$

$p(t)$: fonction de pondération de $f(t)$

La fonction de pondération permet de choisir/privilegier une portion de la fonction $f(t)$ sur l'intervalle $[0, +\infty[$, pour des raisons de convergence de l'intégrale et de simplification lors de la dérivée de la fonction de transfert on choisi $p(t) = e^{-at}$ où $a > 0$

$$M_k(p) = \int_0^{+\infty} t^k f(t) e^{-at} dt \quad (1)$$

Cette intégrale sera évalué en utilisant l'intégration numérique par la méthode des Simpson déjà élaboré en TP.

Relation entre les moments et $\mathcal{L}[f(t)]$

$$M_k(a) = (-1)^k \frac{d^k F(s)}{ds^k} \Big|_{s=a} \quad (2)$$

Comme on le remarque avec la fonction $p(t) = e^{-at}$ on retrouve facilement la relation entre le moment d'ordre k et la dérivé d'ordre k de $F(s)$, le seul changement qu'on constate c'est que la dérivé est maintenant évaluée quand $S = a$ au lieu de $S = 0$

Moments impulsionnels

$$M_k^H(a) = \frac{1}{M_0^E(a)} [M_k^S(a) - \sum_{i=0}^{k-1} C_k^i M_i^H(a) M_{k-i}^E] \quad (3)$$

Avec : $M_0^H(a) = \frac{M_0^S}{M_0^E}$

$M_k^H(a)$: Moment impulsif d'ordre k

$M_k^E(a)$: Moment d'entrée d'ordre k

$M_k^S(a)$: Moment de sortie d'ordre k

2.2 Principe

La méthode s'applique pour n'importe quel type d'entrée et pour la mise en œuvre de cette dernière il faut impérativement suivre les étapes suivantes :

1. Choix du modèle c'est la partie qualitative.
2. Choix du paramètre "a" qui se fait par tâtonnement.
3. Calcul numérique des moments de l'entrée et de la sortie M^E et M^S pondérés.
4. Calcul des moments de la réponse impulsif M^H
5. Calcul des paramètres c'est la partie quantitative ayant les relations entre les moments et les paramètres.
6. Validation suivant un critère, pour notre cas on a pris l'erreur quadratique.

2.3 Calcul des paramètres

2.3.1 Forme polynomiale

La fonction de transfert est de la forme

$$H(s) = \frac{b_0 + b_1 s + \dots + b_{n-1} s^{n-1}}{a_0 + a_1 s + \dots + a_{n-1} s^{n-1} + a_n s^n} \quad (4)$$

Après dérivations et calculs on trouve la relation linéaire suivante :

$$B^{(p)}(a) = \sum_{i=0}^p (-1)^i C_p^{p-i} M_i^H A^{(p-i)}(a) \quad (5)$$

Avant de pouvoir résoudre le système d'équation on a écrit la relation pour $n = 1$ et $n = 2$ pour généraliser la forme pour les autres cas.

Pour la résolution de ce système d'équations linéaire on a généré en code la matrice reliant les inconnu a_i et b_j aux données ($Dx = c$) pour le résoudre par la suite en utilisant la fonction inverse pour avoir $x = inv(D)C$ où x est un vecteur qui contient les éléments a_i et b_j .

Exemple de la matrice pour le cas où $n = 2$:

$$X = \begin{bmatrix} b_0 \\ b_1 \\ a_0 \\ a_1 \end{bmatrix}; \quad b(a) = \begin{bmatrix} a^2 C_0^0 M_0^H \\ -a^2 C_1^0 M_1^H + 2a C_2^1 M_0^H \\ a^2 C_2^0 M_2^H - 2a C_2^1 M_1^H + C_2^2 M_0^H \\ -a^2 C_3^0 M_3^H + 2a C_3^2 M_2^H - 2C_3^3 M_1^H \end{bmatrix}$$

$$D(a) = \begin{bmatrix} 1 & a & -C_0^0 M_0^H & -a C_0^0 M_0^H \\ 0 & 1 & C_1^0 M_1^H & a C_1^0 M_1^H - C_1^1 M_0^H \\ 0 & 0 & -C_2^0 M_2^H & -a C_2^0 M_2^H + C_2^1 M_1^H \\ 0 & 0 & C_3^0 M_3^H & a C_3^0 M_3^H - C_2^1 M_2^H \end{bmatrix}$$

2.3.2 Forme de STREJ avec retard

La fonction de transfert est de la forme :

$$H(s) = \frac{K e^{-\tau s}}{(1 + sT)^n} \quad (6)$$

Pour cette forme on a rencontré différents problèmes car les relations disponibles sur le document sont faussent donc on a recalculé de nouveau les paramètres en utilisant la formule (2) pour le modèle de STREJ. Les paramètres trouvés :

$$T_0 = \frac{2m_1^3 - 3m_2m_1 + m_3}{-2m_1^2 + 2m_2} \quad T = \frac{T_0}{1 - aT_0}$$

$$\tau = \frac{m_1^2m_2 + m_3m_1 - 2m_2^2}{2m_1^3 - 3m_2m_1 + m_3} \quad n = 4 \frac{(-m_1^2 + m_2)^3}{(2m_1^3 - 3m_2m_1 + m_3)^2}$$

$$K = m_0 \frac{(1 + aT)^n}{e^{-a\tau}}$$

3 Réalisation

Pour la réalisation le code est divisé en 3 parties :

1. Le fichier fonctions.m qui comporte :

- La fonction *CalcMoment* pour calculer les moments de l'ordre 1 jusqu'à n, cette fonction prends comme paramètre :
 - n : L'ordre jusqu'où on arrête le calcul des moments.
 - t : Vecteur temps.
 - y : Contient la fonction f évalué pour tous les points de t.
 - M : Vecteur contenant tous les moments calculés.
- La fonction *IntSimpson* intégration de simpson.
- La fonction *CalcMomentGen* qui prends comme entrée les moments du signal d'entrée et celui de la sortie en utilisant la formule (3) :
 - M^E : Les moments du signal d'entrée.
 - M^S : Les moments du signal de sortie.

- La fonction *CalcParamPolynome* qui crée la matrice D et b présente dans la section (2.3.1) en ayant les moments impulsionnels et la valeur de 'a'.
 - La fonction *TransferFunction_text* qui est nécessaire pour l'application.
2. Un fichier qui n'est pas visible qui a permis de créer l'application (interface graphique).
 3. Un fichier principal contenant la déclaration du signal d'entrée, du système, du vecteur temps, des affichages des fonctions, le calcul des moments, calcul des paramètres ...

3.1 Script MATLAB

```

1 %Contient toutes les fonctions utilisees
2 classdef fonctions
3     methods(Static)
4         function [M] = CalcMoment(n,t,y)
5             %Fonction pour le calcul des moments
6             % n : Le nombre de moments que vous voulez calculer
7             % t : Vecteur temps (uni colonne)
8             % y : Vecteur contenant la fonction f(t) (uni colonne)
9             % M : Vecteur contenant les moments calcules
10            if nargin~=3
11                error('Introduisez trois parametres.');
12            else
13                M=[];
14                for i=0:n-1
15                    M=[M fonctions.IntSimpson(t'.^i.*y,t(2))];
16                end
17            end
18        end
19
20        function [Sy] = IntSimpson(Y,DX)
21            % Fonction qui retourne l'integrale de Simpson,
22            % Y : Les Abcisses & DX : Le pas d'integration.
23            if not(mod(length(Y),2))
24                error('Le Nombre de points doit etre impaire.');
25            end
26            Sy = DX*( sum(Y(1:end-1)) + sum(Y(2:end)) + 2*sum(Y(2:2:end-1)) )/3
27                ;
28        end
29
30        function [M] = CalcMomentGen(Me,Ms)
31            % Fonction qui calcule les moments impulstionnels
32            % A partir des moments de l'entree et de la sortie
33            % Me : Moments de l'entree
34            % Ms : Moments de la sortie
35            n=length(Me);
36            M=Ms(1)/Me(1);

```

```

36         for k=1:n-1
37             S=0;
38             for i=0:k-1
39                 S=S+nchoosek(k,i)*M(i+1)*Me(k-i+1);
40             end
41             M=[M (Ms(k+1)-S)/Me(1)];
42         end
43     end
44
45     function [S] = CalcParamPolynome(M,n,a_val)
46         syms a
47         V=[];
48         G=zeros(2*n,n+1);
49         for k=0:n
50             V=[V a^k];
51         end
52         for k=1:n
53             V=[V;diff(V(end,:),a)];
54         end
55         V=subs(V,a,a_val);
56         for i=1:2*n
57             for k=1:n+1
58                 if i+k-1<2*n+1
59                     G(i+k-1,:)=G(k+i-1,:)+nchoosek(i+k-2,k-1)*(-1)^(i-1)*M(i)*V(k
60                         ,:);
61                 end
62             end
63         end
64         G=double([V(:,1:end-1) G]);
65         A=[G(:,1:n) -1.*G(:,n+1:end-1)];
66         b=G(:,end);
67         S=A\b;
68     end
69
70     function [output_string]=TransferFunction_text(TF)
71         clc
72         TF
73         [cmdWin]=com.mathworks.mde.cmdwin.CmdWin.getInstance;
74         cmdWin_comps=get(cmdWin,'Components');
75         subcomps=get(cmdWin_comps(1),'Components');
76         text_container=get(subcomps(1),'Components');
77         output_string=get(text_container(1),'text');
78         F=strfind(output_string,[output_string(1),' ']);
79         output_string=output_string(F(2)+1:F(end)-1);
80     end
81 end

```

Listing 1 – fonctions.m

```

1 clear
2 clc
3 close
4 disp(' Identification des Processus : La Methode des Moments generalises')
5 disp(' Auteurs : NECHAT Ghiles & MEGHNOUDJ Houssem ');
6 disp('          ENP, Automatique 2eme annee. '), pause;
7
8 %initialisation
9     t_fin=10;
10    t=linspace(0,t_fin,1001);
11    a=5;
12    s=tf('s');
13
14 %systeme :
15    %tu peux mettre un choix pour des exemple deja present comme exemple
16    %G=2*exp(-2*s)/(1+2*s)^4;
17    G=(1+2*s)/(5+2*s+s^2);
18    %G=4*exp(-0.2*s)/(1+0.4*s)^3;
19    %G=3/(4+1*s);
20    %G=zpk([],[-1/4 -1/4 -1/4],4/64,'InputDelay',2);
21    %G=input('Introduisez votre fonction de transfert :\n G = ');
22
23 %Entree :
24    for i=1:length(t)
25        u(i,1)=min(exp(t(i)-5),exp(-t(i)+5))-exp(-5);
26    end
27    %u=[ones(100,1);zeros(1001-100,1)];
28    %u=[zeros(1,1);ones(1000,1)];
29    fprintf('\nChoisissez l'un des modeles suivants :\n');
30    fprintf('\t FP Forme Polynomiale\n')
31    fprintf('\t SR STREJC Avec Retard\n')
32    choix=input('\nVotre choix : ','s');
33
34
35 %affichage premier
36 subplot(2,1,1)
37 grid on
38 box off
39 hold on
40 title('Entrée du système à identifier','interpret','latex','FontSize', 13)
41 plot(t,u,'linewidth',1.1)
42 plot(t,exp(-a*t),'r—','linewidth',.01)
43 ylabel('Input','interprete','latex','FontSize', 12)
44 U=exp(-a*t).*u;
45 hLeg=legend({'$u_e=f(t)$','$e^{-a.t}$'},'Location','best','FontSize', 12);
46 set(hLeg,'Interpreter','latex');

```



```

47
48 y=lsim(G,u,t);
49 subplot(2,1,2)
50 grid on
51 box off
52 hold on
53 title('Sortie du syst{\`e}me {\`a} identifier','interpret','latex','FontSize',
    13)
54 plot(t,y,'linewidth',1.1)
55 plot(t,exp(-a*t),'r—','linewidth',.01)
56 xlabel('Temps (s)','interprete','latex','FontSize', 12)
57 ylabel('Output','interprete','latex','FontSize', 12)
58 Y=exp(-a*t).*y;
59 hLeg=legend({'$$y_s=g(t)$$','$e^{-a.t}$$'],'Location','best','FontSize', 12);
60 set(hLeg,'Interpreter','latex');
61 pause
62
63 clf
64 %affichage second
65 subplot(2,1,1)
66 grid on
67 box off
68 hold on
69 title({'Entr{\`e}e du syst{\`e}me {\`a} identifier','apr{\`e}s ajout des
    poids'],'interpret','latex','FontSize', 13)
70 plot(t,U,'linewidth',1)
71 ylabel('Input','interprete','latex','FontSize', 12)
72 hLeg=legend({'$$u_e=f(t).e^{-a.t}$$'],'Location','best','FontSize', 12);
73 set(hLeg,'Interpreter','latex');
74
75 y=lsim(G,u,t);
76 subplot(2,1,2)
77 grid on
78 box off
79 hold on
80 title({'Sortie du syst{\`e}me {\`a} identifier','apr{\`e}s ajout des poids'],'
    'interpret','latex','FontSize', 13)
81 xlabel('Temps (s)','interprete','latex','FontSize', 12)
82 ylabel('Output','interprete','latex','FontSize', 12)
83 plot(t,Y,'linewidth',1)
84 hLeg=legend({'$$y_s=g(t).e^{-a.t}$$'],'Location','best','FontSize', 12);
85 set(hLeg,'Interpreter','latex');
86 pause
87
88 %cote systeme
89 switch choix
90
91     case 'FP'

```

```

92     n=order(G);
93     Me=fonctions.CalcMoment(2*n,t,U);
94     Ms=fonctions.CalcMoment(2*n,t,Y);
95     M =fonctions.CalcMomentGen(Me,Ms);
96     D=fonctions.CalcParamPolynome(M,n,a);
97     D=fliplr(D');
98     Gm=tf(D(n+1:end),[1 D(1:n)]);
99     round_a=round(D(1:n),2);
100    round_b=round(D(n+1:end),2);
101    den=[];
102    num=['S^',num2str(n),' + '];
103    for h=0:n-1
104        if h==0
105            den=[den,num2str(round_b(1+h)),'S^',num2str(n-h-1)];
106            num=[num,num2str(round_a(1+h)),'S^',num2str(n-h-1)];
107        elseif h==n-1
108            den=[den,' + ',num2str(round_b(1+h))];
109            num=[num,' + ',num2str(round_a(1+h))];
110        else
111            den=[den,' + ',num2str(round_b(1+h)),'S^',num2str(n-h-1)];
112            num=[num,' + ',num2str(round_a(1+h)),'S^',num2str(n-h)];
113        end
114    end
115    Val=['$\frac{',den,'}','{',num,'}$'];
116
117
118    case 'SR'
119        n=4;
120        Me=fonctions.CalcMoment(n,t,U);           %Moment de l'entree
121        Ms=fonctions.CalcMoment(n,t,Y);           %Moment de la sortie
122        M =fonctions.CalcMomentGen(Me,Ms);         %Moment impulsionnel
123        m =[M(1) M(2:end)./M(1)];
124
125        %calcul des parametres
126        T_0=(2*m(2)^3 - 3*m(3)*m(2) + m(4))/(- 2*m(2)^2 + 2*m(3));
127        T=T_0/(1-a*T_0);
128        Tau=(m(2)^2*m(3) + m(4)*m(2) - 2*m(3)^2)/(2*m(2)^3 - 3*m(3)*m(2) + m(4)
129            );
130        n=round((4*(-m(2)^2 + m(3))^3)/(2*m(2)^3 - 3*m(3)*m(2) + m(4))^2);
131        K=m(1)*(1+a*T)^n/(exp(-a*Tau));
132        Gm=K*exp(-s*Tau)/(1+s*T)^n
133        Val=['$\frac{',num2str(round(K,2)),' e^{-',num2str(round(Tau,2)),' t
134            }{(1+s',num2str(round(T,2)),' )^',num2str(n),'}$'];
135
136    otherwise
137        choix=input('\nVeuillez introduire correctement votre choix : ');
138    end

```

```

138
139 % Partie validation et comparaison
140 clf
141 choix=1;
142 switch choix
143     case 1 %reponse impulsif
144         ym=impulse(Gm,t);
145         y=impulse(G,t);
146         J=fonctions.IntSimpson((y-ym).^2,t(2));
147         y_label=['J = ',num2str(J)];
148         title({'R{\`e}ponse Impulsionnelle ( Syst{\`e}me + Mod{\`e}le )';Val},'
            'interpret','latex','fontsize',14)
149         xlabel('Temps (s)','interprete','latex','fontsize',12)
150         ylabel({'Erreur Quadratique';y_label},'interprete','latex','fontsize',12)
151         hold on
152         plot(t,y)
153         plot(t,ym,'r—','linewidth',1.1)
154         hold off
155         grid on
156         box off
157         legend({'$$y(t)$$','$$y_m(t)$$'},'Interpreter','latex','Location','best','
            'fontsize',12);
158
159     case 2 %reponse indicielle
160         ym=step(Gm,t);
161         y=step(G,t);
162         J=fonctions.IntSimpson((y-ym).^2,t(2));
163         y_label=['J = ',num2str(J)];
164         title({'R{\`e}ponse indicielle ( Syst{\`e}me + Mod{\`e}le )',Val},'interpret',
            'latex','fontsize',14)
165         xlabel('Temps (s)','interprete','latex','fontsize',12)
166         ylabel({'Erreur Quadratique';y_label},'interprete','latex','fontsize',12)
167         hold on
168         plot(t,y)
169         plot(t,ym,'r—','linewidth',1.1)
170         hold off
171         grid on
172         box off
173         legend({'$$y(t)$$','$$y_m(t)$$'},'Interpreter','latex','Location','best',
            'fontsize',12);
174 end

```

Listing 2 – Projet.m

Remarques et explications

Pour la partie graphique il se peut qu'il y ait des données qui changent par exemple c'est à l'utilisateur d'introduire le signal d'entrée alors que sur le code on a pris un signal quelconque.

- La partie initialisation permet de déclarer le vecteur temps (temps final, nombre de points) ainsi que de déclarer la valeur de (a) .
- La partie système permet de choisir un système.
- La partie entrée permet d'introduire le signal d'entrée avec lequel on a excité le système, pour l'exemple on a pris un signal quelconque.
- Il y a un choix qui permet de choisir si notre modèle est un modèle polynomiale ou de STREJ sans retard.
- Un premier affichage qui affiche notre signal d'entrée et celui de la sortie, ainsi que le signal de pondération en pointillés, cette manière permet de voir quelle partie on va préférer avec notre pondération.
- Un deuxième affichage qui permet de voir les signaux d'entrée et sortie après pondération.
- Un switch qui se fera en fonction du modèle choisi, suivi d'une partie qui calcule les moments d'entrée et de sortie ainsi que les moments impulsionnels, par la suite il y a un calcul des paramètres et une préparation d'affichage des fonctions de transfert.
- La dernière partie est comparative, on a la possibilité de choisir la réponse indicielle ou impulsionnelle à fin de procéder à la validation, affichage de la fonction de transfert identifiée avec l'erreur quadratique.

Plus (a) est grand plus l'erreur quadratique sera petite, par contre il ne faut pas que (a) prenne une grande valeur, car il ne restera aucune information sur le système, de plus pour les systèmes à retard pur il se peut qu'aucune information ne sera observé à la sortie pondérée.

4 Interface graphique

Pour commencer, la partie 'Interface Graphique' de ce projet à été entièrement créée avec l'application 'App Designer' sous MATLAB R2018a.

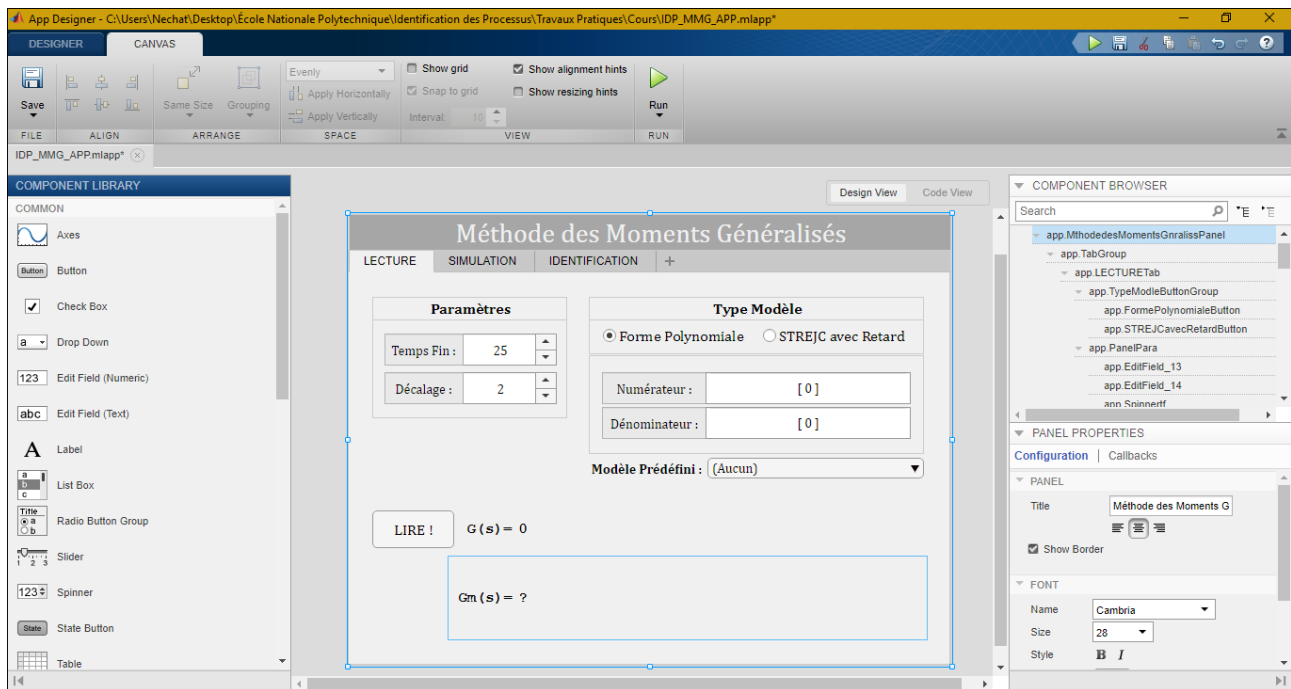


FIGURE 1 – Interface de l'App Designer en DesignView

Il faudra définir les constantes globales de l'application à travers une Property :

- app.TF : La fonction de transfert.
- app.TFt : Le type de la fonction de transfert.
- app.a : le décalage.
- app.t_fin : le temps finale.
- app.U : l'entrée.
- app.Y : la sortie.
- app.Mom : les moments.

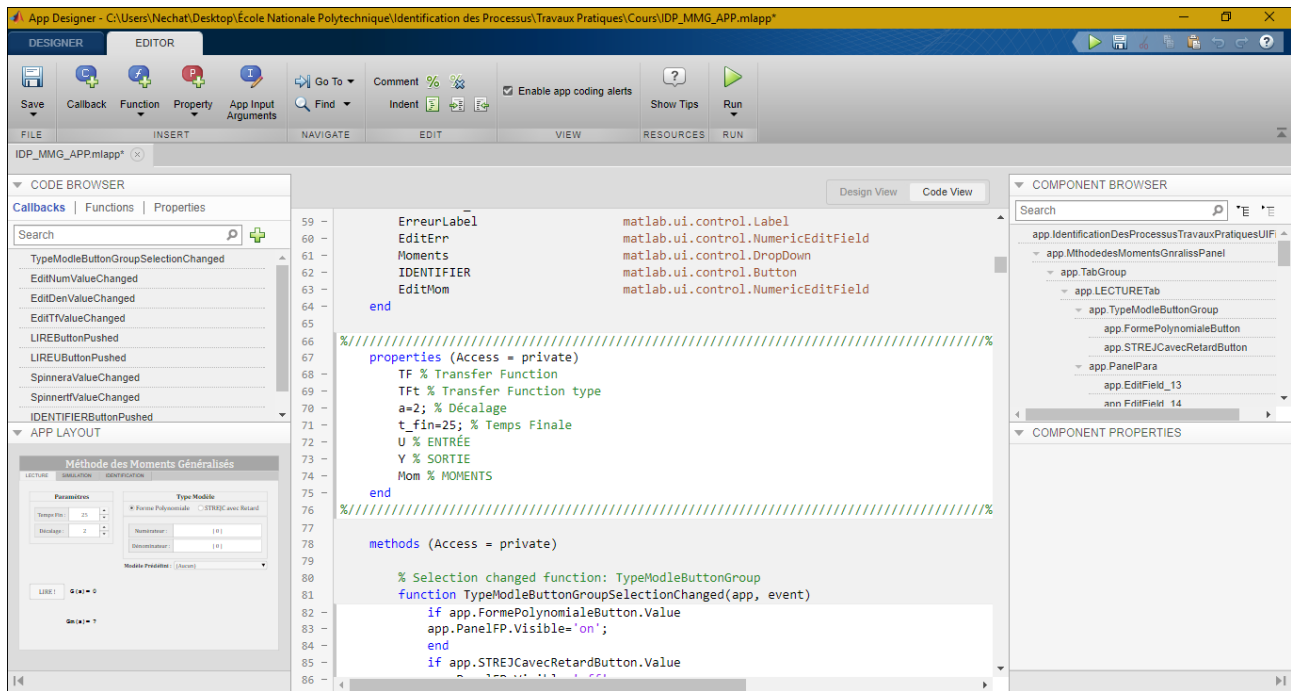


FIGURE 2 – Interface de l'App Designer en CodeView

L'interface graphique est composée comme suit :

1. *La figure principale* : dont le titre 'Identification Des Processus : Travaux Pratiques'
2. *Le Panel principale* : dont le titre 'Méthode des Moments Généralisés'.
3. *Un TabGroup principale* : qui comporte l'identification décomposée en trois parties qui vont être définies dans ce qui suit.

4.1 Lecture Tab

C'est là où se fera la supposée expérience, on y définit le système à identifier :

- **Le Panel *Type Modèle*** : on y choisit la forme du système à identifier avec deux Radio-Button :
 1. Le RadioButton 'Forme Polynomiale' : affiche deux EditField pour introduire le numérateur et le dénominateur sous forme de vecteurs.
 2. Le RadioButton 'STREJC avec Retard' : affiche quatre EditField pour introduire l'amplitude, l'ordre, la constante de temps et le retard pure.
- **Le Panel *Paramètres*** : on y choisit la durée de l'expérience t_{fin} et le décalage (a) à l'aide de deux Spinner à valeur positive.
- **Le DropDown *Modèle Prédéfini*** : on y choisit entre quatre types de modèles déjà définis en mémoire, sinon sera pris le modèle créé cidessus.
- **Le Label *G*** : où sera affiché la fonction de transfert du système $G(s)$ exactement comme le ferait le command window de l'environnement MATLAB.

- **Le Label G_m** : où sera affiché la fonction de transfert du modèle $G_m(s)$ exactement comme le ferait le command window de l'environnement MATLAB, juste après l'identification.
- **Le Button *LIRE !*** : cliquer ce bouton fera exécuter la partie du script nécessaire à la lecture des données système en fonction de ce qui à déjà été défini.

Identification Des Processus : Travaux Pratiques

Méthode des Moments Généralisés

LECTURE SIMULATION IDENTIFICATION

Paramètres

Temps Fin : 25

Décalage : 2

Type Modèle

☐ Forme Polynomiale ☒ STREJC sans Retard

Amplitude : 5 Ordre : 3

Temps : 2 Retard : 0.4

Modèle Prédéfini : Système d'Ordre 2

LIRE !

$$G(s) = \frac{2s + 1}{s^2 + s + 5}$$

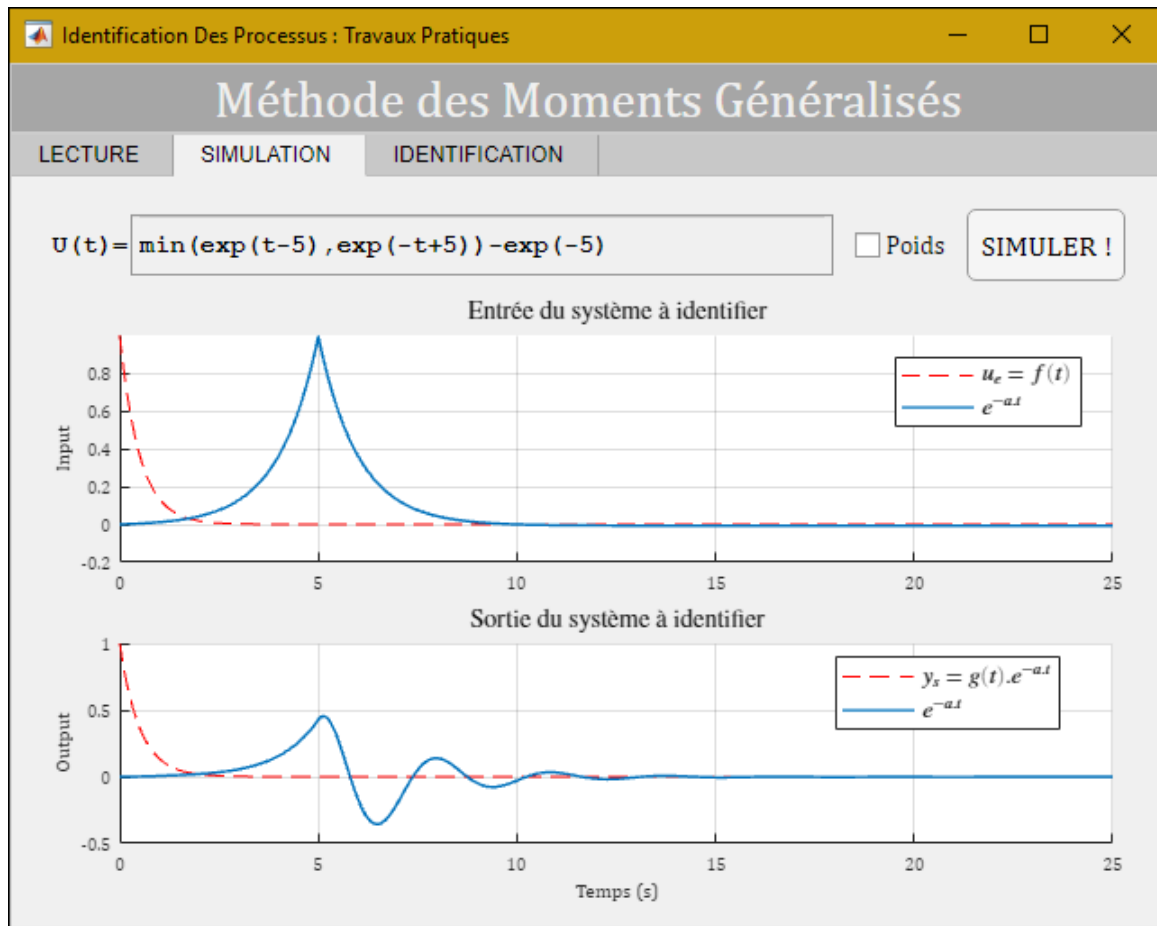
$$G_m(s) = \frac{2.001s + 1}{s^2 + 1.001s + 5.001}$$

FIGURE 3 – Interface Graphique en *LECTURE*

4.2 Simulation Tab

C'est la où se fera la simulation de la réponse du système à une entrée choisie :

- **Le EditField $U(t)$** : qui accepte une valeur texte, étant la fonction $u(t)$ écrite sous forme de commande MATLAB.
- **La CheckBox *Poids*** : pour choisir entre l'algorithme avec ou sans les poids.
- **Les deux Axes $U1$ et $U2$** : pour l'affichage respectif de l'Input et de l'Output.
- **Le Button *SIMULER !*** : cliquer ce bouton fera exécuter la partie du script nécessaire à la simulation de la réponse du système $G(s)$ à l'entrée $U(t)$, ainsi que l'affichage des résultats.

FIGURE 4 – Interface Graphique en *SIMULATION*

4.3 Identification Tab

C'est là où se fera le calcul des moments, le calcul des paramètres, l'affichage comparatif des réponses et le calcul de l'erreur quadratique :

- **Le Panel *Type Réponse*** : on y choisit le type de réponse, ou 'Impulsionnelle' ou 'Indicielle', à laquelle le modèle et le système devront répondre. Muni de deux `RadioButton` :
 1. Le `RadioButton` 'Impulsionnelle' : on fera subir un `impulse()` en entrée.
 2. Le `RadioButton` 'Indicielle' : on fera subir un `step()` en entrée.
- **Le Axes *U*** : pour l'affichage comparatif des deux réponses en même temps.
- **Le `EditField` *Moments*** : Editable étant désactivée, ce champs servira à afficher les moments généraux calculés suivant un `DropDown`.
- **Le `DropDown` *Moments*** : avec lequel se fera le choix de l'ordre du moment à afficher.
- **Le `EditField` *Erreur*** : Editable étant désactivée, ce champs servira à afficher l'erreur quadratique entre les deux réponses.

- Le Button **IDENTIFIER** ! : cliquer ce bouton fera exécuter la partie du script nécessaire au calcul des moments généraux, au calcul des paramètres (FP ou SR selon le 'Type Modèle'), à la déclaration de la fonction de transfert du modèle (l'affichage de ce modèle sur l'onglet 'LECTURE'), à l'affichage des moments généraux et de l'erreur et enfin à tracer les résultats.

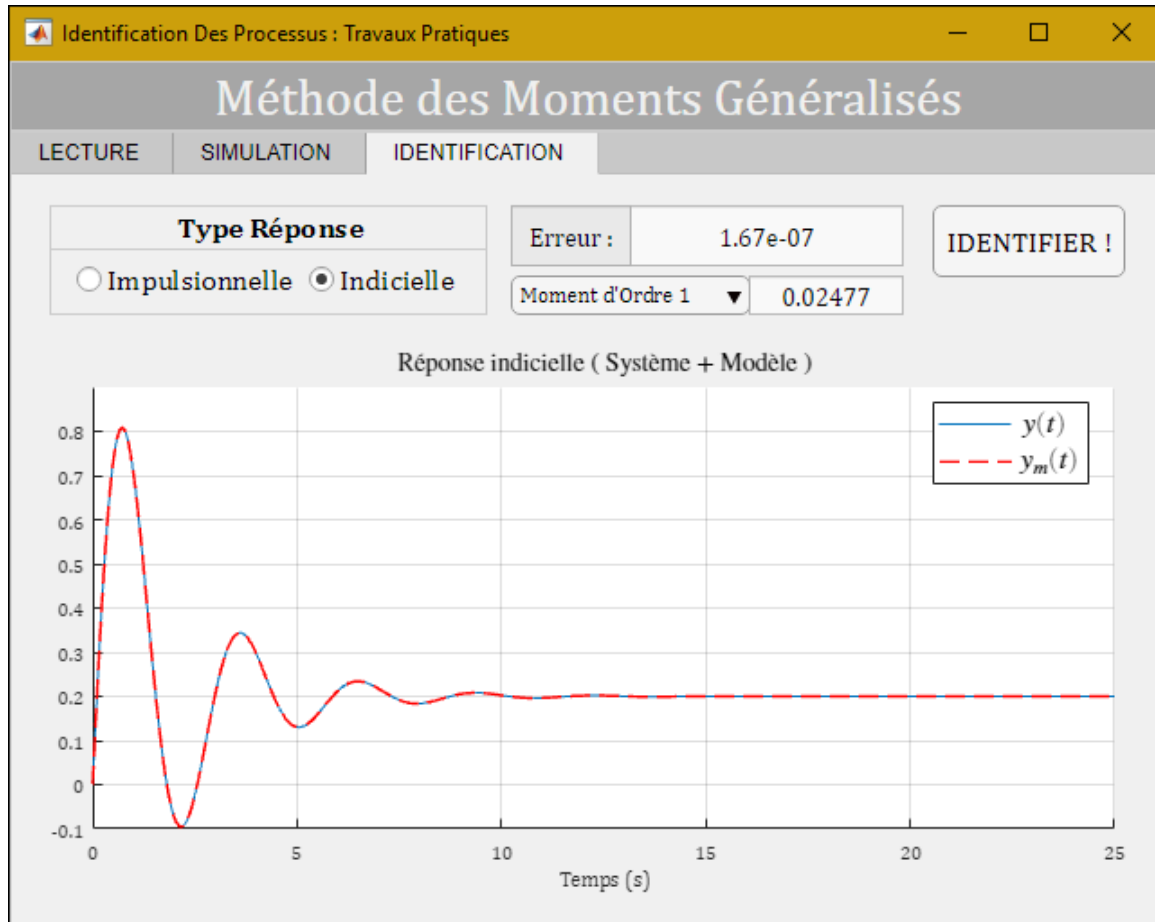


FIGURE 5 – Interface Graphique en *IDENTIFICATION*

4.4 Code MATLAB

```

1 classdef IDP_MMG_APP < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         IdentificationDesProcessusTravauxPratiquesUIFigure matlab.ui.Figure
6         MthodedesMomentsGnralissPanel matlab.ui.container.Panel
7         TabGroup matlab.ui.container.TabGroup
8         LECTURETab matlab.ui.container.Tab
9         TypeModleButtonGroup matlab.ui.container.ButtonGroup
10        FormePolynomialeButton matlab.ui.control.RadioButton
11        STREJCavecRetardButton matlab.ui.control.RadioButton

```

12	PanelPara	matlab.ui.container.Panel
13	EditField_13	matlab.ui.control.EditField
14	TempsFinLabel	matlab.ui.control.Label
15	EditField_14	matlab.ui.control.EditField
16	DcalageLabel	matlab.ui.control.Label
17	Spinnertf	matlab.ui.control.Spinner
18	Spinnera	matlab.ui.control.Spinner
19	PanelSR	matlab.ui.container.Panel
20	EditAmp	matlab.ui.control.NumericEditField
21	EditTem	matlab.ui.control.NumericEditField
22	EditOrd	matlab.ui.control.NumericEditField
23	EditField_7	matlab.ui.control.EditField
24	AmplitudeLabel	matlab.ui.control.Label
25	EditField_8	matlab.ui.control.EditField
26	OrdreLabel	matlab.ui.control.Label
27	EditField_9	matlab.ui.control.EditField
28	TempsLabel	matlab.ui.control.Label
29	EditField_10	matlab.ui.control.EditField
30	RetardLabel	matlab.ui.control.Label
31	EditRet	matlab.ui.control.NumericEditField
32	PanelFP	matlab.ui.container.Panel
33	EditField_15	matlab.ui.control.EditField
34	NumrateurLabel	matlab.ui.control.Label
35	EditField_16	matlab.ui.control.EditField
36	DnominateurLabel	matlab.ui.control.Label
37	EditNum	matlab.ui.control.EditField
38	EditDen	matlab.ui.control.EditField
39	ModlePrdfiniDropDownLabel	matlab.ui.control.Label
40	ModlePrdfiniDropDown	matlab.ui.control.DropDown
41	LIREButton	matlab.ui.control.Button
42	GsLabel	matlab.ui.control.Label
43	LabelG	matlab.ui.control.Label
44	GmsLabel	matlab.ui.control.Label
45	LabelGM	matlab.ui.control.Label
46	SIMULATIONTab	matlab.ui.container.Tab
47	EditU	matlab.ui.control.EditField
48	LIREU	matlab.ui.control.Button
49	UtLabel	matlab.ui.control.Label
50	AxesU1	matlab.ui.control.UIAxes
51	AxesU2	matlab.ui.control.UIAxes
52	Poids	matlab.ui.control.CheckBox
53	IDENTIFICATIONTab	matlab.ui.container.Tab
54	TypeRponseButtonGroup	matlab.ui.container.ButtonGroup
55	ImpulsionnelleButton	matlab.ui.control.RadioButton
56	IndicielleButton	matlab.ui.control.RadioButton
57	AxesU	matlab.ui.control.UIAxes
58	EditField_18	matlab.ui.control.EditField
59	ErreurLabel	matlab.ui.control.Label

```

60         EditErr          matlab.ui.control.NumericEditField
61         Moments            matlab.ui.control.DropDown
62         IDENTIFIER        matlab.ui.control.Button
63         EditMom           matlab.ui.control.NumericEditField
64     end
65
66     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
67     properties (Access = private)
68         TF % Transfer Function
69         TFt % Transfer Function type
70         a=2; % Decalage
71         t_fin=25; % Temps Finale
72         U % ENTReE
73         Y % SORTIE
74         Mom % MOMENTS
75     end
76     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
77
78     methods (Access = private)
79
80         % Selection changed function: TypeModleButtonGroup
81         function TypeModleButtonGroupSelectionChanged(app, event)
82             if app.FormePolynomialeButton.Value
83                 app.PanelFP.Visible='on';
84             end
85             if app.STREJCavecRetardButton.Value
86                 app.PanelFP.Visible='off';
87             end
88         end
89
90         % Callback function
91         function EditNumValueChanged(app, event)
92             % JE N'ARRIVE PAS A ME DEBARRASER DE CE CALLBACK %
93         end
94
95         % Callback function
96         function EditDenValueChanged(app, event)
97             % JE N'ARRIVE PAS A ME DEBARRASER DE CE CALLBACK %
98         end
99
100        % Callback function
101        function EditTfValueChanged(app, event)
102            % JE N'ARRIVE PAS A ME DEBARRASER DE CE CALLBACK %
103        end
104
105        % Button pushed function: LIReButton
106        function LIReButtonPushed(app, event)
107            if strcmp(app.ModlePrdfiniDropDown.Value, '(Aucun)')

```

```

108         if app.FormePolynomialeButton.Value
109             Num=str2num(app.EditNum.Value);
110             Den=str2num(app.EditDen.Value);
111             if any(Den~=0)
112                 app.TF=tf(Num,Den);
113                 app.TFt=1;
114                 app.LabelG.Text=fonctions.TransferFunction_text(app.TF)
115                 ;
116             end
117         end
118         if app.STREJCavecRetardButton.Value
119             A=app.EditAmp.Value;
120             n=app.EditOrd.Value;
121             T=app.EditTem.Value;
122             Tau=app.EditRet.Value;
123             if T~=0 || n==0
124                 app.TF=zpk([],repmat(-1/T,[1 n]),A/T^n,'DisplayFormat',
125                     'Time Constant','InputDelay',Tau);
126                 app.TFt=2;
127                 app.LabelG.Text=fonctions.TransferFunction_text(app.TF)
128                 ;
129             end
130         end
131         if strcmp(app.ModlePrdfiniDropDown.Value,'Système d''Ordre 1')
132             app.TF=tf(1,[2 3]);
133             app.TFt=1;
134             app.LabelG.Text=fonctions.TransferFunction_text(app.TF);
135         end
136         if strcmp(app.ModlePrdfiniDropDown.Value,'Système d''Ordre 2')
137             app.TF=tf([2 1],[1 1 5]);
138             app.TFt=1;
139             app.LabelG.Text=fonctions.TransferFunction_text(app.TF);
140         end
141         if strcmp(app.ModlePrdfiniDropDown.Value,'STREJC d''Ordre 3')
142             app.TF=zpk([],repmat(-1/(0.4),[1 3]),4/(0.4)^3,'DisplayFormat',
143                 'Time Constant','InputDelay',0.2);
144             app.TFt=2;
145             app.LabelG.Text=fonctions.TransferFunction_text(app.TF);
146         end
147         if strcmp(app.ModlePrdfiniDropDown.Value,'STREJC d''Ordre 4')
148             app.TF=zpk([],repmat(-1/2,[1 4]),2/2^4,'DisplayFormat','Time
149                 Constant','InputDelay',2);
150             app.TFt=2;
151             app.LabelG.Text=fonctions.TransferFunction_text(app.TF);
152         end
153         app.LabelGM.Text=' ?';
154     end

```

```

151
152 % Button pushed function: LIREU
153 function LIREUButtonPushed(app, event)
154     t=linspace(0,app.t_fin,1001);
155
156     eval(['u(:,1)=' ,app.EditU.Value, ';' ]);
157     app.U=exp(-app.a*t').*u;
158     y=lsim(app.TF,u,t);
159     app.Y=exp(-app.a*t').*y;
160
161     if not(app.Poids.Value)
162         cla(app.AxesU1)
163         cla(app.AxesU2)
164         title(app.AxesU1,'Entr{\''e}e du syst{\`e}me {\`a} identifier',
165             'interpret','latex','FontSize',14);
166         plot(app.AxesU1,t,exp(-app.a*t'), '--r', 'linewidth',.01)
167         hold(app.AxesU1);
168         plot(app.AxesU1,t,u, 'linewidth',1.1)
169         legend(app.AxesU1,{'$u_e=f(t)$','$e^{-a.t}$'}, 'Location', '
170             best', 'Interpreter', 'latex', 'FontSize',10);
171
172         title(app.AxesU2,'Sortie du syst{\`e}me {\`a} identifier', '
173             interpret','latex', 'FontSize',14)
174         plot(app.AxesU2,t,exp(-app.a*t'), '--r', 'linewidth',.01)
175         hold(app.AxesU2);
176         plot(app.AxesU2,t,y, 'linewidth',1.1)
177         legend(app.AxesU2,{'$y_s=g(t).e^{-a.t}$','$e^{-a.t}$'}, '
178             Location', 'best', 'Interpreter', 'latex', 'FontSize',10);
179
180         hold(app.AxesU1);
181         hold(app.AxesU2);
182         clc
183
184     else
185         cla(app.AxesU1)
186         cla(app.AxesU2)
187         title(app.AxesU1,'Entr{\''e}e du syst{\`e}me {\`a} identifier
188             apr{\''e}s ajout des poids', 'interpret','latex', 'FontSize'
189             ,14)
190         plot(app.AxesU1,t,app.U, 'linewidth',1)
191         legend(app.AxesU1,{'$u_e=f(t).e^{-a.t}$'}, 'Location', 'best', '
192             Interpreter', 'latex', 'FontSize',10);
193
194         title(app.AxesU2,'Sortie du syst{\`e}me {\`a} identifier apr{\`
195             e}s ajout des poids', 'interpret','latex', 'FontSize',14)
196         plot(app.AxesU2,t,app.Y, 'linewidth',1)
197         legend(app.AxesU2,{'$y_s=g(t).e^{-a.t}$'}, 'Location', 'best', '
198             Interpreter', 'latex', 'FontSize',10);

```

```

190
191     end
192 end
193
194 % Value changed function: Spinnera
195 function SpinneraValueChanged(app, event)
196     app.a = app.Spinnera.Value;
197 end
198
199 % Value changed function: Spinnertf
200 function SpinnertfValueChanged(app, event)
201     app.t_fin = app.Spinnertf.Value;
202 end
203
204 % Button pushed function: IDENTIFIER
205 function IDENTIFIERButtonPushed(app, event)
206     cla(app.AxesU)
207     t=linspace(0,app.t_fin,1001);
208     if app.TFt==1
209         N=order(app.TF);
210         Me=fonctions.CalcMoment(2*N,t,app.U);
211         Ms=fonctions.CalcMoment(2*N,t,app.Y);
212         M =fonctions.CalcMomentGen(Me,Ms);
213         app.Mom=M;
214         D=fonctions.CalcParamPolynome(M,N,app.a);
215         D=flip1r(D');
216         Gm=tf(D(N+1:end),[1 D(1:N)]);
217         app.LabelGM.Text=fonctions.TransferFunction_text(Gm);
218     end
219     if app.TFt==2
220         N=4;
221         Me=fonctions.CalcMoment(N,t,app.U);           %Moment de l'entree
222         Ms=fonctions.CalcMoment(N,t,app.Y);           %Moment de la sortie
223         M =fonctions.CalcMomentGen(Me,Ms);           %Moment impulsionnel
224         app.Mom=M;
225         m =[M(1) M(2:end)]./M(1)];
226
227         %calcul des parametres
228         T_0=(2*m(2)^3 - 3*m(3)*m(2) + m(4))/(- 2*m(2)^2 + 2*m(3));
229         T=T_0/(1-app.a*T_0);
230         Tau=(m(2)^2*m(3) + m(4)*m(2) - 2*m(3)^2)/(2*m(2)^3 - 3*m(3)*m
231             (2) + m(4));
232         n=round((4*(-m(2)^2 + m(3))^3)/(2*m(2)^3 - 3*m(3)*m(2) + m(4))
233             ^2);
234         K=m(1)*(1+app.a*T)^n/(exp(-app.a*Tau));
235         Gm=zpk([],repmat(-1/T,[1 n]),K/T^n,'DisplayFormat','Time
236             Constant','InputDelay',Tau);
237         app.LabelGM.Text=fonctions.TransferFunction_text(Gm);

```

```

235         end
236         N=length(M);
237         C=cell(1,N);
238         for i=1:N
239             C{i}=[ 'Moment d\'Ordre ',num2str(i-1)];
240         end
241         app.Moments.Items=C;
242         app.Moments.Value=C{1};
243         app.EditMom.Value=M(1);
244
245         if app.ImpulsionnelleButton.Value
246             ym=impulse(Gm,t);
247             y=impulse(app.TF,t);
248             title(app.AxesU,'R{\`e}ponse Impulsionnelle ( Syst{\`e}me +
                Mod{\`e}le )','interpret','latex','FontSize',14)
249             hold(app.AxesU)
250             plot(app.AxesU,t,y)
251             plot(app.AxesU,t,ym,'r—','linewidth',1.1)
252             hold(app.AxesU)
253             legend(app.AxesU,{'$y(t)$','$y_m(t)$'},'Interpreter','latex',
                ', 'Location','best','FontSize',12);
254         end
255         if app.IndicielleButton.Value
256             ym=step(Gm,t);
257             y=step(app.TF,t);
258             title(app.AxesU,'R{\`e}ponse indicielle ( Syst{\`e}me + Mod{\`e}le )',
                'interpret','latex','FontSize',14)
259             hold(app.AxesU)
260             plot(app.AxesU,t,y)
261             plot(app.AxesU,t,ym,'r—','linewidth',1.1)
262             hold(app.AxesU)
263             legend(app.AxesU,{'$y(t)$','$y_m(t)$'},'Interpreter','latex',
                ', 'Location','best','FontSize',12);
264         end
265         clc
266         app.EditErr.Value=fonctions.IntSimpson((y-ym).^2,t(2));
267     end
268
269     % Value changed function: Moments
270     function MomentsValueChanged(app, event)
271         C = app.Moments.Value;
272         app.EditMom.Value=app.Mom(1+str2num(C(16:end)));
273     end
274 end
275
276 % App initialization and construction
277 methods (Access = private)
278

```

```

279 % Create UIFigure and components
280 function createComponents(app)
281
282 % Create IdentificationDesProcessusTravauxPratiquesUIFigure
283 app.IdentificationDesProcessusTravauxPratiquesUIFigure = uifigure;
284 app.IdentificationDesProcessusTravauxPratiquesUIFigure.Position =
    [100 100 640 480];
285 app.IdentificationDesProcessusTravauxPratiquesUIFigure.Name = '
    Identification Des Processus : Travaux Pratiques';
286
287 % Create MthodedesMomentsGnralissPanel
288 app.MthodedesMomentsGnralissPanel = uipanel(app.
    IdentificationDesProcessusTravauxPratiquesUIFigure);
289 app.MthodedesMomentsGnralissPanel.ForegroundColor = [0.9412 0.9412
    0.9412];
290 app.MthodedesMomentsGnralissPanel.TitlePosition = 'centertop';
291 app.MthodedesMomentsGnralissPanel.Title = 'Methode des Moments
    Generalises';
292 app.MthodedesMomentsGnralissPanel.BackgroundColor = [0.651 0.651
    0.651];
293 app.MthodedesMomentsGnralissPanel.FontName = 'Cambria';
294 app.MthodedesMomentsGnralissPanel.FontSize = 28;
295 app.MthodedesMomentsGnralissPanel.Position = [1 1 640 480];
296
297 % Create TabGroup
298 app.TabGroup = uitabgroup(app.MthodedesMomentsGnralissPanel);
299 app.TabGroup.Position = [-1 0 640 441];
300
301 % Create LECTURETab
302 app.LECTURETab = uitab(app.TabGroup);
303 app.LECTURETab.Title = ' LECTURE';
304 app.LECTURETab.BackgroundColor = [0.9412 0.9412 0.9412];
305
306 % Create TypeModleButtonGroup
307 app.TypeModleButtonGroup = uibuttongroup(app.LECTURETab);
308 app.TypeModleButtonGroup.SelectionChangedFcn = createCallbackFcn(
    app, @TypeModleButtonGroupSelectionChanged, true);
309 app.TypeModleButtonGroup.TitlePosition = 'centertop';
310 app.TypeModleButtonGroup.Title = 'Type Modele';
311 app.TypeModleButtonGroup.FontName = 'Cambria';
312 app.TypeModleButtonGroup.FontWeight = 'bold';
313 app.TypeModleButtonGroup.FontSize = 16;
314 app.TypeModleButtonGroup.Position = [256 330 356 61];
315
316 % Create FormePolynomialeButton
317 app.FormePolynomialeButton = uiradiobutton(app.TypeModleButtonGroup
    );
318 app.FormePolynomialeButton.Text = 'Forme Polynomiale';

```



```
319     app.FormePolynomialeButton.FontName = 'Cambria';
320     app.FormePolynomialeButton.FontSize = 16;
321     app.FormePolynomialeButton.Position = [15 9 154 22];
322     app.FormePolynomialeButton.Value = true;
323
324     % Create STREJCavecRetardButton
325     app.STREJCavecRetardButton = uiradiobutton(app.TypeModleButtonGroup
        );
326     app.STREJCavecRetardButton.Text = 'STREJC avec Retard';
327     app.STREJCavecRetardButton.FontName = 'Cambria';
328     app.STREJCavecRetardButton.FontSize = 16;
329     app.STREJCavecRetardButton.Position = [185 9 155 22];
330
331     % Create PanelPara
332     app.PanelPara = uipanel(app.LECTURETab);
333     app.PanelPara.TitlePosition = 'centertop';
334     app.PanelPara.Title = 'Parametres';
335     app.PanelPara.FontName = 'Cambria';
336     app.PanelPara.FontWeight = 'bold';
337     app.PanelPara.FontSize = 16;
338     app.PanelPara.Position = [26 262 207 129];
339
340     % Create EditField_13
341     app.EditField_13 = uieditfield(app.PanelPara, 'text');
342     app.EditField_13.Editable = 'off';
343     app.EditField_13.HorizontalAlignment = 'center';
344     app.EditField_13.BackgroundColor = [0.902 0.902 0.902];
345     app.EditField_13.Position = [13 56 84 34];
346
347     % Create TempsFinLabel
348     app.TempsFinLabel = uilabel(app.PanelPara);
349     app.TempsFinLabel.HorizontalAlignment = 'center';
350     app.TempsFinLabel.FontName = 'Cambria';
351     app.TempsFinLabel.FontSize = 14;
352     app.TempsFinLabel.Position = [12 56 85 34];
353     app.TempsFinLabel.Text = 'Temps Fin :';
354
355     % Create EditField_14
356     app.EditField_14 = uieditfield(app.PanelPara, 'text');
357     app.EditField_14.Editable = 'off';
358     app.EditField_14.HorizontalAlignment = 'center';
359     app.EditField_14.BackgroundColor = [0.902 0.902 0.902];
360     app.EditField_14.Position = [13 15 84 34];
361
362     % Create DcalageLabel
363     app.DcalageLabel = uilabel(app.PanelPara);
364     app.DcalageLabel.HorizontalAlignment = 'center';
365     app.DcalageLabel.FontName = 'Cambria';
```

```
366     app.DcalageLabel.FontSize = 14;
367     app.DcalageLabel.Position = [12 15 85 34];
368     app.DcalageLabel.Text = 'Decalage :';
369
370     % Create Spinnertf
371     app.Spinnertf = uispinner(app.PanelPara);
372     app.Spinnertf.Step = 5;
373     app.Spinnertf.Limits = [0 Inf];
374     app.Spinnertf.ValueDisplayFormat = '%5.4g';
375     app.Spinnertf.ValueChangedFcn = createCallbackFcn(app, @
        SpinnertfValueChanged, true);
376     app.Spinnertf.HorizontalAlignment = 'center';
377     app.Spinnertf.FontName = 'Cambria';
378     app.Spinnertf.FontSize = 14;
379     app.Spinnertf.Position = [96 56 100 34];
380     app.Spinnertf.Value = 25;
381
382     % Create Spinnera
383     app.Spinnera = uispinner(app.PanelPara);
384     app.Spinnera.Limits = [0 Inf];
385     app.Spinnera.ValueDisplayFormat = '%4.2g';
386     app.Spinnera.ValueChangedFcn = createCallbackFcn(app, @
        SpinneraValueChanged, true);
387     app.Spinnera.HorizontalAlignment = 'center';
388     app.Spinnera.FontName = 'Cambria';
389     app.Spinnera.FontSize = 14;
390     app.Spinnera.Position = [96 15 100 34];
391     app.Spinnera.Value = 2;
392
393     % Create PanelSR
394     app.PanelSR = uipanel(app.LECTURETab);
395     app.PanelSR.Position = [256 224 356 105];
396
397     % Create EditAmp
398     app.EditAmp = uieditfield(app.PanelSR, 'numeric');
399     app.EditAmp.ValueDisplayFormat = '%4.2g';
400     app.EditAmp.HorizontalAlignment = 'center';
401     app.EditAmp.FontName = 'Cambria';
402     app.EditAmp.FontSize = 14;
403     app.EditAmp.Position = [96 56 108 34];
404
405     % Create EditTem
406     app.EditTem = uieditfield(app.PanelSR, 'numeric');
407     app.EditTem.ValueDisplayFormat = '%4.2g';
408     app.EditTem.HorizontalAlignment = 'center';
409     app.EditTem.FontName = 'Cambria';
410     app.EditTem.FontSize = 14;
411     app.EditTem.Position = [88 13 87 34];
```

```
412
413 % Create EditOrd
414 app.EditOrd = uieditfield(app.PanelSR, 'numeric');
415 app.EditOrd.ValueDisplayFormat = '%.0f';
416 app.EditOrd.HorizontalAlignment = 'center';
417 app.EditOrd.FontName = 'Cambria';
418 app.EditOrd.FontSize = 14;
419 app.EditOrd.Position = [279 56 63 34];
420
421 % Create EditField_7
422 app.EditField_7 = uieditfield(app.PanelSR, 'text');
423 app.EditField_7.Eitable = 'off';
424 app.EditField_7.HorizontalAlignment = 'center';
425 app.EditField_7.BackgroundColor = [0.902 0.902 0.902];
426 app.EditField_7.Position = [14 56 83 34];
427
428 % Create AmplitudeLabel
429 app.AmplitudeLabel = uilabel(app.PanelSR);
430 app.AmplitudeLabel.HorizontalAlignment = 'center';
431 app.AmplitudeLabel.FontName = 'Cambria';
432 app.AmplitudeLabel.FontSize = 14;
433 app.AmplitudeLabel.Position = [14 56 83 34];
434 app.AmplitudeLabel.Text = 'Amplitude :';
435
436 % Create EditField_8
437 app.EditField_8 = uieditfield(app.PanelSR, 'text');
438 app.EditField_8.Eitable = 'off';
439 app.EditField_8.HorizontalAlignment = 'center';
440 app.EditField_8.BackgroundColor = [0.902 0.902 0.902];
441 app.EditField_8.Position = [212 56 68 34];
442
443 % Create OrdreLabel
444 app.OrdreLabel = uilabel(app.PanelSR);
445 app.OrdreLabel.HorizontalAlignment = 'center';
446 app.OrdreLabel.FontName = 'Cambria';
447 app.OrdreLabel.FontSize = 14;
448 app.OrdreLabel.Position = [212 56 68 34];
449 app.OrdreLabel.Text = 'Ordre :';
450
451 % Create EditField_9
452 app.EditField_9 = uieditfield(app.PanelSR, 'text');
453 app.EditField_9.Eitable = 'off';
454 app.EditField_9.HorizontalAlignment = 'center';
455 app.EditField_9.BackgroundColor = [0.902 0.902 0.902];
456 app.EditField_9.Position = [14 13 75 34];
457
458 % Create TempsLabel
459 app.TempsLabel = uilabel(app.PanelSR);
```

```
460     app.TempsLabel.HorizontalAlignment = 'center';
461     app.TempsLabel.FontName = 'Cambria';
462     app.TempsLabel.FontSize = 14;
463     app.TempsLabel.Position = [14 13 75 34];
464     app.TempsLabel.Text = 'Temps :';
465
466     % Create EditField_10
467     app.EditField_10 = uieditfield(app.PanelSR, 'text');
468     app.EditField_10.Editable = 'off';
469     app.EditField_10.HorizontalAlignment = 'center';
470     app.EditField_10.BackgroundColor = [0.902 0.902 0.902];
471     app.EditField_10.Position = [182 13 72 34];
472
473     % Create RetardLabel
474     app.RetardLabel = uilabel(app.PanelSR);
475     app.RetardLabel.HorizontalAlignment = 'center';
476     app.RetardLabel.FontName = 'Cambria';
477     app.RetardLabel.FontSize = 14;
478     app.RetardLabel.Position = [182 13 72 34];
479     app.RetardLabel.Text = 'Retard :';
480
481     % Create EditRet
482     app.EditRet = uieditfield(app.PanelSR, 'numeric');
483     app.EditRet.ValueDisplayFormat = '%4.2g';
484     app.EditRet.HorizontalAlignment = 'center';
485     app.EditRet.FontName = 'Cambria';
486     app.EditRet.FontSize = 14;
487     app.EditRet.Position = [253 13 89 34];
488
489     % Create PanelFP
490     app.PanelFP = uipanel(app.LECTURETab);
491     app.PanelFP.Position = [256 224 356 105];
492
493     % Create EditField_15
494     app.EditField_15 = uieditfield(app.PanelFP, 'text');
495     app.EditField_15.Editable = 'off';
496     app.EditField_15.HorizontalAlignment = 'center';
497     app.EditField_15.BackgroundColor = [0.902 0.902 0.902];
498     app.EditField_15.Position = [14 53 111 34];
499
500     % Create NumrateurLabel
501     app.NumrateurLabel = uilabel(app.PanelFP);
502     app.NumrateurLabel.HorizontalAlignment = 'center';
503     app.NumrateurLabel.FontName = 'Cambria';
504     app.NumrateurLabel.FontSize = 14;
505     app.NumrateurLabel.Position = [14 53 111 34];
506     app.NumrateurLabel.Text = 'Numerateur :';
507
```

```

508 % Create EditField_16
509 app.EditField_16 = uicontrolfield(app.PanelFP, 'text');
510 app.EditField_16.Eitable = 'off';
511 app.EditField_16.HorizontalAlignment = 'center';
512 app.EditField_16.BackgroundColor = [0.902 0.902 0.902];
513 app.EditField_16.Position = [14 16 111 34];
514
515 % Create DnominatorLabel
516 app.DnominatorLabel = uicontrolfield(app.PanelFP);
517 app.DnominatorLabel.HorizontalAlignment = 'center';
518 app.DnominatorLabel.FontName = 'Cambria';
519 app.DnominatorLabel.FontSize = 14;
520 app.DnominatorLabel.Position = [14 16 111 34];
521 app.DnominatorLabel.Text = 'Denominator :';
522
523 % Create EditNum
524 app.EditNum = uicontrolfield(app.PanelFP, 'text');
525 app.EditNum.HorizontalAlignment = 'center';
526 app.EditNum.FontName = 'Cambria';
527 app.EditNum.FontSize = 14;
528 app.EditNum.Position = [124 53 218 34];
529 app.EditNum.Value = '[ 0 ]';
530
531 % Create EditDen
532 app.EditDen = uicontrolfield(app.PanelFP, 'text');
533 app.EditDen.HorizontalAlignment = 'center';
534 app.EditDen.FontName = 'Cambria';
535 app.EditDen.FontSize = 14;
536 app.EditDen.Position = [124 16 218 34];
537 app.EditDen.Value = '[ 0 ]';
538
539 % Create ModlePrdfiniDropDownLabel
540 app.ModlePrdfiniDropDownLabel = uicontrolfield(app.LECTURETab);
541 app.ModlePrdfiniDropDownLabel.HorizontalAlignment = 'center';
542 app.ModlePrdfiniDropDownLabel.FontName = 'Cambria';
543 app.ModlePrdfiniDropDownLabel.FontSize = 14;
544 app.ModlePrdfiniDropDownLabel.FontWeight = 'bold';
545 app.ModlePrdfiniDropDownLabel.Position = [256 198 123 22];
546 app.ModlePrdfiniDropDownLabel.Text = 'Modele Predefini :';
547
548 % Create ModlePrdfiniDropDown
549 app.ModlePrdfiniDropDown = uicontrolfield(app.LECTURETab);
550 app.ModlePrdfiniDropDown.Items = {'(Aucun)', 'Systeme d''Ordre 1',
    'Systeme d''Ordre 2', 'STREJC d''Ordre 3', 'STREJC d''Ordre 4'
    };
551 app.ModlePrdfiniDropDown.FontName = 'Cambria';
552 app.ModlePrdfiniDropDown.FontSize = 14;
553 app.ModlePrdfiniDropDown.Position = [382 198 230 22];

```

```

554         app.ModlePrdfiniDropDown.Value = '(Aucun)';
555
556         % Create LIREButton
557         app.LIREButton = uibutton(app.LECTURETab, 'push');
558         app.LIREButton.ButtonPushedFcn = createCallbackFcn(app, @
            LIREButtonPushed, true);
559         app.LIREButton.FontName = 'Cambria';
560         app.LIREButton.FontSize = 16;
561         app.LIREButton.Position = [26 124 87 40];
562         app.LIREButton.Text = 'LIRE !';
563
564         % Create GsLabel
565         app.GsLabel = uilabel(app.LECTURETab);
566         app.GsLabel.HorizontalAlignment = 'center';
567         app.GsLabel.FontName = 'Courier New';
568         app.GsLabel.FontSize = 16;
569         app.GsLabel.FontWeight = 'bold';
570         app.GsLabel.Position = [119 124 63 40];
571         app.GsLabel.Text = 'G(s)=';
572
573         % Create LabelG
574         app.LabelG = uilabel(app.LECTURETab);
575         app.LabelG.FontName = 'Courier New';
576         app.LabelG.FontSize = 16;
577         app.LabelG.FontWeight = 'bold';
578         app.LabelG.Position = [162 103 452 82];
579         app.LabelG.Text = '  0';
580
581         % Create GmsLabel
582         app.GmsLabel = uilabel(app.LECTURETab);
583         app.GmsLabel.HorizontalAlignment = 'center';
584         app.GmsLabel.FontName = 'Courier New';
585         app.GmsLabel.FontSize = 16;
586         app.GmsLabel.FontWeight = 'bold';
587         app.GmsLabel.Position = [110 51 72 40];
588         app.GmsLabel.Text = 'Gm(s)=';
589
590         % Create LabelGM
591         app.LabelGM = uilabel(app.LECTURETab);
592         app.LabelGM.FontName = 'Courier New';
593         app.LabelGM.FontSize = 16;
594         app.LabelGM.FontWeight = 'bold';
595         app.LabelGM.Position = [162 30 450 82];
596         app.LabelGM.Text = '  ?';
597
598         % Create SIMULATIONTab
599         app.SIMULATIONTab = uitab(app.TabGroup);
600         app.SIMULATIONTab.Title = '  SIMULATION';

```

```

601
602 % Create EditU
603 app.EditU = uicontrol(app.SIMULATIONTab, 'text');
604 app.EditU.FontName = 'Courier New';
605 app.EditU.FontSize = 14;
606 app.EditU.FontWeight = 'bold';
607 app.EditU.BackgroundColor = [0.9412 0.9412 0.9412];
608 app.EditU.Position = [68 362 390 34];
609 app.EditU.Value = 'min(exp(t-5),exp(-t+5))-exp(-5)';
610
611 % Create LIREU
612 app.LIREU = uicontrol(app.SIMULATIONTab, 'push');
613 app.LIREU.ButtonPushedFcn = createCallbackFcn(app, @
    LIREUButtonPushed, true);
614 app.LIREU.FontName = 'Cambria';
615 app.LIREU.FontSize = 16;
616 app.LIREU.Position = [532 359 88 40];
617 app.LIREU.Text = 'SIMULER !';
618
619 % Create UtLabel
620 app.UtLabel = uicontrol(app.SIMULATIONTab);
621 app.UtLabel.HorizontalAlignment = 'center';
622 app.UtLabel.FontName = 'Courier New';
623 app.UtLabel.FontSize = 14;
624 app.UtLabel.FontWeight = 'bold';
625 app.UtLabel.Position = [22 362 47 34];
626 app.UtLabel.Text = 'U(t)=';
627
628 % Create AxesU1
629 app.AxesU1 = uiaxes(app.SIMULATIONTab);
630 title(app.AxesU1, ' ')
631 ylabel(app.AxesU1, 'Input')
632 app.AxesU1.FontName = 'Cambria';
633 app.AxesU1.FontSize = 10;
634 app.AxesU1.XGrid = 'on';
635 app.AxesU1.YGrid = 'on';
636 app.AxesU1.Position = [22 186 597 165];
637
638 % Create AxesU2
639 app.AxesU2 = uiaxes(app.SIMULATIONTab);
640 title(app.AxesU2, ' ')
641 xlabel(app.AxesU2, 'Temps (s)')
642 ylabel(app.AxesU2, 'Output')
643 app.AxesU2.FontName = 'Cambria';
644 app.AxesU2.FontSize = 10;
645 app.AxesU2.XGrid = 'on';
646 app.AxesU2.YGrid = 'on';
647 app.AxesU2.Position = [22 15 597 165];

```

```
648
649 % Create Poids
650 app.Poids = uicheckbox(app.SIMULATIONTab);
651 app.Poids.Text = 'Poids';
652 app.Poids.FontName = 'Cambria';
653 app.Poids.FontSize = 14;
654 app.Poids.Position = [470 368 55 22];
655
656 % Create IDENTIFICATIONTab
657 app.IDENTIFICATIONTab = uitab(app.TabGroup);
658 app.IDENTIFICATIONTab.Title = ' IDENTIFICATION';
659
660 % Create TypeRponseButtonGroup
661 app.TypeRponseButtonGroup = uibuttongroup(app.IDENTIFICATIONTab);
662 app.TypeRponseButtonGroup.TitlePosition = 'centertop';
663 app.TypeRponseButtonGroup.Title = 'Type Reponse';
664 app.TypeRponseButtonGroup.FontName = 'Cambria';
665 app.TypeRponseButtonGroup.FontWeight = 'bold';
666 app.TypeRponseButtonGroup.FontSize = 16;
667 app.TypeRponseButtonGroup.Position = [23 339 243 61];
668
669 % Create ImpulsionnelleButton
670 app.ImpulsionnelleButton = uiradiobutton(app.TypeRponseButtonGroup)
671 ;
672 app.ImpulsionnelleButton.Text = 'Impulsionnelle';
673 app.ImpulsionnelleButton.FontName = 'Cambria';
674 app.ImpulsionnelleButton.FontSize = 16;
675 app.ImpulsionnelleButton.Position = [15 9 125 22];
676 app.ImpulsionnelleButton.Value = true;
677
678 % Create IndicielleButton
679 app.IndicielleButton = uiradiobutton(app.TypeRponseButtonGroup);
680 app.IndicielleButton.Text = 'Indicielle';
681 app.IndicielleButton.FontName = 'Cambria';
682 app.IndicielleButton.FontSize = 16;
683 app.IndicielleButton.Position = [144 9 85 22];
684
685 % Create AxesU
686 app.AxesU = uiaxes(app.IDENTIFICATIONTab);
687 title(app.AxesU, ' ')
688 xlabel(app.AxesU, 'Temps (s)')
689 app.AxesU.FontName = 'Cambria';
690 app.AxesU.FontSize = 11;
691 app.AxesU.XGrid = 'on';
692 app.AxesU.YGrid = 'on';
693 app.AxesU.Position = [23 19 597 301];
694
695 % Create EditField_18
```



```

695     app.EditField_18 = uieditfield(app.IDENTIFICATIONTab, 'text');
696     app.EditField_18.Eitable = 'off';
697     app.EditField_18.HorizontalAlignment = 'center';
698     app.EditField_18.BackgroundColor = [0.902 0.902 0.902];
699     app.EditField_18.Position = [279 366 67 34];
700
701     % Create ErreurLabel
702     app.ErreurLabel = uilabel(app.IDENTIFICATIONTab);
703     app.ErreurLabel.HorizontalAlignment = 'center';
704     app.ErreurLabel.FontName = 'Cambria';
705     app.ErreurLabel.FontSize = 14;
706     app.ErreurLabel.Position = [279 366 67 34];
707     app.ErreurLabel.Text = 'Erreur :';
708
709     % Create EditErr
710     app.EditErr = uieditfield(app.IDENTIFICATIONTab, 'numeric');
711     app.EditErr.Eitable = 'off';
712     app.EditErr.HorizontalAlignment = 'center';
713     app.EditErr.FontName = 'Cambria';
714     app.EditErr.FontSize = 14;
715     app.EditErr.Position = [345 366 152 34];
716
717     % Create Moments
718     app.Moments = uidropdown(app.IDENTIFICATIONTab);
719     app.Moments.Items = {'Moments'};
720     app.Moments.ValueChangedFcn = createCallbackFcn(app, @
        MomentsValueChanged, true);
721     app.Moments.FontName = 'Cambria';
722     app.Moments.Position = [279 339 133 22];
723     app.Moments.Value = 'Moments';
724
725     % Create IDENTIFIER
726     app.IDENTIFIER = uibutton(app.IDENTIFICATIONTab, 'push');
727     app.IDENTIFIER.ButtonPushedFcn = createCallbackFcn(app, @
        IDENTIFIERButtonPushed, true);
728     app.IDENTIFIER.FontName = 'Cambria';
729     app.IDENTIFIER.FontSize = 16;
730     app.IDENTIFIER.Position = [513 360 107 40];
731     app.IDENTIFIER.Text = 'IDENTIFIER !';
732
733     % Create EditMom
734     app.EditMom = uieditfield(app.IDENTIFICATIONTab, 'numeric');
735     app.EditMom.ValueDisplayFormat = '%2.4g';
736     app.EditMom.Eitable = 'off';
737     app.EditMom.HorizontalAlignment = 'center';
738     app.EditMom.FontName = 'Cambria';
739     app.EditMom.FontSize = 14;
740     app.EditMom.Position = [411 339 86 22];

```

```
741         end
742     end
743
744     methods (Access = public)
745
746         % Construct app
747         function app = IDP_MMG_APP
748
749             % Create and configure components
750             createComponents(app)
751
752             % Register the app with App Designer
753             registerApp(app, app.
                IdentificationDesProcessusTravauxPratiquesUIFigure)
754
755             if nargin == 0
756                 clear app
757             end
758         end
759
760         % Code that executes before app deletion
761         function delete(app)
762
763             % Delete UIFigure when app is deleted
764             delete(app.IdentificationDesProcessusTravauxPratiquesUIFigure)
765         end
766     end
767 end
```

Listing 3 – InterfaceGraphique

5 Exemples

5.1 Forme de STREJ avec retard

Comme on le remarque sur la figure suivant ce n'est pas nécessaire que le temps d'observation soit plus grand que le temps de réponse du système pour avoir un résultat, car vu la fonction de pondération e^{-at} ne gardera que le début de simulation (il faut s'assurer que après pondération la fonction converge)

La réponse introduite est celle du système : $F(s) = \frac{4e^{-0.2s}}{(1 + 0.4s)^3}$

Identification Des Processus : Travaux Pratiques

Méthode des Moments Généralisés

LECTURE SIMULATION IDENTIFICATION

Paramètres

Temps Fin : 14

Décalage : 1.2

Type Modèle

☒ Forme Polynomiale ☐ STREJC avec Retard

Numérateur : [0]

Dénominateur : [0]

Modèle Prédéfini : STREJC d'Ordre 3

LIRE ! $G(s) = \exp(-0.2*s) * \frac{4}{(1+0.4s)^3}$

$G_m(s) = \exp(-0.2*s) * \frac{4.0081}{(1+0.4005s)^3}$

FIGURE 6 – Forme de STREJ LECTURE

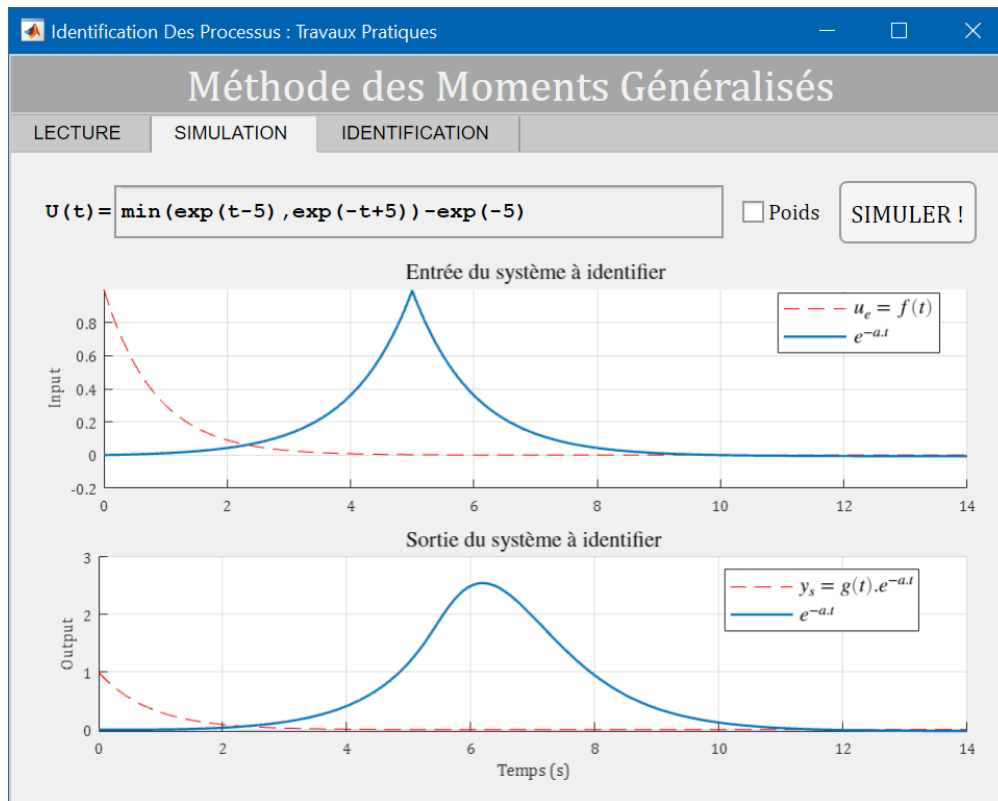


FIGURE 7 – Forme de STREJ SIMULATION sans pondération

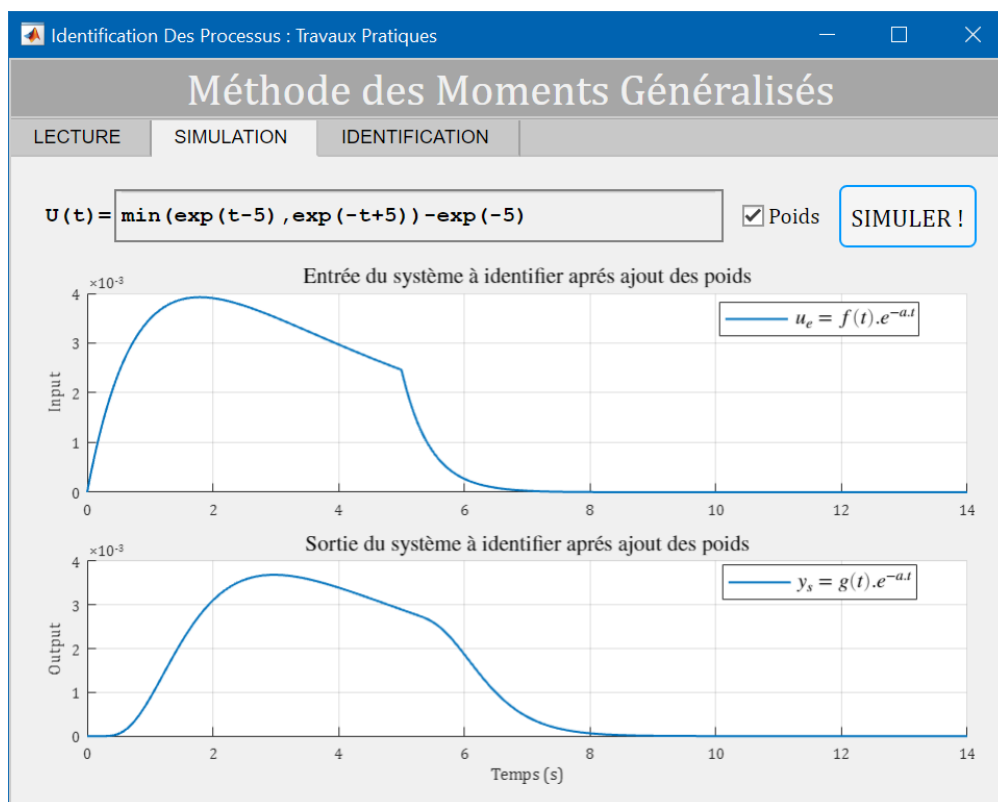


FIGURE 8 – Forme de STREJ SIMULATION avec pondération

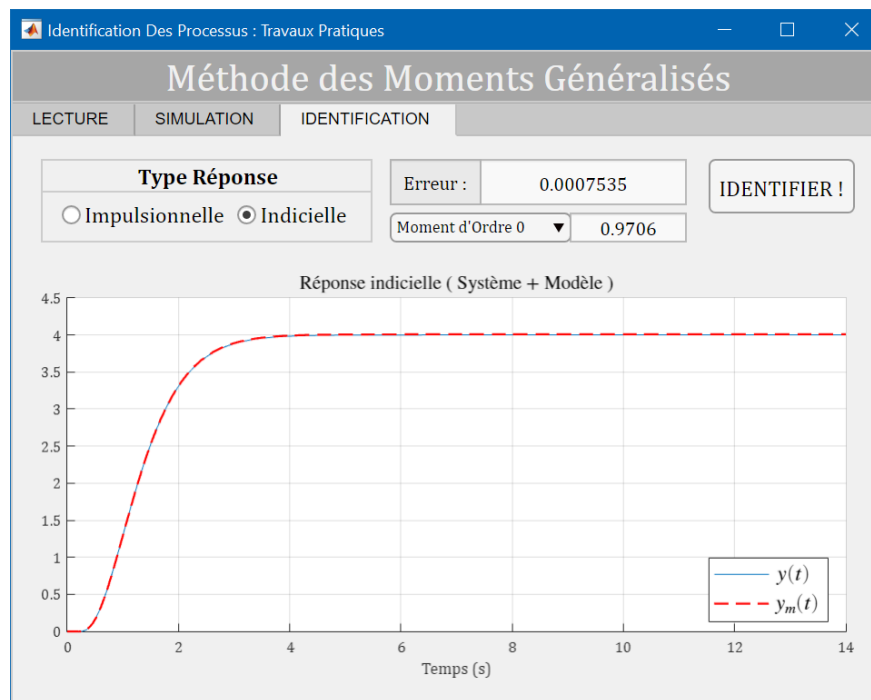


FIGURE 9 – Forme de STREJ IDENTIFICATION

5.2 Forme polynomiale, avec variation de (a)

Pour cette partie on a choisi la forme polynomiale suivante : $F(s) = \frac{2s + 1}{s^2 + s + 5}$

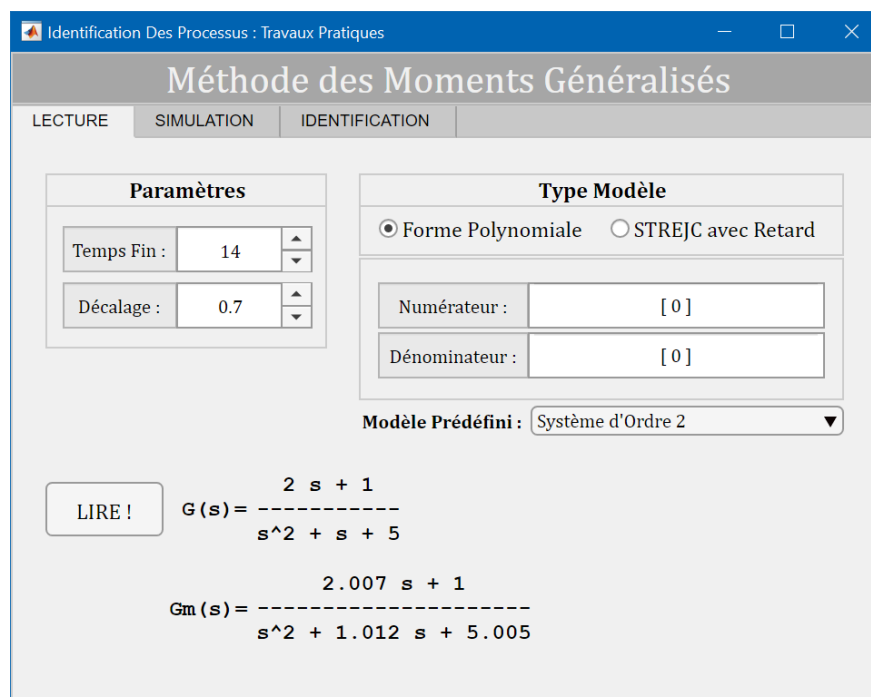


FIGURE 10 – Forme polynomiale LECTURE

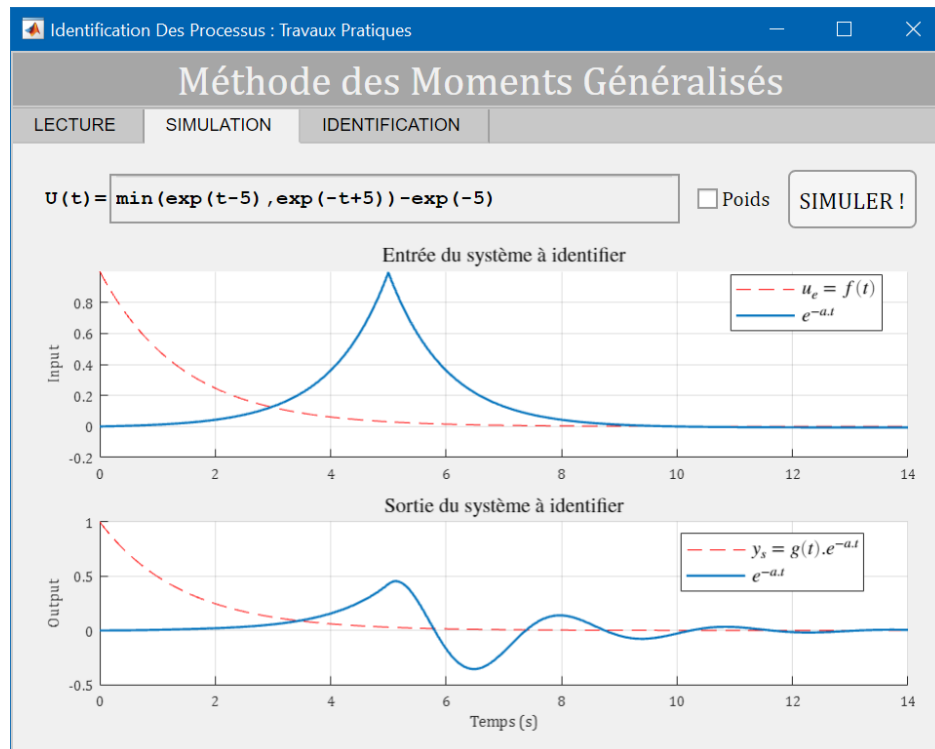
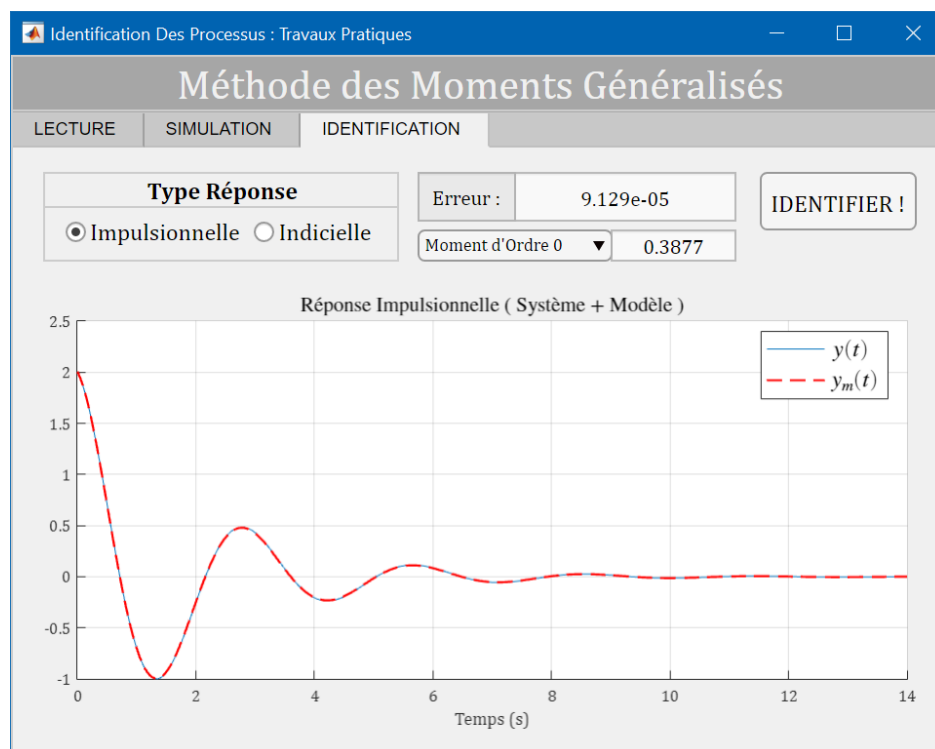
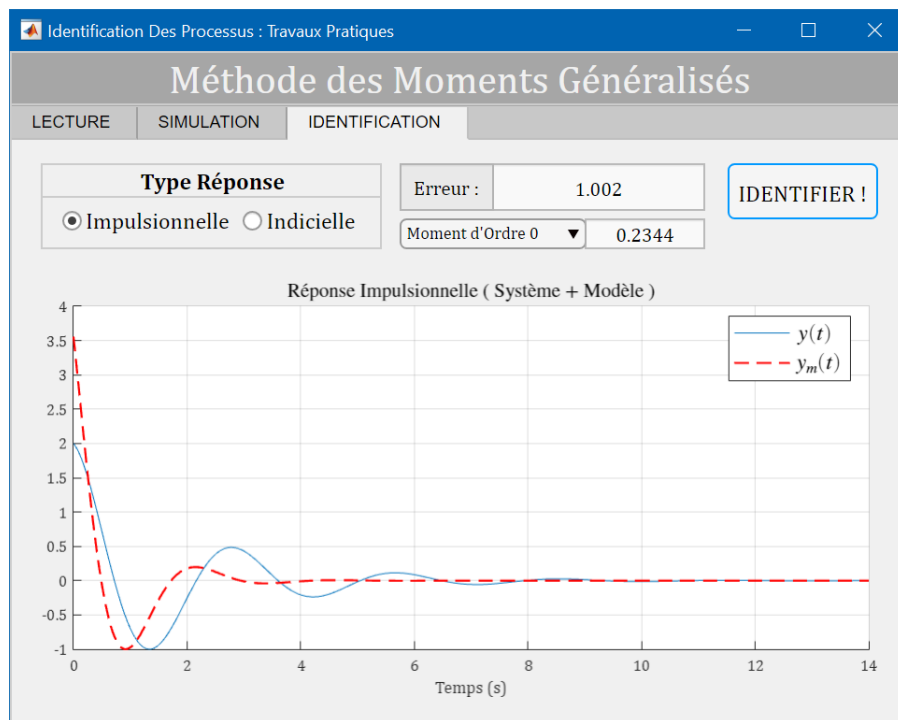
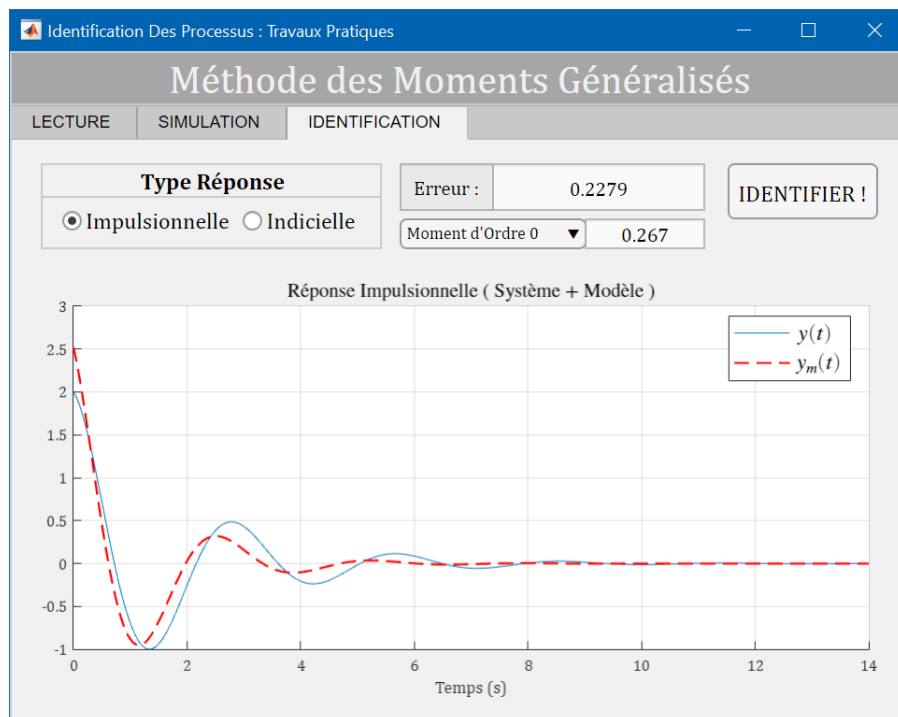


FIGURE 11 – Forme polynomiale SIMULATION

FIGURE 12 – Forme polynomiale IDENTIFICATION pour $a = 0.7$

L'erreur quadratique et le graphe montrent que le système a bien été identifié.

Variation de (a)

FIGURE 13 – Forme polynomiale, $a = 0.1$ FIGURE 14 – Forme polynomiale, $a = 0.2$

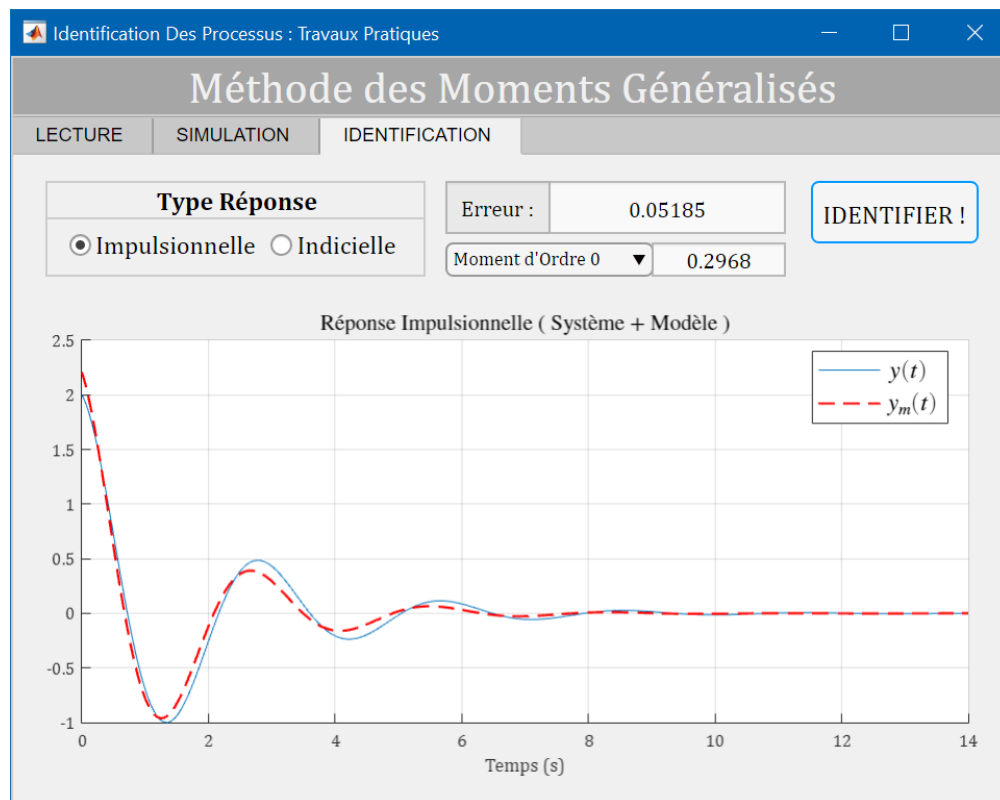


FIGURE 15 – Forme polynomiale, $a = 0.3$

Comme cité dans la partie remarques plus (a) est grand plus l'erreur quadratique est petite.

5.3 Système instable

Pour montrer la puissance de cette méthode on a choisi de faire un exemple avec un système instable de la forme : $F(s) = \frac{1}{s-1}$
 Il faut faire attention à la convergence de l'intégrale si ce n'est pas le cas il faut augmenter la valeur du paramètre (a).

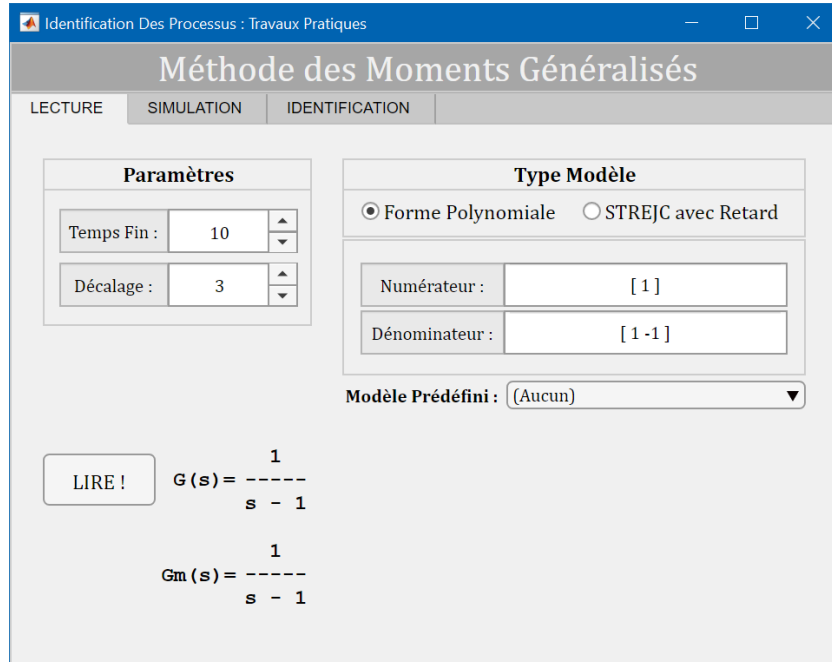


FIGURE 16 – Système instable LECTURE

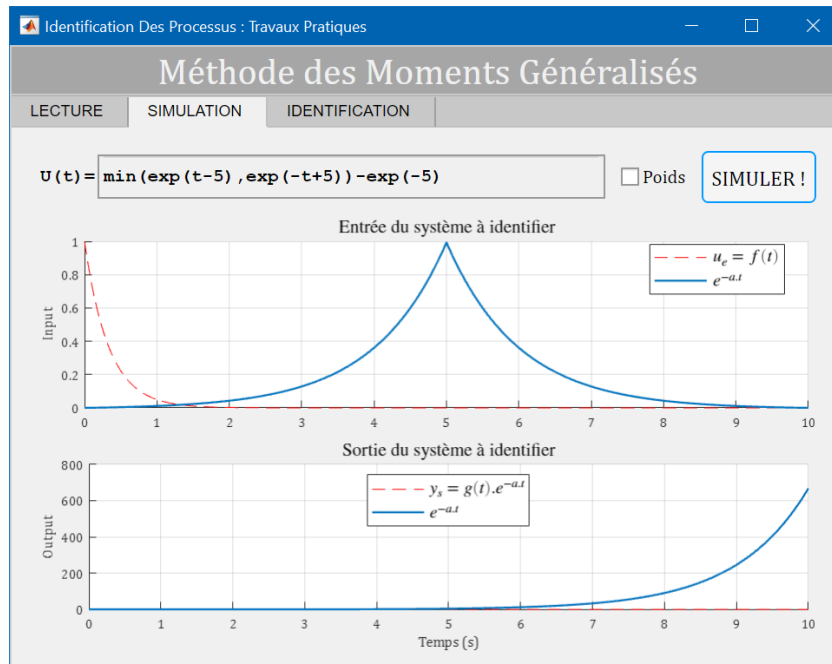


FIGURE 17 – Système instable SIMULATION

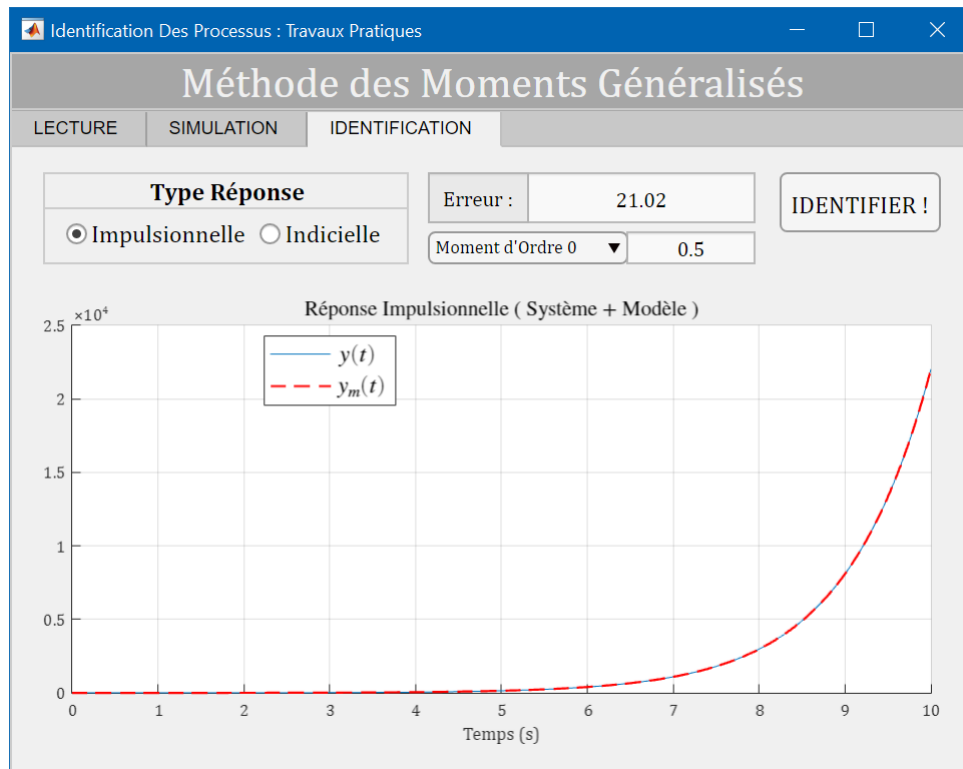


FIGURE 18 – Système instable IDENTIFICATION

Même si le système diverge on remarque que le système a été bien identifié ce qui démontre la puissance de cette méthode.

6 Conclusion

La méthode des moments généralisés appliquée en identification assistée par ordinateur s'avère être à la fois rapide et précise, personnellement, on ne s'attendait pas à de si bon résultats.

En particulier, comparée à la méthode des moments, celle-ci donne à l'utilisateur la possibilité de jouer sur un certain nombre de degrés de liberté, ce qui la rend plus flexible et facilement opérable. Les répercussions, niveau complexité (Script/Algorithme), sont une intégrale sur la fonction étant pondérée et une formule de calcul des moments plus large.

En générale, l'identification par moments généralisés est :

performante : étant donné des signaux quelconques en entrée, et appliquée à des procédés de nature apériodique, périodique, ou même instable, elle y répond toujours correctement.

faiblement couteuse : les temps infimes d'exécution de l'algorithme en sont la meilleure preuve.