

Projet 47 IMA4 2016-2017

POLYTECH'LILLE

Modélisation d'un robot mobile

Réalisé par : MOHAMED Djamil & KERROUM Hamza

Encadré par : Mr. PEKPE Komi Midzodzi

REMERCIEMENTS

On tient à présenter nos sincères remerciements et notre respect à Mr.PEKPE Komi Midzodzi pour son encadrement, son bon sens d'orientation et ses remarques pertinentes. On voudrait le remercier pour tous les efforts qu'il nous a accordés, ainsi que son temps et sa patience.

Mr.PEKPE nous a fait l'honneur d'être notre encadrant, pour cela, ainsi que pour ses conseils et ses recommandations sur notre travail, on lui exprime notre profonde reconnaissance et notre énorme respect.

Sommaire

REMERCIEMENT.....	2
INTRODUCTION GENERALE.....	4
Partie 1: Présentation matérielle et logicielle.....	5
1-Le ROBOTINO.....	6
2- MATLAB Simulink.....	9
Partie 2: Méthode d'identification des sous-espaces.....	10
1-La méthode d'identification N4SID.....	11
Partie 3: Identification et validation du modèle.....	16
1- Identification et validation du modèle hors-ligne.....	17
2- Validation du modèle en ligne.....	19
CONCLUSION.....	23

INTRODUCTION GENERALE

Ce rapport recueille les étapes de la modélisation d'un robot mobile, et a été réalisé pour le projet de 4ème année IMA durant le semestre 8. Le projet en lui-même nous a permis d'expérimenter ce que nous avons appris durant notre cursus en IMA et d'acquérir de nouvelles compétences.

Pour ce projet on a choisi "la modélisation d'un robot mobile" proposé par Mr.PEKPE, car pour nous les notions traitées dans ce projet sont les plus proches de notre formation "Systèmes Autonome" en ce semestre.

Permettant l'obtention d'un modèle mathématique des systèmes pour réaliser leur commande, la modélisation est primordiale pour plusieurs disciplines comme l'aéronautique, la robotique, etc. La modélisation nous permet de comprendre le fonctionnement d'un système afin d'assurer et de surveiller, par correction son comportement et son fonctionnement au cas où un défaut ou une défaillance surgit.

Cette modélisation peut se faire par des lois physiques ou encore par des données. Elle se fera suivant plusieurs étapes sur le robot à trois roues omnidirectionnelles (Le Robotino). Tout d'abord les données collectées sur ce robot, la détermination du modèle à partir de ces mesures, la validation du modèle avec d'autres mesures qui n'ont pas servi à le déterminer et le test qui se fera en mode en ligne sur le modèle.

La validation consistera à comparer le comportement du modèle à celui du robot en appliquant les mêmes commandes à eux deux.

Partie 1: Présentation matérielle et logicielle

1- ROBOTINO :

Dans de nombreux milieux et plus particulièrement de celui de l'industrie, la robotique mobile autonome est de plus en plus présente. Elle est utilisée dans principalement pour le transport de matérielle ou dans la logistique. Elle permet alors une grande flexibilité de flux de matière.



Figure 1 : Robotino

Le Robotino a été créé en partant d'applications industrielles concrètes. Il est la plateforme de robot mobile pour la recherche et l'éducation. C'est un système mécatronique mobile complet qui fait de lui un concentré de technologies.

Le Robotino a un concept atypique. Le diamètre de son châssis est de 350mm et sa hauteur est 200mm. Il pèse 11 Kg et a une capacité de charge de 6 Kg. Ces principales composantes sont un PC-104 embarqué, une IHM (Interface Homme-Machine), une caméra, un jeu de batteries, un châssis en acier, un capteur de choc le long du châssis, neuf capteurs infrarouges analogiques, trois encodeurs et de trois moteurs DC avec roues omnidirectionnelles.



Figure 2 : Interface homme machine

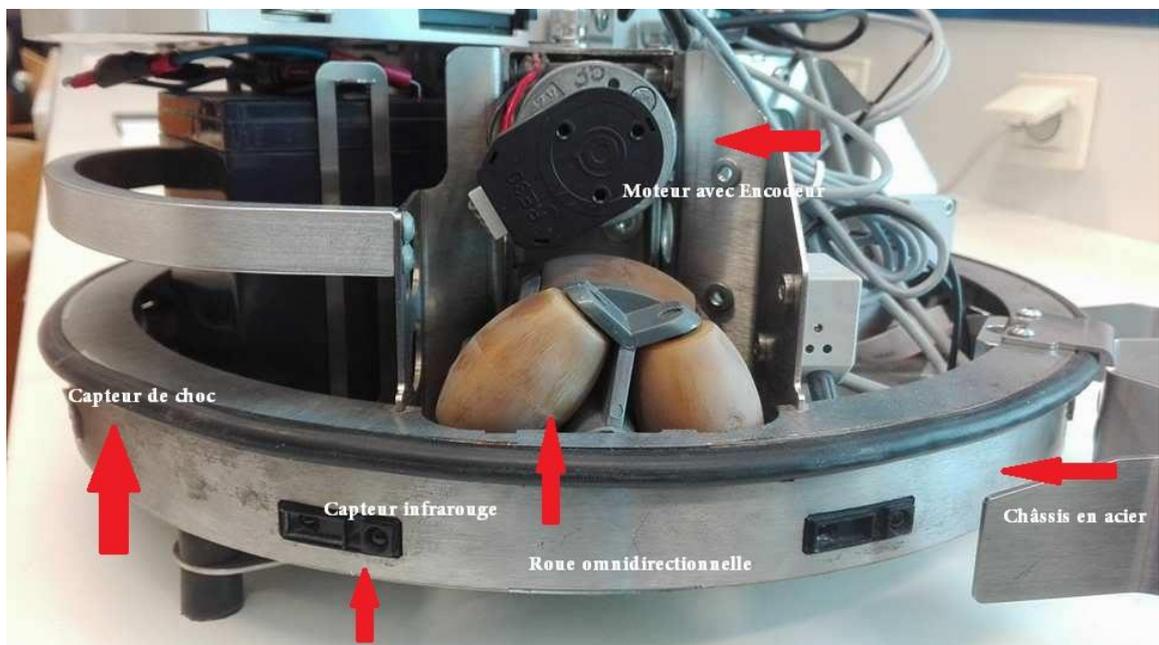


Figure 3 : Différents composants du Robotino

Sa partie commande est très complète car elle est composée d'un processeur PC cadencée à 500 MHz, d'une SDRAM 128 MB, d'une carte mémoire 1 GO, d'un OS Linux, d'une point d'accès LAN Wireless, d'un port Ethernet, de deux port USB 2.0 et d'un port VGA. La communication avec le Robotino peut donc être réalisée soit par Wifi, soit avec un câble Ethernet.

L'extension de capteurs et/ou actionneurs est possible car le Robotino possède huit entrées analogiques, huit E/S digitales et deux relais pour des actionneurs supplémentaires. Les Robotinos présents à l'école possèdent deux types de capteurs supplémentaires qui ont été ajouté grâce à cela : un capteur inductif et deux capteurs à fibre optiques.

Le Robotino a l'avantage d'avoir trois roues omnidirectionnelles lui permettant de faire des mouvements dans toutes les directions. Ces roues sont reliées à trois moteurs avec un encodeur par moteur.

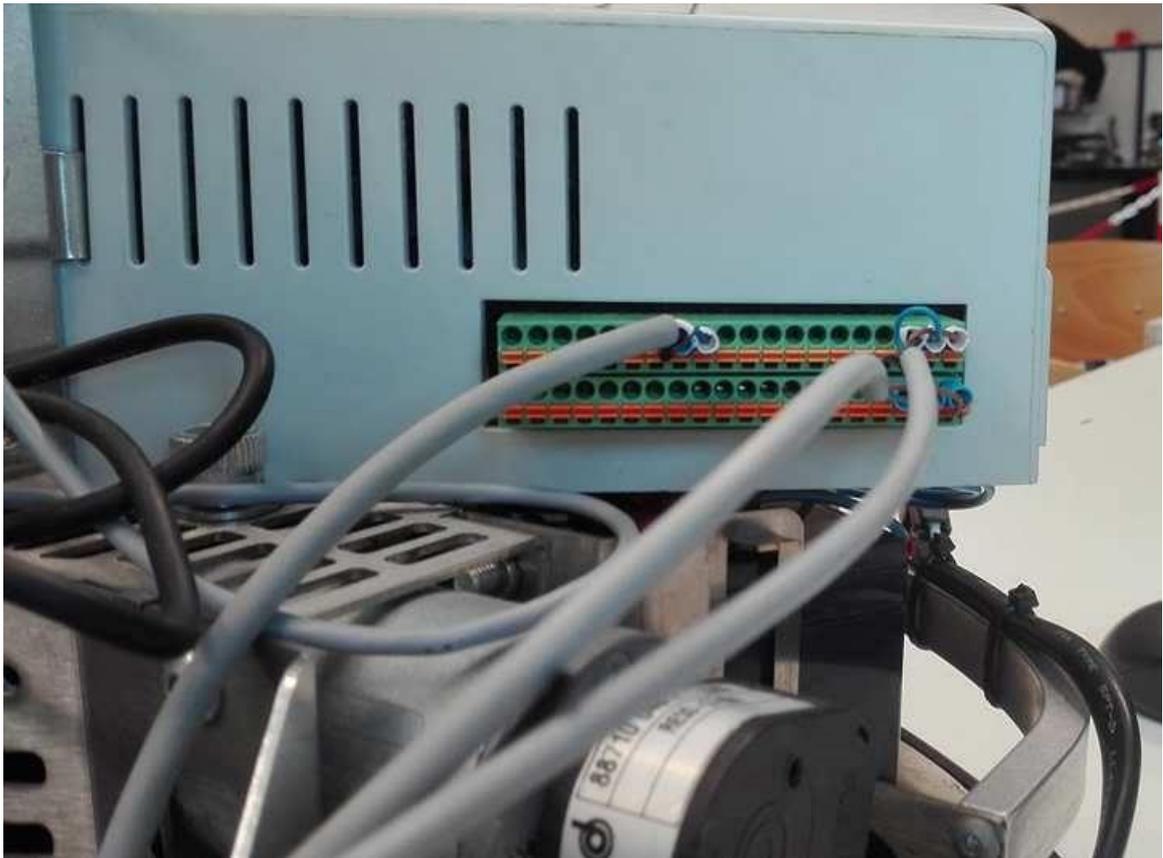


Figure 4 : Entrées analogiques et E/S digitales

Robotino est une plateforme de programmation très évoluée. Les langages de programmation les plus importants et les systèmes sont disponibles pour les applications individuelles de programmation. Il peut être programmé avec RobotinoView, Visual Basic, Matlab, Labview ou en C, C++ et encore JAVA.

Comme nous le voyons, Robotino peut être utilisé de manière très flexible.

2-Matlab Simulink :

MATLAB vient de « Matrix Laboratory » est un langage de programmation. Il est utilisé afin de d'analyser et concevoir des systèmes ou produits. Ce langage est présent dans des réseaux électriques intelligents, des systèmes automobiles ou encore des véhicules spatiaux. Il est utilisé dans le domaine de l'automatique, la robotique, la conception de de contrôleurs et bien plus.

MATLAB est aussi appelé le langage du calcul technique. MATLAB est perfectionné afin de résoudre les problèmes scientifiques et techniques. Il est l'outil le plus adapté pour exprimer les mathématiques computationnelles. Nous pouvons facilement dégager les informations recherchées à l'aide de ces graphiques intégrés.

C'est pour cela que MATLAB est l'outil le plus adapté pour notre application et la modélisation d'un système. Il possède également une grande bibliothèque de boîtes à outils prédéfinies ce qui est un fort gain de temps pour l'utilisation d'un algorithme. Dans notre cas, nous allons utiliser la méthode N4SID dans notre identification du Robotino et cet algorithme est présent sous MATLAB.

Simulink est quant à lui un logiciel de modélisation destiné à la simulation multidomaine. Il est composé d'un environnement graphique et d'un ensemble de bibliothèques contenant des blocs de modélisation. Il est intégré à MATLAB, ainsi nous pouvons insérer les algorithmes MATLAB dans les modèles de Simulink et/ou exporter le résultat d'une simulation vers MATLAB afin de parfaire des analyses.

Partie 2: Méthode d'identification des sous-espaces

1-La méthode d'identification N4SID :

La méthode des sous – espaces, est une technique qui permet l'identification du modèle estimé d'un système. Il existe plusieurs méthodes dont la « n4sid », une méthode dite classique qui nous sera utile pour estimer l'état du modèle. L'algorithme de la méthode est présenté comme ceci :

q : 2 indices i et j tel que $q = j+2i-1$ avec $i < j$, où q est le nombre des mesures des entrées et des sorties.

Dans notre système, nous utiliserons la matrice d'Hankel d'entrée passée U_0 , et la matrice d'entrée future U_i , ainsi que la matrice de sortie passée Y_0 et de sortie future Y_i .

Matrice de Hankel mesures passées : $T_0 = \begin{pmatrix} Y_0 \\ U_0 \end{pmatrix}$.

Pareil bruit d'état passé V_0 , future V_i et bruit de sortie passé W_0 et future W_i .

Pour l'identification du système, c'est d'estimer la matrice d'observabilité étendue ou la matrice des séquences d'états.

Tout d'abord, la relation matricielle de base dans les méthodes classique des sous-espaces est : $Y_i = \Gamma_i \cdot X_i + M_i^V \cdot V_i + w_i$, où :

W_i =matrice bruit de sortie future,

V_i bruit d'état futur,

M_i^V Toeplitz inférieure ,

X_i état future,

Γ_i Observabilité étendue,

Y_i sortie future.

On va supprimer l'influence des entrées pour avoir plus que l'action de l'état en multipliant tous les paramètres par $\Pi_{U_i}^\perp$:

$$\Pi_{U_i}^\perp = (I - \Pi_{U_i}) \text{ Et } \Pi_{U_i} = \Pi_{U_i}^T (U_i \cdot U_i^T)^{(-)} \cdot U_i$$

$$\text{Donc } \Pi_{U_i}^\perp = I - U_i^T (\Pi_{U_i} \Pi_{U_i} = \Pi_{U_i}^T (U_i \cdot U_i^T)^{(-)} \cdot U_i$$

$$Y_i \cdot \Pi_{U_i}^\perp = \Gamma_i \cdot X_i \cdot \Pi_{U_i}^\perp + H_i^V \cdot V_i \cdot \Pi_{U_i}^\perp + W_i \cdot \Pi_{U_i}^\perp$$

Les bruits ne sont pas corrélés aux entrées donc $E[V_i \cdot \Pi_{U_i}^\perp] = E[W_i \cdot V_i \cdot \Pi_{U_i}^\perp] = 0$

On améliore notre estimation en multipliant nos paramètres par W_1 et W_2 qui sont des matrices de Hankel du bruit future

Ces deux matrices permettent la conservation du rang de la matrice Γ_1 . X1si $\text{rang}(W_1. \Gamma_1) = \text{rang}(\Gamma_1)$ et $\text{rang}(X_i. \Pi_{U_i}^\perp. W_2) = \text{rang}(X_i)$

Le rang n de l'ordre du système c'est le $\text{rang}(T_0. W_1. Y_i. \Pi_{U_i}^\perp. W_2)$, décomposition en valeur singulière. Et

$$W_1. Y_i. \Pi_{U_i}^\perp. W_2 = (U_1 \quad U_2) \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

Où U_1 et U_2 sont les matrices de Hankel d'entrée future

V_1 et V_2 matrice de Hankel de bruit d'état future

Permettant ainsi l'estimation par :

$$W_1. \Gamma_i = U_1. S_1^{1/2}$$

$$X_i. \Pi_{U_i}^\perp. W_2 = S_1^{1/2}. V_1^T$$

Système discret :

A, B, C et D sont les paramètres du système, X_k l'état du système, U_k son entrée et Y_k sa sortie

Le but est de trouver les matrices A, B, C et D ou une résolution équivalente $T^{-1}. A. T, T^{-1}. B, C. T, D$ où T est une matrice inversible

$$X_{k+1} = A. X_k + B. U_k$$

$$Y_k = C. X_k + D. U_k$$

On a :

$$X_{k+2} = A. X_{k+1} + B. U_{k+1}, \text{ en remplaçant } X_{k+1} \text{ par sa valeur :}$$

$$X_{k+2} = A. (A. X_k + B. U_k) + B. U_{k+1} = A^2. X_k + A. B. U_k + B. U_{k+1}$$

$$X_{k+3} = A^3. X_k + A^2. B. U_k + A. B. U_{k+1} + B. U_{k+2}$$

Donc de façon générale :

$$X_{k+i} = A^i. X_k + A^{i-1}. B. U_k + A^{i-2}. B. U_{k+1} + \dots + B. U_{k+i-1}$$

Sous forme matricielle :

$$X_{k+i} = A^i. X_k + [A^{i-1}. B \mid A^{i-2}. B \mid \dots \mid AB \mid B] \begin{bmatrix} U_k \\ \vdots \\ U_{k+i-1} \end{bmatrix}$$

$$\text{Où : } [A^{i-1}. B \mid A^{i-2}. B \mid \dots \mid AB \mid B] = \bar{H}$$

$$\text{Et } \begin{bmatrix} U_k \\ \vdots \\ U_{k+i-1} \end{bmatrix} = \overline{U_k}$$

$$X_{k+i} = A^i \cdot X_k + \overline{H} \cdot \overline{B} \cdot \overline{U_{k+1}}$$

$$Y_k = C \cdot X_k + D \cdot U_k$$

$$Y_{k+1} = C \cdot A (X_k + \overline{H} \cdot \overline{U_{k+1}}) + D \cdot U_{k+1}$$

$$Y_{k+i} = C \cdot A^i \cdot X_k + C \cdot \overline{H} \cdot \overline{U_{k+1}} + D \cdot U_{k+1}$$

$$C \cdot \overline{H} \cdot \overline{U_{k+1}} = [C \cdot A^{i-1} \cdot B \mid C \cdot A^{i-2} \cdot B \mid \dots \mid C \cdot A \cdot B \mid C \cdot B]$$

$$C \cdot \overline{H} \cdot \overline{U_{k+1}} = [C \cdot A^{i-1} \cdot B \mid C \cdot A^{i-2} \cdot B \mid \dots \mid C \cdot A \cdot B \mid C \cdot B] \begin{bmatrix} U_k \\ \vdots \\ U_{k+i} \end{bmatrix}$$

$$Y_{k+i} = C \cdot A^i \cdot X_k + H_i \cdot \tilde{U}_k$$

$$Y_k = C \cdot A^i \cdot X_{k-i} + H_i \cdot \tilde{U}_{k-i}$$

Connaissons-nous Z, M et E ?

$Z = M \cdot E$, où M appartient à $R^{n1 \cdot n2}$ avec $n1 > n2$ (plus de lignes que de colonnes)

$$Z1 = m_1^1 \cdot e_1^1 + m_2^1 \cdot e_2^1$$

$$Z2 = m_1^2 \cdot e_1^1 + m_2^2 \cdot e_2^1 \quad \Leftrightarrow \quad \begin{bmatrix} Z1 \\ Z2 \\ Z3 \end{bmatrix} = \begin{bmatrix} m_1^1 & m_2^1 \\ m_1^2 & m_2^2 \\ m_1^3 & m_2^3 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

$$Z3 = m_1^3 \cdot e_1^1 + m_2^3 \cdot e_2^1$$

On peut déterminer E si $\text{rang}(M) = n2$

M appartient à $R^{n1 \cdot n2}$ donc $\text{rang}(M) \leq \min(n1, n2)$

$$E = M^T \cdot Z, \text{ avec } M^T = (M^T \cdot M)^{-1} \cdot M^T$$

Où M^T en pseudo inverse de Moore Penrose

où M appartient à $R^{n1 \cdot n2}$, M^T appartient à $R^{n2 \cdot n1}$ et $M^T \cdot M$ appartient à $R^{n2 \cdot n2}$

$$Z = M \cdot E \Leftrightarrow M^T \cdot Z = (M^T \cdot M)^{-1} \cdot M^T \cdot M \cdot E = E$$

$$\text{Où : } (M^T \cdot M)^{-1} \cdot M^T = M^T \text{ Et } (M^T \cdot M)^{-1} \cdot M^T \cdot M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I_3$$

$$\begin{aligned} [Y_{k+i} | Y_{k+i+1} | \dots | Y_{k+j}] &= [C \cdot A^i \cdot X_{k+1} + H_i \cdot \tilde{U}_{k+1} | \dots | C \cdot A^i \cdot X_{k+j} + H_i \cdot \tilde{U}_{k+j}] \\ &= C \cdot A^i \cdot [X_k | X_{k+1} | \dots | X_{k+j}] + H_i \cdot [\tilde{U}_k | \tilde{U}_{k+1} | \dots | \tilde{U}_{k+j}] \end{aligned}$$

$$\text{Où : } \overline{X}_k = [X_k | X_{k+1} | \dots | X_{k+j}]$$

$$U_0 = [\tilde{U}_k | \tilde{U}_{k+1} | \dots | \tilde{U}_{k+j}]$$

$$\text{Donc : } \overline{Y}_k = C \cdot A^i \cdot \overline{X}_k + H_i \cdot U_0$$

$$U_0 = \begin{bmatrix} U_k & \dots & U_{k+j} \\ \vdots & \ddots & \vdots \\ U_{k+i} & \dots & U_{k+j+i} \end{bmatrix}, \text{ matrice de Hankel d'entrée}$$

On a alors

$$\begin{bmatrix} \overline{Y}_k \\ \overline{Y}_{k+1} \\ \vdots \\ \overline{Y}_{k+i} \end{bmatrix} = \begin{bmatrix} C \cdot \overline{X}_k \\ C \cdot A \cdot \overline{X}_k \\ \vdots \\ C \cdot A^i \cdot \overline{X}_k \end{bmatrix} + \begin{bmatrix} H_0 \cdot U_0 \\ H_1 \cdot U_0 \\ \vdots \\ H_i \cdot U_0 \end{bmatrix}$$

$$\overline{Y}_k = \begin{bmatrix} C \\ C \cdot A \\ \vdots \\ C \cdot A^i \end{bmatrix} \cdot \overline{X}_k + \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_i \end{bmatrix} \cdot U_0$$

$$\text{Où : } \begin{bmatrix} C \\ C \cdot A \\ \vdots \\ C \cdot A^i \end{bmatrix} = \text{matrice d'observabilité}$$

$$\overline{Y}_k = \Gamma_i \cdot \overline{X}_k + \begin{bmatrix} D & \dots & \dots & 0 \\ CB & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ C \cdot A^{i-1} \cdot B & C \cdot A^{i-2} \cdot B & \dots & D \end{bmatrix}$$

$$\text{Où } H = \begin{bmatrix} D & \dots & \dots & 0 \\ CB & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ C \cdot A^{i-1} \cdot B & C \cdot A^{i-2} \cdot B & \dots & D \end{bmatrix}, \text{ matrice toeplitz inférieure du système}$$

$\overline{Y}_k = \Gamma_i \cdot \overline{X}_k + H \cdot U_0 + \tilde{H} \cdot W_0$, où \tilde{H} est la matrice de toeplitz inférieure des bruits

$$\begin{aligned}\Pi_M &= I - M^T \cdot (M \cdot M^T)^{-1} \cdot M \\ M \cdot \Pi_M &= M \cdot (I - M^T \cdot (M \cdot M^T)^{-1} \cdot M) \\ &= M - M \cdot M^T \cdot (M \cdot M^T)^{-1} \cdot M \\ &= M - M = 0\end{aligned}$$

$$Y_k \cdot \Pi_{U_0} = (\Gamma_i \cdot \overline{X_k} + H \cdot U_0 + \tilde{H} \cdot W_0) \cdot \Pi_{U_0}$$

Les bruits ne sont pas corrélés

$$Y_k \cdot \Pi_{U_0} = \Gamma_i \cdot \overline{X_k} \cdot \Pi_{U_0} + \tilde{H} \cdot W_0 \cdot \Pi_{U_0}$$

V1 et V2 sont indépendants donc $E[V_1 \cdot V_2^T] = 0$

$$W_1 \cdot Y_k \cdot \Pi_{U_0} \cdot W_2 = W_1 \cdot \Gamma_i \cdot \overline{X_k} \cdot \Pi_{U_0} \cdot W_2$$

Pour améliorer $\Gamma_i \cdot \overline{X_k}$ obtenue, deux matrices de pondération de pondération W_1 et W_2 sont utilisées

L'ordre du système est $\text{rang}(W_1 \cdot Y_k \cdot \Pi_{U_0} \cdot W_2)$

$$\begin{aligned}\text{Où } W_1 \cdot Y_k \cdot \Pi_{U_0} \cdot W_2 &= [U_1 \quad U_2] \cdot \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \cdot \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \simeq W_1 \cdot \Gamma \cdot \widetilde{X}_2 \cdot W_2 \\ &\simeq U_1 \cdot S_1 \cdot V_1^T \simeq W_1 \cdot \Gamma \cdot \widetilde{X}_k \cdot \Pi_{U_0} \cdot W_2\end{aligned}$$

$$\text{Où } W_1 \cdot \Gamma = U_1 \cdot S_1^{1/2} \text{ Et } \widetilde{X}_k \cdot \Pi_{U_0} \cdot W_2 = S_1^{1/2} \cdot V_1^T$$

$$\text{Et } Z = U \cdot S \cdot U^T, \text{ avec } I = U \cdot U^T \text{ et } S \text{ diagonale}$$

On peut estimer alors les matrices A et C à partir de la matrice d'observabilité Γ_i , B et D à partir de la matrice de Toeplitz U_0

$$\begin{aligned}\widetilde{X_{i+1}} &= A \cdot \widetilde{X_i} + B \cdot U_i \\ Y_i = C \cdot \widetilde{X_i} + D \cdot U_i &\Leftrightarrow \begin{bmatrix} \widetilde{X_{i+1}} \\ Y_i \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} + \begin{bmatrix} \widetilde{X_i} \\ U_i \end{bmatrix}\end{aligned}$$

Partie 3:

Identification et validation du modèle

1-Identification et validation du modèle hors-ligne :

Après avoir identifié le modèle avec la méthode n4sid, nous procédons alors à la validation de ce modèle. Pour cela un programme MATLAB a été mis au point pour pouvoir valider.

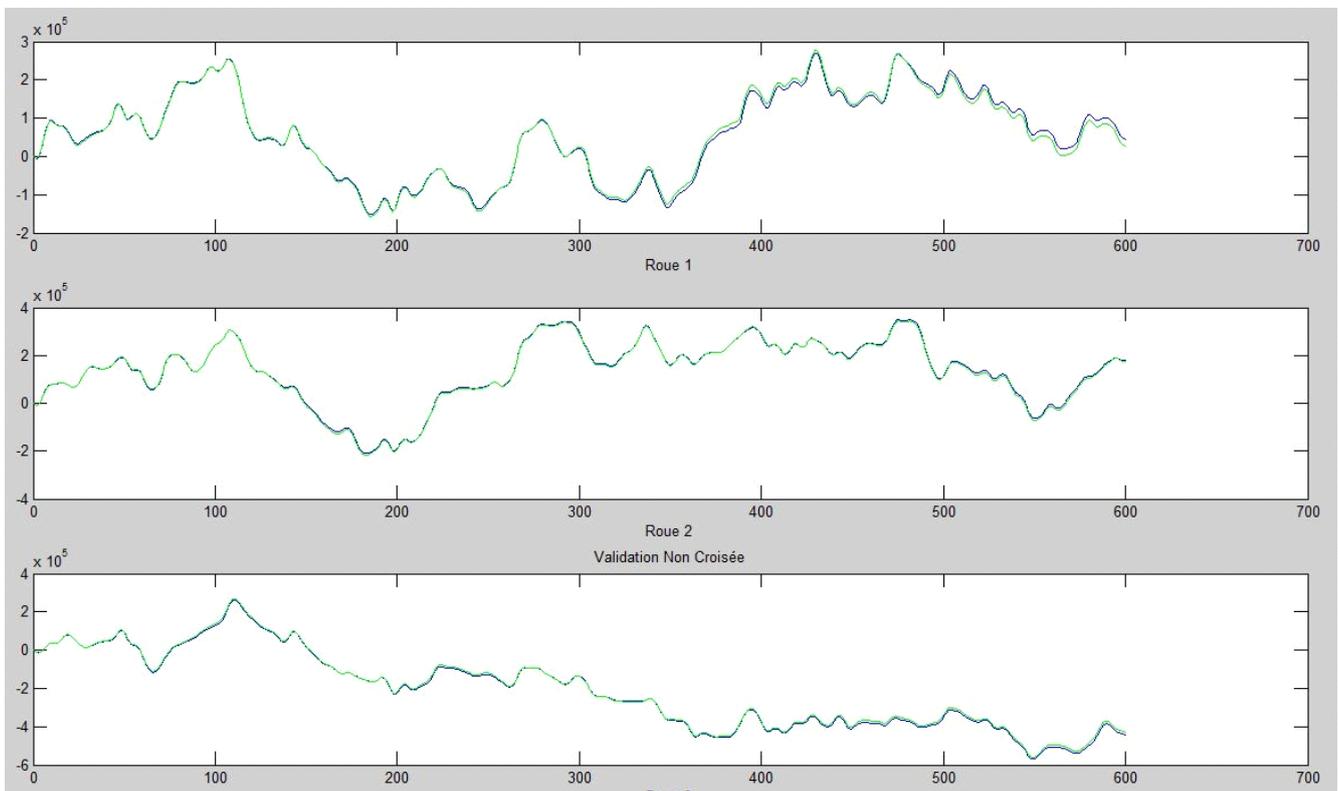
Dans ce programme on définit le nombre de données sur lequel on va travailler ou le temps et on déclare le rang ou l'ordre du système à 16 car on se base sur la modélisation du Robotino en repère mobile avec en sortie les positions des 3 roues.

On charge après ces données recueillies à l'aide de l'acquisition des compagnes faite avec le robotino dans le programme.

On déclare aussi la taille de la matrice d'entrée U et on fixe la période d'échantillonnage à 1/3.

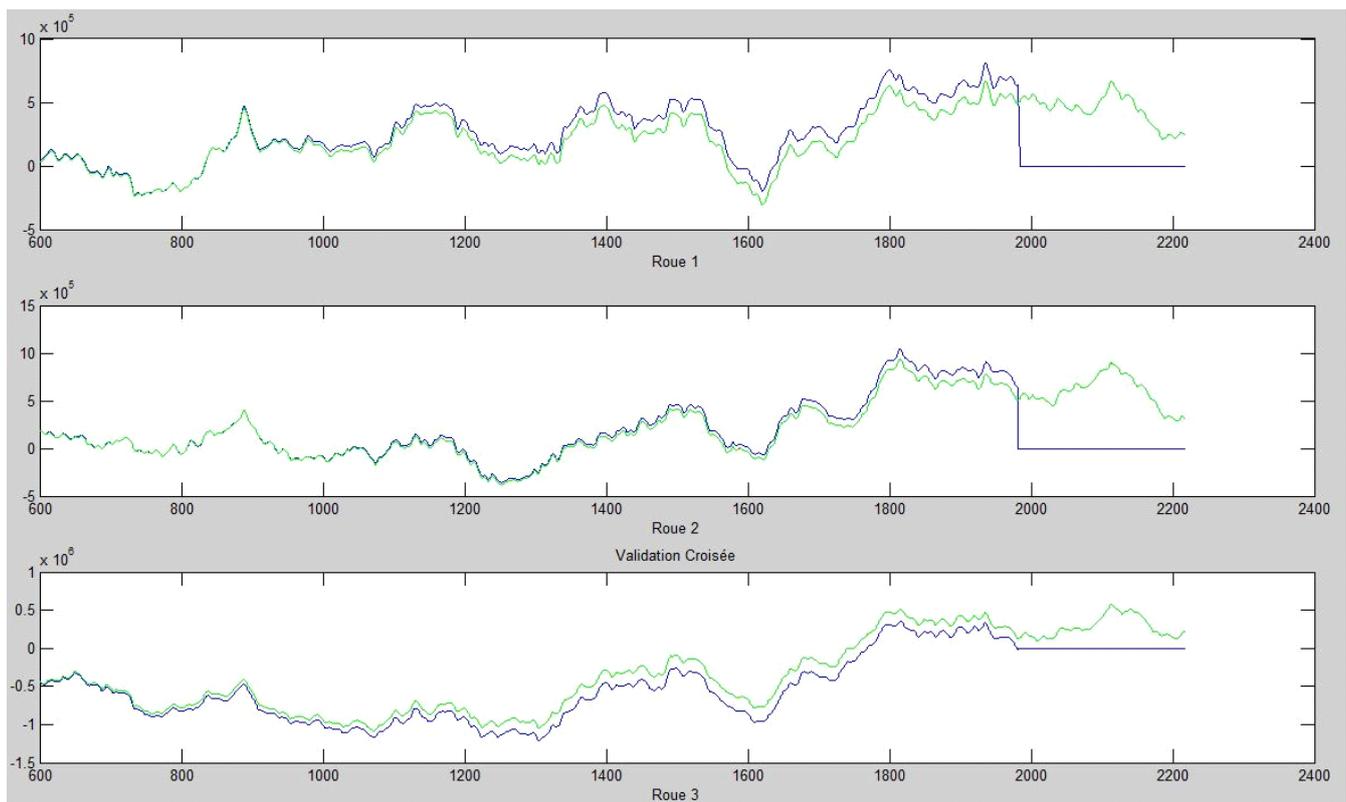
On tire le modèle avec le tiers des données grâce à la méthode n4sid expliqué dans la deuxième partie du rapport, mais qui existe déjà dans les fonctions proposées par MATLAB ce qui nous évite de perdre beaucoup de temps à mettre en place un programme prenant en compte toutes les étapes de la partie théorique.

On fait aussi appel à un autre programme qui a pour but la comparaison du modèle obtenu et de la sortie Y du système qui représente la position des trois roues du Robotino par rapport à son axe mobile lié à lui sur le premier tiers des points utilisés, ce qu'on appelle la validation non croisée : (l'axes des abscisses : le temps, l'axe des ordonnées : grandeur mesurée (angle : position des roues)).



On constate que sur ce graphe les 3 roues du Robotino suivent le modèle obtenu et qu'il y a aussi des erreurs pas importantes vers la fin.

Sur ce même programme utilisé pour la comparaison, on essaye maintenant de voir le comportement des roues par rapport au modèle mais cette fois avec le reste des données qui n'ont pas servi à faire le modèle, ce qu'on appelle la validation croisée : (l'axe des abscisses : le temps, l'axe des ordonnées : grandeur mesurée (angle : position des roues))



On remarque qu'au bout d'un certain temps les positions des roues présentent des comportements qui divergent du modèle, ce qui sera optimisé dans la partie suivante où on essaiera d'annuler les écarts grâce aux observateurs mais cette fois en mode en ligne.

Après avoir vérifié et validé le modèle, il ne faut pas oublier de convertir le système de discret en continu, et de sauvegarder le modèle notamment les matrices A, B, C et D.

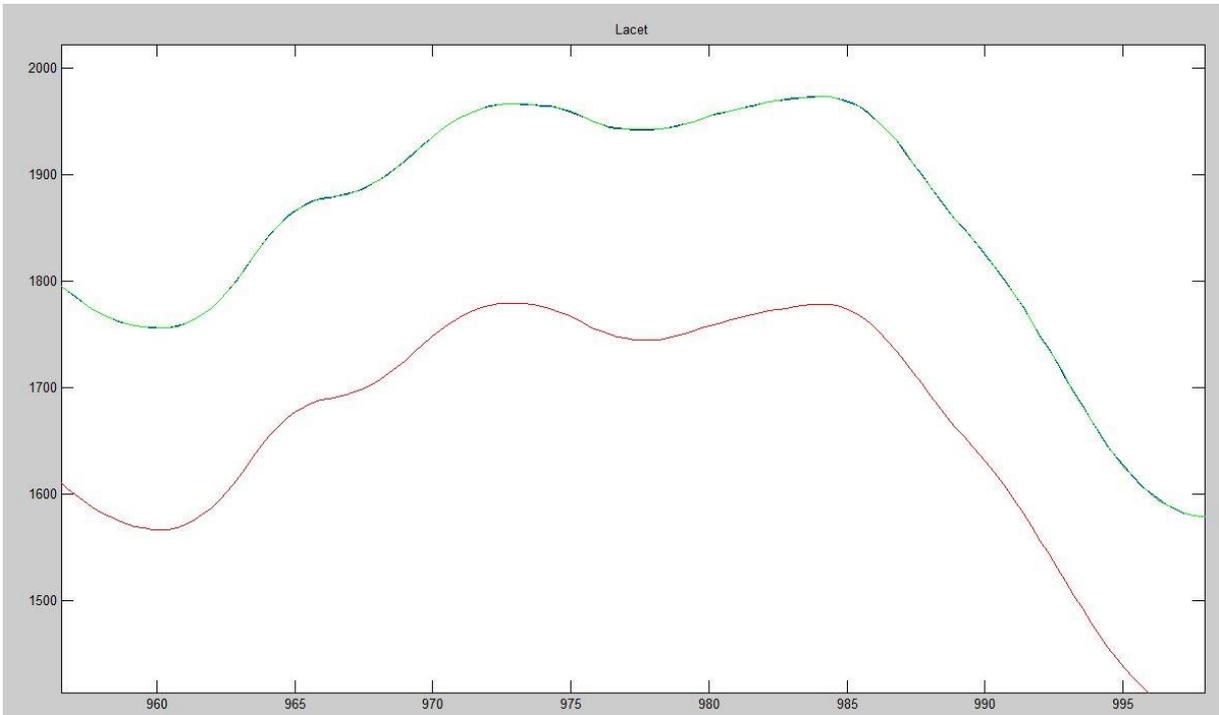
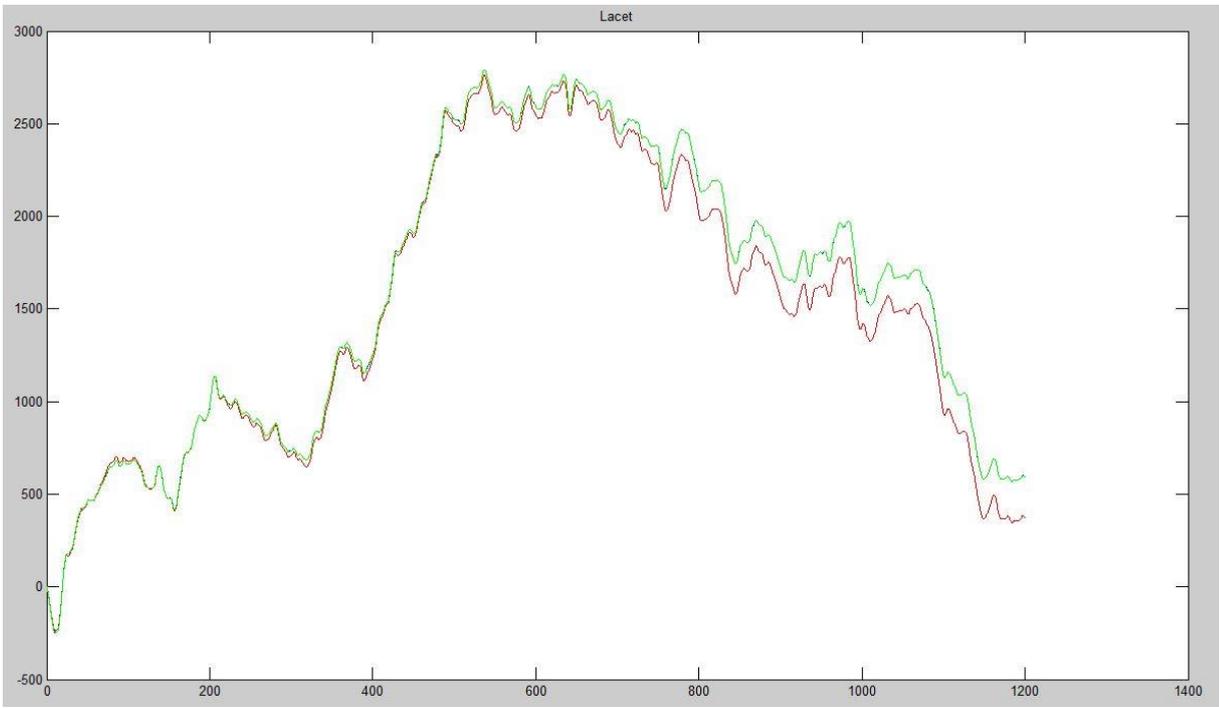
2-Validation du modèle en ligne :

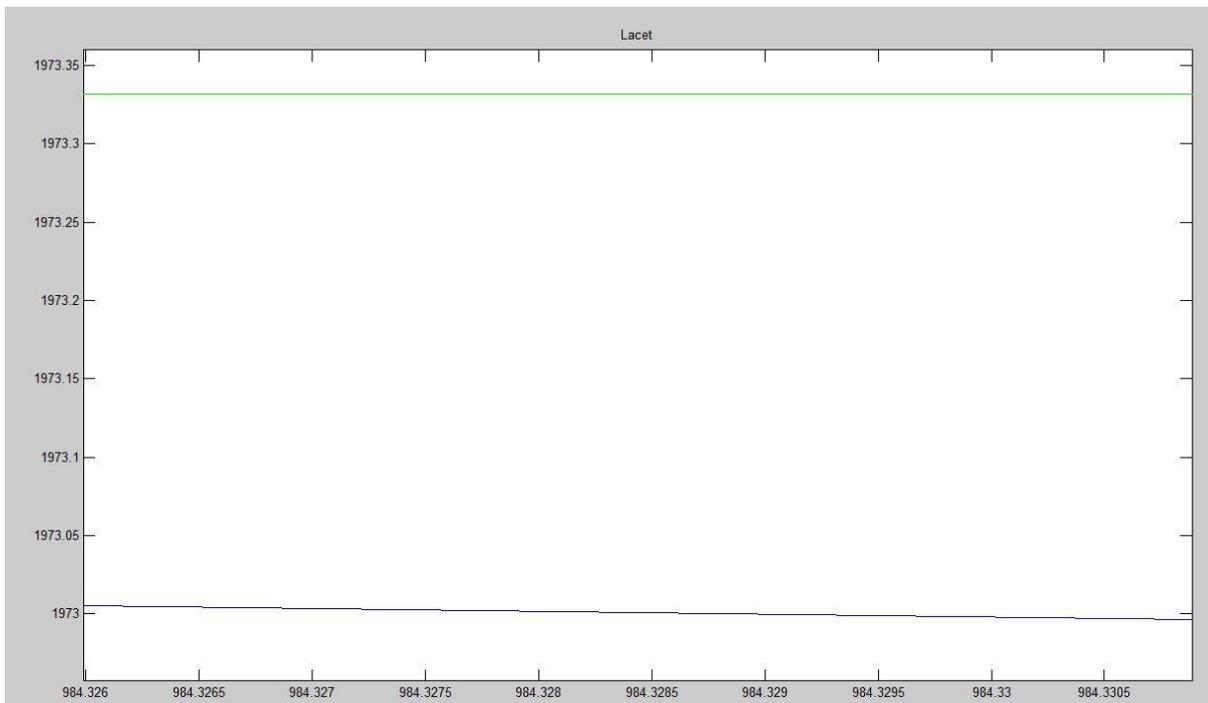
Nous devons à présent faire la validation en ligne du modèle du Robotino avec en sortie le lacet. Nous avons décidé d'ajouter un observateur d'état à notre modèle. Un observateur d'état est l'extension d'un modèle représenté. L'observateur a pour but de reconstruire l'état lorsque que celui-ci n'est plus mesurable à l'aide des mesures d'autres grandeurs. Tout d'abord, nous devons vérifier que notre matrice A soit commandable et observable. La commandabilité et l'observabilité sont deux caractéristiques d'une représentation d'un système. La commandabilité caractérise la possibilité que la commande exerce une influence sur un des états. L'observabilité indique la capacité pour un système à déterminer l'historique d'un état seulement à partir de la connaissance des variables de sorties mesurées. Pour la commandabilité, nous avons besoin des matrices A et B. Ici n=16, c'est la taille de la matrice A. $Com = [B \ A * B \ A^2 * B \ \dots \ A^{n-1} * B]$. Si le rang de Com est égal n alors le modèle est commandable. Ce qui est notre cas. Pour l'observabilité, nous avons besoin des matrices A et C. $Obs = [C ; C * A ; C * A^2 ; \dots ; C * A^{n-1}]$. Si le rang d'Obs est égal à n alors le modèle est observable. Ce qui est notre cas ici. Nous pouvons donc réaliser notre observateur d'état K. L'observateur d'état devrait logiquement permettre à notre modèle de toujours coller avec les données réelles et cela même sur une durée de temps très longue. L'écriture matricielle du modèle change et on a présent : où P la position réelle des roues

$$X' = A.X + B.U + K.P$$

- Modèle du Robotino pour le lacet

Notre encadrant nous a fourni le modèle du Robotino pour le lacet qui a été identifié et validé hors-ligne. Nous chargeons ce modèle dans notre script MATLAB. Nous chargeons également un fichier de commande. Celui-ci correspond à la commande U caractérisant la consigne en vitesse sur chaque roue. C'est un tableau à 3 colonnes (pour les 3 entrées) et un très grand nombre de lignes qui correspond à la commande pendant le laps de temps. Cette commande est celle qui est donné au robot et celle donné au modèle. Nous pouvons donc recueillir la position du lacet réel du Robot et celle estimée avec notre modèle car ils ont tous les deux la même commande. Nous avons décidé de réaliser la validation de notre modèle dans l'arène et non en roues libres comme pour la validation hors-ligne afin de voir quel impact cela aura sur le modèle. Nous avons donc réalisé une campagne d'acquisition sur une durée de 20 min dans laquelle nous comparons la position du lacet en temps réel en bleu, la position du lacet estimée avec le modèle sans observateur en rouge et la position du lacet estimée avec le modèle avec observateur en vert.

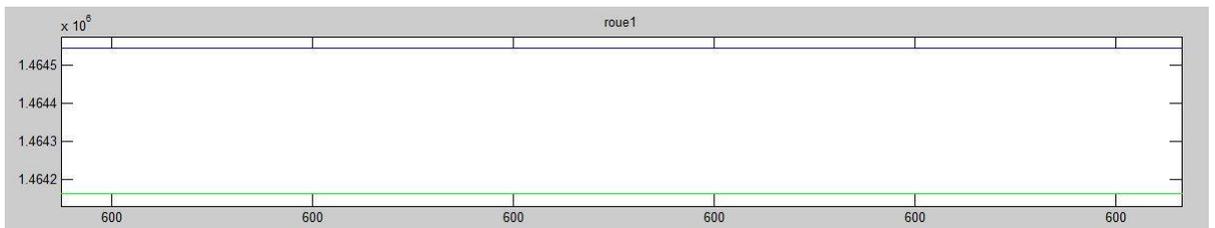
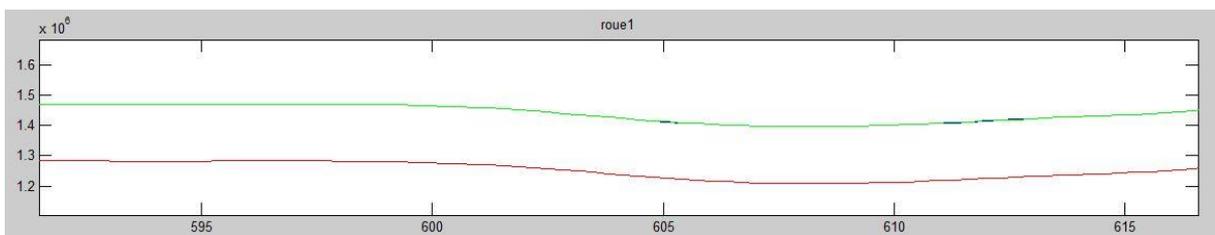
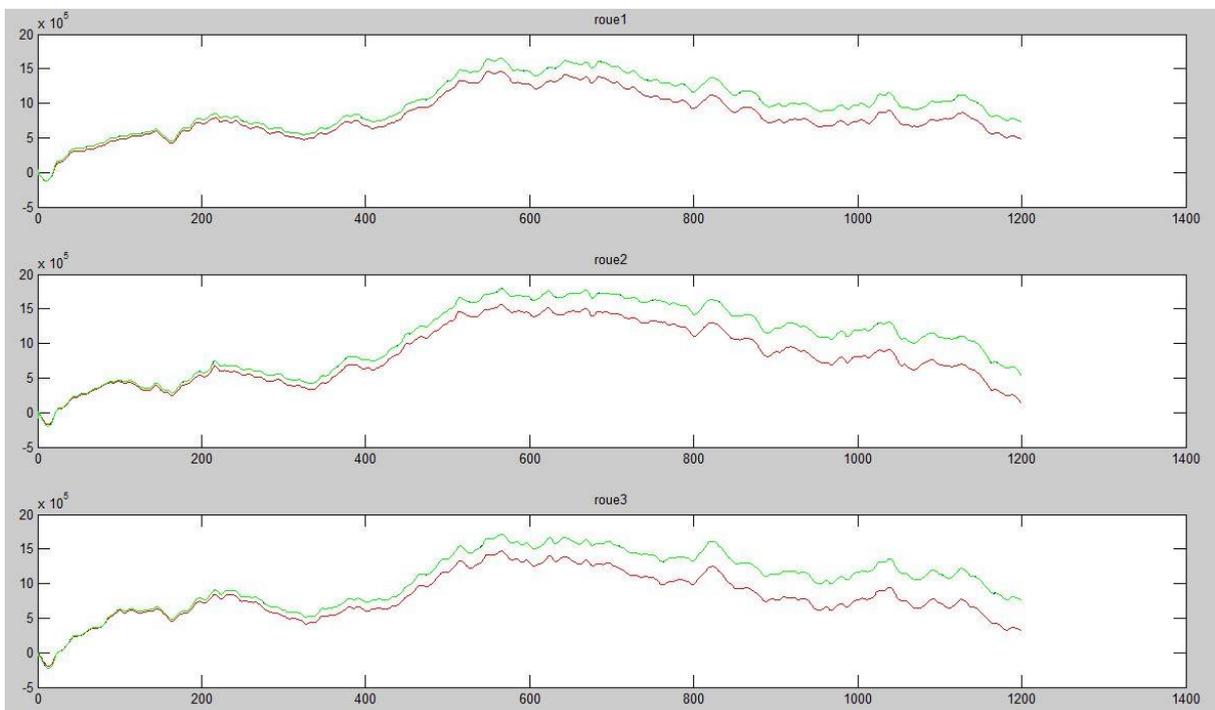




Comme nous le voyons sur les figures ci-dessus, l'écart entre la variation réelle du lacet et celle estimée avec le modèle seul est faible jusqu'à environ 50% de notre campagne d'acquisition. Notre modèle est donc validé. Quant à l'écart entre la variation réelle et celle estimée par la modèle avec observateur, il est de l'ordre du dixième tout le long de la campagne d'acquisition. L'écart est tellement faible que nos courbes se chevauchent. Cela montre que notre observateur fonctionne très bien mais lorsque nous sommes dans l'arène. On peut donc conclure que le modèle du Robotino pour le lacet a été validé en ligne.

- Modèle du Robotino pour la position des roues

Nous réalisons exactement les mêmes étapes que précédemment, sauf que l'on charge le modèle que nous avons identifié avec pour sortie la position des roues. Nous réalisons une campagne d'acquisition sur une durée de 20 min dans laquelle nous comparons la position de chaque roue en temps réel en bleu, la position de chaque roue estimée avec le modèle sans observateur en rouge et la position de chaque roue estimée avec le modèle avec observateur en vert.



Nous arrivons à la même conclusion que pour le modèle lacet. Ici l'écart entre le comportement réel et celui estimé par le modèle avec observateur est pratiquement nulle. Ce qui prouve que notre observateur d'état fonctionne très bien. Avec ce modèle, le comportement estimé diverge plus rapidement que dans le cas précédent. Cette divergence est due aux forces de frottements. Nous aurions pu estimer ces forces de frottements en ajoutant à notre équation matricielle un observateur à entrée inconnue qui aurait caractérisé ces forces.

CONCLUSION

Ce projet est très enrichissant, car non seulement on s'investit dans le travail durant les séances et en dehors des séances de projet, mais il nous ouvre les yeux sur d'autres études qui se font dans le domaine de la robotique.

Le fait de se cultiver en apprenant des documents qui traitent la modélisation ou l'identification nous a fait développer nos connaissances, notre manière de travailler et notre savoir-faire.

Ce que nous avons réussi à faire peut encore être optimisé, en estimant les forces de frottements à l'aide d'un observateur à entrée inconnue qui n'est pas dans notre programme.

Grâce aux remarques et aux conseils pertinents de notre encadrant Mr. PEKPE on a su répondre au cahier des charges qu'on avait défini avant le début des séances des projets.