

Modélisation AxNum avec Simulink®

Savoirs et compétences développés :

- Analyser
 - Analyse fonctionnelle
 - Caractériser les écarts et interpréter.
- Modéliser
 - Proposer un modèle de comportement adapté pour un objectif.
- Résoudre en utilisant une démarche numérique.
- Expérimenter : observer un protocole expérimental pour valider un modèle.

Introduction

Objectif du TP

Dans ce TP nous allons **modéliser un axe de positionnement robotique**. Pour **résoudre l'évolution temporelle** du modèle le logiciel de **simulation** par schémas blocs Simulink sera utilisé. Ce TP permet de préparer le premier projet de modélisation et simulation de systèmes.

Système

Nous étudierons l'axe numérique Didalab. Ce système permet de positionner ou de réguler la vitesse d'un chariot motorisé par une machine à courant continu. Il pourrait par exemple servir à positionner la tête d'impression d'une imprimante 3D, la broche d'une machine-outil (avec un moteur plus puissant) ou encore la tête d'une imprimante grandes dimensions. Cet axe permet de faire des expériences dans un environnement de prototypage contrôlé et de valider des choix techniques avant de réaliser une machine plus complète. Idéalement cet axe doit être rapide et précis, des critères quantifiés étant disponible dans le cahier des charges de la machine que l'on souhaite réaliser.

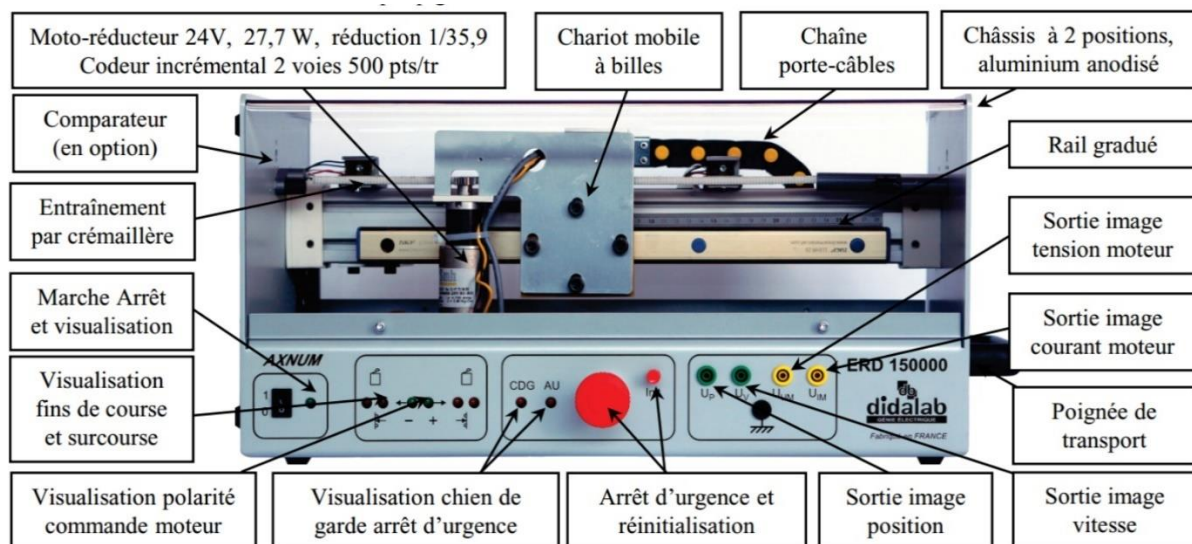


Figure 1 Axe numérique, composants

Une alimentation 24V 2.9A est utilisée.

Présentation de Matlab® Simulink®

Matlab® est un langage de programmation dédié aux ingénieurs et aux scientifiques. Il dispose de nombreuses similarités avec Python : Il ne se compile pas et le typage est faible : par exemple il n'est pas nécessaire de déclarer le nom ou le type (booléen, entier, flottant...) des variables avant de les utiliser. L'interpréteur se charge de « deviner » les intentions en fonction du type de calcul. Matlab est un logiciel propriétaire. Il dispose d'un environnement de programmation intégré : éditeur de texte, interpréteur bibliothèques et éléments de visualisation sont dans un même package.

Matlab intègre aussi de nombreux outils dédiés aux problèmes scientifiques et d'ingénierie que l'on ne retrouve pas dans Python. Par exemple Simulink est un outil permettant de mettre en place des modèles par schéma blocs avec une interface graphique et un solveur intégré. Ainsi l'utilisateur dispose les blocs dans une fenêtre, les relie, définit les équations de leurs fonctions de transfert, les entrées et les sorties souhaitées. Ensuite il peut résoudre et nous permettre d'observer les résultats de la modélisation. Simulink est très largement utilisé dans l'industrie (aérospatiale, environnement, défense etc...). Par rapport à une résolution analytique l'ordinateur permet de traiter des systèmes plus complexes et éventuellement de s'affranchir des hypothèses linéarité/continuité/invariance indispensable dans une étude analytique. Simulink permet aussi de s'interfacer avec du matériel (carte de mesure, système physique) ou d'autre logiciel (Solidworks, meca3D).

Objectif de la modélisation

On souhaite simuler le fonctionnement du système lorsqu'il est asservi en vitesse. Cette étude préalable nous permettra d'acquérir et valider sur une expérience nos compétences de modélisation d'un axe asservi. En particulier elle nous permettra d'observer certaines difficultés de modélisation inhérente à ce type de système comme les **effets de seuil** provenant des **frottements secs** ou les effets de **saturation** provenant des limitations des composants. Ces compétences pourront alors nous servir à modéliser des systèmes plus complexes.

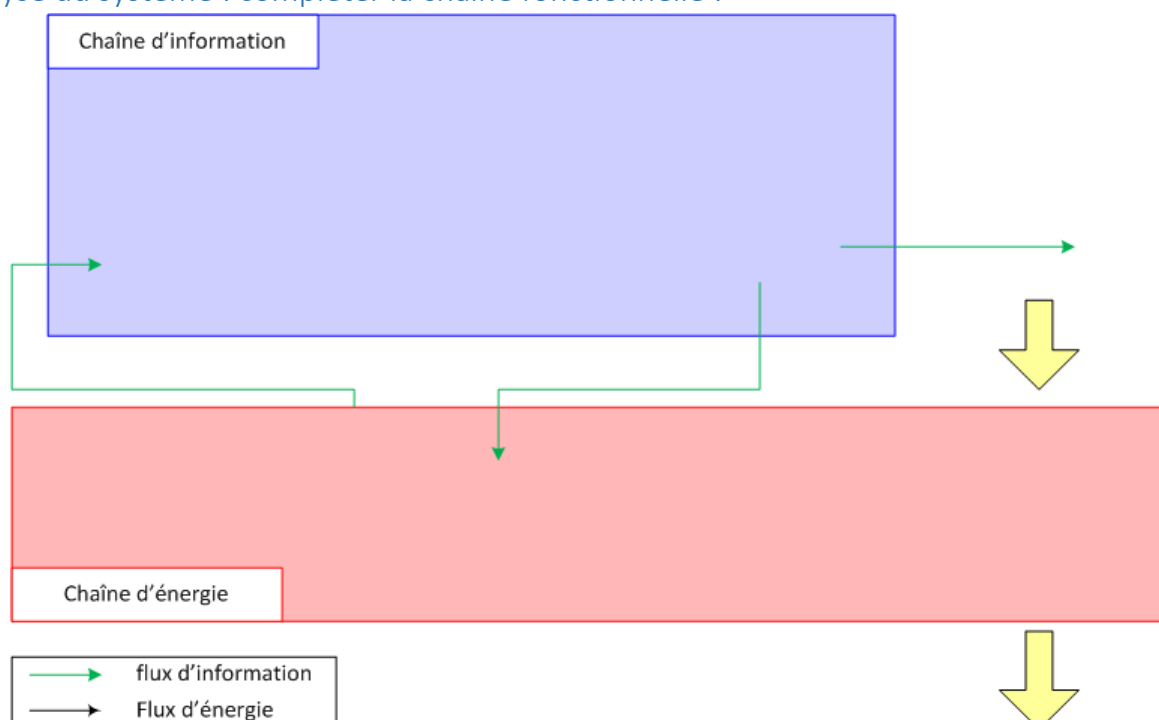
Une première étape consistera à modéliser le système en boucle ouverte (système commandé) afin de valider une section du modèle asservi complet.

Déroulement du TP :

1. Analyse de la chaîne fonctionnelle.
2. Mise en place d'un **modèle de connaissance** avec schéma bloc, axe en boucle ouverte.
3. Schéma bloc pour la boucle fermée.
4. Prise en main de Simulink
5. Modélisation de l'axe numérique en **boucle ouverte** et interface de **courant**.
6. Modélisation de l'axe numérique en **boucle fermée de vitesse** et interface de **courant**.
7. Modélisation de l'axe numérique en **boucle fermée de position** et interface de **courant**.
8. Modélisation de l'axe numérique en **boucle fermée de position avec retour tachymétrique** et interface de **courant**.
9. Modélisation de l'axe numérique en **boucle ouverte** et interface de **tension**.

Préparation

Q1 Analyse du système : compléter la chaîne fonctionnelle :



Q2 Schéma bloc du système Boucle ouverte et Interface de courant.

Ou souhaite étudier l'évolution temporelle de la vitesse du chariot (dans le cadre de la future étude en mode asservi). En boucle ouverte le logiciel permet de contrôler le courant d'alimentation. Identifier l'entrée et la sortie du modèle recherché (**indispensable** pour identifier le contenu des blocs).

Entrée :

Sortie :

Placer les éléments suivants : Couple réducteur **Cr**, Force motrice **F**, Courant commande **Im**, Couple moteur **Cm**, Accélération **Mx''**, et Vitesse **Mx'**.



Les gains ou fonctions des blocs sont à spécifier. Une équation du mouvement sera nécessaire.

- Documentation du motoréducteur en annexe 1.
- Diamètre primitif d'un pignon : $D = m Z$ avec m le module et Z le nombre de dents.
- Masse équivalente du chariot : 120 Kg (masse chariot et inertie motoréducteur combinées).

Q3 Schéma bloc du système Boucle fermée vitesse et Interface de courant.

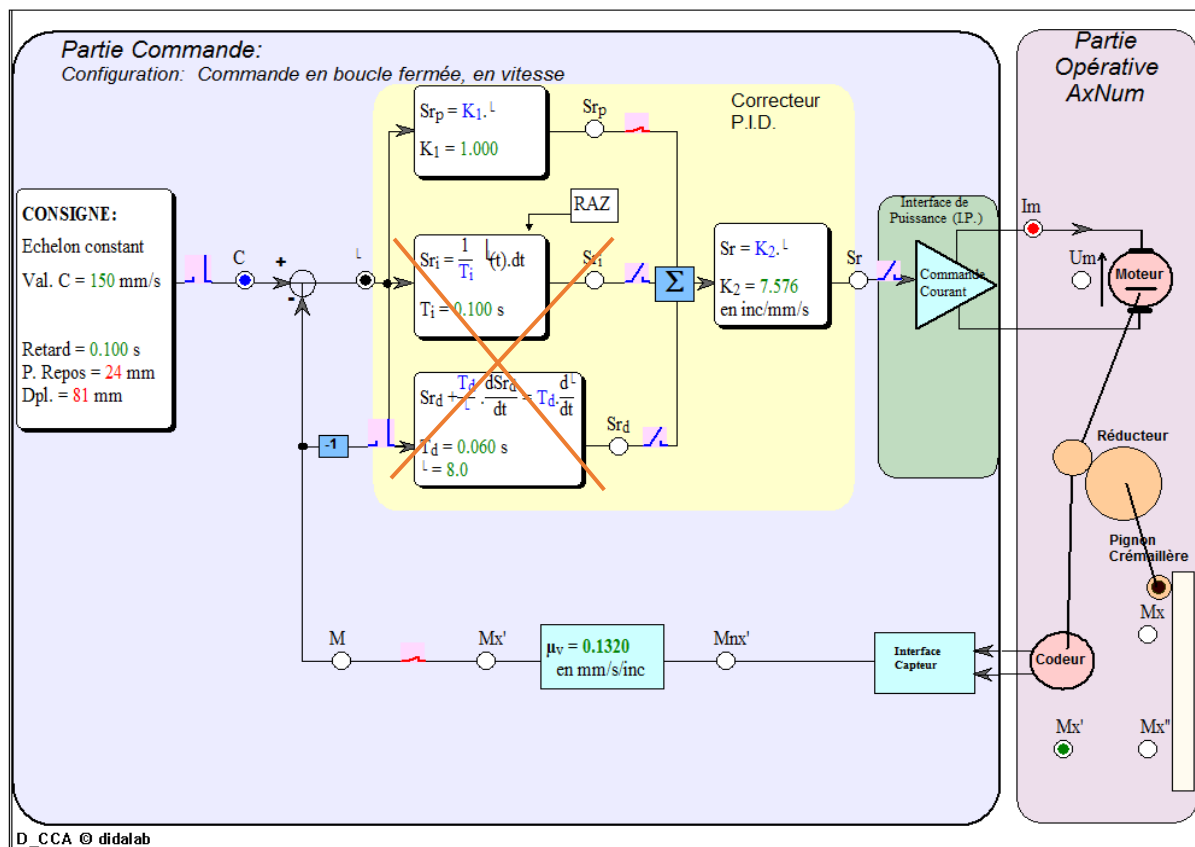


Figure 2 Boucle fermée vue logiciel

Identifier l'entrée et la sortie du modèle. Réutiliser le travail sur la boucle ouverte pour la partie opérative (chaîne d'énergie). S'inspirer du cours et de la vue du logiciel pour modéliser la correction. Pour le capteur de vitesse on le considérera idéal et un retour unitaire de la vitesse sera utilisé. Une conversion m/s mm/s restera nécessaire avant le comparateur.

Schéma bloc en boucle fermée :

Q4 Expérimentation avec Simulink

- Créer un répertoire SimAxnum dans un répertoire perso sauvegardé. Lancer MATLAB (dernière version). Placer l'adresse du répertoire dans la barre d'adresse de Matlab.
- Tester le mode calculatrice en tapant un calcul dans la barre d'exécution en bas de l'écran.
- Faire click droit dans la fenêtre de gauche « New Script » pour créer un fichier de commandes Matlab dans le répertoire de travail. Ce fichier permettra de faire des calculs, des programmes et de définir des variables ce qui sera utile pour Simulink. Par exemple taper « $A=3+\pi$ » dans le script et l'exécuter (onglet lecture). La variable A apparaît dans le « Workspace ».
- Désormais presser le bouton Simulink puis choisir un « Blank model » pour créer votre premier schéma bloc à simuler. En pressant le bouton de couleur on peut choisir ses blocs. Dans les rubriques « commonly » (blocs courants) « source » (entrées) « sinks » (sorties) « continuous » (intégrale, dérivé, fonction de transfert) et « Math » (polynômes, trigonométrie, seuils) vous trouverez l'essentiel des blocs de base dont vous aurez besoin.
- Glisser un signal échelon « source » dans le nouveau modèle et paramétrez le à la valeur finale 3, ajoutez un bloc gain dont vous donnez la valeur 7 et un bloc Scope « sink ». Reliez-les et exécutez (bouton lecture). Ouvrir le scope et commenter le résultat.
- Remplacer la valeur 3 par la variable A et observer le nouveau résultat. En modifiant la source faite démarrer le « Step » dès $t=0.1s$.
- Taper $B=2$ dans le script, le ré-exécuter, remplacer A par B, et ajouter un bloc intégrateur « Continuous/Integrator » dans la boucle. Commenter.

Q5 Résolution du schéma bloc en boucle ouverte.

Placer dans un nouveau modèle les gains et blocs nécessaires pour le modèle en boucle ouverte.

Ajouter une entrée échelon d'amplitude 200mA avec un retard de 0.1s. Limiter la durée du calcul à 2s (à côté du bouton play). En sortie placer un scope et un bloc «sink/to workspace » pour observer la vitesse afin d'enregistrer les résultats et les comparer à ceux mesurer sur la machine. Nommer la variable vitesse mxp dans le bloc. Pour effectuer la comparaison récupérer les fichiers du répertoire Axnum sur le réseau et les placer dans votre répertoire SimAxnum. Après avoir lancé la simulation la variable mxp apparaît dans le workspace. On peut alors exécuter le script compareBO pour afficher la courbe de simulation et la courbe expérimentale (Figure 3). Les résultats expérimentaux sont aussi disponibles dans la figure « BOC200mA ». Commenter les écarts : ...

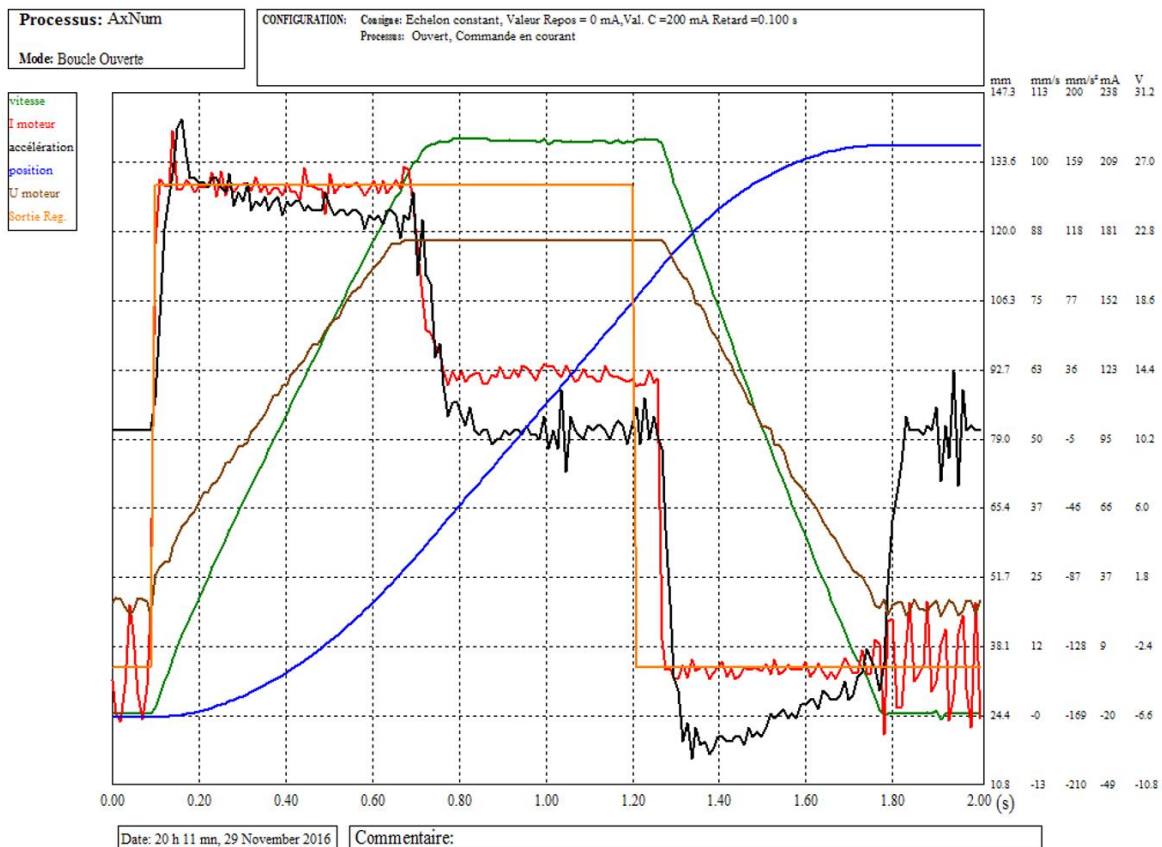


Figure 3 Résultats expérimentaux boucle ouverte

Q6 Amélioration du modèle boucle ouverte

Le modèle accélère trop fort car les frottements sont négligés. On constate que 50mA sont nécessaire pour mettre l'Axnum en mouvement (effet de frottement sec). Placer un bloc « Discontinuités/Deadzone » avant le moteur pour prendre en compte cet effet. Exécuter de nouveau la comparaison et observer l'amélioration.

Dans un second temps ajouter des frottements visqueux. Pour cela on ajoutera une force proportionnelle à la vitesse $F_v = f \cdot v$. Exprimer la nouvelle équation du mouvement. A l'aide d'un gain et d'un sommateur (commonly used block) mettre en place le frottement visqueux. $f = 1$. Donner l'unité de f , observer le résultat :

...

Au bout de 0.7s le système s'arrête brutalement d'accélérer (vitesse constante) contrairement au modèle. En observant la mesure de tension proposer une explication à cette limitation. On pourra aussi raisonner sur le comportement sur un temps long du modèle comparé à la mesure et sa pertinence.

...

Q7 Modélisation asservissement de vitesse, commande courant

Résultat expérience : Figure 4 et « BF100mmsVI ». Copier le modèle précédent que l'on va réutiliser pour modéliser le système physique en boucle fermée. Ajouter les éléments manquant et la sollicitation (cf Q3, Figure 2). Lancer le script compareBF pour comparer les résultats de la simulation avec l'expérience. Commenter sur la qualité de l'asservissement puis de la simulation.

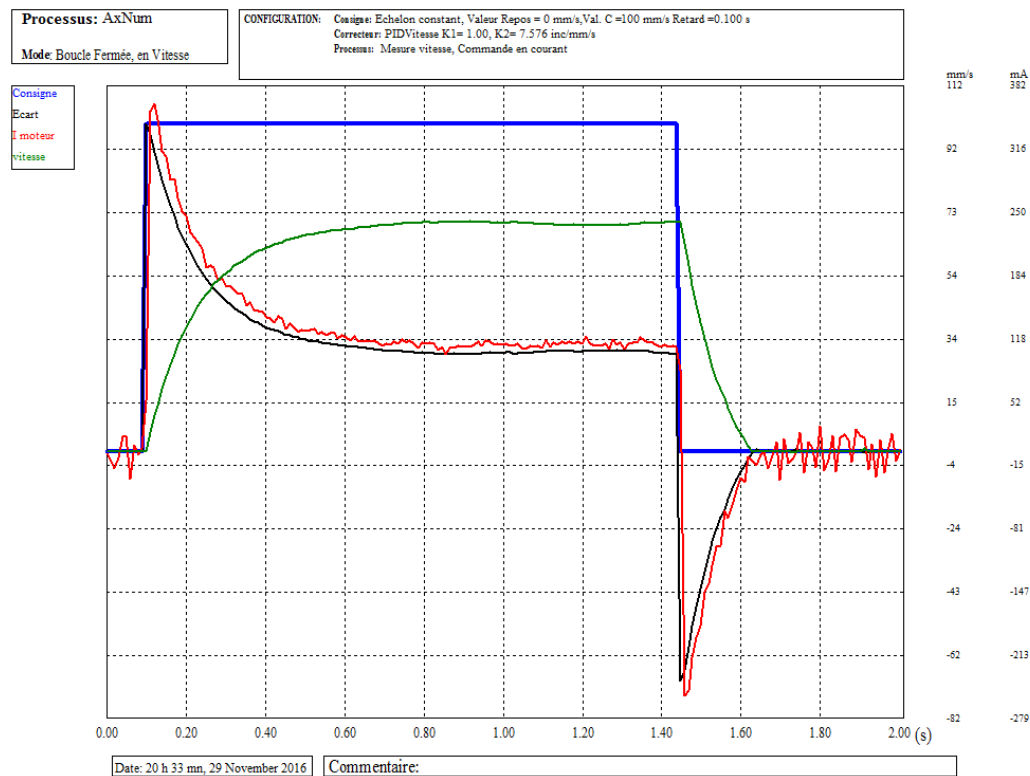


Figure 4 Résultats boucle fermée vitesse

Q8 Amélioration des performances de l'asservissement.

Pour améliorer les performances on double la valeur de la correction proportionnelle : $K1=2$. Résultats :

Q9 Aller plus loin : Boucle fermée de position

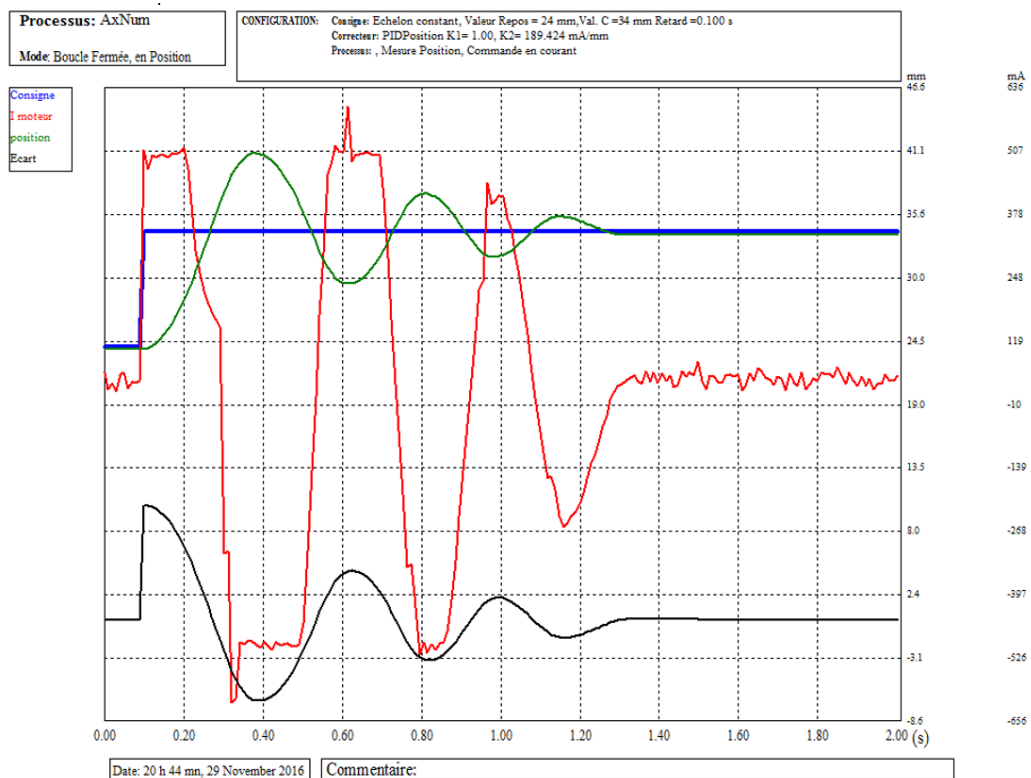
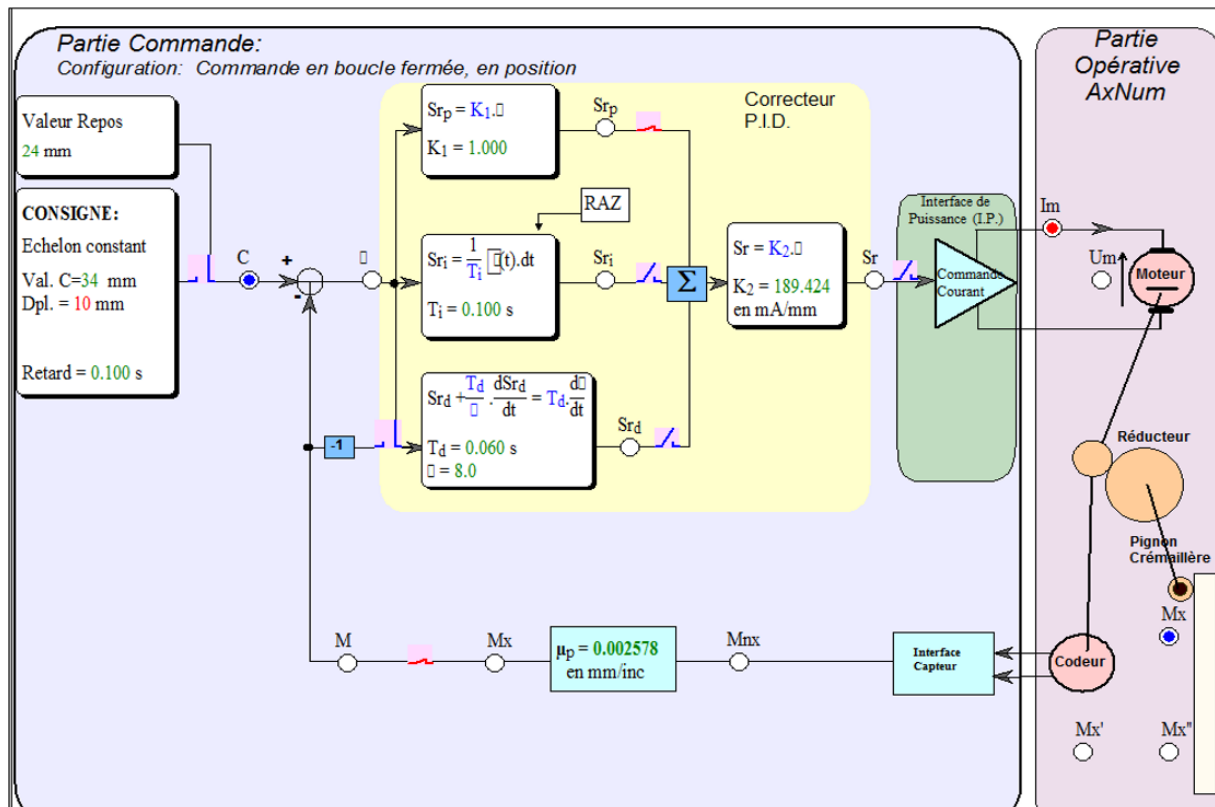


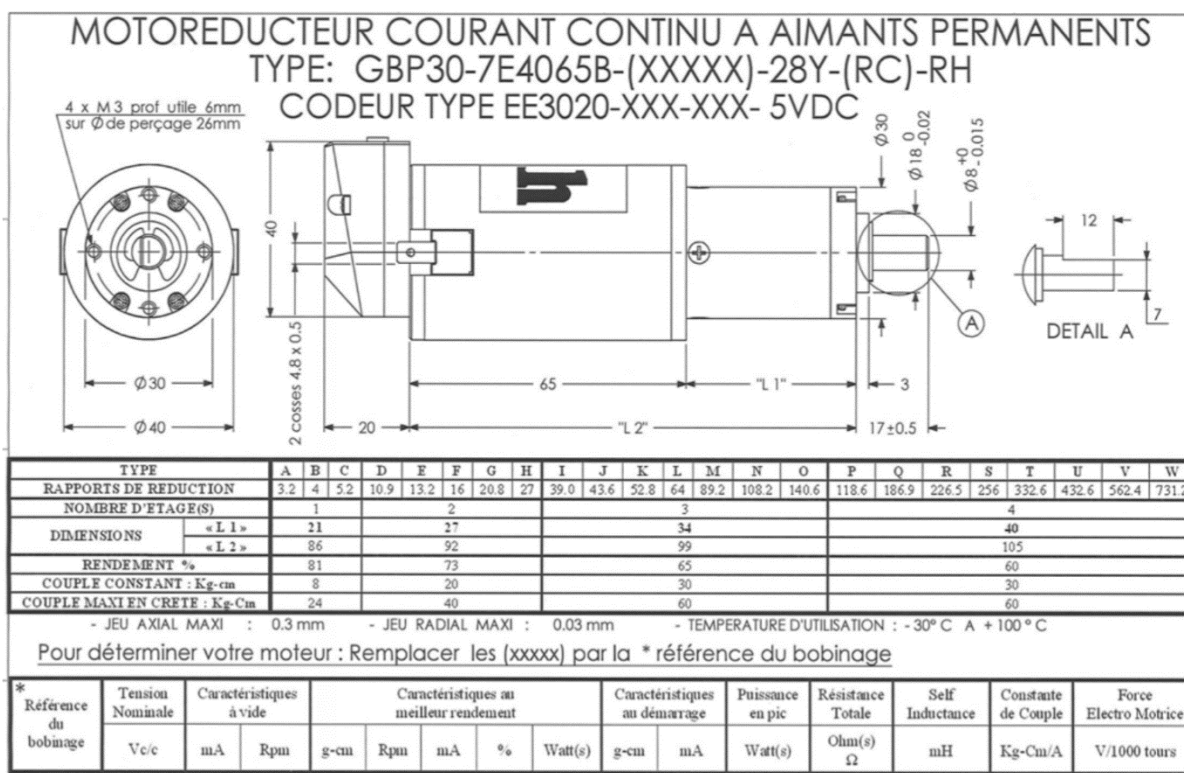
Figure 5 Boucle fermée de position

Commenter la qualité de l'asservissement de position. Le simuler.

Annexe 1 documentation Motoréducteur/pignon SMH

CARACTERISTIQUES	Valeur	Unités
Tension d'alimentation	24	Vdc
Vitesse moteur à vide	5 200	Tr/min
Courant au meilleur rendement	911	mA
Constante de couple	4.32	N-Cm/A
Rendement moteur maximum	72.34	%
Puissance utile au meilleur rendement	12.94	W
Rapport de réduction du réducteur	1/35,9	
Nombre de fentes codeur	500 pts	2 voies
Nombre de dents pignon	21	
Module pignon	0.8	

- Entraînement par pignon crémaillère
- Guidage par billes sur rail
- Réducteur 1/35,9 à train épicycloïdal
- Moteur 24 Vdc
P utile max 27,77 W
- Codeur optique 500 pts/tour



23080	24	85	4400	242	3808	555	71.08	9.47	1800	3580	20.33	6.70	5.10	0.52	5.30
25068	24	110	5200	280	4500	745	72.34	12.94	2080	4830	27.77	4.97	3.77	0.44	4.50
27058	24	122	6100	287	5348	911	72.12	15.76	2328	6520	36.46	3.68	2.70	0.36	3.80
29050	24	150	7100	303	6233	1119	72.23	19.39	2480	8080	45.21	2.97	2.00	0.31	3.30

Bobinage 25068, Inertie : 0.043 Kg.cm² Poids : 280g