

Séquence 02 :

Modéliser et simuler des comportements de systèmes complexes multi-physiques.

Table des matières

1.	Introduction	4
A.	Architecture matérielle du projet	5
B.	La phase d'expression et de spécification du besoin	5
C.	La phase de Conception-Modélisation-Simulation	6
1.	La modélisation White box.....	8
2.	La modélisation multi-physique	9
3.	La simulation, les modèles	10
4.	La comparaison des performances simulées et mesurées	11
5.	La modélisation « grise box ».....	11
6.	Le Model-in-the-loop (MIL).....	15
D.	La phase de Codage Implémentation	15
1.	Le Software-in-the-loop (SIL)	16
2.	Le Processor-in-the-loop (PIL).....	16
E.	La phase d'Intégration Vérification.....	17
1.	Le Hardware-In-the-Loop	17
F.	La Phase de Validation Recette.....	20

Compétences de SII de la séquence 2

Compétences visées :

A2. Analyser le système - identifier les éléments transformés et les flux ; - décrire les liaisons entre les blocs fonctionnels ;

B1. Identifier et caractériser les grandeurs agissant sur un système - définir, justifier la frontière de tout ou partie d'un système et répertorier les interactions ; - choisir les grandeurs et les paramètres influents en vue de les modéliser.

B2. Proposer ou justifier un modèle - associer un modèle à un système ou à son comportement ;

B4. Valider un modèle - interpréter les résultats obtenus ; - valider un modèle optimisé fourni.

Savoirs visés :

Chaîne d'énergie	Identifier et décrire la chaîne d'énergie du système Analyser les apports d'énergie, les transferts, le stockage, les pertes énergétiques Réaliser le bilan énergétique d'un système
Frontière de l'étude	Isoler un système et justifier l'isolement Identifier les grandeurs traversant la frontière d'étude
Caractéristiques des grandeurs physiques (mécaniques, électriques, thermiques, acoustiques, lumineuses, etc.)	Qualifier les grandeurs d'entrée et de sortie d'un système isolé Identifier la nature (grandeur effort, grandeur flux) Décrire les lois d'évolution des grandeurs Utiliser les lois et relations entre les grandeurs
Énergie et puissances Notion de pertes	Associer les grandeurs physiques aux échanges d'énergie et à la transmission de puissance Identifier les pertes d'énergie
Chaîne d'énergie	Associer un modèle à une source d'énergie Associer un modèle aux composants d'une chaîne d'énergie
Modèle de connaissance	Vérifier la compatibilité des résultats obtenus (amplitudes et variations) avec les lois et principes physiques d'évolution des grandeurs Comparer les résultats obtenus (amplitudes et variations) avec les données du cahier des charges fonctionnel

Centres d'intérêts visés : CI5 : Concevoir et utiliser un modèle relatif à un système en vue d'évaluer les performances de la chaîne d'énergie.

Les ingénieurs sont au cœur du processus de conception des systèmes complexes et doivent chaque jour relever les défis de la compétitivité, de l'innovation et de la performance. Cela ne peut se faire sans l'intégration de processus industriels structurés, ni sans la maîtrise des outils modernes de modélisation et de simulation. A chaque étape du cycle de conception, les méthodes mises en œuvre doivent permettre de baisser les coûts, de réduire le risque d'erreur et d'en minimiser les impacts.

Au cœur de ce processus, la modélisation et la simulation numérique jouent un rôle majeur et permettent aux ingénieurs d'anticiper, de comprendre et de vérifier les analyses qu'ils mènent tout au long du projet.

Les démarches industrielles standards, comme le cycle en V intègrent pleinement la simulation numérique au travers de méthodes associées comme le « Model Based Design » (conception basée sur le modèle). Les outils modernes de simulation permettent de créer des modèles globaux complexes intégrant tous les comportements du système et prenant en compte l'ensemble des interactions, cette démarche est appelée modélisation multi-physique. Le système réel peut avantageusement être remplacé par son modèle numérique pour réaliser des tests qui auparavant mobilisaient des moyens matériels et humains importants. Cette démarche impose de disposer de modèles validés qui reproduisent fidèlement le comportement des systèmes réels.

1. Introduction

L'évolution rapide des technologies amène l'homme à concevoir des systèmes de plus en plus complexes dont le comportement ne peut plus être modélisé sans l'aide des logiciels de simulation.

Les interactions entre les composants étant nombreuses, il est primordial d'être en mesure de modéliser le système dans sa globalité en prenant en compte sa structure, son dispositif de commande et son environnement. Les chaînes d'énergie et d'information ne peuvent plus être modélisées séparément mais doivent être intégrées dans un modèle unique. A partir de ce modèle il sera possible d'anticiper le comportement du système réel en phase de conception afin de permettre au système réel de remplir sa fonction.

La modélisation des systèmes complexes est au cœur de la formation des ingénieurs. Cette approche, qui s'appuie sur la décomposition de la complexité fait apparaître des sous-ensembles qui mobilisent chacun des technologies différentes. Le découplage des univers technologiques est donc nécessaire à la mise en place d'une démarche de modélisation pertinente. L'évolution très rapide des logiciels de simulation nous permet maintenant de modéliser toutes les parties d'un système à l'aide d'un logiciel unique. Cette approche multidisciplinaire, permet de gérer les échanges entre les sous-systèmes et de disposer de résultats qui prennent en compte la totalité des phénomènes physiques qui entrent en compte dans le fonctionnement du système. La pertinence des résultats est accrue et la mise en évidence des interactions devient possible. En contrepartie, la maîtrise des outils de simulation mobilise un champ de compétences nouveau, plus large et qui impose un processus d'apprentissage structuré. Cette nouvelle approche requiert un outil de modélisation et de simulation à environnement unique, visuel et interactif, validé industriellement et proposant tous les outils de modélisation de la chaîne d'information (régulateur classique, diagramme d'états...) et de la chaîne d'énergie (bibliothèques de composants avancées, maquette CAO intégrée...).

Afin de rendre efficace la démarche de formation, les méthodes et les concepts mis en œuvre doivent être les mêmes dans le cycle de formation des futurs ingénieurs que dans le monde industriel. Nous allons dans un premier temps voir comment il est possible de relier les standards de conception industrielle avec les concepts en vigueur dans les cursus de formation des ingénieurs.

La démarche mise en œuvre industriellement par les équipes d'ingénieurs pour concevoir et développer un système est un processus complexe qui respecte le plus souvent des standards industriels comme le « Cycle en V » et des méthodes associées comme le « Model Based Design ». Ces démarches imposent de disposer de modèles validés que les équipes d'ingénieurs utiliseront pour remplacer le matériel réel dans les phases de test et de réglage. La compétence « modéliser » est donc au cœur du processus d'apprentissage et reste l'une des compétences de base de l'ingénieur. L'évolution des logiciels de modélisation et de simulation a fortement modifié les méthodes d'acquisition de cette compétence pour les futurs ingénieurs. Le processus de modélisation ne se limite plus à écrire les équations différentielles qui régissent le comportement d'un système et à tenter de les résoudre analytiquement dans les rares cas où cela est possible. L'exploitation pertinente des logiciels de modélisation et de simulation permet de mettre en œuvre différentes approches afin de modéliser et d'obtenir rapidement des résultats valides :

A. Architecture matérielle du projet

Afin de mettre en œuvre la démarche, la Figure 4 présente l'environnement matériel du projet. Il n'est pas nécessaire de disposer de ce matériel pour comprendre les concepts abordés dans cette partie.

- *Un ordinateur équipé d'un logiciel de modélisation, de simulation et de pilotage :*

Le rôle de l'ordinateur est de servir de support à la conception des modèles numériques, à leur simulation et à la compilation des résultats. L'ordinateur va également servir à créer des interfaces de mesures et de pilotage afin de commander le moteur via la carte de commande du moteur.

- *La carte de commande du moteur :*

Le rôle de la carte de commande du moteur est de piloter le moteur. Elle possède des entrées, pour récupérer les informations issues des capteurs (position angulaire du moteur...) et prendre en compte la consigne issue de l'ordinateur, et des sorties pour fournir une tension aux bornes du moteur et permettre sa mise en rotation. La tension fournie est modulable en fonction de la logique de commande programmée. Dans notre exemple elle fera également office de carte de puissance.

- *Le moteur à courant continu :*

Il est alimenté par une tension continue et délivre une vitesse de rotation. Un capteur de position intégré renvoie à la carte de commande la position angulaire du moteur.

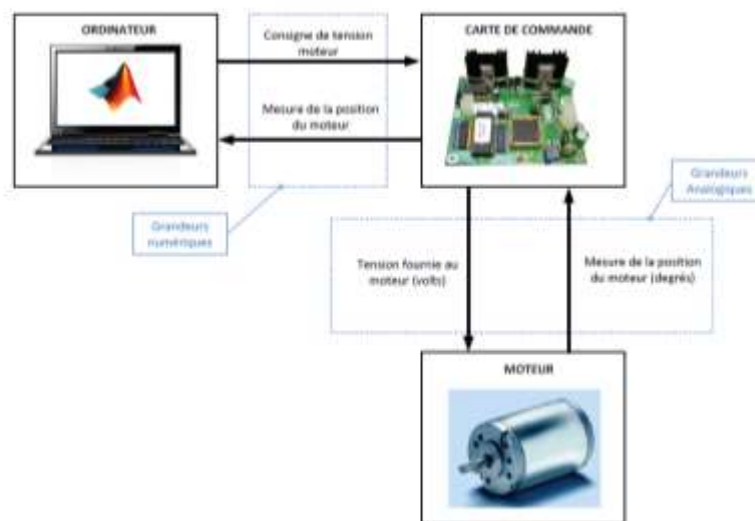


Figure 4 : architecture matérielle nécessaire à la mise en place de la démarche

B. La phase d'expression et de spécification du besoin

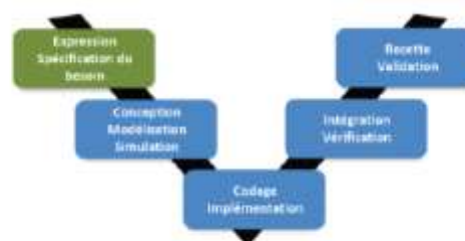


Figure 5 : la phase d'expression et de spécification du besoin

Le cycle démarre par l'**Expression et la Spécification** des besoins du système. Dans cette phase l'ensemble des exigences que doit remplir le système est formalisé. Les critères de performances sont définis et quantifiés. L'ensemble de ces informations est regroupé dans le cahier des charges fonctionnel. Il est analysé par les équipes de concepteur qui devront produire les solutions technologiques adéquates. Le diagramme des exigences de la Figure 6 regroupe les exigences de l'asservissement de position à concevoir.

- La modélisation « white box » :
- La modélisation « multi-physique » :
- La modélisation « grise box » :
- La modélisation « black box » :

Ces différentes approches ne sont pas exclusives dans le processus de modélisation. Elles sont au contraire complémentaires et il est fréquent de les combiner pour mener la démarche avec la plus grande pertinence. La connaissance et la maîtrise de ces concepts par les étudiants et les ingénieurs leur permettent d'être compétents dans la démarche de modélisation. Dans tous les cas la confrontation à l'expérimentation et à la mesure permettront de valider le comportement du modèle.

La modélisation du moteur à courant continu consiste à établir la relation entre la tension aux bornes de l'induit du moteur et sa vitesse de rotation.

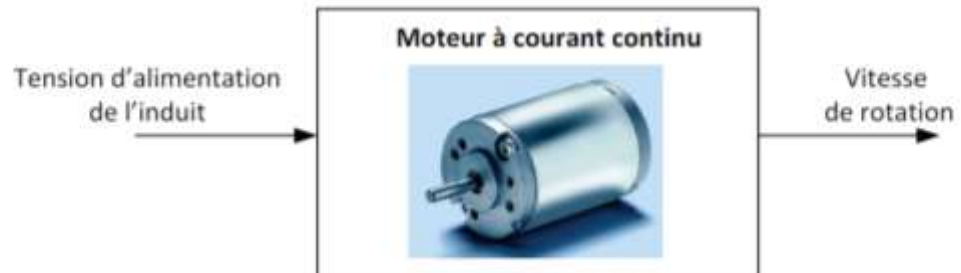


Figure 8 : relation entrée/sortie pour le moteur à courant continu

- *La modélisation « white box » :*
Cette approche consiste à écrire les équations et à les résoudre à l'aide de l'outil numérique. Cela permet d'obtenir une première approximation de la structure du modèle. Dans le cas général, les paramètres qui caractérisent le comportement du système ne sont jamais totalement connus. Il faut alors les négliger ou les estimer pour mener à bien la simulation. Les résultats obtenus doivent alors être confrontés à l'expérimentation afin d'optimiser et de valider le modèle.
- *La modélisation « multi-physique » :*
Cette approche consiste à assembler des composants dont le comportement est déjà modélisé dans le logiciel. Cette méthode ne nécessite pas l'écriture des équations et permet un gain de temps considérable. Les bibliothèques de composants permettent de modéliser dans tous les domaines physiques (mécanique, électrique, hydraulique, thermique...). Il est alors très simple de relier les différents domaines pour obtenir le modèle global d'un système. Il est également possible d'intégrer la maquette 3D de la structure modélisée ce qui permet de prendre en compte très facilement le comportement dynamique et les non-linéarités liées à la géométrie.
- *La modélisation « grise box » :*
Cette approche vient compléter le modèle « white box » ou « multi-physique » en estimant les paramètres inconnus par comparaison entre la réponse du modèle et la réponse du système réel. Des algorithmes permettent d'automatiser cette tâche et de donner des valeurs aux paramètres inconnus afin d'annuler les écarts entre les performances du modèle et les performances du système réel.
- *La modélisation « black box » :*
Cette approche, appelée également identification, consiste à associer un modèle mathématique à un comportement mesuré expérimentalement. L'outil numérique permet d'automatiser cette tâche en proposant des modèles mathématiques d'identification complexes. Cette méthode donne uniquement un modèle du comportement global du système sans se préoccuper de l'influence séparée des différents paramètres sur les performances.

La combinaison de ces approches permet de construire un modèle rapidement et de le valider vis-à-vis de l'expérimentation. Cette partie propose la mise en œuvre sur un cas concret d'une approche de type « Model Based Design ». En partant du cahier des charges et des exigences liées à la conception, les différentes étapes du cycle en V seront parcourues en utilisant les outils modernes de modélisation, de simulation et de validation. Les différentes approches de modélisation sont comparées et les avantages et inconvénients des différentes méthodes mis en évidence. La démarche permet de créer un lien entre le processus de formation et la réalité industrielle. Les étapes importantes de la démarche « Model Based Design » et de la détection d'erreur dans la génération de code C sont illustrées (Model in the loop, Software in the loop, Processor in the loop et Hardware in the loop).

1. La modélisation White box

Les équations électrique, mécanique et de couplage d'un moteur représentent le modèle de connaissance du moteur à courant continu et sont détaillées sur la Figure 9 :

$$\begin{cases} u(t) = e(t) + R i(t) + L \frac{d i(t)}{dt} \\ e(t) = K_e \omega_m(t) \\ J \frac{d \omega_m(t)}{dt} = C_m(t) - C_r(t) - f \omega_m(t) \\ C_m(t) = K_t i(t) \end{cases}$$

$u(t)$: tension de commande du moteur (V)
 $e(t)$: force contre électromotrice du moteur (V)
 $i(t)$: intensité dans le moteur (A)
 $C_m(t)$: couple exercé par le moteur (N.m)
 $C_r(t)$: couple résistant ramené sur l'arbre moteur (N.m)
 $\omega_m(t)$: vitesse de rotation du moteur (rad/s)
 R : résistance de l'induit (Ω)
 L : inductance de l'induit (H)
 J : inertie équivalente ramenée sur l'arbre moteur (kg.m^2)
 f : paramètre de frottement visqueux (N.m.s)
 K_t : constante de couple (N.m/A)
 K_e : coefficient de force contre électromotrice (V.s/rad)

Figure 9 : équations de comportements du moteur à courant continu

Après avoir appliqué la transformée de Laplace à ces équations, il est facile d'obtenir un modèle numérique sous forme de schéma bloc que l'on pourra modéliser et résoudre à l'aide du logiciel de simulation numérique (Figure 10). Ce principe de résolution est basé sur le principe de causalité. Le comportement dynamique d'un système est caractérisé par un bloc contenant, par exemple, la fonction de transfert du système avec une entrée et une sortie. L'information circulant dans les connexions entre deux blocs est un signal numérique orienté. La sortie du bloc est calculée numériquement en déterminant pour chaque pas de calcul la transformation du signal d'entrée imposé par le contenu du bloc. Cette approche largement utilisée industriellement nécessite une parfaite connaissance des lois physiques qui caractérisent le comportement des systèmes.

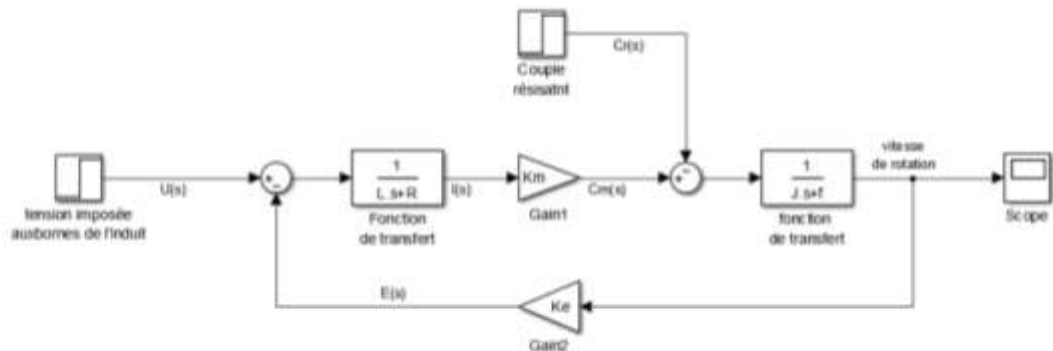


Figure 10 : modélisation sous forme de schéma bloc du moteur à courant continu

Il est maintenant possible de remplacer le système par son modèle comme représenté sur la Figure 11.

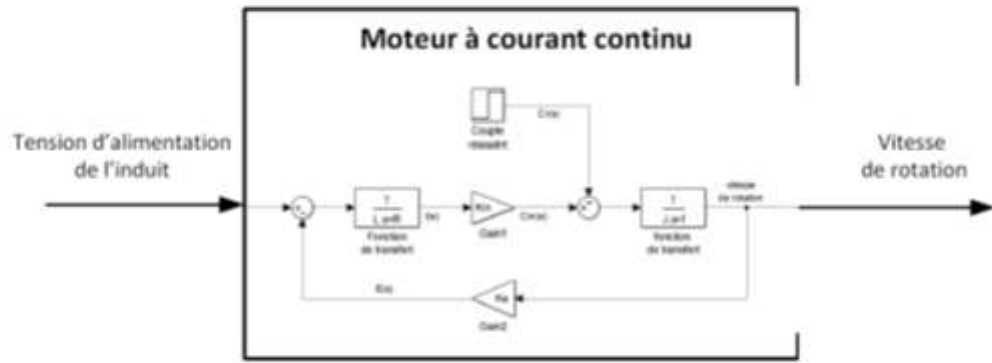


Figure 11 : modélisation « white box » du moteur à courant continu

2. La modélisation multi-physique

La modélisation multi-physique propose une approche acausale de la modélisation et permet de faire de la modélisation par assemblage de composants issus de bibliothèques prédéfinies. Le comportement physique des composants est pris en compte directement par le logiciel, il est donc possible de modéliser un système sans avoir à écrire les équations différentielles qui caractérise son comportement.

Les connexions entre deux composants ne sont pas orientées, ont une signification physique et transmettent un niveau d'information supérieure aux connexions en modélisation causale. Ces connexions peuvent être un fil électrique (transfert d'information de type courant et tension), un arbre moteur (transfert d'information de type couple et vitesse angulaire), l'extrémité de la tige d'un vérin (transfert d'information de type force et vitesse linéaire)... Le principe de calcul s'appuie sur un bilan de puissance à chaque nœud du modèle et ne repose pas sur le principe de causalité, d'où le nom de modélisation acausale.

La modélisation multi-physique du moteur à courant continu est représentée sur la Figure 12.

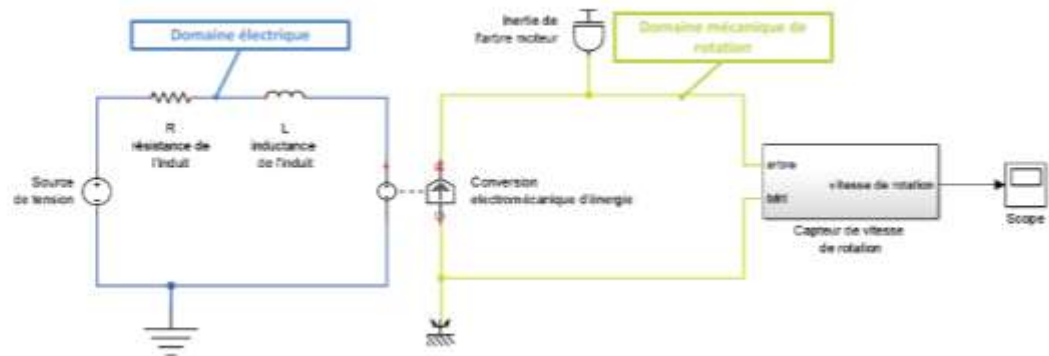


Figure 12 : modélisation par assemblage de composants du moteur à courant continu

Il est maintenant possible de remplacer le système par son modèle comme représenté sur la Figure 13.

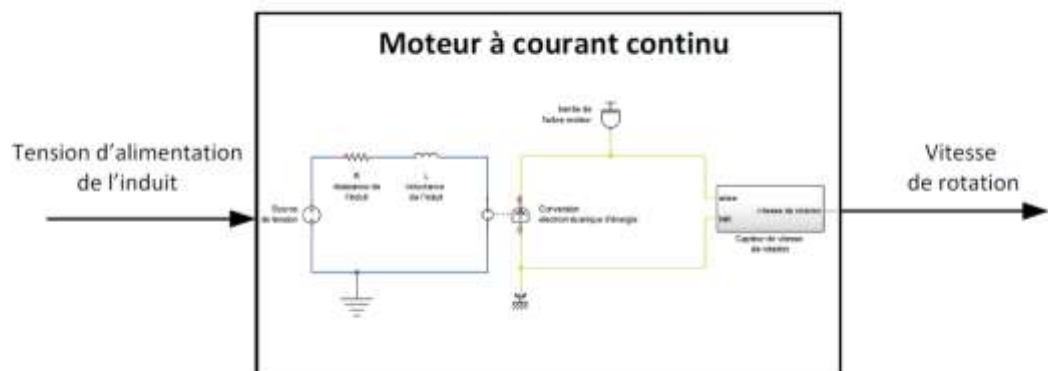


Figure 13 : modélisation multi-physique acausal du moteur à courant continu

3. La simulation, les modèles

La simulation et la résolution des modèles par le logiciel de simulation imposent de connaître les valeurs de tous les paramètres numériques. Dans notre exemple tous les paramètres sont connus sauf le paramètre f caractérisant les frottements. Il sera donc considéré comme nulle en première approximation ce qui revient à négliger l'influence des frottements.

Quelle que soit l'approche choisie, de type « white box » ou « multi-physique », les résultats obtenus par simulation numériques sont identiques (Figure 14). Dans les deux cas les phénomènes physiques pris en compte sont les mêmes, la différence réside dans la méthode utilisée pour construire le modèle et la méthode de résolution numérique utilisée qui diffère.

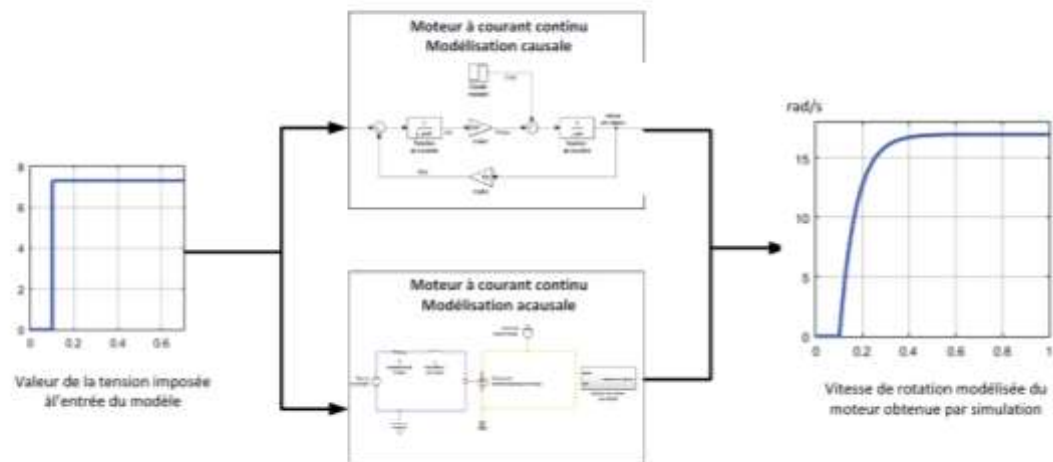


Figure 14 : résolution des modèles et visualisation des résultats

Les résultats obtenus permettent d'avoir une première approximation de l'évolution de la vitesse de rotation du moteur en fonction du temps. Dans le cas général cette approche est très souvent insuffisante et le seul moyen de vérifier la validité du modèle est d'évaluer les écarts entre les performances modélisées et les performances mesurées sur le système réel. En imposant une tension identique aux bornes de l'induit du moteur réel, il est possible de relever l'évolution de la vitesse de rotation réelle du moteur comme le montre la Figure 15.

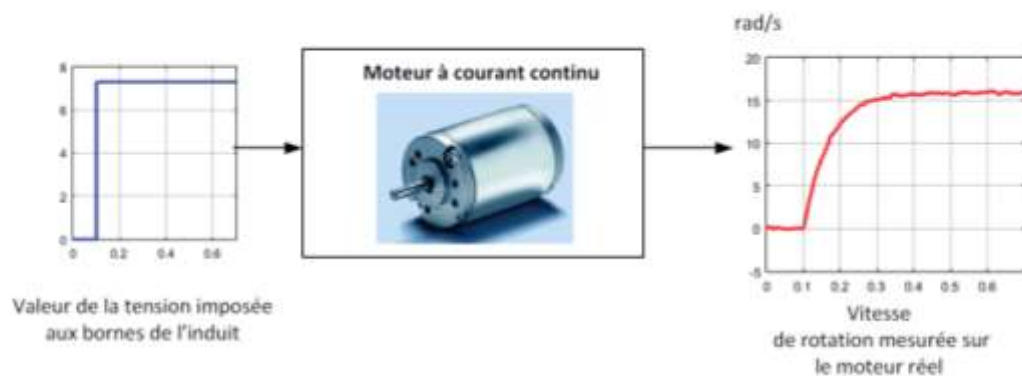


Figure 15 : mesure de l'évolution de la vitesse de rotation du moteur en fonction du temps

4. La comparaison des performances simulées et mesurées

Les performances simulées et mesurées sont alors comparées sur la Figure 16.

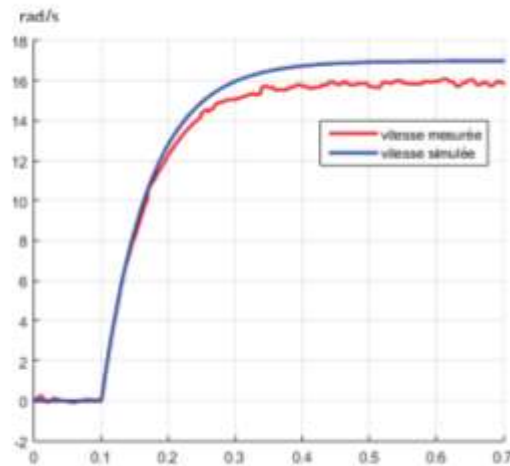


Figure 16 : évaluation des écarts entre les performances simulée et les performances mesurées

Dans le cas général et à ce stade, la mise en évidence d'écarts de comportement entre le modèle et le réel est très fréquent. Il faut alors évaluer ces écarts et trouver des pistes pour améliorer la modélisation en prenant en compte un phénomène nouveau ou en adaptant la valeur de certains paramètres.

Dans notre exemple l'analyse des écarts montre que le modèle n'est pas validé. L'absence de prise en compte des frottements est à l'origine de cet écart. Il faut alors évaluer les paramètres inconnus, ici le paramètre de frottement f . L'approche de type « grise box » va permettre de compléter le processus de modélisation en proposant une méthode de détermination des paramètres inconnus.

5. La modélisation « grise box »

Le principe de cette modélisation est d'estimer les paramètres inconnus du modèle par comparaison entre les performances simulées et les performances mesurées. Un algorithme intégré au logiciel de simulation va effectuer le travail d'analyse des écarts et proposer des valeurs possibles pour les paramètres inconnus permettant l'annulation des écarts. La méthode consiste à effectuer un essai sur le système réel en relevant le signal imposé à l'entrée du système et le signal obtenu à la sortie (Figure 17).

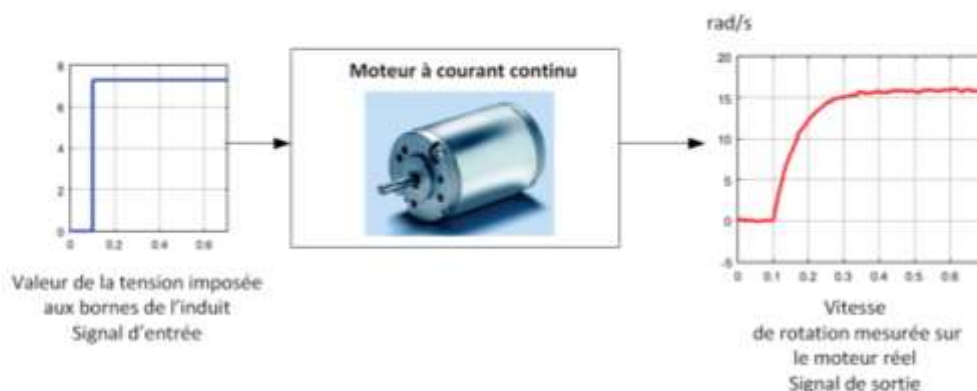


Figure 17 : essai réalisé en vue de l'estimation des paramètres inconnus

Un signal d'entrée identique est alors imposé à l'entrée du modèle et le logiciel va donner des valeurs possibles pour les paramètres inconnus de manière à annuler les écarts entre le réel et le modèle. Le modèle va chercher à atteindre la réponse cible, celle du système réel (Figure 18). Il est possible d'utiliser aussi bien le modèle causal que le modèle acausal pour réaliser cette optimisation.

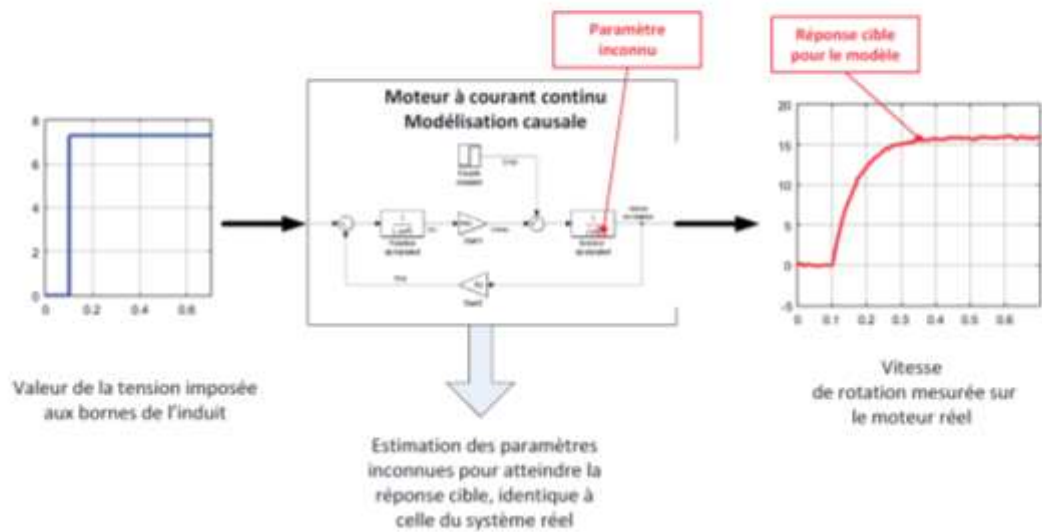


Figure 18 : illustration du principe de l'estimation de paramètre

La Figure 19 montre la réponse cible en rouge que va chercher à atteindre l'algorithme d'estimation de paramètres. La réponse du modèle avant estimation des paramètres est représentée sur la courbe bleue et la réponse du modèle après estimation des paramètres sur la courbe verte. L'intérêt est évident et permet d'obtenir un modèle validé dont le comportement est très proche de celui du système réel. Les écarts entre les performances mesurées et les performances simulées ont été annulés.

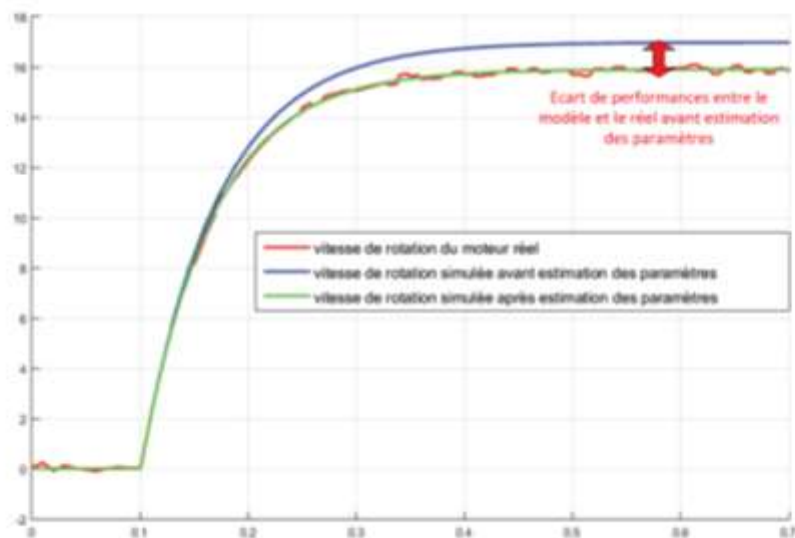


Figure 19 : les réponses du modèle avant et après estimation des paramètres inconnus

A l'issue de cette phase le projet dispose de modèles validés. Il est alors possible de remplacer le matériel (ici le moteur) par son modèle. La mise en œuvre de la phase de conception et de réglage de la commande se fera uniquement à l'aide du logiciel de simulation en travaillant sur le modèle. Les tests sur le matériel réel se feront après que les performances du modèle atteignent les performances souhaitées par le cahier des charges.

Nous pouvons alors utiliser le modèle validé du moteur pour construire l'asservissement en position du moteur à courant continu représenté sur la Figure 20.

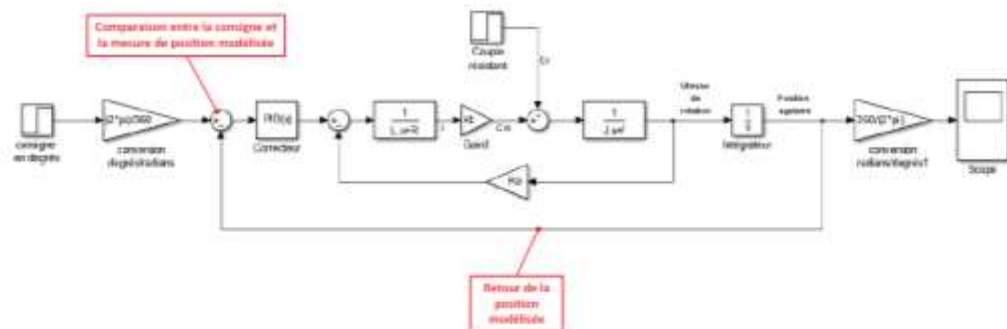


Figure 20 : modélisation de l'asservissement de position du moteur

Dans un premier temps le correcteur n'est pas actif et la réponse de l'asservissement de position est relevée sur la Figure 21.

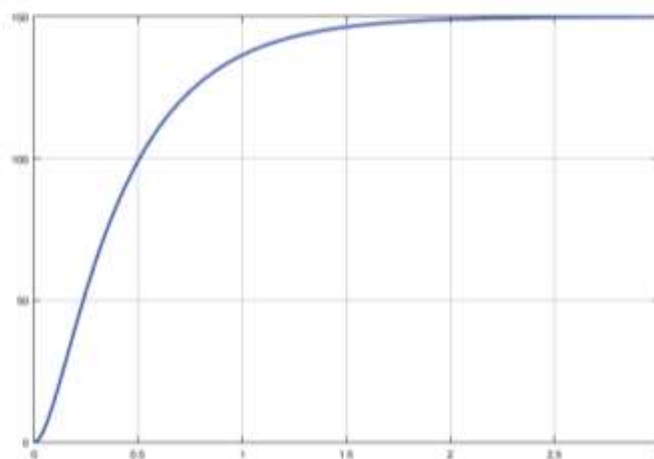


Figure 21 : réponse de l'asservissement de position non corrigé

Le diagramme des exigences impose les critères de performance suivants :

- Temps de réponse à 5% inférieur à 0.8 s
- Temps de montée à 80% inférieur à 0.4 s
- Ecart statique inférieur à 5% de la consigne

Il apparaît que les performances modélisées de l'asservissement ne sont pas conformes aux performances souhaitées. Le temps de réponse à 5% est de l'ordre de 1.3 s et le temps de montée à 80 % de l'ordre de 0.7s.

Il faut donc optimiser la commande du moteur en procédant au réglage du correcteur PID. Le logiciel de simulation propose alors de nombreuses fonctionnalités permettant de mettre en œuvre la démarche. La Figure 22 illustre l'utilisation de la démarche de réglage d'un correcteur dans le domaine fréquentiel à l'aide d'outils spécifiques.

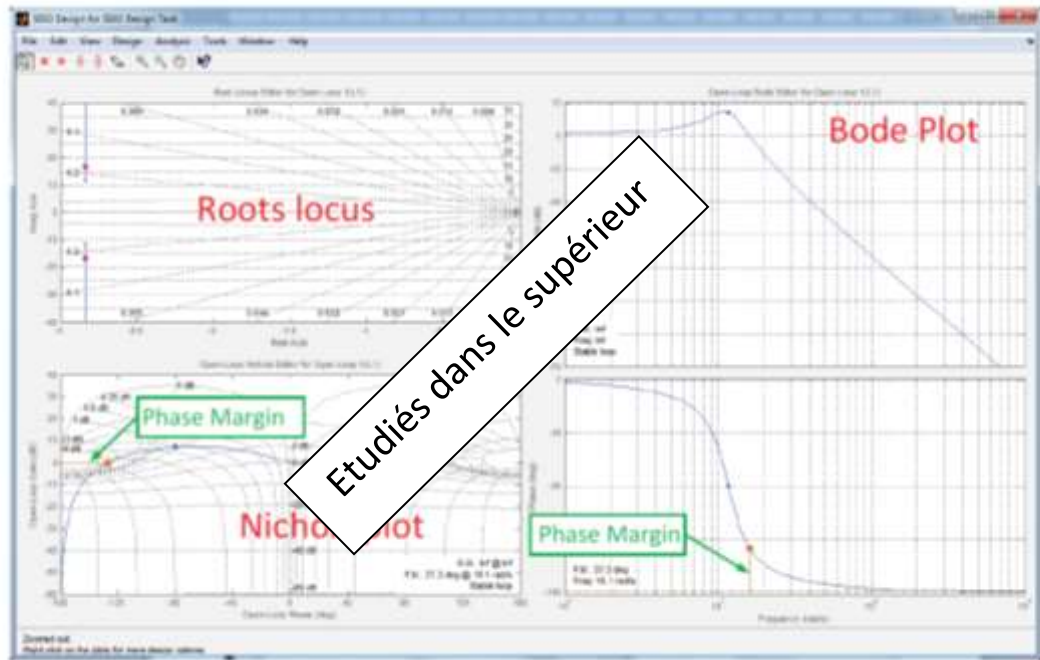


Figure 22 : illustration de l'utilisation d'un outil de contrôle commande pour régler un correcteur

A l'issue de cette phase de réglage, le correcteur est déterminé pour obtenir la réponse représentée sur la Figure 23.

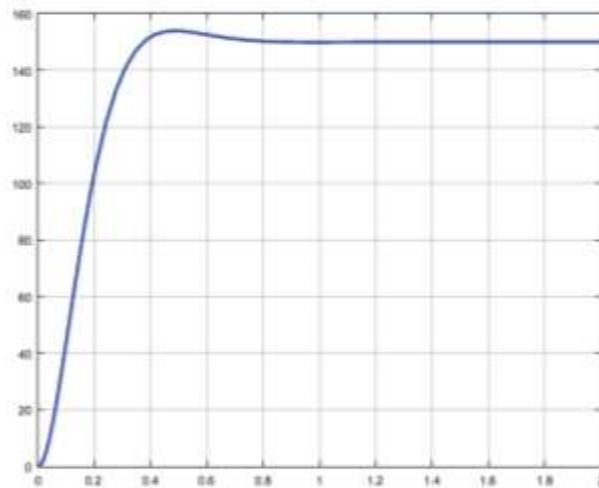


Figure 23 : réponse de l'asservissement de position corrigée

Le temps de réponse à 5% est maintenant de l'ordre de 0.3 s et le temps de montée à 80 % de l'ordre de 0.25s. Les performances modélisées sont conformes aux performances souhaitées et la phase de **Codage Implémentation** va pouvoir commencer.

6. Le Model-in-the-loop (MIL)

Cette étape se situe entièrement dans le domaine de la simulation et correspond à la simulation du modèle complet précédemment décrit. Le modèle est décrit entièrement en code MATLAB, il s'exécute entièrement sur l'ordinateur avec lequel a été construit le modèle.

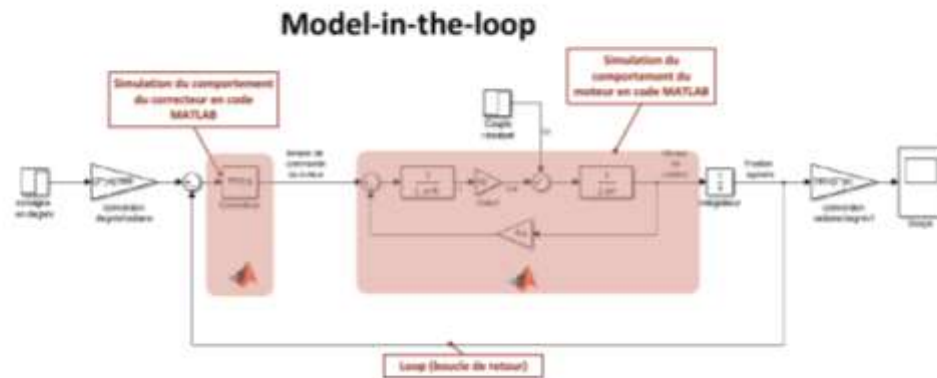


Figure 24 : le Model-in-the-loop (MIL)

D. La phase de Codage Implémentation



Figure 25 : la phase de Codage Implémentation

Cette phase permet de migrer progressivement du domaine de la simulation vers la phase de test sur le matériel réel.

A ce stade il est important de revenir sur la structure du modèle validé qui vient d'être finalisée à l'issue de la phase de **Conception Modélisation Simulation**. La Figure 26 montre les deux parties du modèle qu'il faudra bien distinguer pour comprendre les étapes qui vont suivre.

- la partie du modèle correspondant au correcteur qui était à la base des objectifs de conception du projet.
- la partie du modèle correspondant au moteur qui permet de disposer d'une modélisation validée des actionneurs.

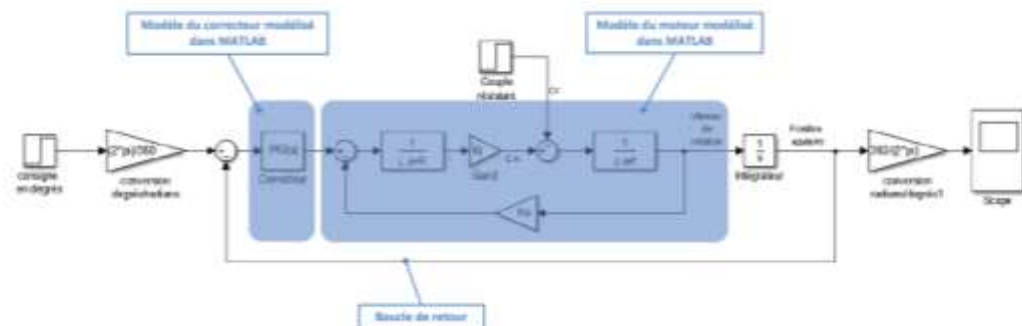


Figure 26 : structure du modèle à l'issue de la phase de Conception Modélisation Simulation

Cette phase permet de traduire des parties du modèle en Code C. Le code C est le langage de programmation utilisé par la carte de commande qui ne peut pas interpréter directement le code MATLAB. Une première méthode consiste alors à isoler la partie du modèle que l'on veut remplacer par du code C et à faire le bilan des entrées et des sorties de cette partie du modèle. Il est possible de réaliser ce codage manuellement. Dans ce cas une équipe de codeurs spécialiste du langage C vont réaliser les programmes qui généreront la même relation entre les entrées et les sorties que celle obtenue avec du code MATLAB. Cette approche est longue et mobilise des compétences de haut niveau en codage et en programmation.

Les logiciels de simulation permettent également de générer du code C automatiquement et MATLAB propose des interfaces permettant d'automatiser ce processus afin de gagner un temps précieux.

Qu'il soit automatique ou manuel, le processus de codage en langage C, peut générer des erreurs et le comportement du modèle codé en langage C peut différer de son comportement codé en MATLAB. Afin de détecter et de corriger ces erreurs avant qu'elles ne puissent impacter la phase de test sur du matériel réel, il faut mettre en place deux configurations de fonctionnement :

- le software-in-the-loop (SIL)
- le processor-in-the-loop (PIL)

1. Le Software-in-the-loop (SIL)

Dans cette configuration, une partie du modèle est remplacé par du code C qui va s'exécuter sur l'ordinateur. Cela permet de vérifier que la génération de code C n'a pas engendré d'erreur en terme de traduction de code C vers le code MATLAB.

Dans notre exemple nous allons remplacer le correcteur codé dans MATLAB par une fonction C qui va générer du code C en remplacement du code MATLAB et exécuter ce code C sur l'ordinateur (Figure 27).

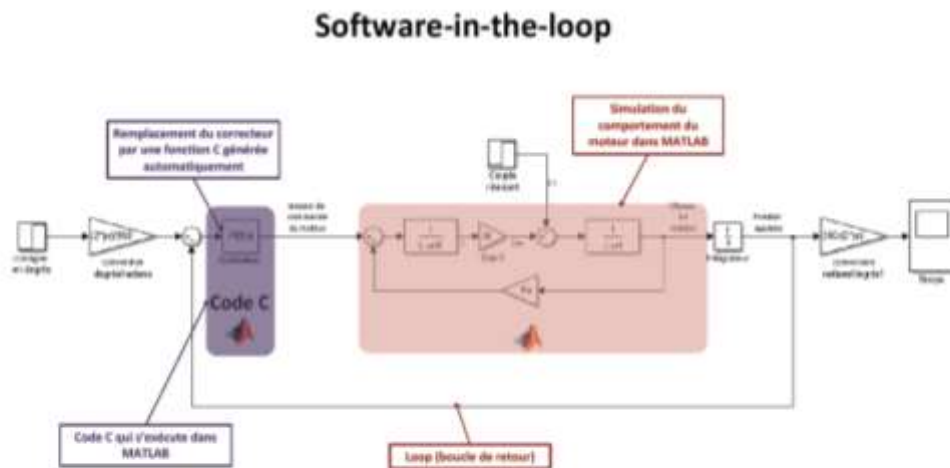
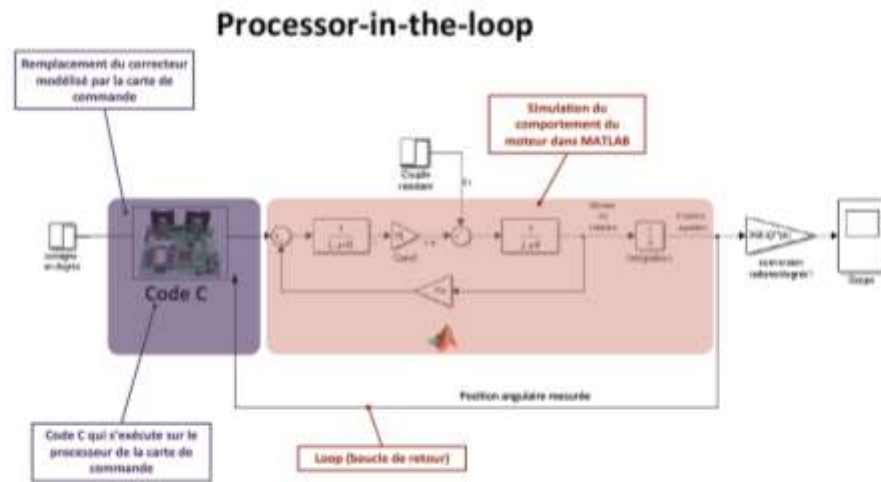


Figure 27 : le Software-in-the-loop(SIL)

Il faut alors vérifier que le comportement du modèle est le même, qu'il soit piloté avec le correcteur codé en code MATLAB ou avec le correcteur codé en code C. Une fois cette vérification effectuée, les concepteurs peuvent exclure toute erreur due à la génération de code C.

2. Le Processor-in-the-loop (PIL)

Cette configuration permet d'éviter les erreurs d'exécution du code C. En effet les conditions d'exécution du code C dépendent largement des capacités et des performances de la cible matérielle sur laquelle il s'exécute. En mode software-in-the-loop, le code s'exécute sur le processeur de l'ordinateur, en mode processor-in-the-loop, le code s'exécute sur la cible matérielle qui embarquera le code dans les conditions réelles de fonctionnement. Ces deux processeurs peuvent gérer différemment l'exécution d'un même programme. Le correcteur modélisé est alors remplacé par du code C qui s'exécutera directement sur la carte de commande. Dans cette phase, le processeur de la carte pilote le modèle du moteur.



A l'issue de ces deux phases, les erreurs de génération et d'exécution de code C ont été corrigées et la phase de test sur les actionneurs réels va pouvoir commencer.

E. La phase d'Intégration Vérification

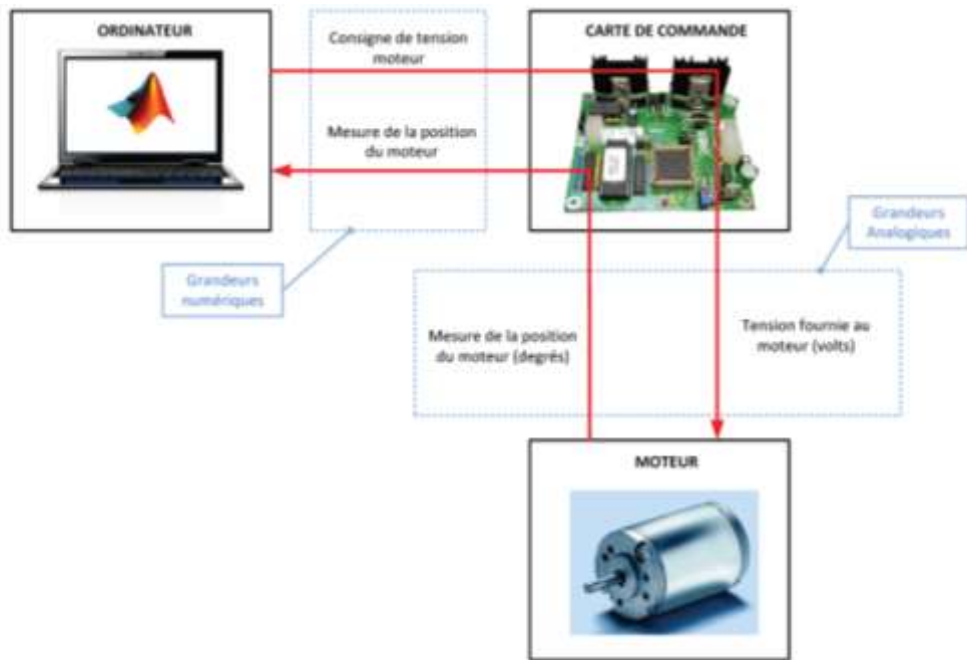


1. Le Hardware-In-the-Loop

Cette étape consiste à remplacer le modèle du moteur par le moteur réel. Le correcteur et les calculs associés peuvent être effectués soit par l'ordinateur (mode externe), soit par la carte de commande qui fonctionne de manière autonome (mode embarqué).

HIL - mode externe (Figure 30 et Figure 31)

Dans cette phase tous les calculs sont réalisés sur l'ordinateur et la carte de commande fonctionne uniquement en interface de puissance. Les fonctions de calcul du processeur de la carte de commande ne sont pas sollicitées.



Hardware-in-the-loop Mode externe

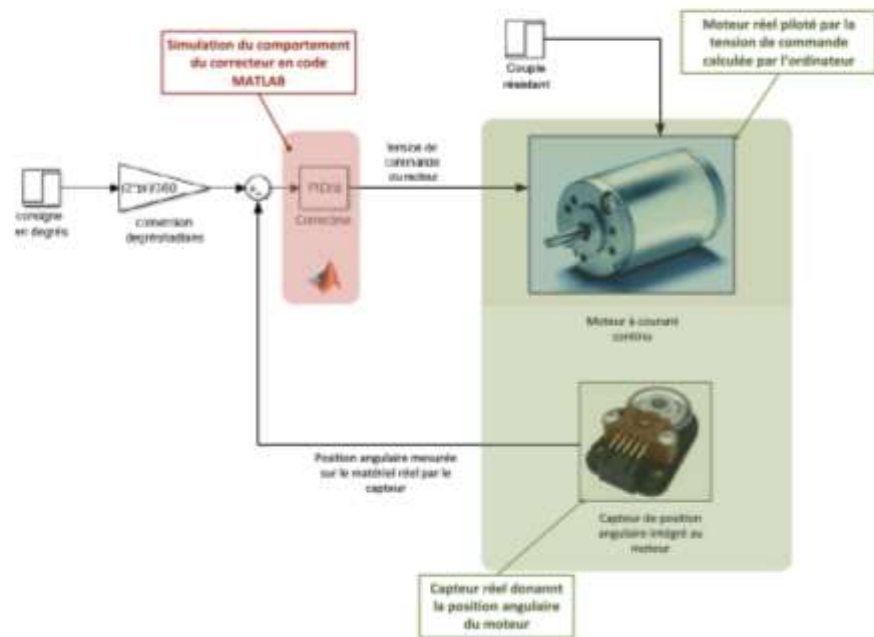


Figure 31 : le hardware-in-the-loop en mode externe: intégration du matériel dans la boucle

III. - mode embarqué (Figure 32 et Figure 33)

Dans cette phase tous les calculs sont réalisés par la carte de commande. La logique de commande s'exécute sur le processeur de la carte de commande.

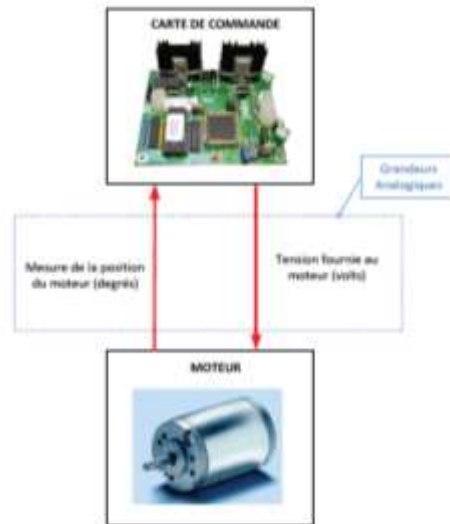


Figure 32 : architecture matérielle du Hardware-in-the-Loop en mode embarqué

Hardware-in-the-loop Mode embarqué

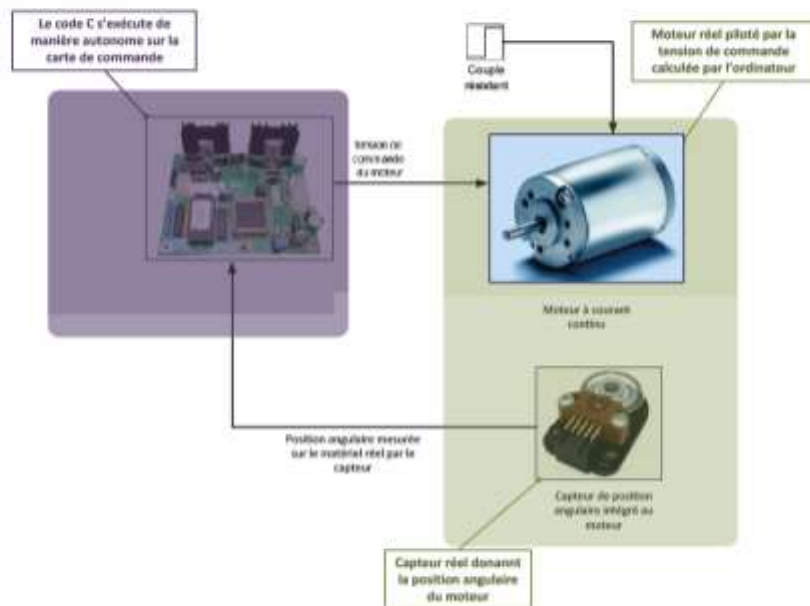


Figure 33 : le Hardware-in-the-Loop mode embarqué: fonctionnement autonome du matériel

L'une ou l'autre de ces configurations permettent de vérifier que le correcteur mis au point permet d'atteindre les performances spécifiées sur le matériel réel. Il est alors possible de relever la vitesse effective du moteur sur la Figure 34. Il est alors possible de passer à la phase de **Validation**.

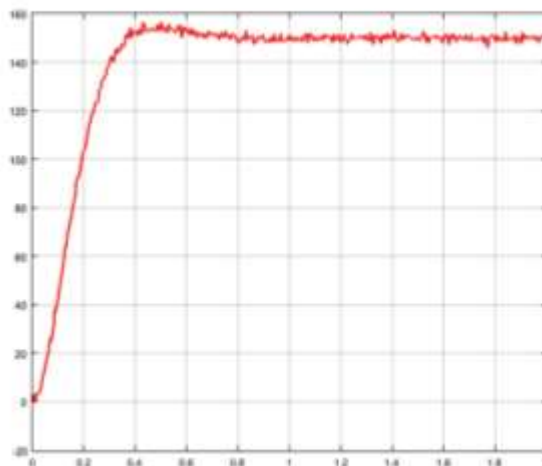


Figure 34 : visualisation de la vitesse de sortie du moteur à l'issu de la phase Hardware-in-the-loop

F. La Phase de Validation Recette

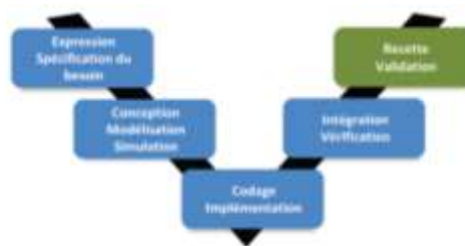


Figure 35 : la phase de Validation Recette

Dans cette phase, il reste à vérifier que les performances mesurées correspondent aux performances spécifiées dans le cahier des charges. L'analyse de la Figure 34 permet de voir que le temps de réponse à 5% est de l'ordre de 0,3 s et le temps de montée à 80 % de l'ordre de 0,25 s. Les performances réelles sont conformes aux performances modélisées et aux performances souhaitées.

La phase de test a été réduite au strict minimum et les spécifications imposées par le cahier des charges sont respectées.