**Eléments sur l'ingénierie des systèmes basée sur le modèle**

Le Model-Based Systems Engineering (MBSE), en français « ingénierie des systèmes basée sur le modèle » est une démarche d'ingénierie axée sur la création et l'exploitation de modèles, souvent multiphysiques. Ces modèles sont élaborés à partir des exigences du système, décrites en particulier dans l'analyse SysML, et sont utilisés tout au long des phases de conception, développement et validation du produit.

Intégration du MBSE dans le cycle en V d'un projet

**3**

Validation

Run-on-Hardware

Exigences spécifications

Deploy

Modèle vérifié par simulation

MIL

Modèle

multi-physique

Intégration

Tests

HIL

**3**

Prototypage

Run-on-Hardware

**1**

Simulations

SIL

**2.1**

Conception du système

**2.2**

Connected

I/O

PIL

**4**

Code exécutable

L'outil logiciel Matlab-Simulink-Stateflow permet d'appliquer une démarche MBSE pour la conception-réalisation d'un système technique intégrant une partie matérielle et une partie logicielle.

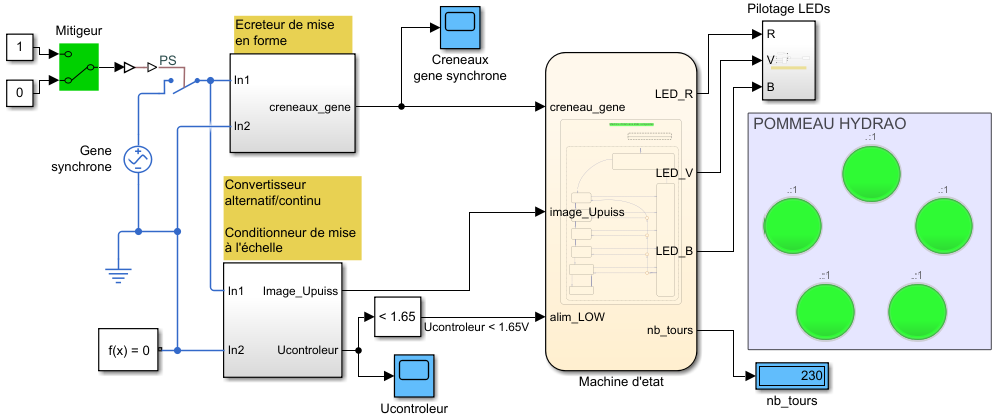
À partir de l'exemple du pommeau intelligent Hydrao, on montre ci-après l'utilisation de modèles, volontairement simples pour cette présentation, à différentes étapes de développement de la partie logicielle.

1. **Conception du modèle multi-physique : model-in-the-loop (MIL)**

Le modèle multiphysique\* composé du modèle de la partie contrôlée (A) du système, ici avec « simulink-simscape », et du modèle de contrôleur (B), ici une machine d'état « stateflow », est créé à partir de la description du système, ici par les diagrammes SysML.

Ce modèle est exécuté entièrement sur l'ordinateur, dans son environnement logiciel de conception modélisation (en langage « matlab » ici). Cette configuration de fonctionnement est nommée « Model-in-the-loop » (MIL).

*Nota : le modèle multiphysique retenu n'intègre que les éléments « nécessaires et suffisants » afin de ne pas surcharger les calculs lors de la simulation et donc de ne pas pénaliser inutilement la rapidité de traitement. Notamment toute la partie en amont du générateur d'énergie électrique est volontairement absente.*



**A**

**B**

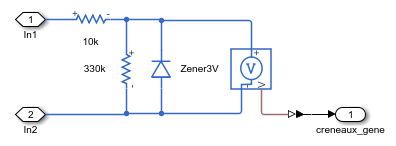
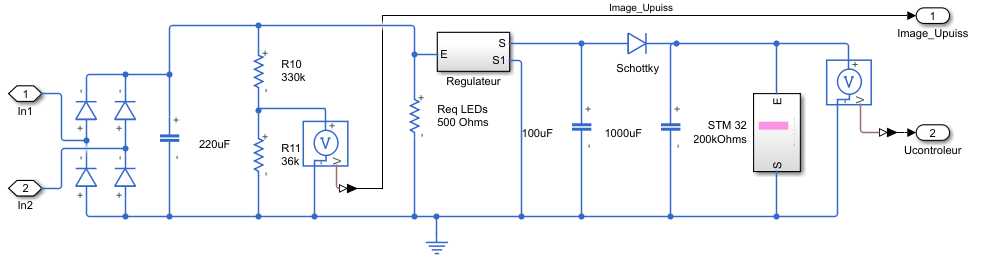


Figure 1 : model-in-the-loop



1. **Test du code dans la cible**

Dans la phase précédente, le modèle du contrôleur, décrit par un diagramme « stateflow » ici, est traité en code « matlab », code qui n’est pas exécutable par la carte programmable cible envisagée, par exemple carte arduino ou STmicroelectronics. Des différences de fonctionnement peuvent apparaitre entre le code « matlab » exécuté par l’ordinateur et le code implémenté et exécuté dans la carte cible. Dans la phase de test du code dans la cible, on remplace le modèle du contrôleur « stateflow » par son équivalent, en code C par exemple, qu’on implémente dans la carte cible et on effectue le test éventuellement en deux étapes :

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

L'ordinateur exécute seul la simulation de la partie contrôlée du système (A) et du code du contrôleur (B). Cette configuration de fonctionnement est nommée « Software-in-the-loop » (SIL). À noter que Matlab-Simulink permet de réaliser le codage en C automatiquement (fonctionnalité Create Sotfware-In-the-Loop block).

* 1. **Processor-in-the-loop (PIL)**

La carte cible est raccordée à l'ordinateur. Le code est généré et implémenté dans le processeur de la carte cible. Lors de la simulation, l'ordinateur simule la partie contrôlée du système (A) et, en interaction avec l'ordinateur, le processeur de la carte cible exécute le code du contrôleur (B). Cette configuration de fonctionnement est nommée « Processor-in-the-loop » (PIL). Avec Matlab-Simulink il suffit d'utiliser le mode d'exécution « Run on hardware », sous-mode « Monitor and Tune ».

1. **Prototypage**

Dans cette phase la carte cible\* est entièrement utilisée. Il n'y a donc plus de simulation et on se trouve en « temps réel ». Deux options sont possibles :

* 1. **Run on hardware**

L'ordinateur peut encore être raccordé à la carte cible pour « monitorer » des signaux ou régler des paramètres. Pour cela, avec Matlab-Simulink, on utilise, comme précédemment, le mode d'exécution « Run on hardware », sous-mode « Monitor and Tune », en revanche le modèle de la partie contrôlée du système (A) est remplacé par le réel raccordé aux entrées/sorties de la carte cible. Ce mode est très utile pour effectuer la phase de prototypage/mise au point puis la phase finale de validation/recette du produit puisqu'il permet de vérifier si les performances mesurées sont conformes aux spécifications du cahier des charges.

* 1. **Deploy**

Après avoir implémenté le code dans la carte cible, l'ordinateur est déconnecté et la carte cible est autonome. Pour cela, avec Matlab-Simulink, on supprime dans le modèle les éléments de « monitoring » et on utilise le mode d'exécution « Deploy », sous-mode « Build, Deploy&Start ».

*(\*) En début de phase prototypage, la carte cible utilisée n'est pas nécessairement la carte cible définitive.*

1. **Simulink en mode Connected I/O**

Dans ce mode particulier à Matlab-Simulink, la partie contrôlée est le réel, et la carte cible est raccordée à l'ordinateur. Les entrées/sorties de cette carte sont implémentées dans le modèle. L'exécution est entièrement gérée par Simulink sur le PC et elle est accompagnée d'une interaction avec la carte cible qui n'est utilisée que comme passerelle vers le réel via son serveur d'entrées-sorties. Il n'y a donc pas de génération de code vers la carte cible, ce qui fait gagner du temps de mise au point. Ce mode permet de tester partiellement la partie matérielle, avec toutefois des limitations liées à la communication PC-carte cible, notamment dans le cas d'intégration de processus à réponse rapide tels que les détections-gestions d'interruptions matérielles.

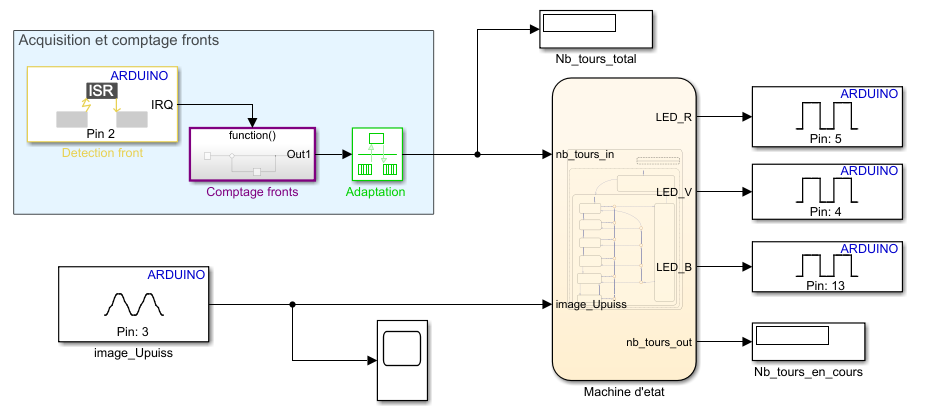


Figure 2 : Run-on-hardware (ou Connected I/O)

1. **Tableau récapitulatif**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mode | Partie contrôlée du système | Contrôleur | Interaction PC ↔ objet contrôlé | Phases de conception / réalisation |
| Model-in-the-loop | Modélisé sur PC | Modélisé sur PC (code matlab) | Non | Conception générale |
| Software-in-the-loop | Modélisé sur PC | Modélisé sur PC (code du processeur de la cible) | Non | Conception générale |
| Processor-in-the-loop | Modélisé sur PC | Réel sur carte cible (code implémenté) | Oui | Tests d'intégration logicielle |
| Connected I/O | Réel | Modélisé sur PC (code matlab) | Oui | Tests d'intégration matérielle |
| Run on hardware  Monitor and Tune | Réel | Réel sur carte cible (code implémenté) | Oui | Prototypage  Tests système  Validation |
| Deploy  Build, Deploy&Start | Réel | Réel sur carte cible (code implémenté) | Système autonome | Mise en service |