

Un ascenseur miniature jumeau réel d'une maquette numérique

Jean-Paul Chemla¹, Matthieu Lescieux¹, and Bernard Riera²

jean-paul.chemla@univ-tours.fr

¹LIFAT, Polytech Tours, Université de Tours

²Crestic, Université de Reims Champagne Ardenne

Résumé

Cet article présente une démarche originale pour la création d'une maquette pédagogique pour les TP de programmation d'automate. Cet ascenseur est piloté par un automate programmable (mode standard) mais a aussi un micro-contrôleur intégré pour gérer la sécurité et un mode démo qui permet aux étudiants de mieux comprendre le cahier des charges. Cette maquette d'ascenseur a été réalisée par des étudiants en projet en prenant comme modèle le fonctionnement d'une partie opérative simulée créée à l'aide du logiciel Factory I/O. Le matériel et son jumeau ont été testés en TP avec des étudiants de Polytech Tours.

Mots-cles : enseignement automatique, programmation API, pédagogie par projet, jumeau numérique, maquette ascenseur.

1 Introduction

1.1 Virtual commissioning

L'industrie 4.0 (ou l'usine du futur) est une expression utilisée pour décrire la quatrième révolution industrielle, caractérisée par l'intégration de technologies de pointe telles que l'Internet des objets (IoT), l'intelligence artificielle (IA), la robotique avancée et l'analyse des données dans les processus de production industriels. L'objectif de l'industrie 4.0 est de créer des systèmes de production plus efficaces, plus flexibles et plus intelligentes, capables de répondre rapidement aux demandes changeantes du marché tout en réduisant les coûts de production. Le jumeau numérique est considéré comme l'une des technologies clés de l'industrie 4.0. Il s'agit d'une représentation

virtuelle en temps réel d'un objet, d'un processus ou d'un système physique. La création d'un jumeau numérique implique souvent l'utilisation de modèles de simulation pour représenter le comportement du système physique en question. Les données générées par ces simulations sont une des sources utilisées pour créer le jumeau numérique, qui peut ensuite être utilisé pour surveiller, optimiser, tester ou simuler différentes conditions de fonctionnement sans avoir à toucher à l'objet physique réel. On notera que ces simulations peuvent être très utiles pendant la phase de conception du système de production pour tester des lois de commande avant la phase de mise en service. A ce titre, dans l'industrie 4.0, la mise en service virtuelle (« virtual commissioning ») est considérée comme un outil novateur permet-

tant aux techniciens et ingénieurs de simuler différents scénarios de production, d'analyser les performances du système et de détecter les erreurs de conception avant la mise en service réelle. Cette méthode permet donc de réduire les coûts de mise en service, car les tests peuvent être effectués virtuellement sans la nécessité de construire un prototype physique [1]. Il existe plusieurs logiciels de simulation qui permettent de créer des jumeaux numériques de systèmes de production de faire de la mise en service virtuelle. Parmi les logiciels couramment utilisés pour la création de jumeaux numériques pour le « virtual commissioning », nous pouvons citer :

- Siemens NX : Il s'agit d'un logiciel de conception assistée par ordinateur (CAO) qui peut être utilisé pour créer des jumeaux numériques de produits industriels.
- Dassault Systèmes' 3DEXPERIENCE platform : Cette plateforme propose des outils pour la création de jumeaux numériques de produits, de bâtiments, de villes et d'autres systèmes complexes.
- Factory I/O [2] est un logiciel qui permet de simuler en 3D, de façon simple, réaliste et interactive, des systèmes de production. Connecté à un API logiciel ou matériel, Factory I/O permet de faire du « virtual commissioning » pédagogique.

Pour l'enseignement de l'automatique séquentielle et la programmation des API (automates programmables industriels), l'utilisation de maquettes pédagogiques qui sont souvent une partie ou un modèle réduit d'un système automatisé industriel est très utile en travaux pratiques [3]. L'étudiant doit pouvoir essayer en toute sécurité un programme automate qui ne respecte pas forcément ni le cahier des charges ni les règles d'utilisation

du matériel (par exemple pour un ascenseur : monter la cabine alors qu'elle se trouve déjà à l'étage le plus haut).

1.2 besoins pour les TP d'automatique

Polytech Tours avait une maquette d'ascenseur ancienne et qui présentait plusieurs dysfonctionnements. Pour continuer à faire les TP nous avons tout d'abord utilisé le logiciel Factory I/O avec lequel un modèle d'un ascenseur à 4 étages avait été réalisé par un étudiant pendant un projet. En effet ce logiciel permet de construire une machine à sa convenance en partant de briques élémentaires. Dans sa bibliothèque, il y a un monte-charge (avec deux vitesses), des capteurs et des boutons lumineux, ce qui rendait possible la réalisation de cet ascenseur simulé.

Nous avons ensuite voulu réaliser une nouvelle maquette qui reprenne quelques caractéristiques du modèle d'ascenseur réalisé avec Factory I/O : 2 capteurs par étage et 2 vitesses possibles pour une approche en vitesse lente de l'étage désiré. Nous avons aussi voulu avoir une gestion de la sécurité qui ne dépende pas de l'automate programmable et un mode démo qui permet de présenter le fonctionnement attendu. La réalisation de cette maquette a été confiée à un groupe de 6 étudiants sur un projet de conception de systèmes du département de mécanique de l'école.

La première partie décrit la maquette virtuelle réalisée avec le logiciel Factory I/O, ses caractéristiques et son mode de pilotage. Le cahier des charges donné aux étudiants pour la réalisation de la maquette réelle est présenté dans une seconde partie. Pour finir nous montrons ce qui a été réalisé et faisons un retour d'expérience.

2 Jumeau numérique

Le logiciel Factory I/O permet une simulation réaliste et interactive de parties opératives industrielles. Cette simulation peut être reliée à un automate programmable ou tout autre système de commande qui communique (via le réseau ou une carte d'acquisition) avec ce logiciel.

Pour l'apprentissage de la programmation des automates programmables, un TP apprécié des étudiants est la commande d'un ascenseur. Pendant la crise sanitaire, cet apprentissage à distance a nécessité l'utilisation de maquettes virtuelles adaptées à cette situation. A l'aide du logiciel Factory I/O, nous avons réalisé un ascenseur virtuel à l'aide des briques élémentaires que propose ce logiciel :

- un monte-charge qui accepte les commandes pour monter, descendre, vitesse lente, charger, décharger. Ces deux dernières commandes s'expliquent car le sol du monte-charge est un tapis roulant. Les mouvements de charger/décharger pourront symboliser les mouvements d'une porte (ouverture/fermeture). Il suffit de placer un petit objet sur le tapis.
- 4 plateformes pour symboliser les 4 étages
- 2 x 4 capteurs d'étage. A chaque étage, l'ascenseur est bien positionné quand les deux capteurs sont actifs. L'activation d'un seul des deux capteurs permet d'activer la vitesse lente si le monte charge doit s'arrêter à cet étage.
- des boutons d'appels intérieurs reportés sur l'armoire de commande. Les boutons sont munis de voyant pour visualiser la mémorisation de l'appel.
- des boutons d'appels extérieurs. Aux étages intermédiaires, il y a deux boutons d'appel : pour monter ou pour des-

cendre.



Vue de la simulation d'ascenseur

Pour faire le TP à distance, les étudiants disposent de Factory I/O et du logiciel Unity Pro de Schneider (devenu EcoStruxure control expert) en version étudiant en utilisant le mode simulation de l'automate. La communication entre ce dernier et Factory I/O se fait avec le

protocole Modbus TCP.

3 Le jumeau réel

3.1 le cahier des charges

La conception et la réalisation de cette maquette a été confiée à un projet de groupe d'étudiants en 4ème année d'école d'ingénieur. Le cahier des charges reprend le fonctionnement de la maquette virtuelle de Factory I/O :

- 4 étages (numérotés de 1 à 4)
- deux vitesses de mouvement, 2 capteurs par étage
- des boutons munis de voyants pour appeler l'ascenseur
- une sécurité contre les mauvaises utilisations en particulier : l'arrêt du moteur si la cabine est trop haute, trop basse ou si l'arrêt d'urgence est enclenché (indépendant de l'automate)

Nous avons ajouté à ces caractéristiques le fait de pouvoir déplacer manuellement la cabine (ou avec des commandes indépendantes de l'automate) et éventuellement un mode "démonstration".

Enfin, la maquette doit être robuste, réglable (position des capteurs par exemple), transportable et réalisée avec le plus de moyens locaux de fabrication possible (découpeuse laser, imprimante 3D).

3.2 Matériels mis en oeuvre

Après analyse, les étudiants ont choisi d'utiliser :

- un moteur pas à pas avec une vis sans fin pour l'encombrement minimal et la gestion simple des vitesses.
- des capteurs mécaniques pour les surcourses et les capteurs de porte, des capteurs à effet hall pour les positions aux étages (moins sensible à la précision de la position de la cabine).

- une structure en profilé en aluminium
- un habillage en PMMA découpé et gravé au laser
- une cabine imprimée en 3D (ainsi que les charnières des portes)

3.3 Arduino in the middle

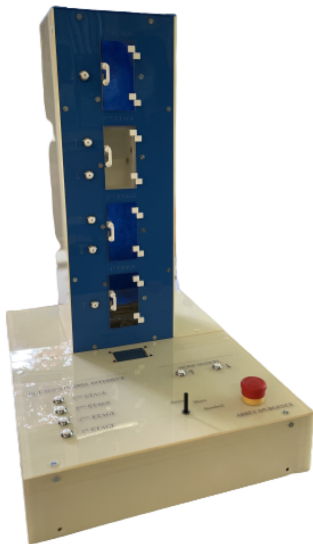
Pour la partie électronique, il existe un grand choix de capteurs bon marchés et des pilotes pour commander le moteur pas à pas adaptés aux micro-contrôleurs. Il a donc été décidé de mettre un Arduino (Mega) dans le socle de la maquette. Cette platine s'est avérée insuffisante pour exécuter l'ensemble des instructions dans le temps de cycle imparti de 10ms. Nous avons donc ajouté un deuxième Arduino (Micro) pour sous-traiter la scrutation et l'actualisation de l'ensemble des modules d'extension GPIO sur le bus I2C. Les 2 Arduino communiquent selon une liaison série via deux UART dédiés, respectifs. Ces micro-contrôleurs permettent entre autre de :

- piloter le moteur pas à pas à travers un driver adapté
- gérer l'ensemble des boutons d'appels et leurs voyants, les sur-courses, le bouton d'arrêt d'urgence
- d'avoir un mode manuel pour déplacer la cabine à l'aide de deux boutons
- d'envoyer les informations des capteurs à l'automate à travers des cartes munies d'opto-coupleurs
- de récupérer les ordres de l'automate par d'autres cartes (opto-coupleurs) en filtrant les ordres de façon à gérer la sécurité
- d'assurer un mode démonstration indépendant de l'automate

Le bus I2C a largement été utilisé pour communiquer avec les boutons d'appel ainsi qu'avec les cartes reliées à l'automate.

3.4 Maquette réalisée

Le cahier des charges initial a été respecté. Les étudiants ont mis sur un espace de stockage partagé l'ensemble des documents utilisés (plans des pièces, schéma électrique, liste des entrées et sorties, programme Arduino et leur rapport) (lien sur demande). Pour le mode démonstration, un des auteurs a réalisé un programme avec une gestion d'états (démonstration/manuel/standard). L'automate est connecté à la maquette avec des nappes pour les E/S et l'alimentation 24V. Les nappes permettent un changement d'automate simplifié. La maquette a été utilisée pour les TP cette année (environ 14 binômes ont fait un TP de 8 heures). Le mode démonstration est très pratique pour expliquer comment fonctionne un ascenseur (en particulier les rôles des boutons d'appels pour monter ou pour descendre). Une vidéo du fonctionnement est visible ici : <https://youtube.com/shorts/rfm1Vpc19VQ>.



Maquette réalisée

Il nous reste à proposer une structure de programme API qui permette aux étudiants de ne faire qu'un seul programme qui puisse être testé indifféremment sur la maquette numérique de Factory I/O ou sur la maquette réelle réalisée. C'est le rôle de la partie suivante.

4 Programmation API

Pour l'étudiant, l'objectif premier est de concevoir une commande par un grafcet. Pour tester la validation de cette conception, ce grafcet doit être converti en programmation d'automate (principalement en utilisant le langage SFC et quelques blocs fonctionnels). Nous mettons à disposition un programme initial qui comprend la liste des variables qui servent d'entrée sorties à leur programme.

Pour assurer son rôle de jumeau numérique, le programme automate réalisé doit pouvoir indifféremment commander la

maquette réelle ou son jumeau numérique en fonction du contexte. Nous envisageons l'utilisation du jumeau numérique dans un contexte où l'étudiant n'a accès qu'à un ordinateur où Factory I/O s'exécute et communique avec une simulation d'automate sur le même poste en utilisant le protocole Modbus TCP. Si le programme est téléversé sur un vrai automate relié par ses entrées et sorties à la maquette réelle, le programme doit commander la maquette réelle.

Pour que les variables définies dans le sujet du TP puissent être indifféremment utilisées pour la maquette réelle ou son jumeau numérique, il doit y avoir deux programmes additionnels : un préliminaire qui va soit récupérer les vraies entrées de l'automate soit celles de Factory I/O via une communication Modbus. Le choix entre ces deux modes doit être transparent donc une détection du mode (réel ou simulé) doit être mise en oeuvre. Le programme préliminaire doit être exécuté avant les parties de programme réalisées par l'étudiant. Un programme post-traitement va s'occuper de diriger des variables de sortie du programme de l'étudiant pour commander les vraies sorties ou les actions de la maquette virtuelles.

Pour obtenir ce canevas de programme, nous pouvons utiliser le site web : https://crestic-fiotoiplcmdb.univ-reims.fr/canvas_generator.php développé dans le cadre du projet *ET-LIOS* qu'il faudra modifier pour utiliser aussi la maquette réelle.

5 Conclusion

Notre démarche a été de s'inspirer d'une maquette numérique d'ascenseur réalisée avec le logiciel Factory I/O pour créer une maquette réelle de ce système. Ces maquettes servent de TP pour l'enseignement de l'automatique

séquentielle. Ces maquettes ont été réalisées par des étudiants lors de projets.

La maquette réelle est originale car elle intègre un micro-contrôleur pour assurer les mouvements de la cabine (2 vitesses), la sécurité, les mouvements manuels et un mode démonstration qui permet aux étudiants de comprendre le cahier des charges du TP. Enfin une structure de programme API permet aux étudiants de s'exercer hors TP avec le jumeau numérique de l'ascenseur. L'étape suivante serait d'optimiser l'utilisation de la maquette réelle en demandant aux étudiants de n'utiliser que le jumeau numérique pendant la moitié du temps de TP (et chez eux s'ils le souhaitent).

Références

- [1] Jules Jouffroy, Bernard Riera, Alexandre Philippot, François Gellot, Stéphane Lecasse. *Retour d'expérience sur la réalisation du jumeau numérique d'une Cellule Flexible avec Emulate3D*. 17ème colloque national S-mart AIP-PRIMECA, Université Polytechnique Hauts-de-France [UPHF], Mar 2021, LAVAL VIRTUAL WORLD, France. <hal-03296136>
- [2] Riera B., Vigário B. *HOME I/O and FACTORY I/O : a virtual house and a virtual plant for control education* IFAC-PapersOnLine, 50 (2017), pp. 9144-9149.
- [3] Florence Lecroq, Jean Grieu, Hadhoum Boukachour, Youssouf Ibrahim-Elmi, Maxence Hamel, et al.. *Jumeaux numériques et automates programmables industriels : un modèle hybride pour l'enseignement de l'informatique industrielle*. Rencontres Universitaires Numériques Normandes, Nov 2019, Caen, France. <hal-02454362>