

Les outils de simulation de systèmes automatisés

STÉPHANE MASSART, DIDIER GRALAND^[1]

Les logiciels de simulation de systèmes automatisés connaissent un essor important depuis quelques années avec l'apparition de produits de plus en plus performants et accessibles en termes de prix. Après une présentation de la technologie et de l'apport de ces outils, nous aborderons deux études de cas.

Même si des solutions étaient dans les faits déjà utilisées par détournement de l'utilisation de produits de supervision à des fins de simulation, il semble évident que la simulation des systèmes automatisés avec un logiciel informatique est entrée dans une phase active et ascendante. La raison en est l'accès à des technologies sophistiquées à moindre coût, rendu possible notamment par l'industrie du jeu vidéo, ainsi qu'une prise de conscience de l'importance de la simulation par un plus grand nombre d'acteurs industriels de l'automatisation, au-delà des secteurs habituels où la simulation est depuis longtemps indispensable (le nucléaire, par exemple).

Les buts de la simulation

Les principaux buts visés par la simulation sont la mise au point des programmes d'automates (plus connue sous le nom de *virtual commissioning*), l'aide à la définition d'un système automatisé (cadences...), la formation des opérateurs, la simulation de pannes et la présentation d'un système automatisé, pour aider à le vendre avant sa réalisation, par exemple.

Dans l'industrie, la mise au point des programmes d'automates a pour but, au-delà de la fiabilisation de ceux-ci, de réduire les temps de mise au point sur site. Le gain dans ce cas sera la différence entre le temps consacré à la création du modèle virtuel et le temps économisé sur la mise au point sur site. Cette économie sera encore plus importante dans le cas de la mise au point d'un système sur un site éloigné géographiquement ou dans une ambiance difficile, ou d'un système critique (les temps d'arrêt pour modifications et remise en service devant être le plus possible réduits).

Mots-clés
automatismes,
logiciel,
modélisation,
simulation

La technologie

Les principaux constituants d'un logiciel de simulation de systèmes automatisés sont un moteur de rendu, un moteur physique, un module de dialogue permettant à un système externe au logiciel (automate ou automate virtuel) de piloter le système automatisé, et d'éventuels modules annexes, de gestion d'une IHM (interface homme-machine), de simulation des schémas, etc.

Le moteur de rendu

Permettant simplement de visualiser le système automatisé, il peut aujourd'hui bénéficier de fonctionnalités facilement accessibles sur un ordinateur moderne : 3D, caméras, objets texturés, etc., et pourquoi pas l'affichage en 3D avec les lunettes adéquates.

Le moteur physique

Le moteur physique, bibliothèque logicielle, permet le traitement de lois physiques qui vont s'appliquer aux éléments de la scène, comme la gravité, la détection des collisions entre les différents objets, la cinétique...

Il convient d'être prudent et de comprendre en quoi son utilisation est importante mais parfois délicate – nous parlerons plus loin du « delta » (c'est-à-dire de la différence) entre la simulation et le monde réel.

Notons que la technologie développée par NVidia avec son moteur PhysX a l'avantage de pouvoir s'exécuter sur le processeur de la carte vidéo, ce qui peut être intéressant en termes de performances de simulation, notamment si la taille du système à simuler est importante.

Le module de dialogue

Il permet de piloter le modèle virtuel à partir d'un ou plusieurs automates programmables réels ou simulés (simulateurs intégrés aux logiciels constructeurs) connectés au PC. Des passerelles universelles (interfaces pour l'échange de données entre les différents composants d'un système d'automatisation industriel, OPC par exemple) peuvent aussi être utilisées, bien qu'engendrant parfois un surcroît de complexité au niveau de la configuration des échanges. Le rôle du module de dialogue sera d'échanger les entrées/sorties entre le programme de pilotage du système et le système simulé. Idéalement, ces échanges s'effectueront en un temps le plus court possible et n'entraîneront aucune modification sur le programme de l'application d'automatisme. Le but ultime est de « laisser croire » au système pilotant qu'il est en lien avec un système réel

^[1] Respectivement dirigeant de la société IRAI (Innovation recherche automatisme informatique), et professeur agrégé en SII option ingénierie électrique au lycée Nicolas-Appert d'Orvault (44).

et non une simulation. Pour fixer cette idée, pensons au film de science-fiction *Matrix* et à sa simulation du monde réel abusant l'esprit humain.

Les autres modules

D'autres éléments pourront se greffer à la simulation, par exemple la création d'IHM, la simulation de la partie schémas (électriques, pneumatiques, hydrauliques, etc.), le lien avec un simulateur numérique (type Matlab/Simulink), un module de simulation de pannes, etc.

L'ouverture du logiciel de simulation aux multiples systèmes pouvant venir s'y greffer est importante. Prenons l'exemple de la simulation de la gestion des bagages sur des tapis roulants d'un aéroport : il paraît plus qu'intéressant que la simulation puisse accéder à une base de données pour gérer l'aiguillage des bagages.

Les performances

La performance du système de simulation est un point clé et critique ; il convient en effet de ne pas trop s'éloigner du modèle réel et de sa réactivité. En la matière, la conception de la structure du simulateur est au moins d'importance égale à la puissance du PC sur lequel il tourne.

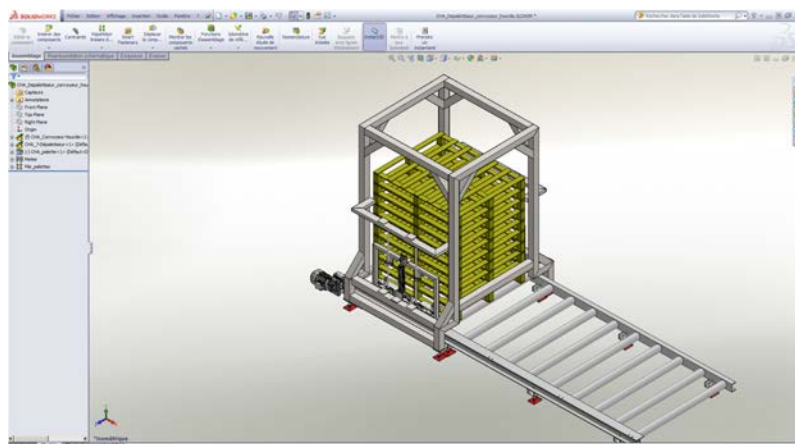
Idéalement, le simulateur devra privilégier la réactivité physique aux tâches de rendu ou tout du moins ne pas rendre tributaires de celles-ci la simulation physique et le dialogue avec le logiciel externe ou l'automate. Les technologies multitâches permettent aujourd'hui de gérer cela de façon correcte.

L'amélioration de la vitesse de rendu pourra quand même être traitée dans le cas de géométries anormalement complexes (ce qui est parfois le cas lorsqu'on exporte des géométries depuis les logiciels de CAO) par la réduction de celles-ci (technologie de décimation de triangles) ou la suppression d'éléments n'apportant que peu d'intérêt à l'aspect visuel de la simulation (éléments de faible volume, vis et écrous de petite taille, etc.). Il conviendra dans ce domaine de trouver un bon compromis entre un aspect visuel de qualité et un affichage fluide du système simulé, au cas par cas, suivant la taille du système simulé et les performances du PC.

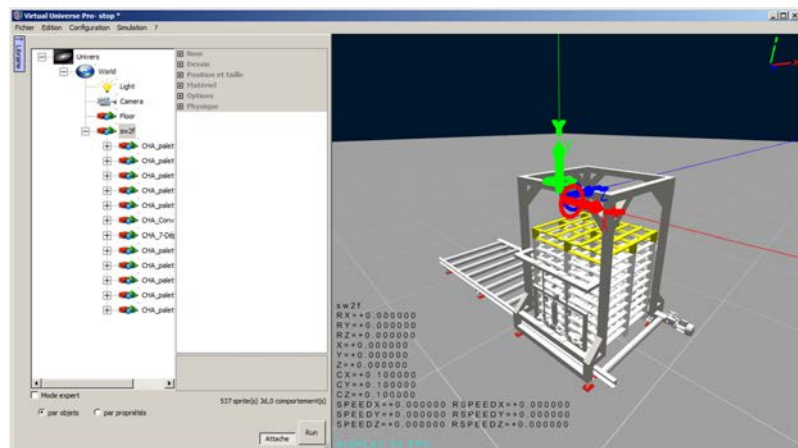
Notons également que la possibilité de « scinder » une simulation sur plusieurs PC permet de s'affranchir de la limitation de traitement d'un seul PC ; on peut alors envisager des projets de simulation sans réelle limitation de taille.

Simulation et monde réel

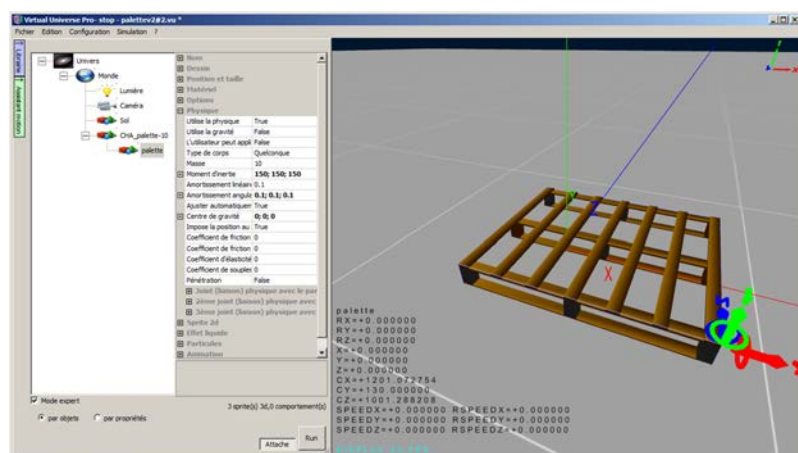
Sujet clé lorsqu'on parle de simulation : le delta entre simulation et système réel. La question du débutant en la matière est bien entendu : « Est-ce que la machine simulée fonctionne comme la machine réelle ? » Si la réponse doit être « oui » ou « non », ce sera forcément « non ». Mais, pour que cette réponse soit plus juste, reformulons la question de la façon suivante : « Quel est l'écart entre la simulation et la machine réelle ? » Car, évidemment, ce delta existe, et il n'est pas forcément facile de le quantifier sous tous ses aspects. Ce que l'on peut dire, c'est qu'il est important que le logiciel de simulation puisse permettre de le réduire lorsque ce sera primordial, mais également que la simulation puisse admettre un delta plus important si, en contrepartie, sa création peut en être facilitée. Le créateur de la simulation sera ensuite libre de fixer le curseur du réalisme de la simulation en fonction de la quantité de travail à fournir et des buts à atteindre. Prenons le cas d'un simple actionneur pneumatique. Une simulation basique consistera à définir un simple temps pour la sortie de ce vérin ; à l'opposé, la définition des formules physiques de calcul de pressions, de pertes de charges, de forces, etc., permettrait d'obtenir une simulation beaucoup plus proche de la réalité. Si le but est simplement, vis-à-vis d'un programme automate, de contrôler qu'un mouvement a bien été réalisé en un temps donné et d'avoir l'information de fin de course, la définition du temps de la course semble suffisante. Encore une fois, l'outil de simulation devra proposer et mettre à dispo-



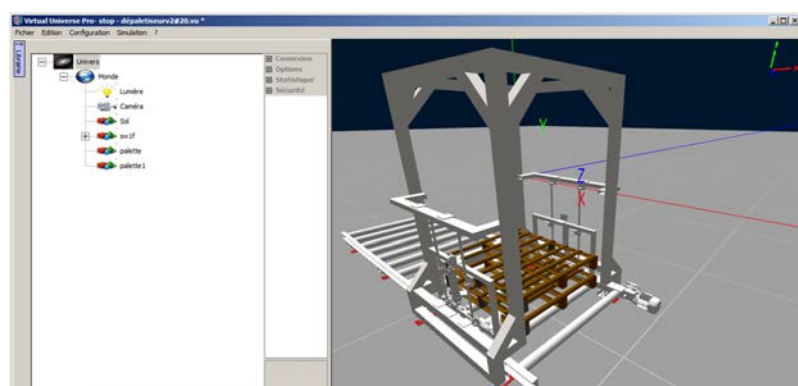
1 L'assemblage du dépiqueur créé dans SolidWorks



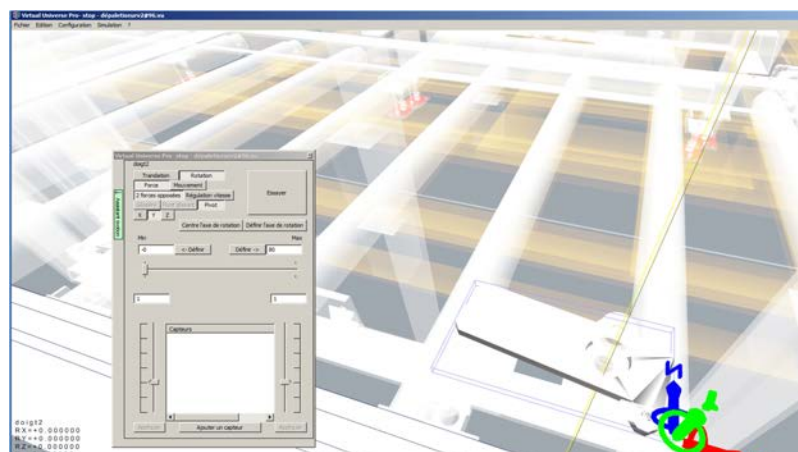
2 L'assemblage du dépileur importé dans Virtual Universe Pro



3 L'objet « palette » configuré dans Virtual Universe Pro



4 L'objet « palette » configuré, intégré et dupliqué



5 L'assistant de définition des mouvements

sition des outils de simulation aussi sophistiqués que possible, mais ne pas en imposer l'utilisation systématique. Ce choix devra être réservé au concepteur de la simulation.

Dans l'enseignement

Officialisée dans certains programmes comme celui du nouveau BTS CRSA (Conception et réalisation de systèmes automatisés), la simulation de systèmes automatisés offre de nombreux avantages : multiplication des systèmes à moindre coût (même s'il ne faut bien entendu pas céder au « tout-simulé » pour des raisons évidentes d'éloignement des contraintes du monde réel), gain de temps lors de la phase de mise au point sur le système réel, sécurisation du système réel et des personnes.

En BTS CRSA : la création d'une machine simulée avec Virtual Universe Pro

Nous partons d'un modèle numérique de dépalettiseur (thème de BTS CRSA du lycée Vauvenargues d'Aix-en-Provence) réalisé sous SolidWorks **1**.

La procédure de transfert de la maquette consiste à l'exporter depuis SolidWorks au format 3D XML (par un simple « enregistrer sous »), et à importer ce même fichier directement dans Virtual Universe Pro **2**.

Une fois importée la partie dessin de la machine, une phase de configuration des mouvements et autres comportements débute. Intéressons-nous tout d'abord aux pièces libres, les palettes. Après isolation d'une pièce (par commodité), ses propriétés physiques sont définies. L'aspect (ici la couleur) peut également être retouché **3**. L'objet ainsi paramétré est réintégré et dupliqué dans le modèle original **4**.

À l'aide d'assistants, les mouvements du modèle sont définis, par exemple la rotation des « doigts » et ceux du vérin qui anime ces mêmes doigts **5**. Les assistants offrent un choix complet de types de pilotage, pouvant aller jusqu'à une gestion d'axes avec profils d'accélération et de décélération. Ces mêmes assistants permettent de définir des capteurs.

Des contrôleurs virtuels intégrés au projet de Virtual Universe Pro permettent de créer tout ou partie du programme de pilotage du système, même si l'intérêt est bien entendu in fine de piloter le modèle virtuel avec le programme qui fera fonctionner la machine réelle **6**.

La phase de pilotage du modèle simulé par un automate ou un simulateur intégré à un logiciel constructeur reste simple : après sélection du type de l'automate, ici un Schneider M258 sous SoMachine **7**, on associe les pilotages de mouvements et les capteurs à des variables automatiques **8**.

En mode connecté à l'automate, on suivra l'évolution du programme dans l'outil constructeur et l'évolution de la machine simulée simultanément **9**.

En T^{le} STI2D :**les économies d'énergie
dans les systèmes de ventilation**

Selon l'expert Michel Sivo, se basant sur une étude du Gimelec (Groupement des industries de l'équipement électrique, du contrôle-commande et des services associés), ces systèmes représenteraient aujourd'hui un gisement d'économie d'énergie considérable : « Dans beaucoup d'installations, les ventilateurs travaillent à la vitesse nominale fixe. Des registres ou des inclineurs font varier le débit de l'air. Ces registres ou inclineurs augmentent les pertes de charge et deviennent la source de gaspillage d'énergie » (voir l'encadré « En ligne »).

Présentation

L'exemple choisi est un ventilateur de tunnel routier. Dans notre pôle d'étude, spécialité « énergie et environnement », il est matérialisé par un système didactique de type Ventelec 2, mais il aurait pu l'être par un banc machine équipé d'un moteur asynchrone et d'un frein à poudre. Le moteur est alimenté soit par un contacteur selon le principe du démarrage direct soit par un variateur de vitesse. Le but de cette étude est, en fin de projet, de comparer les consommations d'énergie mesurées de ces deux choix d'alimentation en énergie.

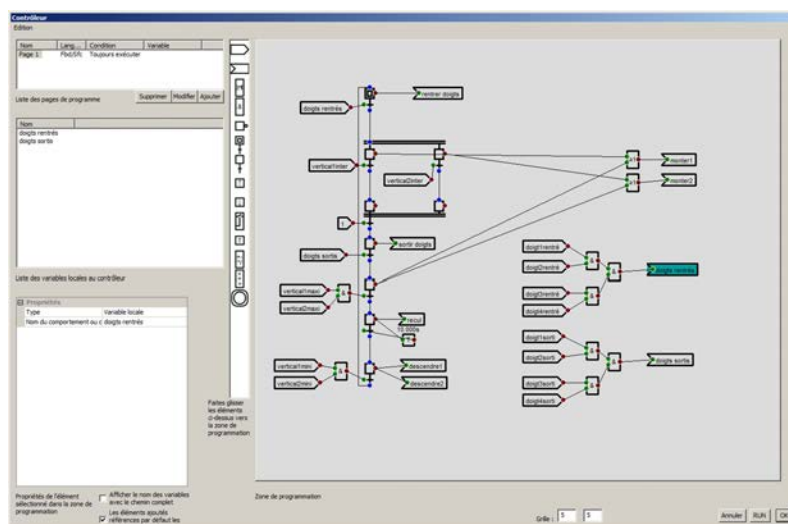
Pour donner plus de réalisme et aborder les contraintes d'automatisme imposées dans un tunnel réel, nous avons mis à disposition des élèves un tunnel virtuel issu d'une bibliothèque 3D disponible sur le site TurboSquid (www.turbosquid.com). Ces objets 3D, souvent développés initialement pour les jeux vidéo, peuvent être configurés et implantés dans Virtual Universe. Il est possible de piloter par une variable la demande en véhicules à l'intérieur du tunnel, de visualiser le flux de véhicules entrants et de connaître en temps réel par une autre variable le nombre de véhicules dans le tunnel. La vitesse de rotation des pâles du ventilateur visible à l'écran se modifie alors en fonction des contraintes imposées au moteur par l'automatisme. L'élève peut également se déplacer virtuellement dans le tunnel et modifier son angle de vision **10**.

Dans une première approche, le débit d'air dépend du nombre de véhicules présents dans le tunnel. Il est possible ensuite, dans une étude complémentaire, de prendre en considération le taux de monoxyde de carbone mesuré dans le tunnel **11**.

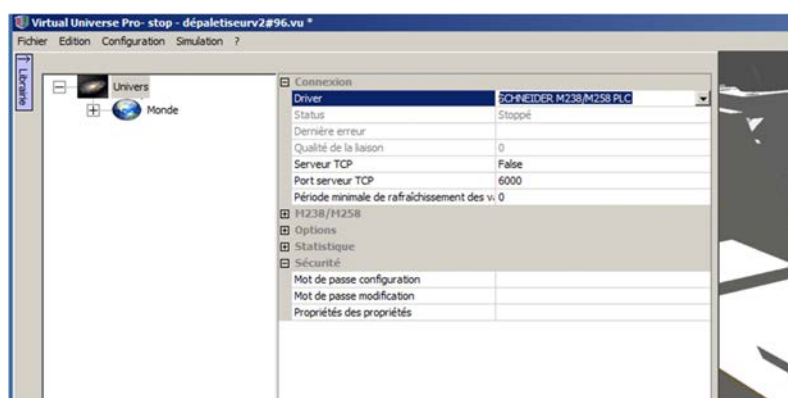
L'aspect matériel

Tous les composants d'automatisme communiquent avec un automate programmable (S7-1200) via un réseau informatique **12**.

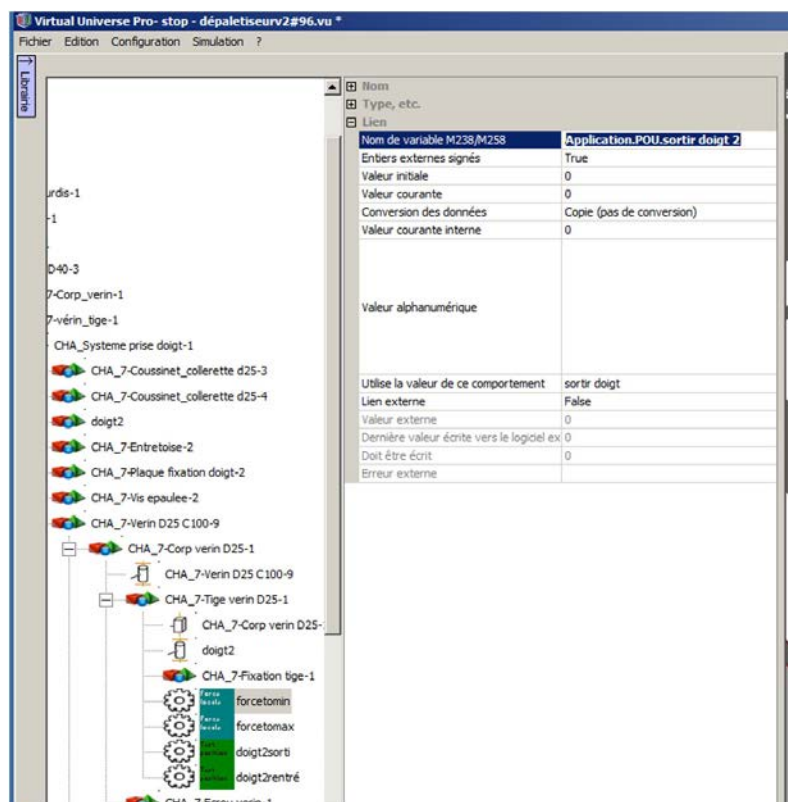
La console de dialogue (KTP600) permet de saisir le nombre de véhicules demandés à l'intérieur du tunnel, de mettre en marche ou à l'arrêt le système. Elle supervise également l'évolution du nombre de véhicules en fonction du temps.



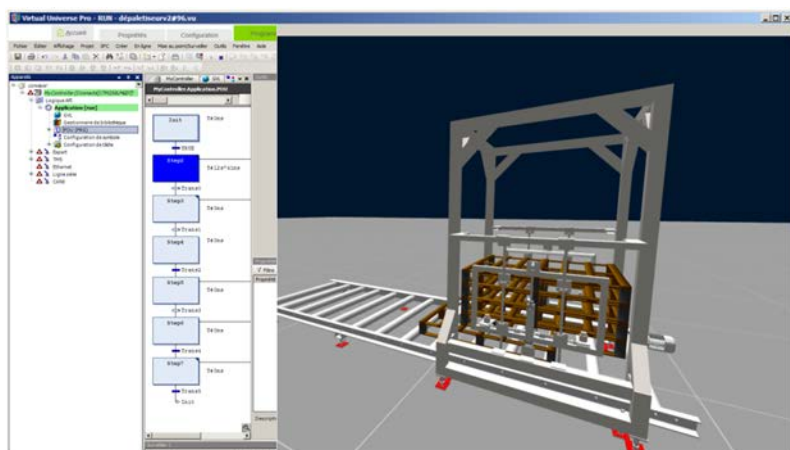
6 Le programme réalisé dans un contrôleur virtuel de Virtual Universe Pro



7 La sélection et la configuration du driver de communication



8 La définition d'un lien entre Virtual Universe Pro et une variable de l'automate Schneider Electric M258



9 Le pilotage du modèle virtuel par l'automate Schneider Electric M258



10 La simulation 3D du tunnel

Nombre de véhicules	Débit en % du débit max
de 0 à 9	20 %
de 10 à 19	40 %
de 20 à 29	60 %
de 30 à 39	80 %
40 et plus	100 %

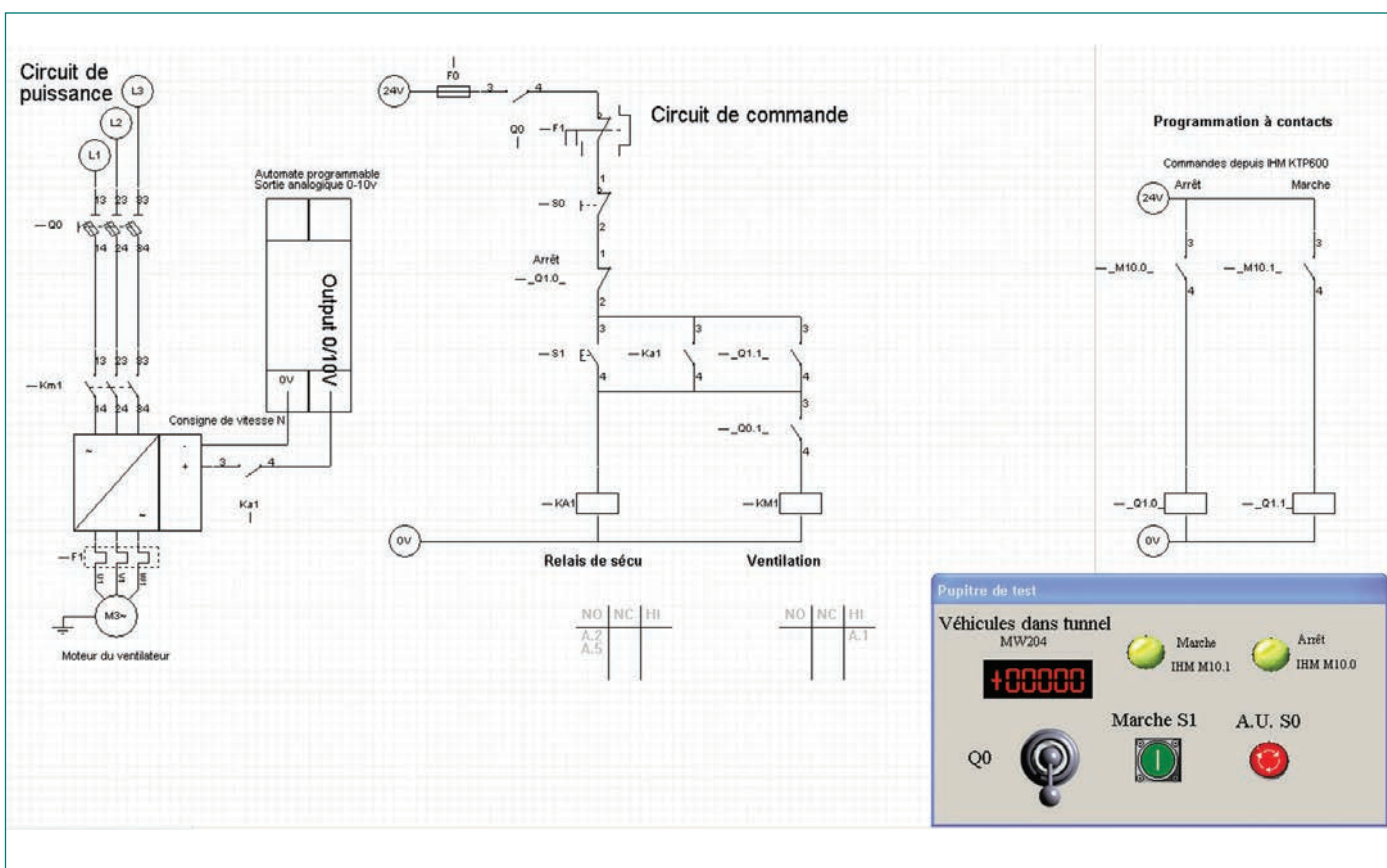
11 Le débit d'air en fonction du nombre de véhicules présents dans le tunnel



12 La structure matérielle

Table de variables standard	
Nom	Type de données - Adresse
Vitesse ventilateur	Word %MW200
Véhicules demandés dans tunnel	Word %MW202
Véhicules présents dans tunnel	Word %MW204
Sortie analogique automatique	Word %QW300
Bouton Arrêt KTP600	Bool %M10.0
Bouton Marche KTP600	Bool %M10.1
Entrée info relais Sécu KA1	Bool %I1.0
Validation IEM1	Bool %Q0.1
Sortie Arrêt depuis KTP600	Bool %Q1.0
Sortie Marche depuis KTP600	Bool %Q1.1
Taux CO	Word %MW206

14 La table des variables dans TIA Portal



13 La supervision des schémas électriques

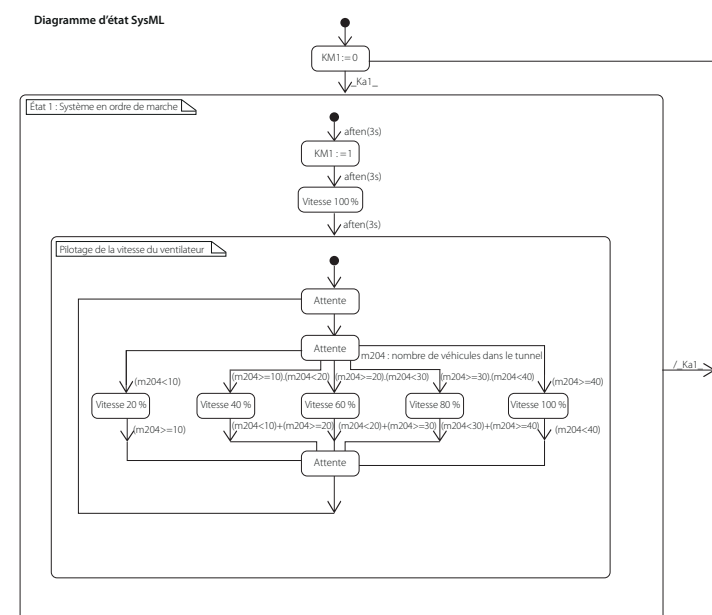
Driver S7 net pour AUTOMGEN 1.001

Paramètres de communication

Adresse IP : 192.168.1.11 Position UC : 0

Type	Variable S7	Variable	Mode	Période
Ecriture vers l'automate S7	%MW200	m200	Toujours échanger	0
Lecture depuis l'automate S7	%MW202	m202	Toujours échanger	0
Lecture depuis l'automate S7	%MW204	m204	Toujours échanger	0
Lecture depuis l'automate S7	%MW206	m206	Toujours échanger	0
Lecture depuis l'automate S7	%I1.0	i10	Toujours échanger	0
Ecriture vers l'automate S7	%Q1.0	o1	Toujours échanger	0
Ecriture vers l'automate S7	%QV300	m300	Toujours échanger	0
Ecriture vers l'automate S7	%Q1.0	o10	Toujours échanger	0
Ecriture vers l'automate S7	%Q1.1	o11	Toujours échanger	0
Lecture depuis l'automate S7	%M10.0	i100	Toujours échanger	0

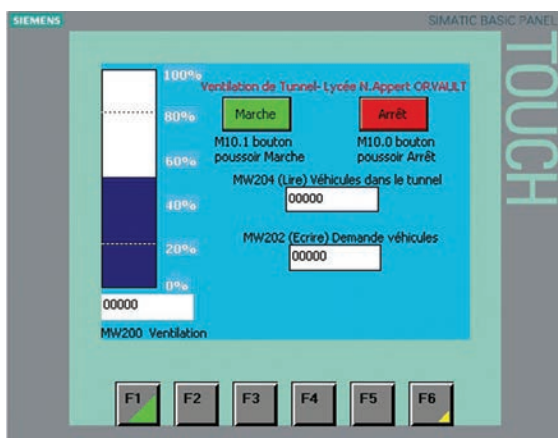
15 La configuration des liens entre Automgen8 et l'automate S7-1200



16 Le programme écrit en diagramme d'état SysML



17 L'ajout du driver de communication dans Automgen



18 L'IHM créée dans TIA Portal

L'aspect logiciel :

programmation et supervision

La supervision des schémas électriques et la programmation en langage SysML de l'automate sont réalisées avec le logiciel Automgen8 intégré comme Virtual Universe à SLI, la suite logicielle IRAI **13**.

Pour les tests du schéma électrique et la mise au point de la programmation SysML, l'élève n'a pas besoin de connexion avec le réseau informatique. Un « pupitre de test » disponible sur Automgen8 permet de simuler les commandes utiles. Pendant la compilation, le logiciel n'étant pas en communication avec l'automate S7-1200, les boutons Marche IHM et Arrêt IHM présents sur le pupitre de test sont activés. Le nombre de véhicules présents dans le tunnel peut directement être saisi à ce niveau pour une mise au point plus rapide au lycée ou pour des essais réalisés par les élèves chez eux en préparation des séances de projet (l'accès gratuit à un serveur de licences pour les étudiants est proposé par la société IRAI sur simple demande).

Grâce à la simple concordance de la table de variables du logiciel TIA Portal Basic **14**, fourni avec l'automate Siemens, et de celle d'Automgen8 **15**, la programmation est réalisée sous forme d'un diagramme d'état SysML **16**.

La rigueur pour la configuration du réseau informatique et pour la déclaration des tables de variables est de mise, comme pour tout projet de programmation. Ensuite, le travail devient très ludique **17**. TIA Portal Basic permet de configurer le réseau, de paramétrer les variables et de configurer l'IHM par glissés-dépôts à partir des objets de supervision présents dans sa bibliothèque **18**.

En fin de projet, les élèves peuvent suivre sur ordinateur l'évolution dynamique du schéma électrique et du diagramme SysML synchronisé avec les composants réels. L'exercice de lecture d'un schéma électrique, peu pratiqué dans les nouvelles formations, devient un véritable jeu.

La logique des diagrammes d'état SysML est plus facile à comprendre grâce à l'animation en temps réel.

Les résultats

Des simulations effectuées en début de projet pour un ventilateur de tunnel réel avec le progiciel SinaSave **19** de la société Siemens montrent l'intérêt de la solution à vitesse variable par rapport à celle utilisant le réglage du débit d'air par étranglement.

Les mesures comparatives de puissance effectuées en fin de projet sur la maquette, d'une puissance de 2,2 kW, bien inférieure à celle du ventilateur réel du tunnel de Toulon (250 kW) exploité dans la simulation SinaSave, confirment par « homothétie » l'intérêt de la solution basée sur la variation de vitesse du moteur de ventilation. Ces mesures sont réalisées à partir d'appareils classiques ou de la centrale de

mesure PAC3200 reliée au réseau informatique, qui permet de récupérer les informations au niveau de l'IHM et de visualiser l'évolution de la puissance en fonction du temps.

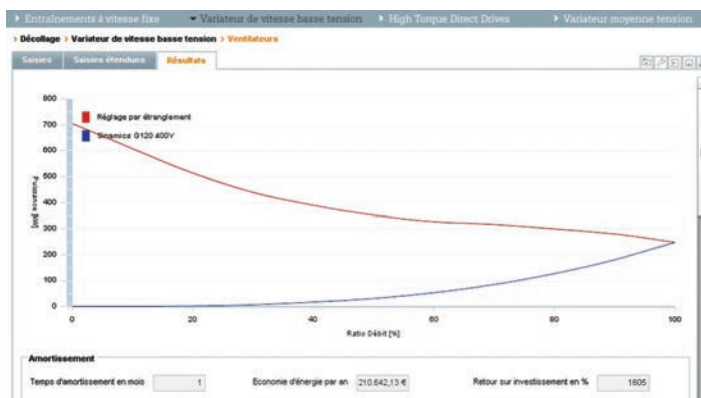
Conclusion

L'avantage essentiel de la suite logicielle IRAI est qu'elle permet de piloter un matériel via un automate programmable, comme ici le S7-1200, ou de faire du contrôle-commande, à partir d'un boîtier d'acquisition USB-6009 National Instruments par exemple, en fonction des contraintes du projet, et de ne pas en rester au stade de la seule simulation.

Il est possible de prélever des informations physiques (vitesse, puissance, intensité...) via des capteurs sur une maquette ou un système et de les faire agir sur des modélisations : schéma électrique, diagramme d'état SysML, schéma-bloc...

Après cette première expérience en projet de terminale spécialité « énergie et environnement », nous avons constaté que certains élèves se passionnaient pour cette manière de travailler. Leurs projets d'orientation postbac ont même été influencés par l'utilisation de ces nouveaux outils.

Le matériel mis en œuvre cette année permettra de réaliser très facilement d'autres projets sur d'autres thèmes les années suivantes. ■



19 Le logiciel SinaSave

En ligne

SIVO (M.), « Moteurs de ventilation : gisement d'économies », sur le site Xpair.com :

http://conseils.xpair.com/consulter_parole_expert/moteurs-de-ventilation-gisement-economies.htm

Téléchargement des ressources disponibles sur le site d'IRAI :

www.iraicom/techno/

SLI, La suite logiciel IRAI :

<http://www.iraifrance.com/#!sli/ci1a>

Le logiciel SinaSave de Siemens :

<https://eb.automation.siemens.com/mall/fr/fr/Catalog/Products/10041814>

Le logiciel Simatic Step 7 Basic (TIA Portal) :

www.industry.siemens.com/topics/global/fr/tia-portal/logiciel-controller-tia-portal/simatic-step7-basic-tia-portal/pages/default.aspx