



Département EEA - Faculté Sciences et Ingénierie

Master STS - EEAS

Électronique Énergie Automatique Signal

Parcours ISTR-RODECO

Systèmes à événements discrets, modélisation et analyse

Année 2020-2021

Travaux pratiques

Table des matières

Déroulement des séances de travaux pratiques de SED	1
1 Commande d'un ascenseur (Automates à états finis)	3
1.1 Objectif de la manipulation	3
1.2 Prérequis	3
1.3 Matériel utilisé	3
1.4 Travail à effectuer	4
2 Commande d'un banc de contrôle industriel (Automates à états finis)	6
2.1 Objectif de la manipulation	6
2.2 Prérequis	6
2.3 Matériel utilisé	6
2.4 Travail à réaliser	8
3 Commande d'un système de transport (Réseaux de Petri)	11
3.1 Objectif de la manipulation	11
3.2 Prérequis	11
3.3 Système commandé	11
3.4 Travail à effectuer	12
4 Commande d'une station de tri robotisée (Réseaux de Petri)	14
4.1 Objectif de la manipulation	14
4.2 Prérequis	14
4.3 Matériel utilisé	14
4.4 Travail à effectuer	15
Annexe de travaux pratiques de SED	17
1 Où récupérer les fichiers mis en partage sur le réseau?	17
2 Où récupérer le logiciel SEDMA?	17
3 Synthèse de commande et génération de code avec SEDMA	17

Déroulement des séances de travaux pratiques SED

Affectation des binômes sur les manipulations de TP

La séquence des TPs est : **Ascenseur** → **BCI** → **Trains** → **Robots**

Les étudiants sont répartis en trois groupes. La séance de TP qui sera indiquée aux binômes sera celle à préparer pour la **première séance de TP**.

Déroulement des TP

Pour ces travaux pratiques, l'outil SEDMA vous est fourni et cache tous les aspects de mise en œuvre de vos modèles de commande. Ces aspects seront traités au second semestre dans le module "Techniques de mises en œuvre de Systèmes à événements discrets". Ce logiciel résulte du travail de l'équipe pédagogique et de la participation de deux groupes d'étudiants de votre master pendant leur module "Initiation à la Recherche et Projet". N'hésitez pas à nous signaler les bugs que vous pourriez constater.

Pour tous les TP de ce module, les parties manipulatoires seront donc constituées des tests et mises au point de vos modèles. Afin ne pas perdre de temps pendant la séance, vous devrez élaborer les modèles du procédé et des objectifs avant chaque TP. Il est rappelé que SEDMA est disponible sur Moodle (section "Le logiciel SEDMA"). Il est d'ailleurs conseillé de le télécharger chaque fois que vous l'utilisez, car il s'agit d'un outil qui évolue très fréquemment.

La documentation de SEDMA est accessible après avoir lancé le logiciel par un clic droit sur la fenêtre de gauche (fond blanc) puis en sélectionnant "Manuel de l'utilisateur".

Vos encadrants en travaux pratiques ne répondront à aucune question sur l'utilisation basique du logiciel SEDMA.

Nous attendons qu'à votre arrivée en séance de TP, vous soyez capable d'utiliser toutes les fonctionnalités de SEDMA et que vous disposiez de modèles déjà établis. Nous vérifierons à chaque séance le résultat de la préparation que vous aurez menée au préalable et ceci comptera significativement dans la note de contrôle de TP.

Évaluation des TP

Compte-rendu écrit

Chaque binôme devra rédiger en LaTeX deux comptes-rendus écrits, l'un correspondant à une manipulation utilisant le formalisme réseaux de Petri et l'autre à une manipulation utilisant le formalisme des automates/langages.

Vous trouverez sur Moodle (section "Travaux Pratiques") un fichier de base LaTeX que vous complèterez pour rédiger votre compte-rendu. Ne pas modifier la structure, la page de garde et les sections/sous-sections.

Compte-rendu oral

Pour l'évaluation, un sujet automate ou réseau de Petri sera tiré au sort parmi les deux comptes-rendus préparés. Chaque étudiant devra répondre individuellement aux questions d'un enseignant au cours d'une entrevue dont la durée ne dépassera pas 20mn.

Les questions porteront sur cette manipulation, mais également sur les éléments théoriques développés en cours et en TD.

Le compte-rendu qui n'a pas été tiré au sort pour l'oral sera déposé sur Moodle avant le dimanche 20 décembre 2020 à 23h59, **délai de rigueur**, au format pdf en donnant **impérativement** le nom suivant au fichier déposé : "nom_etudiant-crTP.pdf".

Rien ne vous empêche de retravailler le compte-rendu à déposer après votre passage à l'oral pour tenir compte d'éventuelles remarques qui pourraient vous être faites.

Cet oral aura lieu le mercredi 16 décembre 2020 après-midi. L'affichage d'une liste des horaires nominatifs de passage sera déposé sur Moodle.

Pour toute question : utiliser exclusivement le forum moodle du cours EMEAR1G1.

Commande d'un ascenseur (Automates à états finis)

1.1 Objectif de la manipulation

L'objectif de cette manipulation est de commander grâce au formalisme automate, un procédé typique des systèmes à événements discrets qu'est l'ascenseur grâce à la technique de synthèse de commande vue en cours et en TD. La commande de ce système requiert de prendre en compte de nombreuses contraintes de fonctionnement. Ainsi, nous vous proposons de découper proprement ces contraintes en différents automates que vous associerez à vos modèles d'objectifs grâce à la composition parallèle. Les cahiers de charges décrivant les objectifs vont spécifier des comportements de plus en plus complexes.

1.2 Prérequis

- Automates à états finis : construction, opérations entre automates.
- Méthodes de synthèse de commande vue en cours et en TD.

1.3 Matériel utilisé

1.3.1 Système commandé

Il s'agit d'une maquette d'ascenseur à quatre étages, entraînée par deux actionneurs **M** pour la montée et **D** pour la descente. La position de la cabine est repérée par quatre capteurs d'étage **et1**, **et2**, **et3** et **et4** (avec **eti** = 1 cabine à l'étage *i*). A l'intérieur de la cabine, quatre poussoirs d'appel **ap1**, **ap2**, **ap3** et **ap4** permettent aux utilisateurs de demander le déplacement de la cabine. Un dispositif de sécurité coupe le mouvement en cours si la cabine descend au dessous de l'étage 1 ou si elle monte au dessus de l'étage 4. Six poussoirs palier **p4d**, **p3m**, **p3d**, **p2m**, **p2d**, **p1m** (**pxd** et **pxm** pour palierxMontee ou palierxDescente) permettent à l'utilisateur d'appeler la cabine à partir du palier de l'étage *x*. Un contact de porte ouverte **po** passe à "1" dès qu'une des portes donnant accès à l'ascenseur est ouverte. Deux interrupteurs **s** (pour stop) et **r** (pour reprise) sont destinés, en particulier, à la gestion des urgences sur cette maquette.

La maquette dispose par ailleurs de deux afficheurs sept segments pouvant servir à afficher des informations au cours du fonctionnement (numéro de l'état actif par exemple). Les capteurs, interrupteurs et boutons poussoirs utilisés dans cette manipulation sont représentés sur la figure 1.1.

1.3.2 Système de commande

Le système de commande est constitué d'une carte Altera UP1 (University Program) comportant deux circuits programmables : un CPLD et un FPGA. Pour des raisons de simplicité, transparentes dans ce type de manipulation, c'est le CPLD qui est utilisé comme cible d'implantation du système de commande. Une nappe de fils assure la connexion électrique entre la carte UP1 et la maquette. Le circuit dans lequel est implantée la commande à spé-



FIGURE 1.1 – Maquette d’ascenseur

cifier est programmé par l’intermédiaire d’un logiciel fourni par la société Altera (Quartus). Le langage d’entrée de cette spécification est VHDL.

1.4 Travail à effectuer

1.4.1 Modélisation du procédé

Déterminer les composants du procédé dont le comportement peut être indépendant des autres éléments. Vous représenterez l’ensemble des évolutions possibles de chaque composant par un automate à états finis. Le modèle complet du procédé sera obtenu en effectuant le produit parallèle des automates des différents éléments.

1.4.2 Méthodologie

Quatre étapes de modélisation vous sont proposées, chaque étape rendant la commande plus complexe que la précédente. Pour chacune d’entre elles, vous devrez :

- Proposer un modèle des objectifs de la commande ;
- Effectuer la synthèse de cette commande, c’est-à-dire calculer le produit parallèle des automates puis le restreindre aux trajectoires utiles pour les objectifs ;
- Vérifier que le modèle ainsi obtenu possède bien les deux propriétés essentielles : non bloquant et pas d’instabilité incontrôlable ;
- Tester et mettre au point le modèle de commande obtenu sur la maquette.

Toutes ces phases seront réalisées dans le logiciel SEDMA.

1.4.3 Contraintes de fonctionnement

Afin de simplifier l'obtention des modèles de commandes, il vous est proposé d'utiliser, si besoin, les contraintes supplémentaires suivantes (chaque contrainte est représentée par un automate) :

- On ne peut prendre en compte un appel qu'en face d'un étage.
- On ne peut s'arrêter que devant un étage.
- Un mouvement doit être provoqué par un appel et tout appel doit provoquer un mouvement.

Ces contraintes sont vues comme des restrictions du fonctionnement du modèle du procédé (comme des objectifs donc).

1.4.3.1 Cahier des charges 1 : fonctionnement en monte-charge

Dans ce premier fonctionnement, la cabine se déplace entre les étages 1 et 2 uniquement. Chaque changement d'étage est provoqué par un appui sur le bouton poussoir **ap1**.

A l'état initial, la cabine est à l'arrêt devant l'étage 1 en attente d'un appel.

1.4.3.2 Cahier des charges 2 : fonctionnement en ascenseur simple

Chaque appel **api** doit provoquer au moins un passage devant l'étage *i*. Tout appel doit provoquer au moins un passage devant un étage.

A l'état initial, la cabine est à l'arrêt devant l'étage 1 en attente d'un appel.

1.4.3.3 Cahier des charges 3 : prise en compte des arrêts d'urgence

Le système d'arrêt d'urgence est sensible à deux signaux **s** et **r**. L'appui sur **s** (stop) doit figer le mouvement de la cabine. Le retour au mode automatique n'est alors possible que lorsque l'opérateur appuie sur **r** (reprise).

1.4.3.4 Cahier des charges 4 : mémorisation des appels paliers

Les appels palier peuvent maintenant être pris en compte à n'importe quel moment. Un appel pour l'étage *x* est mémorisé dès que l'expression **pxd + pxm** devient "vraie". L'appel sera ensuite oublié dès que la porte de la cabine s'ouvre à l'étage *x*.

Commande d'un banc de contrôle industriel (Automates à états finis)

2.1 Objectif de la manipulation

Dans cette séance de travaux pratiques, on vous propose d'utiliser la théorie des langages pour développer une commande de la maquette de Banc de Contrôle Industriel (BCI). Cette maquette est une plateforme pédagogique automatisée permettant de réaliser des fonctions de tri, d'assemblage et de contrôle de pièces. La séance de TP portera plus précisément sur la fonction de contrôle et d'évacuation.

Vous aurez l'occasion de mettre en pratique la démarche systématique de modélisation vue lors du cours et des TD. Dans un premier temps, vous modéliserez le procédé seul, à partir de ses spécifications matérielles et d'une brève analyse de son fonctionnement. Ensuite, vous modéliserez les objectifs à réaliser, en vous basant sur les données des cahiers des charges proposés. Enfin, en fusionnant ces deux modèles et en simplifiant le résultat, vous obtiendrez un modèle de la commande désirée.

2.2 Prérequis

- Bases de la théorie du langage.
- Méthodes de synthèse des automates à états finis.

2.3 Matériel utilisé

2.3.1 Système commandé

La fonction principale de la maquette de BCI est de trier deux types de pièces arrivant de manière désordonnée sur un *convoyeur*, de les assembler et de contrôler le résultat obtenu sur un *tapis roulant*. Voici les pièces que nous considérerons :

- Les pièces hautes grises, embases métalliques, que l'on assimilera à des *bouteilles*.
- Les pièces basses blanches, anneaux en matière plastique que l'on assimilera à des *bouchons*.
- Lorsqu'un anneau est correctement emboîté sur une embase, on appellera ceci un *assemblage*.

Dans cette séance, vous ne travaillerez pas explicitement sur les fonctions de tri et d'assemblage. Le guidage des bouteilles de la *zone de Tri* vers la *zone d'Assemblage* est réalisé mécaniquement, de même que l'assemblage d'un bouchon et d'une bouteille dans la *zone d'Assemblage*. Les capteurs et actionneurs qui leur sont liés sont les suivants :

- **c1**, détecteur photo-électrique de proximité, détecte toutes les pièces dans la *zone de Tri*.
- **c2**, détecteur photo-électrique de proximité, détecte uniquement les pièces hautes dans la *zone de Tri*.
- **c3**, détecteur photo-électrique de proximité, détecte un bouchon dans la *zone d'Assemblage*.
- **A1**, actionneur linéaire à solénoïde, permet d'éjecter une pièce de la *zone de Tri* vers

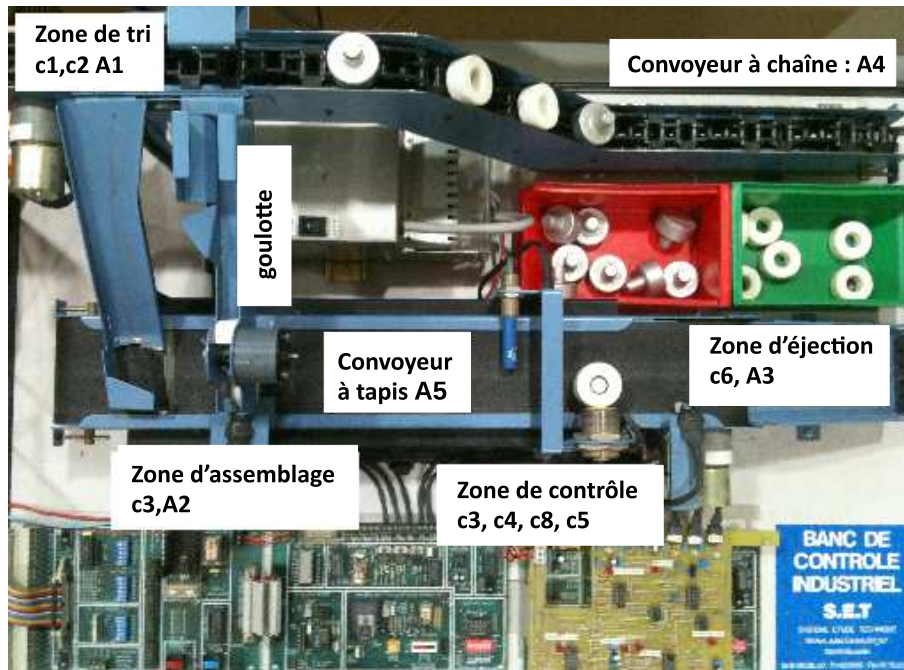


FIGURE 2.1 – Banc de Contrôle Industriel

la *goulotte*.

- A2, actionneur rotatif à solénoïde, permet d'admettre un bouchon de la *goulotte* dans la *zone d'Assemblage*.
- A4 est le moteur du convoyeur à chaîne apportant des pièces dans la *zone de Tri*.

En ce qui concerne la fonction de contrôle et d'éjection, voici les capteurs et actionneurs correspondants :

- c4, détecteur photo-électrique par barrage, détecte toutes les pièces arrivant vers la *zone de Contrôle*.
- c5, détecteur photo-électrique de proximité, détecte toutes les pièces dans la *zone de Contrôle*.
- c6, détecteur photo-électrique de proximité, détecte toutes les pièces dans la *zone d'Éjection*.
- c7, détecteur de proximité inductif, détecte les bouteilles, métalliques, arrivant vers la *zone de Contrôle*.
- c8, détecteur de proximité capacitif placé en hauteur, détecte les pièces hautes dans la *zone de Contrôle*.
- A3, actionneur linéaire à solénoïde, permet d'éjecter une pièce de la *zone d'Éjection*.
- A5 est le moteur du convoyeur à bande (tapis roulant) apportant des pièces de la *zone d'Assemblage* vers la *zone de Contrôle* et la *zone d'Éjection*.

2.3.2 Système de commande

Ce matériel sera commandé par un Automate Programmable Industriel (API) TSX. Il est relié aux actionneurs et capteurs de la maquette. Dans cette manipulation, le logiciel UNITY sera utilisé uniquement pour générer l'application (à partir du code produit par le logiciel SEDMA) puis la téléverser sur l'automate pour exécution et tests. L'ensemble des opérations de saisie, d'analyse, de composition, de simplification des modèles et de la production du code sera faite avec le logiciel SEDMA.

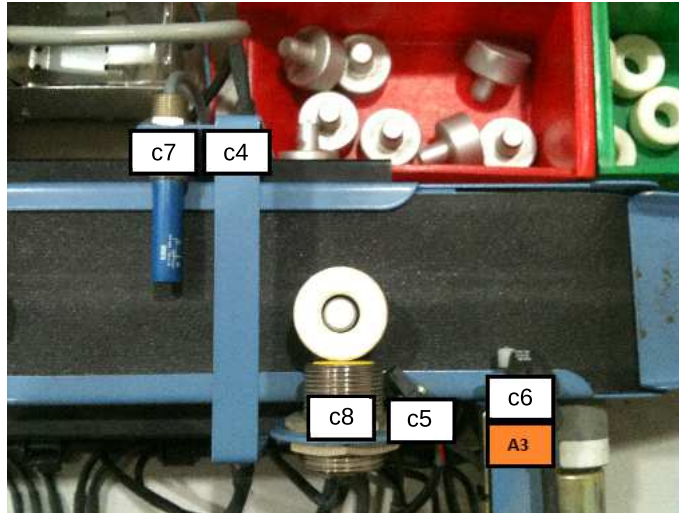


FIGURE 2.2 – Zones de Contrôle et d'Éjection

2.4 Travail à réaliser

Posons les hypothèses suivantes :

- Au maximum une pièce est présente entre les capteurs c7 et c6.
- Le convoyeur à bande A5 est en fonctionnement permanent (hors synthèse donc)

2.4.1 Analyse et modélisation du procédé

Comme les deux capteurs c5 et c8 sont de nature physique différente, les zones respectives dans laquelle ils peuvent détecter une pièce ne sont pas identiques. Cependant, la conception de la maquette fait que ces deux zones se chevauchent.

- Représentez sur un schéma les quatre dispositions possibles des deux zones de détection.
- Identifiez celle qui existe réellement sur la maquette proposée.

Considérons à présent les deux hypothèses énoncées plus haut :

- Écrivez les séquences possibles pour les pièces *bouchon*, *bouteille* et *assemblage*, en prenant en compte les quatre dispositions possibles des capteurs c5 et c8. Vous représenterez chaque changement d'état possible de capteur ou d'actionneur.

On suppose enfin que des pièces arrivent successivement et donc que le procédé effectue une boucle.

- Trouvez l'expression régulière correspondante.
- Tracez le graphe de l'automate équivalent.
- Pour des états finaux correspondant à l'éjection ou l'admission (non éjection) d'une pièce reconnue (*bouchon*, *bouteille* ou *assemblage*) et un état initial correspondant à un tapis vide, quel est le langage marqué reconnu par l'automate ?
- Pour des états finaux correspondant uniquement à l'admission (non éjection) d'une pièce reconnue (*bouchon*, *bouteille* ou *assemblage*) et un état initial correspondant à un tapis vide, quel est le langage marqué reconnu par l'automate ?

2.4.2 Modélisation des objectifs

2.4.2.1 Cahier des charges 1

Réalisez sous la forme d'un automate la modélisation des objectifs correspondant au cahier des charges suivant :

- On admet tous les *assemblages*.
- On éjecte toutes les pièces qui ne sont pas des *assemblages*.
- Le fonctionnement est répétitif.

2.4.2.2 Cahier des charges 2

Réalisez sous la forme d'un automate la modélisation des objectifs correspondant au cahier des charges suivant :

- On admet un *bouchon* puis une *bouteille*.
- On éjecte toutes les pièces qui ne correspondent pas à cette séquence.
- Le fonctionnement est répétitif.

2.4.3 Synthèse de la commande

Pour chacun des deux cahiers des charges, calculez le produit parallèle de l'automate des objectifs de commande avec celui du procédé.

Modifier chacun des deux automates obtenus lors du produit parallèle pour n'obtenir que les trajectoires utiles pour les objectifs.

2.4.4 Tests et mise au point

Réaliser en séance de TP les tests et mise au point des modèles obtenus. Lors du premier test de commande, demandez l'aide d'un enseignant.

2.4.5 Nouvelles hypothèses de modélisation

On vous propose maintenant d'étudier le cas où la première hypothèse (au maximum une pièce est présente entre les capteurs c7 et c6) n'est pas vérifiée.

Posons de nouvelles hypothèses remplaçant celle devenue caduque :

- Au maximum deux pièces sont présentes entre les capteurs c7 et c6.
- Au maximum une pièce est présente devant le capteur c7.
- Au maximum une pièce est présente devant le capteur c4.
- Au maximum une pièce est présente devant les capteurs c5 ou c8.
- Au maximum une pièce est présente devant le capteur c6.
- Les pièces avancent à la même vitesse sur le tapis et ne peuvent donc pas se doubler.

2.4.5.1 Analyse

Dans le travail demandé précédemment, vous avez identifié trois mots correspondants aux pièces *bouchon*, *bouteille* et *assemblage* passant devant les capteurs c7, c4, c5 et c8. Découpez les trois mots *bouchon*, *bouteille* et *assemblage* en trois syllabes :

- S_1 (signature laissée devant le capteur c7),
- S_2 (signature laissée devant le capteur c4),
- S_3 (signature laissée devant les capteurs c5 et c8).
- Identifiez aussi deux mots correspondant aux actions *éjecter la pièce* et *admettre la pièce*. Ils forment la quatrième syllabe S_4 .

Lorsque deux pièces sont présentes entre les capteurs **c7** et **c6**, on peut considérer que deux mots sont en train d'être lus. Les quatre autres hypothèses proposées permettent en réalité de ne pas mélanger ces deux mots : une lettre lue ne peut correspondre qu'à une unique syllabe d'un unique mot.

- Dénombez et identifiez à tout instant les situations de lecture de syllabe en cours.
- En appelant l_i une des lettres composant la syllabe S_i et f_i une lettre la finissant, tracez schématiquement l'évolution de ces situations de lecture en fonction des lettres lues.

2.4.5.2 Modélisation du procédé

La modélisation du procédé, dans ce cas, est loin d'être simple et fera intervenir plusieurs dizaines d'états. Il vous est uniquement demandé d'esquisser l'automate correspondant, soyez méthodiques et patients.

Commande d'un système de transport (Réseaux de Petri)

3.1 Objectif de la manipulation

L'objectif de cette étude est de développer une commande du système d'approvisionnement en matières premières constitué de wagonnets navigant entre des postes de chargement et de déchargement. La spécification de ce système de commande sera effectuée par réseau de Petri. Chaque cahier des charges vous demandera de réaliser une commande de plus en plus complète ce qui vous obligera à prendre en compte des ressources partagées et donc des contraintes supplémentaires. Les bonnes propriétés des réseaux de commande obtenus seront également analysées.

3.2 Prérequis

- Modélisation par des réseaux de Petri interprétés
- Composition de modèles réseaux de Petri par la méthode de fusion des transitions
- Restriction de la composition aux séquences réalisant les objectifs
- Etude des propriétés d'un réseau de Petri à partir d'une analyse structurale

3.3 Système commandé

Le système à commander est composé de 4 voies A, B, C et D reliées par 4 aiguillages et 2 croisement-aiguillages. La disposition des voies empruntées par les wagonnets est donnée sur la figure 3.1.

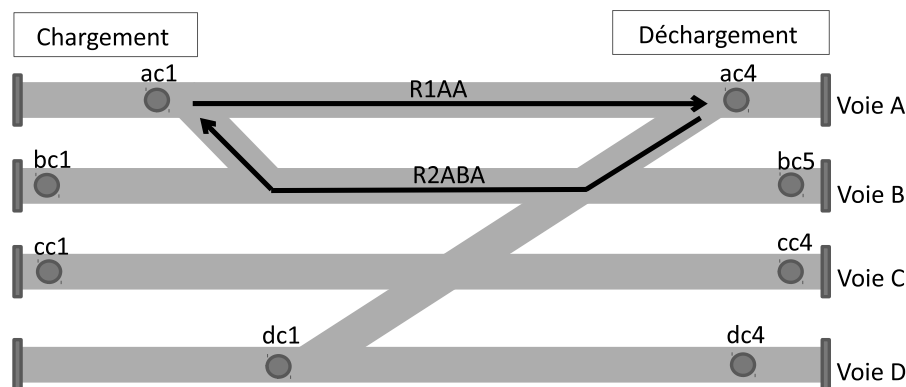


FIGURE 3.1 – Système de transport

Des postes de chargement se trouvent aux extrémités gauche des voies et des postes de déchargement se trouvent aux extrémités droite des voies. La position des wagonnets à ces postes est détectable par un ensemble de capteurs (ac1 à dc4).

Les wagonnets se déplacent d'un poste de chargement à un poste de déchargement à l'aide d'une commande $R1X$ où X représente les noms des voies empruntées. Par exemple, deux commandes permettent qu'un wagonnet se déplace de ac1 en ac4, R1AA permet d'effectuer

un trajet direct uniquement sur la voie A et R1ABA emprunte également la voie B. Sur la maquette, il existe donc 12 trajets possibles pour se déplacer d'un poste de chargement à un poste de déchargement : R1AA, R1ABA, R1AB, R1BB, R1BA, R1CC, R1CB, R1CBA, R1DD, R1DC, R1DCB et R1DCBA.

Les wagonnets se déplacent d'un poste de déchargement à un poste de chargement à l'aide d'une commande R2Y où Y représente les noms des voies empruntées pour le trajet de retour. Par exemple, la commande R2BCD permet le déplacement d'un wagonnet de bc5 en dc1. Il existe également 12 trajets possibles pour le retour d'un wagonnet à un poste de chargement : R2AA, R2ABA, R2AB, R2ABC, R2ABCD, R2BB, R2BA, R2BC, R2BCD, R2CC, R2CD et R2DD.

Un interrupteur bp1 est également disponible pour commander le système de transport.

3.4 Travail à effectuer

3.4.1 Cahier des charges et méthodologie

Durant les différentes phases de la manipulation, nous nous intéresserons d'abord aux voies A et B, puis aux voies A, B et C. Le but est de commander des wagonnets pour acheminer des produits d'une zone de chargement vers une de zone de déchargement. Après déchargement, chaque wagonnet doit revenir à un poste de chargement (pas forcément celui de départ). Le départ de chaque cycle de chargement/déchargement/retour en poste de chargement est déclenché par l'utilisateur en appuyant le bouton poussoir bp1.

La synthèse de la commande désirée est décomposée pour être réalisée progressivement, chaque étape s'appuie sur la précédente. Pour chacune de ces étapes, vous devrez :

- Proposer un modèle du procédé afin de représenter toutes les évolutions potentielles du système ;
- Proposer un modèle des objectifs de la commande ;
- Effectuer la synthèse de cette commande, c'est-à-dire réaliser la fusion des modèles du procédé et des objectifs sans oublier de le restreindre aux trajectoires utiles pour les objectifs ;
- Calculer les invariants de places et les invariants de transitions du modèle de commande obtenu afin de conclure quant aux bonnes propriétés du réseau (vivant, non-bloquant, ré-initialisable).

3.4.2 Cahier des charges 1 : commande d'un wagonnet w1 sur les voies A et B

Un seul wagonnet est présent dans ce procédé et il ne peut naviguer que sur les voies A et B.

L'objectif est de diriger le wagonnet w1 initialement situé au poste de chargement de la voie A vers le poste de déchargement de la voie B et de le faire revenir ensuite au poste de chargement de la voie B. Le trajet emprunté par w1 sera laissé au choix du concepteur.

Modification du procédé à ne pas prendre en compte par la suite

On considère maintenant que le procédé est capable de mémoriser le nombre d'allers-retours (poste de chargement - poste de déchargement - poste de chargement).

- Modifier le modèle du procédé pour prendre en compte cette mémorisation.
- Quelles sont les conséquences de cette modification concernant les bonnes propriétés du réseau de Petri du procédé.
- Proposer un modèle d'objectif représentant un comportement cyclique du wagonnet.

3.4.3 Cahier des charges 2 : commande des wagonnets w1 et w2 sur les voies A et B

On considère maintenant qu'un second wagonnet w2 peut circuler sur les voies A et B dans le procédé. Afin d'éviter toute collision entre les deux wagonnets w1 et w2, il faut empêcher qu'ils empruntent simultanément les mêmes voies. Cela peut se modéliser grâce à des contraintes liées au partage des ressources.

- Le premier objectif est que les wagonnets w1 et w2, respectivement positionnés aux postes de chargement de la voie A et de la voie B, doivent effectuer un déchargement puis revenir un à poste de déchargement.
- Un second objectif demande maintenant à ce que les deux wagonnets échangent leur position initiale après avoir effectué le déchargement décrit dans le premier objectif.

3.4.4 Cahier des charges 3 : commande des wagonnets w1 et w2 sur les voies A, B et C

La voie C de la maquette est maintenant considérée dans le procédé.

L'objectif est que les wagonnets w1 et w2, respectivement positionnés aux postes de chargement de la voie A et de la voie C, doivent se déplacer en bc1 et ac1.

Commande d'une station de tri robotisée (Réseaux de Petri)

4.1 Objectif de la manipulation

Cette manipulation a pour but d'utiliser les réseaux de Petri afin de synthétiser une commande d'une station de tri robotisée en s'appuyant sur la méthode vue en cours et en TD. La commande à réaliser étant très complète car faisant appel à différents éléments indépendants, nous vous proposons de modéliser le procédé global par une approche modulaire de chaque élément du système. La commande globale sera alors réalisée en fusionnant les différentes commandes obtenues pour chacun de ses éléments. Vous rechercherez également les bonnes propriétés des modèles réseaux de Petri développés.

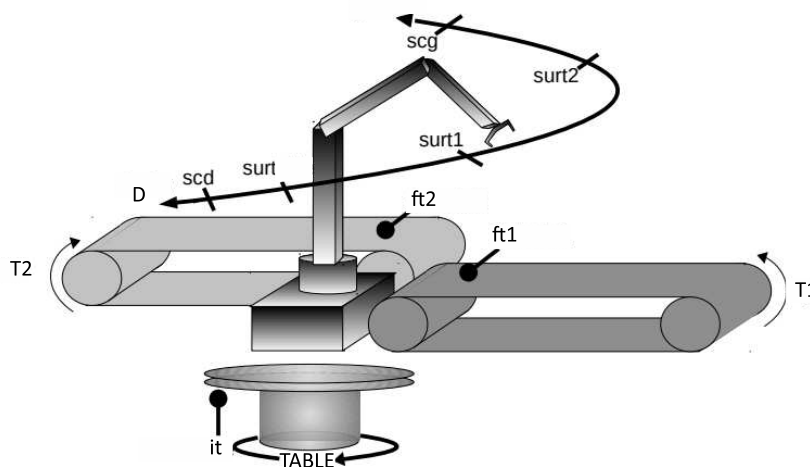
4.2 Prérequis

- Modélisation par des réseaux de Petri interprétés
- Composition de modèles réseaux de Petri par la méthode de fusion des transitions
- Restriction de la composition aux séquences réalisant les objectifs
- Etude des propriétés d'un réseau de Petri à partir d'une analyse structurale

4.3 Matériel utilisé

4.3.1 Système commandé

Le système à commander est composé d'un robot, de deux tapis convoyeurs, d'une table d'indexage et d'un poste d'emballage. Les éléments sont disposés de sorte que le robot puisse accéder aux tapis, à la table et au poste d'emballage.



Le fonctionnement de ces éléments est le suivant.

Le robot : son rôle, dans cette manipulation, consiste à déplacer des pièces entre les deux tapis, la table et le poste d'assemblage. A cet effet, il peut tourner sur sa base (actionneurs **G** pour gauche et **D** pour droite) entre deux capteurs de surcourse (**scd** et **scg**) entre lesquels quatre capteurs de position correspondant aux éléments sur lesquels le robot peut être amené à saisir ou déposer des pièces (**surt1**, **surt2**, **surt** (table d'indexage) et **suremb** (poste d'emballage)). La prise (resp. la pose) d'une pièce à la position où se trouve le robot est déclenchée par l'actionneur **PRISE** (resp. **POSE**), la fin de cette opération considérée comme atomique dans cette manipulation est signalée par le capteur **fprise** (resp. **fpose**).

Les tapis : permettent d'acheminer des pièces dans une direction unique matérialisée sur la figure par une flèche. L'avancement du tapis 1 (resp. tapis 2) est provoqué par l'actionneur **T1** (resp. **T2**). Un capteur de présence permet de détecter l'arrivée d'une pièce dans la zone accessible par le robot. Ce capteur **ft1** (resp. **ft2**) est électromécanique et fonctionne en logique négative (enclenché, le capteur est à 0) et c'est la pièce acheminée par le tapis qui l'enclenche. Pour cette raison, la position du robot (**surt1** ou **surt2**) correspond à la position de la pièce sur le tapis au moment où celle-ci relâche le capteur de présence (sans cette précaution, le robot heurterait le capteur en venant saisir la pièce).

La table : il s'agit d'une table circulaire pouvant tourner sur son axe (actionneur **TABLE**). Un capteur nommé **it** est enclenché tous les huitièmes de tour. Dans cette manipulation nous considérerons que cette table peut occuper huit positions différentes repérées par l'apparition du front montant sur le capteur **it**.

Pour les besoins de la manipulation, nous disposons également de sept interrupteurs (**a**, **b**, **c**, **d**, **e**, **f**, **g**) sur un boîtier d'interface standard, ainsi que 8 diodes électroluminescentes pouvant être allumées ou éteintes grâce aux actionneurs **led1** à **led8**.

4.3.2 Système de commande

Le système de commande est constitué d'un ordinateur PC fonctionnant sous linux RT. Une carte d'entrée/sortie spécialisée permet de commander le robot. Un logiciel est fourni avec cette carte pour engendrer les mouvements composés du robot (la prise et la pose d'une pièce).

4.4 Travail à effectuer

4.4.1 Cahier des charges et méthodologie

On souhaite obtenir le fonctionnement suivant : des pièces sont acheminées par les deux tapis et doivent être déposées sur la table circulaire qui effectue une rotation d'un quart de tour à chaque fois qu'une pièce est posée. La table a donc une capacité de 4 pièces. Un opérateur peut enlever une à une les pièces de la table circulaire en respectant l'ordre dans lequel elles ont été déposées sur la table. Chaque fois qu'il enlève une pièce de la table, l'opérateur le signale en appuyant sur le poussoir **f**.

Plutôt que d'aborder l'ensemble du problème de commande frontalement, nous vous proposons une démarche progressive de modélisation et de génération de la commande via le formalisme réseau de Petri. Cette démarche va tirer profit de la relative indépendance des éléments constituant cette maquette. Ainsi, pour chaque élément, vous devrez :

- Proposer un modèle du procédé de l'élément ;
- Proposer un modèle des objectifs de la commande ;

- Effectuer la synthèse de cette commande, c'est-à-dire réaliser la fusion des modèles du procédé et des objectifs sans oublier de le restreindre aux trajectoires utiles pour les objectifs.
- Effectuer une analyse structurelle du modèle de commande obtenu afin de rechercher en particulier les invariants de transitions correspondant au fonctionnement cyclique décrit par les objectifs de la commande ;
- Tester et mettre au point le modèle de commande obtenu sur la maquette.

N'oubliez pas les boutons poussoirs dans vos modèles de procédés.

4.4.2 Cahier des charges 1 : la commande des tapis

Objectif : l'appui sur un poussoir **a** démarre le tapis et dès qu'une pièce est détectée à la position de saisie par le robot, le tapis doit être arrêté. Le cycle est relancé dès que le poussoir **a** est à nouveau enclenché.

4.4.3 Cahier des charges 2 : la commande du robot

Concernant le procédé du robot, noter que la rotation de la base et les mouvements du bras (prise et pose) sont totalement indépendants. Mais vous n'envisagerez pas le cas où deux ordres contradictoires sont engagés simultanément (droite et gauche) et (prise et pose). Afin de simplifier le modèle du procédé, vous supposerez également que le robot ne peut arrêter son mouvement que lorsque sa position est repérée par un capteur.

Objectif 1 : la prise d'une pièce sur un des tapis est déclenchée par un appui sur le poussoir **b** (tapis 1) et **c** (tapis 2). L'autorisation de déposer la pièce sur la table est donnée par un appui sur le poussoir **d**. Les actions de prise et de pose d'une pièce doivent être effectuées lorsque le robot est à l'arrêt à la position correspondante.

Objectif 2 : modéliser l'alternance de la prise des pièces sur les tapis.

4.4.4 Cahier des charges 3 : la commande de la table d'indexage

Objectif : la pose d'une pièce est signalée par une action sur le poussoir **e**, la table doit effectuer un quart de tour immédiatement après le dépôt d'une pièce. Remarque : le dépôt d'une pièce sur la table ne peut se faire que si celle-ci est à l'arrêt.

4.4.5 Commande globale de l'ensemble de la maquette

1. Compte tenu des trois réseaux de Petri de commande que vous avez obtenus aux question précédentes, proposez une démarche pour obtenir le réseau de commande global dans lequel les actions sur les poussoirs **b**, **c**, **d** et **e** sont remplacés par des éléments structurels reliant les trois réseaux de Petri.
2. Montrer également par l'analyse structurelle que certaines propriétés liées au cahier des charges sont bien présentes dans ce modèle de commande. Il convient au préalable d'identifier ces propriétés et d'expliquer comment elles doivent se traduire dans le réseau de Petri.
3. Tester et mettre au point le programme de commande globale sur la maquette.

Annexe de travaux pratiques de SED

1 Où récupérer les fichiers mis en partage sur le réseau ?

Sous Windows, après exécution du script de connexion au serveur Electre, deux disques supplémentaires apparaîtront :

- **m1eea_xxx sur Electre** : il s'agit de votre espace de travail,
- **commun sur Electre** : il s'agit d'un dossier contenant tous les fichiers mis en partage, entre autres le sous dossier **M1_ISTR_SED**. Quatre sous dossiers existent, un pour chaque manipulation que vous avez à effectuer.

Sous Linux, ces fichiers mis en partage sont accessibles à partir de `home/partage/commun/M1_ISTR_SED`.

2 Où récupérer le logiciel SEDMA ?

Le logiciel SEDMA doit être récupéré via moodle, section "Le logiciel SEDMA".

Il est essentiel de l'avoir manipulé intensivement avant de venir en séance. Il est conseillé de le télécharger à chaque utilisation pour bénéficier de la version la plus stable.

3 Synthèse de commande et génération de code avec SEDMA

La saisie, l'analyse, la composition d'automates ou de réseaux de Petri se fait grâce au logiciel SEDMA. Ce logiciel est fourni avec une documentation complète (document pdf) qu'il convient d'avoir en permanence à portée de main. Cette documentation est accessible par un clic droit sur le fond blanc de la fenêtre de gauche de SEDMA (item "Manuel de l'utilisateur")

Ce logiciel est également capable de générer le code source des applications que vous définissez pendant la synthèse. La seule opération qu'il faudra effectuer ensuite correspond à la compilation, l'intégration dans un projet VHDL ou l'importation dans l'automate programmable industriel.

Voici les spécificités liées à chaque manipulation (maquettes + système de commande).

3.1 Commande d'un ascenseur

Le sous dossier **ASCENSEUR** contient l'ensemble des fichiers définissant un projet VHDL pour le logiciel QUARTUS (en fait un dossier **Ascenseur** contenant le fichier projet **Ascenseur.qpf**). Ce dossier doit être copié dans votre espace de travail (pas sur le bureau ou dans "Mes documents" !) mais sur votre compte sur le serveur "Electre". **Les fichiers présents dans ce dossier (Ascenseur) ne doivent pas être modifiés.**

L'ouverture du logiciel QUARTUS se fait en double cliquant sur le fichier "Ascenseur.qpf" (qpf = Quartus Project File).

Le générateur de code de SEDMA produit un fichier "commande.vhd" dans le dossier "CodeGenere" de votre projet SEDMA (voir la documentation pour plus de précisions) qu'il suffit de copier dans le dossier **Ascenseur** du projet Quartus. Il faut ensuite lancer la compilation du projet, puis transférer le code binaire obtenu sur le CPLD de la carte UP1.

Remarque : l'interrupteur E_0 (27) correspond à l'entrée `init`, l'interrupteur E_1 (28) à l'entrée `s` et l'interrupteur E_2 (29) à l'entrée `r`.

3.2 Commande d'un BCI

Aucun fichier de configuration n'est à rechercher sur le disque `commun` sur `Electre` pour cette manipulation.

La programmation de l'automate se fait grâce au logiciel "UNITY PRO" dont l'icône (une pyramide grise) est présente sur le bureau de votre session Windows .

De son côté, SEDMA génère un fichier "xxxxx.XEF" dans le dossier "CodeGenere" de votre projet SEDMA, il suffit de l'importer dans "UNITY PRO" via le menu "Importer". "xxxxx" est le nom du graphe à partir duquel vous avez généré le code.

Là encore, à cette étape et pour votre première manipulation du logiciel UNITY, faites vous aider par un enseignant.

3.3 Commande d'une station de tri robotisée

Pour cette manipulation, vous n'oublierez pas de démarrer votre ordinateur sous RTAI !

Le logiciel SEDMA génère un fichier "xxxxx.c" dans le dossier "CodeGenere" de votre projet SEDMA. "xxxxx" est le nom du graphe à partir duquel le code a été généré. Il suffit donc de compiler le fichier à l'aide du Makefile récupéré sur le réseau puis de l'exécuter par les commandes suivantes :

```
>> make nom_pgm=xxxxx
>> sudo run xxxxx.exe
```

Attention si vous travaillez à distance à travers une connexion `ssh` : la compilation doit être effectuée sur la machine connectée aux robots, sous RTAI.

3.4 Commande d'un système de transport

Cette manipulation n'utilise pas de matériel physique et permet de travailler uniquement sous le logiciel SEDMA qui permet la simulation et l'analyse de propriétés de modèles réseaux de Petri.

Toutefois, si vous avez terminé suffisamment tôt le travail demandé, il est possible de générer le code correspondant à cette application pour effectuer un test. Ceci ne sera pas pris en compte pour l'évaluation, mais il est toujours très satisfaisant de voir fonctionner la commande dont on vient de faire la synthèse.

La commande peut être appliquée par le biais de l'automate programmable industriel connectée à la gare de triage. Le fichier produit par le générateur de code "xxxxx.XEF" se trouve dans le dossier "CodeGenere" de votre projet SEDMA. Lancer le logiciel "UNITY PRO", puis importez le fichier "xxxxx.XEF". Faites vous assister par un enseignant pour les premières manipulations de "UNITY PRO".