

# TP MODELES POUR LE PARALLELISME

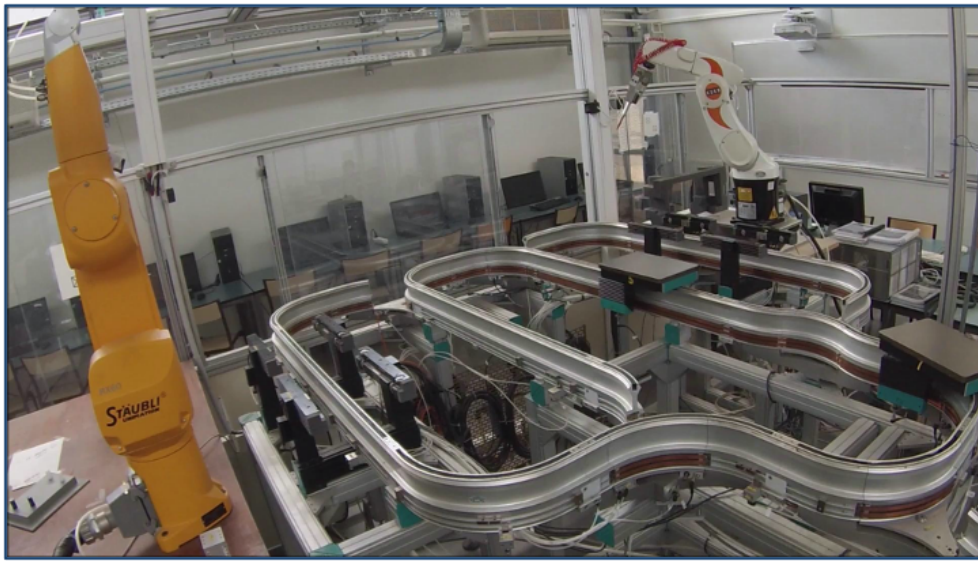
## 2A SRI 2023-2024



# SUJET (12H) : MISE EN OEUVRE D'UNE COMMANDE D'UNE CELLULE FLEXIBLE

## INTRODUCTION

Ces TP ont pour but d'illustrer différentes manières de commander un système de transport industriel. La cellule flexible des espaces robotiques de la Maison de la Formation Jacqueline Auriol (MFJA) est considérée et différents cahiers des charges sont proposés, du plus simple au plus complexe. Dans un premier temps, une commande Grafset du système de transport sera réalisée. On mettra ensuite en œuvre une commande modélisée par réseaux de Petri et implémentée en langage ST. On utilisera le logiciel propriétaire UnityPro permettant de programmer des automates industriels.



## PRESENTATION DU SYSTEME

On considère un système de transport composé de sections et d'aiguillages sur lesquels circulent des navettes autonomes. Un automate permet de contrôler les parties du système représentées sur la figure suivante. Les navettes transportent des produits d'un poste de travail en autre (représenté par les segments bleus sur la figure). Les aiguillages permettent de dériver les navettes vers des postes de travail robotisés.

Les navettes se déplacent dans le sens inverse des aiguilles d'une montre dès le système mis sous tension et activent à leur passage des capteurs de présence notés :

- $PS_i$  disposés en amont des aiguillages  $A_i$  ( $i \in \{1, 2, 11, 12\}$ );
- $CP_i$  disposés en aval des aiguillages  $A_i$  ( $i \in \{1, 2, 9, 10\}$ );

Une navette s'arrête sur la position  $PS_i$  ( $i \in \{1, 2, 11, 12\}$ ) si l'actionneur  $ST_i$  ( $i \in \{1, 2, 11, 12\}$ ) est positionné à 0. Si tous les stops sont désactivés, les navettes avancent donc sans s'arrêter (2 navettes peuvent éventuellement se rattraper mais une détection de proximité évite les collisions). Il est possible d'indexer les navettes une fois celle-ci arrêtée devant un poste de travail grâce aux actionneurs  $PI_i$  ( $i \in \{1, 2, 7, 8\}$ ), les capteurs  $CPI_i$  ( $i \in \{1, 2, 7, 8\}$ ) permettent de connaître l'état de l'indexage.

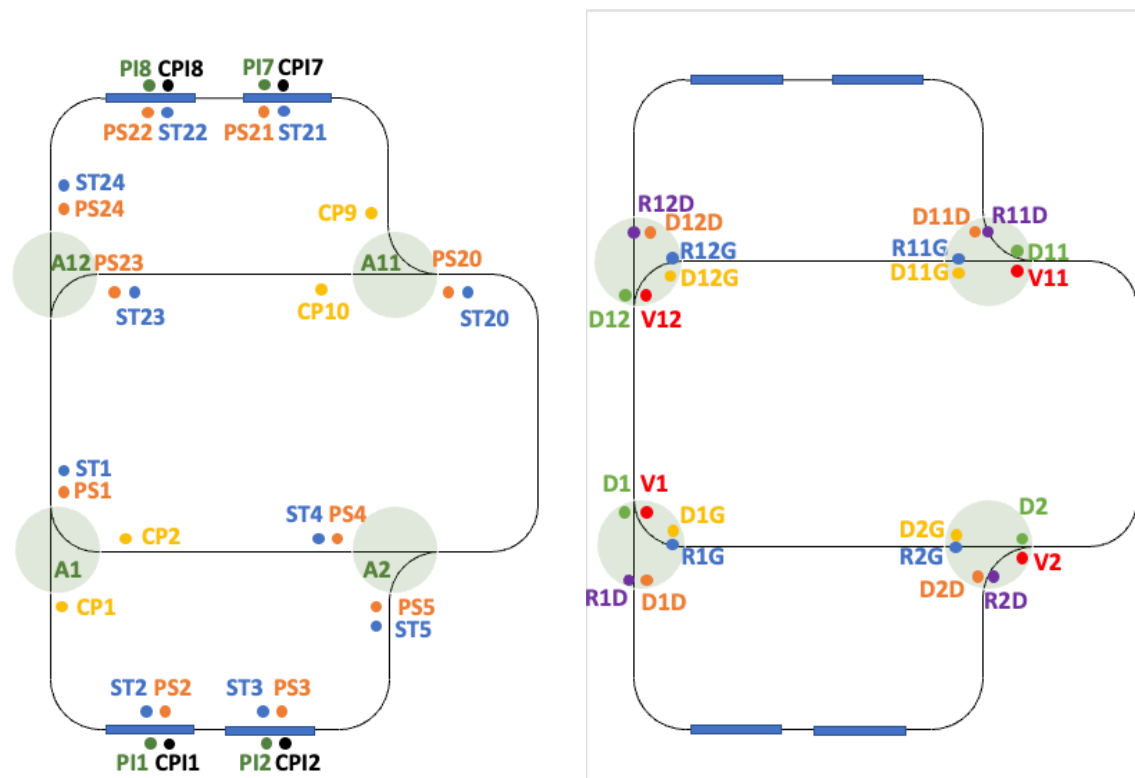


Figure 1. Capteurs et actionneurs

Chaque aiguillage  $A_i$  ( $i \in \{1, 2, 11, 12\}$ ) comporte 4 actionneurs pneumatiques et 2 capteurs :

- Les actionneurs pneumatiques  $R_{id}$ ,  $R_{ig}$  permettent de commander une rotation des aiguillages à droite ou à gauche, respectivement ;
- Les actionneurs pneumatiques  $D_i$ ,  $V_i$  permettent de déverrouiller/verrouiller les aiguillages, respectivement ;
- Les capteurs de position  $D_{id}$ ,  $D_{ig}$  permettent de connaître le positionnement à droite ou à gauche de l'aiguillage, respectivement.

Une commande rotation ne peut être exécutée que si l'aiguillage est déverrouillé (et s'il n'est pas déjà dans la configuration finale souhaitée). Après chaque rotation, l'aiguillage doit de nouveau être verrouillé pour empêcher le déraillement des navettes qui le traversent. De façon générale, il faut interdire toute rotation de l'aiguillage pendant le passage d'une navette.

On suppose que la configuration initiale au moment de la mise sous tension sera contrôlée humainement pour assurer qu'elle soit sans risque pour le matériel. La manipulation consiste à satisfaire les cahiers des charges suivants. Le Cahier des Charges 1 permet une prise en main du matériel. Les autres cahiers des charges sont incrémentaux et seront réalisés au sein d'un même projet par ajout successif de programmes. Des projets UnityPro préconfigurés seront fournis permettant en particulier de faire le lien entre les noms symboliques des capteurs et actionneurs et leur adresse sur l'automate.

### CAHIER DES CHARGES 1 – PRISE EN MAIN

Créer un Grafcet où tous les aiguillages basculent à droite, puis aussitôt à gauche. Libérer alors les stops de sorte que les navettes présentes dans le système circulent de façon ininterrompue, dans la boucle interne (on suppose qu'aucune navette n'est localisée sur un aiguillage au moment de leur rotation).

## CAHIER DES CHARGES 2 – COMMANDE D'UN AUGUILLAGE 2-EN-1

Proposez et testez un Grafcet pour commander l'aiguillage  $A2$  afin de permettre un passage sans collision des navettes arrivant par l'un ou l'autre des entrées de l'aiguillage. Cette commande devra fonctionner quel que soit le nombre de navettes présentes dans le système.

## CAHIER DES CHARGES 3 – COMMANDE D'UN AUGUILLAGE 1-EN-2

Proposez et testez un modèle de commande Grafcet permettant d'assurer que l'aiguillage  $A01$  oriente chaque navette se présentant à son entrée alternativement à droite, puis à gauche. Cette commande doit être pensée pour fonctionner quel que soit le nombre de navettes présentes dans le système.

## CAHIER DES CHARGES 4

Les commandes précédentes des aiguillages  $A1$  et  $A2$  sont conservées. On s'intéresse à présent à concevoir et tester une nouvelle commande modélisée par Réseaux de Petri pour les aiguillages  $A11$  et  $A12$  (que l'on implémentera en langage ST). La commande doit dériver les navettes vers les postes  $P7$  et  $P8$  en assurant qu'il n'y ait jamais plus de deux navettes présentes dans la zone de dérivation (comprise entre les capteurs  $CP9$  et  $PS24$ ). La première navette entrante est arrêtée devant le poste  $P8$  grâce à l'actionneur  $ST22$  et est indexée (actionneur  $PI8$ ) pendant 15 secondes, puis libérée pour repartir. Si une deuxième navette est dérivée vers les postes  $P7$  et  $P8$  alors qu'une navette est déjà présente, elle est arrêtée devant le poste  $P7$  via l'actionneur  $ST21$  tant que le poste  $P8$  est occupé, puis acheminée à son tour vers le poste  $P8$  où elle stationne pendant 15 secondes avant de repartir à nouveau.

## MISE EN ŒUVRE D'UN RDP EN LANGAGE ST

Les variables  $P_i$  sont de type entier (INT),  $inter1$ ,  $a$ ,  $b$ ,  $S$  sont des bits d'entrée/sortie (type EBOOL)

(\*initialisation\*)

IF (Inter1) THEN

$P1 := 1$  ;  
 $P2 := 0$  ;  
 $P3 := 0$  ;

ELSE

(\*franchissement t1\*)

IF (( $P1 \geq 1$ ) and  $a$ ) THEN

$P1 := P1 - 1$  ;  
 $P2 := P2 + 2$  ;  
 $P3 := P3 + 1$  ;

END\_IF;

(\*franchissement t2\*)

IF (( $P2 \geq 2$ ) and ( $P3 \geq 1$ ) and  $b$ ) THEN

$P1 := P1 + 1$  ;  
 $P2 := P2 - 2$  ;  
 $P3 := P3 - 1$  ;

END\_IF;

$S := (P3 > 0)$  ;

$V := (P2 > 0)$  ;

END\_IF;

