

BUT GMP 1

SAÉ 2.2

Analyse de la simulation

16 juin 2022



Mathis Lebreton
Noam Lazrague

Sommaire

Introduction 3

Pré-étude 3

- Cahier des charges 3
- Choix du robot 6

Création de l'outil 7

Simulation Roboguide 9

Conclusion 15

Introduction

Dans le cadre de la SAÉ 2.2, nous avons pour objectif la réalisation d'un cas simple, l'implantation d'un îlot robotisé de production.

Dans notre cas, la société ManivinMeca SA souhaite moderniser son outil de production, en remplaçant un robot obsolète par un voire deux robots du catalogue FANUC.

Dans cette optique, nous devons créer une pince sur le logiciel Solidworks, afin d'avoir un préhenseur spécialement conçu pour la prise des pignons que l'on usine.

Ces robots doivent avoir la capacité de répondre aux besoins et attentes du client, tout en tenant compte des contraintes de la chaîne de production.

Pré-étude

A cette occasion, il nous est demandé de réaliser une pré-étude technique en reprenant le cahier des charges du client.

Puis, nous devons proposer des ébauches de solutions d'implantation de la nouvelle ligne de production.

Cahier des charges

Contexte du projet

Dans l'optique de modernisation de son outil de production, la société ManivinMeca SA souhaite modifier sa ligne de rectification de pignons en remplaçant un robot obsolète par un robot FANUC.

Descriptif des pièces

Voici la ligne de rectification de pignon :

		X (en mm)	Y (en mm)
Prise sur bac	A	-55	480
Mesure Balourd 1	B	435.7	1283.8
Usinage	C	999	1490.5
Balourd 2	D	1762.3	1283
Rebut non équilibré	E	2083	779
Graissage	F	2358	529.5
Rebut non équilibré	G	2194	64
Sortie	H	340	-234

PIGNON



Données utiles :

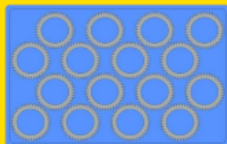
- Diamètre extérieur = 80 mm
- Diamètre intérieur = 55 mm
- hauteur = 12 mm
- masse = 0,198 g

CAISSE



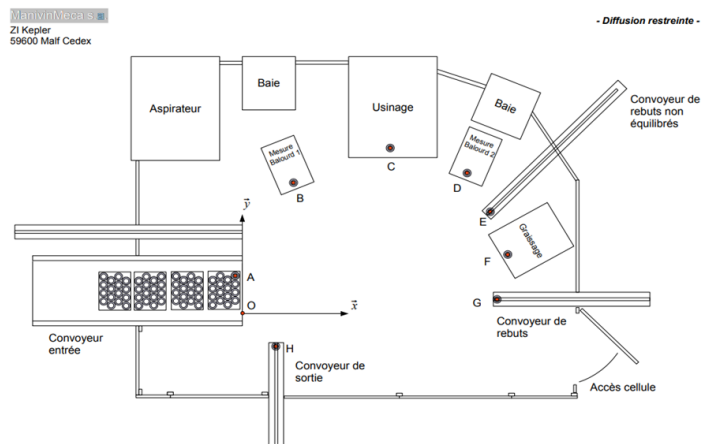
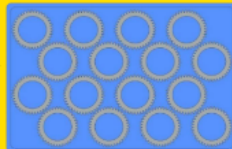
Données utiles :

- 300 x 450 x 25 mm



Disposition des pignons caisse n

Disposition des pignons caisse n+1

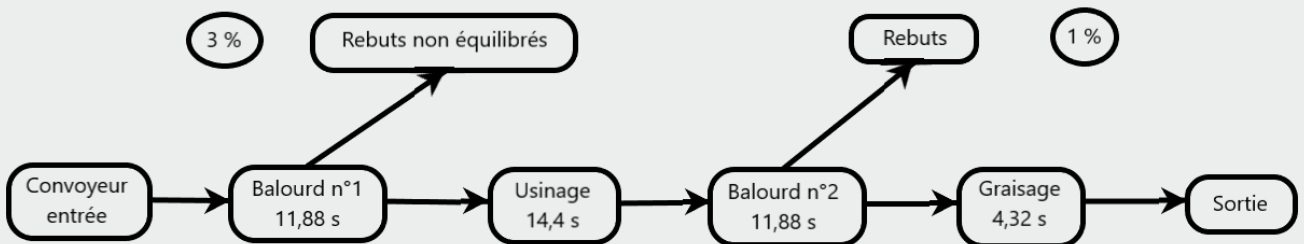


Description de l'îlot

La ligne de rectification se déroule en 6 étapes :

- Les pignons arrivent dans les bacs sur le convoyeur d'entrée
- Ils subissent une mesure de balourd initial dans la machine 'Mesure Balourd 1' au point B.
 - si les mesures sont hors-tolérance, la pièce est évacuée, sur le convoyeur 'Rebut pièces non équilibrées' au point E
- Sinon, le pignon est réusiné au point C dans la machine 'Usinage' à partir des données de mesure
- Un dernier contrôle de balourd résiduel est réalisé dans la machine 'Mesure Balourd 2' au point D
- En fonction des résultats de cette dernière mesure, la pièce est évacuée :
 - vers la machine de graissage au point F puis sur le convoyeur de sortie au point H si les mesures sont conformes
 - sur le convoyeur de rebuts au point G si le balourd résiduel est non conforme
- Les pignons devront être présentés face supérieure vers le haut pour tous les points de prise/dépose. Cette seule face est usinée dans la phase de reprise de balourd

Analyse de cycle :



Donnée d'exploitation :

Horaire de production = $3 \times 8h / 220j / an$

Ce qui équivaut à une production annuelle de 5 280h

Donnée utile :

L'ensemble des points de prise/dépose sont mesurés à une hauteur de 1 200 mm du sol fini.

Donnée utile :

- Zone bleu = Zone d'action
- Zone rouge = Position robot

- Positions des points de prise/dépose :

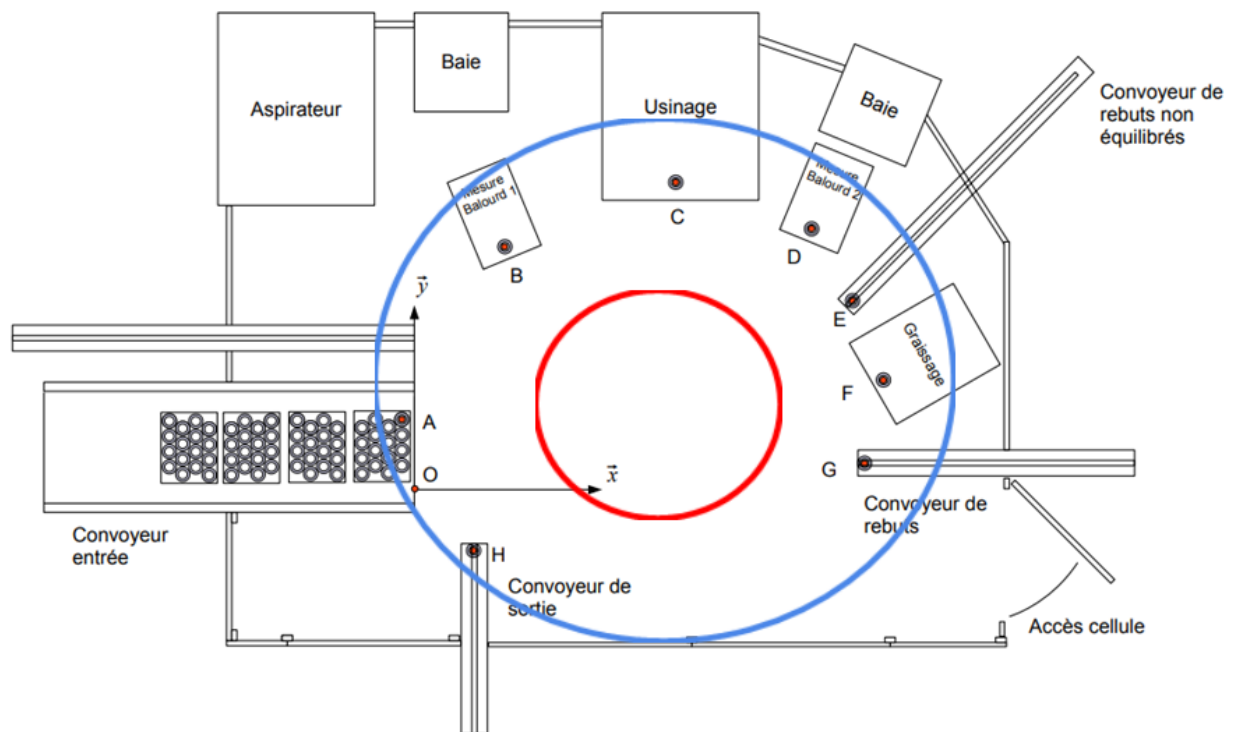
		X (en mm)	Y (en mm)
Prise sur bac	A	-55	480
Mesure Balourd 1	B	435.7	1283.8
Usinage	C	999	1490.5
Balourd 2	D	1762.3	1283
Rebut non équilibré	E	2083	779
Graissage	F	2358	529.5
Rebut non équilibré	G	2194	64
Sortie	H	340	-234

- Rayon d'action minimum :

Graissage	F	2358	529.5
Prise sur bac	A	-55	480

- Si nous prenons un rayon d'action légèrement supérieur à 2 413 mm, cela permettra à notre robot situé dans la zone rouge, d'atteindre chaque point de position de nos pignons (en rouge sur la cellule).

ManivinMeca s.
ZI Kepler
59600 Malf Cedex



Choix du robot

Afin de choisir un robot adapté à notre situation, nous avons étudié différents robots du catalogue FANUC, ce qui nous a permis de sélectionner ces trois robots correspondant à nos critères :



Données utiles des robots :

- Série M-710 : Capable de supporter une charge de 70 kg au poignet et à un rayon max de 3 123 mm.
- Série M-20 : Capable de supporter une charge de 35 kg au poignet et à un rayon max de 2 272 mm.
- Série M-10 : Capable de supporter une charge de 16 kg au poignet et à un rayon max de 2 028 mm.

Choix finals :

- Parmi ces 3 robots respectant nos critères, celui qui répond le plus aux attentes est le robot 3 Série M-10 version iD type 12.
- Celui-ci accepte une charge jusqu'à 12 kg et à un rayon d'action maximal de 1 441 mm > 1 206.5 mm. Nous avons arrêté notre choix sur ce robot, car la charge admissible est inférieure aux autres, mais suffisante pour répondre à notre besoin. Le rayon d'action optimal correspond à notre pré-étude de la cellule et de notre besoin.
- Côté budgétaire, nous supposons qu'il est moins coûteux, car ces caractéristiques sont inférieures aux autres robots sélectionnés.

Création de l'outil

Parmi les différentes tâches demandées dans le cadre de la SAÉ 2.2, il nous a été fixé comme objectif, la création d'un préhenseur à l'aide du logiciel de conception Solidworks qui devra être relié à un vérin fourni.

Pour cela, les professeurs en charge du projet, nous ont fourni quelques informations utiles sur le vérin (cotation de la pièce) pour nous permettre de commencer une conception papier lors de la première séance.

Schéma vérin :

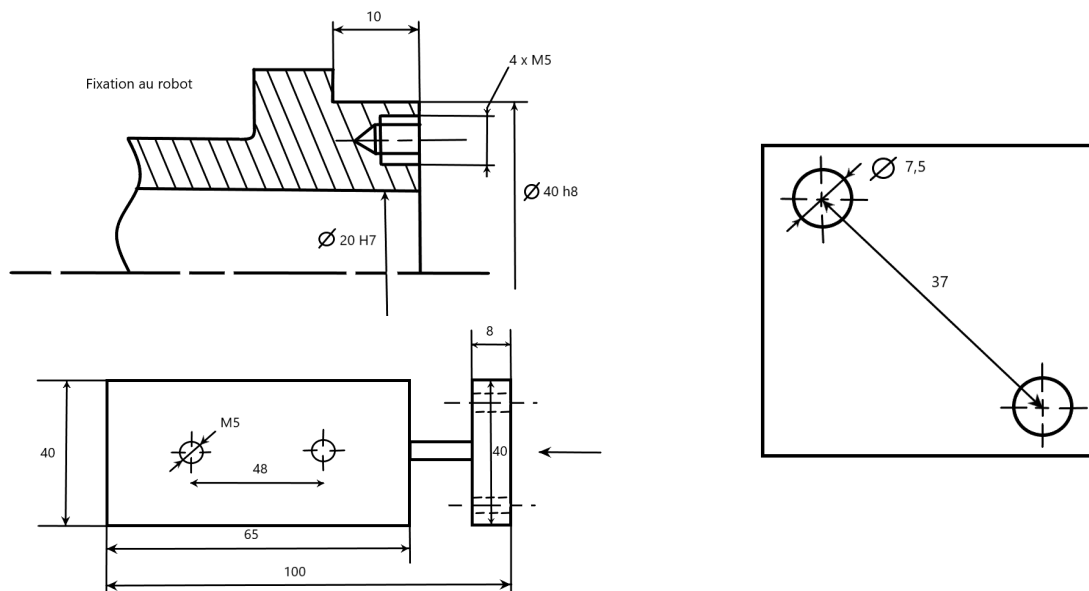
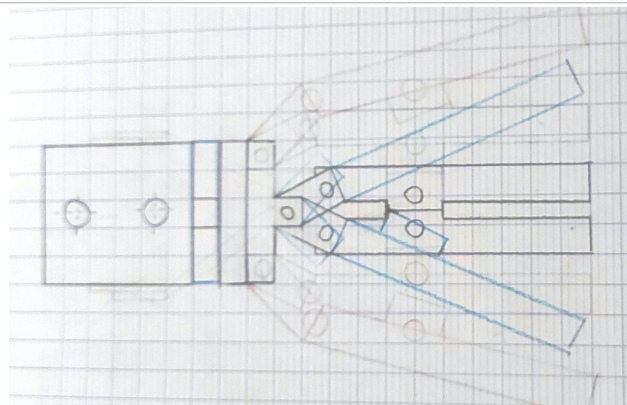


Schéma vérin + préhenseur :



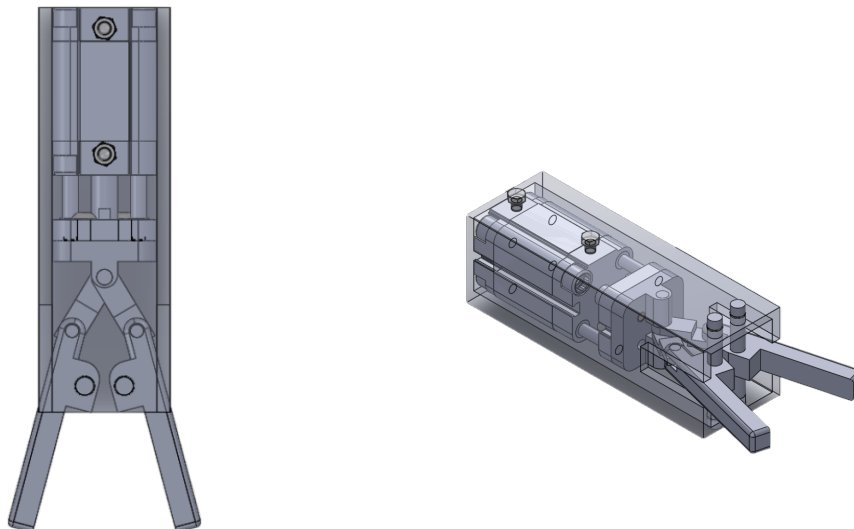
Explication de la réalisation du préhenseur :

- Pour le préhenseur, nous avons décidé de créer deux pinces, qui une fois ouverte iront se coller au diamètre intérieur du pignon, afin de pouvoir le déplacer facilement. Nous devons prendre en compte que la prise des pignons avec le préhenseur est faisable, seulement sur la partie interne du pignon, soit le diamètre intérieur (plus petit). Car avant la prise du pignon situé dans le bac, celle-ci est trop proche des autres pièces (regrouper, coller, espace restreint). Il n'y a pas assez de place ou d'espace pour pouvoir s'agripper sur les bords. De plus, les pignons sont recouverts de dentures et ne permettent pas à la pince d'avoir une bonne prise.
- Pour permettre l'ouverture et la fermeture de nos deux pinces, nous nous sommes basées sur un système simple qui est celui des ciseaux.
- Pour cela, nous avons commencé par fixer une plaque au vérin. Ensuite, nous avons relié cette plaque par une liaison pivot à deux biellettes superposées. Puis, nous avons relié chaque biellette à une pince par son bout, grâce à une liaison pivot. Ainsi, nous avons réussi à relier le vérin aux pinces de façon à ce que les pinces accompagnent le vérin lors d'un mouvement. C'est en reliant nos pinces à des biellettes par liaisons pivots, grâce à des axes. Il en est de même pour la liaison entre les biellettes et la plaque fixée au vérin, ces deux liaisons permettent de relier directement les pinces au vérin.
- Il ne restait plus qu'un seul problème à régler, celui de l'ouverture et la fermeture des pinces. Pour cela, nous avons eu l'idée de créer un bloc, dans lequel nous fixerons notre vérin, ainsi que nos pinces. Nous avons alors conçu 2 axes reliés par coaxialité avec nos pinces, qui seront ensuite fixés au bloc, grâce à deux circlips à chaque extrémité du bloc qui viennent bloquer la translation de l'axe.

Par conséquent, nous nous retrouvons avec cinq liaisons pivot au total :

- La première permet d'accompagner la tige du vérin, lorsqu'elle se ferme et s'ouvre.
- La deuxième est directement reliée aux pinces (une par pince).
- Il ne reste plus que les deux dernières, qui sont des liaisons fixes reliées au bloc, cette rotation permettra de diriger les pinces, en fonction de la position du vérin, ce qui nous donnera une position ouverte et fermée.

Rendu de la pièce 3D sur Solidworks :

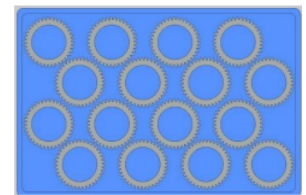
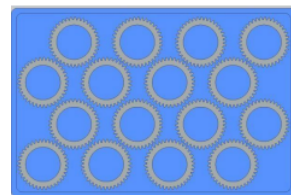
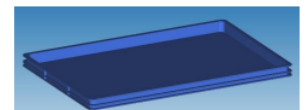
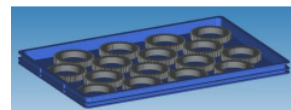
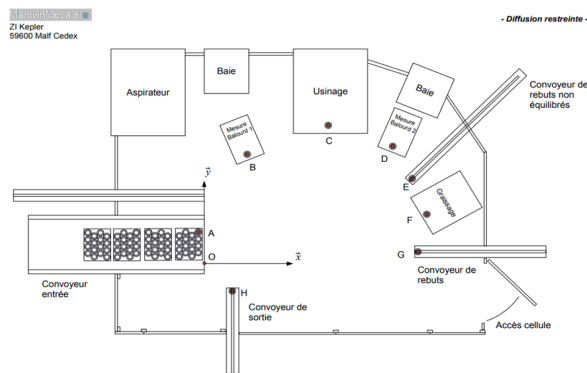


Simulation Roboguide

Modélisation de la cellule :

1) Reproduction du plan

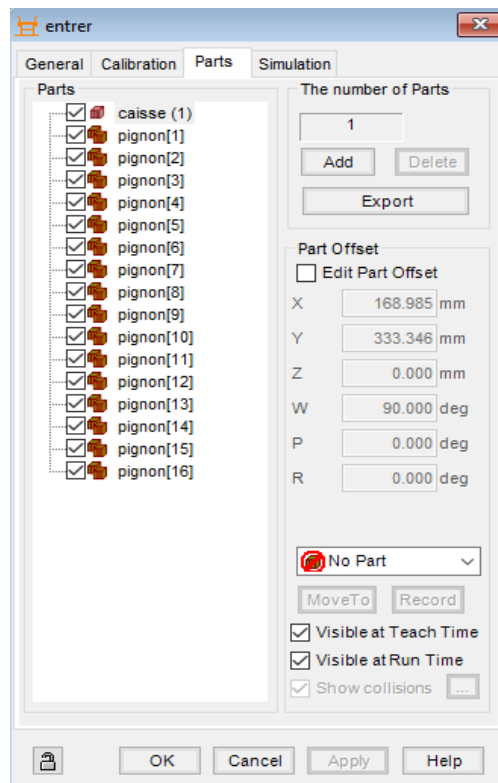
Nous avons cherché à reproduire le schéma présenté sur le sujet de la SAÉ 2.2. Nous avons tout d'abord déposé les éléments dans la simulation, puis nous avons utilisé le tableau contenant les coordonnées des pièces, afin de pouvoir placer correctement chacun des éléments dans la simulation. Nous les nommerons des fixtures et des parts.



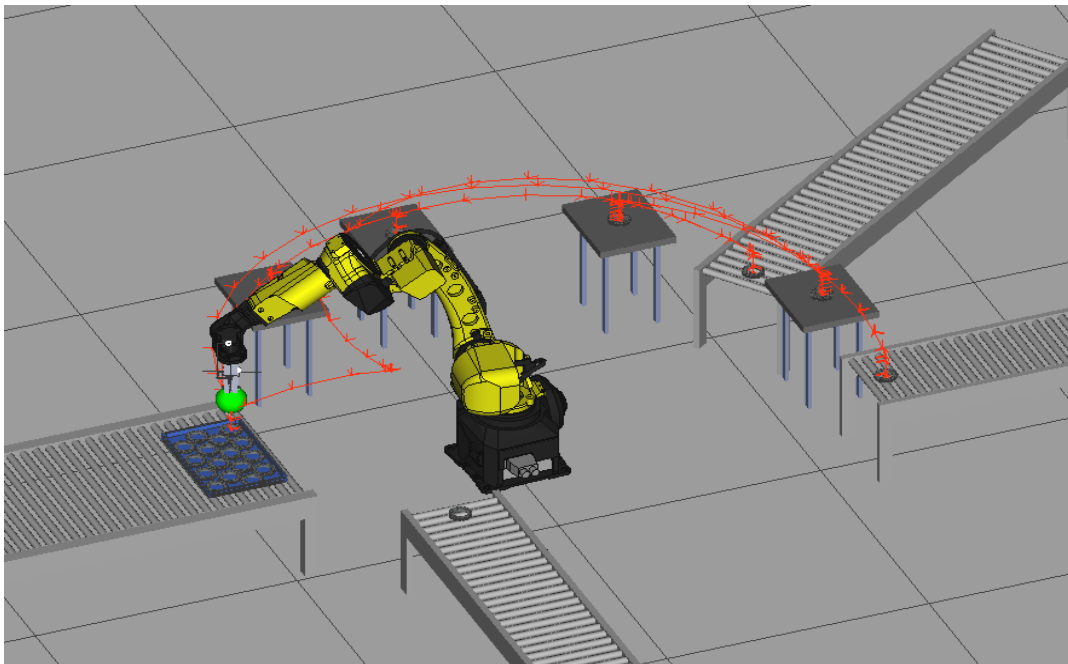
		X (en mm)	Y (en mm)
Prise sur bac	A	-55	480
Mesure Balourd 1	B	435.7	1283.8
Usinage	C	999	1490.5
Balourd 2	D	1762.3	1283
Rebut non équilibré	E	2083	779
Graissage	F	2358	529.5
Rebut non équilibré	G	2194	64
Sortie	H	340	-234

- Fixtures
 - balourd1
 - balourd2
 - entrer
 - graissage
 - nonequilibre
 - rebut
 - sortie
 - usinage
- Parts
 - caisse (1)
 - pignon

Les fixtures sont les éléments qui ne bougent pas, qui ne font aucune action dans la simulation. Elles sont représentées par des tables et des convoyeurs. Les parts sont des éléments qui devront être ajoutés aux fixtures, afin de les positionner et de simuler des mouvements. Dans notre cas, les parts sont les pièces qui bougent dans la réalité.

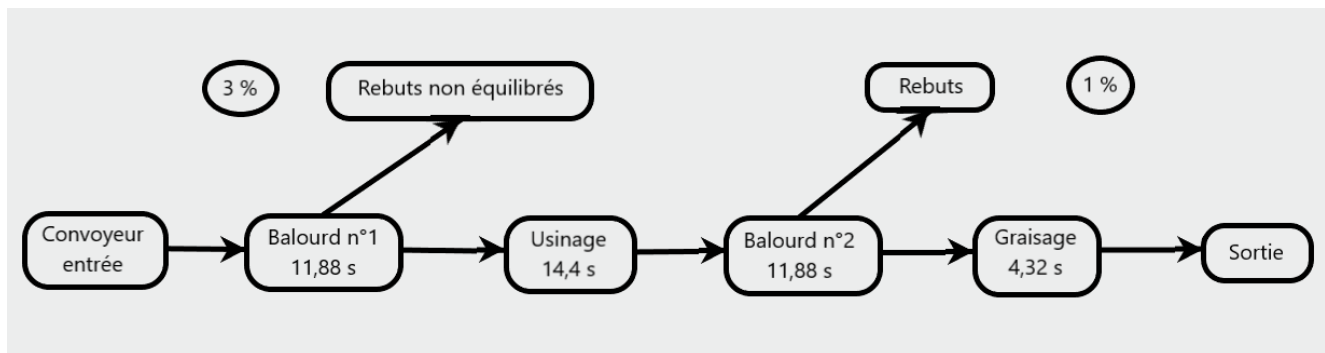


2) Création de la simulation



La tâche de création de la simulation à réaliser est complexe pour plusieurs raisons :

- Premièrement, il faut respecter les différents trajets possibles des pignons
Par exemple, si durant la cellule nous imaginons que la pièce est diagnostiquée non conforme, il doit pouvoir exister un chemin adapté à ce type de pièce.
- Deuxièmement, la recherche d'optimisation pour se rapprocher le plus d'une simulation de cellule réelle, en l'occurrence le robot que nous avons retenu dans la pré-étude plus haut (M-10 ID 12) doit toujours être en mouvement. L'objectif est d'avoir le rendement le plus élevé en sortie de simulation, en d'autres termes, de perdre le moins de temps possible, pour cela, nous devons utiliser des timers (le timer permet de gérer le temps et d'accompagner des conditions).



- Pour faciliter l'organisation de nos programmes et cellules, nous avons tout d'abord créé une cellule fonctionnant avec des fonctions WAIT, qui permettent d'associer une variable de temps (seconde) à la fin d'un programme ou étape.

Les programmes contiennent différentes valeurs et fonctions, comme des boucles permettant de refaire une liste d'actions sur un nombre donné. Les calls programmes permettent d'appeler les sous-programmes ou programmes esclaves, afin de faire fonctionner la cellule entière à travers un programme global.

Les users frames a pour rôle d'associé un repaire à une fixture, pour qu'en cas de problème, nous puissions déplacer celle-ci, sans à avoir à refaire des datas (data = prise origine d'un point unique).

Les offsets sont des points permettant dans notre cas, d'éloigner légèrement la machine de notre pièce, pour montrer, simuler une zone de sécurité autour de celle-ci.

Et pour finir, les call add simulation permettent de simuler le dépôt et la prise des pièces sur la fixture choisie (exemple : étape d'usinage call drop usinage ou encore call pick usinage).

Chacune de ces fonctions devra être entrée dans un petit écran appeler commande.



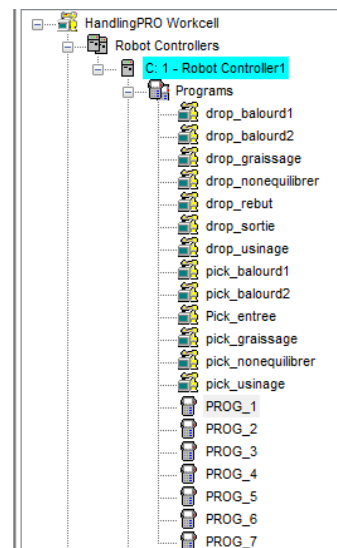
Dans cette commande, nous entrerons 7 programmes différents qui auront chacun une action.

Le programme N°1 a pour but d'exécuter la première action et démarrer la boucle, permettant de faire fonctionner tous les programmes, en les appelant chacun leur tour, avec les bons temps et les bons users frames associés.

```

17: CALL PROG_2
18: WAIT 14.40(sec)
19:
20: UFRAME_NUM=3
21: CALL PROG_3
22: WAIT 11.88(sec)
23:
24: UFRAME_NUM=4
25: CALL PROG_4
26:
27: UFRAME_NUM=5
28: CALL PROG_5
29: WAIT 4.32(sec)
30:
31: UFRAME_NUM=6
32: CALL PROG_6
33:
34: UFRAME_NUM=7
35: CALL PROG_7
36:

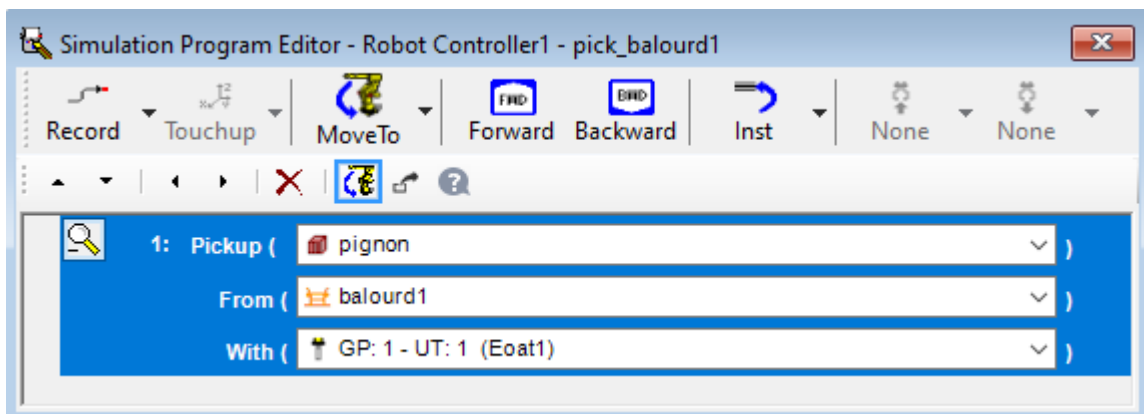
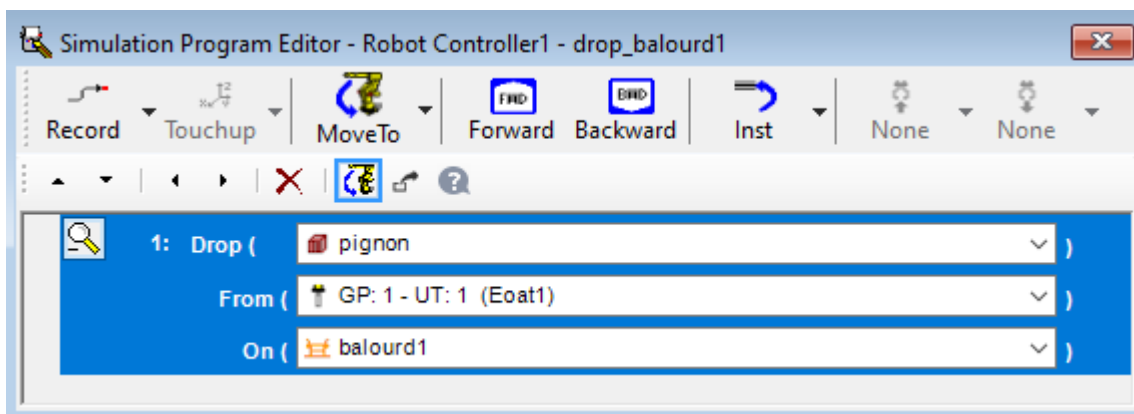
```



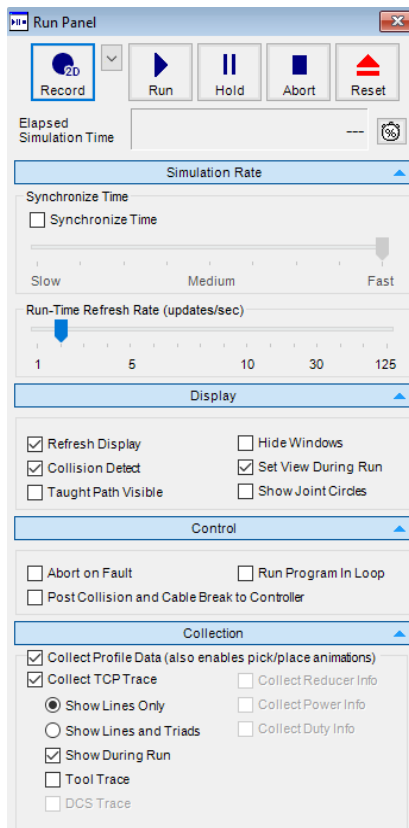
Il s'agit de la fin du programme 1, qui appelle les programmes 2 à 7, en respectant les bons users frames et les temps associés à chaque étape.

Nous avons utilisé une fonction qui se nomme « Add simulation programme » qui permet de rendre la simulation plus réaliste. C'est-à-dire, simuler la prise et le dépôt d'une pièce.

Drop = dépôt
Pick = prise



Durant la simulation, nous avons le choix de la vitesse, car les secondes de la simulation ne sont pas les mêmes que les secondes réelles. Nous pouvons via un bouton avoir accès aux vitesses, lorsque la simulation est lancée.



Cette version de la simulation avec la fonction WAIT, répond aux attentes de la SAÉ, mais en aucun cas n'est le reflet de la réalité. Car, en réalité dans l'industrie tout est optimisé, afin d'être compétitif.

Dans notre cas, une pièce met 1 mn à faire le tour du circuit, soit environ 16 mn l'ensemble du bac.

Pour essayer de concevoir une simulation qui respecte une véritable cellule de contrôle et production (simulation optimisée, nous devons utiliser les fonctions Timer et If permettant de créer des boucles imposant des conditions.

En utilisant cette fonction, nous pouvons demander au programme de s'effectuer selon l'ordre que la cellule aura jugé plus rapide, par exemple : Si durant une période d'usinage de 14.4s, le robot a le temps de récupérer une pièce et de la poser sur le balourd 1, alors, il le fera. Sinon, il continuera la suite du programme d'origine et passera au balourd 2. A ce moment, il contrôlera les temps et jugera s'il est possible ou non de commencer une nouvelle étape en parallèle.

Pour pouvoir utiliser les timers, il faut tout d'abord les assignés au registre et leur donner une valeur temporelle. Ils seront ensuite associés à une étape de la cellule en particulier, afin d'être exécutés correctement. Pour pouvoir optimiser cette cellule, dont la finalité est de la rendre plus réaliste, nous utiliserons des fonctions If qui auront pour but de dire si la condition est respectée. Alors, nous l'exécuterons, sinon nous continuerons comme il est écrit par la suite.

Le programme suivant signifie :

- Nous appelons les users frames 0 (repère associé au robot)
- Nous réinitialiserons tous les timers, car le début d'une boucle programmes représente aussi la fin. Le mettre à la fin comme au début ne change rien, il s'agit juste de rappel
- Nous créons une boucle enregistrée dans le registre 6, qui fera 10001 fois le programme ci-dessous
- Nous nommons les registres 2 à 5 en tant que timer pour pouvoir les distinguer et les associés aux bonnes étapes de la cellule (balourd 1, usinage, balourd 2, graissage)
- Nous expliquons en partant de la fin, chacune des conditions, si les temps d'usinage et supérieur au temps associé à l'origine de 4.32s. Alors, nous pouvons appeler le programme suivant et nous répétons le processus jusqu'au début du programme global, soit programme 1 première étape.

Ce programme ne fonctionne pas, nous avons eu quelques problèmes de programmation.

```
1: UFRAME_NUM=0
2: TIMER[1]=RESET
3: TIMER[2]=RESET
4: TIMER[3]=RESET
5: TIMER[4]=RESET
6: R[1]=1
7: FOR R[6]=0 TO 10000
8: R[2:timer1]=TIMER[1]
9: R[3:timer2]=TIMER[2]
10: R[4:timer3]=TIMER[3]
11: R[5:timer4]=TIMER[4]
12: IF R[2:timer1]=0,CALL PROG_1
13: WAIT .50(sec)
14: R[2:timer1]=TIMER[1]
15: R[3:timer2]=TIMER[2]
16: R[4:timer3]=TIMER[3]
17: R[5:timer4]=TIMER[4]
18: IF R[5:timer4]>4.32,CALL PROG_7
19: IF R[4:timer3]>11.88 AND
: R[5:timer4]=0,CALL PROG_5
```

Conclusion

Nous avons tiré énormément d'enseignements concernant le logiciel Roboguide et nous sommes déçus de ne pas avoir pu aboutir à l'optimisation de cette cellule.

Cependant, nous avons constaté la différence entre une cellule classique et optimisée. Le gain de temps et d'argent est exorbitant.

Le choix du robot est complexe et la création de la simulation minutieuse. Notre cellule respecte néanmoins l'objectif demandé.