

Conception d'un système robotique avancé pour une application de micro pick-and-place

Travail de Bachelor

Département TIN
Filière Génie Électrique
Orientation Électronique et Automatisation industrielle

Gaëtan Worch

6 février 2025

Supervisé par :
Prof. G. Costanzo (HEIG-VD)

Préambule

Ce travail de Bachelor (ci-après **TB**) est réalisé en fin de cursus d'études, en vue de l'obtention du titre de Bachelor of Science HES-SO en Ingénierie.

En tant que travail académique, son contenu, sans préjuger de sa valeur, n'engage ni la responsabilité de l'auteur, ni celles du jury du travail de Bachelor et de l'École.

Toute utilisation, même partielle, de ce TB doit être faite dans le respect du droit d'auteur.

HEIG-VD
Le Chef du Département

Yverdon-les-Bains, le 6 février 2025

Authentification

Je soussigné, Gaëtan Worch, atteste par la présente avoir réalisé seul ce travail et n'avoir utilisé aucune autre source que celles expressément mentionnées.

Gaëtan Worch

A handwritten signature in black ink, appearing to read 'Gaëtan Worch', written over a horizontal line.

Yverdon-les-Bains, le 6 février 2025

Résumé

Table des matières

Table des figures

Liste des tableaux

Liste des codes sources

Chapitre 1

Introduction

1.1 SwissCat+

Test **einstein** Les catalyseurs est une espèce chimique qui permet ou accélère la réaction chimique sans être consommé dans le processus. Les catalyseurs sont utilisées dans divers domaines (Agricole, militaire, chimie, traitements des déchets, transformation de polluants, ...). Ils sont indispensables dans notre société, cependant, ces produits requièrent souvent des terres rares. Par exemple, les catalyseurs dans les voitures (qui servent à réduire les émissions de gaz polluant en transformant notamment le monoxyde de carbone, les hydrocarbures imbrûlés et de l'oxyde d'azote en eau, dioxyde de carbone et dioxyde d'azote) sont composée d'alumine, d'oxyde de cérium mais surtout ils sont composés d'au moins trois platinoïde.

C'est dans ce contexte de réduction d'utilisation de terre rare que le laboratoire SwissCat+ a été créé dans le but d'optimiser la composition des catalyseurs et d'en trouver des nouveaux. Le laboratoire est scindé en deux, le laboratoire Est, située à l'ETHZ à Zurich, s'occupe de faire des recherches sur les catalyseurs hétérogène, tandis que la partie du laboratoire Ouest, situé à l'EPFL à Lausanne, fera des recherches sur les catalyseurs homogènes.

1.1.1 laboratoire ouest

Le laboratoire doit être automatiser afin de pouvoir de grandes quantités d'expériences afin de pouvoir explorer un maximum de l'espace chimique. Pour avoir un espace chimique le plus grand possible, il faut que le laboratoire puisse manipuler des matières liquides et de larges quantité de poudre (0.1 à 50 mg).

1.2 Contexte et objectifs du projet Storm

C'est pour pouvoir gérer cette plage de quantité de poudre que le projet STORMS (*STOchastic Robotized Micro Sampling*), (en collaboration entre l'EPFL, l'HEIG-VD, Chemspeed Technologies et Dietrich Engineering Consultants) est né. STORMS est composés de plusieurs modules :

- la standardisation,
- le stockage,
- le micro-échantillonnage (« microsampling »),
- la recombinaison,
- la box d'Edy

— les box de Chemspeed

La standardisation a pour but de donner des récipients de taille standard aux micro-sampling qui va créer des microcapsules avec des quantités qui ciblent les quantités souhaitées qui sont stockées sur des plaques nommées « *wellplate* », qui sont ensuite rangées dans des racks dans le stockage. Ce stockage sert de lien entre la standardisation, le « microsampling » et la recombinaison. La recombinaison doit déterminer quelles microcapsules choisir pour toutes les recettes présentées dans le « *batch* ».

1.3 Recombinaison de microcapsules

Le processus de recombinaison consiste à assembler différentes microcapsules de poudres pour répondre aux besoins expérimentaux. Le logiciel de recombinaison sélectionne les capsules disponibles afin d'obtenir une composition approchant les quantités nécessaires pour une expérience donnée. Le but est de maximiser le nombre d'expériences réalisées.

Chapitre 2

Problématique de la recombinaison

2.1 Objectifs

Concevoir et développer une solution permettant le transfert automatisé, précis et rapide de microcapsules de réactif chimique entre le stockage et les réacteurs. Un algorithme devra être conçu pour optimiser la sélection des microcapsules afin de maximiser le nombre de recettes réalisées. L'ensemble du système devra garantir un haut niveau de fiabilité et de sécurité dans le processus.

2.2 Cahier des charges fonctionnel

Table 2.1 – *Cahier des charges fonctionnel*

fonction	énoncé de la fonction	exigence
FP 1	Manipuler des microcapsules de manière automatisée, sans les endommagées	— Ne pas détériorer la microcapsule
FP 1.1	Prélever les microcapsules dans une plaque	— Position de prise arbitraire — Contrôler que la microcapsule soit saisie
FP 1.2	Déplacer les microcapsules	
FP 1.3	Déposer les microcapsules	— Position de dépose arbitraire dans la plaque de réacteurs
Continued on next page		

CHAPITRE 2. PROBLÉMATIQUE DE LA RECOMBINAISON

Table 2.1– continued from previous page

fonction	énoncé de la fonction	exigence
FP 2	Déterminer les microcapsules les plus adaptées pour chaque réacteur, selon une recette donnée	
FP 2.1	Recevoir la recette pour chaque réacteur	
FP 2.2	Accéder à la base de donnée du stock	
FP 2.3	Déterminer la combinaison de microcapsules optimal pour délivrer la masse de produit donnée	<ul style="list-style-type: none"> — Précision dans la masse délivrée : varie à chaque recette. — Nombre maximale de microcapsules par réacteur : 5
FP 2.4	Transmettre la position des microcapsules dans le stock et sur la plaque	
FP 2.5	Informé le stock des microcapsules prélevées	
FC 1	Respecter les dimensions de l'endroit confiné	Dimension de la boîte : $133 \times 95 \times 94$ cm
FC 2	Utiliser les énergies disponibles	<ul style="list-style-type: none"> — Électrique : <ul style="list-style-type: none"> — 400 V triphasé — 230 V monophasé — Pneumatique : 8 bar
FC 3	Utiliser les plaques déjà présentes	<ul style="list-style-type: none"> — Nombre de trou sur la plaque de prise : 384 — Nombre de trou sur la plaque de dépose : 48
FC 4	S'adapter aux éléments déjà présents	<ul style="list-style-type: none"> — API : Beckhoff
FC 5	Dimension des microcapsules : $\varnothing 3$ mm	

Chapitre 3

Hardware

3.1 Défis techniques du Hardware

Le hardware doit pouvoir manipuler des plaques de microcapsules ou de réacteurs, ainsi que des microcapsules en verre de 3 mm de diamètre en garantissant leur intégrité. Le tout doit être dans un espace confiné.

3.2 Recherches de solution

Pour la recherche des solutions, le hardware a été décomposé par les fonctions suivantes :

- saisie et dépose des microcapsules ;
- déplacement des microcapsules ;
- saisie et dépose des plaques.

La position des plaques dans la glove box a également été étudié avec 6 configurations différentes.

3.2.1 Saisie et dépose des microcapsules

Pour la saisie des microcapsules, les grandes familles de solutions proposées sont :

- Aspiration ;
- Mécanique ;
 - Pince à doigt ;
 - Pince Gecko.

Avantage et inconvénients**Table 3.1** – *Avantages et inconvénients des solutions de saisie des microcapsules*

Solution	Avantages	Inconvénient
Aspiration	<ul style="list-style-type: none"> - Exerce moins de pression directe - Simple à installer - Nécessite peu d'entretien 	<ul style="list-style-type: none"> - Apporter l'énergie pneumatique - Bruyant
Pince à doigt	<ul style="list-style-type: none"> - Contrôle précis - Faible coût 	<ul style="list-style-type: none"> - Maintenance fréquente - Ne convient pas au petits objets - Espace limité, pour pouvoir ouvrir et fermer la pince - Nécessite un contrôle de force - Capacité de charge
Pince Gecko	<ul style="list-style-type: none"> - Saisie non intrusive - Ne nécessite pas de source d'énergie externe 	<ul style="list-style-type: none"> - Nécessite un nettoyage pour maintenir l'adhérence - Détachement complexe

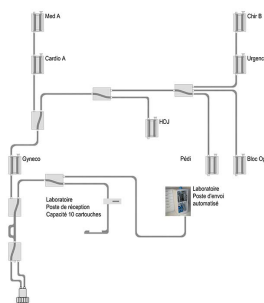
3.2.2 Déplacement des microcapsules

Pour le déplacement des microcapsules de leur plaque jusqu'aux réacteurs, trois idées ont été étudiées :

- Transport pneumatique par tube ;
- convoyeur ;
- robot.

Transport pneumatique par tube

Le système de transport pneumatique par tube¹, serait des tuyaux dans lesquelles naviguent les microcapsules grâce à une différence de pression de chaque côté de la microcapsule. Ce système est déjà présent dans les hôpitaux et dans les grandes surfaces.

**(a)** Schéma d'un réseau de transport pneumatique²**(b)** Cartouche de transport³**Figure 3.1** – *Exemples de réseau de transport pneumatique par tube*

1. Tube pneumatique - Wikipédia

1. <https://www.transport-pneumatique.fr/transport-pneumatique-centres-hospitaliers/>

2. <https://www.transport-pneumatique.fr/cartouches-pochettes/>

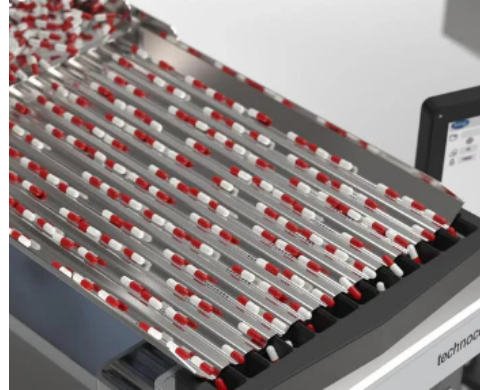
3.2. RECHERCHES DE SOLUTION

Transport par convoyeur

Pour déplacer les microcapsules, un convoyeur peut être utilisé, il faut néanmoins que le convoyeur soit adapté au microcapsule, les microcapsules étant cylindriques, elles risqueraient de rouler sur un convoyeur à bande lisse, mais une bande à tasseau (cf. Figure 3.2a) ou un demi-tube (cf. Figure 3.2b) conviendraient parfaitement.



(a) Convoyeur avec bande à tasseaux⁴



(b) Convoyeur à tube⁵

Figure 3.2 – Exemple de convoyeur

Quant aux différents moyens de mouvoir les microcapsules, il y a :

- les vibrations ;
- le déplacement de la bande ;
- la gravité.

La dernière option nécessite des surface lisses, que le système soit en pente et le temps de déplacement n'est pas réglable. Les deux autres solutions ne se distinguent pas vraiment pour l'instant, car dans tous les cas, l'utilisation d'un moteur électrique est nécessaire.

Déplacement à l'aide d'un robot

Pour le déplacement des microcapsules, seuls les axes T_x , T_y et T_z sont nécessaires, soit 3 degrés de liberté. Un robot de type *SCARA*, cylindrique ou Delta peuvent correspondre.

2. <https://fr.m.wikipedia.org/wiki/Convoyeur>

3. <https://doser-compter.com/products/ligne-de-comptage-king>

Avantages et inconvénients**Table 3.2** – *Avantages et inconvénients des solution de transport des microcapsules*

Solution	Avantages	Inconvénient
Transport pneumatique par tube		<ul style="list-style-type: none"> - Peu modulable - Bruyant - Aiguillage complexe - Maintenance fréquente
Convoyeur	-	<ul style="list-style-type: none"> - Ne convient pas au petits objets - Espace limité, pour pouvoir ouvrir et fermer la pince - Nécessite un contrôle de force - Coût
Robot	<ul style="list-style-type: none"> - Place - Modulable 	<ul style="list-style-type: none"> - Nécessite un nettoyage pour conserver l'adhérence dans le temps - Détachement complexe

3.2.3 Analyse des solutions

De par la complexité et le manque de modularité du transport pneumatique et du convoyeur, le choix d'utiliser un robot a été choisi.

La position des plaques dans la *glove box* est importante pour la suite, elle permettra de choisir le type de robot à utiliser. Il est important de savoir comment placer les deux plaques dans la *glove box*.

Positionnement des plaques dans la glove box

Pour la position des plaques, les solutions trouvées sont :

1. Les plaques ne bougent pas et restent dans les sas ;
2. Les plaques sont positionnées symétriquement par rapport au centre de la *glove box* ;
3. La plaque de microcapsules est déplacée à côté de la plaque de réacteur, cette dernière ne bouge pas ;
4. La plaque de microcapsules reste dans le sas tandis que la plaque de réacteur est transportée à ses côtés ;
5. Les deux plaques sont mises l'une à côté de l'autre en étant plus proches du stock ;
6. Les deux plaques sont mises l'une à côté de l'autre en étant plus proches de la sortie ;

3.2. RECHERCHES DE SOLUTION

Le temps de cycle est le suivant :

$$\begin{aligned}
 tpsCycle &= tpsDeplacementPlaquesCapsule + tpsDeplacement_{Capsule} \\
 &+ tpsAttentePlaque + tpsDeplacement_{PlaquesReacteur} \\
 &= nbrePlaques \cdot \frac{d_{PlaqueMicrocapsules}}{v_{robot}} \\
 &+ nbreMicrocapsules \cdot \frac{d_{EntrePlaque}}{v_{robot}} \\
 &+ 2 \cdot nbrePlaques \cdot tpsAttente + \frac{d_{PlaqueReacteur}}{v_{robot}}
 \end{aligned}$$

Certaines valeurs ne peuvent être connues que lors de la mise en route, notamment l'accélération, la vitesse d'approche, il est nécessaire de faire certaines hypothèses. Les distances sont également arbitraires afin de se faire une idée du temps de cycle des solutions. Ici, la position optimale n'est pas recherchée. En utilisant les hypothèses suivantes :

- $v_{robot} = 1 \frac{m}{s}$;
- $d_{EntrePlaque} = 1.33, 0.3, 0.3, 0.3, 0.3$ et 0.3 m s^{-1} ;
- $d_{PlaqueMicrocapsules} = 0, 0.515, 1.03, 0, 0.83$ et 0.2 m ;
- $d_{PlaqueReacteur} = 0, 0.515, 0, 1.03, 0.2$ et 0.83 m ;
- $d_{PlaqueMicrocapsules} = 0, 0.515, 1.03, 0, 0.83$ et 0.2 m ;
- L'accélération du robot est infinie ;
- La vitesse d'approche n'est pas prise en compte.

Avec ces informations, il est possible de calculer le temps moyen d'un cycle (cf. Figure 3.3) en fonction du nombre de microcapsule demandées ainsi que du nombre de microcapsule par plaque, et ce, pour chaque solution.

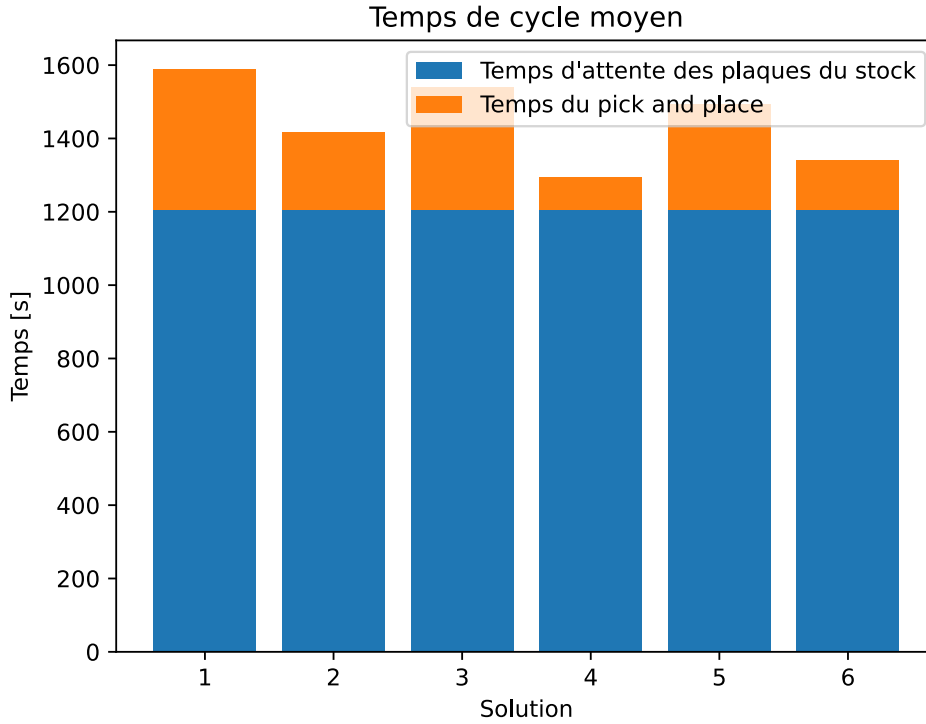


Figure 3.3 – Temps de cycle moyen des différentes solutions

Il est possible de voir que les solutions 4 et 6 semblent les plus rapide. Le temps d'attente des plaques est cependant très important, environ 84 % du temps total. Pour réduire ce délai, il peut être intéressant de paralléliser les tâches.

parallélisation des robots

La parallélisation des tâches consiste à effectuer le *pick and place* des microcapsules, la prise et la dépose des plaques indépendamment. De par la configuration des sas, il n'est possible d'y mettre qu'une seule plaque, la parallélisation des solutions 1, 3 et 4 n'est donc pas possible.

3.2. RECHERCHES DE SOLUTION

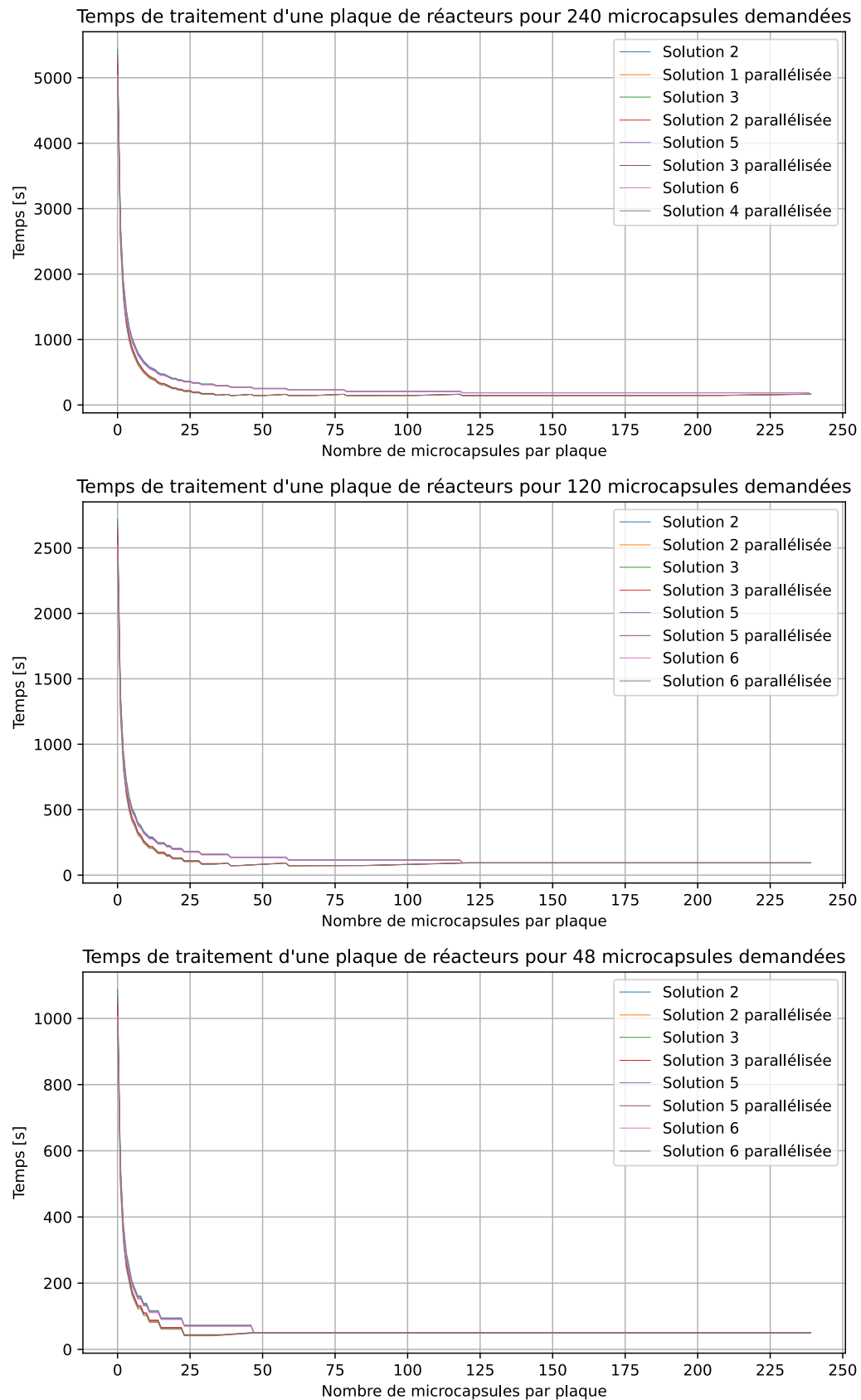


Figure 3.4 – Comparaison du temps de cycle des solutions en fonction du nombre de microcapsules par plaque

Sur Figure 3.4, il est possible de voir que les solutions parallélisées ont toujours un temps de cycle inférieur aux solutions séquentielles tant que le nombre de microcapsules par plaque est inférieur au nombre de microcapsules demandées. Plus le nombre de microcapsules demandées est élevé, plus le gain de temps est significatif, allant, en moyenne, de 1.3 % pour 48 microcapsules à 27 % pour 240 microcapsules.

3.3 Présentation de la solution choisie

L'utilisation d'un seul robot positionné au centre de la *glove box*, qui s'occupera de déplacer les plaques ainsi que de faire le *pick and place* des microcapsules a été retenue. Cette décision est due à :

- un robot déjà présent dans le laboratoire ;
- la grande flexibilité de cette solution (la *glove box* sera utilisée pour d'autres processus par la suite).

3.4 Description du matériel utilisé

Voici la liste de matériel utilisé lors du TB :

- Robot « UR3e » ;
- Pince « Hand-E » ;
- Adaptation de la pince précédente sur mesure ;
- Module d'aspiration ;
- Support pour les plaques.

3.5 Intégration

Afin de maximiser l'espace accessible par le robot, ce dernier a été surélevé (cf. Figure 3.5). La moitié gauche est réservée pour le *pick and place* des microcapsules, tandis que la partie à droite servira pour la suite du processus. Pour les outils, la pince se met sur la bride du robot et elle possède deux trous taraudés qui seront utilisés pour fixer le module d'aspiration (cf. Figure 3.6). L'outil pour saisir les plaques fait un angle à 45° pour pouvoir déposer les plaques sur tout le plan *repPickAndPlace*. Afin de gagner de la place, les deux outils sont placés dans des directions opposées.



Figure 3.5 – Intégration du robot dans la *glove box*

3.5. INTÉGRATION



Figure 3.6 – *Module d'aspiration*

Chapitre 4

Software

4.1 Algorithme de recombinaison

4.1.1 Spécification de l'algorithme

Objectif

L'algorithme a pour objectif de sélectionner les microcapsules, dans le stock, à utiliser pour chaque recette d'un batch, soit environ 300 recettes.

Contraintes

Les contraintes de l'algorithme sont les suivantes :

1. Maximiser le nombre de recettes réalisées.
 - Se situer entre dans la plage de tolérance de quantité pour chaque produit ;
 - respecter la quantité maximale de microcapsules de chaque réacteur.
2. Minimiser le nombre de microcapsules utilisées par réacteur.

Entrées

Il possède en entrée :

- les recettes du batch ;
- les microcapsules présentes dans le stock ;
- le nombre maximal de microcapsule par réacteur.

Sorties

Les sorties de l'algorithme incluent :

- Un tableau contenant pour chaque recette un liste avec les ids des capsules à utilisées ave le numéros de recettes sous la forme : $[(id_1, id_2, \dots, id_m), n^{\circ}recette_1], \dots, [(id_1, id_2, \dots, id_m), n^{\circ}recette_m]$ avec m le nombre de recette ;
- Un tableau contenant la quantité de chaque produit pour chaque recette sous la forme : $(("produit_1" : q_{produit_1}, \dots, "produit_n" : q_{produit_n}, n^{\circ}recette_1), \dots, ("produit_1" : q_{produit_1}, \dots, "produit_n" : q_{produit_n}, n^{\circ}recette_m))$
- Le numéro des recettes qui ne sont pas réalisable
- Les éléments manquants pour réaliser les recettes non réalisable avec la forme : $(("Produit_{manquant\ 1}" : q_{produitmanquant1}, \dots, "Produit_{manquant\ n}" : q_{produitmanquantn}, n^{recipe1}), \dots, ("Produit_{manquant\ 1}" : q_{produitmanquant1}, \dots, "Produit_{manquant\ n}" : q_{produitmanquantn}, n^{recipe_m}))$
- Les éléments manquants pour réaliser les recettes non réalisable avec la forme : $(("Produit_{manquant\ 1}" : q_{produitmanquant1}, \dots, "Produit_{manquant\ n}" : q_{produitmanquantn}, n^{recipe1}), \dots, ("Produit_{manquant\ 1}" : q_{produitmanquant1}, \dots, "Produit_{manquant\ n}" : q_{produitmanquantn}, n^{recipe_m}))$

4.1.2 Définition du problème

Le problème consite à trouver une combinaison de microcapsules pour chaque recette qui maximise le nombre de recettes réalisées.

$$\max \left(\sum_i \text{RecetteRealisée}_i \right) \quad (4.1)$$

Maximisation du nombre de recettes réalisées

Avec :

- N , le nombre de microcapsules dans le stockage, $N \in \mathbb{N}^*$;
- k , le nombre moyen de microcapsules par recette, $k \in \mathbb{N}^*$;

4.1. ALGORITHME DE RECOMBINAISON

— R , le nombre de recette réalisable, $R \in \mathbb{N}^*$.

Le nombre théorique de recettes réalisable est :

$$R_{th} = \left\lfloor \frac{N}{k} \right\rfloor \quad (4.2)$$

L'équation (??), implique une maximisation du nombre théorique de recettes réalisables (??).

$$\max \left(\sum_i \text{RecetteRealisée}_i \right) \Rightarrow \max (R_{th}) \quad (4.3)$$

$$\max(R_{th}) = \max \left(\left\lfloor \frac{N}{k} \right\rfloor \right) \Rightarrow \max(N) \vee \min(k) \quad (4.4)$$

Or, N est constant, donc :

$$\max \left(\sum_i \text{RecetteRealisée}_i \right) \Rightarrow \min(k) \quad (4.5)$$

À des fins de faciliter, l'utilisation de la minimisation de k (??) sera préférée.

Contraintes

La liste des contraintes pour l'optimisateur sont les suivantes :

- La quantité dans chaque réacteur doit être comprises dans la plage souhaitée ;
- le nombre de microcapsules dans un réacteur doit être inférieur ou égal à la quantité maximal de microcapsules dans le réacteur ;
- une microcapsules doit être utilisée une seule fois.

Knapsack problem

« The knapsack problem (KP) can be formally defined as follows : We are given an instance of the knapsack problem with item set N , consisting of n items j with profit P_j and weight W_j , and the capacity value c . (Usually, all these values are taken from the positive integer numbers.) Then the objective is to select a subset of N such that the total profit of the selected items is maximized and the total weight does not exceed c . » (**KnapsackProblemsBook**)

Le problème posé ressemble donc au *knapsack problem*, cependant étant donné qu'il y a plusieurs réacteurs (l'équivalent du sac) le problème est donc plutôt un *Multiple knapsack problem*¹. Il y a encore une nuance entre le problème posé et un *Multiple knapsack problem*, c'est que dans un réacteur, il peut y avoir plusieurs produits dans un seul réacteur. Donc pour chaque batch, il y a plusieurs problèmes du type *Multiple knapsack problem*.

4.1.3 Méthode d'optimisation

Optimisation générale

L'optimisation générale consiste à traiter chaque produit séparément avec certaines contraintes (cf. ??).

Figure 4.1 – Algorithme de l'optimisateur.

1. (**KnapsackProblemsBook**)

Optimisateur

L'approche la plus intuitive pour optimiser le problème consiste à calculer toutes les combinaisons possibles puis de sélectionner la solution qui répond le mieux aux critères définis.

Le nombre de combinaisons C possibles pour k microcapsules et un stock n , se calcul comme suit :

$$C_{k,n} = \frac{n!}{k! \cdot (n - k)!} \quad (4.6)$$

$$C_n = \sum_{k=1}^l C_{k,n} = \sum_{k=1}^l \frac{n!}{k! \cdot (n - k)!} \quad (4.7)$$

Nombre de recombinaison des microcapsules (échelle semi-logarithmique)

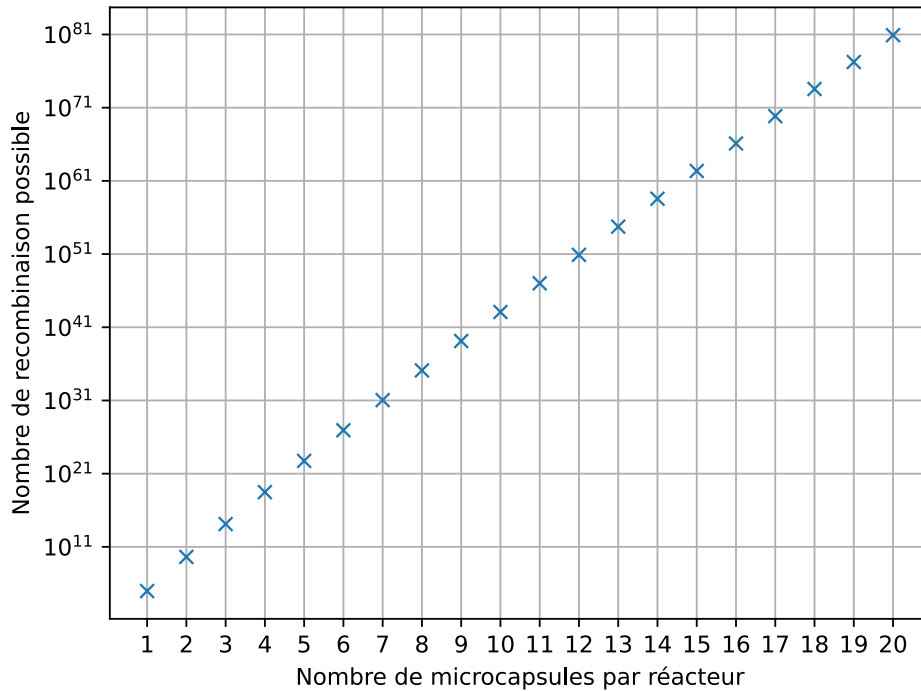


Figure 4.2 – Nombre de combinaisons possible en fonction de la taille des réacteurs

4.1. ALGORITHME DE RECOMBINAISON

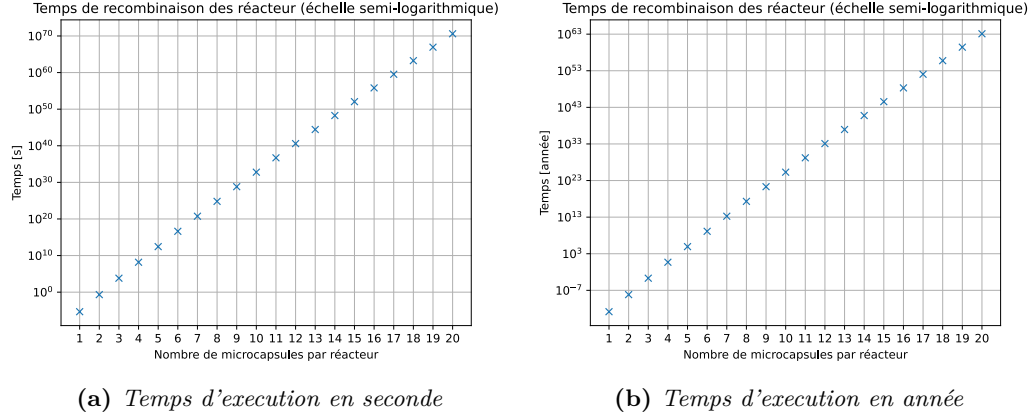


Figure 4.3 – Temps de calcul en fonction de la taille des réacteurs

La ?? montre, en prenant en compte le nombre de combinaisons possible (cf. ??) le nombre des calcul par secondes moyen pour les ordinateurs en 2020 soit environ 10^{11} opérations par secondes, le temps nécessaire pour calculer toutes les combinaisons possibles. Il est possible d'observer que le temps nécessaire pour les combinaisons dépassant x microcapsules maximal par réacteur, devient des durées non concevable pour cette application. Cette approche, bien qu'elle trouve toujours la solution optimale et qu'elle soit facilement compréhensible, n'est pas adapté au projet.

Plusieurs algorithmes existes pour résoudre ce type de problème :

- glouton ;
- programmation dynamique ;
- optimisation linéaire ;
- heuristique ;
- *branch and cut* ;
- optimisation linéaire en nombres entiers ;
- un algorithme génétique.

Le problème peut être interprété comme un problème contraint d'entier (*Constraints Integer Problems* (CIPs)), car la sélection des microcapsules se fait de manière binaire (une microcapsule est sélectionner ou non). Pour les CIPs, il existe des frameworks (notamment SCIP (*Solver Constraint Integer Programs*)) utilisant certains des algorithmes citées précédemment ???. L'utilisation de SCIP (qui permettra de ne pas implémenter chaque algorithme indépendamment) à travers le wrapper de Google or-tools sera utilisé afin de faciliter l'intégration en python.

L'utilisation de SCIP se fait avec la forme de l'optimisation avec contraintes :

$$\begin{aligned}
 &\min && x \\
 &\text{subject to} && \sum_i (a_i x_i) \leq b \\
 &\text{and} && x \in \mathbb{N}
 \end{aligned}$$

2. petite_analyse_nbrcalculs_par_sec.

ou plus g n ralement :

$$\begin{array}{ll} \min & x \\ \text{subject to} & Ax \leq b \\ \text{and} & x \in \mathbb{N} \end{array}$$

Avec b un vecteur et A une matrice. Pour utiliser l'optimisateur, il faut d finir la fonction de co t (cf. ??)   minimiser, et les contraintes (cf. ??).

Matrice de d cision

L'optimisateur doit utilis  un vecteur de d cision pour optimiser l'utilisation des microcapsules. Pour une recette le vecteur \vec{x} est d finie :

$$x_i \in \{0, 1\}, \forall i \in \{1, 2, \dots, n\} \text{ avec } n \text{ le nombre de microcapsules en stock.} \quad (4.8)$$

Ce vecteur (??) est valable pour 1 seule recette. Id alement pour plusieurs recettes, il faudrait une matrice X (cf. ??), qui est par la suite « vectoris e » afin d'obtenir le vecteur de d cision final (cf. ??).

$$X = \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,n} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,0} & x_{m,1} & \cdots & x_{m,n} \end{bmatrix} \quad (4.9)$$

$$\vec{x} = [x_{0,0}, \dots, x_{0,n}, x_{1,0}, \dots, x_{1,n}, x_{m,0}, \dots, x_{m,n}], \quad (4.10)$$

avec n le nombre de microcapsules, et m le nombre de recettes.

Matrices de contraintes

Pour d finir les deux matrices de contraintes (A et b), il faut commencer par d finir b car la structure de A en d pendra. b est un vecteur colonne qui comprends pour chaque recette :

1. La quantit  maximale souhait e (not  Q_{max} suivi du num ro de recette).
2. La quantit  minimale souhait e (not  Q_{min} suivi du num ro de recette).
3. Le nombre maximal de microcapsule par r acteur (not  l).

Puis, s'en suit une colonne de n ligne de 1, correspondant aux nombre de fois qu'une microcapsule peut  tre utilis .

$$b = \begin{bmatrix} Q_{max}1 \\ -Q_{min}1 \\ l \\ \vdots \\ Q_{max}n \\ -Q_{min}n \\ l \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad (4.11)$$

4.1. ALGORITHME DE RECOMBINAISON

En sachant que $Ax \leq b$, il est possible d'en déduire que A sera décomposée en sous matrices (des matrices identité I_n et une autre matrice nommée m_1).

$$A = \begin{bmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_1 \\ I_n & I_n & \cdots & I_n \end{bmatrix} \quad (4.12)$$

Pour la composition de m_1 , la première ligne sert à déterminer si la somme de la quantité des capsules sélectionner est inférieur à la quantité limite, la deuxième à déterminer si la somme de quantité des microcapsules sélectionnées est supérieur à la quantité minimale et la dernière ligne est présente pour vérifier que la somme des microcapsules sélectionner ne dépasse pas la limite de microcapsule maximum par réacteur.

$$m_1 = \begin{bmatrix} Q_1 & Q_2 & \cdots & Q_n \\ -Q_1 & -Q_2 & \cdots & -Q_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \quad (4.13)$$

Fonction de coût

L'objectif de l'optimisation est de réduire le nombre de microcapsules utilisé, avec x le vecteur de décision, le fonction de coût $f(\vec{x})$ est donc :

$$f(\vec{x}) = \sum \vec{x} \quad (4.14)$$

Cependant en utilisant la fonction (cf. ??), si une seule des recettes du batch n'est pas réalisable, l'optimisateur retournera le fait que le problème n'est résoluble, sans fournir les résultats des recettes dont il a trouvé des solutions. Pour résoudre ce problème, l'ajout d'une *slack variable*, nommé z est indispensable. Cette variable prendra la quantité manquante de certaines recettes non réalisables. Afin de ne pas tomber dans l'utilisation excessive de \vec{z} , il est nécessaire de rendre le coût de celle-ci plus importante grâce à un ratio α . La fonction de coût définitive devient :

$$f(\vec{x}) = \sum (\vec{x} + \alpha \vec{z}) \quad (4.15)$$

Avec α définit arbitrairement à 10^4 .

4.2 Robot

4.2.1 Objectif

Le robot doit manipuler les microcapsules en effectuant des tâches de *pick and place* depuis leur zone de stockage jusqu'aux réacteurs. Il doit également être capable de manipuler des plaques de microcapsules et de « Para-dox ».

4.2.2 Contraintes

1. Respect des limites de la *glove box* pour éviter toute collision ;
2. Gestion de deux outils (pince et module d'aspiration) sans changement manuel ;

4.2.3 Méthode de programmation

La programmation est effectuée en deux parties :

- Les programmes intégrés au robot gèrent les déplacements : « PickAndPlaceVial », « TakePlate », et « GivePlate » ;
- Un programme externe en Python appelle les fonctions du robot et transmet les informations nécessaires via l'interface RTDE (cf. ??).

4.2.4 Outils

Les outils nécessaires pour réaliser la manipulation sont :

- une pince pour saisir les *well plate* et les plaques « Para-dox » ;
- un module pour l'aspiration des microcapsules.

Une pointe a également été fabriquée, pour faciliter la programmation des différents repères nécessaires.

4.2.5 Sécurité

Limites de sécurité

Les robots d'Universal Robot permettent de configurer des plans de sécurité, qui ne seront pas franchissables par le TCP du robot. Dans ce cas, deux limites des sécurités sont nécessaires (une limite par porte de la *glove box*). Pour définir ces limites, il faut :

1. Définir un point en mettant l'outil perpendiculaire au plan souhaité.
2. Définir un plan de sécurité qui passe par ce point et qui sera normal à l'outil du robot.
3. Définir la sécurité à mettre en place (Normal, réduit, les deux ou mode de déclenchement réduit). Dans notre cas, nous ne voulons pas que le TCP dépasse ces plans, le mode « les deux » doit donc être appliqué.

4.2.6 Repères de travail

Deux repères seront utilisés :

- le repère de base qui sera utilisé pour déplacer les plaques ;
- le repère « repPickAndPlace » (cf. ??), sera utilisé pour le déplacement des microcapsules.

4.3. INTERACTION ENTRE LE LOGICIEL ET LE MATÉRIEL POUR LA RECOMBINAISON



Figure 4.4 – Position physique du plan « repPickAndPlace »

4.2.7 Communication avec le robot

L'interface *Real-Time Data Exchange* (RTDE) permet l'échange bidirectionnel en temps réel des données entre le robot et le système externe, facilitant ainsi le démarrage des programmes robot et la lecture/écriture des registres nécessaires.

4.3 Interaction entre le logiciel et le matériel pour la recombinaison

Chapitre 5

Résultat et Analyse

5.1 Performance du système

5.2 Analyse des erreurs

Chapitre 6

Discussion

Chapitre 7

Conclusion

Gaëtan Worch

Annexes

