TP

Intelligence artificielle
Robot pick & place

Problème technique

* Caractériser le comportement d'un robot pick & place utilisé dans une chaîne de conditionnement alimentaire : performances de l'asservissement en position et reconnaissance de biscuits secs.

Compétences à mettre en œuvre

* Analyser les principes d'intelligence artificielle.
* Extraire un indicateur de performance pertinent à partir du cahier des charges ou de résultats issus de l'expérimentation ou de la simulation.
* Choisir une démarche de résolution d'un problème d'ingénierie numérique ou d'intelligence artificielle.
* Proposer une démarche permettant d'évaluer les performances des systèmes asservis.
* Résoudre un problème en utilisant une solution d'intelligence artificielle.

1. Mise en situation

L'industrie agroalimentaire est très concurrentielle. Afin de produire et d'emballer de manière rentable des produits différents, les systèmes automatisés sont indispensables. Ils doivent être efficaces, fiables et modulaires.

La solution est très souvent l'utilisation de la technologie pick & place consistant à prélever le produit en sortie de chaîne de fabrication pour le positionner dans son emballage.

Figure 1 – Assortiments de biscuits secs rangés dans leur barquette

L'étude suivante porte sur une chaîne robotisée de conditionnement de biscuits secs.
L'activité est divisée en quatre parties comme le montre la figure suivante.

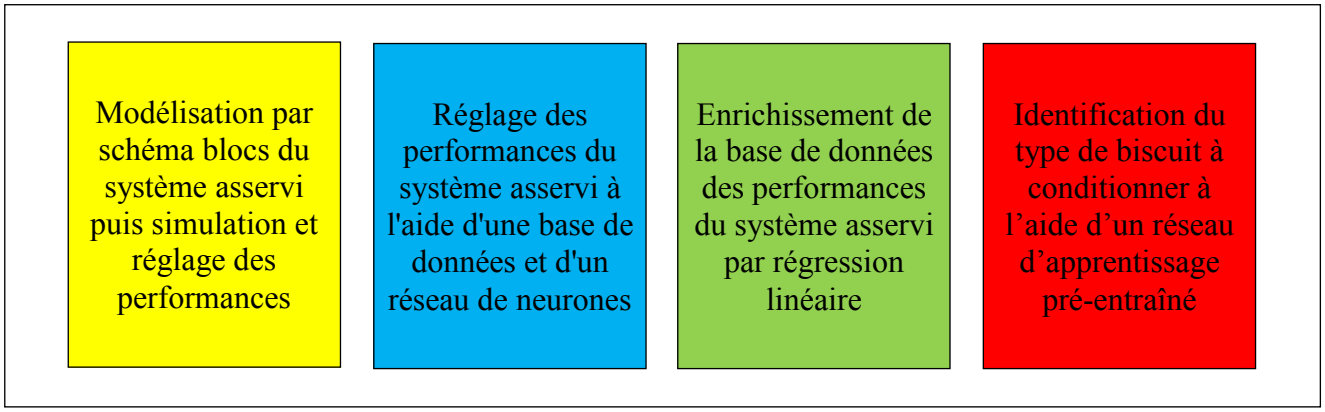


Figure 2 – Activités relatives au système asservi de positionnement des biscuits

1.1. Chaîne de conditionnement des biscuits

Lorsque les biscuits sont façonnés et cuits, ils sont nappés ou fourrés de chocolat puis passent dans un tunnel de refroidissement. Ensuite, les biscuits sortent sur un large tapis roulant prêts pour le conditionnement.

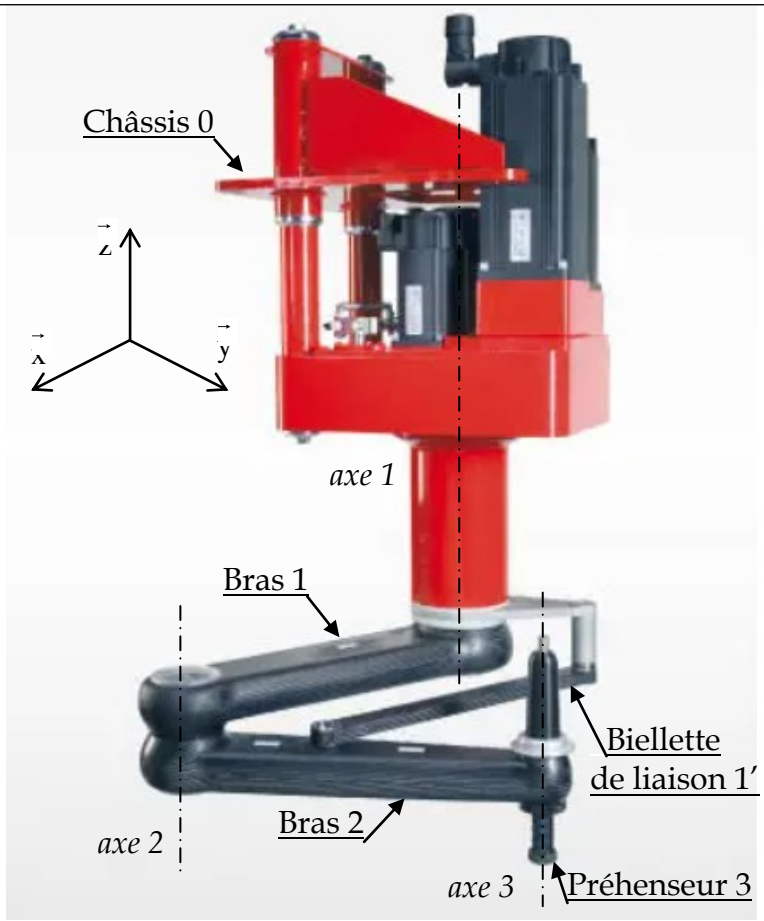
Figure 3 – Sortie de la chaîne de refroidissement des biscuits

Le remplissage des boîtes d'assortiments contenant des biscuits aux formes multiples, est réalisé de manière automatisé. Grâce à une caméra, le robot détecte le biscuit sur la ligne de production puis le positionne dans le logement prévu de la barquette.

Figure 4 – Chaîne de conditionnement des biscuits secs

1.2. Robot SCARA F4

Le dispositif pick & place comporte notamment un robot de type SCARA F4 à quatre axes qui, en coopération avec le système de vision, prélève les biscuits sur la bande transporteuse en marche continue et les regroupe dans une barquette de conditionnement.



Caractéristiques	
Type	SCARA F4
Nombre d'axes	4 (rotations axes 1 + 2 + 3 et translation axe 3)
Fonction	Pick and place
Autres caractéristiques	Système de vision Préhenseur interchangeable
Vitesse	Jusqu'à 4,7 m/s
Accélération	Jusqu'à 10 g
Niveau de bruit	70 dB

Figure 5 – Robot SCARA modèle F4 fabriqué par l'entreprise allemande Schubert

SCARA est l'acronyme de Selective Compliance Articulated Robot Arm. La structure du robot est composée de deux bras articulés à leurs extrémités.

Trois moteurs électriques pilotés indépendamment (axes 1, 2 et 3) permettent de contrôler le mouvement du préhenseur dans le plan $\left(\vec{x}, \vec{y}\right)$. L'emplacement final du biscuit dans le plan $\left(\vec{x}, \vec{y}\right)$ à l'extrémité du bras 2 est fonction de l'angle θ_1 (entre le châssis 0 et le bras 1), de l'angle θ_2 (entre le bras 1 et le bras 2), de l'angle θ_3 (entre le bras 2 et le préhenseur 3), de la longueur du bras 1 et de la longueur du bras 2. Les axes d'articulations du SCARA sont parallèles à l'axe $\vec{z}$. Le dernier axe de translation suivant $\vec{z}$ est assuré par un vérin pneumatique.

Les robots SCARA F4 Schubert peuvent travailler en équipe et poursuivre mutuellement l'objectif de garantir des emballages entièrement remplis. Pour atteindre une cadence plus élevée, le nombre de robots peut être simplement augmenté.
Le système est complété par des désempileurs de barquettes et des convoyeurs à chaîne.

Figure 6 – Équipe de robots pick & place SCARA F4 pour l'emballage des biscuits

1.3. Pick & place avec contrôle qualité intégré

Les biscuits entrent dans la machine d'emballage via une large bande. Chaque machine pick & place est équipée d'un traitement d'image industriel. Le système de vision détecte la position, la forme et la qualité des biscuits sur la bande transporteuse en flux continu. Afin de garantir des biscuits de haute qualité, seuls les produits conformes aux spécifications programmées sont approuvés pour l'emballage.

- Balayage télécentrique de la bande de produits
- Vue du produit sans distorsion d'image
- Contrôle exact des formes, dimensions et couleurs
- Taille modulaire du scanner de 200 à 1800 mm
- Cadence jusqu'à 5000 produits /minute

Figure 7 - Scanner à lumière incidente pour contrôle de la couleur et de la surface

1.4. Systèmes de préhension

Les robots SCARA F4 Schubert saisissent les biscuits avec précision sur la bande transporteuse et les déposent dans les barquettes de conditionnement.

Les outils de préhension et d'aspiration facilement interchangeables permettent de traiter une large gamme de produits et de formats.

Figure 8 – Différentes pinces et ventouses de préhension des biscuits

1.5. Chaînes fonctionnelles du système pick & place

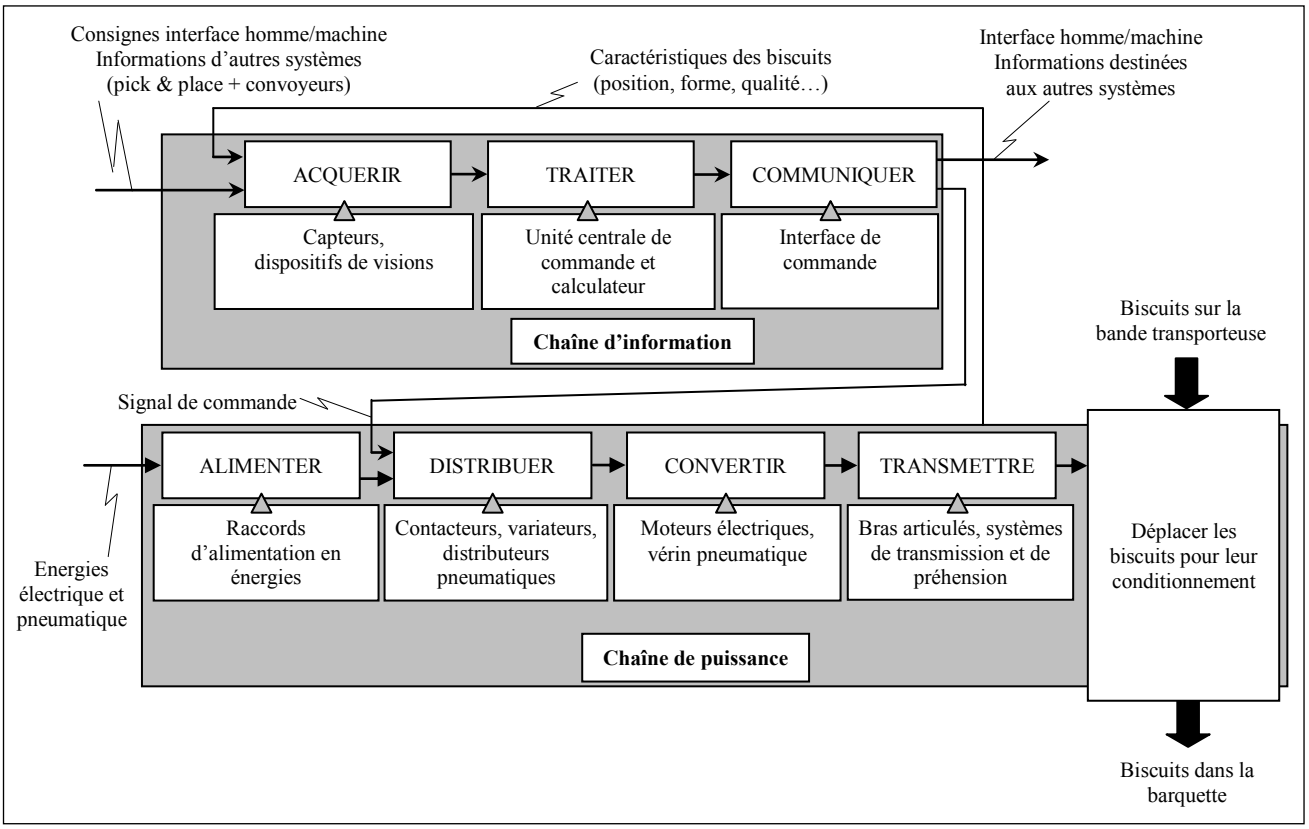


Figure 9 – Chaîne d'information et chaîne de puissance du système pick & place

1.6. Liens vidéos pour découvrir le robot en situations de fonctionnement

Vidéo N°1 : Rangement de biscuits dans des barquettes

Vidéo N°2 : Conditionnement de biscuits doubles

Vidéo N°3 : Conditionnement d'assortiments de chocolats

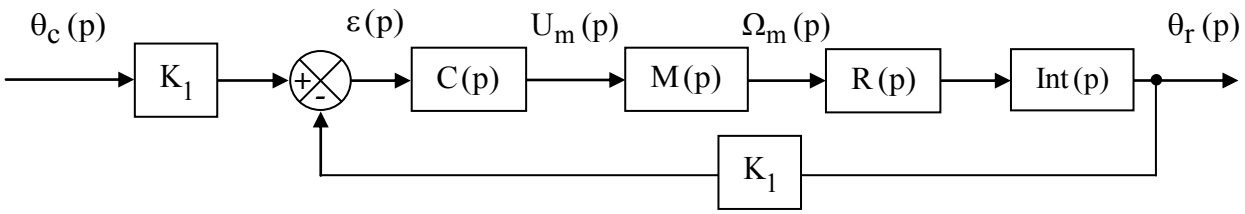
Vidéo N°4 : Emballage de pâtisseries

2. Gestion asservie de l'axe 1 du robot

Objectif : Déterminer les réglages de la commande asservie du bras 1 du robot permettant d'assurer le déplacement demandé du biscuit.

2.1. Modélisation du système asservi et extrait du cahier des charges

Pour une configuration particulière d'essai de fonctionnement, le modèle simplifié de pilotage est présenté ci-dessous.
La maquette numérique correspondante réalisée avec le logiciel *Matlab-Simulink* est disponible dans le fichier *Moteur_robot_pick_and_place.slx*.



$\theta_c(p)$: consigne angulaire de rotation du bras 1 (50°)

K_1 : gain équivalent à l'adaptateur de consigne et au capteur de position angulaire

$C(p) = P + I.\left(\frac{1}{p}\right)$: correcteur Proportionnel Intégral PI

$U_m(p)$: tension de commande du motovariateur (V)

$M(p)$: motovariateur

$\Omega_m(p)$: vitesse de rotation du motovariateur (rad/s)

$R(p) = 1$: système de transmission et convertisseur d'unité

$Int(p)$: intégrateur

$\theta_r(p)$: réponse angulaire de rotation du bras 1 (°)

Figure 10 – Schéma bloc simplifié du pilotage du bras 1 pour la configuration étudiée

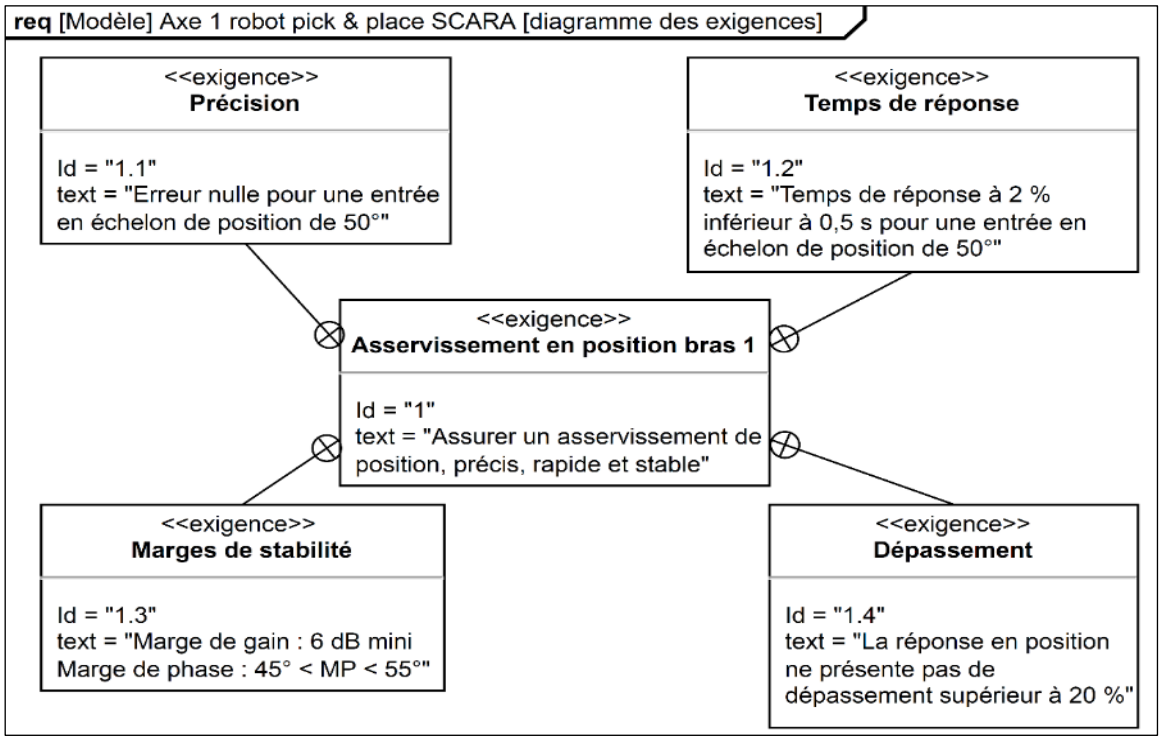


Figure 11 – Diagramme partiel des exigences (extrait du cahier des charges)

2.2. Réglage du correcteur PI à l'aide d'un modèle virtuel

Problème à résoudre : Déterminer les réglages numériques du correcteur PI afin de satisfaire les exigences du cahier des charges.

La résolution du problème s'appuiera sur l'utilisation du logiciel *Matlab-Simulink* et de l'élément de la bibliothèque *Function Block Parameters* intitulé *PID Controller*.

Une lecture attentive du tutoriel *Matlab PID Tuner* est recommandée.

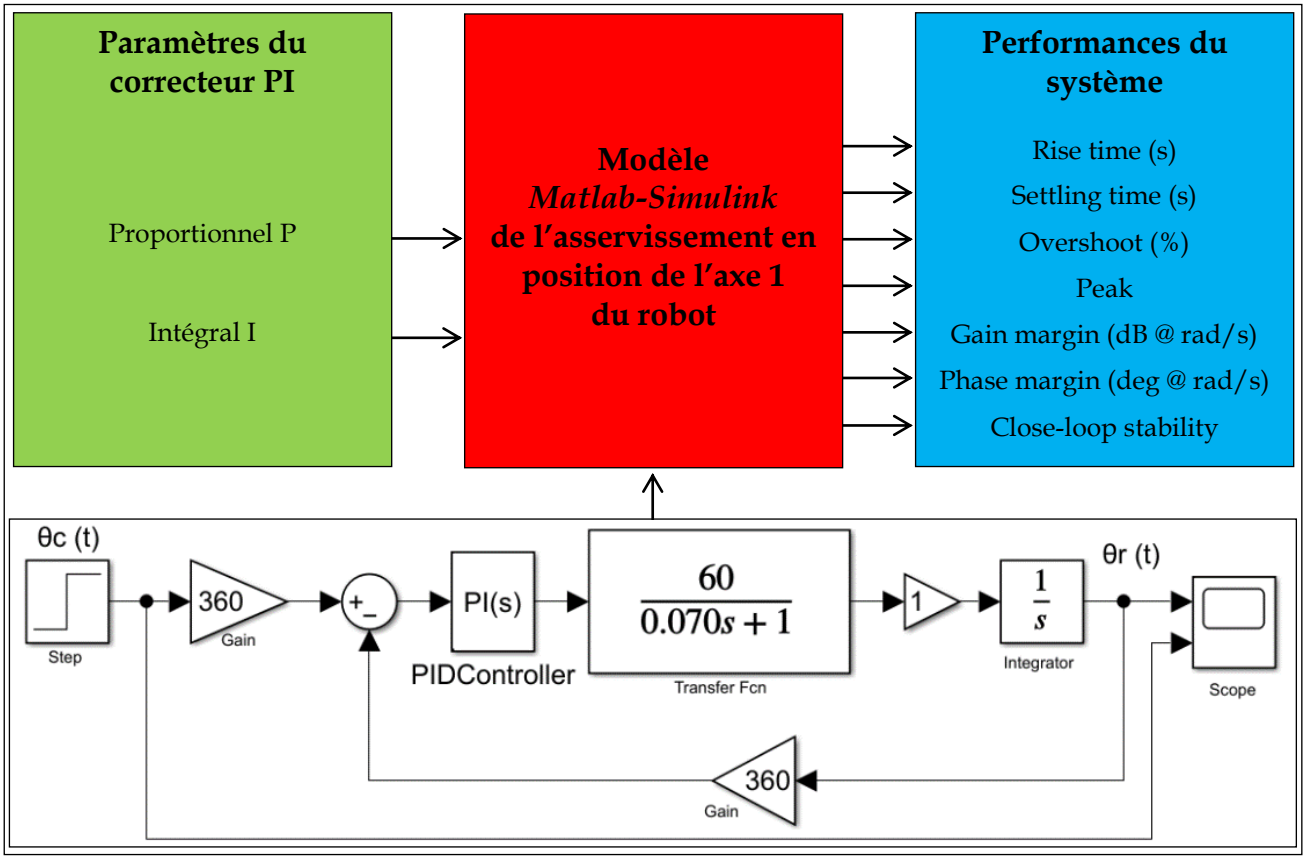


Figure 12 – Paramètres de réglage du modèle du robot et performances obtenues

Q1. Ouvrir le fichier *Moteur_robot_pick_and_place.slx*. Lancer une simulation avec les paramètres par défaut du correcteur PI (P = 1 et I = 1). Commenter brièvement les performances du système par rapport aux exigences du cahier des charges.

Q2. Avec l'application *PID Tuner App* associé au bloc correcteur *PID Controller*, rechercher les réglages optimaux des paramètres P et I permettant de satisfaire les exigences du cahier des charges. Compléter le tableau ci-dessous et rédiger quelques commentaires.

		Essai 1	Essai 2
Controller Parameters	P		
	I		
Performance and Robustness	Rise time 10 % - 90 % (s)		
	Settling time 2 % (s)		
	Overshoot (%)		
	Peak		
	Gain margin (dB @ rad/s)		
	Phase margin (deg @ rad/s)		
	Close-loop stability		

Figure 13 – Réglages du correcteur et performances du système

2.3. Réglage du correcteur à l'aide d'un réseau de neurones artificiels

Problème à résoudre : À partir d'une base de données relative aux performances du bras 1 du robot, déterminer les réglages des coefficients *P* et *I* du correcteur pour satisfaire les exigences du cahier des charges.

La résolution du problème s'appuiera sur l'utilisation du logiciel *Matlab-Simulink* et de l'application *Neural Net Fitting* (*Machine Learning and Deep Learning*).

Une lecture attentive du tutoriel *Matlab Neural Net Fitting* est recommandée.

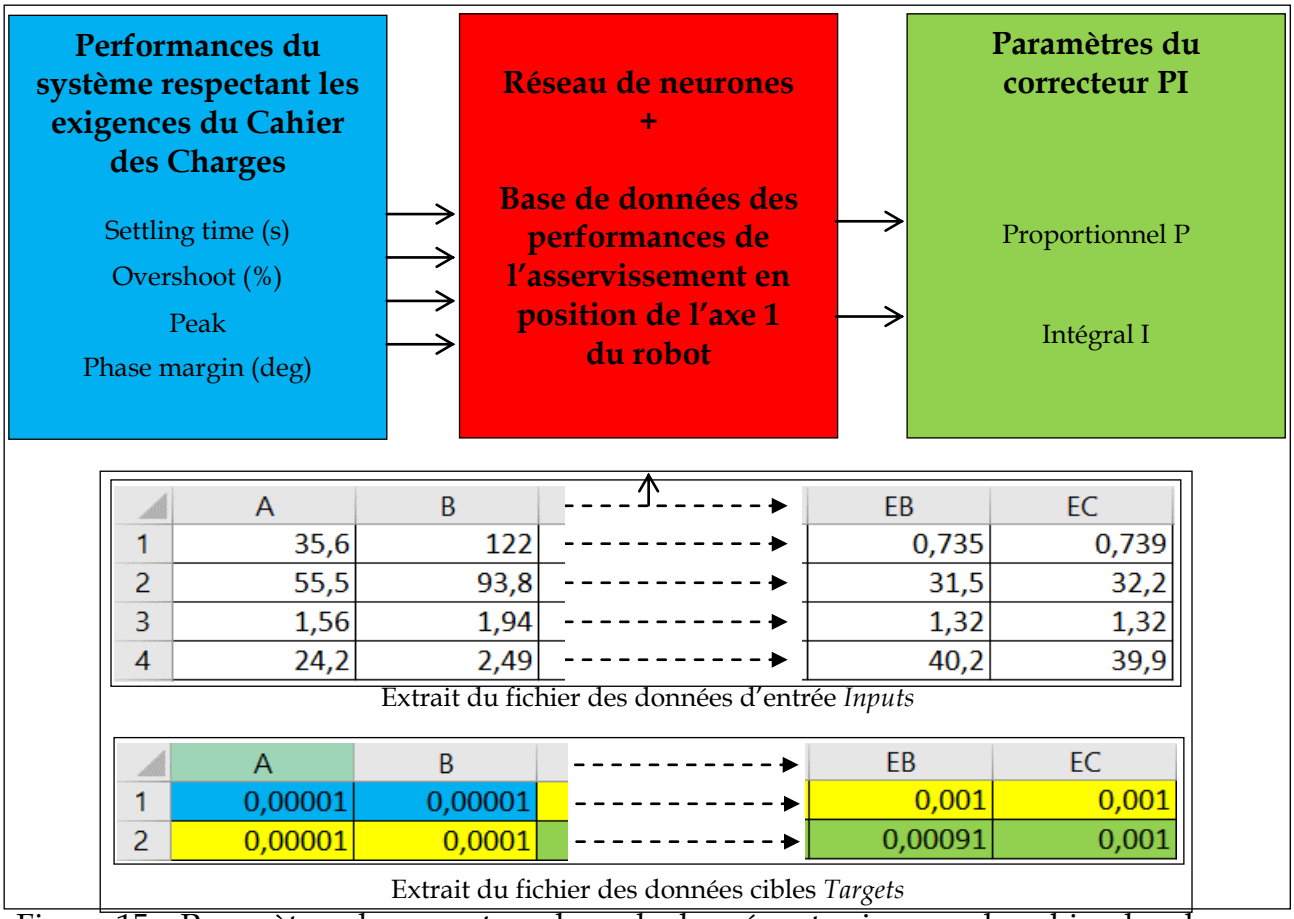
La base de données du bras du robot comporte un tableau de 8 lignes et 133 colonnes.
Elle est disponible en intégralité dans le fichier *Base de données Robot.xlsx*.
Pour différentes valeurs des paramètres *P* et *I* fixés, les performances de sortie sont relevées :

	A	B	C		EB	EC	ED
1	**P**	0,00001	0,00001		0,001	0,001	0,001
2	**I**	0,00001	0,0001		0,00082	0,00091	0,001
3	**Rise time 10 % - 90 % (s)**	2,22	0,71		0,0809	0,0807	0,0804
4	**Settling time 2 % (s)**	35,6	122		0,729	0,735	0,739
5	**Overshoot (%)**	55,5	93,8		30,9	31,5	32,2
6	**Peak**	1,56	1,94		1,31	1,32	1,32
7	**Phase margin (°)**	24,2	2,49		40,6	40,2	39,9
8	**Pulsation (rad/s)**	0,49	1,47		14,9	14,9	14,9

Figure 14 – Extrait de la base de données

Afin de déterminer les paramètres *P* et *I* du correcteur pour satisfaire les exigences du cahier des charges, cette base de données est séparée en deux :

- un fichier des données d'entrée *Inputs.xlsx* (*Settling time*, *Overshoot*, *Peak* et *Phase margin*),
- un fichier des données cibles *Targets.xlsx* (paramètres du correcteur *P* et *I*).



	A	B		EB	EC
1	35,6	122		0,735	0,739
2	55,5	93,8		31,5	32,2
3	1,56	1,94		1,32	1,32
4	24,2	2,49		40,2	39,9

Extrait du fichier des données d'entrée *Inputs*

	A	B		EB	EC
1	0,00001	0,00001		0,001	0,001
2	0,00001	0,0001		0,00091	0,001

Extrait du fichier des données cibles *Targets*

Figure 15 – Paramètres du correcteur, base de données et exigences du cahier des charges

Q3. Ouvrir l'application *Neural Net Fitting* et saisir les fichiers des données d'entrées et cibles à utiliser. Réaliser plusieurs entraînements pour différents nombres de neurones cachés (10, 500, 800, 1000 et 3000). Relever les valeurs demandées dans les tableaux ci-dessous. Commenter les résultats obtenus.

Number of Hidden Neurons = 10					
Entraînement	N°1	N°2	N°3	N°4	N°5
Training (R =)					
Validation (R =)					
Test (R =)					
Plot Regression All (R =)					

Number of Hidden Neurons = 500					
Entraînement	N°1	N°2	N°3	N°4	N°5
Training (R =)					
Validation (R =)					
Test (R =)					
Plot Regression All (R =)					

Number of Hidden Neurons = 800					
Entraînement	N°1	N°2	N°3	N°4	N°5
Training (R =)					
Validation (R =)					
Test (R =)					
Plot Regression All (R =)					

Number of Hidden Neurons = 1000					
Entraînement	N°1	N°2	N°3	N°4	N°5
Training (R =)					
Validation (R =)					
Test (R =)					
Plot Regression All (R =)					

Number of Hidden Neurons = 3000					
Entraînement	N°1	N°2	N°3	N°4	N°5
Training (R =)					
Validation (R =)					
Test (R =)					
Plot Regression All (R =)					

Q4. D'après les résultats précédents, indiquer approximativement le nombre de neurones cachés afin de former le réseau optimal pour résoudre le problème demandé. Lancer un entraînement, puis enregistrer le *Simulink Diagram* correspondant.

Q5. Saisir en données d'entrée du *Simulink Diagram* les exigences du cahier des charges (*Settling time = 0,5 s ; Overshoot* = 20 % ; *Peak* = 1,2 et *Phase margin* = 50°). Lancer la simulation et relever les 2 valeurs de sortie du réseau de neurones (paramètres du correcteur *P* et *I)*.

Q6. Ouvrir le fichier *Moteur_robot_pick_and_place.slx*. Saisir les paramètres *P* et *I* du correcteur déterminés à la question précédente dans le bloc *PID Controller*. Lancer une simulation et vérifier que les performances du système respectent les exigences du cahier des charges.

3. Enrichissement de la base de données

Objectif : Compléter partiellement la base de données existante relative aux performances du système asservi pour de nouvelles valeurs *P* et *I* du correcteur.

3.1. Régression linéaire

Problème à résoudre : Compléter, par régression linéaire, la base de données des performances du système asservi pour de nouvelles valeurs *P* et *I* du correcteur.

La résolution du problème s'appuiera sur l'utilisation d'algorithmes de régression linéaire disponibles dans l'application *Matlab Regression Learner*.

Une lecture attentive du tutoriel *Matlab Regression Learner* est recommandée.

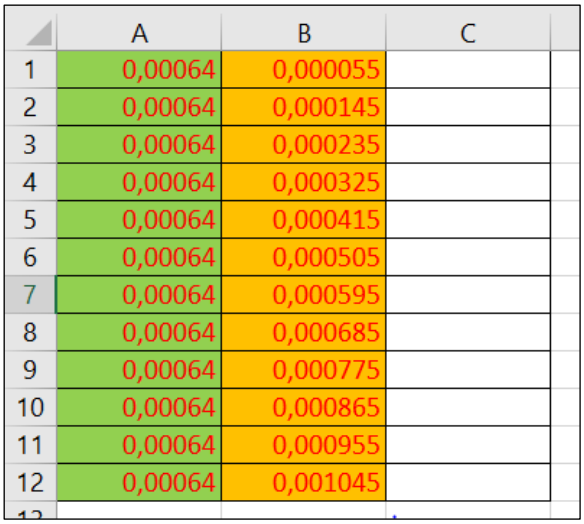
Plusieurs dossiers et fichiers sont mis à disposition dans le dossier *Enrichissement BdD* :

Dossiers	Fichiers
1 – Recherche Rise time	*1 Modele C123.xlsx*
	2 Recherche C3bis.xlsx
	3 Resultat applic C3bis.xlsx
2 – Recherche Settling time	*1 Modele C1234.xlsx*
	2 Recherche C4bis.xlsx
	3 Resultat applic C4bis.xlsx
3 – Recherche Overshoot	*1 Modele C12345.xlsx*
	2 Recherche C5bis.xlsx
	3 Resultat applic C5bis.xlsx
4 – Recherche Peak	*1 Modele C123456.xlsx*
	2 Recherche C6bis.xlsx
	3 Resultat applic C6bis.xlsx
5 – Recherche Phase margin	*1 Modele C1234567.xlsx*
	2 Recherche C7bis.xlsx
	3 Resultat applic C7bis.xlsx
6 – Recherche Pulsation	*1 Modele C12345678.xlsx*
	2 Recherche C8bis.xlsx
	3 Resultat applic C8bis.xlsx

	A	B	C
1	0,00064	0,00001	0,118
2	0,00064	0,0001	0,116
3	0,00064	0,00019	0,115
4	0,00064	0,00028	0,114
5	0,00064	0,00037	0,113
6	0,00064	0,00046	0,112
7	0,00064	0,00055	0,111
8	0,00064	0,00064	0,11
9	0,00064	0,00073	0,109
10	0,00064	0,00082	0,108
11	0,00064	0,00091	0,107
12	0,00064	0,001	0,107
13			

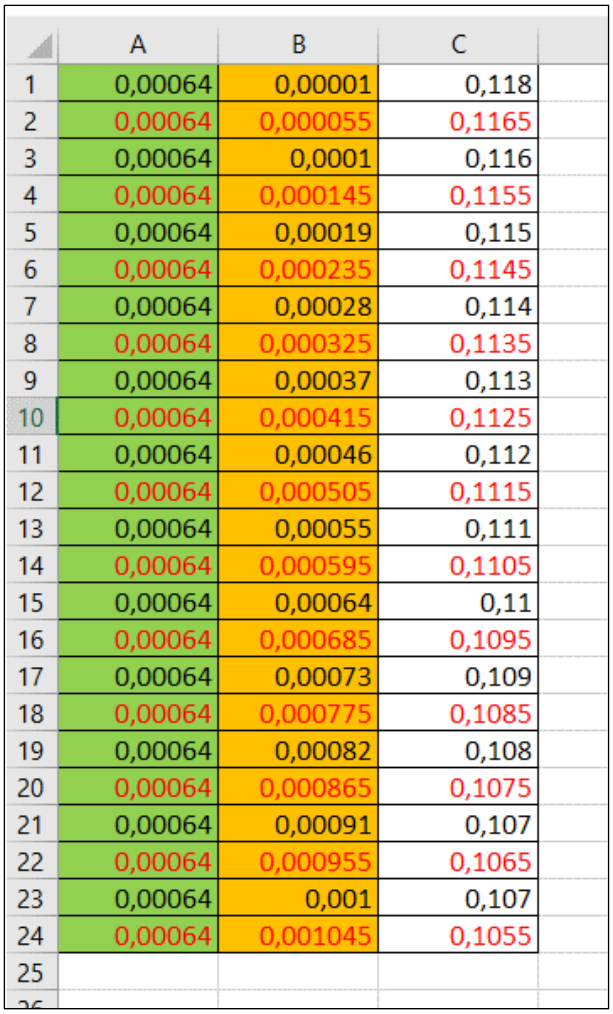
Figure 16 – Fichier intitulé *1 Modele C123.xlsx* qui comprend les valeurs numériques issues de la simulation du système (*Simulink*). Il comporte 3 colonnes de couleur noire A, B et C pour respectivement *P*, *I* et *Rise time*.

Figure 17 – Fichier intitulé *2 Recherche C3bis.xlsx* qui comprend les nouvelles valeurs numériques *P* et *I* du correcteur pour lesquelles on recherche les performances du système *Rise time*. Il ne comporte que 2 colonnes de couleur rouge A et B pour respectivement *P* et *I*.



	A	B	C
1	0,00064	0,000055	
2	0,00064	0,000145	
3	0,00064	0,000235	
4	0,00064	0,000325	
5	0,00064	0,000415	
6	0,00064	0,000505	
7	0,00064	0,000595	
8	0,00064	0,000685	
9	0,00064	0,000775	
10	0,00064	0,000865	
11	0,00064	0,000955	
12	0,00064	0,001045	

Figure 18 – Fichier complété avec l'application *Matlab Regression Learner* intitulé *3 Resultat applic C3bis.xlsx*. Il comprend les valeurs numériques de *Rise time* issues de la simulation (couleur noire) et de l'application *Regression Learner* (couleur rouge).



	A	B	C
1	0,00064	0,00001	0,118
2	0,00064	0,000055	0,1165
3	0,00064	0,0001	0,116
4	0,00064	0,000145	0,1155
5	0,00064	0,00019	0,115
6	0,00064	0,000235	0,1145
7	0,00064	0,00028	0,114
8	0,00064	0,000325	0,1135
9	0,00064	0,00037	0,113
10	0,00064	0,000415	0,1125
11	0,00064	0,00046	0,112
12	0,00064	0,000505	0,1115
13	0,00064	0,00055	0,111
14	0,00064	0,000595	0,1105
15	0,00064	0,00064	0,11
16	0,00064	0,000685	0,1095
17	0,00064	0,00073	0,109
18	0,00064	0,000775	0,1085
19	0,00064	0,00082	0,108
20	0,00064	0,000865	0,1075
21	0,00064	0,00091	0,107
22	0,00064	0,000955	0,1065
23	0,00064	0,001	0,107
24	0,00064	0,001045	0,1055
25			

Q7. Après avoir suivi le tutoriel proposé pour déterminer les nouvelles valeurs de *Rise time*, compléter le fichier *2 Recherche C4bis.xlsx* avec les valeurs trouvées (pas forcément identiques à celles du tutoriel). Ce fichier devra comporter 3 colonnes (*P*, *I* et *Rise time*) de couleur rouge. Avec l'application *Regression Learner*, importer les données d'apprentissage et de recherche (*1 Modele C1234.xlsx* et *2 Recherche C4bis.xlsx*) afin de déterminer les nouvelles valeurs *Settling time*. Entraîner les différents modèles de régression. Choisir le plus performant, puis compléter le fichier *3 Resultat applic C4bis.xlsx* avec les valeurs numériques obtenues. Ce fichier comporte 4 colonnes (*P*, *I*, *Rise time* et *Settling time*).

Q8. Reproduire la démarche précédente pour compléter les autres dossiers :

- *3 – Recherche Overshoot*
- *4 – Recherche Peak*
- *5 – Recherche Phase margin*
- *6 – Recherche Pulsation*

3.2. Confrontation des résultats

Problème à résoudre : Comparer les performances du système déterminées à l'aide de l'application de régression linéaire (*Regression Learner*) et celles obtenues par simulation (modèle *Simulink*).

Q9. Compléter le fichier *Excel* intitulé *2 – Confrontation Résultats Applic vs Simu.xlsx*. Observer les écarts entre les performances obtenues par les deux méthodes. Rédiger quelques commentaires sur l'enrichissement de la base de données par régression linéaire et par simulation.

4. Identification du type de biscuit

Objectif : Identifier le type de biscuit sec pour un conditionnement adapté.

Les systèmes classiques de vision pick & place sont basés sur le contrôle au moyen d'un traitement d'images conventionnel des produits avec quelques paramètres seulement. Pour les produits dits "vivants", comme les biscuits, présentant des tolérances de production plus élevées, les divergences sont si importantes qu'elles ne peuvent être décrites que difficilement ou pas du tout sur le plan mathématique. Le logiciel devrait vérifier de trop nombreux paramètres corrélés et interdépendants. C'est pourquoi le système de contrôle des biscuits choisi s'appuie sur une solution d'intelligence artificielle pour son système de vision. Selon ce concept, le système de traitement d'images apprend par lui-même et trie les produits automatiquement en catégories, au cours d'un processus d'apprentissage. Cela signifie que les paramètres sont sélectionnés par le système et non plus par le programmeur.

Problème à résoudre : Classer des images de biscuits secs en utilisant un réseau pré-entraîné basé sur une solution d'intelligence artificielle

La résolution du problème s'appuiera sur l'utilisation d'un algorithme disponible dans l'application *Deep Network Designer*.

Une lecture attentive du tutoriel *Matlab Deep Network Designer* est recommandée.

Les photos des biscuits nécessaires à l'activité sont mises à disposition dans le dossier *Photos biscuits apprentissage*.

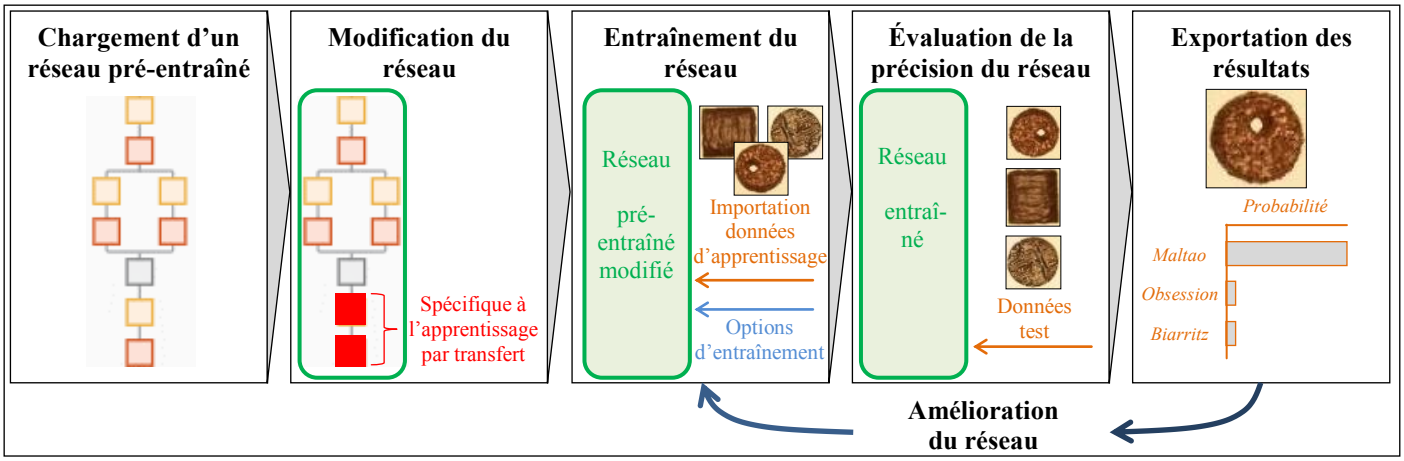


Figure 19 – Démarche d'apprentissage à l'aide d'un réseau pré-entraîné

 Délichoc noir : biscuit croquant recouvert d'une tablette de chocolat noir croustillant	 Délichoc lait : biscuit croquant recouvert d'une tablette de chocolat au lait croustillant
 Obsession : biscuit enrobé d'une fine couche de chocolat noir intense	 Cigarette russe : biscuit croustillant à la vanille et à la poudre d'amandes
 Biarritz : biscuit léger nappé de chocolat noir et saupoudré de noix de coco	 Cœur blanc : biscuit sablé cacaoté enrobé de chocolat blanc craquant
 Matadi : gaufrette croustillante enrobée de chocolat noir intense	 Maltao : biscuit très craquant à la pâte caramélisée enrobée de chocolat au lait
 Marquisette : biscuit sablé fondant allié à un fin chocolat noir	 Nordica : gaufrette légère et craquante enrobée d'un chocolat blanc fondant

Figure 20 – Répertoire des biscuits secs

4.1. Importation des données d'apprentissage

Q10. En s'appuyant sur le tutoriel proposé, importer les photos des biscuits secs permettant l'apprentissage du réseau *GoogLeNet*. Indiquer le nombre d'observations et de classes dans l'ensemble de données.

4.2. Vérification et formation du réseau

Q11. Après avoir modifié l'architecture du réseau proposé (voir tutoriel), vérifier qu'il est prêt pour l'apprentissage et relever le nombre de couches utilisées.

Q12. Former le réseau avec les paramètres d'apprentissage proposés ci-dessous. Relever et commenter les caractéristiques obtenues.

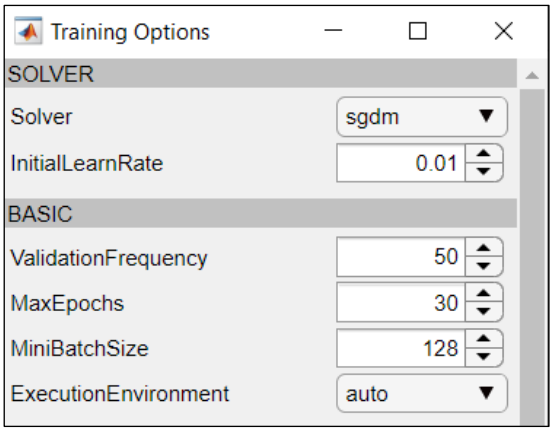


Figure 21 – Paramètres d'entraînement du réseau

4.3. Test du modèle d'apprentissage

Q13. En utilisant les photos de test mises à disposition dans le dossier *Photos biscuits à identifier*, vérifier la qualité des réponses proposées par le réseau construit précédemment. Rédiger quelques commentaires et proposer éventuellement des pistes d'amélioration du modèle d'apprentissage.
Remarque : les biscuits tests sont photographiés sur un fond rouge pour éviter la confusion avec les biscuits d'entraînement photographiés sur un fond clair.

Figure 22 – Exemples de photos de biscuits à identifier