Concours e3a-Polytech, France

Session 2024, filière MP

Sciences industrielles de l'ingénieur

Durée : 4 heures

Ce que traite le sujet

- Partie I : déroulement d'une préparation de commande
- Partie II : trajectoire du robot en phase de déplacement au sol
- Partie III : sollicitation du robot en mouvement vertical
- Partie IV : asservissement de l'assiette du robot
- Partie V : choix d'itinéraire d'un robot

Sujet reproduit sans retouche à partir de la page suivante.

Résumé des parties, chapitres de cours liés et autres années : prepahelp.com/annales/e3a/mp/2024/si

Prepa Help ne revendique aucun droit sur le sujet, propriété de l'organisateur du concours.

ÉPREUVE MUTUALISÉE AVEC E3A-POLYTECH

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

SCIENCES INDUSTRIELLES

Durée : 4 heures

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

RAPPEL DES CONSIGNES

- *Utiliser uniquement un stylo noir ou bleu foncé non effaçable pour la rédaction de votre composition ; d'autres couleurs, excepté le vert, bleu clair ou turquoise, peuvent être utilisées, mais exclusivement pour les schémas et la mise en évidence des résultats.*
- *Ne pas utiliser de correcteur.*
- *Écrire le mot FIN à la fin de votre composition.*

Les calculatrices sont autorisées.

Le sujet est composé de cinq parties, toutes indépendantes.

Le sujet comporte :
- Énoncé : 20 pages ;
- Document Réponse (DR) : 8 pages.

Le Document Réponse doit être rendu dans son intégralité avec la copie (le QR Code doit être collé sur la première page de la copie).

Skypod, système automatisé de préparation de commande

Présentation

Le système Skypod est une solution d'aide à la préparation de commande dans des zones de stockage de grande capacité. Il est conçu en France (région Hauts-de-France) par la société Exotec. Sa flexibilité et son adaptabilité en font un leader de son domaine, ce qui a permis à Exotec de devenir en 2022 la première licorne industrielle française (« start-up » valorisée à plus d'un milliard de dollars US). De grands noms internationaux du commerce en ligne ou physique comptent parmi ses principaux clients.

L'une de ses spécificités est son robot manipulateur de bac qui peut évoluer dans les trois dimensions. Il peut ainsi se déplacer sur le sol pour circuler dans les allées et rejoindre les postes de livraison (**figure 1**). Mais il peut également évoluer verticalement pour atteindre les bacs dans lesquels les produits sont stockés (**figure 2**).



Figure 1 - Robot évoluant horizontalement



Figure 2 - Robot évoluant verticalement

Quatre composants permettent de mettre en œuvre cette solution (**figure 4**) :

- la flotte de robots qui transporte les bacs entre opérateurs et zones de stockage ;
- les racks, permettant de stocker les bacs, et dont le positionnement et la structure permettent les déplacements horizontaux et verticaux des robots ;
- les stations qui permettent aux opérateurs de déposer et récupérer les produits dans les bacs apportés par les robots ;
- le serveur permettant de gérer la flotte de robots en lien avec les consignes données par les stations.

Ce sujet s'intéresse particulièrement aux déplacements du robot. Un extrait du cahier des charges est donné sous forme de diagramme d'exigence par la **figure 25** de l'**annexe 2**.

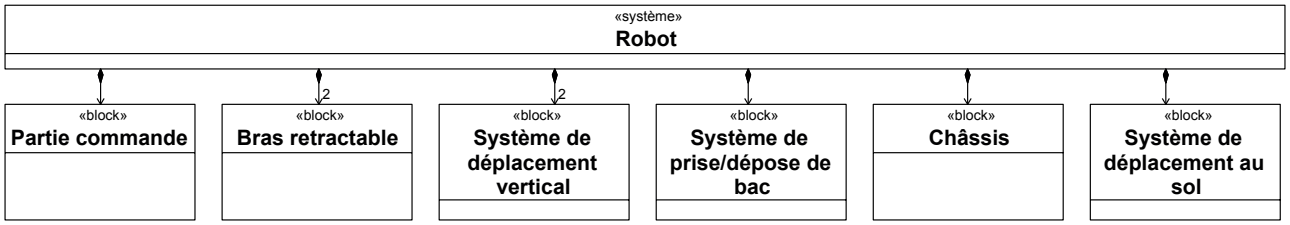


Figure 3 - Diagramme de définition de blocs d'un robot Skypod

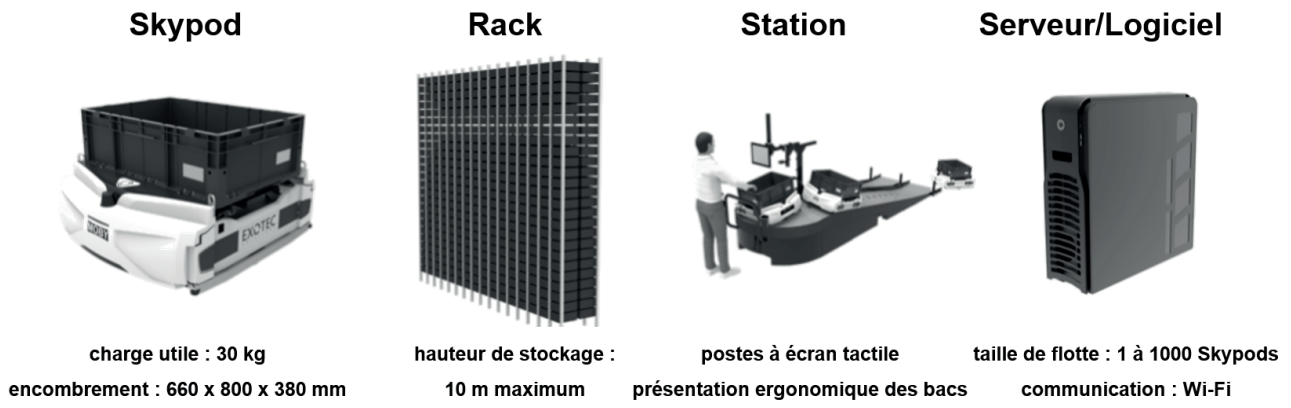


Figure 4 - Principaux composants de la solution Skypod

Analyse structurelle du robot

La structure du robot est présentée **figure 3**. Ce robot est constitué d'un *châssis* et de différents sous-systèmes.

- La *partie commande* contrôle le robot, communique avec le serveur et fournit des consignes aux autres sous-systèmes.
- Le *système de déplacement au sol* permet au robot de se déplacer au sein de l'entrepôt. Il est composé de deux roues motrices (droite et gauche) commandées par deux moteurs et de deux autres roues libres permettant de garantir une bonne stabilité au sol (**figure 1**).
- Deux *systèmes de déplacement vertical* (droit et gauche) permettent au robot de s'élever entre les racks afin d'atteindre une hauteur donnée. Ils entraînent quatre pignons situés dans chaque coin du robot qui eux-même engrènent sur des chaînes tendues verticalement le long des racks (**figure 2** et **figure 13**). La liaison pignon/chaîne se comporte comme une liaison pignon/crémaillère.
- Deux *systèmes de bras rétractables* facilitent le déplacement au sol, chaque système de déplacement vertical est positionné sur un bras rétractable. Ils sont rentrés au sein du châssis lors des déplacements au sol, puis sortis lors des déplacements verticaux afin que les pignons entrent en contact avec les chaînes tendues. Un robot ne peut donc évoluer horizontalement qu'au sol.
- Un *système de prise et dépose de bac* translate le bac de sa position de stockage dans le rack à sa position de transport sur le robot (ou inversement). Il est constitué d'une fourche télescopique qui se déploie (**figure 5** et **figure 6**) afin de se positionner sous le bac, puis rentre en déplaçant le bac avec elle.



Figure 5 - Prise d'un bac (fourche sous le bac dans un rack)

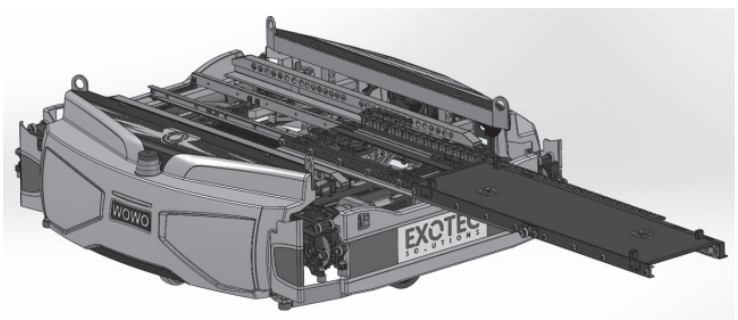


Figure 6 - Fourche déployée

Partie I - Déroulement d'une préparation de commande

***Objectif** : étudier le comportement séquentiel du robot lors d'une préparation de commande.*

Une préparation de commande se déroule en plusieurs étapes. Après réception de la demande, un traitement est effectué afin d'affecter les tâches à réaliser à un ou plusieurs robots (en fonction du nombre de produits demandés) et à un opérateur situé sur une station. Le ou les robots vont ensuite chercher les bacs contenant les produits nécessaires à la commande et les rapportent à la station. Enfin, l'opérateur effectue la mise en carton et expédie la commande.

Le déplacement d'un robot lui permettant d'aller prendre un bac s'effectue en plusieurs phases. Le robot se déplace d'abord au sol de sa position de départ jusqu'au pied de la rangée verticale du rack qui contient le bac à prendre. Une fois en position, ses bras rétractables sortent, ce qui permet aux pignons motorisés d'engrener sur les chaînes tendues verticalement, générant ainsi l'ascension du robot. Pour prendre un bac, le robot s'immobilise de façon à ce que la surface de la fourche soit située 5 cm en-dessous du fond du bac. La fourche télescopique est alors sortie. Puis le robot monte de 10 cm, la fourche est ensuite rentrée afin de translater le bac sur le robot. Le robot peut désormais descendre jusqu'au sol, puis rentrer ses bras rétractables et se déplacer de nouveau au sol jusqu'à la station.

Le comportement séquentiel du robot est décrit par le diagramme d'état donné par la **figure 1** du **document réponse** (**DR**). Le **tableau 1** fournit la description des ordres des moteurs et le **tableau 2** les informations reçues par la partie commande. Dans cette partie, les durées des phases d'accélération et de décélération sont négligées pour tous les mouvements.

Ordres moteurs	Rôle	Valeurs
Système de déplacement au sol :		
M_RoueG	Consigne de vitesse de la roue gauche	Valeurs continues
M_RoueD	Consigne de vitesse de la roue droite	entre $-V_{max}$ et V_{max}
Bras rétractables :		0 pas de déplacement
M_BrasG	Déplacement du bras gauche	$+1$ sortie des bras
M_BrasD	Déplacement du bras droit	-1 rentrée des bras
Système de déplacement vertical :		GV : grande vitesse
M_PignonG	Consignes de vitesse des pignons gauches	PV : petite vitesse
M_PignonD	Consignes de vitesse des pignons droits	$-$GV / $-$PV : descente
		$+$GV / $+$PV : montée
Fourche télescopique :		0 pas de déplacement
M_Fourche	Déplacement de la fourche	$+1$ sortie de la fourche
		-1 rentrée de la fourche

Tableau 1 - Variables de sortie de la partie commande

Informations	Rôle
Capteurs fin de course (détecteur) : BrasG_sorti et BrasG_rentré BrasD_sorti et BrasD_rentré Fourche_sortie et Fourche_rentrée	**Leur sortie vaut 1 si la position détectée est atteinte, 0 sinon** Capteurs du bras rétractable gauche Capteurs du bras rétractable droit Capteurs de la fourche
Position courante du robot : alt_courante pos_courante	 position verticale du robot position au sol du robot
Position à atteindre : bac (bac = 0 → prise de bac) pos_finale etag	 présence de bac sur le robot (mission de prise ou de dépose de bac) position au sol d'arrivée numéro d'étagère (0 = étagère du bas) où se situe le bac à prendre

Tableau 2 - Variables d'entrée de la partie commande

On s'intéresse au scénario d'un robot allant chercher un bac sur l'étagère 1 (etag = 1). On notera que la distance verticale entre deux étagères est de 40 cm. Pour ce scénario, le temps est discrétisé $t_0, t_1, ..., t_{k-1}, t_k, t_{k+1}, ...$ pour $k \in \mathbb{N}$ avec un pas de temps $t_{k+1} - t_k = \Delta t$ constant. On notera les constantes suivantes :

Temps de rentrée/sortie de la fourche	$2.\Delta t$
Temps de rentrée/sortie des bras rétractables	$2.\Delta t$
Valeurs de consigne pour le déplacement vertical	$GV = 5/\Delta t$ cm·s^{-1} et $PV = \frac{1}{2}GV$

Q1. Compléter le chronogramme sur le **DR1** en suivant le scénario suivant :

- à t_0, le robot reçoit une nouvelle destination (pos_finale et etag sont connus) ;
- à t_1, le calcul de l'itinéraire est fini et le robot commence à se déplacer vers sa destination pos_finale, sa trajectoire est en ligne droite (M_RoueG = M_RoueD) et la consigne est de V_{max} ;
- à t_3 le robot arrive à pos_finale et commence à sortir les bras ;
- le robot monte et récupère le bac, puis redescend au sol ;
- une fois au sol, le robot retourne à sa position de départ (consigne de $-V_{max}$).

Partie II - Trajectoire du robot en phase de déplacement au sol

***Objectif** : étudier le comportement cinématique du robot en phase de déplacement au sol.*

Pour ses déplacements au sol, le robot possède deux roues motrices **2** et **3** insérées dans deux chaînes de puissance différentes (et donc entraînées par deux moteurs différents). Pour assurer la stabilité au sol, deux autres roues laissées libres de leurs mouvements (dites « folles ») sont associées à un mécanisme permettant aux quatre roues d'être en contact avec le sol en permanence.

On s'intéresse à une trajectoire composée d'un segment droit, d'une courbe permettant au robot de tourner d'un quart de tour vers la gauche, puis d'un dernier segment droit (**figure 7**).

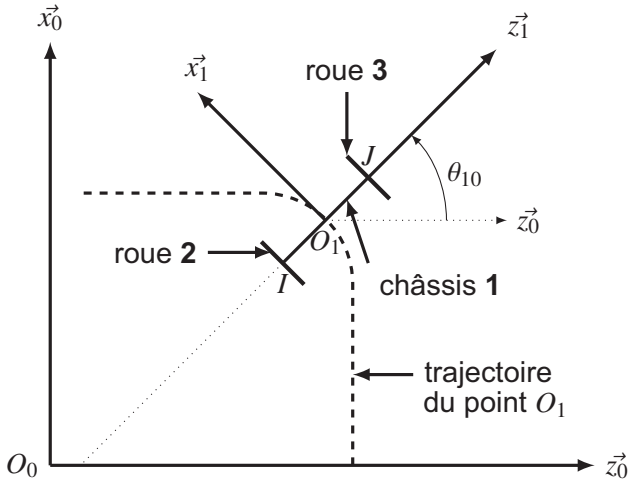


Figure 7 - Paramétrage du robot au sol

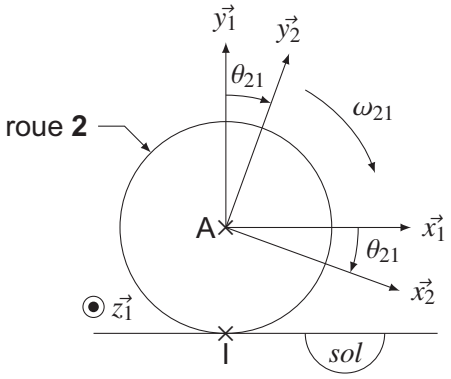


Figure 8 - Paramétrage de la roue **2** en contact avec le sol en I

Le robot est vu comme un ensemble de trois solides : le châssis **1**, la roue gauche **2** en contact avec le sol **0** au point I et la roue droite **3** en contact avec le sol **0** au point J.

Les roues **2** et **3** sont en liaison pivot d'axe respectivement $(A, \vec{z_1})$ et $(B, \vec{z_1})$ avec le châssis **1** et on considérera qu'il y a roulement sans glissement de la roue **2** (respectivement roue **3**) par rapport au sol **0** en I (respectivement en J). Pour des raisons de simplification, les roues « folles » ne sont pas étudiées ou représentées mais imposent que le châssis **1** ne peut être en rotation que selon $\vec{y_1}$.

On définit :

- $\mathcal{R}_0(O_0, \vec{x_0}, \vec{y_0}, \vec{z_0})$ le référentiel supposé galiléen lié au sol **0**. On notera que $\vec{y_0}$ est donc la verticale ascendante ;
- $\mathcal{R}_1(O_1, \vec{x_1}, \vec{y_1}, \vec{z_1})$ le référentiel lié au châssis **1**, avec $\theta_{10} = (\vec{z_0}, \vec{z_1}) = (\vec{x_0}, \vec{x_1})$, $\omega_{10}(t) = \dfrac{d\theta_{10}(t)}{dt}$ et la vitesse du robot $\vec{V}(O_1 \in 1/0) = V(t)\vec{x_1}$;
- $\mathcal{R}_2(A, \vec{x_2}, \vec{y_2}, \vec{z_2})$ le référentiel lié à la roue gauche **2**, avec A le centre de la roue, $R = 20$ mm son rayon, $\theta_{21}(t) = (\vec{x_1}, \vec{x_2}) = (\vec{y_1}, \vec{y_2})$ et $\omega_{21}(t) = \dfrac{d\theta_{21}(t)}{dt}$ (**figure 8**) ;
- $\mathcal{R}_3(B, \vec{x_3}, \vec{y_3}, \vec{z_3})$ le référentiel lié à la roue droite **3**, avec B le centre de la roue, $R = 20$ mm son rayon, $\theta_{31}(t) = (\vec{x_1}, \vec{x_3}) = (\vec{y_1}, \vec{y_3})$ et $\omega_{31}(t) = \dfrac{d\theta_{31}(t)}{dt}$ (paramétrage analogue à la roue **2**) ;
- L la largeur du robot : $\overrightarrow{AO_1} = \overrightarrow{O_1B} = \dfrac{L}{2}\vec{z_1}$.

Q2. À l'aide de la condition de roulement sans glissement en I, donner la relation entre $V(t)$, $\omega_{21}(t)$ et $\omega_{10}(t)$.

Q3. Par un raisonnement analogue, donner directement la relation entre $V(t)$, $\omega_{31}(t)$ et $\omega_{10}(t)$.

Q4. En déduire $\omega_{10}(t)$ en fonction de $\omega_{21}(t)$ et de $\omega_{31}(t)$.

On s'intéresse dans un premier temps à une trajectoire rectiligne. On note $\omega_{moy}(t)$ la vitesse de rotation des roues pour cette trajectoire pour une vitesse $V(t) = 2$ m·s^{-1}.

Q5. Démontrer que pour une trajectoire rectiligne, $\omega_{21}(t) = \omega_{31}(t)$. En déduire $\omega_{moy}(t)$ et donner sa valeur.

On s'intéresse dans un deuxième temps à la réalisation du virage de 90° vers la gauche (**figure 7**). Pour cela, on fait varier la vitesse ω_{21} par une loi en trapèze et la vitesse ω_{31} par une seconde loi en

trapèze telle que $\dot{\omega}_{21}(t) = -\dot{\omega}_{31}(t)$, avec $\dot{\omega}_{21}$ et $\dot{\omega}_{31}$ les accélérations angulaires des roues **2** et **3** par rapport au châssis **1**.

Q6. Donner dans ces conditions la valeur de l'accélération angulaire $\dot{\omega}_{10}(t)$ en fonction de $\dot{\omega}_{21}(t)$.

Ces lois de commande de vitesse des roues aboutissent à l'évolution de $\omega_{10}(t)$ au cours d'un virage de 90° donnée **figure 9**. La norme de l'accélération angulaire pendant les phases 1 et 3 est constante et notée γ_{10}.

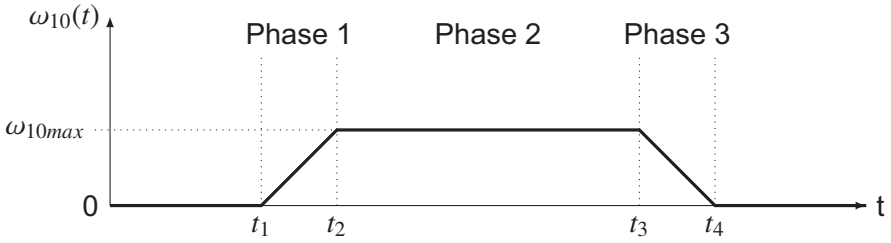


Figure 9 - Évolution de ω_{10} lors d'un virage de 90°

Q7. En faisant l'hypothèse que $t_1 = 0$, donner les expressions littérales de t_2, de t_3 et de t_4 en fonction de ω_{10max} et de γ_{10}.

On propose plusieurs simulations de trajectoire pour différentes valeurs de ω_{10max} dont le point de départ est de coordonnées (0,0) (**figure 10**). À des fins de comparaison, la **figure 11** présente ces trajectoires en version normalisée (point de départ et d'arrivée de coordonnées (0,0) et (-1,1)).

Q8. À l'aide des **figure 10** et **figure 11**, commenter l'influence de ω_{10max} sur la trajectoire du robot lors d'un virage.

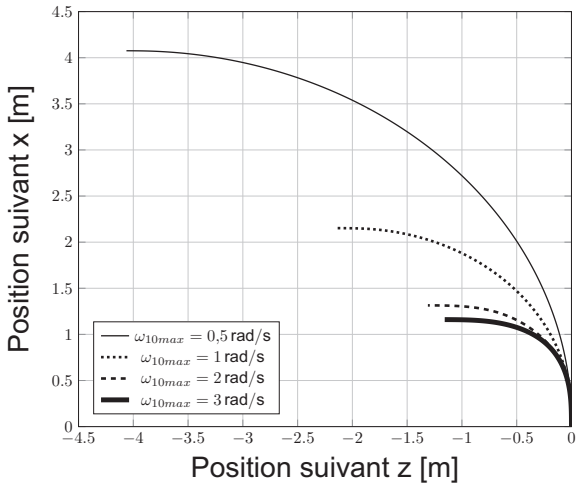


Figure 10 - Trajectoires selon plusieurs valeurs de ω_{10max}

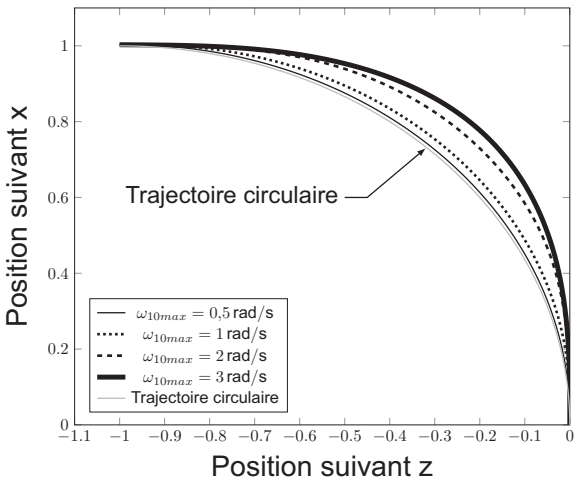


Figure 11 - Trajectoires normalisées selon plusieurs valeurs de ω_{10max}

Partie III - Sollicitation du robot en mouvement vertical

***Objectif** : déterminer la puissance des moteurs nécessaire pour assurer le mouvement vertical du robot.*

Une fois le robot positionné aux pieds des racks, les bras rétractables sont déployés afin de mettre en contact les pignons sur les chaînes tendues le long des poteaux constituant les racks. La rotation des pignons entraînera alors l'ascension du robot, **figure 12** et **figure 13**.

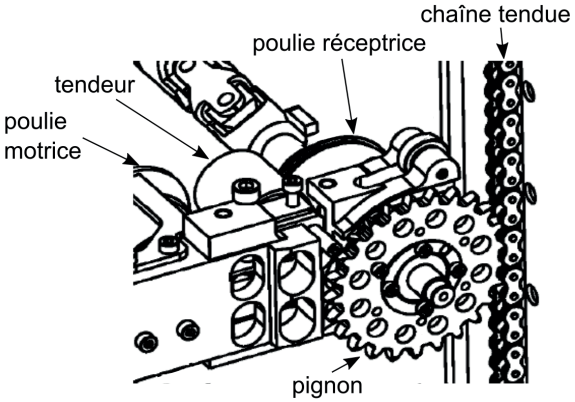


Figure 12 - Détail de l'engrènement d'un pignon sur la chaîne



Figure 13 - Vue du robot en montée (sans charge ni bac), avec pignons et chaînes redessinés

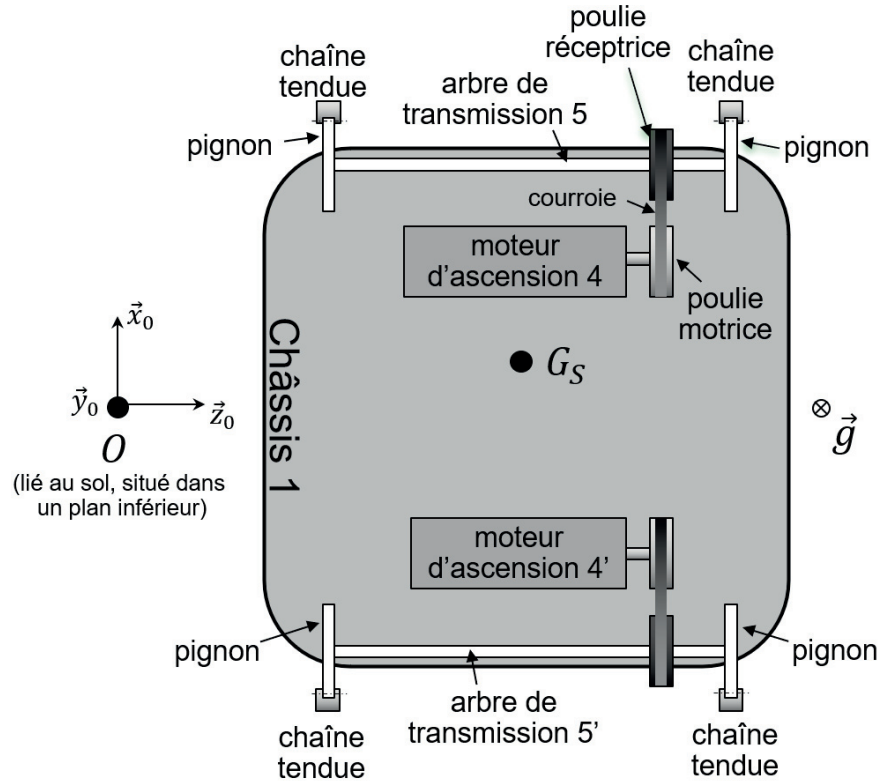


Figure 14 - Schématisation de la chaîne de puissance associée à l'ascension du robot. Vue de dessus du robot, pesanteur selon $-\vec{y}_0$.

La structure de la chaîne de puissance permettant l'ascension du robot est schématisée **figure 14**. Elle est composée de deux ensembles similaires, chacun étant constitué d'un moteur d'ascension et

d'un système poulie/courroie dont la poulie motrice est liée à l'arbre de sortie du moteur et la poulie réceptrice à un arbre de transmission. Ce dernier supporte également les deux pignons qui engrènent sur les chaînes tendues. La rotation des pignons va donc permettre de générer le mouvement vertical du robot. Un système annexe, non étudié ni représenté ici, permet de s'assurer que les pignons soient toujours en prise dans les chaînes.

Hypothèses et paramétrage

- On note $\mathcal{R}_0(O_0, \vec{x_0}, \vec{y_0}, \vec{z_0})$ le référentiel supposé galiléen lié au sol **0**.
- On note S l'ensemble des pièces en mouvement :

  S = {Châssis **1**+Moteurs+Systèmes Poulie/Courroie+Arbres transmission+Pignons+Bac rempli}, de masse m_S et de centre d'inertie G_S, tel que :

  $$\overrightarrow{OG_S} = x_{G_S}\,\vec{x_0} + y_{G_S}(t)\,\vec{y_0} + z_{G_S}\,\vec{z_0}\ .$$

  La position du centre de gravité n'est pas le centre géométrique du robot car la masse transportée est potentiellement excentrée.
- Le mouvement du châssis **1** par rapport au sol **0** est une translation de vitesse $v_a(t)\vec{y_0} = \dot{y}_{G_S}(t)\vec{y_0}$. Ainsi $v_a(t)$ est la vitesse d'ascension du robot.
- On définit les vitesses de rotation du moteur **4** et de l'arbre de transmission **5** par rapport au châssis **1** par respectivement $\overrightarrow{\Omega}_{m/1} = \omega_{m1}(t)\vec{z_0}$ et $\overrightarrow{\Omega}_{5/1} = \omega_{51}(t)\vec{z_0}$. Les vitesses de rotation du moteur **4'** ($\overrightarrow{\Omega}_{m'/1} = \omega_{m'1}(t)\vec{z_0}$) et de l'arbre de transmission **5'** ($\overrightarrow{\Omega}_{5'/1} = \omega_{5'1}(t)\vec{z_0}$) sont opposées respectivement à celles de **4** et de **5**.
- L'action mécanique du moteur **4** est modélisée par un torseur couple dont le moment sur son axe de rotation vaut $C_m(t)$. On suppose ici que les deux moteurs délivrent le même couple, mais en sens opposés.
- Les différents frottements (internes et externes) sont ramenés sur l'axe de rotation de chaque arbre de transmission et sont modélisés par un moment constant $C_f\vec{z_0}$ sur l'arbre **5**, $C'_f\vec{z_0}$ sur l'arbre **5'**. Ces deux couples sont opposés aux vitesses de rotation des arbres **5** et **5'**, de sorte que la valeur numérique de C_f est de signe opposé à celui de $\omega_{51}(t)$ et que la valeur numérique de C'_f est de signe opposé à celui de $\omega_{5'1}(t)$.
- On note respectivement R_m, R_r et R_p les rayons d'enroulement des poulies motrices et réceptrices et le rayon primitif des pignons.
- On note respectivement J_m et J_5 les moments d'inertie selon leurs axes de rotation d'un arbre moteur (poulie comprise) et d'un arbre de transmission (poulie et roues dentées comprises).
- L'attraction de pesanteur est portée par $-\vec{y}_0$ de sorte que $\vec{g} = -g\vec{y_0}$.
- L'inertie de la courroie est négligée.
- Le châssis inclut la batterie, les cartes électroniques et la connectique.

Q9. Déterminer l'expression de l'énergie cinétique $E_c(S/\mathcal{R}_0)$ de l'ensemble S par rapport au repère $\mathcal{R}_0$.

La liaison entre un pignon et une chaîne correspond à un roulement sans glissement sur le rayon primitif R_p du pignon.

Q10. Exprimer le vecteur de la vitesse d'ascension du robot $v_a(t)\,\vec{y_0}$ en fonction de la vitesse de rotation de l'arbre moteur $\omega_{m1}(t)$.

Q11. En déduire l'expression de l'inertie équivalente notée J_{eq} de l'ensemble S rapportée à l'arbre moteur, en fonction de m_s, J_m, J_5 et des grandeurs géométriques.

Q12. Déterminer l'expression de la somme des puissances extérieures (galiléennes) et intérieures à l'ensemble S. On ne fera apparaître que $\omega_{m1}(t)$ comme variable cinématique.

Q13. Déduire des questions précédentes, en justifiant rigoureusement, l'expression du couple moteur $C_m(t)$ en fonction de $\dot{\omega}_{m1}(t)$ et des grandeurs caractéristiques constantes du système.

On donne **figure 15** le profil de vitesse choisi pour l'évolution de la vitesse verticale du robot pour un cycle de montée/descente et dans le **tableau 3** les valeurs numériques associées. On souhaite déterminer l'évolution correspondante du couple moteur.

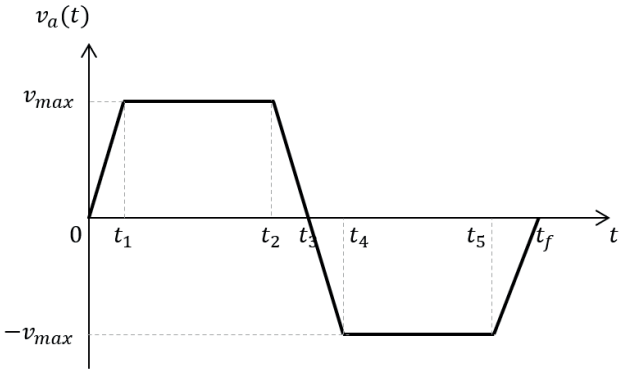


Figure 15 - Pilotage adopté pour la vitesse d'ascension pour un cycle de montée/descente

Données	
$R_m = R_r$ = 30 mm	t_1 = 2 s
$\|C_f\|=\|C'_f\|$ = 1 N·m	t_2 = 8 s
J_{eq} = 1,14.10^{-1} kg·m^2	t_3 = 10 s
R_p = 38 mm	t_4 = 12 s
v_{max}= 2 m·s^{-1}	t_5 = 18 s
m_s = 78 kg	t_f = 20 s

Tableau 3 - Valeurs numériques utiles

Q14. Compléter le **DR2** en traçant l'évolution de l'accélération angulaire $\dot{\omega}_{m1}(t)$ au cours du temps.

Q15. Tracer sur le **DR3** l'évolution de C_f au cours du temps.

Il est alors possible, grâce aux travaux précédents, de calculer l'évolution du couple moteur $C_m(t)$ pour les différentes phases (**tableau 4**).

intervalle	de 0 à t_1	de t_1 à t_2	de t_2 à t_3	de t_3 à t_4	de t_4 à t_5	de t_5 à t_f
$C_m(t)$	-17 N·m	-15,5 N·m	-14 N·m	-12 N·m	-13,5 N·m	-15 N·m

Tableau 4 - Évolution de $C_m(t)$ au cours du temps.

D'après les exigences, le moteur envisagé pour la motorisation est capable de fournir une puissance mécanique de 1 000 W.

Q16. Après avoir précisé l'instant correspondant, déterminer la valeur numérique de la puissance maximale que le moteur devra développer pour assurer le mouvement étudié. Conclure quant au choix du moteur envisagé.

Partie IV - Asservissement de l'assiette du robot

***Objectif** : modéliser et mettre au point l'asservissement de l'inclinaison du robot.*

Cette partie s'intéresse au contrôle de l'inclinaison du robot lors de la phase d'ascension, repérée par son angle d'inclinaison par rapport à l'horizontale, noté $\alpha(t)$. Cette phase est schématisée **figure 16**, dans le cas horizontal ($\alpha = 0$) et dans le cas incliné ($\alpha \neq 0$). Les pignons sont ramenés dans le plan médian. Pour rappel, un sous-système non étudié ici permet de garantir le contact pignon/chaîne.

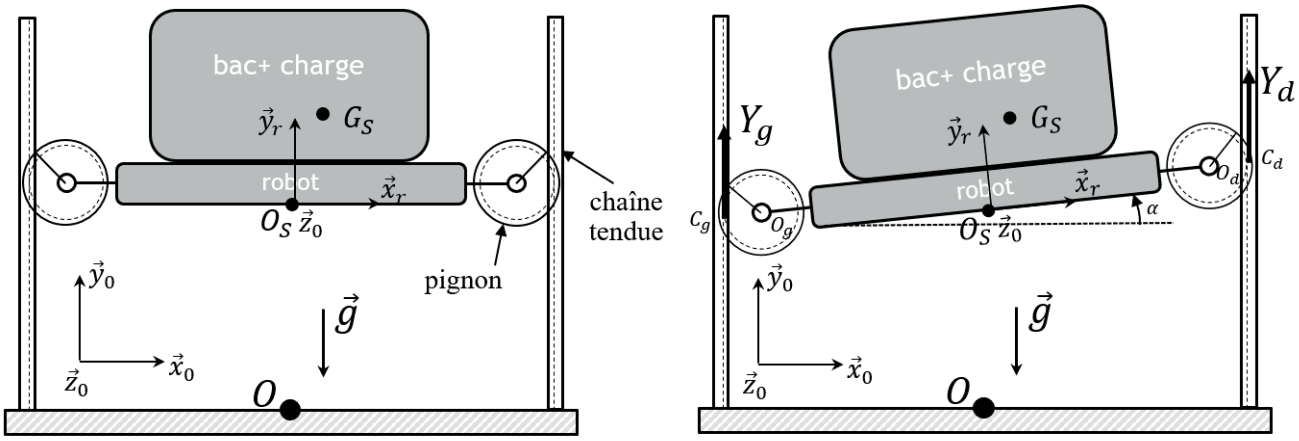


Figure 16 - Schématisation du robot en phase d'ascension, dans la configuration horizontale (à gauche) et inclinée (à droite).

On présente **figure 17** la structure générale de l'asservissement associée à la phase étudiée. Cette inclinaison est modifiée grâce au différentiel de vitesse et de position des pignons gauche et droit. Un inclinomètre mesurant l'inclinaison du robot permet au calculateur de gérer l'asservissement.

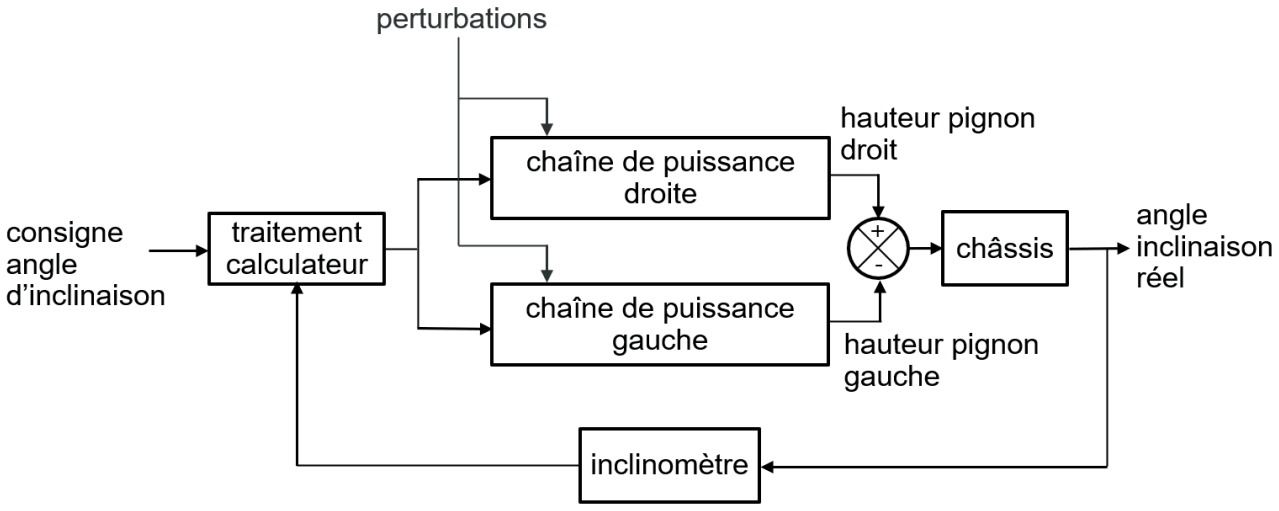


Figure 17 - Structure générale de l'asservissement en inclinaison du robot en phase d'ascension.

IV.1 - Modélisation de la motorisation

Un moteur d'ascension peut se modéliser comme une machine à courant continu dont on rappelle les équations :

$$u_m(t) = e(t) + R_m i_m(t) + L_m \frac{di_m(t)}{dt} \; ; \; e(t) = K_e \omega_{m1}(t) \; ; \; C_m(t) = K_c i_m(t) \; ; \; J_{eq} \frac{d\omega_{m1}(t)}{dt} = C_m(t) - C_r - C_f(t)$$

et dont les notations sont détaillées **tableau 5**. Le schéma-blocs correspondant est ébauché dans le **DR4**. Les notations des variables dans le domaine de Laplace sont répertoriées dans le **tableau 6**.

Variables	**Constantes**
$u_m(t)$: tension moteur (V) $i_m(t)$: intensité moteur (A) $e(t)$: force contre-électromotrice (V) $C_m(t)$: couple électromagnétique (N·m) C_r : couple résistant dû à la pesanteur ramené sur l'arbre moteur (N·m) $C_f(t)$: couple résistant dû à l'ensemble des frottements secs ramené sur l'arbre moteur (N·m)	$R_m = 3\ \Omega$: résistance de l'induit $L_m = 1\ mH$: inductance de l'induit $K_c = 2$ N·m·A^{-1} : constante de couple $K_e = 2$ V·s·rad^{-1} : constante électrique $J_{eq} = 1,14.10^{-1}$ kg·m^2 : inertie équivalente des pièces mobiles entraînées par le moteur, ramenée sur l'arbre moteur

Tableau 5 - Variables et valeurs numériques utiles

Notation temporelle	Notation Laplace	Notation temporelle	Notation Laplace
$e(t)$	$E(p)$	$u_m(t)$	$U_m(p)$
$i_m(t)$	$I_m(p)$	$\omega_{m1}(t)$	$\Omega_{m1}(p)$
$C_m(t)$	$C_m(p)$	$C_f(t)$	$C_f(p)$
échelon unitaire	$\frac{1}{p}$	C_r	$\frac{C_r}{p}$

Tableau 6 - Notations dans le domaine de Laplace

Q17. Compléter les blocs du **DR4** au niveau des « ... », avec les fonctions de transfert et variables manquantes dans le schéma-blocs du moteur.

Afin de prédire au mieux le comportement du robot, il est nécessaire de modéliser avec plus de précision le couple résistant C_r que subit chaque moteur. On ne s'intéresse d'abord qu'au moteur droit. La démarche est identique pour le moteur gauche.

Hypothèses et paramétrage

- On note $\mathcal{R}_0(O_0, \vec{x_0}, \vec{y_0}, \vec{z_0})$ le référentiel supposé galiléen lié au sol.
- On note S l'ensemble des pièces en mouvement :

  S = {Châssis 1+Moteurs+Systèmes Poulie/Courroie+Arbres transmission+Pignons+Bac rempli}, de masse m_S.
- L'attraction de la pesanteur est notée $-g\vec{y_0}$.
- On note α l'angle d'inclinaison du robot et $\mathcal{R}_0(O_S, \vec{x_r}, \vec{y_r}, \vec{z_0})$ le référentiel lié au robot, tel que $\alpha = (\vec{x_0}, \vec{x_r}) = (\vec{y_0}, \vec{y_r})$.
- On supposera que l'angle α reste très petit.
- On note G_S le centre d'inertie de l'ensemble S, tel que $\overrightarrow{OG_S} = \overrightarrow{OO_S} + \overrightarrow{O_S G_S}$ avec $\overrightarrow{OO_S} = h\vec{y_0}$ et $\overrightarrow{O_S G_S} = x_{G_S}\vec{x_r} + y_{G_S}\vec{y_r}$. La position du centre de gravité n'est pas le centre géométrique du robot car la masse transportée est potentiellement excentrée.

- On note O_g et O_d les centres des pignons gauche et droit ramenés dans le plan d'étude. On note C_g et C_d les points d'engrènement entre les pignons et les chaînes, de sorte que : $\overrightarrow{O_gO_S} = \frac{L}{2}\vec{x_r} - a\vec{y_r}$, $\overrightarrow{O_SO_d} = \frac{L}{2}\vec{x_r} + a\vec{y_r}$, $\overrightarrow{O_gO_d} = L\vec{x_r}$, $\overrightarrow{C_gO_g} = \overrightarrow{O_dC_d} = R_p\vec{x_r}$.
- Les actions mécaniques aux points C_g et C_d seront modélisées par les glisseurs simplifiés suivants :

$$\left\{\mathcal{T}^{C_g}{}_{\text{chaîne}\to\text{pignon gauche}}\right\} = \left\{\begin{array}{c} Y_g\vec{y_0} \\ \vec{0} \end{array}\right\}_{C_g} \text{ et } \left\{\mathcal{T}^{C_d}{}_{\text{chaîne}\to\text{pignon droit}}\right\} = \left\{\begin{array}{c} Y_d\vec{y_0} \\ \vec{0} \end{array}\right\}_{C_d}.$$

- Le couple résistant C_r dû à la pesanteur sera distingué pour chaque moteur. On note C_r^g le couple résistant appliqué à l'arbre moteur gauche et C_r^d pour le moteur droit.

Il est rappelé que compte tenu du fait que les rayons primitifs des poulies sont identiques et que la transmission est considérée comme parfaite, tout se passe comme si chaque moteur était directement en prise sur l'arbre du pignon correspondant.

Q18. En se plaçant à l'équilibre statique et en précisant la démarche, déterminer les expressions de Y_g et Y_d en fonction de m_S, g, L, R_p et de x_{G_S}. On rappelle que l'angle α est supposé très petit.

Q19. En déduire l'expression des couples C_r^g et C_r^d.

La fonction de transfert du moteur sans perturbation s'écrit :

$$H_m(p) = \frac{\Omega_{m1}(p)}{U_m(p)} = \frac{1/K_e}{1 + \frac{RJ_{eq}}{K_eK_c}p + \frac{LJ_{eq}}{K_eK_c}p^2}.$$

Il est alors possible de modifier le schéma-blocs initial du moteur (**DR4**), avec perturbation, pour le mettre sous la forme proposée **figure 18**, dans laquelle $P(p)$ dépend du signal dû à la perturbation $C_f(p) + \frac{C_r}{p}$.

Q20. Exprimer $H_p(p)$ en fonction des caractéristiques du moteur. Préciser l'unité de $P(p)$.

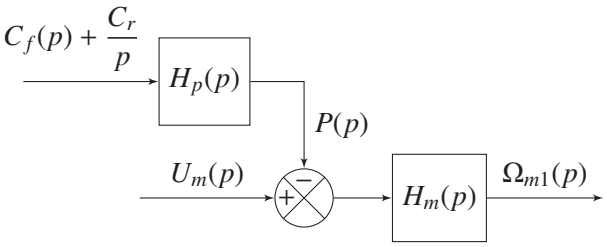


Figure 18 - Schéma-blocs modifié du moteur

Dans la suite, le signal $P(p)$ sera distingué en un signal $P_d(p)$ pour le moteur droit et $P_g(p)$ pour le moteur gauche.

IV.2 - Mise en place et correction de l'asservissement

Les activités précédentes permettent de mettre en place la modélisation complète de l'asservissement en inclinaison du robot dans la phase d'ascension (**figure 19**) en considérant que l'angle d'inclinaison reste petit. Les différentes variables dans le domaine de Laplace introduites sont détaillées dans le **tableau 7**.

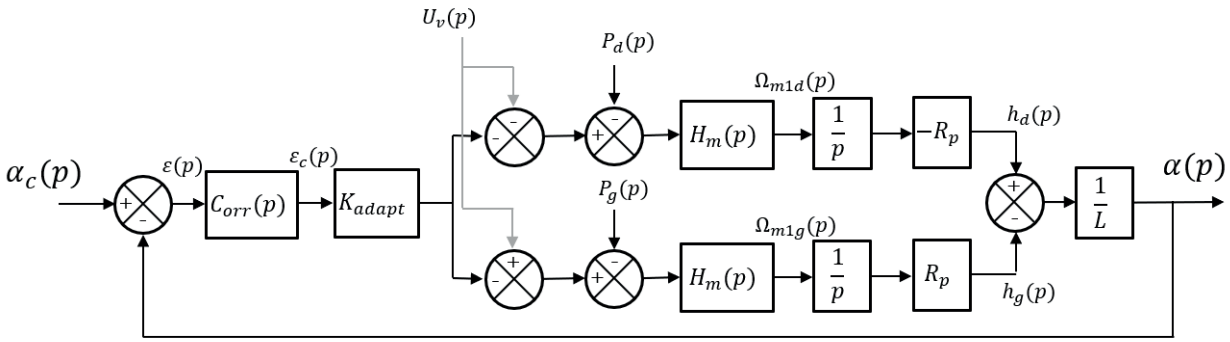


Figure 19 - Schéma-blocs de l'asservissement en inclinaison, en phase de mouvement vertical

Variables	
$\alpha_c(p)$: consigne angulaire d'inclinaison	$\alpha(p)$: angle d'inclinaison du robot
$\varepsilon(p)$: écart	$\Omega_{m1d}(p)$: vitesse de rotation moteur droit
$\varepsilon_c(p)$: écart corrigé	$\Omega_{m1g}(p)$: vitesse de rotation moteur gauche
$P_g(p)$: perturbation moteur gauche	$P_d(p)$: perturbation moteur droit
$h_g(p)$: altitude point O_g	$h_d(p)$: altitude point O_d
$U_v(p)$: tension de consigne correspondant à la vitesse d'ascension souhaitée	K_{adapt} : gain adaptateur
Fonctions de transfert	
$H_m(p)$: fonction de transfert du moteur	$C_{orr}(p)$: correcteur
R_p : rayon primitif du pignon	$L = \|\|\overrightarrow{O_gO_d}\|\|$

Tableau 7 - Définition des variables et fonctions de transfert (dans le domaine de Laplace)

Q21. Justifier les signes du comparateur ayant pour entrées $h_d(p)$ et $h_g(p)$.

Q22. Exprimer $h_d(p)$ et $h_g(p)$ en fonction des variables $\varepsilon_c(p)$, $U_v(p)$ et $P_d(p)$ ou $P_g(p)$.

On suppose maintenant que la charge contenue dans le bac est centrée sur le robot, c'est-à-dire $P_d(p) = -P_g(p)$.

Q23. Montrer qu'il est alors possible de mettre le schéma-blocs initial sous la forme présentée par la **figure 20**. Pour cela, exprimer $H_{eq}(p)$ en fonction du contenu des blocs du schéma initial de la **figure 19**.

Q24. Donner la fonction de transfert en boucle ouverte (notée $H_{BO}(p)$) de ce système en fonction de $H_m(p)$, $C_{orr}(p)$, K_{adapt} et des paramètres géométriques. Préciser son ordre et sa classe si $C_{orr}(p) = 1$.

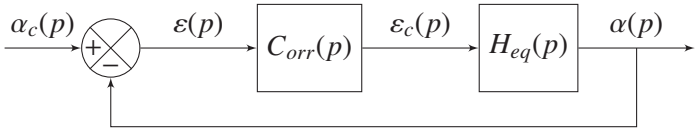


Figure 20 - Schéma-blocs simplifié de l'asservissement en inclinaison

On donne sur le **DR5** la réponse fréquentielle (en boucle ouverte) et la réponse indicielle (en boucle fermée) du système non corrigé (c'est-à-dire pour $C_{orr}(p) = 1$). Attention les courbes sont sur les pages **5/8** et **6/8** du **document réponse**.

Q25. Répondre sur le **DR5** : vérifier si les exigences associées à l'asservissement en inclinaison du robot sont vérifiées. Mettre en place les tracés permettant la vérification des critères considérés.

Quels que soient les résultats précédents, la fonction de transfert $H_{eq}(p)$ sera désormais prise de la forme :

$$H_{eq}(p) = \frac{K_{eq}}{p(1+\tau_1 p)(1+\tau_2 p)} \text{ avec } \tau_2 > \tau_1 \ .$$

Q26. Sur le **DR5**, tracer les diagrammes de Bode asymptotiques correspondant aux tracés proposés.

Q27. En déduire les valeurs numériques de τ_1 et de τ_2.

Afin d'améliorer les performances de l'asservissement, on se propose d'introduire un correcteur de type proportionnel intégral : $C_{orr}(p) = K_p \dfrac{1+T_i p}{T_i p}$.

Q28. Justifier le choix d'un tel correcteur.

Q29. Donner la valeur de T_i permettant de compenser le pôle dominant de $H_{eq}(p)$ et en déduire la forme de la fonction de transfert en boucle ouverte $H_{BO}(p)$ sous forme canonique.

Les diagrammes de Bode obtenus avec la valeur de T_i déterminée **Q29** sont tracés **figure 21**, en prenant $K_p = 1$.

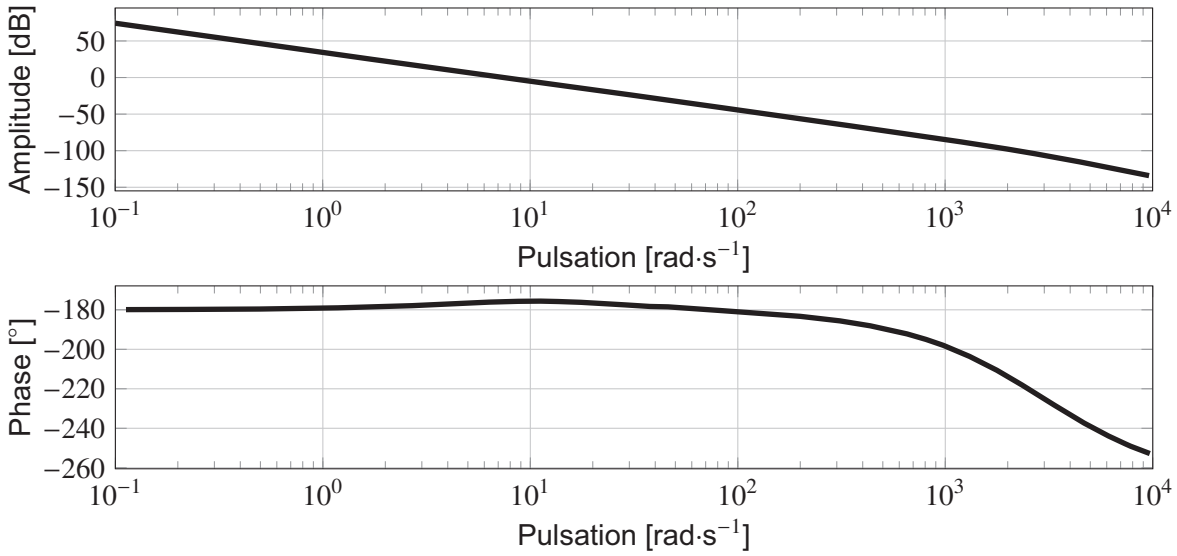


Figure 21 - Diagrammes de Bode avec correcteur PI ($K_p = 1$ et T_i choisi **Q29**)

On décide maintenant d'associer au correcteur précédent un correcteur à avance de phase (voir **annexe 1**, **page 19**) de paramètres a et T_{av}, de sorte que :

$$C_{orr}(p) = K_p \frac{1+T_i p}{T_i p} \cdot \frac{1+aT_{av}p}{1+T_{av}p} \text{ avec } a > 1.$$

Q30. Justifier le choix d'un tel correcteur.

Q31. Déterminer la valeur de a permettant d'apporter la phase nécessaire au niveau de la pulsation ω_{0dB} visée (20 rad·s^{-1}) permettant de respecter la marge de phase.

Q32. En déduire la valeur de T_{av} qu'il faut choisir.

Le dernier réglage à effectuer est celui de K_p. Pour cela, on exploite une simulation numérique paramétrique, qui donne les résultats **figure 22**.

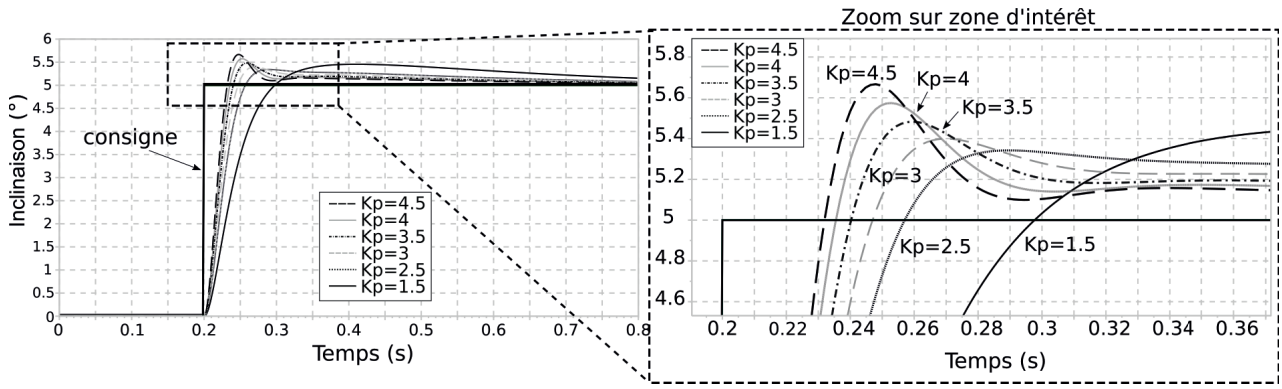


Figure 22 - Réponse indicielle pour différentes valeurs de K_p

Q33. À l'aide de la **figure 22**, donner une valeur approchée de K_p permettant de respecter les exigences observables. Préciser ces exigences.

Les travaux précédents permettent de régler au mieux le correcteur permettant l'asservissement en inclinaison du robot.

Partie V - Choix d'itinéraire d'un robot (Informatique Commune)

Les signatures des fonctions à produire, en langage Python, seront fournies pour préciser les types des arguments d'entrée et du résultat. Ainsi,

```
def F_exemple(n:int, L:[float]) -> {(int,str):bool}:
```

signifie que la fonction $\mathrm{F_exemple}$ prend deux arguments n et L avec n un entier et L une liste de nombres à virgule flottante et renvoie un dictionnaire dont les clés sont des 2-uplets constitués d'un entier, puis d'une chaîne de caractères et les valeurs associées sont de type booléen.

Il sera **inutile** pour les candidats de réécrire toutes ces informations sur la copie : la syntaxe « $\mathrm{def\ F_exemple(n,L):}$ » est suffisante. De même, les spécifications données dans les questions ne sont pas à réécrire.

Afin d'éviter de longues répétitions inutiles de code et pour permettre aux candidats n'ayant pas traité entièrement une question précédente, les candidats peuvent et doivent réutiliser les fonctions précédemment définies dans le sujet quand cela est possible.

Les candidats porteront une attention particulière au nom des variables et fonctions qu'ils pourraient être amenés à définir. Les programmes devront être commentés de façon pertinente afin d'aider le lecteur à comprendre la logique algorithmique proposée.

Enfin, quelques syntaxes de manipulation de liste et dictionnaire en Python sont données dans le tableau situé à la première page du **DR** (**page 1/8**).

***Objectif** : définir l'itinéraire le plus rapide d'un robot au sein d'un entrepôt.*

On s'intéresse dans cette partie au choix de l'itinéraire d'un robot entre une position de départ et une position d'arrivée en prenant en compte les positions des autres robots qui peuvent bloquer certains chemins. Ces blocages sont évolutifs car les autres robots évoluent eux-mêmes et bloquent, puis libèrent les chemins au cours du temps.

Afin de simplifier le problème, on représente l'entrepôt où évoluent les robots comme un ensemble de positions finies, chacune de ces positions pouvant être directement atteinte depuis un sous-ensemble de positions appelées positions voisines. Ainsi, on modélise l'entrepôt par un graphe non orienté dont les sommets représentent les positions et les arêtes représentent les déplacements possibles d'une position à ses voisines.

Un exemple d'entrepôt est représenté par le graphe de la **figure 23**. On fera les hypothèses suivantes :

- toutes les arêtes représentent des déplacements de même longueur : ainsi le graphe n'est pas pondéré, les arêtes sont toutes de longueur égale à 1 ;
- les sommets, appelés positions, sont placés sur un quadrillage : ainsi une position est définie et nommée selon sa ligne et sa colonne par un couple (ligne,colonne). Par exemple, la position en bas à gauche est la position (0,0) et celle en haut à gauche la position (3,0).

On implémente le graphe sous la forme d'une liste d'adjacence, à l'aide d'un dictionnaire associant chaque position sous la forme d'un *2-uplets* (i,j) (clé du dictionnaire) à l'ensemble de ses positions voisines (valeur associée à la clé). Ces positions voisines sont sous la forme de *2-uplets* (i,j) et contenues dans un tableau de type *list*. Un exemple est présenté dans le **tableau 8**. Cette structure de type *{(int,int) :[(int,int)]}* sera nommée *graph* dans les signatures des fonctions utilisées ci-après.

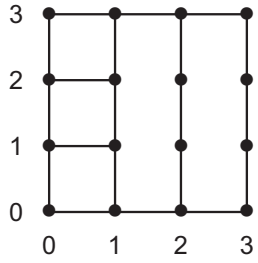


Figure 23 - Représentation sous forme de graphe d'un entrepôt exemple

```
{(0,0) :[(0,1),(1,0)],
 (0,1) :[(0,0),(0,2),(1,1)],
  ...
 (3,2) :[(2,2),(3,1),(3,3)]
 (3,3) :[(2,3),(3,2)]}
```

Tableau 8 - Liste d'adjacence d'un graphe

Par simplification, on considère le temps comme discret $t_0, t_1, ..., t_{k-1}, t_k, t_{k+1}, ...$ pour $k \in \mathbb{N}$ avec un pas de temps $t_{k+1} - t_k = 1$ constant, pas de temps juste nécessaire pour qu'un robot se déplace d'une position à une position voisine. L'instant initial t_0 représente l'instant où le robot dont on cherche le chemin est à sa position de départ.

L'une des difficultés pour définir un chemin dans ce contexte est de prendre en compte les autres robots de la flotte. Ainsi, certaines positions sont indisponibles sur une plage temporelle donnée car un autre robot s'y trouve, puis ces positions sont libérées car les robots se déplacent (rendant indisponibles d'autres positions).

On considère que l'on dispose d'une fonction $\mathrm{position_skypod}$, prenant en argument un instant t_k. La commande $\mathrm{pos_indispo{=}position_skypod(t)}$ permet d'affecter à la variable $\mathrm{pos_indispo}$ un dictionnaire dont les clés sont des *2-uplets* représentant les positions non atteignables à l'instant t, et les valeurs associées sont des chaînes de caractères représentant le nom du robot occupant cette position. Par exemple, $\mathrm{pos_indispo}$ est de la forme : $\{(1,0)$: 'R1', $(3,2)$: 'R2', $(1, 2)$: 'R3', $(2,3)$: 'R4'$\}$.

V.1 - Stratégie de la position la plus proche

Pour pouvoir obtenir le chemin souhaité, on envisage tout d'abord un algorithme permettant de sélectionner chaque nouvelle position l'une après l'autre sans jamais remettre en question les choix déjà effectués. La position suivante (position à l'instant t_{k+1}) est choisie parmi les positions voisines de la position courante (position à l'instant t_k) non encore atteintes et atteignables à l'instant t_{k+1}. On sélectionne celle qui est la plus proche (au sens de la distance euclidienne) de la position d'arrivée.

Q34. En utilisant la fonction sqrt du module math qui renvoie la racine carrée d'un nombre passé en argument (vous préciserez les commandes nécessaires à sa bonne utilisation), donner une fonction distance de signature et spécification suivante :

```
def distance(som1 :(int, int),som2 :(int, int))-> float :
    """
    Renvoie la distance euclidienne entre les deux positions som1 et som2
    """
```

Q35. Écrire une fonction voisin_dispo de signature et spécification suivante :

```
def voisin_dispo (G:graph,sommet :(int, int), pos_indispo :{(int, int):str})
    ->[(int, int)]:
    """
    G : graphe sous forme de liste d'adjacence
    sommet : position courante du robot (au temps t)
    pos_indispo : ensemble des positions indisponibles au temps (t+1)
    Renvoie les positions voisines accessibles au temps (t+1)
    """
```

Q36. Écrire une fonction voisin_suivant de signature et spécification suivante :

```
def voisin_suivant(ens_voisin :[(int, int)], Sarr :(int, int))->(int, int):
    """
    Renvoie la position la plus proche de la position Sarr (au sens de la
    distance euclidienne) parmi les positions contenues dans la liste ens_voisin
    """
```

Le **DR6** présente la fonction trajet incomplète, ainsi que sa signature et sa spécification.

Q37. Compléter sur le **DR6** le code de la fonction trajet. On fera l'hypothèse que $t_0 = 0$.

On note n l'ordre du graphe et m le degré du sommet du graphe de degré maximal. On fera l'hypothèse que la fonction position_skypod est de complexité constante.

Q38. En utilisant si nécessaire le **tableau 9**, évaluer la complexité en temps dans le pire cas de la fonction trajet.

Complexité constante	Complexité linéaire
• fonction sqrt • Ajout en fin de liste (liste.append) • Longueur de la liste (len)	• Test dans une liste (in) (en fonction de la taille de la liste)

Tableau 9 - Tableau des complexités

Q39. Donner, en justifiant, le chemin obtenu pour le graphe représenté sur la **figure 23**, en partant de la position (0,0) et arrivant à la position (2,2). On considère que toutes les positions sont atteignables à tout moment (il n'y a pas d'autre robot dans l'entrepôt). Que constatez-vous ? Conclure sur l'algorithme proposé.

V.2 - Stratégie par parcours de graphe

On envisage une autre stratégie : on souhaite parcourir le graphe afin d'obtenir le temps minimum pour atteindre chaque position depuis la position de départ. Pour cela, on donne sur le **DR7** la fonction temps_accès_position.

Q40. Compléter la signature et la spécification de la fonction temps_accès_position sur le **DR7**.

Q41. Compléter le graphe **DR8** en remplissant les cadres indiquant les temps d'accès minimum de chaque position selon la fonction temps_accès_position. On considérera un chemin partant de la position $(0,0)$ à $t_0 = 0$ arrivant à la position $(2,2)$ pour une valeur $\mathrm{tmax} = 10$. On suppose que la position $(1,2)$ n'est pas atteignable pour $t = 2$ et $t = 3$ à cause d'autres robots.

Q42. La fonction temps_accès_position est une version modifiée d'un algorithme de parcours de graphe. Lequel ? Justifier.

Q43. Sans écrire de code, expliquer le principe algorithmique à mettre en œuvre pour obtenir le trajet entre une position de départ Sdep et d'arrivée Sarr à l'aide du résultat de la fonction temps_accès_position.

ANNEXE 1

Caractéristiques du correcteur à avance de phase

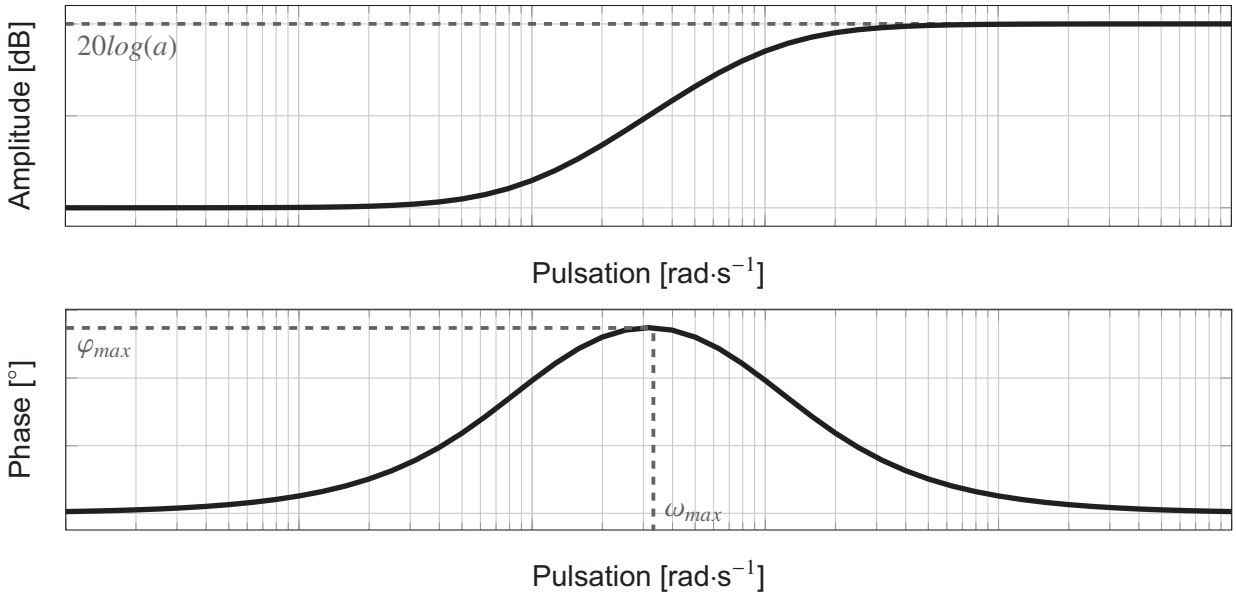


Figure 24 - Diagrammes de Bode du correcteur à avance de phase (sans gain) $\frac{1 + aT_{av}p}{1 + T_{av}p}$

$$\text{avec : } a > 1,\ sin(\varphi_{max}) = \frac{a-1}{a+1} \text{ et } \omega_{max} = \frac{1}{T_{av}\sqrt{a}}$$

ANNEXE 2

Extrait du cahier des charges du robot

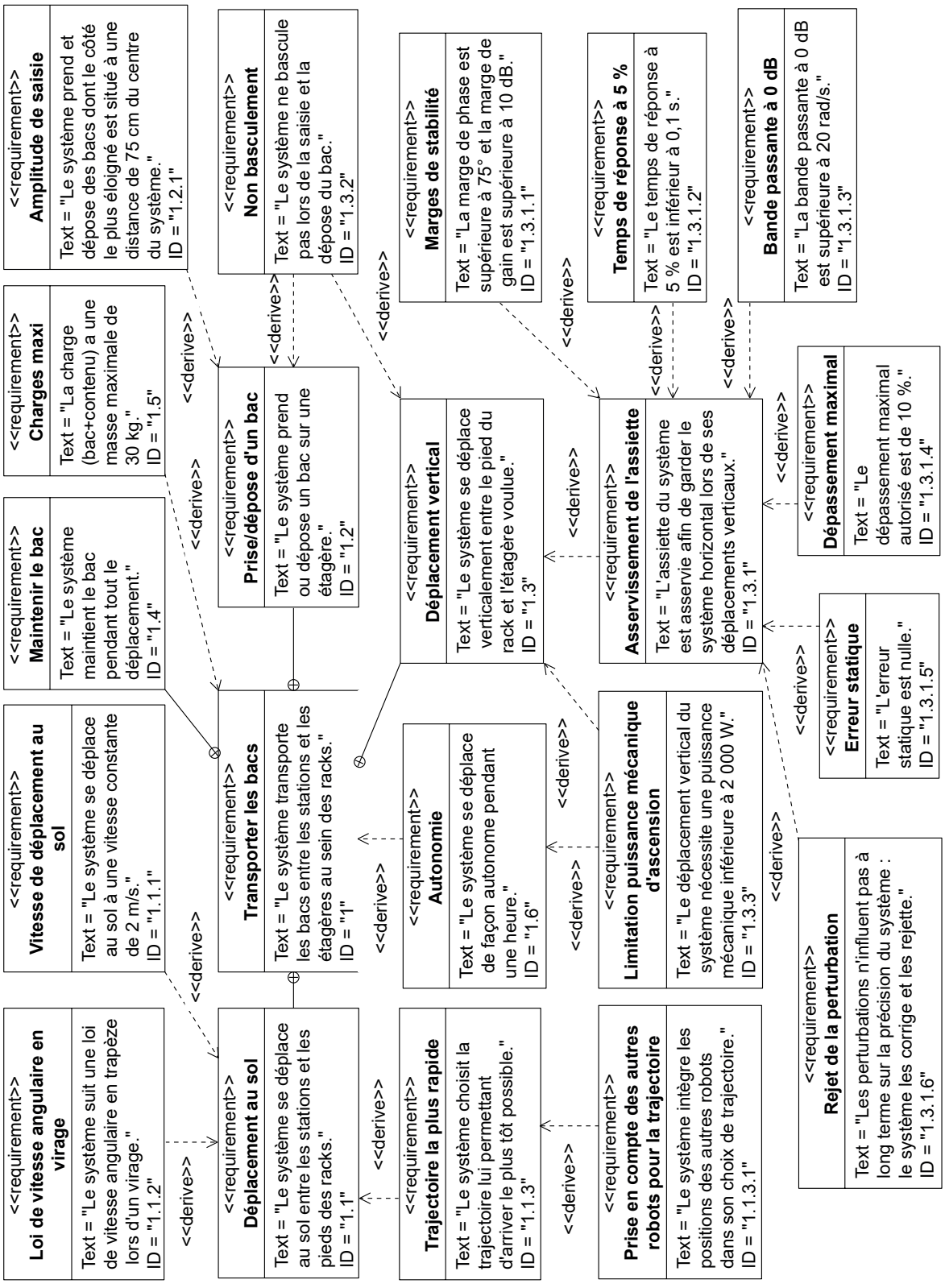


Figure 25 - Diagramme d'exigence extrait du cahier des charges du robot

FIN

IMPRIMERIE NATIONALE – 24 1093 – D'après documents fournis

CONCOURS COMMUN INP

Numéro d'inscription

Numéro de table

Né(e) le

Nom :

Prénom :

Emplacement QR Code

Filière : MP

Session : 2024

Épreuve de : SCIENCES INDUSTRIELLES

Consignes

- *Remplir soigneusement l'en-tête de chaque feuille avant de commencer à composer*
- *Rédiger avec un stylo non effaçable bleu ou noir*
- *Ne rien écrire dans les marges (gauche et droite)*
- *Numéroter chaque page (cadre en bas à droite)*
- *Placer les feuilles A3 ouvertes, dans le même sens et dans l'ordre*

MP7SI

DOCUMENT RÉPONSE

Ce Document Réponse doit être rendu dans son intégralité avec la copie (le QR Code doit être collé sur la première page de la copie).

Diverses syntaxes en Python

Python	
L.insert(pos,val)	insère la valeur val en position pos dans la liste L
[e for e in r if cond]	crée une liste et insère les éléments e contenus dans r si e respecte la condition (cond prend la valeur True).
{k : v for k in r}	crée un dictionnaire et insère les clés k contenues dans r en les associant à la valeur v.

NE RIEN ÉCRIRE DANS CE CADRE

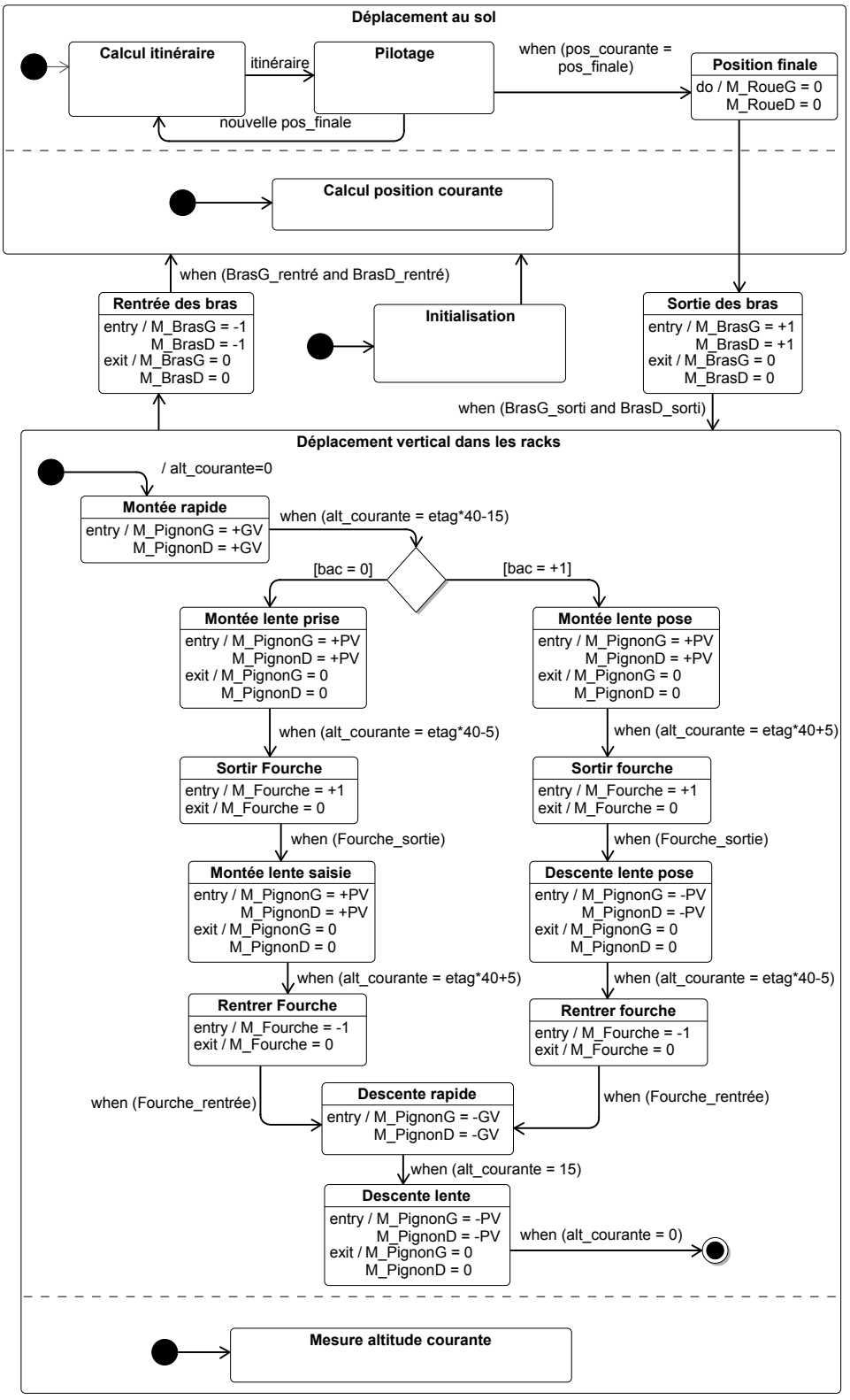
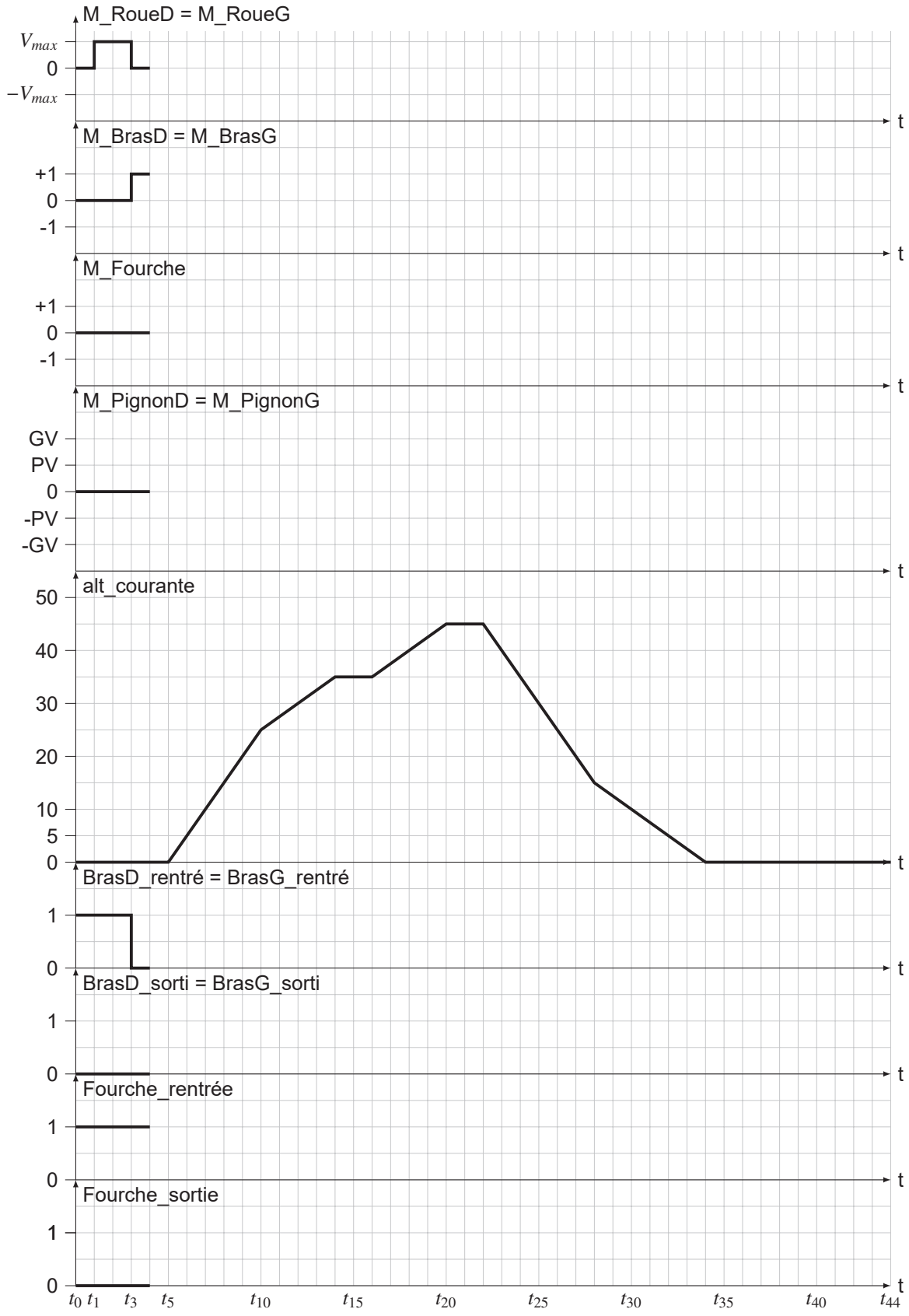


Figure 1 - Diagramme d'état décrivant le comportement du robot

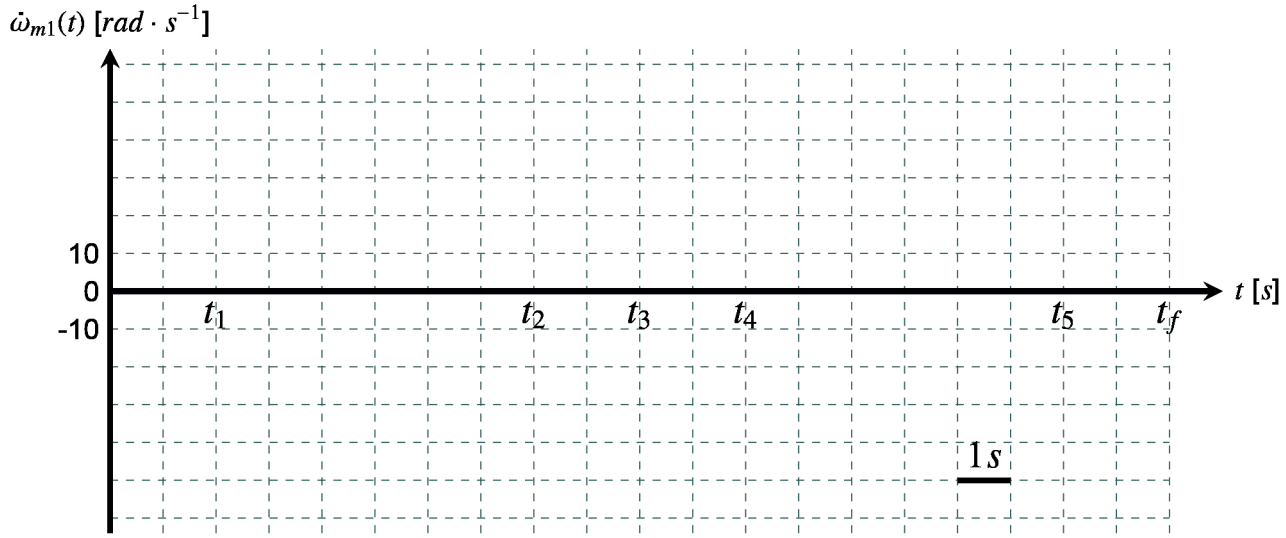
DR1 - Comportement séquentiel du système

Q1. Compléter le chronogramme



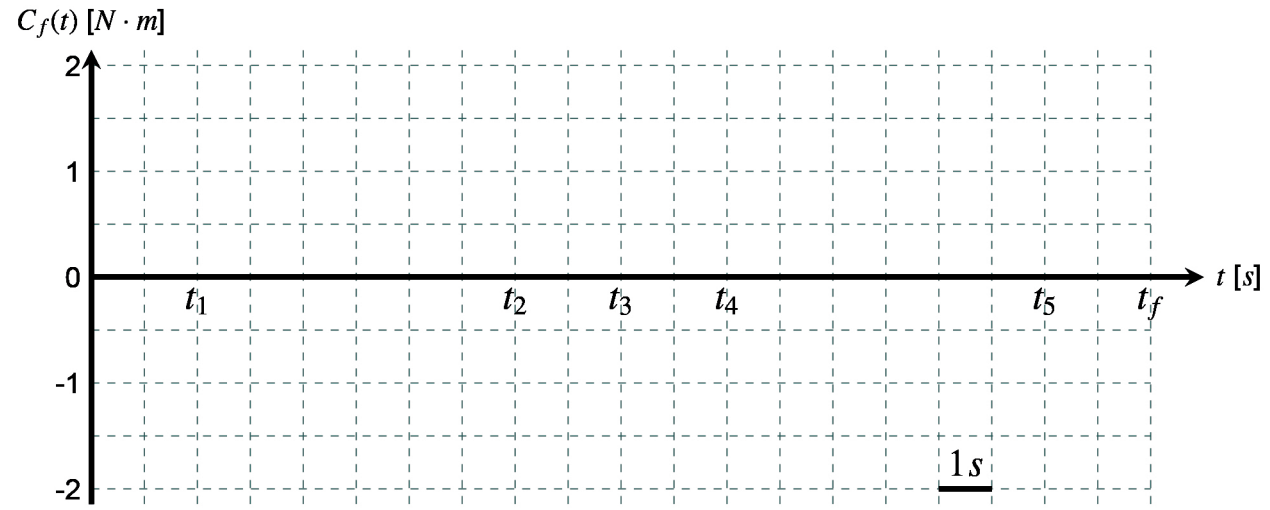
DR2 - Évolution de l'accélération angulaire

Q14. Tracer l'évolution de $\dot{\omega}_{m1}(t)$



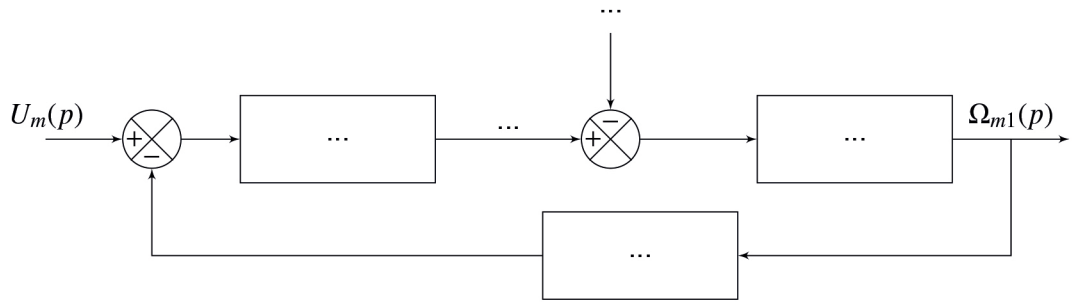
DR3 - Évolution du couple C_f

Q15. Tracer l'évolution de $C_f(t)$



DR4 - Schéma-blocs associé à un moteur

Q17. Compléter le schéma-blocs



Numéro d'inscription

Numéro de table

Né(e) le

Nom :

Prénom :

Emplacement QR Code

Filière : MP

Session : 2024

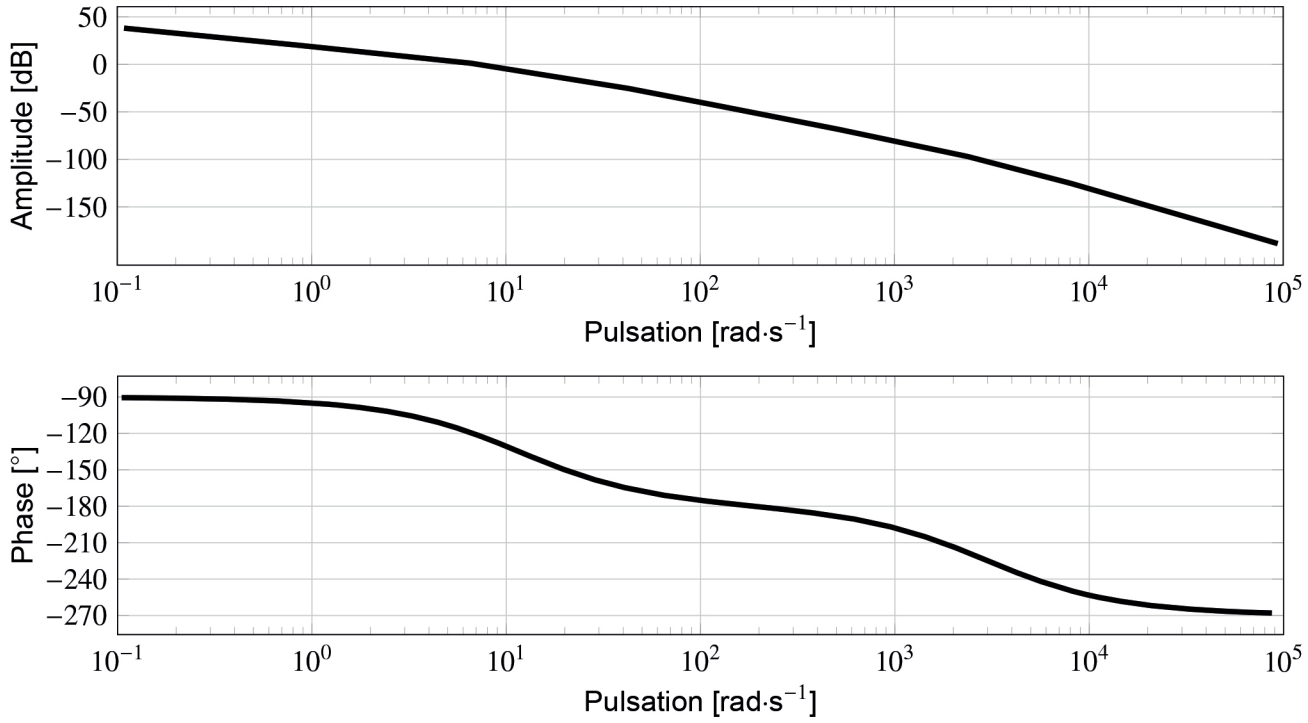
Épreuve de : SCIENCES INDUSTRIELLES

Consignes

- *Remplir soigneusement l'en-tête de chaque feuille avant de commencer à composer*
- *Rédiger avec un stylo non effaçable bleu ou noir*
- *Ne rien écrire dans les marges (gauche et droite)*
- *Numéroter chaque page (cadre en bas à droite)*
- *Placer les feuilles A3 ouvertes, dans le même sens et dans l'ordre*

DR5 - Réponses fréquentielle (BO) et temporelle (BF), système non corrigé ($C_{orr}(p) = 1$)

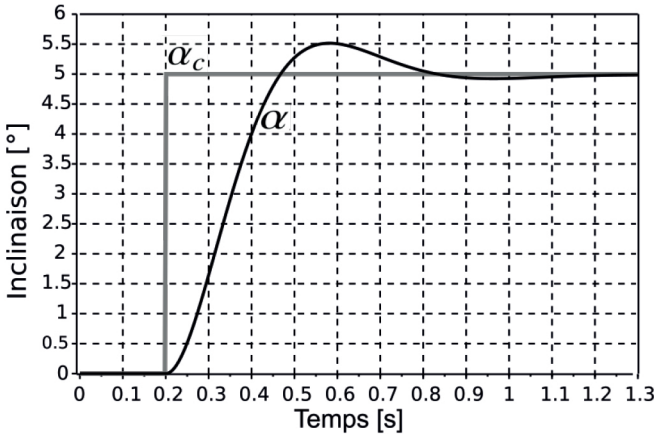
Q25 et Q26. Vérification des exigences et tracé asymptotique



Diagrammes de Bode de la fonction de transfert en boucle ouverte non corrigée.

Vérification exigences

NE RIEN ÉCRIRE DANS CE CADRE



Réponse temporelle du système non corrigé en boucle fermée

Vérification exigences

DR6 - Trajet par position la plus proche

Q37. Compléter la fonction trajet

```
def trajet(G:graph,tmax:int,Sdep:(int,int),Sarr:(int,int)) -> [(int,int)]:
    """
    G : graphe sous forme de liste d'adjacence
    Sdep : position de départ
    Sarr : position d'arrivée
    tmax : temps maximum pour effectuer le trajet
    Renvoie le chemin entre Sdep et Sarr en choisissant comme
    position suivante la position la plus proche de Sarr parmi les
    positions voisines non encore atteintes .
    Si le temps de parcours du chemin dépasse tmax,
    il n'y a pas de chemin (on renvoie une liste vide).
    """
    chemin=[Sdep]   #début du chemin en Sdep
    for t in range(tmax+1):

        if                                                  :          # à compléter
            return chemin

        # Calcul de l'ensemble des positions accessibles à t+1 :

        pos_indispo=position_skypod(                                  ) # à compléter

        voisin_disponible=voisin_dispo(                               ) # à compléter

        voisin_possible=[v for v in voisin_disponible if v not in chemin]

        # Si position(s) accessible(s) ==> la plus proche de Sarr
        if len(voisin_possible)>0:

            chemin.append(                                        )     # à compléter
    return []
```

DR7 - Temps d'accès par parcours de graphe

Q40. Compléter signature et spécification de la fonction temps_accès_position

```
def temps_accès_position(G        ,tmax       ,Sdep             )->                  :
    """ à compléter
    """
    # temps d'accès initiaux infinis :
    temps_visite={i:float("inf") for i in G}

    # gestion des sommets à visiter :
    file=[Sdep]

    for t in range(tmax+1):
        nbre_sommet=len(file) # ne pas visiter 2 fois un sommet à t
        while nbre_sommet>0:
            som=file.pop()       # som : nouvelle position courante
            nbre_sommet-=1
            if t<temps_visite[som]:      # temps pour atteindre som
                temps_visite[som]=t

            presence_voisin_non_visite=False
            for voisin in G[som]:

                # voisin non encore atteint :
                if temps_visite[voisin]>t :

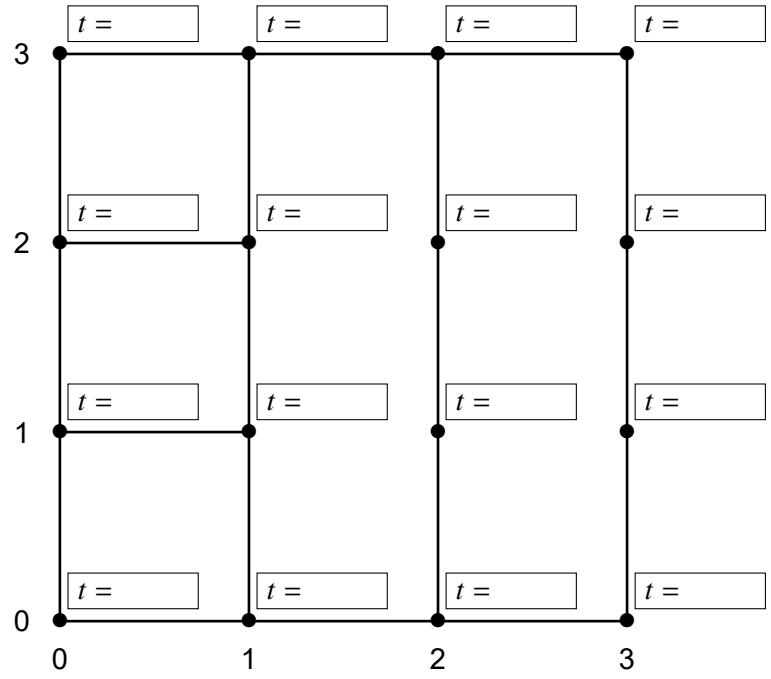
                    # et atteignable à t+1
                    if voisin not in position_skypod(t+1):

                        # évite les doublons dans file
                        if voisin not in file:
                            file.insert(0,voisin)
                    else:
                        presence_voisin_non_visite=True

                # ré-insertion des sommets ayant au moins un voisin
                # non encore atteint et non atteignable à t+1
                if presence_voisin_non_visite and som not in file:
                    file.insert(0,som)
    return temps_visite
```

DR8 - Temps d'accès aux positions de l'entrepôt exemple

Q41. Compléter les temps d'accès avec $\mathrm{Sdep}=(0,0)$, $\mathrm{Sarr}=(2,2)$ et $\mathrm{tmax}=10$.
La position $(1,2)$ n'est pas atteignable à $t = 2$ et $t = 3$.



Travailler les chapitres de ce sujet

Prepa Help suit le programme de MP chapitre par chapitre : cours détaillé, méthodes, pièges, exercices et problèmes corrigés pas à pas, tirés des concours. L'essai gratuit donne accès aux chapitres offerts de ta filière, sans carte bancaire.

Chapitres liés à ce sujet

- Ingénierie système : besoin, exigences et diagrammes SysML
- Cinématique des mécanismes : composition des mouvements, contact ponctuel, lois entrée-sortie et fermetures
- Cinématique du solide : paramétrage, vitesse, taux de rotation et torseur cinématique
- Cinétique : torseurs cinétique et dynamique, énergie cinétique, inertie équivalente
- Transmetteurs de puissance et trains d'engrenages
- Dynamique des systèmes de solides : principe fondamental, stratégie d'isolement, équilibrage
- Bases de Python : expressions, types, instructions, fonctions
- Complexité des algorithmes itératifs et récursifs

Ouvrir l'essai gratuit

Scanne le code ou clique sur le bouton.

prepahelp.com/#essai

Une question sur le concours ou sur ta filière : **écrire sur WhatsApp**

Les autres années et épreuves : prepahelp.com/annales/e3a/mp