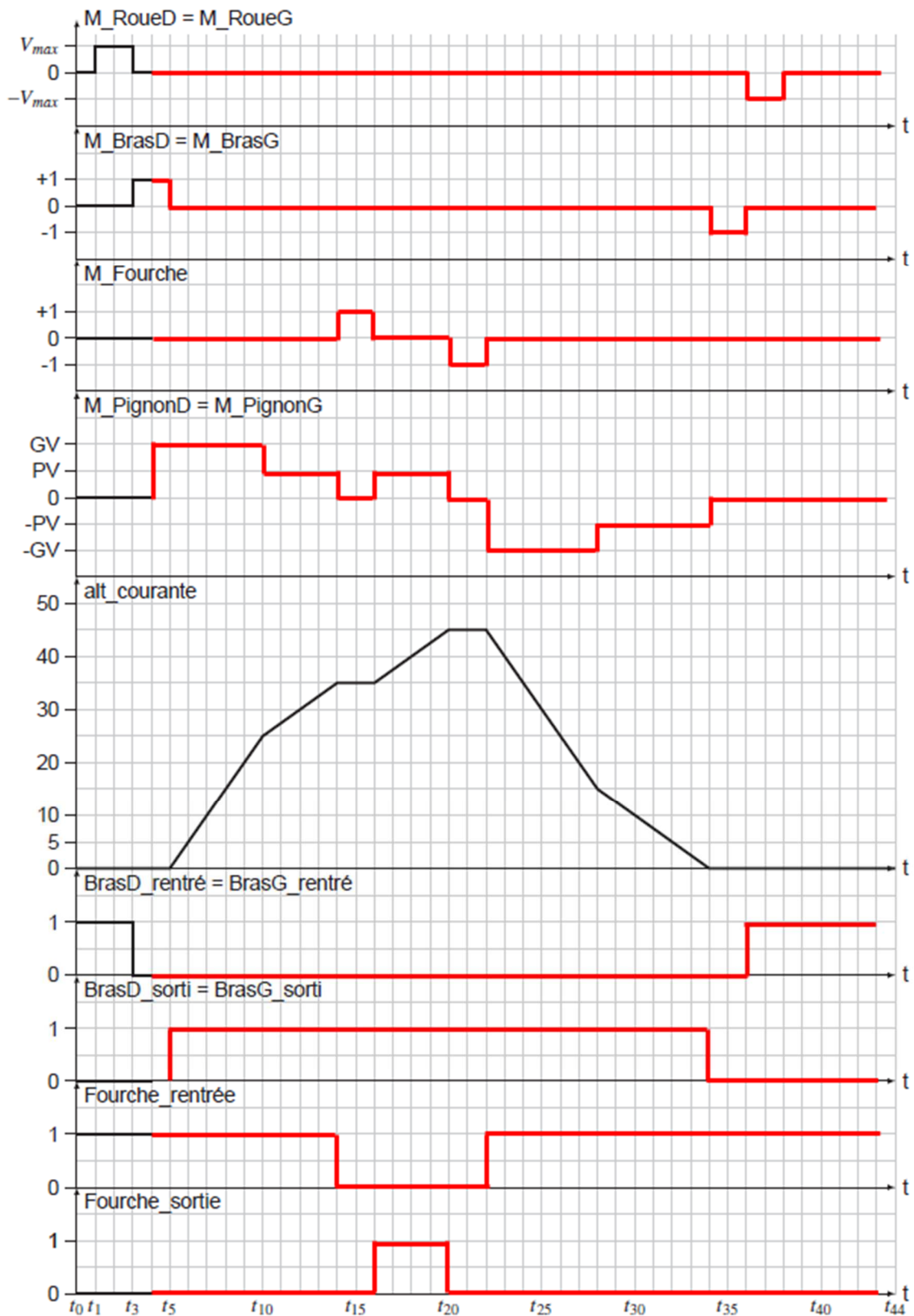


Corrigé DS 4 - Concours Blanc MP

Skypod, système automatisé de préparation de commande

PARTIE I - DEROULEMENT D'UNE PREPARATION DE COMMANDE

Question 1 Compléter le chronogramme sur le DR1 en suivant le scénario suivant :



PARTIE II - TRAJECTOIRE DU ROBOT EN PHASE DE DEPLACEMENT AU SOL

Question 2 À l'aide de la condition de roulement sans glissement en I, donner la relation entre $V(t)$, $\omega_{21}(t)$ et $\omega_{10}(t)$.

① **Hypothèse** : Roulement sans glissement en I de la roue 2 sur le sol 0 : $\vec{V}_{I \in 2/0} = \vec{0}$

② **Composition des vitesses au point I** : $\vec{V}_{I \in 2/0} = \vec{V}_{I \in 2/1} + \vec{V}_{I \in 1/0}$

③ **Calculs des 2 vitesses** :

$$\vec{V}_{I \in 2/1} = \vec{V}_{A \in 2/1} + \vec{IA} \wedge \vec{\Omega}_{2/1} = \vec{0} + R\vec{y}_1 \wedge \omega_{21}(t)\vec{z}_1 = R\omega_{21}(t)\vec{x}_1 \quad (\text{A sur l'axe du pivot entre 2 et 1})$$

$$\vec{V}_{I \in 1/0} = \vec{V}_{O_1 \in 1/0} + \vec{IO}_1 \wedge \vec{\Omega}_{1/0} = V(t)\vec{x}_1 + \left(R\vec{y}_1 + \frac{L}{2}\vec{z}_1\right) \wedge \omega_{10}(t)\vec{y}_1 = V(t)\vec{x}_1 - \frac{L}{2}\omega_{10}(t)\vec{x}_1$$

④ **Conclusion** :

En projection sur \vec{x}_1 : $R\omega_{21}(t) + V(t) - \frac{L}{2}\omega_{10}(t) = 0$

Commentaires :

Faire attention aux produits vectoriels !
Méthode à connaître.

Question 3 Par un raisonnement analogue, donner directement la relation entre $V(t)$, $\omega_{31}(t)$ et $\omega_{10}(t)$.

On remplace 2 par 3 et $L/2$ par $-L/2$: $R\omega_{31}(t) + V(t) + \frac{L}{2}\omega_{10}(t) = 0$

Question 4 En déduire $\omega_{10}(t)$ en fonction de $\omega_{21}(t)$ et de $\omega_{31}(t)$.

On isole $V(t)$ dans les expressions précédentes : $V(t) = \frac{L}{2}\omega_{10}(t) - R\omega_{21}(t) = -R\omega_{31}(t) - \frac{L}{2}\omega_{10}(t)$

$$L\omega_{10}(t) = R(\omega_{21}(t) - \omega_{31}(t))$$

$$\omega_{10}(t) = \frac{R}{L}(\omega_{21}(t) - \omega_{31}(t))$$

Question 5 Démontrer que pour une trajectoire rectiligne, $\omega_{21}(t) = \omega_{31}(t)$. En déduire $\omega_{moy}(t)$ et donner sa valeur.

En trajectoire rectiligne $\omega_{10}(t) = 0$ donc $\omega_{21}(t) = \omega_{31}(t) = \omega_{moy}(t)$.

$$\omega_{moy}(t) = -\frac{V(t)}{R} = -\frac{2}{20 \cdot 10^{-3}} = -100 \text{ rad} \cdot \text{s}^{-1}$$

Question 6 Donner dans ces conditions la valeur de l'accélération angulaire $\dot{\omega}_{10}(t)$ en fonction de $\dot{\omega}_{21}(t)$.

$\omega_{10}(t) = \frac{R}{L}(\omega_{21}(t) - \omega_{31}(t))$ donc $\dot{\omega}_{10}(t) = \frac{R}{L}(\dot{\omega}_{21}(t) - \dot{\omega}_{31}(t))$, de plus $\dot{\omega}_{21}(t) = -\dot{\omega}_{31}(t)$, donc :

$$\dot{\omega}_{10}(t) = \frac{2R}{L} \dot{\omega}_{21}(t)$$

Question 7 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} .

Sur la première phase (accélération) : $t_2 = \frac{\omega_{10max}}{\gamma_{10}}$

Même accélération et décélération en norme, donc même durée des phases 1 et 3 : $t_2 - t_1 = t_4 - t_3$

On veut faire un virage de 90° , donc l'intégrale de $\omega_{10}(t)$ (aire sous la courbe) vaut $\frac{\pi}{2} rad$:

$$\frac{\pi}{2} = \int_0^{t_4} \omega_{10}(t) dt = 2 \cdot \left(\frac{t_2 \cdot \omega_{10max}}{2} \right) + (t_3 - t_2) \omega_{10max} = t_3 \omega_{10max}$$

Donc $t_3 = \frac{\pi/2}{\omega_{10max}}$ et $t_4 = t_2 + t_3 = \frac{\omega_{10max}}{\gamma_{10}} + \frac{\pi/2}{\omega_{10max}}$

Question 8 À l'aide des figure 10 et figure 11, commenter l'influence de ω_{10max} sur la trajectoire du robot lors d'un virage.

Pour effectuer un virage de 90° , plus ω_{10max} est élevé plus le robot sort à l'extérieur du virage en comparaison d'un arc de cercle. Toujours avec de grandes vitesses, le rayon de courbure du virage est plus faible et la trajectoire se rapproche de l'angle droit (rayon de 1,25 m pour $\omega_{10max} = 3 rad.s^{-1}$ et rayon de 4,1 m pour $\omega_{10max} = 0,5 rad.s^{-1}$).

Dans tous les cas, le robot tourne bien de 90° .

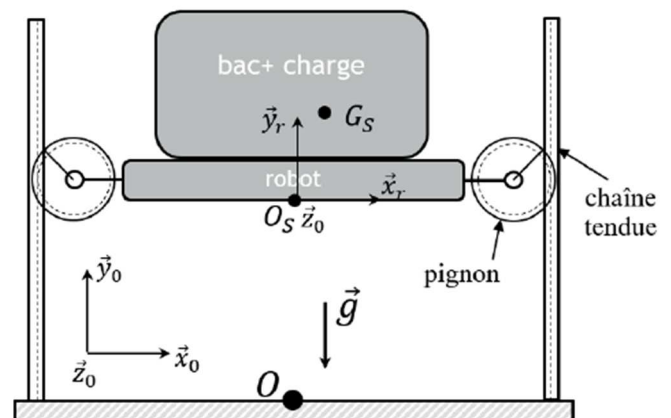
PARTIE III - SOLLICITATION DU ROBOT EN MOUVEMENT VERTICAL

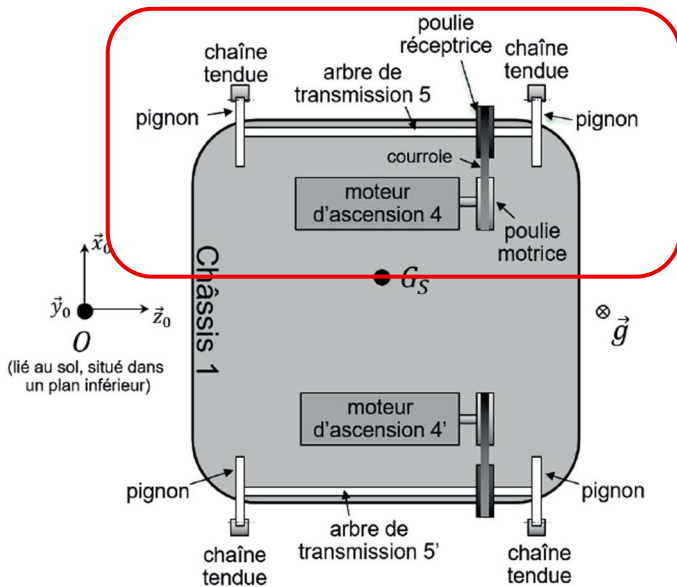
Question 9 Déterminer l'expression de l'énergie cinétique $E_c(S/R_0)$ de l'ensemble S par rapport à R_0 .

Pour un mouvement de translation du châssis, les deux moteurs tournent à la même vitesse mais avec des sens opposés. $\omega_{m1} = -\omega_{m2}$ et $\omega_{51} = -\omega_{52}$

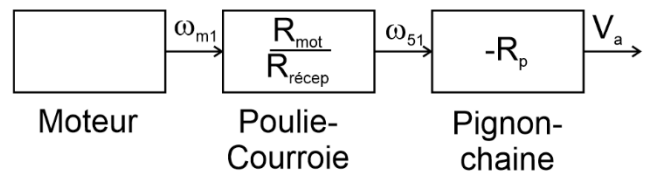
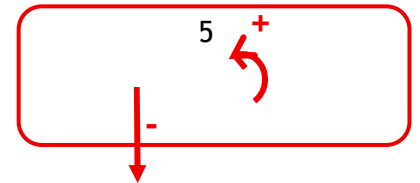
$$E_c(S/R_0) = \frac{1}{2} m_S \cdot (v_a(t))^2 + 2 \times \left(\frac{1}{2} J_m \omega_{m1}^2 + \frac{1}{2} J_5 \omega_{51}^2 \right)$$

Question 10 Exprimer le vecteur de la vitesse d'ascension du robot $v_a(t) \vec{y}_0$ en fonction de la vitesse de rotation d l'arbre moteur $\omega_{m1}(t)$.





5'



Roulement sans glissement du pignon sur la crémaillère : $v_a(t) = -\omega_{51} \cdot R_p$

(Si $\omega_{51} > 0$, $v_a(t) < 0$ d'après la figure ci-dessus)

Système poulie courroie : $\frac{\omega_{51}}{\omega_{m1}} = \frac{R_m}{R_r}$

$$v_a(t) = -\frac{R_m \cdot R_p}{R_r} \cdot \omega_{m1}(t)$$

Question 11 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.

On considère que J_{eq} est l'inertie équivalente ramenée sur l'arbre d'un des deux moteurs.

On réécrit : $2 \cdot E_c(S/R_0) = J_{eq} \cdot \omega_{m1}^2(t)$

$$2 \cdot E_c(S/R_0) = \left[m_s \cdot \left(\frac{R_m \cdot R_p}{R_r} \right)^2 + 2 \cdot J_m + 2 \cdot J_5 \left(\frac{R_m}{R_r} \right)^2 \right] \omega_{m1}^2$$

$$J_{eq} = m_s \cdot \left(\frac{R_m \cdot R_p}{R_r} \right)^2 + 2 \cdot J_m + 2 \cdot J_5 \left(\frac{R_m}{R_r} \right)^2$$

Question 12 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.

puissances extérieures galiléennes	puissances intérieures
<ul style="list-style-type: none"> Pesanteur : 	<ul style="list-style-type: none"> 2 moteurs : $P(\text{chassis} \leftarrow \text{mot} \rightarrow \text{rotors}) = 2C_m \cdot \omega_{m1}$

$P(pes \rightarrow S/R_0) = -m_s g \cdot v_a = m_s g \cdot \frac{R_m \cdot R_p}{R_r} \cdot \omega_{m1}$ $(\omega_{m1} < 0 \text{ en montée})$ <ul style="list-style-type: none"> Crémaillère sur les pignons : $P(0 \rightarrow S/0) = 0$ <p>car roulement sans glissement des pignons sur les crémaillères.</p>	$(C_m < 0 \text{ et } \omega_{m1}(t) < 0 \text{ en montée})$ <ul style="list-style-type: none"> Frottements sur arbres de transmission : $P(frott) = C_f \cdot \omega_{51} + C'_f \cdot \omega_{5r1} = (C_f - C'_f) \frac{R_m}{R_r} \omega_{m1}$ <p>(en montée $\omega_{m1} < 0$, $C_f > 0$ et $C'_f < 0$)</p> <ul style="list-style-type: none"> $P(S_i \leftrightarrow S_j) = 0$ (liaisons parfaites) $P(poulies \leftrightarrow courroie) = 0$ (RSG)
---	--

Somme des puissances :
$$P_{ext \rightarrow S/R_0} + P_{int \text{ à } S} = m_s g \cdot \frac{R_m \cdot R_p}{R_r} \cdot \omega_{m1}(t) + 2C_m \cdot \omega_{m1}(t) + (C_f - C'_f) \frac{R_m}{R_r} \omega_{m1}(t)$$

Question 13 Dédire 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 applique le théorème de l'énergie cinétique à l'ensemble S en mouvement par rapport à R_0 (galiléen) :

$$\frac{d}{dt} E_c(S/R_0) = P_{ext \rightarrow S/R_0} + P_{int \text{ à } S}$$

$$2 \cdot \frac{1}{2} (2 \cdot J_{eq}) \cdot \omega_{m1}(t) \cdot \dot{\omega}_{m1}(t) = m_s g \cdot \frac{R_m \cdot R_p}{R_r} \cdot \omega_{m1}(t) + 2C_m(t) \cdot \omega_{m1}(t) + (C_f - C'_f) \frac{R_m}{R_r} \omega_{m1}(t)$$

$$2 \cdot J_{eq} \cdot \dot{\omega}_{m1}(t) = m_s g \cdot \frac{R_m \cdot R_p}{R_r} + 2C_m(t) + (C_f - C'_f) \frac{R_m}{R_r}$$

$$C_m(t) = J_{eq} \cdot \dot{\omega}_{m1}(t) - \frac{m_s}{2} g \cdot \frac{R_m \cdot R_p}{R_r} - \frac{(C_f - C'_f) R_m}{2 R_r}$$

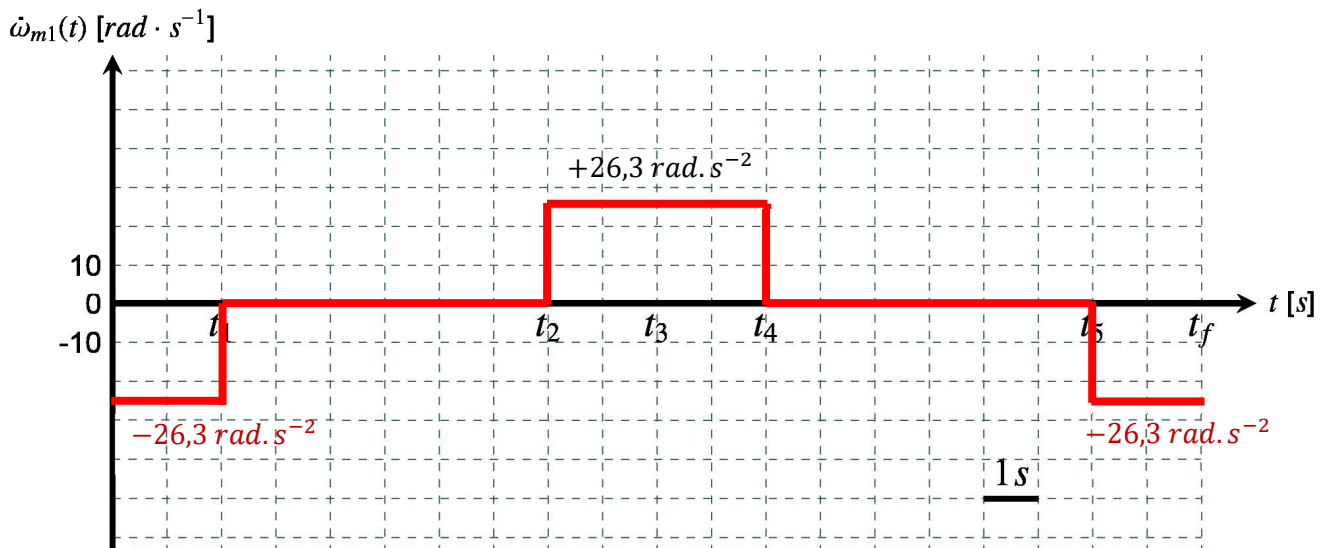
En accélération vers le haut chaque terme est négatif, car $\dot{\omega}_{m1} < 0$, $C_f > 0$ et $C'_f < 0$. Le couple moteur doit bien vaincre l'action du poids, des frottements et contribue à l'accélération.

Question 14 Compléter le DR2 en traçant l'évolution de l'accélération angulaire $\dot{\omega}_{m1}(t)$ au cours du temps.

$$\omega_{m1}(t) = -\frac{R_r}{R_m \cdot R_p} \cdot v_a(t) \text{ donc } \dot{\omega}_{m1}(t) = -\frac{R_r}{R_m \cdot R_p} \cdot \dot{v}_a(t)$$

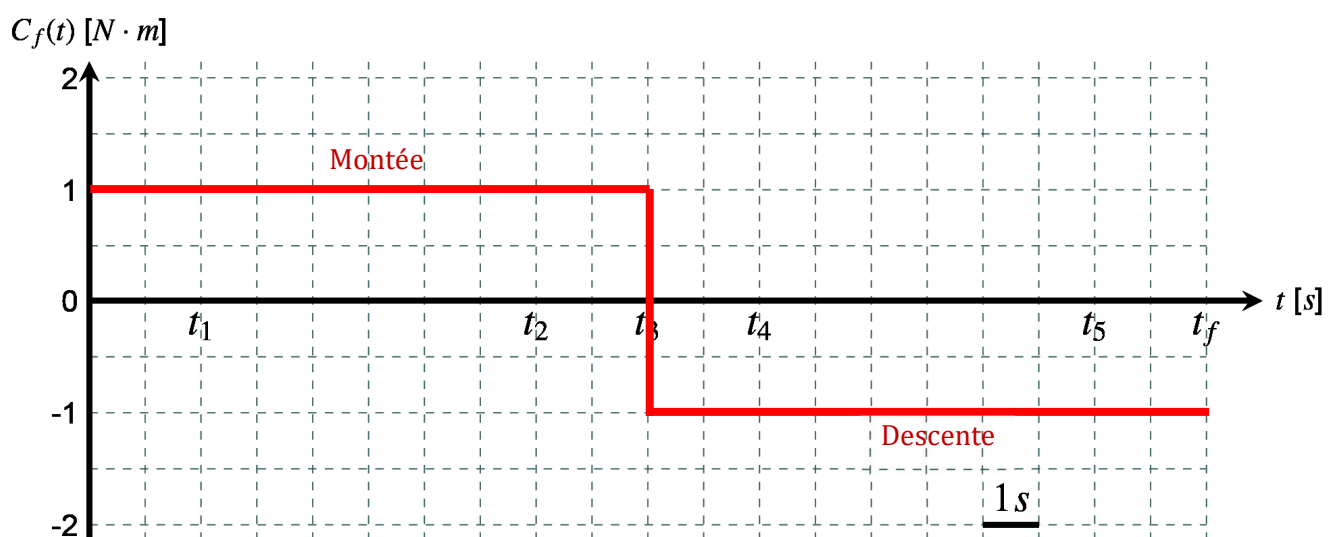
Pour la première phase d'accélération : $\dot{v}_a(t) = \frac{v_{max}}{t_1} = \frac{2}{2} = 1 m \cdot s^{-2}$ donc :

$$\dot{\omega}_{m1}(t) = -\frac{30}{30 \times 38 \cdot 10^{-3}} \cdot 1 = -26,3 \text{ rad} \cdot s^{-2}$$



Question 15 Tracer sur le DR3 l'évolution de C_f au cours du temps.

$C_f(t)$ est opposé à $\omega_{51}(t)$ qui est lui-même opposé à $v_a(t)$, donc $C_f(t)$ est du même signe que $v_a(t)$.



Question 16 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.

La puissance est maximale en phase de montée (car il faut vaincre l'action du poids), en phase d'accélération et à vitesse maximale. On se place donc à l'instant t_1 (ou juste avant).

$$\omega_{m1}(t_1) = -\frac{R_r}{R_m \cdot R_p} \cdot v_{max}$$

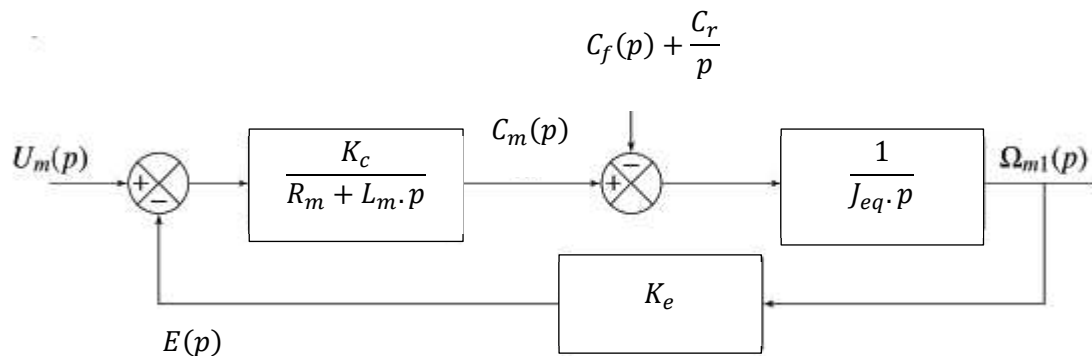
$$P_{max} = C_m(t_1) \cdot \omega_{m1}(t_1) = -17 \times -52,6 = 895 \text{ W}$$

$$P_{max} = 895 \text{ W} < 1000 \text{ W}$$

Le moteur envisagé d'une puissance de 1000 W est satisfaisant pour respecter les performances attendues.

PARTIE IV - ASSERVISSEMENT DE L'ASSIETTE DU ROBOT

Question 17 Compléter les blocs du DR4 au niveau des « ... », avec les fonctions de transfert et variables manquantes dans le schéma-blocs du moteur.



Question 18 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.

On isole l'ensemble S en équilibre dans R0 galiléen et à l'horizontale (α petit), on effectue le bilan des actions mécaniques extérieures :

- Pesanteur en G_S : $\{\tau_{pes \rightarrow S}\} = \left\{ \begin{array}{c} -m_s g \vec{y}_0 \\ \vec{0} \end{array} \right\}_{G_S}$
- Action de la chaîne gauche en C_g : $\{\tau_{chaîne \rightarrow pignon G}\} = \left\{ \begin{array}{c} Y_g \vec{y}_0 \\ \vec{0} \end{array} \right\}_{C_g}$
- Action de la chaîne droite en C_d : $\{\tau_{chaîne \rightarrow pignon D}\} = \left\{ \begin{array}{c} Y_d \vec{y}_0 \\ \vec{0} \end{array} \right\}_{C_d}$

On applique à S en équilibre dans R0 galiléen :

① Théorème du moment statique en C_g projeté sur \vec{z}_0 : $0 + Y_d(L + 2R_p) - m_s g \left(R_p + \frac{L}{2} + x_{G_S} \right) = 0$

$$Y_d = m_s g \left(\frac{R_p + \frac{L}{2} + x_{G_S}}{L + 2R_p} \right) = \boxed{m_s g \left(\frac{1}{2} + \frac{x_{G_S}}{L + 2R_p} \right) = Y_d}$$

② Théorème de la résultante statique projeté sur \vec{y}_0 : $Y_g + Y_d - m_s g = 0$

$$Y_g = m_s g - Y_d = \boxed{m_s g \left(\frac{1}{2} - \frac{x_{G_S}}{L + 2R_p} \right) = Y_g}$$

Question 19 En déduire l'expression des couples C_r^g et C_r^d .

Pour le moteur droit : en supposant le système poulie-courroie parfait, avec les mêmes rayons en entrée et sortie, on retrouve le même couple en sortie du système poulie courroie C_r^d exercé sur le pignon droit.

C_r^d est le couple exercée par le poids sur l'axe moteur, donc le couple exercée sur le pignon est $-C_r^d$.

Le pignon droit est donc en équilibre sous l'action du couple $-C_r^d$ sur l'axe (O_d, \vec{z}_0) de la force $Y_d \vec{y}_0$ en C_d et de l'action du châssis par l'intermédiaire d'une liaison pivot d'axe (O_d, \vec{z}_0) . Le théorème de moment statique en O_d projeté sur \vec{z}_0 : $-C_r^d + Y_d \cdot R_p + 0 = 0$

$$\text{Donc } C_r^d = Y_d \cdot R_p = R_p m_S g \left(\frac{1}{2} + \frac{x_{G_S}}{L+2R_p} \right)$$

$$\text{Par analogie : } C_r^g = -Y_g \cdot R_p = -R_p m_S g \left(\frac{1}{2} - \frac{x_{G_S}}{L+2R_p} \right)$$

Question 20 Exprimer $H_p(p)$ en fonction des caractéristiques du moteur. Préciser l'unité de $P(p)$.

$$H_p(p) = \frac{R_m + L_m \cdot p}{K_c}$$

$$P_p(p) \text{ en volt (V)}$$

Question 21 Justifier les signes du comparateur ayant pour entrées $h_d(p)$ et $h_g(p)$.



Avec α petit : $\sin \alpha = \frac{h_d - h_g}{L} \approx \alpha$ donc $\alpha = \frac{1}{L} \cdot (+h_d - h_g)$ ce qui est cohérent avec les signes du schéma-blocs.

Question 22 Exprimer $h_d(p)$ et $h_g(p)$ en fonction des variables $\varepsilon_c(p)$, $U_c(p)$ et $P_d(p)$ ou $P_g(p)$.

$$h_g(p) = R_p \cdot \frac{1}{p} \cdot H_m(p) \left(-P_g(p) + U_v(p) - K_{adapt} \cdot \varepsilon_c(p) \right)$$

$$h_d(p) = -R_p \cdot \frac{1}{p} \cdot H_m(p) \left(-P_d(p) - U_v(p) - K_{adapt} \cdot \varepsilon_c(p) \right) = R_p \cdot \frac{1}{p} \cdot H_m(p) \left(P_d(p) + U_v(p) + K_{adapt} \cdot \varepsilon_c(p) \right)$$

Question 23 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.

Si $P_d(p) = -P_g(p)$:

$$\alpha(p) = \frac{1}{L} (h_d(p) - h_g(p)) = R_p \cdot \frac{1}{p} \cdot H_m(p) \cdot 2 \cdot K_{adapt} \times \varepsilon_c(p) = H_{eq} \times \varepsilon_c(p)$$

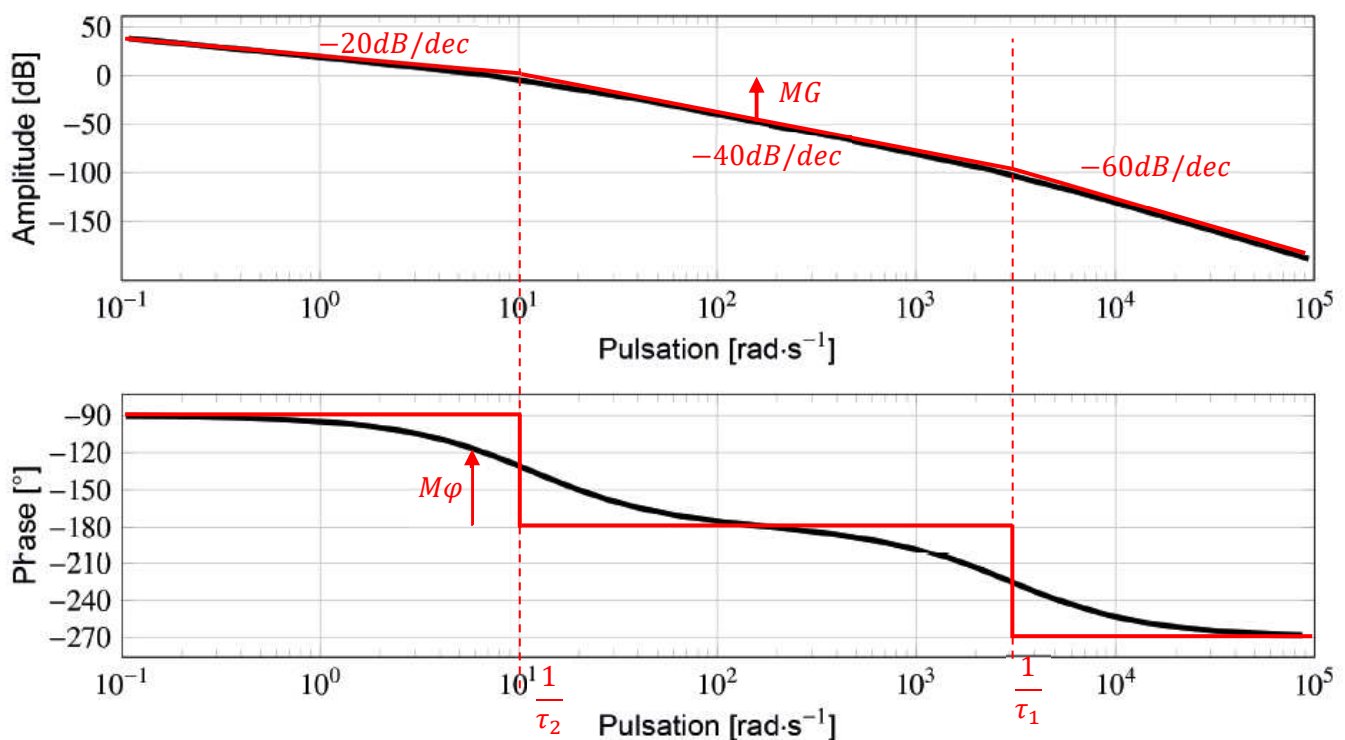
$$H_{eq}(p) = \frac{2 \cdot R_p \cdot H_m(p) \cdot K_{adapt}}{p \cdot L}$$

Question 24 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 $C_{orr}(p) = 1$.

$$H_{BO}(p) = \frac{\alpha(p)}{\varepsilon(p)} = C_{orr}(p) \times H_{eq}(p) = C_{orr}(p) \times \frac{2 \cdot R_p \cdot H_m(p) \cdot K_{adapt}}{p \cdot L}$$

$H_m(p)$ est d'ordre 2, donc si $C_{orr}(p) = 1$, $H_{BO}(p)$ est d'ordre 3 et le système est de classe 1.

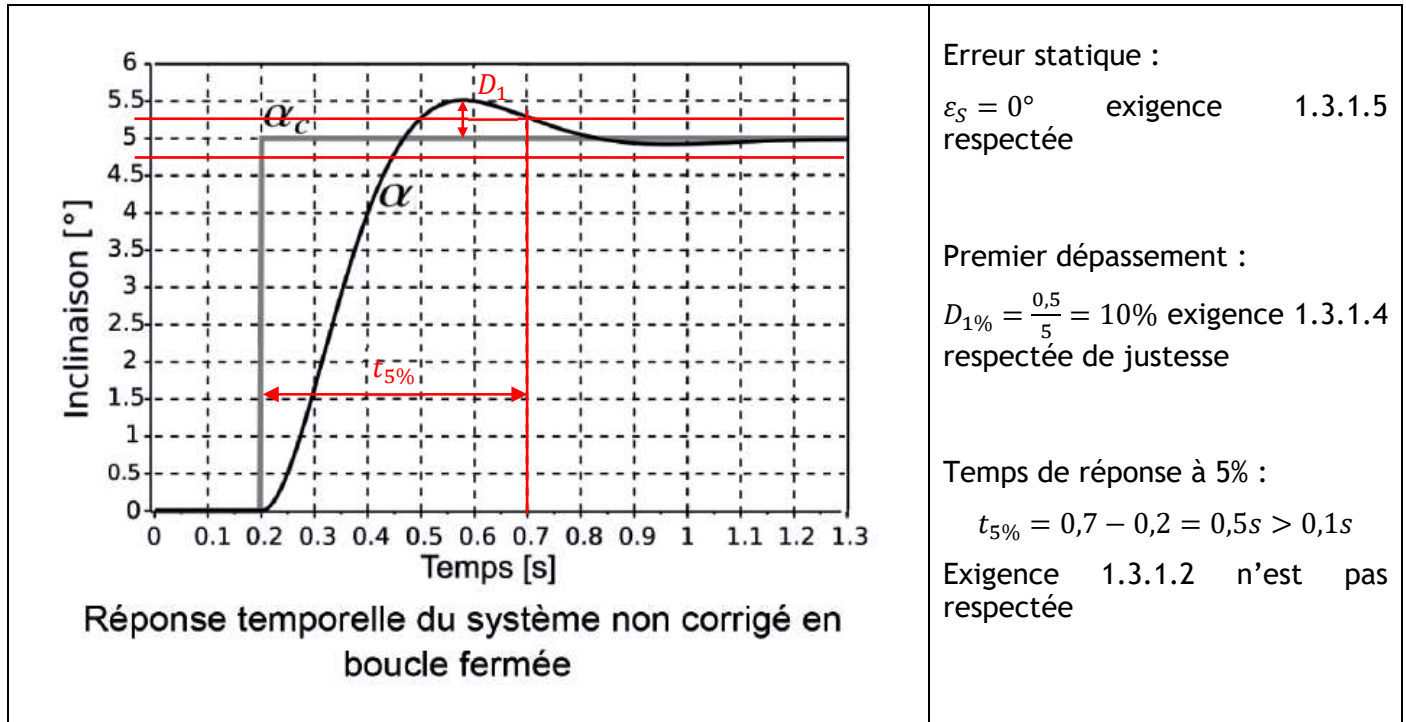
Question 25 Répondre sur le DR5 : vérifier si les exigences associées à l'asservissement en inclinaison du robot sont vérifiées. Mettre en place les tracées permettant la vérification des critères considérés.



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

On mesure $M\varphi = 65^\circ < 75^\circ$ et $MG = 50\text{dB} > 10\text{dB}$ donc l'exigence 1.3.1.1 n'est pas respectée à cause de la marge de phase.

On mesure une bande passante à 0 dB de $7 \text{ rad.s}^{-1} < 20 \text{ rad.s}^{-1}$, donc l'exigence 1.3.1.3 n'est pas respectée.



Erreur statique :

$\varepsilon_s = 0^\circ$ exigence 1.3.1.5 respectée

Premier dépassement :

$D_{1\%} = \frac{0,5}{5} = 10\%$ exigence 1.3.1.4 respectée de justesse

Temps de réponse à 5% :

$$t_{5\%} = 0,7 - 0,2 = 0,5s > 0,1s$$

Exigence 1.3.1.2 n'est pas respectée

Question 26 Sur le DR5, tracer les diagrammes de Bode asymptotiques correspondant aux tracés proposés.

Cf Question 25

Question 27 En déduire les valeurs numériques de τ_1 et de τ_2 .

D'après l'énoncé $\tau_2 > \tau_1$ donc $\omega_{c2} = \frac{1}{\tau_2} < \omega_{c1} = \frac{1}{\tau_1}$

$$\omega_{c2} = 10 \text{ rad.s}^{-1} \text{ donne } \tau_2 = 0,1 \text{ s}$$

$$\omega_{c1} = 3000 \text{ rad.s}^{-1} \text{ donne } \tau_1 = \frac{1}{3000} \text{ s} = 3,3 \cdot 10^{-4} \text{ s} = 0,33 \text{ ms}$$

Question 28 Justifier le choix d'un tel correcteur.

L'exigence 1.3.1.6 stipule que les perturbations ne doivent pas influencer la précision à long terme. Le correcteur Proportionnel Intégral (PI) permet d'ajouter une intégration avant la perturbation, et ainsi de rejeter les perturbations de type échelon. Mais le schéma-blocs de la figure 20 qui est utilisé pour la simulation du DR5 ne présente pas de perturbation...

Mais ce correcteur va nécessairement faire diminuer la marge de phase quel que soit le réglage, alors que cette dernière n'était déjà pas bonne. Utiliser uniquement un correcteur PI ne semble pas adapté, il faut également améliorer la marge de phase, d'où le double correcteur proposé dans la suite du sujet.

Commentaires :

Le rejet des perturbations désigne la capacité d'un système à **limiter** ou **annuler** l'effet des perturbations extérieures sur la sortie. Donc les effets de la perturbations sur l'erreur statique.

Question 29 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.

Le pôle dominant (le plus proche de l'axe des imaginaires) correspond à la constante de temps la plus grande (la plus lente), on choisit donc :

$$\tau_i = \tau_2 = 0,1 \text{ s}$$

$$H_{BO}(p) = \frac{K_p K_{eq}}{\tau_2 \cdot p^2 \cdot (1 + \tau_1 \cdot p)}$$

Question 30 Justifier le choix d'un tel correcteur.

Sur le diagramme de Bode de la FTBO avec correcteur PI, la marge de phase est de $4^\circ < 75^\circ$ ce qui est très faible. L'ajout du correcteur à avance de phase va permettre de retrouver la stabilité souhaitée en ajoutant une bosse de phase à l'endroit souhaité.

Question 31 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 satisfaire la marge de phase.

En $\omega = 20 \text{ rad.s}^{-1}$ ($=\omega_{0dB}$ visée) la marge de phase est de 3° sans le correcteur à avance de phase, il faut donc ajouter $\varphi_{max} = 72^\circ$ de phase en $\omega = 20 \text{ rad.s}^{-1}$

$$\sin(\varphi_{max}) = \frac{a-1}{a+1} \text{ donne } a = \frac{1+\sin(\varphi_{max})}{1-\sin(\varphi_{max})} = 40$$

Question 32 En déduire la valeur de T_{av} qu'il faut choisir.

$$\omega_{max} = \frac{1}{T_{av}\sqrt{a}} = 20 \text{ rad.s}^{-1} \text{ donne } T_{av} = \frac{1}{\omega_{max}\sqrt{a}} = 8.10^{-3} \text{ s} = 8 \text{ ms}$$

Question 33 À l'aide de la figure 22, donner une valeur approchée de K_p permettant l'asservissement en position du robot.

D'après la figure 22, seule la valeur $K_p = 3,5$ permet de satisfaire les exigences temporelles suivantes :

- $D_{1\%} = 10\% \leq 10\%$ (exigence 1.3.1.4)
- $t_{5\%} = 0,08\text{s} < 0,1\text{s}$ (exigence 1.3.1.2)
- $\varepsilon_S = 0^\circ$ (exigence 1.3.1.5)

PARTIE V - CHOIX D'ITINERAIRE D'UN ROBOT (INFORMATIQUE COMMUNE)

Question 34 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
    """
```

```
import math
```

```
def distance(som1 : (int, int), som2 : (int, int)) -> float:
```

```

"""
Renvoie la distance euclidienne entre les deux positions som1 et som2
"""
dist = 0
for i in range(len(som1)):
    dist += (som1[i] - som2[i])**2
return math.sqrt(dist)

```

Question 35 Écrire une fonction `voisin_dispo` de signature et spécification suivante :

```

def voisin_dispo(G : dict, 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).
    """

```

```

def voisin_dispo(G : dict, 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).
    """

```

```

    return [l for l in G[sommet] if l not in pos_indispo]

```

Cette forme de liste par compréhension est proposée dans les diverses syntaxes en python en début de document réponse et un exemple était proposé dans la fonction `trajet` du document réponse à la question Q37 pour créer la liste `voisin_possible`.

Question 36 Écrire une fonction `voisin_suisant` de signature et spécification suivante :

```

def voisin_suisant(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
    """
    sommet : position courante du robot (au temps t).
    pos_indispo : ensemble des positions indisponibles au temps (t+1).
    """

```

```

def voisin_suisant(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
    """

```

```

    dist_min = math.inf
    suivant = (None, None)
    for voisin in ens_voisin:
        if distance(voisin, Sarr) < dist_min:
            dist_min = distance(voisin, Sarr)
            suivant = voisin
    return suivant

```

Question 37 Compléter sur le DR6 le code de la fonction `trajet`. On fera l'hypothèse que $t_0 = 0$.

```

def trajet(G : dict, 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 plus proche de Sarr parmi les positions voisines non encore atteintes.

Si le temps de parcours du chemin dépasse t_{max} , 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 chemin[-1] == Sarr: # à compléter
        return chemin
    #calcul de l'ensemble des positions accessibles à t+1
    pos_indispo = position_skypod(t + 1) # à compléter
    voisin_dispo = voisin_dispo(G, chemin[-1], pos_indispo) # à compléter
    voisin_possible = [v for v in voisin_dispo if v not in chemin]
    # Si position(s) accessible(s) ==> la plus proche de Sarr
    if len(voisin_possible) > 0:
        chemin.append(voisin_suisant(voisin_possible, Sarr)) # à compléter
return []

```

Question 38 En utilisant si nécessaire le **tableau 9**, évaluer la complexité en temps dans le pire cas de la fonction trajet.

m : degré maximal d'un sommet, n : ordre du graphe

Dans le pire cas, la boucle for est exécutée $t_{max} + 1$ fois, dans cette boucle on trouve :

- La fonction position_skypod de complexité constante
- La fonction voisin_dispo est de complexité $m \times n$ (on remplit une liste de n éléments en réalisant pour chacun un test in de complexité linéaire m)
- La création de la liste voisin_possible est de complexité au pire de $m \times t_{max}$ (on parcourt les voisins_disponibles ($\leq m$) et on vérifie qu'il n'est pas dans chemin ($\leq t_{max} + 1$)).
- La fonction voisin_suisant est de complexité linéaire m (une boucle for parcourue au maximum m fois qui appelle la fonction distance de complexité linéaire)

Le nombre d'opération est donc de l'ordre de :

$$(t_{max} + 1) \times (cst1 + m \times n \times cst2 + m \times t_{max} \times cst3 + m \times cst4)$$

La fonction trajet est donc de complexité au pire de cas : $t_{max} \times m \times \max(t_{max}, n)$

Si on considère que le nombre de place indisponible est faible devant l'ordre du graphe n , dans la formule précédente, on peut assimiler n à une constante, ainsi :

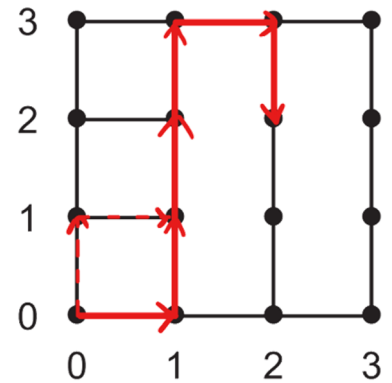
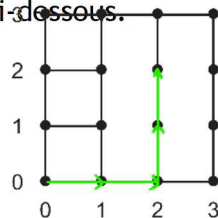
La fonction trajet est donc de complexité au pire de cas : $m \times t_{max}^2$

Question 39 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.

Le chemin parcouru est représenté sur la figure ci-contre.

En fonction de l'algorithme de sélection du vois le plus proche, le chemin en trait plein ou en trait discontinu sera choisi.

On constate que la solution n'est pas optimale, il existe un plus court chemin comme représenté en vert ci-dessous.



Question 40 Compléter la signature et la spécification de la fonction temps_accès_position sur le DR7.

```
def temps_accès_position(G : dict, tmax : int, Sdep : (int, int)) -> {(int, int) : float}:
    """ à compléter
    G : Graphe sous forme de liste d'adjacence.
    tmax : temps maximum pour effectuer le trajet.
    Sdep : position de départ.
    Renvoie un dictionnaire des temps minimum pour atteindre chaque sommet depuis Sdep
    """

    # temps d'accès initiaux infinis
    temps_visite = {i : float("int") for i in G}

    # gestion des sommets à visiter
    file = [Sdep]

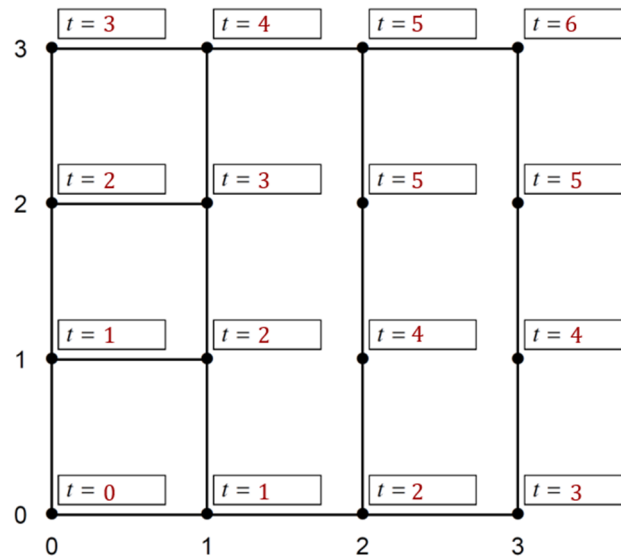
    for t in range(tmax + 1):
        nbre_sommet = len(file) # ne pas visiter 2 fois un sommet à t
        while nbre_sommet > 0:
            som = file.pop() # som : nouvelle position courante
            nbre_sommet -= 1
            if t < temps_visite[som] : # temps pour atteindre som
              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

Question 41 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 $t_{max} = 10$. On suppose que la position (1,2) n'est pas atteignable pour $t = 2$ et $t = 3$ à cause d'autres robots.

La position (1, 2) n'étant pas atteignable pour $t = 3$, elle sera atteinte au temps suivant.



Question 42 La fonction `temps_accès_position` est une version modifiée d'un algorithme de parcours de graphe. Lequel ? Justifier.

La fonction `temps_accès_position` reprend le principe de l'algorithme de Dijkstra pour le parcours de graphe.

La différence réside dans le fait que l'algorithme de Dijkstra s'utilise sur des graphes pondérés pour déterminer le plus court chemin, alors que pour cet algorithme la « pondération » (temps d'accès entre 2 sommets voisins) est variable dans le temps, car les sommets ne sont pas tous accessibles à chaque instant.

Question 43 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`.

Pour trouver le trajet entre `Sdep` et `Sarr`, après avoir calculé les temps d'accès minimum avec la fonction `temps_accès_position`, on part du sommet `Sarr` et on parcourt le graphe en suivant les voisins de proche en proche en choisissant celui qui a le temps d'accès minimum à chaque fois.