



Numéro
d'inscription

Numéro
de table

Né(e) le

Nom : _____

Prénom : _____

Emplacement
QR Code

Filière :

Session :

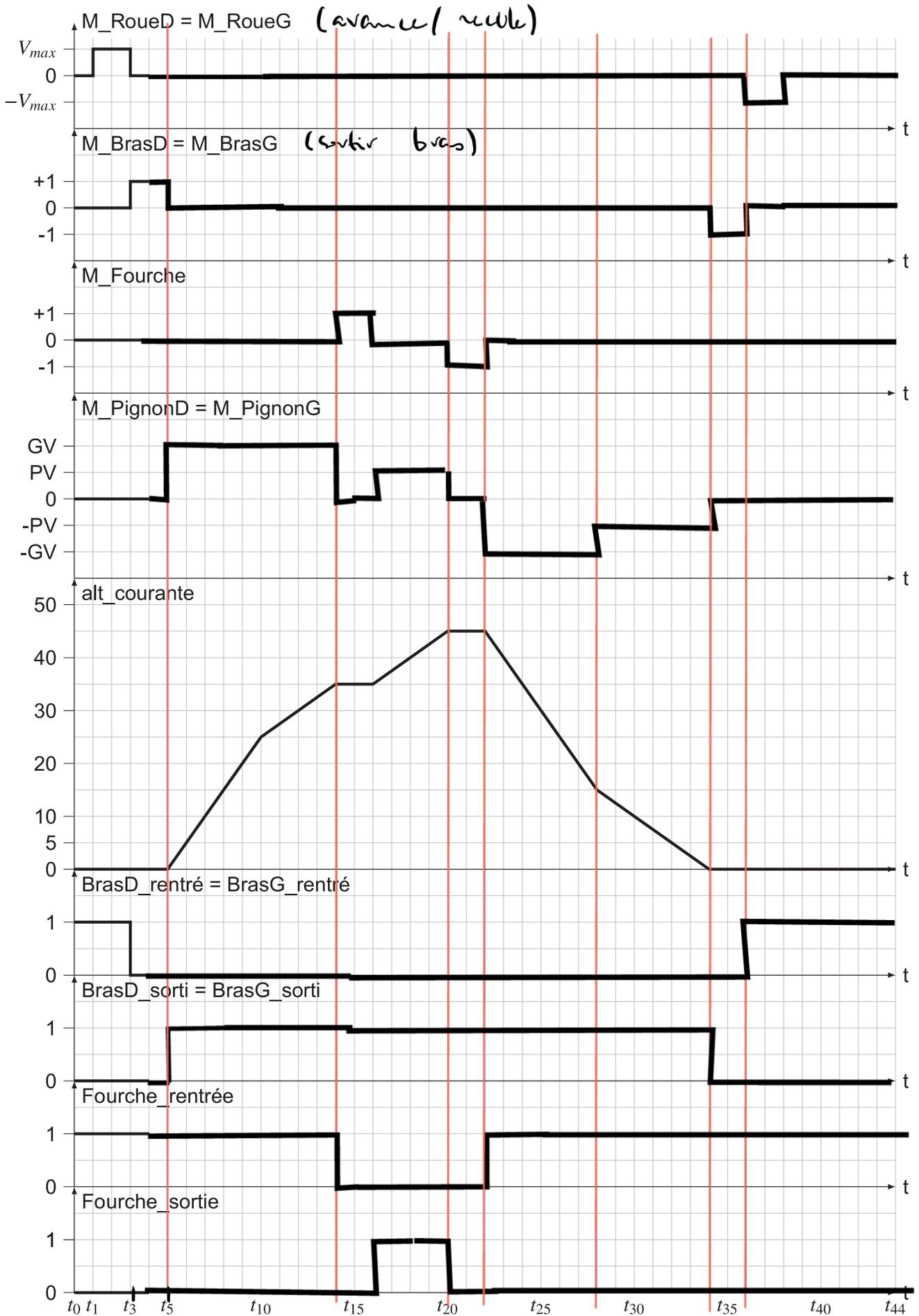
Épreuve de : **SCIENCES INDUSTRIELLES**

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)
- Numéroté chaque page (cadre en bas à droite)
- Placer les feuilles A3 ouvertes, dans le même sens et dans l'ordre

Skypod, système automatisé de préparation de commande

Q1. Compléter le chronogramme (voir page suivante).



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

condition de roulement sans glissement : $\vec{v}_{I \in 2/10} = \vec{0}$

Or $\vec{v}_{I \in 2/10} = \vec{v}_{I \in 2/11} + \vec{v}_{I \in 11/10}$ ou $\vec{v}_{I \in 2/11} = \vec{v}_{A \in 2/11} + \vec{IA} \wedge \underline{\omega_{21}} \cdot \vec{z}_1$

$= R \cdot \omega_{21} \cdot \vec{x}_1$

Et $\vec{v}_{I \in 11/10} = \vec{v}_{O_1 \in 11/10} + \vec{IO_1} \wedge \underline{\omega_{10}} \cdot \vec{y}_1 = v \cdot \vec{x}_1 - \frac{L}{2} \cdot \omega_{10} \cdot \vec{x}_1$

On a donc : $v - \frac{L}{2} \cdot \omega_{10} + R \cdot \omega_{21} = 0$

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

le raisonnement est le même mais $\vec{BO_1} = -\vec{AO_1}$ et donc :

$v + \frac{L}{2} \cdot \omega_{10} + R \cdot \omega_{31} = 0$

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

En soustrayant les équations :

$L \cdot \omega_{10} + R \cdot (\omega_{31} - \omega_{21}) = 0$

Et donc $\omega_{10} = \frac{R}{L} \cdot (\omega_{21} - \omega_{31})$

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

En trajectoire rectiligne $\omega_{10} = 0$ et donc $\omega_{21} = \omega_{31}$.

Dans ce cas : $\omega_{\text{moy}} = -\frac{v}{R} = -\frac{2 \text{ m/s}}{20 \cdot 10^{-3} \text{ m}} = -100 \text{ rad/s}$

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

On a : $\dot{\omega}_{10} = \frac{R}{L} \cdot (\dot{\omega}_{21} - \dot{\omega}_{31}) = -\dot{\omega}_{21}$

donc $\dot{\omega}_{10} = 2 \cdot \frac{R}{L} \cdot \dot{\omega}_{21}$

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} .

• $t_2 = \frac{\omega_{10max}}{\gamma_{10}}$ et si il y a la même décélération : $t_4 - t_3 = t_2$

• $\Delta\theta = \int_{t_1}^{t_4} \omega_{10}(t) \cdot dt = \text{aire sous la courbe} = t_3 \cdot \omega_{10max}$

donc $t_3 = \frac{\Delta\theta}{\omega_{10max}}$

• Enfin : $t_4 = \frac{\omega_{10max}}{\gamma_{10}} + \frac{\Delta\theta}{\omega_{10max}}$

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.

- Dans tous les cas, le robot n'a pas de trajectoire circulaire. Le robot est toujours "à l'extérieur" du virage.
- Plus ω_{10max} est grand, moins la trajectoire est circulaire.
- Le robot fait bien un virage de 90° .

Q9. Déterminer l'expression de l'énergie cinétique $Ec(S/R_0)$ de l'ensemble S par rapport au repère R_0 .

$Ec(S/R_0) = \frac{1}{2} \cdot m_s \cdot v_a^2 + \left[\frac{1}{2} \cdot J_s \cdot \omega_{S1}^2 + \frac{1}{2} \cdot J_m \cdot \omega_{m1}^2 \right] \times 2$

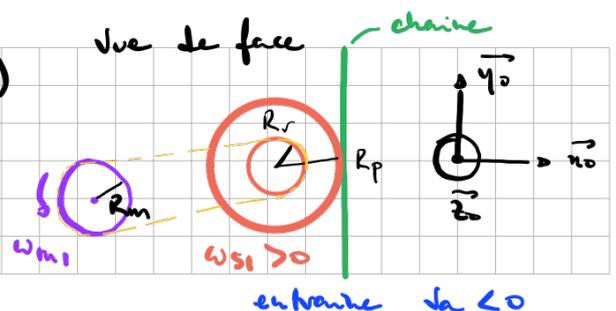
car $\omega_{S1} = -\omega_{S1}$ et $\omega_{m1} = -\omega_{m1}$

Q10. Exprimer le vecteur de la vitesse d'ascension du robot $v_a(t) \cdot \vec{y}_0$ en fonction de la vitesse de rotation de l'arbre moteur $\omega_{m1}(t)$.

• $v_a = \pm R_p \cdot \omega_{S1} = -R_p \cdot \omega_{S1}$ (voir schéma)

• Et $\frac{\omega_{S1}}{\omega_{m1}} = \frac{R_m}{R_r}$

Donc $v_a = -R_p \cdot \frac{R_m}{R_r} \cdot \omega_{m1}$



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 avec $E_c(S/R_0) = \frac{1}{2} \cdot J_{eq} \cdot \omega_{m1}(t)^2$.

On a donc :

$$E_c(S/R_0) = \frac{1}{2} \cdot \underbrace{\left[m_s \cdot R_p^2 \cdot \frac{R_m^2}{R_r^2} + 2 \cdot J_m + 2 \cdot J_5 \cdot \frac{R_m^2}{R_r^2} \right]}_{J_{eq}} \cdot \omega_{m1}^2$$

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.

Extérieures : $P_{\text{poids}} \rightarrow S/R_0 = -m_s \cdot g \cdot v_a = m_s \cdot g \cdot R_p \cdot \frac{R_m}{R_r} \cdot \omega_{m1}$

Intérieures : $P_{\text{motrice}} = \underbrace{2}_{\text{2 moteurs}} \cdot C_m \cdot \omega_{m1}$

$P_{\text{frottements}} = 2 \cdot \underbrace{C_f}_{\text{avec } C_f < 0 \text{ si } \omega_{S1} > 0} \cdot \omega_{S1} = 2 \cdot C_f \cdot \frac{R_m}{R_r} \cdot \omega_{m1}$

Donc :

$$P_{\text{puissances}} = \left[m_s \cdot g \cdot R_p \cdot \frac{R_m}{R_r} + 2 \cdot C_m + 2 \cdot C_f \right] \cdot \omega_{m1}$$

Q13. Dédurre 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.

Le théorème de l'énergie cinétique appliqué à S donne :

$$P_{\text{puissances}} = J_{\text{eq}} \cdot \dot{\omega}_{m1}$$

Et donc :

$$m_s \cdot g \cdot R_p \cdot \frac{R_m}{R_r} + 2 \cdot C_m + 2 \cdot C_f = J_{\text{eq}} \cdot \dot{\omega}_{m1}$$

D'où :

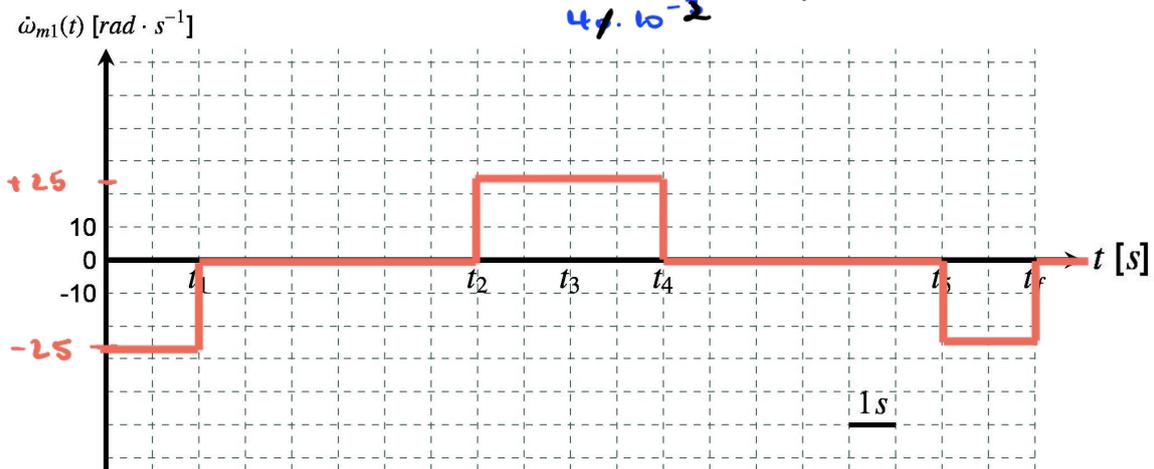
$$C_m = \frac{1}{2} \cdot J_{\text{eq}} \cdot \dot{\omega}_{m1} - C_f - \frac{1}{2} \cdot m_s \cdot g \cdot R_p \cdot \frac{R_m}{R_r}$$

$\underbrace{\hspace{10em}}_{C_a} \quad \underbrace{\hspace{2em}}_{1 \text{ N.m}} \quad \underbrace{\hspace{10em}}_{\approx 16 \text{ N.m}}$

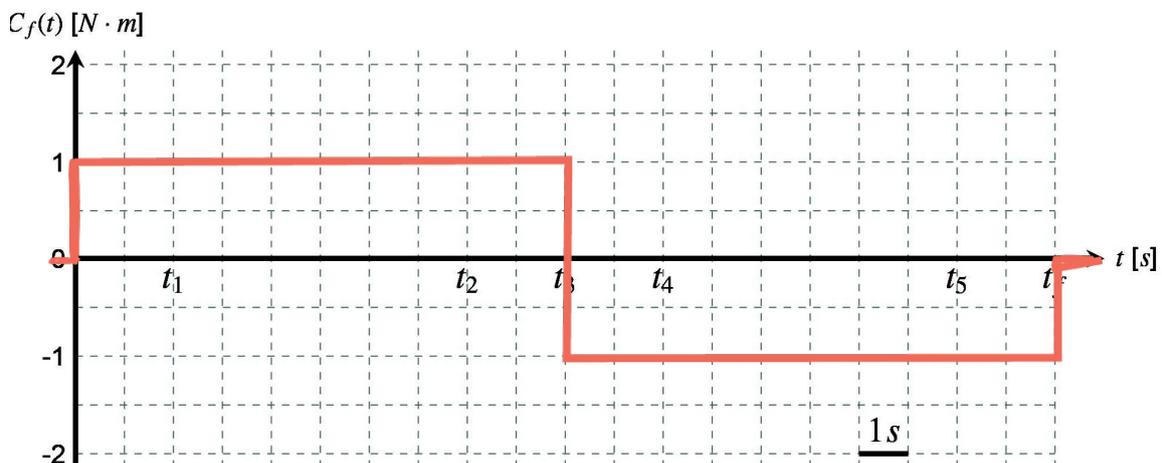
Lors de l'accélération: $\gamma = \pm \frac{v_{\text{max}}}{t_1} \approx 1 \text{ m/s}^2$ et $R_p \cdot \frac{R_m}{R_r} \approx 38 \cdot 10^{-3} \text{ m}$
 $\dot{\omega}_{m1} = -\gamma \cdot \frac{R_r}{R_p R_m} \approx \pm 25 \text{ rad/s}^2$

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

$$\pm \frac{1}{4 \cdot 10^{-2}} = 0,25 \cdot 100 = 25$$



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



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.

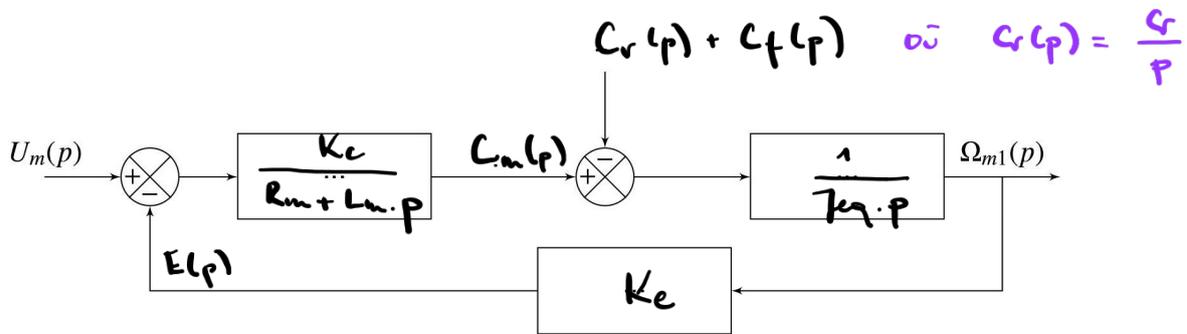
Lorsque $t = t_1 - dt$ (où $dt \ll t_1$) alors :

$$C_m = C_{m,max} \quad \text{et} \quad \omega_{m1} \approx \omega_{m1,max}$$

donc $P_{max} = \overbrace{C_{m,max}}^{-17 \text{ N.m}} \cdot \underbrace{\omega_{m1,max}}_{= -v_{max} \cdot \frac{R_r}{R_p \cdot R_m}} = -v_{max} \cdot \frac{R_r}{R_p \cdot R_m} = -\frac{2 \text{ m/s}}{40 \cdot 10^{-3} \text{ m}} \approx -50 \text{ rd/s}$

Donc $P_{max} \approx 850 \text{ W} < 1000 \text{ W}$ donc la motorisation envisagée est adaptée.

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



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.

J'isole S soumis à :

- charge \rightarrow pignon $\cdot g$ (PG)
- " " " " " " " " (pd)
- ps \rightarrow S

J'écris le th. des moments en C_d et en projection sur \vec{z}_0 :

$$\vec{M}_{C_d, c \rightarrow pg} \cdot \vec{z}_0 + \vec{M}_{C_d, c \rightarrow pd} \cdot \vec{z}_0 + \vec{M}_{C_d, ps \rightarrow s} \cdot \vec{z}_0 = 0$$

$\vec{M}_{C_d, c \rightarrow pg} \cdot \vec{z}_0 = \vec{M}_{C_g, c \rightarrow pg} \cdot \vec{z}_0 + (C_d C_g \wedge (Y_g \vec{y}_0)) \cdot \vec{z}_0 = 0$

$$\approx -2 \cdot (R_p + \frac{L}{2}) \cdot Y_g$$

$\vec{M}_{C_d, ps \rightarrow s} \cdot \vec{z}_0 = \vec{M}_{G_s, ps \rightarrow s} \cdot \vec{z}_0 + (C_d G_s \wedge (-m_s \cdot g \cdot \vec{y}_0)) \cdot \vec{z}_0$

$$\approx (R_p + \frac{L}{2} - x_{G_s}) \cdot m_s \cdot g$$

Donc :

$$Y_g = \left[\frac{1}{2} - \frac{x_{G_s}}{2 \cdot R_p + L} \right] \cdot m_s \cdot g$$

Et de la même manière, avec un th. des moments en C_G :

$$Y_d = \left[\frac{1}{2} + \frac{x_{G_s}}{2 \cdot R_p + L} \right] \cdot m_s \cdot g$$

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

• Il faut que $-C_r^d \cdot \omega_{ms} = \gamma_d \cdot v_a = \text{Puissance "résistive"}$

$$\text{donc } C_r^d = -\gamma_d \cdot \frac{v_a}{\omega_{ms}} = +\gamma_d \cdot R_p \cdot \frac{R_m}{R_r}$$

$$C_r^d = \left[\frac{1}{2} + \frac{2as}{2 \cdot R_p + L} \right] \cdot ms \cdot g \cdot R_p \cdot \frac{R_m}{R_r}$$

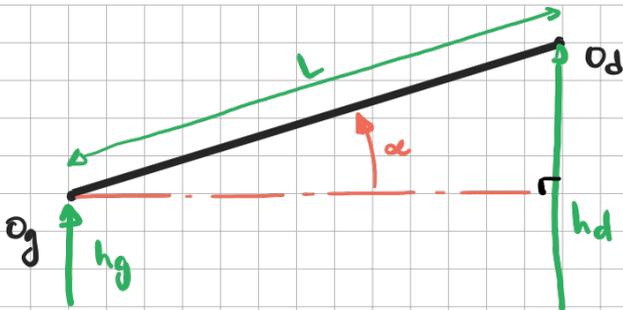
• Et de même: $C_r^g = - \left[\frac{1}{2} - \frac{2as}{2 \cdot R_p + L} \right] \cdot ms \cdot g \cdot R_p \cdot \frac{R_m}{R_r}$

Q20. Exprimer $H_p(p)$ en fonction des caractéristiques du moteur. Préciser l'unité de la grandeur représentée par $P_v(p)$.

• On a directement: $H_p(p) = \frac{R_m + L_m \cdot p}{K_c}$

• P est une tension (Volts).

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



On a donc:

$$\sin \alpha = \frac{h_d - h_g}{L}$$

et $\alpha \ll 1$

$$\text{Donc: } \alpha \approx \frac{h_d - h_g}{L}$$

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

$$h_d(p) = -H_m(p) \cdot \frac{R_p}{P} \cdot \left[-\varepsilon_c(p) \cdot K_{edapt} - U_v(p) - P_d(p) \right]$$

$$h_g(p) = H_m(p) \cdot \frac{R_p}{P} \cdot \left[-\varepsilon_c(p) \cdot K_{edapt} + U_v(p) - P_g(p) \right]$$

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.

$$\begin{aligned}
 \text{On a } \alpha(p) &= \frac{1}{L} \cdot (h_d(p) - h_g(p)) \\
 &= \frac{1}{L} \cdot \left(+ H_m(p) \cdot \frac{R_f}{F} \cdot [+ \varepsilon_c(p) \cdot K_{adapt} + \cancel{U_r(p)} + \cancel{P_d(p)}] \right. \\
 &\quad \left. - H_m(p) \cdot \frac{R_f}{F} \cdot [- \varepsilon_c(p) \cdot K_{adapt} + \cancel{U_r(p)} + \cancel{P_d(p)}] \right) \\
 \alpha(p) &= \left[2 \cdot \frac{H_m(p) \cdot R_f}{L \cdot F} \cdot K_{adapt} \cdot \varepsilon_c(p) \right] \\
 &= H_{eq}(p)
 \end{aligned}$$

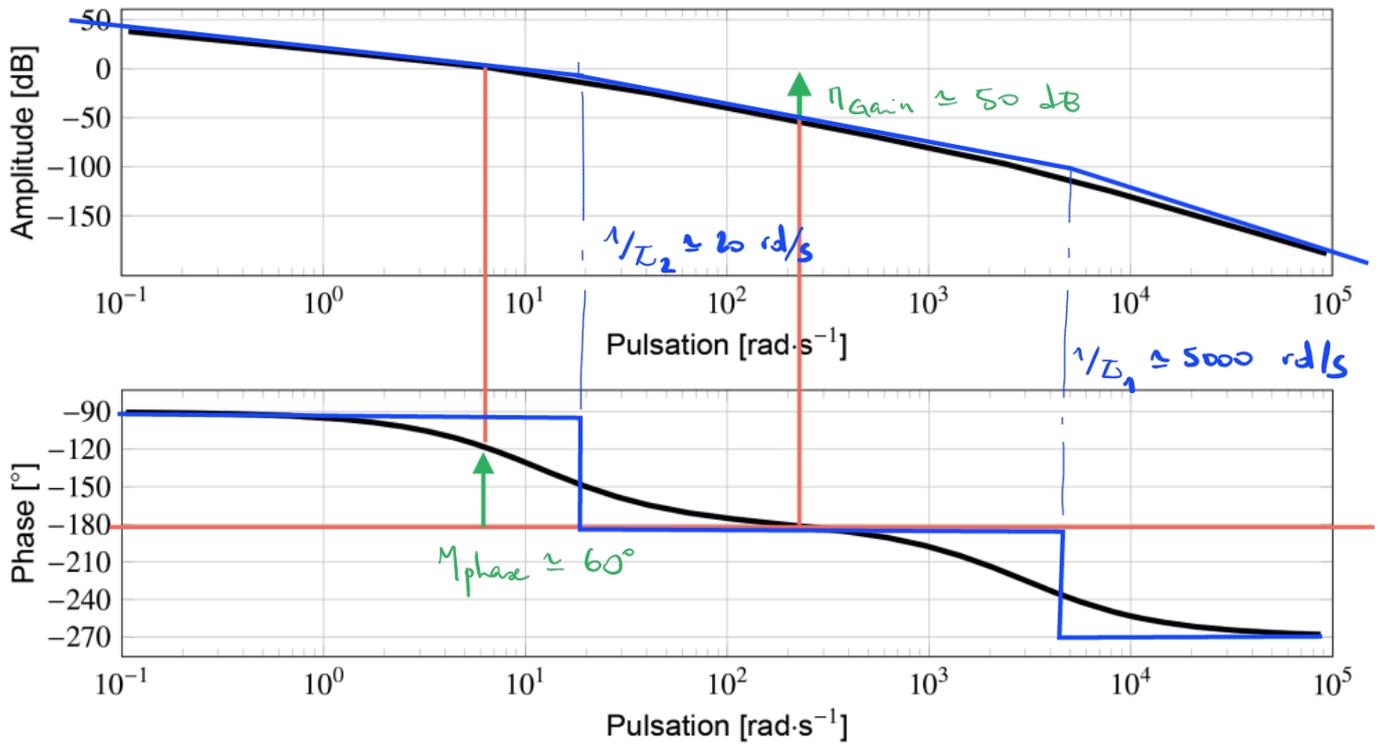
Q24. Donner la fonction de transfert en boucle ouverte (notée $H_{Bo}(p)$) de ce système en fonction de $H_m(p)$, $C_{orr}(p)$, K_{adapt} et des paramètres géométriques. Préciser son ordre et sa classe si $C_{orr}(p) = 1$.

$$H_{Bo}(p) = C_{orr}(p) \cdot H_{eq}(p) = 2 \cdot \frac{H_m(p) \cdot R_f}{L \cdot F} \cdot K_{adapt} \cdot \underbrace{C_{orr}(p)}_{=1}$$

H_m st d'ordre 2 donc H_{Bo} st d'ordre 3
 (et classe 0)

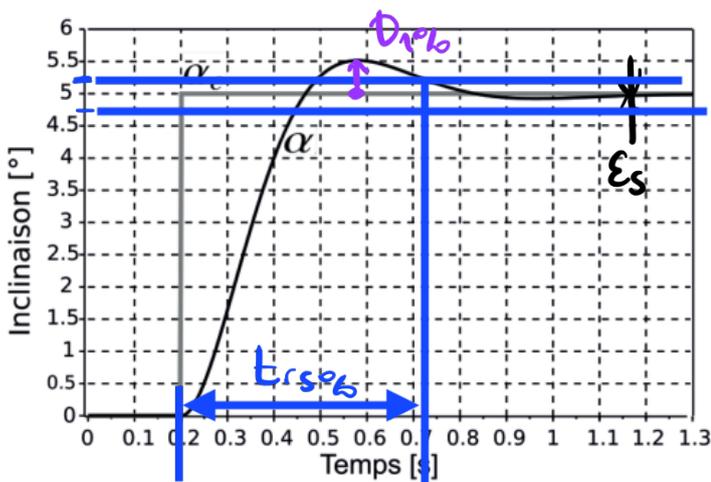
et de classe 1.

Q25. 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.

Vérification exigences	
On a	$M_{\text{Gain}} > 10 \text{ dB}$ (exigence respectée)
	$M_{\text{Phase}} < 75^\circ$ (exigence NON respectée)



Réponse temporelle du système non corrigé en boucle fermée

Vérification exigences	
•	Erreur statique $E_s = 0$ donc exigence respectée.
•	$t_{rs0.6} \approx 0,52 \text{ s} > 0,1 \text{ s}$ donc exigence NON respectée.
•	$D_{10.6} \approx 10\% \leq 10\%$ donc exigence respectée.

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 .

$$\left. \begin{array}{l} \frac{1}{T_2} \approx 20 \text{ rad/s} \\ \frac{1}{T_1} \approx 5000 \text{ rad/s} \end{array} \right\} \text{ donc } T_2 \approx 0,05 \text{ s} \quad \text{et} \quad T_1 \approx 2 \cdot 10^{-4} \text{ s}$$

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.

On a $T_i = T_2$ et dans ce cas :

$$H_{BO}(p) = K_p \cdot \frac{1 + T_i p}{T_i p} = \frac{K_{eq}}{p \cdot (1 + T_1 p) \cdot (1 + T_2 p)}$$

$$H_{BO}(p) = \frac{\frac{K_p \cdot K_{eq}}{T_i}}{p^2 \cdot (1 + T_1 p)}$$

Q30. Justifier le choix d'un tel correcteur.

On voit que la marge de phase est d'environ 5° , il faut donc augmenter la phase, ce que permet ce correcteur.

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

• Il faut ajouter environ 70° de phase, il faut donc :

$$\sin(\gamma_{max}) = \frac{a-1}{a+1} \quad \text{donc} \quad a \cdot \sin \gamma_{max} + \sin \gamma_{max} = a - 1$$

$$\text{donc} \quad a \cdot (1 - \sin \gamma_{max}) = 1 + \sin \gamma_{max}$$

$$\text{donc} \quad a = \frac{1 + \sin \gamma_{max}}{1 - \sin \gamma_{max}} \approx \frac{1,94}{0,04} \approx \frac{2}{0,04} \approx 50 \approx a$$

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

• Il faut aussi $\omega_{dB} = 20 \text{ rad/s} = \frac{1}{T_{av} \cdot \sqrt{a}}$ donc $T_{av} = \frac{1}{\omega_{dB} \cdot \sqrt{a}}$ *

* $\frac{1}{20 \cdot \sqrt{50}} \approx 0,05 \cdot \frac{1}{\sqrt{49}} \approx \frac{0,05}{7} \approx \frac{0,049}{7} \approx 0,01$

$T_{av} \approx 0,01 \text{ s}$

Q33. À l'aide de la figure 22, donner une valeur approchée de K_p permettant l'asservissement en position du robot.

• On vérifie:

- $\epsilon_s = 0$ (toujours vrai)
- $D_{10\%} \leq 10\%$ (donc $\alpha_{max} < 5,5^\circ$)
- $t_{rise} \leq 0,1 \text{ s}$

• La seule valeur de K_p permettant de vérifier ces trois critères est :

$K_p = 3,5$

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
    """

import math

def distance(som1, som2):
    return sqrt((som1[0] - som2[0])**2 + (som1[1] - som2[1])**2)
```

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)
    """

def voisin_dispo(G, pos_indispo):
    return [l for l in G[sommet] if l not in pos_indispo]
```

voir doc fournie en début de sujet.

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
    """
```

```
def voisin_suivant(ens_voisin, Sarr):
    suivant = ens_voisin[0]
    dist_min = distance(suivant, Sarr)

    for i in range(1, len(ens_voisin)):
        if distance(ens_voisin[i], Sarr) < dist_min:
            dist_min = distance(ens_voisin[i], Sarr)
            suivant = ens_voisin[i]

    return suivant
```

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

```
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 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_disponible=voisin_dispo(G, chemin[-1], pos_indispo) # 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(voisin_suivant(voisin_possible, Sarr)) # à compléter
```

```
return []
```

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

Boucle "for" de réalisé t_{max} fois. Dans cette boucle:

- posit-skypod est de complexité constante -
- voisin-dispo est de complexité n , la longueur du graphe.
- voisin-possible est de complexité $n \times t_{max}$ (au pire).

parcours de
voisin-dispo

test dans chemin

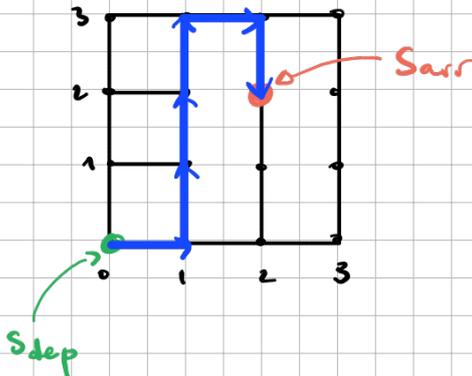
La complexité, dans le pire des cas, est de l'ordre de :

$$n \times t_{max}^2$$

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.

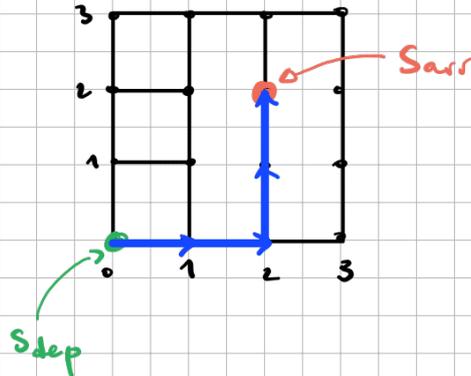
le robot va se déplacer de la manière suivante:

▲ : Parcours



Alors qu'il est plus rapide de faire:

▲ : Parcours



L'algorithme ne donne pas le chemin optimal.

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

```
def temps_accès_position (G          ,tmax          ,Sdep          )->          :
    """ à compléter

    G: graphe sous forme de liste d'adjacence
    tmax: temps maximum pr effectuer le trajet
    Sdep: sommet de départ
    Renvoie un dictionnaire des temps minimum pr atteindre
    chaque sommet depuis Sdep

    """
    # 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 :

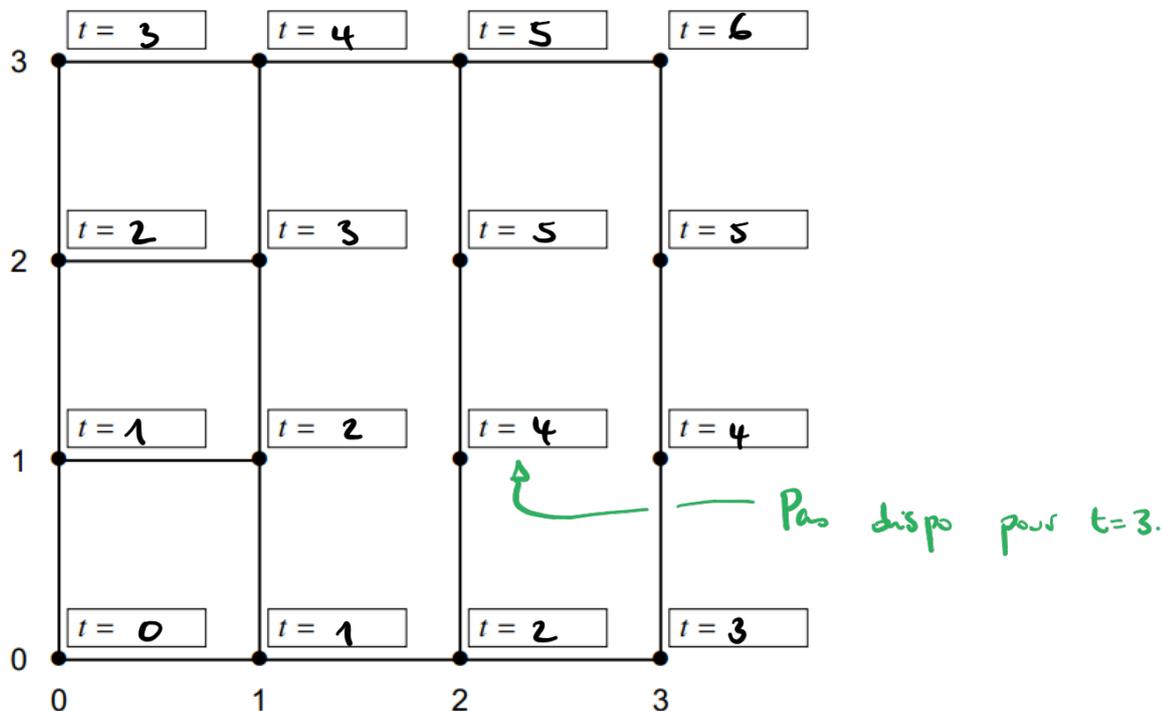
                    # et atteignable à t+1
                    if voisin not in position_skypod(t+1):

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

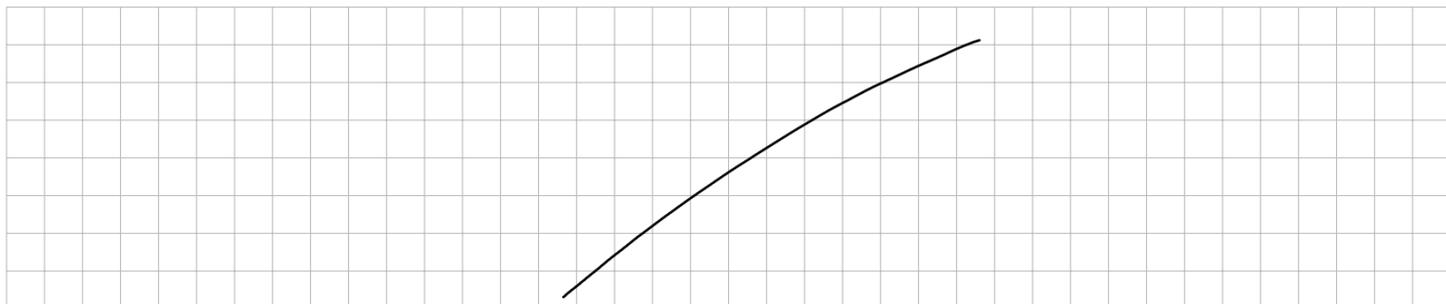
                # ré-insertion des sommets ayant au moins un voisin
                # non encore atteint et non atteignable à t+1
                if presence_voisin_non_visite and som not in file :
                    file.insert(0,som)
    return temps_visite
```

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 $t_{max} = 10$. 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'est pas atteignable à $t = 2$ et $t = 3$.



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 S_{dep} et d'arrivée S_{arr} à l'aide du résultat de la fonction `temps_accès_position`.

