CONCOURS COMMUN

Numéro d'inscription

Numéro de table

Né(e) le

Nom : _____

Prénom : _____

Emplacement QR Code

Filière :

Session :

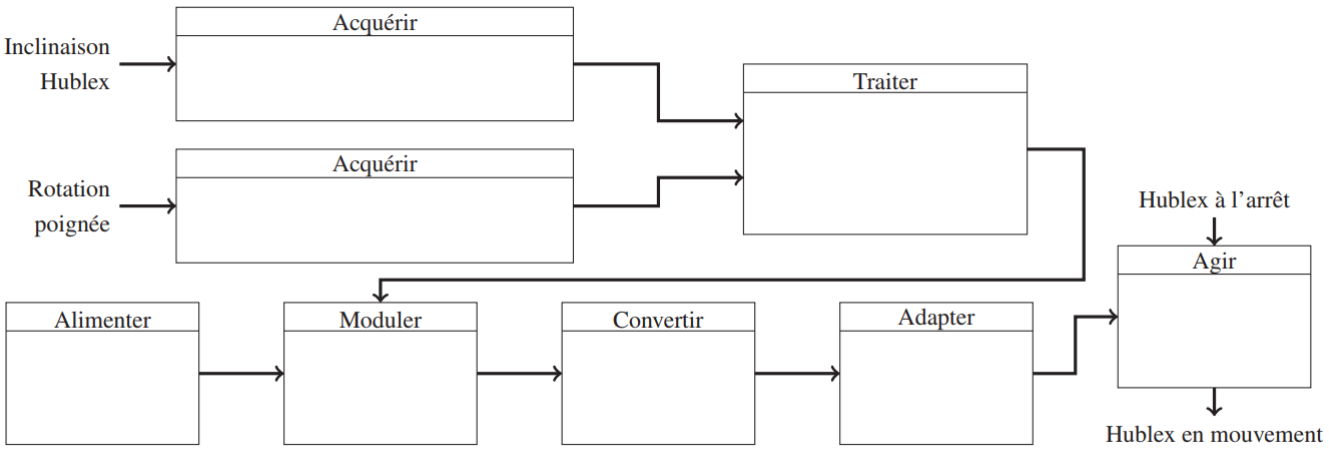
Épreuve de : SCIENCES INDUSTRIELLES

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

Gyropode à usage professionnel HUBLEX

Q1. Compléter le schéma fonctionnel du **DR1**, en précisant le nom des composants associés aux fonctions, ainsi que le type de chaque flux (I pour information, E pour énergie, M pour matière). On y reportera uniquement les composants présents dans le diagramme de bloc interne (figure 5).



Q2. Montrer que $V = r_c \cdot \omega_{10}$.

Q3. En utilisant l'hypothèse de roulement sans glissement, établir la relation entre V, ω_{10}, ω_{21} et les constantes L et R.

Q4. En déduire l'expression de la vitesse de rotation du moteur gauche ω_{41} en fonction de V, ω_{10}, du rapport de transmission k et d'autres paramètres géométriques.

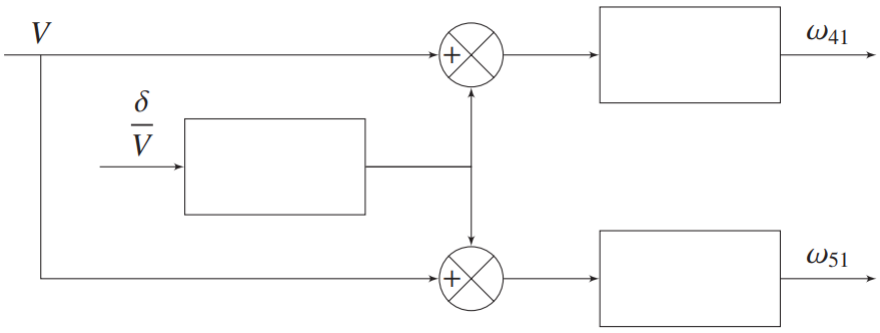
Q5. En déduire la relation entre V, ω_{10} et la vitesse de rotation du moteur droit ω_{51}.

Q6. Donner la résolution de ce capteur, c'est-à-dire sa précision angulaire.

Q7. Donner le nombre de positions effectivement mesurables avec la poignée du Hublex, ainsi que la plage des valeurs centrée autour de 0.

Q8. Établir la relation entre a_f, V et ω_{10}. En déduire la valeur maximale ω_{10max} du taux de rotation admissible satisfaisant l'exigence «1.4» et ses sous-exigences.

Q9. Compléter le schéma bloc du DR2, représentant la génération des commandes des deux moteurs à partir des consignes données par le pilote permettant de respecter l'exigence «1.1.1» notamment.



Q10. Justifier la forme de la matrice d'inertie $[I(G'_S, S')]$.

Q11. Déterminer l'expression littérale du moment cinétique au point $G_{S'}$ de S' par rapport à R_0 noté $\overrightarrow{\sigma}(G'_S, S'/R_0)$.

Q12. En déduire les expressions littérales de $\overrightarrow{\delta}(G'_S, S'/R_0)$ puis de $\overrightarrow{\delta}(G'_S, S/R_0)$.

Q13. En utilisant la dérivation vectorielle, déterminer l'expression littérale de $\overrightarrow{V}(G_S, S/R_0)$, la vitesse du centre d'inertie de S par rapport à R_0.

Q14. Calculer, au point J, la somme des moments des actions mécaniques extérieures appliquées à S selon $\overrightarrow{y_{1'}}$: $\overrightarrow{M}(J, \overline{S} \to S) \cdot \overrightarrow{y_{1'}}$.

Q15. Déduire de ce qui précède la relation liant N_I, ω_{10}, les grandeurs géométriques, cinématiques (et leurs dérivées) et inertielles. On précisera le principe ou théorème utilisé.

Q16. Dans le cadre de ce modèle, quelle est la condition permettant de définir l'apparition du basculement ? En déduire l'expression de la vitesse limite ω_{10lim} conduisant au basculement.

Q17. En exploitant cette courbe, donner la vitesse limite v_{lim} que peut avoir le Hublex dans ces conditions sans basculer. Conclure vis-à-vis des exigences.

Q18. Commenter, en justifiant, la validité des courbes de la figure 11 au-delà de v_{lim}. En une phrase, préciser comment modifier la modélisation pour étudier le comportement de S au-delà de v_{lim}.

Q19. Indiquer la démarche permettant de déterminer l'équation (3). On ne demande pas de faire les calculs.

Q20. Déterminer l'expression littérale de $(P_{ext} + P_{int})$, somme des puissances galiléennes des actions mécaniques extérieures appliquées à l'ensemble S, notée P_{ext}, et de la puissance intérieure à ce même système, notée P_{int}.

Q21. Déterminer l'expression littérale de l'énergie cinétique $E_C(S/R_0)$ de l'ensemble S par rapport au référentiel galiléen R_0, en fonction de ω_{R1} et des grandeurs inertielles et géométriques.

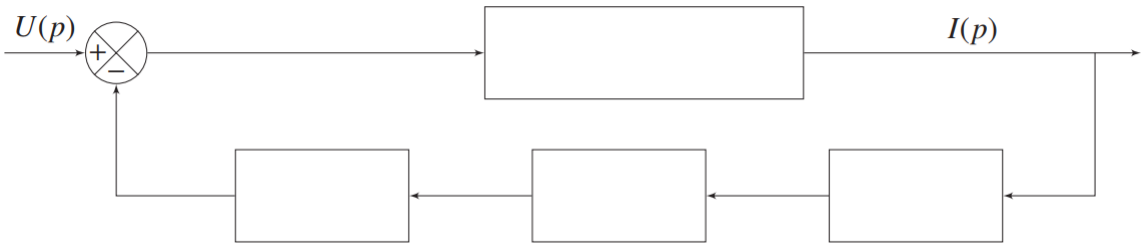
Q22. En précisant le théorème ou principe utilisé, déterminer la relation liant C_m, ω_{R1} et les grandeurs inertielles et géométriques (et leurs dérivées).

Q23. En déduire, à l'aide de l'équation (3), l'expression de $\dot{V}$ en fonction de β.

Q24. Justifier alors que la consigne β imposée par le pilote correspond à une consigne d'accélération et conclure sur le respect de l'exigence «1.1.4.1». Préciser la valeur de l'angle β pour que l'ensemble S avance à une vitesse constante.

Q25. Donner, dans le domaine de Laplace, les 4 équations caractéristiques associées au modèle de machines à courant continu.

Q26. Compléter alors le schéma bloc du moteur dans le DR3. On précisera la grandeur associée à chaque lien.



Q27. Donner l'expression de la fonction de transfert $H_m(p) = \frac{I(p)}{U(p)}$. La mettre sous la forme canonique donnée.

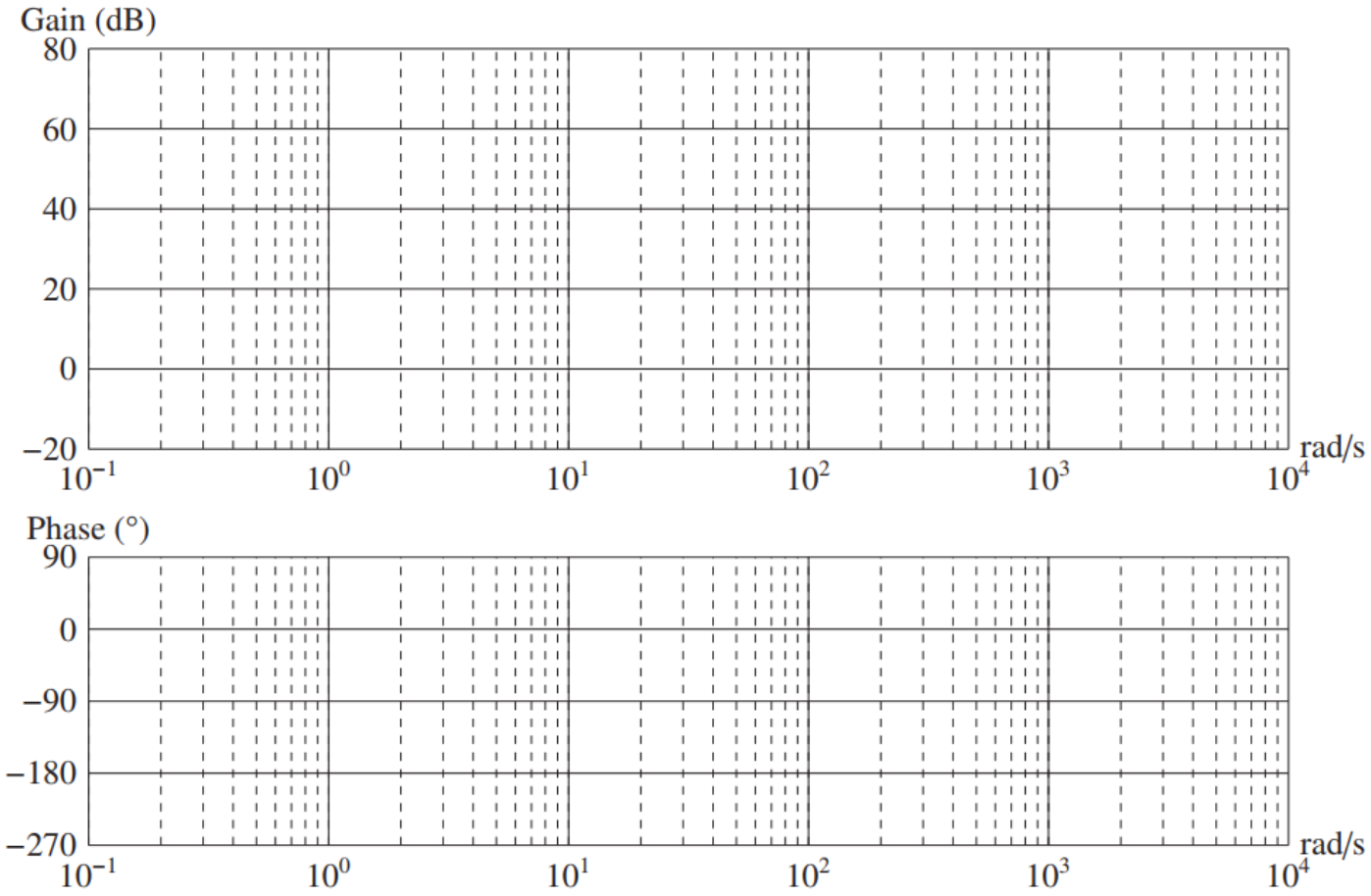
Q28. Préciser, en justifiant, quelle valeur donner à K_{iu}, caractéristique du convertisseur IU.

Q29. Déterminer l'expression littérale de l'erreur en régime permanent notée μ_s, pour une entrée indicielle (i.e. $I_c(p)$ est un échelon unitaire), en fonction de K_{iu}, K_p et K_m.

Q30. Conclure, lorsque cela est possible, quant au respect des sous-exigences de l'exigence « 1.7.1.1 » avec ce type de correcteur.

Q31. Préciser l'influence de ce correcteur sur la précision du système. Justifier le choix de ce type de correcteur dans le cas étudié.

Q32. Tracer sur le DR4 (**page suivante**), les diagrammes de Bode asymptotiques du correcteur, ainsi que l'allure des courbes réelles pour $K_p = 10$ et $K_i = 1000$. On précisera les valeurs numériques associées aux valeurs caractéristiques.



Q33. En suivant cette méthode, déterminer en justifiant la valeur numérique de K_p.

Q34. Déterminer alors la valeur numérique de K_i.

Q35. Commenter le résultat obtenu vis-à-vis de l'exigence « 1.7.1.1.4 ». Expliquer pourquoi, à partir des exigences du D6, cet asservissement n'est pas directement implanté en l'état dans le système.

Q36. Préciser quelle ultime modification a apporté le constructeur afin de respecter les exigences de l'asservissement.

Q37. Proposer en langage SQL une requête permettant d'obtenir, pour toutes les réservations valides du 7 mai 2020, leur numéro, leur heure de début et leur durée.

Q38. Proposer en langage SQL une requête permettant d'obtenir l'ensemble des informations concernant les utilisateurs ayant effectué au moins une réservation valide sur la période allant du 4 mai 2020 au 7 mai 2020.

Q39. Donner le nom de l'algorithme de tri utilisé et sa complexité moyenne. Donner également la complexité dans le pire et le meilleur des cas en précisant les propriétés que les listes doivent remplir pour être dans chacun de ces deux cas.

Q40. Écrire en langage Python une fonction `conflit2(R1,R2)` prenant en argument deux listes représentant deux réservations `R1` et `R2` et renvoyant un booléen indiquant si les deux réservations sont en conflit potentiel, c'est-à-dire que l'intersection de leurs deux créneaux horaires est non vide.

Q41. En exploitant la fonction `conflit2`, écrire en langage Python une fonction `sans_conflitL(R1,L)` prenant deux arguments : une liste `R1` représentant une réservation et une liste `L` contenant plusieurs listes (liste de listes) représentant des réservations. Cette fonction devra renvoyer un booléen indiquant si la réservation `R1` est en conflit avec au moins l'une des réservations contenues dans `L`.

Q42. Écrire en langage Python une fonction `ind_max` de complexité linéaire prenant en argument une liste et renvoyant l'indice de position de la valeur maximale.

Q43. Justifier de l'utilité de la ligne `L=charge.copy()` de la fonction `ordre_hublex`.

Q44. En exploitant les fonctions définies dans les questions précédentes, compléter, sur le DR5, en langage Python, les lignes 11 et 12 de la fonction `insertion_reservation`. Cette fonction prend en argument une liste charge contenant la charge pour chaque Hublex, une liste planning contenant le planning actuel et une liste `reserv` représentant la réservation à insérer.

- voir page suivante -

DR5 - Listing des fonctions informatiques pour la gestion du parc Hublex

Q44.

```
def tri_reservations(L):
   for i in range(len(L)):
        elem=L[i].copy()
        j=i-1
        while j>=0 and L[j][2]<elem[2]:
            L[j+1]=L[j]
            j-=1
        L[j+1]=elem.copy()

def ordre_hublex(charge):
    """renvoie une liste contenant les numéros des hublex triés en
    fonction de leur charge dans l'ordre croissant."""
    L=charge.copy()
    res=[]
    while len(res)<len(charge):
        ind=ind_max(L)
        res.insert(0,ind)
        L[ind]=-1
    return res

def insertion_reservation(charge,planning,reserv):
    """insère une tâche en fonction de la charge"""
    nbre=len(planning)
    Lj=ordre_hublex(charge)
    j=0
    flag=True
    while j<nbre and flag:
        num_hublex=Lj[j]
        possible=sans_conflitL(reserv,planning[num_hublex])
        if possible:

                                            # ligne 11 à compléter

                                            # ligne 12 à compléter

            charge[num_hublex]+=reserv[2]
        else:
            j+=1

def creation_planning(nbre,extraction):
    """renvoie une affectation des tâches en fonction de la charge
    (affecte au moins chargé)."""
    tri_reservations(extraction)
    planning_hublex=[[] for i in range(nbre)]
    charge=[0 for i in range(nbre)]
    for i in range(len(extraction)):
        insertionReservation(charge,planning_hublex,extraction[i])
    return planning_hublex,charge
```