

AFFECTATION DES TACHES				NOM	Prénom	alternant	S7	S9
4 groupes de 6 étudiants – 12 tâches par groupes - 2 tâches par personnes qui doivent appartenir à des domaines différents.								
N°		titre	description					
1	Ctrl/Cd/Acquisition	Intégration acquisition/commande carte ARDUINO (x2)	<p><b>* Intégration des codes existants :</b> Actuellement des codes arduino tournent</p> <ul style="list-style-type: none"> <li>- d'une part pour faire l'acquisition des données</li> <li>- d'autre part pour asservir la rotation du moteur.</li> </ul> <p>Il est nécessaire de faire l'intégration complète de ses deux sous système.</p> <p>Dans le meilleur des cas : Est-il possible de tout intégrer sur une seule carte Arduino ? Sinon comment gérer plusieurs cartes Arduino ? Est-ce que chaque carte doit être reliée au PC hôte par un câble USB (nécessité d'avoir 2 ports USB libre) ou peut-on relier qu'une seule carte, et faire de la communication inter-carte pour avoir un système plus simple à utiliser du point de vue utilisateur.</p> <p>Prévoir pour l'asservissement en vitesse, la possibilité de régler les gains du correcteur depuis l'IHM, et d'avoir un mode debug donnant un retour sur la qualité de l'asservissement.</p> <p><b>* Optimisation des codes existants et des protocoles :</b> Faire une analyse critique sur la façon de transférer les acquisitions sur le port série. Est-ce efficace ? Par exemple, sachant que le débit du port série est de 115200 baud, combien d'échantillon par seconde pouvons-nous attendre à pouvoir transmettre ? Quelle est la vitesse actuelle ? Réfléchir, et implémenter des modifications du code pour améliorer les performances si nécessaire.</p> <p>Vérifier que la période d'échantillonnage est stable et qu'il est possible de la choisir.</p>					
2		Intégration et optimisation acquisition/commande carte STM32 (x1)	<p>Dans cette partie il y a 2 tâches principales :</p> <p><b>* Portage des codes arduino actuels sur une cible STM32F411</b></p> <p>Prévoir pour l'asservissement en vitesse, la possibilité de régler les gains du correcteur depuis l'IHM, et d'avoir un mode debug donnant un retour sur la qualité de l'asservissement.</p> <p><b>* Optimisation des codes existants et des protocoles :</b></p> <p>Faire une analyse critique sur la façon de transférer les acquisitions sur le port série. Est-ce efficace ? Par exemple, sachant que le débit du port série est de 115200 baud, combien d'échantillon par seconde pouvons-nous attendre à pouvoir transmettre ? Quelle est la vitesse actuelle ? Réfléchir, et implémenter des modifications du code pour améliorer les performances si nécessaire.</p> <p>Vérifier que la période d'échantillonnage est stable et qu'il est possible de la choisir.</p> <p>Comparaison des performances obtenues avec les groupes arduino.</p>					
3		Intégration électrique	<ul style="list-style-type: none"> <li>- Réalisation d'une carte électronique regroupant, les cartes uP, toutes les fonctions de sécurité et un connecteur d'alimentation.</li> <li>- fixation des cartes sur le bâti (accessibilité)</li> <li>- implantation des BP de sécurité câblées (ARU, mise en marche/arrêt)</li> <li>- positionnement et fixation écran numérique</li> <li>- schéma détaillé de câblage, et câblage réel sous goulotte</li> </ul>					
4		Amélioration IHM	<p>Utilisation d'une librairie python et de bibliothèques standard (open source).</p> <p>Prévoir un mode debug permettant de régler les paramètres de l'asservissement de vitesse du moteur, et avoir un retour sur la qualité de cet asservissement.</p> <p>Prévoir la remontée d'alarmes de debug permettant d'identifier les problèmes de fonctionnement (communication, surcharge moteur, arrêt d'urgence...)</p>					

6	Méca	Amélioration de la conception CAO/réalisation de la transformation cinématique du mouvement	- critique de la solution existante - proposition d'amélioration - dessin d'ensemble de la transmission avec cotation des jeux - dessin de définition des pièces (avec masse-Inertie-cdg et tolérancement des dimensions) - nomenclature détaillée					
7		Conception CAO/réalisation globale	- Bâti – <b>fixation capteurs – fixation servo de déclenchement d'essai</b> – pupitre opérateur – placements cartes électroniques... - Dessin d'ensemble avec nomenclature détaillée					
8		Modélisation dynamique	écrire les équations du mouvement mettre sous xcos les équations électro-mécaniques du système					
9		Identification des paramètres du modèle dynamique	imaginer un processus expérimental de détermination des coefficients nécessaires au modèle dynamique le mettre en place relever les valeurs					
10	Vibration	Code numérique oscillation <b>libre</b> + validation abaqus [*]	Formation Abaqus (éléments linéaires en formulation dynamique) – écriture en python d'un script donnant la réponse du régllet soumis à un régime d'oscillations libres via la méthode de superposition modale – validation du code par confrontation avec Abaqus – intégration dans l'IHM					
11		Code numérique oscillation <b>forcée</b> + validation abaqus [*]	Formation Abaqus (éléments linéaires en formulation dynamique) – écriture en python d'un script donnant la réponse du régllet soumis à un régime d'oscillations forcées via la méthode de superposition modale – validation du code par confrontation avec Abaqus – intégration dans l'IHM					
12		Reconstruction expérimentale des déformées modales et intégration dans l'IHM. [*]	Construction de signaux théoriques à l'aide du cours – écriture d'un script en python pour déterminer la réponse en fréquence le long du régllet et validation à partir des signaux simulés – mise en œuvre expérimentale de la méthode de reconstruction des premiers modes de déformation sous oscillations libres et forcées – intégration dans l'IHM					
5		Traitement et interprétation des résultats expérimentaux	Post-traitement données capteurs pour affichage IHM (flexion libre et forcée) et comparaison avec modèle analytique (calcul manuscrit)					

[\*] Tâches 10,11 et 12 : le cours de vibration doit être terminé pour réaliser cette tâche : il faut prendre en parallèle une tâche réalisable sur P1