



**MINISTÈRES  
ÉDUCATION  
JEUNESSE  
SPORTS  
ENSEIGNEMENT  
SUPÉRIEUR  
RECHERCHE**

*Liberté  
Égalité  
Fraternité*

**Direction générale des ressources humaines**

## **RAPPORT DU JURY**

**SESSION 2025**

**Concours : Agrégation externe**

**Section : Sciences industrielles de l'ingénieur**

**Option : Sciences industrielles de l'ingénieur et ingénierie informatique**

Rapport de jury présenté par : Florian KOLBL, Maître de conférences  
Président du concours



# Table des matières

<b>1</b>	<b>Épreuves d'admissibilité</b>	<b>5</b>
1.1	Épreuve de sciences industrielles de l'ingénieur . . . . .	5
1.2	Épreuve de modélisation d'un système, d'un procédé ou d'une organisation . . . . .	17
1.3	Épreuve de conception d'un système, d'un procédé ou d'une organisation . . . . .	32
<b>2</b>	<b>Épreuves d'admission</b>	<b>51</b>
2.1	Épreuve d'exploitation pédagogique d'une activité pratique relative à l'approche globale d'un système pluritechnologique . . . . .	51
2.2	Épreuve d'exploitation pédagogique d'une activité pratique relative à l'approche spécialisée d'un système pluritechnologique . . . . .	68
2.3	Épreuve de soutenance d'un dossier industriel . . . . .	91

## Avant-propos

Le concours 2025 était la neuvième session de l'agrégation de sciences industrielles de l'ingénieur (SII) de l'option « ingénierie informatique ». Ce rapport est dans la continuité de ceux des sessions précédentes.

L'ingénierie informatique appliquée aux systèmes analyse et résout des problèmes scientifiques et technologiques communs à l'informatique générale et spécifiques à l'informatique embarquée (software et hardware). Ces problèmes s'inspirent de défis sociétaux liés à l'utilisation massive d'objets connectés (IoT), toujours plus intelligents, et à faibles empreintes énergétiques. Les spécificités et contraintes sont nombreuses et variées en fonction des applications comme le temps réel, la miniaturisation, l'autonomie et la consommation énergétiques, les interfaces homme-machine, la numérisation de grandeurs analogues à des grandeurs physiques (capteurs), le traitement de la donnée, les réseaux informatiques et de télécommunication, la cyber sécurité, la fiabilité, la sûreté de fonctionnement, etc.

Les attentes du concours de l'agrégation SII sont définies par l'arrêté du 28 décembre 2009 fixant les sections et les modalités d'organisation des concours de l'agrégation. Les concours de recrutement d'enseignants n'ont pas pour seul objectif de valider les compétences scientifiques et technologiques des candidats ; ils doivent aussi valider les compétences professionnelles qui sont souhaitées par l'État employeur qui recrute des professeurs. L'excellence scientifique et la maîtrise disciplinaire sont indispensables pour présenter le concours, mais pour le réussir, les candidats doivent aussi faire preuve de qualités didactiques et pédagogiques et de bonnes aptitudes à communiquer.

Les trois épreuves d'admissibilité sont construites de manière à évaluer un spectre large de compétences scientifiques et technologiques ; la première épreuve est commune aux quatre options de l'agrégation SII, les deux autres spécifiques à l'option. Les trois épreuves d'admission sont complémentaires des épreuves d'admissibilité ; la première épreuve d'admission est commune aux quatre options, les deux autres spécifiques à l'option. Elles permettent l'évaluation des compétences pédagogiques des futurs professeurs et s'appuient sur le référentiel des compétences professionnelles des métiers du professorat et de l'éducation (publié au BOEN du 25 juillet 2013). Elles comportent un entretien avec le jury qui permet d'évaluer la capacité du candidat à s'exprimer avec clarté et précision, à réfléchir aux enjeux scientifiques, technologiques, didactiques, épistémologiques, culturels et sociétaux que revêt l'enseignement du champ disciplinaire du concours. Ces épreuves d'admission, dont le coefficient total est le double de celui des épreuves d'admissibilité, ont eu une influence significative sur le classement final.

Les candidats et leurs formateurs sont invités à lire avec application les commentaires et conseils donnés dans ce rapport et dans ceux des sessions antérieures afin de bien appréhender les compétences ciblées. La préparation à ces épreuves commence dès l'inscription au concours. Proposer une séquence pédagogique à partir d'activités expérimentales ne s'improvise pas et nécessite une préparation rigoureuse. De même, la qualité du dossier dépend de la pertinence du choix du support. Elle impose aux futurs professeurs de s'engager, dès le début de leur carrière, dans un processus de rapprochement avec le monde de l'entreprise et de la recherche. Ces épreuves permettent « également d'évaluer la capacité du candidat à prendre en compte les acquis et les besoins des élèves, à se représenter la diversité des conditions d'exercice de son métier futur, à en connaître de façon réfléchie le contexte dans ses différentes dimensions (classe, équipe éducative, établissement, institution scolaire, société) et les valeurs qui le portent, dont celles de la République ». Les thématiques de la laïcité et de la citoyenneté trouvent toute leur place lors des entretiens avec le jury ; en effet, la mission première que fixe la Nation à ses enseignants est de transmettre et faire partager aux élèves les valeurs et principes de la République ainsi que l'ensemble des dispositions de la charte de la laïcité.

Pour cette neuvième session, 14 postes sont ouverts pour l'agrégation sciences industrielles de l'ingénieur option ingénierie informatique. Parmi les 151 inscrits, 64 candidats ont été présents aux

trois épreuves d'admissibilité. Le nombre d'inscrits est stable par rapport aux trois années précédentes ; l'édition 2022 avait connu une baisse significative du nombre d'inscrits ; une raison possible pourrait être la création de l'agrégation d'Informatique qui a pu drainer une partie des candidats.

Cette session s'est révélée être d'un très bon niveau. Les candidats retenus ont su démontrer leur maîtrise des savoirs et compétences en ingénierie et en ingénierie informatique, un sens de la pédagogie et une posture professionnelle compatibles avec l'exercice des missions d'enseignant ; le jury les en félicite. Contrairement aux sessions précédentes, les 14 postes ouverts ont été pourvus, et le candidat classé 15 a été placé sur liste d'attente, compte-tenu du niveau académique évalué.

L'agrégation est un concours prestigieux de recrutement de cadres de catégorie A de la fonction publique qui impose de la part des candidats un comportement et une présentation irréprochables. Pour conclure cet avant-propos, le jury souhaite que ce rapport soit une aide efficace aux futurs candidats. Tous sont invités à se l'approprier par une lecture attentive.

## **Remerciements**

Le lycée La Martinière Monplaisir à Lyon a accueilli les épreuves d'admission de cette session 2025 des quatre options de l'agrégation externe section sciences industrielles de l'ingénieur.

Les membres du jury tiennent à remercier le proviseur du lycée, ses collaborateurs et l'ensemble des personnels pour la qualité de leur accueil et l'aide efficace apportée tout au long de l'organisation et du déroulement de ce concours qui a eu lieu dans d'excellentes conditions. Nous remercions en particulier chaleureusement le directeur délégué aux formations du lycée pour son investissement et son dévouement pour tous les aspects logistiques et organisationnels.

## Résultats statistiques

Session	Inscrits	Nombre de postes	Présents aux trois épreuves d'admissibilité	Admissibles	Présents aux épreuves d'admission	Admis
2017	264	15	106	35	33	15
2018	280	12	105	28	23	12
2019	240	14	104	35	31	15*
2020	202	15	82	28	25	15
2021	210	15	92	36	33	15
2022	150	15	63	31	27	17**
2023	152	17	62	34	31	15
2024	164	15	71	36	30	12
<b>2025</b>	<b>151</b>	<b>14</b>	<b>64</b>	<b>36</b>	<b>33</b>	<b>15*</b>

\* un candidat a été inscrit sur liste complémentaire

\*\* deux candidats ont été inscrits sur liste complémentaire

### Statistiques des notes obtenues à l'admissibilité à la session 2025

<b>Note obtenue par le premier candidat admissible</b>	<b>17,87</b>
<b>Note obtenue par le dernier candidat admissible</b>	<b>5,20</b>
Moyenne des candidats présents	7,09
Moyenne des candidats admissibles	9,57
Écart-type des candidats admissibles	3,57

### Statistiques des notes obtenues à l'admission à la session 2025

<b>Note obtenue par le premier candidat admis</b>	<b>17,93</b>
<b>Note obtenue par le dernier candidat admis</b>	<b>10,08</b>
<b>Note obtenue par le dernier candidat sur liste complémentaire</b>	<b>9,58</b>
Moyenne des candidats présents	9,43
Moyenne des candidats admis	13,50
Écart-type des candidats admis	2,06

# Épreuve d'admissibilité de sciences industrielles de l'ingénieur

## A. Présentation de l'épreuve

Arrêté du 28 décembre 2009 modifié

- Durée totale de l'épreuve : 6 heures
- Coefficient 1

L'épreuve est commune à toutes les options. Les candidats composent sur le même sujet au titre de la même session, quelle que soit l'option choisie.

Elle a pour but de vérifier que le candidat est capable de mobiliser ses connaissances scientifiques et techniques pour conduire une analyse systémique, élaborer et exploiter les modèles de comportement permettant de quantifier les performances globales et détaillées d'un système des points de vue matière, énergie et information afin de valider tout ou partie de la réponse au besoin exprimé par un cahier des charges. Elle permet de vérifier les compétences d'un candidat à synthétiser ses connaissances pour analyser et modéliser le comportement d'un système pluritechnologique automatique.

## B. Sujet

Le sujet est disponible en téléchargement sur le site du ministère à l'adresse :

<https://www.devenirenseignant.gouv.fr/media/16117/download>

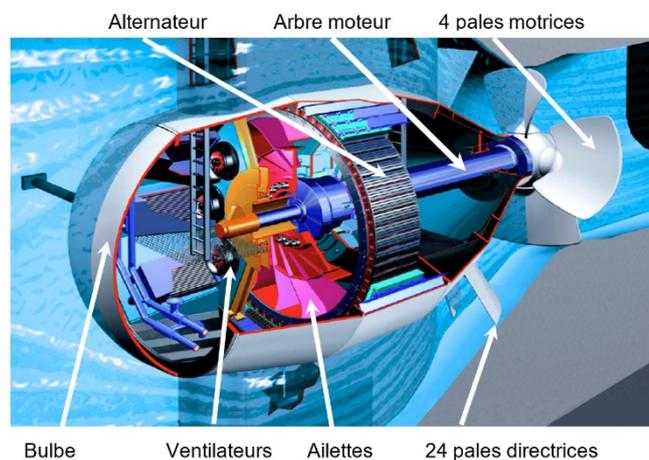
Cette épreuve porte sur la centrale hydroélectrique de Pierre-Bénite, située aux portes de Lyon, qui joue un rôle primordial pour la jonction du Rhône avec la Saône.

Elle est équipée de quatre groupes hydroélectriques à turbines installés chacun dans une conduite hydraulique, dont le débit d'eau peut être modifié en ouvrant plus ou moins la section de passage de la conduite, à l'aide de vingt-quatre pales orientables nommées « pales directrices », disposées autour de la turbine.



La CNR, qui gère les dix-huit centrales hydroélectriques du Rhône, doit assurer en temps réel un équilibre entre la puissance électrique qu'elle produit et la puissance électrique consommée par ses clients, reliés au réseau électrique européen.

Dans ce contexte, les ingénieurs de la CNR ont choisi de réguler la vitesse de rotation des turbines des groupes hydroélectriques, qui sont en prise directe avec leurs alternateurs, afin de réguler la fréquence de ces derniers à celle du réseau (50 Hz). Pour que cette régulation se fasse avec un rendement optimal, ils utilisent deux systèmes asservis qui modifient la vitesse de rotation de la turbine : l'asservissement de la position angulaire des pales directrices et l'asservissement de la position angulaire des pales motrices.



## C. Éléments de correction

### Étude de la production d'Électricité

Question 1. Afin de limiter les variations de fréquence sur le réseau les producteurs d'électricité doivent disposer de réserves de puissances.

La réserve primaire qui permet rapidement de stabiliser la fréquence en cas de dérive.

La réserve secondaire qui permet de restaurer la fréquence à sa valeur nominale et de reconstituer les réserves primaires.

La réserve tertiaire qui permet d'ajuster les consignes de production.

La limite est la valeur de la puissance de réserve disponible.

Question 2.  $\Delta f = 200 \text{ mHz}$        $K_{\min} = 5 \text{ MW.Hz}^{-1}$        $P_C = P_n = 20 \text{ MW}$   
 $f_n = 50 \text{ Hz}$        $f = 50,2 \text{ Hz}$        $P = 20 - 5 \times (50,2 - 50) = 19 \text{ MW}$

Question 3. Une baisse de soutirage électrique par rapport à l'injection provoque une augmentation de la fréquence. Une hausse du soutirage électrique par rapport à l'injection diminution de la fréquence.

Question 4.  $s = \frac{(50,2-50)/50}{(19-20)/20} = -0,08 = -8\%$ , le statisme est bien compris entre 3% et 12%.

Question 5. Avant incident :  $P_{\text{prod}} = 392,4 \text{ MW}$       Après incident : pertes de 72 MW  
 Variation de fréquence :  $\Delta f = s \cdot \frac{(P - P_n)}{P_n} \cdot f_n = -0,08 \times \frac{72}{392,4} \cdot 50 = -0,734 \text{ Hz}$       soit  $f_1 = 49,266 \text{ Hz}$

Question 6. Après réglage primaire, les groupes sont à 100% :  $P_{2\text{prod}} = 352 \text{ MW}$

$\Delta f = s \cdot \frac{(P - P_n)}{P_n} \cdot f_n = -0,08 \times \frac{(392,4-352)}{392,4} \cdot 50 = -0,412 \text{ Hz}$  soit  $f_2 = 49,588 \text{ Hz}$

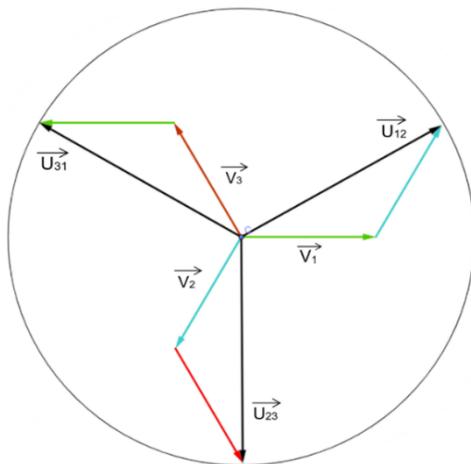
Le réglage primaire permet de ramener la fréquence des signaux électriques dans la plage de fréquence admissible par la norme soit entre  $49,5 \text{ Hz} < f < 50,5 \text{ Hz}$

Question 7. Le risque d'effondrement de la fréquence du réseau européen est faible car un incident majeur de 3000MW de pertes de production entraine une faible baisse de fréquence.

L'intermittence de certaines énergies renouvelables, comme le solaire photovoltaïque et l'éolien, a tendance à provoquer des variations aléatoires de puissance.

La croissance de ces énergies peut impacter la stabilité en fréquence du réseau en provoquant des variations significatives et aléatoires de la fréquence.

Question 8.



$$V_1(t) = \sqrt{2} \cdot V \cdot \sin(\omega \cdot t)$$

$$V_2(t) = \sqrt{2} \cdot V \cdot \sin(\omega \cdot t - 2\pi/3)$$

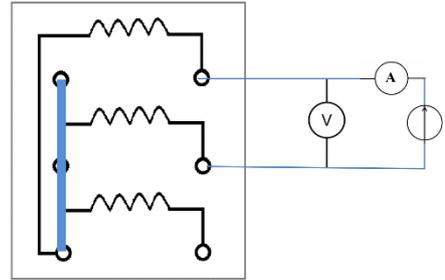
$$V_3(t) = \sqrt{2} \cdot V \cdot \sin(\omega \cdot t - 4\pi/3) = \sqrt{2} \cdot V \cdot \sin(\omega \cdot t + 2\pi/3)$$

$$U_{12}(t) = \sqrt{2} \cdot U \cdot \sin(\omega \cdot t + \pi/6)$$

$$U_{23}(t) = \sqrt{2} \cdot U \cdot \sin(\omega \cdot t - \pi/2)$$

$$U_{31}(t) = \sqrt{2} \cdot U \cdot \sin(\omega \cdot t - 7\pi/6) = \sqrt{2} \cdot U \cdot \sin(\omega \cdot t + 5\pi/6)$$

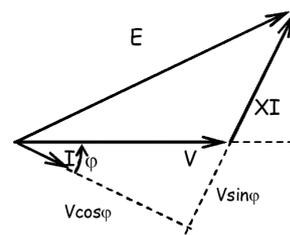
Question 9.  $R_{\text{entre phases}} = 24 \text{ m}\Omega$   $R = 12 \text{ m}\Omega$



Question 10.  $Z = E_0 / I_{CC} = 2000/1800 = 1,1 \Omega$   $X = \sqrt{1,1^2 - 0,012^2} = 1,099 \Omega \sim 1,1 \Omega$

Question 11.  $E = \sqrt{(V + XI \cdot \sin\varphi)^2 + (XI \cdot \cos\varphi)^2}$

La machine synchrone est supposée à son point de fonctionnement nominale.



Question 12.  $f_p = 0,99$   $Q_N = 2850 \text{ kVAR}$   $I_N = 3\,332 \text{ A} = 3,33 \text{ kA}$

Question 13.  $\cos\varphi=0,99$   $V = 2\,021 \text{ V}$   $E = 4\,427,6 \text{ V}$   $I_e = 664 \text{ A}$

Question 14.  $P = 3 \cdot V \cdot I \cdot \cos\varphi$  et  $E \cdot \sin \delta = XI \cos\varphi$   $\delta = \arcsin \frac{P \cdot X}{3 \cdot V \cdot E} = 55^\circ < 90^\circ$  stable

Question 15.  $\Omega_s = f / p = 83,3 \text{ tr} \cdot \text{min}^{-1}$   $f_{90\%} = p \cdot \Omega_s = 36 \times 75 / 60 = 45 \text{ Hz}$

Question 16. Le raccordement des machines synchrones sur le réseau exige les conditions suivantes sur les signaux électriques : égalité de la fréquence, égalité des tensions, même ordre de succession des phases, décalage angulaire nul.

Question 17. fonctionnement dans le quadrant B.

### Étude de la régulation de vitesse d'un groupe hydroélectrique

Question 18.

<b>30</b>	ligne = ligne.strip()
<b>31</b>	ligne = ligne.replace(',',';')

Question 19.

<b>33</b>	if not isVector(valeurs_essai):
-----------	---------------------------------

Pour cette instruction, la fonction « isVector(valeur\_essais) » permet d'identifier pour chaque ligne d'essais lue, si le deuxième élément (essais[1]) et le troisième élément (essais[2]) de la ligne sont des nombres.

Si ce n'est pas le cas, le programme est alors en train de lire une des trois lignes d'entête d'une série donnée.

Le test de la ligne 33 est vrai, et le programme passe à la ligne 34

<b>34</b>	Tester si la série courante contient des essais (sa longueur est non-nulle)
<b>35</b>	Si oui, enregistrer la série courante dans la liste de série « series_list »
<b>36</b>	Puis réinitialiser la liste courante
<b>37</b>	Tester si la première valeur de la ligne est une date (elle contient deux //)
<b>38</b>	Si oui, enregistrer la dernière valeur de cette ligne comme un élément appelé 'hauteur' dans la série courante (il s'agit de la hauteur de la série)

Question 20.

```
41 turbinage=float(valeurs_essai[2])/reference_turbinage
```

Suite à cette instruction, le programme a calculé la valeur relative de turbinage de la ligne d'essais en cours de lecture, à l'aide de la valeur de référence de turbinage donnée en début de programme à 70 (car  $70^\circ = 100\%$  de turbinage).

Question 21. L'exécution de l'instruction de la ligne 43, a permis d'enregistrer les vingt valeurs de vannage en première position de la chaîne ['essais'] de la série en cours de lecture.

```
43 serie_courante['essais'].append([vannage,turbinage])
```

```
60 for essai in series_list[0]['essais']:
```

```
61     V_vannage.append(essai[0])
```

Question 22.

```
88 def interpolation_lineaire(h):
89     i = 0 # Initialiser i à 0 (comptage de pas i des 3 valeurs du vecteur hauteurs)
90     while not(hauteurs[i] <= h and h < hauteurs[i+1]):
           # pour i=0 : tant que la hauteur h lue n'est pas dans le domaine [6 ;9.15]
           # ceci permet d'utiliser la fonction pour toute valeur de h
           # sans avoir à préciser qu'elle est forcément dans [6m ;14.8m] à la ligne 103
           # pour i=1 : tant que la hauteur h lue n'est pas dans le domaine [9.15 ;14.8]
91         i = i + 1
           # on incrémente pour aller au domaine suivant
92     alpha = (h - hauteurs_list[i]) / (hauteurs_list[i+1] - hauteurs_list[i])
           # calculer la valeur de « alpha » dans domaine en cours
93     V_turbinage_lin = V_turbinages_list[i] * (1 - alpha) + V_turbinages_list[i+1] * alpha
           # calculer la valeur de turbinage théorique par interpolation
94     return V_turbinage_lin
           # renvoyer cette valeur (qui sera utilisée à l'instruction de la ligne 105)
```

Question 23. Lorsque le vérin se déplace de 0 m à  $X_{Max} = 1$  m, les pales directrices s'inclinent de  $0^\circ$  à  $90^\circ$ , ce qu'on considère être une ouverture de conduite de 0% à  $V_{\%Max} = 100\% = 1$ .

$$\text{Alors, } K_M = \frac{V_{\%Max}}{X_{Max}} = \frac{100\%}{1m} = 1 m^{-1}$$

Question 24. La tension de mode commun n'est pas amplifiée ni supprimée.

La tension différentielle est amplifiée d'un gain  $G_d$  :  $R_g = \frac{2 \cdot R_1}{G_d - 1} = 20,02 \Omega$

$$\text{Question 25. } V_3^+(t) = \frac{R_1}{R_1 + R_1} \cdot V_{s1}(t) = \frac{V_{s1}(t)}{2} \quad V_3^-(t) = \frac{R_1}{R_1 + R_1} \cdot (V_{s2}(t) + V_s(t)) = \frac{V_{s2}(t) + V_s(t)}{2}$$

$$\text{Question 26. } V_s(t) = \frac{A_d}{2 \cdot (1 + \frac{A_d \cdot A_{MC}}{4})} \cdot (V_{s1}(t) - V_{s2}(t)) + \frac{A_{MC}}{2 \cdot (1 + \frac{A_d \cdot A_{MC}}{4})} \cdot \frac{(V_{s1}(t) + V_{s2}(t))}{2}$$

Si  $A_d \gg 1$  et  $A_d \gg A_{MC}$ , alors  $K_1 \approx 1$  et  $K_2 \approx \frac{A_{MC}}{A_d}$

$$\text{Question 27. } V_s(t) = K_1 \cdot G_d \cdot \left( V_d(t) + \frac{K_2 \cdot G_{MC}}{K_1 \cdot G_d} \cdot V_c(t) \right) \quad G_{dt} = G_d \cdot K_1 \text{ et } G_{MCT} = \frac{K_2 \cdot G_{MC}}{K_1 \cdot G_d}$$

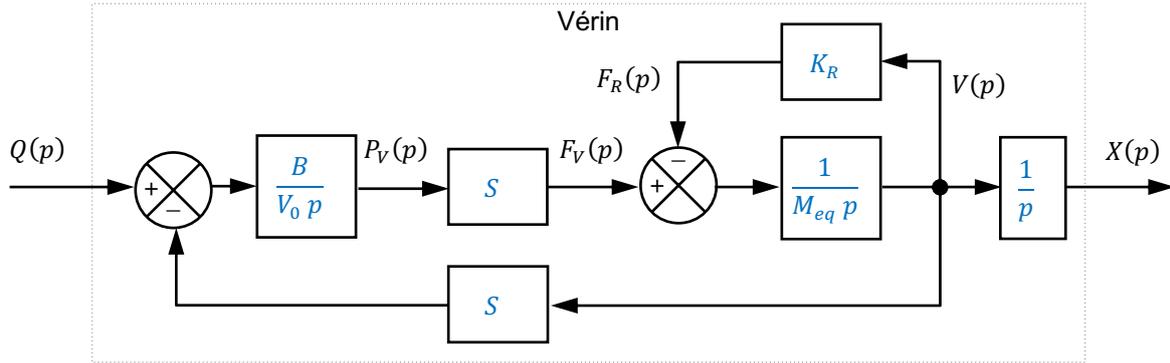
Le gain de l'étage 1 divise  $G_{MCT}$  et donc réduit l'influence du mode commun sur la tension  $V_s(t)$ . Un gain sur l'étage 2 aurait pour impact de ne pas influencer sur  $G_{MCT}$  et donc de ne pas réduire l'impact du mode commun sur  $V_s(t)$ .

Question 28. La résolution du capteur vaut :  $5 \mu m$  Donc  $K_c = \frac{1}{5 \cdot 10^{-6}} = 2 \cdot 10^5 m^{-1}$

$X_M = 1 m \Rightarrow 2 \cdot 10^5$  impulsions. Finalement, puisque  $2 \cdot 10^5 < 2^{18} \Rightarrow$  Un capteur codant au moins à 18 bits donc à 3 octets (24 bits), est donc nécessaire.

Question 29.  $01d4c0_{(H)} = 120000_{(D)}$ , correspond à une position  $X(p) = \frac{1,2 \cdot 10^5}{2 \cdot 10^5} X_M = 600 mm$

Question 30.



$$H_V(p) = \frac{1}{p} \frac{\frac{1}{S}}{1 + \frac{V_0 K_R}{BS^2} p + \frac{V_0 M_{eq}}{BS^2} p^2} = \frac{K_V}{p} \frac{1}{1 + \frac{2z}{\omega_0} p + \frac{1}{\omega_0^2} p^2}$$

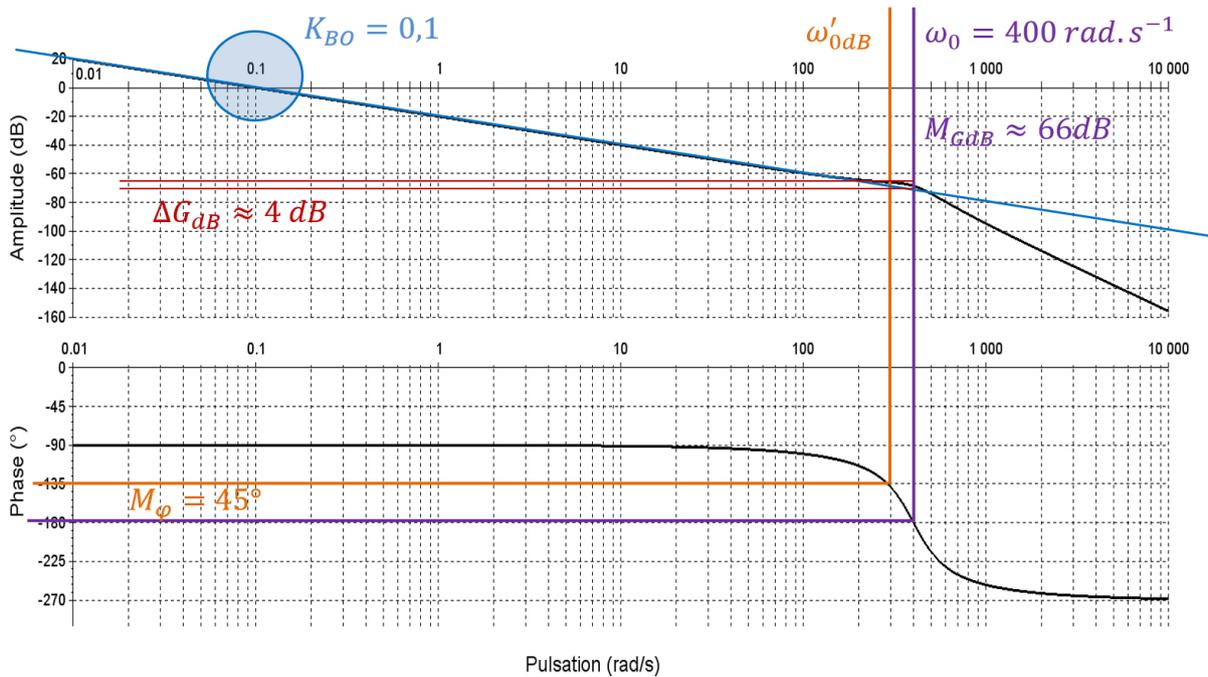
Cette fonction de transfert est de classe 1 et d'ordre 3, avec :

- Le gain :  $K_V = \frac{1}{S} = \frac{1}{0.08 \text{ m}^2} = 12.5 \text{ m}^{-2}$
- La pulsation propre :  $\omega_0 = \sqrt{\frac{BS^2}{V_0 M_{eq}}} = 400 \text{ rad} \cdot \text{s}^{-1}$
- Le coefficient d'amortissement :  $z = \frac{V_0 K_R}{2BS^2} \sqrt{\frac{BS^2}{V_0 M_{eq}}} = \frac{1}{3}$

Question 31.  $Q_{max} \geq S \cdot V_{max} = 72 \text{ L} \cdot \text{min}^{-1}$        $K_D = \frac{80 \text{ L} \cdot \text{min}^{-1}}{10 \text{ Volt}} = \frac{4}{3} \cdot 10^{-4} \text{ m}^3 \cdot \text{s}^{-1} \cdot \text{V}^{-1}$

Question 32.  $H_A(p) = \frac{1}{H_M(p)} = \frac{1}{K_M} = 1 \text{ m}$        $K_A = K_C$

Question 33.



Graphiquement, nous identifions une FTBO de classe  $\alpha = 1$  et d'ordre  $n = 3$  avec  $\omega_{0db} = 0,1 \text{ rad} \cdot \text{s}^{-1}$  et  $K_{BO} = 0,1 \text{ s}^{-1}$

Question 34.

Exigence de stabilité : voir les tracés en orange

Les exigences de stabilité sont respectées. Pour garantir une marge de phase minimale de  $45^\circ$ , il est possible d'augmenter le gain  $K_p$  du correcteur de sorte que la nouvelle pulsation de bande passante atteigne  $\omega'_{0dB} \leq 300 \text{ rad} \cdot \text{s}^{-1}$

Si le gain est relevé de  $66\text{dB}$ ,  $\omega'_{0dB} = 300 \text{ rad} \cdot \text{s}^{-1}$ , mais le système ne respecte plus l'exigence de marge de gain. C'est donc cette dernière qui impose la valeur maximale du gain  $K_p$  du correcteur, qui doit alors relever le gain de  $46\text{dB}$  seulement.

$$\Rightarrow 20 \log(K_{pmax}) \leq 46\text{dB} \Rightarrow K_{pmax} \leq 10^{\frac{46}{20}} \approx 200 \text{ (adimensionnel)}$$

Exigence de précision :

Le système étudié est à retour unitaire, avec une FTBO  $H_{BO}(p)$  de classe  $\alpha = 1$ .

Ainsi, le théorème de la valeur finale permet de prouver que :

- L'erreur statique de position en réponse à un échelon sera nulle.
- L'erreur statique de trainage en réponse à une rampe de  $15\text{mm} \cdot \text{s}^{-1}$  sera égale à :

$$e_{s-t} = \frac{15}{K_{BO}} = \frac{15}{0,1 K_p} \text{ mm}$$

Alors, pour respecter le critère de précision en réponse à une rampe, il faut régler le gain  $K_p$  du correcteur à la valeur minimale :

$$K_{pmin} \geq \frac{15}{0,1 e_{s-t}} = \frac{15 \text{ mm}}{0,1 \cdot 1 \text{ mm}} = 150 \text{ (adimensionnel)}$$

Conclusion : un simple correcteur proportionnel permet de respecter les critères de stabilité et de précision imposés par le cahier des charges. Mais le respect du critère de rapidité n'a pas été vérifié.

Question 35. La réponse à la question précédente a prouvé qu'il est possible de régler un correcteur proportionnel qui permette de respecter les exigences de stabilité et de précision du système d'asservissement de vannage.

Mais même si la conduite est ouverte de façon parfaitement précise, la vitesse de rotation de la turbine qui est prédite en conséquence, reste le résultat d'un calcul théorique (qui dépend de la dynamique des fluides, de l'asservissement de turbinage et de la hauteur de chute qui agit comme une perturbation) et il est possible qu'il y ait un écart avec la vitesse réelle de rotation de la turbine.

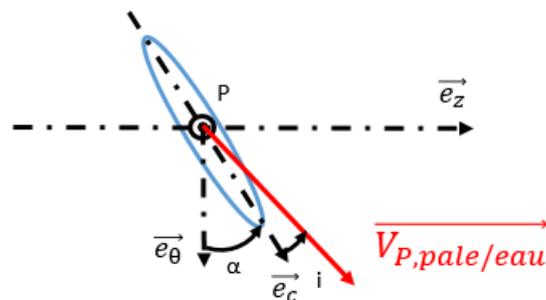
Un écart entre la fréquence du réseau à 50 Hz et la fréquence de l'électricité produite par l'alternateur, reste possible, malgré la précision de l'asservissement de vannage. Le correcteur PID sert à réguler la fréquence du groupe hydroélectrique de façon indépendante de la précision du vannage ou du turbinage.

### Étude dynamique de la turbine

Question 36.  $\vec{V}_{P,pale/eau} = r \cdot \omega \vec{e}_\theta + V_e \vec{e}_z$

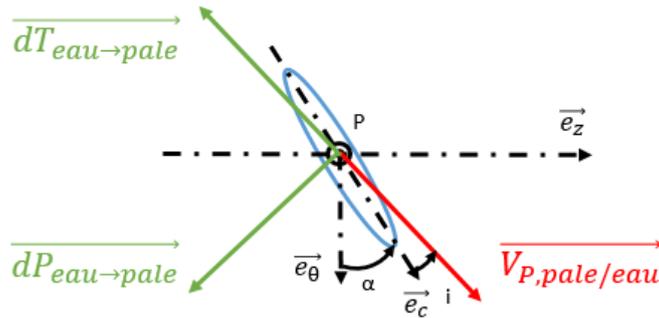
A.N. :  $\vec{V}_{P,pale/eau} = 26,5 \vec{e}_\theta + 25 \vec{e}_z$

Question 37.



Question 38.  $\tan(\alpha + \beta) = \frac{V_e}{r \cdot \omega}$

Question 39.



Question 40.  $\overrightarrow{dM}_{O,eau \rightarrow pale} \cdot \vec{z}_0 = r \cdot \left| \overrightarrow{dP}_{eau \rightarrow pale} \right| \cdot \sin(\alpha + \beta) - r \cdot \left| \overrightarrow{dT}_{eau \rightarrow pale} \right| \cdot \cos(\alpha + \beta)$

Question 41. Pour conserver un angle d'attaque constant sur toute la longueur de la pale, il est possible de modifier l'angle de calage en vrillant la pale sur sa longueur, ce que l'on observe sur le document DT5.

Question 42.  $\Gamma_{turbine/bulbe} = -\frac{\Omega_{emb}}{t_f} = -\frac{314 \times 2\pi}{60 \times 180} = -0,183 \text{ rad} \cdot \text{s}^{-2}$

Question 43. Théorème de l'énergie cinétique à l'ensemble en mouvement ou théorème du moment dynamique en projection sur l'axe de rotation :

$$C_T - C_a - C_f = J_{eq} \cdot \Gamma_{turbine/bulbe} \quad C_f = 86,6 \cdot 10^4 \text{ N} \cdot \text{m}$$

Question 44.  $\left\{ \tau_{pistons \rightarrow disque} \right\}_A = \begin{Bmatrix} 0 \\ 0 \\ 2fP_{ap} \frac{\pi d_p^2}{4} \end{Bmatrix} \left| \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \right\}$  dans la base  $(\vec{i}, \vec{j}, \vec{k})$

Question 45.  $P_{ap} = \frac{-J_{eq} \Gamma_{turbine/bulbe} + C_T - C_a}{2ef\pi d_p^2}$

Question 46.  $P_{ap} = \frac{86,6 \cdot 10^4}{2 \times 2 \times 0,5 \times \pi \times 0,8^2} = 2,15 \text{ bars} < 7 \text{ bars}$ . L'exigence est respectée.

Question 47. La charge, la vitesse de rotation, la viscosité du lubrifiant et donc la température, la rugosité, la présence d'aspérité ou de corps étrangers, ...

Question 48. Il faut que le centre de gravité  $G_1$  soit équidistant des centres des paliers A et B.

Question 49.  $p = \frac{Mg}{2(2rl)} = \frac{1,58 \cdot 10^5 \cdot 9,81}{2(2 \cdot 0,45 \cdot 0,9)} = 0,96 \text{ MPa} < 10 \text{ MPa}$

La marge est importante pour tenir compte des phases de démarrage avant que le film d'huile soit suffisant.

Question 50.  $\frac{2\pi Q_f}{a r l \omega} = 0,7 \quad Q_f = 0,7 \frac{4 \cdot 10^{-4} \cdot 0,45 \cdot 0,9 \cdot 83 \frac{\pi}{30}}{2\pi} = 1,6 \cdot 10^{-4} \text{ m}^3 \cdot \text{s}^{-1} = 0,16 \text{ dm}^3 \cdot \text{s}^{-1}$

Le débit prévu étant 8 fois supérieur au débit de fuite, la marge est suffisante pour garantir le bon fonctionnement du palier vis-à-vis de ce critère.

Question 51. La forme de la matrice est justifiée par :

- Une géométrie qui présente un axe de révolution,
- Un point d'expression de la matrice -  $G_1$  - qui appartient à l'axe de révolution,
- Une base de calcul  $(\vec{x}, \vec{y}, \vec{z})$  qui est principale d'inertie

La condition d'équilibrage statique est vérifiée car le centre d'inertie  $G_1$  appartient à l'axe de rotation.  
 La condition d'équilibrage dynamique n'est pas vérifiée car l'axe de rotation n'est pas principal d'inertie (absence de plan de symétrie de normale  $\vec{z}_0$ ) : la rotation du rotor va donc engendrer des vibrations dans les paliers.

Question 52. 
$$\overrightarrow{\sigma(G_1, 1/0)} = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_1 & 0 \\ 0 & 0 & C_1 \end{bmatrix}_R \cdot \begin{bmatrix} -\varepsilon \cdot \dot{\theta} \\ 0 \\ \dot{\theta} \end{bmatrix} = -A_1 \cdot \varepsilon \cdot \dot{\theta} \cdot \vec{x} + C_1 \cdot \dot{\theta} \cdot \vec{z}$$

$$\overrightarrow{\delta(G_1, 1/0)} = \frac{d\overrightarrow{\sigma(G_1, 1/0)}}{dt} /_{R_0} = (-A_1 \cdot \dot{\theta}^2 \cdot \varepsilon \cdot \cos \varepsilon + C_1 \cdot \dot{\theta}^2 \cdot \sin \varepsilon) \cdot \vec{y} = (C_1 - A_1) \cdot \varepsilon \cdot \dot{\theta}^2 \cdot \vec{y} \quad (dl1)$$

Question 53. Isolons  $\{1\}$  pour écrire les équations issues du PFD avec  $\overrightarrow{\Gamma(G_1, 1/0)} = \vec{0}$  :

$$TRD / \vec{x}_1 : X_{21} + X_{31} - m_1 \cdot g \cdot \sin \theta = 0$$

$$TRD / \vec{y}_1 : Y_{21} + Y_{31} - m_1 \cdot g \cdot \cos \theta = 0$$

$$TMD \text{ en } G_1 / \vec{x}_1 : L \cdot Y_{21} - L \cdot Y_{31} = 0$$

$$TMD \text{ en } G_1 / \vec{y}_1 : -L \cdot X_{21} + L \cdot X_{31} = (C_1 - A_1) \cdot \varepsilon \cdot \dot{\theta}^2$$

La résolution de ce système de 4 équations à 4 inconnues donne :

$$Y_{21} = Y_{31} = \frac{m_1 \cdot g \cdot \cos \theta}{2}$$

$$X_{21} = \frac{m_1 \cdot g \cdot \sin \theta}{2} - \frac{(C_1 - A_1) \cdot \varepsilon \cdot \dot{\theta}^2}{2L}$$

$$X_{31} = \frac{m_1 \cdot g \cdot \sin \theta}{2} + \frac{(C_1 - A_1) \cdot \varepsilon \cdot \dot{\theta}^2}{2L}$$

Et donc :

$$\begin{aligned} \overrightarrow{R_A} &= X_{21} \cdot \vec{x}_1 + Y_{21} \cdot \vec{y}_1 \\ &= \left( \frac{m_1 \cdot g \cdot \sin \theta}{2} - \frac{(C_1 - A_1) \cdot \varepsilon \cdot \dot{\theta}^2}{2L} \right) \cdot \vec{x}_1 + \frac{m_1 \cdot g \cdot \cos \theta}{2} \cdot \vec{y}_1 \\ &= \frac{(C_1 - A_1) \cdot \varepsilon \cdot \dot{\theta}^2}{2L} \cdot \vec{y}_1 + \frac{m_1 \cdot g}{2} (\sin \theta \cdot \vec{x}_1 + \cos \theta \cdot \vec{y}_1) \\ &= \frac{(C_1 - A_1) \cdot \varepsilon \cdot \dot{\theta}^2}{2L} \cdot \vec{y}_1 + \frac{m_1 \cdot g}{2} \cdot \vec{y}_0 \end{aligned}$$

$$\overrightarrow{R_A} = F_0 \cdot \vec{y}_0 + F_1 \cdot \vec{y}_1 \quad \text{avec} \quad F_0 = \frac{m_1 \cdot g}{2} \quad \text{et} \quad F_1 = \frac{(C_1 - A_1) \cdot \varepsilon \cdot \dot{\theta}^2}{2L}$$

Question 54.  $F_1 = F_0 \Leftrightarrow \frac{|(C_1 - A_1) \cdot \varepsilon| \cdot \dot{\theta}^2}{2L} = \frac{m_1 \cdot g}{2} \quad \omega_{lim} = \sqrt{\frac{m_1 \cdot g \cdot L}{|(C_1 - A_1) \cdot \varepsilon|}}$

Question 55.

Représentation de $-\vec{R}_A$ dans la base $(\vec{x}_0, \vec{y}_0)$	
Pour $\dot{\theta} < \omega_{lim}$	Pour $\dot{\theta} > \omega_{lim}$
<p><b>Conclusion :</b> La zone où la charge s'applique sur le palier est la circonférence entre les points M0 et M1.</p>	<p><b>Conclusion :</b> La zone où la charge s'applique sur le palier couvre l'intégralité de la circonférence du palier.</p>
Représentation de $+\vec{R}_A$ dans la base $(\vec{x}_1, \vec{y}_1)$	
Pour $\dot{\theta} < \omega_{lim}$	Pour $\dot{\theta} > \omega_{lim}$
<p><b>Conclusion :</b> La zone où la charge s'applique sur le rotor couvre l'intégralité de la circonférence du rotor.</p>	<p><b>Conclusion :</b> La zone où la charge s'applique sur le rotor est la circonférence entre les points M'0 et M'1.</p>

Question 56. En phase de fonctionnement normal, la vitesse de rotation étant suffisamment inférieure à la vitesse limite, si le rotor était amené à se déformer alors la zone du palier où la charge s'appliquerait resterait limitée à celle prévue (à l'opposé des orifices d'injection du lubrifiant). Par contre, en phase de délestage lorsque la vitesse de rotation va augmenter, celle-ci intervenant au carré dans l'expression de  $F_1$ , la zone du palier où la charge va s'appliquer va croître très rapidement entraînant un risque de portance insuffisante et donc une usure prématurée du palier.

### Étude des moyens de maintenance des Éléments d'un groupe hydroélectrique

Question 57.

- Longueur  $\gg$  devant la section ;
- Matériau linéaire homogène et isotrope ;
- Les déformations sont petites ;
- Hypothèses de Navier-Bernoulli : les sections planes et perpendiculaires à l'axe de la poutre avant déformation restent planes et perpendiculaires à l'axe neutre après déformation ;
- Hypothèse de Saint Venant : les résultats de la RDM ne s'appliquent valablement qu'à une distance suffisamment éloignée de la région d'application des forces.

Question 58. Le poids propre de la poutre est modélisé par une charge uniformément répartie. La charge utile du pont roulant est modélisée par une charge ponctuelle. Le modèle possède 3 inconnues de liaisons que l'on peut déterminer grâce aux 3 équations du PFS dans le plan, le système est donc isostatique.

Question 59.  $q = \rho \cdot g \cdot S = 56 \text{ kN.m}^{-1}$        $F_C = 600 \times 10 / 2 = 3000 \text{ kN}$

Question 60.  $\{\tau_{coh}\} = \begin{pmatrix} N(x) & 0 \\ V(x) & 0 \\ 0 & M(x) \end{pmatrix}$

$N(x)$  = effort normal

$V(x)$  = effort tranchant

$M(x)$  = moment fléchissant

$0 < x < 5,8\text{m}$

$N(x) = 0$

$V(x) = q \cdot x - R_{yA}$

$M(x) = -q \cdot x^2/2 + R_{yA} \cdot x$

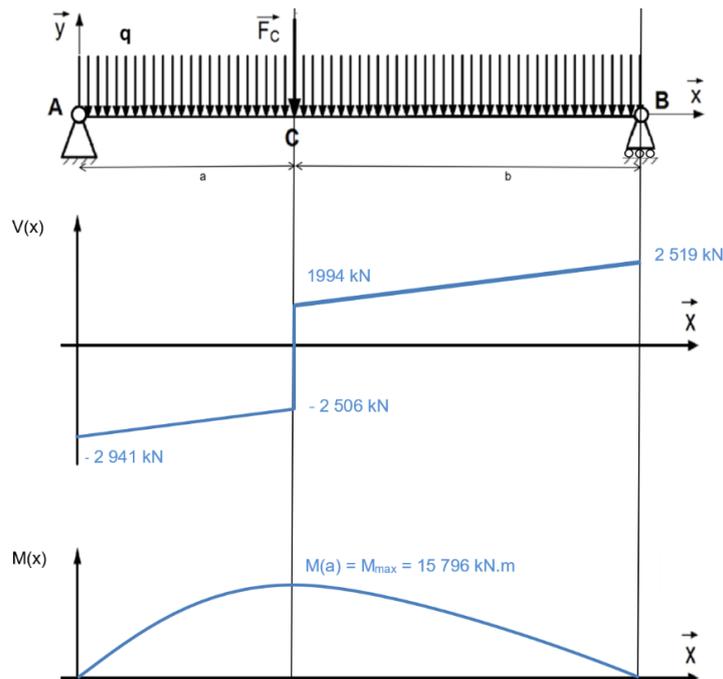
$5,8\text{m} < x < 12,8\text{m}$

$N(x) = 0$

$V(x) = F_C + q \cdot x - R_{yA}$

$M(x) = -F_C \cdot (x - 5,8) - q \cdot x^2/2 + R_{yA} \cdot x$

Question 61.



Question 62.  $I = 0,2462 \text{ m}^4$                        $\sigma_{\max} = 57,7 \text{ Mpa}$                        $5 \sigma_{\max} = 288,5 \text{ Mpa} < 345 \text{ Mpa}$

Question 63. E.I .  $f_r(x) = q.L.x^3/12 - q.x^4/24 - q.L^3.x/24$                        $f_r(x) = \frac{q.(2L.x^3 - x^4 - L^3.x)}{24.E.I}$

Question 64.  $f_{r \text{ MAX}} = -\frac{5 q .L^4}{384 .E.I} = - 0,53 \text{ mm}$

Flèche maximale admissible  $L/400 = 12,8 /400 = 32 \text{ mm}$ , l'exigence est bien respectée.

Question 65. La valeur maximale de la déformée est observée au point d'abscisse  $x = 6,24 \text{ m}$ .

Question 66.  $f_r(6,24) = - 0,32 \text{ mm}$                        $f_p(6,24) = - 3,95 \text{ mm}$

Flèche maximale :                       $\delta_{\max} = f_r(6,24) + f_p(6,24) = - 0,32 - 3,95 = - 4,27 \text{ mm}$

Flèche maximale admissible :  $f_{\max} = 12,8/750 = 17,06 \text{ mm}$

$\delta_{\max} < f_{\max}$  La poutre est correctement dimensionnée vis-à-vis de la flèche.

## D. Commentaires du jury

Le jury constate que la grande majorité des candidats concentre leurs efforts sur les questions proches de leur spécialité ingénierie électrique, certains ne traitant aucune question mobilisant d'autres compétences. Cette stratégie ne permettant pas d'aboutir à une note élevée, il est important que les candidats préparent cette épreuve avec une approche transversale.

De plus, les lauréats du concours de l'agrégation seront appelés à intervenir dans les enseignements de sciences pour l'ingénieur en lycée pré-baccalauréat (filières STI2D et SSI) et post-baccalauréat (Brevet de technicien supérieur et classe préparatoire aux grandes écoles). Les professeurs agrégés peuvent également être affectés dans des établissements tels que les IUT, les universités ou les écoles d'ingénieurs. Un travail de préparation sur les différentes options est donc fondamental en vue d'une potentielle affectation hors de son domaine de spécialité.

Le barème de cette épreuve n'évaluant pas exclusivement les résultats finaux, les candidats doivent accorder davantage d'importance à leurs justifications en proposant une rédaction qui permette aux correcteurs de comprendre la démarche suivie et les hypothèses formulées.

De plus, même si l'évaluation ne tient pas compte de la qualité de l'écriture des candidats, le jury insiste sur le fait qu'un futur enseignant se doit de fournir des documents lisibles, tant du point de vue de la graphie que de l'orthographe et de la grammaire.

### Étude de la production d'électricité

Cette première partie est la mieux traitée par les candidats, les questions étant très proches de leur domaine d'expertise. Cependant, le jury a trop souvent observé un manque de rigueur avec le formalisme, notamment avec la manipulation des complexes.

De plus, lorsque le candidat obtient une application numérique aberrante, il est attendu de celui-ci qu'il le mentionne dans sa copie. Les ordres de grandeur doivent être maîtrisés.

### Étude de la régulation

Pour les premières questions de cette partie, qui portaient sur la programmation en langage Python, les réponses des candidats sont souvent approximatives, tant dans l'interprétation d'un code que dans l'écriture de celui-ci. Globalement, ces questions n'ont été traitées que par la moitié des candidats.

Les questions suivantes, qui portaient sur l'électronique et l'automatique linéaire, sont largement traitées par les candidats, mais le jury observe que trop de candidats semblent traiter les questions indépendamment les unes des autres, sans bien comprendre leur enchaînement. Chaque partie du sujet répond à un objectif, et le candidat doit donc inscrire sa réflexion dans celle-ci. Ainsi, les conclusions formulées doivent nourrir la réponse à cette problématique.

### Étude dynamique de la turbine

Que ce soit pour le calcul d'un vecteur vitesse, ou l'écriture d'un torseur d'action mécanique, ou l'application du principe fondamental de la dynamique, le jury observe là-aussi une rigueur insuffisante,

notamment avec le formalisme vectoriel. Les candidats doivent être plus exigeants dans l'écriture mathématique de leurs calculs.

De plus, les phénomènes physiques propres à la mécanique comme les notions de portance et de trainée, de pression de contact, de frottement ou d'équilibrage ne sont pas suffisamment maîtrisés. Le jury note que la moitié des candidats n'a pas du tout traité cette partie.

#### Étude des moyens de maintenance

Dans cette partie qui cherchait à dimensionner une poutre, trop peu de candidats ont su déterminer les actions mécaniques extérieures aux appuis. Dans ce contexte, l'étude de résistance des matériaux ne peut être conduite convenablement.

Le jury rappelle aux candidats l'importance de vérifier l'homogénéité de leurs résultats, notamment pour piéger d'éventuelles erreurs d'inattention.

Enfin, trop de candidats ont réalisé de longs calculs sans les mener à leur terme.

# Épreuve d'admissibilité de modélisation d'un système, d'un procédé ou d'une organisation

## A Présentation de l'épreuve

Arrêté du 28 décembre 2009 modifié

- Durée totale de l'épreuve : 6 heures
- Coefficient 1

L'épreuve est spécifique à l'option choisie.

À partir d'un dossier technique comportant les éléments nécessaires à l'étude, l'épreuve a pour objectif de vérifier que le candidat est capable de synthétiser ses connaissances pour modéliser un système technique dans le domaine de la spécialité du concours dans l'option choisie en vue de prédire ou de vérifier son comportement et ses performances.

## B Sujet

Le sujet est disponible en téléchargement sur le site du ministère à l'adresse :

<https://www.devenirenseignant.gouv.fr/media/15951/download>

## C Éléments de correction

### C.1 Modélisation dysfonctionnelle

Question 1 :

$$\begin{cases} x_1(t+dt) = x_1(t) - \underbrace{\lambda dt x_1(t)}_{\text{vers } x_2} - \underbrace{\lambda dt x_1(t)}_{\text{vers } x_3} \\ x_2(t+dt) = x_2(t) - \underbrace{\lambda dt x_2(t)}_{\text{vers } x_4} + \underbrace{\lambda dt x_1(t)}_{\text{depuis } x_1} \\ x_3(t+dt) = x_3(t) - \lambda dt x_3(t) + \lambda dt x_1(t) \\ x_4(t+dt) = x_4(t) + \lambda dt x_2(t) + \lambda dt x_3(t) \end{cases}$$

$$(x_1(t+dt), x_2(t+dt), x_3(t+dt), x_4(t+dt)) = (x_1(t), x_2(t), x_3(t), x_4(t)) \times \begin{pmatrix} 1 - 2\lambda dt & \lambda dt & \lambda dt & 0 \\ 0 & 1 - \lambda dt & 0 & \lambda dt \\ 0 & 0 & 1 - \lambda dt & \lambda dt \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$X(t+dt) = X(t) \times \begin{pmatrix} 1 - 2\lambda dt & \lambda dt & \lambda dt & 0 \\ 0 & 1 - \lambda dt & 0 & \lambda dt \\ 0 & 0 & 1 - \lambda dt & \lambda dt \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Question 2 :  $X_n = X_0 P^n$

Question 3 : On a  $X_n = X_0 P^n = X_0 R D^n R^{-1}$ . Mais seule la valeur sur la dernière composante nous intéresse, et  $X_0$  est nul sauf sur la première composante : il est donc possible d'éviter beaucoup de calculs.

$$X_0 R = (1 \quad 1 \quad 0 \quad -1)$$

$$D^n = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & (1-\lambda)^n & 0 & 0 \\ 0 & 0 & (1-\lambda)^n & 0 \\ 0 & 0 & 0 & (1-2\lambda)^n \end{pmatrix} \Rightarrow D^n R^{-1} = \begin{pmatrix} / & / & / & 1 \\ / & / & / & -2(1-\lambda)^n \\ / & / & / & -(1-\lambda)^n \\ / & / & / & -(1-2\lambda)^n \end{pmatrix}$$

$$X_n = X_0 P^n = X_0 R D^n R^{-1} = (/ \ / \ / \ 1 - 2(1-\lambda)^n + (1-2\lambda)^n)$$

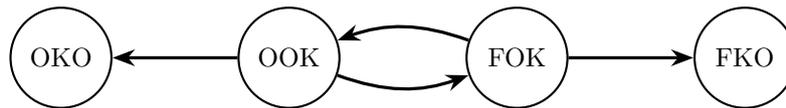
Donc  $x_4(t_n) = 1 - 2(1-\lambda)^n + (1-2\lambda)^n$ .

**Question 4 :** Il faut déterminer la proportion des systèmes tombant en panne à la date  $t$  en dérivant  $x_4(t)$ , puis calculer la date moyenne de défaillance en intégrant  $t \times \frac{dx_4}{dt}$  à l'infini :

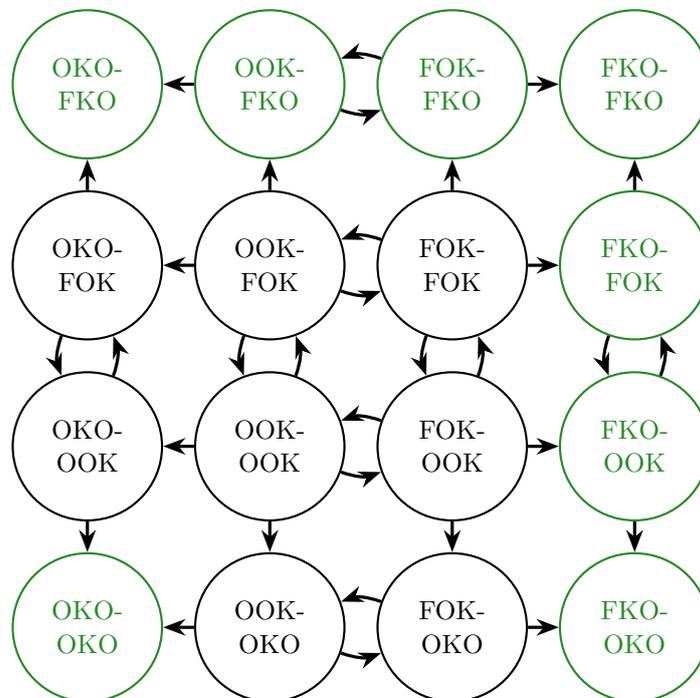
$$\text{MTTF} = \int_{t=0}^{+\infty} t \times \frac{dx_4}{dt}(t)$$

**Question 5 :** La propriété de Markov n'est pas respectée : la commande d'ouverture ou de fermeture (et surtout leurs dates) provient de l'extérieur à l'état.

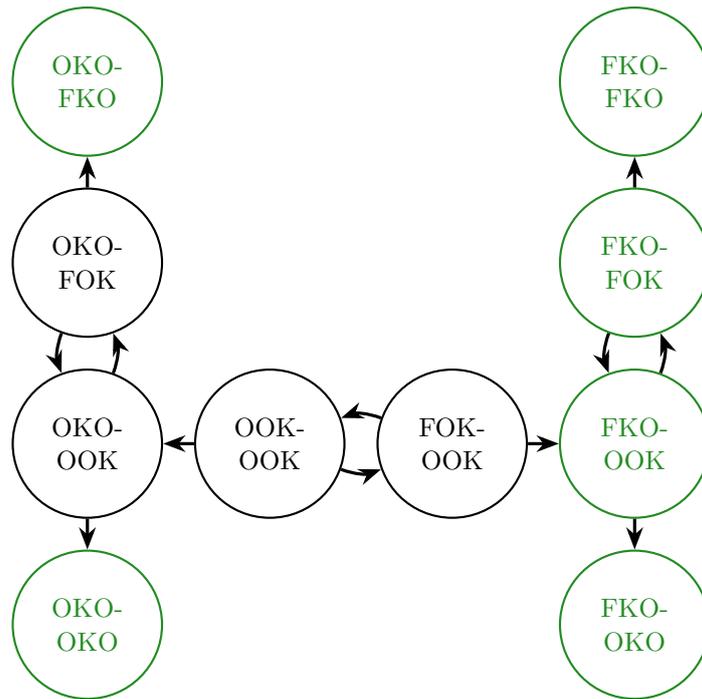
**Question 6 :**



**Question 7 :**



**Question 8 :** A partir de la partie précédente, on peut éliminer les sommets pour lesquels le composant 2 (redundant) est KO mais pas le composant 1, ainsi que les deux sommes où les deux sont en bon fonctionnement mais le composant 2 est fermé (puisqu'il est redondant froid, il doit être ouvert).



**Question 9 :** Si l'on fait l'hypothèse que chaque composant est modélisé par un graphe à 2 sommets, on obtient  $2^{1000} \approx 10^{300}$  sommets. Si l'on voulait stocker en mémoire ce graphe, à l'aide d'un octet par sommet (ce qui est déjà trop peu), il faudrait  $1 \times 10^{289}$  To. Autre point de vue, c'est  $10^{220}$  fois plus que le nombre d'atomes dans l'univers connu. Il va donc être impossible de construire ce graphe, et il est encore moins raisonnable d'espérer le parcourir pour en calculer une disponibilité. Une autre difficulté va être de déterminer les sommets critiques. Enfin, la détermination de la probabilité d'atteindre un état critique sera impossible (inversion d'une matrice  $2^{1000} \times 2^{1000}$ ).

## C.2 Transmission des informations

**Question 10 :** Le signal est lu en mesurant la différence de tension entre les deux fils, afin de s'affranchir de l'altération du signal par la superposition d'un parasite : le même parasite apparaît nécessairement sur les deux lignes à la fois car elles se situent dans le même environnement. C'est la différence des signaux qui est prise en compte, donc le résultat n'est pas altéré. L'utilisation de paires torsadées permet de s'affranchir des émissions du câble en inversant à chaque torsade les vecteurs induction produits, ce qui les annule deux à deux. Elle permet également de réduire les perturbations extérieures en annulant par opposition les courants induits dans chaque boucle par un champ perturbateur.

**Question 11 :** Une trame CAN comporte un champ de CRC, sur 16 bits, qui a un rôle de somme de contrôle : cela permet de détecter une trame qui comporte une ou plusieurs erreurs, et donc de les ignorer, voire de corriger les erreurs.

**Question 12 :** Pour 2 octets de données utiles, une trame CAN a une taille de  $1+12+6+16+16+2+7+3$  63 bits. Le bus doit donc transmettre  $\frac{63 \times 15}{4 \times 10^{-2}} = 2,3625 \times 10^5$  bits/s  $\approx 24$  kbits/s, soit une charge de 2,4% par rapport à sa capacité de charge maximale, ce qui est assez faible : il n'y a donc pas de risque de saturation du bus.

**Question 13 :** La taille de la fenêtre de collision doit être au moins égale à deux fois le temps de propagation entre les deux stations les plus éloignées sur le bus (le premier nœud commence l'envoi de sa

trame, qui arrive  $\tau$  après au nœud lointain qui commence justement sa trame. Il faut alors que le signal revienne au premier nœud avec un temps  $\tau$  pour que la collision soit détectée par tous les nœuds.).

$$t_{fc} = 2\tau$$

**Question 14 :** Pour que le nœud émetteur soit certain que sa trame a été transmise sans collision, le temps d'émission entre chaque bit de la trame doit être supérieur ou égal à la fenêtre de collision :

$$\frac{1}{f} \geq t_{fc}$$

**Question 15 :**

Le temps de propagation est lié à la distance par :

$$\tau = \frac{d}{v}$$

On a d'après les questions précédentes :

$$t_{fc} = 2\tau \leq \frac{1}{f} \Rightarrow d \leq \frac{v}{2f}$$

La distance maximale entre deux nœuds sur le bus CAN afin d'éviter une collision qui ne serait pas détectée par l'émetteur est donc :

$$D_{\max} = \frac{v}{2f}$$

À 1 Mbit/s, la distance maximale entre deux nœuds est :

$$D_{\max} = \frac{1}{2 \times 5 \times 10^{-9} \times 10^6} = \frac{1}{10^{-2}} = 100 \text{ m}$$

La distance entre deux nœuds (40 m) étant inférieure à la distance maximale pour qu'une collision soit détectée, le bus CAN est viable concernant ce critère.

**Question 16 :** La saturation du bus a pour effet de ne pas transmettre les informations nécessaires, ce qui entraîne le dysfonctionnement du système. Toutefois, le calcul réalisé montre que le bus ne sera pas saturé (sauf si des composants défont et envoient plus de données qu'attendues (composants bavards)). Les erreurs de transmission des trames peuvent faire prendre de mauvaises décisions aux calculateurs. Mais :

- il y a un mécanisme pour détecter les erreurs de transmission ;
- le câble utilisé est conçu pour limiter les interférences/parasites/perturbations ;
- les informations sont transmises avec une période assez faible (40 ns), alors que les temps des mouvements physiques attendus sont de l'ordre de la seconde : une trame transmettant une information erronée sera très rapidement contredite par les trames suivantes, limitant les conséquences sur le système matériel.

Enfin, nous avons démontré que les distances entre les nœuds permettent à l'algorithme de détection de collisions d'être efficace.

Les défaillances matérielles des composants auront donc des probabilités bien plus importantes de survenir, avec des conséquences plus lourdes, que les modes de défaillance étudiés ici.

### C.3 Modélisation par un Automate à Transitions Gardées (GTS)

**Question 17 :** De façon naïve (sans optimisation), il faut :

- 2 transitions, une pour chaque mode de défaillance ;
- 2 variables d'état, pour savoir dans quel mode de défaillance on est
- une assertion pour chaque cas (dont le mode normal)

Détecteur de fin de course			
<code>working</code>	: Boolean	<code>True</code>	
<code>extinction</code>	: Boolean	<code>False</code>	
<code>courtCircuit</code>	: Boolean	<code>False</code>	
<code>modeExtinction</code>	exponentielle( $10^{-6}$ )	: <code>working</code>	-> <code>working, extinction = False, True</code>
<code>modeCourtCircuit</code>	exponentielle( $10^{-7}$ )	: <code>working</code>	-> <code>working, courtCircuit = False, True</code>
<code>fIn</code>	: Boolean	<code>False</code>	
<code>fOut</code>	: Boolean	<code>False</code>	
<code>working</code>	: out	= in	
<code>extinction</code>	: out	= <code>False</code>	
<code>courtCircuit</code>	: out	= <code>True</code>	

**Question 18 :**

- ajout de 2 variables d'état : une pour savoir que l'on est dans l'état collé, et un pour se souvenir de la valeur de la sortie lors de la défaillance.
- 1 transition pour entrer dans le mode collé ;
- 1 assertion pour affecter la sortie à la valeur de la variable d'état lorsque l'on est dans le mode collé.

Détecteur de fin de course			
<code>working</code>	: Boolean	<code>True</code>	
<code>extinction</code>	: Boolean	<code>False</code>	
<code>courtCircuit</code>	: Boolean	<code>False</code>	
<code>colle</code>	: Boolean	<code>False</code>	
<code>colleValeur</code>	: Boolean	<code>False</code>	
<code>modeExtinction</code>	exponentielle( $10^{-6}$ )	: <code>working</code>	-> <code>working, extinction = False, True</code>
<code>modeCourtCircuit</code>	exponentielle( $10^{-7}$ )	: <code>working</code>	-> <code>working, courtCircuit = False, True</code>
<code>modeColle</code>	exponentielle( $10^{-5}$ )	: <code>working</code>	-> <code>working, colle, colleValeur = False, True, fIn</code>
<code>fIn</code>	: Boolean	<code>False</code>	
<code>fOut</code>	: Boolean	<code>False</code>	
<code>working</code>	: out	= in	
<code>extinction</code>	: out	= <code>False</code>	
<code>courtCircuit</code>	: out	= <code>True</code>	
<code>colle</code>	: out	= <code>colleValeur</code>	

**Question 19 :** Plusieurs façons de faire, par exemple de façon assez naïve :

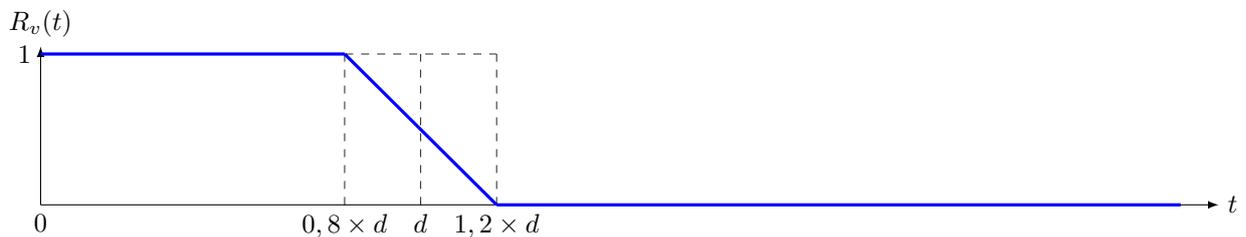
- ajout de 3 variables d'état : une pour la validité de chacune des informations ;
- 1 variable d'état pour se souvenir qu'il y a une anomalie (optionnel, on peut se débrouiller en comptant le nombre d'invalidité) ;
- 3 transitions immédiates pour la première détection, au moins une pour la détection d'anomalie ;
- assertions pour le mode normal, au moins une pour le mode dégradé, et une pour le mode d'anomalie.

Fusion d'information des détecteurs			
working1	: Boolean	True	
working2	: Boolean	True	
working3	: Boolean	True	
anomalie	: Boolean	False	
defaillance1	constante(0)	: not anomaly and working1 and fIn1!= fIn2 and fIn1!= fIn3	-> working1 = False
defaillance2	constante(0)	: not anomaly and working2 and fIn2!= fIn1 and fIn2!= fIn3	-> working2 = False
defaillance3	constante(0)	: not anomaly and working3 and fIn3!= fIn2 and fIn3!= fIn1	-> working3 = False
defaillance1b	constante(0)	: not anomaly and not working1 and fIn2!= fIn3	-> anomalie = True
defaillance2b	constante(0)	: not anomaly and not working2 and fIn1!= fIn3	-> anomalie = True
defaillance3b	constante(0)	: not anomaly and not working3 and fIn1!= fIn2	-> anomalie = True
fIn1	: Boolean	False	
fIn2	: Boolean	False	
fIn3	: Boolean	False	
fOut	: Boolean	False	
fAnomalie	: Boolean	False	
working1	: fOut	= fIn1	
working2 and not working1	: fOut	= fIn2	
True	: anomalie	= fAnomalie	

## C.4 Dates des évènements

**Question 20 :** Pour choisir aléatoirement un délai suivant une loi probabiliste correspondant au modèle d'un composant, il suffit de choisir aléatoirement un nombre entre 0 et 1, et de chercher l'antécédent de  $R(t)$ .

**Question 21 :**



**Question 22 :**

```

1 | def delaiwerin(d:float) -> float:
2 |     # Choix aléatoire
3 |     r = random.random()
4 |     # Equation de la droite dp = a r + b (donc en inversant abscisse/ordonnée)
5 |     # b = 0.8*d
6 |     # a = (1.2*d-0.8*d)/(1-0)
7 |     dp = 1.2*d-0.4*d*r
8 |
9 |     # Dans le cas ou la fonction de répartition a été (faussetment)
10 |    # prise croissante dans la question précédente
11 |    # dp = 0.4*d*r+0.8*d
12 |
13 |    return dp

```

**Question 23 :**

```

1 | SELECT DATEDIFF(operations.date, composants.firstusage_date)
2 | FROM operations JOIN composants ON operations.sn = composants.sn
3 | WHERE composants.model = "P500" AND operations.state = "replacement" ;

```

**Question 24 :**

```

1 def LambdaExperimental(durees, t, dt):
2     cok = 0 # Compteur de composants en bon fonctionnement à la date t
3     cdef = 0 # Compteur de composants ayant défailli à la date t+- dt/2
4     for duree in durees:
5         if duree > t + dt/2:
6             # Le composant est en bon fonctionnement
7             cok += 1
8         elif abs(duree-t) < (dt/2):
9             # Le composant a défailli autour de t
10            cdef += 1
11    if cok == 0:
12        # S'il n'y a plus aucun composant en bon état
13        # Alors il n'y a plus rien à défailir...
14        return 0
15    else:
16        # Calcul du taux de défaillance
17        return cdef/(cok * dt)

```

### Question 25 :

```

1 def TracerLambdaExperimental(durees):
2     tmax = max(durees)
3     T = np.linspace(0, tmax, 100) # On trace arbitrairement sur 100 points
4     dt = T[1] # Détermination du pas de temps
5     lambdas = []
6     for t in T:
7         lambdas.append(LambdaExperimental(durees, t, dt))
8
9     # Tracé
10    plt.plot(T, lambdas)
11    plt.xlabel("Temps")
12    plt.ylabel("Taux de défaillance")
13    plt.title("Taux de défaillance en fonction du temps")
14    plt.show()
15

```

### Question 26 :

```

1 def LambdaWeibull(alpha, mu, eta, t):
2     return ((alpha+mu*t)/eta)*((t/eta)**(alpha+mu*t-1))
3
4 def erreur(durees, alpha, mu, eta):
5     somme = 0.0
6     tmax = max(durees)
7     for t in np.linspace(0, tmax, 1000):
8         somme += (LambdaWeibull(alpha, mu, eta, t) - LambdaExperimental(durees, t))**2
9     return somme/1000

```

**Question 27 :** Les ordres de grandeur semblent correspondre aux données expérimentales. On remarque que le premier modèle a tendance à surestimer le taux de défaillance aux petites dates, tandis que le second modèle va plutôt l'approcher de façon basse. Le premier modèle va donc être plus pessimiste (il modélise des dates de pannes plus petites) que le second modèle : c'est une bonne chose, puisque cela évitera de sous-dimensionner le système. On utilisera donc le premier modèle.

## C.5 Simulation d'un GTS

### Question 28 :

Étape	1	2.1	2.2	2.3	3
Input		True	True	True	True
Output			True	True	True
EV1.fLeft		True	True	True	True
EV1.fRight		True	True	True	True
EV2a.fLeft		True	True	True	True
EV2a.fRight				True	True
EV2b.fLeft			True	True	True
EV2b.fRight			True	True	True

**Question 29 :** Pour arrêter, il suffit que l'une de ces deux conditions soit vérifiée :

- toutes les variables ont été définies;
- l'itération précédente n'a mis à jour aucune variable.

Seule la deuxième condition est strictement nécessaire, mais la première permet d'éviter une itération de calcul.

Une condition du type « autant d'itérations qu'il y a de variables de flux » permet d'assurer le critère d'arrêt, mais au prix d'un nombre beaucoup plus important d'itération (augmentation d'un ordre de complexité).

**Question 30 :**

```

1 def Propagation(self):
2
3     # Réinitialisation de toutes les variables de flux
4     for vf in self.getVariablesFlux():
5         vf.setNonDefinie()
6
7     nvfp, nvf = -1, 0 # Nombre de variables de flux précédemment définies, et actuellement définies
8
9     while nvfp != nvf: # Tant que des variables de flux sont nouvellement définies
10        nvfp = nvf
11        for assertion in self.getAssertions():
12            # Peut-t-on utiliser l'assertion ?
13            toutesdefiniees = True
14            for vf in assertion.getVariablesFluxNecessaires():
15                if not vf.isDefinie():
16                    toutesdefiniees = False
17
18            # Si elles sont toutes définies, on peut l'utiliser
19            if toutesdefiniees:
20                if assertion.isConditionVerifiee():
21                    transition.affecter()
22
23            # Décompte du nombre de variables de flux définies
24            nvf = 0
25            for vf in self.getVariablesFlux():
26                if vf.isDefinie():
27                    nvf += 1
28
29
30        # On met toutes les variables de flux non définies à leurs valeurs par défaut
31        for vf in self.getVariablesFlux():
32            if not vf.isDefinie():
33                vf.setValeurDefaut()

```

**Question 31 :** Il faut :

- retirer les assertions de la liste une fois qu'on les a utilisés;
- donc travailler sur une copie de la liste (pour ne pas modifier celle du GTS);
- et utiliser une boucle while pour itérer sur l'ensemble des assertions (itération sur une liste que l'on modifie);
- changer le critère d'arrêt.

```

1 def Propagation(self):
2     assertions = []
3     # Création d'une copie de la liste des assertions
4     for assertion in self.getAssertions():
5         assertions.append(assertion)
6
7     # Réinitialisation de toutes les variables de flux
8     for vf in self.getVariablesFlux():
9         vf.setNonDefinie()
10
11
12     na = -1 # Nombre d'assertions à la précédente itération
13     while na != len(assertions) and len(assertions) > 0: # Tant que le nombre d'assertions restantes
14         # n'a pas évolué
15         na = len(assertions)
16
17         while i < len(assertions): # Boucle while
18             assertion = assertions[i]
19             # Peux-t-on utiliser l'assertion ?
20             toutesdefinies = True
21             for vf in assertion.getVariablesFluxNecessaires():
22                 if not vf.isDefinie():
23                     toutesdefinies = False
24
25             # Si elles sont toutes définies, on peut l'utiliser
26             if toutesdefinies:
27                 if assertion.isConditionVerifiee():
28                     transition.affecter()
29                     # Dans tous les cas (condition vérifiée ou non), l'assertion n'est plus à utiliser
30                     assertions.pop(i)
31                 else: # Si on n'a pas supprimé d'assertion
32                     i += 1
33
34             # Plus d'assertion utilisable, on met toutes les variables de flux non définies à leurs valeurs
35             # par défaut.
36             for vf in self.getVariablesFlux():
37                 if not vf.isDefinie():
38                     vf.setValeurDefaut()

```

**Question 32 :** Il faut :

- supprimer de l'échéancier les transitions qui ne sont plus tirables ;
- ajouter à l'échéancier les transitions qui sont tirables (sans être présentes dans l'échéancier), en choisissant une date de tir (calculée par la date actuelle plus le délai aléatoire).

```

1 def MiseAJourEcheancier(self):
2     for transition in self._transitions:
3         # Recherche de la transition dans l'échéancier
4         found = False
5         i = 0
6         while i < len(self._echeancier) and not found:
7             if self._echeancier[i][1] == transition:
8                 found = True
9                 # Est-ce que l'on doit la retirer ?
10                if not transition.Garde():
11                    echeancier.pop(i)
12                i += 1
13
14            # Est-ce que l'on doit l'ajouter ?
15            if transition.Garde() and not found:
16                datetir = self._date + transition.Temporisation()
17                self._echeancier.append((datetir, transition))
18            # Pas de return, modification en place

```

**Question 33 :**

```

1 # Prochaine transition : celle qui à la date de tir la plus basse
2 def ProchaineTransition(self):

```

```

3     datemin = self._echeancier[0][0]
4     indicemin = 0
5     for i in range(len(self._echeancier)):
6         if self._echeancier[i][0] < datemin:
7             indicemin = i
8     return indicemin
9

```

### Question 34 :

MiseAJourEcheancier() :

- double boucle imbriquée : pour chaque transition, pour chaque élément de l'échéancier ;
- si l'on suppose que le nombre de transitions dans l'échéancier est du même ordre que le nombre de transitions, on est en  $\mathcal{O}(n_t^2)$  opérations élémentaires, avec  $n_t$  le nombre de transitions du modèle ;
- pop en  $\mathcal{O}(\text{len}(L))$  opérations élémentaires, réalisé au pire  $n_t$  fois, donc en  $\mathcal{O}(n_t^2)$  opérations élémentaires aussi.

ProchaineTransition() :

- simple boucle sur chaque transition de l'échéancier ;
- si l'on suppose que le nombre de transitions dans l'échéancier est du même ordre que le nombre de transitions, on est en  $\mathcal{O}(n_t)$  opérations élémentaires, avec  $n_t$  le nombre de transitions du modèle.

### Question 35 : Il faut :

- choisir la prochaine transition à tirer ;
- avancer la date actuelle à la date de tir ;
- tirer la transition ;
- calculer les nouvelles valeurs de variables de flux ;
- mettre à jour l'échéancier.
  - en n'oubliant pas de retirer la transition qui vient d'être tirée, et en la réinsérant si nécessaire.

```

1 def ProchainTir(self):
2     i = self.ProchaineTransition()
3     self.date = self._echeancier[0] # Mise à jour de la date actuelle
4     transition = self._echeancier[i][1]
5     transition.Action() # Tir de la transition, ce qui modifie les variables d'état
6     self.Propagation() # Mise à jour des variables de flux
7     self._echeancier.pop(i) # Retrait de la transition qui vient d'être tirée de l'échéancier
8     self.MiseAJourEcheancier() # Mise à jour de l'échéancier

```

### Question 36 :

```

1 def MiseAJourEcheancier(self):
2     for i in range(len(self._echeancier)):
3         datetir, transition = self._echeancier[i]
4         if transition.Garde() and datetir == None:
5             # La transition est tirable, et n'a pas de date de tir : il faut l'ajouter
6             self._echeancier[i][0] = date + transition.Temporisation()
7
8         elif transition.Garde() and datetir != None:
9             # La transition n'est pas tirable mais a une date de tir : il faut la retirer
10            self._echeancier[i][0] = None

```

```

1 # Prochaine transition : celle qui à la date de tir la plus basse différente de None
2 def ProchaineTransition(self):
3     datemin = self._echeancier[0][0]
4     indicemin = 0
5     for i in range(len(self._echeancier)):
6         if self._echeancier[i][0] != None: # Prise en compte des transitions non tirables
7             if self._echeancier[i][0] < datemin:
8                 indicemin = i
9     return indicemin
10

```

```

1 def ProchainTir(self):
2     i = self.ProchaineTransition()
3     self.date = self._echeancier[0] # Mise à jour de la date actuelle

```

```

4 transition = self._echeancier[i][1]
5 transition.Action() # Tir de la transition, ce qui modifie les variables d'état
6 self.Propagation() # Mise à jour des variables de flux
7
8 # Modification
9 self._echeancier[i][0] = None # Retrait de la transition qui vient d'être tirée de l'échéancier
10
11 self.MiseAJourEcheancier() # Mise à jour de l'échéancier
12

```

### Question 37 :

MiseAJourEcheancier() :

- en  $\mathcal{O}(n_t)$  opérations élémentaires, avec  $n_t$  le nombre de transitions du modèle.
- c'était en  $\mathcal{O}(n_t^2)$  opérations élémentaires : c'est une amélioration.

ProchaineTransition() :

- c'est toujours en  $\mathcal{O}(n_t)$  opérations élémentaires, avec  $n_t$  le nombre de transitions du modèle.
- pas d'amélioration, mais ce n'est pas la complexité limitante.

On a donc une amélioration de la complexité temporelle (passage de quadratique à linéaire en nombre de transitions du modèle).

### Question 38 :

```

1 # Construction de la partie Action->Etat
2 # Les variables d'état intéressantes sont celles qui sont affectées par l'action
3 Lvaretats = inspect(transition._action)[0]

```

### Question 39 :

```

1 # Construction de la partie Etat->Flux
2 # Les variables d'état intéressantes sont celles qui sont directement affectées par des variables
3 # d'état, ainsi que celles qui sont indirectement affectées par les variables de flux affectées
4 Lvarflux = []
5 longueurprecedente = -1
6 # On parcourt les assertions, tant que l'on découvre de nouvelles variables de flux modifiées
7 while len(Lvarflux) != longueurprecedente:
8     longueurprecedente = len(Lvarflux)
9     for assertion in GTS.getAssertions():
10        write, read = inspect(assertion)
11        estutile = False
12        for v in read:
13            if v in Lvaretats or v in Lvarflux:
14                estutile = True
15        # Si l'assertion utilise une des variables affectées
16        if estutile:
17            for v in write:
18                if v not in Lvarflux:
19                    Lvarflux.append(v)

```

### Question 40 :

```

1 # Construction de la liste Etat+Flux-> Gardes
2 Lgardes = []
3 for tr in GTS.getTransitions():
4     write, read = inspect(tr._garde)
5     affecte = False
6     for v in read:
7         if v in Lvaretats or v in Lvarflux:
8             affecte = True
9     if affecte:
10        Lgardes.append(tr)

```

Question 41 : ProchainTir() doit être modifiée, uniquement pour indiquer à MiseAJourEcheancier() la transition tirée.

```

1 def ProchainTir(self):
2     i = self.ProchaineTransition()
3     date = self._echeancier[0] # Mise à jour de la date actuelle
4     transition = self._echeancier[i][1]
5     transition.action() # Tir de la transition, ce qui modifie les variables d'é
6     tat
7     self.Propagation() # Mise à jour des variables de flux
8     self._echeancier[i][0] = None # Retrait de la transition qui vient d'être tirée de l'é
9     cheancier
10
11     self.MiseAJourEcheancier(transition) # Mise à jour de l'échéancier

```

MiseAJourEcheancier(transition) doit être redéfinie pour travailler uniquement sur les transitions affectées

```

1 def MiseAJourEcheancier(self, transitiontiree):
2     for i in range(len(self._echeancier)):
3         datetir, transition = self._echeancier[i]
4
5         # Modification
6         if transition in transitiontiree.gardesaffectees or transition == transitiontiree:
7             # Il faut dans tous les cas réétudier le cas de la transition qui vient d'être tirée.
8             # Fin modification
9
10            if transition.garde() and datetir == None:
11                # La transition est tirable, et n'a pas de date de tir : il faut l'ajouter
12                self._echeancier[i][0] = self.date + transition.evenement()
13
14            elif transition.garde() and datetir != None:
15                # La transition n'est pas tirable mais a une date de tir : il faut la retirer
16                self._echeancier[i][0] = None

```

**Question 42 :** De la même manière, il est possible de ne pas avoir à recalculer toute la propagation des assertions, et de se limiter uniquement aux assertions et variables de flux qui sont affectées par le tir d'une transition.

## C.6 Simulation stochastique

**Question 43 :**

- Condition d'arrêt : `BonFonctionnement == False`, on s'arrête dès que l'état de bon fonctionnement a été quitté;
- Grandeur mesurée : `date`, on a besoin de la date à laquelle l'état de bon fonctionnement a été quitté.

**Question 44 :** En raison des problèmes d'approximation numérique / représentation des nombres en mémoire.

Comme l'on a un grand nombre de mesures, pour le calcul de la moyenne, on va rapidement additionner un petit nombre à un grand nombre.

De plus, cela peut représenter un grand nombre de valeurs à stocker en mémoire.

**Question 45 :**

```

1 def SimulationMTTF(GTS):
2     ta = 1.960
3     n = 0 # Nombre de simulations réalisées
4     largeur = 10 # Largeur de l'intervalle de confiance relativement à la moyenne
5
6     moyenne = 0.0
7     variance = 1.0
8     M2 = 0.0
9
10    # On récupère la variable de flux "BonFonctionnement"
11    for vf in GTS.getVariablesFlux():
12        if vf.getNom() == "BonFonctionnement":

```

```

13         BonFonctionnement = vf
14
15     while largeur > 0.01:
16         # Début d'une simulation
17         GTS.setdate(0.0)           # Initialisation de la date
18         for v in GTS.getVariablesEtat():
19             v.setValeurInitiale() # Réinitialisation de toutes les variables d'état
20         GTS.Propagation()         # Propagation de l'état initial
21         GTS.MiseAJourEcheancier() # Echeancier initial
22
23         while BonFonctionnement.getValeur(): # Boucle de simulation
24             GTS.ProchainTir()
25
26         # Calcul des statistiques
27         n += 1
28         date = GTS.getDate()
29         moyennenuvelle = moyenne + (date - moyenne)/n
30         M2nouvelle = M2 + (date-moyennenuvelle)*(date-moyenne)
31         if n > 1:
32             variancenuvelle = M2nouvelle / (n - 1)
33
34         moyenne = moyennenuvelle
35         M2 = M2nouvelle
36         variance = variancenuvelle
37
38         # Largeur de l'intervalle de confiance
39         largeur = 2*sqrt(variance)*ta / sqrt(n)
40
41     return moyenne

```

**Question 46 :** Il faut détecter quand l'évènement survient :

- Une solution est, dans le modèle, d'avoir une transition qui a
  - pour garde la commande de sortie du train d'atterrissage ET NON (les roues sorties et verrouillées et les portes fermées);
  - pour temporisation 15 secondes;
  - pour action une affectation sur une variable booléenne, qui sera celle surveillée dans `SimulationMTTF()` pour compter l'occurrence de l'évènement.

Et, dans `SimulationMTTF()` :

- La fin de simulation est quand une panne est signalée, ou éventuellement quand l'évènement survient;
- Il faut sommer la durée de simulation (à la fin de chaque simulation, sommer la date à un accumulateur);
- Il faut compter le nombre de fois où l'évènement survient;
- Il faut renvoyer le ratio (nombre de fois où l'évènement survient / durée de simulation).
- Le nombre de simulations à réaliser peut être décidé arbitrairement (pas de solution évidente, il faudrait donner les formules de calcul d'intervalle de confiance).

**Question 47 :** Les taux de défaillance sont assez faibles :

- la MTTF indique un temps moyen avant défaillance d'environ 20 ans. Un avion ayant une durée de vie de 25 à 30 ans, il est probable qu'une grande partie des avions rencontre un problème dans leur vie. Mais une maintenance régulière va permettre de fortement limiter ce risque, en remplaçant les composants défaillant avant que leurs redondances ne tombent aussi en panne.
- le taux d'occurrence concernant l'exigence ( $R_{11}$ ) indique que le problème surviendra une fois tous les 20'000 ans, ou plutôt, pour un avion sur 690 (avec une durée de vie de 30 ans). C'est un taux très faible, voir improbable, et qui sera encore plus réduit avec une maintenance régulière.

On peut aussi augmenter les redondances pour réduire encore ces taux.

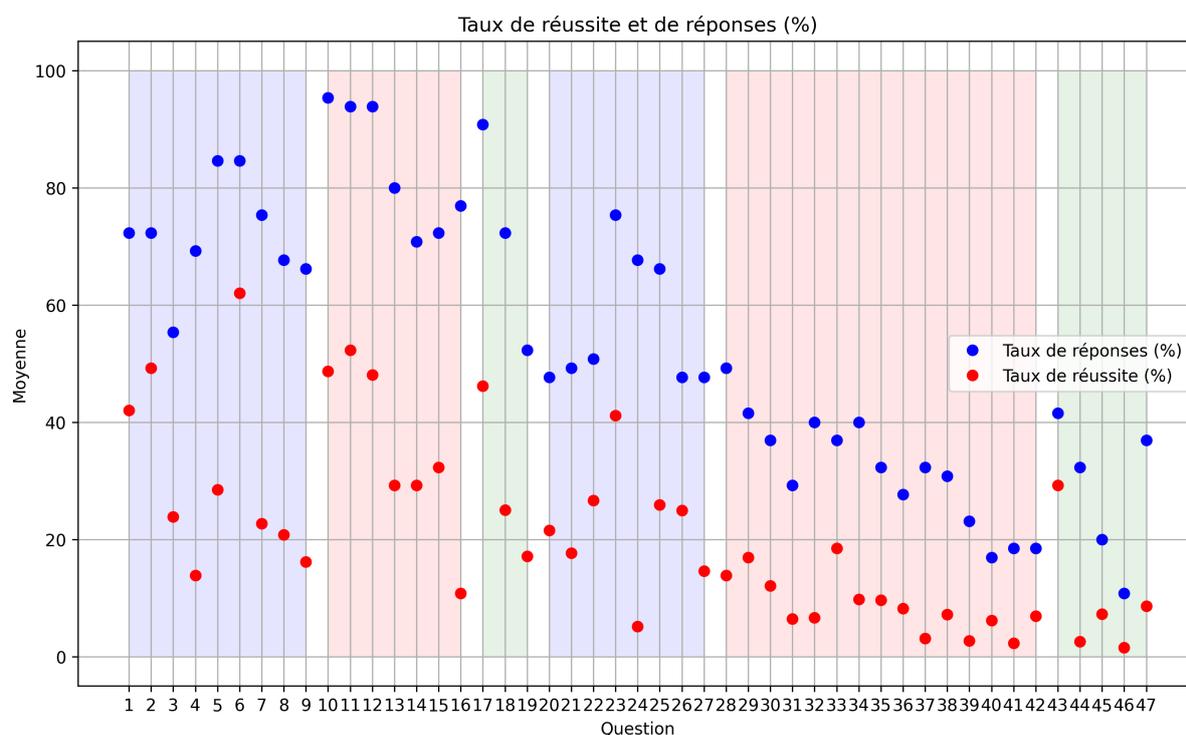
## D Commentaires du jury

Le sujet portait sur l'étude dysfonctionnelle d'un système aéronautique : l'objectif était de mettre en place un modèle afin de prédire plusieurs de ses performances dysfonctionnelles, notamment sa durée

moyenne avant d'être défaillant.

Pour cela, un premier outil de modélisation était d'abord étudié (chaînes de Markov), qui s'avère inadapté à une étude d'un système complexe. Ensuite, les modes de défaillance les plus probables du réseau de communication étaient passés en revue. Enfin, un modèle de plus haut niveau de modélisation que les chaînes de Markov était mis en place, permettant de simuler de façon stochastique et sur des grands temps le comportement du système, et ainsi de calculer par simulation de Monte-Carlo les performances du système.

Les 6 parties du sujet étaient indépendantes. Le sujet a été globalement traité dans son entièreté. Toutes les parties ont été abordées dans la grande majorité des copies.



La première partie introduisait la modélisation par des chaînes de Markov d'un seul composant, ou d'un ensemble de deux composants. Aucune connaissance sur les chaînes de Markov n'était attendue ou nécessaire. En revanche, des connaissances sur les systèmes d'équations linéaires, sur les produits vecteur-matrice (utilisés dans plusieurs domaines des Sciences de l'Ingénieur), et sur les graphes (au programme de la spécialité NSI en classe de Terminale) étaient nécessaires.

L'écriture du système d'équations linéaires par la méthode d'Euler a été correctement réalisée, mais son écriture sous forme matricielle, puis l'utilisation de sa diagonalisation (donnée dans le sujet) pour calculer sa  $n$ -ième puissance ont posé des difficultés. La modélisation du comportement de deux composants en redondance soit froide, soit chaude, en représentant la chaîne de Markov sous forme de graphe orienté a aussi été source de difficultés, avec des réponses hors sujet, des graphes incomplets, non orientés, avec plusieurs sommets pour un même état.

La question de conclusion sur cet outil de modélisation que sont les chaînes de Markov a été assez mal traitée : la grande majorité des candidats mettant en avant la difficulté à traiter la modélisation "à la main", en oubliant que l'utilisation d'objets mathématiques tels que les graphes et le calcul matriciel est avant-tout pour permettre de construire et manipuler des modèles complexes par un ordinateur, et que le calcul "sur feuille" n'est qu'un moyen de vérifier la compréhension de l'outil étudié. Si le nombre de sommets du graphe est parfois correct, il a rarement été comparé aux capacités d'un ordinateur qui sont ici largement dépassées.

La seconde partie était consacrée à l'étude des modes de défaillance du réseau de communication. Elle a été globalement traitée, mais les réponses ont souvent été imprécises et peu rigoureuses, par exemple pour expliquer l'utilité des deux fils de la liaison CAN (dans la partie « Erreur de transmission des trames »). La partie sur la saturation du bus demandait de déterminer la taille d'une trame, à l'aide

de la documentation en annexe : cette taille n'a pas toujours été correctement calculée. Enfin, l'étude des collisions de trames a donné lieu à des relations erronées, souvent non-homogènes. La question de conclusion a rarement été traitée correctement, et a souvent consisté en une justification du besoin d'une étude dysfonctionnelle, plutôt qu'en une comparaison des probabilités et conséquences des différents modes de défaillance.

La troisième partie proposait la modélisation du système par un automate à état-transition. Aucune connaissance sur les GTS n'était attendue ou nécessaire, mais des notions liées aux automates à états (tel que ceux des diagrammes états-transitions, au programme de la spécialité « Sciences de l'ingénieur », ou dans des automates industriels) étaient nécessaires. La représentation de tels automates par des listes de variables, de transitions et d'assertion, et non graphiquement, a probablement perturbé certains candidats. Cette représentation permettait toutefois d'aborder plus simplement la simulation de ces modèles.

La quatrième partie portait sur la modélisation du caractère stochastique des défaillances des composants, obligeant à choisir aléatoirement quand une transition de l'automate est tirée, à partir d'une distribution de probabilité. Cette partie a mis en évidence des difficultés dans la manipulation de ces notions :

- confusion entre image et antécédent d'une fonction mathématique ;
- compréhension d'une documentation Python, pour l'utilisation de la bibliothèque `random` et le tracé de courbes avec `matplotlib.pyplot` ;
- concernant les bases de données, une méconnaissance du principe de jointure symétrique.

La cinquième partie proposait la mise en place des mécanismes de simulation des automates à état-transition étudiés, avec une étude de la simulation des flux entre composants de façon acausale, puis la mise en place de la gestion des événements et des transitions associées en cherchant à optimiser les structures de données utilisées. Cette partie faisait appel à des notions d'algorithmie, de graphes, de structures de données, et de complexité algorithmique.

Une des questions demandait de décrire, via un tableau, la mise à jour d'un ensemble de variables au fur et à mesure de l'exécution d'un algorithme décrit dans le sujet. Cette question d'algorithmie a été assez mal traitée.

Un ensemble de questions, concernant la gestion de l'échéancier, demandait de définir des fonctions utiles à la simulation de l'automate (l'algorithme était décrit, avec plus ou moins de précisions, dans le sujet), puis d'en déterminer la complexité temporelle. La détermination de la complexité temporelle a été rarement réussie, et encore plus rarement justifiée correctement. L'utilisation de la notation "grand O" pour décrire le comportement asymptotique de la complexité d'une fonction n'est que très peu utilisée.

Enfin, la dernière partie proposait de mettre en place un algorithme de simulation de Monte-Carlo, en utilisant la modélisation étudiée dans les parties précédentes pour générer un grand nombre de simulations différentes et en calculer des indicateurs statistiques sur les modèles dysfonctionnels. Il n'était pas nécessaire d'avoir défini dans les parties précédentes les fonctions appelées ici.

Dans cette partie, l'algorithme de Welford était proposé : celui-ci permet de contourner le problème de la représentation des nombres en mémoire lors du calcul de moyennes et d'écart-types d'un grand nombre de valeurs, ce qui est un problème classique en ingénierie numérique.

Cette partie se terminait par une conclusion sur des résultats obtenus par simulation, qui étaient à interpréter du point de vue des écarts avec les éléments de contexte donnés en début de sujet. Assez peu de candidats ont traduit les résultats numériques dans une unité compréhensible, et ont ainsi pu conclure sur la viabilité du système étudié en justifiant leur argumentation.

## E Recommandations aux futurs candidats

Il est recommandé aux futurs candidats à l'Agrégation de Sciences industrielles de l'Ingénieur en spécialité Ingénierie Informatique de s'approprier le programme de ce concours, ainsi que ceux des filières dans lesquelles les futurs enseignants seront amenés à exercer (SNT, SI, STI2D, BTS, CPGE...).

Il est aussi recommandé de s'attacher à lire attentivement les questions posées dans le sujet, et de chercher à répondre à toute la question mais uniquement à la question posée. Les copies qui demandent au correcteur de sélectionner les éléments pertinents parmi plusieurs réponses possibles sont pénalisées lors de la correction.

# Épreuve d'admissibilité de jury de l'épreuve de conception préliminaire d'un système, d'un procédé ou d'une organisation

## 1 Présentation de l'épreuve

*Arrêté du 28 décembre 2009 modifié*

- Durée totale de l'épreuve : 6 heures
- Coefficient 1

L'épreuve est spécifique à l'option choisie. A partir d'un dossier technique comportant les éléments nécessaires à l'étude, l'épreuve a pour objectif de vérifier les compétences d'un candidat à synthétiser ses connaissances pour proposer ou justifier des solutions de conception et d'industrialisation d'un système pluritechnologique dans le domaine de la spécialité du concours dans l'option choisie.

## 2 Sujet

Le sujet est disponible en téléchargement sur le site du ministère à l'adresse :

<https://www.devenirenseignant.gouv.fr/media/15954/download>

Il traite d'un système composé de multiples drones pour les spectacles lumineux. Il est découpé en 4 parties indépendantes. La première partie est consacrée à l'analyse de la commande d'un quadricoptère. La deuxième, analyse l'exploitation informatique des données issues d'un détecteur d'obstacles de type Lidar. La troisième partie étudie le traitement de la télémétrie de chaque drone de l'essaim. La dernière partie traite de la suite logicielle utile aux opérations de suivi et de maintenance des drones.

## 3 Remarques du jury

**Remarques générales :** La moyenne de l'épreuve est de 4,76/20, avec un maximum à 15,32/20 et un écart type de 3,72/20. On observe une répartition déséquilibrées des notes (voir Figure 1). Si plusieurs candidats ont pu démontrer leurs connaissances et leur capacité de raisonnement, le jury souligne cependant, chez un trop grand nombre de candidats, une tendance à survoler le sujet et à répondre aux questions de manière vague, sans réellement traiter le fond des questions. Le jury invite les candidats se préparant aux épreuves, à lire attentivement les questions pour y répondre de manière pertinente et précise. L'épreuve a mis en évidence, chez certains candidats, une méconnaissance des notions de précision pour les nombres en virgule flottante, des difficultés à convertir des données décimales en binaire et en hexadécimal et des candidats n'ont pas démontré une maîtrise élémentaire des notions de Python, de C, de C++, de modélisation orientée objet et de complexité des algorithmes. Le jury invite les candidats à s'assurer de la maîtrise de ces points pour les sessions suivantes et à consulter attentivement le programme du concours.

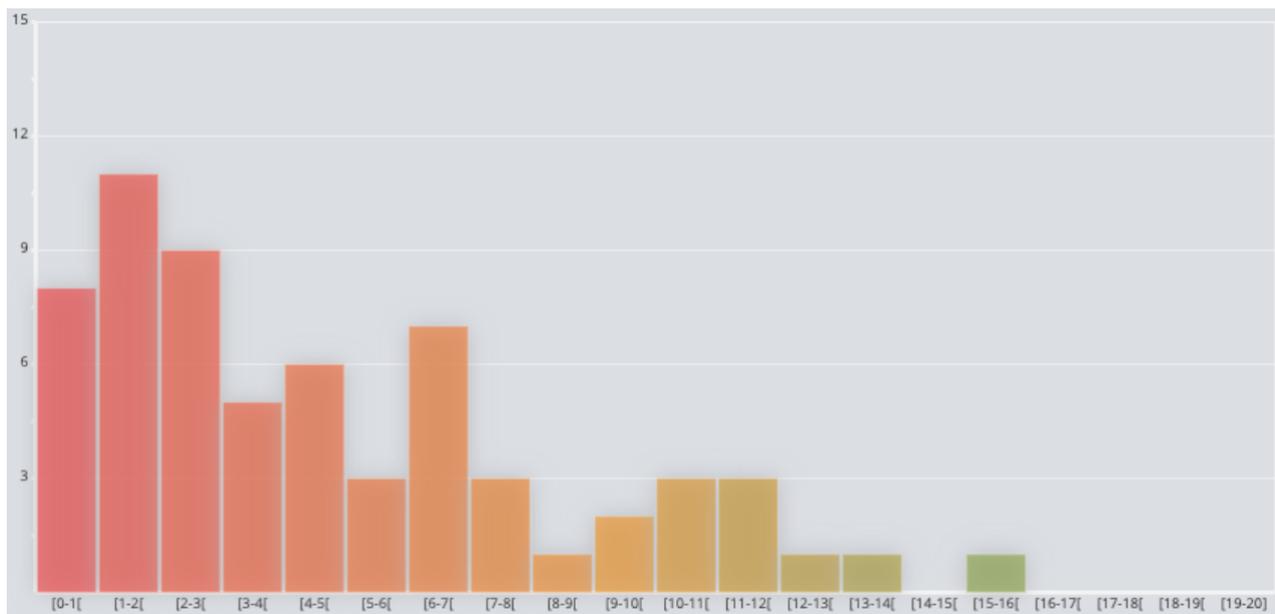


FIGURE 1 – Histogramme des notes

**Partie 1 :** L'objectif de cette partie est d'analyser la logique de pilotage d'un drone en étudiant les actionneurs et les capteurs afin d'asservir les grandeurs de positionnement (tangage, lacet, roulis, altitude). Pour traiter cette partie, le candidat doit maîtriser le positionnement d'un objet dans le plan, les asservissements numériques, les bus de terrain ainsi que la modélisation UML notamment la lecture d'un diagramme de classes. Dans l'ensemble, cette partie a été correctement traitée.

92% des candidats ont traité cette partie. Les notes s'étalent de 0 à 20 points sur 25 avec une moyenne de 7,24/25. 12 copies ont obtenu plus de la moitié des points de cette partie.

**Partie 2 :** L'objectif de cette partie est de mettre en œuvre un détecteur d'obstacles de type Lidar ainsi que traiter les données récupérées afin de les stocker dans une mémoire tampon circulaire et de les exploiter afin de détecter un ou plusieurs obstacles. Cette partie traite aussi de la logique de déviation de trajectoire du drone afin de les contourner. Cette partie demande au candidat de maîtriser la programmation orientée objet, le langage de programmation C++ et la gestion de processus concurrents.

92% des candidats ont traité cette partie. Les notes s'étalent de 0 à 19,5 points sur 25 avec une moyenne de 5,62/25. 6 copies ont obtenu plus de la moitié des points de cette partie.

**Partie 3 :** L'objectif de cette partie est d'étudier le système de télémétrie mis en place pour surveiller les drones et le cas échéant les localiser pour récupération. Il s'agit de mesurer et de traiter les informations de différents capteurs, les analyser et transmettre des messages à intervalles réguliers via un réseau Zigbee et un courtier de messages MQTT. 89% des candidats ont traité cette partie. Les notes s'étalent de 0,5 à 24,35 points sur 33 avec une moyenne de 8,37/33. 9 copies ont obtenu plus de la moitié des points de cette partie.

**Partie 4 :** L'objectif de cette partie est d'étudier les différents services qui composent la suite logicielle à disposition des intervenants techniques du spectacle. Sur la base d'un schéma présentant l'architecture générale du système et d'extraits de code, il est demandé : d'analyser des classes et des tables de bases de données, de traiter de notions de réseau et de sécurité ainsi que de concurrence d'accès aux données.

66% des candidats ont traité cette partie. Les notes s'étalent de 0 à 17 points sur 17 avec une moyenne de 6,88/17. 16 copies ont obtenu plus de la moitié des points de cette partie.

## 4 Éléments de correction

Q1. La rotation autour de l'axe x est due au moment créé par la différence entre les forces de portance des rotors 2 et 4, ce moment est donné par la relation suivante :  $T_x = d \times b \times (\omega_4^2 - \omega_2^2)$ .

La rotation autour de l'axe y est due au moment créé par la différence entre les forces de portance des rotors 1 et 3, ce moment est donné par la relation suivante :  $T_y = d \times b \times (\omega_3^2 - \omega_1^2)$ .

La rotation autour de l'axe z est due à un couple réactif provoqué par les couples de traînée dans chaque hélice, ce moment est donné par la relation suivante :  $T_z = k \times (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$ .

Q2.  $F_p = b \times (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$ .

Q3.

$$P = \begin{pmatrix} 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Q4.

$$P^{-1} = \begin{pmatrix} 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 0 & -\frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ -\frac{1}{2} & 0 & -\frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

Q5. On mesure la différence des masses puis on calcule la force de pesanteur résultante notée  $F_r$ .

$$F_r = (0,632 - 0,612) \times 9,81 \Rightarrow F_r = 0,1962 \text{ N}$$

Cette force correspond à 4 fois la force exercée par une hélice.

$$F_r = 4 \cdot F_p \Rightarrow F_p = 49,05 \text{ mN}$$

$$F_p = b \times \omega^2$$

$$\omega = \frac{2\pi N}{60}$$

$$b = \frac{F_p}{\omega^2} \Rightarrow b = 3,69 \times 10^{-6} \text{ N rad}^{-2} \text{ s}^2$$

Q6.

$$k = \frac{F_t * d}{2 \times \omega^2}$$

Pour cette question, tout raisonnement intégrant la distance entre le centre de gravité et l'axe moteur a été compté juste.

Q7.

```
void Motor::setPulse(double percentageSpeed)
{
    float period;
    if (direction==true) {
        // sens rotation sens trigonométrique
        period =(float) (1520-(( percentageSpeed *420)/100))/1000;
    } else {
        // sens rotation sens horaire
        period = (float) (((percentageSpeed *420)/100)+1520)/1000;
    }
    //réglage de la vitesse du moteur à la voie channel.
    pwm->set_duty_cycle(channel,period);
}
```

Q8. C'est le calculateur embarqué pour les deux questions car il est le maître du bus.

Q9. Bus SPI (1Mhz) est plus rapide que le bus I2C (400kHz), tous les deux sont synchrones, SPI utilise 4 lignes alors que l'I2C n'en utilise que 2 (lecture du Block Diagram de la documentation MPU9255).

Q10. L'adresse hexadécimale de l'esclave I<sup>2</sup>C (capteur MPU9255) est 0x68. C'est une requête d'écriture car  $R/\overline{W}=0$ . L'adresse du registre 0x3B correspond à l'octet de poids fort de l'accélération sur l'axe x.

	Start	Adresse esclave	$R/\overline{W}$	ACK	Adresse registre	ACK
bits	0	110 1000	0	0	0111011	0
Valeur hexadécimale		0x68			0x3B	

Q11.

Accélérations	Orientations
Ax = 0xFF31 soit - 207	Gx = 0xFFE8 soit -24
Ay = 0x0113 soit 275	Gy = 0x0009 soit 9
Az = 0x086F soit 2159	Gz = 0x0002 soit 2

Q12.

```
int16_t * MPU9255::readGyro()
{
    uint8_t adrDepart= 0x43 ou 67 en décimal; // adresse extraite de la documentation
    uint8_t gyr[6];
    int16_t g[3];
    i2c->read(ADR_MPU9255, adrDepart, 06, gyr);
    int16_t g[0]=gyr[0]<<8 | gyr[1];
    int16_t g[1]=gyr[2]<<8 | gyr[3];
    int16_t g[2]=gyr[4]<<8 | gyr[5];
    return g;
}
```

Q 13.

$$\tilde{u}_{\psi_n} = K_{p\psi} \varepsilon_{\psi_n} + K_{i\psi} \sum_{i=0}^{i=n} \varepsilon_{\psi_i} T_e + K_{d\psi} \frac{\varepsilon_{\psi_n} - \varepsilon_{\psi_{n-1}}}{T_e} \quad (1)$$

Q 14. Il faut prévoir un bloc de saturation afin de limiter l'action de l'intégrateur.

Q 15.

```
PIP ::PID(double Kp, double Ki, double Kd)
{
  _Kp=Kp;
  _Ki=Ki;
  _Kd=Kd;
  _lastTime=getTimeMs();
  _lastValue=0;
  _sumError=0;
}
```

Q 16.

```
PID::correction(double reference, double measure)
{
  double t= getTimeMs();
  error= reference-measure;
  _sumError+=error;
  diffError=error - _lastError;
  Te=t-_lastTime;
  // calcul
  double output=_Kp*error + _Ki*_sumError*Te + _Kd*diffError/Te;
  _LastError=error;
  _lastTime=t;
}
```

Q 17.

$$\begin{aligned} K_p &= 1,8 \\ K_i &= 1,33 \\ K_d &= 0,6 \end{aligned}$$

- Q 18.   ▪ Gain proportionnel : rapidité,
- Gain intégral : précision ou annule l'erreur statique,
  - Gain dérivé : stabilité.

Q 19. On doit augmenter le gain dérivé ou baisser le gain intégral afin de diminuer le dépassement.

Q 20. Zone A :

```
speedM1 = m1->calculPercentageSpeed(outPIDAltitude/4, 0, -outPIDPitch/2, outPIDYaw/4);
speedM2 = m2->calculPercentageSpeed(outPIDAltitude/4, -outPIDRoll/2, 0, -outPIDYaw/4);
speedM3 = m3->calculPercentageSpeed(outPIDAltitude/4, 0, outPIDPitch/2, outPIDYaw/4);
speedM4 = m4->calculPercentageSpeed(outPIDAltitude/4, outPIDRoll/2, 0, -outPIDYaw/4);
```

On met 0,5 si le candidat indique les calculs partiels de la question 4. Zone B :

```
m1->setPulse(speedM1);
m2->setPulse(speedM2);
m3->setPulse(speedM3);
m4->setPulse(speedM4);
```

Q 21. Les informations issues des capteurs de la centrale inertielle devront être écrites et lues en même temps ce qui pose un problème d'accès à ces mêmes variables.

Q 22. On peut programmer 2 threads séparés en y intégrant une exclusion mutuelle (mutex) ou un sémaphore afin de bloquer l'accès à la lecture pendant l'écriture des informations capteurs et inversement lors de la lecture des informations.

Q 23. Le temps de détection de l'obstacle est de 0,1s (fréquence de rotation du moteur 10Hz) plus le temps de réaction de l'asservissement de lacet et du traitement numérique de la déviation d'obstacle est de 1,5s donc le temps de réaction total du drone vis à vis d'un obstacle est de 1,6s. La portée du lidar étant de 8 m la vitesse maximum de déplacement doit être inférieure à  $8/1,6$  soit 5 m/s environ 18 km/h.

Q 24. L'erreur de distance est égale à 0,5 mètres pour 5m/s soit 6,5% du seuil de distance de détection. A 60 km/h, l'erreur de distance serait de 1,66 m soit 20,75% du seuil de distance de détection. Il faudrait compenser chaque mesure de distance en tenant compte de la vitesse et de la trajectoire réelle de déplacement du drone.

Q 25. Début de la trame : 0xA55A

La taille de la trame est de : 0x3A soit 58 octets

La vitesse de rotation est de : 0x1043 soit 4163  $\mu$ s soit 4163x24 99,912ms par tour

L'angle de départ est de : 0x191B soit 6427 centième de degré soit 64,27°

L'angle d'arrivée est de : 0x1DCB soit 7627 centième de degré soit 76,27°

Checksum : 0x28

$\Delta\theta : (76,27 - 64,27)/16 = 0,75^\circ$

À l'angle 64,27 °, distance à l'obstacle 0x03FD soit 1021 mm, intensité lumineuse 0xA8 soit 168

À l'angle 65,02 °, distance à l'obstacle 0x040D soit 1037 mm intensité lumineuse 0xAC soit 172

À l'angle 65,77 °, distance à l'obstacle 0x042C soit 1068 mm intensité lumineuse 0xB3 soit 179

D'après la constitution de la trame, le LIDAR effectue 16 mesures de distances et d'intensités lumineuses entre l'angle de départ et l'angle de fin.

Q 26. Dans le constructeur de la classe Lidar360, on instancie un objet de type ComSerie avec les paramètres de la liaison série donnée par le constructeur du lidar soit le nom du port pris en argument du constructeur, la vitesse de 230400 bauds et pas de parité. On doit allouer de la mémoire pour les variables de type pointeur.

```
Lidar360::Lidar360(std::string port)
{
    com = new ComSerie(port, 230400, NO);
    trame = new unsigned char[64];
    dataLidar= new uint32_t*[512];
    for (int i=0; i < 512; i++)
        dataLidar[i]= new uint32_t[3];
}
```

Q27. On doit restituer la mémoire pour les attributs où l'on a alloué de la mémoire dans le constructeur (utilisation de new ou malloc).

```
Lidar360::~~Lidar360()
{
    delete com;    ou free com; au choix
    delete[ ] trame;
    for (int i=0; i < 512; i++) delete[ ] dataLidar[i];
    delete[ ] dataLidar;
}
```

Q 28.

```
void Lidar360::lireTrame()
do
{
    com->recevoir (&trame[0], 3); //lecture des 3 premiers octets
}
while (0xA5 != trame[0] && trame[1] != 0x5A); //détection de l'entête de la trame

com->recevoir(&trame[3], trame[2]-3); //récupération du reste de la trame
}
```

Q 29.  $360/0,75=480$  valeurs

```

1  Q 30.
2  void Lidar360::stockerDataLidar()
3  {
4      uint32_t angle_debut = (uint32_t)(trame[5] << 8 | trame[6]);
5      uint32_t angle_fin = (uint32_t)(trame[55] << 8 | trame[56]);
6
7      if(angle_fin < angle_debut){angle_fin = angle_fin + 36000;};
8      uint32_t delta_angle = (uint32_t)((angle_fin - angle_debut)/16);
9      int j = 0;
10     uint32_t depart=(uint32_t)(angle_debut/delta_angle);
11     mx.lock(); // Exclusion mutuelle Question suivante
12     for (uint32_t i = depart; i < depart +16; i++)
13     {
14         dataLidar[i][0]=angle_debut + (i-depart)*delta_angle;
15         dataLidar[i][1]=(uint32_t)(trame[7 + j] << 8 | trame[8 + j]);
16         dataLidar[i][2]=(uint32_t)(trame[9 + j]);
17         if (i>=480) i=0; // si l'indice en cours correspond à l'angle 360°
18                         // on revient à l'indice 0 du tableau soit 0°.
19         j = j + 3;
20     }
21     mx.unlock(); // Fin exclusion mutuelle Question suivante
22 }

```

Q 31. cf lignes 10 et 20 du code précédent.

Q 32. On doit rechercher les angles correspondant à la distance la plus faible à l'obstacle sur les secteurs droits et gauches du drone.

Q 33. On doit agir sur le bloc comparateur au niveau du contrôle du cap du drone. Cela correspond à créer une perturbation de cap le temps de dévier l'obstacle.

```

1  Q 34.
2  uint32_t[] Lidar360::rechercheObstacle(int indiceMin, int indiceMax)
3  {
4      uint32_t distance, distanceMin;
5      int iMin= indiceMin;
6      uint32_t resultat[2];
7      mx.lock();
8      distanceMin= dataLidar[indiceMin][1];
9
10     for (int i=indiceMin+1; i < indiceMax; i++ )
11     { distance=dataLidar[i][0];
12         if (distance< distanceMin)
13             { distanceMin=distance;
14                 iMin=i;
15             }
16     }
17     resultat[0] = distanceMin;
18     resultat[1]= dataLidar[iMin][0];
19     mx.unlock();
20     return resultat;
21 }
22 La complexité de cet algorithme est de O(n) ou O(n-1).

```

Q 35. cf lignes 6 et 18 du code précédent.

Q 36. Pour un angle entre 0-48°, indice 0 à 48/0,75 soit l'indice 64,  
 Pour un angle entre 312-360°, indice 312/0,75 soit 416 à 360/0,75 soit l'indice 480.

Q 37. L'angle entre les 2 obstacles doit être au minimum de :  $2.l_d = d.\delta\theta$  soit  $\delta\theta = 2.l_d/d$  Autre réponse possible avec  $\sin(\delta\theta/2) = l_d/d$

```

1  Q 38.
2  float Lidar360::deviationCap(uint32_t seuil, float largeurDrone)
3  {
4      uint32_t droit[2], gauche[2];
5      float dMin, deltaTeta;
6
7      droit = rechercheObstacle(0, 64);
8      gauche = rechercheObstacle(416, 480);
9
10     // si l'obstacle est détecté à droite du drone (0-48°)
11     if ((droit[0]< seuil) && (gauche[0] > seuil)) return float(40000/droit[0]);
12
13     // si l'obstacle est détecté à gauche du drone (310-360°)
14     else if ((gauche[0]< seuil) && (droit[0]> seuil)) return float(40000/gauche[0]);
15
16     // si les obstacles sont détectés à gauche et à droite
17     else if (gauche[0]<seuil && droit[0]<seuil) {
18         //recherche distance à l'obstacle minimum
19         if (droit[0]<gauche[0]) dMin=(float)droit[0];
20         else dMin=(float)gauche[0];
21
22         // calcul de la différence d'angle entre les 2 obstacles
23         deltaTeta= droit[1]+36000-gauche[1];
24
25         // retourne l'angle médian entre les 2 obstacles
26         if (largeurDrone/dMin < deltaTeta*2*M_PI/36000) return deltaTeta/200;
27         else {
28             if droit[0]>gauche[0]
29                 return (float)(droit[1]+ (largeurDrone*36000)/( droit[1]*M_PI ));
30             else return (float)(gauche[1]+ (largeurDrone*36000)/( gauche[1]* M_PI ));
31         }
32     }
33     else return 0; // pas d'obstacles détectés
34 }

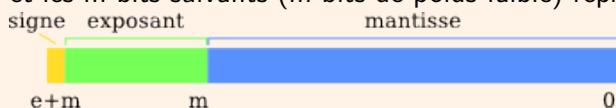
```

Ligne 1 \$GPRMC GPS  
 Q 39. Ligne 2 \$GPGGA GPS  
 Ligne 5 \$GNGLL GPS + GLONASS

Q 40. La trame \$GPGGA permet de placer le drone dans l'espace en donnant la latitude, la longitude et l'altitude.  
 \$GPGGA,173523.000,4838.4880,N,00148.8110,E,1,7,1.0,199.6,M,0.0,M,,\*5A  
 nous donne une latitude de 48° 38.4880' Nord, une longitude de 1° 48.8110' Est et une altitude de 199,6m.

Q 41. La trame \$GPGGA indique 7 satellites en vue.  
 Les deux trames \$GNGSA donnent les identifiants des 7 satellites : (19, 17, 06, 09) de la constellation GPS et (87, 71, 72) de la constellation GLONASS.

Q42. Un nombre flottant est formé de trois éléments : la mantisse  $M$  ( $m$  bits), l'exposant  $E$  ( $E$  bits) et le signe  $S$  (1 bit). Le bit de poids fort est le bit de signe : si ce bit est à 1, le nombre est négatif, et s'il est à 0, le nombre est positif. Les  $e$  bits suivants représentent l'exposant biaisé (sauf valeur spéciale), et les  $m$  bits suivants ( $m$  bits de poids faible) représentent la mantisse.



Le nombre représenté a pour valeur  $(-1)^s \times 1.M \times 2^{E-127}$

Avec  $M = \sum_{i=1}^m (m_i) * \frac{1}{2^i}$

Q43. La dynamique des nombre en virgule flottante est fixée par la taille de la mantisse.  $m = 23$  pour les float et  $m = 52$  pour les doubles. Le nombre de chiffres significatifs dépend de l'écart entre 1 et le plus petit élément représentable avec  $M$  soit  $\frac{1}{2^m}$  float :  $\frac{1}{2^{23}} = 1.19 \times 10^{-7}$ .  
double :  $\frac{1}{2^{52}} = 2.22 \times 10^{-16}$ .

Avec un float la précision de la position sur la terre est de  $(40000000/2) * 1,19 \times 10^{-8}$  2,3 m soit l'ordre de grandeur de la précision d'une mesure GPS grand public.

Avec un double la précision de la position sur la terre est de  $(40000000/2) * 2,2 \times 10^{-16}$  4,5 nm soit bien au-delà de la précision du GPS.

Lors des calculs, par exemple de moyenne, on souhaite éviter d'introduire des erreurs d'arrondis dus à la représentation en virgule flottante. L'introduction des doubles garanti de ne pas ajouter d'imprécision significative à l'imprécision du GPS.

Q44. Les algorithmes sont identiques pour la latitude, la longitude et l'altitude. On traite ici la latitude.

---

**Algorithme 1** : Moyenne mobile des 10 dernières mesures de latitude en régime permanent

---

**Data** : LAT[10] : un tableau de 10 latitudes

lat : la mesure courante

idx : l'index courant débutant à 0

sommelat : la somme courante des 10 précédentes valeurs de latitude

**Result** : Moyenne des 10 dernières mesures

1 sommelat  $\leftarrow$  sommelat - LAT[idx] + lat;

2 LAT[idx]  $\leftarrow$  lat;

3 idx  $\leftarrow$  (idx+1) modulo 10;

4 **return** sommelat/10;

---

**Algorithme 2** : Moyenne mobile des idx + 1 dernières mesures de latitude avant la 11<sup>ème</sup> mesure

---

**Data** : LAT[10] : le tableau des idx dernières latitudes dans un tableau de taille 10

lat : la mesure courante

idx : l'index courant débutant à 0

sommelat : la somme courante des idx + 1 dernières valeurs

**Result** : Moyenne des idx + 1 dernières mesures

1 sommelat  $\leftarrow$  sommelat + lat;

2 LAT[idx]  $\leftarrow$  lat;

3 idx  $\leftarrow$  (idx+1);

4 **if** idx > 9 **then**

5 | idx  $\leftarrow$  0 ;

6 | **return** sommelat/10;

7 | *Puis passage au régime permanent*

8 **else**

9 | **return** sommelat/idx;

10 **end**

---

Q45. Les algorithmes sont identiques pour la latitude, la longitude et l'altitude. On traite ici la latitude en régime permanent.

---

**Algorithme 3** : Médiane 7 dernières mesures de latitude en régime permanent

---

**Data** : LAT[7] : un tableau de 7 latitudes

ordLAT[7] : un tableau de 7 float

lat : la mesure courante

idx : l'index courant débutant à 0

**Result** : Médiane des 7 dernières mesures

1 idx  $\leftarrow$  (idx+1) modulo 7;

2 LAT[idx]  $\leftarrow$  lat;

3 OrdLAT  $\leftarrow$  Tri(LAT);

4 **return** OrdLAT[3];

---

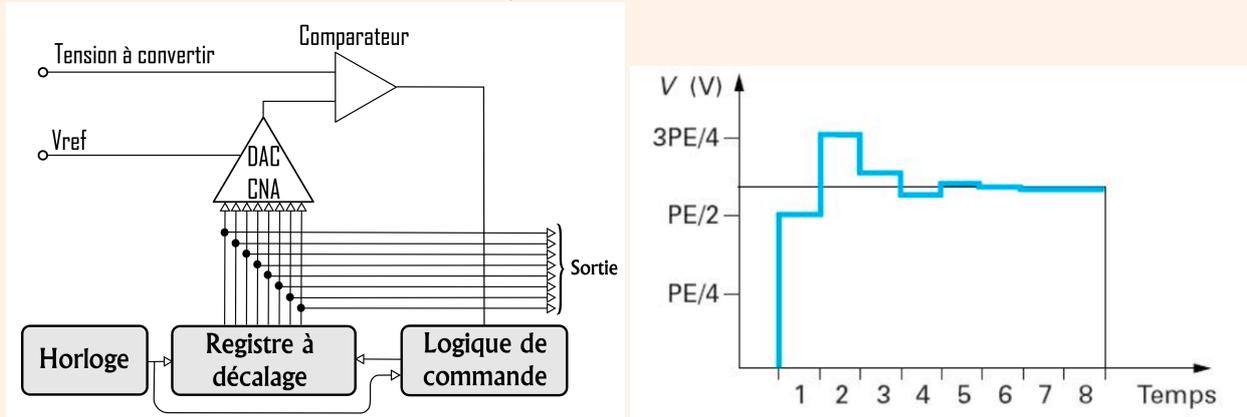
Q46. Les informations de position GPS ne sont pas suffisamment précise et répétables pour pouvoir obtenir la vitesse et l'accélération du drone dans l'espace. Comme vu dans la première partie une centrale inertielle (IMU) permet d'obtenir les données nécessaires pour le contrôle actif de la position.

Q 47. Le montage est un pont diviseur de tension suivi par un montage suiveur :

$$\frac{VMes}{VBATT} = \frac{R2}{R2 + R1} = 10/43 = 0,23$$

Q 48. Le convertisseur SAR (pour *Successive approximation register*) effectue la mesure en utilisant une processus de dichotomie.

L'ADC compare le signal à numériser avec la sortie d'un DAC commandé à partir d'un registre 12bits. En partant des poids forts et en allant vers les poids faibles, le système va reconstruire la tension à numériser en le comparant à  $V_{Ref}/2$  puis à  $V_{Ref}/2 + V_{Ref}/4$  si la tension est  $>$  à  $V_{Ref}/2$  et à  $V_{Ref}/4$  si la tension est inférieure à  $V_{Ref}/2$  en ainsi de suite. Après 12 cycles d'horloge l'ADC fournit une représentation numérique du signal à  $V_{Ref}/2^{12}$  près.



Q 49. La résolution est suffisante car la tension mesurée est précise (sans considérer le bruit) à  $3000/2^{12} = 0,73\text{mV}$ . Ramené à la tension batterie cela correspond à  $3,14\text{mV}$ .

Q 50. `uint16_t vbat = (uint16_t)((float)AD_MES * 3000) / (4096 * 0.23);`

Q 51. Nous utiliserons un pointeur sur `UID_BASE` puis `UID_BASE+4` octets puis `UID_BASE+8`

```
uint32_t uid[3];
uid[0] = *(uint32_t *)UID_BASE;
uid[1] = *(uint32_t *)(UID_BASE + 4);
uid[2] = *(uint32_t *)(UID_BASE + 8);
```

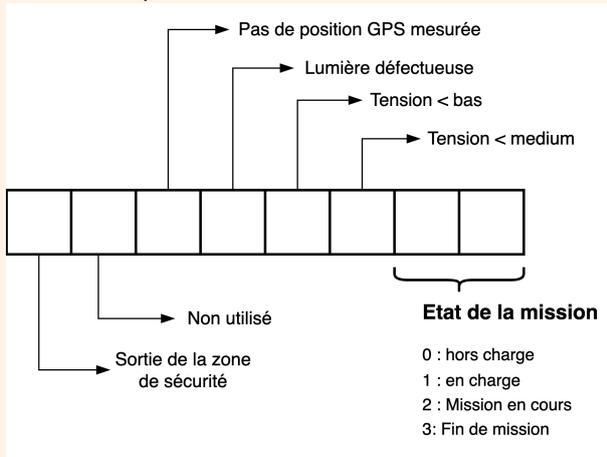
Réduire cette taille (au sens du nombre d'information) sans autre information ne permettrait pas de garantir l'unicité de l'ID.

Les solutions alternatives (et moins directes) seraient :

- L'attribution d'un autre UID au moment de la programmation : risque d'erreur, nécessité de tenir une base de données à jour coté base.
- Ne demander donner l'UID que lors du premier échange et obtenir en réponse un id pour la durée de la mission (2 octets suffisent)

Réduire la taille (au sens du stockage) est possible en recodant les données BCD ou ASCII sous une forme plus compacte mais en gardant intacte la quantité d'information.

Q 52. Composition de l'octet.



Q 53. La structure prendra la forme :

```
struct Telemetrie
{
    float latitude;
    float longitude;
    float altitude;
    uint16_t batterie;
    char etat;
    uint32_t uid[3];
}
```

Q 54. 3 float + 1 uint16\_t + 1 uint8\_t + 3 uint32\_t → 27 octets.

Oui la structure est sujette au padding : le uint16\_t ne nécessite pas d'octets de padding car il est bien aligné sur une adresse paire mais le char etat sera complété par un octet pour que l'UID débute bien à une adresse multiple de 4 octets soit la taille d'un uint32\_t. La structure prendra 28 octets en mémoire au lieu de 27.

Q 55. Un réseau zigbee va permettre d'avoir un réseau dont la portée s'étend avec le nombre de drone, chacun pouvant retransmettre les données à ses proches dans le cadre d'un réseau maillé. Organisation :

- coordinateur : la station sol
- routeurs : chaque drone.
- terminaux : aucun, les fonctionnalités étant couvertes par les routeurs.

Q 56. 0x7E 0X00 0x17

La trame débute toujours par 0x7E et les 2 octets suivants représentent la taille de la trame en hexadécimal. 0x17 = 16+7 = 23.

Q 57. Calcul de la taille de la payload :

- 12 pour la position (4 octets par float),
- 2 octets pour le niveau de batterie,
- 1 octet pour le code d'état
- 12 octets pour le UID.

Total : 27 Octets de payload de données à transférer.

Il faut ajouter :

- 17 octets pour le début de la frame start delimiter → Transmit options
- Les 27 octets de payload
- 1 pour le checksum.

Grand total : 45 Octets.

Q 58. La documentation indique : Checksum 0xFF minus the 8-bit sum of bytes from offset 3 to this byte (between length and checksum).

---

**Algorithme 4** : Calcul du checksum

---

**Data** : Mes[N]

**Result** : Mes[N] mis à jour

```
1 charsomme ← 0;
2 for i ← 3 to N - 2 do
3   | somme ← somme + Mes[i];
4 end
5 Mes[N - 1] ← 0xFF - somme;
```

---

Q 59.

```
void write_float(uint8_t Mes[], int pos, float val) {
    Mes[pos] = (val >> 24) & 0xFF;
    Mes[pos + 1] = (val >> 16) & 0xFF;
    Mes[pos + 2] = (val >> 8) & 0xFF;
    Mes[pos + 3] = (val) & 0xFF;}

void write_uint16_t(uint8_t Mes[], int pos, uint16_t val) {
    Mes[pos] = (val >> 8) & 0xFF;
    Mes[pos + 1] = (val) & 0xFF;}

void write_uint32_t(uint8_t Mes[], int pos, uint32_t val) {
    Mes[pos] = (val >> 24) & 0xFF;
    Mes[pos + 1] = (val >> 16) & 0xFF;
    Mes[pos + 2] = (val >> 8) & 0xFF;
    Mes[pos + 3] = (val) & 0xFF;}

void createFrameFromStruct(
    uint8_t Mes[],
    struct Telemetrie tel,
    uint64_t destAddress64,
    uint16_t destAddress16
) {
    Mes[0] = 0x7E;
    write_uint16_t(Mes, 1, 27 + 17); // Message + structure
    Mes[3] = 0x10; // Transmit Request
    Mes[4] = 0x00; // 0x01 to ask for an answer and 0x00 for no answer
    Mes[5] = (destAddress64 >> 7*8) & 0xFF;
    Mes[6] = (destAddress64 >> 6*8) & 0xFF;
    Mes[7] = (destAddress64 >> 5*8) & 0xFF;
    Mes[8] = (destAddress64 >> 4*8) & 0xFF;
    Mes[9] = (destAddress64 >> 3*8) & 0xFF;
    Mes[10] = (destAddress64 >> 2*8) & 0xFF;
    Mes[11] = (destAddress64 >> 1*8) & 0xFF;
    Mes[12] = (destAddress64) & 0xFF;
    write_uint16_t(Mes, 13, destAddress16);
    Mes[15] = 0;
    Mes[16] = 0;
    write_float(Mes, 17, tel.latitude);
    write_float(Mes, 21, tel.longitude);
    write_float(Mes, 25, tel.altitude);
    write_uint16_t(Mes, 29, tel.batterie);
    Mes[31] = tel.etat;
    write_uint32_t(Mes, 32, tel.uid[0]);
    write_uint32_t(Mes, 36, tel.uid[1]);
    write_uint32_t(Mes, 40, tel.uid[2]);
    uint8_t Cksum = 0x00;
    for (uint16_t i = 3; i < 44; i++) Cksum += Mes[i];
    Mes[44] = 0xFF - Cksum;
}
```

Q 60. Une attaque par rejeu (en anglais, replay attack ou playback attack) est une forme d'attaque réseau dans laquelle une transmission est malicieusement répétée par un attaquant qui a intercepté la transmission. L'attaquant renvoie simplement un message déjà envoyé.

Q 61. Dans le cadre de notre application, les dangers d'une attaque par rejeu sont :

- De pouvoir faire croire qu'un drone appartient toujours à l'évènement alors qu'il a été désactivé. C'est un risque associé à du vol.
- De saturer le réseau de télémétrie en envoyant des messages qui semblent légitimes en quantité trop grande Déni de service (DOS)

Il n'y a pas de risque d'agir négativement sur l'évènement.

<i>EcouteMQTT</i>
+ client
+ topics
+ broker_address
+ port
+ t
+ messages
+ __init__()
+ on_connect()
+ on_message()
+ start()
+ stop()

Q 62.

Q 63. `EcouteMQTT("b53z91.etincelle.fx", 1883, "topics/position")`

Q 64. C'est le DNS (Domain Name Server). Comme la liste des sous-domaines est publique dans un DNS, la sécurité apportée est faible. Elle empêche juste l'attaque opportuniste de quelqu'un cherchant tous les sous domaines avec le mot mqtt, mais pas de quelqu'un qui sais que cette société dispose d'un serveur mqtt. Une sécurisation par mot de passe, un changement de port par défaut et un cryptage des données sera une solution plus efficace.

<i>SQLiteManager</i>
+ conn
+ cursor
+ create_rq
+ insert_rq
+ update_rq
+ cleanup_rq
+ __init__()
+ execute()
+ insert_or_update()
+ affiche()
+ close()

Q 65.

- Q 66.   ▪ CREATE TABLE : création d'une table
- IF NOT EXISTS : si elle n'existe pas déjà
  - drones : nom de la table
  - id\_drones CHAR 12 : pour l'UID sur 12 octets
  - NOT NULL : ne peut pas être nul sans créer une violation
  - PRIMARY KEY : c'est la clef primaire - il ne peut pas y en avoir 2 identiques sans créer une violation
  - date\_enregistrement text NOT NULL : un texte pour la date qui ne peut pas être nul sans créer une violation
  - 3 réels (format flottant) pour lat, long et alt

Q 67. Cette méthode s'appuie sur un mécanisme d'exception. Si l'insertion échoue alors on fait une mise à jour. L'insertion échoue ici quand l'ID est déjà présent (clef primaire). L'insertion échoue aussi en cas d'erreur de syntaxe mais ce cas est traité par le développeur pendant la conception.

```
Q 68. requete = f"SELECT id_drone, lat, long, alt FROM drones
        WHERE date_enregistrement < '{iso_limite}'"
```

Q 69.

```
# Au début du programme
client = mqtt.Client()
broker_address = "b53z91.etincelle.fx"
broker_ports = 1883
client.connect(broker_address, broker_ports)

# Ligne 41
for drone in drones_perdus:
    message = drone[0] + ":" +
               str(drone[1]) + ":" + str(drone[2]) + ":" + str(drone[3])
    client.publish("topic/alerte", message, 0)
```

Q 70. Deux threads différents accèdent potentiellement en même temps à messages. on\_message ajoute les messages à la liste messages lors de l'arrivée d'un message mqtt (asynchrone par rapport au thread principal). Dans le même temps le thread principal supprime l'ensemble des messages lorsqu'il a fini de traiter sa liste. Si un message arrive dans le (très bref) laps de temps entre le dernier test et l'effacement des messages alors ce message est perdu. La proportion de ces événements augmente avec la fréquence des messages mqtt.

Q 71. A la ligne 22 de `ecoute_mqtt.py` (executé par le thread `t`) :

- Attendre tant que `self.verrou = True`
- Puis positionner `self.verrou` à `True`
- Réaliser l'ajout du message
- Puis repositionner `self.verrou` à `False`

A la ligne 18 de `mqtt\_monitor.py`, avant la séquence d'analyse des messages :

- Attendre tant que `self.verrou = True`
- Puis positionner `self.verrou` à `True`
- Réaliser le traitement des messages
- Puis repositionner `self.verrou` à `False`

Q 72. Python propose un mécanisme de queues :

Le module `queue` implémente des files multi-productrices et multi-consommatrices. C'est particulièrement utile en programmation multi-thread, lorsque les informations doivent être échangées sans risques entre plusieurs threads. La classe `Queue` de ce module implémente tout le verrouillage nécessaire.

# Épreuve d'admission d'exploitation pédagogique d'une activité pratique relative à l'approche globale d'un système pluritechnologique

## A. Présentation de l'épreuve

### Textes de référence

<http://www.devenirenseignant.gouv.fr/cid98734/les-epreuves-de-l-agregation-externe-section-sciences-industrielles-de-l-ingenieur.html>

[Arrêté du 24 juin 2019 modifiant l'arrêté du 28 décembre 2009 fixant les sections et les modalités d'organisation des concours de l'agrégation](#)

- Durée totale 6 heures (activités pratiques 4 heures, préparation de l'exposé 1 heure, exposé 30 minutes maximum, entretien 30 minutes).
- Coefficient 2.
- 10 points sont attribués à la partie liée aux activités pratiques et 10 points à la partie liée à l'exposé et à l'entretien avec le jury.

L'épreuve fait appel à des connaissances technologiques et scientifiques communes à l'ensemble des options.

Le candidat est amené au cours de cette épreuve à élaborer **une séquence pédagogique dont le contexte est imposé**. Il doit notamment y intégrer le développement d'une séance à caractère expérimental. Elle sera construite autour d'activités pratiques proposées par le candidat, sur un support didactique imposé.

L'exploitation pédagogique demandée est relative aux enseignements non spécifiques de la spécialité ingénierie, innovation et développement durable du cycle terminal sciences et technologies de l'industrie et du développement durable (STI2D) ou de l'enseignement des sciences de l'ingénieur du lycée général et des classes préparatoires aux grandes écoles.

Le support didactique fourni est un système pluritechnologique qui permet une analyse systémique globale.

Au cours de l'entretien, le candidat est conduit à préciser certains points de sa présentation. Il est amené à expliquer et justifier les choix didactique et pédagogique qu'il a opérés notamment dans l'élaboration de la séquence de formation présentée ainsi que pour les contenus de la séance à caractère expérimental conçue.

### Déroulement de l'épreuve

L'objectif de la première épreuve d'admission de l'agrégation de SII est de permettre d'évaluer chez les candidats leurs compétences pour s'inscrire dans la démarche d'un agrégé de sciences industrielles de l'ingénieur capable d'élaborer une exploitation pédagogique à partir d'une activité pratique relative à l'approche globale d'un système pluritechnologique.

Le titre d'une séquence pédagogique étant imposé, le candidat doit d'abord préparer la trame détaillée de celle-ci en respectant le niveau de formation visé et les effectifs de la classe. Il doit ensuite répondre à une problématique technique et scientifique comprenant des activités pratiques. Le candidat prépare enfin une séance à caractère expérimental s'inscrivant dans la séquence imposée. **Les activités expérimentales proposées doivent être différentes de celles déjà effectuées pour répondre à la**

**problématique technique et scientifique et être adaptées au niveau de formation visé précisé dans l'énoncé du sujet.**

Les compétences attendues par le jury sont pédagogiques, comportementales et scientifiques.

Le candidat doit montrer ses aptitudes à :

- concevoir, organiser et décrire une séquence dans un contexte pédagogique imposé ;
- s'approprier un système réel ou un équipement et son environnement ;
- élaborer, justifier, conduire et exploiter un protocole expérimental ;
- analyser le comportement d'un système à partir d'un modèle ;
- maîtriser, conduire et exploiter une simulation numérique ;
- formuler des conclusions pour choisir et décider ;
- mener des démarches avec rigueur et évoluer avec autonomie.

L'évaluation du candidat s'effectue en trois phases.

### **Phase 1 – Conception et organisation d'une séquence de formation à un niveau imposé (durée 4h00)**

Cette première phase d'une durée totale de 4h00 compte quatre parties.

Elle se déroule dans un laboratoire où sont mis à disposition du candidat un support d'étude, un environnement numérique de travail connecté à Internet, des moyens de mesure ou de simulation et si besoin des logiciels spécifiques d'acquisition.

#### ➤ **Première partie (durée 0h45) – Réflexions pédagogiques sur la séquence imposée**

Pour cette première partie, le candidat doit réfléchir et proposer une séquence de formation parmi deux qui lui sont proposées. Pour chacune d'entre elles, le contexte pédagogique est imposé. Ce dernier est composé :

- du titre de la séquence imposée ;
- du niveau de formation visé ;
- d'une proposition de progression didactique liée à la formation visée ;
- du programme du niveau de formation visé ;
- d'une liste non exhaustive de supports matériels pédagogiques d'un laboratoire de Sciences de l'Ingénieur.

Le candidat doit recenser les compétences à développer, en intégrant les savoir-faire et savoirs du programme du niveau imposé en lien avec le titre d'une des deux séquences proposées au choix. Puis il doit proposer une trame détaillée de celle-ci (activités, durée, coordination). Les pré-requis de la séquence doivent être identifiés vis-à-vis de la progression didactique proposée et présentée. Le candidat doit justifier ses choix pédagogiques et didactiques (TP, TD, cours, projet...). L'ensemble de ces éléments doit être rédigé sur un support de présentation numérique, qui sera présenté et évalué lors de la troisième phase.

#### ➤ **Deuxième partie (durée 0h30) – Prise en main du support**

Pour cette deuxième partie, les manipulations proposées ont pour objectif de faciliter la compréhension du fonctionnement global du système. À la fin de cette première partie, l'examineur s'assure que le candidat s'est bien approprié le support de TP. L'objectif de cette partie est de faire émerger une problématique technique et scientifique à résoudre.

#### ➤ **Troisième partie (durée 2h00) – Expérimentations pour répondre à une problématique technique et scientifique imposée**

Pour cette partie, le candidat doit répondre aux activités à caractère expérimental proposées afin de résoudre la problématique technique et scientifique, par la mobilisation de compétences caractéristiques du niveau de l'agrégation, sur un spectre large permettant une approche pluritechnologique. L'exploitation des résultats obtenus (hypothèses, modèles, résultats expérimentaux, valeurs numériques...), la mise en évidence des écarts entre les performances souhaitées, les performances mesurées et les performances simulées et la proposition de solutions pour les réduire doivent permettre d'apporter une réponse aux problèmes posés.

➤ **Quatrième partie (durée 0h45) – Élaboration du scénario d'une séance à caractère expérimental**

Pour cette quatrième partie, le candidat doit décrire une séance à caractère expérimental s'insérant dans la séquence pédagogique en :

- situant la séance à caractère expérimental dans sa proposition de séquence pédagogique ;
- précisant l'organisation matérielle et pédagogique de la séance (nombre d'élèves, systèmes utilisés, travail en îlots,...) ;
- décrivant la (ou les) démarche(s) pédagogique(s) retenue(s) (démarche d'investigation, de résolution de problème technique, de projet,...) ;
- détaillant le scénario des activités que doivent réaliser les élèves ;
- proposant et en mettant en œuvre au moins un protocole expérimental différent de ceux qu'il a effectués dans la troisième partie ;
- explicitant clairement l'apport de la séance proposée dans le développement des compétences des élèves.

Pendant toute la durée de cette partie, le candidat a accès aux logiciels de simulation, au système et aux matériels de travaux pratiques. Le candidat doit donc entreprendre de réaliser de nouvelles simulations ou expérimentations utiles pour étayer et créer la trame de sa séance. Les examinateurs n'évaluent pas durant cette partie mais sont disponibles en tant qu'assistant technique.

**Phase 2 – Préparation de l'exposé (durée 1h00)**

Le candidat prépare son intervention devant le jury permanent en complétant son support de présentation numérique. Le candidat n'a plus accès au matériel de travaux pratiques, c'est-à-dire, ni au système, ni aux modèles associés, ni aux logiciels de simulation, mais conserve à sa disposition l'ensemble des ressources associées au sujet. Il dispose d'un poste informatique connecté à Internet et doté des logiciels courants de bureautique, et des résultats obtenus lors de la phase précédente qu'il aura stockés dans un espace dédié sur un serveur.

**Phase 3 – Exposé oral et entretien avec le jury en salle (durée 1h00)**

Le candidat a à sa disposition un tableau, un ordinateur et un vidéoprojecteur pour la présentation devant le jury.

L'exposé du candidat devant le jury a une durée de 30 minutes maximum sans intervention du jury.

L'exposé doit comporter :

- la description du contexte pédagogique imposé ;
- la présentation de ses réflexions pédagogiques et la justification de ses choix de modalités pédagogiques ;
- la présentation de la trame de la séquence pédagogique en y intégrant l'évaluation ;
- la présentation des savoir-faire et savoirs à transmettre dans chaque séance ;
- la justification de la pertinence du support didactisé dans un contexte pédagogique (durée maximale 5 minutes) ;
- la démarche mise en œuvre dans la séance à caractère expérimental ;

- la présentation d'une ou des activités que devraient mener les élèves durant la séance d'activités à caractère expérimental ;
- la présentation de la valeur ajoutée pédagogique dans la formation de la séance proposée.

Il est à noter que durant la présentation des travaux devant le jury, il n'est absolument pas attendu des candidats qu'ils présentent à nouveau les résultats aux activités menées dans le cadre des deuxième et troisième parties de la phase 1. En effet, ceux-ci ont déjà conduit à une évaluation par le jury en salle de TP. Seule est attendue la présentation des activités envisagées de faire réaliser aux élèves lors de la séance à caractère expérimentale incluse dans la séquence pédagogique exposée. Néanmoins les résultats expérimentaux ou de simulation numérique peuvent être utilisés afin d'illustrer la séquence ou la séance expérimentale.

L'exposé du candidat est suivi d'un entretien avec le jury d'une durée de 30 minutes.

Le jury est amené à interroger les candidats, afin d'apprécier leur connaissance des principes fondamentaux du système éducatif et du cadre réglementaire de l'école, sur la manière dont ils envisagent d'accompagner les élèves dans leur parcours de formation, ou bien sur leur positionnement au sein de la communauté éducative.

Au cours de l'entretien, les candidats sont amenés à :

- préciser certains points de leurs présentations ;
- expliciter et justifier les choix de nature didactique et/ou pédagogique qu'ils ont opérés ;
- prendre en compte les propositions alternatives faites par le jury.

### **Utilisation des logiciels pendant l'interrogation**

Aucun pré-requis ne peut être exigé du candidat concernant l'utilisation d'un logiciel. Les consignes d'utilisation sont indiquées dans le sujet ou fournies oralement. Les modeleurs volumiques ne sont pas utilisés comme outil de conception de formes mais comme un outil de lecture de documents.

Les supports retenus lors de la session 2025 étaient les suivants :

- volet roulant ;
- système de travelling ;
- imprimante 3D ;
- axe linéaire didactisé ;
- maquette didactisée d'un bras de pelleuse ;
- ventilation mécanique contrôlée double flux ;
- exosquelette contre le mal de dos ;
- attelle de remobilisation du genou.

Ces supports ont permis aux candidats de mettre en œuvre leurs compétences à haut niveau scientifique sur les activités suivantes :

- élaboration et mise en œuvre d'un protocole expérimental ;
- identification des comportements de constituants ou d'un système ;
- mesure de comportement de constituants ou d'un système ;
- détermination des paramètres significatifs d'une chaîne de mesure ;
- détermination des paramètres significatifs d'une chaîne d'information ;
- détermination des paramètres significatifs d'une chaîne de puissance ;
- détermination des paramètres significatifs d'une modélisation ;
- analyse d'algorithmes simples ou de quelques lignes de programmes simples (en langage python, arduino, etc) ;

- recalage d'un modèle multiphysique ou non ;
- choix des modèles de comportement ou de connaissance ;
- validation de modèles ;
- simulation et prédiction de performance ;
- évaluation des écarts ;
- analogie entre différents domaines.

## **B. Commentaires du jury**

### **• Analyse des résultats**

**Les candidats préparés** mobilisent à bon escient leurs compétences pour répondre à la problématique pédagogique demandée. Le déroulement de la séquence pédagogique est structuré et cohérent. Ils positionnent convenablement la séance à caractère expérimental en présentant de nouvelles activités pratiques qu'ils ont réalisées durant la quatrième partie de la première phase. Les pré-requis, les objectifs, les démarches pédagogiques et d'évaluation sont bien assimilés et correctement décrits lors de l'exposé oral. Ces candidats ont généralement produit une présentation orale de qualité. La conduite des expérimentations pour répondre à la problématique technique et scientifique est traitée par la majorité des candidats.

Certains candidats présentent une séquence pédagogique qui ne respecte pas le contexte imposé, se plaçant ainsi hors sujet. Il n'est pas rare que des candidats ne se réfèrent pas du tout aux programmes en vigueur. Lors de l'exposé oral, quelques candidats présentent le système et les résultats obtenus pendant la troisième partie de la première phase, or ce ne sont pas les attendus de l'épreuve. De même, les activités pratiques réalisées pendant la troisième partie de la première phase sont souvent reprises dans la séance, alors que le jury en attend de nouvelles. L'explication de la pertinence du système, dans le cadre de la séance expérimentale proposée est souvent oubliée.

Quelques rares candidats ne présentent pas du tout de séquence pédagogique, et se contentent de rappeler les résultats obtenus lors des activités pratiques. Un tel décalage avec les attendus de l'épreuve surprend au regard de l'exigence du concours de l'agrégation.

### **• Commentaires sur les réponses apportées et conseils aux futurs candidats**

#### **Phase 1 – Première partie : réflexions pédagogiques sur la séquence imposée**

Le jury constate que :

- les déroulements des séquences sont souvent imprécis et peu approfondis ;
- les compétences et connaissances visées sont peu ciblées, voire non identifiées ;
- le choix des stratégies pédagogiques mises en œuvre est rarement pertinent et justifié ;
- l'évaluation et la remédiation sont souvent absentes de la séquence.

Le jury attend une séquence pédagogique structurée en lien avec une thématique sociétale. Elle doit faire apparaître les pré-requis, les compétences et connaissances associées, le positionnement temporel, le déroulement des différentes séances la constituant et l'évaluation adéquate.

Il est proposé au candidat le choix entre deux séquences pédagogiques associant des compétences différentes d'un même niveau. Ces deux séquences sont de difficulté équivalente. Cette possibilité

laissée au candidat est bien appréhendée. Pour tous les sujets, l'une et l'autre des propositions ont été traitées.

Les outils et méthodes de l'ingénierie pédagogique doivent être connus et maîtrisés. Le jury ne peut se satisfaire d'un exposé de pédagogie formel ou d'une récitation d'un extrait de programme. Il souhaite qu'il soit fait preuve d'imagination et de créativité dans le contenu pédagogique présenté afin de susciter l'intérêt et la motivation des élèves.

Pour les futures sessions, le jury conseille aux candidats d'étudier préalablement et attentivement les programmes et les objectifs des formations dont peuvent être issus les contextes pédagogiques imposés : enseignement de spécialité « sciences de l'ingénieur », enseignements technologiques de spécialités du cycle terminal STI2D et enseignement « sciences de l'ingénieur » des CPGE. Cette étude, ainsi que la lecture des documents « ressources pour faire la classe » et des guides d'équipement, leur permettront de proposer une exploitation pédagogique en adéquation avec le niveau imposé. Une réflexion pédagogique sur les objectifs de formation de ces séries et classes post-bac est indispensable pour réussir cette partie de l'épreuve.

Le jury engage les candidats à clairement indiquer la ou les démarches pédagogiques qui structureront l'organisation pédagogique retenue (démarche d'investigation, démarche de résolution de problème technique, démarche scientifique ou encore démarche de projet technologique).

#### **Phase 1 – Deuxième partie : prise en main du support**

Pour cette partie, les manipulations ainsi que les activités proposées ont pour objectif de faciliter la compréhension du fonctionnement global du système, de s'approprier le support du travail pratique et la problématique technique et scientifique proposée. Les candidats disposent d'un dossier technique, d'un dossier ressource, ainsi que diverses ressources numériques. Le système proposé au candidat peut être le système réel ou un système didactisé.

Les manipulations proposées sont très guidées de sorte que le candidat peut rapidement appréhender l'environnement logiciel et matériel du support. Certains candidats se trouvent en difficulté dès cette phase de prise en main. Une lecture attentive et le respect des consignes fournies doit permettre d'avancer efficacement dans les activités de prise en main proposées. Le jury recommande aux candidats de se confronter plus régulièrement à la manipulation de systèmes réels et/ou didactisés.

#### **Phase 1 – Troisième partie : expérimentations pour répondre à une problématique technique et scientifique imposée**

Pour cette phase, le jury tient à porter à l'attention des candidats les points suivants :

- la maîtrise du raisonnement scientifique et la caractérisation des échanges d'énergie, de matière et d'information à un niveau de généralités permettent de s'adapter à une large diversité de systèmes ;
- l'extraction des informations pertinentes dans les ressources mises à disposition constitue un préalable indispensable à l'appropriation du système et de la problématique ;
- la problématique scientifique et technique doit être comprise afin d'y répondre. Elle permet d'appréhender correctement le fil directeur des activités et manipulations proposées ;
- les analyses externes et internes des systèmes gagnent en pertinence lorsqu'elles s'appuient sur des outils formalisés (schéma des chaînes de puissance et d'information, diagrammes SysML) ;
- une bonne culture personnelle pluritechnologique, fondée sur l'observation et l'analyse de systèmes variés et modernes, est indispensable.

Le candidat est amené à :

- utiliser une instrumentation spécifique dédiée à la mesure de grandeurs physiques sur les systèmes instrumentés ;
- mettre en œuvre différents outils informatiques (logiciels de pilotage et/ou d'acquisition dédiés aux supports, logiciels de simulation, modéleur, logiciel de calculs par éléments finis, tableurs, traitements de textes, logiciels de calcul ou de visualisation, environnements de programmation...).

Le jury assiste le candidat en cas de difficultés matérielles ou de mise en œuvre des différents outils informatiques. La maîtrise de ces logiciels n'est pas exigée.

Lors des activités pratiques, le jury souhaite que les candidats s'attachent à :

- lire et analyser l'ensemble du sujet proposé ;
- maîtriser la durée consacrée à chaque activité ;
- maîtriser les outils d'analyse courants (structurels, fonctionnels et comportementaux) ;
- exploiter et interpréter l'ensemble des résultats des expérimentations et des mesures dans leur totalité et de façon rigoureuse ;
- suivre rigoureusement les protocoles expérimentaux qui leur sont proposés ;
- corréler les résultats des simulations et des expérimentations en les associant à des phénomènes physiques et à des solutions technologiques ;
- effectuer une analyse critique des résultats expérimentaux ;
- vérifier la cohérence et la pertinence des résultats expérimentaux ;
- mettre en œuvre une démarche de résolution du problème technique et scientifique posé ;
- réfléchir à de nouvelles activités pratiques pouvant nourrir la séance expérimentale qui sera présentée.

Le jury précise que les supports de travaux pratiques sont principalement issus des laboratoires de SI, STI2D et CPGE et couvrent l'ensemble des champs disciplinaires transversaux des sciences industrielles de l'ingénieur.

### **Phase 1 – Quatrième partie : élaboration du scénario d'une séance à caractère expérimental**

Le jury constate que :

- cette partie est la plus délicate pour un grand nombre de candidats ;
- la séance à caractère expérimental n'intègre pas toujours des activités pratiques différentes de celles réalisées lors de la troisième partie de la première phase ;
- les activités proposées aux élèves sont peu détaillées. Des illustrations intégrant des captures d'écran montrant les résultats obtenus permettent de rendre plus concrètes les manipulations proposées.

Le candidat doit développer une séance expérimentale contextualisée, positionnée avec pertinence dans la séquence pédagogique proposée. Il est attendu la description des activités proposées aux élèves. La configuration pédagogique choisie doit être justifiée.

Les manipulations et protocoles de mesures insérés dans la séance doivent être adaptés au niveau requis. Ils doivent être différents de ceux réalisés lors de la troisième partie « expérimentations » tout en utilisant le système de l'épreuve. Des modalités d'évaluation doivent également être présentées et justifiées pédagogiquement. Les éventuelles remédiation doivent être précisées.

### **Phase 3 – Exposé oral et entretien avec le jury en salle**

La majorité des candidats n'utilise pas le temps imparti pour la présentation de 30 minutes. L'utilisation de la durée prévue leur permettrait de préciser leurs réflexions pédagogiques.

Il n'est pas attendu des candidats la présentation des activités réalisées lors de la première phase de cette épreuve, déjà évaluées au cours de la première phase.

Le jury attend lors de cette phase de présentation de la séquence pédagogique que le candidat soit capable de :

- présenter le contexte pédagogique imposé ;
- situer la séquence de formation en l'inscrivant dans la formation au niveau requis ;
- expliciter les connaissances et les compétences visées par la séquence ;
- décrire le déroulement de la séquence ;
- situer la séance expérimentale dans la séquence pédagogique ;
- expliciter les connaissances et compétences visées dans la séance ;
- présenter la pertinence du système pour les activités pratiques de la séance ;
- définir l'enchaînement des activités réalisées par les élèves dans la séance ainsi que les résultats attendus ;
- justifier les choix pédagogiques retenus ;
- présenter les moyens de l'évaluation des connaissances et compétences acquises par les élèves ou étudiants.

Le jury est amené à proposer des mises en situation permettant de vérifier la capacité du candidat à prendre en compte les conditions d'exercice de son métier futur (travail en équipe pédagogique, personnels ressources dans les établissements, construction des parcours des élèves et étudiants, etc.).

Le jury attend également du candidat qu'il mette en œuvre des compétences professionnelles telles que :

- produire un discours clair, précis et rigoureux en sachant attirer l'attention du jury ;
- être pertinent et réactif aux questions posées ;
- être capable de dégager l'essentiel, de donner du sens aux connaissances développées et de captiver l'auditoire.

Le jury conseille aux candidats qui n'en auraient pas eu la possibilité au cours de leur formation, de prendre contact avec un établissement scolaire dispensant les filières de formation visées par le concours et de s'y déplacer afin de prendre connaissance des réalités matérielles, humaines et organisationnelles du terrain.

### **Comportement des candidats**

Les candidats doivent être méthodiques et rigoureux pour appréhender un système pluritechnologique dans sa globalité et dans sa complexité. L'exploitation pédagogique d'une activité pratique relative à l'approche globale et transversale d'un système pluritechnologique ne s'improvise pas. Elle doit se préparer tout au long des formations conduisant à l'agrégation. Les candidats doivent éviter les présentations stéréotypées ne permettant pas de mettre en valeur la qualité de leur réflexion personnelle. Les contenus scientifiques des séquences doivent être maîtrisés par les candidats, l'accès à Internet étant toujours possible sur des sites publics. Quelques candidats ont fait le choix de s'appuyer sur des propositions émanant d'intelligences artificielles génératives. Cet étayage n'apporte pas de plus-value dans la mesure où il n'est pas accompagné des justifications appropriées quant aux choix pédagogiques et didactiques opérés.

Le jury apprécie lorsque les diapositives sont numérotées lors de la présentation orale : cela lui permet de se référer plus facilement aux éléments de présentation. Le jury déplore les fautes d'orthographe dans les supports numériques présentés ; elles nuisent à l'image du candidat. Les candidats se présentant au concours de l'agrégation se destinent à être de futurs cadres de l'éducation nationale. Ils

se doivent d'avoir un vocabulaire, un comportement et une tenue en adéquation avec le métier d'enseignant.

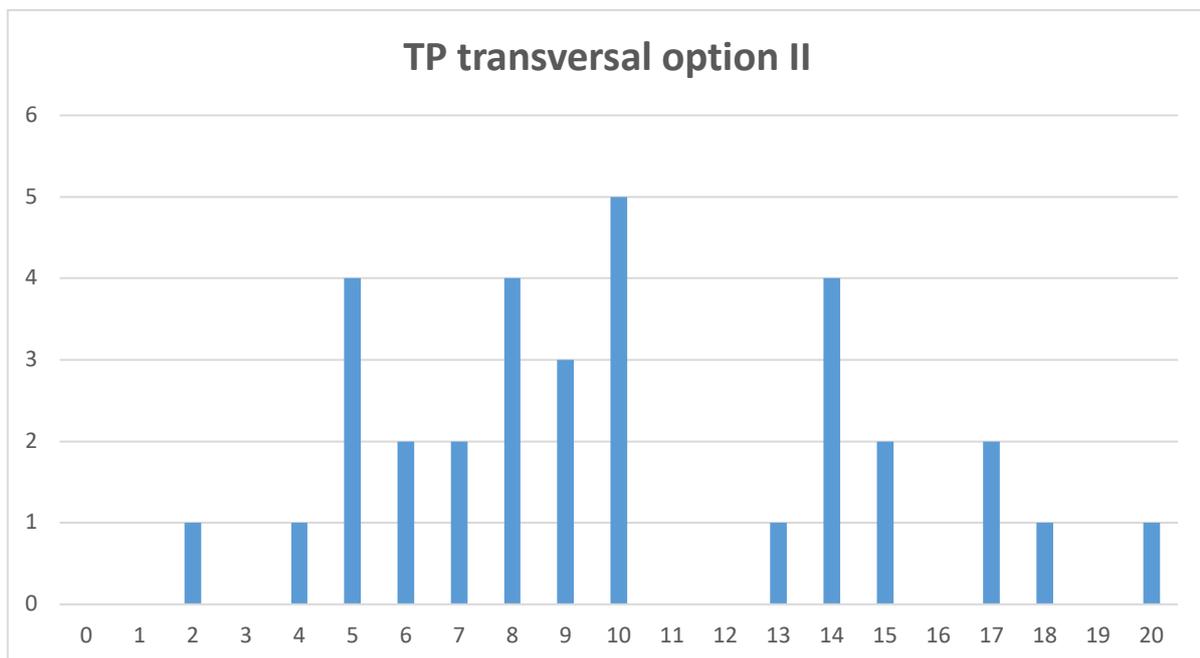
## Conclusion

L'augmentation du nombre de candidats présentant un niveau très insuffisant révèle leur impréparation à une épreuve qui est exigeante. Les candidats en capacité de mener à la fois des expérimentations rigoureuses et une réflexion pédagogique approfondie ont été très peu nombreux.

La session 2025 de l'agrégation externe SII confirme que l'usage d'un laboratoire unique, composé de systèmes pluritechnologiques, commun aux quatre options de l'agrégation de sciences industrielles de l'ingénieur, impose aux candidats de s'appropriier tous les champs disciplinaires transversaux liés au triptyque « matière, énergie et information ». Au-delà même de la juxtaposition des savoirs pluridisciplinaires indispensables pour réussir cette épreuve, cette session met aussi en évidence toute l'importance, pour un candidat, d'être réellement apte à appréhender les systèmes dans leur globalité. Enfin, les compétences en ingénierie pédagogique attendues lui imposent une connaissance approfondie des différents programmes, des objectifs de formation associés et des stratégies pédagogiques préconisées.

## C. Résultats

La moyenne des notes obtenues est de 10,0/20. L'écart-type est de 4,6. La meilleure note est 19,9. La plus faible est 1,6. La médiane est de 9,3.



# Exemple de sujet pour l'exploitation pédagogique d'une activité pratique relative à l'approche globale d'un système pluritechnologique

## Volet roulant solaire

**Phase 1 – Conception et organisation d'une séquence de formation à un niveau imposé (durée : 4h00)**

**Partie 1.1 – Conception de l'architecture de la séquence de formation imposée (0h45)**

Objectifs : s'approprier le besoin pédagogique imposé par le jury et concevoir l'architecture de la séquence de formation.

### Contexte pédagogique de la séquence de formation imposée

La séquence pédagogique à construire est associée à un des deux contextes pédagogiques suivants, à choisir par le candidat :

	Choix 1	Choix 2		
<b>Titre de la séquence</b>	Comment les nouvelles technologies peuvent-elles améliorer l'habitat ?	Comment optimiser l'utilisation de l'énergie de façon autonome tout en respectant l'environnement ?		
<b>Niveau de formation visé</b>	STI2D enseignements communs de 2I2D en Terminale			
<b>Supports pédagogiques</b>	Les supports suivants sont disponibles dans le laboratoire de sciences de l'ingénieur. Ils sont choisis judicieusement pour répondre au besoin pédagogique de la séquence imposée : <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;"> <ul style="list-style-type: none"> <li>- ventilation mécanique contrôlée double-flux ;</li> <li>- système de climatisation ;</li> <li>- panneaux photovoltaïques ;</li> <li>- volet roulant solaire autonome ;</li> <li>- système Eolicc ;</li> </ul> </td> <td style="width: 50%; border: none;"> <ul style="list-style-type: none"> <li>- systèmes variés de chauffage ;</li> <li>- système d'éclairage communicant de salles de spectacles ;</li> <li>- thermostat connecté et intelligent ;</li> <li>- pompe à chaleur ;</li> <li>- ...</li> </ul> </td> </tr> </table>		<ul style="list-style-type: none"> <li>- ventilation mécanique contrôlée double-flux ;</li> <li>- système de climatisation ;</li> <li>- panneaux photovoltaïques ;</li> <li>- volet roulant solaire autonome ;</li> <li>- système Eolicc ;</li> </ul>	<ul style="list-style-type: none"> <li>- systèmes variés de chauffage ;</li> <li>- système d'éclairage communicant de salles de spectacles ;</li> <li>- thermostat connecté et intelligent ;</li> <li>- pompe à chaleur ;</li> <li>- ...</li> </ul>
<ul style="list-style-type: none"> <li>- ventilation mécanique contrôlée double-flux ;</li> <li>- système de climatisation ;</li> <li>- panneaux photovoltaïques ;</li> <li>- volet roulant solaire autonome ;</li> <li>- système Eolicc ;</li> </ul>	<ul style="list-style-type: none"> <li>- systèmes variés de chauffage ;</li> <li>- système d'éclairage communicant de salles de spectacles ;</li> <li>- thermostat connecté et intelligent ;</li> <li>- pompe à chaleur ;</li> <li>- ...</li> </ul>			
<b>Effectif</b>	Classe de 24 élèves, groupe à effectif réduit de 12 élèves			
<b>Volume horaire</b>	12 heures hebdomadaires (2h à effectif complet + 10h à effectif réduit)			

Les documents suivants sont fournis et accessibles dans le dossier « contexte pédagogique » :

- le programme du niveau de formation visé (fichier Programme IT-I2D-2I2D.pdf) ;
- des documents d'accompagnement (dossier Ressources Pédagogiques STI2D) ;
- une proposition de séquences adaptées au niveau de formation visé (fichier Progression didactique STI2D.xlsx).

### Production attendue

Une architecture de séquence pédagogique doit être proposée en s'assurant de la cohérence, de la faisabilité et de la pertinence des choix effectués après avoir :

- contextualisé la séquence pédagogique dans une grande thématique ;
- recensé les compétences à développer et les savoir-faire et savoirs à faire acquérir aux élèves ;

- identifié les prérequis et le positionnement temporel de la séquence dans une progression pédagogique (vis-à-vis de la proposition de liste de séquences fournie) ;
- spécifié les modalités pédagogique et didactique (TP, TD, cours, projet, évaluation, remédiation, ...), leurs coordinations et leurs organisations.

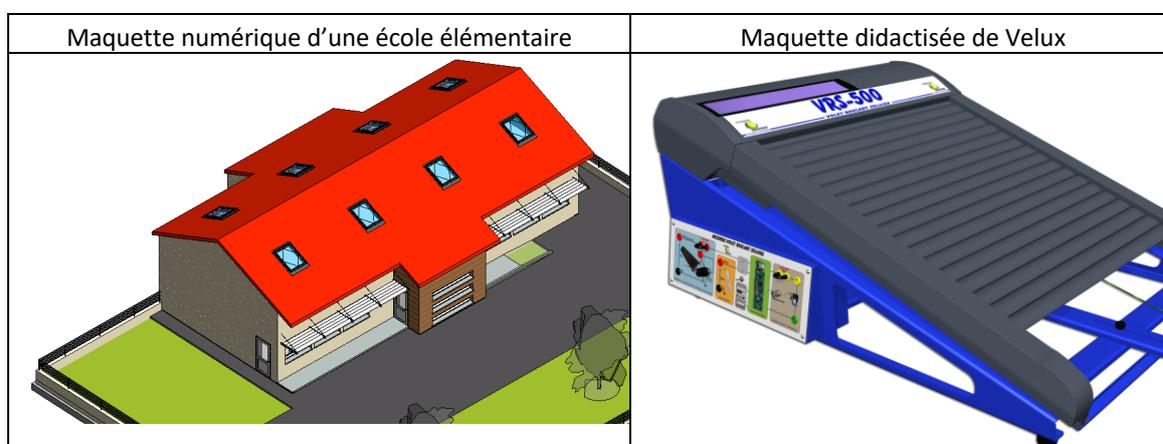
## Partie 1.2 – prise en main du support didactisé (durée : 0h30)

Objectif : s'approprier l'environnement et la structure du support didactisé du laboratoire.

Le candidat dispose des éléments suivants :

la maquette numérique d'une école élémentaire sous ArchiWizard ;

- le banc didactique d'étude combiné d'un volet roulant VRS500 et du sous-système développement durable ;
- des appareils de mesures (multimètre et thermomètre) ;
- la documentation technique des différents constituants du banc didactique combiné ;
- trois modèles de simulation Matlab « VRS\_S\_model », « simumaquette » et « simubatiment ».



Le contexte d'étude est celui d'une commune décidant de créer une nouvelle classe au premier étage dans les combles de son école maternelle / primaire (voir plans au DT1).

L'architecte souhaite favoriser au maximum l'éclairage naturel grâce à des Velux tout en maîtrisant le confort d'été en optimisant les besoins en chauffage et en éclairage.

Il souhaite notamment respecter la cible 10 « confort visuel » du label HQE (Haute Qualité Environnementale) préconisant un environnement intérieur satisfaisant sur le plan visuel sous une forme plus qualitative que quantitative.

L'accent est mis sur l'accès à la lumière naturelle avec la recherche de l'équilibre des luminances et la limitation de l'éblouissement ; l'indicateur opérationnel retenu est le facteur de lumière du jour (FLJ minimum de 2,5 % dans les bureaux et 2 % dans l'enseignement). Des données sur l'éclairage sont fournies par le DT3.

L'éclairage artificiel, conçu pour fonctionner en l'absence d'éclairage naturel ou en appoint de ce dernier, distingue comme indicateurs opérationnels :

- le niveau d'éclairage moyen à maintenir selon les usages (300 lux dans les bureaux, 300 lux dans les salles de classes primaire et secondaire et 500 lux dans les salles de cours du soir et d'enseignement pour adultes) ;
- le facteur d'uniformité (0,7 à 0,8 suivant l'implantation) ;
- l'équilibre de luminance et la limitation de l'éblouissement direct des sources d'éclairage ;
- la température de couleur et le rendu des couleurs.

La conception du bâtiment considéré amène, de plus, l'architecte à considérer les matériaux utilisés pour optimiser les échanges thermiques et ainsi la consommation énergétique globale du bâtiment.

### **Activité 1 Découverte de la maquette de l'école élémentaire**

En suivant le protocole expérimental 1 (DT2) et en complétant le document réponse DR1, relever les grandeurs principales caractérisant les besoins de chauffage et d'éclairage, le confort d'été, l'éclairage naturel et l'éclairage artificiel de la salle de classe située dans les combles du bâtiment. Indiquer dans quelle mesure la cible 10 du label HQE est respectée.

### **Activité 2 Exploitation de la maquette de l'école élémentaire**

À partir de vos données et du DT4, formuler des hypothèses sur l'influence du nombre de Velux, du type de vitrage et de la composition des parois de la salle sur les paramètres relevés précédemment.

Le système instrumenté à disposition se compose :

- d'un banc didactique VRS 500 ;
- d'une maquette instrumentée simulant un local permettant de mesurer :
  - la température intérieure du local, notée  $T_{int}$ , à l'aide d'un capteur Grove MCP9808 ;
  - l'éclairement du local, notée  $E_L$ , à l'aide d'un capteur Grove LS06-S ;
  - les températures de surface de la vitre, notées respectivement  $T_{si}$  et  $T_{se}$  à l'aide de sondes de température PT1000.

Dans une habitation, l'installation d'un volet roulant automatisé nécessite, dans la pièce considérée, la présence d'un capteur d'éclairement, ainsi que celle d'un capteur de température.

Deux environnements virtuels sont disponibles (raccourcis sur le bureau) :

- VRS-500\_EMP permet d'identifier les différents constituants et d'appréhender l'architecture du banc didactique ;
- VRS-500\_DD\_EMP permet d'identifier les différents constituants et d'appréhender l'architecture de la maquette instrumentée simulant un local.

### **Activité 3 Découverte du banc de mesure**

Ouvrir le volet à l'aide de la télécommande afin de localiser les différents éléments du système instrumenté décrit ci-dessus. À partir de vos observations, de l'environnement virtuel VRS-500\_EMP disponible sur le bureau (menu « le produit », puis « Description SSL-CK02 », puis « détail du mécanisme ») du sous-système volet roulant et du 2<sup>ème</sup> environnement virtuel VRS-500\_DD\_EMP, compléter le document réponse DR2 en indiquant les différents sous-systèmes formant la chaîne de puissance et la chaîne d'information :

- du dispositif de gestion de l'éclairement du local ;
- du dispositif de régulation de température intérieure du local.

Compléter les flux des chaînes de puissance et d'information.

Indiquer par ailleurs les différents composants présents faisant partie de la didactisation du banc d'étude, mais n'intervenant pas dans la gestion de l'éclairement ou la régulation de la température du local.

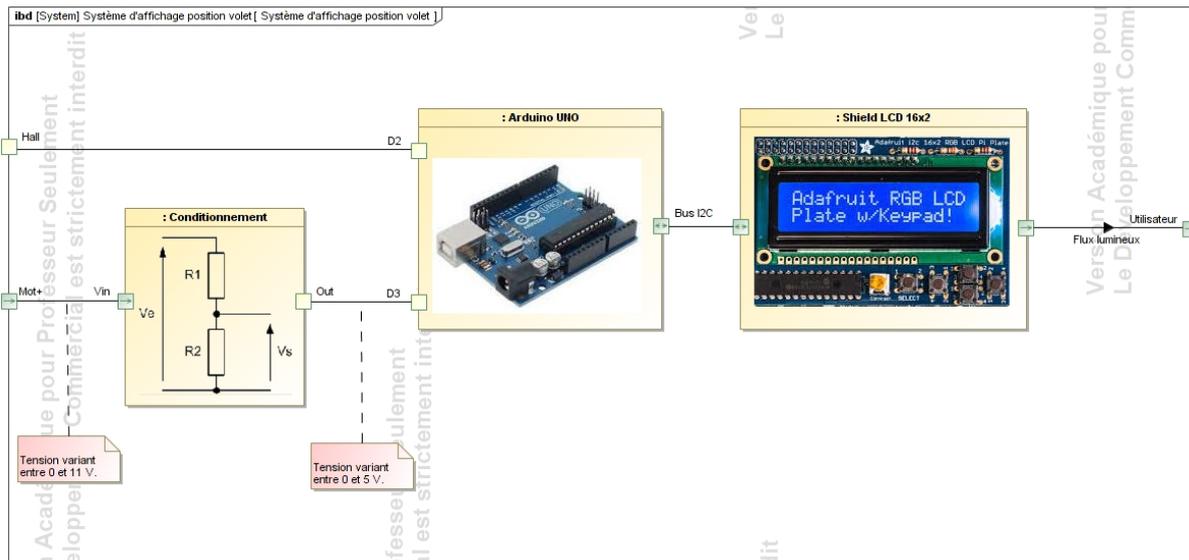
## **Partie 1.3 – expérimentations pour répondre à la problématique technique et scientifique (durée : 2h00)**

Problématique technique et scientifique : Comment optimiser l'éclairement naturel et artificiel, ainsi que la température d'un local ?

### **Étude de l'obtention de la position**

Le positionnement du volet à sa position optimale nécessite de récupérer l'information de position. Le volet roulant étant dépourvu de capteurs de fin de course, l'information de position du volet roulant est obtenue par le compte/décompte des impulsions Hall générées au niveau du moteur. Le système exploite cette information pour anticiper les arrivées en fin de courses haute et basse et gérer de manière optimale l'arrêt dans ces conditions.

On dispose pour cela d'une interface programmable à base de microcontrôleurs constituée comme indiqué sur la figure suivante.

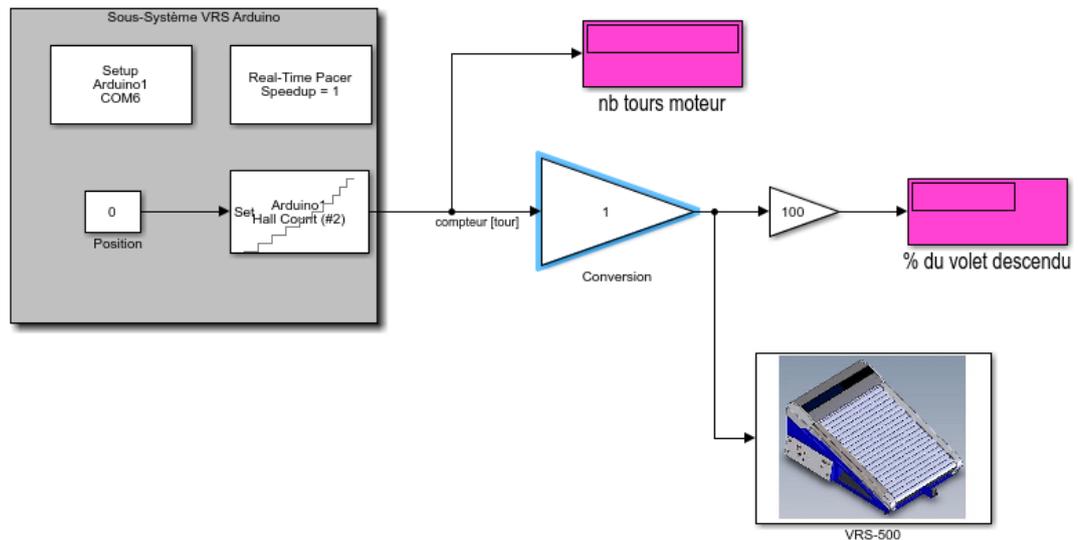


En plus du signal issu du capteur à effet Hall, les tensions aux bornes du moteur sont acquises via un étage de conditionnement. La carte Arduino se charge, selon les tensions mesurées aux bornes du moteur, d'incrémenter/décrémenter en conséquence la valeur d'un compteur. Cette valeur est remontée via une liaison USB au logiciel Matlab. On peut ainsi animer en temps-réel un modèle virtuel de la même manière que le système réel (on parle alors de « jumeau numérique »).



Mettre le volet en position « totalement ouvert ». Après avoir réalisé le câblage du sous-système, comme indiqué sur la figure précédente, ouvrir le modèle Matlab disponible dans le dossier « Fichiers Matlab/Virtual VRS-500 » par le fichier « VRS\_S\_model.mdl » et faire vérifier votre montage par votre examinateur.

### Modèle pour la commande d'un VRS-500 virtuel, piloté par le VRS réel



#### Activité 4 Obtention de la position du volet

En étudiant la transmission de mouvement du moteur au volet (voir VRS-500\_EMP « En savoir plus » onglet « Transmission et dispositif de tension lames », « Réducteur épicycloïdal » et « Le système à étoile d'avancement des lames »), déterminer le facteur de conversion (en bleu ci-dessus, fixé arbitrairement à 1) à appliquer dans le modèle Matlab pour que le jumeau se comporte de la même manière que le système réel. Renseigner la valeur et lancer la simulation en cliquant sur « Run ». Comparer le comportement du réel et du jumeau numérique lors d'une fermeture partielle puis complète, et faire valider votre modèle par votre examinateur.

Une fois l'activité 4 terminée, défaire le câblage du sous-système, fermer Matlab et relier le câble USB de la maquette instrumentée à l'ordinateur.

#### Étude de l'ouverture optimale du volet roulant

Le volet roulant du Velux a pour fonction de se protéger de la chaleur en été tout en optimisant l'éclairage intérieur naturel. L'objectif de cette partie est de déterminer expérimentalement l'ouverture optimale correspondant au réglage : « protection solaire sans obscurité complète » de ce volet roulant tel que le propose la société Velux (kit « Velux active ») dont la documentation est fournie en document ressource (voir Dossier Technique/VELUX Télécommande KRX100, page 62/113).

La maquette instrumentée, à l'avant du banc de mesure, est constituée (entre autres) :

- d'un capteur de température ambiante haute précision de type Grove MCP9808 ;
- d'un capteur d'éclairage intérieur.

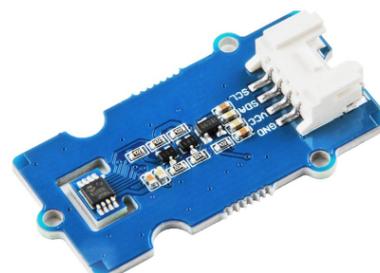
Le programme Arduino « activite5.ino » permet la mise en œuvre de ces capteurs, une fois la maquette instrumentée connectée via l'USB à l'ordinateur.

Après avoir lancé le logiciel Arduino, il faudra vérifier le numéro de port utilisé (« Outil/port ») et sélectionner le type de carte utilisée.

Pour visualiser les valeurs, il faudra cliquer sur le moniteur série (icône loupe en haut à droite) et éventuellement régler la vitesse de communication en bauds.



Capteur d'éclairage



Capteur de température ambiante

### Activité 5 Variations des température et éclairement ambiants en fonction de l'ouverture du volet

Proposer un protocole expérimental permettant de mesurer la température intérieure  $T_i$  et l'éclairage intérieur  $E_i$  en régime permanent en fonction de l'ouverture du volet, et le présenter à votre examinateur avant de le mettre en œuvre.

Tracer  $T_i = f(\% \text{ ouverture volet})$  et  $E_i = f(\% \text{ ouverture volet})$ .

Conclure sur l'ouverture optimale de ce volet roulant et comparer avec les préconisations du constructeur.

### Étude des échanges thermiques mis en jeu dans la régulation en température de la pièce

Dans le dimensionnement d'un bâtiment, afin d'optimiser la consommation énergétique globale, il est nécessaire de pouvoir évaluer les échanges thermiques mis en jeu. Ainsi le choix des matériaux entrant en jeu dans l'isolation revêt une importance toute particulière. En ce sens, il est nécessaire de connaître les caractéristiques thermiques des matériaux utilisés. Des valeurs théoriques sont données par les fournisseurs mais, bien souvent, afin d'aboutir à une solution optimale, les cabinets d'expertises en charge du projet mettent en œuvre une campagne expérimentale pour leur dimensionnement. La finalité de la maquette instrumentée est de pouvoir déterminer la résistance thermique de chaque matériau participant à l'isolation afin de pouvoir implémenter et/ou valider les modèles de simulations numériques du bâtiment considéré.

On s'intéresse ici à la détermination de la résistance thermique du matériau utilisé pour réaliser la vitre présente sous le volet roulant. Comme précisé précédemment, la vitre du banc d'essai est instrumentée à l'aide de deux sondes PT1000 sur chacun de ses cotés, afin de pouvoir évaluer la température réelle de chacune de ses parois. Une sonde PT1000 est aussi disponible hors du banc didactisé à toutes fins utiles.

La sonde PT1000 est un composant purement résistif dont la résistance  $R$  varie en fonction de sa température. On peut écrire :

$$R(T) = R_0 + K.T$$

avec :

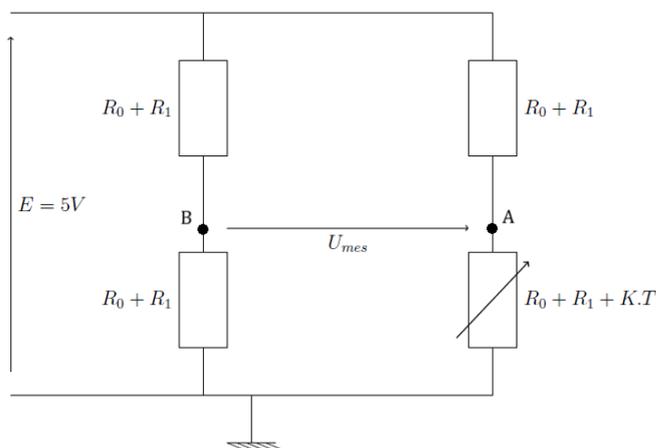
- $T$ , la température en  $^{\circ}\text{C}$  de la paroi sur laquelle la sonde est attachée ;
- $R_0$ , la valeur de  $R(T)$  pour  $T=0^{\circ}\text{C}$ , en  $\Omega$  ;
- $K$ , la valeur du gain de la sonde en  $\Omega \cdot ^{\circ}\text{C}^{-1}$ .

Attention : La sonde PT1000 est un capteur très fragile. Ne manipuler que la plaque sur laquelle il est soudé.

### Activité 6 Étude de la loi de comportement du capteur

Proposer un protocole expérimental permettant de déterminer les valeurs de  $K$  et  $R_0$  pour la sonde PT1000. Après avoir appelé l'examinateur pour valider ce protocole, le mettre en œuvre et déterminer alors la valeur des deux constantes  $K$  et  $R_0$  de la sonde PT1000 mise à votre disposition.

Afin de mesurer la température de chacune des parois de la vitre, la sonde PT1000 est insérée dans le montage du pont de Wheatstone de la figure suivante.



On donne  $R_1 = 10 \text{ k}\Omega$ . On obtient :

$$U_{mes} \approx \frac{E \cdot K \cdot T}{4 \cdot (R_0 + R_1)}$$

L'acquisition de la mesure est envisagée à l'aide d'une carte Arduino. Les entrées analogiques de la carte sont codées sur 10 bits pour une variation de tension allant de 0V à 5V.

### Activité 7 Étude de la résolution de mesure

Donner la résolution de mesure (quantum de conversion) en V,  $r_v$ , d'une entrée analogique de la carte Arduino et en déduire alors la résolution de mesure en °C,  $r_c$ , obtenue si l'on branchait directement les points de mesure du pont de Wheatstone précédent sur des entrées analogiques de la carte. Commenter la valeur trouvée, en considérant une plage de variation de température à évaluer allant de 10°C à 70°C, et proposer une modification du dispositif de mesure afin d'obtenir une résolution de 0,2°C.

Une paroi peut être modélisée, en régime permanent, comme une résistance thermique vérifiant la relation :

$$r_{th,vitre} \cdot \varphi = (T_{se} - T_{si})$$

avec :

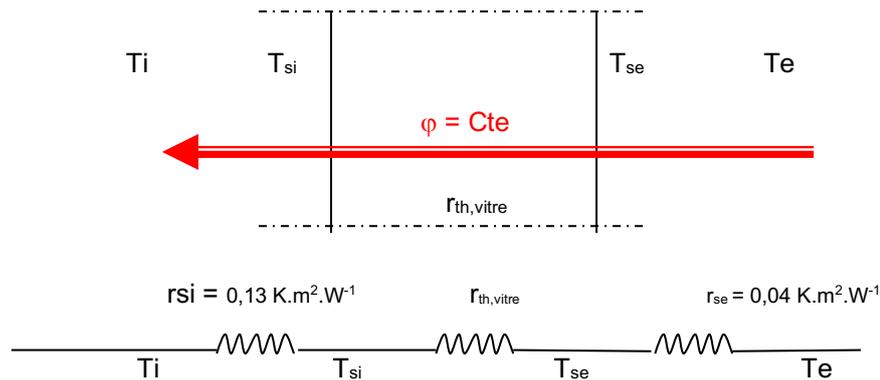
- $T_{se}$  et  $T_{si}$ , respectivement les températures de surface extérieure et intérieure en K ;
- $r_{th,vitre}$ , la résistance thermique du vitrage considéré en  $K.m^2.W^{-1}$  ;
- $\varphi$  : le flux thermique à travers la paroi en  $W.m^{-2}$ .

Afin de pouvoir implémenter un modèle de simulation pour dimensionner un local, et évaluer les échanges thermiques, il est nécessaire de connaître la valeur de la résistance thermique  $r_{th,vitre}$  du vitrage utilisé.

Le programme Arduino « activite8.ino » permet de lancer l'acquisition nécessaire pour traiter l'activité 8. Pour cette activité, allumer le spot.

### Activité 8 Détermination de la résistance thermique de la vitre utilisée

À partir de mesures que vous effectuerez et du schéma ci-dessous, déterminer la valeur expérimentale de  $r_{th,vitre}$ .



Le constructeur donne  $r_{th,vitre} = 0,075 K.m^2.W^{-1}$ . Comparer la valeur expérimentale et la valeur constructeur et analyser.

Des modèles de simulation Matlab, « simumaquette » pour la maquette didactisée et « simubatiment » pour l'école considérée, sont fournis dans le dossier candidat. Ils permettent de simuler le fonctionnement du système de chauffage de la maquette et de l'école considérée. Seul le modèle « simumaquette » est utilisé dans les activités suivantes. Pour utiliser chaque modèle, il faut double-cliquer sur le fichier « .mat », cliquer sur « Finish », puis ouvrir le fichier « .slx » dans l'arborescence.

### Activité 9 Analyse du modèle de simulation

Détailler le lien entre chacun des composants du modèle « simumaquette » avec le système réel en complétant les zones de commentaires en pointillés.

### Activité 10 Mise en œuvre du modèle de simulation

Après avoir renseigné la valeur de  $r_{th,vitre}$  déterminée précédemment (« clic droit » dans le modèle simulink, « Find Referenced Variables », « Based workspace »), lancer la simulation et analyser les résultats obtenus. Proposer des améliorations et/ou des modifications du modèle de simulation afin d'obtenir une réponse plus rigoureuse vis-à-vis de la réalité.

### Activité 11 Synthèse

Au regard des différentes activités réalisées, répondre à la problématique posée et conclure sur les paramètres à prendre en compte pour optimiser le confort dans l'habitat (éclairage et température à l'intérieur).

#### **Partie 1.4 – élaboration du scénario d'une séance à caractère expérimental (durée : 0h45)**

Objectif : développer une séance à caractère expérimental s'intégrant dans la séquence pédagogique proposée dans la partie 1.1.

##### **Production attendue**

Une séance à caractère expérimental pertinente doit être proposée après avoir :

- situé cette séance dans la séquence pédagogique (objectifs et prérequis) ;
- décrit l'organisation matérielle et pédagogique de la séance (nombre d'élèves, systèmes utilisés, travail en îlots ou autres) ;
- décrit et justifié la (ou les) démarche(s) pédagogique(s) retenue(s) (démarche d'investigation, de résolution de problème technique, de projet ...)
- détaillé le scénario des activités que doivent réaliser les élèves sur le support didactisé à l'aide des documents fournis ci-après ;
- réalisé concrètement au moins une des activités expérimentales proposées dans la séance développée. Cette activité doit être nouvelle et différente de celles réalisées dans la partie 1.3. Préciser l'objectif de la manipulation entreprise, proposer et mettre en œuvre son protocole expérimental comme le feraient les élèves et analyser les résultats obtenus ;
- explicité clairement l'apport de la séance proposée dans le développement des savoir-faire et compétences des élèves.

Les documents fournis pour mener cette partie sont les suivants :

- le banc didactique d'étude combiné d'un volet roulant VRS500 et le sous-système complémentaire développement durable ;
- les différents fichiers Arduino permettant d'effectuer des acquisitions ;
- les différents fichiers de simulation Matlab ;
- une caméra thermique.

#### **Phase 2 – préparation de l'exposé (durée : 1h00)**

Objectif : finaliser le support de présentation pour l'exposé devant le jury.

##### **Production attendue**

Un document numérique doit être réalisé afin de :

- présenter la séquence pédagogique ;
- présenter la pertinence du support didactisé par rapport au besoin pédagogique ;
- présenter la séance à caractère expérimental.

#### **Phase 3 – exposé oral et entretien avec le jury en salle (durée : 1h00)**

# Épreuve d'admission d'exploitation pédagogique d'une activité pratique relative à l'approche spécialisée d'un système pluritechnologique

## A. Présentation de l'épreuve

### Textes de référence

<http://www.devenirenseignant.gouv.fr/cid98734/les-epreuves-de-l-agregation-externe-section-sciences-industrielles-de-l-ingenieur.html>

[Arrêté du 24 juin 2019 modifiant l'arrêté du 28 décembre 2009 fixant les sections et les modalités d'organisation des concours de l'agrégation](#)

- Durée totale 6 heures (activités pratiques 4 heures, préparation de l'exposé 1 heure, exposé 30 minutes maximum, entretien 30 minutes).
- Coefficient 2.
- 10 points sont attribués à la partie liée aux activités pratiques et 10 points à la partie liée à l'exposé et à l'entretien avec le jury.

Le support de l'activité pratique proposée permet, à partir d'une approche systémique globale, l'analyse d'un problème technique particulier relatif à la spécialité du concours, dans l'option choisie. La proposition pédagogique attendue, directement liée aux activités pratiques réalisées, est relative :

- aux enseignements technologiques de la spécialité du cycle terminal « sciences et technologies de l'industrie et du développement durable (STI2D) » du lycée ;
- aux programmes de BTS CIEL option « Informatique et Réseau » ainsi que des BUT Génie Électrique et Informatique Industrielle, Réseau et Télécommunications et Informatique dans les champs couverts par l'option Ingénierie Informatique.

L'épreuve a pour but d'évaluer l'aptitude du candidat à :

- mettre en œuvre des matériels ou équipements, associés si besoin à des systèmes informatiques de pilotage, de traitement, de simulation, de représentation ;
- conduire une expérimentation, une analyse de fonctionnement d'une solution, d'un procédé, d'un processus, dans la spécialité du concours, afin d'analyser et de vérifier les performances d'un système technique ;
- exploiter les résultats obtenus et formuler des conclusions ;
- concevoir et organiser une séquence de formation pour un objectif pédagogique imposé à un niveau de classe donné et présenter de manière détaillée un ou plusieurs points-clés des séances de formation qui la constituent. Elle prend appui sur les investigations et les analyses effectuées au préalable par le candidat au cours d'activités pratiques relatives à un système technique.

Le candidat est amené au cours de sa présentation orale à expliciter sa démarche méthodologique, à mettre en évidence les informations, les données et les résultats issus des investigations conduites au cours des activités pratiques qui lui ont permis de construire sa proposition pédagogique.

Au cours de l'entretien, le candidat est conduit plus particulièrement à préciser certains points de sa présentation ainsi qu'à expliquer et justifier les choix de nature didactique et pédagogique qu'il a opérés dans la construction de la séquence de formation présentée.

### Déroulement de l'épreuve

L'évaluation du candidat s'effectue en trois phases.

### **Phase 1 – Manipulation expérimentale au laboratoire (durée 4h00)**

Cette phase, d'une durée totale de 4 heures, se déroule en trois parties dans le laboratoire où sont mis à disposition les différents supports techniques qui permettent à chaque candidat de proposer une séquence pédagogique. Cette dernière s'appuie sur les activités pratiques réalisées par le candidat.

#### Première partie (durée ≈ 0h30)

Dans cette partie de prise en main du système, les études et activités proposées au candidat ont pour objectif de faciliter la compréhension de l'architecture globale du système et de son fonctionnement. À la fin de cette première partie, le jury s'assure que le candidat s'est bien approprié le support de l'activité pratique ainsi que la problématique proposée.

#### Deuxième partie (durée ≈ 2h00)

Dans cette partie d'approfondissement, le candidat doit suivre les études et les activités proposées permettant de répondre à la problématique de l'activité pratique. Cette partie doit permettre au candidat, par la mobilisation de compétences et de connaissances caractéristiques du niveau de l'agrégation, de développer/intégrer des modules logiciels, d'intégrer du code, de déterminer un programme, de résoudre les problèmes posés, puis d'exploiter les résultats obtenus (modèles, représentation UML/SysML, algorithmes, jeu de tests, interface graphique résultats numériques, etc).

#### Troisième partie (durée 1h30)

Dans cette partie, le candidat dispose librement du support de TP pour préparer la trame détaillée de sa séquence pédagogique. En s'appuyant sur les développements, les investigations et les analyses effectués durant les deux premières parties ou d'autres éléments qu'il a la possibilité de concevoir, le candidat doit proposer un ou plusieurs protocoles expérimentaux lui permettant de répondre à la fois à la problématique scientifique et la problématique pédagogique.

Cette phase 1 se déroule dans le laboratoire dans lequel se trouve le support de TP utilisé.

L'exploitation pédagogique est relative à l'enseignement spécifique du cycle terminal :

- en lycée, de la voie technologique sciences et technologies de l'industrie et du développement durable de la spécialité systèmes d'information et numérique (STI2D SIN) ;
- en post bac, des programmes du BTS CIEL option « Informatique et Réseau » et des BUT Génie Électrique et Informatique Industrielle, Réseau et Télécommunications et Informatique relatifs aux champs couverts par l'option ingénierie informatique.

Les candidats disposent de l'ensemble des moyens nécessaires à l'expérimentation et d'un poste informatique, relié à Internet, doté des logiciels courants de bureautique et des logiciels plus spécifiques liés au sujet qui leur est proposé.

### **Phase 2 – Préparation de l'exposé (durée 1h00)**

Durant cette phase d'une heure, le candidat s'appuie sur la trame de la séquence qu'il a construite dans la troisième partie de la phase 1 et prépare l'intervention qu'il fera devant le jury. Il dispose d'un poste informatique relié à Internet doté des logiciels courants de bureautique et de tous les résultats de mesure, analyse ou investigations issus de la phase 1.

### **Phase 3 – Exposé oral et entretien avec le jury en salle (durée 1h00)**

L'exposé oral est d'une durée maximale de 30 minutes. Le jury n'intervient pas pendant l'exposé du candidat. L'entretien avec le jury est d'une durée maximale de 30 minutes.

Le candidat est amené au cours de sa présentation orale à présenter :

- le système (durée maximale 5 minutes) ;
- une synthèse des activités menées dans la deuxième partie de la première phase de l'activité pratique (durée maximale 5 minutes) ;  
son exploitation pédagogique (durée maximale 20 minutes).

Au cours de l'entretien, le candidat est amené à :

- préciser certains points de sa présentation ;
- expliquer et justifier les choix de nature didactique et pédagogique.

Pour la présentation devant le jury, les candidats ont à leur disposition un tableau, un ordinateur et un vidéoprojecteur. Ils disposent d'un poste informatique doté des logiciels courants de bureautique, connecté à Internet et des résultats obtenus lors des phases 1 et 2, stockés dans l'espace qui leur est dédié.

Les supports retenus lors de la session 2025 étaient les suivants :

- Drone instrumenté communicant ;
- Système de contrôle aérien ;
- Robot pédagogique ;
- Système de contrôle d'accès RFID ;
- Détection de localisation de tumeurs cérébrales par traitement d'images IRM ;
- Réception et traitement de données issues de modules météorologiques ;
- **TP Laurent**

Ces études permettent aux candidats de mettre en œuvre leurs compétences à un haut niveau scientifique dans les tâches suivantes :

- modélisation de systèmes (UML/SysML ...) ;
- analyse critique et validation de modèles (de calcul, de données, de logiciels, d'échange) et proposition de modifications ;
- programmation en langages C/C++, Python, SQL notamment ;
- suivi de programmes et trace des variables ou instances ;
- configuration, déploiement et test de réseaux informatiques ;
- intégration et utilisation de bibliothèques logicielles (traitement d'images ou de données massives, interface graphique, protocoles réseaux, bases de données ...) ;
- développement d'algorithmes spécifiques et prise en compte de leurs performances.

## B. Commentaires du jury

### • **Compétences attendues par le jury**

Lors de cette épreuve d'activités pratiques, le jury évalue chez les candidats les compétences suivantes :

- faire preuve d'une maîtrise de l'ingénierie pédagogique en élaborant une séance d'activités pratiques dans une séquence pédagogique cohérente, inscrite dans une progression pédagogique ;
- proposer des activités pratiques ou des modélisations nouvelles, adaptées au niveau imposé et aux objectifs de formation ;
- s'approprier un support, un environnement de développement, un système d'exploitation ;
- s'approprier la problématique associée ;
- élaborer, conduire et justifier un algorithme, un développement et/ou un protocole de test ;
- exploiter et analyser des résultats de tests ;
- élaborer, justifier et analyser de manière critique un modèle ;
- évoluer en autonomie en mobilisant toutes ses connaissances et ses compétences ;
- présenter oralement ses travaux avec clarté, précision et rigueur ;
- être réactif et pertinent dans les réponses aux questions posées par les membres du jury.

- **Analyse des résultats**

**Concernant la phase 1** : le jury considère que les candidats répondent favorablement aux attentes quant à leur capacité :

- à analyser le système d'un point de vue de l'architecture logicielle et de l'infrastructure informatique ;
- à explorer rapidement les différentes parties d'un logiciel selon des méthodes de recherche intégrées dans de nombreux environnements de développement pour leur séquence pédagogique ;
- à concevoir et réaliser de nouveaux développements pour leur séquence pédagogique.
- à utiliser avec aisance le système d'exploitation GNU/Linux.

**Concernant la phase 3** : la prestation orale des candidats lors de la présentation des travaux devant le jury se révèle souvent incomplète et/ou inconsistante pour les raisons suivantes :

- la présentation du système et de ses composantes est souvent insuffisante. Les candidats ne proposent pas suffisamment de représentations UML/SysML ou de type boîtes noires faisant apparaître les entrées sorties du système pour en expliciter le fonctionnement. Les candidats peuvent cependant s'appuyer sur les éléments qui leur sont fournis dans le sujet ;
- l'analyse des résultats expérimentaux est trop souvent superficielle, trop qualitative, pas assez quantitative et ne permet pas au jury de percevoir ce que le candidat a compris de la problématique de l'activité pratique ;
- les temps de présentation préconisés ne sont pas toujours respectés (système, résultats expérimentaux, exploitation pédagogique) et les 30 minutes ne sont que rarement utilisées ;
- le choix des savoirs des séquences pédagogiques est très souvent cohérent mais le découpage en séances permettant d'évaluer les savoir-faire de manière progressive n'est pas maîtrisé ou proposé ;
- certains candidats présentent des activités sans rapport avec les compétences visées ;
- l'exploitation pédagogique envisagée se limite trop souvent à proposer une séquence pédagogique incluant la seule activité de travaux pratiques conduite précédemment au laboratoire sans fil conducteur avec d'autres éléments ;
- les moyens matériels à mettre en œuvre compte tenu du nombre d'élèves ou d'étudiants présents dans la section considérée ne sont pas souvent explicités et ne permettent pas de vérifier le réalisme de la solution pédagogique ;
- l'exploitation pédagogique n'utilise pas de manière pertinente le support proposé en travaux pratiques ou un autre support représentant des solutions techniques similaires, ce qui ne permet pas au jury d'apprécier la capacité du candidat à créer du contenu pédagogique à partir de systèmes réels ou didactiques ;
- le positionnement de la séquence pédagogique dans une progression pédagogique est très rarement proposé ;
- les prérequis de la séquence pédagogique présentée ainsi que les méthodes d'évaluation des compétences visées (tant leur choix, organisation dans la séquence et leur mise en œuvre) sont insuffisamment énumérés.

- **Conseils du jury**

Pour réussir au mieux les deux phases évaluées (Phases 1 et 3), le jury invite les candidats à maîtriser parfaitement l'utilisation des systèmes d'exploitation largement utilisés en ingénierie informatique (Windows, GNU/Linux, systèmes embarqués temps réel), les logiciels courant de la bureautique (Office ou libre Office, utilitaires classiques de capture d'écran) et dessin technique (type draw.io). Les candidats doivent maîtriser les concepts de programmation (compilation/utilisation d'un interpréteur, programmation orientée objet et leur modélisation) et leur implémentation dans les langages de programmation suivants : C, C++ et Python. Les candidats doivent maîtriser l'utilisation de bases de données et les requêtes SQL. Les candidats doivent garder à l'esprit que cette agrégation demande des qualités dépassant largement le simple exercice de programmation.

Le jury conseille aux candidats d'étudier par exemple les logiciels de contrôle commande moderne qui sont très souvent construits en couche partant du bas niveau (capteur, micro-contrôleur) jusqu'aux applications de haut niveau (contrôle d'une trajectoire, interface Homme-machine, API) en passant par des couches intermédiaires (pilotes logiciels, systèmes de messagerie, gestionnaire de tâches). Dans ces systèmes, les concepts de la programmation orientée objet sont utilisés afin de procurer à la solution informatique des caractéristiques pertinentes pour le domaine de l'ingénierie informatique.

Le jury attend que les candidats mettent à profit le temps dont ils disposent durant la troisième partie de la phase 1 (1h30) pour conduire d'autres développements (exemples de manipulation réalisées par les apprenants) que ceux demandés précédemment, sur lesquels ils pourront s'appuyer pour proposer ensuite une exploitation pédagogique originale et personnelle. Le jury rappelle que la gestion du temps, le respect des durées indiquées sur les sujets permettent de guider le candidat et de faciliter son évaluation dans le contexte d'un concours sélectif.

Contrairement à la première épreuve d'admission relative à l'approche globale d'un système pluri-technologique, le jury de la seconde épreuve d'admission relative à l'approche spécialisée d'un système pluri-technologique rappelle que le candidat doit présenter le système étudié, les expérimentations effectuées et les résultats obtenus. Le jury cherche à évaluer la clarté d'expression, les facultés du candidat à s'approprier, à synthétiser et à restituer avec précision et rigueur les contenus techniques et scientifiques exploités lors de la première partie.

Le jury rappelle qu'il s'agit pour le candidat de présenter une séquence pédagogique inscrite dans une progression pédagogique. Dans cette séquence, doit apparaître de façon précise un ou plusieurs points caractéristiques des séances de formation proposées. L'ensemble doit prendre appui sur les investigations et les analyses effectuées au préalable par le candidat au cours d'activités pratiques relatives au support proposé, en tenant compte du niveau de la formation visée et en adaptant l'exploitation du support à celui-ci.

Le jury rappelle que la conception d'une séquence pédagogique visant des savoir-faire et des savoirs doit suivre une méthodologie dont les étapes peuvent être les suivantes :

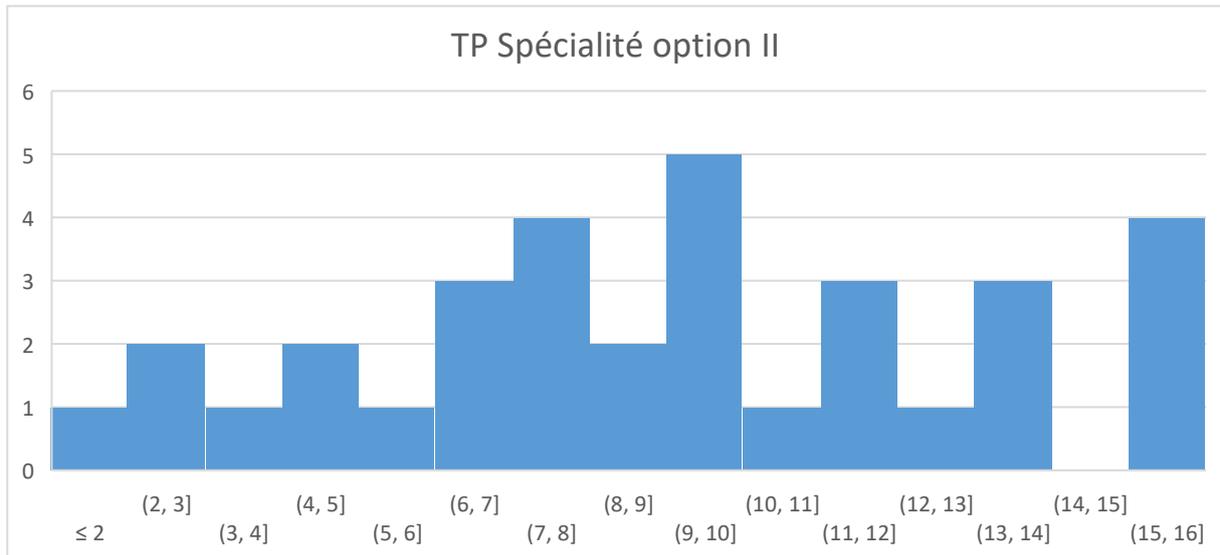
- le choix de l'objectif pédagogique qui doit s'appuyer sur un support technique représentatif des sciences de l'ingénieur. Ce dernier est imposé par le sujet de TP ;
- le choix des compétences visées par le candidat. Il n'est pas nécessaire de choisir l'ensemble des compétences définies dans le sujet de TP ;
- le choix des savoir-faire définis pour la ou les compétences choisies. L'attestation de la construction d'une compétence est obtenue en validant l'acquisition de tous les savoir-faire. Des indicateurs observables doivent permettre de les évaluer précisément ;
- le choix des savoirs associés aux savoir-faire (à positionner en prérequis, au lancement, en ressource durant les séances ou bien en synthèse) ;
- et enfin la répartition des savoir-faire et des savoirs dans chaque séance.

Le jury conseille aux candidats de préparer la partie pédagogique en utilisant les référentiels des diplômes cités dans cette épreuve et qu'ils ont à leur disposition dans les espaces de travail qui leur sont fournis.

Le jury incite également les candidats à l'agrégation à pousser la porte des lycées et des IUT ainsi que des INSPÉ ou des ENS, afin d'obtenir de la part d'enseignants de terrain ou formateurs, un maximum d'informations d'ordre pédagogique leur permettant de préparer au mieux cette épreuve.

## C. Résultats

Trente-six candidats étaient inscrits à l'épreuve. 3 candidats absents n'ont pas pu composer et s'exercer à cette épreuve. La moyenne des notes obtenues est de 9,12/20 avec un écart type de 4,07. La meilleure note est de 15,8/20 et la note la plus basse est de 1,9/20. Un histogramme de répartition des notes est donné ci-dessous



### Exemple de sujet pour l'exploitation pédagogique d'une activité pratique relative à l'approche spécialisée d'un système pluritechnologique

Ci-dessous est donné un exemple de sujet, ainsi que quelques éléments de correction.

## 3 Mise en situation

### 3.1 Présentation

---

Depuis le déploiement des objets connectés, les amateurs de météorologie peuvent publier des informations grâce à des appareils plus ou moins sophistiqués. Les plus simples sont des stations météo équipées de modules sans fil.

Ces appareils ne proposent pas toujours un accès facile aux informations. La station permet de les afficher mais il n'est pas toujours possible de les récupérer simplement pour ensuite les traiter ou les stocker.

L'idée du système proposé ici est d'enlever la station et de la remplacer par un module informatique complet permettant de récupérer des données depuis les modules. On pourra ensuite stocker les données dans une base de données et les afficher par l'intermédiaire de pages Web.

Le système est composé :

- de trois capteurs météorologiques sans fil sous forme de modules émettant les données à 433 MHz ;
- d'un sous ensemble de réception des trames avec un module de réception 433 MHz et un arduino UNO contenant le programme de décodage des informations provenant des modules ;
- d'une carte Raspberry Pi pour le logiciel applicatif.

Le diagramme de déploiement de l'ensemble est en annexe (sec. 7.1, fig. 5).

### 3.2 Travail à réaliser

---

Le travail proposé dans la suite consiste à :

1. comprendre et modifier le logiciel de décodage des trames radio pour prendre en compte un nouveau type de module ;
2. comprendre et modifier le logiciel de réception des trames série pour y ajouter également ce nouveau type de module ;
3. modifier le logiciel pour accueillir plusieurs modules de températures en utilisant les canaux ;
4. ajouter la fonctionnalité de stockage dans une base de données ;
5. réaliser le service Web qui présente les données stockées.

## 4 Première phase, première partie (durée 0H30)

### 4.1 Accès à la Raspberry Pi via VNC

---

Objectif : accéder à la Raspberry via VNC

La carte Raspberry ne possède pas d'écran ni de clavier. Elle est uniquement accessible à distance.

#### Q 1 Brancher la Raspberry Pi et s'y connecter

1. Brancher la Raspberry Pi.
2. Lancer VNC (le logiciel est installé sur le poste de travail) et se connecter à la carte Raspberry Pi (login : `pi`, mot de passe : `raspberrypi`). Il faut éventuellement vérifier que le logiciel n'utilise pas de proxy dans Fichier/Préférences.

⇒ 192.168.1.115, 116...

Une fois le système alimenté, il suffit de rentrer l'adresse IP de la carte Raspberry dans la barre d'adresse de VNC. Une fenêtre d'authentification apparaît : compléter nom d'utilisateur et mot de passe.

Si l'authentification réussit le bureau de la Raspberry Pi apparaît. Si l'authentification échoue il se peut que la Raspberry Pi ait changé d'adresse, ce qui est rare. Normalement le DHCP donne la même adresse IP à la même adresse MAC. Sinon il faut refaire un `scan-arp` depuis le poste Linux du concepteur (ou `arp -a` sous windows). On peut aussi essayer `nmap` pour obtenir toutes les IPs du réseau local.

### 4.2 Chaîne de décodage et de réception des trames

---

Objectif : valider l'ensemble de la chaîne de décodage et de réception des trames

Les modules météo émettent leurs informations à 433 MHz. Ces informations sont réceptionnées par un module spécialisé dont le câblage avec un Arduino est donné fig. 1

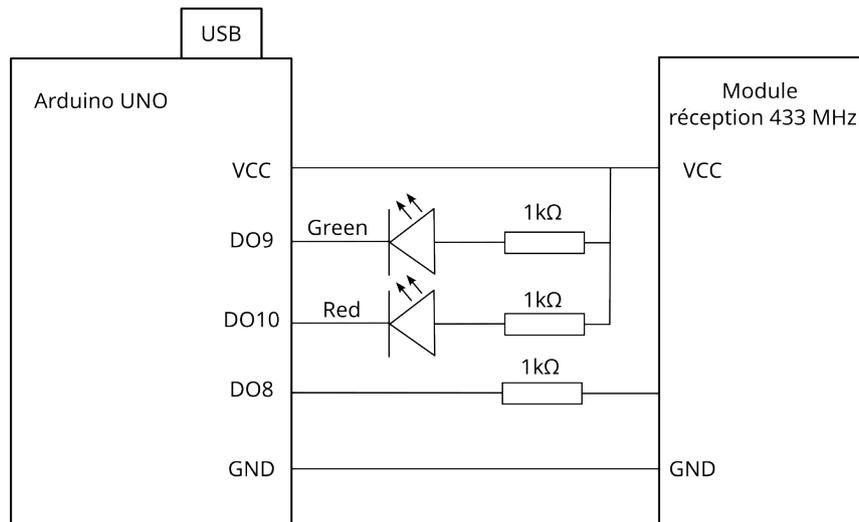


Figure 1: Schéma de câblage du module de réception avec la carte Arduino

L'Arduino contient le logiciel de décodage des trames et envoie les informations sur une ligne de type RS232 (qui est en fait la liaison USB vue comme une liaison série). Le sous-ensemble de réception (Arduino + module de réception 433 MHz) est déjà câblé.

### Q 2 Réceptionner les trames sur la Raspberry Pi

1. Brancher l'Arduino à la Raspberry Pi.
2. Trouver le fichier de périphérique permettant de réceptionner les trames sur la Raspberry Pi.
3. Vérifier logiciellement la réception des trames venues de l'Arduino (`gtkterm` est installé sur la carte Raspberry Pi). La vitesse d'émission est 57600 bauds. Le format des trames est suffisamment évident pour ne pas être détaillé ici.
4. Faire valider l'ensemble.

❖ Le système est déjà câblé. Il suffit au candidat de brancher la carte Arduino à la carte Raspberry par le câble USB.

Ouvrir `gtkterm`, configurer la bonne vitesse de réception et le bon périphérique (en fait `gtkterm` propose le fichier de périphérique de la Raspberry). Poser la question au candidat pour savoir comment trouver un périphérique de type USB et série inconnu sur Raspberry Pi. La réponse attendue est d'aller voir les fichiers périphériques dans `/dev/` et/ou consulter les messages du système avec `dmesg`.

Lorsque `gtkterm` est correctement configuré on voit s'afficher les trames texte envoyées depuis la carte Arduino pour les différents modules.

### 4.3 Mise en service du système de réception des trames sur la carte Raspberry Pi

Objectif : mettre en marche et faire un compte-rendu oral du fonctionnement de l'application de base.

L'application de réception des trames décodées est développée à l'aide de l'IDE CodeLite sur la carte Raspberry (menu Programming après avoir cliqué sur la framboise de l'interface graphique de la Raspberry Pi).

### Q 3 Prise en main de l'application de réception des trames sur la Raspberry Pi

1. Analyser le code du logiciel (dossier `stationMeteo` sur le bureau) sur la base de son diagramme de classes et des diagrammes de séquence fournis en annexe (sec. 7.1).
2. Modifier, si nécessaire, le port série à utiliser pour réceptionner les trames.
3. Lancer le logiciel et vérifier que les données reçues sont bien affichées dans une console.
4. Expliquer l'architecture de l'application (diagramme de classes) et montrer une bonne compréhension de son fonctionnement en présentant les points intéressants tel que les héritages, les interfaces, etc.

La description pourra être appuyée sur un déroulement partant de la réception d'une trame à son traitement par les différents objets.

❖ Il suffit de lancer Codelite, le logiciel sur le poste candidat est fonctionnel. Si on exécute le code, on voit la réception des trames et la récupération des données.  
Le candidat doit analyser le diagramme de classes. Il est proposé de suivre le cheminement de l'information. Il est important que le candidat ait relativement bien compris cette architecture pour la suite.  
Cheminement de l'information : Une trame est reçue par un objet de type `tRS232`. La méthode de cette classe est bloquante.  
Cette classe est composée avec `tDispatcheur`. Cette classe sert d'aiguilleur pour les trames reçues. Pour gérer le fait que l'appel est bloquant, on utilise un thread avec la méthode `capture` de `tDispatcheur`.  
La méthode `capture` va trier les trames reçues et insérer les données dans le bon objet.  
Pour cela, les objets intéressés par les données (de type `tModule`) doivent s'inscrire auprès de `tDispatcheur` (dans le `main`). Une fois la donnée insérée, l'objet appelle l'observer qui s'est lui-même inscrit auprès de lui par la méthode `processData`.  
Ce `processData` peut ensuite être utilisé pour ce qu'on veut. Dans un premier temps c'est l'affichage, ce pourrait être l'enregistrement des données dans une base de données.

## 5 Première phase, deuxième partie (durée 2H00)

### 5.1 Logiciel de décodage des trames, ajout d'un module météorologique (45min)

Objectif : comprendre le processus d'acquisition des trames émises par les capteurs et ajouter le décodage d'un nouveau module de pluviométrie.

Une copie du logiciel implanté dans la carte Arduino est présent sur le bureau de la carte Raspberry Pi dans le répertoire `osv3_decode` ou sur le bureau de votre poste de travail (même répertoire).

⇒ Le logiciel en question n'est pas encore fonctionnel, contrairement à celui qui s'exécute actuellement sur l'Arduino.

Un extrait de la documentation sur le `Timer1` et les interruptions du microcontrôleur Atmega328P présent sur la carte Arduino est donné en annexe (sec. 7.2) ou sur le poste de travail.

Un extrait de la documentation sur le protocole RF des stations Oregon Scientific est disponible en annexe (sec. 7.3) ou dans le répertoire documentation de votre poste de travail.

#### Q 4

 Décrire la structure du programme Arduino

Ouvrir le programme `osv3-decode` dans l'IDE Arduino, décrire la structure du programme. En particulier, expliquer le principe de l'acquisition des trames (comment s'assure-t-on que toute nouvelle trame reçue par le récepteur 433MHz sera décodée sans délai). Vous utiliserez la méthode de description de votre choix. Il n'est pas nécessaire de détailler chaque fonction mais plutôt de comprendre leur rôle.

❖ Cette partie se fait avec l'IDE Arduino sur la Raspberry ou sur le poste candidat au choix. On apporte au candidat un module supplémentaire de pluviométrie qui n'est pas décodé par le système pour l'instant.  
Dans la description du logiciel Arduino on attend du candidat qu'il parle de l'interruption Timer sur capture d'événement de la broche D8 de la carte Arduino.  
La gestion de l'acquisition par interruption permet de s'assurer que toute trame arrivant sur le récepteur radio sera traitée. L'interruption capture du Timer permet de calculer le temps de chaque pulse de la trame en repérant tous les fronts (montants et descendants). Le Timer est configuré dans la fonction `setup()`. La fonction `loop()` décale les bits du tableau `packet []` contenant la trame de 1 vers la droite (nécessaire à cause du mode de décodage de la largeur des pulses sur le code Manchester qui décale d'un bit entre le preamble et le sync) et aiguille le paquet vers la fonction de décodage du module correspondant en fonction de son premier octet.

Pour l'aiguillage des trames il faut que le candidat comprenne que le tableau `packet[]` contient la trame à partir du nibble sync et que dans la fonction `loop()` les bits des nibbles ne sont pas inversés, donc le nibble de sync n'est pas un A mais un 5 et le nibble suivant est le premier de l'ID du module à décoder. Ensuite chaque module possède une fonction de décodage qui vérifie le checksum et décode les données en fonction du format de la trame propre au module (dans ces fonctions les bits des nibbles sont inversés et il faut tenir compte de l'envoi LSB first).

#### Q 5 Régler les timings

1. En utilisant l'extrait de documentation du Timer de l'ATmega328P donné en annexe (sec. 7.2), chercher la fréquence de l'horloge en entrée du `Timer1` d'après sa configuration dans la fonction `setup()`.
2. À partir des extraits du protocole Oregon Scientific donnés en annexe (sec. 7.3), compléter les timings en début de programme pour assurer le décodage des trames (timing de `SHORT_PULSE_MIN`, `SHORT_PULSE_MAX`, `LONG_PULSE_MIN` et `LONG_PULSE_MAX`). La fréquence d'horloge du microcontrôleur est de 16MHz.
3. Vérifier le bon décodage des trames du module de température déjà présent.

❖ ⇒ Partie à revoir. Une doc sur la manipulation bas niveau des entrées/sorties du microcontrôleur de l'Arduino est disponible ici : [https://electronoobs.com/eng\\_arduino\\_tut130.php](https://electronoobs.com/eng_arduino_tut130.php).

Dans la fonction `setup()` le registre `TCCR1B` permet de configurer l'horloge à l'entrée du Timer. `CS11` et `CS10 = 1` donnent Horloge/64 d'après la doc. L'horloge de l'ATmega328P de la carte Arduino est configurée à 16MHz. Il faut ensuite calculer les timings pour respecter le tableau donné dans la doc Oregon Scientific (les timings de l'état RF On fonctionnent).

```
//Configuration Timer 1
TCCR1A = B00000000;
TCCR1B = ( _BV(ICNC1) | _BV(CS11) | _BV(CS10) );
SET_INPUT_CAPTURE_RISING_EDGE();
TIMSK1 = ( _BV(ICIE1) | _BV(TOIE1) );
```

Les timings à donner :

```
#define SHORT_PULSE_MIN      50      // 200us
#define SHORT_PULSE_MAX      189     // 758us
#define LONG_PULSE_MIN       190     // 760us
#define LONG_PULSE_MAX       330     // 1320us
```

À la place de 189-190, on peut mettre aussi 153-154 (env 612us).

#### Q 6 Ajouter le module de pluviométrie

1. En utilisant les extraits du protocole Oregon Scientific donnés en annexe (sec. 7.3) et en s'appuyant sur les fonctions déjà présentes dans le programme, proposer une fonction de décodage du module de pluviométrie permettant de transmettre le taux de pluie en pouces par heure ainsi que le total reçu en pouces et en mm.
2. Choisir le format du message transmis sur la liaison série de la carte Arduino en cohérence avec les messages des autres modules.
3. Après validation auprès de l'examineur, coder ces modifications et faire valider le bon fonctionnement.

❖ Le nouveau module est ajouté à partir de la doc Oregon Scientific et des autres fonctions déjà présentes. Il faut juste remarquer que comme le tableau `packet` contient le nibble de sync, l'indice des nibbles est décalé de 1 par rapport à la documentation. Il est demandé au candidat d'ajouter le décodage et l'envoi de 3 données et pas seulement 2 et de choisir le contenu de la trame émise. Par exemple :

```

// PCR800 Rain Gauge Sensor
// Sample Data as received (nibbles LSB first):
// 5 4 9 8 2 CH RC0 RC1 FL RRO RR1 RR2 RR3
// TR0 TR1 TR2 TR3 TR4 TR5 CHK0 CHK1 Postamble
//
// RR3RR2.RR1RRO = Rain Rate inch per hour
// TR5TR4TR3.TR2TR1TR0 = Total Rain inch
// CHK1CHK0 = Checksum
void DecodeRain()
{
    // check the packet CRC
    if (!ValidCRC(19))
    {
        Serial.println("Rain: CRC Error!");
        return;
    }

    // Grab the Rain Rate value
    rain_rate = (GetNibble(12)*10) + GetNibble(11) + \
                GetNibble(10)*0.1 + GetNibble(9)*0.01;

    // Grab the total Rain value
    int total=0;
    for (int x=0;x<6;x++){
        total = total*10;
        total += GetNibble(18-x);
    }
    rain_total = total * 0.001;
    rain_bucket_tips = rain_total / 0.039; // en mm (plutôt 0.042)

    Serial.print("Rain total: ");
    Serial.print(rain_total);

    Serial.print("  rate: ");
    Serial.print(rain_rate);

    Serial.print("  tips: ");
    Serial.println(rain_bucket_tips);
}

```

Le code qui dispatche :

```
case 0x54: DecodeRain(); break;
```

L'octet 0x54 vient du nibble 5 (synchro en LSB First) suivi du nibble 4, qui est la lecture à l'envers du premier 1/2 octet de l'ID tel que donné dans la doc.

Pour un capteur pluvio, les nibbles reçus sont :

Sync	ID0	ID1	ID2	ID3	
A	2	9	1	4	
5	4	9	8	2	<- LSB First

Le code teste le premier octet, donc 54.

Il faut aussi déclarer les 3 vars globales :

```
double rain_bucket_tips = 0;
double rain_total = 0;
double rain_rate = 0;
```

Si le candidat passe trop de temps sur cette première partie, on peut éventuellement lui proposer de fournir un code fonctionnel en programmant la carte Arduino et de passer à la question suivante.

Le code est dans `osv3_decode_rain`

## 5.2 Ajout d'un nouveau module météorologique au logiciel Raspberry Pi (45 min)

Objectif : modifier le programme de la carte Raspberry Pi pour prendre en compte un nouveau module météorologique de pluviométrie.

Dans cette partie, vous devez appliquer toutes les modifications nécessaires pour que les données du nouveau module soient prises en compte par l'application de la Raspberry Pi.

### Q 7 Récupérer les données de pluviométrie sur la Raspberry Pi : conception

1. Choisir et mettre en oeuvre une démarche permettant de vérifier la transmission de la nouvelle trame de l'Arduino vers la Raspberry Pi.
2. À partir du diagramme de classes, noter les modifications à apporter au logiciel de la carte Raspberry Pi.

❖ ⇒ La compilation marche pas, avec les réglages en cours, si la cible est pas Debug (avec les autres cibles on a une erreur sur `pthread`).

Ce développement se fait avec CodeLite sur la Raspberry Pi.

Ajout du module de pluviométrie : Le candidat doit développer une classe `tPluvio` et l'intégrer au programme fourni. Il peut utiliser `gtkterm` pour vérifier la réception de la trame émise par la carte Arduino. Dans le diagramme de classes il suffit d'ajouter une classe module fille sur le modèle des autres modules.

### Q 8 Récupérer les données pluviométrie sur la Raspberry Pi : réalisation

Après validation de la question précédente auprès de l'examineur, coder ces modifications et faire valider le bon fonctionnement.

❖ Il y a pas mal de modifications à faire dans : `tModuleType.h`, `main.cpp`, `tDispatcheur.cpp`. Il faut aussi créer `tPluvio.h` et `tPluvio.cpp`

```
#ifndef T_PLUVIO
#define T_PLUVIO

#include "tModule.h"

class tPluvio : public tModule
{
private:
    vector<float> data;
    vector<tInterfaceData*> ListeObserveurs;
    TypeModule Type = PLUVIO;

public:
    tPluvio();
    void setObserver(tInterfaceData* obs);
    void setData(vector<float> pData);
    vector<float> getData();
    TypeModule getType();
};
#endif
```

Fichier tPluvio.cpp :

```

#include "tPluvio.h"
#include <iostream>
using namespace std;

tPluvio::tPluvio()
{ }

void tPluvio::setData(vector<float> pData)
{
    if (pData.size()==3)
        //if (pData.size()==2)
        {
            data.clear();
            data.push_back(pData[0]);

            if (ListeObserveurs.size() >0)
                { for (int i=0; i<ListeObserveurs.size();i++) {
                    ListeObserveurs[i]->processData(pData,Type);
                }
            }
        }
}

vector<float> tPluvio::getData()
{ //on retourne une copie des données
    return vector<float>(data);
}

void tPluvio::setObserver(tInterfaceData* obs)
{ ListeObserveurs.push_back(obs);
}

TypeModule tPluvio::getType()
{ return PLUVIO;
}

```

Ensuite, il faut ajouter les éléments suivants pour que le nouveau module soit pris en compte :  
Ajouter un type dans la classe tModuleType.h :

```
enum TypeModule{UV, TEMP_HYGRO, GIROUETTE, PLUVIO};
```

Modifier tDispatcheur.cpp pour prendre en compte la nouvelle trame :

```

else if (buffer[0]=='R') //trame Pluvio
{
    for (unsigned int i=0;i<ListeModules.size();i++)
    {
        if (ListeModules[i]->getType() == PLUVIO)
        {
            float total, rate, tips;
            sscanf(buffer,"Rain total: %f rate: %f tips: %s", \
                &total, &rate, &tips);
            vector<float> vtemp;
            vtemp.push_back(total);
            vtemp.push_back(rate);
            vtemp.push_back(tips);
            ListeModules[i]->setData(vtemp);
        }
    }
}

```

Créer un objet de type tPluvio et l'ajouter dans le dispatcheur depuis le main et ajouter tGestion comme observateur à l'objet modulePluvio :

```

#include "tPluvio.h"
...
//On créé un module pour le donner au dispatcheur
tPluvio modulePluvio;
modulePluvio.setObserver(&Gestion);
//on ajoute le module à la liste du dispatcheur.
dispatch.addModule(&modulePluvio);

```

Le candidat peut alors vérifier que l'application reçoit bien les trames et extrait les informations par l'affichage.

### 5.3 Prise en compte de plusieurs capteurs de température (30 min)

⇒ La suite du travail peut être réalisée si cette section n'est pas traitée. Il est donc envisageable de continuer directement en sec. 5.4.

Objectif : Modifier le programme pour accepter et différencier plusieurs modules de température (ou analyser la modification).

Il est possible d'utiliser plusieurs modules de température pour une même station. Pour différencier les données entre ces modules, il faut modifier leur canal (bouton(s) situé(s) dans le compartiment à piles du module).

#### Q 9 Prise en compte du numéro de canal

1. À l'aide de la documentation sur le protocole des stations Oregon donnée en annexe (sec. 7.3), modifier le programme de l'Arduino pour décoder et envoyer le numéro de canal.
2. Modifier le logiciel de la carte Raspberry pour prendre en compte le numéro de canal sur les modules concernés.
3. Tester et valider le bon fonctionnement.

❖ D'après la documentation Oregon Scientific fournie, on peut voir que le canal des capteurs est transmis après le nibble de sync et les 4 nibbles de l'ID du module. Ce canal est en général à 0 sauf pour les capteurs de température hygrométrie sur lesquels on peut changer le canal. Il est donc demandé au candidat d'ajouter un décodage de ce canal et de le transmettre dans la trame série pour pouvoir différencier les capteurs. Il faut ensuite ajouter le décodage de ce canal dans le logiciel Raspberry. On

pourra demander au candidat de filtrer les données en fonction du canal.

## 5.4 Base de données

Objectif : la Raspberry Pi doit enregistrer toutes les données reçues dans la table d'une base de données `sqlite`. Un service Web séparé peut accéder à cette base de données et resservir les informations (valeurs au cours de la journée, courbes, moyennes...).

Dans la table à créer, un enregistrement correspondra à une seule mesure. Par conséquent la réception de données d'un seul capteur peut donner lieu à la création de plusieurs enregistrements (par exemple température et hygrométrie).

Chaque enregistrement de la table contiendra :

- le numéro du canal utilisé par le capteur (ou 0 si ce champs n'est pas utilisé) ;
- le type d'enregistrement (température, hygrométrie...) ;
- la valeur de la mesure ;
- un horodatage.

### Q 10 Modèle de données

Proposer un modèle pour la table en question (type de chaque colonne, clés...) en se basant sur les types disponibles dans `sqlite` : <https://www.sqlite.org/datatype3.html>

❖ Par exemple :

```
"id"          INTEGER, PRIMARY KEY("id" AUTOINCREMENT))
"channel"     INTEGER,
"varname"    TEXT,
"value"      REAL,
"timestamp"  INTEGER,
```

- `channel` est le numéro du canal configuré sur le capteur (si cette partie n'est pas réalisée, mettre systématiquement 0 dans ce champ).
- `varname` est le nom de la variable enregistrée, par exemple :
  - `Temp` pour une température
  - `Hygro` pour l'humidité de l'air
  - `HPluvio` ...
- `value` est la valeur de la variable (un seul nombre)
- `timestamp` est l'horodatage de la mesure en nombre de secondes écoulées depuis le 1er janvier 1970 (ce que renvoie la fonction C : `time(0)`)

On peut envisager d'autres façons de procéder.

⇒ Il est possible dans la suite de modifier le code C++ de la Raspberry Pi pour qu'il peuple la base de données au fur et à mesure de la réception des données des capteurs (questions 11 à 13). Il est aussi envisageable de réaliser uniquement la partie Web (Python), en utilisant une base de données préremplie (questions 14 à 18). Le candidat pourra commencer par la partie qu'il juge la plus intéressante pour la suite de ses développements.

### Q 11 Création de la base de données

Dans le répertoire `stationMeteo`, créer un fichier de base de données `sqlite` nommé `database.db`, contenant une table nommée `mesures` sur le modèle précédemment validé. Pour cela :

1. Commencer par créer le fichier de base de données : `sqlite3 database.db`
2. Puis entrer les commandes SQL permettant de créer la table en utilisant par exemple :
  - Documentation `sqlite` : <https://www.sqlite.org/lang.html>
  - Création d'une table : [https://www.sqlite.org/lang\\_createtable.html](https://www.sqlite.org/lang_createtable.html)

- Tutoriel sur la création des tables : <https://www.sqlitetutorial.net/sqlite-create-table/>

❖ ⇒ Penser à installer `sqlite3` : `sudo apt-get install sqlite3`

```
cd stationMeteo/Debug #<<<<< ATTENTION, Le code C++ tourne dans le rép Debug
sqlite3 database.db
CREATE TABLE IF NOT EXISTS "mesures" (
    "id"          INTEGER,
    "channel"     INTEGER,
    "varname"     TEXT,
    "value"       REAL,
    "timestamp"  INTEGER,
    PRIMARY KEY("id" AUTOINCREMENT));
```

### Q 12 Enregistrement dans la base de données

En s'inspirant de l'exemple donné en annexe (sec. 7.4), ajouter la classe `tBDD` représentée dans le diagramme de classe (sec. 7.1, fig. 6) et l'utiliser dans le logiciel `stationMeteo` pour qu'il enregistre les données reçues par les capteurs dans la base qui vient d'être créée. Il est conseillé de proposer des choix d'implémentation aux examinateurs avant de se lancer dans l'écriture du code. Le module `tGestion` devra être modifié.

⇒ Il sera probablement nécessaire d'ajouter la bibliothèque `sqlite3` lors de l'étape d'édition de liens (Options du projet / Settings / Linker).

- ❖ • Le traitement des données est réalisé par la classe `tProcessData`.
- Ajouter une classe `tBDD` qui réalise l'enregistrement dans la base.
- Les données étant reçues à un rythme peu soutenu, le programme tournant sur la Raspberry Pi rouvrira et refermera la connexion à la BDD à chaque enregistrement.

⇒ Penser à installer la lib `sqlite3` : `sudo apt-get install libsqlite3-dev`. Par ailleurs, il faut ajouter la lib dans l'éditeur de liens : Options du projet (gears au dessus des fichiers) / Settings / Linker / et ajouter `sqlite3` en plus `depthread`.

```
/* dans tGestion.h */

#include "tBDD.h" // AJOUT
using namespace std;
class tGestion : public tInterfaceData
{
private :
    tBDD* bdd ; // AJOUT
```

```

/* dans tGestion.cpp */
tGestion::tGestion()
{
    bdd = new tBDD("database.db"); // AJOUT
}

void tGestion::processData(vector<float> pData, TypeModule pType )
{
    //Affichage uniquement pour debug
    ...

    // AJOUT
    cout << "Ecriture dans la base..." << endl;
    switch (pType) {
        case TEMP_HYGRO: // Temp and Hygro
            // 0 canal par défaut
            bdd->enregistrer_temp_hygro(time(0), 0, pData[0], pData[1]);
            break;
        default:
            cerr << "Pas de méthode d'écriture" << endl;
    }
}

```

```

/* fichier tBDD.h */
#ifndef tBDD_H
#define tBDD_H
#include <time.h>
#include <sqlite3.h>

class tBDD {

private:
    char filename[256];

public:
    tBDD(const char* filename);
    int enregistrer(time_t timestamp, int channel,
                    const char * varname, float value);
    int enregistrer_temp_hygro(time_t timestamp,
                               int channel, float temp, float hygro);
    int enregistrer_uv(time_t timestamp, int channel, float uv);
    int enregistrer_pluvio(time_t timestamp, int channel,
                           float total, float rate, float tips);
};

#endif

```

```

/* fichier tBDD.cpp */
#include "tBDD.h"
#include <string.h>
#include <time.h>
#include <iostream>
using namespace std;

tBDD::tBDD(const char* fname)
{
    strcpy(filename, fname);
}

int tBDD::enregistrer(time_t timestamp, int channel, const char* var, float value)
{
    sqlite3 *db;
    char *zErrMsg = 0;
    int rc;
    char sql[1024];

    /* Open database */
    rc = sqlite3_open(filename, &db);
    if (rc) {
        cerr << "Can't open database: " << sqlite3_errmsg(db) << endl;
        return -1;
    }
    cout << "Database opened" << endl;

    /* Create SQL statement */
    sprintf(sql, "INSERT INTO MESURES (channel, varname, value, timestamp) "\
        " VALUES (%d, \"%s\", %.1f, %lu );",
        channel, var, value, timestamp);

    /* Execute SQL statement */
    rc = sqlite3_exec(db, sql, NULL, NULL, &zErrMsg);

    if (rc != SQLITE_OK) {
        cerr << "SQL error: " << zErrMsg << endl;
        sqlite3_free(zErrMsg);
        return -2;
    }

    cout << "Records created successfully" << endl;

    sqlite3_close(db);
    // free db ?
    return 0;
}

```

Suite du fichier :

```

int tBDD::enregistrer_temp_hygro(time_t timestamp, int channel,
                                float temp, float hygro) {
    int r;
    r = enregistrer(timestamp, channel, "Temp", temp);
    if (r < 0) return r;
    return enregistrer(timestamp, channel, "Hygro", hygro);
}

int tBDD::enregistrer_uv(time_t timestamp, int channel, float uv) {
    return enregistrer(timestamp, channel, "UV", uv);
}

int tBDD::enregistrer_pluvio(time_t timestamp, int channel, float total,
                              float rate, float tips) {
    int r;
    r = enregistrer(timestamp, channel, "RainTotal", total);
    if (r < 0) return r;
    r = enregistrer(timestamp, channel, "RainRate", rate);
    if (r < 0) return r;
    return enregistrer(timestamp, channel, "RainTips", tips);
}

```

### Q 13 Test de l'enregistrement

Montrer un protocole de test permettant de s'assurer du bon fonctionnement de l'enregistrement des données.

❖ On met en route tout le système, puis on se connecte à la base :

```
sqlite3 database.db
```

On fait une première requête pour voir l'état des enregistrements :

```
select * from mesures;
```

Dès réception d'un capteur, on vérifie manuellement que les données ont bien été ajoutées :

```
select * from mesures;
```

## 5.5 Service Web

Objectif : Terminer la réalisation d'un service Web (programme séparé) qui sert les données des capteurs enregistrées dans la base de données (valeurs au cours de la journée, courbes, moyennes...).

La restitution est réalisée à l'aide d'un service Web programmé en Python utilisant le module `flask`. Ce service répond à des requêtes Web, et pioche les données dans la base `sqlite`.

Une base de données exemple et déjà préremplie est disponible à la demande.

❖ Le fichier est caché : `application_web/.database.db` Le fournir au candidat s'il n'a pas pu faire d'enregistrement suffisamment intéressant de son côté.

```
cd application_web
mv .database.db database.db
```

### Q 14 Exécution de l'application flask

L'application Web est disponible sur la Raspberry Pi dans le répertoire `application_web`. Le script Python est nommé `webapp.py`.

Sur la Raspberry Pi, exécuter cette application :

```
$ python3 webapp.py
* Serving Flask app "webapp" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 555-235-234
```

Puis consulter la page depuis le poste de travail : `http://<ip raspberry pi>:5000` et vérifier qu'elle s'affiche sans erreur (la page contient seulement le titre Menu).

Pour comprendre l'architecture de cette application, ouvrir le fichier `webapp.py` dans le répertoire `application_web`. Le principe est d'associer une fonction Python à une URL (appelée *route* dans le vocabulaire `flask`). Il est habituel, quoi que non obligatoire, que le nom de la fonction soit le même que celui de la route. La consultation de l'URL provoque l'exécution de la fonction associée, et ce que renvoie la fonction est affiché dans le navigateur (il s'agit généralement d'une page HTML). Par exemple :

```
@app.route('/data')
def data():
    # Faire des choses intéressantes
    return <chaîne de caractères contenant du HTML>
```

#### Q 15 Modification du menu

Rechercher les deux fonctions qui correspondent aux URLs :

- `http://192.168.100.10:5000/` ou `http://192.168.100.10:5000/index.html`
- `http://192.168.100.10:5000/data?varname=Temp&channel=0`

Noter comment, dans le cas de la seconde URL, deux paramètres sont passés par le biais de la requête.

Dans la fonction associée à la première URL, la variable `page` contient l'essentiel du corps de la page Web. Modifier le contenu de cette variable de manière à ce qu'elle contienne un lien cliquable vers :

- `http://192.168.100.10:5000/data?varname=Temp&channel=0`

Ce lien ne doit pas être un lien absolu, mais un lien relatif, car l'application pourrait être déplacée. Vérifier que votre modification est prise en compte en consultant à nouveau la page. Si ce n'est pas le cas, couper l'application serveur avec `Ctrl+C`, puis la relancer.

```
❖ @app.route('/')
  @app.route('/index.html')
  def index():
      page = """\
      <h1>Menu</h1>

      <a href="data?varname=Temp&channel=0">Température, channel 0</a>
      """
      return HEADER + page + FOOTER
```

#### Q 16 Affichage des dernières valeurs enregistrées

Modifier le fichier `webapp.py` pour que la consultation de l'URL :

`http://192.168.100.10:5000/data?varname=Temp&channel=0` affiche les 10 dernières valeurs de la variable `Temp`, sur le canal 0 (ou tout autre canal qui correspondrait aux enregistrements que vous avez pu

réaliser). L'annexe 7.5 donne un exemple de consultation d'une base de données `sqlite3` en Python. Faire en sorte que le timestamp des mesures s'affiche de manière lisible (par exemple 01/06 15:00 au lieu de 1685624400), en utilisant éventuellement le module `datetime` de Python.

```
❖ import datetime

def get_data(varname, channel, last):
    db = sqlite3.connect("database.db")
    c = db.cursor()
    req = c.execute("select timestamp, value from mesures "
                   "where varname=? and channel=? "
                   "order by timestamp desc LIMIT ?", (varname, channel, last))
    res = req.fetchall()
    db.close()
    return res

@app.route('/data')
def data():
    varname = request.args.get('varname', type = str)
    channel = request.args.get('channel', default = 0, type = int)
    page = "<h1> {varname}, {channel}</h1>".format(varname=varname, channel=channel)
    res = get_data(varname, channel, 10)
    # print(res)
    page += "<ul>"
    for data in res:
        ldate=datetime.datetime.fromtimestamp(data[0]).strftime("%d/%m à %H:%M:%S")
        page += "<li>{ldate}{}

```

#### Q 17 Test complet

Tester le système complet. L'émission d'un nouveau capteur doit se solder par l'apparition de la nouvelle donnée dans la page Web. Noter qu'à l'aide d'un peu de Javascript, il est possible de faire en sorte que cette page se remette à jour automatiquement au bout de quelques secondes ou minutes.

❖ Dans la page Web :

```
<script>
function refresh() {
    window.location.reload(true);
    //window.location.href = "...";
}
setTimeout(refresh, 30000); // toutes les 30 secondes
</script>
```

#### Q 18 Améliorations

Il est envisageable de servir l'information sous des formes plus conviviales :

- courbes ;
- moyenne journalières ;
- relevés annuels ;
- etc.

## Épreuve d'admission de soutenance d'un dossier industriel

### A. Présentation de l'épreuve

#### Texte de référence

<http://www.devenirenseignant.gouv.fr/cid98734/les-epreuves-de-l-agregation-externe-section-sciences-industrielles-de-l-ingenieur.html>

- Durée de la préparation des moyens de l'exposé : 30 minutes
- Durée totale de l'épreuve : 1 heure (présentation : 30 minutes maximum, entretien avec le jury : 30 minutes)
- Coefficient 2

L'épreuve consiste en la soutenance devant le jury d'un dossier technique et scientifique réalisé par le candidat dans un des domaines de l'option préparée, suivie d'un entretien.

L'épreuve a pour but de vérifier que le candidat est capable de rechercher les supports de son enseignement dans le milieu économique et d'en extraire des exploitations pertinentes pour son enseignement en cycle terminal du lycée, en classes préparatoires aux grandes écoles, en sections de techniciens supérieurs et instituts universitaires de technologie.

Le dossier présenté par le candidat est relatif à un système technique dont la dominante est choisie par le candidat. Son authenticité et son actualité sont des éléments décisifs.

L'exposé et l'entretien permettent d'apprécier l'authenticité et l'actualité du problème choisi par le candidat, sa capacité à en faire une présentation construite et claire, à mettre en évidence les questionnements qu'il suscite et à en dégager les points remarquables et caractéristiques. Ils permettent également au candidat de mettre en valeur la qualité de son dossier et l'exploitation pédagogique qu'il peut en faire dans le cadre de son enseignement.

En utilisant les moyens courants de présentation (vidéoprojecteur et informatique associée, en particulier), le candidat présente le support technique qu'il a choisi pour l'épreuve ainsi que les investigations et développements qu'il a conduits pour s'en approprier totalement le fonctionnement et les évolutions potentielles. Lors de la présentation, le candidat justifiera le choix du support d'études et les investigations conduites qui pourraient, selon lui, donner lieu à des exploitations pertinentes.

Pendant l'entretien, le jury conduit des investigations destinées à se conforter dans l'idée que le dossier présenté résulte bien d'un travail personnel du candidat et s'en faire préciser certains points.

Les éléments constitutifs du dossier sont précisés par note publiée sur le site internet du ministère chargé de l'Éducation. Les dossiers doivent être déposés au secrétariat du jury cinq jours francs au moins avant le début des épreuves d'admission.

## B. Attendus de l'épreuve

L'épreuve de dossier nécessite anticipation, développements et préparation spécifique. Les candidats n'ont pas à présenter leur parcours professionnel lors de cette épreuve.

### • Construction d'un dossier technique et développement scientifique

- un **partenariat réel et actuel** avec une entreprise est attendu. Les éléments témoignant des échanges entre le ou la candidate et l'entreprise constituent un gage d'authenticité ;
- il est souhaitable, dans la mesure du possible, qu'un accord de confidentialité soit signé entre les parties prenantes, afin de protéger les intérêts réciproques. En cas de confidentialité réclamée, il est impératif de la mentionner clairement dans le dossier ;
- le dossier porte sur un système authentique et actuel, relevant d'un domaine significatif de l'ingénierie informatique en sciences industrielles de l'ingénieur, en lien avec des enjeux sociétaux. Les systèmes novateurs sont appréciés ;
- la ou les problématiques scientifiques et/ou technologiques liées au système sont à exprimer avec clarté et précision ;
- le candidat doit développer un contenu scientifique et technologique personnel d'ingénierie informatique, référencé dans le dossier et relevant du niveau master 2, avec une démarche rigoureuse fondée sur des bases scientifiques (théories, publications référencées, brevets, etc.) ;
- la modélisation, l'implémentation, la programmation, l'architecture des systèmes, la communication (réseaux, IoT, IHM, IA, etc.) constituent des éléments essentiels de l'ingénierie informatique. Il est attendu du candidat qu'il utilise les outils « métiers » de l'ingénierie informatique adaptés aux problèmes traités ;
- une analyse critique (qualitative, quantitative), une prise de recul, et des pistes d'amélioration en lien avec la problématique doivent être présentées ;
- une simple analyse fonctionnelle ou descriptive d'un produit est hors sujet.

### • Proposition d'activités pédagogiques

- les exploitations pédagogiques doivent être en cohérence avec le système étudié, la problématique et les développements scientifiques et technologiques ;
- elles doivent être en lien avec les référentiels ou programmes choisis, au regard de la spécialité ingénierie informatique, des compétences et connaissances associées, ainsi que de l'organisation matérielle des enseignements ;
- les séquences proposées doivent s'inscrire dans une progression générale clairement formalisée dans le cycle de formation choisi en lien avec le référentiel ou programme ciblé ;
- la séquence pédagogique proposée se doit d'être complète autant de par sa structure, sa démarche, les périodes de différenciations éventuelles et les remédiations ;
- au moins une activité doit être développée en cohérence avec une des séquences pédagogiques proposées ;
- la description des activités doit permettre aux membres du jury de percevoir ce que font les élèves ;
- des propositions d'exploitations pédagogiques dans une perspective d'activités pluri-technologiques ou interdisciplinaires sont attendues ;
- des modalités d'évaluation des compétences, justifiées, prenant en compte la diversité des élèves et l'inclusion scolaire, doivent être présentées ;
- le niveau de maîtrise des outils « métiers » ainsi que collaboratifs à des fins pédagogiques est à mettre en évidence lors des propositions d'activités ;
- le candidat apportera dans cette partie la preuve de la mise en œuvre des compétences issues du référentiel des compétences professionnelles des métiers du professorat et de l'éducation ;
- le développement pédagogique est l'occasion de mettre en évidence **le partage des valeurs de la république.**

- **Communication**

- **Dossier écrit**

La forme et la qualité de réalisation du dossier montrent que le candidat a réfléchi à la teneur du message qu'il souhaite communiquer aux membres du jury. Pour cela :

- les règles de citation des sources (des tables, illustrations, articles, références ...) doivent être respectées ;
- le dossier doit comporter un titre, un sommaire, une conclusion et être paginé ;
- les tableaux, graphiques et annexes doivent être correctement référencés et récapitulés dans des tables dédiées ;
- le dossier ne doit en aucun cas être un manuel d'utilisation ou une documentation commerciale ;
- le dossier doit être rédigé dans une langue française soignée en tenant compte des règles grammaticales et orthographiques ;
- les développements et résultats des expérimentations et mesures réalisées doivent être référencés dans le dossier et présent en annexes ;
- le dossier doit résulter d'un travail personnel du candidat.

- **Présentation orale**

Durant l'exposé, le candidat doit mettre en valeur ses qualités de communicant pour expliquer ses choix, ses démarches et ses analyses, dans le champ de l'ingénierie informatique. Les documents vidéoprojetés doivent être lisibles, numérotés, et de bien structurés. Le candidat peut utiliser tout support permettant d'attester des réalisations (vidéo de présentation ou lien vers des démonstrations). Le choix des éléments présentés à l'oral est important. **Le jury veillera à ce que le candidat ne dépasse pas les 30 minutes autorisées.**

L'entretien de **30 minutes qui suit**, permet d'approfondir certains points abordés dans le dossier ou durant l'exposé. Cet échange porte tant sur les développements scientifiques et technologiques que sur les propositions d'exploitation pédagogique qui en découlent.

## **C. Commentaires du jury**

Pour la session 2025, le jury constate :

- que certains candidats n'énoncent pas clairement la ou les problématiques traitées, ni la méthodologie employée ;
- qu'une majorité de développements scientifique et technique ne sont pas d'un niveau M2 voire M1 ;
- qu'un nombre non négligeable de séquences présentées ne tiennent pas compte de la diversité et de l'hétérogénéité des élèves ;
- que certaines activités sont présentées sans lien avec les compétences visées par le programme ou le référentiel ;
- que la pluridisciplinarité, bien que présente dans les textes, soit peu envisagée par les candidats ;
- que peu de candidats proposent des outils de suivi individualisé des compétences ;
- que rares sont les activités pédagogiques détaillées comprenant des documents et des productions d'élèves ;
- que les compétences professionnelles du référentiel des métiers du professorat et de l'éducation sont rarement mobilisées explicitement.

Le jury valorise :

- les dossiers comportant un développement scientifique de niveau master en informatique mettant en œuvre la démarche de l'ingénieur et la démarche de projet ;
- les séquences pédagogiques intégrées dans une progression pédagogique mobilisant les compétences sur un cycle de formation complet ;
- l'usage pertinent des outils numériques pédagogiques (ENT, espace collaboratif, classe inversée, temps hors la classe, etc.) ;
- les dossiers présentant un développement pédagogique en pré-bac et en post-bac.

Enfin, le jury cherche à évaluer la capacité du candidat, en qualité de futur agent du service public d'éducation, à prendre en compte la diversité des conditions d'exercice du métier et à incarner les valeurs de la République.