MINISTÈRE
DE L'ÉDUCATION
NATIONALE,
DE LA JEUNESSE
ET DES SPORTS

Liberté
Égalité
Fraternité

Rapport du jury

Concours : AGRÉGATION EXTERNE

Section : sciences industrielles de l'ingénieur

Option : ingénierie informatique

Session 2026

Rapport de jury présenté par : Florian Kolbl, président du jury
Maître de Conférences

© https://www.devenirenseignant.gouv.fr

Table des matières

1 Épreuves d'admissibilité **6**
1.1 Épreuve de sciences industrielles de l'ingénieur 6
1.2 Épreuve de conception d'un système, d'un procédé ou d'une organisation 18
1.3 Épreuve de modélisation d'un système, d'un procédé ou d'une organisation 62

2 Épreuves d'admission **88**
2.1 Épreuve d'exploitation pédagogique d'une activité pratique relative à l'approche globale d'un système pluritechnologique 88
2.2 Épreuve d'exploitation pédagogique d'une activité pratique relative à l'approche spécialisée d'un système pluritechnologique 106
2.3 Épreuve de soutenance d'un dossier industriel 120

Avant-propos

Le concours 2026 marquait la dixième session de l'option « ingénierie informatique » de l'agrégation de sciences industrielles de l'ingénieur (SII). Cette option a été créée en 2017 afin de répondre à une problématique de recrutement d'enseignants pouvant intervenir dans le cadre de programmes exigeants liés à l'informatique et, plus généralement, aux sciences du numérique.

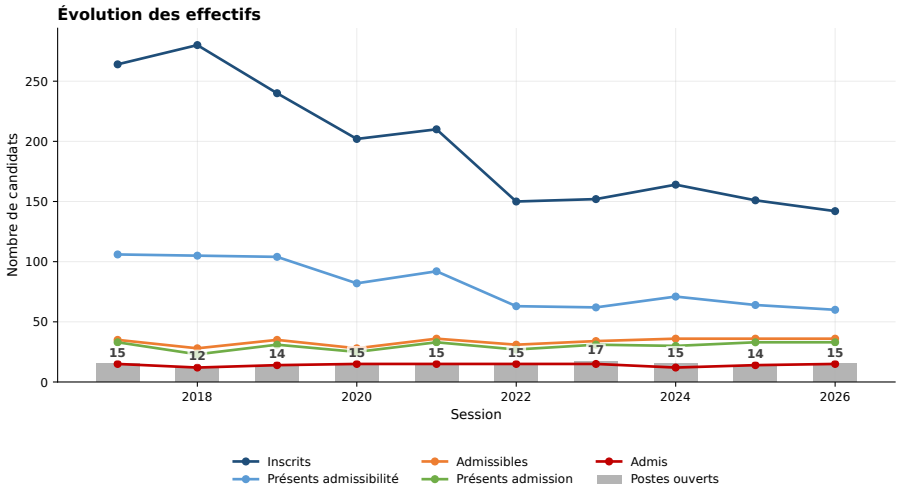
L'ingénierie informatique appliquée aux systèmes analyse et résout des problèmes scientifiques et technologiques communs à l'informatique dans sa globalité, mais également spécifiques à l'informatique embarquée (logicielle et matérielle). Ces problèmes s'inspirent de défis sociétaux liés non seulement à l'utilisation massive d'objets connectés (IoT), toujours plus intelligents et à faible empreinte énergétique, mais également à la généralisation de grands modèles d'intelligence artificielle, qui nécessitent des quantités importantes de données et présentent une forte empreinte énergétique.

Ces spécificités et contraintes, toujours plus nombreuses, concernent notamment le temps réel, la miniaturisation, l'autonomie et la consommation énergétique, les interfaces homme-machine, l'acquisition et la numérisation de grandeurs physiques analogiques au moyen de capteurs, le traitement des données, les réseaux informatiques et de télécommunication, la cybersécurité, la fiabilité, la sûreté de fonctionnement, etc.

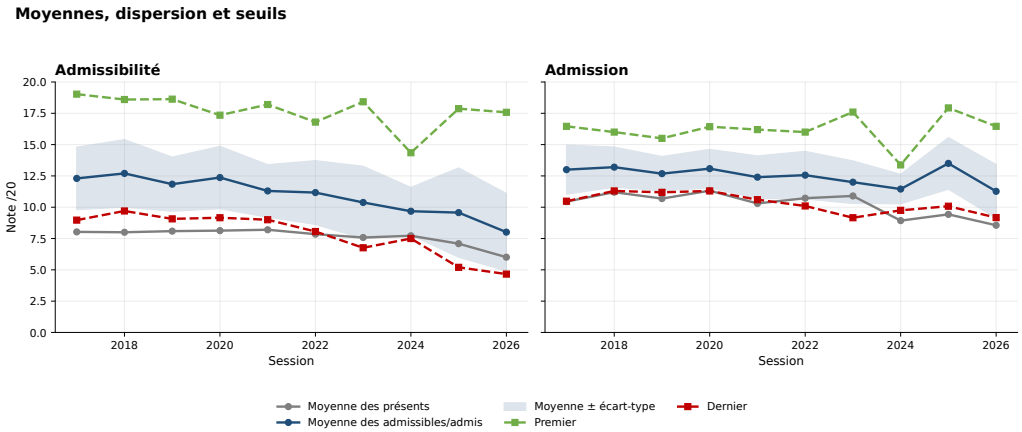
Les attentes du concours de l'agrégation SII sont définies par l'arrêté du 28 décembre 2009 fixant les sections et les modalités d'organisation des concours de l'agrégation. Les concours de recrutement d'enseignants n'ont pas pour seul objectif de valider les compétences scientifiques et technologiques des candidats ; ils doivent aussi valider les compétences professionnelles attendues par l'État employeur. 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. **Il est à noter que, pour la prochaine session, le programme évalué dans la spécialité « ingénierie informatique » a été adapté afin de suivre l'évolution des formations concernées et d'inclure, de manière plus explicite, des éléments liés à l'intelligence artificielle et à la cybersécurité.**

Les trois épreuves d'admissibilité sont construites de manière à évaluer un large spectre de compétences scientifiques et technologiques : la première épreuve est commune aux quatre options de l'agrégation SII, tandis que les deux autres sont spécifiques à l'option. Les trois épreuves d'admission complètent les épreuves d'admissibilité : la première est commune aux quatre options, tandis que les deux autres sont 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*



(a) Évolution des effectifs du concours.



(b) Évolution des moyennes, de la dispersion et des seuils.

Figure 1 – Évolution des effectifs et des résultats du concours au cours des dix premières sessions.

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 dixième session, 15 postes ont été ouverts pour l'agrégation de sciences industrielles de l'ingénieur, option « ingénierie informatique ». Parmi les 142 inscrits, 60 candidats se sont présentés aux trois épreuves d'admissibilité. Le nombre d'inscrits reste globalement stable par rapport aux quatre années précédentes, bien qu'il tende à diminuer légèrement. L'édition 2022 avait connu une baisse significative du nombre d'inscrits, très probablement en raison de la création de l'agrégation d'informatique, qui a pu drainer une partie des candidats. L'observation de ces deux agrégations distinctes au cours des dernières années montre que les effectifs sont globalement stables et que les candidats identifient bien ces deux concours, distincts par leurs programmes respectifs. L'évolution des effectifs au cours des dix premières sessions est présentée dans la figure 1a.

Cette session s'est révélée être d'un bon niveau. Les candidats retenus ont su démontrer leur maîtrise des savoirs et des 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. Les 15 postes ouverts ont été pourvus et le candidat classé 16^{e} a été inscrit sur liste complémentaire au regard de son niveau académique. L'évolution des moyennes, de leur dispersion et des seuils d'admissibilité et d'admission est présentée dans la figure 1b.

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

Lors de cette session 2026, le lycée La Martinière Monplaisir, à Lyon, a accueilli les épreuves d'admission 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 dans 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
2025	151	14	64	36	33	15*
2026	142	15	60	36	33	15*

* 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 2026

Note obtenue par le premier candidat admissible	**17,58**
Note obtenue par le dernier candidat admissible	**4,66**
Moyenne des candidats présents	6,01
Moyenne des candidats admissibles	8,01
Écart-type des candidats admissibles	3,09

Statistiques des notes obtenues à l'admission à la session 2026

Note obtenue par le premier candidat admis	**16,46**
Note obtenue par le dernier candidat admis	**9,17**
Note obtenue par le dernier candidat sur liste complémentaire	**9,07**
Moyenne des candidats présents	8,56
Moyenne des candidats admis	11,27
Écart-type des candidats admis	2,14

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/19056/download

Le porte-conteneurs Meltem

D'après l'Organisation maritime internationale (OMI), le transport maritime représente au niveau mondial près de 3% des émissions de gaz à effet de serre.

Moyen de transport efficace d'un point de vue énergétique rapporté aux volumes transportés, le transport maritime n'en est pas moins quasi exclusivement dépendant des énergies fossiles. Il doit aujourd'hui opérer une transition énergétique sans précédent afin de respecter les engagements fixés au niveau international et européen, mais aussi contribuer à l'effort national de décarbonation. Pour répondre aux objectifs, plusieurs technologies sont à l'étude comme l'utilisation de piles à combustibles, la motorisation électrique et la propulsion vélique. C'est cette dernière qui est étudiée dans la suite.

Le porte-conteneurs Meltem (Figure 1), développé par l'entreprise Zéphyr Borée, est doté de quatre paires d'ailes, conçues par l'entreprise CWS. Il possède une passerelle 3/4 avant, de manière à pallier l'obstruction de la visibilité sur l'avant du navire. La propulsion mécanique est assurée par deux moteurs thermiques de 4 800 kW. Afin de limiter la hauteur du centre de voilure, la hauteur maximum des bays de conteneur est réduite par rapport à un navire conventionnel de mêmes dimensions. La capacité maximale géométrique est donc moindre, mais la capacité de chargement à 14 tonnes homogènes reste identique à celle d'un navire de mêmes dimensions sans ailes.



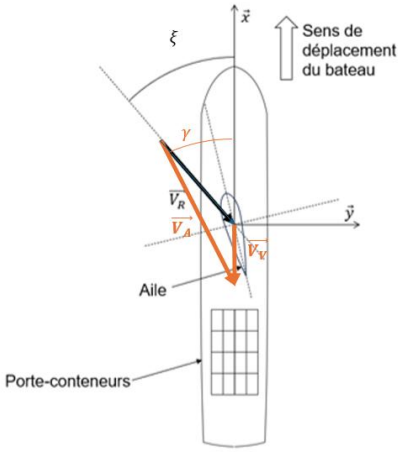
Figure 1 – Meltem Porte conteneur de Zéphyr et Borée, dotée de 4 paires d'ailes CWS.
Source : lemarin.ouest-France.fr

C. Éléments de correction

Partie 1 : Pertinence de l'usage d'ailes rigides

Q1.

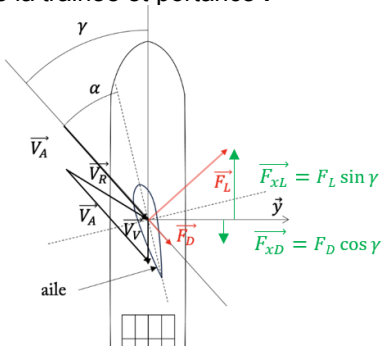
$\overrightarrow{V_V}$= 22,2 km.h^{-1} soit 2,2 cm
$\overrightarrow{V_A}$ (5,24cm en y et 2,68cm en x) soit 5,9cm. $\overrightarrow{V_A} = 59$ km.h^{-1}
γ =63°



Q2. $\gamma = \tan^{-1}\left(\frac{V_R \sin\xi}{V_R \cos\xi + V_V}\right)$ ou $\gamma = \sin^{-1}\left(\frac{V_R \sin\xi}{\sqrt{(V_R \sin\xi)^2 + (V_R \cos\xi + V_V)^2}}\right)$

$V_A = \sqrt{(V_R \sin\xi)^2 + (V_R \cos\xi + V_V)^2}$

Q3. Construction graphique de la trainée et portance **:**



Q4. $F_x = F_L \sin\gamma - F_D \cos\gamma$

Q5. $F_x = \frac{1}{2}\rho S V_A^2((a\alpha + b)\sin\gamma - (C_{D0} + c\alpha^2)\cos\gamma)$

Q6. Expression mathématique d'un extremum : $\frac{dF_x}{d\alpha} = 0$ $donc$ $2c\alpha\cos\gamma + a\sin\gamma = 0$.

$$\alpha = \frac{a\sin\gamma}{2c\cos\gamma} = \frac{a}{2c}\tan\gamma$$

$$\alpha = \frac{a}{2c}.\frac{V_R \sin\xi}{V_R \cos\xi + V_V}$$

Q7. Plage de variation de ξ : + ou – 45°
Plage de variation de α:

$\alpha = \frac{a}{2c}.\frac{V_R \sin\xi}{V_R \cos\xi + V_V}$ avec V_R =40 km.h^{-1}, V_V =22.2 km.h^{-1}

$\alpha = \frac{a}{2c}.\frac{V_R.\sqrt{2}/2}{V_R\sqrt{2}/2 + V_V} = 0.28 * \frac{a}{c}$ ou -0.28 $* \frac{a}{c}$

On souhaite avoir une précision correcte sur la variation de α, avec un pilotage du moteur cohérent et donc avoir un rapport de réduction suffisant pour cela.

Q8. Puissance moyenne propulsive vélique : $P_{moy} = 8 * F_x * V_V$= 543,25 kW

Q9. Expression de l'énergie moyenne produite par les ailes :

$E1 = 0.6 * P_{moy} * t$=85 073 kWh avec t=261h.

Q10. Expression de l'énergie moyenne consommée par les deux moteurs thermiques :

$E2 = 2.P_{moteur} \times t = 2\,505\,600\ kWh$

Q11. Expression du gain de fioul : $G_{f\%} = \frac{E1}{E2}.100 = 3.4\%$

Q12. Gain en $C0_2$: $G_{CO2} = 49$ t.CO_2

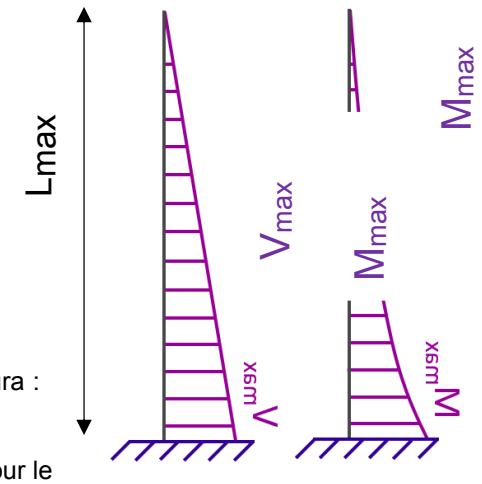
Q13. La présence d'aile n'ajoute que peu de poids supplémentaire et enlève un très faible volume de conteneurs. D'un point de vue écologique, elles permettent de réduire significativement l'empreinte carbone des navires. La baisse de consommation de fioul des moteurs thermiques permet également de générer un gain économique significatif avec l'augmentation des prix des carburants. La principale limite de ce système réside dans le fait que les ailes rigides ne fonctionnent qu'à de faibles vitesses de navigation. Ainsi, pour pouvoir maximiser les réductions de consommations, il est nécessaire d'augmenter les délais de livraison.

Partie 2 : Prédimensionnement du mât de l'aile

Q14. Sollicitations mécaniques : Compression (sous son poids propre+ équipement), flexion simple (sous la pression du vent). La torsion a été également acceptée.

Q15. Calcul des charges :
Calcul de G seule : G=50*9.81=490,5kN
Calcul de $P_{aéro}$ seule : P_{aero}=3601,5 N.m^{-2}
Prise en compte des coefficients de sécurité sur les deux actions :
W=11885 N.m^{-1}
G=663kN

Q16. 2 tronçons ici :
Coupure 1 : $z \in [17; 38m]$:
N=$-G$
T_y =$-W.(L-z)$
$M_x(z) = -W.(L-z)^2/2$
Coupure 2 : $z \in [0; 17m]$
N=0
T_y= $-W.(L-z)$
$M_x(z) = -W * \frac{(L-z)^2}{2}$
(Valeurs maximum : M_{max}=-8580970N.m,
Tmax= -451630,0 N)



Q17. Au niveau du système de fermeture, à x=18m, on aura :
T_y =237,7 kN.m
M_x =2377 kN.m
Note : la modélisation considère une charge ponctuelle pour le poids à 17m, alors qu'en réalité le poids est réparti et des efforts de compression existent au niveau du système de fermeture.

Q18. Un profilé creux présente un moment quadratique plus élevé à masse équivalente.

Q19. Calcul de la surface minimale S_{min} permettant de reprendre la compression seule :

$$S_{min} = \frac{G}{\sigma_{c,max}} = \frac{663000}{355} = 1868\ mm^2$$

Surface du carré creux : $S = 700 * 700 - 600 * 600 = 130000\ mm^2$
$S > S_{min}$ Le profilé convient. Le flambement devra être ensuite vérifié par la suite.

Q20. Vérification du critère de dimensionnement :
Expression et valeur de la contrainte normale de compression $\sigma_c = \frac{G}{S} = 5{,}1 MPa$
Expression et valeur de la contrainte normale de flexion $\sigma_{fx} = \frac{M_{fx}}{I_{Gx}}.\vartheta = 326{,}2\ MPa$
(AN : $I_{Gx} = \frac{a^4}{12} - \frac{(a-2e)^4}{12} = 9{,}2084.10^9$ mm^4 , $\vartheta = 0{,}35m$, $M_{fx} = 8580{,}97\ kN.m$)
$\sigma_c + \sigma_f = 331{,}3\ arrondi$ à $332 MPa \leq 355\ MPa$ car on utilise de l'acier S355.
Le profilé étant en flexion composée, on aurait pu vérifier le déversement et le flambement.

Q21. Calcul de la flèche

Expression de la flèche $v(z)$= $\frac{W}{EI_{Gx}}.(L^2.\frac{z^2}{4} - L.\frac{z^3}{6} + \frac{z^4}{24})$

Maximale en L : $v(L) = \frac{W}{EI_{Gx}}.\frac{L^4}{8} = 1{,}6m$

Q22. Vérification des hypothèses

1ère hypothèse : calcul de l'angle d'inclinaison α= 2,29°, donc cos(α)=0.998, donc 1% d'erreur sur la valeur de l'effort.

2ème hypothèse : angle de 2.29° donc la longueur AB proposée sur le modèle serait de 38,03m. Soit 0.08% d'erreur sur la longueur.

Q23. Equation du moment fléchissant, sur le modèle déformé :

Coupure 1 : $z \in [17; 38m]$: $M_x(z) = -W.(L-z)^2/2$

Coupure 2 : $z \in [0; 17m]$ $M_x(z) = -W * \frac{(L-z)^2}{2} - G * \sin\alpha\,.(17-z)$

Valeur maximale : 9031,3 kN.m

Q24. Ecart relatif entre le modèle déformé et non déformé

Situation déformée : M_{max}= M(0)= 9031,3 kN.m

Situation non déformée : M_{max}= M(0)= 8581,0 kN.m

Écart relatif de 5% : Écart faible entre les deux modèles, mais le modèle déformé est plus pénalisant. Pour aller dans le sens de la sécurité, il est donc nécessaire de prendre la situation déformée en compte.

Q25. Comparaison des modèles en contraintes :

- Étude en grands déplacements (modélisation SolidWorks) : σ_{fx} =355,2 MPa
- Étude en petits déplacements : σ_{fx} =337,7 MPa

Écart relatif : 4.9 %, dans le sens négatif, c'est-à-dire qu'on sous-estime la contrainte réelle. Il est donc dangereux de ne pas se mettre en grands déplacements.

Vérification du profilé : en théorie non, en pratique on est très proche de la limite élastique donc on pourrait accepter pour éviter un surdimensionnement inutile.

Partie 3 : Dimensionnement, consommation et commande des moteurs de pivotement des ailes

Q26. Coefficient aérodynamique maximum angle d'incidence associé

$C_{Mz\,max}$=0,7 à l'angle d'incidence + 30°

Q27. Expériences et simulations

Les simulations aérodynamiques demandent beaucoup de temps de calcul et sont énergivores. Concernant l'essai en soufflerie, une fois l'installation opérationnelle, il est très facile et rapide de faire varier l'angle d'incidence et d'acquérir les données. La simulation est validée car les écarts sont plutôt faibles entre ses résultats et les résultats en soufflerie. Il sera alors possible d'utiliser le modèle de simulation pour faire varier la géométrie et les dimensions sans avoir besoin de réaliser une nouvelle maquette réelle.

Q28. Moment aérodynamique maximum

On trouve : $M_{z_{max}} = 1540\ kN \cdot m$ (3 chiffres significatifs maximum).

Q29. Couple moteur nécessaire

C_m = 10+ 2 000 000 / 1200 = 10+1670 = 1680 N·m.

Le couple de démarrage du réducteur est négligeable lors de vents forts.

Q30. Moteur choisi

Attention, le couple de démarrage n'est pas exprimé en N.m ou KN.m mais en ratio par rapport au couple nominal. Cela donne :

LS 315 SN : Couple de démarrage : 2,5*0,729 = 1,82 kN·m < 2 kNm → insuffisant

LS 315 MP : Couple de démarrage : 2,4*0,877 = 2,10 kN·m > 2 kNm → OK s=1,1

LS 315 MR : Couple de démarrage : 2,4*1,072 = 2,57 kN·m > 2 kNm → OK s=1,3

Vitesse de rotation : 1000/1200 = 0,83 tr·min-1 → Critère de vitesse respecté pour les 4.

Q31. Courant de démarrage

$I_{démarrage}$ = 7,2 * 164 = 1 180 A

$I_{démarrage\ 8\ voiles}$ = 7,2 * 164 * 8 = 9 450 A

Pour éviter cet appel de courant, il est envisageable d'étager les démarrages car la variation d'orientation sera très lente. Ainsi l'impact de ces 8 moteurs sur les circuits électriques du navire sera réduit.

Q32. Temps d'exécution

$$N_m = 980\ tr \cdot min^{-1} = 5880\ ° \cdot s^{-1}$$

Rampe de vitesse en 10s pour atteindre la vitesse nominale donc : $\ddot{\theta}_m = 588\ ° \cdot s^{-2}$

Réduction de 1200 donc : $\Delta\theta_m = 3 \cdot 1200 = 3600\ °$

Aire sous la courbe d'accélération et de décélération : $\Delta\theta_m = 2 \cdot \ddot{\theta}_m \cdot \frac{{t_f}^2}{2}$

$$t_f = \sqrt{\frac{\Delta\theta_m}{\ddot{\theta}_m}} = \sqrt{\frac{3600}{588}} = 5\ s$$

Q33. Consommation des moteurs

À partir des données météorologiques, les constructeurs évaluent qu'il est nécessaire de faire varier l'angle d'incidence de 3° chaque minute or pour réaliser ce mouvement le moteur n'a pas le temps d'atteindre sa vitesse nominale. Le contrôleur VFD impose donc une loi de commande écourtée qui comporte une accélération suivie d'une décélération : démarrage et freinage.

$$P_{instantanée\ consommée} = 3\ P_{nominale} = 270kW$$

$$P_{moyenne\ consommée} = \frac{5\ P_{instantanée\ consommée}}{60} = 22{,}5kW << 4800kW$$

Consommation d'1 moteur d'orientation : 0,5%

Consommation de 8 moteurs d'orientation : 3,7%

La consommation d'un seul moteur est négligeable mais la consommation moyenne des 8 moteurs augmenterait la consommation de 3,7% ce qui n'est pas négligeable et qui devra être pris en compte dans le bilan énergétique de l'installation des ailes.

Q34. Démonstration – Loi de Faraday

Le stator crée un champ tournant à la pulsation ω_S, et le rotor tourne à la vitesse Ω. Du point de vue électrique, le rotor a une pulsation de $\Omega \cdot p = \omega$. Dans le repère électrique du rotor, par composition, le stator a donc une pulsation de $\omega_S - \omega$.

D'après la loi de Faraday :

$$\underline{E_R} = -j(\omega_S - \omega)\underline{\psi_R}$$

Q35. Vitesse de synchronisme

On sait que $\omega_S = 2\pi f_S$, d'où :

$$\Omega_S = \frac{\omega_S}{p} = \frac{2\pi f_S}{p} = \frac{2\pi \times 50}{3} = 105\ rad/s$$

$$= \frac{2\pi \times 50}{3} \times \frac{60}{2\pi} = 1000\ tr/min$$

Q36. Glissement nominal

D'après la documentation, $\Omega_n = 980\ tr/min$ au rotor dans les conditions nominales donc :

$$g_n = \frac{1000 - 980}{1000} = 2\%$$

Q37. Expression de $\underline{E_R}$

$$g = \frac{\Omega_S - \Omega_R}{\Omega_S} = \frac{\omega_S - \omega_R}{\omega_S}$$

Donc :

$$\underline{E_R} = -j(\omega_S - \omega_R)\Psi_R = -jg\omega_S\Psi_R$$

Q38. Rotor en court-circuit - expression de $\underline{E_R}$

Rotor en court-circuit donc $V_R = 0$. Ainsi, d'après la loi des mailles et la loi d'Ohm : $\underline{E_R} = -R_R\underline{I_R}$.

Q39. Rotor en court-circuit - expression de $\underline{E_R}$

$\underline{E_R} = -R_R\underline{I_R}$ d'après la Q38
$= -jg\omega_S\Psi_R$ d'après la Q37
$= -jg\omega_S\Psi_S$ car il n'y a pas de fuite de champ magnétique
$= +g\underline{E_S}$ d'après la formule de $\underline{E_S}$ donnée dans le sujet

Par identification : $\underline{E_S} = -\frac{R_R}{g}\underline{I_R}$

Le couplage étant parfait (pas de fuite de champ magnétique), on a :

$$E_S I_S = E_R I_R$$

Donc :

$$\frac{1}{g} = \frac{E_S}{E_R} = \frac{I_R}{I_S}$$

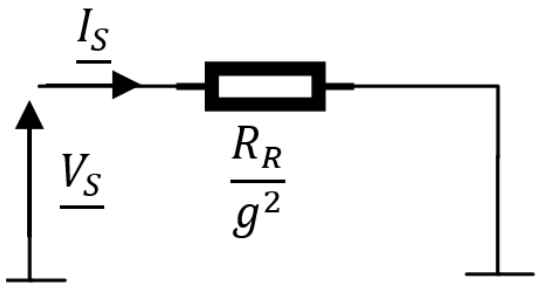
Remarque : ces deux équations (en module) peuvent aussi s'écrire en complexe. On aurait alors :

$$\underline{E_S} \cdot \underline{I_S^*} = \underline{E_R} \cdot \underline{I_R^*} \qquad \text{et} \qquad \frac{1}{g} = \frac{\underline{E_S}}{\underline{E_R}} = \frac{\underline{I_R}}{\underline{I_S}}$$

Dans tous les cas, on obtient :

$$\underline{E_S} = -\frac{R_R}{g^2}\underline{I_S}$$

donc la résistance équivalente du rotor ramenée au stator est R_R/g^2. Schéma :



Q40. Puissance active consommée

$$P_n = \frac{3V_S^2}{R_R/g_n^2} = \frac{U_S^2 g_n^2}{R_R}$$

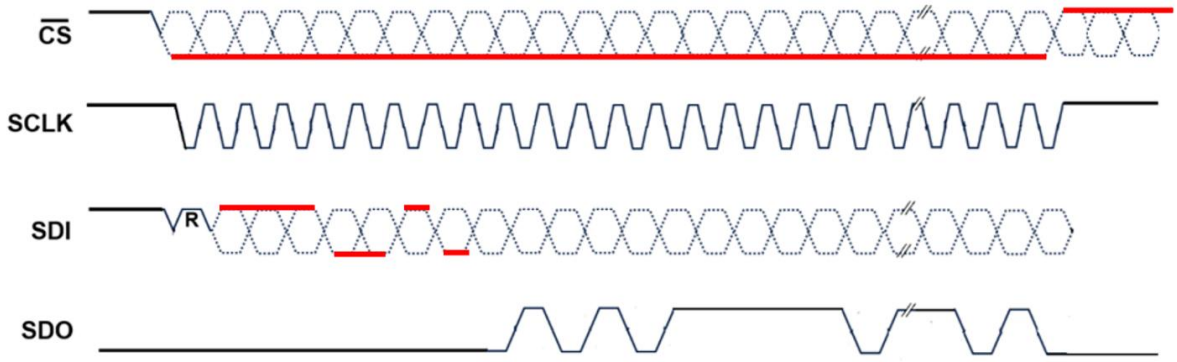
Q41. Commande U/f constant

On alimente la machine avec une tension $V_S(t)$ telle que la quantité V_S/ω_S soit constante pour que le couple soit constant et maximal. Le convertisseur permettant de générer une tension alternative $V_S(t)$ de fréquence variable est l'onduleur (ou cyclo-convertisseur).

Partie 4 : Contrôle de la position des ailes

Q42. Lecture multiple
Tant que CS n'est pas désactivé, le capteur pointera vers le registre suivant. Les registres correspondant aux mesures (X-Axis Data 0, Data 1…) étant séquentiels (0x32, 0x33…), on doit donc pointer en premier vers 0x32, soit 110010 en binaire. Le bit MB doit être à 1.

Q43. Compléter la trame



Q44. Lecture
0101 0111 en binaire, soit 87.

Q45. Compléter le code Python

```
def PO(force, d_theta):
        theta = mesure_theta()
        consigne_position(theta + d_theta)
        sleep(10) # attendre 10 secondes
        force_prime = mesure_force()
        if force_prime < force :
                d_theta = -d_theta
        return force_prime, d_theta

# initialisation
force = mesure_force()
d_theta = 10

while True:
        force, d_theta = PO(force, d_theta)
```

Q46. Limitations et pistes d'améliorations de l'algorithme P&O
Consommation : Les puissances et les inerties mises en jeu sont très grandes, et en plus les moteurs consomment beaucoup au démarrage : changer l'orientation de chaque voile toutes les 10 secondes risque de coûter trop cher en énergie. On peut se permettre d'augmenter le délai Δt, jusqu'à par exemple 1min. De toute manière, on n'utilise pas les ailes pour changer abruptement la trajectoire du bateau !
Convergence : on remarque qu'avec cet algorithme, on risque d'osciller autour de la position d'équilibre (faire $\pm 10°$ en permanence). On pourrait mettre un seuil sur $\delta\theta$ en-deçà duquel on ne modifie pas l'orientation de l'aile.
Rapidité : On pourrait aussi paramétrer le pas : au lieu d'un pas fixe, on utilise un pas dynamique, qui diminue lorsqu'on se rapproche du maximum

Q47. Gradient
On a : $\Delta F = \frac{F'-F}{\theta'-\theta}$, d'où :

```
        gradient = (force_prime – force) / (theta_prime - theta)
```

Q48. Discussion
Consommation : rien n'a changé donc non.
Convergence : si le paramètre d'apprentissage eta est trop grand, on peut toujours osciller autour de la position d'équilibre, voire diverger. En plus, l'estimation du gradient peut amener un risque de division par zéro si θ varie trop peu (notamment à l'approche du maximum). La solution serait toujours de mettre un seuil sur $\theta' - \theta$ en-deçà duquel on ne modifie pas l'orientation de l'aile. Rapidité : si le paramètre d'apprentissage est trop petit, ça va être lent.
Rapidité : s'il est bien réglé, il peut potentiellement être plus rapide que l'algo P&O, mais pas de garantie.
Conclusion : l'algo de montée de gradient n'est pas mieux que l'algo P&O, qui est plus simple, mais plus robuste.

Q49. Avantages et inconvénients de l'approche par apprentissage automatique
Avantage :

- Permet de réduire le nombre d'accéléromètres -> réduire coût (même si en réalité pas très cher, 25€), simplification matérielle

Inconvénients :

- Comportement imprévisible : Dépendance à la qualité des données d'entraînement (si nouvelles conditions météorologiques ?), manque de garanties théoriques sur les performances -> garde-fous ?
- Perte d'information : un unique accéléromètre central ne permet peut-être pas d'accéder aux forces spécifiques sur chaque voile, ce qui limite la précision
- Complexité algorithmique : ça introduit une complexité supplémentaire en termes de conception, d'entraînement et de mise à jour du modèle

Panne : en cas de défaillance du capteur unique, le système perd complètement sa capacité de mesure, alors que ce n'est pas le cas avec plusieurs capteurs répartis

Q50. Fonction de transfert

$$H(p) = \frac{\Omega(p)}{C_{mot}(p) - C_r(p)} = \frac{1}{f + J \cdot p} = \frac{\frac{1}{f}}{1 + \frac{J}{f} \cdot p} = \frac{K}{1 + T \cdot p}$$

Avec $K = 1/f$ et $T = J/f$.

Q51. Protocole expérimental

- On mesure pour chaque essai la valeur du couple moteur $c_{mot\infty}$ et de la vitesse de rotation ω_∞ en régime permanent.
- On trace le nuage de points $(\omega_\infty, c_{mot\infty})$ pour tous les essais.
  On a alors : $c_{mot\infty} - c_r - f.\omega_\infty = 0$, donc :
  $$c_{mot\infty} = c_r + f.\omega_\infty$$
- On fait une régression affine. Le coefficient directeur sera f et l'ordonnée à l'origine sera le couple résistant c_r en l'absence de vent.

Q52. Valeurs numériques des paramètres
En régime permanent, $\omega_\infty = K \cdot (c_0 - c_r)$ donc : $K = \frac{\omega_\infty}{c_0 - c_r} = \frac{997 \times 2\pi/60}{100-10} = 1{,}2\ rad.s^{-1}.N^{-1}.m^{-1}$. On détermine la valeur de la constante de temps en utilisant le fait que : soit après T on atteint 63% de la valeur finale, soit après $3T$ on atteint 95% de la valeur finale, soit la pente à l'origine coupe l'asymptote en T. On devrait trouver : $T \simeq 3{,}4\ s$

Q53. Choix du correcteur
Un correcteur PI permet d'augmenter la classe de la FTBO et de le rendre parfaitement précis. Un correcteur P, même s'il était plus rapide, ne permettrait d'améliorer la précision qu'au prix d'une grande valeur de gain car les grandeurs mises en jeu sont grandes.

Q54. Pulsation de coupure

$$|H(\omega_{0dB})| = 1 \Leftrightarrow \frac{K}{\sqrt{1 + (T \cdot \omega_{0dB})^2}} = 1 \Leftrightarrow \omega_{0dB} = \frac{1}{T}\sqrt{K^2 - 1} = 0{,}19\ rad.s^{-1}$$

On veut que $\omega_{0dB,corr} = 2 \cdot \omega_{0dB} = 0{,}39\ rad/s$

Q55. Réglage du correcteur

On place $1/T_i$ une décade avant la pulsation de coupure à 0 dB souhaitée, donc :

$$\frac{1}{T_i} = \frac{1}{10}\omega_{0dB,corr} \quad \Rightarrow \quad T_i = \frac{10}{\omega_{0dB,corr}} = 26\ s$$

Comme $1/T_i$ est placé bien avant ω_{0dB}, on a : $C(p) \simeq K_p$, donc :

$$\left|H(\omega_{0dB,corr}).C(\omega_{0dB,corr})\right| = 1 \Leftrightarrow \frac{K_p K}{\sqrt{1+\left(T\cdot\omega_{0dB,corr}\right)^2}} = 1 \Leftrightarrow K_p = \frac{\sqrt{1+\left(T\cdot\omega_{0dB,corr}\right)^2}}{K} = 1{,}4\ USI$$

Q56. Marges de stabilité

Graphiquement : La phase de la FTBO corrigée n'atteint jamais -180° donc marge de gain infini. Pour la marge de phase, on trace la flèche qui va de -180° jusqu'à la courbe en phase à la pulsation où le gain vaut 0dB (entre $0{,}6$ et $0{,}7\ rad/s$) : la flèche doit aller vers le haut ! Valeur : $M_\varphi \simeq 100°$. Le calcul analytique des marges de stabilité était aussi acceptable (bien que, inutilement plus long).

Q57. Synthèse

Stabilité : très stable (d'après Q56) car marge de gain infini et marge de phase très supérieure à 45° (valeur habituelle). Précision : parfaitement précis en réponse à un échelon de vitesse (d'après Q53). Rapidité : temps de réponse deux fois plus rapide car pulsation de coupure à 0 dB deux fois plus grande (d'après Q54).

Partie 5 : Déploiement et inversion des ailes

Q58. Fermeture géométrique

Dans le polygone $OAHD$:

$$\overrightarrow{OA} + \overrightarrow{AH} + \overrightarrow{HD} + \overrightarrow{DO} = \vec{0}$$

$$\lambda(t)\overrightarrow{z_2} + \frac{l_6}{2}\overrightarrow{y_6} - h_6\overrightarrow{z_6} - l\overrightarrow{y_0} - L\overrightarrow{z_0} = \vec{0}$$

En projection dans la base B_6 :

$$\begin{cases} /\overrightarrow{y_6}: & -\lambda(t)\sin\alpha_2 + \dfrac{l_6}{2} - l\cos\gamma - L\sin\gamma = 0 \\ /\overrightarrow{z_6}: & \lambda(t)\cos\alpha_2 - h_6 + l\sin\gamma - L\cos\gamma = 0 \end{cases}$$

D'où : $\lambda(t) = \sqrt{\left(\frac{l_6}{2} + l\sin\frac{\beta}{2} - L\cos\frac{\beta}{2}\right)^2 + \left(h_6 - l\cos\frac{\beta}{2} - L\sin\frac{\beta}{2}\right)^2}$

Q59. Course des vérins

$C = 2600$ mm pour 2750 mm sur la documentation technique. La course du vérin choisi est suffisante.

Q60. Relation affine

$a = -875$mm.rad^{-1} et $b = 612$mm

Q61. Matrice d'inertie

La matrice $\overline{\overline{I}}_{G_1}(1)$ présente la forme de celle d'un solide de plan de symétrie $(G_1, \overrightarrow{x_1}, \overrightarrow{y_1})$.

Q62. Energie cinétique

Les éléments d'inertie des solides 2 à 6 sont négligeables devant ceux du solide 1 : $E_{c,\Sigma/0} = E_{c,1/0}$

La matrice d'inertie est donnée en G_1, d'où : $E_{c,1/0} = \frac{1}{2}\left[m_1 \cdot \left\|\overrightarrow{V_{G_1,1/0}}\right\|^2 + \overrightarrow{\Omega_{1/0}} \cdot \overline{\overline{I}}_{G_1}(1) \cdot \overrightarrow{\Omega_{1/0}}\right]$

$E_{c,1/0} = \frac{1}{2}\left[m_1 \cdot \left({e_1}^2 + {h_1}^2\right) + A_1\right]\dot{\beta}^2$ avec $J_{eq} = \left[m_1 \cdot \left({e_1}^2 + {h_1}^2\right) + A_1\right] = 1{,}29 \times 10^6$ kg.m²

Q63. Puissances galiléennes

$P_{v_1 \to 6/0} = \overrightarrow{R_{\substack{v \\ 0\to 6}}} \cdot \overrightarrow{V_{A,6/0}} = -pS\dot{\lambda}(t) = -apS\dot{\beta}(t)$

$P_{\text{pes}\to 1/0} = \overrightarrow{R_{\text{pes}\to 1}} \cdot \overrightarrow{V_{G_1,1/0}} = -m_1 g\overrightarrow{z_0} \cdot (-e_1\overrightarrow{z_1} - h_1\overrightarrow{y_1})\dot{\beta} = m_1 g\dot{\beta}(e_1\cos\beta + h_1\sin\beta)$

$P_{\text{frott}\to 1/0} = \begin{Bmatrix} X_{01}\overrightarrow{x_0} + y_{01}\overrightarrow{y_0} + Z_{01}\overrightarrow{z_0} \\ -\mu\dot{\beta}\overrightarrow{x_0} + y_{01}\overrightarrow{y_0} + Z_{01}\overrightarrow{z_0} \end{Bmatrix}_D \otimes \begin{Bmatrix} \dot{\beta}\overrightarrow{x_0} \\ \vec{0} \end{Bmatrix}_D = -\mu\dot{\beta}^2$

$P_{\substack{v \\ 6\leftrightarrow 1}} = \overrightarrow{R_{1\to v_2}} \cdot \overrightarrow{V_{B,6/1}} = -pS\dot{\lambda}(t) = -apS\dot{\beta}(t)$

$P_{\text{tot}} = -apS\dot{\beta}(t) + m_1 g\dot{\beta}(e_1 \cos\beta + h_1 \sin\beta) - \mu\dot{\beta}^2 - apS\dot{\beta}(t)$

Q64. Théorème de l'énergie cinétique

$\frac{dE_{c,\Sigma/0}}{dt} = P_{\bar{\Sigma}\to\Sigma/0} + P_{int}$

$J_{eq}\dot{\beta}\ddot{\beta} = -2apS\dot{\beta} + m_1 g\dot{\beta}(e_1 \cos\beta + h_1 \sin\beta) - \mu\dot{\beta}^2$

$p = -\frac{1}{2aS}\left[J_{eq}\ddot{\beta} + \mu\dot{\beta} - m_1 g(e_1 \cos\beta + h_1 \sin\beta)\right]$

Q65. Loi en trapèze

Une loi en trapèze permet de limiter les effets dynamiques. Spécifiquement en hydraulique, cela permet de préserver l'intégrité du réseau.

Dans le trapèze, on écrit :

$$\dot{\beta}_{max}(T - T_a) = 180$$
$$\dot{\beta}_{max} = \ddot{\beta}T_a$$

On obtient : $T_a = 14$ s, d'où $\dot{\beta}_{max} = 0{,}7°.s^{-1}$.

Q66. Prédimensionnement vérin

Le vérin est utilisé en rentrée de tige. La surface utile est donc de $S = \frac{\pi}{4}(D^2 - d^2)$ avec $D = 230$mm et $d = 110$mm.

$S = \frac{\pi}{4}(0{,}23^2 - 0{,}11^2) = 0{,}032$m².

Q67. Dimensionnement vérin en phase d'accélération

En phase d'accélération : $\beta(T_a) = \beta(0) + \Delta\beta$

Avec $\Delta\beta = \frac{\ddot{\beta}}{2}T_a^2 = 5°$, d'où : $\beta(T_a) = -175°$

$p = -\frac{1}{2aS}\left[J_{eq}\ddot{\beta} - m_1 g(e_1 \cos\beta + h_1 \sin\beta)\right]$

$p_{\max_acc} = -\frac{1}{2aS}\left[J_{eq}\ddot{\beta} - m_1 g(-e_1)\right]$

$p_{\max\limits_{acc}} = -\frac{1}{2\times(-875\times10^{-3})\times0{,}032}\left[1{,}29\times10^6 \times 0{,}05 \times \frac{2\pi}{360} - 16{,}4\times10^3 \times 9{,}81 \times (-1{,}125)\right]$

$p_{\max\limits_{acc}} = 3\ 252\ 000$Pa = 32,5 bars

Q68. Dimensionnement vérin au bras de levier maximal

$p_{\max_cst} = -\frac{1}{2aS}[m_1 g h_1]$

$p_{\max_cst} = -\frac{1}{2\times(-875\times10^{-3})\times0{,}032}[16{,}4\times10^3 \times 9{,}81 \times 8{,}1]$

$p_{\max_cst} = 23\ 270\ 000$ Pa

$p_{\max_cst} = 233$ bars

Q 69. Conclusion dimensionnement

Le vérin répond aux exigences de course et de pression.

Q 70. Conclusion

$E = P * \Delta t$

AN : $8 \times 5 \times 270/3600 \times 20 = 60$ kWh.

Q 71. Conclusion finale

Les consommations du système pilotant les ailes représentent 8% de la production. Ces consommations ne sont pas négligeables - donc à prendre en compte dans le bilan énergétique - mais sont limitées.

Il faut rajouter à cette étude une analyse de l'ensemble de cycle de vie du produit : extraction des matériaux, fabrication, transport, installation, entretien et recyclage. Il faudra que la durée de vie de ce système soit grande pour diminuer significativement la consommation énergétique globale et l'empreinte écologique de ces moyens de transport.

Si la durée de vie du système est suffisante, le système pourra donc être performant et pertinent d'un point de vue développement durable :

- écologique : réduction des émissions de CO2 et possibilité d'une conversion éventuelle vers une autre source d'énergie bas carbone ; système de propulsion hydraulique moins puissant donc perturbant moins les écosystèmes marins.

- économique : réduction de la consommation de fioul suffisante à terme pour compenser le coût de l'installation ;
- social : création d'emplois locaux pour l'installation des ailes et du système de commande sur des bateaux existants. Création d'emplois moins locaux pour la fabrication des ailes et de leur système de commande.

D. Commentaires du jury

Les candidats ayant réussi cette épreuve sont ceux qui ont fait preuve de transversalité et qui ont fait l'effort d'aborder chacune des parties. Chaque partie était conçue avec une difficulté croissante des questions, permettant aux candidats des différentes spécialités à la fois de pouvoir aborder partiellement chaque problématique, mais également de s'affirmer dans son domaine de prédilection.
Les candidats ont majoritairement traité les différentes parties dans l'ordre proposé par le sujet.

La première partie a été abordée par l'ensemble des candidats. Le jury a constaté que les expressions littérales ont été majoritairement correctement établies, il regrette cependant de nombreuses erreurs d'application numérique essentiellement liées à une mauvaise lecture du sujet.

La seconde partie portant sur le prédimensionnement du mât a été traitée par un grand nombre de candidats. Trop de candidats ont donné les valeurs maximales du moment fléchissant ou de l'effort tranchant mais rares sont ceux qui ont réalisé correctement les coupures et donné l'équation en fonction de « z » de ces éléments. Le jury regrette que peu de candidats aient traité la dernière question, qui était une question de lecture de résultats sur une modélisation.

La troisième partie concernait le dimensionnement, la consommation et la commande des moteurs de pivotement. Le dimensionnement moteur était abordable par l'ensemble des candidats mais un grand nombre de candidats a oublié de vérifier l'exigence de vitesse (et éventuellement de puissance) en sus du couple nécessaire. De trop nombreux chiffres significatifs dans les résultats numériques ont été pénalisés par le jury car ils montrent un manque de recul sur les incertitudes mais aussi sur la portée et les limites des modèles. La modélisation de la commande du moteur asynchrone est maitrisée par les candidats qui ont travaillé ce sujet lors de leur préparation.
La quatrième partie portant sur le contrôle de la position des ailes a été traitée par la majorité des candidats. Cependant, le jury fait état de nombreuses erreurs, notamment sur le protocole SPI dont la documentation est entièrement donnée dans le sujet et sur la mise à jour des paramètres dans la boucle infinie du code Python. Les questions portant sur l'asservissement ont été assez peu abordées.

La cinquième partie portant sur le déploiement des ailes a été traitée par peu de candidats. Les premières questions mêlaient géométrie et lecture de courbes. La fermeture géométrique et notamment les projections dans une base doivent être traitées de manière beaucoup plus rigoureuse.
Les calculs d'énergie cinétique (solide en rotation autour d'un axe fixe) et de puissances ont été très mal traités par la majorité des candidats. Dans la très grande majorité des cas, le théorème de l'énergie cinétique n'est pas su ou pas écrit. Les dernières questions nécessitaient d'avoir obtenu les résultats aux questions précédentes.

Dans l'ensemble, le jury enjoint les candidats à faire preuve de plus de rigueur dans la manipulation des expressions et des grandeurs. Par ailleurs, certains candidats ne remarquent pas les erreurs flagrantes d'ordres de grandeur (par exemple dans le calcul de la surface du piston ou de la pression dans celui-ci). Il est fondamental pour un futur enseignent de sciences de l'ingénieur de maitriser ceux-ci.

La conclusion finale pouvait être traitée par l'ensemble des candidats car les résultats nécessaires étaient donnés. Trop peu de candidats prennent le temps de répondre à cette question alors qu'il parait incontournable de conclure pour répondre à la problématique générale proposée. Quelques candidats ont réussi à nuancer l'impact positif de l'installation des ailes en mentionnant la nécessité de réaliser une étude de l'ensemble du cycle de vie du système complet.

Le jury constate que les meilleurs résultats sont obtenus par les candidats ayant abordé au moins trois parties. Le jury encourage ainsi fortement les candidats à traiter toutes les parties du sujet et à montrer qu'ils maîtrisent l'ensemble des domaines des sciences industrielles de l'ingénieur. Les résultats démontrent que ceux qui sélectionnent les questions et se restreignent à celles relatives à une spécialité donnée de l'ingénierie ne réussissent pas cette épreuve, car la note finale se trouve alors fortement

limitée. Par conséquent, le jury conseille aux futurs candidats de s'investir sérieusement dans toutes les parties du programme du concours et d'acquérir l'ensemble des compétences et des connaissances associées aux disciplines qui constituent les sciences industrielles de l'ingénieur. Les candidats doivent également s'attacher à utiliser leurs connaissances universitaires afin de résoudre des problématiques techniques associées à des systèmes industriels. Les plus efficients ont su ne pas perdre de vue que les analyses, les justifications et les choix technologiques doivent être toujours menés en gardant à l'esprit les enjeux du contexte industriel spécifique à l'étude.

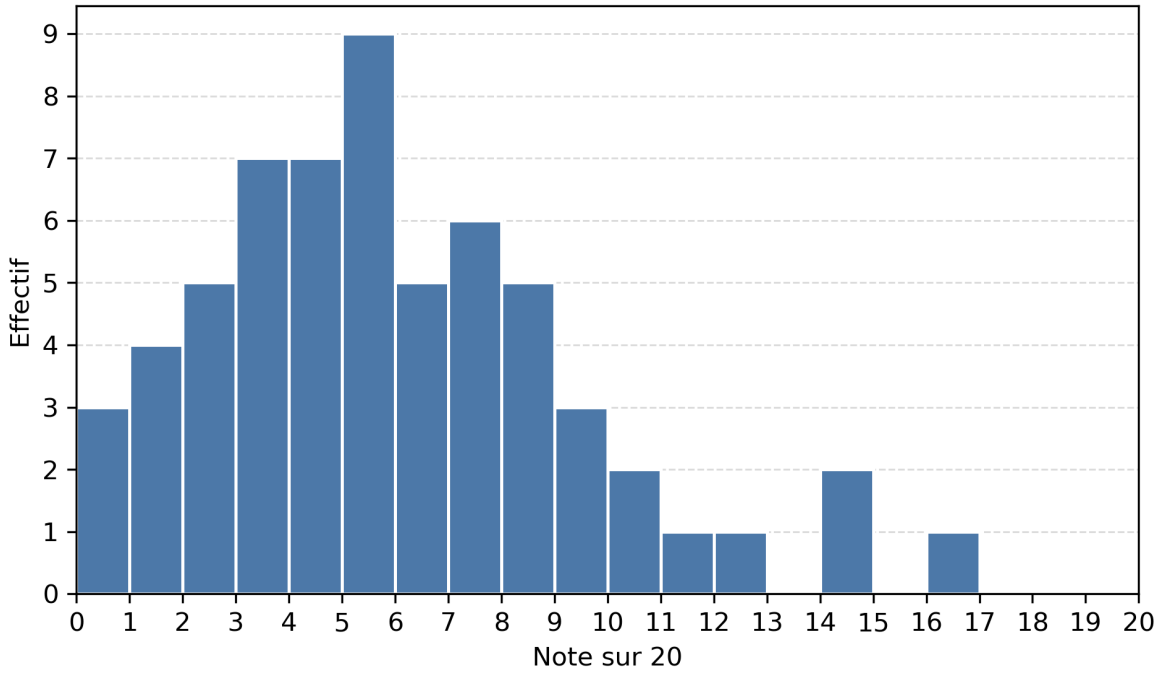
Le jury constate trop souvent un manque de rigueur, notamment dans l'écriture des expressions littérales, dans la manipulation des grandeurs scalaires et vectorielles, de précision dans la présentation des copies et dans la rédaction. La présentation de la copie doit être irréprochable, les notations imposées dans le sujet doivent être scrupuleusement respectées. Trop de copies ressemblent à un brouillon tant au niveau de l'organisation que de l'écriture peu soignée. On rappelle que la rédaction des réponses doit toujours commencer par indiquer le numéro exact de la question traitée. Il convient aussi de rappeler qu'il est attendu d'un fonctionnaire de l'État qu'il maîtrise convenablement la langue française et qu'il respecte les règles de l'orthographe et de la grammaire française afin de s'assurer que ce qu'il souhaite exprimer soit compréhensible et lisible.
Les réponses doivent être détaillées et argumentées : des résultats donnés directement, sans calcul, sans justification de principe, ne peuvent pas être considérés comme justes. Par ailleurs, les réponses montrant une maîtrise de la démarche mais n'arrivant pas jusqu'à la conclusion sont valorisées. Le jury apprécie aussi l'esprit critique face à des résultats aberrants et admet le choix délibéré de commenter ces résultats pour continuer le traitement du sujet.
Réussir cette épreuve demande :

- de s'approprier en un temps limité un sujet technique pluridisciplinaire décrit avec les outils de modélisation de l'ingénierie système ;
- de maîtriser les modèles de connaissance des différents domaines d'étude de l'ingénierie ;
- d'analyser et d'interpréter des résultats d'étude, afin de formuler des conclusions cohérentes et pertinentes en concordance avec une problématique scientifique et technique.

Le sujet a été conçu pour permettre aux candidats d'exprimer au mieux leurs compétences dans différents champs d'application d'un système pluritechnologique correspondant au cadre de cette épreuve transversale. Le jury engage fortement les futurs candidats à se préparer conformément aux attendus de l'arrêté du 28 décembre 2009 modifié.

E. Résultats

L'histogramme pour cette épreuve est donné ci-dessous.



Commentaires du jury

Le sujet porte sur l'étude en conception préliminaire d'un robot agricole de l'INRAE dans une synergie industrielle. L'INRAE (Institut National de Recherche sur l'Agriculture et l'Environnement) est un institut de recherche qui apporte son expertise en robotique agricole, notamment à travers son unité TSCF, développant des robots adaptables aux milieux naturels. SABI AGRI, pionnier des agroéquipements électriques, conçoit et fabrique ses tracteurs en France.

Une synergie entre ces deux acteurs vise à accélérer l'innovation dans la robotique agricole pour répondre aux enjeux environnementaux et économiques. Ce contexte de collaboration industrielle et de recherche sur plusieurs modèles de robot apporte des contraintes d'industrialisation, de réutilisation du code, d'analyse générique, de gestion de l'autonomie et d'accès distant aux données, …

Le sujet est découpé en quatre parties dont le questionnement est indépendant :

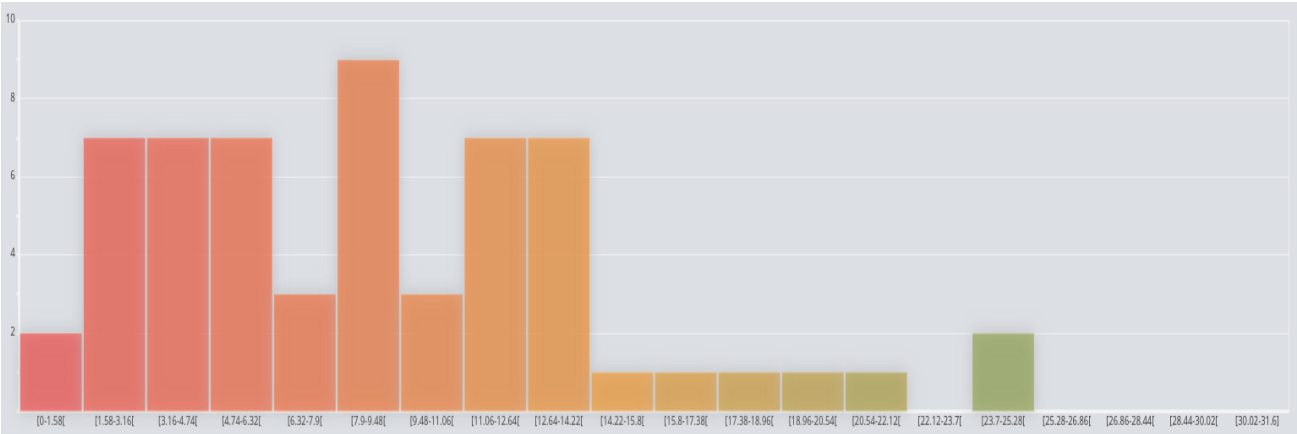
1- Algorithmes de pilotage (dans ROS2) (23 questions)

- Code de publication et de souscription aux topics ROS2
- Equations cinématiques
- Code pour piloter le robot ALPO
- Odométrie
- Code commande des moteurs
- Pilotage des roues directrices
- BUS CAN

Le jury constate que les parties calculatoires sont souvent traitées de manière approximative, et souvent pas traitées du tout. Le jury regrette que les démonstrations sont souvent obscures.

Le langage C est souvent mal connu (code C souvent écrit avec une syntaxe de code Python) : le jury rappelle que la maîtrise des langages de programmation est une composante du concours. Les calculs des probabilités sont très peu traités.

Cette première partie montre déjà une certaine hétérogénéité de niveau des candidats.



Répartition des résultats sur la partie 1 : Algorithmes de pilotage (dans ROS2)

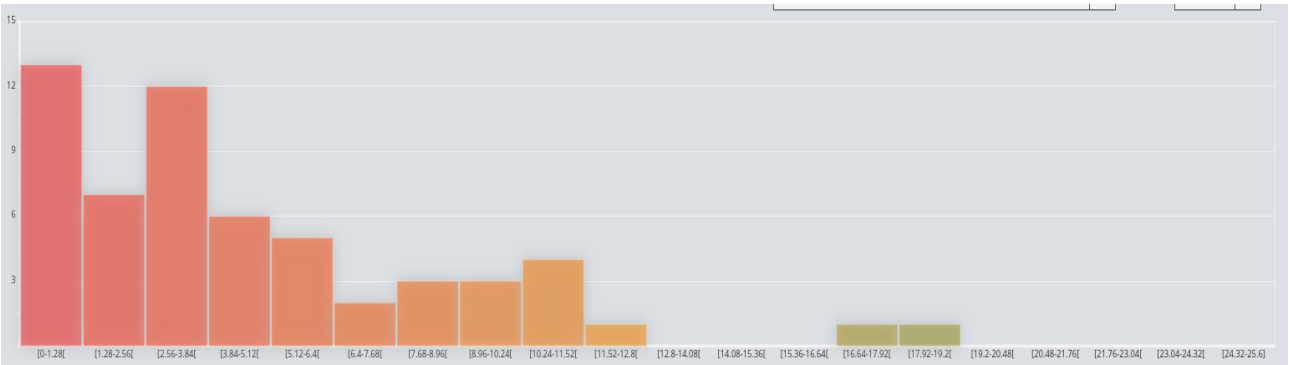
2- Mise en œuvre de différents capteurs (22 questions)

- Capteur IMU
- Camera stéréo

- GNSS RTK

Le jury constate que cette partie est traitée avec un certain essoufflement tout au long des réponses. Les résultats sont globalement très faibles. Deux candidats n'ont même pas du tout abordé cette partie.

Le jury regrette que les calculs matriciels soient globalement non maitrisés par les candidats.



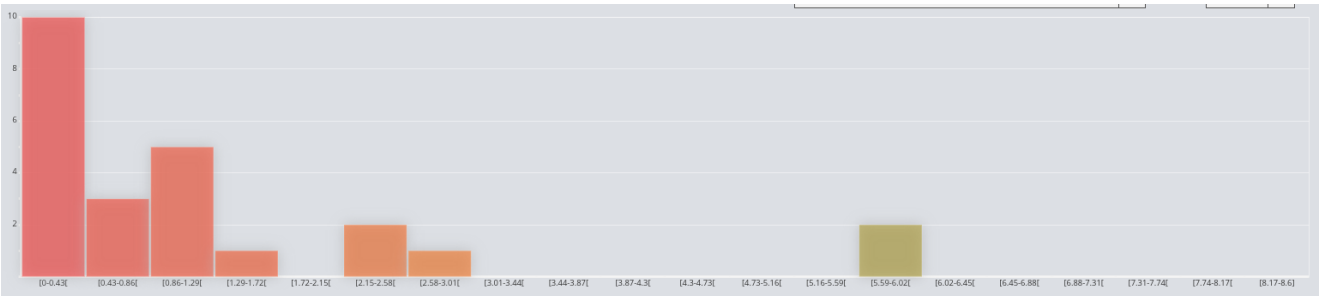
Répartition des résultats sur la partie 2 : Mise en œuvre de différents capteurs

3- Fusion des données capteurs (9 questions)

- Fusion des capteurs
- Localisation du robot
- Filtre de Kalman étendu

Cette partie n'a été abordée que par 25 candidats. Deux questions de cette partie n'ont été traitées que par deux candidats.

Le jury constate que les notions de variable probabiliste ne sont généralement pas maitrisées



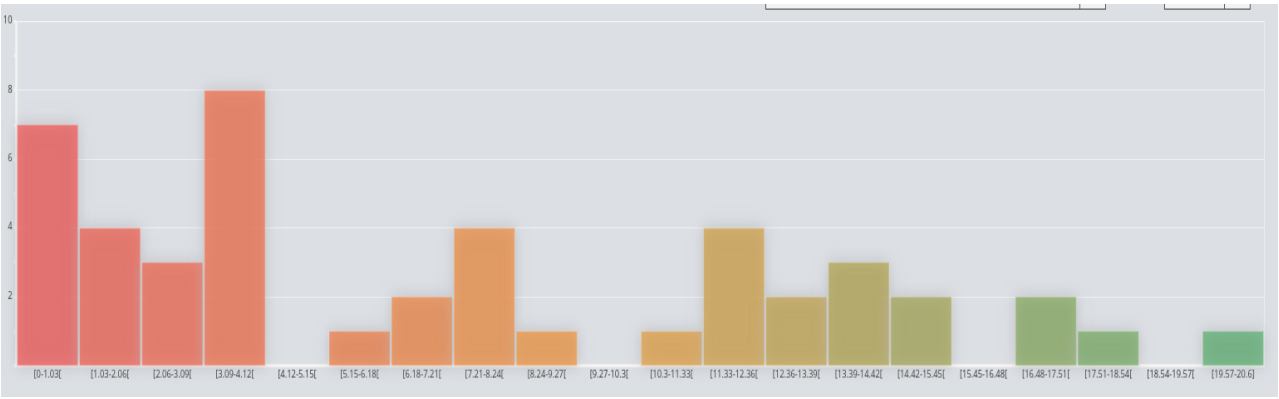
Répartition des résultats sur la partie 3- Fusion des données capteurs

4- Stockage des données et sécurité (10 questions)

- Élaboration d'une modélisation conceptuelle
- Analyse fonctionnelle, sécuritaire et réglementaire
- (RGPD) sur une modélisation donnée

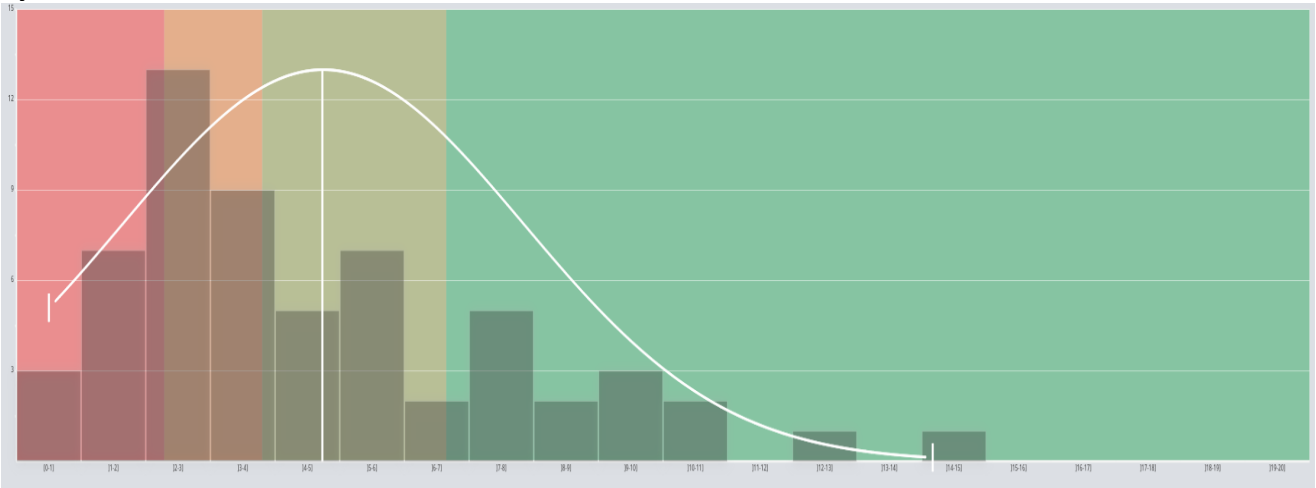
Le jury constate que cette partie n'a été abordée que par 46 candidats. Il regrette que les notions fondamentales de bases de données, et en particulier la notion de clés étrangères, soient souvent méconnues. De même il est constaté que la syntaxe SQL est souvent très approximative et mal maîtrisée.

Avec un taux de réponse aux questions qui demeure assez moyen la dispersion des candidats est très importante.



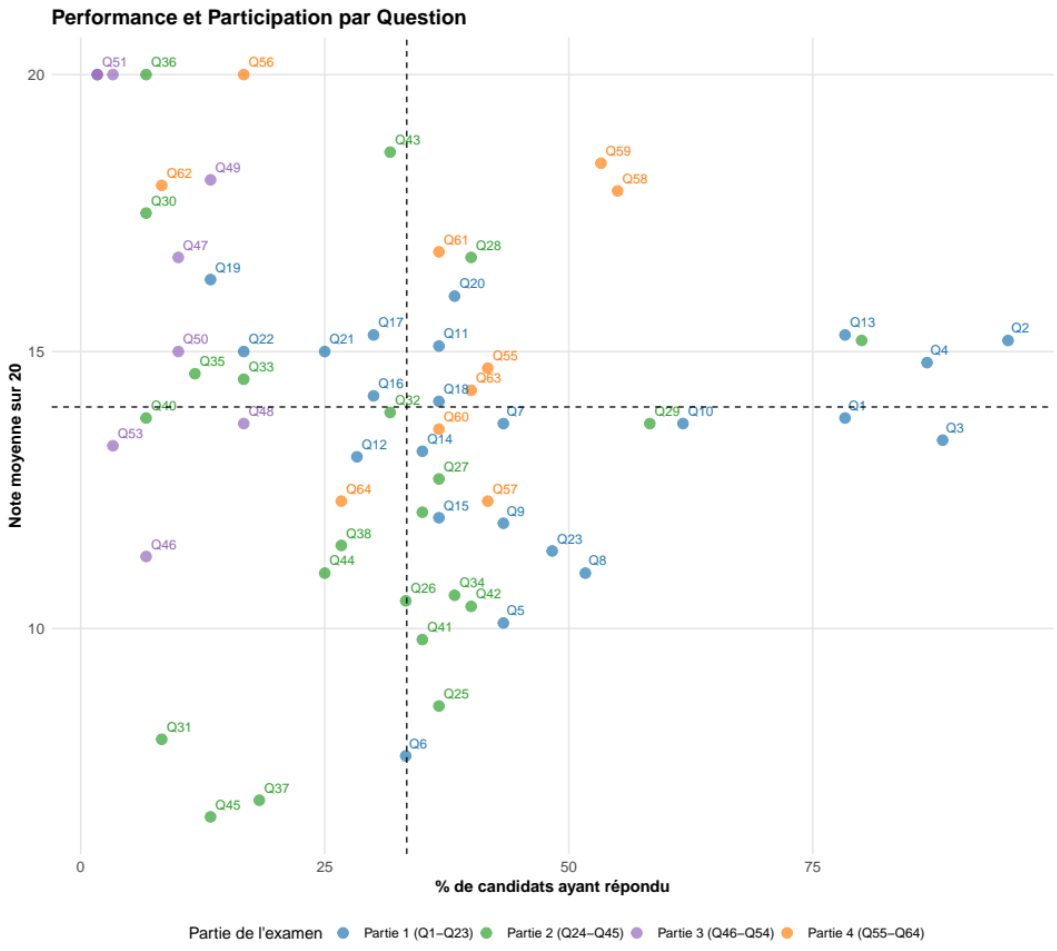
Répartition des résultats sur la partie 4- Stockage des données et sécurité

Synhèse :

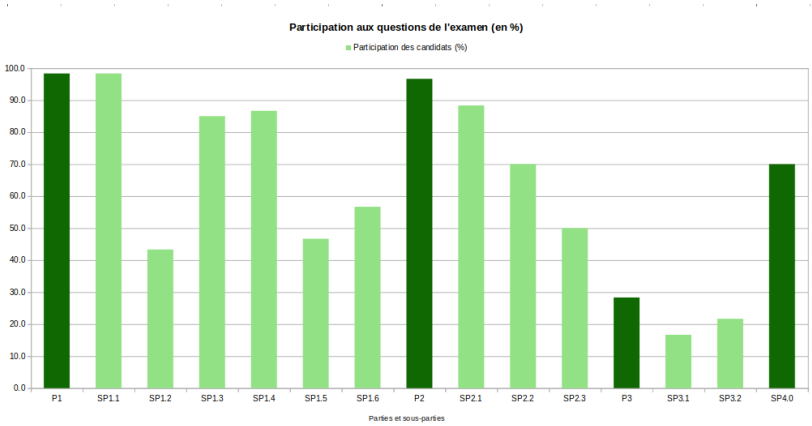


Répartition des résultats sur la totalité de l'épreuve

La moyenne se positionne à 4,6/20 avec des notes se dispersant entre 0,7/20 et 14,19/20



Relation entre performance et participation par question



Taux de réponse aux diférentes parties

Ce sujet de conception couvre, de manière soutenue, un champ assez large du programme de l'agrégation qui est mis en œuvre dans plusieurs programmes du supérieur.

Le jury constate que le taux de non-réponse aux questions est assez élevé et que la qualité des réponses est souvent faible ou obscure, en particulier dans les démonstrations, au regard des attendus pédagogiques à venir.

Avec une moyenne de 4,6/20, les résultats sont souvent très décevants, tant au niveau de la prise en main globale du sujet que de la maîtrise des langages de programmation. Les meilleures copies sont assez sélectives et ne couvrent généralement pas toutes les parties.

Robot agricole autonome de l'INRAE

Présentation du système



Figure 1 : tracteur électrique ALPO de SABI-AGRI dans le contexte de la ferme expérimentale de Montoldre (Allier)[1]

Présentation du contexte

L'INRAE (Institut National de Recherche sur l'Agriculture et l'Environnement) est un institut de recherche qui apporte son expertise en robotique agricole, notamment à travers son unité TSCF, développant des robots adaptables aux milieux naturels. SABI AGRI, pionnier des agroéquipements électriques, conçoit et fabrique ses tracteurs en France. Une synergie entre ces deux acteurs vise à accélérer l'innovation dans la robotique agricole pour répondre aux enjeux environnementaux et économiques.

Au sein de l'unité de recherche TSCF (Technologies et systèmes d'information pour les agrosystèmes - Clermont-Ferrand) de l'INRAE, l'équipe ROMEA, basée à Aubière, conçoit des systèmes reconfigurables et à autonomie partagée, pour accroître les performances et la sécurité des engins œuvrant en milieux naturels, en particulier ceux rencontrés dans l'agriculture.

La ferme expérimentale de Montoldre dans l'Allier permet à l'équipe ROMEA de tester les algorithmes en situation réelle. La ferme est, entre autres, un vivarium de robots mobile manipulateurs pour la recherche dans le domaine de la robotique en milieux tout-terrain et agricoles.

L'équipe ROMEA de l'unité de recherche TSCF travaille avec plusieurs modèles de robots regroupant les principales architectures utilisées dans le domaine de la robotique agricole.

[1] Pierre, Cyrille & Lenain, Roland & Jean, Laneurit & Rousseau, Vincent. (2022). A Multi-Control Strategy to Achieve Autonomous Field Operation. AgriEngineering. 4. 770-788. 10.3390/agriengineering4030050.

Robots de l'équipe ROMEA

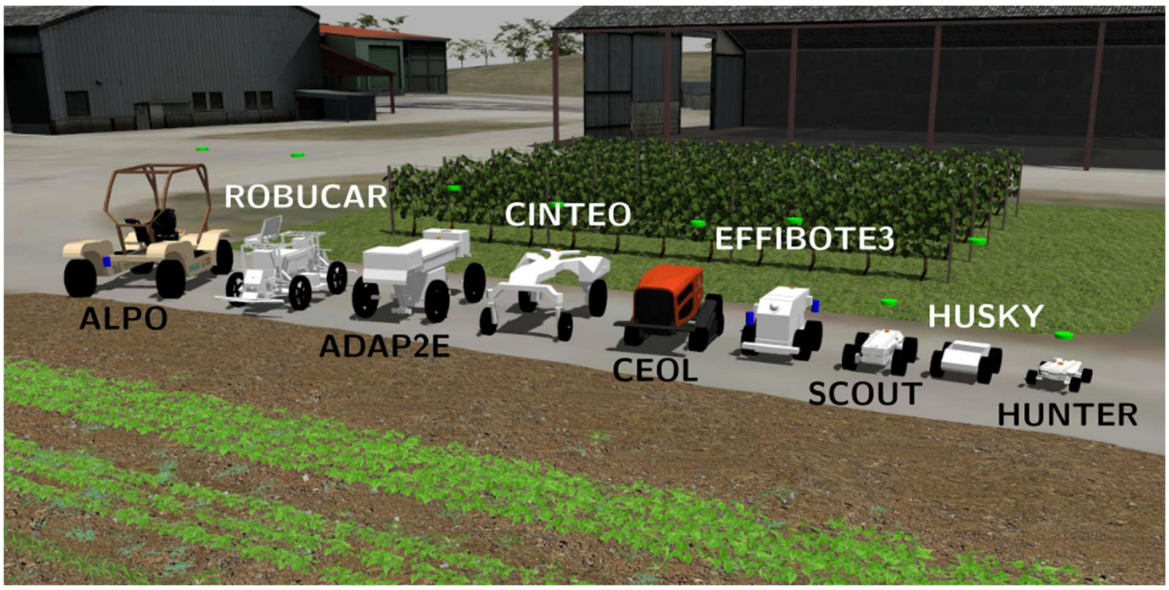


Figure 2 : doubles numériques des robots de l'unité de recherche TSCF

Les robots disponibles pour l'unité de recherche TSCF recouvrent les catégories de robots suivantes :

- véhicules à direction par dérapage (***Skid steering vehicles***) :
  - **Effibote3, Scout, Husky** : 4 roues motrices (4 wheel drive vehicle noté **4WD**) ;
  - **Ceol** : 2 chenilles (2 track drive vehicle : **2TD**) ;
- véhicules à un essieu directeur (***One axle steering vehicles***) :
  - **Hunter, Cinteo** : 1 roue directrice avant + 2 roues motrices arrières (1 front steering axle + 2 rear wheel drive vehicle : **1FAS2RWD**) ;
- véhicules à deux essieux directeurs (***Two axle steering vehicles***) :
  - **Robucar** : 2 steering axles + 4 rear wheel drive vehicle (**2AS4RWD**) ;
- véhicules à deux roues directrices ***(Two wheel steering vehicles***) :
  - **Alpo Slim** : 2 front wheel steering + 4 wheel drive vehicle (**2FWS4WD**) ;
  - **Alpo Fat** : 2 front wheel steering + 2 rear wheel drive vehicle (**2FWS2RWD**) ;
- véhicules à quatre roues directrices (***Four wheel steering vehicles***) :
  - **Adap2e** : 4 wheel steering + 4 wheel drive vehicle (**4WS4WD**).

ROS (*Robot Operating system*)

Ces robots utilisent ROS2 (Robot Operating System 2) qui est une plateforme *open-source* conçue pour le développement de logiciels pour robots. Elle facilite la création, le déploiement et la gestion d'applications robotiques complexes.

L'architecture de ROS2 repose sur plusieurs concepts clés :

- nœuds (*nodes*) : un nœud exécute un processus qui effectue une tâche spécifique, comme le contrôle des moteurs ou le traitement des données des capteurs ;
- *topics* : Les *topics* sont des canaux de communication asynchrones utilisés par les nœuds pour échanger des messages. Un nœud peut publier des messages sur un *topic*, et d'autres nœuds peuvent s'abonner à ce *topic* pour recevoir les messages. Les *topics* permettent un découplage temporel et spatial entre les nœuds. Les nœuds n'ont pas besoin de connaître l'existence les uns

des autres pour communiquer. Ils se contentent de publier ou de s'abonner à des *topics* spécifiques ;

- services : les services permettent une communication bidirectionnelle synchrone entre les nœuds. Un nœud peut offrir un service, et un autre nœud client peut appeler ce service pour obtenir une réponse ;
- actions : les actions permettent une communication asynchrone avec des mises à jour de progression (*feedback*). Le nœud client peut recevoir des mises à jour régulières sur l'état de la tâche en cours ;
- messages : les messages sont des structures de données utilisées pour transmettre des informations entre les nœuds. Ils sont définis à l'aide de fichiers de description de message ;
- paramètres : un paramètre est une valeur configurable qui peut être définie, modifiée et lue par un nœud (*node*) pendant son exécution. Les paramètres permettent de personnaliser le comportement d'un nœud sans avoir à recompiler le code.

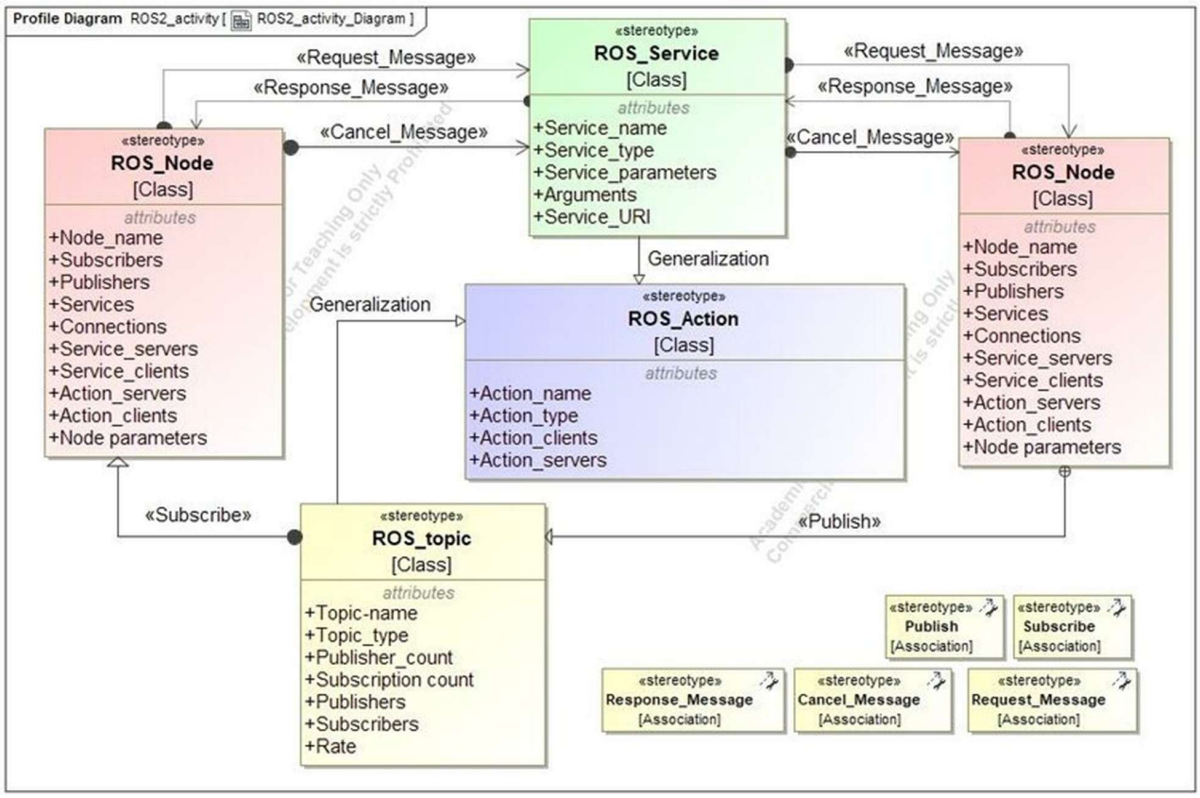


Figure 3 : diagramme d'activité ROS2[2]

Un système robotique complet est composé de nombreux nœuds fonctionnant de concert. Dans ROS2, un seul exécutable (programme C++, programme Python, etc.) peut contenir un ou plusieurs nœuds. Chaque nœud peut envoyer (en publiant) et recevoir (en souscrivant) des données vers ou depuis d'autres nœuds via des *topics*, des services ou des actions.

Une représentation souvent utilisée dans les projets ROS est obtenue par l'outil RQt (*ROS Qt GUI Tool*). À partir du code, cet outil génère un diagramme permettant une visualisation rapide de l'organisation d'un projet (*ROS Graph*). Un tel diagramme montre comment les nœuds (entourés par

[2] Guizani, Amir & Aloui, Khalil & Hammadi, Moncef & Soriano, Thierry & Haddar, Mohamed. (2023). A new SysML profile for autonomous mobile robots development: ROS2ML. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. 237. 095440622211493. 10.1177/09544062221149314.

des ellipses) communiquent les uns avec les autres par le biais des *topics*, services et actions (entourés par des rectangles). Les flèches montrent la circulation de l'information.

Le graphe ROS représentant la communication de base sur le robot ALPO est présenté sur le document **DT1**.

Dans ce graphe, le nœud ***/robot/base/mobile_base_controller*** publie sur les *topics* :

- ***/robot/base/controller/kinematic*** ;
- ***/robot/base/controller/odom*** ;
- ***/robot/base/controller/odometry***.

Il a souscrit au *topic* ***/robot/base/controller/cmd_one_axle_steering***.

Système de l'étude

L'objet de l'étude ci-après se concentre sur le robot ALPO de la société SABI AGRI.

Le robot ALPO, développé par SABI AGRI en collaboration avec INRAE, est un tracteur électrique conçu pour l'agriculture durable et robotisée.

Caractéristiques principales du robot ALPO :

- électrique et écologique : le tracteur ALPO est entièrement électrique, émettant zéro CO2, ce qui contribue à la transition agroécologique ;
- polyvalence et robotisation : il est adapté à diverses cultures et peut être utilisé comme valet de ferme ou tracteur principal. Sa conception permet une automatisation complète pour des tâches agricoles variées ;
- collaboration avec le robot ZILUS : Dans le cadre de l'Accord Robotique, le tracteur ALPO travaille en tandem avec le robot tout-terrain ZILUS. Ensemble, ils réalisent des opérations culturales complémentaires, optimisant le temps et réduisant la pénibilité pour les agriculteurs.

L'équipe ROMEA utilise le robot ALPO comme plateforme d'expérimentation. Le tracteur ALPO a donc été adapté pour utiliser ROS2. Il est pilotable par le *joystick* d'origine dans la cabine, mais également par les algorithmes développés par l'équipe ROMEA afin de le rendre autonome.

Le tracteur électrique ALPO est un robot à deux ou quatre roues motrices utilisant une direction semi Ackermann, c'est à dire que les points de pivotement de direction des roues avant ne sont pas reliés par un tirant, chaque roue directrice est pilotée indépendamment par un vérin.



Figure 4 : robot autonome ALPO

Du point de vue cinématique, un tel robot peut être assimilé à un robot tricycle ayant une roue unique à l'avant. Le robot est alors piloté en donnant l'angle de cette roue ainsi que la vitesse linéaire du robot.

Le robot ALPO est donc de type 2FWS2FRWD ou 2FWS4WD.

L'équipe ROMEA fournit des packages ROS2 génériques adaptés aux différentes catégories de robots, notamment les packages capables de piloter le hardware du robot ALPO.

Le sujet traite de la conception d'un robot agricole adapté aux travaux agricoles. Il est divisé en quatre parties.

- la première partie est consacrée à l'analyse de la commande du robot à partir de sa géométrie ;
- la deuxième partie s'intéresse aux capteurs permettant au robot de se localiser dans son environnement ;
- la troisième partie s'intéresse à l'amélioration de la précision de la localisation ;
- la quatrième partie traitera de la sauvegarde des trajets dans une base de données et des aspects sécuritaire liés à son utilisation.

Partie 1 : Algorithmes de pilotage

Cette partie permet de déterminer les algorithmes de pilotage en fonction de la géométrie du tracteur ALPO.

> **Question 1 :** **Expliquer** en quoi l'utilisation de ROS2 est intéressante pour le TSCF de l'INRAE pour le développement de plateformes de robotiques agricoles.

Le TSCF utilise des robots ayant des architectures différentes, l'utilisation d'un middleware comme ROS2 permet de réutiliser le code et d'intégrer plus rapidement de nouveaux éléments (capteur, actionneur ou algorithmes). Uniformisation et standardisation de la plateforme de test.

Sous-partie 1.1 : Code de publication et de souscription aux *topics*

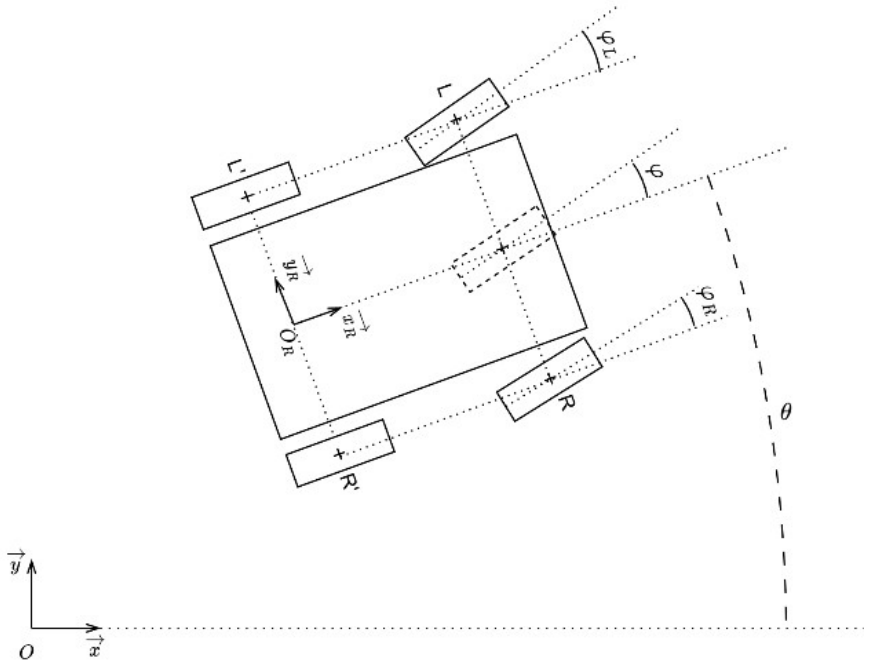


Figure 5 : schéma de la géométrie du robot ALPO

- $l = LR = L'R'$: voie (*wheelTrack*) ;
- $W = LL' = RR'$: empattement (*wheelBase*) ;
- $\frac{l}{2} = O_R R' = L' O_R$: demie voie (*halfTrack*) ;
- φ_L : Angle de braquage avant gauche (*frontLeftSteeringAngle*) ;
- φ_R : Angle de braquage avant droit (*frontRightSteeringAngle*) ;
- φ : Angle de braquage du tricycle équivalent (*SteeringAngle*) ;
- $\mathcal{R}_{robot} = (O_R, \vec{x_R}, \vec{y_R})$: repère lié au robot ALPO ;
- θ : Angle entre $\vec{x}$ et $\vec{x_R}$;
- $\mathcal{R} = (O, \vec{x}, \vec{y})$: repère du monde.

La commande du robot se fait avec 2 paramètres : l'angle φ (*steeringAngle*) de la roue tricycle équivalent et la vitesse longitudinale $V_{O_R \in robot/sol} \cdot \vec{x_R}$ (*linearSpeed*) qui sont publiés sur le *topic* ***/robot/base/controller/cmd_one_axle_steering***.

Les messages échangés sur les *topics* sont définis dans des fichiers .msg.

Afin de piloter le robot, le nœud **/robot/base/mobile_base_controller_fat** souscrit au *topic* **/robot/base/cmd_one_axle_steering** dans lequel est publié un message de type OneAxleSteeringCommand.msg dont la description est la suivante :

```
float64 longitudinal_speed   # m/s
float64 front_steering_angle # rad
```

Un programme minimum pour publier régulièrement ce type de message pourrait être le suivant (le fichier alpo_control.hpp est généré automatiquement par ROS à la compilation depuis le fichier OneAxleSteeringCommand.msg) :

```
#include "rclcpp/rclcpp.hpp"
#include " romea_mobile_base_msgs/OneAxleSteeringCommand.hpp"

int main(int argc, char * argv[])
{
  rclcpp::init(argc, argv);
  auto node = rclcpp::Node::make_shared("minimal_alpo_publisher");
  auto publisher = node->create_publisher<romea_mobile_base_msgs::msg::OneAxleSteeringCommand>(
    "/robot/base/cmd_one_axle_steering", 10);
  rclcpp::WallRate loop_rate(100ms);
  auto message = romea_mobile_base_msgs::msg::OneAxleSteeringCommand();
  message.longitudinal_speed = 1.0;
  message.steering_angle = 0.0;

  RCLCPP_INFO(node->get_logger(), "Publication sur /robot/base/cmd_one_axle_steering");

  while (rclcpp::ok()) {
    publisher->publish(message);
    rclcpp::spin_some(node);
    loop_rate.sleep();
  }
  rclcpp::shutdown();
  return 0;
}
```

Question 2 : **Indiquer**, à partir de ce code de publication minimum, quelle bibliothèque est utilisée pour publier sur un *topic*. **Préciser** quelles sont les valeurs des consignes de pilotage du robot ALPO dans cet exemple.

Rclcpp (#include rclcpp.hpp)

$V_{O_R \in robot/sol} = 1m/s$ et $\varphi = 0\,rad$

Une version minimale d'un programme permettant de souscrire au *topic* **/robot/base/controller/cmd_one_axle_steering** pourrait être :

```
#include "rclcpp/rclcpp.hpp"
#include "romea_mobile_base_msgs/OneAxleSteeringCommand.hpp"

int main(int argc, char * argv[])
{
  rclcpp::init(argc, argv);
  auto node = rclcpp::Node::make_shared("minimal_alpo_subscriber");

  auto subscription = node-
>create_subscription<romea_mobile_base_msgs::msg::OneAxleSteeringCommand>(
    "/robot/base/controller/cmd_one_axle_steering", 10,
    [](romea::core::OneAxleSteeringCommand::SharedPtr msg) {
      printf("Speed: %.2f m/s, Angle: %.2f rad\n", msg->longitudinal_speed, msg->steering_angle);
    });

  rclcpp::spin(node);
  rclcpp::shutdown();
  return 0;
}
```

Question 3 : **Indiquer** le résultat de la sortie console du programme de publication ci-dessus. **Préciser** quels sont les types des paramètres affichés.

Speed: 1.00 m/s, Angle: 0.00 rad

Speed: 1.00 m/s, Angle: 0.00 rad

…

Les paramètres sont de types float ou long float (%.2**f**)

Question 4 : À l'aide du document **DT1**, **déterminer** le nœud publiant sur le *topic /robot/base/controller/cmd_one_axle_steering*. **Justifier** la présence de ce nœud.

Le nœud */robot/base/cmd_mux* publie sur le *topic /robot/base/controller/cmd_one_axle_sterring*. Ce nœud souscrit aux *topics /robot/base/teleop/cmd_one_axle_steering* et */robot/path_following/cmd_one_axle_steering*. Le premier concerne la commande manuelle (Le *topic* racine est */robot/joystick/joy*), le second le suivi automatique de trajectoire. Il faut donc un nœud prioritisant les commandes publiées sur le nœud */robot/base/controller/cmd_one_axle_steering*

Sous-partie 1.2 : Equations cinématiques

ROS pilote donc le robot en publiant sur le *topic* ***/robot/base/controller/cmd_one_axle_steering*** les consignes de vitesse $V_{O_R \in robot/sol}$ et l'angle φ. Afin d'orienter correctement les roues avant directrices pour ces consignes, il faut, à partir de la géométrie du robot, déterminer les angles φ_L et φ_R. Le déplacement du robot est étudié sans glissement.

Par définition, le rayon de braquage est la distance entre O_R et le CIR (Centre Instantané de Rotation). La courbure (*curvature* en anglais) est alors l'inverse du rayon de braquage.

Question 5 : **Montrer** que $\varphi_L = atan(\frac{tan\varphi}{1-\frac{l}{2}\frac{1}{r}})$ avec r le rayon de braquage. **Donner** φ_R.

$tan\varphi_L = \frac{W}{r-l/2} = \frac{W}{r} \cdot \frac{1}{1-\frac{l}{2r}}$ or $tan\varphi = \frac{W}{r}$ donc $tan\varphi_L = \frac{tan\varphi}{1-\frac{l}{2r}}$

$\varphi_L = \arctan\left(\frac{\tan\varphi}{1-\frac{1}{r}\times\frac{l}{2}}\right)$ et $\varphi_R = \arctan\left(\frac{\tan\varphi}{1+\frac{1}{r}\times\frac{l}{2}}\right)$ et $r = \frac{W}{\tan\varphi}$ donc $c = \frac{\tan\varphi}{W}$

Question 6 : **Montrer** que $V_{L \in robot/sol} = V_{O_R \in robot/sol}\sqrt{\left(1-\frac{\frac{l}{2}}{r}\right)^2 + \tan^2(\varphi)}$. **Donner** $V_{R \in robot/sol}$.

Varignon :

$$\overrightarrow{V_{L \in robot/sol}} = \overrightarrow{V_{O_R \in robot/sol}} + \overrightarrow{\omega_{robot/sol}} \wedge \overrightarrow{O_R L}$$

$$\overrightarrow{V_{L \in robot/sol}} = V_{O_R \in robot/sol} \cdot \vec{x_R} + \frac{V_{O_R \in robot/sol}}{r} \cdot \vec{z} \wedge (W \cdot \vec{x_R} + \frac{l}{2} \cdot \vec{y_R})$$

$$\overrightarrow{V_{L \in robot/sol}} = V_{O_R \in robot/sol} \cdot \vec{x_R} + \frac{V_{O_R \in robot/sol}}{r} \cdot W \cdot \vec{y_R} - \frac{V_{O_R \in robot/sol}}{r}\frac{l}{2} \cdot \vec{x_R}$$

$$V_{L \in robot/sol} = V_{O_R \in robot/sol}\sqrt{\left(1-\frac{l}{2}{r}\right)^2 + \left(\frac{W}{r}\right)^2} = V_{O_R \in robot/sol}\sqrt{\left(1-\frac{l}{2}{r}\right)^2 + \tan^2(\varphi)}$$

De la même façon $V_{R\in robot/sol} = V_{O_R\in robot/sol}\sqrt{\left(1+\frac{\frac{l}{2}}{r}\right)^2 + \tan^2(\varphi)}$

Sous-partie 1.3 : Code pour piloter le robot ALPO

Le diagramme de classe *axle_steering* **DT2** présente les classes assurant le calcul des paramètres cinématiques de certains robots utilisés par l'équipe ROMEA, en particulier le robot ALPO. Les caractéristiques du robot sont stockées dans la structure *parameters*.

Question 7 : **Justifier** que *computeInstantaneouscurvature(double&,double&)* soit accessible depuis une instance de *TwoWheelSteeringKinematic*.

TwoWheelSteeringKinematic hérite de *OneAxleSteeringKinematic* et la méthode *computeInstantaneousCurvature* est déclarée publique dans celle-ci. Elle est donc utilisable

Question 8 : **Écrire** le constructeur de la structure *parameters* en initialisant les paramètres avec une valeur nulle.

```
//------------------------------------------------------------------------
TwoWheelSteeringKinematic::Parameters::Parameters()
: wheelBase(0),
  wheelTrack(0),
  wheelLinearSpeedVariance(0),
  wheelSteeringAngleVariance(0)
{
}
```

Question 9 : **Écrire** une implémentation de la méthode *OneAxleSteeringKinematic::computeInstantaneousCurvature()* retournant la courbure instantanée.

```
double OneAxleSteeringKinematic::computeInstantaneousCurvature(
  const double & tanSteeringAngle,
  const double & wheelBase)
{
  return tanSteeringAngle / wheelBase;
}
```

Question 10 : **Compléter**, sur le document réponse **DR1**, une implémentation des méthodes de la classe *TwoWheelSteeringKinematic* permettant de calculer les angles φ_L et φ_R ainsi que les vitesses linéaires des roues.

```
//------------------------------------------------------------------------
double TwoWheelSteeringKinematic::computeLeftWheelSteeringAngle(
  const double & tanSteeringAngle,
  const double & instantaneousCurvature,
  const double & halfTrack)
{
  return std::atan(tanSteeringAngle / (1 - instantaneousCurvature * halfTrack));
}

//------------------------------------------------------------------------
double TwoWheelSteeringKinematic::computeRightWheelSteeringAngle(
  const double & tanSteeringAngle,
  const double & instantaneousCurvature,
  const double & halfTrack)
{
  return std::atan(tanSteeringAngle / (1 + instantaneousCurvature * halfTrack));
}
 //------------------------------------------------------------------------
double OneAxleSteeringKinematic::computeWheelLinearSpeedRatio(
  const double & tanSteeringAngle,
  const double & instaneousCurvature,
  const double & halfTrack)
```

```
{
  return std::sqrt(std::pow(1 + instaneousCurvature * halfTrack, 2.) + tanSteeringAngle *
tanSteeringAngle);
}

//------------------------------------------------------------------------
double OneAxleSteeringKinematic::computeLeftWheelLinearSpeed(
  const double & linearSpeed,
  const double & tanSteeringAngle,
  const double & instaneousCurvature,
  const double & halfTrack)
{
  return linearSpeed * computeWheelLinearSpeedRatio(
    -tanSteeringAngle,
    -instaneousCurvature,
    halfTrack);
}

//------------------------------------------------------------------------
double OneAxleSteeringKinematic::computeRightWheelLinearSpeed(
  const double & linearSpeed,
  const double & tanSteeringAngle,
  const double & instaneousCurvature,
  const double & halfTrack)
{
  return linearSpeed * computeWheelLinearSpeedRatio(
    tanSteeringAngle,
    instaneousCurvature,
    halfTrack);
}
```

Le nœud ***/robot/base/mobile_base_controller_fat*** alimente une structure *OdometryFrame2FWS4WD* qui sera utilisée par la méthode *ControllerInterface2FWS4WD::write()* pour envoyer les commandes aux différents contrôleurs. Cette structure regroupe les paramètres (de type double) nécessaires à la commande d'un robot de type 2FWS4WD.

Question 11 : **Déclarer** la structure OdometryFrame2FWS4WD en utilisant des noms de variables adaptées.

```
struct OdometryFrame2FWS4WD
{
  double frontLeftWheelLinearSpeed;
  double rearLeftWheelLinearSpeed;
  double frontLeftWheelSteeringAngle;
  double frontRightWheelLinearSpeed;
  double rearRightWheelLinearSpeed;
  double frontRightWheelSteeringAngle;
};
```

Afin de contrôler le robot, le nœud ***/robot/base/mobile_base_controller_fat*** utilise une instance de la classe *mobile_base_controller* dont un extrait de la définition est donné ci-dessous :

```
template<typename InterfaceType, typename KinematicType>
class MobileBaseController : public controller_interface::ControllerInterface
{
 …
using MobileBaseController2FWS2RWD =
  MobileBaseController<ControllerInterface2FWS2RWD, core::TwoWheelSteeringKinematic>;
using MobileBaseController2FWS4WD =
  MobileBaseController<ControllerInterface2FWS4WD, core::TwoWheelSteeringKinematic>;
using MobileBaseController2WD =
  MobileBaseController<ControllerInterface2WD, core::SkidSteeringKinematic>;

 …
}
```

Question 12 : **Justifier** de l'intérêt d'utiliser un modèle de classe (*template*).

Les templates de classes sont des classes génériques qui permettent d'écrire un modèle de classe flexible qui fonctionne avec n'importe quel type de données, sans avoir à dupliquer le code pour chaque type.

Les robots autonomes possèdent de nombreux capteurs tels des récepteurs GNSS[3], LiDAR, caméras multiples, centrale inertielle, capteurs odométriques, etc. Certains sont proprioceptifs : ils effectuent leurs mesures par rapport à ce qu'ils perçoivent localement du déplacement du robot. D'autres sont extéroceptifs : ils se basent sur des mesures prises par rapport à son environnement global (repère absolu).

Sous-partie 1.4 : Odométrie

Il est nécessaire de pouvoir calculer la position du robot lorsqu'aucun capteur extéroceptif ne fournit de mesures. Dans ce cas le robot se déplace « à l'aveugle » (*dead reckoning* en anglais) et s'appuie uniquement sur ses capteurs proprioceptifs. L'odométrie consiste à estimer les déplacements du robot mobile en utilisant ses capteurs proprioceptifs.

> **Question 13 :** **Indiquer** dans quels cas le robot ALPO peut se retrouver dans une situation où seuls ses capteurs proprioceptifs fournissent de l'information.

Par exemple lorsque le robot ALPO se retrouve sans signal GPS et sans caméra (lorsqu'il entre dans un hangar métallique clos mal éclairé par exemple).

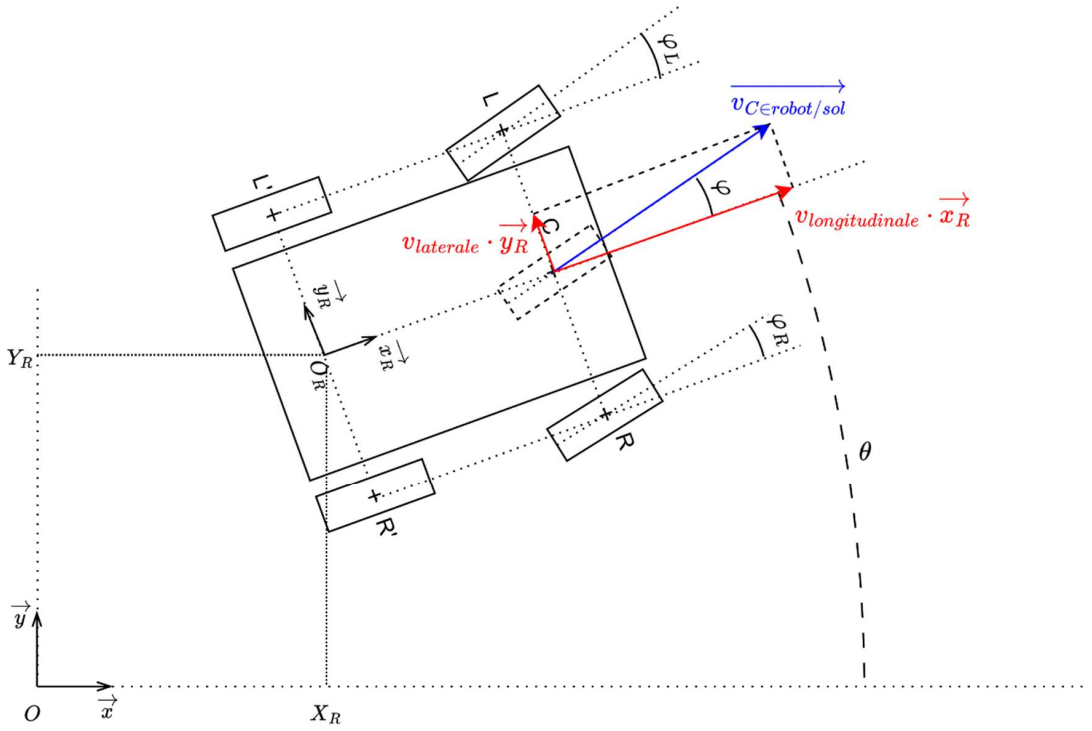


Figure 6 : schéma du robot ALPO dans le repère $\mathcal{R}$

> **Question 14 :** **Calculer** dx (déplacement selon $\vec{x}$) et dy (déplacement selon $\vec{y}$) lors d'un déplacement du robot à la vitesse $V_{C\in robot/sol}$ pendant un temps dt. Utiliser $\overrightarrow{V_{C\in robot/sol}} = V_{longitudinale} \cdot \overrightarrow{x_R} + V_{laterale} \cdot \overrightarrow{y_R}$.

[3] GNSS (*Global Navigation Satellite System*) est un terme générique qui englobe tous les systèmes mondiaux de navigation par satellites (BEIDOU, GALILEO, GLONASS, GPS etc.).

$$\overrightarrow{V_{C\in robot/sol}} = V_{longitudinale} \cdot \overrightarrow{x_R} + V_{laterale} \cdot \overrightarrow{y_R}$$

$$V_{longituginale} \cdot \overrightarrow{x_R} = V_{longitudinale} \cdot cos\theta \cdot \vec{x} + V_{longitudinale} \cdot \sin\theta \cdot \vec{y}$$

$$V_{laterale} \cdot \overrightarrow{y_R} = -V_{laterale} \cdot \cos\left(\frac{\pi}{2} - \theta\right) \cdot \vec{x} + V_{laterale} \cdot \sin\left(\frac{\pi}{2} - \theta\right) \cdot \vec{y}$$

$$= -V_{laterale} \cdot \sin\theta \cdot \vec{x} + V_{laterale} \cdot \cos(\theta) \cdot \vec{y}$$

donc

$$dx = V_{longitudinale} \cdot dt \cdot cos\theta - V_{laterale} \cdot dt \cdot \sin\theta$$

$$dy = V_{longitudinale} \cdot dt \cdot sin\theta + V_{laterale} \cdot dt \cdot \cos\theta$$

La méthode *update()* de la classe *DeadReckoning* met à jour les attributs contenant la position du robot lors d'un fonctionnement « à l'aveugle ».

romea::ros2::DeadReckoning
- previous_update_time_ : std::optional< rclcpp :: Time >
- x_ : double
- y_ : double
- theta_ : double
- previous_longitudinal_speed_ : double
- previous_lateral_speed_ : double
- previous_angular_speed_ : double
+ DeadReckoning() «constructor»
+ update(time : const rclcpp::Time&, kinematic_measure : const core::KinematicMeasure&)
+ getX() : const double&
+ getY() : const double&
+ getTheta() : const double&
+ reset()

Figure 7 : classe DeadReckoning

La structure *KinematicMeasure* contient les mesures des paramètres cinématiques du robot ALPO :

```
struct KinematicMeasure
{
  KinematicMeasure();
  double longitudinalSpeed;
  double lateralSpeed;
  double angularSpeed;
  double instantaneousCurvature;
  Eigen::Matrix4d covariance;
  EIGEN_MAKE_ALIGNED_OPERATOR_NEW
};
```

Question 15 : **Compléter** (sur votre copie) la méthode *Dead_Reckoning::update()* de mise à jour de l'odométrie du robot.

```
void DeadReckoning::update(
  const rclcpp::Time & time,
  const core::KinematicMeasure & kinematic_measure)
```

```
{
  if (previous_update_time_.has_value()) {
    double dt = (time - *previous_update_time_).seconds();
    //A compléter sur votre copie

    double dx = previous_longitudinal_speed_ * dt;
    double dy = previous_lateral_speed_ * dt;
    double ch = std::cos(theta_);
    double sh = std::sin(theta_);
```

```
    // dx and dy are rotated using the previous heading state
    x_ = x_ + dx * ch - dy * sh;
    y_ = y_ + dx * sh + dy * ch;
    theta_ = core::between0And2Pi(theta_ + previous_angular_speed_ * dt);
  }

  previous_longitudinal_speed_ = kinematic_measure.longitudinalSpeed;
  previous_lateral_speed_ = kinematic_measure.lateralSpeed;
  previous_angular_speed_ = kinematic_measure.angularSpeed;
  previous_update_time_ = time;
}
```

Question 16 : **Écrire** le modèle de classe (*template*) *between0And2Pi()* permettant de retourner un angle compris entre 0 et 2π pour un angle passé en paramètre compris entre -4π et 4π.

```
static constexpr auto M_2PI = 2 * M_PI;
static constexpr auto M_4PI = 4 * M_PI;
//-----------------------------------------------------------------------------
template<typename Scalar>
Scalar between0And2Pi(Scalar val)
{
  assert(val > -M_4PI && val < M_4PI);

  double value = std::fmod(val, M_2PI);
  return value < 0 ? value + M_2PI : value;
}
```

Sous-partie 1.5 : Code commande moteurs

Les moteurs utilisés pour la traction sont pilotables en openCAN. La classe *AlpoHardware* (voir **DT3**) assure ce pilotage en utilisant une instance de la classe *ros2_socketcan* fournie par ROS2 (voir **DT4**).

La méthode *send_data()* de la classe *AlpoHardware* est implémentée de la façon suivante :

```
bool AlpoHardware::send_data_(uint32_t id)
{
  try {
    drivers::socketcan::CanId can_id(id, 0, 8);
    can_sender_.send(sended_frame_data_.data(), 8, can_id, TIMEOUT);
    return true;
  } catch (drivers::socketcan::SocketCanTimeout & e) {
    RCLCPP_ERROR_STREAM(
      rclcpp::get_logger("AlpoHardware"),
      "Send can data" << std::hex << id << ": timeout");
  } catch (std::runtime_error & e) {
    RCLCPP_ERROR_STREAM(
      rclcpp::get_logger("AlpoHardware"),
      "Send can data" << std::hex << id << ": " << e.what());
  }
  return false;
}
```

Question 17 : **Écrire** l'implémentation de la méthode *encode_odo_data()* qui doit copier ses paramètres de type *FLOAT* dans l'attribut *sended_frame_data*, avant que soit appelée la méthode *send_data()* (voir annexes **DT6** et **DT7**).

```
void AlpoHardware::encode_odo_data_(float left_data, float right_data)
{
  std::memcpy(&sended_frame_data_[0], &left_data, sizeof(left_data));
  std::memcpy(&sended_frame_data_[4], &right_data, sizeof(right_data));
}
```

La méthode *decode_odo_data_()* (voir le diagramme de classe *AlpoHardware* **DT3**) est appelée par les méthodes *decode_front_wheel_speeds_()*, *decode_rear_wheel_speeds_()* et *decode_front_wheel_angles_()* pour obtenir les vitesses et angles correspondants mesurés par les

capteurs au niveau des moteurs et vérins. Ces mesures sont régulièrement envoyées sur le bus CANBUS par un thread qui exécute la méthode *received_data()*. Cette méthode copie les mesures dans l'attribut *received_data*. La méthode *decode_odo_data()* est appelée pour lire la valeur de *received_data* et les recopier dans les variables passées en paramètre.

> **Question 18 :** **Justifier** de l'intérêt, dans ce cas, d'utiliser le type *std ::atomic<float>* pour stocker les valeurs mesurées. **Proposer** un dispositif alternatif de programmation qui pourrait permettre d'utiliser des types *FLOAT* à la place (Voir **DT8**).

Les opérations sur les objets de type std::atomic<T> sont asynchrones, c'est-à-dire qu'une deuxième opération atomique sur le même objet à partir d'un thread différent peut obtenir l'état de l'objet uniquement avant ou après la première opération atomique. Ainsi, lors d'une écriture d'une nouvelle valeur d'odométrie la lecture est bloquée. Cela permet d'éviter l'utilisation d'un mutex.

Sous-partie 1.6 : Pilotage des roues directrices

Chacune des roues directrices à l'avant du robot ALPO sont orientées par un vérin. Le schéma cinématique ci-dessous en présente le fonctionnement.

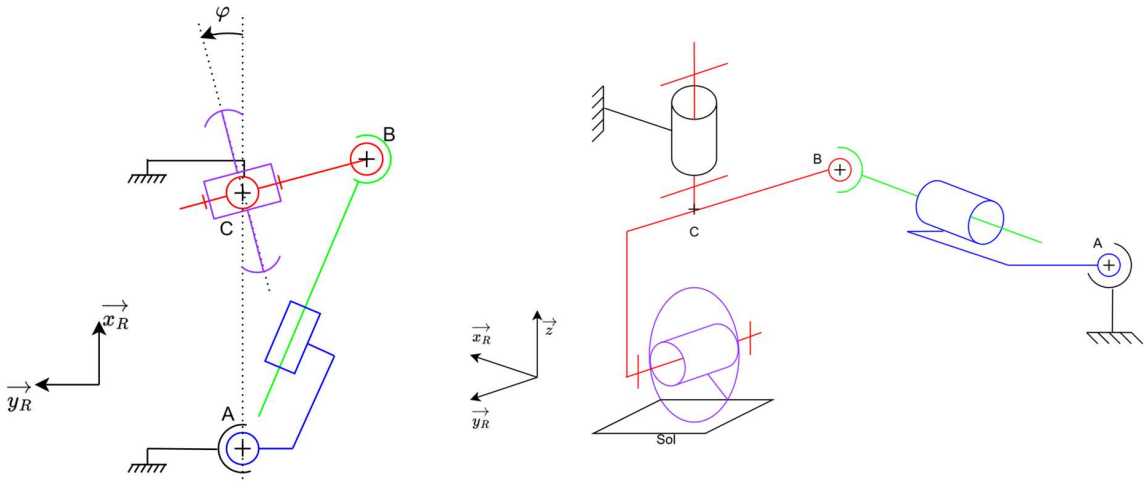


Figure 8 : schémas cinématiques de la direction de la roue du robot ALPO

- La longueur variable selon la longueur de tige du vérin sortie $AB = l = l_mini + \Delta l$;
- La longueur de l'axe de roue constant $BC = e$;
- La longueur entre l'attache du vérin et le pivot de roue $AC = d$;
- La course du vérin : Δl_max ;
- Les points A, B et C sont dans le plan $(\overrightarrow{x_R}, \overrightarrow{y_R})$.

Les vérins électriques sont pilotés, par l'intermédiaire d'une carte de puissance, par un signal PWM provenant d'un processeur STM32F072RB. Les vérins sont équipés d'un potentiomètre permettant d'estimer la longueur l grâce à une mesure de la tension à ses bornes. Le processeur réalise une conversion analogique numérique sur 12 bits de ces tensions.

Ce processeur est programmable en langage C et peut être connecté comme un nœud sur un bus de données CAN par l'intermédiaire d'un circuit spécialisé (MCP2551).

Le STM32F072RB va recevoir, par le bus de données CAN, les consignes des angles φ_L et φ_R depuis *AlpoHardware::send_data_()* sous forme de 2x4 octets représentant 2 *FLOAT*. Il renvoie les valeurs mesurées des angles $\varphi_{L_{mes}}$ et $\varphi_{R_{mes}}$ sur le bus de données CAN. Ces valeurs seront récupérées par la méthode *decode_front_wheel_angles_()* sous forme de 2x4 octets également.

Question 19 : **Montrer** que $\varphi = sin^{-1}(\frac{l^2-d^2-e^2}{2d})$.

Al Kashi : $l^2 = d^2 + e^2 - 2de \cdot \cos\left(\varphi + \frac{\pi}{2}\right) = d^2 + e^2 + 2de \cdot \sin(\varphi)$

Donc $\varphi = \sin^{-1}(\frac{l^2-d^2-e^2}{2de})$

OU

$$\overrightarrow{AB} = l.\overrightarrow{x_1}, \overrightarrow{CB} = -e.\overrightarrow{y_3}, \overrightarrow{AC} = d.\overrightarrow{x_3}$$

$$\overrightarrow{AB} + \overrightarrow{BC} + \overrightarrow{CA} = \vec{0}$$

$$= l.(cos\theta.\overrightarrow{x_R} + sin\theta.\overrightarrow{y_R}) + e(-sin\varphi.\overrightarrow{x_R} + cos\varphi.\overrightarrow{y_R}) - d.\overrightarrow{x_R} = \vec{0}$$

$$\begin{cases} lcos\theta - esin\varphi - d = 0 \\ lsin\theta + ecos\varphi = 0 \end{cases}$$

$$\begin{cases} (lcos\theta)^2 = (d + esin\varphi)^2 \\ (lsin\theta)^2 = (-ecos\varphi)^2 \end{cases}$$

$l^2 = (d + e.sin\varphi)^2 + (ecos\varphi)^2 = d^2 + e^2.sin^2\varphi + 2de.sin\varphi + e^2cos^2\varphi = d^2 + e^2 + 2de.sin\varphi$

Donc $\varphi = \sin^{-1}(\frac{l^2-d^2-e^2}{2de})$

En dehors de la configuration du processeur, le programme en C utilise les variables et les prototypes de fonctions suivants :

```c
/* Constantes géométriques */
#define D_CONSTANT 150.0f  // Constante d en mm
#define E_CONSTANT 200.0f  // Constante e en mm
#define LMINI_CONSTANT 100.0f  // Constante l_mini en mm
#define DLMAX_CONSTANT 200.0f  // Constante dl_max en mm

/* Adresses bus CAN */
#define CAN_ID_SETPOINTS_L1_L2  0x17  // Réception des consignes l1, l2
#define CAN_ID_FEEDBACK_P1_P2   0x26  // Envoi des positions p1, p2

/* Parametres Fuzzy PID */
typedef struct {
    float kp, ki, kd;
    float integral;
    float prev_error;
    float output_min, output_max;
} FuzzyPID_t;

/* Variables globales */
volatile float setpoint_l1 = 150.0f, setpoint_l2 = 150.0f; // Consignes de longueur depuis CAN
volatile float setpoint_p1 = 0.0f, setpoint_p2 = 0.0f;     // consignes d'angles calculées
volatile float actual_l1 = 0.0f, actual_l2 = 0.0f;         // Longueurs réelles mesurées
volatile float actual_p1 = 0.0f, actual_p2 = 0.0f;         // Angles réels calculés

FuzzyPID_t pid1 = {.kp = 2.0f, .ki = 0.1f, .kd = 0.5f, .output_min = -100.0f, .output_max =
100.0f};
FuzzyPID_t pid2 = {.kp = 2.0f, .ki = 0.1f, .kd = 0.5f, .output_min = -100.0f, .output_max =
100.0f};

float calculate_length_from_angle(float angle);
float calculate_angle_from_length(float length);
```

```
float fuzzy_pid_compute(FuzzyPID_t *pid, float setpoint, float input, float dt);
void read_analog_sensors(void);
void calculate_angular_setpoints(void);
void send_positions_can(void);
void control_actuators(float cmd1, float cmd2);
```

Question 20 : **Écrire** une implémentation des fonctions *calculate_length_from_angle()* et *calculate_angle_from_length()*.

```
float calculate_length_from_angle(float angle)
{
    /* l = sqrt(d² + e² + 2*d*e*sin(angle)) */
    float sin_angle = sinf(angle);
    return sqrtf(D_CONSTANT*D_CONSTANT + E_CONSTANT*E_CONSTANT +
                 2*D_CONSTANT*E_CONSTANT*sin_angle);
}

float calculate_angle_from_length(float length)
{
    /* angle = arcsin((l² - d² - e²) / (2*d*e)) */
    float numerator = length*length - D_CONSTANT*D_CONSTANT - E_CONSTANT*E_CONSTANT;
    float denominator = 2*D_CONSTANT*E_CONSTANT;
    float ratio = numerator / denominator;

    /* Limitation pour éviter les erreurs */
    if (ratio > 1.0f) ratio = 1.0f;
    if (ratio < -1.0f) ratio = -1.0f;

    return asinf(ratio);
}
```

La fonction *read_analog_sensors()* est appelée pour mesurer les longueurs l des vérins.

Question 21 : **Compléter** la fonction *read_analog_sensors()*

```
void read_analog_sensors(void)
{
    uint32_t adc_value1, adc_value2;

    /* Lecture de la tension du potentiomètre 1 (CAN 0) */
    HAL_ADC_Start(&hadc);
    HAL_ADC_PollForConversion(&hadc, 100);
    adc_value1 = HAL_ADC_GetValue(&hadc);

    /* Lecture de la tension du potentiomètre 2 (CAN 1) */
    HAL_ADC_Start(&hadc);
    HAL_ADC_PollForConversion(&hadc, 100);
    adc_value2 = HAL_ADC_GetValue(&hadc);

    /* Conversion CAN vers longueurs */
    // A compléter sur votre copie
    actual_l1 = LMINI_CONSTANT + (adc_value1 * DLMAX_CONSTANT) / 4095.0f;
    actual_l2 = LMINI_CONSTANT + (adc_value2 * DLMAX_CONSTANT) / 4095.0f;
}
```

1.6.1. BUS CAN

Un problème de place en mémoire empêche d'utiliser *string.h* et donc d'utiliser *memcpy()*.

Question 22 : **Proposer** une solution pour convertir les 2 x 4 octets *rx_data[0]* à *rx_data[7]* reçus sur le bus de données CAN en 2 *FLOAT* sans utiliser string.h (respecter l'alignement en mémoire).

Pour être sûr de respecter l'alignement en mémoire, il faut utiliser une union entre un float (4 octets) et un tableau de 4 uint8_t (4x1 octets)

```
typedef union {
    float f;
```

```
    uint8_t bytes[4];
} FloatBytes_t;
```

```
            FloatBytes_t l1_union, l2_union;

            l1_union.bytes[0] = rx_data[0];
            l1_union.bytes[1] = rx_data[1];
            l1_union.bytes[2] = rx_data[2];
            l1_union.bytes[3] = rx_data[3];
            l2_union.bytes[0] = rx_data[4];
            l2_union.bytes[1] = rx_data[5];
            l2_union.bytes[2] = rx_data[6];
            l2_union.bytes[3] = rx_data[7];

            setpoint_l1 = l1_union.f;
            setpoint_l2 = l2_union.f;
```

> **Question 23 :** **Indiquer** deux dispositifs intégrés au bus de données CAN qui permettent de fiabiliser l'intégrité des données qu'il transporte.

- le champ de CRC

- la procédure de « bit-stuffing » est utilisée.

Elle consiste, dans le cas où l'on a émis cinq bits de même polarité d'affilée, d'ajouter à la suite un bit de polarité contraire, pour casser des chaînes trop importantes de bits identiques.

Partie 2 : Mise en œuvre des capteurs

Cette partie s'intéresse au développement des programmes permettant aux capteurs du robot de le localiser dans son environnement.

Sous-partie 2.1 : Capteur IMU

2.1.1. Capteur XSense MTi-1

Le capteur Xsense MTi-1 est un capteur IMU 9DOF (voir **DT9**)

> **Question 24 :** **Expliquer** ce qu'est qu'un capteur IMU. **Préciser** ce que signifie 9DOF. **Détailler** ce que mesure le capteur Xsense Mti-1.

Inertial Mesurement Unit : centrale inertielle

9DOF (9 Degree Of Freedom)

Le capteur Xsens MTi1 contient 1 gyroscope 3 axes, 1 accéléromètre 3 axes et 1 magnétomètre 3 axes.

Il mesure les vitesses angulaires les accélérations et les champs magnétiques

Les caractéristiques du capteur sont données en annexe **DT9**.

> **Question 25 :** **Calculer** la dérive en position liée au bruit sur une durée de 10 s. **Conclure** sur la pertinence d'utilisation d'un capteur IMU.

La densité de bruit pour l'accélération est de $120\mu g/\sqrt{Hz}$ avec $f = 324Hz$ donc $\sigma_a = 120 \times 10^{-6} \times 9.81 \times \sqrt{324} = 0{,}021\ m/s^2$

Erreur de position après 1s (double intégration) :

$\frac{1}{2} \times 0.021 \times 10^2 = 1{,}06m$

→ Dérive de position de 1m en 10s.

L'IMU peut être utilisé pour déterminer la position pendant un temps assez court.

2.1.1.1. Bruits de mesure

D'une façon générale, les mesures issues des capteurs sont bruitées :

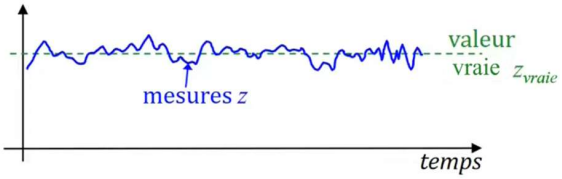


Figure 9 : mesures d'un capteur bruité en fonction du temps

Afin de prendre en compte l'incertitude de mesure et les bruits, il faut donc utiliser des variables aléatoires.

Une variable aléatoire prend des valeurs au hasard, et répond aux lois des probabilités.

La densité de probabilité de la variable aléatoire X est donc $\int_{-\infty}^{\infty} p(X)dX = 1$

En général, les variables aléatoires utilisées en robotique sont gaussiennes, c'est-à-dire qu'elles suivent des lois normales :

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

- μ : moyenne
- σ^2 : variance

Par exemple, un capteur sera modélisé par un modèle probabiliste tel que la probabilité d'avoir la mesure z sachant l'état x du système est :

$$p(z \mid x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z - f_{\text{capteur}}(x))^2}{2\sigma^2}}$$

C'est à dire qu'un capteur mesurant la grandeur x retourne $z = x + \mathcal{N}(\mu, \sigma^2)$. La valeur μ est alors un biais.

Afin de déterminer la probabilité d'avoir l'état x du système sachant z il suffira d'utiliser le théorème de Bayes.

« Le théorème de Bayes permet d'inverser les probabilités. C'est-à-dire que si l'on connaît les conséquences d'une cause, l'observation des effets permet de remonter aux causes. » (*wikipedia.org*)

$$P(A_i \mid B) = \frac{P(B \mid A_i) \times P(A_i)}{\sum_{j=1}^{n} P(B \mid A_j) \times P(A_j)}$$

En laboratoire, une expérimentation est menée pour déterminer le comportement de l'accéléromètre du capteur XSense MTi-1.

- a_x : accélération vraie $(m \cdot s^{-2})$

- z : mesure du capteur ($m \cdot s^{-2}$)
- $P(a_x|z)$: probabilité de l'accélération vraie sachant la mesure
- $P(z|a_x)$: probabilité de la mesure sachant l'accélération réelle

Une calibration du capteur donne les résultats suivants :

a_x vraie ($m \cdot s^{-2}$)	$z = 0{,}95$	$z = 0{,}98$	$z = 1.00$	$z = 1.02$	$z = 1{,}05$
0,90	0,15	0,05	0,02	0,01	0,00
0,95	0,40	0,20	0,05	0,02	0,01
1,00	0,05	0,15	0,50	0,15	0,05
1,05	0,01	0,02	0,05	0,20	0,40
1,10	0,00	0,01	0,02	0,05	0,15

Tableau 1 : tableau de calibration $P(z \mid a_x)$

Question 26 : Le capteur mesure $z = 1{,}02\, m \cdot s^{-2}$. **Calculer** la probabilité à posteriori $P(a_x|z = 1{,}02)$ pour toutes les valeurs de a_x possibles. **Préciser** la valeur de a_x la plus probable. La probabilité *a priori* $P(a_x)$ n'étant pas connue, une distribution uniforme $P(a_x)$ sera utilisée pour toutes les valeurs de a_x.

Pour x = 0.9 :

P(z=1.02 | x=0.9) * P(x=0.9) = 0.01 * 0.2 = 0.002

Pour x = 0.95 :

P(z=1.02 | x=0.95) * P(x=0.95) = 0.02 * 0.2 = 0.004

Pour x = 1.00 :

P(z=1.02 | x=1.00) * P(x=1.00) = 0.15 * 0.2 = 0.03

Pour x = 1.05 :

P(z=1.02 | x=1.05) * P(x=1.05) = 0.20 * 0.2 = 0.04

Pour x = 1.10 :

P(z=1.02 | x=1.10) * P(x=1.10) = 0.05 * 0.2 = 0.01

\sum{P(z=1.02|x) =P(z=1.02) = 0.002+0.004+0.03+0.04+0.01 = 0.086

Pour x = 0.9 :

P(z=1.02 | x=0.9) * P(x=0.9) / P(z=1.02) = 0.01 * 0.2 / 0.086 = 0.0233

Pour x = 0.95 :

P(z=1.02 | x=0.95) * P(x=0.95) / P(z=1.02) = 0.02 * 0.2 / 0.086 = 0.0465

Pour x = 1.00 :

P(z=1.02 | x=1.00) * P(x=1.00) / P(z=1.02) = 0.15 * 0.2 / 0.086 = 0.3488

Pour x = 1.05 :

P(z=1.02 | x=1.05) * P(x=1.05) / P(z=1.02) = 0.20 * 0.2 / 0.086 = 0.4651

Pour x = 1.10 :

P(z=1.02 | x=1.10) * P(x=1.10) / P(z=1.02) = 0.05 * 0.2 / 0.086 = 0.1163

Etant donné la mesure z=1.02 m/s², la valeur la plus probable pour l'accélération réelle a_x est 1.05 m/s²

Deux mesures temporelles des accélérations sont réalisées pour deux vitesses du robot, $0\ m \cdot s^{-1}$ et $1\ m \cdot s^{-1}$.

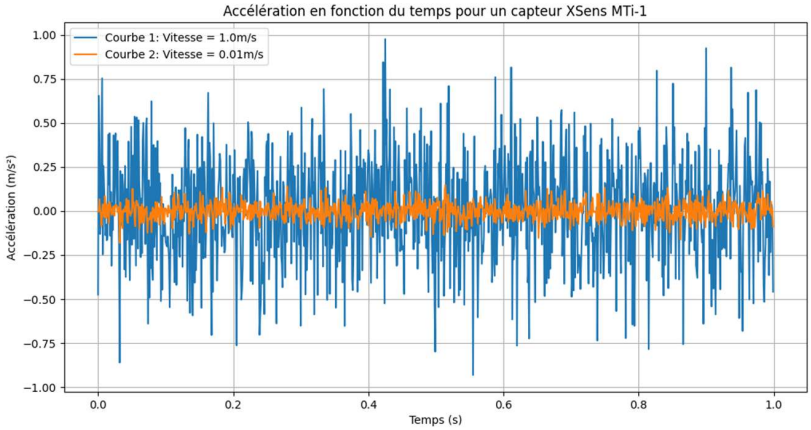


Figure 10 : accélérations en fonction du temps

Question 27 : **Indiquer** quels sont les facteurs pouvant augmenter les variances de mesure d'un capteur IMU dans ce système robotique.

Une IMU mesurant les accélérations et les vitesses angulaires est plus bruitée lors des déplacements. Sur les courbes d'accélération en fonction du temps, l'écart type (donc la variance) est plus faible à l'arrêt qu'à vitesse non nulle.

2.1.2. Algorithme *ZeroVelocityEstimator*

La classe *OnlineVariance* hérite de *OnlineAverage*. La méthode *update()* est codée :

```
void OnlineVariance::update(const double & value)
{
  std::lock_guard<std::mutex> lock(mutex_);
  long long int integerValue = static_cast<long long int>(value * multiplier_);
  long long int squaredIntegerValue = integerValue * integerValue;

  sumOfData_ += integerValue;
  sumOfSquaredData_ += squaredIntegerValue;

  if (data_.size() != windowSize_) {
    data_.push_back(integerValue);
    squaredData_.push_back(squaredIntegerValue);
  } else {
    sumOfData_ -= data_[index_];
    sumOfSquaredData_ -= squaredData_[index_];
    data_[index_] = integerValue;
    squaredData_[index_] = squaredIntegerValue;
  }

  double average = sumOfData_ / (double(multiplier_) * data_.size());
  double squaredAverage = (sumOfSquaredData_) / double(squaredMultiplier_);
  double variance = (squaredAverage - data_.size() * average * average) / (windowSizeMinusOne_);
  average_ = average;
  variance_ = variance;

  index_ = (index_ + 1) % windowSize_;
}
```

```
bool ZeroVelocityEstimator::update(
```

```
  const double & accelerationAlongXBodyAxis,
  const double & accelerationAlongYBodyAxis,
  const double & accelerationAlongZBodyAxis,
  const double & angularSpeedAroundXBodyAxis,
  const double & angularSpeedAroundYBodyAxis,
  const double & angularSpeedAroundZBodyAxis)
{
  varAccelerationAlongXBodyAxis_.update(accelerationAlongXBodyAxis);
  varAccelerationAlongYBodyAxis_.update(accelerationAlongYBodyAxis);
  varAccelerationAlongZBodyAxis_.update(accelerationAlongZBodyAxis);
  varAngularSpeedAroundXBodyAxis_.update(angularSpeedAroundXBodyAxis);
  varAngularSpeedAroundYBodyAxis_.update(angularSpeedAroundYBodyAxis);
  varAngularSpeedAroundZBodyAxis_.update(angularSpeedAroundZBodyAxis);

  if (varAccelerationAlongXBodyAxis_.isAvailable()) {

    return varAccelerationAlongXBodyAxis_.getVariance() < accelerationVarianceThreshold_ &&
           varAccelerationAlongYBodyAxis_.getVariance() < accelerationVarianceThreshold_ &&
           varAccelerationAlongZBodyAxis_.getVariance() < accelerationVarianceThreshold_ &&
           varAngularSpeedAroundXBodyAxis_.getVariance() < angularSpeedVarianceThreshold_ &&
           varAngularSpeedAroundYBodyAxis_.getVariance() < angularSpeedVarianceThreshold_ &&
           varAngularSpeedAroundZBodyAxis_.getVariance() < angularSpeedVarianceThreshold_;
  } else {
    return false;
  }
}
```

> **Question 28 :** **Indiquer** quelles sont les conditions pour que *ZeroVelocityEstimator ::update()* estime la vitesse nulle. **Justifier** en quoi, dans ces cas, la vitesse peut être réellement considérée nulle.

La méthode update retourne un booléen qui sera vrai si les variances des accélérations sur les 3 axes sont inférieures à un seuil `accelerationVarianceThreshold_` ET les variances des vitesses angulaires sont inférieures à un seuil `angularSpeedVarianceThreshold_`

En effet, si les variances sont faibles, c'est-à-dire que l'écart entre les mesures successives sont faibles, il est vraisemblable que le robot soit arrêté.

Sous-partie 2.2 : Camera stéréo

Cette partie valide l'utilisation d'une paire de caméras utilisée comme caméra stéréo pour l'observation de l'environnement du robot et la détection des obstacles.

2.2.1. Modélisation d'une caméra

2.2.1.1. Détermination des paramètres intrinsèques et extrinsèques

Pour la suite nous utiliserons un modèle de caméra dit « sténopé non inverseur », qui ne peut être réalisé en pratique mais qui est plus commode du point de vue mathématique car il évite d'inverser l'image.

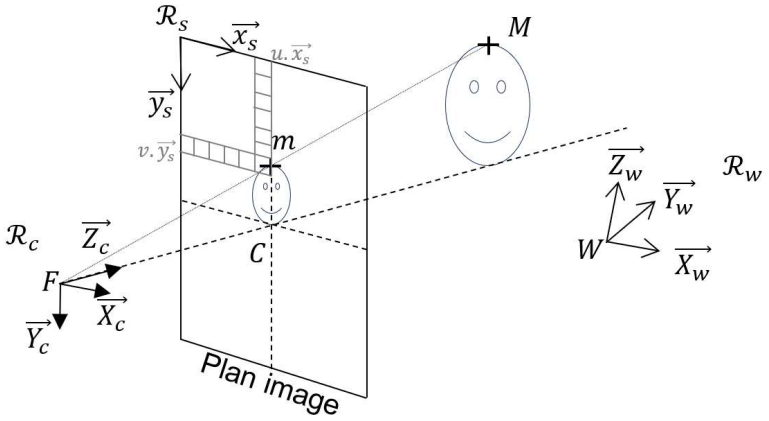


Figure 11 : schéma du sténopé non inverseur

- C : centre optique (centre du plan image) de coordonnées C_x, C_y dans le repère $\mathcal{R}_s$;
- F : point focal ;
- CF = focale f de la caméra ;
- $\mathcal{R}_W$: le repère du monde ;
- $\mathcal{R}_c$: le repère de la caméra ;
- $\mathcal{R}_s$: le repère du capteur de la caméra.

Modélisation d'une caméra

En vision par ordinateur, une caméra est modélisée par une matrice de projection P de dimension 3×4 telle qu'un point 3d M se projette sur le pixel $m = PM$.

Une matrice de projection se décompose en une matrice K des paramètres intrinsèques, et une matrice T des paramètres extrinsèques.

Paramètres extrinsèques

Les paramètres extrinsèques définissent le positionnement de la caméra dans l'espace 3D (transformations à appliquer pour effectuer le changement de repère de R_w vers R_c).

La matrice des paramètres extrinsèques peut donc être décomposée en une matrice de rotation R et d'un vecteur t de translation.

La matrice des paramètres extrinsèques se compose de la façon suivante :

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{12} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} = (R|t)$$

Avec r_{ij} les paramètres de rotation et t_k ceux de translation

Paramètres intrinsèques

La matrice des paramètres intrinsèques (pour des dimensions en pixels) se compose de la façon suivante :

$$K = \begin{bmatrix} f & 0 & C_x \\ 0 & f & C_y \\ 0 & 0 & 1 \end{bmatrix}$$

À noter que ces matrices sont exprimées en coordonnées homogènes. Les coordonnées homogènes sont une représentation mathématique utilisée en géométrie projective et en robotique pour simplifier les transformations (translations, rotations, changements d'échelle) dans un espace à n dimensions en les exprimant comme des multiplications matricielles dans un espace à n+1 dimensions :

- Pour un point 3D $[X\ Y\ Z]^T \rightarrow s \cdot [X\ Y\ Z\ 1]^T$ où s est un facteur d'échelle.

L'avantage d'utiliser les coordonnées homogènes est de simplifier, entre autres, les translations qui deviennent des multiplications matricielles.

Si le pixel $m(u, v)$ dans $\mathcal{R}_s$ est la projection du point $M(x, y, z)$ dans $\mathcal{R}_W$, alors :

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\mathrm{T} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Deux caméras de même référence commerciale sont utilisées. Afin de calibrer les caméras, un même damier est filmé avec les deux caméras. Les caractéristiques du damier sont connues et permettent à *openCV* de déterminer les positions des coins de celui-ci. La méthode *calibrateCamera()* d'*openCV* permet alors de calculer les matrices intrinsèques KL (caméra gauche) et KR (caméra droite) des caméras.

La calibration avec *calibrateCamera()* a donné les résultats suivants :

```
KL =  [[694.73129118   0.         323.36232288]
 [  0.         693.46394108 244.73613664]
 [  0.           0.           1.        ]]
KR=  [[672.9993327    0.         317.45531544]
 [  0.         670.58191223 248.92553524]
 [  0.           0.           1.        ]]
```

> **Question 29 :** **En déduire** la résolution probable des caméras utilisée. **Préciser** leur distance focale. **Déterminer** si les caméras sont identiques.

Résolution 640x480 car C est le centre de l'image et $C_x \approx 240$ et $C_y \approx 320$

Focale $f_L = 694\ pixels$ et $f_R = 670\ pixels$ soit $f_L = 694 \times 3\mu m = 2mm$ et $f_R = 2mm$

Les caméras ne sont pas tout à fait identiques

2.2.1.2. Caméra stéréo

Une expérimentation est réalisée avec les deux caméras afin de déterminer si un tel système est utilisable comme caméra stéréo afin d'avoir un capteur de distance pour le robot ALPO.

Le montage des caméras est tel que les plans images sont confondus et les axes optiques sont parallèles. Les caméras sont séparées d'une distance b selon la direction $\overrightarrow{x_S}$.

Un point M de coordonnées $[X\ Y\ Z\ 1]^T$ dans $\mathcal{R}_c$ est vu dans l'image de gauche comme le point m_L de coordonnées $[u_L v_L 1]^T$ et sur la caméra de droite comme le point m_R de coordonnées $[u_R v_R 1]^T$.

On notera la disparité (*disparity*) d telle que : $u_R - u_L = d$.

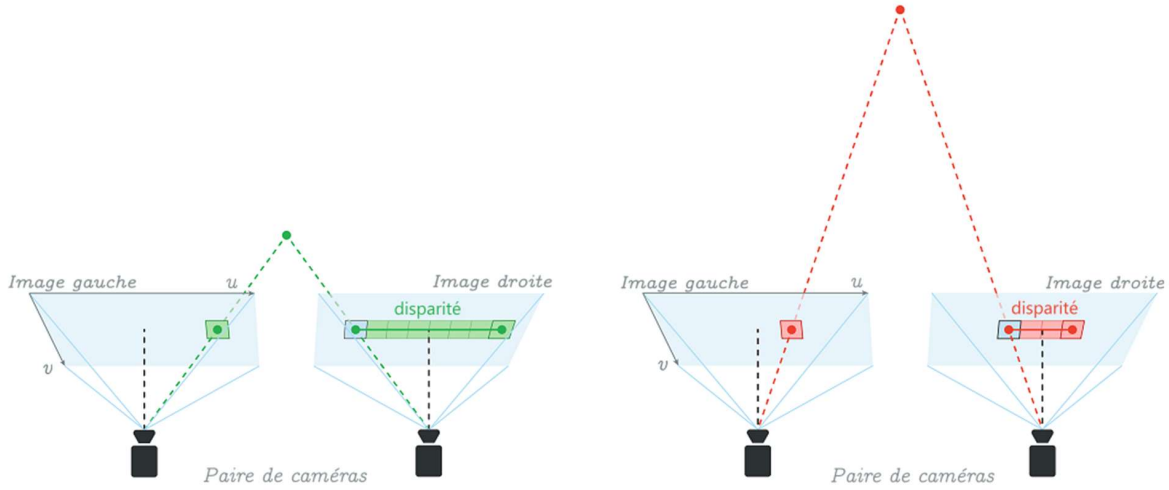


Figure 12 : disparités pour une caméra stéréo

Question 30 : **Écrire** la matrice $[X\ Y\ Z\ 1]^T$ en fonction de $f, b, d, u_L, v_L, C_x\ et\ C_y$.

$$s_L.\begin{bmatrix} u_L \\ v_L \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & C_x \\ 0 & f & C_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$u_L = \frac{fX + C_xZ}{Z} \text{ et}$$
$$v_L = \frac{fY + C_yZ}{Z}$$

$$s_R.\begin{bmatrix} u_R \\ v_R \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & C_x \\ 0 & f & C_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & b \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$u_R = \frac{f(X + b) + C_xZ}{Z} \text{ et}$$
$$v_R = \frac{fY + C_yZ}{Z}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{b(u_L - C_x)}{d} \\ \frac{b(v_L - C_y)}{d} \\ \frac{fb}{d} \\ 1 \end{bmatrix}$$

Question 31 : **Déterminer** v_L et v_R. **Expliquer** ce qu'implique la relation entre v_L et v_R.

$v_L = \frac{fY+C_yZ}{Z}$ et $v_R = \frac{fY+C_yZ}{Z}$ donc $v_L = v_R$

Cette égalité signifie qu'un même point M de R_c est projeté à la même hauteur sur les deux images. L'appariement des points est ainsi facilité. Pour apparier un point de l'image de droite à un point de celle de gauche, il suffit de parcourir la même ligne sur l'image de gauche et identifier le point correspondant.

Afin de calibrer la caméra stéréo, le même damier que précédemment est filmé.

La méthode *stereoCalibrate()* permet de définir les matrices de rotation R et de translation t à appliquer pour transformer les points donnés dans le système de coordonnées de la première caméra en points dans le système de coordonnées de la deuxième caméra.

La calibration avec *calibrateCamera()* et *stereoCalibrate()* a donné les résultats suivants :

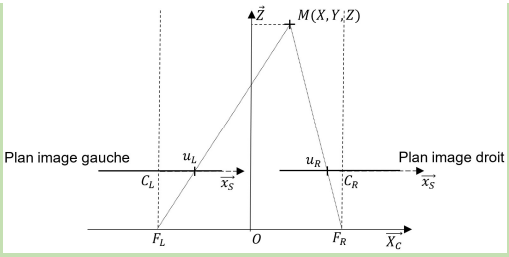
```
R =  [[ 0.99986465 -0.01071751  0.01248304]
 [ 0.01119544  0.99918168 -0.03886691]
 [-0.01205627  0.0390014   0.99916642]]
t =  [[ 0.15626554]
 [ 0.00094655]
 [-0.01880167]]
```

Question 32 : **Montrer** que les caméras sont alignées. **Préciser** la distance b séparant les caméras.

$R \sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ et $T \sim \begin{bmatrix} 0.156 \\ 0 \\ 0 \end{bmatrix}$ donc pas de rotation pour passer d'une image à l'autre et seulement une translation selon $\overrightarrow{x_R}$, les caméras sont alignées.

$b = 0.156\ m$

Question 33 : **Montrer** que $Z = \frac{b}{d} f$, en **déduire** comment calculer la distance d'un point.



Cf Q30 ou avec Thales : $\frac{b}{Z} = \frac{b+u_R-u_L}{Z-f}$ donc $Z = \frac{bf}{d}$

La distance Z est en m car b en m et f et d en pixels.

2.2.2. Appariement des points

La somme des différences absolues (SAD) est le critère d'appariement le plus courant dans les algorithmes d'appariement stéréo, en raison de sa faible complexité, de ses bonnes performances et de la facilité de son implémentation matérielle.

La SAD calcule la somme des différences absolues par éléments de deux fenêtres W^L et W^R de taille $M \times N$,extraites des deux images stéréos.

$$\mathrm{SAD}(W^L, W^R) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left| W_{ij}^L - W_{ij}^R \right|$$

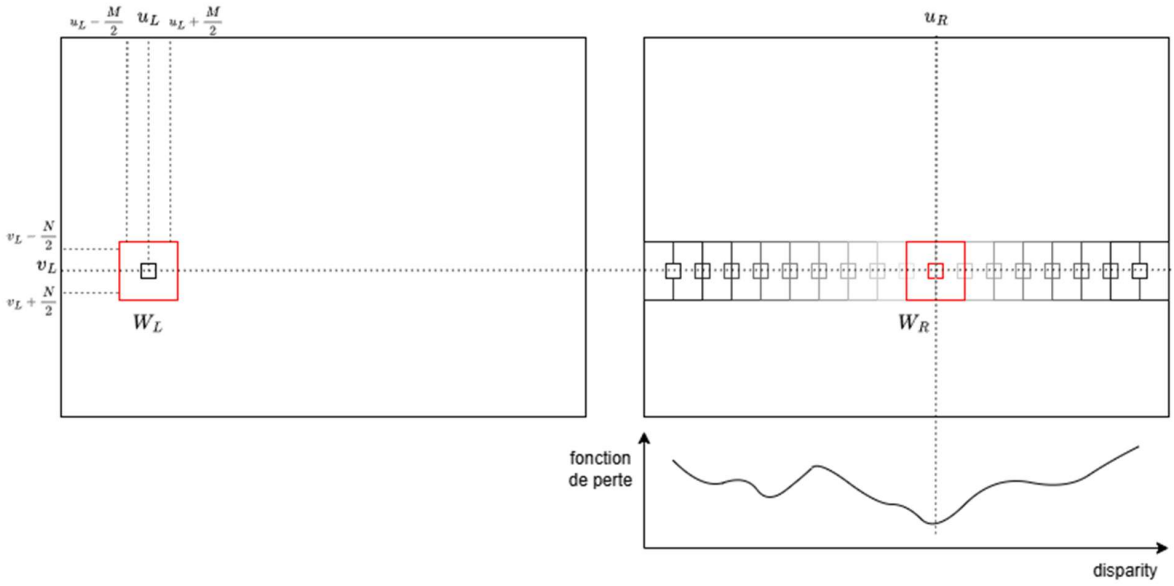


Figure 13 : calcul de la similarité pour chaque bloc de l'image

La carte des disparités se calcule en appariant chaque pixel de l'image de gauche avec un pixel de l'image de droite. De plus, les images stéréos étant alignées (par construction et par rectification logicielle), le pixel correspondant sur l'image de droite (u_R, v_R) est nécessairement sur la même ligne que le pixel (u_L, v_L) sur l'image de gauche. La valeur de disparité est alors $d = u_R - u_L$.

Ainsi, en parcourant l'image de gauche pixel par pixel, il s'agit de trouver le pixel de l'image de droite pour lequel la fonction de perte est minimale. Cette fonction de perte pourra utiliser la somme des différences absolues.

Question 34 : **Écrire** une fonction Python calculant la fonction de perte SAD. Cette fonction prendra en paramètres 2 *numpy.ndarray* représentant W^L et W^R (voir **DT10**)

```
import numpy as np

def sad(WL, WR):
    if WL.shape != WR.shape:
        return -1
    return np.sum(np.abs(WL - WR))
```

Question 35 : **Compléter** le code ci-dessous qui utilise l'algorithme SAD pour apparier les pixels sur les deux images et qui retourne un tableau contenant les disparités.

```
def calculate_disparity(left_image, right_image, window_size):

    left_shape = left_image.shape
    height = left_shape[0]
    width = left_shape[1]
    disparity_map = np.zeros((height, width))

    # calcul de la carte des disparités
    for i in range(window_size//2, height-window_size//2):
        for j in range(window_size//2, width-window_size//2):
            left_block = left_image[i-window_size//2:i+window_size//2+1, j-
window_size//2:j+window_size//2+1]
            cost, best_cost, best_match = 0, np.inf, 0
            for k in range(window_size//2, width-window_size//2):
                right_block = right_image[i-window_size//2:i+window_size//2+1, k-
window_size//2:k+window_size//2+1]
                cost = sad(left_block - right_block)
                if cost < best_cost:
                    best_cost = cost
```

```
                    best_match = k

            # calcul des disparités
            disparity_map[i, j] = np.abs(j - best_match)

    # Normalisation
    normalized_disparity_map = (disparity_map - np.min(disparity_map)) / (np.max(disparity_map) -
np.min(disparity_map)) * 255
    return disparity_map, normalized_disparity_map
```

2.2.3. Connexion avec ROS

Le programme exploitant la caméra utilise *openCV* pour obtenir une image de profondeur. Le robot utilise ROS2, il faut donc convertir cette image de profondeur en un message compatible avec ROS2.

Le package ROS2 *sensor_msgs* fournit un type de message pour les nuages de points 3D nommé *PointCloud2*. Le type *sensor_msgs/PointCloud2* permet de publier des messages donnant les coordonnées des points dans le monde réel. La structure du message *PointCloud2* est donnée en annexe **DT5**.

Question 36 : **Montrer** que $X = \frac{(u_L - C_x)Z}{f}$ et $Y = \frac{(v_L - C_y)Z}{f}$.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{b(u_L - C_x)}{d} \\ \frac{b(v_L - C_y)}{d} \\ \frac{fb}{d} \\ 1 \end{bmatrix} \text{ donc } \frac{Z}{f} = \frac{b}{d}, X = \frac{(u_L - C_x)Z}{f} \text{ et } Y = \frac{(v_L - C_y)Z}{f}$$

Un nœud ROS doit donc être utilisé pour publier les profondeurs dans un nuage de points de type *PointCloud2*. Ce nuage de points indique les coordonnées (X, Y, Z) du point dans le repère $\mathcal{R}_C$.

OpenCV fournit une image comme une variable de type MAT qui représente une matrice n-dimensionnelle dense. La classe *cv::Mat* possède des attributs publics, notamment *cols* et *rows* qui contiennent le nombre de lignes et de colonnes de l'image. La classe *cv::Mat* d'*OpenCV* possède également la méthode *.at()* qui permet d'accéder à un pixel spécifique dans une image ou une matrice. Pour une image 2D la syntaxe est :

```
mat.at<Type>(y, x)
```

Dans notre cas, les profondeurs (en mm) sont stockées dans une matrice de type cv::Mat comme des uint16_t.

Question 37 : **Compléter**, sur le document réponse **DR2**, le code permettant de produire le message *cloud_msg*.

```
…
void depthToPointCloud2(
    const sensor_msgs::msg::Image::ConstSharedPtr& depth_msg,
    const sensor_msgs::msg::CameraInfo::ConstSharedPtr& cam_info,
    sensor_msgs::msg::PointCloud2& cloud_msg)
{
    // Paramètres intrinsèques
    // les float fx, fy, cx, cy contiennent les paramètres intrinsèques

    // depth est de type cv::Mat
    …
    // Préparation du message PointCloud2
    cloud_msg.header = depth_msg->header;
    cloud_msg.height = depth.rows;
```

```
    cloud_msg.width = depth.cols;
    cloud_msg.is_dense = false;
    cloud_msg.is_bigendian = false;
    cloud_msg.fields = {
        sensor_msgs::msg::PointField{"x", 0, sensor_msgs::msg::PointField::FLOAT32, 1},
        sensor_msgs::msg::PointField{"y", 4, sensor_msgs::msg::PointField::FLOAT32, 1},
        sensor_msgs::msg::PointField{"z", 8, sensor_msgs::msg::PointField::FLOAT32, 1}
    };
    cloud_msg.point_step = 12;
    cloud_msg.row_step = cloud_msg.point_step * depth.cols;
    // La méthode resize est utilisée pour redimensionner le vecteur de données afin qu'il ait
    // exactement la taille nécessaire pour contenir toutes les données.
    cloud_msg.data.resize(cloud_msg.row_step * depth.rows);

    // Remplir le nuage de points
    sensor_msgs::PointCloud2Iterator<float> iter_x(cloud_msg, "x");
    sensor_msgs::PointCloud2Iterator<float> iter_y(cloud_msg, "y");
    sensor_msgs::PointCloud2Iterator<float> iter_z(cloud_msg, "z");

    for (int v = 0; v < depth.rows; ++v) {
        for (int u = 0; u < depth.cols; ++u, ++iter_x, ++iter_y, ++iter_z) {
            float z = depth.at<uint16_t>(v, u) * 0.001f; // conversion mm->m
            if (z == 0) {
                *iter_x = *iter_y = *iter_z = std::numeric_limits<float>::quiet_NaN();
                continue;
            }
            *iter_x = (u - cx) * z / fx;
            *iter_y = (v - cy) * z / fy;
            *iter_z = z;
        }
    }
}
```

Question 38 : **Expliquer** ce que réalise la ligne 42 du code présenté sur **DR2**.

```
float z = depth.at<uint16_t>(v, u) * 0.001f; // conversion mm->m
```

Cette ligne récupère la valeur de la profondeur dans l'image cv::Mat depth aux coordonnées (u,v). La méthode at() doit préciser le type récupéré pour cette profondeur. Ensuite cette valeur est multipliée par 0.001f pour la convertir en mètre et en float.

2.2.4. Mise en œuvre de la caméra stéréo OAK D Lite

Afin de gagner en efficacité, une caméra stéréo du commerce est utilisée. La caméra OAK D Lite possède les caractéristiques ci-dessous.

Cette caméra possède un NPU d'une puissance de calcul de 4 TOPS. Ce processeur est donc capable de fournir rapidement, non seulement une *disparity map* (image où les pixels codent la disparité) mais également de faire tourner des algorithmes d'intelligence artificielle.

Cette caméra est capable d'exploiter les modèles d'IA spécialisés dans la reconnaissance d'objets.

La caméra OAK-D Lite peut être paramétrée pour intégrer le modèle YOLO qui assurera la détection d'objets directement sur la caméra. Le modèle YOLO est particulièrement adapté pour la reconnaissance d'objets.

Les modèles de détection d'objets comme YOLO produisent des sorties qui incluent un identifiant de classe sous forme numérique. Cela permet une représentation compacte et efficace des résultats de détection. Chaque classe, dans le *dataset* d'entraînement, est associée à un identifiant numérique unique. Lors de l'inférence, le modèle prédit ces identifiants numériques pour les objets détectés.

Le *dataset* COCO a été utilisé pour l'entrainement. Le fichier *coco.yaml* (voir extrait en annexe **DT11**) contient les identifiants des classes.

Un premier code (annexe **DT13**) a été élaboré à partir d'un exemple fourni par le fabricant. Ce code publie sur le *topic* ***/robot/camera/detections*** un message personnalisé dans lequel est précisé si une personne est détectée et la distance à laquelle elle est détectée.

Question 39 : **Compléter** (sur votre copie) le code ci-dessous pour détecter également les panneaux stop (voir annexe **DT13**),

```
    objectTracker->setDetectionLabelsToTrack({0,11});
…
    while(pipeline.isRunning()) {
        auto imgFrame = preview->get<dai::ImgFrame>();
        auto track = tracklets->get<dai::Tracklets>();

        bool personDetected = false;
        float personZDistance = 0.0f;
        bool stopSignDetected = false;
        float stopSignZDistance = 0.0f;

        auto trackletsData = track->tracklets;
        for(const auto& t : trackletsData) {
            if(t.label == 0) { // Personne
                personDetected = true;
                personZDistance = t.spatialCoordinates.z;
            } else if(t.label == 11) { // Panneau Stop
                stopSignDetected = true;
                stopSignZDistance = t.spatialCoordinates.z;
            }
        }
        // Créer et publier le message
        auto message = camera::msg::PersonDetection();
        message.person_detected = personDetected;
        message.person_z_distance = personZDistance;
        message.stop_sign_detected = stopSignDetected;
        message.stop_sign_z_distance = stopSignZDistance;
        message.header.stamp = node->now();

        publisher->publish(message);
        rclcpp::spin_some(node);
    }
…
```

Question 40 : À l'aide de l'annexe **DT5**, **écrire** le contenu du fichier PersonDetection.msg.

```
bool person_detected
float32 person_z_distance
bool stop_sign_detected
float32 stop_sign_z_distance
```

Sous-partie 2.3 : GNSS RTK

La correction GNSS RTK[4] repose sur l'utilisation de données de correction provenant d'une station fixe de référence (*base*) pour améliorer la précision de la position calculée par un récepteur mobile (*rover*). Les données de correction sont diffusées en temps réel directement de la *base* au *rover* ou via un serveur NTRIP, permettant au *rover* de compenser les erreurs de propagation des signaux GNSS et d'obtenir une position précise à quelques centimètres près.

2.3.1. Correction RTK

Question 41 : **Préciser** quelles sont les conditions sur la *base* pour que la correction de la position du *rover* puisse fonctionner.

[4] RTK (Real Time Kinematic)

La position de la base doit être connue avec précision et la distance entre la base et le rover ne doit pas être trop grande pour que les perturbations de propagation des signaux GNSS soient les mêmes.

2.3.2. Modules GNSS RTK LC29H

Pour déterminer la position du robot ALPO, un essai est réalisé avec deux modules LC29H de la société QUECTEL. L'avantage de ces modules est qu'ils sont paramétrables via leur port série. Le module *rover* peut calculer les corrections à apporter à sa position GNSS pour tenir compte des données de correction du module *base*. Ils ne nécessitent pas de calculateur externe pour calculer ces corrections.

Un extrait du protocole de communication est donné en annexe **DT12**. Ce protocole permet aussi bien de contrôler les modules que de changer leur configuration.

> **Question 42 :** **Préciser** les commandes permettant de paramétrer le LC29H du robot en mode ROVER. Le détail du calcul du checksum n'est pas demandé.

$PQTMRESTOREPAR*13 (optionnel, c'est dans la doc pour être sûr de modifier les paramètres sauvegardés)

$PQTMCFGRCVRMODE,W,1*<checksum> <CR><LF>

$PQTMSAVEPAR*5A

> **Question 43 :** **Élaborer** la commande permettant de paramétrer le LC29H du robot pour qu'il corrige sa position 5 fois par seconde. Le détail du calcul du checksum n'est pas demandé.

$PAIR050,200*<checksum> <CR><LF>

> **Question 44 :** **Élaborer** les commandes permettant de paramétrer le LC29H de la base en mode BASE puis de paramétrer sa position exacte (utiliser X, Y et Z comme coordonnées en mètre de la base). Le détail du calcul du checksum n'est pas demandé.

$PQTMRESTOREPAR*13

$PQTMCFGRCVRMODE,W,2*<checksum> <CR><LF>

$PQTMCFGSVIN,W,2,0,0,X,Y,Z*<checksum> <CR><LF>

$PQTMSAVEPAR*5A

Afin de simplifier le calcul du checksum, une fonction écrite en python doit être réalisée.

> **Question 45 :** **Écrire** une fonction en Python calculant le checksum des commandes (entre $ et * exclus) permettant de paramétrer et contrôler le circuit GNSS LC29H.

```python
def calculate_checksum(commande):
    checksum = 0
    for char in commande:
        checksum ^= ord(char)  # XOR entre le checksum actuel et le code ASCII du caractère
    return checksum
```

Partie 3 : Fusion de capteurs

Cette partie met en œuvre les codes permettant d'améliorer la précision des mesures issues des capteurs.

Sur un robot de nombreux capteurs sont utilisés. Ils mesurent parfois les mêmes grandeurs de façon directe ou non. Par exemple un LiDAR mesure une distance, la caméra stéréo également.

Les capteurs sont modélisés de la façon suivante :

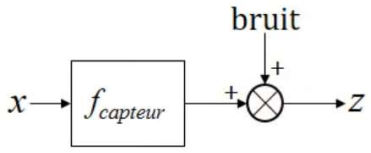


Figure 14 : modélisation d'un capteur

Soit $z \sim N\left(f_{\text{capteur}}\,(x), \sigma^2_{\text{capteur}}\right)$ avec N la normale, $\sigma^2_{\text{capteur}}$ la variance du capteur

> **Question 46 :** **Montrer** que pour deux mesures issues de capteurs, $z_1 \sim N\left(f_1\,(x), \sigma_1^2\right)$ et $z_2 \sim N\left(f_2\,(x), \sigma_2^2\right)$, il existe une pondération optimale w telle que $z_3 = (1-w)z_1 + wz_2$ permettant de réduire la variance σ_3^2.

La moyenne pondérée des 2 capteurs s'écrit :

$$z_3 = (1-w)z_1 + wz_2$$

Donc les variances $\sigma_3^2 = (1-w)^2\sigma_1^2 + w^2\sigma_2^2$ (car $f_{1(x)}$ et $f_{2(x)}$ sont indépendants donc $covar(f_{1(x)}, f_{2(x)}) = 0$)

Pour minimiser σ_3^2, on cherche $\frac{d}{dw}\sigma_3^2 = 0$

$\frac{d}{dw}\sigma_3^2 = \frac{d}{dw}\{(1-w)^2\sigma_1^2 + w^2\sigma_2^2\} = 2(1-w)\sigma_1^2 + 2w\sigma_2^2 = 0$ donc $w = \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$

Sous-partie 3.1 : Localisation du robot

La problématique d'un robot tel que le robot ALPO est de connaitre sa pose (sa position, son orientation, sa vitesse etc.) au cours de ses déplacements. Cela revient à chercher à connaitre la probabilité de l'état du robot au temps présent connaissant toutes les commandes passées envoyées aux actionneurs ainsi que toutes les mesures passées provenant des capteurs.

$$bel(s_t) = P(s_t = s | u_{0:t}, z_{1:t})$$

avec

- $bel()$: la fonction de croyance
- s : la pose réelle du robot
- s_t : état du robot au temps t
- $u_{\{0:t\}} = \{u_0, u_2, \dots, u_t\}$: commandes passées aux actionneurs ou mesures des capteurs proprioceptifs
- $z_{\{1:t\}} = \{z_1, z_2, \dots, z_t\}$: mesures passées en provenance des capteurs extéroceptifs

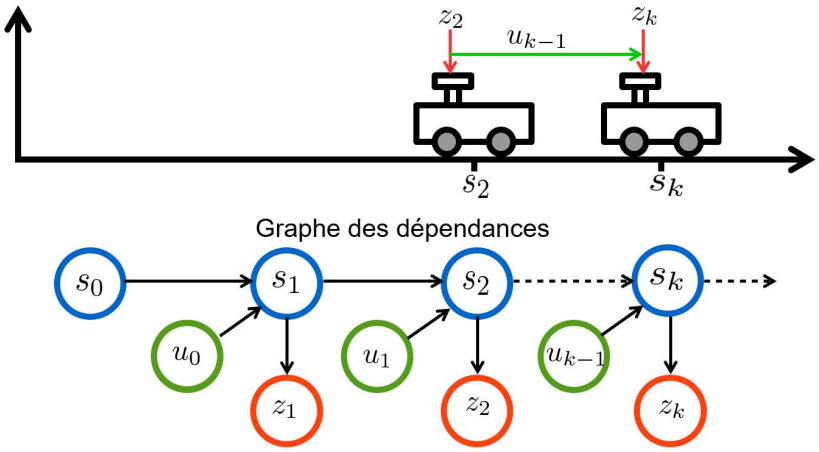


Figure 15 : graphe des dépendances pour un robot se déplaçant

En considérant que l'état du robot est complet, c'est-à-dire qu'il respecte la propriété de Markov, et que les variables sont indépendantes, alors :

$P(s_t|u_{0:t}, z_{1:t}) = P(s_t|s_{t-1}, u_t)$: équation du modèle de déplacement.

$P(z_t|s_{0:t}, u_{0:t-1}, z_{1,t-1}) = P(z_t|s_t = s)$: équation du modèle de capteur.

> **Question 47 :** **Indiquer** comment évolue l'incertitude de la pose lorsqu'une commande est appliquée (avant mesures). **Préciser** comment elle évolue après mesures.

La commande étant bruitée l'incertitude en position augmente, elle diminue ensuite avec la mesure.

La commande du robot est modélisée par : $u \sim \mathcal{N}(u_k, Q_k)$.

La mesure des capteurs est modélisée par : $z \sim \mathcal{N}(z_k, R_k)$.

Dans notre cas, la pose est donnée par $s = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$, la commande par $u = \begin{bmatrix} V_{longitudinale} \\ V_{laterale} \\ \omega_{robot/sol} \end{bmatrix}$.

Le modèle dynamique du robot est modélisé par : $\hat{s}_{k|k-1} = f_s(\hat{s}_{k-1|k-1}, u_k) + w_k$

- $\hat{s}_{k|k-1}$: estimée de la pose au temps k connaissant celle à $k-1$;
- f_s : fonction d'état ;
- $w_k \sim \mathcal{N}(0, T_k)$: bruit d'évolution, gaussien et de matrice de covariance T_k.

Les capteurs sont modélisés par : $\hat{z}_k = h_z(\hat{s}_{k|k-1}) + v_k$

- $\hat{z}_k$: estimée de la mesure au temps k ;
- h_z : fonction de mesure ;
- $v_k \sim \mathcal{N}(0, R_k)$: bruit de mesure, gaussien et de matrice de covariance R_k.

Alors les équations du filtre de Kalman étendu permettent d'estimer :

$bel(s) \sim \mathcal{N}(s_{k|k}, P_{k|k})$

Équations de prédiction (après une commande u_k) :

$\hat{s}_{k|k-1} = f_s(\hat{s}_{k-1|k-1}, u_k)$ (EKF1)

$P_{k|k-1} = FP_{k-1|k-1}F^T + GQ_kG^T + T_k$ (EKF3)

avec $F = \frac{\partial f_{s(s_{k-1|k-1}, u_k)}}{\partial s_{k-1,k-1}}$ et $G = \frac{\partial f_{s(s_{k-1|k-1}, u_k)}}{\partial u_k}$ (EKF2)

Équations d'innovation (après mesures) :

$\tilde{y}_k = z_k - h_z(\hat{s}_{k|k-1})$ (EKF4)

$S_k = HP_{k|k-1}H^T + R_k$ (EKF5)

avec $H = \frac{\partial h_z(s_{k|k-1})}{\partial s_{k|k-1}}$ (EKF6)

Le gain de Kalman est alors : $K_k = P_{k|k-1}H^T + S_k^{-1}$ (EKF7)

Équations de mise à jour :

$\hat{s}_{k|k} = \hat{s}_{k|k-1} + K_k\hat{z}_k$ (EKF8)

$P_{(k|k)} = (I - K_{kH})P_{(k|k-1)}$ (EKF9)

Pour rappel, F et G sont les jacobiennes de la fonction f_s et H le jacobien de h_z.

L'algorithme EKF (*Extended Kalman Filter*) peut être représenté par le diagramme d'état suivant :

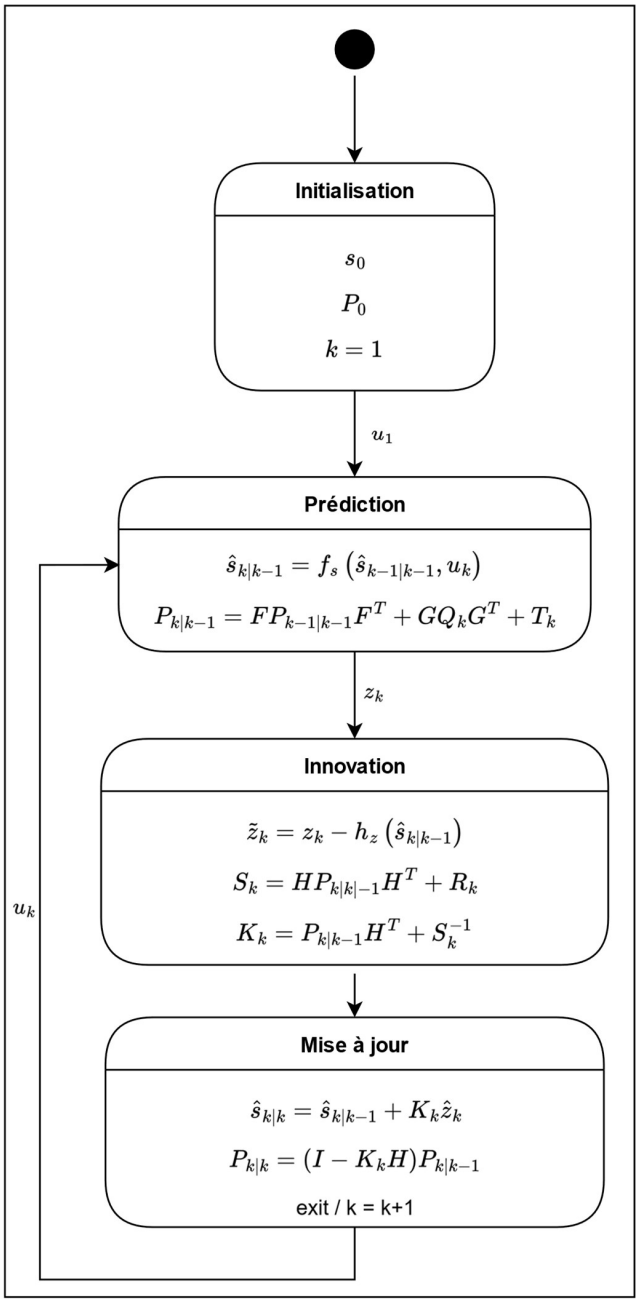


Figure 16 : diagramme d'état pour un EKF

Question 48 : Dans l'algorithme du filtre de Kalman étendu, **indiquer** ce qu'il se passe après application de la commande. **Préciser** d'où provient alors la source d'incertitude.

Le robot se déplace, une nouvelle estimation de sa pose est calculée (EKF1) avec sa variance (EKF3). L'incertitude provient du fait que la commande est bruitée.

Question 49 : **Indiquer** quelle équation donne l'innovation de ce qui devrait être mesuré et compare avec la mesure réalisée.

L'équation EKF4 donne l'innovation de mesure.

Question 50 : **Expliquer** comment la pose est finalement estimée.

La pose est estimée à partir de l'estimation de la position après commande et de la mesure réalisée. Cette mesure est pondérée par le gain de Kalman K_k pour ajuster la correction de la pose liée à la mesure.

La pose est donnée par $s = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$, la commande par $u = \begin{bmatrix} V_{longitudinale} \\ V_{laterale} \\ \omega_{robot/sol} \end{bmatrix}$

Avec $\overrightarrow{V_{C\in robot/sol}} = V_{longitudinale} \cdot \overrightarrow{x_R} + V_{laterale} \cdot \overrightarrow{y_R}$ et $\omega_{robot/sol}$ la vitesse de rotation du robot autour de $\vec{z}$.

Question 51 : En utilisant une approximation de Taylor d'ordre 1, **déterminer** $f_s(s,u)$.

$$\hat{s}_{k|k-1} \approx \hat{s}_{k-1|k-1} + dt \cdot \dot{s}_{k-1|k-1} =$$
$$\begin{Bmatrix} x_{k-1|k-1} + V_{longituginale,k-1|k-1} \cdot dt \cdot cos(\theta_{k-1|k-1} + \omega \cdot dt) - V_{laterale,k-1|k-1} \cdot dt \cdot sin(\theta_{k-1|k-1} + \omega_{k-1|k-1} \cdot dt) \\ y_{k-1|k-1} + V_{longituginale,k-1|k-1} \cdot dt \cdot sin(\theta_{k-1|k-1} + \omega \cdot dt) + V_{laterale,k-1|k-1} \cdot dt \cdot cos(\theta_{k-1|k-1} + \omega_{k-1|k-1} \cdot dt) \\ \theta_{k-1|k-1} + \omega_{k-1|k-1} \cdot dt \end{Bmatrix}$$

(cf. Q16)

Question 52 : **Calculer** les jacobiennes F et G.

$$F = \frac{\partial f_{s(s_{k-1|k-1},u_k)}}{\partial s_{k-1,k-1}} = \begin{bmatrix} \frac{\partial f_s}{\partial x} & \frac{\partial f_s}{\partial y} & \frac{\partial f_s}{\partial \theta} \\ \frac{\partial f_s}{\partial x} & \frac{\partial f_s}{\partial y} & \frac{\partial f_s}{\partial \theta} \\ \frac{\partial f_s}{\partial x} & \frac{\partial f_s}{\partial y} & \frac{\partial f_s}{\partial \theta} \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & -V_{longituginale,k-1|k-1} \cdot dt \cdot sin(\theta_{k-1|k-1} + \omega \cdot dt) + V_{laterale,k-1|k-1} \cdot dt \cdot cos(\theta_{k-1|k-1} + \omega_{k-1|k-1} \cdot dt) \\ 0 & 1 & V_{longituginale,k-1|k-1} \cdot dt \cdot cos(\theta_{k-1|k-1} + \omega \cdot dt) - V_{laterale,k-1|k-1} \cdot dt \cdot sin(\theta_{k-1|k-1} + \omega_{k-1|k-1} \cdot dt) \\ 0 & 0 & 1 \end{bmatrix}$$

$$G = \frac{\partial f_{s(s_{k-1|k-1},u_k)}}{\partial u_k} = \begin{bmatrix} \frac{\partial f_s}{\partial V_{longituginale}} & \frac{\partial f_s}{\partial V_{laterale}} & \frac{\partial f_s}{\partial \omega} \\ \frac{\partial f_s}{\partial V_{longituginale}} & \frac{\partial f_s}{\partial V_{laterale}} & \frac{\partial f_s}{\partial \omega} \\ \frac{\partial f_s}{\partial V_{longituginale}} & \frac{\partial f_s}{\partial V_{laterale}} & \frac{\partial f_s}{\partial \omega} \end{bmatrix}$$
$$= \begin{bmatrix} dt \cdot cos(\theta_{k-1|k-1} + \omega \cdot dt) & -dt \cdot sin(\theta_{k-1|k-1} + \omega \cdot dt) & -V_{longituginale,k-1|k-1} \cdot dt \cdot sin(\theta_{k-1|k-1} + \omega \cdot dt) - V_{laterale,k-1|k-1} \cdot dt \cdot cos(\theta_{k-1|k-1} + \omega_{k-1|k-1} \cdot dt) \\ dt \cdot sin(\theta_{k-1|k-1} + \omega \cdot dt) & dt \cdot cos(\theta_{k-1|k-1} + \omega \cdot dt) & V_{longituginale,k-1|k-1} \cdot dt \cdot cos(\theta_{k-1|k-1} + \omega \cdot dt) - V_{laterale,k-1|k-1} \cdot dt \cdot sin(\theta_{k-1|k-1} + \omega_{k-1|k-1} \cdot dt) \\ 0 & 0 & dt \end{bmatrix}$$

La structure *R2WLocalisationMetaState* est définie comme suit :

```
struct R2WLocalisationMetaState
{
  enum StateIndex
  {
    POSITION_X = 0,
    POSITION_Y,
    ORIENTATION_Z,
    STATE_SIZE
  };

  enum InputIndex
  {
    LINEAR_SPEED_X_BODY = 0,
    LINEAR_SPEED_Y_BODY,
    ANGULAR_SPEED_Z_BODY,
    INPUT_SIZE
  };
};
```

Question 53 : **Compléter**, sur le document réponse **DR3**, le code de la méthode *R2WLocalisationKFPredictor::predictState_()*.

```
void R2WLocalisationKFPredictor::predictState_(
  const State & previousState,
  const Input & previousInput,
  State & currentState)
{
  // Precompute some data
  // X est la pose X, U est la commande
  // MetaState : instance de R2WLocalisationKFMetaState
  x_ = previousState.X(MetaState::POSITION_X);
  y_ = previousState.X(MetaState::POSITION_Y);
  theta_ = previousState.X(MetaState::ORIENTATION_Z);
  vx_ = previousInput.U(MetaState::LINEAR_SPEED_X_BODY);
  vy_ = previousInput.U(MetaState::LINEAR_SPEED_Y_BODY);
  w_ = previousInput.U(MetaState::ANGULAR_SPEED_Z_BODY);

  vxdT_ = vx_ * dt_;
  vydT_ = vy_ * dt_;
  wdT_ = w_ * dt_;

  dT_cos_theta_wdT_ = dt_ * std::cos(theta_ + wdT_);
  dT_sin_theta_wdT_ = dt_ * std::sin(theta_ + wdT_);

  // Predict state vector
  currentState.X(MetaState::POSITION_X) = x_ + vx_ * dT_cos_theta_wdT_ - vy_ * dT_sin_theta_wdT_;
  currentState.X(MetaState::POSITION_Y) = y_ + vx_ * dT_sin_theta_wdT_ + vy_ * dT_cos_theta_wdT_;
  currentState.X(MetaState::ORIENTATION_Z) = between0And2Pi(theta_ + wdT_);

  // Predict state covariance
  jF_(
    MetaState::POSITION_X,
    MetaState::POSITION_X) = 1;
  jF_(
    MetaState::POSITION_X,
    MetaState::ORIENTATION_Z) = -vx_ * dT_sin_theta_wdT_ - vy_ * dT_cos_theta_wdT_;
  jF_(
    MetaState::POSITION_Y,
    MetaState::POSITION_Y) = 1;
  jF_(
    MetaState::POSITION_Y,
    MetaState::ORIENTATION_Z) = vx_ * dT_cos_theta_wdT_ - vy_ * dT_sin_theta_wdT_;
  jF_(
    MetaState::ORIENTATION_Z,
    MetaState::ORIENTATION_Z) = 1;


  jG_(
    MetaState::POSITION_X,
    MetaState::LINEAR_SPEED_X_BODY) = dT_cos_theta_wdT_;
  jG_(
    MetaState::POSITION_Y,
    MetaState::LINEAR_SPEED_X_BODY) = dT_sin_theta_wdT_;
  jG_(
    MetaState::POSITION_X,
    MetaState::LINEAR_SPEED_Y_BODY) = -dT_sin_theta_wdT_;
  jG_(
    MetaState::POSITION_Y,
    MetaState::LINEAR_SPEED_Y_BODY) = dT_cos_theta_wdT_;

  jG_(
    MetaState::POSITION_X,
    MetaState::ANGULAR_SPEED_Z_BODY) = -vxdT_ * dT_sin_theta_wdT_ - vydT_ * dT_cos_theta_wdT_;
  jG_(
    MetaState::POSITION_Y,
    MetaState::ANGULAR_SPEED_Z_BODY) = vxdT_ * dT_cos_theta_wdT_ - vydT_ * dT_sin_theta_wdT_;
  jG_(
    MetaState::ORIENTATION_Z,
    MetaState::ANGULAR_SPEED_Z_BODY) = dt_;

  currentState.P().noalias() = jF_ * previousState.P() * jF_.transpose();
  currentState.P().noalias() += jG_ * previousInput.QU() * jG_.transpose();
```

```
}
```

Question 54 : **Indiquer** comment la fusion de capteurs peut être prise en compte avec un filtre EKF.

À chaque mesure d'un capteur, la mise à jour est réalisée avec la fonction de mesure du capteur considéré, ainsi tous les capteurs sont pris en compte, la précision augmente.

Partie 4 : Base de données

Cette partie traite de la sauvegarde des trajets dans une base de données et des aspects sécuritaires liés à son utilisation.

L'utilisation des robots sur des trajets réguliers et identiques est un atout majeur pour enrichir un Système d'Information Géographique et apporter à l'agriculteur un historique qui lui permette une analyse et une optimisation de ses ressources.

Les machines agricoles utilisent classiquement en interne le protocole de communication ISOBUS qui permet de faire communiquer ses différents sous-systèmes de manière bidirectionnelle sur un même bus de donnée, le bus de données CAN et des connecteurs normalisés. L'ISOBUS permet ainsi faire communiquer en interne au système, le tracteur, les outils (capteurs et actionneurs) et une console. Ces données peuvent aussi être extraites du système de manière automatique pour une collecte et analyse.

Dans le cadre de ce sujet, le fonctionnement des collectes de données ne sera pas étudié.

Actuellement, l'équipe ROMEA fournit au robot les trajectoires à suivre sous forme de fichiers JSON. Ce fichier indique les coordonnées (longitude et latitude exprimées dans le système géodésique WGS84[5]) du point de départ O de la trajectoire puis les coordonnées des points à suivre (*waypoints*) par lesquels le robot doit passer ainsi que la vitesse à laquelle il doit y passer.

Afin d'enrichir les modèles, il est décidé de mettre en place une base de données spécifique de collecte qui aura aussi la possibilité d'enrichir la plate-forme de doubles numériques des robots. Les données à gérer sont :

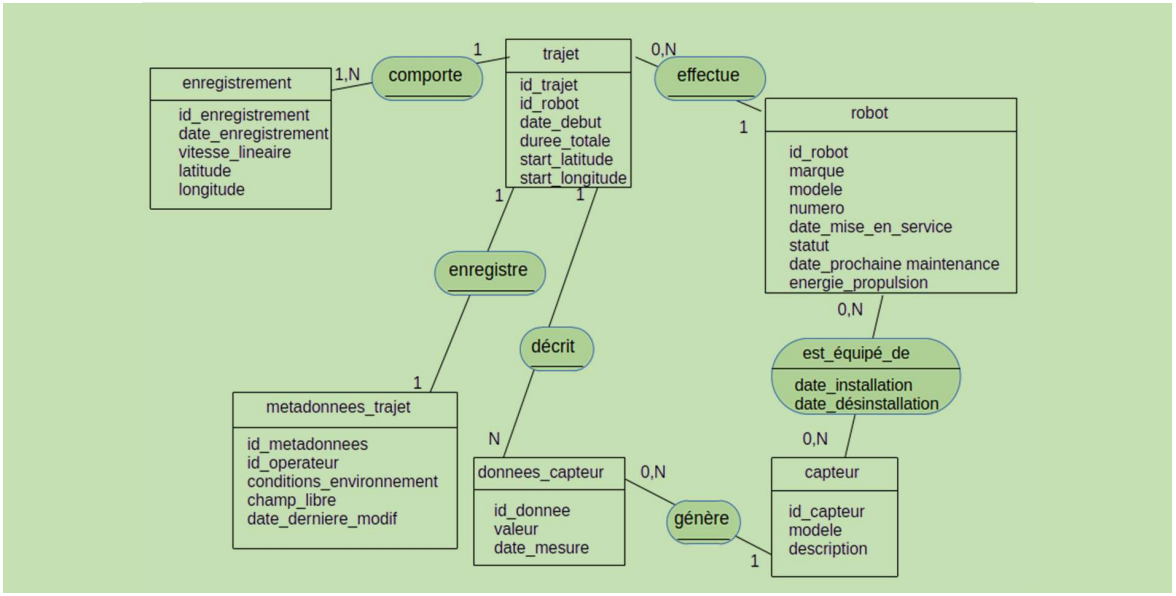
- chaque trajet d'un robot est enregistré sous forme d'une séquence de *waypoints* et est associé à un seul robot. Un trajet est caractérisé par un numéro unique de trajet, une date et une heure de début de trajet, la durée totale du trajet enregistré ainsi que les coordonnées (longitude et latitude) du point de départ. Pour chaque *waypoint* est indiqué : la date/heure de mesure/enregistrement, la longitude, la latitude et la vitesse linéaire $V_{O_R \in robot/sol}$ du robot ;
- à chaque trajet sont associées des métadonnées obligatoires : date et heure du trajet, l'identifiant de l'opérateur (la gestion des utilisateurs est hors périmètre de cette question), conditions environnementales (liste prédéfinie : *pluie*, *soleil*, *neige*) complétées par un champ libre. La date de dernière modification est stockée aussi, ce qui permet d'indiquer le dernier changement ou remarque entré par l'opérateur ;
- les robots sont caractérisés par un identifiant unique (immuable), une marque, un modèle dans cette marque, un numéro unique (inchangé toute sa vie), une date de mise en service, un statut reflétant sont état de fonctionnement (liste fixe : *actif*, *en maintenance*, *hors service*), sa prochaine date de maintenance prévue, son énergie de propulsion (liste fixe : *électrique*, *solaire*, *hybride*, *essence*, *gasoil agricole*) ;
- chaque robot a la capacité d'être équipé en standard de dix capteurs environnementaux standardisés, communiquant sur ISOBUS, et pouvant varier d'un robot à l'autre. Chaque capteur est caractérisé par un identifiant unique, un modèle et une description. Les données

[5] Le WGS 84 (*World Geodetic System 1984*) est notamment le système géodésique associé au système de positionnement par satellite GPS.

renvoyées par les capteurs sont des valeurs numériques de type *FLOAT* et doivent être enregistrées régulièrement lors de chaque trajet ;
- les capteurs peuvent être installés sur un robot ou un autre selon les objectifs des trajets. Pour permettre une traçabilité il est souhaitable de stocker l'historique des installations et désinstallations de chaque capteur sur les robots.

Quelques règles de gestion doivent être prises en compte pour cette base de données :
- le nombre de capteurs et d'actionneurs peut évoluer (structure flexible) ;
- les modifications des métadonnées doivent être traçables (date de dernière modification) ;
- un trajet est toujours associé à un seul robot, et un état à un seul trajet ;
- les opérateurs sont identifiés par un simple champ texte, leur gestion et leur authentification étant gérées par ailleurs.

Question 55 : **Écrire** la modélisation conceptuelle des données la plus adaptée à ce besoin, formalisée en un diagramme entité-relation (Entity-Relationship Diagram).



Question 56 : **Indiquer** comment sera traduite une cardinalité N,N dans une implémentation dans une base de données relationnelle classique (de type MySQL par exemple).

Par une table pivot contenant deux clefs étrangères

Cette partie de gestion des données sera intégrée dans un système de base de données relationnelle plus vaste géré par un fournisseur de logiciel qui va apporter une interface d'utilisation et d'analyse adaptée au métier et permettre une exploitation distante. Cette application offre un accès sécurisé qui utilise deux méthodes d'authentification :
- par login et mot de passe pour les utilisateurs humains ;
- par clef ssh pour les connexions machines.

Voici le MLD associé à cette partie, dans le contexte d'une base de données MySQL.

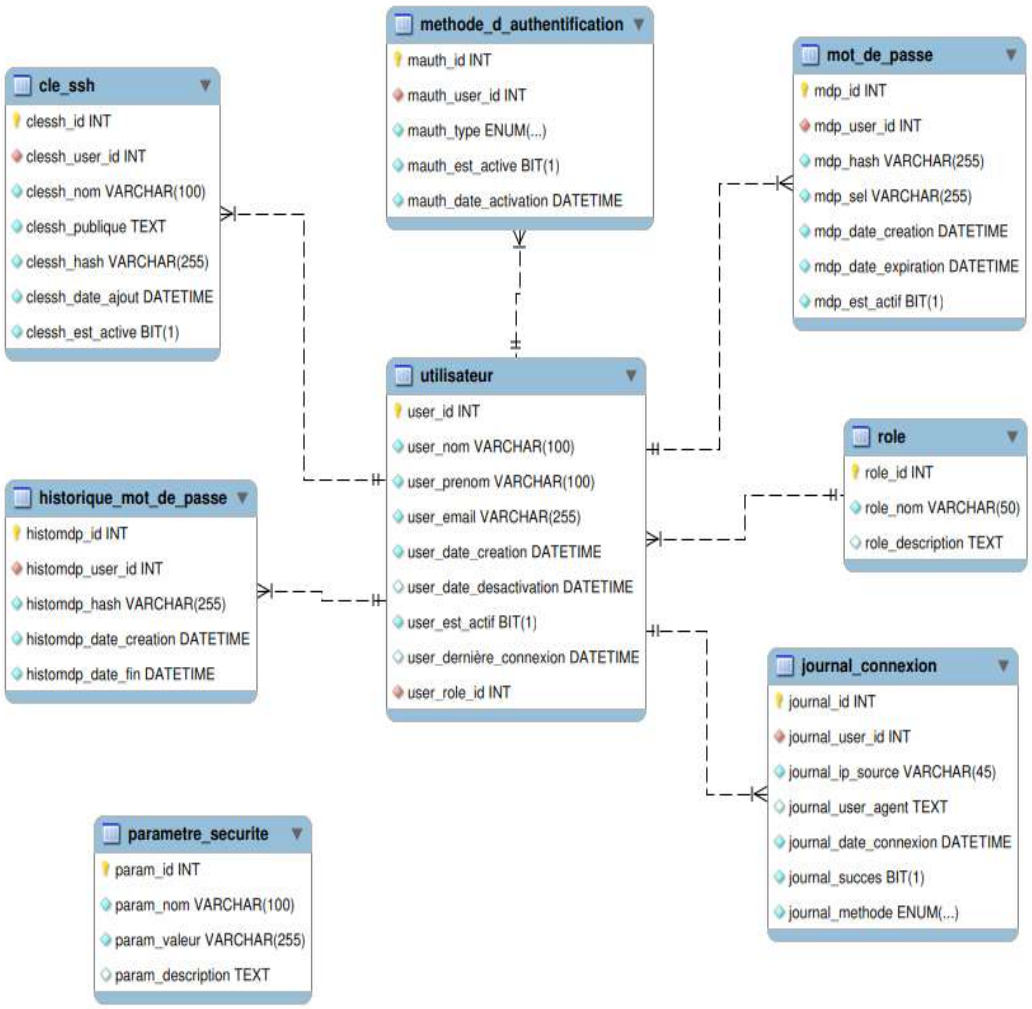


Figure 17 : Modèle Logique des Données (MLD)

Question 57 : **Écrire** le code SQL de création de la table *mot_de_passe.*

```
CREATE TABLE mot_de_passe (
    mdp_id INT AUTO_INCREMENT PRIMARY KEY,
    mdp_user_id INT ,
    mdp_hash VARCHAR(255) ,
    mdp_sel VARCHAR(255) ,
    mdp_date_creation DATETIME ,
    mdp_date_expiration DATETIME ,
    mdp_est_actif BIT(1)  ,
    FOREIGN KEY (mdp_user_id) REFERENCES utilisateur(user_id)
)
```

Question 58 : **Expliquer** pourquoi ce modèle de données ne permet pas de renvoyer son mot de passe à un utilisateur qui l'aurait oublié.

Le mot de passe en clair n'est pas stocké dans la base de données. Seule est stockée son empreinte (*mdp_hash*) générée par une fonction de hash : cette fonction est par définition quasi irréversible.

Question 59 : **Indiquer** si cela est techniquement conforme au Règlement Général sur la Protection des Données (voir **DT14**). **Justifier**.

Oui, la RGPD impose de protéger les données personnelles et les mots de passe sont des données personnelles. Les stocker en clair est une faute grave régulièrement sanctionnée par la CNIL. Il est donc obligatoire de ne stocker que les empreintes (*mdp_hash*).

La table mot_de_passe comporte une colonne mdp_sel pour stocker un sel (*salt*) associé au mot de passe.

> **Question 60 :** **Justifier** l'intérêt du sel (*salt*) dans le cas d'usage de cette base de données.

On hache (le mot de passe + mdp_sel), le sel renforce l'unicité du hash.

Le sel permet de se protéger des attaques par force brut/dictionnaire/ rainbow table : il faut découvrir non seulement le mot de passe mais aussi le sel … qui n'est pas connu de l'utilisateur.

> **Question 61 :** **Préciser** si le modèle de données permet de stocker plusieurs empreintes de mot de passe (*mdp_hash*) identiques.

Oui, ce n'est pas une clef primaire, ni un support de clef étrangère, rien n'interdit a priori les doublons sur ce champ.

Il est souhaité que le journal de connexion soit renseigné de manière automatisée par la base de données elle-même et non pas par un développement applicatif externe.

> **Question 62 :** **Indiquer** quelle fonction interne de la base de données permet de mettre en place cette action.

Un trigger déclenché à la connexion utilisateur.

Dans la table « *parametres_securite* », figure la longueur minimale requise pour les mots de passe. La CNIL recommande depuis 2022 une entropie minimale de 80 bits pour les mots de passe de connexion distante. L'entropie est calculée selon la formule de Shannon :

$$H = \log_2 N^L$$

- H : entropie de Shannon en bits ;
- L : nombre de symboles composant le mot de passe ;
- N : nombre de symboles possibles.

> **Question 63 :** **Indiquer** quelle est la longueur associée pour un mot de passe de 80 bits d'entropie dans un environnement US ASCII (voir **DT15**). **Justifier**.

Table US ASCII : 94 caractères imprimables

Selon Shannon Entropie = L × log2(R)

L=E/log2(R)L=80/log2(96)=12

Afin d'alimenter un système d'alerte, il est demandé d'élaborer une vue *VIEW_tentative_inactif*. Cette vue permet d'afficher le numéro d'utilisateur (*user_id*) le nom d'utilisateur et le nombre de connexions en erreur (*jounal_success* à 1) durant les dernières 24h pour les utilisateurs dont le mot de passe est inactivé (*mdp_est_actif* à 1).

> **Question 64 :** **Écrire** la requête SQL de création de cette vue.

```
create view from VIEW_tentative_inactifs
as select utilisateur.user_id, utilisateur.user_nom , count(journal.journal_user_id)
where utilisateur.user_id=jounal.journal_user_id
and utilisateur.user_id=mot_de_passe.mdp_user_id
and journal_date_connexion<now-24h
and mdp_est_actif=1
and journal_success=1
```

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

I 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.

II Sujet

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

https://sti.eduscol.education.fr/sites/eduscol.education.fr.sti/files/concours-examens/19131/19131-agr-gation-externe-section-sciences-industrielles-de-ing-nieur-option-sii-et-ing-nierie.pdf

Le travail demandé est divisé en **trois parties indépendantes**, formant un total de **47 questions**. Dans les éléments de correction, les intitulés des questions sont rappelés mais il faut se reporter au sujet complet pour avoir connaissance complète du contexte. Toutes les références indiquées (numéros de figure, etc.) sont celles du sujet, et non pas du présent rapport.

III Éléments de correction

1 – Technologie des compteurs optiques de particules

1.1 Caractérisation de l'onde diffusée par une particule

Q1 En s'aidant de l'annexe A.1, donner l'expression littérale de la fonction d'onde incidente $\underline{\boldsymbol{E}}_{\text{inc}}$. Préciser aussi l'expression de son nombre d'onde k et de sa pulsation ω.

Les données sont conformes à celles de l'annexe A.1 avec $\psi = 0$ donc :

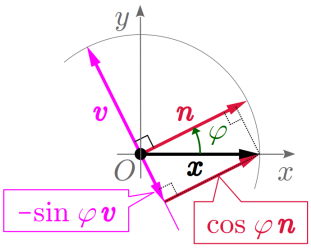
$$\underline{\boldsymbol{E}}_{\text{inc}}(z,t) = E_0\, \mathrm{e}^{\imath(kz-\omega t)}\, \boldsymbol{x} \quad \text{avec} \quad k = \frac{2\pi}{\lambda} \quad \text{(nombre d'onde)} \quad \text{et} \quad \omega = \frac{2\pi c}{\lambda} \quad \text{(pulsation)}$$

Q2 En illustrant le raisonnement par une figure, montrer que la fonction d'onde incidente $\underline{\boldsymbol{E}}_{\text{inc}}$ peut s'exprimer dans la base vectorielle $(\boldsymbol{n}, \boldsymbol{v})$ du plan $(O, \boldsymbol{x}, \boldsymbol{y})$ comme ci-dessous :

$$\underline{\boldsymbol{E}}_{\text{inc}}(z,t) = E_0\, e^{\imath(kz-\omega t)} \begin{pmatrix} \cos\varphi \\ -\sin\varphi \end{pmatrix}_{(\boldsymbol{n},\boldsymbol{v})} \tag{1}$$

Puisque $\underline{\boldsymbol{E}}_{\text{inc}}(z,t) = E_0\, e^{\imath(kz-\omega t)}\, \boldsymbol{x}$, il suffit d'exprimer $\boldsymbol{x}$ dans la base $(\boldsymbol{n}, \boldsymbol{v})$:

$$\begin{aligned} \boldsymbol{x} &= (\boldsymbol{x}\cdot\boldsymbol{n})\,\boldsymbol{n} &+ (\boldsymbol{x}\cdot\boldsymbol{v})\,\boldsymbol{v} \\ &= \cos(\widehat{\boldsymbol{x},\boldsymbol{n}})\,\boldsymbol{n} &+ \cos(\widehat{\boldsymbol{x},\boldsymbol{v}})\,\boldsymbol{v} \\ &= \cos\varphi\,\boldsymbol{n} &+ \cos\left(\frac{\pi}{2}+\varphi\right)\boldsymbol{v} \\ &= \cos\varphi\,\boldsymbol{n} &- \sin\varphi\,\boldsymbol{v} \end{aligned} \quad \square$$



Q3 Montrer que l'expression de la fonction d'onde diffusée $\underline{\boldsymbol{E}}_{\text{sca}}$ dans la base $\boldsymbol{u}, \boldsymbol{v}$ est indépendante de la coordonnée spatiale z. Expliquer à quoi correspond le facteur restant qui est fonction des coordonnées de propagation r et t.

Des équations (1) et (2), par substitution de $\underline{\boldsymbol{E}}_{\text{inc}}$, on tire :

$$\underline{\boldsymbol{E}}_{\text{sca}\,(\boldsymbol{u},\boldsymbol{v})}(r,\theta,\varphi,t) = E_0\,\frac{e^{\imath k(r-z)}\,e^{\imath(kz-\omega t)}}{-\imath kr}\,\underline{\mathbf{S}}(\theta)\cdot\begin{pmatrix} \cos\varphi \\ -\sin\varphi \end{pmatrix}_{(\boldsymbol{n},\boldsymbol{v})}$$

Dans la fraction, le numérateur se factorise en $e^{\imath(kr-kz+kz-\omega t)}$ qui se simplifie en $e^{\imath(kr-\omega t)}$. Dès lors, il n'y a plus d'occurrence de z dans l'expression de $\underline{\boldsymbol{E}}_{\text{sca}}$ $\square$.

Le facteur $\dfrac{e^{\imath(kr-\omega t)}}{r}$ correspondent à la propagation d'une **onde sphérique** .

Q4 Calculer dans le pire cas la distance d'observation à partir de laquelle le modèle de la fonction d'onde diffusée $\underline{\boldsymbol{E}}_{\text{sca}}$ serait a priori valide. Conclure.

Comme pour une onde sphérique, le modèle n'est a priori valide que dans le champ lointain, c'est-à-dire pour $r \gg 2d^2/\lambda$ (distance de Fraunhofer d_F – cf. annexe A.1) où d est le diamètre de la particule. On fait l'application numérique pour le plus grand diamètre de particule $d = 10\,\mu\text{m} = 10^{-5}\,\text{m}$:

$$d_F \approx \frac{2\times(10^{-5})^2}{650\times10^{-9}} = \frac{10^{-1}}{325} \approx 0{,}3\,\text{mm}$$

Cette valeur est satisfaisante pour la suite de l'étude car la cellule photosensible est située largement plus loin (au minimum à quelques mm de la chambre de diffusion).

Q5 En faisant abstraction de la matrice $\underline{\mathbf{S}}$, calculer l'ordre de grandeur du rapport d'amplitudes $\mathcal{H} = |\underline{\boldsymbol{E}}_{\text{sca}}| \,/\, |\underline{\boldsymbol{E}}_{\text{inc}}|$ de l'onde diffusée comparée à l'onde incidente à une distance $r = 5\,\text{mm}$ de la particule (typiquement là où est située la cellule de l'OPC). Conclure brièvement.

En prenant $\underline{\mathbf{S}} = [\mathbf{1}]$, on a : $\mathcal{H} \approx \dfrac{E_0}{kr}/E_0 = \dfrac{1}{kr} = \dfrac{\lambda}{2\pi r}$ car $k = \dfrac{2\pi}{\lambda}$

AN : $\mathcal{H} \approx \dfrac{650 \times 10^{-9}}{2 \times 3{,}14 \times 5 \times 10^{-3}} \approx 2{,}07 \times 10^{-5}$

Cette valeur est assez faible mais pas insignifiante. Une technologie adaptée est donc requise pour détecter la lumière diffusée (source de lumière puissante, photodiode à haute sensibilité, chaîne d'amplification performante).

1.2 Modélisation du mesurande d'un OPC

Q6 Avec toutes les hypothèses précédentes, calculer dans le pire cas le nombre N de coefficients de Mie nécessaires pour modéliser l'éclairement reçu par le capteur. Conclure brièvement.

On calcule le paramètre de taille x dans le cas du plus grand diamètre de particule, c'est-à-dire $d = 10\,\mu\text{m} = 10^{-5}\,\text{m}$, sachant qu'on a par ailleurs $\lambda = 650\,\text{nm} = 6{,}5 \times 10^{-7}\,\text{m}$:

$$x = \frac{\pi \times 10^{-5}}{6{,}5 \times 10^{-7}} \approx \frac{3{,}14}{6,5} \times 10^2 \approx 48{,}3$$

On en déduit une valeur approchée de $N \approx 48{,}3 + 4 \times 48{,}3^{1/3} + 2 \approx 65$

Ce nombre est tout à fait raisonnable, compte tenu des moyens informatiques actuels.

Q7 À l'aide des annexes A.2 et A.3, finaliser le codage des lignes n$^{\text{os}}$ 28 & 29 (champs marqués « `???` ») et à partir de la ligne n$^{\text{o}}$ 34, afin d'obtenir un affichage formaté des valeurs de Q_{sca}, σ_{sca}, Q_{abs} et σ_{abs} à 10^{-2} comme ci-dessous.

```
Size parameter x = 12.08   particle cross section s = 4.91 um^2
    MATERIAL    Qsca    Ssca    Qabs    Sabs
      quartz    2.93   14.40    0.00    0.00
 diesel soot    1.20    5.88    1.15    5.64
 iron oxydes    1.36    6.69    0.98    4.81
```

```
28  x = np.pi * d / wl
29  s = np.pi * (d / 2E3)**2

34  for material in materials:
35    qext, qsca, _, _ = mie.efficiencies(material['cri'], d, wl)
36    qabs = qext - qsca
37    print(f"{material['name']:>12}{qsca:8.2f}{qsca * s:8.2f}"
38          f"{qabs:8.2f}{qabs * s:8.2f}")
```

Q8 En procédant par comparaisons et en s'appuyant sur des connaissances générales, pour chacun des trois matériaux étudiés, commenter brièvement les résultats numériques obtenus dans l'affichage donné à la question précédente.

- Le **quartz** n'est pas absorbant (CRI réel) donc il est logique que son Q_{abs} (et donc σ_{abs}) soient nuls. Intuitivement, c'est aussi le matériau le plus « brillant » des trois donc il n'est pas surprenant qu'il ait la plus grande valeur de Q_{sca}.

- La **suie** est plus absorbante que les oxydes de fer car elle réfléchit moins la lumière et conduit bien la chaleur (étant essentiellement composée de carbone). Il est donc normal que ce soit le matériau avec la plus petite valeur de Q_{sca} dans l'échantillon.

- Les **oxydes de fer** diffractent davantage la lumière que les autres matériaux (partie réelle du CRI élevée) mais en absorbent une part non négligeable (partie imaginaire non nulle). Leur valeur de Q_{sca} se trouve donc logiquement entre celles des deux autres matériaux.

Q9 Toujours à l'aide des annexes A.2 et A.3, compléter les lignes n$^{\text{os}}$ 45, 46 et 49 (champs marqués « `???` ») pour coder les listes de valeurs représentant respectivement les grandeurs θ, μ et p calculées pour $\theta \in [0;\pi]$ avec une résolution de 0,2°.

```
45  theta_range = np.linspace(0, np.pi, 901) # 180 x 5 + 1
46  mu_range    = np.cos(theta_range)

49    p_range = (np.abs(s1_range)**2 + np.abs(s2_range)**2) / 2
```

Q10 Compléter les lignes n$^{\text{os}}$ 43 & 44, 50 & 51, 53 & 54 pour calculer, en puissances entières de 10, les valeurs optimales des bornes respectivement haute et basse de l'échelle radiale logarithmique du diagramme, en fonction des valeurs minimales et maximales mémorisées de p.

```
43  p_max = 0       # lowest initial value
44  p_min = np.inf  # highest initial value

50    p_min   = p_range.min() if p_range.min() < p_min else p_min
51    p_max   = p_range.max() if p_range.max() > p_max else p_max

53  ax.set_rmin(10**(np.floor(np.log10(p_min)-1)))
54  ax.set_rmax(10**(np.floor(np.log10(p_max)+1)))
```

Q11 En termes de proportions relatives, sans distinction de matériaux, commenter brièvement l'allure globale des courbes de la figure 10 selon la valeur de l'angle polaire θ.

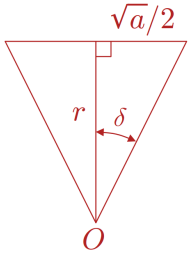
Pour les trois matériaux étudiés, la fonction de phase vaut approximativement :
- entre 2 et 10 lorsque $\theta \in [0° ; 10°]$;
- entre 0,1 et 0,005 – en moyenne, environ 50 à 100 fois moins – lorsque $\theta \in [30°; 180°]$.

Remarque (non attendue par les candidats). De façon surprenante, la diffusion est donc très largement prépondérante en $z+$, c'est-à-dire dans le sens de propagation de l'onde incidente, alors qu'il ne s'agit pas de l'onde incidente (ce phénomène peut notamment s'expliquer par des interférences constructives/destructives entre les photons diffractés/réfléchis par la particule).

Q12 En illustrant le calcul par une figure plane, déterminer au degré près les valeurs des bornes du cône $[\theta_1 ; \theta_2] \times [\varphi_1 ; \varphi_2]$ centré sur la direction $(90°, 0°)$ formant l'angle solide Ω dans lequel la photodiode principale « voit » la lumière diffusée par une particule éclairée en O.

L'angle solide Ω est celui d'une pyramide à base carrée d'aire a, donc de côté $\sqrt{a}$, et de hauteur r. Conformément à la figure ci-contre, le demi-angle au sommet δ de la pyramide vaut :

$$\delta = \arctan(\sqrt{a}/2r) = \arctan(\sqrt{7,5}/9) \approx 17°$$



On a alors $(\theta_1, \theta_2) = (90° - \delta, 90° + \delta) \approx (73°, 107°)$ et $(\varphi_1, \varphi_2) \approx (-17°, +17°)$.

Q13 En considérant `delta_deg` comme une donnée déjà déclarée représentant la mesure en degrés du demi-angle au sommet du cône pyramidal constituant Ω, compléter le code des lignes du programme ci-dessous (champs marqués « `???` ») pour déterminer par des expressions les valeurs des variables représentant ses bornes (θ_1, θ_2) ainsi que la différence $\varphi_2 - \varphi_1$. Quant à la liste `theta_range` des valeurs de θ en radians, elle est à définir avec une résolution de 0,2°.

```
93  delta           = ???  # in rad
94  theta1, theta2  = ???  # also in rad
95  phi_width       = ???  # phi_2 - phi_1
96  theta_range     = ???  # [theta1; theta2] with a 0.2 deg resolution
97  mu_range        = ???  # u = cos(theta)
```

Pas de difficulté ici sauf à la ligne 96 : pour une résolution à 0,2°, il faut $2 \times \delta \times 5 + 1$ donc $\delta \times 10 + 1$ valeurs.

```
93  delta           = np.radians(delta_deg)
94  theta1, theta2  = np.pi / 2 - delta, np.pi / 2 + delta
95  phi_width       = np.radians(delta_deg * 2)
96  theta_range     = np.linspace(theta1, theta2, delta_deg * 10 + 1)
97  mu_range        = np.cos(theta_range)
```

Q14 Compte tenu des spécifications qui viennent d'être décrites, en exploitant les éléments de code pertinents et les résultats des questions précédentes, compléter toutes les lignes de code du programme ci-dessus comportant un champ marqué « ??? ».

```
101  d_range = np.logspace(2, 4, 1001)  # 10^2 to 10^4 nm (500 values / decade)

105      x = np.pi * d / wl              # cf. Q7
106      s = np.pi * (d / 2E3)**2        # cf. Q7
107      qext, qsca, _, _   = mie.efficiencies(material['cri'], d, wl)       # cf. Q7
108      s1_range, s2_range = mie.S1_S2(material['cri'], x, mu_range, 'one') # cf. Q9

110    wp_range = np.sin(theta_range) * (np.abs(s1_range)**2 + np.abs(s2_range)**2) / 2

112    wp_integral = spi.simpson(wp_range, theta_range) * phi_width * qsca * s

114    ax2.plot(d_range / 1E3,   # in um
115             mesurand_range,
...           )
```

Q15 Dans une perspective de modélisation pour exploitation opérationnelle dans le micrologiciel embarqué d'un OPC, commenter les courbes de la figure 12, notamment en termes de linéarité, de monotonie et de positions relatives. Conclure.

- Les courbes ne sont globalement ni linéaires, en particulier pour $d < 1\,\mu\text{m}$, ni monotones (il est assez difficile de déterminer le diamètre d'une particule de l'intervalle $[0{,}3\,;\ 1]\,\mu\text{m}$ connaissant $E_{\text{e sca}}$).

- De plus, les courbes n'ont pas de positions relatives directement déterminées par l'indice de refraction du matériau, puisque la courbe du quartz (qui a l'indice le plus faible) est la plus basse pour $d < 0{,}4\,\mu\text{m}$ mais la plus haute pour $d > 0{,}8\,\mu\text{m}$.

Elles sont donc difficiles à modéliser pour une exploitation opérationnelle.

1.3 Principe de mesure d'un OPC

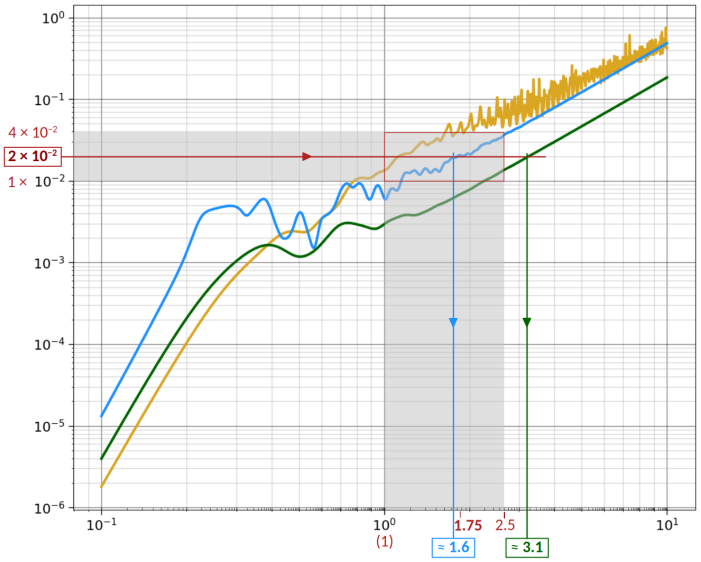
Q16 En exploitant la figure 12 et les explications ci-dessus, pour la valeur particulière du mesurande $E_{\text{e sca}}(\Omega) = 2 \times 10^{-2}$, déterminer en cm^3 : **a)** le volume V_{c} qui serait calculé par l'OPC, **b)** le volume théorique V_{th1} s'il s'agissait d'une particule sphérique d'oxydes de fer, **c)** le volume théorique V_{th2} s'il s'agissait d'une particule sphérique de suie de diesel.

On a $E_{\text{e sca}}(\Omega) = 2 \times 10^{-2} \in [1 \times 10^{-2}\,;\ 4 \times 10^{-2}[$ qui correspond à l'intervalle de diamètres $D_3 = [1\,;\ 2{,}5[\,\mu\text{m}$ dont la médiane vaut $d_3 = (1 + 2{,}5)/2 = 1{,}75\,\mu\text{m}$. On a donc :

$$V_{\text{c}} = \frac{\pi d^3}{6} \approx \frac{3{,}14 \times 1{,}75^3}{6} \approx 2{,}806\,\mu\text{m}^3 \approx 2{,}81 \times 10^{-12}\,\text{cm}^3 \qquad (1\,\mu\text{m} = 1 \times 10^{-4}\,\text{cm})$$

De plus, sur la figure 12, la droite horizontale $E_{\text{e sca}}(\Omega) = 2 \times 10^{-2}$ intersecte les courbes de mesurande des particules d'oxydes de fer et de suie, respectivement aux valeurs de diamètres $d_{\text{th1}} \approx 1{,}6\,\mu\text{m}$ (voire $1{,}7$) et $d_{\text{th2}} \approx 3{,}1\,\mu\text{m}$ (voire $3{,}0$). Après calcul, on obtient :

$$V_{\text{th1}} \approx 2{,}14 \times 10^{-12}\,\text{cm}^3 \quad (2{,}57)$$
$$V_{\text{th2}} \approx 15{,}6 \times 10^{-12}\,\text{cm}^3 \quad (14{,}1)$$



Q17 À partir des valeurs des volumes V_{c}, V_{th1} et V_{th2} déterminées à la question précédente, calculer respectivement en µg les masses m_{c}, m_{th1} et m_{th2} de particules correspondantes. Pour les deux dernières, en déduire à chaque fois l'erreur relative théorique de mesure de l'OPC.

Les calculs de masse doivent intégrer le facteur multiplicatif 10^6 de conversion des g en µg :

$$\begin{aligned} m_{\text{c}} &= \rho_{\text{m}} \cdot V_{\text{c}} \approx 1{,}65 \times 2{,}81 \times 10^{-12}\,\text{g} \approx 4{,}64 \times 10^{-6}\,\mu\text{g} \\ m_{\text{th1}} &= \rho_1 \cdot V_{\text{th1}} \approx 2{,}20 \times 2{,}14 \times 10^{-12}\,\text{g} \approx 4{,}72 \times 10^{-6}\,\mu\text{g} \\ m_{\text{th2}} &= \rho_2 \cdot V_{\text{th2}} \approx 1{,}20 \times 15{,}6 \times 10^{-12}\,\text{g} \approx 18{,}7 \times 10^{-6}\,\mu\text{g} \end{aligned}$$

Les calculs d'erreur relative doivent être opérés en prenant les valeurs m_{th1} et m_{th2} comme référence, puisque c'est la valeur m_{c} qui est comptabilisée par l'OPC sur la base d'un diamètre moyen et non pas en fonction du matériau (qui reste inconnu) :

$$\begin{aligned} \varepsilon_{\text{th1}} &= \frac{m_{\text{c}} - m_{\text{th1}}}{m_{\text{th1}}} \approx \frac{4{,}64 - 4{,}72}{4{,}72} \approx -2\,\% \\ \varepsilon_{\text{th2}} &= \frac{m_{\text{c}} - m_{\text{th2}}}{m_{\text{th2}}} \approx \frac{4{,}64 - 18{,}7}{18{,}7} \approx -75\,\% \end{aligned}$$

Q18 Conclure quant à l'aptitude de l'OPC à fournir des valeurs de concentration massique réalistes sur la base du principe de mesure qui est exposé dans cette étude. Proposer éventuellement des solutions qui permettraient d'améliorer la précision de l'appareil.

La valeur d'erreur relative ε_{th1} peut paraître satisfaisante mais elle est due à une opportune compensation de facteurs. Quant à la valeur ε_{th2}, elle peut paraître décevante mais :
a) il s'agit d'une erreur théorique calculée par rapport à des valeurs de référence qui restent

hypothétiques ; en particulier, les volumes V_{th1} et V_{th2} sont certainement surestimés car calculés par une formule sphérique que ne peut pas complètement compenser la notion de masse volumique apparente ;
b) d'autre part, ces erreurs doivent être remises en perspective dans le contexte général moyen de pollution tel que décrit en figure 4 ; des particules de densité plus faible ou d'indice de réfraction plus proche du celui du matériau d'étalonnage vont venir contrebalancer la sous-estimation massique ici constatée.

En revanche, l'étalonnage, lui, est une procédure fiable qui repose sur la réponse directe de la photodiode, sans recours à la théorie de Mie. On peut donc envisager ensuite une phase de tests de validation avec d'autres types d'aérosols afin d'ajuster par des coefficients correcteurs les formules de calcul de concentrations massiques en fonction des résultats obtenus.

2 – Calibrage de capteurs électrochimiques de NO_2

2.1 Principe du capteur et récupération des données

Q19 À l'aide de la documentation constructeur, donner l'ordre de grandeur ainsi que l'unité de la sensibilité du capteur. Que peut-on déduire de la mesure réalisée et de sa reproductibilité ?

Dans la documentation technique, la sensibilité S (en nA/ppm) est spécifiée par $-650 \leq S \leq -200$. La plage de variation est donc grande et il n'est pas possible de savoir exactement la valeur numérique de la sensibilité d'un capteur en particulier. Il est donc apparent qu'il faut calibrer chaque capteur par rapport à une mesure de référence, la valeur brute n'est sinon que proportionnelle à la variation de concentration de NO_2.

Les courants sur les électrodes auxiliaires (A) et de travail (W) peuvent se décomposer en un courant d'*offset* (annotée 0) et une partie variable dans le temps (annotée E). On peut donc noter :

$$\begin{cases} I_A = I_{A0} + I_{AE} \\ I_W = I_{W0} + I_{WE} \end{cases}$$

Cette même notation (*offset* plus partie variable) peut être reprise pour d'autres quantités.

Q20 À partir du schéma présenté en figure 14 c, montrer que la concentration en NO_2 est une fonction linéaire de la différence de potentiel entre les électrodes notée V_D. Expliciter le coefficient directeur et l'offset.

En utilisant la loi d'Ohm, on a :

$$\begin{cases} V_A = R_L I_A = R_L \left(I_{A0} + I_{AE}\right) = V_{A0} + V_{AE} \\ V_W = R_L I_W = R_L \left(I_{W0} + I_{WE}\right) = V_{W0} + V_{WE} \end{cases}$$

et le capteur mesure

$$[\mathrm{NO_2}] = \frac{I_W - I_A}{S}$$

On a donc par substitution :

$$[\mathrm{NO_2}] = \frac{(V_{W0} + V_{WE}) - (V_{A0} + V_{AE})}{R_L S} = (V_{WE} - V_{AE}) \cdot \frac{1}{R_L S} + \frac{(V_{W0} - V_{A0})}{R_L S}$$

ou encore de manière plus compacte :

$$[\mathrm{NO_2}] = V_{DE} \cdot \frac{1}{R_L S} + \frac{V_{D0}}{R_L S}$$

Q21 En déduire le lien entre la concentration en $\mathrm{NO_2}$ et les tensions des électrodes W et A. Quelle est la conséquence pour le calibrage de la mesure ?

Dans ce cas :

$$[\mathrm{NO_2}] = \frac{I_W}{S_W} - \frac{I_A}{S_A}$$

Par substitution et par raisonnement similaire on obtient :

$$[\mathrm{NO_2}] = \frac{V_{W0} + V_{WE}}{R_L S_W} - \frac{V_{A0} + V_{AE}}{R_L S_A}$$

soit une fonction linéaire de deux variables :

$$[\mathrm{NO_2}] = V_{WE} \cdot \frac{1}{R_L S_W} - V_{AE} \cdot \frac{1}{R_L S_A} + \left(\frac{V_{W0}}{R_L S_W} + \frac{V_{A0}}{R_L S_A} \right)$$

Pour calibrer, il faut donc trouver non plus deux (coefficient directeur et offset) mais trois (coefficient directeur de chaque électrode et offset) constantes.

Q22 En utilisant la sortie de code donnée en figure 16, évaluer l'ordre de grandeur de l'échantillonage des systèmes à bas coût et de la station utilisée pour la phase de calibrage. Proposer deux solutions possibles pour faire coïncider temporellement les données.

Le *Dataframe* des données de la station contient 2237 lignes. On peut constater que les trois heures visibles de mesure sont toutes notées à '0' minutes. La campagne de mesure a duré environ 3 mois, soit environ 90 jours soit $90 \times 24 = 2160$ heures. On peut donc penser que l'échantillonnage des mesures est à l'heure, ou tout au moins de l'ordre de grandeur de l'heure.

À l'inverse le *Dataframe* des données capteurs contient plus de 1,59 million de lignes. 16 capteurs sont utilisés et on peut donc évaluer le nombre de lignes par capteur à environ $1,59 \times 10^6 \div 16 \approx 10^5$. Sur la même période de fonctionnement, il y a donc plusieurs mesures

par heure, au moins 46 ($10^5 \div 2160 \approx 46{,}3$). Sachant que certains capteurs ont été arrêtés en raison de défaillances, on peut penser que l'ordre de grandeur d'échantillonnage de la mesure pour les systèmes à bas coût est de l'ordre de la minute.

Pour comparer les données, on peut envisager :

- augmenter les données de la station – par exemple par interpolation linéaire, mais au risque d'inventer des données sans réalité physique,
- décimer les données des capteurs – par exemple en effectuant des moyennes par heure.

Q23 Proposer le code de la fonction `extract_sensor_data` dont le prototype est donné ci-dessous et qui renvoie un nouveau *DataFrame* qui :

- contient les données correspondant au numéro de capteur choisi,
- contient une nouvelle colonne avec les dates converties en '*pd.TimeStamp*'
- ne contient plus les données des colonnes de l'identifiant du capteur, du format de date initial, du topic, du rssi et du message.

```
def extract_sensor_data(df, sensor_No):
    """
    Load sensor data from the main dataframe.

    Parameters:
    df : DataFrame
        The main DataFrame containing sensor data.
    sensor_No : int
        The sensor number to filter the data.
    Returns:
    DataFrame
        Filtered and processed sensor data with a 'TimeStamp' column.
    """
```

```
def extract_sensor_data(df, sensor_No):
    sensor_data = df.query('id == @sensor_No')
    sensor_data.insert(0, 'TimeStamp', pd.to_datetime(sensor_data['srv_ts'],
        errors='coerce'))
    sensor_data = sensor_data.sort_values(by='TimeStamp')
    sensor_data = sensor_data.drop(columns=['id', 'srv_ts', 'message', '
       topic', 'rssi'])

    return sensor_data
```

Q24 Spécifier ce que réalise le code en détaillant les opérations réalisées par chaque bloc de code. Cette proposition est elle adaptée ? Dans le cas contraire, proposer des corrections nécessaires permettant d'utiliser cette fonction.

Les blocs de code proposés :

- créent un *Dataframe* contenant les données capteurs aux dates spécifiées de calibrage (jour de début ou jour de fin),
- créent une colonne supplémentaire dans le *Dataframe* des données de la station avec la date spécifiée en *TimeStamp*, et en extraient les données aux dates spécifiées de calibrage (jour de début ou jour de fin),
- arrondissent les heures d'enregistrement capteur à l'heure, puis réalisent la moyenne des mesures par heure,
- créent et retournent un *DataFrame* qui est la fusion des deux *DataFrame* (capteurs et station).

Cette fonction réalise donc une décimation avec moyenne des données, cependant la condition logique sur les dates est fausse et il faut inclure l'ensemble des données entre les dates de début et fin. On devrait donc modifier la ligne 6 :

```
1 sensor_data_cal = sensor_data.query('TimeStamp >= @cal_start and TimeStamp
    <= @cal_end')
```

et de la même manière la ligne 12 :

```
1 station_data_cal = station_data.query('TimeStamp >= @cal_start and TimeStamp
    <= @cal_end')
```

2.2 Modélisation linéaire et modélisation multivariée du capteur

Q25 Montrer que ce système a pour solution :

$$\begin{cases} a = \dfrac{\sum_{i=1}^{N}\left(x_i - \overline{\mathrm{X}}\right)\left(y_k - \overline{\mathrm{Y}}\right)}{\sum_{i=1}^{N}\left(x_k - \overline{\mathrm{X}}\right)^2} \\ b = \overline{\mathrm{Y}} - a\overline{\mathrm{X}} \end{cases}$$

où l'opérateur $\bar{\cdot}$ est la moyenne des éléments.

En débutant par le terme b qui est le plus immédiat, par substitution et dérivation :

$$\frac{\partial Q}{\partial b} = \sum_{i=1}^{N} \frac{\partial}{\partial b}\left(y_i - \tilde{y}_i\right)^2 = -2\sum_{i=1}^{N}\left[y_i - (ax_i + b)\right]$$

en posant l'égalité à 0 on obtient :

$$\frac{\partial Q}{\partial b} = 0 \Leftrightarrow b = \frac{1}{N}\sum_{i=1}^{N} y_i - a\frac{1}{N}\sum_{i=1}^{N} x_i$$

où apparaissent les moyennes des éléments des vecteurs Y et X, soit $b = \overline{\mathrm{Y}} - a\overline{\mathrm{X}}$ Par substitution de ce résultat dans la somme des erreurs quadratiques, on obtient :

$$Q = \sum_{i=1}^{N}(y_i - \overline{\mathrm{Y}})^2 - 2a\sum_{i=1}^{N}\left(y_i - \overline{\mathrm{Y}}\right)\left(x_i - \overline{\mathrm{X}}\right) + a^2\sum_{i=1}^{N}\left(x_i - \overline{\mathrm{X}}\right)^2$$

Le premier terme est indépendant de a et disparaîtra à la dérivation, on a donc :

$$\frac{\partial Q}{\partial a} = 0 \Leftrightarrow 2\sum_{i=1}^{N}\left(y_i - \overline{\mathrm{Y}}\right)\left(x_i - \overline{\mathrm{X}}\right) = 2a\sum_{i=1}^{N}\left(x_i - \overline{\mathrm{X}}\right)^2$$

et on obtient donc bien la relation :

$$a = \frac{\sum_{i=1}^{N}\left(x_i - \overline{\mathrm{X}}\right)\left(y_i - \overline{\mathrm{Y}}\right)}{\sum_{i=1}^{N}\left(x_i - \overline{\mathrm{X}}\right)^2}$$

Q26 Le nombre de valeurs étant assez important, on souhaite avoir une implémentation efficace de la régression linéaire. Pour ce faire on dispose des fonctions `mean` et `var` et `cov` de la bibliothèque `numpy` (dont les aides sont données en annexe A.3). En justifiant le raisonnement, compléter le code proposé ci dessous pour calculer le couple de valeur (a, b) en lieu et place des points d'interrogations sur les lignes 11 et 12.

```
1  import numpy as np
2
3  def regression_lineaire(X, Y):
4      """
5      Calculate the slope and intercept of the linear regression line
6      for the given data points X and Y.
7      """
8      X = np.asarray(X)
9      Y = np.asarray(Y)
10
11     a = ???
12     b = ???
13     R2 = 1 - np.sum((Y - (a * X + b)) ** 2) / np.sum((Y - np.mean(Y)) ** 2)
14
15     return a, b, R2
```

On peut remarquer sur le terme b que par division par N au numérateur et au dénominateur on obtient :

$$a = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \overline{\mathrm{X}}\right)\left(y_i - \overline{\mathrm{Y}}\right)}{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \overline{\mathrm{X}}\right)^2} = \frac{\mathrm{Cov}\left[x, y\right]}{\mathrm{Var}\left[x\right]}$$

on peut donc envisager l'implémentation suivante :

```
1  import numpy as np
2
3  def regression_lineaire(X, Y):
4      """
5      Calculate the slope and intercept of the linear regression line
6      for the given data points X and Y.
7      """
8      X = np.asarray(X)
9      Y = np.asarray(Y)
10
11     a = np.cov(X, Y, ddof=0)[0, 1] / np.var(X, ddof=0)
12     b = np.mean(Y) - a * np.mean(X)
```

```
    R2 = 1 - np.sum((Y - (a * X + b)) ** 2) / np.sum((Y - np.mean(Y)) ** 2)

    return a, b, R2
```

Q27 À partir du code donné dans la question précédente, expliciter mathématiquement le terme R^2 renvoyé par la fonction. Vers quelle valeur doit tendre ce terme dans l'idéal, c'est à dire si le modèle permet d'estimer correctement les valeurs vraies ?

À partir du code précédent, on trouve :

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_i - \tilde{y}_i\right)^2}{\sum_{i=1}^{N} \left(y_i - \overline{\mathrm{Y}}\right)^2}$$

le coefficient de détermination de la régression linéaire. On retrouve au numérateur de la fraction Q la quantité minimisée. Cette quantité étant nécessairement positive, elle est au meilleurs cas nulle (pas d'erreur). Dans l'idéal, R^2 tend vers 1.

Q28 En supposant les fonctions demandées justes, proposer une structure de script python permettant :

- d'isoler les données du capteur 55303 sur la durée de la première phase de calibrage,
- d'isoler les données de la station sur la durée de la première phase de calibrage,
- d'uniformiser les données capteur et de la station,
- d'identifier le modèle univarié (fonction linéaire) du capteur.

On peut utiliser le code suivant :

```
# valeurs utiles
sensor_No = 55303
cal_start_date = pd.Timestamp(2016, 6, 2, tz='Europe/Amsterdam')
cal_end_date = pd.Timestamp(2016, 6, 10, tz='Europe/Amsterdam')

# preparation des donnees
sensor_data = extract_sensor_data(data_waag, sensor_No)
aligned_data  = align_data(sensor_data, data_station, cal_start_date,
    cal_end_date)

# regression lineaire
NO_2_true = aligned_data['AB46_NO2']
X = aligned_data['no2a'] - aligned_data['no2b']
a, b, R2 = regression_lineaire(X, NO_2_true)
```

Q29 Commenter les résultats obtenus en figure 17 quant à l'adéquation du modèle aux données.

On peut constater que la valeur de R^2 est proche de 0, donc l'erreur quadratique est importante entre la mesure et la prédiction. Le modèle ne suffit donc pas pour prédire la concentration de NO_2. Ce modèle apportant le même poids aux tensions des électrodes de travail et auxiliaire ne semble donc pas adapté, comme on pouvait s'y attendre par lecture de la documentation technique du constructeur.

Q30 Expliciter la matrice Z d'entrée du modèle et en préciser la taille.

Avec la notation introduite, le vecteur β est de dimension $M+1$, et a comme premier élément l'offset du modèle. Il faut donc une première colonne de 1 dans la matrice Z en plus des données mesurées :

$$\mathrm{Z} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1M} \\ 1 & x_{21} & \cdots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{NM} \end{bmatrix}$$

On a donc une matrice $\mathrm{Z} \in \mathbb{R}^{N\times(M+1)}$

Q31 Par un raisonnement similaire à celui effectué pour la régression linéaire simple, montrer que Q admet un minimum pour $\beta = \left(\mathrm{Z}^T\mathrm{Z}\right)^{-1}\mathrm{Z}^T\mathrm{Y}$

Dans un premier temps, on peut expliciter les résidus par substitution du modèle :

$$Q = (\mathrm{Y} - \mathrm{Z}\beta)^T (\mathrm{Y} - \mathrm{Z}\beta)$$

par développement, on obtient :

$$Q = \mathrm{Y}^T\mathrm{Y} - \mathrm{Y}^T\mathrm{Z}\beta - \beta^T\mathrm{Z}^T\mathrm{Y} + \beta^T\mathrm{Z}^T\mathrm{Z}\beta$$

On peut donc dériver les résidus par rapport à β :

$$\frac{\partial Q}{\partial \beta} = -2\mathrm{Z}^T\mathrm{Z} + 2\mathrm{Z}^T\mathrm{Z}\beta$$

Un minimum est atteint pour la dérivée nulle :

$$\frac{\partial Q}{\partial \beta} = 0 \Leftrightarrow \mathrm{Z}^T\mathrm{Z}\beta = \mathrm{Z}^T\mathrm{Z}$$

et on montre donc bien que :

$$\beta = \left(\mathrm{Z}^T\mathrm{Z}\right)^{-1}\mathrm{Z}^T\mathrm{Y}$$

Q32 En utilisant les fonctions de calcul matriciel de la librairie `numpy`, proposer le code de la fonction suivante renvoyant le vecteur β et la valeur de R^2

```
def multilinear_regression(Y, X):
    """
    Perform multilinear regression

    Parameters:
    Y : array-like
        Dependent variable.
    X : array-like
        Independent variables, should be a 2D array with shape (n_samples,
           n_features).

    Returns:
    arraylike
        Coefficients of the regression model, including the intercept
    float
        R-squared value of the model.
    """
```

On peut proposer le code suivant :

```
import numpy as np


def multilinear_regression(Y, X):
    """
    Perform multilinear regression

    Parameters:
    Y : array-like
        Dependent variable.
    X : array-like
        Independent variables, should be a 2D array with shape (n_samples,
           n_features).

    Returns:
    arraylike
        Coefficients of the regression model, including the intercept
    float
        R-squared value of the model.
    """
    X = np.asarray(X)
    Y = np.asarray(Y)
    (M, N) = X.shape

    # Add a column of ones to X for the intercept term
    X = np.column_stack((np.ones(M), X))

    # Calculate coefficients using the normal equation
    beta = np.linalg.inv(X.T @ X) @ X.T @ Y
    R2 = 1 - np.sum((Y - X @ beta) ** 2) / np.sum((Y - np.mean(Y)) ** 2)

    return beta, R2
```

Remarque : on peut également utiliser les fonctions `np.dot` ou `np.matmul` à la place du @

pour le produit matriciel.

Q33 À partir des résultats obtenus, préciser pour les modèles A et B le nombre de variables d'observations et les colonnes du *Dataframe* utilisées, et analyser de manière quantitative l'adéquation par rapport au capteur utilisé.

À partir des mentions sur les figures, on peut identifier :

- que le modèle A n'utilise que le capteur de NO_2, mais comme il s'agit d'un modèle multivarié, les électrodes W et A sont utilisées. Il y a donc deux variables explicatives (colonnes *no2a* et *no2b*),
- que le modèle B utilise les deux électrodes du capteurs, et des informations issues de capteur de température (T) et d'humidité (RH pour *relative humidity*). Il y a donc quatre variables explicatives (colonnes *no2a*, *no2b*, *temp* et *humidity*)

Les R^2 respectifs des modèles A et B sont 0.298 et 0.825, l'erreur est donc moindre pour le modèle B, qui se rapproche de la droite d'identité sur les nuages de points. On peut en particulier voir une différence notable entre les courbes de la station et du modèle A entre les 2 et 4 juin qui s'atténue pour le modèle B. Cependant, en cas de faibles variation, le modèle B montre également des limites comme entre les 7 et 9 juin où l'amélioration est moins marquée.

Q34 Proposer le code d'une fonction `calculate_ NO2_ concentration` (appelée à la ligne 108) qui implémente le modèle multivarié, et préciser les unités de chaque élément utilisé dans la structure `ModelParameters`.

```
1
2
3 float calculate_NO2_concentration(float we, float aux, float temperature,
      float humidity, ModelParameters model) {
4   return model.a * we + model.b * aux + model.c * temperature + model.d *
        humidity + model.e;
5 }
```

Q35 Quelle est la périodicité de l'appel à la fonction `calculate_NO2_concentration` ? Proposer une stratégie permettant d'avoir un flux de données cohérent par rapport aux données du projet *Urban AirQ*, et en préciser son éventuel intérêt.

La périodicité est de 1 s (voir les délais imposés en lignes 88-94). On peut envisager un moyennage sur 60 valeurs pour retomber à une valeur minute, ce qui permet de réduire du bruit de mesure et de limiter l'influence d'artefacts de mesure.

3 – Cartographie du taux de particules fines dans une zone

3.1 Interpolation par pondération inverse de la distance

Q36 Pour cette question, on suppose que l'on a 3 capteurs $c_1 = (1, 4, 20)$, $c_2 = (4, 3, 40)$ et $c_3 = (2, 1, 50)$. Que vaut $\hat{v}(s)$ pour $s = (2, 3)$? Préciser les calculs intermédiaires.

Pour $s = (2, 3)$, on a $d(s, c_1)^2 = 2$, $d(s, c_2)^2 = 4$, et $d(s, c_3)^2 = 4$. D'où :

$$\sum_{i=1}^{3} w_i(s, c_i) \times m_i = \frac{20}{2} + \frac{40}{4} + \frac{50}{4} = \frac{65}{2}$$

$$\sum_{i=1}^{3} w_i(s, c_i) = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} = 1$$

Et donc, $\hat{v}(s) = \frac{65}{2} = 32{,}5$.

Q37 Donner le code de la méthode `distance_to(self, x, y)` qui retourne la distance euclidienne du capteur considéré avec le sommet s de coordonnées (x, y).

```
def distance_to(self, x, y):
    """Distance euclidienne du capteur au point de coordonnees x et y."""
    return math.sqrt((self.x - x) ** 2 + (self.y - y) ** 2)
```

Q38 On considère un ensemble de capteurs stockés dans `dico_capteurs`.

1. Définir la fonction `Calcul_vchapeau(dico_capteur, x, y)` qui retourne $\hat{v}(s)$ pour un sommet $s \in \{(x, y) \in \{0, \ldots, X_{\max}\} \times \{0, \ldots, Y_{\max}\}\}$.
2. Compléter la fonction `IDW_1(dico_capteur, Xmax, Ymax)` qui retourne dans `grille` les valeurs $\hat{v}(s)$ pour $s \in \{(x, y) \in \{0, \ldots, X_{\max}\} \times \{0, \ldots, Y_{\max}\}\}$.

```
def IDW_1(dico_capteur, Xmax, Ymax):
    grille = [[0 for x in range(Xmax+1)] for y in range(Ymax+1)]

    # A COMPLETER

    return grille
```

3. Évaluer la complexité de ces deux fonctions. Justifier les réponses.

1.

```
def Calcul_vchapeau(dico_capteur, x, y):
    nom = denom = 0

    if (x,y) in dico_capteur:
        return dico_capteur[(x,y)].mesure
```

```
6
7       for c in dico_capteur.values():
8         dist = c.distance_to(x, y)
9         w = 1 / (dist * dist)
10        nom = nom + w * c.mesure
11        denom = denom + w
12
13      return nom / denom
```

2.

```
1       def IDW_1(dico_capteur, Xmax, Ymax):
2       grille = [[0 for x in range(Xmax +1 )] for y in range(Ymax + 1)]
3
4       for y in range(Ymax + 1):
5         for x in range(Xmax + 1):
6           grille[y][x] = Calcul_vchapeau(dico_capteur, x, y)
7
8       return grille
```

3. La fonction `Calcul_vchapeau` parcourt le dictionnaire pour tout c, sa complexité est donc en $\Theta(N)$. La fonction `IDW_1` effectue ce calcul pour tous les éléments de la grille, sa complexité est donc en $\Theta(N \times X_{\max} \times Y_{\max})$.

Q39 On souhaite dans cette question choisir la couleur d'un sommet s en fonction de la valeur de sa mesure m. Pour cela, on fixe les seuils dans la table 3 en fonction des valeurs proposées par l'OMS en 2021, la directive européenne 2008/50/CE et l'AASQA.
Définir la fonction `couleur(m)` qui renvoie le code RGB associé à la mesure m du taux de $PM_{2.5}$, selon l'intervalle auquel m appartient.

```
1   def couleur (m):
2     if (m == -1):
3       return (255,255,255)
4     if (m < 10):
5       return (0,0,255)
6     if (m < 25):
7       return (0,255,0)
8     if (m < 50):
9       return (255,0,0)
10    return (0,0,0)
```

Q40 On souhaite visualiser les valeurs mesurées et estimées sous la forme d'une carte de couleurs. Pour cela, on crée une image `img` sous la forme d'une matrice de triplets de taille $(echelle \times Y_{\max}) \times (echelle \times X_{\max})$. Le paramètre $echelle$ est un entier qui permet d'associer un carré de surface $(echelle)^2$ à chaque point dans le but d'agrandir l'image. L'image sera convertie en fichier `png` dont le nom est également passé en paramètre de la fonction. La figure 20 présente un exemple de figure obtenue.

1. Compléter la fonction `creation_image` dont le code partiel suit et qui effectue ce traitement.
2. Quelle est la complexité de cette fonction ? Justifier la réponse.

1.

```
            couleurRGB = couleur (grille[y][x])
            for i in range(0, echelle):
              for j in range(0, echelle):
                img[echelle * y + i][echelle * x + j] = couleurRGB
```

2. On a 4 boucles imbriquées dont les bornes sont fixées. On obtient donc une complexité en $\Theta(X_{\max} \times Y_{\max} \times (echelle)^2)$.

3.2 Étude d'une zone dense

Q41 1. Décrire un algorithme qui calcule la valeur minimale $R_{\min} \in \mathbb{R}$ telle que, pour tout $s = (x, y) \in \{0, \ldots, X_{\max}\} \times \{0, \ldots, Y_{\max}\}$ il existe au moins un capteur $c_i = (x_i, y_i, m_i)$ tel que $d(s, c_i) \leq R_{\min}$.

2. Quelle est la complexité de cet algorithme dans le pire des cas ? Justifier la réponse.

1. Pour tout sommet s, on peut calculer la valeur $R_{\min}(s)$ correspondant à la distance minimale de s à un capteur en parcourant l'ensemble des capteurs. Par définition, $R_{\min} = \max_s(R_{\min}(s))$ qui se calcule en parcourant les valeurs obtenues.
2. Dans le pire des cas, il faut parcourir tous les capteurs pour calculer $R_{\min}(s)$. La complexité est alors en $\mathcal{O}(Y_{\max} \times X_{\max} \times N)$.

Q42 Pour cette question, on considère les capteurs dont les emplacements sont représentés par la figure 21.

1. Donner sans explication la valeur de $R_{\min}$. En déduire que $R = 2$, considérée par la suite, est une valeur réalisable.
2. Donner les valeurs $\bar{\jmath}$ et $\bar{\imath}$. Donner explicitement les ensembles $\mathcal{P}_{1,2}$ et $\mathcal{P}_{2,4}$.

1. $R_{\min} = \sqrt{2}$. La valeur $R = 2$ est un entier plus grand que $R_{\min}$, elle est donc réalisable.
2. $\bar{\imath} = \lceil \frac{7}{2} \rceil = 4$, $\bar{\jmath} = \lceil \frac{9}{2} \rceil = 5$,
   $\mathcal{P}_{1,2} = \{(x, y), 4 \leq x \leq 6 \text{ et } 2 \leq y \leq 4\}$ soit
   $\mathcal{P}_{1,2} = \{(4, 2), (4, 3), (4, 4), (5, 2), (5, 3), (5, 4), (6, 2), (6, 3), (6, 4)\}$ et
   $\mathcal{P}_{2,4} = \{(x, y), 8 \leq x \leq 9 \text{ et } 4 \leq y \leq 6\}$ soit
   $\mathcal{P}_{2,4} = \{(8, 4), (8, 5), (8, 6), (9, 4), (9, 5), (9, 6)\}$.

Q43 Les éléments de $\mathcal{P}_{i,j} \cap \mathcal{C}$ peuvent être stockés sous la forme d'une liste ou d'un dictionnaire. Quels sont les avantages à choisir un dictionnaire par rapport à une liste ? On rappelle que les dictionnaires sont stockés en *Python* sous la forme d'une table de hachage, tandis que les listes sont des tableaux.

Dans un ensemble, on ne peut avoir deux fois le même élément.

1. Si les ensembles de capteurs sont gérés sous forme de liste, il va falloir rechercher si un élément est dans la liste avant de le placer dedans pour éviter les doublons. Cette opération est de l'ordre de $\mathcal{O}(n)$ où n est le nombre d'éléments de la liste en cours de construction.
2. Si on utilise une table de hachage, ces opérations sont en $\Theta(1)$. C'est donc plus intéressant d'un point de vue complexité.

Q44 Le but de cette question est de programmer le calcul de la matrice PC.

1. Définir la fonction `is_inside (c, i, j, R)` qui teste si le capteur c est dans $\mathcal{P}_{ij}$.
2. En déduire la définition de la fonction `calcul_PC(dico_capteur, i, j, R)` qui retourne un ensemble contenant les éléments de $\mathcal{P}_{i,j} \cap \mathcal{C}$.
3. En déduire la définition de la fonction `matrice_PC(dico_capteur, R, Xmax, Ymax)` qui retourne la matrice PC telle que, pour tout couple $(i, j) \in \{0, \ldots, \bar{i}\} \times \{0, \ldots, \bar{j}\}$, PC_{ij} est l'ensemble des éléments de $\mathcal{P}_{i,j} \cap \mathcal{C}$.

1.
```python
def is_inside(c, i, j, R):
  test_1 = (c.x >= (j*R)) and (c.x <= ((j+1)*R))
  test_2 = (c.y >= (i*R)) and (c.y <= ((i+1)*R))
  return test_1 and test_2
```

2.
```python
def calcul_P(dico_capteur, i, j, R):
  p = dict()
  for c in dico_capteur.values():
    if is_inside(c, i, j, R):
      p[(c.x,c.y)] = c
  return p
```

3.
```python
def matrice_P(dico_capteur, R, Xmax, Ymax):
  ibar = math.ceil(Ymax/R)
  jbar = math.ceil(Xmax/R)

  matrice = [[set() for j in range(jbar)] for i in range(ibar)]

  for i in range (0,ibar):
    for j in range (0,jbar):
      matrice[i][j] = calcul_P(dico_capteur, i, j, R)
  return matrice
```

Q45 Soit $A = \{(i,j) \in \{0,\dots,\bar{i}\} \times \{0,\dots,\bar{j}\}\}$ l'ensemble des couples d'indices réalisables (i,j) de $\mathcal{P}_{i,j}$. Soit $s = (x,y)$ un élément de la zone étudiée.

1. On suppose que x et y sont tous les deux divisibles par R. Définir $\mathcal{D}(s)$.
2. Même question pour x divisible par R et y non divisible par R.
3. Même question pour x et y non divisibles par R.
4. En déduire que, pour tout sommet $s = (x,y) \in \mathcal{P}_{i,j}$, $\mathcal{D}(s)$ est inclus dans un sous-ensemble de A à préciser et qui comporte au plus 9 éléments. On nomme cet ensemble $\mathcal{D}^\star_{ij}$. Donner l'expression de $\mathcal{D}^\star_{ij}$.

1. On suppose que x et y vérifient $x = jR$ et $y = iR$. On observe que dans ce cas, s est élément de $\mathcal{P}_{i,\tilde{j}}$, $\mathcal{P}_{i-1,\tilde{j}}$, $\mathcal{P}_{i,\tilde{j}-1}$, et $\mathcal{P}_{i-1,\tilde{j}-1}$. De plus, l'union de ces 4 ensembles couvrent les coordonnées de $\mathcal{C}(s)$. Ainsi $\mathcal{D}(s) = \{(i,j),(i-1,j),(i,j-1),(i-1,j-1)\} \cap A$.
2. Supposons que $iR < y < (i+1)R$. Comme précédemment :
$$\mathcal{D}(s) = \{(i,j),(i-1,j),(i-1,j),(i-1,j-1),(i,j+1),(i-1,j+1)\} \cap A$$
3. Supposons que $iR < y < (i+1)R$ et $jR < x < (j+1)R$. Alors :
$$\begin{aligned}\mathcal{D}(s) =& \{(i,j),(i-1,j),(i-1,j),(i-1,j-1),(i,j+1),\\ & (i-1,j+1),(i+1,j-1),(i+1,j),(i+1,j+1)\} \cap A\end{aligned}$$
4. On déduit des questions précédentes que :
$$\begin{aligned}\mathcal{D}^\star(s) =& \{(i,j),(i-1,j),(i-1,j),(i-1,j-1),(i,j+1),\\ & (i-1,j+1),(i+1,j-1),(i+1,j),(i+1,j+1)\} \cap A\end{aligned}$$

Q46 Donner la complexité totale du calcul de la matrice $\mathcal{D}^\star_{ij}$. Préciser pour cela les différentes étapes de ce calcul ainsi que leur complexité.

Pour calculer la matrice $\mathcal{D}^\star$, il faut procéder aux étapes suivantes vues précédemment :

1. Il faut d'abord calculer $\mathcal{P}$. Pour cela, on parcourt toutes les cases possibles et à chaque fois le dictionnaire en entier. On obtient donc du $\Theta(N \times X_{\max} \times Y_{\max})$.
2. Ensuite, il faut calculer $\mathcal{D}^\star$. Pour cela, on teste les au plus 9 cases adjacentes pour chaque couple (i,j) de $\mathcal{P}$ et on réunit les capteurs présents dans ces cases. C'est donc également du $\Theta(N \times X_{\max} \times Y_{\max})$.

Q47 On suppose dans cette question que la matrice $\mathcal{D}^\star$ a été calculée dans une phase de pré-traitement.

1. Expliciter comment vous pouvez utiliser la matrice $\mathcal{D}^\star$ pour calculer $\tilde{v}(s)$;
2. En déduire la complexité du calcul de $\tilde{v}(s)$, si s est un sommet de $\mathcal{P}_{ij}$. On suppose que les capteurs sont uniformément répartis sur la zone : dans une case de $\mathcal{P}$, le nombre de capteurs est approximativement égale à $\frac{N}{i \times j}$. Justifier la réponse.
3. En déduire la complexité du calcul de $\tilde{v}(s)$ pour l'ensemble des sommets. Justifier la réponse.
4. Comparer les deux méthodes, c'est-à-dire pour calculer $\hat{v}$ et $\tilde{v}$.

1. On a vu précédemment que pour $s \in \mathcal{P}_{ij}$, $\mathcal{C}(s) \subseteq \mathcal{D}^\star_{ij}$. Il suffit donc de considérer uniquement les sommets de $\mathcal{D}^\star_{ij}$ pour construire $\mathcal{C}(s)$.
2. Puisque on teste une fois chaque capteur de la zone $\mathcal{D}^\star$ associée à s, on a du $\Theta(9 \times \frac{NR^2}{X_{\max} Y_{\max}})$.
3. Comme on le fait pour tous les sommets de $\mathcal{P}$, on obtient du $\Theta(9 \times NR^2)$.
4. Pour récapituler :
   - La première méthode calculait $\hat{v}(s)$ pour tout sommet de la zone en $\Theta(N \times X_{\max} \times Y_{\max})$.
   - La seconde méthode calcule $\tilde{v}(s)$ qui est une valeur approximée de $\hat{v}(s)$ en ne considérant que les capteurs dans un rayon R. Après un pré-traitement qui coûte $\Theta(N \times X_{\max} \times Y_{\max})$, mais effectué une seule fois, le calcul de $\tilde{v}(s)$ prend du $\Theta(9 \times NR^2)$. Si on prend R bien inférieur à $X_{\max}$ et $Y_{\max}$, cette méthode est beaucoup plus rapide, mais le résultat pourrait être moins bon.

IV Commentaires du jury

Le sujet portait sur une problématique sociétale importante (la qualité de l'air) où la modélisation joue un rôle décisif à bien des égards. Les trois parties du sujet adoptaient une approche ascendante au regard de la chaîne globale d'information :

- la partie 1 étant consacrée à l'**obtention des données** à partir des phénomènes physiques, appliquée à l'étude du principe de détection d'un compteur optique de particules fines ;
- la partie 2 se focalisant sur la **calibration des capteurs** à bas coût de mesure de concentration de NO_2 en comparant leurs données avec celles des stations de référence ;
- la partie 3 s'attachant à la **production d'information** à partir des données consolidées, notamment en matière de cartographie du taux de particules fines à destination du grand public.

Chaque partie était l'occasion de solliciter les candidats sur un éventail différent de connaissances et de compétences scientifiques et techniques en matière de modélisation.

Partie 1 : du phénomène physique à la donnée

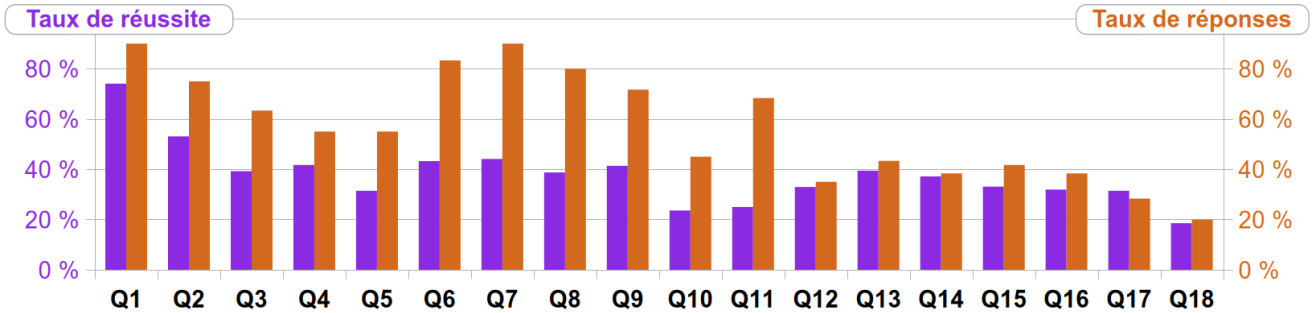
Cette partie portait sur le principe de détection d'un compteur optique de particules (OPC). Elle faisait appel à des connaissances et des compétences normalement acquises en premier cycle d'études supérieures, tant en matière de sciences physiques que d'outils mathématiques et de techniques de programmation. Elle était elle-même divisée en **trois sous-parties indépendantes**.

La première sous-partie (questions Q1 à Q5) mettait en place la **modélisation du phénomène physique** de diffusion d'une onde lumineuse (un faisceau laser) par une particule de matière, basée sur la théorie de Mie. Il n'était pas attendu des candidats de connaître cette théorie, le déroulé des questions étant conçu dans une perspective de découverte, avec l'appui d'une courte annexe (deux pages) qui rappelait le formalisme de représentation des ondes électromagnétiques par des fonctions vectorielles complexes, et quelques propriétés remarquables pour les cas les plus simples (onde monochromatique linéairement polarisée ou sphérique). Les écritures et calculs demandés étaient assez élémentaires, la seule difficulté résidant dans l'appropriation du contexte et du formalisme pour des candidats a priori non spécialisés en sciences physiques.

La deuxième sous-partie (questions Q6 à Q15) abordait la **modélisation informatique du mesurande d'un OPC** à l'aide d'une bibliothèque de fonction développée en langage *Python*, dont seules trois fonctions de haut niveau étaient exploitées, et données dans une courte annexe (deux pages). Ici encore, les questions étaient conçues pour lisser au maximum la difficulté de codage, qui ne demandait aucun algorithmique subtil. Et c'était aussi l'appropriation des données et du contexte qui constituait la « difficulté » de l'exercice. Seuls quelques détails comme les unités des grandeurs calculées, les bornes d'intervalle des itérations, et le passage en coordonnées logarithmiques linéaires ou polaires nécessitaient des précautions, sachant que des commentaires étaient ajoutés sur les listings pour guider les candidats. Dans cette sous-partie également, des résultats numériques de simulation faisaient l'objet de questions particulière d'analyse (Q8, Q11, Q15), requérant une prise de recul. Là, il importait de ne pas perdre de vue la technologie du système étudié, en particulier le fait qu'il ne disposait d'aucun moyen de connaître le matériau des particules détectées, ce qui était rappelé dans le sujet.

Enfin, la troisième sous-partie (questions Q16 à Q18) achevait l'analyse du **principe de mesure d'un OPC** basée sur le calcul théorique de la concentration massique des particules fines dans l'air ambiant en mettant en rapport la somme de la masse cumulée de ces dernières avec le volume d'air analysé, en faisant d'inévitables hypothèses simplificatrices (particules sphériques de masse volumique uniforme). Grâce aux courbes de mesurande obtenues par simulation, on pouvait alors estimer théoriquement l'erreur relative commise pour des matériaux emblématiques (carbone suie, oxydes de fer). Si ces calculs étaient simples, la conclusion attendue en dernière question appelait à une réflexion approfondie, permettant aux meilleurs candidats de se départager.

Sans grandes surprises, cette première partie a été abordée par presque tous les candidats, avec un **taux de réponses moyen** de 57 %. Cependant, le taux de réponses a chuté en dessous de la barre des 50 % à partir de la question Q12 (peut-être à cause des formules d'intégration données dans le sujet, qui pourraient avoir effrayé certains candidats) – mais avec une décroissance notable seulement pour les questions Q17 et Q18, comme on peut le voir sur l'histogramme ci-dessous.

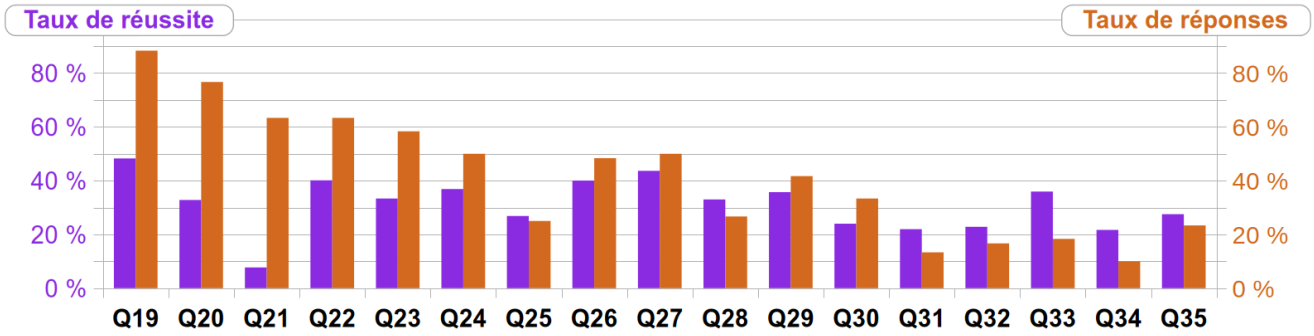


En revanche, le **taux de réussite** a été globalement décevant, puisqu'il a chuté sous la barre des 50 %

dès la question Q3. Et en définitive, la notation des copies a révélé, dès cette première partie, que seulement un sixième de l'effectif des candidats était capable de fournir suffisamment d'éléments de réponse pertinents pour donner aux correcteurs le sentiment qu'ils disposaient d'un socle de connaissances et de compétences théoriques satisfaisant.

Partie 2 : calibration des capteurs

Cette partie portait sur la calibration des capteurs à bas coût de NO_2 en comparant les données qu'ils produisent avec celles issues de stations de mesure de référence, permettant ainsi d'obtenir des données suffisamment fiables pour produire ensuite de l'information. Elle sollicitait les candidats sur des connaissances et des compétences scientifiques et techniques normalement acquises en fin de licence et début de master. Elle était divisée en **deux sous-parties indépendantes**.



Une première sous-partie, faisant appel à l'analyse d'un système de transduction simple, permettait de mettre en évidence les défauts potentiels des capteurs à bas coût de NO_2 et la nécessité de mettre en place un protocole avec une phase de calibrage, en se basant sur une station de référence. Cette partie se terminait par la prise en main de données récoltées sur une campagne de mesure réelle en utilisant des API non-standards de Python de manière guidée. Une seconde partie consistait à mettre en œuvre un modèle linéaire, puis multi-linéaire prenant en compte un ensemble de variables explicatives également enregistrées pendant la campagne de mesure.

Cette dernière sous-partie comprenait une approche formelle sur des techniques maintenant classique à l'heure des modèles basés données : une régression linéaire simple par minimisation des moindres carrés et une régression multi-linéaire par généralisation. Ces méthodes sont maintenant fondamentales et la maitrise de leur principe, de leur mise en œuvre et l'analyse statistique des résultats est un prérequis essentiel pour l'ingénierie informatique actuelle. Le jury constate que ces méthodes sont souvent peu maîtrisées, que des erreurs sur des calculs simples (dérivées, calcul de la taille d'une matrice dans un calcul donné) ou d'analyse des résultats sont régulières.

Partie 3 : de la donnée à l'information

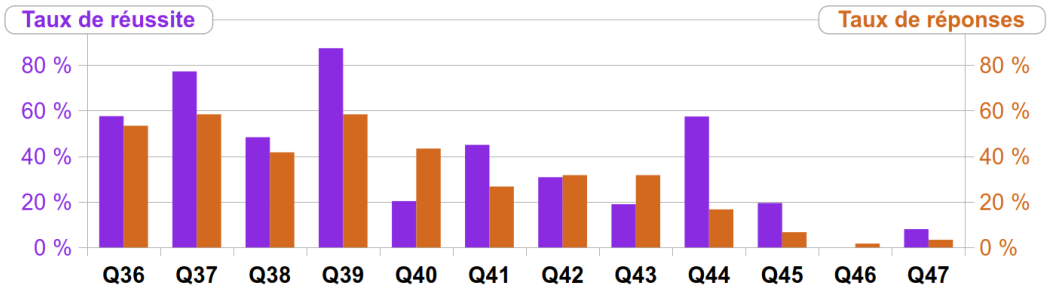
Cette dernière partie portait sur des méthodes de cartographie de taux de particules fines pouvant être utilisées pour fournir des informations à destination du grand public à partir des données issues des capteurs. Là encore, elle sollicitait les candidats sur des connaissances et des compétences scientifiques et techniques normalement acquises en fin de licence et début de master – principalement, l'algorithmique (exprimée en langage *Python*) et la théorie de la complexité. Un peu plus courte que les précédentes, elle était divisée en **deux sous-parties indépendantes**.

Dans la première sous-partie (Q36 à Q40), on modélisait une zone géographique quadrillée équipée de capteurs par une matrice de position de ces derniers. On étudiait alors une première **méthode d'interpollation par pondération inverse de la distance** pour calculer un taux de particules fines supposé représentatif de la réalité en un point quelconque du quadrillage en fonction des mesures établies par tous les capteurs de la zone, mais en tenant compte bien entendu de leur distance au point

considéré. Après quelques applications numériques pour se familiariser avec les formules, les candidats étaient invités à proposer et analyser des algorithmes de calcul de distances, de moyennes pondérées, et de parcours.

Dans la deuxième sous-partie (Q41 à Q47), on abordait l'**étude d'une zone dense** en assimilant les données issus des capteurs à celles issues des modèles aérodynamiques de transport de composants chimiques, considérés suffisamment fiables pour constituer des données consolidées. En effet, une telle situation faisait émerger la nécessité de limiter les données prises en compte à celles situées à une distance limitée du point considéré. Là encore, la modélisation reposait sur une matrice de position, mais faisait intervenir un découpage en sous-zones basée sur une distance limite (à determiner), avec à la clef des considérations de divisibilité, d'appartenance d'un point à une sous-zone, ainsi que de techniques de stockage (liste ou dictionnaire).

Sans doute en raison de son positionnent en fin de sujet, cette partie n'a été abordée que par seulement deux tiers des candidats, avec un **taux moyen de réponses** de 31 %, et une nette décroissance dès la question Q39, comme le montre clairement l'histogramme ci-dessous. En revanche, le **taux de réussite** des questions de cette partie est resté voisin ou supérieur à 50 % pour la moitié d'entre elles. Et la notation des copies a mis en évidence que la pugnacité des candidats ayant fait l'effort d'aller jusqu'au bout du sujet leur a permis de se départager significativement des autres. Car, comme pour les parties précédentes, on a observé environ un sixième des notes nettement au dessus de la moyenne du lot.



A contrario, les correcteurs ont déploré de trop fréquentes erreurs grossières dans le traitement des questions de complexité algorithmique, le terme étant interprété par nombre de candidats dans son sens général et non pas spécialisé. Une telle confusion est particulèrement décevante dans le cadre d'un concours de recrutement d'enseignants en spécialité informatique.

V Recommandation aux futurs candidats

En complément des commentaires supra, le jury recommande aux futurs candidats, en vue d'une bonne préparation à la prochaine épreuve écrite de modélisation :

- de consulter attentivement le programme officiel du concours, et en particulier de tenir compte de ses dernières évolutions ;
- d'étudier avec soin les éléments de correction proposés pour le présent sujet, notamment en matière :
  - ⋆ de concision (les candidats ont tendance à trop détailler leur réponses aux premières questions, perdant ainsi un temps précieux sans récolter davantage de points) ;
  - ⋆ de rigueur dans la rédaction (séparation des calculs littéraux et des applications numériques, expression des unités, vérification intuitive des ordres de grandeur des valeurs obtenues) ;
  - ⋆ de simplicité des algorithmes demandés (les auteurs de sujet ayant a priori conscience qu'on ne peut pas avoir les mêmes exigences pour une solution proposée sur une copie que sur un ordinateur).
- de privilégier la culture scientifique et technique générale à la connaissance spécialisée de tel ou tel domaine, car la thématique des sujets sera toujours différente d'une année à l'autre.

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 ;
- maitriser, 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 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 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'examinateur 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 ;
- l'objectif et les compétences visées par la séance à caractère expérimental ;

- la présentation d'une ou des activités que devraient mener les élèves et l'analyse des résultats 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 de la séquence pédagogique incluant la séance à caractère expérimental développée au regard du contexte imposé.
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 2026 étaient les suivants :
- voiture radiocommandée;
- système de travelling ;
- imprimante 3D ;
- axe linéaire didactisé ;
- maquette didactisée d'un bras de pelleteuse ;
- 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. Il convient de rappeler que cette séance constitue un élément essentiel de la didactique propre aux Sciences de l'Ingénieur. Or, certains candidats se limitent trop souvent à des activités de simulation qui ne sauraient se substituer à une séance expérimentale fondée sur la manipulation, l'observation, la mesure et la confrontation au réel. 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 qui s'inscrit dans un contexte pédagogique imposé. Elle doit faire apparaître les prérequis, 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, modeleur, 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 maitrise 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 ;
- maitriser la durée consacrée à chaque activité ;
- maitriser 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 remédiations 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 l'objectif, 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 maitrisés 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.

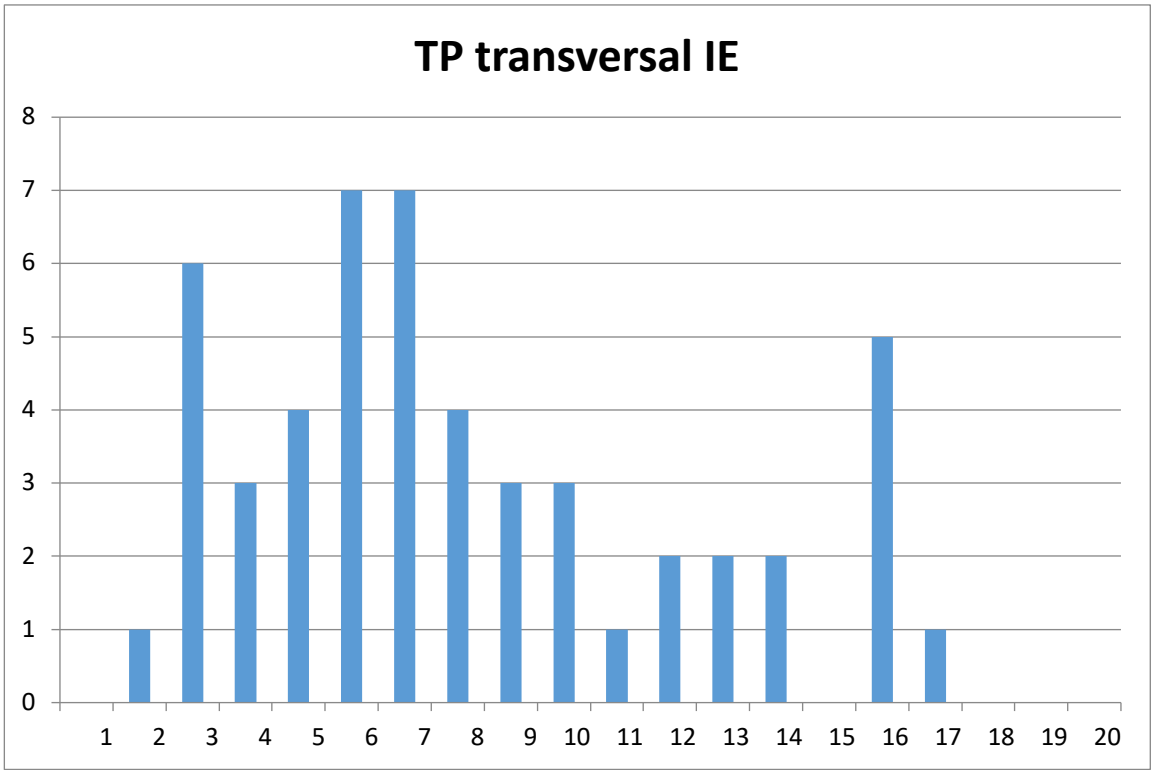
Le jury apprécie lorsque les diapositives sont numérotées lors de la présentation orale : cela lui permet de se référer plus facilement aux éléments de présentation. Le jury déplore les fautes d'orthographe dans les supports numériques présentés ; elles nuisent à l'image du candidat. Les candidats se 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 2026 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'approprier 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, les modalités d'évaluation certificative, des objectifs de formation associés et des stratégies pédagogiques préconisées.

C. Résultats

La moyenne des notes obtenues est de 7,75/20. L'écart-type est de 4,20. La meilleure note est 16,20. La plus faible est 1,20. La médiane est de 6,9.


TP transversal IE
8
7
6
5
4
3
2
1
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

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**
Titre de la séquence	Comment les nouvelles technologies peuvent-elles améliorer l'habitat ?	Comment optimiser l'utilisation de l'énergie de façon autonome tout en respectant l'environnement ?
Niveau de formation visé	STI2D enseignements communs de 2I2D en Terminale	
Supports pédagogiques	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 : - ventilation mécanique contrôlée double-flux ; - système de climatisation ; - panneaux photovoltaïques ; - volet roulant solaire autonome ; - système Eolicc ; - systèmes variés de chauffage ; - système d'éclairage communicant de salles de spectacles ; - thermostat connecté et intelligent ; - pompe à chaleur ; - …	
Effectif	Classe de 24 élèves, groupe à effectif réduit de 12 élèves	
Volume horaire	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 :

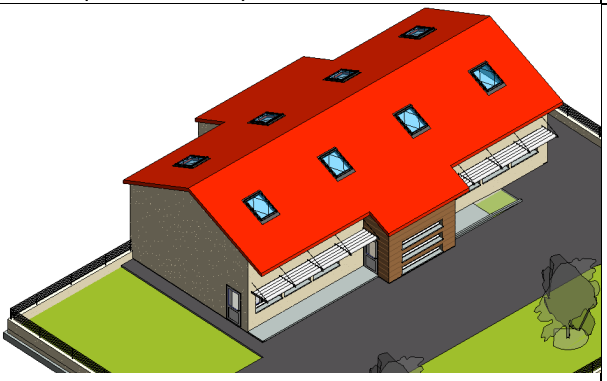
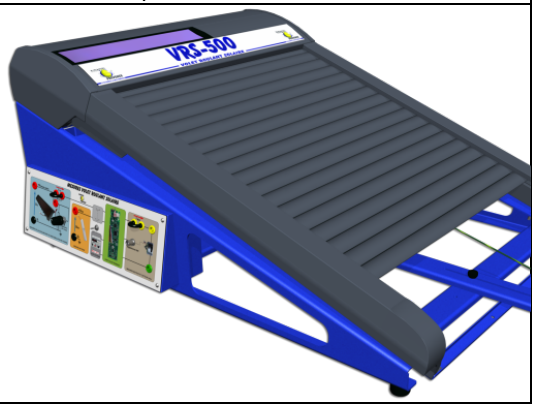
- contextualisé la séquence pédagogique dans une grande thématique ;
- recensé les compétences à développer et les savoir-faire et savoirs à faire acquérir aux élèves ;

- 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 ».

Maquette numérique d'une école élémentaire	Maquette didactisée de Velux
	

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'éclairement naturel grâce à des Velux tout en maitrisant 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'éclairement 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 2ème environnement virtuel VRS-500_DD_EMP, compléter le document réponse DR2 en indiquant les différents sous-systèmes formant la chaine de puissance et la chaine 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.

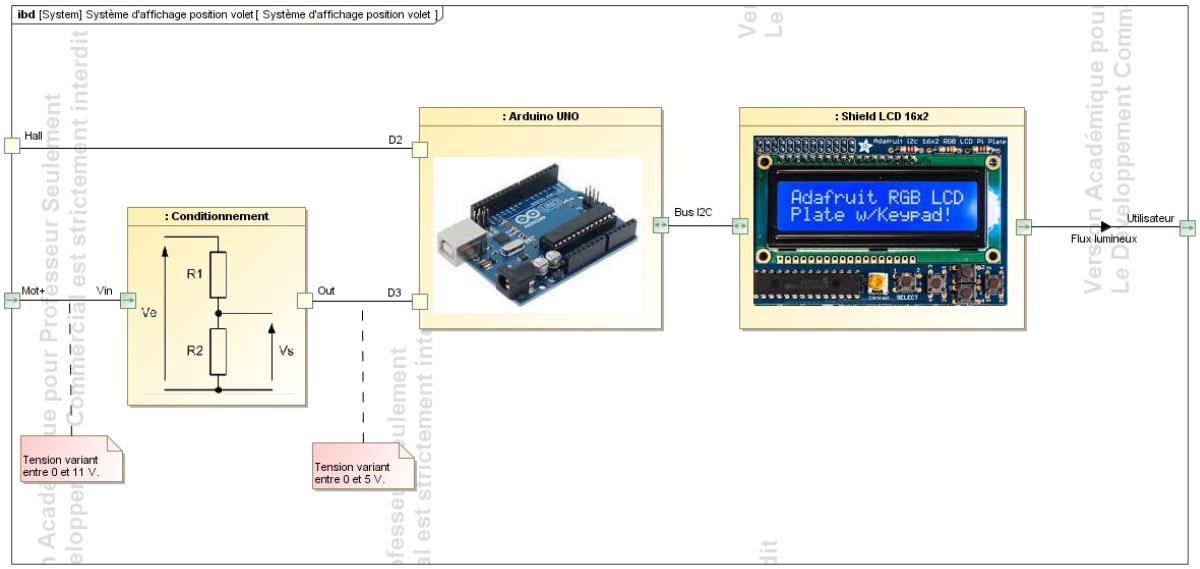
Partie 1.3 – expérimentations pour répondre à la problématique technique et scientifique (durée : 2h00)

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.

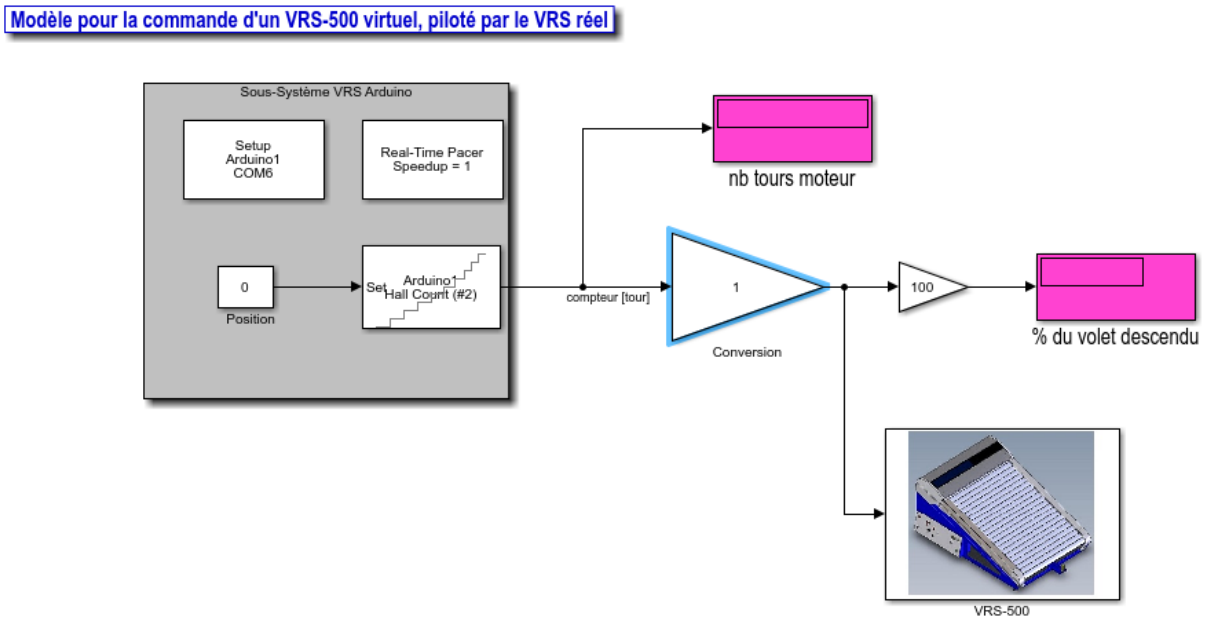
On dispose pour cela d'une interface programmable à base de microcontrôleurs constituée comme indiqué sur la figure suivante.



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.



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.

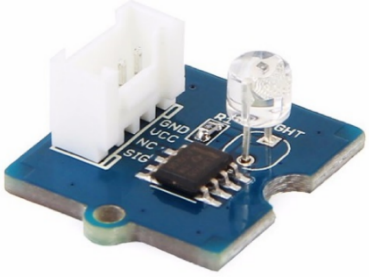
Une fois l'activité 4 terminée, défaire le câblage du sous-système, fermer Matlab et relier le câble USB de la maquette instrumentée à l'ordinateur.

Étude de l'ouverture optimale du volet roulant

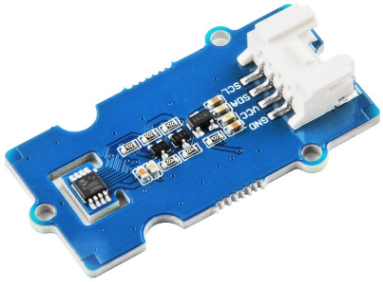
Le volet roulant du Velux a pour fonction de se protéger de la chaleur en été tout en optimisant l'éclairement intérieur naturel. L'objectif de cette partie est de déterminer expérimentalement l'ouverture optimale correspondant au réglage : « protection solaire sans obscurité complète » de ce volet roulant 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'éclairement 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'éclairement



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 Ti et l'éclairement intérieur Ei en régime permanent en fonction de l'ouverture du volet, et le présenter à votre examinateur avant de le mettre en œuvre.
Tracer Ti = f(% ouverture volet) et Ei = f(% 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 connaitre 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 :

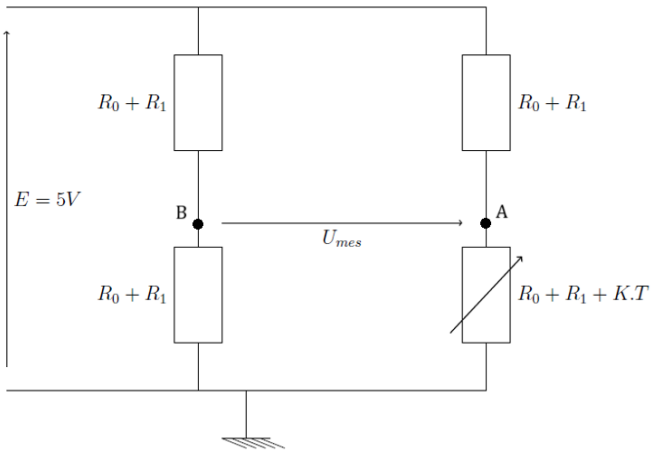
- T, la température en °C de la paroi sur laquelle la sonde est attachée ;
- R0, la valeur de R(T) pour T=0°C, en Ω ;
- K, la valeur du gain de la sonde en $\Omega.°C^{-1}$.

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 R1 = 10 kΩ. 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 :

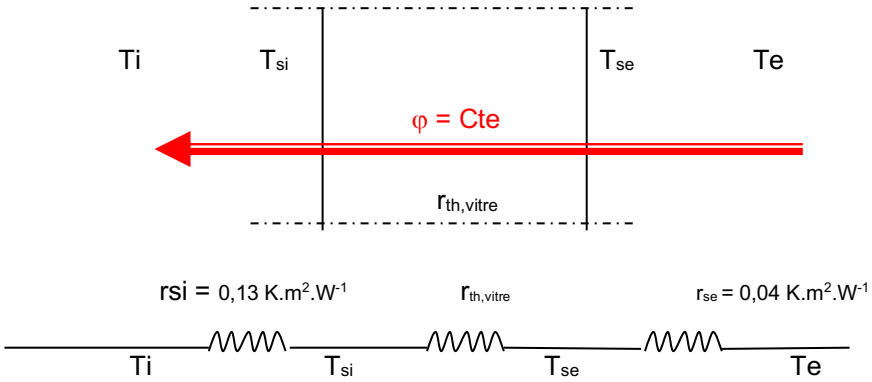
- Tse et Tsi, respectivement les températures de surface extérieure et intérieure en K ;
- rth,vitre, la résistance thermique du vitrage considéré en $K.m^2.W^{-1}$;
- φ : le flux thermique à travers la paroi en $W.m^{-2}$.

Afin de pouvoir implémenter un modèle de simulation pour dimensionner un local, et évaluer les échanges thermiques, il est nécessaire de connaitre 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 rth,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 (éclairement 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 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 courant 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 2026 étaient les suivants :

- badgeuse sécurisée ;
- portique de détection ;
- robot pédagogique ;
- drone instrumenté communicant ;
- traitement d'images IRM pour localisation de tumeurs cérébrales ;
- système de contrôle d'accès RFID ;
- contrôle de trafic aérien.

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 maitrise 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 ainsi que des évaluations de compétences, 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 apparaitre 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 ;
- trop peu de candidats présentent des activités à réaliser par les élèves dans la séance et leur enchaînement
- les modalités d'évaluation des compétences abordées doivent être précisées.

- **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 maitriser 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 maitriser 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 durée 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 :

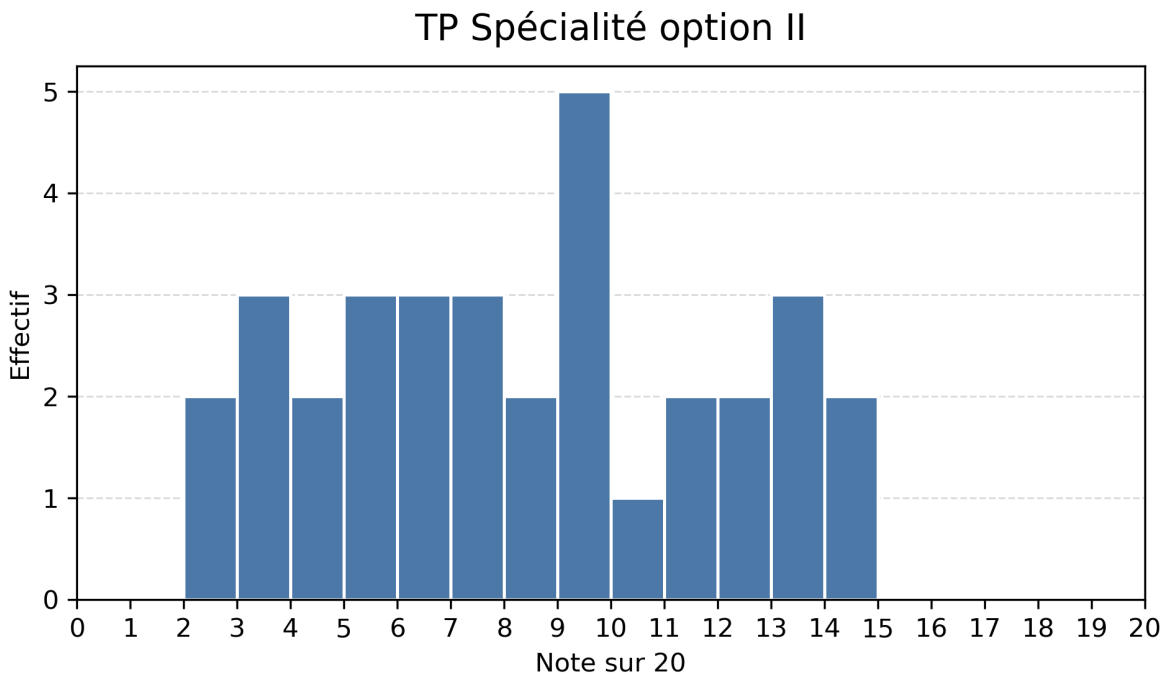
- le choix de l'objectif pédagogique qui doit s'appuyer sur un support technique représentatif des sciences de l'ingénieur. Ce dernier est imposé par le sujet de TP ;
- le choix des compétences visées par le candidat. Il n'est pas nécessaire de choisir l'ensemble des compétences définies dans le sujet 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 8,26/20 avec un écart type de 3,75. La meilleure note est de 14,6/20 et la note la plus basse est de 2/20. 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

Contexte

En imagerie médicale, les techniques de vision assistée par ordinateur ont permis une meilleure interprétation des données de diagnostic et ouvrent la voie à l'amélioration des techniques et technologies de soin. Le contexte expérimental de cette épreuve est centré sur la détection de tumeurs cérébrales sur des données d'imagerie par résonance magnétique (IRM). Ce type de données est obtenue de manière non invasive et permet d'obtenir des vues du système nerveux central avec une résolution en contraste suffisante pour identifier certaines pathologies.

Cette épreuve se focalise sur l'identification et la segmentation des images d'IRM, dans le but de recréer informatiquement la boite crânienne et la ou les tumeurs en 3D pour faciliter le pilotage des interventions chirurgicales.

Sur le plan logiciel, le *framework* est constitué :

- d'une base de donnée d'examens IRM, les méta-données concernant les patients ainsi que la vérité terrain de la localisation des tumeurs ;
- d'un code source incomplet pour l'apprentissage supervisé de la localisation de tumeurs (bibliothèque *learn*) ;
- d'un code source pour la reconstruction des images IRM et leur visualisation (bibliothèque *viewer*).

Le langage utilisé pour cette épreuve est Python. Le candidat dispose des bibliothèques pré-installées sur une machine de calcul accessible depuis le poste de travail. Parmi ces bibliothèques, on trouve notamment : *Numpy*, *Scipy*, *Matplotlib*, *Pandas*, *OpenCV*, *Tensorflow* et *Open3D*. Les accès aux documentations de ces API sont donnés en annexe 7.1. Le candidat dispose de l'environnement de programmation *Visual Studio Code (vscode)*.

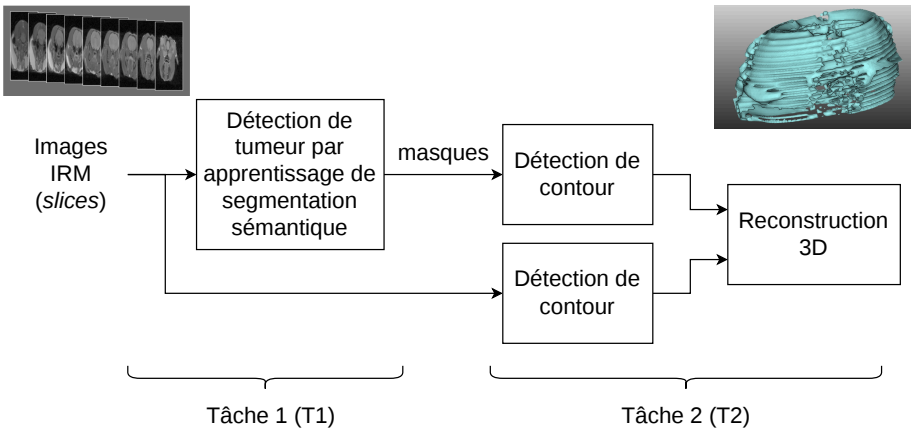


FIGURE 1 – Tâches associées à la détection et localisation de tumeurs cérébrales à partir de donnée IRM

Objectifs

L'objectif de ce travail expérimental est double :

- déployer un modèle d'apprentissage automatique pour la détection de tumeur,
- visualiser les données d'IRM brutes et les données de sortie du modèle afin de reconstruire en 3-dimensions la ou les tumeurs dans la boite crânienne.

Ces deux objectifs, illustrés dans la figure 1 sont chacun associés à deux tâches qui constituent le travail de la deuxième partie de la première phase de l'épreuve. Ces deux parties peuvent être traitées indépendamment.

4 Première phase, première partie (durée 0h30)

4.1 Prise en main de l'environnement de travail

Activité 1. Le client remote-ssh de *vscode* est pré-installé sur le poste de travail. Au début de la manipulation *vscode* est connecté au serveur de calcul distant sur lequel vous ferez la manipulation. En cas de déconnexion le mot de passe à est utiliser est le mot de passe candidat (identique à celui du poste de travail). Se placer dans *Documents/TP_IRM/*, dans le terminal comme dans l'arborescence.

4.2 Prise en main du code de la tâche 1

Activité 2. Ouvrir, lire puis lancer le script *01_test_data.py*. Identifier les classes, méthodes et fonction de la bibliothèque *learn*. Modifier le script pour afficher 9 images sur 3 lignes et 3 colonnes.

Activité 3. Retrouver les tailles des variables *image* et *mask*.

Activité 4. La figure 2 présente le flux des phases d'apprentissage. En utilisant le script *03_test_learn.py*, identifier les objets, attributs et méthodes utilisées pour les phases de la Tâche 1 dans la bibliothèque *learn*.

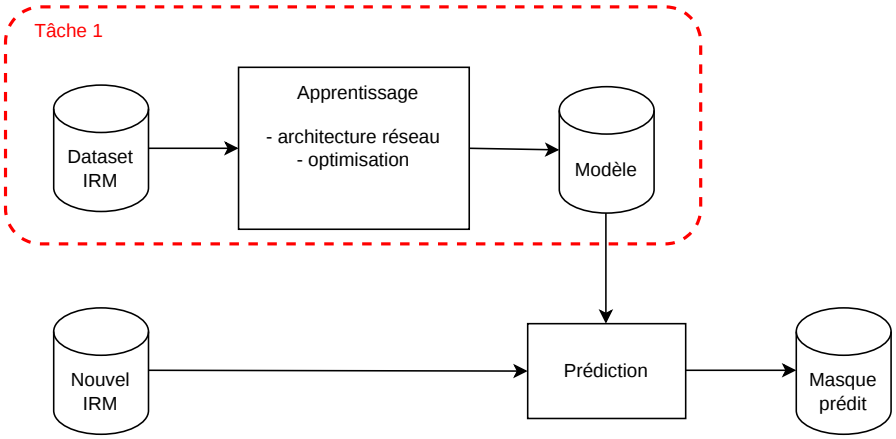


FIGURE 2 – *Workflow* générique d'une tâche d'apprentissage et identification du travail en tâche T1

5 Première phase, deuxième partie (durée 2h00)

Important : la suite du TP est divisée en deux parties pour les tâches T1 et T2. Ces parties sont **indépendantes** et peuvent être réalisées dans le désordre. Il est conseillé d'accorder un temps identique à chaque partie.

5.1 Tâche 1 : localisation des tumeurs sur les images d'IRM par apprentissage profond (durée 1h00)

5.1.1 Métriques pour la segmentation sémantique

L'apprentissage de la localisation des tumeurs est un problème de segmentation sémantique. Cet apprentissage suppose de pouvoir quantifier le degré de précision avec lequel le masque de segmentation prédit par rapport au masque issu du *dataset*. Le code prévu repose sur l'utilisation de deux métriques distinctes :

- la distance de Dice, qui pour deux masques Y_{true} et Y_{prev} est définie par

$$DD = -DC = -\frac{2\,|Y_{true} \cap Y_{prev}|}{|Y_{true}| + |Y_{prev}|}$$

- la distance de Jaccard, qui pour deux masques Y_{true} et Y_{prev} est définie par

$$DJ = -IoU = -\frac{|Y_{true} \cap Y_{prev}|}{|Y_{true} \cup Y_{prev}|}$$

  où DC est le coefficient de Dice, et IoU est le rapport Intersection sur Union ou similarité de Jaccard.

Par définition, sur des matrices de données binarisées on définit :

- l'intersection par le produit de deux matrices terme à terme,
- l'union comme l'addition de deux matrices terme à terme,
- la norme comme la somme des termes.

Activité 5. Dans le fichier *utils.py* de la librairie *learn*, écrire le code des fonctions *dice_ coefficients* et *iou*.

Activité 6. Charger le script *02_ test_ utils.py*, analyser le code et valider la valeur affichée.

Activité 7. L'objectif est de vérifier le comportement des métriques sur des masques simples non alignés :

- Modifier la fonction *get_ Y_ prev* pour que les masques de test soient identiques à ceux présentés en Figure 3. h est le décalage en hauteur du masque et x sa variation de largeur

- pour $x = 0$, faire varier h (sur toute la plage possible) et afficher sur une figure l'évolution des distances de Dice et de Jaccard en fonction de h,
- pour $h = 0$, faire varier x (sur toute la plage possible) et afficher sur une figure l'évolution des distances de Dice et de Jaccard en fonction de x,
- Analyser l'intérêt de ces métriques pour le problème de détection de tumeur.

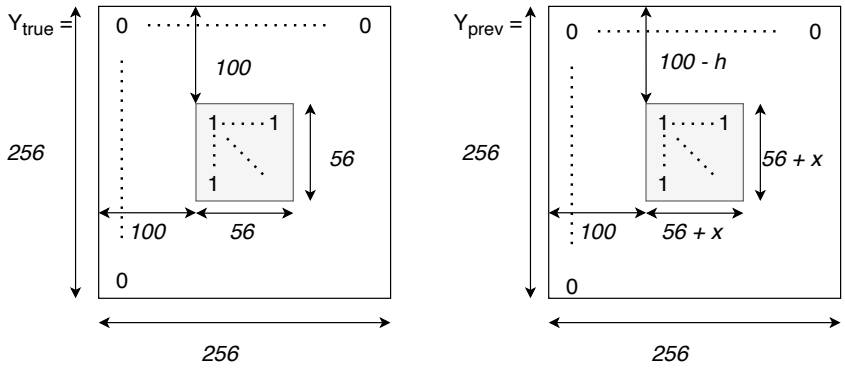


FIGURE 3 – Structure des masques utilisés pour tester les métriques d'apprentissage

5.1.2 Architecture du réseau et apprentissage

Pour cette partie, utiliser le script *03_ test_ learn.py*

Activité 8. L'architecture de réseau à apprentissage profond utilisée est inspirée d'une structure classiquement utilisée pour la segmentation sémantique : 'U-net'. A partir du code décrivant le modèle avec l'API *tensorflow* dans le fichier *unet.py* et la figure 4 :

- compléter les éléments manquants (signalés par un '?', et le schéma des dernières couches) de la Figure 4 et expliciter les annotations numériques de la figure.
- compléter le code python pour compléter le modèle (blocs CONV2, CONV5 et CONV8),

Activité 9. Quel est le principe de fonctionnement de cette architecture ?

Activité 10. Décommenter la ligne appelant la méthode *model_ summary* et lancer le script. En supposant que l'ensemble des paramètres utilise des flottants double précision, quelle taille mémoire occupe le modèle ? Un exemple de modèle sauvegardé est donné dans le fichier *trained_ model.hdf5*

Activité 11. Décommenter le script suivant pour lancer l'apprentissage. Attention l'exécution est très longue : il est souhaitable d'interrompre le processus avec la commande *ctrl+C* une fois que le décompte des *EPOCH* apparaît à la sortie console. A partir de la sortie console donnée en Figure 5, évaluer le temps de calcul nécessaire pour l'apprentissage.

Activité 12. Charger, analyser et exécuter le script *04_ convergence.py*. Tracer sur un diagrame semi-logarithmique en ordonnée (fonction *semilogy* de l'API *matplotlib*) les évolutions de $1-DC$ et $1-IoU$, respectivement les fonctions de perte de Dice et de Jaccard. Que peut-on dire de la convergence ? Combien d'itérations ou *EPOCH* semblent nécessaires pour optimiser le réseau ?

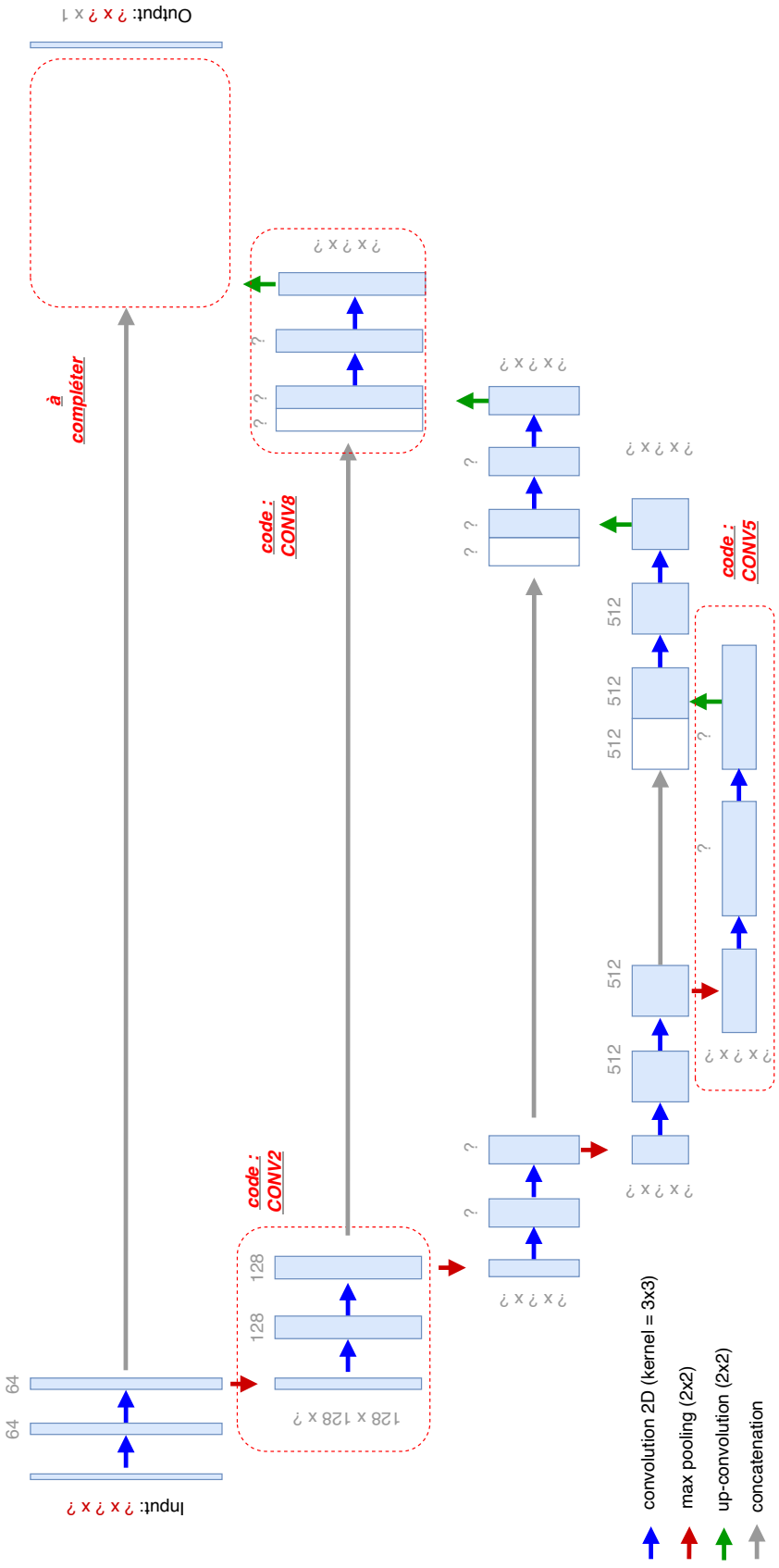


FIGURE 4 – Architecture de réseau U-net utilisée pour la localisation de tumeurs

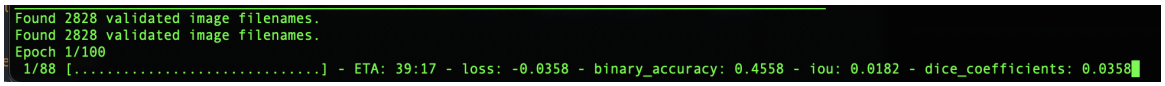


FIGURE 5 – Sortie console du script d'apprentissage pour le premier BATCH de la première EPOCH. L'abréviation ETA signifie *Estimated Time of Arrival*

5.2 Tâche 2 : détection de contours sur les images d'IRM et reconstruction 3D (durée 1h00)

Pour tester cette tâche de manière indépendante de la précédente, on pourra travailler sur les données issues de la base d'apprentissage (répertoire `dataset`), déjà annotée. Il est notamment possible d'utiliser les données contenues dans le sous-dossier *TCGA_CS_5395_19981004*. Le code python à utiliser pour cette tâche est celui de la bibliothèque *viewer*. Remarque : en cas de besoin pour la visualisation des images du répertoire, appeler les examinateurs.

5.2.1 Traitement d'image

Activité 13. Créer un script python qui utilise l'objet *SliceImage*, utiliser les méthodes à disposition pour afficher une image choisie dans le dataset. Quel est le type des données affichées ?

Activité 14. Implémenter la méthode *gamma_correction* qui mute l'objet et effectue une mise à la puissance de l'image : avec le paramètre γ donné en argument, les pixels de sortie p_{out} sont calculés en fonctions des fonction de pixels d'entrée p_{in} par la relation

$$p_{out,i,j} = p_{in,i,j}^{\gamma}, \forall i, j \in [\![0, 255]\!]^2$$

Activité 15. Implémenter la méthode des seuillage *binary_threshold* qui mute l'objet : qui met à 0 les pixels inférieurs à une valeur de seuil, à 1 sinon.

Activité 16. Créer une classe *MaskSlice* spécifique pour les masques (voir Figure 6), et implémenter une méthode de détection de contour de la tumeur. Tester cette classe sur des masques de la base de données d'apprentissage.

Activité 17. Créer une classe *IRMSlice* spécifique pour les images IRM (voir Figure 6), et implémenter :

- une méthode de pré-traitement qui applique une correction gamma ($\gamma = 0.3$) et seuille l'image avec une valeur 0.45,
- une méthode de détection de contour de la boite crânienne, qui utilise la méthode existante *detect_all_contours* et ne conserve que le contour voulu.

Tester cette classe sur des images de la base de données d'apprentissage.

5.2.2 Reconstruction 3D

Activité 18. Créer un script utilisant l'objet *MRIRecorder*.

- Dans la méthode `get_head_points`, ajouter la gestion de la hauteur (dimension z) pour obtenir le nuage de points en 3 dimensions du contour de la boite crânienne.
- S'inspirer de la méthode précédente pour créer la méthode `get_tumor_points` pour obtenir le nuage de points en 3 dimensions du contour de la tumeur.

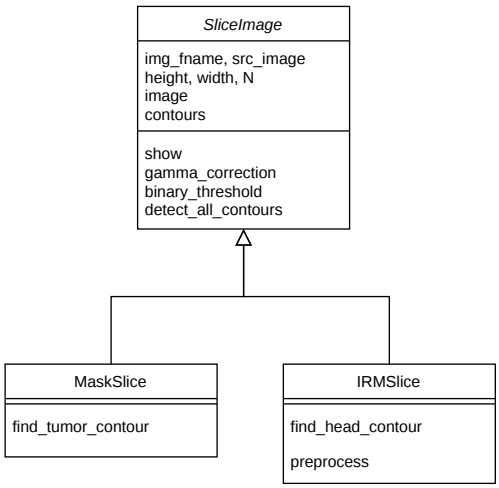


FIGURE 6 – Diagramme des classes à implémenter dans le fichier *slices.py*

Activité 19. Dans le fichier *geom.py*, implémenter une fonction permettant de visualiser une liste de surfaces (structurées ou non).

Activité 20. Rajouter des attributs à la classe *MRIRecorder* permettant de stocker :

- une surface non-structurée et une surface structurée pour la boite crânienne,
- une surface non-structurée et une surface structurée pour la tumeur,

Activité 21. Implémenter une méthode permettant d'extraire les surfaces non structurées de la boite crânienne et de la tumeur à partir des coupes IRM et des masques. Valider le code en ouvrant le fichier *.ply* avec *CloudCompare*.

Activité 22. Par analyse du code de *geom.py*, afficher la surface structurée de la boite crânienne.

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. Ainsi, si la collaboration avec l'entreprise induit un accord de non divulgation (NDA), l'accord est à annexer dans son intégralité au 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.) Le développement scientifique doit être réalisé de manière critique et dépasser le niveau d'une installation ou implémentation d'algorithmes connus ;
- 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 2026, 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.