

INFORMATIQUE ET MODELISATION DES SYSTEMES PHYSIQUES

Durée 4 h

PRESENTATION DU SUJET

Le sujet traitait de la modélisation d'un système de titres de transport sans contact du type Navigo, basé sur une transmission de données par couplage magnétique. Il était organisé en deux parties indépendantes, elles-mêmes constituées de sous-parties indépendantes :

- la première partie (durée conseillée 1h30), après quelques questions qualitatives sur les principes physiques employés, proposait de modéliser le couplage magnétique entre un lecteur RFID et la puce contenue dans un titre de transport, et de montrer que ce couplage permet de transmettre des données binaires par modulation d'amplitude,
- la seconde partie (durée conseillée 2h30), portant sur le programme d'informatique, proposait successivement :
 - de simuler numériquement la démodulation d'amplitude permettant au lecteur de récupérer les données binaires de la puce,
 - de mettre en œuvre quelques algorithmes permettant de détecter et de corriger les erreurs de transmission,
 - de concevoir un petit programme traitant les données reçues pour autoriser ou non le passage d'un voyageur,
 - et, enfin, de récupérer quelques statistiques à partir de la base de données enregistrant les passages aux différents points de validation.

Le poids de chaque partie dans la notation était proportionnel à la durée conseillée.

COMMENTAIRES GENERAUX

Le sujet, par nature pluridisciplinaire, abordait des connaissances en électromagnétisme et en électricité pour sa partie "modélisation", ainsi qu'en ingénierie numérique, algorithmique, programmation et bases de données pour sa partie "informatique". La longueur du sujet ainsi que l'indépendance des différentes sous-parties devaient permettre à tous les candidats de s'exprimer.

En pratique, la plupart des candidats ont abordé les deux grandes parties, mais de façon inégale : la partie "modélisation" a ainsi été assez mal traitée, et de façon très partielle. Les réponses aux questions qualitatives sont souvent de simples paraphrases vagues, tandis que les réponses quantitatives pâtissent souvent d'un manque de rigueur dans la mise en œuvre des modèles. Quant aux applications numériques, elles sont rarement faites, alors qu'un ordre de grandeur correct suffisait à obtenir des points compte tenu de l'interdiction de la calculatrice.

La partie "informatique" a été plus réussie, avec de nombreuses copies abordant les quatre sous-parties et proposant des réponses intéressantes. Néanmoins, des confusions sur le typage des données, sur les entrées et sorties des fonctions, sur la manipulation des listes... sont à déplorer. En outre, quelques candidats ont manifestement fait l'impasse sur les langages informatiques proposés et n'ont

répondu qu'aux questions demandant la lecture d'une courbe ou la mise en œuvre de calculs ou d'algorithmes "à la main".

Dans certaines copies, le candidat récite parfois une formule ou un résultat de cours au lieu de répondre à la question posée. Cela ne rapporte naturellement aucun point.

La présentation des copies est globalement satisfaisante, avec des résultats encadrés, des fonctions lisibles, des indentations respectées (sous Python)... Seuls quelques copies présentent une qualité graphique clairement insuffisante. Pour celles-là, aucune pénalité spécifique n'a été appliquée, mais aucun point n'a été accordé dès lors que la réponse était illisible au point de laisser place au doute. Il en va de même pour les justifications confuses, voire contradictoires, observées tant dans la première partie que dans la deuxième.

COMMENTAIRES SPECIFIQUES A LA PARTIE "MODELISATION"

La première partie du sujet comportait deux sous parties, A et B. la sous partie A abordait des questions qualitatives alors que la sous partie B approfondissait quelques aspects d'un système RFID HF à 13,56MHz.

La première sous partie commençait par une question sur l'expérience historique de Hertz puis venaient des question sur les fréquences utilisées en radio fréquence, sur la portée, le pouvoir de pénétration dans un conducteur et sur la raison de l'utilisation d'une fréquence unique pour les systèmes RFID.

La deuxième sous partie abordait l'émetteur : était amené le champ produit par une bobine circulaire par l'intermédiaire d'un tracé des courbes iso-modules de la valeur efficace du champ magnétique quand celle-ci était alimentée en courant sinusoïdal. De l'analyse de ce document, le candidat devait tracer une courbe de champ à une distance donnée de la bobine en fonction du rayon des spires circulaires et trouver le champ maximal. Venaient des questions sur le flux propre de cette bobine et son inductance propre.

Le TAG était modélisé par une deuxième bobine couplée magnétiquement avec les premières et chargée par une impédance qui pouvait prendre plusieurs valeurs. Le candidat était guidé pour écrire les équations décrivant le circuit électrique de l'émetteur et du TAG, il devait retrouver que la modulation de la charge se répercutait sur la tension d'alimentation de la bobine émettrice associée à un condensateur (rétro-modulation).

Les notions abordées par le sujet couvraient donc une grande partie du programme de première et de deuxième année : électrocinétique, magnétisme, propagation d'ondes électromagnétiques.

Les correcteurs ont remarqué que certains candidats n'ont fait que quelques questions de cette première partie, préférant se consacrer à la deuxième partie informatique, ce qui est dommage car le sujet permettait de récolter des points assez facilement, en particulier sur les question qualitatives du début. Sur ces questions où il faut rédiger quelques lignes, il est toujours préférable d'apporter des réponses concises en utilisant le vocabulaire approprié. On déplore beaucoup de fautes d'orthographe, de grammaire; certaines phrases ne sont même pas finies... Si on ne sait pas, il vaut mieux ne pas répondre que d'écrire des bêtises.

Les applications numériques n'ont été que rarement bien traitées: pourtant elles étaient demandées sous la forme d'ordre de grandeurs. La notion de flux est souvent confondue avec celle de circulation: il est vrai que le sujet demandait un flux écrit à partir d'une intégrale simple. Il faut rappeler aux candidats qu'une intégrale double peut se ramener à une intégrale simple après intégration par rapport à une première variable. et le flux d'une spire confondu avec celui de la bobine. Pire, on donnait la

carte du champ produit par la bobine et on demandait le champ maximal après avoir précisé que la bobine avait cent spires : beaucoup de candidats ont démultiplié ou divisé par cent la valeur. La notion de mutuelle inductance a posé beaucoup de problèmes et un grand nombre de candidat n'est pas allé au-delà des questions faisant intervenir celle-ci. Toutes sortes de relations fausses et inhomogènes furent trouvées dans les copies.

COMMENTAIRES SPECIFIQUES A LA PARTIE "INFORMATIQUE"

Remarques générales

Les candidats avaient le choix entre Scilab et Python, pourvu qu'ils conservent le même langage tout au long de la partie (sauf, bien sûr, pour les questions demandant d'écrire des requêtes SQL). Presque tous les candidats ont choisi Python.

La plupart des candidats ont abordé les quatre sous-parties, qui étaient indépendantes. Néanmoins, rares sont les sous-parties traitées en intégralité ; de nombreux candidats ont préféré passer à la suite plutôt que d'aborder les questions de synthèse situées à la fin des sous-parties, ce qui est dommage car celles-ci étaient souvent bien rétribuées et pouvaient souvent être abordées indépendamment des réponses aux questions précédentes.

Les questions demandant d'écrire une fonction définissaient précisément le type et la valeur des entrées et sorties attendues. Cela permettait notamment d'utiliser ces fonctions dans les questions suivantes, même si elles n'avaient pas été définies. Malheureusement, trop de candidats n'accordent pas une attention suffisante à ces instructions, et confondent notamment "renvoyer" ou "retourner" (fonction return en Python) avec "afficher à l'écran" (fonction print). D'autres se trompent sur les entrées, voire oublient de définir formellement une fonction (par l'instruction def sous Python ou fonction sous Scilab). Rappelons que la cohérence d'une fonction et de son en-tête est essentielle en informatique...

La manipulation des listes est parfois mal maîtrisée, ce qui conduit à des erreurs regrettables. Ainsi, lorsqu'une liste est vide (par exemple si elle a été créée par $T=[]$), il n'est pas possible d'affecter une valeur à un de ses éléments ($T[0]=10$) car cet élément n'existe pas ; il faut lui *ajouter* un élément ayant cette valeur, avec une commande du type $T.append(10)$ (pour un ajout en fin de liste). On note également des confusions entre l'élément $T[i]$ et l'indice i , confusions heureusement rares.

L'utilisation des modules classiques de Python, tels que math, numpy... était explicitement permise par l'énoncé. Certains candidats en ont fait un usage judicieux ; d'autres ont choisi d'utiliser ces modules sans les maîtriser suffisamment, ce qui les a conduit à des confusions dommageables (par exemple entre les fonctions arange et linspace du module numpy).

Enfin, quelques rares candidats ne connaissent pas les structures algorithmiques de base et ont de grandes difficultés à écrire une boucle fonctionnellement correcte, et ce sans même parler de la syntaxe propre à chaque langage. D'autres confondent les instructions if et while. Naturellement, cette méconnaissance les handicape lourdement pour cette épreuve.

Partie 1 - Simulation numérique de la démodulation d'amplitude par le lecteur

Cette partie proposait de simuler numériquement le comportement d'un montage détecteur d'enveloppe, puis d'interpréter graphiquement des résultats fournis. Son originalité venait du fait que, le détecteur d'enveloppe comportant une diode, son comportement dépendait d'une variable interne

pouvant prendre deux états. Pour chacun de ces états, l'expression de la sortie en fonction de l'entrée était donnée par une équation différentielle linéaire du premier ordre.

La liste des instants de calcul (question 1) a été correctement construite, bien que trop de candidats aient visiblement oublié que les arguments de la fonction range doivent être entiers, et que les expressions T_{\max}/dt et $T_{\max} // dt$ ne sont pas synonymes ! Il en va de même pour la représentation de l'entrée modulée (question 2), en dépit de certaines réponses incomplètes ou fausses (avec notamment des confusions entre "période de la porteuse" et "pas de temps" malgré le graphe fourni). La méthode d'Euler explicite (question 3) est connue de tous, mais la suite est moins bien traitée : moins de 20% des candidats arrivent à appliquer la formule d'Euler pour obtenir une condition de changement d'état correcte (question 4) et à programmer correctement l'algorithme de résolution proposé (question 5). Très peu de candidats utilisent une variable interne, bien que cela soit explicitement demandé dans l'énoncé.

Concernant les questions d'analyse, les pas de temps sont correctement identifiés (question 6), mais l'interprétation des différences entre les trois résultats fournis (questions 7 et 8) est souvent vague et peu rigoureuse, car peu de candidats pensent à faire le lien avec l'objectif de la simulation : déterminer si le dispositif est capable ou non de retrouver les deux niveaux d'amplitude d'une sinusoïde modulée. Le théorème de Shannon est parfois invoqué, bien qu'ici il soit nécessaire mais non suffisant : il faut clairement plus de deux points par période pour représenter une sinusoïde de sorte à pouvoir extraire précisément son enveloppe ! Rappelons aux futurs candidats qu'en tant qu'ingénieurs, ils auront à interpréter des simulations numériques, et qu'ils devront donc s'assurer que celles-ci utilisent des données d'entrée suffisamment réalistes compte tenu de l'objectif visé.

Partie 2 - Vérification de l'intégrité des données et correction des erreurs

Cette partie proposait d'analyser et de programmer quelques algorithmes permettant de vérifier qu'une donnée a été correctement transmise : adjonction d'un bit de parité et codage/décodage de Hamming (7,4). Les données étaient représentées par des listes d'entiers valant 0 ou 1, ce qui conduisait à une manipulation aisée des bits. Il s'agit globalement de la partie la mieux traitée.

Le principe du bit de parité (question 9) a été compris de presque tous les candidats, ainsi que la mise en œuvre de son calcul (question 10). Les limites de son utilisation comme test d'intégrité (question 11) ont été à peu près comprises, mais peu de candidats répondent précisément, alors qu'il suffisait de donner un exemple avec un nombre pair d'erreurs puis de remarquer que, même s'il n'y en a qu'une, le test ne donne pas sa position...

Le codage de Hamming (question 12) est programmé correctement par plus de la moitié des candidats, de même que le décodage et la détection d'une erreur (question 13). La correction pose en revanche davantage de problèmes : ainsi, certains candidats testent tous les cas possibles et perdent ainsi un temps précieux (ou des points lorsqu'ils ne traitent qu'un seul cas...), au lieu de convertir simplement la position du bit faux en décimal (l'algorithme proposé la donne directement en binaire). L'étude du comportement de l'algorithme en cas de double erreur (question 14) est rarement menée jusqu'à la conclusion, de même que la proposition d'une solution pour y remédier (question 15) qui donne souvent lieu à des réponses laconiques et incomplètes, voire fantaisistes.

Partie 3 - Utilisation des données de la puce pour autoriser ou non le passage

Cette partie proposait d'extraire des données contenues dans un fichier texte à partir d'un exemple fourni, puis de les traiter et enfin de proposer une fonction autorisant ou non le passage à partir de

règles fournies. Elle nécessitait d'effectuer des manipulations simples sur des données de types variés (chaînes de caractères, listes, entiers ou flottants).

L'identification des types et valeurs des premières données (question 16), qui permettait aux candidats de s'appropriier le script proposé, a été globalement correcte. La récupération des données des passages (question 17) a été moins bien menée ; ainsi, de nombreux candidats ont oublié de convertir les chaînes de caractères en entiers, ou l'ont fait incorrectement, alors que cela était explicitement demandé *et* que l'exemple proposé montrait comment faire. Plus surprenant, un nombre étonnant de candidats se trompent dans l'écriture d'une fonction comparant deux dates (question 18) : leur solution compare les mois alors que les années sont différentes ! (La comparaison de listes donnait ici le résultat escompté en deux lignes ; quelques candidats astucieux ou chanceux y ont pensé...)

Le calcul du nombre de secondes séparant deux horaires (question 19) est mieux mené. Enfin, l'écriture de la fonction autorisant ou non le passage (question 20), fortement rétribuée, est bien menée par un petit nombre de candidats, mais trop peu de copies ne comportent que des réponses partielles (seul un ou deux cas de refus sont traités) ou fausses (confusion entre retour et affichage à l'écran, contresens sur certains cas...).

Partie 4 - Exploitation des données enregistrées par le système

Cette partie nécessitait simplement d'écrire deux requêtes SQL sur une base de données constituée de trois tables. Elle a été peu abordée, et peu de requêtes sont correctes, aussi bien syntaxiquement que sur le fond ; en particulier, les jointures (JOIN... ON...) ne semblent connues et maîtrisées que d'une minorité de candidats.

CONSEILS AUX FUTURS CANDIDATS

On conseille aux futurs candidats de :

- bien connaître les structures algorithmiques de base (for, while, if...else)
- s'entraîner un minimum à la manipulation des listes et des chaînes de caractères,
- savoir respecter scrupuleusement le cahier des charges d'une fonction : "renvoyer" ou "retourner" n'est pas synonyme de "afficher à l'écran" !
- prendre garde aux types des données et notamment, sous Python, à la différence entre entiers et flottants,
- savoir analyser les résultats d'une simulation numérique, et notamment l'influence des différents paramètres ;
- s'entraîner à la formulation de requêtes SQL simples (SELECT... FROM... WHERE) avec utilisation éventuelle de jointures.