

Modélisation d'une MCC – Méthode d'Euler

1 PRESENTATION SIMULATION D'UN MCC

On rappelle les équations du moteur à courant continu :

(1)	$u(t) = e(t) + Ri(t) + L \frac{di(t)}{dt}$	Equations électriques du moteur à courant continu
(2)	$e(t) = k_e \omega(t)$	
(3)	$c_m(t) = k_c i(t)$	
(4)	$c_f(t) = f \omega(t)$	Couple de frottement fluide
(5)	$c_m(t) - c_f(t) - c_r(t) = J \frac{d\omega(t)}{dt}$	Principe fondamental de la dynamique

- $u(t)$: Tension d'entrée aux bornes du moteur (V)
- $e(t)$: Force contre électromotrice (V)
- $i(t)$: Intensité (A)
- $\omega(t)$: Vitesse de rotation du moteur ($rad.s^{-1}$)
- $c_m(t)$: Couple moteur ($N.m$)
- $c_f(t)$: Couple de frottement ($N.m$)
- $c_r(t)$: Couple résistant ($N.m$)
- L : Inductance de la bobine (H)
- f : Coefficient de frottement visqueux ($N.m.s.rd^{-1}$)
- J : Inertie équivalente en rotation de l'arbre moteur ($Kg.m^2$)
- R : Résistance électrique du moteur (Ω)
- k_e : Constante de force électromotrice ($V.rad^{-1}.s$)
- k_c : Constante de couple ($N.m.A^{-1}$)

$$R = 0,45 \Omega \quad ; \quad L = 0,0046 H \quad ; \quad k_e = 0,169 V.rad^{-1}.s \quad ; \quad k_c = 0,17 N.m.A^{-1}$$

$$f = 0,01 N.m.s.rd^{-1} \quad ; \quad J = 0,01 Kg.m^2 \quad ; \quad c_r = 0 Nm$$



Question 1: Mettre en forme ces équations pour une résolution Euler de variables i,w

Question 2: Programmer la résolution Euler explicite des équations pour un échelon de tension de 10V sur 1 secondes avec 1000 points (on ne manipulera que des listes)

Remarque : Veiller à ne pas cumuler d'erreurs d'arrondis dans le calcul de t (ne pas écrire $t += dt$)

Question 3: Tracer l'évolution de la vitesse moteur et de son intensité en fonction du temps

Pour la suite, on considère que la tension de commande du moteur u est une variable globale.

Question 4: Créer la fonction MCC(V0,T0,T1,dt) prenant en argument la liste d'état initial du MCC $V0=[i0,w0]$ (intensité, vitesse), les temps $T0$ de début et $T1$ de fin de résolution et dt le pas de temps de la résolution d'Euler, et renvoyant les listes Lt , Li et Lw des temps, intensités et vitesses moteur

Question 5: Déterminer le nombre de points à utiliser pour obtenir une solution assez fiable sans que les calculs soient trop longs

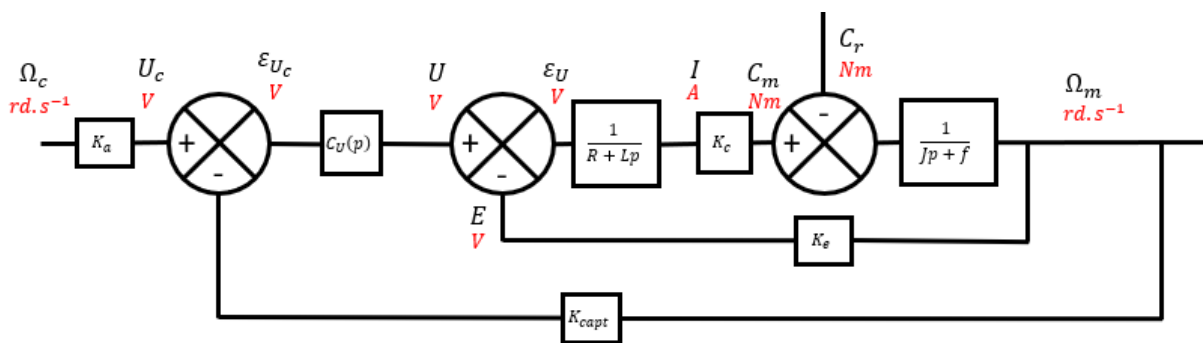
Question 6: En utilisant MCC, déterminer le pas de temps de simulation dt s permettant de supposer que le résultat de la résolution discrète d'Euler est comparable à une résolution continue

Question 7: Réaliser la résolution précédente en utilisant « odeint »

Question 8: Réaliser le modèle Matlab du MCC et comparer les résultats obtenus

2 SIMULATION D'ASSERVISSEMENT DU MCC

On souhaite maintenant mettre en place un asservissement en vitesse de rotation du moteur. On propose le schéma bloc suivant :



On suppose que tout ce qui a été créé dans l'exercice 1 existe et on fixe $dt = 2 \cdot 10^{-4}$ s.

On donne :

$$\omega_c = 1 \text{ rad.s}^{-1} ; \quad K_{capt} = 10 \text{ V.s.rad}^{-1} ; \quad K_a = K_{capt} ; \quad C(p) = 1$$

Les objectifs de ce travail sont doubles :

- 1 : Réaliser la simulation d'un asservissement comme le fait XCOS par exemple, puis dans le dernier exercice, programmer des correcteurs numériques afin de savoir les implémenter sur une carte de contrôle d'un asservissement réel
- 2 : Etudier de manière numérique l'influence de la période de fonctionnement d'une carte de contrôle d'un asservissement numérique d'un système réel (en supposant que notre résolution d'Euler s'apparente à la réponse d'un système continu analogique). Le mieux serait évidemment de réaliser ce travail sur un système analogique réel, mais nous obtiendrons avec la simulation des résultats intéressants

Dans la suite, on utilisera MCC de la question 4 et non odeint.

2.1 Objectif 1 – Simulation numérique d'un asservissement

Dans cette partie, on va utiliser la fonction MCC de la manière suivante :

$$MCC(V0, t, t + dt, dt) ; \quad dt = dt$$

Ainsi, chacun de ces appels réalise le calcul d'un pas de temps d'évolution du MCC avec la tension de commande u (variable globale) modifiée dans la bouche de l'asservissement.

A chaque pas de temps dt , l'asservissement doit :

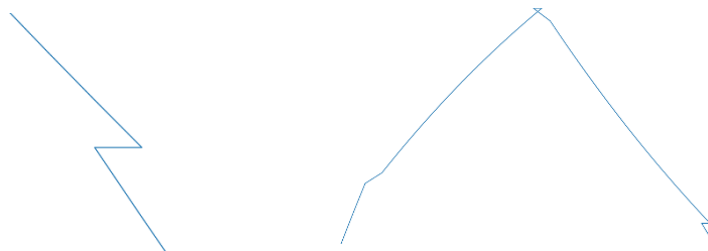
- Mettre à jour la liste $V=[i,w]$ de laquelle le moteur va partir
- Mettre à jour le temps
- Calculer le retour r et l'écart eps
- Mettre à jour la tension u (variable globale pour MCC) à bien mettre AVANT MCC
- Stocker les données associées à la tension u imposée (t, u) dans les listes Ltu et Lu
- Simuler l'évolution du MCC avec cette consigne u depuis l'état V entre les temps t et $t+dt$ avec un pas de temps $dts=dt$
- Compléter les listes Lt, Li, Lw des données de simulations obtenues par paquets par la simulation du MCC It, li et lw

Remarque : Si vous avez veillé à éviter les problèmes de cumul d'erreurs d'arrondis, dans cette partie, les listes Ltu, Lu ont la même dimension que les listes Lt, Li et Lw . Toutefois, je vous recommande de les traiter comme si elles n'avaient pas les mêmes dimensions, ce qui sera le cas dans l'objectif 2.

Question 9: Mettre en place le code permettant de réaliser la simulation de l'asservissement sur 0,2 seconde

Question 10: Tracer l'évolution de l'intensité et de la vitesse du moteur au cours du temps et les comparer à une simulation du même système réalisée sur XCOS

Attention : si vous obtenez des courbes qui présentent des retours en abscisses comme ceci :



Cela veut dire que la résolution MCC réalise un pas de temps de trop. Ainsi, alors qu'elle doit finir à $t+dt$ exactement, elle finit environ à $t+dt+dts$. Origines :

- Programmation de la méthode : En présence d'un while dans Euler, il faut une condition du type $t < t_{fin}$ et non $t \leq t_{fin}$
- Cumul d'erreurs d'arrondis : Dans la programmation d'Euler, il faut incrémenter le temps par la méthode $t = t_0 + i*dt$ et non $t += dt$ qui va cumuler les erreurs d'arrondis sur dt . En appelant MCC entre t et $t+dt=t+n*dts$, $t = t_0 + i*dt$ (dt étant alors égal à dts) sera forcément égal à $t+dt$ au cours de la résolution d'Euler car i et n sont entiers 😊

2.2 Objectif 2 – Effets d'un asservissement numérique sur un système réel

Dans cette partie, on va utiliser la fonction MCC de la manière suivante :

$$MCC(V0, t, t + dt, dts) \\ dt = n * dts \quad n \in \mathbb{N}^*$$

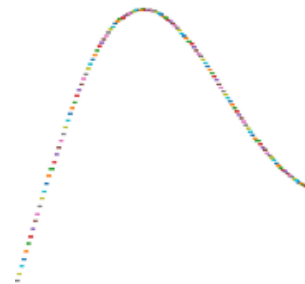
On appelle T_e la période de fonctionnement de la carte de commande du système de traitement des informations. On aura donc $dt = T_e$.

Ainsi, chacun de ces appels réalise le calcul de n pas de temps d'évolution du MCC avec la tension de commande u (variable globale) supposée constante dans l'intervalle de temps $[t, t + dt]$.

Même si cela n'est pas vrai, on suppose que l'appel de la fonction MCC permet de simuler l'évolution d'un MCC réel analogique. On peut alors, comme dans un système réel asservi numériquement, mesurer sa vitesse de rotation ω et envoyer une nouvelle tension de commande u constante, toutes les T_e secondes.

Question 11: En adaptant légèrement le code réalisé dans l'objectif 1, simuler l'asservissement du MCC pour des valeurs de n dans $[1,10,20,30]$ – On tracera la tension de consigne du moteur sous la forme d'une courbe en marches d'escaliers

Cela devrait ressembler à cette image :



Question 12: Conclure sur l'effet de la fréquence de traitement de la carte de commande sur le comportement du système

Remarque importante : Nous avons programmé une résolution d'Euler à pas de temps dt_s pour l'évolution du MCC et prétendu qu'elle représentait l'évolution du MCC analogique réel. C'est faux. Si vous diminuez dt_s , vous verrez que le comportement observé quand n augmente est retardé. Si la conclusion que vous avez donnée reste vraie, rien ne remplacera un système analogique pour étudier l'influence de la période de traitement de la carte de commande. **Dans la pratique, on montre que si la période de traitement T_e de la carte de commande est plus petite que $\frac{\tau}{10}$ avec τ la constante de temps du système asservi, la réponse analogique et la réponse numérique sont comparables.**

3 CORRECTEURS NUMERIQUES

Si nous n'avions qu'à simuler un asservissement sans imaginer implémenter de correcteur numérique dans un système réel, on pourrait reprendre l'exercice 1 de ce sujet et ajouter une 3^e équation représentant le correcteur. Toutefois, l'objectif de cette partie est de vous apprendre à implémenter un correcteur numérique sur un système réel. Quoi de mieux donc, de savoir l'implémenter, et en plus de pouvoir le vérifier avec une simulation ? Voilà notre objectif.

Pour une résolution numérique optimale, on fixe à nouveau $dt = dt_s = T_e$, soit $n = 1$.

A l'itération k de la boucle d'asservissement, on appelle :

- e_k l'écart au comparateur
- u_k la tension de commande du moteur élaborée par le correcteur

Sans correcteur, $u_k = e_k$.

ATTENTION : il faudra dans votre code noter u , et non u_k , car la fonction d'Euler utilise la tension u comme variable globale.

On propose les correcteurs suivants :

P	PI	Avance de phase
K_p	$K_p + \frac{K_i}{p}$	$\frac{1 + aTp}{1 + Tp} ; a > 1$

Question 13: En utilisant la méthode des équations aux différences, déterminer pour ces trois correcteurs l'expression de u_k en fonction de u_{k-1} , e_k et e_{k-1}

Question 14: Proposer les trois fonctions python CP, CPI et CAP des correcteurs numériques étudiés en précisant les arguments nécessaires

Attention, danger : la fonction MCC fait appel à la variable globale u . Il faut donc veiller à appeler la tension u , et non u_k par exemple

Question 15: En utilisant le code de l'exercice 2, ajouter ces 3 correcteurs à l'asservissement numérique réalisé et vérifier leur fonctionnement avec Matlab.