

Institut Supérieur de l'Électronique et du Numérique

AUTOMATIQUE CONTINUE

Fonctionnement du TD/TP

Les travaux pratiques se déroulent sur **3 heures en trinôme ou binôme**. A l'issue de ces trois heures, un compte rendu est à rendre à l'enseignant vous encadrant. Ce compte rendu fera l'objet d'une note.

Contenu du TP

Ce TP a pour but de vous faire connaître les bases de Matlab et de Simulink dans le cadre de l'automatique continue. Nous allons réaliser, dans ce TP, l'étude d'un asservissement en position. Il aborde les différentes étapes d'une partie du travail à réaliser sous forme décomposée :

- identification du processus
- analyse temporelle
- analyse fréquentielle
- amélioration du comportement du système.

Dans le cadre de ce TP, nous allons définir le fonctionnement du pilote automatique d'un navire, en s'intéressant au safran et à la motorisation du navire.

1. Identification du système

Ne pouvant directement obtenir les équations différentielles régissant ce système, nous faisons un essai en boucle ouverte pour « voir comment il fonctionne » avec en entrée un échelon unitaire. A partir de cet essai, nous pouvons en déduire, sans la moindre équation différentielle, la fonction de transfert de notre système (voir chapitre 7 du cours). Cette opération s'appelle l'identification du système.

L'essai en boucle ouverte nous permet d'obtenir le tableau de valeurs ci-dessous :

t en s	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
s(t)	0	0.0088	0.0312	0.0626	0.1002	0.1419	0.1863	0.2326	0.2801
t en s	0.9	1	1.1	1.2	1.3	1.4			
s(t)	0.3284	0.3773	0.4265	0.4760	0.5257	0.5755			

Nous allons tracer cette courbe à l'aide du logiciel Matlab.

Lancez le logiciel Matlab (voir Figure).

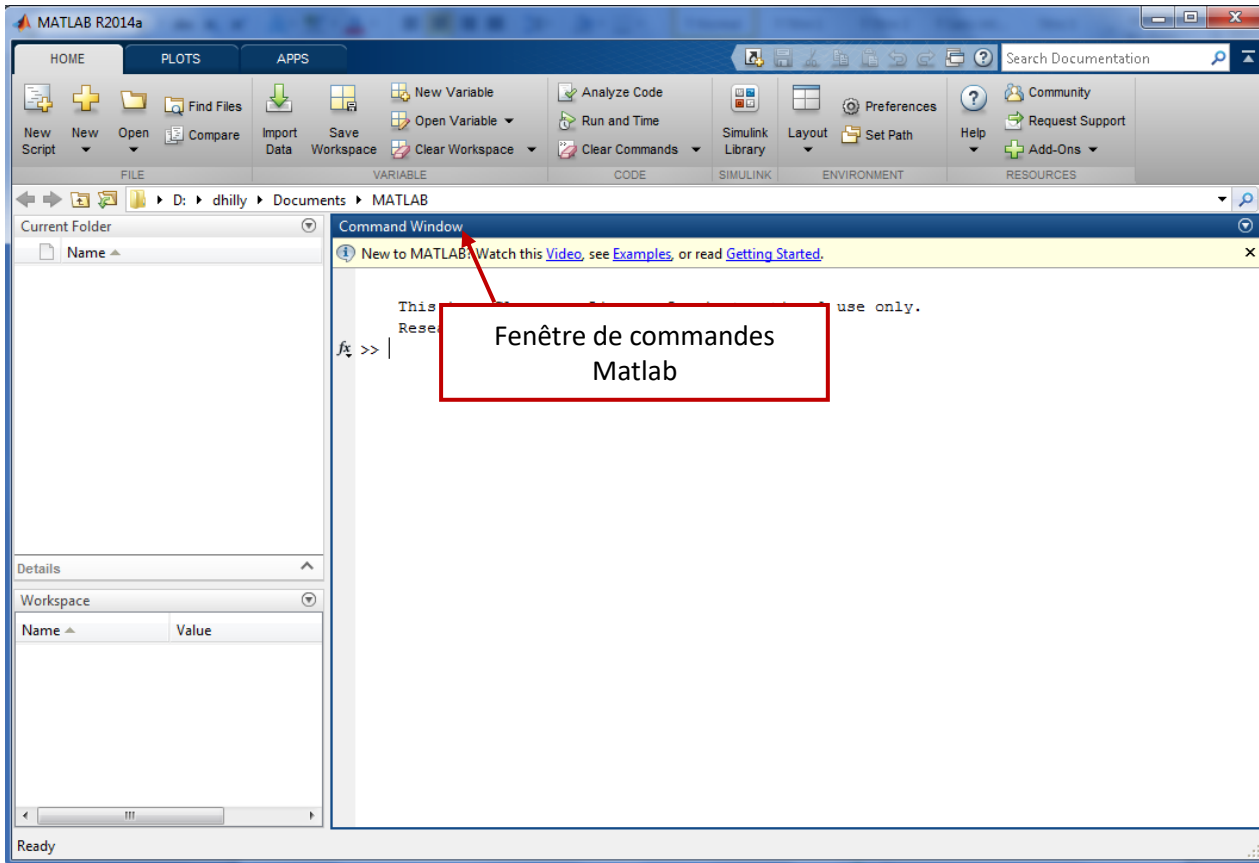


Figure 1 : Logiciel Matlab

Pour tracer la courbe, vous allez créer un script. Dans l'onglet principal **HOME**, cliquez sur **New Script**.

Dans ce script, vous devez mettre les lignes suivantes :

```
t = 0:0.1:1.4 %ça veut dire que t varie de 0 à 1.4 avec un pas de 0.1  
s = [0 88e-4 312e-4 .....] %remplir complètement bien évidemment avec les valeurs données  
                                précédemment  
plot(t,s) % Tracé de la courbe de s=f(t)  
grid on% grille
```


Remarques sur l'écriture des scripts :

% permet de mettre des commentaires dans votre programme

Un ; en bout de ligne de commande permet de ne pas afficher le résultat dans la fenêtre de commande.

N'oubliez pas de sauvegarder votre fichier **.m**. Attention le nom du fichier **.m** ne doit pas contenir des espaces, tirets, symboles, commencer par des chiffres etc...Donner un nom simple à retenir.



Ensuite, vous devez exécuter votre programme, soit en cliquant sur  de la fenêtre d'édition des Scripts ou alors en appelant votre programme dans la fenêtre de commandes de Matlab « *command window* ».

Vous pouvez également rajouter un titre, une légende en abscisse et en ordonnée directement sur la figure avec l'instruction **insert**.

Pour savoir comment une fonction Matlab fonctionne **help** nom de la fonction par exemple :

>> help plot(à faire dans le « *command window* »)

La forme de la réponse $s(t)$ ainsi obtenue fait que nous prenons comme modèle un système du premier ordre possédant un intégrateur dont la fonction de transfert est :

$$H(p) = \frac{K}{p(1 + Tp)}$$

Il nous faut ensuite, à partir de la courbe, définir les valeurs de K et de T . Cette méthode est explicitée dans le polycopié du cours et rappelée ci-dessous :

1. Tracer l'asymptote à la courbe qui coupe l'axe des abscisses au point A
2. Le temps entre l'origine et le point A détermine la valeur de T
3. Le gain K est donné par la pente de l'asymptote

Question 1 : Imprimez la courbe en format paysage et donnez votre fonction de transfert, appelé H(p), ainsi identifiée

Vous allez pouvoir désormais valider votre modèle en traçant la réponse du système que vous avez obtenue après identification (en mettant en entrée un échelon unitaire), avec la réponse indicielle initiale du système soit $s(t)$.

Pour cela, vous devez reprendre votre script initial, vous effacez la commande **plot(t,s)** que vous aviez mise et vous rajoutez les lignes suivantes :

K=...;	(Mettre la valeur de K que vous avez trouvée par identification)
T=...;	(Mettre la valeur de T que vous avez trouvée par identification)
num = K;	
den = [T 1 0];	(den = dénominateur de la fonction de transfert = vecteur donné par puissance de p décroissante – Le dénominateur de votre fct de transfert est Tp^2+p)
H2p = tf(num,den);	(help tf pour son fonctionnement – permet de créer la fonction de transfert)
u = ones(1,15);	(Création du vecteur échelon unitaire de valeur 1)
y = lsim(H2p,u,t);	(Obtention de la sortie de la fonction de transfert identifiée en mettant en entrée un échelon unitaire)
plot(t,s,'b',t,y,'r');	(Tracé sur le même graphe de la courbe réelle $s(t)$ et de la courbe identifiée $y(t)$ par le modèle)

Remarques :

% permet de mettre des commentaires dans votre programme

Un ; en bout de ligne de commande permet de ne pas afficher le résultat dans la fenêtre de commande.

Vous obtenez alors les 2 courbes (réelle et modèle sur le même graphe). Pour que votre modèle soit validé, il faut que l'erreur relative entre le modèle et la courbe réelle soit en moyenne inférieure à 5 %. Si ce n'est pas le cas, vous devez rectifier les valeurs de K et de T dans votre fonction de transfert.

Pour cela, nous allons calculer le vecteur ligne contenant les erreurs relatives en pourcentage et afficher ces valeurs en fonction du temps.

Vous devez rajouter les lignes suivantes :

erreur = abs(s-y')./s *100 (y' représentant la transposée de y, ./ permettant de faire une division élément par élément)
plot(t, erreur) (représente le tracé de l'erreur en fonction du temps)

Question 2 : En étudiant le tracé de l'erreur, pouvez-vous valider votre modèle ?

2. Analyse temporelle et analyse fréquentielle du système

En série avec H(p) mais placé en amont, nous trouvons l'actionneur G(p) qui permet de piloter le safran. Nous avons :

$$G(p) = \frac{1}{1 + 0.1p}$$

Nous allons maintenant dessiner le schéma bloc de ce système sous simulink. (**ATTENTION ! La notation anglo-saxonne de la variable de Laplace "p" est "s" pour Matlab.**)

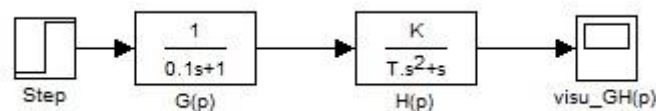


Figure 2 : Schéma bloc à réaliser

Pour créer votre schéma sous simulink vous devez sélectionner dans le menu initial **HOME>> NEW>> Simulink Model**, puis cliquez sur Blank Model. Vous obtenez alors la Figure 3.

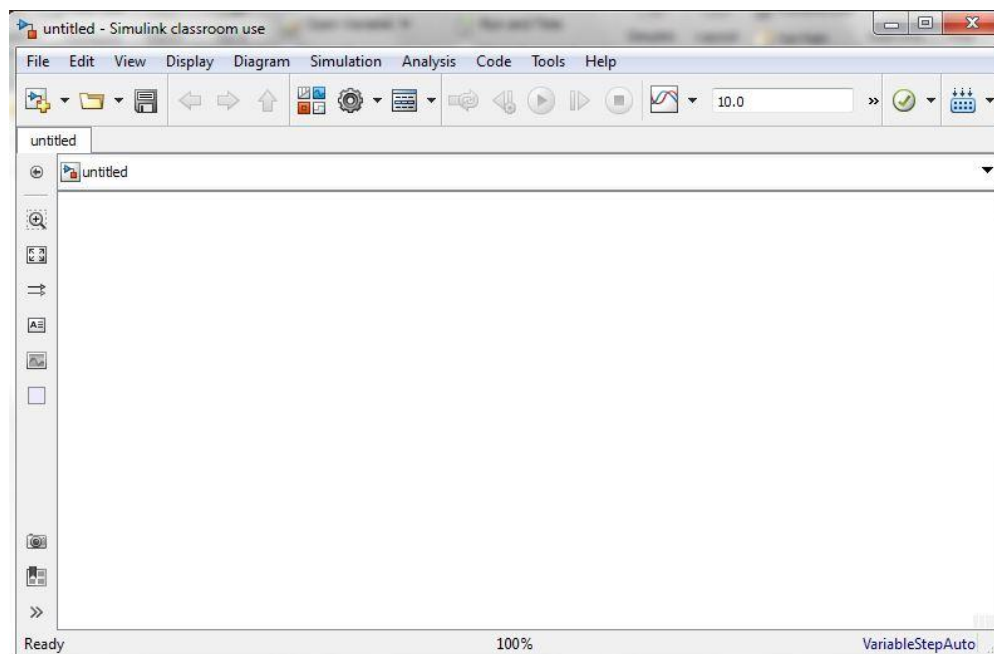



Figure 3 : Fenêtre pour créer un schéma-bloc avec simulink

Pour insérer les éléments, il faut faire des glisser-déposer à partir des éléments de la bibliothèque que vous

obtenez en cliquant sur l'icône .

Création du schéma bloc :

1 Bloc échelon :

Sélectionnez un **step** dans la bibliothèque **Simulink>>Sources**, et faire un glisser déposer dans votre fenêtre simulink.

Double-cliquez sur le step afin de régler les paramètres (Mettre les paramètres comme indiqués sur la Figure 4).

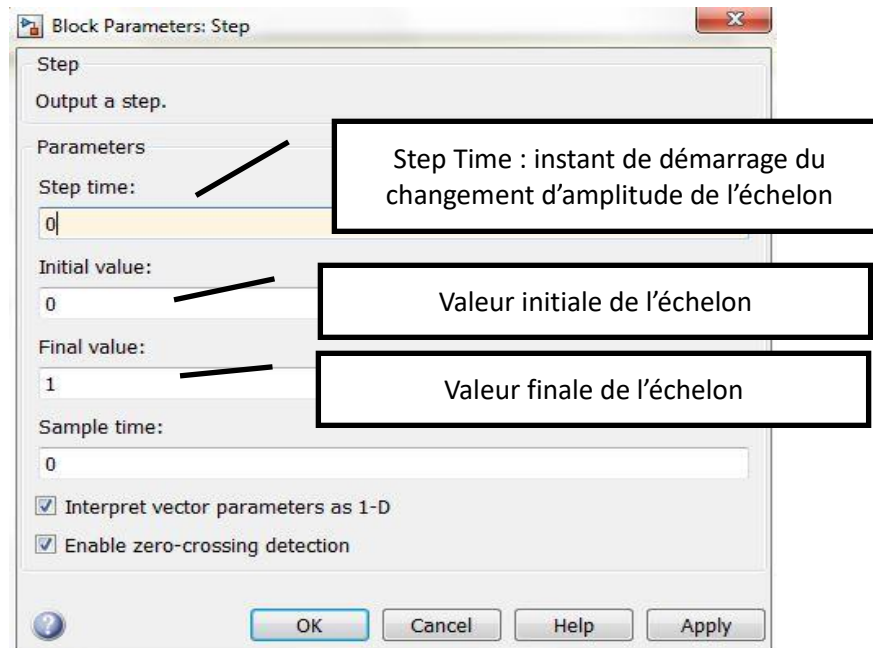


Figure 4 : Réglage des paramètres de l'échelon

2 Bloc Fonction de transfert

Sélectionnez **Transfer fcn** dans **simulink>>continuous**. Double cliquez sur l'icône afin de régler vos paramètres. Vous devez mettre les valeurs obtenues pour K et T comme indiqué sur la Figure 5.

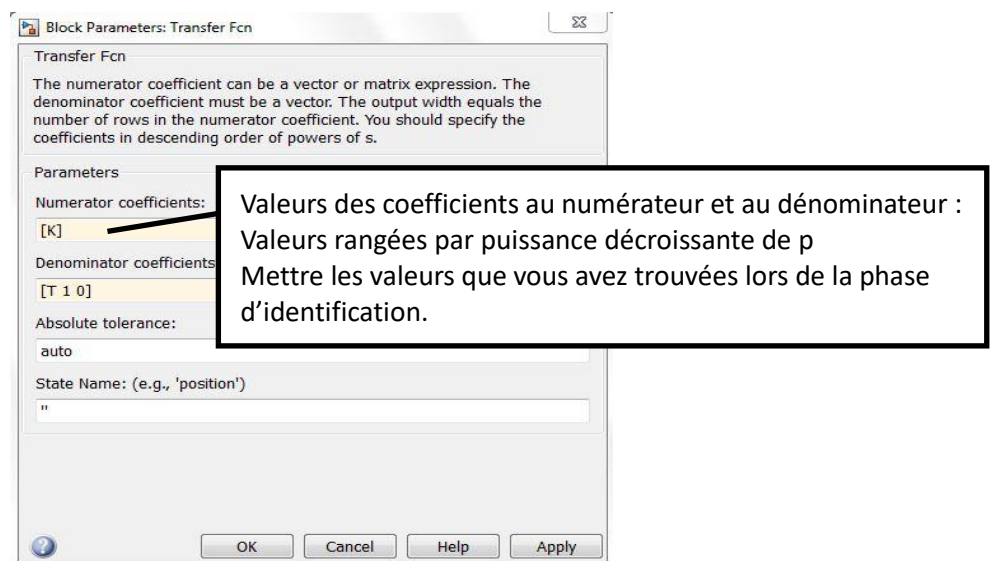


Figure 5 : Réglage des paramètres de la fonction de transfert

3 Bloc Scope

Sélectionner **Scope** dans **simulink>>Sinks**.

Il ne vous reste plus qu'à relier les différents blocs entre eux et vous obtenez le schéma de la Figure 6.

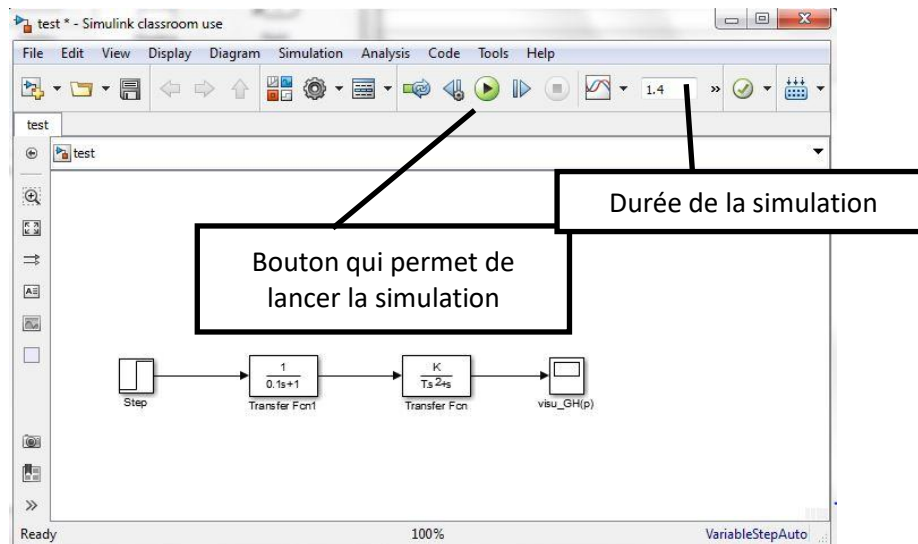



Figure 6 : Schéma Bloc créé sous simulink

Vous devez maintenant changer la durée de simulation, mettre 1.4s au lieu de la valeur de 10s donnée par défaut.

Pour pouvoir analyser la réponse temporelle de ce système, vous devez lancer la simulation et double-cliquez sur le Scope.

Dans la fenêtre scope, cliquez sur le bouton suivant  afin d'effectuer un réglage automatique.

Question 3 : Imprimez la réponse obtenue, qui correspond à la réponse indicielle de votre système en boucle ouverte ; celle-ci vous servira à la question 11.

Afin d'observer l'effet produit par ajout de l'actionneur (modélisé par la fonction de transfert $G(p)$) sur le système, vous devez réaliser le schéma représenté sur la Figure 7.

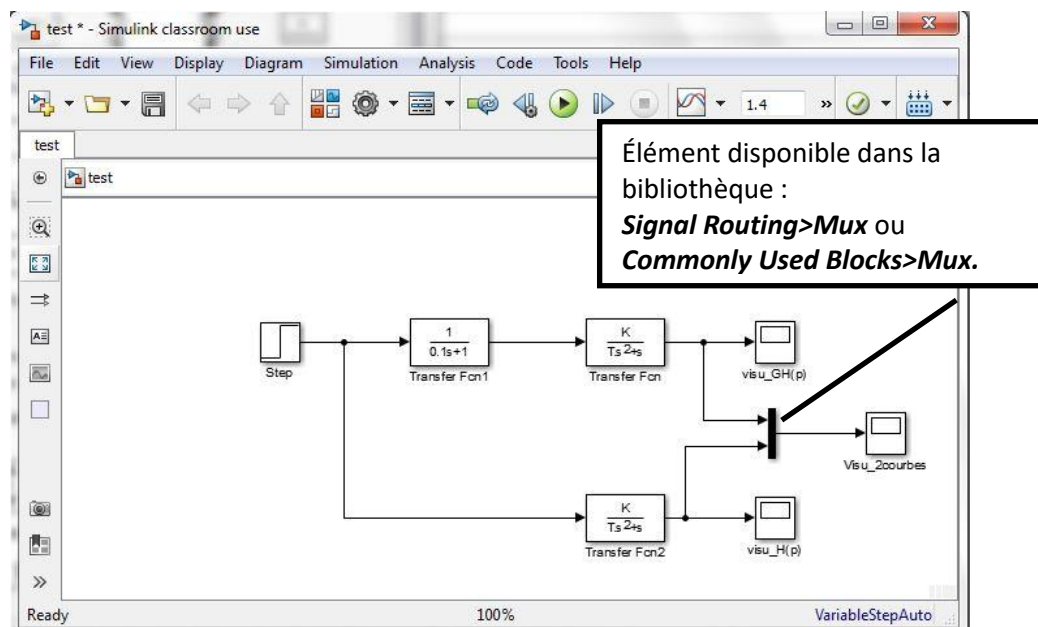


Figure 7 : Schéma Simulink pour réaliser la question 4

Question 4 : En observant le scope de visualisation des 2 courbes, que provoque selon vous (d'un point de vue temporel), l'ajout de l'actionneur modélisé par la fonction de transfert du 1^{er} ordre $G(p)$?

Modifier le schéma bloc précédent afin d'obtenir celui représenté sur la Figure 8 pour effectuer une analyse fréquentielle de notre système.

Pour cela, nous devons définir des points d'entrée et des points de sortie. Pour le point d'entrée, effectuez un clic droit sur le fil de liaison avant la fonction de transfert, choisissez **Linear Analysis Points >> Open-loop Input**.

Pour les points de sortie, vous allez procéder de même en effectuant un clic droit après $H(p)$ et $H2(p)$ et en choisissant **Linear Analysis Points >> Output Measurement**.

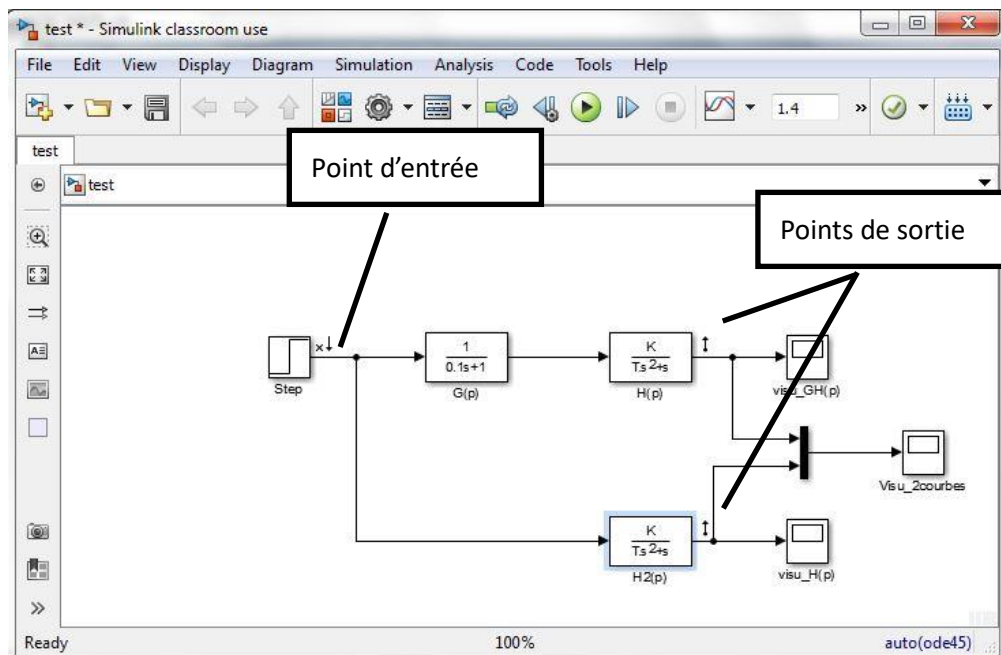


Figure 8 : Schéma simulink avec création des points d'entrée et de sortie

Dans la fenêtre simulink du modèle créé, allez dans l'onglet **Analysis>>Control design>>Linear Analysis**. Vous obtenez alors la fenêtre suivante.

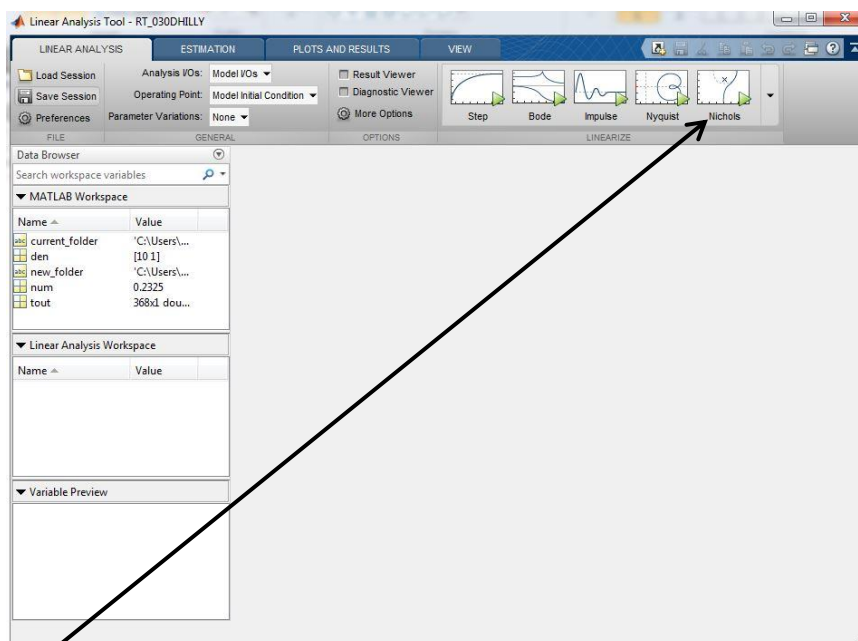


Figure 9 :Fenêtre control design sous simulink

Choisir **NicholsPlot**. Vous obtenez alors le lieu de Black de votre système identifié en Boucle ouverte.

Faire un clic droit sur le lieu et sélectionner grid (pour faire apparaître l'abaque de Black) et régler les axes de votre lieu.

Pour remettre les 2 courbes sur le même graphique, il suffit d'effectuer un clic droit sur les figures et choisir **I/O grouping >>all**.

Pour connaître le gain et la phase des systèmes représentés, il suffit d'effectuer un clic gauche sur la courbe à l'endroit où vous voulez connaître ces valeurs.

Question 5 : Donner pour les deux systèmes :

- les valeurs obtenues pour le module lorsque la phase est à -180°
- les valeurs obtenues pour la phase lorsque le gain est à 0db.

Question 6 : Donner ensuite la marge de gain et la marge de phase pour les 2 systèmes.

Pour le moment, vous avez étudié le système en boucle ouverte pour étudier la stabilité du système asservi avec un retour unitaire. Afin d'étudier la précision du système, nous allons réaliser la boucle de retour sous simulink.

Réalisez sous simulink le schéma de la Figure 10. Pour cela, vous devez rajouter un comparateur (attention il faut le configurer en comparateur) et reboucler l'ensemble.

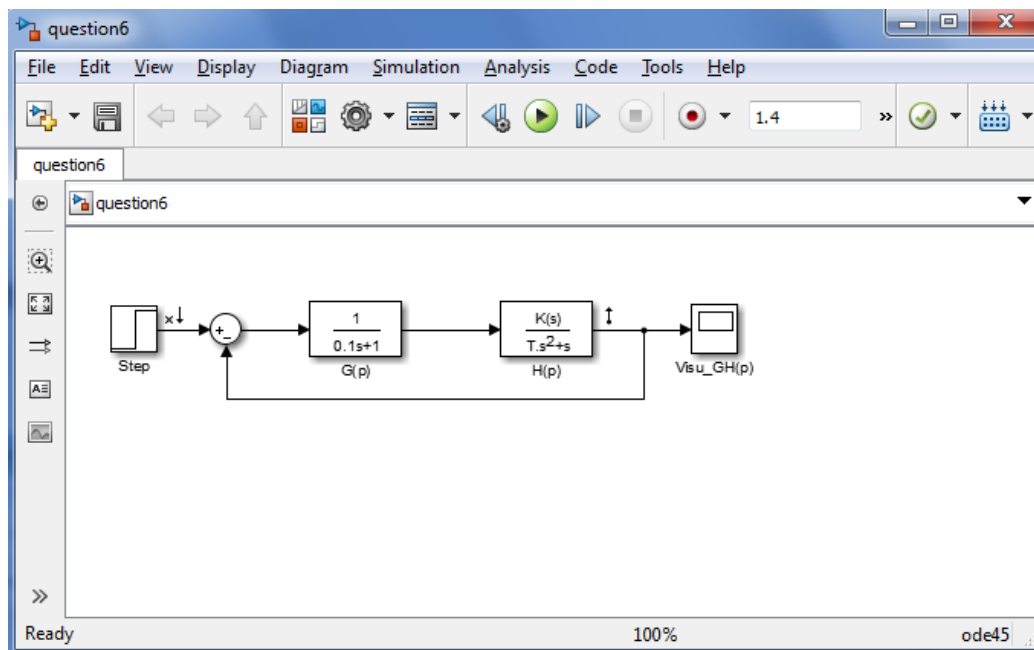


Figure 10 : Schéma sous simulink du système bouclé

Pour analyser la réponse temporelle, vous devez comme tout à l'heure, mettre un point d'entrée et un point de sortie. Relancez **Analysis>>Control design>>Linear Analysis**, choisissez step pour l'affichage

Question 7 : Relever sur cette courbe, en effectuant un clic droit sur la courbe et en sélectionnant **characteristics** toutes les caractéristiques temporelles de cette réponse :

- rise time (temps de montée entre 10% et 90%)
- settling time (temps de réponse à 5%)
- steady state (valeur finale)

Pour changer les valeurs par défaut des caractéristiques temporelles du système, faire un clic droit et sélectionnez propriétés puis options (régler les paramètres comme indiqués sur la Figure 11).

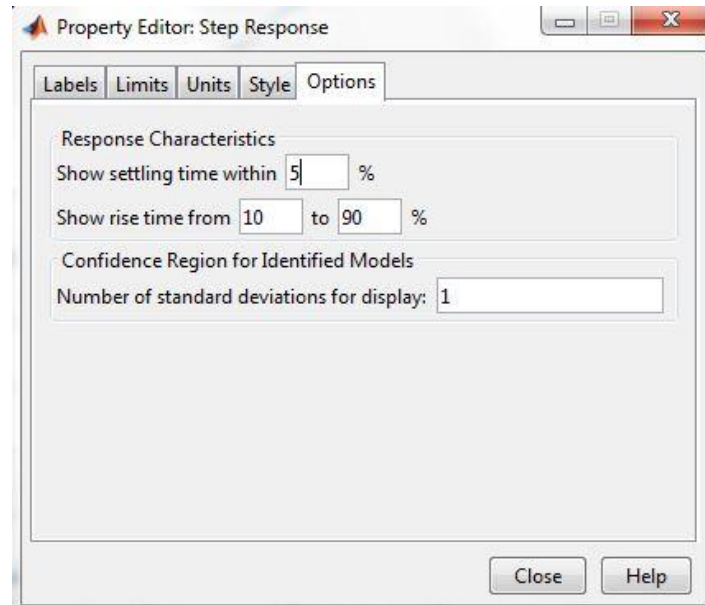


Figure 11 : Fenêtres de réglage des paramètres des caractéristiques de la réponse à un échelon

3. Erreurs en asservissement et en régulation

Nous allons désormais rajouter une perturbation sous Simulink. Nous définissons désormais l'ordre de barre (l'entrée, le step de consigne) comme étant une demande de 10° .

Au bout de 100s, nous avons une perturbation de $+1^\circ$ (résultante d'un courant marin ou du vent par exemple). Nous obtenons le schéma représenté sur la Figure 12.

Réaliser ce schéma en réglant :

- un temps de simulation à 150s,
- un step d'entrée (step time=0s; final value = 10°)
- un step de perturbation (step time=100s, final value= 1°)

Il ne faut pas oublier de modifier la durée de simulation.

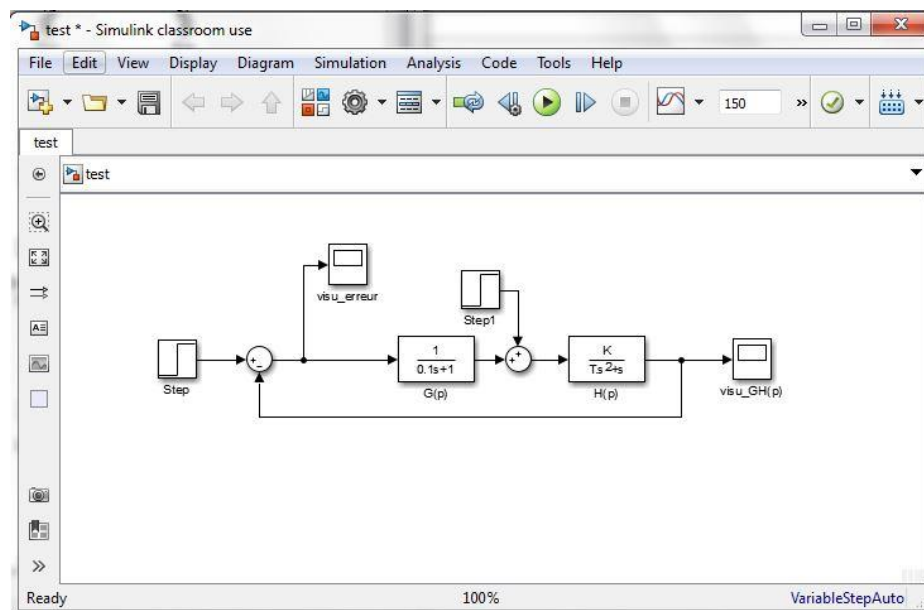


Figure 12 : Schéma Simulink pour étudier les erreurs de régulation et d'asservissement

Question 8 : A partir du Scope :

- Identifier l'intervalle de temps correspondant à l'asservissement et celui correspondant à la régulation
- Donner les valeurs de l'erreur finale en asservissement et en régulation

Question 9 : Valider les valeurs obtenues sur votre scope en calculant ces erreurs finales de manière théorique.

Que concluez-vous sur le fonctionnement de ce système ?

4. Correction du système

Nous allons désormais tenter de corriger le système en rajoutant un gain K_c (correcteur proportionnel) entre le comparateur et l'actionneur (voir Figure 13).

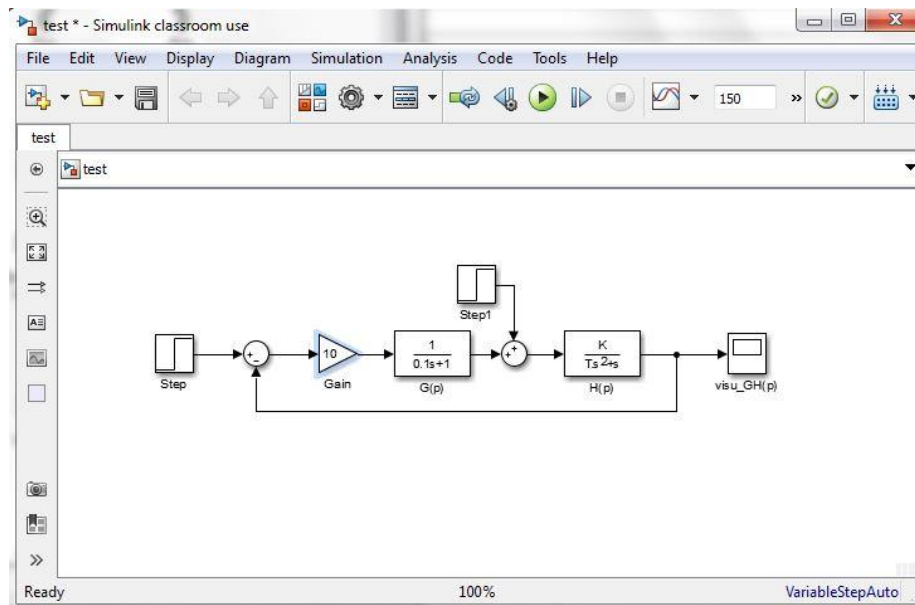


Figure 13 : Schéma bloc du système corrigé avec un correcteur proportionnel

Question 10 : En prenant $K_c=10$, indiquer l'influence d'un gain K_c sur la réponse indicielle du système.

Quelle est la valeur de l'erreur lorsque $K_c=10$?

Quel est le nom de cette correction ?

Déterminer, en modifiant sur le logiciel la valeur du gain K_c , le gain $K_{c_{critique}}$ qui positionne le système à la limite de la stabilité ?

Comment pouvez-vous retrouver cette valeur par rapport à ce que vous avez fait au préalable dans le tp lors de l'analyse fréquentielle de ce système en boucle ouverte ?

Nous allons maintenant remplacer le gain K_c par un correcteur proportionnel-intégrale (PI) parallèle dont la fonction de transfert $C(p)$ est la suivante :

$$C(p) = K_c + \frac{I_c}{p}$$

L'insertion de ce correcteur sous Simulink se fait en introduisant un correcteur PID (**Continuous > PID controller**)

Afin de calculer les paramètres K_c et de I_c nécessaire au correcteur PI, nous allons nous servir de la courbe imprimée à la question 3.

Question 11 : Identifier les valeurs de K_c et de I_c , en vous servant de la méthode pratique présentée dans le cours (chapitre 8) pour déterminer les valeurs du correcteur PID (correcteur construction Parallèle).

Dans le cours, la méthode permet de déterminer le paramètre du correcteur intégral noté T_i alors que Matlab utilise $I \Rightarrow I = \frac{1}{T_i}$.

Insérer le correcteur PID en paramétrant la valeur de K_c en « Proportionnal », I_c en « Integral » et 0 en « Derivative ». Ainsi le correcteur PID devient un correcteur PI.

Question 12: Observer les résultats obtenus. Que pouvez-vous en conclure sur ce correcteur ?

Il est aussi possible d'utiliser une fonction plus directe pour régler les paramètres du correcteur. Il s'agit de l'outil "Tune" que l'on trouve sous les paramètres K, I et D sous la forme d'une icône sur laquelle il faut cliquer. (Cet outil nécessite le module "Simulink Control Design" lors de l'installation de Matlab)

→ Une fenêtre s'ouvre alors avec la réponse indicielle corrigée. Cliquer sur "Automatically Update block parameters". Il est alors possible de modifier le temps de réponse et la valeur du 1^{er} dépassement à sa convenance; les paramètres K et I étant automatiquement modifiés en conséquence.

Question 13: Appliquer cet outil dans le cas présent. Présenter la courbe réglée et les paramètres du correcteur modifié.

Pour finir ce TP, nous allons également simuler le comportement des vagues sur notre navire en rajoutant, comme perturbations, un générateur de fonctions qui fournit des sinusoïdes de fréquences 0,2 Hz dont l'amplitude est réglée à 1 (attention au réglage des paramètres du signal generator voir Figure 15). De plus, nous sommes désormais dans une zone où le courant fait dériver le navire de 5° (perturbation simulée par le step1). Vous devriez avoir le schéma Simulink suivant :

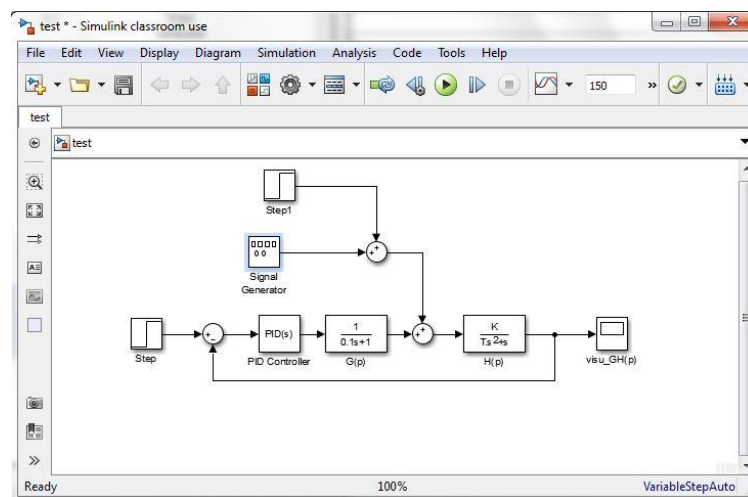


Figure 14 : Schéma bloc du système complet

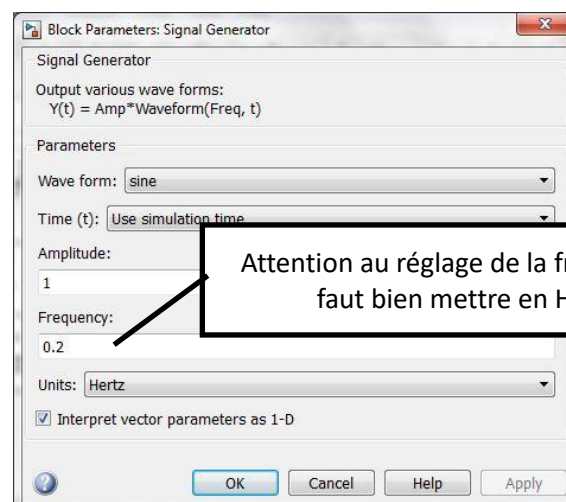


Figure 15 : Réglage des paramètres du bloc signal generator

Question 14 : Analyser le comportement de votre système sans le PID (mettre dans le bloc PID $K_c=1$ et $I_c=0$) puis avec le PID (mettre les valeurs calculées à la question précédente) pour cette nouvelle configuration de l'ensemble.

Utiliser l'outil "Tune" pour améliorer le résultat. Mettre en œuvre éventuellement le paramètre Dérivé "D".

Conclure.