Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Τμήμα Μαθηματικών

Προπτυχιακό Τμήμα Σπουδών

ΘΕΜΑ

Η χρήση του Scilab στην μελέτη των συστημάτων στον χώρο των καταστάσεων

Ειρήνη Άννα Συμεωνίδου (ΑΕΜ: 16275)
Χαράλαμπος Σεργίου (ΑΕΜ: 16934)

Επιβλέπων Καθηγητής: Καραμπετάκης Νικόλαος
Καθηγητής Τμήμα Μαθηματικών Α.Π.Θ.

ΘΕΣΣΑΛΟΝΙΚΗ
Ιούλιος 2020

Περιεχόμενα

Εισαγωγή

Η αναπαράσταση του συστήματος στον χώρο των καταστάσεων θέτει τα θεμέλια για τη σύγχρονη θεωρία ελέγχου. Επιλύει πολλούς από τους περιορισμούς της θεωρίας κλασικού ελέγχουστην οποία χρησιμοποιήθηκαν συναρτήσεις μεταφοράς για να αξιολογηθεί η συμπεριφορά ενός συστήματος κλειστού βρόχου.

Ένα μοντέλο στον χώρο των καταστάσεων (state-space) περιγράφει τη συμπεριφορά ενός δυναμικού συστήματος ως ένα σύνολο συνηθισμένων διαφορικών εξισώσεων πρώτης τάξης (ODE). Εάν ένα δυναμικό μοντέλο περιγράφεται από ODE υψηλότερης τάξης, χρησιμοποιώντας χώρο καταστάσεων, το ίδιο μοντέλο μπορεί να περιγραφεί ως ένα σύνολο συζευγμένων ODE πρώτης τάξης. Οι εσωτερικές μεταβλητές του μοντέλου state-space ονομάζονται μεταβλητές κατάστασης και περιγράφουν πλήρως το δυναμικό σύστημα και την απόκρισή του για ορισμένες εισόδους.

Το πλήθος των μεταβλητών κατάστασης του μοντέλου state-space είναι ίσο με την υψηλότερη τάξη του ODE που περιγράφει το δυναμικό σύστημα. Οι μεταβλητές κατάστασης μπορούν επίσης να οριστούν ως το μικρότερο σύνολο ανεξάρτητων μεταβλητών που περιγράφουν πλήρως το σύστημα. Το σύνολο των μεταβλητών κατάστασης δεν είναι μοναδικό και μπορεί να οριστεί ως προς τις φυσικές μεταβλητές που μπορούν να μετρηθούν ή με όρους μεταβλητών που δεν μπορούν να μετρηθούν άμεσα.

Για ένα δεδομένο μοντέλο state-space, ο αριθμός των μεταβλητών κατάστασης είναι ίσος με τον αριθμό των αρχικών συνθηκών που απαιτούνται για την πλήρη επίλυση του μοντέλου συστήματος. Οι μεταβλητές κατάστασης είναι ακριβώς αυτές οι μεταβλητές για τις οποίες απαιτούνται αρχικές συνθήκες. Ο αριθμός των μεταβλητών κατάστασης είναι ίσος με τη σειρά με το ODE που περιγράφει το σύστημα.

Ένα μοντέλο state-space σχηματίζεται από:

- εξισώσεις κατάστασης
- εξισώσεις εξόδου

Οι **εξισώσεις κατάστασης** έχουν τη γενική μορφή:

$$\dot{x} = A \cdot x + B \cdot u$$

όπου:

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{Bmatrix}_{n \times 1} \quad \text{διάνυσμα κατάσταση}$$

$$u = \begin{Bmatrix} u_1 \\ u_2 \\ \cdots \\ u_m \end{Bmatrix}_{m\times 1}$$ διάνυσμα εισόδου

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}_{n\times n}$$ πίνακας κατάστασης

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix}_{n\times m}$$ πίνακας εισόδου

Οι **εξισώσεις εξόδου** έχουν τη γενική μορφή:

$$y = C \cdot x + D \cdot u$$

όπου:

$$y = \begin{Bmatrix} y_1 \\ y_2 \\ \cdots \\ y_p \end{Bmatrix}_{p\times 1}$$ διάνυσμα εξόδου

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{p1} & c_{p2} & \cdots & c_{pn} \end{bmatrix}_{p\times n}$$ πίνακας εξόδου

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ d_{p1} & d_{p2} & \cdots & d_{pm} \end{bmatrix}_{p\times m}$$ πίνακας άμεσης μετάδοσης

Με την τοποθέτηση μαζί των εξισώσεων κατάστασης και την εξίσωση εξόδου, θα έχουμε τη γενική μορφή του χώρου καταστάσεων :

$$\begin{cases} \dot{x}_{n\times 1} = A_{n\times n} \cdot x_{n\times 1} + B_{n\times m} \cdot u_{m\times 1} \\ y_{p\times 1} = C_{p\times n} \cdot x_{n\times 1} + D_{p\times m} \cdot u_{m\times 1} \end{cases}$$

Για να θυμόμαστε ευκολότερα τη γενική μορφή του μοντέλου state-space, μπορούμε να το γράψουμε χωρίς να καθορίσουμε το μέγεθος των διανυσμάτων και των πινάκων.

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u \\ y = C \cdot x + D \cdot u \end{cases}$$

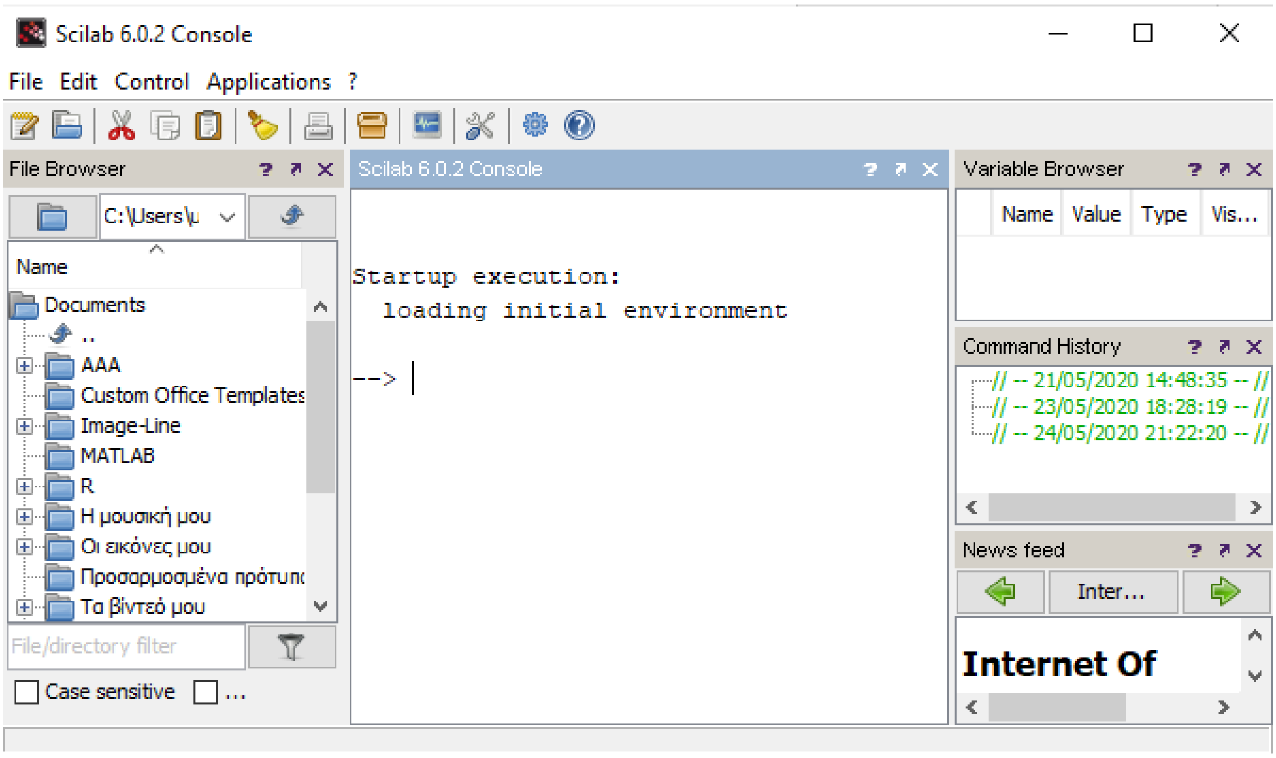
Εισαγωγή στη Scilab

Η Scilab είναι ένα αριθμητικό υπολογιστικό λογισμικό και μία αριθμητικά προσανατολισμένη γλώσσα προγραμματισμού. Η Scilab κατασκευάστηκε το 1990 από ερευνητές της INRIA και της École nationale des ponts et chaussées (ENPC). Είχε ως απώτερο σκοπό να προωθήσει την Sci lab ως παγκόσμιο λογισμικό αναφοράς στον ακαδημαϊκό τομέα και τη βιομηχανία.

Σήμερα, αξιοποιείται για την επεξεργασία σήματος, στατιστική ανάλυση, βελτίωση εικόνας, αριθμητική βελτιστοποίηση και μοντελοποίηση, μεγάλη συνεισφορά έχει και στην μοντέρνα θεωρία ελέγχου όπως θα δούμε παρακάτω κ.α. Επίσης η Sci lab και η Gnu octave είναι οι δύο κύριες γλώσσες ανοικτού κώδικα σε Matlab (Το MATLAB (matrix laboratory) είναι ένας πρόγραμμα αριθμητικής υπολογιστικής και μια προγραμματιστική γλώσσα τέταρτης γενιάς).

Ένα σημαντικό πλεονέκτημα της Sci lab είναι ότι παρέχεται δωρεάν και διατίθεται σε Windows, Linux και Mac OS X. Μπορείτε να την κατεβάσετε στην ακόλουθη ιστοσελίδα : http://www.scilab.org/

Ανοίγοντας την Scilab παρατηρούμε το εξής περιβάλλον



Οι χώροι με τις κύριες δραστηριότητες

a) Ο χώρος ή κονσόλα που ο χρήστης μπορεί να κάνει υπολογισμούς
b) Ο χώρος σύνταξη προγραμμάτων
c) Ο χώρος γραφικών
d) Ο χώρος άμεσης βοήθειας

Παρατηρούμε ότι το αρχικό περιβάλλον χωρίζεται σε υποπαράθυρα.
Τα πιο σημαντικά είναι τα εξής:

- **Command History** : εδώ μπορούμε να δούμε όλες τις εντολές που προηγήθηκαν σε προηγούμενα sessions

- **Variable Browser**: εδώ φαίνονται οι μεταβλητές που έχουμε δηλώσει

- **Scilab Console** : εδώ γίνεται η σύνταξη εντολών από τον χρήστη

- **File Browser**: εδώ μπορούμε να ανακτήσουμε αρχεία από τον υπολογιστή σας

Στην αρχή της κονσόλας παρατηρούμε το **“ --> “**,

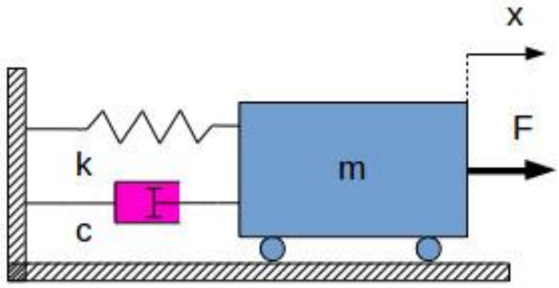
όπου δίπλα από το βέλος μπορούμε να πληκτρολογήσουμε οποιαδήποτε εντολή και στην συνέχεια, πατήσουμε το πλήκτρο Enter (Windows και Linux) ή το πλήκτρο Return (Mac OS X) για να την εκτελέσει.

Στην συνέχεια, σημαντικό είναι να γίνει αναφορά σε ένα «παρακλάδι» της Scilab, το **Xcos**. Το Xcos είναι ένα γραφικό πρόγραμμα επεξεργασίας για το σχεδιασμό μοντέλων δυναμικών συστημάτων. Τα μοντέλα έχουν την δυνατότητα μέσω του Xcos να σχεδιαστούν, να φορτωθούν, να αποθηκευτούν, να μεταγλωττιστούν και να προσομοιωθούν. Έχει υπάρξει εργονομική και αποτελεσματική λύση για βιομηχανικές και ακαδημαϊκές ανάγκες, η Xcos παρέχει λειτουργίες μοντελοποίησης μηχανικών συστημάτων (αυτοκινήτων, αεροναυπηγικής...), υδραυλικών κυκλωμάτων (φράγμα, μοντελοποίηση σωλήνων...), συστημάτων ελέγχου κ.λπ.

Ακολούθως, αφού έχει γίνει μια εισαγωγή στο περιβάλλον της Scilab έτσι ώστε να γίνει πιο οικείο για την χρήση της, μπορεί τώρα να ακολουθήσει η αλληλεπίδραση της Scilab στη μελέτη των δυναμικών συστημάτων στον χώρο των καταστάσεων με τα παρακάτω παραδείγματα.

Παράδειγμα 1

Προσδιορίζουμε το μοντέλο state-space για μια κινούμενη μάζα με ελατήριο και αποσβεστήρα, πάνω στο οποίο δρα μια δύναμη εισόδου *F.*



όπου:

m [kg] - μάζα
k [N/m] - σταθερά ελατηρίου
c [Ns/m] - συντελεστής απόσβεσης
F [N] – εξωτερική δύναμη που ενεργεί στο σώμα
x [m] - μετατόπιση του σώματος

Το σύστημα διέπεται από τις ακόλουθες συνήθεις διαφορικές εξισώσεις.

$$F - kx - c\frac{dx}{dt} - m\frac{d^2x}{dt^2} = 0$$

Το πρώτο βήμα είναι να επαναπροσδιορίσουμε την εξίσωση σε μια νέα μορφή:

$$m\ddot{x} + c\dot{x} + kx = F$$

Όπως μπορούμε να δούμε, η τάξη της διαφορικής εξίσωσης είναι δύο, επομένως πρέπει να επιλέξουμε δύο μεταβλητές κατάστασης, x_1 και x_2, καθεμία ίση με:

$$\begin{cases} x_1 = x \\ x_2 = \dot{x} \end{cases} (1)$$

Τώρα, εάν διαφοροποιήσουμε τις μεταβλητές κατάστασης, λαμβάνουμε:

$$\begin{cases} \dot{x}_1 = \dot{x} \\ \dot{x}_2 = \ddot{x} \end{cases} (2)$$

Λαμβάνοντας υπόψη τον ορισμό των μεταβλητών κατάστασης, τα παράγωγα μπορούν να γραφτούν ως:

$$\begin{cases} \dot{x}_1 = \dot{x}_2 \\ \dot{x}_2 = \ddot{x} \end{cases} (3)$$

Χρησιμοποιώντας τις (1), (2) και (3), μπορούμε να γράψουμε το σύνολο των συνηθισμένων διαφορικών εξισώσεων πρώτης τάξης ως:

$$\begin{cases} \dot{x}_1 = x_2 \\ m\dot{x}_2 + cx_2 + kx_1 = F \end{cases}$$

Η διαίρεση της δεύτερης εξίσωσης με *m* και αναδιάταξη δίνει:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{c}{m}x_2 - \frac{k}{m}x_1 + \frac{F}{m} \end{cases}$$

Εάν γράψουμε το σύνολο εξισώσεων σε μορφή πινάκων, θα λάβουμε τα εξής:

$$\begin{Bmatrix} \dot{x}_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \cdot \{F\}$$

Τώρα μπορούμε εύκολα να αναγνωρίσουμε τον πίνακα κατάστασης *A* και τον πίνακα εισαγωγής *B* .

$$A=\begin{bmatrix} 0 & 1 \\ -\dfrac{k}{m} & -\dfrac{c}{m} \end{bmatrix}, B=\begin{bmatrix} 0 \\ \dfrac{1}{m} \end{bmatrix}$$

Για να προσδιορίσουμε τους πίνακες εξόδου και άμεσης μετάδοσης , πρέπει να αποφασίσουμε ποια είναι η μετρήσιμη έξοδος. Ας θεωρήσουμε ότι θέλουμε να μετρήσουμε τη θέση του σώματος. Γι 'αυτό γράφουμε την εξίσωση εξόδου ως:

$$y = x_1$$

Εάν γράψουμε την εξίσωση εξόδου με μορφή πινάκων, λαμβάνουμε:

$$\{y\} = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \cdot \{F\}$$

Που δίνει τον πίνακα εξόδου *C* και τον πίνακα άμεσης μετάδοσης *D* :

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \end{bmatrix}$$

Για να βεβαιωθούμε ότι έχουμε προσδιορίσει σωστά τις παραμέτρους μοντέλου state-space (πίνακες), θα χρησιμοποιήσουμε το Xcos για να προσομοιώσουμε το δυναμικό μας σύστημα.

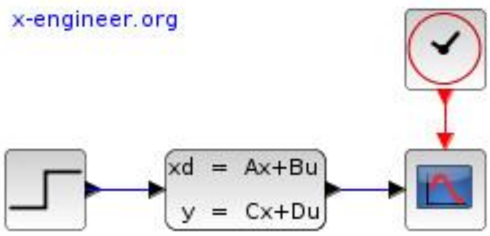
Το πρώτο βήμα είναι να φορτώσετε τις παραμέτρους σε ένα Scilab script και να το εκτελέσουμε.

```
m = 2; // [kg]

c = 1; // [Ns/m]

k = 2; // [N/m]

x0 = 0; // [m]

v0 = 0; // [m/s]


A = [0 1;-k/m -c/m];

B = [0;1/m];
```

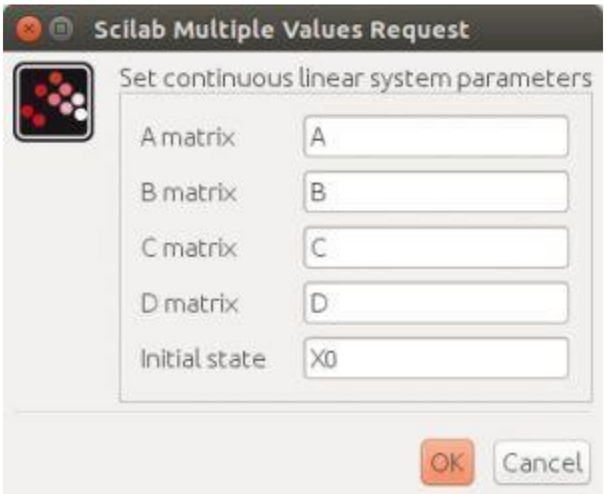
```
C = [1 0];

D = 0;

X0 = [x0 v0];
```

Όπως ίσως παρατηρήσετε, πρέπει να παρέχουμε αρχικές συνθήκες για τις μεταβλητές κατάστασης. Αυτά είναι x_0 και v_0.

Στο διάγραμμα μπλοκ Xcos πρέπει να προσθέσουμε ένα Continuous state-space system μπλοκ (CLSS), το οποίο μπορεί να βρεθεί στην παλέτα συστημάτων συνεχούς χρόνου :

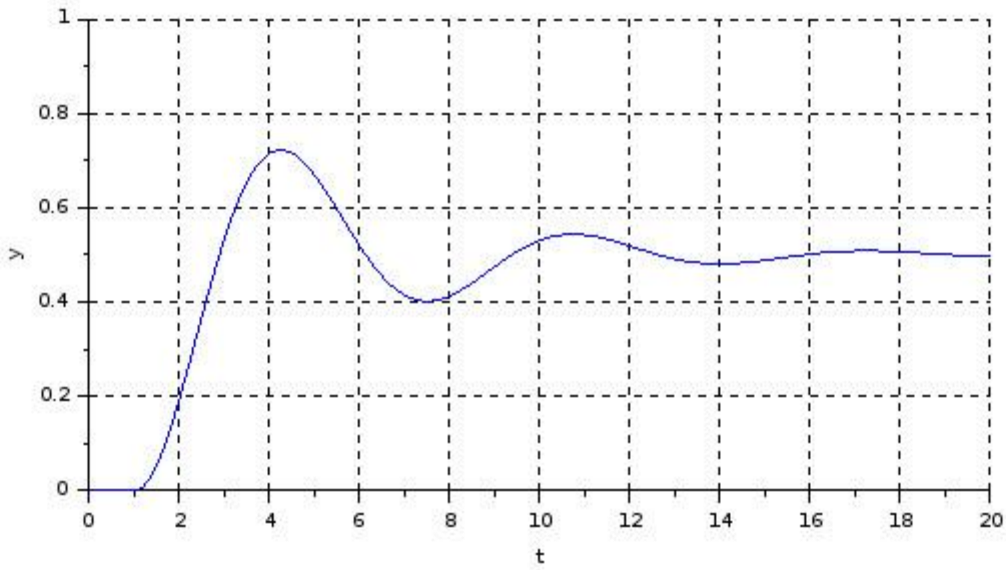


Δεδομένου ότι έχουμε ορίσει τους πίνακες state-space στον χώρο εργασίας, πρέπει να τους προσθέσουμε μόνο ως παραμέτρους Xcos state-space:



Η **είσοδος βήμα** δύναμη *F* έχει ρυθμιστεί για να δώσει **1N**, μετά από **1s** από την προσομοίωση.

Μετά την εκτέλεση της προσομοίωσης **20s** και το μπλοκ Scope ρυθμιστεί με τα σωστά όρια σήματος, η εκτέλεση του μοντέλου δίνει το ακόλουθο παράθυρο γραφικών:



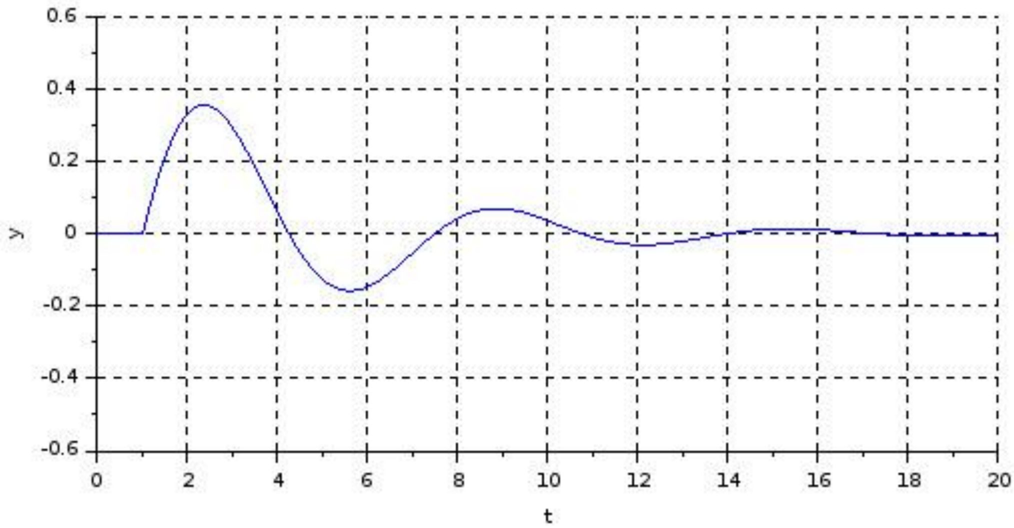
Εάν πρέπει να μετρήσουμε την ταχύτητα, πρέπει να τροποποιήσουμε την εξίσωση εξόδου ανάλογα:

$$y = x_2$$

Αυτό θα έχει ως αποτέλεσμα τον **πίνακα εξόδου** *C* να είναι:

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \end{bmatrix}$$

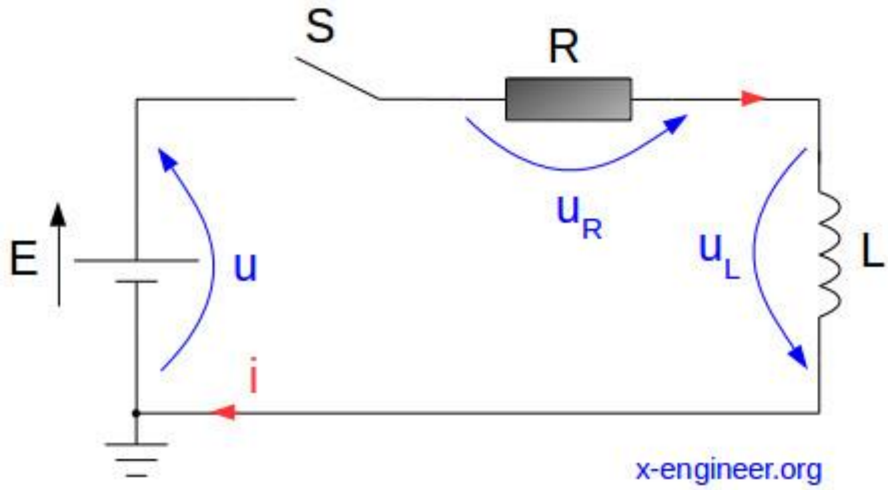
Η εκτέλεση του μοντέλου με το ενημερωμένο *C* matrix δίνει το ακόλουθο παράθυρο γραφικών:



Παράδειγμα 2

Προσδιορίστε το μοντέλο κατάστασης χώρου για ένα απλό κύκλωμα RL συνδεδεμένο σε μια εξωτερική πηγή τάσης.

Το σχήμα του κυκλώματος απεικονίζεται στην παρακάτω εικόνα.



όπου:

E [V] - πηγή συνεχούς τάσης
S – διακόπτης
R [Ω] – αντίσταση
L [H] – επαγωγή
u [V] - πτώση τάσης στο κύκλωμα
u_R [V] - πτώση τάσης στην αντίσταση
u_L [V] - πτώση τάσης στον επαγωγέα
i [A] - ηλεκτρικό ρεύμα μέσω του κυκλώματος

Η διαφορική εξίσωση που διέπει το σύστημα είναι:

$$E = Ri(t) + L\frac{di(t)}{dt} \quad (4)$$

Επειδή η διαφορική εξίσωση είναι τάξης 1, χρειαζόμαστε μόνο μία μεταβλητή κατάστασης:

$$x_1 = i \quad (5)$$

Αντικαθιστώντας το (5) στο (4), δίνεται:

$$E = Rx_1(t) + L\dot{x}_1$$

Διαιρώντας με *L* και αναδιάταξη, έχουμε:

$$\dot{x}_1 = -\frac{R}{L}x_1 + \frac{E}{L}$$

Σε μορφή πινάκων, η εξίσωση γράφεται ως:

$$\{\dot{x}_1\} = \left[-\frac{R}{L}\right]\cdot\{x_1\} + \left[\frac{1}{L}\right]\cdot\{E\}$$

από την οποία μπορούμε να εξαγάγουμε τον πίνακα κατάστασης Α και τον πίνακα εισαγωγής Β:

$$A = \left[-\frac{R}{L}\right], B = \left[\frac{1}{L}\right]$$

Η μετρήσιμη μεταβλητή μας είναι το ηλεκτρικό ρεύμα i, επομένως η εξίσωση εξόδου είναι:

$$y = x_1$$

Εάν γράψουμε την εξίσωση εξόδου σε μορφή πίνακα, λαμβάνουμε:

$$\{y\} = [1]\cdot\{x_1\} + [0]\cdot\{E\}$$

Που δίνει τον **πίνακα εξόδου** *C* και τον **πίνακα άμεσης μετάδοσης** *D* :

$$C = [1], D = [0]$$

Για να ελέγξουμε τις παραμέτρους μοντέλου state-space (πίνακες), θα χρησιμοποιήσουμε το ίδιο διάγραμμα μπλοκ Xcos για να προσομοιώσουμε το δυναμικό μας σύστημα.

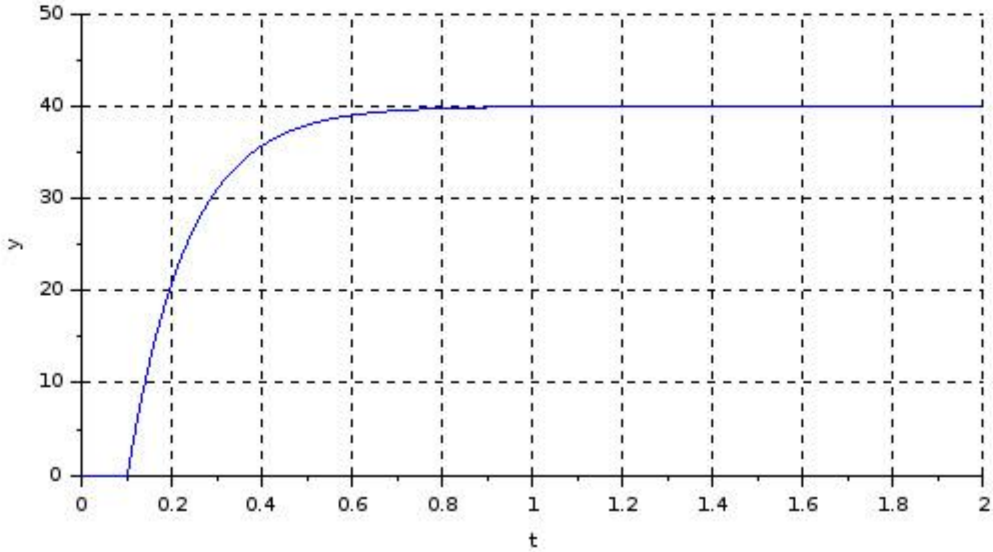
Εφόσον είχαμε μόνο μία μεταβλητή κατάστασης, οι πίνακες κρατικού-διαστήματος έχουν μετατραπεί σε βαθμίδες. Για να προσομοιώσουμε το ηλεκτρικό μας σύστημα, ως μοντέλο κρατικού-διαστήματος, χρειάζεται μόνο να βελτιώσουμε τον ορισμό των πινάκων state-space στο μοντέλο που χρησιμοποιήσαμε:

```
E = 12;

R = 0.3;

L = 0.04;
```

```
A = -R/L;

B = 1/L;

C = 1;

D = 0;

X0 = 0;
```

Η βηματική τάση εισόδου *E* έχει ρυθμιστεί να δίνει 12 V μετά από 0.1s προσομοίωση.

Μετά την εκτέλεση της προσομοίωσης **2s**, λαμβάνουμε το ακόλουθο παράθυρο γραφικών:



Εφαρμογή Scilab μοντέλων State-space

Τα μοντέλα State-space μπορούν να εφαρμοστούν και να προσομοιωθούν και στο Scilab, χρησιμοποιώντας τις προκαθορισμένες λειτουργίες **syslin()** και **csim()**. Με αυτές τις συναρτήσεις ο χρήστης μπορεί να ορίσει γραμμικά συστήματα σε συνεχή ή διακριτό χρόνο και να ελέγξει την απόκριση για συναρτήσεις εισόδου.

Η γενική σύνταξη της **syslin()** συνάρτησης είναι:

sys=syslin(dom,A,B,C [,D [,x0]])

όπου:

- **dom**- συμβολοσειρά χαρακτήρων (**'c'**, **'d'**) ή **[]**, ή κλίμακα, που καθορίζει τον τομέα χρόνου του συστήματος · μπορεί να έχει τις ακόλουθες τιμές: **'c'** για ένα σύστημα συνεχούς χρόνου, **'d'** για ένα διακριτό σύστημα χρόνου, **n** για ένα σύστημα δειγματοληψίας με περίοδο δειγματοληψίας **n** (σε δευτερόλεπτα).
- **A, B, C, D**- πίνακες της αναπαράστασης συστήματος κατάστασης-χώρου (**D** προαιρετικά, με προεπιλεγμένη τιμή **0**)
- **x0**- διάνυσμα που αντιπροσωπεύει τις αρχικές τιμές των μεταβλητών κατάστασης (εάν δεν παρέχεται η προεπιλεγμένη τιμή είναι **0**)
- **sys** - το όνομα του γραμμικού συστήματος που ορίζεται ως **tlist**

Για παράδειγμα το σύστημα μας, για να καθορίσουμε το μοντέλο state-space, πρέπει να εισαγάγουμε τις ακόλουθες οδηγίες Scilab:

```
sysMech = syslin ('c', A, B, C, D, X0 ');
```

Παρατηρήστε ότι το αρχικό διάνυσμα συνθηκών πρέπει να είναι ένα διάνυσμα στήλης, γι 'αυτό χρησιμοποιούμε τη ανάστροφο του **X0**.

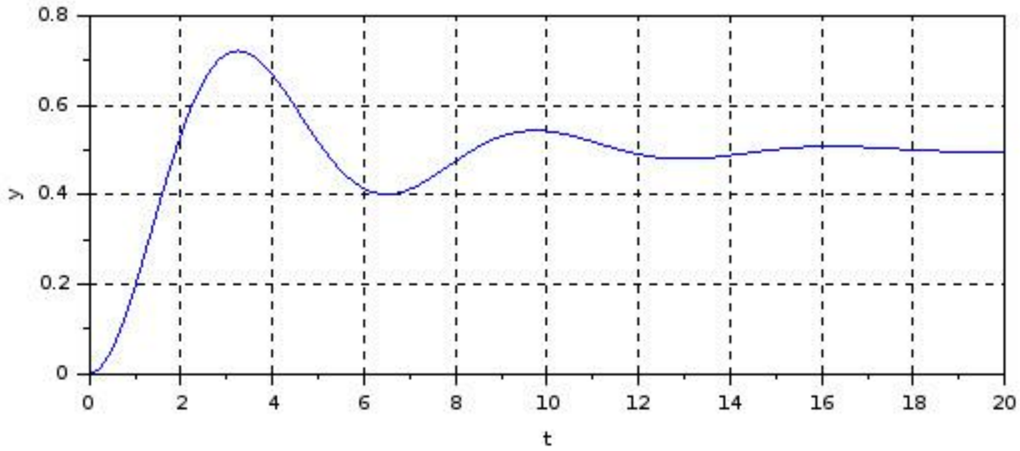
Στη συνέχεια, πρέπει να καθορίσουμε τον τομέα χρόνου για τον οποίο πρόκειται να πραγματοποιήσουμε την προσομοίωση και επίσης τον τύπο της συνάρτησης εισόδου για την οποία θέλουμε να παρατηρήσουμε την απόκριση. Για παράδειγμα, πρόκειται να προσομοιώσουμε **20s** για μια ενιαία είσοδο βημάτων (η τιμή βήματος αντιστοιχεί στο **t = 0 s**).

```
t = 0:0.01:20;

[ys,xs]=csim("step",t,sysMech);
```

Η απόκριση του συστήματος αποθηκεύεται στο **ys** διάνυσμα. Για να σχεδιάσουμε την παραλλαγή του στο χρόνο, θα χρησιμοποιήσουμε τις ακόλουθες οδηγίες Scilab:

```
plot(t,ys), xgrid()

xlabel('t')

ylabel('y')
```

Η εκτέλεση των οδηγιών θα δημιουργήσει το ακόλουθο παράθυρο γραφικών:



Όπως μπορούμε να δούμε, έχουμε την ίδια απάντηση όπως και για το μοντέλο διάγραμμα μπλοκ Xcos, η μόνη διαφορά είναι ότι ο ενιαίος χρόνος βημάτων είναι 0s για το μοντέλο Scilab και 0.1s για το μοντέλο Xcos.

Παράδειγμα 3

Ας πάρουμε το απλό μοντέλο σύνδεσης κινητήρα DC ως παράδειγμα. Μια δυναμική εξίσωση που διέπει την κίνηση μπορεί να γραφτεί ως

$$J\ddot{\theta}(t) + B\dot{\theta}(t) = u(t) \quad (6)$$

Για να απλοποιηθεί η σημειογραφία, παραλείπεται ο χρόνος που εξαρτάται. Ορίζουμε τις καταστάσεις συστήματος ως τη θέση και την ταχύτητα του συνδέσμου

$$\begin{aligned} x_1 &= \theta \\ x_2 &= \dot{\theta} \end{aligned} \quad (7)$$

Χρησιμοποιώντας αυτές τις μεταβλητές κατάστασης, (6) μπορεί να ξαναγραφεί ως σύστημα διαφορικών εξισώσεων πρώτης τάξης

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{B}{J}x_2 + \frac{1}{J}u(t) \end{aligned} \quad (8)$$

ή σε μορφή πινάκων ως

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} u \quad (9)$$

και την εξίσωση εξόδου, με y = θ

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{(10)}$$

Γενικά, ένα γραμμικό αμετάβλητο σύστημα (LTI) μπορεί να περιγραφεί ως

$$\dot{x} = Ax + Bu \text{ (11)}$$
$$y = Cx + Du \text{ (12)}$$

όπου x, u, y αντιπροσωπεύουν τα διανύσματα κατάστασης, εισόδου και εξόδου, αντίστοιχα. Η (11) ονομάζεται εξίσωση κατάστασης και (12) εξίσωση εξόδου. Σημειώστε ότι αυτή η αναπαράσταση δεν είναι μοναδική, αλλά εξαρτάται από τον τρόπο καθορισμού των καταστάσεων. Επιπλέον, αν και στο παράδειγμα του κοινού ρομπότ, οι καταστάσεις είναι βολικά τοποθετημένες σε θέση και ταχύτητα, για τη γενική περίπτωση, οι καταστάσεις ενδέχεται να μην έχουν καμία φυσική σημασία, και ως εκ τούτου δεν μπορούσαν να μετρηθούν.

Για ένα σύστημα που αντιπροσωπεύεται από τα (11) και (12), είναι εύκολο να αποδειχθεί ότι η αντίστοιχη συνάρτηση μεταφοράς ισούται με

$$P(s) = C(sI - A)^{-1}B + D \text{ (13)}$$

Για μετατροπή μεταξύ του χώρου κατάστασης και της λειτουργίας μεταφοράς στο Scilab, χρησιμοποιήστε τις εντολές ss2tf και tf2ss. Για παράδειγμα, για μια συνάρτηση μεταφοράς εγκατάστασης P (s) = 1 / (s ^ 2 + s)

```
-> s = poly (0, 's');
-> P = 1 / (s ^ 2 + s)
 P =
    1
   -----
      2
   s + s
```

Αυτό μπορεί να μετατραπεί σε μορφή state-space έως

```
-> Pss = tf2ss (P);
```

Βεβαιωθείτε ότι οι πίνακες συμμορφώνονται με (4) με

```
-> Pss.A
 ans =
   0. 1.
   0. - 1.
```

```
-> Pss.B
 ans =
   0.
   1.
-> Pss.C
 ans =
   1. 0.
```

Μετατροπή σε συνάρτηση μεταφοράς από ss2tf

```
-> P1 = ss2tf (Pss)
 P1 =
    1
   -----
      2
   s + s
```

Σημειώνουμε ότι η συνάρτηση μεταφοράς για ένα φυσικό σύστημα πρέπει να είναι απολύτως σωστή, δηλαδή η απόκριση συχνότητας πρέπει να μηδενίζεται καθώς η συχνότητα πλησιάζει το άπειρο. (Κανένα σύστημα δεν θα μπορούσε να έχει απεριόριστο εύρος ζώνης στην πραγματικότητα.) Αυτό σημαίνει ότι η αναπαράσταση κατάστασης-χώρου πρέπει να έχει μηδενικό πίνακα D.

Έλεγχος ανατροφοδότησης κατάστασης

Προφανώς, ένας έλεγχος ανατροφοδότησης κατάστασης είναι εφικτός στην πράξη όταν όλες οι καταστάσεις είναι μετρήσιμες, όπως η δυναμική άρθρωση του ρομπότ στο (9) με τη θέση της άρθρωσης και την ταχύτητα ως μεταβλητές κατάστασης. Ο σχεδιασμός ανατροφοδότησης κατάστασης μπορεί να εκτελεστεί με ένα σχήμα γνωστό ως τοποθέτηση πόλων ή με πιο συστηματικό τρόπο χρησιμοποιώντας τη φόρμουλα του Ackerman .

Η ιδέα της τοποθέτησης πόλων είναι απλή. Καθορίζουμε τις θέσεις πόλων του συστήματος κλειστού βρόχου που αποδίδει τα επιθυμητά κριτήρια σταθερότητας και απόδοσης. Στη συνέχεια, επιλέγουμε το κέρδος του ελέγχου κατάστασης ανάδρασης για να μετακινήσουμε τους πόλους κλειστού βρόχου. Απαραίτητη προϋπόθεση είναι ότι το σύστημα πρέπει να είναι σταθεροποιήσιμο.

Ο ελεγκτής ανάδρασης κατάστασης είναι απλώς ένα διάνυσμα κερδών $K = [k_1, k_2 \ldots k_n]^T$ που συνδέει τις καταστάσεις με την είσοδο του συστήματος. Έτσι, για ένα σύνολο καθορισμένων πόλων κλειστού βρόχου, ο σχεδιαστικός στόχος είναι ο υπολογισμός k_i , i = 1,…, n. Για να το δούμε πιο καθαρά, υποθέστε ότι η είσοδος είναι μηδέν. Έχουμε στην είσοδο του συστήματος

$$u = -Kx \quad (14)$$

και την εξίσωση κατάστασης κλειστού βρόχου

$$\dot{x} = (A - BK)x \quad (15)$$

Οι πόλοι κλειστού βρόχου μπορούν να υπολογιστούν από

$$\det(sI - A + BK) = 0 \quad (16)$$

Εν τω μεταξύ, ο καθορισμός των πόλων κλειστού βρόχου p_i , i = 1,…, n αποδίδει το χαρακτηριστικό πολυώνυμο

$$a(\mathrm{s}) = (\mathrm{s} - \mathrm{p}_1)(s - p_2)\ldots(s - p_n) \quad (17)$$

Ως εκ τούτου, μπορούμε να συγκρίνουμε (16) και (17) να λύσουμε χειροκίνητα ως προς το K, κάτι που θα μπορούσε να είναι κουραστικό για εξισώσεις υψηλότερης τάξης. Το Scilab παρέχει μια βολική εντολή ppol για εύρεση του K, δεδομένου των πινάκων A, B και ενός διανύσματος των επιθυμητών πόλων ως επιχειρήματα.

Παράδειγμα 4

Ας σχεδιάσουμε έναν έλεγχο ανατροφοδότησης κατάστασης για τον απλό σύνδεσμο ρομπότ που περιγράφεται στο (9), (10) με J = 1, B = 1. Καθορίζουμε τις επιθυμητές ιδιότητες του συστήματος κλειστού βρόχου ως εξής:

1. υπέρβαση λιγότερο από 5%
2. χρόνος αύξησης κάτω από 0,1 δευτερόλεπτο

Με τυπική ανάλυση του συστήματος 2ης τάξης, έχουμε ότι η προδιαγραφή (6) μεταφράζεται σε αναλογία απόσβεσης ζ ≥ 0,7, και χρησιμοποιώντας τη σχέση $t_r = 1{,}8 / \omega_n$, έχουμε για την (7) ότι $\omega_n \geq$ 18rad/s. Αντικαθιστώντας αυτές τις δύο τιμές στα χαρακτηριστικά πολυωνύμων κλειστού βρόχου

$$\Lambda(s) = \mathrm{s}^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 25.2s + 324 \quad (18)$$

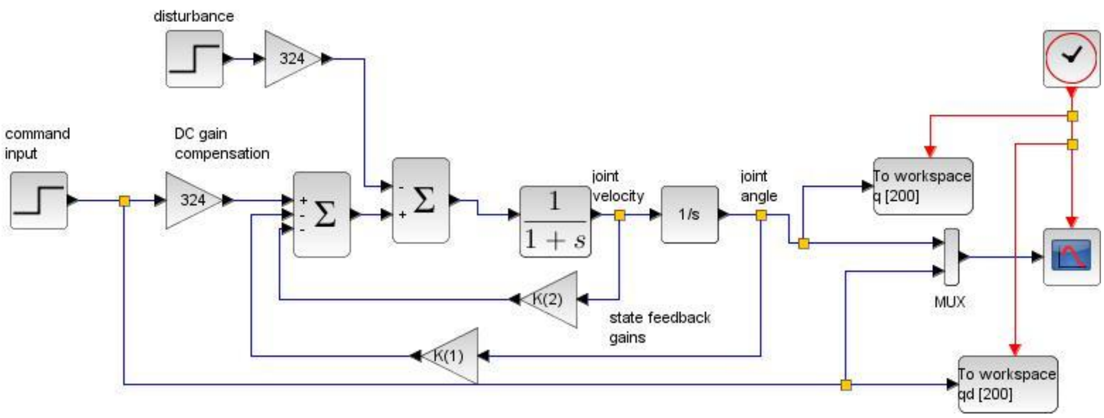
με πόλους σε -12,6 ± 12,8546i. Η παραπάνω διαδικασία μπορεί να πραγματοποιηθεί με τις ακόλουθες εντολές Scilab

```
-> z = 0,7; wn = 18;
-> lamda = s ^ 2 + 2 * z * wn * s + wn ^ 2;
-> clpoles = ρίζες (lamda)
 clpoles =
  - 12,6 + 12,854571i
  - 12,6 - 12,854571i
```

και με τα δεδομένα του συστήματος από το Pss που υπολογίστηκαν νωρίτερα, τα κέρδη από την κατάσταση ανατροφοδότησης μπορούν να υπολογιστούν με την εντολή

```
-> K = ppol (Pss.A, Pss.B, clpoles)
 K =
   324. 24.2
```

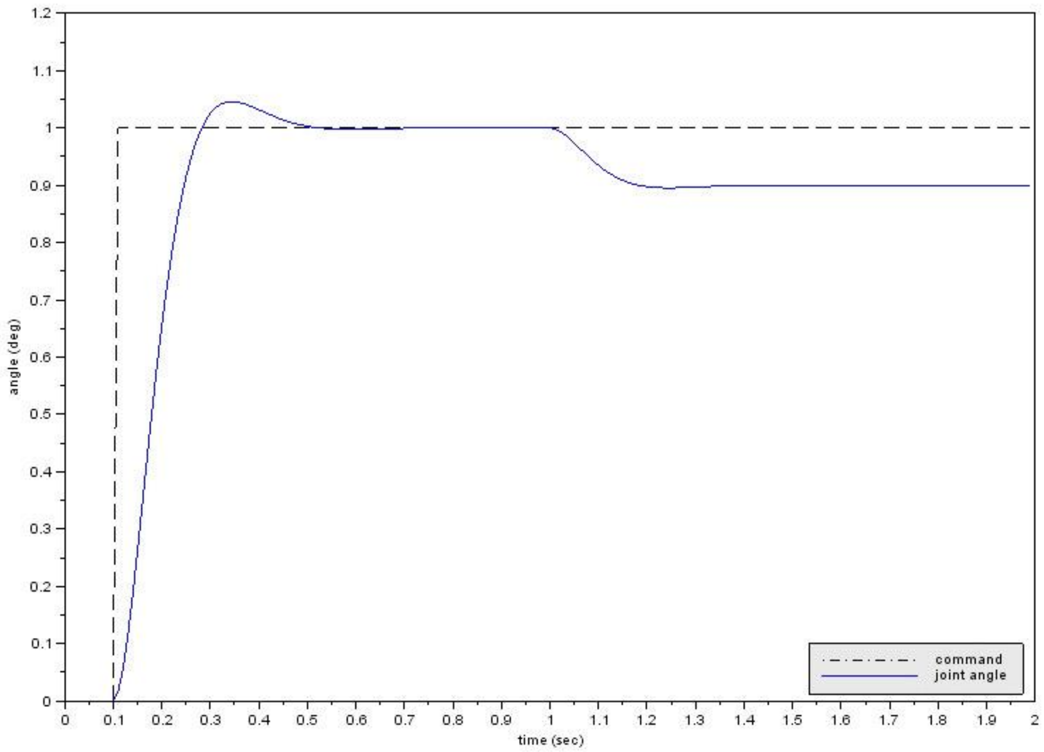
Κατασκευάσαμε το μοντέλο Xcos στο Σχήμα 1 για να προσομοιώσουμε το σύστημα. Παρατηρούμε στο διάγραμμα ότι το σύστημα αντιπροσωπεύεται από μια συνάρτηση μεταφοράς, καθώς η ταχύτητα και η γωνία της άρθρωσης είναι προσβάσιμες. Επίσης, κατά τον υπολογισμό των κερδών ανάδρασης κατάστασης, δεν λαμβάνουμε υπόψη την είσοδο εντολών. Ως εκ τούτου, η απόκριση βήματος θα έχει μη σταθερό σφάλμα σταθερής κατάστασης που πρέπει να αντισταθμιστεί με ένα κέρδος τροφοδοσίας. Ένας εύκολος τρόπος για να υπολογίσετε αυτό το κέρδος είναι με τον έλεγχο του DC κέρδους του συστήματος ανατροφοδότησης



Σχήμα 1 διάγραμμα ppol.zcos Xcos για έλεγχο ανατροφοδότησης κατάστασης

```
-> cltf = ss2tf (syslin ('c', Pss.A-Pss.B * K, Pss.B, Pss.C))
 cltf =
        1
   ---------------
             2
   324 + 25,2s + s
```

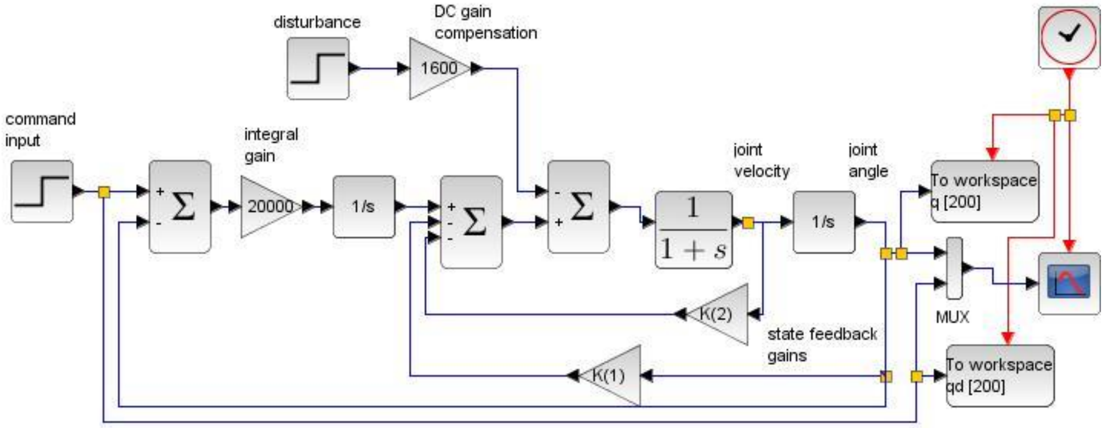
που μπορεί να βρεθεί αφήνοντας το s = jω, και στο ω = 0 το αποτέλεσμα είναι 1/324. Έτσι, για την αντιστάθμιση αυτής της μετατόπισης DC, εφαρμόζουμε ένα τροφοδοτικό κέρδος 324 και στις εισόδους βημάτων και διαταραχών. Με τη διαταραχή μεγέθους 0,1 να εισέλθει στο σύστημα σε χρόνο t = 1 δευτερόλεπτο, η προσομοίωση αποδίδει την απόκριση βημάτων στο Σχήμα 2. Βλέπουμε ότι η μεταβατική περίοδος συμμορφώνεται με την επιθυμητή προδιαγραφή. δηλαδή, (1) υπέρβαση λιγότερο από 5% (2) χρόνος αύξησης λιγότερο από 0,1 δευτερόλεπτο. Ωστόσο, το σύστημα κλειστού βρόχου δεν μπορεί να απαλλαγεί από τη σταθερή διαταραχή του 0,1 μετά από t = 1 δευτερόλεπτο. Αυτό το αποτέλεσμα είναι προβλέψιμο, επειδή η ανατροφοδότηση κατάστασης είναι απλώς ένα ζευγάρι στατικών κερδών χωρίς δυναμική για την αντιστάθμιση της διαταραχής που εισέρχεται στην είσοδο της εγκατάστασης. Στο επόμενο παράδειγμα, προτείνουμε έναν τρόπο παράκαμψης αυτού του μειονεκτήματος.



Σχήμα 2 βήμα απόκρισης από το μοντέλο Xcos στο Σχήμα 1

Παράδειγμα 5

Από το παράδειγμα PID που συζητήθηκε προηγουμένως, δείχνουμε το πλεονέκτημα του ακέραιου όρου στην εξάλειψη του σφάλματος σταθερής κατάστασης. Αυτή η αρχή μπορεί να εφαρμοστεί στο σχήμα ανατροφοδότησης κατάστασης με την αύξηση ενός ολοκληρωτή όπως φαίνεται στο Σχήμα 3. Υπάρχει κάποια συστηματική διαδικασία για την αύξηση ενός ολοκληρωτή και τον σχεδιασμό όλων των κερδών ταυτόχρονα, αλλά αυτό σημαίνει ότι η σχέση δεύτερης τάξης στο προηγούμενο παράδειγμα δεν είναι πλέον εφαρμόσιμη. Έτσι, σε αυτό το παράδειγμα, εξακολουθούμε να χρησιμοποιούμε το pole-placement για σύστημα δεύτερης τάξης όπως και πριν, και στη συνέχεια να προσαρμόσουμε το ολοκληρωμένο κέρδος μετά για να επιτύχουμε την επιθυμητή απόκριση.



Σχήμα 3 ppol_int.zcos Xcos μοντέλο ανατροφοδότησης κατάστασης με ενσωματωτή

Παρατηρήστε ότι ο ολοκληρωτής αντικαθιστά την αντιστάθμιση κέρδους DC στο Σχήμα 1 για να διορθώσει την απόκριση στην τιμή σταθερής κατάστασης στόχου. Η χρήση της ίδιας κατάστασης αποκτά αποτελέσματα οδηγεί σε βραδύτερη παροδική απόκριση από το Σχήμα 2, οπότε επανασχεδιάζουμε το βήμα τοποθέτησης πόλων αυξάνοντας & ωμέγα σε 40 rad / s.

```
-> z = 0,7; wn = 40;
-> lamda = s ^ 2 + 2 * z * wn * s + wn ^ 2;
-> clpoles = ρίζες (lamda)
 clpoles =
  - 28. + 28.565714i
  - 28. - 28.565714i
```

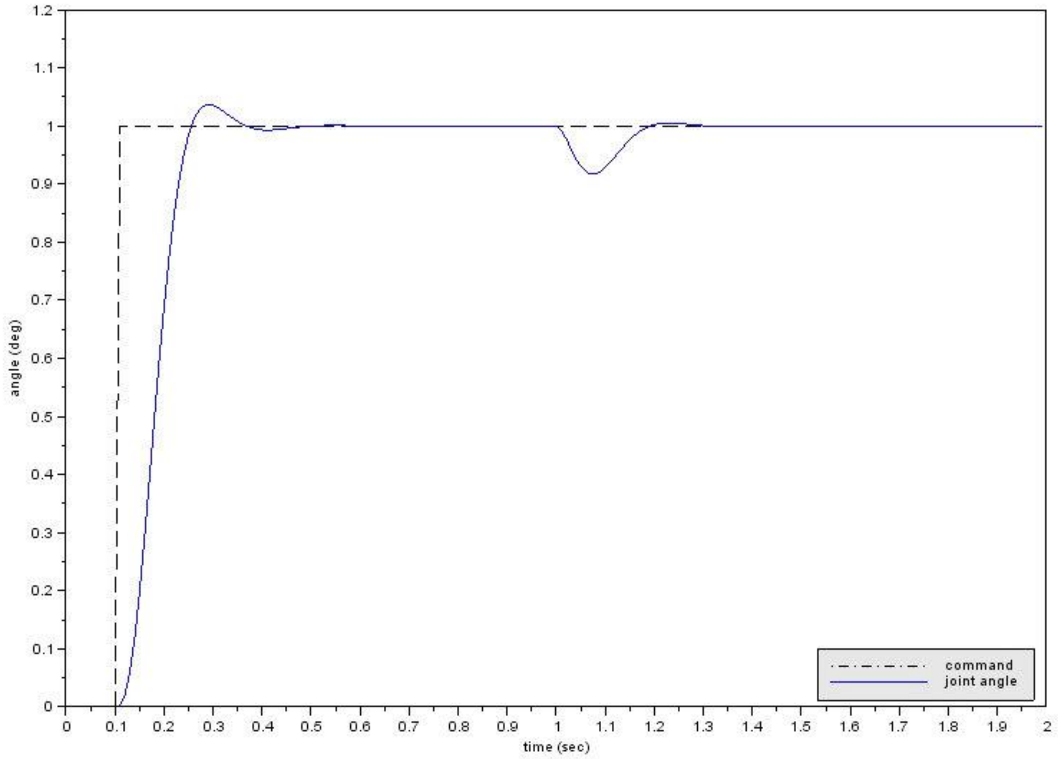
Αυτό αποφέρει ένα νέο ζεύγος κερδών ανατροφοδότησης από την κατάσταση.

```
-> K = ppol (Pss.A, Pss.B, clpoles)
 K =
   1600. 55.
```

Το κέρδος DC του συστήματος κλειστού βρόχου

```
-> cltf = καθαρό (ss2tf (syslin ('c', Pss.A-Pss.B * K, Pss.B, Pss.C)))
 cltf =
        1
   ------------ -
              2
   1600 + 56s + s
```

ισούται με 1/1600. Έτσι, ένα κέρδος 1600 εφαρμόζεται ως αποζημίωση στην είσοδο διαταραχής για να αποδώσει το ίδιο επίπεδο με το προηγούμενο παράδειγμα. δηλαδή, d = 0,1. Με τον πειραματισμό με το ακέραιο κέρδος, επιλέγουμε K i = 20000, το οποίο δίνει την απόκριση όπως στο Σχήμα 4. Ο χρόνος ανύψωσης και η υπέρβαση ικανοποιούν τις προδιαγραφές που δίνονται στο Παράδειγμα 4, ενώ το σύστημα επανέρχεται στην επιθυμητή τιμή αφού εφαρμοστεί η διαταραχή σε t = 1 sec.



Σχήμα 4 Βήμα απόκρισης από το μοντέλο Xcos στο Σχήμα 3

Βιβλιογραφία

- https://www.scilab.org/sites/default/files/Scilab_beginners_0.pdf
- https://en.wikipedia.org/wiki/Scilab
- https://www.scilab.org/software/xcos
- V.Toochinda. Robot Analysis and Control with Scilab and RTSX. Mushin Dynamics, 2014.

- https://help.scilab.org/docs/6.1.0/en_US/section_64a8529216e858b335b0e6c058385350.html