<u>ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ</u>
<u>ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ</u>

Κρανιάς Δημήτριος, Α.Μ.:03116030
Γιαννιός Γεώργιος-Ταξιάρχης, Α.Μ.:03116156
Χατζηαντωνίου Παυλίνα, Α.Μ.:03116186
Ντόκου Μυρσίνη, Α.Μ.:03116179
Σαλπέα Ναταλία, Α.Μ.:03116083

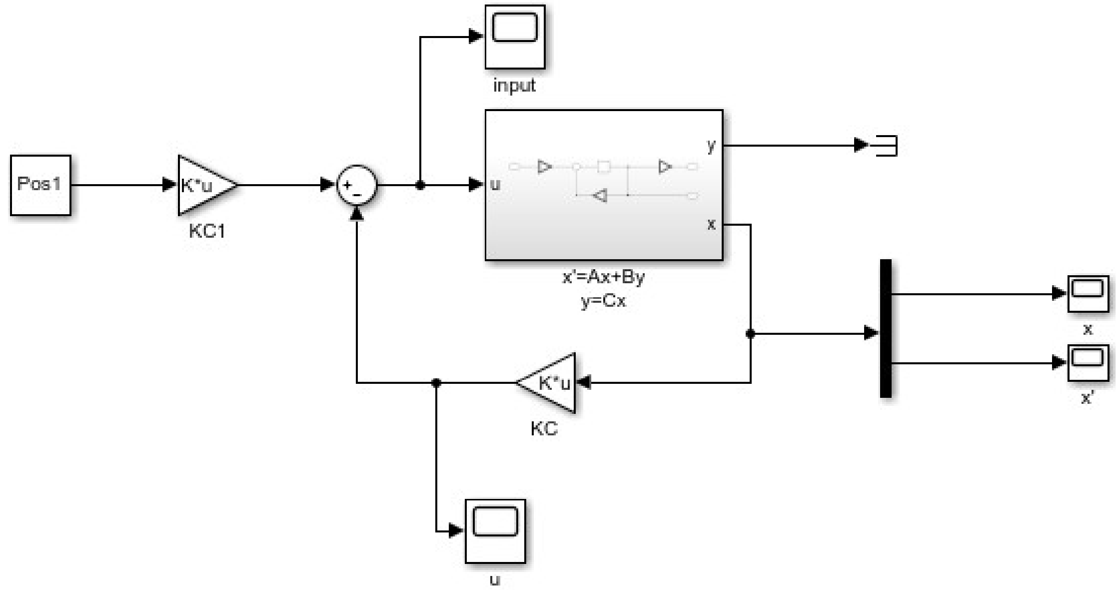
6ο Εξάμηνο

Σχεδίαση Συστημάτων Αυτομάτου Ελέγχου
Αναφορά 2ου Εργαστηρίου

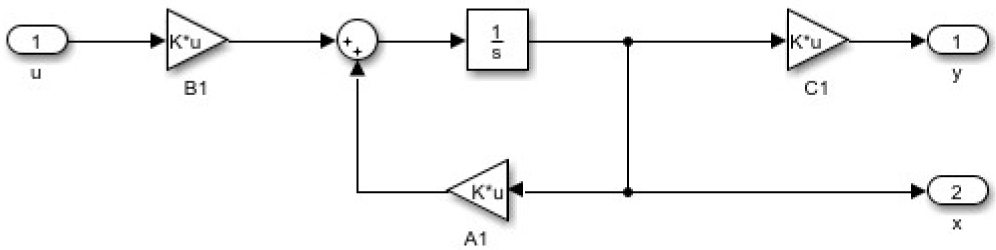
Στο δεύτερο εργαστήριο εξετάσαμε τον έλεγχο ενός βαγονιού χωρίς και με ανεστραμμένο εκκρεμές στην κορυφή του. Σκοπός μας ήταν να σχεδιάσουμε κατάλληλο feedback ώστε να επιστρέφει αυτό στην θέση ισορροπίας του(την οποία εμείς ορίσαμε με κατάλληλη σχεδίαση) γρήγορα και με μικρές ταλαντώσεις.

1ο Μέρος

Στο 1ο μέρος θα σας παρουσιάσουμε δοκιμές ελέγχου του βαγονιού χωρίς το ανεστραμμένο εκκρεμές.
Η σχεδίαση μας στο simulink:



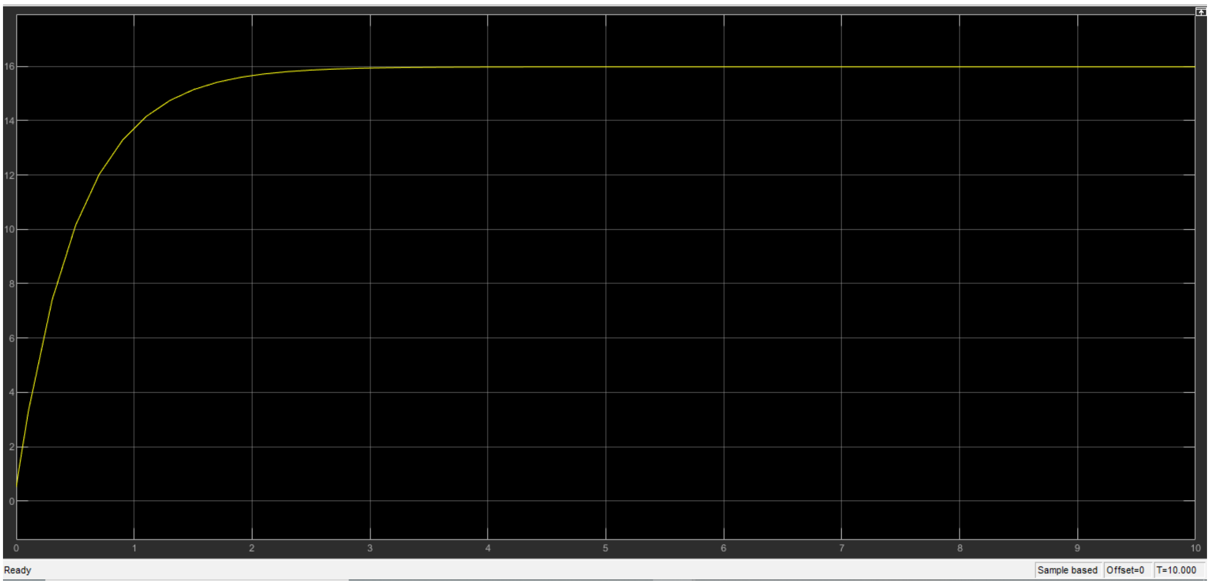
Το υποσύστημα:



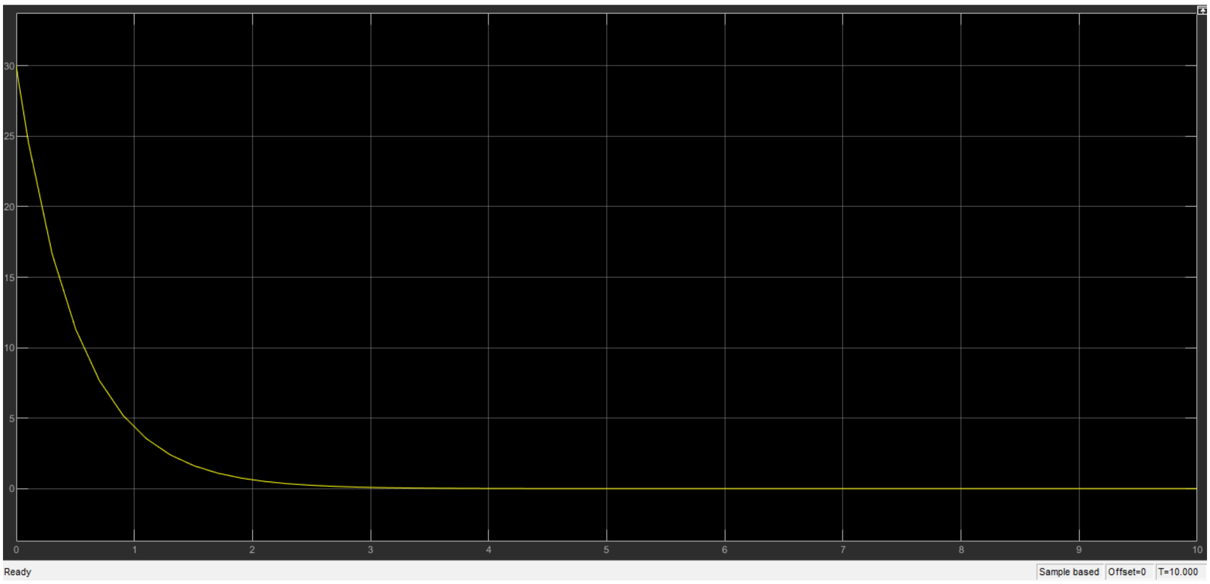
Για τις παρακάτω μετρήσεις έχουμε ορίσει αρχικές συνθήκες [0.5;30].

Αρχικά με μηδενικό feedback gain **KC=[0 0]**(Όπου το C στο KC σημαίνει Cart):

x



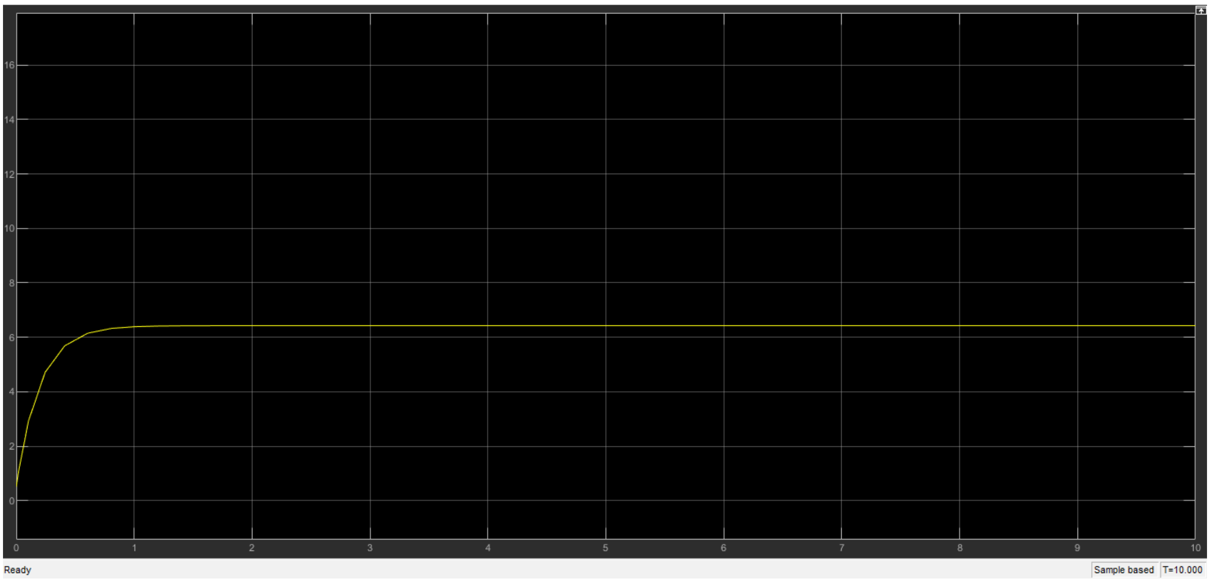
x’



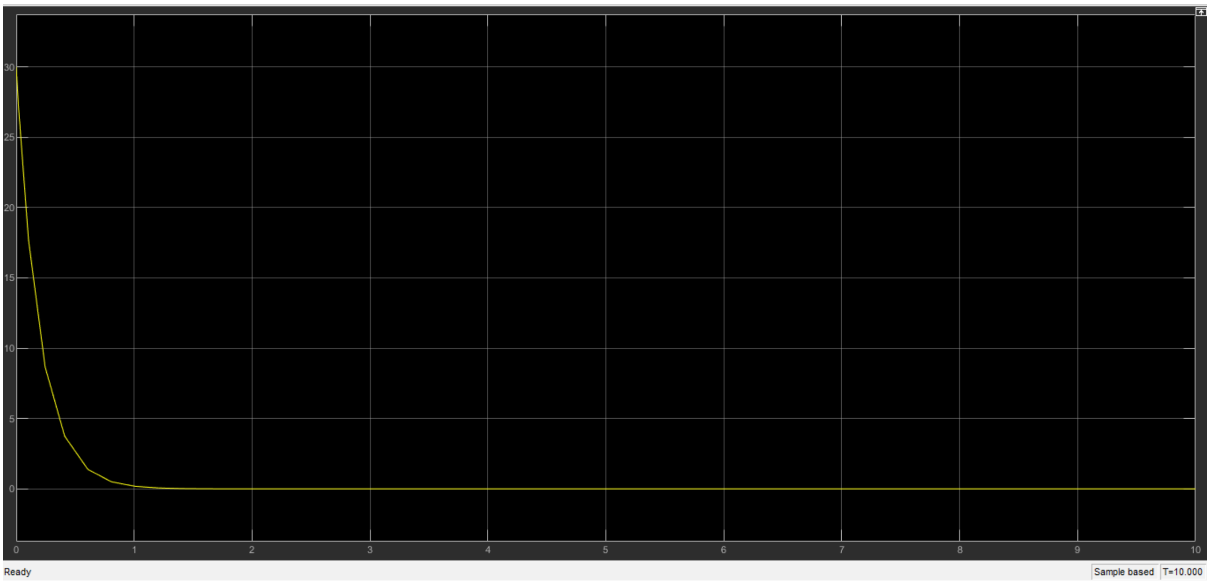
Παρατηρούμε ότι η ταχύτητα μηδενίζεται(όπως είναι λογικό αφού κάποια στιγμή το βαγονάκι σταματάει), αλλά το βαγόνι αντί να πάει προς τη θέση 0(όπως έχουμε ορίσει με την μεταβλητή Pos1), πάει στη θέση 16. Άρα, χωρίς feedback το σύστημα μας έχει πρόβλημα.

Το ίδιο πρόβλημα διαπιστώνουμε και στην περίπτωση όπου **KC=[0 KC2]** όπου KC2 τιμή μεγαλύτερη του μηδενός. Αυτή τη φορά η ταχύτητα τείνει στο 0 πιο γρήγορα αλλά το βαγονάκι πάλι καταλήγει σε λάθος θέση(έστω και αν είναι πιο κοντά στο 0). Τα παρακάτω γραφήματα δίνονται για KC2=10(Άμα επιλέγαμε μεγαλύτερη τιμή απλά θα μηδενιζόταν πιο γρήγορα η ταχύτητα και η τελική θέση θα ήταν πιο κοντά στο 0).

x

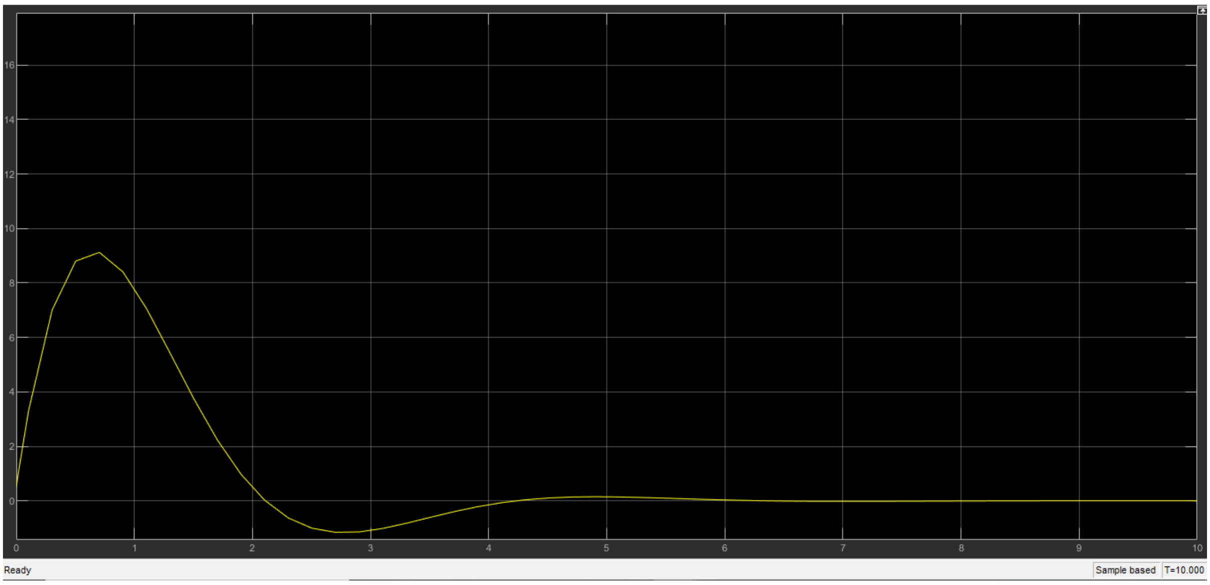


x’

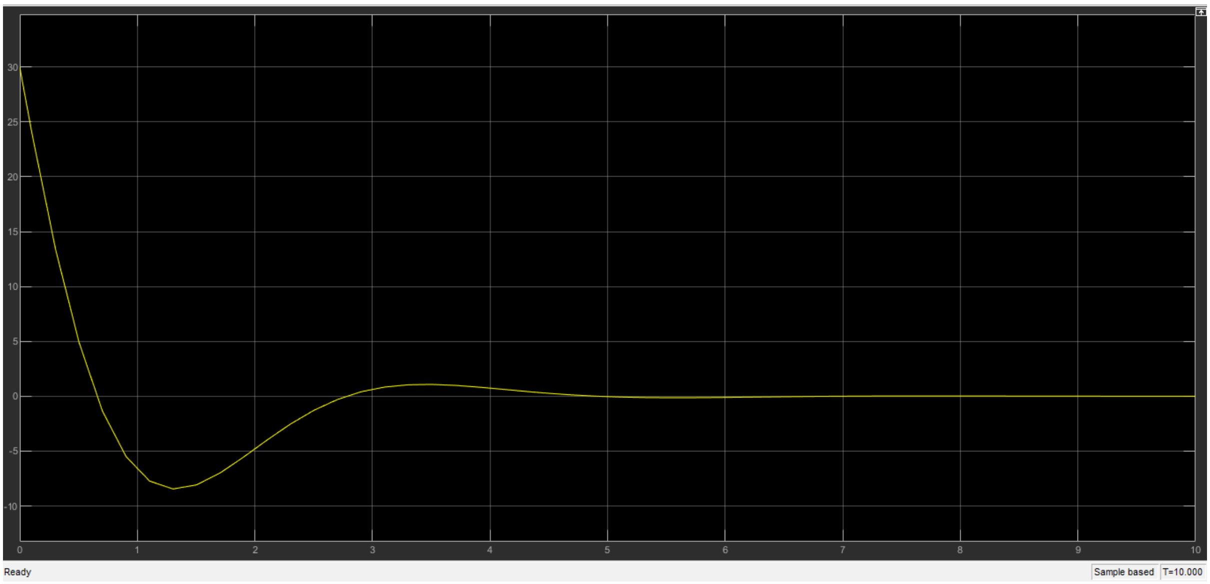


Αν τώρα επιχειρήσουμε να βάλουμε KC=[KC1 0] με KC1>0 διαπιστώνουμε ότι το βαγόνι πάει στη θέση που ορίσαμε(0). Όσο αυξάνουμε το KC1 τόσο πιο μικρές ταλαντώσεις θα κάνει(δηλαδή δεν θα απομακρυνθεί πολύ από την θέση ισορροπίας του) αλλά αυξάνεται ο αριθμός των ταλαντώσεων. Τα παρακάτω γραφήματα είναι για KC1=10.

x



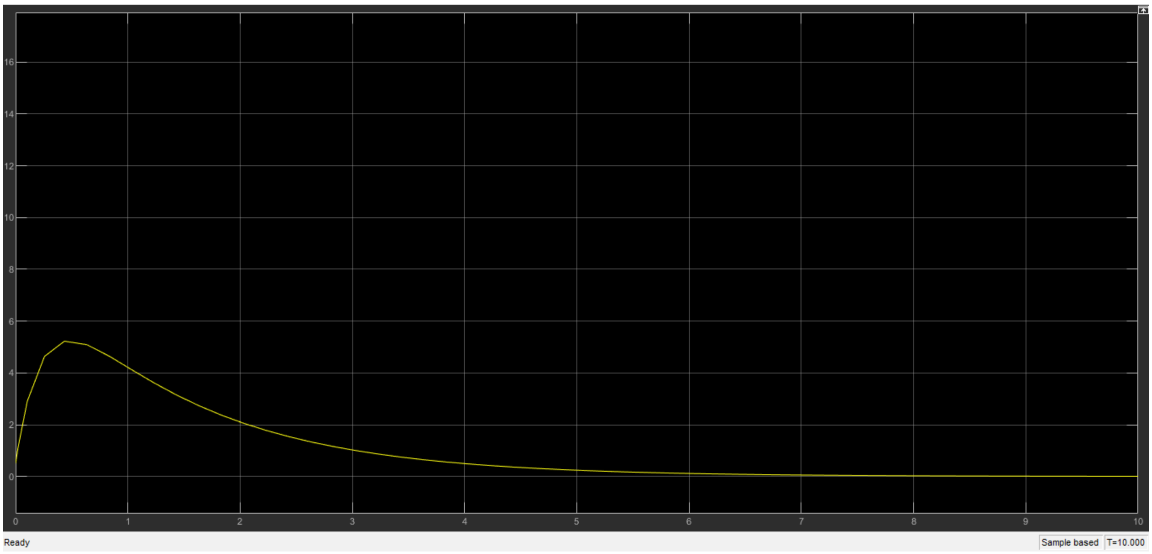
x’



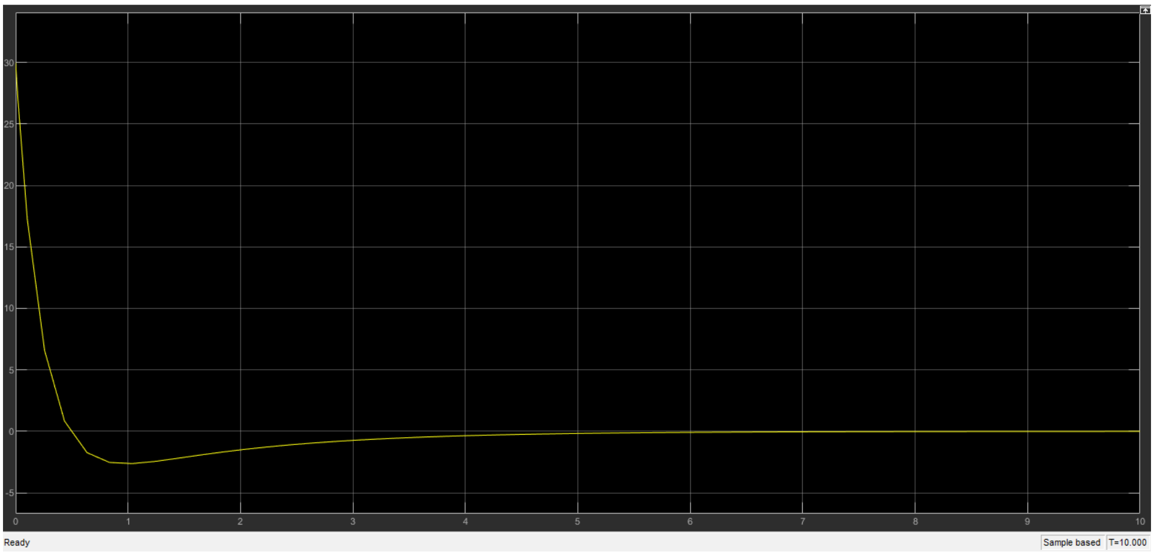
Τέλος θα μελετήσουμε την περίπτωση KC=[KC1 KC2]. Εδώ παρατηρούμε μεγάλη βελτίωση σε σχέση με την προηγούμενη περίπτωση, κυρίως στον αριθμό των ταλαντώσεων που είναι πολύ μικρότερος, αλλά και στο πλάτος τους που πάλι είναι μικρότερο. Όσο αυξάνουμε τα KC1, KC2 παρατηρούμε βελτιώσεις στον χρόνο αποκατάστασης και στο πλάτος των ταλαντώσεων. Βέβαια αυτό έρχεται με το κόστος μεγαλύτερου input value. Από εκεί και πέρα έγκειται στην προσωπική κρίση του καθενός να επιλέξει τι θέλει περισσότερο. Παρακάτω εμφανίζουμε τα γραφήματα x, x’, input για KC=[10 10], [100 100]:

KC=[10 10]

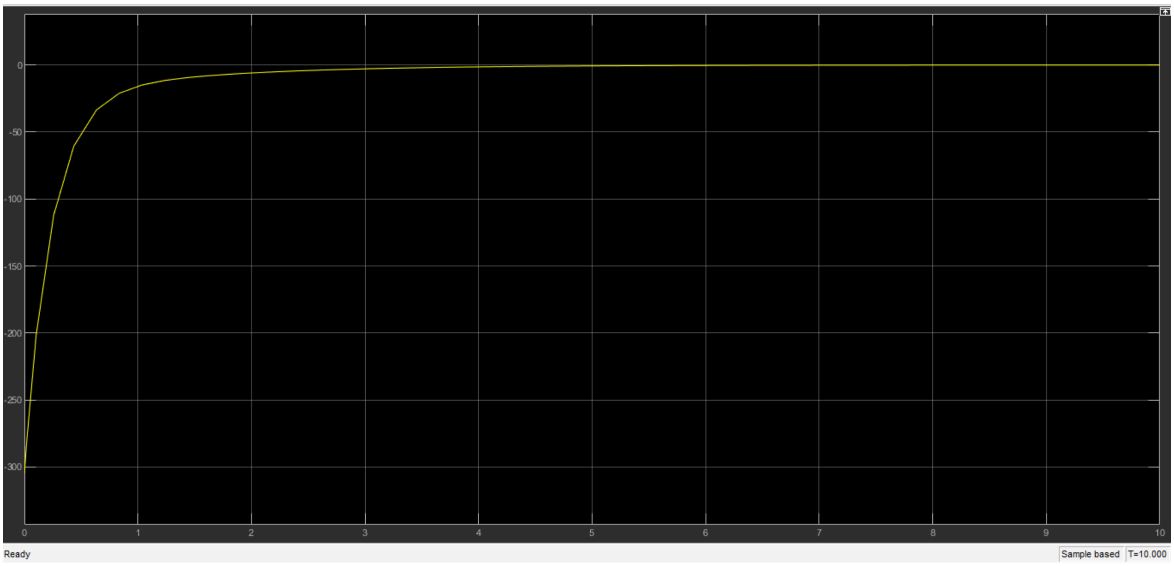
x



x'

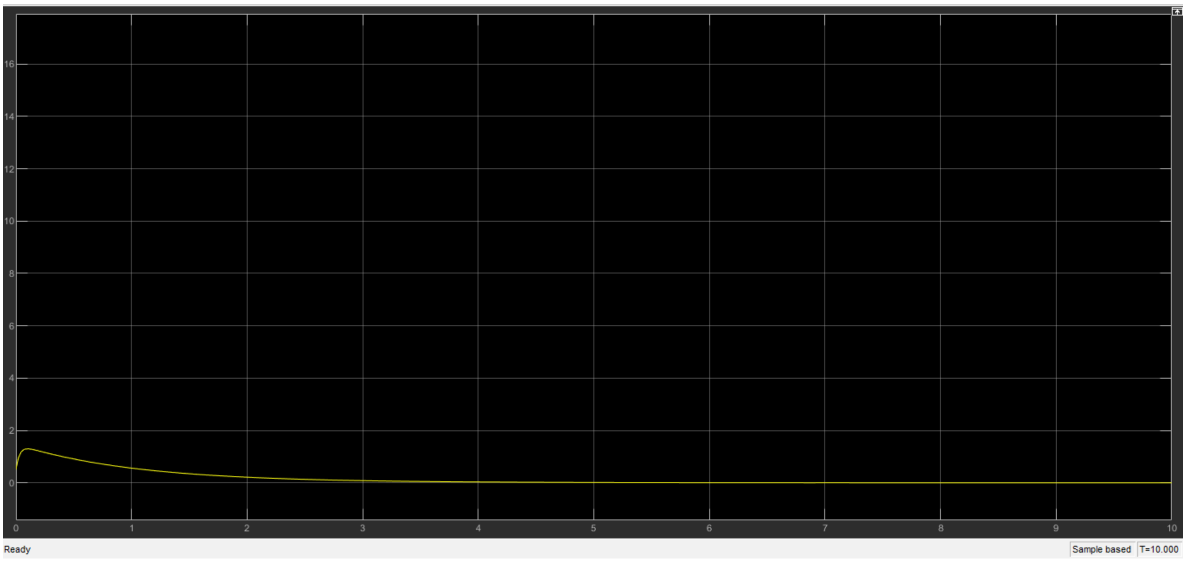


input

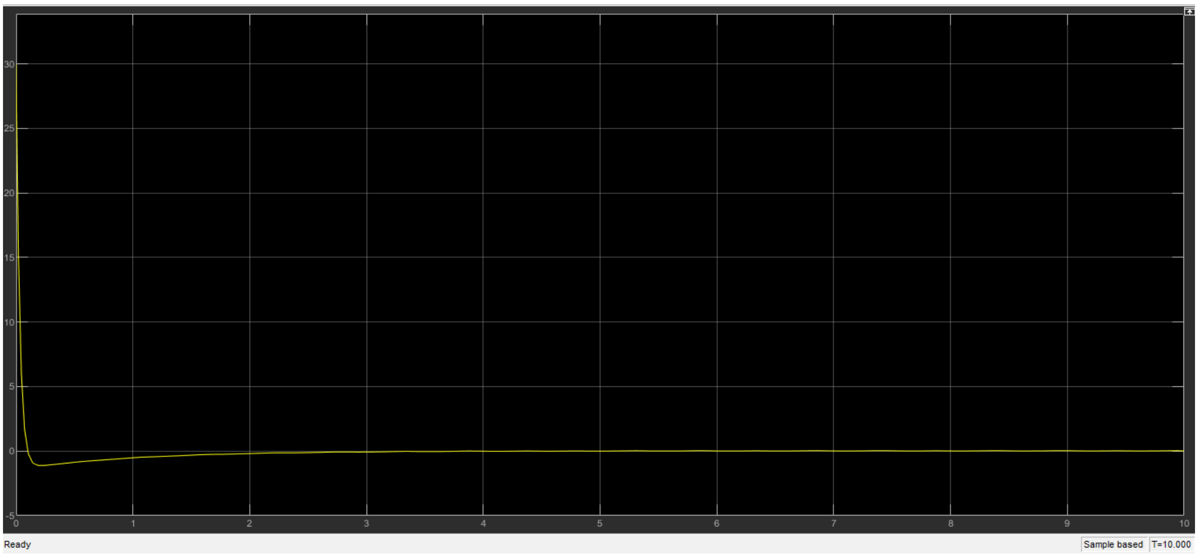


KC=[100 100]

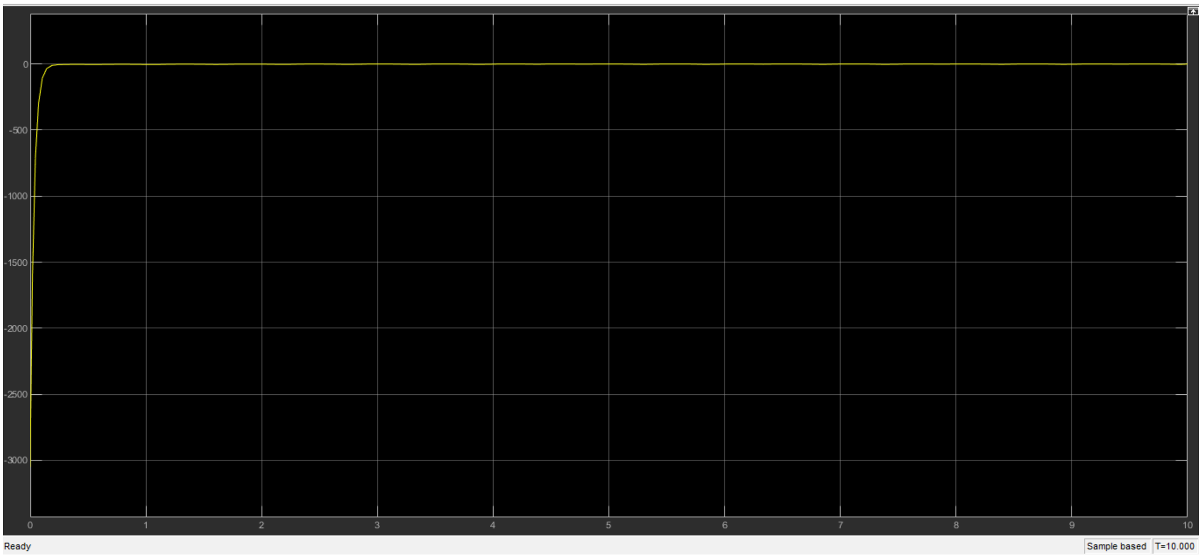
x



x’

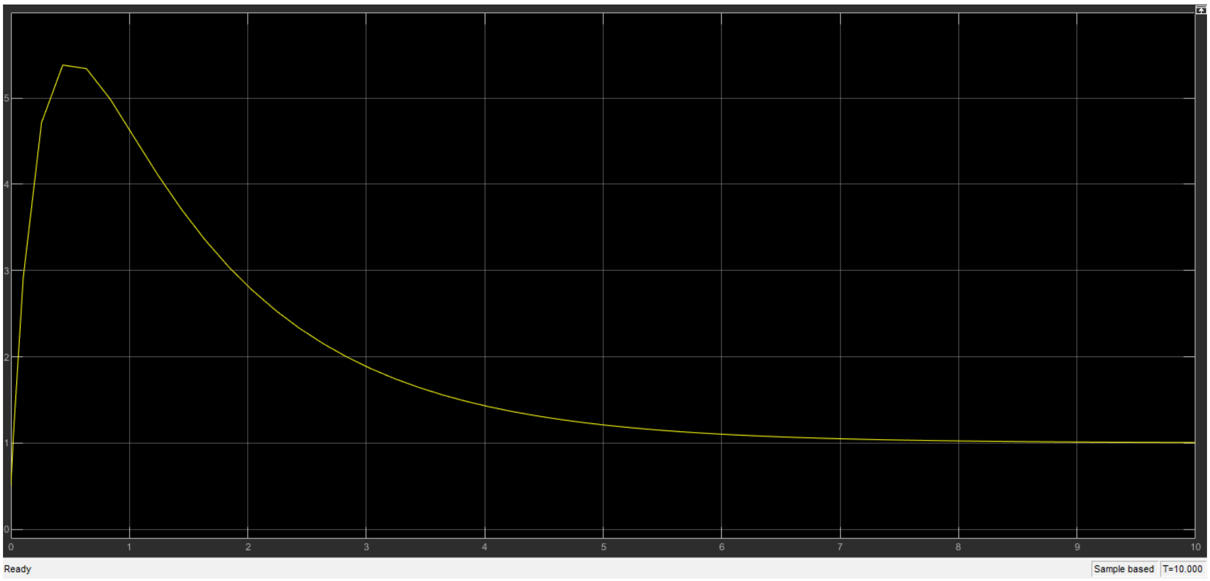


input



Δοκιμάζουμε λοιπόν για KC=[10 10] να μεταφέρουμε το βαγονάκι σε θέση ισορροπίας x=1.

2ο Μέρος

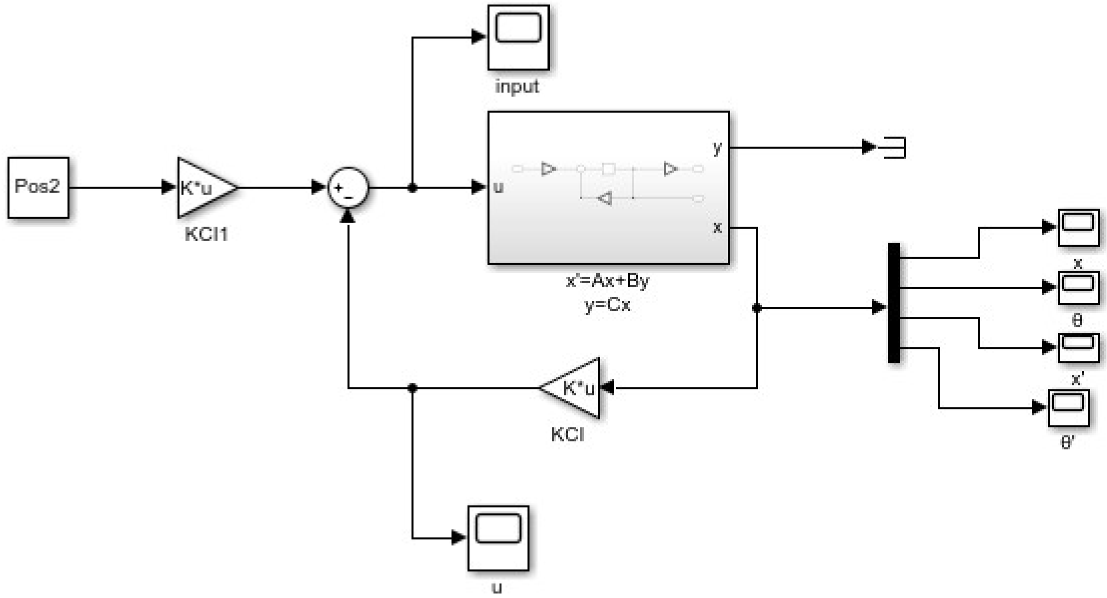


Οπότε χάρη στην ανάδραση βλέπουμε ότι μπορούμε να μεταφέρουμε το βαγόνι όπου εμείς θελήσουμε(αν αυτό είναι φυσικά δυνατόν με την αρχική ταχύτητα που επιλέξαμε).

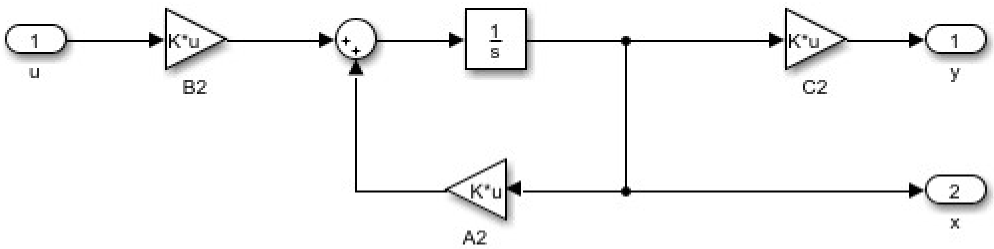
2ο Μέρος

Στο 1ο μέρος θα σας παρουσιάσουμε δοκιμές ελέγχου του βαγονιού με το

ανεστραμμένο εκκρεμές.
Η σχεδίαση μας στο simulink(σχεδόν ίδια με του 1ου μέρους):



Το υποσύστημα:



Για τις παρακάτω μετρήσεις έχουμε ορίσει αρχικές συνθήκες [0.3;-5*pi/180;20;15*pi].

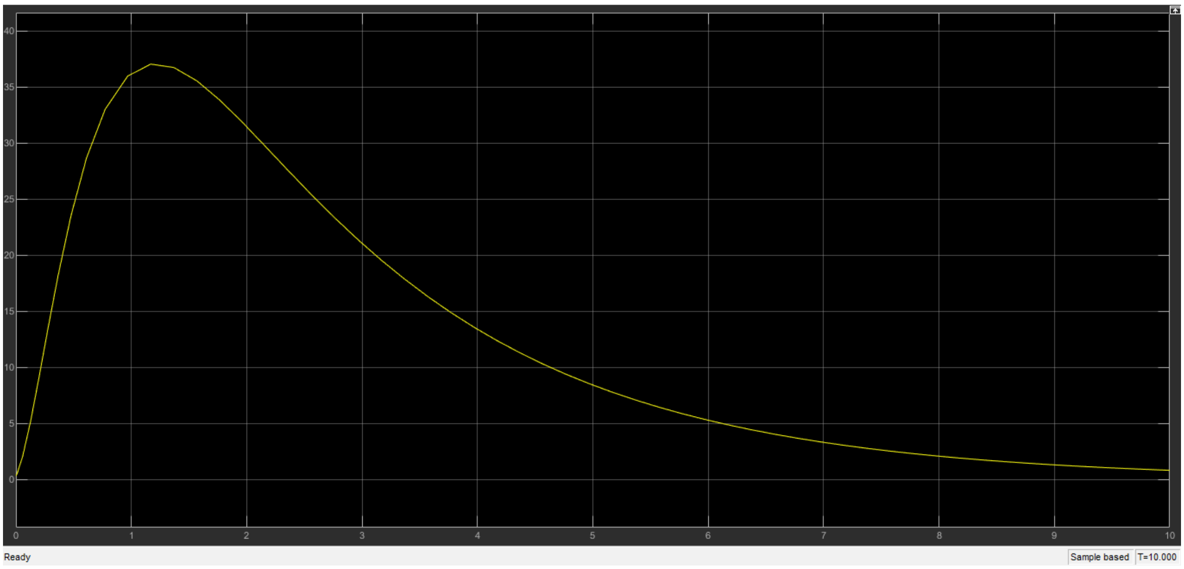
Παρ’ όλο που προστέθηκαν 2 επιπλέον μεταβλητές στο σύστημα μας(θ, θ’) γνωρίζουμε περίπου πως θα συμπεριφερθεί με βάση τις μετρήσεις που κάναμε στο προηγούμενο μέρος. Το κέρδος ανάδρασης KCI=[KCI1 KCI2 KCI3 KCI4] (Όπου τα C, I είναι για τις λέξεις Cart και Inverted Pendulum) πρέπει να οριστεί με ακρίβεια ώστε να επιστρέψει το βαγόνι στην θέση ισορροπίας του. Αν και μελετήσαμε αρκετά τυχαία παραδείγματα, παρόμοια με του προηγούμενου μέρους, σε αυτό το μέρος θα κάνουμε χρήση της μεταβλητής lqr(Linear-Quadratic Regulator) που μας δίνει το κατάλληλο κέρδος KCl έτσι ώστε να ελαχιστοποιηθεί το τετραγωνικό κόστος της παρακάτω συνάρτησης:

For a continuous time system, the state-feedback law $u = -Kx$ minimizes the quadratic cost function

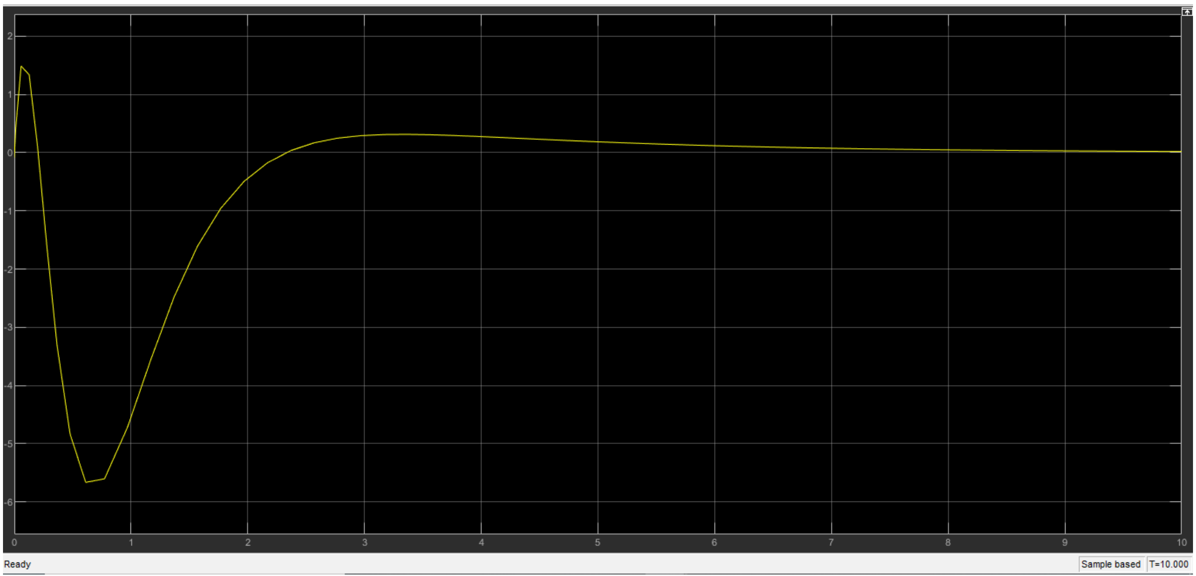
$$J(u) = \int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u)dt$$

Καλώντας την συνάρτηση KCl=lqr(A2,B2,Q,N) όπου A2,B2 δοσμένοι πίνακες και Q=I_4, N=0.1 προκύπτουν τα παρακάτω γραφήματα:

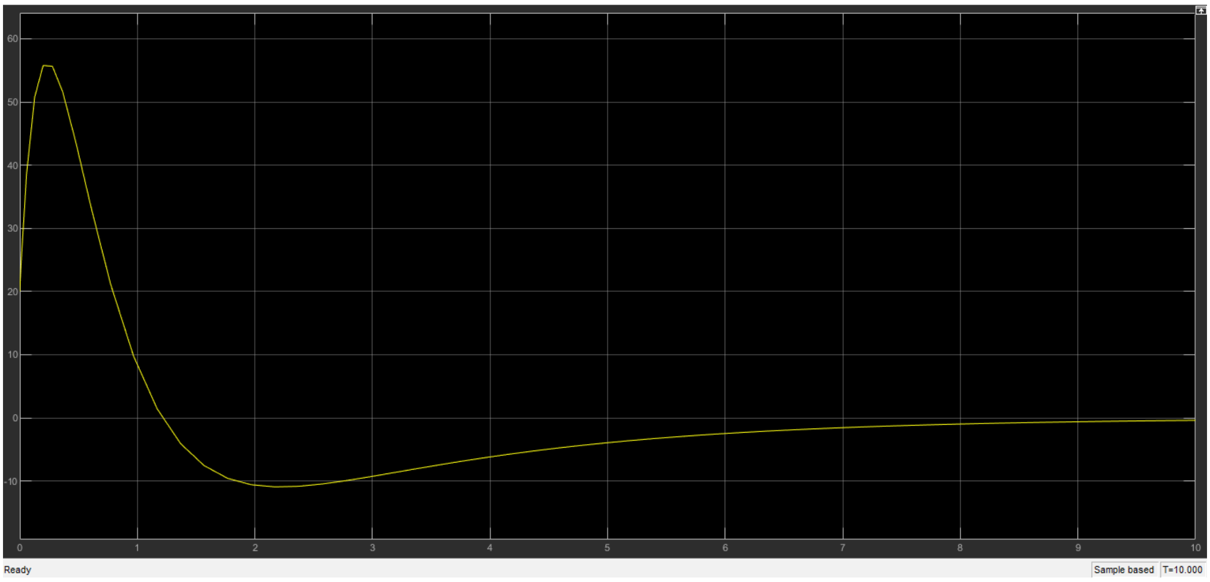
x



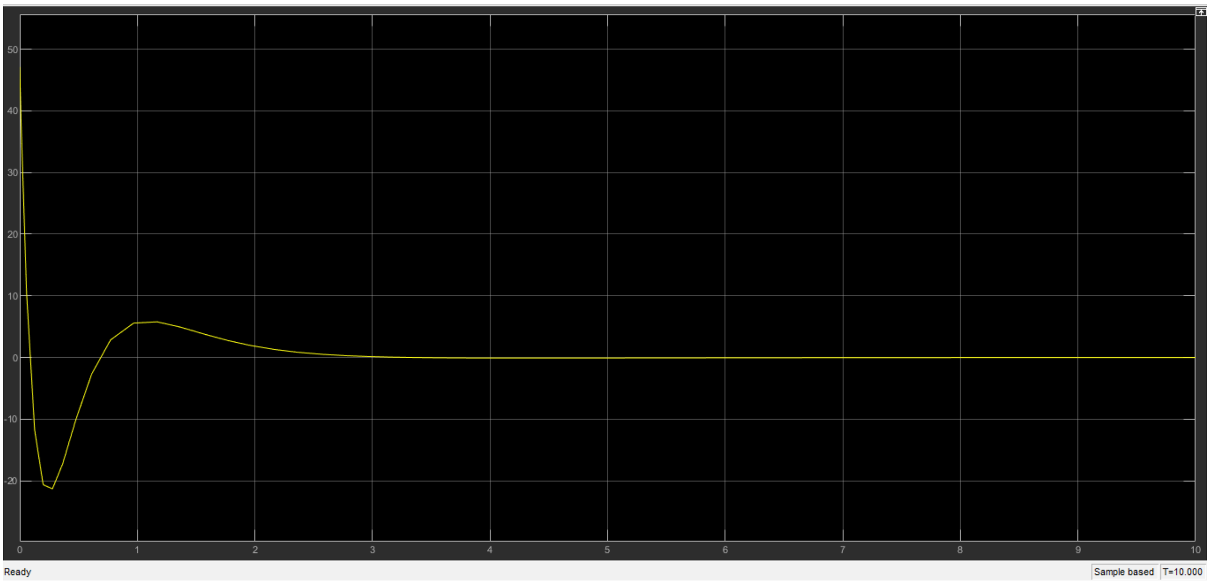
θ



x’

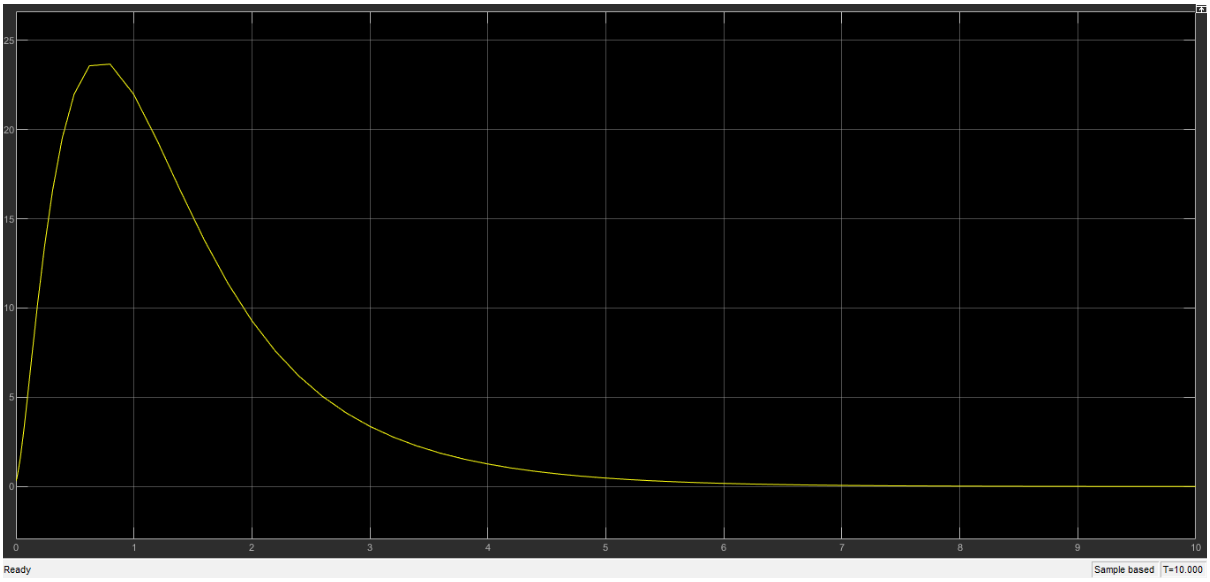


θ’

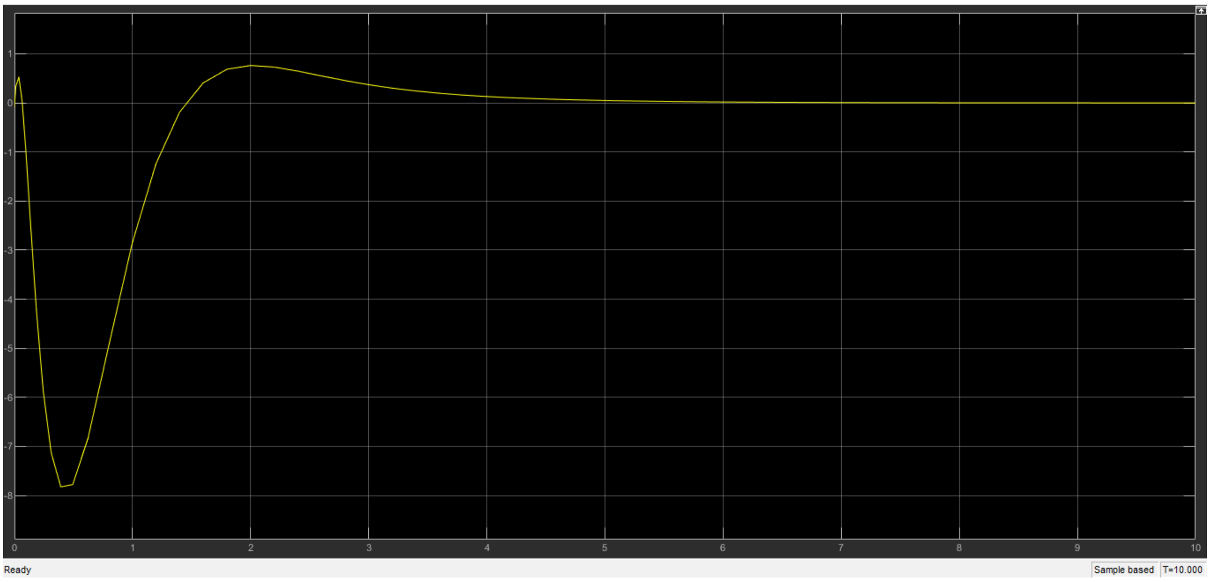


Αν επιχειρήσουμε να μειώσουμε το N ή να αυξήσουμε το Q βελτιώνεται ο χρόνος αποκατάστασης και μειώνονται οι ταλαντώσεις. Αυξάνεται βέβαια το input cost. Κάνουμε μια τελευταία δοκιμή για Q=5*I4 και N=0.01 και προκύπτουν:

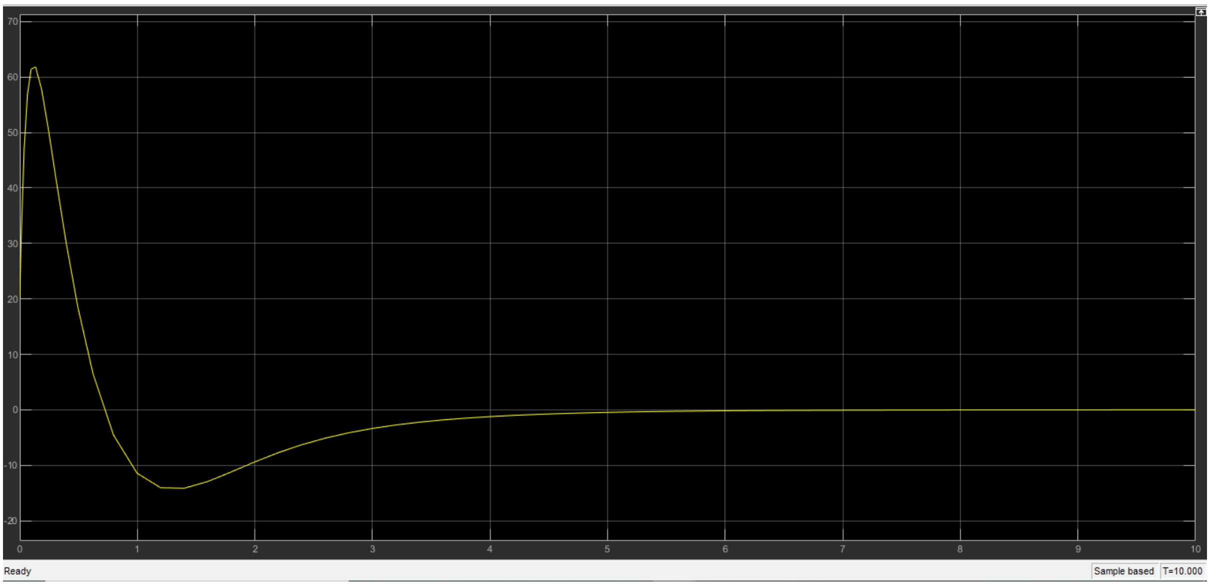
x


25
20
15
10
5
0
0
1
2
3
4
5
6
7
8
9
10
Ready
Sample based
T=10.000

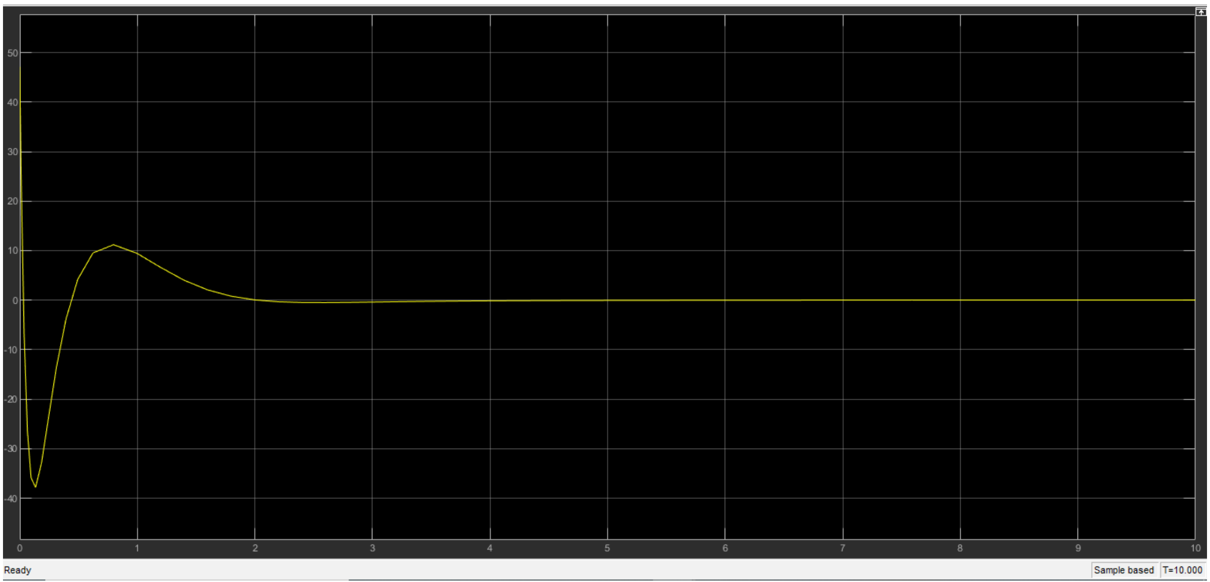
θ


1
0
-1
-2
-3
-4
-5
-6
-7
-8
0
1
2
3
4
5
6
7
8
9
10
Ready
Sample based
T=10.000

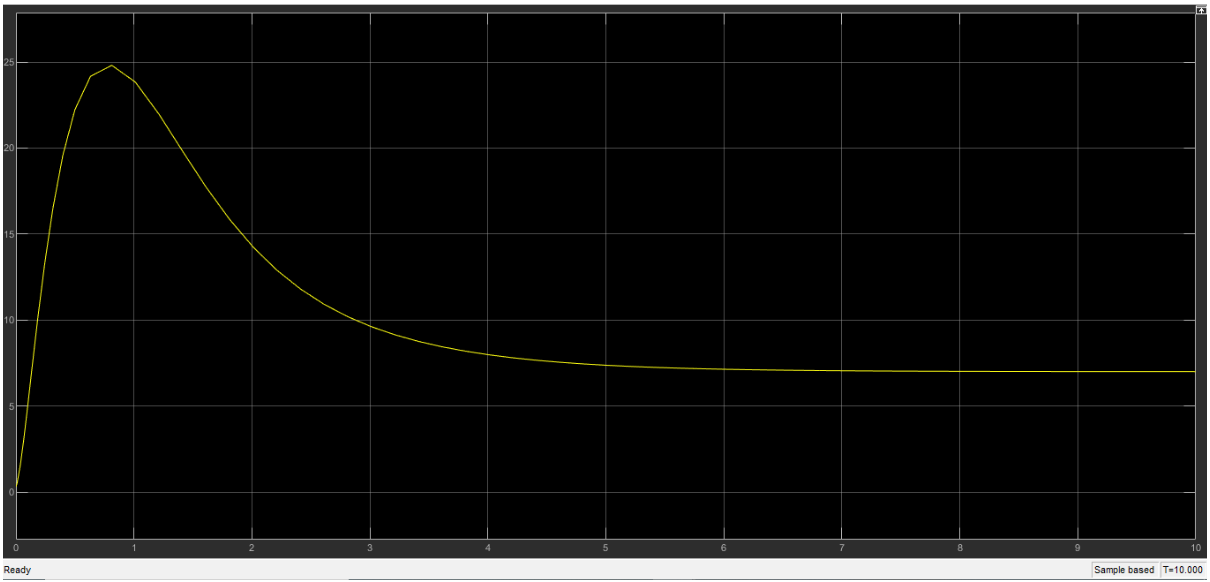
x’



θ’



Με αυτή την τελική σχεδίαση αν επιχειρούσαμε να πάμε το βαγόνι στην θέση x=7:



Συμπεραίνουμε λοιπόν ότι είναι μια καλή σχεδίαση. Αν είχαμε υψηλότερες απαιτήσεις(όπως γρηγορότερο χρόνο αποκατάστασης) τότε θα αυξάναμε και άλλο το feedback gain.

Γενικότερη παρατήρηση από το όλο πείραμα είναι ότι με την αύξηση των μεταβλητών του συστήματος ο έλεγχος γίνεται πιο δύσκολος και χρειάζεται περισσότερη προσοχή στο feedback, αλλά και στην υπόλοιπη σχεδίαση του συστήματος. Επιπλέον, αν έπρεπε να συγκρατήσουμε το βαγόνι εντός συγκεκριμένου ορίου τιμών(πχ [-1,1]) θα έπρεπε να γίνει πιο προσεκτική επιλογή του feedback gain ώστε να μην βγαίνει εκτός ορίων το βαγόνι.