



Συγκριτική Ανάλυση και Προσομοίωση του Συστήματος Ελέγχου Ταχύτητας Ταξιδιού δυο διαφορετικών μοντέλων αυτοκινήτων χρησιμοποιώντας PID Ελεγκτή

Ιωάννης Μίτρο
imitro@uth.gr

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ

Πανεπιστήμιο Θεσσαλίας

Βόλος, Ελλάδα

Υποβολή: 6 Φεβρουαρίου, 2020

Περίληψη

Τα τελευταία χρόνια, στον τομέα της αυτοκινητοβιομηχανίας, το σύστημα ελέγχου ταχύτητας ταξιδιού (cruise control system) είναι μια από τις πιο σημαντικές πτυχές που απαιτούν έναν αποτελεσματικό σχεδιασμό ενός ελεγκτή που μπορεί να υποστηρίξει τη νέα εξέλιξη σε αυτόν τον τομέα της τεχνολογίας. Σε αυτή την εργασία, το σύστημα ελέγχου ταχύτητας ταξιδιού του αυτοκινήτου θα διερευνηθεί στα πλαίσια του σχεδιασμού, της μοντελοποίησης και της προσομοίωσης. Η φυσική δομή του συστήματος θα βασίζεται σε δύο πρώτης τάξης μοντέλα αυτοκινήτου με παραμέτρους που αντιπροσωπεύουν τα μοντέλα των πραγματικών αυτοκινήτων 2019 BMW Σειρά 3(Μηχανή: 1,998 (Engine:1,998cm³ four-cylinder) και 2018 Audi A4(Engine:2.0 TFSI four-cylinder). Τα συστήματα αυτά θα πρέπει να προσομοιωθούν και να αποσκοπούν στη βελτίωση των χαρακτηριστικών απόδοσης του συστήματος. Επιπλέον, θα υπάρξει μια συγκριτική ανάλυση των PI και PID ελεγκτών για το σύστημα ελέγχου ταχύτητας ταξιδιού(cruise control system) μας χρησιμοποιώντας Matlab/ Simulink. Ο στόχος του συστήματος είναι να διατηρήσει την ταχύτητα του αυτοκινήτου στην ταχύτητα αναφοράς αυτού του μοντέλου. Για να λάβουμε τον επιθυμητό χρόνο ανόδου, θα προσαρμόσουμε τα κέρδη του κάθε ελεγκτή, έτσι ώστε να λάβουμε ικανοποιητικά αποτελέσματα που αντιστοιχούν στα δεδομένα μας. Με άλλα λόγια, ο κύριος στόχος αυτής της εργασίας επικεντρώνεται στην κατασκευή του βασικού μοντέλου των συμβατικών συστημάτων ελέγχου ταχύτητας(conventional cruise control systems) και των συναρτήσεων μεταφοράς τους, στην ανάλυση των επιδόσεων του συστήματος και στη διερεύνηση της απόκρισης εξόδου αυτών των μοντέλων.

Λέξεις-Κλειδιά : σύστημα ελέγχου ταχύτητας ταξιδιού, PI, PID ελεγκτές, Matlab/Simulink, κέρδη

Πίνακας Περιεχομένων

1. Εισαγωγή.....	4
2. Μοντελοποίηση Συστήματος.....	5
2.1 Φυσική Δομή.....	5
2.2 Εξισώσεις και Παράμετροι Συστήματος.....	7
2.3 Χώρος Κατάστασης και Συνάρτηση Μοντέλου.....	8
3. Ανάλυση Συστήματος.....	10
3.1 Μοντέλο Συστήματος, παράμετροι και προδιαγραφές απόδοσης.....	10
3.2 Βηματική απόκριση και πόλοι/ μηδενικά ανοιχτού βρόγχου.....	10
4. Σχεδιασμός PID Ελεγκτή.....	15
4.1 Μοντέλο Συστήματος, Παράμετροι και Προδιαγραφές απόδοσης.....	15
4.2 Επισκόπηση του PID Ελεγκτή.....	15
4.3 Έλεγχος με χρήση P (Proportional Control).....	17
4.4 Έλεγχος με χρήση PI.....	21
4.5 Έλεγχος με χρήση PID.....	24
4.6 Σύγκριση του PI και PID.....	27
5. Μοντελοποίηση Συστήματος και σχεδιασμός Ελεγκτή σε Simulink.....	36
5.1 Φυσική Δομή, Εξισώσεις Συστήματος και κατασκευή μοντέλου.....	36
5.2 Σχεδίαση P,PI,PID ελεγκτή.....	38
6. Συμπεράσματα και Προοπτικές.....	41

1. Εισαγωγή

Ο σκοπός ενός συστήματος ελέγχου ταχύτητας ταξιδιού είναι να διατηρεί με ακρίβεια την ταχύτητα που ορίζεται από τον οδηγό χωρίς καμία εξωτερική διαταραχή. Αυτό συμβαίνει με τον έλεγχο της σύνδεσης πεντάλ γκαζιού-επιταχυντή. Το σύστημα ελέγχου ταχύτητας ταξιδιού ελέγχει την ταχύτητα του αυτοκινήτου με τον ίδιο τρόπο με μας, ρυθμίζοντας τη θέση του γκαζιού. Ωστόσο, ο αυτόματος πιλότος εμπλέκει μια βαλβίδα με ένα καλώδιο που συνδέεται με τον ενεργοποιητή, κάτι το οποίο αντικαθιστά το πάτημα του πεντάλ. Η βαλβίδα αυτή που αναφέρθηκε ελέγχει την ισχύ και την ταχύτητα του κινητήρα περιορίζοντας το ποσοστό του αέρα που εισέρχεται μέσα σε αυτήν (αφού πρόκειται για έναν κινητήρα εσωτερικής καύσης). Χρησιμοποιώντας τα κουμπιά ελέγχου ταχύτητας ταξιδιού, η ταχύτητα του οχήματος ρυθμίζεται και αυτόματα διατηρείται στην απαιτούμενη ταχύτητα χωρίς την εφαρμογή του πεντάλ γκαζιού. Όταν το σύστημα ελέγχου ταχύτητας ταξιδιού ενεργοποιηθεί, το αυτοκίνητο τρέχει σταθερά και ως εκ τούτου, η περιττή μεταβολή της ταχύτητας μειώνεται [1]. Αυτό οδηγεί σε βελτιστοποίηση της κατανάλωσης καυσίμου και την ίδια στιγμή, αυξάνει την απόδοση του κινητήρα. Επίσης, η ρύπανση που προκαλείται από την εξάτμιση μειώνεται και η άνεση του οδηγού βελτιώνεται, δεδομένου ότι μειώνει την κούραση του σε μεγάλη απόσταση που ίσως θα πρέπει να διανύσει με το αυτοκίνητο. Ένα άλλο πλεονέκτημα της χρήσης συστημάτων ελέγχου ταχύτητας ταξιδιού είναι ότι μειώνουν την πιθανότητα της πιθανής συντριβής. Το σύστημα ελέγχου ταχύτητας ταξιδιού θεωρείται ως ένα από τα πιο σημαντικά χαρακτηριστικά που προστέθηκε στα αυτοκίνητα για να θεωρούνται «Αυτόματα Οχήματα». Τα τελευταία χρόνια, πολλές μελέτες για ευφυή (έξυπνα) οχήματα έχουν γίνει για να ξεπεραστούν προβλήματα που υπάρχουν, όπως ατυχήματα, η ροή της κυκλοφορίας και οι μειώσεις επιβαρύνσεων του οδηγού. Η οδήγηση στην πραγματικότητα αποτελεί μια πολύ απαιτητική δραστηριότητα και για αυτό το λόγο απαιτεί υψηλό επίπεδο συγκέντρωσης για μεγάλο χρονικό διάστημα. Επίσης, ο οδηγός πρέπει να αντιδράσει μέσα σε κλάσματα του δευτερολέπτου για να αλλάξει μια κατάσταση. Για αυτούς τους λόγους, το σύστημα ελέγχου ταχύτητας ταξιδιού χρησιμοποιείται για μεγάλες αποστάσεις για να βοηθήσει τον οδηγό. Υπάρχουν πολλές μέθοδοι που χρησιμοποιούνται για το σύστημα ελέγχου ταχύτητας. Μία από τις πιο δημοφιλείς μεθόδους είναι ο συμβατικός ελεγκτής PID, που θα αναλυθεί στην παρούσα εργασία. Για τη μέθοδο αυτή, θα πρέπει να επιλεγθεί η βέλτιστη τιμή του κέρδους για να έχουμε την καλύτερη δυνατή απόκριση.

2. Μοντελοποίηση Συστήματος

2.1 Φυσική Δομή

Πολλά οχήματα στις μέρες μας χρησιμοποιούν αυτόματο σύστημα ελέγχου ταχύτητας ταξιδιού, το οποίο είναι ένα σύστημα ελέγχου ανάδρασης. Ο κύριος ρόλος του συστήματος ελέγχου ταχύτητας ταξιδιού(cruise control system) είναι να διατηρηθεί σταθερή η ταχύτητα του οχήματος, μειώνοντας την επίδραση των διαταραχών που θα μπορούσαν να επηρεάσουν την ταχύτητα του οχήματος. Αυτές οι διαταραχές είναι οι αλλαγές στις καιρικές συνθήκες, όπως ο άνεμος ή αλλαγές στην κλίση του δρόμου. Προκειμένου να επιτευχθεί αυτό, η ταχύτητα του οχήματος συγκρίνεται με την ταχύτητα αναφοράς ή την επιθυμητή ταχύτητα και μέσω αυτής της σύγκρισης η βαλβίδα ρυθμίζεται αυτόματα χρησιμοποιώντας νόμους ελέγχου(control laws). Θεωρούμε ένα απλό μοντέλο της δυναμικής του οχήματος, όπως φαίνεται στο Σχήμα 1. Μια δύναμη ελέγχου f εφαρμόζεται στο όχημα με μάζα m . Η δύναμη ελέγχου αντιπροσωπεύει τη δύναμη, που παράγεται στην επιφάνεια δρόμου/ελαστικών. Για το μοντέλο μας θα υποθέσουμε ότι μπορούμε να ελέγξουμε αυτήν την δύναμη αγνοώντας τους πιθανούς παράγοντες, όπως συστήματα μετάδοσης κίνησης ή ελαστικά, που έχουν ως αποτέλεσμα την παραγωγή της δύναμης. Ο παράγοντας bv , ο οποίος θεωρείται ως δύναμη αντιστάσεως που προκαλείται από αντίσταση κύλισης και αντίσταση του αέρα, είναι γραμμική μεταβλητή που εξαρτάται από την ταχύτητα του οχήματος, v , και έχει αντίθετη κατεύθυνση της κίνησης του οχήματος.

Αρχικά, θα αρχικοποιήσουμε τα χαρακτηριστικά του μοντέλου μας στον Πίνακα 1 προκειμένου να εξαχθούν κάποια δεδομένα από τις μαθηματικές εξισώσεις, όπως ο συντελεστή απόσβεσης. Τα δεδομένα που χρησιμοποιούνται για την εξαγωγή των αποτελεσμάτων μας βρέθηκαν σε [2,3].



Σχήμα 1. Μοντέλο συστήματος

Πίνακας 1. Δεδομένα για τον υπολογισμό των παραμέτρων των συστημάτων	
<u>2019 BMW 3 Series</u>	<u>2018 Audi A4</u>
Μήκος (L) = 4716mm	Μήκος (L) = 4726mm
Πλάτος (W) = 1826mm	Πλάτος (W) = 1841mm
Ύψος (H) = 1442mm	Ύψος (H) = 1427mm
0-96.5 km/h σε 5.6 δευτερόλεπτα (0-26.8m/s)	0-96.5 km/h σε 7.1 δευτερόλεπτα (0-26.8m/s)
Μέγιστη ροπή : 400Nm@440rpm	Μέγιστη ροπή : 370Nm@4500 rpm
Βάρος οχήματος : 1624 kg	Βάρος οχήματος : 1645 kg
Πυκνότητα του αέρα (ρ) : 1.2 kg/m ³	Πυκνότητα του αέρα (ρ) : 1.2 kg/m ³
Συντελεστής αντίστασης λόγω αέρα (Cd) = 0.23	Συντελεστής αντίστασης λόγω αέρα (Cd) = 0.23
Συντελεστής αντίστασης κύλισης (Cr) = 0.0125	Συντελεστής αντίστασης κύλισης (Cr) = 0.0125
Επιθυμητή ταχύτητα, ταχύτητα αναφοράς (r): 26.8m/s	Επιθυμητή ταχύτητα, ταχύτητα αναφοράς (r) : 26.8m/s
Δύναμη αντίστασης, αεροδυναμική (Fd) = Cd*0.5*ρ*v ² *A = 0.23*0.5*1.2*2.6*28 ² =281N	Δύναμη αντίστασης, αεροδυναμική (Fd) = Cd*0.5*ρ*v ² *A = 0.23*0.5*1.2*2.6*28 ² =281N
Δύναμη αντίστασης κύλισης (Fr) = Cr*m*g=0.0125*1624*9.81=199N	Δύναμη αντίστασης κύλισης (Fr) = Cr*m*g=0.0125*1645*9.81=201N
Συνολική δύναμη αντίστασης (Fdrag) = Fd+Fr = 281+ 199 = 480N	Συνολική δύναμη αντίστασης (Fdrag) = Fd+Fr = 281+ 201 = 482 N
Συντελεστής απόσβεσης (b) = (Fdrag/r) = 480/26.8 = 17.91N*s/m	Συντελεστής απόσβεσης (b) = (Fdrag/r) = 482/26.8 = 17.98N*s/m

Ονομαστική δύναμη ελέγχου (f1) = Μέγιστη ροπή/ακτίνα ελαστικού = 480N προκειμένου να φτάσουμε την επιθυμητή ταχύτητα των 26.8 m/s	Ονομαστική δύναμη ελέγχου (f2) =Μέγιστη ροπή/ακτίνα ελαστικού = 482N προκειμένου να φτάσουμε την επιθυμητή ταχύτητα των 26.8 m/s
<u>Κριτήρια Σχεδίασης</u>	
Χρόνος ανύψωσης (Rise Time) < 7 secs	Χρόνος ανύψωσης (Rise Time) < 8 secs
Υπερύψωση (Overshoot) < 25%	Υπερύψωση (Overshoot) < 25%
Σφάλμα σταθερής κατάστασης (Steady State Error) < 7%	Σφάλμα σταθερής κατάστασης (Steady State Error) < 7%

2.2 Εξισώσεις και Παράμετροι Συστήματος

Με τα παραπάνω δεδομένα έχουμε ένα σύστημα πρώτης τάξης. Αθροίζοντας τις δυνάμεις στην κατεύθυνση-x και εφαρμόζοντας τον 2ο νόμο του Newton, παίρνουμε την ακόλουθη εξίσωση:

$$m\dot{v} + bv = force = f \quad (1)$$

Εφόσον ενδιαφερόμαστε για τον έλεγχο της ταχύτητας του οχήματος, η εξίσωση εξόδου επιλέγεται να είναι η παρακάτω:

$$y = velocity = v \quad (2)$$

Οι παράμετροι των συστημάτων όπως υπολογίστηκαν και φαίνονται στον πίνακα παραπάνω θα είναι:

- (m1) μάζα οχήματος 1(vehicle mass of first model) 1624 kg
- (b1) συντελεστής απόσβεσης πρώτου μοντέλου(damping coefficient of first model) 17.91 N.s/m
- (f1) ονομαστική δύναμη ελέγχου πρώτου μοντέλου(nominal control force of first model) 480 N
- (m2) μάζα οχήματος δεύτερου μοντέλου (vehicle mass of second model) 1645 kg
- (b2) συντελεστής απόσβεσης δεύτερου μοντέλου(damping coefficient of second model) 17.98 N.s/m
- (f2) ονομαστική δύναμη ελέγχου δεύτερου μοντέλου(nominal control force of second model) 482 N

2.3 Χώρος Κατάστασης και Συνάρτηση Μεταφοράς Μοντέλου

Τα συστήματα πρώτης τάξης έχουν μόνο μια κατάσταση αποθήκευσης ενέργειας, στην περίπτωση αυτή την κινητική ενέργεια του αυτοκινήτου, και για αυτό μόνο μια μεταβλητή κατάστασης χρειάζεται, η ταχύτητα. Η αναπαράσταση του χώρου κατάστασης έτσι είναι:

$$\dot{x} = [\dot{v}] = \left[\frac{-b}{m} \right] [v] + \left[\frac{1}{m} \right] [f] \quad (3)$$

$$y = [1][v] \quad (4)$$

Εισάγουμε αυτό το μοντέλο χώρου κατάστασης για τα δύο μας μοντέλα στο MATLAB χρησιμοποιώντας τις ακόλουθες εντολές [4]:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%State-space model

m1 = 1624; %%vehicle mass(in kg) for 2019 BMW3 Series
b1 = 17.91; %% damping coefficient(in Newton*sec/meters)
f1 = 480; %% nominal control force(in Newton)
m2 = 1645; %%vehicle mass for 2018 Audi A4(in kg)
b2 = 17.98; %% damping coefficient(in Newton*sec/meters)
f2 = 482; %% nominal control force(in Newton)
r = 26.8; %% reference speed(m/s),it's the same for our two models

A1 = -b1/m1;
B1 = 1/m1;
A2 = -b2/m2;
B2 = 1/m2;
C = 1;
D = 0;

cruise_model_ss1 = ss(A1,B1,C,D);%%creates a state-space model object
representing the continuous-time state-space model
cruise_model_ss2 = ss(A2,B2,C,D);

```

Λαμβάνοντας τον μετασχηματισμό Laplace και υποθέτοντας μηδενικές αρχικές συνθήκες, βρίσκουμε πως η συνάρτηση μεταφοράς του συστήματος ελέγχου ταχύτητας ταξιδιού είναι:

$$P(s) = \frac{V(s)}{F(s)} = \frac{1}{ms+b} \left[\frac{m/s}{N} \right] \quad (5)$$

Εισάγουμε τις συναρτήσεις μεταφοράς των μοντέλων στο MATLAB χρησιμοποιώντας τις ακόλουθες εντολές [4]:


```
%Transfer function model
```

```
s = tf('s');
```

```
P_transfer_cruise1 = 1/(m1*s+b1);
```

```
P_transfer_cruise2 = 1/(m2*s+b2);
```

3. Ανάλυση Συστήματος

3.1 Μοντέλα συστήματος, παράμετροι και προδιαγραφές απόδοση

Η συνάρτηση μεταφοράς του μοντέλου για το σύστημα ελέγχου ταχύτητας ταξιδιού (cruise control) πρόβλημα δίνεται στη σχέση (5) που υπολογίσαμε παραπάνω. Οι παράμετροι που χρησιμοποιούνται είναι αυτές που φαίνονται στην Ενότητα 2.2 παραπάνω.

Το επόμενο μας βήμα είναι να βρούμε κάποια σχεδιαστικά κριτήρια που το σύστημα μας πρέπει να πληροί. Όταν η μηχανή του αυτοκινήτου δημιουργεί μια δύναμη των 480 Newton στην πρώτη περίπτωση, το αυτοκίνητο θα φτάσει σε μια μέγιστη ταχύτητα των 26.8m/s, όπως φαίνεται και από την βηματική απόκριση ανοιχτού βρόγχου παρακάτω. Ένα αυτοκίνητο πρέπει να είναι σε θέση να επιταχύνει μέχρι αυτή την ταχύτητα σε λιγότερο από 7 δευτερόλεπτα στην πρώτη μας περίπτωση. Σε αυτή την εφαρμογή, ένα ποσοστό 25% υπερύψωσης (overshoot) και 7% σφάλματος σταθερής κατάστασης (steady-state error) είναι επαρκή. Παρομοίως, όταν η μηχανή δημιουργεί μια δύναμη 482 Newton στη δεύτερη περίπτωση, το αυτοκίνητο φτάνει στη μέγιστη ταχύτητα των 26.8m/s. Σε αυτή την περίπτωση, το αυτοκίνητο πρέπει να είναι σε θέση να επιταχύνει μέχρι αυτή την ταχύτητα σε λιγότερο από 8 δευτερόλεπτα. Και σε αυτή την εφαρμογή, ένα ποσοστό 25% υπερύψωσης (overshoot) και 7% σφάλματος σταθερής κατάστασης (steady-state error) είναι επαρκή. Έχοντας τα παραπάνω στο μυαλό, προτείνουμε τα παρακάτω σχεδιαστικά κριτήρια για το πρόβλημα μας:

2019 BMW3 Series

- Χρόνος ανύψωσης (Rise time) < 7 δευτερόλεπτα
- Υπερύψωση (Overshoot) < 25%
- Σφάλμα σταθερής κατάστασης (Steady-state error) < 7%

2018 Audi A4

- Χρόνος ανύψωσης (Rise time) < 8 δευτερόλεπτα
- Υπερύψωση (Overshoot) < 25%
- Σφάλμα σταθερής κατάστασης (Steady-state error) < 7%

3.2 Βηματική απόκριση και πόλοι/μηδενικά ανοιχτού βρόγχου

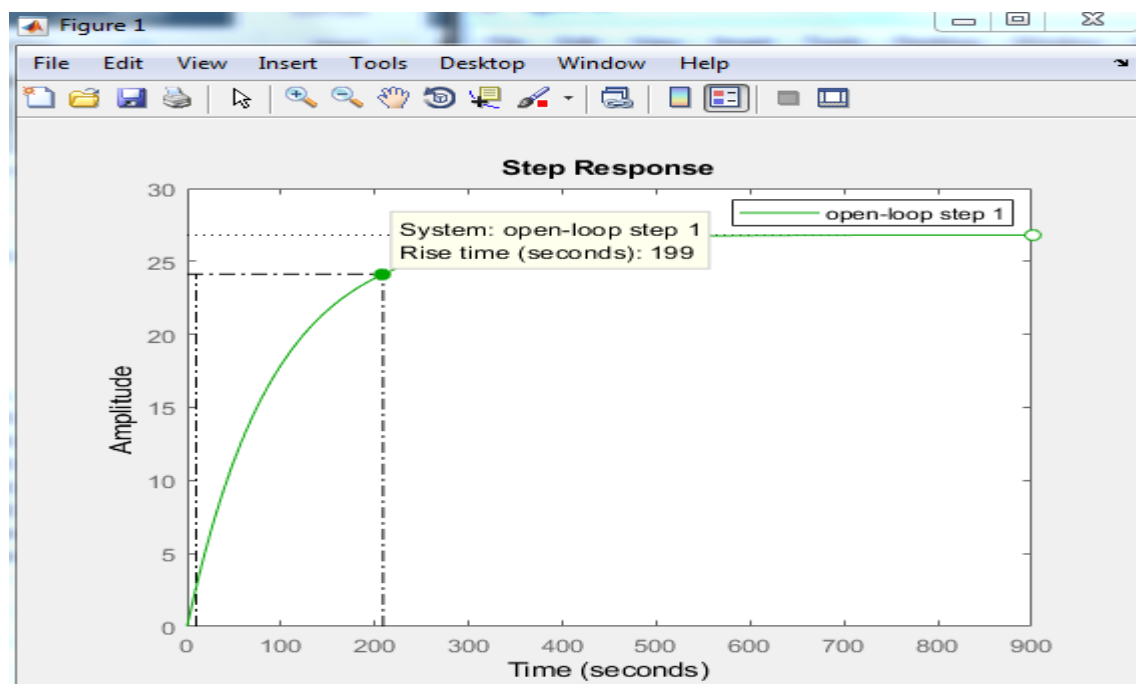
Η απόκριση ανοιχτού βρόγχου των συστημάτων, χωρίς έλεγχο ανατροφοδότησης, σε μια δύναμη εισόδου των 480 Newton στην πρώτη και των 482 Newton στην δεύτερη περίπτωση προσομοιώνεται στο MATLAB ακολούθως [4]:

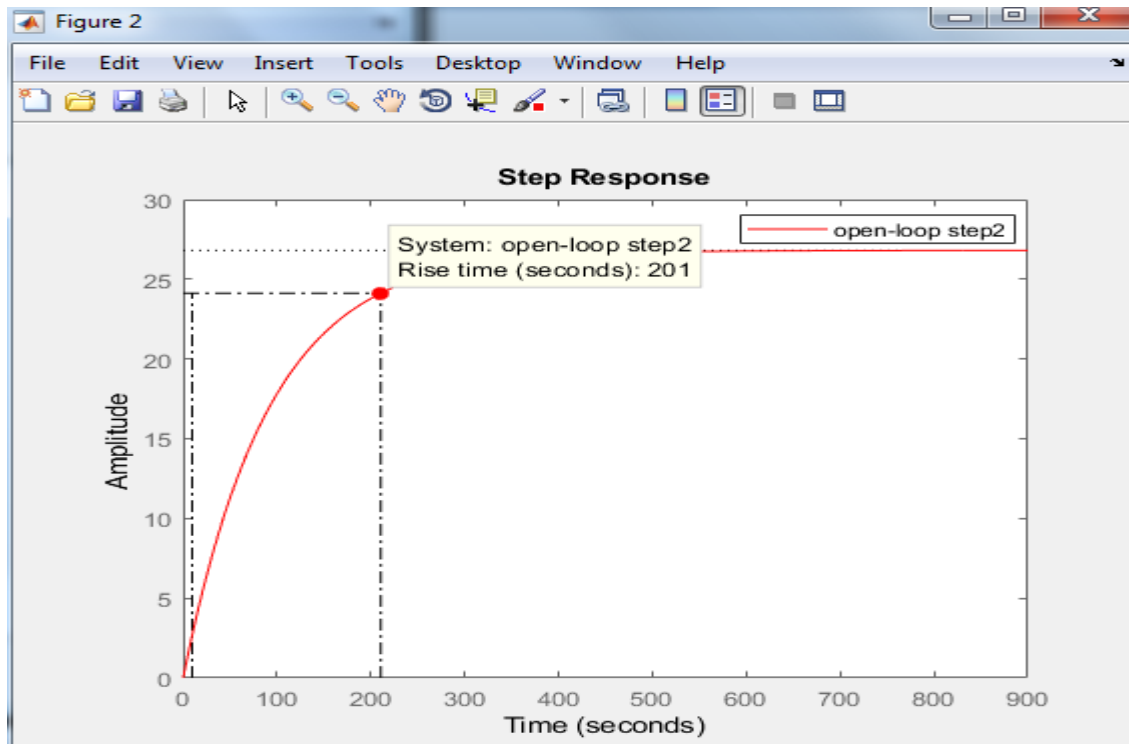
```

%%%%%%%%%System Analysis%%%%%%%%%

%The open-loop step response
open_step_response1 = f1*P_transfer_cruise1;
open_step_response2 = f2*P_transfer_cruise2;
figure(1)
%%compare the step-responses of the open-loop system and specify plot colors
and styles for each response
step(open_step_response1,'g-')
legend("open-loop step 1")
figure(2)
step(open_step_response2,'r-')
legend("open-loop step2")

```



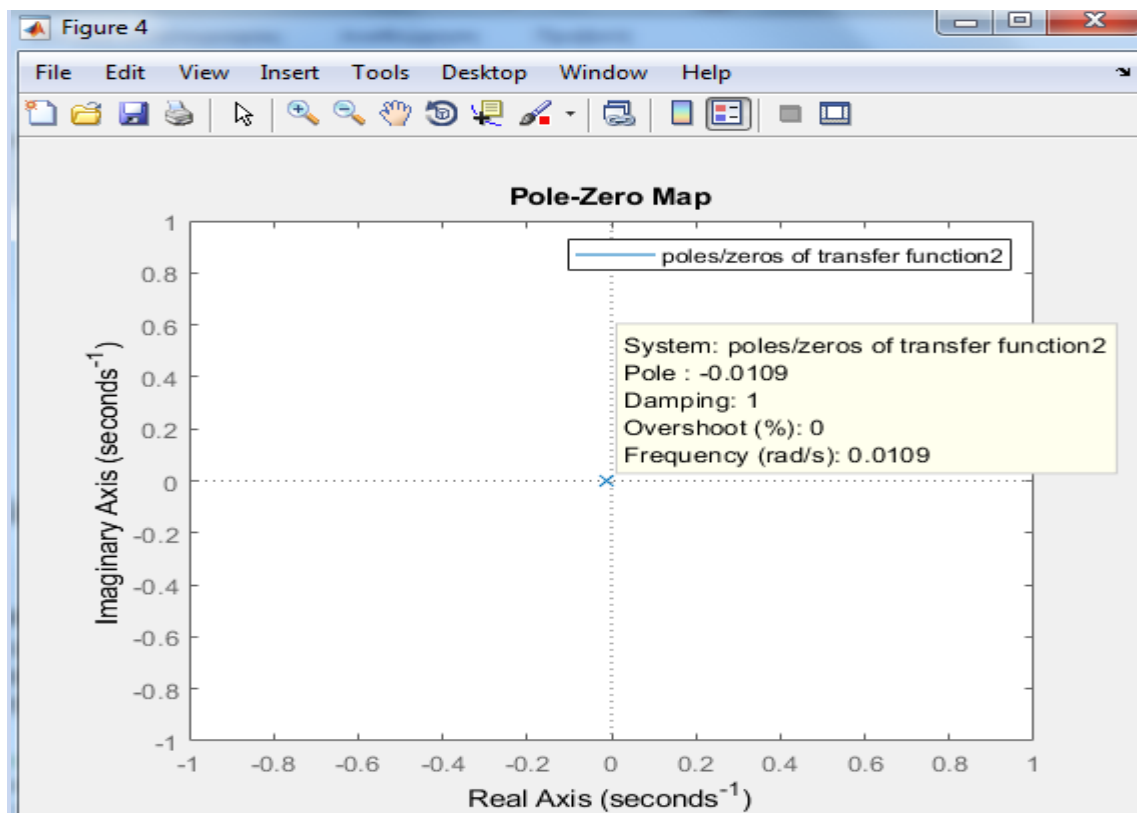
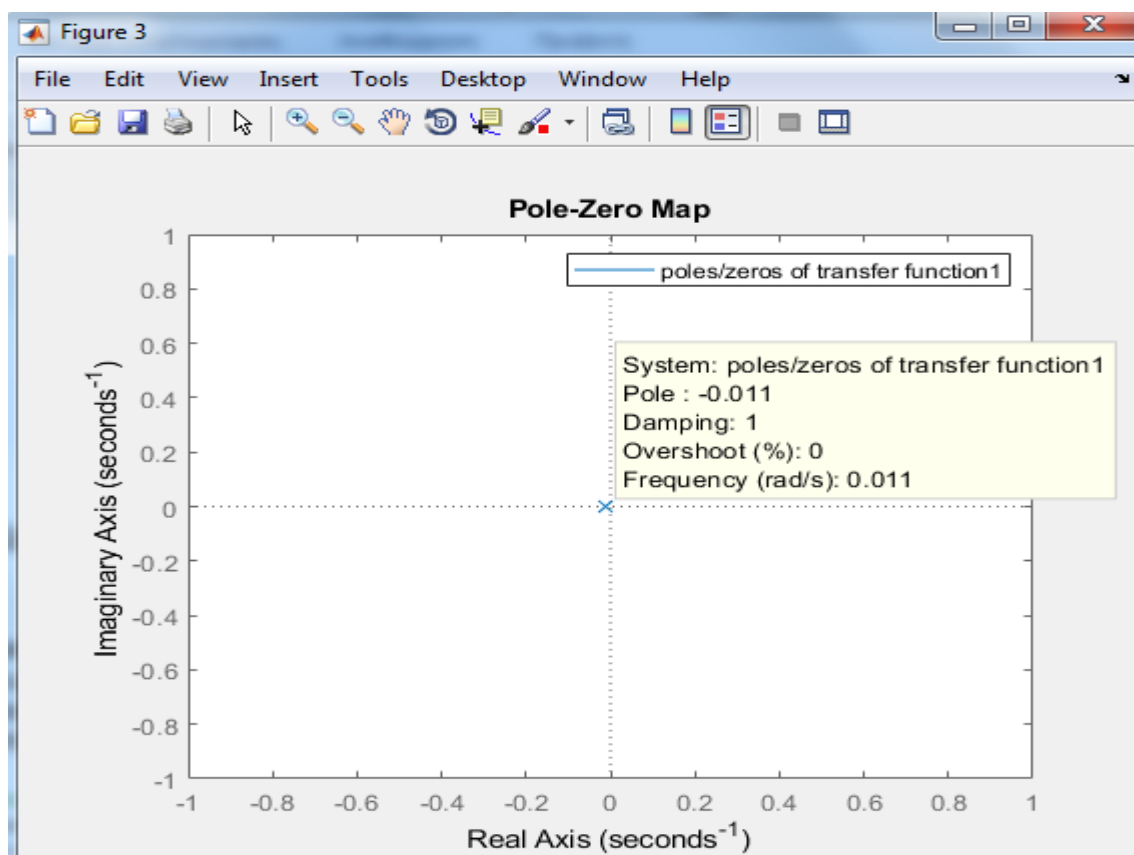


Βλέπουμε ότι τα δύο συστήματα ανοικτού βρόγχου δεν επιδεικνύουν καμία υπερύψωση ή ταλαντώσεις, και φθάνουν την επιθυμητή ταχύτητα σταθερής κατάστασης των 26.8 m /s. Ωστόσο, ο χρόνος ανόδου είναι πολύ αργός, 199 s για την πρώτη περίπτωση μας και 201 s για την δεύτερη περίπτωση . Προκειμένου να εξάγουμε τα αποτελέσματα προσεγγιστικά μεγεθύνουμε το σχήμα μας για να βρούμε τις κατάλληλες τιμές μετά από δοκιμές. Επομένως, πρέπει να σχεδιάσουμε το σύστημα μας εισάγοντας έναν ελεγκτή-ανάδρασης στην ανάδραση του κλειστού βρόγχου. Η εισαγωγή του ελεγκτή επιταχύνει την απόκριση σημαντικά, ενώ η ανάδραση έχει ως αποτέλεσμα την μείωση του σφάλματος σταθερής κατάστασης καθώς και την μείωση της επίδρασης διαταραχών .

Το επόμενο μας βήμα είναι ο έλεγχος της ευστάθειας του συστήματος μας. Αυτό θα γίνει με την εύρεση των πόλων και μηδενικών του συστήματος. Το σύστημα cruise control έχει έναν πόλο στο $\text{pole} = -b/m$, το οποίο μπορούμε να δούμε σχεδιασμένο στο διάγραμμα πόλων μηδενικών παρακάτω χρησιμοποιώντας τις ακόλουθες εντολές του MATLAB [4]:

```
%Open-loop poles/zeros
```

```
figure(3)
pzmap(P_transfer_cruise1)%if we want to get the values p,z--->poles,zeros
axis([-1 1 -1 1])
legend("poles/zeros of transfer function1")
figure(4)
pzmap(P_transfer_cruise2)%if we want to get the values p,z--->poles,zeros
axis([-1 1 -1 1])
legend("poles/zeros of transfer function2")
```



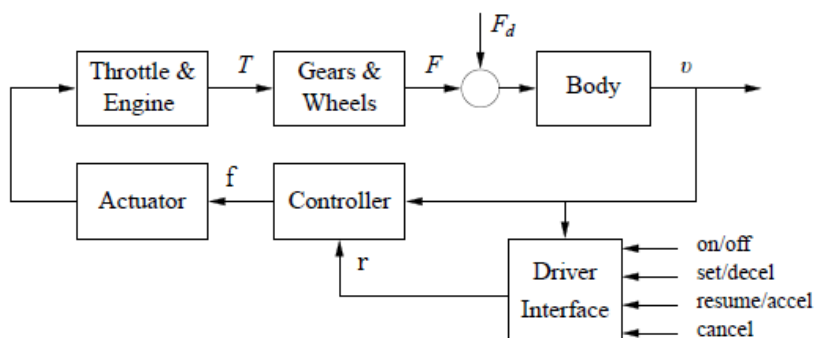
Οι πόλοι είναι πραγματικοί και αρνητικοί και στις δύο περιπτώσεις (-0.011 στην πρώτη περίπτωση και -0.0109 στη δεύτερη περίπτωση, όπως μπορούμε να δούμε πιο πάνω), οπότε παρατηρούμε ότι το σύστημα ανοιχτού βρόγχου είναι σταθερό και δεν ταλαντώνεται. Επιπλέον, η ταχύτητα της απόκρισης καθορίζεται από το μέγεθος αυτού

του πόλου, b / m : όσο μεγαλύτερο είναι το μέγεθος, τόσο πιο γρήγορα το σύστημα πλησιάζει την τιμή σταθερής κατάστασης. Λόγω του γεγονότος ότι δεν είμαστε σε θέση να αλλάξουμε τις παραμέτρους του συστήματος για να αλλάξουμε την δυναμική απόκριση του συστήματος, θα πρέπει αντί αυτού να σχεδιάσουμε ελεγκτές που μεταβάλλουν τους πόλους και τα μηδενικά του συστήματος κλειστού βρόχου για να καλύψουμε τις επιθυμητές προδιαγραφές απόδοσης [5].

4 Σχεδιασμός PID Ελεγκτή

4.1 Μοντέλο συστήματος και παράμετροι

Όπως φαίνεται και στο Σχήμα 2, ο ελεγκτής λαμβάνει τα σήματα ταχύτητας εξόδου (v) και επιθυμητής ταχύτητας ή ταχύτητα αναφοράς (r) και παράγει ένα σήμα ελέγχου f που στέλνεται στον ενεργοποιητή, ο οποίος ελέγχει την θέση του πεντάλ γκαζιού. Το πεντάλ γκαζιού με την σειρά του ελέγχει την ροπή T που στέλνεται μέσω της μηχανής και η οποία ροπή μεταδίδεται μέσω των ελαστικών και των γραναζιών, παράγοντας μια δύναμη F που κινεί το αυτοκίνητο. Η δύναμη αυτή στην ουσία προκαλείται από την επαφή των ελαστικών με την επιφάνεια του δρόμου. Μπορεί να υπάρχουν δυνάμεις διαταραχής, όπως η F_d λόγω αλλαγών στη κλίση του εδάφους, καθώς επίσης και δυνάμεις αντίστασης κύλισης ή δυνάμεις που προκαλούνται λόγω του αέρα [6].



Σχήμα 2. Διάγραμμα λειτουργίας συστήματος ελέγχου ταχύτητας ταξιδιού ενός αυτοκινήτου με χρήση ελεγκτή.

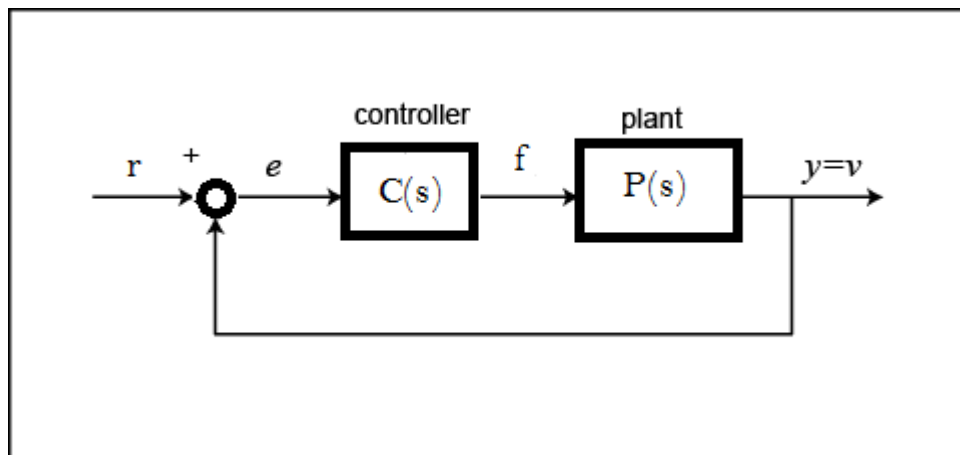
Η συνάρτηση μεταφοράς του μοντέλου για το δική μας εργασία δίνεται στη σχέση (5) που υπολογίστηκε παραπάνω. Οι παράμετροι που χρησιμοποιούνται σε αυτή την ενότητα είναι οι m_1, b_1, m_2, b_2 και η ταχύτητα αναφοράς που φαίνεται παρακάτω:

- ταχύτητα αναφοράς ή επιθυμητή έξοδος (r) 26.8 m/s

Οι προδιαγραφές απόδοσης θα είναι ίδιες με αυτές της Ενότητας 3.1

4.2 Επισκόπηση PID

Στο Σχήμα 3 που φαίνεται παρακάτω απεικονίζεται το σύστημα κλειστού βρόγχου μοναδιαίας ανατροφοδότησης [7].



Σχήμα 3. Σύστημα κλειστού βρόγχου μοναδιαίας ανατροφοδότησης

Οι μεταβλητές του παραπάνω διαγράμματος είναι η επιθυμητή έξοδος r (reference velocity), η μεταβλητή e που είναι το σφάλμα, το οποίο αναπαριστά τη διαφορά μεταξύ της επιθυμητής εξόδου r και της ακριβούς εξόδου του συστήματος που λαμβάνουμε y . Η μεταβλητή f είναι η ονομαστική δύναμη ελέγχου που πρόκειται ουσιαστικά για ένα σήμα ελέγχου.

Η συνάρτηση μεταφοράς του PID ελεγκτή είναι:

$$C(s) = K_p + \frac{K_i}{s} + K_d * s = \frac{s^2 + K_p * s + K_i}{s}$$

Ένας PID ελεγκτής στο MATLAB μπορεί να οριστεί χρησιμοποιώντας την συνάρτηση μεταφοράς απευθείας:

```
Kp1 = 9484.9;%rise time at 0.375 and 26.8m/s in 5.57 secs which is
acceptable(for 5,6 secs) and peak response in 20 secs
Ki1 = 1;
Kd1 = 1;
```

```
Kp2 = 9520;%26.8 m/s in 7.09(in 7,1 secs) secs which is accepable and rise
time 0.378 secs and peak resposnse 20 secs
Ki2 = 1;
Kd2 = 1;
```

```
s = tf('s');
C_PID1 = Kp1 + Ki1/s + Kd1*s
C_PID2 = Kp2 + Ki2/s + Kd2*s
```

Όπου C_PID είναι η συνάρτηση μεταφοράς συνεχούς χρόνου με τιμή:

$$C_{pid1} = \frac{9484.9s^2 + s + 1}{s}$$

$$C_{pid2} = \frac{9520s^2 + s + 1}{s}$$

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε την συνάρτηση `pid` του MATLAB προκειμένου να παράγουμε έναν αντίστοιχο ελεγκτή συνεχούς χρόνου όπως ακολουθεί:

```
C_PID_altern1 = pid(Kp1,Ki1,Kd1)
C_PID_altern2 = pid(Kp2,Ki2,Kd2)
```

που είναι ένας PID ελεγκτής συνεχούς χρόνου σε παράλληλη μορφή με τιμή:

$$C_{pidaltern} = K_p + K_i * \frac{1}{s} + K_d * s$$

με $K_{p1} = 9484.9, K_{i1} = K_{d1} = 1$ και $K_{p2} = 9520, K_{i2} = K_{d2} = 1$

4.3 Αναλογικός Έλεγχος (Proportional Control)

Το πρώτο πράγμα που πρέπει να γίνει σε αυτό το πρόβλημα είναι να βρούμε μια συνάρτηση μεταφοράς κλειστού βρόγχου με έναν αναλογικό έλεγχο ($C=K_p$) να προστεθεί σε αυτή. Μειώνοντας την μοναδιαία ανατροφοδότηση του διαγράμματος τετραγώνων, η συνάρτηση μεταφοράς με αναλογικό ελεγκτή γίνεται:

$$T(s) = \frac{Y(s)}{R(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{K_p}{m * s + b + K_p}$$

Ένας αναλογικός ελεγκτής, K_p , μειώνει τον χρόνο ανόδου, που είναι επιθυμητό σε αυτή την περίπτωση.

Για τη δεδομένη στιγμή, θα χρησιμοποιήσουμε ένα αναλογικό κέρδος $K_{p_proportional1}$ ίσο με 9582, $K_{p_proportional2}=9620$ και μια ταχύτητα αναφοράς 26.8 m/s

Στο αρχείο μας εισάγουμε τις παρακάτω εντολές:

```
Kp_proportional1 = 9582; %%26.8 m/s at 2.77 with rise time 0.37 and peak
response in 6.2 secs
C_proportional1 = pid(Kp_proportional1);
Kp_proportional2 = 9620; %%peak response at 6.3 seconds 26.8 m/s and rise time
0.372 secs and 26.8m/s in 2.68 secs
C_proportional2 = pid(Kp_proportional2);

T1_proportional = feedback(C_proportional1*P_transfer_cruise1,1)
```

```

T2_proportional = feedback(C_proportional2*P_transfer_cruise2,1)
step1_proportional = r*T1_proportional
step2_proportional = r*T2_proportional

t = 0:0.1:20;

figure(7)
step(step1_proportional,t)
legend("step1proportional")
axis([0 20 0 26.8])%we use the following axes for our desired reference
velocity

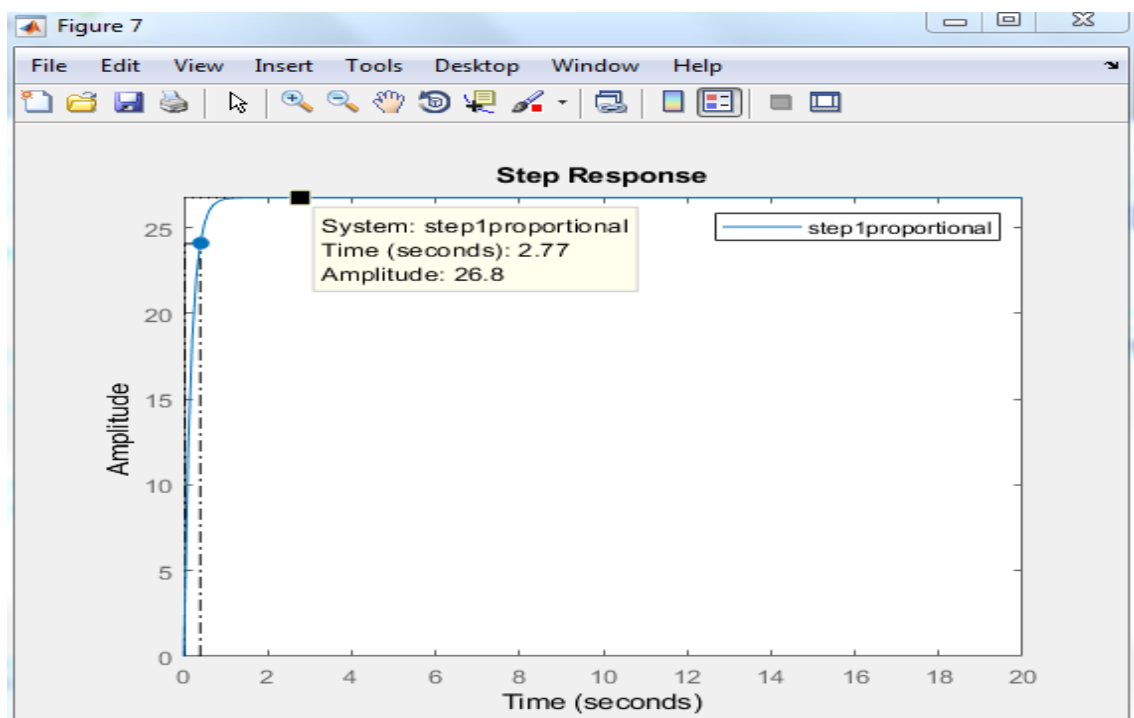
figure(8)
step(step2_proportional,t)
legend("step2proportional")
axis([0 20 0 26.8])

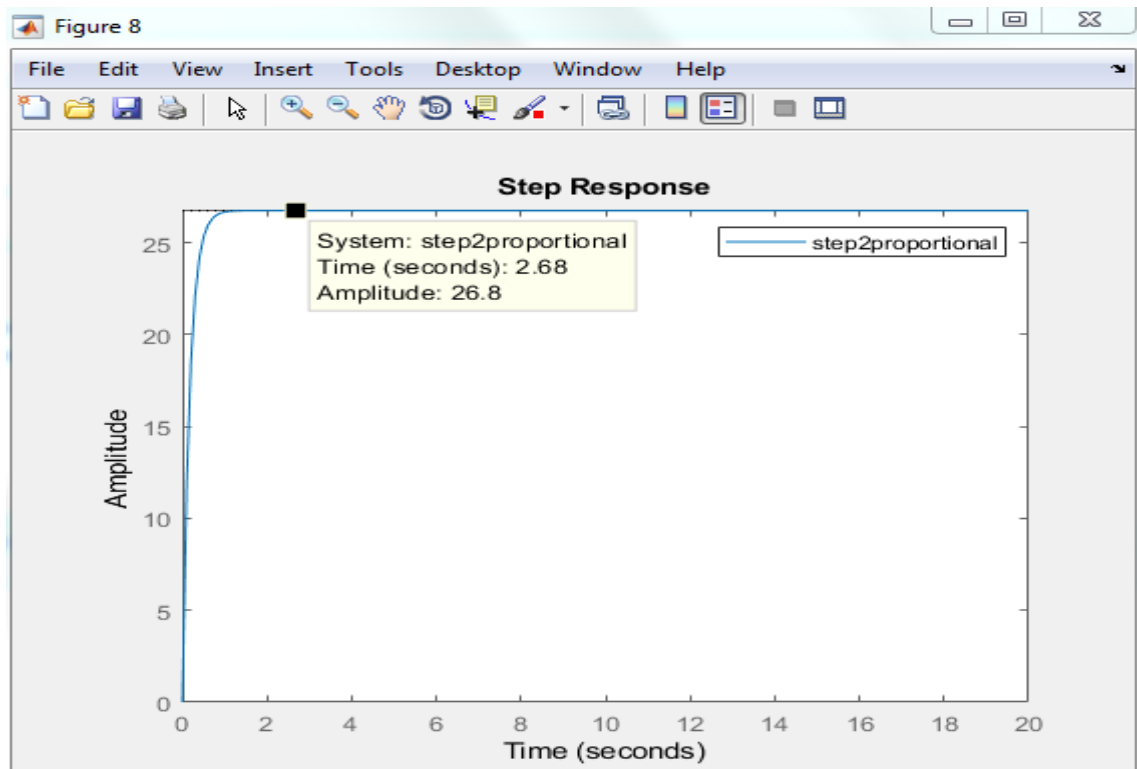
```

όπου $T1_{proportional}$ και $T2_{proportional}$ είναι συναρτήσεις μεταφοράς συνεχούς χρόνου με τιμή και βηματικές αποκρίσεις που φαίνονται παρακάτω:

$$T1_{proportional} = \frac{200}{1624 * s + 217.9}$$

$$T2_{proportional} = \frac{200}{1645 * s + 218}$$





Αξίζει να σημειωθεί πως χρησιμοποιήσαμε την εντολή `feedback` του MATLAB για να απλοποιήσουμε τη μείωση του διαγράμματος τετραγώνων του συστήματος κλειστού βρόγχου [4]. Τρέχοντας το m-file στο MATLAB παίρνουμε την βηματική απόκριση παρακάτω [4]. Όπως φαίνεται και στη γραφική παράσταση, ούτε το σφάλμα σταθερής κατάστασης ούτε ο χρόνος ανόδου ικανοποιούν τα σχεδιαστικά μας κριτήρια. Μπορούμε να αυξήσουμε το αναλογικό κέρδος K_p , για να μειώσουμε το χρόνο ανόδου και το σφάλμα σταθερής κατάστασης. Αλλάζουμε την τιμή του $K_{p_proportional1}$ στο 10550 και του $K_{p_proportional2}$ στο 9650 και λαμβάνουμε το ακόλουθο σχήμα :

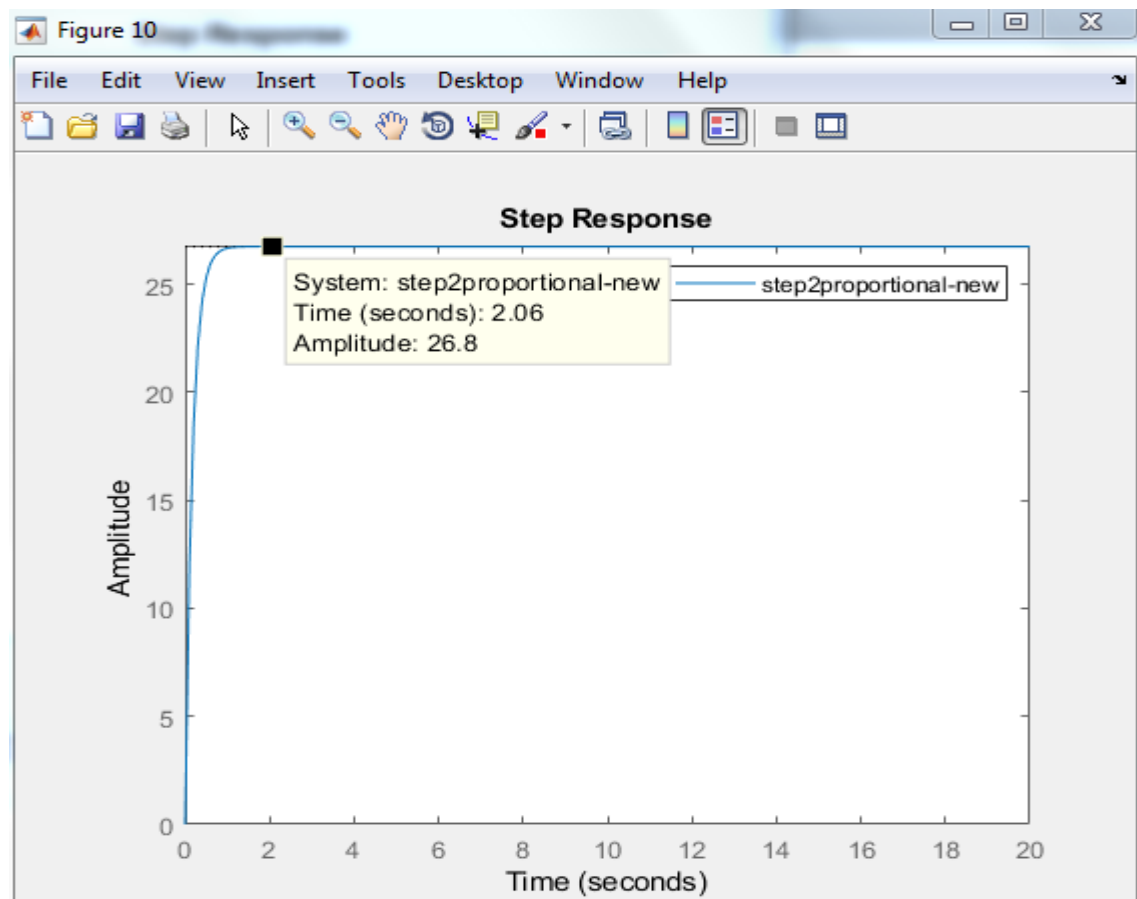
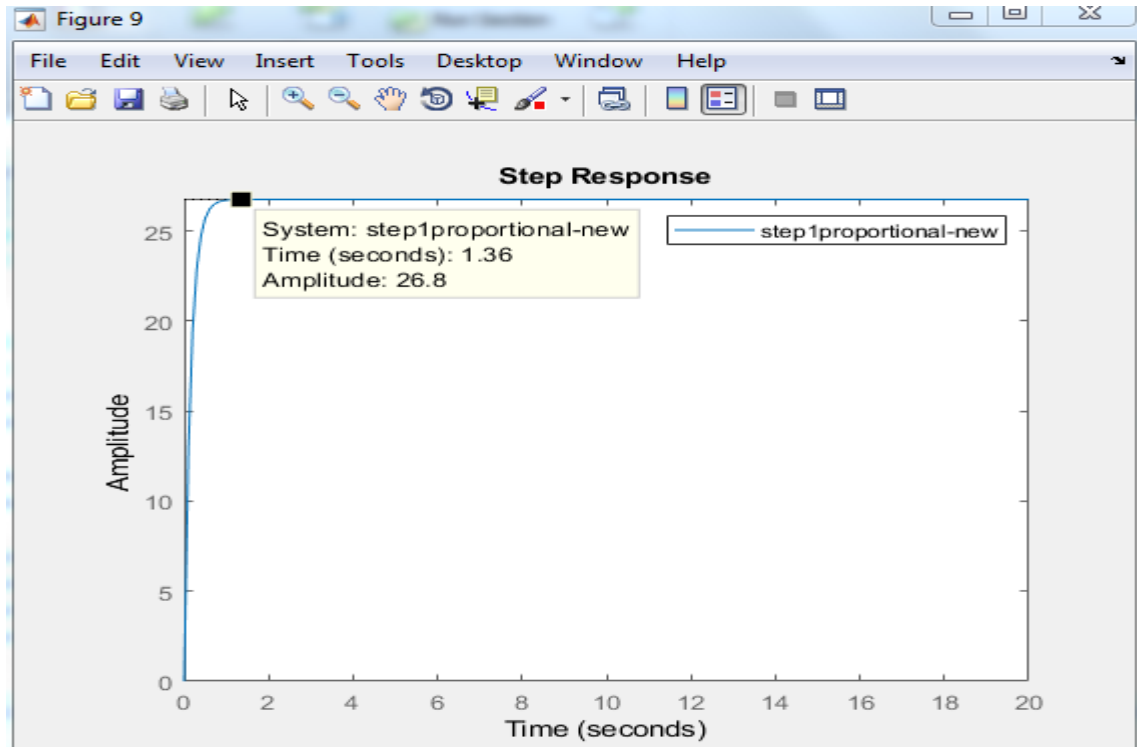
```
Kp_proportional_new1 = 10550; %%peak response at 5.6 secs and rise time 0.341 secs
Kp_proportional_new2 = 9650; %%peak response 26.8 m/s at 6.1 secs and rise time at 0.371 ssecs
```

```
C_proportional_new1 = pid(Kp_proportional_new1);
C_proportional_new2 = pid(Kp_proportional_new2);
```

```
T1_proportional_new = feedback(C_proportional_new1*P_transfer_cruise1,1)
T2_proportional_new = feedback(C_proportional_new2*P_transfer_cruise2,1)
step1_proportional_new = r*T1_proportional_new
step2_proportional_new = r*T2_proportional_new
```

```
figure(9)
step(step1_proportional_new,t)
legend("step1proportional-new")
axis([0 20 0 26.8]) %%we use the following axes for our desired reference velocity
```

```
figure(10)  
step(step2_proportional_new,t)  
legend("step2proportional-new")  
axis([0 20 0 26.8])
```



Το σφάλμα σταθερής κατάστασης είναι πλέον ουσιαστικά μηδέν, και ο χρόνος ανόδου έχει μειωθεί σημαντικά. Ωστόσο, αυτή η απόκριση δεν είναι ρεαλιστική, διότι ένα πραγματικό σύστημα ελέγχου ταχύτητας ταξιδιού(cruise control system) γενικά δεν μπορεί να αλλάξει την ταχύτητα του οχήματος από 0 έως 26,8 m / s σε λιγότερο από 2 δευτερόλεπτα, λόγω των περιορισμών ισχύος του κινητήρα και του συστήματος μετάδοσης. Οι περιορισμοί του ενεργοποιητή συναντώνται πολύ συχνά στην πράξη, στα συστήματα ελέγχου της μηχανικής, και ως εκ τούτου, η απαιτούμενη ενέργεια ελέγχου πρέπει πάντα να λαμβάνεται υπόψη όταν προτείνει ένα νέο ελεγκτή. Η λύση σε αυτό το πρόβλημα στην περίπτωση αυτή είναι να επιλέξουμε ένα χαμηλότερο αναλογικό κέρδους, K_p , που θα δώσει ένα λογικό χρόνο ανύψωσης(rise time), και να προσθέσουμε έναν ολοκληρωτικό ελεγκτή για την εξάλειψη του σφάλματος σταθερής κατάστασης.

4.4 Έλεγχος με χρήση PI

Η συνάρτησης μεταφοράς κλειστού βρόγχου του συστήματος ελέγχου ταχύτητας ταξιδιού με ελεγκτή PI ($C = K_p + K_i/s$) είναι:

$$T(s) = \frac{Y(s)}{R(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{K_p + K_i}{m * s^2 + (b + K_p) * s + K_i}$$

Μια πρόσθεση ενός ολοκληρωτικού ελεγκτή στο σύστημα εξαλείφει το σφάλμα σταθερής κατάστασης. Τη δεδομένη στιγμή, υποθέτουμε πως $K_{p_pi1} = 9484.9$, $K_{p_pi2} = 9520$ και K_{i_pi1} , K_{i_pi2} είναι ίσο με 1 και παίρνουμε τα ακόλουθα αποτελέσματα:

```
%%%%PI Control
```

```
Kp_pi1 = 9484.9;%rise time at 0.375 secs and 5.57 secs for 26.8 m/s
```

```
Ki_pi1 = 1;
```

```
C_pi1 = pid(Kp_pi1,Ki_pi1);
```

```
Kp_pi2 = 9520;%rise time at 0.377 secs and peak response at 20 secs and 26.8 m/s in 7.09 secs
```

```
Ki_pi2 = 1;
```

```
C_pi2 = pid(Kp_pi2,Ki_pi2);
```

```
T1_pi = feedback(C_pi1*P_transfer_cruise1,1)
```

```
T2_pi = feedback(C_pi2*P_transfer_cruise2,1)
```

```
step1_pi = r*T1_pi
```

```
step2_pi = r*T2_pi
```

```
t = 0:0.1:20;
```

```
%%step(r*T1,t)%%if we want to get independently step responses
```

```
%%step(r*T2,t)
```

```
%%otherwise we get at the same plot
```

```
figure(11)
```

```

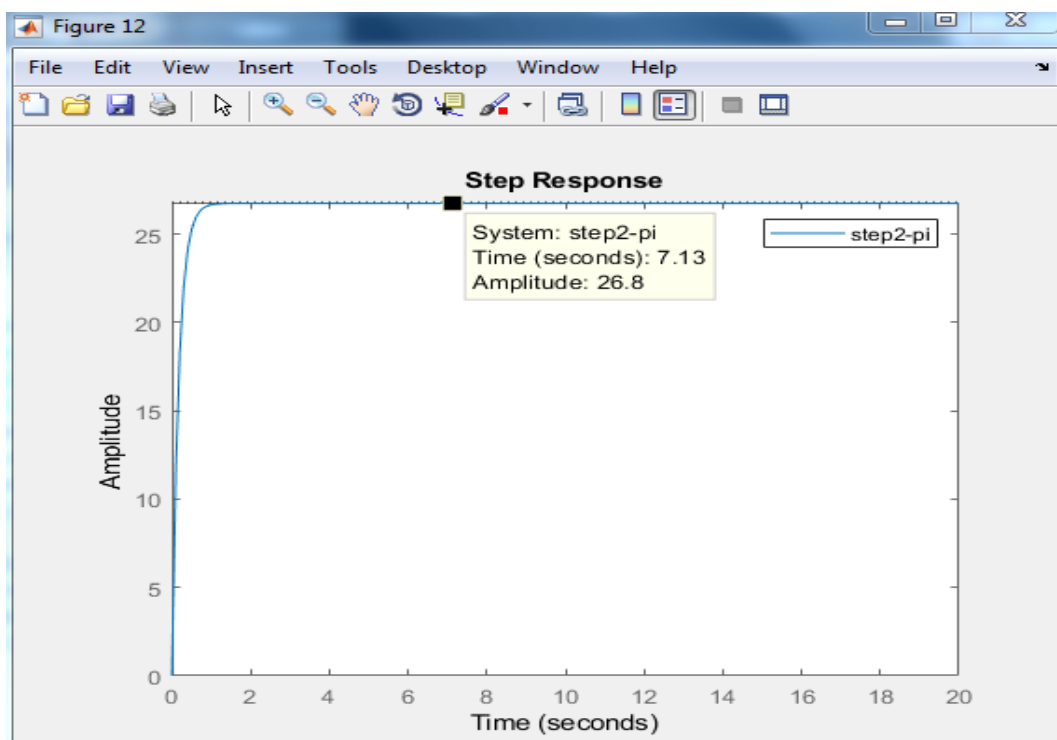
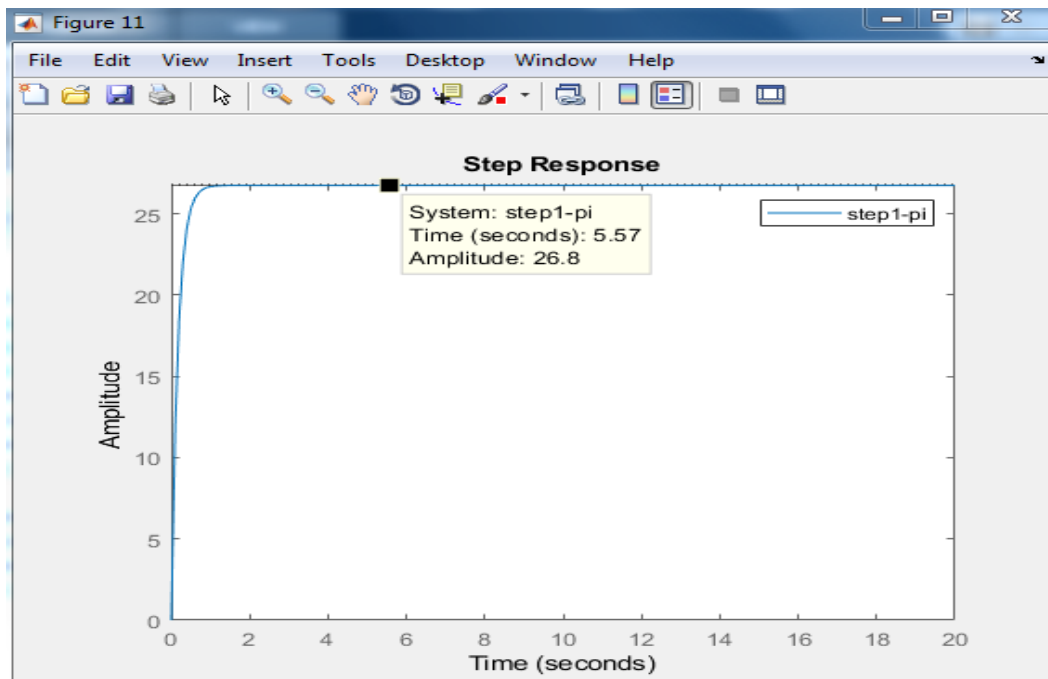
step(step1_pi,t)
legend("step1-pi")
axis([0 20 0 26.8])

```

```

figure(12)
step(step2_pi,t)
legend("step2-pi")
axis([0 20 0 26.8])

```



Τώρα προσαρμόζουμε το αναλογικό κέρδος, K_p , και το ολοκληρωτικό κέρδος, K_i , για να λάβουμε την επιθυμητή απόκριση. Όταν ρυθμίζουμε το ολοκληρωτικό κέρδος K_i , ξεκινάμε με μικρές τιμές καθώς ένα μεγάλο K_i μπορεί να αποσταθεροποιήσει την απόκριση. Όταν $K_{p_pi_new1}=9485, K_{p_pi_new2}=9521$, οι βηματικές αποκρίσεις θα είναι οι ακόλουθες:

```

%%changing Kp tp 700 and Ki to 7 to see the changes of response
Kp_pi_new1 = 9485;%rise time at 0.375 secs and peak response at 20 secs and
26.8 m/s in 5.47 secs(as we increase the value of Kp the speed 26.8m/s is
reached faster)
Ki_pi_new1 = 1;
C_pi_new1 = pid(Kp_pi_new1,Ki_pi_new1);

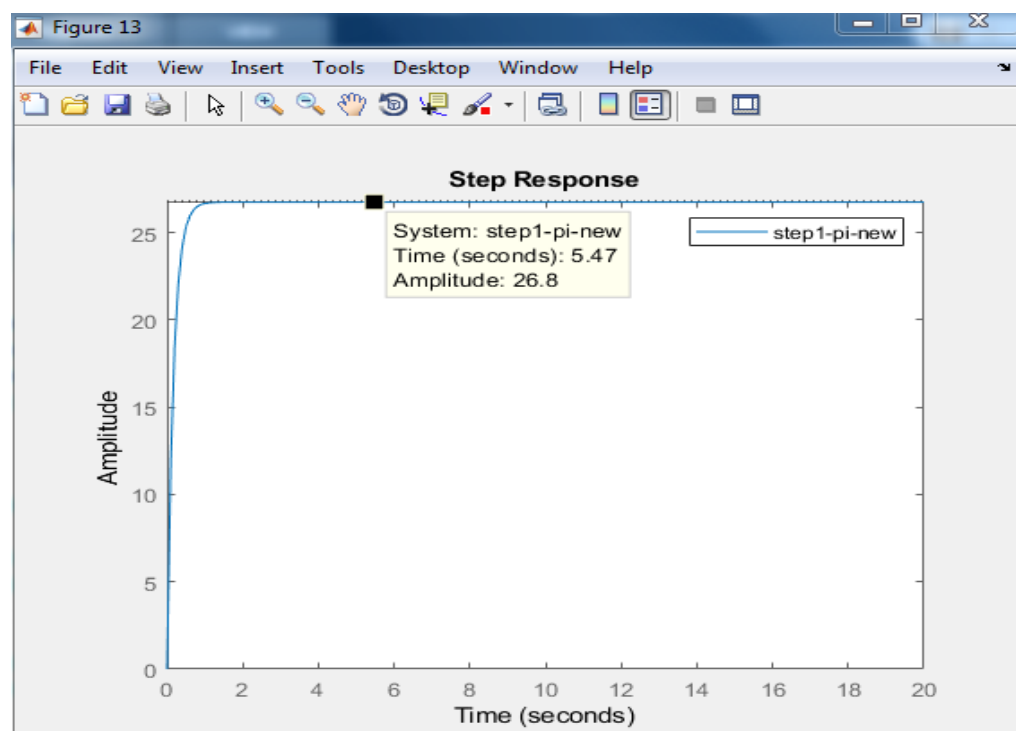
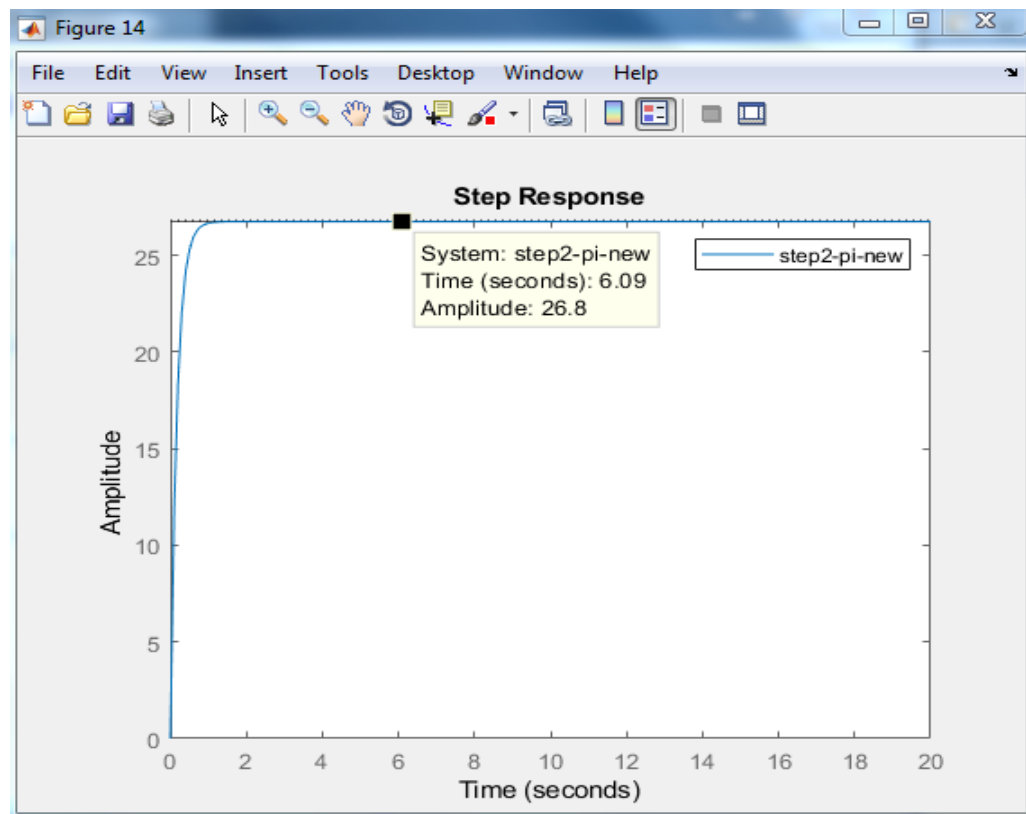
Kp_pi_new2 = 9521;%rise time at 0.377 secs and 26.8m /s in 6.09 secs
Ki_pi_new2 = 1;
C_pi_new2 = pid(Kp_pi_new2,Ki_pi_new2);

T1_pi_new = feedback(C_pi_new1*P_transfer_cruise1,1)
T2_pi_new = feedback(C_pi_new2*P_transfer_cruise2,1)
step1_pi_new = r*T1_pi_new
step2_pi_new = r*T2_pi_new

figure(13)
step(step1_pi_new,t)
legend("step1-pi-new")
axis([0 20 0 26.8])

figure(14)
step(step2_pi_new,t)
legend("step2-pi-new")
axis([0 20 0 26.8])

```



4.5 Έλεγχος με χρήση PID

Για αυτό το συγκεκριμένο παράδειγμα, η υλοποίηση ενός διαφορικού ελεγκτή δεν ήταν απαραίτητη για να ληφθεί η απαιτούμενη έξοδος. Η συνάρτηση μεταφοράς κλειστού

βρόχου για αυτό το σύστημα ελέγχου ταχύτητας ταξιδιού (cruise control system) με έναν ελεγκτή PID ($C = K_p + K_i / s + K_d * s$) είναι:

$$T(s) = \frac{Y(s)}{R(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{K_d * s^2 + K_p * s + K_i}{(m + K_d) * s^2 + (b + K_p) * s + K_i}$$

Ας υποθέσουμε πως $K_{p1} = 9484.9, K_{p2} = 9520, K_{i1} = K_{i2} = 1$ και $K_{d1} = K_{d2}$ που ισούται με 1 :

```

%%%%%%PID Control

C_pid1 = pid(Kp1,Ki1,Kd1);
C_pid2 = pid(Kp2,Ki2,Kd2);

T1_pid = feedback(C_pid1*P_transfer_cruise1,1)
T2_pid = feedback(C_pid2*P_transfer_cruise2,1)

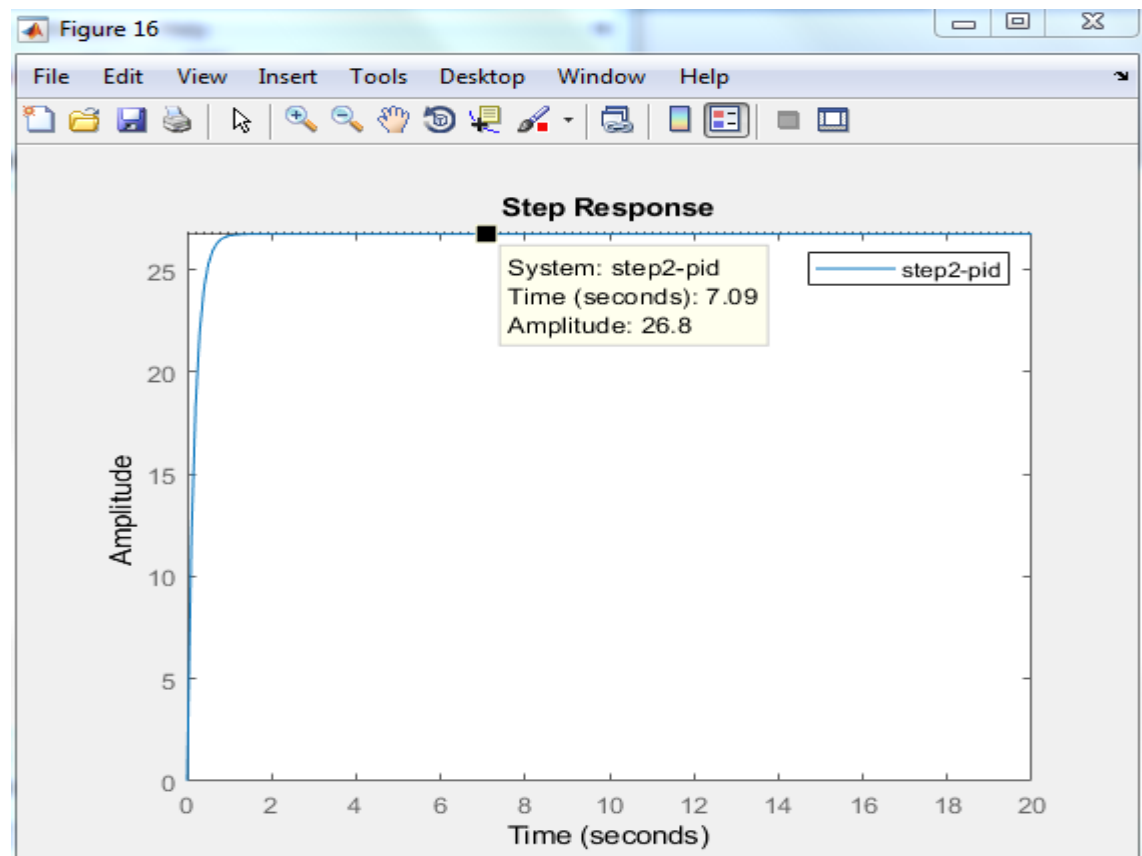
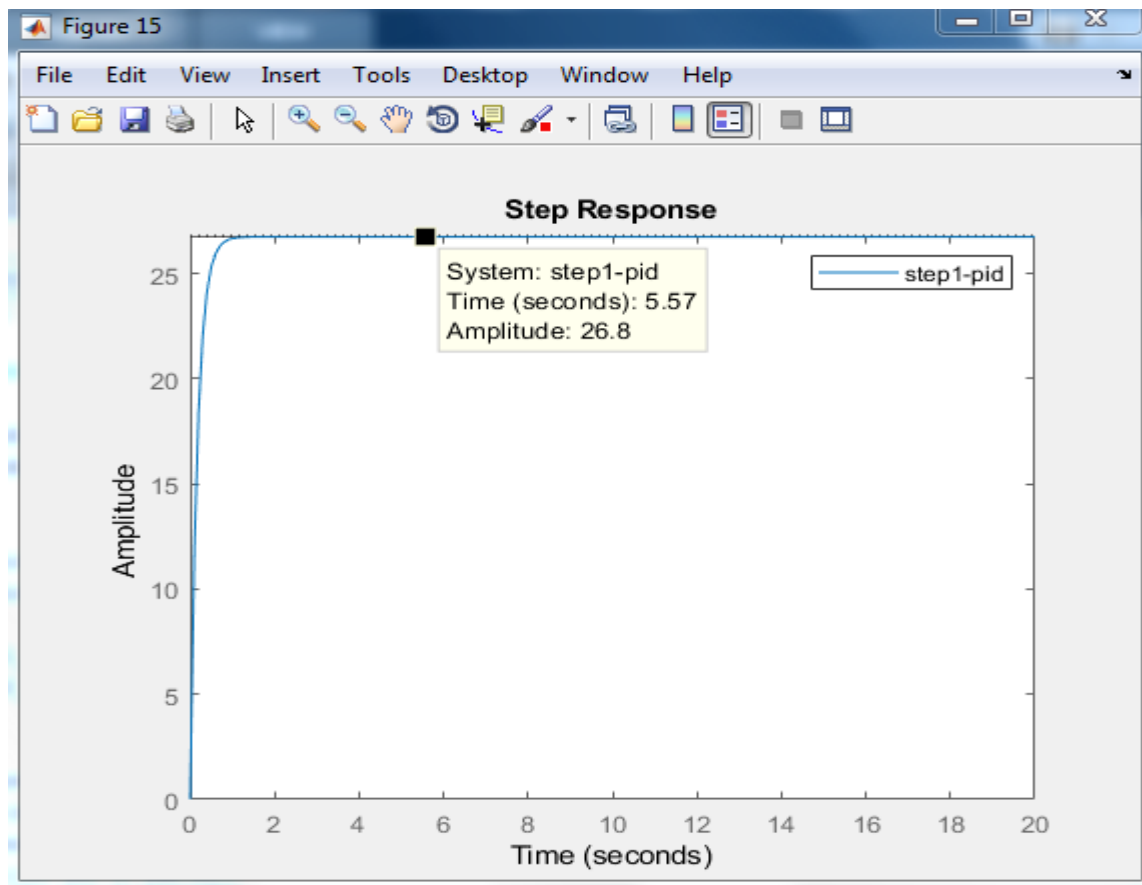
step1_pid = r*T1_pid
step2_pid = r*T2_pid

figure(15)
step(step1_pid,t)
legend("step1-pid")
axis([0 20 0 26.8])

figure(16)
step(step2_pid,t)
legend("step2-pid")
axis([0 20 0 26.8])

```

Μετά από δοκιμές προσαρμόσαμε τις τιμές των κερδών μέχρι να λάβουμε τα επιθυμητά αποτελέσματα, ούτως ώστε να λάβουμε τον επιθυμητό χρόνο ανύψωσης που είναι 7 δευτερόλεπτα για την πρώτη περίπτωση και 8 δευτερόλεπτα για την δεύτερη περίπτωση, λαμβάνοντας υπόψη ότι επιπλέον χρειαζόμαστε ένα ελάχιστο σφάλμα σταθερής κατάστασης. Οι βηματικές αποκρίσεις που λαμβάνουμε είναι οι ακόλουθες:



4.6 Σύγκριση του PI και PID

Στον Πίνακα 2 φαίνονται τα αποτελέσματα της σύγκρισης των δύο ελεγκτών PI και PID των δύο μοντέλων μας με βάση την χειροκίνητη ρύθμιση των τιμών που βασίστηκε στην παρατήρηση των αποτελεσμάτων.

Πίνακας 2.Αποτελέσματα σύγκρισης για ελεγκτές PI και PID		
PI		PID
Kp	Kp1= 9582, Kp2= 9620	Kp1= 9484.9, Kp2= 9520
Ki	Ki1= 1, Ki2= 1	Ki1= 1, Ki2= 1
Kd	-	Kd1= 1, Kd2= 1
Χρόνος ανύψωσης	i) 0.375 δευτερόλεπτα για το πρώτο μοντέλο ii) 0.377 δευτερόλεπτα για το δεύτερο μοντέλο	i) 0.375 δευτερόλεπτα για το πρώτο μοντέλο ii) 0.378 δευτερόλεπτα για το δεύτερο μοντέλο
Ρυθμός αλλαγής ταχύτητας	i) 26.8 m/s σε 5.57 δευτερόλεπτα για το πρώτο μοντέλο ii) 26.8 m/s σε 7.09 δευτερόλεπτα για το δεύτερο μοντέλο	i) 26.8 m/s σε 5.57 δευτερόλεπτα για το πρώτο μοντέλο ii) 26.8 m/s σε 7.09 δευτερόλεπτα για το δεύτερο μοντέλο
Kp_new	Kp1= 9485, Kp2= 9521	Kp1_new = 9484.9, Kp2_new= 9520 (παραμένουν ίδια)
Ki_new	Ki1= 1, Ki2= 1	Ki1= 1, Ki2= 1 (παραμένουν ίδια)
Kd_new	-	Kd1= 1, Kd2= 1 (παραμένουν ίδια)
Νέος χρόνος ανύψωσης	i) 0.375 δευτερόλεπτα για το πρώτο μοντέλο ii) 0.377 δευτερόλεπτα για το δεύτερο μοντέλο	i) 0.375 δευτερόλεπτα για το πρώτο μοντέλο ii) 0.378 δευτερόλεπτα για το δεύτερο μοντέλο (παραμένουν ίδια)
Νέος ρυθμός αλλαγής ταχύτητας	i) 26.8 m/s in 5.47 δευτερόλεπτα για το πρώτο μοντέλο ii) 26.8 m/s in 6.09 δευτερόλεπτα για το δεύτερο μοντέλο	i) 26.8 m/s in 5.57 δευτερόλεπτα για το πρώτο μοντέλο ii) 26.8 m/s in 7.09 δευτερόλεπτα για το δεύτερο μοντέλο (παραμένουν ίδια)

Τα συμπεράσματα μας για την χειροκίνητη ρύθμιση των κερδών του κάθε ελεγκτή και με βάση τα οποία ορίσαμε τις τιμές των κερδών είναι πως:

- Αυξάνοντας την τιμή του κέρδους K_p μειώνεται το σφάλμα σταθερής κατάστασης.
- Μετά από ένα συγκεκριμένο σημείο, η αύξηση του K_p προκαλεί μόνο την δημιουργία υπερύψωσης.
- Αυξάνοντας την τιμή του κέρδους K_p μειώνεται ο χρόνος υπερύψωσης [7].
- Η εισαγωγή ολοκληρωτικού ελέγχου(I) εξαλείφει το σφάλμα σταθερής κατάστασης.
- Μετά από ένα συγκεκριμένο σημείο, η αύξηση του K_i προκαλεί μόνο την αύξηση του ποσοστού υπερύψωσης.
- Αυξάνοντας την τιμή του κέρδους K_d μειώνεται η υπερύψωση.
- Αυξάνοντας την τιμή του κέρδους K_d μειώνεται ο χρόνος αποκατάστασης.

Παρακάτω φαίνεται ο κώδικας που χρησιμοποιήθηκε στο MATLAB για τον αυτόματο υπολογισμό των ελεγκτών P,PI και PID. Οι γραφικές παραστάσεις φαίνονται επίσης παρακάτω και τα αποτελέσματα αποτυπώνονται στον Πίνακα 3 [8,9]. Για την χρήση του controller PIDF,σε συγκεκριμένο crossover frequency η συνάρτηση ελέγχου που χρησιμοποιείται είναι η ακόλουθη [4] :

$$C = K_p + K_i * \frac{1}{s} + K_d * \frac{s}{(T_f * s + 1)}$$

```
%this method is used for auto computation of gains
[C_prop1,info] = pidtune(P_transfer_cruise1,'P')
[C_prop2,info] = pidtune(P_transfer_cruise2,'P')

[C_PI1,info] = pidtune(P_transfer_cruise1,'PI')
[C_PI2,info] = pidtune(P_transfer_cruise2,'PI')
%%we increase the crossover frequency for results optimization
[C_PI1_fast,info] = pidtune(P_transfer_cruise1,'PI',1.0)
[C_PI2_fast,info] = pidtune(P_transfer_cruise2,'PI',1.0)

[C_PID1,info] = pidtune(P_transfer_cruise1,'PID')
[C_PID2,info] = pidtune(P_transfer_cruise2,'PID')

[C_PID1_fast,info] = pidtune(P_transfer_cruise1,'PID',1.0)
[C_PID2_fast,info] = pidtune(P_transfer_cruise2,'PID',1.0)
```

```
T_PI1_pre = feedback(C_PI1*P_transfer_cruise1, 1);
```

```

T_PI1 =r*T_PI1_pre
T_PI2_pre = feedback(C_PI2*P_transfer_cruise2, 1);
T_PI2 =r*T_PI2_pre

T_PI1_fast_pre = feedback(C_PI1_fast*P_transfer_cruise1,1);
T_PI2_fast_pre = feedback(C_PI2_fast*P_transfer_cruise2,1);
T_PI1_fast= r*T_PI1_fast_pre
T_PI2_fast= r*T_PI2_fast_pre

T_PID1_pre = feedback(C_PID1*P_transfer_cruise1,1);
T_PID2_pre = feedback(C_PID2*P_transfer_cruise2,1);
T_PID1= r*T_PID1_pre
T_PID2= r*T_PID2_pre

T_PID1_fast_pre = feedback(C_PID1_fast*P_transfer_cruise1,1);
T_PID2_fast_pre = feedback(C_PID2_fast*P_transfer_cruise2,1);
T_PID1_fast= r*T_PID1_fast_pre
T_PID2_fast= r*T_PID2_fast_pre

R=26.8;
[y,t]=step(R*T_PI1_pre);
sserror_pi1=abs(R-y(end))

[y,t]=step(R*T_PI2_pre);
sserror_pi2=abs(R-y(end))

[y,t]=step(R*T_PI1_fast_pre);
sserror_pi1_fast=abs(R-y(end))

[y,t]=step(R*T_PI2_fast_pre);
sserror_pi2_fast=abs(R-y(end))

[y,t]=step(R*T_PID1_pre);
sserror_PID1=abs(R-y(end))

[y,t]=step(R*T_PID2_pre);
sserror_PID2=abs(R-y(end))

[y,t]=step(R*T_PID1_fast_pre);
sserror_PID1_fast=abs(R-y(end))

[y,t]=step(R*T_PID2_fast_pre);
sserror_PID2_fast=abs(R-y(end))

figure(17)
step(T_PI1,T_PI1_fast)
legend('PI1','PI1,fast')

figure(18)
step(T_PI2,T_PI2_fast)

```

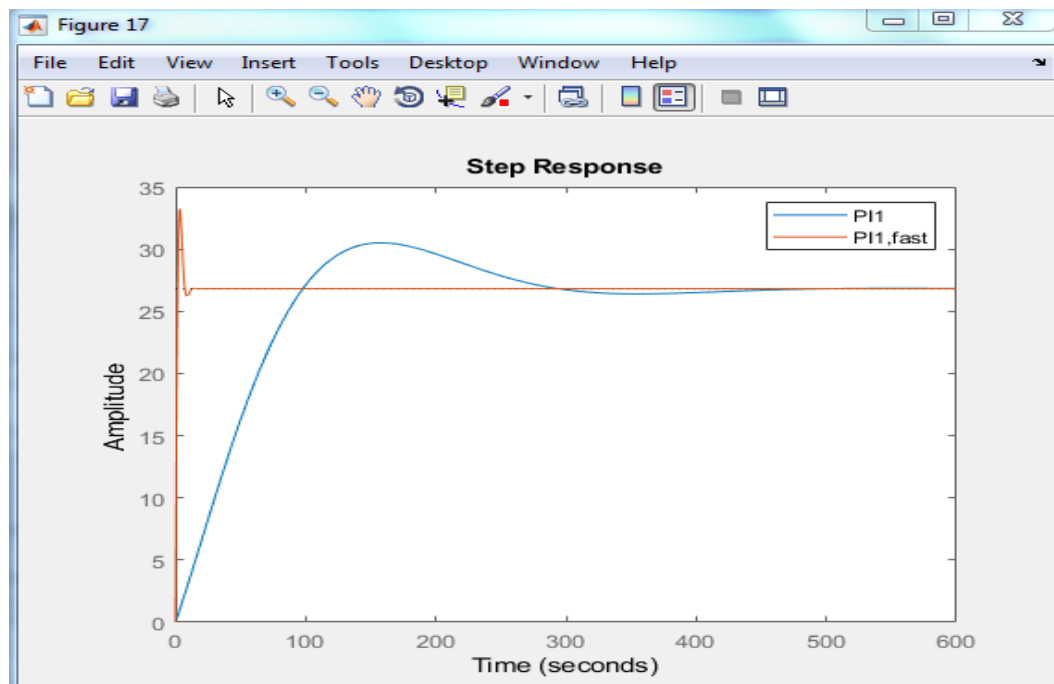
```
legend('PI2','PI2,fast')
```

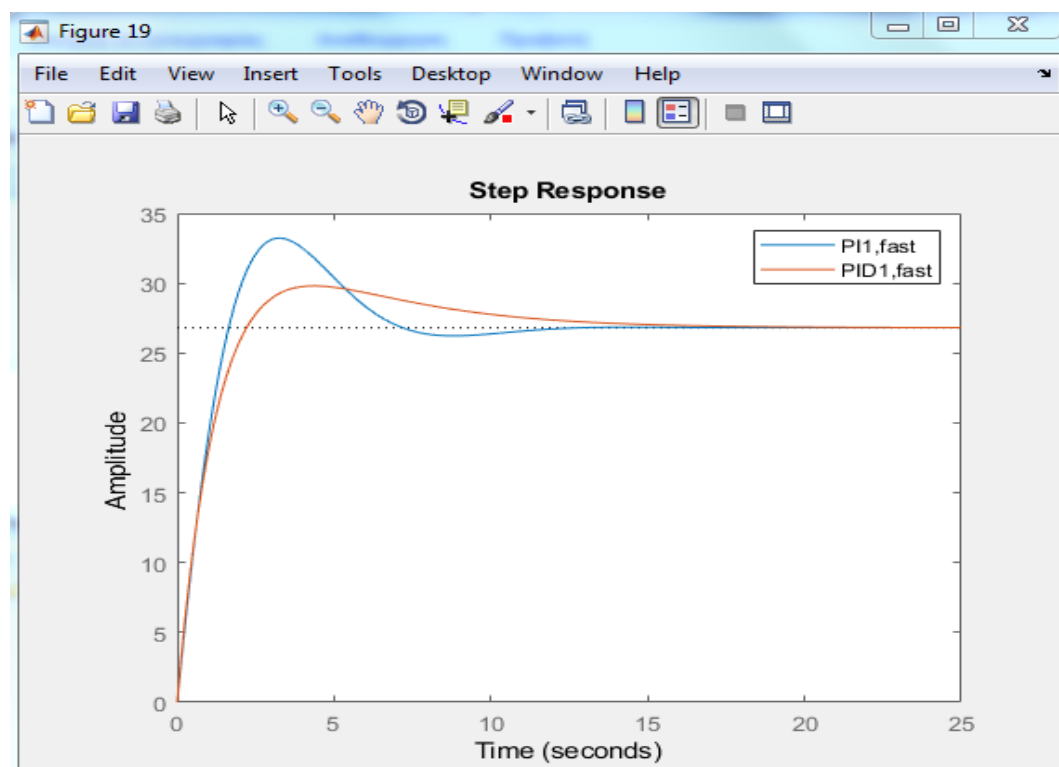
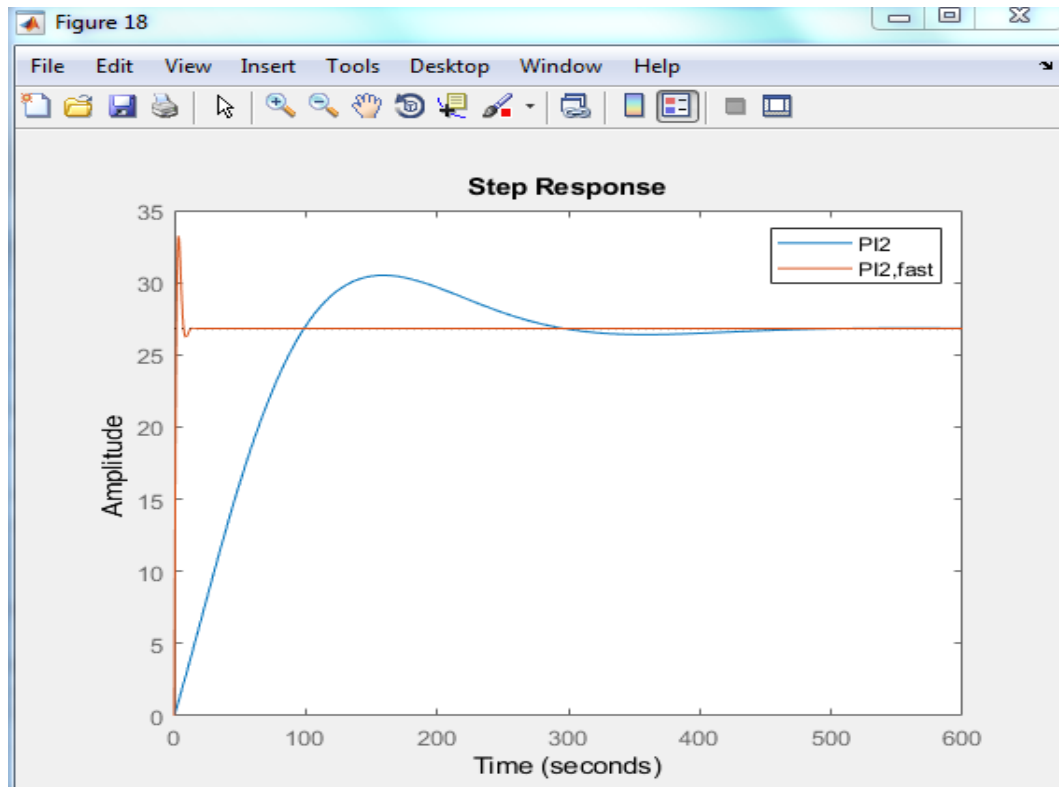
```
figure(19)
step(T_PI1_fast, T_PID1_fast);
legend('PI1,fast','PID1,fast');
```

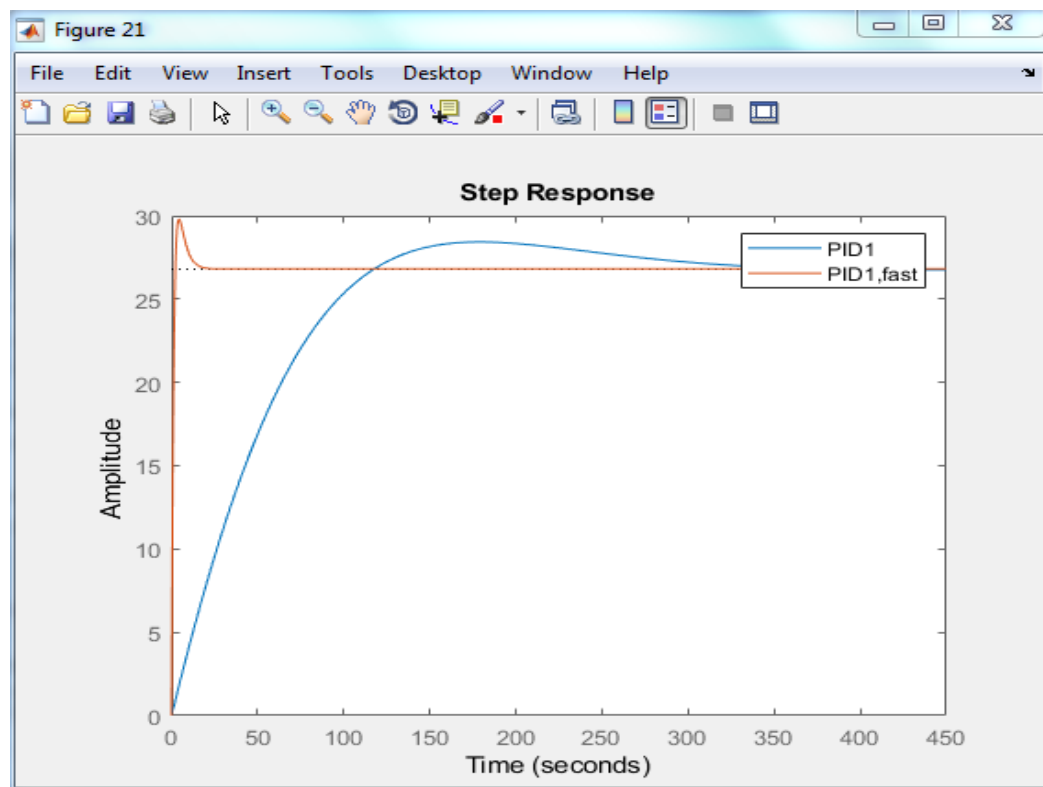
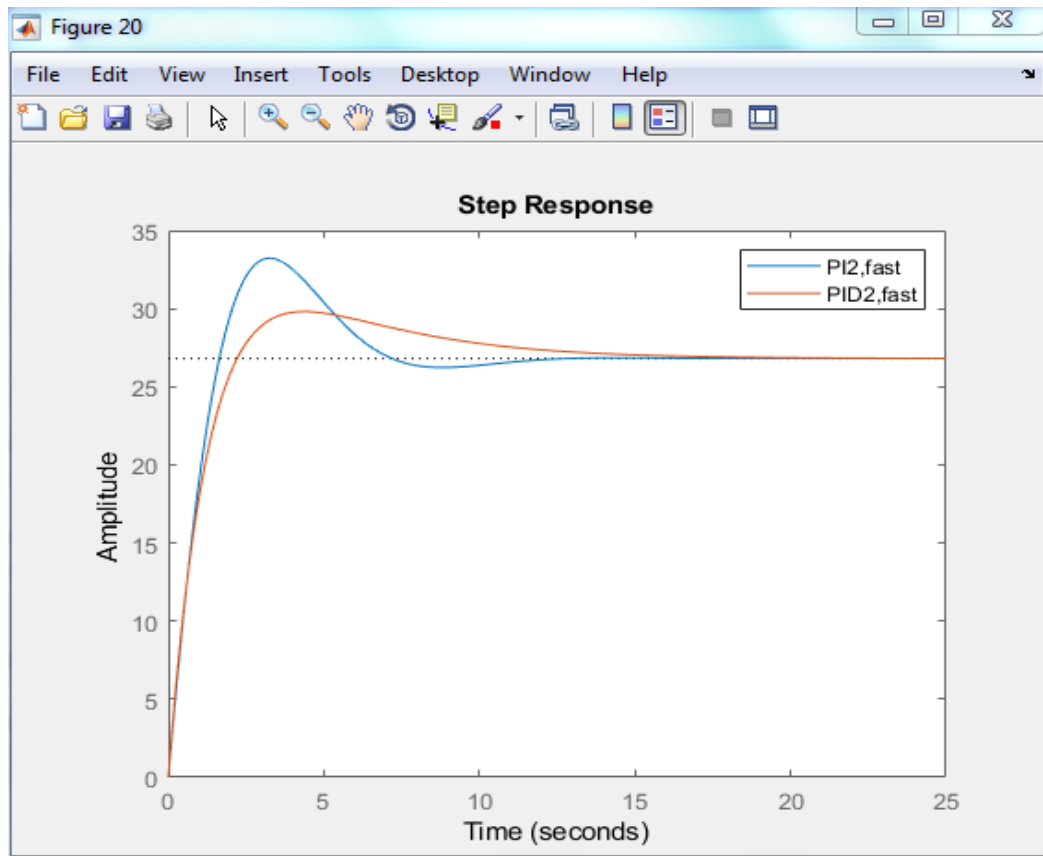
```
figure(20)
step(T_PI2_fast, T_PID2_fast);
legend('PI2,fast','PID2,fast');
```

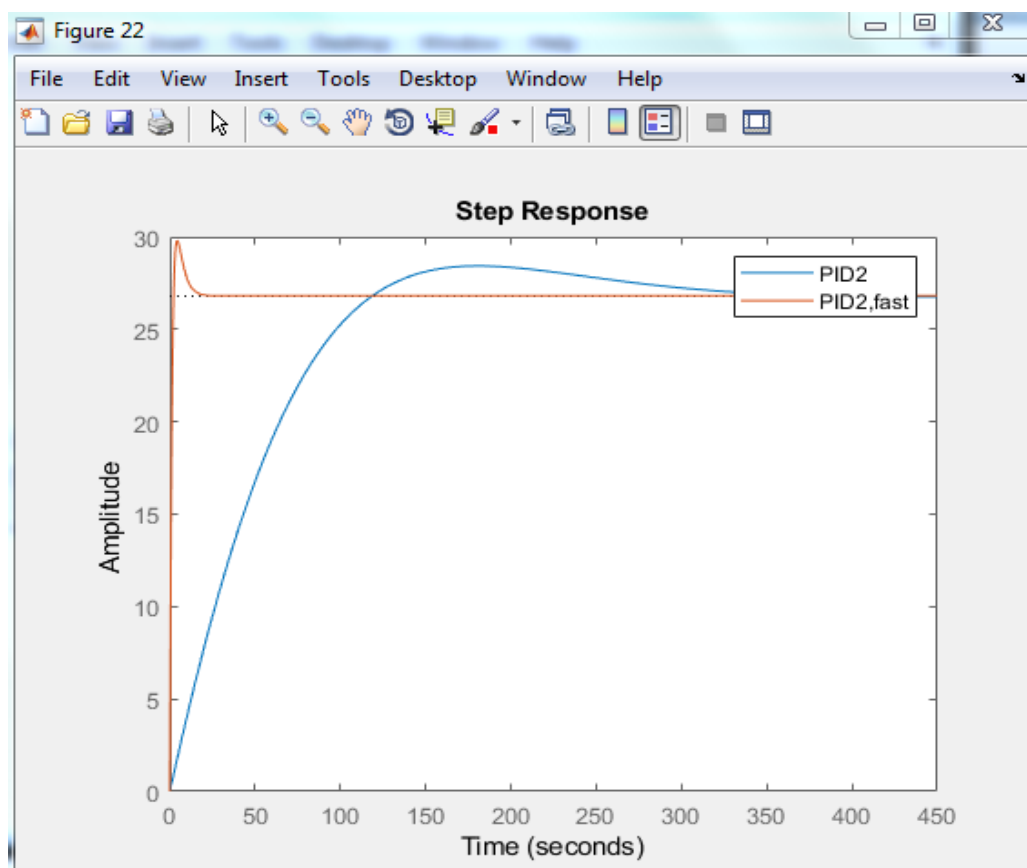
```
figure(21)
step(T_PID1, T_PID1_fast);
legend('PID1','PID1,fast')
```

```
figure(22)
step(T_PID2, T_PID2_fast);
legend('PID2','PID2,fast')
```









Πίνακας 3.Αποτελέσματα σύγκρισης για ελεγκτές PI και PID με χρήση της συνάρτησης pidtune του Matlab		
	PI	PID
K_p	$K_{p1}= 18.7, K_{p2}= 18.7$	$K_{p1}= 25.6, K_{p2}= 25.7$
K_i	$K_{i1}= 0.618, K_{i2}= 0.615$	$K_{i1}= 0.513, K_{i2}= 0.51$
K_d	-	$K_{d1}= 0, K_{d2}= 0$ (για PID) $K_{d1}= -78.9, K_{d2}= -79.9$ (για $T_f=1$ και PIDF)
Συχνότητα διασταύρωσης(crossover frequency)	i) 0.0196 rad/s για το πρώτο μοντέλο ii) 0.0195 rad/s για το δεύτερο μοντέλο	i) 0.0196 rad/s για το πρώτο μοντέλο ii) 0.0195 rad/s για το δεύτερο μοντέλο
Χρόνος ανύψωσης	i) 73.4 secs για το πρώτο μοντέλο ii) 74 secs για το δεύτερο μοντέλο	i) 78.5 secs για το πρώτο μοντέλο (PIDF) ii) 79.2 secs για το δεύτερο μοντέλο(PIDF) i) 82.7 δευτερόλεπτα για το πρώτο μοντέλο (PID) ii) 83.5 δευτερόλεπτα για το δεύτερο μοντέλο (PID)
Ρυθμός αλλαγής	i) 26.8 m/s σε 97.5	i) 26.8 m/s σε 109 δευτερόλεπτα για

ταχύτητας	<p>δευτερόλεπτα για το πρώτο μοντέλο (PIDF και PID)</p> <p>i) 26.8 m/s σε 99 δευτερόλεπτα για το δεύτερο μοντέλο (PIDF)</p> <p>ii) 26.8 m/s σε 98.4 δευτερόλεπτα για το δεύτερο μοντέλο (PID)</p>	<p>το πρώτο μοντέλο (PIDF)</p> <p>ii) 26.8 m/s σε 110 δευτερόλεπτα για το δεύτερο μοντέλο (PIDF)</p> <p>i) 26.8 m/s σε 118 δευτερόλεπτα για το πρώτο μοντέλο (PID)</p> <p>ii) 26.8 m/s σε 118 δευτερόλεπτα για το δεύτερο μοντέλο (PID)</p>
Kp_new (για συχνότητα διασταύρωσης 1.0 rad/sec)	$K_{p1} = 1.4 e^3$, $K_{p2} = 1.42 e^3$	<p>$K_{p1_new} = 1.76 e^3$, $K_{p2_new} = 1.79 e^3$</p> <p>(με $T_f = 1$ και τα δύο μοντέλα, PIDF)</p> <p>$K_{p1_new} = 1.59 e^3$, $K_{p2_new} = 1.61 e^3$</p> <p>(PID)</p>
Ki_new (για συχνότητα διασταύρωσης 1.0 rad/sec)	$K_{i1} = 828$, $K_{i2} = 838$	<p>$K_{i1} = 344$, $K_{i2} = 349$ (PIDF)</p> <p>$K_{i1} = 328$, $K_{i2} = 332$ (για PID)</p>
Kd_new (για συχνότητα διασταύρωσης 1.0 rad/sec)	-	<p>$K_{d1} = -509$, $K_{d2} = -515$ (PIDF)</p> <p>$K_{d1} = 0$, $K_{d2} = 0$ (PID)</p>
Σφάλμα σταθερής κατάστασης (new steady state error)	<p>i) 0.0425 για το πρώτο μοντέλο</p> <p>ii) 0.0425 για το δεύτερο μοντέλο</p>	<p>i) 0.0419 για το πρώτο μοντέλο (PID)</p> <p>ii) 0.0419 για το δεύτερο μοντέλο (PID)</p>
Χρόνος ανύψωσης (για συχνότητα διασταύρωσης 1.0 rad/sec)	<p>i) 1.26 δευτερόλεπτα για το πρώτο μοντέλο</p> <p>ii) 1.26 δευτερόλεπτα για το δεύτερο μοντέλο</p>	<p>i) 1.45 δευτερόλεπτα για το πρώτο μοντέλο (PIDF)</p> <p>ii) 1.45 δευτερόλεπτα για το δεύτερο μοντέλο (PIDF)</p> <p>(παραμένουν ίδια)</p> <p>i) 1.58 δευτερόλεπτα για το πρώτο μοντέλο (PID)</p> <p>ii) 1.58 δευτερόλεπτα για το δεύτερο μοντέλο (PID)</p>
Ρυθμός αλλαγής ταχύτητας (για συχνότητα διασταύρωσης 1.0 rad/sec)	<p>i) 26.8 m/s σε 1.64 δευτερόλεπτα για το πρώτο μοντέλο</p> <p>ii) 26.8 m/s σε 1.64 δευτερόλεπτα για το δεύτερο μοντέλο</p>	<p>i) 26.8 m/s σε 1.95 δευτερόλεπτα για το πρώτο μοντέλο (PIDF)</p> <p>ii) 26.8 m/s σε 1.95 δευτερόλεπτα για το δεύτερο μοντέλο (PIDF)</p> <p>(παραμένουν ίδια)</p> <p>i) 26.8 m/s σε 2.21 δευτερόλεπτα για το πρώτο μοντέλο (PID)</p> <p>ii) 26.8 m/s σε 2.21 δευτερόλεπτα για το δεύτερο μοντέλο (PID)</p>

Νέο σφάλμα σταθερής κατάστασης (new steady state error)	i) 0.0317 για το πρώτο μοντέλο ii) 0.0317 για το δεύτερο μοντέλο (για pid)	i) 0.0417 για το πρώτο μοντέλο (PID) ii) 0.0418 για το δεύτερο μοντέλο (PID)

Παρατηρήσεις

Με βάση τα αποτελέσματα του Πίνακα 2 διαπιστώνουμε πως η χρήση του ελεγκτή PI οδηγεί σε βελτιωμένα αποτελέσματα σε σύγκριση με τον ελεγκτή PIDF ή PID. Αυτό συμβαίνει διότι με την χρήση αυτού του ελεγκτή σε crossover frequency 1 rad/sec έχουμε μικρότερο χρόνο ανύψωσης 1.26 δευτερόλεπτα σε σχέση με τον αντίστοιχο χρόνο του PID που ισούται με 1.45 δευτερόλεπτα. Αντίστοιχα, ο χρόνος που φτάνουμε την επιθυμητή ταχύτητα είναι και αυτός μικρότερος με την χρήση του PI ελεγκτή και πιο συγκεκριμένα 1.64 δευτερόλεπτα, σε αντίθεση με εκείνο του PID που ισούται με 2.21 δευτερόλεπτα. Με τον ελεγκτή PID πετυχαίνουμε ένα ποσοστό υπερύψωσης 11.2% που είναι μικρότερο από εκείνο με την χρήση του PI που ισούται με 24%. Ωστόσο, το σφάλμα σταθερής κατάστασης είναι μικρότερο με την χρήση του PI ελεγκτή καθώς ισούται με 0.0317 σε αντίθεση με τον PID που πετυχαίνει ένα σφάλμα 0.0418.

5. Μοντελοποίηση Συστήματος και Σχεδιασμός Ελεγκτή σε Simulink

5.1 Φυσική δομή, εξισώσεις συστήματος και κατασκευή μοντέλου

Η φυσική δομή του συστήματος ελέγχου ταχύτητας ταξιδιού μας είναι αυτή που αναφέρθηκε και στην Ενότητα 2.1 στο Σχήμα 1. Χρησιμοποιώντας το 2ο Νόμο του Newton παίρνουμε την εξίσωση (1) της Ενότητας 2.2. Οι παραμέτρους που βρέθηκαν και χρησιμοποιήθηκαν και στις προηγούμενες ενότητες είναι αυτοί εκείνοι που χρησιμοποιήθηκαν και στην Ενότητα 2.3 .

Το σύστημα θα μοντελοποιηθεί αθροίζοντας τις δυνάμεις που επιδρούν στη μάζα και ολοκληρώνοντας την επιτάχυνση για να λάβουμε την ταχύτητα.

Αρχικά, θα μοντελοποιήσουμε το ολοκλήρωμα της επιτάχυνσης χρησιμοποιώντας Simulink [4]:

$$\int \frac{dv}{dt} dt = v$$

- Εισάγουμε ένα ολοκληρωτικό κουτί (από την βιβλιοθήκη Continuous του Simulink) και σχεδιάζουμε γραμμές από και προς την είσοδο και έξοδο του τερματικού.
- Βάζουμε ως όνομα της γραμμής εισόδου "vdot" και ως όνομα της γραμμής εξόδου "v".

Από την στιγμή που η επιτάχυνση (dv/dt) ισούται με το άθροισμα των δυνάμεων διαιρούμενο δια την μάζα, θα διαιρέσουμε το εισερχόμενο σήμα με την μάζα.

- Εισάγουμε ένα τετράγωνο κέρδους (Gain block από την βιβλιοθήκη Math Operators) που συνδέεται με το ολοκληρωτικό τετράγωνο της γραμμής εισόδου και σχεδιάζουμε μια γραμμή που οδηγεί στην είσοδο του τετραγώνου κέρδους.
- Επεξεργαζόμαστε την τιμή του τετραγώνου κέρδους και αλλάζουμε την τιμή του σε $1/m$ για κάθε περίπτωση μοντέλου.

Στη συνέχεια:

- Επισυνάπτουμε ένα τετράγωνο αθροίσματος (Sum block από την βιβλιοθήκη Math Operations) στη γραμμή που προηγείται του τετραγώνου κέρδους inertia.

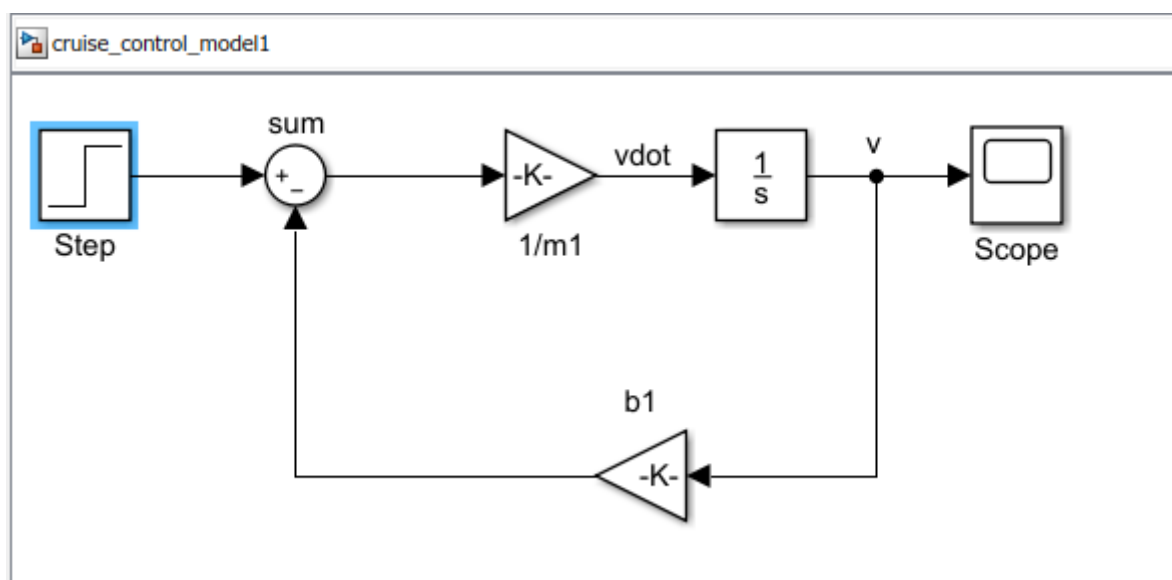
- Αλλάζουμε τα σύμβολα του τετραγώνου αθροίσματος σε "+-"
- Εισάγουμε ένα τετράγωνο κέρδους κάτω από το τετράγωνο που παίρνει την τιμή $1/m$.
- Ορίζουμε την τιμή του τετραγώνου σε "b" και μετονομάζουμε αυτό το τετράγωνο σε "b1" και "b2" αντίστοιχα.
- Τραβάμε μια γραμμή από την έξοδο του ολοκληρωτικού τετραγώνου και την συνδέουμε στην είσοδο του block κέρδους απόσβεσης (damping Gain block)
- Σχεδιάζουμε μια γραμμή από την έξοδο του τετραγώνου κέρδους απόσβεσης στην αρνητική είσοδο του τετραγώνου αθροίσματος (Sum block)

Η δεύτερη δύναμη που επιδρά στην μάζα είναι η είσοδος ελέγχου, f . Θα εφαρμόσουμε μια βηματική είσοδο.

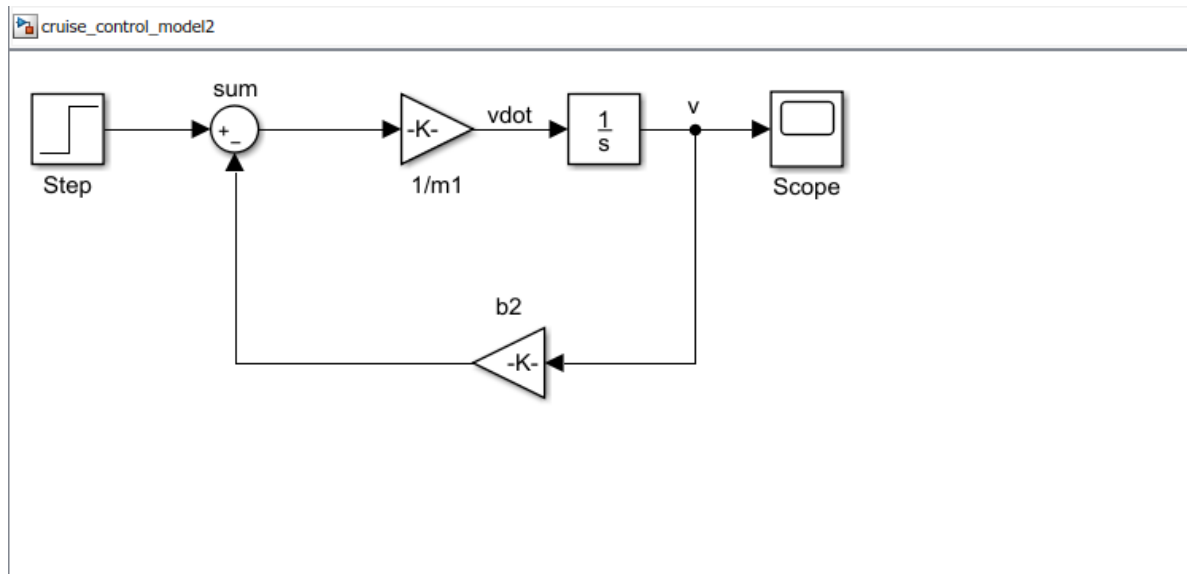
- Εισάγουμε ένα τετράγωνο Step (από την βιβλιοθήκη Sources) και το συνδέουμε με μια γραμμή στη θετική είσοδο του τετραγώνου αθροίσματος
- Για να δούμε την ταχύτητα εξόδου, εισάγουμε ένα τετράγωνο Scope (από την βιβλιοθήκη Sinks) συνδεδεμένο με την έξοδο του Ολοκληρωτή.
- Για να έχουμε την επιθυμητή βηματική είσοδο των 480 N να ισούται με 0, στις ρυθμίσεις παραμέτρων στο block ορίζουμε την τιμή Step σε 0 και την τελική τιμή σε f .

Το αποτέλεσμα της παραπάνω διαδικασίας για τα δύο μοντέλα μας φαίνεται παρακάτω.

Για το πρώτο μοντέλο



Για το δεύτερο μοντέλο



5.2 Σχεδίαση P, PI, PID ελεγκτή

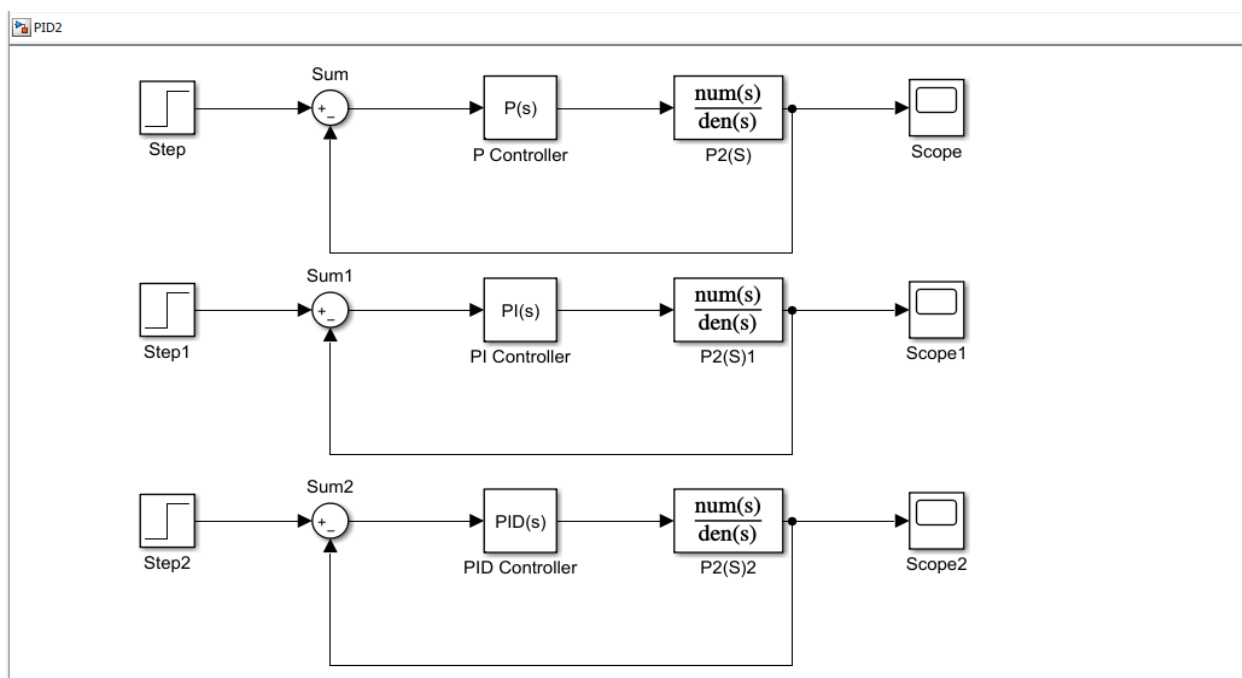
Υπάρχουν δυο διαφορετικοί τρόποι υλοποίησης του PID ελεγκτή. Ο πρώτος τον οποίο και θα υλοποιήσουμε παρακάτω χρησιμοποιεί ένα block PID που περιλαμβάνεται στην βιβλιοθήκη αναζήτησης του Simulink, ενώ ο δεύτερος υλοποιείται χωρίς την χρήση αυτού του block, αλλά με βάση τις εξισώσεις που προκύπτουν για τον ελεγκτή και την βοήθεια διαφόρων που συνιστούν αυτές τις εξισώσεις [4].

1. Το πρώτο μας βήμα αφού ανοίξουμε ένα blank μοντέλο για σχεδίαση στο Simulink, είναι η επιλογή του τμήματος Continuous από τη βιβλιοθήκη αναζήτησης (Simulink Library Browser).
2. Στη συνέχεια από το τμήμα αυτό επιλέγουμε το block PID, το οποίο θα χρησιμοποιηθεί και ως PID ελεγκτής.
3. Επόμενο μας βήμα είναι η μετάβαση στο τμήμα Sources, όπου και επιλέγουμε το block βηματικής απόκρισης (step response) για να εφαρμόσουμε στο PID.
4. Μετέπειτα, επιλέγουμε το τμήμα Sinks από το Simulink Library Browser, και από αυτό επιλέγουμε για εισαγωγή το block Scope, που χρησιμοποιείται για αναπαράσταση της εξόδου.
5. Επιπλέον, χρειαζόμαστε ένα σύστημα, δηλαδή την συνάρτηση μεταφοράς του συστήματος για να εφαρμόσουμε τον PID ελεγκτή. Από το τμήμα Continuous και πάλι επιλέγουμε το block Transfer Fcn.

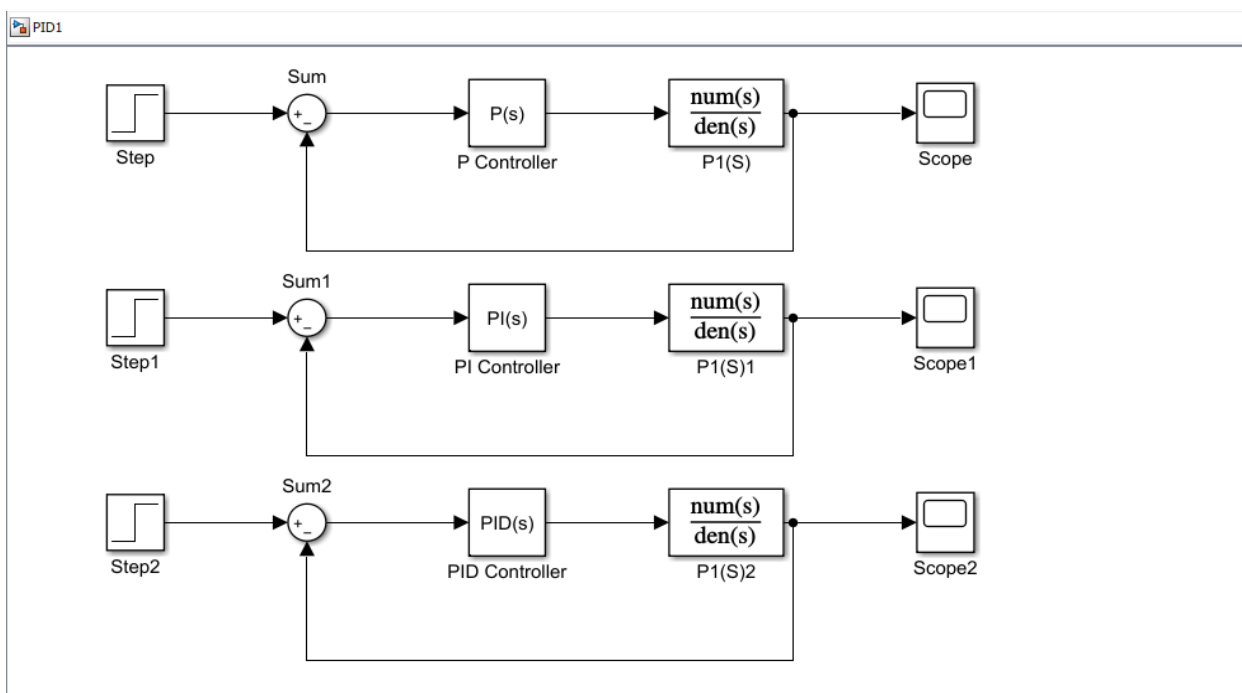
6. Τελευταίο μας βήμα είναι η τοποθέτηση του block Sum από το τμήμα Commonly Used Blocks, το οποίο χρησιμοποιείται για να αφαιρέσει το μονοπάτι ανατροφοδότησης του συστήματος κλειστού βρόγχου.

Να σημειωθεί πως ο έλεγχος που χρησιμοποιούμε σε κάθε περίπτωση, δηλαδή αναλογικός (P) ,(PI) ή PID ρυθμίζεται στο PID block. Το ίδιο ισχύει και για την ρύθμιση των χαρακτηριστικών που επιθυμούμε και στα υπόλοιπα blocks. Τα αποτελέσματα των παραπάνω βημάτων σχεδίασης για το μοντέλο συστήματος μας φαίνονται παρακάτω:

Για το πρώτο μοντέλο



Για το δεύτερο μοντέλο



Για να λάβουμε τα αποτελέσματα αποκρίσεων κλειστού βρόγχου, θα πρέπει να ρυθμίσουμε τις τιμές κέρδους κάθε ελεγκτή σε εκείνες που βρέθηκαν στους πίνακες της ενότητας τέσσερα.

6. Συμπεράσματα και Προοπτικές

Ύστερα από την ολοκλήρωση των παραπάνω θεματικών ενοτήτων προκύπτει πως η χρήση ελεγκτών παρέχει μια σημαντική βελτίωση στα χαρακτηριστικά απόδοσης του συστήματος ελέγχου ταχύτητας ταξιδιού(cruise control system). Τυπικά, όπως επιβεβαιώθηκε και από τις παραπάνω συγκρίσεις μας, συνηθέστερη είναι η χρήση του PI controller. Επίσης, αξίζει να σημειωθεί πως το cruise control σύστημα έχει εξελιχθεί στα νέας τεχνολογίας αυτοκίνητα σε ένα νεοσύστατο σύστημα, το οποίο παρέχει επιπλέον πλεονεκτήματα σε σχέση με το συμβατικό cruise control σύστημα. Πρόκειται για το Adaptive Cruise Control System (ACC) ,το οποίο αυτόματα ρυθμίζει την ταχύτητα του οχήματος για να διατηρεί μια ασφαλή απόσταση από το προπορευόμενο όχημα. Στα πλαίσια αυτού του εξελιγμένου συστήματος υπάρχει δυνατότητα μια πλήρους σύγκρισης και ανάλυσης εκ νέου των ελεγκτών P,PI και PID. Οι αριθμητικοί παράμετροι ελέγχου K_p , K_i , K_d μπορούν αυτή τη φορά να υπολογιστούν κατάλληλα εκτός από τους τρόπους που χρησιμοποιήθηκαν στην εργασία μας, και με μεθόδους όπως αυτή του Ziegler - Nichols και Tyreus-Luyben , ούτως ώστε να λάβουμε επιθυμητά αποτελέσματα για κάθε ελεγκτή. Επιπλέον, διάφορα κριτήρια σφάλματος και προδιαγραφές στο τομέα του χρόνου μπορούν να συγκριθούν για την λήψη καλύτερης απόδοσης μείωσης της ταχύτητας-φρεναρίσματος.

Βιβλιογραφία

- [1] How Does the Cruise Control System in cars Work
<https://www.scienceabc.com/innovation/what-is-cruise-control-system-cars-work.html>
- [2] Measure the Drag Coefficient of Your Car
<https://www.instructables.com/id/Measure-the-drag-coefficient-of-your-car/>
- [3] Michael Fira, 2019 BMW 3 Series VS 2019 Audi A4
<https://www.topspeed.com/cars/2019-bmw-3-series-vs-2019-audi-a4-ar182814.html>
- [4] MATLAB 2018a, The MathWorks, Inc., Natick, Massachusetts, United States
- [5] BobYrok, "Frequency Responses and Bode Plots"(2009)
- [6] Karl Johan Aström Richard M. Murray, "Feedback Systems: An Introduction for Scientists and Engineers.", Version v2.10b (February 22, 2009)
- [7] R.C. Dorf, R.H. Bishop ,Σύγχρονα συστήματα αυτόματου ελέγχου, Θεσσαλονίκη [GR] : Α. Τζιόλα & Υιοί Α.Ε.,2016,ISBN: 978-960-418-537-5
- [8] Andrea Scherlozer, Mestaro Orsini, Sulvane Patole. Simulation and Numerical Analysis and Comparative Study of a PID Controller Based on Ziegler-Nichols and Auto Turning Method. 12th IEEE International Conference on Control and Automation, Jun 2016, Kathmandu, Nepal. hal-01571931v2
- [9] Pradhan, Rosy, et al. "Performance Evaluation of PID Controller for an Automobile Cruise Control System using Ant Lion Optimizer." Engineering Journal (Eng. J.) 21.5 (2017): 347-361.