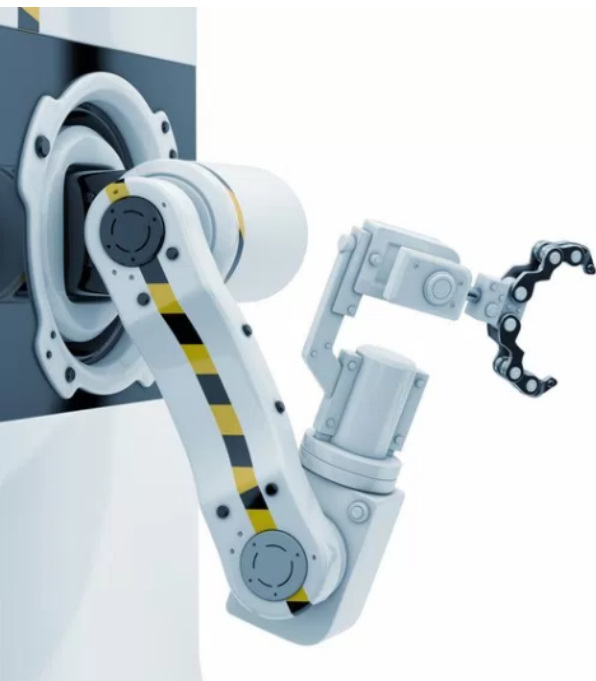


ΠΤΣΑΕ

## Εργασία στον βέλτιστο έλεγχο

Βέλτιστος έλεγχος ηλεκτρικού τρένου

---



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ  
ΠΟΛΥΤΕΧΝΕΙΟ

Ρούλιος Χαράλαμπος  
Α.Μ. 03114004  
Εξάμηνο 9<sup>ο</sup>

## 1 Μοντελοποίηση και ανάλυση βελτίστου ελέγχου

Με βάση τη δυναμική του μοντέλου της κίνησης του τρένου και το κριτήριο κόστους, μπορούμε να καταστώσουμε τις εξισώσεις του συστήματος αλλά και τις συνοριακές συνθήκες έτσι ώστε να καταλήξουμε σε ένα πρόβλημα συνοριακών τιμών δυο σημείων. Αρχικά, αν το κριτήριο κόστους είναι της μορφής:

$$J = g(x(\tau)) + \int_0^\tau L(x, u, t) dt$$

Τότε η Hamiltonian ορίζεται ως:

$$H(x, u, \lambda, t) = L(x, u, t) + \lambda^T f(x, u, t)$$

όπου  $\lambda$  οι άγνωστες συναρτήσεις των πολλαπλασιαστών lagrange. Στη προκειμένη περίπτωση προκύπτει ότι:

$$H(x, u, \lambda, t) = k_4 x_2 u + R u^2 + \lambda_1 x_2 - \lambda_2 (k_2 x_2^2 + k_1 x_2 - k_3 u) \quad (1)$$

Τώρα, η δυναμική των συναρτήσεων  $\lambda_1$  και  $\lambda_2$  προκύπτουν παίρνοντας το αντίθετο της μερικής παραγώγου της Hamiltonian ως προς  $\lambda$ . Συγκεκριμένα παίρνουμε:

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x_1} \Rightarrow \dot{\lambda}_1 = 0 \Rightarrow \lambda_1 = \text{const} \quad (2)$$

$$\dot{\lambda}_2 = -\frac{\partial H}{\partial x_2} \Rightarrow \dot{\lambda}_2 = \lambda_2 (k_1 + 2k_2 x_2) - k_4 u - \lambda_1 \quad (3)$$

Για να βρούμε τον βέλτιστο νόμο ελέγχου  $u$ , θα πρέπει να ελαχιστοποιήσουμε την  $H$  ως προς το  $u$ . Αφού η  $H$  είναι παραγωγίσιμη συνάρτηση, το βέλτιστο  $u$  μπορούμε να το λάβουμε ως εξής:

$$\frac{\partial H}{\partial u^*} = 0 \Rightarrow u^* = \frac{-(k_4 x_2 + k_3 \lambda_2)}{2R} \quad (4)$$

όμως λόγω του κορεσμού της εισόδου, τελικά το πραγματικό βέλτιστο  $u$  θα είναι της μορφής:

$$u_{real}^* = \min(\max(u^*, I_{min}), I_{max}) \quad (5)$$

Η εύρεση των παραπάνω εξισώσεων μπορεί να γίνει και με τη χρήση του MATLAB, με βάση τον κώδικα που ακολουθεί:

```
1 %% 4
2 clear all
3 %Constants
4 syms c1 c2 x1f x0 k1 k2 k3 k4 R Imin Imax T
5 % global c1 c2 x1f x0 k1 k2 k3 k4 R Imin Imax T
6 % c1 = 1000;
7 % c2 = 1000;
8 % x1f = 10;
9 % k1 = 0.5;
```

```

10 % k2 = 0.1;
11 % k3 = 1;
12 % k4 = 10;
13 % R = 0.3;
14 % Imin = -2;
15 % Imax = 2;
16 % T = 10;
17 % x0 = [0 0];
18
19 % State equations
20 syms x1 x2 p1 p2 u;
21 Dx1 = x2;
22 Dx2 = -k1*x2 -k2*x2^2 + k3*u;
23 % Cost function inside the integral
24 syms L;
25 L = k4 * x2 * u + R*u^2;
26 % Hamiltonian
27 syms H;
28 H = L + p1*Dx1 + p2*Dx2;
29 % Costate equations
30 Dp1 = -diff(H,x1);
31 Dp2 = -diff(H,x2);
32 % solve for control u
33 du = diff(H,u);
34 sol_u = solve(du, u);
35 % Substitute u to state equations
36 Dx2 = subs(Dx2, u, sol_u);
37 Dp2 = subs(Dp2, u, sol_u);

```

Εφόσον τώρα έχουμε ένα σύστημα τεσσάρων διαφορικών εξισώσεων  $(x_1, x_2, \lambda_1, \lambda_2)$ , με δύο γνωστές συνοριακές συνθήκες για τα  $x$  (αρχική θέση και ταχύτητα του τρένου), χρειαζόμαστε δύο ακόμη συνοριακές συνθήκες για την επίλυση του προβλήματος συνοριακών τιμών, οι οποίες θα ληφθούν με βάση το κριτήριο κόστους, από τη παρακάτω εξίσωση:

$$\lambda_1(\tau) = \frac{\partial g(x(\tau))}{\partial x_1} \Rightarrow \lambda_1(\tau) = \lambda_1 = 2c_1(x_1(\tau) - x_{1f}) \quad (6)$$

$$\lambda_2(\tau) = \frac{\partial g(x(\tau))}{\partial x_2} \Rightarrow \lambda_2(\tau) = 2c_2x_2(\tau) \quad (7)$$

Συνολικά λοιπόν, το σύστημα των τεσσάρων διαφορικών εξισώσεων μπορεί να συνοψιστεί σε μια εξίσωση της μορφής:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\lambda}_1 \\ \dot{\lambda}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -k_1 x_2 - k_2 x_2^2 + k_3 u_{real}^* \\ 0 \\ \lambda_2 (k_1 + 2k_2 x_2) - k_4 u_{real}^* - \lambda_1 \end{bmatrix} = \mathbf{F}(x_1, x_2, \lambda_1, \lambda_2) \quad (8)$$

με συνοριακές συνθήκες:

$$\begin{bmatrix} x_1(0) \\ x_2(0) \\ \lambda_1(\tau) \\ \lambda_2(\tau) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2c_1(x_1(\tau) - x_{1f}) \\ 2c_2 x_2(\tau) \end{bmatrix} \quad (9)$$

και όπου  $u_{real}^*$ , αυτό της εξίσωσης (5).

## 2 Αριθμητική επίλυση του TPBVP και προσομοίωση του συστήματος

Ξεκινώντας, αρχικά θα βάλουμε σε σχόλιο τη γραμμή 4 και θα αφαιρέσουμε τα σχόλια στις γραμμές 5-17, έτσι ώστε πλέον οι σταθερές να λάβουν συγκεκριμένη τιμή. Το σύστημα (8) με τις συνοριακές συνθήκες (9) είναι πολύ δύσκολο, έως αδύνατο να λυθεί με το χέρι, και απαιτεί να καταφύγουμε σε αριθμητικές μεθόδους επίλυσης. Ο κώδικας που επιλύει-προσομοιώνει και αναπαριστά γραφικά το σύστημα μετά την εφαρμογή βελτίστου ελέγχου είναι ο παρακάτω:

```

1  %% 5
2
3  % Initial guess for the solution
4  global samples
5  samples = 200;
6  time_lin = linspace(0,T,samples);
7  solinit = bvpinit(time_lin, [0 0 0.5 0.5]);
8
9  options = bvpset('Stats','on','RelTol',1e-1);
10
11 sol = bvp4c(@BVP_ode, @BVP_bc, solinit);
12
13 t = time_lin;
14 x = deval(sol,t);
15
16 % Calculate u(t) from x1,x2,p1,p2
17 global ut x1t_opt x2t_opt
18 ut = -(k3*x(4,:)+k4*x(2,:))/(2*R);
19 ut = min(max(ut,Imin),Imax);
20

```

```

21 x1t_opt = x(1,:);
22 x2t_opt = x(2,:); %for ricatti
23
24 %Calculate cost J
25 J = c1*(x(1,samples)-x1f)^2 + c2*x(2,samples)^2 + T*(k4*x(2,:)*ut' + R*(ut*ut'))/samples
26
27 %Plotting results
28 figure;
29 subplot(2,2,1)
30 plot(t, x(1,:));
31 xlabel('time')
32 title('Optimal x1(t)')
33 grid on
34
35 subplot(2,2,2)
36 plot(t, x(2,:), 'g');
37 xlabel('time')
38 title('Optimal x2(t)')
39 grid on
40
41 subplot(2,2,[3,4])
42 plot(t, ut, 'r');
43 xlabel('time')
44 title('Optimal saturated input u*')
45 grid on

```

Η συνάρτηση που πραγματοποιεί την αριθμητική επίλυση είναι η *bvp4c*, η οποία δέχεται τρία ορίσματα. Τα ορίσματα είναι τα εξής:

- Το όρισμα *solinit*, το οποίο περιέχει τον χρόνο διακριτοποιημένο, από το 0 έως το T και μια εκτίμηση της λύσης του συστήματος των 4 εξισώσεων.
- Το όρισμα *BVP\_ode*, το οποίο ουσιαστικά είναι μια συνάρτηση η οποία με παραμέτρους τον χρόνο  $t$  και τρέχον διάνυσμα κατάστασης  $\vec{y} = (x_1, x_2, \lambda_1, \lambda_2)$  αποτιμά την συνάρτηση  $\mathbf{F}$  του συστήματος, δηλαδή της εξίσωσης (8). Παρακάτω φαίνεται ο κώδικας της συνάρτησης αυτής:

```

1 function dydt = BVP_ode(t,y)
2     global k1 k2 k3 k4 R Imin Imax
3     u = -(k3*y(4) + k4*y(2))/(2*R);
4     u = min(max(u,Imin),Imax);
5     dydt = [y(2); -k2*y(2)^2-k1*y(2)+k3*u; 0; y(4)*(k1+2*k2*y(2)) - k4*u - y(3)];
6 end

```

- Το όρισμα  $BVP\_bc$  είναι και αυτό μια συνάρτηση η οποία αναπαριστά τις συνοριακές συνθήκες. Σαν  $ya$  ορίζεται η τιμή του διανύσματος  $\vec{y}$  στον αρχικό χρόνο (εδώ 0) ενώ σαν  $yb$  ορίζεται η τιμή στον τελικό χρόνο (εδώ T). Η συνάρτηση αυτή είναι η εξής:

```

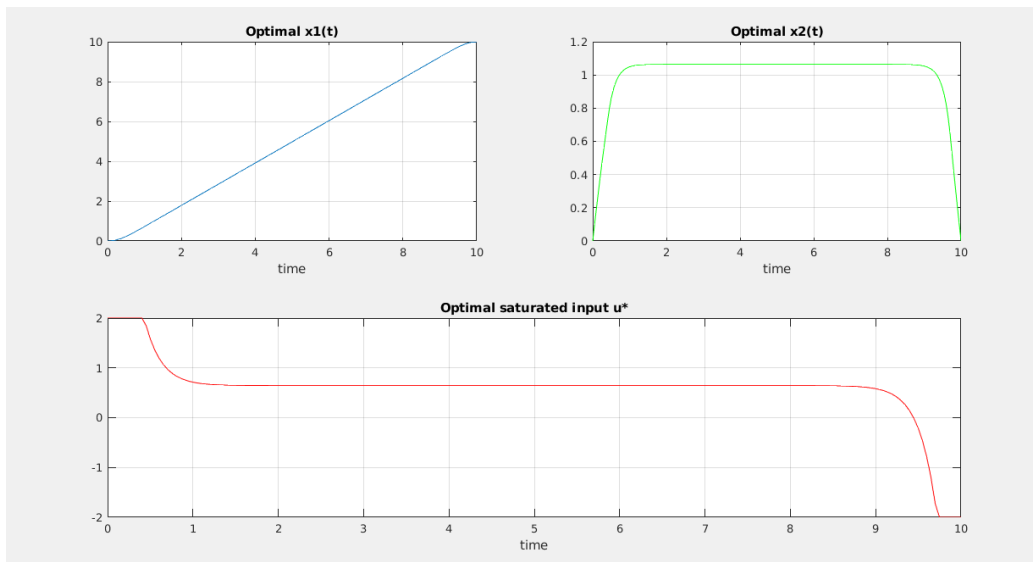
1  % The boundary conditions:
2  function res = BVP_bc(ya,yb)
3      global x0 c1 c2 x1f;
4      res = [ya(1) - x0(1); ya(2) - x0(2); yb(3) - 2*c1*(yb(1) - x1f); yb(4) - 2*c2*(yb(1) - x1f)];
5  end

```

Έπειτα από τον ορισμό των συναρτήσεων αυτών, το σύστημα μπορεί να λυθεί με την εντολή `bvp4c(@BVP_ode,@BVP_bc,solinit);`, δηλαδή είναι πλέον γνωστά τα διανύσματα κατάστασης και οι πολλαπλασιαστές langrange στον χρόνο. Με τη γνώση αυτών των συναρτήσεων κάθε χρονική στιγμή, μπορεί να βρεθεί η είσοδος  $u(t)$  στον χρόνο, πράγμα που γίνεται στις γραμμές 18 και 19. Στη συνέχεια υπολογίζεται το κόστος  $J$  και γίνεται plot της θέσης, της ταχύτητας και της εισόδου ελέγχου για το τρένο. Το κόστος σε αυτή τη φάση ισούται με:

$$J = 64.5937$$

ενώ παρατηρούμε ότι ο βέλτιστος έλεγχος οδηγεί σε μια σύντομη φάση επιτάχυνσης, σε μια μεγάλη φάση σταθερής ταχύτητας και σε μια πάλι σύντομη φάση επιβράδυνσης του τρένου:



Σχήμα 1: Θέση, ταχύτητα και ροπή του τρένου μετά από βέλτιστο έλεγχο

Να σημειωθεί ότι το  $x_1$  καταλήγει ακριβώς στην θέση  $x_{1f}$  με  $x_2(T) = 0$  λόγω των μεγάλων σταθερών  $c_1, c_2$ .

### 3 Προσομοίωση του συστήματος με την βέλτιστη είσοδο αλλά διαφορετικές αρχικές συνθήκες

Η διαδικασία επίλυσης και προσομοίωσης του συστήματος θα είναι η ίδια με τη προηγούμενη, με τη διαφορά ότι η είσοδος πλέον είναι δεδομένη. Έτσι, για την επίλυση του συστήματος δεν χρειάζεται να προσδιοριστούν πολλαπλασιαστές lagrange και έτσι έχουμε να λύσουμε TPBVP 2 εξισώσεων με τις 2 νέες συνοριακές συνθήκες (το αρχικό σύστημα των  $x_1, x_2$ ). Ο κώδικας που πραγματοποιεί την προσομοίωση φαίνεται παρακάτω, έχοντας επιλέξει  $x_{10} = 0.4$  ως αρχική θέση και  $x_{20} = 0.6$  ως αρχική ταχύτητα:

```
1 %% 6
2 % same u input, different initial state
3 x0 = [0.4 0.6];
4
5 solinit2 = bvpinit(time_lin, [x0(1) x0(2)]);
6 sol = bvp4c(@BVP_ode2, @BVP_bc2, solinit2);
7 x = deval(sol,t);
8
9 J = c1*(x(1,samples)-x1f)^2 + c2*x(2,samples)^2 + T*(k4*x(2,:)*ut' + R*(ut*ut'))/samples
10
11 %Plotting results
12 figure;
13 subplot(2,2,1)
14 plot(t, x(1,:));
15 xlabel('time')
16 title('x1(t) with previous input')
17 grid on
18
19 subplot(2,2,2)
20 plot(t, x(2,:), 'g');
21 xlabel('time')
22 title('x2(t) with previous input')
23 grid on
24
25 subplot(2,2,[3,4])
26 plot(t, ut, 'r');
27 xlabel('time')
28 title('Previous optimal saturated input u*')
29 grid on
```

Η αντίστοιχη συνάρτηση *BVP\_ode2* είναι η:

```

1 function dydt = BVP_ode2(time,y)
2     global k1 k2 k3 ut T samples
3     index = max(ceil(time*samples/T),1);
4     u = ut(index);
5     dydt = [y(2); -k2*y(2)^2-k1*y(2)+k3*u];
6 end

```

ενώ η συνάρτηση *BVP\_bc2* για τις νέες συνοριακές συνθήκες είναι:

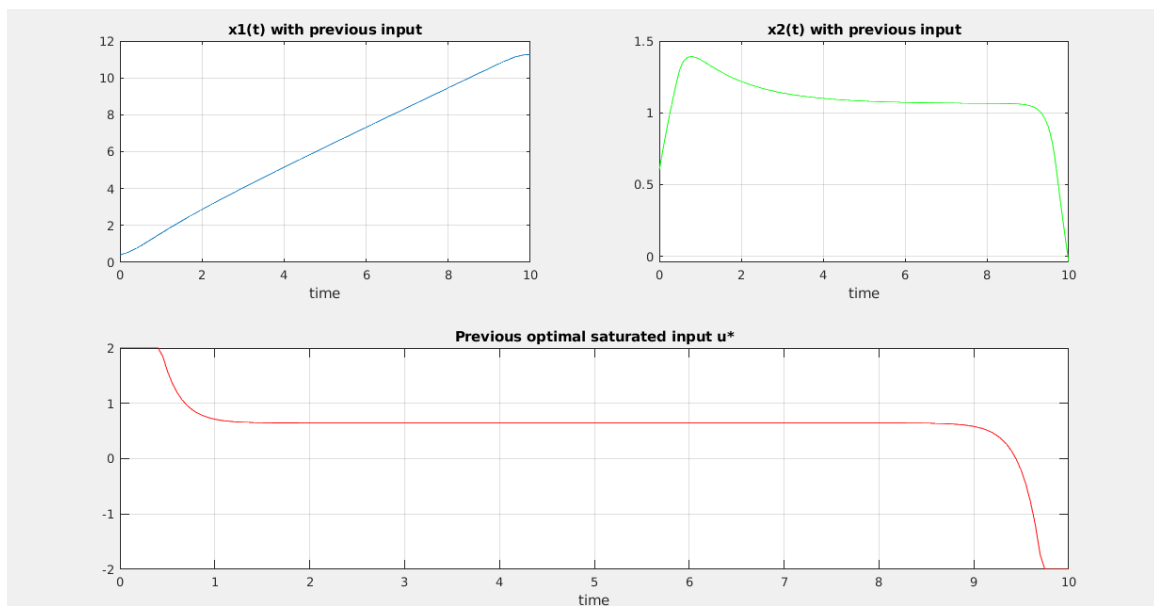
```

1 % The boundary conditions:
2 function res = BVP_bc2(ya,yb)
3     global x0;
4     res = [ya(1) - x0(1); ya(2) - x0(2)];
5 end

```

Είναι φανερό ότι οι συναρτήσεις αυτές πλέον επιστρέφου διανύσματα διαστάσεως 2, μιας και πλέον θέλουμε να επιλύσουμε 2 εξισώσεις. Στη συνέχεια φαίνεται το νέο κόστος μετά τη προσομοίωση και οι γραφικές παραστάσεις των νέων μεγεθών με την παλιά είσοδο:

$$J = 1.6680e + 03$$



**Σχήμα 2:** Θέση, ταχύτητα και ροπή του τρένου με τις νέες αρχικές συνθήκες και τη προηγούμενη είσοδο

Παρατηρούμε ότι το κόστος αυξήθηκε κατά 2 τάξεις μεγέθους, ενώ οι τελικές τιμές των μεταβλητών κατάστασης δεν είναι οι επιθυμητές (το  $x_1$  καταλήγει κοντά στο 12 και η τελική



ταχύτητα είναι ελαφρώς αρνητική). Το γεγονός αυτό οφείλεται στο ότι εφαρμόζεται έλεγχος **ανοιχτού βρόχου** με τη παλιά είσοδο, δηλαδή εφαρμόζεται η είσοδος "τυφλά" με βάση τον χρόνο και όχι με βάση τις καταστάσεις και έτσι ο έλεγχος αφήφά τυχόν μεταβολές των καταστάσεων.

## 4 Εύρεση βέλτιστης πρόσθετης εισόδου $v(y)$ και επίλυση της Ricatti

Το  $y$  αποτελεί μια μικρή απόκλιση από τη βέλτιστη τροχιά (η οποία υπολογίστηκε προηγουμένως) και προσεγγιστικά έχει δυναμική γραμμικού συστήματος. Αν συμβολίσουμε  $x_{opt}(t)$  τη βέλτιστη τροχιά που υπολογίστηκε προηγουμένως, τότε λαμβάνοντας της μερικές παραγώγους της συνάρτησης  $f$ , προκύπτουν οι εξής  $A(t)$  και  $B(t)$ :

$$A(t) = \begin{bmatrix} 0 & 1 \\ 0 & -2k_2x_{opt}(t) - k_1 \end{bmatrix} \quad (10)$$

και

$$B(t) = \begin{bmatrix} 0 \\ k_3 \end{bmatrix} \quad (11)$$

Ακόμη, βλέπουμε ότι αν  $y = [y_1 \ y_2]^T$  είναι το διάνυσμα κατάστασης για την απόκλιση από τη βέλτιστη τροχιά, τότε το κριτήριο κόστους είναι της μορφής:

$$J = y(\tau)^T S y(\tau) + \int_0^\tau (y^T Q y + u^T R u) dt \quad (12)$$

όπου:

$$S = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}, \quad Q = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad R = 1$$

Συνεπώς έχουμε να βελτιστοποιήσουμε ένα linear quadratic πρόβλημα. Το πρόβλημα αυτό τυπικά μπορεί να λυθεί με την εύρεση του μεταβατικού πίνακα κατάστασης της Hamiltonian Matrix αλλά κάτι τέτοιο θα μας έδινε ξανά μια λύση ανοιχτού βρόχου  $v(t)$ . Για να γίνει εύρεση λύσης κλειστού βρόχου  $v(y)$ , θα πρέπει να γίνει υπόθεση ότι οι πολλαπλασιαστές lagrange είναι της μορφής  $\lambda(t) = P(t)y$  και η οποία υπόθεση τελικά οδηγεί στην εφαρμογή ελέγχου της μορφής  $v(y) = -R^{-1}B^T P y$  και στην εξίσωση Ricatti για τον πίνακα  $P(t)$ :

$$\dot{P} = -PA - A^T P + PBR^{-1}B^T P - Q, \quad P = P^T > 0 \quad (13)$$

με τη συνοριακή συνθήκη:

$$P(\tau) = 2S \quad (14)$$

Στη δική μας περίπτωση, ο πίνακας  $P$  είναι ένας  $2 \times 2$  πίνακας, ο οποίος είναι συμμετρικός. Έτσι, θα πρέπει να βρούμε τις 3 άγνωστες συναρτήσεις  $p_{11}(t), p_{12}(t), p_{22}(t)$ . Αντικαθιστώντας

λοιπόν στην εξίσωση (13) τους πίνακες  $A, B, R, Q$  αλλά και τις σταθερές  $k_1, k_2, k_3$  με τις τιμές τους, προκύπτει το εξής σύστημα διαφορικών εξισώσεων:

$$\begin{bmatrix} \dot{p}_{11} & \dot{p}_{12} \\ \dot{p}_{12} & \dot{p}_{22} \end{bmatrix} = \begin{bmatrix} p_{12}^2 - 2 & p_{12}(x_{2opt}/5 + 1/2) - p_{11} + p_{12}p_{22} \\ p_{12}(x_{2opt}/5 + 1/2) - p_{11} + p_{12}p_{22} & 2p_{22}(x_{2opt}/5 + 1/2) - 2p_{12} + p_{22}^2 - 2 \end{bmatrix}$$

ή αλλιώς:

$$\begin{bmatrix} \dot{p}_{11} \\ \dot{p}_{12} \\ \dot{p}_{22} \end{bmatrix} = \begin{bmatrix} p_{12}^2 - 2 \\ p_{12}(x_{2opt}/5 + 1/2) - p_{11} + p_{12}p_{22} \\ 2p_{22}(x_{2opt}/5 + 1/2) - 2p_{12} + p_{22}^2 - 2 \end{bmatrix} \quad (15)$$

με συνοριακές συνθήκες:

$$\begin{bmatrix} p_{11}(\tau) \\ p_{12}(\tau) \\ p_{22}(\tau) \end{bmatrix} = \begin{bmatrix} 40 \\ 0 \\ 40 \end{bmatrix} \quad (16)$$

Όμοια με πριν, καλούμαστε να λύσουμε ένα TPBVP με 3 αγνώστους αυτή τη φορά. Ο κώδικας MATLAB που βρίσκει τους πίνακες A,B και  $\dot{P}$  συμβολικά και τελικά λύνει το σύστημα (15) με τις συνοριακές συνθήκες (16) παρουσιάζεται παρακάτω:

```

1  %% 7
2  % closed loop v(y) = -R^(-1)*B'*P(t)*y
3  % Need to calculate P(t) through the Ricatti equation
4  syms p11 p12 p22
5  global S B
6  R = 1;
7  Q = [2 0; 0 2];
8  S = [40 0; 0 40];
9
10 f = [x2; -k1*x2-k2*x2^2+k3*u];
11
12 A = [diff(f,x1) diff(f,x2)];
13 B = diff(f,u);
14
15 P = [p11 p12; p12 p22];
16
17 %Ricatti
18 DP = - P*A - A'*P + P*B*inv(R)*B'*P - Q;
19
20 solinit_ricatti = bvpinit(time_lin, [40 0 40]);
21 sol_ricatti = bvp4c(@BVP_ode_ricatti, @BVP_bc_ricatti, solinit_ricatti);
22

```

```

23 p = deval(sol_ricatti,t);
24
25 global p11t p12t p22t
26 p11t = p(1,:);
27 p12t = p(2,:);
28 p22t = p(3,:);
29
30 % v(y) = -R^(-1)*B'*P(t)*y and y = x_real-x_opt
31 % v(y) = -(1/R)*(p12*y1+p22*y2)

```

με τη συνάρτηση *BVP\_ode\_ricatti* να είναι η:

```

1 function dpdt = BVP_ode_ricatti(time,y)
2     global x2t_opt T samples
3     index = max(ceil(time*samples/T),1);
4     x2 = x2t_opt(index);
5     dpdt = [y(2)^2-2; y(2)*(x2/5 + 1/2) - y(1) + y(2)*y(3); 2*y(3)*(x2/5 + 1/2) - 2*y(2)
6 end

```

και τη συνάρτηση *BVP\_bc\_ricatti* να είναι η:

```

1 % The boundary conditions:
2 function res = BVP_bc_ricatti(ya,yb)
3     global S;
4     res = [yb(1) - S(1,1); yb(2) - S(1,2); yb(3) - S(2,2)];
5 end

```

## 5 Προσομοίωση συστήματος με τη νέα είσοδο ελέγχου και τις διαφορετικές αρχικές συνθήκες

Η διαδικασία που θα ακολουθηθεί εδώ θα είναι η ίδια με το ερώτημα 6 (ενότητα 3 στην παρούσα αναφορά) με τη διαφορά ότι πλέον σαν είσοδος θα εφαρμοστεί η ποσότητα  $u_{opt}(t) + v(y)$ , όπου  $u_{opt}$  είναι η αρχική βέλτιστη είσοδος του ερωτήματος 5. Στην πραγματικότητα, λόγω του κορεσμού, η είσοδος που θα εφαρμοσθεί θα είναι της μορφής:

$$u_{new} = \min(\max(u_{opt} + v(y), I_{min}), I_{max}) \quad (17)$$

Να σημειωθεί ότι το  $u_{opt}$  είναι γνωστό κάθε χρονική στιγμή από προηγούμενως, ενώ το  $v(y)$ , λαμβάνοντας υπόψη ότι  $y = [x_1 - x_{1opt} \ x_2 - x_{2opt}]^T$  εφόσον το  $y$  είναι η απόκλιση του  $x$  από τη βέλτιστη τροχιά και ότι το  $P(t)$  υπολογίστηκε προηγούμενως, ισούται με:

$$v(y) = -R^{-1}B^T P(t)y \quad (18)$$

Έτσι, ο κώδικας που επιλύει το σύστημα διαφορικών των 2 μεταβλητών κατάστασης με τις νέες αρχικές συνθήκες, που υπολογίζει τη νέα βέλτιστη είσοδο, το κόστος  $J$  και κάνει plot των αντίστοιχων μεγεθών της προσομοίωσης, είναι:

```

1  %% 8
2  x0 = [0.4 0.6];
3  R = 1;
4
5  sol = bvp4c(@BVP_ode8, @BVP_bc2, solinit2);
6  x = deval(sol,t);
7
8  y = [x(1,:)-x1t_opt; x(2,:)-x2t_opt];
9  v = -(1/R)*(p12t.*y(1,:) + p22t.*y(2,:));
10 u_new = ut + v;
11 u_new = min(max(u_new,Imin),Imax);
12
13 J = c1*(x(1,samples)-x1f)^2 + c2*x(2,samples)^2 + T*(k4*x(2,:)*u_new' + R*(u_new*u_new'));
14
15
16 %Plotting results
17 figure;
18 subplot(2,2,1)
19 plot(t, x(1,:));
20 xlabel('time')
21 title('x1(t) with new input')
22 grid on
23
24 subplot(2,2,2)
25 plot(t, x(2,:), 'g');
26 xlabel('time')
27 title('x2(t) with new input')
28 grid on
29
30 subplot(2,2,[3,4])
31 plot(t, u_new, 'r');
32 xlabel('time')
33 title('New optimal saturated input u*')
34 grid on

```

ενώ η νέα συνάρτηση *BVP\_ode8* που χρησιμοποιείται, είναι η:

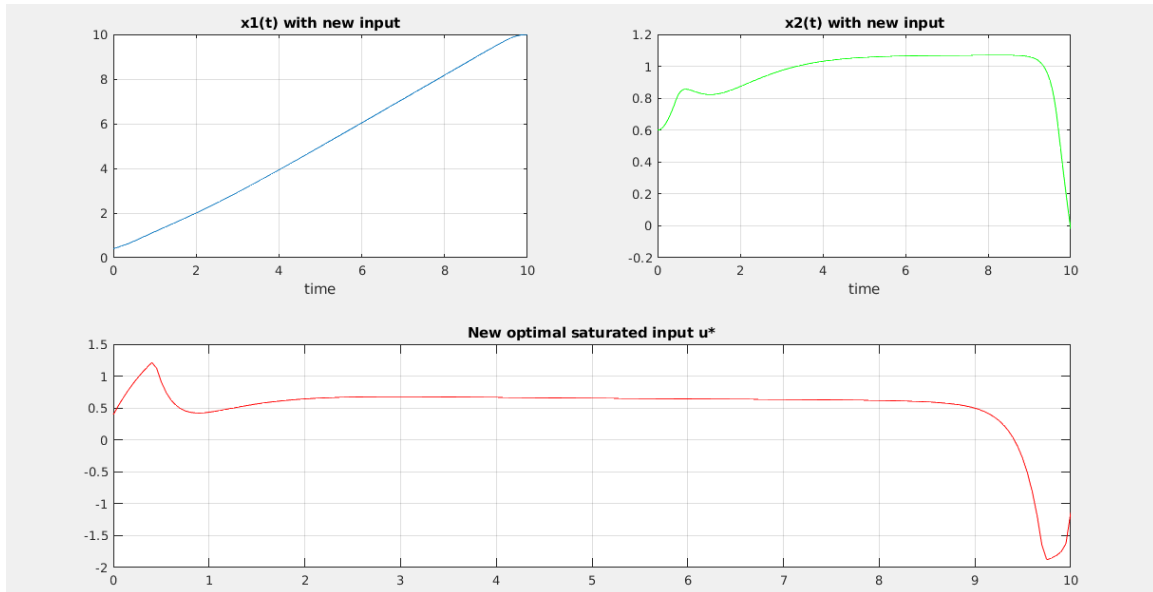
```

1 function dxdt = BVP_ode8(time,x)
2     global k1 k2 k3 ut T samples p12t p22t x1t_opt x2t_opt R Imin Imax
3
4     index = max(ceil(time*samples/T),1);
5
6     y = [x(1) - x1t_opt(index); x(2) - x2t_opt(index)];
7
8     %v = -(1/R)*B'*P*y = -(1/R)*(p12*y1+p22*y2);
9     v = -(1/R)*(p12t(index)*y(1) + p22t(index)*y(2));
10    u = ut(index) + v;
11    u = min(max(u,Imin),Imax);
12
13    dxdt = [x(2); -k2*x(2)^2-k1*x(2)+k3*u];
14 end

```

Τα αποτελέσματα της τελευταίας αυτής προσομοίωσης φαίνονται παρακάτω, με το κόστος  $J$  και τις γραφικές παραστάσεις της θέσης, ταχύτητας και εισόδου, με αλλαγμένες τις αρχικές συνθήκες:

$$J = 60.0522$$



Σχήμα 3: Θέση, ταχύτητα και ροπή του τρένου με τις νέες αρχικές συνθήκες και τη νέα είσοδο

Σαν τελικές παρατηρήσεις μπορούν να σημειωθούν τα εξής. Αρχικά παρατηρούμε ότι με τη νέα στρατηγική ελέγχου, παρόλο που οι αρχικές συνθήκες είναι πλέον διαφορετικές, το τρένο στον τελικό χρόνο καταλήγει στην ίδια κατάσταση με στη βέλτιστη τροχιά, δηλαδή στη θέση

10 με ταχύτητα 0, σε αντίθεση με τη περίπτωση του ερωτήματος 6. Για τον λόγο αυτό επίσης παρατηρούμε δραστική μείωση του κόστους συγκριτικά με το ερώτημα 6.

Ο λόγος που συμβαίνει αυτό, είναι διότι πλέον εφαρμόσαμε έλεγχο κλειστού βρόχου, δηλαδή ο έλεγχος εξαρτάται από τις καταστάσεις  $x$  (εφόσον  $y = y(x)$ ). Σε αντίθεση με το ερώτημα 6, στο οποίο εφαρμόζαμε τυφλά μια είσοδο στον χρόνο ανεξάρτητα από το ποια μπορεί να είναι τα  $x_1, x_2$ , πλέον η είσοδος εξαρτάται από την τωρινή κατάσταση του τρένου, πράγμα που κάνει τον έλεγχο ανεκτικό σε μεταβολές των καταστάσεων, ενώ η συγκεκριμένη είσοδος είναι τέτοια ώστε με την ανάδραση να προσπαθεί να επιτύχει ελαχιστοποίηση του  $y$ , δηλαδή της απόκλισης από τη βέλτιστη τροχιά.