Strojniški vestnik - Journal of Mechanical Engineering 52(2006)12, 852-862
UDK - UDC 681.511:517.9
Izvirni znanstveni članek - Original scientific paper (1.01)

Programljivi logični krmilniki na temelju rešitve algebraične Riccatijeve enačbe

Programmable Logic Controllers Based on the Algebraic Riccati Equation Solution

Primož Podržaj - Zoran Kariž
(Fakulteta za strojništvo, Ljubljana)

V prispevku je predstavljena sinteza podoptimalnega krmilnika, ki sloni na rešitvi algebraične Riccatijeve enačbe (ARE). Predstavljen je numerični postopek, s katerim lahko pridemo do te rešitve. V praksi se parametri krmiljenega sistema mnogokrat precej razlikujejo od tistih v ARE. V tem primeru sta vprašljivi optimalnost in celo stabilnost krmilnega sistema. Zelo uporabno bi torej bilo, če bi lahko zasnovali prilagodljivi linearni podoptimalni krmilnik. Tak krmilnik bi bil zmožen odkriti spremembe v parametrih sistema in prirediti parametre.

(Ključne besede: sistemi prilagodljivi, enačbe Riccatijeve, krmilniki logični, krmilniki programljivi, algoritmi numerični)

This paper deals with the synthesis of a suboptimal controller, based on the solution of the algebraic Riccati equation (ARE). The numerical procedure for obtaining the solution is presented. In applications the controlled system parameters often differ from the ones used in the ARE. In this case the optimality of the control system and even its stability are questionable. Therefore, it would be very useful to design an adaptive linear suboptimal controller. Such a controller should be able to detect changes in the system parameters and adjust its parameters.

(Keywords: adaptive systems, Riccati equations, programmable logic controllers, numerical algorithms)

0 UVOD

Linearni časovno neodvisni sistem (LČN) z diferencialno enačbo stanj:

0 INTRODUCTION

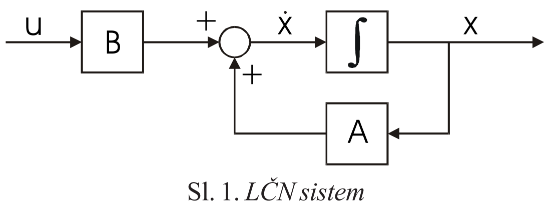
A linear time-invariant (LTI) system with the state differential equation:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1),$$

prikazan na sliki 1 je optimalen, če mu je dodana povratna zveza:

shown in Figure 1 is said to be optimal if the feed-back:

$$u(t) = -F(t)x(t)$$



Sl. 1. *LČN sistem*
Fig. 1. *LTI system*

tako, da je integralski kriterij:

is added in such a way that the following cost functional:

$$\int_{t_0}^{t_1}\left[x^T(t)R_1(t)x(t)+u^T(t)R_2(t)u(t)\right]\cdot dt+x^T(t_1)P_1x(t_1)$$

najmanjši. Pri tem so:

- $R_1(t)$ in $R_2(t)$ pozitivno definitni simetrični matriki za $t_0 = t = t_1$,
- P_1 pozitivno poldefinitna simetrična matrika.

Pokazati se da [1], da je treba za minimizacijo funkcionala rešiti naslednjo matrično Riccatijevo enačbo (MRE):

is minimized. This means that

- $R_1(t)$ and $R_2(t)$ are positive definite symmetrical matrices for $t_0 = t = t_{1,}$
- P_1 is a positive semi-definite symmetrical matrix

It can be shown [1], that in order to minimize the cost functional the following matrix Riccati equation (MRE) has to be solved:

$$\dot{P}(t)=R_1-P(t)BR_2^{-1}B^TP(t)+P(t)A+A^T(t)P$$

Če rešitev MRE konvergira, potem je limita:

If the solution of the MRE converges, then the limiting solution:

$$P=\lim_{t\to\infty}P(t)$$

rešitev naslednje algebraične Riccatijeve enačbe (ARE):

is the solution of the following algebraic Riccati equation (ARE):

$$R_1-PBR_2^{-1}B^TP+PA+A^TP=0$$

Krmilnik, ki je zasnovan na temelju te rešitve, se imenuje podoptimalni krmilnik.

The controller based on this solution is called a suboptimal controller.

1 OPIS PROBLEMA

Podoptimalni krmilniki se veliko uporabljajo. Dejstvo pa je, da se lastnosti krmiljenega objekta, dane z matriko A, spreminjajo s časom zaradi staranja in spremenljivih delovnih razmer, čeprav se predpostavlja, da so časovno neodvisne. Matrika A je lahko časovno neodvisna, če opazujemo tipični čas prehodne funkcije krmilnega sistema, toda je časovno odvisna med dobo trajanja sistema. Če se dejanska matrika stanja A_R razlikuje od tiste, ki smo jo uporabili v ARE (matrika A) za Δ, se lahko pojavijo težave.

1 PROBLEM FORMULATION

Suboptimal controllers are frequently used. However, it is a fact that the properties of the controlled system given in matrix A change over time, due to ageing and different working conditions etc., even though they are assumed to be time-invariant. The matrix A may be time-invariant if we consider a typical transient response time of a system, but it is time variant during the lifetime of the system. Many problems can be encountered if the real system matrix, A_R, which differs from the one used in the ARE (matrix A) by the amount Δ, is introduced.

$$A_R=A+\Delta\ ;\quad \Delta=\begin{bmatrix}\delta_{11} & \cdots & \delta_{1n}\\ \vdots & \ddots & \vdots\\ \delta_{n1} & \cdots & \delta_{nn}\end{bmatrix}$$

Zaradi tega se izgubi podoptimalnost sistema. V mnogih primerih je vprašljiva celo njegova stabilnost. Zaželeno bi torej bilo, da se zgradi krmilnik, ki bi lahko:

- določil dejansko matriko stanja A_R,
- rešil ARE na podlagi dejanske matrike stanja A_R.

Consequently, the system's suboptimal performance is lost. Even its stability may be questionable in many cases. Therefore, it is desirable to design a controller capable of:

- determining the real system matrix, A_R,
- solving the ARE considering the real system matrix, A_R,

2 DOLOČEVANJE DEJANSKE MATRIKE STANJA A_R

Celoten postopek določevanja dejanske matrike stanja A_R je razdeljen na dva koraka, in sicer v odvisnosti od tega ali matrika stanja A_R ni poznana (prvi korak), ali pa je vsaj približno poznana (drugi korak).

2.1 Prvi korak

Če v enačbi (1) uporabimo dejansko matriko stanja v prvem koraku, dobimo:

2 DETERMINING THE REAL SYSTEM MATRIX, A_R

The whole procedure for determining the real system matrix, A_R, is divided into two steps, depending on whether the system matrix, A_R, is not known at all (first step) or is approximately known (second step).

2.1 First step

When the real system matrix, A_{R1}, is used in the first step, Equation (1) implies the following equality:

$$\dot{x}(t) - Bu(t) = A_{R1}x(t) \tag{2}$$

Matrika stanja A_{R1} pomeni linearno preoblikovanje vektorskega prostora $\Re^n$ v vektorski prostor $\Re^n$ in je enolično določena z baznimi vektorji prostora $\Re^n$. Torej je za določitev elementov sistemske matrike treba najti n linearno neodvisnih vektorjev stanj in njihove slike. Predpostavimo, da lahko merimo vse komponente vektorja stanj x in da je med fazo branja vhodni signal signal u enak 0 (sl. 2).

Uporabljena je bila tudi predpostavka, da je mogoče izračunati odvode vseh komponent vektorja stanja. Postopek, ki ga lahko uporabimo v ta namen, je opisan na primer v [8] in [9]. Levo stran enačbe (2) lahko torej določimo za vsak čas t. Recimo, da krmilnik med fazo branja (u=0) za nek čas vzorčenja t_S prebere $x(0^+)$, $x(T_S)$, $x(2T_S)$ in tako naprej, dokler ne najde n linearno neodvisnih vektorjev stanj $x_1, x_2, ..., x_n$. Potem lahko oblikujemo naslednji matriki:

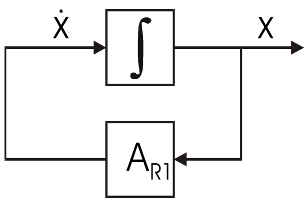
The system matrix, A_{R1}, represents a linear transformation of the space $\Re^n$ into the space $\Re^n$ and is uniquely determined by the images of the vectors on the basis of $\Re^n$. Therefore, in order to determine the elements of the system matrix we only need to find n linearly independent state vectors x and their images. Let us suppose that all the components of the state vector x can be measured, and that during the acquisition phase the control signal u equals 0 (Fig. 2).

Another supposition, that the derivatives of all the components of the state vector x can be calculated, was also made. The procedure, which might be used in this case, is given for example in [8] and [9]. Therefore the left-hand side of Equation (2) can be determined at any time t. Now, let us say that the controller reads $x(0^+)$, $x(T_S)$, $x(2T_S)$ and so on during the acquisition phase (u=0) for a specified sampling time T_S, until n linearly independent state vectors $x_1, x_2, ..., x_n$ are found. Then, the following matrices can be formed:

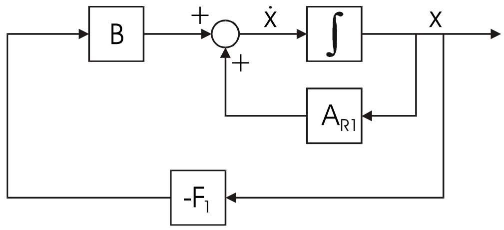
$$\dot{X} = \begin{bmatrix}\dot{x}_1 & \dot{x}_2 & \cdots & \dot{x}_n\end{bmatrix} \quad ; \quad X = \begin{bmatrix}x_1\, x_2 \cdots x_n\end{bmatrix}$$

Dejansko matriko stanja A_{R1} lahko torej določimo iz naslednje enačbe:

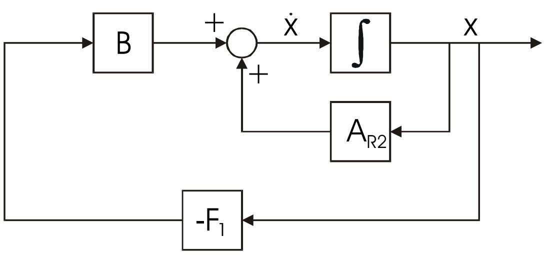
Therefore, the real system matrix, A_{R1}, can be determined by the following equation:



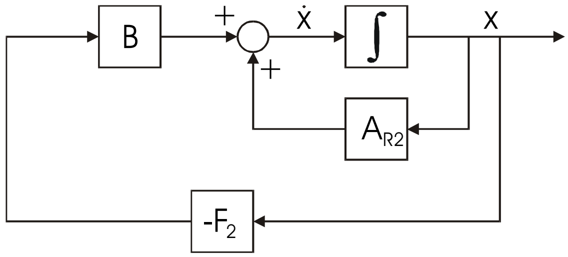
Sl. 2. *Prvi korak, faza branja*
Fig. 2. *First-step acquisition phase*



Sl. 3. *Prvi korak, delovna faza*
Fig. 3. *First-step working phase*



Sl. 4. *Drugi korak, faza branja*
Fig. 4. *Second-step acquisition phase*



Sl. 5. *Drugi korak, delovna faza*
Fig. 5. *Second-step working phase*

$$A_{R1} = \dot{X} \cdot X^{-1}$$

Z določeno matriko stanja A_{R1} lahko pridemo do rešitve Riccatijeve enačbe. Njena rešitev P_1 se nato uporabi za določitev ustrezne povratne matrike F_1. S tem se faza branja konča. Sistem brez povratne zveze je spremenjen v sistem s povratno zvezo, prikazan na sliki 3.

Ta postopek je treba izvesti le prvič, ko uporabimo tak tip krmilnika na sistemu, o katerem ne vemo ničesar, razen razsežnosti vektorja stanj. Če je matrika stanja A_R (A_{R1}) vsaj približno poznana, potem je pametneje uporabiti povratno matriko F_1, ki ustreza tej matriki stanja in nadaljevati z drugim korakom.

After the system matrix, A_{R1}, is determined, the Riccati equation can be solved. Its solution P_1 is then used to find the appropriate feedback matrix, F_1. At this moment the acquisition phase is ended and the working phase starts. The open-loop system is changed to the closed-loop system shown in Figure 3.

But this procedure only needs to be implemented the first time we use this type of controller on a system we know nothing about, except the number of states. If the system matrix, A_R (A_{R1}), is known approximately, then it is better to use the feedback matrix F_1 associated with this system matrix and proceed immediately to the second step.

2.2 Drugi korak

Ko je matrika stanja določena (ali pa je uporabljena približna), je treba analizirati sistem s povratno zvezo, prikazan na sliki 4.

Med fazo branja je uporabljena povratna matrika F_1 iz prvega koraka. Dejanska matrika stanja je tekoča in ima zato indeks 2. Sistem opišemo z diferencialno enačbo:

2.2 Second step

Once the system matrix has been determined (or an approximate matrix has been used), the closed-loop system shown in Figure 4 should be analyzed.

During the acquisition phase the feedback matrix F_1 from the first step is used. The real system matrix, however, is current and therefore has index 2. The system is described by the differential equation:

$$\dot{x}(t) = \left(A_{R2} + BF_1\right)x(t)$$

Krmilnik mora tako kot v prvem koraku prebrati $x(0^+)$, $x(T_S)$, $x(2T_S)$ in tako naprej. Nato sestavimo matriki X in $\dot{X}$. Realno sistemsko matriko v drugem koraku A_{R2} lahko po končani fazi branja v drugem koraku določimo na podlagi naslednje enačbe:

The controller has to read $x(0^+)$, $x(T_S)$, $x(2T_S)$ and so on, as in the first iteration. Then the matrices X and $\dot{X}$ can be formed. The real system matrix in the second step, A_{R2}, can then be determined after the acquisition phase of the second step is ended, according to the following equation:

$$A_{R2} = \dot{X}X^{-1} - BF_1$$

Ko je nova povratna matrika F_2 določena, se lahko začne delovna faza v drugem koraku. Končni blokovni diagram je prikazan na sliki 5.

Drugi korak se nato iterativno ponavlja.

After the new feedback matrix F_2 is determined the working phase of the second step can begin. The associated block diagram is shown in Figure 5.

The second step is then iteratively continued.

3 REŠEVANJE ARE

Uporabljeni sta bili dve metodi reševanja ARE, in sicer:
- Metoda z uporabo enačb Ljapunova,
- Potterjeva metoda.

3.1 Metoda z uporabo enačb Ljapunova

Ta metoda se je izkazala za dokaj počasno in zato ne bo opisana. Podroben opis je podan v [6] in [7].

3.2 Potterjeva metoda

Ta metoda je bila uporabljena za reševanje ARE. Sestavimo matriko M_r reda 2x2 [1]:

3 SOLUTION OF THE ARE

Two methods for solving the ARE were considered:
- a method using Lyapunov equations
- Potter's method

3.1 Method using Lyapunov equations

This method turned out to be rather slow and will therefore not be described. A detailed description can be found in [6] and [7].

3.2 Potter's method

This method was used in order to solve the ARE. Let us consider the following 2x2 matrix M_r [1]:

$$M_r = \begin{bmatrix} A & -B \cdot R_2^{-1} \cdot B^T \\ -R_1 & -A^T \end{bmatrix}$$

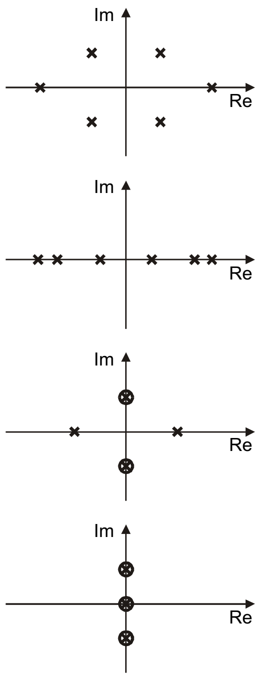
Matriko R_1 lahko zapišemo v obliki:

The matrix R_1 can be expressed as:

$$R_1 = H \cdot H^T$$

Če je matrična dvojica:
- (A,B) ustaljiva in
- (A,H) pregledljiva,

If the matrix pair:
- (A,B) is stabilizable
- (A,H) is observable



Sl. 6. *Mogoče razporeditve lastnih vrednosti matrike reda 6x6*
Fig. 6. *Possible eigenvalue locations of a 6x6 matrix*

potem se vse lastne vrednosti matrike M_r pojavljajo v dvojicah (λ_i,$-\lambda_i$). Nekaj mogočih primerov je prikazanih na sliki 6.

Naj bo lastni vektor ali korenski vektor y_i matrike M_r, ki ustreza lastni vrednosti λ_i z negativno realno komponento, oblike:

then all the eigenvalues of matrix M_r appear in pairs (λ_i,$-\lambda_i$). Some possible examples are shown in Figure 6.

Let the eigenvector or generalized eigenvector y_i of the matrix M_r corresponding to λ_i, which has a negative real part, be:

$$y_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix}$$

Potem se rešitev ARE dobi z [5]:

Then, the solution of the ARE is given by [5]:

$$P = \left[v_1, \cdots, v_n\right] \cdot \left[u_1, \cdots, u_n\right]^{-1}$$

3.2.1 Izračun lastnih vrednosti in lastnih vektorjev

Da pridemo do rešitve ARE, je najprej treba dobiti lastne vektorje matrike M_r. Ta postopek se največkrat izvede v štirih korakih, in sicer:

- z balansiranjem matrike [4],
- s preoblikovanjem matrike v Hessenbergovo [3],
- s QR algoritmom [3],
- z izračunom lastnih vektorjev [3].

3.2.1 Calculation of the eigenvalues and eigenvectors

In order to solve the ARE we must find the eigenvectors of the M_r. This procedure is usually done in four steps, by:

- matrix balancing [4],
- reducing a matrix to the Hessenberg form [3],
- performing the QR algorithm [3],
- computing the eigenvectors [3].

4 ZGRADBA SISTEMA

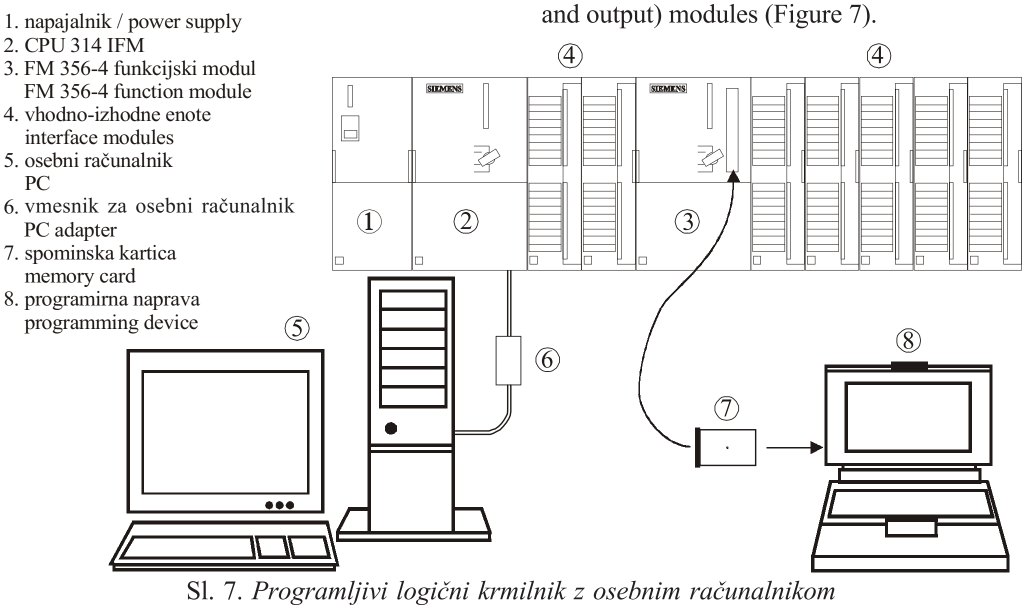
4.1 Zgradba krmilnika

Uporabljen je bil programljivi logični krmilnik SIEMENS S7-300 s funkcijskim modulom FM 356-4 in dodatnimi vhodno izhodnimi enotami (slika 7).

4 SYSTEM DESIGN

4.1 Controller design

As an application of the introduced method the controller was built using the SIEMENS S7-300 programmable logic controller along with the FM 356-4 function module and several interface (input and output) modules (Figure 7).



Sl. 7. *Programljivi logični krmilnik z osebnim računalnikom*
Fig. 7. *PLC with supporting PC*

Spominska kartica EPROM se uporablja za vgradnjo in zagon operacijskega sistema ter za nalaganje uporabniškega programa. Za vgradnjo opravilnega sistema na spominsko kartico je potrebna posebna programirna naprava. Uporabniški program pa se lahko spreminja in shranjuje z uporabo osebnega računalnika in vmesnika zanj, ki je priključen na osrednjo procesno enoto (CPU). Program se naloži v CPU, ki je prek vodila (BUS) povezan s funkcijskim modulom (FM).

A flash EPROM memory card is used to install and boot the operating system and to load the user software. A special programming device is needed to install the operating system on the memory card. The user software, on the other hand, can be modified and saved with the PC along with the appropriate PC adapter, which is connected to the CPU. The software is downloaded to the CPU, which is connected to the function module (FM) via a backplane BUS.

4.2 Modeliranje krmiljenega sistema

Za simulacijo krmiljenega objekta je bil uporabljen iterativni analogni računalnik Meda 41TC.

4.2 Controlled system modelling

A Meda 41TC iterative analogue computer was used to simulate the controlled system.

5 REZULTATI

Kot primer je prikazan sistem z matrikama:

5 RESULTS

As an example, a system with the following matrices:

$$A_R = \begin{bmatrix} 1 & 1 & 4 \\ 2 & 3 & 1 \\ 4 & 2 & 5 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \tag{3}$$

in začetnim pogojem:

and the initial condition:

$$x(0) = \begin{bmatrix} 3 \\ -3 \\ 6 \end{bmatrix}$$

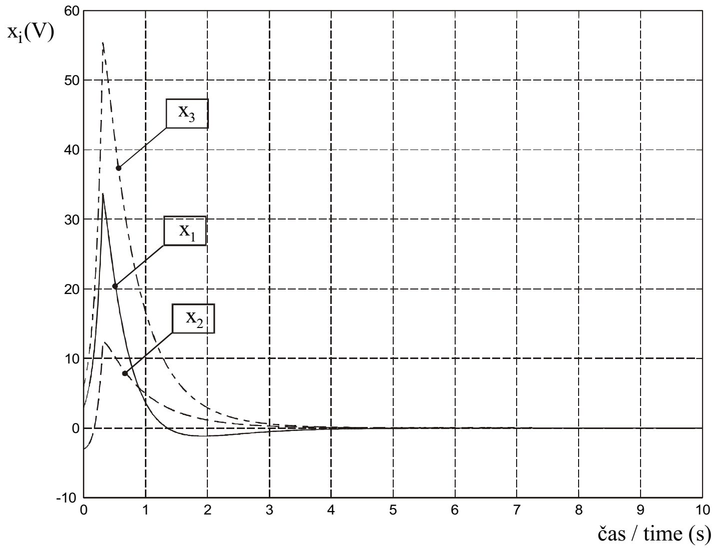
Matriki R_1 in R_2 v integralskem kriteriju sta:

was used. The matrices R_1 and R_2 in the cost functional were:

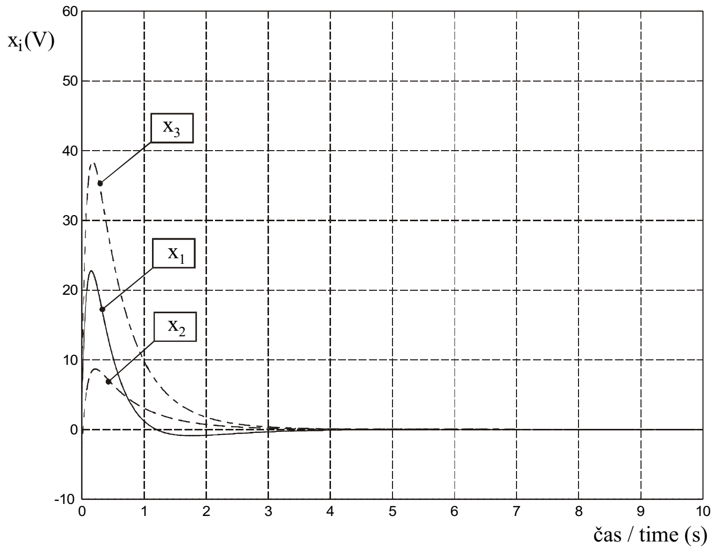
$$R_1 = \begin{bmatrix} 3 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix} \quad ; \quad R_2 = [1]$$

Odgovor prilagodljivega podoptimalnega krmilnika (tri komponente vektorja stanj) v prvem koraku za T_s=0,1 je prikazan na sliki 8. Dejanska matrika stanja A_R je bila na analognem računalniku namerno modelirana tako, da je sistem brez povratne zveze nestabilen. Zaradi tega je na sliki 8 jasno vidna faza branja. Rezultate lahko primerjamo z odgovorom linearnega podoptimalnega krmilnika, katerega povratna matrika F ustreza matriki A_R (sl. 9). Očitno je, da je v primeru, ko je povratna matrika F v neprilagodljivem krmilniku točna, le-ta boljši od podoptimalnega. Do sprememb pa pride, če se matrika stanja A, za katero je izračunana povratna matrika F, le malo razlikuje od dejanske sistemske matrike A_R. Recimo, da je dejanska matrika stanja A_R enaka kakor v enačbi (3), povratna matrika neprilagodljivega krmilnika pa je izračunana za matriko A, definirano v enačbi (4).

The response of an adaptive suboptimal controller (three components of the state vector) during the first step for T_s=0.1 is shown in Fig. 8. The real system matrix, A_R, was deliberately modelled on an analogue computer in such a way that the open-loop system is unstable. Therefore, the acquisition phase can clearly be seen in Fig. 8. These results can be compared to the response of a linear suboptimal controller whose feedback matrix F is associated with the matrix A_R (Fig. 9). It is obvious that an adaptive controller does not perform as well as a non-adaptive controller if the feedback matrix in the non-adaptive controller is accurate. The situation changes, however, if the system matrix A with which the feedback matrix of an non-adaptive controller is associated differs from the real system matrix, A_R, only slightly. Let us say that the real system matrix, A_R, is the same as in Eq. (3), but the non-adaptive controller's feedback matrix is associated with the matrix A defined in Eq. (4).



Sl. 8. *Odgovor prilagodljivega linearnega podoptimalnega krmilnika*
Fig. 8. *The response of an adaptive linear suboptimal controller*



Sl. 9. *Idealni odgovor neprilagodljivega linearnega podoptimalnega krmilnika*
Fig. 9. *Ideal non-adaptive linear suboptimal controller response*

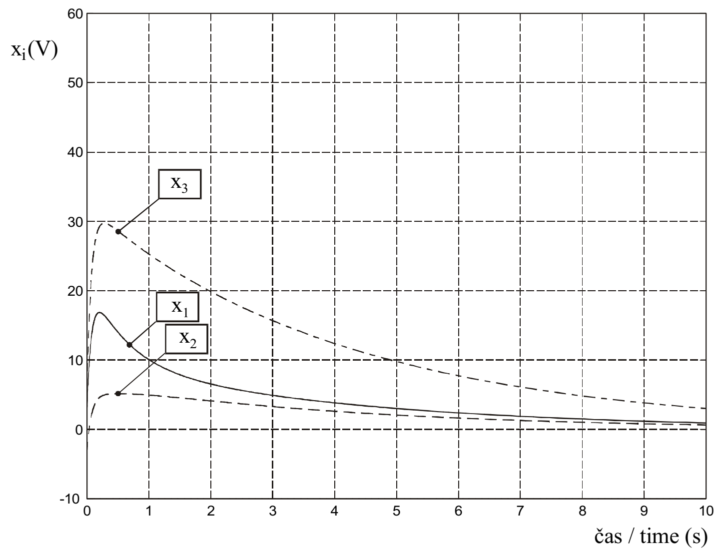
$$A = \begin{bmatrix} 1 & 1,1 & 4,1 \\ 2 & 3,1 & 1,1 \\ 4 & 2 & 4,8 \end{bmatrix} \quad (4).$$

Odgovor neprilagodljivega krmilnika za ta primer je prikazan na sliki 10.

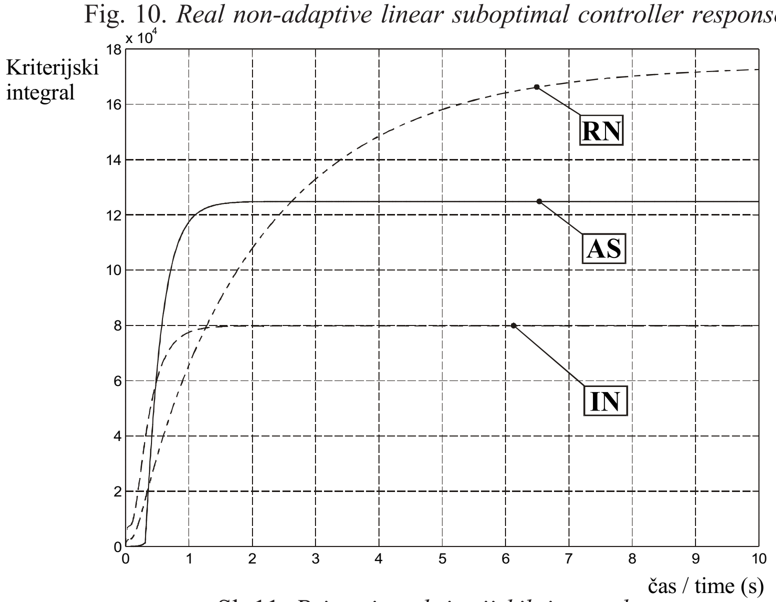
Sklepamo lahko, da se obnašanje neprilagodljivega linearnega podoptimalnega krmilnika že pri majhnih spremembah parametrov sistema hitro poslabšuje. Kriterijski integral za vse tri primere (prilagodljivi podoptimalni (AS), idealni neprilagodljivi (IN), realni neprilagodljivi (RN)) je prikazan na sliki 11.

The response of the non-adaptive linear controller for such a case is shown in Figure 10.

We can conclude that with only a slight change in the system parameters the performance of the non-adaptive linear suboptimal controller deteriorates rapidly. The cost functionals for all three cases (adaptive suboptimal (AS), ideal non-adaptive (IN), real non-adaptive (RN)) are shown in Figure 11.



Sl. 10. *Realni odgovor neprilagodljivega linearnega podoptimalnega krmilnika*
Fig. 10. *Real non-adaptive linear suboptimal controller response*



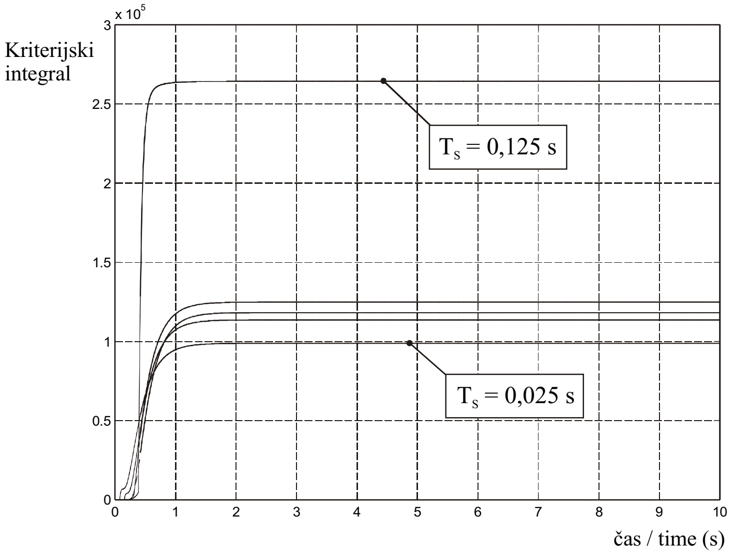
Sl. 11. *Primerjava kriterijskih integralov*
Fig. 11. *Comparison of the cost functionals*

Jasno je, da se obnašanje prilagodljivega linearnega podoptimalnega krmilnika spreminja v povezavi s časom vzorčenja T_S, še posebej, če je sistem brez povratne zveze nestabilen, kajti med fazo branja v prvem koraku le-te ni. Slika 12 prikazuje kriterijske funkcionale za različne čase vzorčenja (od T_S=0,025s do T_S=0,125s s korakom 0,025s).

Jasno je, da je treba za izboljšano obnašanje zmanjšati čas vzorčenja T_S. To je mogoče v teoriji in do neke mere tudi na analognem računalniku. V praksi pa lahko pride do problemov, ker se prebrani vektorji stanj zaradi

It is to be expected that the performance of an adaptive linear suboptimal controller varies with sampling time, T_S, especially if the open-loop system is unstable, because during the acquisition phase of the first step, there is no feedback. Figure 12 shows cost functionals depending on different sampling times (from T_S=0.025s to T_S=0.125s with a 0.025s step).

It is obvious that in order to improve the performance of the controller we must decrease the sampling time, T_S. This can be easily done in theory and to some extent on the analogue computer. In practice, on the other hand, problems may occur because the acquired



Sl. 12. *Primerjava časov vzorčenja*

Fig. 12. *Cost functional versus sampling time*

šumov ne izražajo v pravi dejanski sistemski matriki A_R. Ti učinki se povečujejo, ko gremo s časom vzorčenja proti 0.

state vectors during the acquisition phase do not result in the correct real system matrix, A_R, due to the noise and other disturbances. These effects are magnified as the sampling time approaches 0.

6 SKLEP

Jasno so bile prikazane prednosti prilagodljivega linearnega podoptimalnega krmilnika pred neprilagodljivim. Te prednosti se povečujejo s povečanimi odstopanji v parametrih sistema. V tem primeru prilagoldjivi krmilnik ohrani svoje zmožnosti. Obnašanje neprilagodljivega krmilnika pa se hitro poslabša.

Če matrike stanja ne poznamo, imamo med fazo branja v prvem koraku sistem brez povratne zveze. Obnašanje sistema je torej odvisno od trajanja te faze, le-to pa od časa vzorčenja. Nadaljnje raziskave je torej treba osredotočiti na določevanje optimalnega časa vzorčenja za različne jakosti šuma.

6 CONCLUSION

The advantages of an adaptive linear suboptimal controller over a non-adaptive controller have been clearly demonstrated. These advantages increase when deviations in the system parameters become larger. In this situation, the adaptive linear suboptimal controller maintains its capability. The performance of a non-adaptive controller, on the other hand, deteriorates rapidly.

However, if the approximate system matrix is not known in advance, we are basically operating the open-loop system during the first-step acquisition phase. The overall performance is therefore dependent on the duration of this phase, which further depends on the sampling time. Further research should, therefore, focus on a determination of the optimal sampling time with regard to the presence of different levels of noise.

7 LITERATURA
7 REFERENCES

[1] H. Kwaakernak, R. Sivan (1972) Linear optimal control systems, *John Wiley & Sons Inc.*

[2] F. L. Lewis (1988) Optimal control, *John Wiley & Sons Inc.*

[3] David M. Young, Robert Todd Gregory (1973) A survey of numerical mathematics, *Addison—Wesley Publishing Company.*

[4] J. H. Wilkinson, C. Reinsch (1973) Die Grundlehren der mathematischen Wissenschaften: Linear Algebra, *Springer Verlag.*

[5] Katsuhisa Furuta, Akira Sano, Derek Atherton (1988) State variable methods in automatic control, *John Wiley & Sons Inc.*
[6] E. J. Davison, F. T. Man (1968) The numerical solution of A'Q+QA=-C, *IEEE Transactions on Automatic Control*, August, str.: 448-449.
[7] David L. Kleinman (1968) On an iterative technique for Riccati equation computations, *IEEE Transactions on Automatic Control*, February, str.: 114-115.
[8] В.А.Бесекерский, В.В.Изранцев (1987) Системы автоматического управления с микроЭВМ, *«Наука»*
[9] Н.Н.Иващенко (1978) Автоматическое регулирование, *«Машиностроение»*

Naslov avtorjev: dr. Primož Podržaj
doc. dr. Zoran Kariž
Univerza v Ljubljani
Fakulteta za strojništvo
Aškerčeva 6
1000 Ljubljana
primoz.podrzaj@fs.uni-lj.si
zoran.kariz@fs.uni-lj.si

Authors' Address: Dr. Primož Podržaj
Doc. Dr. Zoran Kariž
University of Ljubljana
Faculty of Mechanical Eng.
Aškerčeva 6
1000 Ljubljana, Slovenia
primoz.podrzaj@fs.uni-lj.si
zoran.kariz@fs.uni-lj.si

Prejeto:
Received: 12.4.2005

Sprejeto:
Accepted: 25.10.2006

Odprto za diskusijo: 1 leto
Open for discussion: 1 year