Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A computing capability test for a switched system control design using the Haris-Rogers method

Mohd Amin AT-TASNEEM[1,2], Sallehuddin Mohamed HARIS[‡2], Zulkifli Mohd NOPIAH[3]

([1]*Faculty of Mechanical Engineering, University Malaysia Pahang, Pekan 26600, Malaysia*)

([2]*Department of Mechanical and Materials Engineering, Faculty of Engineering and the Built Environment, Universiti Kebangsaan Malaysia, UKM Bangi 43600, Malaysia*)

([3]*Unit of Fundamental Engineering Studies, Faculty of Engineering and the Built Environment, Universiti Kebangsaan Malaysia, UKM Bangi 43600, Malaysia*)

E-mail: tasneem@ump.edu.my; {salleh, zmn}@eng.ukm.my

Received Mar. 19, 2012; Revision accepted July 31, 2012; Crosschecked Sept. 11, 2012

Abstract: The problem of finding stabilizing controllers for switched systems is an area of much research interest as conventional concepts from continuous time and discrete event dynamics do not hold true for these systems. Many solutions have been proposed, most of which are based on finding the existence of a common Lyapunov function (CLF) or a multiple Lyapunov function (MLF) where the key is to formulate the problem into a set of linear matrix inequalities (LMIs). An alternative method for finding the existence of a CLF by solving two sets of linear inequalities (LIs) has previously been presented. This method is seen to be less computationally taxing compared to methods based on solving LMIs. To substantiate this, the computational ability of three numerical computational solvers, LMI solver, cvx, and Yalmip, as well as the symbolic computational program Maple were tested. A specific switched system comprising four second-order subsystems was used as a test case. From the obtained solutions, the validity of the controllers and the corresponding CLF was verified. It was found that all tested solvers were able to correctly solve the LIs. The issue of rounding-off error in numerical computation based software is discussed in detail. The test revealed that the guarantee of stability became uncertain when the rounding off was at a different decimal precision. The use of different external solvers led to the same conclusion in terms of the stability of switched systems. As a result, a shift from using a conventional numerical computation based program to using computer algebra is suggested.

Key words: Linear inequalities, Hybrid systems, Stability, Common quadratic Lyapunov function, Numerical computation, Symbolic computation
doi:10.1631/jzus.C1200074 **Document code:** A **CLCnumber:** TP273

1 Introduction

Stability is an important area of research in the dynamics and control of switched systems, as established concepts for linear time invariant systems do not fully hold for this type of system. It is possible to have an unstable switching system made up entirely of stable subsystems, and the converse is also true.

The key idea of stability in a hybrid system is that if we can find the Lyapunov function for each subsystem, then by suitably constraining the switching mechanism, we can possibly guarantee stability. Consider a switched linear system composed of N subsystems (Zhu *et al.*, 2007):

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_{\sigma(t)}\boldsymbol{x}, \qquad (1)$$

where $\sigma(t)$: $[0, +\infty)\rightarrow\Lambda$ is a right continuous function, $\Lambda=\{1, 2, \ldots, N\}$. To ensure the stability of switched system (1) under arbitrary switching, a common quadratic Lyapunov function is sufficient. A quadratic Lyapunov function, $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}$, with positive definite matrix $\boldsymbol{P}>\boldsymbol{0}$ is called a common quadratic Lyapunov function (CQLF) of $\{\boldsymbol{A}_\lambda|\lambda\in\Lambda\}$ if

‡ Corresponding author

$$\boldsymbol{P}\boldsymbol{A}_\lambda+\boldsymbol{A}_\lambda^{\mathrm{T}}\boldsymbol{P}<\boldsymbol{0} \quad \forall\lambda\in\Lambda. \tag{2}$$

To support the function, King and Shorten (2006) found that for a finite collection of stable matrices $\mathcal{A}=\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m\}$, there is no CQLF if, and only if, there are positive semidefinite matrices $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m$, not all zero, such that

$$\sum_{i=1}^{m}(\boldsymbol{A}_i\boldsymbol{P}_i+\boldsymbol{P}_i\boldsymbol{A}_i^{\mathrm{T}})=\boldsymbol{0}.$$

A multitude of methods for designing stabilizing controllers for switching systems have been introduced. Most methods are based on Lyapunov stability, where establishing the existence of a common Lyapunov function (CLF) or a multiple Lyapunov function (MLF) lies at the centre of the analysis. Included among these is a convex design method for switching feedback gains based on fixed Lyapunov matrices (Montagner *et al.*, 2006). A quadratic Lyapunov function with a common matrix is used to derive a stabilizing switching control strategy that guarantees, for any arbitrary switching rule:

1. The location of the poles of each linear subsystem of a continuous-time switched linear system is inside a chosen circle in the left-hand half of the complex plane.

2. There is a minimum disturbance attenuation level for the closed-loop switched system.

These features are considered to be important because the first will improve the dynamic response assigning bounds for overshoot, settling time, and frequency of oscillation. The second will ensure the robustness of the switched system facing disturbances that are energy signals. The sufficient linear matrix inequality (LMI) design condition based on quadratic Lyapunov functions with a common matrix was presented. This LMI condition allows the determination of switched state feedback gains that stabilize the closed loop system, including the pole locations and robustness of the system.

The idea is to find a multicontroller that effectively switches among several linear time-invariant controllers, keeping in mind that one single controller would not be able to stabilize the complex system for every possible switching sequence. Hespanha and Morse (2002) believed that if the multicontroller is chosen properly, then for every switch between subsystems, the system can be guaranteed to be uniformly exponentially stable. Every stable controller transfer matrix is expressed using Youla parameterization, which means that instead of switching between the controllers, the system is switched between the corresponding values of the parameter. The development of a multicontroller in Youla parameterization must reach a common Lyapunov function. The discrete-time version of this technique was presented by Stewart and Dumont (2006).

Eq. (2), used to ensure the stability of switched system (1) under arbitrary switching, is a common quadratic Lyapunov function which is sufficient to become a conservative way to determine the stability of a switched system. However, Mason and Shorten (2003) conjectured that asymptotic stability can be found in positive linear systems with arbitrary switching by testing the Hurwitz stability of the convex hull of a set of Metzler matrices. For a real polynomial

$$p(s)=a_0s^n+a_1s^{n-1}+\ldots+a_{n-1}s+a_n,$$

its Hurwitz matrix is a square structured $n\times n$ matrix which is built up of the polynomial constants as

$$\boldsymbol{H}(p)=\begin{bmatrix} a_1 & a_3 & a_5 & a_7 & \cdots & 0 \\ a_0 & a_2 & a_4 & a_6 & \cdots & 0 \\ 0 & a_1 & a_3 & a_5 & \cdots & 0 \\ 0 & a_0 & a_2 & a_4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_n \end{bmatrix}.$$

According to Zhang *et al*. (2008), there are two criteria to test the stability of a Hurwitz matrix. One of the indirect methods is by testing the eigenvalue of the matrix including computing its Jordan canonical function and calculating the invariant factors. However, it is not an easy task to complete these computations due to the computational complexity. Another method deals with the stability based directly on the entries of a given matrix.

The state transformation method is a way to transform the initial states of a system to other forms in which a stabilizing strategy for the system may be found before it is transformed back into its original coordinates. Davrazos and Koussoulas (2002) used a canonical form in canonical coordinates as a medium to find a stabilizing control strategy for switched linear systems. The problem of stabilization via state feedback control is then illustrated in canonical composition. A state estimator is introduced to approximate the state variables in the feedback loop, and a separation property is established for the whole system before it is constructively transformed back to its original orientation. The main objective is to find a suitable control input and switching laws which guarantee that the system will be uniformly asymptotically stable. Before the transformation process, it is essential to ensure that the system is controllable and observable. The system is stabilizable only if the state space matrices are Hurwitz or equivalent, and the unstable modes of the state space matrices are controllable (Geng, 2010). Linear state transformation was introduced by Decarlo *et al.* (2000) to find a stable convex combination for a class of switched systems. The linear state transformation decomposes each subsystem into stable and unstable parts, and for each stable part there naturally exists a Lyapunov function. Under some conditions imposed on the original switched system, the sum of these Lyapunov functions is shown to be a Lyapunov function for a convex combination of the whole switched system.

Another important finding in research on the stabilizability of switched systems was obtained through the application of Lie algebra, which is generated from stable matrices of a linear system. Lie algebra is a graphical structure which is usually used when learning graphical objects (Zhai *et al.*, 2006). Research by Liberzon *et al.* (1999) raised many questions. They conjectured that if the Lie algebra which is generated from matrices of stable linear time invariant systems is nilpotent (which means that the Lie brackets of sufficiently high order equal zero), then the system is asymptotically stable for any switching signal. They also proved that this conjecture is true for two subsystems and third-order Lie brackets.

Commonly, CLFs and MLFs are found by formulating a set of LMIs. However, solving LMIs is computationally taxing. In Haris *et al*. (2007), a method was proposed for finding controller gains that guarantee the stability of the whole system, where all subsystems are of second order. This method, which we call the Haris-Rogers method, reduces the problem to solving two sets of linear inequalities (LIs), significantly reducing the computational burden inherent in solving LMIs. This method was realized in the form of a Matlab program (Haris and Rogers, 2008). In this study, we tested the computational ability of some solvers in successfully executing the Haris-Rogers method.

2 The Haris-Rogers method

In this section, we briefly describe the Haris-Rogers method. Proofs have been omitted here. Interested readers may refer to Haris (2006) and Haris *et al*. (2007) for details. In the following, $A[i, j]$ refers to the element in the ith row and jth column of matrix $\boldsymbol{A}$, and $x[i]$ refers to the ith element of vector $\boldsymbol{x}$.

The class of switching systems considered here is denoted by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{B}_i u, \tag{3}$$

where $\boldsymbol{x}\in\mathbb{R}^{2\times1}$, $u\in\mathbb{R}$, $\boldsymbol{A}_i\in\mathbb{R}^{2\times2}$, $\boldsymbol{B}_i\in\mathbb{R}^{2\times1}$, and i=1, 2, …, N. The method was developed to guarantee quadratic stability of system (3), i.e., to determine if there exists a set of feedback control laws $u_i=\boldsymbol{K}_i\boldsymbol{x}$ such that all $\boldsymbol{A}_i+\boldsymbol{B}_i\boldsymbol{K}_i$ share a common quadratic Lyapunov function $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{x}$.

The common quadratic Lyapunov function which ensures the stability of the system, if it exists, can be found if the subsystems that make up the switched system are in the Brunovsky controllable canonical form. Hence, each subsystem i is first converted into Brunovsky coordinates by

$$\begin{cases} \boldsymbol{AB}_i = \boldsymbol{A}_i\cdot\boldsymbol{B}_i,\ \boldsymbol{ABI}_i = \boldsymbol{AB}_i - \dfrac{\boldsymbol{A}_i\boldsymbol{AB}_i}{\boldsymbol{B}_i\,|\,\boldsymbol{AB}_i}[2]\cdot\boldsymbol{B}_i, \\ \boldsymbol{C}_i = \dfrac{1}{\boldsymbol{ABI}_i\,|\,\boldsymbol{B}_i}, \end{cases} \tag{4}$$

where $\boldsymbol{C}_i$ is the transformation matrix which converts subsystem i into the Brunovsky form:

$$\dot{\boldsymbol{x}} = \tilde{\boldsymbol{A}}_i\boldsymbol{x} + \tilde{\boldsymbol{B}}_i u.$$

However, $(\tilde{\boldsymbol{A}}_i,\ \tilde{\boldsymbol{B}}_i)$ are in different coordinate frames; i.e., every subsystem in Brunovsky form lies in a space that is separate from the other subsystems. For uniformity, all subsystems are transformed once more so as to align with frame 1. $\boldsymbol{T}_k$ (k=1, 2, …, n−1) is the transformation matrix that brings frame k+1 onto frame 1, given by

$$\boldsymbol{T}_k=\boldsymbol{C}_1\boldsymbol{C}_{k+1}^{-1}. \tag{5}$$

The matrix

$$\boldsymbol{M}=\begin{bmatrix} m_1 & m_2 \\ m_2 & m_3 \end{bmatrix}$$

characterizes the quadratic Lyapunov function

$$\boldsymbol{M}\tilde{\boldsymbol{A}}_i+\tilde{\boldsymbol{A}}_i^{\mathrm{T}}\boldsymbol{M}<\boldsymbol{0} \tag{6}$$

if, and only if, $m_2>0$. $\boldsymbol{M}_k=\boldsymbol{T}_k^{\mathrm{T}}\boldsymbol{M}\boldsymbol{T}_k$ retains the behavior of $\boldsymbol{M}$ under transformation

$$\boldsymbol{T}_k=\begin{bmatrix} t_1 & t_2 \\ t_3 & t_4 \end{bmatrix} \tag{7}$$

if, and only if, the quadratic equation

$$t_3t_4y_a^2+(t_2t_3+t_1t_4)y_a+t_1t_2>0 \tag{8}$$

has a positive solution $y_a>0$.

Each k is classified into one of three categories depending on the value of $Z_k=t_3\cdot t_4$ as follows:

$$k\in\begin{cases} S_p, & \text{if } Z_k>0, \\ S_n, & \text{if } Z_k<0, \\ S_z, & \text{if } Z_k=0. \end{cases} \tag{9}$$

Now, define

$$\begin{cases} P_k=t_2t_3+t_1t_4,\ Q_k=t_1t_2, & \text{if } k\in S_z, \\ P_k=\dfrac{t_1}{t_3},\ Q_k=\dfrac{t_2}{t_4}, & \text{if } k\in S_p\cup S_n. \end{cases} \tag{10}$$

For every $k\in S_z$, a linear polynomial is formed as

$$L_k=P_ky_a+Q_k, \tag{11}$$

and for every $k\in S_p\cup S_n$ a quadratic polynomial is formed as

$$R_k=y_a^2+(P_k+Q_k)y_a+P_kQ_k. \tag{12}$$

A value for y_a is chosen from the range of y_a that meets the solution of the first set of LIs (13):

$$\begin{cases} R_k<0, & \text{if } k\in S_n, \\ R_k>0, & \text{if } k\in S_p, \\ L_k>0, & \text{if } k\in S_z. \end{cases} \tag{13}$$

Without loss of generality, let m_1=1. Then, a common quadratic Lyapunov function for the subsystems in the Brunovsky coordinates of frame 1 is defined by

$$\begin{cases} \boldsymbol{M}_1\tilde{\boldsymbol{A}}_i+\tilde{\boldsymbol{A}}_i^{\mathrm{T}}\boldsymbol{M}_1<\boldsymbol{0} \\ \boldsymbol{M}_1=\begin{bmatrix} 1 & m_2 \\ m_2 & m_3 \end{bmatrix}>\boldsymbol{0} \end{cases} \tag{14}$$

if and only if

$$\begin{cases} m_3+(P_k+Q_k)m_2+P_kQ_k>0, & k\in S_p, \\ m_3+(P_k+Q_k)m_2+P_kQ_k<0, & k\in S_n, \\ P_k\cdot m_2+Q_k>0, & k\in S_z. \end{cases} \tag{15}$$

Letting $m_3=m_2^2+\varepsilon$, where ε is a small positive number, the second set of LIs (15) can also be written as

$$\begin{cases} \varepsilon+m_2^2+(P_k+Q_k)\cdot m_2+P_kQ_k>0, & k\in S_p, \\ \varepsilon+m_2^2+(P_k+Q_k)\cdot m_2+P_kQ_k<0, & k\in S_n, \\ P_k\cdot m_2+Q_k>0, & k\in S_z. \end{cases} \tag{16}$$

Then by substituting $m_2=y_a$, an appropriate ε can be found such that Eq. (16) is satisfied. Now, find $\boldsymbol{M}_{k+1}$ for the corresponding Brunovsky coordinates by

$$\boldsymbol{M}_{k+1}=\boldsymbol{T}_k^{\mathrm{T}}\boldsymbol{M}_1\boldsymbol{T}_k. \tag{17}$$

By letting $(\tilde{\boldsymbol{A}}_i, \tilde{\boldsymbol{b}}_i)$ represent system (3) in the Brunovsky form given by $\tilde{\boldsymbol{A}}_i=\boldsymbol{C}_i\boldsymbol{A}_i\boldsymbol{C}_i^{-1}$, $\tilde{\boldsymbol{b}}_i=[0\ 1]^{\mathrm{T}}$, i=1, 2, …, N, each subsystem takes the form of

$$\dot{\boldsymbol{x}}=\begin{bmatrix} 0 & 1 \\ \tilde{A}_i[2,1] & \tilde{A}_i[2,2] \end{bmatrix}\boldsymbol{x}+\begin{bmatrix} 0 \\ 1 \end{bmatrix}u. \tag{18}$$

Controller gains in Brunovsky form can be found by

$$u = \boldsymbol{k}_i \boldsymbol{x} = \left[-1 - \tilde{A}_i[2,\ 1] \quad \frac{-1 \cdot M_i[2,\ 2] - M_i[1,\ 1]}{M_i[1,\ 2]} - \tilde{A}_i[2,\ 2] \right] \boldsymbol{x}. \tag{19}$$

Lastly, transform each state feedback controller back to the original coordinates by

$$\boldsymbol{K}_i = \boldsymbol{k}_i \boldsymbol{C}_i, \quad i=1, 2, \ldots, N. \tag{20}$$

Then

$$\boldsymbol{M}_0 = \boldsymbol{C}_1^{\mathrm{T}} \boldsymbol{M}_1 \boldsymbol{C}_1 \tag{21}$$

is the Lyapunov matrix defining the common quadratic Lyapunov function in the original coordinates. Stability of the overall system can easily be verified by the quadratic Lyapunov function:

$$\boldsymbol{M}_0(\boldsymbol{A}_i + \boldsymbol{B}_i \boldsymbol{K}_i) + (\boldsymbol{A}_i + \boldsymbol{B}_i \boldsymbol{K}_i)^{\mathrm{T}} \boldsymbol{M}_0 < \boldsymbol{0}. \tag{22}$$

The overall system is quadratically stable if inequality (22) is true for all i=1, 2, …, N.

3 Computational ability test

Consider the case of a continuous time switched system X as in Eq. (3) made up of four subsystems as follows:

$$\boldsymbol{A}_1 = \begin{bmatrix} -4 & -2 \\ 9 & 5 \end{bmatrix}, \ \boldsymbol{B}_1 = \begin{bmatrix} -1 \\ 2 \end{bmatrix},$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 2 & -1/3 \\ -3 & 0 \end{bmatrix}, \ \boldsymbol{B}_2 = \begin{bmatrix} 5 \\ -6 \end{bmatrix},$$

$$\boldsymbol{A}_3 = \begin{bmatrix} -7 & -6 \\ 19/2 & 8 \end{bmatrix}, \ \boldsymbol{B}_3 = \begin{bmatrix} 6 \\ -7 \end{bmatrix},$$

$$\boldsymbol{A}_4 = \begin{bmatrix} -12 & -11 \\ 13 & 12 \end{bmatrix}, \ \boldsymbol{B}_4 = \begin{bmatrix} -7 \\ 8 \end{bmatrix}.$$

This system was used as our benchmark case in comparing the computational ability of different solvers in solving the two sets of LIs (13) and (16) and in executing the whole algorithm.

Three numerical computation tools, LMI solver, cvx, and Yalmip, as well as the symbolic computation software Maple, were tested for their computational ability in executing the Haris-Rogers method.

3.1 LMI solver

The LMI solver comes with the LMI Lab Package as part of the Robust Control Toolbox in Matlab. It provides three solvers whose respective functions are to check for the feasibility of the problem, to minimize a linear objective under LMI constraints and to solve generalized eigenvalue minimization problems.

After all variables and LMIs have been adequately structured, the appropriate function is called to solve the problem. However, LMI Editor is not designed to solve linear inequalities. Therefore, the mixed set of LIs (13) had to be appropriately modified. As for LIs (16), the LMI Editor graphical user interface (GUI) was used and the function feasp was called to check for feasibility.

The two sets of LIs were defined as

```
for k=1:N-2
  if B(k,1)>0 && B(k+1,1)<0
    f3=find((p(k,1)+q(k,1)-p(k+1,1)-q(k+1,1))*x
      +(p(k,1)*q(k,1))-(p(k+1,1)*q(k+1,1))>0);
  else if B(k+1,1)>0 && B(k,1)<0
    f3=find((p(k,1)+q(k,1)-p(k+1,1)-q(k+1,1))*x
      +(p(k,1)*q(k,1))-(p(k+1,1)*q(k+1,1))>0);
  end
end
for k=1:N-1
  if B(k,1)==0
    f4=find(p(k,1)*x+q(k,1))>0;
  end
end
setlmis([]);
e=lmivar(2,[1 1]);
for k=1:N-1
  if B(k,1)>0
    lmiterm([-1 1 1 e],1,1);
    lmiterm([-1 1 1 0],(X^2)+((p(k,1)+q(k,1))*X)
      +(p(k,1)*q(k,1)));
  else if B(k,1)<0
    lmiterm([2 1 1 e],1,1);
    lmiterm([2 1 1 0],(X^2)+((p(k,1)+q(k,1))*X)
      +(p(k,1)*q(k,1)));
  end
end
lmiterm([-4 1 1 e],1,1);
gete=getlmis;
[tmin, vector]=feasp(gete);
e=dec2mat(gete,vector,e);
```

The matrix obtained was $\boldsymbol{M}_0 = \begin{bmatrix} 32.0 & 26.5 \\ 26.5 & 22.0 \end{bmatrix}$.

The state feedback controllers for system *X* found using the LMI solver were previously obtained by Haris and Rogers (2008) and are shown in Table 1 for brevity.

Table 1 State feedback controller gains using the LMI solver (Haris and Rogers, 2008)

Subsystem No.	State feedback controller, $\boldsymbol{K}_i$
1	[−11.29 −8.29]
2	[−18 −11.33]
3	[−28 −21]
4	[28 22]

3.2 cvx

cvx is a modelling system designed for disciplined convex programming (DCP). It contains rules called the DCP ruleset, developed from the fundamentals of convex analysis. cvx is based on Matlab fundamental operators and can be used for linear programming (LP), nonlinear programming (NLP), and quadratic programming (QP).

To solve Eqs. (13) and (16) for system *X*, the problem was described as follows:

```
cvx_begin
  variable Xn;
  variable Yn;
  minimize Xn
  subject to
  for k=1:N-1
    if B(k,1)==0
      p(k,1)*Xn+q(k,1)>=Yn;
    else if B(k,:)>=0
      (Xn+p(k,1))'*p(k,1)*(Xn+q(k,1))>=Yn;
    else if B(k,1)<=0
      Xn^2+(p(k,1)+q(k,1))*Xn+p(k,1)*q(k,1)<=Yn;
    end
  end
cvx_end

cvx_begin
  variable Xx;
  variable Yx;
  maximize Xx
  subject to
  for k=1:N-1
    if B(k,1)==0
      p(k,1)*Xx+q(k,1)>=Yx;
    else if B(k,:)>=0
      (Xx+p(k,1))'*p(k,1)*(Xx+q(k,1))>=Yx;
    else if B(k,1)<=0
      Xx^2+(p(k,1)+q(k,1))*Xx+p(k,1)*q(k,1)<=Yx;
    end
  end
cvx_end

M2=(Xx+Xn)/2;

cvx_begin
  variable en;
  variable Yen;
  minimize en
  subject to
  for k=1:N-1
    if B(k,1)==0
      en+p(k,1)*M2+q(k,1)>=Yen;
    else if B(k,:)>=0
      en+M2^2+(p(k,1)+q(k,1))*M2+p(k,1)*q(k,1)>=Yen;
    else if B(k,1)<=0
      en+M2^2+(p(k,1)+q(k,1))*M2+p(k,1)*q(k,1)<=Yen;
    end
  end
  en>=0;
  en<=10;
  Yen==0;
cvx_end

cvx_begin
  variable ex;
  variable Yex;
  maximize ex
  subject to
  for k=1:N-1
    if B(k,1)==0
      ex+p(k,1)*M2+q(k,1)>=Yex;
    else if B(k,:)>=0
      ex+M2^2+(p(k,1)+q(k,1))*M2+p(k,1)*q(k,1)>=Yex;
    else if B(k,1)<=0
      ex+M2^2+(p(k,1)+q(k,1))*M2+p(k,1)*q(k,1)<=Yex;
    end
  end
  ex>=0;
  ex<=10;
  Yex==0;
cvx_end

e=(ex+en)/2;
```

Referring to the DCP rule as set by Grant and Boyd (2012), for each convex equation, each convex scalar should start from a power $p\geq1$ and $p\neq3, 5, 7, 9, \ldots$. Fundamentally, the Haris-Rogers method is met with a constraint for its concave equation, where each concave scalar should be only of a power $p\in(0, 1)$. This means that only linear concave equations are allowed. To overcome this problem, the quadratic concave equation was modified into the form

$$(x+p(k,1))'Q(x+q(k,1))\geq 0,$$

where x is the parameter to be found and Q is a positive constant such that the equation is concave without changing the validity of the algorithm. The matrix obtained was

$$\boldsymbol{M}_0=\begin{bmatrix}33.368437298790745 & 27.754400857224852\\ 27.754400857224852 & 23.140364415658958\end{bmatrix},$$

and the resulting state feedback controllers were as follows:

$\boldsymbol{K}_1$=[−11.402913968856573 −8.402913968856574],
$\boldsymbol{K}_2$=[−10.687448453596975 −6.458298969064620],
$\boldsymbol{K}_3$=[−38.636527223007754 −28.977395417255817],
$\boldsymbol{K}_4$=[36.863772685840530 29.091018148672386].

3.3 Yalmip

'Yet Another LMI Parser', or Yalmip, is a modelling language designed with the help of external solvers such as SeDuMi, BNB, and CUTSDP, to solve convex and nonconvex optimization problems. It was developed as a Matlab toolbox. To solve any particular problem, Yalmip suggests and chooses the best solver according to the needs of the problem. However, users may still choose a different solver if they wish.

SeDuMi, which is an external solver for semidefinite programming, can solve second-order conic inequalities and the linear and quadratic problems from Eq. (13), posed as follows:

```
X=intvar(1,1);
F=[X>0];
F=[F,0<X<10];
F=[F,(((X^2+(p(1,1)+q(1,1))*X+p(1,1)*q(1,1))<=0)),
    (((X^2+(p(2,1)+q(2,1))*X+p(2,1)*q(2,1))<=0)),
    (((X^2+(p(3,1)+q(3,1))*X+p(3,1)*q(3,1))<=0))];
MN=solvesdp(F,X);
Xn=double(X);
checkset(F);

X=intvar(1,1);
G=[X>0];
G=[G,0<X<10];
G=[G,(((X^2+(p(1,1)+q(1,1))*X+p(1,1)*q(1,1))<=0)),
    (((X^2+(p(2,1)+q(2,1))*X+p(2,1)*q(2,1))<=0)),
    (((X^2+(p(3,1)+q(3,1))*X+p(3,1)*q(3,1))<=0))];
MX=solvesdp(G,-X);
Xx=double(X);
checkset(G);

M2=(Xx+Xn)/2;
```

The external solver LINPROG and internal solver bnb were used to solve LI (16):

```
eN=intvar(1,1);
H=[eN+M2^2+(p(1,1)+q(1,1))*M2+p(1,1)*q(1,1)<0,
    eN+M2^2+(p(2,1)+q(2,1))*M2+p(2,1)*q(2,1)<0,
    eN+M2^2+(p(3,1)+q(3,1))*M2+p(3,1)*q(3,1)<0];
obj=eN;
EN=solvesdp(H,-obj);
en=double(eN);
checkset(H);

eX=intvar(1,1);
J=[eX+M2^2+(p(1,1)+q(1,1))*M2+p(1,1)*q(1,1)>0,
    eX+M2^2+(p(2,1)+q(2,1))*M2+p(2,1)*q(2,1)<0,
    eX+M2^2+(p(3,1)+q(3,1))*M2+p(3,1)*q(3,1)>0];
obje=eX;
EX=solvesdp(J,-obje);
ex=double(eX);
checkset(J);
e=(ex+en)/2;
```

The results gave

$$\boldsymbol{M}_0=\begin{bmatrix}31.750000000000000 & 26.250000000000000\\ 26.250000000000000 & 21.750000000000000\end{bmatrix}$$

and the state feedback controllers as follows:

$\boldsymbol{K}_1$=[−11.214285714285715 −8.214285714285715],
$\boldsymbol{K}_2$=[−36.000000000002139 −23.333333333334668],
$\boldsymbol{K}_3$=[−18.000000000000043 −13.500000000000030],
$\boldsymbol{K}_4$=[58.000000000014730 46.000000000011724].

3.4 Maple

Maple was used in an attempt to solve the problem by symbolic computation. LIs (13) and (16) were solved as follows:

```
for n to k-1 do
  if Z[n]>0 then
    RR[n]:=xx^2+(P[n]+Q[n])*xx+P[n]*Q[n]>=yy;
  else if Z[n]<0 then
    RR[n]:=xx^2+(P[n]+Q[n])*xx+P[n]*Q[n]<=yy;
  else if Z[n]=0 then
    RR[n]:=PSz[n]*xx+QSz[n]>=yy;
  end
end
```

```
elow:=NLPSolve(obj, cnsts[k-1], xx=0, 1, ..., 10);

for n to k-1 do
   if Z[n]>0 then
      e1[n]:=e+vx^2+(P[n]+Q[n])*vx+P[n]*Q[n]>=vy;
   else if Z[n]<0 then
      e1[n]:=e+vx^2+(P[n]+Q[n])*vx+P[n]*Q[n]<=vy;
   else if Z[n]=0 then
      e1[n]:=e+PSz[n]*vx+QSz[n]>=0;
   end
end

low:=LPSolve(eobj, ecnsts[k-1], e=0, 1, ..., 10,
        vy=0, 1, ..., 10);
high:=LPSolve(eobj, ecnsts[k-1], e=0, 1, ..., 10,
        vy=0, 1, ..., 10, maximize);
```

The LIs were solved in 3.54 s using 42.67 MB of computer memory. The results gave

$$\boldsymbol{M}_0=\begin{bmatrix}31.9999977359999974 & 26.4999979560000014\\ 26.4999979629999984 & 21.9999981829999988\end{bmatrix}$$

and the state feedback controllers as follows:

$\boldsymbol{K}_1$=[−11.2857141799999994 −8.2857141749999990],
$\boldsymbol{K}_2$=[−18.0000080300000000 −11.3333386899999997],
$\boldsymbol{K}_3$=[−28.0000113100000014 −21.0000084800000018],
$\boldsymbol{K}_4$=[27.9999591400000014 21.9999673000000016].

4 Simulation results

To illustrate the effectiveness of the control design method, simulations were carried out using the test switching system under the influence of the controllers designed using Maple. The simulated system was set to switch in sequence at a time interval of 300 s. Fig. 1 shows the step response simulation results.

While stability is clearly evident, the performance of the system was not maintained in the simulation. Steady state outputs of each subsystem varied significantly; i.e., the value changed from 1 when subsystem 1 was in operation, to 2 during subsystem 2, −6 during subsystem 3, and −4 during subsystem 4. Also note that spikes occurred in the response at some switching instances (e.g., from subsystems 1 to 2 and from subsystems 3 to 4). Such behavior is not really surprising as the current method emphasizes only stability and no attempt was made to design the controllers to meet performance requirements.

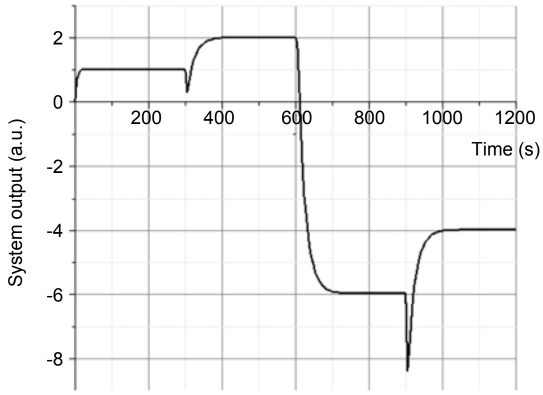

Fig. 1 Transient response of a switched system solved using Maple for switching every 300 s time interval

5 Discussion

The Haris-Rogers method was tested in terms of its level of consumption of computational demands or powers in solving two sets of LIs for the switched hybrid system *X*. Three types of numerical computation based programs, LMI solver, cvx, and Yalmip, were used alongside one type of symbolic computation based program, Maple. The results show that the two sets of LIs from the Haris-Rogers method can be solved to obtain the feedback multi-controller for the switched system *X* using all the programs. Table 2 shows the Lyapunov matrix and eigenvalues of the feedback system matrices of the switched system *X* obtained using the Haris-Rogers method.

The Lyapunov matrix and the eigenvalues for each of the four subsystems show slight differences among the four program types (Table 2). The eigenvalues up to the 15th decimal place are required to guarantee the stability of the whole system. The decimal precision for each calculation is important in determining the stability of switched systems. Any changes in value may transform the eigenvalue of the quadratic Lyapunov function (6) from negative to positive, which nullifies the required negative definite property, and thus will not guarantee the stability of the switched system. Therefore, the selection of the type of numerical data is important for the data calculation. Different data types will provide different memory capacities for the analysis. In this research, data type 'long' was used instead of 'short',

Table 2 Lyapunov matrix and eigenvalues for the switched system *X* for different types of computing programs

Type of computing program	Lyapunov matrix and eigenvalue of each subsystem	Stability of system
cvx	$\boldsymbol{M}_0=\begin{bmatrix}33.368437298790745 & 27.754400857224852\\27.754400857224852 & 23.140364415658958\end{bmatrix}$ Subsystem 1: $10^3\times\begin{bmatrix}-0.463245880674346\\-0.000015983150277\end{bmatrix}$ Subsystem 2: $10^3\times\begin{bmatrix}-0.057269582773575\\-0.000129285532872\end{bmatrix}$ Subsystem 3: $10^3\times\begin{bmatrix}-0.624172878035841\\-0.000011862304157\end{bmatrix}$ Subsystem 4: $10^3\times\begin{bmatrix}-1.518376033878618\\-0.000004876347073\end{bmatrix}$	Guaranteed
LMI solver	$\boldsymbol{M}_0=\begin{bmatrix}32.000000000000000 & 26.500000000000000\\26.500000000000000 & 22.000000000000000\end{bmatrix}$ Subsystem 1: $10^3\times\begin{bmatrix}-0.429313695430464\\-0.000016304569536\end{bmatrix}$ Subsystem 2: $10^3\times\begin{bmatrix}-0.095922961802304\\-0.000073704864362\end{bmatrix}$ Subsystem 3: $10^3\times\begin{bmatrix}-0.484485551684719\\-0.000014448315281\end{bmatrix}$ Subsystem 4: $10^3\times\begin{bmatrix}-1.223994281019031\\-0.000005718980970\end{bmatrix}$	Guaranteed
Yalmip (SeDuMi-1.3, LINPROG)	$\boldsymbol{M}_0=\begin{bmatrix}31.750000000000000 & 26.250000000000000\\26.250000000000000 & 21.750000000000000\end{bmatrix}$ Subsystem 1: $10^3\times\begin{bmatrix}-0.417771352362701\\-0.000014361923014\end{bmatrix}$ Subsystem 2: $10^3\times\begin{bmatrix}-0.172965310963419\\-0.000034689036586\end{bmatrix}$ Subsystem 3: $10^3\times\begin{bmatrix}-0.297479830565360\\-0.000020169434543\end{bmatrix}$ Subsystem 4: $10^3\times\begin{bmatrix}-2.452997554013643\\-0.000002445986950\end{bmatrix}$	Guaranteed
Maple	$\boldsymbol{M}_0=\begin{bmatrix}31.999997735999997 & 26.499997956000001\\26.499997962999998 & 21.999998182999999\end{bmatrix}$ Subsystem 1: $10^3\times\begin{bmatrix}-0.428983635250474\\-0.000016317640675\end{bmatrix}$ Subsystem 2: $10^3\times\begin{bmatrix}-0.095927093860799\\-0.000072972091891\end{bmatrix}$ Subsystem 3: $10^3\times\begin{bmatrix}-0.484485854345949\\-0.000014448307572\end{bmatrix}$ Subsystem 4: $10^3\times\begin{bmatrix}-1.223992714270676\\-0.000005718989269\end{bmatrix}$	Guaranteed

which is the default data type of numerical computation based software, Matlab. The memory capacity of 'long' is larger, with a capacity of four bytes with 15 decimal numbers compared to 'short' with two bytes and only four decimal numbers. This larger capacity is used to prevent any loss of information, such as the loss of decimal numbers, due to rounding-off errors that bring variations in the results of stability analysis of switched systems. Table 3 shows examples of such rounding-off errors. The Lyapunov matrix obtained using cvx was tested for negative definiteness when rounded to four decimal places and also when rounded to one decimal place. The eigenvalues were found to be a mix of negative and positive values and also zero, and thus stability is not guaranteed.

In this research, the external solver of semidefinite programming type, named SeDuMi-1.3, was chosen to solve Eq. (13) using Yalmip. Nevertheless, there are many other external solvers that could be used, such as LMILAB, SDPT3, SDPA, and DSDP. Yet, it is difficult to identify which external solver is the most suitable as no studies have identified the external solver with the least rounding-off errors. Moreover, the final solutions usually differ slightly every time a new update of SeDuMi is installed.

When LMILAB was used to solve Eq. (13) and LINPROG for Eq. (16), the state feedback controller gains obtained were as follows:

$\boldsymbol{K}_1$=[−11.500000000000000 −8.500000000000000],
$\boldsymbol{K}_2$=[−8.999999999999968 −5.333333333333286],
$\boldsymbol{K}_3$=[−16.000000000000028 −12.000000000000020],
$\boldsymbol{K}_4$=[−16.999999999999442 −13.999999999999545],

while the Lyapunov matrix and the eigenvalues of the quadratic Lyapunov functions are shown in Table 4.

The Lyapunov matrix obtained, rounded to 15 and four decimal numbers, are different from those obtained using SeDuMi-1.3. As a result, the solutions are not negative definite, and the stability of the switched system cannot be ascertained.

In contrast, when symbolic computation or computer algebra system (CAS) based software is used, any variables in the algebraic equation will remain in variable form, where every step of the calculations will be simplified in algebraic form. For example, when the input of the symbolic calculation is $y=3x-x+1$, the output will be $y=2x+1$. The numerical result will be revealed when a value is assigned to the variable. Furthermore, symbolic computation deals with precise values such as a fraction number

Table 3 Rounding off the decimal of the Lyapunov matrix and the eigenvalues for a switched system X using cvx

	Lyapunov matrix	Eigenvalue for each subsystem	Stability of system
Rounded to four decimals	$\boldsymbol{M}_0=\begin{bmatrix}33.3684 & 27.7544\\27.7544 & 23.1403\end{bmatrix}$	Subsystem 1: $10^3\times\begin{bmatrix}-0.4632\\0\end{bmatrix}$ Subsystem 2: $10^3\times\begin{bmatrix}-0.0573\\-0.0001\end{bmatrix}$ Subsystem 3: $10^3\times\begin{bmatrix}-0.6242\\0\end{bmatrix}$ Subsystem 4: $10^3\times\begin{bmatrix}-1.5183\\0\end{bmatrix}$	Not guaranteed
Rounded to one decimal	$\boldsymbol{M}_0=\begin{bmatrix}33.4 & 27.8\\27.8 & 23.1\end{bmatrix}$	Subsystem 1: $10^3\times\begin{bmatrix}-0.4625\\0\end{bmatrix}$ Subsystem 2: $10^3\times\begin{bmatrix}-0.0618\\0.0006\end{bmatrix}$ Subsystem 3: $10^3\times\begin{bmatrix}-0.6242\\0.0011\end{bmatrix}$ Subsystem 4: $10^3\times\begin{bmatrix}-1.5183\\0.0006\end{bmatrix}$	Not guaranteed

Table 4 Rounding off the decimal of the Lyapunov matrix and the eigenvalues for a switched system X using Yalmip (LMILAB and LINPROG)

	Lyapunov matrix and eigenvalue for each subsystem	Stability of system
Rounded to 15 decimals	$\boldsymbol{M}_0=\begin{bmatrix}37.000000000000000 & 31.000000000000000\\ 31.000000000000000 & 26.000000000000000\end{bmatrix}$ Subsystem 1: $10^2\times\begin{bmatrix}-5.339925092582186\\ -0.000074907417813\end{bmatrix}$ Subsystem 2: $10^2\times\begin{bmatrix}-0.298660687473191\\ -0.001339312526813\end{bmatrix}$ Subsystem 3: $10^2\times\begin{bmatrix}-1.409716254956561\\ -0.000283745043440\end{bmatrix}$ Subsystem 4: $10^2\times\begin{bmatrix}0.000082305920629\\ 4.859917694079162\end{bmatrix}$	Not guaranteed
Rounded to four decimals	$\boldsymbol{M}_0=\begin{bmatrix}37.0000 & 31.0000\\ 31.0000 & 26.0000\end{bmatrix}$ Subsystem 1: $\begin{bmatrix}-533.9925\\ -0.0075\end{bmatrix}$; Subsystem 2: $\begin{bmatrix}-29.8661\\ -0.1339\end{bmatrix}$; Subsystem 3: $10^2\times\begin{bmatrix}-140.9716\\ -0.0284\end{bmatrix}$; Subsystem 4: $\begin{bmatrix}0.0082\\ 485.9918\end{bmatrix}$.	Not guaranteed

1/3 and π instead of 0.333 and 3.142 used in numerical computation based software. The use of this kind of software is essential in stability research where every decimal place is significant. A computer algebra processor uses a hardware floating-point environment to handle the decimals, which might cause cramming in the computerized process while doing calculations. This processor can better cope with the computational burden and will increase the efficiency of calculation. Therefore, the transition from using numerical computation based software to using symbolic computation based software is needed to gain the exact solution that guarantees the stability of the switched system.

After dealing with numerical computation based software to test the Haris-Rogers method, the same test should be done using symbolic computation based software (such as Mathematica, Mathcad, or MuPAD) which is capable of solving the algorithm and LMIs. The software must be able to find solutions for switched systems consisting of many subsystems and higher-order systems. Nevertheless, an accurate solution can be defined using any computational software if the algorithm consists of simple equations such as LIs.

6 Conclusions and future work

A study into the computational ability of the Haris-Rogers method in establishing controllers for switched systems has been presented where stability is assured by the existence of a CLF. While most methods for CLF based controller design for switched systems pose the problem as a set of LMIs, the Haris-Rogers method reduces it to solving two sets of linear inequalities (LIs), thus significantly reducing the computational burden.

Three types of numerical computation and one symbolic computation program were tested to solve the two LIs (13) and (16). The results showed that the LIs were successfully solved by all tested computational tools. The CLFs (22) obtained by the solvers could easily be verified to be negative definite, proving that the controlled switched system is stable for any switching sequence and dwell time. The issue of rounding-off error in numerical computation based software was discussed in detail. The tests revealed that the guarantee of stability became uncertain when the rounding off had different levels of decimal precision. The use of different external solvers led to the same conclusion in relation to the stability of switched systems.

As a result, we suggest a shift from conventional programs to computer algebra. This type of program allows the exact solution for the entire calculation. The processor, using a hardware floating-point environment, reduces the cramming in the computing process. This type of computation is vital in this area of research in that every decimal place affects the range of the allowed controller parameters that guarantee the stability of the switched system. By using Maple, the computation was successfully executed using almost 43 MB of memory.

Future research should look into the possibility of incorporating performance optimization tools such as pole placement, root locus, and LQR/LQG into the method without jeopardizing system stability. Extending the method to higher-order systems is another direction for future research.

We have also demonstrated that the method could successfully be implemented on symbolic computation software. This opens up the possibility of using methods from computer algebra, such as Grobner Bases, Quantifier Elimination, and the Sum of Roots, to reduce the high number of parameters to smaller more computable inequality sets.

References

Davrazos, G., Koussoulas, N.T., 2002. A General Methodology for Stability Analysis of Differential Petri Nets. 10th Mediterranean Conf. on Control and Automation, p.296-302.

Decarlo, R.A., Branicky, M.S., Pettersson, S., Lennartson, B., 2000. Perspectives and results on the stability and stabilizability of hybrid systems. *Proc. IEEE*, **88**(7):1069-1082. [doi:10.1109/5.871309]

Geng, Z., 2010. Switched Stability Design on Canonical Forms. Int. Conf. on Information and Automation, p.289-293.

Grant, M., Boyd, S., 2012. Cvx Users' Guide for Cvx Version 1.21.

Haris, S.M., 2006. Analysis and Design of Classes of Hybrid Control Systems. PhD Thesis, University of Southampton, UK.

Haris, S.M., Rogers, E., 2008. A Matlab Toolbox for Finding Stabilizing Controllers for a Class of Switched Systems. Int. Conf. on Computational Intelligence for Modelling Control & Automation, p.238-242. [doi:10.1109/CIMCA.2008.115]

Haris, S.M., Saad, M.H.M., Rogers, E., 2007. A Method for Determining Stabilizability of a Class of Switched System. 7th WSEAS Int. Conf. on Systems Theory and Scientific Computation, p.27-32.

Hespanha, J.P., Morse, A.S., 2002. Switching between stabilizing controllers. *Automatica*, **38**(11):1905-1917. [doi:10.1016/S0005-1098(02)00139-5]

King, C., Shorten, R., 2006. Singularity conditions for the non-existence of a common quadratic Lyapunov function for pairs of third order linear time invariant dynamic systems. *Linear Algebra Its Appl.*, **413**(1):24-35. [doi:10.1016/j.laa.2005.07.023]

Liberzon, D., Hespanha, J.P., Morse, A.S., 1999. Stability of switched systems: a Lie-algebraic condition. *Syst. Control Lett.*, **37**(3):117-122. [doi:10.1016/S0167-6911(99)00012-2]

Mason, O., Shorten, R., 2003. A Conjecture on the Existence of Common Quadratic Lyapunov Functions for Positive Linear Systems. American Control Conf., p.4469-4470.

Montagner, V.F., Leite, V.J.S., Oliveira, R.C.L.F., Peres, P.L.D., 2006. State feedback control of switched linear systems: an LMI approach. *J. Comput. Appl. Math.*, **194**(2):192-206. [doi:10.1016/j.cam.2005.07.005]

Stewart, G.E., Dumont, G.A., 2006. Finite Horizon Based Switching Between Stabilizing Controllers. American Control Conf., p.1550-1556.

Zhai, G., Liu, D., Imae, J., Kobayashi, T., 2006. Lie algebraic stability analysis for switched systems with continuous-time and discrete-time subsystems. *IEEE Trans. Circ. Syst. II*, **53**(2):152-156. [doi:10.1109/TCSII.2005.856033]

Zhang, W., Shen, S.Q., Han, Z.Z., 2008. Sufficient conditions for Hurwitz stability of matrices. *Latin Am. Appl. Res.*, **38**:253-258.

Zhu, Y.H., Cheng, D.Z., Qin, H.S., 2007. Constructing common quadratic Lyapunov functions for a class of stable matrices. *Acta Autom. Sin.*, **33**(2):202-204. [doi:10.1360/aas-007-0202]