International Journal of Control
Vol. 00, No. 00, Month 200x, 1–15

RESEARCH ARTICLE

On the Stability and Control of Discrete Linear Systems with Clock Synchronization Errors

Marek Przedwojski[a], Krzysztof Galkowski[b] , Peter Bauer[c], and Eric Rogers[a]

[a]*School of Electronics and Computer Science, University of Southampton, UK*; [b]*Institute of Control and Computation Engineering, University of Zielona Gora, Poland*; [c]*Department of Electrical Engineering, University of Notre Dame, USA*

(*v3.5 released August 2008*)

This paper considers discrete linear time-invariant systems that can be decomposed into subsystems whose states are synchronized by a common clock whose signal reaches them with delays. In particular, stability for the case where all subsystems have the same sampling frequency, but different switching times, is investigated. In contrast to previous work, the approach taken here models the set of system matrices that arise using a polytopic uncertainty approach, which has seen extensive application in robust control theory for linear systems. Stabilization is then achieved by state feedback and a method to handle the combinatorial explosion of the number of polytope vertices is developed and illustrated using an example from swarm system navigation.

Keywords: synchronization errors, stability, control, detection.

1 Introduction

Classical digital signal processing and control system analysis techniques are based on the fundamental fact that all components of the overall system switch at exactly the same time instant. In particular, it is assumed that the system is driven by a single clock and that the differences in clock propagation time between the different system components are negligible. Over the last decade, many applications have emerged that routinely violate this basic assumption. Typical examples include wireless sensor networks, vehicle networks and swarms, tele-operation-systems, and distributed actuator systems.

Synchronization of different system components to a degree that would allow them to be treated as a synchronous system is either impossible or very expensive. Consequently research on the modeling, analysis and design of asynchronous systems is of critical importance with many applications awaiting applicable results and algorithms.

The problems arising from the introduction of non-ideal characteristics into applications such as those mentioned above can be grouped, at a general level, into time-variant communication delays caused by, for example, packet drop, queuing, access delay, and synchronization errors respectively. Moreover, the vast majority of the research has been in the former area and in Lorand and Bauer (2005) it is suggested that this is due to a focus on applications supported by wide area networks, where communication delays are commonly expected to be more critical than synchronization errors. In a local area network, however, the average communication delays can be made small enough and then synchronization errors become the critical issue.

This paper exclusively deals with synchronization errors, building on previous work Lorand and Bauer (2005, 2006a,b), Kleptzyn et al. (1984), Su et al. (1997), Lorand (2004) on modeling

*Corresponding author. Email: k.galkowski@issi.uz.zgora.pl

ISSN: 0020-7179 print/ISSN 1366-5820 online
© 200x Taylor & Francis
DOI: 10.1080/0020717YYxxxxxxxx
http://www.informaworld.com

and stability analysis. In particular, Lorand and Bauer (2005) showed that linear synchronous systems are subject to errors that result in asynchronous operation and moreover that this can be caused by even small errors. This previous work also demonstrated that classical design techniques cannot be directly applied to systems that operate asynchronously.

It would clearly be very beneficial to have a characterization of stability for these systems that leads on to control law design, where in the general subject area there is a very extensive literature on stability see, for example, Gazi (2008), Li (2008) but less on stability plus control law design. This paper considers the case when the overall system is composed of subsystems driven by the same clock but the switching times are different between them. This scenario describes, for example, high-speed circuits and low-speed networks that periodically re-synchronize. Even though high speed circuits have been designed to function as a synchronous system Thierrauf (2004), extremely high clock speed could result in propagation delay differences of the order of a period of the system clock.

The results in this paper can be grouped into two main areas, the first of which is stabilization of an asynchronous network of systems through state feedback. Here we model the set of system matrices that arise using a polytope approach where each vertex corresponds to a possible state matrix of the overall system. This model enables the design of stabilizing control laws using Linear Matrix Inequalities (LMIs), which is illustrated by an example from swarm navigation. The major point to note here that the novel contribution lies in the application of a standard tool in robust control theory to the problem of systems with synchronization errors rather than in robust control theory itself. A method is also developed for dealing with the computational load which arises.

The notation of this paper uses $\Gamma \succ 0$ and $\Gamma \prec 0$ to denote symmetric matrices that are positive and negative definite respectively.

2 Background

The starting point for the work reported here is the discrete linear time-invariant state-space model with, in particular, state vector x and control input vector u. Asynchronous switching in linear time-invariant systems is discussed in, for example, Lorand and Bauer (2006b), Kleptzyn et al. (1984) and this paper begins from the problem setup considered in this previous work. We consider the case where every state vector entry x_i is fed by a clock with rate T_i, $i = 1, 2, \ldots, n$. The clock rates are equal, that is, $T_i = T$, $i = 1, 2, \ldots, n$, but the associated signals are out of phase, and these effects are termed synchronization errors. In order to capture effects of asynchronous switching, an event driven time index k is introduced which following Lorand and Bauer (2006b), is increased by an integer equal to the number of variables that switch simultaneously. Consequently over a full clock period the time index is incremented by n, but there is no information about the order in which these variables have been updated. Also the system output samples can only be read periodically at the sampling times. We also assume that the input is updated periodically in similar way as the subsystems, but without increasing the time index.

Suppose that there are d switching events in one full clock period, where these are described by the sequence s of mutually disjoint subsets of indices

$$s = (i_1, i_2, \ldots, i_d), \quad i_j \subseteq \{1, \ldots, n\}, \quad j = 1, \ldots, d. \tag{1}$$

These subsets satisfy

$$p \neq r \;\Rightarrow\; i_p \cap i_r = \emptyset \qquad p, r = 1, \ldots, d, \tag{2}$$

and

$$\bigcup_{j=1}^{d} i_j = \{1, \ldots, n\}. \tag{3}$$

The number of elements in each subset i_j

$$h_j = \mathrm{card}(i_j), \tag{4}$$

is the number of entries that switch simultaneously during the event j and it is straightforward to see that

$$h_1 + \cdots + h_d = n. \tag{5}$$

We use $\mathcal{S}$ to denote the set of all possible switching events

The state equation for a single event numbered j can now be written as

$$\begin{aligned}\mathbf{x}(ld + h_1 + \cdots + h_j) = {} & A_{i_j}\mathbf{x}(ld + h_1 + \cdots + h_{j-1}) \\ & + B_{i_j}\mathbf{u}(ld + h_1 + \cdots + h_{j-1}),\end{aligned} \tag{6}$$

where $l = 0, 1, 2, \ldots$, the model matrix $A_{i_j} \in \mathcal{R}^{n\times n}$ given, for example, $i_j = \{\ldots, p, \ldots, q, \ldots\}$, is

$$A_{i_j} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{p1} & a_{p2} & \ldots & a_{p(n-1)} & a_{pn} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{q1} & a_{q2} & \ldots & a_{q(n-1)} & a_{qn} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix}, \tag{7}$$

and the corresponding input matrix $B_{i_j} \in \mathcal{R}^{n\times l}$ is

$$B_{i_j} = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ b_{p1} & b_{p2} & \ldots & b_{p(l-1)} & b_{pl} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ b_{q1} & b_{q2} & \ldots & b_{q(l-1)} & b_{ql} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 \end{bmatrix}. \tag{8}$$

Here A_{i_j} contains only those rows from the state matrix A that are relevant to simultaneously switched variables (at least one), and all remaining rows are taken from the identity matrix. The matrix B is constructed in a similar way where all rows except p and q have only zero entries. In general, after a clock period, all d entries have switched and consequently the index l is incremented by n over a full clock period.

Example 2.1 Consider the case of zero inputs, $n = 2$, and state matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \tag{9}$$

with an event pattern which can be represented as in Fig. 1, where □ and •, respectively, are used to denote that the first and second entries in the state vector have switched.

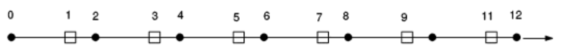

Figure 1. $T_1 = T_2$ with clock signals out of phase.

The sequence s that describes event pattern for this example is

$$s = (\{1\}, \{2\}) \tag{10}$$

Event 1 □: $i_1 = \{1\} \quad h_1 = 1$

$$\begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix},$$

Event 2 •: $i_2 = \{2\} \quad h_2 = 1$

$$\begin{bmatrix} x_1(2) \\ x_2(2) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix},$$

and a full new state vector has been created as the switching event pattern is periodic. Also by back-substitution we have

$$\begin{bmatrix} x_1(2l+2) \\ x_2(2l+2) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(2l) \\ x_2(2l) \end{bmatrix},$$

for $l = 0, 1, \ldots$. In general, after a clock period all d entries have switched and consequently index l is incremented by d over a full clock period.

Consider now systems with n state variables and event pattern described by the sequence $s = (i_1, \ldots, i_d)$. Then

$$\begin{aligned} \mathbf{x}(ld + h_1) &= A_{i_1}\mathbf{x}(ld) + B_{i_1}\mathbf{u}(ld), \\ \mathbf{x}(ld + h_1 + h_2) &= A_{i_2}\mathbf{x}(ld + h_1) + B_{i_2}\mathbf{u}(ld + h_1), \\ \cdots \qquad \cdots \qquad & \cdots, \\ \mathbf{x}(ld + h_1 + \cdots + h_{d-1}) &= A_{i_{d-1}}\mathbf{x}(ld + h_1 + \cdots + h_{d-2}), \\ &\quad + B_{i_{d-1}}\mathbf{u}(ld + h_1 + \cdots + h_{d-2}), \\ \mathbf{x}(ld + n) &= A_{i_d}\mathbf{x}(ld + h_1 + \cdots + h_{d-1}), \\ &\quad + B_{i_d}\mathbf{u}(ld + h_1 + \cdots + h_{d-1}), \end{aligned}$$

for $l = 0, 1, \ldots$ Suppose also that the inputs are updated just after the new full state vector has been created, and hence

$$\mathbf{u}(ld + h_1 + \cdots + h_{d-1}) = \cdots = \mathbf{u}(ld + h_1) = \mathbf{u}(ld), \tag{11}$$

and by back-substitution we obtain

$$\begin{aligned}\mathbf{x}(ld+d) = A_{i_d}\cdots A_{i_1}\mathbf{x}(ld)+ \\ (B_{i_d}+A_{i_d}B_{i_{d-1}}+\cdots+A_{i_d}\cdots A_{i_2}B_{i_1})\mathbf{u}(ld).\end{aligned} \tag{12}$$

For a given sequence $s=(i_1,\ldots,i_d)$ define

$$A_s = A_{i_d}\cdots A_{i_1} \tag{13}$$

and

$$B_s = B_{i_d}+A_{i_d}B_{i_{d-1}}+\cdots+A_{i_d}\cdots A_{i_2}B_{i_1} \tag{14}$$

Then we can write (12) as

$$\mathbf{x}(ld+d) = A_s\mathbf{x}(ld)+B_s\mathbf{u}(ld) \tag{15}$$

and, returning to the full period counter, (15) can be written in the form of a state updating equation for a discrete linear system as

$$\mathbf{x}(k+1) = A_s\mathbf{x}(k)+B_s\mathbf{u}(k). \tag{16}$$

3 Stability and Robustness

It is known Lorand and Bauer (2006b), Kleptzyn et al. (1984) that the presence of synchronization errors can effect the stability of the overall system. In particular, consider a system with state dynamics described by

$$x(k+1) = Ax(k)+Bu(k) \tag{17}$$

where at instant k $x(k)$ is the $n\times 1$ state vector, and $u(k)$ is the $l\times 1$ control input vector. Suppose that in operation synchronization errors occur resulting in state dynamics that can be written in the form (16). Then it is possible that (17) is stable, that is, all eigenvalues of A have modulus strictly less than unity, but some of the state matrices in (16) resulting from the presence of synchronization errors do not have this essential property. Also the exact time sequence of arriving signals to subsequent sub-systems is not known, which makes stability analysis in the presence of these errors very difficult. In this section, we develop methods for this task by treating the complete set of possible systems as the effects of uncertainty on some nominal model. This releases Lyapunov-type methods from robust control of linear time-invariant systems for use in this problem area where, as a starting point, we use a polytopic robustness characterization. Next we introduce this characterization in terms of a system described by (17).

The assumption made at this stage is that in the presence of uncertainty in the system model, the system matrix A of (17) takes values in a fixed polytope, see, for example, Boyd et al. (1994),

$$A \in \mathrm{Co}\{A^1, A^2, \ldots, A^h\}, \tag{18}$$

where matrices $A^1, A^2, \ldots, A^h$ are given vertices and

$$\mathrm{Co}\{A^1, A^2, \ldots, A^h\} = \left\{\sum_{k=1}^{h}\alpha_k A^k : \alpha_k \geq 0, \sum_{k=1}^{h}\alpha_k = 1\right\}, \tag{19}$$

denotes the convex hull of $A^1, \dots, A^h$, that is, the polytope of matrices with given vertices $A^1, \dots, A^h$. Moreover, stability in the presence of such uncertainty can be determined by checking if this property holds for each of the polytope vertices as this guarantees that every system matrix formed as a convex combination of them is also stable Boyd et al. (1994). One method of doing this is to use Linear Matrix Inequalities (LMI's) and hence this property holds when

$$A^{i^{\mathrm{T}}} P A^i - P \prec 0 \tag{20}$$

for $i = 1, 2, \dots, h$ where $P \succ 0$.

Returning to the linear systems of the previous section which have clock synchronization errors characterized by the d-element sequence of events $s = \{i_1, \dots, i_d\}$ with state-space model (16), suppose that the system matrices A_s and B_s take values in the polytope

$$[A_s\ B_s] \ \in \ \mathrm{Co}\{[A^i\ B^i] : i = 1, \dots, h\}, \tag{21}$$

and apply the state feedback control law

$$\mathbf{u}(ln) = K\mathbf{x}(ln). \tag{22}$$

Then

$$\mathbf{x}(ln + n) = (A_s + B_s K)\mathbf{x}(ln), \tag{23}$$

where

$$A_s + B_s K \in \mathrm{Co}\{A^i + B^i K : i = 1, \dots, h\}. \tag{24}$$

Hence (23) is stable if there exists a $P \succ 0$ such that the following system of inequalities is satisfied

$$(A^i + B^i K)^T P (A^i + B^i K) - P \prec 0 \qquad i = 1, \dots, h. \tag{25}$$

The difficulty now is that this last system is not linear with respect to the matrix K and therefore cannot be easily solved numerically. Direct application of the result in Crusius and Trofino (1999) now gives the following result.

Lemma 3.1: *The condition of (25) holds if there exist compatibly dimensioned matrices $Q \succ 0$ and R such that the following system of LMIs*

$$\begin{bmatrix} -Q & A^i Q + B^i R \\ Q A^{i^T} + R^T B^{i^T} & -Q \end{bmatrix} \prec 0, \qquad i = 1, \dots, h. \tag{26}$$

are feasible, and then

$$K = RQ^{-1}, \tag{27}$$

is a stabilizing control law matrix.

The solution of (26) can be conservative since the system of LMIs are solved with a common decision matrix Q and hence a common Lyapunov function. This problem is well known in the robust control literature and many ways of reducing the levels of conservativeness have been developed, and these can be applied to the current application. One difficulty for the systems considered in this paper is a very large number of sequences s in a given application, which is addressed next.

Consider a sequence of length d that represents one event-switching pattern for a system with n state variables $s = \{i_1, \ldots, i_d\}$, and suppose we want to add another variable to the system. Then this could be positioned in the sequence s either as an element of i_j, $j = 1, \ldots, d$, or before each i_j, or at the end. Thus, from one sequence of length d, which represents one switching pattern for the system with n state variables, we obtain d sequences of length d, and $d+1$ sequences of length $d+1$. Starting from the sequence of length 1 for system with 1 state variable ($s = (\{1\})$, we obtain 1 sequence of length 1 and two sequences of length 2, that is, $(\{1,2\})$ and $(\{2\},\{1\})$, $(\{1\},\{2\})$. Hence the system with 2 state variables has $1 + 2 = 3$ different sequences.

If we add a new state variable the sequence $(\{1,2\})$ generates 1 sequence of length 1 - $(\{1,2,3\})$ and 2 sequences of length 2 - $(\{3\},\{1,2\})$,$(\{1,2\},\{3\})$. Similarly, $(\{2\},\{1\})$ generates 2 sequences of length 2, that is, $(\{2,3\},\{1\})$,$(\{2\},\{1,3\})$ and 3 of length 3, that is. $(\{3\},\{2\},\{1\})$,$(\{2\},\{3\},\{1\})$,$(\{2\},\{1\},\{3\})$. Also the sequence $(\{1\},\{2\})$ gives $(\{1,3\},\{2\})$,$(\{1\},\{2,3\})$ and $(\{3\},\{1\},\{2\})$,$(\{1\},\{3\},\{2\})$,$(\{1\},\{2\},\{3\})$. Hence a system with 3 state variables has $1*(1+2)+2*(2+3) = 13$ different sequences that can be represented by the tree shown in Fig. (2).

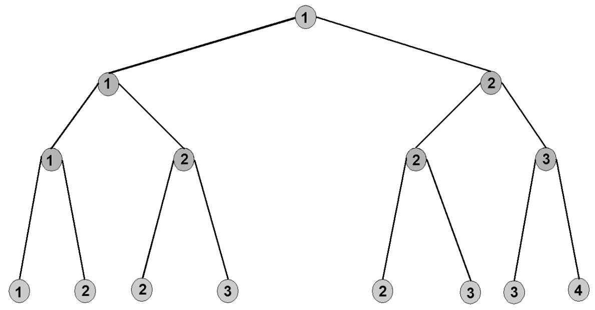

Figure 2. The tree representation detailed to a depth of three.

The number of sequences for given n, corresponding to a tree of depth +1, equals the sum of all products of values at nodes from the root to the leaves. For example, in the case when $n = 3$ the number of sequences, denoted n.o.s, is

$$\text{n.o.s} = 1 \cdot 1 \cdot 1 + 1 \cdot 1 \cdot 2 + 1 \cdot 2 \cdot 2 + 1 \cdot 2 \cdot 3 = 1 + 2 + 4 + 6 = 13 \qquad (28)$$

For large n we can estimate this number where for a lower bound we assume that there is only one leaf with value of n (and one path). Moreover, for the upper limit we assume that all leaves have the value n and the nodes have a depth of +1. Hence for given n the n.o.s is estimated as

$$n! \leq \text{n.o.s} \leq 2^{n-1} n!. \qquad (29)$$

Consider direct computation, where we treat all the product matrices A_s and B_s as vertices of the polytope, that is, the system of LMI's is solved directly for the product matrices. Then this method requires that all these matrices are stored in memory and hence there are serious capacity limits. One alternative is to attempt to find a polytope that contains all these product matrices and is characterized by a lower number of vertices. There are, however, many reasons why this route is also not feasible except for very low values of n. One of these is the computer memory requirements. Consequently we next describe a new method based on computing a bounding polytope with lower number of vertices that allows a solution to be computed in much shorter time.

4 A fast algorithm for polytope computation

The basic idea is to treat all the product matrices as vectors and, by linear operations, enclose them in a simple structure for which it is easy to obtain their convex hull. Here this is taken as the unit ball and we now develop the algorithm.

Let $\mathcal{M}_{m\times n}(\mathbb{R})$ be the space of $m \times n$ matrices with real entries and consider the map $\phi : \mathcal{M}_{m\times n}(\mathbb{R}) \to \mathbb{R}^{mn}$ defined for a given matrix

$$M = [m_{ij}], \quad 1 \le i \le n, 1 \le j \le m,$$

as

$$\begin{aligned}\phi(M) = [\,& m_{11}, m_{21}, \ldots, m_{m1}, m_{12}, m_{22}, \ldots, m_{m,n-1},\\ & m_{1n}, m_{2n}, \ldots\ m_{mn}].\end{aligned} \tag{30}$$

Then ϕ has an inverse and for $\mathbf{x} = [\,x_1, x_2, \ldots, x_{n\cdot m}\,]^{\mathrm{T}}$, this is given by

$$M = \phi^{-1}(\mathbf{x}) = \begin{bmatrix} x_1 & x_{m+1} & \cdots & x_{(n-1)\cdot m+1} \\ x_2 & x_{m+2} & \cdots & x_{(n-1)\cdot m+2} \\ \cdots & \cdots & \cdots & \cdots \\ x_m & x_{2m} & \cdots & x_{n\cdot m} \end{bmatrix}, \tag{31}$$

and it is easy to show that this map is linear and continuous.
Assume that

$$\mathcal{M}_p = \{[A_{si}, B_{si}] : 1 \le i \le n_p\}, \tag{32}$$

and

$$\mathbf{x}_i = \phi([A_{si}, B_{si}]) \quad 1 \le i \le n_p\}. \tag{33}$$

Then the image of $\mathcal{M}_p$ in the vector space is

$$\mathcal{P} = \phi(\mathcal{M}_p). \tag{34}$$

Introduce now its center of the mass

$$\mathbf{c} = -\frac{1}{n_p}\sum_{i=1}^{n_p} \mathbf{x}_i, \tag{35}$$

and also

$$\mathcal{P}(\mathbf{c}) = \tau_c(\mathcal{P}) = \{\mathbf{x} \in \mathbb{R}^{n^2+n\cdot l} : \mathbf{x} - \mathbf{c} \in \mathcal{P}\}, \tag{36}$$

that is, the translation of the set $\mathcal{P}$ by the vector $\mathbf{c}$, which is denoted by τ_c.

Denote the subspace spanned by $\mathcal{P}(\mathbf{c})$ as

$$V_c = \mathrm{lin}\mathcal{P}(\mathbf{c}), \quad d = \dim V_c \tag{37}$$

and in the case when $l = n$ we have that

$$d \le n^2 - n, \tag{38}$$

that is, the resulting space is of lower dimension. However, this may not be true if $\mathcal{P}(\mathbf{c})$ is of lower dimension than the space itself. In order to prevent such a case from arising, and to simplify calculations, we need to obtain the co-ordinates of $\mathbf{x} \in \mathcal{P}$ in the basis of V_c.

Let

$$\mathcal{B}_1 = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_d\}, \tag{39}$$

be the orthonormal basis of V_c and

$$\mathcal{B}_2 = \{\mathbf{b}_{d+1}, \dots, \mathbf{b}_{n^2+n\cdot l}\}, \tag{40}$$

the orthonormal basis of the orthogonal complement $V_c^{\perp}$. Then

$$V_c \oplus V_c^{\perp} = \mathbb{R}^{n^2+n\cdot l}, \tag{41}$$

and if we define the matrix

$$B = [\,\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{n^2+n\cdot l}], \tag{42}$$

we have $B^T \cdot B = I$, and also

$$\mathbf{x}' = \mathbf{x}B^{\mathrm{T}}, \quad x'_{d+1} = \cdots = x'_{n^2+n\cdot l} = 0. \tag{43}$$

Introduce $\mathbf{x} = B \cdot \mathbf{x}'$, where $\mathbf{x}'$ denotes the coordinates of vector $\mathbf{x}$ in the orthonormal basis of V_c. Also define the injection $\omega : V_c \to \mathbb{R}^d$ as

$$\mathbf{y} = \omega(\mathbf{Bx}) = [\,x'_1, x'_2, \dots, x'_d\,]^{\mathrm{T}}, \tag{44}$$

such that $\mathbf{y}$ contains the coordinates of $\mathbf{x}$ in $\mathcal{B}_1$. Finally, introduce

$$\mathcal{R} = \omega(\mathcal{P}(\mathbf{c})), \tag{45}$$

and the map

$$\lambda = \phi \circ \tau_c \circ \omega, \tag{46}$$

such that

$$\mathcal{R} = \lambda(\mathcal{M}_p). \tag{47}$$

Figure 3 gives a graphical illustration of the transformations just defined.

Computations can now be performed in a lower dimension space, that is R^d, where d has been defined in the previous section (the clock synchronization errors are characterized by the d-element sequence of events $s = \{i_1, \dots, i_d\}$. In particular, consider the set R. Then using any of the well known methods we can compute the minimal volume ellipsoids, that is the matrix E and the point $\mathbf{e}$ such that the set

$$E^{\star} = \{\mathbf{y} \in \mathbb{R}^d : (\mathbf{y} - \mathbf{e})^{\mathrm{T}}\}E(\mathbf{y} - \mathbf{e}) \leq 1\}, \tag{48}$$

is of minimal volume. In this case $E \succ 0$, and by Cholesky factorization we can obtain the matrix H such that $E = H^{\mathrm{T}}H$. Let $z = H\mathbf{y}, \quad f = H\mathbf{c}$, then the set $H(E^{\star})$ can be written as

$$H(E^{\star}) = \{\mathbf{z} \in \mathbb{R}^d : (\mathbf{z} - \mathbf{f})^{\mathrm{T}}(\mathbf{z} - \mathbf{f}) \leq 1\} = B(\mathbf{f}, 1), \tag{49}$$

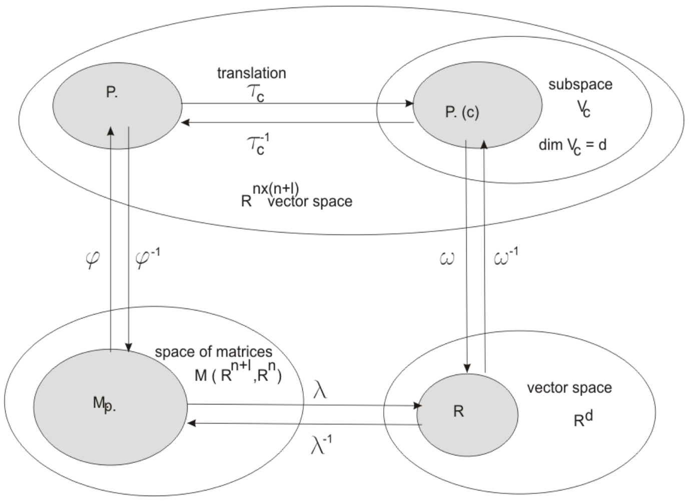

Figure 3. Diagram of transformation for matrices $Z_s = [\,As, Bs\,]$

and the entire ellipsoid has been transformed into the ball of unit radius, as illustrated in Fig. 4.

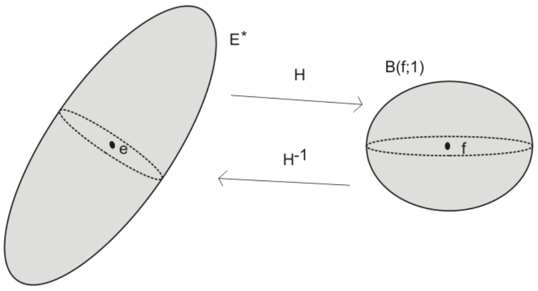

Figure 4. Illustrating the transformation of the ellipsoid into unit ball.

In the case of the polytope, let $k > 0$ be fixed and also let $\mathcal{D}$ be the set of vectors $\mathbf{d}_{ij}$, $1 \le i \le d, j \in \{1, 2\}$, such that

$$\begin{aligned}
\mathbf{d}_{11} &= [-k, 0, 0, \ldots, 0]^{\mathrm{T}} + \mathbf{f},\\
\mathbf{d}_{21} &= [0, -k, 0, \ldots, 0]^{\mathrm{T}} + \mathbf{f},\\
&\ldots\ldots\ldots,\\
\mathbf{d}_{21} &= [\ k, 0, 0, \ldots, 0\,]^{\mathrm{T}} + \mathbf{f},\\
\mathbf{d}_{21} &= [\ 0, k, 0, \ldots, 0\,]^{\mathrm{T}} + \mathbf{f},\\
&\ldots\ldots\ldots,\\
\mathbf{d}_{ij} &= -(1)^j \cdot [\ 0, ..., k, 0, \ldots, 0\,]^{\mathrm{T}} + \mathbf{f},\\
&\ldots\ldots\ldots,\\
\mathbf{d}_{d1} &= [\ 0, 0, 0, \ldots, -k\,]^{\mathrm{T}} + \mathbf{f},\\
\mathbf{d}_{d2} &= [\ 0, 0, 0, \ldots, k\,]^{\mathrm{T}} + \mathbf{f},
\end{aligned}$$

that is,

$$\mathcal{D} = \{\mathbf{d}_{ij} \in \mathbb{R}^d :\ 1 \le i \le d, j \in \{1, 2\}\ \} \tag{50}$$

Now consider the points $\mathbf{d}_{i2}$, $i = 1, \ldots, d$, with positive entries. Then these span the $(d-1)$-

dimensional hyperplane in $\mathbb{R}^{d_Z}$. Also the point

$$\mathbf{p} = (p_1, p_2, \ldots, p_d) \in \mathbb{R}^d, \tag{51}$$

belongs to this plane if

$$(p_1 - f_1) + (p_2 - f_2) + \cdots + (p_d - f_d) = k. \tag{52}$$

Also, by symmetry, the closest point to the center of the ball in this plane, denoted by $\mathbf{f}$, is

$$\mathbf{d}^\star = (\frac{k}{d}, \ldots, \frac{k}{d}), \tag{53}$$

and hence

$$\|\mathbf{d}^\star - \mathbf{f}\|_2 = k \cdot \sqrt{d}. \tag{54}$$

If we choose $k = \frac{1}{\sqrt{d}}$ then the distance between plane and the point $\mathbf{f}$ is 1. Also, by symmetry, if we take as vertices

$$V = \{\mathbf{v}_{ij} \in R^d : \mathbf{v}_{ij} = (-1)^j \cdot \frac{1}{\sqrt{d}} \cdot \mathbf{d}_{ij}\, i = 1, .., d,\ j = 1, 2\} \tag{55}$$

then the ball $B(\mathbf{f}, 1)$ belongs to the convex hull with vertices V. Moreover, the transformation $\gamma_Z^{-1} : \mathbb{R}^d \to \mathcal{M}_{n\times(n+l)}(\mathbb{R})$ is linear and hence under its action the convex hull remains convex and the resulting vertices are simply $\gamma^{-1}(V)$

$$\gamma^{-1}(\text{Co } (V)) = \text{Co } (\gamma^{-1}(V)). \tag{56}$$

Figure 5 illustrates this polytope construction method. The polytope obtained is now used to

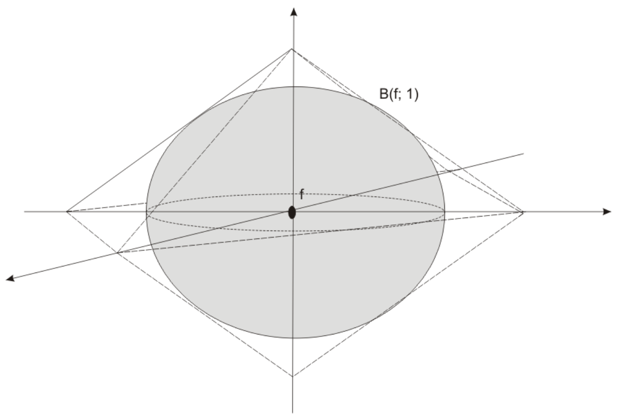

Figure 5. Visualization of the polytope formation in $\mathbb{R}^3$

produce the set of LMI's. If these are feasible then we accept them as a solution. If these LMIs have no solution then the method fails.

The main advantage of the algorithm developed above is that is fast and requires the solution of a much lower number of LMI's. If n is the state dimension of the system, the number of vertices of the polytope is $2(n^2 - n)$. Table 1 below gives a comparison of the time needed to compute the solution by both methods, where the entry $---$ is used to denote the fact that no solution could be computed by the direct method. Table 2 gives comparative figures on the numbers of product matrices and polytope vertices.

n	direct computation avg time (sec)	computation with new algorithm avg time(sec)
1	-	-
2	0.187	0.3
3	0.829	0.7
4	10.109	2.8
5	148.14	11.2
6	12000	101.8
7	—	6000

Table 1. Time needed to solve the whole problem in the case when $l = n$.

n	number of product matrices	$(1.41)^n n!$	number of vertices of the polytope $2(n^2 - n)$
1	1	1.41	0
2	3	3.98	4
3	13	16.82	12
4	75	94.86	24
5	541	668.77	40
6	4683	5657.79	60
7	47293	55842.43	84
8	545835	629902.6	112
9	7087261	7993464	144
10	102247563	11270780	180

Table 2. Number of product matrices and vertices in the case when $l = n$.

Note that for $n = 6$ the method is over 100 times faster than direct computation and this advantage should increase for $n > 6$. An interior point algorithm, which is a polynomial algorithm, is used to solve the set of LMI's for both (direct and the one developed here) methods, but here the number of input matrices is reduced to polynomial of order n using a fast polytope computation algorithm (which is also polynomial) and then solves the LMI's. The other new step is a polynomial algorithm of the number of input matrices but of lower dimension than the interior point method. Note also that the number of input matrices can be approximated as $\sqrt{2}^n \cdot n!$.

5 An Example

Consider a swarm system consisting of M agents each of which is modeled as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i (n+1) = \begin{bmatrix} \delta_i & -\omega_i \\ \omega_i & \delta_i \end{bmatrix} \cdot \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i (n) - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_i (n) \right), \tag{57}$$

where $i = 1, 2, \dots M$, and

$$\delta_i^2 + \omega_i^2 = 1 - \varepsilon. \tag{58}$$

The agents work together and aim to meet at the rendezvous point. The input $[\, u_1 \quad u_2]_i^T$ is

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} (n) = \frac{1}{M} \left(\sum_{i=1}^{M} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i (n) \right) + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}_i (n), \tag{59}$$

where $[\, e_1,\, e_2\,]_i^T$, $i = 1, \ldots, M$ denotes an independent agent input vector.

Introducing the variables $\mathbf{x}' \in \mathbb{R}^{2M}$ and $\mathbf{u}' \in \mathbb{R}^{2M}$ as

$$x'_j = \begin{cases} (x_1)_{\frac{j+1}{2}} & \text{if j is odd} \\ (x_2)_{\frac{j}{2}} & \text{if j is even} \end{cases} \quad j = 1, 2, \ldots 2M, \tag{60}$$

$$u'_j = \begin{cases} (e_1)_{\frac{j+1}{2}} & \text{if j is odd} \\ (e_2)_{\frac{j}{2}} & \text{if j is even} \end{cases} \quad j = 1, 2, \ldots, 2M, \tag{61}$$

enables the system model to be rewritten as

$$\mathbf{x}'(n) = A_1 \mathbf{x}'(n) - A_1 \cdot \left(A_2 \mathbf{x}'(n) + \mathbf{u} \right), \tag{62}$$

or

$$\mathbf{x}'(n) = A_1 \left(I - A_2 \right) \mathbf{x}'(n) - A_1 \mathbf{u}', \tag{63}$$

where

$$A_1 = \begin{bmatrix} \delta_1 & -\omega_1 & 0 & \ldots & \ldots & 0 \\ \omega_1 & \delta_1 & 0 & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & \delta_i & -\omega_i & \ldots & 0 \\ 0 & \ldots & \omega_i & \delta_i & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & 0 & \delta_M & -\omega_M \\ 0 & \ldots & \ldots & 0 & \omega_M & \delta_M \end{bmatrix}, \tag{64}$$

and

$$A_2 = \begin{bmatrix} \frac{1}{M} & 0 & \ldots\ldots & \frac{1}{M} & 0 \\ 0 & \frac{1}{M} & \ldots\ldots & 0 & \frac{1}{M} \\ \ldots & \ldots & \ldots\ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots\ldots & \ldots & \ldots \\ \frac{1}{M} & 0 & \ldots\ldots & \frac{1}{M} & 0 \\ 0 & \frac{1}{M} & \ldots\ldots & 0 & \frac{1}{M} \end{bmatrix}. \tag{65}$$

Introducing

$$A' = A_1 \cdot (I - A_2) \quad B' = -A_1, \tag{66}$$

yields the state-space model

$$\mathbf{x}'(n+1) = A' \mathbf{x}'(n) + B' \mathbf{u}'. \tag{67}$$

In order to deal with synchronization errors we assume that agents' clocks are out of phase but they have the same time period T. Then we can use the model of the synchronization errors with only minor modifications. We consider only those product matrices that are relevant to the agent's work, that is, we assume that pairs x_i, x_{i+1}, $i = 1, 3, 5, \ldots, 2M-1$. work synchronously. Then we apply the algorithm to find an admissible control law.

Suppose that the parameters in the model here vary as follows

$$\epsilon \in [0,1] \qquad \omega_i \in [0, \sqrt{1-\epsilon}] \qquad \delta_i^2 + \omega_i^2 = 1-\epsilon, \tag{68}$$

Then in Fig. 6 synchronization errors occur in the dark grey region, that is, some product matrices are unstable. The stable region is marked in light grey.

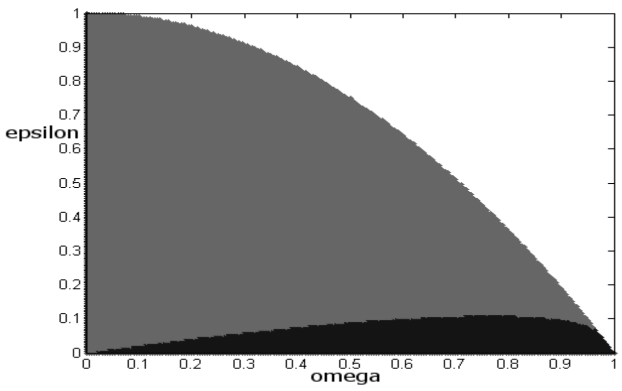

Figure 6. Unstable region; dark grey, stable region: light grey.

Consider now the above system for $M = 6$ with

$$\delta_i = \delta_j \qquad \omega_i = \omega_j \qquad i,j = 1,2,\ldots,6, \tag{69}$$

where $\varepsilon = 0.1819$, $\omega_1 = 0.9 \rightarrow \delta_1 = 0.09$ and the system is controllable. If we assume that each agent works synchronously and that synchronization errors only arise when agents are performing common tasks, we have only 4683 product matrices. As the sequence $\{\{9,10,11,12\},\{7,8\},\{5,6\},\{3,4\},\{1,2\}\}$ illustrates, some of these product matrices are unstable. Using the algorithm developed here we can find a control law to guarantee that the closed-loop system is stable independent of the synchronization errors. The computation time was 500 sec as compared to 640 sec for direct computation and this advantage should increase with the number of agents present.

6 Conclusions

This paper has considered synchronization errors that can arise when the switching of all components of a system driven by a single clock do not switch at exactly the same time instant, that is, they operate in an asynchronous mode. The underlying assumption made for subsystems driven by a single clock is that the differences in clock propagation time between the different system components are negligible. Currently, there are many applications that routinely violate this basic assumption and it is known that system stability can be lost that if the mode of operation switches from synchronous to asynchronous.

Previous work had addressed the stability question and the first contribution of this paper is state feedback based stabilization of an asynchronous network. The method developed here treats the complete set of possible systems as the effects of uncertainty on some nominal model. As a result, Lyapunov type methods from robust control of linear time-invariant systems for use in this problem area which results in LMI based computations for determining stability and control law design. The novelty here lies in the application of tools from the robust control theory for standard linear systems in a different problem area. Future research topics include the use of alternatives to polytopic robustness that may give improved results in at least some examples.

The problem of the number of computations involved has also been considered resulting in the development of a computationally less intense approach. Again there is much further work here to enable efficient handling of large scale problems.

References

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, Vol. 15 of *Studies in Applied Mathematics*, Philadelphia, PA: SIAM (1994).

Crusius, C.A.R., and Trofino, A. (1999), "Sufficient LMI conditions for output feedback control problems," *IEEE Trans. Automatic Control*, 44, 1053–1057.

Gazi, V. (2008), "Stability of an asynchronous swarm with time-dependent communication links," *IEEE Trans Systems, Man and Cybernetics, Part B.*, 38, 267–274.

Kleptzyn, A.F., Kozyakin, V.S., Krasnosel'skii, M., and Kuznetsov, N. (1984), "Effects of small synchronization errors on complex systems," *Avtomatika o Telemekhanika, Translated in: Aut. and Remote control*, pp. 1014–1018.

Li, W. (2008), "Stability analysis of swarms with general topology," *IEEE Trans Systems, Man and Cybernetics, Part B.*, 38, 1084–1097.

Lorand, C. (2004), "A Theory of synchronization errors," PhD. dissertation, University of Notre Dame, Dept. of Electrical Engineering.

Lorand, C., and Bauer, P. (2005), "Distributed discrete time systems with synchronization errors: Models and stability," *IEEE Trans on Circ & Syst. II: Express Briefs*, 52, 208–213.

Lorand, C., and Bauer, P. (2006b), "Clock Synchronization Errors in Circuits: Models, stability and fault detection," *IEEE Transaction on Circ. and Syst I: Regular papers*, 53, 2299–2305.

Lorand, C., and Bauer, P. (2006a), "On synchronization errors in networked feedback systems," *IEEE Trans. on Circ & Syst I: Regular Papers*, 53, 2306–2317.

Su, Y.F., Bhaya, A., Kaszkurewicz, E., and Kozyakin, V. (1997), "Further results on stability of asynchronous discrete time linear system," in *Proc. IEEE Conf. on Decision and Control*, San Diego, CA, USA, dec, pp. 915–920.

Thierrauf, S.C., *High-speed Circuit Board Signal Integrity*, Norwood, MA: ARTECH House, Inc (2004).