2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Stability and robustness of systems with synchronization errors

Marek Przedwojski, Krzysztof Galkowski, Peter H Bauer, Eric Rogers

***Abstract*— This paper addresses the problem of synchronization errors in distributed dynamical systems. In particular, it focuses on the question of stability for the case where all subsystems have the same sampling frequency, but different switching times. In contrast to previous work, the approach taken here models the set of system matrices that arise using a polytopic uncertainty approach, where a polytope vertex corresponds to a possible state matrix of the overall system. System stabilization is then approached through state feedback and LMI techniques are used to generate the control law matrices. A method to handle the combinatorial explosion of the number of polytope vertices is developed and illustrated using an example from swarm system navigation.**

I. 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 other words, 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 it is essential to shed more light on the modeling, analysis and design of asynchronous systems. Compared to other networking effects [1], such as delay, packet drop and bandwidth limitations, the area of synchronization errors has received relatively little interest in the literature. Increasingly, synchronization errors play an important role in both networked systems and high speed circuitry. Even though high speed high speed circuits have been designed to function as a synchronous system, extremely high clock speed could result in propagation delay differences of the order of a period of the system clock.

Several approaches to tackling this problem have already been investigated [2]-[4]. It was shown in [5] that most synchronous systems totally change their response if operated asynchronously. Even stability is often lost [5]-[6] and there is generally no stability robustness with respect to synchronization errors, with the exception of work in [7], [8] that are very limited in terms of application to practically relevant cases.

In general, two types of asynchronous behavior can arise [5], where in one all subsystems are driven by the same clock but switching times are different between subsystems, and in the other individual subsystems are driven by the same clocks with different frequencies. In the first case, switching patterns are simple but in the second case are very hard to specify [9]. The first case typically describes high speed circuits and low speed networked systems that periodically re-synchronize. The second case typically occurs in high speed control networks or networked systems where local system time is not monitored and corrected.

In this paper we consider the first case. Previous work has addressed system models and the resulting stability behavior [5]-[9], but no attempt has been made to stabilize the overall asynchronous system network. In this paper we give major new results in this area by showing how to stabilize an asynchronous network of systems through state feedback.

In this paper $\Gamma \succ 0$ and $\Gamma \prec 0$ respectively are used to denote symmetric matrices which are positive definite and negative definite respectively.

II. Background

A discrete time system in this work is taken to mean any processing block that takes an input sequence of samples an produces an output sequence of samples. Here we consider the linear time-invariant case with state-space model

$$\begin{aligned}\mathbf{x}(k+1) &= A\mathbf{x}(k)+B\mathbf{u}(k),\\ \mathbf{y}(k) &= C\mathbf{x}(k)+D\mathbf{u}(k),\end{aligned} \qquad (1)$$

where at time k, $\mathbf{x}(k)$ is the $n\times 1$ state vector, $\mathbf{y}(k)$ is the $m\times 1$ output vector, $\mathbf{u}(k)$ is the $l\times 1$ input vector, and A, B, C and D are matrices of compatible dimensions with real entries. Asynchronous switching in a linear time-invariant systems is discussed in, for example, [6], [7] and here we begin 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, $i = 1, 2, ..., n$. The clock rates are equal ($T_i = T, i = 1, 2, ..., n$) but the associated signals are out of phase. In order to capture effects of asynchronous switching, an event driven time index k is introduced, which is increased by an integer equal to the number of variables that switch simultaneously.

Following [6] we assume that after each switching event the time index is incremented by the number of simultaneously switching variables. Consequently over a full clock

M. Przedwojski and E. Rogers are with the School of Electronics and Computer Science, University of Southampton, UK `mp08r,etar@ecs.soton.ac.uk`

K. Galkowski is with the Institute of Control and Computation Engineering University of Zielona Gora, Poland

P. H. Bauer is with the Department of Electrical Engineering, University of Notre Dame, USA

978-1-4244-4524-0/09/$25.00 ©2009 AACC

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

$$s = (i_1, i_2, \ldots, i_d), \quad i_j \subseteq \{1, \ldots, n\}, \quad j = 1, \ldots, d. \tag{2}$$

These subsets satisfy

$$p \neq r \ \Rightarrow \ i_p \cap i_r = \emptyset \qquad p, r = 1, \ldots, d, \tag{3}$$

and

$$\bigcup_{j=1}^{d} i_j = \{1, \ldots, n\}. \tag{4}$$

The number of elements of the each subset i_j

$$h_j = \text{card}(i_j), \tag{5}$$

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

$$h_1 + \cdots + h_d = n. \tag{6}$$

Now the equation for a single event numbered j can be written as

$$\begin{aligned} \mathbf{x}(ld + h_1 + \cdots + h_j) \;&=\; A_{i_j}\mathbf{x}(ld + h_1 + \cdots + h_{j-1}) \\ &+\; B_{i_j}\mathbf{u}(ld + h_1 + \cdots + h_{j-1}), \end{aligned} \tag{7}$$

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

$$A_{i_j} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{p1} & a_{p2} & \ldots & a_{p(n-1)} & a_{pn} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{q1} & a_{q2} & \ldots & a_{q(n-1)} & a_{qn} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix}, \tag{8}$$

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

$$B_{i_j} = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ b_{p1} & b_{p2} & \ldots & b_{p(l-1)} & b_{pl} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ b_{q1} & b_{q2} & \ldots & b_{q(l-1)} & b_{ql} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 \end{bmatrix}. \tag{9}$$

Here A_{i_j} contains only those rows from the state-space matrix A that are relevant to simultaneously switched variables (one at least), 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.

Consider now an n-variable system with event pattern described by the sequence $s = (i_1, \ldots, i_d)$. Then for $l = 0, 1, \ldots$

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

Suppose also that the inputs are updated just after the new full state vector has been created and hence

$$\mathbf{u}(ln + h_1 + \cdots + h_{d-1}) = \cdots = \mathbf{u}(ln + h_1) = \mathbf{u}(ln), \tag{10}$$

and by back-substitution we obtain

$$\begin{aligned} \mathbf{x}(ln + n) &= A_{i_d} \cdots A_{i_1}\mathbf{x}(ln) \\ &\quad + (B_{i_d} + A_{i_d}B_{i_{d-1}} + \cdots + A_{i_d} \cdots A_{i_2}B_{i_1})\mathbf{u}(ln). \end{aligned} \tag{11}$$

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

$$A_s = A_{i_d} \cdots A_{i_1}, \tag{12}$$

and

$$B_s = B_{i_d} + A_{i_d}B_{i_{d-1}} + \cdots + A_{i_d} \cdots A_{i_2}B_{i_1}. \tag{13}$$

Then we can write (11) as

$$\mathbf{x}(ln + n) = A_s\mathbf{x}(ln) + B_s\mathbf{u}(ln), \tag{14}$$

and returning to the full period counter the state equation (14) can be written in the standard form as

$$\mathbf{x}(k + 1) = A_s\mathbf{x}(k) + B_s\mathbf{u}(k). \tag{15}$$

III. Stability and Robustness

Consider again the discrete linear system (1), which is asymptotically stable if, and only if, the spectral radius (if H is a $h \times h$ matrix with eigenvalues λ_i, $i \leq i \leq h$, its spectral radius, written $r(H)$, is given by $r(H) = \max_{1 \leq i \leq h} |\lambda_i|$) of the system matrix A is less than unity, or there exists a symmetric positive definite matrix P, written $P \succ 0$, such that

$$A^T P A - P \prec 0. \tag{16}$$

It is well known [6], [7] that the synchronization errors can effect the stability of the overall system, i.e. a system with no synchronization errors described by (1) can be stable but some of the systems (14) resulting from synchronization errors can be unstable. Also the exact time sequence of arriving signals to subsequent sub-systems is not known, which makes stability analysis very difficult. In this paper, 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.

A. Polytopic Uncertainty Analysis

For a linear time-invariant system of the form (1) the assumption is that in the presence of uncertainty the system matrix A takes values in a fixed polytope (see, for example, [10]):

$$A \in \text{Co}\{A^1, A^2, \ldots, A^h\}, \tag{17}$$

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

$$\text{Co}\{A^1, A^2, \ldots, A^h\} = \left\{ \sum_{k=1}^{h} \alpha_k A^k : \alpha_k \geq 0, \sum_{k=1}^{h} \alpha_k = 1 \right\}, \tag{18}$$

denotes the convex hull of $A^1, \ldots, A^h$, (the polytope of matrices with given vertices $A^1, \ldots, A^h$). To investigate stability in the presence of such uncertainty it is only necessary to check if this property holds for the polytope vertices as this guarantees that every system matrix formed from a convex combination of them is also stable [10]. Hence only the following set of Linear Matrix Inequalities (LMI's) needs to be satisfied for robust stability to hold

$${A^i}^{\mathrm{T}} P A^i - P \prec 0, \tag{19}$$

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

For the system with no input and clock synchronization errors characterized by the d-element sequence of events $s = \{i_1, \ldots, i_d\}$ i.e.

$$\mathbf{x}(ln + n) = A_s \mathbf{x}(ln), \tag{20}$$

the system matrix A_s takes values in the polytope

$$A_s \in \text{Co}\{A^i : i = 1, \ldots, h\}. \tag{21}$$

Hence, to check the stability for every possible synchronization errors it is sufficient to solve the LMIs (19) for all vertices. Note here that the interior of the polytope newer occurs as a switching matrix and hence only the vertices are important for our analysis.

Consider now the system with clock synchronization errors characterized by the d-element sequence of events $s = \{i_1, \ldots, i_d\}$

$$\mathbf{x}(ln + n) = A_s \mathbf{x}(ln) + B_s \mathbf{u}(ln), \tag{22}$$

where system matrices A_s and B_s take values in the polytope

$$[A_s \ B_s] \in \text{Co}\{[A^i \ B^i] : i = 1, \ldots, h\}, \tag{23}$$

and apply the state feedback control law

$$\mathbf{u}(ln) = K\mathbf{x}(ln). \tag{24}$$

Then

$$\mathbf{x}(ln + n) = (A_s + B_s K)\mathbf{x}(ln), \tag{25}$$

where

$$A_s + B_s K \in \text{Co}\{A^i + B^i K : i = 1, \ldots, h\}. \tag{26}$$

Hence (25) is stable if there exists a $P \succ 0$ such that the following system of inequalities is satisfied

$$(A^i + B^i K)^T P (A^i + B^i K) - P \prec 0 \qquad i = 1, \ldots, h. \tag{27}$$

The difficulty now is that this last system is not linear with respect to the matrix K and therefore cannot be easily solved numerically. However, using the Schur's complement formula and the approach in [11] we can replace (27) by the following system of LMIs

$$\begin{bmatrix} -Q & A^i Q + B^i R \\ Q{A^i}^T + R^T {B^i}^T & -Q \end{bmatrix} \prec 0 \qquad i = 1, \ldots, h, \tag{28}$$

Also if this LMI system is feasible

$$K = RQ^{-1}, \tag{29}$$

is a stabilizing control law matrix.

The solution of (28) can be conservative since we solve the system of LMIs with common decision matrix Q (or Lyapunov function). To reduce this, it is possible to use, for example, variable Lyapunov functions [10]. Also an estimate of the number of sequences (n.o.s) for given n is

$$n! \leq \text{n.o.s} \leq 2^{n-1} n!. \tag{30}$$

The solution developed here consists of the following steps.

1) Calculate the vertices of a polytope that contains all product matrices representing system behavior in case of synchronization errors.
2) Find a stabilizing control by solving the set of LMI's (28) for the vertices obtained in the previous set.

In order to efficiently compute the solution the number of polytope vertices obtained in the first step here should be significantly smaller than the number of product matrices. Moreover, the accuracy of the polytope should guarantee that the control law can be found if it exists but currently available LMI solvers do not guarantee that this can be achieved. Hence in order to manage the compromise between speed, accuracy and number of vertices a new algorithm is developed in the remainder of this paper and compared in tests against direct computation.

IV. Fast algorithm for polytope computation

The basis of the algorithm given below is to treat all the product matrices as vectors and, by linear operations, to enclose them in a simple structure (unit ball) and hence lessen the computational load incurred in obtaining the convex hull containing them. Each step is now detailed.

1) Mapping matrices onto vectors: Let $\mathcal{M}_{m\times n}(\mathbb{R})$ be the space of the $m \times n$ matrices with real entries, write

$$M = [\ m_{ij}\ ]_{1\le i\le n, 1\le j\le m} \qquad \mathbf{x} = [\, x_k\,]_{1\le k\le mn}$$

and define the invertible map $\phi : \mathcal{M}_{m\times n}(\mathbb{R}) \to \mathbb{R}^{m\times n}$ as

$$\begin{aligned} \mathbf{x} &= \phi(M), \\ &= [m_{11}, m_{21}, \ldots, m_{m1}, m_{12}, m_{22}, \ldots, m_{m2}, \ldots \\ &\quad \ldots, m_{1n}, m_{2n}, \ldots m_{mn}], \end{aligned} \tag{31}$$

and

$$M = \phi^{-1}(\mathbf{x}) = \begin{bmatrix} x_1 & x_{m+1} & \ldots & x_{(n-1)\cdot m+1} \\ x_2 & x_{m+2} & \ldots & x_{(n-1)\cdot m+2} \\ \ldots & \ldots & \ldots & \ldots \\ x_m & x_{2m} & \ldots & x_{n\cdot m} \end{bmatrix}. \tag{32}$$

Then it is easy to show that this map is linear and continuous. Also Let $\mathcal{M}_p \subset \mathcal{M}_{n\times n+l}(\mathbb{R})$ denote the input set of compound product matrices, i.e.

$$\mathcal{M}_p = \{[As_i, Bs_i] : 1 \le i \le n_p\}, \tag{33}$$

where n_p is the number of product matrices. Then the image of $\mathcal{M}_p$ is

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

2) Calculation of the center of mass and translation: Introduce

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

and

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

i.e. the translation of the set $\mathcal{P}$ by the vector $\mathbf{c}$. Then the subspace spanned by $\mathcal{P}(\mathbf{c})$ is denoted by V_c and is isomorphic to $\mathbb{R}^d$, where d denotes the dimension of V_c.

3) Calculating the orthonormal basis of subspace spanned and new coordinates: Let

$$\mathcal{B}_1 = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\}, \tag{37}$$

be the orthonormal basis of V_c and introduce $B_{n^2+n\cdot l\times d}$ representing the coordinates of the orthonormal basis as

$$B = [\,\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d]. \tag{38}$$

Then $B^T \cdot B = I$ and let $\mathbf{x}'$ denote the vector of coordinates $\mathbf{x} \in \mathcal{P}(c)$ in the basis of V_c, i.e.

$$\mathbf{x} = x'_1\mathbf{b}_1 + \cdots + x'_d\mathbf{b}_d = B\mathbf{x}'. \tag{39}$$

Hence

$$\mathbf{x} = B\mathbf{x}' \qquad \mathbf{x}' = B^{\mathrm{T}}\mathbf{x}, \tag{40}$$

and we now introduce

$$\mathcal{R} = B^{\mathrm{T}}(\mathcal{P}(\mathbf{c})) \in \mathbb{R}^d. \tag{41}$$

4) Minimal volume ellipsoid: Computations can now be performed in $\mathbb{R}^d$, where d has been defined above, i.e. a much reduced dimension in comparison to the direct method. Using any of the available methods, we can now compute the minimal volume ellipsoids, i.e. the matrix E and the point $\mathbf{e}$, such that the set

$$E^\star = \{\mathbf{y} \in \mathbb{R}^d : (\mathbf{y} - \mathbf{e})^{\mathrm{T}}E(\mathbf{y} - \mathbf{e}) \le 1\}, \tag{42}$$

is of minimal volume and contains $\mathcal{R}$.

5) Changing ellipsoid into the ball of unit radius: The matrix E in (42) is symmetric and positive definite and by using a Cholesky factorization we can obtain H such that

$$E = H^{\mathrm{T}}H. \tag{43}$$

Now let

$$\mathbf{z} = H\mathbf{y}, \quad \mathbf{f} = H\mathbf{e}, \tag{44}$$

and the set $H(E^\star)$ can be written as

$$H(E^\star) = \{\mathbf{z} \in \mathbb{R}^d : (\mathbf{z} - \mathbf{f})^{\mathrm{T}}(\mathbf{z} - \mathbf{f}) \le 1\} = B(\mathbf{f}, 1), \tag{45}$$

and $H : \mathbb{R}^d \to \mathbb{R}^d$:

$$\mathbf{z} = H\mathbf{y} \qquad \mathbf{y} = H^{-1}\mathbf{z}. \tag{46}$$

maps the ellipsoid into the unit radius ball centered at f.

6) Choosing vertices of the target convex hull: Let $k > 0$ be fixed and $\mathcal{D}$ be the set of vectors $\mathbf{d}_{ij}$, $1 \le i \le d, j \in \{1, 2\}$, such that

$$\begin{aligned} \mathbf{d}_{11} &= [-k, 0, 0, \ldots, 0]^{\mathrm{T}} + \mathbf{f}, \\ \ldots &\ \ldots\ \ldots, \\ \mathbf{d}_{d1} &= [\ 0, 0, 0, \ldots, -k\,]^{\mathrm{T}} + \mathbf{f}, \\ \mathbf{d}_{12} &= [\ k, 0, 0, \ldots, 0\,]^{\mathrm{T}} + \mathbf{f}, \\ \ldots &\ \ldots\ \ldots, \\ \mathbf{d}_{d2} &= [\ 0, 0, 0, \ldots, k\,]^{\mathrm{T}} + \mathbf{f}, \end{aligned}$$

i.e.

$$\mathcal{D} = \{\mathbf{d}_{ij} \in \mathbb{R}^d :\ 1 \le i \le d, j \in \{1, 2\}\ \}. \tag{47}$$

Now consider the points $\mathbf{d}_{i2}$, $i = 1, \ldots, d$, with positive entries. These span the $(d-1)$-dimensional hyperplane in $\mathbb{R}^d$ and

$$\mathbf{p} = [\ p_1, p_2, \ldots, p_d\ ]^{\mathrm{T}} \in \mathbb{R}^d, \tag{48}$$

belongs to this plane if

$$(p_1 - f_1) + (p_2 - f_2) + \cdots + (p_d - f_d) = k. \tag{49}$$

By symmetry, the point $\mathbf{d}^\star$ of the plane that is closest to the center of the ball is the one with all coordinates equal, where these also have to satisfy (49). Hence

$$\mathbf{d}^\star = [\ \frac{k}{d}, \ldots, \frac{k}{d}\ ]^{\mathrm{T}} + \mathbf{f}, \tag{50}$$

and the distance to the center $\mathbf{f}$ is

$$\|\mathbf{d}^\star - \mathbf{f}\|_2 = \sqrt{d \cdot \frac{k^2}{d^2}} = \sqrt{\frac{k^2}{d}} = \frac{k}{\sqrt{d}}, \tag{51}$$

but, since this point belongs to the surface of the ball

$$\frac{k}{\sqrt{d}} = 1 \quad \Rightarrow \quad k = \sqrt{d}. \tag{52}$$

By symmetry we also have that if we take the vertices

$$\begin{aligned} V &= \left\{ \mathbf{v}_{ij} \in \mathbf{R}^d : \right. \\ \mathbf{v}_{ij} &= (-1)^j \cdot [\, 0, \dots, 0, (\sqrt{d})_{ith\ entry}, 0, \dots, 0]^{\mathrm{T}} \\ &+ \left. \mathbf{f},\ i = 1, .., d,\ j = 1, 2 \right\}, \end{aligned} \tag{53}$$

then the ball $B(\mathbf{f},\, 1)$ belongs to the convex hull with vertices V.

7) Obtaining the coordinates of the vertices in the original space: All of the transformations used in the steps above are linear and invertible. Hence it is routine to argue that the convex hull in $\mathbb{R}^d$ remains convex in $\mathcal{M}_{m\times n}(\mathbb{R})$. The polytope obtained is now used to produce the set of LMI's. If these are feasible then we accept them as a solution.

V. Test Results

The main advantage of the algorithm developed in the previous section 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 $2(n^2 - n)$. The table below gives a comparison of the time needed to compute the solution by both methods. Note

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

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. Also the number of input matrices can be approximated by $\sqrt{2}^n \cdot n!$.

VI. An Example

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

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i (n+1) = \begin{bmatrix} \delta_i & -\omega_i \\ \omega_i & \delta_i \end{bmatrix} \cdot \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i (n) - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_i (n) \right), \tag{54}$$

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

$$\delta_i^2 + \omega_i^2 = 1 - \varepsilon. \tag{55}$$

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

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}(n) = \frac{1}{M} \left(\sum_{i=1}^{M} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_i (n) \right) + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}_i (n), \tag{56}$$

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

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

$$x'_j = \begin{cases} (x_1)_{\frac{j+1}{2}} & \text{if j is odd} \\ (x_2)_{\frac{j}{2}} & \text{if j is even} \end{cases} \qquad j = 1, 2, \dots 2M, \tag{57}$$

$$u'_j = \begin{cases} (e_1)_{\frac{j+1}{2}} & \text{if j is odd} \\ (e_2)_{\frac{j}{2}} & \text{if j is even} \end{cases} \qquad j = 1, 2, \dots, 2M, \tag{58}$$

enables the system model to be written as

$$\mathbf{x}'(n) = A_1 \mathbf{x}'(n) - A_1 \cdot (A_2 \mathbf{x}'(n) + \mathbf{u}), \tag{59}$$

or

$$\mathbf{x}'(n) = A_1 (I - A_2) \mathbf{x}'(n) - A_1 \mathbf{u}', \tag{60}$$

where

$$A_1 = \begin{bmatrix} \delta_1 & -\omega_1 & 0 & \dots & \dots & 0 \\ \omega_1 & \delta_1 & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \delta_i & -\omega_i & \dots & 0 \\ 0 & \dots & \omega_i & \delta_i & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & \delta_M & -\omega_M \\ 0 & \dots & \dots & 0 & \omega_M & \delta_M \end{bmatrix}, \tag{61}$$

and

$$A_2 = \begin{bmatrix} \frac{1}{M} & 0 & \dots & \dots & \frac{1}{M} & 0 \\ 0 & \frac{1}{M} & \dots & \dots & 0 & \frac{1}{M} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{1}{M} & 0 & \dots & \dots & \frac{1}{M} & 0 \\ 0 & \frac{1}{M} & \dots & \dots & 0 & \frac{1}{M} \end{bmatrix}. \tag{62}$$

Introducing

$$A' = A_1 \cdot (I - A_2) \quad B' = -A_1, \tag{63}$$

yields the state-space model

$$\mathbf{x}'(n+1) = A' \mathbf{x}'(n) + B' \mathbf{u}'. \tag{64}$$

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 , i.e. we assume that pairs $x_i, x_{i+1},\ i = 1, 3, 5, \dots, 2M-1$. work synchronously. Then we apply the algorithm to find an admissible control law.

Suppose that the parameters in the model here vary as follows

$$\epsilon \in [0, 1] \qquad \omega_i \in [0, \sqrt{1-\epsilon}] \qquad \delta_i^2 + \omega_i^2 = 1 - \epsilon, \tag{65}$$

Then in Fig. 1 synchronization errors occur in the dark grey region, i.e. some product matrices are unstable. The stable region is marked in light grey.

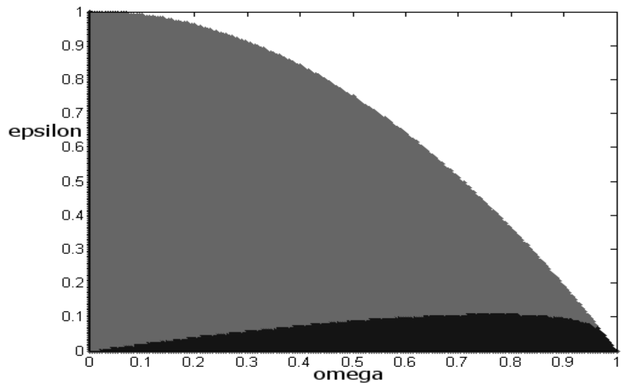

Fig. 1. Unstable region; dark grey, stable region: light grey.

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

$$\delta_i = \delta_j \qquad \omega_i = \omega_j \qquad i, j = 1, 2, \ldots, 6, \tag{66}$$

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.

VII. Conclusions

Synchronization error induced instability of dynamical systems is of particular relevance to networked control systems, sensor-actuator networks, tele-operation systems, and many other network centric applications. This paper has developed a computationally feasible approach to the problem of distributed system stabilization in the presence of such errors. Using state feedback, with LMI based computations, it has been shown how distributed dynamical systems can be stabilized by also making use of the well known polytopic uncertainty description from robust control theory for linear systems. The method developed drastically reduces the complexity of the polytope, i.e. the number of system vertices that need to be stabilized. The effectiveness of the method was illustrated using navigation control in a multi-agent swarm.

References

[1] P. Antsaklis and Tabuada, "Networked embedded sensing and control", NESC Workshop, University of Notre Dame, Notre Dame, October 2005, IN.

[2] H.Sutoh, K. Yamakoshi, and M. Ino ,"Circuits techniques for skew free clock distribution", Proc. IEEE Custom Integrated Circuits pp. 163-166, May 1995

[3] S.I. Liu, J.H. Lee, and H.W. Tsao, "Low power clock de-skew buffer for high speed digital circuits", IEEE J. of Solid State Circuits, vol 34, No 4, p. 554-558, April 1999

[4] D. E. Brueske and S. H. K. Embabi, "A dynamic clock synchronization technique for large scale systems", IEEE Trans. Comp. Package&Manufact. Technol.,vol 17, pp. 350-361, Aug 1994

[5] C. Lorand and P. Bauer, "Distributed discrete time systems with synchronization errors: Models and stability", IEEE Trans on Circ & Syst., CAS-II Express Briefs, Vol. 52, No. 4, pp. 208-213, April 2005

[6] C.Lorand and P. Bauer. "Clock Synchronization Errors in Circuits: Models, stability and fault detection", IEEE Transaction on Circ. and Syst,CASS-I Regular papers, Vol 53, No. 10, October 2006

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

[8] Y. F. Su, A. Bhaya, E. Kaszkurewicz and V.S. Kozyakin, "Further results on stability of asynchronous discrete time linear system", Proc. IEEE Conf. on CDC, pp . 915-920, San diego, CA 1997

[9] C. Lorand, "A Theory of Synchronization Errors", Ph.D Thesis, University of Notre Dame, 2004, Notre Dame, IN.

[10] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan "Linear Matrix Inequalities in System and Control Theory", volume 15 of *SIAM studies in Applied Mathematics*. SIAM, Philadelphia, 1994.

[11] C. A. R. Crusius, A. Trofino Neto, "Sufficient LMI conditions for output feedback control problems", IEEE Trans. Automatic Control, pp 1053 -1057, vol 44, May 1999.