A numerical evaluation of solvers for the Periodic Riccati Differential Equation*

Sergei Gusev[*], Stefan Johansson[†], Bo Kågström[†], Anton Shiriaev[‡], and Andras Varga[§]

UMINF 09.03

[1] Department of Mathematics and Mechanics,
St. Petersburg State University, St. Petersburg, Russia.
gusev@ieee.org

[2] Department of Computing Science,
Umeå University, Umeå, Sweden.
{stefanj,bokg}@cs.umu.se

[3] Department of Applied Physics and Electronics,
Umeå University, Umeå, Sweden.
Anton.Shiriaev@tfe.umu.se

[4] Institute of Robotics and Mechatronics,
German Aerospace Center, DLR, Oberpfaffenhofen, Germany.
Andras.Varga@dlr.de

Abstract

Efficient and robust numerical methods for solving the periodic Riccati differential equation (PRDE) are addressed. Such methods are essential, for example, when deriving feedback controllers for orbital stabilization of underactuated mechanical systems. Two recently proposed methods for solving the PRDE are presented and evaluated on artificial systems and on two stabilization problems originating from mechanical systems with unstable dynamics. The first method is of the type multiple shooting and relies on computing the stable invariant subspace of an associated Hamiltonian system. The stable subspace is determined using algorithms for computing a reordered periodic real Schur form of a cyclic matrix sequence, and a recently proposed method which implicitly constructs a stable subspace from an associated lifted pencil. The second method reformulates the PRDE as a maximization problem where the stabilizing solution is approximated with finite dimensional trigonometric base functions. By doing this reformulation the problem turns into a semidefinite programming problem with linear matrix inequality constraints.

KEY WORDS: Periodic systems, periodic Riccati differential equations, orbital stabilization, periodic real Schur form, periodic eigenvalue reordering, Hamiltonian systems, linear matrix inequality, numerical methods.

*Financial support has been provided by the *Swedish Foundation for Strategic Research* under the frame program grant A3 02:128.

1 Introduction

In this paper, we evaluate numerical methods for solving the *periodic Riccati differential equation* (PRDE) [1, 9, 51]:

$$-\dot{X}(t) = A(t)^T X(t) + X(t)A(t) - X(t)B(t)R(t)^{-1}B(t)^T X(t) + Q(t). \tag{1.1}$$

The PRDE arises for example in the *periodic linear quadratic regulator* (periodic LQR) problem for *periodic linear time-varying* (periodic LTV) systems

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \\ y(t) &= C(t)x(t), \end{aligned} \tag{1.2}$$

where $A(t) \in \mathbb{R}^{n\times n}$, $B(t) \in \mathbb{R}^{n\times m}$, and $C(t) \in \mathbb{R}^{p\times n}$ are continuous T-periodic matrices, i.e., $A(t) = A(t+T)$, $B(t) = B(t+T)$, and $C(t) = C(t+T)$ for all $t \geq 0$. In the last few years, the interest of robust solvers for the PRDE has been strengthened and most of the existing methods for solving the PRDE are unreliable and not suited for systems of high order or with large period. However, recently new methods that better cope with these problems have been proposed. In this paper, we examine two of these methods: the periodic multi-shot method [48, 50] (an invariant subspace approach) and the SDP method (a convex optimization approach) [18]. The two methods are tested and evaluated on a selection of both artificial systems as well as problems arising in experimental setups with periodic behaviors. A third method which is not evaluated in this paper has recently been proposed in [2]. It is an iterative method that approximates the solution of the PRDE with a sequence of PRDEs with a negative semidefinite quadratic term. Future evaluation of this method will show how it compares to the other two methods.

The multi-shot method is based on discretization techniques, which turn a continuous-time problem into an equivalent discrete-time problem [50]. The method is a further development of the one-shot generator method [9, 29, 51]. To solve the PRDE a linear Hamiltonian system must be integrated. The importance of using symplectic integration methods for solving the Hamiltonian system has recently been emphasized in, e.g., [31, 32, 47, 48]. This is also demonstrated by the numerical results in this paper. The solution of the PRDE is computed using two different approaches. The first approach relies on computing the stable invariant subspace of the monodromy matrix associated with the Hamiltonian system using the periodic real Schur form [10, 30] and the reordering of eigenvalues in the periodic real Schur form [23, 24]. The second approach implicitly constructs a stable deflating subspace from an associated lifted pencil using the fast method [50].

The SDP method is based on approximation of the stabilizing solution of the PRDE by finite dimensional trigonometric base functions. By doing this approximation and reformulating the PRDE as a maximization problem, the problem of solving the PRDE is turned into a *semidefinite programming* (SDP) problem with *linear matrix inequality* (LMI) constraints [18].

The paper is organized as follows. In Section 2, we define the periodic LQR problem and the associated PRDE. Section 3 considers invariant subspace approaches, where the main focus is on the periodic multi-shot method. Section 4 presents the convex optimization approach. In Section 5, the multi-shot method and the convex optimization approach are tested and evaluated. We end with some conclusions in Section 6.

2 Linear optimal control

The LQR problem belongs to the class of *linear optimal control* problems which also includes, e.g., *linear quadratic gaussian* (LQG), H_∞ and H_2 optimal control problems. The aim of the methods for solving these optimal control problems is to find a control law for a linear system such that some integral quadratic criteria are minimized. In Section 2.2, we outline the LQR problem, and in Section 2.3 fundamental theory and results for the periodic Riccati differential equation are summarized. We begin with presenting some basic terminology and definitions for periodic LTV systems in Section 2.1.

2.1 Preliminaries

Before addressing the periodic LQR problem, we introduce some fundamental theory and definitions of the periodic LTV system (1.2). For a detailed discussion of periodic systems see, e.g, [1, 9].

Let $\Phi_A(t, t_0)$ be the *transition matrix* associated with $A(t)$ satisfying

$$\frac{\partial}{\partial t}\Phi_A(t, t_0) = A(t)\Phi_A(t, t_0), \quad \Phi_A(t_0, t_0) = I_n.$$

For a T-periodic system, the transition matrix evaluated over one period is known as the *monodromy matrix* $\Psi_A(t_0) = \Phi_A(t_0 + T, t_0)$. The eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\Psi_A(t_0)$ are called the *characteristic multipliers* of $A(t)$ at time t. These eigenvalues are independent of t_0, thus $\Psi_A(t_0)$ has the same spectrum for all t_0. $A(t)$ is said to be a *stable* periodic matrix if all characteristic multipliers are inside the unit circle (open unit disc), i.e., $|\lambda_i| < 1$, for all i.

A characteristic multiplier λ of $A(t)$ is said to be *unreachable* if $\Psi_A(t_0)^T x = \lambda x$, $x \neq 0$, imply that $B(t)^T \Phi_A(t_0, t)^T x = 0$ almost everywhere for $t \in [t_0, t_0 + T]$. Otherwise the characteristic multiplier is said to be *reachable*. The system (1.2) is *stabilizable* if there exists a periodic matrix $K(t)$ such that $A(t) - B(t)K(t)$ is stable, or, equivalently, if all characteristic multipliers λ of $A(t)$ with $|\lambda| \geq 1$ are reachable.

A characteristic multiplier λ of $A(t)$ is said to be *unobservable* if $\Psi_A(t_0)x = \lambda x$, $x \neq 0$, imply that $C(t)\Phi_A(t, t_0)x = 0$ almost everywhere for $t \in [t_0, t_0 + T]$. Otherwise the characteristic multiplier is said to be *observable*. The system (1.2) is *detectable* if there exists a periodic matrix $L(t)$ such that $A(t) - L(t)C(t)$ is stable, or, equivalently, if all characteristic multipliers λ of $A(t)$ with $|\lambda| \geq 1$ are observable.

2.2 The linear quadratic regulator problem

The optimal control problem we consider is to compute a stabilizing controller for the periodic LTV system (1.2). The optimal periodic controller is obtained via solving the LQR problem [3, 41, 44, 51], i.e., by minimizing the *quadratic cost function* for (1.2):

$$\min_{u(t)} \int_0^\infty \left[x(t)^T Q(t) x(t) + u(t)^T R(t) u(t)\right] dt, \tag{2.3}$$

where $Q(t) \in \mathbb{R}^{n\times n}$ and $R(t) \in \mathbb{R}^{m\times m}$ are continuous T-periodic *weighting matrices*, and $Q(t) = Q^T(t) \geq 0$ and $R(t) = R^T(t) > 0$ for all t. The inequality (strict inequality) sign means that a matrix $M \in \mathbb{R}^{n\times n}$ is *positive semidefinite* (*positive definite*), i.e., $z^T M z \geq 0$ ($z^T M z > 0$) for all nonzero $z \in \mathbb{R}^n$. As we will see later, the positive semidefinite assumption on $Q(t)$ is not necessary.

Provided the pair $(A(t), B(t))$ is stabilizable and the pair $(A(t), Q(t)^{1/2})$ is detectable, where $\left(Q(t)^{1/2}\right)^T Q(t)^{1/2} = Q(t)$, the optimal control input $u^*(t)$ that stabilizes (1.2) and minimizes (2.3) is

$$u^*(t) = -K(t)x(t), \quad \text{where} \quad K(t) = R(t)^{-1}B(t)^T X(t). \tag{2.4}$$

The periodic matrix $X(t) \in \mathbb{R}^{n\times n}$ in (2.4) is the unique symmetric positive semidefinite T-periodic stabilizing solution of the continuous-time PRDE (1.1):

$$-\dot{X}(t) = A(t)^T X(t) + X(t)A(t) - X(t)B(t)R(t)^{-1}B(t)^T X(t) + Q(t).$$

The solution $X(t)$ is called a *stabilizing solution* of (1.1) if the closed-loop matrix

$$A(t) - B(t)K(t) = A(t) - B(t)R(t)^{-1}B(t)^T X(t)$$

is stable.

2.3 Periodic Riccati differential equations

Let us first consider the PRDE

$$-\dot{X}(t) = M_{22}(t)X(t) + X(t)M_{11}(t) - X(t)M_{12}(t)X(t) + M_{21}(t), \tag{2.5}$$

where $M_{11}(t) \in \mathbb{R}^{n\times n}$, $M_{12}(t) \in \mathbb{R}^{n\times m}$, $M_{21}(t) \in \mathbb{R}^{m\times n}$, and $M_{22}(t) \in \mathbb{R}^{m\times m}$ are piecewise continuous, locally integrable and T-periodic matrices defined on the interval $[t_0, T]$. A well known result for the PRDE is the relationship with linear systems of differential equations.

Theorem 2.1 [1] *Let $M_{11}(t) \in \mathbb{R}^{n\times n}$, $M_{12}(t) \in \mathbb{R}^{n\times m}$, $M_{21}(t) \in \mathbb{R}^{m\times n}$, and $M_{22}(t) \in \mathbb{R}^{m\times m}$ be T-periodic matrices. Then the following facts hold:*

(i) Let $X(t) \in \mathbb{R}^{m\times n}$ be a solution of (2.5) in the interval $[t_0, T] \subset \mathbb{R}$. If $U(t) \in \mathbb{R}^{n\times n}$ is a solution of the initial value problem

$$\dot{U}(t) = (M_{11}(t) - M_{12}(t)X(t))U(t), \quad U(t_0) = I_n,$$

and $V(t) = X(t)U(t)$, then $\begin{bmatrix} U(t) \\ V(t) \end{bmatrix}$ is a solution of the associated linear system (of differential equations)

$$\frac{d}{dt}\begin{bmatrix} U(t) \\ V(t) \end{bmatrix} = \begin{bmatrix} M_{11}(t) & -M_{12}(t) \\ -M_{21}(t) & -M_{22}(t) \end{bmatrix}\begin{bmatrix} U(t) \\ V(t) \end{bmatrix}. \tag{2.6}$$

(ii) If $\begin{bmatrix} U(t) \\ V(t) \end{bmatrix}$ is a real solution of the system (2.6) such that $U(t) \in \mathbb{R}^{n\times n}$ is regular for $t \in [t_0, T] \subset \mathbb{R}$, then

$$X(t) = V(t)U(t)^{-1},$$

is a real solution of (2.5).

From Theorem 2.1 it follows that the solution of the T-periodic PRDE (1.1) can be computed from the system of differential equations:

$$\frac{d}{dt}\begin{bmatrix} U(t) \\ V(t) \end{bmatrix} = \begin{bmatrix} A(t) & -B(t)R(t)^{-1}B(t)^T \\ -Q(t) & -A(t)^T \end{bmatrix}\begin{bmatrix} U(t) \\ V(t) \end{bmatrix}, \quad \begin{bmatrix} U(t_0) \\ V(t_0) \end{bmatrix} = \begin{bmatrix} U_0 \\ V_0 \end{bmatrix}, \tag{2.7}$$

where $U(t) \in \mathbb{R}^{n\times n}$, $V(t) \in \mathbb{R}^{n\times n}$, and $t_0 \le t \le T$. Provided there exists a solution of (1.1) with initial conditions $X(t_0) = V_0U_0^{-1}$, the matrix $U(t)^{-1}$ exists and the solution of (1.1) is

$$X(t) = V(t)U(t)^{-1}.$$

In the next step, we show that there exists a finite solution to (1.1), i.e., the solution $X(t)$ does not blow up on a finite interval. Define the Riccati operator $\mathcal{R}$ of (1.1) as

$$\begin{aligned}\mathcal{R}(X(t),t) = \dot{X}(t) + A(t)^TX(t) + X(t)A(t)\\ - X(t)B(t)R(t)^{-1}B(t)^TX(t) + Q(t).\end{aligned} \tag{2.8}$$

The following theorem regarding the existence of a stabilizing solution is derived in [1, 14], which is a slight generalization of the theorem in [8]. Note that there is no positive definite assumption on $Q(t)$.

Theorem 2.2 [1, 14] *Suppose that $R(t) = I_m$, $Q(t) = Q(t)^T$, $(A(t), B(t))$ is stabilizable and that there exists a T-periodic stabilizing solution $X(t)$ of the Riccati differential inequality*

$$\mathcal{R}(X(t),t) \ge 0, \quad \forall t \ge 0.$$

Then there exists a T-periodic equilibrium $X_+(t)$ of the PRDE (1.1)*, where*

$$X_+(t) \ge X(t), \quad \forall t \ge 0.$$

In particular, $X_+(t)$ is the maximal *T-periodic stabilizing solution of the PRDE* (1.1)*.*

This theorem is further generalized in [18] to include positive definite time-varying weighting matrices $Q(t)$ and $R(t)$.

Theorem 2.3 [18] *Suppose that the time-varying matrices $A(t)$, $B(t)$, $Q(t)$, $R(t)$ and $R(t)^{-1}$ are bounded, and $Q(t) > 0$ and $R(t) > 0$ for all $t \ge 0$. Let $X_+(t)$ be a stabilizing solution of* (1.1)*. Then, any bounded matrix $X(t)$, satisfying the Riccati differential inequality*

$$\mathcal{R}(X(t),t) \ge 0, \quad \forall t \ge 0, \tag{2.9}$$

also satisfies the inequality

$$X_+(t) \ge X(t), \quad \forall t \ge 0.$$

For a set of matrices $W_j = W_j^T > 0$ and distinct time instances $t_j \ge 0$, where $j = 1, \ldots, N$ and $N \in \mathbb{N}$, define the functional

$$J(X(t)) = \sum_{j=1}^{N} \mathrm{tr}\left(W_jX(t_j)\right), \tag{2.10}$$

where $\mathrm{tr}(A)$ denotes the *trace* of a matrix A. It follows from Theorem 2.3 that a maximum of (2.10) over the set of bounded matrices $X(t)$, satisfying (2.9), is achieved at the stabilizing solution $X_+(t)$ [18].

3 Invariant subspace approaches

In this section, we describe methods based on invariant subspace approaches. The approach we are mainly considering is the periodic multi-shot method proposed in [48, 50]. It is based on the one-shot generator method, described briefly in Section 3.3, and uses methods explicitly designed for computing the invariant subspace of periodic systems: the ordered periodic real Schur form (Section 3.1) or the fast algorithm (Section 3.5). The associated Hamiltonian differential system is solved using symplectic (structure preserving) integration methods, see Section 3.2. The multi-shot method is presented in Section 3.4 and we end by an overview of the MATLAB implementation of the method in Section 3.6.

3.1 Periodic Schur form and reordering

Consider a P-cyclic matrix sequence $A_P, A_{P-1}, \ldots, A_1$ usually associated with the matrix product $A = A_P A_{P-1} \ldots A_1$, where $A_k \in \mathbb{R}^{n\times n}$ and $A_{k+P} = A_k$ for any positive integer k. A common problem is to compute the eigenvalues and/or the corresponding eigenvectors (invariant subspaces) of the matrix product A. For example, to solve the PRDE with the multi-shot method we are interested in the stable periodic invariant subspace of a matrix product. While computing the eigenvalues and invariant subspaces, it is not advisable to explicitly evaluate the matrix product, which is both costly and can lead to significant loss of accuracy and even to under- and overflows [10]. Instead an implicit decomposition of these matrices is used, called the *periodic* (*real*) *Schur form*.

3.1.1 Periodic real Schur form

If real elements in the computed Schur form are required, which is the case for us, the *periodic real Schur form* (PRSF) is used [10, 30].

Let A_k, $k = 1, \ldots, P$, be $n \times n$ real matrices and P-cyclic, i.e., $A_{k+P} = A_k$. Then there exists a P-cyclic orthogonal matrix sequence $Z_k \in \mathbb{R}^{n\times n}$:

$$Z_{k+1}^T A_k Z_k = S_k, \quad k = 1, \ldots, P, \tag{3.11}$$

with $Z_{k+P} = Z_k$ and where one of the S_k matrices, say S_r, is upper quasi-triangular and the remaining are upper triangular. The quasi-triangular matrix S_r has 1×1 and 2×2 blocks on the main diagonal and can appear anywhere in the sequence (typically as S_1 or S_P). The product of the conforming diagonal blocks of the matrix sequence S_k gives the real (1×1 blocks) and complex conjugated pairs (2×2 blocks) of eigenvalues, respectively, of the matrix product $A_P \cdots A_2 A_1$.

3.1.2 Periodic eigenvalue reordering

When computing the PRSF it is not possible to simultaneously specify the order of the eigenvalues of the matrix product $S_P \cdots S_1$. One case when the order is of particular interest is when we are only interested in the invariant subspace corresponding to a specified set of eigenvalues. A direct method for reordering the eigenvalues of a periodic matrix sequence in PRSF, without computing the corresponding matrix product explicitly, is presented in [23].

Let the matrix sequence $S_P, \ldots, S_1$ be in the PRSF (3.11) and assume thet we have q sets of selected eigenvalues. Then there exists an orthogonal matrix sequence $Q_k \in \mathbb{R}^{n\times n}$,

$k = 1, \ldots, P$, such that

$$Q_{k+1}^T S_k Q_k = T_k \equiv \begin{bmatrix} T_{11}^{(k)} & \star & \cdots & \star \\ 0 & T_{22}^{(k)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \star \\ 0 & \cdots & 0 & T_{qq}^{(k)} \end{bmatrix},$$

with $Q_{k+P} = Q_k$ and $T_{k+P} = T_k$, and where the eigenvalues of the matrix product $T_{ii}^{(P)} \cdots T_{ii}^{(1)}$ corresponding to the i-th set of eigenvalues, where $i = 1, \ldots, q$.

In the periodic multi-shot method, we need to compute the ordered periodic real Schur form with

$$T_k = \begin{bmatrix} T_{11}^{(k)} & T_{12}^{(k)} \\ 0 & T_{22}^{(k)} \end{bmatrix}, \quad k = 1, \ldots, P,$$

where $T_{11}^{(k)} \in \mathbb{R}^{p \times p}$ and $T_{22}^{(k)} \in \mathbb{R}^{n-p \times n-p}$, and the matrix product $T_{11}^{(P)} \cdots T_{11}^{(1)}$ has p eigenvalues inside the unit circle, and $T_{22}^{(P)} \cdots T_{22}^{(1)}$ has $n - p$ eigenvalues outside the unit circle[1]. Then the first p columns of the sequence Q_k span the *stable* right periodic invariant subspace, and the last $n - p$ columns span the *unstable* right periodic invariant subspace.

3.2 Hamiltonian systems and symplectic matrices

When solving the PRDE (1.1) with an invariant subspace approach, a linear Hamiltonian system with symplectic flow must be solved. This section gives an introduction to Hamiltonian systems, symplectic matrices and symplectic integration methods. For details, see for example [13, 27, 36].

We first consider an *ordinary differential equation* (ODE) of the form $\dot{y} = f(y)$ with the initial value $y(t_0) = y_0$. The *flow* over time t for an ODE is the mapping φ_t from any initial point y_0 in phase space to a final point $y(t)$ associated with the initial value y_0. Thus, the map φ_t is defined as

$$\varphi_t(y_0) = y(t), \quad y(0) = y_0. \tag{3.12}$$

A (canonical[2]) *Hamiltonian system* is an ODE of the form

$$\begin{aligned} \dot{p} &= \bigtriangledown_q F(p, q), \\ \dot{q} &= -\bigtriangledown_p F(p, q), \end{aligned} \tag{3.13}$$

where $\bigtriangledown_x F(x)$ is the gradient of $F(x)$ with respect to x, p and q are vectors of length d, and $F(p, q) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is an arbitrary (smooth) function called the *Hamiltonian* function (or just the Hamiltonian). Let $y = (p, q)^T$ and (3.13) can be written in compact form as

$$\dot{y} = J^{-1} \bigtriangledown_y F(y), \tag{3.14}$$

where J is the skew-symmetric matrix

$$J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}. \tag{3.15}$$

[1] In finite precision, computed eigenvalues may appear on or close to the boundary of the unit circle.

[2] The term *canonical* for Hamiltonian systems means that the system is of even dimension and J is defined as in (3.15).

We consider the *linear Hamiltonian system* defined by a quadratic Hamiltonian $F(y) = \frac{1}{2}y^T Ly$, where $L \in \mathbb{R}^{2d\times 2d}$ is symmetric. The resulting differential equation is thus

$$\dot{y} = J^{-1}Ly = Hy,$$

where H is a *Hamiltonian matrix* and satisfies $H^T J + JH = 0$.

Next, we consider one of the most important properties of the Hamiltonian system (3.14); symplecticity [27]. Consider a two-dimensional parallelogram lying in $\mathbb{R}^{2d}$. Let the two vectors

$$u = \begin{bmatrix} u_p \\ u_q \end{bmatrix}, \quad \text{and} \quad v = \begin{bmatrix} v_p \\ v_q \end{bmatrix},$$

where $u_p,\ u_q, v_p$ and v_q are in $\mathbb{R}^d$, span the parallelogram

$$\mathcal{P} = \{tu + sv : 0 \le t \le 1, 0 \le s \le 1\}.$$

Denote the (oriented) area of the parallelogram by $\omega(u, v)$, see left illustration in Figure 1.

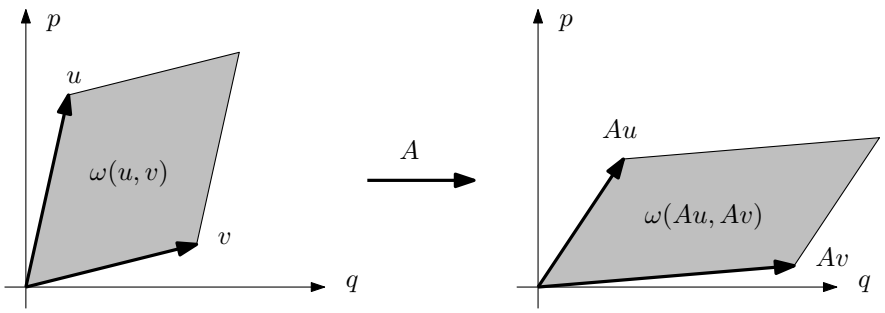


FIGURE 1: A symplectic linear map A is area preserving.

A linear mapping (transformation) $A : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is called *symplectic* if

$$A^T JA = J, \quad \text{where} \quad J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix},$$

or, equivalently, $\omega(Au, Av) = \omega(u, v)$, i.e., the linear mapping A preserves the area $\omega(u, v)$ in phase space, see Figure 1. The matrix $A \in \mathbb{R}^{2d\times 2d}$ is referred to as a *symplectic matrix*. In the case $d = 1$,

$$\omega(u, v) = \det \begin{pmatrix} u_p & v_p \\ u_q & v_q \end{pmatrix} = u_p v_q - u_q v_p.$$

In the general case $d \ge 2$, $\omega(u, v)$ is the sum of oriented areas

$$\omega(u, v) = \sum_{i=1}^{d} \det \begin{pmatrix} u_p^i & v_p^i \\ u_q^i & v_q^i \end{pmatrix} = \sum_{i=1}^{d} \left(u_p^i v_q^i - u_q^i v_p^i\right),$$

where u^i and v^i are projections of u and v, respectively, onto the coordinate planes (p_i, q_i), $i = 1, \ldots, d$. Hence, for $d \ge 2$ symplecticity means that the sum of oriented areas $\omega(u, v)$ of the projections of $\mathcal{P}$ onto (p_i, q_i) is the same as the area of the transformed parallelograms $A(\mathcal{P})$. The area $\omega(u, v)$ is also called the *symplectic two-form* on the phase space $\mathbb{R}^{2d}$ and has in matrix notation the form

$$\omega(u, v) = u^T Jv,$$

where J is the matrix (3.15).

For a Hamiltonian system the following holds, e.g., see [36].

Theorem 3.1 *The flow map φ_t of a Hamiltonian system* (3.14) *is symplectic.*

To preserve the symplectic characteristic of the Hamiltonian system (3.14) an integrator that preserves the symplectic flow of the problem must be used. One example of a symplectic one-step integration method is the symplectic and symmetric Gauss Runge-Kutta method [27, 28, 40], which is used in Section 5. It is an implicit Runge-Kutta method with fixed time steps where the nonlinear system is solved using fixed-point iteration.

We now turn our attention towards the PRDE (1.1) associated with the periodic LTV system (1.2) [9, 51]. Let $H(t) \in \mathbb{R}^{2n\times 2n}$ be the periodic time-varying Hamiltonian matrix

$$H(t) = \begin{bmatrix} A(t) & -B(t)R(t)^{-1}B(t)^T \\ -Q(t) & -A(t)^T \end{bmatrix},$$

i.e., $H(t)$ satisfies $H(t)^T J + JH(t) = 0$ for all t, where J is defined by (3.15). From the initial value problem

$$\frac{\partial}{\partial t}\Phi_H(t,t_0) = H(t)\Phi_H(t,t_0), \quad \Phi_H(t_0,t_0) = I_{2n}, \tag{3.16}$$

the *transition matrix* $\Phi_H(t,t_0)$ associated with $H(t)$ is computed. The system (3.16) is a linear Hamiltonian system where the transition matrix $\Phi_H(t,t_0)$ for all $t > t_0$ has eigenvalues symmetric with respect to the unit circle and is symplectic. We recall from Section 2.1, that for a periodic system the transition matrix evaluated over one period is known as the monodromy matrix $\Psi_H(t_0) = \Phi_H(t_0+T,t_0)$.

3.3 One-shot generator method

The one-shot generator method solves the Hamiltonian system (3.16) over one period T and computes the stabilizing solution of the PRDE (1.1) from the stable invariant subspace of the solution [9, 29, 51]. The method is outlined in Algorithm 1.

Algorithm 1

1. Compute the monodromy matrix $\Psi(t_0) = \Phi(t_0+T,t_0)$ by solving the initial value problem (3.16) over one period.
2. Compute the ordered real Schur form of $\Psi(t_0)$ [22]:

$$\begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}^T \Psi(t_0) \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix},$$

   where $S_{11} \in \mathbb{R}^{n\times n}$ is upper quasi-triangular with n eigenvalues inside the unit circle, and $S_{22} \in \mathbb{R}^{n\times n}$ is upper quasi-triangular with n eigenvalues outside the unit circle. Then the stable subspace of $\Psi(t_0)$ is spanned by the columns of the $2n \times n$ matrix

$$\begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix}.$$

3. Solve the matrix differential equation

$$\dot{Y}(t) = H(t)Y(t), \quad Y(t_0) = \begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix}, \tag{3.17}$$

by integrating from $t = t_0$ to $t = t_0 + T$.

4. Partition the solution of (3.17) into $n \times n$ blocks as

$$Y(t) = \begin{bmatrix} Y_1(t) \\ Y_2(t) \end{bmatrix}.$$

Then the solution of the PRDE is

$$X(t) = Y_2(t)Y_1(t)^{-1}, \quad t \in [t_0, t_0 + T].$$

As discussed in [50], this method has some major disadvantages and is potentially numerically unreliable. In steps 1 and 3, and ODEs with unstable dynamics are solved. For systems with large periods these steps will result in a significant accumulation of roundoff errors. The second ODE also depends on the first one and consequently they must be solved in sequence. Moreover, if a non-symplectic solver is used there will also be a drift in the solution of the linear Hamiltonian system. Special symplectic solvers for periodic (stiff) problems are considered in [17].

3.4 Multi-shot method

The alternative multi-shot method [48, 50] reduces the impact of the numerical problems which occurs with the one-shot method. The main idea is to turn the continuous-time problem into an equivalent discrete-time problem. This is achieved by considering the following product form of the monodromy matrix $\Psi_H(t_0)$ with $t_0 = 0$:

$$\Psi_H(0) \equiv \Phi_H(T, 0) = \Phi_H(T, T - \Delta) \cdots \Phi_H(2\Delta, \Delta)\Phi_H(\Delta, 0), \tag{3.18}$$

where $\Delta = T/N$ for a suitable integer N[3]. In the following, denote $\Phi_k = \Phi_H(k\Delta, (k-1)\Delta)$, $k = N, \ldots, 1$. Notably, $\Phi_N, \ldots, \Phi_1$ is an N-cyclic matrix sequence of $2n \times 2n$ matrices. The linear Hamiltonian system (3.16) can now be integrated for each transition matrix Φ_k, and methods for periodic eigenvalue problems can be used to compute the stable subspace.

The consequence is that the multi-shot method has several advantages compared to the one-shot method:

(i) The linear Hamiltonian system, which has unstable dynamics, is solved over short subparts of the period. This makes the method more reliable for problems with large periods.

(ii) Only one ODE (in a multi-shot fashion) must be solved, in contrast to the one-shot method where two ODEs must be solved in sequence.

(iii) The system's periodicity is exploited, by explicitly using methods designed for periodic systems.

[3] In practice, the constant N can be chosen such that the discrete-time solutions of $X(t)$, at $t = T(k-1)/N$, coincide with the sampling time of the stabilizing controller.

(iv) The numerical integration of the Hamiltonian system can easily be parallelized. This is of great value since this part can be very computational intensive.

(v) Since the integration of the Hamiltonian system is done over short subparts, the importance of using a symplectic solver is not critical.

In the absence of parallelization, the only disadvantage of the multi-shot method compared to the one-shot method is that it is more time consuming.

The multi-shot method is presented in Algorithm 2. For high accurate solution, the Hamiltonian system in step 1 is preferably solved with a symplectic solver like the symplectic Gauss Runge-Kutta [27, 28], but as pointed out in item (v) above this is not always necessary.

ALGORITHM 2

1. Compute the transition matrices $\Phi_N, \ldots, \Phi_2, \Phi_1$ by solving the linear Hamiltonian system (3.16) for each interval $[(k-1)\Delta, k\Delta]$, where $k = N, N-1, \ldots, 1$.
2. Compute the periodic real Schur form associated with the matrix product $\Psi_H(0) = \Phi_N \cdots \Phi_2 \Phi_1$:

$$Z_{k+1}^T \Phi_k Z_k = S_k, \quad k = N, N-1, \ldots, 1, \tag{3.19}$$

   with $Z_{N+1} = Z_1$ and S_1 upper quasi-triangular.
3. Reorder the periodic real Schur form such that

$$Q_{k+1}^T S_k Q_k = \begin{bmatrix} T_{11}^{(k)} & T_{12}^{(k)} \\ 0 & T_{22}^{(k)} \end{bmatrix}, \quad k = N, N-1, \ldots, 1, \tag{3.20}$$

   with $Q_{N+1} = Q_1$ and where the matrix product $T_{11}^{(N)} \cdots T_{11}^{(1)}$ has n eigenvalues inside the unit circle, and $T_{22}^{(N)} \cdots T_{22}^{(1)}$ has n eigenvalues outside the unit circle. Here, Q_k is the sequence of orthogonal transformation matrices that perform the eigenvalue reordering of the PRSF (3.19).
4. For each k, partition the product of the transformation matrices from (3.19) and (3.20) into four $n \times n$ blocks as

$$Z_k Q_k = \begin{bmatrix} Y_{11}^{(k)} & Y_{12}^{(k)} \\ Y_{21}^{(k)} & Y_{22}^{(k)} \end{bmatrix}.$$

   Then the solution of the PRDE $X_k \equiv X(t)$ at $t = (k-1)\Delta$, $k = N, N-1, \ldots, 1$, is

$$X_k = Y_{21}^{(k)} (Y_{11}^{(k)})^{-1}.$$

To acquire the solution of the PRDE between two discretization moments $t_0 = (k-1)\Delta$ and $t_f = k\Delta$, the methods described in [15, 16] can be used to integrate the PRDE (1.1) in backward time with $X(t_f) = X_{k+1}$.

3.5 Fast algorithm

An alternative approach to the ordered PRSF to compute the stable subspace is the *fast algorithm* proposed in [50], which is an extension of the *swapping and collapsing* approach [6, 7] for discrete-time algebraic Riccati equations. Notably, for generalized periodic matrix sequences this method is not suitable, since it includes computing inverses of presumptive ill-conditioned matrices. However, in our case this method only performs numerically robust operations.

Provided that the transition matrices $\Phi_N, \ldots, \Phi_1$ are computed as in step 1 of Algorithm 2, the fast algorithm implicitly constructs a stable deflating subspace from an associated lifted pencil. The approach takes advantage of that the solution $X(t)$ of two successive time steps $(k-1)\Delta$ and $k\Delta$ are related as (e.g., see [9])

$$X_k = \left(X_{k+1}\Phi_{12}^{(k)} - \Phi_{22}^{(k)}\right)^{-1}\left(\Phi_{21}^{(k)} - X_{k+1}\Phi_{11}^{(k)}\right), \tag{3.21}$$

where $k = 1, \ldots, N$ and Φ_k is partitioned in $n \times n$ blocks

$$\Phi_k = \begin{bmatrix} \Phi_{11}^{(k)} & \Phi_{12}^{(k)} \\ \Phi_{21}^{(k)} & \Phi_{22}^{(k)} \end{bmatrix}.$$

Define the *associated lifted pencil* to the periodic matrix pair (Φ_k, I_{2n}):

$$S - zT = \begin{bmatrix} \Phi_1 & -I_{2n} & 0 & \cdots & 0 \\ 0 & \Phi_2 & -I_{2n} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & & & \Phi_{N-1} & -I_{2n} \\ -zI_{2n} & 0 & \cdots & 0 & \Phi_N \end{bmatrix}. \tag{3.22}$$

If a solution of the PRDE exists, the matrix pencil $S - zT$ is regular and has no eigenvalues on the unit circle. Using an orthogonal transformation matrix U_k, compress the rows of the matrix $\begin{bmatrix} -I_{2n} \\ \Phi_{k+1} \end{bmatrix}$ to $\begin{bmatrix} R_k \\ 0 \end{bmatrix}$, where R_k is a nonsingular matrix. Applying this recursively to (3.22) transforms the matrix pencil $S - zT$ to the reduced pencil

$$\widetilde{S} - z\widetilde{T} = \left[\begin{array}{c|cccc} \widetilde{\Phi}_1 & R_1 & -U_{12}^{(1)} & 0 & \cdots & 0 \\ \widetilde{\Phi}_2 & 0 & R_2 & -U_{12}^{(2)} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ \vdots & \vdots & & & & -U_{12}^{(N-2)} \\ \widetilde{\Phi}_{N-1} - zU_{12}^{(N-1)} & 0 & & \cdots & 0 & R_{N-1} \\ \hline \widetilde{\Phi}_N - zU_{22}^{(N-1)} & 0 & & \cdots & & 0 \end{array}\right],$$

where the $n \times n$ matrix $U_{ij}^{(k)}$ is the ij-th block of the matrix U_k, and the regular matrix pencil $\widetilde{\Phi}_N - zU_{22}^{(N-1)}$ contains all finite eigenvalues of $S - zT$.

The initial solution X_1 to (3.21) is computed from an ordered generalized Schur decomposition

$$Q^T(\widetilde{\Phi}_N - zU_{22}^{(N-1)})Z = \begin{bmatrix} \hat{S}_{11} - z\hat{T}_{11} & \hat{S}_{12} - z\hat{T}_{12} \\ 0 & \hat{S}_{22} - z\hat{T}_{22} \end{bmatrix},$$

where Q and Z are orthogonal matrices, and the upper quasi-triangular matrix pencil $\hat{S}_{11} - z\hat{T}_{11}$ has only finite eigenvalues inside the unit circle. If Z is partitioned as

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix},$$

then $X_1 = Z_{21}Z_{11}^{-1}$. The remaining solutions X_k for $k = N, \ldots, 2$ are computed iteratively using (3.21), with $X_{N+1} = X_1$. As pointed out in [50], the main advantage of this method is its ease of implementation, since only standard robust numerical routines are used.

3.6 The MATLAB implementations

To compute the stable subspace, the multi-shot method uses Fortran subroutines for computing the PRSF [39] and periodic eigenvalue reordering [23] (to be available in the upcoming *PEP toolbox* [25]). The fast multi-shot method is available in the *Periodic System Toolbox* for MATLAB [49]. To solve the linear Hamiltonian system, builtin ODE solvers in MATLAB and a MATLAB implementation of the symplectic Gauss Runge-Kutta are used. See, e.g., [26, 38], for other possible ODE solvers.

4 A convex optimization approach

A second approach to solve the PRDE (1.1) is based on convex optimization. By reformulating the PRDE as a convex optimization problem the solution can be obtained by solving a semidefinite programming problem with linear matrix inequality constraints, see Section 4.1. This method is proposed by Gusev et.al. [18] and an improved version of it is presented in Section 4.2.

4.1 Semidefinite programming and linear matrix inequalities

A *semidefinite programming* (SDP) problem is a special class of convex optimization problems and has the form [5, 11, 12, 33]:

$$\begin{aligned} &\min\ c^T x, \\ &\text{s.t.}\ \ F(x) \geq 0, \end{aligned} \tag{4.23}$$

where

$$F(x) = F_0 + \sum_{j=1}^{n} x_j F_j.$$

For the SDP problem (4.23), $x \in \mathbb{R}^n$ is the variable vector and the vector $c \in \mathbb{R}^n$ and the symmetric matrices $F_0, \ldots, F_n \in \mathbb{R}^{m \times m}$ are given. The constraint $F(x) \geq 0$ is called a *linear matrix inequality* (LMI). Multiple LMIs $F^{(1)}(x) \geq 0, \ldots, F^{(p)} \geq 0$ can be formulated as a single LMI as $\mathrm{diag}(F^{(1)}(x), \ldots, F^{(p)}(x)) \geq 0$.

4.2 Reformulation of the PRDE

As stated in the end of Section 2.3, the maximal stabilizing solution $X_+(t)$ of the PRDE (1.1) is achieved by maximizing the cost function J in (2.10). This is an infinite dimensional SDP problem, for which the linear functional J is maximized over a convex set of matrices $X(t)$, satisfying the inequality (2.9). To solve this infinite dimensional optimization problem

we first approximate it with a sequence of finite dimensional problems. This approach to solve the PRDE is proposed by Gusev et.al. in [18]. The first outline of the method had a smaller system of inequalities but depended on a larger number of parameters.

From the Schur complement[4], it follows that the Riccati inequality (2.9) associated with the T-periodic PRDE (1.1) can be reformulated as the LMI

$$\mathcal{S}(X(t),t) \geq 0, \tag{4.24}$$

where

$$\mathcal{S}(X(t),t) = \begin{bmatrix} \dot{X}(t) + A(t)^T X(t) + X(t)A(t) + Q(t) & X(t)B(t) \\ B(t)^T X(t) & R(t) \end{bmatrix}, \ \forall t \geq 0,$$

and $\dot{X}(t) + X(t)A(t) + A(t)^T X(t) + Q(t)$ is symmetric. The next step is to approximate the stabilizing solution and its derivative by finite dimensional trigonometric base functions. These base functions can be chosen such that the characteristics of the underlying system is emphasized. A suitable (general) base function is the finite dimensional Fourier expansion, which is used in the following. Let $T = 2\pi/\omega$ and $q \geq 1$, $q \in \mathbb{N}$, then

$$\widetilde{X}(t) = \sum_{k=-q}^{q} e^{\mathrm{i}k\omega t} X_k, \text{ and} \tag{4.25}$$

$$\frac{d\widetilde{X}(t)}{dt} = \sum_{k=-q}^{q} \mathrm{i}k\omega \ e^{\mathrm{i}k\omega t} X_k, \tag{4.26}$$

with the matrices $X_{-k} = \overline{X}_k, \ k = 1, \ldots, q$. Consequently, $\mathcal{S}(X(t),t)$ can be approximated by $\mathcal{S}_j = \mathcal{S}(\widetilde{X}(t_j), t_j)$, where $0 \leq t_j \leq T$ is some time instances and $j = 1, \ldots, N$ for a suitable integer N.

We can now formulate the finite dimensional SDP problem as

$$\begin{aligned} &\min \ \ -J(\widetilde{X}(t)), \\ &\text{s.t.} \ \ \mathcal{S}_j \geq 0, \quad j = 1, \ldots, N, \end{aligned} \tag{4.27}$$

where the objective function is obtained from (2.10) and (4.25):

$$J(\widetilde{X}(t)) = \sum_{j=1}^{N} \operatorname{tr} \left(W_j \sum_{k=-q}^{q} e^{\mathrm{i}k\omega t_j} X_k \right). \tag{4.28}$$

Note that minimizing $-J(\widetilde{X}(t))$ is equivalent to maximizing $J(\widetilde{X}(t))$. By solving (4.27) an approximate stabilizing solution $\widetilde{X}_+(t)$ of (1.1) can be computed, where

$$\lim_{q\to\infty} \widetilde{X}_+(t) = \lim_{q\to\infty} \sum_{k=-q}^{q} e^{\mathrm{i}k\omega t_j} X_k = X_+(t),$$

for all $t \in [t_0, T]$.

Let a general SDP problem have n_{SDP} variables and an $m_{\text{SDP}} \times m_{\text{SDP}}$ LMI constraint matrix. Then the global worst-case complexity for a dense SDP problem is $O(m_{\text{SDP}}^{6.5} \log \epsilon^{-1})$, where ϵ is the desired accuracy, and $n_{\text{SDP}} = O(m_{\text{SDP}}^2)$ is assumed. In practice, the complexity is much lower. For the SDP problem (4.27) we have $(2q+1)(n(n+1)/2)$ variables and the $(n+m)N \times (n+m)N$ matrix $\operatorname{diag}(\mathcal{S}_1, \ldots, \mathcal{S}_N)$ forms the LMI constraints.

[4]The Schur complement of the block D of the matrix $\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$ is $A - BD^{-1}C$ [52].

4.3 The MATLAB implementation

The MATLAB implementation by Gusev [18] uses *SeDuMi* [43, 46] (a MATLAB toolbox for optimization over symmetric cones) to solve the LMI problem and *YALMIP* [37] for modeling the optimization problem. Default options are used both for SeDuMi and YALMIP. In the MATLAB implementation, the weight matrices $W_1, \ldots, W_N$ in (4.28) are set to the identity matrix. However, if necessary these matrices could be changed to improve the numerical stability of the SDP problem.

5 Numerical experiments

We have implemented, evaluated and compared three PRDE solvers for both artificially constructed periodic LTV systems with known solutions and stabilization problems originating from two experimental control systems, the *Furuta pendulum* and *pendulums on carts*. The three methods used are the multi-shot method using the ordered PRSF, the multi-shot method using the fast algorithm and the SDP method. The corresponding solvers are in the following called the *multi-shot solver*, the *fast multi-shot solver*, and the *SDP solver*, respectively.

For the two multi-shot solvers we have solved the linear Hamiltonian system (3.16) using three different ODE solvers: The two general purpose MATLAB ODE solvers `ode45` (Dormand-Prince Runge-Kutta $(4, 5)$) and `ode113` (variable order Adams-Bashforth-Moulton PECE), and `sgrk` a MATLAB implementation of the symplectic 6-stage (order 12) Gauss Runge-Kutta method, with fixed time steps). For `ode45` and `ode113` we have used the relative tolerance 10^{-9} and the absolute tolerance 10^{-16}. For `sgrk` we have used an initial value of 4 time steps, and if no convergence in the fixed-point iteration is achieved the time steps are doubled until convergence or 64 time steps are reached.

For the SDP solver we have used default options for both SeDuMi and YALMIP. The best results from the SDP solver have a relative error in the solution around 10^{-11}.

When nothing else is stated, the number of time instances N in the product of the transition matrices (3.18) in the multi-shot method is set to $N = 100$. We have based our choice of N on the results in [32, 50]. For consistency, the number of time instances N of the LMI constraints in (4.27) is set to $N = 100$. Moreover, the stabilizing solution and its derivative are approximated with the finite dimensional Fourier expansion as in (4.25) and (4.26), with $q = 10$.

The implementations of the three PRDE solvers have been done in MATLAB, utilizing built-in functions and gateways to existing Fortran subroutines. All computations were carried out in double precision ($\epsilon_{\text{mach}} = 2.2 \cdot 10^{-16}$) on an Intel Core Duo T7200 (2GHz) with 2GB memory, running Windows XP[5] and MATLAB[6] R2006b.

5.1 A set of artificial systems

In the first set of examples, we have investigated how the solvers manage to compute an accurate solution of the PRDE associated with artificial LTV systems with respect to the number of states and the periodicity of the systems. All the artificial systems have known solutions, called the *reference solutions*, and they are constructed as follows.

Consider an LTI system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t_0) = x_0, \tag{5.29}$$

[5] Windows is a registered trademark of Microsoft Corporation.
[6] MATLAB and Simulink are registered trademarks of The MathWorks, Inc.

with n states (as we will see, must be a multiple of two) and m inputs, i.e., $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. In addition, the quadratic cost function of the LTI system is

$$J = \int_0^\infty \left[x^T Q x + u^T R u\right] dt,$$

resulting in the optimal feedback control

$$u^*(t) = -Kx(t), \text{ where } K = R^{-1}B^T X. \tag{5.30}$$

For LTI systems, the matrix X in the optimal feedback control (5.30) is obtained by solving the algebraic Riccati equation (ARE)

$$A^T X + XA - XBR^{-1}B^T X + Q = 0. \tag{5.31}$$

To solve (5.31) an existing stable solver is used [4, 35], e.g., care in MATLAB or preferably slcaresc in Slicot [45].

Next, the LTI system (5.29) is transformed into a periodic LTV system by change of coordinates

$$z(t) = P(t)x(t), \tag{5.32}$$

where $z(t)$ is the state vector in the new coordinates and

$$P(t) = \begin{bmatrix} R(t) & & \\ & \ddots & \\ & & R(t) \end{bmatrix}, \quad \text{with} \quad R(t) = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix},$$

for a given $\omega > 0$. Notably, the number of states in $x(t)$ must be a multiple of two. After differentiating both sides of (5.32) we get

$$\begin{aligned} \dot{z}(t) &= \frac{dP(t)}{dt}x(t) + P(t)\dot{x}(t) \\ &= \frac{dP(t)}{dt}P(t)^{-1}z(t) + P(t)\left(AP(t)^{-1}z(t) + Bu(t)\right). \end{aligned}$$

This results in the T-periodic LTV system

$$\dot{z}(t) = \widetilde{A}(t)z(t) + \widetilde{B}(t)u(t), \tag{5.33}$$

where

$$\begin{aligned} \widetilde{A}(t) &= \frac{dP(t)}{dt}P(t)^{-1} + P(t)AP(t)^{-1}, \text{ and} \\ \widetilde{B}(t) &= P(t)B, \end{aligned}$$

with period $T = 2\pi/\omega$. The cost function (2.3) for the resulting transformed system (5.33) has the weighting matrices $Q(t) = P(t)^{-T}QP(t)^{-1}$ and $R(t) = R$. The optimal feedback of (5.33) can now be expressed as

$$\begin{aligned} u^*(t) &= -K(t)z(t) \\ &= -R^{-1}\widetilde{B}(t)^T\widetilde{X}(t)z(t), \end{aligned}$$

where $\widetilde{X}(t) \equiv \widetilde{X}_k$ is the computed solution of the PRDE (1.1) at $t = (k-1)T/N$. The solution $\overline{X}(t) = P(t)^{-T} X P(t)^{-1}$, where X is the solution of (5.31), corresponds to the exact solution at time t (our *reference solution*).

The accuracy of the computed solution $\widetilde{X}(t)$ is evaluated using the *relative error* e_{rel} of the PRDE solution with respect to the reference solution $\overline{X}(t)$, computed as

$$e_{\text{rel}} = \sum_{k=1}^{N} \left(\frac{\|\widetilde{X}_k - \overline{X}_k\|_F}{\|\overline{X}_k\|_F} \right) / N,$$

where $\overline{X}_k = \overline{X}((k-1)T/N)$. We have tested if the computed solution is a stabilizing solution by first approximating $\widetilde{X}_k$ by a finite dimensional Fourier expansion (like in (4.25)). The LQR problem is then tested on a closed loop linear system in Simulink[6].

We have based our tests on two different LTI systems, which are transformed into periodic LTV systems. We have only considered the weighting matrices $Q = I_n$ and $R = I_m$, for both cases. The different system matrices are as follows:

$$A = \begin{bmatrix} * & * & \cdots & * \\ 0 & * & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * \end{bmatrix}, \qquad B = \begin{bmatrix} * & * \\ \vdots & \vdots \\ * & * \end{bmatrix}, \tag{5.34}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ & & & 0 & 1 \\ 0 & \cdots & & 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{5.35}$$

where $*$ are uniformly distributed random numbers on the interval $[-10, 10]$. The system matrices (5.35) correspond to an LTI system with n integrators connected in series with a feedback controller applied to the nth system [35] (see also [34]).

5.1.1 Size of system

First the methods are tested with respect to the number of states n of the system (order of the system). How does the number of states affect the computed solution and how large systems can the different methods solve? We have run the tests on the system matrices (5.34) and (5.35). In both cases we used $\omega = 2$, so the period of the resulting LTV system (5.33) is $T = \pi$.

First we have examined which ODE solver of `ode45`, `ode113`, and `sgrk`, is best suited for solving the linear Hamiltonian system (3.16) in the multi-shot method. As we see in Table 1, when the number of states of the system increases the run-time for `ode45` increases rapidly while for `ode113` and `sgrk` the run-time increases linearly. Considering that the three solvers produce solutions with almost the same relative error, the preferred solver should either be `ode113` or `sgrk`. For this example, we have chosen to use `sgrk` as it is marginally more accurate and faster than `ode113`.

Next we have compared the run-time of the two approaches to compute the stable subspace in the multi-shot method: the fast algorithm and the ordered PRSF. The test is

TABLE 1: The accuracy of the PRDE solution using the multi-shot method with different ODE solvers. The results are shown for increasing size n of the periodic LTV system. First table shows the results for system matrices (5.34), and the second for system matrices (5.35).

	Relative error (run-time [sec])					
n	ode45		ode113		sgrk	
4	$6.2 \cdot 10^{-15}$	(8.7)	$8.7 \cdot 10^{-14}$	(5.5)	$2.7 \cdot 10^{-15}$	(4.7)
10	$7.1 \cdot 10^{-11}$	(20.0)	$1.0 \cdot 10^{-10}$	(8.5)	$6.3 \cdot 10^{-11}$	(7.7)
16	$4.3 \cdot 10^{-6}$	(43.5)	$4.4 \cdot 10^{-6}$	(10.5)	$3.6 \cdot 10^{-6}$	(9.9)
20	$1.4 \cdot 10^{-8}$	(93.0)	$2.4 \cdot 10^{-8}$	(19.2)	$1.2 \cdot 10^{-8}$	(21.1)
26	$1.8 \cdot 10^{-5}$	(214.1)	$4.5 \cdot 10^{-5}$	(28.1)	$7.6 \cdot 10^{-6}$	(25.1)

n	ode45		ode113		sgrk	
4	$3.6 \cdot 10^{-14}$	(6.2)	$3.1 \cdot 10^{-14}$	(4.6)	$6.1 \cdot 10^{-14}$	(3.5)
10	$4.1 \cdot 10^{-11}$	(8.4)	$3.2 \cdot 10^{-11}$	(5.6)	$4.8 \cdot 10^{-11}$	(5.3)
16	$1.3 \cdot 10^{-8}$	(11.7)	$2.6 \cdot 10^{-8}$	(7.8)	$2.0 \cdot 10^{-8}$	(5.7)
20	$3.0 \cdot 10^{-6}$	(19.9)	$3.7 \cdot 10^{-6}$	(11.5)	$3.1 \cdot 10^{-6}$	(10.8)
26	$3.0 \cdot 10^{-3}$	(31.5)	$2.8 \cdot 10^{-3}$	(16.0)	$2.1 \cdot 10^{-3}$	(13.1)

TABLE 2: The run-time (in seconds) of the fast algorithm and the ordered PRSF.

n	4	10	16	20	26	30
Ordered PRSF	0.051	0.073	0.17	0.30	0.54	0.81
Fast algorithm	0.028	0.032	0.064	0.14	0.22	0.30

TABLE 3: The solvers tested with respect to the size n of the periodic LTV system using system matrices (5.34). N.S. denotes that the computed solution is not stabilizing.

	Relative error (run-time [sec])					
n	Multi-shot		Fast multi-shot		SDP	
4	$1.9 \cdot 10^{-15}$	(4.5)	$3.0 \cdot 10^{-15}$	(4.6)	$2.7 \cdot 10^{-11}$	(10.4)
10	$4.4 \cdot 10^{-12}$	(7.8)	$4.7 \cdot 10^{-12}$	(7.9)	$6.9 \cdot 10^{-12}$	(240)
16	$3.7 \cdot 10^{-11}$	(9.1)	$1.3 \cdot 10^{-9}$	(9.2)	$4.6 \cdot 10^{-10}$	(3137)
20	$9.7 \cdot 10^{-11}$	(18.1)	$2.5 \cdot 10^{-9}$	(19.1)	Out of memory	(−)
26	$1.0 \cdot 10^{-7}$	(25.5)	$2.9 \cdot 10^{-6}$	(23.8)	Out of memory	(−)
30	$4.9 \cdot 10^{-5}$	(25.1)	N.S.	(27.2)	Out of memory	(−)
36	N.S.	(38.6)	N.S.	(38.5)	Out of memory	(−)

TABLE 4: The solvers tested with respect to the size n of the periodic LTV system using system matrices (5.35). N.S. denotes that the computed solution is not stabilizing.

	Relative error (run-time [sec])					
n	Multi-shot		Fast multi-shot		SDP	
4	$6.1 \cdot 10^{-14}$	(3.3)	$5.1 \cdot 10^{-15}$	(3.4)	$4.5 \cdot 10^{-11}$	(9.1)
10	$4.8 \cdot 10^{-11}$	(4.9)	$1.7 \cdot 10^{-12}$	(5.3)	$4.5 \cdot 10^{-11}$	(128)
16	$2.0 \cdot 10^{-8}$	(6.2)	$6.4 \cdot 10^{-9}$	(6.1)	Fail	(−)
20	$3.1 \cdot 10^{-6}$	(10.4)	$7.6 \cdot 10^{-8}$	(11.9)	Out of memory	(−)
26	$2.1 \cdot 10^{-3}$	(13.8)	$5.7 \cdot 10^{-4}$	(13.4)	Out of memory	(−)
30	$1.7 \cdot 10^{-1}$	(15.7)	$5.1 \cdot 10^{-2}$	(15.5)	Out of memory	(−)
36	N.S.	(22.5)	N.S.	(22.8)	Out of memory	(−)

run on the system matrices (5.35). In Table 2, we see that the fast algorithm is faster than the ordered PRSF. Theoretically the two approaches have comparable complexity of $O(N(2n)^3)$, but the fast algorithm better utilizes so-called level 3 BLAS operations [50]. However, for both multi-shot solvers the time it takes to compute the stable subspace is negligible compared to the time it takes to solve the Hamiltonian system.

We now solve the two systems with the three PRDE solvers. The results are displayed in Tables 3 and 4. For both systems, the SDP solver runs out of memory for systems with more than 16 states. The reason is the high number of variables together with the high dimension of the LMI constraints, which for a system with 20 states and 2 inputs are 2856 and 1700, respectively. Moreover, for (5.35) with 16 states the objective function for the SDP problem is unbounded and therefore the solver fails. As we see, the run-time for the SDP solver also increases rapidly together with the size.

The accuracy and run-time for the two multi-shot methods are comparable. However, the multi-shot and fast multi-shot methods do not compute a stabilizing solution for (5.34) with $n \geq 36$ and $n \geq 30$, respectively. For (5.35), the two multi-shot methods compute a stabilizing solution up to 30 states, after that the solution is not stabilizing.

5.1.2 Periodicity

In the second test, the solvers are evaluated on periodic LTV systems with different periods T. As above, the periodic LTV systems tested are constructed from the system matrices (5.34) and (5.35), respectively, where $A \in \mathbb{R}^{4\times 4}$. The ODE solvers `ode113` and `sgrk` have been used with the two multi-shot solvers. Moreover, as the period T is increased the constant N in (3.18) and (4.27) has been chosen as:

Period T	2π	$2\pi \cdot 10$	$2\pi \cdot 10^2$	$2\pi \cdot 10^3$	$2\pi \cdot 10^4$	$2\pi \cdot 10^5$
N	100	100	100	1000	1000	10000
$\Delta = T/N$	0.063	0.63	6.3	6.3	63	63

The results from the two multi-shot solvers are not completely consistent, see Tables 5 and 6. Consider the results when the `sgrk` solver is used. In the first example (Table 5), the multi-shot solvers have a rather low accuracy already at a period of $2\pi \cdot 10^2$ and they fail to compute any solution when the period reaches $2\pi \cdot 10^4$. However, in the second example they still compute a solution with high accuracy at a period of $2\pi \cdot 10^5$ (see Table 6). These results have not been analyzed in detail, but one cause of the poor results in the first example is the large gap in the eigenvalues of the transition matrices. That the two multi-shot solvers fail even earlier when `ode113` is used, indicates the importance of a symplectic ODE solver

TABLE 5: The solvers tested with respect to the period T of the periodic LTV system using system matrices (5.34).

	Relative error (run-time [sec])					
Period	Multi-shot, ode113		Multi-shot, sgrk		SDP	
2π	$1.3 \cdot 10^{-12}$	(11.3)	$3.9 \cdot 10^{-15}$	(6.1)	$2.1 \cdot 10^{-10}$	(11.8)
$2\pi \cdot 10$	$1.6 \cdot 10^{-11}$	(9.9)	$7.7 \cdot 10^{-10}$	(21.2)	$2.6 \cdot 10^{-11}$	(10.6)
$2\pi \cdot 10^2$	Fail	(−)	$3.3 \cdot 10^{-5}$	(81.5)	$5.2 \cdot 10^{-11}$	(10.7)
$2\pi \cdot 10^3$	Fail	(−)	$3.1 \cdot 10^{-6}$	(814)	$2.3 \cdot 10^{-11}$	(88.9)
$2\pi \cdot 10^4$	Fail	(−)	Fail	(−)	$2.3 \cdot 10^{-11}$	(88.5)
$2\pi \cdot 10^5$	Fail	(−)	Fail	(−)	Out of memory	(−)

Period	Fast multi-shot, ode113		Fast multi-shot, sgrk	
2π	$1.3 \cdot 10^{-12}$	(5.2)	$5.6 \cdot 10^{-15}$	(6.1)
$2\pi \cdot 10$	$2.0 \cdot 10^{-11}$	(9.5)	$7.7 \cdot 10^{-10}$	(21.7)
$2\pi \cdot 10^2$	Fail	(−)	$3.4 \cdot 10^{-5}$	(82.9)
$2\pi \cdot 10^3$	Fail	(−)	$3.1 \cdot 10^{-6}$	(851)
$2\pi \cdot 10^4$	Fail	(−)	Fail	(−)
$2\pi \cdot 10^5$	Fail	(−)	Fail	(−)

TABLE 6: The solvers tested with respect to the period T of the periodic LTV system using system matrices (5.35).

	Relative error (run-time [sec])					
Period	Multi-shot, ode113		Multi-shot, sgrk		SDP	
2π	$1.5 \cdot 10^{-14}$	(3.7)	$4.2 \cdot 10^{-15}$	(3.5)	$6.7 \cdot 10^{-11}$	(8.3)
$2\pi \cdot 10$	$1.5 \cdot 10^{-13}$	(4.6)	$1.4 \cdot 10^{-15}$	(4.6)	$8.4 \cdot 10^{-11}$	(8.2)
$2\pi \cdot 10^2$	$1.1 \cdot 10^{-12}$	(7.9)	$2.2 \cdot 10^{-11}$	(19.1)	$4.0 \cdot 10^{-11}$	(8.5)
$2\pi \cdot 10^3$	$5.1 \cdot 10^{-13}$	(77.2)	$2.1 \cdot 10^{-12}$	(186)	$1.5 \cdot 10^{-11}$	(79.8)
$2\pi \cdot 10^4$	Fail	(−)	$6.1 \cdot 10^{-14}$	(1591)	$1.5 \cdot 10^{-11}$	(80.8)
$2\pi \cdot 10^5$	Fail	(−)	$2.0 \cdot 10^{-14}$	(15684)	Out of memory	(−)

Period	Fast multi-shot, ode113		Fast multi-shot, sgrk	
2π	$1.4 \cdot 10^{-14}$	(3.8)	$2.4 \cdot 10^{-15}$	(3.6)
$2\pi \cdot 10$	$1.5 \cdot 10^{-13}$	(4.6)	$5.3 \cdot 10^{-15}$	(5.5)
$2\pi \cdot 10^2$	$1.5 \cdot 10^{-12}$	(8.1)	$2.2 \cdot 10^{-11}$	(18.9)
$2\pi \cdot 10^3$	$6.6 \cdot 10^{-13}$	(101)	$2.2 \cdot 10^{-12}$	(213)
$2\pi \cdot 10^4$	Fail	(−)	$9.3 \cdot 10^{-14}$	(1629)
$2\pi \cdot 10^5$	Fail	(−)	$5.3 \cdot 10^{-13}$	(18236)

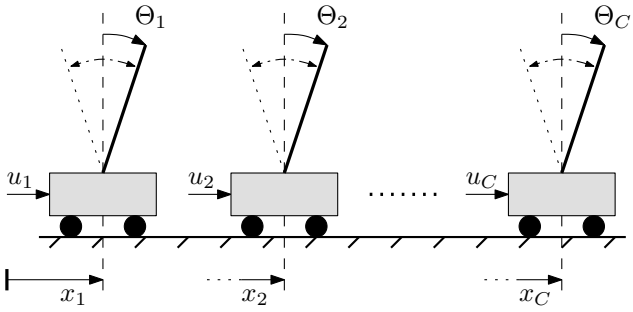


FIGURE 2: C identical Pendulum-cart systems. The coordinates $x_1, \ldots, x_C$ represent positions of the carts along the horizontal axis, $\theta_1, \ldots, \theta_C$ are the angles of the pendulums with respect to the vertical axis, and $u_1, \ldots, u_C$ are the control inputs.

for problems with large periodicity. We also observe that for the problems with large period (and large N), the fast algorithm is slower than the ordered PRSF.

For the SDP solver the results are much more consistent. The run-time for the SDP solver is only depending on the choice of N and the size of the problem, it is not affected by the period of the system. For both examples, the solver has no problem up to a period of $2\pi \cdot 10^4$. After that the size of the LMI constraints gets too big as the number of time instances N is increased to 10000, and the SDP solver runs out of memory. At this point, the number of variables are of moderate size 210, but the LMI constraints are of dimension 6000.

5.2 Examples of orbital stabilization of cycles for mechanical systems

All three solvers have also been used for deriving feedback controllers for orbital stabilization of non-trivial periodic solutions for two mechanical systems, where the first one can have an arbitrary large number of degrees of freedom, and the second one can have a cycle of arbitrary large period. Here nonlinear controllers are constructed based on linear ones found by stabilizing transverse dynamics of the systems along cycles.

5.2.1 Synchronization of oscillations of C-copies of pendulums on carts

The first example is stable synchronization of oscillations of C-copies of identical pendulum-cart systems around their unstable equilibriums[7], see Figure 2. Assuming that for each system the masses of the cart and the pendulum are 1 [kg], and the distance from the suspension to the center of mass of the pendulum is 1 [m], the dynamics have the form

$$2 \cdot \ddot{x}_i + \cos(\theta_i) \cdot \ddot{\theta}_i - \sin(\theta_i) \cdot \dot{\theta}_i^2 = u_i, \text{ and} \tag{5.36}$$

$$\cos(\theta_i) \cdot \ddot{x}_i + \ddot{\theta}_i - g \cdot \sin(\theta_i) = 0, \quad i = 1, \ldots, C. \tag{5.37}$$

The system has $2C$-degrees of freedom and C control variables.

[7]The steps for planning motion and analytical arguments for controller design are from [19].

Planning a cycle: Suppose the C^2-smooth function[8] $\phi(\cdot)$ is chosen such that the invariance of the relations

$$x_1 = \phi(\theta_1), \quad x_2 = \phi(\theta_2), \quad \ldots, \quad x_C = \phi(\theta_n), \tag{5.38}$$

results in C identical equations with $\theta = \theta_i$, $i = 1, \ldots, C$,

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = 0,$$

having a T-periodic solution[9] $\theta_\star(t) = \theta_\star(t+T)$. Here

$$\begin{aligned} \alpha(\theta) &= \cos(\theta) \cdot \phi'(\theta) - 1, \\ \beta(\theta) &= \cos(\theta) \cdot \phi''(\theta), \text{ and} \\ \gamma(\theta) &= -g \cdot \sin(\theta). \end{aligned}$$

The solutions written in pairs for all systems

$$\left[\theta_1 = \theta_\star(t), x_1 = \phi\left(\theta_\star(t)\right)\right], \quad \ldots, \quad \left[\theta_C = \theta_\star(t), x_C = \phi\left(\theta_\star(t)\right)\right], \tag{5.39}$$

are the synchronous oscillations of all C pendulums on carts.

Orbital stabilization of (5.39) can be achieved from a stabilization of the linearization of the transverse dynamics (5.37) along the cycle (5.39). Introducing new coordinates $[\theta, y^T]^T = [\theta, y_1, \ldots, y_{2C-1}]^T$ by the relations:

$$\begin{aligned} \theta &= \theta_1, \\ y_i &= x_i - \phi(\theta), \ i = 1, \ldots, C, \\ y_{C+j} &= \theta - \theta_{j+1}, \ j = 1, \ldots, C-1, \end{aligned} \tag{5.40}$$

one can check that the dynamics of (5.37) can be rewritten in the form

$$\begin{aligned} \alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) &= -\cos\theta \cdot v_1, \\ \ddot{y}_i &= v_i, \quad i = 1, \ldots, C, \\ \ddot{y}_{C+j} &= F_{C+j}(\cdot) + G_{(C+j),1}(\cdot)v_1 \\ &\quad + G_{(C+j),(j+1)}(\cdot)v_{j+1}, \quad j = 1, \ldots, C-1, \end{aligned} \tag{5.41}$$

where $G_{(n+1),1}(\cdot) = \cdots = G_{(2n-1),1}(\cdot) = -\frac{\cos\theta}{\alpha(\theta)}$, and for $j = 1, \ldots, C-1$,

$$\begin{aligned} F_{C+j}(\cdot) &= \frac{\beta\left(\theta - y_{C+j}\right)\left(\dot{\theta} - \dot{y}_{C+j}\right)^2 + \gamma\left(\theta - y_{C+j}\right)}{\alpha\left(\theta - y_{C+j}\right)} - \frac{\beta\left(\theta\right)\dot{\theta}^2 + \gamma\left(\theta\right)}{\alpha\left(\theta\right)}, \\ G_{(C+j),(j+1)}(\cdot) &= \frac{\cos\left(\theta - y_{C+j}\right)}{\alpha\left(\theta - y_{C+j}\right)}, \end{aligned}$$

and where the feedback transform from the original control variables $[u_1, \ldots, u_C]$ to $[v_1, \ldots, v_C]$ has been uniquely defined by the following targeted equations

$$\ddot{x}_1 - \phi''(\theta_1)\dot{\theta}_1^2 - \phi'(\theta_1)\ddot{\theta}_1 = v_1, \quad \ldots, \quad \ddot{x}_C - \phi''(\theta_C)\dot{\theta}_C^2 - \phi'(\theta_C)\ddot{\theta}_C = v_C.$$

[8] A continuous function is called a C^2-smooth function if the first and second derivatives exist and are continuous.

[9] The way to plan a cycle for one cart-pendulum system and to make it then orbitally stable is described in [42].

Transverse coordinates $x_\perp$ for (5.37) along the solution

$$\theta = \theta_\star(t), \quad y_1(t) = y_2(t) = \cdots = y_{(2C-1)}(t) = 0, \quad v_1 = v_2 = \cdots = v_C = 0,$$

of (5.41) are defined by $x_\perp = [I, y^T, \dot{y}^T]^T$ with (5.40) and

$$I\left(\theta(t), \dot{\theta}(t), \theta_\star(0), \dot{\theta}_\star(0)\right) = \dot{\theta}^2(t) - e^{\left\{-\int_{\theta_\star(0)}^{\theta(t)} \frac{2\cos(\tau)\cdot\phi''(\tau)}{\cos(\tau)\cdot\phi'(\tau)-1} d\tau\right\}} \left[\dot{\theta}_\star(0)\right]^2 -$$

$$- \int_{\theta_\star(0)}^{\theta(t)} e^{\left\{\int_{\theta(t)}^{s} \frac{2\cos(\tau)\cdot\phi''(\tau)}{\cos(\tau)\cdot\phi'(\tau)-1} d\tau\right\}} \frac{2\,g\cdot\sin(s)}{\cos(s)\cdot\phi'(s) - 1} ds. \tag{5.42}$$

The coefficients of the linearization of dynamics for transverse coordinates $x_\perp$ can be computed as follows

$$\frac{d}{dt}\begin{bmatrix} I_\bullet \\ Y_{1\bullet} \\ Y_{2\bullet} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11}(t) & 0 & 0 \\ 0 & 0 & I_{2C-1} \\ 0 & A_{22}(t) & A_{23}(t) \end{bmatrix}}_{\mathcal{A}(t)} \begin{bmatrix} I_\bullet \\ Y_{1\bullet} \\ Y_{2\bullet} \end{bmatrix} + \underbrace{\begin{bmatrix} b_{11}(t) & 0 & 0 \\ 0 & 0 & 0 \\ B_{21}(t) & \cdots & B_{2C}(t) \end{bmatrix}}_{\mathcal{B}(t)} \begin{bmatrix} V_{1\bullet} \\ \vdots \\ V_{C\bullet} \end{bmatrix} \tag{5.43}$$

where $a_{11}(t) = -\frac{2\beta(\theta_\star(t))\dot{\theta}_\star(t)}{\alpha(\theta_\star(t))}$, $b_{11}(t) = -\frac{2\cos(\theta_\star(t))\dot{\theta}_\star(t)}{\alpha(\theta_\star(t))}$, and

$$A_{22} = \mathrm{diag}\{\underbrace{0,\ldots,0}_{C}, \underbrace{a(t),\ldots,a(t)}_{C-1}\}, \quad A_{23} = \mathrm{diag}\{\underbrace{0,\ldots,0}_{C}, \underbrace{b(t),\ldots,b(t)}_{C-1}\},$$

$$\begin{aligned} B_{21} &= [\underbrace{1,0,0,\ldots,0}_{C}, \underbrace{c(t),c(t),\ldots,c(t)}_{C-1}]^T, \\ B_{22} &= [\underbrace{0,1,0,\ldots,0}_{C}, \underbrace{-c(t),0,0,\ldots,0}_{C-1}]^T, \\ B_{23} &= [\underbrace{0,0,1,\ldots,0}_{C}, \underbrace{0,-c(t),0,\ldots,0}_{C-1}]^T, \\ &\vdots \\ B_{2C} &= [\underbrace{0,0,\ldots,0,1}_{C}, \underbrace{0,\ldots,0,-c(t)}_{C-1}]^T, \end{aligned}$$

with $b(t) = a_{11}(t)$, $c(t) = -\frac{\cos(\theta_\star(t))}{\alpha(\theta_\star(t))}$, and

$$a(t) = \frac{\left[\beta(\theta_\star(t))\dot{\theta}_\star^2(t) + \gamma(\theta_\star(t))\right]\alpha'(\theta_\star(t))}{\alpha^2(\theta_\star(t))} - \frac{\beta'(\theta_\star(t))\dot{\theta}_\star^2(t) + \gamma'(\theta_\star(t))}{\alpha(\theta_\star(t))}.$$

For this example the linear system (5.43) has $4C-1$ states and only C control inputs. As argued in [42], the function $\phi(\cdot)$ in (5.38) can be chosen to meet various specifications on a periodic motion, e.g., its period, amplitude etc. For instance, with the choice

$$\phi(\theta) = -\left[1 + \frac{g}{\omega^2}\right]\cdot\log\left(\frac{1+\sin\theta}{\cos\theta}\right) \tag{5.44}$$

there are oscillations of each of the cart-pendulum systems around their unstable equilibria of period $T \approx 2\pi/\omega$.

By solving the PRDE associated with the LTV system (5.43) we can find a stabilizing solution that synchronizes the oscillations of the pendulums and carts. For the PRDE, we have used the constant weighting matrix $R = I_C$ and the $(4C-1) \times (4C-1)$ time-varying weighting matrix

$$Q(t) = \begin{bmatrix} f_*(t) & 0 & \cdots & & 0 \\ 0 & 3 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & 0 \\ 0 & \cdots & & 0 & 3 \end{bmatrix},$$

where

$$f_*(t) = \frac{10}{\sqrt{\dot{\theta}(t)^2 + \ddot{\theta}(t)^2}}.$$

Using the multi-shot method, we have successfully computed a stabilizing solution for 40 carts (system with 159 states and 40 control inputs) and with the fast multi-shot method for 50 carts (system with 199 states and 50 control inputs). Figures 3 and 4 show the simulation of the closed loop nonlinear system[10] for 40 carts simulated over 50 seconds with the target trajectory of the period $T \approx 5$ [sec] and the amplitude 0.2 [rad]. The initial states of the pendulums and carts are chosen randomly in vicinity of the tangent orbit. We have not run into any numerical problems with the solvers and we believe that a stabilizing solution could be computed for a much higher number of carts. However, the memory is a limit of how high we can increase the number of states of the system, e.g., in the case of the multi-shot solver we ran out of memory when solving for 50 carts.

The SDP solver, however, could only compute a stabilizing solution for three carts. For four carts the system has already 15 states and 4 control inputs, and inevitably the SDP solver runs out of memory.

5.2.2 Orbital stabilization of Furuta pendulum

The Furuta pendulum is a mechanical system with two degrees of freedom (see Figure 5), where ϕ denotes the angle of the arm rotating in the horizontal plane, and θ is the angle of the pendulum attached to the end of the arm. The arm is directly actuated by a *DC*-motor, while the pendulum can freely rotate in the vertical plane perpendicular to the arm. Its behavior is controlled through mechanical coupling with the dynamics of the arm, i.e. through an acceleration of the arm.

The equations of motion of the Furuta pendulum are, [20]:

$$(p_1 + p_2 \sin^2\theta)\ddot{\phi} + p_3 \cos\theta\ddot{\theta} + 2p_2 \sin\theta\cos\theta\dot{\theta}\dot{\phi} - p_3 \sin\theta\dot{\theta}^2 = \tau_\phi, \tag{5.45}$$

$$p_3 \cos\theta\ddot{\phi} + (p_2 + p_5)\ddot{\theta} - p_2 \sin\theta\cos\theta\dot{\phi}^2 - p_4 \sin\theta = 0, \tag{5.46}$$

where τ_ϕ is the external torque that allows us to control the rotation of the arm. The constants p_1–p_5 are positive and defined by physical parameters of the system. For the

[10] See [19] for a nonlinear feedback design for (5.37) based on stabilization of transverse linearization (5.43).

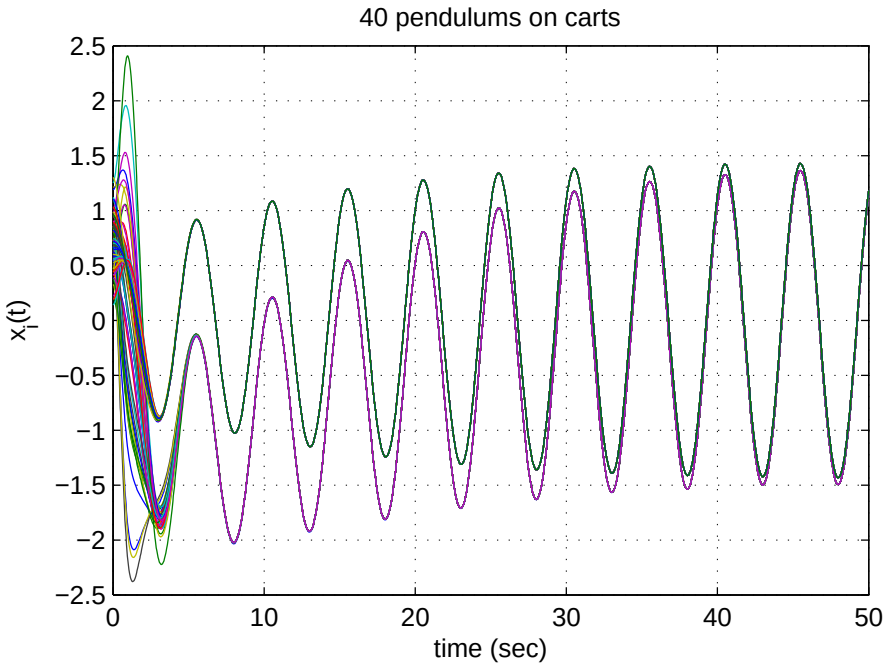


FIGURE 3: Simulation of 40 pendulums on carts. The coordinates x_i, $i = 1, \ldots, 40$, are the positions of the carts along the horizontal axis.

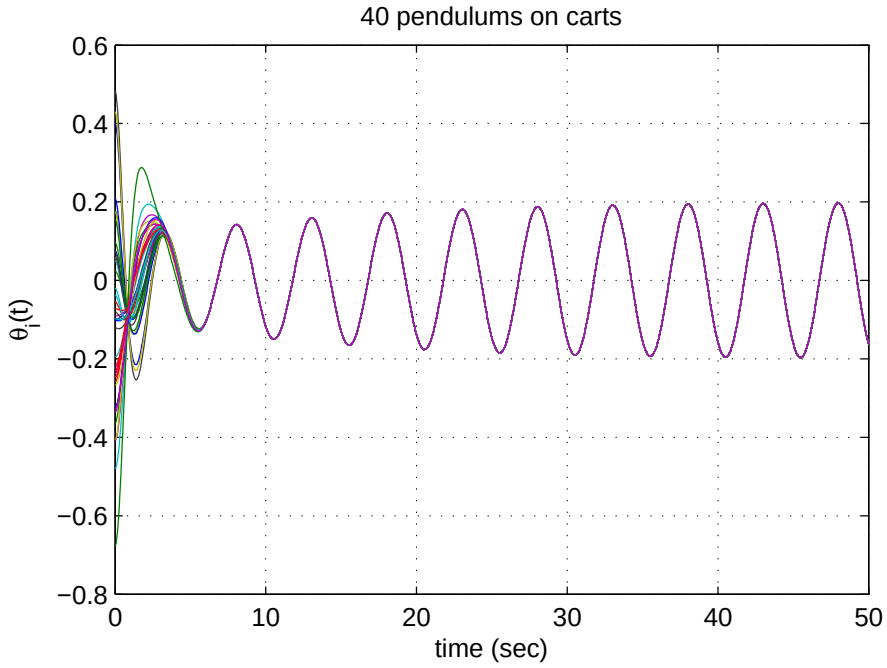


FIGURE 4: Simulation of 40 pendulums on carts. The angles θ_i, $i = 1, \ldots, 40$, are the angles of the pendulums with respect to the vertical axis.

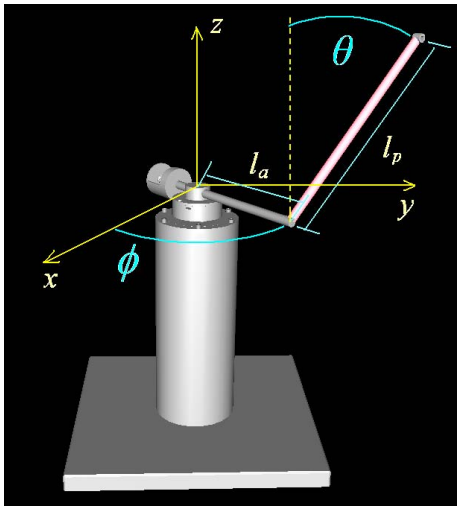


FIGURE 5: An illustration of the Furuta pendulum built at Department of Applied Physics and Electronics, Umeå University.

present setup[11] they are:

$$p_1 = 1.8777 \cdot 10^{-3}, \quad p_2 = 1.3122 \cdot 10^{-3},$$
$$p_3 = 9.0675 \cdot 10^{-4}, \quad p_4 = 5.9301 \cdot 10^{-2}, \text{ and } \quad p_5 = 1.77 \cdot 10^{-4}.$$

Swinging up the Furuta pendulum is a classical problem for this mechatronic device. One solution to this problem is based on an idea of orbital stabilization of homoclinic curves of the pendulum (the passive link). If successful, such design ensures that the solutions of the closed loop system will visit any neighborhood of the upright unstable equilibrium infinitely many times, and where controllers can be switched to achieve local stabilization of this equilibrium.

An extension of this idea is suggested in [21], where constructive conditions are proposed for presence of periodic motions (cycles) of the Furuta pendulum that are located arbitrary close to some homoclinic curves. In addition, it is described how to plan a family of these homoclinic curves of the pendulum and steps for orbital stabilization of periodic cycles are outlined. Closeness of a found family of cycles to homoclinic curves imply that their periods grow without bound if initial conditions of these cycles are chosen to approach the homoclinic curves. An example of planning such cycles and orbital stabilization is presented below.

As shown in [21], if one defines the geometrical relations

$$\phi = K \cdot \arctan(\theta), \text{ and } \quad 0.01 \times \frac{p_2 + p_5}{p_3} < K < 6.6 \times \frac{p_2 + p_5}{p_3}, \tag{5.47}$$

between the angles of the Furuta pendulum invariant by a control variable τ_ϕ, the upright equilibrium will have a pair of homoclinic curves surrounded by a family of periodic solutions filling their neighborhood along a 2-d sub-manifold defined by (5.47). The phase portrait of the θ-variable on this sub-manifold is shown in Figure 6.

Orbital stabilization of any such newly shaped periodic solution can be achieved via stabilization of Furuta pendulum's transverse dynamics defined for each cycle. Let us rewrite

[11] Built at Department of Applied Physics and Electronics, Umeå University.

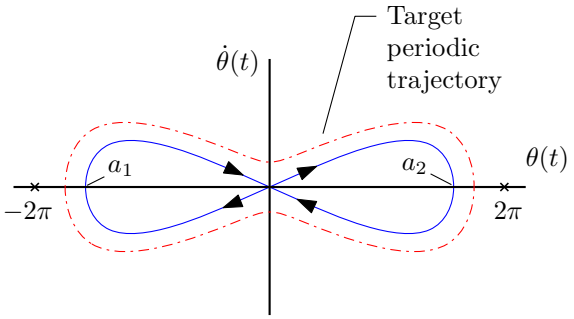


FIGURE 6: Two homoclinic curves of the equilibrium at $\theta = 0$ are shown on the phase portrait. One intersects the $\dot\theta$-axis at a_1 and the other at a_2. The dashed line illustrates one example of a periodic trajectory orbiting the two homoclinic curves. [21]

the Furuta pendulum dynamics in new coordinates as

$$\alpha(\theta)\ddot\theta + \beta(\theta)\dot\theta^2 + \gamma(\theta) = p_2 \sin\theta\cos\theta\dot y^2 + \frac{2Kp_2\sin\theta\cos\theta}{1+\theta^2}\dot y\dot\theta - p_3\cos\theta v,$$
$$\ddot y = v.$$

Here the variable y and the control signal v are defined by the relations

$$y = \phi - K\cdot\arctan(\theta), \quad v = \ddot\phi - \frac{K}{1+\theta^2}\ddot\theta + \frac{2K\theta}{(1+\theta^2)^2}\dot\theta^2, \quad \gamma(\theta) = -p_4\sin\theta,$$

and the functions

$$\alpha(\theta) = p_2 + p_5 + \frac{Kp_3\cos(\theta)}{(1+\theta^2)}, \text{ and } \quad \beta(\theta) = \frac{-K\cos\theta}{(1+\theta^2)^2}\left(2p_3\theta + Kp_2\sin\theta\right).$$

The periodic motions of the Furuta pendulum consistent with the constraint (5.47) will be cycles of the dynamical system

$$\alpha(\theta)\ddot\theta + \beta(\theta)\dot\theta^2 + \gamma(\theta) = 0.$$

The linearization of transverse dynamics along any such nontrivial periodic motion

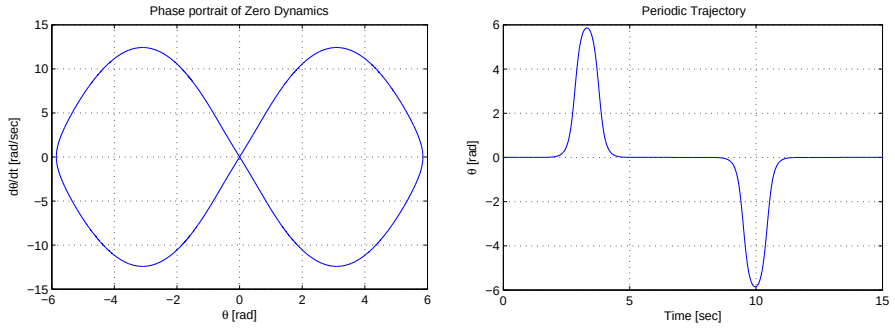
$$\theta_\star(t) = \theta_\star(t+T), \quad \phi_\star(t) = K\cdot\arctan(\theta_\star(t)),$$

has the form

$$\frac{d}{dt}\begin{bmatrix} I_\bullet \\ Y_{1\bullet} \\ Y_{2\bullet}\end{bmatrix} = \underbrace{\begin{bmatrix} a_{11}(t) & 0 & a_{13}(t) \\ 0 & 0 & 1 \\ 0 & 0 & 0\end{bmatrix}}_{=A(t)}\begin{bmatrix} I_\bullet \\ Y_{1\bullet} \\ Y_{2\bullet}\end{bmatrix} + \underbrace{\begin{bmatrix} b_1(t) \\ 0 \\ 1\end{bmatrix}}_{=B(t)} V_\bullet, \tag{5.48}$$

where the T-periodic coefficients of $A(\cdot)$ and $B(\cdot)$ are

$$a_{11}(t) = -\frac{2\cdot\dot\theta_\star(t)\cdot\beta(\theta_\star(t))}{\alpha(\theta_\star(t))},$$
$$a_{13}(t) = \frac{4Kp_2\cdot\sin\theta_\star(t)\cdot\cos\theta_\star(t)\cdot\dot\theta_\star^2(t)}{(1+\theta_\star^2(t))\cdot\alpha(\theta_\star(t))},$$
$$b_1(t) = -\frac{2\cdot\dot\theta_\star(t)\cdot p_3\cos(\theta_\star(t))}{\alpha(\theta_\star(t))}.$$



(a) (b)

FIGURE 7: (a) Phase portrait of the Furuta pendulum with period 13.32 seconds. (b) Periodic trajectory of θ as a function of time.

By solving the PRDE associated with the LTV system (5.48) we can find an orbital stabilizing solution corresponding to a particular periodic motion of the Furuta pendulum. For the PRDE, we have used the constant weighting matrix $R = 10$ and the time-varying weighting matrix

$$Q(t) = \begin{bmatrix} f_*(t) & 0 & 0 \\ 0 & \overline{f}_*(t) & 0 \\ 0 & 0 & 0.01 \end{bmatrix},$$

where

$$f_*(t) = 0.05\sqrt{\dot{\theta}(t)^2 + \ddot{\theta}(t)^2},$$

and $\overline{f}_*(t)$ is the mean of $f_*(t)$.

By using the one-shot method to solve the PRDE, a periodic trajectory with the period $T \approx 4.0454$ seconds has successfully been stabilized on a physical set-up of the Furuta pendulum. We here show that a stabilizing solution with a period of at least $T \approx 8.095$ seconds can be found by using one of the proposed solvers.

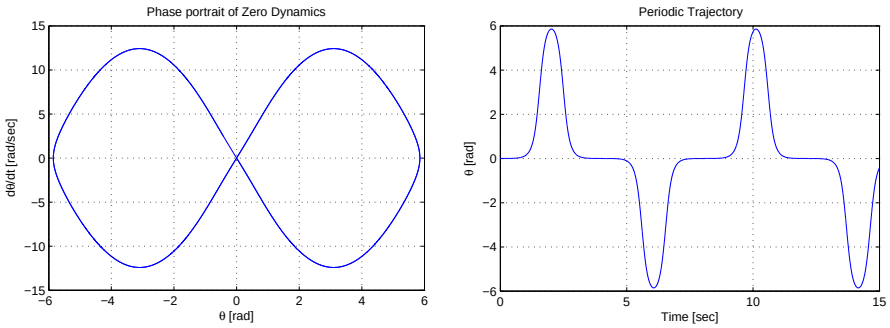
As mentioned above, a desired target orbit can be obtained by choosing the initial conditions such that they approach the homoclinic curves:

$$\phi(0) = 0, \quad \dot{\phi}(0) = 0, \quad \theta(0) = 0, \text{ and } \quad \dot{\theta}(0) = \epsilon,$$

where $\epsilon \neq 0$. By choosing ϵ close to zero we can, theoretically, find an orbit with an arbitrary large period T. However, numerically it is not possible to chose the initial states too close to the upright equilibrium ($\theta(0) = 0$ and $\dot{\theta}(0) = 0$), since at some point the accuracy of the numerical methods will reach its limits[12].

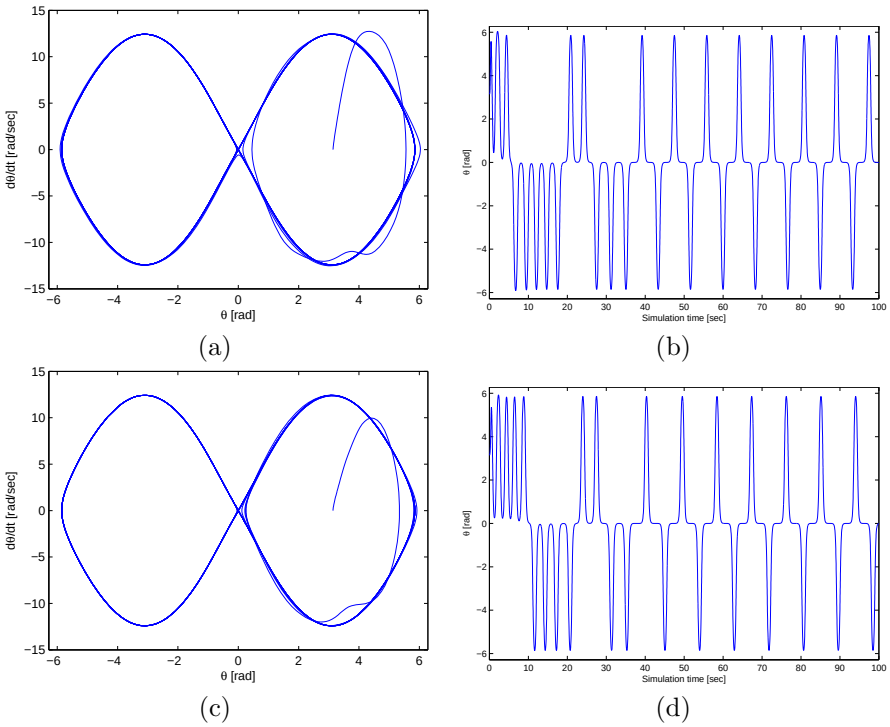
Numerically, we have successfully found a periodic motion trajectory of the pendulum with the initial condition $\dot{\theta}(0) = 3 \cdot 10^{-6}$, which has a period of $T \approx 13.32$ seconds. Figure 7(a) shows the phase portrait of this trajectory and Figure 7(b) the periodic trajectory of the angle θ as a function of time. However, for this period the computed solutions of the PRDE are not orbital stabilizing the nonlinear Simulink model of the Furuta pendulum.

[12]For the physical setup there is also a limit of how large period we can get.This limit will (usually) occur before the numerical methods reaches their limits.



(a) (b)

FIGURE 8: (a) Phase portrait of the Furuta pendulum with period 8.095 seconds. (b) Periodic trajectory of θ as a function of time.



(a) (b)

(c) (d)

FIGURE 9: The resulting phase portrait and periodic trajectory of θ from simulation of orbital stabilizing solutions with a periodic trajectory of 8.095 seconds: (a)-(b) With the PRDE solution from the SDP solver. (c)-(d) With the PRDE solution from the multi-shot solver.

Using any of the three PRDE solvers, the largest period for which we can find a orbital stabilizing solution for is $T \approx 8.095$ seconds (with the initial condition $\dot{\theta}(0) = 0.005$). Figure 8 shows desired periodic cycle of the Furuta pendulum, and Figure 9 displays the results from the simulation using the PRDE solutions from the SDP solver and the multi-shot solver (the fast-multi solver give similar results as the multi-shot solver). As we can see the results are similar but not identical.

6 Summary of test results and future work

We do not see any significant differences between the two multi-shot solvers. The two solvers are comparable both with respect to run-time and accuracy. As we showed in Section 5.1.2, for some cases the run-time for the multi-shot solver can even be shorter than for the *fast* multi-shot solver. For solving the underlying Hamiltonian system the preferred ODE solver (of those tested) is a symplectic solver like the symplectic Gauss Runge-Kutta. It is especially important to use a symplectic solver for systems with large periods.

From the test results of the artificial systems we can see an indication of that if the SDP solver can solve the PRDE, the computed solution is of high accuracy. This is in contrast to the multi-shot solvers which compute solutions of various degrees of accuracy.

One major limitation of the SDP solver is the high storage requirement, which depends on the size of the system. Even if the memory usage can be reduced, the test results clearly show that the memory issue is a significant drawback for the SDP solver. This disadvantage can be critical, for example, when the PRDE system must be solved online in a physical setup. If the system is of small size ($n \lesssim 15$) this will not be of any problem, however, real-world applications usually have a high degree of freedoms which leads to medium- to large-sized problems. We remark that we have only used SeDuMi to solve the SDP problem (4.27).

When solving the PRDE online the run-time also becomes an important factor. Generally, any of the two multi-shot solvers are significant faster than the SDP solver. However, one case when the SDP solver is faster is for small-sized systems with large periods. The run-time for the SDP solver is independent of the period. Therefore, this solver is to be preferred for such systems as long as the memory requirement is satisfied. The two multi-shot solvers get a longer run-time and for some cases also lower accuracy when the period of the system increases.

In summary, the test results show that for small-sized problems ($n \lesssim 15$) with large periods the SDP solver is a good choice. For medium-sized problems ($15 \lesssim n \lesssim 500$) the two multi-shot solvers have a shorter run-time and require less memory.

One question we still do not have a clear answer to is; Do there exist some types of problems that can be solved with the SDP solver but not with the multi-shot solvers, and vice versa?

Future work includes a parallel implementation of the SDP solver and the multi-shot solver. The solvers will also be tested on physical setups and on systems with a large number of degrees of freedoms.

Acknowledgements

We want to thank Ernst Hairer for providing us with a Fortran implementation of the Gauss Runge-Kutta method, which our MATLAB solver `sgrk` is based on.

References

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank. *Matrix Riccati Equations: in Control and Systems Theory*. Birkhäuser, Basel, Switzerland, 2003. ISBN 3-7643-0085-X.

[2] B. Anderson and Y. Feng. An iterative algorithm to solve periodic Riccati differential equations with an indefinite quadratic term. In *Proc. of the 47th IEEE Conference on Decision and Control, CDC'08*, Cancun, Mexico, 2008.

[3] B. Anderson and J. Moore. *Optimal Control: Linear Quadratic Methods*. Dover, 2007. ISBN 0486457664.

[4] W. Arnold and A. Laub. Generalized eigenproblem algorithms and software for algebraic Riccati equations. In *Proc. IEEE*, volume 72, pages 1746–1754, 1984.

[5] V. Balakrishnan and L. Vandenberghe. Algorithms and software for LMI problems in control. *IEEE Control Syst. Mag.*, 17(5):89–95, 1997.

[6] P. Benner and R. Byers. Evaluating products of matrix pencils and collapsing matrix products. *Numer. Linear Algebra Appl.*, 8:357–380, 2001.

[7] P. Benner, R. Byers, R. Mayo, E. S. Quintana-Orti, and V. Hernandez. Parallel algorithms for LQ optimal control of discrete-time periodic linear systems. *Journal of Parallel and Distributed Computing*, 62:306–325, 2002.

[8] S. Bittanti, P. Colaneri, and G. De Nicolao. A note on the maximal solution of the periodic Riccati equation. *IEEE Trans. Autom. Contr.*, 34(12):1316–1319, 1989.

[9] S. Bittanti, P. Colaneri, and G. De Nicolao. The periodic Riccati equation. In S. Bittanti, A. J. Laub, and J. C. Willems, editors, *The Riccati Equation*, chapter 6, pages 127–162. Springer-Verlag, Berlin, 1991.

[10] A. Bojanczyk, G. H. Golub, and P. Van Dooren. The periodic Schur decomposition; algorithm and applications. In F. T. Luk, editor, *Proc. SPIE Conference*, volume 1770, pages 31–42, 1992.

[11] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM studies in applied mathematics. SIAM, Philadelphia, 1994. ISBN 0-89871-334-X.

[12] S. Boyd and L. Vandenberghe. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[13] M.P. Calvo and J.M. Sanz-Serna. *Numerical Hamiltonian problems*. Chapman & Hall, London, 1994.

[14] Y.-Zh. Chen, S.-B. Chen, and J.-Q. Liu. Comparison and uniqueness theorems for periodic Riccati differential equations. *Int. J. Control*, 69(3):467–473, 1998.

[15] L. Dieci. Numerical integration of the differential Riccati equation and some related issues. *SIAM J. Numer. Anal.*, 29(3):781–815, 1992.

[16] L. Dieci and T. Eirola. Positive definiteness in the numerical solution of Riccati differential equations. *Numer. Math.*, 67:303–313, 1994.

[17] J.M. Franco and I. Gmez. Fourth-order symmetric DIRK methods for periodic stiff problems. *Num. Algorithms*, 32:317–336, 2003.

[18] L. Freidovich, S. Gusev, and A. Shiriaev. LMI approach for solving periodic matrix Riccati equation. In *Proc. of the 3rd IFAC Workshop on periodic control systems, PSYCO'07*, St Petersburg, Russia, 2007.

[19] L. Freidovich, S. Gusev, and A. Shiriaev. Computing a transverse linearization for mechanical systems with two and more underactuated degrees of freedom. Submitted, 2008.

[20] L. Freidovich, R. Johansson, A. Robertsson, A. Sandberg, and A. Shiriaev. Virtual-holonomic-constraints-based design of stable oscillations of Furuta pendulum: Theory and experiments. *IEEE Trans. Robotics*, 23(4):827–832, 2007.

[21] L. Freidovich, P. La Hera, U. Mettin, and A. Shiriaev. New approach for swinging up the Furuta pendulum: Theory and experiments. Submitted to *IFAC Journal of Mechatronics*, 2009.

[22] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[23] R. Granat and B. Kågström. Direct eigenvalue reordering in a product of matrices in periodic Schur form. *SIAM J. Matrix Anal. Appl.*, 28(1):285–300, 2006.

[24] R. Granat, B. Kågström, and D. Kressner. Computing periodic deflating subspaces associated with a specified set of eigenvalues. *BIT*, 47:763–791, 2007.

[25] R. Granat, B. Kågström, and D. Kressner. Matlab tools for solving periodic eigenvalue problems. In *Proc. of the 3rd IFAC Workshop, PSYCO'07*, St Petersburg, Russia, 2007.

[26] E. Hairer. Software for ODE solvers. Department of Mathematics, University of Genéve, Switzerland, June 2008. http://www.unige.ch/~hairer/software.html.

[27] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-preserving algorithms for ordinary differential equations*. Springer-Verlag, Berlin, 2nd edition, 2006. ISBN 3-540-30663-3.

[28] E. Hairer, R.I. McLachlan, and A. Razakarivony. Achieving Brouwer's law with implicit Runge-Kutta methods. *BIT*, 48(2), 2008.

[29] J. J. Hench, C. S. Kenney, and A. J. Laub. Methods for the numerical integration of Hamiltonian systems. *Circuits Systems Signal Process*, 13(6):695–732, 1994.

[30] J. J. Hench and A. J. Laub. Numerical solution of the discrete-time periodic Riccati equation. *IEEE Trans. Autom. Contr.*, 39(6):1197–1209, 1994.

[31] G. Hu. Symplectic Runge-Kutta methods for the Kalman-Bucy filter. *IMA J. Math. Control Info.*, 2007.

[32] S. Johansson, B. Kågström, A. Shiriaev, and A. Varga. Comparing one-shot and multi-shot methods for solving periodic Riccati differential equations. In *Proc. of the 3rd IFAC Workshop on periodic control systems, PSYCO'07*, St Petersburg, Russia, 2007.

[33] E. Klerk. *Aspects of semidefinite programming: Interior point algorithms and selected applications*. Applied optimization. Kluwer Academic Pub., Dordrecht, 2002. ISBN 1-402-00547-4.

[34] D. Kressner, V. Mehrmann, and T. Penzl. CTDSX – a collection of benchmark examples for state-space realizations of continuous-time dynamical systems. SLICOT working note 1998-9, WGS, 1998.

[35] A.J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Autom. Contr.*, AC-24:913–921, 1979.

[36] B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, Cambridge, 2004. ISBN 0-521-77290-7.

[37] J. Löfberg. Yalmip homepage. Automatic Control Laboratory, ETH Zurich, Switzerland, January 2009. http://control.ee.ethz.ch/~joloef/yalmip.php.

[38] C. Ludwig. ODE MEXfiles Homepage. Zentrum Mathematik Technische Universität München, Germany, January 2009.
http://www-m3.ma.tum.de/twiki/bin/view/Software/ODEHome.

[39] K. Lust. psSchur Homepage. Department of Mathematics, K.U.Leuven, Belgium, January 2009. http://perswww.kuleuven.be/~u0006235/ACADEMIC/r_psSchur.html.

[40] R. McLachlan. A new implementation of symplectic Runge-Kutta methods. *SIAM J. Sci. Comput.*, 29(4):1637–1649, 2007.

[41] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*, volume 163 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, 1991.

[42] J. Perram, A. Robertsson, A. Sandberg, and A. Shiriaev. Periodic motion planning for virtually constrained mechanical system. *Systems & Control Lett.*, 55(11):900–907, 2006.

[43] SeDuMi homepage. Advanced Optimization Laboratory, McMaster University, Canada, January 2009. http://sedumi.ie.lehigh.edu/.

[44] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, NY, 1996.

[45] SLICOT homepage. Germany, June 2008. http://www.slicot.org.

[46] J. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimizaition over symmetric cones (updated for version 1.05). Technical report, Department of Econometrics, Tilburg University, Tilburg, The Netherlands, 2001.

[47] S. Tan and W. Zhong. Numerical solutions of linear quadratic control for time-varying systems via symplectic conservative perturbation. *Appl. Math. Mech.*, 28(3):277–287, 2007.

[48] A. Varga. On solving periodic differential matrix equations with applications to periodic system norms computation. In *Proc. of CDC'05*, Seville, Spain, 2005.

[49] A. Varga. A periodic systems toolbox for MATLAB. In *Proc. of 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

[50] A. Varga. On solving periodic Riccati equations. *Numer. Linear Algebra Appl.*, 15(9):809–835, 2008.

[51] V.A. Yakubovich. Linear-quadratic optimization problem and frequency theorem for periodic systems. *Siberian Mathematical Journal*, 27(4):181–200, 1986.

[52] F. Zhang. *The Schur Complement and Its Applications*. Numerical Methods and Algorithms. Springer-Verlag, New York, 2005. ISBN 978-0-387-24271-2.