BIT Numer Math (2010) 50: 301–329
DOI 10.1007/s10543-010-0257-5

BIT

A numerical evaluation of solvers for the periodic Riccati differential equation

Sergei Gusev · Stefan Johansson · Bo Kågström · Anton Shiriaev · Andras Varga

Received: 18 June 2009 / Accepted: 4 February 2010 / Published online: 23 February 2010

Abstract Efficient and accurate structure exploiting numerical methods for solving the periodic Riccati differential equation (PRDE) are addressed. Such methods are essential, for example, to design periodic feedback controllers for periodic control systems. Three recently proposed methods for solving the PRDE are presented and evaluated on challenging periodic linear artificial systems with known solutions and

Communicated by Axel Ruhe.

Financial support has been provided in part by the *Swedish Foundation for Strategic Research* (the frame program grant A3 02:128), by the *Swedish Research Council* (the grant 2008-5243), EU Mål2 Structural Funds (UMIT-project), and by the *Russian Federal Agency for Science and Innovation* (the project 02.740.11.5056).

S. Gusev
Department of Mathematics and Mechanics, St. Petersburg State University, St. Petersburg, Russia
e-mail: gusev@ieee.org

S. Johansson (✉) · B. Kågström
Department of Computing Science and HPC2N, Umeå University, Umeå, Sweden
e-mail: stefanj@cs.umu.se

B. Kågström
e-mail: bokg@cs.umu.se

A. Shiriaev
Department of Applied Physics and Electronics, Umeå University, Umeå, Sweden

A. Shiriaev
Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway
e-mail: Anton.Shiriaev@itk.ntnu.no

A. Varga
Institute of Robotics and Mechatronics, German Aerospace Center, DLR, Oberpfaffenhofen, Germany
e-mail: Andras.Varga@dlr.de

applied to the stabilization of periodic motions of mechanical systems. The first two methods are of the type multiple shooting and rely on computing the stable invariant subspace of an associated Hamiltonian system. The stable subspace is determined using either algorithms for computing an ordered periodic real Schur form of a cyclic matrix sequence, or a recently proposed method which implicitly constructs a stable deflating subspace from an associated lifted pencil. The third method reformulates the PRDE as a convex optimization problem where the stabilizing solution is approximated by its truncated Fourier series. As known, this reformulation leads to a semidefinite programming problem with linear matrix inequality constraints admitting an effective numerical realization. The numerical evaluation of the PRDE methods, with focus on the number of states (n) and the length of the period (T) of the periodic systems considered, includes both quantitative and qualitative results.

Keywords Periodic systems · Periodic Riccati differential equations · Orbital stabilization · Periodic real Schur form · Periodic eigenvalue reordering · Hamiltonian systems · Linear matrix inequalities · Numerical methods

Mathematics Subject Classification (2000) 15A21 · 15A39 · 34K13 · 49N05 · 65F15 · 65P10 · 70M20 · 70Q05 · 90C22

1 Introduction

In this paper, we evaluate numerical methods for solving for $X(t) \in \mathbb{R}^{n\times n}$ in the *periodic Riccati differential equation* (PRDE) [1, 10, 41]:

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

where the state matrix $A(t) \in \mathbb{R}^{n\times n}$ and the input matrix $B(t) \in \mathbb{R}^{n\times m}$ are locally integrable on $[0,\ T]$ and T-periodic, i.e., $A(t) = A(t+T)$ and $B(t) = B(t+T)$ for all $t \geq 0$. The continuous T-periodic weighting matrices $Q(t) \in \mathbb{R}^{n\times n}$, and $R(t) \in \mathbb{R}^{m\times m}$ are symmetric, and positive semidefinite and positive definite, respectively. The PRDE arises, for example, in solving the *periodic linear quadratic regulator* (periodic LQR) problem, where the *quadratic performance index*

$$\int_0^\infty \Big[x(t)^T Q(t)x(t) + u(t)^T R(t)u(t)\Big]\,dt, \tag{1.2}$$

is to be minimized, with respect to $u(t)$, on solutions of the *periodic linear control system*:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(0) = x_0. \tag{1.3}$$

In the last few years, the interest for developing robust solvers for the PRDE has increased, mainly because the existing, so-called, *periodic generator methods* [28, 33, 50] for solving the PRDE are unreliable for systems with a large number of states n or with a large period T. However, recently new methods have been proposed. In this paper, we examine three of these methods: two variants of the periodic multi-shot method [47, 49] (invariant subspace approaches) and the SDP method [18] (a convex optimization approach). PRDEs with an indefinite quadratic term is not considered in this paper (see [2]).

The periodic multi-shot methods are based on discretization techniques, which turn a continuous-time problem into an equivalent discrete-time problem. The methods are a further development of the periodic generator method (also known as the one-shot method). To solve the PRDE a linear periodic Hamiltonian system must be integrated over one period. The importance of using symplectic integration methods for solving the Hamiltonian system of ODEs has recently been emphasized in, e.g., [29–31, 46, 47]. 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 [11, 27] and the reordering of eigenvalues in the periodic real Schur form [23]. The second approach implicitly constructs a stable deflating subspace from an associated lifted pencil [49].

In addition to the multi-shot methods we explore a convex optimization approach, which is based on the approximation of the stabilizing solution of the PRDE by its truncated Fourier series. By doing this approximation and reformulating the search of a stabilizing solution of the PRDE as a maximization problem, the task is turned into a *semidefinite programming* (SDP) problem with *linear matrix inequality* (LMI) constraints [18].

The purpose of this contribution is two-fold: (i) to review the three PRDE methods briefly mentioned above and discuss algorithms for their effective implementation in terms of fast and robust MATLAB software; (ii) present a numerical evaluation performed on challenging PRDE problems which includes quantitative as well qualitative results and comparisons. The methods are first compared on a set of artificially constructed periodic linear control systems with known solutions, and then on two problems for which solutions are unknown and of interest for the task of orbital stabilization of forced oscillations in controlled mechanical systems.

The paper is organized as follows. In Sect. 2, we present some preliminaries including system characteristics of periodic systems and introduce the LQR problem. Sections 3 and 4 deal with invariant subspace approaches and the convex optimization approach, respectively, for solving the PRDE (1.1). In Sect. 5, the multi-shot and the SDP methods are tested and evaluated. We end with an evaluation summary and some conclusions in Sect. 6.

2 Preliminaries

We start by introducing some concepts and important results known for the periodic linear system (1.3) in Sect. 2.1. For a detailed discussion of periodic systems see, e.g., [1, 7, 10]. In Sect. 2.2, we briefly describe the underlying control problem and state the conditions for existence of a stabilizing solution.

2.1 Some system characteristics of periodic linear systems

First, consider a square time-varying matrix $M(t)$[1] and let $\Phi_M(t, t_0)$ be the *transition matrix* associated with $M(t)$ satisfying

$$\frac{\partial}{\partial t}\Phi_M(t, t_0) = M(t)\Phi_M(t, t_0), \quad \Phi_M(t_0, t_0) = I,$$

where I denotes the identity matrix of the same size as M. For a T-periodic $M(t)$, the transition matrix evaluated over one period is known as the *monodromy matrix* $\Psi_M(t_0) = \Phi_M(t_0 + T, t_0)$. The eigenvalues of $\Psi_M(t_0)$ are called the *characteristic multipliers* of $M(t)$. These eigenvalues are independent of t_0, thus $\Psi_M(t_0)$ has the same spectrum for all t_0. $M(t)$ is said to be a *stable* periodic matrix if all characteristic multipliers λ_i are inside the unit circle (open unit disc) centered in the origin, i.e., $|\lambda_i| < 1$, for all i.

Next, we recall the definitions of reachability/stabilizability and observability/detectability for a periodic linear control system (1.3) (see for example [1, 10]). 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 (1.3) 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.

Now, consider a periodic linear control system with an output: $y(t) = C(t)x(t)$, where $C(t) \in \mathbb{R}^{p\times n}$ is a T-periodic matrix. 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 (1.3) 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 Problem description

The optimal control problem we consider is to compute a stabilizing controller for the periodic linear control system (1.3), where the optimal periodic controller is obtained by solving the LQR problem [3, 39, 43, 50]. 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 a quadratic performance index is minimized.

It is well known that the optimal control input $u^*(t)$ for the periodic linear control system (1.3) with the associated quadratic performance index (1.2) has the form

$$u^*(t) = -K(t)x(t), \quad \text{where } K(t) = R(t)^{-1}B(t)^T X(t), \tag{2.1}$$

[1] We remark that M is used as an arbitrary matrix with different meanings (time-varying, positive (semi)definite, M_i in a cyclic matrix sequence). The actual use of M is stated within its context.

and the periodic matrix $X(t) \in \mathbb{R}^{n\times n}$ is the so-called *stabilizing solution* for (1.1). This means that $X(t)$ is symmetric and positive semidefinite, and the closed-loop matrix $A(t) - B(t)K(t)$ is stable. The next statement from [9] describes the sufficient conditions for the existence of a T-periodic stabilizing solution for (1.1).

Theorem 2.1 *The PRDE* (1.1) *admits a unique positive semidefinite periodic stabilizing solution* $X(t) = X(t)^T \geq 0$ *if the pair* $(A(t), B(t))$ *is stabilizable and the pair* $(A(t), Q(t)^{1/2})$ *is detectable*, *where* $(Q(t)^{1/2})^T Q(t)^{1/2} = Q(t)$.

3 Invariant subspace approaches

In this section, we are considering the periodic multi-shot methods proposed in [47, 49]. They are based on the periodic generator method [28, 33, 50], which obtains a periodic solution of (1.1) by computing the stable invariant subspace of the transition matrix of the Hamiltonian system associated with (1.2) and (1.3). The multi-shot methods use techniques explicitly designed for computing the invariant subspace of periodic systems: the ordered periodic real Schur form (Sect. 3.3.1) or the ordered generalized real Schur form of a lifted pencil (Sect. 3.3.2). The associated Hamiltonian differential system is solved using a symplectic (structure preserving) integration method, see Sect. 3.1. The two multi-shot methods are presented in Sects. 3.4 and 3.5. We end with a brief overview of the MATLAB implementations of the methods in Sect. 3.6.

3.1 Computing the transition matrix

When solving the PRDE (1.1) associated with the periodic linear control system (1.3) using an invariant subspace approach, a linear Hamiltonian system with symplectic flow must be solved. For details on Hamiltonian systems, symplectic matrices, and symplectic integration methods see, e.g., [12, 25, 36].

The periodic *Hamiltonian matrix* $H(t) \in \mathbb{R}^{2n\times 2n}$ associated with (1.1) is

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

where the time-varying $H(t)$ satisfies $H(t)^T J + JH(t) = 0$ for all t, and $J = -J^T = -J^{-1}$ is the skew-symmetric matrix

$$J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix},$$

with I_n the $n \times n$ identity matrix. From the initial value problem

$$\frac{\partial}{\partial t}\Phi_H(t, t_0) = H(t)\Phi_H(t, t_0), \qquad \Phi_H(t_0, t_0) = I_{2n}, \tag{3.1}$$

the *transition matrix* $\Phi_H(t, t_0)$ associated with $H(t)$ is computed. The system (3.1) is a linear Hamiltonian system where the transition matrix $\Phi_H(t, t_0)$ for all $t \geq t_0$ is *symplectic*. Note, $\Phi_H(t,t_0)$ is symplectic because it satisfies $\Phi_H(t,t_0)^T J \Phi_H(t,t_0) = J$ and therefore its $2n$ eigenvalues (characteristic multipliers) appear in reciprocal pairs $(\lambda_i, 1/\lambda_i)$, for $i = 1, \ldots, n$. We recall from Sect. 2.1, that the transition matrix evaluated over one period is known as the monodromy matrix $\Psi_H(t_0) = \Phi_H(t_0 + T, t_0)$.

The (time) flow map of a Hamiltonian system (3.1) is symplectic, e.g., see [36]. To preserve the symplectic characteristic of the Hamiltonian system (3.1) an integrator that preserves the symplectic flow of the problem must be used. For the computational experiments in Sect. 5, we use the symplectic and symmetric Gauss Runge-Kutta method [25, 26]. It is a one-step implicit Runge-Kutta method with fixed time steps where the intervening nonlinear system is solved using fixed-point iteration.

3.2 Periodic generator versus multi-shot methods

Traditionally, the PRDE (1.1) is solved using a periodic generator method (e.g., see [28, 33, 50]), which is a *single shooting type* method. This approach involves solving the linear Hamiltonian system (3.1) over one period T, where the knowledge of the stable invariant subspace of $\Phi_H(t,t_0)$ allows to compute the stabilizing solution of the PRDE (1.1). The periodic generator method has some major disadvantages and is potentially numerically unreliable. Depending on which approach is used (see [49]), one or two ODEs with unstable dynamics must be solved. For systems with large periods this will result in a significant accumulation of truncation errors. 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, 28].

The alternative periodic multi-shot methods [47, 49] reduce the impact of the numerical difficulties caused by the periodic generator 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 $\Psi_H(t_0) \in \mathbb{R}^{2n\times 2n}$ with $t_0 = 0$:

$$\Psi_H(0) \equiv \Phi_H(T,0) = \Phi_H(T, T-\Delta)\cdots\Phi_H(2\Delta,\Delta)\Phi_H(\Delta,0), \tag{3.2}$$

where $\Delta = T/N$ for a suitable number of time instances[2] $N \in \mathbb{N}$. In the following, denote $\Phi_k = \Phi_H(k\Delta, (k-1)\Delta)$, $k = 1, \ldots, N$. Notably, $\Phi_1, \ldots, \Phi_N$ is an N-cyclic matrix sequence of $2n \times 2n$ real matrices. The linear Hamiltonian system (3.1) can now be integrated for each transition matrix Φ_k, and methods for periodic eigenvalue problems can be used to compute the stable invariant subspace. Notably, since each Φ_k is symplectic the eigenvalues of $\Psi_H(0)$ appears in reciprocal pairs and consequently it has n eigenvalues inside and n eigenvalues outside the unit circle.

In consequence, the multi-shot methods have several advantages compared to the periodic generator method:

[2] In practice, the constant N can be chosen such that the time grid points $T(k-1)/N$ at which the discrete-time values of the solution $X(t)$ are computed, coincide with the sampling times used to implement the resulting continuous-time stabilizing periodic controller.

(i) The linear Hamiltonian system, which always has unstable dynamics, is solved over short time intervals of the period T. This avoids excessive accumulation of errors and 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 periodic generator method where (possibly) two ODEs are solved in sequence.
(iii) The system's periodicity is exploited, by explicitly using methods designed for periodic systems.
(iv) The numerical integration of the Hamiltonian system can easily be parallelized. This is of great value since this part can be very computationally intensive.
(v) Since the integration of the Hamiltonian system is done over short time intervals, the importance of using a symplectic solver is not critical.

3.3 Computing stable invariant subspaces of products of cyclic matrix sequences

When solving the PRDE with a multi-shot method we are interested in the stable periodic invariant subspace of a matrix product $M = M_P \cdots M_2 M_1$ associated with a P-cyclic matrix sequence $M_1, \ldots, M_P$, where each M_k is a square matrix and $M_{k+P} = M_k$ for any positive integer k. 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 [11]. Here, we review two methods for computing a stable subspace that are used in the multi-shot methods discussed in Sects. 3.4 and 3.5. The first is based on an ordered periodic Schur decomposition and the second is based on computing a stable deflating subspace of a lifted pencil.

3.3.1 Ordered periodic real Schur form

Given a cyclic sequence of $n_c \times n_c$ real matrices M_k, $k = 1, \ldots, P$, the *periodic real Schur form* (PRSF) is defined as follows [11, 27]: there exists a P-cyclic orthogonal matrix sequence $Z_k \in \mathbb{R}^{n_c \times n_c}$ such that

$$Z_{k+1}^T M_k Z_k = S_k, \quad k = 1, \ldots, P, \tag{3.3}$$

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 $M_P \cdots M_2 M_1$.

When computing the PRSF it is not possible to simultaneously specify the order of the eigenvalues of the matrix product $S_P \cdots S_1$. In the periodic multi-shot methods, we need to be able to separate the stable and unstable eigenvalues of the P-cyclic matrix sequence (3.2) of $n_c \times n_c$ matrices with $P = N$, $n_c = 2n$, and $M_k = \Phi_k$ in (3.3). For this purpose we can use a direct method for reordering the eigenvalues of a periodic matrix sequence in PRSF that results in an *ordered PRSF* [23]: there exists a periodic orthogonal matrix sequence $Q_k \in \mathbb{R}^{2n\times 2n}$, such that

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

where $T_{11}^{(k)} \in \mathbb{R}^{n\times n}$ and $T_{22}^{(k)} \in \mathbb{R}^{n\times n}$. 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.[3] Then the first n columns of the sequence Q_k span the *stable* right periodic invariant subspace, and the last n columns span the *unstable* left periodic invariant subspace.

3.3.2 Stable deflating subspace of a lifted pencil

An alternative approach to compute the stable invariant subspace is the *fast algorithm* proposed in [49]. This method belongs to the family of "fast" methods [6, 14] for *discrete-time algebraic Riccati equations*, and is an extension of the *swapping and collapsing* approach [5, 6] of quotient-products. Provided a P-cyclic matrix sequence $M_1, \ldots, M_P$, where $M_k \in \mathbb{R}^{n_c\times n_c}$ for $k = 1, \ldots, P$, the fast algorithm constructs a stable invariant subspace of the matrix product $M_P \cdots M_2 M_1$ from a stable deflating subspace of an associated lifted pencil.

Define the *associated lifted pencil* to the cyclic matrix pairs (M_k, I_{n_c}):

$$S - zT = \begin{bmatrix} M_1 & -I_{n_c} & 0 & \cdots & 0 \\ 0 & M_2 & -I_{n_c} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & & & M_{P-1} & -I_{n_c} \\ -zI_{n_c} & 0 & \cdots & 0 & M_P \end{bmatrix}. \tag{3.4}$$

The finite eigenvalues of this pencil are the eigenvalues of the matrix product $M_P \cdots M_2 M_1$. By using an orthogonal transformation matrix $U^{(1)}$ the rows of the matrix $\left[\begin{smallmatrix} -I_{n_c} \\ M_2 \end{smallmatrix}\right]$ are compressed to $\left[\begin{smallmatrix} R_1 \\ 0 \end{smallmatrix}\right]$, where R_1 is an $n_c \times n_c$ nonsingular matrix. The first two block-rows of $S - zT$ now become

$$\begin{bmatrix} \widetilde{M}_1 & R_1 & -U_{12}^{(1)} & 0 & \cdots & 0 \\ \widehat{M}_2 & 0 & -U_{22}^{(1)} & 0 & \cdots & 0 \end{bmatrix},$$

where the $n_c \times n_c$ matrix $U_{ij}^{(1)}$ is the (i, j)-th block of the matrix $U^{(1)}$. Applying the row compression recursively to (3.4) on $\left[\begin{smallmatrix} -U_{22}^{(k-1)} \\ M_{k+1} \end{smallmatrix}\right]$, for $k = 2, \ldots, P-1$, transforms

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

the matrix pencil $S - zT$ to the reduced pencil

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

where the regular matrix pencil $\widetilde{M}_P - zU_{22}^{(P-1)}$ contains all finite eigenvalues of $S - zT$.

The stable deflating subspace is computed from an ordered generalized real Schur decomposition [32]

$$Q^T(\widetilde{M}_P - zU_{22}^{(P-1)})Z = \begin{bmatrix} \widehat{S}_{11} - z\widehat{T}_{11} & \widehat{S}_{12} - z\widehat{T}_{12} \\ 0 & \widehat{S}_{22} - z\widehat{T}_{22} \end{bmatrix}, \tag{3.5}$$

where Q and Z are orthogonal matrices, and the upper quasi-triangular matrix pencil $\widehat{S}_{11} - z\widehat{T}_{11} \in \mathbb{R}^{q \times q}$ has only finite eigenvalues inside the unit circle. The first q columns of Z span the *stable* (right) deflating subspace, which also span the stable invariant subspace of the product $M_P \cdots M_1$. Consequently, the last $n_c - q$ columns of Q span the *unstable* (left) deflating subspace.

3.4 Multi-shot method based on an ordered periodic Schur form

The multi-shot method using the ordered PRSF is presented in Algorithm 1. To enhance the numerical accuracy, the Hamiltonian system in Step 1 is preferably solved with a symplectic solver like the symplectic Gauss Runge-Kutta [25, 26].

Algorithm 1

1. Compute the transition matrices $\Phi_1, \ldots, \Phi_N$ by solving the linear Hamiltonian system (3.1) for each interval $[(k-1)\Delta, k\Delta]$, where $k = 1, \ldots, N$.
2. Compute the ordered PRSF associated with the matrix product $\Psi_H(0) = \Phi_N \cdots \Phi_2\Phi_1$: for $k = 1, \ldots, N$,

$$Q_{k+1}^T Z_{k+1}^T \Phi_k Z_k Q_k = T_k \equiv \begin{bmatrix} T_{11}^{(k)} & T_{12}^{(k)} \\ 0 & T_{22}^{(k)} \end{bmatrix}, \tag{3.6}$$

with $Q_{k+N} = Q_k$, $Z_{k+N} = Z_k$, where the matrix products $T_{11}^{(N)} \cdots T_{11}^{(1)}$ and $T_{22}^{(N)} \cdots T_{22}^{(1)}$ have n eigenvalues inside and n eigenvalues outside the unit circle, respectively.

3. For each k, partition the product of the orthogonal transformation matrices from (3.6) into four $n \times n$ blocks (with $Y_{11}^{(k)}$ nonsingular [33, 50]) as

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

4. Compute $X_k =: X(t)$ at $t = (k-1)\Delta$, $k = 1, \dots, N$, as

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

To obtain 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 Multi-shot method based on a deflating subspace of a lifted pencil

The multi-shot method using a stable deflating subspace of a lifted pencil is presented in Algorithm 2. It is based on Algorithm 1 (computing the transition matrices in Step 1 is the same), but utilizes the fast algorithm in Sect. 3.3.2 for computing a stable subspace. The fast multi-shot approach takes advantage of that the solution $X(t)$ at two successive time steps $(k-1)\Delta$ and $k\Delta$ are related as [10]

$$X_k = \left(X_{k+1} \Phi_{12}^{(k)} - \Phi_{22}^{(k)} \right)^{-1} \left(\Phi_{21}^{(k)} - X_{k+1} \Phi_{11}^{(k)} \right) \quad \text{with } \Phi_k = \begin{bmatrix} \Phi_{11}^{(k)} & \Phi_{12}^{(k)} \\ \Phi_{21}^{(k)} & \Phi_{22}^{(k)} \end{bmatrix}, \tag{3.7}$$

where $X_k =: X(t)$ at $t = (k-1)\Delta$, $k = 1, \dots, N$, and Φ_k is partitioned in four $n \times n$ blocks.

Algorithm 2

1. Compute the transition matrices $\Phi_1, \dots, \Phi_N$ by solving the linear Hamiltonian system (3.1) for each interval $[(k-1)\Delta, k\Delta]$, where $k = 1, \dots, N$.
2. Use the fast algorithm as outlined in Sect. 3.3.2, to compute the regular pencil $\widetilde{\Phi}_N - zU_{22}^{(N-1)}$ containing all finite eigenvalues of $S - zT$ in (3.4) associated with the cyclic $2n \times 2n$ matrix Φ_k, $k = 1, \dots, N$. Notably, the lifted matrix pencil $S - zT$ is not explicitly constructed.
3. Partition the transformation matrix Z in (3.5) into four $n \times n$ blocks as

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

   where the first n columns of Z span the stable right deflating subspace of $\widetilde{\Phi}_N - zU_{22}^{(N-1)}$. Compute the initial solution to (3.7) as $X_1 = Z_{21}(Z_{11})^{-1}$.
4. Compute X_k, $k = N, N-1, \dots, 1$, iteratively using (3.7), with $X_{N+1} = X_1$.

An iterative refinement of accuracy can be performed by repeating the convergent process at Step 4 above 2–3 times, where a new initial solution X_1 is computed from

X_2 using (3.7). We remark that the implementation of Algorithm 2 can be realized using standard robust numerical routines [49].

3.6 MATLAB implementations

To compute the stable invariant subspace, the multi-shot method based on the ordered PRSF uses Fortran subroutines for computing the PRSF [38] and periodic eigenvalue reordering [23] (to be available in the upcoming *PEP toolbox* [24]). The (fast) multi-shot method based on a deflating subspace of a lifted pencil is available in the *Periodic System Toolbox* for MATLAB [48]. To solve the linear Hamiltonian system, builtin ODE solvers in MATLAB and a new MATLAB implementation of the symplectic Gauss Runge-Kutta method are used.

4 A convex optimization approach

A second approach to find a stabilizing solution for 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 SDP problem with LMI constraints. This method has been proposed in [18] and an improved version is presented in Sect. 4.2. But first, necessary theory for the stabilizing solution is introduced in Sect. 4.1, and we end with a brief overview of the MATLAB implementation in Sect. 4.3.

4.1 Extremal property of stabilizing solution

Define the Riccati operator $\mathcal{R}$ of (1.1) as

$$\mathcal{R}(X(t),t)=\dot{X}(t)+A(t)^TX(t)+X(t)A(t)-X(t)B(t)R(t)^{-1}B(t)^TX(t)+Q(t). \tag{4.1}$$

The following theorem regarding the existence of a stabilizing solution, and proved in [1, 13], is a slight generalization of a theorem in [8]. Note that there is no positive definite assumption on $Q(t)$.

Theorem 4.1 [1, 13] *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*

$$\mathcal{R}(X(t),t)\geq 0,\quad \textit{for all } t\geq 0. \tag{4.2}$$

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

$$X_+(t)\geq X(t),\quad \textit{for all } t\geq 0. \tag{4.3}$$

In particular, $X_+(t)$ satisfying (4.3) *is called the* maximal *T-periodic stabilizing solution.*

Notably, the theorem also holds for the cases $R(t) \neq I_m$ or when $R(t)$ is time-varying since $R(t)$ can be transformed into $R(t) = I_m$ by change of input coordinates.

For a set of matrices $W_j = W_j^T > 0$ and L distinct time instances $t_j \geq 0$, where $j = 1, \ldots, L$, define the functional

$$J(X(t)) = \sum_{j=1}^{L} \operatorname{tr}\left(W_j X(t_j)\right). \tag{4.4}$$

Here tr(M) denotes the *trace* of a matrix M. The inequality (4.3) implies that $\operatorname{tr}(W_j X_+(t_j)) \geq \operatorname{tr}(W_j X(t_j))$ for all t_j. Consequently, it follows from Theorem 4.1 that a maximum of (4.4) over the set of bounded matrices $X(t)$, satisfying (4.2), is achieved at the stabilizing solution $X_+(t)$ [18], i.e., $J(X_+(t)) \geq J(X(t))$ for all $X(t)$.

4.2 Reformulation of the PRDE

As stated in the end of Sect. 4.1, the maximal stabilizing solution $X_+(t)$ of the PRDE (1.1) also maximizes the cost function J in (4.4). To compute $X_+(t)$ an infinite dimensional SDP problem can be solved, for which the linear functional J is maximized over a convex set of matrices $X(t)$, satisfying the inequality (4.2). To solve this infinite dimensional optimization problem we first approximate it with a sequence of finite dimensional problems.

From the Schur complement,[4] it follows that the Riccati inequality (4.2) associated with the T-periodic PRDE (1.1) can be reformulated as the LMI

$$\mathcal{S}(X(t), t) \geq 0, \tag{4.5}$$

where

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

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 a truncated series of the solution written in a rich class of basis functions. These basis functions can be chosen such that the periodic characteristics of the underlying system are emphasized. A widely used approximation of periodic matrices relies on truncated Fourier expansions, which are used in the following. Let $\omega = 2\pi/T$ and $q \geq 1$, $q \in \mathbb{N}$, then

$$\widetilde{X}(t) = \sum_{k=-q}^{q} e^{\imath k\omega t} X_k, \quad \text{and} \quad \frac{d\widetilde{X}(t)}{dt} = \sum_{k=-q}^{q} \imath k\omega e^{\imath k\omega t} X_k, \tag{4.6}$$

where X_{-k} is the complex conjugate of 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 $0 \leq t_j \leq T$ are some time instances and $j = 1, \ldots, L$ for a suitable integer L.

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

We can now formulate the finite dimensional SDP problem as

$$\begin{aligned} &\min \quad -J(\widetilde{X}(t)),\\ &\text{s.t.} \quad \mathcal{S}_j \geq 0, \quad j = 1, \dots, L, \end{aligned} \tag{4.7}$$

where the objective function is obtained from (4.4) using $\widetilde{X}(t)$ in (4.6):

$$J(\widetilde{X}(t)) = \sum_{j=1}^{L} \operatorname{tr}\left(W_j \sum_{k=-q}^{q} e^{\imath k \omega t_j} X_k \right). \tag{4.8}$$

Note that minimizing $-J(\widetilde{X}(t))$ is equivalent to maximizing $J(\widetilde{X}(t))$. By solving (4.7) an approximate stabilizing solution $\widetilde{X}_+(t)$ of (1.1) can be computed, where

$$\lim_{q\to\infty} \widetilde{X}_+(t) = \lim_{q\to\infty} \sum_{k=-q}^{q} e^{\imath k\omega t} X_k = X_+(t), \quad \text{for all } t \in [t_0, T].$$

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 $O(m_{\text{SDP}}^{2.5} n_{\text{SDP}} (m_{\text{SDP}} + n_{\text{SDP}}) \log(1/\varepsilon))$, where ε is the desired accuracy [45]. In practice, the complexity is much lower. For the SDP problem (4.7) we have $(2q+1)n(n+1)/2$ variables and the $(n+m)L \times (n+m)L$ block diagonal matrix $\operatorname{diag}(\mathcal{S}_1, \dots, \mathcal{S}_L)$ forms the LMI constraints. The original outline of the method [18] had a smaller number of LMI constraints (m_{SDP}) but depended on a larger number of variables (n_{SDP}).

4.3 The MATLAB implementation

The MATLAB implementation uses *SeDuMi* [42, 45] (a MATLAB toolbox for optimization over symmetric cones) to solve the SDP problem and *YALMIP* [37] for modeling the optimization problem. In the MATLAB implementation, the weight matrices $W_1, \dots, W_L$ in (4.8) are all set to the identity matrix. However, these matrices could be used as tuning parameters in the SDP problem.

5 Numerical experiments

Focusing on two important problem parameters (number of states n and the length of the period T), we evaluate the three methods to solve the PRDE described in the previous sections: the multi-shot method based on ordered PRSF, the multi-shot method based on the fast algorithm, and the convex optimization approach. These are further compared against the periodic generator method realized as in [30, 31, 47]. Below, the corresponding solvers for the four methods are called the *multi-shot solver*, the *fast multi-shot solver*, the *SDP solver*, and the *one-shot solver*, respectively.

For the two multi-shot methods we solve the linear Hamiltonian system (3.1) 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 Gauss Runge-Kutta method with fixed time steps. For ode45 and ode113 we use the relative tolerance 10^{-9} and the absolute tolerance 10^{-16}. For sgrk we use 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.

Based on previous studies [31], only sgrk is used together with the one-shot solver. For the SDP solver we use 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 (3.2) in the multi-shot methods is set to $N = 100$. We have based our choice of N on the results in [31, 49]. For consistency, the number of time instances L of the LMI constraints in (4.7) is set to $L = N$. Moreover, the stabilizing solution and its derivative are approximated with the truncated Fourier expansion as in (4.6) with $q = 10$ (one could argue to set q higher or lower, but $q = 10$ has shown to be a good compromise).

The implementations of the three PRDE solvers have been done in MATLAB, utilizing built-in functions and gateways to existing Fortran subroutines. Version 1.21 of SeDuMi and version R20090505 of YALMIP have been used in the SDP solver. All computations were carried out in double precision ($\varepsilon_{\text{mach}} = 2.2 \cdot 10^{-16}$) on an Intel Core Duo T7200 (2 GHz) with 2 GB memory, running Windows XP[5] and 32-bit version of MATLAB[6] R2009b. The maximum amount of memory that can be allocated by MATLAB on a 32-bit Windows XP operating system is 2 GB.[7]

5.1 A set of artificial systems

In the first set of examples, we investigate the sensitivity of the PRDE solvers with respect to two parameters: number of states (n) and length of period (T) of the associated periodic linear system. This is achieved by evaluating the accuracy of the solution of the PRDE associated with two artificial periodic linear systems, which have known solutions.

We start by considering a *linear time-invariant* (LTI) system

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad x(t_0) = x_0, \tag{5.1}$$

with n states and m inputs, i.e., $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, and search for a $u(t)$, for which the quadratic performance index

$$\int_0^\infty \Big[x^T Qx + u^T Ru\Big]\,dt,$$

[5] Windows is a registered trademark of Microsoft Corporation.

[6] MATLAB is registered trademark of The MathWorks, Inc.

[7] In practice, approximately only 1.2 GB is available.

is minimized. If the problem has a solution and the minimum is finite for any x_0, then, as known, the optimal control signal can be found as $u^*(t) = -Kx(t)$, where $K = R^{-1}B^TX$. The matrix $X \in \mathbb{R}^{n\times n}$ is the stabilizing solution of the algebraic Riccati equation (ARE)

$$A^TX + XA - XBR^{-1}B^TX + Q = 0. \tag{5.2}$$

To solve (5.2) an existing stable solver is used [4, 35], e.g., care in MATLAB or preferably slcaresc in SLICOT [44].

Next, the LTI system (5.1) is transformed into a periodic linear system by change of system coordinates: $z(t) = G(t)x(t)$, where $z(t)$ is the state vector in the new coordinates and

$$G(t) = \mathrm{diag}(\widehat{G}(t), \dots, \widehat{G}(t)) \quad \text{with } \widehat{G}(t) = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix},$$

for a given $\omega > 0$. Notably, the number of states in $x(t)$ is a multiple of two. This results in the T-periodic linear control system

$$\dot{z}(t) = \widetilde{A}(t)z(t) + \widetilde{B}(t)u(t), \tag{5.3}$$

where

$$\widetilde{A}(t) = \frac{dG(t)}{dt}G(t)^{-1} + G(t)AG(t)^{-1} \quad \text{and} \quad \widetilde{B}(t) = G(t)B,$$

with period $T = 2\pi/\omega$. The quadratic performance index (1.2) for the resulting transformed system (5.3) has the weighting matrices $Q(t) = G(t)^{-T}QG(t)^{-1}$ and $R(t) = R$. The optimal feedback of (5.3) can now be expressed as

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

where $\widetilde{X}(t_k) \equiv \widetilde{X}_k$ is the computed solution of the PRDE (1.1) at $t_k = (k-1)T/N$. The solution $\widehat{X}(t) := G(t)^{-T}XG(t)^{-1}$, where X is the solution of (5.2), corresponds to the exact solution at time t.

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 $\widehat{X}(t)$, computed as

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

where $\widehat{X}_k = \widehat{X}((k-1)T/N)$. The fact that the found approximation $\widetilde{X}(t)$ can indeed be used for stabilization of the linear system (5.3), is additionally checked by simulations on a closed-loop linear system modeled in Simulink.[8] In the simulation, the computed solution $\widetilde{X}(t)$ is approximated with a truncated Fourier expansion (like in (4.6)), and the initial states of (5.3) are all chosen to one.

[8] Simulink is registered trademark of The MathWorks, Inc.

Two periodic linear systems (5.3) are used for the tests, denoted System 1 and System 2. For both cases the weighting matrices Q and R of the corresponding LTI system (5.1) were identities. The matrix A in (5.1) associated with *System 1* is chosen as an upper triangular $n \times n$ random matrix and B is a dense $n \times 2$ random matrix. For both matrices we use uniformly distributed random numbers (MATLAB rand) on the open interval $(-10, 10)$.

The state matrix A in (5.1) associated with *System 2* is chosen as a Jordan block of size $n \times n$ associated with the zero eigenvalue (with ones on the super-diagonal) and B is chosen as e_n, the n-th unit vector. This choice of A and B corresponds to an LTI system with n integrators connected in series with a feedback controller applied to the n-th integrator [35] (see also [34]).

5.1.1 Evaluation of the numerical methods with respect to number of states

The PRDE solvers are evaluated with respect to the number of states n (order of the system) of System 1 and System 2 to answer the question: How does the number of states affect the accuracy of the computed solution and what is the largest number of states which can be addressed? For both System 1 and System 2 we use $\omega = 2$, so the period of the resulting periodic linear system (5.3) is $T = \pi$.

First we examine which ODE solver of ode45, ode113, or sgrk, is best suited for solving the linear Hamiltonian system (3.1) in the multi-shot methods (Step 1 of Algorithms 1 and 2). Table 1 indicates that the relative errors are of similar size, but as the number of states n increases the run-time for ode45 increases rapidly, especially for System 1. The preferred solver should either be ode113 or sgrk. For the coming tests, we mainly use the symplectic solver sgrk as it is marginally more accurate and faster than ode113.

Next, we solve the PRDE associated with the two systems using the four PRDE solvers (including the one-shot solver). Sample results are displayed in Table 2.

For the small-sized problems ($n \lesssim 20$), the accuracy and run-time for the two multi-shot solvers are quite similar. Both solvers compute a stabilizing solution up to around 26 states, after that the approximation of the solution is poor or deviate substantially from the trace of the reference solution. With increasing n and problem

Table 1 The accuracy of the PRDE solution and run-time using the multi-shot solver with different ODE solvers. Sample results are shown for increasing size n of the periodic linear systems: System 1 and System 2

		Relative error (run-time [sec])					
	n	Multi-shot, ode45		Multi-shot, ode113		Multi-shot, sgrk	
System 1	4	$7.1 \cdot 10^{-15}$	(6.6)	$1.4 \cdot 10^{-13}$	(3.8)	$8.0 \cdot 10^{-15}$	(3.6)
	16	$1.5 \cdot 10^{-9}$	(28)	$1.6 \cdot 10^{-9}$	(6.8)	$9.9 \cdot 10^{-10}$	(7.0)
	26	$6.3 \cdot 10^{-1}$	(117)	1.6	(17)	$4.1 \cdot 10^{-1}$	(15)
System 2	4	$2.9 \cdot 10^{-14}$	(5.8)	$2.2 \cdot 10^{-14}$	(3.2)	$2.0 \cdot 10^{-14}$	(3.1)
	16	$1.4 \cdot 10^{-8}$	(9.6)	$2.4 \cdot 10^{-8}$	(4.8)	$1.1 \cdot 10^{-8}$	(4.6)
	26	$2.9 \cdot 10^{-3}$	(21)	$1.4 \cdot 10^{-3}$	(11)	$2.7 \cdot 10^{-3}$	(8.6)

Table 2 Accuracy and run-time results for the PRDE solvers for different sizes n of the periodic linear systems: System 1 and System 2. N.S. denotes that the computed solution is not stabilizing

System 1; Relative error (run-time [sec])

n	Multi-shot, sgrk		Fast multi-shot, sgrk		SDP		One-shot, sgrk	
4	$4.4 \cdot 10^{-15}$	(3.6)	$4.8 \cdot 10^{-15}$	(3.6)	$2.1 \cdot 10^{-11}$	(7.3)	N.S.	(25)
10	$5.8 \cdot 10^{-12}$	(5.8)	$2.8 \cdot 10^{-11}$	(5.4)	$1.0 \cdot 10^{-10}$	(96)	N.S.	(61)
16	$9.3 \cdot 10^{-13}$	(7.4)	$2.1 \cdot 10^{-12}$	(7.6)	$1.4 \cdot 10^{-11}$	(1162)	N.S.	(95)
20	$1.1 \cdot 10^{-11}$	(10)	$7.9 \cdot 10^{-11}$	(10)	Out of memory	(–)	N.S.	(102)
26	$7.6 \cdot 10^{-7}$	(16)	N.S	(15)	Out of memory	(–)	N.S.	(140)
30	$6.6 \cdot 10^{-8}$	(16)	$7.4 \cdot 10^{-5}$	(16)	Out of memory	(–)	N.S.	(155)
36	$4.7 \cdot 10^{-6}$	(28)	N.S.	(29)	Out of memory	(–)	N.S.	(360)

System 2; Relative error (run-time [sec])

n	Multi-shot, sgrk		Fast multi-shot, sgrk		SDP		One-shot, sgrk	
4	$2.0 \cdot 10^{-14}$	(3.1)	$5.8 \cdot 10^{-15}$	(2.9)	$4.5 \cdot 10^{-11}$	(6.9)	$4.2 \cdot 10^{-2}$	(1.9)
10	$1.5 \cdot 10^{-11}$	(3.5)	$4.6 \cdot 10^{-12}$	(3.4)	$4.1 \cdot 10^{-11}$	(53)	$4.4 \cdot 10^{-2}$	(2.2)
16	$1.1 \cdot 10^{-8}$	(4.6)	$2.2 \cdot 10^{-9}$	(4.5)	Fail	(–)	$4.4 \cdot 10^{-2}$	(2.9)
20	$2.5 \cdot 10^{-6}$	(5.5)	$1.4 \cdot 10^{-7}$	(5.9)	Out of memory	(–)	$4.4 \cdot 10^{-2}$	(3.5)
26	$2.7 \cdot 10^{-3}$	(8.6)	$1.0 \cdot 10^{-4}$	(9.1)	Out of memory	(–)	$4.5 \cdot 10^{-2}$	(5.1)
30	$1.5 \cdot 10^{-1}$	(13)	$2.5 \cdot 10^{-2}$	(12)	Out of memory	(–)	N.S.	(5.9)
36	N.S.	(19)	2.1	(18)	Out of memory	(–)	N.S.	(11)

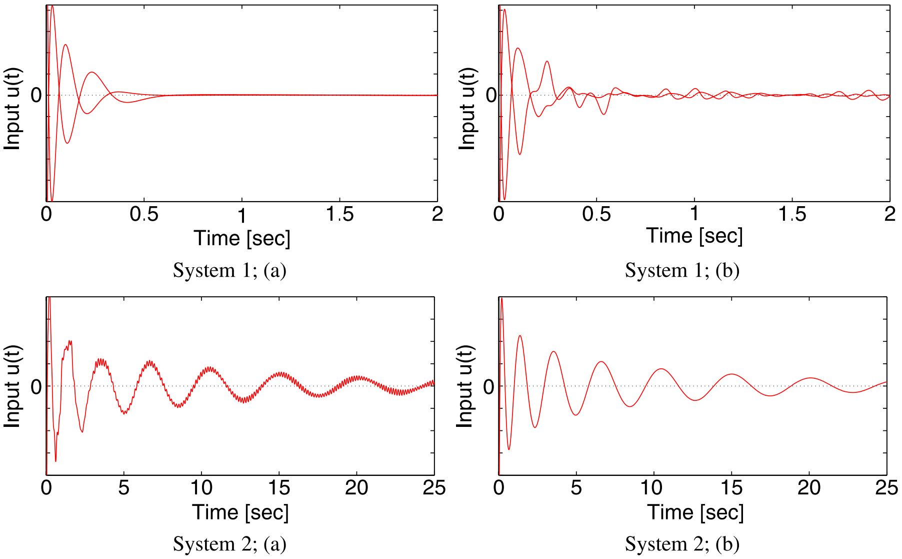
complexity, the computed PRDE solutions are examined by simulation on a closed-loop linear system. The *reliability* of the solvers is determined by examining the convergence of the input $u(t)$ to the zero reference solution. As an example we consider Fig. 1, which shows $u(t)$ for System 1 and System 2, respectively, with $n = 30$. The multi-shot solver is the better choice for System 1, whereas $X_{k+1}\Phi_{12}^{(k)} - \Phi_{22}^{(k)}$ in (3.7) becomes ill-conditioned in the fast multi-shot solver. On the other hand, the fast multi-shot solver is the better choice for System 2.

For both systems, the SDP solver runs out of memory when the number of states exceeds 16. The reason is the high number of variables together with the high dimension of the LMI constraints. Moreover, for System 2 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 n.

The one-shot solver encounters difficulties with both systems. For System 1 it cannot even compute a stabilizing solution for $n = 4$, and for System 2 non-smooth peaks in $u(t)$ appear in the closed-loop simulation already from $n = 4$.

5.1.2 Evaluation of the numerical methods with respect to length of period

The PRDE solvers are evaluated on System 1 and System 2 with different increasing periods T, while keeping the number of states constant ($n = 4$). As the period T is increased the constant N in (3.2) and (4.7) with $L = N$ is chosen as shown in Table 3.



System 1; (a)

System 1; (b)

System 2; (a)

System 2; (b)

Fig. 1 Results obtained by simulating a closed-loop linear system using System 1 and System 2, where $n = 30$ and the solution of PRDE is computed with: (**a**) multi-shot solver using sgrk, (**b**) fast multi-shot solver using sgrk. The graphs show the convergence of $u(t)$ to the zero reference solution

Table 3 Number of time instances N chosen for increasing period T

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 multi-shot solvers encounter difficulties with System 1 for large periods but have no difficulties computing a solution with high accuracy for System 2, see Table 4. For System 1, the multi-shot solvers using sgrk have a rather low accuracy already at $T = 2\pi \cdot 10^2$ and at a period of $2\pi \cdot 10^4$ (and larger) they fail to compute any solution due to a too large gap in the eigenvalues of the monodromy matrix. Particularly, the computed eigenvalues do not appear in reciprocal pairs. To illustrate the importance of using a symplectic ODE solver for problems with large period, also the results from the non-symplectic ode113 are presented.

The run-time and the memory requirement for the SDP solver are only depending on the choice of N and the number of states n; they are not affected by the period T of the system. For both systems, the SDP solver performs very good up to $T = 2\pi \cdot 10^4$. Only after that, when the number of time instances N is increased to 10000, the size of the LMI constraints gets too big and the SDP solver runs out of memory.

As can be expected, the one-shot solver cannot for any period compute a stabilizing solution for System 1, and for System 2 it can only do so for a period of 2π.

Table 4 Accuracy and run-time results for the PRDE solvers for different period lengths T of the periodic linear systems: System 1 and System 2. N.S. denotes that the computed solution is not stabilizing

System 1; Relative error (run-time [sec])

Period	Multi-shot, ode113		Multi-shot, sgrk		SDP	
2π	$1.2 \cdot 10^{-12}$	(3.6)	$2.1 \cdot 10^{-14}$	(4.0)	$3.3 \cdot 10^{-11}$	(7.3)
$2\pi \cdot 10$	$2.2 \cdot 10^{-12}$	(6.4)	$2.6 \cdot 10^{-11}$	(18)	$1.7 \cdot 10^{-11}$	(7.7)
$2\pi \cdot 10^2$	Fail	(–)	$3.1 \cdot 10^{-6}$	(116)	$3.6 \cdot 10^{-11}$	(7.8)
$2\pi \cdot 10^3$	Fail	(–)	$2.8 \cdot 10^{-7}$	(1156)	$2.2 \cdot 10^{-11}$	(65)
$2\pi \cdot 10^4$	Fail	(–)	Fail	(–)	$2.2 \cdot 10^{-11}$	(65)
$2\pi \cdot 10^5$	Fail	(–)	Fail	(–)	Out of memory	(–)
	Fast multi-shot, ode113		Fast multi-shot, sgrk		One-shot, sgrk	
2π	$1.2 \cdot 10^{-12}$	(3.5)	$2.0 \cdot 10^{-14}$	(4.0)	N.S.	(37)
$2\pi \cdot 10$	$2.9 \cdot 10^{-12}$	(6.3)	$2.6 \cdot 10^{-11}$	(18)	N.S.	(7.0)
$2\pi \cdot 10^2$	Fail	(–)	$3.1 \cdot 10^{-6}$	(115)	N.S.	(0.8)
$2\pi \cdot 10^3$	Fail	(–)	$2.8 \cdot 10^{-7}$	(1161)	N.S.	(1.0)
$2\pi \cdot 10^4$	Fail	(–)	Fail	(–)	Fail	(–)
$2\pi \cdot 10^5$	Fail	(–)	Fail	(–)	Fail	(–)

System 2; Relative error (run-time [sec])

Period	Multi-shot, ode113		Multi-shot, sgrk		SDP	
2π	$3.2 \cdot 10^{-14}$	(3.3)	$1.1 \cdot 10^{-14}$	(3.0)	$1.4 \cdot 10^{-10}$	(7.0)
$2\pi \cdot 10$	$5.6 \cdot 10^{-15}$	(4.1)	$7.2 \cdot 10^{-15}$	(4.1)	$8.8 \cdot 10^{-11}$	(7.0)
$2\pi \cdot 10^2$	$5.8 \cdot 10^{-12}$	(7.9)	$6.4 \cdot 10^{-12}$	(17)	$4.9 \cdot 10^{-11}$	(7.2)
$2\pi \cdot 10^3$	$7.7 \cdot 10^{-13}$	(70)	$6.4 \cdot 10^{-13}$	(168)	$2.4 \cdot 10^{-11}$	(63)
$2\pi \cdot 10^4$	Fail	(–)	$1.0 \cdot 10^{-11}$	(1602)	$2.3 \cdot 10^{-11}$	(64)
$2\pi \cdot 10^5$	Fail	(–)	$1.2 \cdot 10^{-12}$	(16698)	Out of memory	(–)
	Fast multi-shot, ode113		Fast multi-shot, sgrk		One-shot, sgrk	
2π	$1.9 \cdot 10^{-14}$	(3.2)	$5.9 \cdot 10^{-15}$	(2.9)	$4.2 \cdot 10^{-2}$	(1.9)
$2\pi \cdot 10$	$5.0 \cdot 10^{-15}$	(4.8)	$6.1 \cdot 10^{-15}$	(4.5)	N.S.	(47)
$2\pi \cdot 10^2$	$5.8 \cdot 10^{-12}$	(6.9)	$6.4 \cdot 10^{-12}$	(16)	N.S.	(10)
$2\pi \cdot 10^3$	$7.7 \cdot 10^{-13}$	(69)	$6.4 \cdot 10^{-13}$	(164)	N.S.	(7.3)
$2\pi \cdot 10^4$	Fail	(–)	$1.1 \cdot 10^{-11}$	(1582)	N.S.	(1.4)
$2\pi \cdot 10^5$	Fail	(–)	$2.4 \cdot 10^{-12}$	(16614)	N.S.	(2.9)

5.2 Examples of orbital stabilization of cycles for mechanical systems

The solvers are also 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 [21].

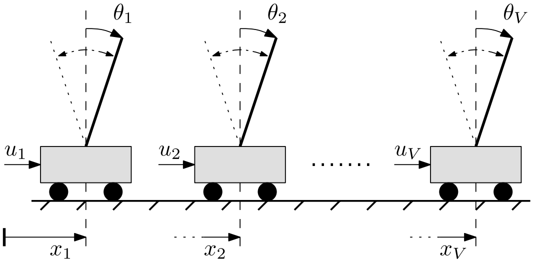


Fig. 2 V identical cart-pendulum systems. The coordinates $x_1, \dots, x_V$ represent positions of the carts along the horizontal axis, $\theta_1, \dots, \theta_V$ are the angles of the pendulums with respect to the vertical axis, and $u_1, \dots, u_V$ are the control inputs

5.2.1 Synchronization of oscillations of cart-pendulum systems

The first example is stable synchronization of forced oscillations of V copies of identical cart-pendulum systems around their unstable equilibriums,[9] see Fig. 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

$$2\ddot{x}_i + \cos(\theta_i)\ddot{\theta}_i - \sin(\theta_i)\dot{\theta}_i^2 = u_i \quad \text{and} \quad \cos(\theta_i)\ddot{x}_i + \ddot{\theta}_i - g\sin(\theta_i) = 0, \tag{5.4}$$

where $i = 1, \dots, V$ and g is the acceleration due to gravity. The system has $2V$ degrees of freedom (x_i, θ_i) and V control variables (u_i).

Planning a cycle: Suppose the twice continuously differentiable function $\phi(\cdot)$ is chosen such that the invariance of the relations

$$x_1 = \phi(\theta_1), \qquad x_2 = \phi(\theta_2), \quad \dots, \quad x_V = \phi(\theta_V), \tag{5.5}$$

results in V identical equations with $\theta = \theta_i$, $i = 1, \dots, V$,

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = 0, \tag{5.6}$$

which has a T-periodic solution[10] $\theta_\star(t) = \theta_\star(t + T)$. Here

$$\alpha(\theta) = \cos(\theta)\phi'(\theta) + 1, \qquad \beta(\theta) = \cos(\theta)\phi''(\theta), \quad \text{and} \quad \gamma(\theta) = -g\sin(\theta).$$

The solutions written in pairs for all systems

$$\big[\theta_1 = \theta_\star(t),\ x_1 = \phi\left(\theta_\star(t)\right)\big], \quad \dots, \quad \big[\theta_V = \theta_\star(t),\ x_V = \phi\left(\theta_\star(t)\right)\big], \tag{5.7}$$

are the synchronous oscillations of all V cart-pendulum systems.

Orbital stabilization of (5.7) can be achieved from a stabilization of the origin of the linearization of the transverse dynamics of (5.4) along the cycle (5.7). As shown in [21], the transverse linearization of (5.4) along the cycle (5.7) is the T-periodic linear

[9] The steps for motion planning and analytical arguments for controller design are from [21].

[10] The way to plan a cycle for one cart-pendulum system and to make it orbitally stable is described in [40].

control system, whose coefficients, the matrices $A(t)$ and $B(t)$, can be computed as

$$\frac{d}{dt}\begin{bmatrix} I_\bullet \\ y_\bullet \\ \dot{y}_\bullet \end{bmatrix} = \begin{bmatrix} a_{11}(t) & 0 & 0 \\ 0 & 0 & I_{2V-1} \\ 0 & A_{32}(t) & A_{33}(t) \end{bmatrix}\begin{bmatrix} I_\bullet \\ y_\bullet \\ \dot{y}_\bullet \end{bmatrix} + \begin{bmatrix} b_{11}(t) & 0 & 0 \\ 0 & 0 & 0 \\ B_{31}(t) & \cdots & B_{3V}(t) \end{bmatrix} v_\bullet, \tag{5.8}$$

where the state vector has $n = 4V - 1$ states with $I_\bullet \in \mathbb{R}^1$, $y_\bullet \in \mathbb{R}^{2V-1}$, $\dot{y}_\bullet \in \mathbb{R}^{2V-1}$ and the control vector $v_\bullet$ has $m = V$ control inputs, $v_\bullet \in \mathbb{R}^V$.

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

$$\begin{aligned} A_{32} &= \operatorname{diag}(0, \dots, 0 \mid a(t), \dots, a(t)), \\ A_{33} &= \operatorname{diag}(0, \dots, 0 \mid a_{11}(t), \dots, a_{11}(t)), \\ B_{31} &= [1, 0, 0, \dots, 0 \mid b(t), b(t), \dots, b(t)]^T, \\ B_{32} &= [0, 1, 0, \dots, 0 \mid -b(t), 0, \dots, 0]^T, \\ B_{33} &= [0, 0, 1, \dots, 0 \mid 0, -b(t), 0, \dots, 0]^T, \qquad \dots, \\ B_{3V} &= [0, 0, \dots, 0, 1 \mid 0, \dots, 0, -b(t)]^T, \end{aligned}$$

with $a(t) = \frac{[\beta(\theta_\star(t))\dot{\theta}_\star^2(t)+\gamma(\theta_\star(t))]\alpha'(\theta_\star(t))}{\alpha^2(\theta_\star(t))} - \frac{\beta'(\theta_\star(t))\dot{\theta}_\star^2(t)+\gamma'(\theta_\star(t))}{\alpha(\theta_\star(t))}$ and $b(t) = -\frac{\cos(\theta_\star(t))}{\alpha(\theta_\star(t))}$. We use the |-notation above (instead of a comma sign) in order to show that everything to the left of | is of size (length) V and everything to the right is of size $V - 1$. As argued in [40], the function $\phi(\cdot)$ in (5.5) can be chosen to meet various specifications on a periodic motion, e.g., its period, amplitude etc. For instance, with the choice

$$\phi(\theta) = -\left[1 + \frac{g}{\omega^2}\right]\log\left(\frac{1+\sin(\theta)}{\cos(\theta)}\right) \tag{5.9}$$

there are oscillations of each of the cart-pendulum systems around their unstable equilibria of period $T \approx 2\pi/\omega$, where $\omega > 0$.

By solving the PRDE for the periodic linear control system (5.8) with reasonably chosen performance index we find a stabilizing controller for transverse linearization and can construct a nonlinear controller which synchronously stabilize the oscillations of the pendulums and carts [21]. For the PRDE, we use the constant weighting matrix $R = I_V$ and the $(4V - 1) \times (4V - 1)$ time-varying diagonal weighting matrix

$$Q(t) = \operatorname{diag}(f_*(t), 3, \dots, 3), \quad \text{where } f_*(t) = \frac{10}{\sqrt{\dot{\theta}(t)^2 + \ddot{\theta}(t)^2}}.$$

Using the one-shot solver together with sgrk we are only able to compute a stabilizing solution for two carts. Using the multi-shot solver together with sgrk, we successfully compute (limited by the memory) a stabilizing solution for 40 carts ($n = 159$ and $m = 40$) and with the fast multi-shot solver together with sgrk for 50 carts ($n = 199$ and $m = 50$). Figure 3 shows the simulation of the closed-loop nonlin-

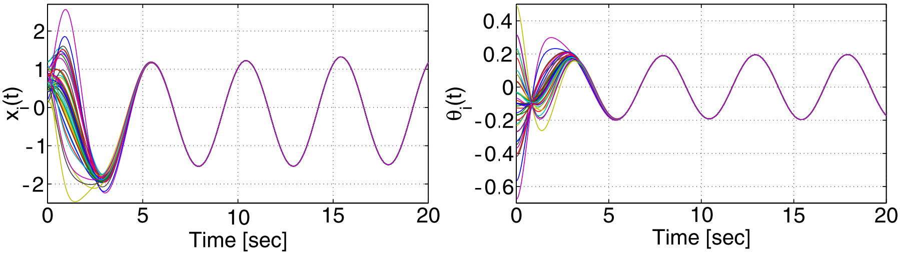


Fig. 3 Simulation of $V = 40$ pendulums on carts. The coordinates x_i, $i = 1, \ldots, 40$, are the positions of the carts along the horizontal axis (*left figure*). The angles θ_i, $i = 1, \ldots, 40$, are the angles of the pendulums with respect to the vertical axis (*right figure*)

ear system[11] for 40 carts simulated over 20 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 do not run into any numerical difficulties with the solvers and it is possible to compute a stabilizing solution for a much higher number of carts. For example, on a 64-bit Linux operating system with the 64-bit version of MATLAB (which provides the possibility to access up to 8TB of memory), the PRDE has successfully been solved for (at least) 150 carts. However, the available memory is still a limit on how high we can increase the number of states of the system.

The SDP solver, however, can only compute a stabilizing solution for three carts. For a larger number of carts the memory requirement is too high. To conclude, it is not suitable to use the SDP solver for this problem with a large number of carts. For example, with 40 carts the SDP problem has 521520 variables and 19900 constraints.

5.2.2 *Orbital stabilization of Furuta pendulum*

The Furuta pendulum [22] is a mechanical system with two degrees of freedom (see Fig. 4), 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. The equations of motion of the Furuta pendulum are [19]:

$$\begin{aligned}(p_1 + p_2\sin^2(\theta))\ddot{\phi} + p_3\cos(\theta)\ddot{\theta} + p_2\sin(2\theta)\dot{\theta}\dot{\phi} - p_3\sin(\theta)\dot{\theta}^2 &= \tau_\phi,\\ p_3\cos(\theta)\ddot{\phi} + (p_2 + p_5)\ddot{\theta} - p_2\sin(\theta)\cos(\theta)\dot{\phi}^2 - p_4\sin(\theta) &= 0,\end{aligned} \tag{5.10}$$

where τ_ϕ is the external torque that allows us to control the rotation of the arm. The constants $p_1, \ldots, p_5$ are positive and defined by physical parameters of the system. For the present setup (see Fig. 4), they are:

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

[11] See [21] for a nonlinear feedback design for (5.4) based on stabilization of transverse linearization (5.8).

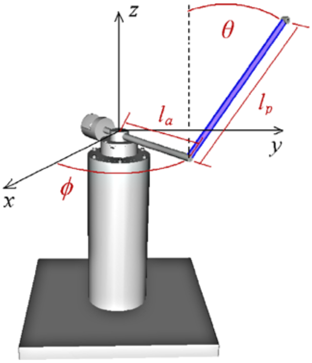


Fig. 4 An illustration of the Furuta pendulum built at Department of Applied Physics and Electronics, Umeå University, where length of arm $l_a = 0.15$ [m] and length of pendulum $l_p = 0.26$ [m]. Moreover, mass of arm is 0.298 [kg], mass of pendulum is $3.2 \cdot 10^{-2}$ [kg], and mass of bob at the end of the pendulum is $7.5 \cdot 10^{-3}$ [kg]

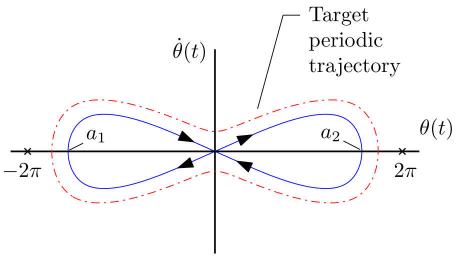


Fig. 5 Two homoclinic curves of the equilibrium at $\theta = 0$ are shown on the phase portrait. One intersects the θ-axis at a_1 and the other at a_2. The *dashed line* illustrates one example of a periodic trajectory orbiting the two homoclinic curves [20]

To plan forced oscillations of the Furuta pendulum with large periods, we suggest searching for its homoclinic curves. It is shown in [20] that the geometrical relation kept invariant between the angles of the Furuta pendulum by a control input

$$\phi = \xi \arctan(\theta), \qquad \text{with } \frac{0.01(p_2 + p_5)}{p_3} < \xi < \frac{6.6(p_2 + p_5)}{p_3}, \tag{5.11}$$

results in the following reduced dynamics of the θ-angle

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

with $\gamma(\theta) = -p_4 \sin(\theta)$ and

$$\alpha(\theta) = p_2 + p_5 + \frac{\xi p_3 \cos(\theta)}{(1+\theta^2)}, \qquad \beta(\theta) = \frac{-\xi \cos(\theta)}{(1+\theta^2)^2}\left(2p_3\theta + \xi p_2 \sin(\theta)\right).$$

For the choice (5.11) the upright equilibrium $\theta = 0$ of this system has two homoclinic curves surrounded by periodic solutions, see Fig. 5. Their periods tend to infinity as long as the initial conditions are chosen close to the homoclinic curves. The linearization of transverse dynamics along any such nontrivial periodic motion

$$\theta_\star(t) = \theta_\star(t+T), \qquad \phi_\star(t) = \xi \arctan(\theta_\star(t)), \tag{5.12}$$

can be computed analytically and has the form

$$\frac{d}{dt}\begin{bmatrix} I_\bullet \\ y_\bullet \\ \dot{y}_\bullet \end{bmatrix} = \begin{bmatrix} a_{11}(t) & 0 & a_{13}(t) \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_\bullet \\ y_\bullet \\ \dot{y}_\bullet \end{bmatrix} + \begin{bmatrix} b_1(t) \\ 0 \\ 1 \end{bmatrix} v_\bullet, \tag{5.13}$$

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

$$a_{11}(t) = -\frac{2\dot{\theta}_\star(t)\beta(\theta_\star(t))}{\alpha(\theta_\star(t))}, \qquad a_{13}(t) = \frac{4\xi p_2 \sin\theta_\star(t)\cos\theta_\star(t)\dot{\theta}_\star^2(t)}{(1+\theta_\star^2(t))\alpha(\theta_\star(t))},$$

$$b_1(t) = -\frac{2\dot{\theta}_\star(t)p_3\cos(\theta_\star(t))}{\alpha(\theta_\star(t))}.$$

Following the arguments of [21], we can synthesize the orbitally stabilizing controller for the cycle (5.12) provided that we stabilize the origin of the transverse linearization (5.13). To this end, the gain ξ in (5.11) has been chosen equal to $1/\pi$. For stabilizing the origin of (5.13) we search for a stabilizing solution of the PRDE with the constant weighting matrix $R = 10$ and with the time-varying weighting matrix

$$Q(t) = \operatorname{diag}(f_*(t), \overline{f}_*, 0.01), \quad \text{where } f_*(t) = 0.05\sqrt{\dot{\theta}(t)^2 + \ddot{\theta}(t)^2},$$

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

By using the one-shot solver together with ode45 to solve the PRDE, a periodic trajectory with the period $T \approx 4.045$ 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 $T \approx 8.249$ seconds can be found by using the SDP solver.

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

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

where $\varepsilon \neq 0$. 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]

Using the SDP solver, the largest period for which we find an orbital stabilizing solution is $T \approx 8.249$ seconds (with $\varepsilon = 0.004$). Using any of the multi-shot solvers together with sgrk, the largest period is $T \approx 8.095$ seconds (with $\varepsilon = 0.005$). Figure 6 shows the desired periodic cycle of the Furuta pendulum with $T \approx 8.095$, and Fig. 7 displays the results from the simulation of the closed-loop nonlinear system using the PRDE solutions from the SDP solver and the multi-shot solver together with sgrk (the results for the fast multi-shot solver are the same). The initial states used are $\theta(0) = \pi$ and $\phi(0) = \dot{\phi}(0) = \dot{\theta}(0) = 0$, which correspond to the pendulum in

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

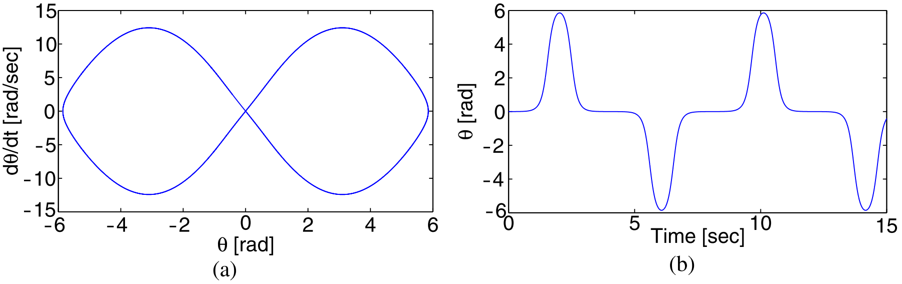


Fig. 6 (**a**) Phase portrait of the Furuta pendulum with period 8.095 seconds. (**b**) Periodic trajectory of θ as a function of time

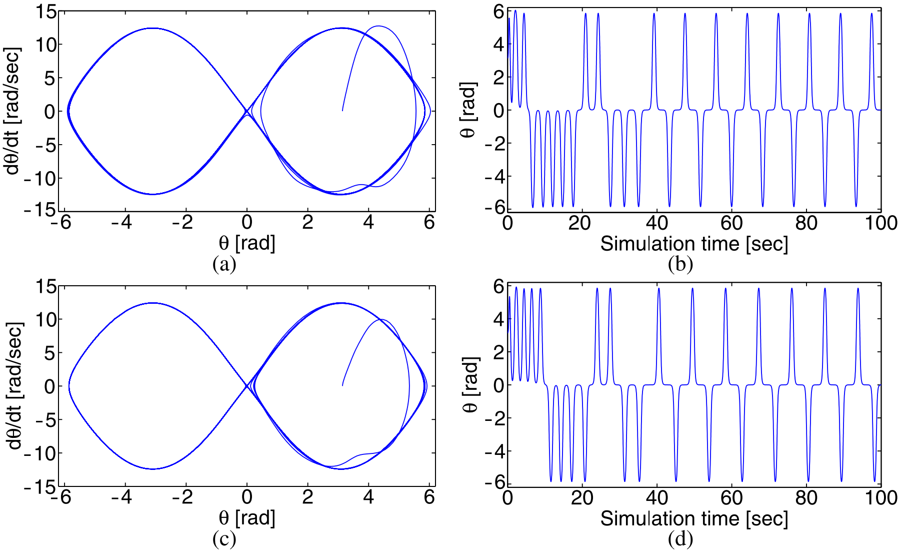


Fig. 7 The resulting phase portrait (*left*) and the periodic trajectory of θ (*right*) from simulation of orbital stabilizing solutions with a periodic trajectory of 8.095 seconds: (**a**)–(**b**) using the PRDE solution from the SDP solver. (**c**)–(**d**) using the PRDE solution from the multi-shot solver

downwards position. As we can see both solutions converge to a periodic trajectory, however, the solutions from the multi-shot solvers converge both faster and more accurately towards the desired phase portrait of the Furuta pendulum.

6 Evaluation summary and conclusions

We have implemented three methods to solve the PRDE: (1) the multi-shot method using the ordered PRSF to compute a stable invariant subspace of a cyclic matrix product (Sect. 3.4); (2) the multi-shot method using the fast algorithm to compute a stable deflating subspace of a lifted pencil (Sect. 3.5); and (3) the SDP method based on a convex optimization approach (Sect. 4). The implementations of these three methods are called the *multi-shot solver*, the *fast multi-shot solver*, and the *SDP solver*, respectively. Additionally, the classical periodic generator method has been implemented and is called the *one-shot solver* (Sect. 3.2). The realizations have been evaluated and compared on challenging problems including artificially constructed periodic linear control systems with known solutions (Sect. 5.1), and on problems, for which analytical solutions are unknown, from two real-world experimental studies (Sect. 5.2). They are associated with the task of orbital stabilization of forced oscillations in controlled mechanical systems. In the first example, the PRDE solvers are used to stabilize a linearization of transverse dynamics for synchronous forced oscillations of a family of cart-pendulum systems (Sect. 5.2.1), and in the second, to stabilize a linearization of transverse dynamics for a forced periodic motion of the Furuta pendulum (Sect. 5.2.2).

For the two artificial systems the *reliability* of the solutions are evaluated both quantitatively (relative errors and run-time results) and qualitatively (reliability measurements of the stabilizing solution). For the mechanical systems, the reliability is measured only qualitatively. In our evaluation, we have both considered the impact of the size n of the PRDE and the length of the period T upon the computed solutions for the different methods considered. These two problem parameters directly affect the complexity of solving a PRDE (1.1) numerically. In Tables 5 and 6, we display a summary of our findings from the numerical experiments documented in Sect. 5. Note that we have only included solutions that can be considered highly reliable. For example, the case $n = 30$ for System 2 solved with the multi-shot method using sgrk is not considered highly reliable, since the input $u(t)$ in the simulated closed-loop linear system oscillates as it converges to the reference solution, see Fig. 1.

Table 5 Summary of test results for the artificial systems. Each number in the table shows the largest n or T which produces a *highly reliable* stabilizing solution for each solver and test case. Bold font indicates the best case(s) for each system, and a dash (–) that no reliable stabilizing solution is found

Test case	System	Multi-shot		Fast multi-shot		SDP	One-shot
		sgrk	ode113	sgrk	ode113		sgrk
States n	System 1	**36**	N/A	20	N/A	16	–
	System 2	26	N/A	**30**	N/A	10	–
Period T	System 1	$2\pi \cdot 10^3$	$2\pi \cdot 10$	$2\pi \cdot 10^3$	$2\pi \cdot 10$	$\mathbf{2\pi \cdot 10^4}$	–
	System 2	$\mathbf{2\pi \cdot 10^5}$	$2\pi \cdot 10^3$	$\mathbf{2\pi \cdot 10^5}$	$2\pi \cdot 10^3$	$2\pi \cdot 10^4$	–

Table 6 Summary of test results for the mechanical systems. Each number in the table shows the largest number of cart-pendulum systems V or largest period T for the Furuta pendulum, which produces a *highly reliable* stabilizing solution for each solver and system. Bold font indicates the best case

Mechanical system		Multi-shot	Fast multi-shot	SDP	One-shot
		sgrk	sgrk		sgrk/ode45
Cart-pendulum	$n, V = m$	159, 40	199, **50**	11, 3	7, 2
Furuta pendulum	T	8.095	8.095	**8.249**	4.045

The multi-shot and SDP methods are significantly better (with respect to accuracy as well as stability) compared to the periodic generator method. Our results show that the SDP solver is the best solver for small-sized systems ($n \lesssim 20$) with large periods T. The run-time for the SDP solver is independent of the period, and therefore this solver is to be preferred for such systems as long as the memory requirement is satisfied.

Moreover, our results show that the multi-shot solvers better cope with medium-sized systems ($20 \lesssim n \lesssim 500$). Generally, the two multi-shot solvers produce rather similar results both with respect to run-time and accuracy. For solving the underlying Hamiltonian system the preferred ODE solver is a symplectic solver like the symplectic Gauss Runge-Kutta. It is especially important to use a symplectic solver for systems with large periods. Impressively, the multi-shot solver computing a reordered periodic real Schur form handles cyclic matrix products of at least size $N = 10000$.

One major limitation of the SDP solver is the high storage requirement, which depends on the size of the system considered. This memory issue can be critical, for example, when the PRDE must be solved online in a physical setup or in real-world (including offline) applications with a large number of degrees of freedom.

Conclusively, our numerical evaluation of the PRDE solvers demonstrates that no single solver is the best for all examples considered. Depending on the problem parameters n and T the choice should be one of the two multi-shot solvers or the SDP solver. These three solvers always outperform the classical periodic generator method.

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. We also thank the referees for their constructive comments.

References

1. Abou-Kandil, H., Freiling, G., Ionescu, V., Jank, G.: Matrix Riccati equations. In: Control and Systems Theory. Birkhäuser, Basel (2003). ISBN 3-7643-0085-X
2. Anderson, B., Feng, Y.: 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. Anderson, B., Moore, J.: Optimal Control: Linear Quadratic Methods. Dover, New York (2007). ISBN 0486457664
4. Arnold, W., Laub, A.: Generalized eigenproblem algorithms and software for algebraic Riccati equations. In: Proc. IEEE, vol. 72, pp. 1746–1754 (1984)
5. Benner, P., Byers, R.: Evaluating products of matrix pencils and collapsing matrix products. Numer. Linear Algebra Appl. **8**, 357–380 (2001)
6. Benner, P., Byers, R., Mayo, R., Quintana-Orti, E.S., Hernandez, V.: Parallel algorithms for LQ optimal control of discrete-time periodic linear systems. J. Parallel Distrib. Comput. **62**, 306–325 (2002)
7. Bittanti, S., Colaneri, P.: Periodic Systems: Filtering and Control. Springer, Berlin (2009). ISBN 978-1-84800-910-3
8. Bittanti, S., Colaneri, P., De Nicolao, G.: A note on the maximal solution of the periodic Riccati equation. IEEE Trans. Automat. Control **34**(12), 1316–1319 (1989)
9. Bittanti, S., Colaneri, P., Guardabassi, G.: Analysis of the periodic Lyapunov and Riccati equations via canonical decomposition. SIAM J. Control Optim. **24**(6), 1138–1149 (1986)
10. Bittanti, S., Colaneri, P., De Nicolao, G.: The periodic Riccati equation. In: Bittanti, S., Laub, A.J., Willems, J.C. (eds.) The Riccati Equation, pp. 127–162. Springer, Berlin (1991), Chap. 6

11. Bojanczyk, A., Golub, G.H., Van Dooren, P.: The periodic Schur decomposition; algorithm and applications. In: Luk, F.T. (ed.) Proc. SPIE Conference, vol. 1770, pp. 31–42. SPIE, Bellingham (1992)
12. Calvo, M., Sanz-Serna, J.: Numerical Hamiltonian Problems. Chapman & Hall, London (1994)
13. Chen, Y.Z., Chen, S.B., Liu, J.Q.: Comparison and uniqueness theorems for periodic Riccati differential equations. Int. J. Control **69**(3), 467–473 (1998)
14. Chu, E., Fan, H., Lin, W., Wang, C.: Structure-preserving algorithms for periodic discrete-time algebraic Riccati equations. Int. J. Control **77**, 767–788 (2004)
15. Dieci, L.: Numerical integration of the differential Riccati equation and some related issues. SIAM J. Numer. Anal. **29**(3), 781–815 (1992)
16. Dieci, L., Eirola, T.: Positive definiteness in the numerical solution of Riccati differential equations. Numer. Math. **67**, 303–313 (1994)
17. Franco, J., Gómez, I.: Fourth-order symmetric DIRK methods for periodic stiff problems. Numer. Algorithms **32**, 317–336 (2003)
18. Freidovich, L., Gusev, S., Shiriaev, A.: 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. Freidovich, L., Johansson, R., Robertsson, A., Sandberg, A., Shiriaev, A.: Virtual-holonomic-constraints-based design of stable oscillations of Furuta pendulum: Theory and experiments. IEEE Trans. Robot. **23**(4), 827–832 (2007)
20. Freidovich, L., La Hera, P., Mettin, U., Shiriaev, A.: New approach for swinging up the Furuta pendulum: Theory and experiments. Mechatronics **19**(8), 1240–1250 (2009)
21. Freidovich, L., Gusev, S., Shiriaev, A.: Transverse linearization for controlled mechanical systems with several passive degrees of freedom. IEEE Trans. Automat. Contr. (2010, in press). doi:10.1109/TAC.2010.2042000
22. Furuta, K., Yamakita, M., Kobayashi, S.: Swing up control of inverted pendulum. In: Proc. of IECON'91, Kobe, Japan (1991)
23. Granat, R., Kågström, B.: Direct eigenvalue reordering in a product of matrices in periodic Schur form. SIAM J. Matrix Anal. Appl. **28**(1), 285–300 (2006)
24. Granat, R., Kågström, B., Kressner, D.: Matlab tools for solving periodic eigenvalue problems. In: Proc. of the 3rd IFAC Workshop, PSYCO'07, St. Petersburg, Russia (2007)
25. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration: Structure-preserving Algorithms for Ordinary Differential Equations, 2nd edn. Springer, Berlin (2006). ISBN 3-540-30663-3
26. Hairer, E., McLachlan, R., Razakarivony, A.: Achieving Brouwer's law with implicit Runge-Kutta methods. BIT **48**(2), 231–243 (2008)
27. Hench, J.J., Laub, A.J.: Numerical solution of the discrete-time periodic Riccati equation. IEEE Trans. Automat. Contr. **39**(6), 1197–1209 (1994)
28. Hench, J.J., Kenney, C.S., Laub, A.J.: Methods for the numerical integration of Hamiltonian systems. Circuits Syst. Signal Process. **13**(6), 695–732 (1994)
29. Hu, G.: Symplectic Runge-Kutta methods for the Kalman-Bucy filter. IMA J. Math. Control Info. (2007)
30. Johansson, S.: Tools for control system design—stratification of matrix pairs and periodic Riccati differential equation solvers. Ph.D. Thesis, Report UMINF 09.04, Department of Computing Science, Umeå University, Sweden (2009). ISBN 978-91-7264-733-6
31. Johansson, S., Kågström, B., Shiriaev, A., Varga, A.: 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)
32. Kågström, B., Poromaa, P.: Computing eigenspaces with specified eigenvalues of a regular matrix pair (A, B) and condition estimation: theory, algorithms and software. Numer. Algorithms **12**, 369–407 (1996)
33. Kano, H., Nishimura, T.: Periodic solutions of matrix Riccati equations with detectability and stabilizability. Int. J. Control **29**(3), 471–487 (1979)
34. Kressner, D., Mehrmann, V., Penzl, T.: CTDSX—a collection of benchmark examples for state-space realizations of continuous-time dynamical systems. SLICOT Working Note 1998-9, WGS (1998)
35. Laub, A.: A Schur method for solving algebraic Riccati equations. IEEE Trans. Automat. Contr. **AC-24**, 913–921 (1979)
36. Leimkuhler, B., Reich, S.: Simulating Hamiltonian Dynamics. Cambridge University Press, Cambridge (2004). ISBN 0-521-77290-7
37. Löfberg, J.: YALMIP homepage. Automatic Control Laboratory, ETH Zurich, Switzerland (2009). http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php

38. Lust, K.: psSchur homepage. Department of Mathematics, K.U. Leuven, Belgium (2009). http://perswww.kuleuven.be/~u0006235/ACADEMIC/r_psSchur.html
39. Mehrmann, V.: The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution. Lecture Notes in Control and Information Sciences, vol. 163. Springer, Berlin (1991)
40. Perram, J., Robertsson, A., Sandberg, A., Shiriaev, A.: Periodic motion planning for virtually constrained mechanical system. Syst. Control Lett. **55**(11), 900–907 (2006)
41. Reid, W.: Riccati Differential Equations. Academic Press, San Diego (1972)
42. SeDuMi homepage. Advanced Optimization Laboratory, McMaster University, Canada (2009). http://sedumi.ie.lehigh.edu/
43. Sima, V.: Algorithms for Linear-Quadratic Optimization. Pure and Applied Mathematics, vol. 200. Dekker, New York (1996)
44. SLICOT homepage. Germany (2008). http://www.slicot.org
45. Sturm, J.: Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones (updated for version 1.05). Tech. Rep., Department of Econometrics, Tilburg University, Tilburg, The Netherlands (2001)
46. Tan, S., Zhong, W.: Numerical solutions of linear quadratic control for time-varying systems via symplectic conservative perturbation. Appl. Math. Mech. **28**(3), 277–287 (2007)
47. Varga, A.: On solving periodic differential matrix equations with applications to periodic system norms computation. In: Proc. of CDC’05, Seville, Spain (2005)
48. Varga, A.: A periodic systems toolbox for MATLAB. In: Proc. of 16th IFAC World Congress, Prague, Czech Republic (2005)
49. Varga, A.: On solving periodic Riccati equations. Numer. Linear Algebra Appl. **15**(9), 809–835 (2008)
50. Yakubovich, V.: Linear-quadratic optimization problem and frequency theorem for periodic systems. Sib. Math. J. **27**(4), 181–200 (1986)