F. Topputo, M. Miani, F. Bernelli Zazzera
Optimal Selection of the Coefficient Matrix in State-Dependent Control Methods
Journal of Guidance Control and Dynamics, Vol. 38, N. 5, 2015, p. 861-873
doi:10.2514/1.G000136

Optimal Selection of the Coefficient Matrix in State-Dependent Control Methods

Francesco Topputo*

Politecnico di Milano, Milano, 20156, Italy

Martino Miani†

Politecnico di Milano, Milano, 20156, Italy

Franco Bernelli-Zazzera‡

Politecnico di Milano, Milano, 20156, Italy

This paper presents a state-dependent method for solving nonlinear, control-affine, finite-time optimal control problems. The problem is set in a state-dependent form and solved as a sequence of time-varying linear quadratic regulators, for which an approach based on state transition matrices has been developed. The novelty of the method consists in the possibility of managing a number of different state-dependent coefficient factorizations of the uncontrolled dynamics, for use with various state-dependent control methods. At each iteration, a nonlinear programming problem is solved to select the state-dependent combination that maximizes the controllability, so minimizing the control effort and, likely, the objective function. Test problems are presented to show the effectiveness of the method.

*Assistant Professor, Department of Aerospace Science and Technology, Via La Masa, 34. E-mail: francesco.topputo@polimi.it

†M.Sc., Department of Aerospace Science and Technology, Via La Masa, 34. E-mail: martino.miani@mail.polimi.it

‡Full Professor, Department of Aerospace Science and Technology, Via La Masa, 34. AIAA Senior Member. E-mail: franco.bernelli@polimi.it

I. Introduction

Optimal control problems are solved with indirect or direct methods. The former stems from the calculus of variations;[1,2] the latter use a nonlinear programming (NLP) optimization.[3,4] Both methods require the solution of a complex set of equations (Euler–Lagrange differential equations or Karush–Kuhn–Tucker algebraic equations, respectively) for which iterative numerical methods are used. These iterative procedures implement some form of Newton's method to find the zeros of a nonlinear function. They are initiated by providing an initial guess solution. Guessing an appropriate initial solution is not trivial, and requires a deep knowledge of the problem at hand. In indirect methods, the initial value of the Lagrange multiplier has to be provided, whose lack of physical meaning makes it difficult to formulate a good guess. In direct methods, the initial trajectory and control have to be guessed at discrete points over the whole time mesh.

Due to their inherent complexity, optimal control problems are often solved within simplified schemes, where some form of approximation is carried out. These methods allow circumventing the original, nonlinear Euler–Lagrange equations, and thus they do not require guessing the initial Lagrange multipliers. Nonetheless, the price to pay is the loss of optimality, since suboptimal solutions are derived. State-dependent methods[5,6] belong to this category. The state-dependent Riccati equations (SDRE)[7–10] is likely the most known example of approximated method due to its simplicity and effectiveness in many applications[11–16]. This method treats the original *infinite-horizon*, nonlinear optimal control problem as an infinite-horizon linear-quadratic regulator (LQR), pointwise. A number of LQR problems are solved sequentially at each time instant in which the time domain is discretized. This is done by using state-dependent matrices, which are evaluated pointwise at each time step. With the SDRE the *closed-loop* control is treated, the control law being function of the present state. The SDRE method can also be used to solve finite-horizon optimal control;[10] one way consists in choosing the state-dependent matrices as functions of the time-to-go.[17]

An alternative approach for the *finite-horizon* optimal control is represented by the approximated sequence of Riccati equations (ASRE)[18,19]. This method requires again representing the dynamics and the objective function through state-dependent coefficient (SDC) matrices. A number of finite-horizon, time-varying linear-quadratic regulator (TVLQR) problems are solved iteratively. The SDC matrices are evaluated with the solution at the previous iteration. It is worth remarking the differences between SDRE and ASRE. With ASRE, the *open-loop* optimal guidance problem is treated, and the end-to-end solution is found iteratively. That is, as the SDRE solves a sequence of pointwise LQR at different times labels (t_k, t_{k+1}, ...), the ASRE iterates on the solution defined over the whole time domain $[t_i,\ t_f]$; t_i, t_f are the initial, final times, respectively. In brief, i) the SDRE solves

the infinite-horizon, closed-loop optimal control *sequentially* by using pointwise LQR; ii) the ASRE solves the finite-horizon, open-loop optimal control *iteratively* by using TVLQR. The TVLQR can be solved with a state transition matrix approach, so avoiding dealing with the differential Riccati equation.

The ASRE method can be implemented to solve a variety of problems.[20–22] To initiate the algorithm, a certain state-dependent representation for the dynamics must be provided. As such, the degree of freedom represented by the initial guess for the Lagrange multiplier is replaced by the choice of the system factorization. Indeed, although one would expect that different SDC matrices lead to the same result, this is not the case, as it is clearly shown in this paper. Moreover, when different SDC matrices are provided, different results in terms of system trajectory and cost function are found. In principle, one has to run the algorithm a number of times, with different state-dependent representations, and choose the best, feasible solution *a posteriori*.

In this paper the focus is, instead, on modifying the ASRE algorithm to allow managing a set of SDC factorizations in a simultaneous way. Given a set of parent SDC matrices, these are optimally combined to find the one that improves the algorithm convergence, the cost function, and the system controllability. Indeed, it is shown that the latter feature can be directly linked to the solution cost. The higher is the controllability, the less control effort the system will use to enforce the boundary conditions and to minimize the cost function. This is achieved by maximizing the controllability Gramian at each algorithm iteration, which involves solving a NLP optimization problem. More specifically, the minimum singular value of the Gramian is maximized. It is shown that this is the most suitable indicator for the purposes of this paper. In [23], an approach to manage multiple SDC factorizations in the SDRE is proposed, where the vector of weighting coefficients is found by minimising the objective function at each time step.

The remainder of the paper is organized as follows. In Section II the background notions are recalled and the motivations are given. The solution method employing the state transition matrix is presented, and implementation issues are discussed. In Section III, the controllability for state-dependent systems is treated, and the metric selected to measure the controllability is discussed. In Section IV, the modified ASRE (MASRE from now on) is formulated and implemented. Section V presents three application cases used to validate the MASRE concept. The achieved results are discussed and final remarks are given in Section VI.

II. Background

An optimal control problem can be stated in a variety of forms, which differ in terms of generality. The most general definition accommodates path constraints, variable final time, control saturation, interior-point constraints, etc. The problems treated in this paper are nonautonomous, nonlinear in the state, and affine (i.e., linear) in the control. The initial state is supposed given, and the final state can be either (in part) specified or unknown. The time span in which the problem is studied is fixed, and both the states and controls are unconstrained.

II.A. Statement of the problem

Given a set of n first-order differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x}, t)\,\mathbf{u}, \tag{1}$$

with $\mathbf{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^{n+1} \to \mathbb{R}^{n\times m}$, the m control functions $\mathbf{u}(t)$ must be determined within initial, final time t_i, t_f, such that the performance index

$$J = \varphi(\mathbf{z}(t_f), t_f) + \int_{t_i}^{t_f} L(\mathbf{x}, \mathbf{u}, t)\,\mathrm{d}t, \tag{2}$$

is minimized; $L : \mathbb{R}^{n+m+1} \to \mathbb{R}$, $\mathbf{z} = (x_{q+1}, \ldots, x_n)$ is the vector of state components *not* specified at final time ($0 \le q \le n$), and $\varphi : \mathbb{R}^{(n-q)+1} \to \mathbb{R}$. The initial condition is simply

$$\mathbf{x}(t_i) = \mathbf{x}_i, \tag{3}$$

whereas the final condition is allowed to take three different forms. These specify the hard constrained problem (HCP), the soft constrained problem (SCP), and the mixed constrained problem (MCP), with the final state fully specified, not specified, and partly specified, respectively. The final conditions for each problem, as well as the term φ in (2) are defined in Table 1.

Table 1. Definition of final state and costate conditions for the three problems (in MCP $j = 1, \ldots, q < n$ and $k = q+1, \ldots, n$).

Problem	φ	Final state	Final costate
HCP	not defined	$\mathbf{x}(t_f) = \mathbf{x}_f$	$\boldsymbol{\lambda}(t_f)$ free
SCP	$\varphi(\mathbf{x}(t_f), t_f)$	$\mathbf{x}(t_f)$ free	$\boldsymbol{\lambda}(t_f) = \partial\varphi/\partial\mathbf{x}$
MCP	$\varphi(x_k(t_f), t_f)$	$x_j(t_f) = x_{j,f}$	$\lambda_k(t_f) = \partial\varphi/\partial x_k$

Given the Hamiltonian $H(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, t) = L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T\left[\mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x}, t)\mathbf{u}\right]$, the problem

consists in finding a solution to the Euler–Lagrange equations,

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}}, \qquad \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}}, \qquad \frac{\partial H}{\partial \mathbf{u}} = 0, \tag{4}$$

where $\boldsymbol{\lambda}$ is the vector of costates. The differential-algebraic system (4), together with the initial conditions (3) and the final conditions (for both states and costates) in Table 1, defines a differential-algebraic two-point boundary value problem (TPBVP) whose solution provides the functions $\mathbf{x}(t)$, $\boldsymbol{\lambda}(t)$, $\mathbf{u}(t)$, $t \in [t_i,\, t_f]$.

II.B. The ASRE method

Suppose that $\mathbf{f}$ and $\mathbf{g}$ are at least continuous in an open set $\Omega \in \mathbb{R}^{n+1}$, and that $\mathbf{f}(\mathbf{0}, t) = \mathbf{0}$, $\forall t \in \mathbb{R}$. Under these conditions,[10] the dynamics (1) can be rewritten in the form

$$\dot{\mathbf{x}} = A(\mathbf{x}, t)\ \mathbf{x} + B(\mathbf{x}, t)\ \mathbf{u} \tag{5}$$

where where $A : \mathbb{R}^{n+1} \to \mathbb{R}^{n\times n}$ is found by mathematical factorization (it is not unique when $n > 1$) and $B : \mathbb{R}^{n+1} \to \mathbb{R}^{n\times m}$. Let also the objective function (2) be redefined in the quadratic-like form

$$J = \frac{1}{2}\mathbf{z}^T(t_f) S(\mathbf{z}(t_f), t_f)\mathbf{z}(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \left[\mathbf{x}^T Q(\mathbf{x}, t)\mathbf{x} + \mathbf{u}^T R(\mathbf{x}, \mathbf{u}, t)\mathbf{u}\right] \mathrm{d}t, \tag{6}$$

where $Q : \mathbb{R}^{n+1} \to \mathbb{R}^{n\times n}$ and $R : \mathbb{R}^{n+m+1} \to \mathbb{R}^{m\times m}$. In analogy with φ in Table 1, $\mathbf{z}$ and S are not defined in HCP, $\mathbf{z} = (x_{q+1}, \ldots, x_n)^T$ and $S : \mathbb{R}^{(n-q)\times(n-q)+1} \to \mathbb{R}^{(n-q)\times(n-q)}$ in MCP, and $\mathbf{z} = (x_1, \ldots, x_n)^T$ and $S : \mathbb{R}^{(n\times n)+1} \to \mathbb{R}^{n\times n}$ in SCP. The nonlinear dynamics (5) and the performance index (6), together with the initial condition (3) and the final conditions in Table 1, define an optimal control problem, which is equivalent to that set up by (1)–(2). Yet, this new problem is solved iteratively by reducing (5)–(6) to a series of TVLQR. These are defined by evaluating the SDC matrices A, B, Q, R, and S using the solution at the previous iteration. This method is known as the approximating sequence of Riccati equations (or ASRE) method.[18,19]

The initial step consists in solving *Problem 0*, defined by

$$\dot{\mathbf{x}}^{(0)} = A^{(0)}(t)\, \mathbf{x}^{(0)} + B^{(0)}(t)\, \mathbf{u}^{(0)}, \tag{7}$$

$$J = \frac{1}{2}(\mathbf{z}^{(0)}(t_f))^T S^{(0)}\, \mathbf{z}^{(0)}(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \left[(\mathbf{x}^{(0)})^T\, Q^{(0)}(t)\, \mathbf{x}^{(0)} + (\mathbf{u}^{(0)})^T\, R^{(0)}(t)\, \mathbf{u}^{(0)}\right] \mathrm{d}t, \tag{8}$$

where

$$A^{(0)}(t) = A(\mathbf{x}_i, t),\ B^{(0)}(t) = B(\mathbf{x}_i, t),\ Q^{(0)}(t) = Q(\mathbf{x}_i, t),\ R^{(0)}(t) = R(\mathbf{x}_i, \mathbf{0}, t),\ S^{(0)} = S(\mathbf{z}_i, t_f). \tag{9}$$

Problem 0 is a standard TVLQR as the arguments of the state-dependent matrices are all given except for the time (it is a LQR if (5) is not time-dependent). This problem is solved to yield $\mathbf{x}^{(0)}(t)$ and $\mathbf{u}^{(0)}(t)$, $t \in [t_i,\, t_f]$. At a generic, subsequent iteration, *Problem* k is defined by

$$\dot{\mathbf{x}}^{(k)} = A^{(k)}(t)\,\mathbf{x}^{(k)} + B^{(k)}(t)\,\mathbf{u}^{(k)}, \tag{10}$$

$$J = \frac{1}{2}(\mathbf{z}^{(k)}(t_f))^T S^{(k)} \mathbf{z}^{(k)}(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \left[(\mathbf{x}^{(k)})^T Q^{(k)}(t)\mathbf{x}^{(k)} + (\mathbf{u}^{(k)})^T R^{(k)}(t)\mathbf{u}^{(k)}\right] \mathrm{d}t, \tag{11}$$

where

$$\begin{gathered} A^{(k)}(t) = A(\mathbf{x}^{(k-1)}(t), t), \quad B^{(k)}(t) = B(\mathbf{x}^{(k-1)}(t), t), \quad Q^{(k)}(t) = Q(\mathbf{x}^{(k-1)}(t), t), \\ R^{(k)}(t) = R(\mathbf{x}^{(k-1)}(t), \mathbf{u}^{(k-1)}(t), t), \quad S^{(k)} = S(\mathbf{z}^{(k-1)}(t_f), t_f). \end{gathered} \tag{12}$$

This is inherently a TVLQR regardless of the time-dependence in (5) (note that $\mathbf{x}^{(k-1)}$ and $\mathbf{u}^{(k-1)}$ are the solutions of Problem $k-1$). Solving Problem k yields $\mathbf{x}^{(k)}(t)$ and $\mathbf{u}^{(k)}(t)$, $t \in [t_i,\, t_f]$. Iterations continue until a certain convergence condition is satisfied. In the present implementation of the algorithm, the convergence is reached when

$$\varepsilon = \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_\infty = \max_{t\in[t_i,\,t_f]} \{|x_j^{(k)}(t) - x_j^{(k-1)}(t)|, j = 1, \ldots, n\} \leq \mathrm{tol} \tag{13}$$

where 'tol' is a prescribed tolerance; i.e., iterations terminate when the difference between each component of the state, evaluated for all times, changes by less than 'tol' between two consecutive iterations. The sequence of solutions $\mathbf{x}^{(k)}$, $\mathbf{u}^{(k)}$ is proven to converge to the solution of the original problem (5)–(6) provided that $A(\mathbf{x}, t)$ and $B(\mathbf{x}, t)$ are Lipschitz continuous with respect to their arguments.[18]

II.C. Solution of the TVLQR by the state transition matrix

The sequence of TVLQR is solved by exploiting the structure of their Euler–Lagrange equations, so avoiding dealing with the matrix differential Riccati equation. This approach, in part, is described in [2], and differs from that implemented in [18, 19]. Suppose the following dynamics are given,

$$\dot{\mathbf{x}} = A(t)\,\mathbf{x} + B(t)\,\mathbf{u}, \tag{14}$$

together with the quadratic objective function

$$J = \frac{1}{2}(\mathbf{z}(t_f))^T \; S \; \mathbf{z}(t_f) + \frac{1}{2}\int_{t_i}^{t_f} \left[\mathbf{x}^T Q(t)\,\mathbf{x} + \mathbf{u}^T R(t)\,\mathbf{u}\right] \mathrm{d}t, \tag{15}$$

where Q, S and R are positive semi-definite and positive definite time-varying matrices, respectively. The necessary conditions (4) for this problem read

$$\begin{aligned} \dot{\mathbf{x}} &= A(t)\,\mathbf{x} + B(t)\,\mathbf{u}, & (16)\\ \dot{\boldsymbol{\lambda}} &= -Q(t)\,\mathbf{x} - A^T(t)\,\boldsymbol{\lambda}, & (17)\\ 0 &= R(t)\,\mathbf{u} + B^T(t)\,\boldsymbol{\lambda}. & (18) \end{aligned}$$

From equation (18) it is possible to get

$$\mathbf{u} = -R^{-1}(t)B^T(t)\,\boldsymbol{\lambda}, \tag{19}$$

which can be substituted into (16)–(17) to yield

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\lambda}} \end{pmatrix} = \begin{bmatrix} A(t) & -B(t)R^{-1}(t)B^T(t) \\ -Q(t) & -A^T(t) \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix}. \tag{20}$$

Since (20) is a system of linear differential equations, the solution can be written as

$$\begin{aligned} \mathbf{x}(t) &= \phi_{xx}(t_i, t)\,\mathbf{x}_i + \phi_{x\lambda}(t_i, t)\,\boldsymbol{\lambda}_i, & (21)\\ \boldsymbol{\lambda}(t) &= \phi_{\lambda x}(t_i, t)\,\mathbf{x}_i + \phi_{\lambda\lambda}(t_i, t)\,\boldsymbol{\lambda}_i, & (22) \end{aligned}$$

where $\mathbf{x}_i$, $\boldsymbol{\lambda}_i$ are the initial state, costate, respectively, and the functions ϕ_{xx}, $\phi_{x\lambda}$, $\phi_{\lambda x}$, and $\phi_{\lambda\lambda}$ are the components of the state transition matrix, which can be found by integrating the following dynamics

$$\begin{bmatrix} \dot{\phi}_{xx} & \dot{\phi}_{x\lambda} \\ \dot{\phi}_{\lambda x} & \dot{\phi}_{\lambda\lambda} \end{bmatrix} = \begin{bmatrix} A(t) & -B(t)R^{-1}(t)B^T(t) \\ -Q(t) & -A^T(t) \end{bmatrix} \begin{bmatrix} \phi_{xx} & \phi_{x\lambda} \\ \phi_{\lambda x} & \phi_{\lambda\lambda} \end{bmatrix}, \tag{23}$$

with initial conditions

$$\phi_{xx}(t_i, t_i) = \phi_{\lambda\lambda}(t_i, t_i) = I_{n\times n}, \quad \phi_{x\lambda}(t_i, t_i) = \phi_{\lambda x}(t_i, t_i) = 0_{n\times n}. \tag{24}$$

If both $\mathbf{x}_i$ and $\boldsymbol{\lambda}_i$ were given, it would be possible to compute $\mathbf{x}(t)$ and $\boldsymbol{\lambda}(t)$ through (21)–(22), and therefore the optimal control function $\mathbf{u}(t)$ with (19). As only $\mathbf{x}_i$ is given, the

issue is computing $\boldsymbol{\lambda}_i$ for the three problems defined above.

II.C.1. Hard Constrained Problem

In a HCP ($\mathbf{x}_f$ fully given, S not defined), the value of $\boldsymbol{\lambda}_i$ can be found by writing (21) at final time

$$\mathbf{x}_f = \phi_{xx}(t_i, t_f)\,\mathbf{x}_i + \phi_{x\lambda}(t_i, t_f)\,\boldsymbol{\lambda}_i, \tag{25}$$

and solving for $\boldsymbol{\lambda}_i$; i.e.,

$$\boldsymbol{\lambda}_i(\mathbf{x}_i, \mathbf{x}_f, t_i, t_f) = \phi_{x\lambda}^{-1}(t_i, t_f)\,[\mathbf{x}_f - \phi_{xx}(t_i, t_f)\,\mathbf{x}_i]\,. \tag{26}$$

II.C.2. Soft Constrained Problem

In a SCP ($\mathbf{x}_f$ not specified, S $n \times n$ positive definite matrix), the transversality condition $\boldsymbol{\lambda}(t_f) = \partial\varphi/\partial\mathbf{x}$ in Table 1 applied to (15) reads

$$\boldsymbol{\lambda}(t_f) = S\,\mathbf{x}(t_f), \tag{27}$$

which can be used to find $\boldsymbol{\lambda}_i$. This is done by writing (21)–(22) at final time and using (27),

$$\mathbf{x}(t_f) = \phi_{xx}(t_i, t_f)\,\mathbf{x}_i + \phi_{x\lambda}(t_i, t_f)\,\boldsymbol{\lambda}_i, \tag{28}$$

$$S\,\mathbf{x}(t_f) = \phi_{\lambda x}(t_i, t_f)\,\mathbf{x}_i + \phi_{\lambda\lambda}(t_i, t_f)\,\boldsymbol{\lambda}_i. \tag{29}$$

Equations (28)–(29) represent a linear algebraic system of $2n$ equations in the $2n$ unknowns $\{\mathbf{x}(t_f), \boldsymbol{\lambda}_i\}$. The system can be solved by substitution to yield

$$\boldsymbol{\lambda}_i(\mathbf{x}_i, t_i, t_f) = [\phi_{\lambda\lambda}(t_i, t_f) - S(t_f)\phi_{x\lambda}(t_i, t_f)]^{-1}\,[S(t_f)\phi_{xx}(t_i, t_f) - \phi_{\lambda x}(t_i, t_f)]\mathbf{x}_i. \tag{30}$$

II.C.3. Mixed Constrained Problem

Let the state be decomposed as $\mathbf{x} = (\mathbf{y}, \mathbf{z})$, where $\mathbf{y} = (x_1, \ldots, x_q)$ are the the q known components at final time, whereas $\mathbf{z} = (x_{q+1}, \ldots, x_n)$ are the remaining $n-q$ free components at final time. The costate is decomposed accordingly, i.e., $\boldsymbol{\lambda} = (\boldsymbol{\xi}, \boldsymbol{\eta})$, and S is $(n-q)\times(n-q)$. The MCP is solved by partitioning the state transition matrix in a suitable form such that,

at final time, equations (21)–(22) read

$$\begin{pmatrix} \mathbf{y}(t_f) \\ \mathbf{z}(t_f) \end{pmatrix} = \begin{bmatrix} \phi_{yy} & \phi_{yz} \\ \phi_{zy} & \phi_{zz} \end{bmatrix} \begin{pmatrix} \mathbf{y}_i \\ \mathbf{z}_i \end{pmatrix} + \begin{bmatrix} \phi_{y\xi} & \phi_{y\eta} \\ \phi_{z\xi} & \phi_{z\eta} \end{bmatrix} \begin{pmatrix} \boldsymbol{\xi}_i \\ \boldsymbol{\eta}_i \end{pmatrix}, \tag{31}$$

$$\begin{pmatrix} \boldsymbol{\xi}(t_f) \\ \boldsymbol{\eta}(t_f) \end{pmatrix} = \begin{bmatrix} \phi_{\xi y} & \phi_{\xi z} \\ \phi_{\eta y} & \phi_{\eta z} \end{bmatrix} \begin{pmatrix} \mathbf{y}_i \\ \mathbf{z}_i \end{pmatrix} + \begin{bmatrix} \phi_{\xi\xi} & \phi_{\xi\eta} \\ \phi_{\eta\xi} & \phi_{\eta\eta} \end{bmatrix} \begin{pmatrix} \boldsymbol{\xi}_i \\ \boldsymbol{\eta}_i \end{pmatrix}, \tag{32}$$

where the dependence on t_i, t_f is omitted for brevity. From the first row of (31) it is possible to get

$$\boldsymbol{\xi}_i = \phi_{y\xi}^{-1}[\mathbf{y}_f - \phi_{yy}\mathbf{y}_i - \phi_{yz}\mathbf{z}_i] - \phi_{y\xi}^{-1}\phi_{y\eta}\boldsymbol{\eta}_i, \tag{33}$$

which can be substituted in the second row of (31) to yield

$$\mathbf{z}(t_f) = [\phi_{zy} - \phi_{z\xi}\phi_{y\xi}^{-1}\phi_{yy}]\mathbf{y}_i + [\phi_{zz} - \phi_{z\xi}\phi_{y\xi}^{-1}\phi_{yz}]\mathbf{z}_i + \phi_{z\xi}\phi_{y\xi}^{-1}\mathbf{y}_f + [\phi_{z\eta} - \phi_{z\xi}\phi_{y\xi}^{-1}\phi_{y\eta}]\boldsymbol{\eta}_i. \tag{34}$$

Equations (33)–(34), together with the transversality condition $\boldsymbol{\eta}(t_f) = S(t_f)\mathbf{z}(t_f)$ (see Table 1), can be substituted in the second row of (32) to compute $\boldsymbol{\eta}_i$; i.e.,

$$\boldsymbol{\eta}_i(\mathbf{x}_i, \mathbf{y}_f, t_i, t_f) = [\tilde{\phi}_{\eta\eta}]^{-1}\mathbf{w}(\mathbf{x}_i, \mathbf{y}_f, t_i, t_f), \tag{35}$$

where $\tilde{\phi}_{\eta\eta} = \phi_{\eta\eta} - \phi_{\eta\xi}\phi_{y\xi}^{-1}\phi_{y\eta} - S(\phi_{z\eta} - \phi_{z\xi}\phi_{y\xi}^{-1}\phi_{y\eta})$ and

$$\begin{aligned} \mathbf{w}(\mathbf{x}_i, \mathbf{y}_f, t_i, t_f) = {} & [S(\phi_{zy} - \phi_{z\xi}\phi_{y\xi}^{-1}\phi_{yy}) - \phi_{\eta y} + \phi_{\eta\xi}\phi_{y\xi}^{-1}\phi_{yy}]\mathbf{y}_i \\ & + [S(\phi_{zz} - \phi_{z\xi}\phi_{y\xi}^{-1}\phi_{yz}) + \phi_{\eta z} + \phi_{\eta\xi}\phi_{y\xi}^{-1}\phi_{yz}]\mathbf{z}_i + [S(\phi_{z\xi}\phi_{y\xi}^{-1}) - \phi_{\eta\xi}\phi_{y\xi}^{-1}]\mathbf{y}_f. \end{aligned} \tag{36}$$

Once $\boldsymbol{\eta}_i$ is known, the remaining part of the initial costate, $\boldsymbol{\xi}_i$, is computed through (33), and therefore the full initial costate is obtained as a function of the initial condition, given final condition, initial, and final time; i.e, $\boldsymbol{\lambda}_i(\mathbf{x}_i, \mathbf{y}_f, t_i, t_f) = (\boldsymbol{\xi}_i(\mathbf{x}_i, \mathbf{y}_f, t_i, t_f), \boldsymbol{\eta}_i(\mathbf{x}_i, \mathbf{y}_f, t_i, t_f))$.

II.D. Implementation of the iterative scheme

The ASRE method illustrated in Section II.B has been coded into a numerical framework by implementing the solution scheme shown in Section II.C. The overall flowchart of the algorithm is shown in Figure 1. The original, fully nonlinear problem (1)–(2) is factorized into the state-dependent form (5)–(6), and then solved iteratively with the three-step procedure in the 'Solver' box. At each iteration the convergence condition (13) is checked, and either another iteration is performed or the solution is found. As the integration of (23) requires evaluating the SDC matrices for different times, an uniform time grid of N points is constructed, $t_i = t_1, t_2, \ldots, t_N = t_f$, and a cubic spline interpolation is used to evaluate

these matrices within the $N-1$ time intervals. The solution is then available only on this grid, i.e., $\mathbf{x}_j = \mathbf{x}(t_j)$, $\boldsymbol{\lambda}_j = \boldsymbol{\lambda}(t_j)$, $\mathbf{u}_j = \mathbf{u}(t_j)$, $j = 1, \ldots, N$, and a cubic spline may be again used if the solution over an intermediate time has to be retrieved.

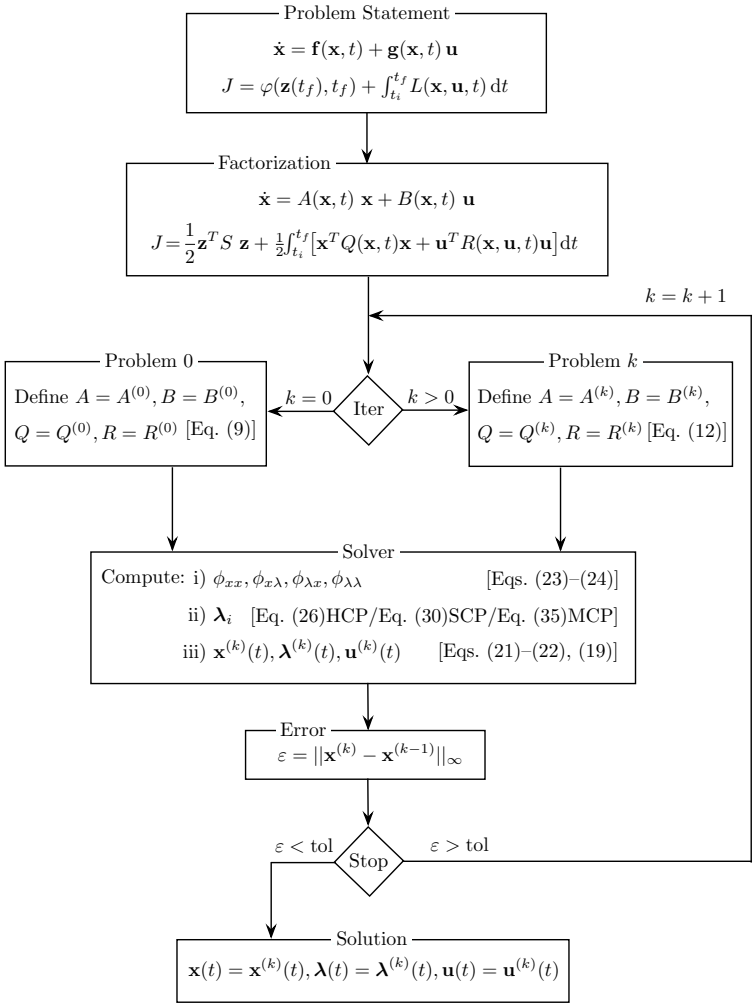


Figure 1. Flowchart for the implementation of the ASRE scheme.

II.E. Dependence on Factorization

The ASRE algorithm, as coded in Figure 1, has been used to solve a number of nonlinear optimal control problems.[20–22] This algorithm requires factorizing the nonlinear dynamics with a state-dependent matrix. Finding a good factorization can be non-trivial, mostly because, for systems with at least two states, there exists an infinite number of factorizations. Indeed, given a dynamical system

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2), \\ \dot{x}_2 &= f_2(x_1, x_2), \end{aligned} \tag{37}$$

it is possible, for instance, to use the SDC matrix $A_1(\mathbf{x})$,

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \underbrace{\begin{bmatrix} f_1(x_1,x_2)/x_1 & 0 \\ 0 & f_2(x_1,x_2)/x_2 \end{bmatrix}}_{A_1(\mathbf{x})} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \tag{38}$$

or $A_2(\mathbf{x})$,

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \underbrace{\begin{bmatrix} 0 & f_1(x_1,x_2)/x_2 \\ f_2(x_1,x_2)/x_1 & 0 \end{bmatrix}}_{A_2(\mathbf{x})} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \tag{39}$$

Not only, as there exists a *family* of linear combinations,

$$A_3(\alpha,\mathbf{x}) = (1-\alpha)\,A_1(\mathbf{x}) + \alpha A_2(\mathbf{x}), \quad \forall \alpha \in \mathbb{R}, \tag{40}$$

which produces infinite factorizations of (37). This procedure can be applied to any system with $n > 1$ states. One could think that the factorization choice does not affect the final solution, but actually this is not the general case, as shown by the following, simple example.

II.E.1. Example 1

The problem consists in minimizing

$$J = \frac{1}{2}\int_{t_i}^{t_f} \left(x_1^2 + x_2^2 + u^2\right) \mathrm{d}t, \tag{41}$$

under the dynamics (taken from [24])

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= x_1 x_2 + u, \end{aligned} \tag{42}$$

with boundary conditions $\mathbf{x}(t_i) = (2,1)^T$, $\mathbf{x}(t_f) = (4,1)^T$, and $t_i = 0$, $t_f = 5$.

Two possible *parent* factorizations of the nonlinear, uncontrolled dynamics (42) are

$$A_1(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ 0 & x_1 \end{bmatrix}, \qquad A_2(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ x_2 & 0 \end{bmatrix}. \tag{43}$$

The remaining SDC matrices for this problem are $B = [0,\ 1]^T$ (in (5)) and $Q = I_{2\times 2}$, $R = 1$ (in (6)); S in (6) is not defined, since Example 1 is an HCP. This problem has been solved with the ASRE scheme by using A_1, A_2 in (43) and also A_3 in (40) with $\alpha = 0.5$.

The three *different* solutions are shown in Figure 2, and their features (cost and iterations) are summarized in Table 2. The qualitative and, most importantly, quantitative difference among these solutions makes it clear that the SDC factorization of the uncontrolled dynamics plays a crucial role in the present, as well as any other, state-dependent methods.

Table 2. Features of the three solutions to Example 1 ($\text{tol} = 10^{-6}$).

$A(\mathbf{x})$	J	Iter
A_1	52.24	12
A_2	64.91	56
A_3	53.13	21

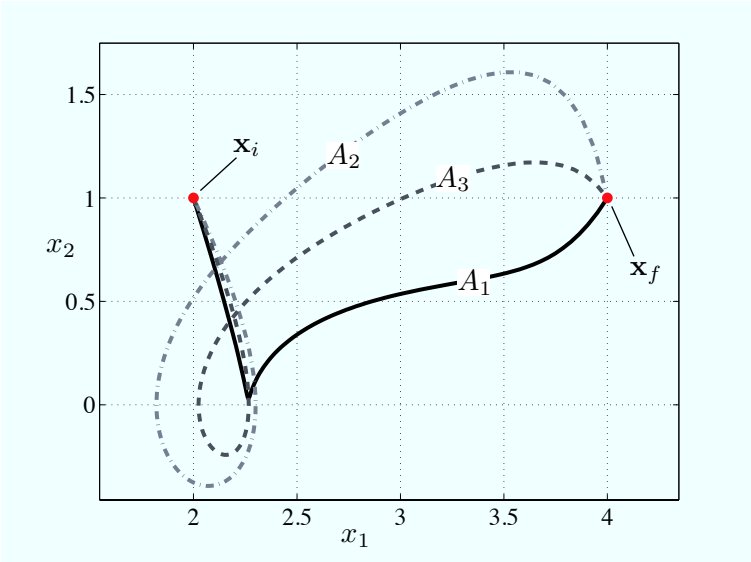


Figure 2. Trajectories of the three solutions to Example 1.

III. Controllability of State-Dependent Dynamics

A property that ASRE requires is *controllability* of the pair $\{A(\mathbf{x},t), B(\mathbf{x},t)\}$ in (5). A system is controllable if and only if there always exists an admissible control that is able to transfer the state from a condition to another in a finite time.[25] When dealing with state-dependent dynamics, the SDC matrices must be selected such that the state-dependent system is controllable for all the possible trajectories. In [26], necessary and sufficient conditions for the existence of SDC matrices are shown. It would be desirable to choose the factorization that grants the maximum controllability of the state-dependent dynamics, since the solution performance is strictly related to the controllability of the system. An increased controllability leads to a reduction of the control effort and likely to a reduced cost.[27] In this way it could be possible to identify the optimal factorization.[10]

However, the controllability of (5) does not lead to the controllability of (1), as there is no trivial relation between the controllability of $\{A(\mathbf{x},t), B(\mathbf{x},t)\}$ and that of $\{\mathbf{f}(\mathbf{x},t), \mathbf{g}(\mathbf{x},t)\}$.[24] The controllability of the state-dependent dynamics is increased with the purpose of easing the convergence of the ASRE method.

A linear system is controllable when the controllability matrix, K_c, is full-rank; i.e.,

$$\operatorname{rank}(K_c) = \operatorname{rank}\begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} = n. \tag{44}$$

Since A, B in (5) are in general state- and time-dependent, K_c is state- and time-dependent as well, and thus the controllability check (44) must be done for all states and times. This is deemed not viable. Another way to study the controllability of state-dependent systems is based on evaluating their distance from being uncontrollable, which can be done by estimating the amplitude of the controllability radius,[28,29]

$$\rho(A,B) = \min_{\lambda\in\mathbb{C}} \sigma_{\min}\left(\begin{bmatrix} A-\lambda I & B \end{bmatrix}\right), \tag{45}$$

where $\sigma_{\min}(\cdot)$ returns the smallest singular value of its matrix argument.

In this paper, the notion of controllability Gramian is used.[25] This is defined as

$$P(t_i,t_f) = \int_{t_i}^{t_f} \Phi(t_i,\tau)B(\tau)B^T(\tau)\Phi^T(t_i,\tau)\mathrm{d}\tau, \tag{46}$$

where $\Phi(t_i,\tau)$ is the solution of

$$\dot{\Phi}(t,\tau) = A(t)\Phi(t,\tau), \qquad \Phi(t,t) = I, \tag{47}$$

with $A(t)$ as in (14). A system is controllable if $P(t_i,t_f)$ is nonsingular; i.e., if $\det P(t_i,t_f) \neq 0$. Notice that this condition for controllability is seeking controllability along the entire trajectory of the system, and is not a combination of pointwise controllability for a given set of values of the states. This is, however, a binary condition: it can be used to check whether a system is controllable or not, but not to *measure* its controllability. As the higher controllability provides the less control effort, the focus is to choose the factorization that maximizes the controllability, and therefore a metric is introduced by defining a figure of merit.

In [10,30,31], the SDC are chosen by maximizing the absolute value of the determinant of $K_cK_c^T$ (note that K_c is not necessarily square. When $P(t_i,t_f)$ is considered, it could seem reasonable to adopt the absolute value of its determinant as a merit figure of controllability. Actually, such a choice would be misleading since it is easy to prove that a nearly singular matrix can show a determinant much greater than zero.[28] Alternatively, the minimum

eigenvalue of P may be used to measure the controllability: a matrix is close to singularity when its smallest eigenvalue is close to zero, regardless of the determinant value. However, the eigenvalues suffer of ill-conditioning as small perturbations of a matrix can lead to big variations of its eigenvalues.[28] To overcome these issues, in the remainder, the Gramian singular values have been used. These are positive real numbers that can be used in place of eigenvalues. More specifically, $\sigma_{\min}(P)$ is considered to measure the system controllability.

In Figure 3, the trend of $\sigma_{\min}(P)$ for the solutions to Example 1 is shown, at each iteration, for the three SDC matrices considered. It can be noticed that the case requiring the lowest number of iterations corresponds to A_1, which presents also the highest $\sigma_{\min}(P)$. From the other two cases (A_2, A_3) it can be inferred that the lower $\sigma_{\min}(P)$, the higher number of iterations, and vice versa. Not only, as from Table 2 it can be seen that the highest $\sigma_{\min}(P)$ involves the best (i.e., lowest) value of cost function.

In summary, different SDC representations lead to different behaviors of the ASRE. The factorizations resulting the most controllable are also those entailing the lowest cost functions and numbers of iterations. Given a nonlinear dynamics, the selection of the state-dependent matrix $A(\mathbf{x})$ is a degree of freedom that can be exploited to enforce the best performances in terms of algorithm convergence and control effort.

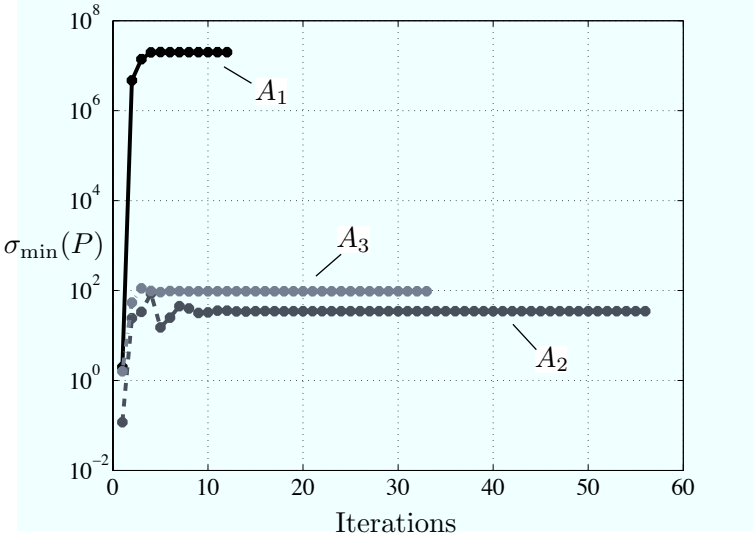


Figure 3. Values of $\sigma_{\min}(P)$ during the iterations for the three factorization cases of Example 1.

IV. A Combined Method

It has been shown that the SDC of the uncontrolled dynamics affect the algorithm convergence, the number of iterations, the solution trajectory, and the objective function value. In principle, once a nonlinear problem is given, it would be possible to find a number of

different SDC matrices and to run the algorithm in Figure 1 several times. Then, the best solution in terms of cost function could be extracted among those feasible. This means using the ASRE algorithm as a black-box tool, not relying on any notion of controllability.

In this paper the focus is instead on handling all the possible factorizations at once, and to optimally combine them to increase the controllability of the resulting system. Given a set of *parent* SDC matrices, the spirit is to treat them in the same fashion as A_1, A_2 are combined to define A_3 in (40). The aims are: 1) to increase the controllability, 2) to ease the algorithm convergence, 3) to improve the solution cost. This is done by maximizing $\sigma_{\min}(P)$ at each iteration, which involves solving a nonlinear programming problem. If needed, the approximate solution so found is used to tackle the TPBVP defined by the original dynamics. It is shown that the approximate solution so found is a good initial guess, and it is only refined to satisfy the Euler–Lagrange equations (4).

IV.A. Managing multiple factorizations

Given a set of n_f parent factorizations of $\mathbf{f}(\mathbf{x},t)$ in (1) having the form $\mathbf{f}(\mathbf{x},t) = A_i(\mathbf{x},t)\mathbf{x}$, $i = 1,\dots,n_f$, the combination that grants the maximum controllability is looked for. The *family* of SDC matrices (generated by the parents A_i) can be written as[32]

$$A(\boldsymbol{\alpha},\mathbf{x},t) = \left[\prod_{j=1}^{n_f-1}(1-\alpha_j)\right] A_1(\mathbf{x},t) + \sum_{i=2}^{n_f-1}\alpha_{i-1}\left[\prod_{j=i}^{n_f-1}(1-\alpha_j)\right] A_i(\mathbf{x},t) + \alpha_{n_f-1}A_{n_f}(\mathbf{x},t), \tag{48}$$

where $\boldsymbol{\alpha} = \left(\alpha_1,\dots,\alpha_{n_f-1}\right)^T$ is a vector of weighting coefficients. The family (48) produces the state-dependent form (5), i.e.,

$$\dot{\mathbf{x}} = A\left(\boldsymbol{\alpha},\mathbf{x},t\right)\mathbf{x} + B(\mathbf{x},t)\mathbf{u}, \tag{49}$$

if the following inequalities are satisfied,

$$\|\boldsymbol{\alpha}\|_1 \leq 1, \quad \text{and} \quad \alpha_j \geq 0,\ j = 1,\dots,n_f-1. \tag{50}$$

Conditions (50) define the region of existence for $\boldsymbol{\alpha}$, which is required in order to ensure a convex domain[32] (see Figure 4). More precisely, the restrictions in (50) guarantee a one-to-one correspondence between $\boldsymbol{\alpha}$ and $A(\boldsymbol{\alpha},\mathbf{x},t)$ in (48). As $\boldsymbol{\alpha}$ is a vector of bounded coefficients, $A(\boldsymbol{\alpha},\mathbf{x},t)$ is Lipschitz continuous when so are the parent factorizations $A_i(\mathbf{x},t)$. Other relations can be formulated in place of (48). In [23], a linear combination is used to define a family of factorizations in the SDRE method; in [10,30,31], such combination is done at scalar level by treating each nonlinear element of the original dynamics independently from the others. Equation (48) is preferred as it is more general. It allows considering

families of nonlinearities in a unified way. This is done to maintain some physical insight into the problem. Indeed, in typical problems dynamics and nonlinearities are related to vector quantities that in the proposed approach would be treated as vectors and not as independent scalar variables.

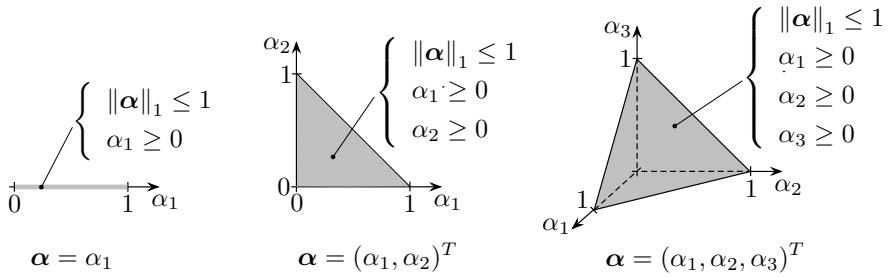


Figure 4. Region of existence for $\boldsymbol{\alpha}$ in one, two, and three dimensions ($n_f = 2, 3, 4$, respectively).

IV.B. Optimal combination of state-dependent factorizations

The expression (48) allows us to handle a set of SDC matrices simultaneously. This is used to modify the ASRE method to accommodate for n_f factorizations. However, the iterative method requires the dynamics to be written in the form (5), which can be achieved by specifying the value of $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n_f})^T$ in (49). It is important to recall that the parent factorizations A_i have been introduced for the purpose of increasing the system controllability. Thus, the weights α_j have to be chosen such that the minimum singular value of the Gramian associated to (49) is maximized. This is done by solving, at each iteration, the following Nonlinear Programming (NLP) problem

$$\max_{\boldsymbol{\alpha}} \sigma_{\min}\left(P(\boldsymbol{\alpha})\right) \quad \text{subject to} \quad \begin{cases} \|\boldsymbol{\alpha}\|_1 \leq 1 \\ \alpha_j \geq 0, \ j = 1, \ldots, n_f - 1 \end{cases} \tag{51}$$

where $P(\boldsymbol{\alpha})$ is the Gramian obtained by plugging (48) into (46)–(47). The degrees of freedom α_j are thus exploited optimally to improve the existing algorithm. This is the ultimate goal of the MASRE. The price to pay is the solution of the algebraic problem (51) at each iteration. This holds as the state-dependent matrices A, B in (46)–(47) are evaluated with the solution at the previous step. Indeed, in analogy with Eqs. (9) and (12), the SDC matrices are

$$A^{(0)}(\boldsymbol{\alpha}, t) = A(\boldsymbol{\alpha}, \mathbf{x}_i, t), \quad \text{in Problem 0;} \tag{52}$$

$$A^{(k)}(\boldsymbol{\alpha}, t) = A(\boldsymbol{\alpha}, \mathbf{x}^{(k-1)}, t), \quad \text{in Problem k.} \tag{53}$$

IV.C. Implementation of the modified scheme

The ASRE scheme in Figure 1 has been modified to manage multiple factorizations and to choose their optimal combination that yields the best controllability. Figure 5 reports the flow chart of the MASRE algorithm. Once the problem is stated, a number of parent SDC matrices for the uncontrolled dynamics are defined, which are then combined to generate the whole family through (48). The general problem is then defined by (49) and (6). This is treated in the same fashion as in Section II except that a value for $\boldsymbol{\alpha}$ is needed this time. In order to obtain the optimal value of $\boldsymbol{\alpha}$, the NLP is solved in the 'Optimization' box. The first guess value for $\boldsymbol{\alpha}$ is the value found at the previous iteration (an initial guess $\boldsymbol{\alpha}^{(0)}$ is provided to run the NLP the first time). The approximate solution $(\mathbf{x}(t), \boldsymbol{\lambda}(t), \mathbf{u}(t))$ is found once the convergence condition is verified. This condition is the same as in the ASRE, where a single SDC matrix is used.[18] In the MASRE, where the SDC matrix may change at each iteration, the conditions for the convergence of the sequence of Riccati equations may be violated. However, numerical experiments suggest that algorithm convergence is achieved once $\boldsymbol{\alpha}$ in (48) stabilizes about an optimal value, such that the sequence of Riccati equations approaches convergence with a fixed SDC matrix (see Section V).

IV.D. Refinement of the approximate solution

The approximate solution is expressed through functions evaluated at discrete locations of a time grid made up of N points (see Section II.D). If needed, this solution can be refined by solving the Euler–Lagrange equations (4). In particular, the approximate solutions can be used as a first guess solutions for a TPBVP solver that deals with Eq. (4) and the boundary conditions as in (3) and Table 1 (see Figure 5). This step is particularly easy when the original problem (2) possesses a quadratic-control objective function. This condition, together with the control affinity in (1), makes it possible to yield the control as a function of the Lagrange multiplier from the third of (4). This makes the Euler–Lagrange equations a purely differential TPBVP. In the present implementation of the method, this step is accomplished through a symbolic manipulation of (1)–(2). (If a TPBVP solver implementing a collocation method is used, the approximate solution discretized over the time grid is already put in the proper form).

In next section it is shown through examples that the solution obtained with the MASRE represents a good approximation of the optimal solution. The latter is labelled $\mathbf{x}^{(\mathrm{opt})}(t)$, $\boldsymbol{\lambda}^{(\mathrm{opt})}(t)$, $\mathbf{u}^{(\mathrm{opt})}(t)$, and the associated cost index $J^{(\mathrm{opt})}$. It is not straightforward to assert whether $J^{(\mathrm{opt})}$ is better or worse than the state-dependent objective function J, as $\mathbf{x}^{(\mathrm{opt})}(t)$ verifies a "different" dynamics than those imposed by the MASRE algorithm.

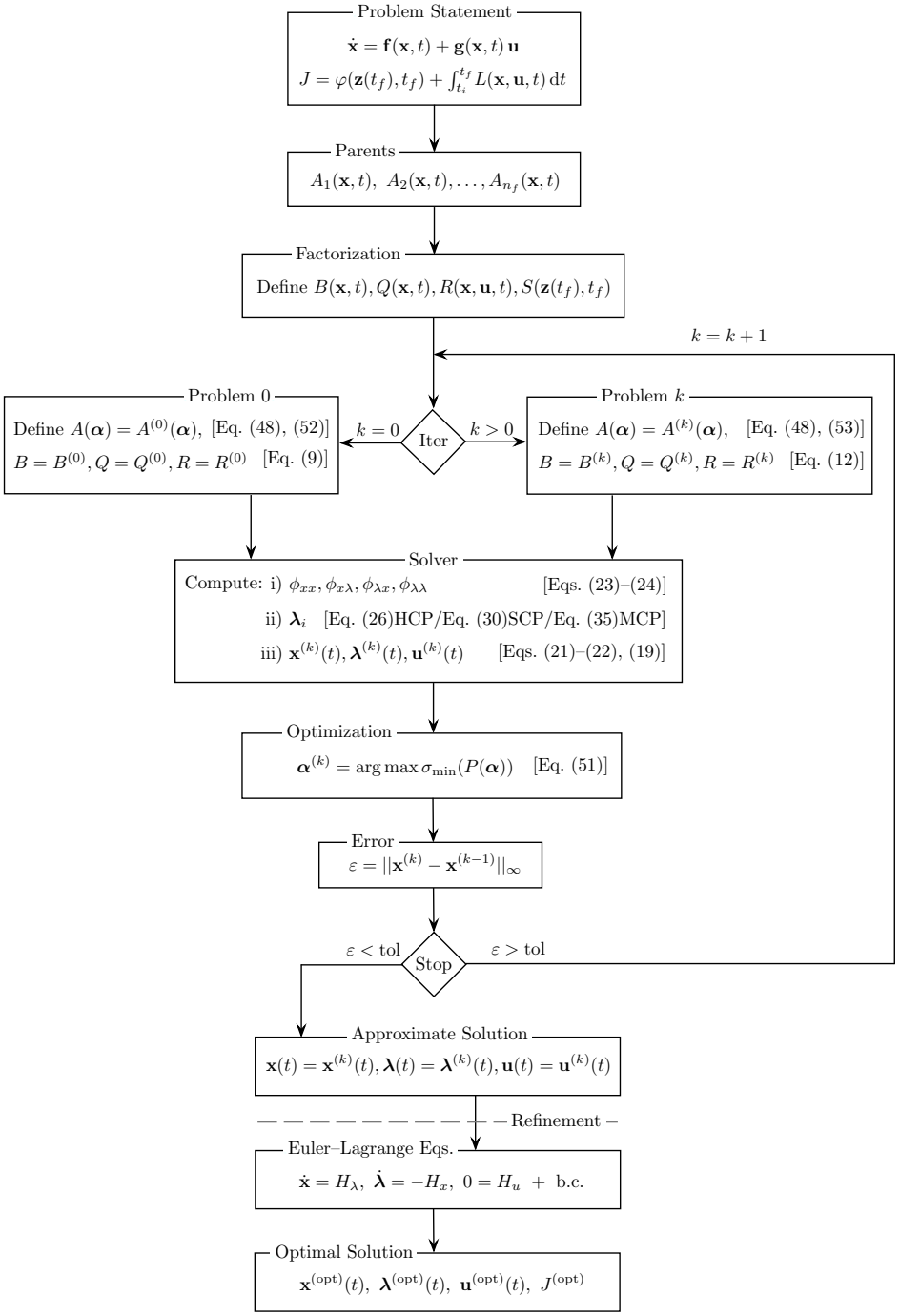


Figure 5. Flowchart for the implementation of the MASRE scheme.

V. Application cases

In this section the MASRE scheme presented in Section IV is applied to three sample problems. The properties of the solution method are discussed. It is shown that the approximate solutions reproduce well the optimal solution. The latter is found by solving the Euler–Lagrange TPBVP using the approximate solutions as first guess (see Figure 5).

V.A. Problem 1

The problem set up is that of Section II.E.1. Beside A_1 and A_2 in (43), two additional SDC matrices have been provided,

$$A_3(\mathbf{x}) = \begin{bmatrix} x_2 & 1-x_1 \\ 0 & x_1 \end{bmatrix}, \qquad A_4(\mathbf{x}) = \begin{bmatrix} x_2 & 1-x_1 \\ x_2 & 0 \end{bmatrix}. \tag{54}$$

It is worth mentioning that the controllability matrices associated to the four factorizations are

$$K_{c_1}(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ 1 & x_1 \end{bmatrix}, \quad K_{c_2}(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad K_{c_3}(\mathbf{x}) = \begin{bmatrix} 0 & 1-x_1 \\ 1 & x_1 \end{bmatrix}, \quad K_{c_4}(\mathbf{x}) = \begin{bmatrix} 0 & 1-x_1 \\ 1 & 0 \end{bmatrix},$$

and it is clear that $\text{rank}(K_{c_1}(\mathbf{x})) = \text{rank}(K_{c_2}(\mathbf{x})) = 2\ \forall \mathbf{x}$, while both $K_{c_3}(\mathbf{x})$ and $K_{c_4}(\mathbf{x})$ become singular at $x_1 = 1$. It is interesting to notice that the factorizations that are not always controllable, A_3 and A_4, are not "natural" representations of the uncontrolled dynamics, as their derivation relies on some sort of addition and subtraction of the same term.

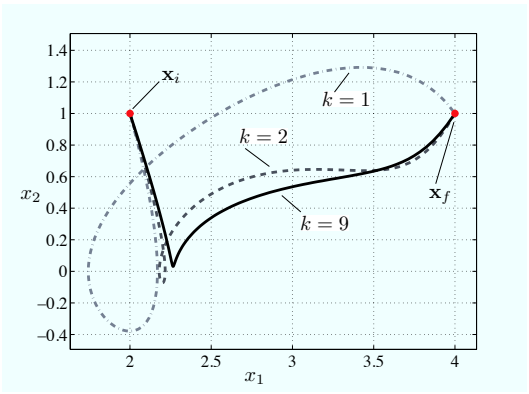
The MASRE algorithm has been initiated with A_1, A_2 in (43), A_3, A_4 in (54), and with B, Q, R as in Section II.E.1. The iterations are summarized in Table 3 where the error, the Gramian minimum singular value, and the objective function are reported for each iteration. The last column in Table 3 reports the parent factorization which the family converges to (through the weighting vector $\boldsymbol{\alpha}$, whose first guess is $\boldsymbol{\alpha}^{(0)} = (1-\sqrt{2}/2, 1-\sqrt{2}/2, 1-\sqrt{2}/2)^T$); the termination tolerance is set to 10^{-6}. Nine iterations are needed to converge to an approximate solution. The algorithm is able to detect the non-controllability associated to A_3, A_4 and to discard them. The optimal matrix turns out to be A_1, and this, remembering the results obtained in Section II.E.1, confirms that the algorithm is able to automatically detect the best factorization in terms of controllability and thus cost function.

Figures 6(a) and 6(b) show the first, second, and last iterations in terms of solution trajectory (x_1, x_2) and control profile $u(t)$. The algorithm is able to converge to the final approximate solution ($k = 9$) although the first iteration lies quite far from it. The approximate solution is then refined by enforcing the Euler–Lagrange TPBVP, and the results are

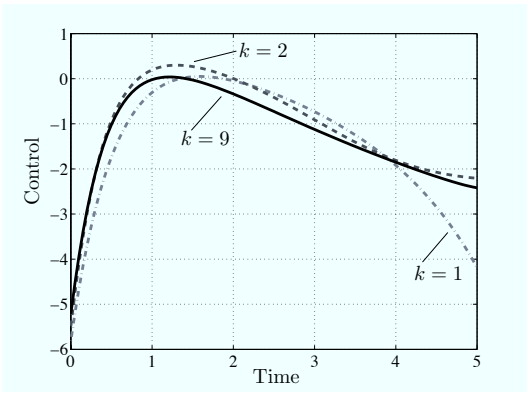
Table 3. MASRE iterations for Problem 1, CPU time 39.9 s (Intel Core2 Quad CPU 2.50 GHz).

k	Error	$\sigma_{\min}(P)$	J	Parent
1	2.000	2.082	2.436×10^{1}	A_1
2	7.173×10^{-1}	1.578×10^{6}	2.505×10^{1}	A_1
3	1.412×10^{-1}	1.130×10^{7}	2.606×10^{1}	A_1
4	1.294×10^{-2}	1.960×10^{7}	2.613×10^{1}	A_1
5	1.648×10^{-3}	2.009×10^{7}	2.612×10^{1}	A_1
6	1.767×10^{-4}	1.997×10^{7}	2.612×10^{1}	A_1
7	1.893×10^{-5}	1.996×10^{7}	2.612×10^{1}	A_1
8	2.259×10^{-6}	1.996×10^{7}	2.612×10^{1}	A_1
9	2.083×10^{-7}	1.996×10^{7}	2.612×10^{1}	A_1

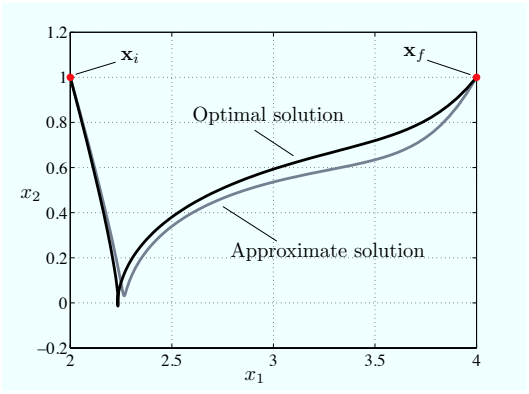
shown in Figures 6(c) and 6(d). From this step it can be inferred that the developed scheme approximates well the final, optimal solution. The cost of the latter is $J^{(\text{opt})} = 2.606 \times 10^{1}$, which means a 0.2% difference with respect to J in Table 3, $k = 9$. This refinement makes it clear that the approximate solution is not optimal, but rather suboptimal.



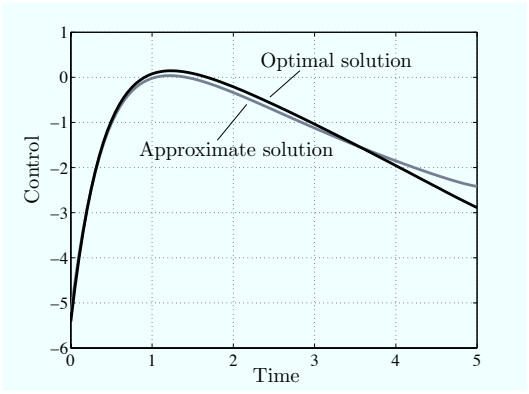
(a) Phase space iterations.



(b) Control profile iterations.



(c) Approximate and optimal orbit.



(d) Approximate and optimal control.

Figure 6. Problem 1: iterated, approximate, and optimal solution.

V.B. Problem 2

This problem consists in minimizing the same performance index of Problem 1 (Eq. (41); i.e., B, Q, R as in Section II.E.1) subject to

$$\begin{aligned} \dot{x}_1 &= x_1^2 x_2 + x_2, \\ \dot{x}_2 &= x_1 x_2 + u, \end{aligned} \tag{55}$$

with the same boundary conditions and initial, final time. The algorithm has been fed with four different SDC matrices of the uncontrolled dynamics

$$A_1(\mathbf{x}) = \begin{bmatrix} x_1 x_2 & 1 \\ 0 & x_1 \end{bmatrix}, \; A_2(\mathbf{x}) = \begin{bmatrix} 0 & x_1^2 + 1 \\ 0 & x_1 \end{bmatrix}, \; A_3(\mathbf{x}) = \begin{bmatrix} x_1 x_2 & 1 \\ x_2 & 0 \end{bmatrix}, \; A_4(\mathbf{x}) = \begin{bmatrix} 0 & x_1^2 + 1 \\ x_2 & 0 \end{bmatrix}, \tag{56}$$

all of which are always controllable; i.e., $\det(K_{c_i}) \neq 0$, $i = 1, \ldots, 4$. Differently from the previous problem, where two factorizations were evidently not "natural" representations of the uncontrolled dynamics, this problem is expressed naturally with four factorizations, so that one would not know a priori which SDC matrix should be used to approximate the optimal solution with the ASRE algorithm. MASRE is initiated with the same settings of Problem 1.

The summary of iterations is reported in Table 4, where not all the steps are reported for brevity sake (the algorithm converges in 21 iterations). It can be seen that the algorithm automatically converges toward the parent A_2 factorization, but uses A_4 in the first two iterations. In particular, $\sigma_{\min}(P)$ is on the order of 10^7 in the first iteration, but then it drops to 10^3 in the second; here the algorithm is able to detect a better controllability associated to A_2, and therefore it adjusts $\boldsymbol{\alpha}$ to underweight A_4 in favor of A_2. As before, the first, second, and last iterations in terms of solution trajectory and control profile are shown in Figures 7(a) and 7(b). In order to assess the quality of this solution, the problem has been solved with the original algorithm, employing the four different factorizations separately, and it has been confirmed that the the best solution is the one related to A_2. Moreover, the original ASRE solver is not able to provide a solution using A_1 or A_3: the former yields a sequence of iterations whose error stably oscillates without converging to any solution, while the latter stops as the transition matrix becomes ill-conditioned after a number of iterations. This implies that controllability is at most a necessary conditions for a converging factorization, but the above discussion also stresses the fact that optimizing the controllability allows us to dodge those factorizations that would not lead to any solution.

The approximate solution has been refined by solving the TPBVP associated to (55) and (41). The results are shown in Figure 7(c) and 7(d). Again, this step makes it clear that

the approximate solution represents well the final, optimal solution. The objective function associated to the latter is $J^{(\mathrm{opt})} = 9.413$, which is 2.9% less than that in Table 4, $k = 21$, so showing once again that the approximate solution is not optimal but only sub-optimal.

Table 4. MASRE iterations for Problem 2, CPU time 231.9 s.

k	Error	$\sigma_{\min}(P)$	J	Parent
1	2.000	1.739×10^{7}	14.442	A_4
2	1.376	2.978×10^{3}	7.876	A_4
3	1.121	1.355×10^{4}	10.858	A_2
4	7.010×10^{-1}	1.615×10^{4}	9.327	A_2
5	2.423×10^{-1}	1.313×10^{4}	9.798	A_2
6	9.391×10^{-2}	1.480×10^{4}	9.767	A_2
7	6.236×10^{-2}	1.495×10^{4}	9.643	A_2
8	2.657×10^{-2}	1.434×10^{4}	9.708	A_2
9	9.479×10^{-3}	1.443×10^{4}	9.710	A_2
10	6.504×10^{-3}	1.452×10^{4}	9.696	A_2
11	2.838×10^{-3}	1.449×10^{4}	9.701	A_2
21	9.011×10^{-7}	1.449×10^{4}	9.701	A_2

V.C. Problem 3

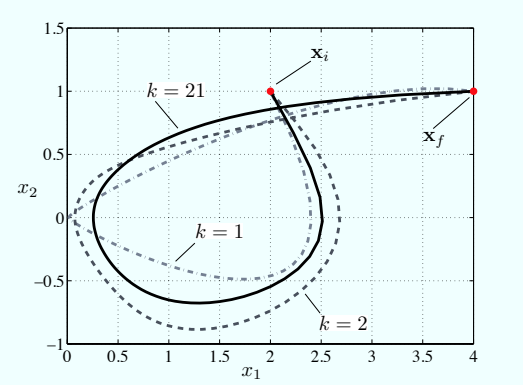
In this problem the controlled, planar Keplerian motion of a spacecraft in polar coordinates is studied. The dynamics are

$$\begin{aligned} \dot{x}_1 &= x_3, \\ \dot{x}_2 &= x_4, \\ \dot{x}_3 &= x_1 x_4^2 - 1/x_1^2 + u_1, \\ \dot{x}_4 &= -2x_3 x_4/x_1 + u_2/x_1, \end{aligned} \tag{57}$$

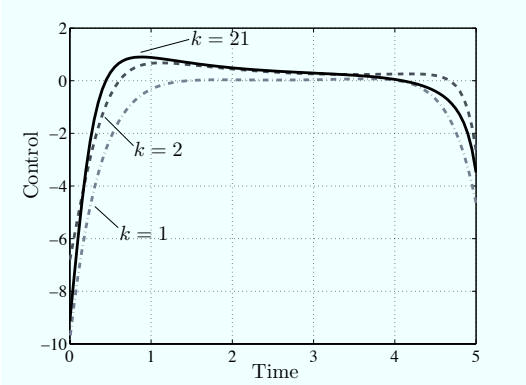
where x_1 is the radial distance from the attractor, x_2 the phase angle, x_3 the radial velocity, and x_4 the transversal velocity; u_1, u_2 are the radial and transversal components of the control acceleration (Eqs. (57) are written in canonical units). The objective function is

$$J = \frac{1}{2} \int_{t_i}^{t_f} (u_1^2 + u_2^2)\, \mathrm{d}t. \tag{58}$$

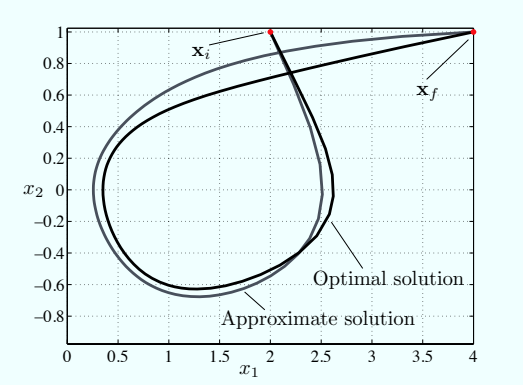
The initial state is $\mathbf{x}_i = (1, 0, 0, 1)$, and $t_i = 0$, $t_f = \pi$. Two different problems are solved, according to the definition of $\mathbf{x}_f$. In the HCP, $\mathbf{x}_f = (1.52, \pi, 0, 1.52^{-3/2})$, while in the MCP $x_{2,f}$ is unspecified; i.e., the final phase angle is left free. (This set up mimics an Earth–



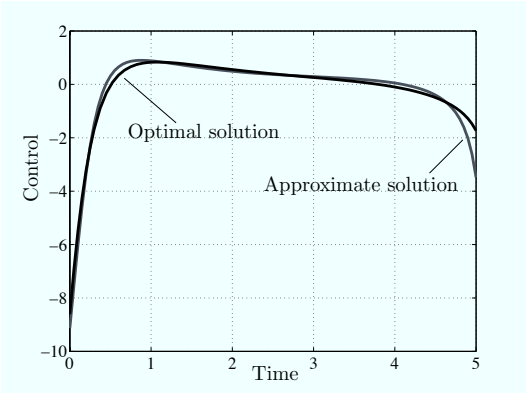
(a) Phase space iterations.



(b) Control profile iterations.



(c) Approximate and optimal orbit.



(d) Approximate and optimal control.

Figure 7. Problem 2: iterated, approximate, and optimal solution.

Mars transfer with the planets moving in circular, coplanar orbits.[33,34]) Eqs. (57) have been factorized by using

$$A_1(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\dfrac{1}{x_1^3} & 0 & 0 & x_1 x_4 \\ 0 & 0 & -2\dfrac{x_4}{x_1} & 0 \end{bmatrix}, \qquad A_2(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ x_4^2 - \dfrac{1}{x_1^3} & 0 & 0 & 0 \\ 0 & 0 & -2\dfrac{x_4}{x_1} & 0 \end{bmatrix},$$

$$A_3(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x_4 x_1 - \dfrac{1}{x_4 x_1^2} \\ 0 & 0 & -2\dfrac{x_4}{x_1} & 0 \end{bmatrix}, \qquad B(\mathbf{x}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \dfrac{1}{x_1} \end{bmatrix}. \tag{59}$$

It can be noticed that all the A_i $(i = 1, 2, 3)$ as well as B are not Lipschitz continuous when $x_1 \to 0$ (and $x_4 \to 0$ in A_3 only). These conditions are however never expected to occur

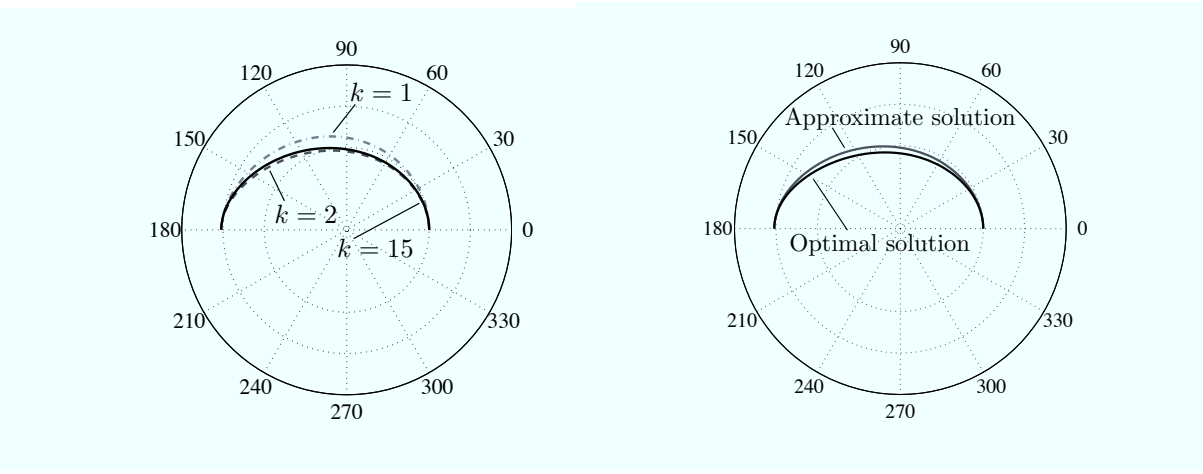
along the solutions of the present problem, and therefore the MASRE is run with the input as in (59). Moreover, the origin is not an equilibrium point of the system, and the conditions for writing (57) into a state-dependent form as per [18] are violated. The objective function (58) is written in the form (6) with $Q = 0_{4\times4}$, $R = I_{2\times2}$, and, $S = 0$ (in the MCP only).

The HCP is solved with 15 iterations, and its most relevant figures are reported in Table 5 (in the first iteration it is not possible to ascertain the closest parent factorization). It is interesting to notice that the factorization associated to A_1 is discarded because of its non-controllability, although the associated ASRE is able to converge (when fed with A_1) to a cost higher than that in Table 5, $k = 15$. This means that the MASRE algorithm prunes away the less performing SDC matrices, and uses the ones that likely produce the best solution. The approximate solutions for some sample iterations are shown in Figures 8(a), 8(c), and 8(e). When refined, the approximate solution leads to the optimal solution in Figures 8(b), 8(d), 8(f). The optimal cost is $J^{(\mathrm{opt})} = 1.830 \times 10^{-1}$, which is 18% less than J in Table 5, $k = 15$. This is mostly due to the difference in the u_2 profiles (see Figure 8(f)).

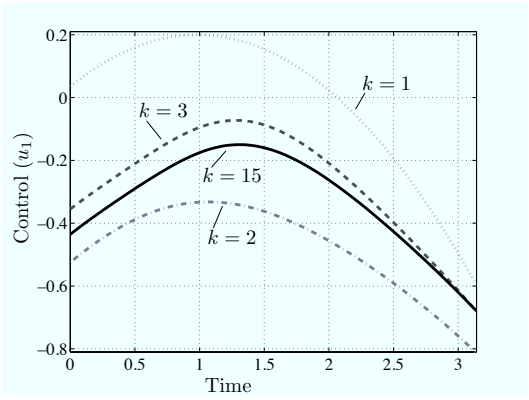
Table 5. MASRE iterations for Problem 3 (HCP), CPU time 60.0 s.

k	Error	$\sigma_{\min}(P)$	J	Parent
1	3.142	4.058×10^{-1}	1.674×10^{-1}	–
2	1.760×10^{-1}	3.519×10^{-1}	3.956×10^{-1}	A_2
3	5.618×10^{-2}	3.095×10^{-1}	1.940×10^{-1}	A_2
4	3.565×10^{-2}	3.082×10^{-1}	2.423×10^{-1}	A_2
5	1.179×10^{-2}	3.040×10^{-1}	2.184×10^{-1}	A_2
6	5.335×10^{-3}	3.049×10^{-1}	2.279×10^{-1}	A_2
7	1.939×10^{-3}	3.046×10^{-1}	2.244×10^{-1}	A_2
8	6.792×10^{-4}	3.048×10^{-1}	2.258×10^{-1}	A_2
9	2.533×10^{-4}	3.047×10^{-1}	2.253×10^{-1}	A_2
10	8.548×10^{-5}	3.047×10^{-1}	2.255×10^{-1}	A_2
11	3.057×10^{-5}	3.047×10^{-1}	2.254×10^{-1}	A_2
12	1.074×10^{-5}	3.047×10^{-1}	2.254×10^{-1}	A_2
13	3.699×10^{-6}	3.047×10^{-1}	2.254×10^{-1}	A_2
14	1.320×10^{-6}	3.047×10^{-1}	2.254×10^{-1}	A_2
15	4.567×10^{-7}	3.047×10^{-1}	2.254×10^{-1}	A_2

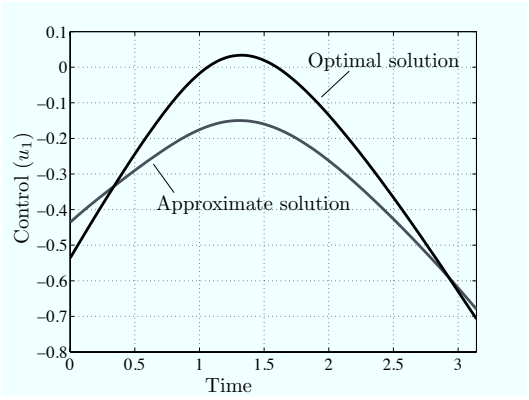
The sequence of iterations for the MCP is reported in Table 6. In this case, the algorithm converges in 13 iterations and uses again A_2 as parent factorization. The cost of the refined, optimal solution is $J^{(\mathrm{opt})} = 3.694 \times 10^{-2}$, which is 23% less than that in Table 6, $k = 13$. A set of iterated solutions, as well as the final approximated and the optimal solution are shown in Figure 9. As it would have been expected, by letting the algorithm to determine the final angle, the MCP cost function is lower than that of the HCP, and even the convergence is



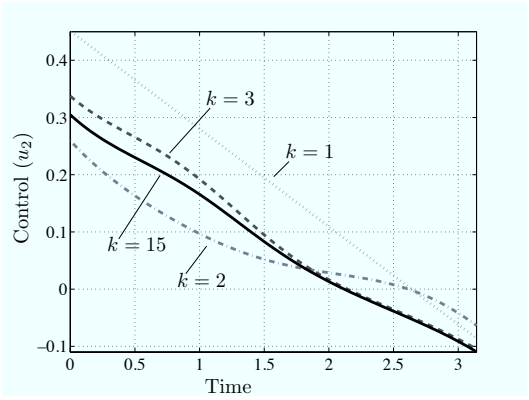
(a) Transfer orbit iterations. (b) Approximate and optimal transfers



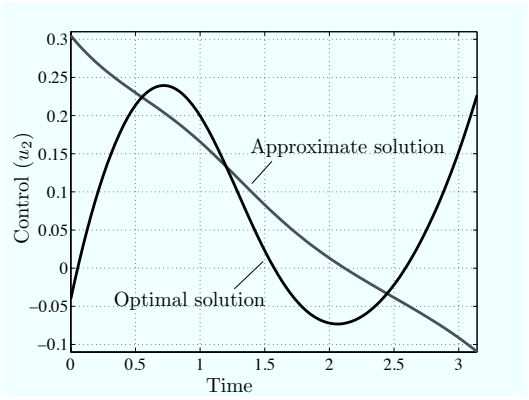
(c) Control u_1 iterations.



(d) Approximate and optimal control u_1.



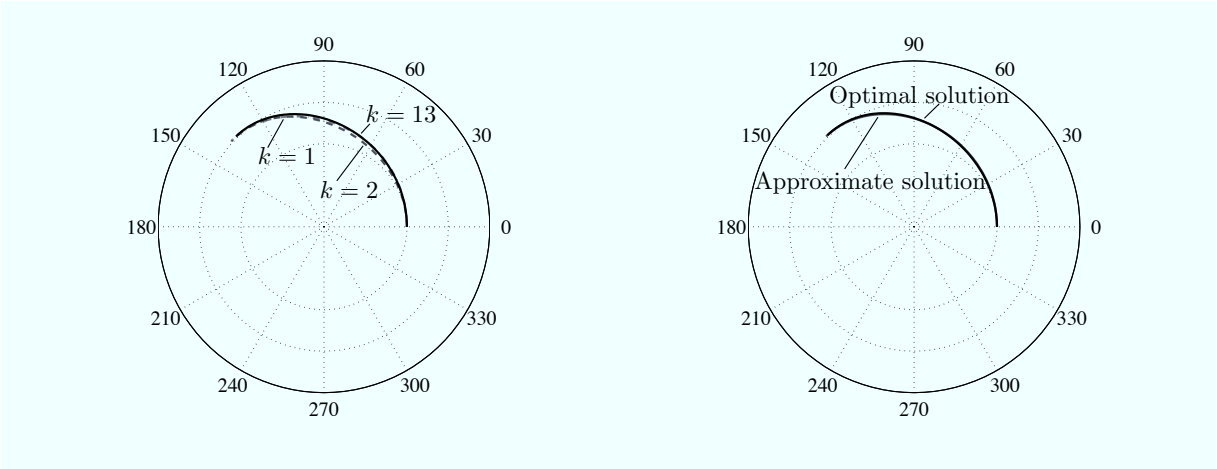
(e) Control u_2 iterations.



(f) Approximate and optimal control u_2.

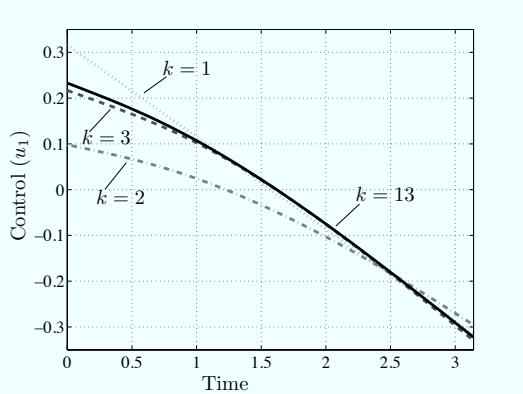
Figure 8. Problem 3 (HCP): iterated, approximate, and optimal solution.

faster, since the additional degree of freedom represented by the unspecified final phase is exploited. The optimal solution in Figure 9 can be compared with that in [2], where the minimum-time problem is solved for the same set up. Albeit with proper differences, transfer orbit and the thrust profile are similar.

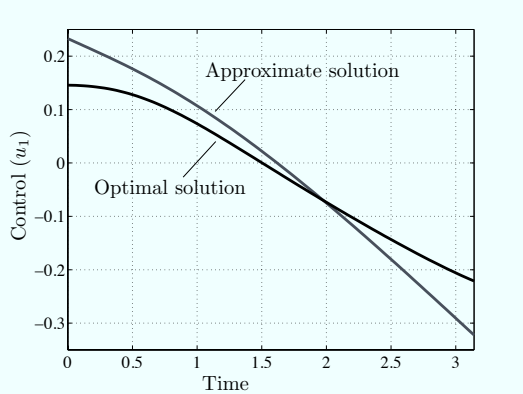


(a) Transfer orbit iterations.

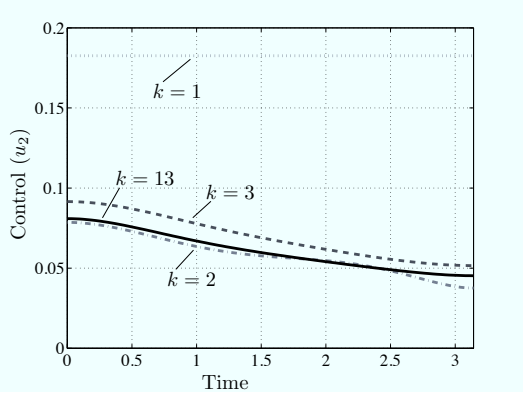
(b) Approximate and optimal transfers



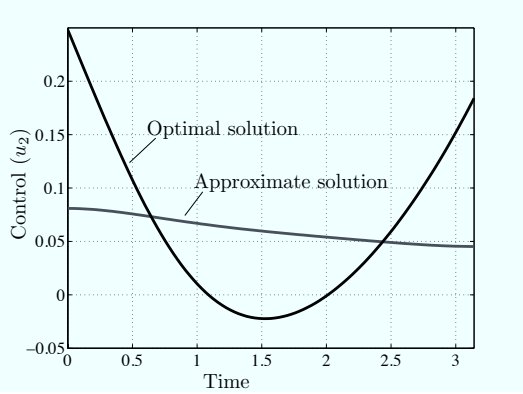
(c) Control u_1 iterations.



(d) Approximate and optimal control u_1.



(e) Control u_2 iterations.



(f) Approximate and optimal control u_2.

Figure 9. Problem 3 (MCP): iterated, approximate, and optimal solution.

Table 6. MASRE iterations for Problem 3 (MCP), CPU time 19.9 s.

	Error	σ_{min}	J	Matrix
1	2.409	4.058×10^{-1}	1.047×10^{-1}	–
2	5.237×10^{-2}	3.434×10^{-1}	3.297×10^{-2}	A_2
3	4.522×10^{-2}	3.146×10^{-1}	4.924×10^{-2}	A_2
4	8.627×10^{-3}	3.126×10^{-1}	4.652×10^{-2}	A_2
5	2.728×10^{-3}	3.127×10^{-1}	4.801×10^{-2}	A_2
6	5.133×10^{-4}	3.129×10^{-1}	4.839×10^{-2}	A_2
7	3.166×10^{-4}	3.131×10^{-1}	4.845×10^{-2}	A_2
8	1.568×10^{-4}	3.132×10^{-1}	4.846×10^{-2}	A_2
9	5.434×10^{-5}	3.132×10^{-1}	4.845×10^{-2}	A_2
10	1.475×10^{-5}	3.132×10^{-1}	4.845×10^{-2}	A_2
11	3.045×10^{-6}	3.132×10^{-1}	4.845×10^{-2}	A_2
12	1.325×10^{-6}	3.132×10^{-1}	4.845×10^{-2}	A_2
13	7.700×10^{-7}	3.132×10^{-1}	4.845×10^{-2}	A_2

VI. Conclusion

Approximate methods for solving nonlinear optimal control problems aim to circumvent the classic Euler–Lagrange equations by using ad hoc, simpler schemes. As this eases the search for a feasible solution, it does not guarantee optimality, as in general these methods lead to sub-optimal solutions. This is the case of the approximating sequence of Riccati equations, which solves the optimal control problem as a sequence of time-varying linear quadratic regulators. This method depends upon the factorization chosen for the uncontrolled dynamics. It has been shown that even for a simple problem, two different factorizations lead to two different solutions, in terms of system trajectory, objective function, and algorithm iterations. This work makes it possible to automatically select the best factorization, from a set of candidate ones, during the solution process. This is done by maximizing, at each iteration, the controllability Gramian associated to a family of factorizations. The augmented effort consists in solving a nonlinear programming problem at each iteration. The developed method can be used also in other state dependent control schemes. Examples show that the method is effective, and approximates well the true optimal solution, though it would be useful to extend it to accommodate further features (e.g., control saturation, variable final time, non-affinity of control, etc.).

References

[1]Pontryagin, L., Boltyanskii, V., Gamkrelidze, R., and Mishchenko, E., *The Mathematical Theory of Optimal Processes*, John Wiley & Sons, New York, 1962, pp. 17–21.

[2]Bryson, A. and Ho, Y., *Applied Optimal Control*, John Wiley & Sons, New York, 1975, pp. 65–69, pp. 150–151.

[3]Betts, J., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM, Philadelphia, 2010, pp. 15–17.

[4]Conway, B., *Spacecraft Trajectory Optimization*, chap. Spacecraft Trajecory Optimization Using Direct Transcription and Nonlinear Programming, Cambridge University Press, 2010, pp. 37–78.

[5]Pearson, J., "Approximation Methods in Optimal Control," *Journal of Electronics and Control*, Vol. 13, 1962, pp. 453–469.

[6]Wernli, A. and Cook, G., "Suboptimal Control for the Nonlinear Quadratic Regulator Problem," *Automatica*, Vol. 11, 1975, pp. 75–84.

[7]Cloutier, R., D'Souza, C. A., and Mracek, C. P., "Nonlinear Regulation and Nonlinear H-infinity Control via the State-Dependent Riccati Equation Technique: Part 1, Theory; Part 2, Examples," *Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace*, 1996, pp. 117–141.

[8]Mracek, C. and Cloutier, J., "Control Designs for the Nonlinear Benchmark Problem via the State-Dependent Riccati Equation Method," *International Journal of Robust and Nonlinear Control*, Vol. 8, 1998, pp. 401–433.

[9]Bracci, A., Innocenti, M., and Pollini, L., "Estimation of the Region of Attraction for State-Dependent Riccati Equation Controllers," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1427–1430.

[10]Çimen, T., "Survey of State-Dependent Riccati Equation in Nonlinear Optimal Feedback Control Synthesis," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 4, 2012, pp. 1025–1047.

[11]Harman, R. R. and Bar-Itzhack, I. Y., "Pseudolinear and State-Dependent Riccati Equation Filters for Angular Rate Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 5, 1999, pp. 723–725.

[12]Tadi, M., "State-Dependent Riccati Equation for Control of Aeroelastic Flutter," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 914–917.

[13]Bhoir, N. and Singh, S. N., "Control of Unsteady Aeroelastic System via State-Dependent Riccati Equation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 1, 2005, pp. 78–84.

[14]Bogdanov, A. and Wan, E. A., "State-Dependent Riccati Equation Control for Small Autonomous Helicopters," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 47–60.

[15]Kim, C.-J., Park, S. H., Sung, S. K., and Jung, S.-N., "Nonlinear Optimal Control Analysis Using State-Dependent Matrix Exponential and Its Integrals," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 309–313.

[16]Vaddi, S., Menon, P. K., and Ohlmeyer, E. J., "Numerical State-Dependent Riccati Equation Approach for Missile Integrated Guidance Control," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, 2009, pp. 699–703.

[17]Ratnoo, A. and Ghose, D., "State-Dependent Riccati-Equation-Based Guidance Law for Impact-Angle-Constrained Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 320–325.

[18]Çimen, T. and Banks, S., "Global Optimal Feedback Control for General Non Linear System with Non-Quadratic Performance Criteria," *System & Control Letters*, Vol. 53, 2004, pp. 327–346.

[19]Çimen, T. and Banks, S., "Nonlinear Optimal Tracking Control with Application to Super-Tankers for Autopilot Design," *Automatica*, Vol. 40, 2004, pp. 1845–1863.

[20]Topputo, F. and Bernelli-Zazzera, F., "Optimal Low-Thrust Stationkeeping of Geostationary Satellites," *Proceedings of the* 3rd *CEAS Air&Space Conference*, 2011, pp. 1917–1925.

[21]Topputo, F. and Bernelli-Zazzera, F., "A Method to Solve Nonlinear Optimal Control Problems in Astrodynamics," *Advances in the Astronautical Sciences*, Vol. 145, 2012, pp. 1531–1544.

[22]Topputo, F. and Bernelli-Zazzera, F., "Approximate Solutions to Nonlinear Optimal Control Problems in Astrodynamic," *ISRN Aerospace Engineering*, Vol. 2013, No. Article ID 950912, 2013, pp. 1–7.

[23]Haim, L. and Choukroun, D., "Optimized State-Dependentependent Riccati Equation Method for Spacecraft Attitude Estimation and Control," *Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, 13–16 August, 2012, Paper AIAA-4557*, 2012, pp. 1–21.

[24]Hammett, K., Hall, C., and Ridgely, D., "Controllability Issues in Nonlinear State-Dependent Riccati Equation Control," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 5, 1998, pp. 767–773.

[25]Friedland, B., *Control System Design: An Introduction to State-Space Methods*, McGraw-Hill, New York, 1987, pp. 203–209.

[26]Liang, Y.-W. and Lin, L.-G., "Analysis of SDC Matrices for Successfully Implementing the SDRE Scheme," *Automatica*, Vol. 49, No. 10, 2013, pp. 3120–3124.

[27]Kang, W. and Xu, L., "A Quantitative Measure of Observability and Controllability," *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009, pp. 6413–6418.

[28]Paige, C., "Properties of Numerical Algorithms Related to Computing Controllability," *IEEE Transactions on Automatic Control*, Vol. AC–26, No. 1, February 1981, pp. 130–138.

[29]Eising, R., "Between Controllable and Uncontrollable," *Systems & Control Letters*, Vol. 4, No. 5, 1984, pp. 263–264.

[30]Cloutier, J. R. and Stansbery, D. T., "The Capabilities and Art of State-Dependent Riccati Equation-Based Design," *American Control Conference*, Vol. 1, IEEE, 2002, pp. 86–91.

[31]Çimen, T., "Systematic and Effective Design of Nonlinear Feedback Controllers via the State-Dependent Riccati Equation (SDRE) Method," *Annual Reviews in Control*, Vol. 34, 2010, pp. 32–51.

[32]Nazari, S. and Shafai, B., "Robust SDC Parameterization fo a Class of Extended Linearization Systems," *2011 American Control Conference, San Francisco, CA, USA, June 29 – July 01*, 2011, pp. 3742–3747.

[33]Topputo, F., Owis, A., and Bernelli-Zazzera, F., "Analytical Solution of Optimal Feedback Control for Radially Accelerated Orbits," *Journal of Guidance, Control and Dynamics*, Vol. 31, No. 5, September–October 2008, pp. 1352–1359.

[34]Owis, A., Topputo, F., and Bernelli-Zazzera, F., "Radially Accelerated Optimal Feedback Orbits in Central Gravity Field with Linear Drag," *Celestial Mechanics and Dynamical Astronomy*, Vol. 103, No. 1, January 2009, pp. 1–16.