A NEW NONLINEAR NEAR-OPTIMAL CONTROL FOR SPACE ROBOTIC SYSTEMS

Z. Qu* and Y. Jin*

Abstract

The robotic system considered in this article consists of an n-DOF robotic manipulator mounted onto an altitude-controlled base (which is either a space shuttle or a space station). To minimize control energy of thrust in maintaining the altitude of the base, nonlinear optimal control problem of robotic manipulator is formulated. Nonlinear optimal controls are outlined, based on which a new nonlinear near-optimal control is proposed for space robotic systems. Compared to an optimal control, which must be solved offline and stored numerically, the proposed control scheme can easily be implemented real time, and its closeness to the optimal control can be measured. Stability results under the proposed suboptimal control are obtained, and a simulation example is included to demonstrate its effectiveness.

Key Words

Space robots, optimal control, near optimality, Riccati equation, stability

1. Introduction

As demonstrated by Sojoutner robot in the mission of Mars exploration, robotic manipulators have been instrumental and will continue to play an important role in space applications. The success of robotic manipulation in inspection/deployment/retrieval/exploration demonstrates the need for precision control of robotic systems. Control of robotic manipulators have been extensively studied in the literature over the last two decades. Typical control strategies such as PD control, PID control, adaptive control, learning control, and robust control can be found in texts [1–8]. These controls have been demonstrated to be very effective in controlling rigid body manipulators. However, two distinct criteria in addition to the standard robot control objectives must be met in designing a successful control for space robotic systems: minimizing control energy and maintaining a fixed altitude for the base. Any change in base altitude is undesirable because it causes problems for many of the communication devices onboard the base.

According to momentum conservation law, any torque exerted by the end-effector of the space manipulator must be balanced by thrust and/or momentum wheels on the base in order to keep the base altitude fixed. Therefore, the problem of minimizing thrust fuel and electrical energy onboard can be cast into a combined problem of offline trajectory planning and real-time optimal control of the manipulator. To simplify the discussion of this article, the altitude of the base is assumed to be stabilized instantaneously; that is, the altitude is assumed to be fixed.

As dynamics of a robotic manipulator are inherently nonlinear, optimal control of such a nonlinear system is not trivial. Two approaches are available to design nonlinear optimal controls: Pontryagin minimum principle (Euler-Lagrange method) and Hamilton-Jacobi-Bellman theory. In this work, we begin by briefly reviewing these two techniques. It is well known that a nonlinear optimal control design requires a solution to a two-point boundary value problem (for either a vector differential equation or a vector partial differential equation). Such a solution can only be found iteratively through numerical search and thus offline, which is the main obstacle in applications (such as space robotics) of nonlinear optimal control. This prompts us to introduce a new design for generating a near-optimal, closed-form control for real-time implementation. The proposed design is an extension of the state-dependent algebraic Riccati equation technique. Specifically, the proposed control contains two parts: one based on the solution to the state-dependent algebraic Riccati equation, and another that minimizes a distance measure between the first part of the proposed control and an optimal control. It is the second part that makes the proposed control as near optimal as possible on the basis of real-time implementation and closed-form solution. Stability analysis also shows that minimization of the distance measure enhances global stability of the proposed control. In short, the proposed design achieves near optimality, is efficient in real-time implementation, and has a guaranteed stability and performance.

The article is organized into six sections. In Section 2 a model of space robotic systems and the corresponding optimal control problem are presented. In Section 3 the near-optimal control design is presented with stability analysis on the proposed control. An illustrative example is included in Section 4, and conclusions are given in

* School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816, USA; e-mail: qu@mail.ucf.edu

(paper no. 806-8609)
2. Dynamics and Control of Space Robotic Systems

Dynamic equation of a rigid-body mechanical system can be derived by using the Lagrange's method, which is given by the following partial differential equation:

\[
\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \tau - F_d(t)\dot{q} - F_s(\dot{q}) - T_d \tag{1}
\]

where \( q \in \mathbb{R}^n \) is a vector of generalized coordinates, \( \dot{q} \) denotes the time derivative of \( q \), \( L \) is the Lagrangian of the system defined by:

\[
L(q, \dot{q}) = K(q, \dot{q}) - P(q), \quad K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}, \tag{2}
\]

\( K(q, \dot{q}) \) represents the kinetic energy, \( M(q) \) is the inertia matrix, \( P(q) \) denotes the potential energy, \( \tau \in \mathbb{R}^n \) is the vector of generalized external input functions (that is, torques by actuators), \( F_d \in \mathbb{R}^{n \times n} \) is a diagonal matrix of viscous and/or dynamic friction coefficients, \( F_s(\dot{q}) \in \mathbb{R}^n \) is the vector of unstructured friction effects such as static friction terms, and \( T_d \in \mathbb{R}^n \) is the vector of any generalized input due to disturbances or unmodelled dynamics. The right-hand side of (1) contains all external forces applied to the robotic system.

By convention, dynamics of a rigid-body robotic manipulator are written symbolically in terms of the following second-order ordinary differential equation:

\[
\tau = M(q)\ddot{q} + N(q, \dot{q}) \tag{3}
\]

where \( M(q) \in \mathbb{R}^{n \times n} \) denotes the inertia matrix, \( N(q, \dot{q}) \) is the lumped sum of all nonlinearities given by:

\[
N(q, \dot{q}) = V_m(q, \dot{q})\dot{q} + G(q) + F_d(q) + F_s(q) + T_d
\]

\( V_m(q, \dot{q}) \in \mathbb{R}^{n \times n} \) is the matrix containing centripetal and Coriolis terms, \( G(q) \in \mathbb{R}^n \) is the vector containing gravity terms, and \( q(t) \in \mathbb{R}^n \) is the vector of joint variables. Equation (3) can be derived directly from Lagrange formulation (1) by simply substituting (2) into (1). Detailed derivation of dynamic equation (3) for space robotic systems can be found in [9, 10].

Let \( q_d(t) \) be the desired trajectory computed offline to minimize the use of thrust fuel at the base without considering state deviation, and let:

\[
\tau = \tau_d + M(q)u \tag{4}
\]

where \( u \) is the feedback control to be designed, and \( \tau_d \) is computed according to (3) as:

\[
\tau_d = M(q_d)\ddot{q}_d + N(q_d, \dot{q}_d) \tag{5}
\]

Let the state of the system be:

\[
x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \triangleq \begin{bmatrix} q - q_d \\ \dot{q} - \dot{q}_d \end{bmatrix}
\]

It follows from equations (3), (4), and (5) that system dynamics can be rewritten as:

\[
\dot{x} = f(x, t) + Bu \triangleq f(x, t) + g(x, t)u \tag{6}
\]

where \( x(t_0) = x_0 \) is the given initial condition:

\[
f(x, t) = M^{-1}(q) \{ [M(q_d) - M(q)]\ddot{q}_d + N(q_d, \dot{q}_d) - N(q, \dot{q}) \},
\]

\[
B = \begin{bmatrix} 0 & I_n \end{bmatrix}^T
\]

The control design objective considered in this article is to find a real-time feedback control law \( u(\cdot) \) that ensures global stability and its optimality (or near optimality).

To determine a stabilizing control \( u(\cdot) \) while minimizing control energy, one can employ the standard nonlinear optimization technique, and the resulting control is called nonlinear optimal control. In optimization, the following type of cost functional is often used to be the performance criterion:

\[
J(t_0, t_f, x_0, u) = \psi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t))dt \tag{7}
\]

in which both functions \( \psi(\cdot) \) and \( L(\cdot) \) are positive semidefinite with respect to their arguments. In addition, a terminal condition in the form of:

\[
\psi(x(t_f), t_f) = 0 \in \mathbb{R}^l \quad (0 \leq l \leq n) \tag{8}
\]

may be imposed. In what follows, two different optimal control designs to minimize (7), based on the Pontryagin minimum principle and Hamilton-Jacobi-Bellman theory, will be reviewed.

2.1 Euler-Lagrange Method

To optimize performance index (7) for system (6) with terminal condition (8), we can use the so-called Lagrange multiplier to convert the equality-constrained optimization problem into an unconstrained one. Specifically, system equation (6) and terminal condition (8) can be adjoined into performance index \( J \) as:

\[
J(t_0, t_f, x_0, u) = \phi(x(t_f), t_f) + \mu^T \psi(x(t_f), t_f) - \lambda^T(x(t_f) - x(t_0)) + \int_{t_0}^{t_f} \{ H(x, u, \lambda, t) + \lambda^T \dot{x} \} dt
\]

where \( \mu \in \mathbb{R}^l \) and \( \lambda \in \mathbb{R}^2n \) are Lagrange multipliers, and \( H(\cdot) \) is the so-called Hamiltonian defined by:
\[ H(x, u, \lambda, t) = L(x(t), u(t)) + \lambda^T [f(x, t) + g(x, t)u] \]  

(9)

The Euler-Lagrange method is based on the calculus of variations [11]. To minimize $J$ locally, the first-order variation $\delta^1 J$ must be zero for arbitrary variation $\delta u$. It is easy to show that $\delta J = 0$ under the following conditions:

\[ \lambda = -\frac{\partial H}{\partial x} \Rightarrow \frac{\partial L}{\partial x} - \frac{\partial f}{\partial x} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x} u_j \]  

(10)

and:

\[ 0 = \frac{\partial H}{\partial u} \Rightarrow 0 = \frac{\partial L}{\partial u} + g^T (x, t) \lambda \]  

(11)

and:

\[ \dot{\lambda}(t_f) = \left( \frac{\partial \phi}{\partial x} + \mu^T \frac{\partial \psi}{\partial x} \right) \bigg|_{t=t_f} \]  

(12)

The above conditions, together with state equation (6), are necessary conditions for local optimality. Equation (12) is a boundary condition on $\lambda(t)$ at time $t = t_f$, and state $x$ has initial condition $x(t_0) = x_0$. Equation (10) is called a costate equation, which must be solved simultaneously with state equation (6). The state and co-state equations consist of 2n first-order differential equations that, combined with initial condition $x_0$ and terminal conditions (8) and (12), represent a two-point boundary-value (TPBV) problem. A successful solution must satisfy $2n$ boundary conditions, $n$ of which are given by initial condition $x(t_0) = x_0$, $l$ are given by (8), and the rest $n-l$ are part of equations in (12) (the remaining equations in (12) can be satisfied by the choice of $\mu \in \mathbb{R}^l$). Such a solution to such TPBV problems can only be found numerically, and techniques available include finite difference method, integral method, shooting method, and differential dynamic programming [14–16]. All these methods are to improve an initially guessed solution through successive approximations, and they cannot be implemented in real time.

Conditions (10) to (12) are generally not sufficient for determining an optimal control. For their solution to be a local minimum of $J$, not only must we have $\delta J = 0$, but also the second-order variation of $J$, denoted by $\delta^2 J$, must be positive. A sufficient condition is that, if $\psi = 0$ and $\phi(\cdot)$ is a convex function of $x$, the Hessian of Hamiltonian $H$ denoted by:

\[ H_x = \begin{bmatrix} \frac{\partial^2 H}{\partial x^2} & \frac{\partial^2 H}{\partial x \partial u} \\ \frac{\partial^2 H}{\partial x \partial u} & \frac{\partial^2 H}{\partial u^2} \end{bmatrix} > 0 \]  

(13)

is positive-definite for all infinitesimal values of $\delta u$.

If integrant $L(\cdot)$ is positive-definite with respect to control $u$, condition (11) can be used to solve for the candidates of optimal control. In fact, it has been shown in [15] that condition (11) can be strengthened to be:

\[ H(x^*, \lambda^*, u^*, t) = \min_{u \in \Omega_u} H(x^*, \lambda^*, u, t) \]  

(14)

where $\Omega_u$ is a set of admissible control, $u^*$ is the solution to (11), and $x^*$ and $\lambda^*$ are the state and costate trajectories corresponding to $u^*$. Set $\Omega_u$ to be describe constraints (if any) on control $u$, and it is chosen to be $\mathbb{R}^m$ in the subsequent discussion (that is, no constraint considered). Condition (14) shows that function $H(x^*, \lambda^*, u, t)$ has an absolute minimum as a function of $u$ over $\Omega_u$, and it must be used if $\Omega_u \neq \mathbb{R}^m$. As a result, condition (14) together with (10) and (12) is also referred as the Pontryagin minimum principle.

### 2.2 Principle of Optimality

An alternative way to derive optimal control is to imbed the optimal control problem into a more general problem by considering the performance measure:

\[ J(t, t_f, x, u) = \phi(x(t_f), t_f) + \int_t^{t_f} L(x(\tau), u(\tau))d\tau \]  

(15)

where $t \in (t_0, t_f)$. The following principle of optimality can be easily concluded by contradiction. Suppose that $u^*$ is an optimal control for system (6), with respect to performance index (7), and with initial condition $x(t_0) = x_0$. Let $x^*$ denote the corresponding optimal state trajectory; then the same control will also be optimal for system (6), with respect to performance index (15), and with initial condition $x(t) = x^*(t)$. Applying the principle of optimality and Taylor series expansion to (15), one can show that the necessary condition to the optimal control problem is described by the following Hamilton-Jacobi equation:

\[ \frac{\partial J^*(t, t_f, x)}{\partial t} = -\min_{u \in \Omega_u} H(x, u, \lambda, t) \bigg|_{\lambda = \frac{\partial J^*(t, t_f, x)}{\partial x}} \]  

(16)

whose boundary conditions are the initial condition given by $x(t)$ and the terminal condition imposed by (15), namely:

\[ J^*(t_f, t_f, x(t_f)) = \phi(x(t_f), t_f) \]  

(17)

Once again, the optimal control problem is a TPBV problem. And a solution to partial differential equation (16) can only be found numerically.

A sufficient condition for local optimality in the Hamilton-Jacobi approach is stated by the following lemma. If $X = \mathbb{R}^{n+1}$, it becomes a sufficient condition for global optimality.

**Lemma 1** [16]. Suppose that $\phi(x(t_f), t_f) = 0$ in (7) and that, for each point $(x, t) \in X \subset \mathbb{R}^{n+1}$, the Hamiltonian in (9) has, as a function of $u$, a unique absolute minimum with respect to all $u$ in $\Omega_u \subset \mathbb{R}^m$ at:

\[ u = u^*(x, \lambda) \]

If there is a solution $J^*(t, t_f, x)$ to the Hamilton-Jacobi equation (16) and if the resulting control is given by:

\[ u^*(x, t) \triangleq u^* \left( x, \frac{\partial J^*(t, t_f, x)}{\partial x}, t \right) \]  

(18)
under which the state remains in \( X \) and satisfies its boundary conditions, then control \( u^{*}(x, t) \) given by (18) is an optimal control with respect to all controls in \( \Omega \) that keep the system trajectory entirely in \( X \). Moreover, \( J^*(t_0, t_f, x_0) \) is the optimal cost for performance index (7) in the sense that \( (x, t) \in X \) and that:

\[
J^*(t_0, t_f, x_0) = J(t_0, t_f, x_0, u^*) = \min_{u \in \Omega} J(t_0, t_f, x_0, u)
\]

2.3 Regulation and Nonlinear Matrix Representation

It is well known that performance of an optimal control system is determined by the criterion of optimization. As energy consumption is a critical factor, the performance index should measure the total control effort, that is, \( t_f = \infty \) and:

\[
J(x_0, u) = \int_{t_0}^{\infty} L(x(t), u(t)) dt
\]  (19)

The choice of \( L(x, u) \) depends upon the control objective. In the previous section, the tracking problem is formulated with \( q_0(t) \) being the desired trajectory. Control energy can be reduced by relaxing transient requirements, and thus optimal control of space robots should be done using the set-point tracking framework, that is, \( q_0(t) = q_d \) and \( j_d = 0 \). In this case, system equation (6) becomes:

\[
\dot{x} = f(x) + Bu
\]  (20)

with initial condition \( x(t_0) = x_0 \), and the corresponding Hamiltonian is:

\[
H(x, u, \lambda) = L(x, u) + \lambda^T [f(x) + Bu]
\]

For systems of form (20), nonlinear matrix representation can be applied to express and calculate optimal control and suboptimal controls. To this end, we need to express system dynamics, performance index, and Lagrange multiplier into their matrix representations and, if any of the matrices is state dependent, the resulting representation is nonlinear. For robotic systems, nonlinearities in \( f(x) \) contain only quadratic and sinusoidal functions, and performance index is typically quadratic like. Therefore, the following matrix representation can be easily performed:

\[
f(x) = A(x)x, \quad L(x(t), u(t)) = \frac{1}{2} x^T Q(x)x + \frac{1}{2} u^T R(x)u,
\]

and \( \lambda = P(x)x \)  (21)

where matrices \( A(x), Q(x), R(x), \) and \( P(x) \) are locally bounded. Obviously, the matrix representation includes local linearization representations as a special case. As matrices \( Q(x) \) and \( R(x) \) appear in quadratic-type function \( (-) \), they can be assumed without loss of any generality to be symmetric. For simplicity, we shall choose in the subsequent discussion that:

\[
Q(x) = e^{q(x)}Q_0, \quad R(x) = e^{r(x)}R_0 \quad (22)
\]

for some constant positive-definite matrices \( Q_0 \) and \( R_0 \) and for some scalar, non-negative functions \( q(x) \) and \( r(x) \).

In terms of nonlinear matrix representation, necessary conditions, sufficient conditions, and boundary conditions of the optimal control problem become more explicit. If the Euler-Lagrange method is used together with matrix representation (21), necessary condition (11) can be satisfied by choosing the optimal control as:

\[
u = -R^{-1}(x)B^T P(x)x \quad (23)
\]

and necessary condition (10) becomes:

\[
0 = \dot{P}(x)x + [P(x)A(x) + A^T(x)P(x)] - P(x)BR^{-1}(x)B^T P(x)x + \frac{1}{2} \frac{\partial r(x)}{\partial x} x^T Q(x)x + \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x_i} P x \right\}
\]  (24)

where boundary conditions are \( x(t_0) = x_0 \) and \( P(x(\infty)) = S \) is a bounded and constant matrix, and:

\[
\text{vec}(z_i) \triangleq \begin{bmatrix} z_1 & \cdots & z_{2n} \end{bmatrix}^T
\]

Noting that performance index (19) is convex under the choice of (21), we know that any stationary point will at least be a local optimum and hence sufficient condition (13) need not be checked.

On the other hand, because system dynamics in (20) do not explicitly depend on \( t \), \( J(t, t_f, x, u) = J(t_f - t, x, u) \) in (15) is a function of \( t_f - t \) and \( x \). Furthermore, as \( (t_f - t) \to \infty \) (that is, \( t_f = \infty \)), boundary condition (17) implies that function \( J(t_f - t, x, u) = J(x, u) \), if it exists, is a function of only \( x \). Therefore, if Hamilton-Jacobi theory is applied to determine optimal control in this case, the left hand of Hamilton-Jacobi equation (16) is zero. In fact, by adopting matrix representation (21), we know that the partial derivative of performance index \( J(x, u) \) must be:

\[
P(x)x = \frac{\partial J(x, u)}{\partial x}
\]  (25)

that the optimal control is given by (23), and that \( P(x) \) is the solution to the Hamilton-Jacobi equation:

\[
P^T(x)A(x) + A^T(x)P(x) + Q(x) - P^T(x)BR^{-1}(x)B^T P(x) = 0
\]  (26)

To find the optimal value of \( J(x, u) \) from (25) or, equivalently, to solve for \( P(x) \) from (26), note that, because
\( J(x,u) \) is a scalar function, its solution exists if its Hessian matrix is symmetrical, that is:

\[
P_{ij}(x) + \sum_{k=1}^{2n} \frac{\partial P_{jk}(x)}{\partial x_j} x_k = P_{ji}(x) + \sum_{k=1}^{2n} \frac{\partial P_{jk}(x)}{\partial x_i} x_k \tag{27}
\]

where boundary conditions are \( x(t_0) = x_0 \) and \( x(\infty) = 0 \). It follows from Lemma 1 that, if it exists, solution \( P(x) \) to (26) and (27) satisfies the sufficient condition for optimality. With the exception that \( P(x) \in \mathbb{R}^{2n \times 2n} \) is not required to be symmetric and that \( P(x) \) is state dependent in general, Hamilton-Jacobi equation (26) is in the same form of the standard algebraic Riccati equation. Due to its symmetry, Hamilton-Jacobi equation (26) itself represents \( n(2n + 1) \) simultaneous equations, and additional \( (2n - 1)n \) equations needed to determine matrix \( P(x) \) are partial differential equations in (27).

Summarizing the above discussion, we know that equation (24) is equivalent to the combination of (26) and (27) for finding the optimal control, that both of them are TPBV problems, that solution \( P(x) \) is in general state dependent and not necessarily symmetrical. Thus, either equation (24) or (26) can be called the optimality condition, and equation (27) is called the symmetry condition. It follows from Hamilton-Jacobi equation (26) that optimality condition (24) can be simplified to be:

\[
0 = \dot{P}(x)x + [P(x) - P^T(x)]A(x)x \\
+ [P^T(x) - P(x)]BR^{-1}BTP(x)x \\
+ \frac{1}{2} \frac{\partial r(x)}{\partial x} x^T P^T(x)BR^{-1}(x)BTP(x)x \\
+ \frac{1}{2} \frac{\partial q(x)}{\partial x} x^T Q(x)x \\
+ \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x_i} P x \right\} \tag{28}
\]

If the system under consideration is linear, the solution \( P \) is constant and symmetrical, equations (24) and (26) reduce to the well-known algebraic Riccati equation, and symmetry condition (27) is trivial.

3. A New Suboptimal Control

To find a real-time control while avoiding the difficulty of solving a two-point boundary value problem, a suboptimal control is sought. The proposed near-optimal control is motivated by two observations. First, the proposed control should include existing ones as special cases. For instance, it is well known that, around the origin, the optimal solution of \( P(0) \) is symmetrical. To this end, the technique of nonlinear matrix representation is used, and a symmetrical and state-dependent matrix solution is solved using a Riccati equation in the form of (26). Second, the proposed design should provide a measure of how close the proposed control is to the optimal control that must be computed offline. To this end, a condition will be developed along the lines of optimality condition (28).

The proposed suboptimal control is designed by the following steps:

\textbf{Step 1.} Given \( x \in \mathbb{R}^{2n} \), determine a symmetrical solution \( P_s(x) \in \mathbb{R}^{2n \times 2n} \) from the state-dependent Riccati equation:

\[
P_s^T(x)A(x) + A^T(x)P_s(x) + Q(x) \\
- P_s(x)BR^{-1}(x)B^T P_s(x) \\
= 0 \tag{29}
\]

Using identity (29), partial derivatives \( \frac{\partial P_s(x)}{\partial x_i} \) can be solved using the following state-dependent Lyapunov equations: for \( i = 1, \ldots, 2n \):

\[
[A - BR^{-1}B^T P_s] \frac{\partial P_s}{\partial x_i} + \frac{\partial P_s}{\partial x_i} [A - BR^{-1}B^T P_s] \\
= - P_s \frac{\partial A}{\partial x_i} - \frac{\partial A^T}{\partial x_i} P_s - \frac{\partial Q}{\partial x_i} P_s - P_s BR^{-1} \frac{\partial R}{\partial x_i} R^{-1} B^T P_s \tag{30}
\]

\textbf{Step 2.} Let the real-time suboptimal control be:

\[
u(t) = -R^{-1}(x)BTP_s(x)x + v(t) \tag{31}
\]

where \( P_s(x) \) is the solution to (29) and \( v(t) \in \mathbb{R}^n \) is an incremental control component to be determined.

\textbf{Step 3.} According to the optimality condition (28), formulate the following error of using \( u \) in (31) to approximate the optimal control:

\[
E(x,v) \triangleq \text{vec} \left\{ x^T \frac{\partial P_s}{\partial x_i} \right\}^T [Az - BR^{-1}B^T P_s x + Bu(t)] \\
+ 2P_s B v(t) + \frac{1}{2} \frac{\partial r(x)}{\partial x} x^T P_sBR^{-1}B^T P_s x \\
+ \frac{1}{2} \frac{\partial q(x)}{\partial x} x^T Q(x)x + \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x_i} P_s x \right\} \tag{32}
\]

\textbf{Step 4.} Let matrix \( C(x) \) be:

\[
C(x) = \text{vec} \left\{ x^T \frac{\partial P_s}{\partial x_i} \right\}^T B + 2P_s B
\]

Then, find pseudo-inverse \( C^+(x) \) of matrix \( C(x) \), and calculate the incremental control \( v(t) \) using:

\[
v(t) = -C^+(x) [\text{vec} x^T \frac{\partial P_s}{\partial x_i} [Az - BR^{-1}B^T P_s x]] \\
+ \frac{1}{2} \frac{\partial r(x)}{\partial x} x^T P_s BR^{-1}B^T P_s x + \frac{1}{2} \frac{\partial q(x)}{\partial x} x^T Q(x)x \\
+ \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x_i} P_s x \right\} \tag{33}
\]

Stability and performance of robotic systems under control (31) are stated in the following theorem, the main result of this article.

\textbf{Lemma 2} [15]. Around the origin, system (20) can be approximated by its linearized version:

\[
\dot{x} = A(0)x + Bu \tag{34}
\]
Consider a space robotic system whose dynamics has been rewritten to be of form (20). Let:

\[ q(x) = r(x) = x^T W x \]  

(35)

(22) for some constant and positive-definite matrix \( W \) and let \( P_{s_0}(x) \) be the solution to state-dependent Riccati equation:

\[
P_{s_0}(x)A(x) + A^T(x)P_{s_0}(x) + Q_0 - P_{s_0}(x)BR_0^{-1}B^T P_{s_0}(x) = 0
\]

Then, control (31) together with (29), (30), and (31) near optimal in the following sense:

- It is locally optimal around the origin.
- Away from the origin, its closeness to the optimal control can be estimated by \( \|E(x,v)\| \) defined by (33).
- If \( E(x,v) = 0 \), the closed-loop system is globally asymptotically stable provided that matrix \( W \) is chosen such that

\[
1 + x^T W x > \frac{\lambda_{\min}(Q_0)}{\lambda_{\max}(P_{s_0}(x))} \left\| \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x} \right\} \right\|
\]

(36)

- If \( E(x,v) \neq 0 \), the closed-loop system remains globally asymptotically stable provided that \( E(x,v) \) is not very large in the sense that the sum:

\[
-(1 + x^T W x)x^T (Q + P_s B R_0^{-1} B^T P_s)x
\]

\[
= -x^T \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x} \right\} P_s x + x^T E(x,v) < 0
\]

(37)

that is, is negative-definite.

**Proof.** It is straightforward to show that the proposed control reduces to the standard linear optimal control around the origin. Thus, local optimality can be concluded stated in Lemma 2. Away from the origin, \( E(x,v) \) can be used to estimate the approximation error, as it is a diffeomorphism of optimality condition (24).

To show global stability, consider Lyapunov function \( V = x^T P_\tau x + 2x^T P_s x \), where \( P_\tau(x) \) is the solution to (29). Cause the pair \( \{A(x), B\} \) is pointwise controllable for \( x \), matrix solution \( P_\tau(x) \) is positive-definite. Note that, in solution \( P_{s_0}(x) \) (in the statement of the theorem), solution to (29) is simply:

\[
P_s(x) = e^{x^T W x} P_{s_0}(x)
\]

follows from (20), (29), and (32) that:

\[
\dot{V} = x^T \dot{P}_\tau x + 2x^T P_s x
\]

\[
= x^T \left[ \{P_\tau + P_s A + A^T P_\tau - 2P_s B R^{-1} B^T P_\tau \}x + 2P_s B v(t) \right]
\]

\[
= -(1 + x^T W x)x^T Q x
\]

\[
-(1 + x^T W x)x^T P_s B R^{-1} B^T P_s x
\]

\[-x^T \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x} \right\} P_s x + x^T E(x,v)\]

Inequality (36) implies that:

\[
(1 + x^T W x)x^T Q x > \left\| \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x} \right\} P_s x \right\|^2
\]

Therefore, stability results can be claimed under either (36) or (37) as \( \dot{V} < 0 \).

Several remarks are worth mentioning here.

**Remark 3.1.** If \( v(t) \) is set to be zero, the proposed control reduces to the existing results of state-dependent algebraic Riccati equation (SDARE) designs [7, 7, 17]. It follows from (33) that, around a small neighbourhood around the origin, \( v(t) = 0 \).

**Remark 3.2.** Control component \( v(t) \) is selected to approximate \( \{P^*(x) - P_s(x)\} \) where \( P^*(x) \) is the optimal solution and \( P_s(x) \) is the solution to (29). It is obvious that, using the operation of generalized inverse, the choice of \( v(t) \) minimizes \( \|E(x,v)\|^2 \). If \( C^T(x)C(x) \) is invertible, then the pseudoinverse is:

\[
C^+(x) = [C^T(x)C(x)]^{-1} C^T(x)
\]

Therefore, the resulting value of \( E(x,v) \) can be viewed as an approximate measure on the closeness between the optimal control and the proposed suboptimal control. On the other hand, inequality (37) (together with (36)) shows that minimizing \( E(x,v) \) is also desirable for achieving global stability.

**Remark 3.3.** It can be assumed without loss of any generality that:

\[
P(x) = P_s(x) + P_u(x)
\]

(38)

where \( P_u(x) \) is the solution to (29), and \( P_u(x) \), denoting the unsymmetrical part of \( P(x) \), would be of form:

\[
P_u(x) = \begin{bmatrix}
0 & \phi_2(x) & \cdots & \phi_{2n-1}(x) \\
0 & 0 & \cdots & \phi_{4n-2}(x) \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & 0
\end{bmatrix}
\]

In this case, the optimal choice of \( v(t) \) would be \( v(t) = -R^{-1}(x)B^TP_u(x)x \), and optimality condition (24) or (28) for determining \( P_u(x) \) would become:
\[ 0 = \sum_{i=1}^{2n} \frac{\partial P_s}{\partial x_i} \dot{x}_i + \dot{P}_u x + [P_u A + A^T P_u - P_u^T B R^{-1} B^T P_u \]
\[ \quad - P_s B R^{-1} B^T P_u - P_s^T B R^{-1} B^T P_s]x + \frac{1}{2} \frac{\partial q(x)}{\partial x} x^T Q x \]
\[ \quad + \frac{1}{2} \frac{\partial r(x)}{\partial x} x^T [P_s + P_s^T] B R^{-1} B^T [P_s + P_u] x \]
\[ \quad + \text{vec} \left\{ x^T \frac{\partial A^T}{\partial x_i} [P_s + P_u] x \right\} \]  
(39)

where \( \dot{x}_i \) is determined by system (20) under control (31). In (39), differential equations are implicit, and even if these equations are rewritten to be explicit and are integrated forward in time by using initial conditions only, properties of the solution and its numerical stability are too difficult to analyze or predict. In comparison, the proposed control design is not to achieve optimality by imposing (39), but rather to minimize the approximation error between the optimal control and that with symmetrical solution \( P_s(x) \). It appears from (39) that the choice of minimizing \( ||E(x, v)||^2 \) (from (32)) is a very good and attainable compromise among optimality, real-time implementation, and closed-loop stability.

\[ \text{Remark 3.4.} \text{ For robotic systems, nonlinear dynamics are at most second order, and hence partial derivatives } \frac{\partial A(x)}{\partial x_i} \text{ are uniformly bounded. Therefore, inequality (36) can always be satisfied through the choice of } W. \text{ Nonetheless, the basic idea is applicable to more general nonlinear systems [20 – 23].} \]

4. Illustrative Example

Dynamics of the two-link revolute-joint manipulator are given by:

\[ \tau_1 = d_{11} \ddot{q}_1 + d_{12} \ddot{q}_2 - 2m_2 L_1 l_2 \sin(q_2) q_1 \dot{q}_2 \]
\[ -m_2 L_1 l_2 \sin(q_2) \dot{q}_2^2 + (m_1 + m_2) l_1 \cos(q_1) \]
\[ + m_2 l_2 g \cos(q_1 + q_2) + f_1(\dot{q}_1) \]
\[ \tau_2 = d_{21} \ddot{q}_1 + d_{22} \ddot{q}_2 + m_2 L_1 l_2 \sin(q_2) \dot{q}_1^2 \]
\[ + m_2 l_2 g \cos(q_1 + q_2) + f_2(\dot{q}_2) \]  
(40)

where:

\[ d_{11} = m_1 l_1^2 + m_2 (L_1^2 + l_2^2 + 2 L_1 l_2 \cos(q_2)) + I_1 + I_2, \]
\[ d_{12} = d_{21} = m_2 (L_1^2 + l_2^2 l_2 \cos(q_2)) + I_2 \]
\[ d_{22} = m_2 l_2^2 + I_2 \]

\( q_1 \) and \( q_2 \) are link angles, \( m_1 \) and \( m_2 \) are link masses, \( I_1 \) and \( I_2 \) are rotational inertias of the links, \( L_1 \) and \( L_2 \) are link lengths, \( l_1 \) and \( l_2 \) are the distances from joint axis to centre mass of two links, respectively, \( f_1 \) and \( f_2 \) are friction functions, and \( \tau_1 \) and \( \tau_2 \) are the output torques of motors reflected to the joint axes.

It is straightforward to formulate state-space model (6) from (40). Without loss of any generality, the desired trajectory is set to be \( q_d = 0 \). Therefore, for set point tracking of a space robot, \( \tau_d \) in (5) is zero as \( g = 0, \dot{q}_d = 0, \) and \( \ddot{q}_d = 0 \).

In the control design, performance index index (19) is set to be such that with (21), (22), (35), and \( W = 30 I_{4 \times 4} \) (matrices \( Q_0 \) and \( R_0 \) are to be chosen shortly). In the simulations, robot dynamics are set to be (for \( i = 1, 2 \)):

\[ m_i = 5, \quad L_i = 0.5, \quad l_i = 0.24, \quad I_i = \frac{1}{3} m_i L_i^2, \]

and \( f_i(\dot{q}_i) = -3 \dot{q}_i \)

To compare the proposed control (31) and the SDARE control (that is, control (31) with \( v(t) = 0 \)), the following eight cases were simulated with respect to initial conditions and weighting matrices:

- Case 1: Initial condition is \( x(0) = [q_1(0), \dot{q}_1(0), q_2(0), \dot{q}_2(0)] = [0.5, 0.5, 0, 0] \), and weighting matrices are \( Q_0 = 0.3 I_{4 \times 4} \) and \( R_0 = 6 I_{2 \times 2} \).
- Case 2: \( x(0) = [-0.5, 0.5, 0, 0] \), \( Q_0 = 0.3 I_{4 \times 4} \), and \( R_0 = 6 I_{2 \times 2} \).
- Case 3: \( x(0) = [0.5, 0.5, 0, 0] \), \( Q_0 = 6 I_{4 \times 4} \), and \( R_0 = 0.3 I_{2 \times 2} \).
- Case 4: \( x(0) = [-0.5, 0.5, 0, 0] \), \( Q_0 = 0.3 I_{4 \times 4} \), and \( R_0 = 6 I_{2 \times 2} \).
- Case 5: \( x(0) = [0.9, 0.8, 0, 0] \), \( Q_0 = 0.3 I_{4 \times 4} \), and \( R_0 = 6 I_{2 \times 2} \).
- Case 6: \( x(0) = [-0.9, 0.8, 0, 0] \), \( Q_0 = 0.3 I_{4 \times 4} \), and \( R_0 = 6 I_{2 \times 2} \).
- Case 7: \( x(0) = [0.9, 0.8, 0, 0] \), \( Q_0 = 6 I_{4 \times 4} \), and \( R_0 = 0.3 I_{2 \times 2} \).
- Case 8: \( x(0) = [-0.9, 0.8, 0, 0] \), \( Q_0 = 6 I_{4 \times 4} \), and \( R_0 = 0.3 I_{2 \times 2} \).

The simulation results are shown in Figs. 1 up to 8 (in which the performance index values under the proposed control and the SDARE control are denoted by the solid line and the dotted line, respectively); the corresponding state errors and controls for case 7 are shown in Figs. 9 to 11 (state errors and controls of the other cases are of the same nature and therefore omitted for brevity); and the comparative results are summarized in Table 1. In Table 1, \( J_{\text{near}} \) and \( J_{\text{sdare}} \) denote values of the performance index under the proposed control and SDARE control, respectively.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J_{\text{near}}/J_{\text{sdare}} )</td>
</tr>
<tr>
<td>( R_0 &gt; Q_0 )</td>
</tr>
<tr>
<td>case 2: 99.7%</td>
</tr>
<tr>
<td>( R_0 &lt; Q_0 )</td>
</tr>
<tr>
<td>case 4: 97.5%</td>
</tr>
</tbody>
</table>
It is clear from the table that, for initial conditions close to the origin, performance of the proposed control is comparable to that of SDARE control. This is expected as, around the origin, both controls are actually good approximations of the linear, locally optimal control. In the cases where initial conditions are not close to the origin, the proposed control has noticeably better performance when $R_0 < Q_0$. If $R_0 > Q_0$ (as in cases 5 and 6), a high penalty is placed on control effort, and calculation of the additive control component $v(t)$ to minimize $E(t)$ becomes less effective. Even so, the proposed control still has comparable performance. Simulations were also done for other initial conditions, and the same observations hold. In summary, the proposed control is computationally efficient and has either comparable or better performance than its predecessor, the SDARE control (which was developed for real-time implementation).

Figure 1. Performance indexes of Case 1.

Figure 2. Performance indexes of Case 2.

Figure 3. Performance indexes of Case 3.

Figure 4. Performance indexes of Case 4.

Figure 5. Performance indexes of Case 5.
Figure 6. Performance indexes of Case 6.

Figure 7. Performance indexes of Case 7.

Figure 9. Case 7: Joint position errors (solid – under the proposed control, dotted – under the SDARE control).

Figure 10. Case 7: Joint velocity errors (solid – under the proposed control, dotted – under the SDARE control).

Figure 11. Case 7: Proposed near-optimal control (solid) and SDARE control (dotted).
5. Conclusion
This Article proposes a near-optimal and real-time control design methodology for space robotic systems in order to minimize control energy. The main goal is to generate a globally stabilizing control law whose closeness to optimal control (which is not attainable online) can first be approximated and then minimized. Using the nonlinear matrix representation and state-dependent Riccati equation, such a control is found in closed form. Results on global stability and approximate optimality are obtained. An illustrative example is used to demonstrate its effectiveness.

References