Robotica (2019) volume 37, pp. 321–337. © Cambridge University Press 2018
doi:10.1017/S0263574718001030

Optimal sliding mode control design based on the state-dependent Riccati equation for cooperative manipulators to increase dynamic load carrying capacity

A. H. Korayem, S. R. Nekoo and M. H. Korayem*

Robotic Research Laboratory, Center of Excellence in Experimental Solid Mechanics and Dynamics, School of Mechanical Engineering, Iran University of Science and Technology, Tehran 1684613114, Iran. E-mails: habibnejad1371@yahoo.com, saerafee@yahoo.com

(Accepted September 17, 2018. First published online: October 9, 2018)

SUMMARY

Cooperative manipulators have uncertainties in their structure; therefore, an optimal sliding mode control method is derived from a combination of the sliding mode control (SMC) and the state-dependent Riccati equation (SDRE) technique. This proposed combination is applied to a class of non-linear closed-loop systems. One of the distinguished features of this control method is its robustness toward uncertainty. Due to the lack of optimality in the SMC method, in this paper, a robust and optimal method is presented by considering the SDRE in design of the sliding surface. Due to the fact that cooperative manipulators have been used for carrying loads, the percentage of load distributions between each manipulator has been derived to increase the dynamic load carrying capacity (DLCC). The proposed control structure is implemented on a Scout robot with two manipulators in cooperative mode, theoretically and practically using LabVIEW software; and the results were compared by considering the uncertainty in its structure. In comparison with the SDRE, the proposed method increased the DLCC almost 10% in the Scout case.

KEYWORDS: Optimal sliding mode controller, Algebraic sliding surface, Cooperative manipulators, DLCC, Load distribution

1. Introduction

Most mechanical systems, especially robotic ones, possess non-linear dynamics and various methods in non-linear control domain are available for their navigation. In order to control cooperative arms, inverse dynamic method was used by Kokkinis.[1] Joint connections of the two arms, were caused the excess constraint of the problem. This issue resolved with regarding the internal force in the dynamic equations. Yun *et al.* introduced an algorithm to control the cooperative arms by taking rotation and sliding constraints into account.[2] In that case, due to arm's dynamics, input–output linearization was considered. Moreover, Wen and Delgado, introduced a position and force control algorithm for cooperative arms which decouples the motion and force control problem.[3] Gao and Xiao used two cooperative arms in order to move massive objects by considering tracking tasks.[4] Moreover, Li proposed the formalism of cooperative manipulator's dynamic and optimal load distribution.[5] Optimization was carried out by minimizing of the arm's energy and finding a solution to Lagrange multipliers. Lin and Tsai provided a technique in order to apply the impedance control with two arms.[6] Also, the neural network compensation was considered to improve the behavior of controller with unknown parameters. Liu and Chen provided an algorithm for robust control of cooperative robots through decomposing of equations from constraints.[7]

A new method proposed to design the optimal path with two degrees of excess freedom for the cooperative rigid arm[8] and the arm with the flexible joints.[9] Subbarao *et al.* used the base model and

* Corresponding author: E-mail: hkorayem@iust.ac.ir

adaptive control method to control two cooperative arms forming a closed chain with consideration of the kinematic constraints.[10] Ge and Wang provided robust adaptive control for the cooperative mobile robots.[11] Moreover, Ghasemi and Keshmiri presented an algorithm for determining the load carrying capacity on the predefined path, by separating controller methods into two distinct systems.[12] Yagiz *et al.* considered sliding mode control (SMC) method in order to carry loads by two arms.[13] As far as dynamic load carrying capacity (DLCC) is concerned, Korayem *et al.* calculated the amount of DLCC by using SMC and adding disturbances to a cable-suspended robot.[14] Chinelato *et al.* investigated the control of two cooperative mobile manipulators for carrying the payload.[15]

On the other hand, there exist various methods to control dynamical systems. Suzuki *et al.* designed a controller for an inverted pendulum and used the state-dependent Riccati equation (SDRE) method to solve the Hamilton–Jacobi–Bellman equation.[16] Moreover, they completed the control relationships by defining a different sliding surface which was based on the theory of variable structure control. The control structure considering the effects of disturbance and noise on the governing equations was also proposed.[17] Due to the fact that there are different models to define the sliding surface, Korayem *et al.* proposed a robust and suboptimal method by considering the SDRE in design of the sliding surface in two types of algebraic and integral sliding surfaces.[18] Moreover, due to the use of the state-dependent differential Riccati equation (SDDRE) in the integral form of the sliding surface, their method is able to provide a robust attitude with the desired finite time control option.

In this paper, the extracted optimality condition from the SDRE is inserted into the sliding surface equation which generates a robust controller with an optimality condition. Moreover, by considering the cost function, which experienced the effects of the applied load, the optimal distribution of the load is investigated. The main contribution of this paper is to present a robust and optimal control method by applying optimality conditions of the SDRE within the sliding surface. According to the defined weighting matrix, the percentage of load distribution can be changed for any manipulator. In Section 2, the optimal sliding mode control (OSMC) structure is expressed. Moreover, Section 3 presents governing relations on the mechanical cooperative arms and section 4 expresses the implementation. Section 5 includes the experimental results and Section 6 discusses the DLCC.

2. Structure of Optimal Sliding Mode Control

In this section, the OSMC is introduced with the aim of the SDDRE and the stability of the method is investigated using the Lyapunov approach. Consider the non-linear time-varying affine system:

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

where vectors $\mathbf{f}(\mathbf{x}(t), t)$ and $\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t)$ in Eq. (1) are presenting actual system and the exact values of them are not available, however, they are bounded and the bound values are known. The next step is to extract the state-dependent coefficient (SDC) parameterization of the system (1) from the nominal system as represented in Eq. (2).

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \hat{\mathbf{f}}(\mathbf{x}(t), t) + \hat{\mathbf{g}}(\mathbf{x}(t), \mathbf{u}(t), t) = \hat{\mathbf{A}}(\mathbf{x}(t), t)\mathbf{x}(t) + \hat{\mathbf{B}}(\mathbf{x}(t), t)\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}(\mathbf{x}(t), t)\mathbf{x}(t), \end{aligned} \tag{2}$$

where $\mathbf{x}(t) \in \Re^n$ is the state variable vector, $\mathbf{y}(t) \in \Re^l$ is the output vector and $\mathbf{u}(t) \in \Re^m$ is the control input vector. $\hat{\mathbf{A}}(\mathbf{x}(t), t) : \Re^n \times \Re^+ \to \Re^{n\times n}$, $\hat{\mathbf{B}}(\mathbf{x}(t), t) : \Re^n \times \Re^+ \to \Re^{n\times m}$ and $\mathbf{C}(\mathbf{x}(t), t) : \Re^l \times \Re^+ \to \Re^{l\times n}$ are called system input and output matrix, respectively. The "$\wedge$" indicate the nominal value of the parameter. $\hat{\mathbf{A}}(\mathbf{x}(t), t)$ and $\hat{\mathbf{B}}(\mathbf{x}(t), t)$ are expressing the nominal SDC parameterizations of the system (2) based on the nominal parameters for control design. Generally, changing system (1) to (2) is called SDC parameterization or apparent linearization.[19] This step of design is not unique and may lead to the different parameterization; however, the concept of this method is on factorization of a vector from a non-linear one to shape multiplication of a matrix to a vector. For the case of robotic manipulators, a structured systematic way to make SDC parameterization is proposed and the controllability of the SDC matrices is investigated.[20] To generate $\hat{\mathbf{A}}(\mathbf{x}(t), t)$ and $\hat{\mathbf{B}}(\mathbf{x}(t), t)$, the dynamic of the system must be expressed with the nominal values of the robot.

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

It is assumed that system (1) satisfies the Lipschitz condition and is piecewise continuous. Due to the control system requirements, vectors $\mathbf{g}(\mathbf{0}, \mathbf{0}, t)$ and $\mathbf{f}(\mathbf{0}, t)$ are uniformly bounded in $t \in [0, t_f]$. Moreover, the matrix $\hat{\mathbf{B}}(\mathbf{x}(t), t)$ should be at least one time differentiable and the first derivative must be non-singular. Moreover, it is assumed that the pairs $\{\hat{\mathbf{A}}(\mathbf{x}(t), t), \hat{\mathbf{B}}(\mathbf{x}(t), t)\}$ and $\{\hat{\mathbf{A}}(\mathbf{x}(t), t), C(\mathbf{x}(t), t)\}$ in system (2) are the controllable and the observable parameterizations of the system.

Now, consider the *algebraic sliding surface* as in the form of

$$\mathbf{s}(\mathbf{x}(t), t) = \hat{\mathbf{B}}^T(\mathbf{x}(t), t)\hat{\mathbf{K}}(\mathbf{x}(t), t)\tilde{\mathbf{x}}(t), \tag{3}$$

where $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_{\text{des}}(t)$ is the error of state vector in which $\mathbf{x}_{\text{des}}(t)$ is the desired value for the states and $\hat{\mathbf{K}}(\mathbf{x}(t), t)$ which is the suboptimal symmetric gain, is the solution of the SDDRE:[20]

$$J = \frac{1}{2}\left(\mathbf{x}^T(t_f)\mathbf{F}\mathbf{x}(t_f) + \int_0^{t_f} \mathbf{x}^T(t)\mathbf{Q}(\mathbf{x}(t), t)\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}(\mathbf{x}(t), t)\mathbf{u}(t)\mathrm{d}t\right), \tag{4}$$

$$-\dot{\hat{\mathbf{K}}} = \hat{\mathbf{A}}^T\hat{\mathbf{K}} + \hat{\mathbf{K}}\hat{\mathbf{A}} - \hat{\mathbf{K}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}} + \mathbf{Q}, \tag{5}$$

with the final boundary condition $\hat{\mathbf{K}}(\mathbf{x}(t_f), t_f) = \mathbf{F}$, regarding the cost function which is represented in (4) where $\mathbf{Q} \in \Re^{n\times n}$ is the weighting matrix for the state variables (in $t \in [0, t_f)$) which is symmetric positive semi-definite, and $\mathbf{R} \in \Re^{m\times m}$ is the weighting matrix for inputs which is symmetric positive definite.

Instead of solving the SDDRE (5) directly, the Lyapunov-based method is considered. The first step is to solve the steady-state form of (5) to reach[20]

$$\hat{\mathbf{A}}^T\hat{\mathbf{K}}_{\text{ss}} + \hat{\mathbf{K}}_{\text{ss}}\hat{\mathbf{A}} - \hat{\mathbf{K}}_{\text{ss}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}}_{\text{ss}} + \mathbf{Q} = \mathbf{0}. \tag{6}$$

When $t \to \infty$, $\hat{\mathbf{K}}(\mathbf{x}(t), t)$ is represented by $\hat{\mathbf{K}}_{ss}(\mathbf{x}(t), t)$. By subtracting (6) from (5),

$$-\dot{\hat{\mathbf{K}}} = \hat{\mathbf{A}}^T[\hat{\mathbf{K}} - \hat{\mathbf{K}}_{\text{ss}}] + [\hat{\mathbf{K}} - \hat{\mathbf{K}}_{\text{ss}}]\hat{\mathbf{A}} - \hat{\mathbf{K}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}} + \hat{\mathbf{K}}_{\text{ss}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}}_{\text{ss}}, \tag{7}$$

and introducing

$$\begin{aligned}&\hat{\mathbf{P}}^{-1}(\mathbf{x}(t), t) = \hat{\mathbf{K}}(\mathbf{x}(t), t) - \hat{\mathbf{K}}_{\text{ss}}(\mathbf{x}(t), t),\\ &\hat{\mathbf{A}}_{\text{cl}}(\mathbf{x}(t)) = \hat{\mathbf{A}}(\mathbf{x}(t), t) - \hat{\mathbf{B}}(\mathbf{x}(t), t)\mathbf{R}^{-1}\hat{\mathbf{B}}^T(\mathbf{x}(t), t)\hat{\mathbf{K}}_{\text{ss}}(\mathbf{x}(t), t),\end{aligned} \tag{8}$$

a state-dependent differential Lyapunov equation results in

$$\dot{\hat{\mathbf{P}}}(\mathbf{x}(t), t) = \hat{\mathbf{A}}_{\text{cl}}(\mathbf{x}(t), t)\hat{\mathbf{P}}(\mathbf{x}(t), t) + \hat{\mathbf{P}}(\mathbf{x}(t), t)\hat{\mathbf{A}}_{\text{cl}}^T(\mathbf{x}(t), t) - \hat{\mathbf{B}}(\mathbf{x}(t), t)\mathbf{R}^{-1}\hat{\mathbf{B}}^T(\mathbf{x}(t), t), \tag{9}$$

with the final boundary condition $\hat{\mathbf{P}}(\mathbf{x}(t_f), t_f) = [\mathbf{F} - \hat{\mathbf{K}}_{\text{ss}}(\mathbf{x}(t), t)]^{-1}$. Then, the closed-form answer to (9) is presented as[20]

$$\hat{\mathbf{P}}(\mathbf{x}(t), t) = \hat{\mathbf{E}}(\mathbf{x}(t), t) + \mathbf{e}^{\hat{\mathbf{A}}_{\text{cl}}(\mathbf{x}(t), t)(t-t_f)}(\hat{\mathbf{P}}(\mathbf{x}(t_f), t_f) - \hat{\mathbf{E}}(\mathbf{x}(t), t))\mathbf{e}^{\hat{\mathbf{A}}_{\text{cl}}^T(\mathbf{x}(t), t)(t-t_f)}, \tag{10}$$

where the solution $\hat{\mathbf{E}}(\mathbf{x}(t), t)$ is to be found by a state-dependent algebraic Lyapunov equation:

$$\hat{\mathbf{A}}_{\text{cl}}(\mathbf{x}(t), t)\hat{\mathbf{E}}(\mathbf{x}(t), t) + \hat{\mathbf{E}}(\mathbf{x}(t), t)\hat{\mathbf{A}}_{\text{cl}}^T(\mathbf{x}(t), t) - \hat{\mathbf{B}}(\mathbf{x}(t), t)\mathbf{R}^{-1}\hat{\mathbf{B}}^T(\mathbf{x}(t), t) = \mathbf{0}. \tag{11}$$

Finally, the suboptimal gain of the SDDRE is obtained as

$$\hat{\mathbf{K}}(\mathbf{x}(t), t) = \hat{\mathbf{K}}_{\text{SS}}(\mathbf{x}(t), t) + \hat{\mathbf{P}}^{-1}(\mathbf{x}(t), t). \tag{12}$$

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

The equivalent control law is obtained by solving $\dot{\mathbf{s}}(\mathbf{x}(t), t) = \mathbf{0}$ for $\mathbf{u}(t)$:

$$\dot{\mathbf{s}} = \dot{\hat{\mathbf{B}}}^T \hat{\mathbf{K}}\tilde{\mathbf{x}} + \hat{\mathbf{B}}^T \dot{\hat{\mathbf{K}}}\tilde{\mathbf{x}} + \hat{\mathbf{B}}^T \hat{\mathbf{K}}[\hat{\mathbf{f}} + \hat{\mathbf{B}}\mathbf{u}] - \hat{\mathbf{B}}^T \hat{\mathbf{K}}\dot{\mathbf{x}}_{\text{des}} = \mathbf{0}, \tag{13}$$

which results in an equivalent control law as shown in (14):

$$\mathbf{u}_{\text{n}} = [\hat{\mathbf{B}}^T \hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}[\hat{\mathbf{B}}^T \hat{\mathbf{K}}\dot{\mathbf{x}}_{\text{des}} - \dot{\hat{\mathbf{B}}}^T \hat{\mathbf{K}}\tilde{\mathbf{x}} - \hat{\mathbf{B}}^T \dot{\hat{\mathbf{K}}}\tilde{\mathbf{x}} - \hat{\mathbf{B}}^T \hat{\mathbf{K}}\hat{\mathbf{f}}]. \tag{14}$$

Now, by considering a correction control law as in the form of

$$\mathbf{u}_{\text{corr}}(t) = -[\hat{\mathbf{B}}^T(\mathbf{x}(t), t)\hat{\mathbf{K}}(\mathbf{x}(t), t)\hat{\mathbf{B}}(\mathbf{x}(t), t)]^{-1}\mathbf{K}_{\text{corr}}(\mathbf{x}(t), t)\text{sgn}(\mathbf{s}(\mathbf{x}(t), t)), \tag{15}$$

where $\mathbf{K}_{\text{corr}}(\mathbf{x}(t), t) : \Re^n \times \Re^+ \to \Re^{m\times m}$. The final control law is obtained as (16)

$$\begin{aligned}\mathbf{u}(t) = \mathbf{u}_{\text{n}}(t) + \mathbf{u}_{\text{corr}}(t) = [\hat{\mathbf{B}}^T \hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}[\hat{\mathbf{B}}^T \hat{\mathbf{K}}\dot{\mathbf{x}}_{\text{des}} - \dot{\hat{\mathbf{B}}}^T \hat{\mathbf{K}}\tilde{\mathbf{x}} \\ -\hat{\mathbf{B}}^T \dot{\hat{\mathbf{K}}}\tilde{\mathbf{x}} - \hat{\mathbf{B}}^T \hat{\mathbf{K}}\hat{\mathbf{f}}] - [\hat{\mathbf{B}}^T \hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}\mathbf{K}_{\text{corr}}\text{sgn}(\mathbf{s}).\end{aligned} \tag{16}$$

Substituting $\dot{\hat{\mathbf{K}}}(\mathbf{x}(t), t)$ results

$$\begin{aligned}\mathbf{u} = [\hat{\mathbf{B}}^T \hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}[\hat{\mathbf{B}}^T \hat{\mathbf{K}}\dot{\mathbf{x}}_{\text{des}} + \hat{\mathbf{B}}^T\{\hat{\mathbf{A}}^T\hat{\mathbf{K}} + \hat{\mathbf{K}}\hat{\mathbf{A}} - \hat{\mathbf{K}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}} + \mathbf{Q}\}\tilde{\mathbf{x}} \\ - \dot{\hat{\mathbf{B}}}^T \hat{\mathbf{K}}\tilde{\mathbf{x}} - \hat{\mathbf{B}}^T \hat{\mathbf{K}}\hat{\mathbf{f}}] - [\hat{\mathbf{B}}^T \hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}\mathbf{K}_{\text{corr}}\text{sgn}(\mathbf{s}).\end{aligned} \tag{17}$$

The bounds of the uncertainty for the system are considered as[21]

$$\begin{aligned}\left|\mathbf{f}(\mathbf{x}(t), t) - \hat{\mathbf{f}}(\mathbf{x}(t), t)\right| \le \mathbf{F}(\mathbf{x}(t), t), \\ \mathbf{B}(\mathbf{x}(t), t) = [\mathbf{I} + \boldsymbol{\Delta}]\,\hat{\mathbf{B}}(\mathbf{x}(t), t),\end{aligned} \tag{18}$$

$$\beta(\mathbf{x}(t), t) = \hat{\mathbf{B}}^T(\mathbf{x}(t), t)\hat{\mathbf{K}}(\mathbf{x}(t), t)\mathbf{B}(\mathbf{x}(t), t)[\hat{\mathbf{B}}^T(\mathbf{x}(t), t)\hat{\mathbf{K}}(\mathbf{x}(t), t)\hat{\mathbf{B}}(\mathbf{x}(t), t)]^{-1}, \tag{19}$$

where $\mathbf{F}(\mathbf{x}(t), t)$ represents deviation of the vector $\mathbf{f}(\mathbf{x}(t), t)$ from a nominal value $\hat{\mathbf{f}}(\mathbf{x}(t), t)$, $\boldsymbol{\Delta}$ is the deviation in $\hat{\mathbf{B}}(\mathbf{x}(t), t)$ matrix, represented as[21]

$$\left|\Delta_{ij}\right| \le D_{ij}, \tag{20}$$

$$\mathbf{B}_{\max}(\mathbf{x}(t), t) = [\mathbf{I} + \mathbf{D}]\,\hat{\mathbf{B}}(\mathbf{x}(t), t) \Rightarrow \mathbf{D} = \mathbf{B}_{\max}(\mathbf{x}(t), t)\hat{\mathbf{B}}^{-1}(\mathbf{x}(t), t) - \mathbf{I}, \tag{21}$$

where $\mathbf{D}$ is the maximum changes in any element of $\hat{\mathbf{B}}(\mathbf{x}(t), t)$ matrix. By considering the presented assumptions in (20) and (21), the modified form of (19) is

$$\beta(\mathbf{x}(t), t) = \hat{\mathbf{B}}^T(\mathbf{x}(t), t)\hat{\mathbf{K}}(\mathbf{x}(t), t)\,[\mathbf{I} + \mathbf{D}]\,\hat{\mathbf{B}}(\mathbf{x}(t), t)[\hat{\mathbf{B}}^T(\mathbf{x}(t), t)\hat{\mathbf{K}}(\mathbf{x}(t), t)\hat{\mathbf{B}}(\mathbf{x}(t), t)]^{-1}. \tag{22}$$

The non-linear time-varying affine system (1) and the nominal system (2) can be stabilized by control input (17) with the algebraic sliding surface (3) in which the non-linear correction gain of control law $\mathbf{K}_{\text{corr}}(\mathbf{x}(t), t)$ is to be found by

$$\mathbf{K}_{\text{corr}}(\mathbf{x}(t), t) \ge |\beta(\mathbf{x}(t), t)|^{-1}\,(\eta + |\mathbf{w}(\mathbf{x}(t), t)|)\,, \tag{23}$$

where $\boldsymbol{\eta}$ is a strict positive vector and

$$\begin{aligned}\mathbf{w} = \dot{\hat{\mathbf{B}}}^T \hat{\mathbf{K}}\tilde{\mathbf{x}} - \hat{\mathbf{B}}^T[\hat{\mathbf{A}}^T\hat{\mathbf{K}} + \hat{\mathbf{K}}\hat{\mathbf{A}} - \hat{\mathbf{K}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}} + \mathbf{Q}]\tilde{\mathbf{x}} + \hat{\mathbf{B}}^T\hat{\mathbf{K}}\mathbf{F} \\ - \hat{\mathbf{B}}^T\hat{\mathbf{K}}\mathbf{D}\hat{\mathbf{B}}[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{f}} - \hat{\mathbf{B}}^T\hat{\mathbf{K}}\dot{\mathbf{x}}_{\text{des}} + \hat{\mathbf{B}}^T\hat{\mathbf{K}}\mathbf{B}\{[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\dot{\mathbf{x}}_{\text{des}} \\ + \hat{\mathbf{B}}^T\{\hat{\mathbf{A}}^T\hat{\mathbf{K}} + \hat{\mathbf{K}}\hat{\mathbf{A}} - \hat{\mathbf{K}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}} + \mathbf{Q}\}\tilde{\mathbf{x}} - \dot{\hat{\mathbf{B}}}^T \hat{\mathbf{K}}\tilde{\mathbf{x}}]\}.\end{aligned} \tag{24}$$

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

Considering the Lyapunov candidate $V_i(\mathbf{x}(t), t) = \frac{1}{2}s_i^2(\mathbf{x}(t), t)$, for $i = 1, ..., m$, where m is the number of actuators in (2).

$$\dot{V}_i(\mathbf{x}(t), t) = s_i(\mathbf{x}(t), t)\dot{s}_i(\mathbf{x}(t), t) \le -\eta_i \left|s_i(\mathbf{x}(t), t)\right| . \tag{25}$$

The relation (25) must be held to prove the stability of the system (1) which provides

$$\mathbf{s}^T \{\dot{\hat{\mathbf{B}}}^T \hat{\mathbf{K}}\tilde{\mathbf{x}} + \hat{\mathbf{B}}^T \dot{\hat{\mathbf{K}}}\tilde{\mathbf{x}} + \hat{\mathbf{B}}^T \hat{\mathbf{K}}(\dot{\mathbf{x}} - \dot{\tilde{\mathbf{x}}}_{\mathrm{des}})\} \le -\eta^T \left|\mathbf{s}\right| . \tag{26}$$

In order to provide a robust input gain respect to uncertainty of the system, actual system (1) will be substituted in (23) to elicit the known bounds:

$$\mathbf{s}^T \{\dot{\hat{\mathbf{B}}}^T \hat{\mathbf{K}}\tilde{\mathbf{x}} + \hat{\mathbf{B}}^T \dot{\hat{\mathbf{K}}}\tilde{\mathbf{x}} + \hat{\mathbf{B}}^T \hat{\mathbf{K}}\mathbf{f} + \hat{\mathbf{B}}^T \hat{\mathbf{K}}\mathbf{B}\mathbf{u} - \hat{\mathbf{B}}^T \hat{\mathbf{K}}\dot{\mathbf{x}}_{\mathrm{des}}\} \le -\eta^T \left|\mathbf{s}\right| . \tag{27}$$

Substituting input (17) in (27) and rearrangement results in

$$\mathbf{s}^T(\mathbf{x}(t), t)\beta(\mathbf{x}(t), t)\mathbf{K}_{\mathrm{corr}}(\mathbf{x}(t), t)\mathrm{sgn}(\mathbf{s}(\mathbf{x}(t), t)) \ge \eta^T \left|\mathbf{s}(\mathbf{x}(t), t)\right| + \mathbf{s}^T(\mathbf{x}(t), t)\mathbf{w}(\mathbf{x}(t), t). \tag{28}$$

The scalar form of (28) could be represented as

$$\begin{aligned} &\sum_{i=1}^{m}\sum_{j=1}^{m} s_i(\mathbf{x}(t), t)\beta_{i,j}(\mathbf{x}(t), t)K_{\mathrm{corr},i,i}(\mathbf{x}(t), t)\mathrm{sgn}(s_i(\mathbf{x}(t), t)) \\ &\quad \ge \sum_{i=1}^{m} (\eta_i \left|s_i(\mathbf{x}(t), t)\right| + s_i(\mathbf{x}(t), t)w_i(\mathbf{x}(t), t)). \end{aligned} \tag{29}$$

Dividing (29) by $s_i(\mathbf{x}(t), t)$ and computing absolute value of (29) results in (30):

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \beta_{i,j}(\mathbf{x}(t), t)K_{\mathrm{corr},i,i}(\mathbf{x}(t), t) \ge \sum_{i=1}^{m} (\eta_i + \left|w_i(\mathbf{x}(t), t)\right|). \tag{30}$$

The matrix form of (30) eventually leads to

$$\mathbf{K}_{\mathrm{corr}}(\mathbf{x}(t), t) \ge \left|\beta(\mathbf{x}(t), t)\right|^{-1} (\eta + \left|\mathbf{w}(\mathbf{x}(t), t)\right|) , \tag{31}$$

in which $\mathbf{K}_{\mathrm{corr}}(\mathbf{x}(t), t) = \mathrm{diag}(K_{\mathrm{corr},1,1}(\mathbf{x}(t), t), ..., K_{\mathrm{corr,m,m}}(\mathbf{x}(t), t))$ is the non-linear positive diagonal gain of input (17) without any unknown parameters of the uncertain system.

OSMC, for control coefficient calculation in each loop, is very time consuming for simulation. This problem also exists in the implementation of this method. In general case for systems with complicated dynamic equations such as mechanical manipulators, exact solution of (12) cannot be found analytically. In this paper, the methodology of Taylor series approximation is used for finding numerical solution of (12). This method is also called power series approximation (PSA).[22–24] For finding the numerical solution to SDRE, consider a non-linear system (2). By rewriting $\hat{\mathbf{A}}(\mathbf{x}(t), t)$ in the following form:[23]

$$\hat{\mathbf{A}}(\mathbf{x}(t), t) = \hat{\mathbf{A}}_0 + \varepsilon\Delta\hat{\mathbf{A}}(\mathbf{x}(t), t). \tag{32}$$

And, representing X as a Taylor series

$$\hat{\mathbf{K}}_{\mathrm{ss}}(\mathbf{x}(t), \varepsilon) = \hat{\mathbf{K}}_{\mathrm{ss}}(\mathbf{x}(t), t)|_{\varepsilon=0}\varepsilon + \frac{\partial^2\hat{\mathbf{K}}_{\mathrm{ss}}(\mathbf{x}(t), t)}{\partial\varepsilon^2}|_{\varepsilon=0}\frac{\varepsilon^2}{2} + \cdots , \tag{33}$$

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

by substituting $\hat{\mathbf{A}}(\mathbf{x}(t), t)$, $\hat{\mathbf{K}}_{ss}(\mathbf{x}(t), t)$ into the SDRE equation, the result will be in the following form:[23]

$$\begin{gathered}(\hat{\mathbf{A}}_0 + \varepsilon\Delta\hat{\mathbf{A}})^T \left[\hat{\mathbf{K}}_{\text{ss}}|_{\varepsilon=0}\varepsilon + \tfrac{\partial^2\hat{\mathbf{K}}_{\text{ss}}}{\partial\varepsilon^2}|_{\varepsilon=0}\tfrac{\varepsilon^2}{2} + \cdots\right] + \left[\hat{\mathbf{K}}_{\text{ss}}|_{\varepsilon=0}\varepsilon + \tfrac{\partial^2\hat{\mathbf{K}}_{\text{ss}}}{\partial\varepsilon^2}|_{\varepsilon=0}\tfrac{\varepsilon^2}{2} + \cdots\right](\hat{\mathbf{A}}_0 + \varepsilon\Delta\hat{\mathbf{A}}) \\ - \left[\hat{\mathbf{K}}_{\text{ss}}|_{\varepsilon=0}\varepsilon + \tfrac{\partial^2\hat{\mathbf{K}}_{\text{ss}}}{\partial\varepsilon^2}|_{\varepsilon=0}\tfrac{\varepsilon^2}{2} + \cdots\right]\hat{\mathbf{B}}_0\mathbf{R}^{-1}\hat{\mathbf{B}}_0^T\left[\hat{\mathbf{K}}_{\text{ss}}|_{\varepsilon=0}\varepsilon + \tfrac{\partial^2\hat{\mathbf{K}}_{\text{ss}}}{\partial\varepsilon^2}|_{\varepsilon=0}\tfrac{\varepsilon^2}{2} + \cdots\right] + \mathbf{Q} = \mathbf{0}.\end{gathered} \tag{34}$$

By expanding this equation and collecting a similar power of ε, three iterative equations are generated: the first equation is an algebraic Riccati equation, the second and third are the state-dependent Lyapunov equations (please see the details in ref. [24]). These equations are simplified by substitution of $\Delta\hat{\mathbf{A}}(\mathbf{x}(t), t) = g(\mathbf{x}(t), t)\Delta\hat{\mathbf{A}}_0$ and result in

$$\hat{\mathbf{A}}_0^T\hat{\mathbf{K}}_{\text{ss},0} + \hat{\mathbf{K}}_{\text{ss},0}\hat{\mathbf{A}}_0 - \hat{\mathbf{K}}_{\text{ss},0}\hat{\mathbf{B}}_0\mathbf{R}^{-1}\hat{\mathbf{B}}_0^T\hat{\mathbf{K}}_{\text{ss},0} + \mathbf{Q} = \mathbf{0}, \tag{35}$$

$$\hat{\mathbf{K}}_{\text{ss},0}\boldsymbol{\Delta}\hat{\mathbf{A}} + \boldsymbol{\Delta}\hat{\mathbf{A}}^T\hat{\mathbf{K}}_{\text{ss},0} + \hat{\mathbf{K}}_{\text{ss},1}(\hat{\mathbf{A}}_0 - \hat{\mathbf{B}}_0\mathbf{R}^{-1}\hat{\mathbf{B}}_0^T\hat{\mathbf{K}}_{\text{ss},0}) + (\hat{\mathbf{A}}_0^T - \hat{\mathbf{K}}_{\text{ss},0}\hat{\mathbf{B}}_0\mathbf{R}^{-1}\hat{\mathbf{B}}_0^T)\hat{\mathbf{K}}_{\text{ss},1} = \mathbf{0}, \tag{36}$$

$$\begin{gathered}\hat{\mathbf{K}}_{\text{ss},n-1}\boldsymbol{\Delta}\hat{\mathbf{A}}_0 + \boldsymbol{\Delta}\hat{\mathbf{A}}_0^T\hat{\mathbf{K}}_{\text{ss},n-1} + \hat{\mathbf{K}}_{\text{ss},n}(\hat{\mathbf{A}}_0 - \hat{\mathbf{B}}_0\mathbf{R}^{-1}\hat{\mathbf{B}}_0^T\hat{\mathbf{K}}_{\text{ss},0}) + (\hat{\mathbf{A}}_0^T - \hat{\mathbf{K}}_{\text{ss},0}\hat{\mathbf{B}}_0\mathbf{R}^{-1}\hat{\mathbf{B}}_0^T)\hat{\mathbf{K}}_{\text{ss},n} \\ - \sum_{m=1}^{n-1}\hat{\mathbf{K}}_{\text{ss},m}\hat{\mathbf{B}}_0\mathbf{R}^{-1}\hat{\mathbf{B}}_0^T\hat{\mathbf{K}}_{\text{ss},n-m} = \mathbf{0}.\end{gathered} \tag{37}$$

Similarly, the state-feedback control gain is obtained

$$\hat{\mathbf{K}}_{\text{ss}}(\mathbf{x}(t), t) = \hat{\mathbf{K}}_{\text{ss},0} + \sum_{n=1} g^n(\mathbf{x}(t), t)\hat{\mathbf{K}}_{\text{ss},n}(\mathbf{x}(t), t). \tag{38}$$

By using the PSA after simulation, the input signal will be defined by an equation which is released from PSA and then the equation of input signal will implement on real robot.

3. Cooperative Arms

In this section, dynamic of cooperative arms is expressed. For this purpose, a comprehensive form is intended for cooperative arms. The intention of citing the comprehensive form is that n number of mechanical arms has being kept the mass m_p and are being moved on a path. Consider (39); this equation displays general dynamics equation of an arm.

$$\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{G}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i) = \mathbf{u}_i + \mathbf{J}_i^T(\mathbf{q}_i)\mathbf{f}_{e,i}, \tag{39}$$

where i index refers to the arm number, $\mathbf{M}_i(\mathbf{q}(t)) \in R^{n\times n}$ represents the inertia matrix and vector $\mathbf{C}_i(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \in R^{n\times 1}$ represents the forces of centrifugal and Coriolis force, $\mathbf{G}_i(\mathbf{q}(t)) \in R^{n\times 1}$ represents force of gravity, $\mathbf{b}_i$ is friction vector and $\mathbf{f}_{e,i}$ is external force vector exerting to arm i by the object. According to Newton and Euler's equation, $\mathbf{f}_e$ vector is obtained as[25]

$$\left[\begin{array}{c|c} m_p\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \\ --- & ---- \\ \mathbf{0}_{3\times 3} & [\mathbf{I}_p]_{3\times 3}\end{array}\right]\begin{bmatrix}\ddot{\mathbf{p}}_{\text{des}} \\ -- \\ \dot{\omega}_{\text{des}}\end{bmatrix} + \begin{bmatrix} m_p\mathbf{g} \\ ---------- \\ \omega_{\text{des}} \times (\mathbf{I}_p\omega_{\text{des}})\end{bmatrix} = -\sum f_{e,i} = -\mathbf{f}_e. \tag{40}$$

In (40), $I_{3\times 3}$ is identity matrix, I_p is object moment of inertia matrix, m_p is mass of the object, $\ddot{\mathbf{p}}_{\text{des}} = [\ddot{x}_{\text{des}}\ \ddot{y}_{\text{des}}\ \ddot{z}_{\text{des}}]^T$ considered linear acceleration in main axis direction, ω_{des} is angular velocity vector and $\dot{\omega}_{\text{des}}$ is angular acceleration vector. $\mathbf{f}_e$ is a vector which represents external forces as well as angular motion value, $H = I\omega$, respect to time. $\mathbf{b}_i$ is friction vector, and it can be calculated from (41):[22]

$$\mathbf{b}_i(\mathbf{q}_i(t)) = \mathbf{b}_i^V\dot{\mathbf{q}}_i(t) + \text{sgn}(\dot{\mathbf{q}}_i(t))\left[\mathbf{b}_i^d + \left(\mathbf{b}_i^S - \mathbf{b}_i^d\right)\exp\left(\frac{-\dot{\mathbf{q}}_i(t)}{\varepsilon}\right)\right], \tag{41}$$

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

where $\mathbf{b}_i^V$ is viscous friction, $\mathbf{b}_i^d$ is dynamic friction, $\mathbf{b}_i^S$ is static friction and ε is a small positive constant. In (41), only unknown parameter is how to divide $\mathbf{f}_e$ into $\mathbf{f}_{e,i}$ components. Now, by writing $\mathbf{f}_{e,i}$ equation in form of (42):

$$\mathbf{f}_{e,i} = \mathbf{J}_i^T(\mathbf{q}_i)^{-1}\left(\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{b}_i(\dot{q}_i) + \mathbf{G}_i(\mathbf{q}_i) - \mathbf{u}_i\right), \tag{42}$$

and using Lagrange multipliers optimization methods which is expressed in (43), the unknown parameter can be obtained.

$$L(\tau,\gamma) = \frac{1}{2}\sum \mathbf{u}_i^T\bar{\mathbf{Q}}_i\mathbf{u}_i + \gamma^T\left(\sum \mathbf{J}_i^T(\mathbf{q}_i)^{-1}\left(\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{G}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i) - \mathbf{u}_i\right) - \mathbf{f}_{e,i}\right). \tag{43}$$

The goal of optimization is distribution of load by constraint (42), into (43), that $\bar{\mathbf{Q}}_i$ is weighing matrix for load distribution among arms, and γ is Lagrange coefficient vector. Also, the weight matrix$\bar{\mathbf{Q}}_i$ consisting of all arms relations, which means load to be coupled optimality. For establishing optimal condition the derived (43) must be equal to zero; therefore, it can be written as

$$\frac{\partial L(\tau,\gamma)}{\partial \mathbf{u}_i} = \bar{\mathbf{Q}}_i\mathbf{u}_i - \left(\mathbf{J}_i^T(\mathbf{q}_i)^{-1}\right)^T\gamma = 0; \qquad \mathbf{u}_i = \bar{\mathbf{Q}}_i^{-1}\left(\mathbf{J}_i^T(\mathbf{q}_i)^{-1}\right)^T\gamma. \tag{44}$$

Now, by substituting $\mathbf{u}_i$ into (43), the amount γ is obtained:[25]

$$\gamma = \sum\left(\mathbf{J}_i^T(\mathbf{q}_i)^{-1}\bar{\mathbf{Q}}_i^{-1}\left(\mathbf{J}_i^T(\mathbf{q}_i)^{-1}\right)^T\right)^{-1}\left(\mathbf{J}_i^T(\mathbf{q}_i)^{-1}\left(\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{G}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i)\right) - \mathbf{f}_{e,i}\right). \tag{45}$$

And finally,

$$\mathbf{f}_{e,i} = \mathbf{J}_i^T(\mathbf{q}_i)^{-1}\left(\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{G}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i) - \bar{\mathbf{Q}}_i^{-1}\left(\mathbf{J}_i^T(\mathbf{q}_i)^{-1}\right)^T\gamma\right). \tag{46}$$

(46) represents the load distribution for each arm. It should be noted that, in order to calculate the load distribution, the end-effector's path should be identified and designated. In other words, the point to point motion in this section is not referred. (47) is written in state-space form and $\mathbf{x}(t) = [q_1(t) \ldots q_m(t)\ \dot{q}_1(t) \ldots \dot{q}_m(t)]^T$ is represented the state variables.

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{q}_1(t) \\ \vdots \\ \dot{q}_m(t) \\ \mathbf{M}_1^{-1}(\mathbf{x}(t))\left[\mathbf{u}_1(t) - \mathbf{C}_1(\mathbf{x}(t)) - \mathbf{G}_1(\mathbf{x}(t)) - \mathbf{b}_1(\mathbf{x}(t)) + \mathbf{J}_1^T(\mathbf{x}(t))f_{e,1}(t)\right] \\ \vdots \\ \mathbf{M}_m^{-1}(\mathbf{x}(t))\left[\mathbf{u}_m(t) - \mathbf{C}_m(\mathbf{x}(t)) - \mathbf{G}_m(\mathbf{x}(t)) - \mathbf{b}_m(\mathbf{x}(t)) + \mathbf{J}_m^T(\mathbf{x}(t))f_{e,m}(t)\right] \end{bmatrix}. \tag{47}$$

Parameterization of (47) should be taken into account to form the two $\mathbf{A}(\mathbf{x}(t))$ and $\mathbf{B}(\mathbf{x}(t))$ matrices and use them in control equations. The parameterization is for n robot with m degree of freedom (DoF):[25]

$$\mathbf{A}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{nm\times nm} & \mathbf{I}_{nm\times nm} \\ \mathbf{0}_{nm\times nm} & \mathbf{W}^r(\mathbf{x}(t)) \end{bmatrix}_{2nm\times 2nm}, \tag{48}$$

$$\mathbf{B}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{nm\times nm} \\ \mathbf{M}^{-1}_{nm\times nm}(\mathbf{x}(t)) \end{bmatrix}_{2nm\times nm}, \tag{49}$$

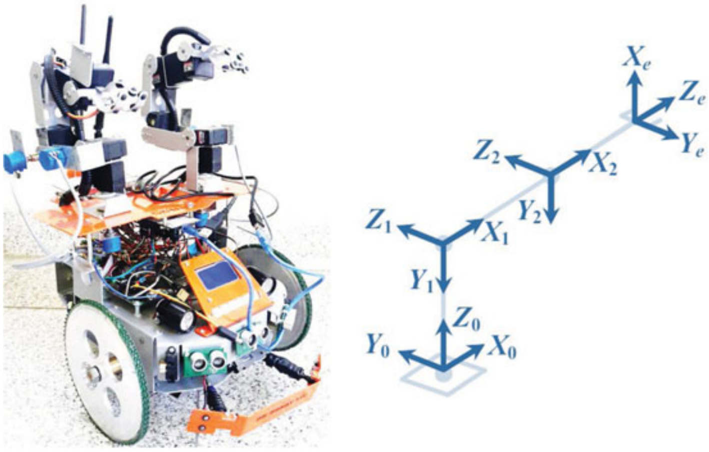


Fig. 1. Scout Robot and its schematic three-DoF modeling.

where $\mathbf{W}^r(\mathbf{x}(t))$ is defined as[25]

$$\mathbf{W}^r(\mathbf{q}(t)) = \begin{bmatrix} \mathbf{W}_1^r(\mathbf{x}(t)) & 0 & \cdots & 0 & 0 \\ 0 & \ddots & 0 & \vdots & 0 \\ \vdots & 0 & \mathbf{W}_i^r(\mathbf{x}(t)) & 0 & \vdots \\ 0 & \cdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{W}_m^r(\mathbf{x}(t)) \end{bmatrix}, \tag{50}$$

and $\mathbf{W}_i^r(\mathbf{x}(t)) = \mathbf{M}_i^{-1}(\mathbf{x}(t))[\bar{\mathbf{C}}_i(\mathbf{x}(t)) + \mathrm{diag}(b_{i,1}^V, ..., b_{i,n}^V)]$. Corrective control law term is defined in (51). (51) shows the corrective control input signal in matrix form for m cooperative arms.

$$\mathbf{u}^{\mathrm{add}}(t) = \begin{bmatrix} \mathbf{u}_1^{\mathrm{add}}(t) \\ \vdots \\ \mathbf{u}_m^{\mathrm{add}}(t) \end{bmatrix} \tag{51}$$

in which $\mathbf{u}_i^{\mathrm{add}}(t) = \mathbf{G}_i(\mathbf{x}(t)) - \mathbf{b}_i^{\mathrm{SDC}}(\mathbf{x}(t)) + \mathbf{J}_i^T(\mathbf{x}(t))\mathbf{f}_{e,i}(t)$.

4. Simulation and Experiment

4.1. Scout model

A Scout mobile robot has two mechanical arms mounted on its base. Each arm has five-DoF. In addition, the Scout robot uses three ultrasonic sensors in the front, a camera on the left arm wrist and six infrared sensors around to detect its surroundings. In Fig. 1, a view of the Scout robot is shown. The robot base has two main wheels and a DC motor is connected to each of them. The third wheel is also located at the end of the robot which is freewheeling to keep the balance. Moreover, all joints in the arms of the Scout robot are rotational.

Specifications of the Scout robot including mass and moment of inertia of the arms and base are expressed in Table I.

Due to the fact that the first link rotate around Z_0 axis aligned with Y_1 axis (Fig. 1), and based on Denavit–Hartenberg direct method, the I_{xx1} and I_{zz1} do not appear in the dynamic equation of motion and they are not relevant because they do not appear in equations. Therefore, their values assume to be constant. The inertia matrix assumed to be diagonal and the entries outside the main diagonal are neglected and assumed to be zero. Moreover, the neglected elements are so small by comparison with the main diagonal values. Since the angular velocity and torque in each of the motors are limited, the torque value of each motor has been considered as saturation function.[20] Via using the saturation

Table I. Scout properties[26].

Description	Values
Links mass (kg)	$m_1 = 0.028, m_2 = 0.1, m_3 = 0.231$
Moment of inertia of fist link (kg.m^2)	I_{xx1} = constant, $I_{yy1} = 0.0005$, I_{zz1} = constant
Moment of inertia of second link (kg.m^2)	$I_{xx2} = 0.000016, I_{yy2} = 0.000034, I_{zz2} = 0.000021$
Moment of inertia of third link (kg.m^2)	$I_{xx3} = 0.000923, I_{yy3} = 0.000915, I_{zz3} = 0.000085$

Table II. The amount of stop torque and no-load speed of Scout robot motors.

Value	Parameter	Description
0.55, 0.96, 0.96	$\tau_{\text{stall},l,1}, \tau_{\text{stall},l,2}, \tau_{\text{stall},l,3}$	Torque stall properties of arm's actuator (N.m)
5.85, 3, 3	$\omega_{nl,l,1}, \omega_{nl,l,2}, \omega_{nl,l,3}$	Angular velocity stall properties of arm's actuator (rad/s)
6.85	$\tau_{\text{stall},w}$	Torque stall properties of base's actuator (N.m)
14	$\omega_{nl,w}$	Angular velocity stall properties of base's actuator (rad/s)

Table III. Denavit–Hartenberg parameters of Scout robot.

Joint	$a_i(m)$	$d_i(m)$	α_i(deg)	θ_i
1	0	$d_1 = 0.06$	-90	θ_1
2	$a_2 = 0.1$	0	0	θ_2
3	$a_3 = 0.21$	0	0	θ_3

function, if in each joint the amount of calculated torque for the actuator is between the minimum and maximum torque value, the calculated control input value will apply to the actuators. But if its amount is less than lower bound or higher than upper bound, the control input value will be equal to the minimum and maximum value which is calculated by (52), respectively.

$$\begin{aligned}\tau_{i,\max}(t) &= \tau_{i,\text{stall}} - \tfrac{\tau_{i,\text{stall}}}{\omega_{i,nl}}\omega_i(t),\\ \tau_{i,\max}(t) &= -\tau_{i,\text{stall}} - \tfrac{\tau_{i,\text{stall}}}{\omega_{i,nl}}\omega_i(t).\end{aligned} \tag{52}$$

Parameters $\tau_{i,\min}(t)$ and $\tau_{i,\max}(t)$ show the minimum and maximum torque value applied to each joint, respectively. Moreover, $\tau_{i,\text{stall}}$ and $\omega_{i,nl}$ are, respectively, stall torque and the angular velocity of the motor i in no-load speed that their amount for arm motors and base motors is expressed in Table II.

After introducing the Scout robot, its dynamics equations will be expressed. In Table III, the value of its Denavit–Hartenberg parameters has been reported. In this experiment, the first three-DOF of each arm have been considered because the feedback for control processes is needed and the potentiometers are available only for the first three-DoF (set-up limitation). By having Denavit–Hartenberg parameters, direct and inverse kinematic of the Scout robot can be achieved and both of them are expressed in reference.[26]

After determining the coordinates of the arms and Denavit–Hartenberg parameters, transformation matrices can be calculated. Therefore, in this part, first, the existing constraints on mobile robot are explained and then by considering the constraints of the system, the dynamic equation of the Scout robot is simplified using the method described in ref. [21]. Because in this paper, robotic arms are more important, Scout robot is considered fix in base mode. Considering the generalized coordinates as (53)

$$q^T = \left[\theta_{1r}\ \theta_{2r}\ \theta_{3r}\ \theta_{1l}\ \theta_{1l}\ \theta_{1l}\right]. \tag{53}$$

And also to show the equations presented in the previous section, state variables will be considered as (54)

$$\mathbf{x}(t) = \begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix}. \tag{54}$$

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

Table IV. Control gain values.

Parameters	R	Q	$\bar{Q}_r$	$\bar{Q}_r$
Values	$0.01\ \mathrm{I}_{6\times 6}$	$100\ \mathrm{I}_{12\times 12}$	$\mathrm{I}_{6\times 6}$	$\mathrm{I}_{6\times 6}$

Finally, dynamic equations of the system can be considered as (55)

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \mathbf{0}_{6\times 6} & \mathbf{I}_{6\times 6} \\ \mathbf{0}_{6\times 6} & -\mathbf{M}_{\mathrm{T}}^{-1}(\mathbf{q}(t)) \times \bar{\mathbf{C}}_{\mathrm{T}}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \mathbf{0}_{6\times 6} \\ \mathbf{M}_{\mathbf{T}}^{-1}(\mathbf{q}(t)) \end{bmatrix} \mathbf{u}(t) \tag{55}$$

in which

$$\mathbf{M}_T(\mathbf{q}(t)) = \begin{bmatrix} \mathbf{M}_r(\mathbf{q}(t)) & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{M}_l(\mathbf{q}(t)) \end{bmatrix}, \tag{56}$$

$$\bar{\mathbf{C}}_T(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = \begin{bmatrix} \bar{\mathbf{C}}_r(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \\ \bar{\mathbf{C}}_l(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \end{bmatrix}. \tag{57}$$

Moreover, as it has been mentioned, the vector $\mathbf{u}(t)$ is considered as

$$\mathbf{u}(t) = \begin{bmatrix} \mathbf{u}_r^{\mathrm{add}}(t) \\ \mathbf{u}_l^{\mathrm{add}}(t) \end{bmatrix}_{6\times 1}. \tag{58}$$

4.2. Experimental setup of Scout

In order to simulation as well as implementation of the complex non-linear control relationships, it is needed to establish a relationship between robot and computing system; therefore, in this paper, the LabVIEW software is considered to establish the communication between the robot and computer. In this software, feedbacks are calculated for each motor individually and the presented control method is described and implemented through a MATLAB port environment inside the LabVIEW software. It should be noted that in the embedded MATLAB environment in the LabVIEW software, control equations are calculated as a chain and correspondingly with control equation modes for each motor and chain transfer is applied to Arduino Due and ultimately to the motor. In this simulation Latin z-shaped path (the z-shaped path has been selected because in this special path there are two critical points where in them the robot should stop and change the direction of each manipulator) and 6 s run time is considered with $[\,0.05\ -0.05\ 0.3\,]$ initial point. There is a basket which is connected by two ropes to the Scout's arms and the loads which are several blocks by the mass of 100 g, have been added to the basket which is held by Scout arm's gripper step by step in order to compute the DLCC. In this experiment, the load's momentum of inertia neglected and the load considered as a spot load. Moreover, the controller constants are represented in Table IV.

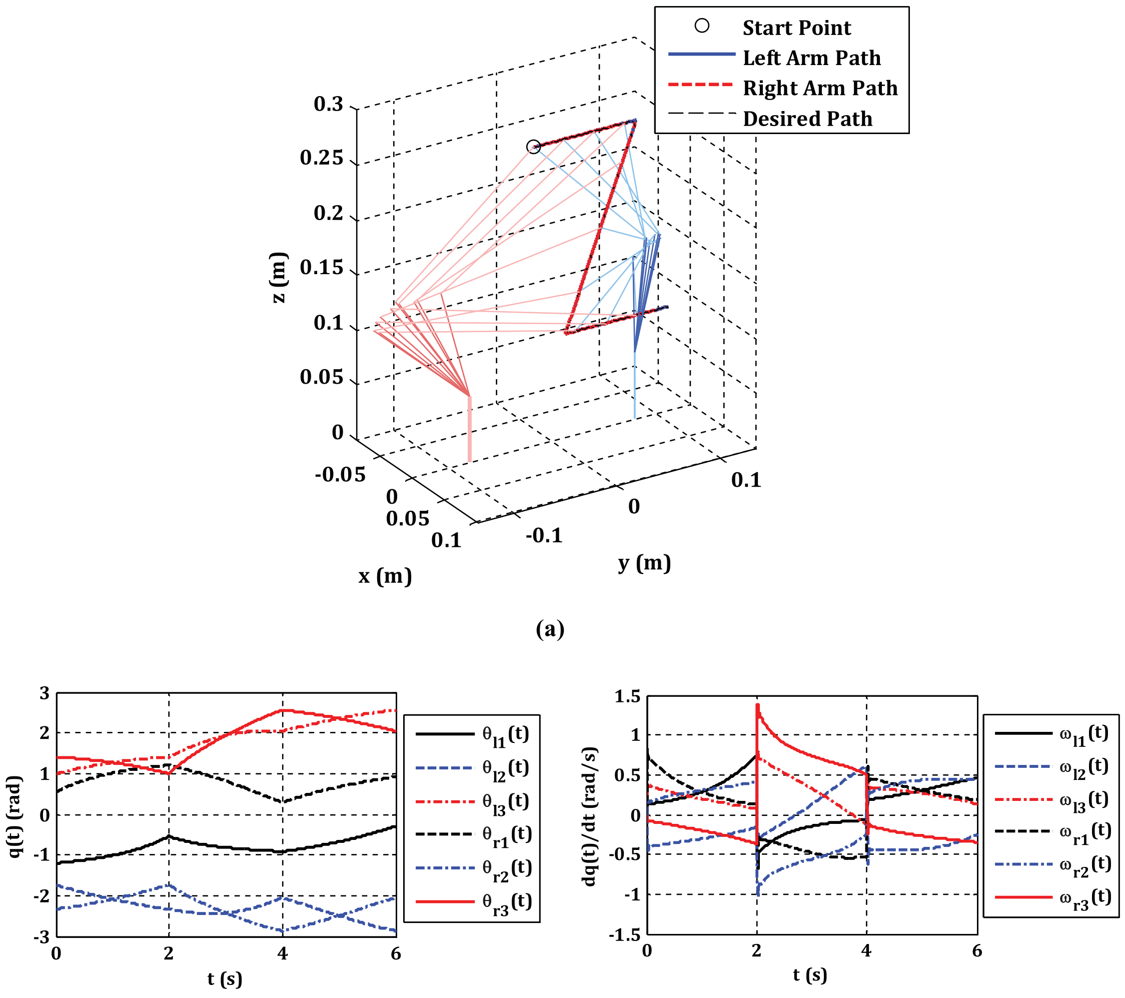
Finally, the covered path shape of two arms is as illustrated in Fig. 2. The z-shaped path has been selected because it has two critical points where in them the robot should stop and change the direction of its movement. The results of this simulation are in Figs. 2 and 3.

According to the results, it is clear that the related control signal to the right arm, respect to left arm is tripled. This is due to the difference between each end-effector's path and that is why controller parameters for both arms are the same as they have been intended.

Figure 4 shows the tracking errors. By regarding the Fig. 4 which is depicted the errors, it can be concluded that if the tolerance is allowed only 1 cm, robot can move the object. Therefore, by designing joint points, and taking into account the displacement of the terminal link to maximum of 1 cm, object can be moved.

The specific path (z-shaped path) causes the large change in the angular velocity at the junction of the path (see the Fig. 2). In the experiment set-up although the large change have been created from the algorithm, actually the duration of the large change in junction is very short and causes a little shake in manipulator but the vibrations are not that much high that make problem.

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press



(a)

(b)

(c)

Fig. 2. The simulation results for the cooperative arms. (a) End-effector path. (b) Generalized coordinate values. (c) Variation of generalized coordinate values.

5. Experimental Results

In performed experimental test, all the conditions including desired path and starting point are as like as simulation conditions. Moreover, the amount of gains which are considered in the previous section, are applied in this test too. Therefore, considering the same conditions for both modeling and experimental test parts, a good comparison on these two different modes can be done. The schematic block diagram of the process is represented in Fig. 5. The actual and nominal error which are represented in Fig. 5, refer to the error for the system with and without uncertainty, respectively. In Fig. 5, the real state and nominal state which are presented by $\mathbf{x}$, $\hat{\mathbf{x}}$.

By using the PSA, the control signals "u" (see Fig. 5 inside Matlab Script section) is defined by equations which are solved numerically based on the methodology of Taylor series. After that, the calculated control signals implement on the real robot. With using the PSA approach, the delay between decision and acting become less because the huge number of equations are solved quickly in each loop of the program. The control law is an algebraic equation and can be programmed into the controller, however, solution to the Riccati equation needs numerical computation methods. In this work, the PSA method provides a set of closed-form equations that provide optimal gain to the SDRE. For the cases that the finite time implementation is required, the use of the PSA is no longer possible and numerical implementation is necessary. So, the optimal gain, generated from PSA for the optimal part (38), is inserted to (15), and is programmed into the Arduino controller. With proper

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

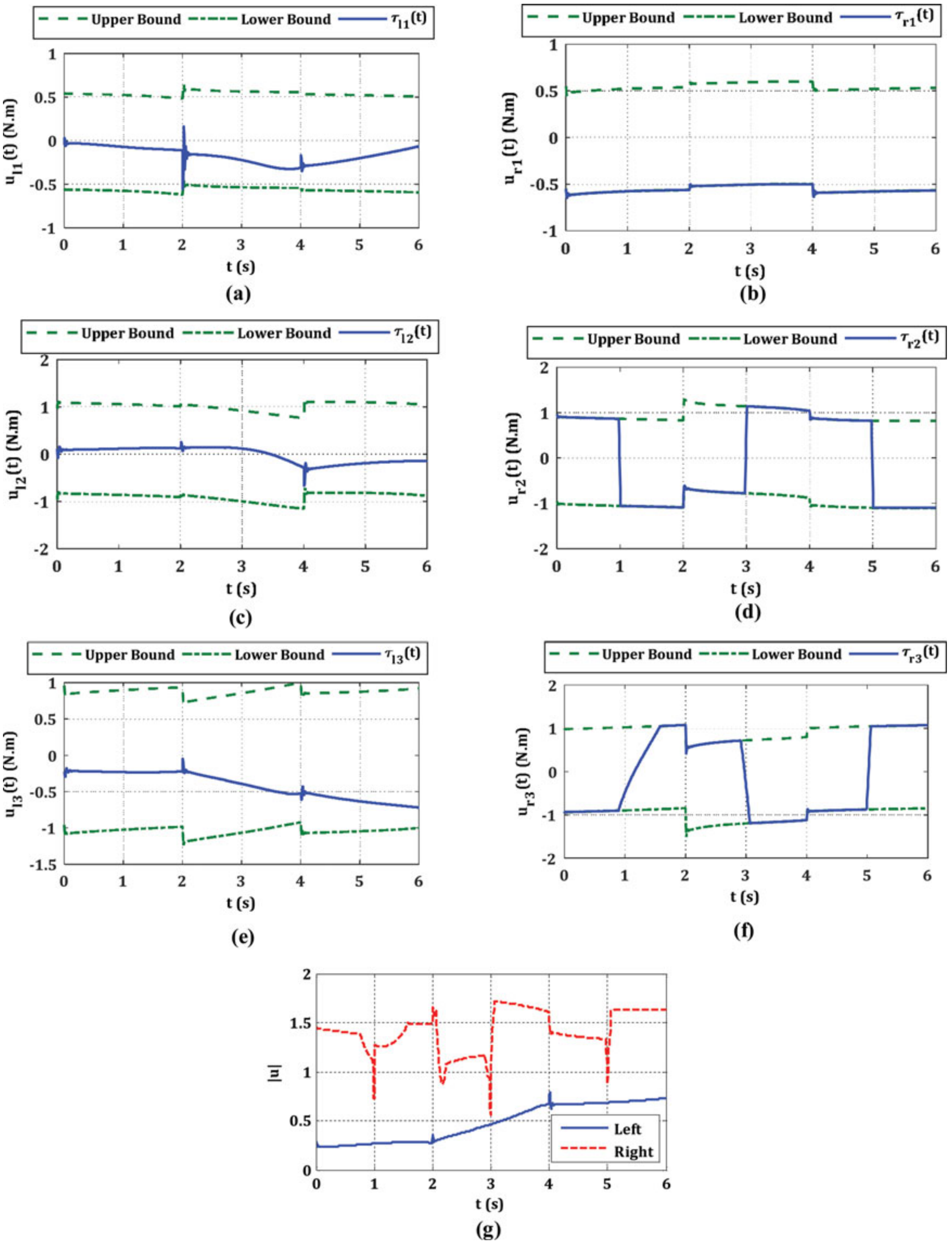


Fig. 3. The input torque values for the cooperative arms. (a) Input torque of first left arm actuator. (b) Input torque of first right arm actuator. (c) Input torque of second left arm actuator. (d) Input torque of second right arm actuator. (e) Input torque of third left arm actuator. (f) Input torque of third right arm actuator. (g) Norm of input torque.

selection of time step, the controller act in the experimental setup and computes the necessary PWM of servo motors.

The results of experimental test are shown in Fig. 6. Object manipulation task in theoretical and experimental implementation is defend as carrying an object with two arms in a cooperative mode;

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

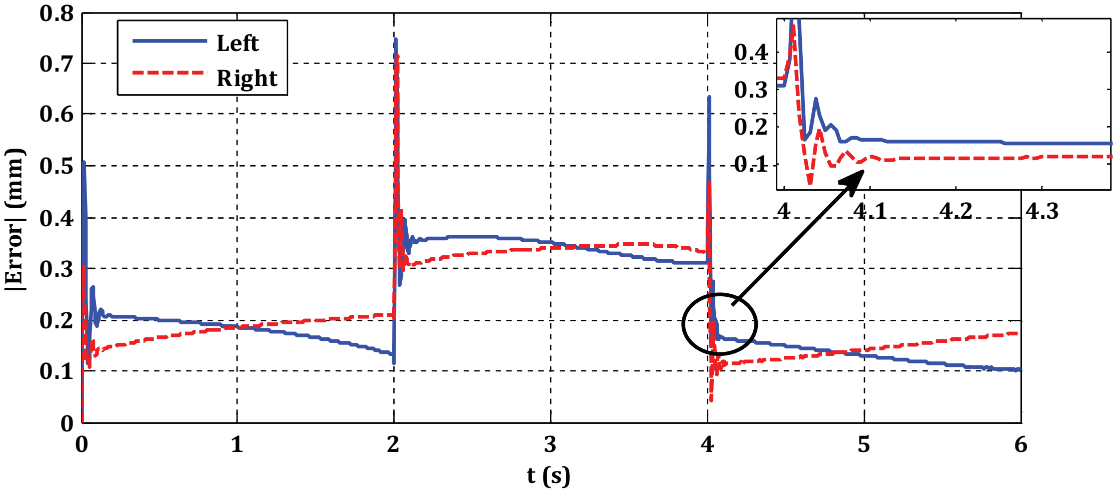


Fig. 4. Norm of the end-effector errors.

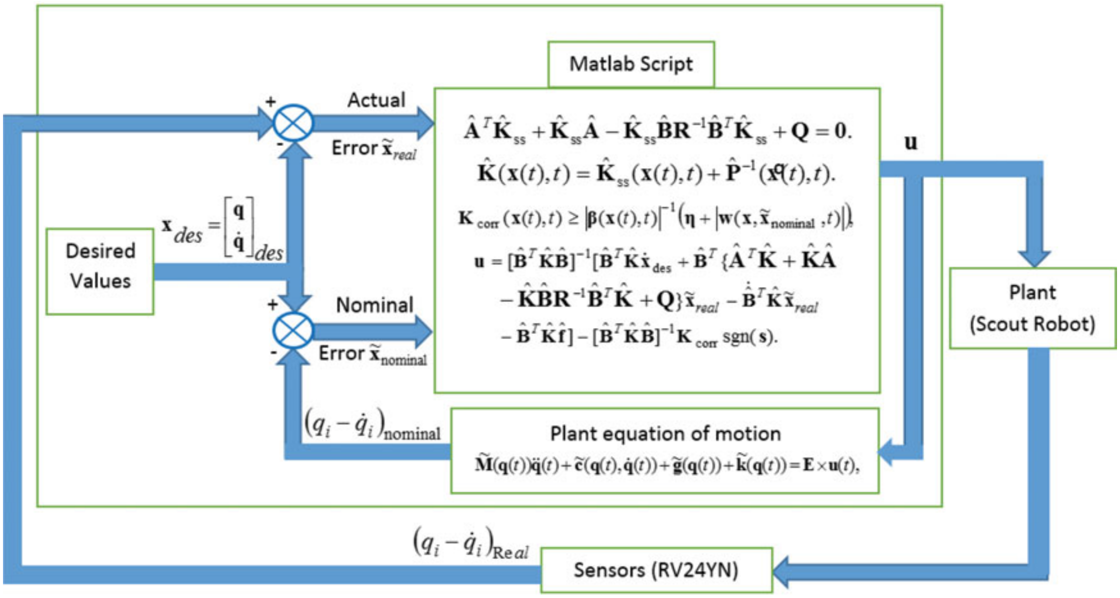

Matlab Script
$\hat{\mathbf{A}}^T\hat{\mathbf{K}}_{ss}+\hat{\mathbf{K}}_{ss}\hat{\mathbf{A}}-\hat{\mathbf{K}}_{ss}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}}_{ss}+\mathbf{Q}=\mathbf{0}.$
$\hat{\mathbf{K}}(\mathbf{x}(t),t)=\hat{\mathbf{K}}_{ss}(\mathbf{x}(t),t)+\hat{\mathbf{P}}^{-1}(\mathbf{x}(t),t).$
$\mathbf{K}_{\text{corr}}(\mathbf{x}(t),t)\geq|\boldsymbol{\beta}(\mathbf{x}(t),t)|^{-1}(\boldsymbol{\eta}+|\mathbf{w}(\mathbf{x},\tilde{\mathbf{x}}_{\text{nominal}},t)|).$
$\mathbf{u}=[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\dot{\mathbf{x}}_{\text{des}}+\hat{\mathbf{B}}^T\{\hat{\mathbf{A}}^T\hat{\mathbf{K}}+\hat{\mathbf{K}}\hat{\mathbf{A}}$
$-\hat{\mathbf{K}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}}+\mathbf{Q}\}\tilde{\mathbf{x}}_{real}-\dot{\hat{\mathbf{B}}}^T\hat{\mathbf{K}}\tilde{\mathbf{x}}_{real}$
$-\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{f}}]-[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}\mathbf{K}_{\text{corr}}\,\text{sgn}(\mathbf{s}).$
u
Actual
Error $\tilde{\mathbf{x}}_{real}$
Nominal
Error $\tilde{\mathbf{x}}_{\text{nominal}}$
+
-
+
-
Desired Values
$\mathbf{x}_{des}=\begin{bmatrix}\mathbf{q}\\ \dot{\mathbf{q}}\end{bmatrix}_{des}$
Plant (Scout Robot)
$(q_i-\dot{q}_i)_{\text{nominal}}$
Plant equation of motion
$\widetilde{\mathbf{M}}(\mathbf{q}(t))\ddot{\mathbf{q}}(t)+\tilde{\mathbf{c}}(\mathbf{q}(t),\dot{\mathbf{q}}(t))+\tilde{\mathbf{g}}(\mathbf{q}(t))+\tilde{\mathbf{k}}(\mathbf{q}(t))=\mathbf{E}\times\mathbf{u}(t),$
$(q_i-\dot{q}_i)_{\text{Real}}$
Sensors (RV24YN)

Fig. 5. The schematic block diagram of structure of control system.

the task of holding the object has not been considered yet. Two arms share the load in same trajectory that the cooperation increases the load capacity.

According to Fig. 6, it is seen that in the initial and final moments, end-effector is to some extent away of the desired path (the z-shaped path), however, on the middle of the path it is almost coincident with the desired path. The results are noisy because the potentiometers model "RV24YN" are used for feedback. The robot is supposed to follow the desired path which is in Z shape; therefore, by using the inverse kinematic of the robot, the desired joint angles obtained which are shown by the red dash line in Fig. 7. According to Fig. 7, it is observed that the error value in the earliest moments has the maximum value so that the maximum error value at the time of 1 s is equal to 32 mm and the end-effector error value of the intended desired path is 15.3 mm. Moreover, by considering the forward kinematic which is used to transform the joint angles to the end-effector path, the small error in joint angle's value may causes larger error in the end-effector path. Moreover, it should be mentioned that, the observed deviations from the desired path are due to non-linearities in the potentiometers.

As it is observed in Fig. 7, the results recorded in the first potentiometer and its curve fitting almost match the desired angle on the first link. The maximum recorded error value by first potentiometer

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

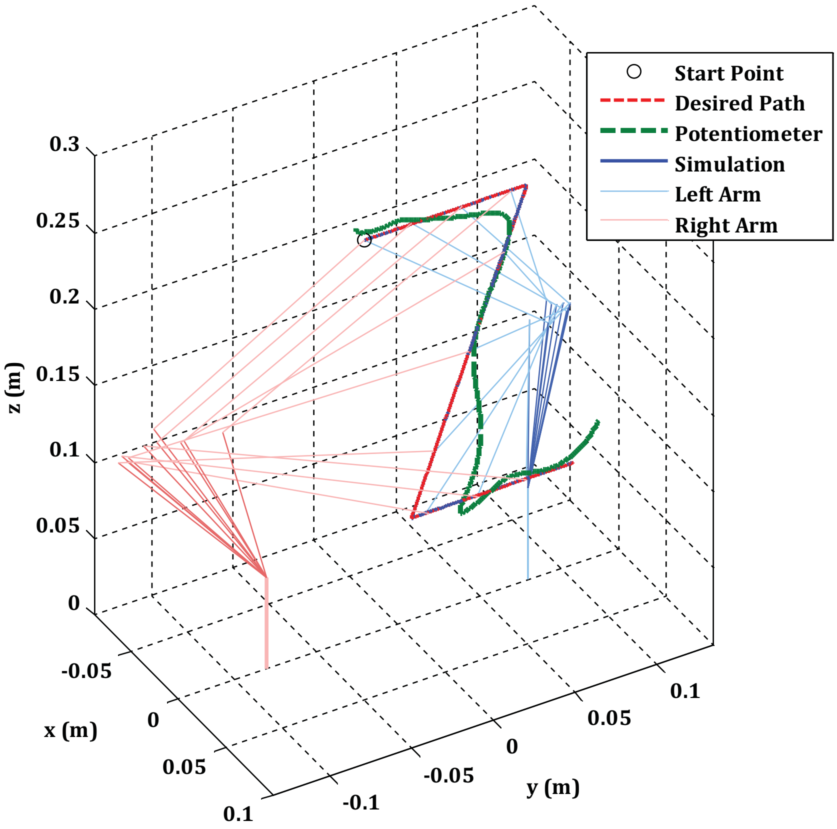


Fig. 6. The measured and simulated end-effector path.

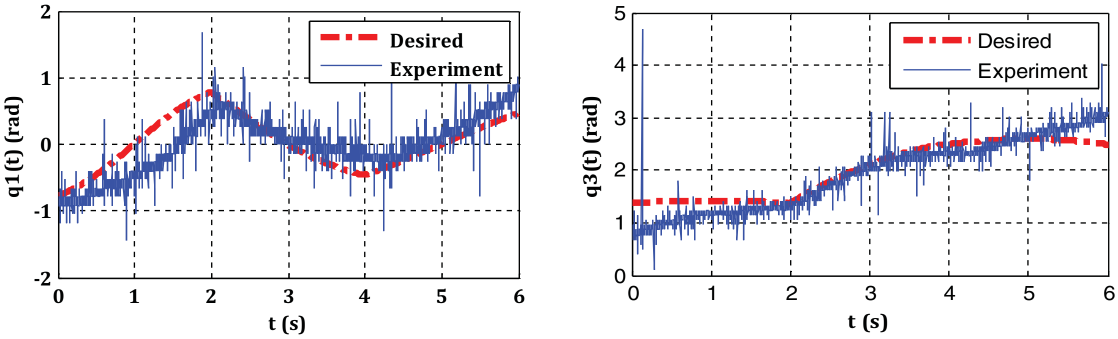


Fig. 7. Experiments data of the generalized coordinate.

of the desired amount is 0.1 radians and the medium error value is 0.03 radians. In the third joint, the recorded data is different from the desired value. The maximum error value of the recorded data into recorded desired angle in the third joint is 0.12 radians and its medium value is 0.04 radians. According to Fig. 7, it is obtained that data recorded in the first joint at the beginning and end of the path are a bit away of the desired value. The maximum error value of the recorded data is 0.21 radians and its medium value is 0.08 radians.

6. Dynamic Load Carrying Capacity (DLCC) Algorithm

DLCC is one of the important features that is discussed in robotics. The DLCC magnitude, by the amount of position error on the end-effector that should not be exceeded than permitted value, is

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

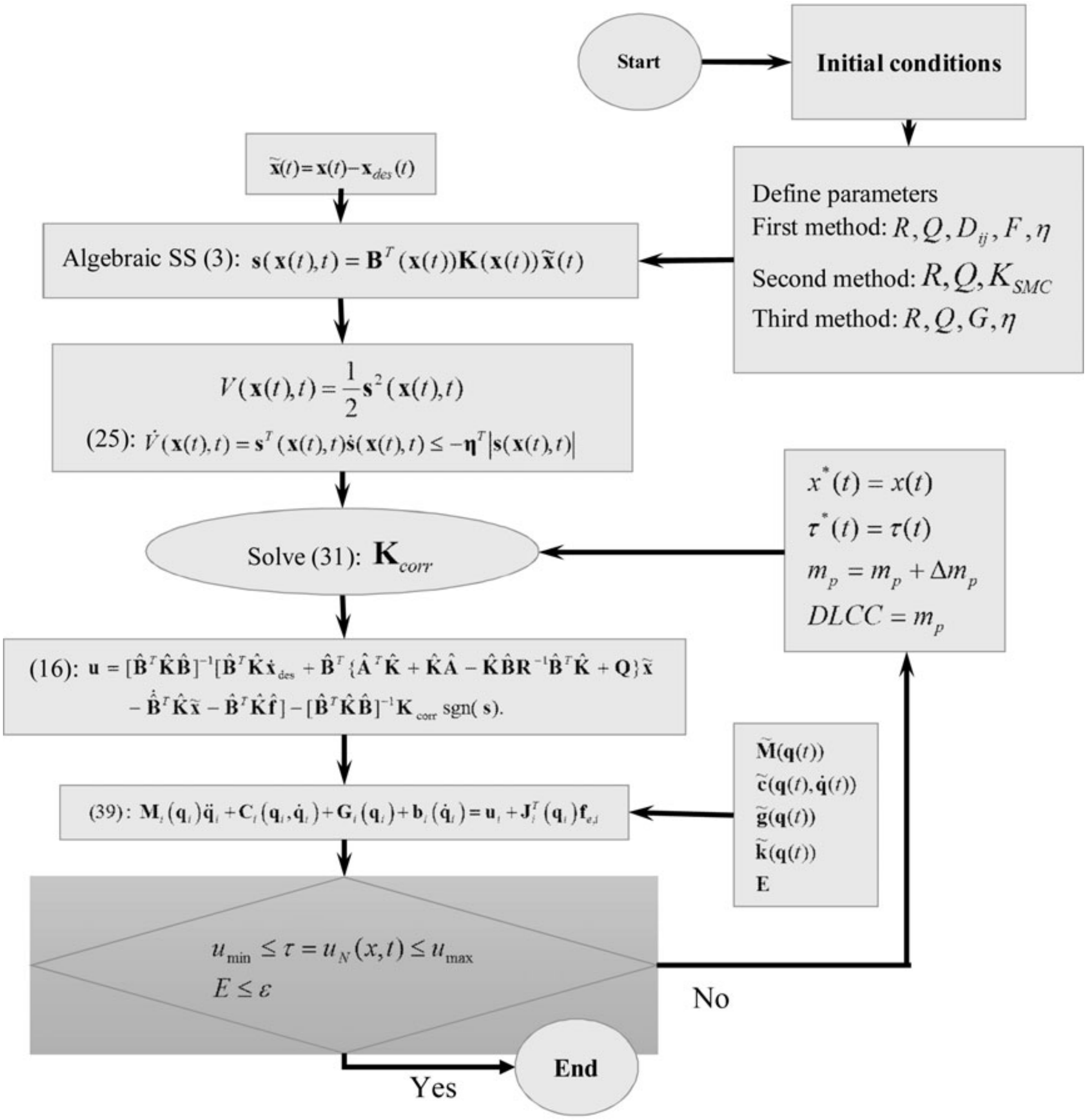

Start
Initial conditions
$\tilde{\mathbf{x}}(t)=\mathbf{x}(t)-\mathbf{x}_{des}(t)$
Define parameters
First method: R,Q,D_{ij},F,η
Second method: R,Q,K_{SMC}
Third method: R,Q,G,η
Algebraic SS (3): $\mathbf{s}(\mathbf{x}(t),t)=\mathbf{B}^T(\mathbf{x}(t))\mathbf{K}(\mathbf{x}(t))\tilde{\mathbf{x}}(t)$
$V(\mathbf{x}(t),t)=\frac{1}{2}\mathbf{s}^2(\mathbf{x}(t),t)$
(25): $\dot{V}(\mathbf{x}(t),t)=\mathbf{s}^T(\mathbf{x}(t),t)\dot{\mathbf{s}}(\mathbf{x}(t),t)\leq-\boldsymbol{\eta}^T|\mathbf{s}(\mathbf{x}(t),t)|$
$x^*(t)=x(t)$
$\tau^*(t)=\tau(t)$
$m_p=m_p+\Delta m_p$
$DLCC=m_p$
Solve (31): $\mathbf{K}_{corr}$
(16): $\mathbf{u}=[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\dot{\mathbf{x}}_{des}+\hat{\mathbf{B}}^T\{\hat{\mathbf{A}}^T\hat{\mathbf{K}}+\hat{\mathbf{K}}\hat{\mathbf{A}}-\hat{\mathbf{K}}\hat{\mathbf{B}}\mathbf{R}^{-1}\hat{\mathbf{B}}^T\hat{\mathbf{K}}+\mathbf{Q}\}\tilde{\mathbf{x}}$
$-\dot{\hat{\mathbf{B}}}^T\hat{\mathbf{K}}\tilde{\mathbf{x}}-\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{f}}]-[\hat{\mathbf{B}}^T\hat{\mathbf{K}}\hat{\mathbf{B}}]^{-1}\mathbf{K}_{corr}\,\mathrm{sgn}(\mathbf{s}).$
(39): $\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i+\mathbf{C}_i(\mathbf{q}_i,\dot{\mathbf{q}}_i)+\mathbf{G}_i(\mathbf{q}_i)+\mathbf{b}_i(\dot{\mathbf{q}}_i)=\mathbf{u}_i+\mathbf{J}_i^T(\mathbf{q}_i)\mathbf{f}_{e,i}$
$\tilde{\mathbf{M}}(\mathbf{q}(t))$
$\tilde{\mathbf{c}}(\mathbf{q}(t),\dot{\mathbf{q}}(t))$
$\tilde{\mathbf{g}}(\mathbf{q}(t))$
$\tilde{\mathbf{k}}(\mathbf{q}(t))$
$\mathbf{E}$
$u_{\min}\leq\tau=u_N(x,t)\leq u_{\max}$
$E\leq\varepsilon$
No
Yes
End

Fig. 8. The control algorithm for computing DLCC.

limited. The tracking error is obtained from (59).

$$E(t)=\sqrt{(x_e(t)-x_{ed}(t))^2+(y_e(t)-y_{ed}(t))^2+(z_e(t)-z_{ed}(t))^2}, \tag{59}$$

which $R_e=[x_e\ y_e\ z_e]^T$ is the actual position of the end-effector and $R_{ed}=[x_e\ y_{de}\ z_{ed}]^T$ is the end-effector desired position vector. The other constraint that limits the amount of allowable load can be pointed out to restrictions on motor (DC motor) torque of each joints. In fact, it is needed to match with the capability of applying the minimum and maximum torque of the motor which is calculated in (52). With this constraint, an iterative trend for load carrying capacity that has been illustrated in Fig. 8 is offered.

The equations of motion for the Scout robot which were explained in the previous section, as well as the control algorithm are considered to calculate the maximum DLCC as shown in illustrated algorithm in Fig. 8. In order to test the amount of DLCC, the loads are increased in each loop of the mentioned algorithm. Therefore, the small basket (connected by two ropes to the Scout's arms) is used and the loads which are several blocks by the mass of 100 g, have been added to the basket step by step in order to compute the DLCC. Moreover, torque values in a state of exhaustion and the

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

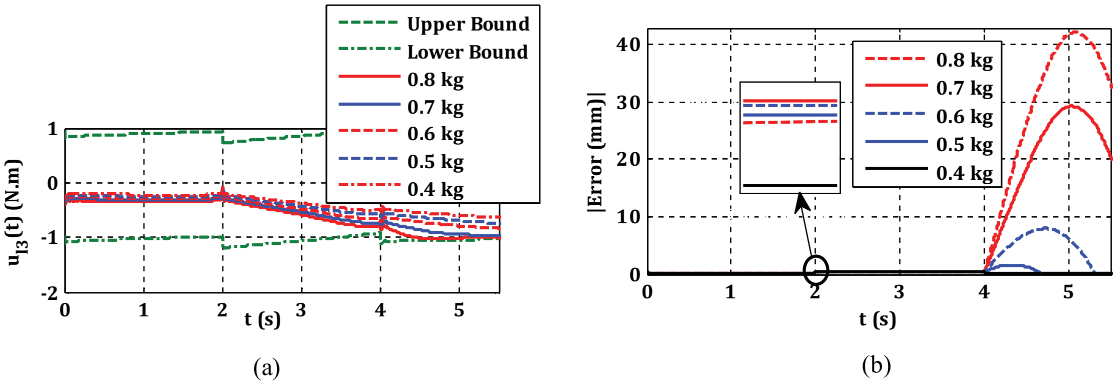


Fig. 9. (a) Torque input with bound consuming. (b) Norm of end-effector error.

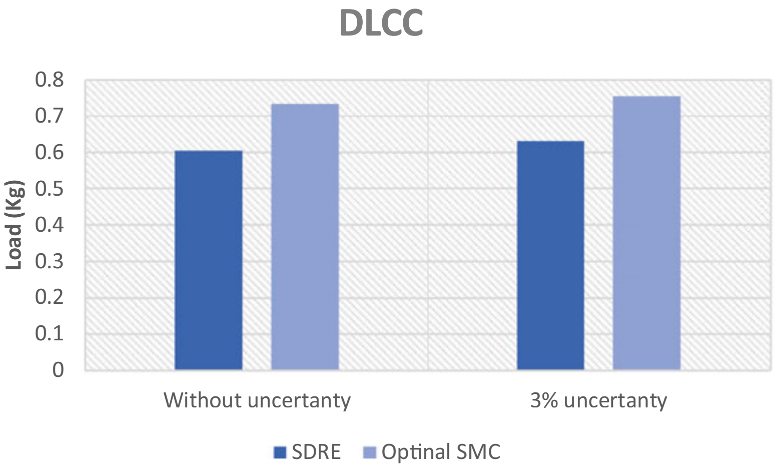


Fig. 10. The amount of DLCC.

angular velocity are required in no-load speed mode that their values for arm's and wheel's actuators are expressed in Table II, which related results of the simulations with different applied loads are shown in Fig. 9.

Also, a comparison of the amount of DLCC by the usual method SDRE has been done and its results are brought in Fig. 10.

After applying the method of calculating the DLCC, results for OSMC method, presented in this paper, have been improved compared to the SDRE control technique and reached to $m_p = 0.756$ kg.

7. Conclusion

In this paper, by considering the algebraic sliding surface, an OSMC method is derived from combination of the SMC and the SDRE technique and the stability of the mentioned method investigated based on Lyapunov second method. By considering both certain and uncertain system in the control algorithm, the control output signal is designed based on the system with uncertainty, however, the control gains are derived for the assumed certain system. Moreover, the OSMC method has been applied for cooperative arms. In addition, the percentage of load distribution between each manipulator has been derived based on cost function in order to increase the DLCC and by comparison with SDRE technique, DLCC increased almost 15% while using OSMC. Although one of the problem in optimal control is the delay for real time implementation, PSA is used for implementation.

References

1. T. Kokkinis, "Dynamic hybrid control of cooperating robots by nonlinear inversion," *Robotics and Autonomous Systems* **5**(4), 359–368 (1989).

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press

2. X. Yun, R. V. Kumar, N. Sarkar and E. Paljug, "Control of multiple arm systems with rolling constraints," Technical Report, MS-CIS-91-79 (1991).
3. J. T. Wen and K. K. Delgado, "Motion and force control of multiple robotic manipulators," *Automatica* **28**(4), 729–743 (1992).
4. W. B. Gao and D. Xiao, "Tracking tasks of massive objects by multiple robot systems with non-firm grasping," *Mechatronics* **3**(6), 727–746 (1993).
5. C. J. Li, "Coordinated motion control of multi-arm robot systems with optimal load distribution," *Systems & Control Letters* **15**(3), 237–245 (1990).
6. S. T. Lin and H. C. Tsai, "Impedance control with on-line neural network compensator for dual-arm robots," *Journal of Intelligent and Robotic Systems* **18**(1), 87–104 (1997).
7. J. S. Liu and S. L. Chen, "Robust hybrid control of constrained robot manipulators via decomposed equations," *Journal of Intelligent and Robotic Systems* **23**(1), 45–70 (1998).
8. J. Zhao and S. X. Bai, "Load distribution and joint trajectory planning of coordinated manipulation for two redundant robots," *Mechanism and Machine Theory* **34**(8), 1155–1170 (1999).
9. Z. Jing and S. X. Bai, "The study of coordinated manipulation of two redundant robots with elastic joints," *Mechanism and Machine Theory* **35**(7), 895–909 (2000).
10. K. Subbarao, A. Verma and J. L. Junkins, "Model Reference Adaptive Control of Constrained Cooperative Manipulators," *Proceedings of the IEEE International Conference on Control Applications*, Mexico City (Sep. 2001) pp. 553–558.
11. Z. Li, S. S. Ge and Z. Wang, "Robust adaptive control of coordinated multiple mobile manipulators," *Mechatronics* **18**(5), 239–250 (2008).
12. A. Ghasemi and M. Keshmiri, "Performance Assessment of a Decentralized Controller for Cooperative Manipulators; Numerical and Experimental Study," *Proceedings of the 6th International Symposium on Mechatronics and its Applications*, Sharjah, UAE (Mar. 2009) pp. 1–6.
13. N. Yagiz, Y. Hacioglu and Y. Z. Arslan, "Load transportation by dual arm robot using sliding mode control," *Journal of Mechanical Science and Technology* **24**(5), 1177–1184 (2010).
14. M. H. Korayem, M. Jalali and H. Tourajizadeh, "Dynamic load carrying capacity of spatial cable suspended robot: Sliding mode control approach," *International Journal of Advanced Design and Manufacturing Technology* **5**(3), 73–81 (2012).
15. Chinelato Caio Igor Gonçalves and Luiz de Siqueira Martins-Filho, "Control of cooperative mobile manipulators transporting a payload," *Proceedings of the Mechanical Engineering (COBEM), International Congress of. ABCM* (2013).
16. S. Suzuki, K. Furuta and Y. Pan, "State-dependent sliding-sector VS-control and application to swing-up control of pendulum," *Proceedings of the 42nd IEEE Conference on Decision and Control Maui*, Hawaii USA (Dec. 2003).
17. S. Suzuki, Y. Pan, K. Furuta and S. Hatakeyama, "Vs-control with time-varying sliding sector design and application to pendulum," *Asian Journal of Control* **6**(3), 307–316 (Sep., 2004).
18. A. H. Korayem, S. R. Nekoo and M. H. Korayem, "Sliding mode control design based on the state-dependent Riccati equation: Theoretical and experimental implementation," *International Journal of Control* **91**, 01–30 (2018).
19. T. Cimen, "Survey of state-dependent Riccati equation in nonlinear optimal feedback control synthesis," *Journal of Guidance, Control, and Dynamics* **35**(4), 1025–1047 (2012).
20. M. H. Korayem and S. R. Nekoo, "Finite-time state-dependent Riccati equation for time-varying nonaffine systems: Rigid and flexible joint manipulator control," *ISA Transactions* **54**, 125–144 (2015).
21. Slotine, E. Jean-Jacques and L. Weiping, *Applied Nonlinear Control*, vol. 199, no. 1 (Englewood Cliffs, NJ: Prentice-Hall, 1991).
22. M. H. Korayem, M. Irani and S. Rafee Nekoo, "Load maximization of flexible joint mechanical manipulator using nonlinear optimal controller," *Acta Astronautica* **69**(7), 458–469 (2011).
23. H. T. Banks, B. M. Lewis and H. T. Tran, "Nonlinear feedback controllers and compensators: A state-dependent Riccati equation approach," *Computational Optimization and Applications* **37**(2), 177–218 (2007).
24. M. H. Korayem, M. Irani and S. R. Nekoo. "Analysis of manipulators using SDRE: A closed loop nonlinear optimal control approach," *Scientia Iranica. Transaction B, Mechanical Engineering* **17**(6), 456–467 (2010).
25. M. H. Korayem and S. R. Nekoo, "The SDRE control of mobile base cooperative manipulators: Collision free path planning and moving obstacle avoidance," *Robotics and Autonomous Systems* **86**, 86–105 (2016).
26. M. H. Korayem and S. R. Nekoo, "Controller design of cooperative manipulators using state-dependent Riccati equation," *Robotica* **36**, 1–32 (2017).

https://doi.org/10.1017/S0263574718001030 Published online by Cambridge University Press