Robotica (2018) volume 36, pp. 484–515. © Cambridge University Press 2017
doi:10.1017/S0263574717000522

Controller design of cooperative manipulators using state-dependent Riccati equation

Moharam Habibnejad Korayem and Saeed Rafee Nekoo*

Robotic Research Laboratory, Center of Excellence in Experimental Solid Mechanics and Dynamics, School of Mechanical Engineering, Iran University of Science and Technology (IUST), 1684613114, Tehran, Iran. Email: hkorayem@iust.ac.ir

(Accepted October 14, 2017, First published online: November 10, 2017)

SUMMARY

This study examined the use of a state-dependent Riccati equation (SDRE) for controller design and analysis of cooperative manipulators. The connection of end-effectors when holding an object imports constraint and complexity into the problem. Optimal load distribution (OLD) was used to divide the load between arms using a desired rate and omitting Lagrange multipliers. General dynamic structure, OLD formulation, and controller design are presented for an arbitrary number of manipulators. State-dependent coefficient parameterizations for rigid and flexible joint manipulators assuming friction for joints of them were investigated by two methods: controlling each robot independently and an entire system of robots uniformly. The effectiveness of the method, a decrease in errors, and increased stability in motion were also observed. The increase in the number of manipulators greatly expanded the state vector of the system. The SDRE was able to address this by simulation of four arms, each one possessing seven degrees of freedom (DoF). Analyses of a practical model (Scout robot) consisting of two arms with three DoF were presented and the results for connected arms and free arms were compared. The experimental data validated the simulation results and indicated that cooperation definitely improves load-carrying capacity and precision of trajectory tracking.

KEYWORDS: SDRE, Cooperative manipulators, SDC parameterization, Optimal load distribution, Load-carrying capacity.

1. Introduction

The present study analyzed controller design of cooperative manipulators using a state-dependent Riccati equation (SDRE). Cooperative robotics has many branches, including load-carrying, assembly, welding, and master-slave systems. The present study focuses on robotic arms which hold an object and carry it on a specified trajectory. Common applications for cooperative manipulators with connected end-effectors include part assembling and load carrying. Changes in dynamics are expected because contact with end-effectors influences mathematical modeling. The objective of assembly is clear and without a connection with manipulators, the goal will not be achieved. The justification for following question is important: why is it better to model, control, and analyze a system of robots with complex dynamics and constraints instead of one stronger robot. No justification is required for carrying objects with large geometries with respect to the size of the robot because the cooperation of multiple arms increases stability in motion and load capacity. The problem is to justify usage of two or more manipulators to carry a lumped mass (load of negligible size with respect to arms). The present study found that cooperation increases load-carrying capacity, even for lumped mass loads.

Hayati *et al*. investigated the control of a dual arm mechanism as a master-slave system.[1] Kokkinis proposed cooperative control of manipulators using the dynamic inversion method.[2] Contact between end-effectors generated additional constraint and redundancy was removed from the model by introduction of the internal force produced by the object. Li developed motion control of a

* Corresponding author. E-mail: saerafee@yahoo.com

www.manaraa.com

multi-arm robotic system using the optimal load distribution method.[3] Optimization was carried out using the Lagrange method by minimizing actuator energy consumption. Yun *et al*. studied control of cooperative robotic systems with rolling constraints.[4] The equation of the system was not input-state linearizable, although input–output linearization was possible. Wen and Delgado proposed motion and force control of multiple manipulators.[5] Gao and Xiao showed the superiority of cooperation in robotics by transporting large-sized objects.[6]

Lin and Tsi used impedance control for a dual arm mechanism.[7] A neural network compensator was also implemented to improve the design for unknown parameters. Yale and Agrawal applied a Lyapunov-based controller for a constrained cooperative system.[8] Liu and Chen proposed a robust hybrid control of constrained robotic manipulators using decomposed equations.[9] Previous studies also addressed load distribution and trajectory planning for two redundant robots[10] and for a flexible joint ones.[11]

Ghasemi and Keshmiri used decentralized control of a dual-arm system to track a predefined path.[12] The system as a closed-chain mechanism was reduced to two degrees of freedom (DoF). A simulation and experimental results were presented. Tavasoli *et al*. investigated two time-scale control and observer design for trajectory tracking of two cooperating robotic manipulators moving a flexible beam.[13] Homaei and Keshmiri studied optimal trajectory planning for minimum vibration of flexible redundant cooperative manipulators.[14] Yagiz *et al*. employed a sliding mode control for a dual-arm system.[15] Rastegari and Moosavian investigated multiple impedance control of free-flying space robots using virtual linkages.[16] The frequent use of cooperative manipulators for load transportation reveals the importance of this subject for carrying objects.[17–19]

Increasing load-carrying capacity was the motivation for using the cooperative robotic systems in the present research. Dynamic load-carrying capacity (DLCC) was computed using the SDRE method for manipulators, though none of them covered the issue of cooperation.[20–22] Primary research on dynamics (without control) for cooperation showed improvement in DLCC, but the complexity of the control design resulted in analysis of a dual-arm robot without link connections.[23] Korayem and Nekoo used cooperative control by means of the SDRE controller for mobile robots.[24] This current paper studies the concept of cooperation more deeply with experimental validation in comparison with.[24] Considering cooperation for flexible joint manipulators and friction in the model is another highlight of this current research.

Notation: $\Re^n$ presents the n-dimensional Euclidean space, $\Re^{m\times n}$ is the set of $m \times n$ real matrices, $(\cdot)^T$ is transpose of a matrix, $\exp(\cdot)$ denotes the exponential function, $\text{sgn}(\cdot)$ means the signum function, $(\cdot)^+$ is generalized inverse (pseudo inverse) of a matrix. $\text{diag}(\cdot)$ means a diagonal matrix with zero non-diagonal arrays, $\mathbf{I}_{n\times n}$ and $\mathbf{0}_{n\times n}$ are $n \times n$ identity and zero matrices with respect. The lower case bold letters in the formulas are dedicated to vectors and the capital letter ones to matrices.

The remainder of this paper is assigned as follows. Section 2 presents the kinematics and dynamics of cooperative manipulators. Section 3 proposes the controller design of them consisting of the SDC parameterizations and the SDRE formulation. Simulations are gathered in Section 4 for various cases, while Section 5 is devoted to experimental implementation. Finally, conclusions are summarized in Section 6.

2. Kinematics and Dynamics of Cooperative Robots

2.1. Kinematics

Consider m manipulators holding and carrying mass of m_p in an arbitrary, but predefined trajectory, illustrated in Fig. 1. *The only condition on trajectory is non-singularity of Jacobian matrices of robots.* Each arm possesses n DoF and obeys the Denavit–Hartenberg (D–H) convention with either revolute or prismatic joints, the type of arms might be different.

Movement, trajectory, and coordinates of robots are measured with respect to main reference frame X_o, Y_o, Z_o. Coordinates of center-of-mass (CoM) of the object is referred to X_e, Y_e, Z_e and X_i, Y_i, Z_i is placed on each end-effector of the manipulators for $i = 1, ..., m$. Load is regarded as two forms: (1) lumped mass and (2) distributed mass. First case considers the load without dimension; hence, the trajectories of load and end-effectors (X_i, Y_i, $Z_i = X_e$, Y_e, Z_e) are similar. Orientation of grippers is not needed for a lumped-mass load, as a result, minimum DoF of arms for trajectory design of planar

www.manaraa.com

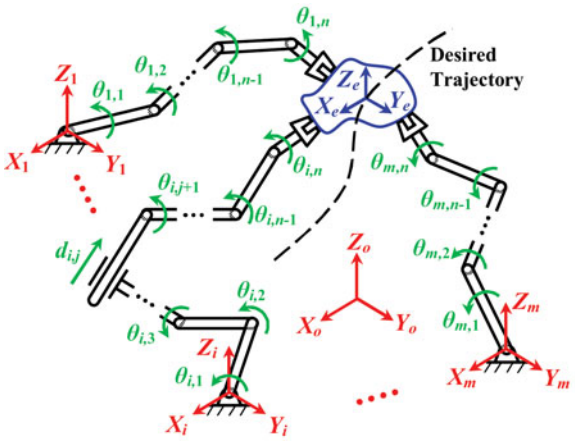


Fig. 1. Schematic view of m manipulators as a cooperative system.

motion is two and for spatial motion is three. Distributed mass demands an object with geometry represented by mass and moment of inertia. Moreover, the trajectory of load and end-effectors might be different $X_i, Y_i, Z_i \neq X_e, Y_e, Z_e$; hence, $m+1$ trajectories must be designed which imposes more complexity on the problem. Orientation of end-effector is necessary to carry an object with desired rotational movement. Therefore, this type of system requires manipulators with higher number of DoF. Following steps after trajectory design for cooperative manipulators are similar to independent arms and could be followed-up with text books such as ref. [25]. Forward kinematics is computed from transformation matrices and the D–H algorithm, while a definite closed-form answer is not regarded for inverse kinematics.

2.2. *Dynamics of rigid joint arms*

Dynamic motion of an object is regarded as[3]

$$\begin{bmatrix} m_p\mathbf{I}_{3\times3} & | & \mathbf{0}_{3\times3} \\ - - - - & | & - - - - - \\ \mathbf{0}_{3\times3} & | & [\mathbf{I}_p]_{3\times3} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}}_e(t) \\ - - - \\ \dot{\omega}_e(t) \end{bmatrix} + \begin{bmatrix} m_p\mathbf{g} \\ - - - - - - - - - - \\ \omega_e(t) \times \left(\mathbf{I}_p\omega_e(t)\right) \end{bmatrix} = -\mathbf{f}_e(t), \tag{1}$$

where $\mathbf{I}_{3\times3}$ is identical matrix of 3×3, $\mathbf{I}_p$ is inertial matrix of the object, m_p is mass of the object, $\ddot{\mathbf{p}}_e(t) = [\ \ddot{X}_e(t) \quad \ddot{Y}_e(t) \quad \ddot{Z}_e(t)\ ]^T$ denotes acceleration vector in reference coordinates, $\omega_e(t)$ is angular velocity, and $\dot{\omega}_e(t)$ is angular acceleration vector. The first three rows of Eq. (1) present Newton's second law and the second three rows of Eq. (1) denote Euler's rotation equations generated from derivative of angular momentum with respect to time $\mathbf{H}_e(t) = \mathbf{I}_p\omega_e(t)$. Gravity vector is shown by $\mathbf{g}$, i.e. $\mathbf{g} = [\ 0, \quad 0, \quad g_0\ ]^T$ expresses that there is gravity acceleration in Z_e direction. External force $\mathbf{f}_e(t) \in \Re^{6\times1}$ caused by the object must be divided between manipulators $\mathbf{f}_e(t) = \sum_{i=1}^{m} \mathbf{f}_{e,i}(t)$ to assign $\mathbf{f}_{e,i}(t)$ to each arm.

Considering the generalized coordinates of an arbitrary n-DoF robot as $\mathbf{q}_i(t) = \{\theta_{i,1}(t), ..., d_{i,j}(t), ..., \theta_{i,n}(t)\}$, the equation of motion of the ith arm is written:

$$\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{c}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{g}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i) = \mathbf{u}_i + \mathbf{J}_i^T(\mathbf{q}_i)\mathbf{f}_{e,i}, \tag{2}$$

where $\mathbf{M}_i(\mathbf{q}_i(t))$ is $n\times n$ inertia matrix, $\mathbf{c}_i(\mathbf{q}_i(t), \dot{\mathbf{q}}_i(t))$ is $n\times1$ Coriolis and centrifugal force vector, $\mathbf{g}_i(\mathbf{q}_i(t))$ is $n\times1$ gravity vector, $\mathbf{b}_i(\dot{\mathbf{q}}_i(t))$ presents $n\times1$ friction vector, $\mathbf{u}_i(t) \in \Re^{n\times1}$ denotes input vector ($f(t)$ is for prismatic actuator and $\tau(t)$ is for rotational one), and $\mathbf{J}_i(\mathbf{q}_i(t))$ is $6\times n$ Jacobian matrix. Friction force of jth link of ith manipulator is[25]

$$b_{i,j}(\dot{q}_{i,j}(t)) = b^v_{i,j}\dot{q}_{i,j}(t) + \mathrm{sgn}(\dot{q}_{i,j}(t))\left[b^d_{i,j} + (b^s_{i,j} - b^d_{i,j})\exp\left(\frac{-|\dot{q}_{i,j}(t)|}{\varepsilon}\right)\right], \tag{3}$$

www.manaraa.com

for $1 < i < m$ and $1 < j < n$. $b^v_{i,j}$ is viscous friction, $b^d_{i,j}$ is dynamic friction, $b^s_{i,j}$ shows static friction, and ε is a small positive constant.

The trajectory is predefined, hence Eq. (1) is a known equation, but the values of $\mathbf{f}_{e,i}(t)$ are unknown. To solve this problem and divide $\mathbf{f}_e(t)$ between robots, optimal load distribution method is presented. Extracting $\mathbf{f}_{e,i}(t)$ from Eq. (2):

$$\mathbf{f}_{e,i} = \left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+ \left[\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{c}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{g}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i) - \mathbf{u}_i\right], \tag{4}$$

and defining Lagrangian[3]:

$$\begin{aligned} L(\mathbf{u}, \gamma) = {} & \frac{1}{2}\sum_{i=1}^{m} \mathbf{u}_i^T \bar{\mathbf{Q}}_i \mathbf{u}_i \\ & + \gamma^T \left\{ \sum_{i=1}^{m} \left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+ \left[\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{c}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{g}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i) - \mathbf{u}_i\right] - \mathbf{f}_{e,i} \right\}, \end{aligned} \tag{5}$$

one may implement the OLD with regard to constraint (4) as follows. Generalized inverse "+" will be changed to inverse whenever $\mathbf{J}_i(\mathbf{q}_i(t))$ is a square matrix. $\bar{\mathbf{Q}}_i$ defines the weight of load distribution and $\gamma(t) \in \Re^{6\times 1}$ is Lagrange multipliers vector. Using necessary condition for optimality,

$$\frac{\partial L(\mathbf{u}(t), \gamma(t))}{\partial \mathbf{u}_i(t)} = \bar{\mathbf{Q}}_i \mathbf{u}_i(t) - \left[\left(\mathbf{J}_i^T(\mathbf{q}_i(t))\right)^+\right]^T \gamma(t) = \mathbf{0}, \tag{6}$$

results in the input law of ith arm

$$\mathbf{u}_i(t) = \bar{\mathbf{Q}}_i^{-1}\left[\left(\mathbf{J}_i^T(\mathbf{q}_i(t))\right)^+\right]^T \gamma(t). \tag{7}$$

Substituting Eq. (7) into Eq. (4) and calculating $\mathbf{f}_e(t) = \sum_{i=1}^{m} \mathbf{f}_{e,i}(t)$ and then extracting $\gamma(t)$, result in

$$\begin{aligned} \gamma = {} & \left(\sum_{i=1}^{m} \left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+ \bar{\mathbf{Q}}_i^{-1} \left[\left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+\right]^T \right)^{-1} \\ & \times \left[\left\{ \sum_{i=1}^{m} \left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+ \left[\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{c}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{g}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i)\right] \right\} - \mathbf{f}_e \right]. \end{aligned} \tag{8}$$

Finally, substituting Eq. (8) in Eq. (7) provides the unknown of the problem (contribution of each arm):

$$\mathbf{f}_{e,i} = \left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+ \left[\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{c}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{g}_i(\mathbf{q}_i) + \mathbf{b}_i(\dot{\mathbf{q}}_i) - \bar{\mathbf{Q}}_i^{-1}\left[\left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+\right]^T \gamma \right]. \tag{9}$$

The term $\ddot{\mathbf{q}}_i(t)$ in Eq. (9) is computable by taking derivative of $[\dot{\mathbf{p}}_e^T(t), \omega_e^T(t)]^T = \mathbf{J}_i(\mathbf{q}_i(t))\dot{\mathbf{q}}_i(t)$ with respect to time and extracting $\ddot{\mathbf{q}}_i(t) = [\mathbf{J}_i(\mathbf{q}_i(t))]^+([\ddot{\mathbf{p}}_e^T(t), \dot{\omega}_e^T(t)]^T - \dot{\mathbf{J}}_i(\mathbf{q}_i(t))\dot{\mathbf{q}}_i(t))$. Movement and rotation of the object in Cartesian coordinates $[\ddot{\mathbf{p}}_e^T(t), \dot{\omega}_e^T(t)]^T$ are available; however, $\ddot{\mathbf{q}}_i(t)$ is unknown vector with regard to the presence of error in controllers. Therefore, minimum trajectory-tracking error is necessary to validate the results of computation. With regard to high accuracy tracking of the SDRE controller, the proposed assumption will be satisfied.

Obtaining the pattern of load distribution between robots, controller design of entire system is pursued. The dynamic equation of entire system is in the form of

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{b}(\dot{\mathbf{q}}) = \mathbf{u} + \mathbf{J}^T(\mathbf{q})\bar{\mathbf{f}}_e, \tag{10}$$

www.manaraa.com

where $\mathbf{q}(t) = \{\mathbf{q}_1(t), ..., \mathbf{q}_m(t)\}$, $\mathbf{u}(t) = \{\mathbf{u}_1(t), ..., \mathbf{u}_m(t)\}$, $\bar{\mathbf{f}}_e(t) = \{\mathbf{f}_{e,1}(t), ..., \mathbf{f}_{e,m}(t)\}$, and

$$\mathbf{M}(\mathbf{q}(t)) = \begin{bmatrix} \mathbf{M}_1(\mathbf{q}_1(t)) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{M}_i(\mathbf{q}_i(t)) & \mathbf{0} & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{M}_m(\mathbf{q}_m(t)) \end{bmatrix}_{mn \times mn},$$

$$\mathbf{c}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = \begin{bmatrix} \mathbf{c}_1(\mathbf{q}_1(t), \dot{\mathbf{q}}_1(t)) \\ \vdots \\ \mathbf{c}_m(\mathbf{q}_m(t), \dot{\mathbf{q}}_m(t)) \end{bmatrix}_{mn \times 1}, \quad \mathbf{g}(\mathbf{q}(t)) = \begin{bmatrix} \mathbf{g}_1(\mathbf{q}_1(t)) \\ \vdots \\ \mathbf{g}_m(\mathbf{q}_m(t)) \end{bmatrix}_{mn \times 1},$$

$$\mathbf{b}(\dot{\mathbf{q}}(t)) = \begin{bmatrix} \mathbf{b}_1(\dot{\mathbf{q}}_1(t)) \\ \vdots \\ \mathbf{b}_m(\dot{\mathbf{q}}_m(t)) \end{bmatrix}_{mn \times 1},$$

$$\mathbf{J}(\mathbf{q}(t)) = \begin{bmatrix} \mathbf{J}_1(\mathbf{q}_1(t)) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{J}_i(\mathbf{q}_i(t)) & \mathbf{0} & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{J}_m(\mathbf{q}_m(t)) \end{bmatrix}_{6m \times mn}.$$

2.3. Dynamics of flexible joint arms

The flexibility in joints is only assumed in the direction of actuation. Considering this assumption, DoF of system will be doubled in size. The schematic view of a system of flexible-joint robots with either prismatic or revolute joint is presented in Fig. 2. Flexibility of joints in mathematical models might have come from a spring between motor and link of a real system which is rare in industrial mechanisms and robots. However, the common point of view for this consideration is replacement of the effects of connections, gearbox, and magnetic adhesion of motors in real systems with an equal mass and spring in theoretical models.[22] The jth rotor's moment of inertia of ith robot, and the same address for mass are named as $J_{i,j}$ and $m^J_{i,j}$; spring constant of that is also $\tilde{k}_{i,j}$.

Generalized coordinates of ith arm with flexible joints are presented as $\mathbf{q}_i(t) = \{\mathbf{q}^l_i(t), \mathbf{q}^m_i(t)\}$ in which $\mathbf{q}^l_i(t) = \{q_{i,1}(t), ..., q_{i,n}(t)\}$ is related to links and $\mathbf{q}^m_i(t) = \{q_{i,n+1}(t), ..., q_{i,2n}(t)\}$ is concerned with actuators. Dimension of generalized coordinate is doubled since an independent motion is considered for each one of the actuators. Hence, two coupled differential equations present the dynamics of the system:

$$\mathbf{M}_i(\mathbf{q}^l_i)\ddot{\mathbf{q}}^l_i + \mathbf{c}_i(\mathbf{q}^l_i, \dot{\mathbf{q}}^l_i) + \mathbf{g}_i(\mathbf{q}^l_i) + \mathbf{b}_i(\dot{\mathbf{q}}^l_i) + \bar{\mathbf{k}}_i(\mathbf{q}^l_i, \mathbf{q}^m_i) = \mathbf{J}^T_i(\mathbf{q}^l_i)\mathbf{f}_{e,i}, \tag{11}$$

$$\text{diag}(J_{i,1}, ..., m^J_{i,j}, ..., J_{i,n})\ddot{\mathbf{q}}^m_i - \bar{\mathbf{k}}_i(\mathbf{q}^l_i, \mathbf{q}^m_i) = \mathbf{u}_i, \tag{12}$$

where

$$\bar{\mathbf{k}}_i(\mathbf{q}^l_i(t), \mathbf{q}^m_i(t)) = \begin{bmatrix} \tilde{k}_{i,1}(q_{i,1}(t) - q_{i,n+1}(t)) \\ \vdots \\ \tilde{k}_{i,n}(q_{i,n}(t) - q_{i,2n}(t)) \end{bmatrix},$$

www.manaraa.com

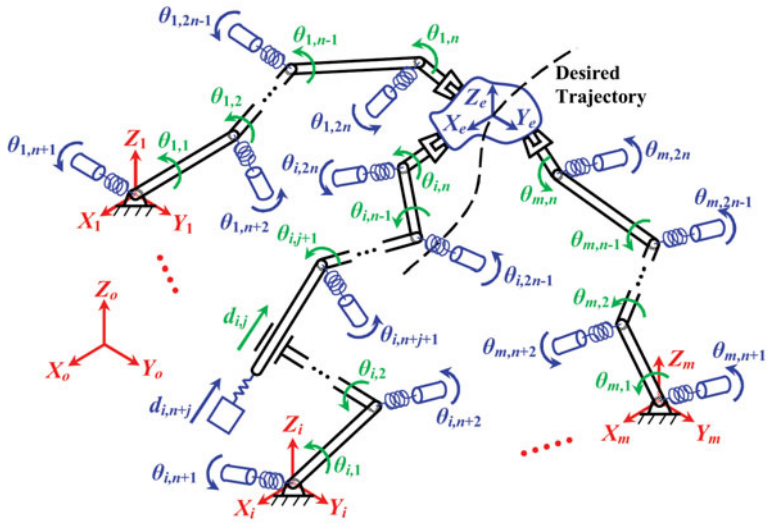


Fig. 2. Schematic view of m flexible-joint manipulators as a cooperative system.

is coupling vector of the two equations. Matrix and vectors in Eq. (11) are similar to the ones in rigid joint case; therefore, it is not necessary to execute Lagrange method (to derive dynamics) for flexible joint model. In order for applying the OLD method, Eq. (12) must be substituted in Eq. (11) by omitting $\bar{\mathbf{k}}_i(\mathbf{q}_i^l(t), \mathbf{q}_i^m(t))$:

$$\mathbf{M}_i(\mathbf{q}_i^l)\ddot{\mathbf{q}}_i^l + \mathbf{c}_i(\mathbf{q}_i^l, \dot{\mathbf{q}}_i^l) + \mathbf{g}_i(\mathbf{q}_i^l) + \mathbf{b}_i(\dot{\mathbf{q}}_i^l) + \text{diag}(J_{i,1}, ..., m_{i,j}^J, ..., J_{i,n})\ddot{\mathbf{q}}_i^m = \mathbf{J}_i^T(\mathbf{q}_i^l)\mathbf{f}_{e,i} + \mathbf{u}_i. \quad (13)$$

Vector $\mathbf{f}_e(t)$ should be divided between arms ($\mathbf{f}_{e,i}(t)$) to define load distribution via Lagrange multipliers method; as a result, $\mathbf{f}_{e,i}(t)$ is extracted from Eq. (13):

$$\mathbf{f}_{e,i} = \left[\mathbf{J}_i^T(\mathbf{q}_i^l)\right]^+ \times \left(\mathbf{M}_i(\mathbf{q}_i^l)\ddot{\mathbf{q}}_i^l + \mathbf{c}_i(\mathbf{q}_i^l, \dot{\mathbf{q}}_i^l) + \mathbf{g}_i(\mathbf{q}_i^l) + \mathbf{b}_i(\dot{\mathbf{q}}_i^l) + \text{diag}(J_{i,1}, ..., m_{i,j}^J, ..., J_{i,n})\ddot{\mathbf{q}}_i^m - \mathbf{u}_i\right). \quad (14)$$

Shaping the Lagrangian

$$L(\mathbf{u}, \gamma) = \frac{1}{2}\sum_{i=1}^{m}\mathbf{u}_i^T\bar{\mathbf{Q}}_i\mathbf{u}_i + \gamma^T\left\{\sum_{i=1}^{m}\left[\mathbf{J}_i^T(\mathbf{q}_i^l)\right]^+\begin{pmatrix}\mathbf{M}_i(\mathbf{q}_i^l)\ddot{\mathbf{q}}_i^l + \mathbf{c}_i(\mathbf{q}_i^l, \dot{\mathbf{q}}_i^l) + \mathbf{g}_i(\mathbf{q}_i^l) + \mathbf{b}_i(\dot{\mathbf{q}}_i^l)\\ +\text{diag}(J_{i,1}, ..., m_{i,j}^J, ..., J_{i,n})\ddot{\mathbf{q}}_i^m - \mathbf{u}_i\end{pmatrix} - \mathbf{f}_{e,i}\right\}, \quad (15)$$

and employing Eq. (6), results in control law

$$\mathbf{u}_i(t) = \bar{\mathbf{Q}}_i^{-1}\left[\left(\mathbf{J}_i^T(\mathbf{q}_i(t))\right)^+\right]^T\gamma(t). \quad (16)$$

Substituting Eq. (16) into Eq. (14) provides $\gamma(t)$ as

$$\gamma = \left(\sum_{i=1}^{m}\left[\mathbf{J}_i^T(\mathbf{q}_i^l)\right]^+ \times \bar{\mathbf{Q}}_i^{-1}\left[\left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+\right]^T\right)^{-1} \times \left[\left\{\sum_{i=1}^{m}\left[\mathbf{J}_i^T(\mathbf{q}_i^l)\right]^+\begin{pmatrix}\mathbf{M}_i(\mathbf{q}_i^l)\ddot{\mathbf{q}}_i^l + \mathbf{c}_i(\mathbf{q}_i^l, \dot{\mathbf{q}}_i^l) + \mathbf{g}_i(\mathbf{q}_i^l)\\ +\mathbf{b}_i(\dot{\mathbf{q}}_i^l) + \text{diag}(J_{i,1}, ..., m_{i,j}^J, ..., J_{i,n})\ddot{\mathbf{q}}_i^m\end{pmatrix}\right\} - \mathbf{f}_e\right], \quad (17)$$

www.manaraa.com

and finally, distributed vector for each arm is computable by

$$\mathbf{f}_{e,i} = \left[\mathbf{J}_i^T(\mathbf{q}_i^l)\right]^+ \begin{pmatrix} \mathbf{M}_i(\mathbf{q}_i^l)\ddot{\mathbf{q}}_i^l + \mathbf{c}_i(\mathbf{q}_i^l, \dot{\mathbf{q}}_i^l) + \mathbf{g}_i(\mathbf{q}_i^l) \\ +\mathbf{b}_i(\dot{\mathbf{q}}_i^l) + \text{diag}(J_{i,1}, ..., m_{i,j}^J, ..., J_{i,n})\ddot{\mathbf{q}}_i^m - \bar{\mathbf{Q}}_i^{-1}\left[\left(\mathbf{J}_i^T(\mathbf{q}_i)\right)^+\right]^T \gamma \end{pmatrix}. \tag{18}$$

Dynamics of entire system is presented as

$$\tilde{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \tilde{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}) + \tilde{\mathbf{g}}(\mathbf{q}) + \tilde{\mathbf{b}}(\dot{\mathbf{q}}) + \tilde{\mathbf{k}}(\mathbf{q}) = \mathbf{E}_u\mathbf{u} + \mathbf{E}_f\mathbf{J}^T(\mathbf{q})\bar{\mathbf{f}}_e, \tag{19}$$

in which modified variables are $\mathbf{q}(t) = \{\mathbf{q}_1^l(t), ..., \mathbf{q}_m^l(t), \mathbf{q}_1^m(t), ..., \mathbf{q}_m^m(t)\}$, $\mathbf{u}(t) = \{\mathbf{u}_1(t), ..., \mathbf{u}_m(t)\}$, and $\bar{\mathbf{f}}_e(t) = \{\mathbf{f}_{e,1}(t), ..., \mathbf{f}_{e,m}(t)\}$, but $\mathbf{J}(\mathbf{q}(t))$ is similar to rigid-joint case. Matrices $\mathbf{E}_u = \begin{bmatrix}\mathbf{0}_{mn\times mn} \\ \mathbf{I}_{mn\times mn}\end{bmatrix}$ and $\mathbf{E}_f = \begin{bmatrix}\mathbf{I}_{mn\times mn} \\ \mathbf{0}_{mn\times mn}\end{bmatrix}$ provide the connections between inputs, external force, and dynamics of system. Matrix and vectors of the system are also as follows:

$$\tilde{\mathbf{M}}(\mathbf{q}(t)) = \begin{bmatrix}\mathbf{M}_{mn\times mn}(\mathbf{q}(t)) & \mathbf{0} \\ \mathbf{0} & \text{diag}(J_{i,1}, ..., m_{i,j}^J, ..., J_{i,n}, ..., J_{m,1}, ..., m_{m,j}^J, ..., J_{m,n})\end{bmatrix}_{2mn\times 2mn},$$

$$\tilde{\mathbf{c}}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = \begin{bmatrix}\mathbf{c}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \\ \mathbf{0}\end{bmatrix}_{2mn\times 1}, \quad \tilde{\mathbf{g}}(\mathbf{q}(t)) = \begin{bmatrix}\mathbf{g}(\mathbf{q}(t)) \\ \mathbf{0}\end{bmatrix}_{2mn\times 1}, \quad \tilde{\mathbf{b}}(\dot{\mathbf{q}}(t)) = \begin{bmatrix}\mathbf{b}(\dot{\mathbf{q}}(t)) \\ \mathbf{0}\end{bmatrix}_{2mn\times 1},$$

$$\tilde{\mathbf{k}}(\dot{\mathbf{q}}(t)) = \begin{bmatrix} \bar{\mathbf{k}}_1(\mathbf{q}_1^l(t), \mathbf{q}_1^m(t)) \\ \vdots \\ \bar{\mathbf{k}}_m(\mathbf{q}_m^l(t), \mathbf{q}_m^m(t)) \\ ---------- \\ -\bar{\mathbf{k}}_1(\mathbf{q}_1^l(t), \mathbf{q}_1^m(t)) \\ \vdots \\ -\bar{\mathbf{k}}_m(\mathbf{q}_m^l(t), \mathbf{q}_m^m(t)) \end{bmatrix}_{2mn\times 1}.$$

3. Structure of the SDRE Controller

3.1. State-space form and SDC parameterization of rigid joint case

According to Eq. (10), the state vector of cooperative system is considered

$$\mathbf{x}(t) = [\mathbf{q}_1^T(t), ..., \mathbf{q}_m^T(t), \dot{\mathbf{q}}_1^T(t), ..., \dot{\mathbf{q}}_m^T(t)]_{2mn\times 1}^T, \tag{20}$$

in which $\mathbf{q}_i(t) = [\theta_{i,1}(t), ..., d_{i,j}(t), ..., \theta_{i,n}(t)]^T$. The state vector (20), leads to state-space representation of system:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{q}}_1(t) \\ \vdots \\ \dot{\mathbf{q}}_m(t) \\ ---------- \\ \mathbf{M}_1^{-1}(\mathbf{q}_1)\left\{\mathbf{u}_1 + \mathbf{J}_1^T(\mathbf{q}_1)\mathbf{f}_{e,1} - \mathbf{c}_1(\mathbf{q}_1, \dot{\mathbf{q}}_1) - \mathbf{g}_1(\mathbf{q}_1) - \mathbf{b}_1(\dot{\mathbf{q}}_1)\right\} \\ \vdots \\ \mathbf{M}_m^{-1}(\mathbf{q}_m)\left\{\mathbf{u}_m + \mathbf{J}_m^T(\mathbf{q}_m)\mathbf{f}_{e,m} - \mathbf{c}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m) - \mathbf{g}_m(\mathbf{q}_m) - \mathbf{b}_m(\dot{\mathbf{q}}_m)\right\} \end{bmatrix}_{2mn\times 1}. \tag{21}$$

www.manaraa.com

The state-dependent coefficient parameterization must be done to prepare $\mathbf{A}(\mathbf{x}(t))$ and $\mathbf{B}(\mathbf{x}(t))$ matrices, to reach the standard form of $\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t)$. Regarding friction of joints in dynamics, and increased dimension of system, different SDC matrices were obtained rather than refs. [21, 22]. Two viewpoints exist for SDC design: (1) computing $\mathbf{A}_i(\mathbf{x}(t))$ and $\mathbf{B}_i(\mathbf{x}(t))$ for each arm and therefore obtaining independent gain $\mathbf{K}_i(\mathbf{x}(t))$ for each one of the robots; (2) structuring $\mathbf{A}(\mathbf{x}(t))$ and $\mathbf{B}(\mathbf{x}(t))$ for entire system and designing one control gain $\mathbf{K}(\mathbf{x}(t))$ for the whole system. The relations of $\mathbf{A}_i(\mathbf{x}(t))$, $\mathbf{B}_i(\mathbf{x}(t))$, and $\mathbf{K}_i(\mathbf{x}(t))$ with $\mathbf{A}(\mathbf{x}(t))$, $\mathbf{B}(\mathbf{x}(t))$, and $\mathbf{K}(\mathbf{x}(t))$ are investigated.

The SDC matrices for case (1) are as follows:

$$\mathbf{A}_i(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{I}_{n\times n} \\ \mathbf{0}_{n\times n} & -\mathbf{M}_i^{-1}(\mathbf{x}(t))\left[\bar{\mathbf{C}}_i(\mathbf{x}(t)) + \operatorname{diag}(b_{i,1}^v, \ldots, b_{i,n}^v)\right] \end{bmatrix}_{2n\times 2n}, \tag{22}$$

$$\mathbf{B}_i(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{n\times n} \\ \mathbf{M}_i^{-1}(\mathbf{x}(t)) \end{bmatrix}_{2n\times n}, \tag{23}$$

where $\bar{\mathbf{C}}_i(\mathbf{x}(t)) = \bar{\mathbf{C}}_i(\mathbf{q}_i(t), \dot{\mathbf{q}}_i(t))$ is found from $\mathbf{c}_i(\mathbf{q}_i(t), \dot{\mathbf{q}}_i(t)) = \bar{\mathbf{C}}_i(\mathbf{q}_i(t), \dot{\mathbf{q}}_i(t))\dot{\mathbf{q}}_i(t)$. The terms that cannot be factored in SDC matrices are added to control law:

$$\mathbf{u}_i^{add}(t) = \mathbf{g}_i(\mathbf{x}(t)) + \mathbf{b}_i^{SDC}(\mathbf{x}(t)) - \mathbf{J}_i^T(\mathbf{x}(t))\mathbf{f}_{e,i}(t), \tag{24}$$

in which

$$\mathbf{b}_i^{SDC}(\mathbf{x}(t)) = \begin{bmatrix} \operatorname{sgn}(\dot{q}_{i,1}(t))\left[b_{i,1}^d + (b_{i,1}^s - b_{i,1}^d)\exp\left(\frac{-|\dot{q}_{i,1}(t)|}{\varepsilon}\right)\right] \\ \vdots \\ \operatorname{sgn}(\dot{q}_{i,n}(t))\left[b_{i,n}^d + (b_{i,n}^s - b_{i,n}^d)\exp\left(\frac{-|\dot{q}_{i,n}(t)|}{\varepsilon}\right)\right] \end{bmatrix}. \tag{25}$$

The reason for inserting $\mathbf{g}_i(\mathbf{x}(t))$, $\mathbf{b}_i^{SDC}(\mathbf{x}(t))$, and $\mathbf{J}_i^T(\mathbf{x}(t))\mathbf{f}_{e,i}(t)$ to input law is incapability of factorization. Eligibility of this operation was analyzed and tested several times i.e. refs. [21, 22]. Dividing the arrays to states is another way to propose the SDC matrices; however, it might lead to singularity when states possess zero value $\mathbf{x}(t) = \mathbf{0}$.[21]

The second case of SDC design (structuring $\mathbf{A}(\mathbf{x}(t))$ and $\mathbf{B}(\mathbf{x}(t))$ for entire system) provides a more complex presentation with different form. The SDC matrices for entire system are

$$\mathbf{A}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{mn\times mn} & \mathbf{I}_{mn\times mn} \\ \mathbf{0}_{mn\times mn} & \mathbf{W}^r(\mathbf{x}(t)) \end{bmatrix}_{2mn\times 2mn}, \tag{26}$$

$$\mathbf{B}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{mn\times mn} \\ \mathbf{M}^{-1}(\mathbf{x}(t)) \end{bmatrix}_{2mn\times mn}, \tag{27}$$

in which

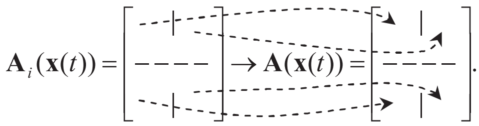
$$\mathbf{W}^r(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{W}_1^r(\mathbf{x}(t)) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{W}_i^r(\mathbf{x}(t)) & \mathbf{0} & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{W}_m^r(\mathbf{x}(t)) \end{bmatrix}, \tag{28}$$

www.manaraa.com

and $\mathbf{W}_i^r(\mathbf{x}(t)) = -\mathbf{M}_i^{-1}(\mathbf{x}(t))[\bar{\mathbf{C}}_i(\mathbf{x}(t)) + \mathrm{diag}(b_{i,1}^v, \ldots, b_{i,n}^v)]$ are held. The corrective terms of control law are also added to input vector:

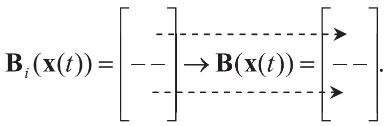
$$\mathbf{u}^{add}(t) = \begin{bmatrix} \mathbf{u}_1^{add}(t) \\ \vdots \\ \mathbf{u}_m^{add}(t) \end{bmatrix}. \tag{29}$$

Relation of sub-matrices of $\mathbf{A}(\mathbf{x}(t))$ and $\mathbf{B}(\mathbf{x}(t))$ do not simply follow diagonal form: $\mathbf{A}(\mathbf{x}(t)) \neq \mathrm{blockdiag}(\mathbf{A}_1(\mathbf{x}(t)), \ldots, \mathbf{A}_m(\mathbf{x}(t)))$ and $\mathbf{B}(\mathbf{x}(t)) \neq \mathrm{blockdiag}(\mathbf{B}_1(\mathbf{x}(t)), \ldots, \mathbf{B}_m(\mathbf{x}(t)))$. If one partitions $\mathbf{A}_i(\mathbf{x}(t))$ and $\mathbf{A}(\mathbf{x}(t))$ to four sub-blocks, each sub-block of $\mathbf{A}_i(\mathbf{x}(t))$ is transferred to same part of $\mathbf{A}(\mathbf{x}(t))$ in diagonal pattern:



(30)

Similar to Eq. (30), relation of $\mathbf{B}_i(\mathbf{x}(t))$ and $\mathbf{B}(\mathbf{x}(t))$ can be presented as



(31)

For example, a system of two manipulators, each one three-DoF, results in

$$\mathbf{A}_i(\mathbf{x}(t)) = \left[\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \\ \hline \mathbf{0}_{3\times 3} & -\mathbf{M}_i^{-1}(\mathbf{x}(t))\left[\bar{\mathbf{C}}_i(\mathbf{x}(t)) + \mathrm{diag}(b_{i,1}^v, \ldots, b_{i,n}^v)\right] \end{array}\right]; i = 1, 2;$$

$$\Rightarrow \mathbf{A}(\mathbf{x}(t)) = \left[\begin{array}{c|c} \mathbf{0}_{6\times 6} & \mathbf{I}_{6\times 6} \\ \hline \mathbf{0}_{6\times 6} & \mathbf{T}_{6\times 6}(\mathbf{x}(t)) \end{array}\right],$$

where

$$\mathbf{T}_{6\times 6} = \begin{bmatrix} -\mathbf{M}_1^{-1}(\mathbf{x}(t))\left[\bar{\mathbf{C}}_1(\mathbf{x}(t)) + \mathrm{diag}(b_{1,1}^v, \ldots, b_{1,n}^v)\right] & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & -\mathbf{M}_2^{-1}(\mathbf{x}(t))\left[\bar{\mathbf{C}}_2(\mathbf{x}(t)) + \mathrm{diag}(b_{2,1}^v, \ldots, b_{2,n}^v)\right] \end{bmatrix}.$$

And the similar approach constructs:

$$\mathbf{B}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{6\times 6} \\ \mathbf{M}^{-1}(\mathbf{x}(t)) \end{bmatrix}_{12\times 6},$$

in which

$$\mathbf{M}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{M}_1(\mathbf{x}(t)) & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{M}_2(\mathbf{x}(t)) \end{bmatrix}.$$

3.2. State-space form and SDC parameterization of flexible joint case

The arrangements of state vector have more options in this case. Simplicity and systematic design were the main reasons for SDC parameterization of this section. With respect to Eq. (19), state vector

المنارة للاستشارات
www.manaraa.com

is selected as

$$\mathbf{x}(t) = [\left(\mathbf{q}_1^l(t)\right)^T, ..., \left(\mathbf{q}_m^l(t)\right)^T, \left(\mathbf{q}_1^m(t)\right)^T, ..., \left(\mathbf{q}_m^m(t)\right)^T, \\ \left(\dot{\mathbf{q}}_1^l(t)\right)^T, ..., \left(\dot{\mathbf{q}}_m^l(t)\right)^T, \left(\dot{\mathbf{q}}_1^m(t)\right)^T, ..., \left(\dot{\mathbf{q}}_m^m(t)\right)^T]_{4mn\times 1}^T, \tag{32}$$

where $\mathbf{q}_i^l(t)$ and $\mathbf{q}_i^m(t)$ are related to coordinates of links and actuators with respect. Equation (32) produces state-space representation of Eq. (19) in the form of

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{q}}_1^l(t) \\ \vdots \\ \dot{\mathbf{q}}_m^l(t) \\ \dot{\mathbf{q}}_1^m(t) \\ \vdots \\ \dot{\mathbf{q}}_m^m(t) \\ \hline \mathbf{M}_1^{-1}(\mathbf{q}_1^l)\left(\mathbf{J}_1^T(\mathbf{q}_1^l)\mathbf{f}_{e,1} - \mathbf{c}_1(\mathbf{q}_1^l, \dot{\mathbf{q}}_1^l) - \mathbf{g}_1(\mathbf{q}_1^l) - \mathbf{b}_1(\dot{\mathbf{q}}_1^l) - \bar{\mathbf{k}}_1(\mathbf{q}_1^l, \mathbf{q}_1^m)\right) \\ \vdots \\ \mathbf{M}_m^{-1}(\mathbf{q}_m^l)\left(\mathbf{J}_m^T(\mathbf{q}_m^l)\mathbf{f}_{e,m} - \mathbf{c}_m(\mathbf{q}_m^l, \dot{\mathbf{q}}_m^l) - \mathbf{g}_m(\mathbf{q}_m^l) - \mathbf{b}_m(\dot{\mathbf{q}}_m^l) - \bar{\mathbf{k}}_m(\mathbf{q}_m^l, \mathbf{q}_m^m)\right) \\ \mathrm{diag}\left(\frac{1}{J_{1,1}}, ..., \frac{1}{m_{1,j}^J}, ..., \frac{1}{J_{1,n}}\right)\left(\mathbf{u}_1 + \bar{\mathbf{k}}_1(\mathbf{q}_1^l, \mathbf{q}_1^m)\right) \\ \vdots \\ \mathrm{diag}\left(\frac{1}{J_{m,1}}, ..., \frac{1}{m_{m,j}^J}, ..., \frac{1}{J_{m,n}}\right)\left(\mathbf{u}_m + \bar{\mathbf{k}}_m(\mathbf{q}_m^l, \mathbf{q}_m^m)\right) \end{bmatrix}_{4mn\times 1}. \tag{33}$$

Similar methods to Section 3.1 are considered to separate Eq. (33) in apparent linearization (or SDC) form. First case has following structure:

$$\mathbf{A}_i(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{2n\times 2n} & \mathbf{I}_{2n\times 2n} \\ \mathbf{W}_i^{f1}(\mathbf{x}(t)) & \mathbf{W}_i^{f2}(\mathbf{x}(t)) \end{bmatrix}_{4n\times 4n}, \tag{34}$$

$$\mathbf{B}_i(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{3n\times n} \\ \mathrm{diag}\left(\frac{1}{J_{i,1}}, ..., \frac{1}{m_{i,j}^J}, ..., \frac{1}{J_{i,n}}\right) \end{bmatrix}_{4n\times n}, \tag{35}$$

where

$$\mathbf{W}_i^{f1}(\mathbf{x}(t)) \\ = \left[\begin{array}{c|c} -\mathbf{M}_i^{-1}(\mathbf{q}_i^l(t))\mathrm{diag}\left(\tilde{k}_{i,1}, ..., \tilde{k}_{i,n}\right) & \mathbf{M}_i^{-1}(\mathbf{q}_i^l(t))\mathrm{diag}\left(\tilde{k}_{i,1}, ..., \tilde{k}_{i,n}\right) \\ \hline \mathrm{diag}\left(\frac{\tilde{k}_{i,1}}{J_{i,1}}, ..., \frac{\tilde{k}_{i,j}}{m_{i,j}^J}, ..., \frac{\tilde{k}_{i,1}}{J_{i,n}}\right) & -\mathrm{diag}\left(\frac{\tilde{k}_{i,1}}{J_{i,1}}, ..., \frac{\tilde{k}_{i,j}}{m_{i,j}^J}, ..., \frac{\tilde{k}_{i,1}}{J_{i,n}}\right) \end{array}\right], \tag{36}$$

$$\mathbf{W}_i^{f2}(\mathbf{x}(t)) = \begin{bmatrix} -\mathbf{M}_i^{-1}(\mathbf{x}(t))\left[\bar{\mathbf{C}}_i(\mathbf{x}(t)) + \mathrm{diag}(b_{i,1}^v, ..., b_{i,n}^v)\right] & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \end{bmatrix}. \tag{37}$$

Additional terms for control law are similar to Eq. (24). Second case requires

$$\mathbf{A}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{2mn\times 2mn} & \mathbf{I}_{2mn\times 2mn} \\ \mathbf{W}^{f1}(\mathbf{x}(t)) & \mathbf{W}^{f2}(\mathbf{x}(t)) \end{bmatrix}_{4mn\times 4mn}, \tag{38}$$

www.manaraa.com

$$\mathbf{B}(\mathbf{x}(t)) = \left[\begin{array}{c} \mathbf{0}_{3mn\times mn} \\ \text{diag}\left(\frac{1}{J_{i,1}}, ..., \frac{1}{m^J_{i,j}}, ..., \frac{1}{J_{i,n}}, ..., \frac{1}{J_{m,1}}, ..., \frac{1}{m^J_{m,j}}, ..., \frac{1}{J_{m,n}}\right) \end{array} \right]_{4mn\times mn}, \tag{39}$$

in which

$$\mathbf{W}^{f1}(\mathbf{x}(t)) = \left[\begin{array}{c|c} -\mathbf{W}^k(\mathbf{x}(t)) & \mathbf{W}^k(\mathbf{x}(t)) \\ \hline \mathbf{W}^J & -\mathbf{W}^J \end{array} \right], \tag{40}$$

$$\mathbf{W}^{f2}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{W}^r(\mathbf{x}(t)) & \mathbf{0}_{mn\times mn} \\ \mathbf{0}_{mn\times mn} & \mathbf{0}_{mn\times mn} \end{bmatrix}, \tag{41}$$

$$\mathbf{W}^k(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{W}^k_1(\mathbf{x}(t)) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{W}^k_i(\mathbf{x}(t)) & \mathbf{0} & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{W}^k_m(\mathbf{x}(t)) \end{bmatrix}, \tag{42}$$

$$\mathbf{W}^J = \begin{bmatrix} \mathbf{W}^J_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{W}^J_i & \mathbf{0} & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{W}^J_m \end{bmatrix}, \tag{43}$$

and moreover, $\mathbf{W}^k_i(\mathbf{x}(t)) = \mathbf{M}^{-1}_i(\mathbf{q}^l_i(t))\text{diag}(\tilde{k}_{i,1}, ..., \tilde{k}_{i,n})$ and $\mathbf{W}^J_i = \text{diag}(\frac{\tilde{k}_{i,1}}{J_{i,1}}, ..., \frac{\tilde{k}_{i,j}}{m^J_{i,j}}, ..., \frac{\tilde{k}_{i,1}}{J_{i,n}})$ are held. The relations of sub-matrices are similar to the ones in Section 3.1.

3.3. Clustering of terms in control input

In this subsection, the effect of clustering of terms, Eq. (24), in optimal control law (51), and the effect of that on the optimality of cost function (50) are analyzed and discussed, since if important dynamics or non-linearities are being clustered and probably being cancelled through state-feedback, the resulting inputs may not be optimal. First, let us revisit the reason for excluding the terms $\mathbf{g}_i(\mathbf{x}(t))$, $\mathbf{b}^{\text{SDC}}_i(\mathbf{x}(t))$ and $-\mathbf{J}^T_i(\mathbf{x}(t))\mathbf{f}_{e,i}(t)$ from SDC matrices. The gravity, a part of friction vector, and Jacobian matrix do not have a term or state that could be factored to be fit in the SDC matrices (22) or (26), they are combinations of trigonometric functions most of the times. First problem is that even if we replace the "cosine" function with its Taylor series expansion form, we will face a shift in the control law when a constant remains out of SDC matrices; e.g.$\cos(x_1(t)) = 1 - \frac{x^2_1(t)}{2!} + \frac{x^4_1(t)}{4!} - \cdots$, where the factored form is $\cos(x_1(t))x_1(t) = (\frac{1}{x_1(t)} - \frac{x_1(t)}{2!} + \frac{x^3_1(t)}{4!} - \cdots)x_1(t)$, that produces $\frac{1}{x_1(t)}$ in the SDC form which enforces singularity situation at the origin. The second problem is the lack of accuracy when we use Taylor expansion.

Dividing arrays to states to shape the SDC matrices is always an option that has the chance of singularity in the design at the origin $\mathbf{x}(0) = \mathbf{0}$. Here in this research, the control problem is limited to trajectory tracking and as a result, we can use this kind of SDC parameterization when we are sure the trajectory in joint space of the robot does not meet the origin for each link. This situation ($\mathbf{x}(0) \neq \mathbf{0}$) is very common and may happen in lots of trajectories. So the following matrix is introduced to fully include all the dynamics in the SDC design for checking the optimality of the proposed SDC

www.manaraa.com

in Eq. (22):

$$\mathbf{A}_i^{\text{opt}}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{I}_{n\times n} \\ \Omega_{n\times n}(\mathbf{x}(t)) & -\mathbf{M}_i^{-1}(\mathbf{x}(t))\left[\bar{\mathbf{C}}_i(\mathbf{x}(t)) + \text{diag}(b_{i,1}^v, ..., b_{i,n}^v)\right] \end{bmatrix}_{2n\times 2n}, \tag{44}$$

in which

$$\Omega_{n\times n}(\mathbf{x}(t)) = \text{diag}\left(\frac{u_{i,1}^{\text{add}}(t)}{q_{i,1}(t)}, \cdots, \frac{u_{i,n}^{\text{add}}(t)}{q_{i,n}(t)}\right) \tag{45}$$

holds and adds all the terms in SDC. This point of view expresses more design variations for cases that some states meet zero. In that event, we can divide the states to just one state that is not facing zero value; for example, if $x_1(t)$ was a safe state such that $x_1(t) \neq 0$, the $\Omega(\mathbf{x}(t))$ in Eq. (44) is desirable in the other form of

$$\Omega_{n\times n}(\mathbf{x}(t)) = \begin{bmatrix} \frac{u_{i,1}^{\text{add}}(t)}{q_{i,1}(t)} & 0 & \cdots & 0 \\ \frac{u_{i,2}^{\text{add}}(t)}{q_{i,1}(t)} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{u_{i,n}^{\text{add}}(t)}{q_{i,1}(t)} & 0 & 0 & 0 \end{bmatrix}. \tag{46}$$

This SDC design in Eq. (44) is optimal and the effect of gravity, friction, and external force is weighed by weighting matrices $\mathbf{Q}(\mathbf{x}(t))$ and $\mathbf{R}(\mathbf{x}(t))$. On the contrary, the control gain resulting from SDC designs (22) and (26) neglects gravity, friction, and external force, and these terms are added to control law (51). This action makes the control suboptimal and balances the non-linearities in dynamical system without any weight. Concluding remarks are as follows:

- The tuning of the optimal SDC (44) is harder than the general SDC form (22) or (26), since more terms are presented the SDC matrices.
- Gravity, friction, and the external force are functions of joint space variation (linear or angular movement) not velocities and weighting them increase the input torque or force.
- The optimal SDC has the problem of singularity and the form is not general, moreover, the trajectory tracking case is only executable. So the point-to-point motion (regulation) is out of reach with optimal SDC design.
- Excluding gravity, friction, and the external force from SDC and adding them to control law (such as Eq. (56)) changes the controller from optimal to suboptimal state.
- The control signals based on optimal SDC start from zero with a sudden jump to necessary value of control law; however, the control signals of Eq. (56) start from the necessary value of control law, $\mathbf{u}^{\text{add}}(t)$, without any jump.

More analysis on optimality will be presented in simulation section, and two SDC designs which are presented in this work can be used based on the trajectory in joint space.

3.4. Formulation of the SDRE controller

State-space equation of systems must be separated in the following from to design the controller:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)), \tag{47}$$

where $\mathbf{x}(t) \in \Re^{\bar{n}}$ is state vector of system, $\mathbf{u}(t) \in \Re^{\bar{m}}$ is input vector, $\mathbf{f}(\mathbf{x}(t)) : \Re^{\bar{n}} \to \Re^{\bar{n}}$, and $\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) : \Re^{\bar{n}} \times \Re^{\bar{m}} \to \Re^{\bar{n}}$ are held in which $\bar{n}$ denotes number of states and $\bar{m}$ represents number of inputs and t stands for time. Initial condition is $\mathbf{x}(0) = \mathbf{x}_0$ and equilibrium point is defined as $\mathbf{f}(\mathbf{0}) = \mathbf{0}$. $\mathbf{f}(\mathbf{x}(t))$ and $\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))$ are vector-valued functions, smooth piecewise continuous which satisfy the Lipschitz condition. $\mathbf{f}(\mathbf{x}(t))$ and $\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))$ do not have any explicit terms of t (time), although they might have several non-linear functions of $\mathbf{x}(t)$. Next step of design is state-dependent

www.manaraa.com

coefficient parameterization

$$\mathbf{f}(\mathbf{x}(t)) = \mathbf{A}(\mathbf{x}(t))\mathbf{x}(t), \tag{48}$$

$$\mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t), \tag{49}$$

where $\mathbf{A}(\mathbf{x}(t)) : \Re^{\bar{n}} \to \Re^{\bar{n}\times\bar{n}}$ and $\mathbf{B}(\mathbf{x}(t)) : \Re^{\bar{n}} \to \Re^{\bar{n}\times\bar{m}}$. The SDC forms for two cases of rigid and flexible joint arm was presented in Sections 3.1 and 3.2. Proper selection of $\bar{n}$ and $\bar{m}$ is necessary for the rest of the design.

The pair of $\{\mathbf{A}(\mathbf{x}(t)), \mathbf{B}(\mathbf{x}(t))\}$ is a controllable (stabilizable) parameterization of non-linear system (47) for all $\mathbf{x}(t)$ in $t \in \Re^+$, if this pair is controllable (stabilizable) in linear sense. In other words, pair of $\{\mathbf{A}(\mathbf{0}), \mathbf{B}(\mathbf{0})\}$ must satisfy controllability condition provided that any terms will not violate that in $t \in \Re^+$.[21,22] Considering proposed assumption, rank of controllability matrix

$$\mathbf{M}_C = [\ \mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}\ ],$$

must be full to warrant the solvability of the Riccati equation.

The optimal control problem is defined by minimizing of the cost function presented in the following integral:

$$J = \frac{1}{2}\int_0^\infty \{\mathbf{x}^T(t)\mathbf{Q}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}(\mathbf{x}(t))\mathbf{u}(t)\}\mathrm{d}t, \tag{50}$$

where $\mathbf{Q}(\mathbf{x}(t)) : \Re^{\bar{n}} \to \Re^{\bar{n}\times\bar{n}}$ is weighting matrix for states and $\mathbf{R}(\mathbf{x}(t)) : \Re^{\bar{n}} \to \Re^{\bar{m}\times\bar{m}}$ is weighting matrix for inputs. The weighting matrix for states has to be set in a way to satisfy the observability condition for the pair of $\{\mathbf{A}(\mathbf{x}(t)), \mathbf{Q}^{1/2}(\mathbf{x}(t))\}$ as a detectable parameterization of system (47) for all $\mathbf{x}(t)$ in $t \in \Re^+$. In a similar way, pair of $\{\mathbf{A}(\mathbf{0}), \mathbf{Q}^{1/2}(\mathbf{0})\}$ must be observable (detectable) without any cause to violate that in $t \in \Re^+$. Regarding the aforementioned assumption, observability matrix

$$\mathbf{M}_O = \begin{bmatrix} \mathbf{Q}^{1/2} \\ \mathbf{Q}^{1/2}\mathbf{A} \\ \mathbf{Q}^{1/2}\mathbf{A}^2 \\ \vdots \\ \mathbf{Q}^{1/2}\mathbf{A}^{\bar{n}-1} \end{bmatrix}.$$

must be fully ranked.

By constructing the Hamiltonian as follows:

$$H(\mathbf{x}(t), \mathbf{u}(t), \lambda(t)) = \frac{1}{2}\left(\mathbf{x}^T(t)\mathbf{Q}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}(\mathbf{x}(t))\mathbf{u}(t)\right) + \lambda^T(t)\left(\mathbf{A}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))\right),$$

and using the necessary optimality condition with $\lambda(t) = \mathbf{K}(\mathbf{x}(t))\mathbf{x}(t)$, one can find the control law as in the standard form of

$$\mathbf{u}(t) = -\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^T(\mathbf{x}(t))\mathbf{K}(\mathbf{x}(t))\mathbf{x}(t). \tag{51}$$

Using the other optimality condition,

$$\frac{\partial H(\mathbf{x}(t), \mathbf{u}(t), \lambda(t))}{\partial \mathbf{x}(t)} = -\dot{\lambda}(t),$$

www.manaraa.com

and mathematical operations, following equation is provided:

$$\dot{\lambda}(t) = -\mathbf{Q}(\mathbf{x}(t))\mathbf{x}(t) - \frac{1}{2}\mathbf{x}^T(t)\frac{\partial \mathbf{Q}(\mathbf{x}(t))}{\partial \mathbf{x}(t)} - \frac{1}{2}\mathbf{u}^T(t)\frac{\partial \mathbf{R}(\mathbf{x}(t))}{\partial \mathbf{x}(t)}\mathbf{u}(t) - \left[\frac{\partial\left(\mathbf{A}(\mathbf{x}(t))\mathbf{x}(t)\right)}{\partial \mathbf{x}(t)}\right]^T \lambda(t) - \left[\frac{\partial\left(\mathbf{B}(\mathbf{x}(t))\mathbf{u}(t)\right)}{\partial \mathbf{x}(t)}\right]^T \lambda(t).$$

By differentiation of $\lambda(t) = \mathbf{K}(\mathbf{x}(t))\mathbf{x}(t)$ with respect the time, one can get

$$\dot{\lambda}(t) = \dot{\mathbf{K}}(\mathbf{x}(t))\,\mathbf{x}(t) + \mathbf{K}(\mathbf{x}(t))\,\dot{\mathbf{x}}(t) = \dot{\mathbf{K}}(\mathbf{x}(t))\,\mathbf{x}(t) + \mathbf{K}(\mathbf{x}(t))\,\mathbf{A}(\mathbf{x}(t))\,\mathbf{x}(t) - \mathbf{K}(\mathbf{x}(t))\,\mathbf{B}(\mathbf{x}(t))\,\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^T(\mathbf{x}(t))\mathbf{K}(\mathbf{x}(t))\,\mathbf{x}(t).$$

Finally, equality of $\dot{\lambda}(t)$ results in

$$\dot{\mathbf{K}}(\mathbf{x})\,\mathbf{x} + \tfrac{1}{2}\mathbf{x}^T\tfrac{\partial \mathbf{Q}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{x} + \tfrac{1}{2}\mathbf{u}^T\tfrac{\partial \mathbf{R}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{u} + \left[\tfrac{\partial(\mathbf{A}(\mathbf{x})\mathbf{x})}{\partial \mathbf{x}}\right]^T \mathbf{K}(\mathbf{x})\,\mathbf{x} + \left[\tfrac{\partial(\mathbf{B}(\mathbf{x})\mathbf{u})}{\partial \mathbf{x}}\right]^T \mathbf{K}(\mathbf{x})\,\mathbf{x} + \begin{bmatrix}\mathbf{K}(\mathbf{x})\,\mathbf{A}(\mathbf{x}) + \mathbf{A}^T(\mathbf{x})\,\mathbf{K}(\mathbf{x}) \\ -\mathbf{K}(\mathbf{x})\,\mathbf{B}(\mathbf{x})\,\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^T(\mathbf{x})\mathbf{K}(\mathbf{x}) + \mathbf{Q}(\mathbf{x})\end{bmatrix}\mathbf{x} = \mathbf{0},$$

which can be divided to two sections: the SDRE which provides the suboptimal solution $\mathbf{K}(\mathbf{x}(t))$ given by

$$\mathbf{A}^T(\mathbf{x})\mathbf{K}(\mathbf{x}) + \mathbf{K}(\mathbf{x})\mathbf{A}(\mathbf{x}) - \mathbf{K}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^T(\mathbf{x})\mathbf{K}(\mathbf{x}) + \mathbf{Q}(\mathbf{x}) = \mathbf{0}, \tag{52}$$

and the following part which is so called "the necessary condition for optimality"

$$\dot{\mathbf{K}}(\mathbf{x}) + \tfrac{1}{2}\left(\tfrac{\partial \mathbf{Q}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{x}\right)^T - \tfrac{1}{2}\left(\tfrac{\partial \mathbf{R}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{u}\right)^T \mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^T(\mathbf{x})\mathbf{K}(\mathbf{x}) + \left[\tfrac{\partial(\mathbf{A}(\mathbf{x})\mathbf{x})}{\partial \mathbf{x}}\right]^T \mathbf{K}(\mathbf{x}) + \left[\tfrac{\partial(\mathbf{B}(\mathbf{x})\mathbf{u})}{\partial \mathbf{x}}\right]^T \mathbf{K}(\mathbf{x}) = \mathbf{0}.$$

The stability of the SDRE controller could be checked by Lyapunov method. Introducing the Lyapunov candidate as $V(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{x}^T(t)\mathbf{K}(\mathbf{x}(t))\mathbf{x}(t)$, and computing the time derivative of that

$$\dot{V}(\mathbf{x}(t), \mathbf{u}(t)) = \dot{\mathbf{x}}^T(t)\mathbf{K}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{x}^T(t)\mathbf{K}(\mathbf{x}(t))\dot{\mathbf{x}}(t) = \left(\mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K}\mathbf{x}\right)^T\mathbf{K}\mathbf{x} + \mathbf{x}^T\mathbf{K}\left(\mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K}\mathbf{x}\right),$$

result in (note that in SDRE case $\dot{\mathbf{K}}(\mathbf{x}(t))$ is zero):

$$\dot{V}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{x}^T\left(\mathbf{A}^T\mathbf{K} - \mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K} + \mathbf{K}\mathbf{A} - \mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K}\right)\mathbf{x},$$

and with substitution of $\mathbf{A}^T\mathbf{K} + \mathbf{K}\mathbf{A} - \mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K} = -\mathbf{Q}$, following expression is found:

$$\dot{V}(\mathbf{x}(t), \mathbf{u}(t)) = -\mathbf{x}^T\left(\mathbf{Q} + \mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K}\right)\mathbf{x}.$$

Considering that $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{K}$ are positive, and $\mathbf{B}\mathbf{B}^T$ is also positive, one could show that the negative derivative of Lyapunov candidate is satisfied $\dot{V}(\mathbf{x}(t), \mathbf{u}(t)) < 0$.

Two methods for proposing solutions are possible: solving the SDRE m-times for each arm independently and solving the SDRE once for entire system. The first case requires the input law for each robot as

$$\mathbf{u}_i(t) = -\mathbf{R}_i^{-1}(\mathbf{x}_i(t))\mathbf{B}_i^T(\mathbf{x}_i(t))\mathbf{K}_i(\mathbf{x}_i(t))\mathbf{x}_i(t) + \mathbf{u}_i^{\text{add}}(t), \tag{53}$$

www.manaraa.com

Table I. Dimensional comparison of two control approaches: independent subsystems and entire system.

(1): Independent arms	State vector	$\mathbf{x}_i = [\mathbf{q}_i^T, \dot{\mathbf{q}}_i^T]^T; [\mathbf{x}_i]_{\bar{n}\times 1}$
	System's matrices	$[\mathbf{A}_i(\mathbf{x}_i)]_{\bar{n}\times\bar{n}}; [\mathbf{B}_i(\mathbf{x}_i)]_{\bar{n}\times\bar{m}}$
	Weighting matrices	$[\mathbf{Q}_i(\mathbf{x}_i)]_{\bar{n}\times\bar{n}}; [\mathbf{R}_i(\mathbf{x}_i)]_{\bar{m}\times\bar{m}}$
	Control law	$\mathbf{u}_i = -\mathbf{R}_i^{-1}(\mathbf{x}_i)\mathbf{B}_i^T(\mathbf{x}_i)\mathbf{K}_i(\mathbf{x}_i)\mathbf{x}_i + \mathbf{u}_i^{\text{add}}; [\mathbf{u}_i]_{\bar{m}\times 1}$
	SDRE	$\mathbf{A}_i^T(\mathbf{x}_i)\mathbf{K}_i(\mathbf{x}_i) + \mathbf{K}_i(\mathbf{x}_i)\mathbf{A}_i(\mathbf{x}_i) - \mathbf{K}_i(\mathbf{x}_i)\mathbf{B}_i(\mathbf{x}_i)\mathbf{R}_i^{-1}(\mathbf{x}_i)\mathbf{B}_i^T(\mathbf{x}_i)\mathbf{K}_i(\mathbf{x}_i) + \mathbf{Q}_i(\mathbf{x}_i) = \mathbf{0}; [\mathbf{K}_i(\mathbf{x}_i)]_{\bar{n}\times\bar{n}}$
(2): Entire system	State vector	$\mathbf{x} = [\mathbf{q}_1^T, ..., \mathbf{q}_m^T, \dot{\mathbf{q}}_1^T, ..., \dot{\mathbf{q}}_m^T]^T; [\mathbf{x}]_{m\bar{n}\times 1}$
	System's matrices	$[\mathbf{A}(\mathbf{x})]_{m\bar{n}\times m\bar{n}}; [\mathbf{B}(\mathbf{x})]_{m\bar{n}\times m\bar{m}}$
	Weighting matrices	$[\mathbf{Q}(\mathbf{x})]_{m\bar{n}\times m\bar{n}}; [\mathbf{R}(\mathbf{x})]_{m\bar{m}\times m\bar{m}}$
	Control law	$\mathbf{u} = -\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^T(\mathbf{x})\mathbf{K}(\mathbf{x})\mathbf{x} + \mathbf{u}^{\text{add}}; [\mathbf{u}]_{m\bar{m}\times 1}$
	SDRE	$\mathbf{A}^T(\mathbf{x})\mathbf{K}(\mathbf{x}) + \mathbf{K}(\mathbf{x})\mathbf{A}(\mathbf{x}) - \mathbf{K}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{R}^{-1}(\mathbf{x})\mathbf{B}^T(\mathbf{x})\mathbf{K}(\mathbf{x}) + \mathbf{Q}(\mathbf{x}) = \mathbf{0}; [\mathbf{K}(\mathbf{x})]_{m\bar{n}\times m\bar{n}}$

where $\mathbf{x}_i(t)$ is the states of ith manipulator. The suboptimal gain is also obtained from

$$\begin{aligned}&\mathbf{A}_i^T(\mathbf{x}_i(t))\mathbf{K}_i(\mathbf{x}_i(t)) + \mathbf{K}_i(\mathbf{x}_i(t))\mathbf{A}_i(\mathbf{x}_i(t))\\ &\quad - \mathbf{K}_i(\mathbf{x}_i(t))\mathbf{B}_i(\mathbf{x}_i(t))\mathbf{R}_i^{-1}(\mathbf{x}_i(t))\mathbf{B}_i^T(\mathbf{x}_i(t))\mathbf{K}_i(\mathbf{x}_i(t)) + \mathbf{Q}_i(\mathbf{x}_i(t)) = \mathbf{0}.\end{aligned} \tag{54}$$

For rigid joint case $\bar{n} = 2n$ and for flexible joint arms $\bar{n} = 4n$ are set; for both cases, the number of actuators is similar $\bar{m} = n$. One can imagine a gain for entire system consisting of elements of $\mathbf{K}_i(\mathbf{x}_i(t))$ with zero non-block-diagonal arrays as

$$\mathbf{K}(\mathbf{x}(t)) = \begin{bmatrix} \mathbf{K}_1(\mathbf{x}_1(t)) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \mathbf{0} & \mathbf{K}_i(\mathbf{x}_i(t)) & \mathbf{0} & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{K}_m(\mathbf{x}_m(t)) \end{bmatrix}_{m\bar{n}\times m\bar{n}}. \tag{55}$$

This point of view operates for each manipulator independently and system has m subcontroller.

Second approach has SDC matrices for entire system and solves the SDRE only once with following control law:

$$\mathbf{u}(t) = -\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^T(\mathbf{x}(t))\mathbf{K}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{u}^{\text{add}}(t). \tag{56}$$

This point of view implements the optimal control on entire system uniformly.

Analogy of two methods was presented in Table I. First case needs computation and solving the Riccati equation m times which solvability of systems with respect to size of matrices is more probable. Second case is hard to solve for systems with more than four manipulators and solution might be more time consuming. The outputs of two methods are similar; however, for small system of cooperation, uniform structure is recommended and for large number of manipulators independent controllers seem suitable. In practical implementation, failure of one controller is not ruining the performance of all system in independent controller design. Therefore, independent controller design is recommended for experimental operations to preserve more safety for the system.

It can be observed that the SDRE equations, Eqs. (52) and (54), were used for the design as the steady state form of the state-dependent differential Riccati equation (SDDRE):

$$\begin{aligned}-\dot{\mathbf{K}}(\mathbf{x}(t)) &= \mathbf{A}^T(\mathbf{x}(t))\mathbf{K}(\mathbf{x}(t)) + \mathbf{K}(\mathbf{x}(t))\mathbf{A}(\mathbf{x}(t))\\ &\quad - \mathbf{K}(\mathbf{x}(t))\mathbf{B}(\mathbf{x}(t))\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^T(\mathbf{x}(t))\mathbf{K}(\mathbf{x}(t)) + \mathbf{Q}(\mathbf{x}(t)),\end{aligned}$$

based on the assumption that $t_f \to \infty$ and as a result $\dot{\mathbf{K}}(\mathbf{x}(t)) = \mathbf{0}$. The SDDRE has several advantages rather than the SDRE such as finite horizon control and providing a control gain considering transient effect of the SDC matrices. However, the solution to the SDDRE is more difficult that the SDRE.

www.manaraa.com

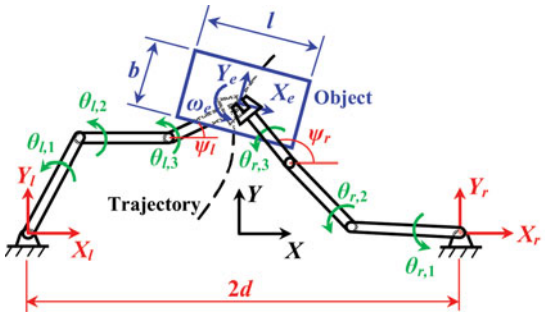


Fig. 3. Two planar manipulators as a cooperative system.

Table II. Specifications of the planar arms.

Description	Parameters	Value	Units
Length of left arm's links	$a_{l,1}, a_{l,2}, a_{l,3}$	1.5, 1, 0.25	m
Center of mass of left arm's links	$a_{cl,1}, a_{cl,2}, a_{cl,3}$	0.75, 0.5, 0.125	m
Mass of links, left arm	$m_{l,1}, m_{l,2}, m_{l,3}$	2, 1.5, 1	kg
Moment of inertia of links, left arm	$I_{zzl,1}, I_{zzl,2}, I_{zzl,3}$	0.375, 0.125, 0.00,520,833	kg.m^2
Distant between bases	d	1.5	m
Viscous friction for left arm	$b^v_{l,1}, b^v_{l,2}, b^v_{l,3}$	0.05, 0.04, 0.03	kg.m/s
Dynamic friction for left arm	$b^d_{l,1}, b^d_{l,2}, b^d_{l,3}$	0.005, 0.004, 0.003	kg.m^2/s^2
Static friction for left arm	$b^s_{l,1}, b^s_{l,2}, b^s_{l,3}$	0.0025, 0.002, 0.0015	kg.m^2/s^2
Small positive constant	ϵ	0.1	–
Gravity in Y-axis direction	g_0	9.81	m/s^2
Stall torque of left arm's motors	$u_{stall,l,1}, u_{stall,l,2}, u_{stall,l,3}$	100, 85, 70	N.m
No-load speed of left arm's motors	$\omega_{nl,l,1}, \omega_{nl,l,2}, \omega_{nl,l,3}$	5, 5.2, 5.5	rad/s
Similar parameters for right arm are considered			

Based on the situation that the trajectory tracking is just practiced in this current research and the cooperative manipulators models are so complicated and massive, the SDRE was selected as the controller.

4. Simulations

4.1. Cooperative system of three-DoF planar arms

Consider two rigid joint planar arms which carry a large object with mass and moment of inertia presented in Fig. 3. Specifications of the manipulators are expressed in Table II.

On the one hand, the third link or third DoF provides redundancy for each arm; and on the other hand, an additional DoF is needed for desired rotation of the object. Hence, the proposed structure is suitable for planar motion of an object with desired orientation, illustrated in Fig. 3.

The arms are symmetric and parameters of right arm are similar to the left one by substituting l to r index. To define inverse kinematics relations, the additional DoF of manipulator must be defined with $\theta_{l,1}(t) + \theta_{l,1}(t) + \theta_{l,3}(t) = \psi_l(t)$. Regarding this constraint, end point of second link is expressed as $x_{e,l,2}(t) = d + x_{e,l}(t) - a_{l,3}\cos(\psi_l(t))$ and $y_{e,l,2}(t) = y_{e,l}(t) - a_{l,3}\sin(\psi_l(t))$. The rest of calculation is similar to conventional two-DoF planar arms. Right arm inverse equations are available only by changing l to r index, except $x_{e,r,2}(t) = -d + x_{e,r}(t) - a_{r,3}\cos(\psi_r(t))$.

Generalized coordinate of the system is $\mathbf{q}_l(t) = \{\theta_{l,1}(t), \theta_{l,1}(t), \theta_{l,3}(t)\}$ and $\mathbf{q}_r(t) = \{\theta_{r,1}(t), \theta_{r,1}(t), \theta_{r,3}(t)\}$; setting $n = 3$, generated from Eq. (2) with l and r indices instead of i. State vector is set as $\mathbf{x}(t) = [\mathbf{q}_l^T(t), \mathbf{q}_r^T(t), \dot{\mathbf{q}}_l^T(t), \dot{\mathbf{q}}_r^T(t)]^T_{12\times1}$ based on Eq. (20), leads to state-space

www.manaraa.com

representation of system as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{q}}_l(t) \\ \dot{\mathbf{q}}_r(t) \\ \mathbf{M}_l^{-1}(\mathbf{q}_l)\left\{\mathbf{u}_l + \mathbf{J}_l^T(\mathbf{q}_l)\mathbf{f}_{e,l} - \mathbf{c}_l(\mathbf{q}_l, \dot{\mathbf{q}}_l) - \mathbf{g}_l(\mathbf{q}_l) - \mathbf{b}_l(\dot{\mathbf{q}}_l)\right\}, \\ \mathbf{M}_r^{-1}(\mathbf{q}_r)\left\{\mathbf{u}_r + \mathbf{J}_r^T(\mathbf{q}_r)\mathbf{f}_{e,r} - \mathbf{c}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r) - \mathbf{g}_r(\mathbf{q}_r) - \mathbf{b}_r(\dot{\mathbf{q}}_r)\right\}, \end{bmatrix}_{18\times 1}. \tag{57}$$

Restrictions of DC motors are applied through

$$u_{\max,i,j}(t) = u_{\text{stall},i,j} - \frac{u_{\text{stall},i,j}}{\omega_{nl,i,j}}\dot{q}_{i,j}(t), \qquad i = l, r;\ j = 1, 2, 3; \tag{58}$$

$$u_{\min,i,j}(t) = -u_{\text{stall},i,j} - \frac{u_{\text{stall},i,j}}{\omega_{nl,i,j}}\dot{q}_{i,j}(t), \qquad i = l, r;\ j = 1, 2, 3; \tag{59}$$

where $u_{\text{stall},i,j}$ is stall torque and $\omega_{nl,i,j}$ is no-load speed of motor of jth link of ith arm. Details of Eq. (57) are neglected such as elements of $\mathbf{M}_i(\mathbf{q}_i(t))$, $\mathbf{c}_i(\mathbf{q}_i(t), \dot{\mathbf{q}}_i(t))$ and $\mathbf{g}_i(\mathbf{q}_i(t))$. They are computable using Lagrange method. Friction vector $\mathbf{b}_i(\dot{\mathbf{q}}_i(t))$ is similar to Eq. (3) and Jacobian matrix for computing desired trajectory in joint space is as follows:

$$\mathbf{J}_{i,\text{des}}(\mathbf{q}_{i,\text{des}}(t)) = \begin{bmatrix} J_{i,\text{des}(1,1)}(\mathbf{q}_i(t)) & J_{i,\text{des}(1,2)}(\mathbf{q}_i(t)) & J_{i,\text{des}(1,3)}(\mathbf{q}_i(t)) \\ J_{i,\text{des}(2,1)}(\mathbf{q}_i(t)) & J_{i,\text{des}(2,2)}(\mathbf{q}_i(t)) & J_{i,\text{des}(2,3)}(\mathbf{q}_i(t)) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}_{6\times 3}, \tag{60}$$

where

$$\begin{aligned}
J_{i,\text{des},(1,1)}(\mathbf{q}_{i,\text{des}}) &= -a_{l,1}\sin(q_{l,\text{des},1}) - a_{l,2}\sin(q_{l,\text{des},1} + q_{l,\text{des},2}) - a_{l,3}\sin(q_{l,\text{des},1} + q_{l,\text{des},2} + q_{l,\text{des},3}), \\
J_{i,\text{des},(1,2)}(\mathbf{q}_{i,\text{des}}) &= -a_{l,2}\sin(q_{l,\text{des},1} + q_{l,\text{des},2}) - a_{l,3}\sin(q_{l,\text{des},1} + q_{l,\text{des},2} + q_{l,\text{des},3}), \\
J_{i,\text{des},(1,3)}(\mathbf{q}_{i,\text{des}}) &= -a_{l,3}\sin(q_{l,\text{des},1} + q_{l,\text{des},2} + q_{l,\text{des},3}), \\
J_{i,\text{des},(2,1)}(\mathbf{q}_{i,\text{des}}) &= a_{l,1}\cos(q_{l,\text{des},1}) + a_{l,2}\cos(q_{l,\text{des},1} + q_{l,\text{des},2}) + a_{l,\text{des},3}\cos(q_{l,\text{des},1} + q_{l,\text{des},2} + q_{l,\text{des},3}), \\
J_{i,\text{des},(2,2)}(\mathbf{q}_{i,\text{des}}) &= a_{l,2}\cos(q_{l,\text{des},1} + q_{l,\text{des},2}) + a_{l,3}\cos(q_{l,\text{des},1} + q_{l,\text{des},2} + q_{l,\text{des},3}), \\
J_{i,\text{des},(2,3)}(\mathbf{q}_{i,\text{des}}) &= a_{l,3}\cos(q_{l,\text{des},1} + q_{l,\text{des},2} + q_{l,\text{des},3}).
\end{aligned}$$

Representative vector of desired path is

$$\dot{\mathbf{q}}_{i,\text{des}}(t) = \mathbf{J}_{i,\text{des}}^{+}(\mathbf{q}_{i,\text{des}}(t)) \begin{bmatrix} \dot{X}_e(t) \\ \dot{Y}_e(t) \\ 0 \\ 0 \\ 0 \\ \omega_e(t) \end{bmatrix}, \tag{61}$$

in which $\dot{X}_e(t)$ is velocity component in X direction, $\dot{Y}_e(t)$ for Y, and $\omega_e(t)$ is desired angular velocity of the object around Z axis; moreover, with respect to dimension of $\mathbf{J}$, inverse of that can be computed by $\mathbf{J}^+ = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T$.

For computation of ith vector of external force, Eq. (4) is employed. With regard to dimension of Jacobian matrix, generalized inverse of $\mathbf{J}_i(\mathbf{q}_i(t))$ is available; however, the following term in Eq. (8):

$$\left(\sum_{i=1}^{m}\left(\mathbf{J}_i^T(\mathbf{q}_i(t))\right)^{+}\bar{\mathbf{Q}}_i^{-1}\left[\left(\mathbf{J}_i^T(\mathbf{q}_i(t))\right)^{+}\right]^T\right)^{-1}, \tag{62}$$

www.manaraa.com

is not invertible since there are so many zeroes in the midst of the matrix. So it is recommended to reduce Jacobian to a square form whenever that is possible. For n<6, is it possible to reduce the Jacobian matrix to square form. Reduced square form of Jacobian for this case study is

$$\mathbf{J}_i(\mathbf{q}_i(t)) = \begin{bmatrix} J_{i(1,1)}(\mathbf{q}_i(t)) & J_{i(1,2)}(\mathbf{q}_i(t)) & J_{i(1,3)}(\mathbf{q}_i(t)) \\ J_{i(2,1)}(\mathbf{q}_i(t)) & J_{i(2,2)}(\mathbf{q}_i(t)) & J_{i(2,3)}(\mathbf{q}_i(t)) \\ 1 & 1 & 1 \end{bmatrix}_{3\times 3}, \tag{63}$$

where the arrays of that are similar to Eq. (60) substituting $(\cdot)_{i,\text{des}}$ index with $(\cdot)_i$. As a result, external force provided by moving object is changed from Eq. (1) into

$$\mathbf{f}_e(t) = -\begin{bmatrix} m_p \ddot{X}_e(t) \\ m_p(\ddot{Y}_e(t) + g_0) \\ I_{zz,p}\dot{\omega}_e(t) + I_{zz,p}\omega_e^2(t) \end{bmatrix}, \tag{64}$$

in which $I_{zz,p} = m_p(b^2 + l^2)/12$. The SDC parameterization can be shaped using Eqs. (22) and (23).

Circular trajectory: Desired trajectory is a circle of 0.5m radius, placed in (0, 0) coordinates to be tracked in $t_f = 3\pi/2$s. Equation of the trajectory is $X_e(t) = 0.5\cos t$and $Y_e(t) = 0.5\sin t$. Rotation of 90° in counter-clockwise direction was defined as desired orientation, the angular velocity of the expression is in the form of $\omega_e(t) = \frac{160}{81\pi^4}t^4 - \frac{160}{27\pi^3}t^3 + \frac{120}{27\pi^2}t^2$. Configuration of arms with respect to system's coordinates suggests $\psi_l(t) = 0$ and $\psi_r(t) = \pi$rad as initial orientation of third links. This setting results in following initial conditions using inverse kinematics equations:

$$\mathbf{q}_l(t) = \{-0.6069, 1.6333, -1.0265\}\text{rad}, \quad \mathbf{q}_r(t) = \{-2.5074, -2.6811, 8.3301\}\text{rad},$$
$$\dot{\mathbf{q}}_l(t) = \dot{\mathbf{q}}_r(t) = \mathbf{0}_{3\times 1}\text{rad/s}.$$

First approach of SDC parameterization (independent control) is used for simulation. Weighting matrices of the SDRE were selected as

$$\mathbf{Q}_l = \mathbf{Q}_r = 10,000 \times \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix}, \ \mathbf{R}_l = \mathbf{R}_r = 0.001 \times \mathbf{I}_{3\times 3}.$$

To show the superiority of the proposed method, the dynamic load carrying capacity index was assessed. The DLCCs of two separated arms were computed in the same path which resulted in 1.09 kg load capacity for the left and 2.29 kg for the right arm, total load is 3.38 kg. It should be remarked that touching the upper or lower bound of actuators defines load carrying capacity. Possessing same control parameters and similar total load of $m_p = 3.38$ kg, but in cooperative form with equal distribution matrices $\bar{\mathbf{Q}}_l = \bar{\mathbf{Q}}_r = \mathbf{I}_{3\times 3}$, show reduction of end-effector errors: 24% for left and 29% for right arm which are presented in Fig. 4. These reductions were happened while the consumption of energy was reduced as well, illustrated in Fig. 5.

Load distribution opportunity changed the share of load in a way that both arms' actuators touched their upper or lower bounds, while control parameters remained the same. This led to selection of $\bar{\mathbf{Q}}_l = 0.49 \times \mathbf{I}_{3\times 3}$ and $\bar{\mathbf{Q}}_r = \mathbf{I}_{3\times 3}$ which resulted in total load of $m_p = 10.2$ kg, almost three times of independent arms. Trajectories and links' configurations are illustrated in Fig. 6. First motor of left and third motor of right arm touched their allowable bounds, presented in Figs. 7 and 8.

Discussions on clustering term in control law: The optimal SDC design of Section 3.3 is considered for analyzing the effect of clustering terms of SDC matrices in control law. The joint space variation of this circular motion in Fig. 6 faces singularity for third link of left arm at 3.34 s, see Fig. 9. As a result, instead of diagonal form of optimal SDC (44) and (45), one should use the second option of $\Omega(\mathbf{x}(t))$ (46) as

$$\Omega_l(\mathbf{x}(t)) = \begin{bmatrix} \frac{u_{l,1}^{\text{add}}(t)}{q_{l,1}(t)} & 0 & 0 \\ \frac{u_{l,2}^{\text{add}}(t)}{q_{l,1}(t)} & 0 & 0 \\ \frac{u_{l,3}^{\text{add}}(t)}{q_{l,1}(t)} & 0 & 0 \end{bmatrix}, \ \Omega_r(\mathbf{x}(t)) = \begin{bmatrix} \frac{u_{r,1}^{\text{add}}(t)}{q_{r,1}(t)} & 0 & 0 \\ \frac{u_{r,2}^{\text{add}}(t)}{q_{r,1}(t)} & 0 & 0 \\ \frac{u_{r,3}^{\text{add}}(t)}{q_{r,1}(t)} & 0 & 0 \end{bmatrix}.$$

www.manaraa.com

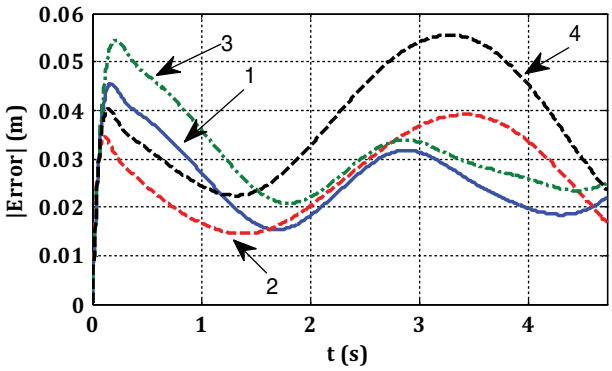


Fig. 4. End-effector error of planar cooperative arms in comparison with separated arms; (1) left cooperative arm; (2) right cooperative arm; (3) left independent arm; (4) right independent arm.

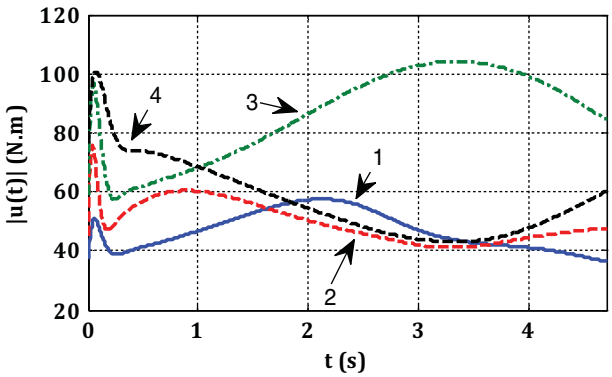


Fig. 5. Comparison of input norms between cooperative and independent arms (legends same as Fig. 4).

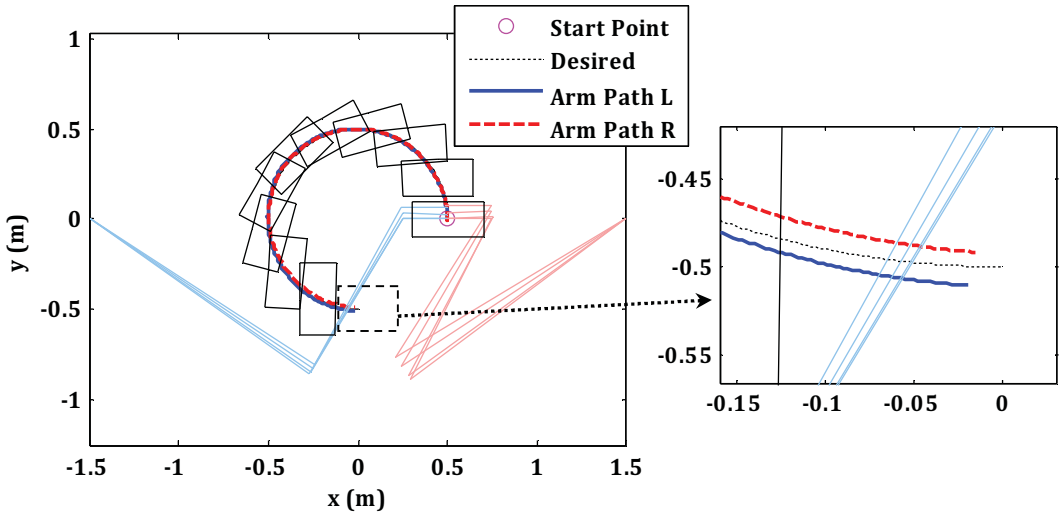


Fig. 6. Trajectories and links' configuration of left and right arm in cooperation with $m_p = 10.2\,\text{kg}$.

Simulations of the both cases show that the remarks in Section 3.3 is true as one can see that the norm of control inputs started from zero in case of optimal SDC design in Fig. 10, and the difference in the norms are not so much.

Linear path: This trajectory is designed to show the similar output of two proposed SDC parameterizations of Section 3.1. Trajectory is a symmetric line with respect to Y-axis which starts from $(0, -0.5)\,\text{m}$ and ends to $(0, 0.5)\,\text{m}$ in $5\,\text{s}$. The path can be expressed by $X_e(t) = 0$ and

www.manaraa.com

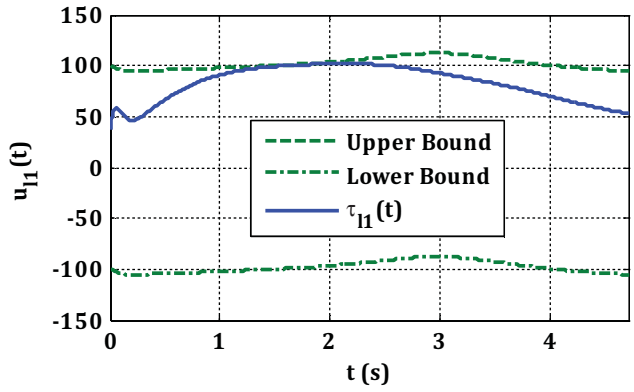


Fig. 7. First input of left arm.

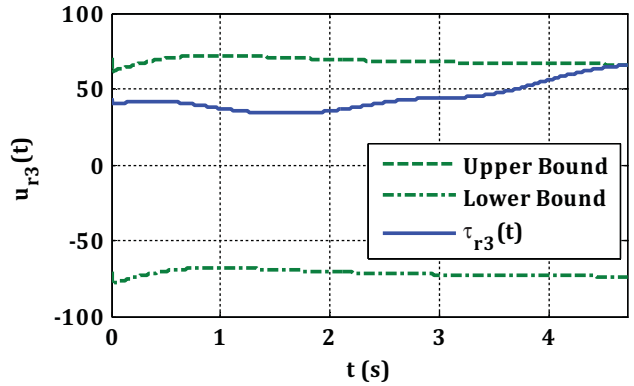


Fig. 8. Third input of right arm.

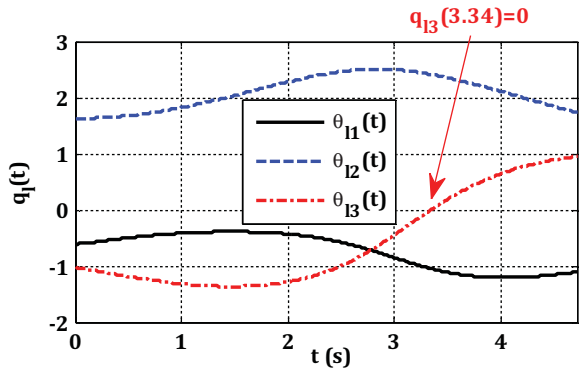


Fig. 9. Joint variation of left arm in circular motion.

$Y_e(t) = 6/625t^4 - 12/125t^3 + 6/25t^2$. Ninety degrees counter-clockwise rotary motion is required which its angular velocity of that is in the form of $\omega_e(t) = \frac{3}{625\pi}t^4 - \frac{6}{125\pi}t^3 + \frac{3}{25\pi}t^2$, and then the initial conditions can be set

$$\mathbf{q}_l(0) = \{-1.0907, 2.0705, -0.9798\}\text{rad}, \quad \mathbf{q}_r(0) = \{-2.0509, -2.0705, 7.2630\}\text{rad},$$
$$\dot{\mathbf{q}}_l(0) = \dot{\mathbf{q}}_r(0) = \mathbf{0}_{3\times 1}\text{rad/s}.$$

The rest of the controller and distribution matrices are similar to circular motion case, except mass of the object. Tracking of separated arms resulted in 3.515 kg load for the left and 3.21 kg load for the right arm; totally 6.725 kg. Regarding 6.725 kg load for cooperative system and equal sharing of load,

www.manaraa.com

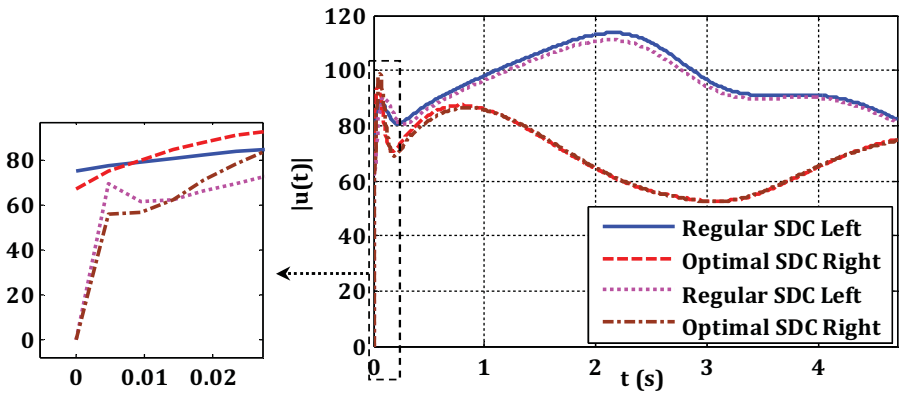


Fig. 10. Comparisons of optimal and general SDC design for norm of control inputs.

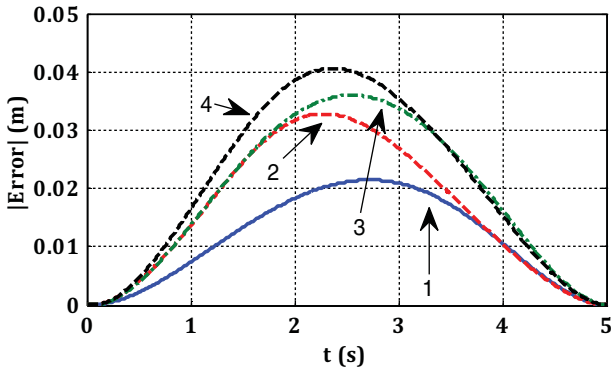


Fig. 11. Comparison of end-effector error of cooperative system with independent arms; (1) left cooperative arm, (2) right cooperative arm, (3) left independent arm and (4) right independent arm.

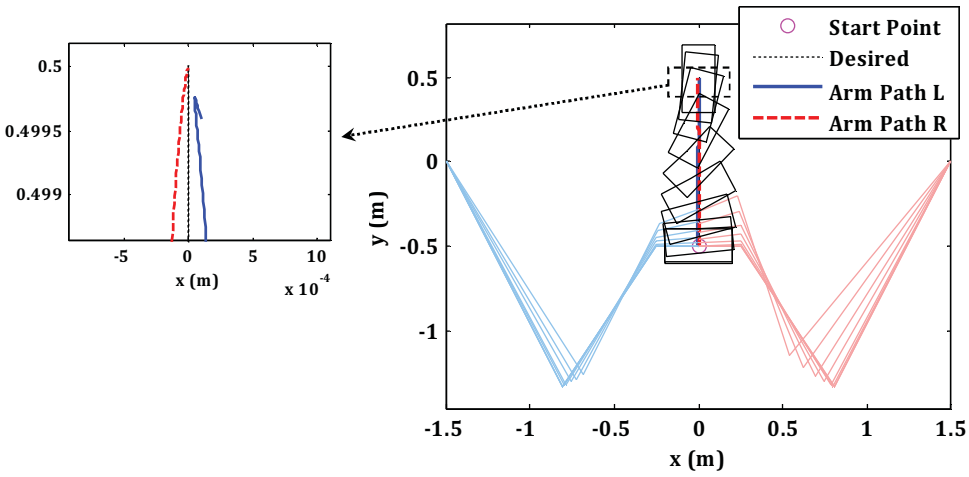


Fig. 12. Trajectory and configuration of planar arms in linear motion.

error of left arm reduced 40% and right one 19%, presented in Fig. 11. Symmetric path suggested equal results for both arms, though the desired orientation changed this balance.

The distribution option of the OLD divides the loads till both arms reach their bound of actuators. The weighting matrices of load distribution were selected as $\bar{\mathbf{Q}}_l = 0.69 \times \mathbf{I}_{3\times3}$ and $\bar{\mathbf{Q}}_r = \mathbf{I}_{3\times3}$ which increased the DLCC to $m_p = 10.67$ kg almost 1.5 times more than separated system. The trajectory and configuration of the arms are shown in Fig. 12.

www.manaraa.com

Table III. The D–H parameters of the seven-DoF arm.

Joint	d (m)	a (m)	θ (rad)	α (rad)	Home
1	d_1	0	$q_1(t)$	$-\pi/2$	0
2	d_2	0	$q_2(t)$	$\pi/2$	0
3	$d_3(t)$	0	0	0	0.2 m
4	d_4	0	$q_4(t)$	$-\pi/2$	0
5	0	a_5	$q_5(t)$	$\pi/2$	$-\pi/2$ rad
6	0	0	$q_6(t)$	-$\pi/2$	0
7	0	a_7	$q_7(t)$	$\pi/2$	0

Table IV. The seven-DoF arm's specifications.

Link	m (kg)	a (m)	a_c (m)	d (m)	d_c (m)	b_l^v (kg.m/s)	b_l^d (kg.m^2/s^2)	b_l^s (kg.m^2/s^2)
1	2	0	0	0.5	0.25	0.05	0.005	0.0025
2	1.5	0	0	0.5	0.25	0.05	0.005	0.0025
3	1	0	0	–	–	0.04 (kg/s)	0.004 (kg.m/s^2)	0.002 (kg.m/s^2)
4	0.75	0	0	0.3	0.15	0.04	0.004	0.002
5	0.75	0.25	0.125	0	0	0.03	0.003	0.0015
6	0.5	0	0	0	0	0.02	0.002	0.001
7	0.25	0.2	0.1	0	0	0.01	0.001	0.0005

The first control approach for implementation was based on independent control of each arm. In an event of failure in motion tracking of one arm, only performance of that subsystem drops and other subsystems continue their work. The distribution was done by the OLD. The second approach designed a uniform controller for entire system in which if an arm fails to finish its task, the entire system will fail. In ordinary situation, both approaches provide the same output, but with different computation time. In singularity points, independent system only quits the singular-situated manipulator, though the uniform system in the same event shuts down the entire work.

4.2. Complex robotic system: four complex manipulators in cooperation

The load distribution expressions for dividing the load out between arms with the aim of Jacobian matrix and generalized inverse was presented; since Eq. (62) in Section 4.1 was not found invertible, the reduction of jacobian $\mathbf{J}_i(\mathbf{q}_i(t))$ to square form when DoF of robot is less than six was proposed as a remedy. For manipulators with more than six DoF, non-square Jacobian exists again, but this section shows that Eq. (62) *is invertible* for this type of systems ($n > 6$); representative of complex manipulators in cooperation. Consider four manipulators, each one with seven DoF (non-square Jacobian) placed in corners of a square with edge of $2d = 0.8$ m. First arm is set in third quarter of X–Y plane of Cartesian coordinates, the second arm is placed in second quarter, third arm in first, and fourth manipulator in fourth quarter. Robots are identical and the D–H parameters of them are expressed in Table III. The mass, inertia, and friction data of robots are also presented in Table IV; and moreover, $\varepsilon = 0.1$, gravity is in Z-axis direction $g_0 = 9.81\,\mathrm{m/s^2}$, and external load is regarded $m_p = 5\,\mathrm{kg}$.

A helical trajectory with center of $(0, 0)$ in XY plane and height of 0.5 m was designed to be tracked in $t_f = 2\pi$ s, which following relations present the motion:

$$X_e(t) = 0.2\cos t,\ \ Y_e(t) = 0.2\sin t,\ \ Z_e(t) = 0.01t.$$

The initial condition was selected as

$$\mathbf{q}_1(0) = \{0.4489\text{rad}, 0.889\text{rad}, 0.2535\text{m}, 0\text{rad}, 0\text{rad}, 0\text{rad}, 0\text{rad}\},$$
$$\mathbf{q}_2(0) = \{5.5561, 0.889, 0.2535, 0, 0, 0, 0\},\ \mathbf{q}_3(0) = \{4.0232, 0.575, 0.1, 0, 0, 0, 0.953\},$$
$$\mathbf{q}_4(0) = \{1.8089, 0.575, 0.1, 0, 0, 0, 0.953\},\ \dot{\mathbf{q}}_1(0) = \dot{\mathbf{q}}_2(0) = \dot{\mathbf{q}}_3(0) = \dot{\mathbf{q}}_4(0) = \mathbf{0}_{7\times 1}.$$

www.manaraa.com

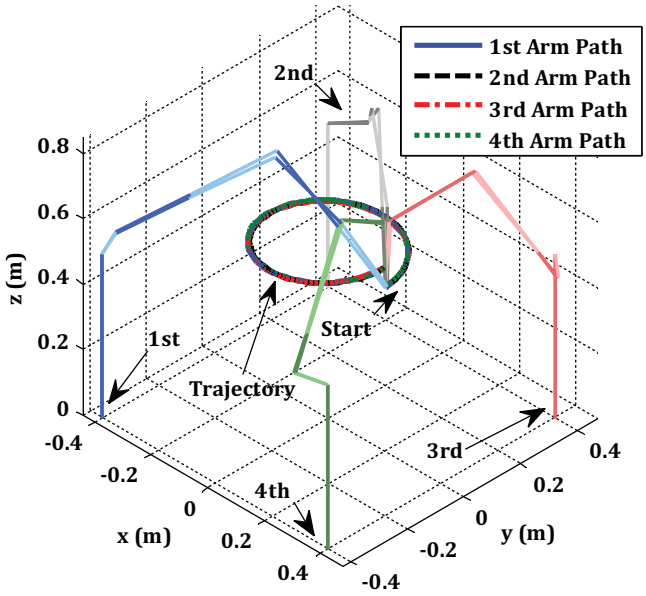


Fig. 13. Trajectory and configuration of links of complex cooperative system.

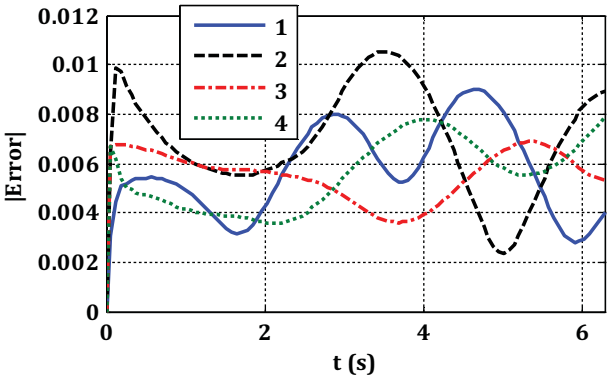


Fig. 14. End-effector errors of complex cooperative system.

The weighing matrices for controller design and distribution rate were chosen as $\mathbf{Q}_i = 10000 \times \text{diag}(\mathbf{1}_{1\times7}, \mathbf{0}_{1\times7})$, $\mathbf{R}_i = 0.001 \times \mathbf{I}_{7\times7}$, and $\bar{\mathbf{Q}}_i = \mathbf{I}_{7\times7}$ for $i = 1, ..., 4$. The dynamic matrices and vectors along with Jacobian are to be found (in symbolic form) by applying the D–H parameters of Table III in general dynamic program, "Section 4.1" of ref. [22]. The trajectories of arms and configuration of links and errors of end-effectors are illustrated in Figs. 13 and 14.

Regarding the size of the manipulators, the accuracy of system is evaluated good and can be even better with changing weighting matrices. This case study shows capability of the proposed formulation for implementing on complex systems as well as a representative for systems with prismatic joints. Moreover, inevitability of Eq. (62) for non-square Jacobian matrix was shown.

4.3. Cooperative Scout robot

4.3.1. Two arms each one three DoF. This section considers Scout robot as a cooperative system with two arms each one three DoF to prepare a study for simulation and experimental validation. Robot and schematic of this system are presented in Fig. 15. As it was mentioned, each arm of Scout has five DoF which reduced to three DoF for the sake of experimental implementation. Hence, two last motors were fixed during the simulation and experimental tests.

Type of arms is PUMA and the D–H parameters and physical specifications of them are presented in Tables V and VI. Regarding the direction of Y-axis in Fig. 15, the right arm is in negative part of that axis.

www.manaraa.com

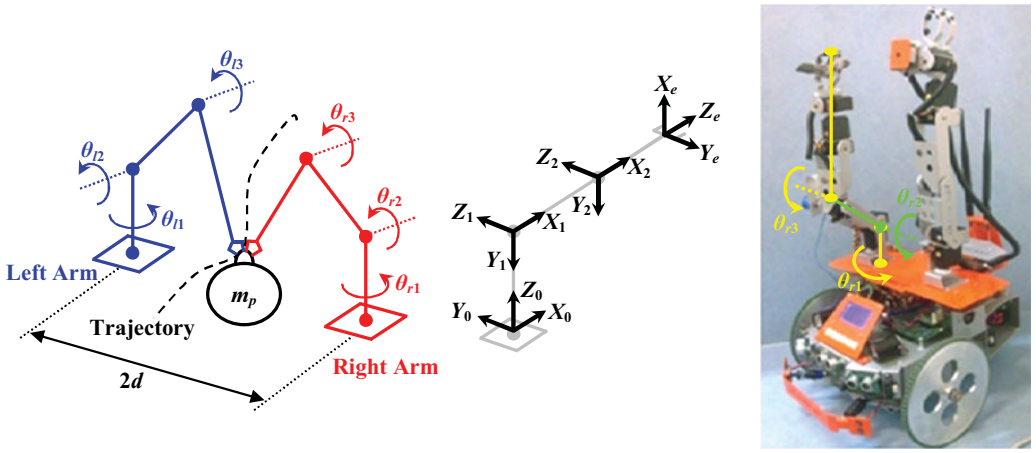


Fig. 15. Scout robot and its schematic with three-DoF modeling.

Table V. The D–H parameters of Scout in three-DoF model.

Joint	d (m)	a (m)	θ (rad)	α (rad)	Home
1	$d_1 = 0.06$	0	$q_1(t)$	$-\pi/2$	0
2	0	$a_2 = 0.1$	$q_2(t)$	0	0
3	0	$a_3 = 0.21$	$q_3(t)$	0	0

Table VI. Specifications of Scout in three-DoF model.

Description	Parameters	Value	Units
Mass of links	m_1, m_2, m_3	0.028, 0.1, 0.231	kg
Moment of inertia of first link	$I_{xx1}, I_{yy1}, I_{zz1}$	Constant, 0.0005, Constant	kg.m^2
Moment of inertia of second link	$I_{xx2}, I_{yy2}, I_{zz2}$	0.000016, 0.000034, 0.000021	kg.m^2
Moment of inertia of third link	$I_{xx3}, I_{yy3}, I_{zz3}$	0.000923, 0.000915, 0.000085	kg.m^2
Distant between bases	d	0.08	m
Viscous friction	$b^v_{l,1}, b^v_{l,2}, b^v_{l,3}$	0.02, 0.02, 0.02	kg.m/s
Dynamic friction	$b^d_{l,1}, b^d_{l,2}, b^d_{l,3}$	0.01, 0.01, 0.01	kg.m^2/s^2
Static friction	$b^s_{l,1}, b^s_{l,2}, b^s_{l,3}$	0.005, 0.005, 0.005	kg.m^2/s^2
Small positive constant	ϵ	0.02	–
Gravity in Z-axis direction	g_0	9.81	m/s^2
Stall torque of motors	$u_{stall,l,1}, u_{stall,l,2}, u_{stall,l,3}$	0.55, 0.96, 0.96	N.m
No-load speed of motors	$\omega_{nl,l,1}, \omega_{nl,l,2}, \omega_{nl,l,3}$	5.85, 3, 3	rad/s

The end-effector positions are expressed as follows:

$$
\begin{aligned}
X_{e,i}(t) &= \cos(\theta_{i,1})(\cos(\theta_{i,2})a_3\cos(\theta_{i,3}) - \sin(\theta_{i,2})a_3\sin(\theta_{i,3}) + a_2\cos(\theta_{i,2})),\\
Y_{e,i}(t) &= \sin(\theta_{i,1})(\cos(\theta_{i,2})a_3\cos(\theta_{i,3}) - \sin(\theta_{i,2})a_3\sin(\theta_{i,3}) + a_2\cos(\theta_{i,2})),\\
Z_{e,i}(t) &= -\sin(\theta_{i,2})a_3\cos(\theta_{i,3}) - \cos(\theta_{i,2})a_3\sin(\theta_{i,3}) - a_2\sin(\theta_{i,2}) + d_1,
\end{aligned}
$$

and the inverse kinematics of them are as follows:

$$
\begin{aligned}
\theta_{i,1}(t) &= \arctan(\bar{Y}_{e,i}(t)/X_{e,i}(t)),\\
\theta_{i,3}(t) &= \arccos\frac{X^2_{e,i}(t)+\bar{Y}^2_{e,i}(t)+(Z_{e,i}(t)-d_1)^2-a_2^2-a_3^2}{2a_2a_3},\\
\theta_{i,2}(t) &= -\arctan\frac{Z_{e,i}(t)-d_1}{\sqrt{X^2_{e,i}(t)+\bar{Y}^2_{e,i}(t)}} - \arctan\frac{a_3\sin\theta_{i,3}(t)}{a_2+a_3\cos\theta_{i,3}(t)},
\end{aligned}
$$

for $i = l, r$ regarding that $\bar{Y}_{e,l}(t) = -d + Y_{e,l}(t)$ and $\bar{Y}_{e,r}(t) = d + Y_{e,r}(t)$ are held. The rest of the dynamics equation, Jacobian, and etc. could be derived by general code in "Section 4.1" of ref. [22].

www.manaraa.com

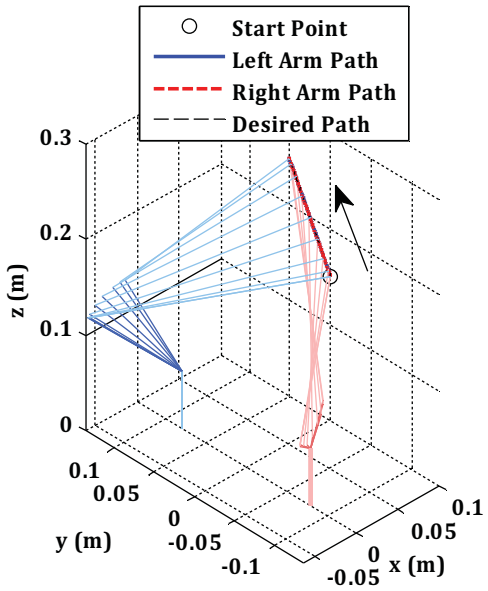


Fig. 16. Scout trajectory with three-DoF modeling.

The state vector of the system is

$$\mathbf{x}(t) = [\theta_{l,1}(t), \theta_{l,2}(t), \theta_{l,3}(t), \theta_{r,1}(t), \theta_{r,2}(t), \theta_{r,3}(t), \dot{\theta}_{l,1}(t), \dot{\theta}_{l,2}(t), \dot{\theta}_{l,3}(t), \dot{\theta}_{r,1}(t), \dot{\theta}_{r,2}(t), \dot{\theta}_{r,3}(t)]^T,$$

which results in state-space representation of system as Eq. (57), but with different values. The limitations of motors are similar to Eqs. (58) and (59) with regard to Table VI.

The path is a line between (0.1, 0, 0.15) m and (0.05, 0, 0.3) m in Cartesian coordinates which is to be tracked in 3 s, symmetric to Y-axis and can be defined by

$$\dot{X}_{e,i}(t) = -1/162t^4 + 1/27t^3 - 1/18t^2, \ \dot{Y}_{e,i}(t) = 0, \ \dot{Z}_{e,i}(t) = 1/54t^4 - 1/9t^3 + 1/6t^2,$$

which led to the following initial conditions:

$$\mathbf{q}_l(0) = \{-0.6747, -2.4951, 2.3529\}\text{rad}, \ \mathbf{q}_r(0) = \{0.6747, -2.4951, 2.3529\}\text{rad}, \ \dot{\mathbf{q}}_l(0) = \dot{\mathbf{q}}_r(0) = \mathbf{0}\,\text{rad/s}.$$

Control and distribution weighting matrices were set as $\mathbf{Q}_l = \mathbf{Q}_r = 100 \times \text{diag}(\mathbf{1}_{1\times3}, \mathbf{0}_{1\times3})$, $\mathbf{R}_l = \mathbf{R}_r = 0.01 \times \mathbf{I}_{3\times3}$, and $\bar{\mathbf{Q}}_l = \bar{\mathbf{Q}}_r = \mathbf{I}_{3\times3}$. First, separated arms were simulated to find the capability of conventional systems, resulted in load capacity of $m_{e,l} = m_{e,r} = 0.279$ kg. The errors of two arms in that case were less than 2 mm and maximum norm of inputs were gained $|u_l|_{\max} = |u_r|_{\max} \approx 0.9$ Nm which touching of third actuator defined the aforementioned load. With the aim of the OLD method, total load for cooperative system resulted in $m_p = 2 \times 0.279 = 0.558$ kg, with maximum error of 0.75 mm and maximum norms of inputs as $|u_l|_{\max} = |u_r|_{\max} \approx 0.84$ Nm. The effect of the OLD method is showing off more on error reduction (62.5%) rather than energy consumption. The load capacity increased to $m_p = 0.737$ kg, keeping the maximum norms of inputs as $|u_l|_{\max} = |u_r|_{\max} \approx 1.04$ Nm and error of 0.75 mm. Once again, the third motor touched its bound to define the DLCC, but increasing the norm of input was due to increase in torque of first motor for providing more stability in motion. The end-effector trajectory and configuration of links are shown in Fig. 16. Norm of inputs for three cases and error of gripper are illustrated in Figs. 17 and 18. Third input of left arm is presented in Fig. 19 which is similar to the right arm as well.

4.3.2. Flexible joint model of two cooperative arms of Scout. The flexible joint model increases the size of state vector twice. The schematic model of flexible case is shown in Fig. 20.

www.manaraa.com

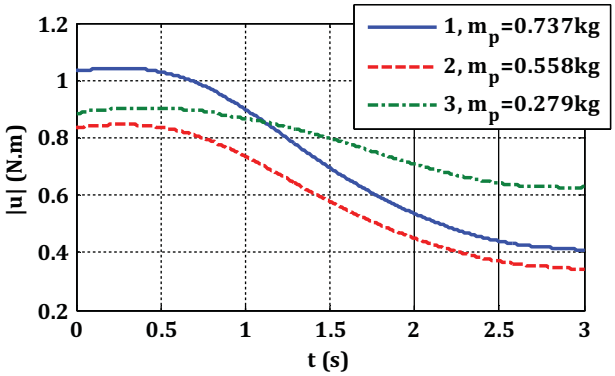


Fig. 17. Norm of inputs Scout robot, three-DoF model; (1) cooperative system with 0.737 kg load, (2) the same with 0.588 kg load, (3) separated arms with 0.279 kg load.

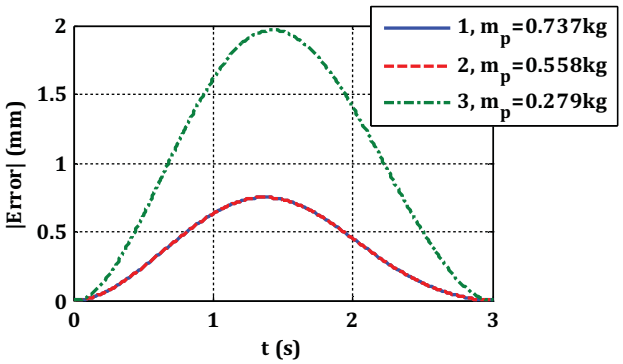


Fig. 18. End-effectors error of Scout for three cases.

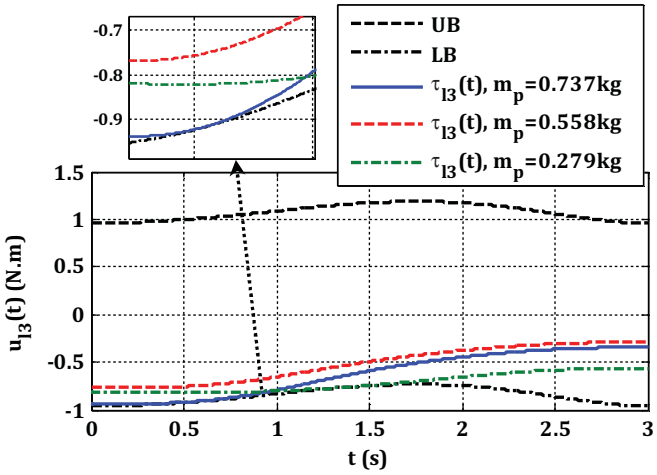


Fig. 19. The third input of left arm.

The state vector is considered as

$$\mathbf{x}(t) = [(\mathbf{q}_l^l(t))^T, (\mathbf{q}_r^l(t))^T, (\mathbf{q}_l^m(t))^T, (\mathbf{q}_r^m(t))^T, (\dot{\mathbf{q}}_l^l(t))^T, (\dot{\mathbf{q}}_r^l(t))^T, (\dot{\mathbf{q}}_l^m(t))^T, (\dot{\mathbf{q}}_r^m(t))^T]_{24\times 1}^T,$$

where $\mathbf{q}_l^l(t) = \{\mathbf{q}_{l,1}(t), \mathbf{q}_{l,2}(t), \mathbf{q}_{l,3}(t)\}$ and $\mathbf{q}_l^m(t) = \{\mathbf{q}_{l,4}(t), \mathbf{q}_{l,5}(t), \mathbf{q}_{l,6}(t)\}$. The state-space equation of system is similar to Eq. (33) with $n = 3$ (number of DoF in rigid case) and $m = 2$ number of arms with $i = r, l$ indices for right and left arm. The SDC parameterization matrices are similar to Eqs. (34)–(37). The new parameters with respect to rigid joint case, Section 4.3.1, are joint flexibility

www.manaraa.com

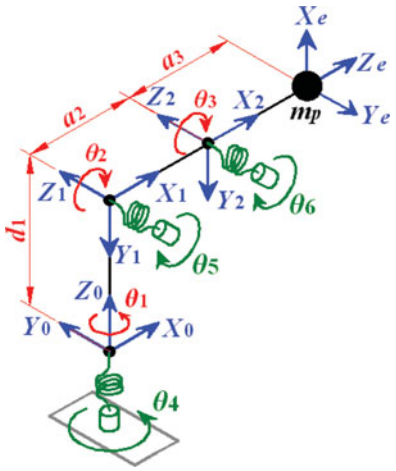


Fig. 20. The schematic view of scout arm with three-DoF model.[21]

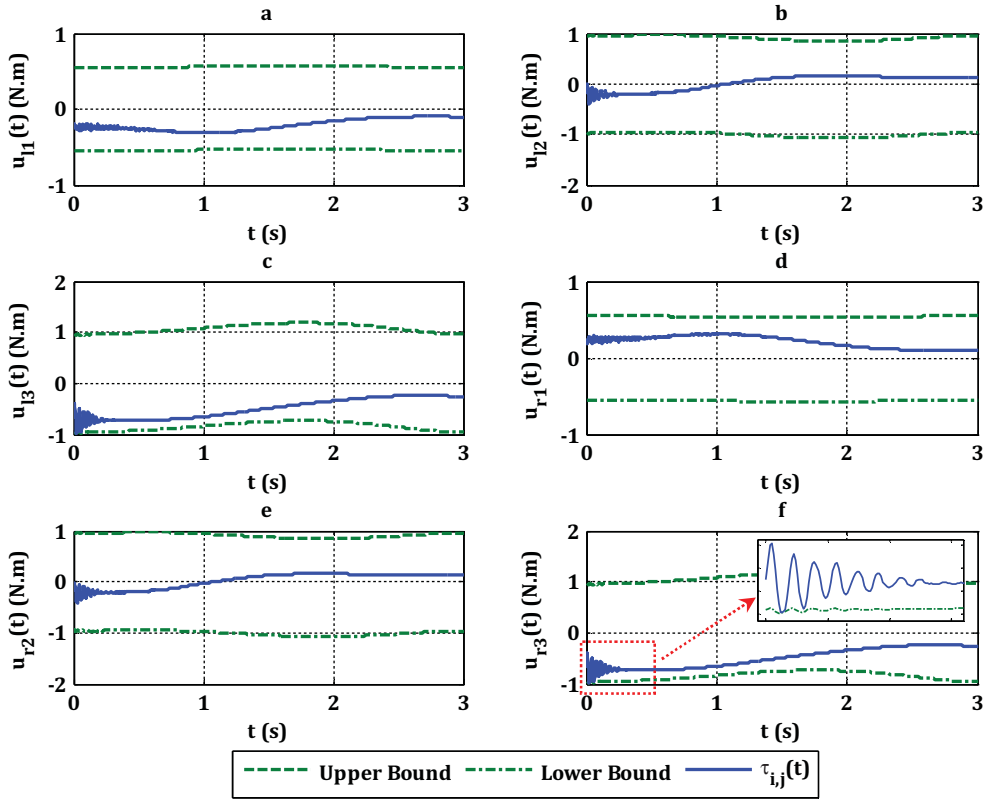


Fig. 21. Input torque of flexible joint model of Scout.

parameters: spring constants and rotors' inertia defined as $J_{i,j} = 0.05\,\text{kgm}^2$ and $\tilde{k}_{i,j} = 100\,\text{Nm/rad}$ for $i = l, r$ and $j = 1, 2, 3$.[24] The rest of conditions are identical to Section 4.3.1. The load capacity due to oscillation of joint reduced to $m_p = 0.5\,\text{kg}$ and the error increased to 4 mm. The trajectories of arms are similar to Fig. 16 with 2 mm more error which does not express the oscillations. Inputs of the system are shown in Fig. 21. Oscillation of third link first touched the lower bound and as a result defined the DLCC. The results show that flexible joint model provides more real outputs near the experimental tests in Section 5.

www.manaraa.com

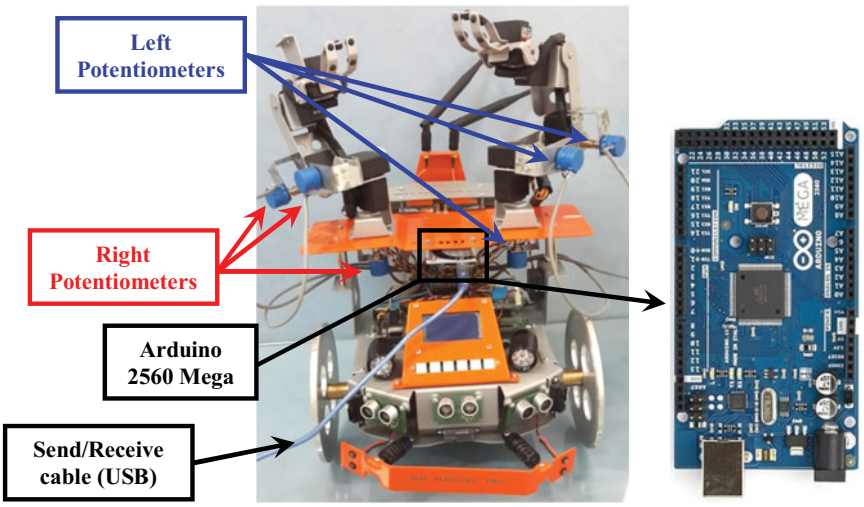


Fig. 22. Details of experimental setup of Scout robot.

5. Experimental Study: Scout Robot

Scout robot was subjected to experimental tests for cooperative motion. The objective was to show that the proposed method is applicable for real systems and to validate the obtained simulation data. Scout has two arms which each one has five DoF. The actuators of arms are Hitec servomotors; HS645MG for first links of left and right and HS322HD for other links. Three external potentiometers for each arm were set to record the variation of links during the motion, presented in Fig. 22. It was not easy to install potentiometers for fourth and fifth links; hence, main results and discussions in simulation parts were gathered for reduced 3R model in Sections 4.3.1 and 4.3.2.

The Arduino Mega 2560, the processor of Scout, is a microcontroller digital board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analogue inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The Arduino board has so many prepared libraries to help users for programming the system and showed its effectiveness in different digital applications.[26,27] These options motivated us to select it as the main processor for computations and communications.

A Hitec servo motor has internal feedback which drives a proportional-integral-derivative loop that cannot be modified. This helps to work with the robot easily though limits the application of the system drastically such as accepting only position as an input command to a servo motor. The necessary library for common drive of servomotors is "Servo" which allows only the position command as

```
myservo.write(90); // set servo to mid-point
```

where 90° input may change between 0–180. This is a drawback since one needs to change the speed of motion. Recently, a new library, "VarSpeedServo", was released which provided variable speed control of servomotors. This command permits to define the speed of motion as

```
myservo.write(90, 30, true); // move to 90 degrees, use a speed of 30.
```

Using the proposed elements, hardware and software, one could track the proposed trajectory in Section 4.3.1. Here, "30" is a constant speed for servo motor motion. The speed could vary between 1–255, which 1 sets the slowest speed and 255 sets the fastest possible one. In order to implement the SDRE, its output in "Nm" scale must be mapped to PWM limits of 0–255. The command "map" for Arduino performs it easily. We inserted the output of the SDRE control law after changing its scale to PWM as a variable speed of motion. Two methods are available for calculation of the SDRE which will be expressed with more details.

There are three steps for implementation of the SDRE on Scout robot. First, the control law of the SDRE must be calculated. Most computational methods for solving Riccati equation are numerical, capable of experimental implementation.[21] Numerical solution to the SDRE leads to a time-varying suboptimal gain for a specific trajectory; another path needs another numerical solution.[21] Power

www.manaraa.com

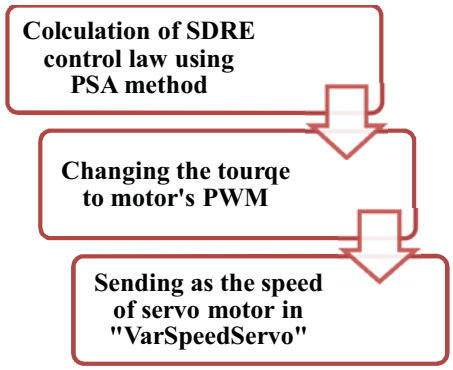


Fig. 23. Implementation steps of the SDRE controller.

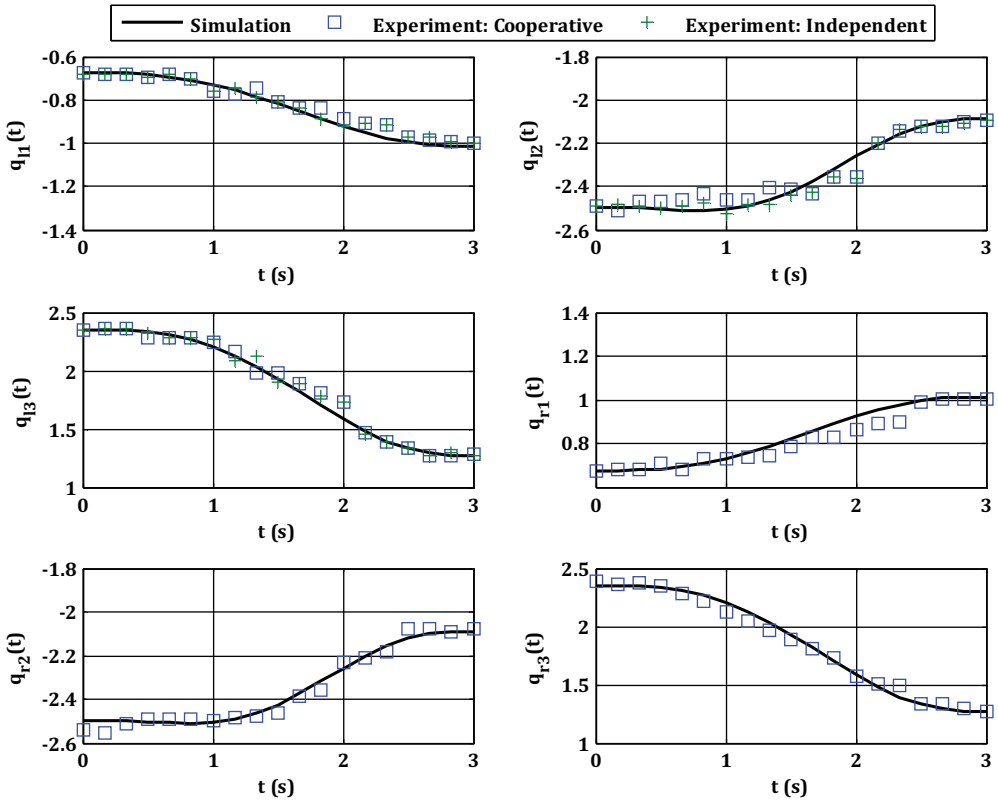


Fig. 24. Joints variation of Scout: simulation and experiment.

series approximation (PSA) is a solution method to the SDRE as well.[20] Using the PSA, one may find the control law as a nonlinear function that can be used for any given trajectory. The PSA method is used to generate control law of Scout robot as a non-linear function to complete the first step. The details of the PSA ought to be found in ref. [20]. Second step is mapping the torque scale of SDRE's output to PWM input command of servo motor. The last step is sending the data to motors. The implementation steps of this section are shown in Fig. 23.

The trajectory was divided to 20 points to be tracked in 3 s. The time step of experiment was set 50 ms as well. Similar to simulation results, the comparison of single free arm motion and cooperative motion was carried out. Left arm could carry 0.09 kg independently with peak of 14 mm error. The joints variations and error of end-effector were presented in Figs. 24 and 25. More loads results in failure in motion and large downward deflection of arm.

www.manaraa.com

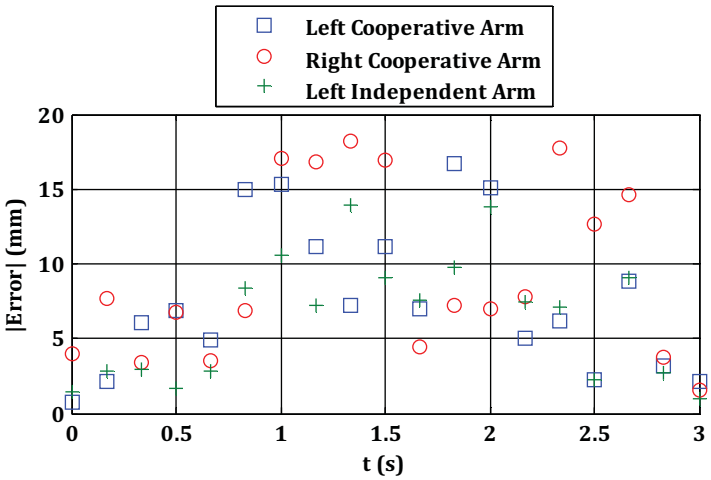


Fig. 25. End-effector error of left arm in an independent motion.

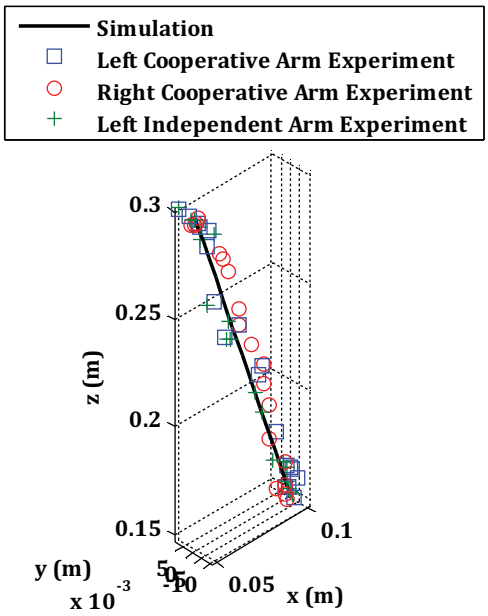


Fig. 26. End-effector trajectories of left and right arms.

For cooperative motion, both arms pulled the load upward after contact to track the predefined path. The manipulators could carry 0.3 kg load in cooperative motion. The peak of error in 3 s was less that 20 mm. Trajectories are illustrated in Fig. 26.

There are not so many options to implement non-linear controllers on servomotors practically. A new point of view for implementation of non-linear controllers, specially the SDRE, was proposed to close the practical tests to theoretical data and simulations. This approach might be used for other non-linear controllers as well. The experimental results were obtained a little different from simulations, but the fact that cooperation increases the load capacity more than two times of independent robots was confirmed.

6. Conclusions

This research presented cooperative control of manipulators using a SDRE in a general approach. The contact of the arms with the object imported constraints and complexity into the problem. The optimal load distribution method was used to split the load between arms and provided the opportunity to

www.manaraa.com

share the load at the desired rate in the system. The dynamics and control of arms were investigated using independent and uniform control for m arms, each possessing n-DoF.

Manipulators with rigid and flexible joints were considered in the dynamics. State-dependent coefficient parameterization is an important step in design using the SDRE method. The proper choice of SDC leads to the solution of the SDRE. Parameterization of independent systems for each arm and a uniform SDC form for the entire cooperative system were introduced. The major differences between these types are the time required for a solution and the number of Riccati equations. The first case solved m SDRE, but the second case only solved a large SDRE. The intention of cooperation was and is to increase load capacity as an important index of the manipulators; however, stability in motion and reduction of error were promising result from the simulations.

A three DoF planar arm was simulated to carry a lumped or distributed mass by orientation of the object. The load capacity in circular motion resulted in a 3.38 kg load for both free arms; cooperation increased the load almost three-fold to $m_p = 10.2$ kg. Simulation of Scout robot in 3D space, where the touch of an actuator bound defined the load capacity, resulted in $m_{e,l} = m_{m,r} = 0.279$ kg for each arm because the trajectory was designed symmetrically. The error for each arm was less than 2 mm. This value for cooperation was 0.75 mm and an increase of $m_p = 0.737$ kg in loading.

The flexibility of the joints provided real results that were nearer to practical test results. This assumption in modeling reduced the load capacity to $m_p = 0.5$ kg with less than 4 mm of error. The results confirmed that cooperation improved load capacity and reduced the error accompanied by stability in motion. The SDRE controller successfully tracked a predefined path, could handle complex systems, and advanced to control of four arms, each one seven DoF (84 states). Diverse simulations and easy implementation were the consequences of the proposed precise formulation.

Supplementary Material

To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574717000522

References

1. S. Hayati, K. Tso and T. Lee, "Dual arm coordination and control," *Robot. Auton. Syst.* **5**(4), 333–344 (1989).
2. T. Kokkinis, "Dynamic hybrid control of cooperating robots by nonlinear inversion," *Robot. Auton. Syst.* **5**(4), 359–368 (1989).
3. C. J. Li, "Coordinated motion control of multi-arm robot systems with optimal load distribution," *Syst. Control Lett.* **15**(3), 237–245 (1990).
4. 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.
5. J. T. Wen and K. K. Delgado, "Motion and force control of multiple robotic manipulators," *Automatica* **28**(4), 729–743 (1992).
6. 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).
7. S. T. Lin and H. C. Tsai, "Impedance control with on-line neural network compensator for dual-arm robots," *J. Intell. Robot. Syst.* **18**(1), 87–104 (1997).
8. G. E. Yale and B. N. Agrawal, "Lyapunov controller for cooperative space manipulators," *J. Guid. Control Dyn.* **21**(3), 477–484 (1998).
9. J. S. Liu and S. L. Chen, "Robust hybrid control of constrained robot manipulators via decomposed equations," *J. Intell. Robot. Syst.* **23**(1), 45–70 (1998).
10. J. Zhao and S. X. Bai, "Load distribution and joint trajectory planning of coordinated manipulation for two redundant robots," *Mech. Mach. Theory* **34**(8), 1155–1170 (1999).
11. Z. Jing and S. X. Bai, "The study of coordinated manipulation of two redundant robots with elastic joints," *Mech. Mach. Theory* **35**(7), 895–909 (2000).
12. A. Ghasemi and M. Keshmiri, "Performance Assessment of a Decentralized Controller for Cooperative Manipulators; Numerical and Experimental Study," *6th International Symposium on Mechatronics and its Applications*, Sharjah, United Arab Emirates (2009) pp. 1–6.
13. A. Tavasoli, M. Eghtesad and H. Jafarian, "Two-time scale control and observer design for trajectory tracking of two cooperating robot manipulators moving a flexible beam," *Robot. Auton. Syst.* **57**(2), 212–221 (2009).
14. H. Homaei and M. Keshmiri, "Optimal trajectory planning for minimum vibration of flexible redundant cooperative manipulators," *Adv. Robot.* **23**(12–13), 1799–1816 (2009).

www.manaraa.com

15. N. Yagiz, Y. Hacioglu and Y. Z. Arslan, "Load transportation by dual arm robot using sliding mode control," *J. Mech. Sci. Technol.* **24**(5), 1177–1184 (2010).
16. R. Rastegari and S. A. A. Moosavian, "Multiple impedance control of space free-flying robots via virtual linkages," *Acta Astronaut*. **66**(5), 748–759 (2010).
17. S. C. Lee and H. S. Ahn, "Multiple Manipulator Cooperative Control using Disturbance Estimator and Consensus Algorithm," *Proceedings of the IEEE American Control Conference*, San Francisco, CA, USA (2011) pp. 4002–4007.
18. Y. Hacioglu, Y. Z. Arslan and N. Yagiz, "MIMO fuzzy sliding mode controlled dual arm robot in load transportation," *J. Franklin Inst.* **348**(8), 1886–1902 (2011).
19. V. Panwar, N. Kumar, N. Sukavanam and J. H. Borm, "Adaptive neural controller for cooperative multiple robot manipulator system manipulating a single rigid object," *Appl. Soft Comput.* **12**(1), 216–227 (2012).
20. M. H. Korayem, M. Irani and S. R. Nekoo, "Analysis of manipulators using SDRE: A closed loop nonlinear optimal control approach," *Sci. Iranica* **17**(6B), 456–467 (2010).
21. 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 Trans*. **54**, 125–144 (2015).
22. M. H. Korayem and S. R. Nekoo, "State-dependent differential riccati equation to track control of time-varying systems with state and control nonlinearities," *ISA Trans*. **57**, 117–135 (2015).
23. M. H. Korayem, R. A. Esfeden and S. R. Nekoo, "Path planning algorithm in wheeled mobile manipulators based on motion of arms," *J. Mech. Sci. Technol.* **29**(4), 1753–1763 (2015).
24. M. H. Korayem and S. R. Nekoo, "The SDRE control of mobile base cooperative manipulators: Collision free path planning and moving obstacle avoidance," *Robot. Auton. Syst.* **86**, 86–105 (2016).
25. R. J. Schilling, *Fundamentals of Robotics Analysis and Control* (Prentice Hall, New Delhi, 2003).
26. K. Y. Lian, S. J. Hsiao and W. T. Sung, "Intelligent multi-sensor control system based on innovative technology integration via ZigBee and wi-fi networks," *J. Netw. Comput. Appl.* **36**(2), 756–767 (2013).
27. M. A. K. Yusoff, R. E. Samin and B. S. K. Ibrahim, "Wireless mobile robotic arm," *Procedia Eng*. **41**, 1072–1078 (2012).

www.manaraa.com

Reproduced with permission of copyright owner. Further reproduction prohibited without permission.

www.manaraa.com