2016 European Control Conference (ECC)
June 29 - July 1, 2016. Aalborg, Denmark

On Null–Space Control of Kinematically Redundant Robot Manipulators

Kamil Cetin, Enver Tatlicioglu, and Erkan Zergeroglu

Abstract— **In this study, we consider the null-space control problem of redundant robot manipulators. Specifically for robot manipulators with kinematically redundancy where at least one extra degree of freedom is present, we introduce a sub–task controller that will ensure the use of the extra degrees of freedom for possible control purposes while still ensuring the main objective. The stability of the main (end–effector tracking) and sub–task objectives are obtained via Lyapunov based arguments. Extension to adaptive controller formulation for robotic devices with uncertain system dynamics is also presented. Numerical studies for the adaptive controller are presented to illustrate the liability of the proposed method.**

I. Introduction

The research on kinematically redundant robot manipulators where the dimension (*i.e.,* n) of the joint space is greater than the dimension (*i.e.,* m) of its operational or task–space [1], has been active for quite a long time. This is mostly due to the fact that the design and implementation of some of the relatively simple problems for conventional robotic manipulator, as path planning or dynamic control, might become quite hard when kinematic redundancy has to be taken into account. To name a few; authors of [2] proposed a control scheme based on the dynamic model of a manipulator in Cartesian space and extended this result for redundant manipulators by using a manipulator Jacobian pseudo-inverse formulation. A configuration control approach in which the end–effector motion in task-space is augmented was proposed in [3]. A robust adaptive controller for kinematically redundant robot manipulators was presented in [4]. Using extended task–space formulation Peng *et al.* proposed a compliant motion controller in [5]. In [6], authors have investigated actuator redundancy, where the additional actuation is used to impose some freedom in performing desired tasks.

For redundant robots, there are an infinite number of joint configurations for any given end–effector pose [7]. And while performing a desired task, among all the possible configurations some can be preferred [8]. One way of obtaining a preferred configuration at joint level is through the use of secondary functions. Examples to this are presented in [9] for obstacle avoidance, and in [10] for grasping. The secondary tasks, or sub–tasks can be utilized as an add–on to the main task–space control objective [11], [12]. Assuming that the exact knowledge of the robot dynamics is known Hsu *et al.* presented a dynamical feedback controller that guarantees end–effector tracking and enables the self motion of the manipulator flow along the projection of a given arbitrary vector field, in [13]. Adaptive controller formulations for the same problem with dynamical uncertainties were presented in [14], [15], [16], [17], while robust controller formulations that also compensates for the unstructured uncertainties were presented in [9], [10], [18], [19]. Recently there has been some work on the use of secondary task formulation for kinematically redundant robot manipulators that obtained practical end–effector tracking despite parametric uncertainty in the robot dynamics as in [20], [21], [22], and [23].

In this paper, we aim to design a novel sub–task controller that achieves asymptotic end–effector and sub–task tracking, as opposed to the practical tracking results presented in the literature. In [15] and [16], Zergeroglu *et al.* only considered an adaptive controller that achieves exponential end–effector tracking and sub–task tracking without integration of the sub–task objective into the stability analysis. Later in [20] and [21], Tatlicioglu *et al.* addressed this issue by designing an adaptive controller that achieves asymptotic tracking in the task–space with systematic integration of the sub–task objective into the stability analysis. Different from the previously proposed controllers of [20] and [21], the proposed methodology includes a new term that depends on the sub–task function and its partial derivative with respect to the joint positions. This new term allowed us to conduct a combined stability analysis for both task–space tracking and sub–task objective. A novel Lyapunov function including square of the scalar sub–task function is then introduced. Exponential stability of the exact model knowledge and asymptotic stability of the adaptive controller that also compensates for parametric uncertainties in the robot dynamics are then ensured via Lyapunov type arguments.

II. Dynamic and Kinematic Model

The dynamic model of an n dof (degree of freedom) revolute joint robot manipulator is described as [13]

$$M(\theta)\ddot{\theta} + C_(\theta, \dot{\theta})\dot{\theta} + G(\theta) + F_d\dot{\theta} = \tau(t) \qquad (1)$$

where $\theta(t)$, $\dot{\theta}(t)$, $\ddot{\theta}(t) \in \mathbb{R}^n$ are the joint position, velocity and acceleration vectors, respectively, $M(\theta) \in \mathbb{R}^{n\times n}$ represents the inertia matrix, $C(\theta, \dot{\theta}) \in \mathbb{R}^{n\times n}$ represents centripetal–Coriolis effects, $G(\theta) \in \mathbb{R}^n$ represents gravitational effects, $F_d \in \mathbb{R}^{n\times n}$ is the constant positive definite diagonal dynamic frictional effects, and $\tau(t) \in \mathbb{R}^n$ is the

K. Cetin is with the Department of Electrical & Electronics Engineering, Gediz University, Izmir, Turkey (E-mail: kamil.cetin@gediz.edu.tr).
E. Tatlicioglu is with the Department of Electrical & Electronics Engineering, Izmir Institute of Technology, Izmir, 35430 Turkey (Phone: +90 (232) 7506536; Fax: +90 (232) 7506599; E-mail: enver@iyte.edu.tr).
E. Zergeroglu is with the Department of Computer Engineering, Gebze Technical University, 41400, Gebze, Kocaeli, Turkey (Email: ezerger@bilmuh.gyte.edu.tr).
This work is funded by The Scientific and Technological Research Council of Turkey via grant number 113E147.
All the correspondence should be addressed to E. Tatlicioglu.

control input torque vector. The subsequent development is based on the following properties [24].

Property 1: The inertia matrix $M(\theta)$ is symmetric and positive–definite, and satisfies the following inequalities

$$m_1 \|\zeta\|^2 \leq \zeta^T M(\theta)\zeta \leq m_2 \|\zeta\|^2 \ \ \forall \zeta \in \mathbb{R}^n \tag{2}$$

where m_1, $m_2 \in \mathbb{R}$ are positive constants, and $\|\cdot\|$ denotes the standard Euclidean norm.

Property 2: The inertia and centripetal–Coriolis matrices satisfy the following skew symmetric relationship

$$\zeta^T[\dot{M}(\theta) - 2C(\theta,\dot{\theta})]\zeta = 0 \ \forall \zeta \in \mathbb{R}^n \tag{3}$$

where $\dot{M}(\theta)$ denotes the time derivative of the inertia matrix.

Property 3: The left–hand side of (1) can be linearly parametrized as shown below

$$Y(\theta,\dot{\theta},\ddot{\theta})\phi = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + F_d\dot{\theta} \tag{4}$$

where $Y(\theta,\dot{\theta},\ddot{\theta}) \in \mathbb{R}^{n\times p}$ being the regression matrix, and $\phi \in \mathbb{R}^p$ is the constant parameter vector that depends on robot parameters.

The end–effector position and orientation in the task–space $x(t) \in \mathbb{R}^m$ (with $n > m$) can be written as a function of the joint vector as shown below

$$x = f(\theta) \tag{5}$$

where $f : \mathbb{R}^n \to \mathbb{R}^m$ is the forward kinematics function. Differentiating (5) with respect to time yields

$$\dot{x} = J\dot{\theta} \tag{6}$$

where $\dot{x}(t) \in \mathbb{R}^m$ is the task–space velocity vector, and $J(\theta) \triangleq \frac{\partial f(\theta)}{\partial \theta} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix. Since $m \neq n$ then we will need pseudo–inverse of the Jacobian matrix, denoted by $J^+(\theta) \in \mathbb{R}^{n\times m}$, which is defined as follows

$$J^+ \triangleq J^T(JJ^T)^{-1} \tag{7}$$

which satisfies

$$JJ^+ = I_m \tag{8}$$

where I_m denotes the standard $m \times m$ identity matrix. The pseudo–inverse defined by (7) satisfies the Moore-Penrose conditions as described in [25].

Following remarks are standard in the dynamic redundant robot control literature.

Remark 1: During the control development, we need $J^+(\theta)$ to be available $\forall \theta(t)$ which is possible by avoiding all kinematic singularities *a priori*.

Remark 2: The dynamic and kinematic terms depend on $\theta(t)$ via trigonometric functions only, and as result of this they remain bounded for all possible $\theta(t)$.

III. Error System Formulation

The task–space tracking error, denoted by $e(t) \in \mathbb{R}^m$, is defined as follows

$$e \triangleq x_d - x \tag{9}$$

where $x_d(t) \in \mathbb{R}^m$ denotes the desired task–space trajectory. It is assumed that $x_d(t)$, $\dot{x}_d(t)$ and $\ddot{x}_d(t)$ are all bounded functions of time.

After taking the time derivative of (9) and substituting (6), we can obtain

$$\dot{e} = \dot{x}_d - J\dot{\theta}. \tag{10}$$

To further facilitate the subsequent analysis, we define an auxiliary error–like signal, denoted by $r(t) \in \mathbb{R}^n$, as follows

$$r \triangleq J^+(\dot{x}_d + \alpha e) + (I_n - J^+J)h - \dot{\theta} \tag{11}$$

where $\alpha \in \mathbb{R}^{m\times m}$ is a diagonal, positive–definite, constant, control gain matrix, and $h(t) \in \mathbb{R}^n$ is a null–space controller that is detailed in Section IV. Premultiplying (11) by J and substituting (6) and (10), we obtain

$$\dot{e} = -\alpha e + Jr \tag{12}$$

where (8), (10), and Moore-Penrose conditions were utilized. For simplicity, we define another auxiliary signal, denoted by $r_a(t) \in \mathbb{R}^n$, as follows

$$r_a \triangleq J^+(\dot{x}_d + \alpha e) + (I_n - J^+J)h \tag{13}$$

which can be substituted into (11) to obtain

$$r = r_a - \dot{\theta}. \tag{14}$$

Taking the time derivative of (14), pre–multiplying by inertia matrix $M(\theta)$, and then substituting (1), we obtain the following open–loop error dynamics

$$M\dot{r} = Y\phi - Cr - \tau \tag{15}$$

where

$$Y\phi = M\dot{r}_a + Cr_a + G + F_d\dot{\theta} \tag{16}$$

with $Y(\theta,\dot{\theta},r_a,\dot{r}_a) \in \mathbb{R}^{n\times p}$ and ϕ was determined by (4).

IV. Controller Design and Stability Analysis

In this section, control design and the accompanying stability analysis will be presented when exact knowledge of the robot dynamics is available (*i.e.*, ϕ in (4) or (16) is known). From the error system development in Section III and the subsequent stability analysis, we design the control input torque $\tau(t)$ as follows

$$\tau = Y\phi + K_r r + J^T e - y_s J_s^T \tag{17}$$

where $y_s(\theta) \in \mathbb{R}$ is the yet to be designed sub–task function, $J_s(\theta) \in \mathbb{R}^{1\times n}$ is also yet to be designed, and $K_r \in \mathbb{R}^{n\times n}$ is a positive–definite, constant, diagonal, control gain matrix.

Remark 3: The exact model knowledge (EMK) version of the adaptive controller in [20] and [21] expressed by using the notation in this paper is of the following form

$$\tau = Y\phi + K_r r + J^T e. \tag{18}$$

As opposed to the controller (18), our proposed controller (17) includes the new term ($y_s J_s^T$) which will subsequently allow us to carry out a combined stability analysis for both task–space tracking and sub–task objective.

After substituting (17) into (15), the closed–loop error system for $r(t)$ can be written as follows

$$M\dot{r} = -Cr - K_r r - J^T e + y_s J_s^T. \tag{19}$$

The sub–task function y_s will be designed to depend only on θ. Special care must be taken when designing the sub–task function y_s. Specifically, in the subsequent stability analysis convergence of y_s to zero will be proven. As a result when designing the sub–task function this should be considered and functions that could be driven to zero must be considered.

The time derivative of y_s can then be obtained as

$$\dot{y}_s = J_s \dot{\theta} \tag{20}$$

where $J_s(\theta)$ is the auxiliary Jacobian–like function defined as

$$J_s \triangleq \frac{\partial y_s}{\partial \theta}. \tag{21}$$

Substituting $\dot{\theta}$ in (20) by the expression on $\dot{\theta}$ derived from (11), we obtain the following expression

$$\dot{y}_s = J_s J^+ (\dot{x}_d + \alpha e) + J_s (I_n - J^+ J) h - J_s r. \tag{22}$$

Based on the structure of (22) and the subsequent stability analysis, we design the null–space controller as follows

$$h = -\frac{(I_n - J^+ J) J_s^T}{\|J_s (I_n - J^+ J)\|^2} [k_s y_s + J_s J^+ (\dot{x}_d + \alpha e)] \tag{23}$$

where $k_s \in \mathbb{R}$ is a positive constant. Notice that, $Jh = 0_{m \times 1}$. Provided that the following sufficient condition holds

$$\|J_s (I_n - J^+ J)\| > 0 \tag{24}$$

there is no singularity issue in $h(t)$.

Remark 4: The sub–task control input in [20] and [21] expressed by using the notation in this paper is of the following form

$$h = -k_s [J_s (I_n - J^+ J)]^T y_s. \tag{25}$$

In contrast to the null–space controller (25), our proposed null–space controller (23) is quite different from (25).

Mathematically speaking, the term $\|J_s(I_n - J^+ J)\|$ will be equal to zero when J_s^T is in the null–space of $(I_n - J^+ J)$. And when this is the case, then, in (22), the null–space controller is multiplied by zero. As a result, it is reasonable to assume that (24) is satisfied. In addition, we would like to note that the condition in (24) was previously introduced in [20] and then utilized in [21], [22], [23].

Finally, substituting (23) into (22), we obtain the following closed–loop dynamics for $y_s(t)$

$$\dot{y}_s = -k_s y_s - J_s r. \tag{26}$$

Now we are ready to proceed to the combined stability analysis. We begin our analysis by introducing the following Theorem.

Theorem 1: For the robot manipulator dynamics given in (1), the controller in (17) with the sub–task controller in (23) ensures exponential task–space tracking and exponential sub–task control.

Proof: We start our proof by defining a nonnegative scalar function (*i.e.*, a Lyapunov function), denoted by $V_1(e, r, y_s) \in \mathbb{R}$, as

$$V_1 \triangleq \frac{1}{2} e^T e + \frac{1}{2} r^T M r + \frac{1}{2} y_s^2. \tag{27}$$

By using (2), it can easily be shown that the following bounds hold for (27)

$$\lambda_1 \|z\|^2 \leq V_1 \leq \lambda_2 \|z\|^2 \tag{28}$$

where $\lambda_1 \triangleq \frac{1}{2} \min\{1, m_1\}$, $\lambda_2 \triangleq \frac{1}{2} \max\{1, m_2\}$, and $z(t) \in \mathbb{R}^{(m+n+1)\times 1}$ is the combined error vector defined as

$$z(t) \triangleq \begin{bmatrix} e^T & r^T & y_s \end{bmatrix}^T. \tag{29}$$

After taking the time derivative of (27), we obtain

$$\dot{V}_1 = e^T \dot{e} + r^T M \dot{r} + \frac{1}{2} r^T \dot{M} r + y_s \dot{y}_s \tag{30}$$

Substituting (12), (19) and (26) into (30) yields

$$\begin{aligned} \dot{V}_1 \;=\; & e^T(-\alpha e + Jr) + \frac{1}{2} r^T \dot{M} r \\ & + r^T(-Cr - K_r r - J^T e + y_s J_s^T) \\ & + y_s(-k_s y_s - J_s r) \end{aligned} \tag{31}$$

and after canceling common terms, we obtain the following expression

$$\dot{V}_1 = -e^T \alpha e - r^T K_r r - k_s y_s^2 \tag{32}$$

where (3) was also utilized. Notice that, we can upper bound (32) as

$$\dot{V}_1 \leq -\lambda_3 \|z\|^2 \tag{33}$$

where $\lambda_3 \triangleq \min\{\lambda_{\min}(\alpha), \lambda_{\min}(K_r), k_s\}$ with $\lambda_{\min}(\cdot)$ denoting the minimum eigenvalue of a matrix.

From (27), (28) and (32), (33), it is easy to see that $V_1(e, r, y_s)$ is exponentially stable [26]. Therefore $z(t)$ and thus $e(t)$, $r(t)$, $y_s(t)$ are exponentially stable. So, $V_1(e, r, y_s) \in \mathcal{L}_\infty$, and thus $z(t)$, $e(t)$, $r(t)$, $y_s(t) \in \mathcal{L}_\infty$. Based on the boundedness of the desired task–space trajectory, from (9), it is clear that $x(t) \in \mathcal{L}_\infty$. In view of Remark 2, boundedness of $e(t)$ and $r(t)$ can be utilized along with (12) to conclude that $\dot{x}(t) \in \mathcal{L}_\infty$. Above boundedness statements can be utilized with (11) to prove that $\dot{\theta}(t) \in \mathcal{L}_\infty$. The above boundedness statements can be utilized with (15) to prove that $\dot{r}(t) \in \mathcal{L}_\infty$ where Remark 2 was utilized. The robot manipulator dynamics in (1) can be utilized to demonstrate $\tau(t) \in \mathcal{L}_\infty$. Standard signal chasing arguments can then be used to prove that all signals remain bounded under the closed–loop operation. ■

Remark 5: The Lyapunov functions for task–space tracking (*i.e.*, V_t) and sub–task objective (*i.e.*, V_s) and their time derivatives in [20] and [21] expressed by using the notation in this paper are of the following forms

$$V_t = \frac{1}{2} e^T e + \frac{1}{2} r^T M r \text{ with } \dot{V}_t = -e^T \alpha e - r^T K_r r$$

$$V_s = \frac{1}{2} y_s^2 \text{ with } \dot{V}_s \leq -\gamma y_s^2 + \epsilon$$

where ϵ, γ are positive scalar constants. While the exact model knowledge version of the task–space controller of [20] and [21] provides exponential stability, their null–space controller achieves ultimately bounded sub–task tracking provided in the following sense

$$|y_s| \leq \sqrt{|y_s^2(t_0)| \exp(-2\gamma t) + \frac{\epsilon}{\gamma}}.$$

On the other hand, our proposed combined controller achieves asymptotic task–space tracking and asymptotic sub–task tracking.

A. Adaptive Controller Extension

In this section, the exact model knowledge controller of (17) is modified to compensate for parametric uncertainties in the dynamic model. We design the adaptive controller as

$$\tau = Y\hat{\phi} + K_r r + J^T e - y_s J_s^T \tag{34}$$

where $\hat{\phi}(t) \in \mathbb{R}^p$ is the estimate of the uncertain parameter vector ϕ, and is updated according to

$$\dot{\hat{\phi}} = \Gamma Y^T r \tag{35}$$

where $\Gamma \in \mathbb{R}^{p\times p}$ is a constant positive definite diagonal adaptive gain matrix.

The only difference between the exact model knowledge controller and the adaptive controller is the time–varying update rule $\hat{\phi}(t)$ introduced to compensate for the lack of accurate knowledge of dynamic model parameters.

The parameter estimation vector $\widetilde{\phi}(t) \in \mathbb{R}^p$ is defined as

$$\widetilde{\phi} \triangleq \phi - \hat{\phi}. \tag{36}$$

Substituting (34) and (36) into (15), the closed–loop error system for $r(t)$ can be written as follows

$$M\dot{r} = -Cr - K_r r - J^T e + y_s J_s^T + Y\widetilde{\phi}. \tag{37}$$

Since the dynamics of task–space tracking error and sub–task function do not depend on the dynamic model parameters they remain unchanged. Now we can state the stability theorem for the adaptive controller.

Theorem 2: The adaptive controller in (34) along with the sub–task controller in (23) and the parameter update law in (35) ensures asymptotic task–space tracking and asymptotic sub–task control in the sense that

$$\|e(t)\| \to 0 \ , \ |y_s(t)| \to 0 \text{ as } t \to \infty. \tag{38}$$

Proof: We start our proof by defining a positive valued scalar function, denoted by $V_2(e, r, y_s, \widetilde{\phi}) \in \mathbb{R}$, as

$$V_2 \triangleq V_1 + \frac{1}{2}\widetilde{\phi}^T \Gamma^{-1} \widetilde{\phi} \tag{39}$$

where $V_1(e, r, y_s)$ was defined in (27). Similar to (28), following bounds can be obtained for V_2

$$\lambda_4 \|s\|^2 \leq V_2 \leq \lambda_5 \|s\|^2 \tag{40}$$

where $\lambda_4 \triangleq \frac{1}{2}\min\{1, m_1, \lambda_{\max}(\Gamma)\}$, $\lambda_5 \triangleq \frac{1}{2}\max\{1, m_2, \lambda_{\min}(\Gamma)\}$ with $\lambda_{\max}(\cdot)$ denoting the maximum eigenvalue of a matrix, and $s(t) \in \mathbb{R}^{(m+n+p+1)\times 1}$ is the combined error vector defined as

$$s(t) \triangleq \begin{bmatrix} e^T & r^T & y_s & \widetilde{\phi}^T \end{bmatrix}^T = \begin{bmatrix} z^T & \widetilde{\phi}^T \end{bmatrix}^T. \tag{41}$$

After taking the time derivative of (39), we obtain

$$\dot{V}_2 = e^T\dot{e} + r^T M\dot{r} + \frac{1}{2}r^T\dot{M}r + y_s\dot{y}_s + \widetilde{\phi}^T\Gamma^{-1}\dot{\widetilde{\phi}}. \tag{42}$$

Substituting (12), (26), (37), time derivative of (36) along with (35), and then canceling common terms, we obtain the following expression

$$\dot{V}_2 = -e^T\alpha e - r^T K_r r - k_s y_s^2 \leq -\lambda_3 \|z\|^2 \tag{43}$$

where λ_3 was introduced in (33) and skew symmetry property in (3) was utilized.

From (39), (40) and (43), $V_2(e, r, y_s, \widetilde{\phi}) \in \mathcal{L}_\infty$, therefore, $s(t)$, and thus $e(t)$, $r(t)$, $y_s(t)$, $\widetilde{\phi}(t) \in \mathcal{L}_\infty$. Similar to the proof of Theorem 1, all the signals can be shown to remain bounded under the closed–loop operation. After integrating (43) in time from 0 to $+\infty$, we obtain

$$\int_0^{+\infty} \dot{V}_2(t)dt \leq -\lambda_3 \int_0^{+\infty} \|z(t)\|^2 dt \tag{44}$$

and after some mathematical manipulations

$$\int_0^{+\infty} \|z(t)\|^2 dt \leq \frac{V_2(0)}{\lambda_3} \tag{45}$$

from which we can see that $z(t)$ is square integrable. Barbalat's Lemma [27] can then be utilized to obtain asymptotic task–space tracking and asymptotic sub–task control results in (38). ■

V. Simulation Results

To illustrate the performance of the task–space tracking with sub–task controller, numerical simulation results are presented on a 3 dof planar robot manipulator with the following dynamics [15]

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{12} & M_{22} & M_{23} \\ M_{13} & M_{23} & M_{33} \end{bmatrix} \quad F_d = \begin{bmatrix} f_{d1} & 0 & 0 \\ 0 & f_{d2} & 0 \\ 0 & 0 & f_{d3} \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad G(\theta) = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}$$

where the entries of the gravitational effect G_i, the inertia matrix M_{ij}, and the centripetal–Coriolis matrix C_{ij}, $i, j \in \{1, 2, 3\}$ (which are formed to satisfy Property 2) are given as follows

$$\begin{aligned} M_{11} &= \beta_1 + 2p_1c_2 + 2p_2c_{23} + 2p_3c_3 & M_{22} &= \beta_2 + 2p_3c_3 \\ M_{12} &= \beta_2 + p_1c_2 + p_2c_{23} + 2p_3c_3 & M_{23} &= \beta_2 + p_3c_3 \\ M_{13} &= \beta_2 + p_2c_{23} + p_3c_3 & M_{33} &= \beta_3 \end{aligned}$$

$$\begin{aligned} C_{11} &= -(p_1s_2 + p_2s_{23})\dot{\theta}_2 - (p_2s_{23} + p_3s_3)\dot{\theta}_3 \\ C_{12} &= -(p_1s_2 + p_2s_{23})(\dot{\theta}_1 + \dot{\theta}_2) - (p_2s_{23} + 2p_3s_3)\dot{\theta}_3 \\ C_{13} &= -(p_2s_{23} + p_3s_3)(\dot{\theta}_1 + \dot{\theta}_3) - p_2s_{23}\dot{\theta}_2 \end{aligned}$$

$$C_{21} = (p_1 s_2 + p_2 s_{23})\dot{\theta}_1 \qquad C_{22} = -p_3 s_3 \dot{\theta}_3$$
$$C_{23} = -p_3 s_3 (2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \qquad C_{31} = (p_2 s_{23} + p_3 s_3)\dot{\theta}_1$$
$$C_{32} = p_3 s_3 (2\dot{\theta}_1 + \dot{\theta}_2) \qquad C_{33} = 0$$

$$\begin{aligned} G_1 &= 0.5 m_{s1} g l_1 c_1 + m_{s2} g (l_1 c_1 + 0.5 l_2 c_{12}) \\ &+ m_{s3} g (l_1 c_1 + l_2 c_{12} + 0.5 l_3 c_{123}) \\ G_2 &= 0.5 m_{s2} g l_2 c_{12} + m_{s3} g (l_2 c_{12} + 0.5 l_3 c_{123}) \\ G_3 &= 0.5 m_{s3} g l_3 c_{123} \end{aligned}$$

where β_1, β_2, β_3, p_1, p_2, p_3, m_{s1}, m_{s2}, m_{s3} denote the model parameters (*i.e.*, mass of the links, center of mass each link, link lengths), f_{d1}, f_{d2}, f_{d3} denote the dynamic friction coefficients, s_i, c_i, $s_{i,j}$, $c_{i,j}$ represents $\sin(\theta_i)$, $\cos(\theta_i)$, $\sin(\theta_i + \theta_j)$, $\cos(\theta_i + \theta_j)$ $(i, j \in \{1, 2, 3\})$, respectively. For simulation purposes, the masses of links 1, 2 and 3 were selected to be $m_{s1} = 3.60$Kg, $m_{s2} = 2.60$Kg and $m_{s3} = 2.00$Kg, respectively. The corresponding link lengths were selected as $l_1 = 0.4$m, $l_2 = 0.36$m, $l_3 = 0.30$m, respectively, and the center of mass of each link was selected to be the midpoint of each link. The gravitational constant was selected as $g = 9.81 \frac{\text{m}}{\text{sec}^2}$. Given the above numerical values, the following values for the model parameters with unit Kg.m^2 and the friction parameters with unit Nm.sec were used in the simulation

$$\begin{array}{ccc} \beta_1 = 1.1956 & \beta_2 = 0.3946 & \beta_3 = 0.0512 \\ p_1 = 0.4752 & p_2 = 0.1280 & p_3 = 0.1152 \\ f_{d1} = 5.3 & f_{d2} = 2.4 & f_{d3} = 1.1. \end{array}$$

The constant parameter vector defined in (4) was constructed as follows

$$\phi = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & p_1 & p_2 & p_3 & f_{d1} & f_{d2} & f_{d3} \end{bmatrix}^T. \tag{46}$$

The end–effector position of the manipulator in the task–space can be obtained to have the following form

$$x(t) = \begin{bmatrix} X(t) \\ Z(t) \end{bmatrix} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ l_1 s_1 + l_2 s_{12} + l_3 s_{123} \end{bmatrix}. \tag{47}$$

Based on the above forward kinematic calculations, the manipulator Jacobian is obtained as

$$J = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & l_1 c_1 + l_2 c_{12} + l_3 c_{123} \\ -l_2 s_{12} - l_3 s_{123} & l_2 c_{12} + l_3 c_{123} \\ -l_3 s_{123} & l_3 c_{123} \end{bmatrix}^T \tag{48}$$

with $c_{123} \triangleq \cos(\theta_1 + \theta_2 + \theta_3)$, $s_{123} \triangleq \sin(\theta_1 + \theta_2 + \theta_3)$.

The manipulator was initialized to be at rest at the following joint position $\theta(0) = \begin{bmatrix} 0.1 & 0.5 & 1.5 \end{bmatrix}$rad. The desired task–space trajectory, for all simulations, was selected as follows

$$x_d = \begin{bmatrix} X_d \\ Z_d \end{bmatrix} = \begin{bmatrix} 0.5 + 0.1\cos(t)(1 - \exp(-0.3t^3)) \\ 0.5 + 0.1\sin(t)(1 - \exp(-0.3t^3)) \end{bmatrix}.$$

Two sets of simulations were performed for the proposed adaptive controller. The first set of simulation was performed without sub–task control. In the second set of simulation, following sub–task function was selected

$$y_s = \theta_2 - 1 \tag{49}$$

with which the position of the second link is forced to go to 1 rad.

The best results for the adaptive controller of (34) were obtained with the following control and adaptation gains; $\alpha = 10I_2$, $K_r = 20I_3$, $\Gamma = \text{diag}\{\begin{array}{ccccccccc} 0.1 & 0.75 & 1.5 & 0.035 & 1.1 & 0.05 & 40 & 30 & 30 \end{array}\}$ for the first set of simulation, and $k_s = 5$ for the second set of simulation. It is highlighted that in the adaptive controller simulation the model parameters were utilized to simulate the robot dynamics and were not utilized in the controller formulation as can be seen in (34).

Figure 1 shows the task–space tracking errors for two sets of simulations; Figure 2 shows the joint positions for two sets of simulations; Figure 3 shows the desired and actual task–space trajectories. From Figure 1, it is clear that task–space tracking objective was met for both simulations. From Figure 2, it is clear that when there is sub–task control the position of the second link went to 1 rad thus satisfying the sub–task function in (49).

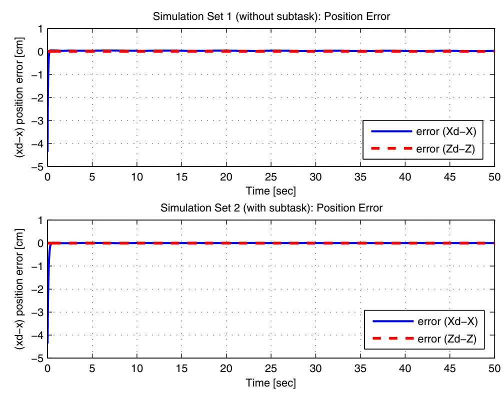

Fig. 1. Task–space end–effector position tracking errors $e(t)$.

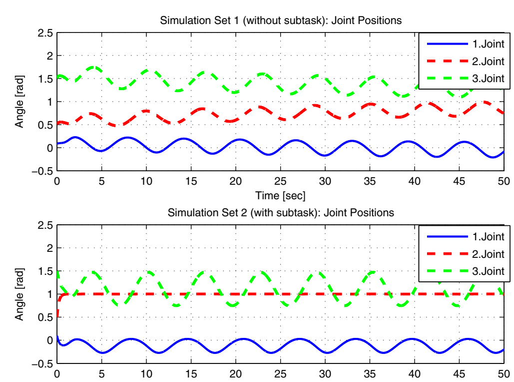

Fig. 2. Joint positions $\theta(t)$.

VI. Conclusions

In this paper, we mainly focused on an asymptotically stable sub–task control design and presented the design and the corresponding analysis of two task–space controllers (one

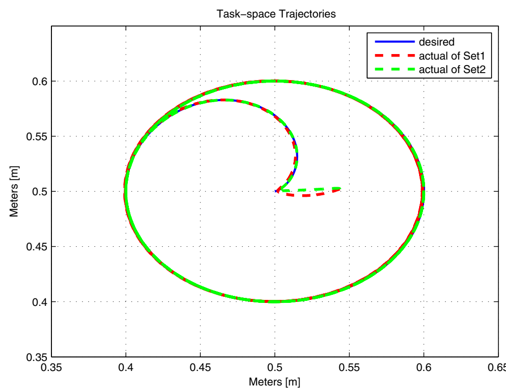

Fig. 3. Desired $x_d(t)$ and actual $x(t)$ task–space trajectories.

exact model knowledge and one adaptive) for redundant robot manipulators that utilizes the extra degrees of freedom of the redundant robot for a predefined sub–task. The sub–task function introduced, has been designed as an error–like function of joint positions. A combined stability analysis that ensures the stability and convergence of both end–effector and sub–task function have been presented. To our best knowledge, the proposed combined analysis is novel when compared to the existing literature on control of redundant robot manipulators. Convergence of the task–space tracking error and the sub–task function to the origin were ensured. Numerical simulations were conducted for the adaptive controller with and without sub–task controllers.

A possible application area of the proposed strategy is when the robot manipulator (possibly a hyper redundant one) is equipped with a camera or a sensor that needs to be kept at a specific pose while performing end–effector motion. In that case, that particular point can be written as a function of the joint positions and the proposed sub–task controller can be utilized to ensure the task.

As future work, experimental verification of the proposed control strategy is aimed. We are currently in the process of developing our own 3 dof planar redundant robot manipulator where we will test the proposed controllers.

A shortcoming of the proposed controllers may be seen as lack of compensation for unstructured uncertainties (*i.e.*, disturbances). In this paper, design of a novel sub–task controller was emphasized so the dynamic model (fully or partially) was considered to be known and thus no disturbance term was considered. It is obvious that, any realistic model should include a disturbance term to represent unmodeled dynamics. So, design of a robustifying term to be fused with the proposed controllers will also be done.

REFERENCES

[1] E. S. Conkur and R. Buckingham, "Clarifying the definition of redundancy as used in robotics," *Robotica*, vol. 15, pp. 583–586, 1997.
[2] O. Khatib, "Dynamic control of manipulators in operational space," in *IFTOMM Cong. Theory of Machines and Mechanisms*, New Delhi, India, 1983, pp. 1–10.
[3] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Trans. Robot. Autom.*, vol. 5, no. 4, pp. 472–490, 1989.
[4] R. Colbaugh and K. Glass, "Robust adaptive control of redundant manipulators," *J. Intelligent and Robotic Systems*, vol. 14, no. 1, pp. 68–88, 1995.
[5] Z. Peng and N. Adachi, "Compliant motion control of kinematically redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 9, no. 6, pp. 831–837, 1993.
[6] Z. Duan, L. Huang, Y. Yao, and Z.-P. Jiang, "On the effects of redundant control inputs," *Automatica*, vol. 48, pp. 2168–2174, 2012.
[7] I. Walker, "Impact configurations and measures for kinematically redundant and multiple armed robot systems," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 670–683, 1994.
[8] D. Nenchev, "Redundancy resolution through local optimization: A review," *J. of Robotic Systems*, vol. 6, no. 6, pp. 769–798, 1989.
[9] A. Kapadia, E. Tatlicioglu, and D. M. Dawson, "Set-point navigation of a redundant robot in uncertain environments using finite range sensors," in *Proc. IEEE Int. Conf. Decision and Control*, Cancun, Mexico, 2008, pp. 4596–4601.
[10] D. Braganza, M. L. McIntyre, D. M. Dawson, and I. D. Walker, "Whole arm grasping control for redundant robot manipulators," in *Proc. American Control Conf.*, Minneapolis, MN, USA, 2006, pp. 3194–3199.
[11] B. Xian, M. S. de Queiroz, D. M. Dawson, and I. D. Walker, "Task-space tracking control of robot manipulators via quaternion feedback," *IEEE Tr. on Robotics and Automation*, vol. 20, no. 1, pp. 160–167, 2004.
[12] J. Nakanishi, R. Cory, M. M. J. Peters, and S. Schaal, "Operational space control: A theoretical and emprical comparison," *Int. J. Robotics Research*, vol. 27, no. 6, pp. 737–757, 2008.
[13] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," *J. of Robotic Systems*, vol. 6, no. 3, pp. 133–148, 1989.
[14] S. Luo and S. Ahmad, "Adaptive control of kinematically redundant robots," *IMA J. of Mathematical Control & Information*, vol. 14, pp. 225–253, 1997.
[15] E. Zergeroglu, D. Dawson, I. D. Walker, and A. Behal, "Nonlinear tracking control of kinematically redundant robot manipulators," in *Proc. American Control Conf.*, Chicago, Illinois, 2000, pp. 2513–2517.
[16] E. Zergeroglu, D. M. Dawson, I. D. Walker, and P. Setlur, "Nonlinear tracking control of kinematically redundant robot manipulators," *IEEE/ASME Tr. on Mechatronics*, vol. 9, no. 1, pp. 129–132, 2004.
[17] K. P. Tee and R. Yan, "Adaptive operational space control of redundant robot manipulators," in *Proc. American Control Conf.*, San Francisco, CA, USA, 2011, pp. 1742–1747.
[18] E. Zergeroglu, H. T. Sahin, U. Ozbay, and U. A. Tektas, "Robust tracking control of kinematically redundant robot manipulators subject to multiple self-motion criteria," in *IEEE Int. Conf. on Control App.*, Munich, Germany, 2006, pp. 2860–2865.
[19] H. T. Sahin, U. Ozbay, and E. Zergeroglu, "Quaternion based robust tracking control of kinematically redundant manipulators subject to multiple self-motion criteria," in *Proc. IEEE Int. Conf. Decision and Control*, San Diego, CA, USA, 2006, pp. 6462–6467.
[20] E. Tatlicioglu, M. L. McIntyre, D. M. Dawson, and I. D. Walker, "Adaptive nonlinear tracking control of kinematically redundant robot manipulators with sub-task extensions," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, Dec 2005, pp. 4373–4378.
[21] ——, "Adaptive non-linear tracking control of kinematically redundant robot manipulators," *Int. J. Robot. Autom.*, vol. 23, no. 2, pp. 98–105, Mar. 2008.
[22] E. Tatlicioglu, D. Braganza, T. C. Burg, and D. M. Dawson, "Adaptive control of redundant robot manipulators with sub-task extensions," in *Proc. American Control Conf.*, Seattle, WA, 2008, pp. 856–861.
[23] ——, "Adaptive control of redundant robot manipulators with sub-task objectives," *Robotica*, vol. 27, pp. 873–881, 10 2009.
[24] F. Lewis, D. Dawson, and C. Abdallah, *Robot Manipulator Control: Theory and Practice*. New York, NY, USA: Marcel Dekker, 2004.
[25] Y. Nakamura, *Advanced Robotics Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1991.
[26] H. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2002.
[27] M. Krstic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. New York, NY, USA: John Wiley & Sons, 1995.