2012 12th International Conference on Control, Automation and Systems
Oct. 17-21, 2012 in ICC, Jeju Island, Korea

General Subtask Controller for Redundant Robot Manipulators

Omar W. Maaroof[1*], Erkin Gezgin[2] and Mehmet İsmet Can Dede[3]

[1] Department of Mechatronics Engineering, Mosul University, Mosul, Iraq
(Tel : +96-4-770-2044149; E-mail: omarmaaroof@gmail.com)* Corresponding author
[2]Department of Robotics Engineering, DGIST, South Korea
(Tel : +82 53 785 6230; E-mail: erkin@dgist.ac.kr)
[3]Department of Mechanical Engineering, İzmir Institute of Technology, 35430 İzmir, Turkey
(Tel : +90 (232) 750-6778; E-mail: candede@iyte.edu.tr)

Abstract: This paper will utilize the property of self-motion for redundant robot manipulators by designing the general subtask controller that control the joint motion in the null-space of the Jacobian matrix. The general subtask controller is used for minimizing the total joint motion and for singularity avoidance in this study. Specifically, objective function for each subtask is formed and then the gradient of the objective function is used in the subtask controller to either minimize the joint motion or avoid singularities while tracking a given end-effector trajectory. A **7-DOF LWA4-Arm (SCHUNK)** is modeled first in SolidWorks® and then converted to MATLAB® Simulink using SimMechanics CAD translator for the simulation tests of the controller. The kinematics and dynamics equations are derived to be used in the controllers and the simulation results are presented for the 7-DOF redundant robot manipulator operating in 3D space.

Keywords: Redundant robot manipulators, subtask control, self-motion, singularity avoidance, minimizing the motion.

1. INTRODUCTION

Redundant manipulators provide increased flexibility for the execution of complex tasks such as obstacle avoidance, specific configuration with respect to the task, etc. Therefore, it has been the subject of significant research in the last decades. In an industrial environment, the redundant robots are expected to reduce the manufacturing costs, increase the productivity, and possibly improve the safety of human co-operators. Redundancy of such manipulators should be effectively used in the near future in medical, assistive and rehabilitation robotics applications to improve the safety by answering specific configuration needs during the manipulation.

Self-motion control of kinematically redundant manipulators has been the subject of intensive research in years. The extra DOFs have been used to satisfy specific additional tasks. Most methods for resolving redundancy in manipulation involve defining an objective function, such as obstacle avoidance [1, 2], mechanical joint-limit avoidance, optimization of user-defined objective functions, and minimization of joint velocities and accelerations [3]. Gertz et al. [4], Walker [5] and Lin et al. [6] have used a generalized inertia-weighted inverse of the Jacobian to resolve redundancy in order to reduce impact forces. In [7] the manipulability measure is used, which has a minimum or maximum value at a desirable configuration. Then, for a given end-effector pose, using the gradient (or its negative) of objective function to control joint velocity in the redundant directions, this motion is referred to *self-motion* since it is not observed in the task space [18], the manipulator will seek the optimal configuration [7, 8]. While many authors have discussed how to specify such joint velocities (this might be termed kinematic control), others provided a dynamic feedback linearizing control law that guarantees the tracking of these redundant joint velocities while providing end effector tracking [9, 10]. Other works on dynamic control laws for redundant manipulators include approaches for redundancy resolution through torque optimization [11, 12] and the task-space space approach [13].

The motivation of this work is to design the general subtask controller that can track end-effector pose while satisfying subtask objectives. The subtask objectives that are considered separately in this study are minimization of the total joint motion and the singularity avoidance.

In spite of the challenging nature of solving the kinematics problem and defining the dynamics model for 7-DOF robot manipulator, the redundant 7-DOF spatial case is taken as a benchmark to gain insight toward a more general and realistic situation.

An existing redundant robot arm, 7-DOF LWA4-Arm by SCHUNK, is selected to be used to employ the developed controller. The robot arm is first modeled in a CAD and then transferred to the simulation environment. The control studies are carried out in the simulation environment and simulation test results are presented to verify the validity of the designed subtask controllers.

2. MODELING THE ROBOTIC ARM

Modeling is done using the virtual prototyping of robot controllers' method [17] in two stages. First, robot arm is modeled by SolidWorks software with respect to the CAD data provided in [16]. COSMOSMotion is used to develop the mechanism by assigning the joints in CAD environment. Then, the CAD model is exported in 3D XML format by using the plug-in, SimMechanics Link, to MATLAB® Simulink. As a result of the transfer of the model from CAD environment to SimMechanics software, the model could be used for simulation studies that are conducted to test the controller.

Second stage includes the modeling of the control system and development of the necessary kinematics and dynamics equations for the robot using MATLAB®

978-89-93215-04-5 95560/12/$15 ⓒICROS

Simulink blocks. The visualization tools of SimMechanics software is also used to display and animate 3D machine geometries, before and during simulation.

3. KINEMATICS AND DYNAMICS MODEL

3.1 Kinematics model

Redundant manipulators have a larger number of DOF, *n,* than the dimension of the workspace, *m*. The end-effector position and orientation in the operation space, denoted by $x(t) \in \Re^m$, is defined as a function of joint position vector as

$$x = k(q) = \begin{bmatrix} p(q) \\ \phi(q) \end{bmatrix}, \quad (1)$$

where $k(q) \in \Re^m$, $m \in \aleph$ is the forward kinematic calculation, $q(t) \in \Re^n$ denote the link position vector of an n-link manipulator $p(q) \in \Re^l$ and $\phi(q) \in \Re^{(m-l)}$ are the vectors representing the end-effector position, and orientation respectively and $l \in \aleph$ is the size of the task space for positioning.

Based on Eq. (1), the differential relationships between the end-effector position and the link position variables are obtained as follows;

$$\dot{x} = J(q)\dot{q}, \quad (2)$$

$$\ddot{x} = \dot{J}(q)\dot{q} + J(q)\ddot{q}, \quad (3)$$

where $J(q) = \partial k(q) / \partial q \in \Re^{m \times n}$, is the Jacobian matrix of the manipulator and $\dot{q}(t), \ddot{q}(t) \in \Re^n$ denote the link velocity and acceleration vectors, respectively. Since *J* is not square for redundant manipulators (***m*** < ***n***), we must use the ***psuedo-inverse***, J^+ defined in Eq. (7), to obtain the inverse relations

$$\dot{q} = J^+\dot{x} + \dot{\theta}_N, \quad (4)$$

$$\ddot{q} = J^+(\ddot{x} - \dot{J}\dot{q}) + \ddot{\theta}_N, \quad (5)$$

where $\dot{\theta}_N$ & $\ddot{\theta}_N$ are vectors of joints velocity and acceleration in the null space of *J*. The psuedo-inverse J^+, is defined as the unique matrix such that [7, 14]

$$\begin{aligned} JJ^+J &= J, & J^+JJ^+ &= J^+, \\ (J^+J)^T &= J^+J, & (JJ^+)^T &= JJ^+. \end{aligned} \quad (6)$$

When *J* has full rank (the manipulator is not in a singular configuration), the pseudo-inverse can be calculated as;

$$J^+ = J^T(JJ^T)^{-1}, \quad (7)$$

so that J^+ satisfies $JJ^+ = I_m$ (I_m is *m* x *m* identity matrix).

3.2 Dynamic model

The dynamic model for an *n*-link, all revolute-joint robot manipulator is developed in the following form [15]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \xi_d = \tau, \quad (8)$$

where $M(q) \in \Re^{n \times n}$ represents the inertia matrix, $C(q,\dot{q}) \in \Re^{n \times n}$ represents the centripetal-Coriolis matrix, $G(q) \in \Re^n$ is the gravity vector, $F(\dot{q}) \in \Re^n$ represents the friction effects vector, $\xi_d \in \Re^n$ is a vector containing the unknown but bounded, additive disturbance effects and $\tau(t) \in \Re^n$ is the torque input vector.

4. CONTROL OBJECTIVE

The objective is to design the control torque input signal τ (*t*) such that the robot end-effector can follow a desired end-effector position demand as closely as possible. The control signal should also include enough information to execute subtasks defined by at least one motion optimization measures. From now on, the task space tracking will be referred as ***main objective*** and enabling the use of manipulators redundancy in optimization as secondary or ***subtask objective***.

4.1 Main-Task Control Objective

Let the control τ (*t*) be given by

$$\tau = M\left\{J^+(\ddot{x}_d + K_v\dot{e} + K_p e - \dot{J}\dot{q}) + \ddot{\theta}_N\right\} + N, \quad (9)$$

where x_d is the desired position defined in operation space, $e = x_d - x$ is the tracking error, *Kv* and *Kp* are constant feedback gain matrices, *N* is the calculated nonlinear terms that appear in the dynamics equation of the robot and $\ddot{\theta}_N$ is designed joint acceleration vector in the null space of *J*. If the manipulator does not go through a singularity, then the control law in Eq. (9) guarantees that the tracking error converges to zero exponentially.

Proof: The closed loop system is given by

$$M\ddot{q} + N = M\left\{J^+(\ddot{x}_d + K_v\dot{e} + K_p e - \dot{J}\dot{q}) + \ddot{\theta}_N\right\} + N, \quad (10)$$

which is simplified to

$$\ddot{q} = J^+(\ddot{x}_d + K_v\dot{e} + K_p e - \dot{J}\dot{q}) + \ddot{\theta}_N, \quad (11)$$

since *M* is uniformly positive definite. Combining Eq. (5) with Eq. (11), the equation is modified to

$$\ddot{e} + K_v\dot{e} + K_p e = 0, \quad (12)$$

since $JJ^+ = I$ when *J* has full rank. The proper choice of K_v and K_p (e.g. $K_v = k_v I$ and $K_p = k_p I$ with $s^2 + k_v s + k_p$ a *Hunvitz polynomial*) in Eq. (12) implies that *e* goes to zero exponentially.

4.2 Sub-Task Control Objective

We consider the case where we are given a vector function $g(\cdot) \in \Re^n$, (which may be a function of time, the current state, etc.) and we want the null space joint velocity to track the projection of *g* onto the null space of *J*. Since ($I_n - J^+J$) projects vectors onto the null space of *J*, this can be formulated in an error signal calculation,

$$\dot{e}_N \equiv (I - J^+J)g - \dot{\theta}_N, \quad (13)$$

which converges to zero.

Assuming the manipulator does not go through a singularity condition, it is needed to design $\ddot{\theta}_N$ to get the desired result for subtask objective. Let missing part the control given in Eq. (9), $\ddot{\theta}_N$, be determined as;

$$\ddot{\theta}_N = (I - J^+J)\dot{g} - (J^+\dot{J}J^+ + \dot{J}^+)Jg + K_N\dot{e}_N, \quad (14)$$

where K_N is a positive definite feedback matrix. Then the joint velocities in the null space converge to $(I - J^+J)g$, i.e., $\dot{e}_N$, and the tracking error e (as defined in Section 4.1) converges to zero.

Proof: First note that $\ddot{\theta}_N$ given by Eq. (14) belongs to the *null space* of J (when J has full rank) since

$$J(I - J^+J) = J - J = 0, \quad (15)$$

$$J(J^+\dot{J}J^+ + \dot{J}^+) = \dot{J}J^+ + J\dot{J}^+ = \frac{d}{dt}(JJ^+) = \frac{d}{dt}I = 0. \quad (16)$$

$\dot{e}_N$ is in the *null space* of J as well, and therefore,

$$JK_N\dot{e}_N = 0. \quad (17)$$

Eqs. (15) ~ (17). prove that the designed equation, Eq. (14), in null space will not affect the main task objective.

Now, let's consider the second derivative of the error that is calculated by taking the derivative of Eq. (13) with respect to time,

$$\ddot{e}_N = (I - J^+J)\dot{g} - (\dot{J}^+J + J^+\dot{J})g - \ddot{\theta}_N. \quad (18)$$

Adding and subtracting $J^+\dot{J}J^+Jg$, to Eq. (18), and substituting $\ddot{\theta}_N$ from Eq. (14), Eq. (19) is derived.

$$\ddot{e}_N = -J^+\dot{J}g + J^+\dot{J}J^+Jg - K_N\dot{e}_N \quad (19)$$

By defining the following *non-negative* scalar Lyapunov function, the designed Eq. (14) can be proven to be stable as follow:

$$v = \frac{1}{2}\dot{e}_N^{\,2}, \quad (20)$$

then

$$\dot{v} = \dot{e}_N^{\,T}\ddot{e}_N,$$
$$= \dot{e}_N^{\,T}\left(-J^+\dot{J}g + J^+\dot{J}J^+Jg - K_N\dot{e}_N\right),$$
$$= -\dot{e}_N^{\,T}K_N\dot{e}_N. \quad (21)$$

In Eq. (21), the following equality in Eq. (22) is used that are formed by using Eq. (13) and $\dot{\theta}_N$ from Eq. (4);

$$\dot{e}_N^{\,T} = (g - \dot{q})^T(I - J^+J), \quad (22)$$

The equalities represented below are the mathematical proofs that were used to calculate for Eq. (21).

$$(I - J^+J)^T = (I - J^+J), \quad (23)$$

$$(I - J^+J)(I - J^+J) = (I - J^+J), \quad (24)$$

$$(I - J^+J)J^+ = 0. \quad (25)$$

Since v is positive definite and $\dot{v}$ is negative definite, $\|\dot{e}_N\|$, goes to zero, exponentially.

5. SUBTASKS

The projection of function g into the null space of J can be considered as the desired null space joint velocities that are needed to accomplish a given subtask. To control self-motion of the joint velocities, the gradient g (or its negative) of the objective function *f(q)* can be used as,

$$g = \nabla f. \quad (26)$$

Many authors have discussed the selection of the null space joint velocity for the purpose of avoiding singularity, joint limit avoidance, obstacle avoidance, minimizing potential energy, impact force configuration and achieving other subtasks. In the next subsections, some subtasks techniques that are going to be used in this work will be presented.

5.1 Joints Motion Minimization

By using the stated principles above, the first sub-task objective is based on minimizing joint motions for a 7-DOF redundant manipulator where the norm (or length) of joint velocity vector, $\|\dot{q}\|$, will minimized, in addition to the Main-Task objective, which is tracking control. To achieve this subtask, *f(q)* is chosen to be zero then, as long as the manipulator is not in a singularity configuration, the null space velocity, $(I - J^+J)g$ will go to zero. However, this choice of control makes no provision for avoiding singularities [11]. If the manipulator gets to a singularity configuration, the control law given by Eq. (9) and Eq. (14) is no longer defined since J^+ is discontinuous.

5.2 Manipulability or Singularity Avoidance

The performance criterion function is selected according to the manipulability measure presented in [7]. The objective function is chosen as,

$$f(q) = k\det(JJ^T), \quad (27)$$

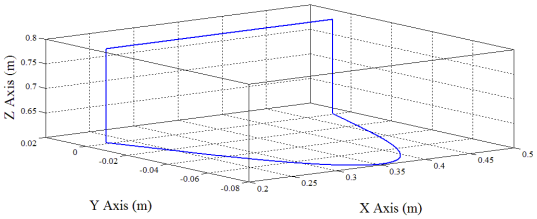
where k is the self-motion control parameter gain, det(.) denotes the determinant of matrix, $J(q)$ is the manipulator Jacobian. This objective function is based on purely robot kinematics. When the manipulator approaches its singularities, $f(q)$ decreases to zero. In order to maximize the manipulability of the manipulator, choosing the gradient as $g = \nabla f$ would keep the manipulator away from singularities.

6. SIMULATION AND RESULTS

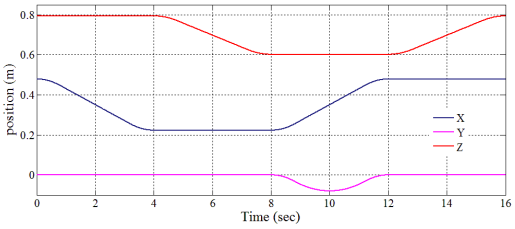
To illustrate the performance of the proposed general subtask controller presented in this work, a set of simulation results are presented in this section. In these simulations, the aim is to utilize the virtual model of 7-DOF LWA4-Arm produced by SCHUNK GmbH. The manipulator's CAD represented in Fig. 1. The manipulator model is then transferred to the simulation environment as explained in Section 2. The simulations are conducted MATLAB® Simulink simulation environment with fixed-step sample time of 0.1 kHz.

Fig. 1 The CAD Model of 7-DOF LWA4-Arm by SCHUNK GmbH

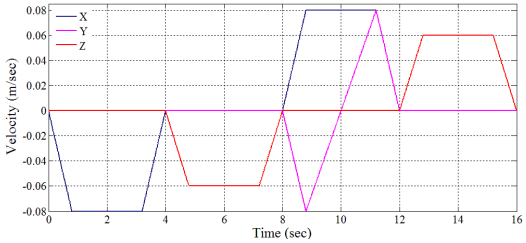
The manipulator is initialized to be at rest at the following link positions $q = [0\ -25\ 0\ -35\ 0\ 0\ 0]^T$ in degrees. Fig. 2 shows the desired task-space trajectories for all simulations. The trajectory is selected to track positions in x-y-z space and the end-effector orientation is left free. In this case, the redundant manipulator had more degrees of freedom (DOF) than is required to perform a task in the task space; hence, these extra DOFs allowed the robot manipulator to perform more dexterous manipulation and/or provided the robot manipulator system with increased flexibility for the execution of sophisticated tasks. Since the dimension, *n*, of the link position variables is **seven** and the number of the task-space variables, *m*, is **three**, then the null space of Jacobian matrix has a minimum dimension of, $n - m$, **four**.



(a)



(b)



(c)

Fig. 2 Desired task-space trajectories: a) 3D task space trajectory, b) desired position trajectory, c) desired velocity

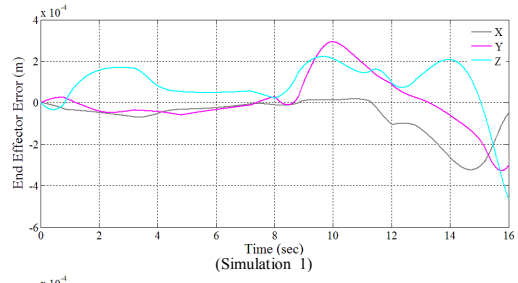
In the main task controller presented Eq. (9), the nonlinear terms that include centripetal and Coriolis $C(q,\dot{q})\dot{q}$, frictional $F(\dot{q})$ and disturbance ξ_d terms are neglected since the robot moves in slow motion. However, the gravity effects are used to form the *N* nonlinear effect cancellation term.

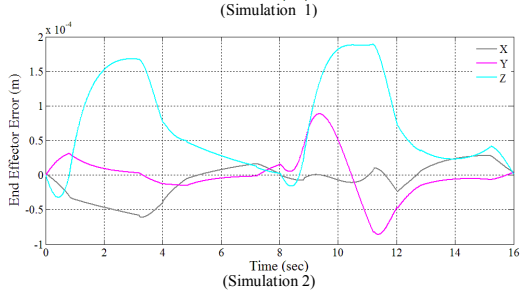
The controller parameters are tuned to the following values after some experimental tests on better performance:

$$k = 50, \qquad k_v = 100, \qquad k_p = 100,$$

$$K_N = diag\{100,\ 100,\ 100,\ 100,\ 100,\ 100,\ 100\}.$$

Three sets of simulations are performed to illustrate the performance of the proposed controller. In the first simulation, the subtask input $\ddot{\theta}_N$ is set to zero where there was no restriction on the self-motion of the robot and only the end-effector tracking objective is enforced. In the second simulation *f(q),* is set to zero to minimize total joints motion (section 5.1). In the third simulation, *f(q)* is selected to maximize the manipulability (section 5.2). Fig. 3 shows tracking error for the end-effector position for each simulation. It can be observed that the end-effector position tracking error is below 0.5mm per axis for all simulation tests.





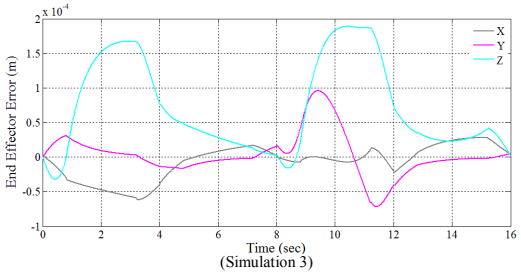


Fig. 3 End Effector position error for each Simulation.

The importance and the effect of assigning subtask objectives on the system can be observed from the link velocity trajectories in Fig. 4. Fig. 5 shows the control input torque signals calculated as a result of the general subtask controller output for each simulation.

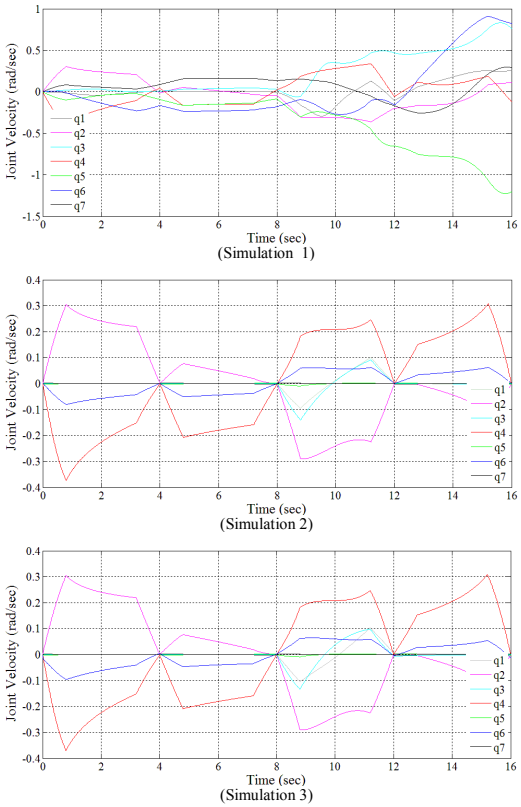
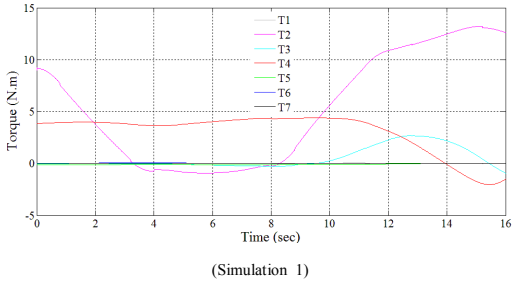


Fig. 4 Joints velocity for each Simulation.



(Simulation 1)

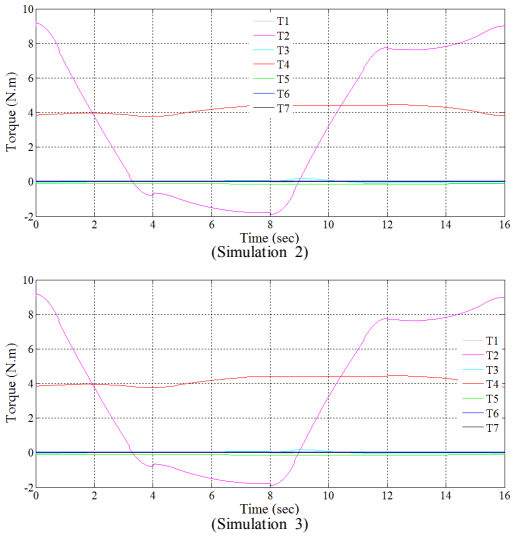


Fig. 5 The control input torque signal calculated for each simulation.

In Fig. 6, the norm (or length) of joint velocity vector is shown for the second simulation comparing with the results from the first one.

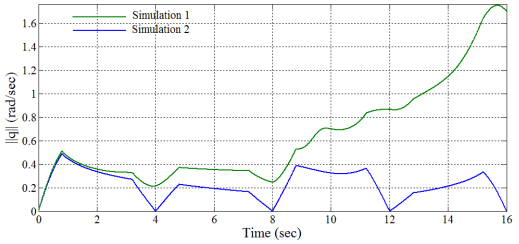


Fig. 6 Joint Motion norm.

The change of the manipulability measure during first and third simulation is presented in Fig. 7. Fig. 8 and Fig. 9 show the error signals, $\dot{e}_N$, for the second and third simulations respectively. Magnitude of the errors indicate that subtasks objective are achieved.

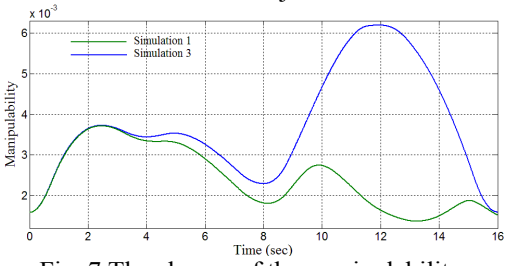


Fig. 7 The change of the manipulability.

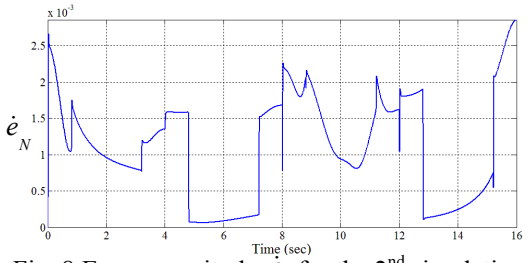


Fig. 8 Error magnitude $\dot{e}_N$ for the 2nd simulation.

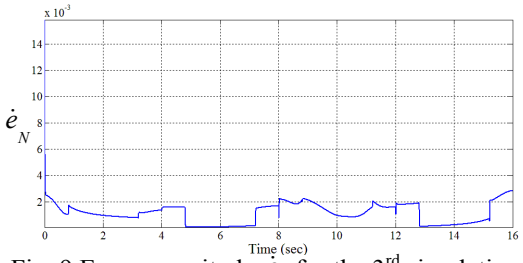


Fig. 9 Error magnitude $\dot{e}_N$ for the 3rd simulation.

7. CONCLUSION

In this paper, the general subtask controller is designed by utilizing self-motion property of redundant robot manipulator. The controller does not place any restriction on the self-motion of the manipulator, thus the extra degrees of freedom are available for subtasks like maintaining manipulability, avoidance of mechanical limits and obstacle avoidance, etc. The 7-DOF LWA4-Arm by SCHUNK GmbH is modeled for simulation test studies to validate the designed controller. Two subtasks, defined as minimization of the total joint motion and singularity avoidance, are applied separately to test the designed controller.

In all simulations, the end-effector trajectory is followed with bounded errors within the magnitude of 0.5 mm in all simulation tests. Therefore, it is valid to state that the main objective of the proposed controller for end-effector position tracking is achieved regardless of the subtask. However, a slight difference in the error magnitude can be observed for the first simulation test. Since the first simulation joint velocities came out to be higher than the others the disregarded Coriolis and centripetal forces in the controller produce more effects on the error signal.

It can also be noticed that the uncontrolled self-motion of the robot (first simulation) results in undesired motion of the links, which could cause many problems. Therefore, it can be concluded that the self-motion definitely should be controlled in redundant manipulators. Overall, it can be stated that the stability and effectiveness of the designed controller is verified by simulation results in this study.

ACKNOWLEDGMENTS

This research was supported by a Marie Curie International Reintegration Grant within the 7th European Community Framework Programme.

REFERENCES

[1] J. Baillieul, “Avoiding Obstacles and resolving kinematic redundancy”, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1698-1704, 1986.

[2] R. Colbaugh, H. Seraji, and K. Glass, “Obstacle avoidance of redundant robots using configuration control”, *Int. Journal of Robotics Research*, vol. 6, pp. 721-744, 1989.

[3] H. Seraji, “Task options for redundancy resolution using configuration control”, *30th IEEE Conf. on Decision and Control*, pp. 2793- 2798, 1991.

[4] M.W. Gertz, J. Kim, and P. Khosla, “Exploiting redundancy to reduce impact force”, *IEEE/RSJ Workshop on Intell. Rob. Sys*, pp. 179-184, 1991.

[5] I.D. Walker, “The use of kinematic redundancy in reducing impact and contact effects in manipulation”, *Proc. IEEE International Conf. Robotics and Automation*, pp. 434-439, 1990.

[6] Z. Lin, R.V. Patel, and C.A. Balafoutis, “Augmented impedance control: An approach to impact reduction for kinematically redundant manipulators”, *Journal of Robotic Systems*, vol. 12, pp. 301- 313, 1995.

[7] T. Yoshikawa, “Analysis and Control of Robot Manipulators with Redundancy”, *in Robotics Research- The First International Symposium, MIT Press, Cambridge, MA,* pp. 735-747, 1984.

[8] Maciejewski, A.A., and C.A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments", *The International Journal of Robotics Research* 4:3, 109-117, 1985.

[9] P. Hsu, J. Hauser, and S. Sastry, “Dynamic Control of Redundant Manipulators”, *Journal of Robotic Systems*, Vol. 6, pp. 133-148, 1989.

[10] E. Zergeroglu, D. M. Dawson, I. W. Walker, and P. Setlur, ”Nonlinear Tracking Control of Kinematically Redundant Robot Manipulators”, *IEEE/ASME Transactions on Mechatronics*, Vol. 9, No 1, pp 129-132,March 2004.

[11] J. Baillieul, J. Hollerbach, and R. Brockett, "Programming and Control of Kinematically Redundant Manipulators", *Proc. 23rd Conf: on Decision and Control, Las Vegas,* 768-774, 1984.

[12] Hollerbach, J.M., and K.C. Suh, "Redundancy Resolution of Manipulators through Torque Optimization". *Proc. of IEEE Int. Conf. on Robotics and Automation,* St. Louis, Missouri, 10 16-1021, 1985.

[13] O. Khatib, "Dynamic Control of Manipulators in Operational Space*", Sixth CISM-IFTOMM Congress on Theory of Machines and Mechanisms, New Delhi, India,* December 1983.

[14] Golub, G.H., and C.F. Van Loan, “*Matrix Computations*”, Baltimore, MD: The Johns Hopkins Press, 1983.

[15] M. W. Spong and M. Vidyasagar, “*Robot Dynamics and Control*”, New York: John Wiley and Sons, Inc., 1989.

[16] http://www.schunk-modular-robotics.com/left-navigation/service-robotics/service-download/simulationcad/cad-data.html.

[17] M. I. C. Dede, "Virtual Prototyping of Robot Controllers," International Journal of Design Engineering, vol. 3 (3), pp. 276-288, 2010.

[18] Y. Nakamora, *Advanced Robotics Redundancy and Optimization*, Addison-Wesley Pub. Co., MA, 1991.