

Experimental Results on Controlling a 6-DOF Robot Manipulator in the Neighborhood of Kinematic Singularities

Stefano Chiaverini Bruno Siciliano

Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
80125 Napoli, Italy

Olav Egeland

Institutt for teknisk kybernetikk
Norges tekniske høgskole
7034 Trondheim, Norway

Abstract— This work reports experimental results on kinematic control of a 6-dof industrial robot manipulator in the neighborhood of kinematic singularities. A weighted damped-least squares inverse kinematics solution with varying damping and weighting factors is devised; a feedback correction error term is added. The performance of the solutions is investigated in two case studies of critical trajectories passing by the shoulder and wrist singularities of the manipulator.

1. Introduction

Close to a kinematic singularity, the usual inverse differential kinematics solutions based on Jacobian (pseudo-)inverse become ill-conditioned, and this is experienced in the form of very high joint velocities and large control deviations. In real-time and sensory control of robotic manipulators, the reference trajectory is not known a priori and some remedies must be taken in order to counteract the unexpected occurrence of singularities. The same kind of problem is encountered in joy-stick control of a robot if the operator attempts to lead the robot through—or nearby—a singularity using end-effector motion increments.

Several methods and/or algorithms aimed at computing well-behaved or robust inverse kinematics solutions have been proposed in the literature. Most of them are based on a modification of the exact inverse differential kinematics mapping by resorting to approximate mappings that offer robustness to singularities at the expense of reduced tracking accuracy.

In the framework of our joint research project, following the earlier satisfactory experimental results obtained on the five-joint ABB Trallfa TR400

manipulator [1], in this work we investigate the performance of a number of schemes for control in singular configurations on the six-joint ABB IRb 2000 manipulator.

The basic inverse kinematics solution is derived using a weighted damped least-squares inverse [2],[3] of the end-effector Jacobian where the use of proper weighting allows shaping of the solution along given end-effector space directions. A recently proposed estimation algorithm of the two smallest singular values is utilized to compute varying damping and weighting factors [4]. Further, the introduction of a feedback correction term to avoid numerical drift instabilities [5] is proposed and its effects on solution accuracy and feasibility are discussed.

A description of the laboratory set-up is provided. Experimental case studies are illustrated with the manipulator passing nearby either a single (wrist) or a double (wrist and shoulder) singularity. The influence of the weighting factor and feedback correction term on solution accuracy is extensively tested out, and the performance of the various solutions is compared.

2. Kinematics

The ABB IRb 2000 is a six-revolute-joint robot manipulator manufactured by ABB Robotics. The manipulator is shown in Fig. 1. Its Craig-Denavit-Hartenberg parameters, joint working ranges and maximum speeds are reported in the tables below.

Link	ϑ [rad]	d [m]	a [m]	α [rad]
1	$\pi/2$	0	0	0
2	$\pi/2$	0	0	$\pi/2$
3	$\pi/2$	0	0.710	0
4	0	0.850	0.125	$\pi/2$
5	0	0	0	$\pi/2$
6	0	0.100	0	$\pi/2$

Joint	Working range [rad]	Max speed [rad/s]
1	$-0.99 \div +0.99$	2.01
2	$-0.85 \div +0.85$	2.01
3	$-2.72 \div -0.49$	2.01
4	$-3.43 \div +3.43$	4.89
5	$-2.00 \div +2.00$	5.24
6	$-3.14 \div +3.14$	5.24

The inner three joints are in the same arrangement as in an elbow manipulator, while the outer three joints constitute the spherical wrist commonly used in industrial robots.

Let q denote the (6×1) joint vector. The (6×6) Jacobian matrix J relates the joint velocity vector $\dot{q}$ to the (6×1) end-effector velocity vector v through

the mapping

$$\nu = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q)\dot{q} \quad (1)$$

where $\dot{p}$ and ω represent end-effector linear and angular velocities, respectively.

In configurations where J has full rank, any end-effector velocity can be attained. When J is rank deficient, i.e. $\text{rank}(J) < 6$, constraints on the feasible end-effector velocity occur and the manipulator is said to be at a singular configuration or at a *singularity*.

The ABB IRb 2000 manipulator has a simple kinematic structure and its singularities are well-understood. We have:

- If $a_3 \cos \vartheta_3 + d_4 \sin \vartheta_3 = 0$, the elbow is stretched out and the manipulator is in the so-called *elbow* singularity; this does *not* correspond to a reachable configuration of the manipulator, due to the mechanical joint range for ϑ_3 , and then is of no interest.
- If the wrist point lies on the axis of joint 1, its position cannot be changed by a rotation of ϑ_1 and the manipulator is in the so-called *shoulder* singularity; from simple geometry, it can be found that the shoulder singularity occurs when $a_2 \sin \vartheta_2 + a_3 \cos(\vartheta_2 + \vartheta_3) + d_4 \sin(\vartheta_2 + \vartheta_3) = 0$.
- If $\vartheta_5 = 0$ the two roll axes of the wrist are aligned and the manipulator is in the so-called *wrist* singularity.

3. Kinematic Control

It is well-known that the control system of a robotic manipulator operates in the joint space generating the driving torques at the joint actuators. The reference trajectory for the joint control servos is to be generated via kinematic inversion of the given end-effector trajectory; this is known as the *kinematic control* problem. When the arm is at—or close to—a singularity, large joint velocities may occur or degenerate directions may exist where end-effector velocity is not feasible. Therefore, the control system of a robotic manipulator should be provided with the capability of handling singularities.

A well-known effective strategy that allows motion control of manipulators in the neighborhood of kinematic singularities is the *damped least-squares* technique originally proposed in [2],[6]. The joint velocity solution can be formally written as

$$\dot{q} = [J^T(q)J(q) + \lambda^2 I]^{-1} J^T(q)\nu \quad (2)$$

where $\lambda \geq 0$ is a damping factor. Small values of λ give accurate solutions but low robustness to occurrence of singular and near-singular configurations. Large values of λ result in low tracking accuracy even when a feasible and accurate solution would be possible.

The damping factor λ determines the degree of approximation introduced with respect to the pure least-squares solution; then, using a constant value for λ may turn out to be inadequate for obtaining good performance over the entire manipulator workspace. An effective choice is to adjust λ as a function

of some measure of closeness to the singularity at the current configuration of the manipulator.

A singular region can be defined on the basis of the estimate of the smallest singular value of $\mathbf{J}$; outside the region the exact solution is used, while inside the region a configuration-varying damping factor is introduced to obtain the desired approximate solution. The factor must be chosen so that continuity of joint velocity $\dot{\mathbf{q}}$ is ensured in the transition at the border of the singular region. We have selected the damping factor according to the following law [7]:

$$\lambda^2 = \begin{cases} 0 & \hat{\sigma}_6 \geq \varepsilon \\ \left(1 - \left(\frac{\hat{\sigma}_6}{\varepsilon}\right)^2\right) \lambda_{max}^2 & \hat{\sigma}_6 < \varepsilon \end{cases} \quad (3)$$

where $\hat{\sigma}_6$ is the estimate of the smallest singular value, and ε defines the size of the singular region; the value of λ_{max} is at user's disposal to suitably shape the solution in the neighborhood of a singularity.

We have computed an estimate of the smallest singular value by the recursive algorithm recently proposed in [4] which is an extension of the algorithm in [8]; this algorithm allows estimating not only the smallest but also the second smallest singular value and then is congenial to handle closeness to a double singularity, whenever crossing of smallest singular values occurs.

The above damped least-squares method achieves a compromise between accuracy and robustness of the solution. This is performed without specific regard to the components of the particular task assigned to the manipulator's end effector. The *user-defined accuracy* strategy introduced in [7] based on the weighted damped least-squares method in [2] allows discriminating between directions in the end-effector space where higher accuracy is desired and directions where lower accuracy can be tolerated.

Let a weighted end-effector velocity vector be defined as

$$\tilde{\mathbf{v}} = \mathbf{W}\mathbf{v} \quad (4)$$

where $\mathbf{W}$ is the (6×6) task-dependent weighting matrix taking into account the anisotropy of the task requirements. Substituting (4) into (1) gives

$$\tilde{\mathbf{v}} = \tilde{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \quad (5)$$

where $\tilde{\mathbf{J}} = \mathbf{W}\mathbf{J}$. It is worth noticing that if $\mathbf{W}$ is full-rank, solving (1) is equivalent to solving (5), but with different conditioning of the system of equations to solve. This suggests selecting only the strictly necessary weighting action in order to avoid undesired ill-conditioning of $\tilde{\mathbf{J}}$. The damped least-squares solution to (5) is

$$\dot{\mathbf{q}} = [\tilde{\mathbf{J}}^T(\mathbf{q})\tilde{\mathbf{J}}(\mathbf{q}) + \lambda^2 \mathbf{I}]^{-1} \tilde{\mathbf{J}}^T(\mathbf{q})\tilde{\mathbf{v}} \quad (6)$$

For the typical elbow geometry with spherical wrist, it is worthwhile to devise a special handling of the wrist singularity which is difficult to predict at the planning level in the end-effector space. It can be recognized that, at the wrist singularity, there are only two components of the angular velocity vector

that can be generated by the wrist itself. The remaining component might be generated by the inner joints, at the expense of loss of accuracy along some other end-effector space directions though. For this reason, lower weight should be put on the angular velocity component that is infeasible to the wrist. For the ABB IRb 2000, this is easily expressed in the frame attached to link 4; let R_4 denote the rotation matrix describing orientation of this frame with respect to the base frame, so that the infeasible component is aligned with the x -axis. We propose then to choose the weighting matrix as

$$W = \begin{bmatrix} I & O \\ O & R_4 \text{diag}\{w, 1, 1\} R_4^T \end{bmatrix} \quad (7)$$

Similarly to the choice of the damping factor as in (3), the weighting factor w is selected according to the following expression:

$$(1 - w)^2 = \begin{cases} 0 & \hat{\sigma}_6 \geq \varepsilon \\ \left(1 - \left(\frac{\hat{\sigma}_6}{\varepsilon}\right)^2\right) (1 - w_{min})^2 & \hat{\sigma}_6 < \varepsilon \end{cases} \quad (8)$$

where $w_{min} > 0$ is a design parameter [7],[3].

The above inverse kinematics solutions are expected to suffer from typical numerical drift, when implemented in discrete time. In order to avoid this drawback, a *feedback correction* term [5] can be keenly introduced by replacing the end-effector velocity ν by

$$\nu_d + K e \quad (9)$$

where the subscript “ d ” denotes the desired reference end-effector velocity, K is a positive definite—usually diagonal—(6×6) matrix, and e expresses the error between the desired and actual end-effector location. The error e is computed as

$$e = \begin{bmatrix} e_t \\ e_o \end{bmatrix} = \begin{bmatrix} p_d - p \\ \frac{1}{2}(\mathbf{n} \times \mathbf{n}_d + \mathbf{s} \times \mathbf{s}_d + \mathbf{a} \times \mathbf{a}_d) \end{bmatrix} \quad (10)$$

where the translation error is given by the (3×1) vector e_t and the orientation error is given by the (3×1) vector e_o . The end-effector position is expressed by the (3×1) position vector p while its orientation by the (3×3) rotation matrix $R = [\mathbf{n} \ \mathbf{s} \ \mathbf{a}]$, with $\mathbf{n}, \mathbf{s}, \mathbf{a}$ being the unit vectors of the end-effector frame.

It is important to notice that, in the neighborhood of a singularity, end-effector errors typically increase along the near-degenerate components of the given end-effector velocity and convergence is slowed down [9]. Therefore, we propose to shape the action of the feedback correction term around the singularities using $K = \varrho K_0$, where K_0 is a constant matrix and ϱ is a varying factor to be properly adjusted.

We have found that it is important to have $\varrho = 0$ inside the singular region defined by $\sigma_6 \leq \varepsilon$. Indeed, if a velocity is assigned along a near-degenerate direction and a nonzero gain is used for the feedback correction term, the error e will eventually approach zero; however, the resulting joint velocities

may cause the manipulator to reach the joint limits. Outside the singular region, interpolation is used to achieve a smooth solution and full value to the gains ($\rho = 1$) is set when far enough from the singularity. In our experience, interpolation had to be performed when $\varepsilon < \sigma_6 < 4\varepsilon$ using a quadratic type function, i.e.

$$\rho = \begin{cases} 0 & \sigma_6 \leq \varepsilon \\ \frac{(\sigma_6 - \varepsilon)^2}{(3\varepsilon)^2} & \varepsilon < \sigma_6 < 4\varepsilon \\ 1 & \sigma_6 \geq 4\varepsilon \end{cases} \quad (11)$$

4. Experimental results

The experiments were run on an ABB IRb 2000 robot manipulator. The original joint servos of the S3 industrial control system were used, and an interface was established at the joint increment level. This allowed implementation of a kinematic control strategy, that is an inverse kinematics module based on the foregoing damped least-squares solution providing the reference inputs to the manipulator joint servos. This was done in cooperation with ABB Robotics who slightly modified the S3 control system by adding an interface board with a dual-port RAM, as well as software modules to establish the required communication protocol. The resulting communication facilitated the transfer of joint variables and increments at a sampling time of 12 [ms], and in addition it allowed for remote initialization and activation of the ABB S3 system.

The inverse kinematics were computed on a 25 [MHz] Motorola 68040 VME board which communicated with the ABB S3 control system through the dual-port RAM. The software was developed on a SUN 350 workstation, and the executable code was downloaded to the 68040 board and executed using Vx-Works. A block diagram showing the interconnection between major components of the laboratory set-up is sketched in Fig. 2.

The program system which was executed on the Motorola 68040 was written in C, and consisted of two activities; an interactive activity containing initialization and user interface, and the real-time controller activity. The real-time controller activity had the following steps: input joint angles, calculate joint increments, output joint increments, store data for logging.

The interactive user interface contained initialization of the S3 control system and of the communication between the ABB controller and the real-time activity. Further it contained initialization of kinematic parameters and a menu-driven system for adjusting kinematic parameters, specifying trajectory data, and selecting the algorithm for the inverse kinematics solution. In addition, a function for transferring logged data to the SUN system was included. The logged data was subsequently written to a file in MATLAB format which allowed for postprocessing and plotting in the SUN UNIX environment.

A simple solution to the problem of communication between the real-time and interactive activities was achieved by including both activities in the same C program. The interactive activity was the main program, while the real-time activity was called from a function which was started by a timer interrupt every 12 [ms]. The data transfer between the two activities was performed

through global data structures containing parameters for the inverse kinematics algorithms, trajectory data, logged data and logical variables for algorithm selection.

In the following we present two case studies to demonstrate the application of the preceding schemes to real-time kinematic control of the ABB IRb 2000 robot manipulator.

Trajectory 1: A reference trajectory through the wrist singularity was studied. The initial configuration was $\mathbf{q} = [0 \ \pi/12 \ -\pi/2 \ 0 \ 0.15 \ 0]^T$ [rad]. An end-effector increment $\Delta\mathbf{p} = [0.18 \ 0.45 \ -0.45]^T$ [m] was interpolated using linear segments with parabolic blends. The blend time was 0.2 [s], and the total time of the trajectory was 1.5 [s]. The resulting cruise velocity between 0.2 [s] and 1.3 [s] was approximately 0.5 [m/s]. The wrist singularity was encountered after approximately 0.6 [s].

The damping factor was computed from (3) with $\varepsilon = 0.04$ and $\lambda_{max} = 0.04$. The estimate $\hat{\sigma}_6$ of the smallest singular value was computed using the algorithm in [4]

The basic damped least-squares scheme (2) was used first; then, for comparison, the same trajectory was tested with the weighted damped least-squares scheme based on (6). The weighting matrix was chosen as in (7),(8) with $w_{min} = 0.1$. The results are shown in Fig. 3 for both the damped least-squares and weighted damped least-squares schemes, without feedback correction term. In both cases the joint velocities were feasible with peak values of approximately 2 [rad/s]. Without weighting, the norm of the translation error at final time t_f was $\|\mathbf{e}_t(t_f)\| = 0.055$ [m], while the orientation error norm was $\|\mathbf{e}_o(t_f)\| = 0.06$ [rad]. With weighting, the errors were $\|\mathbf{e}_t(t_f)\| = 0.0025$ [m] and $\|\mathbf{e}_o(t_f)\| = 0.12$ [rad].

This result clearly demonstrates the advantage of using weighting since our main concern was to achieve accuracy in translation. In fact, weighting resulted in a reduction of the translation error by a factor of approximately 20, while the orientation error was increased only by a factor of two. The effect of weighting on the smallest singular values is seen from Fig. 3; a significant reduction of $\hat{\sigma}_6$ around the singularity is obtained compared to the corresponding σ_6 in the non-weighted case.

The same experiments were repeated with a feedback correction term according to (9),(10) with $\mathbf{K} = \text{diag}\{12 \dots 12\}$. The results are shown in Fig. 4. In this case the joint velocities were higher. In fact, with the damped least-squares solution, joint velocities 4 and 6 saturated between 0.6 and 0.8 [s] at 5 [rad/s] and -5 [rad/s], respectively. With the introduction of weighting, the joint velocities were feasible with peak values less than 5 [rad/s]. Thanks to the feedback correction term, the end-effector error $\mathbf{e}$ converged to zero after leaving the singular region (see Fig. 4). This resulted in a reorientation of joints 4 and 6 by $\pm\pi$ in the non-weighted case, which reflects the fact that large increments in joint angles may result from small end-effector increments when the manipulator is close to a singularity. Remarkably, Figure 3 reveals also that outside the singular region $\hat{\sigma}_6$ tends to σ_6 ; obviously, this was not the case in the previous experiments without feedback correction (see Fig. 3).

Trajectory 2: A reference trajectory involving both the shoulder singularity and the wrist singularity was studied. The initial configuration was $q = [0 \ 0.7893 \ -\pi/2 \ \pi/2 \ -0.05 \ 0]^T$ [rad]. An end-effector increment $\Delta p = [0.1 \ 0.1 \ 0]^T$ [m] was interpolated using linear segments with parabolic blends. The blend time was 0.15 [s], and the total time of the trajectory was 1.0 [s]. The resulting cruise velocity between 0.15 [s] and 0.85 [s] was approximately 0.166 [m/s]. The wrist singularity was encountered after approximately 0.22 [s].

Also for this trajectory, the damping factor was computed from (3) with $\varepsilon = 0.04$ and $\lambda_{max} = 0.04$. The basic damped least-squares solution (2) was used. The estimate $\hat{\sigma}_6$ of the smallest singular value was found: a) by computing both $\hat{\sigma}_5$ and $\hat{\sigma}_6$ as in [4], and b) by computing $\hat{\sigma}_6$ as in [8].

The results without feedback correction are shown in Figs. 5,6. The damped least-squares scheme performs well also in this case. The norm of the translation error at final time t_f was $\|e_t(t_f)\| = 0.03$ [m], while the norm of the orientation error was $\|e_o(t_f)\| = 0.015$ [rad].

In case a) the crossing of the two smallest singular values associated with the wrist and shoulder singularity was successfully detected at 0.15 [s] and 0.37 [s], and an accurate estimate $\hat{\sigma}_6$ of the smallest singular value was found. This gave satisfactory damping around the wrist singularity. The resulting joint velocities had peak values less than 1.2 [rad/s]. In case b) the crossing of the two smallest singular values caused the estimate $\hat{\sigma}_6$ to track σ_5 , and the wrist singularity appearing at 0.22 [s] was not detected. This resulted into a low damping factor around the wrist singularity, and high joint velocities were experienced; in particular, the velocity of joint 1 saturated at -2 [rad/s]. Finally, note that the final errors were a little smaller in case b) since the incorrect estimate of σ_6 produced lower damping on the solution throughout the singular region.

5. Conclusions

The weighted damped least-squares solution with varying damping factor, varying weighting factor and feedback correction term has been implemented on an industrially available hardware indicating that enhanced real-time kinematic control of robot manipulators through singularities is definitely possible.

Acknowledgments

The research work described in this paper was supported partly by *Consiglio Nazionale delle Ricerche* under contract 93.00905.PF67, and partly by *ABB Robotics*.

References

- [1] S. Chiaverini, B. Siciliano, and O. Egeland, "Robot control in singular configurations — Analysis and experimental results," *2nd International Symposium on Experimental Robotics*, Toulouse, F, June 1991, in *Experimental Robotics II*, Lecture Notes in Control and Information Sciences 190, R. Chatila and G. Hirzinger (Eds.), Springer-Verlag, Berlin, D, pp. 25–34, 1993.

- [2] Y. Nakamura and H. Hanafusa, "Inverse kinematic solution with singularity robustness for robot manipulator control," *ASME J. of Dynamic Systems, Measurements, and Control*, vol. 108, pp. 163-171, 1986.
- [3] S. Chiaverini, O. Egeland, R.K. Kanestrøm, "Weighted damped least-squares in kinematic control of robotic manipulators," *Advanced Robotics*, vol. 7, pp. 201-218, 1993.
- [4] S. Chiaverini, "Estimate of the two smallest singular values of the Jacobian matrix: Application to damped least-squares inverse kinematics," *J. of Robotic Systems*, vol. 10, no. 8, 1993.
- [5] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 16, pp. 550-559, 1986.
- [6] C.W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 16, pp. 93-101, 1986.
- [7] S. Chiaverini, O. Egeland, and R.K. Kanestrøm, "Achieving user-defined accuracy with damped least-squares inverse kinematics," *Proc. 5th Int. Conf. on Advanced Robotics*, Pisa, I, pp. 672-677, June 1991.
- [8] A.A. Maciejewski and C.A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *J. of Robotic Systems*, vol. 5, pp. 527-552, 1988.
- [9] C.W. Wampler and L.J. Leifer, "Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators," *J. of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 31-38, 1988.

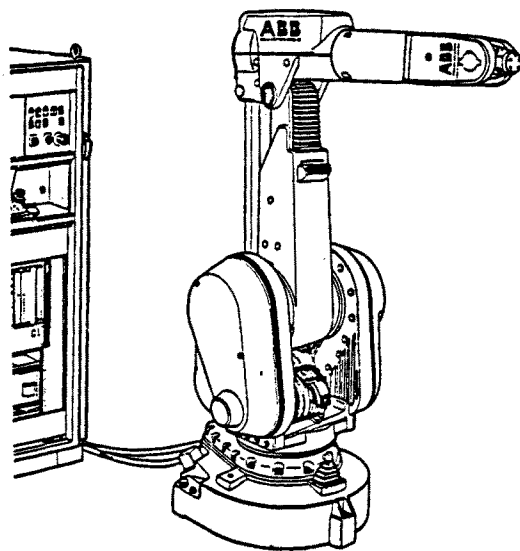


Fig. 1 — The ABB IRb 2000 industrial robot manipulator.

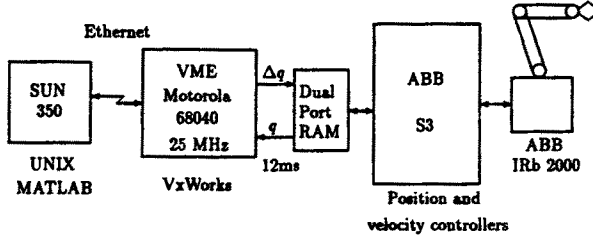


Fig. 2 — Block diagram of laboratory set-up.

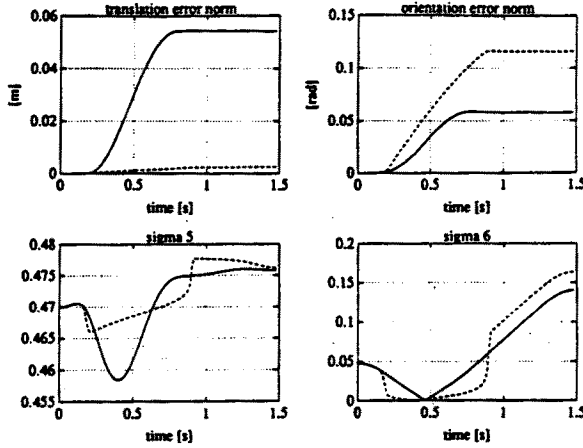


Fig. 3 — Translation and orientation error norms and estimates of the two smallest singular values for Trajectory 1 without feedback correction; damped least-squares scheme (solid), weighted damped least-squares scheme (dashed).

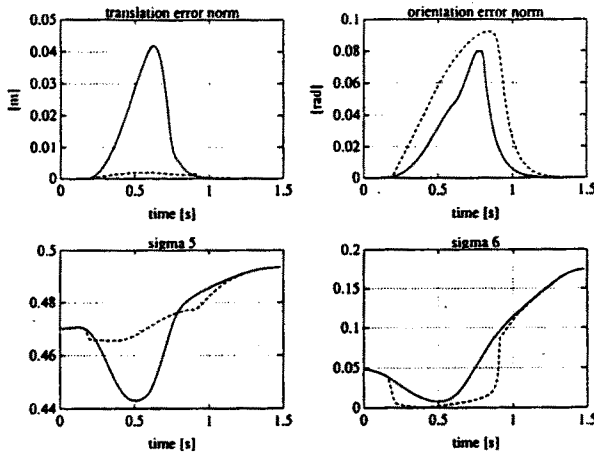


Fig. 4 — Translation and orientation error norms and estimates of the two smallest singular values for Trajectory 1 with feedback correction; damped least-squares scheme (solid), weighted damped least-squares scheme (dashed).

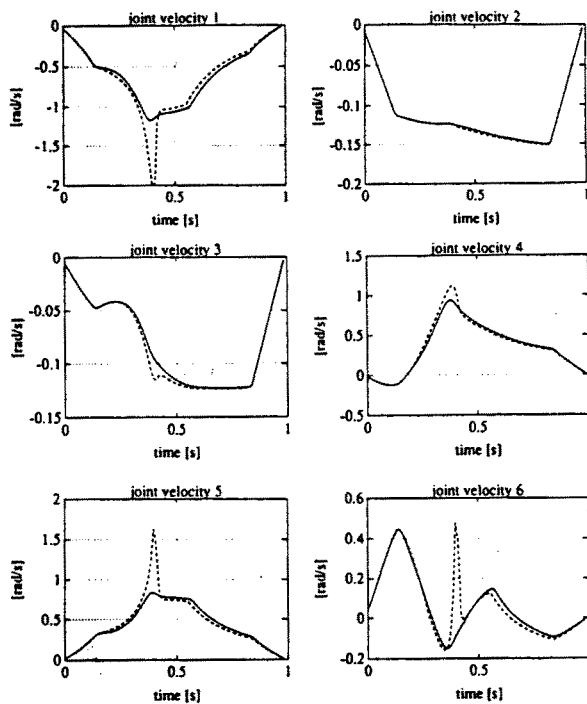


Fig. 5 — Joint velocities for Trajectory 2; case a) (solid), case b) (dashed).

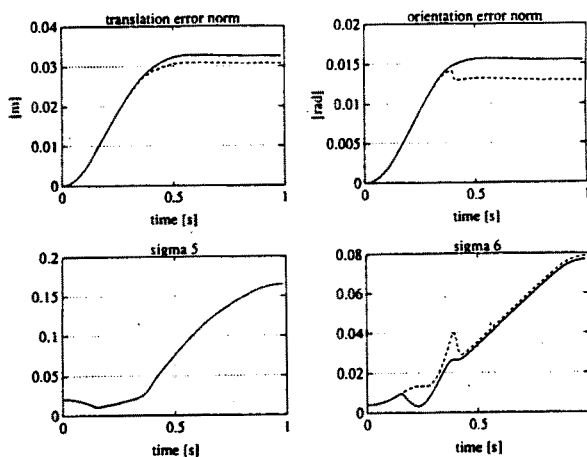


Fig. 6 — Translation and orientation error norms and estimates of the two smallest singular values for Trajectory 2; case a) (solid), case b) (dashed).