Remote Calibration and Trajectory Replanning for Robot Manipulators Operating in Unstructured Environments

Antonio C. Leite, Fernando Lizarralde * Pål Johan From **
Ramon R. Costa, Liu Hsu *

* *Department of Electrical Engineering - COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil. (e-mail: [toni, fernando, ramon, liu]@coep.ufrj.br).*
** *Department of Mathematical Sciences and Technology, Norwegian University of Life Sciences, Ås, Norway. (e-mail: pafr@umb.no).*

Abstract: In this work the problem of remote calibration and trajectory replanning for a subsea robotic manipulator is considered. Because the trajectory planning is normally done in a structured environment, several uncertainties arise when the robot is placed on the seabed and these need to be compensated for to guarantee that the task specifications are fulfilled. We address the particular problem of configuration errors in the robot base with respect to the configuration used during the off-line trajectory planning. A calibration method based on both internal and external sensors is presented in order to estimate the uncertainty in the location of the robot. Moreover, a trajectory replanning strategy in the Cartesian velocity space is proposed to guarantee that the originally planned trajectory is followed. Simulation and experiments performed with a 6-DoF robot manipulator and a real calibration grid show the viability of the proposed planning and control schemes.

Keywords: Robot calibration, Underwater robotics, Trajectory replanning, Robot kinematics.

1. INTRODUCTION

Calibration and task planning of robotic systems in remote and unstructured environments is of vital importance for accurate and robust operation of subsea fields (Augustson and Meggiolaro, 2010). In such scenarios, the planning of a reference trajectory is usually performed on the surface or in a structured place on-shore, or even in a simulation environment (Trevelyan et al., 2008). However, once the robotic cell is inserted into its workplace, uncertainties will arise, for example in the robot's actual location with respect to the location used during planning. The difference between the actual robot base location and the location adopted during the planning characterizes the trajectory replanning problem discussed in this paper.

Several calibration schemes for kinematic and dynamic calibration are described in literature (Mooring et al., 1991; Beyer and Wulfsberg, 2004), but to the authors best knowledge, the specific problem of robot base configuration errors has not previously been addressed in this setting. Due to the nature of the problem the approach presented can be used not only for calibration of fixed-base underwater manipulators, but also to estimate the vehicle pose in underwater vehicle-manipulator systems.

Robot calibration refers to a set of procedures for determining the real values of the geometric dimensions and mechanical characteristics of the robot structure. A taxonomy for robot kinematics calibration methods based on a calibration index which represents the number of independent equations available for calibration was proposed in Hollerbach and Wampler (1996). A well known methodology used to calibrate the robot with respect to a known geometric structure is to endow the robot end-effector with a probe. Upon touching the structure surface in appropriate points it is possible to estimate the transformation which relates the robot coordinates with the structure coordinates of interest (Ikits and Hollerbach, 1997). Also, open-loop and closed-loop calibration methods have been proposed using different sensors such as cameras, laser beams, and triaxial accelerometers (Canepa et al., 1994; Motta et al., 2001; Lei et al., 2004). However, for all the approaches above, it is assumed and required that the location of the robot base is known in advance. This is not always the case, for example when the robot is operating in an unstructured environment.

In this work, a calibration method for estimating the uncertainties in the robot base configuration using a calibration grid is proposed. We present solutions based on both internal and external sensors to solve the calibration problem. A calibration and trajectory replanning strategy in the Cartesian velocity space is presented to solve the problem of trajectory planning in the presence of uncertainties. Empirical studies obtained with a 6-DoF robot manipulator show the viability of the proposed scheme.

⋆ This work was partially supported by the Brazilian Founding Agencies CNPq, FAPERJ, and the Norwegian Research Council.

2. PROBLEM FORMULATION

Consider the tracking problem of a reference trajectory $r(t)$ using a robot manipulator mounted on a fixed base. Fig. 1a illustrates the case where the robot base frame $\bar{E}_b$ coincides with the world frame $\bar{E}_w$ and a trajectory planner generates the appropriate reference inputs to the motion control system. Now, consider a situation in Fig. 1b in which the robot base is dislocated with respect to the original location used during the planning phase. Denote by $\bar{E}_b$ the planned base location and by $\bar{E}_{b_1}$ the actual base position. Then, in order to ensure that the manipulator follows the reference trajectory $r(t)$ specified in the world frame $\bar{E}_w$, we have to execute a replanning of this trajectory considering the misplacement of the robot base. The trajectory replanning is based on the ideal trajectory $r(t)$ as well as the estimation of the configuration error between unknown actual base location and the location used during the planning phase.

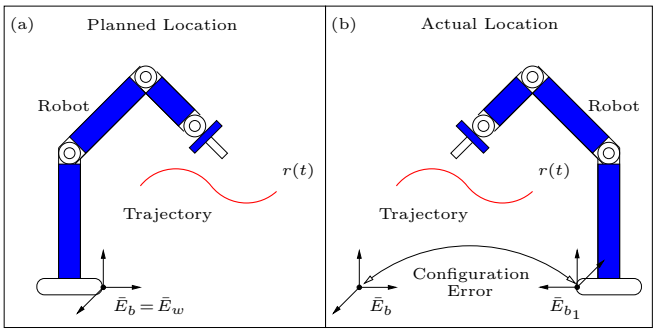

Fig. 1. Trajectory replanning problem.

Remark 1. In this work, we assume that the reference trajectory $r(t)$ is *known* and previously calculated using a trajectory planning algorithm defined either in the joint space or in the operational space.

3. CALIBRATION OF ROBOTIC SYSTEMS

In this section, we discuss the problem of remote calibration of a robotic system. Two calibration methods based on measurements obtained from internal and external sensors are presented to estimate the configuration error between the actual and planned base locations. During the planning phase the base location is known and assumed to coincide with the world frame. We denote this location the *planned base location* and associate it with frame $\bar{E}_b$. When the robot is inserted into the remote workspace it will be placed at a location that in general differs from the planned base location. We will call this location the *actual base location*, denoted $\bar{E}_{b_1}$. Using a calibration grid and a set of endpoint positions on this grid, we estimate the actual base configuration by (i) measuring the joint positions using encoders and (ii) using a fixed camera. The configuration error is determined by an appropriate optimization technique based on the least squares method. It is worth mentioning that in both methods, the endpoints' positions can be obtained either by using guided teleoperation or autonomously by using a combination of force control and a visual servoing approach. In this paper, we assume that the calibration grid is available, but we can just as well use known reference points on the seabed, for example on a subsea installation.

In this context, consider the configuration error between the actual location $\bar{E}_{b_1}$ and the planned location $\bar{E}_b$ represented by the homogeneous transformation matrix as

$$T_{bb_1} = \begin{bmatrix} R_{bb_1} & p_{bb_1} \\ 0^{\mathsf{T}} & 1 \end{bmatrix}, \tag{1}$$

or in terms of vectorial notation

$$\mathbf{x}_{bb_1} = [\, p_{bb_1}^{\mathsf{T}} \;\; \Phi_{bb_1}^{\mathsf{T}} \,]^{\mathsf{T}}. \tag{2}$$

Here $p_{bb_1} = [\, x_{bb_1} \;\; y_{bb_1} \;\; z_{bb_1} \,]^{\mathsf{T}}$ denotes the relative position of the origin of frame $\bar{E}_{b_1}$ expressed in $\bar{E}_b$, $R_{bb_1} \in SO(3)$, and $\Phi_{bb_1} = [\, \phi \;\; \vartheta \;\; \psi \,]^{\mathsf{T}}$ denotes the orientation of $\bar{E}_{b_1}$ with respect to $\bar{E}_b$ in the ZYX Euler angles parameterization, also called Roll-Pitch-Yaw angles (Siciliano et al., 2008).

3.1 Error estimation with internal sensors

The estimation of the configuration error using internal sensors can be achieved from angular sensors (e.g., encoder or resolvers) located in the manipulator joints. To this end, we use a set of endpoints $\lambda_1, \lambda_2, \ldots, \lambda_N$ on the calibration grid where the position of each calibration point λ_i is known with respect to $\bar{E}_b$ and estimated with respect to $\bar{E}_{b_1}$. The configuration error is calculated by solving the following system of equations for p_{bb_1} and R_{bb_1}:

$$\begin{aligned} p_{b\lambda_1} &= p_{bb_1} + R_{bb_1}\, p_{b_1\lambda_1}, \\ &\vdots \\ p_{b\lambda_N} &= p_{bb_1} + R_{bb_1}\, p_{b_1\lambda_N}. \end{aligned}$$

Here, $p_{b\lambda_i} \in \mathbb{R}^3$ and $p_{b_1\lambda_i} \in \mathbb{R}^3$ for $i = 1, 2, \ldots, N$ denote the position of the calibration points with respect to the frames $\bar{E}_b$ and $\bar{E}_{b_1}$ respectively, obtained by the forward kinematics map.

Note that the homogeneous transformation T_{bb_1} can be expressed in terms of the configuration of the calibration points, that is,

$$T_{bb_1} = T_{b\lambda_i}\, T_{b_1\lambda_i}^{-1}, \quad i = 1, 2, \cdots N. \tag{3}$$

Then, to determine the configuration error $\mathbf{x}_{bb_1}$ using an estimation algorithm based on least squares method, such as Newton's method, we have to define an appropriate objective function f which describes the system in terms of its position and orientation parameters, in our case

$$f(p_{bb_1}, R_{bb_1}) = \begin{bmatrix} p_{b\lambda_1} - p_{bb_1} - R_{bb_1} p_{b_1\lambda_1} \\ \vdots \\ p_{b\lambda_N} - p_{bb_1} - R_{bb_1} p_{b_1\lambda_N} \end{bmatrix} \in \mathbb{R}^{3N}. \tag{4}$$

Note that each equation has three position coordinates and six constraints for orientation, and thus the minimum number of necessary endpoints to perform the calibration procedure is three. However, since the measurements are contaminated with noise we measure several endpoints to improve the accuracy of the configuration error parameters.

To minimize the objective function (4) we can use the Jacobian of the function f, denoted by J_f, given by

$$J_f = \nabla f = \begin{bmatrix} \dfrac{\partial^{\mathsf{T}} f}{\partial p_{bb_1}} & \dfrac{\partial^{\mathsf{T}} f}{\partial \Phi_{bb_1}} \end{bmatrix}^{\mathsf{T}}. \tag{5}$$

Note that J_f has a particular form which simplifies the calculations since $\frac{\partial f}{\partial p_{bb_1}} = -[\, I \cdots I \,]^{\mathsf{T}} \in \mathbb{R}^{3N \times 3}$. Then, to obtain (5) it is necessary only to calculate

$$\frac{\partial f}{\partial \Phi_{bb_1}} = -\begin{bmatrix} \frac{\partial R_{bb_1}}{\partial \phi} p_{b_1\lambda_1} & \frac{\partial R_{bb_1}}{\partial \vartheta} p_{b_1\lambda_1} & \frac{\partial R_{bb_1}}{\partial \psi} p_{b_1\lambda_1} \\ \vdots & \vdots & \vdots \\ \frac{\partial R_{bb_1}}{\partial \phi} p_{b_1\lambda_N} & \frac{\partial R_{bb_1}}{\partial \vartheta} p_{b_1\lambda_N} & \frac{\partial R_{bb_1}}{\partial \psi} p_{b_1\lambda_N} \end{bmatrix},$$

where $\frac{\partial R_{bb_1}}{\partial \phi}, \frac{\partial R_{bb_1}}{\partial \vartheta}, \frac{\partial R_{bb_1}}{\partial \psi} \in \mathbb{R}^{3\times 3}$ and $\frac{\partial f}{\partial \Phi_{bb_1}} \in \mathbb{R}^{3N\times 3}$.

Newton's method finds the correction in the estimated configuration error $\mathbf{x}_{bb_1}$ at each iteration k by

$$\mathbf{x}_{bb_1}(k+1) = \mathbf{x}_{bb_1}(k) - \beta\, J_f^{\dagger}(k)\, f(p_{bb_1}(k), R_{bb_1}(k)), \quad (6)$$

where $\beta > 0$ and $J_f^{\dagger}(k) = (\, J_f^{\mathsf{T}}(k)\, J_f(k)\,)^{-1}\, J_f(k)^{\mathsf{T}}$ denotes the left pseudo-inverse of J_f.

Note that the dimension of the Jacobian matrix J_f is $3N\times 6$, so it may grow very large if several endpoints are chosen. Thus, in order to avoid bad numerical conditioning, matrix inversion strategies such as the Levenberg-Marquardt algorithm (Motta et al., 2001) should be adopted. Furthermore, it cannot be guaranteed that this approach converges to a global minimum.

3.2 Error estimation with external sensors

In this section, we propose a method where the configuration error is estimated using a fixed and calibrated CCD camera mounted in a known position relative to the robot, for example on the robot base (see Fig. 2). In addition to calibration purposes, cameras are also useful sensors since they allow the robots to inspect and locate objects without contact, and to share the workspace with human operators or even other robots (Hutchinson et al., 1996). In this work, the following assumptions are made: (A1) the z-axis of the camera and robot base frames are aligned; (A2) the optical axis of the camera is perpendicular to the task plane. We note that the camera configuration can easily be chosen so that conditions (A1) and (A2) hold.

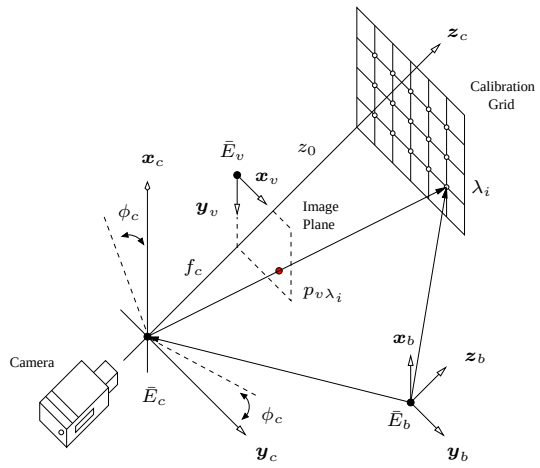

Fig. 2. Camera and robot base frames.

Let p_{bc}, $p_{b\lambda_i} \in \mathbb{R}^3$ be the relative positions of the origins of the camera frame $\bar{E}_c$ and the calibration points λ_i, respectively, and the robot base frame $\bar{E}_b$. $R_{bc} \in SO(3)$ is the rotation matrix of the camera frame with respect to the base frame. Then, the position of the calibration point λ_i with respect to the camera frame $\bar{E}_c$ is given by

$$p_{c\lambda_i} = R_{bc}^{\mathsf{T}}\,(\, p_{b\lambda_i} - p_{bc}\,), \quad (7)$$

or in terms of the its Cartesian coordinates

$$\begin{bmatrix} x_{c\lambda_i} \\ y_{c\lambda_i} \\ z_{c\lambda_i} \end{bmatrix} = R_{bc}^{\mathsf{T}} \left(\begin{bmatrix} x_{b\lambda_i} \\ y_{b\lambda_i} \\ z_{b\lambda_i} \end{bmatrix} - \begin{bmatrix} x_{bc} \\ y_{bc} \\ z_{bc} \end{bmatrix} \right). \quad (8)$$

From the perspective projection model of a pin-hole camera (Hutchinson et al., 1996), the calibration point λ_i is projected to the 2-dimensional image plane with coordinates $p_{v\lambda_i} = [\, x_{v\lambda_i} \;\; y_{v\lambda_i}\,]^{\mathsf{T}}$ given by

$$\begin{bmatrix} x_{v\lambda_i} \\ y_{v\lambda_i} \end{bmatrix} = \frac{f_c}{z_{c\lambda_i}} \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{c\lambda_i} \\ y_{c\lambda_i} \end{bmatrix}, \quad (9)$$

where f_c is the camera focal length (mm) and α_1, α_2 are the camera scaling factors ($pixel\ mm^{-1}$). Based on assumptions (A1) and (A2), and substituting (8) into (9) yields

$$p_{v\lambda_i} = K_{v_i}(\bar{p}_{b\lambda_i} - \bar{p}_{bc}). \quad (10)$$

Here $\bar{p}_{b\lambda_i} = [x_{b\lambda_i} \;\; y_{b\lambda_i}]^{\mathsf{T}}$ and $\bar{p}_{bc} = [x_{bc} \;\; y_{bc}]^{\mathsf{T}}$ are the 2-dimensional projection of $p_{b\lambda_i}$ and p_{bc}, respectively, and

$$K_{v_i} = \frac{f_c}{z_{c\lambda_i}} \begin{bmatrix} 0 & -\alpha_1 \\ \alpha_2 & 0 \end{bmatrix} \begin{bmatrix} \cos(\phi_c) & -\sin(\phi_c) \\ \sin(\phi_c) & \cos(\phi_c) \end{bmatrix}, \quad (11)$$

$$z_{c\lambda_i} = z_{b\lambda_i} + z_{bc} = f_c + z_0, \quad (12)$$

where z_0 is the relative depth (constant) of the calibration points with respect to the image frame $\bar{E}_v$, and ϕ_c is the misalignment angle between the camera and the robot frame (see Fig. 2).

In the general case, the coordinate transformation (10) is non-homogeneous. However, since we consider that the camera is fixed to the base frame, we can assume that the origins of the camera frame and the base frame coincide and $\bar{p}_{bc} = 0$. Then, (10) can be rewritten as

$$p_{v\lambda_i} = K_{v_i}\, \bar{p}_{b\lambda_i}, \quad (13)$$

which are all known quantities. Analogously, performing the same procedure for the current location $\bar{E}_{b_1}$ we obtain

$$p'_{v\lambda_i} = K'_{v_i}\, \bar{p}_{b_1\lambda_i}, \quad (14)$$

which is unknown. Note that from (13) and (14) it is possible to obtain only a 2-dimensional representation of the position of the calibration point λ_i with respect to the frame $\bar{E}_b$ and $\bar{E}_{b_1}$, respectively. Then, to represent the position of the calibration point λ_i using a tridimensional vector we have to augment $\bar{p}_{b\lambda_i}$ with the coordinate $z_{b\lambda_i}$ and, similarly, augment $\bar{p}_{b_1\lambda_i}$ with the coordinate $z_{b_1\lambda_i}$. Thus, the position of the calibration point λ_i with respect to the frames $\bar{E}_b$ and $\bar{E}_{b_1}$ is given by

$$p_{b\lambda_i} = \begin{bmatrix}\, \bar{p}_{b\lambda_i}^{\mathsf{T}} & z_{b\lambda_i}\, \end{bmatrix}^{\mathsf{T}}, \qquad p_{b_1\lambda_i} = \begin{bmatrix}\, \bar{p}_{b_1\lambda_i}^{\mathsf{T}} & z_{b_1\lambda_i}\, \end{bmatrix}^{\mathsf{T}}. \quad (15)$$

Therefore, using (15), and choosing an initial solution $\mathbf{x}_0$ and a finishing criterion, we can use the algorithm proposed in the previous section to estimate the parameters of the homogeneous transformation T_{bb_1}.

It now only remains to find the depth coordinates $z_{b\lambda_i}$ and $z_{b_1\lambda_i}$. One way to obtain these is to use encoder measurements in combination with the forward kinematics map. However, to keep the approach based on external sensors separated from the one with internal sensors, we will adopt a framework to measure the projected area of a target object fixed on the end-effector tip by using a camera. The change in the projected area $A_v \in \mathbb{R}^+$, expressed in the image frame $\bar{E}_v$, is described by (Flandin et al., 2000)

$$\dot{A}_v = -\left(\frac{2\, A_v}{z_c}\right) \dot{z}_c\,. \quad (16)$$

Let $A_{v_0} \in \mathbb{R}^+$ denote the known projected area of the target object at time t_0, such that $A_{v_0} = A_v(t_0)$ corresponds to a distance $z_{c_0} = z_c(t_0)$. Here z_c is the distance between the camera and the area to be measured. Eliminating the time-dependence, we can integrate both sides of (16) over the intervals $[A_{v_0}\ A_v]$ and $[z_{c_0}\ z_c]$ to obtain the following relation:

$$z_c = z_{c_0} \left(\frac{A_{v_0}}{A_v} \right)^{\frac{1}{2}} , \tag{17}$$

where A_v is continuously captured by the camera and the coordinates $z_{b\lambda_i}$ and $z_{b_1\lambda_i}$ are obtained from (12), where $z_{c\lambda_i} = z_c$ at the calibration point.

We have now presented two approaches for estimating the configuration error of the robot base. The two approaches can be used independently or we can apply multi-sensor data fusion in order to obtain an improved configuration error estimate. The Kalman Filter approach allows for state estimation using the statistical characteristics of the process, minimizing the error produced by the noise presented in the process and can be used to estimate the configuration error between the planned and actual robot base locations (Sorenson, 1985; Gao and Harris, 2002).

4. TRAJECTORY REPLANNING

We will first illustrate the general idea of trajectory replanning through a simple example considering the end-effector position only. In this section we assume that T_{bb_1} is either known or estimated using one of the calibration methods presented in the previous section. The purpose of the replanning is to define a new desired trajectory—specified either in the joint space or in the operational space—using the configuration error obtained in the calibration phase. Then, considering the robot positioned in the original location $\bar{E}_b$ and a goal frame $\bar{E}_d$ suitably attached in the reference trajectory, the ideal desired trajectory is given by the forward kinematic map as

$$p_{bd} = k(\theta_{bd}) , \tag{18}$$

where $\theta_{bd} \in \mathbb{R}^n$ is the desired trajectory in the joint space. From the differential kinematics equation, the velocity of the ideal trajectory can be written as

$$\dot{p}_{bd} = J_p(\theta_{bd})\, \dot{\theta}_{bd} . \tag{19}$$

where J_p is the Jacobian of the end-effector position. Assume that the actual base location $\bar{E}_{b_1}$ is different from the location $\bar{E}_b$ used during the planning phase. Then, the new desired trajectory is given by

$$p_{b_1 d} = R_{bb_1}^{\mathsf{T}} \left(p_{bd} - p_{bb_1} \right) . \tag{20}$$

From (19) and differentiating (20) with respect to time, we obtain the velocity of the new desired trajectory as

$$\dot{p}_{b_1 d} = R_{bb_1}^{\mathsf{T}}\, J_p(\theta_{bd})\, \dot{\theta}_{bd} . \tag{21}$$

The joint velocity for the new desired trajectory is then given by an open-loop scheme as

$$\dot{\theta}_{b_1 d} = J_p^{-1}(\theta_{b_1 d})\, R_{bb_1}^{\mathsf{T}}\, J_p(\theta_{bd})\, \dot{\theta}_{bd} . \tag{22}$$

We can now design a closed-loop kinematic control scheme by using

$$\dot{\theta}_{b_1 d} = J_p^{-1}(\theta_{b_1 d}) \left[\dot{p}_{b_1 d} + K_p \left(p_{b_1 d} - p_{b_1 e} \right) \right] , \tag{23}$$

where K_p is the position gain matrix, $\dot{p}_{b_1 d}$ is taken from (21) and $p_{b_1 e}$ is the (measured) end-effector position, expressed in the current location $\bar{E}_{b_1}$. This closed-loop scheme guarantees the tracking of a trajectory planned in the original location $\bar{E}_b$, also when the robot is positioned in the current location $\bar{E}_{b_1}$.

5. CALIBRATION AND REPLANNING IN THE VELOCITY SPACE

In this section, we consider the problem of calibration and trajectory replanning for a robot manipulator in the Cartesian velocity space. The presented solution consists of determining the uncertainty in the absolute location of the robotic cell from knowledge of the linear and angular robot velocities.

5.1 Path planning

A different approach to solve the problem of trajectory planning is to use the differential kinematics of the robot manipulator (Seereeram and Wen, 1995):

$$\dot{\mathbf{x}}(t) = J_k(\theta(t))\, \dot{\theta}(t) , \tag{24}$$

where $J_k(\theta(t)) : \mathbb{R}^n \mapsto \mathbb{R}^m$ is the Jacobian matrix. In this context, the problem is to obtain solutions to (24) subject to equality constraints (e.g., desired trajectory) and inequality constraints (e.g., joint space limit, singularities in the task space and presence of obstacles). Consider the problem of finding a continuous path in the joint space which satisfies a reference trajectory specified in the task space. Then, under assumption of kinematic control ($u = \dot{\theta}$), the problem can be formulated as follows:

For the nonlinear system described by $\dot{\mathbf{x}} = J_k(\theta)\, u$, *with initial and final configurations,* $\mathbf{x}_0$ *and* $\mathbf{x}_d$, *and finite time horizon* $T > 0$, *the goal is to determine* $\bar{u} = \{u(t), t \in [0\ \ T)\}$ *such that the solution of (24) satisfies* $\mathbf{x}(T) = \mathbf{x}_d$.

Here, $\mathbf{x} \in \mathbb{R}^{m \cdot n_d}$ denotes a stacked vector representation for a subset of space task coordinates over the entire path, $\mathbf{x}_d$ is the corresponding desired task vector and n_d is the number of points at which the task specification must be fulfilled. Note that (24) can be rewritten as a nonlinear algebraic equation given by $\mathbf{x} = F(\mathbf{x}_0, \bar{u}(\tau))$. Then, by writing the final state error between the nominal and desired paths as

$$\epsilon = F(\mathbf{x}_0, \bar{u}(\tau)) - \mathbf{x}_d = 0 , \tag{25}$$

the path planning problem can be considered a non-linear least squares problem (Lizarralde and Wen, 1996). The solution can be obtained from the differentiation of (25) with respect to the iteration variable τ, that is,

$$\frac{d\epsilon}{d\tau} = \nabla_{\bar{u}}\, F(\mathbf{x}_0, \bar{u}(\tau))\, \frac{d\bar{u}}{d\tau} , \tag{26}$$

where the mapping $\nabla_{\bar{u}}\, F$ is the Frechet's derivative of $F(\mathbf{x}_0, \cdot)$ with respect to $\bar{u}$. Assuming that $\nabla_{\bar{u}}\, F$ is surjective, a proper choice of update law $\bar{u}$ is given by (Lizarralde and Wen, 1996)

$$\frac{d\bar{u}}{d\tau} = -\beta\, \left[\nabla_{\bar{u}}\, F(\mathbf{x}_0, \bar{u}(\tau)) \right]^{\dagger}\, \epsilon(\tau) , \tag{27}$$

where $\beta > 0$ and $[\,\cdot\,]^{\dagger}$ denotes the Moore-Penrose pseudoinverse. Note that the differential equation (27) is continuous version of the Newton's method and it defines an initial condition problem in $\bar{u}$ for a given $\bar{u}(0)$. Then, the solution

$\bar{u}(\tau)$ can be obtained by solving (27) through a numeric integration method.

A sufficiency condition for algorithm convergence is that $\nabla_{\bar{u}} F$ is surjective or full rank for all τ (Lizarralde and Wen, 1996). Hence, under this condition and substituting (27) into (26) we have that $\frac{d\epsilon}{d\tau} = -\beta\,\epsilon$, which implies that the norm of the error $\|\epsilon\|$ decreases monotonically in τ, such that, $\|\epsilon(\tau)\| \leq \|\epsilon(0)\|\, e^{-\beta\tau}$. The derivative $\nabla_{\bar{u}} F$ can be obtained from the system (24) linearized around the trajectory $(\mathbf{x}, u)$

$$\partial\dot{\mathbf{x}} = A(t)\,\partial\mathbf{x} + B(t)\,\partial u\,, \quad \partial\mathbf{x}(0) = 0\,,$$

where $A(t) = \left[\, \frac{\partial J_k}{\partial\theta_1}\, u(t) \;\cdots\; \frac{\partial J_k}{\partial\theta_n}\, u(t) \,\right]$ and $B(t) = J_k(\theta)$. The discrete version of the linearized system is given by

$$\delta\mathbf{x}(k+1) = \Psi(k)\,\delta\mathbf{x}(k) + \Gamma(k)\,\delta u(k)\,, \quad \delta\mathbf{x}(0) = 0\,, \quad (28)$$

where $\Psi(k) = e^{A(kh)h}$, $\Gamma(k) = \int_{kh}^{(k+1)h} e^{A(kh)s} ds\, B(kh)$ and h is the sampling time period. Then, solving (28) for a horizon M we have

$$\delta\mathbf{x}(M) = D_M\,\delta u\,, \quad (29)$$

where $D_M = [\prod_{j=1}^{M-1} \Psi(j)\;\Gamma(0),\; \prod_{j=2}^{M-1} \Psi(j)\;\Gamma(1),\; \cdots,\; \Psi(M-1)\;\Gamma(M-2),\; \Gamma(M-1)\,]$. Therefore, $\nabla_{\bar{u}} F = D_M$ since D_M relates infinitesimal variations in $\bar{u}$ with infinitesimal variations in $\mathbf{x}(M)$. The update law (27) can be implemented in iterative form according to

$$\bar{u}(k+1) = \bar{u}(k) - \gamma_k\, [\,\nabla_{\bar{u}} F(\mathbf{x}_0, \bar{u}(k))\,]^{\dagger}\, \epsilon(k)\,, \quad (30)$$

where the scalar γ_k can be found by using a line search technique (Lizarralde et al., 2003). It is worth mentioning that the presented approach can be extended to also include additional constraints in both task and operational space. Constraints such as the joint limits and collision avoidance can then be added to the trajectory following, as presented in Seereeram and Wen (1995).

5.2 Replanning in the Velocity Space

The proposed replanning method consists of using the robot velocities in the Cartesian space to estimate the homogeneous transformation T_{bb_1} implicitly, rather than explicitly representing this transformation. In this context, the estimation of T_{bb_1} is equivalent to the estimation of the linear velocity $\upsilon_c(t)$ and angular velocity $\omega_c(t)$, which solve the following differential equations:

$$\dot{p}(t) = \upsilon_c(t)\,, \quad p(0) = 0;\; p(T) = p_{bb_1}\,, \quad (31)$$

$$\dot{R}(t) = \omega_c(t) \times R(t)\,, \quad R(0) = I;\; R(T) = R_{bb_1}\,. \quad (32)$$

In order to avoid singularities we can rewrite (31)-(32) in terms of the unit quaternion q_{bb_1} obtained from R_{bb_1} as

$$\dot{p}(t) = \upsilon_c(t)\,, \quad p(0) = 0;\; p(T) = p_{bb_1}\,, \quad (33)$$

$$\dot{q}(t) = \frac{1}{2} J_q^{\mathsf{T}}(q)\,\omega_c(t)\,, \quad q(0) = [1\;\; 0_{1\times 3}]^{\mathsf{T}};\; q(T) = q_{bb_1}\,. \quad (34)$$

Now, we consider that the configuration of calibration points $\lambda_1, \lambda_2, \ldots, \lambda_N$ with respect to the frames $\bar{E}_b$ and $\bar{E}_{b_1}$ can be represented by

$$\mathbf{x}_{0,grid} = [\; \mathbf{x}_{b\lambda_1}^{\mathsf{T}} \;\; \mathbf{x}_{b\lambda_2}^{\mathsf{T}} \;\; \cdots \;\; \mathbf{x}_{b\lambda_N}^{\mathsf{T}} \;]^{\mathsf{T}}\,,$$

$$\mathbf{x}_{d,grid} = [\; \mathbf{x}_{b_1\lambda_1}^{\mathsf{T}} \;\; \mathbf{x}_{b_1\lambda_2}^{\mathsf{T}} \;\; \cdots \;\; \mathbf{x}_{b_1\lambda_N}^{\mathsf{T}} \;]^{\mathsf{T}}\,,$$

where $\mathbf{x}_{b\lambda_i} = [\, p_{b\lambda_i}^{\mathsf{T}} \;\; q_{b\lambda_i}^{\mathsf{T}} \,]^{\mathsf{T}}$ and $\mathbf{x}_{b_1\lambda_i} = [\, p_{b_1\lambda_i}^{\mathsf{T}} \;\; q_{b_1\lambda_i}^{\mathsf{T}} \,]^{\mathsf{T}}$, for $i = 1, \cdots, N$. Then, rewriting (33) and (34) in vectorial notation we have

$$\dot{\mathbf{x}} = \bar{J}\,\mathbf{v}_c\,, \quad (35)$$

where

$$\bar{J} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} \\ 0_{4\times 3} & \frac{1}{2} J_q^{\mathsf{T}}(q_{\lambda_1}) \\ I_{3\times 3} & (p_{\lambda_1} - p_{\lambda_2})\times \\ 0_{4\times 3} & \frac{1}{2} J_q^{\mathsf{T}}(q_{\lambda_2}) \\ \vdots & \vdots \\ I_{3\times 3} & (p_{\lambda_1} - p_{\lambda_N})\times \\ 0_{4\times 3} & \frac{1}{2} J_q^{\mathsf{T}}(q_{\lambda_N}) \end{bmatrix}$$

$\mathbf{x}_{\lambda_i} = \left[p_{\lambda_i}^{\mathsf{T}} \;\; q_{\lambda_i}^{\mathsf{T}}\right]^{\mathsf{T}}$ is the end-effector posture at the calibration point λ_i, and $\mathbf{v}_c = [\, \upsilon_c^{\mathsf{T}} \;\; \omega_c^{\mathsf{T}} \,]^{\mathsf{T}}$. Similarly to (25), we can write the final state error as

$$e = F(\mathbf{x}_0, \bar{\mathbf{v}}_c(\tau)) - \mathbf{x}_d = 0\,, \quad (36)$$

where $\mathbf{x}_0 = \mathbf{x}_{0,grid}$ and $\mathbf{x}_d = \mathbf{x}_{d,grid}$. The gradient of F can be obtained from the system (35) linearized around a trajectory $(\mathbf{x}, \mathbf{v}_c)$ by

$$\partial\dot{\mathbf{x}} = \bar{A}(t)\,\partial\mathbf{x} + \bar{B}(t)\,\partial\mathbf{v}_c\,, \quad \partial\mathbf{x}(0) = 0\,,$$

where $\bar{A}(t) = \left[\frac{\partial \bar{J}}{\partial \mathbf{x}_{\lambda_1}}\,\mathbf{v}_c(t) \cdots \frac{\partial \bar{J}}{\partial \mathbf{x}_{\lambda_N}}\,\mathbf{v}_c(t)\right]$ and $\bar{B}(t) = \bar{J}(\mathbf{x})$. According to (30), the update law for $\bar{\mathbf{v}}_c$ is given by

$$\bar{\mathbf{v}}_c(k+1) = \bar{\mathbf{v}}_c(k) - \gamma_k\, [\,\nabla_{\bar{\mathbf{v}}_c} F(\mathbf{x}_0, \bar{\mathbf{v}}_c(k))\,]^{\dagger}\, e(k)\,. \quad (37)$$

5.3 Kinematic control

Here, we present the design of the kinematic control strategy that guarantees the tracking of a trajectory planned in the original location $\bar{E}_b$, when the robot base is positioned in the current location $\bar{E}_{b_1}$. First, we suppose that the homogeneous transformations T_{bb_1} is *known* and T_{be} is provided, where $T_{be} = T_{bb_1}\, T_{b_1 e}$. If the frames $\bar{E}_b$ and $\bar{E}_{b_1}$ are fixed relative to each other, the spatial velocity of the end-effector frame $\bar{E}_e$ satisfies:

$$\mathbf{v}_{be} = \mathrm{Ad}_{T_{bb_1}}\,\mathbf{v}_{b_1 e}\,, \quad (38)$$

where the term Ad_T denotes the adjoint mapping used to transform the velocities appropriately (Murray et al., 1994). The relationship between the joint velocity and the end-effector velocity can be used to drive a robot manipulator from one end-effector configuration to another without calculating the inverse kinematics for the manipulator. Then, if J_k is invertible, we can write the kinematic control problem as

$$\dot{\theta}_{b_1 e} = J_k^{-1}(\theta_{b_1 e})\,\mathbf{v}_{b_1 e}\,. \quad (39)$$

Substituting (38) into (39) yields:

$$\dot{\theta}_{b_1 e} = J_k^{-1}(\theta_{b_1 e})\,\mathrm{Ad}_{T_{bb_1}}^{-1}\,\mathbf{v}_{be}\,, \quad (40)$$

where $\mathbf{v}_{be} = [\, v_{be} \;\; \omega_{be} \,]^{\mathsf{T}}$ and the position and orientation control signals can be given respectively by

$$v_{be} = \dot{p}_{bd} + K_p\,(\, p_{bd} - R_{bb_1}\, p_{b_1 e} - p_{bb_1}\,), \quad (41)$$

$$\omega_{be} = \omega_{bd} + K_o\, e_{qv}, \quad e_q = q_{bb_1} * q_{b_1 e} * q_{bd}^{-1}, \quad (42)$$

where $\omega_{bd} \in \mathbb{R}^3$ is the desired angular velocity, K_o is the orientation gain matrix, $e_{qv} \in \mathbb{R}^3$ is the vectorial part of the error quaternion e_q and the symbol "$*$" denotes the operator of the quaternion product. The block diagram of a closed-loop kinematic control scheme is shown in Fig. 3 and the stability analysis for the kinematic control scheme given by (41)-(42) is presented in (Leite et al., 2009).

Now, we suppose that the homogeneous transformations T_{bb_1} is *unknown* and T_{be} is provided, where $T_{be} = T_c\, T_{b_1 e}$

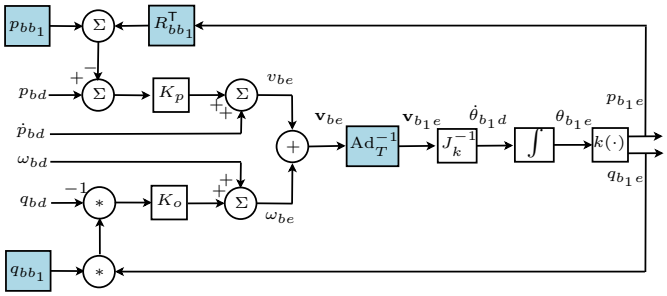

Fig. 3. Kinematic control scheme for the robot posture.

and $T_c = \widetilde{T}_{bb_1}$. From the definition of spatial velocity (Murray et al., 1994) and by using the chain rule we have:

$$\mathbf{v}_{be} = \mathbf{v}_c + \mathrm{Ad}_{T_c}\,\mathbf{v}_{b_1e}\,, \quad \mathrm{Ad}_{T_c} = \begin{bmatrix} R_c & p_c \times R_c \\ 0 & R_c \end{bmatrix}. \quad (43)$$

Then, if J_k is invertible, the kinematic control problem in the joint space becomes

$$\dot{\theta}_{b_1e} = J_k^{-1}(\theta_{b_1e})\,\mathrm{Ad}^{-1}_{T_c}\,[\mathbf{v}_{be} - \mathbf{v}_c]\,, \quad (44)$$

where $\mathbf{v}_c$ is obtained by using the proposed calibration method in the velocity space.

6. SIMULATIONS AND EXPERIMENTS

In this section, simulation and experimental results obtained with a 6-DoF robot manipulator are presented to illustrate the performance and viability of the proposed scheme. The robot used is the Zebra Zero (Integrated Motions, Inc.) with link lengths $l_1 = 279.4\,mm$ and $l_2 = 228.6\,mm$, for links 1 and 2 respectively. The control parameters are $K_p = 20\mathrm{I}\,mm\,s^{-1}$, $K_o = 10\mathrm{I}\,rad\,s^{-1}$, $\beta = 0.75$, and $u(0) = 0$. The task consists of following a reference trajectory in the xz-plane described by

$$p_{bd}(t) = [\,r_o\,sin(\omega_o\,t)\;\;0\;\;r_o\,cos(\omega_o\,t)\,]^{\mathsf{T}},$$

where $r_o = 75\ mm$ and $\omega_o = \frac{\pi}{5}\ rad/s$ are the ratio and the angular velocity of the trajectory respectively.

6.1 Experimental Results

A calibration grid with 50 endpoints, experimentally measured by using the robotic manipulator was employed to estimate the configuration error of the robot base. The actual base location of the robot was dislocated with respect to the planned location and the algorithm of Section 3.1 was applied to estimate the location of the base. Figure 4 shows the estimated values of the base location for the first 16 iterations.We see that the estimated location converges to the values $p_{bb_1} = [-11.8\ -57.7\ 2.6]^{\mathsf{T}}\ mm$ and $\Phi_{bb_1} = [0.38\ 0\ 0]^{\mathsf{T}}\ rad$. The estimated location is to be compared to the location of the base measured during the experiments, i.e., $\bar{p}_{bb_1} \approx [-10\ -60\ 0]^{\mathsf{T}}\ mm$ and $\bar{\Phi}_{bb_1} \approx [\frac{\pi}{9}\ 0\ 0]^{\mathsf{T}}\ rad$.

It must be noted that during the experiments it was not possible to dislocate the robot base by exactly p_{bb_1}, Φ_{bb_1}, so these only give a rough estimate of the actual location of the robot base. Due to the rather large error involved with moving the robot from one location to another, we therefore can not expect very accurate results. What is clear from the experiments, however, is that the algorithm converges relatively quickly as approximately 5 iterations are needed. We also see that the estimated values p_{bb_1} and Φ_{bb_1} are fairly close to the expected values given by $\bar{p}_{bb_1}$ and $\bar{\Phi}_{bb_1}$.

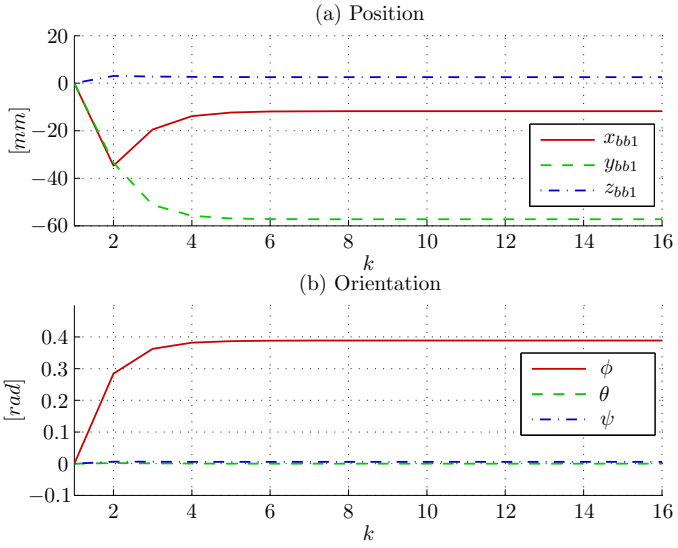

Fig. 4. Estimated base location.

6.2 Simulation Results

Here we present simulation results to show the feasibility of the calibration and replanning method presented in Section 5. A virtual calibration grid with 18 endpoints was used in the calibration phase. The configuration error $\widetilde{T}_{bb_1}$ between the planned and actual base locations, estimated from the experimental results in the previous section, was employed to generate the new configuration of the endpoints when the robot is positioned in the actual location. Figure 5 depicts the calibration control signal, composed of the linear velocity $v_c(t)$ and the angular velocity $\omega_c(t)$, that guarantee the successful execution of the planned trajectory. The figure shows the control signal for the first $5\,s$ only, i.e., for the time it takes for the end effector to catch up with the planned trajectory.

Figure 6 illustrates the tracking of a reference trajectory $p_{bd}(t)$ (dashdot line) replanned with respect to the actual location $\bar{E}_{b_1}$ using the trajectory replanning method in the velocity space. From this simulation, it is observed that good performance was achieved for the trajectory tracking (solid line) also when the robot base is misplaced with respect to the location used during planning.

7. CONCLUSIONS

This work addresses the remote calibration and trajectory replanning for robot manipulators. A trajectory replanning strategy in the Cartesian velocity space is proposed to solve the path planning problem in the presence of uncertainties in the absolute location of the robotic cell with respect to the location used during planning. Furthermore, a calibration method is presented to estimate the uncertainty in the robot configuration from a calibration grid using internal and external sensors. Simulation and experimental results obtained with a 6-DoF robot manipulator and a real calibration grid illustrate the viability of the proposed planning and control scheme.

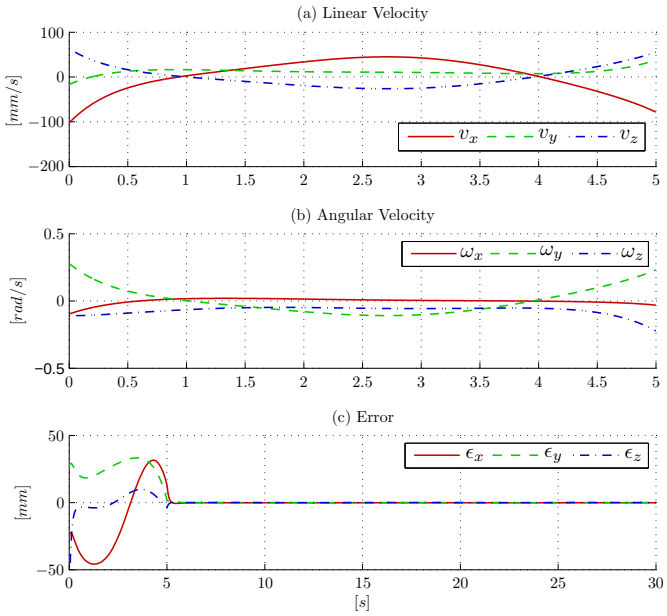

Fig. 5. The calibration control signal $\mathbf{v}_c$ and the corresponding error. Figure (a) shows the linear part and Figure (b) the angular part of the calibration control signal which guarantees that the trajectory is followed. We see from the error in Figure (c) that the end effector reaches the desired trajectory after approximately $5\,s$. Note the different time scales on the figures.

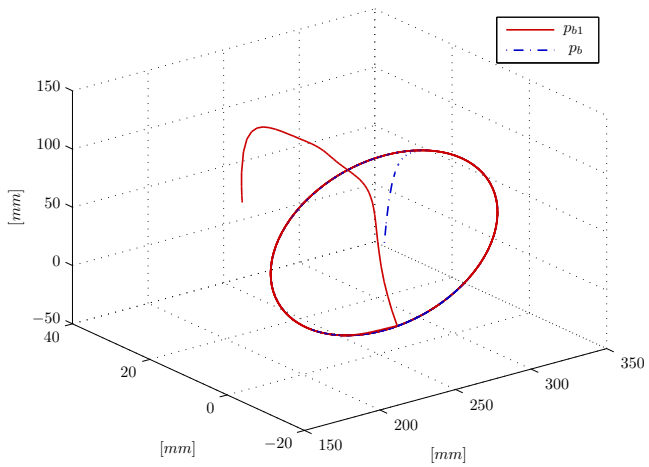

Fig. 6. Trajectory replanning in the Cartecian velocity space.

REFERENCES

Augustson, T.M. and Meggiolaro, M.A. (2010). *Vision Based In-Situ Calibration of Robots: with Application in Subsea Interventions*. LAP Lambert Academic Publishing, Saarbrücken, Germany, 1st edition.

Beyer, L. and Wulfsberg, J. (2004). Practical Robot Calibration with ROSY. *Robotica*, 22(5), 505–512.

Canepa, G., Hollerbach, J.M., and Boelen, A.J.M.A. (1994). Kinematic Calibration by Means of a Triaxial Accelerometer. In *IEEE Int. Conf. on Robotics & Automation*, 2776–2782. San Diego, CA, USA.

Flandin, G., Chaumette, F., and Marchand, E. (2000). Eye-in-hand/eye-to-hand Cooperation for Visual Servoing. In *IEEE Int. Conf. on Robotics & Automation*, 2741–2746. San Francisco, CA.

Gao, J.B. and Harris, C.J. (2002). Some remarks on kalman filters for the multisensor fusion. *Information Fusion 3*, 191–201.

Hollerbach, J.M. and Wampler, C.W. (1996). A Calibration Index and Taxonomy for Robot Kinematic Calibration Methods. *The Int. Journal of Robotics Research*, 15(6), 573–591.

Hutchinson, S., Hager, G., and Corke, P. (1996). A Tutorial on Visual Servo Control. *IEEE Trans. on Robotics and Automation*, 12(5), 651–670.

Ikits, M. and Hollerbach, J.M. (1997). Kinematic Calibration using a Plane Constraint. In *IEEE Int. Conf. on Robotics & Automation*, 3191–3196. Albuquerque, NM, USA.

Lei, S., Jingtai, L., Weiwei, S., Shuihua, W., and Xingbo, H. (2004). Geometry-based Robot Calibration Method. In *IEEE Int. Conf. on Robotics & Automation*, 1907–1912. New Orleans, LA.

Leite, A.C., Lizarralde, F., and Hsu, L. (2009). Hybrid Adaptive Vision-Force Control for Robot Manipulators Interacting with Unknown Surfaces. *The Int. Journal of Robotics Research*, 28(7), 911–926.

Lizarralde, F., Nunes, E.V.L., Hsu, L., and Wen, J.T. (2003). Mobile robot navigation using sensor fusion. In *IEEE Int. Conf. on Robotics & Automation*, 1–6.

Lizarralde, F. and Wen, J.T. (1996). Feedback stabilization of nonholonomic systems in presence of obstacles. In *IEEE Int. Conf. on Robotics & Automation*, 2682–2687. Minneapolis, MN.

Mooring, B., Driels, M., and Roth, Z. (1991). *Fundamentals of Manipulator Calibration*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

Motta, J.M.S.T., de Carvalho, G.C., and McMaster, R.S. (2001). Robot Calibration using a 3D Vision-based Measurement System with a Single Camera. *Robotics and Computer Integrated Manufacturing*, 17(6), 487–497.

Murray, M., Li, Z., and Sastry, S.S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press LLC, Boca Raton, FL, USA, 1st edition.

Seereeram, S. and Wen, J.T. (1995). A global approach to path planning for redundant manipulators. *IEEE Trans. on Robotics and Automation*, 11(1), 152–160.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2008). *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Inc., 1st edition.

Sorenson, H.W. (1985). *Kalman Filtering: Theory and Application*. IEEE Press, New York, NY, USA.

Trevelyan, J., Kang, S.C., and Hamel, W.R. (2008). Robotics in Hazardous Applications. In B. Siciliano and O. Khatib (eds.), *Springer Handbook of Robotics*, 1101–1126. Springer.