

AN EXPERIMENTAL VALIDATION OF COLLISION-FREE TRAJECTORIES FOR PARALLEL MANIPULATORS[#]

G. Carbone¹, F. Gómez-Bravo², and O. Selvi³

¹Laboratory of Robotics and Mechantronics (LARM), University of Cassino and South Lantium, Cassino, Italy

²Grupo de Electronica y Mecatronica, Escuela Técnica Superior, Universidad de Huelva, La Rabida, Spain

³Izmir Institute of Technology, Izmir, Turkey

This article proposes a systematic approach for validating the trajectories generated by a motion planning method in presence of obstacles for parallel manipulators having less than 6 degrees of freedom (d.o.f.s). The planning algorithm is based on combining a quick random search algorithm together with an optimization method, which aims to obtain feasible and reliable trajectories. The validation approach uses a probabilistic method, which includes Kalman filtering of experimental data. Experimental tests have been carried out by operating a Cassino Parallel Manipulator (CaPaMan) prototype at LARM in Cassino. Results are reported and discussed to show the effectiveness of the proposed approach to generate and validate suitable collision-free trajectories for parallel manipulators.

Keywords: Collision-free trajectories; Parallel manipulators motion planning; Trajectory validation.

1. INTRODUCTION

Autonomous precise manipulation among obstacles is a great challenge and represents a valuable tool for numerous tasks. For instance, surgical robot applications will benefit from the development of manipulators capable of avoiding collision with different elements involved in surgery (Lueth and Bier, 1999). Considerable research activity has been carried out in order to obtain optimal paths with serial robots, and the corresponding literature is very rich (Barraquand and Latombe, 1991; Berenson et al., 2009; Bohlin and Kavraki, 2000; Ceccarelli et al., 1996; Da Graça et al., 2010; Fang and Dissanayake, 1998; Gu and Ceccarelli, 2012; Khatib, 1986; Lahouar et al., 2006; Lin et al., 1983; Ong and Gilbert, 1998; Saramago and Valder, 2001; Saramago and Ceccarelli, 2004; Savavanan et al., 2008; Shin and Mckay, 1986; Simeon et al., 2000).

For instance, a procedure to determine a cubic polynomial joint trajectory through an algorithm for minimizing the traveling time, subject to physical

Received November 1, 2011; Accepted April 1, 2012

[#]Communicated by M. Ceccarelli and V. Mata.

Correspondence: G. Carbone, Laboratory of Robotics and Mechantronics (LARM), Department of Civil and Mechanical Engineering, University of Cassino and South Lantium, Via Di Biasio 43, Cassino, FR 03043, Italy; E-mail: carbone@unicas.it

constraints on joint velocities, accelerations, and jerks has been proposed in Lin et al. (1983). A solution to the problem of minimizing the power consumption of moving a serial robotic manipulator along a specified end-effector path subject to input torque/force constraints, by taking into account the dynamics of the manipulator has been presented in Shin and McKay (1986). A neural network-based method for time-optimal trajectory planning has also been considered in Fang and Dissanayake (1998). In Saramago and Ceccarelli (2004), a path planning strategy that takes into account actuating energy has been developed. In Savavanan et al. (2008), an evolutionary algorithm is applied in order to obtain collision-free trajectories for the end effector. An evolutionary computation for the motion planning of a redundant manipulator has been considered in Da Graça et al. (2010). In Berenson et al. (2009), a random search algorithm for a 3 degree of freedom (d.o.f.s) serial robot has been described. Similarly, attempts have been made to address the path planning of robots having parallel architecture (Carbone et al., 2008; Dash et al., 2005; Dasgupta and Mruthyunjaya, 2000; Merlet, 2001; Sen et al., 2003). Nevertheless, it is still missing a systematic approach to generate optimal collision-free trajectories for parallel manipulators with less than 6 d.o.f.s. In fact, these types of manipulators have a very narrow workspace. Additionally, there might be singularities within the workspaces that are not reachable or must be carefully avoided due to control problems (Ceccarelli, 2004; Castelli et al., 2008; Itu and Pisla, 2009; Liua and Maa, 2010; Merlet, 2000; Pisla et al., 2012).

This article proposes a systematic approach for validating the trajectories generated by a motion planning method that provides collision-free optimal paths for parallel robots having less than 6 d.o.f.s in presence of obstacles. After that, validation test have been implemented by taking into account the values measured by set of sensors. In validation tests, values of outstanding variables are not directly available and it is necessary to estimate them from the data provided by sensors. But, sensors and/or the input signals are affected by a significant uncertainty so that deterministic techniques do not provide enough tools to estimate the system state (Maybeck, 1979). Therefore, it is necessary to use probabilistic approaches. Several probabilistic methods have been developed to cope with these problems (Grewal and Andrews, 2002; Gomez-Bravo et al., 2007; Kalman, 1960; Schiele and Crowley, 1994). The most known technique is the Kalman Filter (KF) (Kalman, 1960) and the Extended Kalman Filter (EKF) (Gordon et al., 1993; Ribeiro, 2004). The KF provides an optimal estimation of the system state, but only when linear systems and Gaussian noises are involved. For nonlinear systems, the EKF can be used, which approximates the system by its first order linearization.

In this article, the general collision-free motion planning and its basic characteristic are described in Section 2. Section 3 reports a case of study of the proposed path planning procedure to obtain collision-free optimal trajectories for the parallel manipulator Cassino Parallel Manipulator (CaPaMan) that has been designed and built at Laboratory of Robotics and Mechatronics (LARM) in Cassino (Ceccarelli, 1997, 2010; Hernandez-Martinez et al., 2010; Liang et al., 2009). Section 4 proposes a systematic approach for validating the computed collision-free trajectories as based on a suitable Kalman filtering in order to decrease the sensors uncertainty. Moreover, Section 5 describes the experimental set-up and tests that have been carried out by operating CaPaMan parallel manipulator. Experimental

data have been processed by means of the proposed systematic approach. Finally, theoretical and experimental data are compared.

2. COLLISION-FREE MOTION PLANNING

A systematic approach for computing collision-free optimal trajectories for parallel robots needs to take into account different aspects, such as presence, size, and shape of obstacles; number of d.o.f.s that can be controlled at same time; movable ranges, maximum reachable velocities, accelerations, and jerks; control speed (clock rate), safety issues (Brancati et al., 2007; Yan et al., 2010).

The technique applied in this article presents specific peculiarities when comparing with other similar approaches. Thus, different to the method proposed in Savavanan et al. (2008), where the collision test takes only into account the end-effector trajectory, the method applied in this article considers possible collision of the whole manipulator. In Da Graça et al. (2010), the collision test is implemented in the workspace, however, only 2D simulated representation are considered. On the contrary, in this approach realistic 3D models are considered and real experiments have been performed. In Berenson et al. (2009), a random search in the configuration space is developed, but no optimization procedure is implemented.

The proposed motion-planning algorithm focuses attention on obtaining collision-free trajectories in the joint space Ω . The dimension of Ω will be equal to the manipulator d.o.f.s. The proposed method attempts to increase the reliability of the trajectories by optimizing the distance to the obstacles and the length of the path, both in the configuration space. At the same time, the trajectory has to accomplish the constraints imposed by the mechanical characteristics of the manipulator.

The solution addressed in this article is based on the assumption that the manipulator will evolve in a known scenario. The task of the robot will be specified as a set of configurations (the task configurations – T.C.s) that the tool carried by the manipulator has to reach (position and orientation of a target). Additionally, the scenario will be described as a set of obstacles distributed along the Cartesian workspace of the robot. The proposed approach has to determine the collision-free sub-space ($\Omega_f \in \Omega$), i.e., the set of configurations in which no collision exists. Thus, the planning method has to provide a sequence of joint configurations (a joint path) Ξ , accomplishing $\Xi \in \Omega_f$.

This procedure can be suitable, for instance, for surgery applications. However, the requirements and limitations in the operating surgical field will depend on the type of procedure as it is stated, for example, in Bozovic (2008); Rosen et al. (2011). Then, the proposed technique can be applied in those situations in which the scenario (the patient and the surgical instrument positions) can be defined prior to the operation. Additionally, no changes in the obstacles distribution are expected and an optimal motion is desired. If that is the case, T.C.s would represent different points over the patient that the tool would have to reach in order to perform the surgical task. Then, considering the inverse kinematics model, the T.C.s are turned to a set of task-points (T.P.s) in the joint space that the manipulator has to reach. After that, a controller will make the manipulator joints follow the joint path so that the robot accomplishes the predefined task. The flow-chart in Fig. 1 illustrates the above-mentioned general procedure.

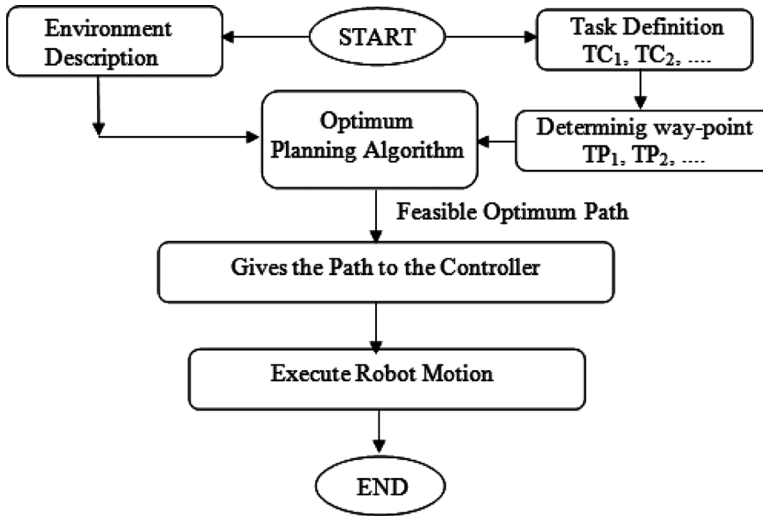


Figure 1 Flow chart of the general planning/control approach.

The first step of the proposed technique involves generating the joint matrix Θ that contains information about the regions of the joint space that will not present collision with the obstacles of the scenario (i.e., Ω_f). Then, the information on collisions is transferred from the Cartesian to the joint space. At first, the proposed algorithm uses a mechanism for collision detection based on a discrete description of the Cartesian space and manipulator geometry. The manipulator is described by means of a set of prisms defined by surfaces and edges. At a later time, points regularly spaced will represent those edges and surfaces (the cluster points C_p). In this way, the manipulator will get defined by a spotted set of C_p being defined by three Cartesian coordinates. At same time, a grid matrix for the Cartesian space (the Cartesian matrix Ψ) is defined. Each element of this matrix represents a portion of the space, whose value is defined according to

$$\Psi_{ijk} = \Psi([x, y, z]) = \begin{cases} 1 & \text{if } [x, y, z] \text{ is occupied by an obstacle} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Given a manipulator's configuration, Ξ_h , the collisions detection can be performed by applying the equations of the direct kinematics to each one of the C_p . Using these equations, it is possible to know the Cartesian clusters's positions at the joint configuration $\Xi_h({}^h C_p[x, y, z])$. Then, the following free collision configuration condition can be derived.

2.1. Collision-Free Configuration Condition

Let a manipulator be geometrically defined by k cluster points C_p , let Ξ_h be a configuration of a ε -grid representation of the manipulator's joint space Ω , and

Ψ the Cartesian matrix of the manipulator's environment. Then, Ξ_h accomplishes $\Xi_h \in \Omega_f$ if and only if

$$\sum_{l=1}^k \Psi({}^h C_p[x, y, z]_l) = 0 \tag{2}$$

The collision detection can be implemented by checking the cells of the Cartesian matrix associated to each of the robot cluster position. From (2), if only one ${}^h C_p$ is located in an occupied area, it means that this configuration presents a collision. Thus, it is possible to use Ψ to determine Θ . An iterative algorithm has been implemented, testing each configuration against collision. The flow chart in Fig. 2 illustrates the above-mentioned steps.

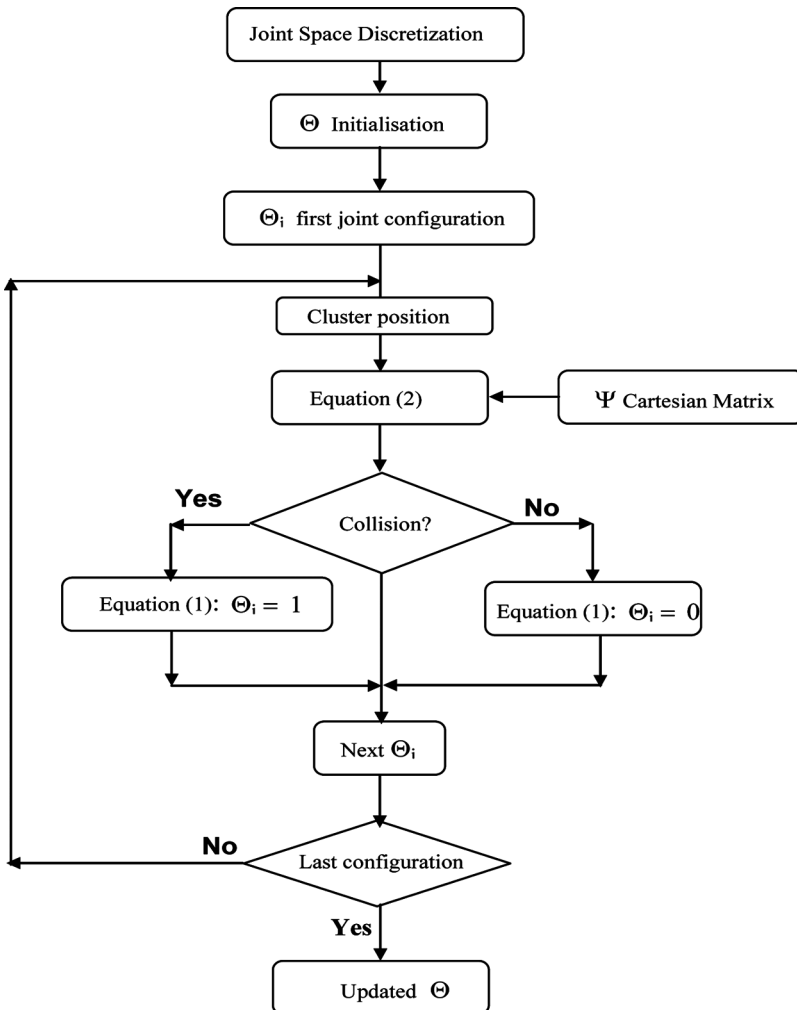


Figure 2 Flow chart describing the Joint Matrix generation procedure.

Once the joint matrix Θ has been obtained, and considering the T.P.s as via-points, the optimum-planning algorithm will provide an optimum feasible path (Fig. 1). The planning algorithm is based on generating optimal joint paths by means of a random generation's algorithm (LaValle and Kuffner, 1999), represented by the "Rapidly Exploring Random Trees" (RRT), along with an optimization method provided by Genetics Algorithms (G.A.).

The general idea is to build a initial population of nonoptimal free collision trajectories provided by RRT, and apply the G.A. in order to provide a solution that improves a fitness function (Gomez-Bravo et al., 2009). The G.A. apply the typical procedures of this technique (cross, mutation, evaluation, and selection) over the initial population. The fitness function that has been proposed for evaluating a joint path $\Xi = \{\Xi_1, \dots, \Xi_m\}$ is:

$$J(\Xi) = \Lambda + \sum_{i=1}^m d(\Xi_i, \Xi_{i+1}) + k_i \cdot \beta \tag{3}$$

The first component Λ has the mission of penalizing the existence of collision in accordance with (1):

$$\Lambda = \begin{cases} \infty & \text{if } \Psi[\Xi] > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Thus, if one of the configurations presents collision, the joint path will be discarded. The second component of the fitness function is the summation of two terms for the m configurations that composes the evaluated trajectory. The term $d(\Xi_i, \Xi_{i+1})$ represents the Euclidean distances between each two consecutive configurations. The term k_i weights the proximity to the obstacles using the following expression:

$$k_i = \begin{cases} 1 & \text{if } \rho_i > \delta \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where ρ_i is the distance from the configuration Ξ_i to the nearest collision configuration. Then, the fitness function will increments a value of β each time a configuration lays too close of a possible collision (i.e., when ρ_i is lower than δ).

According to this fitness function, the algorithm will supply the shorter trajectory that is far enough to the obstacles, both in the joint space. It is worth highlighting that, shorter path in the joint space represent less effort and energy consumption as the motion of the actuator are shorter. Likewise, being far from the obstacles in Ω means that not only the end effector but also the whole manipulator is taken into account for avoiding collision.

In Section 4, the previously mentioned general procedure is implemented with a specific case of study by referring to the parallel manipulator CaPaMan.

3. PROBABILISTIC TECHNIQUES FOR EXPERIMENTAL VALIDATION

3.1. Probabilistic Approach

The state estimation problem using probabilistic techniques can be formulated according to Ribeiro (2004). If (9) is the model of a non-linear and time-variant system,

$$\begin{aligned} X_{k+1} &= f(X_k, u_k, w_k) \\ Z_k &= h(X_k, v_k) \end{aligned} \quad (6)$$

where

- k denotes the time instant
- X is the state vector.
- $f(\dots)$ defines the system dynamics.
- u is the inputs vector.
- w is the vector that models the system error sources.
- Z is the sensor measurements vector.
- $h(\dots)$ defines the sensor model.
- v is the vector that models the sensor measurement error sources.

The problem consists in obtaining the best estimation of X_k that minimizes the error for some given criteria.

From the Bayesian point of view, the system propagates the probability density function (PDF) of the state vector, conditioned on the sensor measurement data. That is, a function that defines the probability of a specific state vector being the real state of the system given the data provided by the sensor. More formally, the PDF can be specified by

$$P(X_k | Z_1 \dots Z_k, u_0 \dots u_{k-1}) \quad (7)$$

This expression defines the likelihood of being X , the real state at the time instant $t = k$, knowing the sensor measurement until $t = k$ and the system inputs until $t = k - 1$. Given this PDF, the state estimation is calculated by minimizing some criteria, as the mean, the mode, or the median (Ribeiro, 2004). Usually (Grewal and Andrews, 2002), linear relations have been used for modeling the evolution of the measurement provided by a sensor. In this case, a traditional Linear Time Invariant (LTI) representation is considered as

$$\begin{aligned} X_{k+1} &= AX_k + Bu_k + Gw_k \\ Z_{k+1} &= CX_k + v_{k+1} \end{aligned} \quad (8)$$

where w_k is a matrix representing the uncertainty of the model and v_k is a matrix modeling the natural noise of the sensor. Both will be modeled as zero mean Gaussians variables, with covariance matrices R and Q , respectively.

These equations are a particular case of Eq. (6), when linear relation are involved. This model will be completely observable due to the sensor measurements (Z).

3.2. Kalman Filtering

The Kalman Filter (KF) (Kalman, 1960) is a set of mathematical equations that supplies a computationally efficient state estimation of a linear system exposed to Gaussian noise and uncertainties. This estimation minimizes the mean quadratic error using the state system model and the sensor measurements. The filter algorithm is divided in two phases: prediction and correction. In the first one, the evolution of system state is predicted at time instant $k + 1$ using the data (state, input, and covariance matrices) available at instant k . In the second one, this prediction is corrected with the sensor measurement at time instant $k + 1$.

The following equations denote the prediction phase

$$\begin{aligned}\widehat{X}_{k+1}^* &= A\widehat{X}_k + BU_k \\ \widehat{P}_{k+1}^* &= A\widehat{P}_kA^T + GQG^T\end{aligned}\quad (9)$$

where \widehat{X}_{k+1}^* is the predicted value of the estate at the time instant $k + 1$, \widehat{P}_k is the covariance matrix of the estate components estimated at the time instant k , and \widehat{P}_{k+1}^* is the predicted covariance matrix at the time instant $k + 1$.

The following equations represent the correction phase

$$\begin{aligned}K_{k+1} &= \widehat{P}_{k+1}^*C^T(C\widehat{P}_{k+1}^*C^T + R)^{-1} \\ \widehat{X}_{k+1} &= \widehat{X}_{k+1}^* + K_{k+1}(Z_{k+1} - C\widehat{X}_{k+1}^*) \\ \widehat{P}_{k+1} &= (I - K_{k+1}C)\widehat{P}_{k+1}^*\end{aligned}\quad (10)$$

where K_{k+1} is known as the optimal Kalman gain, that allow obtaining the estimated value \widehat{X}_{k+1} . Likely, \widehat{P}_{k+1} , the final estimated covariance matrix, is also calculated from K_{k+1} .

The KF is only optimal when the equations in (9) are linear. For nonlinear systems the EKF can be used, but it is necessary to linearize the system (Grewal and Andrews, 2002) and the estimate is not optimal.

4. A CASE OF STUDY OF COLLISION-FREE MOTION

In the following, a case of study is presented by applying the procedures in Figs. 1 and 2 to CaPaMan parallel manipulator. The CaPaMan prototype has been already successfully tested for several applications, including the experimental evaluation of earthquake effects by means of a set of 9–12 accelerometers data, as reported in Carvalho et al. (1999); Selvi and Ceccarelli (2010). The proposed task, defining a sequence of T.C.s is shown in Fig. 3(a). This figure illustrates CaPaMan and the tool moving through the T.C.s T.C.1, T.C.2, T.C.3, and again T.C.1. The T.C.s have been selected so that the tool has to move around a cylindrical obstacle. Obviously, direct motion between the configurations will cause collisions. In Fig. 3(b), the trajectory followed by the end of the tool in the Cartesian space is represented with a continuous black line. It is to note, how this trajectory avoids colliding with the obstacle. The trajectory keeps within a circle which radius is 6cm, far from the obstacle. This proposed path planning illustrates the capability of the proposed method for providing free collision paths even if the selected

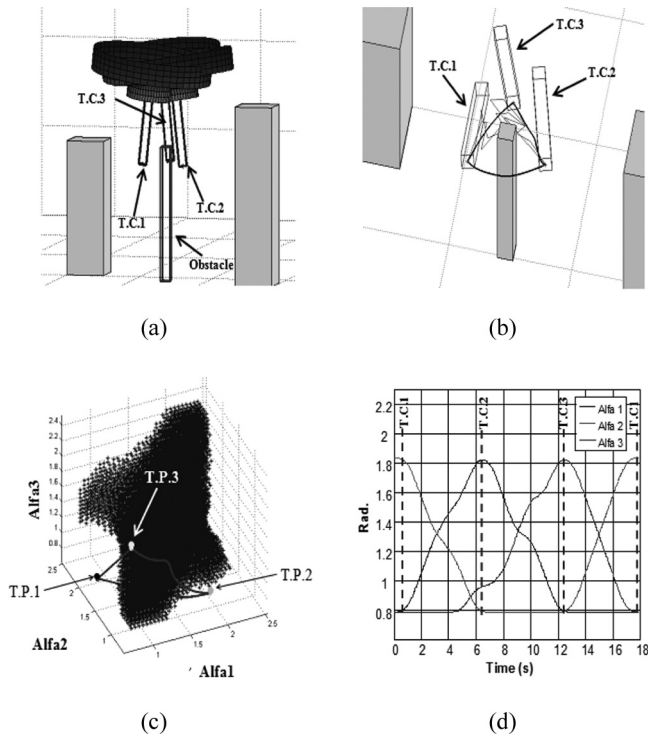


Figure 3 Results of collision-free motion planning for CaPaMan: (a) task configurations; (b) Cartesian trajectory of the tool; (c) CaPaMan joint space and the joint path and (d) joint variables evolution.

T.C. lay very close to the obstacles. Figure 3(c) illustrates the joint space and the joint path provided by the G.A. Figure 3(d) presents the evolution of the parallel manipulator joints variables. The dashed lines show the moments in which the correspondent T.P. is reached. The time required by the planning algorithm to obtain this trajectory was an average of 12 sec with a 2.93 GHz Intel Core i3.

5. EXPERIMENTAL VALIDATION OF THE PLANNED TRAJECTORY

A laboratory test-bed has been set-up for validating both the proposed collision-free motion planning and the proposed validation procedure. The test-bed consists of CaPaMan prototype together with suitable accelerometers that have been located beneath the movable plate. In particular, four 3-axial accelerometers have been properly installed at points P1 (P1x, P1y, and P1z), P2 (P2x, P2y, and P2z), P3 (P3x, P3y, and P3z), and P4 (P4x, P4y, and P4z) as shown in Fig. 4. The values of the linear accelerations can also be used for the calculation of all the components of both the angular accelerations and angular velocities for a 3D motion of a rigid body. For this purpose, the above redundant set of 12 linear accelerations allows a close-form direct calculation of all the components of the accelerations and angular velocities for a 3D motion of a rigid body, as demonstrated, for example in Peters et al. (2005) and Zappa et al. (2001). Moreover, the proposed symmetric set-up gives the possibility to minimize the position error in attaching the sensors to the movable plate.

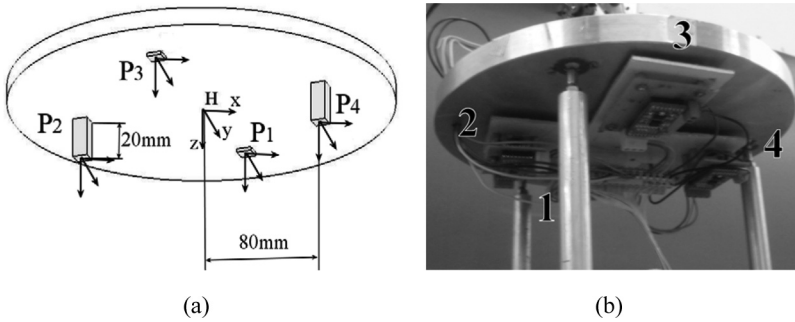


Figure 4 Location and orientation of accelerometers: (a) a scheme and (b) a photo of the sensors installed on CaPaMan.

The previously mentioned sensors are connected to a 5V power supply and to a National Instruments acquisition board, as described in the scheme of Fig. 5. The controller of CaPaMan has been synchronized with the acquisition board to obtain a synchronized measurement during the operation of CaPaMan. The control of the CaPaMan has been achieved by writing a suitable routine in agent communication language (ACL), a programming language that is a dedicated language of the CaPaMan controller.

A suitable virtual instrument has been developed in Lab View environment to manage the signals coming from the sensors. Then, the measured acceleration data from the accelerometers have been used to estimate the accelerations of the point H at the centre of the movable plate and the plate angular velocities.

The acceleration of a point P fixed on a rigid body with a position \mathbf{r} can be expressed as reported, for example, by Schopp et al. (2009), in the form.

$$\mathbf{a}_P = \mathbf{a}_B + \boldsymbol{\alpha}_B \times \mathbf{r} + \boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r}) \tag{11}$$

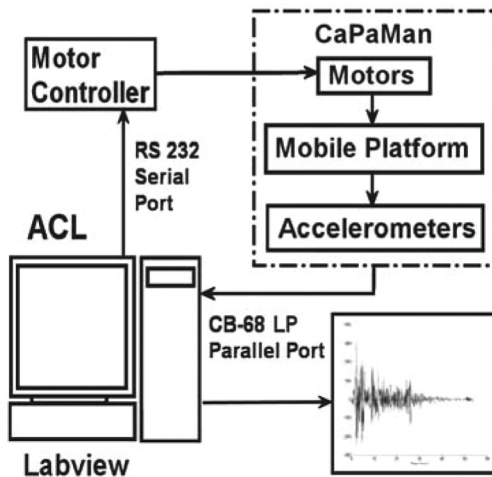


Figure 5 A scheme of the proposed test-bed.

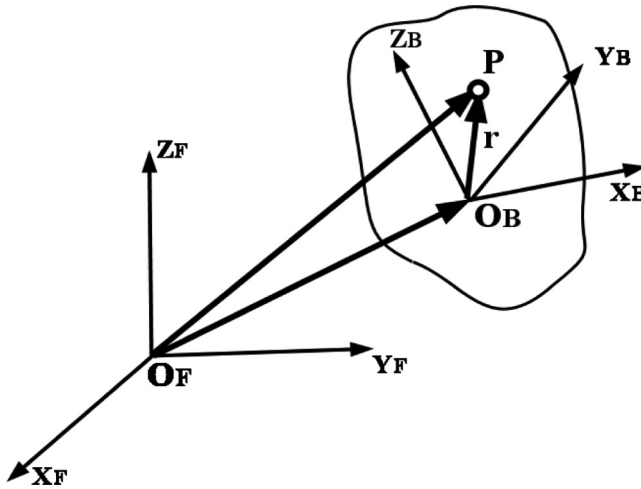


Figure 6 Point P in Frames OB and OF.

where acceleration \mathbf{a}_B , the angular velocity $\boldsymbol{\omega}_B$, and the angular acceleration $\boldsymbol{\alpha}_B$ are described for the relative movement of the rigid body OB with respect to the fixed frame OF. The terms of the equation $\boldsymbol{\alpha}_B \times \mathbf{r}$ can be described as tangential acceleration and $\boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r})$ as centripetal acceleration (Fig. 6).

In order to calculate the acceleration as measured by a sensor that is attached at position \mathbf{r} within a body, the sensitivity axis \mathbf{s} and the sensor's metrological signal offset \mathbf{a}_0 must be added in above equation

$$\mathbf{a}_S = \mathbf{s}^T (\mathbf{a}_B + \boldsymbol{\alpha}_B \times \mathbf{r} + \boldsymbol{\omega}_B \times (\boldsymbol{\omega}_B \times \mathbf{r})) + \mathbf{a}_0 \tag{12}$$

This can be written in vector form as

$$\mathbf{a}_S = \mathbf{c} \mathbf{z} + \mathbf{a}_0 \tag{13}$$

where

$$\mathbf{c} = \begin{bmatrix} S_x \\ S_y \\ S_z \\ S_z r_y - S_y r_z \\ S_x r_z - S_z r_x \\ S_y r_x - S_x r_y \\ -(S_y r_y + S_z r_z) \\ -(S_x r_x + S_z r_z) \\ -(S_x r_x + S_y r_y) \\ S_x r_y + S_y r_x \\ S_x r_z + S_z r_x \\ S_y r_z + S_z r_y \end{bmatrix} \quad \text{and} \quad \mathbf{z} = \begin{bmatrix} \mathbf{a}_{B,x} \\ \mathbf{a}_{B,y} \\ \mathbf{a}_{B,z} \\ \boldsymbol{\alpha}_{B,x} \\ \boldsymbol{\alpha}_{B,y} \\ \boldsymbol{\alpha}_{B,z} \\ \omega_{B,x}^2 \\ \omega_{B,y}^2 \\ \omega_{B,z}^2 \\ \omega_{B,x} \omega_{B,y} \\ \omega_{B,x} \omega_{B,z} \\ \omega_{B,y} \omega_{B,z} \end{bmatrix} \tag{14}$$

by using four sensors with totally 12 axis it is possible to directly compute the quadratic terms of α_B as well as \mathbf{a}_B and ω_B . So, the system becomes linear and can be written in matrix vector form as

$$\mathbf{y} = \mathbf{Mz} + \mathbf{a}_{0S} \tag{15}$$

where

$$\mathbf{y} = \begin{bmatrix} \mathbf{a}_{S1} \\ \mathbf{a}_{S2} \\ \vdots \\ \mathbf{a}_{S12} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} c_{S1} \\ c_{S2} \\ \vdots \\ c_{S12} \end{bmatrix} \quad \text{and} \quad \mathbf{a}_{0,S} = \begin{bmatrix} \mathbf{a}_{0,S1} \\ \mathbf{a}_{0,S2} \\ \vdots \\ \mathbf{a}_{0,S12} \end{bmatrix} \tag{16}$$

By inverting \mathbf{M} it is possible to calculate the relative body movement held by vector \mathbf{z} for a given measurement vector \mathbf{y} applying

$$\mathbf{z} = \mathbf{M}^{-1}(\mathbf{y} - \mathbf{a}_{0S}) \tag{17}$$

The values obtained from the sensors are filtered by applying a KF, so that a robust estimation of the acceleration and velocity components are obtained.

For this purpose, the filter has been conveniently configured in order to be applied to each accelerometer. Thus, the vector state of the accelerometer i , is composed by the values measured along each axe and the corresponding rate of change, i.e., $X_{1i} = P_{ix}$, $X_{2i} = \Delta a_{mix} / \Delta t \dots$, and so on. Accordingly, each state vector has six components. The model for predicting the future values is built with the matrices:

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad B = 0 \tag{18}$$

As was mentioned before, the uncertainty associated to the prediction is modeled by a zero mean Gaussian distribution, by defining the covariance matrix Q and the matrix G :

$$Q = \begin{bmatrix} \mu_{m1}^2 & \mu_{m1} \cdot \mu_{m2} & 0 & 0 & 0 & 0 \\ \mu_{m1} \cdot \mu_{m2} & \mu_{m2}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu_{m1}^2 & \mu_{m1} \cdot \mu_{m2} & 0 & 0 \\ 0 & 0 & \mu_{m1} \cdot \mu_{m2} & \mu_{m2}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu_{m1}^2 & \mu_{m1} \cdot \mu_{m2} \\ 0 & 0 & 0 & 0 & \mu_{m1} \cdot \mu_{m2} & \mu_{m2}^2 \end{bmatrix}; \quad G = I \tag{19}$$

where I is the identity matrix, and μ_{m1} and μ_{m2} are the standard deviations associated to the prediction uncertainty.

Table 1 Standard deviations

μ_{m1} (m/s ²)	μ_{m2} (m/s ³)	σ_{px} (m/s ²)	σ_{py} (m/s ²)	σ_{pz} (m/s ²)
0.01	0.05	0.5	0.5	0.5

The uncertainty of the virtual sensor has been modeled using a zero mean Gaussian distribution function with the covariance matrix R . Thus, the observation model is achieved by defining C and R :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad R = \begin{bmatrix} \sigma_{ax}^2 & 0 & 0 \\ 0 & \sigma_{ay}^2 & 0 \\ 0 & 0 & \sigma_{az}^2 \end{bmatrix} \quad (20)$$

where σ_{ax} , σ_{ay} , and σ_{az} are the standard deviations of the noise of each of the accelerator axes.

The characterization of the probabilistic model of the uncertainty gave the values in Table 1.

According to this, from each accelerometer i , a set of estimated values (a_{ix} , a_{iy} , and a_{iz}) are obtained. The value of the acceleration of the point H, and the angular velocity of the rigid body is calculated by using the equations:

$$\begin{bmatrix} \frac{-a_{1x} - a_{2x}}{2} \\ \frac{-a_{1y} - a_{2y}}{2} \\ \frac{-a_{1z} - a_{2z}}{2} \end{bmatrix} = \begin{bmatrix} a_{Hx} \\ a_{Hy} \\ a_{Hz} \end{bmatrix}; \quad \begin{aligned} \omega_X &= \pm \sqrt{\frac{1}{2} \left(\frac{a_{HY} - a_{P2Y} + a_{HX} - a_{P1X}}{r} + \frac{a_{P3Z} - a_{HZ}}{L} \right)} \\ \omega_Y &= \pm \sqrt{\frac{1}{2} \left(\frac{a_{P1X} - a_{HX} - a_{HY} - a_{P2Y}}{r} + \frac{a_{P3Z} - a_{HZ}}{L} \right)} \\ \omega_Z &= \pm \sqrt{\frac{1}{2} \left(\frac{a_{HY} - a_{P2Y} - a_{HX} - a_{P1X}}{r} - \frac{a_{P3Z} - a_{HZ}}{L} \right)} \end{aligned} \quad (21)$$

The previously mentioned equations have been derived from the theorem of accelerations of a rigid body and described in Eqs. (13)–(17).

Several experiments have been made in order to test the trajectories provided by the proposed approach. In the experiments, an obstacle is placed under the CaPaMan and motor position data needed for the trajectory avoidance is sent to motors with the CaPaMan motor controller by using the ACL programming language. Experiments have been made at different speeds. Twenty-one points on the trajectory are used for the motion. Time differences between the points are experimented as 0.09 sec, 0.45 sec, and 0.9 sec to see the motion at different speeds. The acceleration values are collected by the accelerometers on the mobile platform through the National instruments USB-6020 data acquisition card. Experimental data are processed and exported as excel files.

The experimental results refer to the manipulation motion described in Section 2. Along these movements, the tool describes a closed path around an obstacle, without presenting collision. Videos of the operation of CaPaMan have been made and verified at slow speed to validate the operation of CaPaMan moving

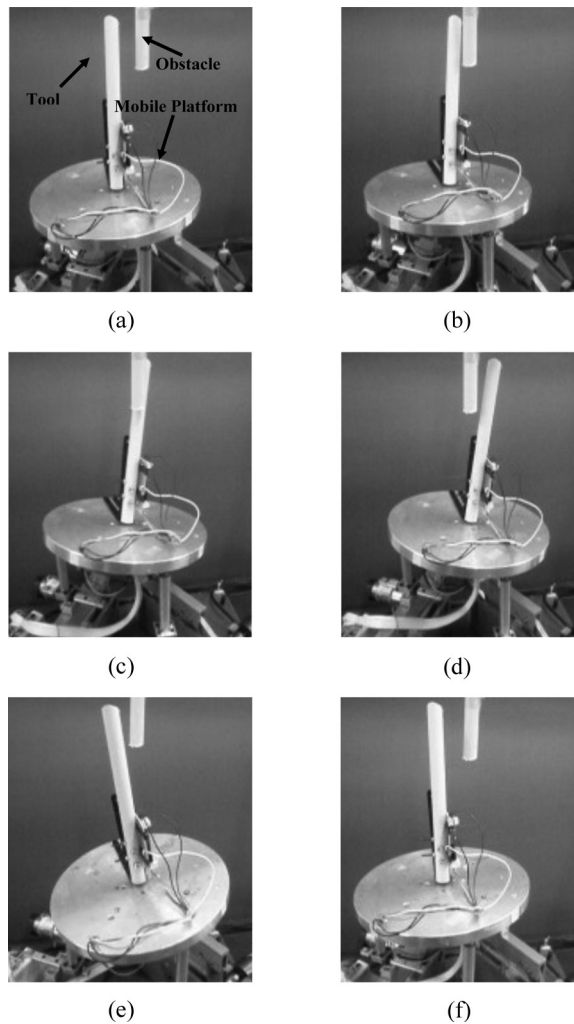


Figure 7 Snapshots of CaPaMan operation during a path to avoid a cylindrical obstacle: (a) first configuration; (b) second configuration; (c) third configuration; (d) fourth configuration; (e) fifth configuration; and (f) sixth configuration.

the tool along the desired path. In particular, Fig. 7 shows a photo sequence of CaPaMan during this operation with the relative positions of the movable plate of CaPaMan, the tool, and a cylindrical obstacle. Six significant configurations have been identified (Fig. 7) to show how the tool can move smoothly from the initial to the final configuration without having the tool (or the CaPaMan itself) too close to the obstacle. Any collision with the obstacle is also avoided as planned. Several similar tests have been carried to validate the effectiveness and engineering feasibility of the proposed path planning strategy.

Experimental data have been acquired from the accelerometers on the mobile platform of CaPaMan during the motion shown in Fig. 7. Figures 8 and 9 show the

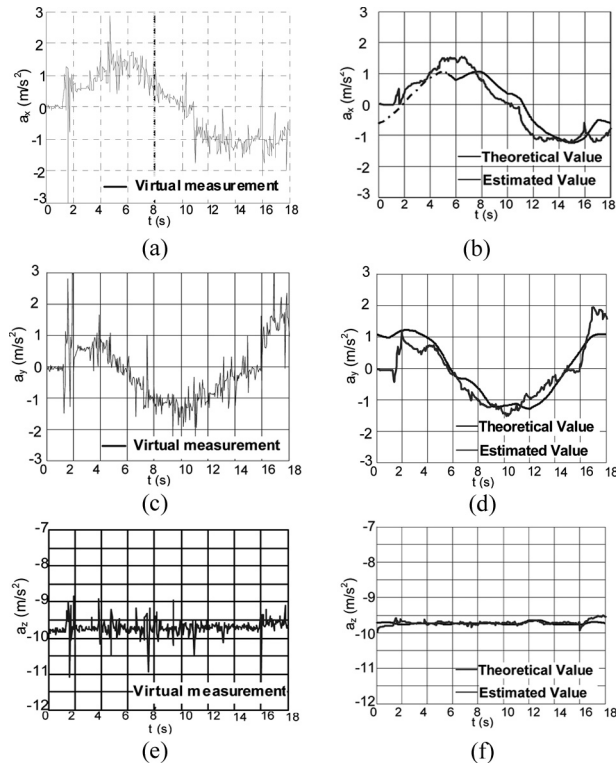


Figure 8 Acceleration components versus time for performing the planned path in 18sec: (a) measured acceleration along x -axis data before applying KF; (b) comparison of theoretical and measured acceleration along x -axis data after applying KF; (c) measured acceleration along y -axis data after applying KF; (d) comparison of theoretical and measured acceleration along y -axis data after applying KF; (e) measured acceleration along z -axis data before applying KF; and (f) comparison of theoretical and measured acceleration along z -axis data after applying KF.

plots of the data that have been acquired versus time. In particular, Fig. 8 presents the components of the acceleration of the centre of the movable plate when the path was performed in 18sec. In Fig. 8(a), (c), and (e), the non-filtered values of the acceleration components, calculated from Eq. (12) are represented. Figures 8(b), (d), and (f) show the plots of the values that have been estimated with the KF in comparison with the theoretical data. In Fig. 9(a), (c), and (e), the nonfiltered values of the angular velocity components, calculated from Eq. (12) are represented. In Fig. 9(b), (d), and (f), the values estimated with the KF versus the theoretical values are illustrated. The comparison of theoretical and experimental results show very good match of shapes and values of both accelerations and angular velocities. These results demonstrate feasibility and usefulness of the proposed procedure for a systematic experimental validation of the CaPaMan operation. The proposed procedure for a systematic experimental validation can be easily extended to any parallel manipulator provided that suitable experimental data can be made available by means of a proper sensor set-up.

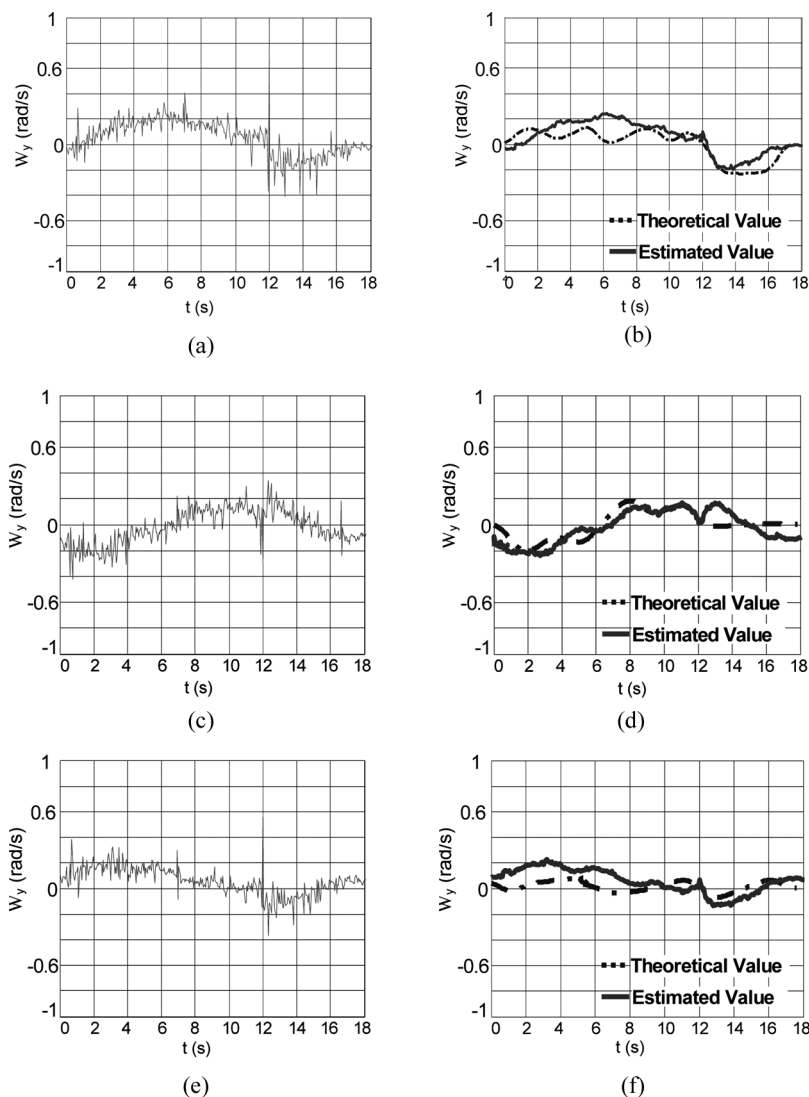


Figure 9 Angular velocities components versus time for performing the planned path in 18sec: (a) measured angular velocity along x-axis data before applying KF; (b) comparison of theoretical and measured angular velocity along x-axis data after applying KF; (c) measured angular velocity along y-axis data before applying KF; (d) comparison of theoretical and measured angular velocity along y-axis data after applying KF; (e) measured angular velocity along z-axis data before applying KF; and (f) comparison of theoretical and measured angular velocity along z-axis data after applying KF.

6. CONCLUSIONS

A planning algorithm has been implemented to compute optimal collision-free trajectories for the parallel manipulator CaPaMan. A suitable systematic validation approach has been proposed as based on a suitable test-bed and proper processing of experimental data. Experimental tests have been carried out

by operating CaPaMan at LARM in Cassino. Experimental results have been processed as based on the proposed validation procedure. Results show that suitable collision-free trajectories have been generated for CaPaMan. The experimentally measured trajectories very well match the trajectories that have been theoretically computed at path planning stage in terms of positions, velocities, and accelerations. Additionally, the proposed validation approach has been found suitable for systematically validating the operation of CaPaMan parallel manipulator. Results can be conveniently extended to any parallel manipulator provided that suitable experimental data can be made available by means of a proper sensor set-up.

ACKNOWLEDGMENTS

O. Selvi author wishes to thank Izmir Institute of Technology and Dr. Rasim Alizade for supporting of six months stay at LARM in 2010 within Erasmus-Socrates Program.

REFERENCES

- Barraquand, J., Latombe, J. C. (1991). Robot motion planning: a distributed representation approach. *International Journal of Robotics Research* 10(6):628–649.
- Berenson, D., Srinivasa, S. S., Ferguson, D., Kuffner, J. J. (2009). Manipulation planning on constraint manifolds. *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 625–632.
- Bohlin, R., Kavraki, L. (2000). Path-planning using lazy PRM. *Proceedings of IEEE International Conference on Robotics and Automation*. San Francisco, pp. 521–528.
- Brancati, R., Rossi, C., Timpone, F. (2007). Dynamic behavior and motion planning of a robot arm with non-rigid transmission. *Mechanics Based Design of Structures and Machines* 35(4):347–362.
- Bozovic, B. (2008). *Medical Robotics*. Vienna: I-Tech Education and Publishing.
- Carbone, G., Ceccarelli, M., Oliveira, P. J., Saramago, S. F. P., Carvalho, J. C. M. (2008). An optimum path planning of CaPaMan (cassino parallel manipulator) by using inverse dynamics. *Robotica* 26(2):229–239.
- Carvalho, J. C. M., Ceccarelli, M. (1999). Seismic motion simulation based on cassino parallel manipulator. *XVth Brazilian Congress on Mechanical Engineering COBEM'99*. Campinas: CD Proceedings.
- Castelli, G., Ottaviano, E., Ceccarelli, M. (2008). A fairly general algorithm to evaluate workspace characteristics of serial and parallel manipulators. *Mechanics Based Design of Structures and Machines* 36(1):14–33.
- Ceccarelli, M. (1997). A new 3 D.O.F. parallel spatial mechanism. *IFTOMM Journal Mechanism and Machine Theory* 32(8):895–902.
- Ceccarelli, M. (2004). *Fundamentals of Mechanics of Robotic Manipulation*. The Netherland: Kluwer Academic Publishers.
- Ceccarelli, M. (2010). Parallel manipulator architectures from CAPAMAN design. *Proceedings of the RAAD 2010, 19th International Workshop on Robotics in Alpe-Adria-Danube Region RAAD*. Budapest: Paper 55, doi:10.1109/RAAD.2010.5524588.

- Ceccarelli, M., Valero, F., Mata, F., Cuadrado, I. (1996). Generation of adjacent configurations for a collision-free path-planning of manipulators. *Robotica* 14(4):391–396.
- Da Graça, M., Tenreiro, J. A., Azevedo, T. (2010). An evolutionary approach for the motion planning of redundant and hyper-redundant manipulators. *Nonlinear Dynamic* 60(1):115–129.
- Dasgupta, B., Mruthyunjaya, S. (2000). Singularity-free path-planning for the Stewart platform manipulator. *Mechanism and Machine Theory* 33(6):711–725.
- Dash, A. K., I-Meng, C., Yeo, H., Yang, G. (2005). Workspace generation and planning singularity-free path for parallel manipulators. *Mechanism and Machine Theory* 40(7):776–805.
- Fang, G., Dissanayake, M. W. M. G. (1998). A neural network-based method for time-optimal trajectory planning. *Robotica* 16(2):143–158.
- Gomez-Bravo, F., Carbone, G., Lopez, D., Fortes, J. C. (2009). Planificación de trayectorias libres de colisión en manipuladores híbridos. [Collision free trajectory planning for hybridmanipulators.] In Spanish. *IX Congreso Iberoamericano de Ingeniería Mecánica*. Las Palmas de Gran Canaria, Spain, pp. 1065–1072.
- Gomez-Bravo, F., Vale, A., Ribeiro, M.I. (2007). Navigation strategies for cooperative localization based on a particle-filter approach. *Integrated Computer-Aided Engineering* 14(3):263–279.
- Gordon, N. J., Salmond, D. J., Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian bayesian state estimation. *Radar and Signal Processing IEE Proceedings-F* 140(2):107–113.
- Grewal, M. S., Andrews, A. P. (2002). Frontmatter and Index. *Kalman Filtering: Theory and Practice*. 2nd ed. New York: John Wiley & Sons, Inc.
- Gu, H., Ceccarelli, M. (2012). A multiobjective optimal path planning for a 1-DOF clutched ARM. *Mechanics Based Design of Structures and Machines* 40(1):109–121.
- Hernandez-Martinez, E. E., Conghui, L., Carbone, G., Ceccarelli, M., Lopez-Cajun, C. S. (2010). Experimental and numerical characterization of CaPaMan 2bis operation. *Journal of Applied Research and Technology* 8(1):101–119.
- Itu, T., Pisla, D. (2009). On the kinematics and dynamics of 3-DOF parallel robots with triangle platform. *Journal of Vibroengineering* 11(1):188–200.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering* 82:35–45.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research* 5(1):90–98.
- Lahouar, S., Zegloul, S., Romdhane, L. (2006). Real-time path-planning for multi-DoF robot manipulators in dynamic environment. *International Journal of Advanced Robotic Systems* 3(2):125–132.
- LaValle, S. M., Kuffner, J. J. (1999). Randomized kinodynamic planning. *Proceedings IEEE International Conference on Robotics and Automation*. Michigan: Detroit, pp. 473–479.
- Liang, C., Hernández-Martínez, E. E., Carbone, G., Ceccarelli, M. (2009). A comparison of simulations and experimental tests on operation performance of CapaMan2 bis. *IEEE International Conference on Mechatronics and Automation*, China: Changchun, p. 640.

- Lin, C. S., Chang, P. R., Luh, J. Y. S. (1983). Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Transactions on Automatic Control* 28(12):1066–1073.
- Liua, C. H., Maa, C. M. (2010). Graphical analysis for locating direct singular positions of the parallel manipulator MIPS. *Mechanics Based Design of Structures and Machines* 38(4):453–467.
- Lueth, T., Bier, J. (1999). Robot assisted intervention in surgery. In: Gilsbach, J. M., Stiel, H. S., eds. *Neuronavigation-Neurosurgical and Computer Scientific Aspects*. Wien: Springer-Verlag.
- Maybeck, P. S. (1979). *Stochastic Models, Estimation, and Control*. New York: Academic Press, Inc.
- Merlet, J. P. (2000). *Parallel Robots*. The Netherland: Kluwer Academic Publishers.
- Merlet, J. P. (2001). A generic trajectory verifier for the motion planning of parallel robots. *ASME Journal of Mechanical Design* 123(4):510–515.
- Ong, C. J., Gilbert, E. G. (1998). Robot path planning with penetration growth distance. *Journal of Robotic Systems* 15(2):57–74.
- Peters, C., Buhmann, A., Maurath, D., Manoli, Y. (2005). A novel full accelerometer based inertial navigation system. *Proceedings of Mikrosystemtechnik Kongress*, Freiburg, Germany, pp. 233–236.
- Pisla, D., Gherman, B., Vaida, C., Plitea, N. (2012). Kinematic modeling of a 5 DOF parallel hybrid robot designed for laparoscopic surgery. *Robotica*, doi: 10.1017/S0263574711001299.
- Ribeiro, M. I. (2004). *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*. Available in <http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf>.
- Rosen, J., Hannaford, B., Satava, R. M., eds. (2011). *Surgical Robotics, Systems Applications and Visions*. 1st ed. New York: Spinger.
- Saramago, S. F. P., Ceccarelli, M. (2004). Effect of some numerical parameters on a path planning of robots taking into account actuating energy. *Mechanism and Machine Theory* 39(3):247–270.
- Saramago, S. F. P., Valder, S. J. (2001). Trajectory modeling of robots manipulators in the presence of obstacles. *Journal of Optimization Theory and Applications* 110(1):17–34.
- Savavanan, R., Ramabalan, S., Balamurugan, C. (2008). Evolutionary collision free optimal trajectory planning for intelligent robots. *The International Journal of Advanced Manufacturing Technology* 36(11):1234–1251.
- Schiele, B., Crowley, J. L., (1994). A comparison of position estimation techniques using occupancy grids. *Proceedings of IEEE Conference on Robotics and Automation*. San Diego, CA, 2:628–1634.
- Schopp, P., Klingbeil, L., Peters, C., Buhmann, A., Manoli, Y. (2009). Sensor fusion algorithm and calibration for a gyroscope free IMU. *Proc. Chem. Proc. Eurosensors XXIII Conference*. Lausanne, Switzerland, pp. 1323–1326.
- Selvi, O., Ceccarelli, M. (2010). An experimental evaluation of earthquake effects on mechanism operation. *Proceedings of the International Symposium of Mechanism and Machine Science*, Izmir, Turkey, pp. 408–416.
- Sen, S., Dasgupta, B., Mallik, K. (2003). Variational approach for singularity-free path-planning of parallel manipulators. *Mechanism and Machine Theory* 38(11):1165–1183.

- Shin, K. G., McKay, N. D. (1986). A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Transactions on Automatic Control* 31(6):491–500.
- Simeon, T., Laumond, J. P., Nissoux, C. (2000). Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics* 14(6):477–494.
- Yan, C., Gao, F., Zhang, Y. (2010). Kinematic modeling of a serial–parallel forging manipulator with application to heavy-duty manipulations. *Mechanics Based Design of Structures and Machines* 38(1):105–129.
- Zappa, B., Legnana, G., Van der Bogert, A. J., Adamini, R. (2001). On the number and placement of accelerometers for angular velocity and acceleration determination. *ASME Journal of Dynamics Systems Measurement and Control* 123(3):552–554.