Trajectory-tracking Control of Semi-Regular Stewart Platform Manipulator

Sadiq Mohamed Anwar, Sandipan Bandyopadhyay

Abstract

Trajectory-tracking control of parallel manipulators is more difficult than in the case of serial ones due to the presence of the loop-closure constraints and constraint forces resulting from them. One needs to eliminate these forces to get to the equation of motion, and then apply a control scheme.

In this paper, such a control scheme is presented through application on a semi-regular Stewart platform manipulator (SRSPM). The manipulator has six degrees-of-freedom; however, it is modelled by a system of 18 coupled nonlinear ordinary differential equations (ODE) using the constrained Lagrangian formulation. The model is then linearised through feedback, and controlled by a linear PD servo scheme. Numerical simulations over a non-singular path show that the scheme is fairly accurate, at the cost of being computationally expensive. The scheme is general in nature, and as such, it is expected to work in the case of other parallel manipulators as well.

Keywords: Stewart platform, Computed torque control, Feedback linearisation, Control-law partitioning, Parallel manipulator, Lagrangian dynamics, Trajectory-tracking

1 Introduction

The task of trajectory-tracking in parallel manipulators is generally more complicated in parallel manipulators than in their serial counterparts. One reason for this is the geometric complexity of their singularity manifold, which makes it difficult to find a non-singular path connecting two arbitrary configurations in the workspace. Even when such a path is obtained, accurate tracking requires the understanding of the dynamic model of the manipulator, which in turn involves actuated, as well as passive links. In order to complete the dynamic model, one needs to find the motion of the passive links in addition to the actuated ones. While this can be done in principle using the forward kinematic analysis, in manipulators such as the Stewart platform (SPM), forward kinematics is a hard problem to solve.

However, very structure of a parallel manipulator provides an easier alternative to the above method. Since each leg of a parallel manipulator connects the moving platform to the fixed base in an independent manner, the inverse kinematic problem is naturally *decoupled*. If the end-effector's (i.e., the moving platform's) pose is known

Sadiq Mohamed Anwar
Department of Engineering Design, Indian Institute of Technology Madras, Chennai-600036, Tamil Nadu, India, E-mail:sadm.ne@gmail.com.

Sandipan Bandyopadhyay
Department of Engineering Design, Indian Institute of Technology Madras, Chennai-600036, Tamil Nadu, India, E-mail:sandipan@iitm.ac.in.

at an instant of time, then the active as well as the passive link positions can be found out easily from inverse kinematics (see, e.g., [1]). This allows for the formulation of the equation of motion of the manipulator as a *constrained Lagrangian system*, subject to the *loop-closure* constraints. Though the model is in higher dimensions than the degree-of-freedom (DoF) (due to the presence of passive links), it affords a dynamic model that can be included in the trajectory-tracking scheme.

Such a scheme is adopted in this paper to apply a trajectory-tracking control to an SRSPM. A *computed torque scheme* is adopted in which inertia, Coriolis forces are represented *exactly*. This is in contrast with a related work [2], where an approximation to such terms have been used. In [3], an optimal control scheme has been developed in the *task space*, where the forward kinematics is approximated by polynomials using a predicted square error cost function. In [4], authors have discussed an extended torque control scheme where sensors are introduced in passive joints and the redundant information is used to simplify the model-based scheme. However, that would arise in a physical implementation of the scheme have not been discussed as the model is verified only by a simulation. In the present work, the trajectory chosen is verified to be non-singular following the method described in been described in [5]. Numerical simulations performed show that the trajectory can be tracked to an accuracy of $10^{-5}m$. However, it requires high temporal resolution, and hence it is computationally expensive.

The rest of the paper is organised as follows: Section 2 describes the development of the kinematic model of the SRSPM. In Section 3, the dynamic model is presented, which forms the back-bone of the model-based control scheme presented in Section 4. The simulation results are discussed in Section 5, followed by the conclusions in Section 6.

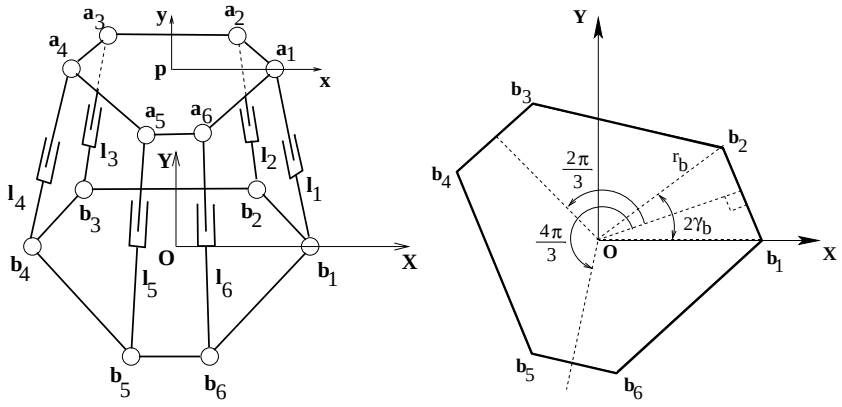
2 Kinematic Model of the SRSPM

The geometry of the SRSPM manipulator studied in this paper (shown in Fig. (1a)) is identical to the one presented in [6].

2.1 Geometry and loop-closure

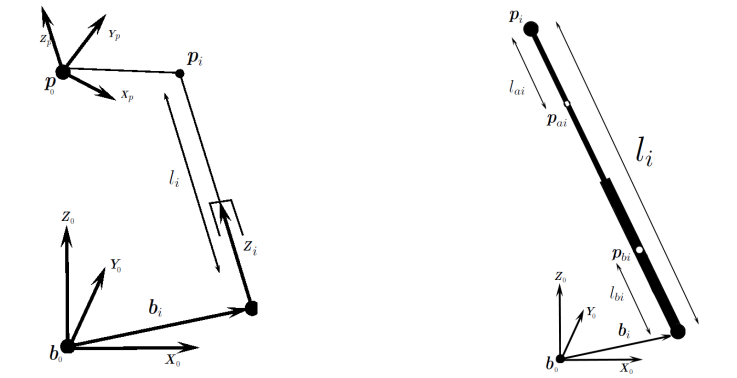
The top and the bottom platforms are semi-regular hexagons contained within circles of radii $2r_b$ and $2r_t$ respectively, with the angular separation between the leg-connection points as $2\gamma_b$ and $2\gamma_t$ respectively. The legs are UPS in structure, with the P-joint actuated in each case. The geometry of a single leg is shown in Fig. (2). The $i\text{th}$ leg vector is found as $\boldsymbol{l}_i = (0, 0, l_i)^T$ in its *local* frame. Two successive rotations, given by $\boldsymbol{R}_y(\phi_i)\boldsymbol{R}_x(\psi_i)$ account for the U-joint at the base of each leg. Thus, the *configuration space* of the manipulator is given by $\boldsymbol{q} = (\boldsymbol{l}^T, \boldsymbol{\phi}^T)^T$, where $\boldsymbol{l} = (l_1, \ldots, l_6)^T$ and $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_6, \psi_1, \ldots, \psi_6)^T$. The loop-closure constraints in global frame are:

$$\begin{aligned} &\boldsymbol{p}_i = \boldsymbol{b}_i + \boldsymbol{l}_i \\ \Rightarrow &\boldsymbol{p} + \boldsymbol{R}\boldsymbol{a}_i - \boldsymbol{b}_i - \boldsymbol{R}_y(\phi_i)\boldsymbol{R}_x(\psi_i)(0,0,1)^T = \boldsymbol{0},\ i = 1, \ldots, 6 \end{aligned} \tag{1}$$



(a) Geometry of the SRSPM (b) Geometry of the top platform

Figure 1: Schematic of the SRSPM (adopted from [6])



(a) The ith leg in the global frame (b) Geometric details of a single leg

Figure 2: Geometry of a leg of the SRSPM

where $\boldsymbol{p} = (x, y, z)^T$ and $\boldsymbol{R} \in SO(3)$ denote the position of the centre of the top platform, and the orientation of the top platform with respect to the fixed base, respectively. The top and bottom ends of the ith leg are given by $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ respectively. Further, $\boldsymbol{R}$ is parametrised in terms of the *Rodrigue's parameters*, $\boldsymbol{c} = (c_1, c_2, c_3)^T$.

2.2 Velocity kinematics

From Eq. (1), one finds:

$$\boldsymbol{p}_i = \boldsymbol{b}_i + \boldsymbol{R}_y(\phi_i)\boldsymbol{R}_x(\psi_i)(0,0,1)^T \text{ and} \tag{2}$$

$$\boldsymbol{p} = 1/6 \sum_{i=1}^{6} \boldsymbol{p}_i \tag{3}$$

Velocity of the centre of the top platform is found by time differentiation:

$$\dot{\boldsymbol{p}} = \boldsymbol{J}_{\boldsymbol{pq}}\dot{\boldsymbol{q}},\ \boldsymbol{J}_{\boldsymbol{pq}} = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} \tag{4}$$

Similarly, mass-centres of the top and the bottom parts of the ith leg are respectively given by (see Fig.(2)):

$$\boldsymbol{p}_{ai} = \boldsymbol{b}_i + \boldsymbol{R}_y(\phi_i)\boldsymbol{R}_x(\psi_i)\boldsymbol{l}_{ai},\ \boldsymbol{l}_{ai} = (0,0,l_{ai})^T \tag{5}$$

$$\boldsymbol{p}_{bi} = \boldsymbol{b}_i + \boldsymbol{R}_y(\phi_i)\boldsymbol{R}_x(\psi_i)\boldsymbol{l}_{bi},\ \boldsymbol{l}_{bi} = (0,0,l_i - l_{ai})^T \tag{6}$$

Velocities of these points are obtained in terms of the corresponding Jacobian matrices:

$$\dot{\boldsymbol{p}}_{ai} = \boldsymbol{J}_{paqi}\dot{\boldsymbol{q}},\ \boldsymbol{J}_{paqi} = \frac{\partial \boldsymbol{p}_{ai}}{\partial \boldsymbol{q}} \text{ and } \dot{\boldsymbol{p}}_{bi} = \boldsymbol{J}_{pbqi}\dot{\boldsymbol{q}},\ \boldsymbol{J}_{pbqi} = \frac{\partial \boldsymbol{p}_{bi}}{\partial \boldsymbol{q}} \tag{7}$$

Finding the angular velocities of the legs are simpler, as both parts of the prismatic joint have the same angular velocity. The rotation matrix of each leg is given by $\boldsymbol{R}_{li} = \boldsymbol{R}_y(\phi_i)\boldsymbol{R}_x(\psi_i)$, and the corresponding *body-fixed angular velocity matrix,* $\boldsymbol{\Omega}_i \in so(3)$ is given by:

$$\boldsymbol{\Omega}_i = \boldsymbol{R}_{li}^T \dot{\boldsymbol{R}}_{li} \tag{8}$$

Extracting the angular velocity vector from the 3×3 skew-symmetric matrix:

$$\boldsymbol{\omega}_{li} = \boldsymbol{J}_{\boldsymbol{\omega lq}i}\dot{\boldsymbol{q}} \tag{9}$$

Similarly, one obtains the angular velocity of the top platform from the knowledge of its orientation given by $\boldsymbol{R}$:

$$\boldsymbol{\omega} = \boldsymbol{J}_{\boldsymbol{\omega q}}\dot{\boldsymbol{q}} \tag{10}$$

Computations of these Jacobian matrices typically require the use of a Computer Algebra System (CAS) as computing so many partial derivatives by hand is tedious as well as error-prone. In this work, a commercial CAS `Mathematica` [7] has been used for all symbolic computations.

3 Dynamic Model of the Manipulator

The *constrained Lagrangian formulation* is used to develop a model of the manipulator's dynamics.

3.1 Forward dynamics

The Lagrangian of the manipulator, $\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T - V$, is computed in terms of the kinetic energy T, and the potential energy V. The kinetic energy is a sum of the kinetic energies of the individual legs, as well as that of the top platform:

$$T = \frac{1}{2}\dot{\boldsymbol{q}}^T(\boldsymbol{I}_l + \boldsymbol{I}_p)\dot{\boldsymbol{q}}, \text{ where, } \boldsymbol{I}_p = \boldsymbol{J}_{pq}^T m_p \boldsymbol{J}_{pq} + \boldsymbol{J}_{\omega q}^T \boldsymbol{I}_p \boldsymbol{J}_{\omega q} \tag{11}$$

$$\boldsymbol{I}_l = \sum_{i=1}^{6} \left(\boldsymbol{J}_{pbqi}^T m_{bi} \boldsymbol{J}_{pbqi} + \boldsymbol{J}_{\omega lqi}^T (\boldsymbol{I}_{lai} + \boldsymbol{I}_{lbi}) \boldsymbol{J}_{\omega lqi} + \boldsymbol{J}_{paqi}^T m_{ai} \boldsymbol{J}_{paqi} \right) \tag{12}$$

where m_{bi}, m_{ai}, m_p are the masses of lower and upper parts of the ith leg, and the platform, respectively. Further, $\boldsymbol{I}_{lbi}, \boldsymbol{I}_{lai}, \boldsymbol{I}_p$ are the inertia matrices of lower and upper parts of the ith legs, and the top platform in the respective body-fixed frames, respectively. For calculating the potential energy V, the base platform height is taken as the datum. The total potential energy is calculated as the sum of the potential energies of each of the moving parts: $V = \sum_{i=1}^{6}(V_{bi} + V_{ai}) + V_p$, where, $V_{bi}, V_{ai}, V_p \quad (i = 1, \ldots, 6)$ are the potential kinetic energies of the lower and upper links of the legs, and the platform, respectively.

The equations of motion for the constrained system with the *loop-closure* constraints $\boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{0}$ are derived as (see, e.g., [1]):

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = F_i + (\boldsymbol{J}_{\boldsymbol{\eta q}}\boldsymbol{\lambda})_i, \ \boldsymbol{J}_{\boldsymbol{\eta q}} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{q}}, \ i = 1, \ldots, 12 \tag{13}$$

where $\boldsymbol{J}_{\boldsymbol{\eta q}}$ is the constraint Jacobian, F_is are the externally applied leg forces, and $\boldsymbol{\lambda}$ is the vector of *Lagrange multipliers*. In a matrix form, Eq. (13) can be written as

$$\boldsymbol{M}(\boldsymbol{q}, \dot{\boldsymbol{q}})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{F} + \boldsymbol{J}_{\boldsymbol{\eta q}}^T \boldsymbol{\lambda} \tag{14}$$

Here, $\boldsymbol{M}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{I}_l + \boldsymbol{I}_p$ is the 18×18 mass matrix; $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the centripetal and Coriolis force matrix, which can be derived from $\boldsymbol{M}$ (see, e.g., [1]); $\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial V}{\partial \boldsymbol{q}}$ is the vector of gravitational loads; $\boldsymbol{F}$ is the vector of external forces; and $\boldsymbol{J}_{\boldsymbol{\eta q}}^T \boldsymbol{\lambda}$ is the vector of *constraint forces*. The Lagrange multipliers $\boldsymbol{\lambda}$ can be obtained as (see, e.g., [1]):

$$\boldsymbol{\lambda} = -\boldsymbol{A}^{-1}\left(\dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}} + \boldsymbol{J}_{\boldsymbol{\eta q}}\boldsymbol{M}^{-1}\boldsymbol{f}\right), \ \boldsymbol{A} = \boldsymbol{J}_{\boldsymbol{\eta q}}\boldsymbol{M}^{-1}\boldsymbol{J}_{\boldsymbol{\eta q}}^T, \ \boldsymbol{f} = \boldsymbol{F} - \boldsymbol{C}\dot{\boldsymbol{q}} - \boldsymbol{G} \tag{15}$$

Using Eq. (15) in Eq. (14), the *forward dynamic equation* is obtained as:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{f} - \boldsymbol{J}_{\boldsymbol{\eta q}}^T \boldsymbol{A}^{-1}\left(\dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}} + \boldsymbol{J}_{\boldsymbol{\eta q}}\boldsymbol{M}^{-1}\boldsymbol{f}\right) \tag{16}$$

3.2 Inverse dynamics

The *inverse dynamic equation*, i.e., the expression for the externally applied force for a certain motion of the manipulator is obtained by some manipulation of Eq. (16):

$$\begin{aligned}
&\boldsymbol{M}\ddot{\boldsymbol{q}} = \left(\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{\eta q}}^T \boldsymbol{A}^{-1}\boldsymbol{J}_{\boldsymbol{\eta q}}\boldsymbol{M}^{-1}\right)\boldsymbol{f} - \boldsymbol{J}_{\boldsymbol{\eta q}}^T \boldsymbol{A}^{-1}\dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}} \\
\Rightarrow &\boldsymbol{f} = \boldsymbol{B}^{\#}\left(\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{J}_{\boldsymbol{\eta q}}^T \boldsymbol{A}^{-1}\dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}}\right), \ \boldsymbol{B} = \left(\boldsymbol{I} - \boldsymbol{J}_{\boldsymbol{\eta q}}^T \boldsymbol{A}^{-1}\boldsymbol{J}_{\boldsymbol{\eta q}}\boldsymbol{M}^{-1}\right) \\
\Rightarrow &\boldsymbol{F} = \boldsymbol{B}^{\#}\left(\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{J}_{\boldsymbol{\eta q}}^T \boldsymbol{A}^{-1}\dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}}\right) + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G}
\end{aligned} \tag{17}$$

where $\boldsymbol{I} \in \mathbb{R}^{18\times18}$ is an identity matrix, and $(\cdot)^{\#}$ denotes the *pseudo-inverse* of a matrix. It is observed, that at *non-singular* points in the trajectory, the matrix $\boldsymbol{B} \in \mathbb{R}^{18\times18}$ has rank 6. This corroborates with the fact that in Eq. (17), only 6 of the entries in $\boldsymbol{BF}$ are allowed to be *free*, as the mechanism has only those many degrees-of-freedom[1]. Thus, *pseudo-inverse* needs to be used in Eq. (17) rather than inverse.

4 Trajectory-tracking Control

The trajectory is chosen as cubic functions in the coordinates (x, y, z) of the point $\boldsymbol{p}$, and the Rodrigue's parameters (c_1, c_2, c_3) describing the orientation of the top platform. It is verified *a priori* that the chosen path is free of singularities following [5].

A linear (PD) control scheme is adopted for the tracking task after *feedback-linearisation* of the manipulator's dynamics (see, e.g., [1]). The following assumptions are inherent in such schemes: (a) the model is deterministic with no parametric uncertainties, (b) there are no unmodelled dynamics, e.g., friction, and (c) full state feedback is available without delay and noise. The required control effort $\boldsymbol{F}$ is got from Eq. (17):

$$\boldsymbol{BF} = \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{J}_{\boldsymbol{\eta q}}^{T}\boldsymbol{A}^{-1}\dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}} + \boldsymbol{B}(\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G}) \tag{18}$$

Assume that $\boldsymbol{F}$ partitions such that: $\boldsymbol{BF} = \boldsymbol{\alpha F}' + \boldsymbol{\beta}$, and choose:

$$\boldsymbol{\alpha} = \boldsymbol{M},\ \boldsymbol{\beta} = \boldsymbol{J}_{\boldsymbol{\eta q}}^{T}\boldsymbol{A}^{-1}\dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}} + \boldsymbol{B}(\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{G}) \tag{19}$$

This reduces the manipulator's dynamics to a *linear, decoupled, unit inertia system:*

$$\boldsymbol{F}' = \ddot{\boldsymbol{q}} \tag{20}$$

A PD control scheme is proposed for the linear system:

$$\boldsymbol{F}' = \ddot{\boldsymbol{q}}_d(t) + \boldsymbol{K}_v\dot{\boldsymbol{e}}(t) + \boldsymbol{K}_p\boldsymbol{e}(t),\ \boldsymbol{e}(t) = \boldsymbol{q}_d(t) - \boldsymbol{q}(t) \tag{21}$$

where $\boldsymbol{q}_d(t)$ denotes that *desired* trajectory, and $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ denote the velocity and position gain matrices, respectively. Application of the control input in the unit inertia system results in a second order linear ODE in the error $\boldsymbol{e}(t)$:

$$\ddot{\boldsymbol{e}}(t) + \boldsymbol{K}_v\dot{\boldsymbol{e}}(t) + \boldsymbol{K}_p\boldsymbol{e}(t) = \boldsymbol{0} \tag{22}$$

The gains are chosen so that the errors die out in a *critically damped* manner:

$$\boldsymbol{K}_p = k_p\boldsymbol{I},\ \ \boldsymbol{K}_v = 2\sqrt{k_p}\boldsymbol{I},\ k_p > 0 \tag{23}$$

[1]To study this further, it has been tested, separately, that at a singularity where $\mathrm{rank}\,(\boldsymbol{J}_{\boldsymbol{\eta q}})$ goes down by one, $\mathrm{rank}\,(\boldsymbol{B})$ goes up by the same number.

5 Simulation Results and Discussions

The mass and inertia properties for the following simulations were adopted from [8]. The initial pose is chosen as $(x_i, y_i, z_i) = (0.3520, 0.7040, 1.7040)$, $(c_{1i}, c_{2i}, c_{3i}) = (0.2056, 0.3408, 0.5760)$, while the final pose is given by $(x_f, y_f, z_f) = (0.3520, 0.7040, 1.7040)$, $(c_{1f}, c_{2f}, c_{3f}) = (0.4, 0.6, 0.9)$ (see Fig. (3)). Two different simula-

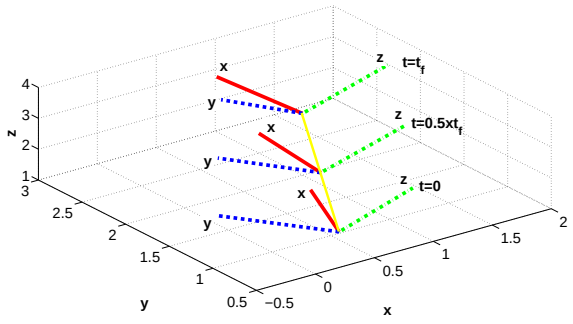


Figure 3: Trajectory of the end-effector frame

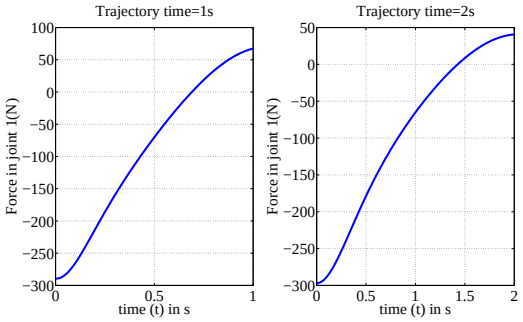
tions have been performed: for *fast* motion where the path is covered in $t_f = 1$ second, *slow* motion where the path is covered in $t_f = 2$ seconds. The value of k_p is chosen by numerical trials, and fixed at 2000.

Fig. (4) shows sample results from the simulations conducted at different speeds of the top platform. As expected, the tracking errors as well as the actuator efforts are higher for the fast motion than the slow. The plots show that the errors are brought down faster when in slow motion than the fast. This is due to the fact that sampled values of the non-linear portions of the controller (inside the numerical ODE integrator) is closer to the real values in slow motion (assuming similar time steps in both the simulations).

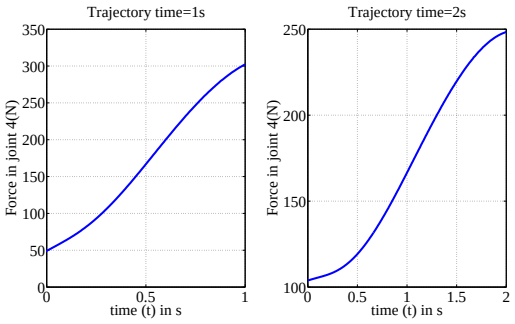
The CPU time consumed is approximately 10 minutes for the fast motion case and 20 minutes for the slow. There are two reasons for this: (a) the expressions for $\boldsymbol{M}$ and $\boldsymbol{C}$ are huge in size (i.e., 5.8 MB and 463 MB of `Mathematica` expressions, respectively); and (b) very tight error tolerances were set up for the ODE solver, so as to obtain very accurate results (i.e., tracking error of the order of $10^{-5}m$, compared to errors of 10% in [2] and 0.68% in [3]), purely for the sake of demonstration. The computational speed can be enhanced by performing symbolic simplifications on the matrices $\boldsymbol{M}$, $\boldsymbol{C}$ and by increasing the numerical tolerances to realistic levels.

6 Conclusions

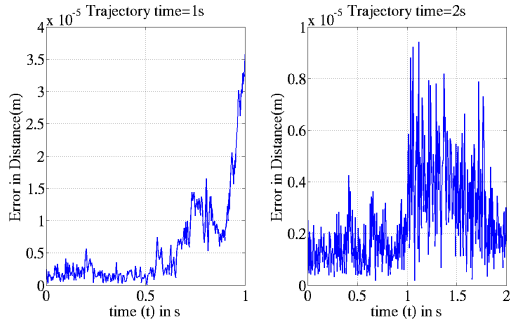
A study on the trajectory-tracking control of SRSPM has been presented. It has been shown that the feedback linearisation scheme with a simple PD servo part works effectively, yielding tracking errors in the range of $10^{-5}m$. It has been observed that the



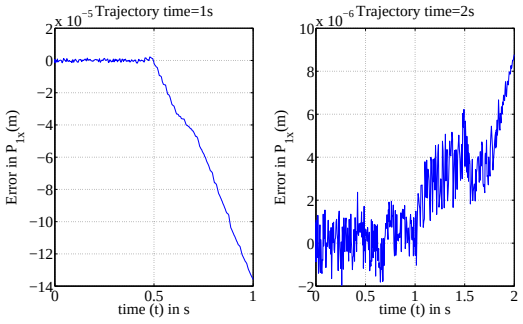
(a) Force in the first leg



(b) Force in the fourth leg



(c) Tracking error (Cartesian distance)



(d) Error in the x coordinate of point $\boldsymbol{p}_1$

Figure 4: Results of simulation for fast and slow motions

formulation involves the inversion of a 18×18 matrix with rank 6. However, using the pseudo-inverse allows the computation to proceed in a stable manner.

The control scheme is applicable to *any* parallel manipulator or closed-loop mechanisms in general. However, the computational speed is an issue at this point, which can be improved in the future using symbolic simplifications.

References

[1] A. Ghosal, *ROBOTICS: Fundamental Concepts and Analysis*. Oxford University Press, 2008.

[2] S.-H. Lee, J.-B. Song, W.-C. Choi, and D. Hong, "Position control of a Stewart Platform using inverse dynamics control with approximate dynamics," *Mechatronics*, vol. 13, no. 6, pp. 605 – 619, 2003.

[3] A. Omran and A. Kassem, "Optimal task space control design of a stewart manipulator for aircraft stall recovery," *Aerospace Science and Technology*, vol. In Press, Corrected Proof, pp. –, 2010.

[4] A. Zubizarreta, M. Marcos, I. Cabanes, and C. Pinto, "A procedure to evaluate extended computed torque control configurations in the stewartgough platform," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 770 – 781, 2011.

[5] S. Bandyopadhyay, "Verification of the trajectories of Stewart Platform Manipulators against singularities," *14th National Conference on Machines and Mechanisms (NaCoMM - 09)*, 2009.

[6] S. Bandyopadhyay and A. Ghosal, "Geometric characterization and parametric representation of the singularity manifold of a 6-6 Stewart Platform Manipulator," *Mechanism and Machine Theory*, vol. 41, no. 11, pp. 1377 – 1400, 2006.

[7] *Mathematica*. Champaign, Illinois: Wolfram Research, Inc, version 7.0 ed., 2008.

[8] A. M. Lopes, "Dynamic modeling of a Stewart Platform using the generalized momentum approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 8, pp. 3389 – 3401, 2009.