Comparision of Joint Space and Task Space Integral Sliding Mode Controller Implementations for a 6DOF Parallel Robot

Dereje Shiferaw
Indian Institute of Technology Roorkee
Department of Electronics and Computer Engineering
line 3: Roorkee, India
e-mail: derejde@ernet.in.ac

Anamika Jain (Author)
Graphic Era University
Department of Electronics and Communication
Dehradun, India
e-mail: anamikajain@gmail.com

Abstract—Controllers for parallel manipulators can be designed in task space or joint space, each having its own advantages and disadvantages. In this paper, integral sliding mode controllers designed in joint space and in task space are compared using MATLAB simulation. In both cases, genetic algorithm is used to determine optimal sliding surface gain. The performance and robustness, smoothness of control signal and ease of implementation of the controllers is compared using a smooth trajectory. The performance of the controllers is measured by the mean square value of the tracking error over the whole trajectory while robustness is compared by the percentage error increase in mean square error between no load and full load conditions. Extensive simulations showed that joint space implementation gives slightly bigger mean square error value but needs a small control effort. The control signal is also very smooth in case of joint space.

Keywords—Integral sliding mode control, parallel robot, robust control.

I. INTRODUCTION

Parallel kinematics manipulators have high structural rigidity and stiffness which makes them much more preferable than serial robots for precision applications such as machining, robotic surgery, pointing and so on [1][2][3]. However, unstructured uncertainties resulting from payload variation and structured uncertainties due to actuator friction, backlash and unmodeled dynamics together with their highly coupled dynamics makes them difficult for controller design. One of the promising control methods, which are able to compensate the uncertainties, is the sliding mode control approach [4]. Because of this many authors have proposed control of the Gough-Stewart platform using SMC [2] [3] [5] [6]. But in practical applications, conventional sliding mode controllers have certain drawbacks including, chattering, lack of robustness in reaching phase and lack of robustness to unmatched uncertainties [7][8].

To solve these problems, integral sliding mode control (ISMC) has been proposed [8]. ISMC is an improvement to conventional SMC and uses a nonlinear sliding surface having an integral term [8][9][10]. The integral sliding surface is designed to constrain the system states to be on sliding mode from initial time and hence it totally removes the reaching phase problem. Moreover, using integral sliding surface improve the stability of sliding dynamics and hence it enables to enhance robustness against unmatched uncertainties [8][9]. Therefore using integral sliding mode controller for Stewart platform manipulator will help to enhance the performance of the manipulator if it is designed properly.

In designing a controller for Stewart platform manipulator, there are two possible approaches: task space or joint space [11]. In joint space control, individual legs of a parallel manipulator are considered as independent systems and the coupling effect from other legs is considered as a disturbance, hence the system is single input single output (SISO) [11]. This SISO implementation is less costly and can be implemented in parallel easily. However, it results in synchronization error [12] and hence cannot achieve high performance, especially when the manipulator is moving at high speed. On the other hand, task space control results in multiple input multiple output (MIMO) system and has the potential to compensate coupling errors. Furthermore it can achieve superior 6 DOF tracking performance [11]. In this paper, we will compare the performance of an integral sliding mode controller implemented in these two approaches. One of the most important design steps in integral sliding mode control is the optimal selection of the nonlinear sliding surface gain. In [8][9][10] design methods have been proposed for systems with matched and unmatched uncertainty. However, in the present case, we use genetic algorithm to optimally select the gains of the integral sliding surface.

The paper is organized as follows: section two gives a summary of the idea of integral sliding mode and problem statement. Section three and four discuss the kinematic and dynamic modeling of Stewart platform manipulator. Section five gives the design of the sliding surface and controller for the two cases separately. Section six contains the simulation result and discussion and then conclusion follows.

II. PROBLEM STATEMENT

Consider an uncertain nonlinear system given as

\[ \dot{x} = f(x, t) + g(x, t)u + d(t) \]  

where \( x \) is \( n \times 1 \) dimensional state vector, \( f(x, t) \) and \( g(x, t) \) are \( n \times 1 \) and \( n \times m \) dimensional vector and matrix valued smooth nonlinear functions, \( d \) is \( n \times 1 \) dimensional vector of...
The uncertainties and \( u \) is \( m \times 1 \) dimensional vector of control inputs. The integral sliding surface \( s \) and the control signal \( u \) are given by

\[
s(x,t) = C \left[ x - x(0) - \int_{t_0}^t (f(x,\tau) + g(x,\tau)u_0) \, d\tau \right]
\]

\[u = u_0 + u_1\]  \hspace{1cm} (2)

Where \( x(0) \) is the initial value of the states \( u_0 \) is the nominal control signal and \( u_1 \) is a discontinuous control signal given by

\[u_1 = -Kf_s(s)\]  \hspace{1cm} (3)

\( K \) is the gain and \( f_s(s) \) is switching function

Then the problem in integral sliding mode control is finding a control signal \( u \) (3), and matrix \( C \) such that the sliding surface given by (2) and its derivative remain zero for all time \( t>0 \).

The mathematical formulation given in (2) can be seen as a minimization problem where the sliding variable \( s \), which is the mismatch between the nominal and perturbed system, is minimized. And hence we employ genetic algorithm for the design of \( C \) and \( u \) is designed using the equivalent control method. In block diagram, it can be shown as in Fig. 1.

**III. MODELING OF THE MANIPULATOR**

**A. Kinematic and geometric modeling**

For geometric and kinematic modeling, the following conventions are used. The centers of the universal and spherical joints are denoted by \( B_i \) \((i = 1, 2 \ldots 6)\) and \( P_i \) \((i = 1, 2 \ldots 6)\) respectively. Reference frames \( F_b \) and \( F_p \) are attached to the base and the platform as shown in Fig. 2. The position vector of the center of universal joints \( B_i \) in frame \( F_b \) is \( b_i \) and the position vector of the center of spherical joints \( P_i \) in frame \( F_p \) is \( p_i \). Let \( r = [r_x, r_y, r_z] \) be the position of the origin \( O_b \) with respect to \( O_b \) and also let \( R \) denote the orientation of frame \( F_p \) with respect to \( F_b \). Thus the Cartesian space position and orientation of the moveable platform or end effector is specified by \( X = [r_x, r_y, r_z, \alpha, \beta, \gamma] \) where the three angles \( \alpha, \beta, \gamma \) are three rotation angles that constitute the transformation matrix \( R \).

Then length of leg \( i \) is the magnitude of the vector \( B_iP_i \) which is given by

\[
|B_iP_i| = \|Rb_i + r - b_i\| = q_i.
\]

This is the inverse geometric formula that gives the length of each leg for a given desired position and orientation of the end effector. The direct geometric model which gives the position \( r = [r_x, r_y, r_z] \) and orientation angles \( \alpha, \beta, \gamma \) for a given measured value of \( q_i \), \( i = 1, 2 \ldots 6 \) is nonlinear and is solved using numerical methods.

The inverse kinematic model gives the velocity of the active joint \( \dot{q} \) for a given end effector linear and angular velocity and is given as

\[
\dot{X} = J^{-1}\dot{q}
\]

Where \( J \) is the Jacobean matrix of the platform with respect to the base frame [11]

**B. Dynamic modeling**

The dynamic modeling of Stewart platform manipulator has been extensively studied by many researchers. The methods used are Lagrangian, Newton Euler and principle of virtual work [13]. Using Lagrangian method, the actuator torque \( \tau \) is given in task space as

\[
M(X)\ddot{X} + V(X,\dot{X})\dot{X} + G(X) = J^{-T}\tau
\]

Where \( X = [r_x, r_y, r_z, \alpha, \beta, \gamma] \) is the task space position and orientation of center of movable platform, \( M(X) \) is the inertia matrix, \( V(X,\dot{X}) \) is the coriolis/centrifugal force coefficient matrix and \( G(X) \) is the gravitational torque. In the above dynamic model, actuator dynamics and friction have been neglected. The system will have uncertainties because of inertia loading, unmodelled dynamics and
friction from actuators. The uncertainties are assumed to have bounds and each term can be expressed as nominal and deviation as in (8) below.

\[
M = M_N + \Delta M \\
V = V_N + \Delta V \\
G = G_N + \Delta G
\]  

(8)

The perturbations \( \Delta M, \Delta C \) and \( \Delta G \) are assumed to have the following bounds

\[
\|\Delta M\| \leq M_m \\
\|\Delta V\| \leq V_m \\
\|\Delta G\| \leq G_m
\]

(9)

Using (8) and (9), (7) can be rewritten in state space form as

\[
\dot{X}_1 = X_2 \\
X_2 = M^{-1}\left( J^{-T}\tau - V(X_1, X_2) X_2 - G(X_1)\right) + d
\]

(10)

Where \( X_1 \) is (6X1) state vector of Cartesian space positions and orientations, \( X_2 \) is (6X1) state vector of the Cartesian space velocities and \( d \) is the lumped uncertainty term given by

\[
d = M^{-1}\left( -\Delta M \dot{X} - \Delta V \dot{X} - \Delta G \right)
\]

(11)

Comparing (10) and (1),

\[
f = \begin{bmatrix} X_2 \\ -M^{-1}(V(X_1, X_2) X_2 + G(X_1)) \end{bmatrix}
\]

(12)

and

\[
g = \begin{bmatrix} 0 \\ M^{-1}J^{-T} \end{bmatrix}
\]

(13)

The following assumptions are taken.

**Assumption 1:** The inertia matrix \( M \) is invertible

**Assumption 2:** The mechanical system is designed so that the Jacobean matrix is nonsingular in the whole workspace

**Assumption 3:** The uncertainties in the inertia, coriolis and centrifugal and gravitational matrixes are bounded as in (9).

IV. DESIGN OF CONTROLLERS

A. Joint space integral sliding mode controller

In this section we will show the design of the integral sliding mode controller in joint space. Actually, the model based part of the controller is partly implemented in task space for ease of computation.

The joint space tracking error can be given as

\[
e = q_d - q
\]

(14)

where \( q_d \) is the desired joint elongation. Then, using (6) and the uncertainties given in (9), the error dynamics can be written in joint space as:

\[
\dot{e}_1 = e_2 \\
\dot{e}_2 = \dot{q}_d - D_N^{-1}\left( \tau - B_N(q, \dot{q}) \dot{q} - Q_N(q) \right) + d_{21}
\]

(15)

Where

\[
D_N = J^T M_N J^{-T} \\
B_N = J^T \left( M_N J^{-T} + V_N J^{-T} \right) \\
Q_N = J^T G
\]

(16)

\[
d_{21} = -D_N^{-1} J \left( \Delta M \dot{J} \dot{q} + 2 \Delta M \dot{J} + \Delta V \dot{J} \right) \dot{q} + \Delta G \right) + D_N^{-1} f
\]

(19)

Comparing (15) and (1), we have

\[
x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}
\]

(20)

\[
f = \begin{bmatrix} e_2 \\ \dot{q}_d + D_N^{-1} \left( B_N(q, \dot{q}) \dot{q} + Q_N(q) \right) \end{bmatrix}
\]

(21)

\[
g = \begin{bmatrix} 0 \\ \dot{d} \end{bmatrix}
\]

(22)

and \( d \) is

\[
d = \begin{bmatrix} 0 \\ \dot{d}_{21} \end{bmatrix}
\]

(23)

From (2), the integral sliding surface for the Stewart platform manipulator is given as

\[
S = C \left( x - x(0) - \int_0^t \left( f(x, \omega) + g(x, \omega) \tau_0 \right) d\omega \right)
\]

(24)

where \( \tau_0 \) is the nominal control torque, \( x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \) and \( x(0) \) is the initial condition of the error dynamics and \( f \) and \( g \) are as given in (23) and (24) above.

Taking the derivative of \( S \)

\[
\dot{S} = C(\dot{x} - f - g \tau_0)
\]

(25)

Substituting (17), (22) and (23) and setting it equal to zero, the equivalent controller becomes

\[
\tau_{eq} = \tau_0 + D_N \dot{d}_{21}
\]

(26)

If the nominal controller to be used is chosen as

\[
\tau_0 = D_N \left( \ddot{q}_d + K_p \dot{e} + K_d \dot{e} \right) + B_N(q, \dot{q}) \dot{q} + Q_N(q)
\]

(27)

for some positive diagonal matrices \( K_p \) and \( K_d \) then the sliding dynamics of the system becomes

\[
\dot{e}_1 = e_2 \\
\dot{e}_2 = -K_p \dot{e} - K_d \dot{e}
\]

(28)

which shows a stable sliding dynamics. However, since the disturbance signal is now known, rather its bound, the control signal in (29) is replaced by

\[
\tau = \tau_0 + K_f S
\]

(29)
where $\tau_0$ is the nominal control signal given by (29), $K$ is gain of switching function, $f_i(S)$ is switching function.

The magnitude of $K$ required to achieve stability is

$$K \geq \left\| -D_N^{-1}J^T \left( \Delta M j^{-T} \dot{q} + \left( \Delta M j^{-T} + \Delta V j^{-T} \right) \dot{q} + \Delta G \right) \right\| + \left\| D_N^{-1} \right\|$$

(30)

B. Task space integral sliding mode control

Let $X_d$ be $(6 \times 1)$ vector of desired task space trajectories. Then, the task space tracking error vector and its rate vector are given as

$$e = X_d - X$$

(31)

$$\dot{e} = \dot{X}_d - \dot{X}$$

(32)

In the case of the Stewart platform manipulator, in task space also the input matrix $g$ is a function of the states as given by (35) and is not constant. The integral sliding surface for the Stewart platform manipulator is given as

$$S = C \left( X - X_0 - \int_0^t \left( f(x,\tau) + g(x, v) \tau \right) d\tau \right),$$

(33)

where $X_0$ is the initial condition of the states and $\tau_0$ is the nominal control torque. Following a similar procedure as in joint space case, taking derivative of (37) and equating it to zero, the equivalent controller becomes

$$\tau_{eq} = \tau_0 + (Cg)^{-T} Cd$$

(34)

The nominal controller is obtained using the computed torque control method and is given by

$$\tau_0 = J^T \left( M_N (\ddot{X}_d + K_p X_1 + K_d X_2) + V_N (X, \dot{X}) \dot{X} + G_N \right)$$

(35)

Where $K_p$ and $K_d$ are $6 \times 6$ constant diagonal matrixes determined from stiffness of material and desired transient performance.

Using this equivalent controller into (38), the sliding mode dynamics of the system in task space becomes

$$X_1 = X_2$$

$$\dot{X}_2 = \ddot{q}_d - M^{-1}_N \left\{ J^{-T} \left( \tau_0 + (G_{\theta})^{-1} G_d \right) - V_N (q, \dot{q}) \dot{q} - G_N (q) \right\} + d_{21}$$

(36)

Substituting for the nominal controller from (35), the sliding dynamics becomes

$$\dot{X}_2 = -K_p X_1 - K_d X_2 - \left( d_{21} - M^{-1}_N J^{-T} (Cg)^{-T} Cd \right)$$

(37)

This shows that the uncertainty can be compensated and the sliding mode dynamics is stable if the gain matrix $C$ is selected such that the last term in the bracket is made to be zero. However since the disturbance is not exactly known but only its bounds, the equivalent controller given by (34) cannot be realized. Moreover the value of the Jacobian matrix varies as the position of the manipulator varies. Therefore, the controller given by (34) is replaced by a switching function as follows

$$\tau = \tau_0 + J^{-T} K \frac{s}{\|s + \phi\|}$$

(38)

Where $\phi$ is a small positive boundary value and $K$ is chosen such that

$$K \geq \left\| d_m \right\| = \left\| M^{-1} \left( \Delta m X + \Delta C X + \Delta G \right) \right\|$$

(39)

The nominal control signal $\tau_0$ is calculated from the unperturbed model of the system in a feed forward manner as given in (6.52).

V. SIMULATION RESULTS AND DISCUSSION

For the simulation study of the performance of the controller, a typical 6-6 geometry Stewart platform with the geometric parameters given in table I [1] is implemented using Simmechanics tool box of MATLAB.

<table>
<thead>
<tr>
<th>Geometric Specifications of Stewart platform</th>
<th>Joint positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base radius 0.8m</td>
<td>$\frac{\pi}{6}$</td>
</tr>
<tr>
<td>Platform radius 0.5m</td>
<td>$\frac{\pi}{4}$</td>
</tr>
<tr>
<td>Mass of platform 32kg</td>
<td>$\frac{\pi}{6}$</td>
</tr>
<tr>
<td>Mass of upper leg 4kg</td>
<td>$\frac{\pi}{12}$</td>
</tr>
<tr>
<td>Mass of lower leg 4kg</td>
<td>$\frac{\pi}{12}$</td>
</tr>
<tr>
<td>Initial Height 1.5m</td>
<td>$\frac{\pi}{2}$</td>
</tr>
<tr>
<td>Platform Inertia Ixx=2, Iyy=2 and Izz=4</td>
<td>$\frac{\pi}{12}$</td>
</tr>
<tr>
<td>Leg Inertia upper Ixx=0.75, Iyy=0.75, Izz=0.018</td>
<td>$\frac{\pi}{12}$</td>
</tr>
<tr>
<td>Leg Inertia lower Ixx=0.03, Iyy=0.03, Izz=0.002</td>
<td>$\frac{\pi}{12}$</td>
</tr>
<tr>
<td>CG of upper leg 0.75m from top</td>
<td>$\frac{\pi}{12}$</td>
</tr>
<tr>
<td>CG of lower leg 0.15m from base</td>
<td>$\frac{\pi}{12}$</td>
</tr>
</tbody>
</table>

The trajectory used to test the performance of the controllers is a fast trajectory having heave motion, circular motion in XY plane and angle twists is given below [8].

$$x(t) = 0.5 \left\{ 1 - \exp \left( -\pi t \right) \right\} \cos \left( 1.88\pi t \right), m$$

$$y(t) = 0.5 \left\{ 1 - \exp \left( -\pi t \right) \right\} \sin \left( 1.88\pi t \right), m$$

$$z(t) = 3 + 0.02 \sin \left[ 2\pi \left( \frac{0.1 + 5.9t}{10.5} \right) + \frac{\pi}{24} \right], m$$

$$\alpha(t) = 0, deg$$

$$\beta(t) = 0.5 \left\{ 1 - \exp \left( -\pi t \right) \right\} \sin \left( 0.86\pi t \right), deg$$

$$\gamma(t) = 0.5 \left\{ 1 - \exp \left( -\pi t \right) \right\} \sin \left( 0.74\pi t \right), deg$$

A. Joint space integral sliding mode controller

The integral sliding surface given in (24) is implemented after the expressions (21)-(23) are substituted. After
s = C \begin{bmatrix} e_1(t) - e_1(0) \\ e_2(t) - e_2(0) \\ \int_0^t (K_p e_1 - K_d e_2) \, dt \end{bmatrix} 
\end{equation} (48)

where C is a 6x12 matrix to be determined using genetic algorithm and K_p and K_d are 6x6 diagonal matrixes used in the nominal control signal. As in the joint space case, the gain matrix C can be partitioned into two 6x6 square matrices as \( C = [C_1 \ C_2] \) and \( C_2 \) is assumed to be identity matrix while \( C_1 \) is taken to be a diagonal matrix. This helps in reducing the number of parameters to be optimized by the genetic algorithm and also reduces the time required for computation. Moreover, it decouples the six sliding surfaces and speed up the computation of control signal. Using C1 and C2 Then (48) can be rewritten as

\[ S = C_1 X_1 + X_2 + \int_0^t (-C_1 X_2 + K_p X_1 + K_d X_2) \, dt \] (49)

Similar procedure and parameter is used to determine controller parameters.

The tracking performance of the controller when the platform has a payload of 200kg is shown for the 6 DOF’s for the given trajectory. In all cases the error is not asymptotically decaying but is bounded. Overall, the error is much smaller. Especially the integral SMC is able to compensate gravitational torque which can be easily seen from the tracking performance in z direction. The orientation angle errors are slightly greater than the translational motion error of x, y and z. This is due to the error in the numerical algorithm. The initial guess taken for the numerical algorithm is same for all trajectory positions. Comparing the tracking performances, the task space implementation shows better performance in most of the cases. This is because of the synchronization error problem of joint space implementation and the capacity of task space in handling synchronization error. However, when we consider the improvements obtained with respect to the need for the costly forward kinematics, it can be easily seen that joint space implementation is better. Hence it can be concluded that the joint space implementation of sliding mode control is better than task space implementation.

VI. CONCLUSION

In this paper we compared the performance of task space and joint space integral sliding mode implementations. The nonlinear integral sliding surface is designed using genetic algorithm and a smooth six dimensional trajectory is used to test the performance of the two controllers. Simulation results showed that joint space implementation has better overall performance than task space implementation.
Fig. 7. Comparison of trajectory tracking performance in x direction

Fig. 6.27. Comparison of trajectory tracking performance in y direction

Fig. Comparison of trajectory tracking performance in x direction

Fig. 6.27. Comparison of trajectory tracking performance in y direction

Fig. 6.27. Comparison of trajectory tracking performance in y direction
REFERENCES


