Genetic Algorithm based Integral Sliding Surface Design and Its Application to Stewart Platform Manipulator Control

S. Dereje, Mahantesh K. Pattanshetti, Anamika Jain and R. Mitra

***Abstract*—**Integral sliding mode control has the potential to solve some of the drawbacks of simple sliding mode but the design of the integral sliding surface for nonlinear systems having unmatched uncertainty is a difficult task. In this paper, a design method using genetic algorithm is proposed and its effectiveness is tested using highly nonlinear system having unmatched uncertainty. The design of the integral nonlinear sliding surface is formulated as an optimization problem which minimizes the error between the nominal and perturbed system and an optimal gain is found using genetic algorithm. Then the problem of trajectory tracking control of Stewart platform manipulator is employed as a test bed. The controller is implemented in task space and joint space/task space hybrid and performances were compared. Simulations showed that the genetic algorithm based integral sliding mode controller has superior performance than existing controllers. Furthermore joint/task space hybrid implementation gives slightly bigger mean square error value in some directions but needs a smaller control effort compared to the pure task space implementation.

***Keywords*—**Genetic algorithm, Integral sliding mode control, Nonlinear control, Robust control, Sliding mode control, Stewart platform manipulator

I. Introduction

Integral sliding mode control (ISMC) is an mprovement to conventional sliding mode control that uses a nonlinear sliding surface having an integral term[1][2]. The integral sliding surface is designed to constrain the system states to be on sliding mode from initial time and by that it gets rid off the reaching phase problem of simple sliding mode control. Moreover, integral sliding surface improves the stability of sliding dynamics and hence enables to enhance robustness against unmatched uncertainties [3][4]. Like any other sliding mode control design, the design of the controller involves two basic steps: the deisgn of stable sliding surface and design of a control law. The controller law in ISMC consists of a nominal controller used to stablize the nominal system and a discountinous one which will be used to reject the uncertainties. The nominal controller can be designed by various design methods [3]. On the other hand, the design of the integral sliding surface is very important and is not a simple task, specifically when the system to be controller is nonlinear and has unmatched uncertainties.

In the literature, methods such as equivalent transfer function, matrix fraction description and Lyapunov's direct method [5] have been proposed for the design of integral sliding surface for systems having matched uncertainty. However, there are no established methods for systems having unmatched uncertainty. A recently proposed method is to make use of linear matrix inequalities(LMI) [4]. Though, the authors have shown the effectiveness of the method, formulating the design of integral sliding surface in to LMI for complex system like the Stewart platform manipulator is not an easy task.

Stewart platform manipulator is a parallel kinematics manipulator having high structural rigidity and stiffness which makes it much more preferable than serial robots for precision applications such as machining, robotic surgery, pointing and so on [6][7][8]. To effectively utilize the structural advantages of the manipulator to the above mentioned applications, a robust and high performance controller is necessary. But, the dynamic model of this manipulator, the relationship between the forces/torques which have to be given at the legs and the acceleration of the center of the platform, is highly nonlinear and coupled making the design of controller challenging.Various authors have proposed simple sliding mode controller for robust control of the manipulator [7][9]. However, drawbacks of simple sliding mode control, namely chattering, nonrobustness of its reachng phase and lack of robustness to unmathced uncertainties have prevented the practical applicability of simple sliding mode controller to this manipulator[10][11][12]. Hence integral sliding mode control is a promising solution for high performance control of this manipulator[13].

Therefore the main objectives of the paper are: to show how genetic algorithm can effectively be used to design an integral sliding manifold for multiple input multiple output systems like Stewart platform manipulator and to compare joint space and task space implementations of integral sliding mode controllers in Stewart platform manipulator. Genetic algorithm is a multidimenstionsal search algorithm and has been used for design of various controllers [14][15]. In our proposed

Manuscript received March 29, 2011

S. Dereje was with IIT Roorkee, India and is now with Graphic Era University, Bell Road, Clement Town, Dehradun, 248002, India, (corresponding author to provide phone: +91-9720-573668; fax: +91-135-2644025; e-mail: dnderejesh@gmail.com).

Mahantesh K. Pattanshetti and Anamika Jain, are with Graphic Era University, Bell Road, Clement Town, Dehradun, 248002, India,(e-mail: geudoon@gmail.com , anamikajain2829@gmail.com).

R. Mitra is with department of Electronics and computer Engineering , IIT Roorkee, 247667, India (e-mail: rmtrafec@iitr.ernet.in).

controllers, integral sliding surface is designed using genetic algorithm. To apply genetic algorithm for the design of integral sliding surface, first the integral sliding surface is formulated as a parameter which shows the mismatch between nominal and actual systems and then genetic algorithm is used to select gain values which minimize the mismatch in the presence of bounded uncertainties. The fitness function used in the genetic algorithm is designed to achieve multiple objectives: to constrain system states on sliding surface, to reduce effect of unmatched uncertainty and to reduce chattering. In applying the designed integral sliding surface for trajectory control of Stewart platform manipulator, a nominal controller which is designed using equivalent control approach is employed. The implementation of the nominal controller needs computation of the dynamic parameters and has to be carefully analyzed. Hence two different implementations, namely task space and joint space implementations, have been compared. Moreover, the comparison of the two implementations helps to analyse the effect of unmatched uncertainty, since it is very less in the second case. The simulation results have shown that, integral sliding mode controller designed using genetic algorithm reduces chattering and achieves high performance as compared to simple sliding mode control and PID. The control signals were smooth and practically realizable. The selection of range of values for the parameters and stability issues have also been discussed.

The paper is organized as follows: section two gives background on integral sliding mode control and formally presents the problem statement. Section three discusses the kinematic and dynamic modeling of Stewart platform manipulator. Section four deals with the design of integral sliding surface using genetic algorithm. The task space and joint space designs of integral sliding mode controller for the Stewart platform manipulator is discussed in section five. Section six contains the simulation results and discussion and then conclusion follows.

II. PROBLEM FORMULATION

Consider an uncertain nonlinear system given as

$$\dot{x} = f\left(x,t\right) + g\left(x,t\right)u + d(t) \tag{1}$$

where x is nx1 dimensional state vector, f(x, t) and g(x, t) are nX1 and nxm dimensional vector and matrix valued smooth nonlinear functions, d is nx1 dimensional vector of the uncertainties and u is mx1 dimensional vector of control inputs.

The integral sliding surface s and the control signal u are given by

$$s\left(x,t\right) = C\left[x - x\left(0\right) - \int_{t=0}^{t}\left(f\left(x,\tau\right) + g\left(x,\tau\right)u_0\right)d\tau\right] \tag{2}$$

$$u = u_0 + u_1 \tag{3}$$

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

$$u_1 = -Kf_s(s) \tag{4}$$

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

Then the problem in integral sliding mode control is to find 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.

In the mathematical formulation of the integral sliding surface given in (2), the sliding variable s can be seen as the mismatch between the nominal and perturbed system. Hence the objective is to find the gain C and control law u which minimizes this sliding parameter. And hence we employ genetic algorithm for the design of C and u is designed using the equivalent control method. The above formulation can be drawn in block diagram as shown in Fig. 1.

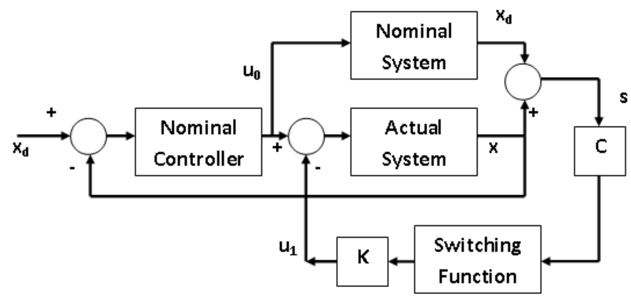

Fig.1 block diagram representation of ISMC

III. MODELING THE STEWART PLATFORM 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 … 6) and P_i (i = 1, 2… 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_p 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, α, β, γ] where the three angles α, β, γ 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

$$\left|B_iP_i\right| = \left\|Rp_i + r - b_i\right\| = q_i \tag{5}$$

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 α, β, γ for a given measured value of q_i, i=1, 2… 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

$$\dot{X} = J^{-1}\dot{q} \tag{6}$$

Where J is the Jacobean matrix of the platform with respect to the base frame.

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 [8][10]. Using Lagrangian method, the actuator torque τ is given in task space as

$$M(X)\ddot{X} + V(X,\dot{X})\dot{X} + G(X) = J^{-T}\tau \tag{7}$$

Where $X=[r_x, r_y, r_z, \alpha, \beta, \gamma]$ is the task space position and orientation of center of movable platform, M(.) is the inertia matrix, V(.,.) is the coriolis/centrifugal force coefficient matrix and G(.) is the gravitational torque.

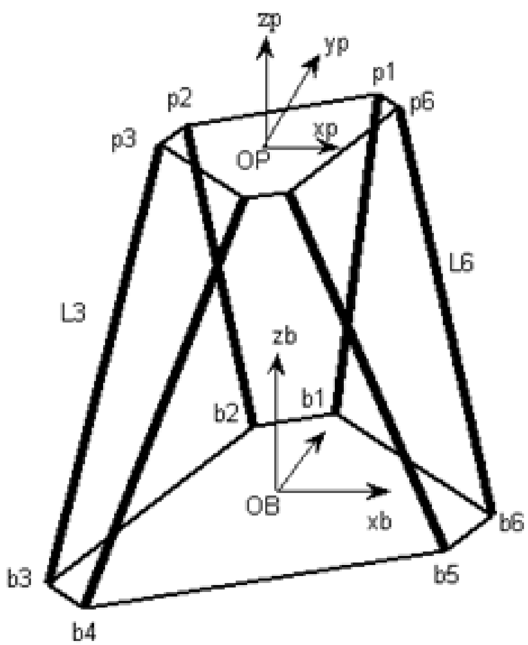

Fig.2 Generalized Stewart platform manipulator

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.

$$\begin{aligned} M &= M_N + \Delta M \\ V &= V_N + \Delta V \\ G &= G_N + \Delta G \end{aligned} \tag{8}$$

The perturbations ΔM, ΔC and ΔG are assumed to have the following bounds

$$\begin{aligned} \|\Delta M\| &\le M_m \\ \|\Delta V\| &\le V_m \\ \|\Delta G\| &\le G_m \end{aligned} \tag{9}$$

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

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= M^{-1}\left(J^{-T}\tau - V(X_1, X_2)X_2 - G(X_1)\right) + d \end{aligned} \tag{10}$$

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

$$d = M^{-1}\left(-\Delta M\ddot{X} - \Delta V\dot{X} - \Delta G\right) \tag{11}$$

Comparing (10) and (1),

$$f = \begin{bmatrix} X_2 \\ -M^{-1}\left(V(X_1, X_2)X_2 + G(X_1)\right) \end{bmatrix} \tag{12}$$

and

$$g = \begin{bmatrix} 0 \\ M^{-1}J^{-T} \end{bmatrix} \tag{13}$$

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 Integral Sliding Surface

In this section we will illustrate the application of genetic algorithm for the design of integral sliding surface. Before we give the details of the design of sliding surface, we will briefly revise the basics of genetic algorithm.

A. Genetic algorithm

Genetic algorithm is an evolutionary algorithm based on Darwin's theory of selection of the fittest. It is a multidimensional search algorithm which solves the local minima problem of classical algorithms[15]. In the literature, the basic element of a genetic algorithm is known the chromosome as it is based on the evolutionary theory of Darwin. The chromosome contains the genetic information for a given solution and can be coded by using either binary or real string. The algorithm starts by generating some number of chromosomes randomly as candidate solutions to a given problem. A fitness function which in effect is a performance index is used to select the best solutions in the population to be parents to the offspring that comprises the next generation[15]. The more fit the parent is, it will have greater probability of selection.

The selection of parent chromosomes is done using various methods including the roulette wheel method. Then offsprings are produced by selecting parent chromosomes for breeding and crossing over some of the genetic material. This process is known as crossover. Another operator which is used to introduce some element of randomness into the solution is mutation. In the process of mutation a randomly selected gene of an offspring is changed. Mutation occurs in not all offsprings but very few ones and it is used to introduce some randomness. This process continuous until a global solution is obtained. Figure 3 shows the algorithm described above.

Therefore, in a GA optimization, parameters such as, the initial population size, crossover rate and mutation rate, coding size of chromosomes and fitness function, have to be selected. The most important one which determines the problem at hand is the fitness function.

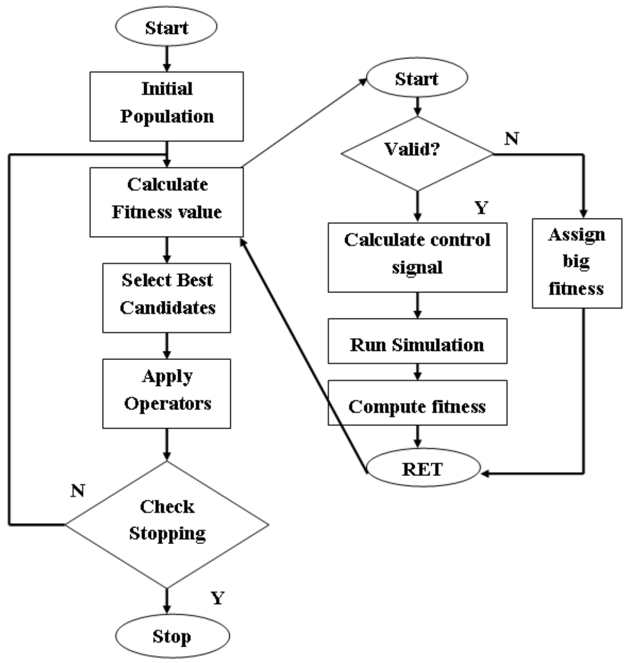

Fig. 3 Flow chart for genetic algorithm based integral sliding surface design

B. *Fitness function and parameter selection*

The most important parameter in a GA optimization is the fitness function since it determines the objective of the optimization itself. A poorly selected objective function may give a completely wrong result. In [6], the main objective was to decrease chattering and obtain fast response and the authors used fitness function given by

$$f = e^{-(t_h/w_1)^2} \text{ x } e^{-(c_m/w_2)^2} \tag{10}$$

where t_h is the time to hit the sliding surface, c_m is the amount of chattering and w_1 and w_2 are weight factors. In the current discussion, the first term is not needed because the system is on the sliding surface from the initial time. The main objective here is to keep the system on the sliding surface from the initial time. Another objective is to minimize chattering and the effect of the unmatched uncertainty as shown in (9). Hence for the fitness function, the product of two terms is taken: the first one is used to constrain the states on the sliding surface and the second one is used to minimize the effect of unmatched uncertainty. The fitness function is formulated as in (11) below.

$$f_f = e^{-(c_s/w_1)^2} \text{ x } e^{-(u_m/w_2)^2} \tag{11}$$

Where c_s is the term used to constrain the states to the sliding surface and is taken as

$$c_s = \sum_{i=1}^{N} \|s_i\| \tag{12}$$

s_i is the value of the sliding variable in iteration i. In the implementation, a chromosome which has an initial value of s greater than zero by some upper limit is given a big penalty.

V. Design of Controller

As stated above, Stewart platform manipulator is a parallel kinematics manipulator having high structural rigidity and stiffness which makes it much more preferable than serial robots for precision applications such as machining, robotic surgery, pointing and so on[6][7][8]. However, the relationship between the forces/torques which have to be given at the legs to drive the system in a desired trajectory and the acceleration of the center of the platform is highly nonlinear and coupled. Moreover, the dynamic model of the manipulator has matched uncertainties from parameter variation, actuator friction and so on. Furthermore, it also has unmatched uncertainties when control algorithms are implemented in the task space and forward kinematics is estimated using numerical algorithms [10]. All this makes the controller design very challenging. Therefore to effectively utilize the structural advantages of the manipulator to the above mentioned applications, a robust and high performance controller is necessary.

A controller in Stewart platform manipulator has to generate torque signals which will be applied to the leg actuators such that the moveable platform moves in a desired trajectory at a desired speed. Generally, there are two approaches to the controller design problem. The first one is to convert the desired task space position, velocity and acceleration of the platform center to desired joint leg lengths and close the loop by using measured leg lengths as feedback. This approach is known as joint space. In this approach, the individual leg measurements and desired values are taken separately and control is single input single output(SISO). In the second approach, the desired task space position, velocity and acceleration is not converted to desired leg length rather it is used directly by taking measured or estimated task space position, velocity and acceleration as feedback. Hence in this approach, control signal is calculated in task space and then it will be converted to joint space using the Jacobian matrix. The manipulator control system in this approach is therefore a multiple input multiple output system(MIMO). Simplified block diagram of these two approaches is given in Fig.4 and Fig.5. Both of these two implementations have their own advantages and disadvantages. In the joint space approach, measured leg lengths which will be used for feedback can be obtained using readily available and less costly sensors and hence it results in less costly implementation. Moreover, the individual SISO loops can be implemented parallely resulting in a faster controller. However lack of synchronization puts a limit to its performance and hence joint space approach can not give high performance. Furthermore, the difficulty of computing dynamic parameters in joint space makes it unsuitable for model based controller implementation. On the other hand, in task space approach, the controller is MIMO and synchronization error is minimal. Hence it has a potential to give high performance. The drawback in this is getting the task space feedback signals. Either a complex estimation method or costly measurement is necessary. In this section, we will design integral sliding mode controller in both joint space and task space and compare their performances in the next sections. Actually, the joint space implementations is not pure joint space, it is a kind of hybrid. The model based nominal controller is implemented in task space using desired task space positions, velocity and acceleration but the feedback signal used in the discontinous controller is obtained from measured leg lengths.

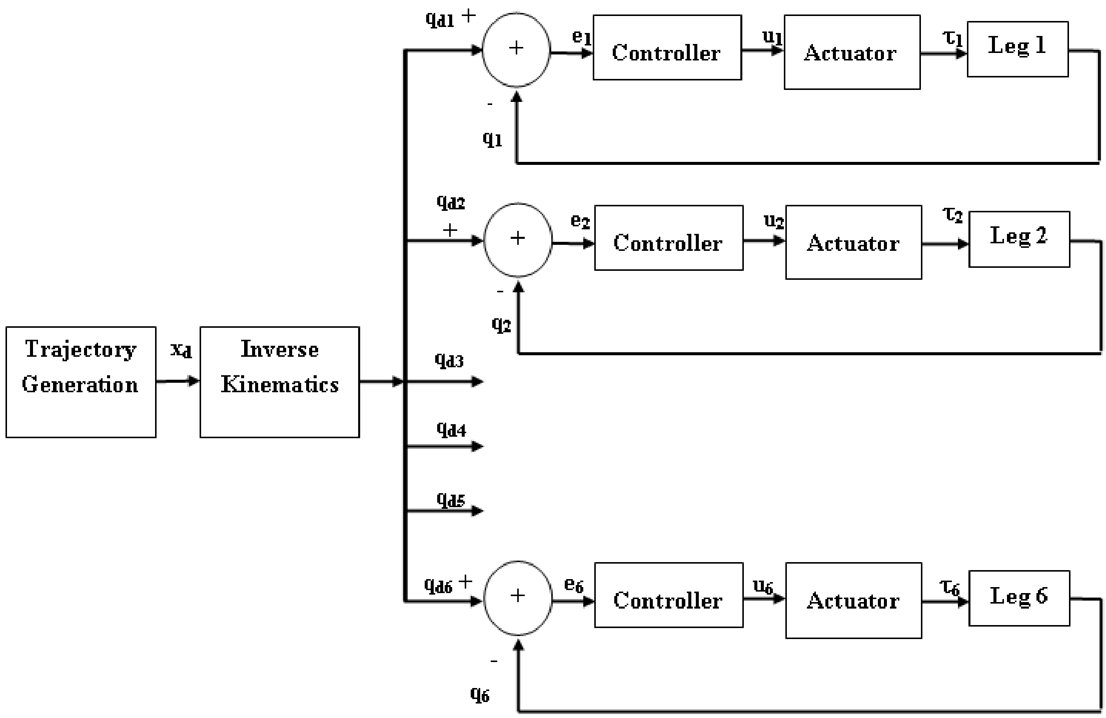

Fig.4 Joint space trajectory tracking control approach for a Stewart platform manipulator

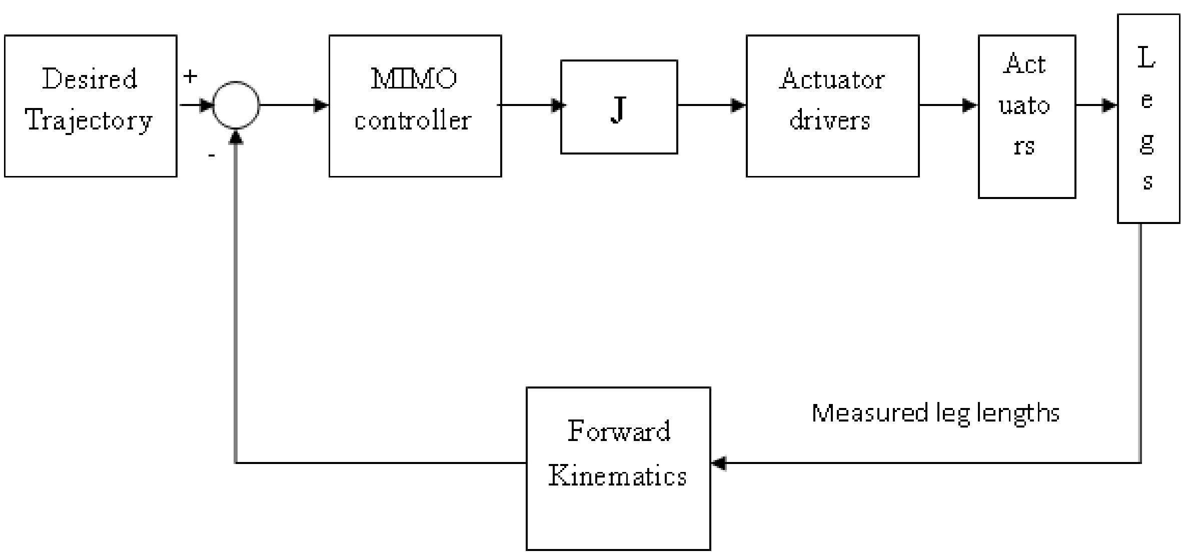

Fig.5 Task space trajectory tracking control approach for a Stewart platform manipulator

A. Task space integral sliding mode controller

Let X_d be (6x1) vector of desired task space trajectories. Then, the task space tracking error vector and its rate vector are given as

$$e = X_d - X \tag{13}$$

$$\dot{e} = \dot{X}_d - \dot{X} \tag{14}$$

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

$$S = C\left(X - X_0 - \int_{t_0}^{t} \left(f(x,\tau) + g(x,v)\tau_0\right) d\tau \right), \tag{15}$$

where X_0 is the initial condition of the states and τ_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

$$\tau_{eq} = \tau_0 + (Cg)^{-1} Cd \tag{16}$$

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

$$\tau_0 = J^T \left(M_N \left(\ddot{X}_d + K_p X_1 + K_d X_2 \right) + V_N \left(X, \dot{X} \right) \dot{X} + G_N \right) \tag{17}$$

Where K_p and K_d are 6x6 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

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = \ddot{q}_d - M_N^{-1} \begin{pmatrix} J^{-T} \left(\tau_0 + (G_s g)^{-1} G_s d \right) - \\ V_N (q, \dot{q}) \dot{q} - G_N (q) \end{pmatrix} + d_{21} \tag{18}$$

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

$$\dot{X}_2 = -K_p X_1 - K_d X_2 - \left(d_{21} - M_N^{-1} J^{-T} (Cg)^{-1} Cd \right) \tag{19}$$

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

$$\tau = \tau_0 + J^{-T} K \frac{s}{\|s + \varphi\|} \tag{20}$$

Where φ is a small positive boundary value and K is chosen such that

$$K \geq \|d_m\| = \left\| M^{-1} \left(\Delta m_m \ddot{X} + \Delta C_m \dot{X} + \Delta G_m \right) \right\| \tag{21}$$

The nominal control signal τ_0 is calculated from the unperturbed model of the system in a feed forward manner as given in (6.52)

B. Joint/task space hybri integral sliding mode controller

In this section we will show the design of the integral sliding mode controller in joint space. As stated above, the model based part of the controller is implemented in task space for ease of computation and hence the combined controller is joint space/task space hybrid controller.

The joint space tracking error can be given as

$$e = q_d - q \tag{22}$$

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:

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= \ddot{q}_d - D_N^{-1} \left(\tau - B_N (q, \dot{q}) \dot{q} - Q_N (q) \right) + d_{21} \end{aligned} \tag{23}$$

Where

$$D_N = J^T M_N J^{-T} \tag{24}$$

$$B_N = J^T \left(M_N \dot{J}^{-T} + V_N J^{-T} \right) \tag{25}$$

$$Q_N = J^T G \tag{26}$$

$$d_{21} = -D_N^{-1} J^T \left(\Delta M J^{-T} \ddot{q} + \left(\Delta M \dot{J}^{-T} + \Delta V J^{-T} \right) \dot{q} + \Delta G \right) + D_N^{-1} f_f \tag{27}$$

Comparing (15) and (1), we have

$$x = \begin{bmatrix} e & \dot{e} \end{bmatrix}^T \tag{28}$$

$$f = \begin{bmatrix} e_2 \\ \ddot{q}_d + D_N^{-1} \left(B_N (q, \dot{q}) \dot{q} + Q_N (q) \right) \end{bmatrix} \tag{29}$$

and

$$g = \begin{bmatrix} 0 & -D_N^{-1} \end{bmatrix}^T \tag{30}$$

and d is

$$d = \begin{bmatrix} 0_{6x1} & d_{21} \end{bmatrix}^T \tag{31}$$

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

$$S = C\left(x - x(0) - \int_0^t \left(f(x,\omega) + g(x,\omega)\tau_0 \right) d\omega \right), \tag{32}$$

where τ_0 is the nominal control torque, $x = \begin{bmatrix} e & \dot{e} \end{bmatrix}^T$ 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

$$\dot{S} = C\left(\dot{x} - f - g\tau_0 \right) \tag{33}$$

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

$$\tau_{eq} = \tau_0 + D_N d_{21} \tag{34}$$

If the nominal controller to be used is chosen as

$$\tau_0 = D_N \left(\ddot{q}_d + K_p e + K_d \dot{e} \right) + B_N (q, \dot{q}) \dot{q} + Q_N \tag{35}$$

for some positive diagonal matrices K_p and K_d, then the sliding dynamics of the system becomes,

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= -K_p e - K_d \dot{e} \end{aligned} \tag{36}$$

which shows a stable sliding dynamics. However, since the disturbance signal is not known, rather it's bound; the control signal in (29) is replaced by

$$\tau = \tau_0 + K f_s(S) \quad (37)$$

where τ_0 is the nominal control signal given by (29), K is gain of switching function, $f_s(S)$ is switching function.

The magnitude of K required to achieve stability is

$$K \geq \left\| -D_N^{-1} J^T \left(\Delta M J^{-T} \ddot{q} + \left(\Delta M \dot{J}^{-T} + \Delta V J^{-T} \right) \dot{q} + \Delta G \right) \right\| + \left\| D_N^{-1} f_f \right\| \quad (38)$$

VI. Simulation Results and Discussion

For the simulation study of the performance of the proposed design of integral sliding surface and 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 I Geometric Specifications of Stewart platform

Joint positions						
Base	$\frac{\pi}{6}$	$\frac{\pi}{2}$	$\frac{5\pi}{6}$	$\frac{7\pi}{6}$	$\frac{3\pi}{2}$	$\frac{11\pi}{6}$
Platform	$\frac{\pi}{12}$	$\frac{7\pi}{12}$	$\frac{3\pi}{4}$	$\frac{5\pi}{4}$	$\frac{17\pi}{12}$	$\frac{23\pi}{12}$
Base radius 0.8m						
Platform radius 0.5m						
Mass of platform 32kg						
Mass of upper leg 4kg						
Mass of lower leg 4kg						
Initial Height 1.5m						
Platform Inertia Ixx=2,Iyy=2 and Izz=4						
Leg Inertia upper Ixx=0.75,Iyy=0.75,Izz=0.018						
Leg Inertia lower Ixx=0.03, Iyy=0.03, Izz=0.002						
CG of upper leg 0.75m from top						
CG of lower leg 0.15m from base						

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 as given below [8].

$$x(t) = 0.5\{1 - \exp(-\pi t)\}\cos(1.88\pi t) \quad (39.1)$$

$$y(t) = 0.5\{1 - \exp(-\pi t)\}\sin(1.88\pi t) \quad (39.2)$$

$$z(t) = 3 + \frac{0.02}{1+0.9t}\sin\left\{2\pi t\left(\frac{0.1+5.9t}{10.5}\right) + \frac{\pi}{24}\right\} \quad (39.3)$$

$$\alpha(t) = 0 \quad (39.4)$$

$$\beta(t) = 0.5\{1 - \exp(-\pi t)\}\sin(0.86\pi t) \quad (39.5)$$

$$\gamma(t) = \gamma_0\{1 - \exp(-\pi t)\}\sin(0.74\pi t) \quad (39.6)$$

A. Task space integral sliding mode controller

The integral sliding surface (33) is implemented after it is rewritten as in (48) by substituting the nominal controller (34) and system dynamics functions into (32). The last expression after the substitutions is:

$$s = C\left\{\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} - \begin{bmatrix} e_1(0) \\ e_2(0) \end{bmatrix} - \begin{bmatrix} \int_0^t e_2 d\omega \\ \int_0^t \left(-K_p e_1 - K_d e_2\right) d\omega \end{bmatrix}\right\} \quad (41)$$

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. To reduce the number of parameters to be optimized, C is partitioned into two sub matrices of each 6x6 and is rewritten as in (41) below.

$$C = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \quad (42)$$

Using this form, the (41) can be rewritten as

$$S = C_1 X_1 + X_2 + \int_0^t \left(-C_1 X_2 + K_p X_1 + K_d X_2\right) d\omega \quad (43)$$

In (42) C_2 is taken as a diagonal matrix of elements 1. 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. The range of values used for C_1 in the genetic algorithm optimization is determined using step response of each dimension. The procedure followed is as followes.

1. The controller is first used for single direction regulation control and the step response is observed.
2. Then the controller gains are tuned until a desired step in terms of settling time, overshoot and steady state error is obtained.
3. Then the controller parameters used to obtain the best and worst step responses are used as range of values in the genetic algorithm optimization.

Using this procedure, the diagonal elements of C_1 are allowed to vary between 100 and 400 and the gain value K_1 of the switching function is allowed to vary between 5000 and 15000. The parameters used in the genetic algorithm are default values of MATLAB and are as follows

Initial population 20
Crossover rate =0.8
Mutation rate =0.01
Maximum number of generations 20

With these parameters, the optimal solutions obtained using the multi-objective genetic algorithm described in section three is given below.

The parameters of the integral sliding surface are

$$C_1 = \mathrm{diag}(117 \quad 113 \quad 87 \quad 155 \quad 157 \quad 150) \quad (43)$$

And the gain value used for the switching function are

$$k = \mathrm{diag}(7983 \quad 7218 \quad 13074 \quad 6563 \quad 8468 \quad 6739) \quad (44)$$

The Kp and Kd values used in the nominal controller are determined assuming a damping factor of 0.7 so that

$$K_d = 2\sqrt{K_p} \quad (45)$$

and hence the values used are chosen as

$$K_p = \mathrm{diag}(4 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4) x10^4 \quad (46)$$

$$K_d = \mathrm{diag}(400 \quad 400 \quad 400 \quad 400 \quad 400 \quad 400) \quad (47)$$

The trajectory tracking performance of the above joint space controller is shown in Figure 3-Figure 9. The results are given in task space after the joint space measurements are transformed to task space using numerical forward kinematics. As can be seen from the results, the joint space integral sliding mode controller performance is much better than PID controller and simple sliding mode controllers.

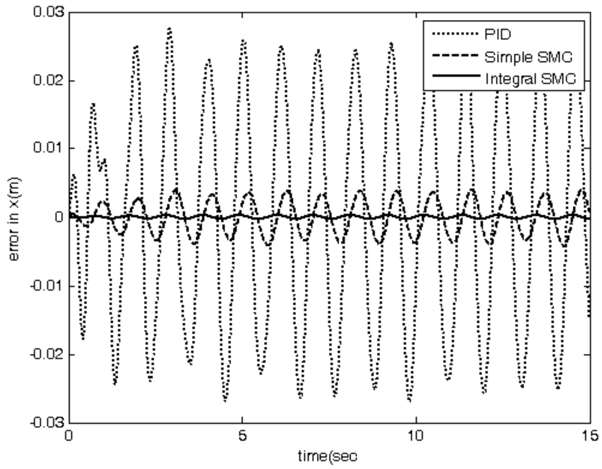

Fig. 4 Compariosn of tracking performances of task space ISMC, simple sliding mode and PID in x direction

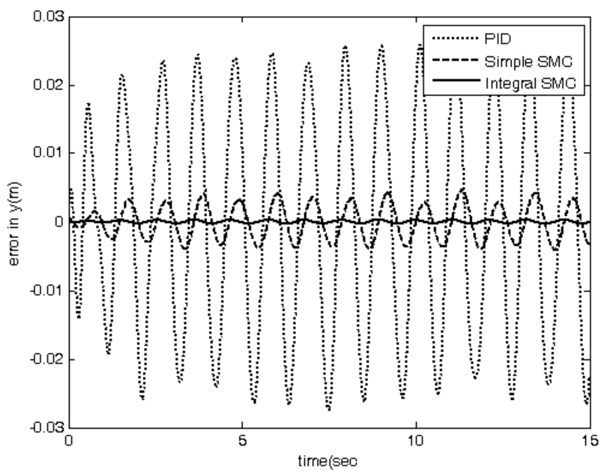

Fig. 5 Compariosn of tracking performances of task space ISMC, simple sliding mode and PID in y direction

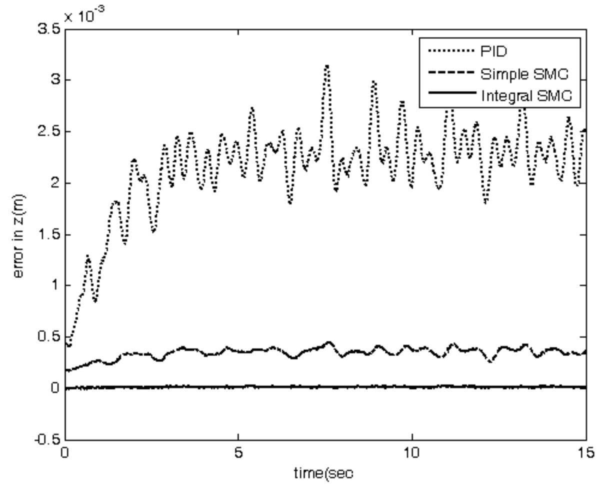

Fig. 6 Compariosn of tracking performances of task space ISMC, simple sliding mode and PID in z direction

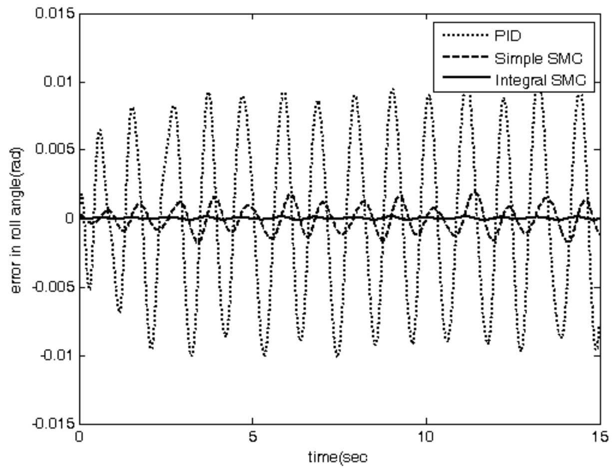

Fig. 7 Compariosn of tracking performances of task space ISMC, simple sliding mode and PID for roll angle

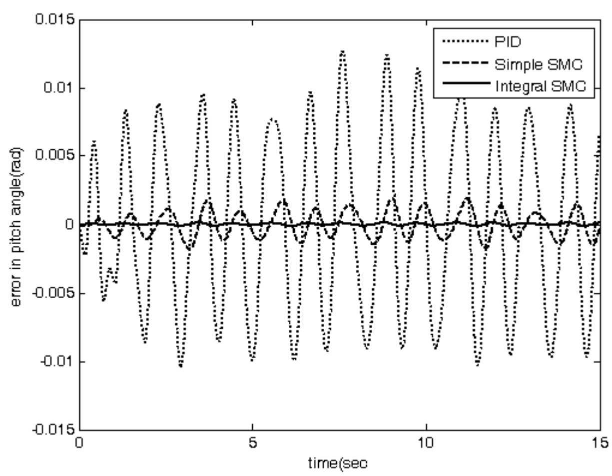

Fig. 8 Compariosn of tracking performances of task space ISMC, simple sliding mode and PID for pitch angle

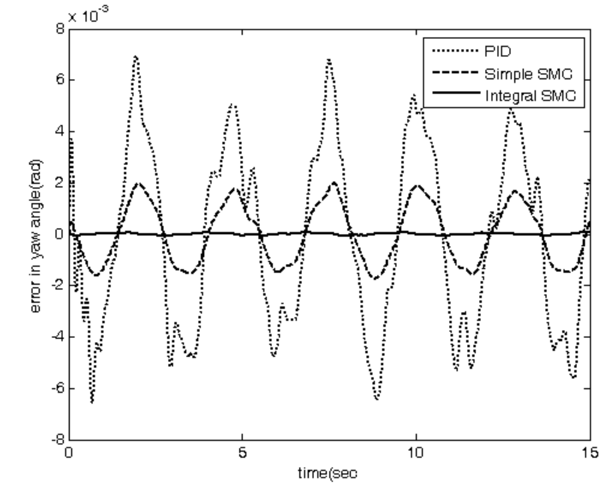

Fig. 9 Compariosn of tracking performances of task space ISMC, simple sliding mode and PID for yaw angle

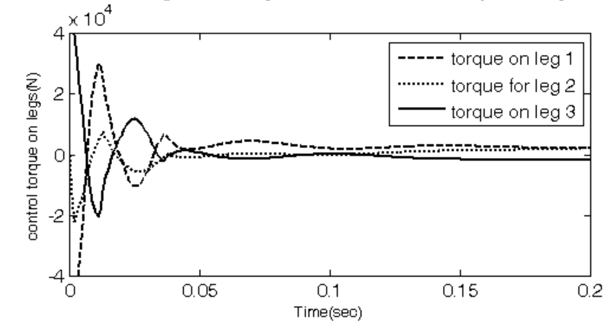

Fig. 10 Control forces of the task space ISMC for the first three legs

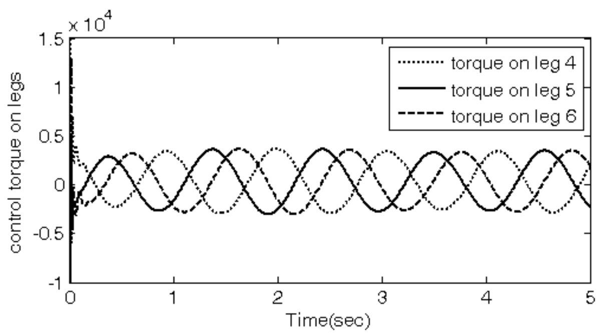

Fig. 11 Control forces of the task space ISMC for the first three legs

B. Joint/Task space hybrid integral sliding mode controller

The integral sliding surface given in (24) is implemented after the expressions (21)-(23) are substituted. After simplification it becomes

$$S = C\left(e - e(0) - \int \dot{e}\right) + \left(\dot{e} - e(0) + \int K_p e + K_d \dot{e}\right) \quad ()$$

where C is 6x12 diagonal matrix K_p and K_d are 6x6 diagonal matrixes used in the nominal control signal. As in the task 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 diagonal matrix with value of unity while C_1 is determined using genetic algorithm. Again in this case also, the range of values of C_1 are determined in a similar procedure as in the task space case.

The trajectory tracking results of the above integral sliding mode controller are given in Fig. 10-12. The results are given in task space after the joint space measurements are transformed to task space using numerical forward kinematics. As can be seen from the results, the joint space/task space hybrid integral sliding mode controller performance is also much better than simple sliding mode controller.

C. Comparison of task space and hybrid implementations

Comparing the tracking performances of task space and joint space implementations, 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

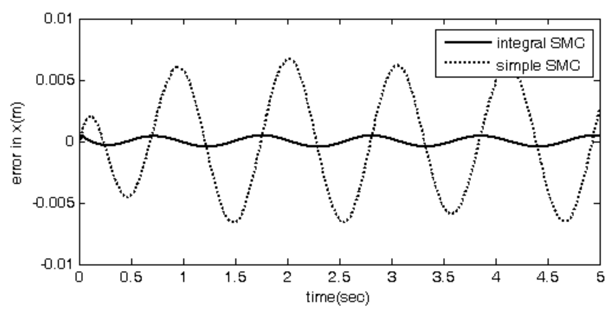

Fig. 12 Compariosn of tracking performances of task space ISMC with simple sliding mode in x direction

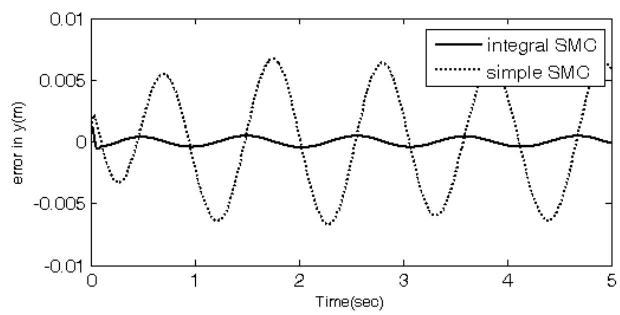

Fig. 13 Compariosn of tracking performances of task space ISMC with simple sliding mode in y direction

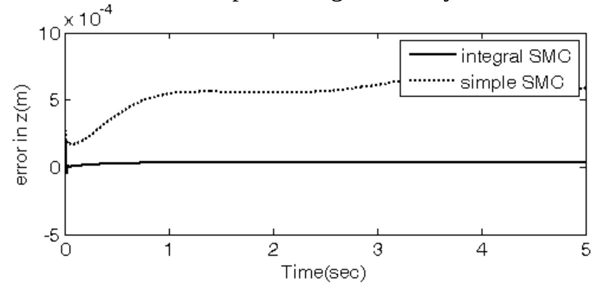

Fig. 14 Compariosn of tracking performances of task space ISMC with simple sliding mode in z direction

VII. Conclusion

In this paper, we have shown how genetic algorithm can be used to design integral sliding surface in sliding mode control and we have compared two implementations of the designed controllers. The controllers were designed for the highly nonlinear 6DOF parallel robot of Stewart platform manipulator. The MATLAB simulation results have shown that the designed integral sliding mode controller performs better than simple sliding mode controller and PID controller. With regard to the comparsion of task space and joint space/task space hybrid implementations, it was shown that the hybrid implementation results in a better controller as it gives reasonable performance with less control effort.

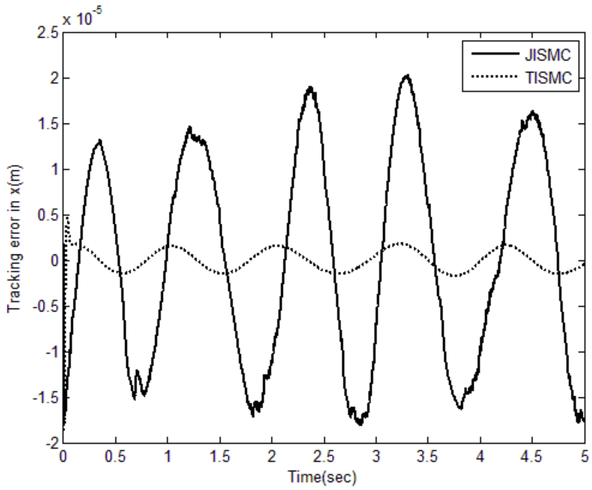

Fig. 15 Comparison of x direction tracking performance of the task space ISMC and joint/task space hybrid implementations

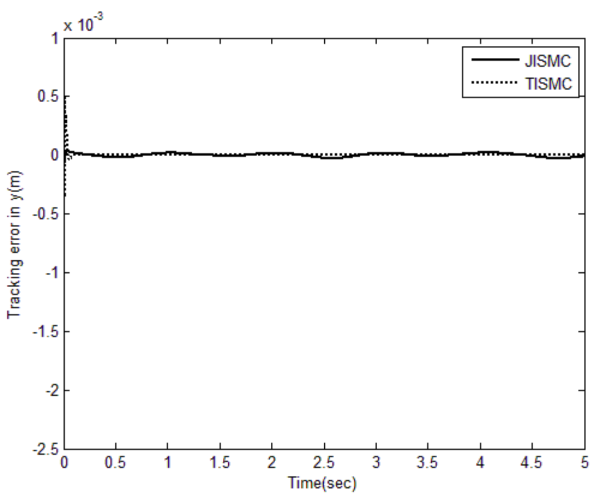

Fig. 16 Comparison of y direction tracking performance of the task space ISMC and joint/task space hybrid implementations

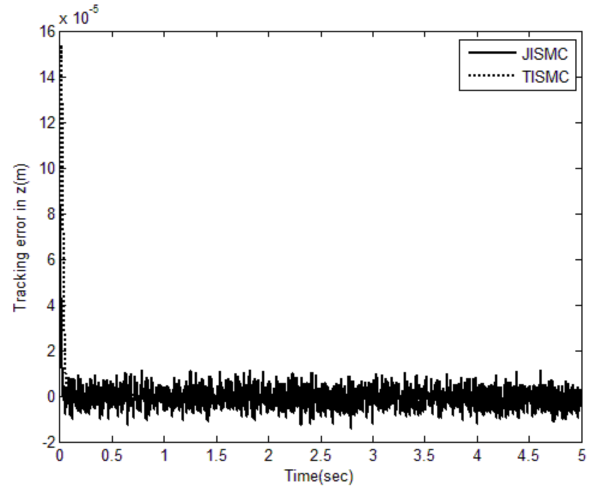

Fig. 17 Comparison of z direction tracking performance of the task space ISMC and joint/task space hybrid implementations

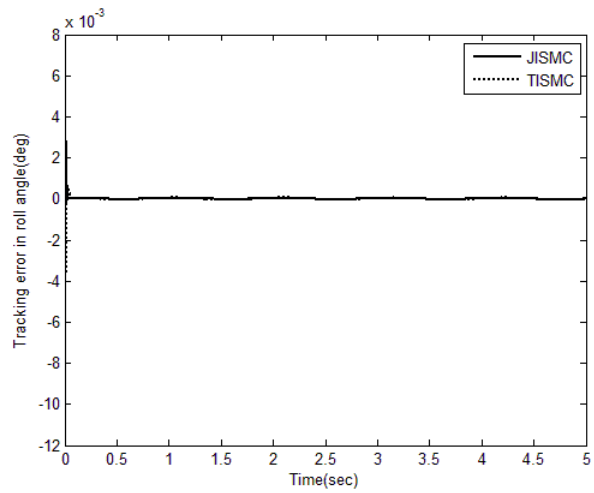

Fig. 18 Comparison of roll angle tracking performance of the task space ISMC and joint/task space hybrid implementations

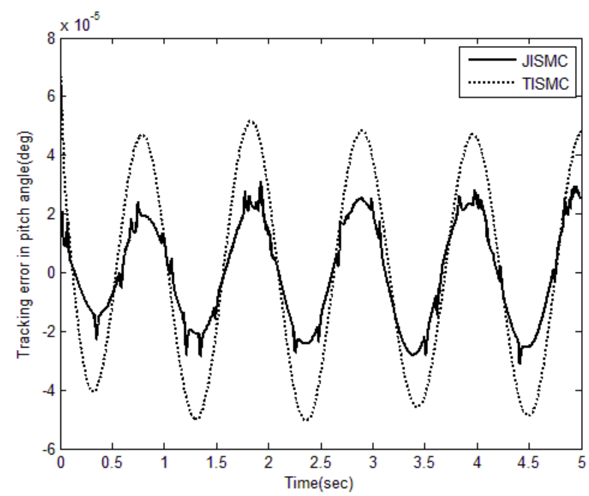

Fig. 19 Comparison of pitch angle tracking performance of the task space ISMC and joint/task space hybrid implementations

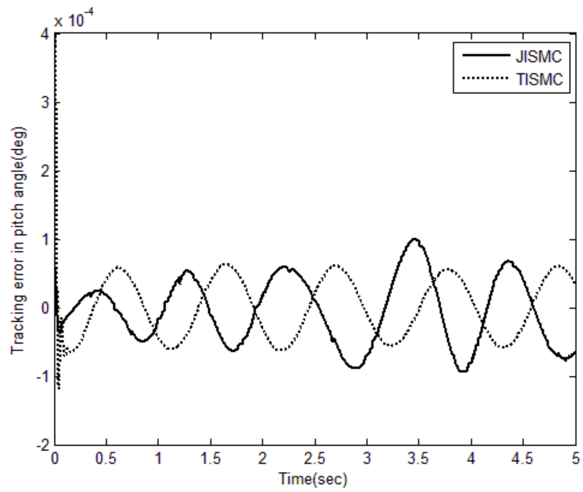

Fig. 20 Comparison of yaw angle tracking performance of the task space ISMC and joint/task space hybrid implementations

REFERENCES

[1] Y. Xu, W. Tang and Z. Yuan "Control design for uncertain system based on integral-type sliding surface," in Proc. Of 6th WSEAS int. conf. on Robotics, control and manufacturing technology, Hangzou, China, pp.99-102, April 16-18, 2006

[2] YM Sam, NM Suib and JHS Osman "Proportional Integral Sliding Mode Control for the Half-Car Active Suspension System with Hydraulic Actuator," in proc of 8th WSEAS Int. Conf. on ROBOTICS, CONTROL and MANUFACTURING TECHNOLOGY (ROCOM '08), Hangzhou, China, April 6-8, 2008, pp. 52-57

[3] Fernando Castaños and Leonid Fridman, "Analysis and Design of Integral Sliding Manifolds for Systems With Unmatched Perturbations," IEEE Transactions on Automatic Control, vol. 51, no. 5, pp. 853-858, May 2006

[4] Han Ho Choi, "LMI-Based Sliding Surface Design for Integral Sliding Mode Control of Mismatched Uncertain Systems", IEEE Transactions on Automatic Control, vol. 52, no. 4, pp. 736-742, April 2007

[5] Jian-Xin Xu, Ya-Jun Pan and Tong-Heng Lee, "Analysis and Design of Integral Sliding Mode Control Based on Lyapunov's Direct Method," Proceedings of the American Control Conference, Denver, Colorado June 4-6, 2003

[6] A. Ghobakhloo, M. Eghtesad and M. Azadi, "Position Control of Stewart-Gough Platform Using Inverse Dynamics with Full Dynamics," IEEE Conference 9th Int. Workshop on Advanced Motion Control (AMC'06), pp.50-55, Istanbul-Turkey 2006.

[7] S. Iqbal and A. I. Bhatti, "Robust Sliding-Mode Controller Design for a Stewart Platform, " Proceedings of International Bhurban Conference on Applied Sciences & Technology, Islamabad, Pakistan, 8th- 11th January, pp. 155-160, 2007

[8] G. Fried, K. Djouani, D. Borojeni and S. Iqbal , "An inverse dynamic model of the Gough-Stewart platform," WSEAS tran. On systems, vol. 2, no. 7, pp. 88-99, Feb. 2008

[9] Chin-I Huang, and Li-Chen Fu, "Smooth Sliding Mode Tracking Control of the Stewart Platform" Proceedings of the 2005 IEEE Conference on Control Applications, Toronto, Canada, pp. 43-48, August 28-31, 2005

[10] Hag Seong Kima, Young Man Chob and Kyo-II Lee, "Robust nonlinear task space control for 6 DOF parallel manipulator," Automatica vol. 41, pp. 1591-1600, 2005

[11] Dereje S. and Mitra R., "Neuro-Fuzzy Sliding Mode Control: Design and Stability Analysis", International Journal of Computational Intelligence Studies, Inder Science publishers, in press.

[12] E. M. Jafarov, "Robust Sliding Mode Control of Multivariable Time-Delay Systems," Proceedings of the 11th WSEAS International Conference on Automatic Control, Modelling and Simulation, pp.430-437

[13] Dereje Shiferaw and R. Mitra, "Integral sliding mode control for trajectory tracking control of Stewart platform manipulator," in Proc. Int. Conf. Industrial and Information Systems, IEEE- ICIIS 2010, Surathkal, India, July 2010,pp. 650-654

[14] B. Badreddine, B. Faouzi and K. Maekki, "Neural networks and genetic algorithms for robust predictive control," in proc. 8th WSEAS int. conf. on Automatic control, modeling and simulation, Prague, Czech Rep

[15] S. N. Sivanandam and S.N.Deepa, Introduction to Genetic Algorithms, Springer-Verlag Berlin Heidelberg, 2008