

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Position Control and Trajectory Tracking of the Stewart Platform

Selçuk Kizir and Zafer Bingul

*Mechatronics Engineering, Kocaeli University
Turkey*

1. Introduction

Demand on high precision motion systems has been increasing in recent years. Since performance of today's many mechanical systems requires high stiffness, fast motion and accurate positioning capability, parallel manipulators have gained popularity. Currently, parallel robots have been widely used several areas of industry such as manufacturing, medicine and defense. Some of these areas: precision laser cutting, micro machining, machine tool technology, flight simulators, helicopter runway, throwing platform of missiles, surgical operations. Some examples are shown in Figure 1. Unlike open-chain serial robots, parallel manipulators are composed of closed kinematic chain. There exist several parallel kinematic chains between base platform and end moving platform. Serial robots consist of a number of rigid links connected in serial so every actuator supports the weight of the successor links. This serial structure suffers from several disadvantages such as low precision, poor force exertion capability and low payload-to-weight-ratio. The parallel robot architecture eliminates these disadvantages. In this architecture, the load is shared by several parallel kinematic chains. This superior architecture provides high rigidity, high payload-to-weight-ratio, high positioning accuracy, low inertia of moving parts and a simpler solution of the inverse kinematics equations over the serial ones. Since high accuracy of parallel robots stems from load sharing of each actuator, there are no cumulative joint errors and deflections in the links. Under heavy loads, serial robots cannot perform precision positioning and oscillate at high-speeds. Positioning accuracy of parallel robots is high because the positioning error of the platform cannot exceed the average error of the legs positions. They can provide nanometer-level motion performance. But they have smaller workspace and singularities in their workspace.

The most widely used structure of a parallel robot is the Stewart platform (SP). It is a six degrees of freedom (DOF) positioning system that consists of a top plate (moving platform), a base plate (fixed base), and six extensible legs connecting the top plate to the bottom plate.

SP was invented as a flight simulator by Stewart in 1965 (Stewart, 1965). This platform contained three parallel linear actuators. Gough had previously suggested a tire test machine similar to Stewart's model (Bonev, 2003). In the test machine, six actuators were used as a mechanism driven in parallel. Gough, the first person, developed and utilized this type parallel structure. Therefore, SP is sometimes named as Stewart-Gough platform in the literature. Stewart's and Gough's original designs are shown in Figure 2.

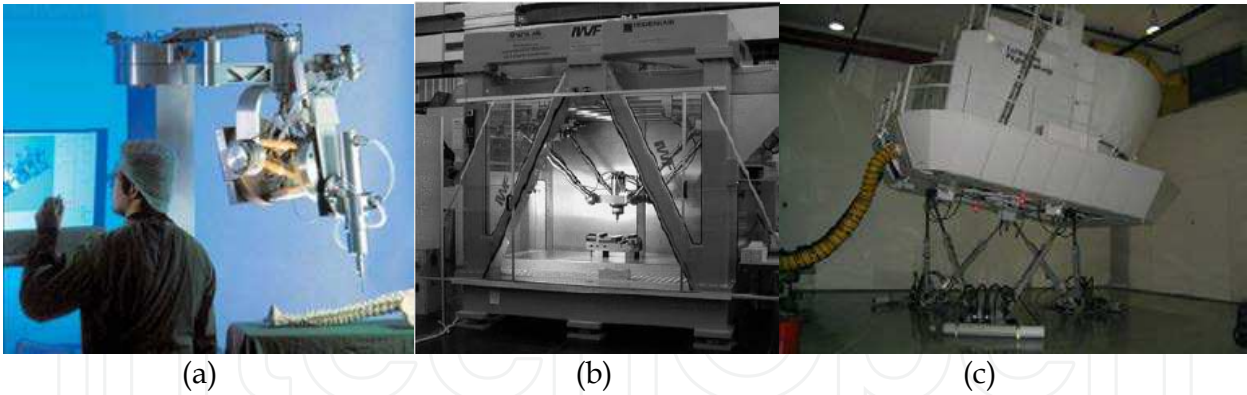


Fig. 1. Applications of the Stewart Platform: medical, manufacturing and flight simulator (Niesing, 2001; Merlet, 2006; Wikipedia)

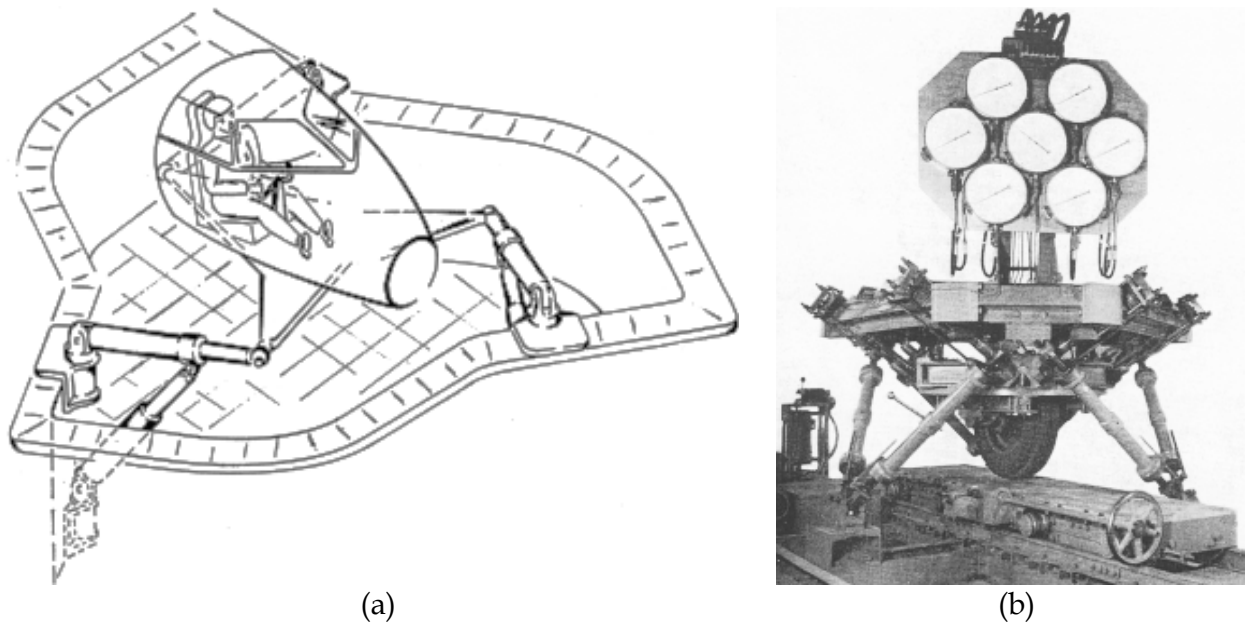


Fig. 2. Stewart (a) and Gough (b) original design (Bonev, 2003)

SP was not attracted attention during the first 15 years since the first invention. Then, Hunt indicated the advantages of parallel robots. After 1983, researchers realized their high load carrying capacity and high positioning ability of these robots. Researchers were then started to study a detailed analysis of these structures. The widely used structure of SP, where top platform is connected to base platform using 6 linear axis with universal joints, was then developed (Hunt, 1983).

It is a well known fact that the solution of the forward kinematics problem is easier than the inverse kinematics problem for serial robot manipulators. On the other hand, this situation is the just opposite for a parallel robot. Inverse kinematics problem of parallel robot can be expressed as follows: position vector and rotation matrix in Cartesian space is given, and asked to find length of each link in joint space. It is relatively easy to find the link lengths because the position of the connecting points and the position and orientation of the moving platform is known. On the other hand, in the forward kinematics problem, the rotation matrix and position vector of the moving platform is computed with given the link lengths. Forward kinematic of the SP is very difficult problem since it requires the solution of many

non-linear equations. In the literature, solutions of the forward (Chen & Song, 1994; Liao et al., 1993; Merlet, 1992; Nauna et al., 1990) and the inverse (Fitcher, 1986; Kim & Chung, 1999; Sefrioui & Gosselin, 1993) kinematics has been given in detail (Kizir et al., 2011).

In this study, design and development stages were given about position control and trajectory tracking of a 6 DOF-Stewart platform using Matlab/Simulink® and DS1103 real time controller. Matlab® (Mathworks Inc.) is a well known and one of the most popular technical computing software package that it is used in a wide area of applications from financial analysis to control designs. Matlab/Simulink® allows easiest way of programming and technical computing to its users. It enables simulations and real time applications of various systems. Third party co-developers improve its abilities allowing using hundreds of hardware. Dspace® company is one of the third party participate of Matlab® that produces rapid control prototyping and hardware-in-the-loop simulation units. DS1103 is a powerful real time controller board for rapid control prototyping (Dspace Inc.).

This chapter is organized in the following manner. System components and real-time controller board are introduced in section 2 and 3, respectively. Position and trajectory tracking control with PID and sliding mode controllers are described in section 4. Finally, experimental results are given in detail.

2. Stewart platform system

The system components are two main bodies (top and base plates), six linear motors, controller, space mouse, accelerometer, gyroscope, laser interferometer, force/torque sensor, power supply, emergency stop circuit and interface board. They are shown in Figure 3.



Fig. 3. Stewart platform system

A simple emergency stop circuit was designed to protect the motors, when they move to out of the limits. This circuit controls the power supply which gives the energy to the motors

based on the signal of hall-effect sensors on each motor. A switch-mode 150W power supply with inhibit input and EMI filter is used to supply required energy. Also, an interface board was designed between controller and motors.

The Dspace DS1103 real time controller is used to implement control algorithms. DS1103 is a rapid prototyping controller that developed for designing and analyzing complex and difficult control applications. It has various inputs and outputs such as digital, analog digital converter, digital analog converter, serial interface, can-bus, pulse width modulation (PWM) channels and encoders in order to be used lots of peripheral unit like actuators and sensors. DS1103 has a real time interface (RTI) that allows fully programmable from the Simulink® block diagram environment. A dspace toolbox will be added to Simulink® after installing RTI, so it can be configured all I/O graphically by using RTI. You can implement your control and signal processing algorithms on the board quickly and easily. A general DS1103 controller board system is shown in Figure 4. It consists of a DS1103 controller card in expansion box, CLP1103 input-output connector and led panel, DS817 link card and a computer.



Fig. 4. A general DS1103 set-up

3. DSPACE tool box and design procedures

While Dspace offers various development tools, it is needed at least Control Desk and RTI block set software packages in order to develop projects under Simulink. After installing software, a tool box shown in Figure 5 is added to Simulink®. It can be also opened dspace library shown in Figure 5 typing 'rti' in the matlab command window. Block sets in the library are divided into two categories: master processor and slave dsp. While master have blocks such as ADC, serial, encoder, digital I/O, slave has such as PWM, ADC and digital I/O.

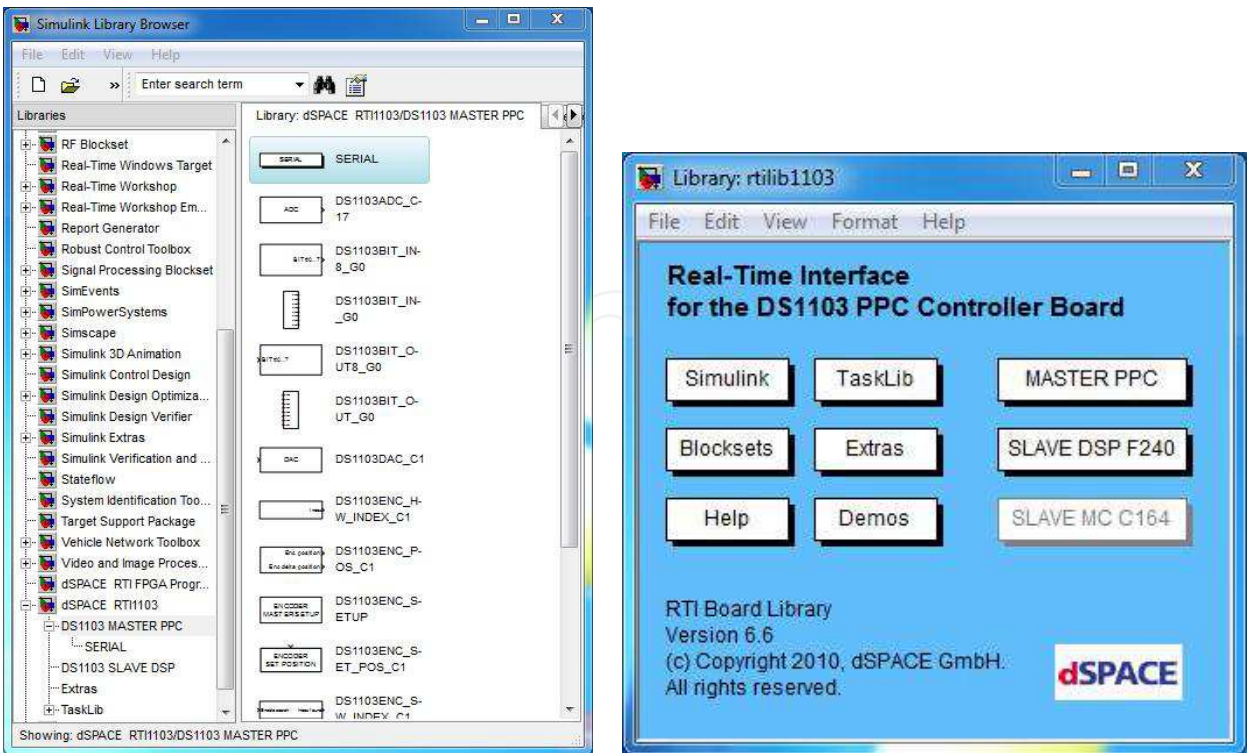


Fig. 5. Dspace toolbox and library

Another software component is the control desk (interface is shown in Figure 6) which allows downloading applications, doing experiments, easily creating graphical user interface and data acquisition.

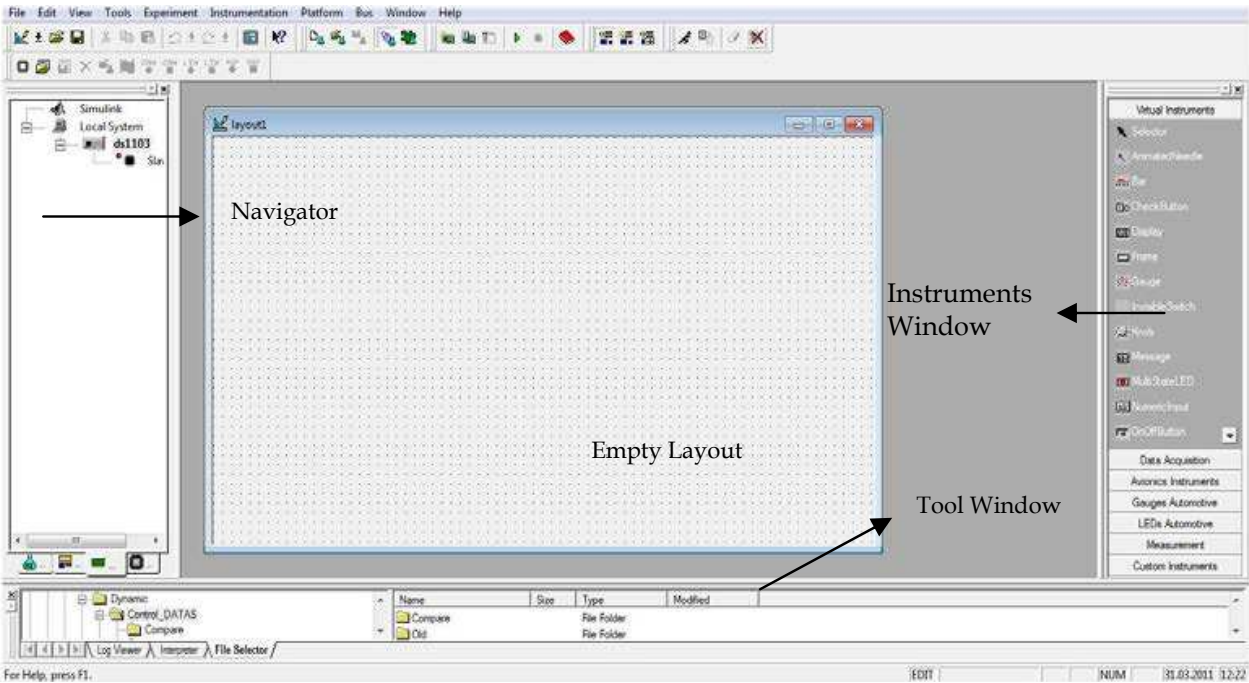


Fig. 6. Control desk interface

As can be seen from Figure 6, panel on the left side is called “Navigator” and it has four tabs: experiment, instrumentation, platform and test automation. All files written for

conducting experiment are listed in the experiment tab. Instrumentation tab allows building instrument panels in order to change and monitor the variables of a model. Supported simulations and connected boards are shown in platform tab. Test automation tab has functions about automation tasks, other software solution of Dspace.

Bottom side is called “Tool Window” having log viewer, file selector, interpreter and open experiment tabs. It is seen errors and warnings in the log viewer tab and files under selected folder are listed where an application can be loaded by drag and drop action.

In order to create a GUI for an experiment, it should be opened an empty layout from file-new layout click. A lot of instruments are listed in the instrument selector right side on the control desk. Virtual instruments and data acquisition elements are shown in Figure 7 below. An instrument can be placed on the layout plotting with mouse left clicked after selecting from virtual instruments panel. Its position and size can be changed and its properties such as color, text, precision and etc can be settled according to needs. It can be saved and added to an experiment after completing GUI.

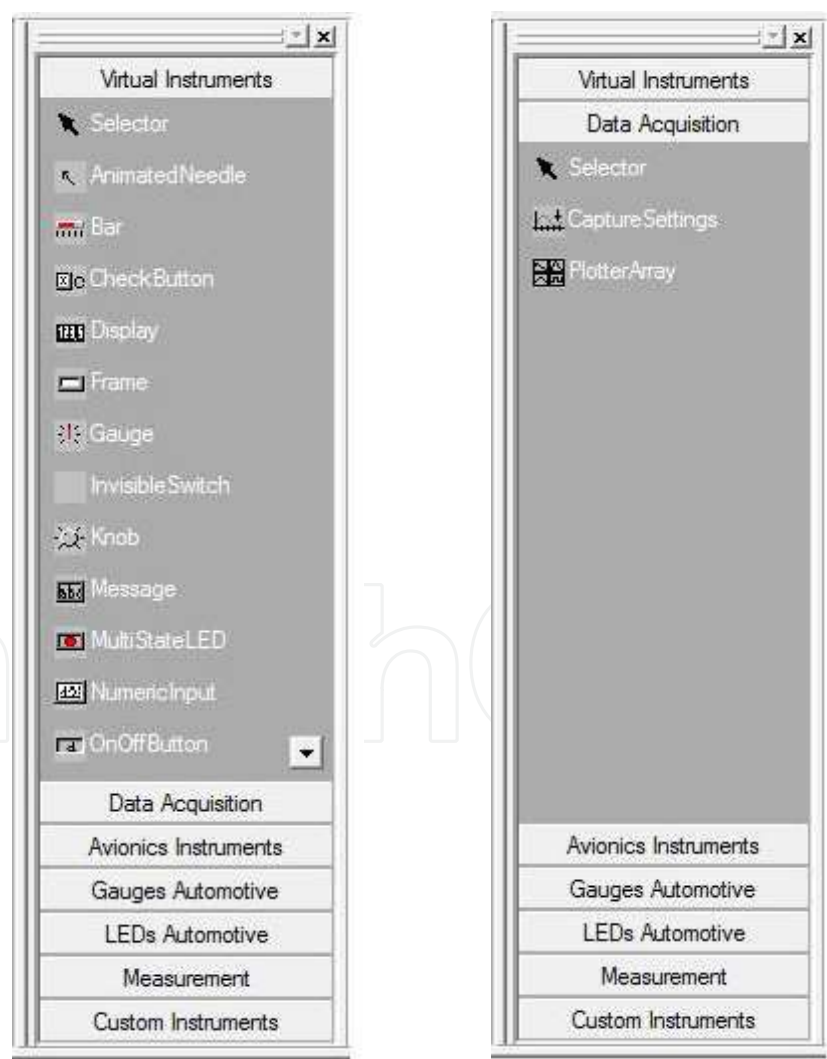


Fig. 7. Virtual instruments and data acquisition in the instruments selector window

All development steps can be illustrated basically in the Figure 8 below. This figure is illustrated for DS1103 in an expansion box. It should be finished all connections before this procedure.

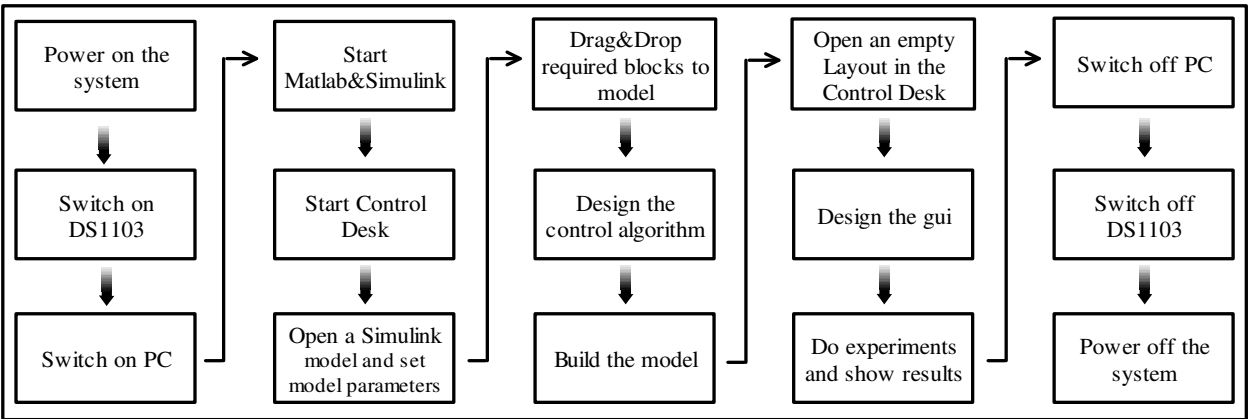


Fig. 8. Basic flow diagram for developing a project

4. Control

A controller is needed to move top platform from initial position to desired position and orientation. It will generate required forces for each motor. Position and trajectory control of the platform can be reduced to leg position control after inverse kinematic and path planning algorithms. A PID (proportional-integrator-derivative) and sliding mode position controllers were developed and implemented. Control algorithms designed in Simulink environment and embedded in the Dspace DS1103 real time controller.

All robots are electro-mechanic devices consisting of actuators, sensors and mechanical structure. In order to control the robots for desired motions kinematic and dynamic equations of the system should be known. Firstly kinematic solution should be computed before controller design. A schematic model of the SP for kinematic solution is illustrated in Figure 9. In the figure, base $B=\{X,Y,Z\}$ and top $T=\{x,y,z\}$ coordinate systems are placed and base and top joint points are labeled as B_i ($i=1,2, \dots 6$) and T_i ($i=1,2, \dots 6$).

It is needed to find leg lengths to reach the moving platform to its desired position and orientation according to fixed platform (inverse kinematics). Required leg vectors (L_i) for given position vector P and orientation matrix R are obtained by using the following equation. Finally, norm of the vectors (L_i) are leg lengths (l_i) (Fitcher, 1986; Kim & Chung, 1999; Sefrioui & Gosselin, 1993).

$$L_i = R_{XYZ}T_i + P - B_i \quad i : 1,2,\dots,6$$

(1)

In order to have T_i and B_i position vectors based on robot structure, an m-file is written and a Simulink model is designed to obtain inverse kinematic solution by using Equation 1. The model shown in Figure 10 uses the m-file to get required variables and takes the desired position (x, y, z) and orientation (φ, θ, ψ) of the top platform. It outputs the leg lengths. Desired block in this model is shown in Figure 11. The reference inputs can be entered in this block.

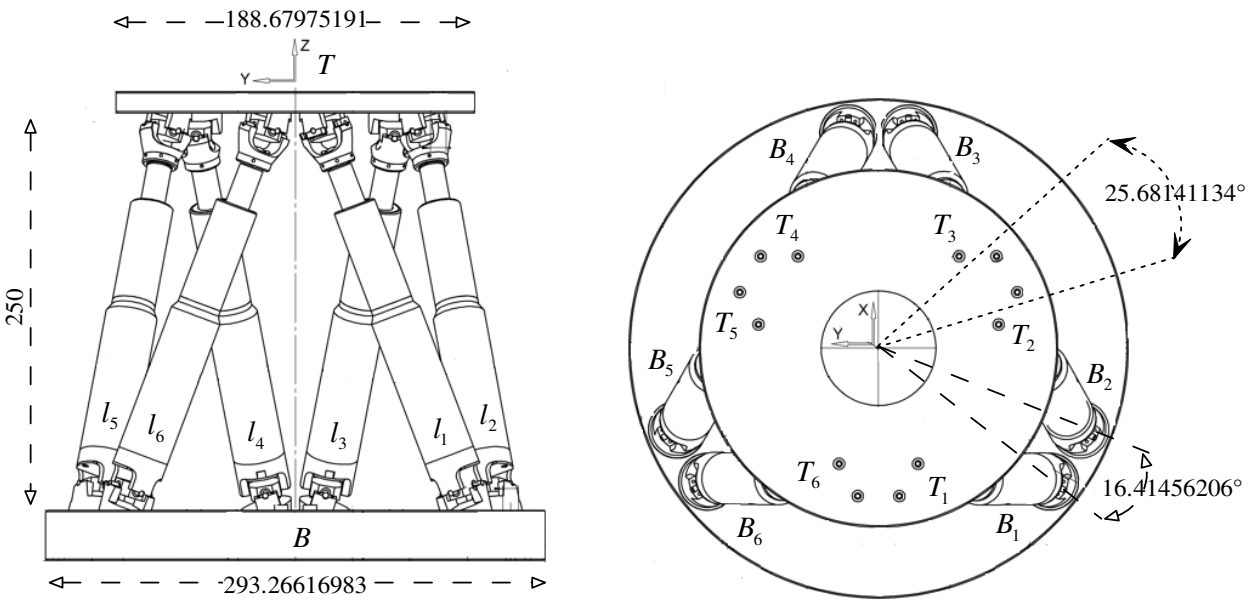


Fig. 9. Schematic diagram of the Stewart platform

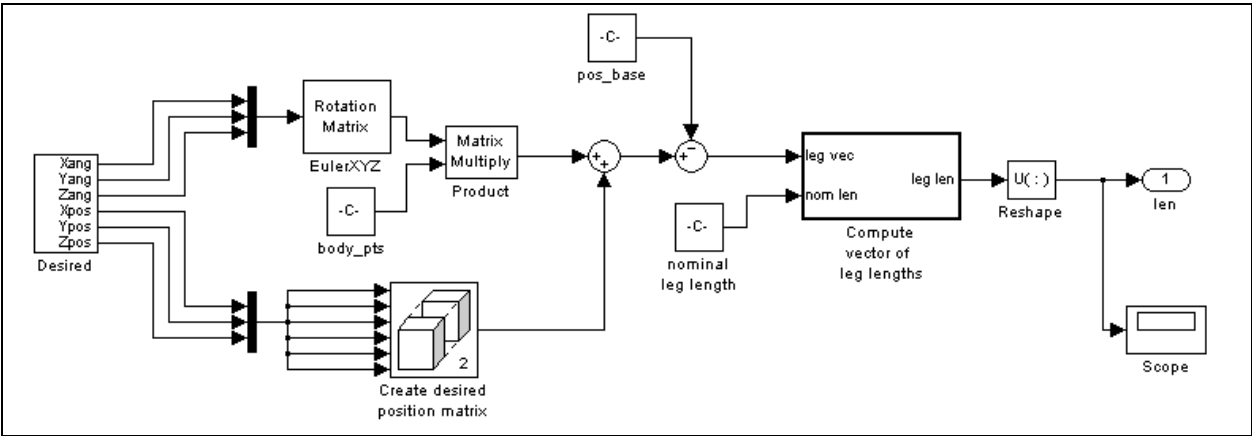


Fig. 10. Simulink model for inverse kinematic solution

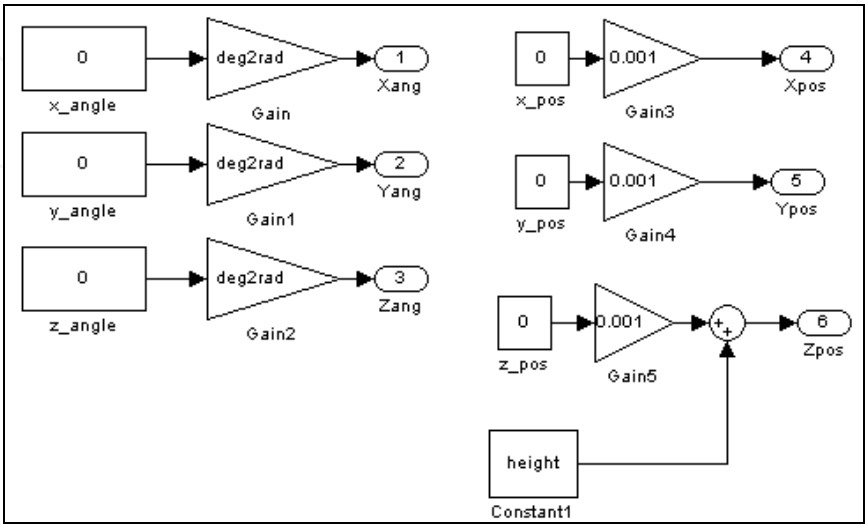


Fig. 11. A subsystem for desired references

4.1 Leg model

The leg system is basically composed of dc motor, precision linear bearing & ball screw and coupling elements. Dc motor model is given below (Küçük & Bingül, 2008).

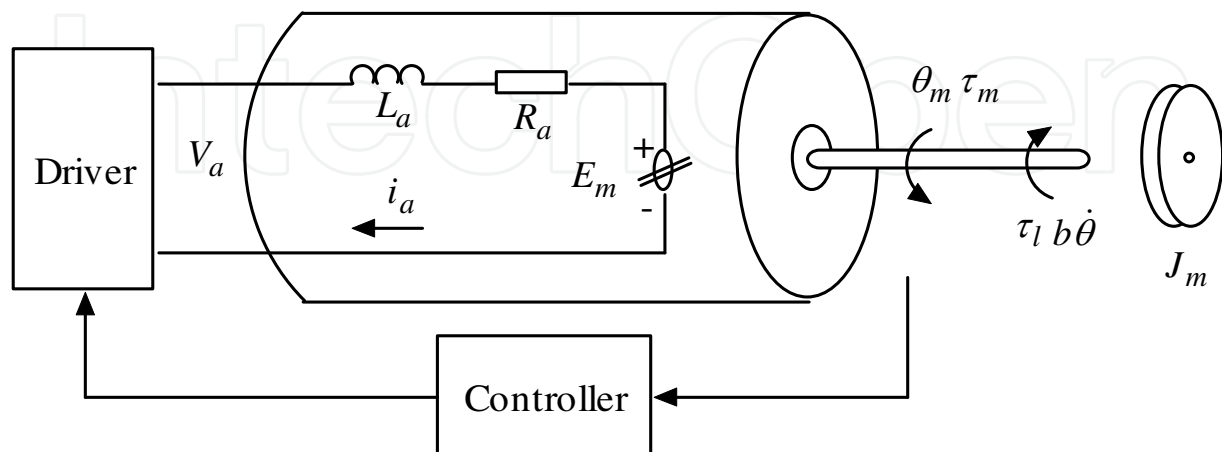


Fig. 12. DC motor model

The symbols represent the following variables here; θ_m is the motor position (*radian*), τ_m is the produced torque by the motor (*Nm*), τ_l is the load torque, V_a is the armature voltage (*V*), L_a is the armature inductance (*H*), R_a is the armature resistance (Ω), E_m is the reverse EMF (*V*), i_a is the armature current (*A*), K_b is the reverse EMF constant, K_m is the torque constant.

$$L_a \frac{di_a}{dt} + R_a i_a = V_a - E_m \quad (2)$$

$$E_m = K_b \frac{d\theta_m}{dt} \quad (3)$$

$$\tau_m = K_m i_a \quad (4)$$

$$\tau_m - \tau_l = J_m \frac{d^2\theta_m}{dt^2} \quad (5)$$

4.2 Startup algorithm

Before controller design a startup algorithm is needed to get robot to its home position. The position of each motor is controlled after startup. Motors have incremental encoders therefore firstly they must be brought their zero or home position. When SP system is energized, an index search algorithm looks what the position of the each leg is. The algorithm simply searches index signal of the leg and when it is found encoders are reset by hardware. Designed Simulink model for this purpose is illustrated in Figure 13. Movements to the home position for possible two situations (from upper and lower sides to zero) are shown in Figure 14.

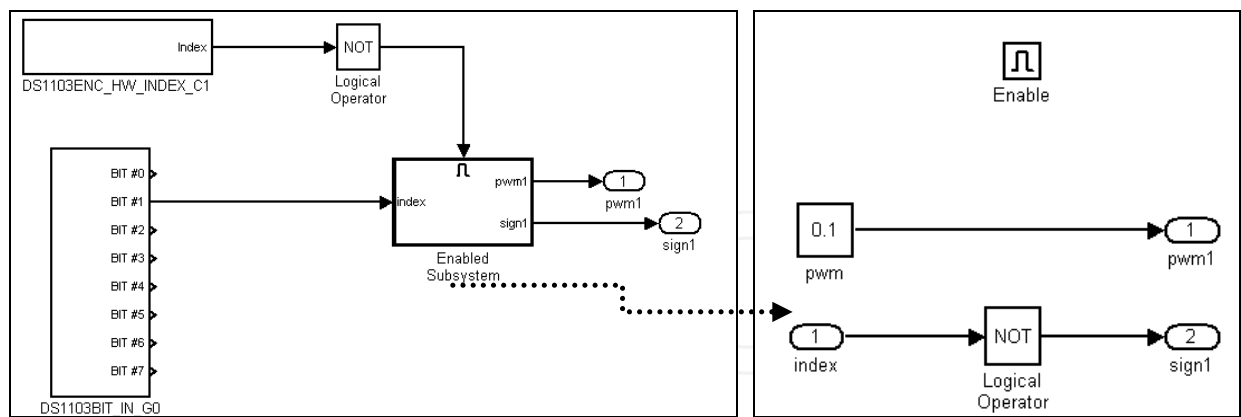


Fig. 13. Simulink model for initialization algorithm

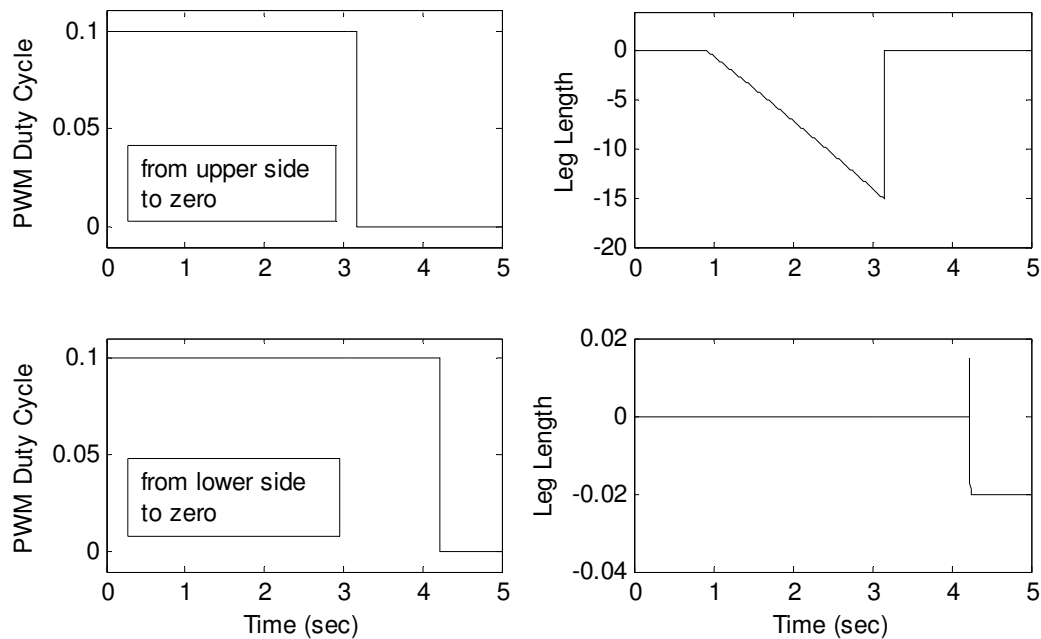


Fig. 14. Initialization routine

4.3 Trajectory generation

For step inputs, the leg lengths obtained from inverse kinematics solution input to independent position control system for each motor. This movement is defined in the joint space. In order to move robot along a straight line, a trajectory planning algorithm is developed. Thus, it can be determined start and stop times of the motion besides desired position and orientation inputs. Also, motors are synchronized each other during the motion. If classical polynomial-based trajectory equations are examined, the following deficiencies are determined: i) there are need many initial and finish values in order to find the polynomial coefficients ii) the acceleration values, especially initially require high levels, iii) the coefficients need to be calculated again each time when the conditions changes. Kane's transition function is used to resolve these shortcomings and it is given the following equation (Reckdahl, 1996).

$$y(t) = y_0 + (y_f - y_0) \frac{t - t_0}{t_f - t_0} - \frac{y_f - y_0}{2\pi} \sin\left(2\pi \frac{t - t_0}{t_f - t_0}\right) \quad (6)$$

where $y(t)$ is the position function, y_0 is the initial position, y_f is the finish position, t is time, t_0 is initial time and t_f is the finish time. An example of the trajectory generation is given below (Figure 15) using Equation 6. As can be seen from figure, position, velocity and acceleration curves are given for two seconds and velocity and acceleration start and finish with zero.

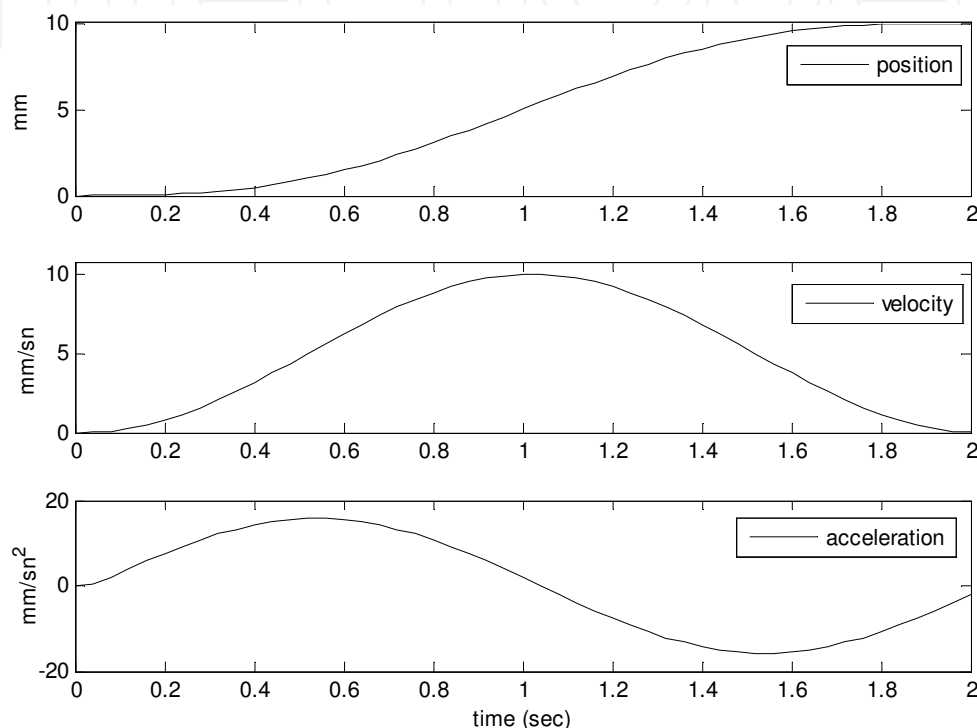


Fig. 15. An example of trajectory generation with Kane function

Equation 6 was embedded in Simulink block. Only end position and path period is given to this structure and other parameters are automatically calculated. This form of motion starts and finishes with zero velocity and zero acceleration. Position, velocity and acceleration values are quite soft changes. This is very important for motors to start and stop more softly. Main model of the trajectory planning with reference inputs is shown in Figure 16 below. Detail of the blocks named 'path' in the Figure 16 is shown in Figure 17 which implements the Equation 6.

4.4 PID controller

PID control is one of the classical control methods and widely used in the industrial applications. The difference between the set point and the actual output is represented by the $e(t)$ error signal. This signal is applied to a PID controller, control signal, $u(t)$ is as follows.

$$u(t) = K_p e(t) + K_I \int e(t) dt + K_D \frac{d}{dt} e(t) \quad (7)$$

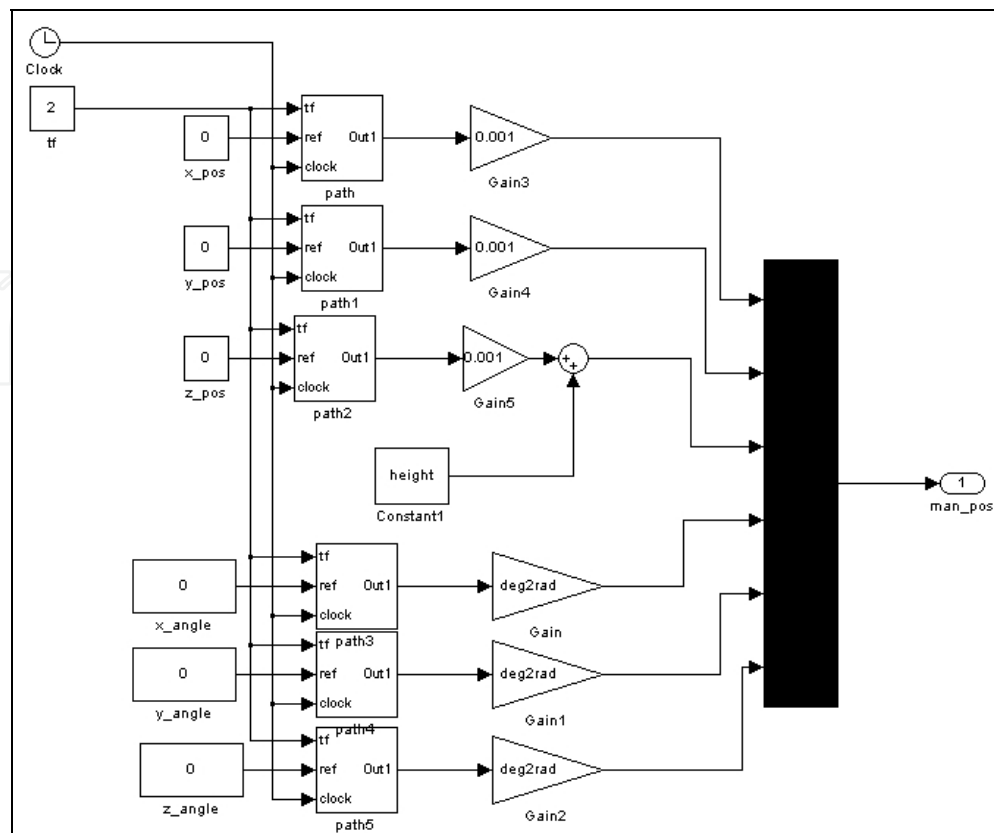


Fig. 16. Reference inputs with trajectory planning

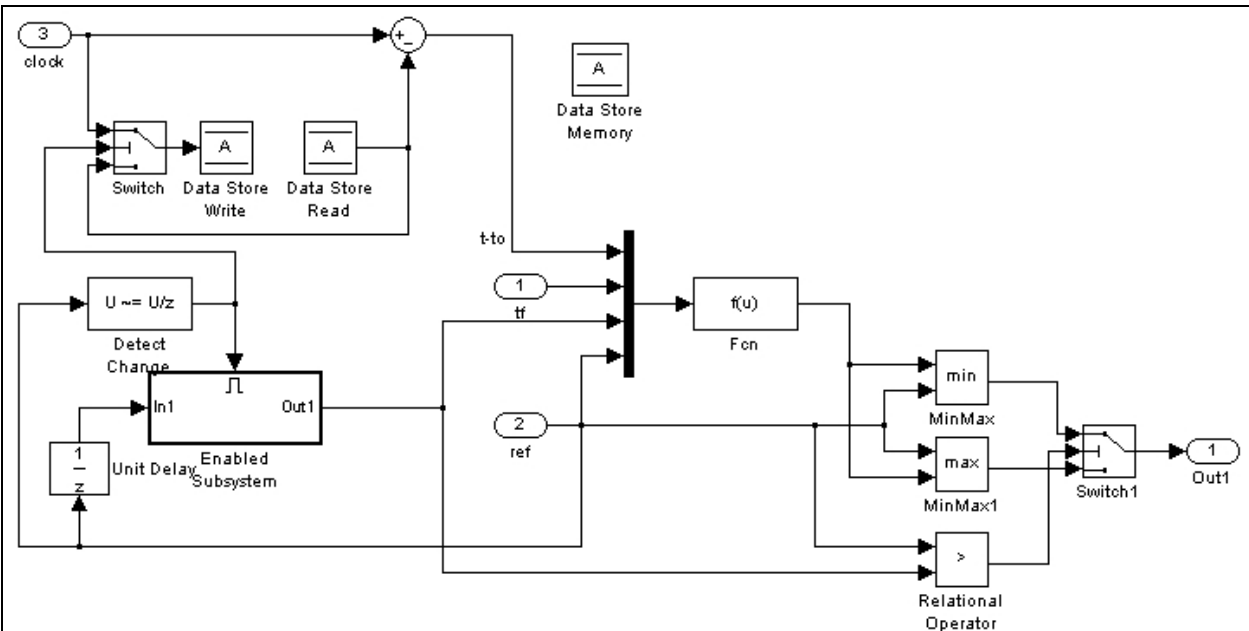


Fig. 17. Implementing Equation 6 in Simulink model

General control schema is summarized as follows: initialization, reference input, inverse kinematic, measurement, closed loop controller, input/outputs and additional blocks for safety reasons. Main PID Simulink model diagram is given Figure 18 below. The model

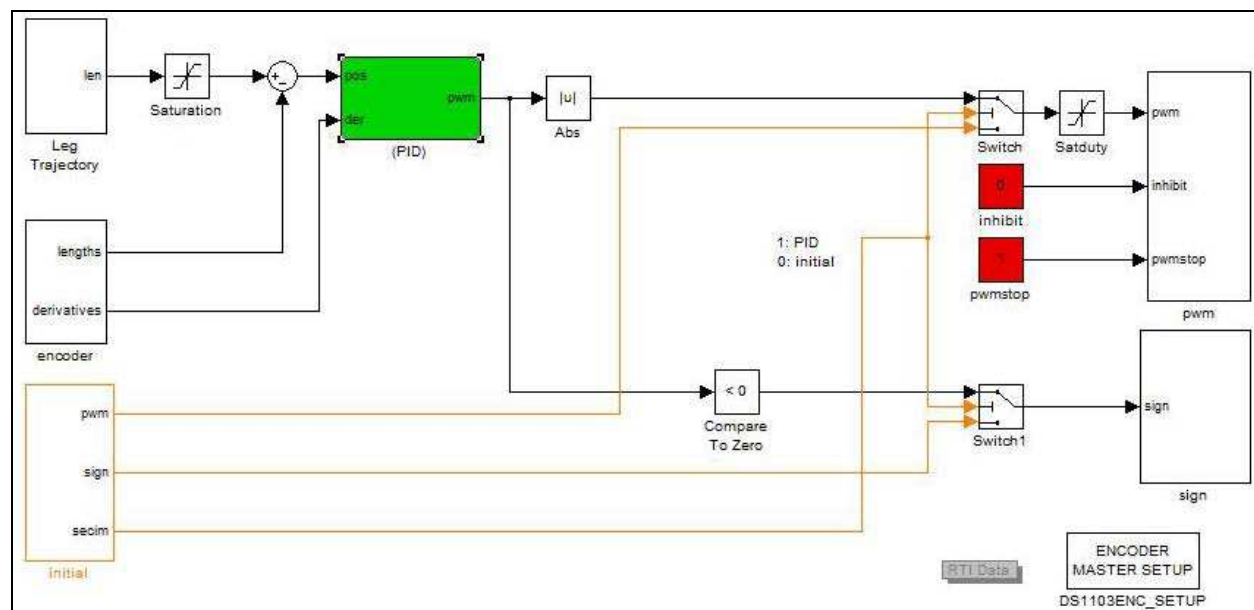


Fig. 18. Main PID controller Simulink model

contains some subsystems such as leg trajectory, encoder, initial, PID, pwm and sign. These subsystems perform the tasks mentioned above.

Project has a Control Desk interface for experiments and it is shown below. All system information can be entered through this interface. It contains variables that can be used in the development phase. Reference input values can be easily entered through the interface. Leg trajectory subsystem detail was given before, so it will be continued giving other blocks. Firstly, encoder subsystem given in Figure 20 is described. Each encoder is read using an encoder block and then pulses are converted to metric unit (mm) and scaled. Round subsystem shown in Figure 21 is used to eliminate errors less than 500 nanometer.

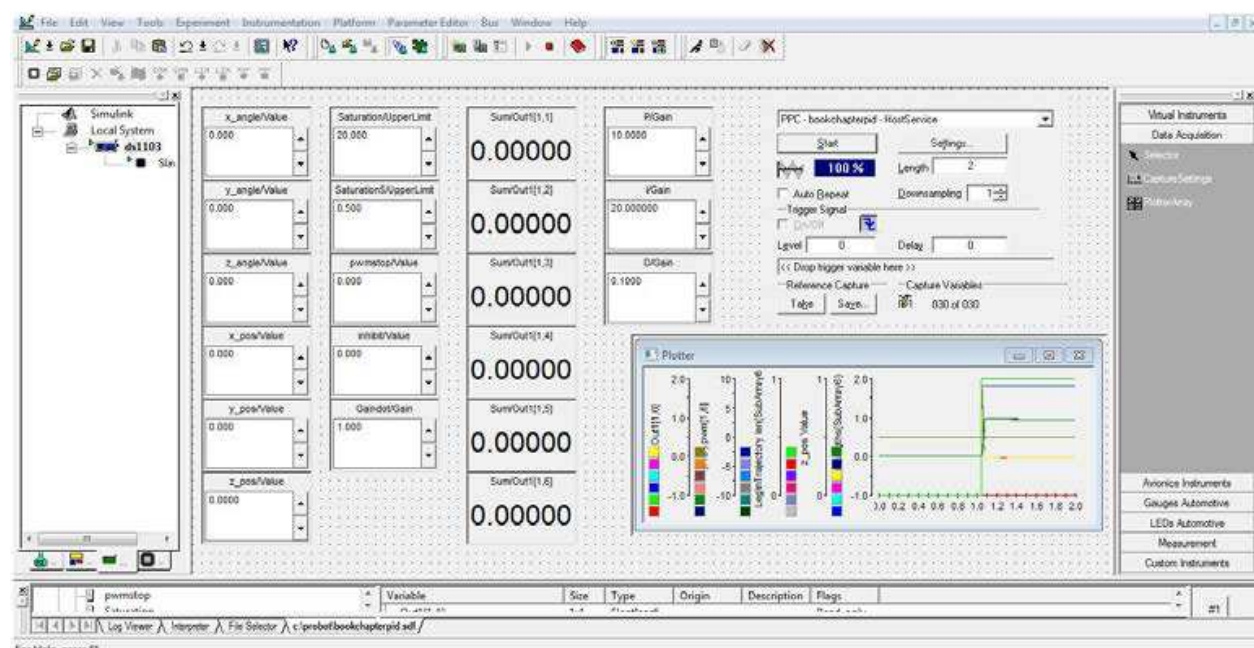


Fig. 19. Control Desk GUI for data acquisition and parameter update

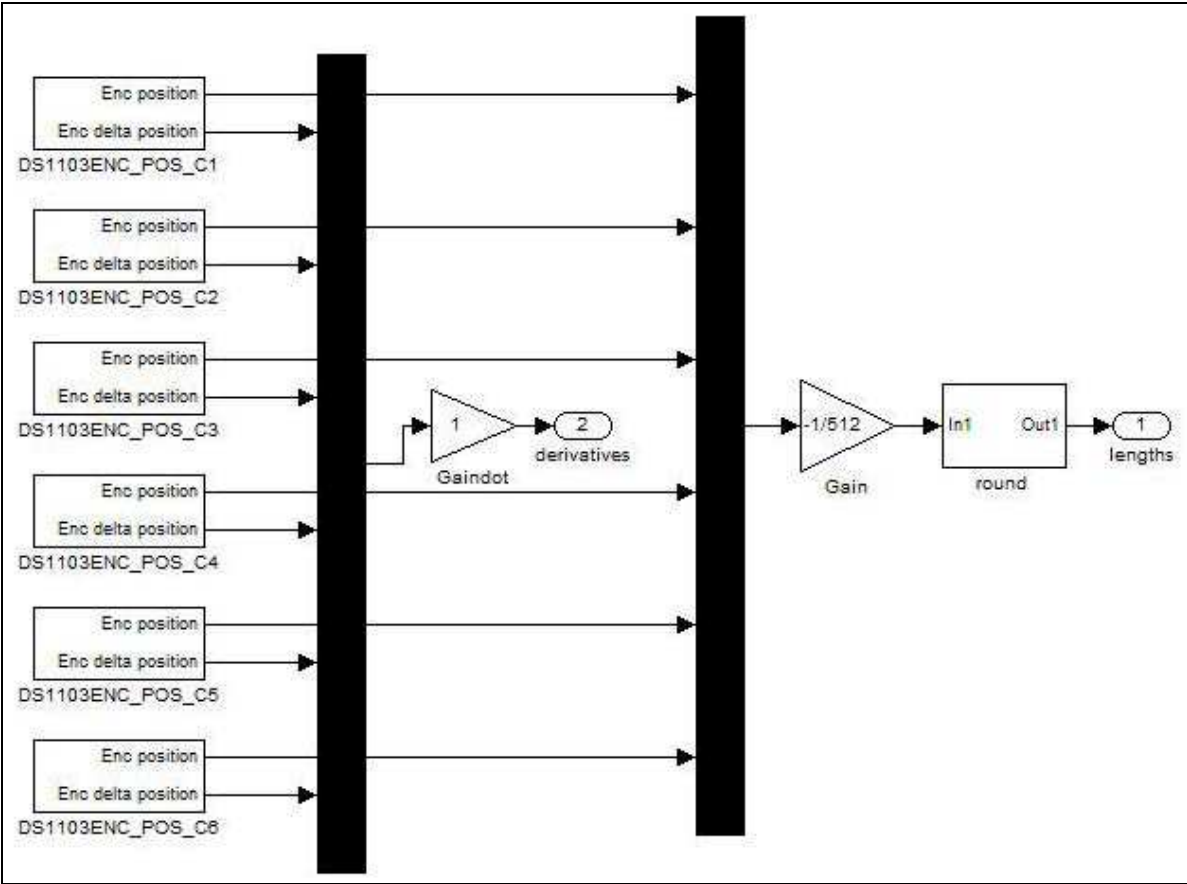


Fig. 20. Encoder subsystem

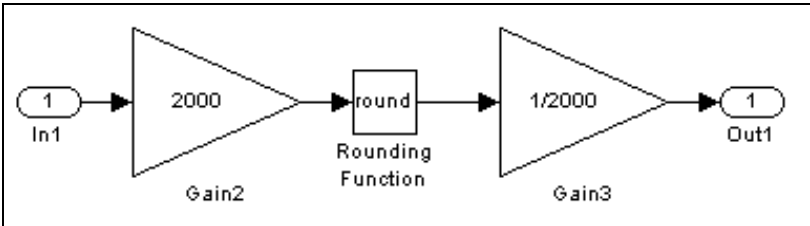


Fig. 21. Round subsystem

A PID controller is added for each leg. These subsystems are shown in figure below. It is a classic way of creating simple and efficient closed control loops.

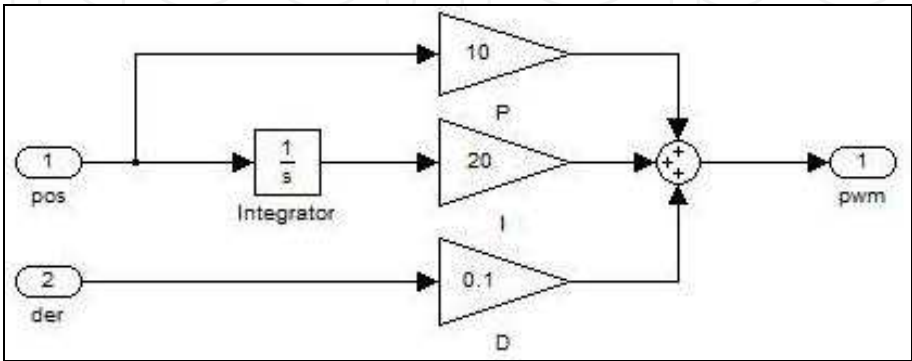


Fig. 22. PID subsystem in main model and simple PID structure

Initial subsystem model is shown in Figure 23 and its details were given before. It takes the index signals of the motors and produces a predefined duty cycle value for PWM generation, motor direction signals and status of the initialization routine.

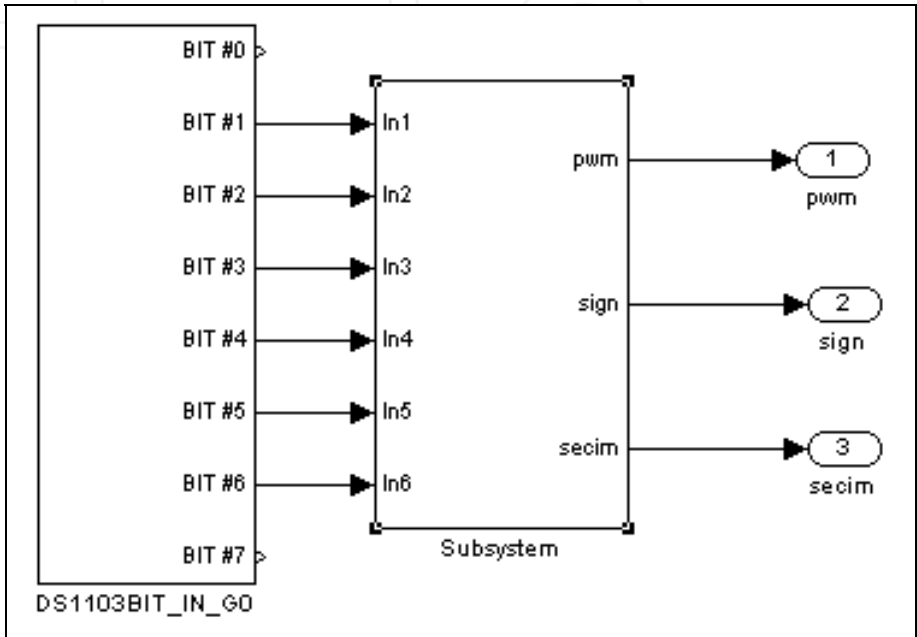


Fig. 23. Initial subsystem

Motors are controlled using 25 KHz PWM signals for amplitude and sign signals for motor directions according to controller outputs. In order to generate PWM signals slave dsp of the DS1103 is used. PWM block in the main model is given below. Saturation blocks are used to determine upper limit of the PWM duty cycles. In Figure 24, PWM and sign subsystems in the main model are shown.

4.4.1 Position control

After creating controller model and user interfaces, experiments can be done and results can be observed. Firstly, PID parameters were determined via trial-error experiments in order to obtain desired responses. A lot of experiments can be done simply and system responses can be taken easily. Output data can be saved ‘*.mat’ extension in order to analyze in detail in Matlab. Some real time responses are given below. For 1 mm motion along the z direction, responses of the legs with references and control outputs are shown in Figure 25. An orientation step response in x direction and a 500 nm step response in z direction were given in Figure 26. As can be seen from figures, PID parameters were selected to obtain fast rising time, no steady state error and smaller overshoot. PID controller works in a wide range between 500 nm to 100 mm with good control behavior.

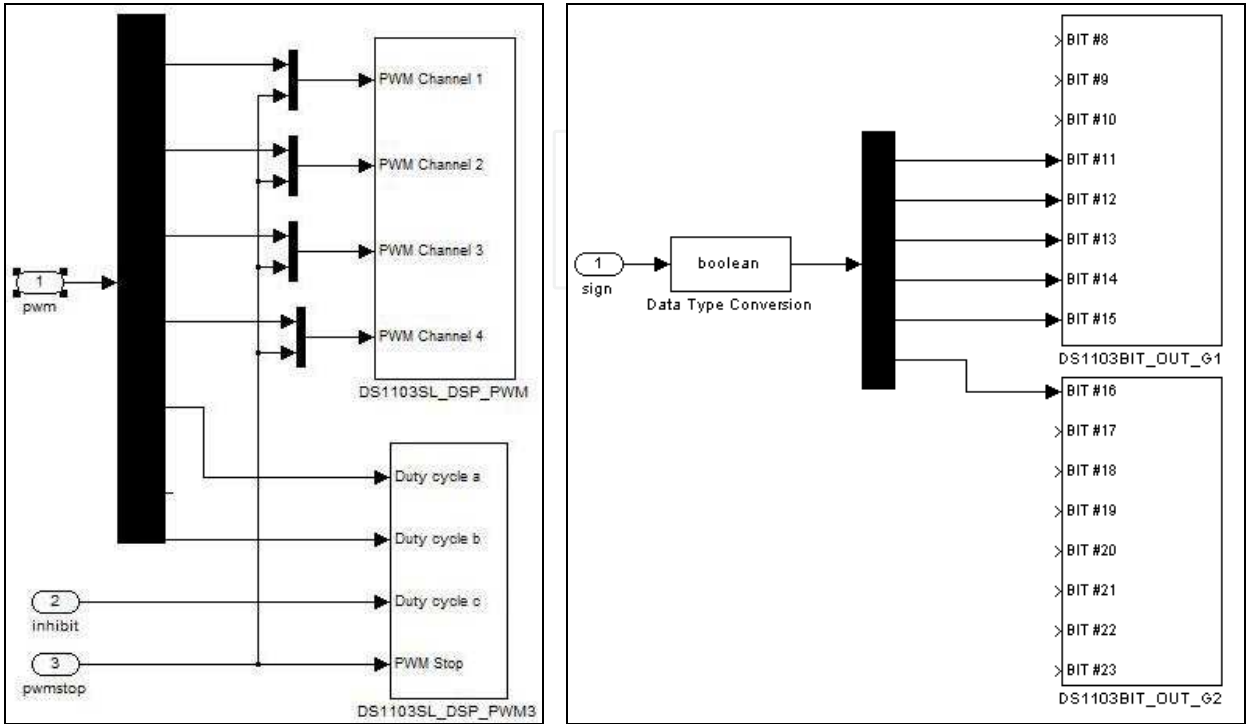


Fig. 24. PWM and Sign subsystems shown in the main model

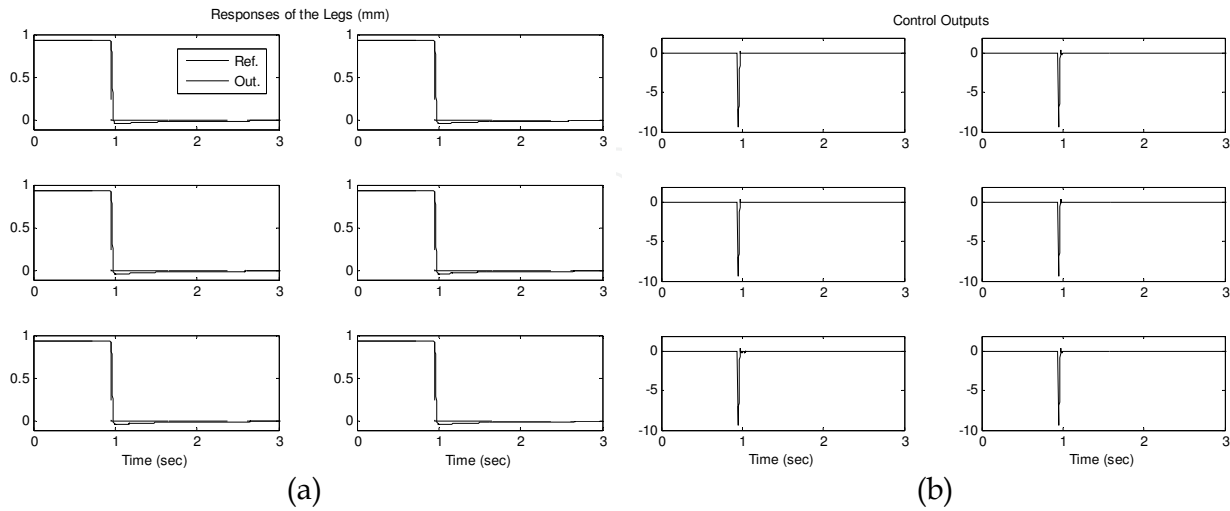


Fig. 25. (a) 1 mm step response of the top position in z direction (b) PID controller outputs

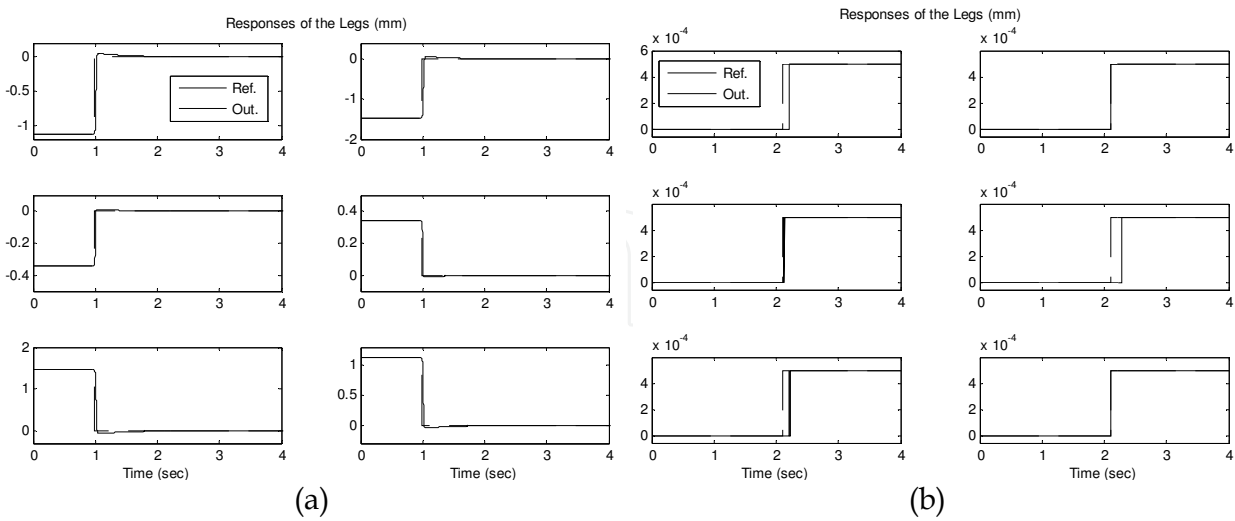


Fig. 26. (a) 1° rotation in x direction (b) 500 nm linear motion in z direction

4.4.2 Trajectory control

Several trajectory experiments were performed using trajectory generation algorithm mentioned in section 4.3. If motion time and end points are entered in to the user interface, the trajectory is created automatically and motion starts. In order to show the performance of trajectory tracking, some examples are given below. For these examples, references, trajectories and leg errors are illustrated in Figure 27-31. Several cases are examined in the results.

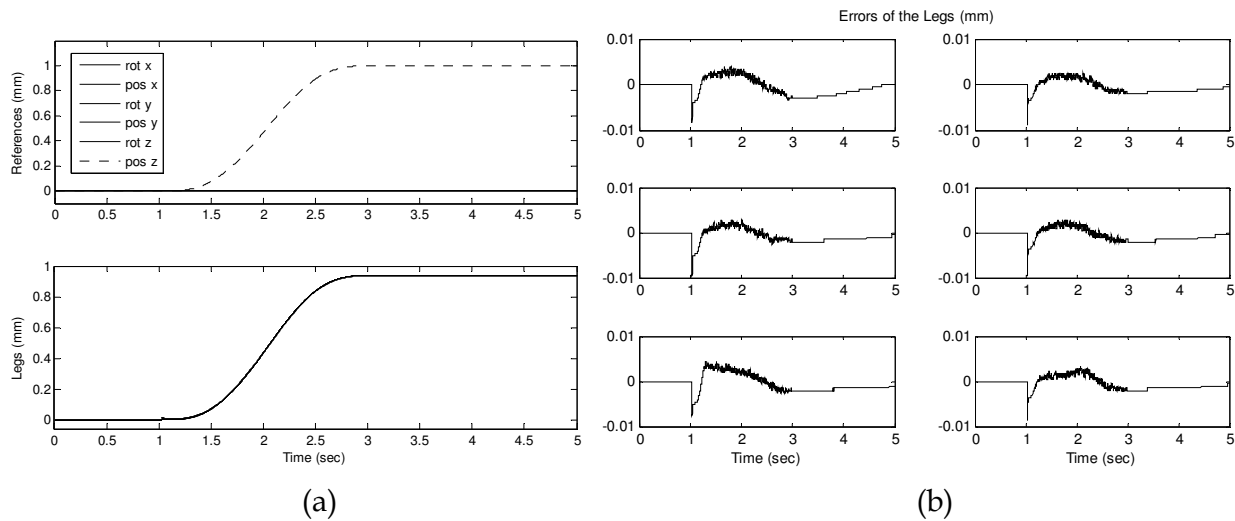


Fig. 27. (a) Trajectory tracking in z direction (b) Errors of the legs

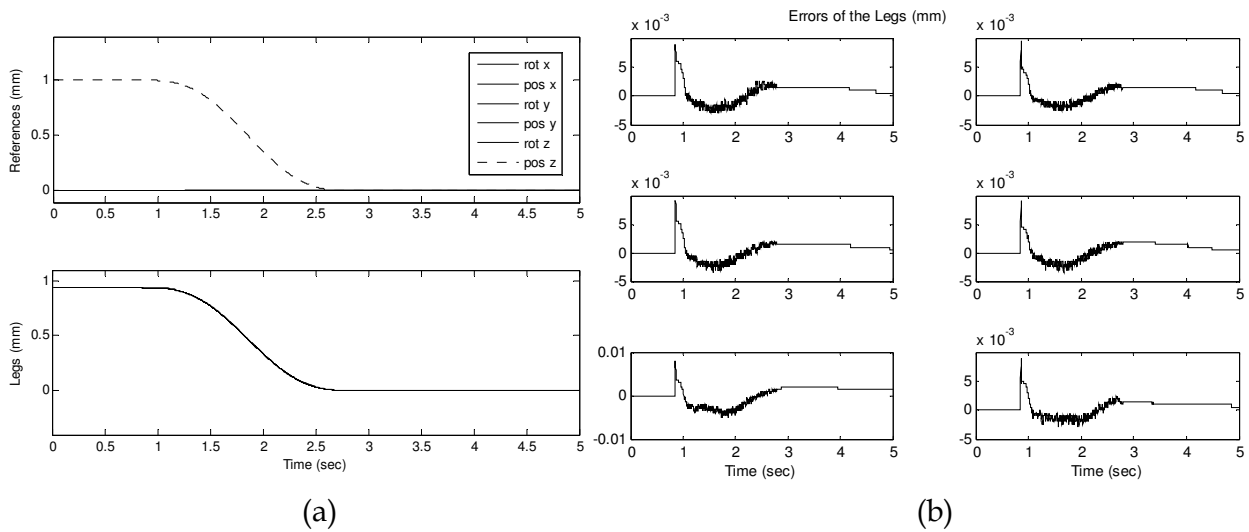


Fig. 28. (a) Trajectory tracking in z direction (b) Leg errors

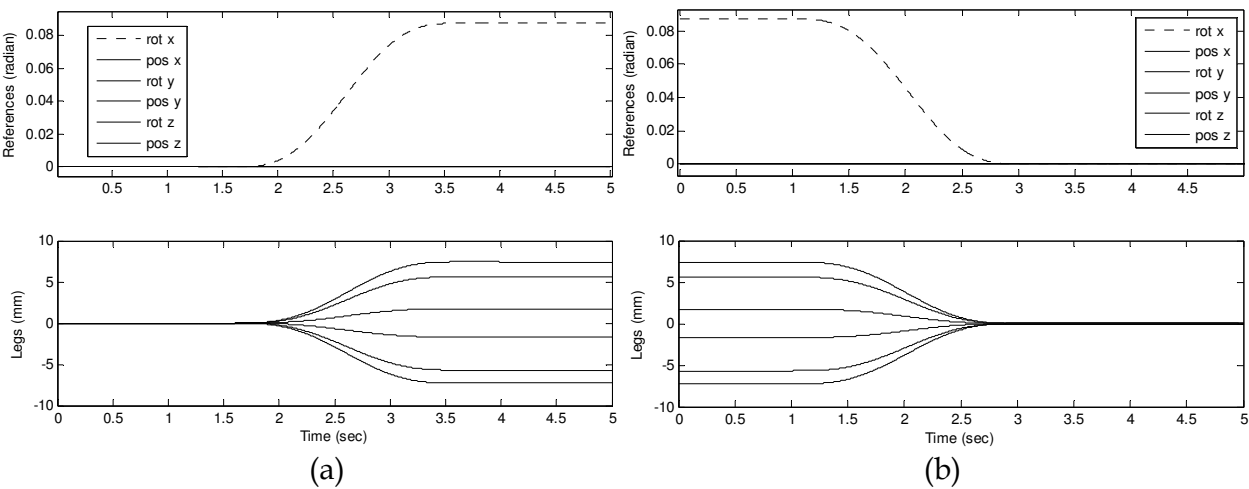


Fig. 29. (a) Rotation in x direction (b) Rotation in x direction

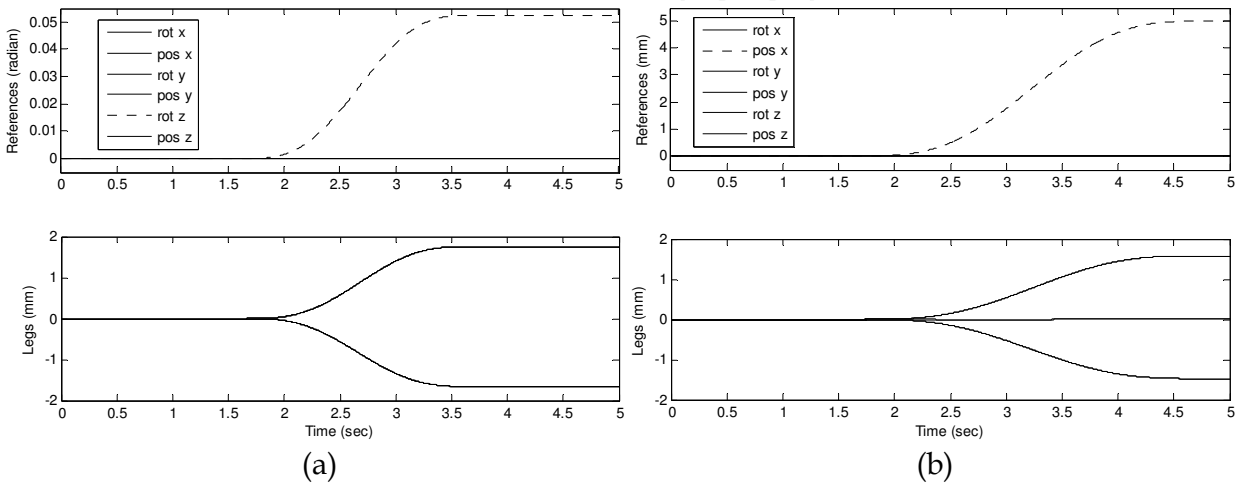


Fig. 30. (a) Rotation in z direction (b) Linear motion in x direction

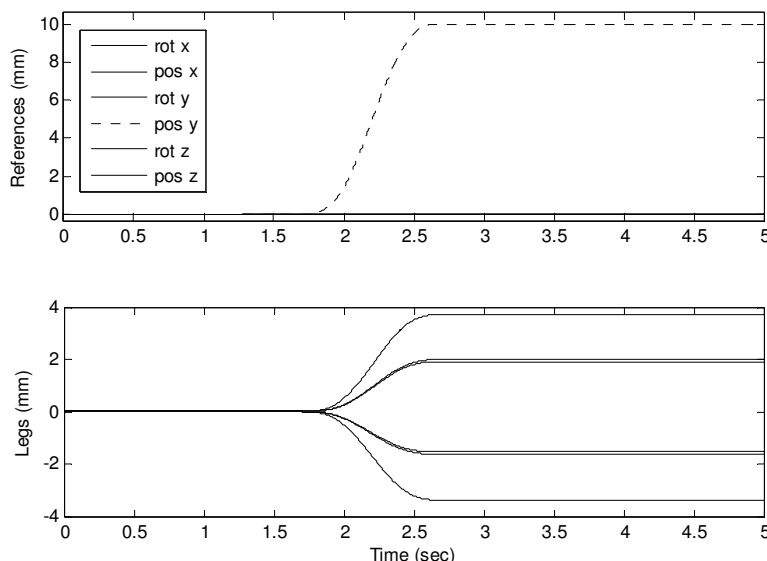


Fig. 31. Linear motion in y direction

4.5 Sliding mode controller

Nonlinear control has a very important role in the robot control applications. An important reason for this situation is to create knowledge-based systems which simplify the modeling of complex dynamics in the robot control. SMC is one of the suitable methods used in the knowledge-based systems (Küçük & Bingül, 2008).

SMC is a special case of the variable structure control systems. Variable structure control systems have a structure using feedback control laws and decision making laws together. Decision-making rule called as switching function selects a special feedback control structure according to the behavior of the system. Variable structure control system is designed to force the system states to slide a special surface called the sliding surface in the state space. Once the sliding surface is reached, the SMC tries to keep the states very close to the sliding surface (Küçük & Bingül, 2008).

If it is considered the change in the position error, second order equation of motion given with Equation 2, 3, 4 and 5 can be written as follows according to $x_1 = \dot{\theta}_r - \dot{\theta}_m$ (Utkin, 1993).

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a_1 x_1 - a_2 x_2 + f(t) - bu\end{aligned}\quad (8)$$

a_1 , a_2 and b are the positive parameters and $f(t)$ is a function depending on load torque, reference input and their derivatives in Equation 8.

For discontinuous control:

$$u = u_0 \text{sign}(s), \quad s = cx_1 + x_2 \quad (9)$$

and as Equation 10 is linear and doesn't depend on $f(t)$, sliding mode on the $s = 0$ line allows to decreasing error exponentially.

$$cx_1 + \dot{x}_1 = 0 \quad (10)$$

Derivative of the sliding surface is:

$$\dot{s} = cx_2 - a_1x_1 - a_2x_2 + f(t) - bu_0sign(s)$$
(11)

From equation,

$$bu_0 > |cx_2 - a_1x_1 - a_2x_2 + f(t)|$$
(12)

s and \dot{s} have opposite signs and the state $s = 0$ will approach the sliding line after a while. Inequality given with Equation 12 determines the required voltage to force the system to sliding mode (Utkin, 1993). A candidate Lyapunov function can be selected as follows for stability analysis (Kassem & Yousef, 2009),

$$v = \frac{1}{2}\sigma^2$$
(13)

The stability condition from Lyapunov's second theorem,

$$\frac{1}{2} \frac{d\sigma^2}{dt} = \sigma\dot{\sigma} \leq -K|\sigma|$$
(14)

where K is a positive constant. After theoretical steps, sliding mode controller was designed in Simulink similar to PID controller. Main model is shown in Figure 32. Details of some subsystems different from using subsystems in PID will be given only.

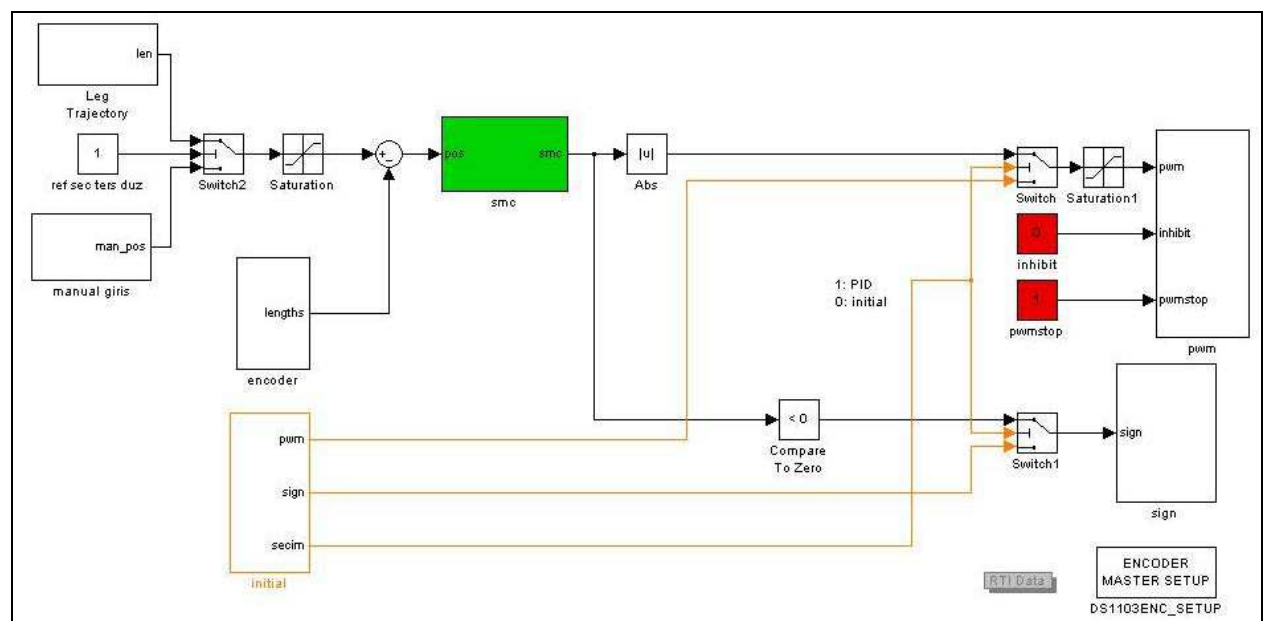


Fig. 32. Main sliding mode controller model

"smc" subsystem is shown in the Figure 33. This model contains sliding mode control and integrator algorithm for one leg. In order to reduce the chattering, a rate limiter block was added to output. In order to eliminate the steady state error, an integrator was added to the

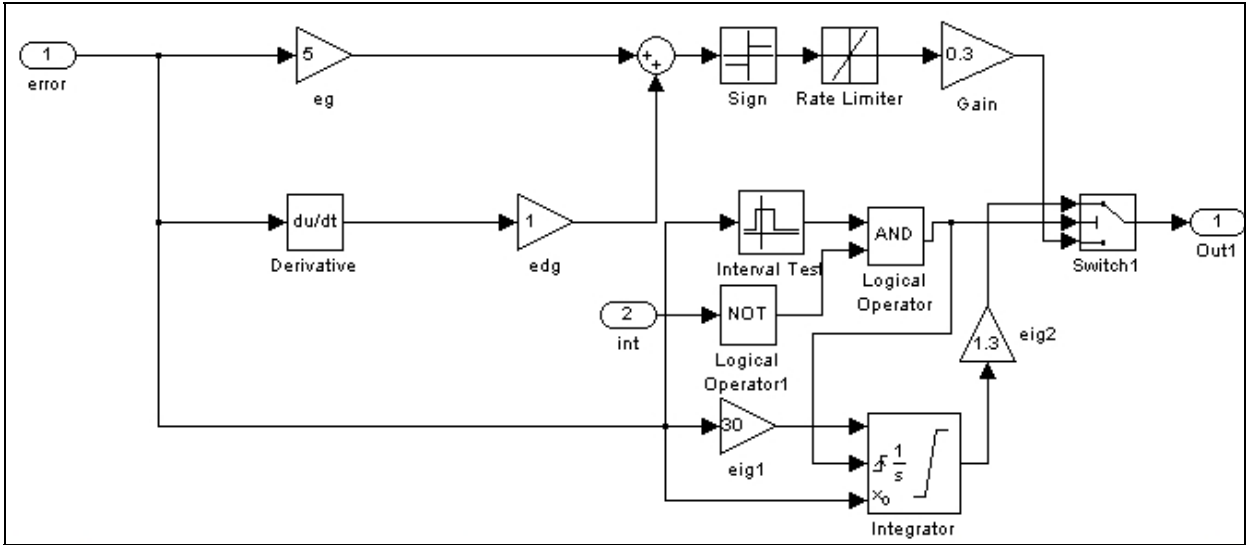


Fig. 33. Simulink model of SMC with integrator for one leg

controller. This integrator is switched on only small limited range between $\pm 0,008$ mm. This integrator does not affect the performance of sliding mode controller. It does not slow down the system response. Also, integrator is disabled during trajectory tracking by switching.

4.5.1 Position control

Some real time responses for position control are given below. In Figure 34, errors of the legs were shown in the motions linear and rotation in the x and z direction with 15 mm and 20° inputs. As can be seen from the figures, overshoot and steady state error are very small. But, system response is slower.

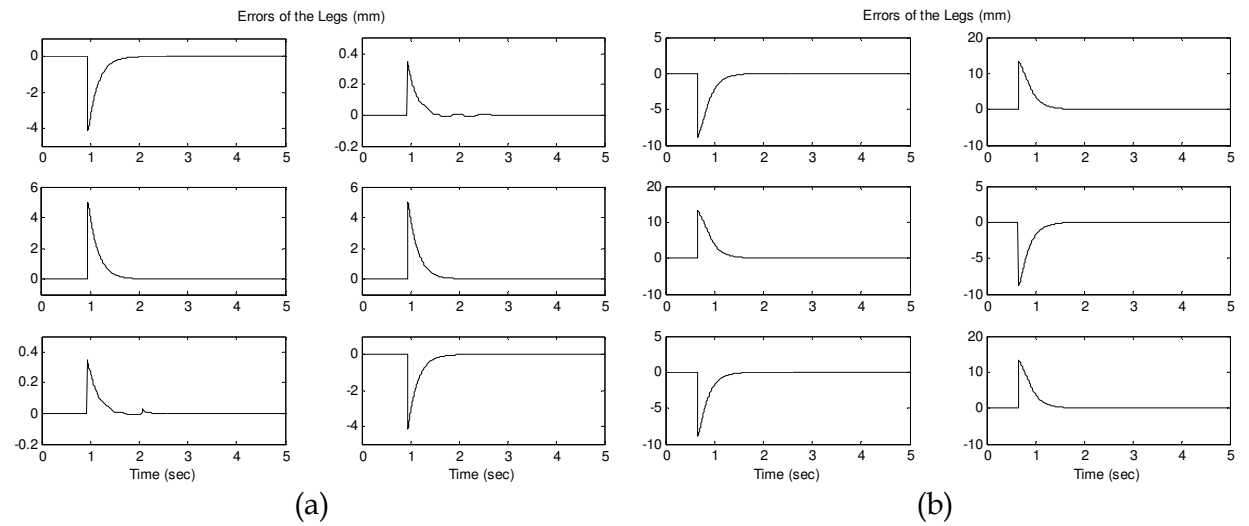


Fig. 34. (a) Linear motion in x direction with 15 mm reference (b) Rotation in z direction with 20° reference

Figure 35 shows a phase diagram of the system with SMC. In the phase diagram, the states of the system are leg position and leg velocity. As can be seen from the figure, SMC pushes states to sliding line and the states went to the desired values along sliding line when 5 mm step input along the z axis in Cartesian space was applied to the system.

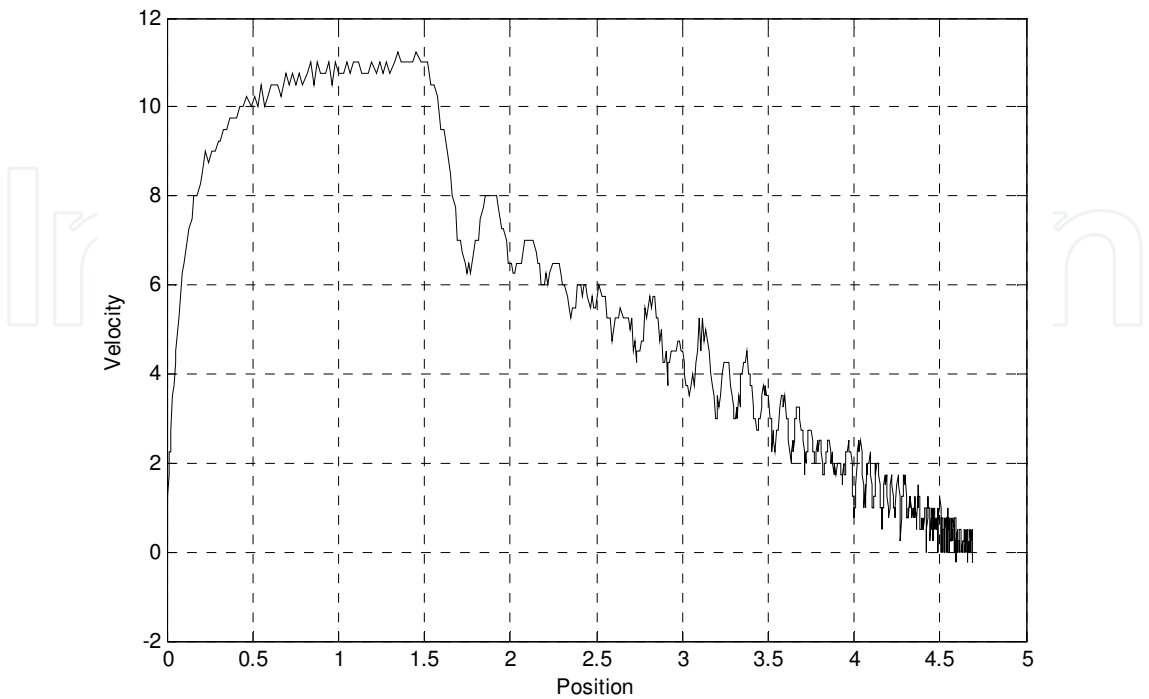


Fig. 35. Phase diagram of the SMC position control

4.5.2 Trajectory control

Different situations in trajectory control are considered in this section. These are shown in Figure 36-38. As can be seen from the figures legs followed the desired trajectories synchronous.

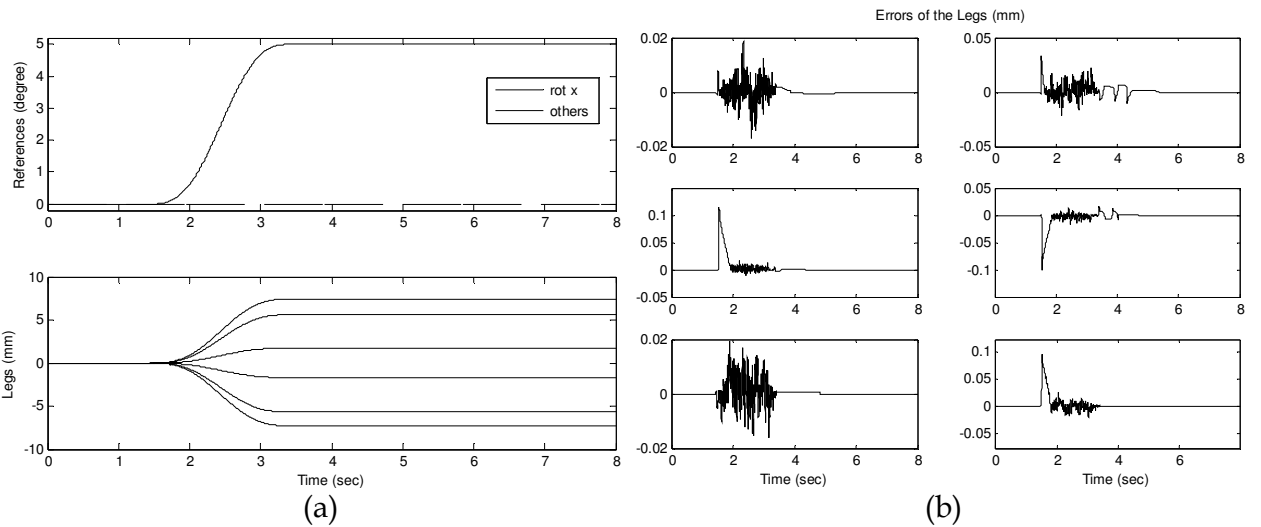


Fig. 36. (a) Rotation in x direction (b) Errors of the legs

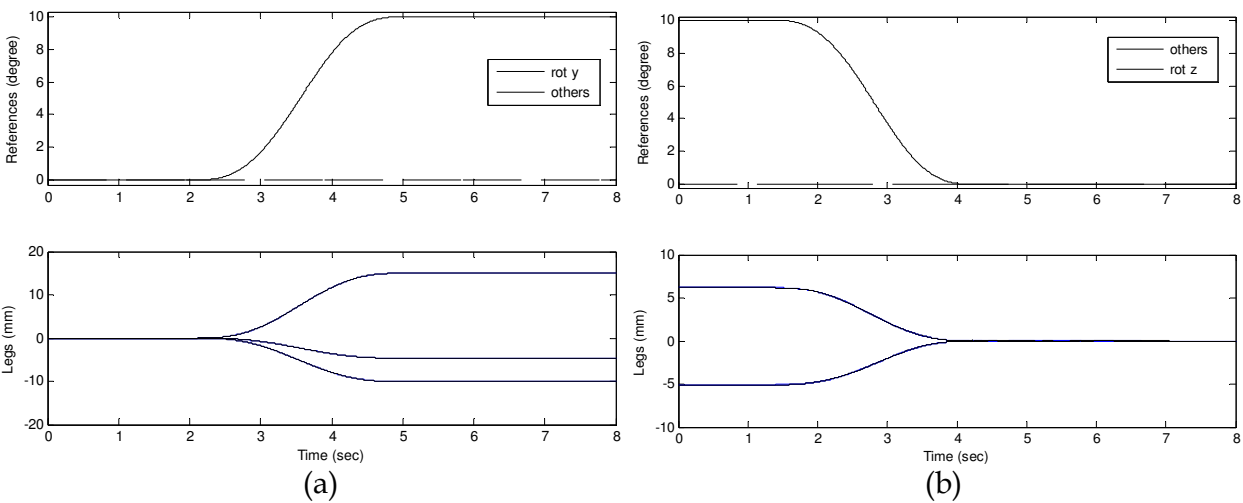


Fig. 37. (a) Rotation in y and (b) z direction

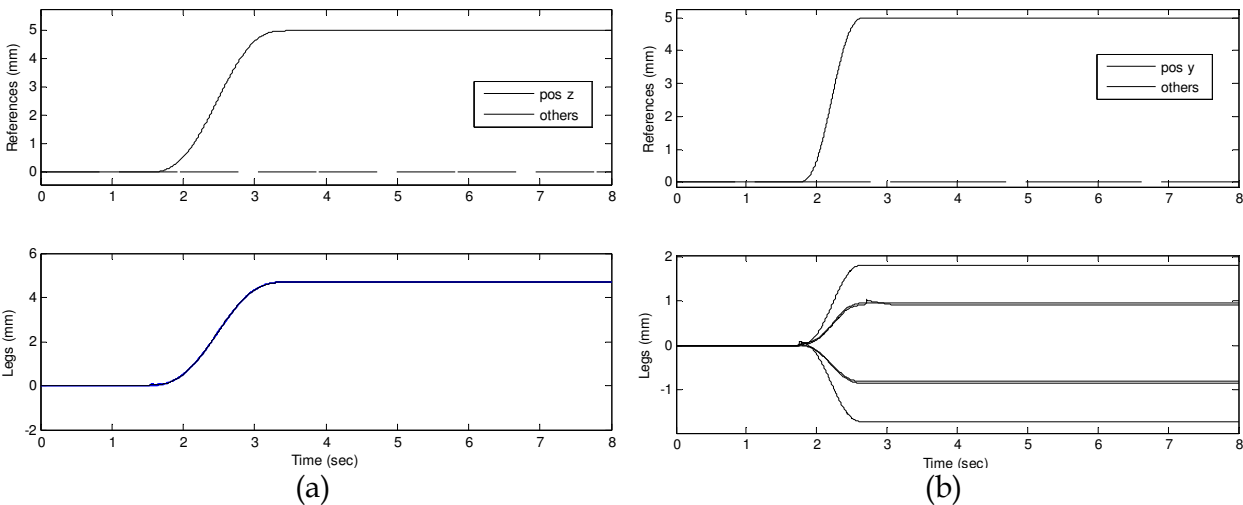


Fig. 38. (a) Linear motion in z and (b) in y direction

5. Conclusion

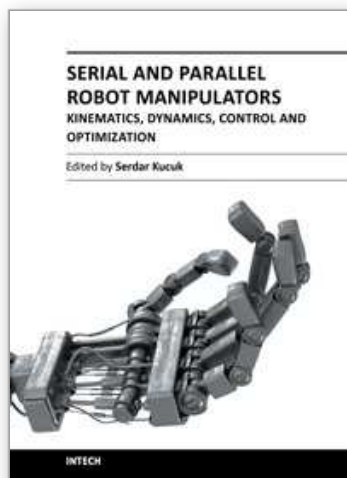
In this study, a high precision 6 DOF Stewart platform is controlled by a PID and sliding mode controller. These controllers were embedded in a Dspace DS1103 real time controller which is programmable in the Simulink environment. Design details and development stages of the PID and SMC are given from subsystems to main model in Simulink. This study can be a good example to show how a real time controller can be developed using Matlab/Simulink and Dspace DS1103. In order to test the performance of the controllers, several position and trajectory tracking experiments were conducted. Step inputs are used for position control and Kane transition function is used to generate trajectory. In the position experiments using both controllers, there is no steady state error and moving plate of the SP is positioned to the desired target with an error less than $0.5\ \mu m$. Sliding mode controller is better performance in terms of overshoot than PID but PID has faster response due to high gain. In the tracking experiments, PID and SMC have similar responses under no load. If nonlinear external forces are applied to moving platform, control performance of the SMC will be better than PID.

6. Acknowledgment

This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under the Grant No. 107M148.

7. References

- Bonev, I. (2003). The True Origins of Parallel Robots, 06.04.2011, Available from: <http://www.parallemic.org/Reviews/Review007.html>
- Chen, N.X.; Song, S.M. (1994). Direct position analysis of the 4-6 Stewart Platform, *ASME J. of Mechanical Design*, Vol. 116, No. 1, (March 1994), pp. (61-66), ISSN 1050-0472
- Dspace Inc., 06.04.2011, Available from: <http://www.dspaceinc.com/en/inc/home.cfm>
- Hunt, K.H. (1983). Structural kinematics of in-parallel-actuated robot-arms, *ASME J. Mech., Trans. Automat. Des.*, Vol. 105, No. 4, (December 1983), pp. (705-712), ISSN 0738-0666
- Fitcher, E.F. (1986). A Stewart Platform-Based Manipulator: General Theory and Practical Construction, *Int. J. of Robotics Research*, Vol. 5, No. 2, (June 1986), pp. (157-182), ISSN 0278-3649
- Kassem, A.M.; Yousef, A. M. (2009). Servo DC Motor Position Control Based on Sliding Mode Approach, *5th Saudi Technical Conference and Exhibition STCEX 2009*, January 2009
- Kim, D.; Chung, W. (1999). Analytic Singularity Equation and Analysis of Six-DOF Parallel Manipulators Using Local Structurization Method, *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 4, (August 1999), pp. (612-622), ISSN 1042-296X
- Kizir, S. ; Bingül, Z. ; Oysu, C. ; Küçük, S. (2011). Development and Control of a High Precision Stewart Platform, *International Journal of Technological Sciences*, Vol. 3, No. 1, pp. (51-59)
- Küçük, S.; Bingül, Z. (2008). *Robot Dinamiği ve Kontrolü*, Birsen press, ISBN 978-975-511-516-0, İstanbul
- Liao, Q.; Seneviratne, L.D.; Earles, S.W.E. (1993). Forward kinematic analysis for the general 4-6 Stewart Platform, *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'93*, ISBN 0-7803-0823-9, Yokohama, Japan, July 1993
- Mathworks Inc., 06.04.2011, Available from: <http://www.mathworks.com/>
- Merlet, J.P. (1992). Direct kinematics and assembly modes of parallel manipulators, *Int. J. of Robotics Research*, Vol. 11, No. 2, (April 1992), pp. (150-162), ISSN 0278-3649
- Merlet, J.P. (Ed(s).). (2006). *Parallel Robots*, Springer, ISBN-10 1-4020-4133-0, Netherlands
- Nauna, P.; Waldron, K.J.; Murthy, V. (1990). Direct kinematic solution of a Stewart Platform, *IEEE Trans. Robotics Automat.*, Vol. 6, No. 4, (August 1990), pp. (438-444), ISSN 1042-296X
- Niesing, B. (2001). Medical Engineering, *Fraunhofer Magazine*, Vol. 2
- Reckdahl, K.J. (1996). Dynamics and control of mechanical systems containing closed kinematic chains, *Phd Thesis*, Stanford University
- Sefrioui, J.; Gosselin, C.M. (1993). Singularity analysis and representation of planar parallel manipulators, *Robot. Autom. Syst.*, Vol. 10, No. 4, pp. (209-224)
- Stewart, D. (1965). A Platform with Six Degrees of Freedom, *Proceedings of the Institute of Mechanical Engineering*, Vol. 180, Part 1, No. 5, pp. (371-386)
- Utkin, V.I., (1993). Sliding mode control design principles and applications to electric drives, *IEEE Transactions on Industrial Electronics*, Vol. 40, No. 1, pp. (23-36), ISSN 1042-296X
- Wikipedia, 06.04.2011, Available from: http://en.wikipedia.org/wiki/Stewart_platform



Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization

Edited by Dr. Serdar Kucuk

ISBN 978-953-51-0437-7

Hard cover, 458 pages

Publisher InTech

Published online 30, March, 2012

Published in print edition March, 2012

The robotics is an important part of modern engineering and is related to a group of branches such as electric & electronics, computer, mathematics and mechanism design. The interest in robotics has been steadily increasing during the last decades. This concern has directly impacted the development of the novel theoretical research areas and products. This new book provides information about fundamental topics of serial and parallel manipulators such as kinematics & dynamics modeling, optimization, control algorithms and design strategies. I would like to thank all authors who have contributed the book chapters with their valuable novel ideas and current developments.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Selçuk Kizir and Zafer Bingul (2012). Position Control and Trajectory Tracking of the Stewart Platform, Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization, Dr. Serdar Kucuk (Ed.), ISBN: 978-953-51-0437-7, InTech, Available from: <http://www.intechopen.com/books/serial-and-parallel-robot-manipulators-kinematics-dynamics-control-and-optimization/position-control-and-trajectory-tracking-of-the-stewart-platform->

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen