Dynamic Analysis of Vehicle Simulator using CAD Applications

Mukesh Sadana[1*], Subir Kumar Saha[2] and P. V. M. Rao[3]
[1]Sponsored M. Tech. Student from Solid State Physics Lab. (DRDO), Lucknow Road, Delhi-54, India
[1, 2, 3] Department of Mechanical Engineering, Indian Institute of Technology Delhi, New Delhi, India

[*] Mukesh Sadana (email: mukesh_sadana@yahoo.co.in)

Abstract

One of the basic and foremost problems in the field of vehicle simulator is the real time calculations of actuation forces for positioning and orienting the driver's cabin which is mounted on a six degrees of freedom motion platform (Stewart Platform). The task requires a dedicated inverse dynamics algorithm for computation of these actuation forces. This has given a stimulus to the development of inverse dynamics algorithms which are extremely efficient and accurate, since the realism incorporated relies directly on them. In developing such algorithms, one certainly needs validation at various stages of developments. In an effort to develop and validate one such algorithm based on the Decoupled Natural Orthogonal Complement (DeNOC) technique [4], an attempt has been made to completely model such platforms using CAD applications, which is reported in this paper. The validation procedure followed is explained here. The issues related to modeling and simulations have also been discussed in sufficient details. Stewart platform models with different architectural constructions have been generated considering actual working dimensions. These are then subjected to kinematic and dynamic analyses for various motion trajectories. Results were generated and used for validation of dynamic algorithms. The effect of using different constructions, e.g., UPS and SPS, on dynamics of system has also been studied.

Keywords: CAD, DeNOC, UPS, SPS

1 Introduction

Most of the vehicle simulators use a 6-degree of freedom (DOF) motion platform (better known as Stewart platform) to provide the motion cues to the driver's cabin for various terrain conditions. These vehicle simulators find large application in defence and aerospace organizations to impart trainings. In fact, such simulators can be used for entertainment industry as well. The Stewart motion platform, as shown in Fig. (1), is parallel closed chain manipulator which has six linear actuators (known as limbs) connecting a top mobile platform to bottom fixed platform. The fixed platform is connected to the limbs by six universal joints, and the mobile platform on which driver's cabin (end-effector) is mounted by six spherical joints. The motion of the end-effector is controlled by changing the limb lengths in various proportions. This job is done by linear actuators, which connects the lower part of the limb (lower leg) to the upper part (upper leg).

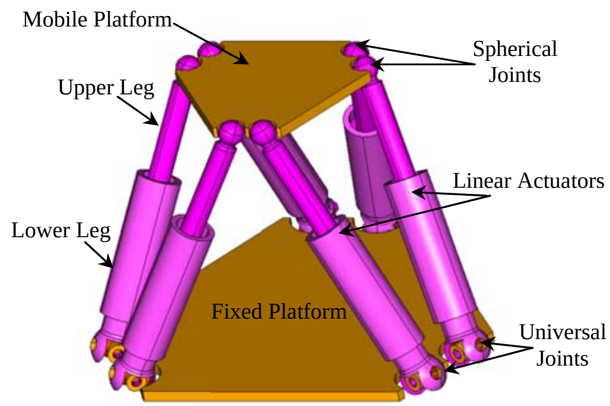

Fig. 1: I-deas Screenshot of a 6-DOF Stewart Platform (Model-I)

To change the leg lengths accurately, the actuators require exact amount of time varying force. This is accomplished by inverse dynamics algorithm. The inputs to these algorithms are the motion trajectories to be acquired by the end-effector in terms of its DOF and, mass and inertia properties of the system. The motion trajectories are in terms of position and orientation of mobile platform with respect to fixed platform. The outputs are the actuation forces required by all six actuators to achieve the required motion. Conversely, to predict and simulate the motion of the end-effector with given input actuation forces, forward dynamics algorithm is required.

In this paper an attempt is made to model and analyse such platforms using CAD applications in order to study its dynamics behavior. Both inverse and forward dynamics simulations are performed. This is particularly useful when need arises of validation of the correctness of the developed dynamics algorithms. This is to be stated that the modeling and dynamic analyses are done in an attempt to validate one such algorithm based on the Decoupled Natural Orthogonal Complement (DeNOC) matrices and subsystem approach [1].

I-deas CAD package is used for solid part modeling, assembly, and mechanism design. Further, ADAMS is used for dynamic analysis, simulation and post processing of results. The dynamic algorithm whose validation is required was programmed in MATLAB. The paper is organized as follows: Section 2 explains the validation procedure, followed by section 3 on part modeling, assembly, and mechanism design. Sections 4 and 5 present kinematic and dynamic analyses, respectively. Finally, the conclusions are given in section 6.

2 Validation of Dynamics Algorithm

One way to validate a developed dynamic algorithm is to refer to standard data and results published in written media like journals and books. In this way, one can validate only at final stage of results. However, a step by step validation at various stages may not be possible to build a convincing confidence for proceeding further. On the other hand the use of CAD applications not only enables one to check the final results but one can verify the correctness of the algorithm at any intermediate stage. For example, orientation and positions of the limbs can be verified at any stage. The procedure followed for validation of the aforesaid algorithms is shown in Fig. (2). It can be seen that the left block represents the CAD part which interacts with the right block, i.e., the developed dynamic algorithm. Some of the inputs are essentially being provided by the CAD model to the computer program. These include the mass and inertia properties and architectural inputs, which may be often time consuming and difficult to calculate when complex geometries are present. Examples of architectural inputs are the vectors defining the stationary axis of universal joints in global frame and the platform connection points in their respective local frames of reference [3]. Note that the input motion trajectories of end-effector are common to CAD and developed programs. Once the model is ready one can play with various parameters easily at own wish. All these advantages make the use of CAD modeling rather essential for completeness of validation.

3 Modeling and Simulation

Basically there are two different kinds of architectural constructions possible based on the joints used, i.e., SPS (spherical-prismatic-spherical) and UPS (universal-prismatic-spherical), UPS being the most used in practice. Modeling has been done for both of these. Solid part modeling, assembly and mechanism formulation are done in I-deas. After that mechanism is imported to ADAMS; kinematic analysis, dynamic analysis, and simulation are performed.

In total three different models have been generated, which are as follows:

1. Model-I of assumed arbitrary dimensions with UPS construction, as shown in Fig. (1).

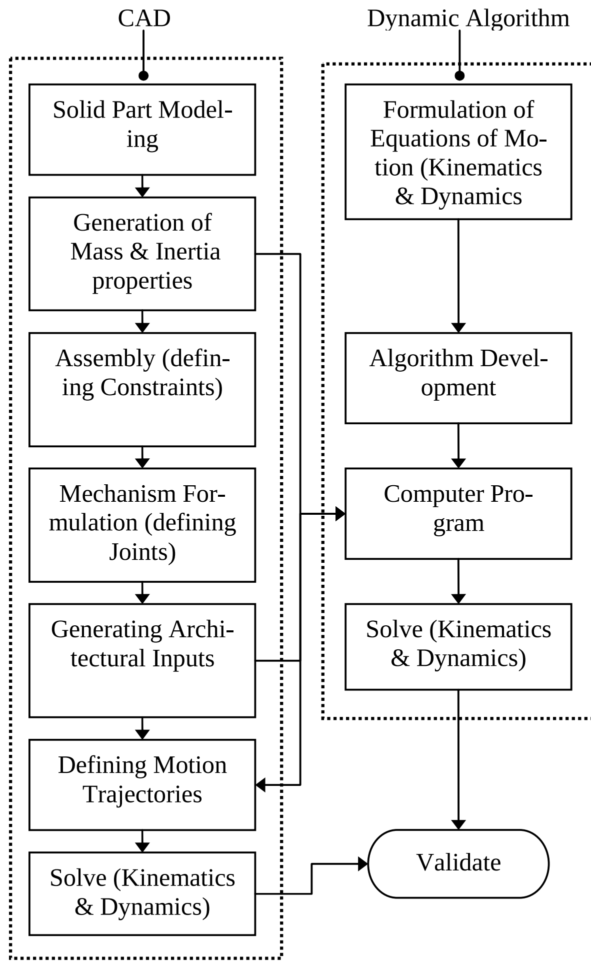

Fig. 2: Validation procedure

2. Model-II based on actual dimensions and architecture, as shown in Fig. (3).
3. Model-III in both UPS and SPS constructions as shown in Fig. (4).

In Model-II, all the higher DOF joints are made as serially connected single-DOF joints. Model-III is of same dimensions as of Model-I but with different construction for the lower leg. The purpose of generating this Model is to compare the dynamic characteristics of UPS and SPS constructions.

3.1 Architecture and Frame Attachment

Architecture for top and bottom platform is selected as semi regular hexagons, which is most commonly used due to its favorable architectural singularity characteristics [4]. The geometric parameters describing the architecture of platforms are c_t, c_b, d_t and d_b, which are shown in Fig. (5). Coordinate frames can be attached anywhere on the platform, but in general the bottom platform frame should be placed at one of the faces of platform in such a way that the origin lies at the intersection of perpendicular bisectors of any two large sides. In case of top platform, the frame should be attached such that its origin lies at the center of mass of the platform. The Y-

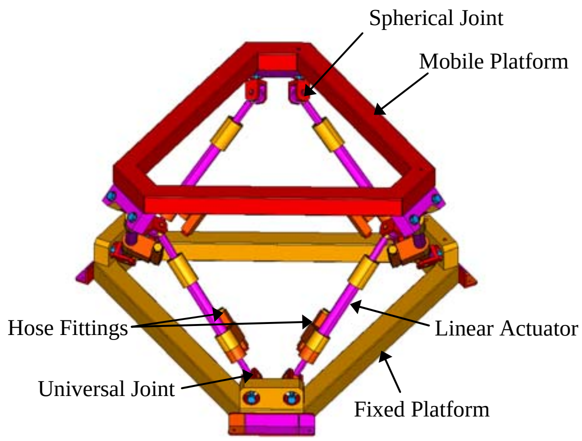

Fig. 3: I-deas Screenshot of Model-II

axis points towards the perpendicular of any large side for both the platforms. The Z-axis is taken perpendicular to face and along vertically upward direction, and X-axis completes the coordinate system according to the right hand rule. Frame placements for the top platform (X_0-Y_0-Z_0) and bottom platform (X-Y-Z) are shown in Fig. (5). Frames are assigned in order to define the trajectories, and later for the measurement of various vectors and motion parameters related to the mobile platform. Proper assignment of these is necessary since various inputs for dynamics algorithms are defined with respect to these frames.

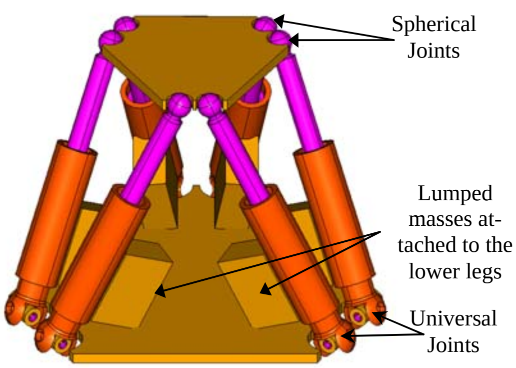

Fig. 4: I-deas Screenshot of Model-III (UPS)

For specifying and obtaining various vectors related to the legs in the fixed platform frame X-Y-Z, a frame of reference D (x-y-z) is attached to lower leg as shown in the Fig. (6). Origin of this frame is fixed at base point of lower leg (B_i). Here, x-axis is along the leg, y-axis is along the rotating axis of universal joint fixed to the leg, and z-axis completes the coordinate system according to right hand rule. Another frame of reference U with same orientation is attached to the upper leg with the origin at platform connection points A_i.

3.2 Difficulties in Modeling

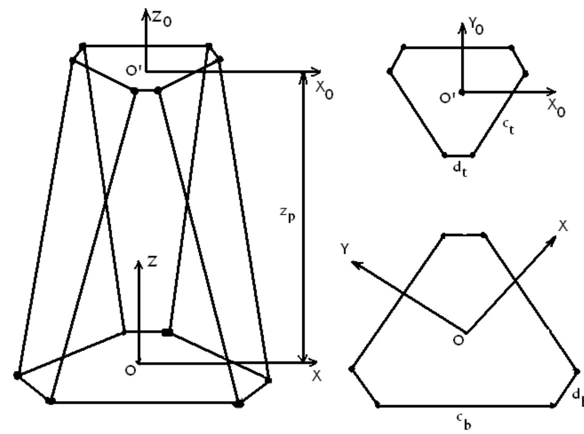

Fig. 5: Assignment of Platform Coordinate Frames

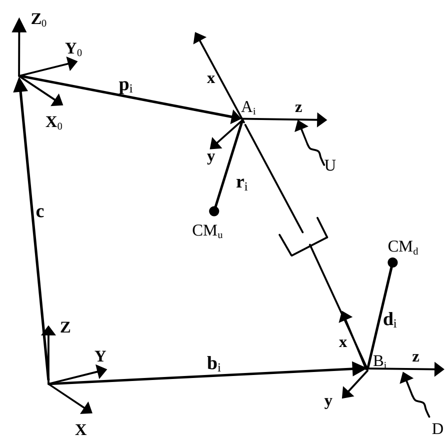

Fig. 6: Assignment of Leg Coordinate Frame

CAD modeling and subsequent dynamic analysis of complex systems involving parallel kinematic chain with large degrees of freedom like Stewart platform involves many difficulties. Some of them, which one most likely comes across, are discussed below:

1. It may not be easy to model solid parts with complex geometry in applications specially intended for dynamic analysis like ADAMS. This is especially true when number of parts is many. The use of CAD modeling software like I-deas provides a greater flexibility for this purpose, and results in saving in time and effort.
2. Likewise, assembly and mechanism formulation tasks are also easy to be carried out in I-deas itself. This is because one need not calculate the configuration (orientation and position) of the parts and joints, which decides the desired configuration of the assembly. This otherwise is an essential task to be performed before doing assembly in ADAMS.
3. Material should be decided during the modeling stage in I-deas. This is to save repetition of efforts for applying it to various identical parts in assembly, e.g., six identical lower legs. This is not possible in ADAMS, where material has to be applied to each individual part, no matter even if it is a replica of another part. Likewise, mass and inertia properties should be defined at solid modeling level. These

properties generated by the solid model can also be changed at the dynamic simulation stage if required.

4. I-deas is having the feature of applying the joints automatically after making the assembly. This can be done by 'map assembly constraints' option enabled. For example, if some point in a part is coincident with a point of another part, it automatically creates a 3-DOF spherical joint. Likewise, centerline to centerline collinear-coincident constraint makes a 2-DOF cylindrical joint, which can easily be converted into a single-DOF translational joint if desired. This feature may be utilized to save time and effort.
5. ADAMS provides greater flexibility in application of motion parameters (trajectories), dynamic solution and post processing of results.

4 Kinematic Analysis

Basically solution is obtained for inverse kinematics, in which end-effector (moving platform) positions and orientations are known as a function of time, and the interest lies in finding out the motion parameters of various other elements of the system as a history of time. These motion parameters are the position, velocity and acceleration, both linear and angular components. Examples are leg displacements, leg sliding velocities and accelerations, leg angular velocities and accelerations, etc.

In general, for Stewart platform, solution for kinematics is obtained by considering SPS construction. This is done in order to make the derivations simpler, by assuming that the rotations of the legs about their own axis are not allowed. The algorithm to be validated uses loop-closure equations and is based on the said assumption. Models I and II with UPS constructions are solved in ADAMS for kinematics with trajectories given in next section. Subsequently the results obtained from kinematic algorithm are compared.

4.1 Motion Trajectories

The models are solved using three different trajectories as inputs. Data for these are chosen arbitrarily. Trajectories are as follows:

Trajectory-I: Pure translation along Z-axis upwards with constant linear acceleration.
Acceleration = $[0,\ 0,\ 20]^T$ mm/sec^2 ;Time for simulation =1 sec.

Trajectory-II: Pure rotation about Z-axis with constant angular acceleration.
Angular acceleration = $[0,\ 0,\ 0.1349]^T$ rad/sec^2 ; Time for simulation =0.75 sec.
Initial position and orientation of top platform frame with respect to the base frame for the above trajectories are $[0,\ 0,\ 235]^T$ mm; $[0,\ 0,\ 60*pi/180]^T$.

Trajectory-III: Only Model-II is subjected to this trajectory in addition to the above two trajectories. The trajectory is given by [2]

$$q_i = q_i(0) + \frac{q_i(T) - q_i(0)}{T}\left[t - \frac{T}{2\pi}\sin\left(\frac{2\pi}{T}t\right)\right]$$

where T=10sec. For pure translation initial and final values considered are

$q_i(0) = 945\text{mm};\ \dot{q}_i(0) = \ddot{q}_i(0) = 0,$

$q_i(T) = 1045\text{mm};\ \dot{q}_i(T) = \ddot{q}_i(T) = 0.$

For pure rotation initial and final values are

$q_i(0) = 1.047\text{ rad};\ \dot{q}_i(0) = \ddot{q}_i(0) = 0,$

$q_i(T) = 1.57\text{ rad};\ \dot{q}_i(T) = \ddot{q}_i(T) = 0,$

The input trajectories are shown in Figs. (7-10). These trajectories are also used to obtain the solution for the dynamics and its validation.

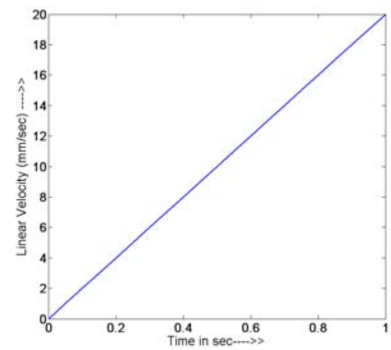

Fig. 7: Platform Linear Velocity along Z (Trajectory-I)

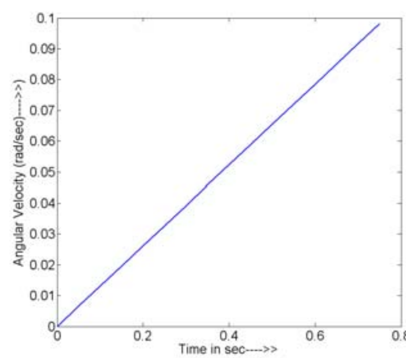

Fig. 8: Platform Angular Velocity about Z (Trajectory-II)

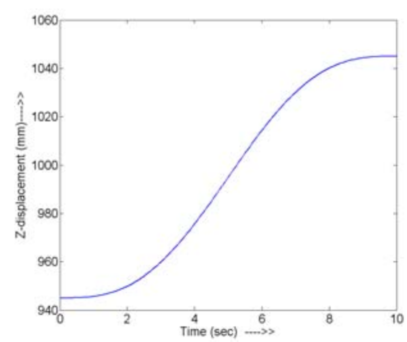

Fig. 9: Trajectory-III (Platform in Pure Translation)

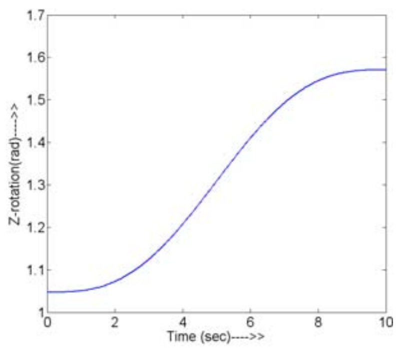

Fig. 10: Trajectory-III (Platform in Pure Rotation)

4.2 Results and Validation

The results for Model-I were obtained using ADAMS. Leg sliding velocities are shown in Figs. (11) and (12). These are shown for trajectories I and II for arbitrary legs. Comparisons with the MATLAB program based on an in-housed developed algorithm are done by importing and superimposing the ADAMS and MATLAB results.

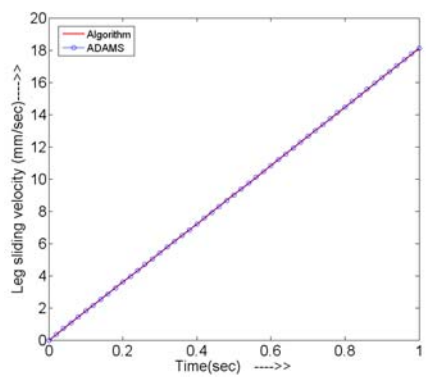

Fig. 11: Leg Sliding Velocity (Model-I, Trajectory-I)

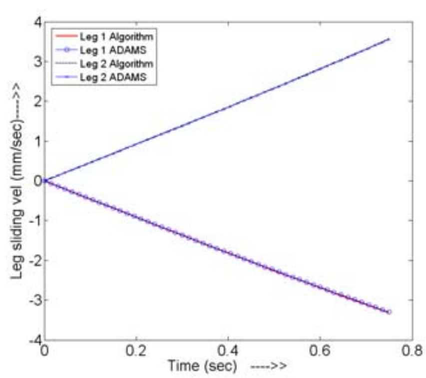

Fig. 12: Leg Sliding Velocity (Model-I, Trajectory-II)

5 Dynamic Analysis

Stewart platform models are analyzed for both inverse and forward dynamics in ADAMS.

5.1 Inverse Dynamic Analysis

Inverse dynamic analysis is done for the calculation of actuation forces to acquire the input end-effector motions. Performing inverse dynamics is not so straight forward in ADAMS. In ADAMS, It is not possible to define and apply the motions to the end-effector and measure forces at the prismatic joints directly. Forces can only be measured at the point where the actuators are placed. In other words, it is needed to apply such motion trajectories to the prismatic joints, which causes the end-effector to move in the specified way. To accomplish this, the model was used to solve for inverse kinematics and the numerical data related to the motion parameters for each individual leg are obtained with time as independent axis. Then one of the motion parameters, e.g., leg sliding accelerations, was fitted as polynomial in time or as splines for all the six legs.

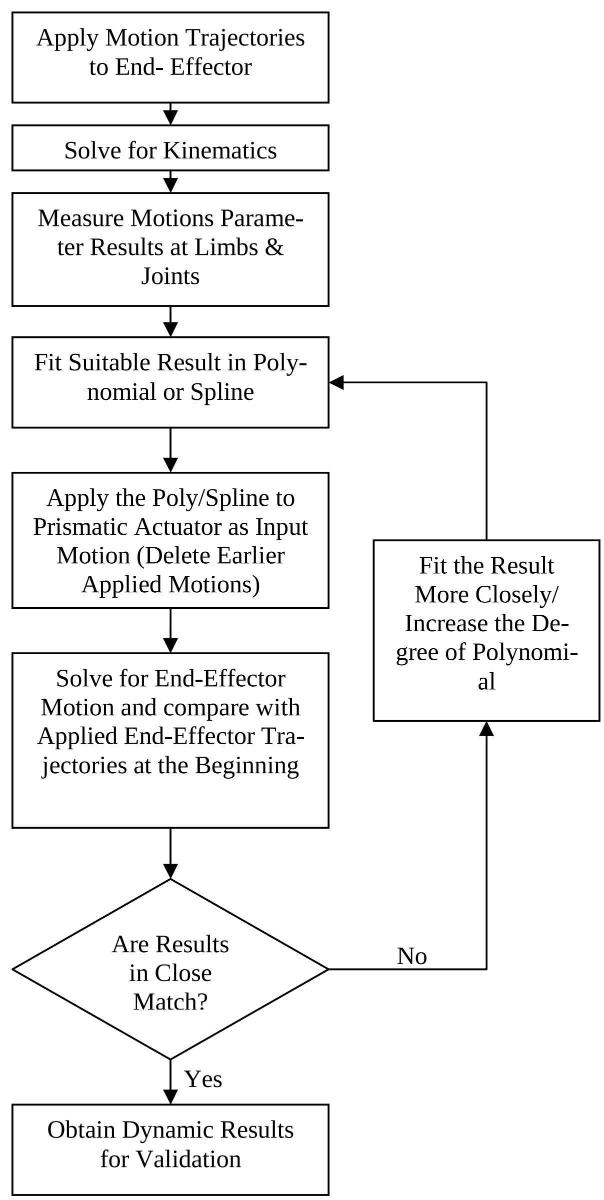

Fig. 13: Steps for Inverse Dynamic Analysis in ADAMS

Later, these motion splines or polynomials are applied to the respective legs, deleting the end-effector motions applied earlier. The end-effector motions attained are then compared with the desired motions and the actuation forces are measured related to applied leg motions. The procedural steps for inverse dynamics are enumerated in Fig. (13).

5.1.1 Results and Validation

Model-I was analysed for the first two trajectories, i.e., Trajectories I and II, whereas Model-II based on actual dimensions was subjected to Trajectory III also, in addition to other two trajectories. Trajectory-I and Trajectory III in pure translation has resulted in approximately same actuation force for all the legs. Hence these are shown for one leg only, whereas for pure rotational trajectories, results for any two consecutive legs are shown since same results were observed for every alternate leg. Payload was neglected which can be easily incorporated. The description of all the three models is given in the Appendix. The ADAMS results for leg actuation forces are shown in Figs. (14-18). Comparisons with the results obtained from the in-house algorithm are also shown. The results vary within ±0.3 percent. This may be attributed mainly due to the approximated trajectories.

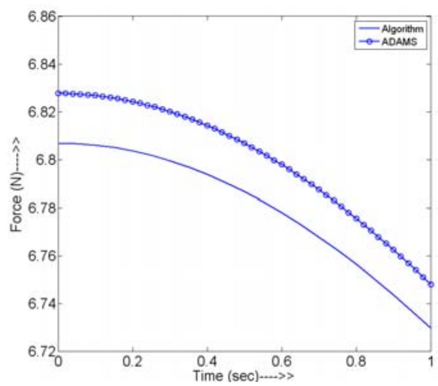

Fig. 14: Actuation Force (Model-I, Trajectory-I)

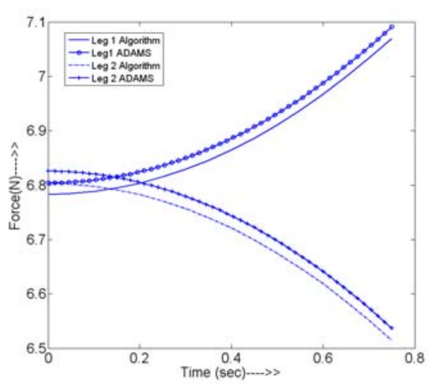

Fig. 15: Actuation Force (Model-I, Trajectory-II)

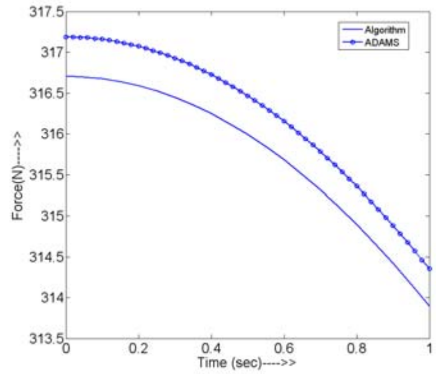

Fig. 16: Actuation Force (Model-II, Trajectory-I)

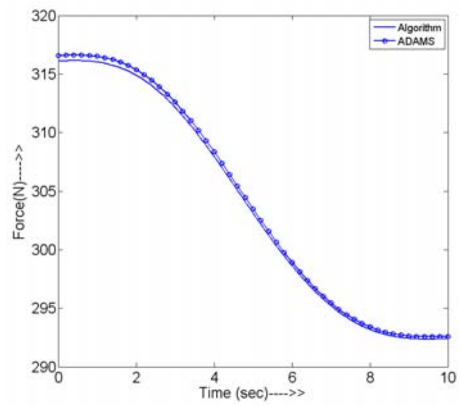

Fig. 17: Actuation Force (Model-II, Trajectory-III under Pure Translation)

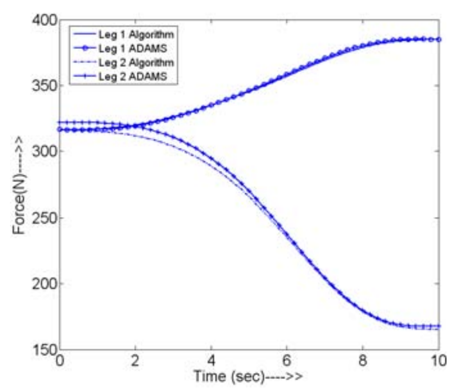

Fig. 18: Actuation Force (Model-II, Trajectory-III under Pure Rotation)

5.1.2 Comparison of SPS and UPS Dynamics

The SPS construction adds one rotational DOF per limb to the manipulator in comparison to the use of UPS construction. Model-III is generated in a way that the limbs are not symmetric and the center of mass (CM) lies away from the center line. For this, lower legs of Model I were modified by attaching lumped masses, as shown in Fig. (4), keeping the other dimensions exactly same. Model III was made considering the UPS construction at first. Then for making the SPS construction, spherical joint was made at the stationary point where the two axes of the universal joints meet, and removing the cross shaft. This is shown in Fig. (19). Then these Models are solved for dynamics with Trajectory-I. The result for actuation force of a leg is shown in Fig. (20). It can be observed that the actuation force fluctuates due to the rotation of the limbs around their own axes, in SPS construction. This is due to the rotation of the CM around the line joining the two spherical joints of the limb. Note that, Model-I was also solved with SPS construction, in addition to the UPS construction, the results of which are not shown here. In this case the actuation force does not differ for these two constructions. This may be attributed to the unchanged path of CM of lower and upper leg, regardless of the leg rotation. In actual practice, there are hydraulic hose fittings attached to the limbs, as shown in Fig. (3), which makes the CM of the legs off the line connecting the two joints of the limb. This is one of the reasons that the SPS is not used in practice.

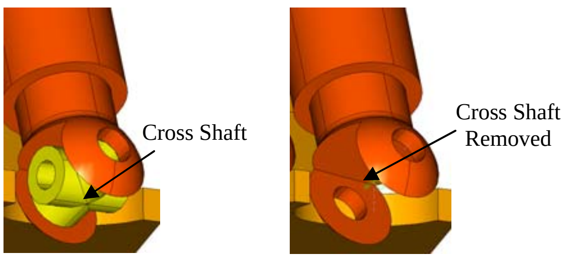

Fig. 19: Conversion of UPS to SPS in I-deas

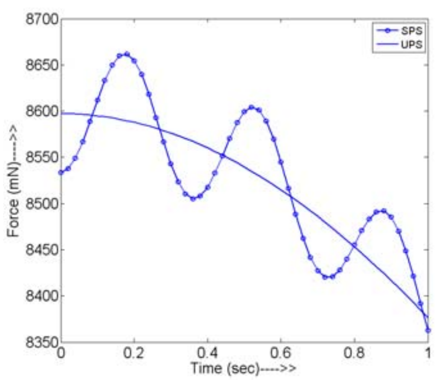

Fig. 20: Actuation Force Comparison of UPS and SPS Constructions (Model-III)

5.2 Forward Dynamics and Simulation

Basically, the task-space forward dynamics is performed here. The initial motion state of the system and the time history of the driving force at the actuated joints are known, and it is desirable to determine the resulting motion history of the task-space, i.e., the end-effector. Forward dynamics algorithm is validated, which is also based on the DeNOC matrices and subsystem approach. The approach was used to derive the closed form of dynamic equations of motion. A MATLAB code is written for this purpose, the inputs to which are six actuation forces as well as the initial positions and velocities of the end-effector with respect to the fixed base. Outputs are the time histories of position and velocities of end-effector.

5.2.1 Results

Model-I was simulated in ADAMS for forward dynamics for both free and forced conditions. For free simulation only gravity was considered and the model was analyzed under free fall. For forced simulation all the six actuators were given constant actuator force of 20mN without considering the gravity or any other external force. The model was solved for initial positions stated in section 4.1; initial velocities (angular as well as linear) are all being zero initially. The results for free and forced simulation are shown in Figs. (21) and (22). Comparisons with MATLAB results are also shown.

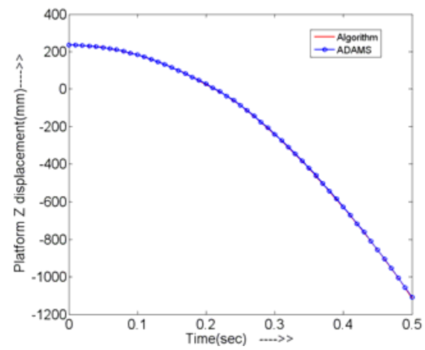

Fig. 21: Free Simulation results for Position (Model-I)

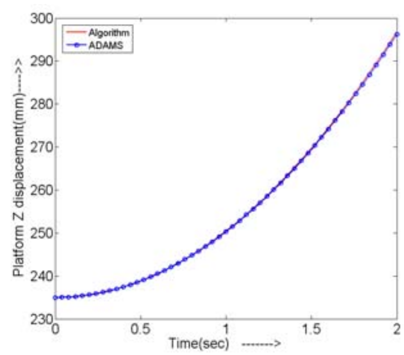

Fig. 22: Forced Simulation results for Position (Model-I)

6 Conclusions

Complete dynamic modeling of a complex Stewart platform for vehicle simulator applications was carried out using different CAD applications. The advantages of the CAD modeling to validate a newly developed algorithm are shown to be many, which make the use of these tools essential. A systematic approach for validation has been explained. Various modeling issues are also being discussed. The kinematic and inverse dynamics analyses were done using different models of arbitrary as well as actual dimensions and architectures. Different trajectories were considered while solving. The results were compared with those obtained from the in-house MATcompared with those obtained from the in-house MATLAB programs based on the DeNOC matrices and subsystem approach. The dynamics of the SPS and UPS constructions were also compared. In the end, forward dynamic analysis was also performed. In this way, a new algorithm has been shown to be validated using CAD applications. The either model can now be tested for various complex trajectories involving all the six degrees of freedom.

References

[1] H. Chaudhary, and S.K. Saha, "Constraint wrench formulation for closed-loop systems using two-level recursions," *Transactions of the ASME Vol. 129, December 2007, pp. 1234- 1242.*

[2] H. Chaudhary, and S.K. Saha, "Dynamics and balancing of multibody systems," *Springer-Verlag Berlin Heidelberg, 2009.*

[3] B. Dasgupta, and T. S. Mruthyunjaya, "A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator," in *Mechanism and Machine Theory, 1998, Vol. 33, No.6, pp. 1135-1152.*

[4] S.K. Saha, "Introduction to Robotics," *Tata McGraw Hill, New Delhi, 2008.*

[5] O. Ma, and J. Angeles, "Optimum architecture design of platform manipulators," *IEEE, 1991 7803-0078/9110600-11308.*

Appendix

The appendix defines the kinematic and dynamic parameters used in the simulation in SI units.

Description of Model-I

Unit vectors along the fixed axes of universal joint are

$$\begin{bmatrix} -0.5001 & -0.5001 & -0.5001 & -0.5001 & 1.0000 & 1.0000 \\ 0.8660 & 0.8660 & -0.8660 & -0.8660 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Mass of upper leg= 0.37 kg; Mass of lower leg= 0.72kg
Mass of moving platform= 1.36 kg
Moment of inertia matrix for upper leg about its CM

$$= \begin{bmatrix} 22.44 & 0 & 0 \\ 0 & 681.37 & 0 \\ 0 & 0 & 681.37 \end{bmatrix} \text{kg.mm}^2$$

Moment of inertia matrix of platform about its CM

$$= \begin{bmatrix} 1976 & 0 & 0 \\ 0 & 1976 & 0 \\ 0 & 0 & 3861 \end{bmatrix} \text{kg.mm}^2$$

Moment of inertia matrix of lower leg at its CM

$$= \begin{bmatrix} 174.20 & 0 & 0 \\ 0 & 1487.3 & 0 \\ 0 & 0 & 1487.3 \end{bmatrix} \text{kg.mm}^2$$

CM of lower and upper legs [see Fig. (6)]

$\mathbf{d}_i = \begin{bmatrix} 77.25 & 0 & 0 \end{bmatrix}^T$; $\mathbf{r}_i = \begin{bmatrix} -67.86 & 0 & 0 \end{bmatrix}^T$

Platform points (A_i) or vectors $\mathbf{p}_i$

$$\begin{bmatrix} 14.00 & 78.79 & 64.79 & -64.79 & -78.79 & -14.00 \\ -82.90 & 29.32 & 53.57 & 53.57 & 29.32 & -82.90 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Base points (B_i) or vectors $\mathbf{b}_i$

$$\begin{bmatrix} 157.20 & 129.20 & -129.20 & -157.20 & -28.00 & 28.00 \\ 58.43 & 106.92 & 106.92 & 58.43 & -165.35 & -165.35 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Description of Model-II

Unit vectors along the fixed axes of universal joint are

$$\begin{bmatrix} 37.5 & 37.5 & -75 & -75 & 37.5 & 37.5 \\ 64.95 & 64.95 & 0 & 0 & -64.9 & -64.9 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Mass of upper leg= 2.78kg; Mass of lower leg= 20.71 kg
Mass of moving platform= 115.78kg
Moment of inertia matrix for upper leg about its CM

$$= \begin{bmatrix} 435.08 & 0 & 0 \\ 0 & 31288.46 & 0 \\ 0 & 0 & 31288.46 \end{bmatrix} \text{kg.mm}^2$$

Moment of inertia matrix of platform about its CM

$$= 10^3 . \begin{bmatrix} 26278.8 & 0 & 0 \\ 0 & 26228.5 & 0 \\ 0 & 0 & 51738.7 \end{bmatrix} \text{kg.mm}^2$$

Moment of inertia matrix of lower leg at its CM =

$$10^3 . \begin{bmatrix} 71.45 & 0 & 0 \\ 0 & 715.19 & 0 \\ 0 & 0 & 662.39 \end{bmatrix} \text{kg.mm}^2$$

CM of lower and upper legs

$\mathbf{r}_i = \begin{bmatrix} -231.83 & 0 & 0 \end{bmatrix}^T$; $\mathbf{d}_i = \begin{bmatrix} 289.93 & 0 & -36.21 \end{bmatrix}^T$

Platform points (A_i) or vectors $\mathbf{p}_i$

$$\begin{bmatrix} 257.25 & -687.74 & -687.74 & 257.25 & 430.57 & 430.57 \\ 645.60 & 100 & -100 & -645.60 & -545.6 & 545.60 \\ -136.39 & -136.39 & -136.39 & -136.39 & -136.39 & -136.39 \end{bmatrix}$$

Base points (B_i) or vectors $\mathbf{b}_i$

$$\begin{bmatrix} 448.38 & 286.68 & -732.9 & -732.9 & 284.5 & 446.22 \\ 587.41 & 680.77 & 94.6 & -92.11 & -682.01 & -588.65 \\ 160.5 & 160.5 & 160.5 & 160.5 & 160.5 & 160.5 \end{bmatrix}$$

Description of Model-III

Only lower leg parameters are given, since all the other parameters are same as model-I.
Mass of lower leg= 0.72kg
Moment of inertia matrix of lower leg at its CM

$$= \begin{bmatrix} 1284.26 & 0 & 0 \\ 0 & 2210.43 & 0 \\ 0 & 0 & 3147.05 \end{bmatrix} \text{kg.mm}^2$$

CM of lower leg (in local frame) $\mathbf{d}_i = \begin{bmatrix} 73.11 & -24.79 & 0 \end{bmatrix}^T$