Dynamic Analysis and Control of a Stewart Platform Using A Novel Automatic Differentiation Method

Xiaoli Bai, *James D. Turner, †and John L. Junkins ‡
Texas A & M University, College Station, Texas, 77843-3141, USA

This paper presents a kinematic based Lagrangian approach to generate the equations of motion and design an adaptive control law for multi-body systems. This methodology is applied to dynamic analysis and controller design study for a Stewart platform. Novel means of utilizing automatic differentiation are employed to generate and solve the equations of motion, using only high level geometric and kinematic descriptions of the system. Based on deriving and coding only the kinematic descriptions of the system, the nonlinear motion of the platform is solved automatically and the analyst is freed from deriving, coding, and validating the lengthy nonlinear equations of motion. Lyapunov stability theory and concepts from adaptive control are used to formulate a nonlinear feedback control law. The control law is of the model reference adaptive structure, designed to track a prescribed smooth trajectory. By designing an adaptive update rule for the system mass and inertia parameters, the tracking errors are proven to be asymptotically stable for arbitrary parameter errors. Also, a PID adaptive control law is designed to guarantee bounded stability in the presence of bounded disturbances. Numerical results are included to illustrate the performance of the algorithm in the presence of large parameter errors and external disturbance.

I. Introduction

Deriving and solving dynamic equations are usually the initial step for the study of multi-body systems. The two most common methods are Newton-Euler equations and Lagrangian method. Using the Newton-Euler approach, calculations of constraint forces are required but must be eliminated to obtain the final equations of motion. The Lagrangian method is well constructed to automatically eliminate the constraint forces but requires extensive analytic or symbolic derivations.

Purely symbolic methods can lead to an explosive growth of algebraic expressions and corresponding lengthy computer codes, and as a consequence, the symbolic manipulation approaches are not attractive for nonlinear multi-body problems with many degrees of freedom. In recent years, automatic differentiation (AD) has been proposed for the equations of motion generation. Using chain rule-based evaluation, implemented by either operator overloading or source code transformation and automatic programming, AD generates exact numerical results for the derivative of coded functions while requiring only a modest additional user coding, which makes it a very promising tool in developing more versatile multi-body dynamics programme.¹

Object Oriented Coordinate Embedding Algorithm (OCEA) is an automatic differentiation environment. Its most important characteristics are embedded chain rule and operator overloading. Operator overloading is a recent idea which means we can define new computing rules for user-defined data structures. This is an important point because it makes the application-oriented OCEA very extensible. Higher level operators can be defined by programmers to reduce coding and minimize computations for specific applications. The data structure of a second-order OCEA variable is given by \( f := (f, \nabla f, \nabla^2 f) \), which means the computer automatically generates the first and second order partial derivatives \( \nabla f \) and \( \nabla^2 f \) at the same time when the scalar part of \( f \) is being computed. Refer to Tuner’s papers²-³ for a detailed mathematical description of OCEA. Since the algebra grows directly from differentiation of easily derived and coded position or velocity

---

*Graduate Assistant Research, Department of Aerospace Engineering, 3141-TAMU, Member AIAA  
†Adjunct Professor, Department of Aerospace Engineering, 3141-TAMU, Senior Member AIAA  
‡Distinguished Professor, Department of Aerospace Engineering, 3141-TAMU, Fellow AIAA

American Institute of Aeronautics and Astronautics
kinematic equations, and through invoking well-known dynamical recipes for equation of motion generation (e.g., Lagrange's equations, Gibbs-Appell equations, Kane's equations, etc.) it is natural to rely on OCEA for automation of the algebra and calculus in this process and avoid the hand derivation and coding of these expressions altogether. We also avoid the necessity of using a generic multi-body code with all of the specific assumptions and coordinate constraints that this entails. One need to code only position and velocity level kinematics, but notice that in deriving and coding these expressions, the analyst is free to choose any reference frames, coordinates, and to invoke assumptions desired to idealize/approximate the specific system. The structure of this approach is attractive, because one can efficiently explore a wide selection of analysis options with minimal derivation and coding. Mapping these decisions forward to employ OCEA also for the design of corresponding nonlinear model reference adaptive control laws is the main contribution of this paper. We also use OCEA to simulate the closed-loop controlled response of the system.

Griffith et al.\textsuperscript{1,4,5} used OCEA AD to formulate the equations of motion for flexible multi-body dynamical systems. Their formulations were for the system of multi-bodies translating and rotating in planar (2D) motion.\textsuperscript{1} Now we extend the previous results to a more general form for multi-body systems translating and rotating in three dimensional (3D) motion in this paper. The mass matrix follows from velocity level kinematics to play the key role in the derivation for the 3D case as for the 2D case. For 2D motions, the formulations are developed by recursively forming the position vectors locating the mass center of each body.\textsuperscript{4} For 3D motion, the direction cosine matrix between each body is an additional requirement. Another contribution we make in this paper is an adaptive control law for the multi-body systems which have uncertain parameters, and furthermore, this control law is based on the same kinematic descriptions that have been derived to generate the equations of motion. These improvements extend the previous OCEA AD work to a more general code for 3D automatic equations of motion generation and adaptive control law design process, which frees the analysts from deriving, coding, and validating the dynamic analysis and control formulations. These concepts are applied to the Stewart platform and the simulation results illustrate the performance of our approach.

The Stewart platform is a six degrees of freedom multi-body system. The geometry of the Stewart platform is shown in Fig. 1(a) which is called the Stewart platform (SPM) in most papers, where the platform is connected to six extensible legs by three spherical joint pairs, centered at $T_1$, $T_2$, and $T_3$. A more general parallel mechanism, which is a generalization of the SPM, and which is actually a less idealized version of the classical SPM, has six legs, connecting the platform to the base through six spherical pairs at $T_i$ (or $B_i$), and universal joints at $B_i$ (or $T_i$). In this paper, we focus on the classical Stewart platform, which is shown in Fig. 1(a), when we derive for the dynamic equations and design control law. To study the generalization of the SPM (Fig. 1(b)), only minor geometry description modifications are needed once we solved the dynamic and control problem for the classical Stewart platform (Fig. 1(a)).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{stewart_platform.png}
\caption{Geometry}
\end{figure}

Since Stewart proposed the original form of Stewart platform for ground simulation of aircraft flight motion in 1965,\textsuperscript{6} there has evolved a significant literature about this topic. However, compared with the extensive research on kinematics analysis, research papers on dynamics and control are relatively rare. Also most previous work developed the dynamic equations based on simplifications that ignored some or all leg
dynamics. Do and Yang used Newton-Euler method to solve the inverse dynamics of the platform assuming the legs were symmetrical and thin.\textsuperscript{7} Liu et al\textsuperscript{8} developed the equations of motion using Lagrangian approach under the assumption that each leg can be modeled by one moving point. Until in recent years, dynamics for the Stewart platform with a complete architecture and general parameters have been proposed but complicated symbolic derivations are usually involved. Dasgupta and Mruthyunjaya\textsuperscript{9} developed a complete formulation of the inverse dynamics problem through the Newton-Euler approach, which was shown to be well-suited for parallel computation, but the math model became very large and complex. In addition, implicitly, the appropriate idealized models, when used in a control formulation, rely on the controller robustness to compensate for model errors. Since our modeling goal is to support high precision control and performance simulation, previous simplifying assumptions are not appropriate. For example, Yung et al\textsuperscript{10} neglected the rotational contribution of the legs when they designed the controller. Some other control formulations for the Stewart platform also have their simplifying assumptions. Liu et al\textsuperscript{8} designed a tracking control law when only final position control was concerned. Nguyen\textsuperscript{11} proposed a direct adaptive tracking control algorithm on the assumption that the robot motion was slow compared to the controller adaption rate. In the current paper, we approach the controller design problem with the most general methodology to date. On the other hand, we must always anticipate model errors, and the formulation should be designed to remain stable in the presence of model errors and disturbances.

Using kinematic based Lagrangian approach with the help of OCEA, the dynamic equations for the Stewart platform, which consider all the bodies with general parameters, are derived conveniently compared to the previous work.\textsuperscript{12} Based on this general model, adaptive control laws are designed in this paper, which can account for the system parameter uncertainties, model errors and external disturbances. In the event of a change in modeling assumptions, the changes need only be accounted for at the kinematics level; the remainder of the developments are readily automated. We prove stable closed loop control for a large class of model errors and disturbances.

This paper is organized as followings. To complete the advantages of OCEA AD approach for dynamics and control of multi-body systems and make the derivation for the adaptive control law design easy to understand, in the next section, we rewrite some of the main results shown by Bai et al.\textsuperscript{12} Then, we design an adaptive control law based on our dynamic equation formulations. After that, we apply our methodology to the Stewart platform. Thereafter, the simulation results of the Stewart platform tracking problem are shown. Finally, we present our conclusions.

II. Dynamic Model Formulation

Bai et al\textsuperscript{12} proposed using a kinematic analysis approach to implement Lagrangian equations, and formulate a general code to automate generation of the equations of motion for 3D multi-body systems based on AD. Considering the minimal coordinate systems and unconstrained motion, Lagrangian equations of motion is

\[
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \mathbf{Q}
\]  

where the Lagrangian is defined as \( L = T - V \), \( T \) is the kinetic energy, \( V \) is the potential energy, \( \mathbf{q} \) is the generalized coordinate vector, \( \dot{\mathbf{q}} \) is the generalized velocity vector and \( \mathbf{Q} \) is the generalized force vector.

Introducing transformations in Eqs. 2 and 3, the mass center velocity and angular velocity of the \( i^{th} \) body is

\[
\mathbf{V}_i = \tilde{\mathbf{R}}_i = A_i(q)\dot{q} = \mathbf{B}_i(q)\dot{q}
\]

\[
\omega_i = B_i(q)\dot{q}
\]

where \( \mathbf{V}_i \) is the translational velocity vector for the \( i^{th} \) body and \( \omega_i \) is the angular velocity vector for the \( i^{th} \) body. The components of \( \mathbf{V}_i \) are taken in the inertial frame, whereas the components of \( \omega_i \) are taken in the \( i^{th} \) body frame; \( A_i(q) \) and \( B_i(q) \) are easily derived consistent with this choice. Suppose the model is an \( n \) body system and the number of generalized coordinates is \( m \). The kinetic energy \( T = \frac{1}{2} \sum_{i=1}^{n} (m_i V_i^T V_i + \omega_i^T I_i \omega_i) \) is expressed as:

\[
T = \frac{1}{2} \sum_{i=1}^{n} q^T (m_i A_i^T A_i + B_i^T I_i B_i) \dot{q} = \frac{1}{2} \dot{q}^T M(q) \dot{q}
\]
where

\[ M(q) = \sum_{i=1}^{n} (m_i A_i^T A_i + B_i^T I_i B_i) \]

(5)

From Eq. 5, we compute

\[ \frac{\partial M}{\partial q} = \sum_{i=1}^{n} \left( m_i \frac{\partial A_i}{\partial q} A_i + m_i A_i^T \frac{\partial A_i}{\partial q} + \frac{\partial B_i}{\partial q} I_i B_i + B_i^T I_i \frac{\partial B_i}{\partial q} \right) \]

(6)

two key terms needed in the Lagrangian equations are calculated from

\[ \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}} \right) = \frac{d}{dt} (M(q)\dot{q}) = M(q)\ddot{q} + \left( \sum_{j=1}^{m} \frac{\partial M}{\partial q_j} \dot{q}_j \right) \dot{q} \]

(7)

\[ \frac{\partial T}{\partial q} = \frac{1}{2} \dot{q}^T \frac{\partial M}{\partial q} \dot{q} \]

(8)

In Eq. 8, the value in the \( i \)th row is

\[ \left( \frac{\partial T}{\partial q} \right)_i = \frac{1}{2} \dot{q}_i^T \frac{\partial M}{\partial q} \dot{q} \]

(9)

After formulating all these terms, the Lagrangian equations are rewritten as:

\[ M(q)\ddot{q} + \left( \sum_{j=1}^{m} \frac{\partial M}{\partial q_j} \dot{q}_j \right) \dot{q} - \left( \frac{1}{2} \dot{q}^T \frac{\partial M}{\partial q} \dot{q} - \frac{\partial V}{\partial q} \right) = Q \]

(10)

All the derivations required in the Lagrangian equations are executed by OCEA in the background automatically. We mention, that even though we choose to write the partials of \( M(q) \) explicitly in Eq. 6, even these derivatives are routinely evaluated by OCEA, so only the simple equation for \( M(q) \) of Eq. 5 needs be coded along with \( V(q) \); all other partials in Eq. 10 are generated automatically by OCEA.

The major points are: by recursively forming the position vectors locating the mass centers of each body and the direction cosine matrix between each body and the common reference frame, kinematic descriptions \( A_i \) and \( B_i \) can be solved by OCEA directly, where \( A_i \) and \( B_i \) are constructed using Eqs. 11, 12 and 13. As a result, the numerical results of the motion are obtained automatically, including

\[ A_i = \frac{\partial R_i}{\partial q} = \frac{\partial \dot{R}_i}{\partial q} \]

(11)

\[ B_i = \frac{\partial \omega_i}{\partial q} \]

(12)

\[ [\omega_i^\times] = -\left( \frac{d}{dt} [I_i] \right) [I_i]^T \]

(13)

where \([L_i I]\) transforms vectors written in the \( L_i \) frame into vectors expressed in the inertial frame.

### III. Adaptive Control Law Derivation

#### 1. General Approach and Adaptive PD Control Law

A basic property of robotics that we exploit to formulate the adaptive control law is that the robot dynamic equation is linear in the mass parameters. Taking advantages of the formulations given in reference,\(^{12}\) this property can be easily proved and a general form of an adaptive controller for the multi-body systems is derived herein. As a first step, Eq. 10 is simplified as

\[ M(q)\ddot{q} + N_m(q, \dot{q}) \dot{q} + G(q) = Q \]

(14)
where

\[ G(q) = \frac{\partial V}{\partial q} \]  \hspace{1cm} (15)

To study the linear parametric model property of the dynamic equations, we need to analyze the mass matrix for every body. From Eq. 5, the mass matrix for each body is expressed as

\[ M^i(q) = m^i(A^i)^T A^i + (B^i)^T B^i \]  \hspace{1cm} (16)

Supposing the inertia matrix is diagonal, we can express the \( p^{th} \) row and \( q^{th} \) column terms in matrix \( M_i \) as

\[ M^i_{p,q} = m^i((A^i)^T A^i)_{p,q} + \sum_{k=1}^{3} (I^i(k)) B^i_{k,p} B^i_{k,q} \]  \hspace{1cm} (17)

where superscript \( i \) is used to denote the \( i^{th} \) body, \( k \) is used to denote the \( k^{th} \) diagonal terms in the inertial matrix, \( p \) and \( q \) are the row and column index for the matrix terms. We emphasize that diagonal inertial matrix is utilized here to express succinctly our derivation for the adaptive control law while our control method is not limited to the cases with symmetrical geometry.

Using Eq. 17, the mass matrix for each body is expressed as a linear combination of its mass and its three diagonal inertial values. Accordingly, the mass matrix and kinetic energy for the whole system can be formulated as a linear combination of system parameters. Because the potential energy is a linear function of \( \dot{q} \), we obtain

\[ V = 1/2 r^T M(q) r + 1/2 \dot{\phi}^T \Gamma^{-1} \dot{\phi} \]  \hspace{1cm} (19)

where

\[ r = \Lambda e + \dot{e} \]  \hspace{1cm} (20)

\[ \dot{\phi} = \dot{\phi} - \ddot{\phi} \]  \hspace{1cm} (21)

\[ e = q_d - q \]  \hspace{1cm} (22)

\[ \dot{e} = \dot{q}_d - \dot{q} \]  \hspace{1cm} (23)

\( q_d \) and \( \dot{q}_d \) are used to denote the desired position and velocity values, \( \dot{\phi} \) is the estimate of the parameter vector, \( \Lambda \) and \( \Gamma \) are positive, diagonal matrix. The auxiliary signal \( r \) can be considered as a filtered tracking error vector which frees the adaptive controller from depending on the measurement of \( \dot{q} \).\textsuperscript{13} Taking the time derivative of Eq. 19, we obtain

\[ \dot{V} = r^T M(q) \dot{r} + 1/2 r^T \dot{M}(q) r + \dot{\phi}^T \Gamma^{-1} \dot{\phi} \]  \hspace{1cm} (24)

Using Eqs. 18, 20 and 22, Eq. 24 is further expressed as

\[ \dot{V} = r^T (Y(\cdot) \phi - Q) + r^T (1/2 \dot{M}(q) - V_m(q, \dot{q})) r + \dot{\phi}^T \Gamma^{-1} \dot{\phi} \]  \hspace{1cm} (25)

where

\[ Y(\cdot) \phi = M(q)(\dot{q}_d + \Lambda \dot{e}) + N_m(q, \dot{q})(\dot{q}_d + \Lambda e) + G(q) \]  \hspace{1cm} (26)

and

\[ \cdot \equiv (q_d, \dot{q}_d, \dot{q}_d, e, \dot{e}) \]  \hspace{1cm} (27)

Another important property for the robotics equations that we are going to use is that \( N_m(q, \dot{q}) \) in Eq. 14 is frequently skew symmetric.\textsuperscript{13} Here we must be careful because the Coriolis and centripetal components are not skew symmetric. If the equations have a more
general velocity dependence, these terms must be added in the equations and further development should be explored to account for these terms. In the following derivations, we adopt

\[ N_m(q, \dot{q}) = \frac{1}{2}(M + U^T - U) \]  

(28)

where

\[ U = \dot{q}^T \frac{\partial M}{\partial q} \]  

(29)

In Eq. 29, the \( i^{th} \) row vector of matrix \( U \), which has the same dimension as \( M \), is

\[ (\dot{q}^T \frac{\partial M}{\partial q})_i = \dot{q}^T \frac{\partial M}{\partial q_i} \]  

(30)

Using the skew symmetric property, Eq. 25 is simplified to

\[ \dot{V} = r^T (Y(\cdot)\dot{\phi} - Q) + \dot{\phi}^T \Gamma^{-1} \dot{\phi} \]  

(31)

By choosing the control torque to be

\[ Q = Y(\cdot)\dot{\phi} + K_v \dot{r} \]  

(32)

and the adaptive update law for the parameters as

\[ \dot{\phi} = \Gamma Y^T r \]  

(33)

we obtain the final form of the time derivative of the Lyapunov function as

\[ \dot{V} = -r^T K_v r \]  

(34)

Now we can prove the tracking errors \( e \) and \( \dot{e} \) are asymptotically stable. Firstly, by choosing \( K_v \) in Eq. 34 positive definite, we can make \( \dot{V} \) negative semidefinite, which makes \( V \) upper bounded. Because \( M(q) \) is a positive definite matrix, we can say that \( r \) and \( \dot{\phi} \) are bounded from Eq. 19. According to the definition of \( r \), we can see \( e \) and \( \dot{e} \) are bounded. Since \( e, \dot{e}, r \) and \( \dot{\phi} \) are bounded, we can show \( \dot{r} \) is bounded. Secondly, since \( M(q) \) is lower bounded (in fact, the robot mass matrix is bounded above and below\(^{13} \)), we can state that \( V \) is lower bounded. Since \( V \) is lower bounded, \( \dot{V} \) is negative semidefinite, using Barbalat’s lemma,

\[ \lim_{t \to \infty} \dot{V} = 0 \]  

(35)

which means by Rayleigh-Ritz theorem\(^ {13} \)

\[ \lim_{t \to \infty} \dot{r} = 0 \]  

(36)

From the linear property of Eq. 20, we can state that

\[ \lim_{t \to \infty} e = 0 \]  

(37)

\[ \lim_{t \to \infty} \dot{e} = 0 \]  

(38)

While we have asymptotic global stability of tracking motion, we can not guarantee that \( \lim_{t \to \infty} \dot{\phi} = 0 \), which usually requires a persistently exciting condition.\(^ {13} \)

It is easy to show that for the ideal cases when the parameters are known and certain, the control torque in Eq. 32 takes the same form as the computed toque control in Eq. 39, which has been used in Bai et al.\(^ {12} \)

\[ Q = M(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + N(q, \dot{q}) \]  

(39)

where

\[ N(q, \dot{q}) = N_m(q, \dot{q}) \dot{q} \]  

(40)

and one of the choices for \( N_m \) is the form we defined in Eq. 28. The details to prove the equivalence are omitted here as the derivation is a standard practice.
2. Adaptive PID Control Law

In addition to the above adaptive control law, based on analogous ideas from linear control, we add an integration term in Eq. 20 to make the controller robust with respect to constant external disturbances. We have Eqs. 41 and 42 instead of Eq. 20 and we call it an adaptive PID controller.

\[ r = \Lambda e + \dot{e} + k_i \epsilon \]  
\[ \dot{e} = e \]  
\[ (41) \]

\[ \dot{\epsilon} = e \]  
\[ (42) \]

Equation 26 generalizes to

\[ Y(\cdot)\varphi = M(q)(\dot{q}_d + \Lambda \dot{e} + K_i e) + N_m(q, \dot{q})(\dot{q}_d + \Lambda e + k_i \epsilon) + G(q) \]  
\[ (43) \]

The control torque law and parameter adaptive rules have the same form as Eqs. 32 and 33.

3. Adaptive Control Law for Part Parameter Uncertainties

The above adaptive control law is designed based on the assumption that all the parameters are unknown. The methodology can also be applied to the system that only part of parameters are unknown. The dynamic Eq. 18 can be rewritten as

\[ M(q)\ddot{q} + N_m(q, \dot{q})\dot{q} + G(q) = W_1(q, \dot{q}, \ddot{q})\varphi_1 + W_2(q, q, \dot{q})\varphi_2 \]  
\[ (44) \]

where \( \varphi_1 \) is the vector of the parameters having a high certainty and \( \varphi_2 \) is the uncertain parameter vector. Using the same procedure as the previous derivation, the control torque in Eq. 32 changes to the form

\[ Q = Y_1(\cdot)\varphi_1 + Y_2(\cdot)\hat{\varphi}_2 + K_e r \]  
\[ (45) \]

and the adaptive update law for the parameters is

\[ \dot{\hat{\varphi}}_2 = \Gamma Y_2^T r \]  
\[ (46) \]

IV. Dynamics and Adaptive Control Law for the Stewart Platform

Refer to Bai et al\cite{12} for a detailed development of the equations of motion generation for the Stewart platform. By recursively locating the mass center and the direction cosine matrix of the platform and each leg, a complete dynamics model was developed. They utilized the Lyapunov method to design a computed-torque control law for the Stewart platform, which was based on the assumption that all the system parameters are exactly known. Now using the approach proposed in the previous section, we design an adaptive control law for the Stewart platform. The tracking errors are shown to be asymptotically stable under uncertain parameters and bounded disturbances.

For the Stewart platform, we assume all mass parameters

\[ \varphi = \{m_{top}, I_{top1}, I_{top2}, I_{top3}, m_{lowleg}, I_{lowleg1}, I_{lowleg2}, I_{lowleg3}, m_{upleg}, I_{upleg1}, I_{upleg2}, I_{upleg3}\} \]  
\[ (47) \]

as a 12 \times 1 unknown parameter vector. The major step for the adaptive controller design lies in solving for \( W \), which is a matrix of system kinematics and can be easily obtained from Eq. 17 based on OCEA. For the systems that only part of parameters are unknown, Eqs. 45 and 46 can also be applied.

In addition, the actuator input force \( f \) and the generalized force \( Q \) utilized in the previous derivation are related by the following way. Here we choose the mass center location of the platform \( R_c = [X_c, Y_c, Z_c]^T \) and the three Euler angles \( \theta_1, \theta_2, \theta_3 \) of the platform as the generalized coordinates and \( q = [X_c, Y_c, Z_c, \theta_1, \theta_2, \theta_3]^T \). Since there is a unique mapping from \( q \) to the leg length vector \( l = [l_1, l_2, l_3, l_4, l_5, l_6]^T \), we suppose there is a function \( g \) such that

\[ l = g(q) \]  
\[ (48) \]

For infinitesimal displacements of actuators \( \delta l \) and motion of the platform \( \delta q \), we have relationship:

\[ \delta l = J\delta q \]  
\[ (49) \]
where \( J = \frac{\partial l}{\partial q} = J(q) \). Using the principle of virtual work, we have

\[
f^T \delta l = Q^T \delta q
\]  

(50)

Substituting Eq. 49 into Eq. 50, we obtain

\[
Q = J^T f
\]  

(51)

Because the mapping Jacobian matrix \( J \) is a first order partial derivative matrix and can be obtained from OCEA directly, we design the generalized force \( Q \) first and then solve for the actuator input force \( f \) from Eq. 51.

We note that for the dynamics and control of the Stewart platform, there exists some disagreement about whether the dynamic equations should be formulated in the configuration work space based on platform rigid body coordinates \( q \) or in the joint space using leg lengths \( l \) as the coordinate vector. Using \( q \), the equations of motion are simplified to Eq. 52

\[
M(q)\ddot{q} + N(q, \dot{q}) = J^T f
\]  

(52)

The problem with this approach is that the measurement of the position and orientation of the upper platform is difficult to obtain directly. Usually the leg lengths and velocity are measured by sensors in the leg actuators, then the position and velocity of the platform are derived using Eq. 48 and 53.

\[
\dot{l} = J \dot{q}
\]  

(53)

If we choose the leg lengths \( l \) as the generalized coordinates, we can use the Jacobian matrix \( J = J(q) \) to transform the equations of motion Eq. 52 to link space as

\[
\overline{M} \dot{l} + \overline{N} = f
\]  

(54)

where

\[
\overline{M} = J^{-T} M J^{-1}
\]

\[
\overline{N} = J^{-T} N - J^{-T} M J^{-1} J \dot{J} J^{-1} l
\]  

(55)

Although the leg lengths are easy to be measured accurately, since the Jacobian matrix is an explicit function of the position and orientation of the upper platform, while there is not a closed-form mapping from the leg lengths to the platform coordinates, to solve for Eqs. 55 and 56, we still need to calculate \( q \) from \( l \). Because the desired trajectory is almost always planned in the platform coordinates \( q \), we choose to use the usual platform rigid body coordinate vector \( q \) to formulate the dynamic equations and design the controller.

V. Simulation Results

In the simulations, the Stewart platform is designed to track a reference path, which is constructed by a rest-to-rest third order spline function.

\[
q_r(t) = q_{r_0} + f(t)(q_{r_f} - q_{r_0})
\]

(57)

\[
f(t) = \tau^2(3 - 2\tau)
\]

(58)

\[
\tau = t/T_f
\]

(59)

where \( q_{r_0} \) is the initial state, \( q_{r_f} \) is the final desired condition, \( t \) is the current time and \( T_f \) is the required time to finish the maneuver. This is the simplest reference trajectory used in the example simulations. The formulation is not restricted to this reference motion.

To show the performance of our adaptive control method, we designed three kinds of simulations. For all these simulations, we suppose the system parameters have 30% errors. The results show that the adaptive controller we derived can make the tracking errors asymptotically stable, the adaptive PID control law makes the system robust under external disturbances, and the adaptive control law works good for systems with part parameter uncertainty.

Firstly, we compare the tracking errors when we use adaptive control law with Eqs. 32 and 33 and computed-torque control in Eq. 39. Fig. 2 shows that the adaptive control law makes the tracking errors
asymptotically stable while computed-torque control results in tracking errors under the supposed parameter uncertainties. We mention that the turning parameters are selected in such a way that we can achieve the tracking in about 10 seconds, which is the parameter we used to design the tracking profile, also we tune to make the largest control efforts of the adaptive controller and the computed-torque controller with almost the same magnitude.

![Tracking Errors with Parameter Uncertainty](image)

Figure 2. Tracking Errors with Parameter Uncertainty

Secondly, we impose an external constant disturbance of \([20, 20, 20, 10, 10, 10] N \cdot m\) and compare the tracking errors for the adaptive PID controller and the adaptive PD controller. The results are shown in Fig. 3, while the errors of the adaptive PID controller and the computed torque controller are shown in Fig. 4.

Thirdly, when the only unknown parameter is the mass of the platform, we compare the adaptive control results with the computed torque results in Fig. 5.

VI. Conclusion

In this paper, we present a novel approach to use kinematic description-based Lagrangian method to automatically generate three dimensional multi-body dynamic equations and design an adaptive controller for the systems with parameter uncertainties. This methodology is utilized here for automated dynamic analysis, control design, and simulation of the Stewart platform dynamics. This paper makes several evident contributions: (i) In the structure and stability guarantees of our controller. (ii) The highly automated means we have used to develop and implement the nonlinear equations of motion, and the control law. (iii) The use of OCEA AD to accomplish (i) and (ii), and to numerically validate the stable closed loop performance. The methodology is quite general. Alternate modeling assumptions may be quickly accommodated by changing a few lines of kinematics code; the rest of the formulation is fully automated. The main development is a demonstration that by only focusing on the kinematic description of the system, nonlinear motion and an adaptive controller for the multi-body systems can be generated automatically with the help of automatic differentiation, which saves laborious derivation, coding and validation work. Furthermore, we need not specialize an exceedingly general hand-coded version of the multi-body equations of motion, with specific rules for coordinates, reference frames and means to handle constraints. In addition, we note that dynamics
and control for the systems of redundant manipulators and those that have fewer controls than the desired degrees of freedom can also be considered to generate a more general version of this code but are not discussed in this paper. These topics will be studied in the future.

References

Figure 4. Tracking Errors with Parameter Uncertainty and Disturbance

Figure 5. Tracking Errors When only Platform Mass is Unknown