Robust Adaptive Control of the Stewart-Gough Robot in the Task Space

Eugenio Yime
Universidad del Atlántico
Klm 7 ant. via Pto. Colombia
Barranquilla, Colombia
eyime@uniatlantico.edu.co

Roque Saltaren
Universidad Politécnica de Madrid
Calle Jose Gutierrez Abascal, 2
Madrid, 28006, Spain
rsaltaren@etsii.upm.es

James Diaz
BMT Designers and Planners Inc.
2120 Washington Blvd Suite 200
Arlington, VA, 22204, USA
jdiaz@dandp.com

Abstract—This paper deals with the robust adaptive control of the Stewart-Gough platform based on the task space dynamic equations. These equations were obtained using the virtual work approach, and include all the thirteen bodies presented in the platform, i.e. two bodies per each leg and the upper ring. The performance of the robust adaptive control law is evaluated using a sinusoidal path for position and orientation of the upper ring. Simulation were carried out to show the performance of the controller.

I. INTRODUCTION

The Stewart-Gough platform is a well studied parallel robot in kinematics. However, it is not well explored in dynamics and control, [1]. In the last ten years, there have been some remarkable attempts to control the Stewart-Gough robot in the task space, some of them are [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. The control strategies, based on the dynamic model of the Stewart-Gough robot, vary from a simplistic dynamic model regarding only the end effector, [2], [6], [10], [15], or a simplified dynamic model with some considerations, [11], [14], to a complete dynamic model based on the Newton-Euler or Lagrangian formalism, [3], [5], [8], [13]. In [3] an adaptive control law is formulated based on the full dynamics obtained with the Lagrangian formalism. In [5] the dynamic model is obtained using the Newton-Euler approach. On the other hand, the control law is obtained by combining the inverse dynamics and $H_\infty$ controllers. A full dynamic model was obtained by Ting et. al. using the Newton-Euler approach, thus an adaptive control law was derived [8]. Finally, in [13] the dynamic equations are obtained using the Lagrangian approach. Similarly, the control law is derived based on inverse dynamics.

A robust adaptive controller for the Stewart-Gough robot is presented in this paper. This control strategy is based on a complete dynamic model derived from the virtual work approach. The aforementioned procedure presents two notable advantages respect to Newton-Euler and Lagrangian formalisms. The first advantage is that this strategy does not need an inverse acceleration analysis, which is required in the case of the Newton-Euler formalism. The second advantage is that this strategy does not need to compute cumbersome terms in mass and Coriolis obtained after differentiating energy functions, such as those required by the Lagrangian formalism. On the other hand, its main disadvantage is the requirement to compute the link Jacobian matrices and its differential, a minimal requirement if it is compared to the work required by the Newton-Euler and Lagrangian approaches. Finally, the regressor matrix needed in the robust adaptive control law can be computed straightforward, using symbolic software as MATLAB or MAPLE, after obtaining a valid dynamic model.

II. DYNAMIC MODEL

The nomenclature illustrated in Figure 1 is used for computing the dynamic model, where the vector $r_p$ define the position of the end effector and the angles $\alpha_i$ and $\alpha_j$ define the position of the spherical joint, in the upper ring, and the universal joint, in the lower ring, respectively. The vectors $r_i$ and $r_j$ define the location of the spherical and universal joint in the upper and lower ring. The unit vector $u_{3,i}$ is along the first rotational union in the lower universal joint, $u_{3,i}$ is a unit vector parallel to the translational actuator, and $u_{2,i}$ is a unit vector perpendicular to the vectors $u_{3,i}$ and $u_{1,i}$. The unit vector $u_{4,i}$ is perpendicular to the vectors $u_{2,i}$ and $u_{3,i}$, so $u_{4,i}$, $u_{2,i}$ and $u_{3,i}$ form a Cartesian frame, and the vector $d_i$ is located along the $i-th$ leg. The final result of the

Figure 1. Vector defined for computing the dynamic model.

...
dynamic model is described in Eqs. (1)-(6).
\[ J_p^T \tau_{act} + \tau_{ext,p} = M_t \ddot{x}_p + C_t \dot{x}_p + g_t \]
\[ \ddot{x}_p = \left[ \begin{array}{c} \ddot{v}_p \\ \ddot{\omega}_p \end{array} \right] \]
\[ M_t = M_p + \sum_i J_i^T M_i J_i \]
\[ C_t = C_p + \sum_i (J_i^T C_i J_i + J_i^T M_i J_i) \]
\[ g_t = g_p + \sum_i J_i^T g_i \]

where \( J_p \) is the Jacobian of the upper ring; \( \tau_{act} \) are the forces produced by the linear actuators; \( \tau_{ext,p} \) are the external torque and force applied in the upper ring; \( \dot{x}_p \) and \( \ddot{x}_p \) are the acceleration and velocity vectors of the upper ring in the task space; \( M_t \) and \( M_i \) are the total mass matrix and the link mass matrix in the task space, respectively; \( C_t \) and \( C_i \) are the task space total and link Coriolis matrices, respectively; \( g_i \) and \( g_t \) are the vectors associated with gravity and they are applied in the center of gravity of the end effector and each link, respectively; and \( J_i \) is the Jacobian of \( i \)-th link.

The expressions for the matrices \( M_p \) and \( M_i \), for simplicity denoted \( M_j \), are given in Eq. (7).
\[ M_j = \left[ \begin{array}{cc} m_j & 0 \\ 0 & I_j^T \end{array} \right] \]

where \( m_j \) and \( I_j^T \) are the \( j \)-link mass and inertia, respectively.

On the other hand, the expression for matrices \( C_i \) and \( C_p \), symbolized \( C_j \), is given by Eq. (8).
\[ C_j = \left[ \begin{array}{cc} 0 & 0 \\ 0 & S(\omega)_j I_j^T \end{array} \right] \]

where the operator \( S(\cdot) \) is used to represent the cross product,
\[ S(\omega) = \left[ \begin{array}{ccc} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{array} \right] \]

In a similar way to \( C_j \), the expression for vectors \( g_p \) and \( g_i \) is given by Eq. (10),
\[ g_j = \left[ \begin{array}{c} -g \\ 0 \end{array} \right] \]

Finally, in the case of the matrices \( J_p \) and \( J_i \), the former is the robot Jacobian, and it is given in Eq. (11),
\[ J_p = \left[ \begin{array}{cccc} u_{2,1} & -u_{3,1} & S(1,1) \end{array} \right] \]

and the latter is the Jacobian link matrix, where there are two cases for the whole robot, the first case is for the lower body in the prismatic joint,
\[ J_{1,i} = \left[ \begin{array}{ccc} -S(c_i,1) K_{1,i} & S(c_i,1) K_{1,i} & S(r_i) \\ K_{1,i} & -K_{1,i} & 0 \end{array} \right] \]

where \( c_i,1 \) is the position vector of the center of gravity of the first link and usually its value is \( c_i,1 = L_i u_{3,i} / 2 \).

The jacobian matrix for the second link, which is related to the moving part of the translational actuator, is given by Eqs. (17) and (18).
\[ J_{2,i} = \left[ \begin{array}{ccc} -K_{2,i} & K_{2,i} S(r_i) \\ K_{1,i} & -K_{1,i} S(r_i) \end{array} \right] \]
\[ K_{2,i} = S(c_2,i) K_{1,i} + u_{3,i} u_{3,i}^T \]

where \( c_2,i = d_i - L_2 u_{3,i} / 2 \) is the position vector of the center of gravity of the second link.

In the expression for \( C_i \) it is showed the first derivative of the jacobian link \( J_i \) which also have two components. The first component is related with \( J_{1,i} \), and it is given by Eqs. (19)-(23).
\[ J_{1,i} = \left[ \begin{array}{ccc} -S(c_i,1) K_{1,i} + S(c_i,1) K_{1,i} \\ K_{1,i} & -K_{1,i} S(r_i) \end{array} \right] \]
\[ S(c_i,1) K_{1,i} + S(c_i,1) K_{1,i} S(r_i) + S(c_i,1) K_{1,i} S(r_i) \]
\[ -K_{1,i} S(r_i) - K_{1,i} S(r_i) \]

\[ J_{2,i} = \left[ \begin{array}{ccc} -K_{2,i} & K_{2,i} S(r_i) + K_{2,i} S(r_i) \\ K_{1,i} & -K_{1,i} S(r_i) - K_{1,i} S(r_i) \end{array} \right] \]

The expressions of Eqs. (3) to (25) are used for building up a procedure for computing the vector \( g_i \) and the matrices \( J_p, M_t \) and \( C_t \). This procedure was validated with MSC.ADAMS in [17].

III. DYNAMIC MODEL PROPERTIES

The dynamic model of the robot must have three important properties which help to obtain a robust adaptive control law. These properties are the symmetry of the mass matrix, \( M_t \), the skew symmetric properties of the relation \( M_i - 2C_i \), and the linearization in the dynamic parameters of the dynamic equation.
A. Symmetry of the Mass Matrix

Symmetry of the mass matrix can be easily obtained after transposing the matrix \( M_t \), Eq. (4),

\[
M_t^T = M_p^T + \sum_i J_i^T M_i J_i = M_t \tag{26}
\]

B. Skew Symmetric Property

The equations (4) and (5) can be replaced in \( M_t - 2C_t \) to verify the skew symmetric property,

\[
M_t - 2C_t = M_p + 2 \sum_i J_i^T C_i J_i - 2 \sum_i J_i^T M_i J_i
= -2C_p - 2 \sum_i J_i^T C_i J_i
\]

\[
M_t - 2C_t = -(M_t - 2C_t)^T \tag{27}
\]

C. Linearity in Dynamic Parameters

The linearization search for a regressor matrix, \( Y \), where the dynamic equation (1) can be represented by Eq. (28),

\[
J_p^T \tau_{act} = Y(\bar{x}, \bar{x}) \varphi + \tau_d \tag{28}
\]

where \( \tau_d \) is an external disturbance torque and \( \varphi \) are the robot dynamic parameters, which in the case of the Stewart-Gough platform are 53 variables formed by four parameters for each of the thirteen bodies in the parallel robot, i.e. the mass, \( m_i \), and inertia parameters, \( I_{xx,i}, I_{yy,i}, I_{zz,i} \), as is given in Eqs. (29) and (30)

\[
\varphi = [\varphi^T_1 \ldots \varphi^T_{12}]^T \tag{29}
\]

\[
\varphi_i = [m_i, I_{xx,i}, I_{yy,i}, I_{zz,i}]^T \tag{30}
\]

The first step to linearize the dynamic equation is to rewrite Eq. (1) as,

\[
J_p^T \tau_{act} = M_p \bar{x}_p + (\sum_i J_i^T M_i J_i) \bar{x}_p +
C_p \bar{x}_p + (\sum_i J_i^T C_i J_i) \bar{x}_p +
(\sum_i J_i^T M_i J_i) \bar{x}_p +
g_p + \sum_i (J_i^T g_i) + \tau_d \tag{31}
\]

then, to introduce some regressor matrices for each term,

\[
J_p^T \tau_{act} = Y_{M_p} \varphi_p + \sum_i (Y_{M_i}(\varphi_i)) +
Y_{C_p} \varphi_p + \sum_i (Y_{C_1,i}(\varphi_i)) +
\sum_i (Y_{C_2,i}(\varphi_i)) +
Y_{g_p} \varphi_p + \sum_i (Y_{g_i}(\varphi_i)) + \tau_d \tag{32}
\]

and finally group the regressor matrices into one big matrix,

\[
J_p^T \tau_{act} = Y(\bar{x}, \bar{x}) \varphi + \tau_d \tag{33}
\]

\[
Y_p = Y_{M_p} + Y_{C_p} + Y_{g_p} \tag{34}
\]

\[
Y_i = Y_{M_i} + Y_{C_1,i} + Y_{C_2,i} + Y_{g_i} \tag{35}
\]

\[
Y = \left[ \begin{array}{c} Y_p \ Y_1 \ldots \ Y_{12} \end{array} \right] \tag{36}
\]

\[
\varphi = \left[ \begin{array}{c} \varphi^T_p \ \varphi^T_1 \ldots \ \varphi^T_{12} \end{array} \right]^T \tag{37}
\]

The expressions for (34), (35) and (36) can be obtained symbolically in MATLAB or MAPLE. The results are in the MATLAB files submitted with the paper.

IV. ADAPTIVE CONTROL

A. Orientation Control using Quaternion

Quaternion was selected for representing the error in attitude, some similar approaches are in Yuan, [18], Caccavale et al., [19], Xian et al., [20], and Nakanishi et al., [21]

Being a unit quaternion defined by,

\[
\tilde{e} \in \mathbb{R}^4, \| \tilde{e} \| = 1, \tilde{e} an \ e^T \tilde{e} = 1
\]

\[
t_\text{quat} \in \mathbb{R}^3, \| t_\text{quat} \| = 1, t_\text{quat} = \tilde{e} e^T, \tilde{e} an \ e^T \tilde{e} = 1
\]

\[
\tilde{e} = \left[ \begin{array}{c} \tilde{e}_0 \ e^T \end{array} \right] = [e_0, e_1, e_2, e_3]^T
\]

\[
1 = q^T q = e_0^2 + e_1^2 + e_2^2 + e_3^2
\]

the rotation matrix is obtained by,

\[
R = \left[ \begin{array}{cccc} e_0^2 + e_1^2 - 0.5 & e_1 e_2 - e_0 e_3 & e_1 e_3 + e_0 e_2 & e_1 c_3 - e_0 c_1 \\
0.5 & e_0^2 + e_3^2 - 0.5 & e_2 e_3 - e_0 e_1 & e_2 c_3 + e_0 c_1 \\
e_1 e_2 + e_0 e_3 & e_0^2 + e_3^2 - 0.5 & e_2 e_3 + e_0 e_1 & e_2 c_3 - e_0 c_1 \\
e_1 e_3 - e_0 e_2 & e_1 e_2 - e_0 e_3 & e_2 e_3 - e_0 e_1 & e_0^2 + e_1^2 - 0.5 \end{array} \right]
\]

and the error in attitude, using rotation matrix or \( R_e \), is calculated after premultiplying the transpose of the actual rotation matrix, \( R_a \), to the desired rotation matrix, \( R_d \) as is given in Eq. (41).

\[
R_e = R_a^T R_d \tag{41}
\]

The quaternion, \( q_e \), associated with \( R_e \) is,

\[
q_e = [\tilde{e}_e, \tilde{e}^T] \tag{42}
\]

\[
e_0 = -e_0 a e_{0,d} - e_a e_{0,d} \tag{43}
\]

\[
e = -e_0 a e_{0,d} + e_0, a e_{0,d} - e a \tag{44}
\]

where \( q_e = [e_x, e_y]^T \) is the quaternion related with \( R_a \), and the same is for \( q_{ad} = [e_{ad}, e_{ad}^T] \) and \( R_d \).

Since the quaternion for two aligned frames is a unit quaternion, i.e. \( R_e = I \) and \( q_e = [1, 0, 0, 0]^T \), the vector \( \tilde{e} \) is usually taken as a measurement of the error in attitude.

The relation between the angular velocity error, \( \omega_e = \omega_{ad} - \omega_a \) and the quaternion velocity error, \( q_e \), is given by

\[
q_e = \frac{1}{2} \left[ \begin{array}{c} \tilde{e} \\
e_0 I_{3x3} + S(\tilde{e}) \end{array} \right] \omega_e \tag{45}
\]

\[
q_e = U_e \omega_e \tag{45}
\]
B. Control Law and Stability Analysis

This section uses the approach of Li and Slotine, [22], as explained by Lewis et al., [23].

Being the end effector translation error defined by,
\[
\mathbf{r} = [x_d, y_d, z_d]^T - [x_p, y_p, z_p]^T
\]  
and the magnitude of the quaternion error,
\[
\mathbf{q} = [1, 0, 0, 0]^T - \mathbf{q}_e
\]  
the spatial error in robot position is then,
\[
\mathbf{x} = [\mathbf{r}^T, \mathbf{e}^T]^T
\]  
which has for first differential,
\[
\dot{\mathbf{x}} = \begin{bmatrix}
I_{3	imes3} & 0 \\
0 & \frac{1}{2} \mathbf{e}_0 I_{3	imes3} + \frac{1}{2} \mathbf{S} (\hat{\mathbf{e}})
\end{bmatrix} \dot{\mathbf{x}}_p
\]  
(49)
\[
= \mathbf{U}_x \dot{\mathbf{x}}_p
\]  
(50)
it is preferred \( \omega_p \) instead of \( \mathbf{e} \) for computing velocity errors in the end effector,
\[
\dot{\mathbf{x}}_p = \dot{x}_{p,d} - \dot{x}_p
\]  
(51)
then the differential will be used for errors in linear and angular acceleration,
\[
\ddot{\mathbf{x}}_p = \ddot{x}_{p,d} - \ddot{x}_p
\]  
(52)
Finally, the vector \( \mathbf{s} \), similar to a sliding surface, is defined as,
\[
\mathbf{s} = \dot{\mathbf{x}}_p + \Lambda \dot{\mathbf{x}}
\]  
(53)
and its differential will be,
\[
\dot{\mathbf{s}} = \ddot{\mathbf{x}}_p + \Lambda \ddot{\mathbf{x}}
\]  
(54)

In order to obtain the control and the adaptive laws, it is proposed the following candidate Lyapunov function,
\[
V = \frac{1}{2} s^T M_1 \mathbf{s} + \frac{1}{2} \dot{\mathbf{s}}^T \mathbf{G} \dot{\mathbf{s}}
\]  
(55)
where \( \dot{\mathbf{s}} = \mathbf{s} - \dot{\mathbf{s}} \) is the error in parameters. The Lyapunov function is positive definite due to symmetry in \( M_1 \) and positiveness in \( \mathbf{G} \).

Differentiating the Lyapunov function,
\[
\dot{V} = s^T M_1 \ddot{\mathbf{s}} + \frac{1}{2} \dot{\mathbf{s}}^T \mathbf{G} \dot{\mathbf{s}} - \dot{\mathbf{s}}^T \mathbf{G} \dot{\mathbf{s}}
\]  
(56)

it can be further simplified if the following relations are used,
\[
\mathbf{Y}(\dot{\mathbf{x}}_d, \dot{\mathbf{x}}_d, \dot{\mathbf{x}}, \dot{\mathbf{x}}) \phi = \mathbf{M}_1(\dot{\mathbf{x}}_d + \Lambda \dot{\mathbf{x}}) + \mathbf{C}_1(\dot{\mathbf{x}}_d + \Lambda \dot{\mathbf{x}})
\]  
(57)
\[
\mathbf{M}_1 \ddot{\mathbf{s}} = \mathbf{Y}(\dot{\mathbf{x}}_d, \dot{\mathbf{x}}_d, \dot{\mathbf{x}}, \dot{\mathbf{x}}) \phi - \mathbf{C}_1 \mathbf{s}
\]  
(58)
replacing (58) in (56),
\[
\dot{V} = s^T (\mathbf{Y}(\dot{\mathbf{x}}_d, \dot{\mathbf{x}}_d, \dot{\mathbf{x}}, \dot{\mathbf{x}}) \phi - \mathbf{J}_p^T \tau_{act} + \tau_d) - \dot{\phi}^T \mathbf{G} \dot{\phi}
\]

Defining the control law by,
\[
\tau_{act} = \mathbf{J}_p^T (\mathbf{Y}(\dot{\mathbf{x}}_d, \dot{\mathbf{x}}_d, \dot{\mathbf{x}}, \dot{\mathbf{x}}) \phi + \mathbf{K} \mathbf{s} + k_d \text{sgn} (\mathbf{s}))
\]  
(59)
the result is,
\[
\dot{V} = -s^T \mathbf{K} \mathbf{s} + s^T (\dot{\mathbf{x}}_d, \dot{\mathbf{x}}_d, \dot{\mathbf{x}}, \dot{\mathbf{x}}) \phi - \dot{\phi}^T \mathbf{G} \dot{\phi} +
\]
(70)
\[
\tau_d = A_d \text{random}(1, -1)
\]  
(70)
Finally, if the adaptive law is defined by,
\[
\dot{\phi} = \Gamma^{-1} \mathbf{Y}^T \mathbf{s}
\]  
(60)
and \( k_d \) is,
\[
k_d > \max \{|\tau_{d,i}|\}
\]  
(61)
equation (56) is reduced to,
\[
\dot{V} \leq -s^T \mathbf{K} \mathbf{s}
\]  
(62)

The system stability in a Lyapunov-like sense can be proved using the Barbalat’s Lemma. Since \( \dot{V} \) is lower bounded and uniformly continuous, \( \dot{V} \leq 0 \), then \( V \) tends to zero as time increase, \( V \to 0 \) as \( t \to \infty \). Thus \( s, \dot{\phi} \to 0 \) as \( t \to \infty \). However, we cannot prove asymptotic stability of the whole state, since \( \dot{\phi} \) is only guaranteed to be bounded.

V. SIMULATION

A. Path Planning

The robot model was built in a Simulink s-function in \( m \) using the dynamics equations of section II making possible to obtain the end effector path for a set of forces applied in the actuators. Additionally, the end effector path planning was built using a sinusoidal motion, both for translation and orientation, the equations are,
\[
\mathbf{r} = r_0 + A_s \mathbf{u}_s \sin(ut)
\]  
(63)
\[
\theta = A_o \sin(ut)
\]  
(64)
\[
\mathbf{q} = \begin{bmatrix}
\cos(\theta/2) \\
\mathbf{u}_o \sin(\theta/2)
\end{bmatrix}
\]  
(65)
where \( r_0 \) is the initial position of the end effector, \( A_s \) and \( A_o \) are constants that define the motion amplitude, \( \mathbf{u}_s \) and \( \mathbf{u}_o \) are unit vectors along the linear motion and angular velocity, respectively, and \( w \) is the periodic motion frequency.

The equations for velocity and acceleration are obtained after differentiating the previous one,
\[
\dot{\mathbf{r}} = A_s \mathbf{u}_s \cos(ut)
\]  
(66)
\[
\mathbf{r} = -A_s w^2 \mathbf{u}_s \sin(wt)
\]  
(67)
\[
\omega = A_o \mathbf{u}_o \cos(ut)
\]  
(68)
\[
\dot{\omega} = -A_o w^2 \mathbf{u}_o \sin(ut)
\]  
(69)

In Table II are grouped the values used in the path planning simulations.

B. Disturbances

The external disturbances, \( \tau_d \), are applied to the end effector and they were modeled with the function given in Eq. (70),
\[
\tau_d = A_d \text{random}(1, -1)
\]  
(70)
where \( A_d \) is the amplitude and it has a value of 1 N for force and 10 mNm for torque.
C. Robot Model

The robot model was built using the nomenclature showed in Figure 1. In Table I are the values of the variables $\alpha_i, \alpha_j$, which define the position of each joint in the upper and lower rings. This table also contains the radius of the rings, $r_{lower}$ and $r_{upper}$, and the length of the two links located in a leg, $L_1$ and $L_2$.

D. Controller

Table II has the constant values used in the controller, which are $\Lambda$ in (53), $K$ in (59) and $\Gamma$ in (60). Additionally, the mass and inertia robot parameters, used for building a start value for $\Phi$ in (59), are a 10% lower than the real values located in Table I.

E. Numerical Results

The numerical simulation starts with an initial error in the position of the end effector, allowing in this way a more insight of the controller performance. The error is 20mm for each axis, so while the robot starts in $[20, 650, 20]^T$, the path planning has an initial position of $r_0 = [0, 630, 0]^T$.

The simulation results are shown in Figure 2 to 4. Figure 2 shows the actuators forces versus time. The error in position and orientation, are illustrated in Figure 3.

The maximum steady state values of the position errors are $[0.02, 0.02, 0.08]$ mm after 7s. For orientation case, the maximum steady state error in quaternion, $||\hat{q}_e||_\infty$, has a value of $3e-03$ after the same time 7s.

The linear and angular error behavior are illustrated in Figure 4. The maximum absolute error respect to the linear velocity, $||\dot{r}||_\infty$, is less than 0.4 mm/s after 4s as is shown in Figure 4(a). Respect to the angular velocity error, Figure 4(b), it is obtained a maximum absolute steady state value, $||\dot{\omega}||_\infty$, of 0.01mm/s after 8s.

Velocity errors are shown in Figure 4, where Figure 4(a) shows the errors respect to the linear velocity, and Figure 4(b) shows the errors respect to angular velocity. The maximum absolute error in the linear velocity, $||\dot{r}||_\infty$, is observed to be less than 0.4 mm/s after 4s.

Figure 5 shows the phase plane respect to the errors in linear velocity, Figure 5(a), and respect to quaternions errors, Figure 5(b). The Figure 5(a) is a plot of linear velocity error, $\hat{r}$, versus the linear position error, $r$. The initial conditions for the three plots were $(-20mm, -10mm/s)$ in $x$, $(-20mm, 65mm/s)$ in $y$ and $(-20mm, -10mm/s)$ in $z$. Figure 5(b) shows the plots of velocity quaternion errors versus the quaternion error. The initial conditions were $(0, 0.36)$ for $(\epsilon_0, \epsilon_0)$, $(0.4, 0)$ for $(\epsilon_1, \epsilon_1)$, $(0, -0.4)$ for $(\epsilon_2, \epsilon_2)$ and $(0, 0.36)$ for $(\epsilon_3, \epsilon_3)$. The convergence to the origin $(0, 0)$ is clearly observed in both figures.

VI. Conclusions

In this paper a robust adaptive control law of the Stewart-Gough parallel robot was developed. The main contribution of this work, as well as its main difference with previous elaborated ones, is that the robust adaptive control law was obtained based on the full task space dynamic equations of the robot using the virtual work formulation, which means that it was taking into account all the dynamic contribution of the thirteen bodies presents in the Stewart-Gough robot,

<table>
<thead>
<tr>
<th>variable</th>
<th>value</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{lower}$</td>
<td>385</td>
<td>mm</td>
</tr>
<tr>
<td>$r_{upper}$</td>
<td>285</td>
<td>mm</td>
</tr>
<tr>
<td>$\alpha_j$</td>
<td>$[-15, 15, 105, 135, -135, -105]^T$</td>
<td>degrees</td>
</tr>
<tr>
<td>$\alpha_i$</td>
<td>$[-45, 45, 75, 165, -165, -75]^T$</td>
<td>degrees</td>
</tr>
<tr>
<td>$L_1$</td>
<td>418.1</td>
<td>mm</td>
</tr>
<tr>
<td>$L_2$</td>
<td>464.5</td>
<td>mm</td>
</tr>
<tr>
<td>$m_p$</td>
<td>27.7</td>
<td>kg</td>
</tr>
<tr>
<td>$m_1$</td>
<td>0.6</td>
<td>kg</td>
</tr>
<tr>
<td>$m_2$</td>
<td>1.14</td>
<td>kg</td>
</tr>
<tr>
<td>$I_{xx,p}$</td>
<td>$1.13 \times 10^6$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$I_{yy,p}$</td>
<td>$1.13 \times 10^6$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$I_{zz,p}$</td>
<td>$2.23 \times 10^6$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$I_{xx,1}$</td>
<td>$8.77 \times 10^4$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$I_{yy,1}$</td>
<td>$8.77 \times 10^4$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$I_{zz,1}$</td>
<td>$0.114 \times 10^4$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$I_{xx,2}$</td>
<td>$2.1 \times 10^6$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$I_{yy,2}$</td>
<td>$2.1 \times 10^6$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$I_{zz,2}$</td>
<td>$0.01 \times 10^6$</td>
<td>kg-mm²</td>
</tr>
<tr>
<td>$g$</td>
<td>$[0, -9800, 0]^T$</td>
<td>kg-mm²</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>variable</th>
<th>value</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_0$</td>
<td>$[0; 630; 0]$</td>
<td>mm</td>
</tr>
<tr>
<td>$A_r$</td>
<td>50</td>
<td>mm</td>
</tr>
<tr>
<td>$w$</td>
<td>1.5</td>
<td>rad/s</td>
</tr>
<tr>
<td>$\alpha_a$</td>
<td>15</td>
<td>degrees</td>
</tr>
<tr>
<td>$\omega_e$</td>
<td>$[0; 1; 0]$</td>
<td></td>
</tr>
<tr>
<td>$\omega_x$</td>
<td>$[0; 1; 0]$</td>
<td></td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>$\begin{bmatrix} 10I_{3,3} &amp; 0_{3,3} \ 0_{3,3} &amp; 8I_{3,3} \end{bmatrix}$</td>
<td></td>
</tr>
<tr>
<td>$K$</td>
<td>$\begin{bmatrix} 5000I_{3,3} &amp; 0_{3,3} \ 0_{3,3} &amp; 3000I_{3,3} \end{bmatrix}$</td>
<td></td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>$10^4I_{52,52}$</td>
<td></td>
</tr>
<tr>
<td>$\theta_{threshold}$</td>
<td>20</td>
<td>m-Nm</td>
</tr>
</tbody>
</table>

Fig. 2. Result for actuator forces
i.e. end effector and twelve bodies, two per link. As a result, the matrix regressor has a size of six rows by fifty two columns, where the parameter vector is built up by the mass and the principal moments of inertia for each of the thirteen bodies. The simulation results showed a good behavior with errors for position, orientation, linear and angular velocity converging asymptotically to zero as was proved using Lyapunov direct method.

ACKNOWLEDGEMENT

The authors want to think the Departamento Administrativo de Ciencia, Tecnologia e Innovacion, COLCIENCIAS, to support this research throught the financing contract RENATA-563.

REFERENCES


Fig. 3. Result for end effector position and orientation error

Fig. 4. Result for robot linear and angular velocity errors

Fig. 5. Phase plane diagram for translation and rotation.