Alain Codourey*

Institute of Robotics
Swiss Federal Institute of Technology Zurich (ETHZ)
8092 Zurich, Switzerland

Dynamic Modeling of Parallel Robots for Computed-Torque Control Implementation

Abstract

In recent years, increased interest in parallel robots has been observed. Their control with modern theory, such as the computed-torque method, has, however, been restrained, essentially due to the difficulty in establishing a simple dynamic model that can be calculated in real time. In this paper, a simple method based on the virtual work principle is proposed for modeling parallel robots. The mass matrix of the robot, needed for decoupling control strategies, does not explicitly appear in the formulation; however, it can be computed separately, based on kinetic energy considerations. The method is applied to the DELTA parallel robot, leading to a very efficient model that has been implemented in a real-time computed-torque control algorithm.

1. Introduction

Parallel robots possess a number of advantages when compared to serial arms. The most important one is certainly the possibility to keep the motors fixed into the base, thus allowing a large reduction of the robot structure's active mobile mass. Keeping the motors on the robot base is a requirement when direct drive is used; thus, parallel robots are well suited to direct-drive actuation. Another advantage of parallel robots is their high rigidity. These features allow more precise and much faster manipulations. They often, however, suffer from a limited workspace that can be seen as the intersection of the individual workspace of each serial arm constituting the robot. This limitation has partially been resolved by the discovery of the DELTA robot, a new kind of parallel robot dedicated to the handling of light objects (Clavel 1988). To fully use the DELTA robot's potential, a direct-drive version has been developed (Codourey 1991). The advantages of direct-drive robots are well known (Asada and Youcef-Toumi 1987): simple mechanical structure, elimination of backlash, reduction of friction and noise, and higher rigidity. This concept requires greater effort and sophistication at the controller level because of the mechanical coupling and inertial variations that are directly felt on each motor axis. Many control schemes have been proposed in the literature to solve this problem. Model-based methods have given improved results for serial direct-drive robots (Khosla 1986). Implementations of the algorithm on parallel robots have also been reported (Kokkinis and Stoughton 1991), but are scarce. The greatest difficulty in this approach lies in developing a numerically simple dynamic model. If this has been mastered in the case of applications for serial robot structures, the same cannot be said for parallel structures. The few attempts to provide a systematic method for the dynamic modeling of parallel robots led to complicated formulations that are not readily applicable for the control of such robots. The scope of this paper is to present an efficient method for dynamic modeling of parallel robots that can be used in model-based control strategies. After an overview of current methods found in the literature, the proposed method is described. The mass matrix of the robot is first established, based on kinetic energy considerations. It is then shown that for parallel robots, the derivation of Coriolis and centrifugal forces from this mass matrix leads to too much computation. Thus, the dynamic model of the robot is calculated based on the virtual work principle that leads to a simple formulation. The method is finally applied to the DELTA parallel robot. The model thus obtained is efficient and can be computed in real time for control purposes. The proposed control scheme as well as some experimental results are presented.

*This work was partly completed while A. Codourey was with the Institute of Microengineering of the Swiss Federal Institute of Technology Lausanne (EPFL).

The International Journal of Robotics Research
Vol. 17, No. 12, December 1998, pp. 1325-1336,

2. Literature Overview

The development of robot-dynamic models has been a subject of intense research interest over the past decades. The principal difficulty lies in finding a solution that is sufficiently representative of the real system, and that can also be easily calculated in real time for implementation into the control algorithm. For parallel structures, the problem is even more complex than for serial robots, mainly because of the analytical difficulty presented by the joint-variable interactions. Research into methods generally applicable to modeling such robots has been carried out, but their results do not readily lend themselves to real-time processing. A way that has often been explored is to cut out the closed-chain mechanism at passive joints, and first consider the dynamics of the tree robot thus created. The closure condition is then relaxed either by the use of Lagrange multipliers (Kleinfinger 1986) or the application of the d'Alembert virtual work principle by the mean of the Jacobian matrix of the manipulator (Kokkinis and Stoughton 1991; Nakamura 1991; Wang and Chen 1994). For special parallel mechanisms, the direct application of the Newton-Euler method has also been studied (Do and Yang 1988; Codourey 1991; Pierrot, Dauchez, and Fournier 1991; Reboulet and Berthomieu 1991), as well as methods based on the virtual work principle (Clavel 1991; Zhang and Song 1993; Codourey and Burdet 1997), Lagrange formalism (Fujimoto et al. 1991; Miller and Clavel 1992; Lebret, Liu, and Lewis 1993; Pang and Shahinpoor 1994), Hamilton's equations (Miller 1993) or other particular methods (Devaquet 1992, Zanganeh 1997). Almost all agree that a complete model, taking into account the masses and inertia of all the links leads to very complicated solutions that are inefficient for control purposes. To speed up the computation some authors have proposed parallel algorithms (Guglielmetti 1994, Gosselin 1996) which take advantage of the particular geometry of parallel structures. However very often, as shown by Ji (1993), the mass and inertia of the legs can be neglected in regards to the dynamics of the platform and the motors. This simplifying hypothesis leads to simple dynamics which can be used in a real-time model-based control algorithm. This will be used for the modelling of the DELTA parallel robot in Section 4.

3. Dynamic Modeling of Parallel Robots

In this paper, we limit our development to the modeling of a particular class of fully parallel robots. As defined by Pierrot (1991), fully parallel robots have a number of kinematic chains that is equal to the number of degrees of freedom of the robot; furthermore, they possess only one actuator in each chain. Most of the parallel robots in the literature have a maximum of two rigid bodies in each chain, of which one is directly attached to the motor. Examples of such robots are: the Gough platform (Gough 1962) unfortunately known as Stewart platform, the HEXA (Pierrot 1991a), the Hexaglide (Wiegand 1995, Honegger 1997) or the DELTA (Clavel 1988). In this section, we first show how to compute the mass matrix of such robots, and then derive a formulation for their dynamics.

3.1. Mass Matrix Computation

The mass matrix of robots can be obtained based on their kinetic energy (Lewis, Abdallah, and Dawson 1993). The kinetic energy of a rigid body i can be written as

$$T_i = \frac{1}{2}\left[m_i v_i^T v_i + \omega_i^T I_i \omega_i\right], \tag{1}$$

where v_i is the velocity vector of the center of mass of the body, ω_i its angular velocity, m_i its mass, and I_i its inertia matrix. The kinetic energy of the robot is then the sum of the kinetic energies of all bodies:

$$T = \sum_{i=1}^{N} T_i. \tag{2}$$

Let q be the vector of generalized coordinates chosen to describe the robot. The velocities of each body can be expressed as a function of $\dot{q}$ by means of Jacobian matrices:

$$\begin{aligned} v_i &= J_{v,i}\dot{q}, \\ \omega_i &= J_{\omega,i}\dot{q}. \end{aligned} \tag{3}$$

Substituting eqs. (2) and (3) into eq. (1) leads to

$$T = \frac{1}{2}\dot{q}^T\left[\sum_{i=1}^{N}(m_i J_{v,i}^T J_{v,i} + J_{\omega,i}^T I_i J_{\omega,i})\right]\dot{q} \equiv \frac{1}{2}\dot{q}^T A\dot{q}, \tag{4}$$

where A is the mass matrix of the robot, defined as

$$A = \sum_{i=1}^{N}(m_i J_{v,i}^T J_{v,i} + J_{\omega,i}^T I_i J_{\omega,i}). \tag{5}$$

The main difficulty in this formulation is finding the Jacobian matrices for each body. This is particularly true for parallel robots, due to axes interdependencies. However, for fully parallel robots having only one rigid body between the motor and the platform, only the Jacobian of the robot will be needed. In fact, the velocity of each body linking the motor to the platform can be calculated with knowledge of the velocities at both of its extremities, as shown in the following section.

3.2. Kinetic Energy of a Rigid Cylinder

Về Động năng của 1 Trụ đặc

We will assume here that the links between the motors and the platform are cylinders of length L, and that the mass m is uniformly distributed. This is not too restrictive, since this is the case for most parallel robots. If the linear velocities v_1 and v_2 at both ends of the cylinder are known, the linear velocity v_c and the angular velocity ω_c of its center of mass can be calculated if we assume that no rotational movement about the axis of the cylinder itself occurs. We then obtain

assumption

$$v_c = \frac{1}{2}(v_1 + v_2), \tag{6}$$

and

$$\omega_c = \frac{1}{L}e_L \times (v_1 - v_2) \text{ or } \omega_c = \frac{1}{L}\hat{e}_L(v_1 - v_2), \tag{7}$$

where e_L is a unit vector along the bar, and $\hat{e}_L$ is its skew-matrix representation forming the cross-product. If v_1 and v_2 can be calculated from $\dot{q}$ using Jacobians J_1 and J_2, respectively, then both of the velocities v_c and ω_c become

$$v_c = J_{vc}\dot{q} \quad \text{with } J_{vc} = \frac{1}{2}(J_1 + J_2), \tag{8}$$

$$\omega_c = J_{\omega c}\dot{q} \quad \text{with } J_{\omega c} = \frac{1}{L}\hat{e}_L(J_1 - J_2). \tag{9}$$

Then, J_{vc} and $J_{\omega c}$ can be directly used in eq. (5) to compute the mass matrix of the robot. use của J_{vc} và J_{\omega c}

3.3. Kinetic Energy of a Rigid Bar Về Động năng của một thanh

To save computational time, the problem can be further simplified if we consider the link as a bar, that is, that all the mass is concentrated along a unique line. In this case, the kinetic energy of the bar can be calculated considering the integral of the kinetic energy of elementary masses along the bar. As shown in Figure 1, the velocity along the bar can be calculated as

assumption

$$v(x) = \left(1 - \frac{x}{L}\right)v_1 + \frac{x}{L}v_2. \tag{10}$$

The kinetic energy of an elementary mass is

$$dT = \frac{1}{2}v^2 dm = \frac{1}{2}v^2 \rho S dx, \tag{11}$$

where S is the section of the bar, ρ is the density of the material, and dx is an elementary displacement along the bar. Based on this, the kinetic energy of the bar can be calculated as

$$T = \int dT = \frac{1}{2}\rho S \int_0^L v^2 dx,$$

which leads to the result

$$T = \frac{1}{2}\left[\frac{1}{3}m(v_1^2 + v_2^2 + v_1 v_2)\right]. \tag{12}$$

Introducing the Jacobian matrices J_1 and J_2 into this result, we find the contribution of the bar into the mass matrix of the robot:

$$A_{\text{bar}} = \frac{1}{3}m(J_1^T J_1 + J_2^T J_2 + J_1^T J_2). \tag{13}$$

3.4. A Lagrange-Based Dynamic Model

It is well known that the dynamic model of any robot can be written in the following form:

$$\tau = A\ddot{q} + V(q, \dot{q}) + G(q). \tag{14}$$

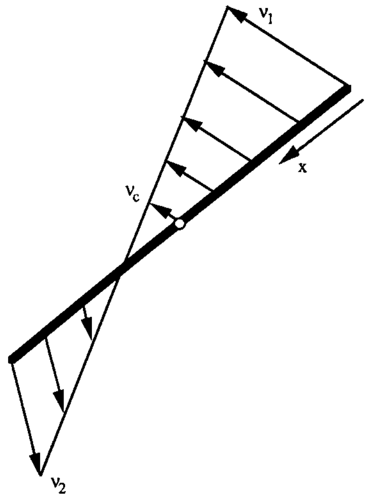

Fig. 1. Velocities along a rigid bar.

It is also known that the Coriolis and centrifugal terms are entirely determined by the mass matrix of the robot. Based on Lagrange's equations of motion and the kinetic energy calculated above, we find (Lewis, Abdallah, and Dawson 1993):

$$V(q, \dot{q}) = \dot{A}(q)\dot{q} - \frac{1}{2}\frac{\partial}{\partial q}(\dot{q}^T A(q)\dot{q}). \tag{15}$$

This formulation is difficult to exploit when modeling parallel robots, because an analytical form of A must be known to efficiently compute the partial derivative. As shown in the example of Section 4, this is not always true for parallel robots, for which A is easier to compute when expressed not only as a function of q but also of the operational vector X. A numerical approach for calculating the partial derivative of eq. (15) could be possible, but would be time consuming. We therefore suggest the computation of the system dynamics with the virtual work principle, which leads to a simple form, without explicitly providing the mass matrix of the robot.

lý do đếch cần đọc 3.4 :)))

3.5. Dynamic Modeling Based on the Virtual Work Principle

The virtual work (or d'Alembert) principle applied to a system of N bodies can be written as follows (Arnold 1989):

Momen quán tính Thành phần Crioslis

$$\sum_{i=1}^{N}[(m_i\ddot{x}_i - F_i)\cdot\delta x_i + (I_i\dot{\omega}_i + \omega_i \times I_i\omega_i - T_i)\cdot\delta\phi_i] = 0, \tag{16}$$

thành phần lực dịch chuyển ảo ở đây là tịnh tiến

thành phần tạo ra momen dịch chuyển ảo ở đây là góc

where:

m_i, I_i are the mass and inertia of body i,

$\ddot{x}_i$ is the acceleration of the center of mass of body i,

$\omega_i, \dot{\omega}_i$ are the angular velocity and acceleration,

F_i, T_i are applied forces (gravity, for example) and moments, and

$\delta x_i, \delta\phi_i$ are virtual displacements.

Expressing the virtual displacements as a function of joint-variable displacements by the mean of the Jacobian matrices $J_{v,i}$ and $J_{\omega,i}$, we get

nguyên hàm Eq.(8) - (9)

$$\sum_{i=1}^{N}\Big[\delta q^T J_{v,i}^T(m_i\ddot{x}_i - F_i) + \delta q^T J_{\omega,i}^T(I_i\dot{\omega}_i + \omega_i \times I_i\omega_i - T_i)\Big] = 0, \quad (17)$$

and finally, because this equation must yield for any δq^T,

$$\sum_{i=1}^{N}\Big[J_{v,i}^T(m_i\ddot{x}_i - F_i) + J_{\omega,i}^T(I_i\dot{\omega}_i + \omega_i \times I_i\omega_i - T_i)\Big] = 0. \quad (18)$$

The applied forces can then be extracted and separated into internal forces τ (actuator forces) and external forces $F_{i,\text{ext}}, T_{i,\text{ext}}$ (gravity and other applied forces). This gives

$$\tau = J_\tau^{-1}\Bigg[\sum_{i=1}^{N}\Big[J_{v,i}^T m_i\ddot{x}_i + J_{\omega,i}^T(I_i\dot{\omega}_i + \omega_i \times I_i\omega_i)\Big] - \sum_{i=1}^{N}\Big[J_{v,i}^T F_{i,\text{ext}} + J_{\omega,i}^T T_{i,\text{ext}}\Big]\Bigg], \quad (19)$$

where J_τ is a special projection matrix allowing us to extract the internal forces (Codourey 1997). To compute this equation, the linear and angular accelerations as well as the angular velocity of each body must be known. They can be calculated by using the time derivatives of the position of each body. In the case of fully parallel robots, the platform position is known from the coordinate transformation. The acceleration of intermediate links can be calculated from the knowledge of the acceleration at both ends of the link using the same procedure as used earlier for the velocities.

3.6. Virtual Work of a Rigid Bar

The virtual work of a rigid bar (without applied forces) can be calculated as

$$\delta W = \int_0^L \delta(x)\cdot a(x)\cdot dm = \rho S\int_0^L \delta(x)\cdot a(x)\cdot dx, \quad (20)$$

where

$$\delta(x) = \left(1 - \frac{x}{L}\right)\delta_1 + \frac{x}{L}\delta_2 \quad (21)$$

is the virtual displacement of a mass element of the bar, given the virtual displacements δ_1 and δ_2 at its extremities, and

$$a(x) = \left(1 - \frac{x}{L}\right)a_1 + \frac{x}{L}a_2 \quad (22)$$

is the acceleration of the mass element with information of the accelerations at both ends of the bar. The resolution of eq. (20) gives

$$\delta W = \frac{1}{3}m\left[a_1\delta_1 + a_2\delta_2 + \frac{1}{2}(a_1\delta_2 + a_2\delta_1)\right].$$

Relating the virtual displacements δ_1 and δ_2 to the joint-variable virtual displacement δ_q, we get

$$\delta W = \delta q^T\cdot\left[\frac{1}{3}m\left[J_1^T\left(a_1 + \frac{1}{2}a_2\right) + J_2^T\left(a_2 + \frac{1}{2}a_1\right)\right]\right], \quad (23)$$

which can be used as the virtual work of body i in eq. (17) to compute the dynamics of the system. This equation requires only the acceleration at both ends of the bar, which are closely related to the accelerations of the platform and the actuators.

4. Application to the DELTA Parallel Robot

In this section, we illustrate the method proposed in Section 3 by an application on the DELTA parallel robot. The robot is first described and its Jacobian matrix is derived. Based on this Jacobian matrix, the mass matrix of the robot is evaluated considering a complete model first. It is then shown that the model can be simplified without substantially affecting its precision. The virtual work principle is finally applied to the DELTA robot, leading to a simple form of its dynamic model.

4.1. The DELTA Robot Description

As illustrated in Figure 2, the DELTA robot is made of three parallel kinematic chains linked at the travelling plate (3). Each chain is moved, driven by a motor (4) fixed to the robot base. Motions of the travelling plate are achieved by the combination of arm movements (1) that are transmitted to the plate by the system of parallel rods (2) through a pair of ball-and-socket passive joints. These parallel rods, also called forearms, ensure that the travelling plate always remains parallel to the robot base.

4.2. Geometric Parameters

The absolute reference frame $\{R\}$ is chosen as shown in Figure 2, at the center of the triangle drawn by the axes of the three motors, with z pointing upward, and x perpendicular to the axis of motor 1. Owing to the robot's triple symmetry, each arm can be treated separately. Its geometric parameters are defined in Figure 3. The index i ($i = 1, 2, 3$) is used to

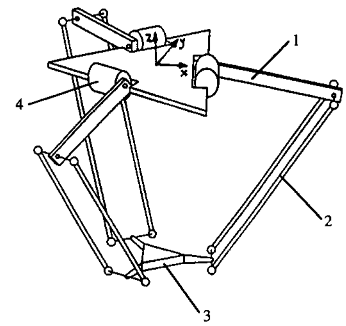

Fig. 2. The DELTA robot. See text for explanation of numbers 1–4.

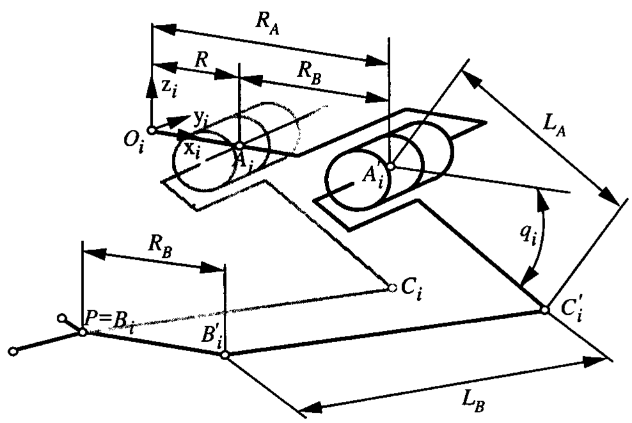

Fig. 3. The geometric parameters of a revolute-actuated DELTA robot.

identify the arm number. Each arm is separated by an angle of 120°. For each arm, a corresponding frame is chosen that is located at the same place as $\{R\}$ but rotated by an angle $\theta_i = 0°, 120°$, and $240°$, for arms 1, 2, and 3, respectively. The transformation matrix between frames $\{R_i\}$ and $\{R\}$ is given by

dùng cho Eq.30

$$ {}^{R}_{i}R = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (24)$$

As the travelling plate can only be translated, a frame attached to it will always keep the same orientation as $\{R\}$. This fact allows us to consider the distance from the reference frame $\{R\}$ to the motor as being $R = R_A - R_B$, and thus $P = B_1 = B_2 = B_3$; that is, the travelling plate is reduced to a single point. This definition will simplify the derivation of the model without affecting the results.

With these definitions, the direct- and inverse-geometric models can be established as proposed by Clavel (1989) or Sternheim (1987). It should be noted, however, that both formulations of the inverse geometric model suffer from mathematical singularities of the type "0/0" that lead to disturbances in the control of the robot at low velocities. To eliminate these singularities, a new model has been proposed (Codourey 1991).

4.3. The Jacobian Matrix of the DELTA Robot

The Jacobian matrix J of a robot expresses the relation between operational-space velocities and joint velocities as follows:

định nghĩa Jacobian matrix

$$\dot{X} = J\dot{q}. \quad (25)$$

For serial robots, a systematic approach can be used to find this matrix. Unfortunately, this is more difficult for parallel robots. Often a loop-closure-constraint equation is used and differentiated to obtain the velocity relationships, i.e., the inverse Jacobian matrix (Waldron, Raghavan, and Roth 1989). The direct Jacobian matrix can then be obtained by numerical inversion of the inverse Jacobian (Merlet 1997). In the case of the DELTA robot, the Jacobian matrix was first established by Codourey (1991). This was based on a numerical computation of the partial derivatives of the direct-geometric model with respect to the joint variables, that is,

cách đầu tiên để tính Jacobian matrix là Eq(26) với hàm X_N = f(q) đã biết.

$$\dot{X}_n = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \frac{\partial f_x}{\partial q_2} & \frac{\partial f_x}{\partial q_3} \\ \frac{\partial f_y}{\partial q_1} & \frac{\partial f_y}{\partial q_2} & \frac{\partial f_y}{\partial q_3} \\ \frac{\partial f_z}{\partial q_1} & \frac{\partial f_z}{\partial q_2} & \frac{\partial f_z}{\partial q_3} \end{bmatrix} \dot{q}, \quad (26)$$

where $X_n = f(q)$ represents the direct-geometric model of the robot. Functions f_x, f_y, and f_z are related to the x-, y-, and z-components of X_n, respectively. The partial derivative is calculated as follows for the first term above:

$$\frac{\partial f_x}{\partial q_1} = \frac{f_x(q_1+\Delta, q_2, q_3) - f_x(q_1, q_2, q_3)}{\Delta}, \quad (27)$$

and so on for the other terms. The quantity Δ must be chosen small enough to reduce the errors in the computation of the Jacobian as much as possible, but large enough to avoid numerical noise. This formulation requires four evaluations of the geometric model for the computation of the Jacobian matrix: $f(q_1, q_2, q_3)$, $f(q_1+\Delta, q_2, q_3)$, $f(q_1, q_2+\Delta, q_3)$, and $f(q_1, q_2, q_3+\Delta)$.

As previously mentioned, another way to compute the Jacobian matrix of parallel robots is to consider a set of constraint equations linking the operational-space variables to the joint-space variables. This method was first applied to the DELTA robot by Guglielmetti (1994). In the following, we describe a simplified version of this latter formulation.

The three constraint equations in the case of the DELTA robot can be chosen as

$$\|C_iB_i\|^2 - L_B^2 = 0 \quad i = 1, 2, 3, \quad (28)$$

signifying that the length of the forearms must be constant. Let s_i be the vector C_iB_i. The previous equation can then be written as

$$s_i^T \cdot s_i - L_B^2 = 0 \quad i = 1, 2, 3, \quad (29)$$

with lưu ý ở dạng vector

$$s_i = O_iB_i - (O_iA_i + A_iC_i)$$
$$= \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} - {}_i^R R \left(\begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} L_A \cos q_i \\ 0 \\ -L_A \sin q_i \end{bmatrix} \right) \quad i = 1, 2, 3. \tag{30}$$

The time derivative of eq. (29) then leads to

$$s_i^T \dot{s}_i + \dot{s}_i^T s_i = 0 \quad i = 1, 2, 3. \tag{31}$$

Owing to the commutativity property of the product, this can be rewritten as

$$s_i^T \dot{s}_i = 0 \quad i = 1, 2, 3, \tag{32}$$

where the time derivative of s_i is given by

$$\dot{s}_i = \begin{bmatrix} \dot{x}_n \\ \dot{y}_n \\ \dot{z}_n \end{bmatrix} + {}_i^R R \begin{bmatrix} L_A \sin q_i \\ 0 \\ -L_A \cos q_i \end{bmatrix} \dot{q}_i = \dot{X}_n + b_i \dot{q}_i \tag{33}$$
$$i = 1, 2, 3,$$

where

$$b_i = {}_i^R R \begin{bmatrix} L_A \sin q_i \\ 0 \\ -L_A \cos q_i \end{bmatrix} \quad i = 1, 2, 3. \tag{34}$$

Using the definitions of eqs. (30) and (34) and rearranging eq. (33) into vector form, the following is obtained:

$$\begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix} \dot{X}_n + \begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix} \dot{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{35}$$

where $\dot{q} = [\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3]^T$ is the joint-space velocity vector. From this last equation, the Jacobian matrix of the robot is obtained,

$$\dot{X}_n = J\dot{q},$$

with Jacobian matrix dùng từ Eq.(30) & (24)

$$J = - \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix}^{-1} \begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix}. \tag{36}$$

It is worth noting here that the Jacobian matrix J is not only a function of q, as is usually the case for serial robots, but also a function of the end-effector position X_n, evaluated using the direct-geometric model of the robot.

4.4. The Acceleration Relation

After derivation of eq. (35) and some transformations, we find

$$\ddot{X}_n = - \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix}^{-1} \left(\begin{bmatrix} \dot{s}_1^T \\ \dot{s}_2^T \\ \dot{s}_3^T \end{bmatrix} J + \begin{bmatrix} \dot{s}_1^T b_1 + s_1^T \dot{b}_1 & 0 & 0 \\ 0 & \dot{s}_2^T b_2 + s_2^T \dot{b}_2 & 0 \\ 0 & 0 & \dot{s}_3^T b_3 + s_3^T \dot{b}_3 \end{bmatrix} \right) \cdot \dot{q} + J\ddot{q}, \tag{37}$$

where $\dot{s}_i$ has been calculated as above, and $\dot{b}_i$ is given by

$$\dot{b}_i = {}_i^R R \begin{bmatrix} L_A \cos q_i \\ 0 \\ -L_A \sin q_i \end{bmatrix} \dot{q}_i \quad i = 1, 2, 3. \tag{38}$$

In eq. (37), the time derivative of the Jacobian $\dot{J}$ can be identified as the term multiplying $\dot{q}$.

4.5. Mass Matrix of the DELTA Robot

The determination of the robot's mass matrix is important when the objective is to decouple the individual axes in the overall robot control. Owing to high coupling between the axes, this is particularly true for parallel robots. As shown in Section 3.1, the mass matrix can be computed based on kinetic-energy considerations. The DELTA robot is built of seven bodies: the travelling plate, three forearms, and three arms. According to eq. (5), the mass matrix of the robot is the sum of the contributions of each body:

$$A = A_n + A_{\text{forearms}} + A_{\text{arms}}. \tag{39}$$

By direct application of eq. (5) to the travelling plate, we obtain

$$A_n = m_n J^T J, \tag{40}$$

where J is the Jacobian matrix of the DELTA (eq. (36)) and m_n is the mass of the travelling plate and the payload. The contribution of the arms can be compacted into one matrix, as follows:

$$A_{\text{arms}} = I_b = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix}, \tag{41}$$

where $I_{b1} = I_{b2} = I_{b3} = I_{bi}$, and

$$I_{bi} = I_m + L_A^2 \left(\frac{m_b}{3} + m_c \right),$$

where I_m is the inertia of the motor, m_b is the mass of the arm, and m_c is the mass of the elbow.

Finally, the contribution of the forearms can be computed, based on eq. (13). For this, the Jacobians relating the velocities at both ends of the bar to the joint-velocity vector must be known. The velocity at the lower end of the bar is actually the velocity of the travelling plate $\dot{X}_n$, related to the joint-velocity vector by the mean of the Jacobian, J. At the upper extremity of the bar, the velocity can be calculated as follows:

$$v_{u,i} = -{}_i^R R \begin{bmatrix} L_A \sin q_i \\ 0 \\ L_A \cos q_i \end{bmatrix} \dot{q}_i . \tag{42}$$

Written as a function of the joint-velocity vector $\dot{q}$, for each of the forearms we get

$$v_{u,1} = \left[-{}_1^R R \begin{bmatrix} L_A \sin q_1 \\ 0 \\ L_A \cos q_1 \end{bmatrix} \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} \right] \dot{q} = J_{u,1}\dot{q}, \tag{43}$$

$$v_{u,2} = \left[\begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \; -{}_2^R R \begin{bmatrix} L_A \sin q_2 \\ 0 \\ L_A \cos q_2 \end{bmatrix} \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \right] \dot{q} = J_{u,2}\dot{q}, \tag{44}$$

$$v_{u,3} = \left[\begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} \; -{}_3^R R \begin{bmatrix} L_A \sin q_3 \\ 0 \\ L_A \cos q_3 \end{bmatrix} \right] \dot{q} = J_{u,3}\dot{q}, \tag{45}$$

where we identify the Jacobians as $J_{u,1}$, $J_{u,2}$, $J_{u,3}$.

Thus, the contribution of each forearm to the mass matrix of the robot can be calculated as (eq. (13))

thực ra, việc symplify forearm về bar không đúng với TH của mình, custom lại cho đúng

$$A_{b,i} = \frac{1}{3} m_{ab} (J^T J + J_{u,i}^T J_{u,i} + J_{u,i}^T J), \tag{46}$$

where m_{ab} is the mass of the forearm.

Finally, the mass matrix of the DELTA robot is expressed as

$$A = I_b + m_n J^T J + \sum_{i=1}^{3} \frac{1}{3} m_{ab} (J^T J + J_{u,i}^T J_{u,i} + J_{u,i}^T J). \tag{47}$$

A careful analysis of this equation shows an interesting property that can lead to further simplifications. From the first term within the sum, we see that for each forearm, a third of its mass can simply be added to the contribution of the travelling plate, resulting in a total mass of $m_{nt} = m_n = m_{ab}$. The second term leads to

$$\sum_{i=1}^{3} \frac{1}{3} m_{ab} J_{u,i}^T J_{u,i} = \frac{1}{3} m_{ab} \begin{bmatrix} L_A^2 & 0 & 0 \\ 0 & L_A^2 & 0 \\ 0 & 0 & L_A^2 \end{bmatrix}, \tag{48}$$

which can be seen as the addition of a third of the mass of each forearm at the extremity of the respective arm, and can directly be added to I_b. The third term is a combination of both of the

Những ô dưới đây đã có rất nhiều simplification made, cần thận khi apply vào mô hình thực tế và nếu để phù hợp cho bài báo sáu này, cần lược bỏ các simplificaation này

Jacobians. Each arm i introduces only a contribution to the same arm i, but is subject to coupling; for example, for arm 1 it has the following form:

$$A_{\text{third},1} = \frac{1}{3} m_{ab} \begin{bmatrix} t_1 & t_2 & t_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{49}$$

where t_1, t_2, and t_3 are functions of the robot's position. For further simplification, we could neglect the coupling and place the resting third of the mass at the extremity of the arm. This means that eventually a third of the mass of each forearm would be added to the travelling plate, and two thirds to the extremity of the arm. This is the simplification that has been adopted for the first dynamic model of the DELTA robot (Codourey 1991). Based on these simplifications, the mass matrix of the robot can finally be written as

$$A = I_{bt} + m_{nt} J^T J, \tag{50}$$

where $I_{bt} = \begin{bmatrix} I_{bt1} & 0 & 0 \\ 0 & I_{bt2} & 0 \\ 0 & 0 & I_{bt3} \end{bmatrix}$

with $I_{bti} = I_m + L_A^2 \left(\frac{m_b}{3} + m_c + \frac{2}{3} m_{ab} \right)$.

The shape of this mass matrix is shown in Figures 4 and 5. The 9 graphs each represent a term of the 3×3 mass matrix for a horizontal cut into the workspace of the robot at the height z = −230 mm and z = −350 mm, respectively. We first recognize the 3-symmetry of the robot by analyzing the diagonal terms of the matrix. The ratio between the maximum and minimum inertias is about 2. Comparing the diagonal terms of Figures 4 and 5, we can see that more variation in the inertias occurs when the robot is at the upper part of its workspace. The coupling between the axes is very important and has the same order of magnitude as the actual inertia of each individual axis. This shows the importance of incorporating a decoupling strategy into the control of the robot. Equation (50) is a simple form that can readily be used for that purpose, as will be discussed in Section 5. The Jacobian can be evaluated either by numerical derivation of the geometric model or by using the formulation developed in Section 4.3.

4.6. Virtual Work-Based Dynamic Model of the DELTA Robot

Based on eq. (18), for the DELTA robot we can write

$$\tau_n + \sum_{i=1}^{3} \tau_{b,i} + \sum_{i=1}^{3} \tau_{ab,i} = 0, \tag{51}$$

where τ_n is the contribution of the forces acting on the travelling plate, $\tau_{b,i}$ is the force/torque contribution of arm i, and $\tau_{ab,i}$ is the force/torque contribution of forearm i. Because the travelling plate always remains parallel to the base, there

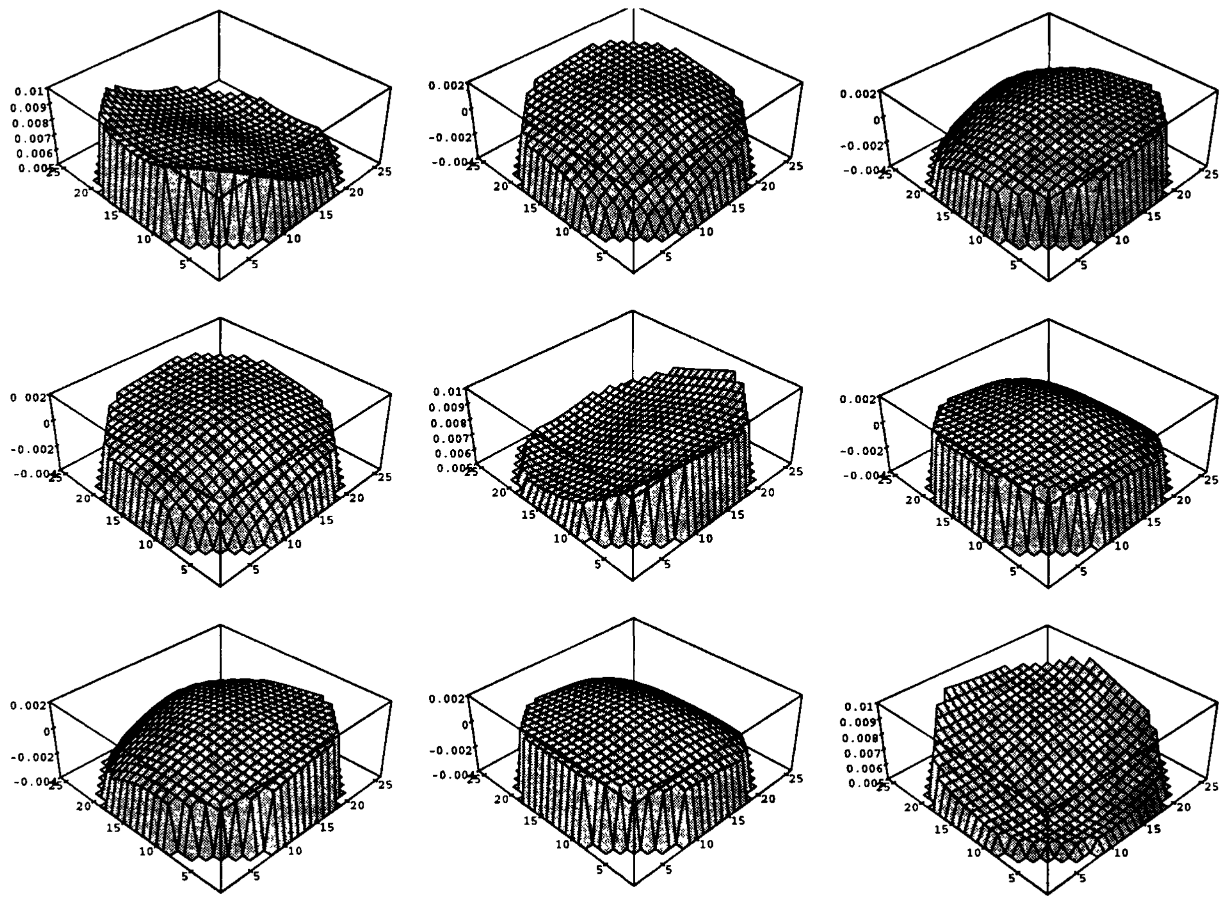

Fig. 4. The mass matrix of the DELTA robot for a horizontal cut into its workspace at height $z = -230$ mm. The upward axis represents the inertia seen from the motor (units are kg·m^2). The scales of the other two axes represent the sample numbers in the xy-plane of the robot, with 0 corresponding to –250 mm, and 25 to +250 mm.

is no contribution from the rotational terms in τ_n. We get

$$\tau_n = J^T(m_n \ddot{X}_n G_n) \tag{52}$$

with $G_n = m_n \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$ and g the gravitational acceleration.

By grouping the contributions of the three arms into vector form and taking into account that their movement is composed only of rotations, we get

$$\tau_b = I_b \ddot{q} - G_b - \tau, \tag{53}$$

where τ is the torque applied by each motor, and

$$G_b = m_{br} r_{Gb} g \begin{bmatrix} \cos q_i & \cos q_2 & \cos q_3 \end{bmatrix}^T \tag{54}$$

is the contribution of gravity, where

$$m_{br} = m_b + m_c \text{ and } r_{Gb} = L_A \frac{\frac{1}{2} m_b + m_c}{m_b + m_c}.$$

To compute the contribution of each forearm, eq. (23) can be used, which leads to

$$\tau_{ab,i} = \frac{1}{3} m_{ab} \left[J^T \left(\ddot{X}_n + \frac{1}{2} a_{u,i} \right) + J_{u,i}^T \left(a_{u,i} + \frac{1}{2} \ddot{X}_n \right) \right] - \frac{1}{2} (J + J_{u,i})^T m_{ab} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \tag{55}$$

where gravity has been added and where the acceleration of the upper part of the forearm can be computed by the time derivative of eq. (34):

$$a_{u,i} = -{}_i^R R \left[\begin{bmatrix} L_A \sin q_i \\ 0 \\ L_A \cos q_i \end{bmatrix} \ddot{q}_i + \begin{bmatrix} L_A \cos q_i \\ 0 \\ -L_A \sin q_i \end{bmatrix} \dot{q}_i^2 \right]. \tag{56}$$

This leads to the following equation for the dynamics of the DELTA robot:

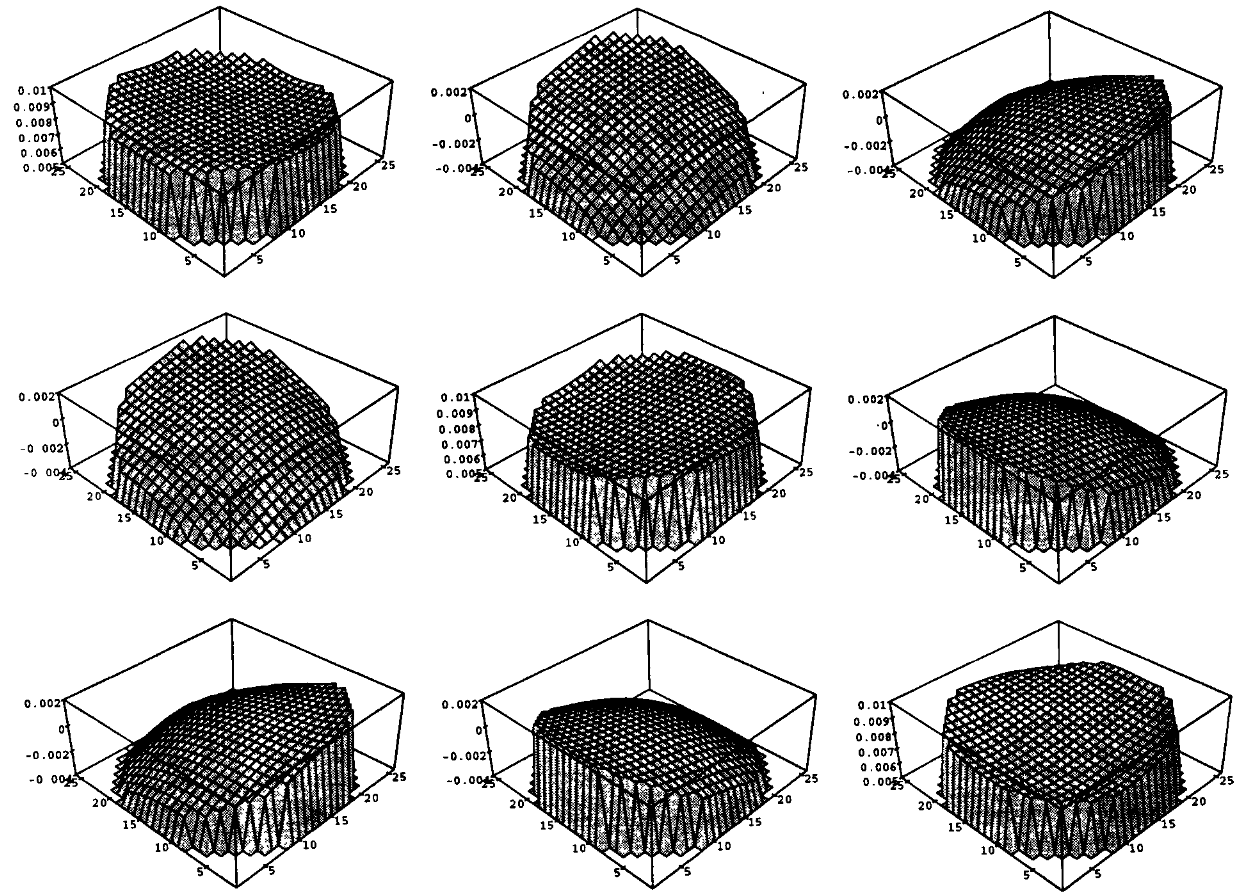

Fig. 5. The mass matrix of the DELTA robot for a horizontal cut into its workspace at height $z = -350$ mm. The upward axis represents the inertia seen from the motor (units are kg·m^2). The scales of the other two axes represent the sample numbers in the xy-plane of the robot, with 0 corresponding to –250 mm, and 25 to +250 mm.

$$\tau = I_b\ddot{q} + J^T\ddot{X}_n - G_b - J^T G_n + \sum_{i=1}^{3} \tau_{ab,i}. \tag{57}$$

As was the case in Section 4.5., an analysis of $\tau_{ab,i}$ shows that a further simplification can be made if we choose to neglect some coupling effects due to the forearms. For inertial contributions, the same mass distribution as before can be chosen, that is, one third of the mass on the travelling plate, and two thirds of the mass at the elbow. The gravitational force of each forearm can be computed without any loss by placing one half of the mass at each of its extremities. The dynamic model of the DELTA robot is thus reduced to

$$\tau = I_{bt}\ddot{q} + J^T m_{nt}\ddot{X}_n - J^T m_{ng}\begin{bmatrix}0 & 0 - g\end{bmatrix}^T - G_{bg}, \tag{58}$$

where $m_{ng} = m_n + \frac{3}{2}m_{ab}$ and

$$G_{bg} = L_A\left(\frac{1}{2}m_b + m_c + \frac{1}{2}m_{ab}\right) g\begin{bmatrix}\cos q_1 & \cos q_2 & \cos q_3\end{bmatrix}^T. \tag{59}$$

In eq. (58), $\ddot{X}_n$ is calculated by numerically differentiating the direct-coordinate transformation twice with respect to time:

$$\ddot{X}_n = \frac{d^2 f(q)}{dt^2}.$$

In a computational sense, this is more efficient than calculating $\dot{J}\dot{q}$ (see eq. (37)), but can lead to numerical noise, especially when the accelerations of the robot are small. Since the DELTA robot is usually driven with very high accelerations, this is not too troublesome; still, it is a point to be retained.

5. The Control Scheme

5.1. Control Architecture

The scheme used for the control of the direct-drive DELTA robot is shown in Figure 6. It is composed of a feedforward block in which the inverse-dynamic model of the robot is calculated based on eq. (58). This block has as inputs both the joint-space (q) and operational-space (X_n) position and acceleration signals. The accelerations are obtained by the double numerical differentiation of the desired motor angles and travelling-plate position. In this controller configuration, the trajectory signal input to the robot regulator is phase shifted with respect to the feedforward torque-generation block. This trick allows us to take account of an interval of $T_m = 2$ msec, which corresponds to the rise time of the current in the motor. The controller is a standard proportional-derivative (PD) regulator. The inertia matrix, calculated with the formulation of eq. (50), is placed in series with the regulator. This allows for the elimination of the interaction or coupling between the three robot axes and their linearization.

This control strategy has been implemented successfully on a network of three transputers, and is sufficient for the implementation of all the desired control-algorithm elements. Three sections include: Cartesian-space trajectory generation, geometric model evaluation, and dynamic model and regulation processing (Codourey, Clavel, and Burckhardt 1991). Table 1 summarizes the time needed for the computation of each of these processes on T800 transputers. The sampling interval was fixed at 1 msec. The OCCAM programming language was used with the TDS2 development system.

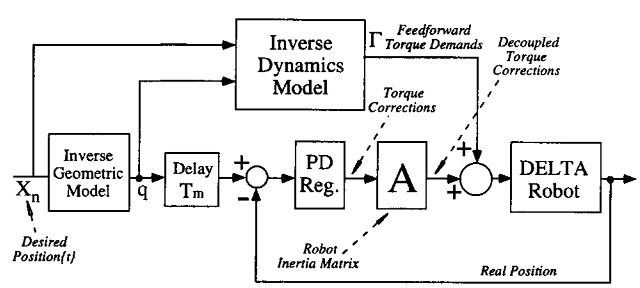

Fig. 6. Feedforward control of the direct-drive DELTA robot with mass-matrix-based, regulator-output-command decoupling.

Table 1. Computation Time on a T800 Transputer

	Time (in μsec)
Trajectory	149
Coordinate transformation	244
Dynamics	704
Mass matrix	233

5.2. Experimental Results

To be able to evaluate the performance of the control algorithm proposed above, it may be instructive to compare it with that of classical regulators, such as the very classical PD controller. Four different types of regulators were tested:

1. an independant PD controller on each axis,
2. a PD controller with an acceleration feedforward component,
3. a PD controller with a full dynamic-model feedforward term with $T_m = 0$, and same as point 3 above, but with $T_m = 2$ msec, all without the use of the mass matrix.

Several trajectories and movement profiles that permit the displacement of the travelling plate in the Cartesian space have been developed and implemented (Codourey 1991). The results presented here were obtained for the case of a semi-ellipse trajectory with a sine-on-ramp profile. Figure 7 shows the tracking errors for the different regulators described above. It should be observed, from the figure, that the more complete the model was made, the better was the trajectory tracking. Overall improved performance was obtained in passing from regulator 1 to regulator 4. The maximal tracking error was found to be reduced by a factor of 6. This corroborates the necessity of introducing accurate dynamical models in the control algorithms of high-speed robots. Other experiments have shown that the use of the mass matrix in the control loop leads to equivalent results; that is, the error is not further decreased. This confirms the result found by An, Atkeson, and Hollerbach (1988) for serial manipulators. However, the mass matrix allows the linearization of the system, which in some cases may be required.

The choices of suitable trajectory and movement profiles also play a capital role in the determination of tracking performances, and therefore in the speed of convergence and the overshoot at the end of the trajectory. Figure 8 shows the arrival toward the endpoint for three different types of velocity profiles on a semi-ellipse. This shows that the choice of a smooth and nonsymmetrical profile (with acceleration higher than deceleration) is very beneficial to the reduction of overshoots at the end of the trajectory. With this proposed control system and trajectories, pick-and-place movements between two points distant by 250 mm have been realized with a working rhythm superior to 3 Hz.

6. Conclusions

In this article, a simple method for the dynamic modeling of parallel robots was presented. The model developed is simple enough to be computed in real time and used in a computed-torque control scheme. The mass matrix of the robot that is needed in such strategies is evaluated based on kinetic energy considerations. The robot's dynamics is then

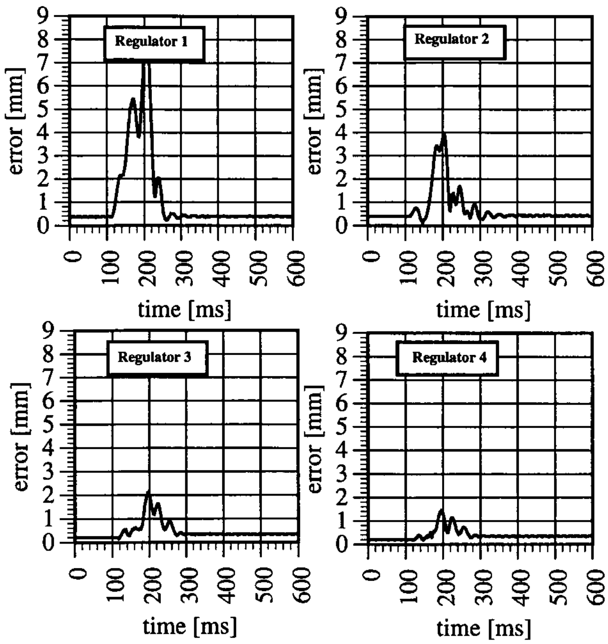

Fig. 7. Tracking errors for the different regulators tested.

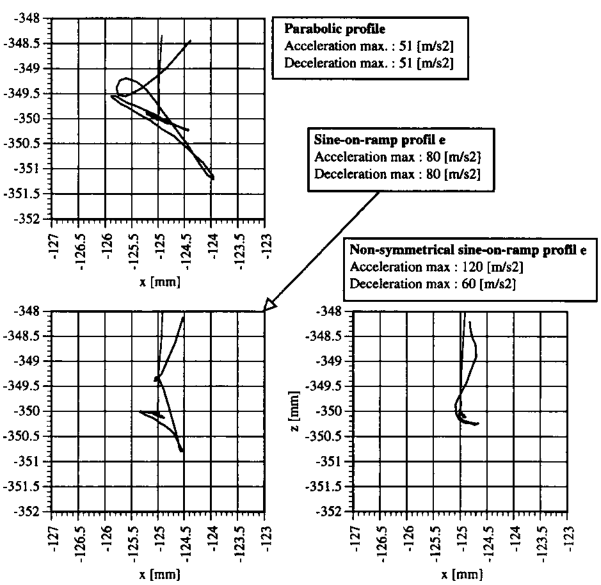

Fig. 8. Arrival toward the endpoint for different types of profiles. Parabolic profile (top) with acceleration and deceleration maximums of 51 m/sec^2; sine-on ramp profile (bottom left) with acceleration and deceleration maximums of 80 m/sec^2; and nonsymmetrical sine-on ramp profile (bottom right) with acceleration maximum of 120 m/sec^2, and deceleration maximum of 60 m/sec^2.

evaluated, based on the virtual work principle that projects forces acting on each body onto the joint space. This method was applied to the DELTA parallel robot. It was shown that a good choice of a simplifying hypothesis can lead to a very efficient computation of the dynamics without much loss in precision. A control scheme was finally proposed that has been successfully implemented on the DELTA robot, allowing fast pick-and-place movements at a speed of up to 3 Hz. Thus, today the DELTA robot is one of the fastest robots in the world.

Acknowledgments

The author would like to thank Etienne Burdet and Laurent Rey for their help and comments while preparing this paper, and is grateful to one anonymous referee for valuable comments that helped to improve the paper.

References

An, C. H., Atkeson, C. G., Hollerbach, J. M. 1988. *Model-Based Control of a Robot Manipulator.* Cambridge, MA: MIT Press.

Arnold, V. I. 1989. *Mathematical Methods of Classical Mechanics.* Springer-Verlag.

Asada, H., and Youcef-Toumi, K. 1987. *Direct-Drive Robots, Theory and Practice.* Cambridge, MA: MIT Press.

Clavel, R. 1988 (April 26–28, Lausanne). Delta, a fast robot with parallel geometry. *Proc. of the 18th Int. Symp. on Industrial Robots (ISIR).* Berlin: Springer-Verlag, pp. 91–100.

Clavel, R. 1989. Une nouvelle structure de manipulateur parallèle pour la robotique légère. *Automatique, Productique, Informatique Industrielle* 23:501–519.

Clavel, R. 1991. Conception d'un robot parallèle rapide à 4 degrés de liberté. Ph.D. thesis, Swiss Federal Institute of Technology Lausanne (EPFL), 925 (in French).

Codourey, A. 1991. Contribution à la commande des robots rapides et précis. Application au robot DELTA à entraînement direct. Ph.D. thesis, Swiss Federal Institute of Technology Lausanne (EPFL), 922 (in French).

Codourey, A., and Burdet, E. 1997 (April 20–25, Albuquerque, NM). A body-oriented method for finding a linear form of the dynamic equation of fully parallel robots. *Proc. of the IEEE Int. Conf. on Robot. and Automat.*, vol. 2. Washington, DC: IEEE, pp. 1612–1618.

Codourey, A., Clavel, R., and Burckhardt, C. W. 1991 (September, Vienna). Control algorithm and controller for the direct drive DELTA robot. *Proc. of the IFAC Symp. on Robot Control*, pp. 169–177.

Devaquet, G., and Brauchli, H. 1992. A simple mechanical model for the DELTA robot. *Robotersysteme* 8:193–199.

Do, W. Q. D., and Yang, D. C. H. 1988. Inverse dynamic analysis and simulation of a platform type of robot. *J. Robot. Sys.* 5(3):209–227.

Fujimoto, K., et al. 1991 (November, Osaka). Derivation and analysis of equations of motion for 6-DOF direct-drive wrist joint. *Proc. of the Int. Workship on Intell. Robots and Sys., IROS'91,* vol. 2, pp. 779–784.

Gosselin, C. M. 1996 (March). Parallel computational algorithms for the kinematics and dynamics of planar and spatial parallel manipulators. *SAME J. Dyn. Sys. Meas. Control* 118:22–28.

Gough, V. E., and Whithall, S. G. 1962. Universal tyre test machine. *Proc. of the 9th Int. Technical Congress, FISITA*, vol. 117, pp. 117-135.

Guglielmetti, P. 1994. Model-based control of fast parallel robots: A global approach in operational space. Ph.D. thesis, Swiss Federal Institute of Technology Lausanne (EPFL), 1228.

Honegger, M., Codourey, A., and Burdet, E. 1997 (April, Albuquerque, NM). Adaptive control of the Hexaglide, a 6-DOF parallel manipulator. *Proc. of the IEEE Conf. on Robot. and Automat.*, vol. 1. Washington, DC: IEEE, pp. 543–548.

Ji, A. 1993 (Atlanta, GA). Study of the effect of leg inertia in Stewart platforms. *Proc. of the IEEE Conf. on Robot. and Automat.* Los Alamitos, CA: IEEE, pp. 121–126.

Khosla, P. 1986. Real-time control and identification of direct-drive manipulators. Ph.D. thesis, Carnegie-Mellon University, Pittsburgh, PA.

Kleinfinger, J.-F. 1986. Modélisation dynamique des robots à chaîne cinématique simple, arborescente ou fermée, en vue de leur commande. Ph.D. thesis, Ecole Nationale Supérieure de Mécanique, Nantes.

Kokkinis, T., and Stoughton, R. 1991. Dynamics and control of closed-chain robot arms with application to a new direct-drive robot arm. *Int. J. Robot. Automat.* 6(1):25–34.

Lebret, G., Liu, K., and Lewis, F. L. 1993. Dynamic analysis and control of a Stewart platform manipulator. *J. Robot. Sys.* 10(5):629–655.

Lewis, F. L., Abdallah, C. T., and Dawson, D. M. 1993. *Control of Robot Manipulators.* New York: Macmillan.

Merlet, J.-P. 1997. *Les robots parallès*, 2e èdition revue et augmentèe. Paris: Editions Hermes.

Miller, K. 1993 (November, Tokyo). The proposal of a new model of direct-drive robot DELTA-4 dynamics. *Proc. of '93 ICAR, the Int. Conf. on Adv. Robot.*, pp. 411–416.

Miller, K., and Clavel, R. 1992. The Lagrange-based model of DELTA-4 robot dynamics. *Robotersysteme* 8(1):49–54.

Nakamura, Y. 1991. *Advanced Robotics: Redundancy and Optimization.* Reading, MA: Addison-Wesley.

Pang, H., and Shahinpoor, M. 1994. Inverse dynamics of a parallel manipulator. *J. Robot. Sys.* 11(8):693–702.

Pierrot, F. 1991. Robots pleinement parallèles légers: Conception, modèlisation et commande. Ph.D. thesis, Université de Montpellier II, Montpellier, France.

Pierrot, F., Dauchez, P., and Fournier, A. 1991 (June, Pisa). HEXA: A fast six-DOF fully-parallel robot. *Proc. of '91 ICAR, the Int. Conf. on Adv. Robot.*

Reboulet, C., and Berthomieu, T. 1991 (June, Pisa). Dynamic models of a six-degree-of-freedom parallel manipulator. *Proc. of '91 ICAR, the Int. Conf. on Adv. Robot.*, vol. 2, pp. 1153–1157.

Sternheim, F. 1987. Computation of the direct and inverse geometric models of the DELTA 4 parallel robot. *Robotersysteme* 3:199–203.

Waldron, K. J., Raghavan, M., and Roth, B. 1989 (June). Kinematics of a hybrid series-parallel manipulation system. *J. Dyn. Sys. Meas. Control* 111:211–221.

Wang, L.-C. T., and Chen, C. C. 1994. On the dynamic analysis of general parallel robotic manipulators. *Int. J. Robot. Automat.* 9(2):81–87.

Wiegand, A., and Weikert, S. 1995. Vorrichtung zur räumlichen gesteuerten Bewegung eines Körpers in drei bis sechs Freiheitsgraden. CH Patentgesuch 3600/95, December 20, 1995.

Zanganeh, K. E., Sinatra, R., and Angeles, J. 1997. Kinematics and dynamics of a six-degrees-of-freedom parallel manipulator with revolute legs. *Robotica* 15:385–394.

Zhang, C.-D., and Song, S.-M. 1993. An efficient method for inverse dynamics of manipulators based on the virtual work principle. *J. Robot. Sys.* 10(5):605–627.