Abstract

This paper presents a general design study of an adaptive force/position control using the (SFB) "strict-feedback backstepping" technique, based on passivity and applied to a robotic system. The advantage of the implemented control algorithm is that it imposes desired stability properties by fixing the storage, output stabilizing and Lyapunov candidate functions of the system. The parameters estimation for the design is made by the direct adaptive technique. The control law which is valid for various types of robotic architectures is very satisfactory when applied to a 4 d.o.f robot, consisting of one prismatic axis (axis 1), and three others rotary axes. The results obtained are satisfactory since the robot follows exactly the desired position and force trajectory. The trajectory and force tracking errors are negligible. The global stability of the system is also ensured.

1-Introduction

Numerous robot tasks (e.g. contour following, grinding, deburring and assembly-related tasks) generate physical contact between the robot end-effector and the environment. In such cases, the force due to the contact with the surface has to be taken into consideration and therefore a hybrid force/motion control is required. Two goals are related to this problem. The first is to maintain a certain force magnitude, applied by the end-effector on the workspace; the second is to maintain the end-effector’s motion control on a desired trajectory. A major advantage of this backstepping method is its flexibility to build the control law by avoiding the cancellation of useful non-linearities.

Khatib [1987] developed a framework for analysis and control of manipulator systems with respect to dynamic behavior of their end-effectors. He presented the fundamentals of the operational space formulation and the unified approach for motion and force control ensuring a decoupling for adaptive force/motion control with an extension for the redundant robots. These results are used in the development of a new and systematic approach for dealing with the problems arising from kinematics singularities. Slotine and Li [1987] showed the use of state feedback to directly modify a manipulator’s energy function for the force/position control. This approach, which has a good performance, ensures a general decoupling of the system into two dynamics related to the task space, one for the position and the other for the force. The global convergence is also shown by the authors. Minf et al. [1996] dealt with the case of a force/motion control of a manipulator robot subjected to holonomic constraints based on the sliding mode. This approach is similar to that of McClamroch and Wang [1988] and consists of introducing the constraint equation into the articulator model in order to obtain the suitable control law. The algorithm guarantees asymptotic stability of the global system and the convergence of the errors in force/position.

The approach presented in this paper consists of designing an adaptive backstepping control law for force/position tracking problems in robotic systems. This design that is made for each recursive step includes the study of the passive system. The theory of decoupling is similar to that of Slotine and Li [1987]. We show the stability of the system by setting up an adaptation law obtained by selecting a suitable Lyapunov candidate function adapted to each design.

This paper is organized as follows. In section 2, the model for the robot manipulator is constructed by considering the rigid robot and actuator models as interconnected subsystems. The dynamic equations of the robot are derived in Cartesian space. Section 3 presents the force/position decoupling logic. The representation of the various matrices is made in the task space. In section 4, the robot dynamic model is established using an SFB technique in the task space. This model is used in the derivation of the backstepping adaptive control law applied to the trajectory tracking problem of section 5. The parameters of the system are presented in section 6 and numerical simulation results are shown in sections 7 and 8. Finally, a conclusion is given in section 9.

2- Dynamic Equations of the Robot

Consider an n-links manipulator. Let \( q \in \mathbb{R}^n \) denote the vector of generalized displacements in articulation coordinates. If \( q \in \mathbb{R}^n \) is differentiable, then we denote by \( \dot{q} \) its time derivative. \( q \in \mathbb{R}^n \) is viewed as a column vector so that its transpose \( q^T \) is a row vector. \( M(q) \in \mathbb{R}^{n \times n} \) is a symmetric, positive definite inertia and mass matrix. \( C(q) \in \mathbb{R}^{n \times n} \) is a matrix of the coriolis and centrifugal forces. \( G(q) \in \mathbb{R}^n \) is the vector of gravity terms.

Then, the manipulator equation of motion in joint space is given by:

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau + f
\]  

(1)

The input torque to the actuators produced by the actuators are given by the following relation:

\[
\tau = K_1 \nu
\]  

(2)

where \( \tau \) and \( f \) are the joint input torque and force vectors. Relation (1) lets one express an important property of the dynamics, namely, that \( M(q) - 2C(q, \dot{q}) \) is skew-symmetric. According to Su and Stepanenko [1997], we have a new form of the dynamic equation, by substituting the expression (2) of the torque in the general equation (1) and by adding a second expression representing the equation of the actuator:

\[
\begin{align*}
M(q) & \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = K_1 \nu + f \\
L \dot{\nu} + R \nu & = v
\end{align*}
\]  

(3)

\( L, R, K_\alpha \) and \( K_\nu \in \mathbb{R}^{n \times n} \) are positive definite diagonal matrices which represent, respectively, the actuator inductance, the actuator resistance, the constant coefficient of the actuator and the constant coefficient characterizing electromechanical conversion between currents and torques. \( v \) \in \mathbb{R}^n \) is the vector armature currents, and \( v \) \in \mathbb{R}^n \) is the vector armature voltages.

The dynamic system in Cartesian space can be expressed as:

\[
M_s(q) \ddot{x} + C_s(q, \dot{x}) \dot{x} + G_s(q) = F
\]  

(4)

where \( F_s \) \in \mathbb{R}^n \) is the vector of the generalized forces.

The matrices \( M_s, C_s \) and \( G_s \) are related to the matrices \( M(q), C(q, \dot{q}) \) by the following relations:

\[
\begin{align*}
M_s(q) & = J^T(q) M(q) J(q) \\
C_s(q) & = J^T(q) C(q, \dot{q}) J(q) - M_s(q) J(q) J^T(q) \\
G_s(q) & = J^T(q) G(q)
\end{align*}
\]  

(5)

where \( J(q) \) is the Jacobian matrix.
where \( J \) is the Jacobean matrix and \( J^T \) is its transpose inverse.

### 3. Decoupling

In the present study, decoupling is applied to determine the part of the system dynamics which must control the position, and the other part that must control the force. The position and force dynamics are decoupled in the workspace. The decoupling is accomplished using the transformation matrix \( R_0 \) given by Khatib [1987].

#### 3.1 Transformation Matrix

The execution of a task by an uncoupled system requires a structure which can be easily managed. This structure, which is the matrix of transformation \( R_0 \) (Khatib, 1987), is made up of two sub-matrices of the same rows. The first sub-matrix \( R_0 \) is related to the manipulation of the forces associated with the positions, while the second sub-matrix \( R_2 \) concerns the moments associated with the orientations. According to the tasks, \( R_1 \) and \( R_2 \) can be fixed or time varying. In fact, we imposed a constraint in the \( z \)-direction speed, and perpendicular to the contact surface. The matrix of transformation thus takes the following form:

\[
R_0 = \begin{bmatrix} R_0 & 0 \\ 0 & R_2 \end{bmatrix}
\]

where \( R_0 \) is the transformation matrix and \( R_1 \) and \( R_2 \) are the rotations matrices.

#### 3.2 Decoupling Organization

The vector of linear and angular velocities in the tool frame is given by:

\[
\dot{X}_e = \begin{bmatrix} v_e \\ \omega_e \end{bmatrix}
\]

This velocity and these accelerations are related to the reference frame as follows:

\[
\begin{align*}
\dot{x} &= R_0 \dot{x}_e \\
\dot{\omega} &= R_0 \dot{\omega}_e + \dot{R}_0 \dot{x}_e
\end{align*}
\]

Substituting (8) into (4), we obtain:

\[
\begin{align*}
M_{ce} \ddot{x}_e + C_{ce} \dot{x}_e + G_{ce} &= F_{e} + F_{cm} \\
M_{c} \ddot{\omega} + C_{c} \dot{\omega} + G_{c} &= \tau
\end{align*}
\]

F \( _{cm} \) is added to (9) to take into account the interaction force between the tool and the contact surface, with respect to the generalized \( x \) coordinates.

In order to express the dynamics in the task space, one can multiply (9) on both sides by \( \dot{R}_0^T \). The following equation is then obtained:

\[
M_{ce} \ddot{x}_e + C_{ce} \dot{x}_e + G_{ce} = \dot{F}_e + F_{cm}
\]

where \( F_{cm} = \dot{R}_0^T F_{cm} \) is the contact force, \( M_{ce} = \dot{R}_0^T M_{ce} \dot{R}_0 \) is the inertia matrix, \( C_{ce} = \dot{R}_0^T C_{ce} \dot{R}_0 \) and \( G_{ce} = \dot{R}_0^T G_{ce} \), the matrix of centrifugal and coriolis terms, \( C_{c} = \dot{R}_0^T C_{c} \dot{R}_0 \), the matrix of gravity terms, all in the task space.

The active force in the task space can be transformed as:

\[
F_{ce} = \dot{R}_0^T F_e = \dot{R}_0^T J_e^T \tau = J_e^T \tau
\]

by introducing \( J_e^T = \dot{R}_0^T J_e^T \) and knowing that \( \tau = K_e I \), we obtain

\[
F_{ce} = J_e^T K_e I
\]

where \( J_e \) is the Jacobean matrix in the task space. Thus, the dynamics in the task space can be expressed as:

\[
M_{ce} \ddot{x}_e + C_{ce} \dot{x}_e + G_{ce} = J_e^T K_e I + F_{cm}
\]

The velocity vector \( \dot{x}_e \) can be partitioned according to the constrained and unconstrained directions as follows:

\[
\dot{x}_e = \begin{bmatrix} \dot{x}_c \\ \dot{x}_u \end{bmatrix}
\]

Let us assume that both the arm with the tool and the contact surface are quite rigid. Then, the motion of the end-effector in the constrained direction is negligible compared with the motion in the unconstrained direction. Therefore \( \dot{x}_c \) can be approximated as:

\[
\dot{x}_c = \begin{bmatrix} \dot{x}_c \\ 0 \end{bmatrix}
\]

We thus obtain the two expressions of the system below, the first one being for the position and the other for the force.

\[
\begin{align*}
\{ M_{ce} \ddot{x}_c + C_{ce} \dot{x}_c + G_{ce} = J_e^T K_e I \\
M_{ce} \ddot{x}_u + C_{ce} \dot{x}_u + G_{ce} = J_e^T K_e I + F_{cm}
\end{align*}
\]

where \( F_{cm} \) is the interaction force between the tool and the surface in force-controlled directions. \( M_{ce1} \), \( M_{ce2} \), \( C_{ce1} \), \( C_{ce2} \), \( G_{ce1} \) and \( G_{ce2} \) represent respectively the inertia matrices, the matrices of centrifugal and coriolis terms and the vectors of gravity terms, in the task space, taking into account decoupling. \( J_{e1} \) and \( J_{e2} \) are the Jacobean matrix decoupled in the task space. In the same way, the equation of the actuator given by the expression

\[
L\ddot{q} + R\dot{q} + K\dot{q} = \tau
\]

can be organized by multiplying on both sides by \( J_e^T \) as follows

\[
J_e^T L\ddot{q} + J_e^T R\dot{q} + J_e^T K\dot{q} = J_e^T \tau
\]

Knowing that \( \dot{q} = J_e^T \dot{x}_e \) and \( x = R_0 \dot{x}_e \), we can write

\[
\dot{q} = J_e^T \dot{x}_e = J_e^T \dot{x}_c
\]

Substituting (18) into (17), one obtains

\[
J_e^T L\ddot{q} + J_e^T R\dot{q} + J_e^T K\dot{q} = J_e^T \dot{x}_c + \dot{x}_u
\]

Let \( J_e^T \dot{x}_c = v_c \) and \( I_c = J_e^T I_c \). This implies

\[
I = J_e I_c
\]

and by differencing (20), one obtains

\[
L = J_e I_c + \dot{J}_e I_c \dot{q}
\]

We have a new expression for the actuator given as:

\[
J_e^T J_e \ddot{q} + J_e^T R\dot{q} + J_e^T K\dot{q} = J_e^T \dot{x}_c + \dot{x}_u
\]

Considering the constraint in the \( z \)-direction of the tool reference system, one obtains

\[
J_e \ddot{z} = \begin{bmatrix} J_e^T \ddot{x}_c \\ 0 \end{bmatrix}
\]

The uncoupled expression of this actuator then takes a new form after a new transformation of the coordinates, presented as:

\[
L_z = J_e^T L_{e1} R_{e1} J_{e1} ; R_z = J_e^T (L_{e2} + R_{e2}) J_{e2} ; K_z = J_e^T K_e J_{e2}
\]

The new models in the task space, for the position and the force respectively, have therefore the following expressions:

\[
\begin{align*}
\{ M_{ce1} \ddot{x}_c + C_{ce1} \dot{x}_c + G_{ce1} = K_{e1} F_{e1} \\
L_{e1} \ddot{q} + R_{e1} \tau + K_{e1} \dot{q} = v_{e1}
\end{align*}
\]

\[
\begin{align*}
\{ M_{ce2} \ddot{x}_c + C_{ce2} \dot{x}_c + G_{ce2} = K_{e2} F_{e2} \\
L_{e2} \ddot{q} + R_{e2} \tau + K_{e2} \dot{q} = v_{e2}
\end{align*}
\]
From the relation (25), \( v_{c1} \) is used to compensate the dynamic coupling forces and gravity, while \( v_{c2} \) is used for force tracking.

4. State-Space Representation

Let us consider the robot shown in Figure 1, which is electrically controlled by the currents \( I_1, I_2, I_3 \) and \( I_4 \). In this case, \( \theta_1, q_1, q_2 \) and \( q_3 \) are the articular joint positions and \( \dot{\theta}_1, \dot{q}_1, \dot{q}_2 \) and \( \dot{q}_3 \) represent the corresponding velocities.

Let us put the system's dynamics in a state-space representation. With the state vectors chosen as

\[ x_{c1} = \begin{bmatrix} x_c \end{bmatrix}, \quad x_{c2} = \begin{bmatrix} \dot{x}_c \end{bmatrix}, \quad \text{and} \quad x_{c3} = \begin{bmatrix} \dot{x}_{c2} \end{bmatrix} \]

the position dynamics of (24) become:

\[
\begin{align*}
\dot{x}_{c1} &= \xi_{c2} \\
\dot{x}_{c2} &= -M^{-1}(C_{c1}x_{c2} + G_{c1}) + M_{c1}K_{m1}x_{c1} \\
\dot{x}_{c3} &= -L_{c1}(R_c x_{c3} + K_{m1}x_{c2}) + L_{c1}v_{c1}
\end{align*}
\]  

(26)

where \( x_{c1}, x_{c2}, x_{c3} \) and \( x_{c1} \in \mathbb{R}^n \) represent, respectively, the position \( (x_p) \), speed \( (\dot{x}_p) \), current \( (I_{c1}) \), and state vectors and the voltages.

5. Backstepping Control Law based on Passivity

The design of the control law will be made in each recursive step of the system. The global stability of the robot is shown using the passivity approach. For each step \( i \) of this technique and for a given output \( y_{c1} \), we have a stabilizing function \( \alpha_{i+1} \), which plays the role of the corresponding control law, an input function \( u_{i+1} \), and a storage function \( W_{i+1} \), which is always fixed. Thus, the stability of the system for each step is guaranteed. The methodology and all the relations between the various elements are explained below. Thus, we will have three recursive steps, which correspond to each subsystem of the cascade system. Our goal here, is to find a control law \( v \) to stabilize the state-space system (3). Since the system (26) satisfies all the requirements of the strict-feedback systems, the design of the control law using the backstepping method can be applied as in Sepulchre et al. [1996] and Krstic et al. [1995].

Step 1: The first step of the system (26)

\[
\dot{\xi}_{c1} = \xi_{c2}
\]  

(27)

has the general form

\[
\dot{\xi}_{c1} = f(\xi_{c1}) + g(\xi_{c1})\xi_{c2}
\]

with \( f(\xi_{c1}) = 0 \) and \( g(\xi_{c1}) = I \), the identity matrix.

To obtain the input of the system \( \xi_{c2} \), it is necessary to determine a control \( u_1 \), such that

\[
\dot{\xi}_{c1} = u_1 = \alpha_0 + u_0
\]  

(28)

and a storage function \( W_0 \) such that the system becomes passive between the output \( y_0 \) and the input \( u_0 \). The output and storage functions must respect the conditions

\[
y_0 = \frac{\partial W_0}{\partial \xi_{c1}} g(\xi_{c1})
\]

(29)

with a trajectory tracking

\[
\dot{\xi}_{c1} = -K_0(\xi_{c1} - \xi_{c1d}) + \xi_{c1d}
\]  

(30)

where \( K_0 \) is a diagonal positive definite matrix.

Assuming the storage function as

\[
y_0 = \frac{1}{2}(\xi_{c1} - \xi_{c1d})^T(\xi_{c1} - \xi_{c1d})
\]

(31)

then, the output will be

\[
y_0 = \xi_{c1} - \xi_{c1d}
\]

(32)

Considering (28) and (31), the stabilizing initial input function is

\[
\alpha_0 = -K_0(\xi_{c1} - \xi_{c1d}) + \xi_{c1d}
\]

and \( u_0 = \xi_{c1d} \).

Step 2: Now, considering the two states of the system (26) as

\[
\begin{align*}
\dot{\xi}_{c2} &= f_2(\xi_{c2}) \\
\dot{\xi}_{c3} &= f_3(\xi_{c3})
\end{align*}
\]

(33)

by using the storage

\[
\begin{align*}
W_1 &= \frac{1}{2}(\xi_{c2} - \xi_{c2d})^T(\xi_{c2} - \xi_{c2d}) + \frac{1}{2}y_1^2
\end{align*}
\]

(34)

we get

\[
y_1 = \xi_{c2} - \xi_{c2d} - \alpha
\]

(35)

and

\[
\alpha_1 = (M_{c1}^T K_{m1})^{-1}(M_{c1}^T C_{c1} x_{c2} + G_{c1} - y_0 + \alpha_0 + u_1)
\]

(36)

with \( \alpha_0 = -K_0(\xi_{c2} - \xi_{c2d}) + \xi_{c2d} \) and \( u_1 = -K_1 y_1 \)

where \( K_1 \) is a diagonal positive definite matrix.

Step 3: In the same way, from relation (26),

\[
y_2 = \xi_{c3} - \xi_{c3d} - \alpha
\]

(37)

the control law can be written as follows

\[
v_{c1} = L_{c1} \left[ L_{c1}^T (R_c x_{c3} + K_{m1} x_{c2}) - y_1 + \frac{\partial \alpha_1}{\partial \xi_{c1}} \xi_{c1} + \frac{\partial \alpha_1}{\partial \xi_{c2d}} \xi_{c2d} + \frac{\partial \alpha_1}{\partial \xi_{c3d}} \xi_{c3d} + u_2 \right]
\]

(38)

with \( \frac{\partial \alpha_1}{\partial \xi_{c1}} = (M_{c1}^T K_{m1})^{-1}(-I - K_{c1} K_{c1}) \)

\[
\frac{\partial \alpha_1}{\partial \xi_{c2d}} = (M_{c1}^T K_{m1})^{-1}(M_{c1}^T C_{c1} x_{c2} - K_0 - K_0)(-M_{c1}^T C_{c1} x_{c2} + G_{c1}) + M_{c1}^T K_{m1} x_{c1d}
\]

(39)

and \( u_2 = -K_2 y_2 \).
Considering (25), the organization of the voltage representing the force can be applied as

\[
\begin{bmatrix}
M_{\text{at}2} \ddot{\xi}_{\text{at}2} + C_{\text{at}2} \dot{\xi}_{\text{at}2} + G_{\text{at}2} = K_{\text{at}2} \dot{\xi}_{\text{at}2} \\
L_{\text{at}2} \ddot{\xi}_{\text{at}2} + R_{\text{at}2} \dot{\xi}_{\text{at}2} + K_{\text{at}2} \xi_{\text{at}2} = v_{\text{at}2}
\end{bmatrix}
\]

(40)

considering:

\[K_{\text{at}2} \dot{\xi}_{\text{at}2} = f_{in}\]

(41)

The expression \(\xi_{o31}\) of the subsystem (40) of the manipulator is equal to

\[\xi_{o31} = K_{\text{o31}}^{-1}(M_{\text{o31}} \ddot{\xi}_{\text{o31}} + C_{\text{o31}} \dot{\xi}_{\text{o31}} + G_{\text{o31}} - K_{\text{o31}} \xi_{\text{o31}})\]

(42)

This implies for the subsystem (42) of the actuator that

\[\dot{V}_{\text{at}2} = L_{\text{at}2} \ddot{\xi}_{\text{at}2} + R_{\text{at}2} \dot{\xi}_{\text{at}2} + K_{\text{at}2} \xi_{\text{at}2} + K_{\text{at}2} \dot{\xi}_{\text{at}2}\]

(43)

from where

\[\dot{V}_{\text{at}2} = L_{\text{at}2} \ddot{\xi}_{\text{at}2} + R_{\text{at}2} \dot{\xi}_{\text{at}2} + K_{\text{at}2} \xi_{\text{at}2} + K_{\text{at}2} \dot{\xi}_{\text{at}2}\]

(44)

\[\dot{V}_{\text{at}2} = L_{\text{at}2} \ddot{\xi}_{\text{at}2} + R_{\text{at}2} \dot{\xi}_{\text{at}2} - K_{\text{at}2} \dot{\xi}_{\text{at}2} - K_{\text{at}2} \dot{\xi}_{\text{at}2}\]

(45)

5.1 Adaptive Backstepping Laws

Although the dynamic model of equation (24) of the manipulator is non linear in terms of position and speed, it can be expressed, in a linear form in terms of appropriate parameters, and the dynamics can thus be written as

\[K_{\text{at}2} \dot{\xi}_{\text{at}2} = W_{\text{at}2} p\]

(46)

where \(W_{\text{at}2} \in \mathbb{R}^{m \times n}\) and \(p \in \mathbb{R}^n\), represent respectively the matrix of known functions of the generalized coordinates and their higher derivatives, and the constant r-dimensional vector of robot parameters. The parameters are mass function \(m_i\), inertial moment \(l_i\), mass center \(x_i\), \(y_i\) and \(z\) and length \(l_i\) of the robot. The parameters estimation for this precise case will be done exclusively on the level of step 2.

Let

\[M_{\text{est}1} = \dot{M}_{\text{est}1} + \ddot{M}_{\text{est}1} ; C_{\text{est}1} = \dot{C}_{\text{est}1} + \ddot{C}_{\text{est}1}\]

and

\[G_{\text{est}1} = \dot{G}_{\text{est}1} + \ddot{G}_{\text{est}1}\]

where \(\dot{M}_{\text{est}1}\), \(\dot{C}_{\text{est}1}\), and \(\dot{G}_{\text{est}1}\) represent respectively the estimated matrices and \(\ddot{M}_{\text{est}1}\), \(\ddot{C}_{\text{est}1}\), and \(\ddot{G}_{\text{est}1}\) represent respectively the error matrices.

The expression of step 2 can be reformulated as

\[(M_{\text{est}1} + \ddot{M}_{\text{est}1}) \ddot{\xi}_{\text{est}1} + (C_{\text{est}1} + \ddot{C}_{\text{est}1}) \dot{\xi}_{\text{est}1} + G_{\text{est}1} - \dot{G}_{\text{est}1} - K_{\text{est}1} u_i\]

(47)

where \(u_i\) is the control of the system. After development

\[\dot{\xi}_{\text{est}1} = -M_{\text{est}1}^{-1}(\ddot{M}_{\text{est}1} \ddot{\xi}_{\text{est}1} + \ddot{C}_{\text{est}1} \dot{\xi}_{\text{est}1} + G_{\text{est}1} - \ddot{G}_{\text{est}1} - K_{\text{est}1} \xi_{\text{est}1})\]

(48)

then, the control law for stabilizing \(\dot{u}_i\) takes the following form

\[u_i = (M_{\text{est}1} + \ddot{M}_{\text{est}1})^{-1}(M_{\text{est}1} \ddot{\xi}_{\text{est}1} + C_{\text{est}1} \dot{\xi}_{\text{est}1} + G_{\text{est}1} - \dot{G}_{\text{est}1} - K_{\text{est}1} \xi_{\text{est}1})\]

(49)

knowing that \(u_i = \dot{u}_i\), and substituting (49) in (48), the step 2 can be reorganized as follows:

\[\dot{\xi}_{\text{est}1} = \xi_{\text{est}1} - \dot{\xi}_{\text{est}1}\]

(50)

The adaptation law will be done by taking into account the force and the position. For the position, relation (50) is reorganized as follows:

\[\dot{\xi}_{\text{est}1} = \xi_{\text{est}1} - \dot{\xi}_{\text{est}1}\]

(51)
\[ \dot{e}_r = \frac{(k_p + 1)}{k_v} e_r - \frac{K_{nc}}{k_v} W_{nc}\tilde{p} \] (67)

While putting (53) and (68) in matrix form, one has

\[ \begin{bmatrix} \dot{e}_r \\ \dot{e}_c \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & k_v + 1 \end{bmatrix} \begin{bmatrix} e_r \\ e_c \end{bmatrix} + \begin{bmatrix} W_{nc} \\ K_{nc} W_{nc} \end{bmatrix} \tilde{p} \] (68)

(68) is the form

\[ \dot{e}_c = A e_c + W\tilde{p} \] (69)

By considering the Lyapunov candidate function below and by calculating its derivative, one has

\[ v(t) = e^T Q e + 2\tilde{p}^T e \] (70)

Taking into account (73), we have the two expressions as shown below

\[ \dot{v} = -e^T e \leq 0 \] (74)

(74) is semi-definite positive. Then \( \tilde{p} \) is chosen such as

\[ 2\tilde{p}^T W \dot{Q} Q \tilde{e} + 2\tilde{p}^T \Gamma \tilde{p} = 0 \] (75)

(75) implies respectively the following adaptation law

\[ \dot{\tilde{p}} = \Gamma \dot{Q} Q \tilde{e} \] (76)

Q of the adaptation law is the solution of equation (72). We ultimately have the three laws of adaptive control as follows

\[ \begin{align*}
\dot{v}_c &= L \left[ L_i^T (R \dot{c}_s) + K_{nc} \xi c - y_1 + \frac{\partial\alpha_1}{\partial \xi c} \dot{\xi} \right] + \\
\frac{\partial\alpha_1}{\partial \xi c} \dot{\xi} + \frac{\partial\alpha_1}{\partial \xi c} \dot{\xi} + \frac{\partial\alpha_1}{\partial \xi c} \dot{\xi} - K_2 Y_2 \\
\dot{v}_{c2} &= R_t \left[ \xi_{c2} - k_n (\xi_{c2} - \xi_{c2}) - k_n (\xi_{c2} - \xi_{c2}) \right] (77)
\end{align*} \]

6. System Parameters

The mass and inertial parameters are given below

<table>
<thead>
<tr>
<th>( p_i )</th>
<th>( p_{i2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_1 + m_2 )</td>
<td>( m_3 + m_4 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( m_3 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( m_3 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( m_3 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( m_3 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( m_3 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( m_3 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( m_3 )</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>( m_3 )</td>
</tr>
</tbody>
</table>

7. Simulation Results
8. Results Analysis

This section presents the simulation results. The trajectory tracking is perfect, as shown in figure 3. The motion tracking and the errors generated are acceptable, as shown in figures 4 and 5. The parameters shown in figures 6 are constant but do not converge with their real values. The desired current was calculated for a desired force of 15 N. As shown in figure 7, the current tracking representing the force is correct. The voltage controls shown in figure 8 are acceptable.

9 Conclusion

In this paper, SFB adaptive control laws have been derived by incorporating both manipulator and actuator dynamics with uncertainties parameters. Simulations were performed with a four-link robot. The results obtained for the control law are relatively correct, and the controlled system performance is shown to be acceptable by the graphical results presented throughout the paper. Finally, global stability was reached for the various types of controls.

REFERENCES


