

INTERACTION CONTROL EXPERIMENTS FOR A ROBOT WITH ONE FLEXIBLE LINK

L. F. Baptista

*Escola Náutica Infante D. Henrique
Av. Engenheiro Bonneville Franco, 2770-058 Paco de Arcos*

J. M. M. Martins, J. M. G. Sá da Costa

*Instituto Superior Técnico, GCAR/IDMEC
Av. Rovisco Pais, 1049-001 Lisboa Codex*

Keywords: Flexible-link manipulator, closed-loop inverse kinematics, interaction control, real-time control.

Abstract: One of the major drawbacks of flexible-link robot applications is its low tip precision, which is an essential characteristic for applications with interaction control with a contact surface. In this work, interaction control strategies considering rigid and flexible contact surfaces are applied on a two degrees of mobility flexible-link manipulator. The interaction strategies are based on the closed-loop inverse kinematics algorithm (CLIK) to obtain the angular references to the joint position controller. The control schemes were previously tested by simulation and further implemented on the flexible-link robot. The obtained experimental results exhibit a good force tracking performance, especially for a rigid surface, and reveal the successful implementation of these control architectures for a robot with one flexible link.

1 INTRODUCTION

The evolution of industrial manufacturing, lead to the necessity of optimize the production, where the main goal is to achieve best quality products at lower prices. The manipulator robot is a crucial automation equipment that fulfill these requirements, due to its high productivity and easy adaptation to a large number of complex and repetitive tasks. The manipulator robots have also the ability to work in adverse environments to the human workers. Due to these characteristics, the study of robot manipulator control has received a growing attention by a lot of researchers during the last decades, in order to design robots with high performance (Canudas de Wit et al., 1998).

In general, industrial robots have rigid mechanical elements which leads to a high power consumption. To overcome this disadvantage, lightweight and flexible links have been considered in the construction of new robots. These new links allow the same mobility capacity as the rigid robots with a lower power consumption. Also, due to the lighter weight of the links, the interaction with the environment, especially in the case of collision, cause less damage.

When a manipulator robot executes an interaction task, the tip or end-effector enters in contact with the environment and a certain force is exerted on the surface. Since it's necessary to achieve an high preci-

sion tip position to obtain a good interaction force control, advanced control algorithms have been developed to obtain a high force tracking performance (Zeng and Hemami, 1997). However, flexible-link manipulators exhibit an important drawback in comparison with rigid robots, due to the difficulty in control its tip or end-point position. The flexibility rises the dynamic coupling, the non-linearities, and gives to the robot infinite degrees of freedom derived from the vibration modes of the flexible elements. Due to these vibrations, the system becomes a non-minimum phase system (Talebi et al., 1998). The zeros in the right semi-plan, due to the non minimum phase lead to an unstable system, when the tip position is directly controlled through feedback.

To avoid these drawbacks, several techniques to efficiently control flexible-link robots have been studied. The control of a flexible manipulator at the joint level has been established by a lot of authors like (Khorrami and Jain, 1994) for the tracking problem and (Vandegrift et al., 1994) for the regulation problem, among others. One of the proposed strategies to solve the inverse kinematics problem for flexible arms, was derived from the closed loop inverse kinematics algorithm (CLIK) developed for rigid manipulators (Siciliano, 1990). The inverse kinematics formulation with feedback of joint coordinates and deflection variables for constrained flexible manipula-

tors was developed by (Siciliano, 1999; Siciliano and Villani, 2001). Finally, more complex algorithms for solving the inverse kinematics problem at high speed velocities with flexible manipulators have been proposed by (Cheong et al., 2004).

The purpose of this work is to obtain experimental results with interaction control algorithms for a planar robot with two revolute joints and two links, where the second link is flexible. The control strategies were implemented considering the CLIK algorithm to obtain the desired angular references to the joint position controller. The interaction control algorithm was applied considering rigid and flexible contact surfaces, respectively.

The outline of the paper is as follows. Section 2 describes the flexible link forward and inverse kinematics formalism and the closed loop inverse kinematics algorithm (CLIK). In section 3 a brief overview of the CLIK-based interaction controllers for rigid and flexible surfaces are described. Section 4 describes the planar flexible-robot setup, the hardware and software control architecture considered for the real-time experiments. In section 5 the obtained interaction control results in real-time are presented. Finally, in section 6 some conclusions are drawn.

2 FLEXIBLE LINK KINEMATICS

Let us consider a planar robot with two degree of mobility, where the first link is rigid and the second link is flexible, as depicted in fig. 1.

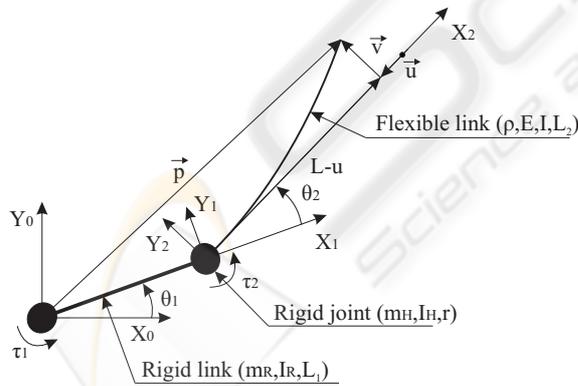


Figure 1: Planar flexible robot schematics.

The robot's flexible link can be modeled as an Euler-Bernoulli cantilever, with length L . The flexible link is attached to a rigid joint. When a torque τ is applied to the rigid joint, the flexible link is rotated by an angle θ between the body frame $\{X, Y, Z\}$ and the base reference frame $\{X_0, Y_0, Z_0\}$, as illustrated in fig. 2.

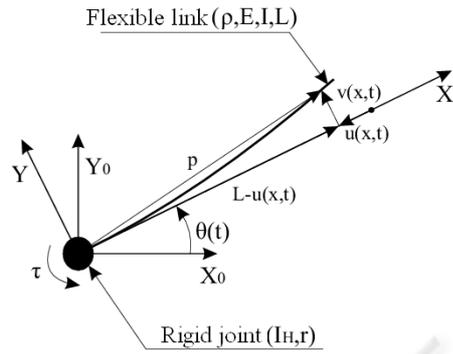


Figure 2: Flexible link scheme deflection caused by an applied torque.

In this figure, $v(x, t)$, represents the lateral displacement point x along the flexible link, relative to X axis. Thus, the projection of x on X axis will be given by $L - u(x, t)$ coordinate. Two models are considered to obtain the length reduction coordinate $u(x, t)$: linear and quadratic models. The linear model approach considers that length reduction is null, i.e. $u(x, t) = 0$. In the quadratic model approach, length reduction is calculated by the following expression

$$u(x, t) = -\frac{1}{2} \int_r^x \left(\frac{dv}{d\xi} \right)^2 d\xi \quad (1)$$

In fig. 3 both length reduction approach due to an elastic link deflection are represented.

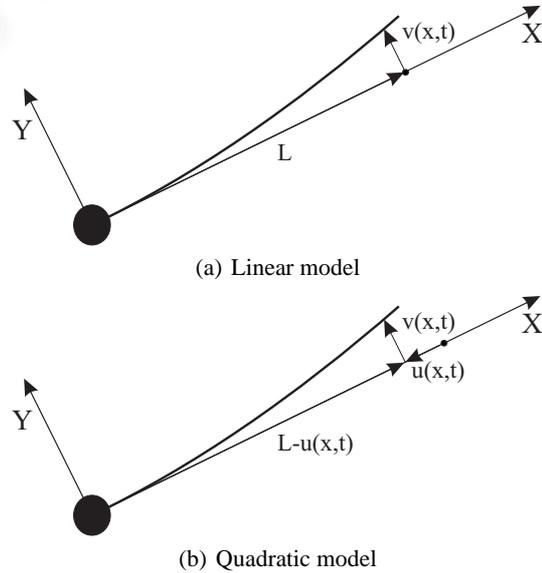


Figure 3: Elastic link length reduction models.

In the following, the Assumed Modes discretizing

method (Martins, 2000) will be considered for the lateral displacement v calculation. This method consider

$$v(x, t) = \chi(x)\delta(t) \quad (2)$$

where χ are the normalized mode shapes and δ are the generalized elastic coordinates. Considering only the two first vibration modes, v is given by

$$v(x, t) = \sum_{k=1}^2 \chi_k(x) \delta_k(t) \quad (3)$$

For this particular flexible-link robot, χ_i are given by (Nabais, 2002):

$$\begin{aligned} \chi_1 &= 3\left(\frac{x}{L}\right)^2 - 2\left(\frac{x}{L}\right)^3 \\ \chi_2 &= -\left(\frac{x}{L}\right)^2 + \left(\frac{x}{L}\right)^3 \end{aligned} \quad (4)$$

where x as referred above, is a point on the elastic link. Vector \vec{p} represented in fig. 1 is the position on the link relative to reference frame $\{X_0, Y_0, Z_0\}$. This vector is represented by

$$\vec{p} = \vec{p}_1 + R_0^2 (\vec{p}_2 - \vec{u} + \vec{v}) \quad (5)$$

where R_0^2 represents the rotation matrix and \vec{p}_1 is the position on the first link relative to reference frame $\{X_0, Y_0, Z_0\}$. Also, \vec{p}_2 is the non-deformed second link end-effector position relative to reference frame $\{X_2, Y_2, Z_2\}$. Summing \vec{p}_2 , \vec{u} and \vec{v} , leads to:

$$\vec{p} = \vec{p}_2 - \vec{u} + \vec{v} \quad (6)$$

The rotation matrix R_0^2 that describes the position p relative to reference frame $\{X_0, Y_0, Z_0\}$ is represented by:

$$R_0^2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (7)$$

Considering that p is the end-effector or tip position, \vec{p} represents the forward kinematics of the flexible-link robot. Thus, the forward kinematic equations are given by:

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} L_1 \cos(\theta_1) \\ L_1 \sin(\theta_1) \end{bmatrix} + R_0^2 \begin{bmatrix} L_2 - \|\vec{u}\| \\ \|\vec{v}\| \end{bmatrix} \quad (8)$$

which leads to the following equations:

$$\begin{aligned} p_x &= L_1 \cos(\theta_1) + (L_2 - \|\vec{u}\|) \cos(\theta_1 + \theta_2) \\ &\quad - \|\vec{v}\| \sin(\theta_1 + \theta_2) \\ p_y &= L_1 \sin(\theta_1) + (L_2 - \|\vec{u}\|) \sin(\theta_1 + \theta_2) \\ &\quad + \|\vec{v}\| \cos(\theta_1 + \theta_2) \end{aligned} \quad (9)$$

where $\|\vec{u}\|$ and $\|\vec{v}\|$ are given by eq. (1) and eq. (3), respectively.

The inverse kinematics equations relate the cartesian position coordinates, given by eq. (9), and the joint θ and deflection δ coordinates. Replacing the equations (1)-(3) into (9), two equations and four unknown variables ($\theta_1, \theta_2, \delta_1, \delta_2$) are obtained. Thus, the system is undetermined and other methods should be exploited to overcome this problem.

2.1 CLIK Algorithm

To solve the problem presented above, the Closed Loop Inverse Kinematics algorithm (CLIK) developed for rigid robots was adopted in this work, according to (Siciliano, 1999). This algorithm feeds back the joint angles θ calculated by the CLIK algorithm in a closed loop dynamic system in order to obtain the reference values to the joint position controller. This algorithm is given by (Siciliano and Villani, 2001):

$$\dot{\theta}_d = J_p^T(\theta) K_P (p_d - p) \quad (10)$$

where:

- $\dot{\theta}_d$ are the desired joint velocities,
- $J_p = J_\theta$ i.e., the rigid part of the Jacobian matrix,
- p_d is the desired tip position,
- p is the cartesian position determined by forward kinematics with coordinates θ ,
- K_P is a proportional gain matrix

To obtain the joint references θ_d for real-time control, it is necessary to integrate the joint velocities given by eq. (10). Thus, the discrete version of θ_d , is given by:

$$\theta_d(t_{k+1}) = \theta_d(t_k) + T_s J_p^T(\theta_d(t_k)) K_P (p_d(t_k) - p(t_k)) \quad (11)$$

where

- θ_d are the reference joint angles for the position controller,
- T_s is the sampling period

3 INTERACTION CONTROL

3.1 Rigid Surface

Let us consider that the robot is in contact with a rigid surface. The restriction imposed by the surface, is described by

$$\phi(p) = \phi(k(\theta, \delta)) = 0 \quad (12)$$

Assuming that the robot is in a static condition, the deflections satisfy the following equation:

$$K\delta = -J_\delta^T(\theta, \delta) \lambda j_\phi \quad (13)$$

where K is the robot stiffness matrix, j_ϕ is the equation gradient (12), defined by

$$j_\phi = \left(\frac{\partial \phi}{\partial p} \right)^T \quad (14)$$

and λ is the Lagrange multiplier associated with the restriction. From eq. (13), δ is given by

$$\delta = -K^{-1} f \quad (15)$$

where,

$$f = J_\delta^T(\theta, \delta)\lambda j_\phi \quad (16)$$

Differentiating (15) leads to

$$\dot{\delta} = -K^{-1}J_f(\theta, \delta)\dot{\theta} \quad (17)$$

where

$$J_f = \frac{\partial f}{\partial \theta} = \lambda j_\phi \frac{\partial J_\delta^T(\theta, \delta)}{\partial \theta} \quad (18)$$

and,

$$\dot{\theta} = J_p^T(\theta, \delta)K_P(p_d - p) \quad (19)$$

The Jacobian J_p is defined by

$$J_p = J_\theta(\theta, \delta) - J_\delta(\theta, \delta)K^{-1}J_f(\theta, \delta) \quad (20)$$

where e_p is the difference between the desired tip position and the cartesian position determined by the robot forward kinematics. The joint angles and the deflection coordinates are given by the CLIK algorithm. Discretizing these coordinates, leads to

$$\theta_d(t_{k+1}) = \theta_d(t_k) + T_s J_p^T(\theta_d(t_k), \delta_d(t_k))K_P e_p(t_k) \quad (21)$$

and

$$\delta_d(t_{k+1}) = -K^{-1}(J_\delta^T(\theta_d(t_k), \delta_d(t_k))\lambda(t_k)j_\phi) \quad (22)$$

Notice that the Jacobian J_p not only depends on θ coordinates, but also depends on δ coordinates. This happens because δ depends on f_d , i.e. the force that should be applied by the robot on the surface.

The overall interaction controller applies a joint position PD controller plus the desired force f_d . The control law is described by

$$\tau = K_p(\theta_d - \theta) - K_d\dot{\theta} + J_\theta^T(\theta, \delta)f_d n \quad (23)$$

where n is the normal to the surface and f_d is the desired force. Notice that the PD controller doesn't control directly the interaction force between the tip of the link and the contact surface. In fact, only the joint angles are controlled. In fig. 4 the simplified block diagram of the CLIK-based interaction controller considering a rigid surface is represented.

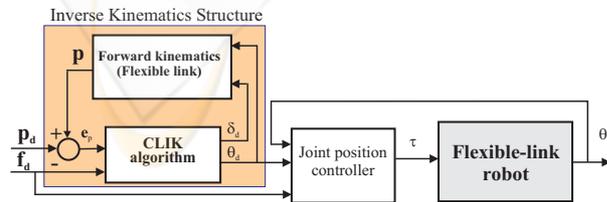


Figure 4: Simplified block diagram of the CLIK-based interaction controller for a rigid surface.

3.2 Flexible Surface

In this case, the interaction control algorithm is similar to the previous presented algorithm for the rigid surface. The main difference concerns with the calculation method of the reference force, which is based on the estimated stiffness surface coefficient k_e . Notice that, for simplicity, the environment is modeled as a linear spring. Due to this assumption, the deflection coordinates δ , the Jacobian J_p and the trajectory planning algorithm will be slightly different (Siciliano and Villani, 2001).

In the interaction control algorithm considering a rigid surface described above, the contact force is represented by the Lagrange multiplier λ . In this case, the force is represented by

$$f_d = k_e \vec{p}_f \quad (24)$$

In the interaction controller with a flexible surface, the desired trajectory has two components, one tangent to the contact surface \vec{p}_s and another component normal to the surface \vec{p}_f . With these two components, the desired reference trajectory is represented by the following equation

$$\vec{p}_d = \vec{p}_s + \vec{p}_f \quad (25)$$

where \vec{p}_d is the desired tip cartesian position, and \vec{p}_f is determined by

$$\vec{p}_f = k_e^{-1} f_d \vec{n} \quad (26)$$

In figure 6 the geometric representation of the desired tip position, considering the desired force f_d on the surface is presented.

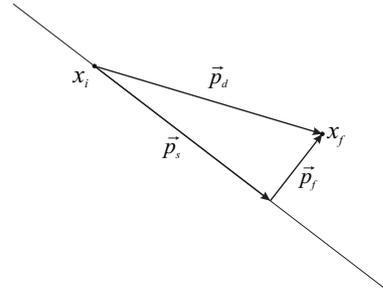


Figure 5: Geometric representation of the desired tip position.

For the flexible contact surface, the Jacobian J_p is then given by

$$J_p = J_\theta(\theta, \delta) - k_e J_\delta(\theta, \delta)K^{-1}J_f(\theta, \delta) \quad (27)$$

where,

$$J_f = \frac{\partial J_\delta^T n}{\partial \theta} (n^T p - n^T p_e) + J_\delta^T n \frac{\partial n^T p}{\partial \theta} \quad (28)$$

The discrete version of the deflection coordinates algorithm, is given by

$$\delta_d(t_{k+1}) = -K^{-1}(k_e J_\delta^T(\theta_d(t_k), \delta_d(t_k)) + n^T p_e) \times n(n^T p(t_k) - n^T p_e) \quad (29)$$

where

- p_e is the non-deformed coordinate of the surface,
- J_θ is the rigid part of the robot's Jacobian,
- J_δ is the flexible part of the robot's Jacobian,

In fig. 6 the simplified block diagram of the CLIK-based interaction controller considering a flexible surface is represented.

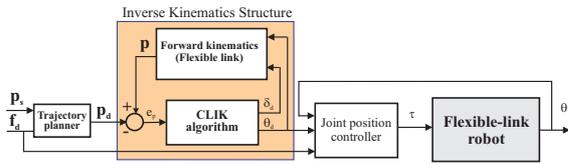


Figure 6: Simplified block diagram of the CLIK-based interaction controller considering a flexible surface.

In (Siciliano, 1999; Siciliano and Villani, 2001), the Jacobian J_f is considered as only dependent on θ . This assumption is valid when small link deflections are considered, i.e. they can be neglected. In this work, the robot's link is extremely flexible, and this assumption is not valid. For this reason, it is assumed that Jacobian J_f is fully dependent on θ and δ .

When the robot is in contact with the environment, the interaction controller described above, apply the desired force f_d on the surface through the joint position PD algorithm described by eq. (29).

4 ROBOT EXPERIMENTAL SETUP

For the purpose of analyze the interaction control algorithm performance, an experimental setup was built at the Robotics Laboratory. In fig. 7, a picture of the planar flexible-link robot used for the experiments, is presented.

Table 1 exhibits the most important physical parameters of the joints and links of this robot.

The control hardware used to drive the flexible robot consists of a host PC Pentium IV 3 GHz computer that runs the Matlab/Simulink software and a target PC Pentium 200 MHz computer, where the real-time target software runs under the Matlab/xPC environment. The signals are processed through a low cost ISA-bus servo I/O board from SERVO TO GO, INC., and the electric d.c. joint motors are driven by linear power amplifiers configured to operate as current



Figure 7: Picture of the experimental robotic setup.

Table 1: Physical parameters of the robot.

Joint 1 and rigid link 1	
L_R - link length	0.32 m
I_{R0} - Inertia of the link	0.25 kgm^2
I_{m1} - Inertia of the actuator	0.093 kgm^2
Joint 2	
r - Radius of the joint	0.075 m
I_H - Rotating inertia	$13.22 \times 10^{-4} \text{ m}^4$
M_H - Mass of the joint	0.47 kg
I_{m2} - Inertia of the actuator	0.024 kgm^2
Flexible link 2	
L - Link length	0.5 m
e - Link thickness	0.001 m
h - Link width	0.02 m
I - Cross section inertia	$1.67 \times 10^{-12} \text{ m}^4$
I_b - Link inertia	$99 \times 10^{-4} \text{ kgm}^2$
m_b - Link mass	0.0785 kg

amplifiers. In this functioning mode, the input control signal is a voltage in the range of $\pm 10 \text{ V}$ with current ratings in the interval $[-3, 3] \text{ A}$. The deflection of the elastic link is measured by three full bridge strain gage sensors located along the link and processed by HOTTINGER BM instrumentation amplifiers. The contact forces are measured by a JR³ 6-axis force/torque sensor mounted on the contact surface (see Fig. 10 for details). The force sensor hardware provide decoupled and digitally filtered data at a frequency rate of 8 KHz for each channel. Figure 8 represent the overall hardware and software control architecture for the flexible-link robot.

5 EXPERIMENTAL RESULTS

For the purpose of analyzing the interaction control performance, the control methodologies presented in sections 3.1 and 3.2 are applied through experimentation to the planar robot represented in fig. 7. Notice that all the interaction tasks described on this

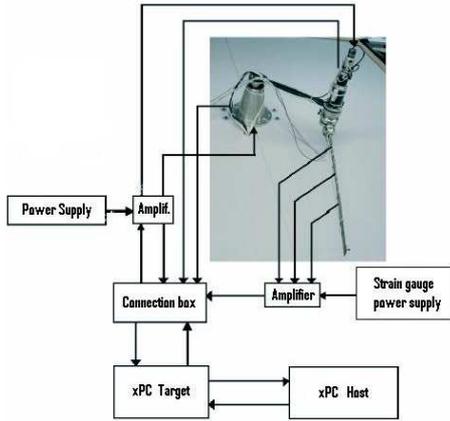


Figure 8: Hardware and software control architecture for the flexible-link robot.

section were previously tested by simulation in Matlab/Simulink in order to obtain the best performance. For this purpose, Virtual Reality Toolbox from Matlab was used to build the tridimensional (3D) robot model, as depicted in fig. 9.



Figure 9: Picture of the 3D planar robot model built in Matlab/Virtual Reality Toolbox. Notice that the dot line represents the undeformed flexible contact surface.

5.1 Rigid Surface

The first task consists of applying a force profile on the rigid surface while maintaining the robot's position. In the second task, the robot should move along the rigid surface with simultaneously application of the desired force profile on the surface, as represented in Fig. 10.

The sampling frequency is 1 kHz and the following controller gains were used in all the experimental

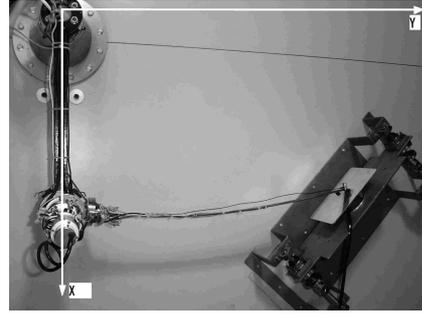


Figure 10: Top view of the flexible robot executing an interaction task.

tasks: $K_P = [2000 ; 2000]$ for the CLIK algorithm and $K_p = [3000 ; 600]$, $K_d = [20 ; 10]$ for the PD controller. Notice that all the experiments were executed considering that tip is already in contact with the surface before the execution of the task.

Due to the maximum allowed values for the deflections adjusted in the robot's supervision and control software and the high degree of flexibility of the link, the maximum force that is possible to apply by the flexible link on the surface, is 1 N. In fig. 11 the results for a task where only a desired force trajectory is applied on the surface are presented. The force is applied at the initial contact point, $P=[0.32 ; 0.575]$ m. The contact surface has 45° of inclination with the reference base frame x -axis. The reference force profile has a maximum value of 0.9 N, a growing time of 15 seconds and a full evolution time of 50 seconds.

In fig. 12 the results for a task with force and position reference trajectories are presented. The reference force has the same profile of the first task and the position reference trajectory executes a straight line movement with a cycloidal profile of 5 cm in 5 seconds along the rigid surface.

From the analysis of the plots, is possible to observe a good force tracking performance in static conditions (fig. 11). Also, when the robot executes a movement along the rigid surface, while executing the desired force profile, an acceptable force tracking performance with low force errors is observed along the trajectory (fig. 12). In all these experiments an overshoot is observed when the applied force begins to decrease, due to tip/surface contact friction effects. Notice that the desired force is applied on the surface without force feedback, but the force errors are kept small. These results validate the interaction control strategy described in section 3.1.

5.2 Flexible Surface

In order to obtain preliminary experimental results for a flexible environment, a soft foam was fixed on the

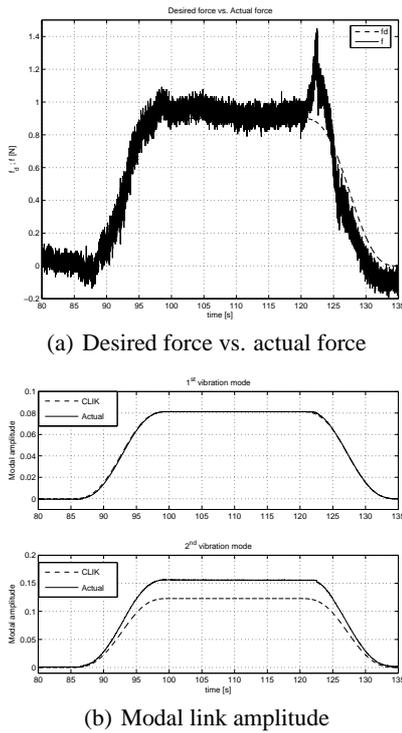


Figure 11: Rigid surface: contact force and modal amplitudes of the flexible link.

plate attached to the force sensor. The overall estimated stiffness coefficient of the device is $k_e \approx 110$ N/m. Since the interaction controller for the flexible surface revealed to be extremely sensitive for k_e values larger than 15 N/m, a desired path profile f_d with a maximum value of 1 N was planned considering the estimated stiffness environment but setting $k_e = 15$ N/m in eq. (29) in order to observe the correspondent applied force and deformation of the environment. From fig. 13 it is possible to observe an acceptable force tracking behavior in static conditions. In this case, a force overshoot is observed at the end of the growing path due to the flexible environment characteristics. From the force trajectory plot, is possible to observe that applied force reach the desired value of 1 N. However, since there are a significant gap between the estimated stiffness coefficient and the k_e value used in eq. (29), the modal amplitudes of the flexible link will not match the desired ones calculated by the CLIK algorithm (fig. 13-b). Also, due to the k_e mismatch described above, the cartesian trajectory evolution along x -axis will exhibit a poor tracking performance, as depicted on fig. 14-a. Finally, is possible to observe that joint position controller reveal an excellent tracking performance (see fig. 14-b for details).

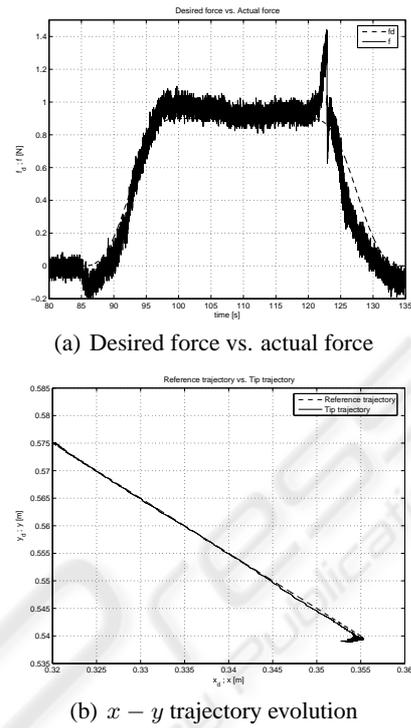


Figure 12: Rigid surface: contact force and cartesian trajectory along the surface.

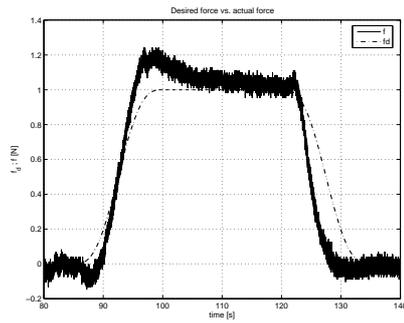
6 CONCLUSIONS

In this article interaction control strategies for a manipulator robot with a two degrees of mobility and a flexible link were analyzed by simulation and experimentation. The interaction control results reveal the successful implementation of the control algorithms in real-time for a robot with one flexible link. The interaction control results were obtained considering rigid and flexible contact surfaces.

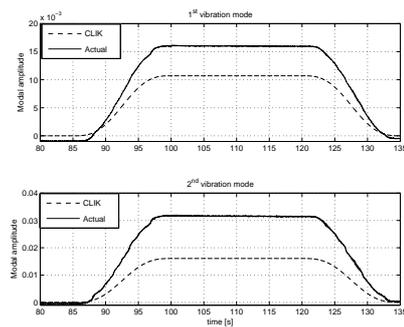
Future research will concentrate on the improvement of the real-time software functionality, the study of more complex inverse kinematic algorithms for flexible arms and the improvement of the interaction controller robustness for the flexible contact surface.

ACKNOWLEDGEMENTS

The authors would like to thank to Professors Bruno Siciliano and Luigi Villani for their support during the CLIK algorithm implementation in Matlab/Simulink.

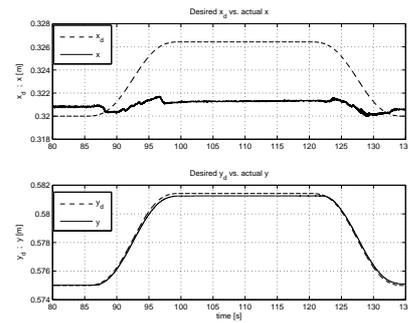


(a) Desired force vs. actual force

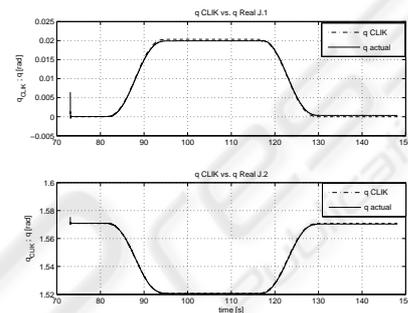


(b) Modal link amplitude

Figure 13: Flexible surface: contact force and modal amplitudes of the flexible link.



(a) Desired trajectory vs. actual trajectory



(b) CLIK reference angles vs. actual angles

Figure 14: Flexible surface: Cartesian and joint trajectories.

REFERENCES

- Canudas de Wit, C., Siciliano, B., and Bastin, G. (1998). *Theory of Robot Control*. Springer Verlag.
- Cheong, J., Chung, W. K., and Youm, Y. (2004). Inverse kinematics of multilink flexible robots for high-speed applications. *IEEE Transactions on Robotics and Automation*, 20(2):269–282.
- Khorrani, F. and Jain, S. (1994). Nonlinear control with end-point acceleration feedback for a two-link flexible manipulator: experimental results. *Journal of Robotic Systems*, 11:591–603.
- Martins, J. (2000). Modeling and identification of flexible manipulators towards robust control. Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisbon.
- Nabais, J. (2002). Sliding model control of a flexible link robot (*in portuguese*). Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisbon.
- Siciliano, B. (1990). A closed-loop inverse kinematic scheme for on-line joint based robot control. *Robotica*, 8:231–243.
- Siciliano, B. (1999). Closed-loop inverse kinematics algorithm for constrained flexible manipulators under gravity. *Journal of Robotics Systems*, 16:353–362.
- Siciliano, B. and Villani, L. (2001). An inverse kinematics algorithm for interaction control of a flexible arm with a compliant surface. *Control Engineering Practice*, 9:191–198.
- Talebi, H., Khorasani, K., and Patel, R. (1998). Neural network based control schemes for flexible-link manipulators: simulations and experiments. *Neural Networks*, 11:1357–1377.
- Vandegrift, M., Lewis, F., and Zhu, S. (1994). Flexible-link robot arm control by feedback linearization/singular perturbation approach. *Journal of Robotic Systems*, 11:591–603.
- Zeng, G. and Hemami, A. (1997). An overview of robot force control. *Robotica*, 15:473–482.