15th Triennial World Congress, Barcelona, Spain

EMBEDDED CONTROL ARCHITECTURE FOR A REDUNDANT ROBOTIC ARM

A. Barreca G. Cannata L. Dentone F. Giorgi

Dept. of Communications, Computer and Systems Sciences, University of Genova, Via Opera Pia 13, 16145 Genova, Italy

Abstract: This paper presents some design aspects related with the development of a human-like robot arm, with fully embedded control architecture. The specific features of the proposed mechanical design are discussed first. Then, a candidate distributed and embedded control architecture is eventually proposed.

Keywords: Robotic Manipulators, Robot Kinematics, Embedded Systems, Fieldbus, Distributed Control, Architectures

1. INTRODUCTION

The topic of this paper is to present an ongoing project aiming at the development of a small robot arm featuring motions and interaction capabilities similar to those of a human one.

In the next section the mechanical structure of the proposed robot arm is discussed. In section III, a fully distributed and embedded control architecture for controlling a robot arm is presented within the framework of *Task Function* based control architectures.

2. MECHANICAL STRUCTURE

The major goal of the project is to develop a human-like robot arm as a testbed for experiments for man-robot interaction. Figures 1, and 2 sketch the mechanical structure of the arm. The robot features 8 degrees of freedom: 4 for the shoulder, 1 for the elbow and 3 for the wrist. The reach is about 1 m with a large working volume ensured by a highly redundant shoulder kinematics, and the adoption of a particular kinematics of the elbow which makes possible complete opposition of the forearm on the arm. Expected payload is 1 Kg including the end-effector.

The robot external envelope has been designed to be smooth enough to host skin-like tactile sensors, and to allow whole arm manipulation tasks. All cables, actuators and low level control/sensing electronics (see section on control) will be hosted inside the arm.

2.1 *Wrist*

The wrist geometry and kinematics has been implemented as a *roll-pitch-yaw* sequence of intersecting axes. This allows emulating the typical human wrist motions. Roll-pitch-yaw axes make possible to implement a singularity-free wrist, as long as the absolute magnitude of yaw angle is less than 90^o.

2.1.1. *Roll Axis* The roll axis (axis 6) is aligned with the forearm and allows its rotation. Rotation about the roll axis is constrained only by the limits of torsion of the cables passing inside the arm; the wrist roll angle has a range of $\pm 90^o$, and is equivalent to the corresponding human motion. A DC motor placed in the forearm actuates the wrist roll axis. The coupling of the motor with the elbow stub is instrumented and operates as a torque sensor; therefore, it is possible to feedback, external torques as well as wrist dynamics about the roll

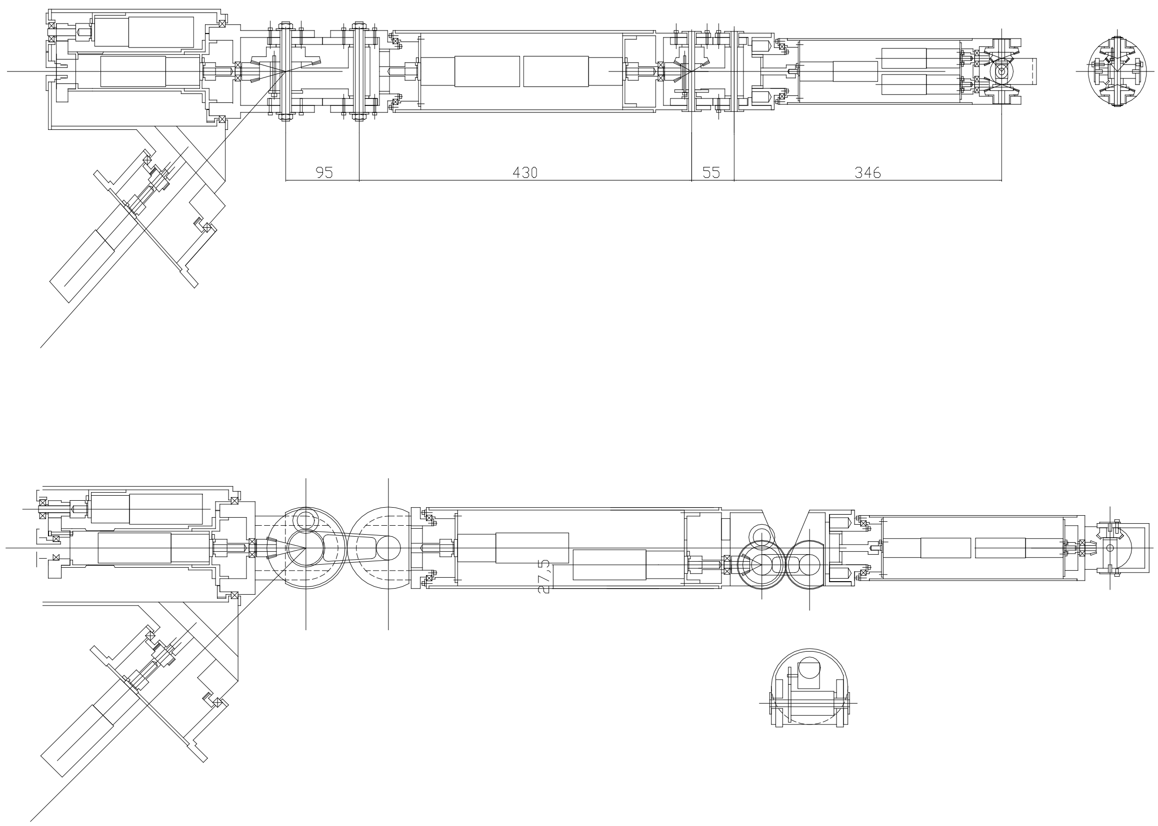

Fig. 2. Side view of the proposed robot arm

axis. The torque sensor is based on strain-gages and is interfaced with local custom electronics providing bridge excitation, signal conditioning and data acquisition.

2.1.2. *Yaw-Pitch Axes* The yaw (axis 7) and pitch (axis 8) axes are mildly coupled and actuate the wrist flange supporting the end-effector [1]. The flange is moved by a differential mechanism driven by two tandem controlled DC motors: motor 7 drives the yaw axis while motor 8 compensate the yaw motions and drives the pitch axis. The yaw range of the present design is larger than $\pm 45^o$ while pitch angle is in excess of $\pm 90^o$.

Due to the intrinsic coupling of the yaw-pitch motions and in order to save room for further upgrades of the arm, both the yaw and pitch motors are controlled by a single control node (see the next section).

2.2 *Elbow*

Elbow geometry affects both the workspace of the arm and the capability of performing whole arm manipulation tasks. Standard industrial robots

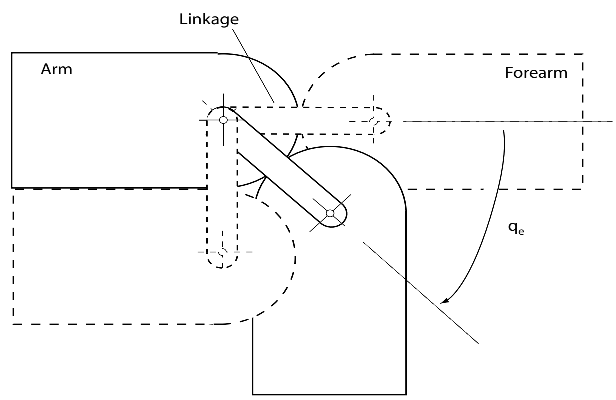

Fig. 3. Mobility of the forearm with respect to the arm

(e.g. PUMA) obtain a wide elbow rotation by placing an offset between arm and forearm; this geometry makes hardly possible the implementation of whole arm manipulation tasks. A zero offset design has been proposed by Barret Technology (Leeser *et al.*, 1994): it allows an elbow rotation of 230^o which makes possible to place the forearm and the arm side by side.

Another solution, inspired by the work of (Rosheim, 1989), is proposed in the present design. The elbow geometry features zero offset between arm and forearm as in (Leeser *et al.*, 1994), but allows for a smoother robot envelope. The basic geometry of the joint is sketched in figure 3.

---

[1] A specific end-effector design is not part of this project. For a more detailed discussion on an articulated human-like hand see (Caffaz and Cannata, 1998), (Casalino *et al.*, 2001).

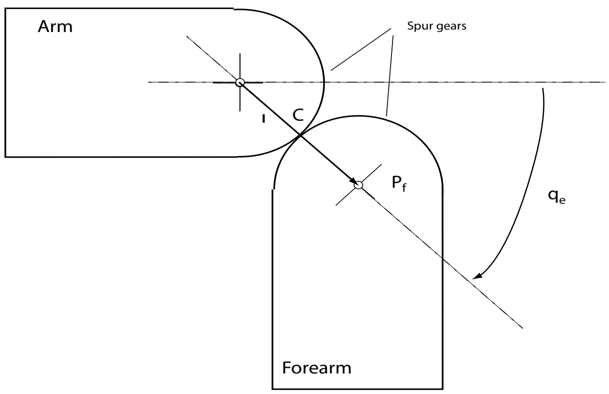

Fig. 4. The motion of the forearm is a combined roto-translation

The forearm is driven by a linkage actuated by a DC motor, and it is constrained to rotate with respect to the arm by two couples of spur gears. The driving linkage as well as the motor is instrumented with strain gauges to sense torques transmitted about the elbow axis.

Although in the present design the rotation of the forearm with respect to the arm is larger than 180^o, we are interested in human like motions (0^o - fully extended arm, 180^o fully flexed arm), which keep arm singularities at the boundary of the elbow angle range. It is interesting to remark that by properly adjusting gear ratio and size of the supporting gears the rotation of the forearm could be made as large as 360^o; therefore, different designs can be envisaged accordingly to more general target applications.

A price is paid by adopting this kind of elbow geometry. The actual motion of the forearm with respect to the arm is a roto-translation since the instantaneous center of rotation is moving. In fact, accordingly to figure 4, the forearm pivot point P_f is moved by the linkage $\mathbf{l}$ with velocity expressed by

$$\mathbf{v}_{P_f} = (\mathbf{k}_e \times \mathbf{l})\, \dot{q_e} \tag{1}$$

where $\mathbf{k}_e$ is the unit vector defining the elbow axis, and $\dot{q_e}$ the angular velocity of the linkage. The forearm has also an instantaneous center of rotation at point C. Hence given $\boldsymbol{\omega}_f$ the angular velocity of the forearm (with respect to the arm) we obtain

$$\mathbf{v}_{P_f} = \boldsymbol{\omega}_f \times \mathbf{d}_f \tag{2}$$

being $\mathbf{d}_f$ the radius $(P_f - C)$.

By equating the two equations above and noting that, depending on the design, $\mathbf{l} = \alpha \mathbf{d}_f$ we obtain

$$\boldsymbol{\omega}_f = \alpha\, \mathbf{k}_e\, \dot{q_e} \tag{3}$$

In the present case $\alpha = 2$,

then the forearm rotates twice faster than q_e.

2.3 *Shoulder*

The proposed shoulder design is based on a 4 joints kinematics. It features 3 coincident axes allowing a pretty large operating volume, plus a fourth roll axis providing axial rotation of the arm. In order to increase shoulder mobility the third shoulder joint has kinematics equivalent to the elbow one, which depending on the arm posture allow rotations in excess of $\pm 135^o$. Axes 2 and 4 are equivalent to wrist roll axis, while joint 1 supports the whole arm and has the role of mitigating the problems related with shoulder singularities.

3. THE CONTROL ARCHITECTURE

A significant research effort has been made in the past few years to formulate general frameworks for the control of generic classes of robotic systems, based on one hand on sound theories and on the other on standard design and engineering guidelines for their implementation.

Among these the *Task Function* control methodology provides, from an engineering point of view, the guidelines for the development of classes of control architectures [2]. Seminal work by Espiau (Samson *et al.*, 1991), extended by various authors including (Aicardi *et al.*, 1995), (M. Aicardi, 1996), has led to the idea of *open control architecture*, where hardware and software modules are designed and implemented using standard design procedures.

Two important features of *open control architectures* based on the *Task Function* approach are *modularity* and *scalability*. *Modularity* is the possibility to access all the elements of the control architecture and makes it possible to upgrade parts of the architecture if required without affecting the whole system. *Scalability* refers to the possibility of integrating within the same control architecture several independent subsystems at different times.

The major goal of the present work is to extend the concept of *open control architectures* to a fully embedded and distributed control system.

3.1 *Task Function Based Architectures*

A *Task Function* is a measure of the completion of a given robot operation. In general it can be considered as an implicit function of the robot generalized coordinates and time, (Aicardi *et al.*,

[2] We intend by *control architecture* the control hardware, software and algorithms as a whole.

Fig. 6. The AMADEUS underwater arm requires over 150 wires to be integrated with the control architecture

1995), and in a more general framework also robot generalized velocities (Samson *et al.*, 1991).

A Task Function based control architecture can be modelled as two nested control loops: an inner dynamic control loop with the goal of tracking joint level velocity signals $\dot{\bar{\mathbf{q}}}$ computed by an external task level control loop with the goal of zeroing a specific task error $\mathbf{e}$ (Aicardi *et al.*, 1995), (Angeletti *et al.*, 1998), (M. Aicardi, 1996). A generic task level control scheme for a multi-robot system is depicted in figure 5

4. EMBEDDED ROBOT CONTROLLERS

The Task Function approach discussed above allows to define models of control architectures for generic robotic systems. There are however other issues which challenge the study and development of new control architectures in particular for small scale manipulators.

Integrating the robot and the controller is a major issue when developing small and possibly redundant manipulators. As a matter of fact for the actuation of each joint may be required up to 20 wires (including ground shields); the situation gets even worse if it is expected to sensorize the device, (4 to 6 wires may be required for joint torque sensors and wrist force/torque sensors; order of N lines may required for distributed skin-like sensors with N^2 tactile elements, etc).

As an example the robot shown in figure 6, developed during the European Commission Project AMADEUS Phase II, (Lane *et al.*, 1998), has seven degrees of freedom and is equipped with brushless motors and resolvers at each joint and include a force/torque sensor at the wrist. There are more than 150 wires entering the robot, and figure 7 shows in detail the lines coming from the wrist.

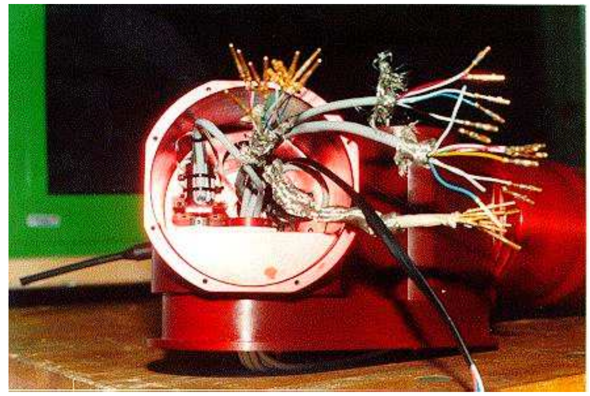

Fig. 7. Detail of the wires arriving at the wrist of the AMADEUS arm

The solution proposed here is to move the electronics (power amplifiers, data and signal processing, etc.) as close as possible to the peripherals (sensors, actuators, etc.), i.e. to embed part of the control architecture, into the robot. In particular the idea is to fully implement the *Inner Loop Control* modules as a network of distributed control and signal processing nodes connected by a field bus to a supervisor node implementing the interface with the *Outer Loop Control* modules.

4.1 *Distributed Inner Loop Controller*

The control architecture consists of a set of control nodes connected through a field bus (f-bus) based network. Each node of the network is dedicated to control a given group of joints or other peripherals (e.g. sensors, special purpose actuators, etc.). Physically adjacent nodes can in turn be connected each other through additional duplex serial links. The first node plays the role of host of the network and operates as a control supervisor. The supervisor is the interface with the *Outer Loop* control computers, and synchronizes the operations of the various nodes.

It is assumed in the current implementation that each control node is able to close a local control loop without feedback from the other nodes. This implies that the *Inner Loop* controller is decoupled. The stability and robustness characteristics of a decoupled *Inner Loop* controller have been discussed in (Aicardi *et al.*, 1995).

In figure 8 a generic embedded architecture is sketched. Each node consists of a computing module (e.g. a micro-controller, DSP, etc.), a suitable f-bus interface, and (custom) electronics to drive the peripherals (including high speed serial links).

Each node, with the exception of the supervisor, is located as close as possible to the associated peripherals. This solution leads to a distributed and embedded control system, which allows reducing

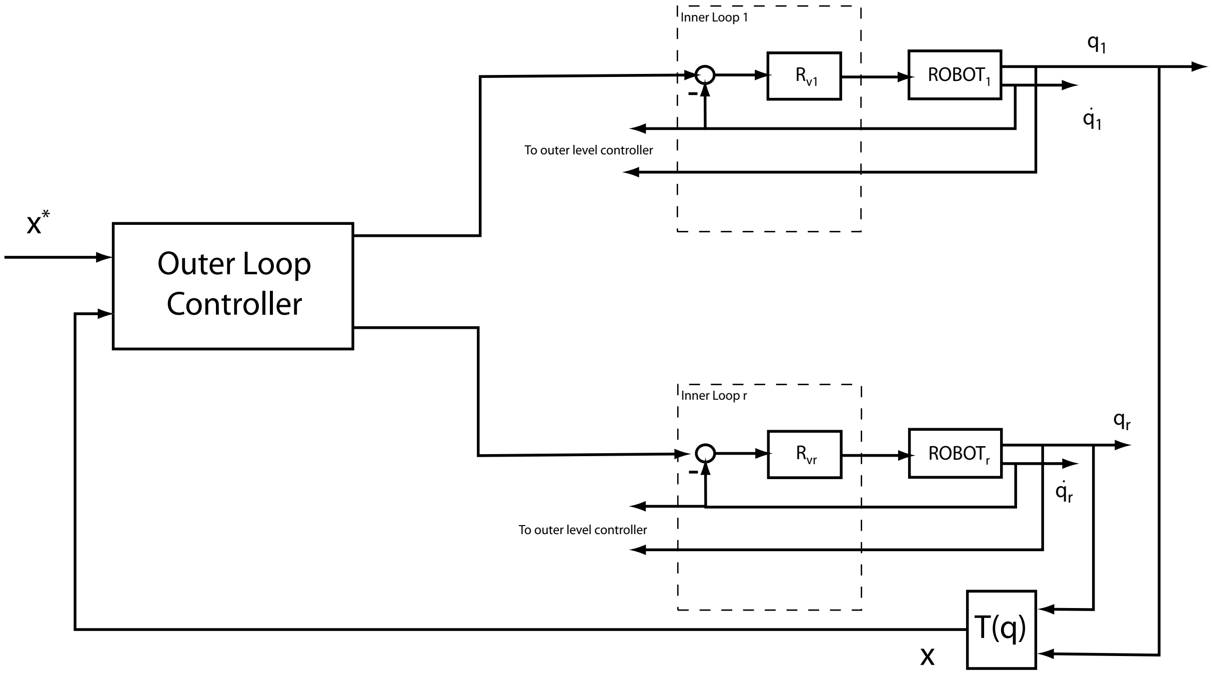

Fig. 5. Task Function based control architectures are highly modular and can be easily scaled

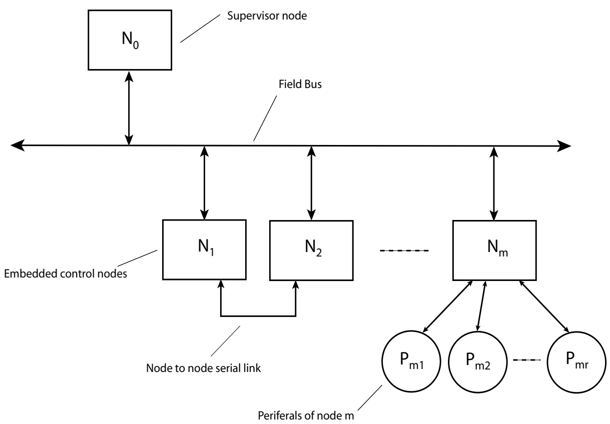

Fig. 8. Example of a network of distributed nodes

significantly the number of cables running through the arm.

Therefore, the hardware architecture proposed is highly modular: specific functions are implemented by dedicated modules/devices, which can be suitably upgraded if required with almost no impact to the other modules; the architecture is also easily scalable: a new function can be added to the system by adding a new node to the network (provided that f-bus bandwidth limits are fulfilled).

4.2 *Current Implementation*

4.2.1. *Node Structure* Each node in the present implementation has been designed to control a single actuator, with the exception of the yaw and pitch wrist axes (axes 7 and 8) which are controlled by a single node. Each control node consists of a microcontroller based computer (INFINEON C167 @ 20MHz), mounted in piggyback fashion on a custom electronic interface module which provide interface with peripherals/actuators.

The peripherals currently managed by each node are: PWM amplifier (bounded to the motor), optical encoder, strain-gage based torque sensors, precision potentiometers. The signals managed by each node are

- PWM logic control signals (logic).
- Emergency brake control signal (logic)
- Current feedback, from PWM amplifier (analogic).
- Emergency temperature flag, from PWM amplifier (logic)
- Strain measurement and bridge excitation signals (analogic), except axes 7, and 8.
- Optical encoder differential signals (logic)
- Precision potentiometer signals (analogic), except axes 6, 7, and 8.
- 2 high speed full-duplex serial lines (logic)
- f-bus lines (logic)

Each node handles about 25 I/O lines (including shields) which are mostly routed to close devices. Each module has approximately the following dimensions $9 \times 5 \times 2$ cm^3, and requires a $5V$ supply. Motors operate at $32V$, except wrist motors which operate at $12V$. Three distinct power supply busses have been implemented.

4.2.2. *Bus Structure* The f-bus technology adopted is based on the CAN 2.0B (CiA-Committee, 2001). There are several advantages that make at the present time CAN appropriate for robotic applications:

- CAN is becoming a *de facto* standard for automotive industry. Several manufacturers have developed low cost components implementing CAN. In particular several micro-controller families (including INFINEON C16x) support built-in CAN interfaces.
- Physical implementation of the bus is based on a two wire cable.
- CAN is a very reliable protocol which embeds mechanisms for detecting errors both at message level (CRC and Frame Check), and at bit level.

More detailed information about CAN can be found in (CiA-Committee, 2001).

4.2.3. *Software Control Architecture* To avoid excessive overhead in communications and to eliminate risks of conflicts for bus access, the whole architecture is synchronized by the supervisor node. At the beginning of each *Outer Loop* sampling period the supervisor polls each node with a message containing control values for that period and possibly other control parameters; it then waits for a reply message containing feedback data, and diagnostics reports. Reply of each node is immediate since it is served by a high priority interrupt routine. This 'postman' model allows avoiding concurrent access of the nodes to the bus. However since CAN allows conflict resolution mechanisms based on message priorities, it is possible to envisage mechanisms to manage possible emergencies based on asynchronous transmission of specific high priority messages.

5. CONCLUSIONS

In this paper we have described some aspects related with the design of a human-like robot arm, with fully embedded control architecture.

From the mechanical point of view the arm kinematics has been selected in order to emulate particular classes of motions including complete forearm opposition to the arm, and singularity free wrist motions.

From the point of view of the control architecture design, the proposed architecture belongs to the class of *Task Function* based control systems and therefore is consistent with a now well established theory. In particular the proposed control architecture is distributed over a field bus network and embedded into the robot arm. This solution looks very promising in order to develop small scale hyper-sensorized robot arms designed to safely interact and cooperate with humans.

6. REFERENCES

Aicardi, M., A. Caiti, G. Cannata and G. Casalino (1995). Stability and robustness analysis of a two layered hierarchical architecture for the closed loop control of robots in the operational space. *1995 IEEE Robotics and Automation Conf., Nagoya, Japan.*

Angeletti, D., G. Cannata and G. Casalino (1998). The control architecture of the amadeus gripper. *International Journal of Systems Science* **29(5)**, 485–496.

Caffaz, A. and G. Cannata (1998). The design and development of the dist-hand dextrous gripper. *1998 IEEE ICRA, Leuven, Belgium* pp. 2075–2080.

Casalino, G., G. Cannata, G. Panin and A. Caffaz (2001). On a two level hierarchical structure for the dynamic control of multifingered manipulation. *2001 IEEE ICRA, Seoul, Corea* pp. 77–84.

CiA-Committee (2001). Can in automation. http://www.can-cia.de, 2001.

Lane, D., G. Cannata, G. Casalino, G. Bartolini and B. Davies (1998). Advanced manipulator for deep underwater sampling: The amadeus research project. *1998 IEEE Int. Conf. On Control Applications, Trieste, Italy* pp. 1068–1073.

Leeser, K., J. P. Donoghue and William T. Townsend (1994). Computer-assisted teach and play: Novel user-friendly robot teach mode using gravity compensation and backdrivability. *Proc. Fifth World Conference on Robotics Research, Cambridge, MA, USA.*

M. Aicardi, G. Cannata, G. Casalino (1996). Task space robot control: Convergence analysis and gravity compensation via integral feedback: Convergence analysis and gravity compensation via integral feedback. *35th IEEE CDC Conference, Kobe, Japan.*

Rosheim, M. (1989). *Robot Wrist Actuators.* Wiley. New York.

Samson, C., B. Espiau and C. Le Borgne (1991). *Robot Control: The Task Function Approach.* Oxford University Press. Oxford, England.