EPM/RT-88/26

ON THE CONTROL OF A ROBOTIC MANIPULATOR

By

Romano M. DeSantis, Professor

Département de Génie Electrique

Ecole Polytechnique de Montréal

Septembre 1988

gratuit

Dépôt légal, 4e trimestre 1988
Bibliothèque nationale du Québec
Bibliothèque nationale du Canada

Pour ce procurer une copie de ce document, s'adresser au:

Editions de l'Ecole Polytechnique de Montréal
Ecole Polytechnique de Montréal
Case postale 6079, Succursale A
Montréal (Québec) H3C 3A7
(514) 340-4000

Compter 0,10$ par page (arrondir au dollar le plus près) et ajouter 3,00$ (Canada) pour la couverture, les frais de poste et la manutention. Régler en dollars canadiens par chèque ou mandat-poste au nom de l'Ecole Polytechnique de Montréal. Nous n'honorerons que les commandes accompagnées d'un paiement, sauf s'il y a eu entente préalable dans le cas d'établissements d'enseignement, de sociétés ou d'organismes canandiens.

PREFACE

The following notes are a partial documentation in support of the eight week course on "Identificazione e Controllo dei Manipolatori Robotici" offered at the Politecnico di Milano, Dipartimento di Elettronica, during the spring 1988. This course is itself a part of "Teoria della Regolazione" a full year course under the responsibility of Professor A. Locatelli. A debt of gratitude on the part of the author is due to Professor A. Locatelli for the opportunity to collaborate with him in this project, and to professor S. Bittanti for making my coming to Milano possible. Many thanks are also due to them for technical interactions, fine hospitality and warm friendship; these thanks must be extended to the other members of the group of Automatica, among them: professors G. Guardabassi, N. Schiavoni, C. Maffezzoni, P. Bolzern, P. Colaneri, and R. Scattolini; and to the various students with whom I have had the pleasure to interact.

Our objective is to introduce the main ideas and techniques associated with the control of a robotic manipulator. The interdisciplinary nature of the subject, the width and depth of its territory, its partly well established and partly rapidly evolving character make this not an easy thing to do. We will therefore opt for a pragmatic approach: to simply aim at opening a window, creating a point of departure, providing the motivation and the capability for the reader to proceed by himself as far down the road as he may wish to go.

We will assume familiarity with a certain number of items such as basic kinematics, dynamics and automatic control theory. Our presentation strategy will be to recall the basic idea behind a technique and to illustrate its application to the case in point; care will be taken to pinpoint connections with the specialized literature; a special attention will be given to provide the student with the background required to benefit from a number of proposed hands-on simulation experiments.

These were the intentions... As it turns out, we do realize that what we end up with is only the beginning of the document we would have liked to get. We are nevertheless confident that the serious student complementing what we have with the references listed in the bibliography (particularly recommended are [Wr.1] and [Fu.1], plus the educational reports [RMDS. 3-5]), should have no difficulty into attaining the intended objective.

ON THE CONTROL OF A ROBOTIC MANIPULATOR

LIST OF FIGURES

1. INTRODUCTION

1.1 Introduction

The physical components of the guidance unit of a robotic system, a robotic manipulator (fig. 1.1) complemented with whatever peripheral equipment may be required to perform a given task (fig. 1.2), are similar to those which are usually found in the supervision unit of an industrial control process: transducers, detectors, actuators, power generators, computers, communication, memory, visualization, interface, software, timing etc.

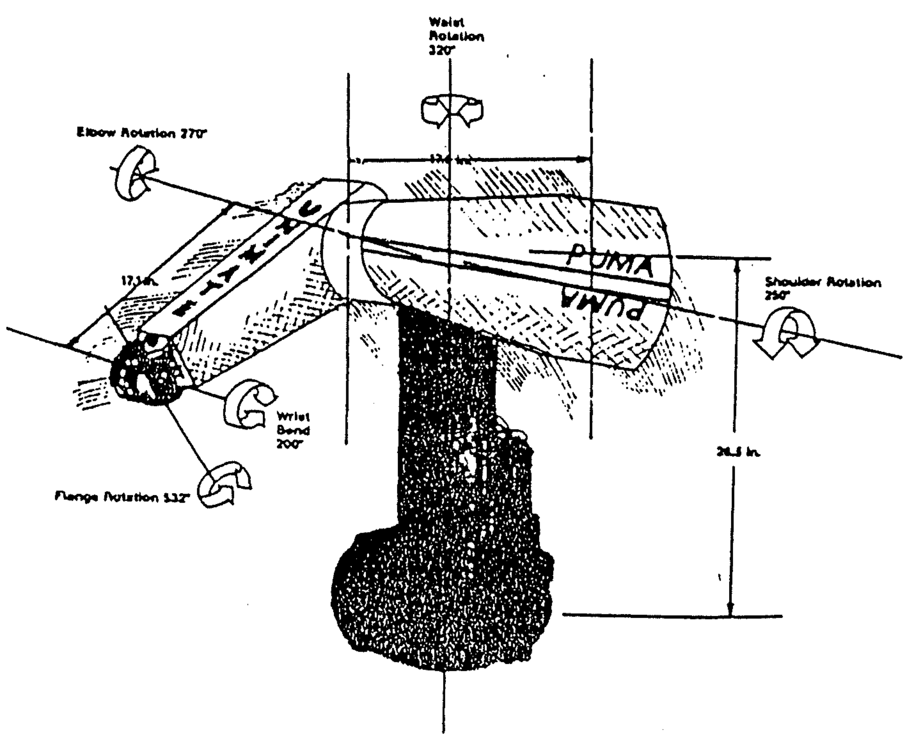


Figure 1.1: A robotic manipulator

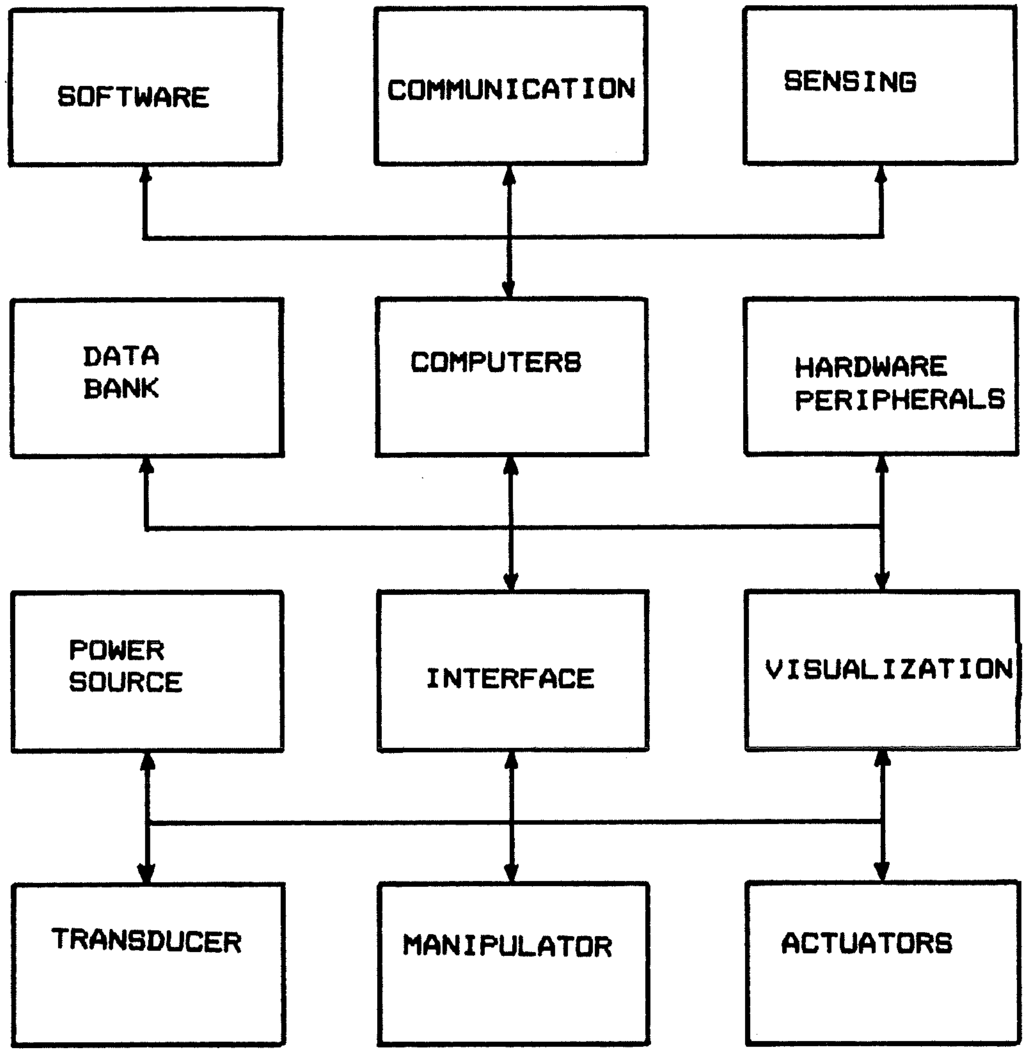


Figure 1.2: Physical components of a robotic system

With reference to fig. 1.3, the conceptual operation of such a system can be described as follows:

i. an operator communicates through a man-machine interface the task to be accomplished;

ii. an artificial intelligence module determines the action strategy needed to accomplish such a task; this strategy is usually a function of the task itself and of the information about the work scene which is provided by the sensing elements;

iii. a trajectory generator module computes the manipulator links position and velocity required to implement such an action;

iv. a controller computes the inputs to the amplifiers driving the actuators so that position and velocity measured by the links transducer coincide as rapidly, precisely and reliably as possible with the required values;

v. this sequence is repeated until the task has been accomplished.

Our purpose in what follows is to focus attention on ideas and techniques at the basis of the design of the controller module. Preliminary requirements to the intended subject are: an exposure to the principles governing the kinematics and the dynamics of articulated chains [Cr.1] ; a general knowledge of the various families of robots available in the industry together with the variety of applications [En.1] ; a good familiarization with the working and the modalities of operation of at least one specific example of a robotic guidance system (a useful educational example can be found in [RMDS.1]); a good familiarization with standard control techniques.

The main questions of interest are: How are standard control techniques applied to robotic manipulators? With what success? With what modifications? What new techniques are considered? What is the interplay between kinematics, dynamics and control?

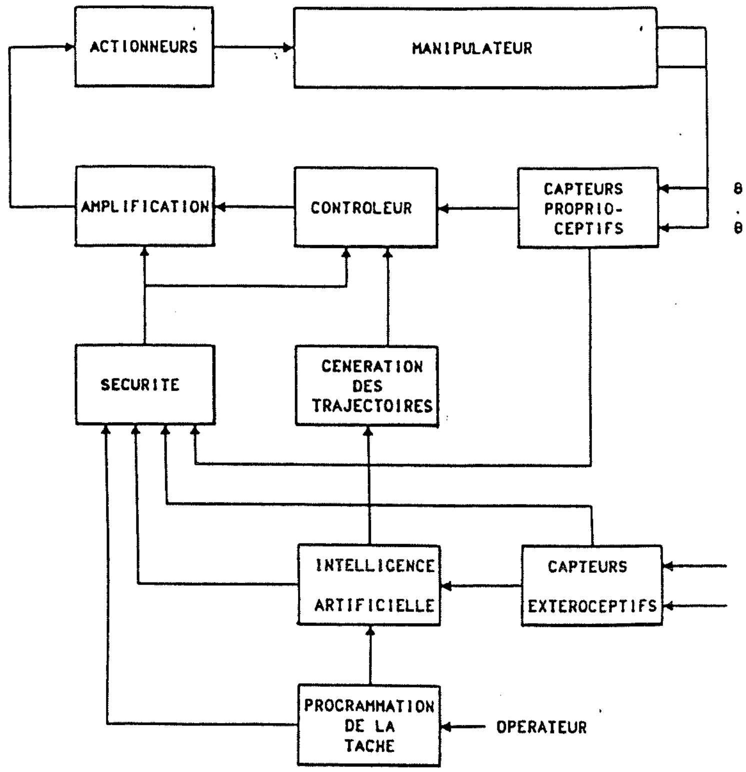


Figure 1.3: The conceptual structure of a robotic system.

1.2 Dynamic model of a robotic manipulator

The dynamic model of a robotic manipulator allows one to determine acceleration speed and position of the robot which are produced by the application of a given set of forces (direct dynamics); or, inversely, to determine the set of forces required to produce a desired acceleration speed and position trajectory (inverse dynamics). Such a model is usually obtained by using either the Newton-Euler approach, particularly efficient for computational purposes, or the Euler-Lagrange approach, particularly convenient for an analytical closed form representation of the physical elements involved.

A systematic application of the Euler-Lagrange approach [Pa.1,chp 4] leads to the following model (fig.2.1):

$$\Gamma_i = \sum_{j=1}^{n} D_{ij}\, \ddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} D_{ijk}\, \dot{q}_j \dot{q}_k + D_i, \quad i=1...n$$

where:

n : is the number of joints (degrees of freedom, links);

Γ_i: is the resultant of the forces, (moments), applied to link i;

$q_i, \dot{q}_i, \ddot{q}_i$: represent position, speed and acceleration of link i with respect to link i-1;

D_{ij}: represents the effective inertia of the manipulator

$$D_{ij} = \sum_{p=\max\, i,j}^{n} \text{Trace}\,(U_{pj}\, J_p\, U'_{pj})$$

J_p: pseudo inertia matrix of the link;

U_{pj}: $= \frac{\partial}{\partial q_j}\, T\,[p,o]$;

$T[p,o]$: transformation matrix associated to the frames of link p and of the base of the manipulator;

D_{ijk} : centripetal and Coriolis force coefficients;

$$D_{ijk} = \sum_{p=\max\, i,j,k}^{n} \text{Trace}\,(U_{pjk}\, J_p\, U'_{pi})$$

$$U_{pjk} = \frac{\partial}{\partial q_i}\left[\frac{\partial}{\partial q_k} T[p,o]\right]$$

The complexity of such a model may be illustrated by simply noticing that for n=6, matrices D_{ij} and D_{ijk} have dimensions respectively equal to 6x6=36 and 6x6x6=216; the number of multiplications required to compute the model is roughly equal to

$$16\times6^4=86\times\frac{5}{12}\times6^3+\frac{171}{4}\times6^2+\frac{53}{3}\times6-128=66.271;$$

the number of additions is

$$25\times6^4+22\times6^3+\frac{129}{2}\times6^2+14\times6-96=51.548.$$

Using standard state and control notations

$$x_1 \triangleq \begin{bmatrix} q_i \\ \vdots \\ q_n \end{bmatrix} \quad ; \quad x_2 \triangleq \dot{x}_1; \qquad x \triangleq \begin{bmatrix} x_1 \\ \text{----} \\ x_n \end{bmatrix} \quad ; \quad u \triangleq \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_n \end{bmatrix}$$

one obtains a state model of the form

$$\dot{x} = f(x) + B(x)u + \beta$$

where

$$f(x) \triangleq \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -[D_{ij}]^{-1}\;[D_{ijk}] \end{bmatrix} x_1 \bullet x_2 + [D_{ij}]^{-1}\,[D_i]$$

$$B(x) \triangleq \begin{bmatrix} 0 \\ [D_{ij}]^{-1} \end{bmatrix}$$

and β is a vector representing perturbation forces (friction, stiction, external perturbations, ...).

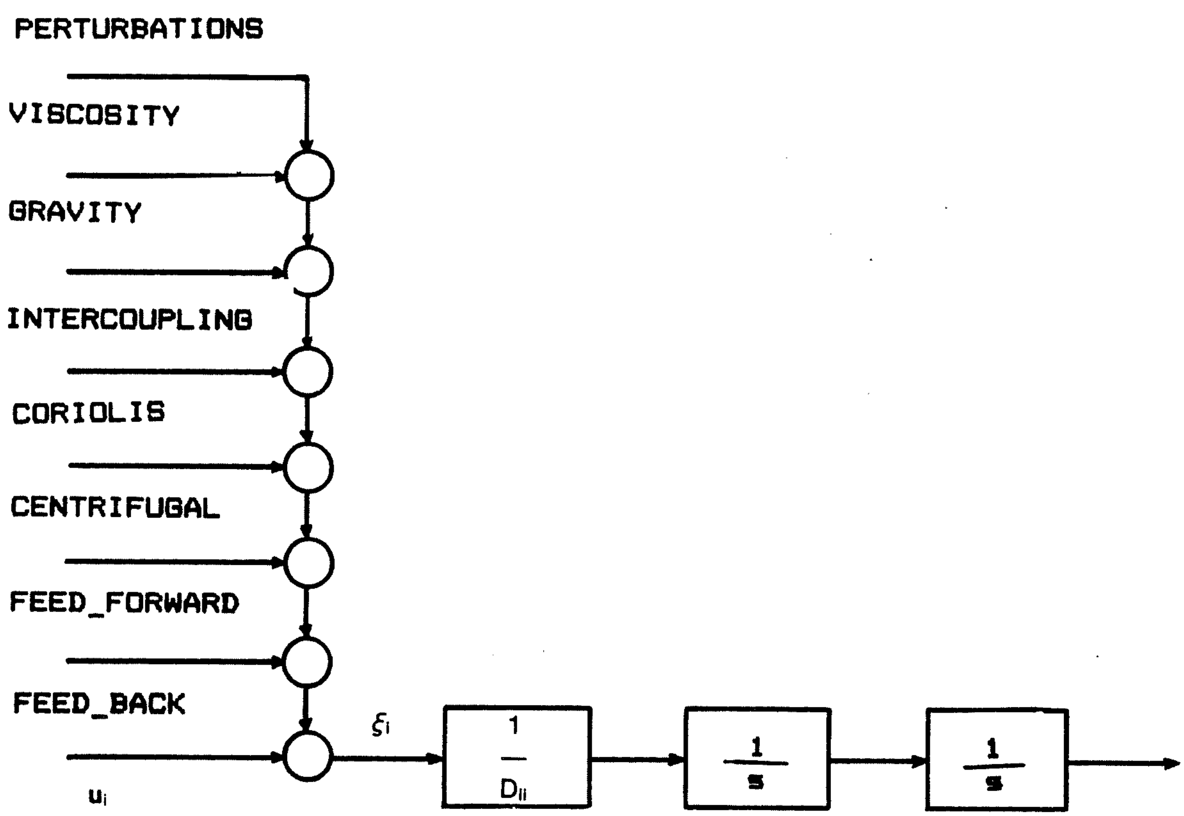


Figure 2.1: Dynamic model of a link.

Usually, the control force applied to a link is made of a centralized feedforward component, u_{FFi}, directed at neutralizing dynamic and perturbation forces (inter-coupling, gravity, centrifugal, Coulomb, ...); plus a de-centralized feedback component, u_{FBi}, directed at reducing the distance between actual and desired link position and velocity.

Indicating with x_{1i} and x_{2i} such position and velocity, one can then represent the dynamics of a link in terms of the following simplified model (fig. 2.2):

$$\dot{x}_{1i} = x_{2i}$$

$$\dot{x}_{2i} = -\frac{x_{2i}}{\tau_i} + \frac{K_{mi}}{\tau_i}\left[u_{FBi} + \xi_i\right]$$

where:

$K_{mi} \triangleq \frac{1}{{}_{ii}} \triangleq$ gain of the link, $\alpha_{ii} \triangleq$ a viscosity coefficient;

$\tau_i \triangleq \frac{\alpha_{ii}}{D_{ii}} \triangleq$ time constant of the link;

$\xi_i \triangleq$ the resultant of u_{FFi} and the dynamic and perturbation forces it is supposed to neutralize.

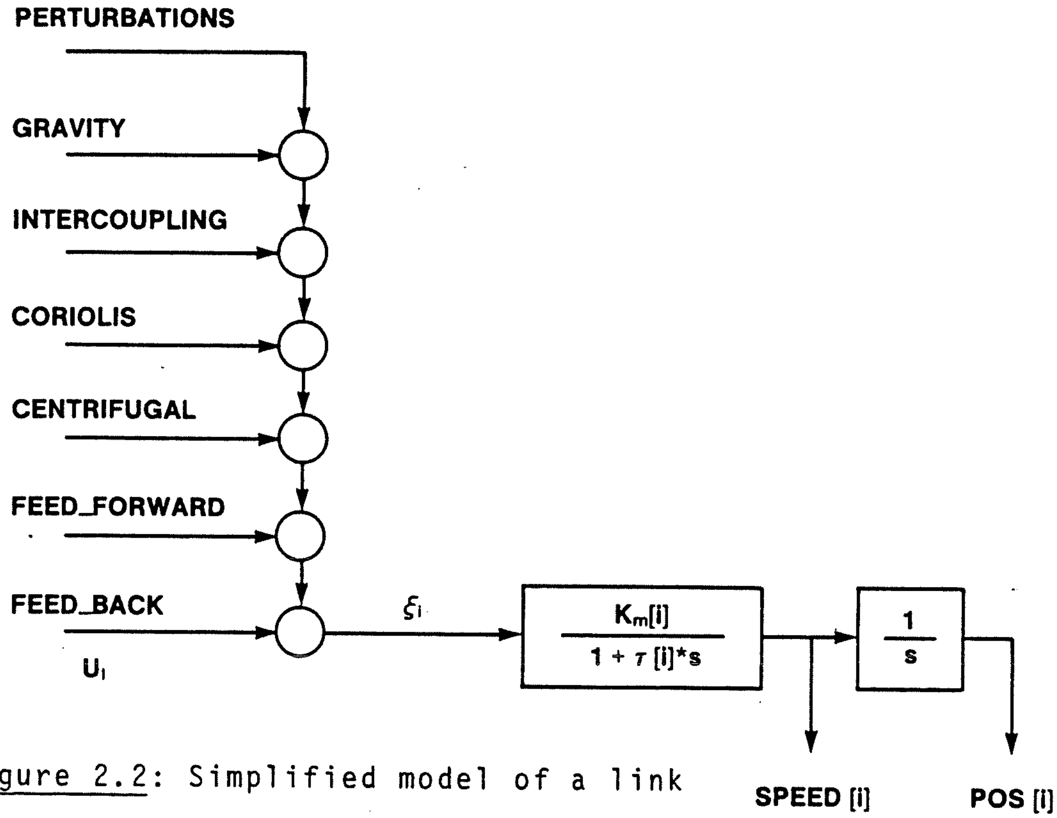


Figure 2.2: Simplified model of a link

1.3 The control problem

The structure of the controller of a robotic manipulator is represented in fig. 3.1: each link is servoed by means of a decentralized feedback loop controller; these controllers are complemented with a centralized coordinator which, on the basis of current and desired configurations of the manipulator, generates a convenient feedforward action; this coordinator may also tune the gains of the feedback loop controllers. The physical components of the feedback controller are described by fig. 3.2 and 3.3.

Representing the manipulator with the model

$$\dot{x} = f(x) + B(x)u + \beta$$

the controller design problem is: to develop a control $u(\cdot)$ capable of forcing $x(\cdot)$ to follow a desired $x_D(\cdot)$ as

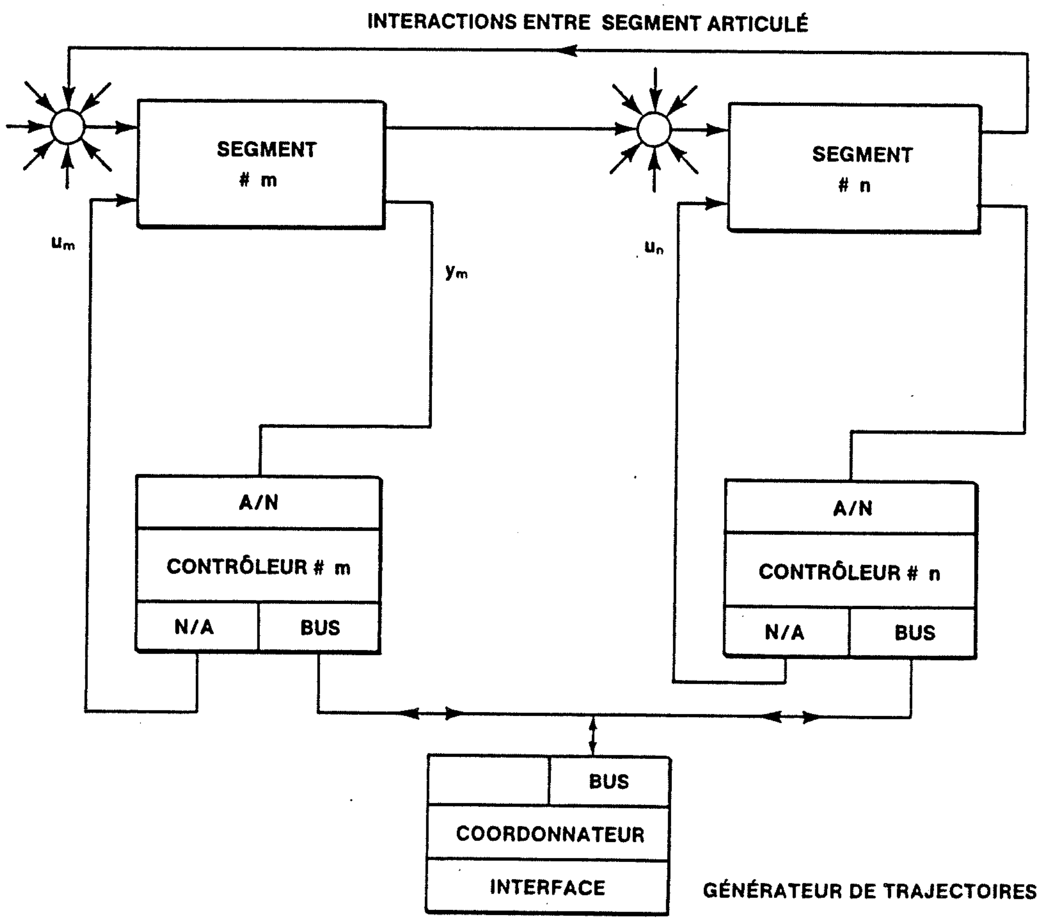


Figure 3.1: Structure of the overall controller

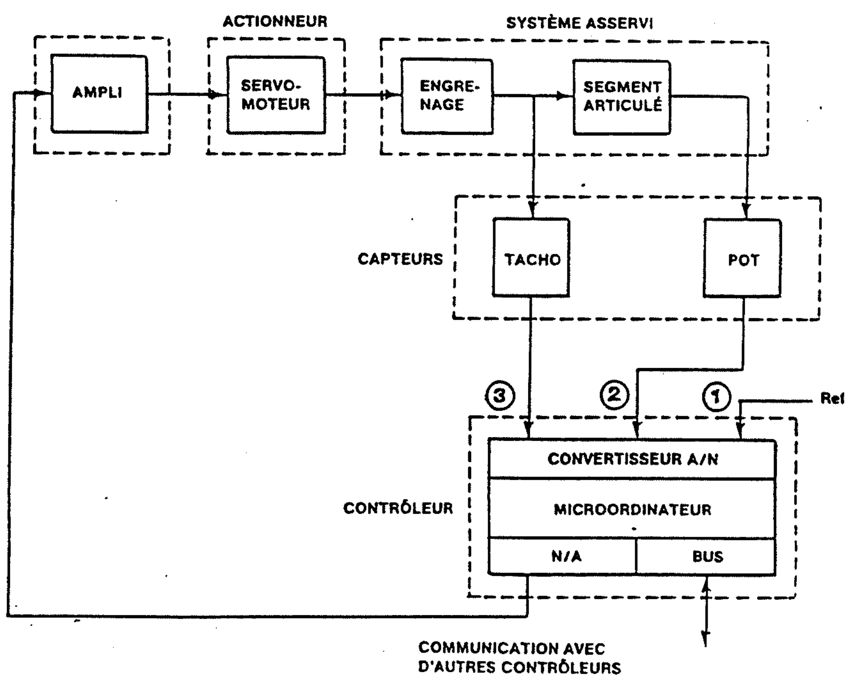


Figure 3.2: Structure of a link feedback controller

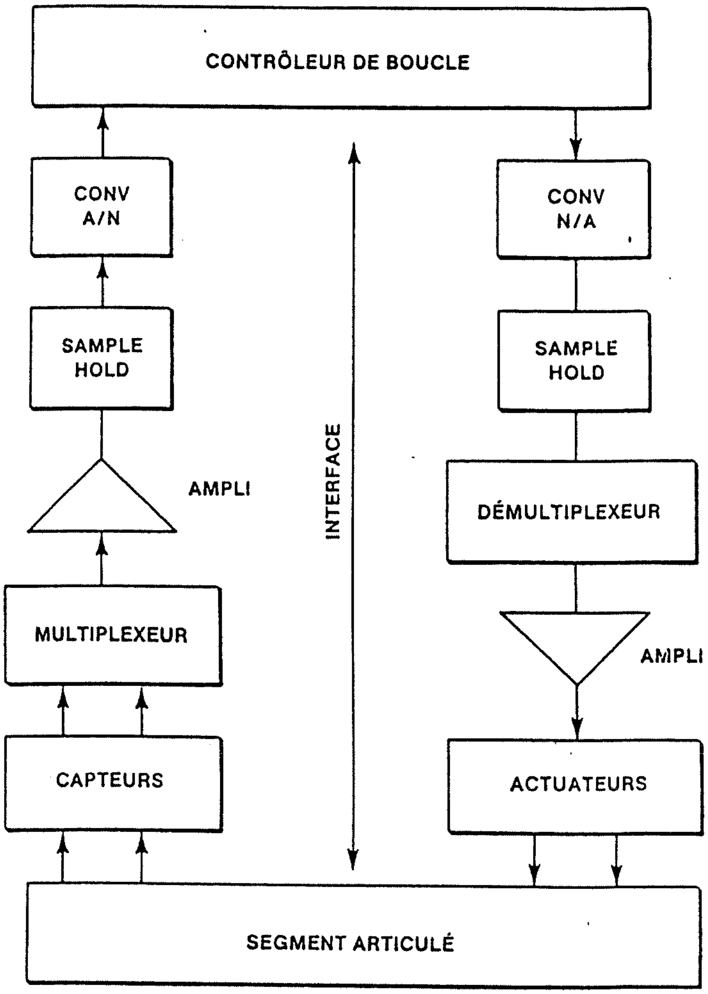


Figure 3.3: Components of a link feedback controller

rapidly, precisely and reliably as possible in spite of the presence of the usually unknown β. Such a problem has been the focus of much attention on the part of control theoreticians and most of the available standard control techniques turn out to be relevant to its solution.
One can mention in particular:

- linearization and classical theory (frequency, transfer function and static space techniques);
- Lyapunov and hyperstability theory;
- optimal control (LQG, maximum principle, Hamilton-Jacobi, etc.);
- adaptive control, identification;
- variable structure systems, etc.

In what follows we will discuss how these techniques are indeed applied to the control of robotic manipulators. In particular: the concept of computed torque and acceleration resolved controllers; the design of currently employed PID controllers; sliding mode controllers; adaptive and learning controllers.

2. MODEL BASED CONTROLLERS

2.1 Introduction

Model based controllers rely on the assumption that the dynamical model of the manipulator is perfectly known together with its parameters values and external perturbations [Le.1, p. 188-195]. Their potential adoption usually also presumes the availability of high performance actuators, transducers and computational units.

In spite of these demanding assumptions, these controllers are interesting in that they offer an ideal vehicle to put into perspective the physical, structural and computational complexity of the problem; they represent an efficient term of comparison against which to discuss the technical features of other controllers; they suggest many of the ideas and techniques which subsequently find at least a partial application in a variety of other more realistic control schemes.

2.2 Computed torque controller

Considering the manipulator dynamics as described in terms of the Lagrange-Euler model,

$$u_i = \sum_{j=1}^{n} D_{ij} \ddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} D_{ijk} \dot{q}_j \dot{q}_k + D_i,$$

the idea of a computed torque controller is to provide a control given by

$$u_i = u_{FFi} + u_{FBi}$$

with

$$u_{FFi} = \sum_{i=1}^{n} D_{ijc}\, \ddot{q}_{jD} + \sum_{j=1}^{n} \sum_{k=1}^{n} D_{ijkc}\, \dot{q}_{jm}\, \dot{q}_{km} + D_{ic}$$

and

$$u_{FBi} = \sum_{i=1}^{n} D_{ijc} \left(K_{1i} \left[q_{iD} - q_{im} \right] - K_{2i} \left[\dot{q}_{iD} - \dot{q}_{im} \right] \right),$$

where the subscripts m,c and D are used to denote measured, computed and desired values.

Under the assumption that measured, computed and actual values all coincide, that is

$$q_i = q_{im} \qquad \dot{q}_i = \dot{q}_{im}$$

and

$$D_{ij} = D_{ijc} \qquad D_{ijk} = D_{ijkc} \qquad D_i = D_{ic},$$

usage of such a control leads to

$$\sum_{j=1}^{n} D_{ij} \left[(\ddot{q}_{iD} - \ddot{q}_i) + K_{1i} (q_{iD} - q_i) + K_{2i} (\dot{q}_{iD} - \dot{q}_i) \right] = 0$$

and, from the nonsingularity of $\left[D_{ij} \right]$,

$$\ddot{q}_{iD} - \ddot{q}_i + K_{1i}(q_{iD} - q_i) + K_{2i}(\dot{q}_{iD} - \dot{q}_i) = 0$$

With a convenient choice of gain values K_{1i}, K_{2i}, one can then come up with a perfectly decoupled, arbitrarily poles assignable, second order link dynamics. Introducing the notation

$$h(q,\dot{q}) \triangleq \left[h_i(q,\dot{q})\right] \triangleq \left[\sum_{j=1}^{n} \sum_{k=1}^{n} D_{ijk}\, \dot{q}_j \dot{q}_k + D_i\right],$$

one can represent the structure of a computed torque controller using the diagram in fig. 2.1. The numerical algorithm employed can be interpreted as an inverse dynamic problem, i.e.: compute the forces required to have an acceleration given by

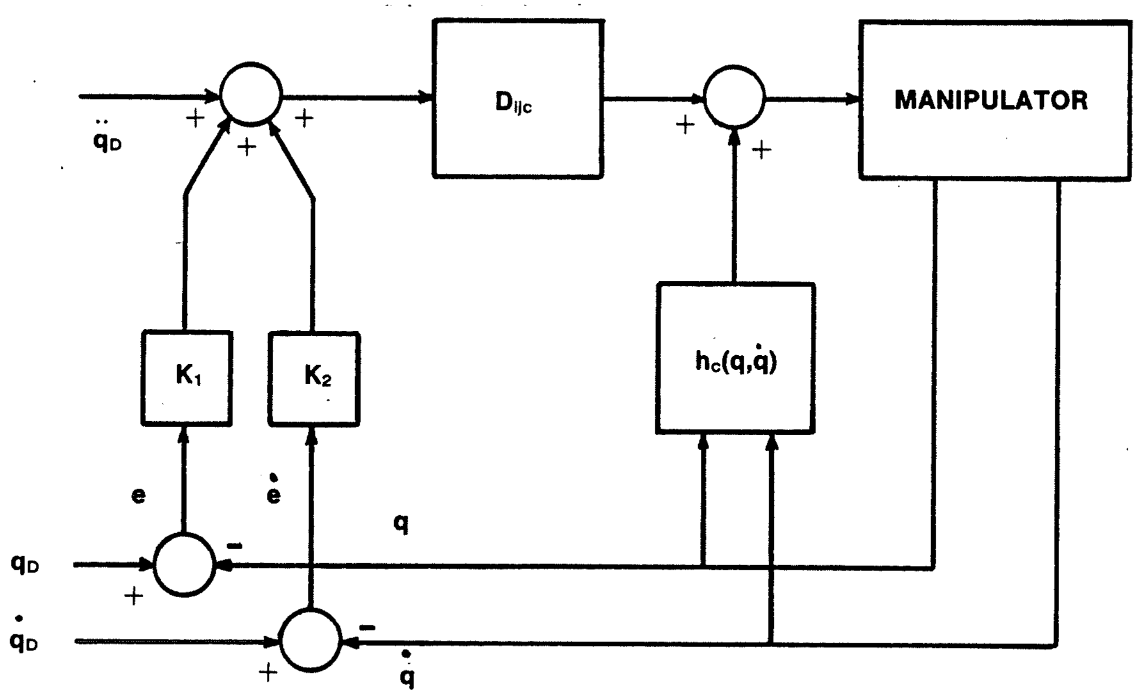


Figure 2.1: Structure of a computed torque controller

$$\ddot{q}^{*}=\ddot{q}_{D}+K_{1}(q_{D}-q)+K_{2}(\dot{q}_{D}-\dot{q})$$

in correspondence to speed and position $\dot{q}$, and q.
An effective way to implement such an algorithm is provided by the Lu-Walker-Paul approach [Fu.1, p.. 117].

2.3 Acceleration resolved motion

An acceleration resolved motion controller is similar to a computed torque controller in that it strives at applying a force capable of generating an acceleration equal to the desired acceleration plus a PD component of the position error. The distinction is that these elements are now directly expressed in work space coordinates. Accordingly, one assumes to have available desired workspace position and orientation trajectories $(p_D(t), \Omega_D(t))$ of the manipulator end effector, together with their linear and angular speed and acceleration denoted, respectively, with the symbols $v_D(t)$, $\omega_D(t)$, and $\dot{v}_D(t), \dot{\omega}_D(t)$. Usually the vectors p_D, v_D and $\dot{v}_D$ refer to desired position speed and acceleration of the origin of the end effector frame with respect to the work space frame; Ω_D, ω_D and $\dot{\omega}_D$ refer to desired actual first and second derivative values of the angles (often Euler or RPY angles) characterizing the orientation of the end effector frame.

Control forces are then applied so as to generate end effector linear and angular accelerations given by the expression

$$\begin{bmatrix} \dot{v}^* \\ \dot{\omega}^* \end{bmatrix} = \begin{bmatrix} \dot{v}_D \\ \dot{\omega}_D \end{bmatrix} + K_1 \begin{bmatrix} p_D - p \\ \Omega_D - \Omega \end{bmatrix} + K_2 \begin{bmatrix} v_D - v \\ \omega_D - \omega \end{bmatrix}$$

where p, Ω, v and ω characterize actual position and velocity values. Under ideal conditions this leads to an error dynamics governed by the equation

$$\begin{bmatrix} \dot{v}_D - \dot{v} \\ \dot{\omega}_D - \dot{\omega} \end{bmatrix} + K_1 \begin{bmatrix} p_D - p \\ \Omega_D - \Omega \end{bmatrix} + K_2 \begin{bmatrix} v_D - v \\ \omega_D - \omega \end{bmatrix} = 0$$

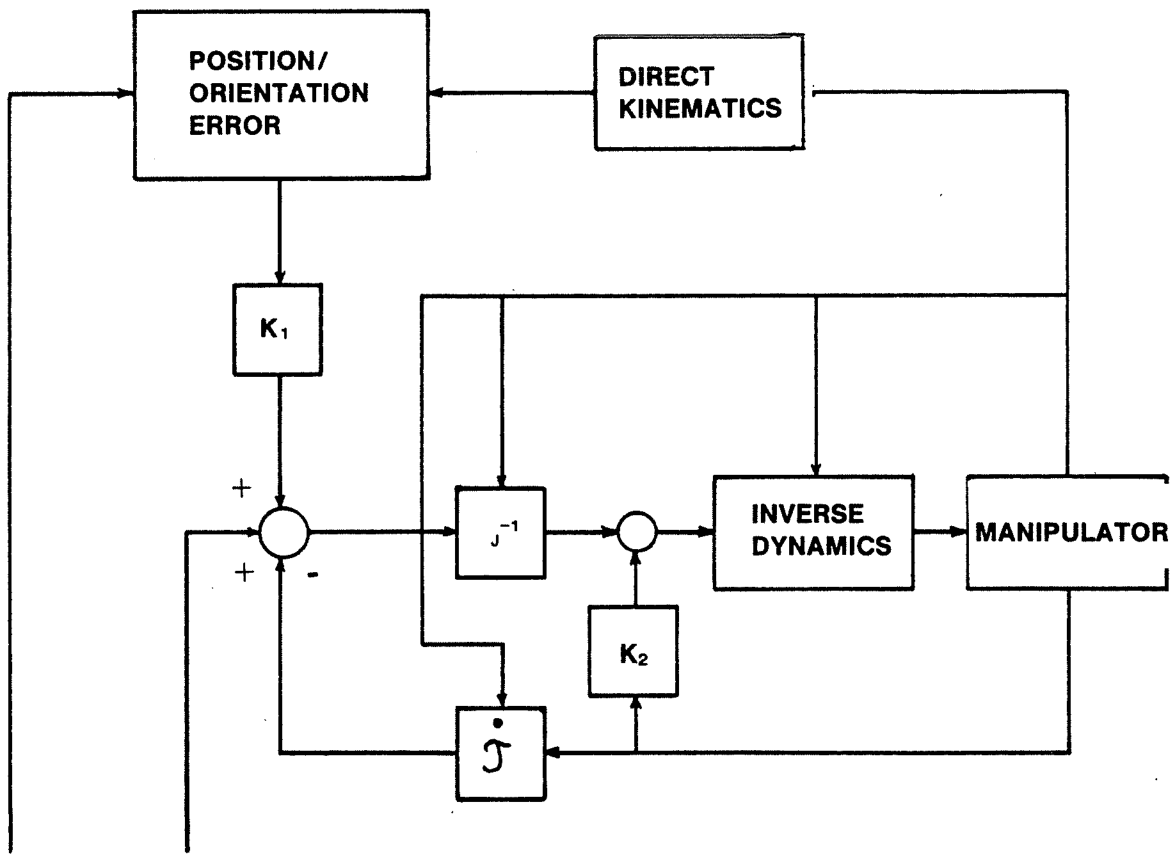
that is, (upon a choice of diagonal K_1 and K_2): a perfectly decoupled in work space coordinates, arbitrarily poles assignable dynamics.

The control forces can be computed by applying the Luh-Walker-Paul inverse dynamic algorithm [Fu.1, p.117]. For this, one needs first to have available the manipulator links actual position and velocity, plus the desired acceleration. Position and velocity q and $\dot{q}$ are usually obtained from the link transducers. Denoting with J the manipulator Jacobian matrix, the links acceleration vector $\ddot{q}$ is computed by exploiting the following incremental kinematic equations

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = J\dot{q}$$

$$\begin{bmatrix} \dot{v}(t) \\ \dot{\omega}(t) \end{bmatrix} = J\ddot{q} + \dot{J}\dot{q}$$

$$\ddot{q} = J^{-1} \begin{bmatrix} \dot{v}(t) \\ \dot{\omega}(t) \end{bmatrix} - J^{-1} \dot{J}\dot{q} \quad .$$



$$\begin{bmatrix} P_D \\ \Omega_D \end{bmatrix} \qquad \begin{bmatrix} \dot{V}_D \\ \dot{\omega}_D \end{bmatrix} + K_2 \begin{bmatrix} V_D \\ W_D \end{bmatrix}$$

Figure 3.1: Structure of an acceleration resolved motion controller

3. PID CONTROLLERS

3.1 Introduction

In current industrial practice the control force applied by the controller to the i-th link is given by

$$u_i = u_{FFi} + u_{FBi}$$

where u_{FFi} is a centralized feed-forward component aimed at compensating for gravity; u_{FBi} is a decentralized feed-back component with the function of reducing the difference $x_{iD}-x_i$, where x_{iD} and x_i represent, respectively, required and actual values of link position and velocity.

The dynamics of the link can then be thought as given by

$$\dot{x}_{1i} = x_{2i}$$

$$\dot{x}_{2i} = -\frac{1}{\tau_i} x_{2i} + \frac{K_{mi}}{\tau_i} (u_{FBi} + \xi_i)$$

where ξ_i, the residual of gravity intercoupling centripetal and perturbation forces which have not been eliminated by the presence of u_{FFi}, is viewed as a perturbation. The control u_{FBi} is usually a nonlinear (relay) or linear (P, PD, PID) function of the difference $x_{iD}-x_i$.

In general, relay controllers are the simplest and the least expensive; however, as they usually produce vibration and imprecision, their use is often confined to low cost, low performance manipulators. Higher class manipulators usually employ controllers of the PID type (fig. 2.1, 2.2).

3.2 Position Servo and Trajectory Tracking PID Controllers

The action provided by a typical position servo PID controller is given by (see fig. 2.1),

$$u_{FBi}=-K_1(x_1-x_{D1})-K_1x_2-K_3\int_0^t(x_1-x_{D1})dt+K_3K_4\ x_{1D},$$

where K_1,K_2,K_3 and K_4 represent proportional, derivative, integral and feedforward gains respectively. The reason to be of these gains is the classical one:

K_1 is indispensable for stability;

K_2 is required to improve speed of response;

K_3 is required to eliminate the influence of stationary perturbations;

K_4 compensates for the transient reponse deteriorating influence of K_3.

Under the hypothesis that K_{mi} and τ_i are constant, the dynamical behaviour of the link is described by the transfer functions

$$F_1=\frac{x_1}{x_{D1}}(s)=\frac{sK_m(K_1+K_3K_4)+K_3K_m}{\tau s^3+(1+K_mK_2)s^2+K_mK_1s+K_mK_3}$$

$$F_2=\frac{x_1}{\xi}(s)=\frac{sK_m}{\tau s^3+(1+K_mK_2)s^2+K_mK_1s+K_mK_3}$$

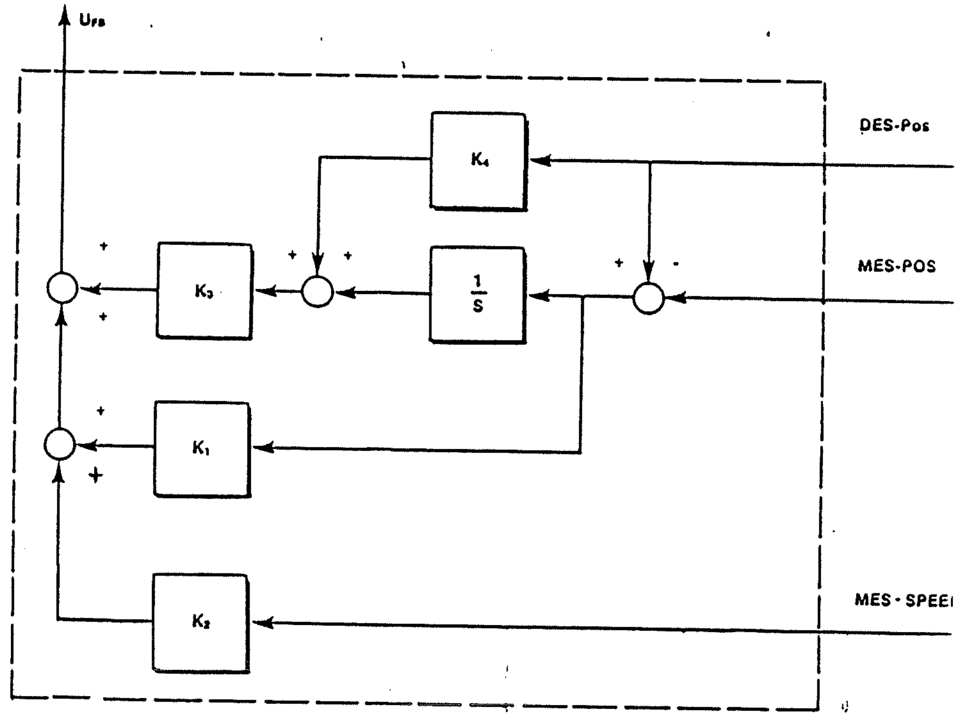


Figure 2.1: PID position servo controller

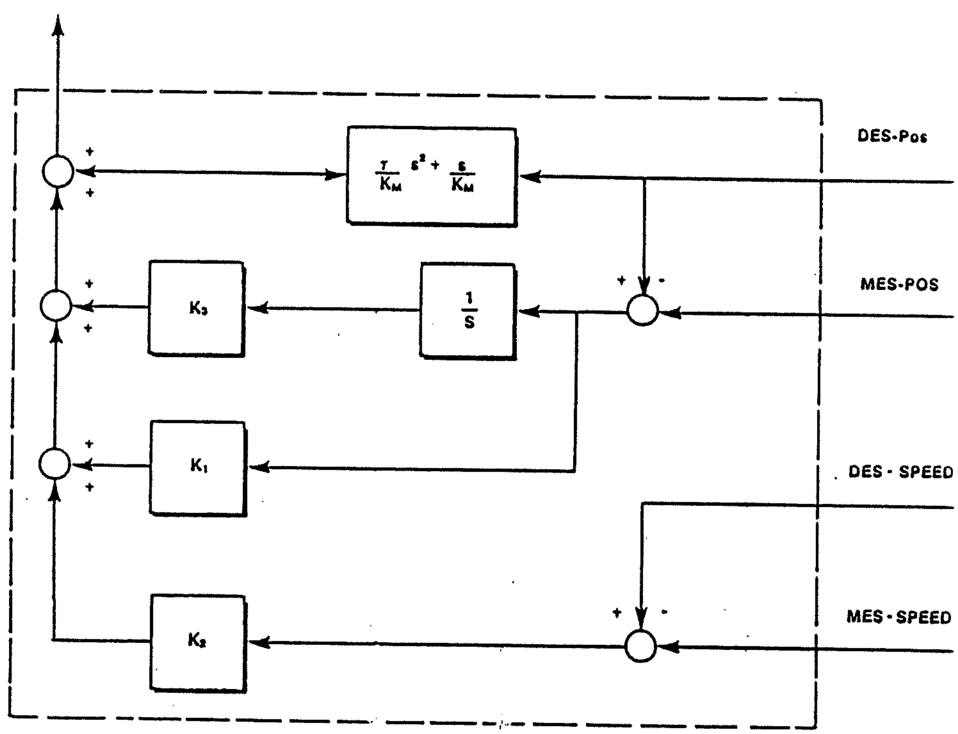


Figure 2.2: PID trajectory tracking controller

For stability one needs

$$K_3 > 0$$
$$K_2 > -1/K_m$$
$$K_1 > \tau\, K_3/(1+K_m K_2).$$

The values of $K_1,\ldots,K_4$ are usually chosen so as to attain some prescribed performance specifications often expressed in terms of response time, settling time, overshoot,control level constraints, sensitivity, robustness and similar. These specifications can usually be translated in terms of poles and zeros.

Indicating with p_1, p_2, p_3 and z the zero-pole configuration correspondent to the required specifications, and using the notations

$$a_1 \triangleq -\left[\frac{1}{p_1} + \frac{1}{p_2}\right] \qquad a_2 \triangleq \frac{1}{p_1 p_2}$$

one has

$$F_1 = \frac{(1-s/z)}{(1+q_1 s + a_2 s^2)\ (1-s/p_3)}$$

$$F_2 = \frac{s}{K_3(1+a_1 s+a_2 s^2)\ (1-s/p_3)}$$

and

$$K_1 = \frac{\tau}{K_m}\ (p_1 p_2 + p_2 p_3 + p_1 p_3)$$

$$K_2 = -\left[(p_1+p_2+p_3)\tau - 1\right] / K_m$$

$$K_3 = -\frac{\tau}{K_m} p_1 p_2 p_3$$

$$K_4 = (-K_3 - K_1 z) / K_3 z$$

If, in addition to x_{D1}, one also has available $\dot{x}_{D1}$ and $\ddot{x}_{D1}$, the PID position servo controller may be replaced by a trajectory tracking PID controller whose action is described by

$$u_{FBi} = \frac{\tau}{K_m}\ddot{x}_{D1} + \frac{x_2}{K_m} - K_1(x_1 - x_{D1}) - K_2(x_2 - \dot{x}_{D1})$$

$$- K_3 \int_0^t (x_1 - x_{D1})dt.$$

Using the notations

$$\Delta x_1 \triangleq (x_1 - x_{D1})$$

$$\Delta x_2 \triangleq (x_2 - \dot{x}_{D1})$$

$$\Delta x_3 \triangleq \int \Delta x_1 \, dt$$

one has

$$\Delta\dot{x}_1 = \Delta x_2$$

$$\Delta\dot{x}_2 = \frac{K_m}{\tau}\left[-K_1\Delta x_1 - K_2\Delta x_2 - K_3\Delta x_3 + \xi\right]$$

$$\Delta \dot{x}_3 = \Delta x_1$$

By requiring that the dynamics of the system be once again characterized by the poles p_1, p_2 and p_3 one has

$$F_1 \triangleq \frac{x_1}{x_{D1}} = 1$$

$$F_2 \triangleq \frac{\Delta x_1}{\xi} = \frac{K_m}{\tau} \frac{s}{(s-p_1)(s-p_2)(s-p_3)}$$

The values and stability requirements of K_1, K_2 and K_3 are identical to the case of the position servo. A more complete discussion of PID controllers may be found in [RMDS.8, Le.1 p. 5-25] . A detailed development concerning a 2 degrees of freedom SCARA manipulator under relay, dual mode, (linear in the proximity of the target, non linear when far away), and PID control may be found in [RMDS.4]. A hands-on analysis of the dynamical behavior of such a system may be developed through the use of the simulators POSSPEED and SIMNOG available at the Computer Center, (Prometeus), (see appendices A,B).

3.3 Observations

The dependence of the link parameters K_{mi} and τ_i on the manipulator configuration may lead to a situation where a given set of gains is satisfactory in one configuration and unsatisfactory in others. This shortcoming represents the main justification for the introduction of an adaptive strategy whereby the values of $K_1,\ldots,K_4$ are continuously adjusted to the estimated values of K_{mi} and τ_i (see forthcoming section).

The dynamical intercoupling generated by the simultaneous displacement of more than one link may lead to ξ values which may not be adequately neutralized by the PID controller. This problem may be solved by introducing a dynamical decoupling in the controller feedforward component; an alternative solution may be provided by a PID/sliding mode controller (see relevant section).

Finally, in high accuracy high speed manipulator centripetal and Coriolis effects may further deteriorate performance. This type of manipulators may require full computed torque controllers complemented with nonlinear multivariable adaptive capabilities. The usage of a learning control may also be required (see relevant sections).

4. ADAPTIVE CONTROLLERS

4.1 Introduction

The dynamic performance of a robotic manipulator equipped with a controller provided by either the classical PID or the more advanced model based approach is dependent upon the knowledge of the plant parameter values. While such a knowledge is in general incomplete, it may often be improved during the operation of the system by monitoring the input/output behavior of the plant. It is quite natural then to consider an adaptive scheme whereby plant parameter knowledge is sequentially updated and exploited to improve the controller action.

Early robotic implementations of such a scheme have given rise to the utilization of classical model reference and self-tuning adaptive controllers [Le.1].
While these controllers have been mainly considered with a view of rendering adaptive the decentralized feedback part of the controller, more recently, new advanced controllers capable of extending adaptivity to the feedforward controller component have been proposed [Cr.1].

4.2 Model reference adaptive controllers (MRAC)

The feedback system characterizing an adaptive controller with a reference model is made of a plant, a classical controller, a reference model and a controller gain adaptation module (fig. 2.1). Its principle of operation is the following: the reference model is chosen so as to generate the desired output trajectory; the controller gains are initially adjusted so that under nominal operating conditions the plant output coincides with the reference model output; when the plant parameters vary, plant and model outputs no longer coincide and a difference is detected; the adaptation module exploits this difference so as to generate a variation in the controller gains having the effect of reducing it.

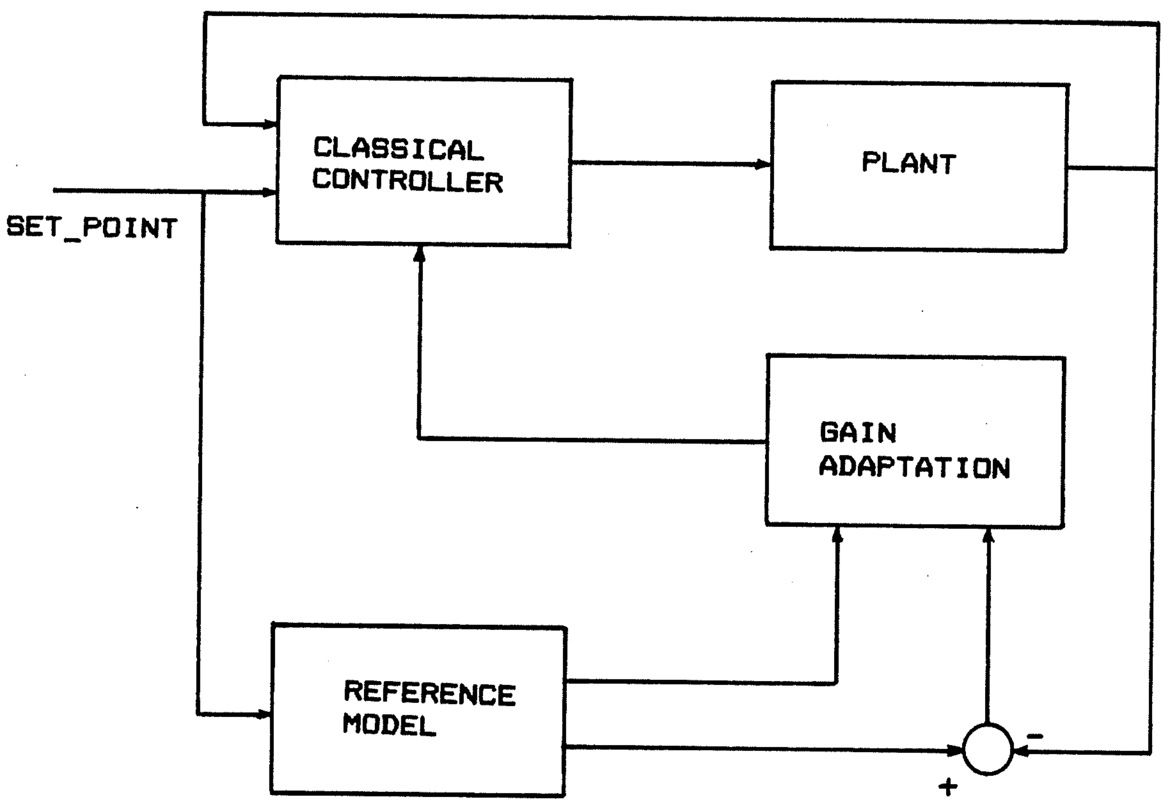


Figure 2.1: Structure of a model reference adaptive controller

To illustrate the implementation of this idea into a computational algorithm, the dynamics of the plant together with the classical controller is modelled by

$$\dot{\underline{x}} = \underline{f}_s(\underline{x}, \underline{\alpha}, \underline{r}, \underline{d})$$

where

$\underline{f}_s$: is a function representing the dynamic behavior of the feedback system;

$\underline{x}$: is the state;

$\underline{\alpha}$: is a variable parameter;

$\underline{r}$: the system entry;

$\underline{d}$: reflects the presence of external disturbances.

The reference model dynamics corresponds to the feedback system desired behavior and is given by

$$\dot{\underline{y}} = \underline{f}_m (\underline{y}, \underline{a}, \underline{r})$$

where:

$\underline{f}_m$: represents the dynamics of the reference model;

$\underline{y}$: is the desired state;

$\underline{a}$: is the nominal value of $\underline{\alpha}$.

The parameter vector α is a function of the controller gains as well as of a certain number of plant parameters subject to unpredictable variations. The idea behind the MRAC is to compensate disturbances and unpredictable variations with adaptively induced gain variations. This is done by attempting to minimize the output error

$$e(t) = y(t) - x(t)$$

As a measure of this error one considers a quadratic function of the type

$$V(e) = \frac{1}{2} e'Qe$$

where Q is a conveniently chosen positive definite symmetric matrix.

An effective procedure to minimize V(e) may be obtained by applying the gradient method. This approach leads to

$$\dot{\alpha} = -\gamma \frac{\partial V}{\partial \alpha} \cong -\gamma \frac{\partial e'}{\partial \alpha} \quad Qe$$

where $\frac{\partial e}{\partial \alpha}$ represents the plant error sensitivity matrix, while γ is a conveniently chosen acceleration factor matrix. The computation of $\frac{\partial e}{\partial \alpha}$ is difficult to implement because α is unknown. This matrix is then approximated by considering $\partial y/\partial a$, the sensitivity matrix of the reference model. Formally one has

$$\frac{\partial e}{\partial \alpha} = \frac{\partial (y-x)}{\partial \alpha} = - \frac{\partial x}{\partial \alpha} \cong - \frac{\partial y}{\partial a}$$

The matrix $\frac{\partial y}{\partial a}$ is computed by using the sensitivity model

$$\frac{\partial \dot{y}}{\partial a} = \frac{\partial f_m(y,a,r)}{\partial y} \frac{\partial y}{\partial a} + \frac{\partial f_m}{\partial a} \quad (y,a,r).$$

A detailed educational implementation of this algorithm to a robotic link may be found in [RMDS.5] . A simulator MRAC, available at the Computer Center, (Prometeus), allows one to gain a hands-on experience with the working of such a system.

4.3 Self-Tuning adaptive controllers

The conceptual operation of a self-tuning adaptive controller is as follows, (see fig. 3.1): an estimator module gives a recursive plant parameter estimate which is the most compatible with the available plant input/output behavior; an adaptation module updates the controller parameters on the basis of such an estimate; the controller operates as a standard classical controller.

In a typical application of this scheme the plant is modelled in terms of a linear difference equation of the type

$$y(m+1)=a_o\ u(m)+a_1\ u(m-1)+\ldots+a_p\ u(m-p)$$

$$-\ b_o\ y(m)-\ \ldots\ -b_n\ y(m-n)$$

where $u(\cdot)$, $y(\cdot)$, a_i, b_j represent respectively, input, output and unknown parameters of the plant.

The plant parameter estimator is usually of the recursive least square type [As.1]. Using the notation

$$\hat{\theta}(m) \triangleq \begin{bmatrix} a_o \\ \vdots \\ a_p \\ b_o \\ \vdots \\ b_n \end{bmatrix} (m)$$

to indicate the parameter estimate at time m, and with

$$x(m) \triangleq \begin{bmatrix} u(m) \\ \vdots \\ u(m-p) \\ -y(m) \\ \vdots \\ -y(m-n) \end{bmatrix}$$

the input/output data vector relevant to the computation of y(m+1), a typical recursive algorithm computes $\hat{\theta}(m+1)$ from $\hat{\theta}(m)$, y(m+1), x(m) according to the following algorithm:

$$\hat{\theta}(m+1) = \hat{\theta}(m) + K(m) \left[y(m+1) - x'(m)\hat{\theta}(m) \right]$$

$$\gamma(m) = 1 / \left[1 + x'(m)P(m)x(m) \right]$$

$$K(m) = \gamma(m)\ P(m)\ x(m)$$

$$P(m+1) \triangleq \left[P(m) - K(m)x'(m)P(m) \right] \frac{1}{\lambda}$$

Where:

$$P(m) \triangleq \left[X'(m)\ X(m) \right]^{-1}$$

$$X'(m) \triangleq \begin{bmatrix} x'(1) \\ ------- \\ x'(m-1) \end{bmatrix}$$

and λ is a forgetting factor. This algorithm gives the casual parameter estimate $\hat{\theta}(m+1)$ which minimizes the expression

$$J \triangleq \sum_{i=1}^{m} \varepsilon^2(i)\ \lambda^{m-i}$$

$$\varepsilon(i) \triangleq \left[y(i+1) - x'(i)\hat{\theta}(i) \right].$$

The classical controller employs a strategy which is usually of the pole placement, minimum variance or linear optimal regulator type. Its gains are computed at each instant m by the gain update module. The algorithm adopted by this module presumes the plant parameters to be time invariant and equal to $\hat{\theta}(m)$ (<u>certainty equivalent principle</u>).

A detailed application of this algorithm to a robotic link is described in [RMDS.7]. A simulator, SELF-TUNE, available at the Computer Center, (Sala Prometeus), allows one to gain a hands on experience with the working of such a system.

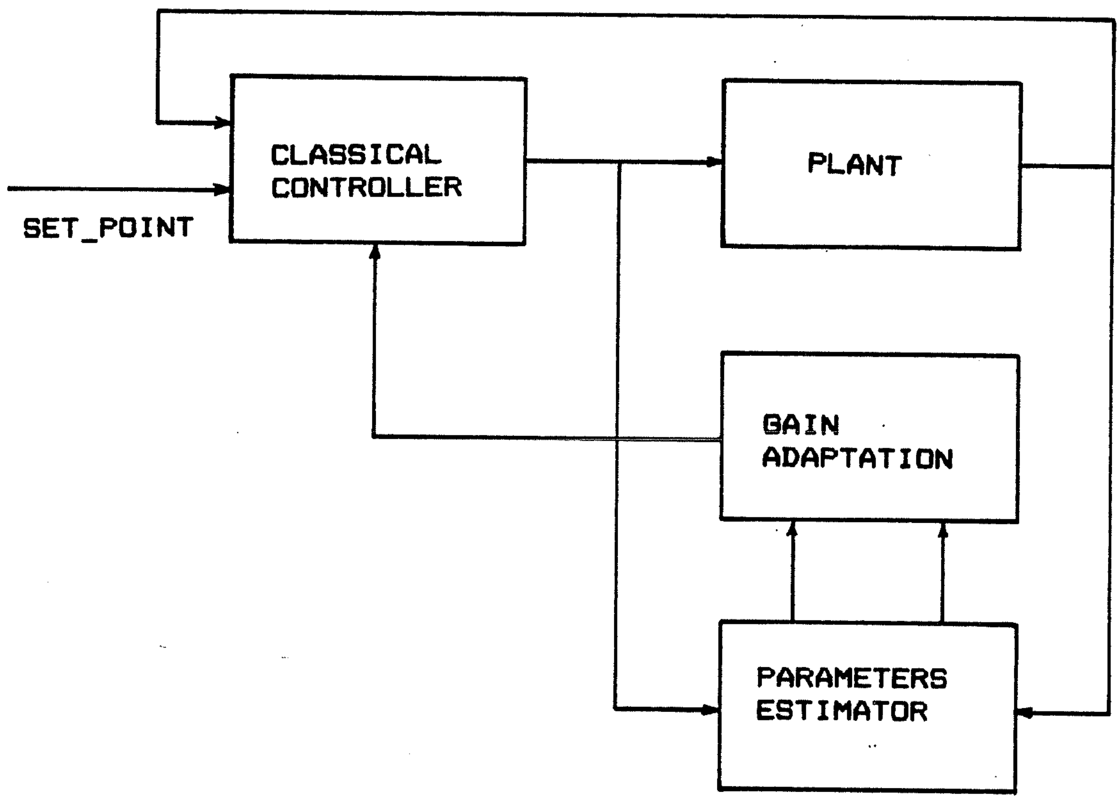


Figure 3.1: Structure of a self-tuning adaptive controller

4.4 Multivariable nonlinear adaptive controllers

The potential of the model reference and self-tuning approaches is somewhat limited by the usual assumption that the plant is linear and made of decoupled scalar systems. More seriously: the adaptivity of these schemes is usually confined to the controller feedback component. Recently, multivariable non-linear schemes have been developed which take into specific account the particularities of the manipulator dynamics [Cr.2]. A major step forward is in that the adaptivity now concerns both the controller feed-forward and feedback components. Too: identification and control are no longer independent.

The main ideas at the basis of this new development can perhaps be best illustrated by discussing its implementation in a scheme where a precomputed torque controller is used. Such a scheme is represented in fig. 4.1, where $\hat{D}$ and $\hat{h}(q,\dot{q})$ represent respectively the estimated values of the manipulator inertia matrix $[D_{ij}]$ and of the vector

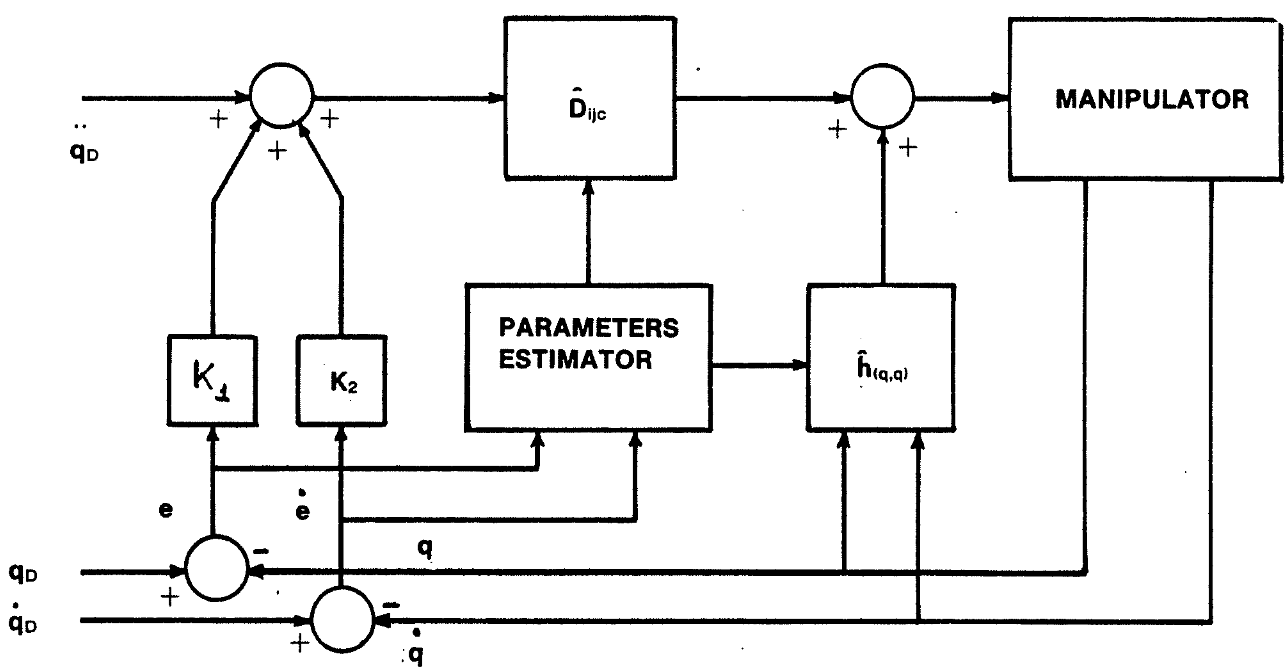


Figure 4.1: Multivariable nonlinear adaptive control of a robotic manipulator

$$h(q,\dot{q}) = \left[h_i(q,\dot{q})\right] \triangleq \left[\sum_{j=1}^{n}\sum_{k=1}^{n} D_{ijk}\,\dot{q}_j\dot{q}_k + D_i\right]$$

The unknown parameters of the manipulator are described in terms of a vector P. This vector is chosen so as to have the representation

$$D\,\ddot{q} = \tilde{D}\,\ddot{q} + W_1\,[q,\ddot{q}]\,P$$

$$h(q,\dot{q}) = \tilde{h}(q,\dot{q}) + W_2\,[q,\dot{q}]\,P$$

where $\tilde{D}$ and $\tilde{h}$ are (known) parts of $D\,\ddot{q}$ and $h(q,\dot{q})$ which do not depend on P; W_1 and W_2 are known functions of $q,\dot{q}$ and $\ddot{q}$. Using the notation $\hat{P}$ to denote the current estimate of P and $W \triangleq [W_1 \,\vdots\, W_2]$, the estimation algorithm is based upon the equation

$$\dot{\hat{P}} = KW'\,\hat{D}^{-1}\,(e+\psi\dot{e})$$

where K and ψ are conveniently chosen real positive diagonal matrices and e denotes the error between the desired and the actual manipulator links position

$$e \triangleq q_D - q.$$

An outline of the justification of such an algorithm goes as follows [Cr.2, p.51] :

i. Usage of the computed torque approach suggests a control action given by

$$\Gamma = \hat{D}(q)\,\ddot{q}^* + \hat{h}(q,\dot{q})$$

where

$$\ddot{q}^* = \ddot{q}_D + K_1 e + K_2 \dot{e}$$

and K_1, K_2 are diagonal matrices.
This, in turn, implies

$$\hat{D}\left[\ddot{q}_D + K_1 e + K_2 \dot{e}\right] = D\ddot{q} + h - \hat{h}$$

$$\hat{D}\left[\ddot{e} + K_1 e + K_2 \dot{e}\right] = \left[D - \hat{D}\right]\ddot{q} + h - \hat{h}$$

$$\ddot{e} + K_1 e + K_2 \dot{e} = \hat{D}^{-1} W(q, \dot{q}, \ddot{q})\Phi$$

where

$$\Phi \triangleq P - \hat{P}.$$

Using the notation

$$u \triangleq \hat{D}^{-1} W\Phi$$

it follows

$$\ddot{e} + K_1 e + K_2 \dot{e} = u$$

ii. Considering

$$e_1 \triangleq e + \psi\dot{e},$$

using Laplace notations one has

$$e_1(s) = \left[I + \psi s\right]\left[s^2 I + sK_2 + K_1\right]^{-1} u.$$

The choice of diagonal ψ is made so that the mapping $u \to e_1$ is strictly positive real.

From a well known strictly positive real result, this implies that the mapping $u \to e_1$ is representable in terms of a dynamical system

$$\dot{x} = Ax + Bu$$
$$e_1 = Cx$$

with positive real symmetric matrices Q_1, Q_2 such that

$$A'Q_2 + Q_2A = -Q_1$$
$$Q_2B = C'.$$

iii. One may now consider a Lyapunov function of the type

$$V(x,\Phi) \triangleq xQ_2x + \Phi' K^{-1}\Phi$$

with K a diagonal positive real matrix. Observing that the chosen estimation strategy implies

$$\dot{\hat{P}} = KW' \hat{D}^{-1} e_1$$

hence

$$\dot{\Phi} = -KW' \hat{D}^{-1} e_1$$

one has

$$\begin{bmatrix} \dot{x} \\ \dot{\Phi} \end{bmatrix} = \begin{bmatrix} A & B\hat{D}^{-1}W \\ -KW'\hat{D}^{-1}C & 0 \end{bmatrix} \begin{bmatrix} x \\ \Phi \end{bmatrix}$$

By computing

$$\dot{V} = \frac{\partial V}{\partial x} \dot{x} + \frac{\partial V}{\partial \Phi} \dot{\Phi}$$

one obtains

$$\dot{V} = -x'Q_1x$$

which implies $x \to 0$, hence $e_1 \to 0$ and $e \to 0$.

<u>iv</u>. It is further shown in [Cr.2, p.58-63] that:

- this procedure is robust with respect to external perturbations;
- under the (persistently exciting equivalent) condition

$$\int_{t_o}^{t_o+\rho} W'W \, dt \geq \alpha I$$

  for $\rho>0$ and some $\alpha>0$, one also has $|\hat{P}-P| \to 0$;
- if one knows that the parameters values lie in a certain interval

$$l_i \leq p_i \leq h_i$$

  then one may advantageously modify the algorithm with the variant

$$\hat{p}_i(t^+) = l_i \qquad \text{if} \qquad \hat{p}_i(t) \leq l_i - \delta$$

$$\hat{p}_i(t^+) = h_i \qquad \text{if} \qquad \hat{p}_i(t) \geq h_i + \delta$$

  for some $\delta>0$.

5. SLIDING MODE CONTROLLERS

5.1 Introduction

Even with the utilization of an adaptive controller, it may still be difficult to guarantee a satisfactory performance in high accuracy high speed manipulators where fast parameter variations and high level nonstationary perturbations may be expected. Sliding mode controllers, with a behavior which can potentially be made solely dependent on the controller and not on the plant parameters, present a natural addition to the arsenal of available tools. As these controllers are not usually covered in a classical automatic control course, in what follows we will give a brief introduction of the concept together with the subsequent presentation of some selected results.

To illustrate the concept, consider a robotic link described by [Ut.1, p. 15-17]

$$\dot{x}_1 = x_2 \qquad\qquad a \triangleq \frac{1}{\tau}$$

$$\dot{x}_2 = -a\, x_2 + b(u+\xi) \qquad\qquad b \triangleq Km/\tau$$

where, as usual,

$x_1 \triangleq$ position of the link ;

$x_2 \triangleq \dot{x}_1$;

$u \triangleq$ controller action ;

τ, $K_m \triangleq$ time constant and static gain of the link;

$\xi \triangleq$ input equivalent perturbation generated by external unknown forces and parameter variations.

Consider the (sliding mode) function

$$slm \triangleq x_2 + c\, x_1$$

together with the control strategy

$$u = -\psi x_1 + \gamma$$

where

$$\psi \triangleq \alpha \quad \text{if } x_1\, slm > 0,$$

$$\psi \triangleq \beta \quad \text{if } x_1\, slm < 0,$$

and

$$\gamma = -M\, \mathrm{SIGN}(slm)$$

One can easily verify that if

$$\alpha > \max \left\{ \frac{1}{b}(a-c)c \ , \ \frac{a^2}{4b} \right\}$$

$$\beta < \frac{1}{b}(a-c)c \quad , \quad M > |\xi|$$

then slm(t) tends to zero. With the (sliding mode) condition

$$slm(t) = 0 \qquad\qquad t \geqq t_0$$

satisfied, one has

$$x_1(t) = e^{-c(t-t_o)} x_1(t_o).$$

This implies that as long as the controller is able to impose the sliding mode condition, the link state trajectory depends on the controller parameter c and is independent from the plant parameter values τ, K_m, ξ.

This example suggests that a robotic controller might be designed by adopting a strategy whereby the desired behavior of the manipulator is first translated in terms of a (sliding mode) condition of the type

$$Cx = 0$$

where x is the state of the manipulator and C is a suitable matrix. A sliding mode control would then be subsequently determined which is capable of satisfying this condition. Expected advantages being in the simplicity of the ensuing controller, a good dynamic and static behavior, a strong robustness to parameter variations and external perturbations. The same type of strategy may be also applied to state estimation and parameter identification; extensions involve adaptive model reference and self-tuning controllers.

5.2 Some Selected Results

Given a dynamical system [Ut.1, p.80]

$$\dot{x} = f(x,t) + B(x,t)u$$

with a controller defined by

$$u_i = \begin{cases} u_i^+(x,t) & \text{for } s_i(x)>0 \quad , \quad i=1\ldots m \\ u_i^-(x,t) & \text{for } s_i(x)<0 \end{cases}$$

where u_i^+, u_i^- and s_i are certain continuous functions, a sliding mode controller is defined by following two conditions:

i. no trajectory entirely generated by one of the 2^m potential control functions is contained in

$$S(x,t) \triangleq \{x \mid x \in \bigcap_i s_i(x)=0 \} \ ;$$

ii. for any $\varepsilon>0$, there exist $\Delta_0>0$ and $\delta>0$ such that: for any x_0 with a distance from $S(x,t_0)$ smaller than δ, the trajectory of

$$\dot{x}=f(x,t)+B(x,t)\tilde{u}(x,t) \quad , \quad x(t_0)=x_0$$

with $\tilde{u}(x,t)$ such that

$$\tilde{u}_i=u_i \quad \text{if} \quad \|s_i(x,t)\| > \Delta_0$$

$$\min(u_i^+,u_i^-) \le \tilde{u}_i \le \max(u_i^+,u_i^-) \text{ otherwise,}$$

has the property that

$$\sum_{i=1}^{n} s_i^2(x,t) < \varepsilon.$$

The control ideas characterizing the sliding mode controller design philosophy are well represented by the following two fundamental results.

Theorem 1. A necessary condition for a sliding mode controller to exist is that: a u_{equ}, (equivalent control), solution of

$$\dot{s} = G\left[f(x,t)+B(x,t)u_{equ}\right] = 0 ,$$

where: G is a matrix whose rows are given by the gradients with respect to x of the s_i components of $s \triangleq \{s_i\}$, exists and satisfies the inequalities

$$\min(u_i^+, u_i^-) \le u_{equ} \le \max(u_i^+, u_i^-)$$

Theorem 2. For the dynamical system to be under sliding mode control in the domain S(x,t), it is sufficient that: for all $x \in S(x,t)$ there exist a function v(s,x,t) continuously differentiable with respect to all its arguments such that in a certain region, Ω, of the subspaces $s_1, \ldots, s_m$ containing the origin:

i. v(s,x,t) is positive definite with respect to s;

ii. on the sphere $||s|| \le R$ for all $x \in \Omega$ and any t:

$\inf v = h_R > 0, \quad \sup v = H_R > 0$

$$||s|| \leq R \qquad ||s|| \leq R$$

with h_R and H_R only dependent on R;

iii. a total time derivative of v is negative everywhere this function is defined and

$$\sup_{||s|| < R} \dot{v} = -m_R.$$

with $m_R > 0$ only dependent on R.

The next theorem illustrates the nature of specialized application results which may be obtained from theorem 1 and 2.

Theorem 3 [Ut.1, p.148] A necessary and sufficient condition for the system

$$\dot{x} = A(t)x + B(t)u,$$

equipped with the controller

$$u = \sum_{i=1}^{n} \psi_i x_i - \delta \cdot SIGN(slm)$$

with $\delta > 0$, $slm \triangleq cx$

$$\psi_i = \alpha_i \quad \text{if } x_i slm < 0$$

$$\psi_i = \beta_i \quad \text{otherwise,}$$

to be under sliding mode control is that

$$\alpha_i \leq \max_i \langle c, a_i \rangle / \langle c, B \rangle$$

$$\beta \geq \min_i < c,a_i > / <c,B >$$

where: a_i is the i-th column of A(t).

The following result gives an indication of the limits of a sliding mode controller.

Theorem 4. [Ut.1, p.213] . Let

$$\dot{x} = Ax+Bu+Df$$

be under sliding mode control. A necessary condition for the system dynamics to be invariant with respect to the perturbation f is that the columns of the D matrix be in the range of the B matrix.

A more complete treatment of this subject may be found in the cited references (see also [Fu.1, p.226]). A detailed educational study of a sliding mode controller for a robotic link with an elastic transmission can be found in [RMDS.3]. A hands-on analysis of the dynamical behavior of such a system can be developed using the simulator ROBEL (available at the Computer Center; see Appendix B).

5.3 PID/Sliding mode controllers

A PID/sliding mode controller is obtained by modifying a classical PID with the parallel addition of a nonlinear (almost) on-off switching element (fig. 3.1). The idea is to complement the well established transient performance and steady state rejection properties of the classical PID with the potential robustness to plant parameter variation and external perturbations of the sliding mode controller.

To illustrate, consider a robotic link described by

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{1}{\tau} x_2 + \frac{K_m}{\tau} (u+\xi)$$

where u is the output of a controller with the structure indicated in fig. 3.1. With reference to this figure, observe that if Δu, the control contribution of the switching element, is capable of imposing the <u>sliding mode</u> condition

$$slm(x) \triangleq a_1 x_1 + a_2 x_2 + x_3 = 0$$

where

$$\dot{x}_3 = x_1 - x_{1D},$$

then it would follow

$$slm = 0$$

which implies

$$x_1(s) = \frac{x_{1D}(s)}{1+a_1 s+a_2 s^2} \quad ;$$

that is: a response completely independent of ξ and of the plant parameters.

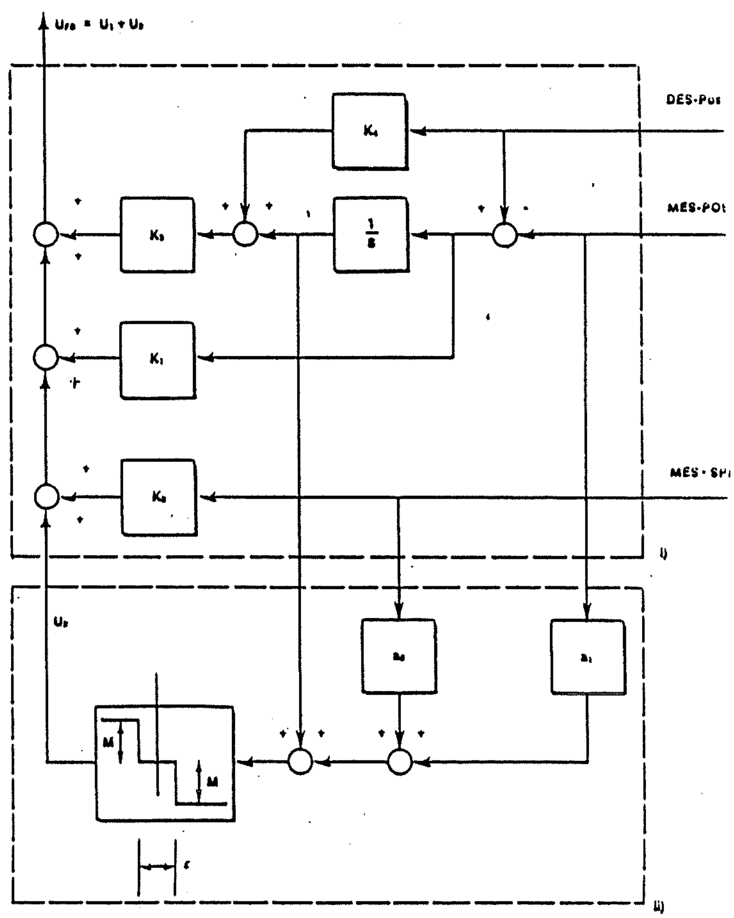


Figure 3.1: PID/Sliding mode servocontroller

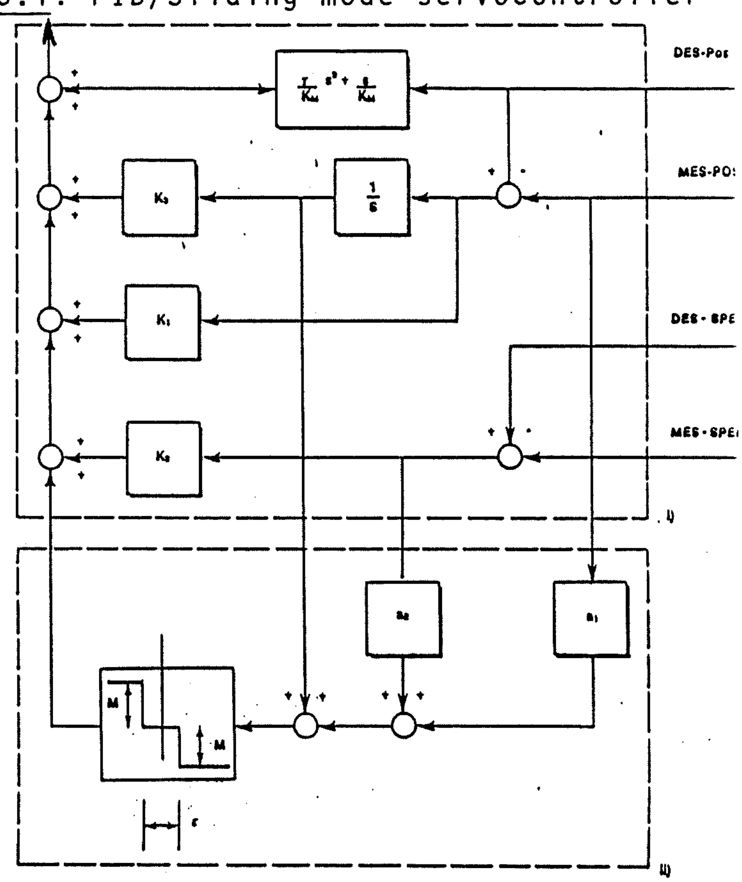


Figure 3.2: PID/Sliding mode trajectory tracking controller

To determine the requirements for a Δu with such a capability, let $K_1 \ldots K_4$ be such that the PID linear component, in the absence of ξ and Δu, generates an input/output transfer function with poles p_1, p_2, p_3 and zero z. Let $a_1 \triangleq -\frac{1}{p_1} - \frac{1}{p_2}$, $a_2 \triangleq \frac{1}{p_1 p_2}$, $z = p_3$.

Observe that, using Laplace notations,

$$slm(s) = \frac{K_m}{\tau} \frac{\Delta u + \xi}{(s - p_3)}$$

and

$$s\dot{l}m(s) = \frac{K_m}{\tau} (\Delta u + \xi) + p_3 \frac{K_m}{\tau} \left(\frac{(\Delta u)}{(s - p_3)}\right)$$

hence

$$\frac{1}{2} \frac{d}{dt} (slm(t)^2) = p_3\ slm^2(t) + \frac{K_m}{\tau} (\Delta u + \xi) slm(t)$$

To obtain $slm(t) \to 0$ it is then sufficient to impose $\frac{d}{dt}(slm^2(t)) < 0$. Since $p_3 < 0$, for this it is sufficient to simply choose

$$\Delta u = -M\ \mathrm{SIGN}(slm)$$

where $M > \sup |\xi|$.

5.4 Observations

The M>sup $|\xi|$ inequality suggests that while one may not need to know the value of the perturbation acting on the link, an evaluation of its upper value is required. Since the perturbation reflects the dynamical behavior of other links of the manipulator which may in turn be itself dependent on the value of M, a convenient procedure to obtain such an evaluation is inspired by the "hierarchy of control method" [Ut.1, p.101] [Le.1,p.210-220]. The controller for link i is designed sequentially by: starting from the last link; assuming the dynamics of all the preceding links to be in a sliding mode; continuing the procedure down to the first link.

The sliding mode action may in practice be approximated by replacing the switching control

$$\Delta u = - M \; SIGN \; (slm)$$

with

$$\Delta u = - M \; SIGN \; (slm) \qquad \text{for} \quad |slm| > \epsilon$$
$$= 0 \qquad \text{for} \quad |slm| < \epsilon$$

The ϵ value is chosen sufficiently small as to retain the desired robustness property of the sliding mode action, and large enough as to reduce the high frequency chattering which characterizes the operation of such an action.

The sliding mode controller potential of neutralizing not only the influence of external perturbations but also that of parameter variations follows from the fact that this latter influence may be represented in terms of an input-equivalent perturbation

$$\xi = \frac{\tau}{K_m} \left[\left[\frac{1}{\tau_{nom}} - \frac{1}{\tau} \right] x_2 + \left[\frac{K_m}{\tau} - \frac{K_{m\ nom}}{K_m\ \tau_{nom}} \right] u_o \right]$$

where the suffix "nom" denotes a nominal value and u_o is the control contribution fo the PID/Sliding Mode linear component.

A similar procedure may be applied to modify a PID trajectory tracking controller into a PID/Sliding mode trajectory tracking controller (figure 3.2).

5.5 Application to real time parameter identification

Sliding mode techniques may be conveniently applied to real time parameter identification. To illustrate let a system be described by

$$\dot{x}_1 = x_2 \qquad a \triangleq \frac{1}{\tau}$$

$$\dot{x}_2 = ax_2 + bu \qquad b \triangleq \frac{K_m}{\tau}$$

with a and b unknown. Consider the problem of estimating a and b at time t from the knowledge of $u(s)$, $x_2(s)$, $t \in (0,\infty)$,

$s \in (-\infty, t]$

To solve this problem we first use sliding mode to get an estimate of $\dot{x}_2$. For this we consider the system

$$\dot{\hat{x}}_1 = \hat{x}_2$$

$$\dot{\hat{x}}_2 = \hat{a}(t)x_2(t)+\hat{b}(t)u(t) + \left[c|x_2(t)|+d|u(t)|\right] u^*(t)$$

where:

$\hat{a}(t)$ is the estimate of a; $\hat{b}(t)$ of b;
$u^* \triangleq -\mathrm{SIGN}(slm)$; $slm \triangleq \hat{x}_2(t)-x_2(t)$;
$c>0$, $d>0$ are to be chosen so that $slm(t) \to 0$.

For $slm(t) \to 0$ we observe that

$$\begin{aligned}\frac{d}{dt}\, slm(t)^2 &\triangleq \dot{slm}(t)\, slm(t)\\ &= (\dot{\hat{x}}_2-\dot{x}_2)\,slm(t)\\ &= (\hat{a}(t)-a)x_2\ slm + (\hat{b}(t)-b)u\ slm\\ &\quad - c|x_2|\,|slm| - d|u|\,|slm|\\ &< 0 \quad \text{if}\end{aligned}$$

$$c > \max|\hat{a}-a| \quad , \quad d > \max|\hat{b}-b| \ .$$

With $slm = 0$ one has

$$\hat{x}_1 = x_1 \qquad \dot{\hat{x}}_2 = \dot{x}_2 \quad \text{for almost every t.}$$

We can therefore estimate $\hat{a}(t)$, $\hat{b}(t)$ from $\ddot{x}_2(\cdot)$, $x_2(\cdot)$, $u(.)$.

For this we set up the following algorithm

$$\varepsilon_1(t) \stackrel{\Delta}{=} \ddot{x}_2(t) - \hat{a}(t)x_2(t) - \hat{b}(t)u(t)$$

$$\varepsilon_2(t) \stackrel{\Delta}{=} \ddot{x}_2(t-D) - \hat{a}(t)x_2(t-D) - \hat{b}(t)u(t-D)$$

$$\dot{\hat{a}}(t) = \varepsilon_1(t)x_2(t) + \varepsilon_2(t)x_2(t-D)$$

$$\dot{\hat{b}}(t) = \varepsilon_1(t)u(t) + \varepsilon_2(t)u(t-D)$$

for some $D > 0$. To convince ourselves that the algorithm converges, we consider the Lyapunov function

$$V = (a-\hat{a}(t))^2 + (b-\hat{b}(t))^2$$

and observe that

$$\begin{aligned}
\dot{V} &= -(a-\hat{a}(t))\dot{\hat{a}} - (b-\hat{b}(t))\dot{\hat{b}} \\
&= -(a-\hat{a}(t)) \left[\varepsilon_1(t)x_2(t) + \varepsilon_2(t)x_2(t-D)\right] \\
&= (b-\hat{b}(t)) \left[\varepsilon_1(t)u(t) + \varepsilon_2(t)u(t-D)\right] \\
&= -\left[(a-\hat{a}(t))\, x_2(t) + (b-\hat{b}(t))\, u(t-D)\right]^2 \\
&\quad - \left[(a-\hat{a}(t)x_2(t-D) + (b-\hat{b}(t))u(t-D)\right]^2
\end{aligned}$$

$$< 0 \quad \text{if} \quad \det \begin{bmatrix} x_2(t) & u(t) \\ x_2(t-D) & u(t-D) \end{bmatrix} \neq 0$$

The technique illustrated by this example may be readily generalized to the problem of identifying the A parameter matrix of

$$\dot{x} = A\ \varphi(x,t)$$

from the knowledge of $x(\cdot)$ and $\varphi(\cdot,\cdot)$. One first estimates $\dot{\hat{x}}$ by considering

$$\dot{\hat{x}}_i = \hat{A}_i\ \varphi(x,t) + \sum_{j=1}^{n} B_{ij}\ |\varphi_j(x,t)|u^*_j$$

with $u^*_j \triangleq -\mathrm{SIGN}(slm_j)$

$$slm_i \triangleq \hat{x}_i - x_i$$

and B_{ij} such that $slm_i \to 0$.
One then identifies $\hat{A}$ from $\dot{\hat{x}}$, x, and $\varphi(x,t)$. More on this in [Ut.1, p. 236-240] .

6. LEARNING CONTROL

6.1 Introduction

To introduce the concept of a learning control it is useful to consider a scenario whereby a robotic manipulator has to repetitively execute a given task; an optimal execution requires the manipulator links position to follow a predetermined trajectory; the presence of an unknown perturbation allows this trajectory to be attained only within a certain error. The problem is to exploit the knowledge of this error in correspondence with a given execution to complement the action of the available controller so as to come up with a smaller error at a subsequent execution (fig. 1.1).

6.2 Implementation in a computed torque scheme

To be more specific, let us introduce the following notations:

$q_D(\cdot)$, $q_k(\cdot)$: manipulator links desired and actual k-th execution position trajectory;

$e_k(\cdot)$: $= q_D(\cdot) - q_k(\cdot)$, trajectory error at the k-th execution;

$f(\cdot)$: the unknown input equivalent perturbation causing the error;

$d_k(\cdot)$: the acceleration equivalent action provided by the learning module in response to e_{k-1}.

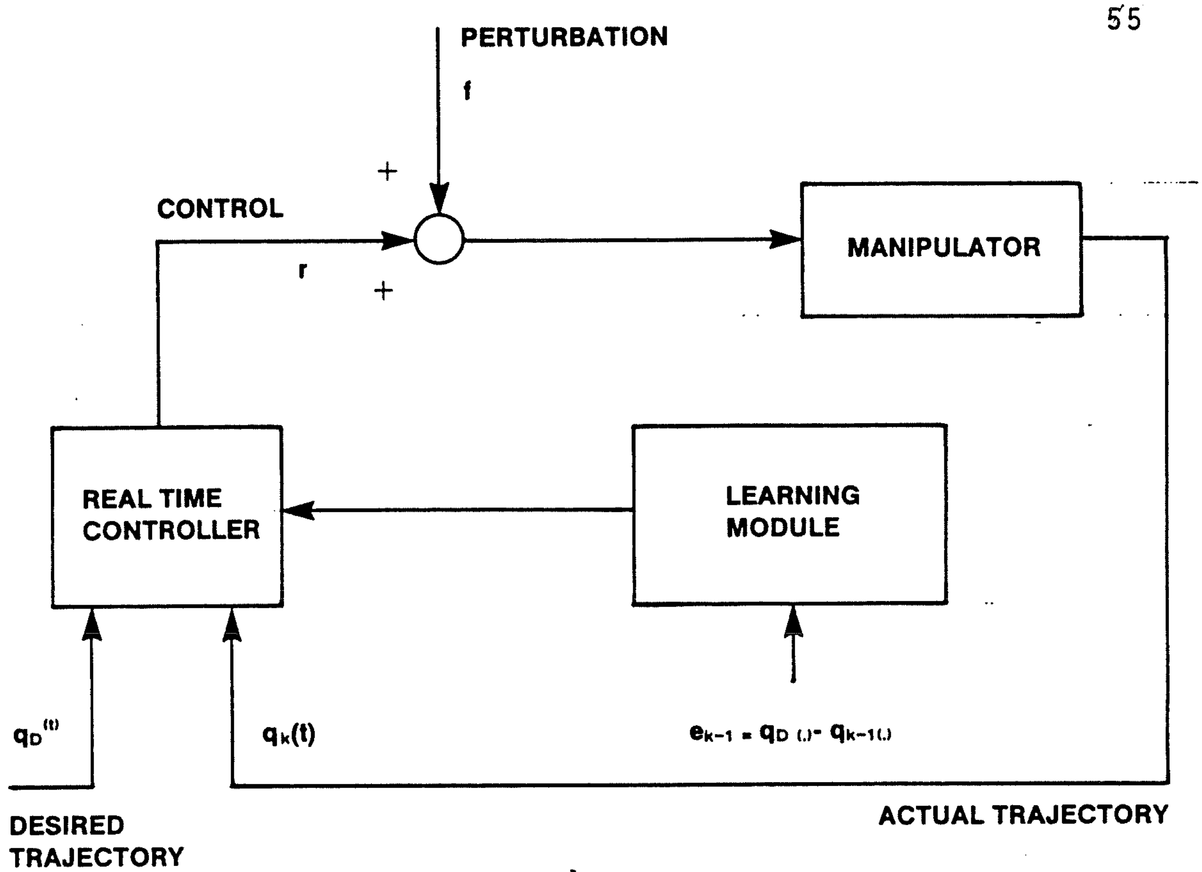


Figure 1.1: Learning control scenario.

Represent these functions as elements of $L_2\,[0,\infty)^n$.

Assuming the utilization of a precomputed torque controller, the manipulator dynamics at the k-th execution is represented by [Cr.2]

$$D_c\left[\ddot{q}_D + K_1 e_k + K_2 \dot{e}_k + d_k\right] = D\ddot{q} + f$$

where D_c and D represent computed and actual inertia matrices. Further, assuming $D_c = D$, it follows:

$$K_1 e_k + K_2 \dot{e}_k + \ddot{e}_k + d_k \triangleq D^{-1} f \triangleq d$$

hence

$$e_k(s) = H(s)\left[(d-d_k)(s)\right]$$

with

$$H(s) = \frac{1}{K_1+K_2 s+s^2}$$

Representing the learning module as described by

$$d_{k+1} = d_k + Qe_k$$

with $Q: L_2[0,\infty)^n \to L_2[0,\infty)^n$ a (not necessarily casual) operator to be determined, one has

$$d_{k+1} = d_k + QH\left[d-d_k\right] .$$

For the intended objective to be attained it is then sufficient to require $|d-d_k| \to 0$, hence to choose Q so that $|I-QH| < 1$. A potential (non casual) choice of such a Q is for example represented by $Q=\alpha H^*$ with H^* the adjoint of H and constant α such that $|H| < 1/\alpha$.

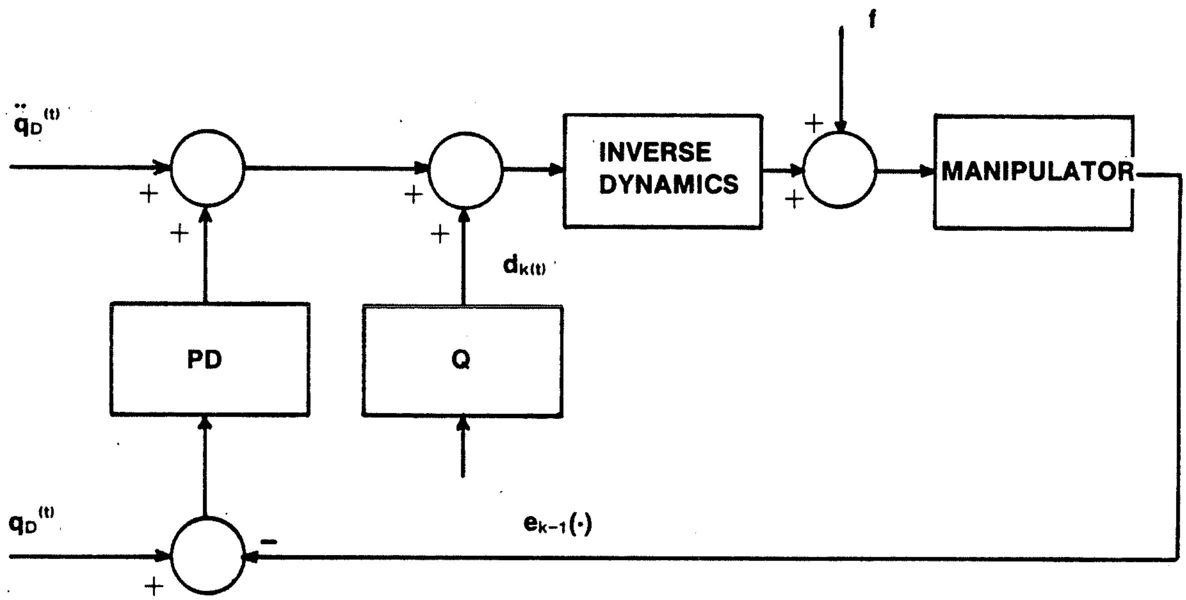


Figure 2.1: Learning control applied to a computed torque controller.

BIBLIOGRAPHY

Cr. 1 Craig, J.J., Introductions to Robotics, Addison Wesley 1986.

As. 1 Asada, H., Slotine, J-J E., Robot Analysis and Control, Wiley 1986.

Fu. 1 Fu, K.S., Gonzales, R.C., Lee, CS.G., Robotics: Control, Sensing, Vision, and Artificial Intelligence, McGraw Hill Book Inc. 1987.

Br. 1 Brady, M., & Co., Robot Motion Planning and Control, MIT Press 1984.

Ut. 1 Utkin, V.I.,Sliding Modes and their Application in Variable Structure Systems, Mir Publisher, Moscow, 1978.

It. 1 Itkins,I.,Control Systems of Variable Structure, John Wiley & Sons Inc., New York, 1976.

La. 1 Landau,Yoan D., Adaptative control, University of Southern California, Dekker 1979.

As. 1 Astrom, K.J. & Co., Theory and Application of Self-Tuning Regulators, Automatica, 13,pp. 457-476, 1977.

En. 1 Engelberger, J.F., Robotics in Practice, AMA-COM, 1980.

Pa. 1 Paul, R.P., Robot Manipulators: Mathematics, Programming and Control, The MIT Press 1981.

Cr. 2 Craig, J.J., Adaptive Control of Mechanical Manipulators, Addison-Wesley 1988.

Le. 1 Lee, C.S.G., Gonzales, R.C., Fu, K.S., Tutorial on Robotics, IEEE Computer Society Press, 1985.

Ecole Polytechnique technical reports

RMDS.1 De Santis, R.M., Théorie des systènes de commande linéaires, EPM Notes de cours, 1986.

RMDS.2 De Santis, R.M., Eléments fondamentaux de la robotique: cinématique, dynamique et commande, EPM Notes de cours, 1986.

RMDS.3 Serfass, C., De Santis, R.M., Contrôleur à structure variable pour une articulation robotique élastique, EPM/RT-87/19, Mai 1987.

RMDS.4 Nguyen, A.T., De Santis, R.M., Etude de Sensibilité du comportement dynamique d'un manipulateur robotique, EPM/RT-87/2, janvier 1987.

RMDS.5 Quintal, M., De Santis, R.M., Commande adaptative d'une articulation robotique avec la méthode du modèle de référence, EPM/RT-88/15, Mai 1988.

RMDS.6 Souphandavong, P., De Santis, R.M., Exécution robotisée du jeu de la Tour d'Hanoi, EMP/RT-87/15, Mai 1987.

RMDS.7 De Santis, R.M., Commande adaptative d'une articulation robotique avec la méthode du Self-Tuning, EPM/RT-88/XX (en préparation).

RMDS.8 De Santis, R.M., PID/Sliding Mode Control of a robotic link, EMP/RT-88/XX (en préparation).

APPENDIX A: LABORATORY SIMULATION EXPERIMENTS

Among the variety of techniques applicable to the control of a robotic link we have seen classical PID, model reference and self-tuning adaptive control, sliding mode control. Our exposition, complemented with the cited bibliography should help the student to familiarize himself with adavantages and shortcomings of each one of these approaches, efficient computational and numerical design procedures, application modalities, and so on.

All this notwithstanding, a number of factors still remain that make these approaches difficult to put into perspective: the complexity of the often nonlinear dynamics, the variety of design criteria, the heuristic nature characterizing some of the design parameters choices, the sometime hard to establish connection between theoretical and physical implications of the available results.

In what follows we suggest a number of questions which may be advantageously clarified via hands-on simulation experiments on a personal computer. These experiments may be carried out by using the simulators described in appendix B.

Simulation Experiment *1: APPLICATION OF A CLASSICAL PID CONTROLLER TO A ROBOTIC LINK

The simulators used in this experiment are POSSPEED and SIMNOG. The questions under investigation are the following:

How well does a PID perform? How to characterize performance? What is the function of each of its components? How to tune the gains? How do gain variations influence performance? What is the influence of other controller parameters? (sampling period, quantization levels, measurement noise, transducer gains, maximum control levels, delay in the control, ...).

What is the influence of the various kinematic and mass manipulator parameters (masses of the links, inertias, lengths, working configuration, viscosity, ...)?

How effective is a PID in neutralizing intercouplings, centripetal and Coriolis forces?

What are a PID most advantageous/disadvantageous features?

Simulation experiment *2: APPLICATION OF MODEL REFERENCE AND SELF-TUNING ADAPTIVE CONTROLLERS TO A ROBOTIC LINK

This experiment uses the simulators MRAC and SELFTUNE. The objective is to analyze via simulation the following type of questions:

Why, how, and with what success an adaptive controller of the model reference or self-tuning type?

What is the comparative difference between the two?

What is the influence of the gain adaptive module parameters?

What are the control strategies?

What "ad hoc" modifications of the off-the-shelf algorithms might be required?

What is the sensitivity to external perturbations, the response time, required control level, robustness, quality of performance?

Simulation experiment *3: SLIDING MODE CONTROL OF A ROBOTIC LINK

The simulator used in this experiment is ROBEL; the objective is to analyze the following type of questions:
Why, how and with what success a sliding mode controller?
Importance of the various parameters of the controller?
Basic structure and required algorithms for the controller?
Comparisons with PID and linear state regulator?
Sensitivity to external perturbations, response time, control level requirements, robustness, reliability, stability, quality of performance.

APPENDIX B: LABORATORY SIMULATORS

In what follows we give a brief description of the simulators used in the laboratory experiments mentioned in Appendix A.

POSSPEED (POSition SPEED control simulator)

POSSPEED simulates a link angular position controller (fig. 1.3.2); the mathematical model is in fig. 1.1; the non-linear element can be anyone of the types in fig. 1.2. The gain values may be computed automatically by POSSPEED so as to position the poles as a function of the damping factor ξ and of the A,B,C acceleration factors indicated in fig. 1.3. Other simulation possibilities include: various PID indutrial configurations (threshold, saturation on the integral component), influence of a delay in the control, external perturbation influence, speed rather than position control.

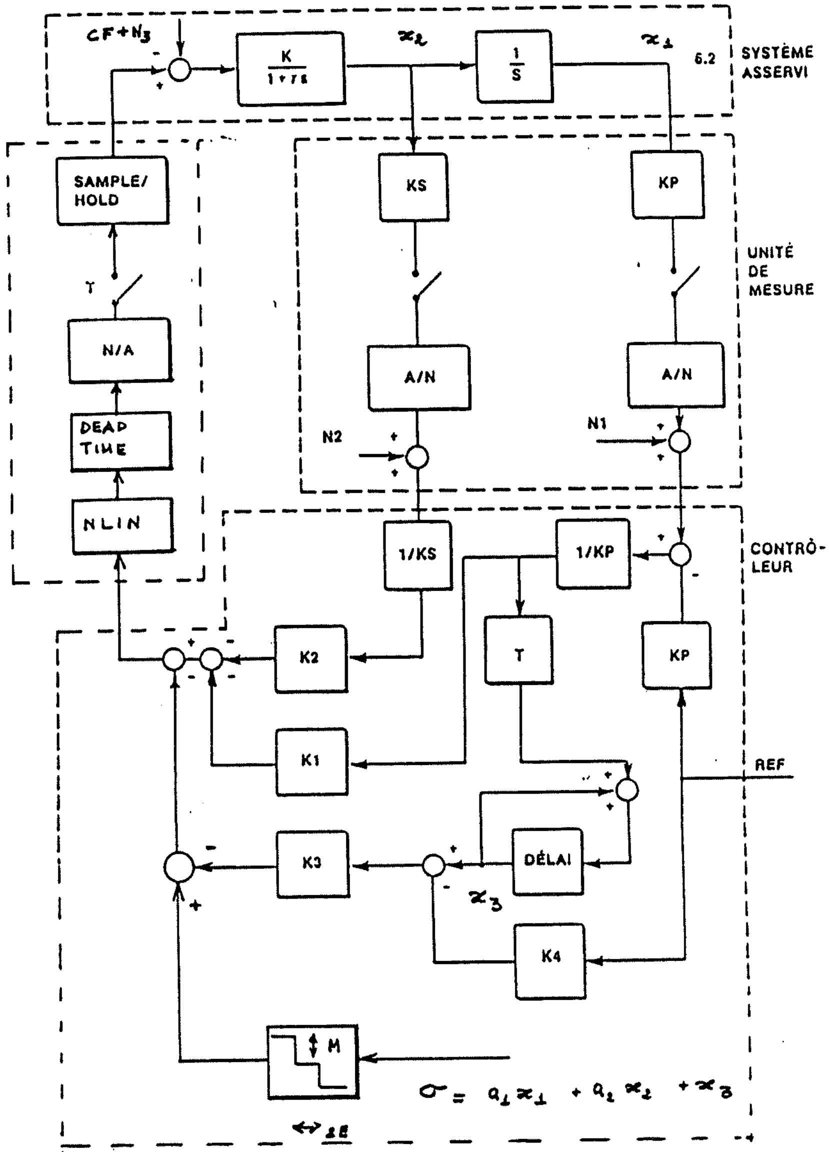


Figure 1.1: Bloc diagram of the system considered in POSSPEED

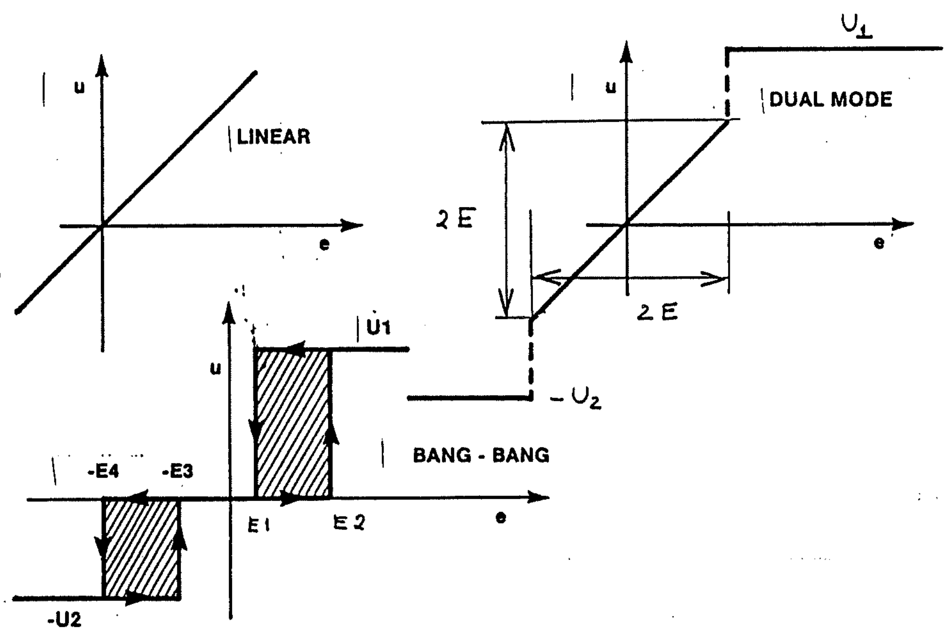
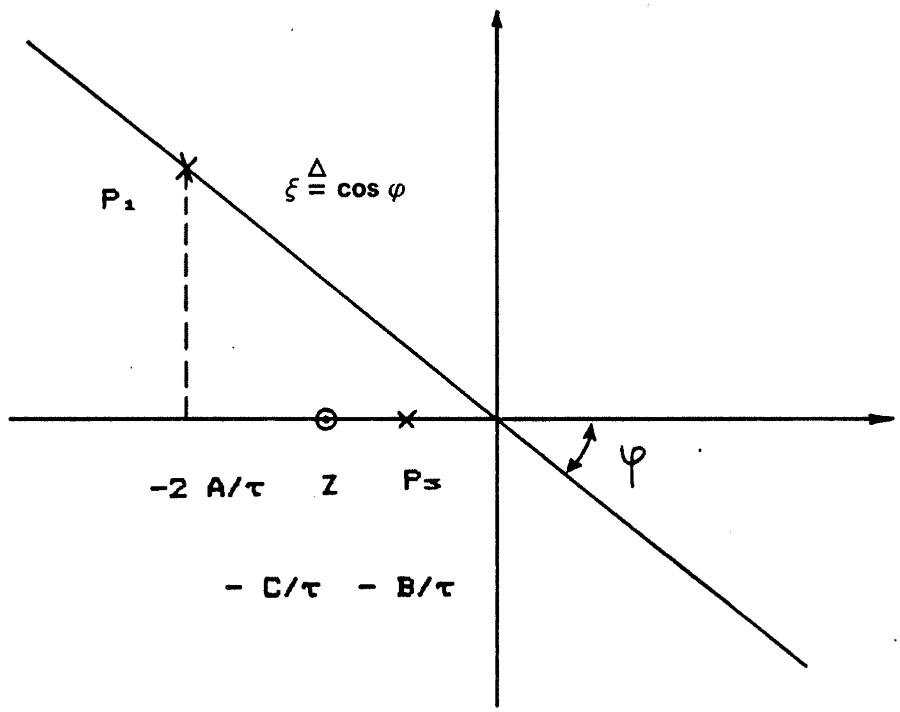


Figure 1.2: The nonlinear element in POSSSPEED



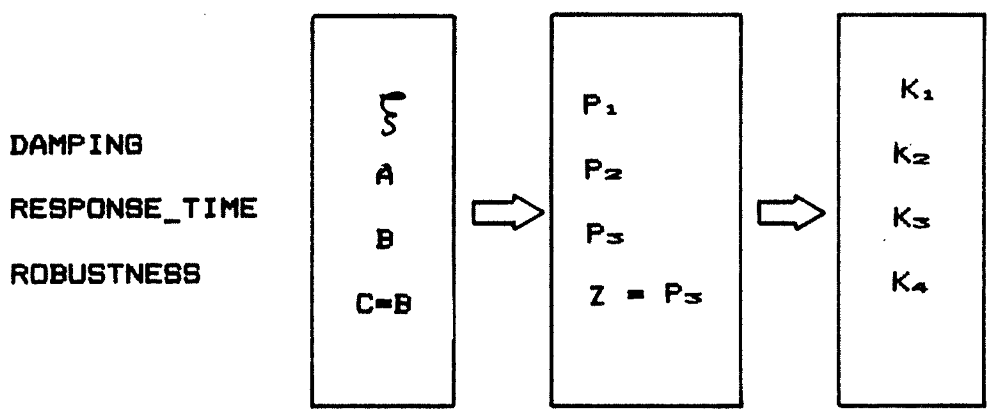
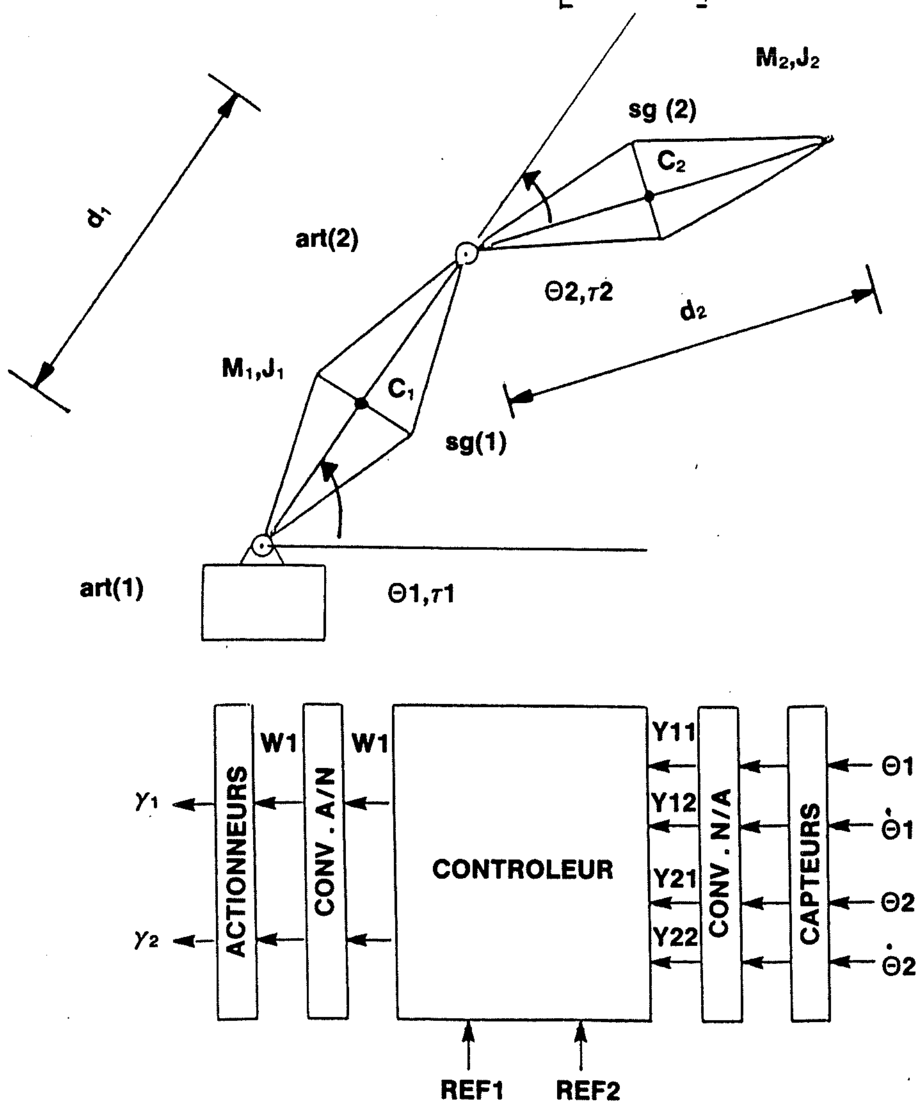


Figure 1.3: Gain adjustment in POSSPEED

<u>SIMNOG</u> (SIMulation with NO Gravity)

SIMNOG simulates the control system in fig. 2.1; the manipulator dynamical model is as in fig. 2.2; the controller is made of two decentralized PID components (fig. 3.2.1), plus a feedforward component compensating for gravity; the nonlinear element is identical to that considered in POSSPEED. More details can be found in [RMDS.4].



<u>Figure 2.1</u>: The robotic system considered in SIMNOG

$$x11 = \theta1$$
$$x12 = \dot{\theta}1$$
$$x13 = \int (\theta1 - REF1)$$
$$x21 = \theta2$$
$$x22 = \dot{\theta}2$$
$$x23 = \int (\theta2 - REF2)$$

$$D11 = (1/4)m1\ d1^2 + m2[d1^2 + (1/4)d2^2 + d1\ d2\ \cos x21] + I1 + I2$$
$$D12 = m2(d2^2/4 + d1\ d2\ \cos x21/2) + I2$$
$$D21 = m2(d2^2/4 + d1\ d2\ \cos x21/2) + I2$$
$$D22 = (1/4)m2\ d2^2 + I2$$
$$D112 = -\ m2\ d1\ d2\ \sin x21$$
$$D122 = -\ (1/2)m2\ d1\ d2\ \sin x21$$
$$D211 = +\ (1/2)m2\ d1\ d2\ \sin x21$$
$$a = 1\ /\ (D11\ D22 - D12\ D21)$$
$$T1' = T1 + T1g + T1v$$
$$T2' = T2 + T2g + T2v$$
$$T1 = -K11(x11-REF1)-K12\ x12-K13\ x13+K13\ K14\ REF1$$
$$T1g = -g[(m2 + (1/2)m1)d1\ \cos x11 + (1/2)m2\ d2\ \cos(x11 + x21)]$$
$$T1v = -\ f1\ x12 + f2\ x22$$
$$T2 = -K21(x21-REF2)-K22\ x22-K23\ x23+K23\ K24\ REF2$$
$$T2g = -g[(1/2)m2\ d2\ \cos(x11 + x21)]$$
$$T2v = -\ f2\ x22$$

$$\dot{x}11 = x12$$
$$\dot{x}12 = a\ D22(T1' - D112\ x12\ x22 - D122\ x22^2) - a\ D12(T2' - D211\ x12^2)$$
$$\dot{x}13 = x11 - REF1$$
$$\dot{x}21 = x22$$
$$\dot{x}22 = -\ a\ D21(T1' - D112\ x12\ x22 - D122\ x22^2) + a\ D11(T2' - D211\ x12^2)$$
$$\dot{x}23 = x21 - REF2$$

Figure 2.2: The manipulator dynamical model considered by SIMNOG

MRAC (Model Reference Adaptive Control)

MRAC simulates the dynamic behavior of an adaptive control system of the Model Reference type (fig. 4.2.1).
The controller/plant part is made of a PD position servo (fig. B.3.1); the reference model is as in fig. B.3.2; the gain adaptation module responds to the scheme in fig. B.3.3.
The simulation allows one to analyze the system behavior with and without adaptation as a function of the following parameters: plant time constant and static gain, initial conditions, reference model dynamic parameters, error weighting coefficients, adaptation module gains. More details can be found in [RMDS.5] .

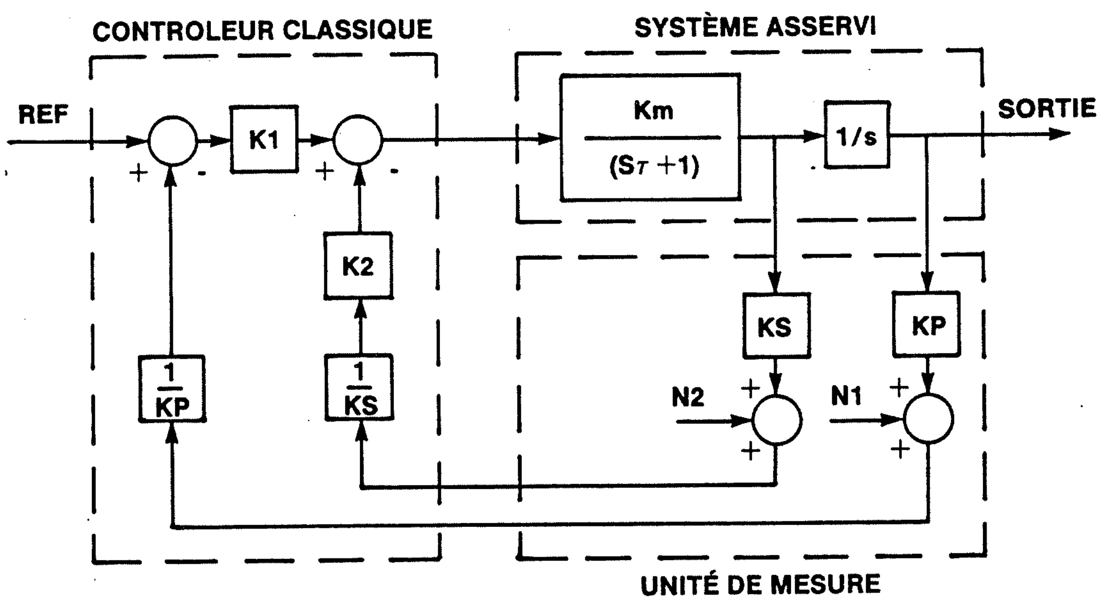
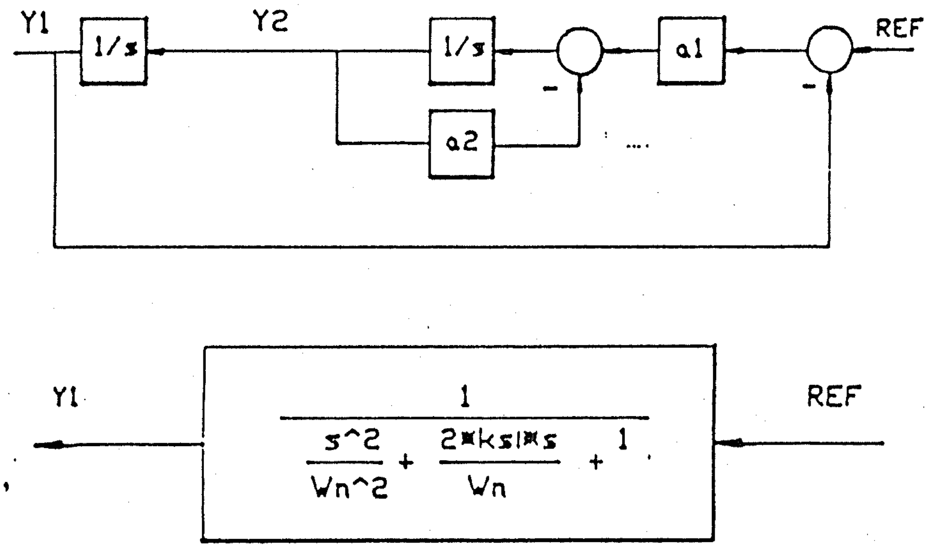


Figure 3.1: Controller/plant ensemble considered in MRAC



$$\frac{1}{\frac{s^2}{Wn^2}+\frac{2*ksi*s}{Wn}+1}$$

Figure 3.2: The reference model in MRAC

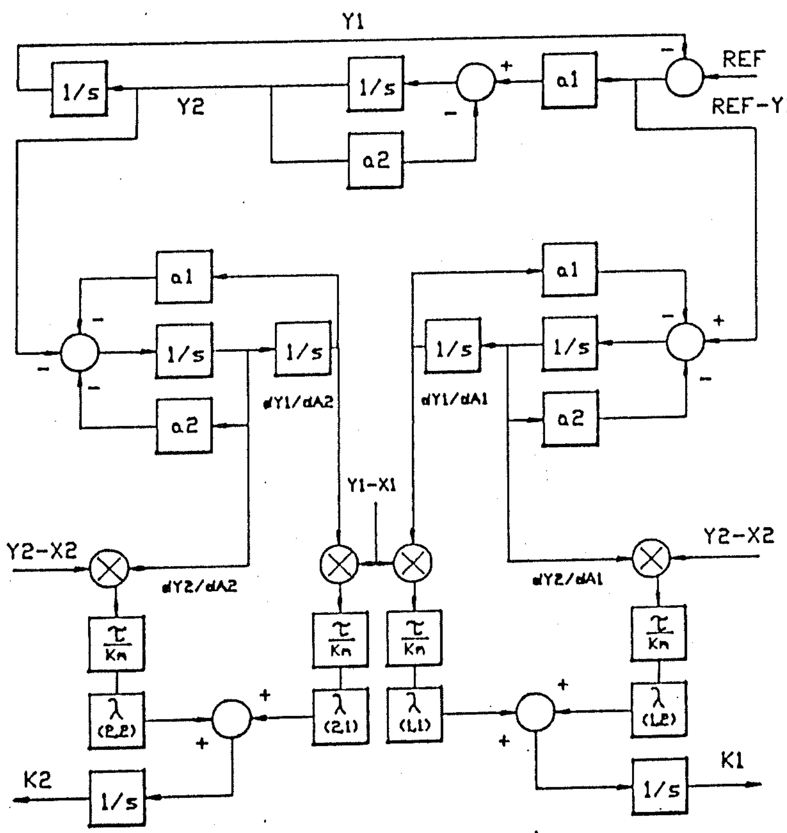
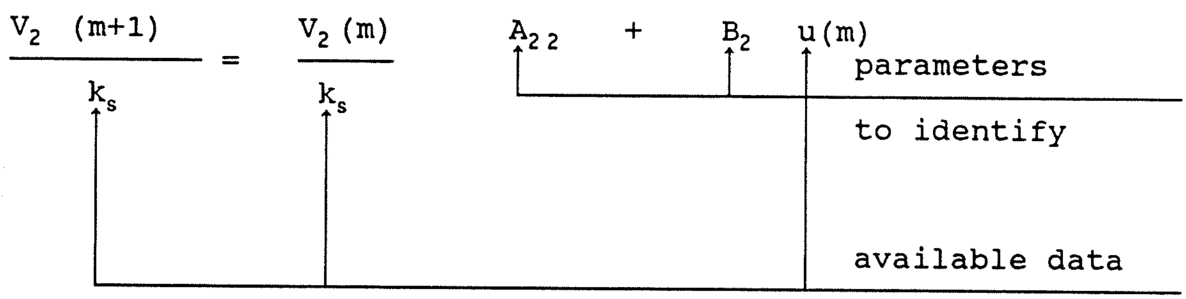


Figure 3.3: Gain adaptation module in MRAC

SELFTUNE (SELFTUNing Controller)

SELFTUNE simulates the dynamic behavior of an adaptive control system of the self-tuning type (see fig. 4.3.1). The controller-plant part is identical to that considered in fig. 1.1. The parameter estimator is of the recursive least square type; it offers the choice to include or not a speed measurement; it uses the formulas given in section 4.3 with the symbol notation illustrated in fig. 4.1 or 4.2. More details are to be found in [RMDS.7] .



$$y(m+1) \stackrel{\Delta}{=} \frac{V_2\ (m+1)}{K_s} \qquad x(m) \stackrel{\Delta}{=} \begin{bmatrix} y(m) \\ u(m) \end{bmatrix}$$

$$\hat{\Theta}\ (m) \stackrel{\Delta}{=} \begin{bmatrix} A_{22} \\ B_2 \end{bmatrix}$$

Figure 4.1: Correspondence of notations in the estimator of SELFTUNE (a speed measurement is available)

$$x_1\,(i+1) = x_1(i) + A_{12}\,x_2(i) + B_1\,u(i)$$

$$x_2(i-1) = \frac{1}{A_{12}}\left[x_1(i) - x_1(i-1)\right] - \frac{B_1}{A_{12}}\,u(i-1)$$

$$x_2(i-1) = A_{22}\,x_2(i-2) + B_2\,u(i-2)$$

$$\frac{1}{A_{12}}\left[x_1(i) - x_1(i-1)\right] - \frac{B_1}{A_{12}}\,u(i-1) = \frac{A_{22}}{A_{12}}\left[x_1(i-1) - x_1(i-2)\right]$$

$$- \frac{A_{22}}{A_{12}}\,B_1\,u(i-2) = B_2\,u(i-2)$$

$$\frac{V_1(i) - V_1(i-1)}{K_p} = A_{22}\left[\frac{V_1(i-1) - V_1(i-2)}{K_p}\right] + B_1\,u(i-1) + (B_2 A_{12} - B_1 A_{22})\,u(i-2)$$

parameters to identify

available data

$$y(m+1) \overset{\Delta}{=} \left[V_1(m+1) - V_1(m)\right] / K_p$$

$$x(m) \overset{\Delta}{=} \begin{bmatrix} y(m) \\ u(m) \\ u(m-1) \end{bmatrix} \qquad \hat{\Theta} \overset{\Delta}{=} \begin{bmatrix} A_{22} \\ B_1 \\ B_2\ A_{22} - B_{22} \end{bmatrix}$$

Figure 4.2: Correspondence of notations in the estimator of SELFTUNE (a speed measurement is not available)

ROBEL (Control of a ROBotic link with an ELastic Transmission)

ROBEL simulates the dynamic behavior of a robotic link position controller characterized by an elastic transmission (fig. 5.1). The controller configuration can be chosen as one of the following:

i. classical PID (fig. 3.2.1)

ii. PID/sliding mode (fig. 4.3.1)

iii) classical linear state regulator (fig. 5.2)

iv) nonlinear sliding mode state regulator (fig. 5.3).

More details are to be found in [RMDS.3] .

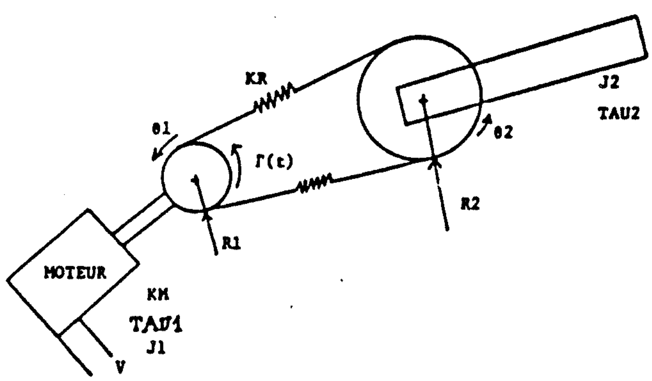


Figure 5.1: Elastic link considered in ROBEL

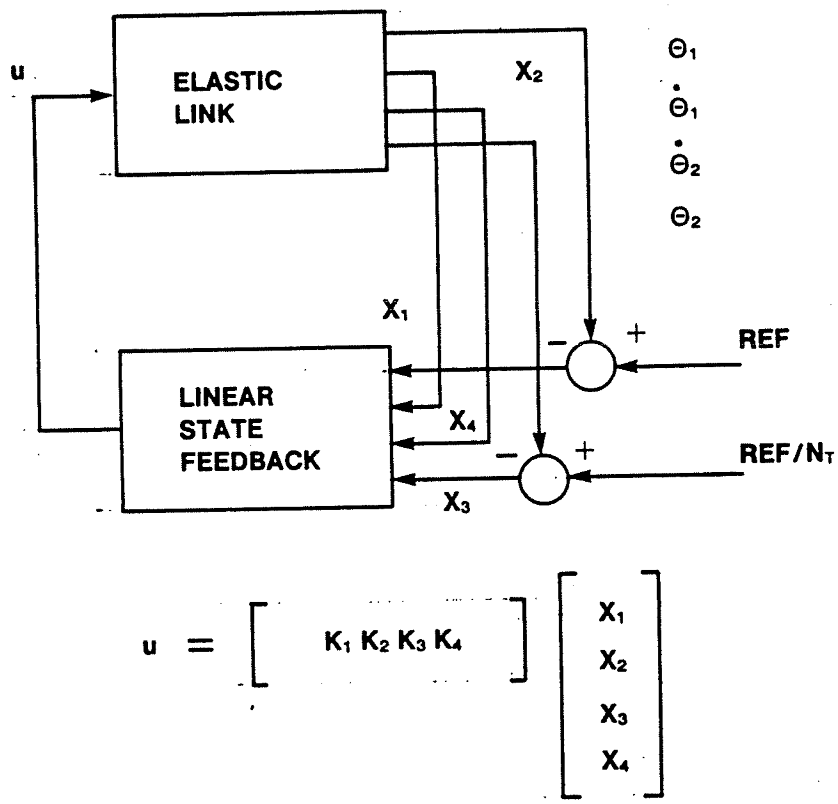


Figure 5.2: Linear state regulator used in ROBEL

$$u = \sum_{i=1}^{n} \psi_i * X_i$$

$$\psi_i = \begin{cases} \alpha_i & \text{si} \quad x_i * \sigma > 0 \\ \beta_i & \text{si} \quad x_i * \sigma < 0 \end{cases} \qquad i = 1..n$$

$$\sigma \triangleq C * X$$

$$C \triangleq [c1 \quad \ldots\ldots \quad cn]$$

$$\alpha_i < l_i$$

$$l_i \triangleq - \frac{A_i^T * C}{B^T * C}$$

$$\beta_i > l_i$$

$$c1 = -(p1+p2+p3)-(1/\tau 2)$$

$$c2 = 1$$

$$c3 = -(J2/(KR*R2^2))*(p1*p2*p3) - c1$$

$$c4 = (J2/(KR*R2^2))*(p3*p1+p3*p2+p1*p2) - 1 - (c1*J2)/(c2*KR*R2^2)$$

$$l1 = (\tau 1/Km)*((KR*R1^2/J1)-(c4*KR*R2^2/J2))$$

$$l2 = (\tau 1/Km)*((1/\tau 1)-c1)$$

$$l3 = (\tau 1/Km)*(c4*(KR*R2^2/J2)-(KR*R1^2/J1))$$

$$l4 = (\tau 1/Km)*((c4/\tau 2)-c3)$$

Figure 5.3: Variable structure controller used in ROBEL