(Adaptive) computed torque control of (flexible) robot systems

(Adaptive) computed torque control of (flexible) robot systems — I.

Ivonne M.M. Lammerts 1990

T.U.E. *W.F.W.*– rapport 90.063

December 1990.

Technical University Eindhoven.
Faculty of Mechanical Engineering.
Fundamental Mechanical Engineering.

Research project: Advanced control concepts for nonlinear flexible mechanical systems.

Control group of: *Prof. dr. ir. J.J. Kok, Dr. ir. F.E. Veldpaus.*

(ADAPTIVE) COMPUTED TORQUE CONTROL OF (FLEXIBLE) ROBOT SYSTEMS

Summary

A *mechanical manipulator* can be defined as a multi degrees of freedom open-loop chain of mechanical linkages interconnected by joints. This mechanism, driven by actuators at the joints, is capable of moving the object at the end of the robot arm along a prescribed trajectory in space. To implement high-performance control, even when the manipulator dynamics are poorly known or when large and unpredictable variations occur, **adaptive control** is considered, being a process of modifying one or more parameters of the structure of the control system and/or the control actions so as to force the response of the closed-loop system towards a desired one. Among various types of adaptive robot control systems, the **Model Reference Adaptive Control (MRAC)** systems are important since they lead to relatively easy-to-implement systems with a high speed of adaptation and can be used in a variety of situations. However, it turns out to be difficult to derive convergence, stability and robustness conditions and it is hoped that a more unified framework for choosing an adaptation algorithm will be developed in future.

For an orientation in the field, five MRAC methods in literature are investigated. Attention is focussed on the adaptive sliding controller of Slotine and Li [1987], in which the robot nonlinearities are compensated by feedback control. The model parameters are estimated on-line by an adaptation algorithm, based on the hyperstability theorem of Popov [1969]. This theorem offers a systematic solution to the *stability* problem, while Lyapunov's second method requires the (probably difficult) choice of an appropriate function candidate. In order to assure robustness in the presence of model uncertainties and (environmental) disturbances, a sliding control term is incorporated into the control input.

Today, industrial robots are used for various purposes. Because of hardware limitations in on-line applications, until now, robot control has been studied extensively under the assumption that the actuator transmissions are stiff and that the links can be modeled as rigid bodies. Therefore, most of today's robots have a very stiff (and thus heavy) construction in order to avoid deformations and vibrations. For *higher operating speeds*, industrial robots should be *lightweight* constructions to reduce the driving force/torque requirements and to enable the robot arm to respond faster. However, a lightweight manipulator may have flexibility in the link structure and elasticity in the transmissions between actuators and links. For most manipulators, elasticity of the motor transmissions has a greater significance for the design of the controller than the deformation of the flexible links. Furthermore, link flexibility can be approximately modeled by a chain of rigid sublinks interconnected by elastic joints. Hence, more accurate models involving *elastic transmissions* should be taken into account to pursue better dynamic performance of industrial robots. The application of more complex control algorithms is possible now due to the availability of advanced multiprocessor equipment for real-time manipulator control.

Contents:

Inleiding

Het project '*Geavanceerde regelconcepten voor niet-lineaire flexibele mechanische systemen*' heeft tot doel het ontwikkelen van concepten voor het on-line regelen van niet-lineaire mechanische systemen en heeft als uitgangspunt een model van die systemen. In dat model wordt rekening gehouden met onzekerheden ten aanzien van (eventueel in de tijd variërende) systeemparameters, met elastische vervormingen en met speling en wrijving. Klassieke concepten uit de lineaire regeltheorie (PID-regelaars, optimale regelaars, etc.) zijn niet zondermeer bruikbaar en derhalve worden momenteel nieuwe technieken ontwikkeld. De in het vervolg met '*Computed Torque Control*' aangeduide methodes zijn gebaseerd op vereenvoudigde modellen van de te regelen systemen. Beschouwd worden mechanische systemen waarvan de onderdelen t.o.v. elkaar grote verplaatsingen en verdraaiingen kunnen ondergaan. Bij dit onderzoek wordt in het bijzonder de aandacht gericht op mechanische manipulatoren die een object langs een vooraf bepaalde gewenste baan in de ruimte dienen voort te bewegen.

Adaptive Computed Torque Control voor stijve manipulatoren met parametrische onzekerheden

In eerste instantie worden de manipulator-elementen stijf verondersteld. Daarnaast wordt aangenomen dat enkele van de systeemparameters, zoals bijvoorbeeld de massa van de last aan het uiteinde van de robotarm, onbekend zijn of eventueel zelfs variëren in de tijd. De taak van een '*Adaptive Computed Torque Control*' systeem is dan om on-line de parameters van het regelmodel te schatten, zodanig dat het geregelde systeem de gewenste baan toch zo goed mogelijk volgt (dus binnen bepaalde marges).

In dit rapport worden vijf adaptieve regelconcepten vergeleken aan de hand van simulaties op een translatie-rotatie (TR) robot (hoofdstukken A en B). De verkregen resultaten zijn bevredigend, maar werpen niet meer licht op de achtergornd van de (verschillen in) deze methodes. Er is wel een algemene aanpak zichtbaar waarbij computed torque control wordt toegepast. In die aanpak worden de model parameters on-line geschat, zodanig dat het geregelde systeem de gewenste trajectorie zo goed mogelijk volgt. De zgn hyperstabiliteitstheorie van Popov [1969], waarop deze methode is gebaseerd, blijkt meer ruimte te scheppen voor het ontwerpen van adaptieve regelingen dan de condities verkregen volgens de meer bekende methode van Lyapunov. Dit opent de weg naar het formuleren van adaptieve regelconcepten voor flexibele manipulatoren.

Geavanceerde regelconcepten voor flexibele manipulatoren

Vanaf hoofdstuk C ligt het accent op het niet-adaptief *regelen van flexibele manipulatoren*. Door het optreden van elastische vrijheidsgraden wordt het totale aantal vrijheidsgraden groter dan het aantal ingangssignalen op de motoren die de robot-elementen ten opzichte van elkaar doen bewegen. Dit levert regelproblemen op die ook door adaptieve regelingen, ontworpen voor stijve manipulatoren, niet kunnen worden opgevangen: simulaties van de eerder gebruikte adaptieve regelingen op dezelfde TR-robot, nu echter met een elastische arm, leiden tot instabiliteiten in het geregelde gedrag. Daarom moet allereerst gezocht worden naar methodes voor het ontwerpen van stabiele regelingen voor flexibele mechanische systemen zonder dynamische onzekerheden.

*) ***Generalized computed torque control***

Heeren [1989] heeft een voorstel gedaan voor een computed torque control versie waarbij, via een zekere vorm van optimalisering ten aanzien van de ingang, getracht wordt alle vrijheidsgraden en tevens de motorkrachten/-koppels binnen bepaalde grenzen te houden (hoofdstuk D). Daarmee wordt voorkomen dat er een ongewenst grote divergentie van deze signalen optreedt, maar enige garantie ten aanzien van de stabiliteit van het resulterende systeemgedrag ontbreekt.

*) ***Sliding computed torque control met stabilizer***

Aan de sliding computed torque regelaar van Slotine en Li [1986] (hoofdstuk B) kan een regelterm worden toegevoegd ter stabilisatie van de optredende elastische oscillaties. Een aanzet daartoe is beschreven in paragraaf D.5. Daarbij wordt gebruik gemaakt van de '*Variable Structure Systems*' (VSS) theorie volgens Utkin [1977]. Deze aanpak heeft tot doel het uitdempen van de optredende elastische trillingen in het systeem, hetgeen bij slappe elementen een heel onnatuurlijk gedrag van het systeem tot gevolg kan hebben.

*) ***Two-time scale computed torque control***

Een zeer in het oog springend alternatief wordt gegeven in het artikel van Slotine en Hong [1986] (hoofdstuk E). Daarbij wordt niet getracht om de flexibele bewegingen volledig te dempen, maar wel om deze te leiden naar een natuurlijker ogend gedrag (de zgn 'manifold'), dat bij benadering kan worden afgeleid uit de bewegingsvergelijkingen van het flexibele systeem door te lineariseren volgens de '*Singular Perturbation Technique*' (Khorosani en Spong [1985], Marino en Nicosia [1984]). Er zijn overeenkomsten te bespeuren met de sliding control-methode van Asada en Slotine [1986], in die zin dat er met een toegevoegde regelterm allereerst gepoogd wordt de flexibele vrijheidsgraden te laten convergeren naar die manifold als zgn. 'switching surface', waarna vervolgens de rest van het systeem, het niet-flexibele gedeelte, als vanouds wordt geregeld om de overige 'stijve' vrijheidsgraden de gewenste trajectorie te laten volgen. Slotine en Hong gebruiken daarbij een regeling volgens de reeds vermelde 'sliding computed torque control' methode van Slotine en Li. Er kan waarschijnlijk evengoed een andere regeling op los gelaten worden.

*) ***(Adaptive) Computed Torque Control van de flexibele manipulator opgesplitst in twee deelsystemen***

Tot slot is in hoofdstuk F het idee uitgewerkt waarbij de bewegingsvergelijkingen van het flexibele manipulator systeem opgesplitst worden in twee deelsystemen (enerzijds de stijve robot-elementen en anderzijds de aandrijvingen), onderling gekoppeld door de elastische verbindingskrachten. Op beide deelsystemen kan vervolgens een vorm van computed torque control worden toegepast, waarbij elastische referentie-verbindingskrachten worden gedefinieerd die het gewenste volggedrag van de stijve robot-elementen tot gevolg kunnen hebben, mits de stabiliteit van beide geregelde deelsystemen in onderlinge samenhang gegarandeerd is. De motor-ingangssignalen worden bepaald, zodanig dat de optredende elastische verbindingskrachten zoveel mogelijk overeenkomen met deze referentiekrachten. De uitbreiding naar adaptief regelen in geval van model-onzekerheden via deze methode wordt kort aangehaald. In eerste instantie wordt getracht volgens de methode van Lyapunov globale stabiliteit te verkrijgen (niet-adaptief); een voor de hand liggend alternatief is de hyperstabiliteitstheorie van Popov (adaptief). Simulatie-resultaten, verkregen bij het regelen van een translatie-roatatie robot met een elastische verbinding tussen de motor en de roterende arm, worden besproken in hoofdstuk G.

Slotopmerking

Verder onderzoek op dit gebied zal vooralsnog fundamenteel van karakter zijn om zodoende te komen tot een basis voor het ontwikkelen van (adaptieve) regelstrategieën voor flexibele manipulatoren. Dit fundamenteel getinte werk zal worden ondersteund door uitwerkingen via simulaties. Op de langere termijn wordt beoogd enkele geselecteerde strategieën voor toetsing aan de praktijk te realiseren op een experimentele xy-tafel.

Referenties

* **Asada, H. and Slotine, J.-J.E.** **(1986)**
  Robot analysis and control.
  John Wiley & Sons,
* **Heeren, T.A.G.** **(1989)**
  On Control of manipulators.
  ISBN 90-9002789-0.
* **Khorosani, K. and Spong, M.W.** **(1985)**
  Invariant manifolds and their application to
  robot manipulators with flexible joints.
  IEEE Int. Conf. on Robotics and Automation, St. Louis.
* **Lammerts, I.M.M.** **(1989)**
  Model Reference Adaptive Control of manipulators –
  Orientation and two methods.
  WFW-rapport nr. 89.033, TUE, Eindhoven.
* **Marino, R. and Nicosia, S.** **(1984)**
  On the feedback control of industrial robots with
  elastic joints: a singular perturbation approach.
  University of Rome, R-84.01.
* **Popov, V.M.** **(1969)**
  Hyperstability of control systems.
  New York, Springer-Verlag.
* **Slotine, J.-J.E. and Hong, S.** **(1986)**
  Two-time scale sliding control of manipulators
  with flexible joints.
  American Control Conference, DSE: DPC 82 AME.
* **Utkin, V.I.** **(1977)**
  Sliding modes and their applications
  to Variable Structure Systems.

A. MRAC CONTROL OF MANIPULATORS.

A short review of 5 methods:

1. Landau, Y.D. (1979)
Adaptive control: the model reference approach.
Marcel Dekker, New York, ISBN 0–8247–6548–6.

2. Horowitz, R. and Tomizuka, M. (1980)
An adaptive control scheme for mechanical manipulators — compensation of nonlinearity and decoupling control.
ASME–paper, no. 8–WA/DSC–6, Winter Annual Meeting, Chicago.

3. Craig, J.J., Hsu, P. and Sastry, S.S. (1986)
Adaptive control of mechanical manipulators.
Proc. of IEEE Intl. Conf. on Robotics and Automation, San Francisco.

4. Sadegh, N. and Horowitz, R. (1987)
Stability analysis of an adaptive controller for robotic manipulator.
Proc. of IEEE Intl. Conf. on Robotics and Automation, San Francisco.

5. Slotine, J.–J. E. and Li, W. (1986)
On the adaptive control of robot manipulators.
ASME Winter Annual Meeting, Anaheim, Dec.

Ivonne Lammerts

A.1 THE DYNAMIC MODEL OF A ROBOT MANIPULATOR.

A **manipulator** is modeled as an open chain of n moving rigid bodies (links) interconnected by cylindrical, revolute or prismatic joints of one degree of freedom, with one end fixed to the ground and the other end free. The actuator forces/torques acting on the joints are the inputs, whereas the joint coordinates represent the outputs.

In the absence of friction, gravity or other disturbances, the dynamic model of a robot manipulator can be written as:

$$\boxed{M(\underline{q})\underline{a} + C(\underline{q},\underline{v})\underline{v} = \underline{u}} \qquad (1),$$

where

$\underline{q}(t)$	is the [n×1] vector of joint displacements (revolute/translational),
$\underline{v}(t)$	is the [n×1] vector of joint velocities,
$\underline{a}(t)$	is the [n×1] vector of joint accelerations,
$\underline{u}(t)$	is the [n×1] vector of applied joint forces/torques,
$M(\underline{q})$	is the [n×n] symmetric,positive definite inertia matrix,
$C(\underline{q},\underline{v})\underline{v}$	is the [n×1] vector of Coriolis and centrifugal forces/torques:

$$C(\underline{q},\underline{v})\underline{v} = \underline{n}(\underline{q},\underline{v}) = [n_i] = [\underline{v}^T N^i(\underline{q})\underline{v}] \qquad ,i = 1,..,n.$$

Equation (1) can be translated into the next state variable differential equation:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} + \underline{k} \qquad (2).$$

Although the equations of motion are complex, nonlinear equations for all but the simplest robots, they have several fundamental properties which can facilitate control system design. It is assumed that the kinematic structure of the manipulator is known, but that the numerical values of some or all of the dynamic robot parameters (such as link masses, moments of inertias, etc.) are unknown. Now, one fundamental property of robot dynamics is that these parameters of interest appear as coefficients in a linear relationship of known functions of the generalized coordinates, so that we may write the dynamic equations (1) as:

$$M(\underline{q})\underline{a} + C(\underline{q},\underline{v})\underline{v} = W(\underline{q},\underline{v},\underline{a})\underline{p} \qquad (3),$$

where

$\underline{p}$ is a [r×1] vector containing known/unknown parameters,
$W(\underline{q},\underline{v},\underline{a})$ is a [n×r] matrix of known functions.

A.2 COMPUTED TORQUE CONTROL.

Computed torque control schemes rely on exact compensation of all nonlinearities in the manipulator system, so that, in the ideal case, the closed–loop system is linear and decoupled.

Using a structure identical to that of the dynamics of the manipulator, the control input is chosen as:

$$\underline{u} = M(\underline{q})\underline{u}_r + C(\underline{q},\underline{v})\underline{v} \qquad (4).$$

Then, by substituting (4) into (1), the problem is reduced to that of controlling the simple system:

$$\underline{a} = \underline{u}_r \qquad (5),$$

since the inertia matrix M is positive definite and therefore invertible.

Expression (5), in turn, represents a set of n decoupled double–integrators, each of which can be controlled independantly by an outer–loop control law with units of the desired acceleration $\underline{a}_d(t)$. This can be defined in terms of a given linear dynamic compensator K(s) as:

$$\underline{u}_r = \underline{a}_d - K(s)\underline{e} \qquad (6),$$

where $\underline{e}(t) = \underline{q}(t) - \underline{q}_d(t)$.

Substituting (6) into (5) leads to the linear error equation: $[\, s^2 I + K(s)\,]\underline{e} = \underline{0}$.

The simplest choice of K(s) in (6) is a PD–compensator: $K(s) = K_V s + K_P$,

which leads to the familiar second order error equation: $\ddot{\underline{e}} + K_V \dot{\underline{e}} + K_P \underline{e} = \underline{0}$.

If the gain matrices K_V and K_P are chosen as diagonal matrices with positive diagonal elements, then the closed–loop system is linear, decoupled and globally stable.

However, this desirable performance is based on the assumption that the values of the parameters appearing in the dynamic model of the control law (4) do match the parameters of the actual manipulator system (1). The major limitation of the computed torque approach is that only estimates of M and C are available in practice, thus we can only apply:

$$\boxed{\underline{u} = \hat{M}(\underline{q})\underline{u}_r + \hat{C}(\underline{q},\underline{v})\underline{v}} \qquad (7),$$

instead of (4), so that we get the closed–loop dynamics

$$\underline{a} = (M^{-1}\hat{M})\underline{u}_r + M^{-1}[\,\hat{C} - C\,]\underline{v} \qquad (8)$$

instead of (5).

Expression (8) shows that the problem is not as simple as (5) looked, and in particular may not be adequately handled by standard linear control techniques.

A.3 ADAPTIVE MANIPULATOR CONTROL.

Imprecise knowledge of manipulator parameters can be solved by application of adaptive control techniques. Most of the MRAC methods, considered in this short review, do rely on nonlinearity compensation in a form as described in the previous section (eventually combinated with a kind of feedback control) plus incorporation of system parameter estimation.

The **adaptive controller design** problem will be:

Given the desired trajectory [$\underline{q}_d$, $\underline{v}_d$ and $\underline{a}_d$], and with some or all manipulator parameters being unknown, derive a control law for the actuator torques/forces $\underline{u}$ and an estimation law for the unknown parameters of $\underline{p}$ such that the manipulator output tracks the desired trajectory after an initial adaptation process:

$$\lim_{t\to\infty} [\,\underline{e}\,] = \lim_{t\to\infty} [\,\underline{q}-\underline{q}_d\,] = \underline{0} \qquad (9)$$

$$\lim_{t\to\infty} [\,\dot{\underline{e}}\,] = \lim_{t\to\infty} [\,\underline{v}-\underline{v}_d\,] = \underline{0} \qquad (10)$$

The global stability of the overall adaptive control scheme of Landau and Horowitz & Tomizuka is based on Popov's Hyperstability theory. It is remarkable, however, that the trend in the other recent works is to utilize Lyapunov's second method.
All of the schemes, described next, insure asymptotic tracking of the desired reference trajectory for all possible initial conditions and with all external signals remaining bounded.

A.3.1 LANDAU (1979).

Landau was one of the first who applied the Hyperstability theory of Popov to the design of MRAC systems. He used the type of MRAC technique called parameter adaptation: the adaptation algorithm adjusts the feedback control gain $K_p(t)$ and the feedforward control gain $K_u(t)$ on–line [see figure [1]]. This, in order to let the closed–loop characteristics of the manipulator closely follow the performance of a reference model. This model, chosen by the designer, specifies the desired closed–loop performance. Landau has chosen a linear time–invariant reference model that must be stable and controllable and which has the same structure as the manipulator system (without the term $\underline{k}$ of equation (2)). The state variable differential equations of the **reference model** are:

$$\dot{\underline{x}}_c = A_c\,\underline{x}_c + B_c\,\underline{u}_r \qquad (11),$$

where

$\underline{x}_c$	is the [2n×1] state vector of the reference model (= the desired trajectory $\underline{x}_d$),
$\underline{u}_r$	is the [n×1] reference input due to the desired performance,
A_c	is the chosen [2n×2n] system matrix of the reference model,
B_c	is the chosen [2n×n] input matrix of the reference model.

The adaptive control law Landau proposed is:

$$\boxed{\underline{u} = -\hat{K}_p\,\underline{x} + \hat{K}_u\,\underline{u}_r} \qquad (12),$$

where

$\hat{K}_p(\underline{e}_x,t)$ is the adjusted feedback control gain,

$\hat{K}_u(\underline{e}_x,t)$ is the adjusted feedforward control gain.

A.3.2 HOROWITZ AND TOMIZUKA (1980).

The overall control system of **Horowitz and Tomizuka** is shown in figure [2]. In the inner–loop MRAC system the adaptation algorithm drives the closed–loop manipulator system to follow the reference model. If this performance equivalence is achieved, the outer–loop PID controller is sufficient to force the response of the adaptive controlled manipulator system towards the desired trajectory.

The reference model, specifying the desired system performance, is chosen to be a double integrator for each degree of freedom:

$$\dot{\underline{q}}_c = \underline{v}_c \qquad \dot{\underline{v}}_c = \underline{a}_c \tag{13}$$

Further, Horowitz and Tomizuka have proposed the following adaptive control law:

$$\boxed{\underline{u} = \hat{M}(\underline{q})\underline{a}_c + \hat{\underline{n}}(\underline{q},\underline{v}) - F_p\underline{e}_c - F_v\dot{\underline{e}}_c} \tag{14}$$

where

$\underline{e}_c(t) = \underline{q}(t) - \underline{q}_c(t)$,

$\dot{\underline{e}}_c(t) = \underline{v}(t) - \underline{v}_c(t)$,

$\underline{a}_c(t)$ is the output of the outer–loop **PID controller**, defined as:

$$\boxed{\underline{a}_c = -\int_{t_0}^{t} K_i\underline{e}\,d\tau - K_p\underline{e} - K_v\dot{\underline{e}}} \tag{15}$$

Each term of $\hat{M}(\underline{q})$ and $\hat{\underline{n}}(\underline{q},\underline{v})$ is adjusted by the adaptation algorithm, in order to obtain:

$$\lim_{t\to\infty}[\,\underline{e}_c\,] = \underline{0} \quad \text{and} \quad \lim_{t\to\infty}[\,\dot{\underline{e}}_c\,] = \underline{0} \tag{16}$$

To show the asymptotic stability of their control scheme, Horowitz and Tomizuka treated the nonlinear, time–varying quantities of $M(\underline{q})$ and $\underline{n}(\underline{q},\underline{v})$ as constants in the stability analysis. Therefore, the underlying assumption was always that the parameter adaptation law is much faster than the manipulator dynamics; i.e. that the manipulator parameter variation is negligible compared with the speed of adaptation.

A.3.3 CRAIG, HSU AND SASTRY (1986).

Craig, Hsu and Sastry present an adaptive version of the computed torque method for robot control. The key point in their paper is the introduction of a parametrization of the dynamic manipulator equations, that yields a linear expression in terms of a suitably selected set of robot and load parameters (equation (3)). Their adaptation law adjusts the unknown, but constant system parameters on–line and uses the latest estimates in the computed torque servo:

$$\underline{u} = \hat{M}(\underline{q})[\,\underline{a}_d - K_p\underline{e} - K_v\dot{\underline{e}}\,] + \hat{\underline{n}}(\underline{q},\underline{v}) \tag{17}$$

whereas the parameter adaptation algorithm is:

$$\dot{\hat{\underline{p}}} = -JW^T(\underline{q},\underline{v},\underline{a})\hat{M}^{-1}(\underline{q})[\,\dot{\underline{e}} + B\underline{e}\,] \tag{18}$$

where J is the [r×r] adaptation gain matrix.

Figure [3] shows the structure of the adaptive computed torque controller of Craig, Hsu and Sastry.

A.3.4 SADEGH AND HOROWITZ (1987).

Refering to the trend in recent work of Craig, Hsu and Sastry (1986), **Sadegh and Horowitz** have been able to remove the slowly time–varying system parameter requirement of Horowitz and Tomizuka (1980), by reparametrizing the nonlinear dynamic manipulator terms as linear functions of unknown but constant parameters (equation (3)), which will be estimated on–line by the parameter adaptation algorithm:

$$\dot{\hat{\underline{p}}} = -KW^T(\underline{q},\underline{v},\underline{v}_c,\underline{a}_c)\dot{\underline{e}}_c \qquad (19).$$

The new adaptive control law will be:

$$\underline{u} = \hat{M}(\underline{q})\underline{a}_c + \hat{n}(\underline{q},\underline{v},\underline{v}_c) - F_v\dot{\underline{e}}_c \qquad (20),$$

where

$$\underline{a}_c = \underline{a}_d - \int_{t_0}^{t} K_i\underline{e}\,d\tau - K_p\underline{e} - K_v\dot{\underline{e}} \qquad (21).$$

Comparing this with the method of Craig, Hsu and Sastry, in the algorithm (19) the acceleration input $\underline{a}_c(t)$ is used instead of the joint accelerations $\underline{a}(t)$ in (18) (which are not measurable in most realistic applications) and no matrix inversion is required.

A.3.5 SLOTINE AND LI (1986).

Craig, Hsu and Sastry (1986) have proposed an adaptive computed torque controller, which, however, requires acceleration measurements and the inversion of the matrix of estimated parameters. This problem is solved by **Slotine and Li** using a natural relationship between the inertia matrix and the Coriolis/centrifgal terms, namely that:

$R = [\,\dot{M}-2C\,]$ is a skew–symmetric matrix
(i.e., that $\underline{x}^T R\underline{x}=\underline{0}$ for all $\underline{x}$, and so $r_{kj} = -r_{jk}$), as can be easily derived from the Lagrangian formulation of the manipulator dynamics .

This property enabled Slotine and Li to define the following adaptive law cq. adaptation algorithm:

$$\underline{u} = \hat{M}(\underline{q})\underline{a}_d + \hat{C}(\underline{q},\underline{v})\underline{v}_d - K_p\underline{e} - K_v\dot{\underline{e}} \qquad (22),$$

$$\dot{\hat{\underline{p}}} = -J^{-1}W^T(\underline{q},\underline{v},\underline{v}_d,\underline{a}_d)\dot{\underline{e}} \qquad (23).$$

However, this adaptive controller does yield zero velocity errors but it may present nonzero position errors. Slotine and Li solved this problem by restricting the residual tracking errors to lie on a **sliding surface**:

$$\underline{s}(t) = \dot{\underline{e}} + \Lambda\underline{e} = \underline{0} \qquad (24),$$

thus guaranteeing asymptotic convergence of the tracking error.

Now, control law (22) and adaptation algorithm (23) are modified into resp.:

$$\underline{u} = \hat{M}(\underline{q})\underline{a}_r + \hat{C}(\underline{q},\underline{v})\underline{v}_r - K_d\underline{s} \qquad (25),$$

$$\dot{\hat{\underline{p}}} = -J^{-1}W^T(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)\underline{s} \qquad (26),$$

where $\underline{s}(t) = \dot{\underline{e}} + \Lambda\underline{e}$,

$\underline{q}_r = \underline{q}_d - \Lambda\int_0^t \underline{e}\,dt \rightarrow \quad \underline{a}_r = \underline{a}_d - \Lambda\underline{\dot{e}}$

A.4 Lyapunov's second method.

To show the global tracking convergence of their adaptive controller, Slotine and Li consider the Lyapunov function candidate:

$$V(t) = \tfrac{1}{2}[\,\underline{s}^T M(\underline{q})\underline{s} + \tilde{\underline{p}}^T J \tilde{\underline{p}}\,] \qquad (27)$$

where $\tilde{\underline{p}}(t)=\hat{\underline{p}}(t)-\underline{p}$ denotes the parameter estimation error vector.

Differentiating V yields:

$$\dot{V}(t) \qquad = \underline{s}^T M\dot{\underline{s}} + \tfrac{1}{2}\underline{s}^T\dot{M}\underline{s} + \tilde{\underline{p}}^T J\dot{\tilde{\underline{p}}} = \qquad (28)$$

$$= \underline{s}^T[\,\underline{u} - C\underline{v} - M\underline{a}_r\,] + \underline{s}^T[\,\tfrac{1}{2}(\dot{M}-2C) + C\,]\underline{s} + \tilde{\underline{p}}^T J\dot{\tilde{\underline{p}}}.$$

Now, Slotine and Li have used the property of skew–symmetry to eliminate the term $\tfrac{1}{2}\underline{\dot{e}}^T(\dot{M}-2C)\underline{\dot{e}}$. With control law (25) $\dot{V}(t)$ becomes:

$$\dot{V}(t) \qquad = \underline{s}^T[\,\hat{M}\underline{a}_r + \hat{C}\underline{v}_r - K_d\underline{s} - C\underline{v} - M\underline{a}_r + C\underline{s}\,] + \tilde{\underline{p}}^T J\dot{\tilde{\underline{p}}} =$$

$$= \underline{s}^T[\,\tilde{M}\underline{a}_r + \tilde{C}\underline{v}_r - K_d\underline{s}\,] + \tilde{\underline{p}}^T J\dot{\tilde{\underline{p}}} =$$

$$= \underline{s}^T[\,W(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)\tilde{\underline{p}} - K_d\underline{s}\,] + \tilde{\underline{p}}^T J\dot{\tilde{\underline{p}}} =$$

$$= -\underline{s}^T K_d\underline{s} + \tilde{\underline{p}}^T[\,W(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)^T\underline{s} + J\dot{\tilde{\underline{p}}}\,] \qquad (29).$$

Finally, Slotine and Li have defined adaptation algorithm (26), such that

$$W(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)^T\underline{s} + J\dot{\tilde{\underline{p}}} = \underline{0}$$

The resulting expression of $\dot{V}$ is: $\qquad \dot{V}(t) = -\underline{s}^T K_d\underline{s} \le \underline{0}.$

This expression shows that the output error converges to the sliding surface $\underline{s}(t) = \underline{0}$, which implies that both the velocity and position tracking errors go to zero.

Substituting control law (25) into the manipulator dynamics (1), one obtains the closed–loop dynamics

$$M(\underline{a}-\underline{a}_r)+(M-\hat{M})\underline{a}_r+C(\underline{v}-\underline{v}_r)+(C-\hat{C})\underline{v}_r+K_d\underline{s}=\underline{0},$$

$$M\dot{\underline{s}} + [\,C + K_d\,]\underline{s} = W(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)\tilde{\underline{p}} \qquad (30).$$

We can conclude that the adaptive robot controller of Slotine and Li consists of a PD feedback part and a full dynamics feedforward compensation part with the unknown manipulator (and load) parameters being estimated on–line.

A.5 Figures.

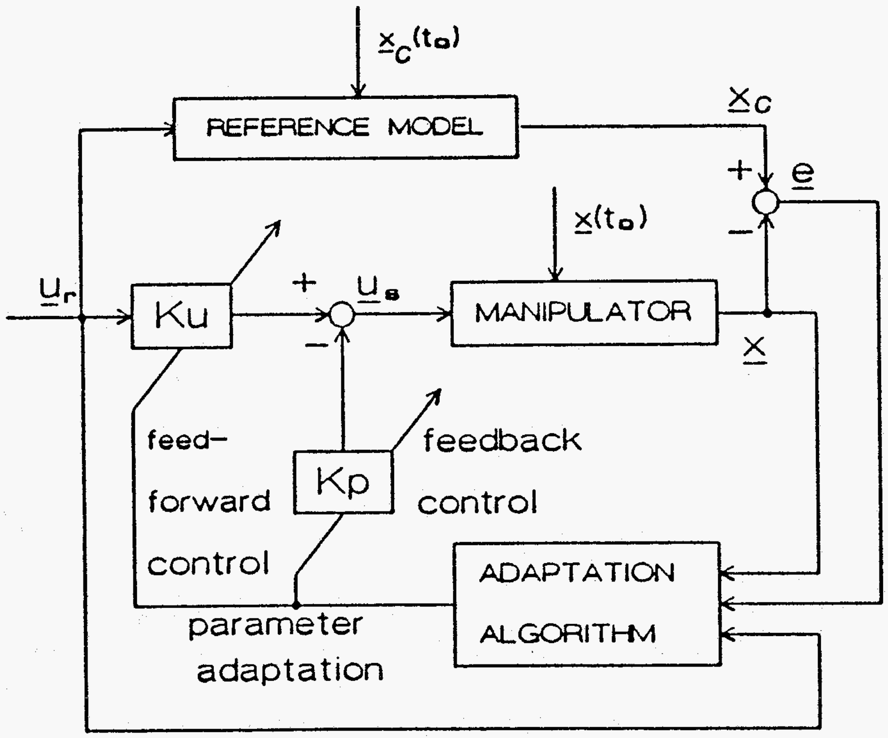


Figure [1]: MRAC system of LANDAU.

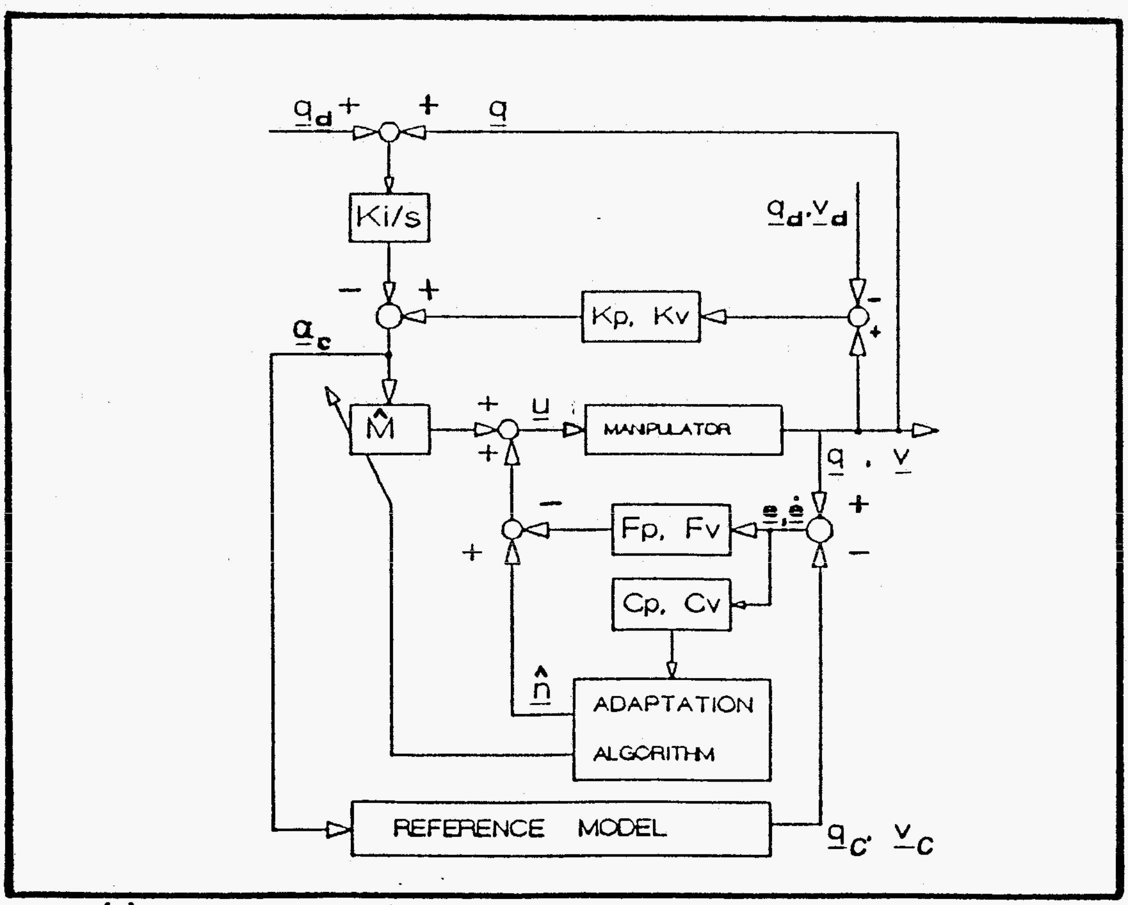


Figure [2]: MRAC scheme of Horowitz and Tomizuka

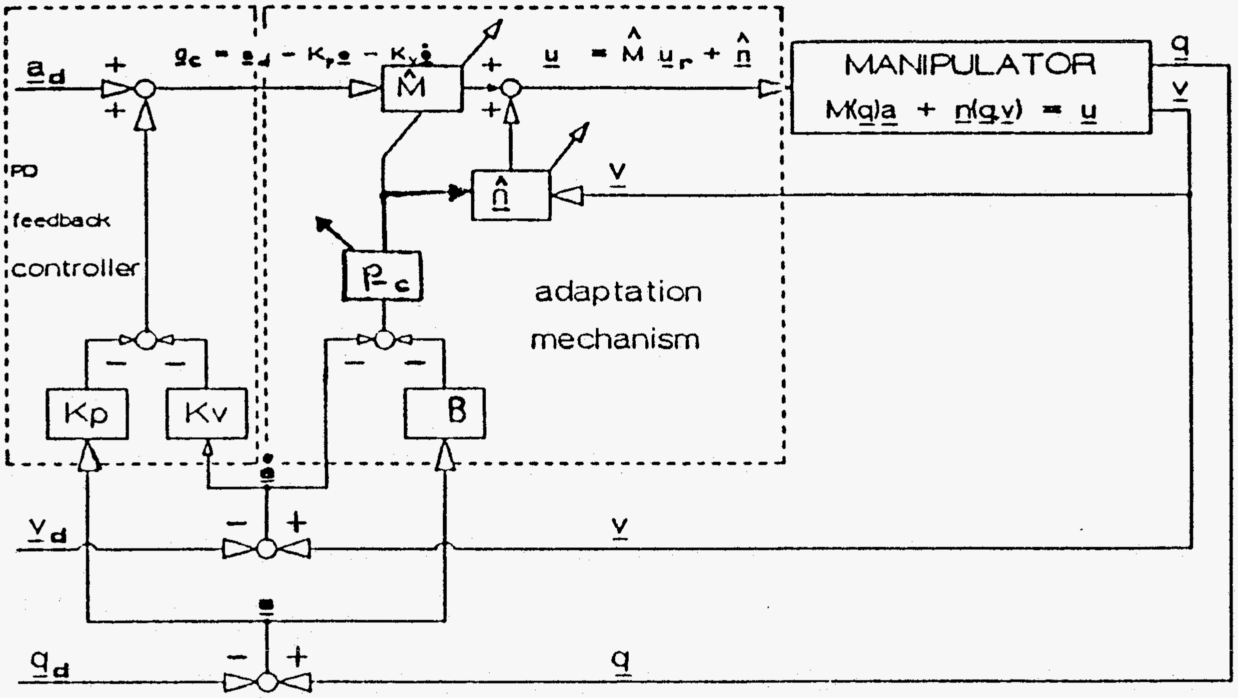
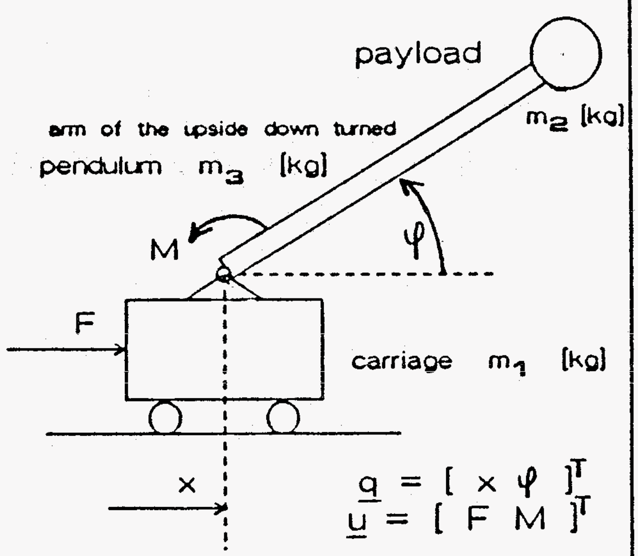
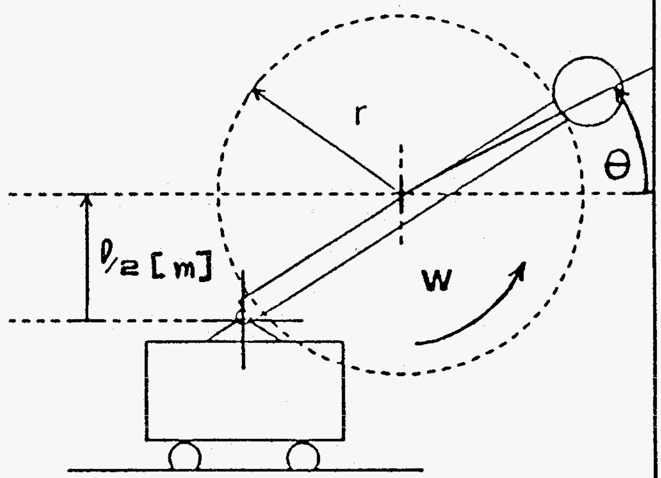


Figure [3]: MRAC scheme of Craig et al.



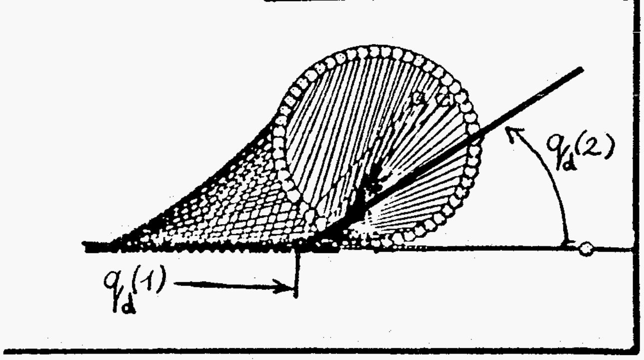
THE NONLINEAR SYSTEM :
a translation / rotation robot.



The desired trajectory of the payload.

A.6 Plots.

Figure [4]: The desired trajectory.



Some simulation results with a translation–rotation robot:

Fixed–parameter control: *figures* [6], [7] and [8].

Figure [5]: PD–controller: $\underline{u} = -K_d\underline{s}$ [K_d=100]

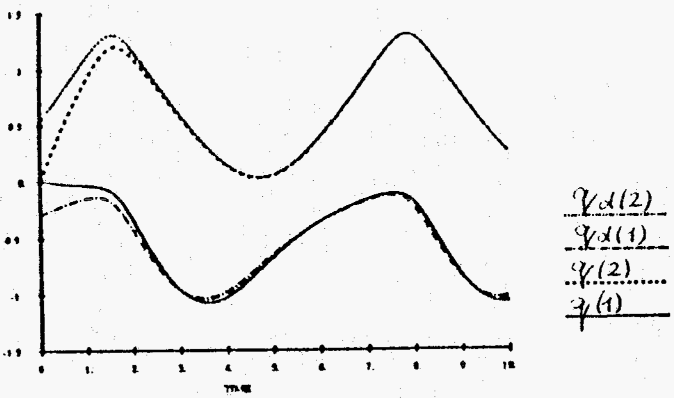


Figure [6] Computed torque with PD feedback
$\underline{u} = M(\underline{q})\underline{a}_d + C(\underline{q},\underline{v})\underline{v}_d - K_d\underline{s}$
[Kd=100].

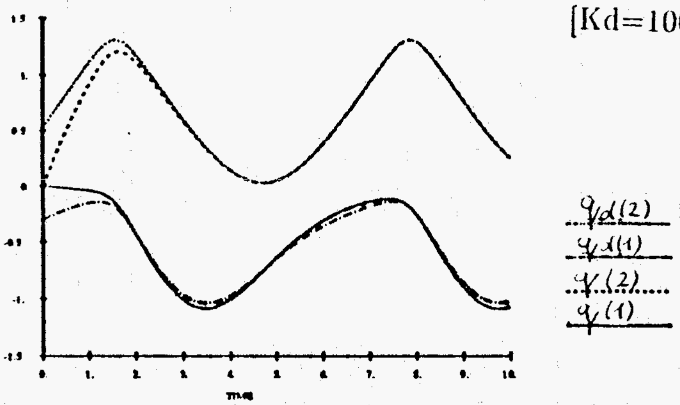


Figure [7]: Computed torque with PD feedback
$\underline{u} = M(\underline{q})[\underline{a}_d - K_p\underline{e} - K_v\dot{\underline{e}}] + C(\underline{q},\underline{v})\underline{v}_d$
[K_p=K_v=100].

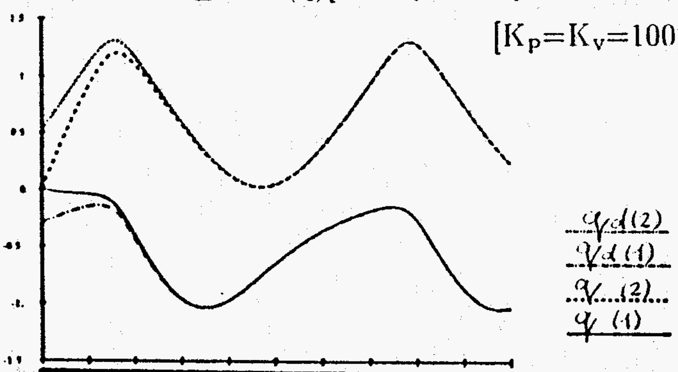
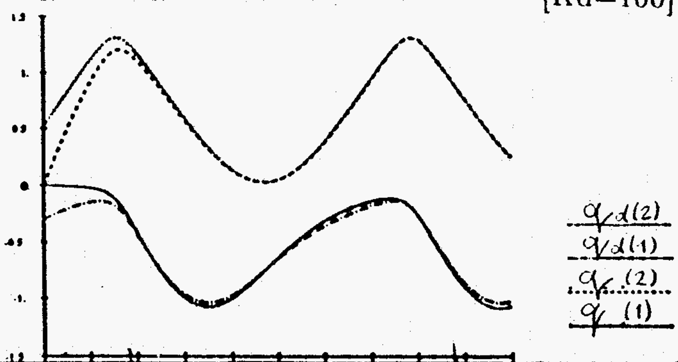
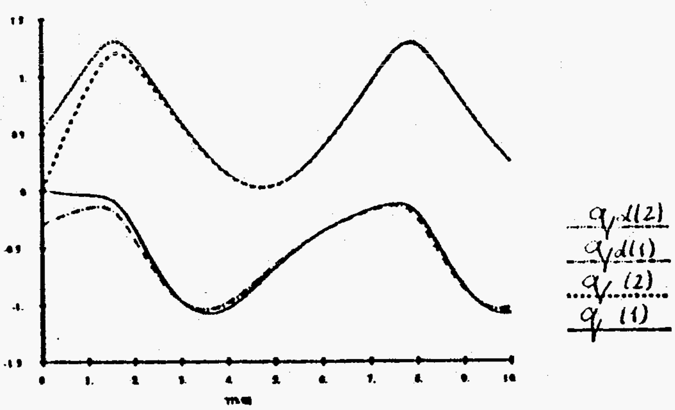


Figure [8]:
As *Fig.*[6] but with $\underline{q}_r(t)$ instead of with $\underline{q}_d(t)$:
$\underline{u} = M(\underline{q})\underline{a}_r + C(\underline{q},\underline{v})\underline{v}_r - K_d\underline{s}$ [Kd=100].



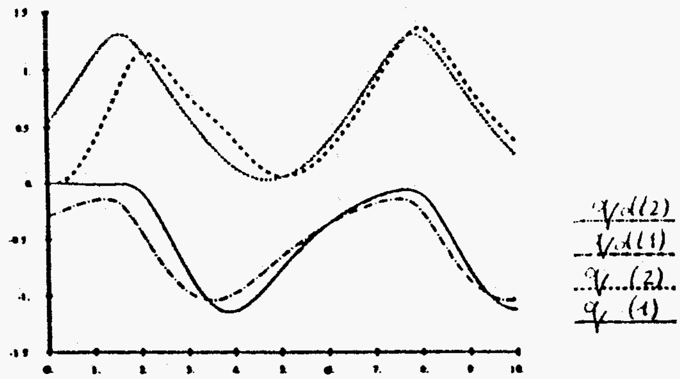
Adaptive control: $\dot{\underline{p}} = -J^{-1}W^T(\underline{q},\underline{v},\underline{v}_d,\underline{a}_d)\dot{\underline{e}}$

Figure [9]:



Adaptive computed torque with PD feedback:
$\underline{u} = \hat{M}(\underline{q})\underline{a}_d + \hat{C}(\underline{q},\underline{v})\underline{v}_d - K_d\underline{s}$.

Figure [10]:



Adaptive computed torque with PD feedback:
$\underline{u} = \hat{M}(\underline{q})[\underline{a}_d - K_p\underline{e} - K_v\dot{\underline{e}}] + \hat{C}(\underline{q},\underline{v})\underline{v}_d$.

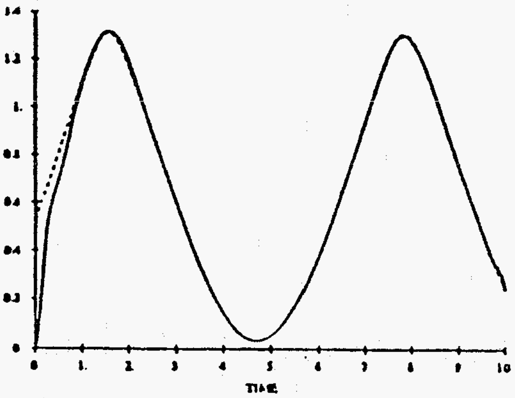


Figure [11]: Landau [$w_0$1,10.0.0.0].

$q_d(2)$
$q(2)$

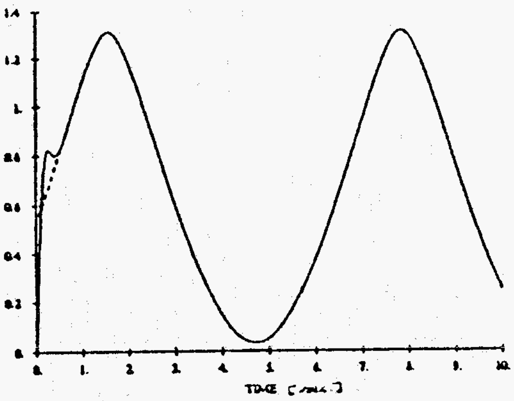


Figure [12]: Horowitz & Tomizuka [g].

$q_d(2)$
$q(2)$

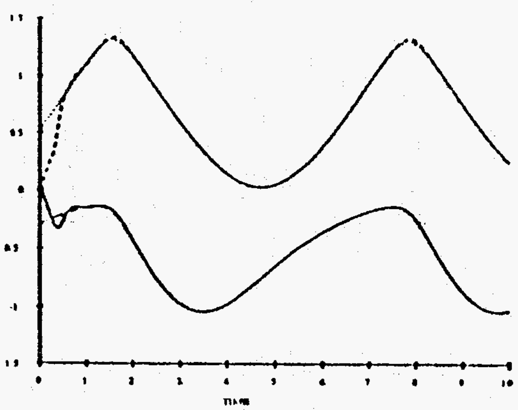


Figure [13]: Craig [smt_0:1.0.1].

$q_d(2)$
$q_d(1)$
$q(2)$
$q(1)$

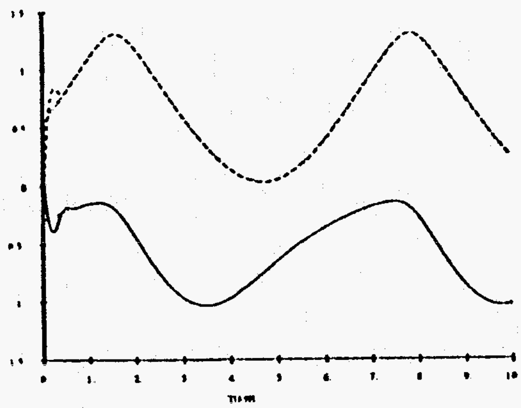


Figure [14]: Sadegh & Horowitz [smt_0:1.0.1].

$q_d(2)$
$q_d(1)$
$q(2)$
$q(1)$

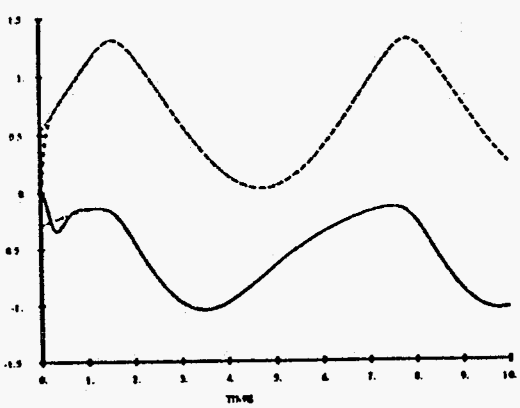


Figure [15]: Slotine & Li [K_d100].

$q_d(2)$
$q_d(1)$
$q(2)$
$q(1)$

B ON THE ADAPTIVE CONTROL OF ROBOT MANIPULATORS

J.–J E. Slotine and W. Li

B.1 INTRODUCTION.

A globally stable adaptive controller for robot manipulators was presented in Craig et al.(1986). The key point in that paper was the introduction of a parametrization of the robot equations that yields a linear expression in terms of a suitably selected set of robot and load parameters. Based on this parametrization an adaptive computed torque controller was proposed. However, it required acceleration measurements and the inversion of the matrix of estimated parameters. Using a natural relationship between the inertia matrix and the Coriolis/centrifugal terms, this problem is solved by **Slotine and Li (1986).**

Slotine and Li have developed a globally asymptotically convergent adaptive controller to control manipulators under certain dynamic uncertainties. Their adaptive robot control algorithm consists of a proportional/differential (PD) feedback part and a full dynamics feedforward compensation part with the unknown manipulator and payload parameters being estimated on–line.

Dynamic model of a robot manipulator.

In the absence of friction or other disturbances,the **dynamic model of a robot manipulator** can be written as:

$$\boxed{\underline{u} = M(\underline{q})\underline{a} + C(\underline{q},\underline{v})\underline{v}} \qquad (1)$$

where

$\underline{u}$ is the [n*1] vector of applied joint torques or forces,
$\underline{q}$ is the [n*1] vector of joint displacements,
$M(\underline{q})$ is the [n*n] symmetric, positive definite inertia matrix,
$\underline{n}(\underline{q},\underline{v})$ is the [n*1] vector of centrifugal, Coriolis gravity and friction torques/forces.

Fundamental properties of manipulator dynamics.

Although the equations of motion (1) are complex, nonlinear equations for all but the simplest robots, they have several **fundamental properties** which can be exploited to facilitate control system design. Two of them are mentioned now:

*) First, Khosla et al. (1985) and Atkeson et al. (1985) have shown that all of the constant parameters of interest such as link masses, moments of inertias, etc., appear as coefficients of known functions of the generalized coordinates. By defining each coefficinet as a separate parameter, a linear relationship results so that we may write the dynamic equations (1) as:

$$\mathbf{M}(\underline{q})\underline{\mathbf{a}} + \mathbf{C}(\underline{q},\underline{v})\underline{\mathbf{v}} = \mathbf{W}(\underline{q},\underline{v},\underline{a})\underline{\mathbf{p}} \qquad (2)$$

where
$\underline{p}$ is a [r*1] vector containing the unknown but constant parameters,
$W(\underline{q},\underline{v},\underline{a})$ is a [n*r] matrix of known functions.

*) Second, as remarked by authors as Arimoto et al. (1984) and Koditschek (1984), the matrix $N = [\,\dot{M}-2C\,]$ is skew-symmetric (i.e., that $\underline{x}^T N \underline{x} = \underline{0}$ for all $\underline{x}$, and so $n_{kj} = -n_{jk}$), as can be easily derived from the Lagrangian formulation of the manipulator dynamics .

Controller design.

The **controller design** problem is as follows:

> Given the desired trajectory, and with some or all manipulator parameters being unknown, derive a control law for the actuator torques/forces and an estimation law for the unknown parameters such that the manipulator output tracks the desired trajectory after an initial adaptation process.

Slotine and Li derive their controller in a few steps:

1. First, in section 1, a simple globally stable adaptive controller is obtained from the Lyapunov stability analysis. The controller strongly exploits the structure of the manipulator dynamics, pointed out in the previous section. However, the adaptive controller does yield zero velocity errors, but it may present nonzero position errors.

2. Slotine and Li solve this problem in section 2 by restricitng the residual tracking errors to lie on a sliding surface, thus guaranteeing asymptotic convergence of the tracking error.

3. Further, in section 3, a sliding control term is incorporated into the control input to make the controller robust either to the uncertainty on parameters not explicitely estimated on–line and to residual time–varying disturbances (such as stiction),

4. Finally, the sliding control term is changed in section 4 into a sonamed saturation control term to avoid control chattering.

B.2 ADAPTIVE COMPUTED TORQUE CONTROLLER WITH PD FEEDBACK.

To derive the control algorithm and adaptation law, Slotine and Li consider the Lyapunov function candidate

$$V(t) = \tfrac{1}{2}[\, \dot{\underline{e}}^T M(\underline{q})\dot{\underline{e}} + \tilde{\underline{p}}^T J \tilde{\underline{p}} + \underline{e}^T K_p \underline{e}\,] \qquad (3)$$

where

$\tilde{\underline{p}}(t)=\hat{\underline{p}}(t)-\underline{p}$ denotes the parameter estimation error vector,
K_p and J are $[r*r]$ symmetric positive definite matrices, usually diagonal,
$\underline{e}(t)=\underline{q}(t)-\underline{q}_d(t)$ is the tracking error.

Differentiating V yields:

$$\dot{V}(t) = \dot{\underline{e}}^T M(d\dot{\underline{e}}/dt) + \tfrac{1}{2}[\, \dot{\underline{e}}^T \dot{M}\dot{\underline{e}}] + \tilde{\underline{p}}^T J \dot{\tilde{\underline{p}}} + \underline{e}^T K_p \dot{\underline{e}} =$$

$$= \dot{\underline{e}}^T[\, \underline{u} - C\underline{v} - M\underline{a}_d\,] + \dot{\underline{e}}^T[\, \tfrac{1}{2}(\dot{M}-2C) + C\,]\dot{\underline{e}} + \tilde{\underline{p}}^T J \dot{\tilde{\underline{p}}} + \dot{\underline{e}}^T K_p \underline{e} =$$

$$= \dot{\underline{e}}^T[\, \underline{u} - C\underline{v}_d - M\underline{a}_d + K_p \underline{e}\,] + \tilde{\underline{p}}^T J \dot{\tilde{\underline{p}}}. \qquad (4)$$

where

Slotine and Li have used the property of skew–symmetry to eliminate the term $\tfrac{1}{2}\dot{\underline{e}}^T(\dot{M}-2C)\dot{\underline{e}}$.

Then Slotine and Li define the following adaptive control law:

$$\boxed{\underline{u}(t) = \hat{M}(\underline{q})\underline{a}_d(t) + \hat{C}(\underline{q},\underline{v})\underline{v}_d - K_p \underline{e} - K_d \dot{\underline{e}}} \qquad (5)$$

where $\hat{M}$ and $\hat{C}$ are the matrices obtained by substituting the known and estimated parameters into M and C.

Now $$\dot{V}(t) = \dot{\underline{e}}^T[\, \tilde{M}(\underline{q})\underline{a}_d + \tilde{C}(\underline{q},\underline{v})\underline{v}_d - K_d \dot{\underline{e}}\,] + \tilde{\underline{p}}^T J \dot{\tilde{\underline{p}}} \quad , \qquad (6)$$

where $\tilde{M}(\underline{q}) = \hat{M}(\underline{q})-M(\underline{q})$,
$\tilde{C}(\underline{q},\underline{v}) = \hat{C}(\underline{q},\underline{v})-C(\underline{q},\underline{v})$.

Further, since the matrices M and C are linear in terms of the manipulator parameters (first mentioned property), we can write

$$\tilde{M}(\underline{q})\underline{a}_d + \tilde{C}(\underline{q},\underline{v})\underline{v}_d = W(\underline{q},\underline{v},\underline{v}_d,\underline{a}_d)\tilde{\underline{p}} \tag{7}$$

and therefore

$$\dot{V}(t) = -\dot{\underline{e}}^T K_d \dot{\underline{e}} + \tilde{\underline{p}}^T [\, J\dot{\tilde{\underline{p}}} + W^T \dot{\underline{e}}\,]. \tag{8}$$

This suggests choosing the following gradient estimator as the adaptation law, such that

$J\dot{\tilde{\underline{p}}} + W^T\dot{\underline{e}} = \underline{0}$ (see e.g. Anderson et al. (1986)):

$$\boxed{\dot{\hat{\underline{p}}} = -J^{-1}W^T(\underline{q},\underline{v},\underline{v}_d,\underline{a}_d)\dot{\underline{e}} \quad .} \tag{9}$$

Note that $\dot{\hat{\underline{p}}}=\dot{\tilde{\underline{p}}}$, since the unknown parameters $\underline{p}$ are constants.

The resulting expression of $\dot{V}$ is $\dot{V}(t) = -\dot{\underline{e}}^T K_d \dot{\underline{e}} \le \underline{0}$. (10)

Therefore the control law (5) and the adaptation law (9) yield a globally stable adaptive controller.

Expression (10) implies that the steady–state joint velocity error goes to zero. However, it does not necessarily guarantee that the steady–state position error is also zero. Slotine and Li now modify the previous adaptive scheme in order to solve this potential problem.

B.3 ELIMINATION OF THE STEADY–STATE POSITION ERRORS.

The undesirable steady–state position errors can be eliminated by restricting them to lie on a **sliding surface**:

$$\underline{s}(t) = \dot{\underline{e}} + \Lambda\underline{e} = \underline{0} \tag{11}$$

where

Λ is a [n*n] constant symmetric positive definite matrix (or more generally, a matrix whose eigenvalues are strictly in the right–half plane).

Formally, this can be achieved by replacing the desired trajectory $\underline{q}_d(t)$ in the above derivation by the virtual 'reference trajectory':

$$\underline{q}_r = \underline{q}_d - \Lambda \int_0^t \underline{e}dt \quad . \tag{12}$$

Accordingly, $\underline{v}_d$ and $\underline{a}_d$ are replaced by

$$\underline{v}_r = \underline{v}_d - \Lambda\underline{e},$$

$$\underline{a}_r = \underline{a}_d - \Lambda\dot{\underline{e}}.$$

Defining $$\underline{s} = \dot{\underline{e}}_r = \underline{v} - \underline{v}_r = \dot{\underline{e}} + \Lambda\underline{e} \quad , \tag{13}$$

control law (5) and adaptation law (9) are modified into

$$\boxed{\begin{aligned} \underline{u}(t) &= \hat{M}(\underline{q})\underline{a}_r(t) + \hat{C}(\underline{q},\underline{v})\underline{v}_r - K_d\underline{s} && (14) \\ \dot{\hat{\underline{p}}} &= -J^{-1}W^T(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)\underline{s} && (15) \end{aligned}}$$

where
- K_d is a [n*n] symmetric positive definite matrix,
- W is now a function matrix of $\underline{v}_r$ and $\underline{a}_r$ instead of $\underline{v}_d$ resp. $\underline{a}_d$, which is defined by the following linearity relation associated with the dynamic model (2):

$$M(\underline{q})\underline{a}_r + C(\underline{q},\underline{v})\underline{v}_r = W(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)\underline{p}.$$

Equation (14) represents a special feedforward plus PD controller, while (15) is a gradient update law.

To show the global tracking convergence of the adaptive controller, consider the Lyapunov function candidate

$$V(t) = \tfrac{1}{2}[\,\underline{s}^T M\underline{s} + \tilde{\underline{p}}^T J\tilde{\underline{p}}\,], \tag{16}$$

instead of (3), which yields (instead of (10)):

$$\dot{V}(t) = -\underline{s}^T K_d \underline{s} \leq \underline{0} \tag{17}$$

Note that control law (14) does not contain a term in K_p, since the position error $\underline{e}$ is already included in $\underline{s}$. Expression (17) shows that the output error converge to the sliding surface $\underline{s}(t)=\underline{0}$. This in turn implies that $\underline{e} \to \underline{0}$ as $t \to \infty$. Thus, the adaptive controller defined by (14) and (15) is globally asymptotic stable and guarantees zero staedy–state position errors, as long as the desired $\underline{q}_d$, $\underline{v}_d$ and $\underline{a}_d$ are bounded.

Substituting the control law (14) into the manipulator dynamics (1), one obtains the closed–loop dynamics

$$M\dot{\underline{s}} + [\,K_d + C\,]\underline{s} = W(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)\tilde{\underline{p}} \tag{18}$$

where $\tilde{\underline{p}}$ is determined by the adaptation law (15).

B.3.1 SLIDING CONTROLLER.

In practice, one may simplify the adaptation algorithm (15) by not explicitely estimating all unknown parameters. Some parameters may have relatively minor importance in the dynamics, in which case one may choose to make the controller robust to the uncertainty on these parameters rather than explicitely estimating them on–line. Similarly, some geometric parameters may already be known with reasonable precision. Further, the controller must be robust to residual time–varying disturbances (such as stiction).

To account for these effects of uncertainties, Slotine and Li have incorporated a **sliding control** term into the control input (14) (see figure [1]):

$$\underline{u}_s = \underline{u} - \underline{k}.\mathrm{sgn}(\underline{s}) \quad , \qquad (19)$$

where

$$\mathrm{sgn}(\underline{s}) = +1 \quad \text{if } \underline{s} > \underline{0},$$
$$\mathrm{sgn}(\underline{s}) = -1 \quad \text{if } \underline{s} < \underline{0}.$$

B.3.2 SATURATION CONTROLLER.

However, the added sliding control term in (19) is discontinuous across the surface $\underline{s}(t)=\underline{0}$, which will lead to control chattering. Chattering of the control input $\underline{u}_s$ is in general highly undesirable in practice, since it involves extremely high control activity and further may excite high–frequency dynamics neglected in the model. Slotine and Li have remedy this situation by smoothing out the control discontinuity. This is achieved by choosing outside a certain boundary B(t) control law $\underline{u}_s$ as before (which guarantees boundary layer attractiveness) and then interpolating $\underline{u}_s$ inside B(t) (see the figure[2]). In other words, the switching function sgn($\underline{s}$) is replaced by the saturation function sat($\underline{s}/\underline{b}$).

$$\underline{u}_s = \underline{u} - \underline{\bar{k}}.\mathrm{sat}(\underline{s}/\underline{b}) \quad , \qquad (19)$$

where

$$\mathrm{sat}(\underline{s}/\underline{b}) = +1 \quad \text{if } \underline{s}/\underline{b} > \underline{\epsilon},$$
$$\mathrm{sat}(\underline{s}/\underline{b}) = \underline{s}/\underline{b} \quad \text{if } -\underline{\epsilon} \le \underline{s}/\underline{b} \le \underline{\epsilon}.$$
$$\mathrm{sat}(\underline{s}/\underline{b}) = -1 \quad \text{if } \underline{s}/\underline{b} < -\underline{\epsilon}.$$

As shown in Slotine (1984), $\underline{s}$ is then guaranteed to converge to the boundary layers with corresponding small tracking errors, and furthermore essentially assigns a lowpass filter structure to the local dynamics of the variable $\underline{s}$, thus eliminating chattering.

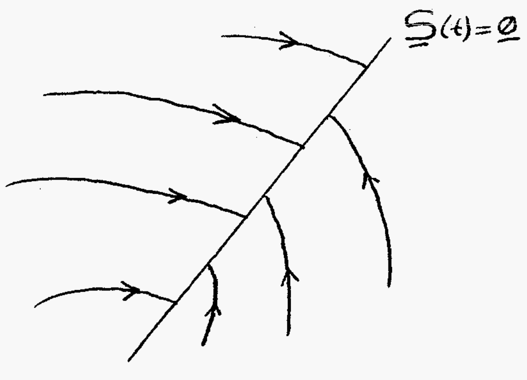


Figure [1]:
Trajectories pointing towards the sliding surface $\underline{s}(t) = \underline{0}$.

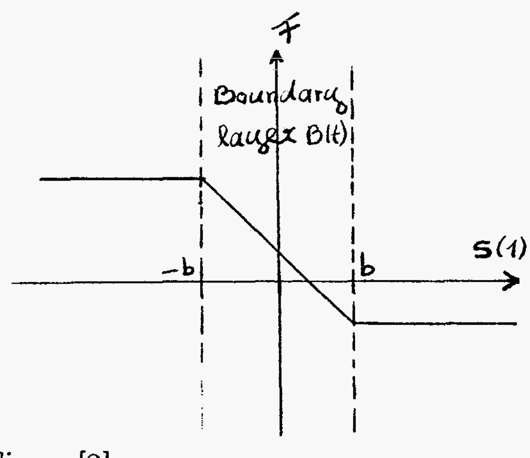


Figure [2]:
Control law interpolation in the boundary layer B(t)

Fixed–parameter control: *figures* [4], [5] and [6].

Figure [3]: PD–controller: $\underline{u} = -K_d\underline{s}$ $[K_d=100]$

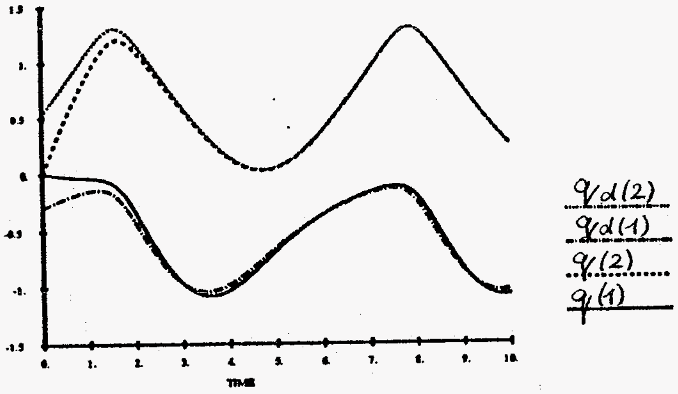


Figure [4]: Computed torque with PD feedback
$\underline{u} = M(\underline{q})\underline{a}_d + C(\underline{q},\underline{v})\underline{v}_d - K_d\underline{s}$
$[Kd=100]$.

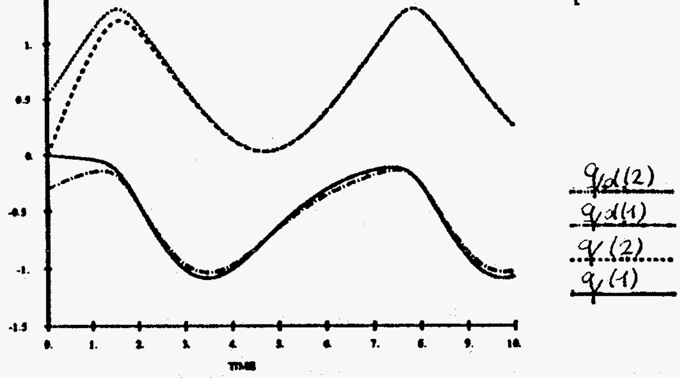


Figure [5]: Computed torque with PD feedback
$\underline{u} = M(\underline{q})[\underline{a}_d - K_p\underline{e} - K_v\dot{\underline{e}}] + C(\underline{q},\underline{v})\underline{v}_d$
$[K_p=K_v=100]$.

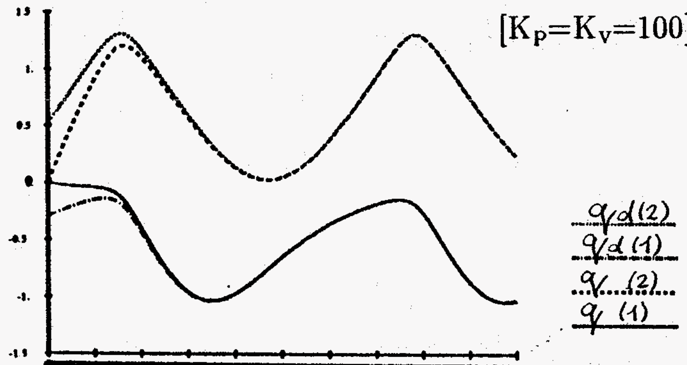
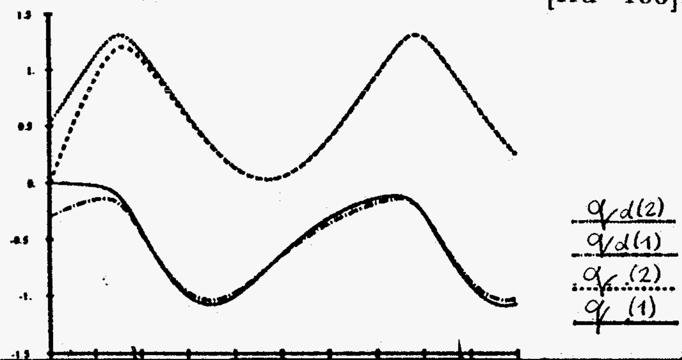


Figure [6]:
As *Fig.*[4] but with $\underline{q}_r(t)$ instead of with $\underline{q}_d(t)$:
$\underline{u} = M(\underline{q})\underline{a}_r + C(\underline{q},\underline{v})\underline{v}_r - K_d\underline{s}$ $[Kd=100]$.



Adaptive control: $\dot{\hat{\underline{p}}} = -J^{-1}W^T(\underline{q},\underline{v},\underline{v}_d,\underline{a}_d)\underline{e}$ (*figures* [7] and [8]).

Figure [7]: Adaptive computed torque with PD feedback:
$\underline{u} = \hat{M}(\underline{q})\underline{a}_d + \hat{C}(\underline{q},\underline{v})\underline{v}_d - K_d\underline{s}$.

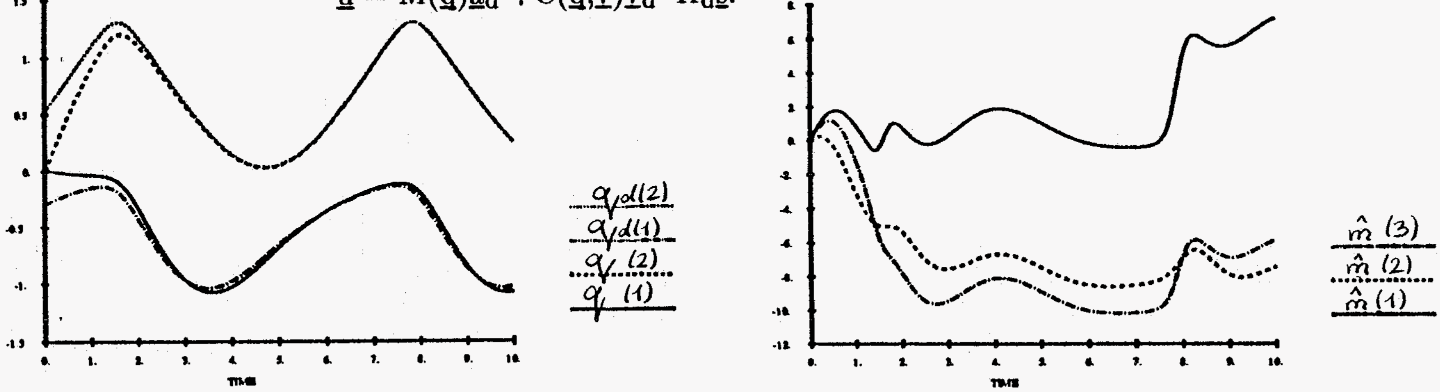
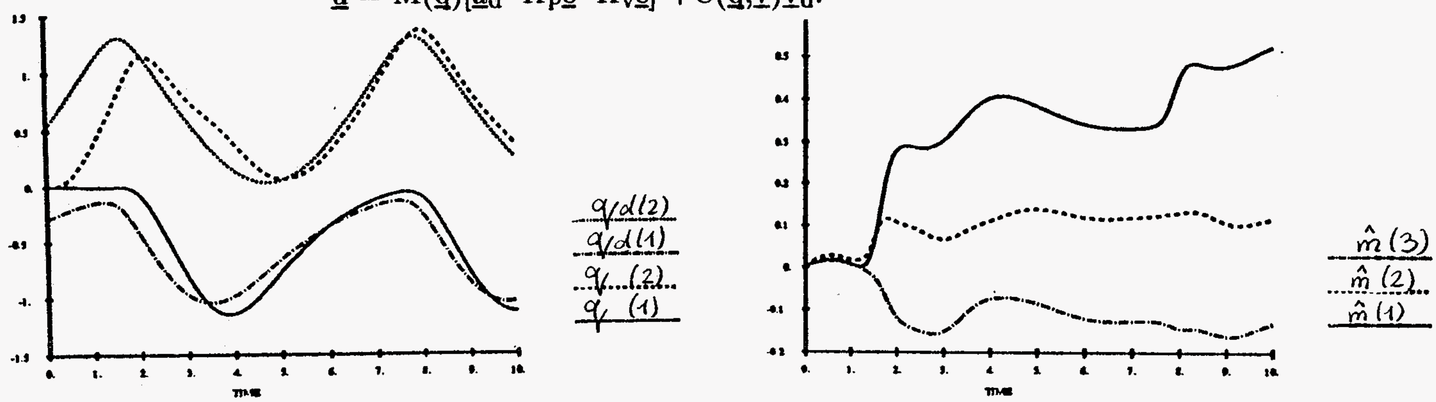


Figure [8]: Adaptive computed torque with PD feedback:
$\underline{u} = \hat{M}(\underline{q})[\underline{a}_d - K_p\underline{e} - K_v\dot{\underline{e}}] + \hat{C}(\underline{q},\underline{v})\underline{v}_d$.



Nonadaptive saturation control: $\underline{u} = M(\underline{q})\underline{a}_r + C(\underline{q},\underline{v})\underline{v}_r - K_d\underline{s} - \underline{k}\,\mathrm{sat}(\underline{s}/\underline{b})$.

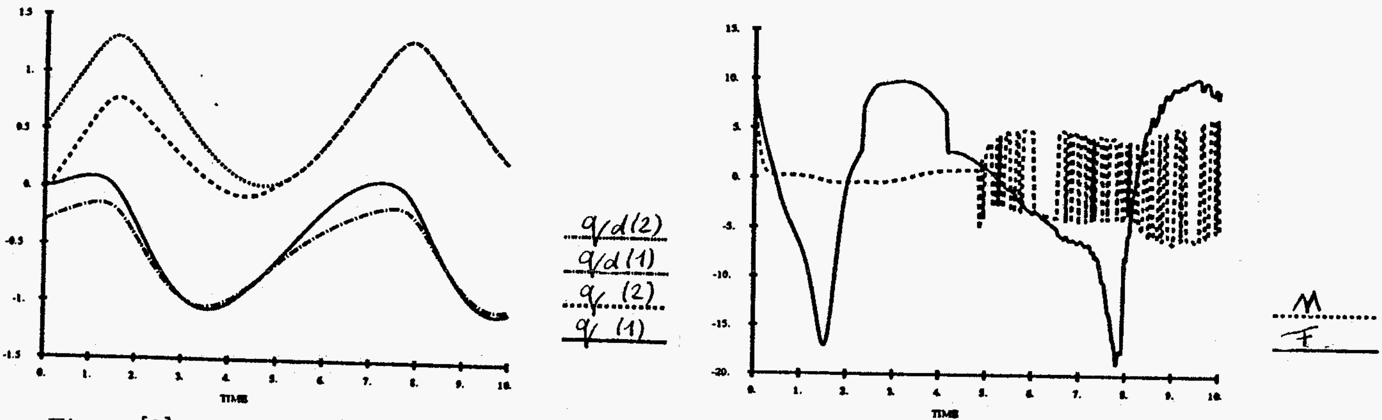


Figure [9]: Saturation control; no PD feedback; computed torque $[K_d=0]$.

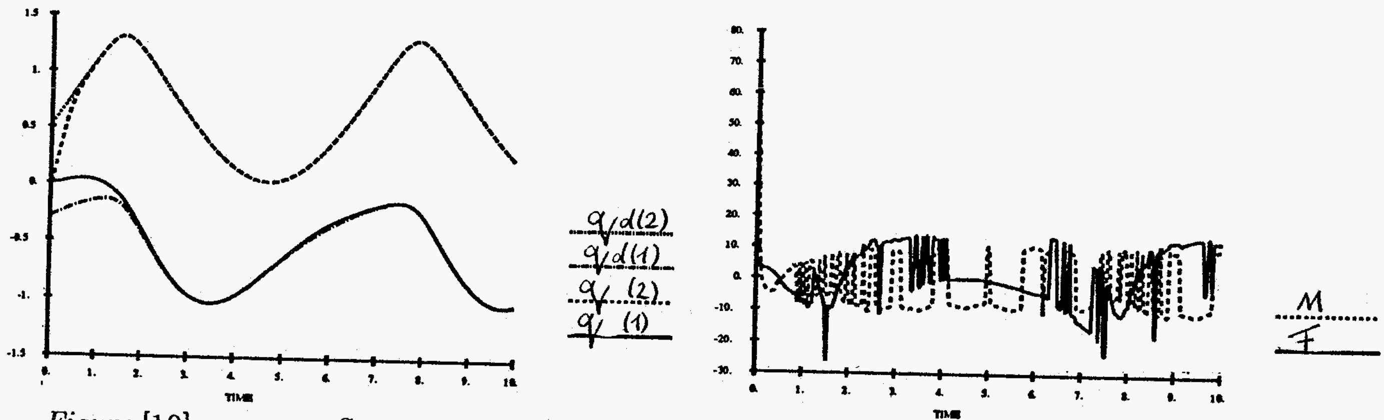


Figure [10]: Saturation control + PD feedback + computed torque $[K_d=10]$.

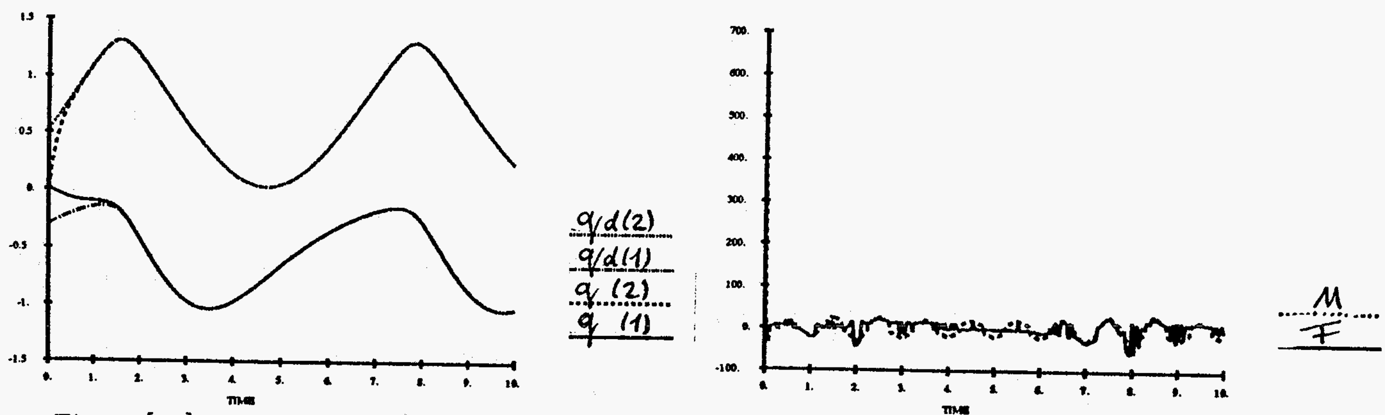


Figure [11]: Saturation control + PD feedback + computed torque $[K_d=100]$.

Adaptive control: $\dot{\hat{\underline{p}}} = -J^{-1}W^{T}(\underline{q},\underline{v},\underline{v}_r,\underline{a}_r)\underline{s}$ (all next *figures*).

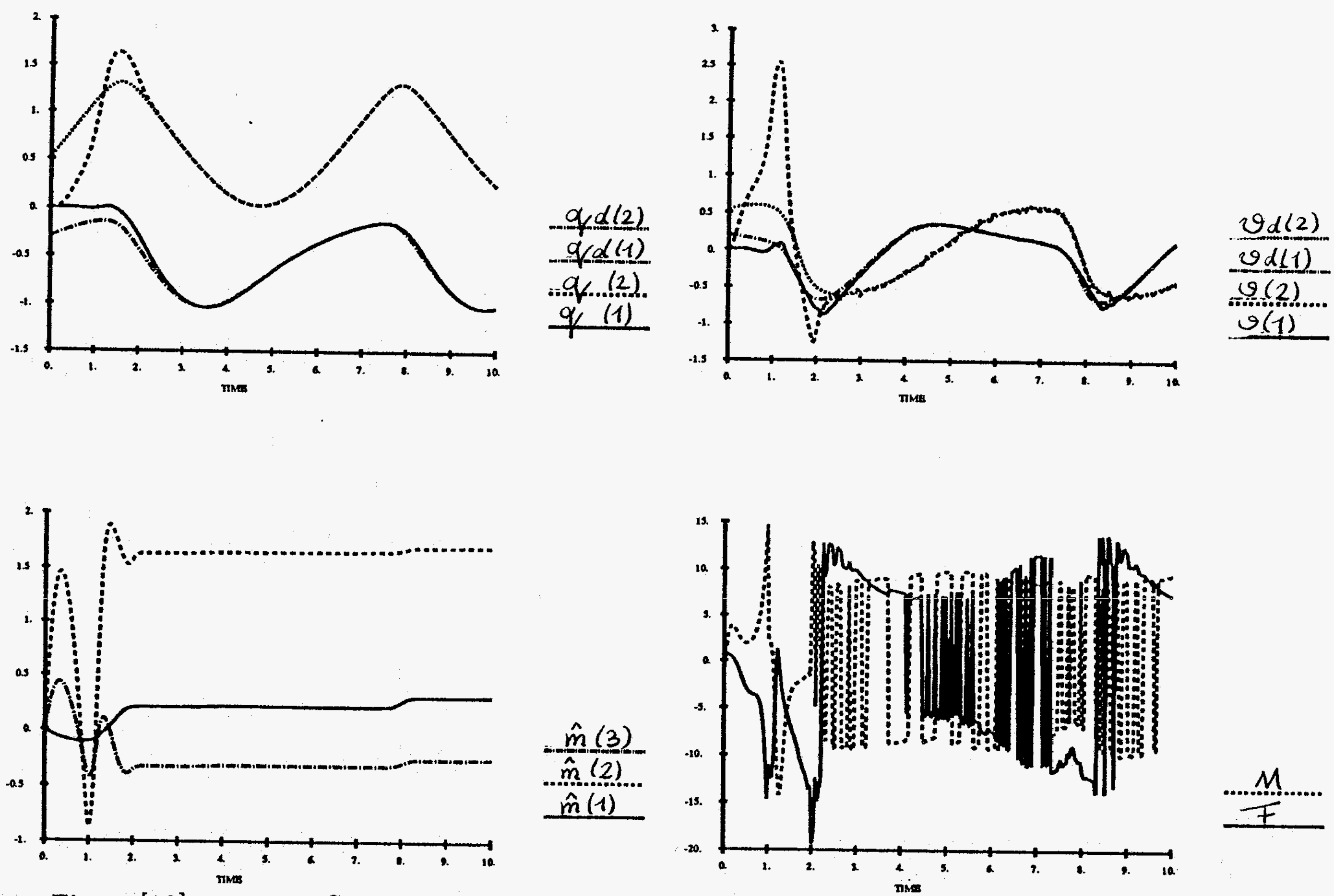


Figure [12]: Saturation control; no PD feedback; ***adaptive*** computed torque.

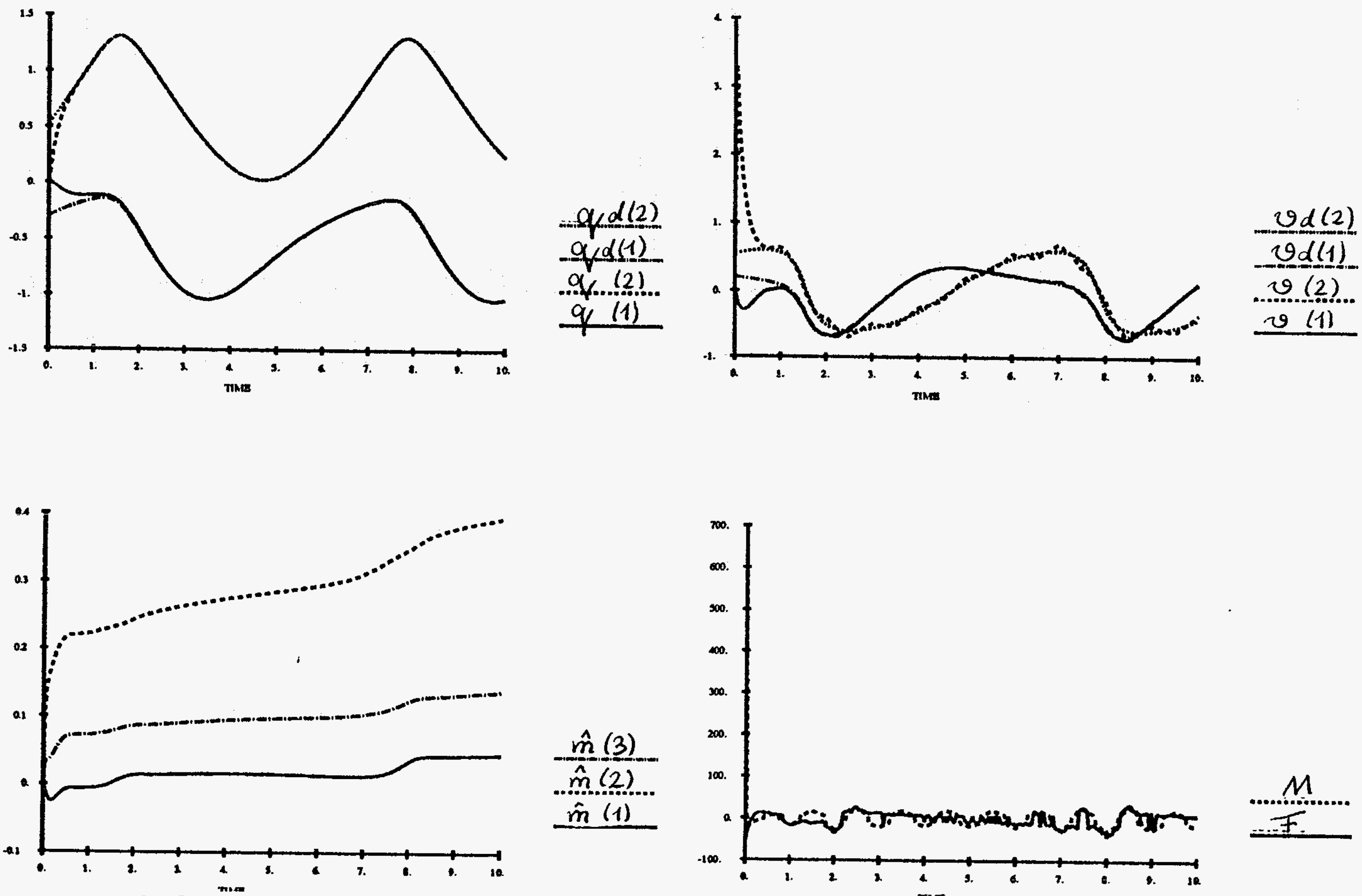


Figure [13]: Saturation control + PD feedback + ***ad. comp. torque*** [K_d=100].

The payload mass is not adjusted on–line: m2 is considered to be 0[kg]. (figures [14], [15], [16], [17])

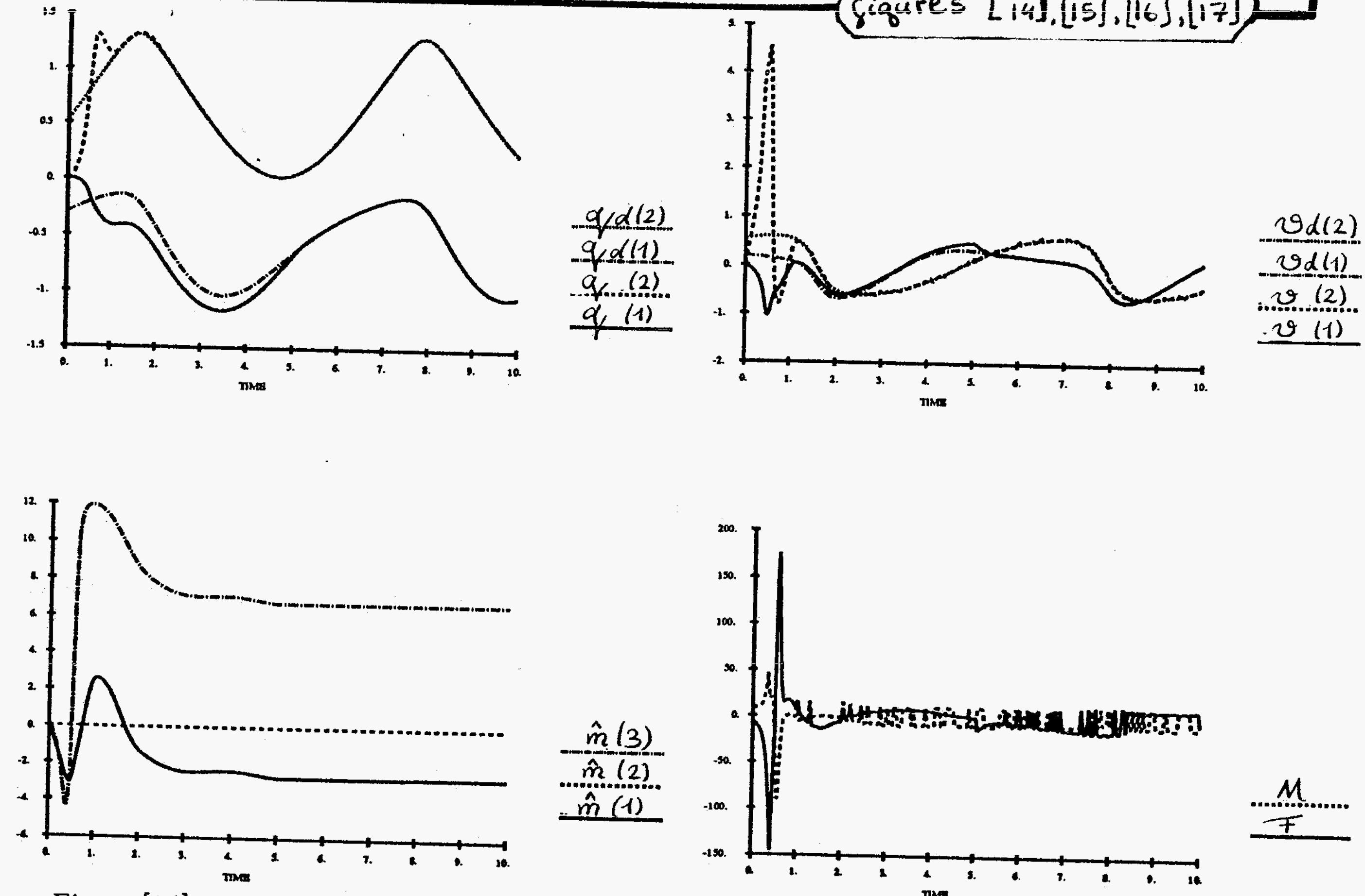


Figure [14]: Sliding control; no PD feedback; **adaptive** computed torque.

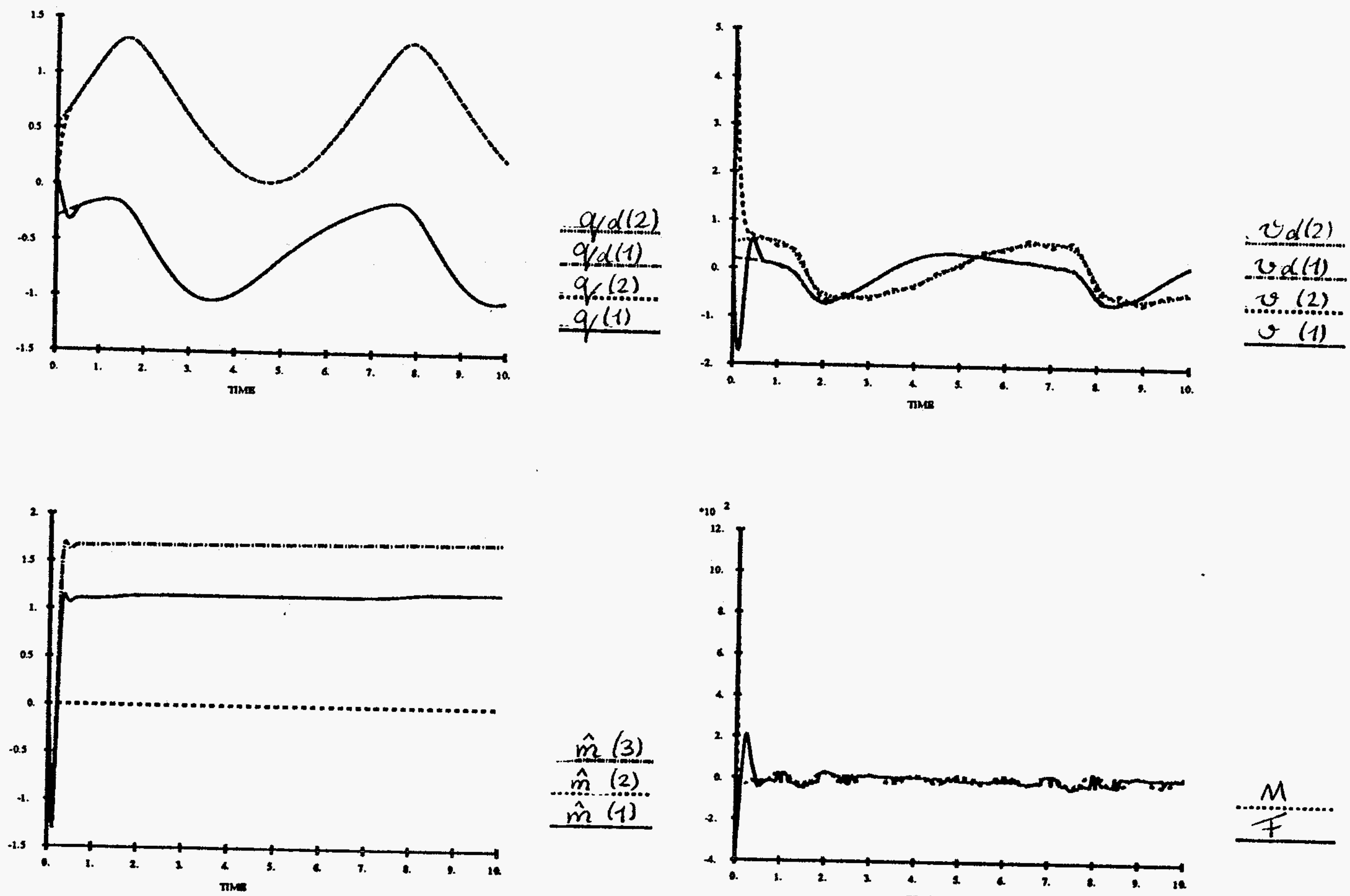


Figure [15]: Sliding control + PD feedback + **ad. comp. torque** [K_d=100].

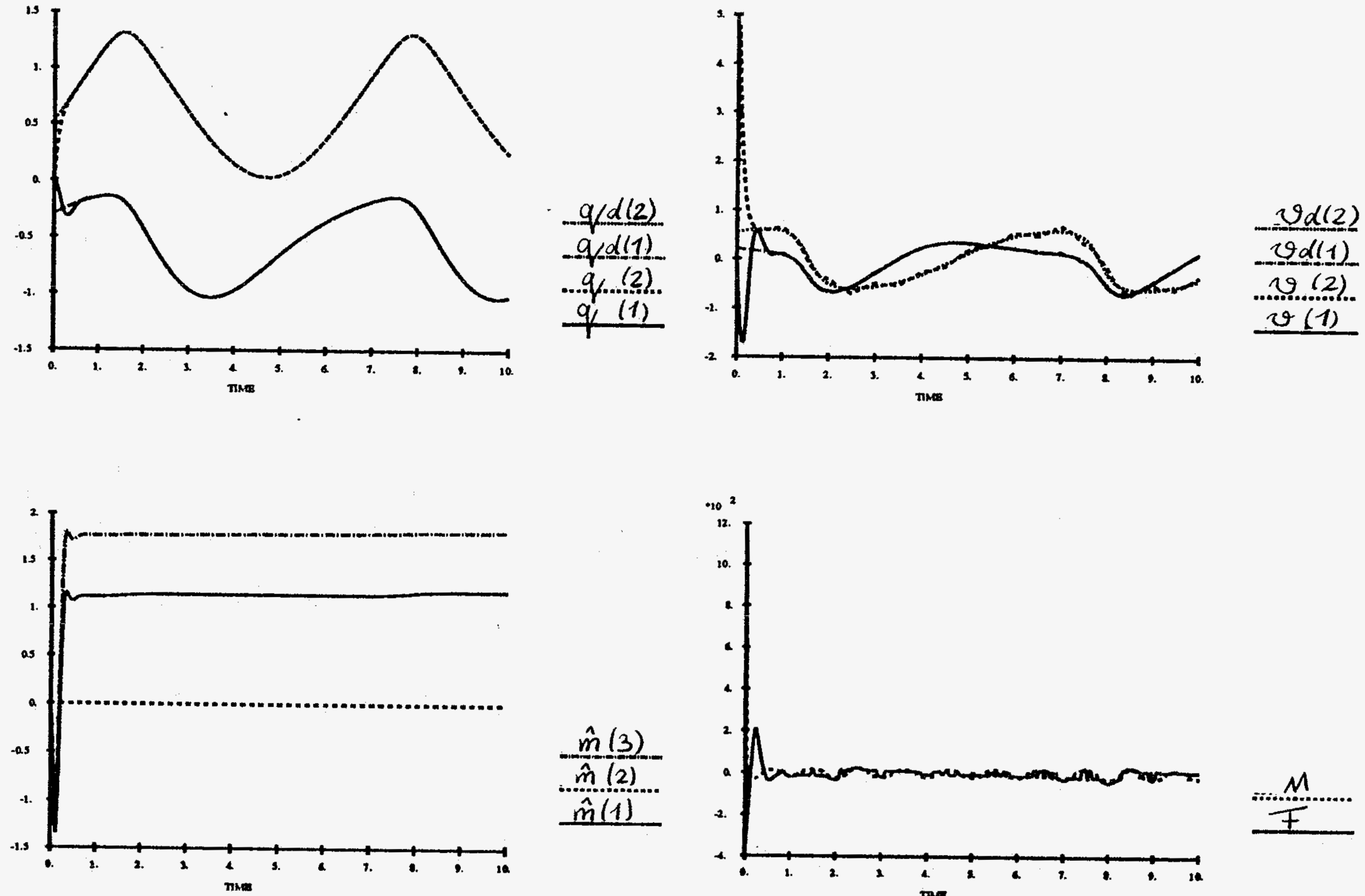


Figure [16]: Saturation control + PD feedback + ad. comp. torque [K_d=100].

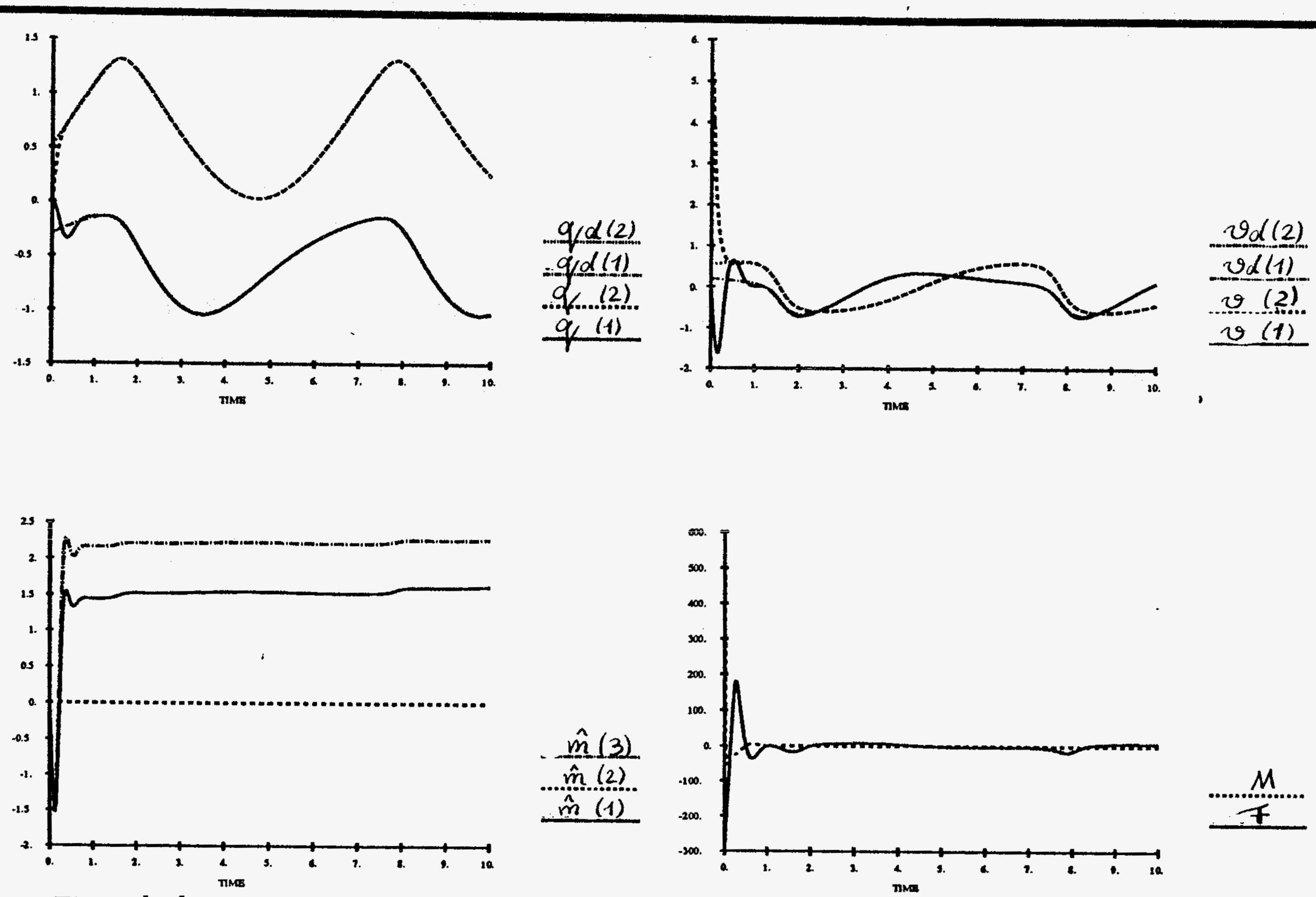
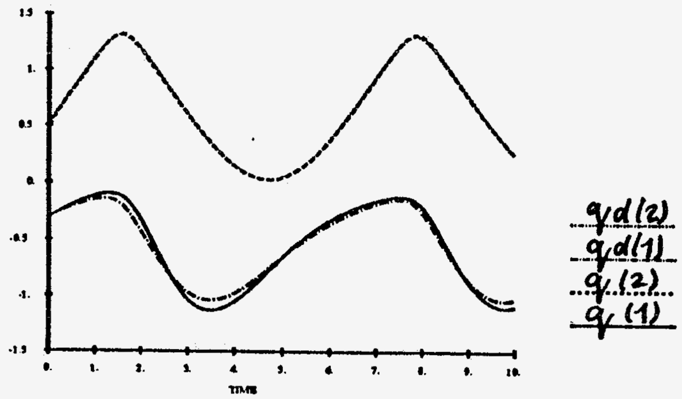
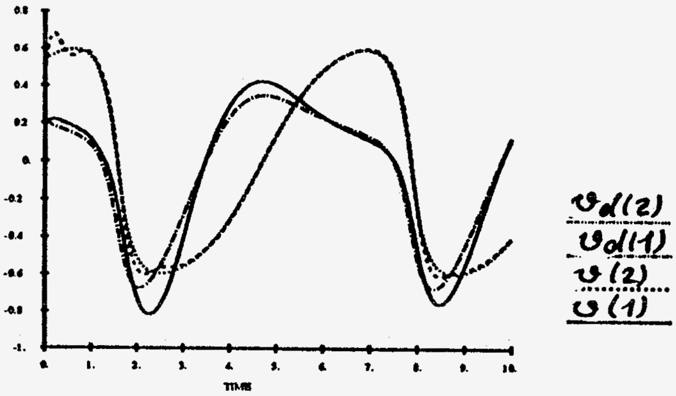


Figure [17]: PD feedback + adaptive computed torque [K_d=100].

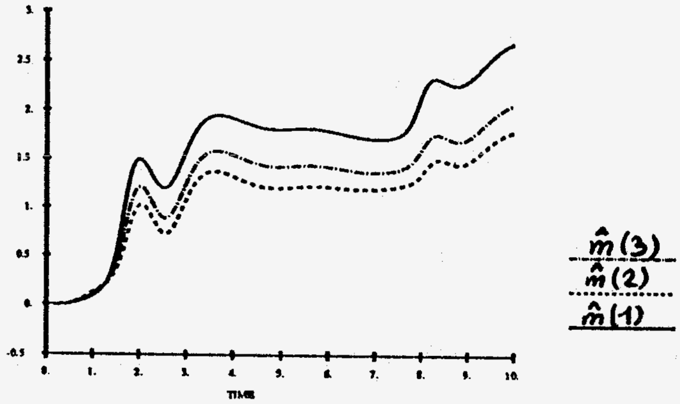
[Ideal initial system state conditions (sic+)]

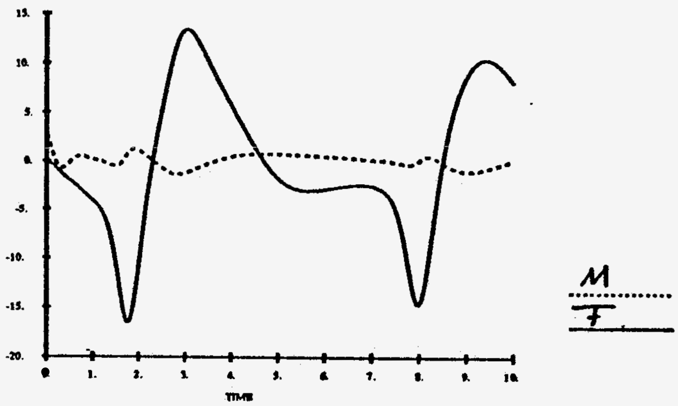
Adaptive computed torque control + PD feedback [Kd=10].



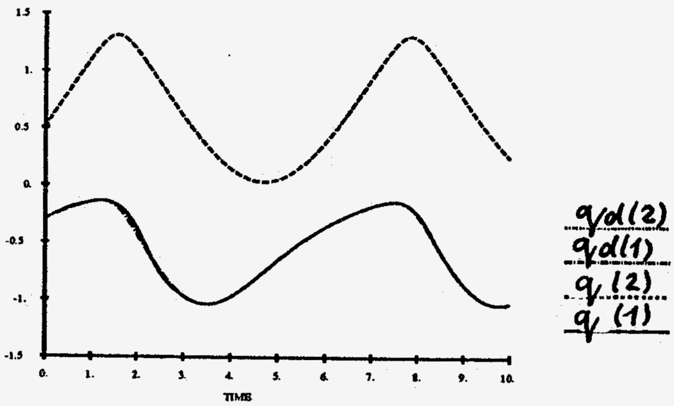


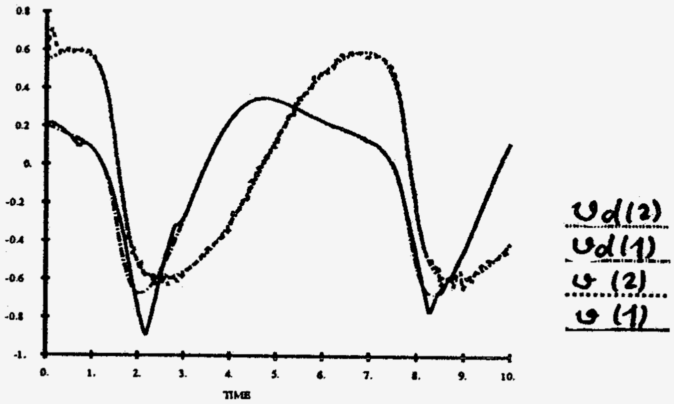
SI#7:Kd10:sic+:85



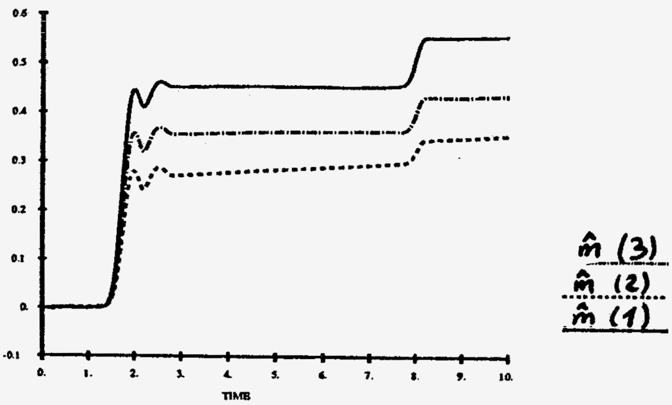


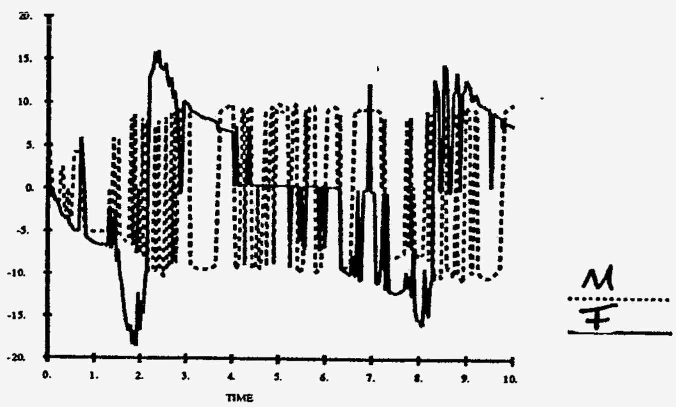
Ad. comp. torque + PD feedback + saturation [b= 0.0005].



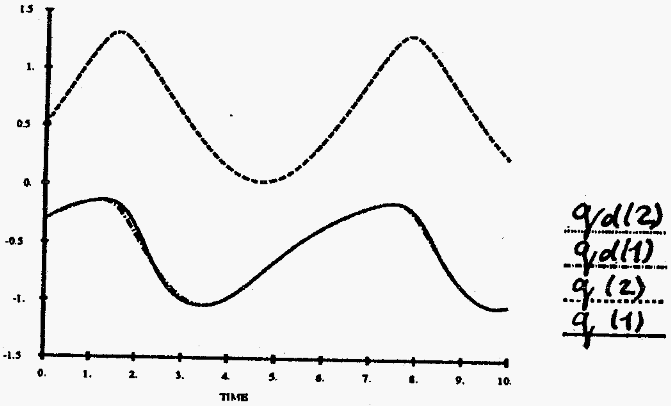


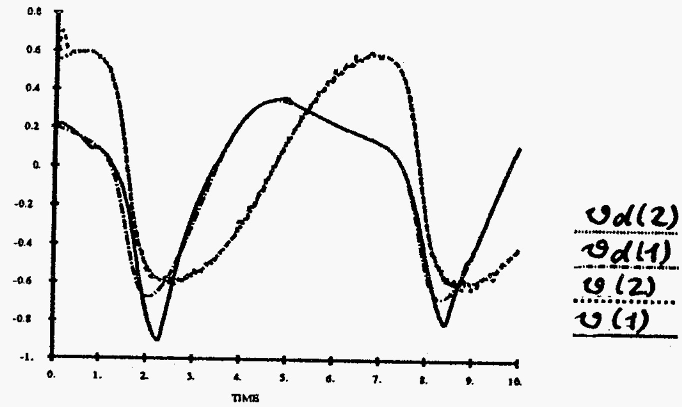
Sat17:Kd10:S1:@0.0005:146



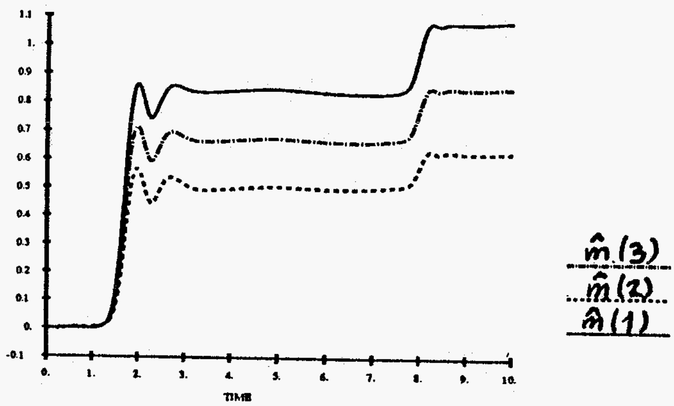


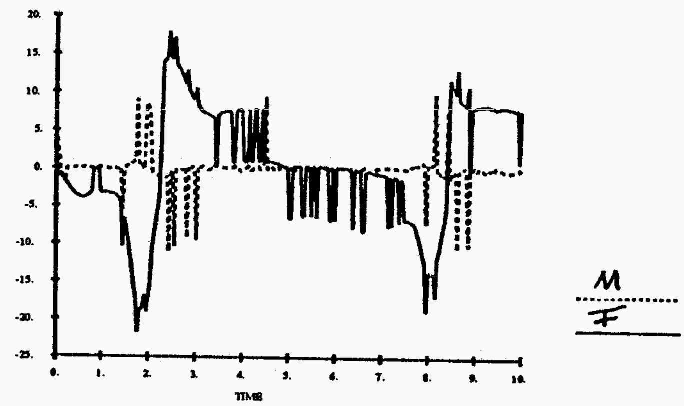
Adaptive computed torque + PD feedback + saturation control [b= 0.005].



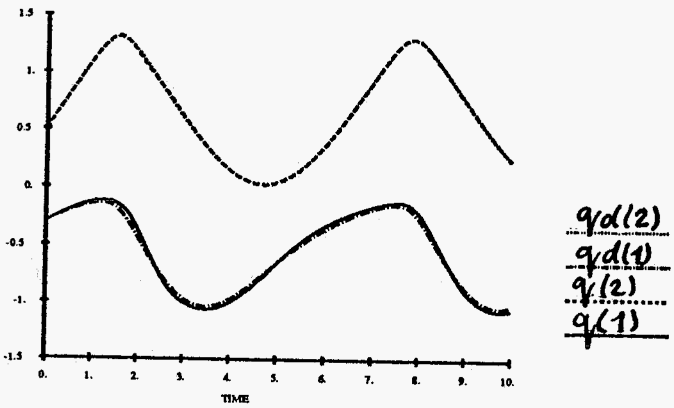


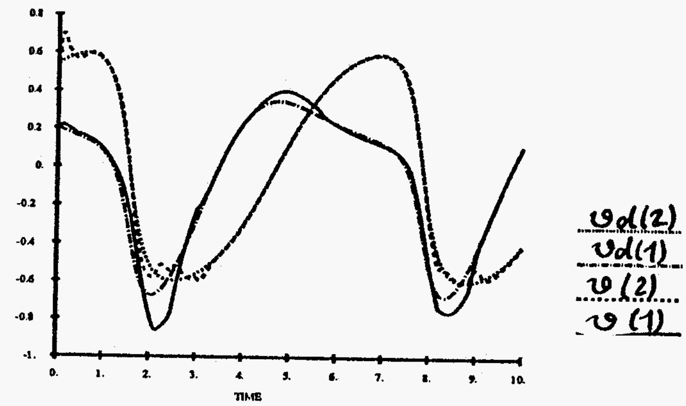
Sat17:Kd10:$1:@0.005:119



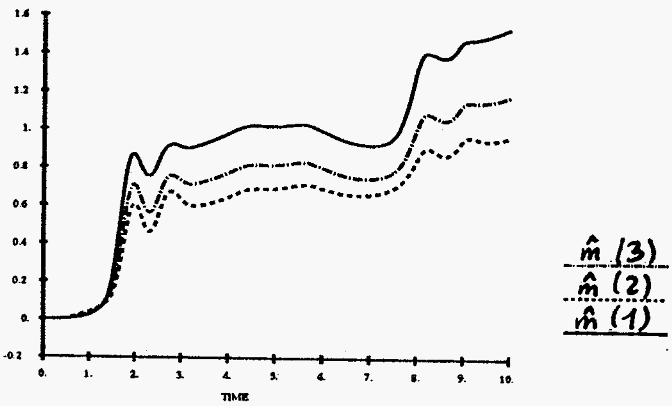


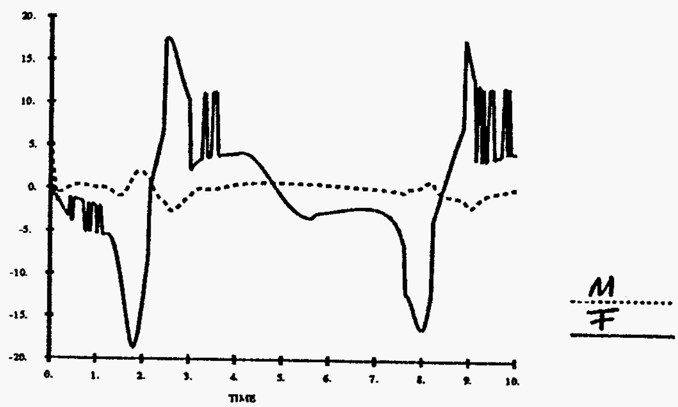
[b=0.05].





7:Kd10:@0.05:labda1:107





C CONTROL OF A FLEXIBLE MANIPULATOR.

C.1 INTRODUCTION.

In the control of robots, besides PID also advanced control techniques as computed torque control and adaptive control have been investigated. However, these control methods appear to be in their origin version only applicable to rigid manipulators. There is needed now an extension for controlling robots with elastic joints and/or flexible links, while the problem of achieving stability is severe.

C.2 RIGID MANIPULATOR CONTROL.

Industrial robots will be of great importance in future. Robots of today are used already for various purposes in different industries. Until now, the control of industrial robots has been studied extensively under the assumption that the (actuator) joints are stiff and that the links can be modelled as rigid bodies. This assumption can be justified for most of today's robots, because of their very heavy construction in order to avoid undesirable positioning inaccuracies that may be caused by elastic deformations and vibrations. The advantage of such stiff constructions is that (angular) encoders at the actuator joints (sonamed 'collocated sensors') can be used to get information about the actual position of the end–effector in space in a purely geometric manner. Therefore, the controller can use this information directly to perform the actuator inputs. The joints are then driven simultaneously, often by a simple PID servo loop. However, the main disadvantage of today's typical stiff robots is that they do relatively slow response.

C.3 FLEXIBLE MANIPULATOR CONTROL.

For higher operating speeds, industrial robots in future shall be made lightweight to reduce the driving torque requirements and to enable the robot arm to respond faster. However, as a consequence of this development, high speed operation leads to high inertial forces which in turn cause considerable elastic deformations of the manipulator members and thus to less end–point positioning accuracy. This makes it necessary to take into consideration the dynamic effects of joint elasticity and (distributed) link flexibility during rapid arm movements by more advanced control algorithms. Therefore, the feedback control system will be equipped with additional sensors giving information about the elastic vibrations to be suppressed or stabilized. Mostly, the control action will then still be carried out by the existing joint actuators (i.e., no additional actuators are used).

The inclusion of the flexible motion in the control action enables to achieve better positioning accuracy with the existing joint actuators. However, theoretically any flexible system has an infinity number of elastic modes, while the limited number of sensors and actuators restricts the controller design to a few critical modes. Therefore, mostly the mathematical robot model will consider only the first eigenmodes of each flexible link (in these papers too).

C.3.1 Computed torque control.

A well known approach to improve the control of robotic manipulators is the computed torque/ inverse dynamics control method. Here, the control law is designed explicitely on the basis of a detailed nonlinear model, in order to compensate the robot nonlinearities and to guarantee a desired closed–loop behavior.

It is well known that the dynamic equations of a rigid robot system may be globally linearized and decoupled by nonlinear feedback. This computed torque control approach transforms the equations of motion of the rigid system into a set of double integrator equations which can then be controlled by adding an 'outer loop' (PID–) control (pole–placement techniques; see for example [1]).
However, the dynamic model of a flexible manipulator is not feedback linearizable in the conventional way as for a rigid robot.

*) The 'rigid' computed torque control technique can be understood as a special case of a more general procedure for transforming a nonlinear system into a linear system, which is known as **external/ feedback linearization** ([2]) and leads to on of the possible approaches in controlling elastic robots (see also references [3], [4], [5], [6] and [7])

— The remarkable result obtained with pole–placement control of a feedback linearized system is that the closed–loop system has a desired behavior in the whole state space.

— However, the feedback linearization technique appears to be computationally expensive in general and requires accurate modeling and full state measurements.

*) A second alternative mentioned in literature for flexible robot control is found in utilizing the concept of **integral manifold** to the equivalent **singular perturbation model of the flexible robot** ([8]). It has been shown that the reduced flexible system obtained then is indeed feedback linearizable.

A short review is given in chapter *E*: 'Two–time scale sliding control of a flexible manipulator — Slotine and Hong'.

C.3.2 Adaptive control.

The use of a computed torque control model requires accurate knowledge of the physical manipulator parameters and its payload or is only meaningful if it is possible to identify the model parameters with satisfactory accuracy. This, because instability of the control algorithm will occur in case of parametric uncertainty. With off–line identification strategies there will always remain the question wether the obtained estimated parameter values are validate for a variety of different desired trajectories or just for one. Otherwise, one would prefer to identify the model parameters for control purposes on–line, because then they can be adjusted at any time for each arbitrary reference trajectory and never need to converge to certain constant values.

An **adaptive control** approach seems to be of great relevance for the control of systems with unknown or time–varying parameters or even with an unknown part of dynamics. A possible way to handle with parametric uncertainties is to implement an adaptive computed torque control law as follows:

— first adopt a suitable (linearization of the) robot model,
— then perform on–line estimation of the model parameters,
— finally apply the computed torque control law with the adjusted parameters.

The adaptive algorithms may for example be derived from Lyapunov global stability considerations or from the Hyperstability theory of Popov [9].

C.3.3 Sliding control.

Finally, it will be necessary to assure robustness to the effects of uncertainties of those model parameters not estimated on–line, unmodelled dynamics and disturbances, for example by incorporating a sliding control term into the control input (Slotine and Li [10]). However, since sliding control gives rise to discontinuous signals (i.e., chattering occurs), one must care about the admissable inputs to the system and probably has to use the boundary layer approach of Asada and Slotine [11] (which gives, however, less tracking accuracy: see chapter *B*: 'On the adaptive control of robot manipulators – Slotine and Li'). Because the physical constraints on the available motor power limit the extension of the actuator inputs, for practical implementation it would perhaps be possible in some way to use the generalization approach of the computed torque control strategy as has been presented by Heeren [10] (see a short desription of it in chapter *D*: 'Generalized computed torque control of a flexible manipulator')

References:

[1] **Kokotovic, P.V.** (1985).
Recent trends in feedback design: an overview.
Automatica, Vol.21, No.3, pp 225–236, 1985.

[2] **Nijmeijer, H. and Van der Schaft, A.J.** (1990).
Nonlinear dynamical control systems.
Springer–Verlag, New–York, 1990.

[3] **Cesareo, G. and Marino, R.** (1984).
On the controllability properties of elastic robots.
6th Int. Conf. on Analysis and Optimization of Systems, INRIA, Nice, 1984.

[4] **Spong, M.W.**
(1985) Robust stabilization for a class of nonlinear systems.
Proc. 7th Int. Symp. on the Math. Theory of Network and Systems, (MTNS) Stockholm, Springer–Verlag, june 1985.

[5] (1987) Modeling and control of elastic joint robots.
Journal of Dynamic Systems, Measurement and Control, Vol. 109, pp 310–319, 1987.

[5] **Spong, M.W. and Vidyasagar, M.** (1985).
Robust nonlinear control of robot manipulators.
Proc. 24th IEEE Conf. on Decision and Control, Fort Lauderdale, pp 1767–1772, dec 1985.

[7] **Khorasani, K. and Kokotovic, P.V.** (1985).
Feedback linearization of a flexible manipulator near its rigid body manifold.
Systems and Control Letters, Vol. 6, pp 187–192, 1985.

[8] **Slotine, J.–J. E. and Hong, S.** (1987)
Two–time scale sliding control of manipulators with flexible joints.
American Control Conference, BSE: DPC 82 AME.

[9] **Popov, V.M.** Hyperstability of control system. (1969)
New York, Springer–Verlag.

[10] **Slotine, J.–J. E. and Li, W.** (1987)
On the adaptive control of robot manipulators.
Proc. ASME Winter Annual Meeting.

[11] **Asada, H. and Slotine, J.–J. E.** (1986)
Robot analysis and control.
John Wiley & Sons.

D GENERALIZED COMPUTED TORQUE CONTROL OF A FLEXIBLE MANIPULATOR.

D.1 MANIPULATOR DYNAMICS.

It has been shown that a fairly general model for a manipulator has a nonlinear structure of the following kind:

$$\boxed{\mathbf{M}(\underline{q})\underline{\ddot{q}} + \mathbf{C}(\underline{q},\underline{\dot{q}})\underline{\dot{q}} = \mathbf{H}\underline{\mathbf{u}}} \qquad (1)$$

where

$\underline{q}(t)$	is the [n×1] vector of independent degrees of freedom of the manipulator model,
$\underline{u}(t)$	is the [m×1] vector of actuator inputs.

D.2.1 MANIPULATOR CONTROL.

The main control objective is usually to make the manipulator's end–effector follow some desired path in space. Frequently, it follows from the manipulator design that the number k of output quantities $\underline{y}(t)$ is equal to the number m of servomotor input variables $\underline{u}(t)$. $\underline{y}(t)$ Determines the end–effector position and orientation and depends on $\underline{q}(t)$:

$$\underline{y} = \underline{y}(\ \underline{q}(t)\ ) \qquad (2)$$

If the desired path for $\underline{y}$ is specified by a known, time dependent function $\underline{y}_d(t)$, then the main objective is to let the tracking error $\underline{y}(t)-\underline{y}_d(t)$ tend to zero. A more complete formulation may also include desires about the derivative of $\underline{y}(t)$.

D.2.2 COMPUTED TORQUE CONTROL.

Often, the computed torque control law for a manipulator is chosen as follows:

$$\underline{u} = H^{-1}\{\ M[\ \underline{\ddot{q}}_d + K_v\underline{\dot{e}} + K_p\underline{e}\ ] + C\underline{\dot{q}}\ \} \qquad (3)$$

where

$\underline{q}_d(t)$	is the [n×1] vector of generalized coordinates due to the desired trajectory,
$\underline{e}(t)$	is the [n×1] tracking error: $\underline{e} = \underline{q} - \underline{q}_d$.

However, if a flexible link manipulator has to be controlled, $\underline{q}_d(t)$ cannot be determined explicitely from equation (2), because n > k. Therefore, if we still decide to use control law (3), this is only possible when assumptions have been made about the behavior of [n–k] variables of $\underline{q}_d(t)$ (for example, that the flexible state variables and their derivatives have to remain zero). This does not indicate that this is an optimal choice (see for more about the reference trajectory: W. Winkelmolen [1987]).

D.3 COMPUTED TORQUE CONTROL WITH DESIRED OUTPUT.

Differentiating equation (2) twice leads to an equation of the next form:

$$\ddot{\underline{y}} = \dot{E}_1(\underline{q},\dot{\underline{q}})\dot{\underline{q}} + E_1(\underline{q})\ddot{\underline{q}} \qquad (4).$$

The computed torque control law (3) will be alternated now with the desired output path $y_d(t)$ into:

$$\boxed{\underline{u} = [E_1 M^{-1} H]^{-1} \{\ddot{\underline{y}}_d + K_v \dot{\underline{e}}_y + K_p \underline{e}_y + E_1 M^{-1} C \dot{\underline{q}} - \dot{E}_1 \dot{\underline{q}}\}} \qquad (5)$$

where $\underline{e}_y(t)$ is the [k×1] tracking output error: $\underline{e}_y = \underline{y} - \underline{y}_d$.

This computed torque control law concentrates only on the main control objective to let the tracking (output) error and its derivative tend to xero. A prove of the stability in the case of a rigid manipulator (n = k) is given by Asada and Slotine [1985]. However, for flexible manipulator models (n > k) only the stability of the output coordinates $\underline{y}(t)$ can be proven, whereas it is in general impossible to prove the stability of all terms of $\underline{q}(t)$. It is even possible to find flexible manipulator models for which state–instability occurs, while the output remains stable (M. Tijdink [1989]). This is a serious disadvantage of the computed torque control concept in combination with flexible manipulators.

D.4 GENERALIZED COMPUTED TORQUE CONTROL.

A generalization of the computed torque control strategy has been presented by T. Heeren [1989], to take into account the desire that each term of $\underline{q}$ and its derivative remains bounded. This control objective can be mathematically formulated in $\underline{z}(t)$; the generalized coordinate $q_i(t)$ can be bounded by introducing a penalty function that produces large values when q_i is out of range:

$$z_q(i) = \begin{cases} B_{qi}(q_i - q_i[\max])^2 & \text{if } q_i > q_i[\max], \\ 0 & \text{else}, \\ B_{qi}(q_i - q_i[\min])^2 & \text{if } q_i > q_i[\min]. \end{cases} \qquad (6)$$

The value of each term of $\underline{z}$ is considered to be an output quantity, so the number of output quantities is now larger than the number of inputs. We define for example:

$$\underline{z} = \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \\ \underline{z}_3 \end{bmatrix} = \begin{bmatrix} \underline{y} - \underline{y}_d \\ \underline{z}_q \\ \underline{z}_{\dot{q}} \end{bmatrix} \qquad (7).$$

Differentiating (7) yields: $$\ddot{\underline{z}} = \dot{E}\dot{\underline{q}} + E\ddot{\underline{q}} - \ddot{\underline{y}}_{d00} \qquad (8).$$

$\underline{z}$ Will be equal to zero as long as the manipulator is in the desired working area. As soon as the tracking error $\underline{e}$ is unequal to zero [$\underline{z}_1 \neq 0$], or as soon as terms of $\underline{q}$ and their derivatives reach the given bounds [$\underline{z}_2 \neq 0$ resp. $\underline{z}_3 \neq 0$], control actions $\underline{z}_d(t)$ will force them to zero.

In this way it is obvious that we want to obtain:

$$\underline{\mathbf{d}} = \ddot{\underline{\mathbf{z}}} - \underline{\mathbf{z}}_{\mathbf{d}} = \underline{\mathbf{0}} \qquad (9),$$

where $\underline{z}_d(t)$ can be chosen as:

$$\underline{z}_d = \begin{bmatrix} K_v\dot{\underline{e}}_y + K_p\underline{e}_y \\ K_b\underline{z}_q \\ K_{bd}\underline{z}_{\dot{q}} \end{bmatrix} \qquad (10).$$

Consider the main control objective: $\underline{d}_1 = \ddot{\underline{z}}_1 - \underline{z}_{d1} = \underline{0}$.
Combinating this with equations (1) and (4) will lead to control law (5).

Unfortunately, it is in general impossible to choose $\underline{u}$ such that equality (10) holds, because the dimension of $\underline{z}$ is larger than the dimension of $\underline{u}$. Hence, all that can be done is to minimize some norm of $\underline{d}$. Because there is further the desire to keep the force/torque inputs $\underline{u}(t)$ bounded too, the minimization of $\underline{d}$ may be combined with the minimization of $\underline{u}$. This can be realized by minimizing a scalar function J, which is a kind of respone quality functional and uses positive weighting matrices W and R in order to denote the relative importance of each objective:

$$J = \underline{d}^T W \underline{d} + \underline{u}^T R \underline{u} \qquad (11).$$

A minimum of J with respect to $\underline{u}$ can be found by requiring that the derivative of J with respect to $\underline{u}$ is equal to zero. This results in the following generalized computed torque control law:

$$\boxed{\underline{u} = [\, Z^T W Z + R\,]^{-1} Z^T W \{\ddot{\underline{y}}_{d00} + \ddot{\underline{z}}_d + EM^{-1}C\dot{\underline{q}} - \dot{E}\dot{\underline{q}}\,\}} \qquad (12)$$

where $Z = EM^{-1}H$.

D.5 COMPUTED TORQUE WITH SLIDING CONTROL.

In this section, the computed torque control law (5) will be described through the sliding control approach of Slotine and Li [1986].

A switching surface, which zeros the tracking error and its derivative, is defined as:

$$\underline{s}(t) = \dot{\underline{e}}_y - \Lambda\underline{e}_y = \underline{0} \qquad (13).$$

The time derivative of the switching surface along a trajectory of the system can be expressed as:

$$\begin{aligned} \dot{\underline{s}} &= \ddot{\underline{e}}_y - \Lambda\dot{\underline{e}}_y = \\ &= \ddot{\underline{y}} - \ddot{\underline{y}}_d - \Lambda\dot{\underline{e}}_y = \\ &= \dot{E}_1\dot{\underline{q}} + E_1\ddot{\underline{q}} - \ddot{\underline{y}}_d - \Lambda\dot{\underline{e}}_y = \\ &= \dot{E}_1\dot{\underline{q}} + E_1M^{-1}H\underline{u} - E_1M^{-1}C\dot{\underline{q}} - \ddot{\underline{y}}_d - \Lambda\dot{\underline{e}}_y \end{aligned} \qquad (14)$$

The equivalent input to the system when it is in sliding mode is:

$$\underline{u}_{eq} = [E_1 M^{-1} H]^{-1} \{ \ddot{\underline{y}}_d + \Lambda \dot{\underline{e}}_y + E_1 M^{-1} C \dot{\underline{q}} - \dot{E}_1 \dot{\underline{q}} \} \qquad (15)$$

This in turn leads to $\dot{\underline{s}}(t) = \ddot{\underline{e}}_y - \Lambda \dot{\underline{e}}_y = \underline{0}$ (16),

which [with (13)] means that both $\underline{e}(t)$ and $\dot{\underline{e}}(t)$ asymptotically tend to zero when the system is in sliding mode; i.e., the sliding mode guarantees total stability of the system.

It is the objective that the trajectories of the system beginning from any initial condition are attracted towards the switching surface $\underline{s}(t)=\underline{0}$ in a finite time and thereafter maintain in sliding mode. The control law satisfying the reaching condition

$$\underline{s}^T \dot{\underline{s}} < 0 \qquad (17),$$

may have many forms and can for example be chosen as follows:

$$\boxed{\underline{u} = \underline{u}_{eq} - [E_1 M^{-1} H]^{-1} \underline{k} \, \mathrm{sgn}(\underline{s})} \qquad (18).$$

Substituting this $\underline{u}$ in equation (14) yields:

$$\dot{\underline{s}} = E_1 M^{-1} H \, [E_1 M^{-1} H]^{-1} \{ \ddot{\underline{y}}_d + \Lambda \dot{\underline{e}}_y + E_1 M^{-1} C \dot{\underline{q}} - \dot{E}_1 \dot{\underline{q}} - k \, \mathrm{sgn}(\underline{s}) \} + \dot{E}_1 \dot{\underline{q}} + - E_1 M^{-1} C \dot{\underline{q}} - \ddot{\underline{y}}_d - \Lambda \dot{\underline{e}}_y \ .$$

$$\dot{\underline{s}} = -\underline{k} \, \mathrm{sgn}(\underline{s}) \qquad (19).$$

This satisfies the reaching condition (17) and thus guarantees that the system will reach the sliding surface in finite time. Then, during sliding mode, the system is insensitive to parameter uncertainties and disturbances.

To decrease the chattering of the input signal, Slotine and Li propose utilizing the boundary layer approach of Asada and Slotine [1985], in which the switching function is replaced by a saturation function:

$$\boxed{\underline{u} = \underline{u}_{eq} - [E_1 M^{-1} H]^{-1} \underline{k}_{sat} \, \mathrm{sat}(\underline{s}/\underline{\Phi})} \qquad (20).$$

In literature, sometimes the switching function is embedded in the robust controller design of a stabilizer for parametric uncertainties, disturbances and damping of elastic oscillations:

$$\underline{u} = \hat{\underline{u}}_{eq} + \underline{u}_{stab} =$$
$$= \hat{\underline{u}}_{eq} - [E_1 \hat{M}^{-1} H]^{-1} \underline{v} - P\hat{B}_f \underline{q}_f \qquad (21),$$

where
$\hat{}$ means that the value of the function with
$\hat{}$ is not necessarily equal to the actual value,
$\underline{q}_f$ represents the flexible motion of the robot
(this last term in (21) is here not considered further),
$\underline{v}$ is a new input to the system
(some indications how to define it are given now).

Substituting equation (21) without its last term into (14) gives:

$$\underline{\dot{s}} = E_1M^{-1}H\,\{\,\underline{\hat{u}}_{eq} - [E_1\hat{M}^{-1}H]^{-1}\,\underline{v}\,\} +$$
$$+\,\dot{E}_1\underline{\dot{q}} - E_1M^{-1}C\underline{\dot{q}} - \underline{\ddot{y}}_d - \Lambda\underline{\dot{e}}_y =$$
$$= E_1M^{-1}H\,[E_1\hat{M}^{-1}H]^{-1}\{\,\underline{\ddot{y}}_d + \Lambda\underline{\dot{e}}_y + E_1\hat{M}^{-1}\hat{C}\underline{\dot{q}} - \dot{E}_1\underline{\dot{q}} - \underline{v}\,\} +$$
$$+\,\dot{E}_1\underline{\dot{q}} - E_1M^{-1}C\underline{\dot{q}} - \underline{\ddot{y}}_d - \Lambda\underline{\dot{e}}_y + \underline{v} - \underline{v} =$$
$$= (\,[E_1M^{-1}H][E_1\hat{M}^{-1}H]^{-1} - I\,)\{\,\underline{\ddot{y}}_d + \Lambda\underline{\dot{e}}_y + E_1\hat{M}^{-1}\hat{C}\underline{\dot{q}} - \dot{E}_1\underline{\dot{q}} - \underline{v}\,\} +$$
$$+\,[\,E_1\hat{M}^{-1}\hat{C} - E_1M^{-1}C\,]\underline{\dot{q}} - \underline{v}.$$

$$\underline{\dot{s}} = \mathbf{P\Lambda\underline{\dot{e}}_y + \underline{r} + \underline{w} + D\underline{\dot{q}} - \underline{v}} \qquad (22),$$

where
$$P = [E_1M^{-1}H][E_1\hat{M}^{-1}H]^{-1} - I\ ,$$
$$\underline{r} = P[\,\underline{\ddot{y}}_d - \dot{E}_1\underline{\dot{q}}\,]\ ,$$
$$\underline{w} = -P\underline{v}\ ,$$
$$D = E_1\hat{M}^{-1}\hat{C} - E_1M^{-1}C\ .$$

Each element of $M(\underline{q})$ and $C(\underline{q},\underline{\dot{q}})$ is assumed to have its upper and lower bound, while the desired trajectory $\underline{\ddot{y}}_d$ is assumed bounded too (because otherwise the trajectory cannot be realized by finite input torques/forces to the system) and $\underline{v}$ has its bounds due to the actuator saturations. Thus, each element of above equation is bounded and one of the stabilizer control inputs, which satisfy the reaching condition (17), is given as follows (Kosuge and Furuta [1988]):

$$\underline{v} = -K_v\underline{\dot{e}}_y - K_p\underline{e}_y - \underline{z} \qquad (23),$$

where
$$z_i = -\mathrm{sgn}(s_i)k_{si} + \underline{a}_i^T\underline{\dot{q}} + \sum_{\substack{j=1\\ i\neq j}}^{m} B_{ij}\dot{e}_y(i),$$
$$k_{si} > \left[\hat{r}_i + \hat{w}_i\right]\mathrm{sgn}(s_i)$$

$a_{ij} =$	a_{ij}^+	$[< -\hat{d}_{ij}]$	when $\dot{q}_j s_i > 0$,
	a_{ij}^-	$[> -\hat{d}_{ij}]$	when $\dot{q}_j s_i < 0$,
$B_{ij} =$	B_{ij}^+	$[< -\Lambda\hat{P}_{ij}]$	when $\dot{e}_{yj} s_i > 0$,
	B_{ij}^-	$[> -\Lambda\hat{P}_{ij}]$	when $\dot{e}_{yj} s_i < 0$,
$k_{vi} = B_{ii}$,			
$k_{pi} =$	k_{pi}^+	$[< 0]$	when $s_i e_{yi} > 0$,
	k_{pi}^-	$[> 0]$	when $s_i e_{yi} < 0$.

In equation (23) the second term could be chosen zero. But in practice, this term can be used in order to decrease k_{si}. Large k_{si} causes a large chattering of v_i.

Equation (23) guarantees that the sufficient reaching condition (17) for the sliding mode to occur is satisfied, as long as v_i does not exceed its physical limit due to the actuator saturations. This means that v_i does not satisfy the sufficient condition if $\hat{M}$ is extremely far from its actual value M and thus w_i becomes very large. But this is not the case in practice, when the parameter uncertainties and variations caused by a payload is not considered so large in usual robot arms. There is the question in which way this stabilizing approach can improve adaptive control of flexible manipulators.

D.6.1 COMPUTED TORQUE CONTROL WITH A FULL REFERENCE TRAJECTORY.

The flexible manipulator dynamics has the next form:

$$\boxed{\underline{M}(\underline{q})\ddot{\underline{q}} + \underline{C}(\underline{q},\dot{\underline{q}})\dot{\underline{q}} = \underline{H}\underline{u}} \qquad (1)$$

where

$\underline{q}(t)$	is the [n×1] vector of independent degrees of freedom of the manipulator model,
$\underline{u}(t)$	is the [m×1] vector of actuator inputs ($m < n$),
H^T	= [I [m×m] 0 [m×(n−m)]].

The known [m×1]desired trajectory vector is: $\underline{y}_d(t) = \underline{y}[\,\underline{q}_r(t)\,]$ (24)
(n unknown variables q_{ri} and only m equations $y_{di}=y(q_{ri})$).

*) **Off–line**, a [n×1] reference trajectory vector $\underline{q}_r(t)$ will be determined with the m equations(24) and the next (n−m) equations of flexible manipulator dynamics (1):

$$\underline{h}^T M(\underline{q}_r)\ddot{\underline{q}}_r + \underline{h}^T C(\underline{q}_r,\dot{\underline{q}}_r)\dot{\underline{q}}_r = \underline{0} \qquad (25)$$

where $\underline{h}^T = [\ \ 0\ [(n-m)\times m]\quad I\ [(n-m)\times(n-m)]\ \ ]$.

*) Which remains from (1) will **on–line** be used in the form of the following computed torque control law:

$$H^T M(\underline{q})\underline{a}_r + H^T C(\underline{q},\dot{\underline{q}})\dot{\underline{q}} = \underline{u} \qquad (26)$$

where $\underline{a}_r = \ddot{\underline{q}}_r + K_v\dot{\underline{e}} + K_p\underline{e}$,
$\underline{e} = \underline{q} - \underline{q}_r$.

D.6.2

Example: **THE TR–ROBOT WITH ONE FLEXIBLE LINK** (only first eigenmode modeled).

The [3×1] vector of independent <u>degrees of freedom</u>:

$$\underline{q}(t) = \begin{bmatrix} \chi(t) \\ \varphi(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \text{translation of the carriage} \\ \text{rotation of the pendulum} \\ \text{bending of the elatic pendulum} \end{bmatrix}$$

The known [2×1] <u>desired trajectory</u> (2) is: $\underline{y}_d(t),\ \dot{\underline{y}}_d(t),\ \ddot{\underline{y}}_d(t).$

$$\underline{y}_d(t) = \underline{y}[\,\underline{q}_r(t)\,] = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} \chi_r + l\cos(\varphi_r) - \omega_r\sin(\varphi_r) \\ l\sin(\varphi_r) + \omega\cos(\varphi_r) \end{bmatrix} \quad \begin{matrix} (5) \\ (6) \end{matrix}$$

The [3×1] <u>reference trajectory</u> $q_r(t)$ and its derivatives to be determined:

$$\omega_r = y_d/\sin(\varphi_r) - 1 \quad \longrightarrow \quad \begin{aligned} \omega_r &= \omega_r(y_d,\ \varphi_r), \\ \dot{\omega}_r &= \omega_r(y_d,\dot{y}_d,\ \varphi_r,\dot{\varphi}_r), \\ \ddot{\omega}_r &= \omega_r(y_d,\dot{y}_d,\ddot{y}_d,\ \varphi_r,\dot{\varphi}_r,\ddot{\varphi}_r). \end{aligned} \quad (7)$$

$$\chi_r = x_d - l\cos(\varphi_r) + \omega_r\sin(\varphi_r) = x_d - l\cos(\varphi_r) + y_d - l\sin(\varphi_r)$$

$$\longrightarrow \quad \begin{aligned} \chi_r &= \chi_r(x_d,\ y_d,\ \varphi_r), \\ \dot{\chi}_r &= \chi_r(x_d,\dot{x}_d,\ y_d,\dot{y}_d,\ \varphi_r,\dot{\varphi}_r), \\ \ddot{\chi}_r &= \chi_r(x_d,\dot{x}_d,\ddot{x}_d,\ y_d,\dot{y}_d,\ddot{y}_d,\ \varphi_r,\dot{\varphi}_r,\ddot{\varphi}_r). \end{aligned} \quad (8)$$

Here, equation (3) from <u>manipulator dynamics</u> (1) can be described as:

$$M_{31}(q_r)\ddot{\chi}_r + M_{32}(q_r)\ddot{\varphi}_r + M_{33}(q_r)\ddot{\omega}_r + n_3(q_r,\dot{q}_r) = 0 \quad (9).$$

By substituting (7) and (8) into (9), one obtains an equation with which φ_r and its derivatives can be determined:

$$f(\varphi_r,\dot{\varphi}_r,\ddot{\varphi}_r) = g(x_d,\dot{x}_d,\ddot{x}_d,\ y_d,\dot{y}_d,\ddot{y}_d) \quad (10).$$

(To facilitate the solution of (10) one can assume for example that φ_r is a second–order polynome:

$$\varphi_r(t) = a + bt + ct^2\ ,$$
$$\dot{\varphi}_r(t) = b + 2ct\ ,$$
$$\ddot{\varphi}_r(t) = 2c\ .)$$

D.6.3 ADAPTIVE COMPUTED TORQUE CONTROL WITH A FULL REFERENCE TRAJECTORY.

At time $t = t_k$ the next variables are <u>known</u>:

$\underline{q}(t_k)$, $\dot{\underline{q}}(t_k)$ are the measured positions and velocities,

$\hat{\underline{p}}(t_k - \Delta t)$ is the vector with adaptively adjusted parameters,

$\underline{y}_d(t_k)$ is the desired output trajectory:

$$\underline{y}_d(t_k) = \underline{y}[\, \underline{q}_r(t_k)\,].$$

In order to determine a <u>reference trajectory</u> $\underline{q}_r(t)$ as far as possible in the same way as in equation (3) but considering also the real manipulator dynamics (1), **parametric uncertainty** forces us to solve the following differential equation of $\underline{q}_r$ on–line:

$$\underline{h}^T \hat{\mathbf{M}}(\, \underline{q}(t_k), \hat{\underline{p}}(t_k - \Delta t)\,)\ddot{\underline{q}}_r(t_k) + \underline{h}^T \hat{\mathbf{C}}(\, \underline{q}(t_k), \dot{\underline{q}}(t_k), \hat{\underline{p}}(t_k - \Delta t)\,)\dot{\underline{q}}(t_k) = \underline{0} \qquad (11).$$

It is now possible to update the estimated parameters $\hat{\underline{p}}$ with an <u>adaptation algorithm</u> of the next form:

$$\boxed{\dot{\hat{\underline{p}}}(t_k) = \Gamma \mathbf{W}^T(\, \underline{q}(t_k), \dot{\underline{q}}(t_k), \dot{\underline{q}}_r(t_k), \ddot{\underline{q}}_r(t_k)\,)\ddot{\underline{e}}(t_k)} \qquad (12),$$

where $\underline{e}(t_k) = \underline{q}(t_k) - \underline{q}_r(t_k)$.

Finally, one utilizes the following <u>adaptive computed torque control law</u>:

$$H^T \hat{\mathbf{M}}(\, \underline{q}(t_k), \hat{\underline{p}}(t_k)\,)\underline{a}_r(t_k) + H^T \hat{\mathbf{C}}(\, \underline{q}(t_k), \dot{\underline{q}}(t_k), \hat{\underline{p}}(t_k)\,)\dot{\underline{q}}(t_k) = \underline{u}(t_k) \qquad (13),$$

where $\underline{a}_r(t_k) = \ddot{\underline{q}}_r(t_k) + K_v\dot{\underline{e}}(t_k) + K_p\underline{e}(t_k)$.

It will be difficult to prove the <u>stability</u> of this adaptive computed torque control system.

It is likely from (11) that

$$\underline{h}^T \hat{\mathbf{M}}(\, \underline{q}(t_k), \hat{\underline{p}}(t_k)\,)\underline{a}_r(t_k) + \underline{h}^T \hat{\mathbf{C}}(\, \underline{q}(t_k), \dot{\underline{q}}(t_k), \hat{\underline{p}}(t_k)\,)\dot{\underline{q}}(t_k) = \underline{d} \qquad (14).$$

Define: $\underline{u}_d = \begin{bmatrix} \underline{u} \\ \underline{d} \end{bmatrix}.$

Then: $$\hat{\mathbf{M}}(\, \underline{q}(t_k), \hat{\underline{p}}(t_k)\,)\underline{a}_r(t_k) + \hat{\mathbf{C}}(\, \underline{q}(t_k), \dot{\underline{q}}(t_k), \hat{\underline{p}}(t_k)\,)\dot{\underline{q}}(t_k) = \underline{u}_d(t_k) \qquad (15).$$

The equivalent error system:

$$\begin{aligned}
M\ddot{\underline{e}} = M\ddot{\underline{q}} - M\ddot{\underline{q}}_r &= \underline{u}_d - \underline{d}_o - C\dot{\underline{q}} - M\ddot{\underline{q}}_r = \\
&= \hat{M}\underline{a}_r + \hat{C}\dot{\underline{q}} - \underline{d}_o - C\dot{\underline{q}} - M\ddot{\underline{q}}_r = \\
&= \hat{M}\ddot{\underline{q}}_r + \hat{M}K_v\dot{\underline{e}} + \hat{M}K_p\underline{e} + \hat{C}\dot{\underline{q}} - \underline{d}_o - C\dot{\underline{q}} - M\ddot{\underline{q}}_r = \\
&= [\hat{M}-M]\ddot{\underline{q}}_r + [\hat{C}-C]\dot{\underline{q}} + \hat{M}K_v\dot{\underline{e}} + \hat{M}K_p\underline{e} - \underline{d}_o :
\end{aligned}$$

$$\mathbf{M}\ddot{\underline{e}} - \hat{\mathbf{M}}\mathbf{K}_v\dot{\underline{e}} - \hat{\mathbf{M}}\mathbf{K}_p\underline{e} = [\hat{\mathbf{M}}-\mathbf{M}]\ddot{\underline{q}}_r + [\hat{\mathbf{C}}-\mathbf{C}]\dot{\underline{q}} - \underline{d}_o = \mathbf{W}(\underline{q}, \dot{\underline{q}}, \dot{\underline{q}}_r, \ddot{\underline{q}}_r)[\, \hat{\underline{p}} - \underline{p}\,] - \underline{d}_o \qquad (16).$$

E TWO–TIME SCALE SLIDING CONTROL OF A FLEXIBLE MANIPULATOR.

Slotine, J.–J. E. and Hong, S. [1986]

E.1 INTRODUCTION.

The main idea of the approach of Slotine and Hong [1986] to the control of flexible manipulators leads to a two–time scale system structure as given in figure [1] at page 6. It starts with a separation of the dynamic equations of a flexible manipulator:

$$\boxed{\mathrm{M}(\underline{q})\ddot{\underline{q}} + \underline{n}(\underline{q},\dot{\underline{q}}) = \mathbf{H}\underline{u}} \qquad (1),$$

in those describing the rigid body motion resp. those describing the flexible effects:

$$\ddot{\underline{q}}_r = \underline{f}_1(\underline{q}_r,\dot{\underline{q}}_r,\underline{q}_f,\dot{\underline{q}}_f) \qquad (2),$$

$$\ddot{\underline{q}}_f = \underline{f}_2(\underline{q}_r,\dot{\underline{q}}_r,\underline{q}_f,\dot{\underline{q}}_f) \qquad (3).$$

Then, they investigate the elasticity properties by the singular perturbation technique of Marino and Nicosia [1984] to decompose the manipulator dynamics into a 'slow' resp. a 'fast' submodel:

$$\boxed{\begin{aligned} \ddot{\underline{\chi}} &= \mathbf{g}_1(\underline{\chi},\dot{\underline{\chi}},\underline{z},\dot{\underline{z}},\mu) \\ \mu\ddot{\underline{z}} &= \mathbf{g}_2(\underline{\chi},\dot{\underline{\chi}},\underline{z},\dot{\underline{z}},\mu) \end{aligned}} \qquad \begin{aligned}(4),\\(5),\end{aligned}$$

Assuming that only small deviations from the rigid body motion will occur and are to be considered, they further use the concept of slow manifold (Khorasani and Spong [1985]) to obtain the next linearized singular perturbation model:

$$\ddot{\underline{\chi}} = \underline{\mathbf{k}}_1(\underline{\chi},\dot{\underline{\chi}},\underline{h},\dot{\underline{h}},\mu) \qquad (6),$$

$$\mu\ddot{\underline{z}} = \underline{\mathbf{k}}_2(\underline{\chi},\dot{\underline{\chi}},\underline{h},\dot{\underline{h}},\mu) \qquad (7),$$

The reduced flexible model (6) represents the system dynamics restricted to a sonamed 'slow manifold' described by:

$$\boxed{\begin{aligned} \underline{z} &= \underline{h}(\underline{\chi},\dot{\underline{\chi}},\underline{u},\mu) \\ \dot{\underline{z}} &= \dot{\underline{h}}(\underline{\chi},\dot{\underline{\chi}},\underline{u},\mu) \end{aligned}} \qquad (8),$$

and is of the same dimension as the rigid model of the flexible system, but preserves and captures the dynamics of the full system to a higher degree of accuracy. It can now be used to design a slow feedback control $\underline{u}_r$. Further, a fast control $\underline{u}_f$ is needed in order to guarantee that the slow manifold is actually attractive, i.e. that all system trajectories converge to the manifold. The composite two–time scale controller is then defined as:

$$\underline{u} = \underline{u}_r + \underline{u}_f \qquad (9),$$

where $\underline{u} = \underline{u}(\underline{\chi},\dot{\underline{\chi}},\underline{z},\dot{\underline{z}},\mu)$,
$\underline{u}_r(\underline{\chi},\dot{\underline{\chi}},\mu)$ controls the rigid body motions,
$\underline{u}_f(\underline{z}-\underline{h}(\underline{\chi},\dot{\underline{\chi}},\underline{u}_r,\mu),\ \dot{\underline{z}}-\dot{\underline{h}}(\underline{\chi},\dot{\underline{\chi}},\underline{u}_r,\mu))$ controls the elastic modes.

For the required robustness to parametric uncertainty, Slotine and Hong finally use the sliding control methodology of Slotine [1984] to design the two–time scale sliding controller:

$$\underline{u}_S = \underline{u}_{S_r} + \underline{u}_{S_f} \qquad (10).$$

*) Based on the control $\underline{u}_r$ for the rigid robot system, the slow sliding controller $\underline{u}_{sr}$ is designed to account for parametric uncertainty on the slow manifold.
*) The purpose of the fast sliding controller $\underline{u}_{sf}$ is then to force the fast variables to follow the slow manifold, despite the presence of uncertainty.

E.2 MANIPULATOR DYNAMICS IN SINGULAR PERTURBATION FORMULATION.

In the paper of Benati and Morro [1988], a chain of flexible links has been modeled as a system with a finite number of degrees of freedom. The main advantage of their approach through Lagrange's formalism is, that it leads to explicit dynamic equations of the flexible manipulator (in terms of well defined geometrical parameters) which, in general, can be written in the next form:

$$M(\underline{q})\underline{\ddot{q}} + \underline{n}(\underline{q},\underline{\dot{q}}) = H\underline{u} \quad (1),$$

where
- $\underline{u}(t)$ is the $[m\times 1]$ vector of actuator inputs,
- $\underline{q}(t)$ is the $[n\times 1]$ vector of generalized coordinates which can be split up into two vectors:

$$\underline{q} = \begin{bmatrix} \underline{q}_r \\ \underline{q}_f \end{bmatrix},$$

where
- $\underline{q}_r(t)$ is the $[k\times 1]$ vector describing the rigid motion of links,
- **$\underline{q}_f(t)$ is the $[(n-k)\times 1]$ vector of flexibility degrees of freedom.**
- $\underline{n}(q,\dot{q})$ can be split up into: $\underline{n}(q,\dot{q}) = \underline{n}_1(q,\dot{q}) + \underline{n}_2(q)kq_f$

With $\underline{\ddot{q}} = M^{-1}[\, H\underline{u}-\underline{n}\,]$ and after some mathematical manipulations, we get:

$$\underline{\ddot{q}}_r = \underline{a}_1(\underline{q}_r,\underline{\dot{q}}_r,\underline{q}_f,\underline{\dot{q}}_f) + A_1(\underline{q}_r,\underline{q}_f)k\underline{q}_f + B_1(\underline{q}_r,\underline{q}_f)\underline{u} \quad (2),$$

$$\underline{\ddot{q}}_f = \underline{a}_2(\underline{q}_r,\underline{\dot{q}}_r,\underline{q}_f,\underline{\dot{q}}_f) + A_2(\underline{q}_r,\underline{q}_f)k\underline{q}_f + B_2(\underline{q}_r,\underline{q}_f)\underline{u} \quad (3).$$

where $\underline{a}(q,\dot{q}) = -M^{-1}(q)\,n_1(q,\dot{q})$
$A(q) = -M^{-1}(q)\,n_2(q)$
$B(q) = M^{-1}(q)H$

Defining:
$$\begin{cases} \chi = \underline{q}_r \\ \underline{z} = k\underline{q}_f \\ \mu = 1/k \end{cases} \quad (11),$$

the equations of motion for the flexible manipulator can be written in the general form of a singular perturbation model:

$$\ddot{\chi} = \underline{a}_1(\chi,\dot{\chi},\mu\underline{z},\mu\underline{\dot{z}}) + A_1(\chi,\mu\underline{z})\underline{z} + B_1(\chi,\mu\underline{z})\underline{u} \quad (12),$$

$$\mu\underline{\ddot{z}} = \underline{a}_2(\chi,\dot{\chi},\mu\underline{z},\mu\underline{\dot{z}}) + A_2(\chi,\mu\underline{z})\underline{z} + B_2(\chi,\mu\underline{z})\underline{u} \quad (13),$$

where
- χ is the $[k\times 1]$ state vector associated with the 'slow' dynamics,
- $\underline{z}$ is the $[(n-k)\times 1]$ state vector associated with the 'fast' dynamics,
- μ is the inverse of the joint/link mechanical stiffness [very small scalar].

By formally setting $\mu=0$ and eliminating $\underline{z}$ from the equations, (12)–(13) are reduced to the equations of motion of a rigid manipulator: $M(\chi)\ddot{\chi} + \underline{n}(\chi,\dot{\chi}) = H\underline{u}$.

E.3 DESIGN OF THE SLOW SLIDING CONTROLLER.

The singular perturbation model (12)–(13) is complex and nonlinear, and is not directly linearizable. However, it can easily be decomposed into a 'slow' and a 'fast' submodel by using the slow manifold:

$$\underline{\mathbf{z}} = \underline{\mathbf{h}}(\chi,\dot{\chi},\underline{u},\mu)$$
$$\dot{\underline{\mathbf{z}}} = \dot{\underline{\mathbf{h}}}(\chi,\dot{\chi},\underline{u},\mu) \qquad (8).$$

which can be obtained as the solution of a partial differential equation formed by substituting its expression into equation (13):

$$\mu\ddot{\underline{\mathrm{h}}} = \underline{\mathbf{a}}_2(\chi,\dot{\chi},\mu\underline{\mathrm{h}},\mu\dot{\underline{\mathrm{h}}}) + \mathbf{A}_2(\chi,\mu\underline{\mathrm{h}})\underline{\mathbf{h}} + \mathbf{B}_2(\chi,\mu\underline{\mathrm{h}})\underline{\mathbf{u}} \qquad (14).$$

Once $\underline{h}$ is determined from this sonamed manifold condition, the dynamics of the system (12)–(13) on the manifold are given by a reduced–order system referred to as the reduced flexible system, which is performed by replacing $\underline{z}$ by $\underline{h}$ in equation (12):

$$\ddot{\chi} = \underline{\mathbf{a}}_1(\chi,\dot{\chi},\mu\underline{\mathrm{h}},\mu\dot{\underline{\mathrm{h}}}) + \mathrm{A}_1(\chi,\mu\underline{\mathrm{h}})\underline{\mathbf{h}} + \mathbf{B}_1(\chi,\mu\underline{\mathrm{h}})\underline{\mathbf{u}} \qquad (15).$$

The computation of the slow feedback control $\underline{u}=\underline{u}_r$ out of this reduced flexible system is complicated by the need to solve the manifold condition (14). Therefore, an approximation to $\underline{h}$ and $\underline{u}_r$ is obtained by expanding them in a power series of μ. In practice, the following first–order expansion is generally adequate to capture the dynamic effects of interest:

$$\underline{\mathrm{h}}(\chi,\dot{\chi},\underline{\mathrm{u}}_r,\mu) = \underline{\mathrm{h}}_0(\chi,\dot{\chi},\underline{\mathrm{u}}_r) + \mu\underline{\mathrm{h}}_1(\chi,\dot{\chi},\underline{\mathrm{u}}_r) \qquad (16),$$

$$\underline{\mathrm{u}}_r(\chi,\dot{\chi},\underline{\mathrm{w}},\mu) = \underline{\mathrm{u}}_0(\chi,\dot{\chi},\underline{\mathrm{w}}) + \mu\underline{\mathrm{u}}_1(\chi,\dot{\chi},\underline{\mathrm{w}}) \qquad (17),$$

where

$\underline{w}$ is a new input which is assumed to be known,
$\underline{h}_0$ and $\underline{u}_0$ are the manifold resp. the control obtained from the rigid model (μ=0):

(14) $\underline{a}_{2o}+A_{2o}\underline{h}_0+B_{2o}\underline{u}_0 = \underline{0}$ –> $\underline{\mathbf{h}}_0 = -\mathrm{A}_{2o}^{-1}[\underline{\mathbf{a}}_{2o} + \mathbf{B}_{2o}\underline{\mathbf{u}}_0]$ (18),

(15) $\underline{a}_{1o}+A_{1o}\underline{h}_0+B_{1o}\underline{u}_0 = \underline{w}$ –> $\underline{\mathbf{u}}_0 = -\mathbf{B}^{-1}[\underline{\mathbf{a}} - \underline{\mathbf{w}}]$ (19),

where $B = [B_{1o}-A_{1o}A_{2o}^{-1}\,B_{2o}]$,
$\underline{a} = [\underline{a}_{1o}-A_{1o}A_{2o}^{-1}\,\underline{a}_{2o}]$.

Using this known rigid manifold $\underline{h}_0$ and control $\underline{u}_0$, we can obtain $\underline{h}_1$ from (14) under the assumption that the fast variables $\underline{z}$ are on the first–order corrected slow manifold [neglecting terms of order μ^2]:

$$\mu\ddot{\underline{\mathrm{h}}}_0 = \underline{\mathrm{a}}_2 + \mathrm{A}_2\underline{\mathrm{h}}_0 + \mu\mathrm{A}_2\underline{\mathrm{h}}_1 + \mathrm{B}_2\underline{\mathrm{u}}_0 + \mu\mathrm{B}_2\underline{\mathrm{u}}_1 \qquad (20).$$

The definitions of:

$$\underline{a}_i(\chi,\dot{\chi},\mu\underline{h},\mu\dot{\underline{h}}) = \underline{a}_{io}(\chi,\dot{\chi},\underline{0},\underline{0}) + \mu\Delta\underline{a}_i(\chi,\dot{\chi},\underline{h}_0,\dot{\underline{h}}_0),$$
$$A_i(\chi,\mu\underline{h}) = A_{io}(\chi,\underline{0}) + \mu\Delta A_i(\chi,\underline{h}_0),$$
$$B_i(\chi,\mu\underline{h}) = B_{io}(\chi,\underline{0}) + \mu\Delta B_i(\chi,\underline{h}_0), \qquad (21)$$

will lead to an expression for $\underline{h}_1$:

$$\underline{\mathbf{h}}_1 = \mathbf{A}_{2o}^{-1}[\underline{\mathrm{d}} - \mathbf{B}_{2o}\underline{\mathbf{u}}_1] \qquad (22),$$

where

expressions with $1/\mu$ are omitted,

$\underline{d} = \ddot{\underline{h}}_0 - \Delta\underline{a}_2 - \Delta A_2\underline{h}_0 - \Delta B_2\underline{u} - B_{2o}\underline{u}_0$.

Also, the first–order corrected slow subsystem can be written [μ^2=0]:

$$\ddot{\underline{\chi}} = \underline{w} + \mu[\,\underline{b} + \underline{c} + \mathbf{B}\underline{\mathbf{u}}_1] \qquad (23),$$

where
$$\underline{w} = \underline{a}_{1o} + A_{1o}\underline{h}_0 + B_{1o}\underline{u}_0,$$
$$\underline{b} = \Delta\underline{a}_1 + \Delta A_1\underline{h}_0 + \Delta B_1\underline{u}_0,$$
$$\underline{c} = [\,A_{1o} + \mu\Delta A_1\,]A_{2o}^{-1}\underline{d}.$$

*) When there is no model uncertainty: $\ddot{\underline{\chi}} = \underline{a}_{1o} + A_{1o}\underline{h}_0 + B_{1o}\underline{u}_0 = \underline{w}$.
Then the corrective control compensating for flexibility will be:

$$\underline{u}_1 = -\mathbf{B}^{-1}[\,\underline{b} + \underline{c}\,] \qquad (24).$$

*) However, with parametric uncertainty, $\underline{h}_0$ and $\underline{u}_0$ are only known as:

$$\underline{h}_0 = -\hat{A}_{2o}^{-1}[\hat{\underline{a}}_{2o} + \hat{B}_{2o}\underline{u}_0] \qquad (25),$$

$$\underline{u}_0 = -\hat{B}^{-1}[\hat{\underline{a}} - \hat{\underline{w}}] \qquad (26),$$

where ˆ indicates available estimates.

Therefore, an additional control term is required in the corrective control $\underline{u}_1$, in order for a flexible manipulator to track the desired trajectory despite the presence of parametric uncertainty. Considering equation (23), Slotine and Hong have derived the following slow sliding control law:

$$\underline{us}_1 = \mu^{-1}\hat{B}^{-1}[\,\underline{u}_x - \hat{\underline{w}}\,] - \hat{B}^{-1}[\,\hat{\underline{b}} + \hat{\underline{c}}\,] \qquad (27),$$

where $\underline{u}_x$ is a new input designed to achieve desired closed–loop specifications:

$$\underline{u}_x = \ddot{\underline{\chi}}_d - K_{V_x}\dot{\underline{e}}_x - K_{P_x}\underline{e}_x - K_{S_x}\,\mathrm{sat}[\,\underline{s}_x/\underline{\Phi}_x\,],$$

$\underline{e}_x$ is the tracking error of the rigid motion state variables:

$$\underline{e}_x = \underline{\chi} - \underline{\chi}_d,$$

$\underline{s}_x$ is the slow sliding surface defined as:

$$\underline{s}_x = \dot{\underline{e}}_x + K_{V_x}\underline{e}_x + K_{P_x}\int_0^t \underline{e}_x\,dt,$$

K_{S_x} and $\mathrm{sat}[\,\underline{s}_x/\underline{\Phi}_x\,]$ are defined as in Asada and Slotine [1986].

Finally, the slow sliding controller u_{sr} is in its first–order expansion defined in the same way as $\underline{u}_r$ in (17)

$$\underline{us}_r = \underline{u}_0 + \mu\underline{us}_1 = \hat{B}^{-1}\{\,-\hat{\underline{a}} + \hat{\underline{w}} + \underline{u}_x - \hat{\underline{w}} - \mu[\,\hat{\underline{b}} + \hat{\underline{c}}\,]\,\}$$

Thus:

$$\boxed{\underline{us}_r = \hat{B}^{-1}\{\,\underline{u}_x - \hat{\underline{a}} - \mu[\,\hat{\underline{b}} + \hat{\underline{c}}\,]\,\}} \qquad (28).$$

E.4 DESIGN OF THE FAST SLIDING CONTROLLER.

The slow submodel was derived under the assumption that the fast variables $\underline{z}$ follow the slow manifold $\underline{h}$. However, this is not necessarily guaranteed in the presence of parametric uncertainty. Therefore, a fast control $\underline{u}_f$ is needed to force the fast variables to follow the desired manifold, i.e. to make the desired manifold attractive despite parametric uncertainty:

$$\underline{u}_f = \underline{u}_f(\underline{\eta}) \quad (29),$$

where $\underline{\eta}$ represents the deviation of the fast variables from the desired manifold:

$$\underline{\eta} = \underline{z} - \underline{h} = \underline{z} - (\,\underline{h}_0 + \mu \underline{h}_1\,) \quad (30).$$

The influence of the fast control on the slow subsystem can be neglected in the design when $\underline{\eta}$ is maintained at a small value by the fast sliding control $\underline{u}_f$ after the decay of fast dynamics. To derive the fast submodel, we can make the assumption that the slow states $\underline{\chi}$ and their derivatives are fixed parameters $\underline{\chi}_o$ and $\underline{\nu}_o$ during this fast transient. This allows to simplify the problem when the slow and fast time scale are significantly different: $t_f < t_r$. By expressing the fast flexible model (13) in the <u>fast time scale</u> defined as $t_f = t/\mu$, we get:

$$\underline{z}' = \dot{\underline{z}}\,(\delta t/\delta t_f) = \mu\dot{\underline{z}},$$
$$\underline{z}'' = \ddot{\underline{z}}(\delta t/\delta t_f) = \mu^2\ddot{\underline{z}}.$$

where $'$ indicates differentiation with respect to t_f.

Equ.(13)–> $\underline{z}'' = \mu \underline{a}_2(\underline{\chi}_o, \underline{\nu}_o, \mu\underline{z}, \underline{z}') + \mu A_2(\underline{\chi}_o, \mu\underline{z})\underline{z} + \mu B_2(\underline{\chi}, \mu\underline{z})\underline{u}$ (31),
which
with (30): $\underline{z} = \underline{\eta} + \underline{h}_o + \mu\underline{h}_1$,
becomes:

$$\begin{aligned}\underline{\eta}'' + \underline{h}_0'' + \mu\underline{h}_1'' &= \mu\underline{a}_2\,(\,\underline{\chi}_o, \underline{\nu}_o, \mu\underline{\eta}+\mu\underline{h}_o+\mu^2\underline{h}_1, \underline{\eta}'+\underline{h}_o'+\mu\underline{h}_1'\,) + \\ &+ \mu A_2\,(\,\underline{\chi}_o, \mu\underline{\eta}+\mu\underline{h}_o+\mu^2\underline{h}_1\,)\,[\,\underline{\eta}+\underline{h}_o+\mu\underline{h}_1\,] + \\ &+ \mu B_2\,(\underline{\chi}_o, \mu\underline{\eta}+\mu\underline{h}_o+\mu^2\underline{h}_1\,)\,[\,\underline{u}_r+\underline{u}_f\,] \quad (32).\end{aligned}$$

<u>The first–order corrected fast manifold</u> can be defined from the equation above:

$$\begin{aligned}\underline{h}_0'' + \mu\underline{h}_1'' &= \mu\underline{a}_2\,(\,\underline{\chi}_o, \underline{\nu}_o, \mu\underline{h}_o+\mu^2\underline{h}_1, \underline{h}_o'+\mu\underline{h}_1'\,) + \\ &+ \mu A_2\,(\,\underline{\chi}_o, \mu\underline{h}_o+\mu^2\underline{h}_1\,)\,[\,\underline{h}_o+\mu\underline{h}_1\,] + \\ &+ \mu B_2\,(\,\underline{\chi}_o, \mu\underline{h}_o+\mu^2\underline{h}_1\,)\,\underline{u}_r \quad (33).\end{aligned}$$

Which remains from (32) is the following expression:

$$\underline{\eta}'' = \mu\underline{a}_2(\underline{\chi}_o, \underline{\nu}_o, \mu\underline{\eta}, \underline{\eta}') + \mu A_2(\underline{\chi}_o, \mu\underline{\eta})\underline{\eta} + \mu B_2(\underline{\chi}_o, \mu\underline{\eta})\underline{u}_f \quad (34).$$

Neglecting terms of order μ^2, we obtain <u>the first–order corrected fast submodel</u>:

$$\underline{\eta}'' = \mu\underline{a}_{2n}(\underline{\chi}_o, \underline{\nu}_o, \underline{0}, \underline{\eta}') + \mu A_{2n}(\underline{\chi}_o, \underline{0})\underline{\eta} + \mu B_{2n}(\underline{\chi}_o, \underline{0})\underline{u}_f \quad (35).$$

In order to make the available manifold attractive in the presence of parametric uncertainty, the fast sliding controller then takes the following form:

$$\boxed{\underline{u}_{s_f} = \hat{B}_{2n}^{-1} \{ \underline{u}_z/\mu - \hat{\underline{a}}_{2n} - \hat{A}_{2n}\underline{\eta} \}} \qquad (36),$$

where $\underline{u}_z = -K_{V_z}\underline{\eta}' - K_{P_z}\underline{\eta} - K_{S_z}\,\mathrm{sat}[\,\underline{s}_z/\underline{\Phi}_z\,]$,

the fast sliding surface is defined as:

$$\underline{s}_z = \underline{\eta}' + K_{V_z}\underline{\eta} + K_{P_z}\int_{t_0}^{t_f} \underline{\eta}\,dt_f.$$

E.5. THE TWO–TIME SCALE SLIDING CONTROLLER.

The composite two–time scale sliding controller is finally defined as:

$$\underline{u}_s = \underline{u}_{s_r} + \underline{u}_{s_f} \qquad (8)$$

$$= \hat{B}^{-1} \{ \underline{u}_x - \hat{\underline{a}} - \mu[\,\hat{\underline{b}}+\hat{\underline{c}}\,] \} +$$

$$+ \hat{B}_{2n}^{-1} \{ \underline{u}_z/\mu - \hat{\underline{a}}_{2n} - \hat{A}_{2n}\underline{\eta} \} \qquad (37).$$

F (ADAPTIVE) TWO–SUBMODEL BASED COMPUTED TORQUE CONTROL OF FLEXIBLE ROBOT SYSTEMS.

F.1 INTRODUCTION.

The problem in controlling a lightweight mechanical manipulator is to perform fast, accurate and robust motions despite structural flexibilities, payload variations and other environmental disturbances. An (adaptive) two–submodel based control approach has been developed to extend the familiar computed torque control scheme for rigid robot systems to flexible robots.

F.1.1 The flexible manipulator system.

A lightweight manipulator may have flexibility in the link structure and/or elasticity in the motor joints. For most manipulators, joint elasticity has a greater significance for control system design than the actual bending modes of the links. Furthermore, the distributed link flexibility can be approximately modeled by a chain of rigid sublinks interconnected by elastic joints. Hence, a more accurate (thus higher order) representation of robot dynamics involving elastic joints should be taken into account to get better control performance.

In comparison with a rigid robot, the system dynamics of a flexible robot with n degrees of freedom is still governed by the same type of second–order, coupled, highly nonlinear differential equations:

$$\boxed{\mathbf{M}(\underline{q})\underline{\ddot{q}} + \underline{\mathbf{n}}(\underline{q},\underline{\dot{q}},t) = \mathbf{H}\underline{\mathbf{u}}\ ,} \tag{1}$$

where $\underline{q}(t)$ is the vector with n generalized coordinates,
$M(\underline{q})$ is the [n*n] mass inertia matrix,
$\underline{u}(t)$ is the vector with m motor input signals,
$\underline{n}(\underline{q},\underline{\dot{q}},t)$ contains the centrifugal and Coriolis forces/torques, gravity, friction, etcetera.

With $\underline{y} = [\ \underline{q}^T\ \ \underline{\dot{q}}^T\ ]^T$ we obtain 2n first–order nonlinear differential equations:

$$\underline{\dot{y}} = \underline{f}(\underline{y},\underline{u},t)\ . \tag{2}$$

However, an important difference is that, when the structural flexibilities are included, the number of inputs is less than the number of degrees of freedom (m<n). A 'rigid' control strategy $\underline{u}_r(t)$, which just tries to track a desired trajectory in space specified by m generalized coordinates, will often result in an instable system behavior. Therefore, the control system must deal with control of the elastic vibrations as well as the joint trajectory tracking. However, it is not possible to find a control input for a lightweight manipulator which will accomplish perfect tracking of any given desired trajectory in joint space while totally damping the undesired flexible deflections. It is more realistic to search for a control law achieving both a reasonable trajectory tracking and a certain stabilization of acceptable vibrations.

F.1.2 Two submodels of the flexible manipulator system.

In first instance, starting with Slotine & Hong [1987], we assume that n=2m and that the dynamic model (2) of the flexible manipulator can be split up into two subsystems:

$$\dot{\underline{x}} = \underline{g}(\underline{x},\underline{z},t) \ , \quad (3)$$

$$\dot{\underline{z}} = \underline{h}(\underline{x},\underline{z},\underline{u},t) \ . \quad (4)$$

If the desired trajectory is specified by $\underline{x}_d(t)$, the tracking error is defined as:

$$\underline{e} = \underline{x}_d - \underline{x} \ . \quad (5)$$

Then, we introduce a reference manifold error:

$$\underline{e}_z = \underline{z}_r - \underline{z} \ , \quad (6)$$

in which the reference vector $\underline{z}_r(\underline{x},t)$ must be chosen in such a way that substitution of (6) in (3) will result in a stable differential tracking error equation:

$$\dot{\underline{e}} = \dot{\underline{x}}_d - \dot{\underline{x}} = \dot{\underline{x}}_d - \underline{g}(\underline{x},\underline{e}_z,t) \ . \quad (7)$$

That is to say: the tracking error $\underline{e}(t)$ will tend to zero in time if $\underline{e}_z=\underline{0}$.

F.1.3 Two–submodel based (adaptive) flexible robot control.

To force the generalized coordinates $\underline{z}(t)$ to their references $\underline{z}_r(t)$, the intention now is to obtain a stable differential reference manifold error equation too:

$$\dot{\underline{e}}_z = \dot{\underline{z}}_r - \dot{\underline{z}} = \dot{\underline{z}}_r - \underline{h}(\underline{x},\underline{z}_r(\underline{x},t)-\underline{e}_z,\underline{u}_f,t) \quad (8)$$

by choosing a suitable 'flexible' control input signal $\underline{u}_f(\underline{x},\underline{z},t)$, which will be composed of a computed torque control part (with internal PD action) and, for example, a sliding control part (in order to obtain robustness against uncertainties and parameter variations: Asada & Slotine [1986]). The computed torque control part appears to be a combination of the 'rigid' computed torque control law $\underline{u}_r(t)$ of the robot model without flexibilities and of a computed torque control term multiplicated with the inverse of the stiffness matrix K of all elastic joints and/or flexible links:

$$\underline{u}_f(\underline{x},\underline{z},t) = \underline{u}_r(\underline{x},t) + K^{-1}\underline{u}_e(\underline{x},\underline{z},t) + \underline{u}_{sliding}. \quad (9)$$

The term with $\underline{u}_e$ tries to force the flexible motions to behave in a more natural way according to the equations of motion of the flexible system.

Unfortunately, the computed torque control method relies heavily on an accurate prior knowledge of the robot system dynamics and, therefore, above approach will further be expanded to an adaptive control technique in which the unknown, but constant system parameters will be adjusted on–line (basically according to the method of Slotine & Li [1987]). Finally, the global asymptotic stability of the control system is guaranteed through the Hyperstability approach of Popov [1969]. The new (adaptive) flexible robot control method will be illustrated by simulation results.

F.2 THE FLEXIBLE MANIPULATOR SYSTEM.

The equations of motion for a manipulator system with linear elastic joints are:

$$\mathbf{M_r}(\underline{\mathbf{q}}_r)\ddot{\underline{\mathbf{q}}}_r + \mathbf{C}(\underline{\mathbf{q}}_r,\dot{\underline{\mathbf{q}}}_r)\dot{\underline{\mathbf{q}}}_r + \mathbf{g}(\underline{\mathbf{q}}_r) = \mathbf{K}(\underline{\mathbf{q}}_f-\underline{\mathbf{q}}_r) \tag{10}$$

$$\mathbf{J}\ddot{\underline{\mathbf{q}}}_f + \mathbf{B}\dot{\underline{\mathbf{q}}}_f + \mathbf{K}(\underline{\mathbf{q}}_f-\underline{\mathbf{q}}_r) = \underline{\mathbf{u}} \tag{11}$$

where $\underline{q}_r(t)$ is the [(n–m)*1] vector with link variables,
$\underline{q}_f(t)$ is the [m*1] vector with actuator variables,
$M_r(\underline{q}_r)$ is the mass inertia matrix of the rigid–link robot,
J is the mass inertia matrix of the joint motors,
K is the diagonal stiffness matrix of the linear elastic joints.

$$\underline{z} = K(\underline{q}_f-\underline{q}_r) \tag{12}$$

are the elastic forces/torques at the joints, coupling equation (10) with (11):

$$M_r\ddot{\underline{q}}_r + C\dot{\underline{q}}_r + \underline{g} = \underline{z} = \underline{u} - J\ddot{\underline{q}}_f - B\dot{\underline{q}}_f \tag{13}$$

The equations of motion for the rigid manipulator system are:

$$[\mathbf{M_r}(\underline{\mathbf{q}}_r)+\mathbf{J}]\ddot{\underline{\mathbf{q}}}_r + [\mathbf{C}(\underline{\mathbf{q}}_r,\dot{\underline{\mathbf{q}}}_r)+\mathbf{B}]\dot{\underline{\mathbf{q}}}_r + \mathbf{g}(\underline{\mathbf{q}}_r) = \underline{\mathbf{u}} \tag{14}$$

According to Slotine and Hong [1987], with the definition of

$\underline{\mathbf{x}} = \underline{\mathbf{q}}_r$	as the 'rigid' variables,	(15)
$\underline{\mathbf{z}} = \mathbf{K}(\underline{\mathbf{q}}_f-\underline{\mathbf{q}}_r)$	as the 'flexible' variables,	(16)
$\mu = \mathbf{K}^{-1}$	as the very small 'parasitic' elasticity matrix,	(17)

the equations of motion of the flexible system (10)–(11) are changed into those of a socalled singularly perturbated system:

$$\boxed{\begin{aligned} &\mathbf{M_r}(\underline{\mathbf{x}})\ddot{\underline{\mathbf{x}}} + \mathbf{C}(\underline{\mathbf{x}},\dot{\underline{\mathbf{x}}})\dot{\underline{\mathbf{x}}} + \mathbf{g}(\underline{\mathbf{x}}) = \underline{\mathbf{z}} && (18)\\ &\mu\mathbf{J}\ddot{\underline{\mathbf{z}}} + \mu\mathbf{B}\dot{\underline{\mathbf{z}}} + [\mathbf{JM}^{-1}+\mathbf{I}]\underline{\mathbf{z}} = \underline{\mathbf{u}} + [\mathbf{JM}^{-1}\mathbf{C}-\mathbf{B}]\dot{\underline{\mathbf{x}}} + \mathbf{JM}^{-1}\mathbf{g} && (19)\end{aligned}}$$

If $\mu \to 0$, equations (18)–(19) become the equations of a quasi–steady state system:

$$[M_r(\underline{x})+J]\ddot{\underline{x}} + [C(\underline{x},\dot{\underline{x}})+B]\dot{\underline{x}} + g(\underline{x}) = \underline{u}\ , \tag{20}$$

which approximates the rigid manipulator model (14) and represents the relationship between the rigid state variables $\underline{x}(t)$ describing the behavior of the flexible system (10)–(11) if it is forced to constrain its 'flexible' evolution of $\underline{z}(t)$ on the reference manifold $\underline{z}_r(t)$.

F.3 RIGID COMPUTED TORQUE CONTROL.

The first step in the derivation of the two–submodel based control law is the formulation of the next rigid computed torque control term:

$$\boxed{\underline{u}_r = [M_r(\underline{x})+J](\ddot{\underline{x}}_r+K_r\dot{\underline{e}}_r) + [C(\underline{x},\dot{\underline{x}})+B]\dot{\underline{x}} + \mathbf{g}(\underline{x})} \quad (21)$$

where $\underline{e}_r=\underline{e}+K_x\int\underline{e}$ is a reference trajectory error (22)

$\underline{e} = \underline{x}_d - \underline{x}$ is the tracking error. (23)

With this control input we can define a certain 'reference manifold', describing the elastic forces/ torques required for manipulating the rigid links along the desired trajectory:

$$\underline{z}_r = M_r(\underline{x})(\ddot{\underline{x}}_r+K_r\dot{\underline{e}}_r) + C(\underline{x},\dot{\underline{x}})\dot{\underline{x}} + \mathbf{g}(\underline{x}) \quad (24)$$

The equivalent reference trajectory error equations of the closed–loop system are:

$$M_r(\ddot{\underline{e}}_r+K_r\dot{\underline{e}}_r) = \underline{e}_z \quad (25)$$

where $\underline{e}_z = \underline{z} - \underline{z}_r$ is the reference manifold error (26)

F.3.1 Sliding control term.

If μ is very small, i.e. if the elastic joints are nearly stiff, there is only needed an extra sliding control term $\underline{u}_s(t)$ (to be added to $\underline{u}_r(t)$) in order to force the flexible state variables $\underline{z}(t)$ towards the reference manifold $\underline{z}_r(t)$ and to keep the system in sliding motion on $\underline{e}_z(t)=\underline{0}$:

$$\underline{u} = \underline{u}_r + \underline{u}_s \quad (27)$$

$\underline{u}_r$ represents the 'equivalent' control term (21) when the system is in sliding motion. A variable structure control (V.S.C.) law is obtained by letting the control function $\underline{u}$ be defined as follows:

$$\underline{u} = \begin{cases} \underline{u}^+ & , e_z > 0 \\ \underline{u}_r & , e_z = 0 \\ \underline{u}^- & , e_z < 0 \end{cases} \quad (28)$$

where a necessary and sufficient existence condition for the local existence of sliding motion on $\underline{e}_z=\underline{0}$ is:

$$\underline{u}^- < \underline{u}_r < \underline{u}^+ \quad (29)$$

As a result of this control policy, the flexible state trajectories of the system reach locally the sliding surface $\underline{e}_z=\underline{0}$ if:

$$\lim_{\underline{e}_z \downarrow \underline{0}} \dot{\underline{e}}_z < \underline{0}\,, \qquad \lim_{\underline{e}_z \uparrow \underline{0}} \dot{\underline{e}}_z > \underline{0} \tag{30}$$

i.e., in the neighbourhood of $\underline{e}_z=\underline{0}$ is the surface reachability condition:

$$\dot{\underline{e}}_z^{\,T}\dot{\underline{e}}_z < \underline{0} \tag{31}$$

which guarantees a crossing of the sliding surface $\underline{e}_z=\underline{0}$ from each side of it by use of a sliding control term (for example: $\underline{u}_s = -k.\mathrm{sgn}(\underline{e}_z)$; see Slotine & Li [1986]).

F.4 FLEXIBLE COMPUTED TORQUE CONTROL.

If the parasitic elasticity parameter μ is not very small, also the 'flexible' terms in the equations of motion have to be compensated for in the computed torque control, instead of approximating only the rigid manipulator system with μ=0.

Considering the flexible robot model (18)–(19), by substituting the elastic forces $\underline{z}(t)$ of (18) into (19) we get:

$$\mu J\ddot{\underline{z}} + \mu B\dot{\underline{z}} + [M_r(\underline{x})+J]\ddot{\underline{x}} + [C(\underline{x},\dot{\underline{x}})+B]\dot{\underline{x}} + \underline{g}(\underline{x}) = \underline{u} \tag{32}$$

The flexible computed torque control law will be

$$\boxed{\begin{aligned} \underline{u} &= \underline{u}_r + \mu\underline{u}_f = \\ &= \underline{u}_r + \mu\{\, J(\ddot{\underline{z}}_r + K_z\dot{\underline{e}}_z) + B\dot{\underline{z}}\,\} \end{aligned}} \tag{33}$$

Now, the equivalent reference trajectory error , cq. reference manifold error equations are:

$$M_r(\ddot{\underline{e}}_r + K_r\dot{\underline{e}}_r) = \underline{e}_z \tag{34}$$

$$M_r\mu J(\ddot{\underline{e}}_z + K_z\dot{\underline{e}}_z) + [M_r+J]\underline{e}_z = \underline{0} \tag{35}$$

Finally, combination of (34) and (35) leads to the equivalent error equations of the total closed–loop system:

$$[M_r+J](\ddot{\underline{e}}_r + K_r\dot{\underline{e}}_r) + \mu J(\ddot{\underline{e}}_z + K_z\dot{\underline{e}}_z) = \underline{0} \tag{36}$$

F.4.1 Reformulation.

Assume measurement of the state variables:
$$\underline{y}(t) = \begin{bmatrix} \underline{y}_1(t) \\ \underline{y}_2(t) \end{bmatrix},$$

where
$$\underline{y}_1 = \begin{bmatrix} \underline{x} \\ \underline{\dot{x}} \end{bmatrix}, \qquad \underline{y}_2 = \begin{bmatrix} \underline{z} \\ \mu\underline{\dot{z}} \end{bmatrix}. \tag{37}$$

Then, the flexible manipulator system (18)–(19) can be rewritten as:

$$\boxed{\begin{aligned} \underline{\dot{y}}_1 &= S_{11}\underline{y}_1 + S_{12}\underline{y}_2 + S_{1g}\underline{g} \\ \underline{\dot{y}}_2 &= S_{21}\underline{y}_1 + S_{22}\underline{y}_2 + S_{2g}\underline{g} + S_u\underline{u} \end{aligned}} \qquad \begin{aligned}(38)\\(39)\end{aligned}$$

where
$$S_{11} = \begin{bmatrix} 0 & I \\ 0 & -M_r^{-1}C \end{bmatrix}, \qquad S_{12} = \begin{bmatrix} 0 & 0 \\ M_r^{-1} & 0 \end{bmatrix},$$
$$S_{21} = \begin{bmatrix} 0 & 0 \\ 0 & (M_r^{-1}C - J^{-1}B) \end{bmatrix}, \qquad S_{22} = \begin{bmatrix} 0 & K \\ (J^{-1}+M_r^{-1}) & -J^{-1}B \end{bmatrix},$$
$$S_{1g} = S_{2g} = \begin{bmatrix} 0 \\ M_r^{-1} \end{bmatrix}, \qquad S_u = \begin{bmatrix} 0 \\ J^{-1} \end{bmatrix}.$$

Further, we define the total error vector
$$\underline{e}_y(t) = \begin{bmatrix} \underline{e}_{y1}(t) \\ \underline{e}_{y2}(t) \end{bmatrix},$$

where
$$\underline{e}_{y1} = \begin{bmatrix} \underline{e}_r \\ \underline{\dot{e}}_r \end{bmatrix}, \qquad \underline{e}_{y2} = \begin{bmatrix} \underline{e}_z \\ \mu\underline{\dot{e}}_z \end{bmatrix}. \tag{40}$$

Finally, the equivalent error equations of the closed–loop system can be rescribed in the next form:

$$\boxed{\underline{\dot{e}}_y = \begin{bmatrix} \underline{\dot{e}}_{y1} \\ \underline{\dot{e}}_{y2} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & -K_r & M_r^{-1} & 0 \\ 0 & 0 & 0 & K \\ 0 & 0 & -(J^{-1}+M_r^{-1}) & -K_z \end{bmatrix} \underline{e}_y = A\underline{e}_y .} \tag{41}$$

F.5 STABILITY OF THE OVERALL CONTROL SYSTEM.

In anticipation on an adaptive flexible control system, Lyapunov's stability approach shall be combined with the Hyperstability theory of Popov [1969] for generalisation.

Define a Lyapunov function candidate:
where P is a positive definite matrix.

$$V = \frac{1}{2}\underline{e}_y^T P \underline{e}_y \tag{42}$$

Differentiating of V(t) gives:

$$\dot{V} = \frac{1}{2}\underline{e}_y^T \dot{P} \underline{e}_y + \underline{e}_y^T P \dot{\underline{e}}_y \,. \tag{43}$$

Using error equations (41) leads to:

$$\dot{V} = \frac{1}{2}\underline{e}_y^T \dot{P} \underline{e}_y + \underline{e}_y^T P A \underline{e}_y = \frac{1}{2}\underline{e}_y^T \{ \dot{P} + PA + A^T P \} \underline{e}_y \,. \tag{44}$$

*) To guarantee the global stability of the overall control system, the time derivative of V(t) has to be semi–negative definite according to Lyapunov's stability approach:

$$\dot{V} = -\underline{e}_y^T Q \underline{e}_y \leq 0 \tag{45}$$

where Q is a positive definite matrix.

*) For a more general approach in deriving a globally stable control law $\underline{u}(t)$, it is however better to use the Hyperstability theory of Popov, which requires that:

$$\dot{V} = -\underline{e}_y^T Q \underline{e}_y + \underline{e}_y^T \underline{w} \tag{46}$$

$$\int_{t_o}^{t} \underline{e}_y^T(\tau)\underline{w}(\tau)d\tau \leq \gamma_o^2$$

with finite γ and where $\underline{w} = \frac{1}{2}\{ \dot{P} + PA + A^T P \} \underline{e}_y$.

F.6 ADAPTIVE FLEXIBLE COMPUTED TORQUE CONTROL.

If there are unknown system parameters p_i in $M_r(\underline{p})$ and $C(\underline{p})$, an adaptive control algorithm will adjust them on–line and put them into the vector $\hat{\underline{p}}(t)$.

Instead of (24), the reference manifold $\underline{z}_r$ will then be approximated by:

$$\hat{\underline{z}}_r = \hat{M}_r(\underline{x},\hat{\underline{p}})(\ddot{\underline{x}}_r + K_r\dot{\underline{e}}_r) + \hat{C}(\underline{x},\dot{\underline{x}},\hat{\underline{p}})\dot{\underline{x}} + \hat{\underline{g}}(\underline{x},\hat{p}) \tag{47}$$

and the adaptive flexible computed torque control law will be:

$$\boxed{\underline{u} = [\hat{M}_r + J](\ddot{\underline{x}}_r + K_r\dot{\underline{e}}_r) + [\hat{C}+B]\dot{\underline{x}} + \hat{\underline{g}} + \mu\{ J(\ddot{\hat{\underline{z}}}_r + K_z\dot{\hat{\underline{e}}}_z) + B\dot{\underline{z}} \}} \tag{48}$$

where $\hat{\underline{e}}_z = \hat{\underline{z}}_r - \underline{z}$ is the adaptive reference manifold error (49)

Because there exists the linear relationship:

$$[\hat{M}_r - M_r](\ddot{\underline{x}}_r + K_r\dot{\underline{e}}_r) + [\hat{C}-C]\dot{\underline{x}} + [\hat{\underline{g}}-\underline{g}] = W(\underline{x},\dot{\underline{x}},\ddot{\underline{x}}_r)[\hat{\underline{p}}-\underline{p}] \tag{50}$$

we can obtain the next equivalent error equations of the two closed–loop subsystems:

(18)–(47): $M_r\ddot{\underline{x}} + C\dot{\underline{x}} + \underline{g} = z = \hat{z}_r - \hat{e}_z = \hat{M}_r(\ddot{\underline{x}}_r + K_r\dot{\underline{e}}_r) + \hat{C}\dot{\underline{x}} + \hat{\underline{g}} - e_z$

$$\longrightarrow \quad M_r(\ddot{\underline{e}}_r + K_r\dot{\underline{e}}_r) + W\underline{e}_p = \hat{\underline{e}}_z \tag{51}$$

(18)–(32): $\mu\{ J\ddot{\underline{z}} + B\dot{\underline{z}} \} + J\ddot{\underline{x}} + B\dot{\underline{x}} + \underline{z} = \underline{u}$

(47)–(48): $\underline{u} = \mu\{ J(\ddot{\hat{\underline{z}}}_r + K_z\dot{\hat{\underline{e}}}_z) + B\dot{\underline{z}} \} + J(\ddot{\underline{x}}_r + K_r\dot{\underline{e}}_r) + B\dot{\underline{x}} + \hat{\underline{z}}_r$

$$\longrightarrow \quad \mu J(\ddot{\hat{\underline{e}}}_z + K_z\dot{\hat{\underline{e}}}_z) + J(\ddot{\underline{e}}_r + K_r\dot{\underline{e}}_r) + \hat{\underline{e}}_z = \underline{0} \tag{52}$$

Finally, with (51) and (52) we have the total equaivalent error equations:

$$\boxed{[M_r + J](\ddot{\underline{e}}_r + K_r\dot{\underline{e}}_r) + \mu J(\ddot{\hat{\underline{e}}}_z + K_z\dot{\hat{\underline{e}}}_z) + W\underline{e}_p = \underline{0}} \tag{53}$$

where $\underline{e}_p = \hat{\underline{p}} - \underline{p}$ is the adapted parameter error. (54)

To guarantee global stability of the controlled system, the adaptation algorithm for on–line estimation of the system parameters has to be derived by using the Hyperstability theory of Popov, as described short in the previous section. Further investigations on this subject will follow.

F.7.1 Conclusions.

The main problem in control of flexible robots, namely the number of control inputs being less than the number of controlled variables, has been faced by a composite control law consisting of the conventional 'rigid' computed torque controller and a 'flexible' computed torque part multiplicated with the inverse of the stiffness matrix. The resultant control system resembles the socalled 'two–time scale sliding control' technique of Slotine & Hong [1987], based on a singular perturbation formulation of the equations of motion and the concept of integral manifold. Fortunately, in this approach the stiffness of each elastic joint does not have to be relatively large neither is there the restriction of Slotine and Hong that there have to be as many elastic joints as motor inputs (no flexible links, no rigid motor joints). Both methods require all system state variables (positions and velocities) for feedback. But then, Lyapunov's stability theorem guarantees that the output trajectory will follow the desired trajectory and that the elastic forces/torques, which are not directly constrained by the output specifications, remain on a certain 'manifold', due to the natural flexibility behavior of the system. In the next chapter, the key concept will be illustrated with simulation results of a Translation–Rotation (TR–) robot with one elastic joint.

In first instance, the motion control of a manipulator with elastic joints based on precise knowledge of the system parameters is considered. Due to parameter uncertainties and/or variations, it is not possible in practice to exactly compensate the manipulator dynamics. Therefore, the earlier mentioned flexible computed torque control method is finally adapted in an adaptive flexible control algorithm, based on the 'rigid' adaptive technique of Slotine & Li [1987]. Stability will be guaranteed by the Hyperstability approach of Popov [1969].

Future research.

While the extension of the computed torque control (CTC) technique to joint–level control of flexible manipulators seems to be quite straightforward, a more involved situation arises for the end–effector trajectory control. The number of outputs is taken equal to the number of available inputs, but if there are more joints than actuators (especially in case of approximated link flexibility) the basic limitation is due to the noncollocation of actuators and controlled outputs. For this class of manipulators, the knowledge of the desired trajectory and of its time derivatives is not enough to determine the required forces/torques instantaneously. Instead, a dynamic inverse system has to be used on–line to generate the natural 'reference' behavior of all system state variables due to the desired output trajectory. To address this problem, future research in 1991 will be concentrated on the development of an output–trajectory based version of the (adaptive) flexible CTC controller.

References:

* **Asada, H. and Slotine, J.–J. E.** **(1986)**
Robot analysis and control..
John Wiley & Sons.

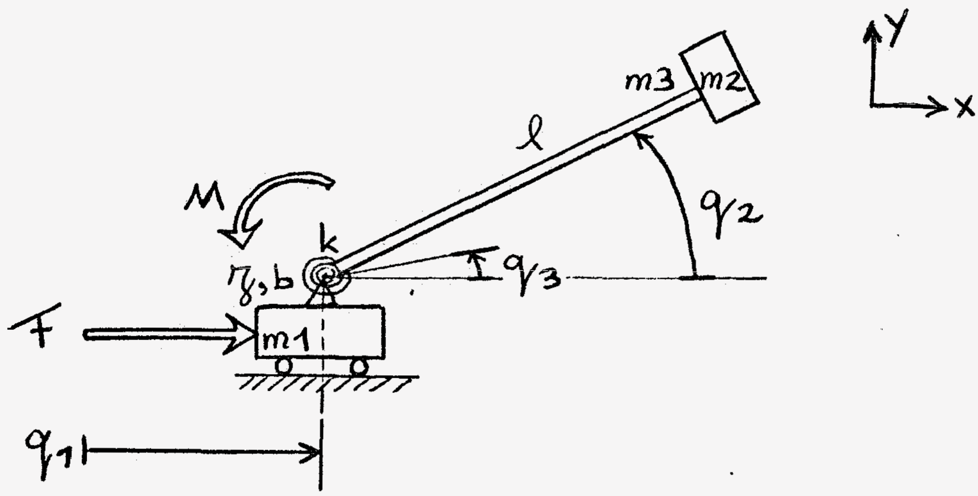
* **Popov, V.M.** **(1969)**
Hyperstability of control system.
New York, Springer–Verlag.

* **Slotine, J.–J. E. and Hong, S.** **(1987)**
Two–time scale sliding control of manipulators with flexible joints.
American Control Conference, BSE: DPC 82 AME.

* **Slotine, J.–J. E. and Li, W.** **(1987)**
On the adaptive control of robot manipulators.

Control simulation results of a

G | 3 Degrees-of-freedom (D.O.F.) elastic-joint Translation-Rotation (TR-) robot with 2 motor input signals.



G.1 The equations of motion are:

$$A_{11}\ddot{q}_1 - A_{12}\sin(q_2)\ddot{q}_2 - A_{12}\cos(q_2)\dot{q}_2^2 = F$$
$$-A_{12}\sin(q_2)\ddot{q}_1 + A_{22}\ddot{q}_2 - k(q_3 - q_2) = 0$$
$$A_{33}\ddot{q}_3 + b\dot{q}_3 + k(q_3 - q_2) = M$$

$A_{11} = m_1 + m_2 + m_3$, $A_{22} = (m_2 + m_3/3)l^2$,
$A_{12} = (m_2 + m_3/2)l$, $A_{33} = J_g$.

- mass of the carriage $m_1 = 10$ [kg]
- mass of the payload $m_2 = 2$ [kg]
- mass of the arm $m_3 = 3$ [kg]
- length of the arm $l = 0.75$ [m]
- stiffness of the elastic joint $k = 2$ [Nm/rad]
- inertia moment of the motor rotor $J_g = 5$ [kgm^2]
- friction constant of the motor $b = 0.5$ [Nm].

G.2 2 D.O.F. rigid TR-robot :

$$\boxed{M_r(q_r)\ddot{q}_r + \underline{n}_r(q_r, \dot{q}_r) = \underline{u}} \qquad (1)$$

where

$$q_r = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \underline{u} = \begin{bmatrix} F \\ M \end{bmatrix}, \quad M_r = \begin{bmatrix} M_{11} & M_{12} \\ M_{12} & M_{22}+M_{33} \end{bmatrix}, \quad \underline{n}_r = \begin{bmatrix} n_1 \\ n_2+n_3 \end{bmatrix}.$$

$M_{11} = A_{11}$
$M_{12} = -A_{12}\sin(q_2)$
$M_{22} = A_{22}$

$n_1 = -A_{12}\cos(q_2)\dot{q}_2^2$
$n_2 = \emptyset$

G.2.1 * PID - control :

$$\boxed{\underline{u} = \ddot{q}_{rd} + K_d\underline{\dot{e}} + K_p\underline{e} + K_i\int \underline{e}\,dt} \qquad (2)$$

where $q_{rd}(t)$ is the desired joint-trajectory $q_{rd}[y_d(t)]$,
$y_d(t)$ is the desired output-trajectory $y_d = [x_d \; y_d]^T$,
$\underline{e} = q_{rd} - q_r$ is the joint-based tracking error,
K_d, K_p, K_i are positive-definite diagonal matrices.

G.2.2 * RIGID Computed - Torque Control :

$$\boxed{\underline{u} = M_r(q_r)[\ddot{q}_{rs} + K_s\underline{\dot{e}}_s] + \underline{n}_r(q_r, \dot{q}_r)} \qquad (3)$$

where $q_{rs} = q_{rd} + \Lambda\int \underline{e}\,dt$ is a sliding reference joint-trajector
$\underline{e}_s = q_{rs} - q_r$ is the sliding reference error,
K_s is a positive-definite diagonal matrix.

The equivalent error equations of the closed-loop system are:

$$\boxed{M_r[\underline{\ddot{e}}_s + K_s\underline{\dot{e}}_s] = \emptyset} \qquad (4)$$

G.3 3 D.O.F. elastic-joint TR-robot:

$$\boxed{M(q)\ddot{q} + \underline{n}(q,\dot{q}) = H\underline{u}} \qquad (5)$$

where

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad \underline{u} = \begin{bmatrix} F \\ M \end{bmatrix}, \quad M = \begin{bmatrix} M_{11} & M_{12} & \emptyset \\ M_{12} & M_{22} & \emptyset \\ \emptyset & \emptyset & M_{33} \end{bmatrix},$$

$$\underline{n} = \begin{bmatrix} n_1 \\ n_2 - k(q_3-q_2) \\ n_3 + k(q_3-q_2) \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

$M_{11} = A_{11}$
$M_{12} = -A_{12}\sin(q_2)$
$M_{22} = A_{22}$
$M_{33} = A_{33}$

$n_1 = -A_{12}\cos(q_2)\dot{q}_2^2$
$n_2 = \emptyset$
$n_3 = b\dot{q}_3$

$$\begin{cases} M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 + n_1 = F & (6) \\ M_{12}\ddot{q}_1 + M_{22}\ddot{q}_2 + n_2 = k(q_3-q_2) & (7) \\ k(q_3-q_2) + M_{33}\ddot{q}_3 + n_3 = M & (8) \end{cases}$$

$$M_{12}\ddot{q}_1 + M_{22}\ddot{q}_2 + M_{33}\ddot{q}_3 + n_2 + n_3 = M \qquad (9)$$

$k \to \emptyset$ superflexibility: $[q_3-q_2] \to \infty$
$k \to \infty$ rigidness: $[q_3-q_2] \to \emptyset$

$\downarrow k.(q_3-q_2)\uparrow$
$\uparrow k.(q_3-q_2)\downarrow$

Question: does the magnitude of the elastic-joint torque $z = k(q_3-q_2)$ ever remains on a certain 'manifold' z_m?
Yes, it depends from the actual joint-trajectory $q(t)$, substituted in (7).

Definition of the elastic-joint torque $z = k(q_3-q_2)$ (10)

By substitution of the new variable z for q_3 in (8) we obtain:

$$z + M_{33}\ddot{q}_2 + \frac{1}{k}M_{33}\ddot{z} + n_3 = M \qquad (11)$$

G.3.1. * RIGID Computed-Torque Control (rigid CTC):

$$\boxed{\underline{u} = M_r(q_r)[\ddot{q}_{rs} + K_s \dot{\underline{e}}_s] + \underline{n}_r(q_r, \dot{q}_r)} \quad (3)$$

(6) $M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 + n_1 = F \quad = M_{11}[\ddot{q}_{s1} + k_{s11}\dot{e}_{s1}] + M_{12}[\ddot{q}_{s2} + k_{s12}\dot{e}_{s2}] + n_1$ (12)

(7) $M_{12}\ddot{q}_1 + M_{22}\ddot{q}_2 + n_2 = Z_r - e_z = M_{12}[\ddot{q}_{s1} + k_{s21}\dot{e}_{s1}] + M_{22}[\ddot{q}_{s2} + k_{s22}\dot{e}_{s2}] + n_2 - e_z$ (13)

(11) $M_{33}\ddot{q}_2 + \frac{1}{k}M_{33}\ddot{z} + n_3 + z = M = \quad Z_r + M_{33}[q_{s2} + k_{s22}\dot{e}_{s2}] + n_3$ (14)

The equivalent error equations of the closed-loop system are:

(12) $M_{11}[\ddot{e}_{s1} + k_{s11}\dot{e}_{s1}] + M_{12}[\ddot{e}_{s2} + k_{s12}\dot{e}_{s2}] = \emptyset$ (15)

(13) $M_{12}[\ddot{e}_{s1} + k_{s21}\dot{e}_{s1}] + M_{22}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}] = e_z$ (16)

(14) $e_z + \frac{1}{k}M_{33}\ddot{z} + M_{33}[\ddot{e}_{s2} + k_{s2}\dot{e}_{s2}] = \emptyset$ (17)

$$\boxed{M_r[\ddot{\underline{e}}_s + K_s\dot{\underline{e}}_s] + \begin{bmatrix} \emptyset \\ \frac{1}{k}M_{33}\ddot{z} \end{bmatrix} = \emptyset} \quad (18)$$

Stable if $k \to \infty$.

(Figures [3] and [4]).

G.3.2 * FLEXIBLE Computed-Torque Control (flexible CTC)

G.3.2.1

$M\{z_r\}$

$$\underline{u} = M_r(q_r)[\ddot{\underline{q}}_{rs} + K_s \dot{\underline{e}}_s] + \underline{n}_r(q_r, \dot{q}_r) + \begin{bmatrix} \emptyset \\ \frac{1}{k} M_{33} [\ddot{z}_r + k_z \dot{e}_z] \end{bmatrix}$$

$$= \underline{u}_r + \cancel{\mu}\, \underline{u}_f \qquad (19)$$

(6) $M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 + n_1 = F = M_{11}[\ddot{q}_{s1} + k_{s11}\dot{e}_{s1}] + M_{12}[\ddot{q}_{s2} + k_{s12}\dot{e}_{s2}] + n_1$ (12

(7) $M_{12}\ddot{q}_1 + M_{22}\ddot{q}_2 + n_2 = z_r - e_z = M_{12}[\ddot{q}_{s1} + k_{s21}\dot{e}_{s1}] + M_{22}[\ddot{q}_{s2} + k_{s22}\dot{e}_{s2}] + n_2 - e_z$ (13

(11) $M_{33}\ddot{q}_2 + \frac{1}{k}M_{33}\ddot{z} + n_3 + z = M = \frac{1}{k}M_{33}[\ddot{z}_r + k_z\dot{e}_z] + M_{33}[\ddot{q}_{s2} + k_{s22}\dot{e}_{s2}] + n_3 + z_r$ (20

The equivalent error equations of the closed-loop system are:

$$M_{11}[\ddot{e}_{s1} + k_{s11}\dot{e}_{s1}] + M_{12}[\ddot{e}_{s2} + k_{s12}\dot{e}_{s2}] = \emptyset \qquad (15)$$

$$M_{12}[\ddot{e}_{s1} + k_{s21}\dot{e}_{s1}] + M_{22}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}] = e_z \qquad (16)$$

$$\frac{1}{k}M_{33}[\ddot{e}_z + k_z\dot{e}_z] + M_{33}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}] + e_z = \emptyset \qquad (21)$$

$$\boxed{M_{FL}[\ddot{\underline{e}}_{FL} + K_{FL}\dot{\underline{e}}_{FL}] = \underline{\emptyset}} \qquad (22)$$

where

$$M_{FL} = \begin{bmatrix} M_{11} & M_{12} & \emptyset \\ M_{12} & M_{22}+M_{33} & \frac{1}{k}M_{33} \end{bmatrix}, \quad \underline{e}_{FL} = \begin{bmatrix} e_{s1} \\ e_{s2} \\ e_z \end{bmatrix},$$

$$K_{FL} = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{K_s} & \begin{array}{c}\emptyset\\ \emptyset\end{array} \\ \hline \emptyset & \emptyset & k_z \end{array}\right]$$ is a positive-definite diagonal matrixe.

Figures [5], [6], [7] and [8])

The equations of motion:

$$
\begin{aligned}
&(6) && M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 + n_1 = F \\
&(7) && \tfrac{1}{k}M_{12}\ddot{q}_1 + \tfrac{1}{k}M_{22}\ddot{q}_2 + \tfrac{1}{k}n_2 + q_2 = \boxed{q_3 = q_{r3} - e_{r3}} \\
&(8) && k(q_3 - q_2) + M_{33}\ddot{q}_3 + n_3 = M
\end{aligned}
$$

G.3.2.2 * FLEXIBLE C.T.C. with q_{r3} instead of z_r:

$M\{q_{r3}\}$

$$q_{r3} = \tfrac{1}{k}M_{12}[\ddot{q}_{s1} + k_{s1}\dot{e}_{s1}] + \tfrac{1}{k}M_{22}[\ddot{q}_{s2} + k_{s22}\dot{e}_{s2}] + \tfrac{1}{k}n_2 + q_2 \qquad (23)$$

$$\boxed{\begin{aligned}
F &= M_{11}[\ddot{q}_{s1} + k_{s11}\dot{e}_{s1}] + M_{12}[\ddot{q}_{s2} + k_{s12}\dot{e}_{s2}] + n_1 && (12)\\
M &= k(q_{r3} - q_2) + M_{33}[\ddot{q}_{r3} + k_{33}\dot{e}_{r3}] + n_3 && (24)
\end{aligned}}$$

The equivalent error equations of the closed-loop system are:

$$
\begin{aligned}
M_{11}[\ddot{e}_{s1} + k_{s11}\dot{e}_{s1}] + M_{12}[\ddot{e}_{s2} + k_{s12}\dot{e}_{s2}] &= \emptyset && (15)\\
M_{12}[\ddot{e}_{s1} + k_{s21}\dot{e}_{s1}] + M_{22}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}] &= k.e_{r3} && (25)\\
k.e_{r3} \qquad + M_{33}[\ddot{e}_{r3} + k_{r33}\dot{e}_{r3}] &= \emptyset && (26)
\end{aligned}
$$

$$\boxed{M_F[\ddot{\underline{e}}_F + K_F\dot{\underline{e}}_F] = \emptyset} \qquad (27)$$

where

$$M_F = \begin{bmatrix} M_{11} & M_{12} & \emptyset \\ M_{12} & M_{22} & M_{33} \end{bmatrix}, \quad \underline{e}_F = \begin{bmatrix} e_{s1} \\ e_{s2} \\ e_{r3} \end{bmatrix}, \quad K_F = \begin{bmatrix} K_s & \begin{matrix}\emptyset \\ \emptyset\end{matrix} \\ \emptyset\,\emptyset & k_{r33} \end{bmatrix}.$$

(Figures [11], [12] and [13])

G.3.2.3 $M\{z\}$

Instead of (20): $M = \frac{1}{k}M_{33}[\ddot{z}_r + k_z \dot{e}_z] + M_{33}[\ddot{q}_{s2} + k_{s22}\dot{e}_{s2}] + n_3 + \underline{\underline{z_r}}$

$$\boxed{M = \frac{1}{k}M_{33}[\ddot{z}_r + k_z \dot{e}_z] + M_{33}[\ddot{q}_{s2} + k_{s22}\dot{e}_{s2}] + n_3 + \underline{\underline{z}}} \tag{28}$$

(Figure [9])

The equivalent error equations of the closed-loop system are:

$$M_{11}[\ddot{e}_{s1} + k_{s11}\dot{e}_{s1}] + M_{12}[\ddot{e}_{s2} + k_{s12}\dot{e}_{s2}] = \emptyset \tag{15}$$

$$M_{12}[\ddot{e}_{s1} + k_{s21}\dot{e}_{s1}] + M_{22}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}] = e_z \tag{16}$$

$$\frac{1}{k}M_{33}[\ddot{e}_z + k_z \dot{e}_z] + M_{33}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}] = \emptyset \tag{29}$$

$$\boxed{M_{FLZ}\left[\ddot{\underline{e}}_{FLZ} + K_{FLZ}\dot{\underline{e}}_{FLZ}\right] = \begin{bmatrix} \emptyset \\ e_z \\ \emptyset \end{bmatrix}} \tag{30}$$

where $M_{FLZ} = \begin{bmatrix} M_{11} & M_{12} & \emptyset \\ M_{12} & M_{22} & \emptyset \\ \emptyset & \emptyset & M_{33}/k \end{bmatrix}$; $\underline{e}_{FLZ} = \begin{bmatrix} e_{s1} \\ e_{s2} \\ e_z \end{bmatrix}$;

$K_{FLZ} = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{K_s} & \begin{matrix}\emptyset \\ \emptyset\end{matrix} \\ \hline \emptyset & \emptyset & k_z \end{array}\right]$ is a positive-definite diagonal matrix.

G.3.2.4 $M\{q_3\}$

Instead of (24): $M = k(\underline{\underline{q_{r3}}} - q_2) + M_{33}[\ddot{q}_{r3} + k_{r33}\dot{e}_{r3}] + n_3$

$$\boxed{M = k(q_3 - q_2) + M_{33}[\ddot{q}_{r3} + k_{r33}\dot{e}_{r3}] + n_3} \tag{31}$$

Then, instead of (26): $M_{33}[\ddot{e}_{r3} + k_{r33}\dot{e}_{r3}] = \emptyset$ (32)

No coupling with (25)

(Figures [14] and [15]).

TUE-WFW-IL
11-1990

G-7

G.4 **Control simulation results on the 3 D.O.F. elastic–joint TR–robot.**

*) The initial system state variables $q_1(t_0)$ and $q_2(t_0)$ are due to the desired trajectory at $t=t_0$: $q_{1d}(t_0)$ resp. $q_{2d}(t_0)$.

*) If nothing else is mentioned: $K_s = \mathrm{diag}[10]$, $k_z = 10$.

page:

Figure [1]: **Open–loop control:** $\underline{u} = \underline{0}$. — 10

Figure [2]: **PD–control:** $\underline{u} = K_p.\underline{e}_y + K_d.\dot{\underline{e}}_y$,
$\underline{e}_y = \underline{y}_d - \underline{y}$, $\underline{y} = [x\ y]^T$, $K_p = K_d = \mathrm{diag}[10]$.

Rigid Computed Torque Control (3):

Figure [3]: $K_s = 0$.
Figure [4]: $K_s = \mathrm{diag}[10]$.

Flexible Computed Torque Control (19) [(12)–(20)]:

Figures [5] *and* [7]: $K_s = 0$. — 11 $\boxed{M\{z_r\}}$
Figures [6] *and* [8]: $Ks = \mathrm{diag}[10]$.

Figure [9]: Flexible CTC (12)–(28). — 12 $\boxed{M\{z\}}$
Figure [10]: Flexible CTC (31): 13

$$q_{r3} = \frac{1}{k}[M_{12}(\underline{q}_d)\ddot{q}_{1d} + M_{22}(\underline{q}_d)\ddot{q}_{2d} + n_2(\underline{q}_d,\dot{\underline{q}}_d)] + q_{2d}.$$

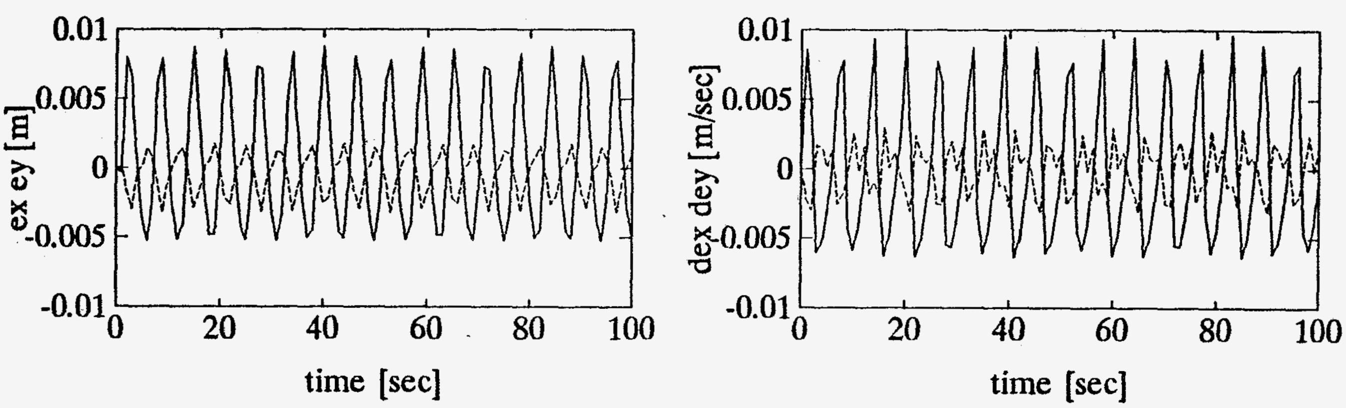
Flexible CTC (12)–(24), q_{r3} (23): $\boxed{M\{q_{r3}\}}$

Figure [11]: $K_s = \mathrm{diag}[10]$, $k_z = 10$.
Figure [12]: $K_s = 0$, $k_z = 10$. — 14
Figure [13]: $K_s = \mathrm{diag}[10]$, $k_z = 0$.

Flexible CTC (12)–(31), q_{r3} (25): $\boxed{M\{q_3\}}$

Figure [14]: $K_s = 0$, $k_z = 0$. — 16
Figure [15]: $K_s = \mathrm{diag}[10]$, $k_z = 10$.

Figure [16] : t = 100 sec.



G.5 Analyzation of the control simulation results on the 3 D.O.F. elastic-joint TR-robot.

$$F = M_{11} ddq_{11} + M_{12} ddq_{12} + n_1 \qquad (12)$$

Figure [9]: $M\{z\} = \frac{1}{k} M_{33} ddz_r + M_{33} ddq_{22} + n_3 + z$ (28)

Figures [6]/[8]: $M\{z_r\} = \frac{1}{k} M_{33} ddz_r + M_{33} ddq_{22} + n_3 + z_r =$ (20)

$$= M_{12} ddq_{21} + (M_{22} + M_{33}) ddq_{22} + n_2 + n_3 + \frac{1}{k} M_{33} ddz_r$$

Figure [15]: $M\{q_3\} = M_{33} ddq_3 + n_3 + k(q_3 - q_2)$ (31)

Figure [11]: $M\{q_{3r}\} = M_{33} ddq_3 + n_3 + k(q_{r3} - q_2) =$ (24)

$$= M_{12} ddq_{21} + M_{22} ddq_{22} + M_{33} ddq_3 + n_2 + n_3$$

Although there is a (disappearing) difference between the actual elastic-joint torque z and the computed reference torque z_r, it seems to be small enough to obtain the same simulation results either with the computed torque control signal $M\{z_r\}$ or with $M\{z\}$: see figures [6]-[8] resp. [9]. This can also be concluded for $M\{q_{r3}\}$ resp. $M\{q_3\}$.
Although there are thus no arguments obtained from these results to use either z_r/q_{r3} or z/q_3, I prefer the first alternative because with the second there is no direct coupling of the elastic-joint reference error e_z/e_{r3}, appearing in error equation (16)/(25), with (29)/(32).

Comparing computed torque method $M\{z_r\}$ in figures [6]-[8] with method $M\{q_{r3}\}$ in figure [11], there can be concluded that the magnitudes of the output errors e_x and e_y both remain between acceptable bounds (see also figure [16]: $t = 100$ sec.). But it is remarkable that there is needed a much larger control torque $M\{z_r\}$ in figure [8] than $M\{q_{3r}\}$ in [11] (just after $t = t_0$). Probably, it has to do with the relatively large (but decreasing) elastic-joint reference error $e_z = z_r - z$ in the same figure [8]. Because, after very short time, there is no visible elastic-joint reference error in figure [11], I think that this method (24) is better.

(13) $z_r = M_{12}\, ddq_{21} + M_{22}\, ddq_{22} + n_2$

(23) $q_{r3} = \frac{1}{k} M_{12}\, ddq_{21} + \frac{1}{k} M_{22}\, ddq_{22} + \frac{1}{k} n_2 + q_2$

$$\Downarrow$$

$$\boxed{z_r = k.(q_{r3} - q_2)} \qquad (33)$$

$$e_z = z_r - z = k(q_{r3} - q_2) - k(q_3 - q_2) = k(q_{r3} - q_3) = k.e_{r3} \qquad (34)$$

(20)
$$M\{z_r\} = \frac{1}{k} M_{33}[\ddot{z}_r + k_z \dot{e}_z] + M_{33}[\ddot{q}_{s2} + k_{s22}\dot{e}_{s2}] + n_3 + z_r =$$
$$= M_{33}[\ddot{q}_{r3} + k_z \dot{e}_{r3}] + M_{33}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}] + n_3 + k(q_{r3} - q_2) =$$
$$= M\{q_{r3}\} + M_{33}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}]$$

if $k_z = k_{r33}$ (35)

In this way, error equation (21) is equivalent to:

$$M_{33}[\ddot{e}_{r3} + k_z \dot{e}_{r3}] + M_{33}[\ddot{e}_{s2} + k_{s22}\dot{e}_{s2}] + k\dot{e}_{r3} = \emptyset \qquad (36)$$

and the total closed-loop error system becomes:

$$\boxed{M_{FLq}\{\ \underline{\ddot{e}}_{FLq} + K_{FLq}\underline{\dot{e}}_{FLq}\ \} = \underline{\emptyset}} \qquad (37)$$

where
$$M_{FLq} = \begin{bmatrix} M_{11} & M_{12} & \emptyset \\ M_{12} & M_{22}+M_{33} & M_{33} \end{bmatrix}, \quad \underline{e}_{FLq} = \begin{bmatrix} e_{s1} \\ e_{s2} \\ e_{r3} \end{bmatrix}, \quad K_{FLq} = \begin{bmatrix} k_{s11} & 0 & 0 \\ 0 & k_{s22} & 0 \\ 0 & 0 & k_z \end{bmatrix}.$$

If $k \to \infty$ (and thus $[q_3 - q_2] \to \emptyset$), we exactly obtain from (22) or (37) the error equations of the closed-loop <u>rigid</u> system (4)! This is not the case with $M\{q_{r3}\}$: equations (27). Is this the reason why Slotine & Hong [1987] use the 'manifold' $\underline{z}_r$ instead of a reference state vector $\underline{q}_r$?

STABILITY Proof of Lyapunov:
$$\boxed{V = \tfrac{1}{2}\,\underline{\dot{e}}^T_{FLq}\, P\, \underline{\dot{e}}_{FLq}} \qquad (38)$$

$P > 0$
$\dot{P} = 0$

From equation (37): $\underline{\ddot{e}}_{FLq} = -K_{FLq}\underline{\dot{e}}_{FLq}$

Differentiating $V(t)$: $\dot{V} = \underline{\dot{e}}^T_{FLq} P \underline{\ddot{e}}_{FLq} = -\underline{\dot{e}}^T_{FLq} P K_{FLq} \underline{\dot{e}}_{FLq} < \emptyset$.

$\underline{\dot{e}}_{FLq} \downarrow \underline{\emptyset}$

* $\underline{\dot{e}}_s = \underline{\dot{e}} + \Lambda \underline{e}$: $\underline{\dot{e}} \to \underline{\emptyset}$ and $\underline{e} \to \underline{\emptyset}$ $\Rightarrow$ trajectory tracking

* $\dot{e}_{r3} = \dot{q}_{r3} - \dot{q}_3 \to \underline{\emptyset} \Rightarrow [q_{r3} - q_3]$ remains bounded in time

Further conclusions with $M\{q_{3r}\}$.

$z(t)$ Referring again to the 'Singular Perturbation' Method of Slotine & Hong [1987], my conclusion is that, indeed, the elastic-joint torque $z(t)$ plays an important role in the flexible control system ($M\{z_r\}/M\{q_{3r}\}$), in that it depends from the system and the desired trajectory, but nót from the elastic-joint stiffness k [pages 13, 17] nór from any control parameters K_S, k_z [pages 13, 14, 15] (after very short 'starting-up time'): in any simulation test the magnitude of $z(t)$ remains on the same 'manifold' $z_m(t)$, as already stated at page 2 (under equation (3)). This, in contradiction with the link-angle q_3 [pages 13, 17].

k It is remarkable that the magnitude of the elastic-joint stiffness k does not influence the range in between which the output errors e_x and e_y fluctuate: it is always the same, until the control parameters are changed [compare page 15 with 17]. If k is alternated, only the differences between the motor angle q_2 and the arm rotation q_3 changes, as expected (page 2, under equation (3)), and a larger computed torque M is needed the smaller the elastic-joint stiffness k is (figure [17]). There seems to be no restriction to the magnitude of k using this control strategy, although Slotine & Hong stated that k has to be 'very large'.

Slotine & Hong and others using the 'Singular Perturbation' Approach always generalize their method for elastic-joint systems where there are twice as many degrees of freedom as motor input signals.
$(n = 2m)$ That is another restriction, while the flexible CTC-method simulated in this paper shows that not all joints (connections between motors and links) have to be elastic (or in case of rigidness: $k \to \infty$). Just split up the system in a rigid part $[q_r]$ and a flexible part $[q_f]$.

$\begin{pmatrix} n_r = m \\ n_f < m \end{pmatrix}$

G.6 Control simulation results on the 3 D.O.F. elastic–joint TR–robot.

Figure [1]: Open–loop control: $\underline{u} = \underline{0}$.

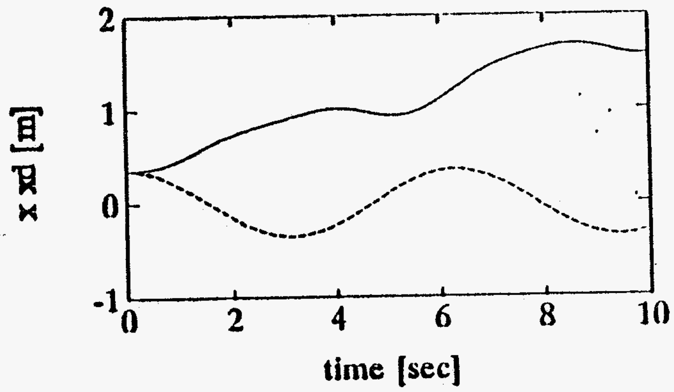


Figure [2]: PD–control: $\underline{u} = K_p.\underline{e}_y + K_d.\dot{\underline{e}}_y$, $\underline{e}_y = \underline{y}_d - \underline{y}$, $\underline{y} = [x\ y]^T$, $K_p = K_d = \text{diag}[10]$.

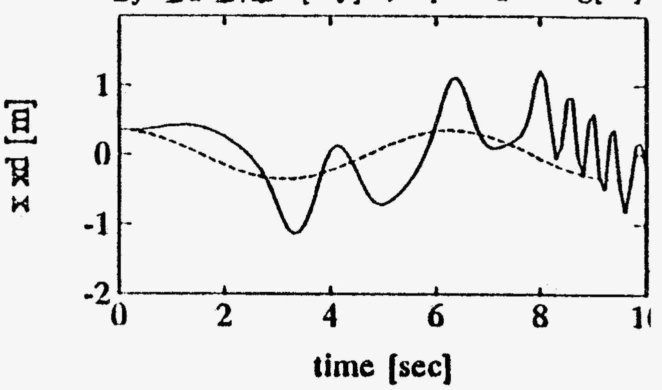


Figure [3]: Rigid CTC: $K_s = 0$.

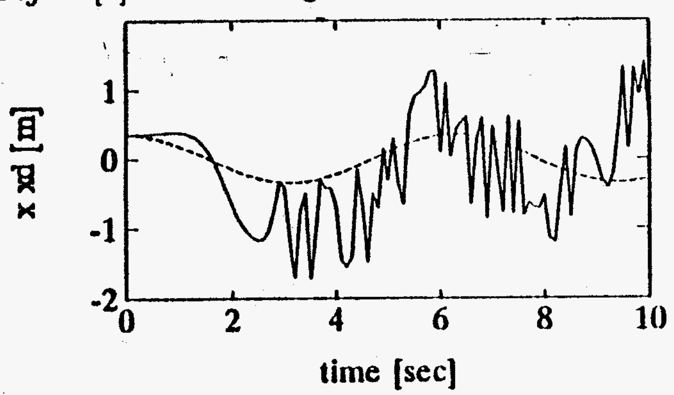


Figure [4]: Rigid CTC: $K_s = \text{diag}[10]$.

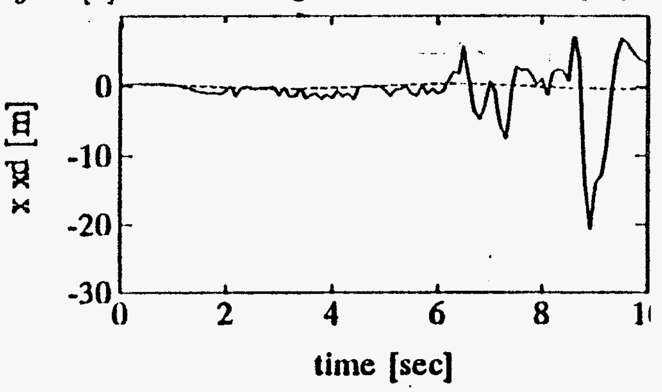


Figure [5]: Flexible CTC (19): $K_s = 0$ {+*fig*[7]}. M{zr}

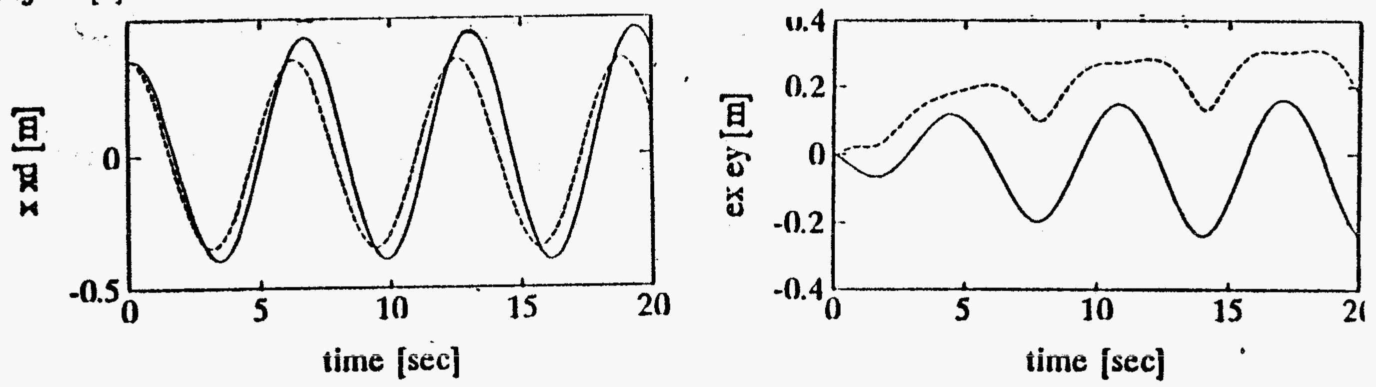
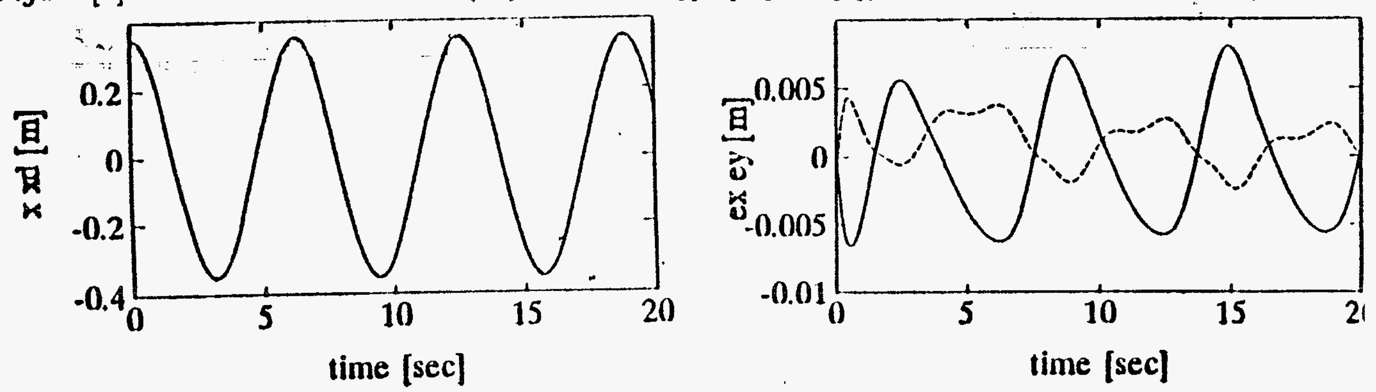


Figure [6]: Flexible CTC (19): $Ks = \text{diag}[10]$ {+*fig*[8]}. M{zr}



TUE - WFW - IL
11-1990

G - 11

Figure [7]: Flexible CTC (19): $K_s = 0$ {+*fig*[5]}. M{zr}

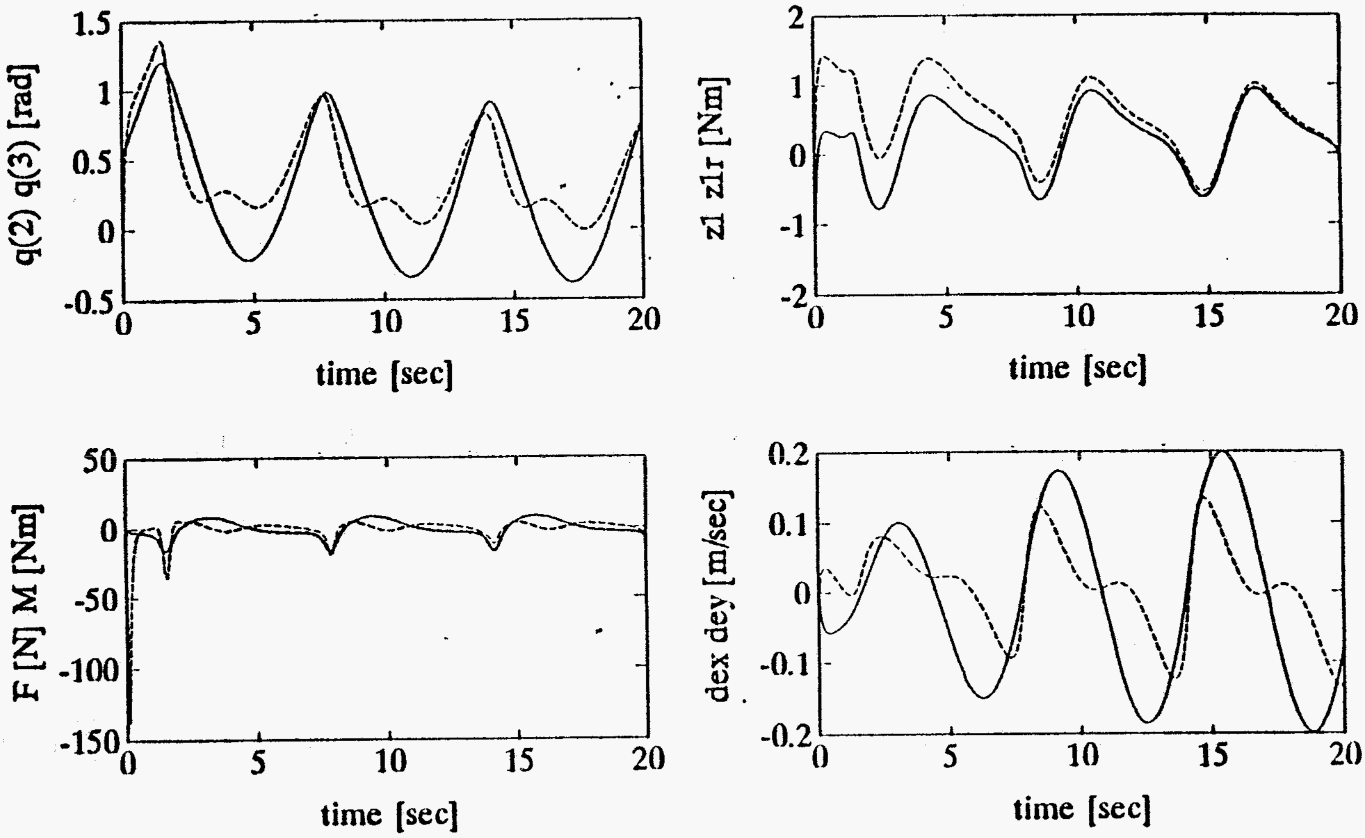
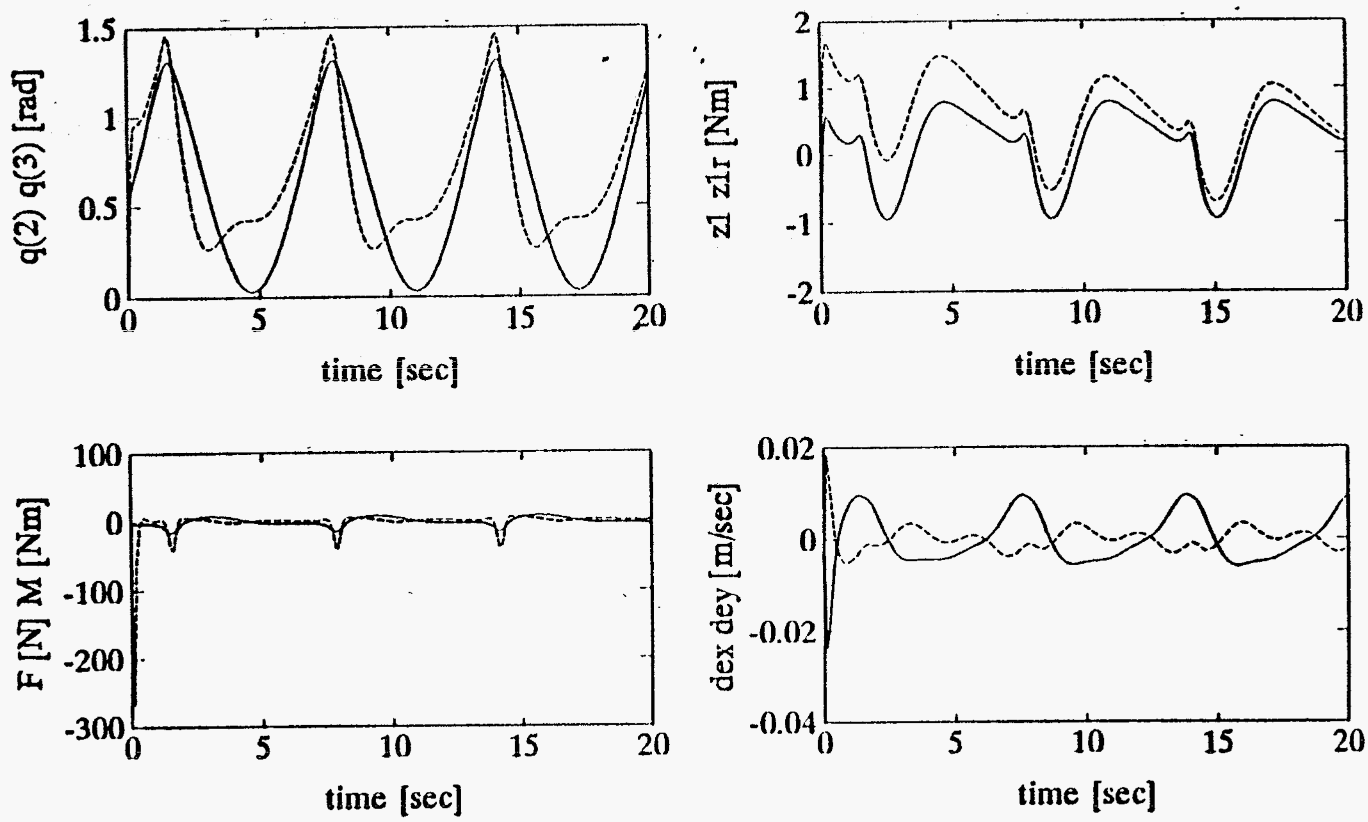


Figure [8]: Flexible CTC (19): $Ks = \text{diag}[10]$ {+*fig*[6]}. M{zr}



If nothing else is mentioned: $K_s = \mathrm{diag}[10]$, $k_z = 10$.

Figure [9]: Flexible CTC (12)–(28). M{z}

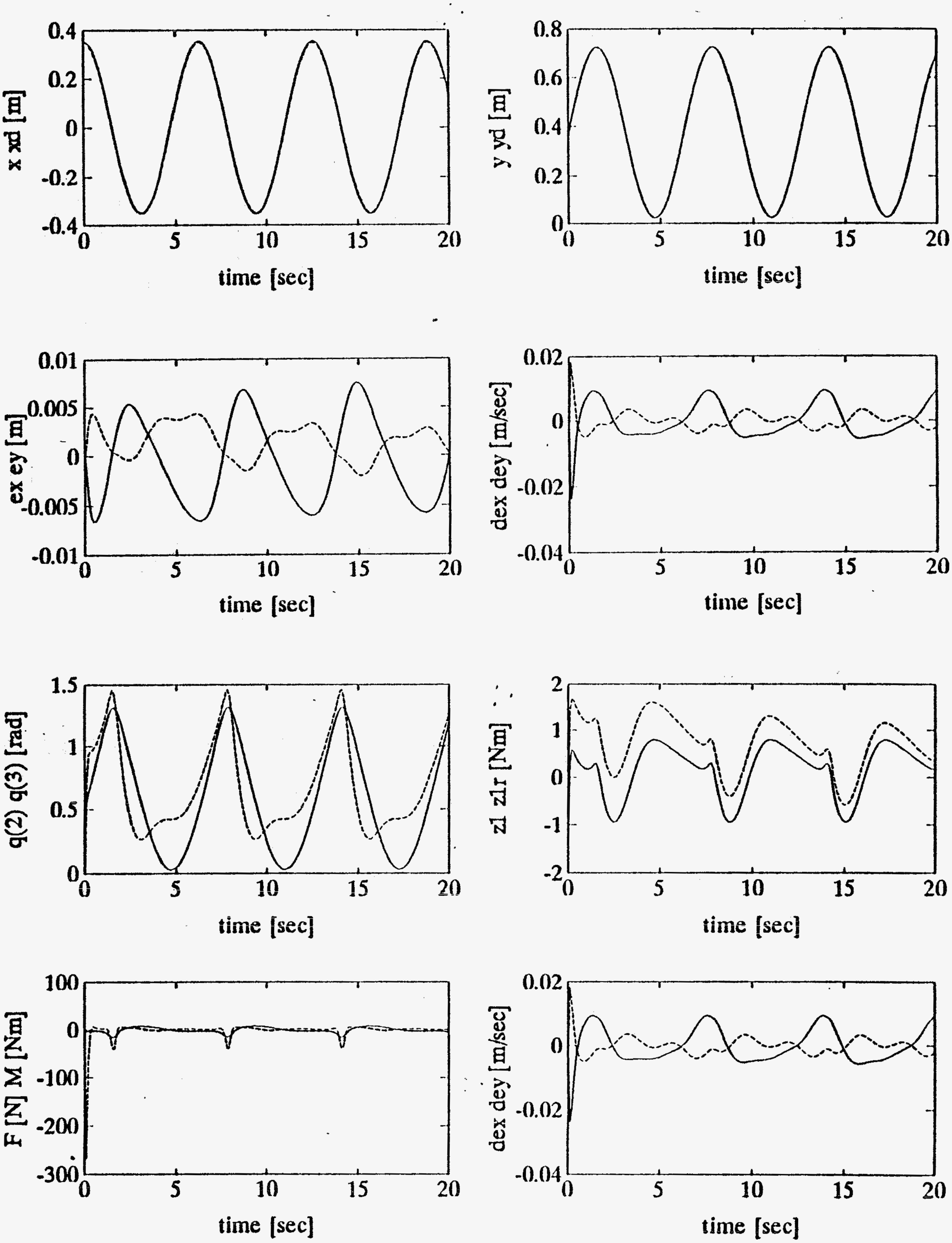


Figure [10]: Flexible CTC (31):

$$q_{r3} = \frac{1}{k}[M_{12}(\underline{q}_d)\ddot{q}_{1d}+M_{22}(\underline{q}_d)\ddot{q}_{2d}+n_2(\underline{q}_d,\dot{\underline{q}}_d)]+q_{2d}.$$

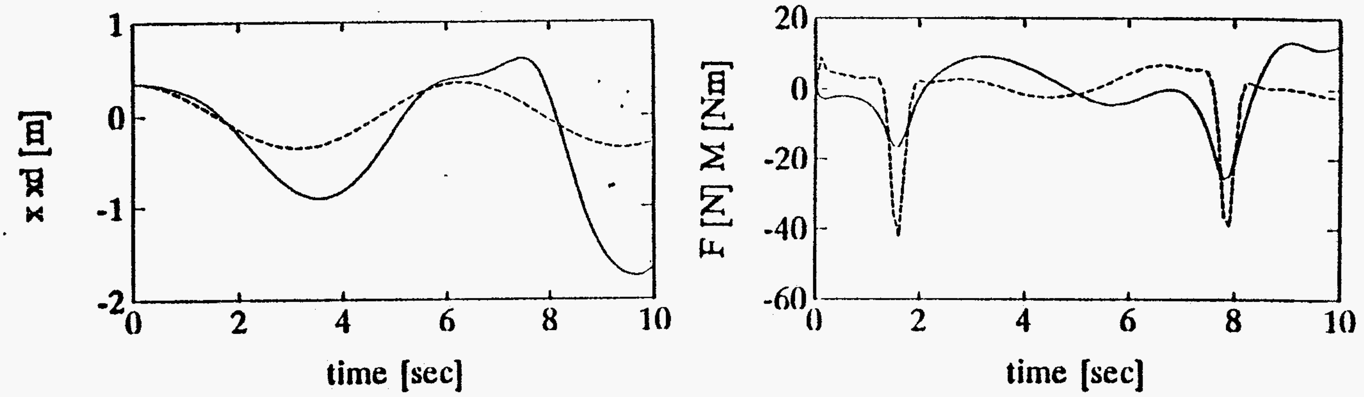
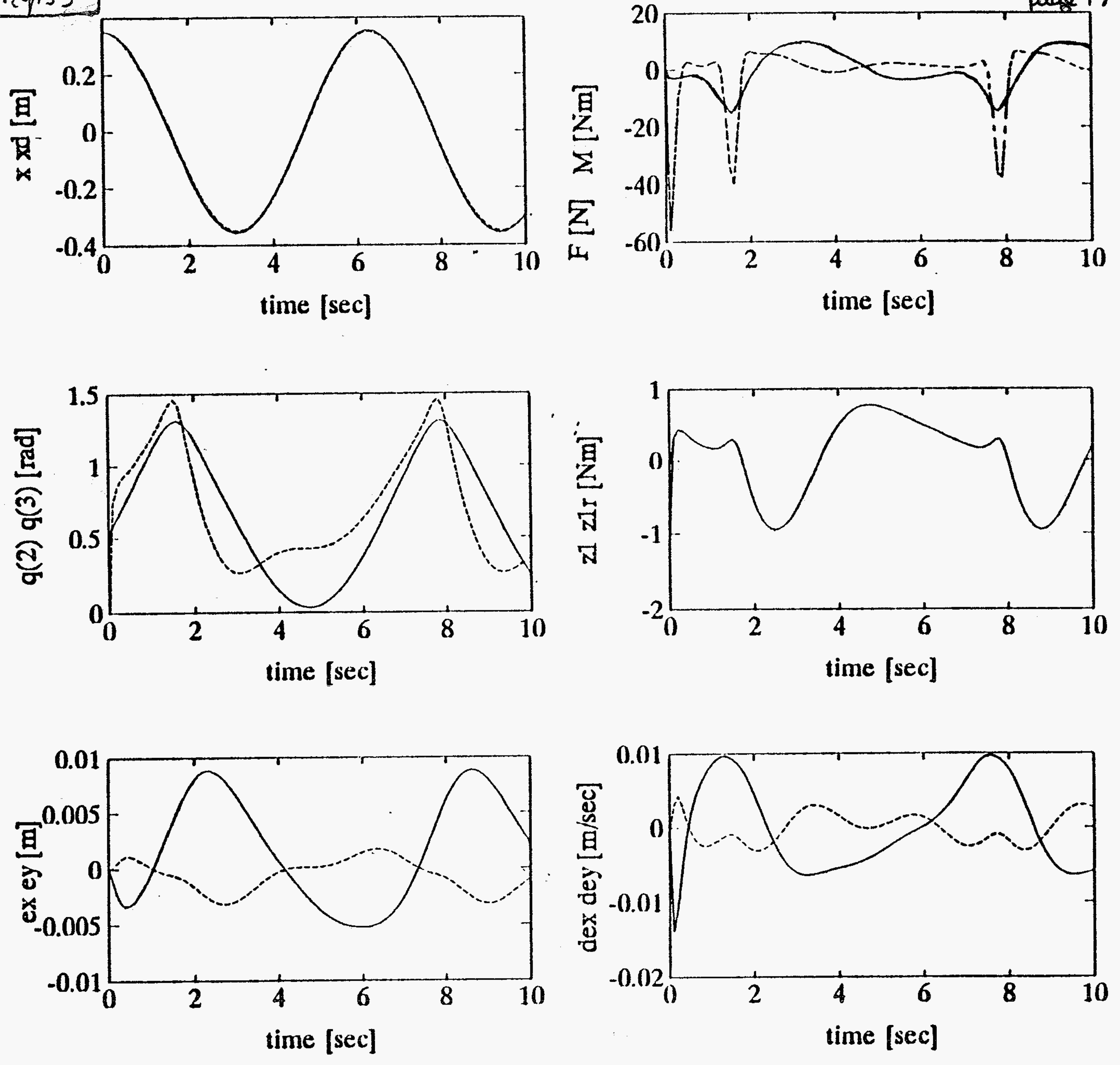


Figure [11]: Flexible CTC (12)–(24), (23): K_s= diag[10], k_z= 10. + fig [16], page [7] + next pages + fig[17] on page 17

$\boxed{M\{q_{r3}\}}$



TUE-WFW-IL
11-1990

Figure [12]: Flexible CTC (12)–(24), (23): $K_s = 0$, $k_z = 10$. $M\{q_{r3}\}$

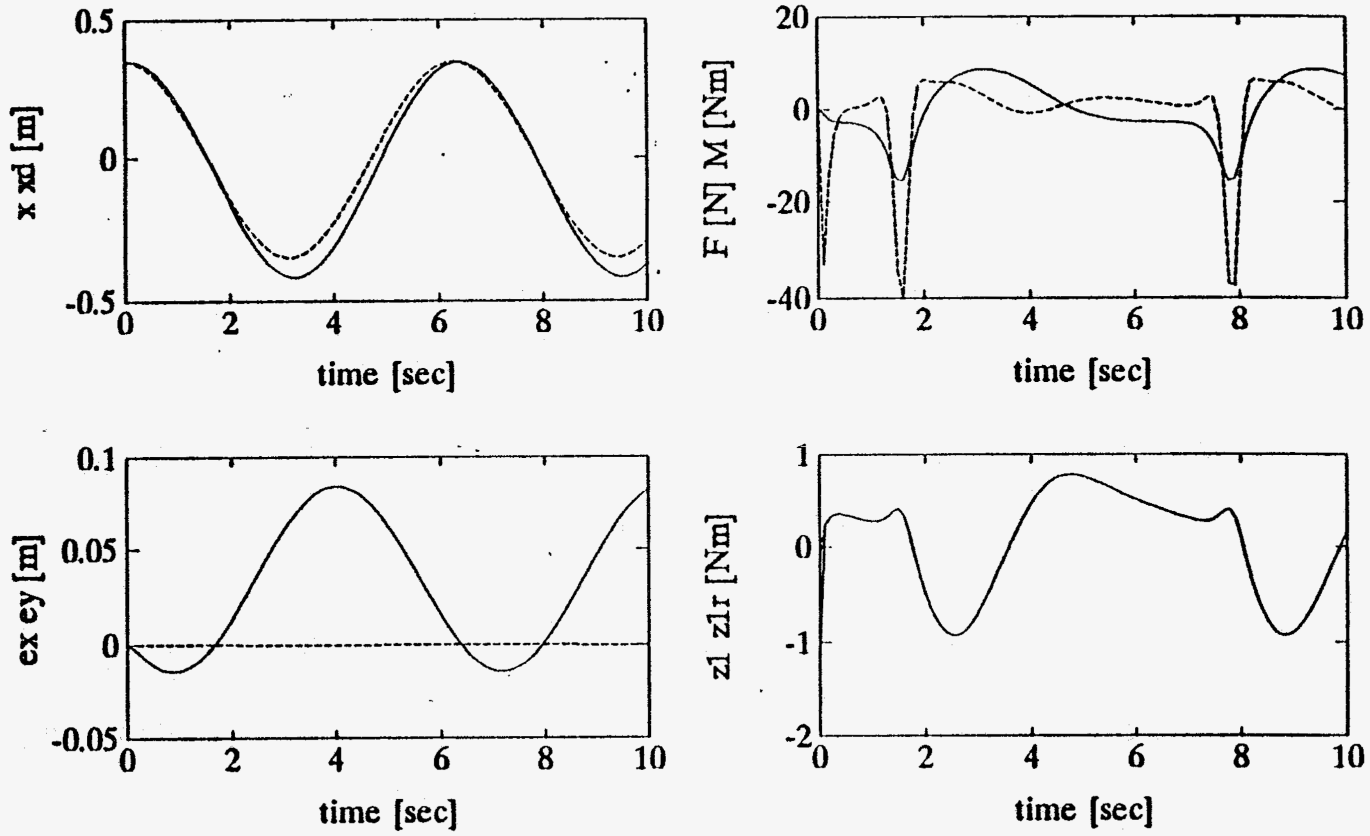
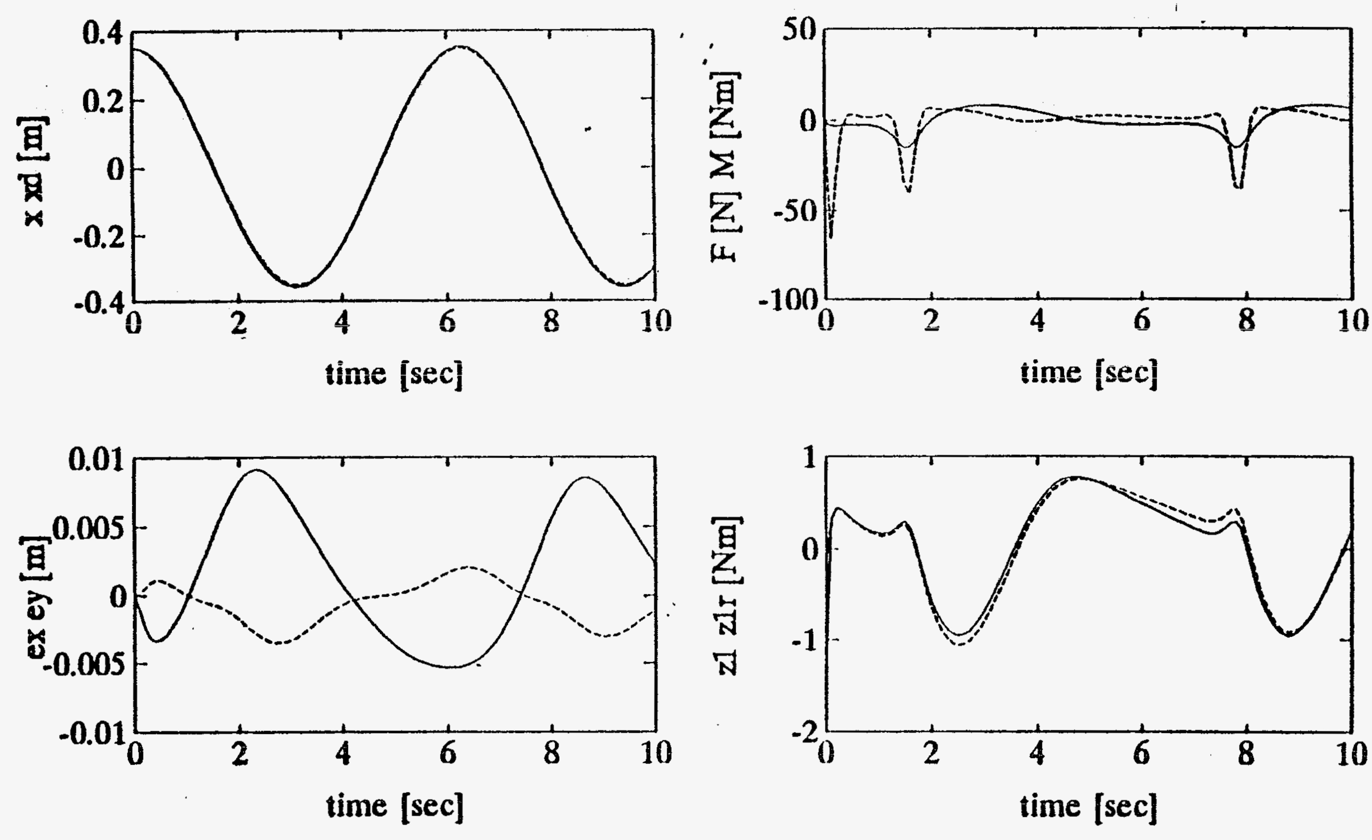


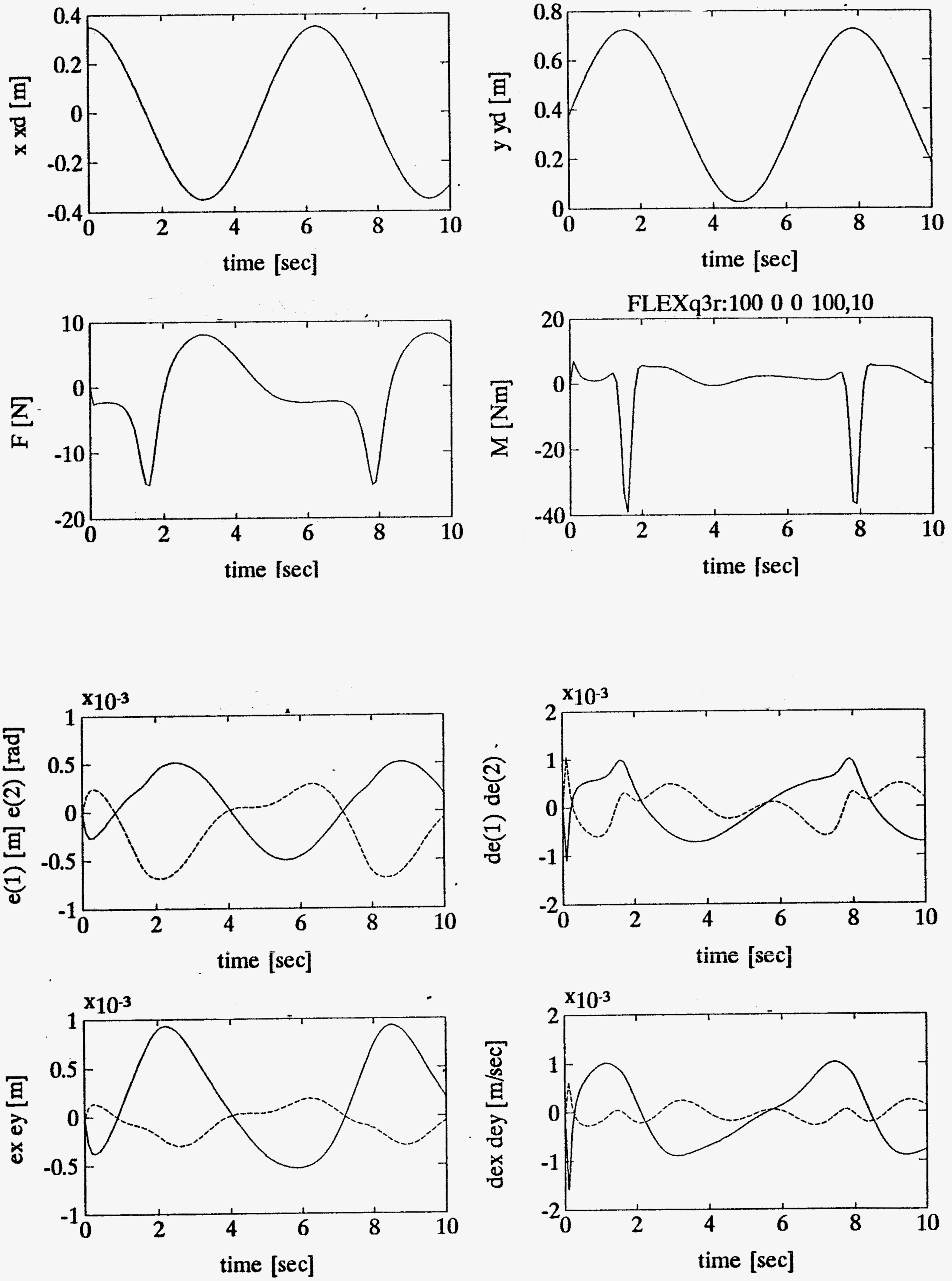
Figure [13]: Flexible CTC (12)–(24), (23): K_s= diag[10], k_z= 0. $M\{q_{r3}\}$



TUE - WFW - IL
11 - 1990

Flexibele CTC (12)(24),(23) : $K_s = \text{diag}\,[100]$, $k_z = 10$.

$M\{q_{r3}\}$



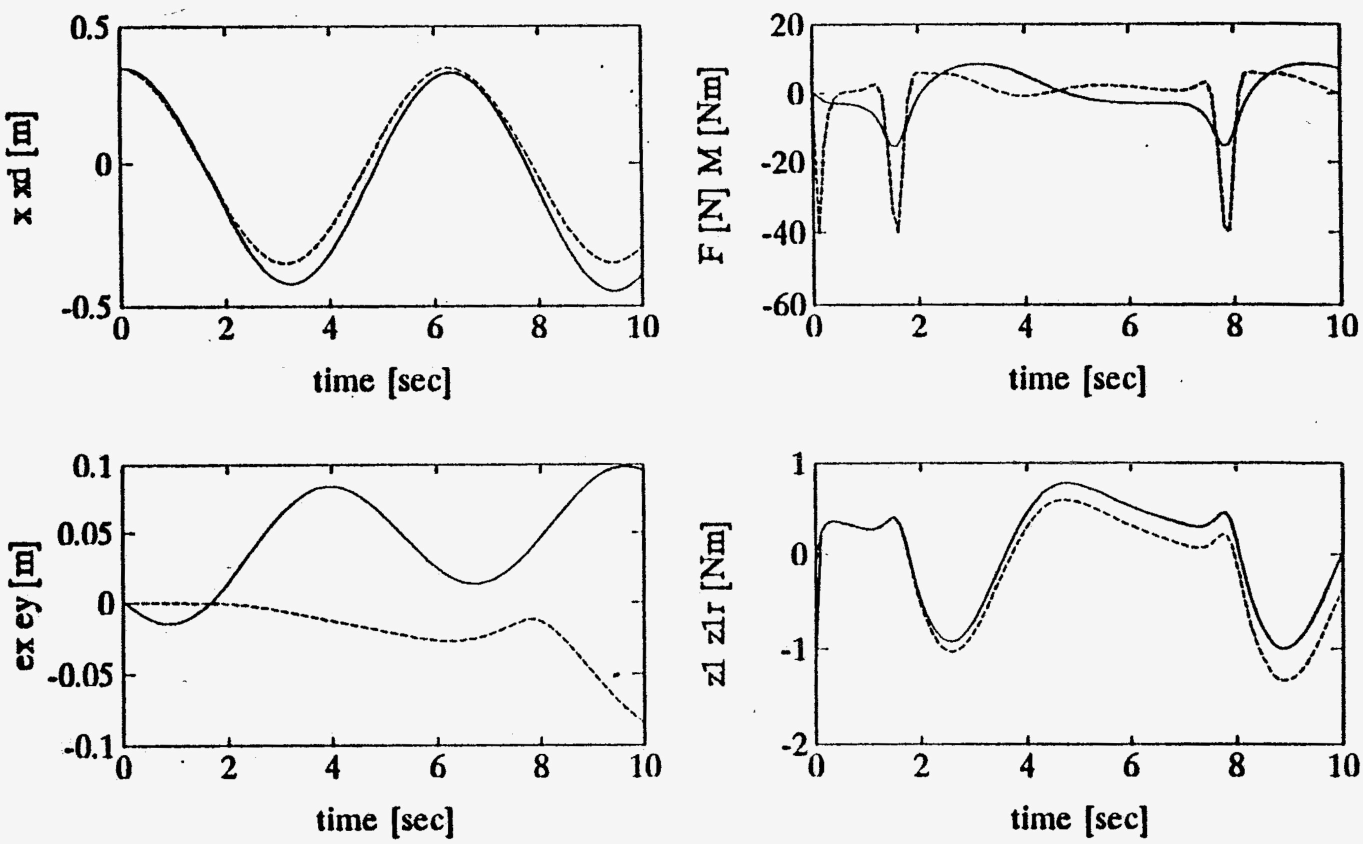
TUE-WFW-IL
11-1990

G- 16

Flexible CTC (12)–(31), q_{r3} (25):

M{q3}

Figure [14]: $K_s = 0$, $k_z = 0$.



M{q3}

Figure [15]: $K_s = \text{diag}[10]$, $k_z = 10$.

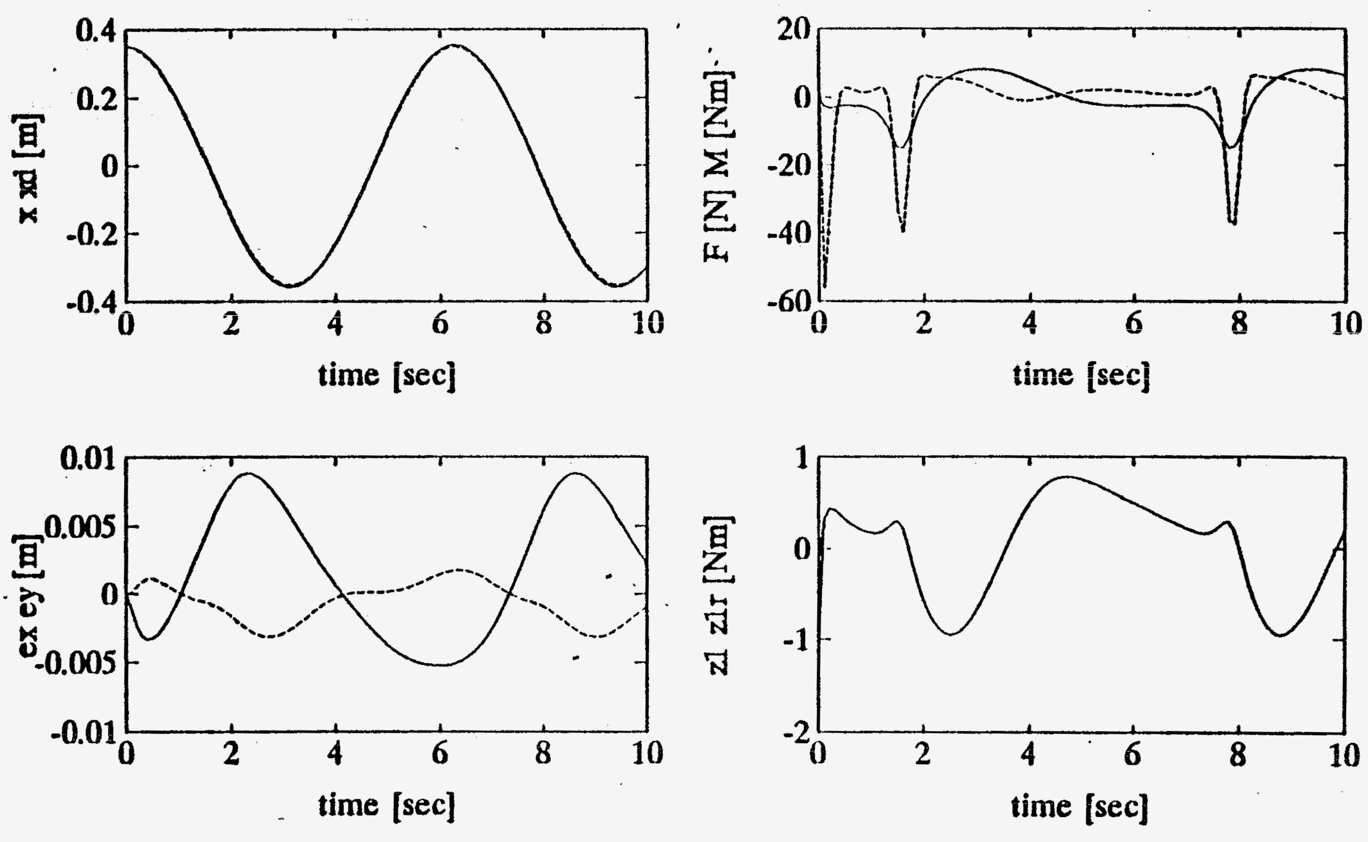
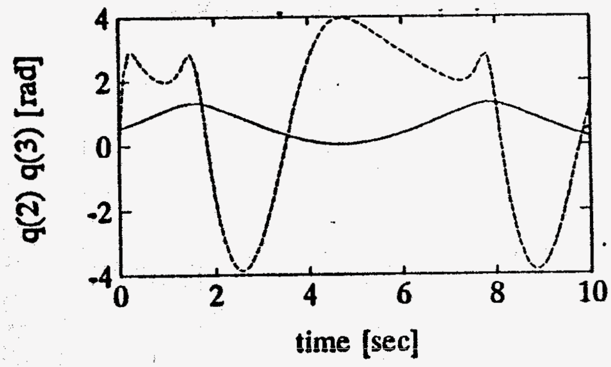
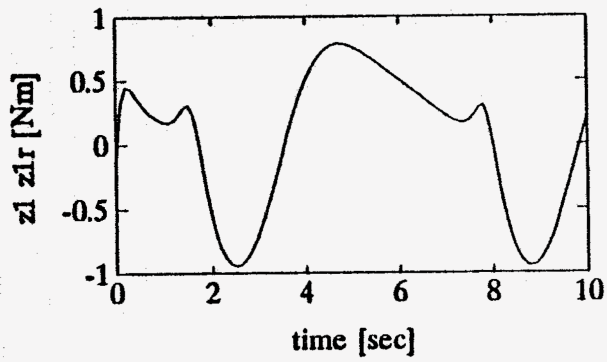


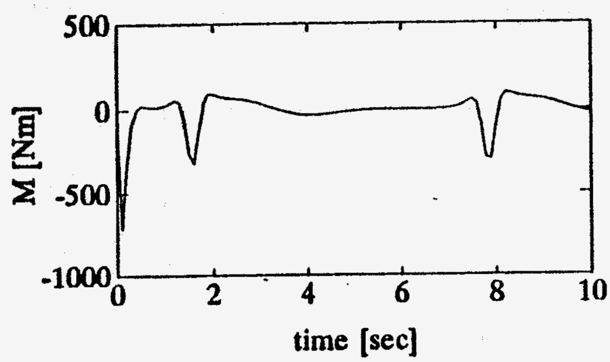
Figure [17]: Flexible CTC (12)(24), (23) : $K_S = diag[10]$, $k_z = 10$.
The elastic-joint stiffness k:

$\boxed{M\{q_{r3}\}}$

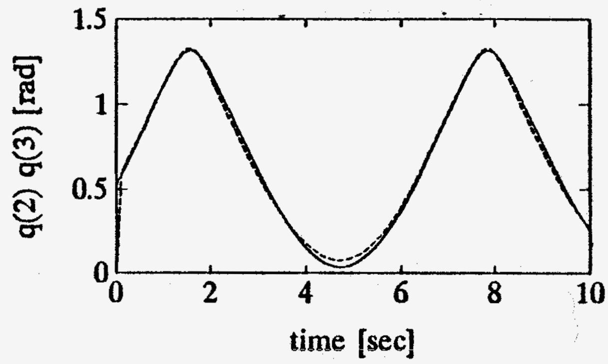
k = 0.2 [Nm/rad]

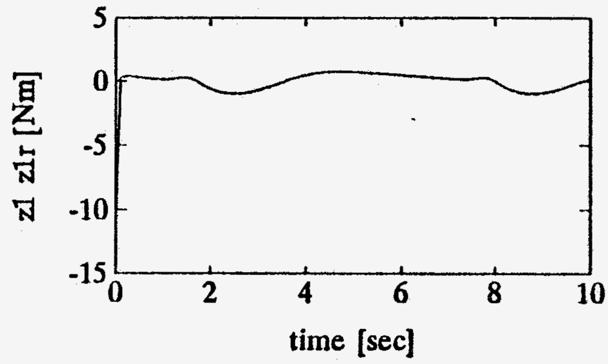


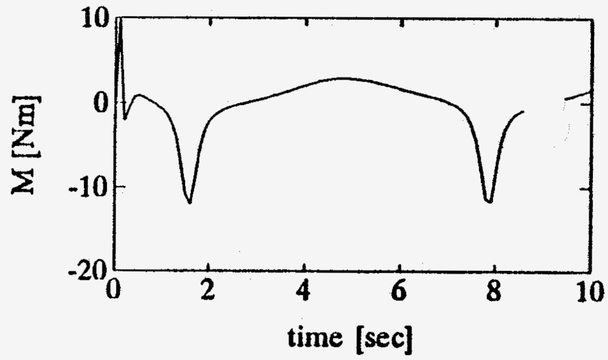




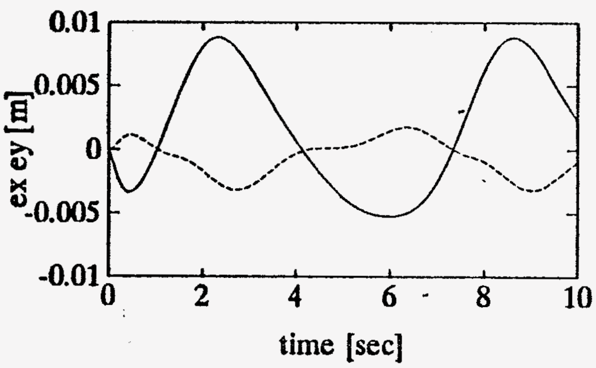
k = 20 [Nm/rad]







k = 0.2 , k = 2, k = 20 [Nm/rad]



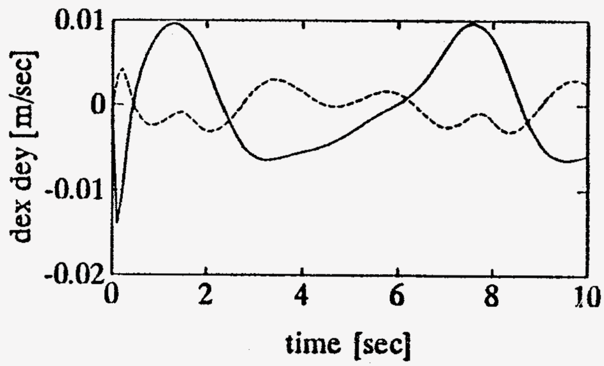
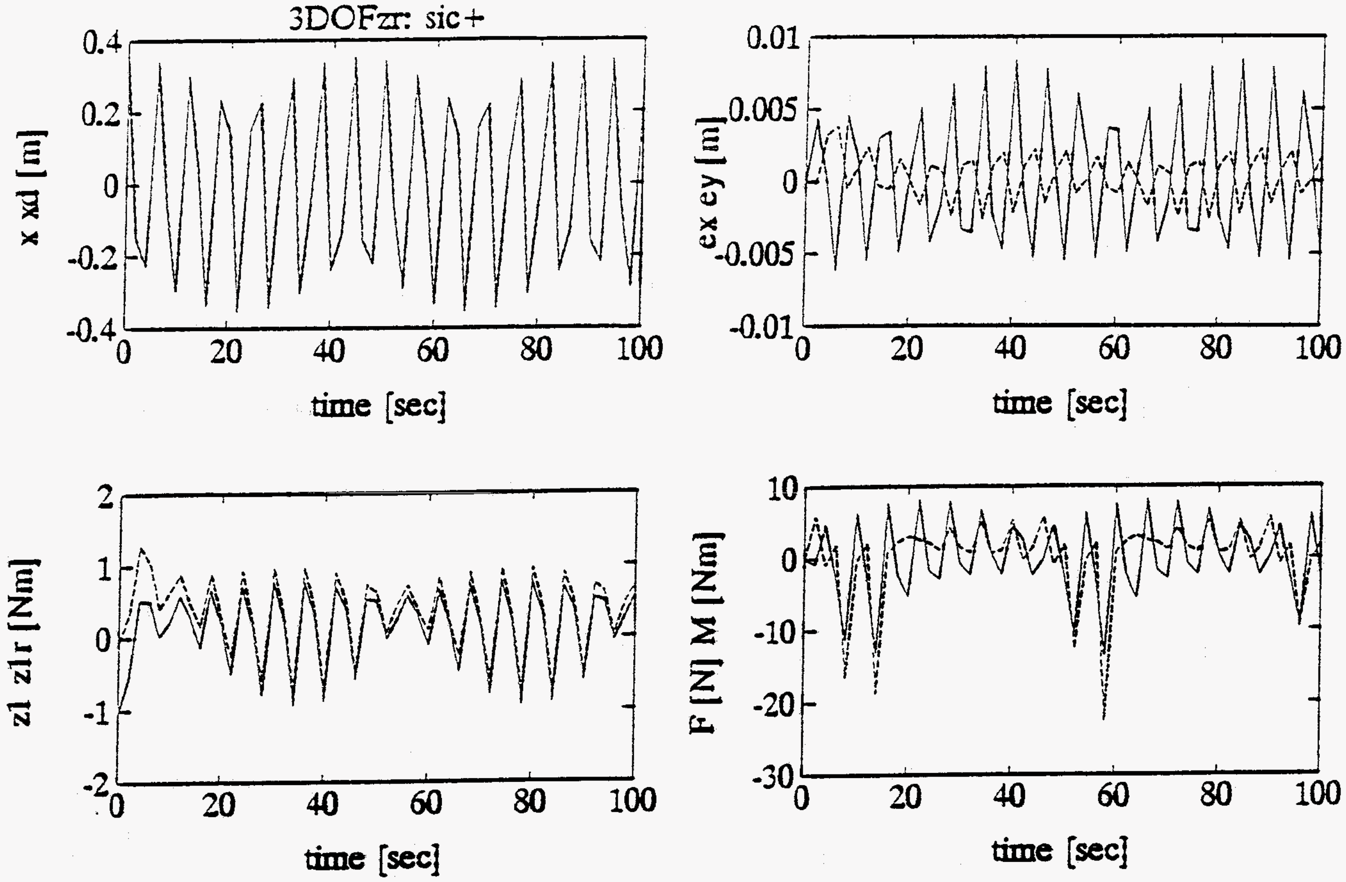


Figure [6] and [8] : $t = 100$ sec $\boxed{M\{zr\}}$: Flexible CTC (19).

(Ideal initial system state conditions)



(The initial system state conditions are all zero)

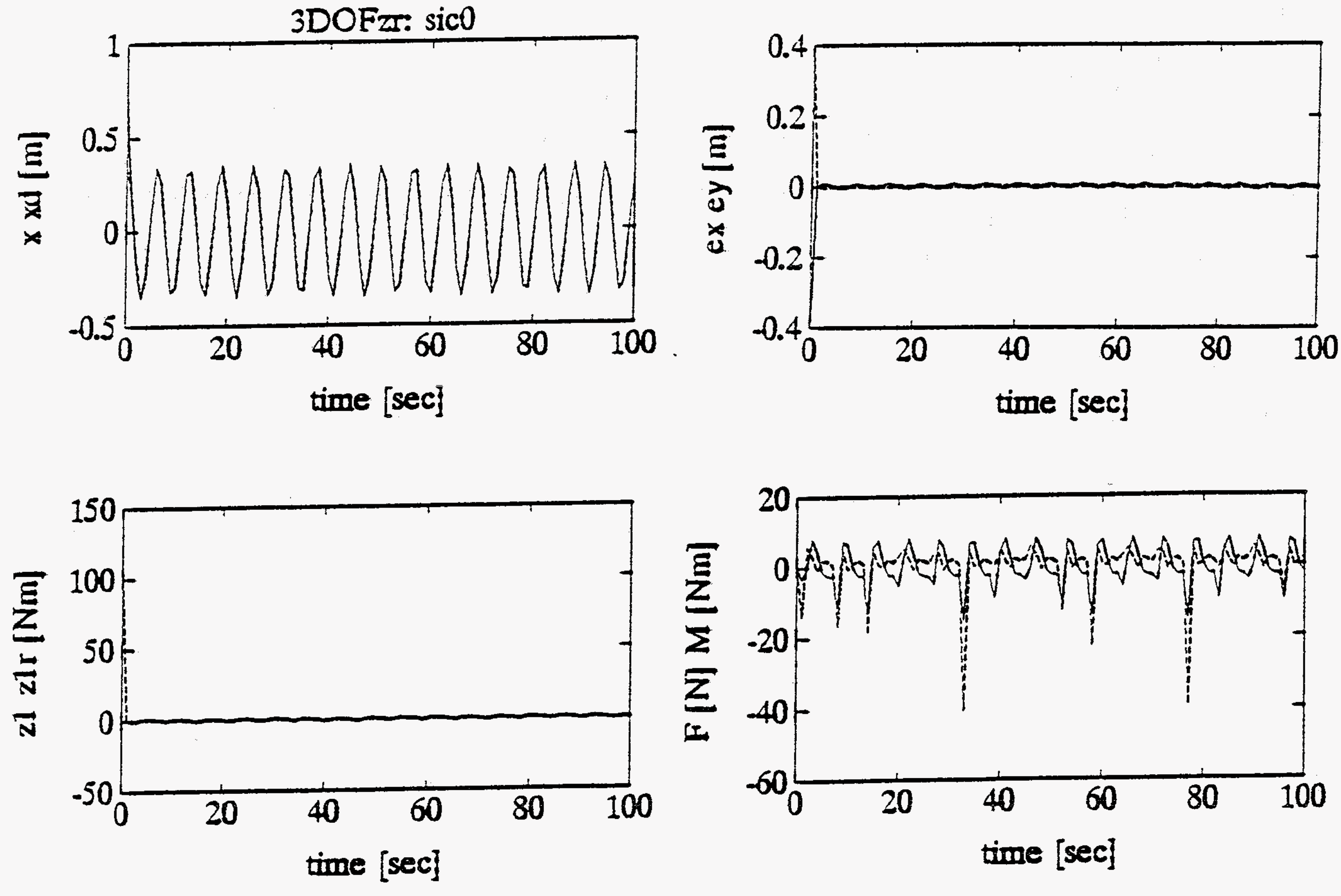
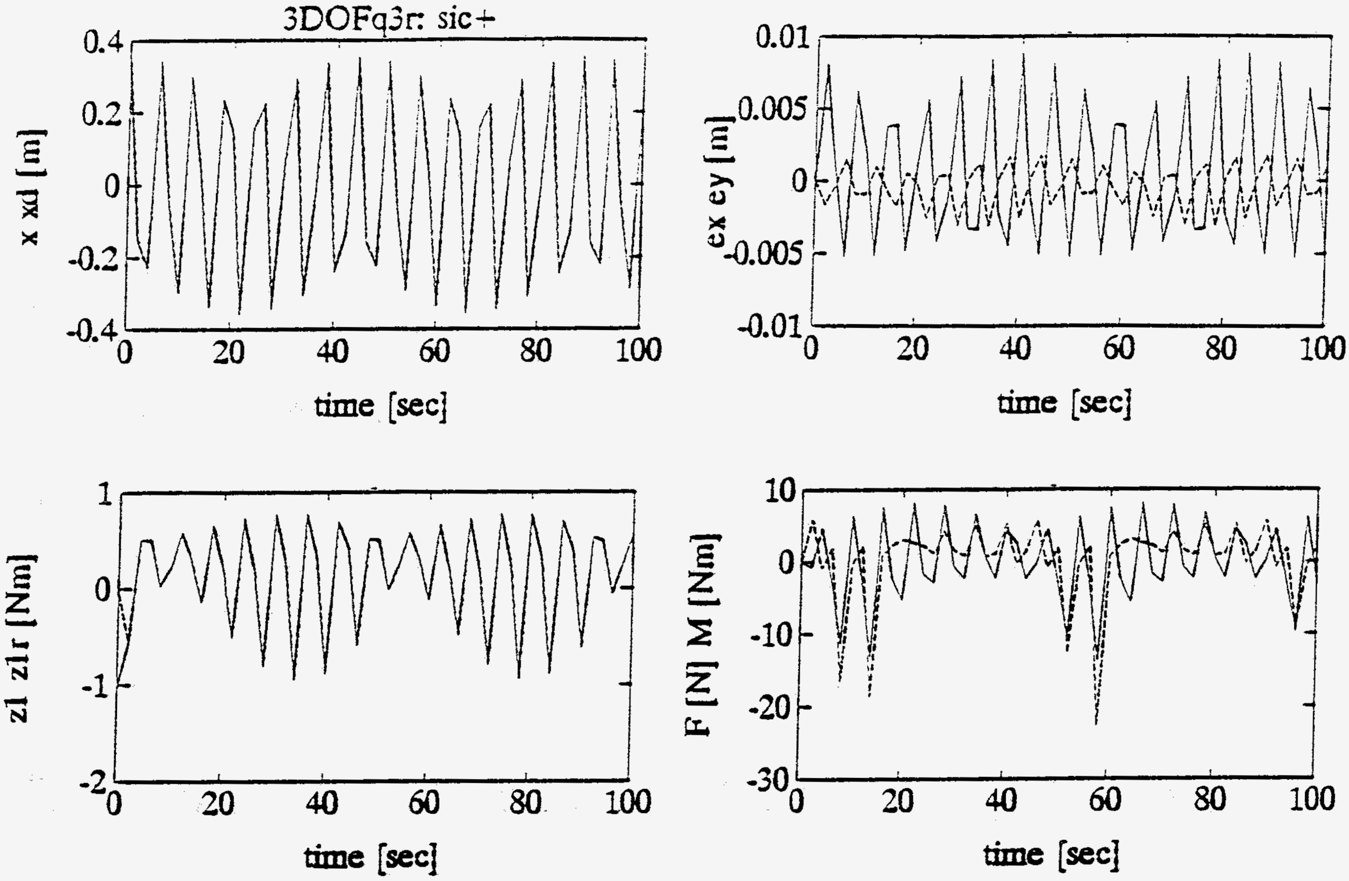
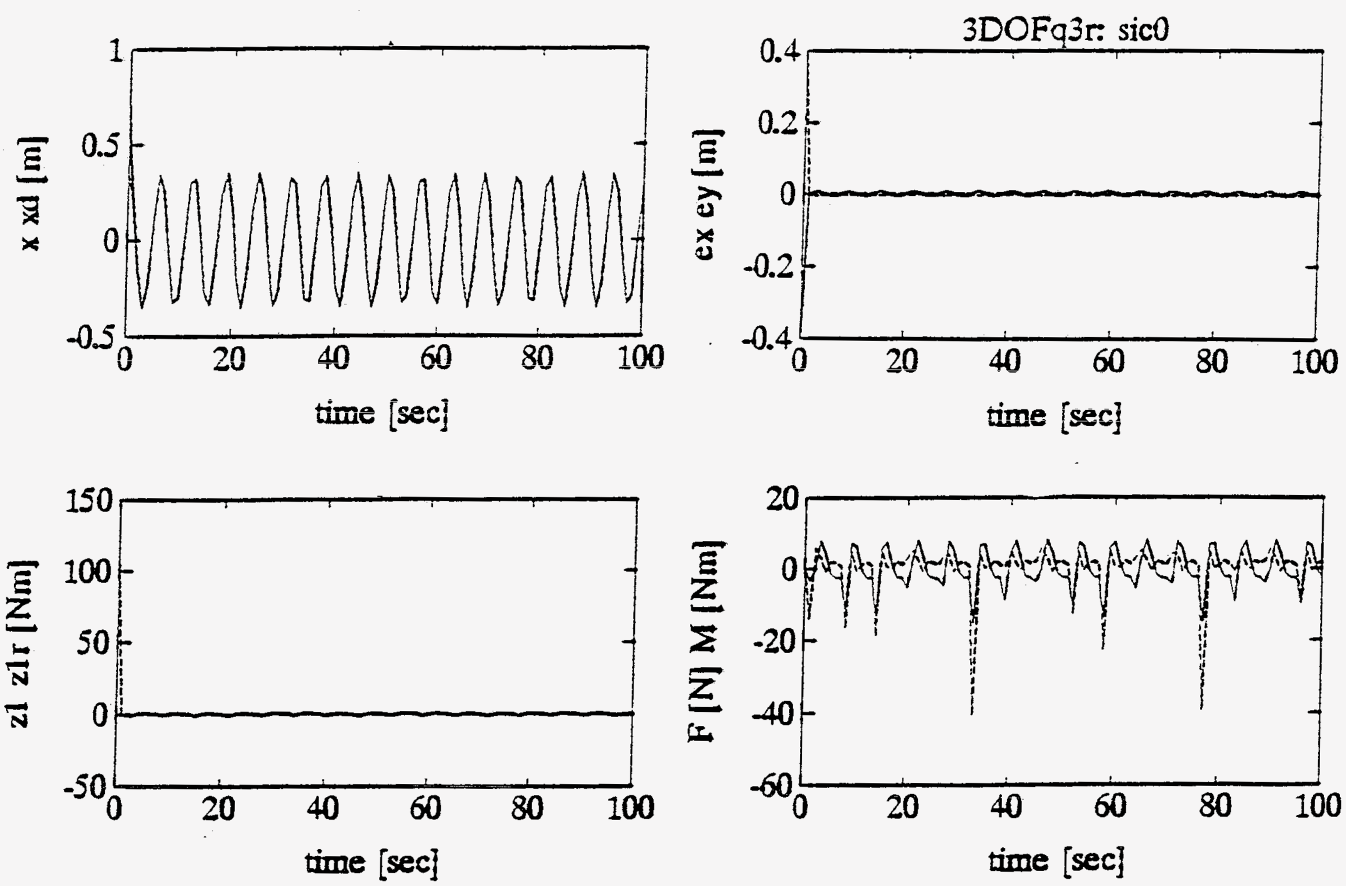


Figure [11] : t = 100 sec $\boxed{M\{q3r\}}$: Flexible CTC(31).

(Ideal initial system state conditions)



(The initial system state conditions are all zero)



H TWO–SUBMODEL BASED COMPUTED TORQUE CONTROL OF ELASTIC–JOINT ROBOTS.

H.1 Introduction.

In contradiction to chapter *F*, in this chapter there is not the restriciton that all motor joints have to be elastic (n=2m): $n \leq 2m$.

H.2 The elastic–joint manipulator system.

The dynamic model of a flexible manipulator is:

F–(1): $$\boxed{\mathbf{M}(\underline{q})\ddot{\underline{q}} + \underline{n}(\underline{q},\dot{\underline{q}},t) = \mathbf{H}\underline{u}}\ , \tag{1}$$

page [*G*–*2*]: The 3–degrees–of–freedom (DOF) translation–totation (TR–) robot:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix},\quad u = \begin{bmatrix} F \\ Mo \end{bmatrix},\quad M = \begin{bmatrix} M_{11} & M_{12} & 0 \\ M_{12} & M_{22} & 0 \\ 0 & 0 & M_{33} \end{bmatrix},\quad H = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},\quad n = \begin{bmatrix} n_1 \\ n_2 - k(q_3 - q_2) \\ n_3 + k(q_3 - q_2) \end{bmatrix}.$$

The equations of motion for a manipulator system with linear elastic joints are:

F–(10): $$\mathbf{M}_r(\underline{q}_r)\ddot{\underline{q}}_r + \underline{n}_r(\underline{q}_r,\dot{\underline{q}}_r) = \underline{z}_f \tag{2}$$

F–(11): $$\mathbf{J}\ddot{\underline{q}}_f + \underline{n}_f + \mathbf{K}(\underline{q}_f - \underline{q}_r) = \underline{u}_z \tag{3}$$

where $$\underline{z}_f = \begin{bmatrix} \underline{u}_x \\ K(\underline{q}_f - \underline{q}_r) \end{bmatrix} = \begin{bmatrix} \underline{u}_x \\ \underline{z} \end{bmatrix}, \tag{4}$$

$\underline{u}_x$ are the rigid–joint actuator forces/torques,
$\underline{z}$ are the elastic forces/torques between the links and motors.

Again, with the definition of

F–(15): $$\underline{x} = \underline{q}_r \tag{5}$$

F–(16): $$\underline{z} = \mathbf{K}(\underline{q}_f - \underline{q}_r) \tag{6}$$

F–(17): $$\mu = \mathbf{K}^{-1} \tag{7}$$

the equations of motion of the flexible system (3)–(4) are changed into:

F–(18): $$\mathbf{M}_x(\underline{x})\ddot{\underline{x}} + \underline{n}_x(\underline{x},\dot{\underline{x}}) = \underline{z}_f \tag{8}$$

F–(19): $$\mu\mathbf{J}_z\ddot{\underline{z}} + \mathbf{J}_z\ddot{\underline{x}} + \underline{n}_z + \underline{z}_f = \underline{u} \tag{9}$$

In the example of the 3–DOF elastic–joint TR–robot:

$$x = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix},\quad z_f = \begin{bmatrix} F \\ k(q_3 - q_2) \end{bmatrix} = \begin{bmatrix} F \\ z \end{bmatrix},\quad n_x = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix},\quad n_z = \begin{bmatrix} 0 \\ n_3 \end{bmatrix},$$

$$u = \begin{bmatrix} F \\ Mo \end{bmatrix},\quad M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{bmatrix},\quad J_z = \begin{bmatrix} 0 & 0 \\ 0 & M_{33} \end{bmatrix}.$$

H.3 Elastic–joint robot control.

The reference forces/torques required for manipulating the rigid links along the desired trajectory:

F–(24): $$\underline{z}_{fr} = \begin{bmatrix} \underline{u}_{xr} \\ \underline{z}_r \end{bmatrix} = M_x(\underline{x})(\ddot{\underline{x}}_r + K_r\dot{\underline{e}}_r) + \underline{n}_x(\underline{x},\dot{\underline{x}}) \quad (10)$$

In the case of the example, the 'rigid' computed torque force is:

G–(12): $F_r = M_{11}(\ddot{q}_{s1}+K_{s11}\dot{e}_{s1})+M_{12}(\ddot{q}_{s2}+K_{s12}\dot{e}_{s2})$,

while the 'reference manifold' is also calculated on–line with:

G–(13): $z_r = M_{12}(\ddot{q}_{s1}+K_{s21}\dot{e}_{s1})+M_{22}(\ddot{q}_{s2}+K_{s22}\dot{e}_{s2})$.

Then, the two–submodel based computed torque control law will be:

F–(33): $$\underline{u} = \underline{u}_r + \mu\underline{u}_f \quad (11)$$

where

*) the rigid computed torque control part:

F–(21): $$\underline{u}_r = J_z(\ddot{\underline{x}}_r + K_r\dot{\underline{e}}_r) + \underline{n}_z + \underline{z}_{fr} = [M_x+J_z](\ddot{\underline{x}}_r+K_r\dot{\underline{e}}_r) + \underline{n}_x + \underline{n}_z \quad (12)$$

*) the flexible computed torque control part:

F–(33): $$\underline{u}_f = J_z(\ddot{\underline{z}}_{fr} + K_z\dot{\underline{e}}_{zf}) \quad (13)$$

The example on page [G–4] shows us that:

G–(20): $$u_r = \begin{bmatrix} F_r \\ z_r+M_{33}(\ddot{q}_{s2}+K_{s22}\dot{e}_{s2})+n_3 \end{bmatrix}, \quad u_f = \mu M_{33}(\ddot{z}_r+K_z\dot{e}_z) .$$

Finally, the equivalent error equations of both subsystems are:

F–(18)/(24), G–(15)/(16): $$\underline{e}_{zf} = \underline{z}_{fr} - \underline{z}_f = M_x(\ddot{\underline{e}}_r + K_r\dot{\underline{e}}_r) \quad (14)$$

F–(19)/(33), G–(21): $$\mu J_z(\ddot{\underline{e}}_{zf}+K_z\dot{\underline{e}}_{zf}) + J_z(\ddot{\underline{e}}_r+K_r\dot{\underline{e}}_r) + \underline{e}_{zf} = \underline{0} \quad (15)$$

The error equations of the total closed–loop system are:

F–(36), G–(22): $$\boxed{[M_x+J_z](\ddot{\underline{e}}_r+K_r\dot{\underline{e}}_r) + \mu J_z(\ddot{\underline{e}}_{zf}+K_z\dot{\underline{e}}_{zf}) = \underline{0}} \quad (16)$$