MODEL PREDICTIVE CONTROL OF A MANIPULATOR ARM WITH FRICTIONAL/UNILATERAL CONTACT

A dissertation submitted
at the
Department of Mechanical Engineering
of the
University of Ottawa
in partial fulfilment of the requirements for
a degree of Ph.D.

by

Jean de Carufel

B.A.Sc., University of Ottawa, 1989
M.A.Sc., University of Ottawa, 1992

May 4, 1998
Ottawa, Canada

0-612-32441-9

To my wife Ann
and my wonderful daughters Malik and Myriam

Abstract

Controlling mechanisms whose equations of motion involve nonlinear discontinuous terms is difficult. A robot manipulator doing a task requiring intermittent contacts with the environment is such a system. Historically, the difficulty was avoided by splitting the global control problem into subproblems defined by the smooth structures of the piecewise discontinuous model. As a result, algorithms for controlling robot manipulators in free motion, transition to contact (impact control) and force/motion in contact were obtained separately and implementation was done using a switching law.

In this thesis, nonlinear Model Predictive Control (MPC) is proposed as a unified solution for controlling robot manipulators with intermittent contacts. The use of a model-based prediction over a receding horizon allows MPC to foresee discontinuous changes in the dynamics and smoothly adjust the control command. Therefore, it was used extensively in the process industry where state and control command saturations are often present. The first contribution in this thesis lies in the use of MPC for controlling systems with discontinuities in the equations of motion. Through analysis and simulation, the ability of the nonlinear MPC approach to provide a unifying solution is demonstrated. The literature on nonlinear MPC being almost inexistent, the work presented herein also contributes to the understanding of how MPC can be applied to nonlinear systems.

Because the complete analytical solution of the nonlinear MPC problem is not prone to real-time applications, two implementation alternatives are also proposed. Both use the operational space information about the task to perform to reduce considerably how much computation is necessary for a solution. The first approach is similar to the resolved-acceleration algorithm with the resolved acceleration being computed from a reduced MPC problem. The second is called the predictive impedance algorithm since its formulation is similar to impedance control with the impedance being replaced by the output of a reduced MPC problem. The applicability of both algorithms has been demonstrated through simulation. Experimental results were also obtained for the predictive impedance solution.

Acknowledgement

Doing this research work under the supervision of Professor Dan S. Necsulescu of the department of Mechanical Engineering of the University of Ottawa, was for me a great fortune. His enthusiasm in research is an important heritage that will follow me throughout my career. Beyond his capability to browse innovative technical ideas, his general knowledge about the history of humankind is unquestionable. I had the pleasure, several times, to be briefed by well-synthesized points of view on various subjects other than technical research. For all this and for his patience, I thank Professor Necsulescu wholeheartedly.

I am also thankful to the Canadian Astronaut Program of the Canadian Space Agency. More than financial support, they provided an interesting practical aspect for the research work I was undertaking. I am particularly grateful to Bjarni Tryggvason, Canadian payload specialist, for his understanding and his highly valuable suggestions.

I will never forget the wonderful moments shared with my colleagues of the Mechanical System Control Laboratory. The discussions we had throughout the years have broadened my cultural horizon. In particular, I would like to thank M. Victor Lonmo for his valuable comments about the writing style presented herein.

Je veux aussi profiter de l'occasion pour remercier mes parents, Jacques et Thérèse, pour l'environnement favorable qu'ils ont su me procurer. Finalement, n'eût été de ma Grand-mère Pauline et de sa vision, je n'aurais possiblement jamais présenté cette dissertation.

Contents

List of Figures

List of Tables

Nomenclature

$\mathbf{A}_d$	State Transition Matrix for Discrete-time LTI systems
$A(q^{-1})$	Polynomial of q^{-1}
$A(d)$	True contact area
$B(q^{-1})$	Polynomial of q^{-1}
$\mathbf{B}_d$	Input Matrix for Discrete-time LTI Systems
$\mathbf{C}$	Output Matrix
C_x	Constraint force in the x direction
$C(q^{-1})$	Polynomial of q^{-1}
d	penetration distance
$D(q^{-1})$	Polynomial of q^{-1}
e	Coefficient of restitution
E'	Equivalent modulus of elasticity
$\mathbf{E}$	Contingent set
e_k	Disturbance at instant k
e_{max}	Maximum coefficient of restitution
e_{min}	Minimum coefficient of restitution
$\mathbf{e}_t(\mathbf{x})$	Tangential unit vector in the direction of motion at point $\bar{\mathbf{x}}$
$f(z)$	pdf of asperity height
$\mathbf{f}$	State equation vector
F	Scalar force
$\mathbf{F}$	Predictor Matrix
$\mathbf{F}()$	Attainability function
f_e	Viscous friction coefficient
$\boldsymbol{f}_e$	Force vector applied on the end effector
$\boldsymbol{f}_{env}$	Environment dynamics
$\boldsymbol{f}_{ext}()$	External force vector
F_f	Friction force scalar
$\mathbf{F}_f$	Friction force vector
$\mathbf{F}_r$	Generalized forces associated with rigid body motion
$\mathbf{F}_e$	Generalized forces associated with elastic body motion
$\mathbf{f}_i$	State equation vector for structure I
F_k	Kinetic friction force
F_{load}	Loading force

F_s	Static friction force
F_{unload}	Unloading contact force
f_v	Coefficient of viscous friction
F_x	Scalar force in the x direction
F_y	Scalar force in the y direction
$\mathbf{G}$	Predictor Matrix
$\mathbf{G(q)}$	Gravity vector (standard in robotics)
$g()$	Steady-state solution of the dynamic model of friction
$\mathbf{g}_{env}()$	Environment Dynamics
$\mathbf{G}_r(\mathbf{q})$	Gravity vector associated with rigid body motion
$\mathbf{G}_e(\mathbf{q})$	Gravity vector associated with elastic body motion
$\mathbf{G}_x(\mathbf{x})$	Gravity vector in operational space (standard in robotics)
$\mathbf{h}$	Output equation vector
$\mathbf{H}$	Predictor Matrix
h_c	Control horizon in sampling steps
$\mathbf{h}_i$	Output equation vector for structure i
h_p	Prediction horizon in sampling steps
$H(q)$	Polynomial of MPC controller
i_m	Motor winding current
I_1	Inertia of link 1 with respect to its centre of mass
I_2	Inertia of link 2 with respect to its centre of mass
J	Inertia of the element
$\mathbf{J}$	Manipulator Jacobian matrix
$\mathbf{Jc}$	Contact Jacobian matrix
J_1	Motor inertia
J_2	Load inertia
K	Shaft stiffness
$\mathbf{K}$	Stiffness Matrix
K_{emf}	Motor back emf constant
K_e	Stiffness of the environment
K_t	Motor torque constant
L	Distance form origin of base frame and the flat wall
L_m	Motor winding inductance
l_1	Length of link 1
l_2	Length of link 2
m	mass

m_1	mass of link 1
m_2	mass of link 2
$\mathbf{M(q)}$	Inertia matrix
$\mathbf{M_{rr}(q)}$	Inertia matrix associated with rigid body motion
$\mathbf{M_{ee}(q)}$	Inertia matrix associated with elastic body motion
$\mathbf{M_{er}(q)}$	Inertial coupling between rigid and elastic body motion
$\mathbf{M_{re}(q)}$	Inertial coupling between rigid and elastic body motion
$\mathbf{M}_x(\mathbf{x})$	Inertia matrix in operational space
n	Order of the system
$\mathbf{n}(\bar{\mathbf{x}})$	Normal unit vector at $\bar{\mathbf{x}}$
N	Normal contact force
N_a	Normal contact force
$\mathbf{N(q)}$	Coriolis and centrifugal term vector (robotic standard)
$\mathbf{N}_r(\mathbf{q})$	Coriolis and centrifugal term vector associated with rigid body motion
$\mathbf{N}_e(\mathbf{q})$	Coriolis and centrifugal term vector associated with elastic body motion
$\mathbf{N}_x(\mathbf{x})$	Coriolis and centrifugal term vector in operational space (robotic standard)
$\mathbf{p}$	Direction of MPC solution in operational space
$P(d)$	Contact force
q	Shift operator
$\mathbf{q}$	Manipulator coordinate vector
$\mathbf{q}_e$	Elastic coordinates of manipulator
$\mathbf{q}_r$	Rigid body coordinates of manipulator
$\mathbf{Q}_1$	Error weighting matrix
$\mathbf{Q}_2$	Output weighting matrix
$\mathbf{r}$	Reference trajectory
r_i	Reference trajectory for output i
r_k	Reference trajectory at instant k
r_1	Distance from preceding joint to centre of mass of link 1
r_2	Distance from preceding joint to centre of mass of link 2
R_m	Motor winding resistance
$\mathbf{s}$	Direction of MPC solution in joint space
S_b	Boundary surface of end effector
S_{env}	Boundary surface of the environment
s_i	i^{th} factor of the B(q) polynomial
t	Time
T_h	Prediction horizon time

T_t	Transition matrix
t_{load}	Contact loading duration
T_u	Control horizon time
t_{unl}	Contact unloading duration
$T(q^{-1})$	Polynomial of MPC controller
$\mathbf{u}$	Input vector
u_k	Scalar input at instant k
$\mathbf{u}_{min}$	Minimum value of vector **u**
$\mathbf{u}_{max}$	Maximum value of vector **u**
$\mathbf{v}$	Measurement noise vector
$V()$	Lyapunov function
v_c	Empirical scalar for impact model
v_{imp}	Normal speed of impact
$V(q^{-1})$	Polynomial of MPC controller
V_m	Motor voltage
v_{rel}	Relative speed
v_t	Tangential speed
w	generalized penetration distance
$\mathbf{w}$	State vector
W	Set of all topological space of a system
W_i	Topological space of the structure i of a dynamic system
w_c	Critical interference
x	Position in x direction
$\mathbf{x}$	Cartesian coordinate of robot manipulator
$\mathbf{x}^*$	Location of environment surface with respect to environment frame
X	State space for a dynamic system
$\mathbf{x}_b$	Location of end effector boundary points with respect to end effector frame
$\mathbf{x}_e$	Operational space vector of the end effector
$\mathbf{x}_{env}$	Location of environment reference frame with respect to inertial frame
${}^p\mathbf{x}_e$	3D position vector of the end effector
$\mathbf{x}_f$	Family state vector for a dynamic system
$\mathbf{x}_i$	State vector for structure i of a dynamic system
$\mathbf{x}_k$	State vector at instant k
$\bar{\mathbf{x}}$	3D position vector for a point on the contact surface
X_w	Manipulator work volume
y	Position in y direction

$\mathbf{y}$	Output vector
$\mathbf{y}_d$	Desired output vector
y_i	Output i
y_k	Output at instant k
YD	set of admissible references
z	Friction state vector or generalized coordinate
Z	Set of distances between environment and end effector
α	Error weighting scalar
β	Average asperity tip radius or empirical factor in impact model
γ	Output weighting scalar
γ_x	Weight on position error direction
γ_v	Weight on velocity error direction
γ_f	Weight on contact force error direction
Γ	Forward kinematics
δ	Empirical parameter for Friction model or small number
Δ	$(1-q^{-1})$: the differentiator
ϵ	small arbitrary number or empirical coefficient
$\varphi(x)$	flexible dynamics mode shape
λ	Lagrange multipliers
λ_{mpc}	Predictive impedance variable
η	Scalar constant (theorem 4.4)
η_1	flexible dynamics coordinate
θ_1	Joint angle for joint 1
θ_2	Joint angle for joint 2
κ	Scalar constant
κ_k	Kinetic friction coefficient with respect to true contact area
κ_s	Static friction coefficient with respect to true contact area
μ_k	Coefficient of kinetic friction
μ_s	Coefficient of static friction
ν	Integer scalar constant or Poisson's ratio
ω_m	Motor angular speed
σ_0	Contact stiffness in friction model
σ_1	Contact damping in friction model
σ_{yield}	Yield strength
τ	Torque
$\boldsymbol{\tau}$	Vector of applied joint torques

τ_c	Control torques
τ_{eq}	Equivalent torque
τ_f	friction torque
τ_f	Vector of joint friction torques
τ_k	Kinetic friction torque
τ_s	Static friction torque
τ_{motor}	Motor torque
ϕ_t	Motor magnetic flux
$\phi_e()$	Descriptor function for the end effector surface
$\phi_{env}()$	Descriptor function for the environment surface
Φ	Cost functional
$\hat{}$	Estimate

Chapter I

Introduction

1.1 - Scope

Since the World entered the era of technology, yesterday's dreams have become today's reality. Beyond socioeconomic concerns, the human desire to control its destiny and its environment have prevailed, and technology has fulfilled the need for the human self-determination. The increasing demand for technical expertise is a challenge for us, and will continue to be for the generations that will follow.

The technology of automation is a good example of a technical field where the new requirements have created a stringent need for innovation. It has reached the level where automatic systems will not only behave correctly with respect to their own states, but will also interact intelligently with their physical environment. The level of behavioural complexity associated with these new objectives is extreme and human-like machines are expected to result from this quest. In that respect, new series of sophisticated sensors and actuators have been and are being developed, all trying in their own way to replicate one of the human senses and physical abilities. Tactile sensors, vision systems, force sensors and smart structures are becoming an integral part of the so-called *intelligent* automation.

Although the new technologies are necessary for achieving the desired level of automation, they are not sufficient. Beyond their improved precision and efficiency, these new devices have to be integrated into complete and comprehensive control schemes to be really useful. A good example is the control of a manipulator arm performing tasks such as drilling, grinding, sanding and cleaning a window, all requiring an intermittent contact with the environment. Contact force, proximity to contacting bodies, position, velocity and acceleration of both the robot and the environment, all these variables can be measured using the appropriate sensing device. The algorithms to interpret the measurements and generate the proper control commands currently limits our ability to control the transition to contact, the force and the motion in a unified framework. The various solutions proposed so far have been obtained for specific conditions of the overall problem. As a result, global solutions are realized using a *switching law* to evaluate the particular condition of the manipulator and to decide which control strategy to apply at a given instant. This switching characteristic of the algorithm requires considerable caution in practical situations. The discontinuity it creates is often severe, and a great deal of stability robustness is necessary to deal with the effect of any model mismatch near the switching conditions.

Since our physical ability allows us (the human) to deal easily with such situations, starting the investigation with the human psychomotor system is natural. Obviously, the approach selected until now to solve the robot manipulator problem is not based on a model of the human psychomotor system. In control system engineering, problems are essentially considered from the point of view of power transfer for which Lyapunov types of analysis are used. However, humans do not visualize the motion of their body parts through power transfer. We use essentially three important components of our psychomotor system. First, we have very sophisticated *built-in* sensing devices allowing us to sense our motion, the motion of the environment and some interaction with the environment (e.g. heat and force). Second, each part of our body is actuated by a series of muscles over which the control depends on the physical abilities specific to each individual. Finally, based on abstract knowledge of our motion and of the expected behaviour of the environment, we use our reflex system for a rapid but inaccurate action, or we use our brain power to predict the future interaction and control the muscles to obtain a desired interaction. These aspects of knowledge and prediction are the key issues of the human's *control approach.*

In this thesis, a control method based on the notions of knowledge and prediction is proposed to control a robot manipulator with unilateral/frictional contacts. The method is known as Model Predictive Control (MPC) and provides a unified solution to the problems of free motion, transition to contact and contact motion/force control of a robot manipulator. In MPC, the knowledge is represented by analytical models of the robot manipulator and the environment. These models are used to predict from the measurements, the motion over a receding horizon extending from the current time (or a delayed instant later for systems with pure delays) over a fixed interval in the future. The actual control command is decided upon some control objectives over the receding horizon. Usually, the control objectives are given in terms of the minimization of some quadratic criteria defined over the receding horizon. To identify the novelty of the approach clearly, the next section of Chapter one is dedicated to a review of the literature dealing with robot manipulator control for impact/contact motion.

1.2 - Literature review

As reported earlier, the problems of controlling free motion, transition to contact (impact control) and position/force while contact, were studied separately. Many books were published reporting the control of robot manipulators in free motion (eg. Lewis et al.[1] and Paul[2]). Impact control was studied first by Youcef-Toumi and Gutz[3,4] following the modelling effort of Zheng and Hemmani[5] and Wang and Mason[6]. From these initial studies of impact control, few results were obtained. Mills[7] and Lokhorst and Mills[8], inspired by Gutman[9], used the notion of Generalized Dynamical Systems (GDS) of Roxin [10,11] to derive the stability conditions for a switching controller during the transition to or from a contact task. Walker[12] studied the effect of robot configurations on the impact properties of robots. He showed how kinematic redundancy can be used to reduce the importance of impact during a transition.

A historical perspective of the methods for position/force control in contact motion was presented by Whitney[13], and a more recent survey was reported by Vukobratović and Tuneski[14]. One can mainly distinguish two families of methods: the passive and the active compliance methods. Passive compliance methods in which the inherent stiffness of the manipulator is used are more academic than practical (see [14]). In reality, industrial robots are made highly rigid, limiting the practicality of using their flexibility for the control of contact force. To compensate for this lack of flexibility, Whitney and Nevins[15] have introduced the idea of Remote Compliance Centre (RCC). RCCs are flexible devices added to the end effector of the manipulator, providing a known contact impedance to be controlled. However, RCCs have to be adapted to the particular task performed, and any task modification may require a change of RCC devices.

An active compliance method, the hybrid position/force controller, was introduced by Raibert and Craig[16] and consists of separating the operational space into orthogonal directions in which the position and force controllers can be designed independently. A compliance selection matrix is used to define the orthogonal directions. In the work of Raibert and Craig, the type of position and force controller is generalized such that any control law can be used. Shin and Lee[17] and Yoshikawa[18] have used the hybrid position/force control schemes. They used the computed torque approach (Paul[2]), decoupling the orthogonal subspace for position and force control. Wen and Murphy[19] investigated the stability issues when the operational space control approach of Khatib[20] is used. In their analysis, Wen and Murphy showed that, unless the environment is infinitely rigid, the force and position loop dynamics are coupled through the effective arm tip mass matrix.

Another active compliance method is the reduced order dynamics control, introduced by McClamroch[21] and later extended by Wang and McClamroch[22], Krishnan and McClamroch[23], Mills and Goldenberg[24] and by Tahboud et al.[25,26]. McClamroch showed the similitude between singularly perturbed systems and dynamic systems defined by differential-algebraic equations. From this similitude, the methods for control system design derived from the theory of singular perturbations, as presented by Kokotovic et al.[27], can be used to design control laws for the constrained model defined by a set of differential (treated as the slow dynamics) and algebraic equations (treated as the fast dynamics). For robot manipulators, this is achieved by modelling the rigid contact as a holonomic constraint to the robot dynamics and reducing the dynamics(see appendix A). The resulting expression for the constraint force represents the algebraic component of the dynamics, considered as the fast dynamics part of the singularly perturbed system. This approach, in its final form, is very similar to the hybrid position/force control approach. In fact, the order reduction of the dynamic model implies the determination of an orthogonal complement (not necessarily in operational space) that defines a subspace in which the motion is defined by the reduced order dynamics. Their similarity is such that they share the same drawbacks.

The impedance control approach, first introduced in the control system community by Hogan[28], can also be considered as an independent approach, although it is apparently the same as the Khatib's operational space hybrid position/force control. In reality, the impedance control approach is much more general than the way it is used by Khatib. As discussed in Lewis et al.[1], the duality principle can be used to define the family of impedances that would control position and force for different types of environments. In hybrid impedance control, the idea is to modulate the control command such that a specified impedance is achieved between the position/force trajectory and the desired position/force trajectory.

Stiffness control, introduced by Salisbury[29], is in fact a precursor to the more general impedance control. In this method, the Cartesian stiffness of the robot is modulated such that some sets point in position and force can be achieved by a unique position controller. A position set point inside the environment leads to a contact force proportional to its depth in the environment. As pointed out by Lewis, a major disadvantage of stiffness control is that it can be used for a set point only.

Another interesting approach, closely related to the impedance control approach, is the virtual environment approach of Fraisse et al.[30]. Based on the reasoning that the major problem in contact motion control be the unknown contact stiffness, they suggest the use of a highly rigid virtual stiffness in parallel with the contact stiffness. In these circumstances, the virtual stiffness becomes dominant and the apparent contact stiffness is well defined.

1.3 - Thesis Preview

As shown in the preceding section, the control of robot manipulators performing contact tasks has been well investigated. Nevertheless, the state of the art has not reached a level where all the aspects of the overall robot control problem can be solved using one unified scheme. Different controllers, defined for special conditions of the robot state, are put together to span all possible conditions of the robot configuration. ***This thesis is concerned with the development of a control method that solves the overall control problem in a unified framework.*** The novelty in this thesis lies in the use of MPC to deal with the control of a system whose model exhibits variable structures (and topology).

Since MPC is a model-based control method, the modelling stage of the control design process is extremely important. In Chapter 2, an extensive discussion about the modelling aspect of the different components of the overall physical system is presented. This presentation includes the study of manipulator dynamics (section 2.1), impact/contact models (section 2.2) and environment models (section 2.3). The emphasis is put on the assumptions that can be made and their range of validity. An overall model is presented in section 2.4.

In Chapter 3, the formulation of the control problem studied is covered. In section 3.1, a statement of the general problem is formulated. The particularities of the problem are addressed in section 3.2 and the methods proposed in the past are reviewed in section 3.3.

In Chapter 4, Model Predictive Control (MPC) is introduced. In section 4.1, the general continuous-time formulation is shown to give a concise picture of MPC. The application of MPC to linear time invariant systems has been well developed and a unified presentation is given in section 4.2. Section 4.3 concentrates on developing MPC for nonlinear systems. Four approaches to solve the nonlinear MPC problem are suggested and revised.

The application of nonlinear MPC to solve the formulated problem is described in Chapter 5. It is shown how the properties of MPC can solve the overall control problem in one common framework. It is explained how this novel approach avoids switching control law and the stability problems associated with it.

Finally, Chapter 6 describes experimental and simulation studies done to illustrate the applicability of the method, using a planar 2 DOF robot contacting a flat wall. The demonstration is simple but general enough to prove the applicability of the method for more complex robot manipulators, environments and tasks.

Chapter II

Modelling

For model-based control methods, modelling is a critical stage in the design process. The validity of the modelling assumptions often determines the success of the method. It is no surprise that most drawbacks of the existing control methods for impact/contact motion control are related to the specific assumptions underlying the various models used. It is therefore the ultimate goal of this chapter to clarify the models and their assumptions. The models reviewed include the ones for the robot manipulator, the joint friction, the contact and impact dynamics, the surface friction and the environment. In the last section of the chapter, an overall model is synthesized by joining the different specific models.

2.1 - Manipulator Arm Dynamics

2.1.1 - Actuators

An impressive variety of actuating devices for robot manipulators are available, but only electromechanical devices are considered herein. For robot manipulator motion control, stepper motors or dc motors are mainly used. For force/position control, joint torques need to be explicitly controlled and dc motors are usually employed.

For armature controlled dc motors, the torque τ is given by

$$\tau = K_t \phi_m i_m \tag{2.1}$$

where K_t is the proportionality constant, ϕ_m is the magnetic flux and i_m is the motor current. The current in the motor winding results from applying the supply voltage V_m and the back emf voltage to the winding's LR circuit,

$$L_m \frac{di_m}{dt} + R_m i_m = V_m - K_{emf}\ \omega_m \tag{2.2}$$

where L_m is the winding inductance, R_m is the motor resistance, K_{emf} is the back emf constant and ω_m is the motor angular speed. When a dc motor is voltage controlled, the torque dynamic is modelled by a first order system with a time constant L_m/R_m and gain $K_t\phi_m/R_m$, disturbed by the angular velocity.

In force control, an explicit control over joint torque is desirable. This is usually achieved by designing a power amplifier that contains a current feedback controller (current loop). With such amplifiers (most of which are Pulse Width Modulated PWM), the bandwidth is increased and the applied voltage maps a desired winding current.

In many situations, transmissions are added to dc motors to improve the torque to weight ratios. The type of gear used for force control has to provide the necessary back-driveability by avoiding backlash at the joint level. The addition of gears generally increases joint friction and sometimes adds flexibility at the joint. The reader who wishes to get more information about joint flexibility models can find good discussions in [31,32,33,34]. Joint friction is discussed in section 2.1.4.

To avoid the problems associated with gears, different types of motors were developed. Of these, the direct-drive brushless dc motors are particularly attractive because the winding is mounted on the stator and no brushes are needed for commutation. This configuration allows the design of motors with more phases and with optimized winding geometry. Consequently, the motor has a higher torque to weight ratio and can be used without gears. Some drawbacks of brushless dc motors include the need for sophisticated commutation electronics, the need of precise build-in position sensors and the torque ripples. The commutation electronics is nowadays extremely fast such that the commutation bandwidth is no longer a limit. In fact, the current loop for each motor phase can be implemented directly in the commutation electronics to provide a current controlled motor with known dynamics characteristics. Moreover, the built-in position sensor prevents the addition of an extra sensor at the joint since it can directly provide position signals for control purposes. Finally, the torque ripples are easily identifiable and can be compensated by static nonlinear feedback.

2.1.2 - Kinematics

The motion of a robot arm can be described in many different ways. For example, the formalism of rigid multi body dynamics, presented in the appendix A, can be used in the case of rigid manipulators. Unfortunately, this approach becomes rapidly impractical, especially if the evaluation of the constraint reaction forces is not needed. For rigid link manipulators, each link represents a rigid body constrained by a kinematic constraint of order five (5 DOF of relative motion removed). To represent each increment in DOF with each link, the Danevit-Hartenberg(DH) convention is usually followed (see Lewis et al.[1] or Paul[2]). Using this convention, each acquired DOF is represented by a relative coordinate defining the configuration of the link with respect with the preceding one. The method simply consists of recursively defining body frames attached to the centre of each joint, and to represent the relation between them using homogeneous transformations. The approach provides a simplified systematic way to represent constrained motion.

In the case of manipulators possessing flexible members, the DH convention is not applicable. Within a body, the transformations between various body frames now include variables describing the flexible states. In the *assumed-mode* approach, specific mode shapes are assumed to always hold. With the help of these mode shapes and the associated generalized flexible coordinates, the position and orientation relationship between all body frames within a body can be defined. Relative joint coordinates are usually used also to define the state of a body with respect to the preceding one.

In any case, the motion of the robot can be defined in two different sets of coordinate systems. In joint space (or configuration space), the motion is described by the generalized coordinate vector $\mathbf{q}$. Essentially, $\mathbf{q}$ is composed of $\mathbf{q}_r$ the *joint coordinate* and $\mathbf{q}_e$ the *elastic generalized coordinates* ($\mathbf{q} = \{ \mathbf{q}_r, \mathbf{q}_e \}^T$). The entries of $\mathbf{q}_r$ are usually easy to measure using simple sensing devices at the joints and, therefore, control of a manipulator is mostly considered in joint space. The entries of $\mathbf{q}_e$ are more difficult to measure and are usually observed using strain gauges or piezoceramic sensors. The motion can also be defined in the operational space (or task space) by a set of coordinates $\mathbf{x}=\{\mathbf{x}_e, \mathbf{q}_e\}$ where $\mathbf{x}_e$ represents the absolute motion of the end effector. These coordinates are more difficult to measure directly. New sensing devices and methods are currently being developed for this purpose (e.g., vision systems based on camera signals and proximity sensors). In the case of redundant manipulators (more joints than DOF), end effector motion in Cartesian space is not a sufficient set of coordinate. In this thesis, the discussion is limited to non-redundant manipulators.

The relation between the joint and the operational space, called the forward kinematics, is given by the surjective nonlinear mapping $\Gamma:\mathbb{R}^n \rightarrow \mathbb{R}^n$ described by

$$x_e = \Gamma(q) \tag{2.3}$$

Differentiating equation (2.3) twice with respect to time gives

$$\dot{x}_e = \nabla_q \Gamma(q)\dot{q} = J(q)\dot{q} \tag{2.4}$$

$$\ddot{x}_e = J(q)\ddot{q} + \dot{J}(q)\dot{q} \tag{2.5}$$

where the matrix $\mathbf{J(q)}$ is called the manipulator Jacobian. The orientation of the end effector may be described using the Euler angles , the Euler quaternion or the orientation matrix.

The surjectivity of the forward kinematic makes the inverse kinematic problem complex. In fact, even for purely rigid robots, to obtain one operational space position and orientation, there may be many solutions in joint space. The selection of a particular solution is usually subject to an extra condition specifying the range of operation of some joints. In an iterative process where the inverse kinematics is solved, the solution retained is usually the one "closest" to the configuration in the previous iterative step. This ad hoc measure insures "smoothness" in the inverse kinematics solution.

Most of the time, the joints of a robot manipulator are physically limited in their motion. By defining the set of feasible joint coordinates Q, one can define the workspace by the set $X_w=\{\mathbf{x}_e|\ \mathbf{x}_e=\Gamma(\mathbf{q}),\ \mathbf{q}\in Q\}$. Therefore, to be consistent and precise, the forward kinematic is a mapping $\Gamma:Q\rightarrow X_w$. Any operational point not in X_w is singular and physically unreachable.

2.1.3 - Dynamics

Several methods were applied to obtain the equations of motion using the recursive Newton-Euler formulation or the Lagrange dynamics. In the recursive Newton-Euler method, the kinematic relationships are used to derive the expression of the generalized accelerations of each link. Starting with the most outer link, the Newton-Euler and the flexible dynamics equations are written, substituting recursively the joint reaction forces calculated from the dynamics of the previous link. This is basically the substitution method described in appendix A for rigid body systems.

The Lagrangian dynamics method is the application of the classical Lagrangian dynamics. In fact, each joint represents an ideal holonomic constraint of order five. Consequently, the kinetic and potential energy can be fully defined by using the constrained coordinate system (**q** and **q̇**), and the classical Lagrange dynamics can be applied.

In any case, the resulting equation of motion for a manipulator on a fixed rigid base is of the form

$$M_{rr}(q)\ddot{q}_r + M_{re}(q)\ddot{q}_e + N_r(\dot{q},q) + G_r(q) = F_r(\tau) - F_{f_r}(\tau_f) + J_r^t f_e \tag{2.6}$$

$$M_{er}(q)\ddot{q}_r + M_{ee}(q)\ddot{q}_e + N_e(\dot{q},q) + G_e(q) = F_e(\tau) - F_{f_e}(\tau_f) + J_f^t f_e \tag{2.7}$$

where M_{rr} is the generalized inertia matrix associated with joint motion, M_{ee} is the generalized inertia matrix associated with flexibility, M_{er} and M_{re} are the generalized inertial coupling matrices, N_r and N_e are the nonlinear terms associated with the joint and flexible motions respectively (contains the centrifugal and Coriolis terms), G_r and G_e are the gravity vectors, $F_r(\tau)$ and $F_e(\tau)$ are the generalized control force vectors, $F_r(\tau_f)$ and $F_e(\tau_f)$ are the generalized joint friction force vectors, $J=[\ J_r J_e]$ is the manipulator Jacobian for a fixed frame of reference, and f_e is the force/torque vector applied on the end effector and expressed along the same fixed frame of reference. The external force f_e is assumed to be exogenous to the model for now. For an N link robot arm, the joint friction τ_f includes all joint friction applied to each links.

In general, for rigid manipulators, the flexibility dynamics (equation 2.7) is considered as a singular perturbation to the manipulator rigid body dynamics (equation 2.6) and the equations of motion reduce to

$$M(q)\ddot{q} + N(\dot{q},q) + G_r(q) = \tau - \tau_f + J^t f_e \tag{2.8}$$

where the joint friction is simply

$$\tau_f(\tau,\dot{q},q) = \begin{Bmatrix} \tau_{f_1} \\ \tau_{f_2} \\ \vdots \\ \tau_{f_{N-1}} \\ \tau_{f_N} \end{Bmatrix} \tag{2.9}$$

Using the forward kinematics (equations 2.3, 2.4 and 2.5), the rigid manipulator dynamics can be expressed in the operational space as

$$\boldsymbol{M}_x(\boldsymbol{x}_e)\ddot{\boldsymbol{x}}_e + \boldsymbol{N}_x(\dot{\boldsymbol{x}}_e, \boldsymbol{x}_e) + \boldsymbol{G}_x(\boldsymbol{x}_e) = \boldsymbol{J}^{-T}\big(\tau - \tau_f(\tau, \dot{q}, q)\big) + \boldsymbol{f}_e \tag{2.10}$$

where $\boldsymbol{M}_x = \boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1}$, $\boldsymbol{N}_x = \boldsymbol{J}^{-T}\big(\boldsymbol{N} - \boldsymbol{M}\boldsymbol{J}^{-1}\dot{\boldsymbol{J}}\dot{\boldsymbol{q}}\big)$ and $\boldsymbol{G}_x = \boldsymbol{J}^{-T}\boldsymbol{G}$. Implicit to this definition of the robot dynamics in operational space is the existence of an inverse solution to the kinematics. This is insured in the robot workspace but at singular locations.

A thorough review of robotic manipulator dynamic models can be found in [1] and [2]. Other important issues concerning robot modelling include kinematic loops and arborescent tree topology. Kinematic loops are obtained, for example, when a manipulator grasps the environment. In this case, the extra constraint modifies the structure of the equations of motion. Tree topology are obtained when a manipulator is terminated by two smaller manipulators. These important issues require extra steps in the definition of the equations of motion, but do not modify the generality nor the validity of equations 2.6 and 2.7. The work presented in this thesis will concentrate on the control of chain topology manipulators in unilateral contact, avoiding the problems associated with tree topology and kinematic loops.

2.1.4 - Joint Friction

Joint friction is a factor that is often neglected in the analysis of control systems for robot manipulators. Nevertheless, it represents one of the important challenges in robot control. Tribology has a long history, but the complexity of the friction process is such that a unified model, including all of its particularities, is yet to be developed. The reader is referred to the appendix B for a more detailed discussion about friction.

In robot joints, friction is mainly the result of motor bearing drag and gear resistance. The motor bearings are usually prestressed to reduce the surface rubbing friction between bearings. This process also has the advantage that the resulting friction level becomes less sensitive to transverse and normal loads on the bearings, making the bearing friction level more deterministic. Gear friction, on the other hand, is difficult to characterize. The relation between load, speed, position and friction force in gears is extremely difficult to capture in a model.

Friction models can be classified according to their dynamic nature. Dynamic models represent friction as a dynamic process and include states whose evolutions are governed by nonlinear differential equations (see appendix B).

To include joint friction in a practical way, it is usually assumed that the static and kinetic friction torques and forces are independent of the robot states. In reality, static and kinetic friction are anisotropic (dependent on the contact conditions such as speed, direction of motion, position). In the robot control literature, a static model of friction known as the Coulomb model is mostly used. This model is described as

$$\tau_{f_i} = \tau_{k_i}\, sgn(\dot{q}_i) \qquad \textbf{(2.11)}$$

where τ_f is the friction torque/force, τ_k is the level of kinetic friction (assumed equal to static friction) and $\dot{q}$ is the joint speed. Since the standard Coulomb model is ill-defined at $\dot{q}$=0, an augmented Coulomb model is often introduced

$$\tau_{f_i} = \begin{cases} \tau_{k_i}\, sgn(\dot{q}_i) & if\ \dot{q}_i \neq 0 \\ \tau_{s_i}\, sgn(\tau_{eq_i}) & if\ \dot{q}_i = 0 \quad and \quad |\tau_{eq_i}| \geq \tau_{s_i} \\ \tau_{eq_i} & if\ \dot{q}_i = 0 \quad and \quad |\tau_{eq_i}| < \tau_{s_i} \end{cases} \qquad \textbf{(2.12)}$$

where $\tau_{eq,i}$ is the equivalent joint torque, $\tau_{k,i}$ is the kinetic friction torque and $\tau_{s,i}$ is the static friction torque. The equivalent torque τ_{eq} is the torque needed to maintain the stuck joint at $\dot{q}$=0 and $\ddot{q}$=0.

In practical situations, however, the friction torque is not discontinuous at $\dot{q}$=0. As described in the appendix B, the increase in friction torque at low joint speed (the Stribbeck effect) can be modelled using an exponential or a Lorentzian model. For the exponential model (called the Bo&Pavelescu model), the friction is given by

$$\tau_{f_i} = \begin{cases} \left(\tau_{k_i} + \left(\tau_{s_i} - \tau_{k_i}\right) e^{-\left(\frac{|\dot{q}_i|}{v_o}\right)^{\delta}} \right) sgn(\dot{q}_i) & if\ \dot{q}_i \neq 0 \\ \tau_{s_i}\, sgn(\tau_{eq_i}) & if\ \dot{q}_i = 0 \quad and \quad |\tau_{eq_i}| \geq \tau_{s_i} \\ \tau_{eq_i} & if\ \dot{q}_i = 0 \quad and \quad |\tau_{eq_i}| < \tau_{s_i} \end{cases} \qquad \textbf{(2.13)}$$

where v_o and δ are empirical parameters.

Models presented so far are static and are described by algebraic relations between the robot states and the friction force. They all inaccurately model friction at low speed. At this particular condition, some phenomena occurring in friction can be modelled exclusively using dynamic models (Dahl effect, friction memory, friction hysteresis). The integrated model of Canudas de Wit et al.[35] is one of the newest dynamic models. It includes one friction state z whose evolution is governed by

$$\dot{z}_i = \dot{q}_i - \frac{|\dot{q}_i|}{g(\dot{q}_i)} z_i \qquad \textbf{(2.14)}$$

where $g(\dot{q}_i)$ defines a static model obtained at steady-state. The friction force is given by the algebraic relation

$$\tau_{f_i} = \sigma_o z_i + \sigma_1 \dot{z}_i \qquad \textbf{(2.15)}$$

where σ_o and σ_1 are contact surface parameters to be identified. It represents a version of Dalh's dynamic model modified to include the Stribbeck effect (dependancy of kinetic friction on sliding velocity). More dynamic models exists and are presented in Appendix B.

The selection of an appropriate friction model depends on observations made in practice. Usually, robots with lubricated gears will exhibit a friction behaviour that can be represented by a static Bo & Pavelescu model or a dynamic model such as the integrated model with a Bo & Pavelescu steady state solution. Direct-drive motors with unlubricated motor bearings are usually well-represented using the augmented Coulomb model.

<u>*Remarks*</u>

① For all models presented in this section, viscous friction was neglected. If required, it can be added as a velocity term of the form $-f_v \dot{q}_i$ to the friction force.

② These models also assume measurable constant static and kinetic friction forces. Occasionally, this assumption is not valid and friction modelling becomes a very difficult task.

2.1.5 - The End Effector

One particular tedious and difficult task in modelling robot manipulators performing contact tasks is to define the geometric models for the end effector and external body. Usually, manipulators are equipped with special devices needed to do a particular task. The device can be a gripper, a drill, a screwdriver, or any other special tool. The end effector motion is typically defined by the motion of a point on the end effector. This point is usually considered as the point of application of all forces on the end effector. But the contact does not necessarily occur at that point (it would be a great coincidence). It would occur somewhere on the boundary of the tool. This boundary surface, designated here by the set S_b, can be described with respect to the end effector reference frame by (see figure 2.1)

$$S_b \triangleq \left\{ {}^{p}x_e + x_b \mid \phi_e(x_b, {}^{p}x_e) = 0, \ {}^{p}x_e \ given \right\} \tag{2.16}$$

where ϕ_e is the surface function defining the boundary points, x_b is a three-dimensional vector representing any point on the boundary with respect to the end effector frame and ${}^{p}x_e$ is the three-dimensional vector representing the position of the end effector frame.

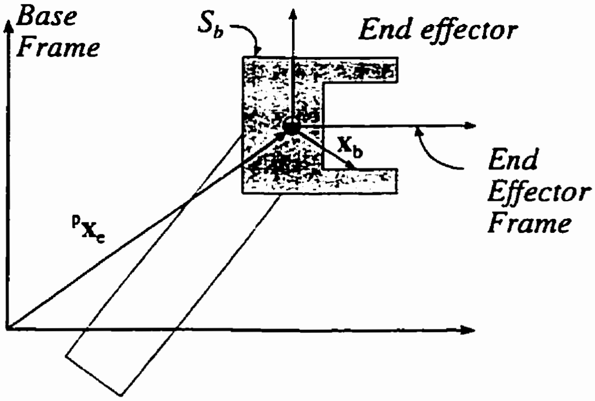


Figure 2.1 - End Effector Geometry

In practice, the object is usually broken down into smaller portions (usually convex objects) and surfaces into smooth portions. This will be discussed later in section 2.4.2. The definition presented here is general enough to include all particularities of the geometric model.

2.2 - Impact/Contact Models

In the dynamic model for the manipulator arm, the contact force f_c was assumed to be exogenous. In reality, the contact force is fully (but not only) dependent on the manipulator dynamics. The dynamics of the environment, the contact parameters and even the contact dynamics can influence the contact force. In this section, contact models are analysed. The analysis includes a study of theoretical models, and the presentation of standard simplified impact/contact models.

2.2.1 - Theoretical Models

From a microscopic point of view, contact models were well investigated. The reader is referred to Gladwell[36] or Johnson[37] for detailed discussions. In the following section, some notions about microscopic models are presented to introduce the general phenomena occurring during contact. A good understanding of these models helps to appreciate the complexity of the contact process.

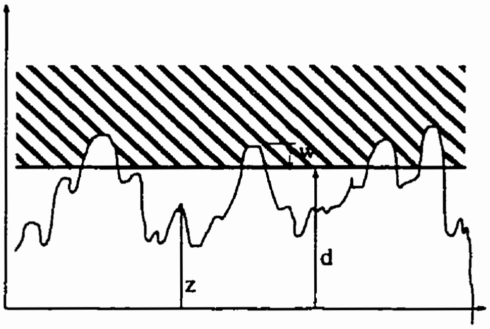


Figure 2.2 - GW Contact profile

One of the well-known elastic contact models is the model of Greenwood and Williamson[38] (GW model). The GW model is an asperity based model that evaluates the true surface of contact A_t and the contact load P as a function of the separating distance d (see figure 2.2). Using a probability density function *f(z)* representing the proportion of asperity of height z, and using Hertz theory of elastic contact for each asperity, the general solution is

$$A_t(d) = N_a\tilde{A}(d) = N_a \pi \beta \int_d^\infty (z-d) f(z)\,dz \tag{2.17}$$

$$P(d) = \frac{4}{3} N_a E' \beta^{\frac{1}{2}} \int_d^\infty (z-d)^{\frac{3}{2}} f(z)\,dz \tag{2.18}$$

where N_a is the total number of asperities in the nominal area of contact, *E'* is the apparent Young's modulus of elasticity ($1/E'=(1-v_1^2)/E_1+(1-v_2^2)/E_2$) and β is the asperity tip radius (assumed equal for each asperity). This model has promoted a better understanding of the contact phenomena at the microscopic level. It has explained why the contact rigidity observed during a contact is not as high as expected from a simple stress analysis.

Contact is known to be a dissipative process. This is well represented in the central direct impact theory by the use of the coefficient of restitution. The GW model is, however, purely linear-elastic and is therefore completely conservative. It fails to represent dissipation mainly because it neglects plastic deformation at the asperity level. Plasticity was introduced in a GW-like model by Chang et al.[39], who have used a volume conservation approach to introduce plastic deformation. Their model resulted in

$$A_t(d)=N_a\pi\beta\left\{\int_d^{d+w_c}(z-d)f(z)dz+\int_{d+w_c}^{\infty}\left(2(z-d)-w_c\right)f(z)dz\right\} \tag{2.19}$$

and

$$P(d)=N_aE'\left\{\frac{4}{3}\beta^{\frac{1}{2}}\int_d^{d+w_c}(z-d)^{\frac{3}{2}}f(z)dz+\frac{\pi\beta\sigma_{yield}}{E'}\int_{d+w_c}^{\infty}\left(2(z-d)-w_c\right)f(z)dz\right\} \tag{2.20}$$

where σ_{yield} is the yield strength of the weakest material in contact and w_c is the critical deformation of an asperity at the onset of plasticity and is given by the Hertz theory of contact as

$$w_c=\left(\frac{\pi\sigma_{yield}}{2E'}\right)^2\beta \tag{2.21}$$

Practically, the yield strength for various metallic materials is related to the Brinell hardness by σ_{yield}=KH where H is the Brinell hardness converted to Pa (for ferrous materials, K≈0.6)[40].

An comparaison between the GW model and Chang's model is presented in example 2.1. As anticipated, the linear model overestimates the local contact rigidity and underestimates the true area of contact.

Example 2.1

Consider the contact between two flat steel surfaces over a 1 cm² portion. The properties of steel are E=207 Gpa, v=.29 and the Brinell Hardness H=200 (1960 Mpa). Assuming a normal

distribution of the asperity height with a mean of 10μm and a standard deviation of 1 μm, with an asperity tip radius of 5 μm and 1.0x10⁸ asperity/m², the contact load and the true surface of contact can be estimated as shown here.

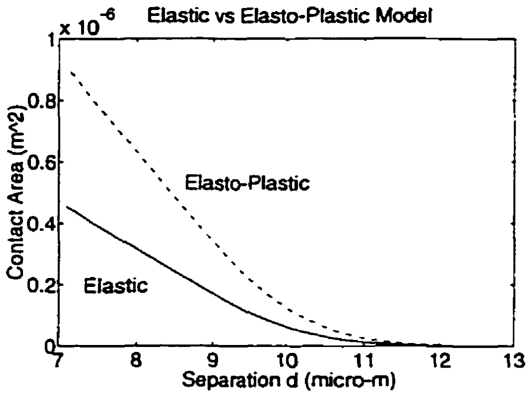


Figure 2.3 - True Contact Area

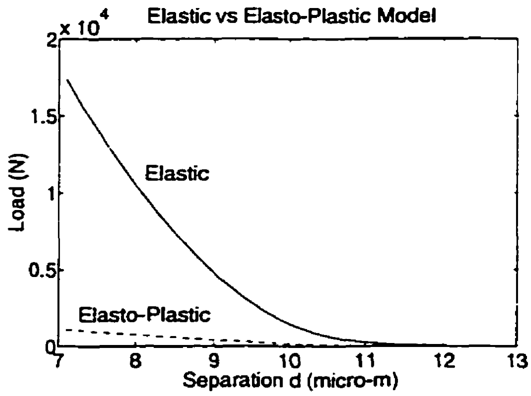


Figure 2.4 - Contact Load

The elasto-plastic model is a valid model of contact only during the loading portion of the contact, until the separation distance d starts to increase. This is caused by the plastic deformation of the asperities modifying the asperity properties and inducing a new relationship between the penetration distance and the contact force. This intuitive observation has motivated Chang et al.[41] to modify their elasto-plastic model to take into consideration the restitution part of the process. Briefly, they had separated the impact phenomenon at the asperity level in three distinctive phases. First, the phase where the elastic strain energy is stored in the vicinity of the impact site as the asperity deforms elastically. Second, the phase where the conditions for plastic deformation arise, and the asperity starts to deform plastically. Then, any additional work is lost in plastic deformation. Third, the phase where only the elastic energy stored is restored. Assuming a continuity in the load and area of contact from the loading phase to the unloading one, they evaluate new asperity properties. They use a purely linear-elastic GW model to represent unloading. Using this approach, they obtain the smallest separation distance d_0 by equating the work done during loading and the difference in kinetic energy of the contacting bodies before impact and when the bodies come to rest relative to each other

$$\begin{aligned} W_l(d_o) &= N_a E'\left\{\frac{8}{15}\beta^{\frac{1}{2}}\int_d^{d+w_c}(z-d)^{\frac{5}{2}}f(z)\,dz+\frac{8}{15}\beta^{\frac{1}{2}}\int_d^{d+w_c}w_c^{\frac{5}{2}}f(z)\,dz+\frac{\pi\beta\sigma_{yield}}{E'}\int_{d+w_c}^{\infty}(z-d)(z-d-w_c)f(z)\,dz\right\} \\ &=\sum E_k^{initial}-\sum E_k^{rest} \end{aligned} \tag{2.22}$$

They finally get the coefficient of restitution by computing the square root of the ratio between the unloading work given d_o and the kinetic energy lost at impact

$$e^2 = \frac{\frac{8}{15}N_a E'\beta^{\frac{1}{2}}\left\{\int_{d_o}^{d_o+w_c}(z-d_o)^{\frac{5}{2}}f(z)dz+w_c\int_{d_o-w_c}^{\infty}\left(2(z-d_o)-w_c\right)^{\frac{3}{2}}f(z)dz\right\}}{\sum E_k^{initial}-\sum E_k^{rest}} \tag{2.23}$$

This model of impact represented accurately the experimental observations about the dependancy of the coefficient of restitution with respect to impact velocity.

2.2.2 - Simplified Models

The models presented in the preceding section are good theoretical representations of the fundamental process underlying contact and impact. On the other hand, their use for model-based control implementations is impractical. The complicated integral limits the feasibility of real-time computation. Simplifications are consequently needed to obtain practical design models.

The results of example 2.1 have demonstrated that elastic contacts do not provide contact forces linearly dependent on the penetration distance d. In fact, the transition to contact is relatively smooth, the contact being achieved first with the highest asperities. The force-separation relationship becomes linear once the contact is fully developed. Typical models of contact used in the literature on manipulator contact motion assumes that contact rigidity is constant right away. One of the simplified elastic model used to obtain the contact load P is

$$P(d) = \begin{cases} -K_e\ d & if\ d<0 \\ 0 & if\ d\geq 0 \end{cases} \tag{2.24}$$

where K_e defines some contact stiffness to be obtained by experiments or a thorough analysis of the contact geometry and conditions. This model is conservative, continuous but nonsmooth ($P(d) \in C^0$). Its conservative nature does not allow the model to represent impact properly. Some have used

$$P(d) = \begin{cases} -K_e\ d - f_e\ \dot{d} & if\ d<0 \\ 0 & if\ d\geq 0 \end{cases} \tag{2.25}$$

where f_e is a viscous damping factor to include dissipation. This model is dissipative, but is not motivated by a good understanding of physics. The addition of the viscous damping term is an artifact to include some kind of damping, perhaps valid just in case of a wet surface.

In some numerical problems, the use of a model comprising a C^0 function may lead to ill-conditioned numerical situations. This is the case for problems where Newton-Raphson type numerical methods are needed (Newton-like optimization methods, shooting methods). For these specific problems, it is usually required to *relax* the discontinuity by including a smooth approximation. For the contact models, this relaxation is well justified since the true phenomena, best described by the model of equations (2.19) and (2.20), is smoother than usually considered. Consequently, using a sigmoid function to relax the transition from free to contact condition is physically valid. For example, an exponential function by can be added to the model (2.24) to give

$$P(w) = -K_e \left(1 - e^{-\left(\frac{|d|}{\epsilon}\right)^{\delta}} \right) d \tag{2.26}$$

where ϵ defines the relaxation boundary layer and δ is the decay factor.

Another typical design model of contact is the purely rigid contact model. In this case, the contact stiffness is assumed infinite and the penetration distance is always considered null. The force of contact is assumed to be represented by a constraint force, constraining the motion of the end effector to the surface of contact. In other words, this model allows the consideration of the contact condition as an extra holonomic constraint on the manipulator motion. Following the formalism shown the appendix A, one can consequently define the constrained motion dynamics and the contact force algebraic relation to model the contact force. More discussion about this contact model will be presented in section 3.3.5 where the Reduced Order Model Control is reviewed. At this point, stating that the purely rigid model is described by the impenetrability condition

$$d \geq 0 \tag{2.27}$$

and the intensibility condition

$$d \cdot P = 0 \tag{2.28}$$

is sufficient. These conditions formalize the notion of purely rigid contact.

None of the simplified models presented so far includes impact. In practice, the classical purely rigid impact model is most commonly used. This model is not revised in detail here, as it is a classical model well described in many references(see Meriam [42]). Nevertheless, it will be used as the basis for a more practical model for the study of unilateral contact.

The classical impact model assumes that the unloading impulse is a fraction of the loading impulse, and that the ratio is a constant defined as the coefficient of restitution

$$e = \frac{\int_{t_{load}}^{t_{load}+t_{unl}} F_{unl}\,dt}{\int_{0}^{t_{load}} F_{load}\,dt} \tag{2.29}$$

where t_{load} and t_{unl} are the loading and unloading times respectively, and F_{load} and F_{unl} are the loading and unloading forces respectively. If the impulse times are equal, one can define some average loading and unloading forces giving

$$e = \frac{\bar{F}_{unl}}{\bar{F}_{load}} \tag{2.30}$$

Considering the contact model of equation (2.24), a modification of contact force that would observe the relation (2.30) is

$$P(d,\dot{d}) = \begin{cases} K_e\, d & \textit{if } d<0 \textit{ and } \dot{d}\leq 0 \\ eK_e\, d & \textit{if } d<0 \textit{ and } \dot{d}>0 \\ 0 & \textit{if } d\geq 0 \end{cases} \tag{2.31}$$

This model is dissipative in the sense that the work done in loading is only partly restituted during unloading. This model of contact dissipation is more representative of the contact/impact phenomena than the viscous model of equation (2.25).

Finally, the classical purely rigid impact model does not take into consideration the dependance of the coefficient of restitution with respect to the impact velocity. To complete the impact model, an empirical relation can be used to evaluate the velocity dependent coefficient of restitution

$$e(v_{imp}) = \begin{cases} e_{\max} & \textit{if } v_{imp} \leq v_c \\ e_{\min} + (e_{\max} - e_{\min})\, e^{-\left(\frac{v_{imp} - v_c}{\beta}\right)^{\beta}} & \textit{if } v_{imp} > v_c \end{cases} \tag{2.32}$$

where β, e_{min}, e_{max} and δ are empirical parameters, and v_{imp} is the impact velocity. The meaning of equation (2.32) is given in figure 2.5.

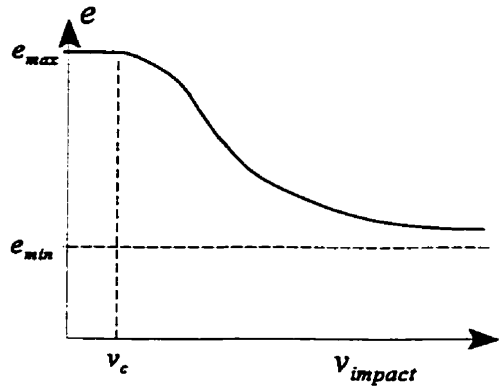


Figure 2.5 - Coefficient of Restitution vs impact speed

2.2.3 - Surface Friction

The tasks performed by a robot manipulator sometimes require the control of force in a direction normal to the contact surface and position in the tangential direction. The tangential motion is obviously influenced by the surface friction developed at the contact site. In opposition with joint friction where most of the time the prestressed components make the static and kinetic friction levels constant, the surface friction definitively depends on the level of normal force. One of Amonton's law of friction states that the resistance to motion (friction force) is proportional to the contact load and independent of contact area, and is described as

$$F_k = \mu_k \; P(d) \tag{2.33}$$

and

$$F_s = \mu_s \; P(d) \tag{2.34}$$

This observation, proven in practice, has for a long time defied common sense.

This century brought light to this intriguing problem. Nowadays, it is known that contact does not occur over the entire nominal surface in contact, but occurs only over a fraction of it. This is well described by the asperity model of GW (section 2.2.1). In fact, the laws of friction should stipulate that the friction force is proportional to the true contact area as shown in the following equation

$$F_k = \kappa_k A(d) \tag{2.35}$$

where F_k and κ_k define the kinetic friction force and proportionality constant respectively. Similarly,

$$F_s = \kappa_s A(d) \tag{2.36}$$

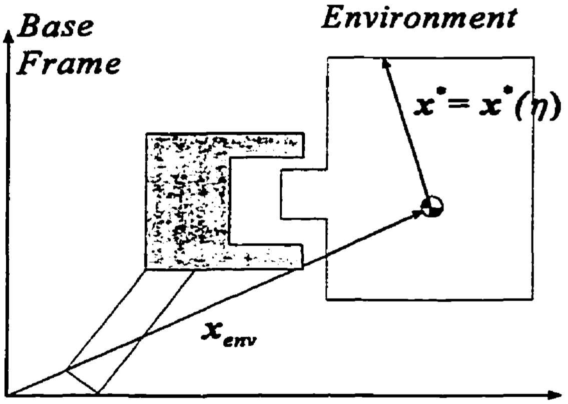
for the static friction force. The true area of contact is essentially proportional to the normal contact load, such that Amonton's law is valid for a wide range of contact conditions. The new law of friction is definitively more intuitive as one can understand that friction is a force necessary to break the contact between the asperities. More asperities in contact require more effort to break the bounds (friction forces).

The models presented for joint friction are also valid for surface friction, provided the integration of the load dependent static and kinetic friction forces in the equations. Obviously, the contact and the friction phenomena are closely linked. Effort is being put in trying to develop a comprehensive unifying model that would permit to representation of friction and contact forces in one common model. Until this model is found, the link between the surface friction and the normal load relies exclusively on the true contact area.

2.3 - Environment Dynamics

So far, the manipulator arm dynamics and the contact models were discussed. The effect of the contact force on the motion of a robot manipulator is described by equations (2.5), (2.6) and (2.7). But the contact force also applies to the environment. It is therefore important, to obtain a complete model, to include the dynamics of the environment.

As for the end effector, modelling the environment model requires the development of a geometric model. The field of geometric



x^* : 3-D vector defining the environment boundary

Figure 2.6 - Environment Geometry

modelling is in constant evolution and only some aspects will be briefly described here. For the particular study presented in this thesis, it is assumed that the environment can have some flexible dynamics described by the shape coordinate vector η and a rigid body motion defined by the coordinate $\mathbf{x}_{env}$ of a point on the body (usually the centre of mass). In a way similar to the end effector boundary points, the surface of the environment (S_{env}) is represented by the set (see figure 2.6)

$$S_{env} \triangleq \left\{ x^*(\eta) + x_{env} \mid \phi_{env}\left(x^*, x_{env}\right) = 0 \right\} \tag{2.37}$$

where $\mathbf{x}^*$ is a vector defining the location of the boundary with respect to a point on the undeformed environment and ϕ_{env} is the environment surface description function. This function defines the environment location of the surface at a time t, knowing the environment coordinates η and $\mathbf{x}_{env}$.,, and the mode shapes of the environment. Again, as for the end effector, the environment surface is usually broken down into smooth segments composed of flat and quadratic surfaces. The motion of the environment is governed by its dynamic equations, described in general terms by

$$\begin{Bmatrix} \dot{x}_{env} \\ \ddot{x}_{env} \\ \dot{\eta} \\ \ddot{\eta} \end{Bmatrix} = f_{env}\left(x_{env}, \dot{x}_{env}, \eta, \dot{\eta}\right) + g_{env}\left(x_{env}, \dot{x}_{env}, \eta, \dot{\eta}\right) \begin{pmatrix} -f_e \\ f_{ext} \end{pmatrix} \tag{2.38}$$

where f_{env} and g_{env} define the environment dynamics, f_{ext} is the vector of exogenous forces applied to the environment and f_c is a force vector representing the interaction between the environment and the robot as defined in equation (2.8). Environment models for which f_{ext}=0 are said to be passive while those with a force $f_{ext} \neq 0$ are said to be active.

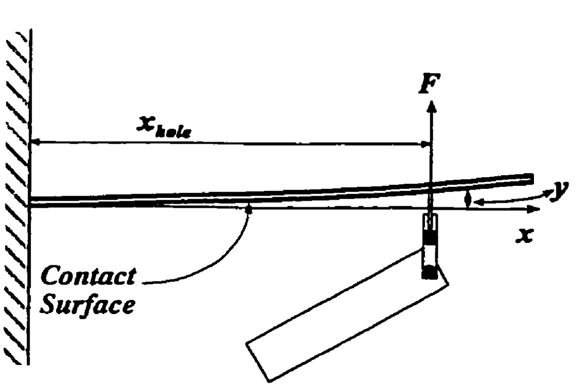


Figure 2.7 - Example 2.2

Example 2.2 - Environment Model

Consider a cantilevered plate on which a robot manipulator has to drill a hole (see figure 2.7). The plate has flexibility with one dominant mode of vibration. Therefore, the position of the surface of the plate is given by the coordinates x and y that verify

$$y - \varphi_1(x)\eta_1 = 0 \tag{2.39}$$

where φ_l is the mode shape of the first mode of vibration and η_l is the mode coordinate. The vibration dynamics can be described by

$$\ddot{\eta}_1 + 2\zeta_1\omega_1\dot{\eta}_1 + \omega_1^2\eta = \omega_1^2\ k_1^{-1}\varphi_1(x_{hole})F \tag{2.40}$$

where ζ_l is the modal damping of the first mode, ω_l is the natural frequency of the first mode (in rad/sec), k_l is the generalized stiffness of the first mode and F is the normal contact force.

2.4 - Manipulator with Unilateral Frictional Impact and Contact[43]

In the previous sections, models for robot manipulators, contact, impact, friction and environment were considered separately. In this section, they are joined to generate the complete dynamic model of a rigid robot arm contacting an object in its environment.

2.4.1 - Analytical Settings

The location of the undeformed surfaces of the end effector and of the environment are given respectively by the surface points defined by the sets shown in equations (2.16) and (2.37). In both cases, the surface function is selected such that the function's gradient is pointing away from the surface. The normal vector **n**, pointing away from the environment surface is defined as (see figure 2.8)

$$\boldsymbol{n}(\boldsymbol{x},\boldsymbol{x}_{env},\eta) = \frac{\nabla_{e_x}\phi_{env}(\boldsymbol{x},\boldsymbol{x}_{env},\eta)}{|\nabla_{e_x}\phi_{env}(\boldsymbol{x},\boldsymbol{x}_{env},\eta)|} \tag{2.41}$$

where $\mathbf{e}_x$ is an orthonormal basis for the fixed frame of reference. The contact force acting on the end effector contact site is given by

$$\boldsymbol{f}_e = P(d)\ \boldsymbol{n} + F_f\ \boldsymbol{e}_t \tag{2.42}$$

where $\mathbf{e}_t$ is the tangential vector along the tangential motion of the end effector with respect to the environment, F_f is the surface friction and d is the separation distance defined as

$$d \triangleq \min_{z\in Z} z \tag{2.43}$$

where Z is the set of values z given by

$$Z = \left\{ z \,|\, \boldsymbol{x}_{env} + \boldsymbol{x} + z\boldsymbol{n} = {}^{P}\boldsymbol{x}_e + \boldsymbol{x}_b,\ \boldsymbol{x}\in S_{env},\ \boldsymbol{x}_b \in S_b,\ {}^{P}\boldsymbol{x}_e \ is\ given \right\} \tag{2.44}$$

In the definition represented by equation (2.44), the leading superscript p refers to the position component of the position/orientation vector of the end effector.

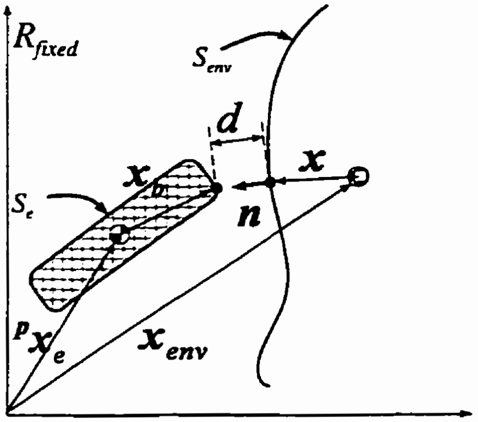


Figure 2.8.a - Contact Geometry with d>0

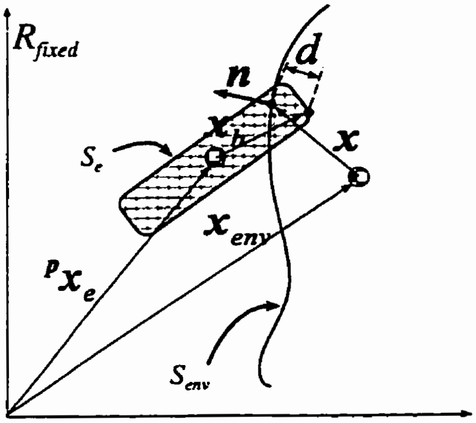


Figure 2.8.b - Contact Geometry with d <0

The relative speed of the end effector with respect to the environment is defined by

$$v_{rel} = \frac{d}{dt}\left({}^p x_e + {}^c x_b\right) - \frac{d}{dt}\left(x_{env} + {}^c x\right) = \dot{d}n + v_t e_t \tag{2.45}$$

where the leading superscript c refers to the vectors resulting from solving equation (2.43). Following this definition, we obtain

$$\dot{d} = v_{rel} \cdot n \tag{2.46}$$

$$e_t = \frac{v_{rel} - (v_{rel} \cdot n)n}{| v_{rel} - (v_{rel} \cdot n)n |} \tag{2.47}$$

The contact force f_c applies to the contact site. Since the manipulator Jacobian of equation (2.4) is defined for the variations of position of the end effector frame with respect to variations in the joint coordinates, the matrix **J** in equation (2.6) should be adjusted to take into account that f_c is applied to the contact point (see Shahinpoor[44] chapter 7). This adjusted Jacobian, called the contact Jacobian, is defined

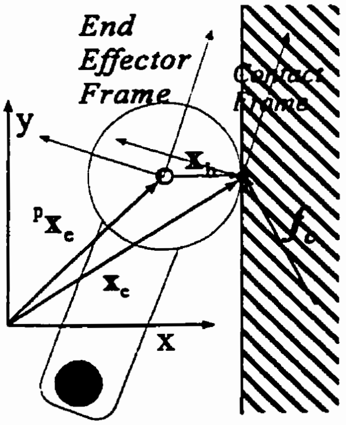


Figure 2.9 - Contact Frame

by the relationship (see figure 2.9)

$$\begin{Bmatrix} {}^{c}\dot{x}_{e} \\ {}^{o}\dot{x}_{e} \end{Bmatrix} = J_{c}(q,x_{c})\dot{q} \qquad \textbf{(2.48)}$$

where ${}^{o}\mathbf{x}_{e}$ defines the orientation component of the end effector frame. Therefore, the new equation of motion defining the robot dynamics is

$$M(q)\ddot{q} + N(\dot{q},q) + G(q) = \tau - \tau_{f}(\tau,\dot{q},q) + J_{c}^{t}f_{e} \qquad (2.49)$$

2.4.2 - Practical Considerations

The interference evaluation problem, defined by equations (2.42) and (2.43), is a complex computational geometry problem. It is not the mandate of this thesis to present research results in the field of computational geometry, but a general discussion on the subject is included.

In practice, objects are represented in a graphical data base and approximated using polyhedra (made of polygon surfaces). Sometimes, when more precision is needed, objects may be represented by collections of flat and quadratic surfaces. One approach to solve the interference problem has been to cut the end effector and the environment into sets of convex sub-bodies and to solve the mathematical programming problem for each pair of possibly interacting objects. Since the surfaces are flat or quadratic, the problem is a linear or quadratic programming problem. Moreover, given that the two objects are convex, a unique solution exits. This approach has been used by Ma and Nahon [45].

Another approach has been developed using the concept of Oriented Bounding Boxes (OBB) and the separating axis theorem [46]. It can deal solely with polyhedra representations (series of flat surfaces) but applies to non convex objects. The algorithm is not based on mathematical programming. It deals exclusively with the concepts of geometric projection and inclusion. The output does not give the interference directly but indicates if a contact has occurred and on which surface. Once a contact is detected, interference can be computed using linear programming constrained to the contacting surfaces, or simply using kinematic relationships.

2.4.3 - A Complete Model for Rigid Manipulators

Example 2.3 - A Complete Model for Rigid Manipulators

The different models for manipulator dynamics, joint friction, end effector geometry, environment dynamics, environment geometry, contact-impact and contact friction can be joined to create the overall model. Such a complete model is given in the following table.

DESCRIPTION	***EQUATIONS***
Rigid Manipulator Dynamics	$M(q)\ddot{q} + N(\dot{q},q) + G(q) = \tau - \tau_f(\tau,\dot{q},q) + J_c^t f_e$
Joint Friction	$\tau_f(\tau,\dot{q},q) = \begin{Bmatrix} \tau_{f_1} \\ \vdots \\ \tau_{f_i} \\ \vdots \\ \tau_{f_N} \end{Bmatrix}$ $\tau_{f_i} = \begin{cases} \tau_{k_i}\ sgn(\dot{q}_i) & if\ \dot{q}_i \neq 0 \\ \tau_{s_i}\ sgn(\tau_{eq_i}) & if\ \dot{q}_i = 0 \quad and \quad \lvert \tau_{eq_i} \rvert \geq \tau_{s_i} \\ \tau_{eq_i} & if\ \dot{q}_i = 0 \quad and \quad \lvert \tau_{eq_i} \rvert < \tau_{s_i} \end{cases}$ τ_{eq} *: obtained by solving a system of n equations-n unknowns* $M(\dot{q})\ddot{q}^* + N(\dot{q},q) + G(q) = \tau - \tau_{eq} + J^T f_e$ $\ddot{q}_i^* = \begin{cases} \ddot{q}_i & if\ \dot{q}_i \neq 0 \\ 0 & otherwise \end{cases}$ $\tau_{eq_i} = \begin{cases} \tau_{k_i} sgn(\dot{q}_i) & if\ \dot{q}_i \neq 0 \\ \tau_{eq_i} & otherwise \end{cases}$

Table 2.1

Manipulator and Contact Kinematics	$x_e = \Gamma(q)$ $\dot{x}_e = \nabla_q \Gamma(q)\dot{q} = J(q)\dot{q}$ $\begin{Bmatrix} {}^c\dot{x}_e \\ {}^o\dot{x}_e \end{Bmatrix} = J_c(q, x_c)\dot{q}$
End Effector Geometry	$S_b \triangleq \left\{ x_b \mid \phi_e(x_b, {}^p x_e) = 0,\ {}^p x_e \textit{ is given} \right\}$
Environment Dynamics	$\begin{Bmatrix} \dot{x}_{env} \\ \ddot{x}_{env} \\ \dot{\eta} \\ \ddot{\eta} \end{Bmatrix} = f_{env}\left(x_{env}, \dot{x}_{env}, \eta, \dot{\eta}\right) + g_{env}\left(x_{env}, \dot{x}_{env}, \eta, \dot{\eta}\right)\begin{pmatrix} -f_e \\ f_{ext} \end{pmatrix}$
Environment Geometry	$S_{env} \triangleq \left\{ x^*(\eta) + x_{env} \mid \phi_{env}\left(x^*, x_{env}\right) = 0 \right\}$
Penetration Distance	$d \triangleq \min_{z \in Z} z$ $Z = \left\{ z \mid \boldsymbol{x}_{env} + \boldsymbol{x}^* + z\boldsymbol{n} = {}^p\boldsymbol{x}_e + \boldsymbol{x}_b,\ \boldsymbol{x}^* \in S_{env},\ \boldsymbol{x}_b \in S_b,\ {}^p\boldsymbol{x}_e \textit{ is given} \right\}$
End Effector and Environment Contact Points	${}^c x_e \in S_b$ *and* ${}^c x_{env} \in S_{env}$ *such that* $\boldsymbol{x}_{env} + {}^c\boldsymbol{x} + d\boldsymbol{n} = {}^p\boldsymbol{x}_e + {}^c\boldsymbol{x}_b$
Environment Surface Normal Vector	$\boldsymbol{n}(x, x_{env}, \eta) = \dfrac{\nabla_{e_x} \phi_{env}(x, x_{env}, \eta)}{\lvert \nabla_{e_x} \phi_{env}(x, x_{env}, \eta) \rvert}$

Contact Force	$f_e = P(d,\dot{d})\,\mathbf{n} + F_f\,\mathbf{e}_t$
Normal Contact Load	$P(d,\dot{d}) = \begin{cases} K_e\,d & \text{if } d<0 \text{ and } \dot{d}\le 0 \\ eK_e\,d & \text{if } d<0 \text{ and } \dot{d}>0 \\ 0 & \text{if } d\ge 0 \end{cases}$
Penetration Speed	$\dot{d} = \mathbf{v}_{rel}\cdot\mathbf{n}$
Contact Points Relative Velocity	$\mathbf{v}_{rel} = \frac{d}{dt}\left({}^{p}\mathbf{x}_e + {}^{c}\mathbf{x}_b\right) - \frac{d}{dt}\left(\mathbf{x}_{env} + {}^{c}\mathbf{x}\right) = \dot{d}\,\mathbf{n} + v_t\mathbf{e}_t$
Direction of Tangential Motion	$\mathbf{e}_t = \frac{\mathbf{v}_{rel} - (\mathbf{v}_{rel}\cdot\mathbf{n})\mathbf{n}}{\lvert \mathbf{v}_{rel} - (\mathbf{v}_{rel}\cdot\mathbf{n})\mathbf{n} \rvert}$
Surface Friction	$F_f = \begin{cases} F_k\ sgn(v_t) & \text{if } v_t \ne 0 \\ F_s\ sgn\left(\mathbf{F}_{eq}\cdot\mathbf{e}_t(\mathbf{x})\right) & \text{if } v_t = 0 \text{ and } \lvert \mathbf{F}_{eq}\cdot\mathbf{e}_t(\mathbf{x}) \rvert \ge F_s \\ \mathbf{F}_{eq}\cdot\mathbf{e}_t(\mathbf{x}) & \text{if } v_t = 0 \text{ and } \lvert \mathbf{F}_{eq}\cdot\mathbf{e}_t(\mathbf{x}) \rvert < F_s \end{cases}$ $\mathbf{F}_{eq} = \mathbf{J}_c^{-t}\left(\tau - \tau_f - \mathbf{G} - \mathbf{N} - \dot{\mathbf{J}}_c\dot{\mathbf{q}}\right)$ $F_k = \mu_k P(d,\dot{d})$ $F_s = \mu_s P(d,\dot{d})$ *NOTE : In contact friction, inertial forces are neglected in F_{eq} because they would cause the overall system to be over-constrained.*

Table 2.1 - Complete Model

Chapter III

Problem Formulation

The models presented in the preceding chapter are relatively complex. Even simplified, their solution is difficult to achieve. With such complexity, defining precisely the conditions of analysis is important. In this chapter, the control problem is formulated with enough precision to restrain the field of specific problems investigated. Some conditions of contact will be considered, while others may only be contemplated in discussions. The problems specifically resulting from the plant model are discussed in some detail, both from the point of view of theory and practice. Finally, in the view of the formulated problem, the methods proposed in the literature are reviewed and discussed.

3.1 - The Control problem

3.1.1 - Generalities

Generally, four types of control system problems exist: regulation, point to point control, path following and tracking[1]. The problem of path following is the primary focus of this thesis. The general control scheme is illustrated by the simplified block diagram of figure 3.1. The main objective of the control system design process is to define an impact/contact motion controller that can control the robot motion, the contact force and the environment motion, such as to follow a preset force/position trajectory. Here, the problem of trajectory following is made difficult by the presence of a non smooth evolution in the dynamic equations. Existing solutions have considered

[1] *In the regulation problem, the reference is fixed and the objective is to maintain a preset value of the controlled variables making disturbance rejection the main issue. In point to point control, the reference changes by steps and the control performance also includes transient response characteristics such as settling time and overshoot. In the path following problem, the controlled system is asked to follow a preset path. This is usually the case for robot control. Finally, for the tracking problem, the system is asked to follow a trajectory whose future evolution is not known, as in the case of missile guidance.*

the global situation by splitting the overall control problem into subproblems, each representing a path following case with a smooth dynamical system. These solutions will be discussed in more details in a subsequent section of this chapter. This thesis is concerned with the definition of a unified solution for the general non smooth dynamical system.

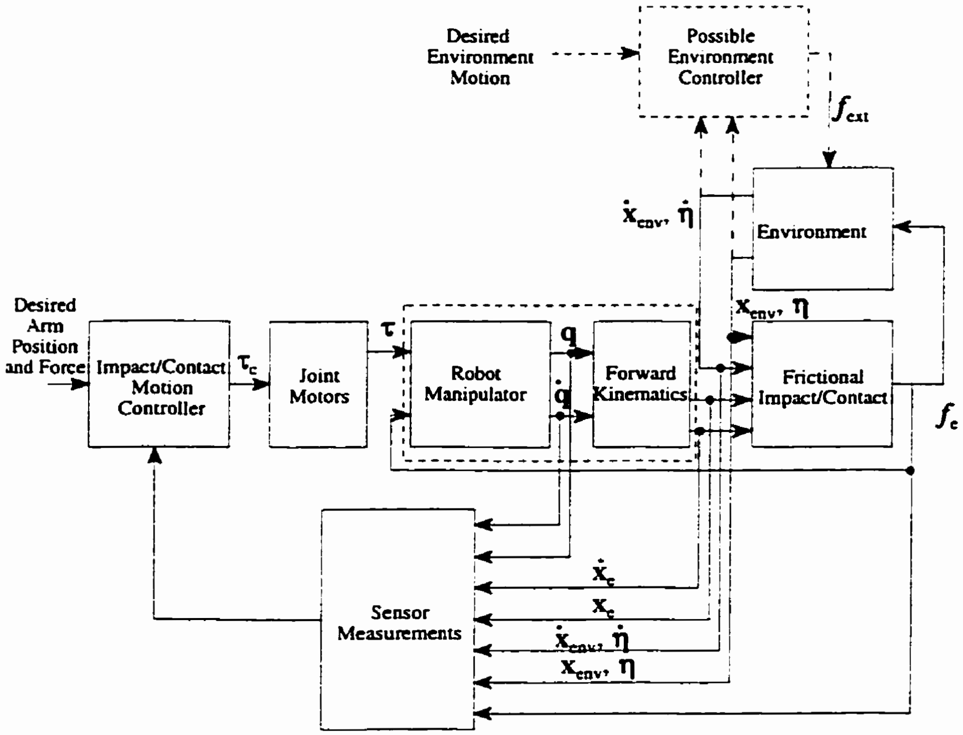

Desired Environment Motion
Possible Environment Controller
f_{ext}
$\dot{x}_{env}$, $\dot{\eta}$
Environment
x_{env}, η
Desired Arm Position and Force
Impact/Contact Motion Controller
τ_c
Joint Motors
τ
Robot Manipulator
q
$\dot{q}$
Forward Kinematics
Frictional Impact/Contact
f_c
Sensor Measurements
$\dot{x}_c$
x_c
$\dot{x}_{env}$, $\dot{\eta}$
x_{env}, η

Figure 3.1 - Simplified Control Block Diagram

In figure 3.1, most of the variables that can be measured are assumed available for feedback. In reality, the number of feedback signals is usually limited and includes only some of those presented in the figure and observers are usually needed to complete the state feedback. The signals necessary for control depend on the type of contact and environment. For example, measuring the environment coordinates (x_c, $\dot{x}_c$, η and $\dot{\eta}$) may be necessary for active environments. The states of the environment and its controller, may be difficult to observe (or maybe unobservable) when relying solely on contact force measurements. This is the case when some environment states do not contribute at all to the contact force. Conversely, for passive environments, all extra measurements may help to verify in real-time the validity of the models used.

3.1.2 - Statement of the Problem

The control system problem can be formulated as follows.

Problem 3.1 - General Formulation

For the robot manipulator with frictional unilateral impact/contact described in example 2.3, define a control law of the form

$$\tau_c = \tau_c(y, y_d) \tag{3.1}$$

that achieves, in a unified framework, the following control objectives.

1. *Control free motion providing measures to avoid excessive high speed impacts.*
2. *Control the transition from free motion to contact conditions (stable contact).*
3. *Control the motion along the contact surface and force in the normal direction.*

In this formulation, the feedback control law can be any relationship between measurements and control commands. For model-based methods, the model of the system to control becomes an integral part of the control law, and therefore one can define an estimated model parameter vector θ such that the control law becomes

$$\tau_c = \tau_c(y, y_d, \theta) \tag{3.2}$$

3.1.3 - Scope

This definition of the control problem is still general and vague. For the research work presented in this thesis, the problem is reduced for simplicity. The method proposed, however, may be adapted to embrace the general control problem. The assumptions are as follow.

Assumption 3.1 - Undisturbed Passive Environment

The environment is assumed passive, meaning the external force $f_{ext}=0$ (see equation 2.38).

Assumption 3.2 - Constrained Environment

The environment is constrained in the fixed frame of reference. Therefore, the location of the centre of mass x_{env} is constant and no dynamics are associated with this variable in equation (2.38).

Assumption 3.3 - Highly Rigid Environment

The environment is assumed highly rigid such that the natural frequency of the first mode of vibration is at a frequency well above the desired close-loop bandwidth of the impact/contact motion controller. Therefore, the flexibility coordinate $\eta=0$ and no flexible dynamics appear in equation (2.38).

Assumption 3.4 - Static Friction

The joint and surface frictions are assumed well-modelled using a static model. This is usually the case for dry sliding contacts and prestressed bearings.

Assumption 3.5 - Highly Rigid Manipulator

The Manipulator is assumed highly rigid such that the natural frequency of the first mode of vibration is at a frequency well above the desired close-loop bandwidth of the impact/contact motion controller.

The result of assumptions 3.1 to 3.4 is the absence of any dynamics in the environment/contact model. Therefore, the block entitled *Possible Environment Controller* in figure 3.1 is absent from the specific control problem considered. The block entitled *Environment* represents simply an algebraic relation whose output is independent of the contact and external forces. Moreover, the block *Frictional Impact/Contact* represents essentially a discontinuous algebraic relation between the robot states and the contact force. Assumption 3.5 reduces the complexity of the manipulator dynamics to demonstrate the control approach proposed without dealing with extra complexities. These assumptions, when considered with respect to the control approach proposed in this thesis, can be made without loss of generality. As it will be shown in chapter 5, they lead to more practical real-time implementable controllers, without discrediting the method if other assumptions are made.

3.2 - Specific Problems

The solution of the control problem formulated in the preceding section is prone to some particular difficulties. In this section, these specific problems are presented and discussed.

3.2.1 - Variable Topology Dynamics

The following two examples introduce the notion of variable topology dynamics (see Pfeiffer [47])

Example 3.1 - Dry Friction with Stick-slip

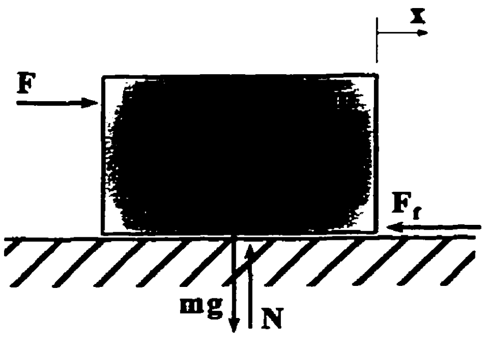


Fig. 3.2 - Stick-Slip Example

Consider the system shown in Fig. 3.2. The equation of motion of the block in the x direction is given by the Newton law of motion, that is,

$$m\,\ddot{x} = F - F_f \tag{3.3}$$

where m is the mass of the block, $\ddot{x}$ is the acceleration, F is the applied force and F_f is the force of friction. The force of friction F_f can be represented by an augmented Coulomb model defined as

$$F_f = \begin{cases} F_k\, sgn(\dot{x}) & if\ \ \dot{x} \neq 0 \\ F & if\ \ \dot{x}=0\ \ and\ \ |F| \leq F_s \\ F_s\, sgn(\dot{x}) & if\ \ \dot{x}=0\ \ and\ \ |F| > F_s \end{cases} \tag{3.4}$$

where F_k is the force of kinetic friction and F_s is the force of static friction. Unlike the standard Coulomb model (F_f=F_k sgn($\dot{x}$)), this model is defined for all values of velocities and includes stiction. Joining eqns. (3.3) and (3.4), the dynamic model of the system, defined as Σ_1, is given by

$$\Sigma_1 := \quad m\ddot{x} = \begin{cases} F - F_k & if\ \ \dot{x}>0 \\ F + F_k & if\ \ \dot{x}<0 \\ F - F_k & if\ \ \dot{x}=0\ \ and\ \ F > F_s \\ F + F_k & if\ \ \dot{x}>0\ \ and\ \ F < -F_s \\ 0 & if\ \ \dot{x}=0\ \ and\ \ |F|\ \leq F_s \end{cases} \tag{3.5}$$

The system modelled by equation 3.5 possesses two distinctive properties. First, although the system is non smooth, the vector space defined by the state vector and the force F can be broken down into subspaces in which the system undergoes a smooth evolution. In the standard Variable Structure

Systems theory (VSS), each of these subspaces is a structure of the dynamical system. The boundaries between the different structures, where the discontinuities occur, are called the switching curves (surface or hypersurface for higher dimensions). When a switching curve, or part of it, is an attractive stable manifold, it is called a sliding manifold. The properties of such sliding manifolds have been well investigated in the Russian literature and represent the basis of the Variable Structure Control (VSC) theory.

As a second property, the manifold corresponding to the last structure ($\dot{x}$=0 and $|F| \leq F_s$) is an equilibrium manifold. Therefore, when the dynamical system enters this structure, it loses its freedom of motion and gains it back only when the force F is sufficiently high to break the sticking created by the static friction force. Such a system undergoing a change in its number of degree of freedom is called a variable topology system. This terminology has been used to illustrate the fact that depending on the structure, the evolution of the dynamical system can be described in a variable dimension topological space (Topology).

The second example illustrates another common application where variable topology occurs: the unilateral constraint.

Example 3.2 - Unilateral Constraint

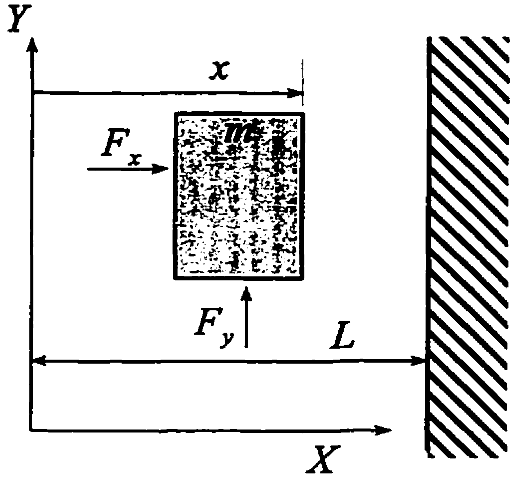


Fig. 3.3 - Unilateral Constraints

Consider the system as shown in Fig. 3.3. The free body dynamics of the system, including the constraint force, is

$$m\,\ddot{x} = F_x + C_x \tag{3.6}$$

and,

$$m\,\ddot{y} = F_y \tag{3.7}$$

where F_x and F_y are the applied forces in the x and y direction respectively and C_x is the constraint force occurring when the contact is made.

For a purely rigid contact, the unilateral constraint can be defined by the impenetrability condition

$$x - L \leq 0 \tag{3.8}$$

and by the compression condition

$$C_x \leq 0 \tag{3.9}$$

For a purely plastic impact during transition, the dynamics of the system can be defined by

$$\Sigma_2 := \begin{cases} \begin{aligned} m\ddot{x} &= F_x \\ m\ddot{y} &= F_y \end{aligned} & \text{if } \begin{cases} (x - L < 0)\, or \\ \big((x - L = 0) \text{ and } (F_x < 0)\big) \end{cases} \\ \\ \begin{aligned} \ddot{x} &= \dot{x} = 0 \\ m\ddot{y} &= F_y \end{aligned} & \text{if } \big((x - L = 0) \text{ and } (F_x \geq 0)\big) \end{cases} \tag{3.10}$$

The two properties of system Σ_1 are also apparent for this system. The non smooth evolution of the dynamics is well represented by the second structure shown in equation 3.10. However, the variable topology aspect depends greatly on the assumption made about the impact/contact model. For example, with the assumption of purely elastic contact, the number of degrees of freedom is maintained and only the dynamic equations change discontinuously.

The analytical and numerical solutions of systems such as systems Σ_1 and Σ_2 have been studied only very recently in the field of contact mechanics. This was achieved using the formalism of convex analysis (see ref. [48-54]). The first objective of these analyses is to explore the existence and uniqueness properties of the solutions for such models. From a pragmatic point of view, however, the real system definitely has a solution since its motion exists.

For control system design purposes, a generalization of the formulation of a variable topology system is necessary. Recalling that for control design and analysis, a system is usually represented by a state equation

$$\begin{aligned} \dot{x}_s &= f(x_s, u, w, t) \\ y &= h(x_s, u, v, t) \end{aligned} \tag{3.11}$$

where $\mathbf{x}_s$ is the state vector, **u** is the control vector, **w** is the state noise vector, **y** is the output vector, **v** is the measurement noise vector and t is the time, a general formulation of a variable topology system is given through the following definitions.

Definition 3.1 *- Generalized state vector*

Let X be a sequence of state vectors $\{x_i \mid i=1, \ldots ,N\}$, *the smallest dimension vector* x_f *such that* $\exists T_i - x_i=T_i x_f \; \forall i=1, \ldots ,N$, *is called the generalized state vector.*

Definition 3.2

The set W is the set of all admissible pairs of vectors (x_i,u).

Definition 3.3 *- Variable Structure System*

A system Σ *defined by*

$$\Sigma \;:=\; \begin{cases} \dot{x}_i = f_i(x_i,u,w_i,t) \\ y = h_i(x_i,u,v,t) \end{cases} \quad if\ (x_f\ ,u)\in W_i \qquad i=1,\ldots,N \tag{3.12}$$

where for $i\neq j$, 1) $f_i\neq f_j$ *or* $h_i\neq h_j$ *(nonequivalent structures)*
2) $W_i \cap W_j=\emptyset$ *(disjoint structures)*
and where $W_i \subseteq W$, *is called a variable structure system (VSS).*

Definition 3.4 *- Variable Topology System*

A VSS system Σ *for which there exist a pair* $\{i,j\}$ *such that* $dim(W_i)\neq dim(W_j)$ *is called a variable topology system.*

Definition 3.5 *- Complete System*

The VSS system Σ *is called complete if the set* $\{W_i \mid i=1,\ldots,N\}$ *forms a basis of W.*

As described, variable structure and variable topology systems are dynamical systems characterized by a dynamic model that varies discontinuously according to the state and control vectors. An important observation is that although their dynamic models change, these systems keep the same control and the same output vector in every structure.

The general model of the robot manipulator with unilateral frictional impact/contact, presented in example 2.3, is a variable topology model. By assuming the absence of dynamics in the environment and in the contact, it has been reduced simply to variable structure system. Nevertheless, most often, dynamical systems with unilateral contact exhibit a variable topology when the contact model contains dynamics or when the contact is assumed purely rigid. The existence and uniqueness of a mathematical solution for such models, is never guaranteed, making them extremely difficult to use for model-based control.

3.2.2 - The Admissible Reference

For obvious reasons, the reference trajectory has to be physically realisable. For example, given a structure W_i, the set W^*_i of all possible pairs of vectors ($\mathbf{x}_f$,$\mathbf{u}$) that may achieve **r** at time t is given by

$$W_i^* \triangleq \boldsymbol{h}_i^{-1}(\boldsymbol{r}) \qquad \textbf{(3.13)}$$

such that the set YD of all admissible references is given by

$$YD = \left\{ r(t) \mid \exists i \Rightarrow W_i^* \cap W_i \neq \emptyset \right\} \qquad \textbf{(3.14)}$$

meaning that the reference should be achievable in at least one structure.

Example 3.3 - Admissible Trajectory

Consider the system defined in example 3.2. Any trajectory $\boldsymbol{r}=\{x_d, y_d, C_{xd}\}$ *in the set* $\{\boldsymbol{r} \mid C_{xd}=0 \Rightarrow xd \leq L$ *and* $C_{xd} \neq 0 \Rightarrow x_d=L \}$ *represents an admissible trajectory.*

For the control of a robot manipulator with frictional/unilateral contact, it has obvious implications. A reference trajectory requesting the position of the end effector to be away from the contacting surface cannot simultaneously command a contact force. For physical reasons, the controller could not track the reference. Therefore, reference trajectory planning is an important part of the control effort. In any case, a good control scheme should have the potential to deal with a situation where an inadmissible reference is requested.

3.2.3 - The Impact/Contact Model

To obtain a reasonable impact/contact model, few assumptions were made. The use of more comprehensive and complex contact models is not recommended since it would result in difficulties with control system design and implementations. As a result, the design model is strongly uncertain and its non smooth nature is prone to numerical difficulties. This aspect limits the choice of control algorithms that can be used, as it will be discussed later.

The friction models have also their own limitations. The model presented in example 2.3 is valid only for rubbing friction, and does not include a plowing term. It invalidates the model when the task performed requests material removal for example.

3.2.4 - Bandwidth Limitations

The complexity of the phenomena occurring during impact and contact is well documented in Eppinger and Seering[55]. For the design of free motion control for rigid robots, fast dynamics associated with the sensors, the structure, the actuators and the digital implementation are usually neglected. During impact and contact, however, the dynamics to be controlled are at frequencies that approach the ones at which these become non negligible. Moreover, at impact and during contact, some dynamics such as the dynamics of the environment/workpiece and of the force sensor, become important. Therefore, the difficulty in contact motion control is to meet the control objectives without exciting these dynamic modes. During impact, this requires to achieve the proper damping or to insure that the transition is smooth enough to avoid exciting them. Fiala and Lumia[56] and Paljung et al.[57] have investigated the effect of digital implementation on contact stability. Qian and De Schutter[58] have studied the effect of Coulomb friction and low pass filtering on contact stability, and Zheng and Fan[59] have reported their analysis of the interaction between force sensors and impact control.

The necessity of limiting the bandwidth of the force signal is easily understandable. Both the force sensor and the robot actuators are bandwidth-limited. Any modes at a frequency above the sensor/actuator bandwidth will be weakly controllable and observable. Another important limitation of the bandwidth lies in the distributed dynamics of the robot arms and joints. These dynamic modes are usually not considered in the model. For most industrial robots, these phenomena occur at frequencies well above the sensor/actuator bandwidth. Although they do not interfere with the low frequency motion controller, care is needed to avoid exciting them during impact.

3.3 - A Review of Existing Methods

Solutions to the overall control problem presented in section 3.1.2, applied to manipulator control with frictional unilateral contact are yet to be found. Generally, the global problem is divided in three subproblems, demanding each an independent solution. The controllers obtained this way have different structures or different control parameters, and the general solution is obtained by defining a switching law that evaluates the system's status and decides which control law to apply.

This section reviews the different methods proposed for position/force control of robot manipulators. The methods to control the free motion are not revised in detail here. They are well described in the literature. For contact motion control, essentially four distinct methods were proposed and they will be described in this section, following the analyses of Lewis [1]. In section 3.3.5, the notion of General Dynamical System (GDS) is introduced. This theory adapts the Lyapunov stability theory for systems with discontinuous components.

3.3.1 - The Stiffness Controller

The Stiffness Controller was introduced by Salisbury[29]. The fundamental idea is to use a virtual stiffness attached to the end effector to generate a virtual impedance force pulling the end effector toward the interior of the environment. If the contact is asymptotically stable, the force reaches a constant level at equilibrium. Consider the robot dynamics

$$M(q)\ddot{q} + N(\dot{q},q) + G(q) = \tau - \tau_f(\tau,\dot{q},q) + J_c^t f_e \tag{3.15}$$

described in chapter 2, and its operational space formulation,

$$M_x(x_e)\ddot{x}_e + N_x(\dot{x}_e,x_e) + G_x(x_e) = J^{-T}\left(\tau - \tau_f(\tau,\dot{q},q)\right) + f_e \tag{3.16}$$

The stiffness control law is given by

$$\tau = \hat{G}(q) + \hat{\tau}_f(\tau,\dot{q},q) + J^t\left(K_s\left({}^p x_{e_d} - {}^p x_e\right) - C_s\,{}^p\dot{x}_e\right) \tag{3.17}$$

where the ^ defines the estimate of the parameter. For simplicity, the environment is assumed fixed, passive, frictionless and composed of a flat wall

$$S_{env} = \left\{ x \mid x_{wall} - x = 0 \right\} \tag{3.18}$$

where x_{wall} is a given parameter, and for which the contact direction is given by

$$n = -\vec{i} \tag{3.19}$$

Further, the end effector is assumed spherical with the contact occurring on the spherical portion.

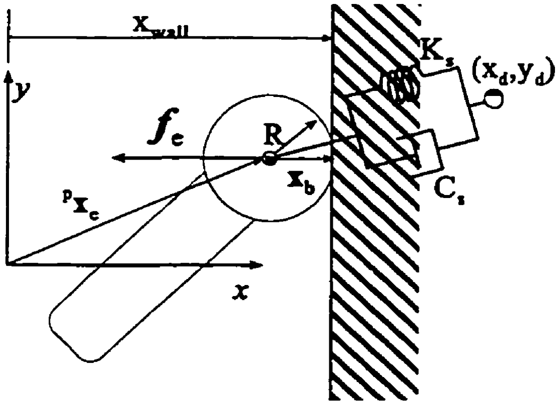


Figure 3.4 Stiffness Control

With these assumptions, the contact model becomes

$$f_e = \begin{cases} -K_e\left(x_{e_x} + R - x_{wall}\right)\vec{i} & if\ \left(x_{e_x} + R - x_{wall}\right) > 0 \\ 0 & if\ \left(x_{e_x} + R - x_{wall}\right) \le 0 \end{cases} \tag{3.20}$$

where R is the end effector radius and x_{ex} defines the x-component of the position vector defining the position of the end effector. A simple representation is shown in figure 3.4. The equilibrium of the dynamics is obtained by setting $\ddot{\mathbf{x}}=\dot{\mathbf{x}}=\mathbf{0}$ in equation 3.16 and is defined by the following conditions;

$$\begin{aligned} &y \to y_d \\ &f_e \to \frac{K_s K_{env}}{K_s + K_{env}}\left(x_d - x_{wall}\right) \approx K_s\left(x_d - x_{wall}\right) \end{aligned} \tag{3.21}$$

Consequently, if the system is stable and the environment is rigid, the contact force almost reaches a level determined by the control parameters K_s and x_d, and by the system parameter x_{wall}.

The stability of this equilibrium can be investigated using the second method of Lyapunov. For example, an appropriate Lyapunov function candidate, valid only if the contact is maintained, is given by

$$V=\frac{1}{2}\left[\dot{q}^{t}M(q)\dot{q} + \tilde{x}^{t}K_{s}\tilde{x} + \gamma^{t}K_{env}\gamma\right] \tag{3.22}$$

where $\tilde{\mathbf{x}}$ defines the error in position ($\tilde{\mathbf{x}} = \mathbf{x}_d - {}^{P}\mathbf{x}_e - \mathbf{R}\mathbf{n}$) and γ is the penetration vector ($\gamma = d\mathbf{n}$). By properly defining

$$N(\dot{q},q) = N^{*}(\dot{q},q)\dot{q} \tag{3.23}$$

one can use the skew-symmetry property of robot manipulators

$$\dot{q}^{T}\left[\dot{M}-2N^{*}(\dot{q},q)\right]\dot{q}=0 \tag{3.24}$$

and the Lyapunov function derivative for constant $\mathbf{x}_d$ becomes

$$\dot{V} = -\dot{q}^{t}J^{t}C_{s}J\dot{q} \tag{3.25}$$

Therefore, from Lasalle's invariance theorem, although $\dot{V}$ is negative semi-definite, $\dot{\mathbf{q}} \to 0$ and $\ddot{\mathbf{q}} \to 0$, and the system reaches the equilibrium defined in equation (3.21).

This proof of stability is based on the assumption that contact is always maintained. This extra condition implies that the variable d should be bounded by d < 0. Since the control law provides only partial linearization without inertial decoupling, this extra condition is difficult to ensure. Therefore, stiffness control is used most exclusively when the desired reference is a fixed point. Moreover, at equilibrium, the contact force is dependent on the contact rigidity. If the rigidity is high, this dependence is extremely weak. On the other hand, for soft environment, the resulting force is far from the desired one. This formulation introduces a direct compensation term for joint friction, but the effect of surface friction is not considered. In reality, friction is present and modifies the equilibrium and the dynamic behaviour. For example, as discussed in the appendix B, unilateral friction contact exhibits dynamic coupling between the normal and tangential motion. Finally, the structure of the stiffness controller does not allow us to consider any of the dynamics of the environment. In situations where this is important, the stiffness controller may fail.

3.3.2 - The Impedance Controller

Some observations made on Stiffness Control were considered by Hogan[28]. To generalize the approach, he suggested using a joint space impedance formulation for the control law. Following Lewis et al. [1], the control law is given by

$$\tau = \hat{M}J^{-1}\left(a - \dot{J}\dot{q}\right) + \hat{N}(\dot{q},q) + \hat{G}(q) - \hat{\tau}_f(\tau,\dot{q},q) - J^t\hat{f}_e \tag{3.26}$$

where a is the desired operational acceleration. Putting equation (3.26) into equation (3.15) and assuming that the estimates used in the control law are exact, one obtains

$$J\ddot{q} + \dot{J}\dot{q} = \ddot{x}_e = a \tag{3.27}$$

By defining the directions in which the position has to be controlled and the ones in which the force has to be controlled, one can define a manipulator frame transformation **T** such that

$$Ta = \begin{Bmatrix} a_p \\ a_f \end{Bmatrix} \quad and \quad Tf_e = \begin{Bmatrix} f_p \\ f_f \end{Bmatrix} \tag{3.28}$$

Therefore, by properly choosing the manipulator impedance $\mathbf{Z}_m$ (s) one obtains the impedance position control law

$$a_p(t) = \mathcal{L}^{-1}\left(\ddot{x}_{d_p}(s) - Z_{m_p}(s)^{-1}f_p(s) \right) \tag{3.29}$$

where the subscript *d* refers to the desired value, and the force control law

$$a_f(t) = \mathcal{L}^{-1}\left(Z_{m_f}(s)^{-1}\left(f_d(s) - f_f(s) \right) \right) \tag{3.30}$$

where $\mathcal{L}$ refers to the Laplace transform. The choice of impedance is governed by some rules. As stated by Lewis[1], one principle that can be used to define the necessary impedance type is the duality principle.

This formulation is an extension of the stiffness controller. It provides complete Cartesian linearization in joint space with Cartesian decoupling. The general definition of the manipulator impedance is such that defining a controller requires taking into account the type of environment dynamics. State dependent impedances can be included if needed. For example, Fraisse et al.[30] have included an extra virtual impedance between the manipulator and the environment to reduce the effect of unknown environment on a stiffness-like force controller. Many researchers have used this property to accommodate transition control, by defining a virtual resistive media in the neighbourhood of a contact surface to reduce kinetic energy before impact. Although it can be adapted to encompass the global control problem defined in section 3.1, it does not provide a unified solution. In fact, the idea of solving the general problem by providing an impedance gain look-up is a switching controller approach that may still be ill-behaved in some situations.

3.3.3 - The Hybrid Position/Force Control

The hybrid position/force controller was introduced by Raibert and Craig[16]. This approach has been the most successful one for providing good simultaneous control on force and position.

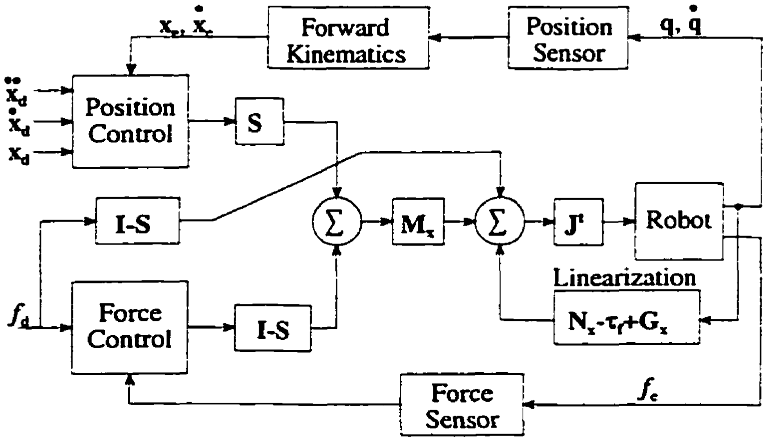


Figure 3.5 - Hybrid Position/Force Control

The structure of the hybrid position/force control is shown in figure 3.5. A matrix called *selection matrix S* is defined to describe, in a way similar to the impedance controller, the directions in which the position is controlled. The direction in which the force is to be controlled is given by **I-S**. Operational space position and force controllers are developed separately, and are joined so that the action of the position controller is in the position control direction and the one of the force controller is in the force control direction only. Decoupling of the two controllers is provided by a Cartesian decoupling scheme.

The similarity with the impedance controller presented in the previous chapter is stringent. In fact, it is only different at the implementation stage. While for the impedance controller, the decoupling between the position and force directions is made before the impedance is applied, for hybrid position/force control, the decoupling is done in a step following the position and force control laws. For impact control, however, the hybrid structure does not permit the use of the force control direction to dissipate kinetic energy before impact using some virtual fluid. To account for transition from free to contact motion, the impact can be limited by appropriate path planning (reduced velocity at contact). When contact is made, the selection matrix is set to account for the new control objectives. The discontinuity then appears in the selection matrix and not in the control gains as for the impedance controller.

As shown by Wen and Murphy[19], the hybrid control scheme is sensitive to modelling errors. For example, a coupled friction model in which the tangential surface friction depends on the contact force is extremely difficult to model properly. In practice, normal and tangential motions are coupled through surface friction (see [60-64]). In such cases, linearization is easily lost and stability is not guaranteed.

3.3.4 - The Reduced-Order Controller

The reduced order control has been developed recently. It was introduced by McClamroch et al.[21,22,23] and further developed by Mills[24] and Tabhoud[25]. It is an extreme generalization of the controllers presented in sections 3.3.2 and 3.3.3. In the reduced order controller, the contact is modelled as a holonomic omnilateral constraint whose orthogonal complement defines the unconstrained dynamics (dynamics in the direction of position control). This formulation generalises the notion of force and position controlled directions such that they can be defined in any coordinate system, not only in the operational space. The subsystem defined by the reduced order dynamics, representing the unconstrained motion, is controlled for position while the subsystem defined by the algebraic relation giving the constraint force is controlled for force.

Using the formalism described in the appendix A, one can define the contact surface in joint coordinates[2]

$$\phi_{env}(q) = 0 \tag{3.31}$$

such that the constraint gradient Ω is defined by

$$\left[\nabla_q \phi\right]\dot{q} = \Omega\dot{q} = 0 \quad and \quad \left[\nabla_q \phi\right]\ddot{q} + \left(\frac{d}{dt}\left[\nabla_q \phi\right]\right)\dot{q} = \Omega\ddot{q} + \dot{\Omega}\dot{q} = 0 \tag{3.32}$$

Using the Lagrange multiplier theorem (see appendix A) and equation (3.15), the descriptor form of the constrained dynamics is given by

[2] *This function can be learned a priori through a teaching mode, by moving the manipulator on the contact surface.*

$$\begin{bmatrix} M & \Omega'(q) \\ \Omega(q) & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \tau + N(q,\dot{q}) - G(q) + \tau_f \\ -\dot{\Omega}\dot{q} \end{Bmatrix} \tag{3.33}$$

A solution to this set of differential-algebraic equations can be found using the orthogonal complement approach. The orthogonal complement of the constraint gradient over the free motion space is given by

$$C = I - \Omega'(\Omega\Omega')^{-1}\Omega \tag{3.34}$$

for which only n-1 columns are independent. Redefining C to take into account only the independent columns, a set of reduced coordinate vectors $\mathbf{q}_r$ exists such that

$$C\dot{q}_r = \dot{q} \tag{3.35}$$

***Theorem 3.1** - Corollary to the inverse function theorem*

If the nxn-1 matrix **C** is full row rank at $\mathbf{q}_0$, then there exists a mapping χ:$\mathbb{R}^{n-1} \rightarrow \mathbb{R}^n$ and its inverse χ^{-1} such that $\mathbf{q}=\chi(\mathbf{q}_r)$ and $\mathbf{q}_r=\chi^{-1}(\mathbf{q})$, valid in a neighbourhood N of $\mathbf{q}_0$.

Therefore, the motion is described by the reduced order dynamics obtained by pre-multiplying equation (3.15) by $\mathbf{C}^t$ and by using theorem 3.1 to give

$$C^tMC\ \ddot{q}_r + C^t\big(N(\dot{q}_r,q_r) + G(q_r) - \tau_f\big) = C^t\tau \tag{3.36}$$

and the constraint contact force λ is obtained by using the result of equation (3.36) into equation (3.15) using the theorem 3.1, and has the form of an algebraic relationship

$$\lambda = \lambda\big(\dot{q}_r, q_r, \tau\big) \tag{3.37}$$

Using this reduced state model, a control law of the form can be used

$$\tau = MC\ \Big[\ddot{q}_{r_d} + K_v\big(\dot{q}_{r_d} - \dot{q}_r\big) + K_p\big(q_{r_d} - q_r\big) + \hat{N} + \hat{G} - \hat{\tau}_f\Big] + \Omega'\big[\lambda_d + K_f\ \big(\lambda_d - \lambda\big)\big] \tag{3.38}$$

where $\mathbf{K}_v$, $\mathbf{K}_p$ and $\mathbf{K}_f$ are control gain. The reduced order closed-loop dynamics is given by

$$C^tMC\ \Big[\big(\ddot{q}_{r_d} - \ddot{q}_r\big) + K_v\big(\dot{q}_{r_d} - \dot{q}_r\big) + K_p\big(q_{r_d} - q_r\big)\Big] = 0 \tag{3.39}$$

and the overall dynamics is given by

$$MC\ \Big[\big(\ddot{q}_{r_d} - \ddot{q}_r\big) + K_v\big(\dot{q}_{r_d} - \dot{q}_r\big) + K_p\big(q_{r_d} - q_r\big)\Big] + \Omega'\ \Big[\big(\lambda_d - \lambda\big)\big(I + K_f\big)\Big] = 0 \tag{3.40}$$

From equation (3.39), the reduced order closed-loop dynamics can be made stable by properly selecting the gains $\mathbf{K}_v$ and $\mathbf{K}_p$, such that the overall closed-loop dynamics insures that $\boldsymbol{\lambda} \rightarrow \boldsymbol{\lambda}_d$.

The method is fully oriented in solving the problem of contact motion control. The approach is appealing, general and formal. Due to its structure, the control law makes it difficult to control the transition. To assume the contact as a holonomic constraint, the contact should be very rigid. If it is soft or if the environment has some flexible dynamics, the definition of the orthogonal complement becomes extremely involved. While this is also true for the impedance controller and the hybrid control scheme, these are easier to deal with since they provide a control force in operational space. This method is very restrictive due to its highly rigid frictionless contact model being a fundamental part of the control scheme. This is clear from the assumptions made to develop the method (see [21]). One of these assumptions states that the conditions to meet theorem 3.1 are valid. Those conditions, however, are not met if contact is lost momentarily.

3.3.5 - Generalized Dynamical System

None of the approaches presented in this section are general solutions for the problem stated in section 3.1. They all require the implementation of a switching law to select the controller to apply during transition to or from contact. The difficulties associated with such controllers possessing variable structures have only been partially discussed. Implementation of switching control laws often demands a particular precaution to ensure smooth transition between controller structures. The study of stability of such switching control laws has been an important subject of research, especially in the Russian literature. In practice, it is well known that if the switching manifold separating different structures is attractive from all structures, the neglected high order dynamics induces a chattering motion that is difficult to control.

The theory of Generalized Dynamical Systems (GDS) was introduced by Roxin[10,11] to study the stability of systems modelled by differential equations with discontinuous right-hand sides. The concept was derived from the study of Fillipov [65]. GDS was applied to study the stability of discontinuous control laws for robot manipulator impact control by Mills [7]. The main idea behind GDS is the description of the evolution of a dynamical system using an attainability function instead of a state space model. The theory of GDS is summarized here following Gutmann[9]. The proofs of the theorems are not given here (see Gutmann[9]).

Considering a dynamical system defined by equation (3.12) and a control law defined by

$$u = u(x_f, t) \tag{3.41}$$

the following definitions can be made regarding GDS.

***Definition 3.6** - Attainability Function*

The attainability function $F()$ is the function defined by

$$x_f(t) \in F(x_f(t_0), t_0, t) \tag{3.42}$$

and represents the set of states x_f attainable at time t from the given initial conditions $\{x_f(t_0), t_0\}$.

***Definition 3.7** - Contingent Equation*

The contingent equation $E(x_f, t)$ is defined by

$$D^* x_f(t) \in E(x_f, t) \tag{3.43}$$

and represents the set of all possible evolutions of the state trajectory. $D^*x_f(t)$ is the contingent derivative of x_f and represents the evolution of the state.

***Definition 3.8** - Generalized Dynamical System*

When the attainability function is submitted to the following conditions

i $F(x_f(t_0), t_0, t)$ *is a closed non-empty subset of* $\mathbb{R}^n$,

ii $F(x_f(t_0), t_0, t_0) = x_f(t_0)$,

iii $F(x_f(t_0), t_0, t)$ *is a semigroup:* $F(x_f(0), t_0, t_2) = \bigcup F(x_f(t_1), t_1, t_2)$ *where* $x(t_1) \in F(x_f(t_0), t_0, t_1)$,

iv *Given* $x(t_1)$, t_1 *and* $t_0 \leq t_1$, *there exists* $x(t_0)$ *such that* $x(t_1) \in F(x_f(t_0), t_0, t)$,

v $F(x_f(t_0), t_0, t)$ *is* c^0 *(continuous in t),*

vi $F(x_f(t_0), t_0, t)$ *is upper semi-continuous (USC) at* $(x_f(t_0), t_0)$ *and uniformly USC in any interval* $[t_1, t_2]$,

then the attainability function defines a Generalized Dynamical System

***Definition 3.9** - Motion and Trajectory*

A mapping $\Theta: [t_0, t_1] \rightarrow \mathbb{R}^n$ such that $t_0 \leq \tau_0 \leq \tau_1 \leq t_1$ implies $\Theta(\tau_1) \in F(x_f(\tau_0), \tau_0, \tau_1)$ is called a motion. The curve it defines in $\mathbb{R}^n$ is called the trajectory.

***Theorem 3.2** - Local Motion Existence*

If $E(x_f(t),t)$ is compact, convex and upper semi-continuous in a δ-neighbourhood N_δ of $(x_f(t_0), t_0)$, then there exists a motion $\Theta(t)$, a solution of the contingent equation $D^* x_f \in E(x_f, t)$ passing through $(x_f(t_0), t_0)$ and continuing until the boundary of N_δ.

***Theorem 3.3** - Generalized Dynamical System*

If a motion locally exists on N_δ and if escape from N_δ is impossible ($\nexists t \in [t_0, t_1]$, s.t. $\Theta(t) \notin N_\delta$), then $F(x_f(t_0), t_0, t) \in \mathbb{R}^n$ described by

$$x_f(t_1) \in F(x_f(t_0), t_0, t) \Leftrightarrow \exists \Theta(t_1) = x_f(t_1) \text{ and } D^* x_f \in E(x_f, t) \tag{3.44}$$

such that $x_f(t_o) = \Theta(t_0)$ and $x_f(t_1) = \Theta(t_1)$, defines a Generalized Dynamical System.

The compactness condition ensures that the state derivative is bounded, the convexity condition implies the existence of the state derivative along any trajectory in the neighbourhood of a point and the upper semi-continuity condition ensures finite step size in the state derivatives. These definitions and theorems represent the essential of the GDS theory. In the GDS theory, the motion resulting from a dynamic system is investigated from the point of view of the state trajectory. The basic assumption of GDS is that although the contingent equation (the system's dynamics) is discontinuous, the trajectory is continuous in time. This generalization of a dynamical system enables the extension of the Lyapunov stability theory to include systems with discontinuous dynamics.

***Definition 3.10** - Strong Stability*

An equilibrium point x_0 is Strongly (Lyapunov) Stable (SLS) if for all $\epsilon>0$ and $t_0 \geq 0$, there exists $\delta(\epsilon,t)>0$ such that for all $\|x\|<\delta$ and $t \geq t_0$, $F(x(t_0), t_0, t) \in \{x \in \mathbb{R}^n \mid \|x\|<\epsilon\}$.

***Definition 3.11** - Uniform Strong Stability*

An equilibrium point x_0 is Uniformly Strongly (Lyapunov) Stable (USLS) if it is SLS and if $\delta(x,t)=\delta(x)$ for all $t \geq t_0$.

***Definition 3.12** - Uniform Asymptotic Strong Stability*

An equilibrium point x_0 is Uniformly Asymptotically Strongly (Lyapunov) Stable (UASLS) if it is USLS, and if there exists $\eta>0$ and a function $\tau(\epsilon)$ defined for sufficiently small ϵ such that if $\|x_0\| \leq \eta$

and $t_0 \geq 0$, then for any motion $\Theta(t)$such that $\Theta(t_0)=x_0$, $\|\Theta(t)\| < \epsilon$ hold for all $t \geq t_0 + \tau(\epsilon)$.

These definitions of Lyapunov stability are standard, but have been presented using the notation of GDS.

***Definition 3.13** - Derivative of a Function for GDS*

Consider a function $V(x,t)$, then the upper derivative of V is defined by

$$D^{+}V(x,t) = \lim_{\tau \to t^{+}} \left\{ \frac{V(y,\tau) - V(x,t)}{\tau - t} \ , \ y \in F(x,t,\tau) \right\} \tag{3.45}$$

the lower derivative is

$$D^{-}V(x,t) = \lim_{\tau \to t^{-}} \left\{ \frac{V(y,\tau) - V(x,t)}{\tau - t} \ , \ y \in F(x,t,\tau) \right\} \tag{3.46}$$

and the centre derivative is

$$D^{0}V(x,t) = \max\left(D^{+}V, D^{-}V\right) \tag{3.47}$$

Finally, the stability theorem adapts the Lyapunov theory of stability to include GDS.

***Theorem 3.4** - GDS Stability*

Let $V(x,t)$ be a positive function $V:\mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}_+$. If $V(x,t)$ and $-D^0V(x,t)$ are positive definite, and if $V(x,t) \to 0$ uniformly in $t \geq 0$, then $x=0$ is uniformly asymptotically strongly Lyapunov stable (UASLS).

According to the GDS theory, the compactness, the convexity and the upper semi-continuity of the state dynamics are a sufficient conditions the study of discontinuous systems using Lyapunov stability theory. The concept of GDS was used by Mills[7] to verify the stability of a switching controller to control transitions to and from contact conditions in robot manipulators. The theory is an attractive formal approach to verify stability, but its validity depends greatly on the model used.

Chapter IV

Model Predictive Control

The methods presented in the preceding chapter are not unified solutions to the control problem formulated in section 3.1. Each represents a particular solution obtained for specific conditions. Such a unified solution is yet to be found and the field of robot control fails in providing the tools necessary to obtain it. As a starting point, it is quite natural to look at how humans perform these tasks. The human psychomotor system is not relying on power transfer computation and simple reaction to senses, as in most control synthesis methods. Its approach is based on the knowledge of the effects of actual actions on future results. For humans, this knowledge is learned with time, stored in memory and used when needed. The mechanism describing how it is learned and stored is still to be fully discovered, but many models have been proposed. The way it is used to predict future events is even more complex, relying on more contextual issues such as culture than the storage mechanism. In control systems theory, the system's knowledge has typically the form of deterministic mathematical models. In recent years, neural-network (one of the brain knowledge models) was also used. In both cases, the idea is to *acquire and keep* knowledge about the behaviour of the entity that has to be controlled. The way models are used in control systems depends on the control method considered. Models are always used in the synthesis stage of the design process, but they can also be an integral part of the control system structure as in model-based control methods.

In this chapter, model-based predictive control (MPC) is presented. This intuitive control approach imitates one aspect of the human complex control system. With MPC, the model of the system is used to predict the system's response over some fixed receding time horizon expanding in the future. The result of the prediction is used to adjust the actual control actions to meet the control objectives over the whole horizon. This approach can easily be exemplified by a car driver. When obstacles appear in the way of a vehicle, drivers use their judgement to predict if an impact will occur. If an impact is predicted, most drivers will slow at a rate proportional to the prediction confidence, up to the limit of the physical system. Better drivers, who have more experience and skills, can make better predictions and react more quickly to a dangerous situation. Poor drivers with little driving experience may have difficulty to avoid impact. This level of knowledge is reflected in MPC by the accuracy of the model and the quality of the sensor and actuator devices. With MPC, the control problem formulated in section 3.1 can be resolved in a unified framework, provided a good model of the dynamic system to control.

In this chapter, the concepts behind MPC are introduced through a general continuous-time formulation. The application to linear time invariant (LTI) systems is used to illustrate how it can be used in typical control applications. The formulation is then extended to include nonlinear systems and several alternatives for solving the nonlinear MPC control problem are reviewed. Various issues related to design and implementation of the nonlinear MPC approach are also discussed.

4.1 - General Continuous-Time Formulation

Typically, controlled systems are represented by general state space models of the form

$$\begin{aligned} \dot{x}_s &= f(x_s,u,\epsilon) \\ y &= h(x_s,u,\nu) \end{aligned} \tag{4.1}$$

where x_s is the state vector, **u** is the control vector (or command), ϵ is the state disturbance vector, **y** is the output vector and υ is the measurement noise vector. The Model Predictive Control MPC design problem is formulated as an optimal control problem described by

$$u^* = \underset{u}{argmin}\ \Phi(y,u) \tag{4.2}$$

where Φ is the penalizing (or cost) functional

$$\Phi(y,u) = \int_{t}^{t+T_h} \left[(y-r)'Q_1(y-r) + u'Q_2u\right]d\tau \tag{4.3}$$

where T_h is the horizon, **r** is the reference trajectory, Q_1 and Q_2 are weighting matrices. Sometimes, the complexity of the optimization problem can be reduced by imposing the constraint

$$\dot{u}(\tau) = 0 \qquad if\ \tau>t+T_u \tag{4.4}$$

where $T_u \leq T_h$ is called the control horizon. In this formulation, bounds on control commands and measurements, as well as a desired final manifold, can be accounted for by including the following additional constraints

$$\boldsymbol{y} \in \boldsymbol{Y} \quad \textit{and} \quad \boldsymbol{u} \in \boldsymbol{U} \quad \textit{and} \quad \Lambda(\boldsymbol{y}_{t+T_h}) = \boldsymbol{0} \tag{4.5}$$

This formulation strongly resembles the finite-time optimal control problem. The major difference lies in the use of a receding horizon for the MPC problem. The feedback mechanism appears in the use of the measurements to provide the initial conditions necessary for the prediction. The model is not restricted to any particular structure, but should at least provide an existing solution to insure the existence of a solution to the constrained optimization problem. This generality makes MPC an attractive control scheme to solve problems where a variable topology dynamic system has to be controlled.

The predictive nature of MPC is indeed a powerful property for the control of discontinuous systems. Historically, very conservative static feedback laws were used to ensure proper transitions between various structures of a variable structure system. A good example is the virtual fluid introduced in the impedance control scheme to handle contact transition when controlling robot manipulator arms. While the virtual fluid represents a static reaction obtained from a sensed prediction of an impact, the model-based prediction of MPC also includes the local dynamics of the robot manipulator. When the model used in MPC contains a good representation of the transition to occur, the resulting dynamic control action can account for control objectives before, during and after the discontinuity has occurred. Consequently, MPC has the potential to provide a unified control scheme for dynamic systems with variable structures.

4.2 - Discrete-time LTI/SISO Formulation

MPC has mainly been formulated for discrete-time LTI SISO systems. In its original formulation, the inequality constraints resulting from the application of equation (4.5) are not considered. Most of the MPC methods formulated over the years differ only by the type of model used to represent the system or the assumptions made about the conditions of optimization (see [66-90]). In the presentation that follows, the unified predictive control formulation of Soeterboek[68] is used.

4.2.1 - Discrete-time LTI Model

In general terms, a SISO discrete-time system can be modelled by

$$y_k = \frac{q^{-d}B(q)}{A(q)} u_{k-1} + \frac{C(q)}{D(q)} e_k \tag{4.6}$$

where y_k is the plant output at time k, q is the shift operator, u_{k-1} is the plant input at time k-1, e_k is the plant disturbance at time k, d is the pure delay of the system and A(q), B(q), C(q) and D(q) are polynomials in descending order of q. Depending on the form of the polynomials, the model takes on a different format (see table 4.1).

Model Name	A	B	C	D
FIR	1	open	1	Δ
IIR	open	open	1	Δ
FSR	1	$B^{*}\Delta$	1	Δ
ARX	open	open	1	A
ARIX	open	open	1	$A\Delta$
ARMAX	open	open	open	A
ARIMAX	open	open	open	$A\Delta$

Table 4.1 - Discrete-Time LTI Models

In table 4.1, B^{*} is any polynomial in q^{-1} and the operator Δ is called the difference operator($\Delta \triangleq 1 - q^{-1}$). The disturbance e_k may be either deterministic or stochastic. For linear systems, the knowledge of the model, the actual and past values of y, the future values of the plant input and disturbance are a sufficient condition for predicting the future values of the output y. When the disturbance is deterministic, it may be predicted directly. For a stochastic disturbance that may be derived from a white noise the Minimum Variance (MV) principle can be used to derive a MV i-step-ahead predictor of the form(see Soeterboek [68])

$$\hat{y}_{k+i} = G_i \, u_{k+i-d-1} + \frac{H_i}{\hat{A}} u_{k-1} + \frac{F_i}{\hat{C}} \left(y_k - \hat{y}_k\right) \tag{4.7}$$

where ^ defines the estimate, G_i and H_i are obtained from the solution of the Diophantine equation

$$\frac{\hat{B}}{\hat{A}} = G_i + q^{-i-d}\frac{H_i}{\hat{A}} \tag{4.8}$$

and where F_i is obtained by the solution of the Diophantine equation

$$\frac{T}{\hat{D}} = E_i + q^{-i}\frac{F_i}{\hat{D}} \tag{4.9}$$

The notation ^ is used on the polynomials to show that these estimates can be updated at each step in a self-tuning adaptive control scheme. In any case, the model of equation (4.7) is sufficient for predicting future values of the output variable, and to construct a prediction model of the form

$$\hat{\boldsymbol{y}} = \boldsymbol{Gu} + \boldsymbol{Hu}^{*} + \boldsymbol{Fc} \tag{4.10}$$

where

$$\begin{aligned}
y &= [\hat{y}_{k+d+1}, \dots, \hat{y}_{k+d+hp}]^T \\
\mathbf{u} &= [u_k, \dots, u_{k+hp-d-1}]^T \\
\mathbf{u}^{*} &= [u^{*}_{k-1}, u^{*}_{k-2}, \dots]^T \\
\mathbf{c} &= [c_k, c_{k-1}, \dots]^T \\
\hat{A}(q)\, u^{*}_k &= u_k \\
\hat{C}(q)\, c_k &= y_k - \hat{y}_k
\end{aligned}$$

and where h_p defines the prediction horizon in sample steps.

4.2.2 - Optimization Criterion

The penalizing functional as formulated in equation (4.2) can take the form of

$$\Phi(\boldsymbol{y},\boldsymbol{u}) = \sum_{i=k+d}^{k+d+hp}\left[\alpha_i(y_i - r_i)^2 + \gamma_i u_{i-d}^2\right] = [\boldsymbol{y} - \boldsymbol{r}]^T \boldsymbol{\alpha}\, [\boldsymbol{y} - \boldsymbol{r}] + \boldsymbol{u}^T \gamma\, \boldsymbol{u} \tag{4.11}$$

where α_i and γ_i are weighting factors, and r_i is the reference at the step i. The optimization is constrained by

$$\Delta u_i = 0 \qquad \forall i \in [h_c, h_p - d] \tag{4.12}$$

where h_c is called the control horizon. When the control horizon is zero, but γ is not zero, the MPC controller obtained is called the mean-level MPC controller. The choice of plant and disturbance models, weighting factors and horizons defines the type of controller used.

In MPC, it is a common practice to solve the optimal control problem over the entire horizon, and to apply at a given time only the first value of the resulting control sequence. This process is repeated at every sampling step. The solution of the optimal control problem defined by equations (4.2, 4.5, 4.10, 4.11, 4.12) differs depending on the type of MPC controller chosen. For the general case of nonadaptive unbounded MPC, the solution to the problem is analytical and a control law is obtained by solving a standard Ricatti problem once off-line. The resulting control law has the general form of an output polynomial feedback controller(see Figure 4.1)

$$R(q^{-1})\ \Delta u_k = T(q) r_k - S(q^{-1}) y_k \tag{4.13}$$

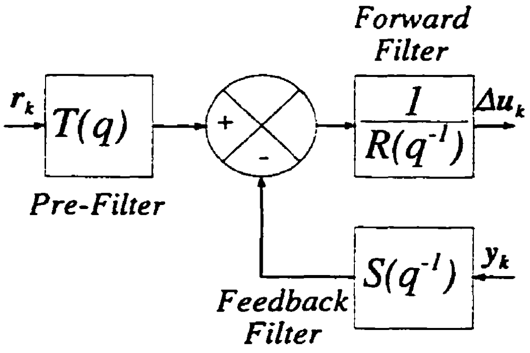


Figure 4.1 - Linear MPC Controller

In adaptive formulations such as the self-tuning Generalized Predictive Control (GPC), the solution is still analytical but solved on line at every sampling step. When bounds on input and output are introduced, the solution requires a real-time numerical solution of the optimization problem. In practice, real-time solutions with bounds on control are obtained using mathematical programming algorithms.

A formulation using state-space models is straightforward. In fact, they lead to easy formulation of the MPC problem for MIMO systems. This is shown in the following example.

Example 4.1 - Mean -level MPC of a Flexible Shaft Servo (See Necsulescu and de Carufel [83])

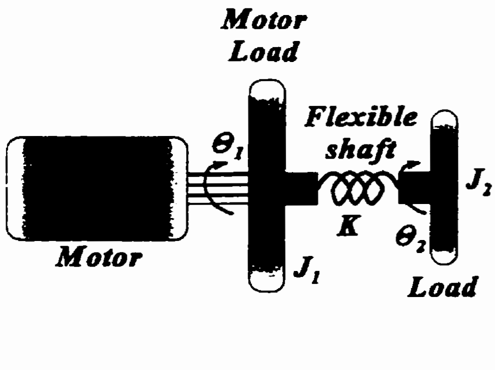


Figure 4.2 - Flexible Shaft Servo

Consider the system represented in figure 4.2. It consists of a servomotor coupled to an external load through a flexible shaft. A dynamic model of the system using a one-element finite element model of the flexible shaft is

$$\begin{bmatrix} J_1+\frac{J}{3} & \frac{J}{6} \\ \frac{J}{6} & J_2+\frac{J}{6} \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} + \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \begin{Bmatrix} \theta_1 \\ \theta_2 \end{Bmatrix} = \begin{Bmatrix} \tau_{motor} \\ 0 \end{Bmatrix} \qquad (4.14)$$

where J_1 and J_2 are the inertia of the motor and the load respectively, J is the inertia of the element, K is the shaft stiffness and τ_{motor} is the motor torque. The following state model can therefore be obtained

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & \boldsymbol{0} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1} \end{bmatrix} \tau_{motor} \qquad (4.15)$$

*where $\boldsymbol{x}=\{\theta_1, \theta_2, \dot\theta_1, \dot\theta_2\}^T$, **I** is the identity matrix, **M** and **K** are respectively the mass and stiffness matrices defined in eqn (4.14) and τ_{motor} is the motor torque. This continuous-time model can be discretized in a standard way, leading to a discrete-time model of the form*

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\, \boldsymbol{x}_k + \boldsymbol{B}_d\, \tau_{motor} \qquad (4.16)$$

where $\boldsymbol{A}_d$ is the state transition matrix and $\boldsymbol{B}_d$ is the discrete-time input matrix. If only the angle on the load side is measured, the output equation is given by

$$y_k = \theta_2 = \boldsymbol{C}\, \boldsymbol{x}_k = \{0\ 1\ 0\ 0\}\, \boldsymbol{x}_k \qquad (4.17)$$

A prediction model of the form of eqn (4.10) can be derived

$$\boldsymbol{y} = \boldsymbol{C} \begin{bmatrix} \boldsymbol{A}_d \\ \boldsymbol{A}_d^2 \\ \vdots \\ \boldsymbol{A}_d^{h_p} \end{bmatrix} \boldsymbol{x}_k + \boldsymbol{C} \begin{bmatrix} \boldsymbol{B}_d & 0 & \cdots & 0 \\ \boldsymbol{A}_d\boldsymbol{B}_d & \boldsymbol{B}_d & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \boldsymbol{A}_d^{h_p-1}\boldsymbol{B}_d & \cdots & \cdots & \boldsymbol{B}_d \end{bmatrix} \boldsymbol{u} = \boldsymbol{F}\, \boldsymbol{x}_k + \boldsymbol{G}\, \boldsymbol{u} \qquad (4.18)$$

where $\boldsymbol{y}=\{y_{k+1}, \ldots, y_{k+hp}\}$ and $\boldsymbol{u}=\{\tau_{motor,k}, \ldots, \tau_{motor,khp-1}\}$. Introducing the prediction function (4.18) in *the penalizing functional (4.11) leads to*

$$\Phi = \alpha\left[\boldsymbol{F x_k} + \boldsymbol{G u} - \boldsymbol{r}\right]^T\left[\boldsymbol{F x_k} + \boldsymbol{G u} - \boldsymbol{r}\right] + \gamma \boldsymbol{u}^T\boldsymbol{u} \qquad (4.19)$$

Using a standard procedure, the optimal solution is given by

$$\frac{\partial\Phi}{\partial\boldsymbol{u}} = 0 = 2\left[\frac{\gamma}{\alpha}\boldsymbol{I} + \boldsymbol{G}^T\boldsymbol{G}\right]\boldsymbol{u} - 2\boldsymbol{G}^T\left[\boldsymbol{r} - \boldsymbol{F x_k}\right] \qquad (4.20)$$

such that the optimal control output sequence is given by

$$\boldsymbol{u} = \left[\frac{\gamma}{\alpha}\boldsymbol{I} + \boldsymbol{G}^T\boldsymbol{G}\right]^{-1}\boldsymbol{G}^T\left[\boldsymbol{r} - \boldsymbol{F x_k}\right] \qquad (4.21)$$

By extracting from eqn (4.21) the expression for the first value of $\tau_{motor,k}$, one obtains a control law of the form of equation (4.13). An easier way to solve the problem is to assume τ_{motor} constant over the horizon. Then, for the predictor (4.18), the G matrix becomes a vector defined by (the convention of lowercase bold letters for vectors is not followed here to avoid confusion)

$$\boldsymbol{G} = \boldsymbol{C}\left\{\begin{matrix} \boldsymbol{I} \\ \boldsymbol{A_d}+\boldsymbol{I} \\ \vdots \\ \boldsymbol{A_d}^{h_p-1}+\boldsymbol{A_d}^{h_p-2}+\dots+\boldsymbol{I} \end{matrix}\right\}\boldsymbol{B_d} \qquad (4.22)$$

*and then the vector **u** in eqn (4.21) can be replaced by the scalar $\tau_{motor,k}$ to give the control law*

$$\tau_{motor,k} = \left[\frac{\gamma}{\alpha}\boldsymbol{I} + \boldsymbol{G}^T\boldsymbol{G}\right]^{-1}\boldsymbol{G}^T\boldsymbol{r} - \left[\frac{\gamma}{\alpha}\boldsymbol{I} + \boldsymbol{G}^T\boldsymbol{G}\right]^{-1}\boldsymbol{G}^T\boldsymbol{F x_k} \qquad (4.23)$$

similar to the control law of eqn (4.13). The control law possesses a state feedback term and a feedforward term that considers future changes in reference.

The parameters of the simulated system are J_1=0.565x10^{-3} kgm^2, J_2=2.88x10^{-3} kgm^2 and K=700 Nm/rad. The inertia of the element is J=0.0036x10^{-3} kgm^2. The control law is evaluated for α=1, γ=2x10^{-4}, h_p=250 steps and implemented at a sampling rate of 10 kHz. The servomotor is requested to perform a sequence of steps form 0 to 1.57 radian (90 °) at .2 seconds from initial time. The results are shown in Fig. 4.3 and 4.4. The effect of prediction is visible by the motion being initiated

when the step reference is in the prediction horizon. Note also that the response obtained for this case is nearly a deadbeat controller. Other sets of parameters will give different transient responses.

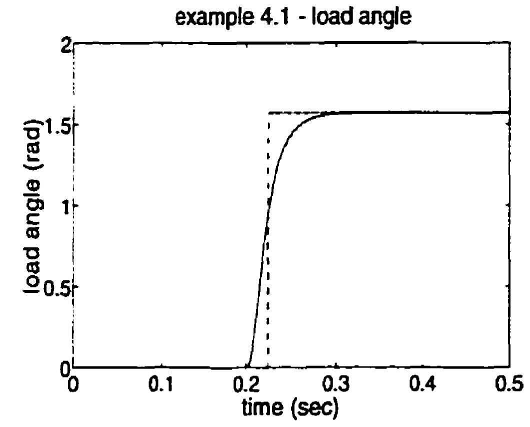


Figure 4.3 - Load Angle

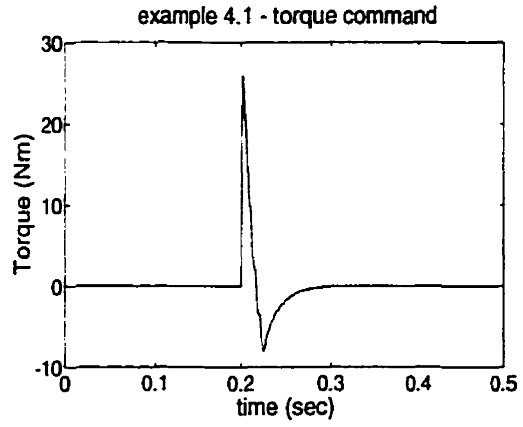


Figure 4.4 - Control Torque

4.2.3 - Stability

The stability conditions for finite time optimal controllers have been known since the late 1970's (Kwon and Pearson[91]). However, the link with MPC has been made only recently (De Nicolao and Scattolini [87]). In summary, the conditions of stability for discrete-time LTI implementation of Constrained Receding Horizon Predictive Control (CRHPC), are given by the following theorems, stated here without proof.

Consider a FSR model of a system of order n (see eqn 4.6), submitted to a control horizon constraint (equation (4.12)) and constrained by

$$y_{k+d+h_p+i} = r_{k+d+h_p+i} \qquad \forall i \in \{1,2,..,m\} \tag{4.24}$$

where m>0 is an extra design factor.

__Theorem 4.1 - CRHPC with Same Prediction and Control Horizon__

If $h_p=h_c>n+1$ and $m=n+1$, then the close-loop system given by the system (eqn 4.6) and the control *law (equation 4.13) is asymptotically stable.*

Theorem 4.1 says that if the horizon is long enough to take in consideration all the influence of the dynamics on future values of the output for a given input, and if the predicted output at the end of the prediction horizon is equal to the reference, then the solution of the optimization should insure that the system is asymptotically stable.

***Theorem 4.2** - Deadbeat Control*

If h_c=n and m=n+1, then the control law obtained is a stable deadbeat controller for the system defined by equation 4.6.

Theorem 4.2 illustrates the link between predictive control and deadbeat controllers. When the control prediction horizon is the order of the system and it is required that the error be zero at the end of the control horizon (eqn 4.24), the deadbeat control conditions are obviously obtained.

***Theorem 4.3** - CRHPC Using Only Output Prediction*

If the system under control is asymptotically stable, $\alpha=0$, m=1 and there exists v such that either

$$s_v \le s_{v-1} \le \ldots \le s_\infty, \quad s_v > \frac{s_\infty}{2}, \quad s_\infty > 0 \qquad \textbf{(4.25)}$$

or

$$s_v \le s_{v+1} \le \ldots \le s_\infty, \quad s_v > \frac{s_\infty}{2}, \quad s_\infty < 0 \qquad \textbf{(4.26)}$$

where s_i is the i^{th} factor of the polynomial G_{hp} resulting from solving the Diophantine equation (eqn 4.8) to define the h_p-step-ahead predictor (equation 4.7), then for $h_p=h_c>v-1$, or for $h_c=0$ and $h_p>v+1$, the close-loop system obtained by applying the control law (eqn 4.13) to the system (equation 4.6), is asymptotically stable.

Theorem 4.3 is directly derived from the Jury criterion applied to the close loop system. It stipulates the conditions that should exist between the coefficient of the system's prediction model (eqn 4.7) and the horizon to verify the Jury criterion.

The analysis of predictive control stability presented here applies only to a small portion of the MPC implementations reported. In practice, the condition defined by equation (4.24) is rarely considered in the optimization process. Moreover, input and output constraints are often included and are not considered in this analysis. Although a rigorous proof of stability for the general linear case is not yet available, MPC has been used successfully in the process industry. An intuitive discussion about the stability of predictive control was presented in Robinson and Clarke[81]. They argued that a good rule-of-thumb for the stability of linear MPC controllers is that the prediction time includes at least one complete cycle of all the dynamics to be controlled. In the discrete-time domain, this intuitive condition is equivalent to stating that the prediction horizon (in sample steps) is at least greater than the ratio of the time constant of the slowest dynamics over the sampling time. This condition has been found practically sufficient.

4.3 - Nonlinear MPC - A Continuous-time Approach

MPC for discrete-time LTI systems is straight forward, but the advantages of prediction for linear systems are not obvious. It is well known that for LTI systems, the knowledge of the actual state and future commands is sufficient for defining any future states. When the command is given by a linear static state feedback law, only the actual states are necessary. Therefore, the advantage of prediction limits itself to its ability to deal with pure delays (Schmidt predictor) and rapid variations in the reference. Except these specific properties, MPC for LTI systems represents just another method to design polynomial controllers.

Studies on the extension of MPC to include nonlinear systems are limited in number. To apply MPC to the problem formulated in chapter 3, its application to nonlinear systems has to be investigated. The following section discusses the different approaches that can be used to solve the nonlinear MPC problem. The discussion concentrates most exclusively on the continuous-time problem, and the digital implementation is obtained by discretizing the final continuous-time solution. In brief, four different approaches to solve the nonlinear MPC problem exist: the variational formulation, the extended MPC, feedback linearization and mathematical programming. Each of these approaches is discussed in this section.

4.3.1 - The Variational Formulation

This formulation is a standard and rigorous procedure to solve almost any dynamically constrained optimization problem and uses the calculus of variations. The presentation that follows can be found in Lewis[92] and concentrates exclusively on the solution of the unbounded optimal control problem.

Considering a dynamic system given by equation (4.1), for a motion from the initial conditions $x_s(0)=x_0$ to the final manifold $\Psi(x_s(t_f), t_f) = 0$, the optimal solution

$$x^*(t) = \underset{x_f(t)}{argmin}\ \Phi_c \quad and \quad u^*(t) = \underset{u(t)}{argmin}\ \Phi_c \tag{4.27}$$

where Φ_c defines a penalty functional of the general form

$$\Phi_c = \Phi_f(x_f(t_f),t_f) + \int_{t_0}^{t_f} L(x,u,\tau)dt \tag{4.28}$$

is given by the simultaneous solution of the following set of equations

$$\begin{aligned}
&\dot{x}_s = \frac{\partial H}{\partial \lambda} = f(x_s,u,t) && \textit{state equations}\\
&\dot{\lambda} = -\frac{\partial H}{\partial x_s} = -\frac{\partial f^t}{\partial x_s}\lambda - \frac{\partial L}{\partial x_s} && \textit{costate equations}\\
&\frac{\partial H}{\partial u} = \frac{\partial L}{\partial u} + \frac{\partial f^t}{\partial u}\lambda && \textit{stationary condition}\\
&x_s(t_o) = x_o && \textit{initial condition}\\
&\left[\frac{\partial \Phi_f}{\partial x_s} - \frac{\partial \Psi^t}{\partial x_s}\nu - \lambda\right]_{t_f}^{t} dx_s(t_f) + \left[\frac{\partial \Phi_f}{\partial t} + \frac{\partial \Psi^t}{\partial t}\nu + H\right]_{t_f} dt_f = 0 && \textit{transversality condition}
\end{aligned} \tag{4.29}$$

where H is the Hamiltonian defined as

$$H = L(x_s(t),u,t) + \lambda^t f(x_s,u,t) \tag{4.30}$$

and where λ is a vector of Lagrange multipliers. The stationary condition can be used to eliminate the control command **u** from the state equations. The result is a homogeneous two point boundary value problem (TPBV).

For the specific penalty functional presented in equation (4.3) (without final manifolds), the TPBV problem reduces to

$$\begin{aligned}
\dot{\boldsymbol{x}}_s &= \boldsymbol{f}(\boldsymbol{x}_s,\boldsymbol{u},t) && \textit{state equations} \\
\dot{\boldsymbol{\lambda}} &= -\frac{\partial \boldsymbol{f}^t}{\partial \boldsymbol{x}_s}\boldsymbol{\lambda} + 2\frac{\partial \boldsymbol{h}^t}{\partial \boldsymbol{x}_s}\boldsymbol{Q}_1\left[\boldsymbol{h}(\boldsymbol{x}_s,\boldsymbol{u})-\boldsymbol{r}\right] && \textit{costate equations} \\
\boldsymbol{u} &= -\frac{1}{2}\boldsymbol{Q}_2^{-1}\frac{\partial \boldsymbol{f}^t}{\partial \boldsymbol{u}}\boldsymbol{\lambda} && \textit{stationary condition} \\
\boldsymbol{x}_s(t_o) &= \boldsymbol{x}_o && \textit{initial condition} \\
\boldsymbol{\lambda}(t_f) &= 0 && \textit{transversality condition}
\end{aligned} \tag{4.31}$$

where **r** is the reference trajectory. The solution of equation (4.31) can be numerically obtained using a method for solving TPBV problems, such as the shooting method. This numerical solution is essentially a Newton-Raphson zero search on the boundary conditions to solve a mutivariable root-finding problem. For example, the solution of equation (4.31) is obtained by finding the n values of $x_s(t_f)$ and $\lambda(t_0)$ such that the n values of initial and transversality conditions are met. Therefore, the problem is to solve $x_s(t_f)$ and $\lambda(t_0)$ such that

$$\left\{\begin{matrix} \boldsymbol{x}_s(\boldsymbol{x}_s(t_f),\boldsymbol{\lambda}(t_0),t_0)-\boldsymbol{x}_o \\ \boldsymbol{\lambda}(\boldsymbol{x}_s(t_f),\boldsymbol{\lambda}(t_0),t_f) \end{matrix}\right\} = \boldsymbol{0} \tag{4.32}$$

<u>Remarks</u>

① The transversality condition in (4.29) accounts for a final manifold. For a desired final manifold of the form x_d-$x_s(t_f)$=0, the stationary condition becomes $x_s(t_f)$=x_d, and both of the costate initial and final values have to be defined.

② To include a control horizon different from the prediction horizon (see equation (4.4)), dynamic extension has to be used to augment the plant dynamics and consider the control command derivative as the systems input.

③ The introduction of bounds on the state and input has to be solved using slack-variables. The reader is referred to Sage and White [93] for more details.

④ The solution of the problem using a shooting method is computationally demanding. Therefore, this approach is not suitable to real-time implementation.

⑤ One problem with this approach is the structure of the costate equation. If the system's model represented by the state dynamics contains stable or marginally stable components, then the costate dynamics are unstable. In a finite-time optimal control problem, this can be dealt with if the costates are bounded over the integration horizon. However, practically, numerical difficulties may occur if extreme values are reached.

⑥ Finally, the existence of a solution to a multidimensional Newton root finding problem is assessed by the Kantorovitch theorem stating *"Having a lower bound β on the norm of the Jacobian matrix (to avoid singular points), a lower bound K on the norm of the Hessian matrix (to avoid discontinuous Jacobian), a lower bound η_0 on the correction term and verifying that $\eta_0\beta K$ is less or equal to 0.5, is sufficient to obtain convergence"*. This is a sufficient condition, and its conclusion is definitive only if it is met.

4.3.2 - Extended MPC

The method presented in the previous section is a formal theoretical approach to solve the optimal control problem. When the system is relatively slow (such as some chemical processes), the time available between samples may be sufficient to solve the variational formulation numerically. For systems with faster dynamics, the method is extremely difficult to realize in real-time, even with the computational power of today's computers.

For control affine systems with mild nonlinearities, which are nonlinearities with relatively slow variations, one possible approach is the Extended MPC (EMPC). In EMPC, the system's dynamics are linearized around the current state, and a linear MPC design procedure, as presented in section 4.2, is applied to define the control command for the given instant and state.

A control affine system is a special case of equation (4.1) having the following structure

$$\begin{aligned} \dot{\boldsymbol{x}}_s &= \boldsymbol{f}(\boldsymbol{x}_s) + \boldsymbol{G}(\boldsymbol{x}_s)\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}_s) + \boldsymbol{L}(\boldsymbol{x}_s)\boldsymbol{u} \end{aligned} \tag{4.33}$$

where $f:\mathbb{R}^n-\mathbb{R}^n$ is a n-dimensional vector function, $G:\mathbb{R}^n-\mathbb{R}^{n\times m}$ is a n by m control direction matrix function, $h:\mathbb{R}^n-\mathbb{R}^l$ is a l dimensional vector function and $L:\mathbb{R}^{n}-\mathbb{R}^{l\times m}$ is a l by m direct link matrix function. A linerization of this dynamic equation around the operating point (x_0,u_0) is given by

$$\begin{aligned} \dot{x}_s &= f(x_0) + G(x_0)u_0 + \left[\nabla_{x_s} f(x_s) + \nabla_{x_s}\left(G(x_s)u_0\right)\right]_{x_0}\left(x_s - x_0\right) + G(x_0)\left(u - u_0\right) \\ y &= h(x_0) + L(x_0)u_0 + \left[\nabla_{x_s} h(x_s) + \nabla_{x_s}\left(L(x_s)u_0\right)\right]_{x_0}\left(x_s - x_0\right) + L(x_0)\left(u - u_0\right) \end{aligned} \tag{4.34}$$

of the form of a standard LTI state space model with a constant drift

$$\begin{aligned} \dot{x}_s &= A\ x_s + Bu + \dot{x}_0 \\ y &= C\ x_s + Du + y_0 \end{aligned} \tag{4.35}$$

for which the procedure shown in example 4.1 can be used. Usually, the linearization is performed around a point (x_0,0) and the constant drift term $\dot{x}_0$ is usually collocated with the input **u** such that it can be compensated directly.

Example *4.2 - Extended MPC for Position Control of a Pendulum*

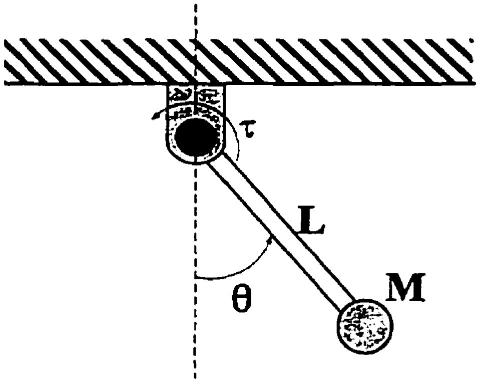


Figure 4.5 - Pendulum

Consider the pendulum shown in figure 4.5. The nonlinear dynamics of the pendulum is given by the following state equations

$$\dot{x}_s = \begin{Bmatrix} \dot{\theta} \\ \ddot{\theta} \end{Bmatrix} = \begin{Bmatrix} \dot{\theta} \\ -\frac{g}{L}\sin(\theta) \end{Bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ML^2} \end{bmatrix}\tau \tag{4.36}$$

If only the angle θ is measured, the output equation is

$$y = \theta = [1 \quad 0]x_s \tag{4.37}$$

The dynamics of the pendulum can be linearized around the point $x_0=\{\theta_0, \theta_0\}$ and $\tau=0$ to give

$$\dot{x}_s = \begin{Bmatrix} \theta \\ \dot{\theta} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L}\cos(\theta_0) & 0 \end{bmatrix}\begin{Bmatrix} \theta \\ \dot{\theta} \end{Bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ML^2} \end{bmatrix}\tau + \begin{Bmatrix} 0 \\ -\frac{g}{L}\left[\sin(\theta_0) - \cos(\theta_0)\theta_0\right] \end{Bmatrix} \tag{4.38}$$

of the form of equation (4.35). Compensating the constant drift term by defining

$$\tau = \tau_{mpc} + LMg\left[\sin(\theta_o) - \cos(\theta_o)\theta_o\right] \qquad \textbf{(4.39)}$$

and following the procedures shown in example 4.1, an EMPC law can be designed for providing τ_{mpc}. The linear control law is valid about the point x_0. The implementation strategy is to perform the linearization at every sampling step, to design the proper feedback law for the linearized system and to apply the resulting control signal. For a pendulum with L=0.5 m and M=5 Kg, using a sampling frequency of 500 Hz, a horizon of 20 samples and with the weighting factor $\gamma/\alpha = 1\times10^{-6}$ (refer to example 2.1), the simulated response to a step reference of 0.5 radian at 0.1 second is given in figure 4.6 and 4.7.

In this example, as in example 2.1, the effect of prediction is visible since the motion is started before the step is requested. The linearized plant is acting as a mass-spring system with a rigidity of g/Lcos(θ_0) with a spring offset of θ_0.

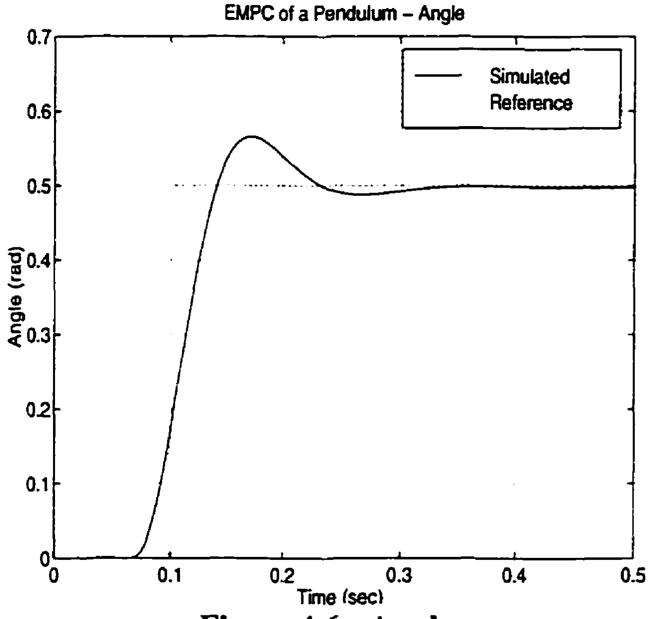


Figure 4.6 - Angle

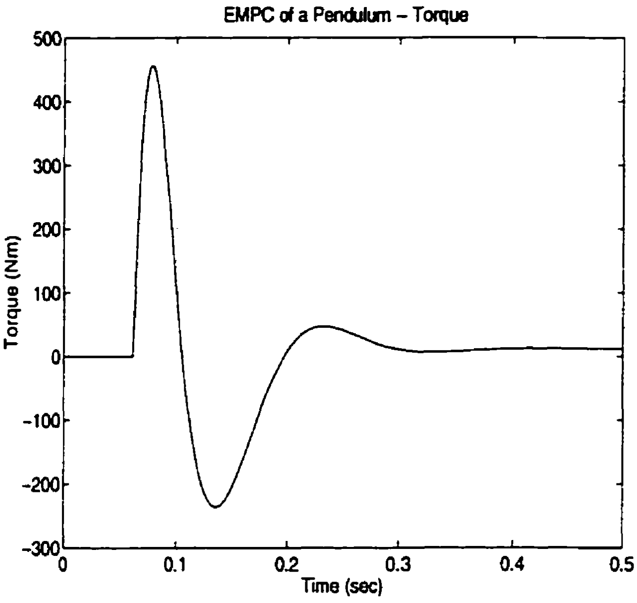


Figure 4.7 - Torque

<u>Remarks</u>

① EMPC defeats the purpose of using MPC to control robot manipulators with unilateral impact/contact. In this latter case, MPC is used to predict changes in the dynamic structure of the system and after linearization, such transition becomes unpredictable.

② To apply EMPC to non smooth systems, all discontinuities should be relaxed.

4.3.3 - Feedback Linearization

When the nonlinearities are well modelled, a nonlinear compensation law cancelling the system's nonlinearity can be found, leaving apparent only the linear portion of the model. Such approaches are known as feedback linearization methods. They are well documented and based on sound theoretical foundations. The description of feedback linearization presented herein is extremely brief. The reader who whishes to get more information is referred to [94, 95 ,96].

Separating the linear and nonlinear components of the drift term, the dynamics of a nonlinear control affine system can be expressed as,

$$\dot{\boldsymbol{x}}_s = \boldsymbol{A}\boldsymbol{x}_s + \boldsymbol{f}(\boldsymbol{x}_s) + \boldsymbol{G}(\boldsymbol{x}_s)\boldsymbol{u} \tag{4.40}$$

where $A \in \mathbb{R}^{n\times n}$ is a constant matrix and f contains the nonlinear component of the model. Essentially, two approaches exist for the linearization of nonlinear systems through feedback: Input/State linearization and Input/Output linearization.

First, when the nonlinearities in the system's model are collocated with the input and all necessary states are available, then there exists a control law of the form

$$\boldsymbol{u} = \boldsymbol{G}(\boldsymbol{x}_s)^{-1}\left(\mathrm{v} - \boldsymbol{f}(\boldsymbol{x}_s)\right) \tag{4.41}$$

linearizing directly the nonlinear systems, and resulting in the compensated system

$$\dot{\boldsymbol{x}}_s = \boldsymbol{A}\boldsymbol{x}_s + \mathrm{v} \tag{4.42}$$

In equation (4.41), G^{-1} represents a generalized inverse. If rank $G=m<n$, the nonlinearities should reside in the m-dimensional space defined by the image of G (definition of collocation). When nonlinearities are not collocated with the input, control laws of the form of (4.41) resulting in (4.42) do not exist. However, a smooth bijective state-transformation (diffeomorphism)

$$z = \boldsymbol{W}(\boldsymbol{x}_s\ ) \tag{4.43}$$

transforming resulting in nonlinearities collocated with the inputs may still exists. Systems for which such diffeomorphisms exists, are said to be feedback linearizable.

In practice, however, applying this procedure directly to Multiple-Input-Multiple-Output (MIMO) systems may be difficult and, in addition, requires the knowledge of all states. For this reason, a second approach called Input/Output (I/O) linearization is often used. In I/O linearization, the control affine model is

$$\begin{aligned} \dot{\boldsymbol{x}}_s &= \boldsymbol{f}(\boldsymbol{x}_s) + \boldsymbol{G}(\boldsymbol{x}_s)\boldsymbol{u} = \begin{Bmatrix} f_1(\boldsymbol{x}_s) \\ \vdots \\ f_n(\boldsymbol{x}_s) \end{Bmatrix} + \{\, g_1(\boldsymbol{x}_s) \;\; \cdots \;\; g_m(\boldsymbol{x}_s) \,\}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}_s,\boldsymbol{u}) = \begin{Bmatrix} h_1(\boldsymbol{x}_s,\boldsymbol{u}) \\ \vdots \\ h_p(\boldsymbol{x}_s,\boldsymbol{u}) \end{Bmatrix} \end{aligned} \tag{4.44}$$

where $g_i:\mathbb{R}^n \rightarrow \mathbb{R}^n$ defines the control direction of the control input I and $h_i:\mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ defines the output I. Then, the procedure consists of continuously differentiating each output with respect to time until at least one input appears collocated with each output derivative. Defining r_i to be the order of the derivative to obtain a collocation for the output i, the following matrix equation is obtained

$$\begin{Bmatrix} y_1^{(r1)} \\ \vdots \\ y_p^{(rp)} \end{Bmatrix} = \begin{Bmatrix} L_f^{(r1)}\boldsymbol{h}_1 \\ \vdots \\ L_f^{(rp)}\boldsymbol{h}_p \end{Bmatrix} + \begin{bmatrix} L_{g_1}L_f^{(r_1-1)}\boldsymbol{h}_1 & \cdots & L_{g_m}L_f^{(r_1-1)}\boldsymbol{h}_1 \\ \vdots & \ddots & \vdots \\ L_{g_1}L_f^{(r_p-1)}\boldsymbol{h}_p & \cdots & L_{g_m}L_f^{(r_p-1)}\boldsymbol{h}_p \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_m \end{Bmatrix} \tag{4.45}$$

where the matrix multiplying the control vector is called the decoupling matrix and is usually denoted by $E(x_s)$. In equation (4.45), the operation L is the Lie derivative operator defined as

$$L_{f(x)}\, h(x) = \nabla_x h(x)\, f(x) \quad \textit{and} \quad L^i_{f(x)}\, h(x) = L_{f(x)}\left(L^{i-1}_{f(x)}\, h(x)\right) \tag{4.46}$$

and represents the projection of the gradient of a function of multiple variables along the direction defined by the vector function *f*. If m≥p and rank(E(x))≥p, then the linearizing control law

$$\boldsymbol{u} = \boldsymbol{E}^{-1}(\boldsymbol{x}) \left\{ -\begin{Bmatrix} L_f^{(r1)}\boldsymbol{h}_1 \\ \vdots \\ L_f^{(rp)}\boldsymbol{h}_p \end{Bmatrix} + \begin{Bmatrix} \nu_1 \\ \vdots \\ \nu_p \end{Bmatrix} \right\} \tag{4.47}$$

results in a linear decoupled input/output model of the form $y_i=v_i$, $\forall i\in[1,p]$. The inverse of the decoupling matrix is also a generalized inverse. Usually, I/O linearization is applied to square systems, that is p=m.

In MPC with feedback linearization, a linear MPC design is applied to the linearized plant, once feedback linearization is applied. This is shown in the following example.

Example 4.3 - Position Control of Pendulum using MPC with Feedback Linearization

Consider the same pendulum as in example 4.2 (figure 4.5) whose dynamics is defined by equations (4.36) and (4.37). Exact linearization results in

$$\tau = ML^2\left(v_l + \frac{g}{L}\sin(y)\right) \tag{4.48}$$

Applying the control law (4.48) to the dynamics (equation (4.36)), one obtains

$$\dot{x}_s = \begin{Bmatrix}\dot{\theta} \\ \ddot{\theta}\end{Bmatrix} = \begin{bmatrix}0 & 1 \\ 0 & 0\end{bmatrix}\begin{Bmatrix}\theta \\ \dot{\theta}\end{Bmatrix} + \begin{bmatrix}0 \\ 1\end{bmatrix} v_l \tag{4.49}$$

which is a linear system of the form of equation (4.15). The linearized control command v_l can be defined using the same procedure as in example 4.1.

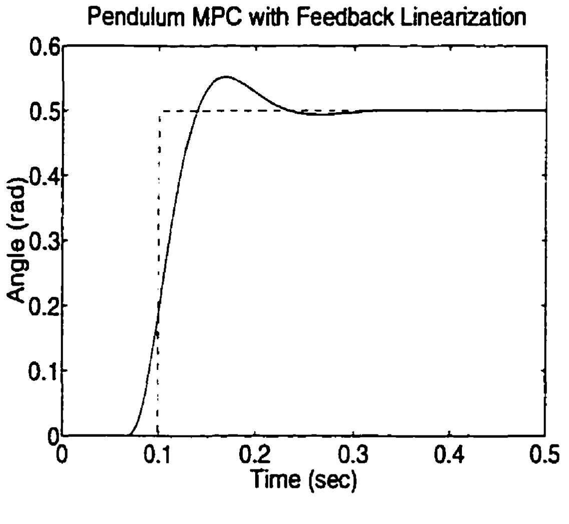


Figure 4.8 - Angle

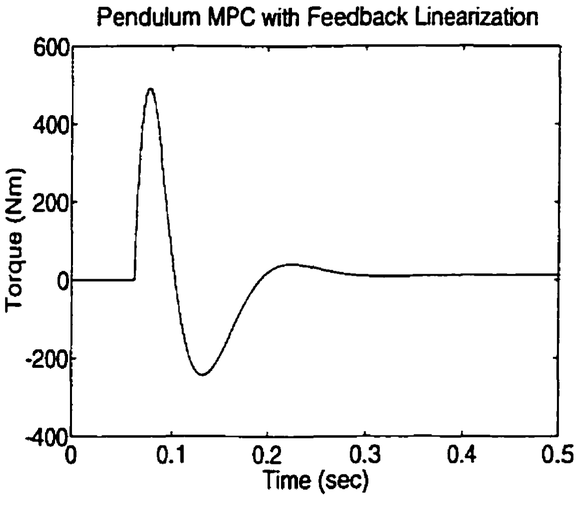


Figure 4.9 - Torque

For a pendulum with L=0.5 m and M=5 Kg, using a sampling frequency of 500 Hz, a horizon of 20 samples and with the weighting factor $\gamma/\alpha = 1\times10^{6}$ (refer to example 2.1), the simulated response to a step reference of 0.5 radian at 0.1 second is given in figure 4.8 and 4.9. This control schemes results in two superimposed feedback loops: one nonlinear compensation loop and one MPC feedback loop. The results obtained are comparable to those for EMPC.

Remarks

① The presentation of feedback linearization shown here is extremely simplified. Nevertheless, it explains the main idea.

② The application of feedback linearization to nonlinear systems requires that the system's nonlinearities be sufficiently smooth (in the mathematical sense) for the existence of the derivatives needed to perform I/O linearization. For the contact model presented in example 2.1, this condition is not met. Therefore, the application of this approach for the control of a robot manipulator with impact/contact motion is limited. In addition, for robot contact motion control, linearization of the contact dynamics using the robot actuators is impossible before contact occurs.

③ Controllers using feedback linearization were found extremely sensitive to model uncertainties. As discussed in section 3.2.3, the models used for impact/contact are extremely simplified, limiting the applicability of feedback linearization to solve the control problem of section 3.1.

④ As discussed in section 3.2.4, the effect of actuator, sensor and interface dynamics is important in impact control. A proper feedback linearization, valid during impact, would require these effects included in the model. With a variable structure system, the result would be different linearization laws for different structures, implemented using a switching law. The purpose of the research presented herein is exactly to avoid that problem.

⑤ Some methods presented in section 3.3 use feedback linearization and apply it to the contact motion control problem (see equation (3.17), (3.26) and (3.38)). The validity of doing so resides on the fact that the methods are developed to solve only a portion of the global control problem defined in chapter 3.

⑥ Model-based prediction has other interesting applications in feedback linearization. Necsulescu et al.[97] have used one-step-ahead dynamic-based predictions of the control commands to rescale the reference when control saturation were anticipated. This was done to prevent failure in feedback linearization. This dynamic based prediction of discontinuities is the main motivation for using MPC in impact/contact motion control of manipulators.

4.3.4 - Mathematical Programming

The discrete-time equivalent of any optimal control problem can be solved through mathematical programming[1]. The terminology *mathematical programming* refers to all the numerical methods providing a direct solution to a dynamically or statically constrained optimization problem. A schematic representation of the existing methods is shown in figure 4.10. The following discussion is not intended to be a presentation of mathematical programming methods. Various references provide good descriptions (see Fletcher [98] for example). The objective of this section is to present an overview of the numerical methods that exists, and to discuss their application for the nonlinear MPC problem.

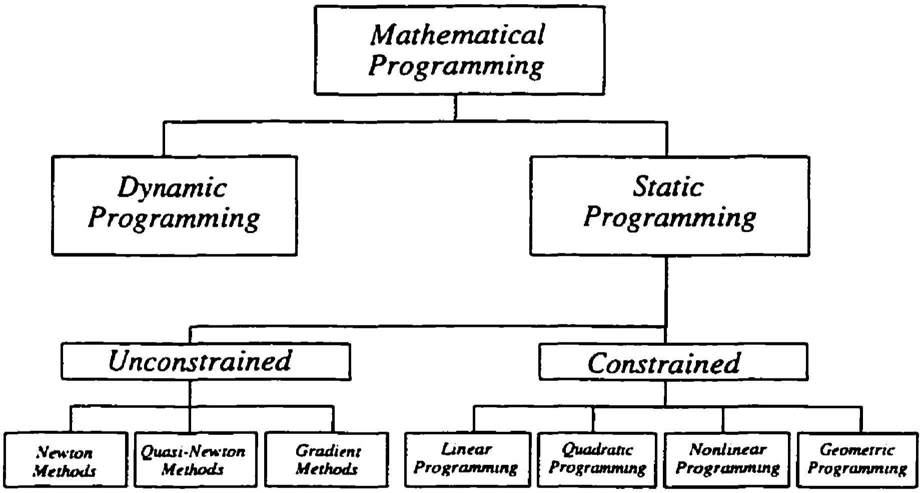


Figure 4.10 - Mathematical Programming

[1] *See Sage and White [93], section 6.4.*

4.3.4.1 Dynamic Programming

The solution of a dynamically constrained optimization problem, as the one defined by an optimal control problem, can be obtained using Dynamic Programming (DP). DP uses the Pontryagin maximum principle[2] to define the Hamilton-Jacoby-Bellman equation,

$$-\frac{\partial \Phi^*}{\partial t} = \min_{u(\tau)} \left(L + \left(\frac{\partial \Phi^*}{\partial x_s} \right)^t f \right) \tag{4.50}$$

that represents the evolution of the optimal cost function with respect to the optimal control command. The function L in (4.50) is called the Lagrangian and is defined in equation (4.28). For this continuous-time formulation, as for the variational formulation, the solution of the Two Point Boundary Value (TPBV) problem represented by solving equations (4.50) and (4.1) backwards is computationally demanding.

A computer algorithm to solve the dynamic programming problem was developed, requiring the discretization of the state, output and control vectors. The algorithm, called to discrete dynamic programming (DDP) algorithm, is much simpler than the solution of the continuous-time problem. However, a huge amount of computer memory is required. For applications where the discretized variables can take a broad range of values and where precision is needed, the so-called *curse of dimensionality*[3] limits the application of real-time discrete dynamics programming. Therefore, this approach is not adapted to the solution of the control problem defined in chapter 3.

4.3.4.2 Unconstrained Static Programming

Finite time optimal control can be solved numerically using a static optimization method. For the dynamical system represented by the state equations (4.1), the optimal solution can be assumed to have a prescribed form

[2] *"An optimal policy is also an optimal policy with regard to the states resulting from previous decisions."*

[3] *Bellman calls the growth in number of arithmetic operations and in the need for bookeeping, as the discretization grid is made finer, the curse of dimensionality. In fact, if there is n states each having s possible values and m inputs each having r possible values, then the DDP solution over an horizon of H_p steps requires at most s^n r^m H_p operations and variables to store.*

$$\boldsymbol{u}^* = \boldsymbol{u}(\boldsymbol{x}^*) \tag{4.51}$$

where $\boldsymbol{u}^*$ is the optimal command and $\boldsymbol{x}^*$ the optimal trajectory. This optimal solution is constrained by the discrete-time dynamics (using forward differences)

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}(\boldsymbol{x}_k)) \cdot dt \\ \boldsymbol{y}_k &= \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}(\boldsymbol{x}_k)) \end{aligned} \tag{4.52}$$

where dt is the time step. Similarly, the discrete-time form of the penalizing functional defined in equation (4.3) becomes

$$\Phi(\boldsymbol{x}_k) = \sum_{i=0}^{i=h_p} \left[(\boldsymbol{y}_i(\boldsymbol{x}_k) - \boldsymbol{r}_i)^t \boldsymbol{Q}_1 (\boldsymbol{y}_i(\boldsymbol{x}_k) - \boldsymbol{r}_i) + \boldsymbol{u}_i(\boldsymbol{x}_i)^t \boldsymbol{Q}_2 \boldsymbol{u}_i(\boldsymbol{x}_i) \right] \cdot dt \tag{4.53}$$

where all the variables $\mathbf{y}_i$ and $\mathbf{u}_i$ are defined using (4.51) and (4.52). The optimization problem resulting from this time discretization is static, and standard static optimization methods can be used.

Unconstrained static programming includes a series of standard methods, based on various approaches to find the minimum of a single-valued function of multiple variables. Some ad hoc methods, such as the Simplex method, were developed and are not shown in figure 4.10. For the three families of methods shown in figure 4.10, the main idea is to perform an iterative minimization process in which the function is successively minimized along different directions

$${}^i\alpha_k^* = \min_{\alpha} \Phi({}^i\boldsymbol{x}_k + \alpha\, {}^i\boldsymbol{s}_k) \tag{4.54}$$

where the leading superscript defines the iteration step, ${}^i\mathbf{x}_k$ is the minimum point at the ith iteration of the kth sample, α is a scalar line parameter, ${}^i\mathbf{s}_k$ is a search direction at the ith iteration of the kth sample. Equation (4.54) represents a single variable minimization along a line. At the ith iteration, the point from which to search at the next iteration is given by

$${}^{i+1}\boldsymbol{x}_k^* = {}^i\boldsymbol{x}_k + {}^i\alpha^*\, {}^i\boldsymbol{s}_k \tag{4.55}$$

The difference between the different methods lies mainly in the way the search direction ${}^i\mathbf{s}_k$ is updated at each iteration. The single-variable line minimization along the search direction, defined by equation (4.54), can be done using any standard method. Commercial softwares, such as the *Numerical Recipes in C* [99] and the *Matlab Optimization Toolbox* [100], uses a Golden section search algorithm for the line minimization.

Gradient methods rely exclusively on the knowledge of the gradient $\nabla_x\Phi$ of the cost function to define the search direction. One simple method in this category is the steepest descent method that defines the search direction s in the opposite direction of the gradient

$$s = -\nabla_x \Phi \qquad (4.56)$$

The decision of using equation (4.56) relies on a linear model of the function to minimize. For example, if the function has the form

$$\Phi(x) = a^t x + b \qquad (4.57)$$

then the search direction becomes -a and the minimum is reached within one iteration. However, the function to be minimized is not necessarily linear. In practice, most descent methods are numerically unstable. The reason for this instability is easily understood. Interestingly, when the function minimized does not have a linear structure, the gradient of the function at the line minimum is necessarily orthogonal to the search direction (by the definition of minimum). Therefore, if the minimum of the function is not reached, the next direction search is orthogonal to the previous one, meaning that initially, the search direction was in the wrong direction.

By considering a quadratic structure to the function of n variables, one can define a set of n independent search directions, called conjugate directions, along which the line minimization performed in any sequence gives the function's minimum. This well known fact, reflected in the conjugate direction theorem, is the essential part of the conjugate direction methods such as the Powell direction set method and the Fletcher-Reeves conjugate gradient method[98].

Another standard method, based on the quadratic model, is the Newton method. It consists essentially of applying a Newton-Raphson root finding algorithm to the function's gradient. This obviously requires the knowledge at any point of the Hessian matrix $\nabla^2_x\Phi$ of the function Φ. Then, the search direction is

$$s = -\left[\nabla^2_x \Phi\right]^{-1} \nabla_x \Phi \qquad (4.58)$$

which corrects the steepest descent method to take into consideration the variation of the gradient.

In most applications, the computation of the Hessian is not possible. For these cases, a series of methods called Newton-like, quasi-Newton or variable metric methods, have been developed to approximate intelligently the Hessian matrix without causing numerical instabilities. Over the years, many algorithms have been proposed. The initial work included the methods by Davidon, Fletcher and Powell[98]. More recently, the Broyden family of algorithms were developed and proven very efficient. Among them, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is the most popular and is mainly used in commercial packages. For the BFGS method, the approximation of the Hessian matrix **H** at the ith iteration step (for the next search direction) is given by

$$ {}^{i+1}\boldsymbol{H} = {}^{i}\boldsymbol{H} + \left(1 + \frac{\gamma^{t}\,{}^{i}\boldsymbol{H}\gamma}{\delta^{t}\gamma} \right) - \left(\frac{\delta\gamma^{t}\,{}^{i}\boldsymbol{H} + {}^{i}\boldsymbol{H}\gamma\delta^{t}}{\delta^{t}\gamma} \right) \tag{4.59} $$

where $\delta = {}^{i+1}x + {}^{i}x$ and $\gamma = {}^{i+1}\{\nabla_x \Phi\} - {}^{i}\{\nabla_x \Phi\}$.

For real-time applications, the static optimization methods require, at each sampling step, the integration of the system's dynamic model over the receding prediction horizon for each iteration of the optimization process. Similarly, the line minimization and the evaluation of the next search direction (if not converged), are executed at each iteration of the optimizing procedure. The amount of computation required is enormous. Moreover, although the number of line search iterations can be proven finite, the line minimization is an exact minimization that cannot be carried out in a finite number of steps. For real-time solutions, where the time allocated is limited, this property is a major drawback. Nevertheless, the mathematical programming approach represents the best alternative for implementing nonlinear MPC for the impact/contact motion control of a robot manipulator.

4.3.4.3 Constrained Static Programming

Constrained static optimization problems (in which some variables are constrained) are solved using the methods shown in figure 4.10: Linear Programming (LP), Quadratic Programming (QP), Nonlinear Programming (NLP) and Geometric Programming (GP). These methods are standard and various commercial software packages are available. The selection of one method depends on the type of function to minimize and on the type of constraints. The amount of computation needed in these cases is not compatible with real-time implementation and, therefore, a detailed presentation is not given here.

Chapter V

Contact Motion Control Using MPC

The models presented in chapter two are far too complex for most control applications. In that respect, the assumptions presented in chapter three narrow down the field of study without lost of generality regarding the applicability of the control approach proposed. According to the concept of MPC, presented in chapter four, the effect of prediction for linear systems is limited to an enhanced ability to deal with pure delays and pro-actively respond to any future change in the control objective. For a discontinuous nonlinear system, however, the ability to predict the discontinuous change in the state dynamics is an important and interesting feature. This property represents the main motivation for using MPC for process control where state and input saturations are common.

In robot manipulator control, it is common practice to introduce a virtual viscous layer on the surface of a contacting object to prevent excessive impacts and improve contact stability. In this particular case, the prediction of an impact relies mainly on a static state reaction and does not involve the manipulator dynamics. As a result, the boundary thickness and the virtual fluid viscosity are defined based exclusively on the worst case scenario(maximum speed, perpendicular to the contact surface, maximum inertia configuration). With MPC, the reaction to a predicted impact can take into account the arm dynamics, and the control action can be adjusted automatically to the dynamic state prevailing just before impact.

In this chapter, the use of MPC for impact/contact motion control is analysed. The equations defining the nonlinear MPC problem are analytically set. A method to convert the continuous-time problem to a discrete-time equivalent is also presented. The various implementation strategies for nonlinear MPC are introduced in details. Questions such as the existence of a solution to the MPC problem and reachability of the reference trajectory are also addressed.

5.1 - Analytical Setting

In chapter four, the general MPC concept was formulated for a dynamic system represented by nonlinear state equations. In this section, the MPC problem is considered from the point of view of the specific control objectives defined in chapter 3. First, considering the assumptions 3.1 to 3.4, a design model is obtained. Then, the resulting nonlinear discontinuous MPC problem is formulated. And finally, an equivalent discrete-time problem is formulated.

5.1.1 - The Design Model

Using assumptions 3.1 to 3.4, the dynamic model presented in example 2.3 can be reduced to the twenty-two equations shown in the following table.

DESCRIPTION	EQUATIONS
Manipulator Dynamics	$M(q)\ddot{q} + N(\dot{q},q) + G(q) = \tau - \tau_f(\tau,\dot{q},q) + J_c^t f_e$ (5.1)
Joint Friction	$\tau_f(\tau,\dot{q},q) = \begin{Bmatrix} \tau_{f_1} \\ \vdots \\ \tau_{f_i} \\ \vdots \\ \tau_{f_N} \end{Bmatrix}$ (5.2) $\tau_{f_i} = \begin{cases} \tau_{k_i}\, sgn(\dot{q}_i) & \text{if } \dot{q}_i \neq 0 \\ \tau_{s_i}\, sgn(\tau_{eq_i}) & \text{if } \dot{q}_i = 0 \text{ and } \tau_{eq_i} \geq \tau_{s_i} \\ \tau_{eq_i} & \text{if } \dot{q}_i = 0 \text{ and } \tau_{eq_i} < \tau_{s_i} \end{cases}$ (5.3)

Table 5.1

	τ_{eq} *: obtained by solving a system of n equations-n unknowns* $\boldsymbol{M}(\dot{\boldsymbol{q}})\ddot{\boldsymbol{q}}^{*} + \boldsymbol{N}(\dot{\boldsymbol{q}},\boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q}) = \tau - \tau_{eq} + \boldsymbol{J}^{T}\boldsymbol{f}_{e}$ **(5.4)** $\ddot{q}_i^{*} = \begin{cases} \ddot{q}_i & \mathit{if}\ \dot{q}_i \neq 0 \\ 0 & \mathit{otherwise} \end{cases}$ $\tau_{eq_i} = \begin{cases} \tau_{k_i} sgn(\dot{q}_i) & \mathit{if}\ \dot{q}_i \neq 0 \\ \tau_{eq_i} & \mathit{otherwise} \end{cases}$
Manipulator and Contact Kinematics	$\boldsymbol{x}_e = \Gamma(\boldsymbol{q})$ **(5.5)** $\dot{\boldsymbol{x}}_e = \nabla_q \Gamma(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$ **(5.6)** $\begin{Bmatrix} {}^{c}\dot{\boldsymbol{x}}_e \\ {}^{o}\dot{\boldsymbol{x}}_e \end{Bmatrix} = \boldsymbol{J}_c(\boldsymbol{q},\boldsymbol{x}_c)\dot{\boldsymbol{q}}$ **(5.7)**
End Effector Geometry	$S_b \triangleq \left\{\boldsymbol{x}_b \mid \phi_e(\boldsymbol{x}_b, {}^{p}\boldsymbol{x}_e) = 0,\ {}^{p}\boldsymbol{x}_e \ \mathit{is\ given}\right\}$ **(5.8)**
Environment Geometry	$S_{env} \triangleq \left\{ \boldsymbol{x}^{*}(\eta) + \boldsymbol{x}_{env} \mid \phi_{env}\left(\boldsymbol{x}^{*},\boldsymbol{x}_{env}\right) = 0 \right\}$ **(5.9)**
Penetration Distance	$d \triangleq \min_{z \in Z} z$ **(5.10)** $Z = \left\{ z \mid \boldsymbol{x}_{env} + \boldsymbol{x}^{*} + z\boldsymbol{n} = {}^{p}\boldsymbol{x}_e + \boldsymbol{x}_b,\ \boldsymbol{x}^{*} \in S_{env},\ \boldsymbol{x}_b \in S_b,\ {}^{p}\boldsymbol{x}_e \ \mathit{is\ given} \right\}$**(5.11)**

End Effector and Environment Contact Points	${}^{c}\mathrm{x}_{e} \in S_{b}$ and ${}^{c}\mathrm{x}_{env} \in S_{env}$ such that $\boldsymbol{x}_{env} + {}^{c}\boldsymbol{x} + d\boldsymbol{n} = {}^{p}\boldsymbol{x}_{e} + {}^{c}\boldsymbol{x}_{b}$ (5.12)
Environment Surface Normal Vector	$\boldsymbol{n}(\boldsymbol{x},\boldsymbol{x}_{env},\eta) = \dfrac{\nabla_{e_x}\phi_{env}(\boldsymbol{x},\boldsymbol{x}_{env},\eta)}{\lvert\nabla_{e_x}\phi_{env}(\boldsymbol{x},\boldsymbol{x}_{env},\eta)\rvert}$ (5.13)
Contact Force	$\boldsymbol{f}_{e} = P(d,\dot{d})\ \boldsymbol{n} + F_{f}\ \boldsymbol{e}_{t}$ (5.14)
Normal Contact Force	$P(d,\dot{d}) = \begin{cases} K_{e}\ d & \text{if } d<0 \text{ and } \dot{d}\leq 0 \\ eK_{e}\ d & \text{if } d<0 \text{ and } \dot{d}>0 \\ 0 & \text{if } d\geq 0 \end{cases}$ (5.15)
Penetration Speed	$\dot{d} = \boldsymbol{v}_{rel}\cdot\boldsymbol{n}$ (5.16)
Contact Points Relative Velocity	$\boldsymbol{v}_{rel} = \dfrac{d}{dt}\left({}^{p}\boldsymbol{x}_{e} + {}^{c}\boldsymbol{x}_{b}\right) - \dfrac{d}{dt}\left(\boldsymbol{x}_{env} + {}^{c}\boldsymbol{x}\right) = \dot{d}\boldsymbol{n} + v_{t}\boldsymbol{e}_{t}$ (5.17)
Direction of Tangential Motion	$\boldsymbol{e}_{t} = \dfrac{\boldsymbol{v}_{rel} - (\boldsymbol{v}_{rel}\cdot\boldsymbol{n})\boldsymbol{n}}{\lvert\boldsymbol{v}_{rel} - (\boldsymbol{v}_{rel}\cdot\boldsymbol{n})\boldsymbol{n}\rvert}$ (5.18)

Surface Friction	$F_f=\begin{cases} F_k\ sgn(v_t) & if\ v_t\neq 0 \\ F_s\ sgn\left(F_{eq}\cdot e_t(x)\right) & if\ v_t=0\ \ and\ \ \lvert F_{eq}\cdot e_t(x)\rvert \geq F_s \\ F_{eq}\cdot e_t(x) & if\ v_t=0\ \ and\ \ \lvert F_{eq}\cdot e_t(x)\rvert < F_s \end{cases}$ (5.19) $F_{eq}=J_c^{-t}\left(\tau-\tau_f-G-N-\dot{J}_c\dot{q}\right)$ (5.20) $F_k=\mu_k P(d,\dot{d})$ (5.21) $F_s=\mu_s P(d,\dot{d})$ (5.22) NOTE : In contact friction, inertial forces are neglected in F_{eq} because they would cause the overall system to be over-constrained.

Table 5.1 - The Design Model

The design model presented here is still too complex. The discontinuous nature of the normal force (5.15), the joint friction (5.3) and the surface friction (5.19-5.22) clearly shows that the model represents a variable structure system. Moreover, since sticking is included in the joint and surface friction models, the system also contains variable structures with variable topological properties. Under the conditions prevailing during stiction, the force of friction can be viewed as a constraint force preventing motion in one joint. Therefore, the equilibrium of the dynamic system is defined as an equilibrium manifold containing all the states for which stiction will occur.

For simplicity in the notation, the model is represented only by

$$\begin{Bmatrix}\dot{q}\\ \ddot{q}\end{Bmatrix}=\begin{Bmatrix}\dot{q}\\ M^{-1}(q)\ [\ -N(\dot{q},q)-G(q)]\end{Bmatrix}+\begin{bmatrix}0 & 0\\ M^{-1}(q)J_c^t & -M^{-1}(q)\end{bmatrix}\begin{Bmatrix}f_e\\ \tau_f(\tau,\dot{q},q)\end{Bmatrix}+\begin{bmatrix}0\\ M^{-1}(q)\end{bmatrix}\tau \quad (5.23)$$

or, defining the state vector $\mathbf{w}=\{\dot{\mathbf{q}}^t, \mathbf{q}^t\}^t$ (w is used instead of the standard **x** to avoid confusion with the Cartesian coordinate of the end effector)

$$\dot{w}=f(w)+L(w)\alpha(w,\tau)+g(w)\tau \quad (5.24)$$

where *f*, ***L*** and ***g*** are smooth mappings defined on $\mathbb{R}^{2*N}$ (N is the number of degrees of freedom of the manipulator), but *α(w)* is a non smooth vector field containing the discontinuous components of the dynamics (contact force and joint friction). If the contact force and the robot states are measured, the output equation is simply

$$y = \left\{ \begin{matrix} w \\ f_e(w) \end{matrix} \right\} = h(w) \tag{5.25}$$

Although equation (5.23) has a standard state-space form, one should remember that, according to equations (5.3) ,(5.15) and (5.19), α is only piecewise continuous in **w** and τ, and has the form

$$\alpha(w) = \begin{cases} \alpha_1(w) & if\ \{w,\tau\} \in S_1 \\ \alpha_2(w) & if\ \{w,\tau\} \in S_2 \\ \vdots & \vdots \\ \alpha_p(w) & if\ \{w,\tau\} \in S_p \end{cases} \tag{5.26}$$

where S_i is the ith state-torque structure and p is the number of structures of α (in which α is continuous in {w, τ}). Each discontinuous joint friction model exhibits 3 continuous structures. According to equation (5.3), each joint friction model includes one condition under which the freedom provided by the joint is momentarily lost. Therefore, from the 3^N structures of $\alpha(w,\tau)$, there is 2^N-1 structures in which at least one degree of freedom is lost, meaning that the system has variable topology.

5.1.2 - The MPC Problem

MPC is formulated as a finite-time optimal control problem where the joint torque τ at the time t_o is given by

$$\tau(t_o) = \underset{\tau}{argmin}\ \Phi(y,\tau) \tag{5.27}$$

where Φ is the cost functional defined by

$$\Phi = \frac{1}{2} \int_{t_o}^{t_o+t_h} \left[(y(\tau)-r(\tau))^t\, Q_1\, (y(\tau)-r(\tau)) + \tau(\tau)^t\, Q_2\, \tau(\tau) \right] d\tau \tag{5.28}$$

where **r**(t) is an admissible trajectory, t_h is the prediction horizon, Q_1 is a positive definite measurement weighting matrix and Q_2 is a positive semi-definite control weighting matrix. This minimization problem is constrained by equations 5.24 and 5.25. MPC differs from the finite-time optimal control in the use of the receding horizon (T_h) requiring the optimal control problem to be solved continuously. An interesting option in the definition of MPC is the extra constraint

$$\dot{\tau}(0) = 0 \qquad \forall t\in[t_o + t_u, t_h] \tag{5.29}$$

where $0\leq t_u \leq t_h$ is called the control horizon. In the case where $t_u=0$, the MPC solution obtained is called the mean-level MPC solution. This situation is particularly important to reduce the complexity in nonlinear MPC.

5.1.3 - Discrete-Time Equivalent

A discrete-time equivalent formulation of the continuous-time MPC problem is obtainable. This procedure is particularly useful to get a simplified numerical solution to the problem. Using the forward difference approximation

$$\dot{w} \approx \frac{w_{k+1} - w_k}{T} \tag{5.30}$$

where T is the discretizing time step, the dynamic equation (5.23) can be rewritten as

$$w_{k+1} = w_k + \left[f_k + L_k\ \alpha_k + g_k \tau_k\right]T \tag{5.31}$$

where, for simplicity, $f_k=f(w_k)$, $L_k=L(w_k)$, $\alpha_k=\alpha(w_k)$ and $g_k=g(w_k, \tau_k)$. Similarly, the output equation is

$$y_k = h_k \tag{5.32}$$

Using this discrete-time approximation of the state dynamics, a ith step ahead predictor can be defined by

$$w_{k+i} = w_k + \sum_{j=0}^{i-1} \left[f_{k+j} + L_{k+j}\ \alpha_{k+j} + g_{k+j}\tau_{k+j}\right] T \tag{5.33}$$

As opposed to the linear ith step ahead predictor presented in chapter 4 (example 4.1), this predictor cannot explicitly be expressed in terms of $\mathbf{w}_k$ and $\{\tau_k, \ldots, \tau_{k+i-1}\}$. Therefore, a direct minimization as done in example 2.1, is impossible since $\mathbf{w}_{k+j}$ is function of all previous commands and states. However, a recursive predictor of the type defined by (5.33) can be used to evaluate the cost function for an initial condition $\mathbf{w}_k=\mathbf{w}_o$ and for sequence of torque commands $\mathbf{T}_k=\{\tau_k, \ldots, \tau_{k+i-1}\}$. Then, the solution of the MPC problem becomes the determination of the optimal sequence $\mathbf{T}^*_k$

$$T_k^* = \underset{T_k}{argmin}\ \Phi_k \tag{5.34}$$

where the discrete-time cost function is

$$\Phi_k = \left[Y_k - R_k\right]^t Q_1^* \left[Y_k - R_k\right] + T_k^t Q_2^* T_k \tag{5.35}$$

and where

$$Q_1^* = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & Q_1 \end{bmatrix}$$

$$Q_2^* = \begin{bmatrix} Q_2 & 0 & \cdots & 0 \\ 0 & Q_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & Q_2 \end{bmatrix} \tag{5.36}$$

$$Y_k = \left\{ h_{k+1}, \ldots, h_{k+h_p} \right\}^T$$

$$R_k = \left\{ y_{k+1}, \ldots y_{k+h_p} \right\}^T$$

In (5.35), h_p is the prediction horizon in number of steps. For a N DOF robot manipulator, this is basically the minimization of a function of N*h_p variables. Using the extra condition defined by equation (5.28) with a zero control horizon, the mean level MPC problem becomes

$$\tau_k^* = \underset{\tau_k}{argmin}\ \Phi_k \tag{5.37}$$

where the cost function is

$$\Phi_k = \left[Y_k - R_k\right]^t Q_1^* \left[Y_k - R_k\right] + \tau_k^t h_p Q_2 \tau_k \tag{5.38}$$

This mean level MPC formulation converts the multivalued minimization problem to a single-valued one.

5.2 - Synthesis Methods

So far, the equations governing the nonlinear MPC problem were presented academically, without concerns about their numerical solution and their real-time feasibility. In the following section, some of the approaches for implementing nonlinear MPC are described. The algorithms take into consideration the implementation issues and the specific problems associated with the control of robots with frictional/unilateral contacts. As mentioned in section 4.3, complete extended MPC and complete feedback linearization are not suitable for the control of robots with contact tasks. These methods defeat the purpose of using MPC to predict changes in the dynamic behaviour. Such techniques can still be used to reduce the complexity of the smooth portion of the dynamics and to obtain more realistic real-time implementations. Next, various alternatives for implementing nonlinear MPC algorithms in real-time are presented without concern about the existence of a solution. This important analytical aspect will be addressed a subsequent section.

5.2.1 - The Variational Formulation

In the variational formulation, the optimal solution is obtained by solving a Two-Point Boundary Value problem in real-time. For the model (equation 5.23) and the cost functional (equation 5.28) presented in section 5.1.1, the variational formulation results in the optimality equations

$$\begin{aligned}
\dot{w} &= f(w) + L(w)\alpha(w,\tau) + g(w)\tau && \textit{state equations} \\
\dot{\lambda} &= -\frac{\partial[f(w) + L(w)\alpha(w,\tau) + g(w)\tau]^{t}}{\partial w}\lambda + 2\frac{\partial h^{t}}{\partial w}Q_1\,\tilde{y} && \textit{costate equations} \\
\tau &= -\frac{1}{2}Q_2^{-1}\left(L(w)\frac{\partial\alpha(w,\tau)}{\partial\tau} + g(w)\right)^{T}\lambda && \textit{stationary condition} \\
w_s(t_o) &= w_o && \textit{initial condition} \\
\lambda(t_f) &= 0 && \textit{transversality condition}
\end{aligned} \tag{5.39}$$

where each components of the model are given in table 5.1. An important observation about the stationary condition is that the partial derivative of α with respect to τ is zero but when the manipulator is stuck by joint friction (equation 5.4) or the end effector friction (equation 5.19 to 5.22). In that condition, the partial derivative is unity for the joint friction and equal to a row of the contact Jacobian when stuck at the end effector. In any cases, the partial derivation is not function of τ and the stationary condition is well defined. One set of optimality conditions is derived for each structure of the dynamic system. The problem consists of finding the initial values of λ (λ_o) obtain $\lambda(t_f)=0$, or

$$\lambda(t_f, \lambda_o) = 0 \tag{5.40}$$

A typical shooting method basically consists of finding the roots of the function defined by equation 5.40. The algorithm is as follows.

Algorithm 5.1 - TPBV Problem by the Shooting Method

1. *Select an initial guess on λ_o.*
2. *Proceed with a first shot by integrating the state and costate equations using the stationary condition in the state equation.*
3. *If the shot is precise enough ($\lambda_i(t_i)<\epsilon$, $i=1,...,n$), proceed with step 8.*
4. *For each independent variation $\delta\lambda_i(0)$ of the initial condition $\lambda_i(0)$, integrate the state and costate equations to numerically evaluate the Jacobian matrix defined by*

$$\begin{bmatrix} \dfrac{\partial\lambda_1(t_f)}{\partial\lambda_1(0)} & \cdots & \dfrac{\partial\lambda_1(t_f)}{\partial\lambda_n(0)} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial\lambda_n(t_f)}{\partial\lambda_1(0)} & \cdots & \dfrac{\partial\lambda_n(t_f)}{\partial\lambda_n(0)} \end{bmatrix} \begin{Bmatrix} \delta\lambda_1(0) \\ \vdots \\ \delta\lambda_n(0) \end{Bmatrix} = \begin{Bmatrix} \delta\lambda_1(t_f) \\ \vdots \\ \delta\lambda_n(t_f) \end{Bmatrix}$$

5. *Correct the shot using ${}^{i+1}\lambda_o = {}^{i}\lambda_o + {}^{i}\Delta\lambda_o$ where the leading superscript refers to the iteration step and ${}^{i}\Delta\lambda_o$ is obtained by solving*

$${}^{i}\Delta\lambda_o = \begin{bmatrix} \dfrac{\partial\lambda_1(t_f)}{\partial\lambda_1(0)} & \cdots & \dfrac{\partial\lambda_1(t_f)}{\partial\lambda_n(0)} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial\lambda_n(t_f)}{\partial\lambda_1(0)} & \cdots & \dfrac{\partial\lambda_n(t_f)}{\partial\lambda_n(0)} \end{bmatrix}^{-1} \begin{Bmatrix} -\lambda_1(t_f) \\ \vdots \\ -\lambda_n(t_f) \end{Bmatrix}$$

6. *Apply divergence tests. If shooting is diverging, abort procedure and switch to another control approach.*
7. *Repeat from 2.*
8. *Compute the first optimal command using w_o, λ_o and the stationary condition.*

The variational approach has many drawbacks. It is computationally very extensive. Each iteration requires n+1 integration of the dynamic equations over the prediction horizon and the solution of set of n equations and n unknowns. Furthermore, a solution is reached in several iteration steps, making real-time implementations less feasible. A large amount of work is also necessary to set the optimality equations properly. Finally, as for any Newton root finding process, the domain of convergence is relatively small and good initial estimates of the solution are necessary.

Obviously, the problem can be set to obtain a slowly varying optimal solution such that the solution at the previous sampling step is a good estimate of the current one. This also has the advantage to reduce the number of iterations to obtain the solution. Moreover, if the expected optimal solution varies slowly with respect to the sampling rate, then a fixed number of iterations can be used. Then, although the optimal solution has not been completely reached, convergence of the shooting process insures a reasonable output. This fixed iteration step approach is attractive for real-time implementation. Nevertheless, the amount of computation is still extraordinary.

5.2.2 - BFGS : A Variable Metric Method

The optimization process can be reduced enormously by solving the equivalent discrete-time problem defined in section 5.1.3. With this setting, the penalizing functional becomes a function of vectors evaluated using a recursive estimator of the dynamics. Under these circumstances, the dynamically constrained optimization problem, as defined in the variational formulation, is converted to an unconstrained optimization problem. This conversion eliminates the need for integrating the costate equations and can be solved using standard efficient methods. Here, the mean-level MPC control approach is used where the control command is assumed constant over the entire prediction horizon. This basic assumption results in the minimization of a single-valued function of N variables (N is the number of DOF of the robot). The algorithm suggested is the following.

Algorithm 5.2 - BFGS with approximate line minimization

```
1. Select an initial estimate of the optimal torque vector ${}^{i}\tau_k^{*}=\tau_{k-1}^{*}$
2. Use the recursive estimator defined by equations 5.32 and 5.33 to evaluate ${}^{I}Y_k$, the output
   estimate for the $i^{st}$ iteration of sample k, using $y_k$ to set the initial conditions.
3. Compute the reference ${}^{I}r_k$ and the cost estimate ${}^{I}\Phi_k$, for the $1^{st}$ iteration of sample k.
4. Numerically compute the first estimate of the gradient of the cost function ${}^{I}\nabla_{\tau}\Phi_k$.
```

```
5. Initialize the estimated Hessian matrix of the cost function to an identity matrix $^{1}\boldsymbol{H}=\boldsymbol{I}$.
6. Use the gradient as the initial search direction $^{1}s = -^{1}\nabla_{\tau}\Phi_{k}$.
7. While not properly converged or not reached maximum number of iterations
    8. Perform approximate minimization along the line defined by $^{i}s$ to get $^{i+1}\tau$.
    9. Use the recursive estimator defined by equations 5.32 and 5.33 to evaluate $^{i+1}Y_{k}$.
    10. Compute the reference $^{i+1}r_{k}$ and the cost estimate $^{i+1}\Phi_{k}$.
    11. Numerically compute the estimate of the gradient $^{i+1}\nabla_{\tau}\Phi_{k}$
    12. Use the BFGS equation to update the cost function Hessian matrix $^{i+1}\boldsymbol{H}$
            $^{i-1}\boldsymbol{H} = {}^{i}\boldsymbol{H} + \left(1 + \frac{\gamma^{t}\,{}^{i}\boldsymbol{H}\gamma}{\delta^{t}\gamma}\right) - \left(\frac{\delta\gamma^{t}\,{}^{i}\boldsymbol{H} + {}^{i}\boldsymbol{H}\gamma\delta^{t}}{\delta^{t}\gamma}\right)$        (5.43)
        where $\delta = {}^{i+1}\tau + {}^{i}\tau$ and $\gamma = {}^{i+1}\{\nabla_{\tau}\Phi\} - {}^{i}\{\nabla_{\tau}\Phi\}$.
    13. Update the search direction to $^{i+1}s = {}^{i}s - {}^{i+1}\boldsymbol{H}\,{}^{i+1}\nabla_{\tau}\Phi_{k}$.
14. Apply the optimal solution $\tau^{*}_{k}$
```

This approach reduces enormously the number of operations for solving the problem since no system of n equation and n unknowns has to be solved in real-time. The implementation is schematically represented in figure 5.1.

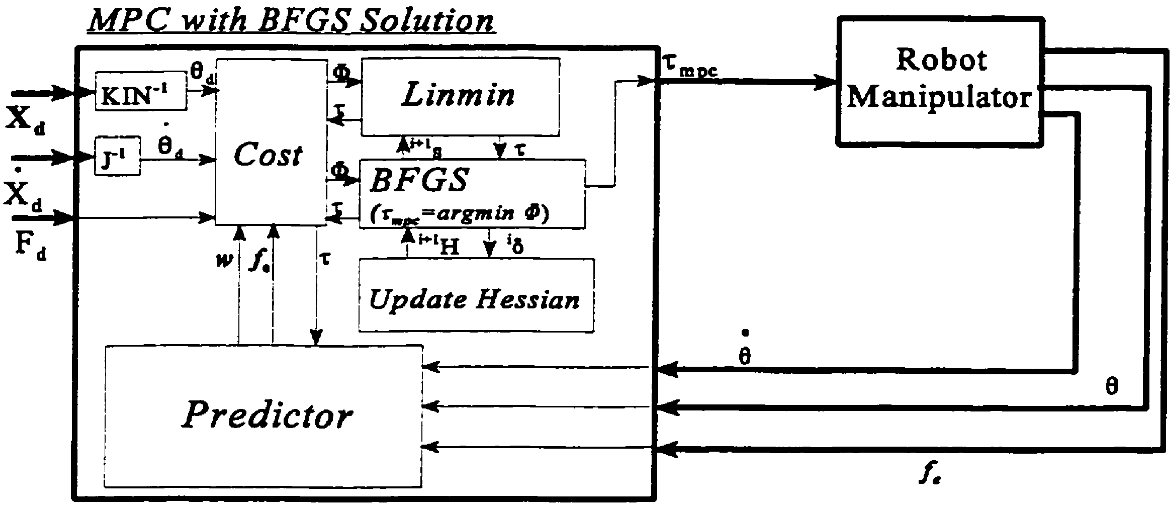


Figure 5.1 - The BFGS Solution

As shown, the feedback mechanism lies in the use of the measurements to set the initial conditions for the predictor. A line minimization is done along the direction s and the resulting optimal torque (compared with the previous one), is used to update the Hessian matrix and compute the next search direction. The iteration process is continued until a convergence has occurred on τ.

5.2.3 - MPC with Partially Linearized Dynamics

The number of arithmetic operations to be performed at every step can be significantly reduced if the known nonlinearities resulting from Coriolis accelerations, centrifugal accelerations and joint friction are compensated directly. Formally, if the control law is specified as

$$\tau = \hat{\tau}_f + \hat{N}(q,\dot{q}) + \hat{G}(q) + \tau_{mpc} \tag{5.44}$$

where the notation ^ indicates "the estimate of", and where τ_{mpc} represents the part of the control resulting from the MPC, then the robot dynamics apparent to the MPC controller are simplified to

$$\ddot{q} = M^{-1}(q)\left(J_c^t f_e + \tau_{mpc}\right) \tag{5.45}$$

reducing how much computation to execute in the recursive predictor (eq. 5.32 and 5.33). The new partially linearized dynamics is still given by equations 5.32 and 5.33, but with the simplified components

$$\begin{aligned} f(w) &= \begin{Bmatrix} \dot{q} \\ 0 \end{Bmatrix} \\ L(w) &= \begin{bmatrix} 0 \\ M(q)^{-1} J_c(q)^T \end{bmatrix} \\ \alpha(w,\tau) &= \begin{Bmatrix} 0 \\ f_e(q,\dot{q},\tau) \end{Bmatrix} \\ g(w) &= \begin{bmatrix} 0 \\ M(q)^{-1} \end{bmatrix} \end{aligned} \tag{5.46}$$

The implementation of this approach is shown in figure 5.2.

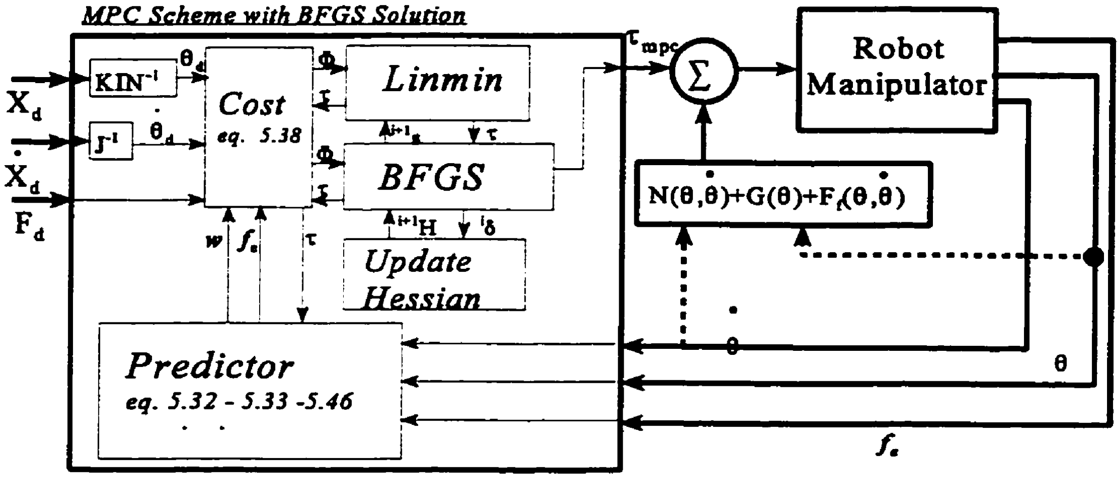
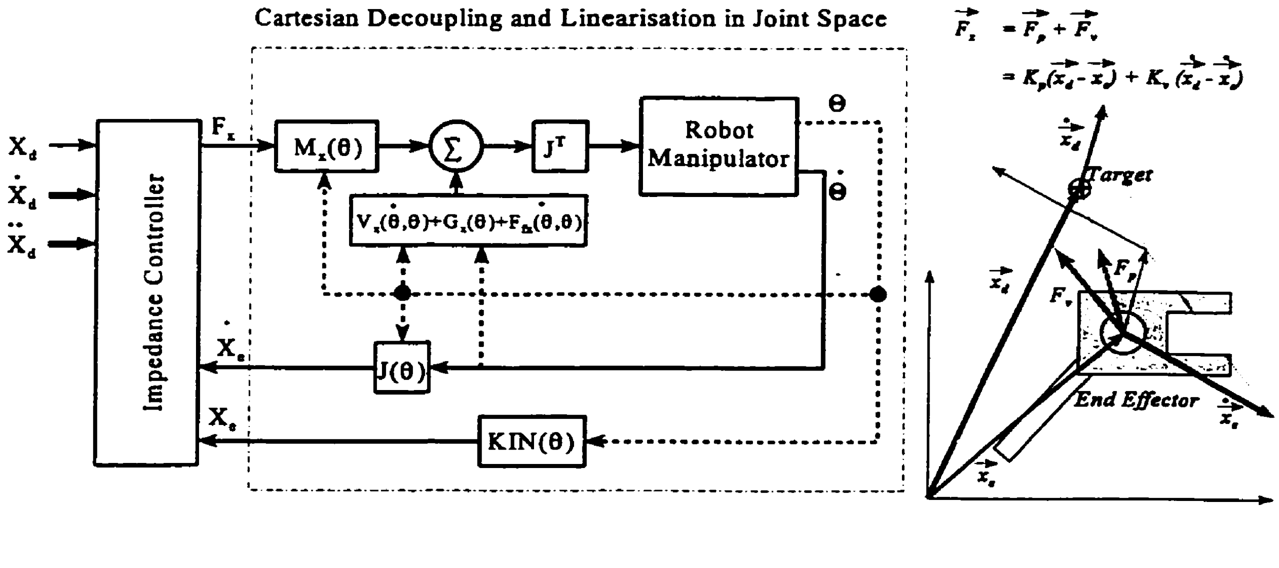


Figure 5.2 - The BFGS Solution with Partial Linearization

5.2.4 - A Feasible Sub-optimal Solution for Robot Control

A major drawback for many optimal control schemes is their unnecessary search of exact solutions to the mathematical problem. In practice, the control design models are only accurate within certain limits and exact mathematical solutions are practically impossible to achieve. For this reason, control theorists have developed new synthesis methods where minimization is based on the sensitivity of the close loop system to modelling errors, and not strictly on time histories. In fact, the LQG/LTR (Linear Quadratic Gaussian Regulator/ Loop Transfer Recovery) design approach proved that a suboptimal solution based on the designer common sense is more likely to be successful than the exact optimal solution. In the LQG/LTR controller, the optimal solution obtained from the LQG formulation is corrected to recover the proper stability margins.

The same drawback applies to the MPC approach presented here. By selecting a solution not optimal in the sense of the control problem of section 5.1, but resulting in another interesting feasible solution, a more desirable control law can be obtained. The solution proposed in this section is based on the analysis of impedance control where the manipulator is linearized and decoupled in Cartesian Space through joint measurements (see figure 5.3) and where joint torques are applied to provide a pre-specified impedance of the end effector.



(a) (b)

Figure 5.3 - Impedance Control

Impedance control cannot, however, easily deal with contact transition since different types of impedance are necessary for force and position control. However, the concept of using knowledge about the desired Cartesian motion to modulate the joint torques is interesting. In nonlinear MPC, most of the effort is spent defining a *search direction* (in the joint torque space) that will result in an overall minimal cost. Due to the nature of the robot dynamics and to the knowledge of the reference force/path trajectory, information about the direction of the control command is readily available. This information can be used to define a search direction along which the cost function could be minimized. Although a global minimum is not obtained, the resulting motion can be satisfactory.

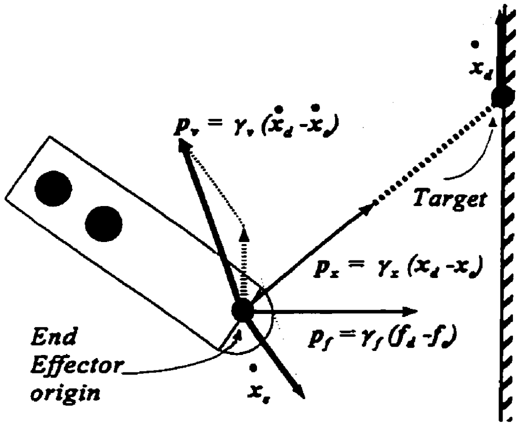

$\dot{x}_d$
$p_v = \gamma_v(\dot{x}_d - \dot{x}_e)$
Target
$p_x = \gamma_x(x_d - x_e)$
$p_f = \gamma_f(f_d - f_e)$
End Effector origin
$\dot{x}_e$

Figure 5.4 - Search Direction

One of the solution proposed in this thesis uses the trajectory information in the follwing manner. First, a vector **p** is defined

$$p = \frac{\gamma_x(x_d - x_e) + \gamma_v(\dot{x}_d - \dot{x}_e) + \gamma_f(f_d - f_e)}{|\gamma_x(x_d - x_e) + \gamma_v(\dot{x}_d - \dot{x}_e) + \gamma_f(f_d - f_e)|} \tag{5.47}$$

representing the weighted direction in which the control commands should act. In equation 5.47, γ_x is the position weight, γ_v is the velocity weight and γ_f is the weight on the force. Then, this Cartesian direction is mapped to the joint space by doing

$$s = \frac{J_c(q)\, p}{|J_c(q)\, p|} \tag{5.48}$$

which can be used as the search direction for the solution. The control law is then defined as

$$\tau = \hat{\tau}_f + \hat{N}(q,\dot{q}) + \hat{G}(q) + \tau_{mpc} s \tag{5.49}$$

and the resulting partially linearized dynamics to solve is

$$\ddot{q} = M^{-1}(q)\left(J_c(q)^t\, f_e(q,\dot{q}) + \tau_{mpc} s\right) \tag{5.50}$$

The MPC problem now becomes a single-valued function minimisation, independent of the number of DOF of the manipulator considered, that has more chance to be solved in real-time. The implementation is shown in figure 5.5.

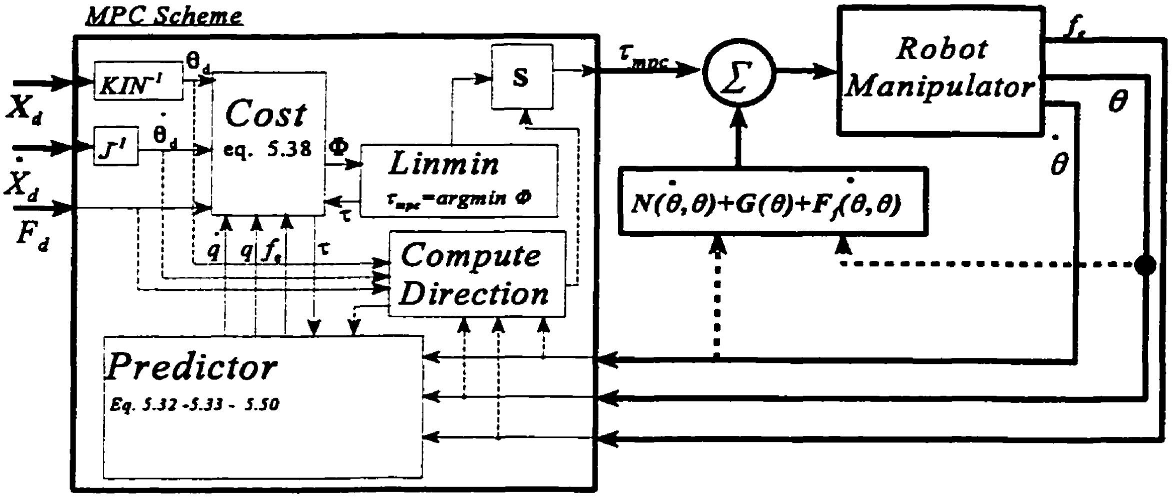


Figure 5.5 - Sub-Optimal MPC

In equation 5.50, while f_e is discontinuous, the terms *M(q)* and $J_c(q)$ have smooth nonlinearities and are functions of position only. If one assumes that the displacement over the prediction horizon is small and generates only small variations in the entries of *M(q)* and $J_c(q)$, then the solution obtained setting $M(q)=M(q_o)$ and $J_c(q)=J_c(q_o)$ can be valid around q_o. Such a local solution can also be used to reduce the number of computations to be done at each iteration step.

5.2.5 - Model Predictive Impedance Approach

So far, the computations for the optimal solution were done in joint space. The Cartesian reference trajectory was translated to a joint space reference trajectory using the inverse kinematics. Since position measurements are done in joint space, it appeared to be a natural approach. However, given that the major uncompensated discontinuity is in Cartesian space (contact force), the dynamics in Cartesian space could be an interesting aspect to consider. Recalling equation 2.11

$$M_x(q)\ddot{x}_e + N_x(\dot{q},q) + G_x(q) = J^{-T}\left(\tau - \tau_f(\tau,\dot{q},q)\right) + f_e \tag{5.51}$$

where

$$\begin{aligned} M_x(q) &= J^{-T}(q)M(q)J^{-1}(q) \\ N_x(q,\dot{q}) &= J^{-T}(q)\left(N(q,\dot{q}) - M(q)J^{-1}(q)\dot{J}(q,\dot{q})\dot{q}\right) \\ G_x(q) &= J^{-T}(q)G(q) \end{aligned} \tag{5.52}$$

If the control torque is defined as

$$\tau = J^{T}(q)\left(\hat{\tau}_f + \hat{N}_x(q,\dot{q}) + \hat{G}_x(q) + \hat{M}_x(q)\lambda_{mpc}\right) \tag{5.53}$$

then the close-loop dynamics becomes

$$\ddot{x}_e = \lambda_{mpc} + M_x(q)^{-1}f_e \tag{5.54}$$

In this formulation, everything but the contact force portion of the dynamics has been decoupled in Cartesian space. Hogan[28] has proposed an approach to decouple the contact force when it is measured. Using his linearization scheme, the MPC control law would be

$$\tau = J^{T}(q)\left(\hat{\tau}_f + \hat{N}_x(q,\dot{q}) + \hat{G}_x(q) + \hat{M}_x(q)M_o^{-1}\lambda_{mpc} - f_e + \hat{M}_x(q)M_o^{-1}f_e\right) \tag{5.55}$$

resulting in the close-loop dynamics

$$\boldsymbol{M_o \ddot{x}_e} = \lambda_{mpc} + \boldsymbol{f_e} \tag{5.56}$$

where decoupling with respect to the end effector force (considered as an exogenous force) is achieved. In reality, however, the end effector force model is highly coupled and perfect decoupling is never achieved. In addition, the force sensor dynamics and limited precision would require the definition of a very robust impedance to maintain good stability margins. Consequently, Hogan's linearization scheme will not be considered in this thesis.

As in section 5.2.4, the equation 5.46 can be used to define the direction of the impedance force needed. The control law is then

$$\tau = \boldsymbol{J}^T(\boldsymbol{q})\left(\hat{\tau}_f + \hat{N}_x(\boldsymbol{q},\dot{\boldsymbol{q}}) + \hat{G}_x(\boldsymbol{q}) + \boldsymbol{M}_x(\boldsymbol{q})\, p\, \lambda_{mpc}\right) \tag{5.57}$$

then the close-loop dynamics becomes

$$\boldsymbol{\ddot{x}_e} = \boldsymbol{p}\, \lambda_{mpc} + \boldsymbol{M}_x(\boldsymbol{q})^{-1} \boldsymbol{f_e} \tag{5.58}$$

With this new definition in Cartesian space, the predictor model (equations 5.31 and 5.32) has changed considerably. The new predictor becomes

$$w_{k+i} = w_k + \sum_{j=0}^{i-1} \left[f_{k+j} + L_{k+j}\ \alpha_{k+j} + g_{k+j} \lambda_{mpc_{k+j}} \right] T \tag{5.59}$$

$$y_k = h_k = \begin{Bmatrix} x_{e_k} \\ \dot{x}_{e_k} \\ f_{e_k} \end{Bmatrix} = \begin{Bmatrix} \Gamma(q_k) \\ J(q_k)\dot{q}_k \\ f_{e_k} \end{Bmatrix} \tag{5.60}$$

where

$$w = \begin{Bmatrix} x_e \\ \dot{x}_e \end{Bmatrix}$$

$$f(w) = \begin{Bmatrix} x_d \\ 0 \end{Bmatrix}$$

$$L(w) = \begin{bmatrix} 0 \\ M_x^{-1} \end{bmatrix} \tag{5.61}$$

$$\alpha(w,\tau) = \begin{Bmatrix} 0 \\ f_e(q,\dot{q},\tau) \end{Bmatrix}$$

$$g(w) = \begin{bmatrix} 0 \\ p \end{bmatrix}$$

The implementation of the predictive impedance controller is shown in figure 5.6.

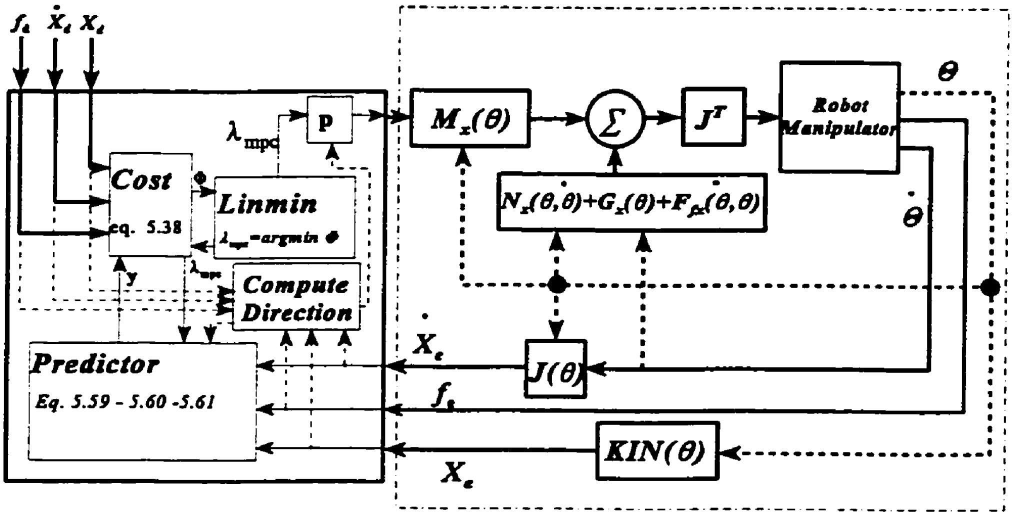


Figure 5.6 - Model Predictive Impedance Control

5.2.6 - The Extended Model Predictive Impedance Approach

In the dynamics described in equation 5.58, the entries to the inertia matrix (and its inverse) are smooth in **q**. For very short prediction horizons, the variation in the entries of $\mathbf{M}_x$ can be relatively small. To improve the real-time applicability of the control method, one can assume that $\mathbf{M}_x$ is not varying over the horizon and is taken as $\mathbf{M}_x(\mathbf{q}_o)$ where $\mathbf{q}_o$ represents the actual measurement of the joint angle.

5.3 - Analysis

The problem formulated in section 5.1 is the nonlinear MPC problem. The existence of a solution and the ability for the solution to control the motion of the robot with frictional/unilateral are investigated in the following sections.

5.3.1 - Local Existence of a Motion

Obviously, the motion exists for the physical system. However, the existence of a solution at any physically realizable state is an essential property for the model. This is particularly true when the model is used in a model-based control method.

In the standard theory of ordinary differential equations, the local existence and unicity of a continuous solution to the general problem

$$\dot{\boldsymbol{x}}_s = \boldsymbol{f}_s(\boldsymbol{x}_s(t), \boldsymbol{u}(\boldsymbol{x}_s(t), t), t), \qquad \boldsymbol{x}_s(t_0) = \boldsymbol{x}_0 \tag{5.62}$$

valid in a neighbourhood N of $(\mathbf{x}_s(t_o), t_o)$, is verified by the sufficient conditions that f_s is continuous in $\mathbf{x}_s$ and t, and that the Lipschitz condition

$$\|\boldsymbol{f}_s(\boldsymbol{x}_1) - \boldsymbol{f}_s(\boldsymbol{x}_2)\| \leq L\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \tag{5.63}$$

be observed for all $\mathbf{x}_1, \mathbf{x}_2 \in N$. Obviously, the model of the robot manipulator with frictional/unilateral contact, presented in table 5.1, does not verify those conditions for every robot state. For example, when the motion of the contact point in the tangential direction of the surface comes to a rest, the friction force may take three different values ($F_f \in \{\mu_k P(d), \mu_s P(d), \mathbf{F}_{cq} \cdot e_t\}$) depending on τ.

For boundary points where the discontinuities occur, the notion of GDS introduced in section 3.3.5 is less restrictive. For example, assuming that the control torque command $\tau\in$U where U is a compact set (this is usually true by construction), then the attainability function **F** defined as

$$F(w(t_0),t_0,t_f) \triangleq \{w(t) \mid \dot{w} = f(w) + L(w)f_e(w) + G(w)\tau, \quad w(t_0) = w_0, \quad \tau\in U, \quad t\in[t_0,t_f]\} \quad (5.64)$$

represents all the possible trajectories from $\mathbf{w}_o$. Similarly, the contingent set **E** defined as

$$E(w_0) \triangleq \{\dot{w}(t) \mid \dot{w} = f(w_0) + L(w_0)f_e(w_0) + G(w_0)\tau, \quad \tau\in U\} \quad (5.65)$$

represents all the possible state derivatives achievable at a point $\mathbf{w}_0$. According to theorem 3.2, a smooth motion $\mathbf{w}$(t)$\in F$ exists in a δ-neighbourhood $N_\delta(w_o,t_o)$ if the set $E(w_0)$ is compact, convex and upper semi-continuous in N_δ. The compactness of E over $N_\delta(w_o)$ is easily verified. Since all the discontinuous terms of the dynamic equation have finite discontinuities, and since the set U is compact, then **E** is compact. The analysis of the convexity of **E** requires the following definitions.

Definition 5.1 - Convex Set

The set B such that for any $x_1,x_2\in B$, the line segment joining x_1 and x_2 is in B , is called a convex set. Lets define $L(\theta)$ as

$$L(\theta) = \{x(\theta) \mid x(\theta) = \theta x_1 + (1-\theta)x_2, \quad \theta\in[0,1]\} \quad (5.66)$$

If $L(\theta)\in B$ $\forall x_1,x_2\in B$, then B is convex.

Concerning the convexity of contingent sets, if all the state derivatives between two possible state derivatives (joint accelerations) are achievable by at least one command τ in U, then the contingent set E is convex. For the discontinuities encountered in the models of table 5.1, this can be interpreted as a condition requiring that the command τ be capable of counteracting any discontinuous changes in the dynamic equations. This is similar to the condition for the controllability of a system with friction, requiring the set of achievable control commands to include at least what is required to counteract the friction force. As stated by Gutmann[9] and Mills[24], this condition can be considered satisfied by construction.

The upper semi-continuity (USC) is the critical condition. To investigate to USC of E, the notion of a USC set is needed.

Definition 5.2 - Distance Between a Set and a Point

The distance $d(x,A)$ between a point $x\in\mathbb{R}^n$ and a set $A\subset\mathbb{R}^n$ is defined as

$$d(\boldsymbol{x},A) = inf\{\|\boldsymbol{x}-\boldsymbol{y}\|,\ \ \boldsymbol{y}\in A\} \tag{5.67}$$

Definition 5.3 - Separation Between Two Sets

The separation $d^*(A,B)$ between a set $A\subset\mathbb{R}^n$ and a set $B\subset\mathbb{R}^n$ is defined as

$$d(A,B) = sup\{d(\boldsymbol{x},B),\ \ \boldsymbol{x}\in A\} \tag{5.68}$$

The definition 5.2 is intuitive. The distance between a point and a set is the smallest distance between the point and any point of the set. The notion of separation requires more attention. The separation is a measure of the difference between the two sets A and B. For example, two similar sets sharing the same boundary but in a neighbourhood of one boundary point, have their separation represented by the largest difference between the boundaries of A and B in the neighbourhood of the boundary points that are not shared. The notion of *Separation between two sets* can be used to define upper semi-continuity of a set.

Definition 5.4 - Upper Semi-Continuous Variable Set

A variable set $A(\boldsymbol{x})\in\mathbb{R}^n$ is upper semi-continuous at a point $\boldsymbol{x}_0\in\mathbb{R}^m$ if given any $\epsilon>0$, $\exists\delta>0$ such that if $\boldsymbol{x}\in N_\delta(\boldsymbol{x}_0)$, then $d^*(A(\boldsymbol{x}),A(\boldsymbol{x}_o))<\epsilon$. In other words, $A(\boldsymbol{x})$ is USC at $\boldsymbol{x}_o$ if its separation is continuous there.

The existence of a motion within each continuous structure of the dynamic model exists and is unique. At the switching boundaries, a sufficient condition for the existence of a motion is the continuity of the separation between contingent sets in the neighbourhood of a boundary point.

Definition 5.5 - Discontinuous Dynamical Systems with Norm-Bounded Discontinuity

Consider a dynamical system of the form

$$\dot{\boldsymbol{w}} = \boldsymbol{f}(\boldsymbol{w}) + \boldsymbol{g}(\boldsymbol{w})\boldsymbol{u} + \boldsymbol{L}(\boldsymbol{w})\alpha(\boldsymbol{w}) \tag{5.69}$$

where f, g and L are smooth mappings and where $\alpha:\mathbb{R}^n\to\mathbb{R}$ is a single valued discontinuous function

$$\alpha(\boldsymbol{w}) = \begin{cases} \alpha_1(\boldsymbol{w}) & \textit{if } \boldsymbol{w}\in N_1 = \{\boldsymbol{w}\mid \phi(\boldsymbol{w})>0\} \\ \alpha_2(\boldsymbol{w}) & \textit{if } \boldsymbol{w}\in N_2 = \{\boldsymbol{w}\mid \phi(\boldsymbol{w})\le 0\} \end{cases} \tag{5.70}$$

Let $\bar{N} = \{w \mid \phi(w)=0\}$ and $N_\delta(w_o)$ be a δ-neighbourhood of w_o If

$$\begin{aligned}
\rho_1 &= \sup_{w_0 \in \bar{N}} \; \sup_{w \in N_\delta \cap N_1} |\alpha_1(w)| < \infty \\
\rho_2 &= \sup_{w_0 \in \bar{N}} |\alpha_2(w_0)| < \infty \\
\rho_3 &= \inf_{w_0 \in \bar{N}} \; \inf_{w \in N_\delta \cap N_1} |\alpha_1(w) - \alpha_2(w_0)| \neq 0
\end{aligned} \tag{5.71}$$

then equation 5.69 represents a discontinuous dynamic system with norm-bounded discontinuity.

Lemma 5.1 - USC Property of the contingent set for a Discontinuous Dynamical Systems with Norm-Bounded Discontinuity

The contingent set of a dynamical system with norm-bounded discontinuity and $\boldsymbol{u} \in B_u = \{\boldsymbol{u} \,|\, \|\boldsymbol{u}\| < \rho_u\}$ *is USC* $\forall w \in \bar{N} = \{ w \,|\, \phi(w) = 0 \}$.

Proof: See Appendix C

Knowing that discontinuities in the model of table 5.1 are bounded, the contingent set defined by (5.65) is USC for every point in the neighbourhood of a discontinuity. Therefore, since **E** is compact, convex and USC, then a motion locally exists.

5.3.2 -Existence of Solution to MPC Problem

The existence of solution to the differential equation representing the dynamic model is crucial for the existence of a solution to the MPC problem defined in section 5.1.2. Obviously, a necessary condition to the existence of a solution to the MPC problem, is the existence of a motion. This justifies the effort put in the preceding section. Still, the existence of a motion is not a sufficient condition for the existence of a solution to the nonlinear MPC problem.

Obviously, the penalty functional defined for the nonlinear MPC problem (equation 5.28) can be non smooth in time, given the non smooth nature of the dynamic constraints (equations 5.23 and 5.25). As a result, the nonlinear MPC problem as stated here is a non smooth dynamic optimization problem. The topic of non smooth optimization is relatively young. The conditions for the existence of a minimum to a non differentiable function f(x) is usually assessed using convex analysis (see

Fletcher [98]). This topic is introduced by the following definitions.

Definition 5.5 - Convex function

Consider a convex set $K \in \mathbb{R}^n$. A function $f:\mathbb{R}\rightarrow\mathbb{R}^n$ for which

$$f\big(\theta x_1 + (1-\theta)x_2)\big) \leq \theta f(x_1) + (1-\theta)f(x_2) \tag{5.72}$$

for any $x_1, x_2 \in K$ is a convex function over K

Definition 5.6 - Subdifferential

Consider a convex set $K \in \mathbb{R}^n$. The set defined as

$$\partial f(x) \triangleq \{g \,|\, f(x+\delta) \geq f(x) + \delta' g, \quad \forall x+\delta \in K\} \tag{5.73}$$

is called the subdifferential of f at x.

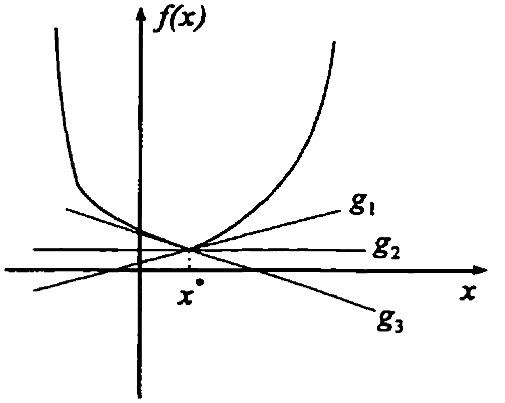


Figure 5.7 - Subdifferential

The convexity of a function over a convex domain, by definition, implies the existence of a minimum on that domain. Definition 5.6 represents, in a way similar to the contingent equation in the GDS theory, the set of all possible gradients (or supporting hyperplanes) in the neighbourhood of **x**. For example, the set of all possible lines supporting the function $f(x)$ at point x^* in figure 5.7 defines the subdifferential of the function at x^*. This definition is a generalization of the gradient of smooth functions to include non smooth convex functions.

Theorem 5.1 - First Order Necessary Condition of minimum at x^*

If x^* is a minimizer of $f(x)$, then $0 \in \partial f(x^*)$.

Theorem 5.1 is essentially a generalization of the conditions for an extrema of a smooth function, requiring that the gradient be null. Since the gradient is not necessarily null for non smooth functions, the set of possible gradient at that point should contain the null gradient.

For dynamically constrained optimization problems with smooth dynamic models, the necessary conditions for an optimal solution are equivalent to those for the existence of a solution of the Two Point Boundary Value (TPBV) problem presented in section 4.3.1. Those conditions were obtained by defining the augmented penalty functional

$$\phi'_c = \phi_c(w_f(t_f),t_f) + v^T\ \Psi(w_f(t_f),t_f) + \int_{t_o}^{t_f} \left[L(w_s,u,\tau) + \lambda^T(f - w_s)\right] \tag{5.74}$$

and by using the Lagrange Principle stating that the constrained optimum of ϕ_c is the unconstrained optimum of ϕ_c'. Using the calculus of variations, the conditions for having the variations $\delta\phi_c$'=0 for any independent variations δw_s, δu, δv, $\delta\lambda$ are extracted leading to the TPBV problem presented in equation 4.29. With non smooth penalty functional, it is sufficient that $0\in\partial\phi_c$' where $\partial\phi_c$' is the subdifferential of ϕ_c'. In fact, the non smooth nature of the dynamics does not influence the solution of the optimization problem, and the only other condition is to verify the convexity of the augmented penalty function ϕ_c'.

If the Lagrangian *L(w, u, t)* has been defined as a convex function by construction (it is a requirement), than an optimal trajectory with a discontinuity in the solution occurring at t=t_d ($t_d\in[t_o,t_f]$) should necessarily verify the following Weierstrass-Erdmann corner conditions

$$\begin{aligned} \lambda(t_d+0^-) &= \lambda(t_d+0^+) \\ H(t_d+0^-) &= H(t_d+0^+) \end{aligned} \tag{5.75}$$

for the penalty functional to be convex at that point. They stipulate that the costate trajectory and the Hamiltonian should be continuous at the time when the discontinuity occurs. Recalling the definition of the Hamiltonian (H=L+$\lambda^T f$) and the fact that L is convex by construction, the continuity in λ implies the Upper Semi Continuity of the contingent set E (equation 5.65). Therefore, if the system is a GDS, then the optimal solution exists.

5.3.3 - Stability and Path Following

Stability is an important property for any control system. MPC has been used, however, for numbers of years without any generalized stability verifications. The cause of this lack of stability results is the nature of MPC itself. In most cases of process control where it has been used, MPC was selected because of its ability to deal with input/output/state constraints as they often appear in chemical processes. For such cases, a typical linear system analysis is not sufficient. The good performances obtained in application did not justify the effort to obtain a complete analysis of stability.

Obviously, a sufficient condition for the path following stability is the existence within the attainability function (equation 5.64) of the reference trajectory for any state within the reference trajectory. More formally, for the system given by

$$\dot{\boldsymbol{w}} = \boldsymbol{f}(\boldsymbol{w}) + \boldsymbol{L}(\boldsymbol{w})\boldsymbol{\alpha}(\boldsymbol{w},\boldsymbol{\tau}) + \boldsymbol{g}(\boldsymbol{w})\boldsymbol{\tau} \tag{5.76}$$

$$\boldsymbol{y} = \begin{Bmatrix} \boldsymbol{w} \\ \boldsymbol{f}_e(\boldsymbol{w}) \end{Bmatrix} = \boldsymbol{h}(\boldsymbol{w}) \tag{5.77}$$

with the reference trajectory $\boldsymbol{r}$(t), if at a time t_o, $y(t_0)=r(t_o)$ and $r(t) \in F(w(t_o), t_o, t)\ \forall w(t_o) \in h^{-1}(r(t_o))$, then the solution of the MPC problem with $T_u=T_h$ gives $y(t) = r(t)$.

This type of reasoning gives some idea about the stability and path following properties of nonlinear optimal control systems. However, the conditions specified for the applicability are practically never true. In fact, stability properties of nonlinear receding finite time optimal control problems are still a subject of research in applied mathematics and no convincing analysis has been found in the literature on optimal control. In the particular case analysed in here, the complexity of the analysis of stability is being increased by the non smooth nature of the dynamic system. Any reasonable proof of stability would necessitate the use of the GDS concept presented in chapter three.

Chapter VI

2 DOF Planar Robot: A Case Study

MPC is an interesting candidate to provide a unified solution to the problem of free motion, contact transition and contact motion control of a robot arm manipulator. The main aspects of the theory and the issues related to control synthesis have been covered in chapter 5. However, the applicability of the method can be best shown through the analysis of a simple case study.

The present chapter is concerned with the application of the proposed method to a real physical system. In the first section, the experimental setup is presented in details. Control design and simulation models for the robot arm are also proposed. The study has been carried out in three stages. First, the parameters of the robot models were evaluated both analytically and experimentally. The analytical parameters were obtained from the design data sheets of the various components of the robot arm and sensors. A Recursive Least Square (RLS) estimate of the robot parameters was also obtained by randomly moving the robot and by identifying its parameters in real-time. In the second stage, a simulator was used to evaluate various solution alternatives. The general structure of the simulator developed for the analysis is presented with its characteristics. The simulation results were then used to select the approach most suitable for real-time implementation. Finally, the best alternative was implemented on the real physical system and the results obtained are presented in details.

6.1 - Description

The objective of the case study is to illustrate nonlinear MPC as a unified method to solve the general control problem presented in chapter 3. To meet this objective, the rigid 2 DOF direct-drive robot of the University of Ottawa's Mechanical Systems Control Laboratory was used. Based on the conceptual design of professor Necsulescu (see Jassemi and Kiguchi [101]), this robot provides a realistic test-bed for verifying the applicability of robot control algorithms.

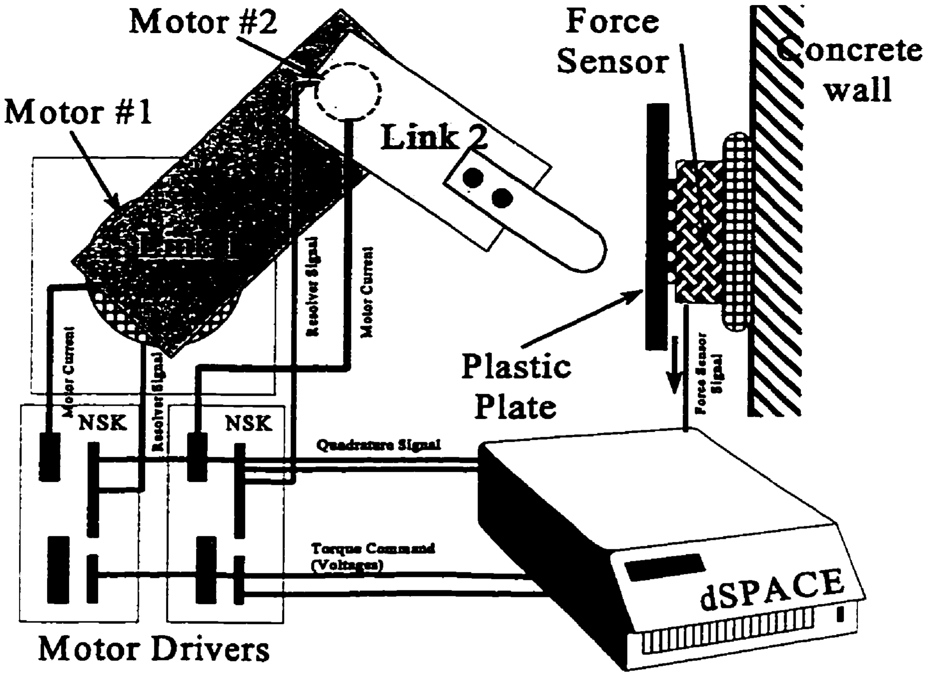


Figure 6.1 - Experimental Setup

6.1.1 - Experimental Setup

The experimental setup is schematically represented in figure 6.1. The robot arms are driven by 2 torque controlled NSK MegaTorque direct-drive motors (ref. [100]). The larger motor, driving link 1, can provide a maximum torque of 90 Nm with linear operation over speeds ranging from 0 to 30 rpm. The smaller motor, for link 2, can linearly deliver a maximum torque of 10 Nm over speeds from 0 to 180 rpm. A pair of current control amplifiers with a 120 Hz bandwidth is used, and the resolution of the delivered torque is 8 bits. Torque ripples are automatically compensated for in the motor driver. The first structural mode of the robot arm is higher than 80 Hz (highly rigid). The dimensions of the robot are shown in appendix D.

For this case study, the contact surface is simply flat wall at a distance x_{wall} from the centre of the first joint. The force is sensed on the environment side using a Barry-Wright FS6-120A 6-axis force sensor. The sensor uses six strain gauge transducers whose bridged signals are filtered at 120 Hz and sampled at 400Hz with a 12-bit resolution. The calibrated data represents six decoupled force/moments with the moments specified about a known fixed point in space. Nominally, the data can be transmitted to a primary control computer through a serial link (RS-232) or can be converted to an analog signal. For this application, the normal contact force is transmitted to the primary control processor (dSPACE) as an analog signal.

The primary control computer is a dSPACE Controller equipped with a TMS320C30 Digital Signal Processor (DSP). Various interfaces are accessed by the DSP via a Peripheral High Speed (PHS) Bus. These include a 16-bit MUX A/D converter board with 32 channels, ten incremental encoder quadrature counters, ten 12-bit D/A converters and a general digital I/O interface. The TMS320C30 can execute 32-bit floating point number operations at speeds up to 33 MFLOPS.

The robot position is measured using the NSK motor's built-in position resolvers. The NSK motor driver module converts the resolver signal into a quadrature that can be sensed by the dSPACE encoder card. The position resolution is of 640,000 steps/rev for motor #1 and 409,600 steps/ rev for motor #2.

6.1.2 - Modelling

The details of the robot dimensions are given in appendix D. The robot model is described using the parameters as they are defined in figure 6.2.

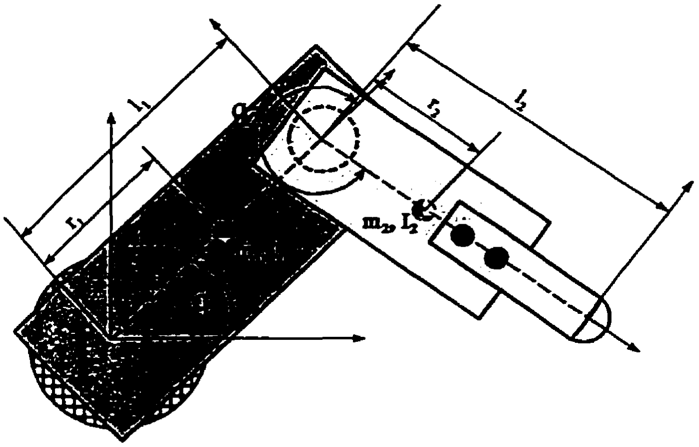


Figure 6.2 - Robot Geometry

The dynamics of the 2 DOF planar robot is given by equation (5.1) with **G(q)=0** and

$$M(q)=\begin{bmatrix} m_1 r_1^2 + m_2(l_1^2 + r_2^2 + 2 l_1 r_2 c_2) + I_1 + I_2 & m_2(r_2^2 + l_1 r_2 c_2) + I_2 \\ m_2(r_2^2 + l_1 r_2 c_2) + I_2 & m_2 r_2^2 + I_2 \end{bmatrix}$$

$$N(\dot{q},q)=\begin{Bmatrix} -m_2 l_1 r_2 s_2 (2\dot{q}_1\dot{q}_2 + \dot{q}_1^2) \\ m_2 l_1 r_2 s_2 \dot{q}_1^2 \end{Bmatrix} \quad J(q)=\begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} + R & l_2 c_{12} + R \end{bmatrix} \tag{6.1}$$

where c_i=cos(q_i), s_i=sin (q_i), c_{12}=cos(q_1+q_2), s_{12}=sin(q_1+q_2), l_i is the length of link i, r_i is the distance between the preceding joint and the centre of mass of the link i. For this model, the centre of mass of each link is assumed to lie on the axis joining the origin of two successive link frames.

Joint friction can be modelled using a static friction model

$$\tau_{f_i}=\begin{Bmatrix} -\tau_{k_i} sgn(\dot{q}_i) & if\ \dot{q}_i \neq 0 \\ \tau_{eq_i} & if\ \dot{q}_i = 0\ \ and\ \ |\tau_{eq_i}| < \tau_{s_i} \\ \tau_{s_i} sgn(\tau_{eq_i}) & if\ \dot{q}_i = 0\ \ and\ \ |\tau_{eq_i}| \geq \tau_{s_i} \end{Bmatrix} \tag{6.2}$$

The term $\tau_{eq,i}$ is obtained from solving

$$M(\dot{q})\ddot{q}^* + N(\dot{q},q) + G(q) = \tau - \tau_{eq} + J^T f_e \tag{6.3}$$

where

$$\ddot{q}_i^* = \begin{cases} \ddot{q}_i & if\ \dot{q}_i \neq 0 \\ 0 & otherwise \end{cases}$$

$$\tau_{eq_i} = \begin{cases} \tau_{k_i} sgn(\dot{q}_i) & if\ \dot{q}_i \neq 0 \\ \tau_{eq_i} & otherwise \end{cases} \tag{6.4}$$

The parameters $\tau_{k,i}$ and $\tau_{s,i}$ represent respectively the kinetic and static level of friction in each joint. This model is valid at every state and command conditions, and includes stiction.

The contact surface is assumed to be defined by x_{wall}-x=0. Therefore, the environment model is

$$S_{env} = \{(x,y) \mid x = x_{wall}, y = 0\} \tag{6.5}$$

and the normal vector **n** is a constant defined by

$$\boldsymbol{n} = \{\ -1,0\ \} \tag{6.6}$$

The penetration distance is obtained directly, without a need of an iterative procedure

$$d = x_e + R - x_{wall} \tag{6.7}$$

Therefore, the contact force is given by

$$f_e = \begin{Bmatrix} K_{env} d \\ F_f \end{Bmatrix} \tag{6.8}$$

where the friction force F_f can be modelled using a static model of the form

$$F_f = \begin{cases} \mu_k K_{env} d\ sgn(\dot{y}_e) & \text{if } \dot{y}_e \neq 0 \\ -F_{eq} & \text{if } \dot{y}_e = 0 \text{ and } |F_{eq}| < \mu_s K_{env} d \\ \mu_s K_{env} d\ sgn(F_{eq}) & \text{if } \dot{y}_e = 0 \text{ and } |F_{eq}| \geq \mu_s K_{env} d \end{cases} \tag{6.9}$$

In equation 6.11, μ_k is the coefficient of kinetic friction, and where the equivalent force F $_{eq}$ represents

$$F_{eq} = \left[J_c^{-T}\left(\tau - \tau_f - G(q) - N(\dot{q},q)\right)\right]_y \tag{6.10}$$

where the symbol $[\bullet]_y$ refers to the y translation component of the force/torque vector. The dynamics of a robot manipulator can be represented linearly in terms of their parameters by grouping parameter terms such as to represent equation 5.1 by

$$\Phi(\ddot{q},\dot{q},q,\tau)^T\ \Theta = \tau \tag{6.11}$$

where the Φ is a nonlinear regression matrix and the Θ is a parameter matrix whose entries are formed by product of parameters such as link masses, inertia, length, friction coefficients, environment stiffness and environment damping. Obviously, the variable structure dynamic system obtained here has different parametric representations, one for each structure of its dynamic model. For the 2 DOF planar robot, the form linearized in terms of the parameters is difficult to obtain for every structure of the model. To evaluate at least the robot link parameters, it is possible to rewrite the dynamics to obtain such a model valid in free motion and when all joint velocities are non zero. This model is obtained by defining

$$\boldsymbol{\Theta} = \begin{bmatrix} m_1 r_1^2 + m_2\left(l_1^2 + r_1^2\right) + I_1 + I_2 \\ m_2 r_2^2 + I_2 \\ m_2 l_1 r_2 \\ \tau_{k_1} \\ \tau_{k_2} \end{bmatrix} \qquad \textbf{(6.12)}$$

and

$$\boldsymbol{\Phi}^T = \begin{bmatrix} \ddot{q}_1 & \ddot{q}_2 & \left(2\ddot{q}_1 + \ddot{q}_2\right)\cos(q_2) - \left(2\dot{q}_1\dot{q}_2 + \dot{q}_1^2\right)\sin(q_2) & sgn(\dot{q}_1) & 0 \\ 0 & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1\cos(q_2) - \dot{q}_1^2\sin(q_2) & 0 & sgn(\dot{q}_2) \end{bmatrix} \qquad \textbf{(6.13)}$$

6.1.3 - The Simulator

A non real-time simulator was developed to test various alternatives to the solution of the nonlinear MPC problem. The simulator was coded in C with the ANSI C standards such that the control modules developed for the simulator could be directly implemented on the dSPACE system. The simulator has the following features:

1. The simulation is based on a rigid body dynamic model of the manipulator, including a joint friction model and a frictional unilateral contact model as presented in chapter 2.
2. Integration is performed using the Runge-Kutta-Nystrom algorithm.
3. Data is stored in ASCII format.
4. Robot motion is animated.
5. The contact force is displayed on the animation screen.
6. Distinct models and parameters for control and simulation.
7. Data reduction and post-processing is done using MATLAB.
8. Execution time of the control algorithm is monitored for comparison.

Figure 6.3 shows the animation display of the simulator. It is realistic enough to provide a good evaluation of the different alternatives to solve the nonlinear MPC control problem.

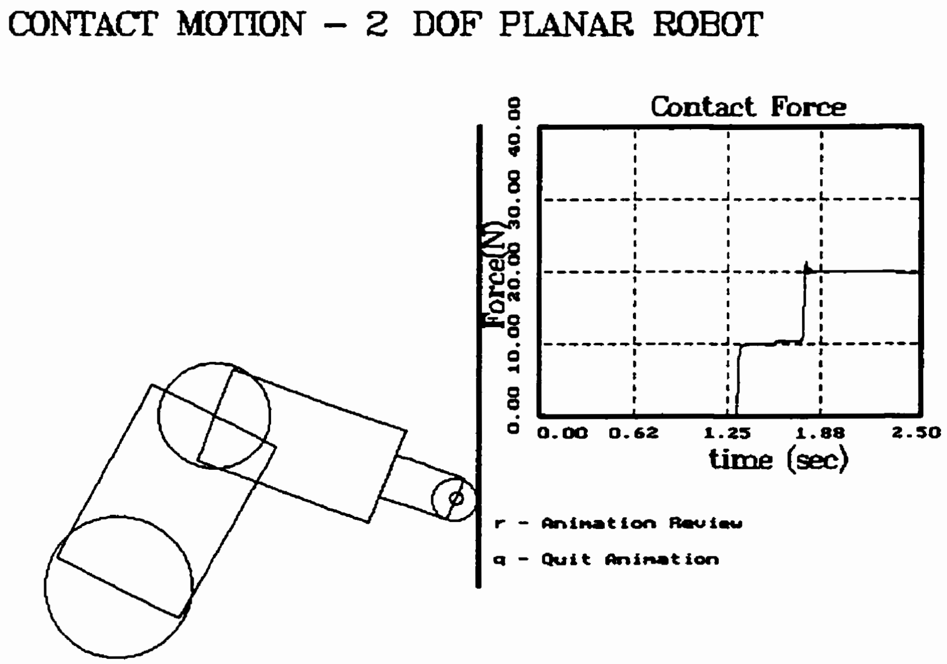


Figure 6.3 - Simulator Interface

6.2 - Parameter Identification

6.2.1 - Nominal Robot Parameters

Nominal parameters were obtained from the design data sheets and drawings. Their precision, however, does not exceed 90%. They are summarized in table 6.1. To obtain a good model for the MPC controller with partial linearization, a parametric identification of the robot was peformed. To achieve this, the nonlinear model was rewritten as a model *linearized in terms of the parameters* as defined in equation 6.11, 6.12 and 6.13. The initial guess on the vector Θ was

$$\boldsymbol{\Theta} = \begin{bmatrix} 0.833\ kg{\cdot}m^2 \\ 0.1084\ kg{\cdot}m^2 \\ 0.1156\ kg{\cdot}m^2 \\ 0.0 \\ 0.0 \end{bmatrix} \qquad (6.14)$$

Parameter	Value
l_1	0.2195 m
l_2	0.271 m
r_1	0.16 m
r_2	0.13 m
m_1	12.88 kg
m_2	4.05 kg
I_1	0.2 kg m^2
I_2	0.04 kg m^2

Table 6.1 - Robot's nominal parameters

6.2.2 - The Robot Identification Process

The empirical evaluation of the vector $\boldsymbol{\Theta}$ was done by moving the robot manipulator randomly using a Stiffness Controller. The desired Cartesian trajectory was obtained by feeding a white noise through a second order low pass filter with a selected bandwidth (see figure 6.4). The level of the noise was adjusted to obtain a desired Root Mean Square (RMS) amplitude for each coordinate. That way, adjusting the frequency band and the amplitude of the excitation to provide proper and sufficient excitation was relatively easy. The resulting control outputs and measurements was used to evaluate the vector $\boldsymbol{\Theta}$ using a Recursive Least Square (RLS) algorithm(see ref. [103]). The joint accelerations and velocities were obtained by digital filtering of the position signal.

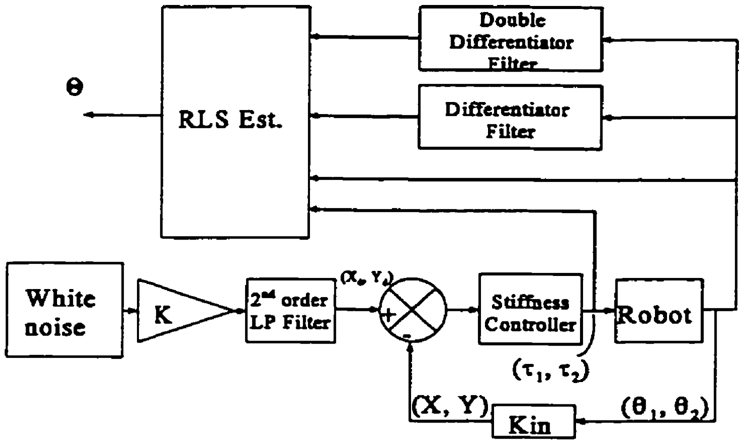


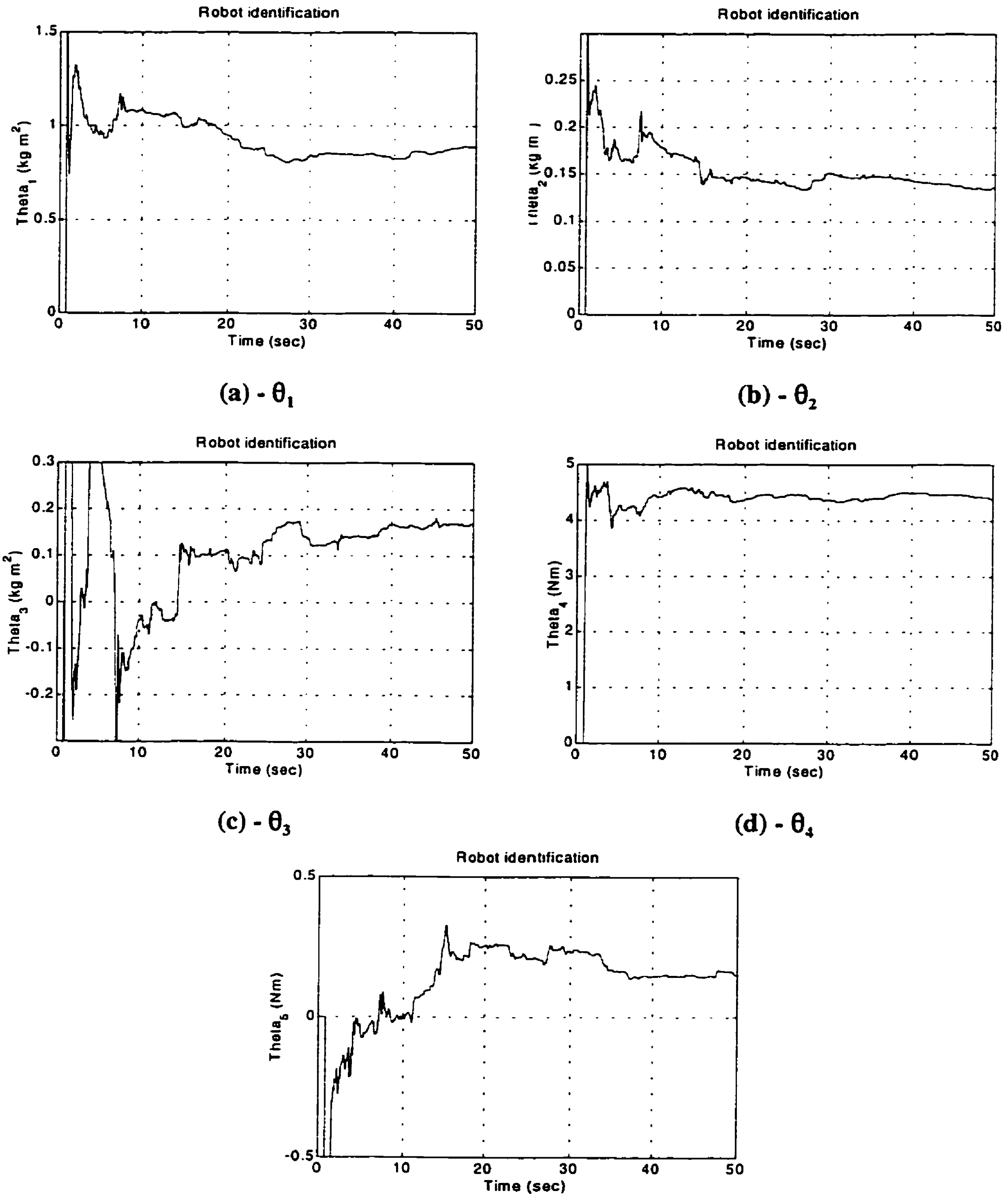
Figure 6.4 - The Identification Process

6.2.3 - Robot Parameters Identification Results

The results are presented in figure 6.5 (a) to (e) in which time histories of the estimated parameters are shown. Asymptotic values are given in the table 5.2.

Parameter	Units	Analytical Value	Identified Value
θ_1	kg m^2	0.833	0.8893
θ_2	kg m^2	0.1084	0.1356
θ_3	kg m^2	0.1156	0.1705
θ_4	Nm	0.0	4.386
θ_5	Nm	0.0	0.01492

Table 6.2 - Comparison Between Identified and Nominal Parameters



(a) - θ_1

(b) - θ_2

(c) - θ_3

(d) - θ_4

(d) - θ_5

Figure 6.5 - Robot Identification Results

6.2.4 - Force Sensor Response

One of the assumptions stated in chapter 3 is that the environment is rigid. In the experimental setup, however, the environment is composed of a thirty (30) centimeter square plate attached to the force sensor at its middle point and will inevitably exhibit dynamics. To establish the dynamic response of the force-torque sensor, a simple impulse response experiment was done. The operation consisted of hitting the force-sensor platform with a small hammer while recording the response. While this simple experiment does not provide proper information about scaling, it nevertheless gives important information about the force-torque sensing apparatus dynamics. Figure 6.6 gives the Impulse Response in terms of the Power Spectral Density. The Sensor/Platform system has a first natural mode at 25 Hz and another at 30 Hz. These are sufficiently high to be neglected, although great care must be taken not to excite them during contact. The high level in the PSD at low frequency is mainly the result of the large offset in the force measurement. During the implementation, this offset must be reduced to a minimum.

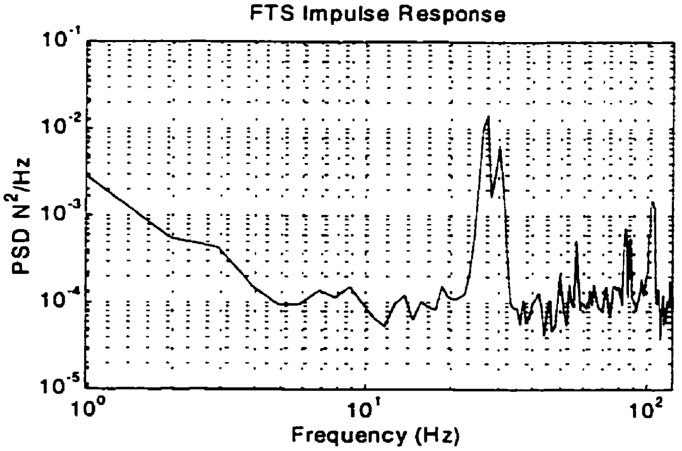
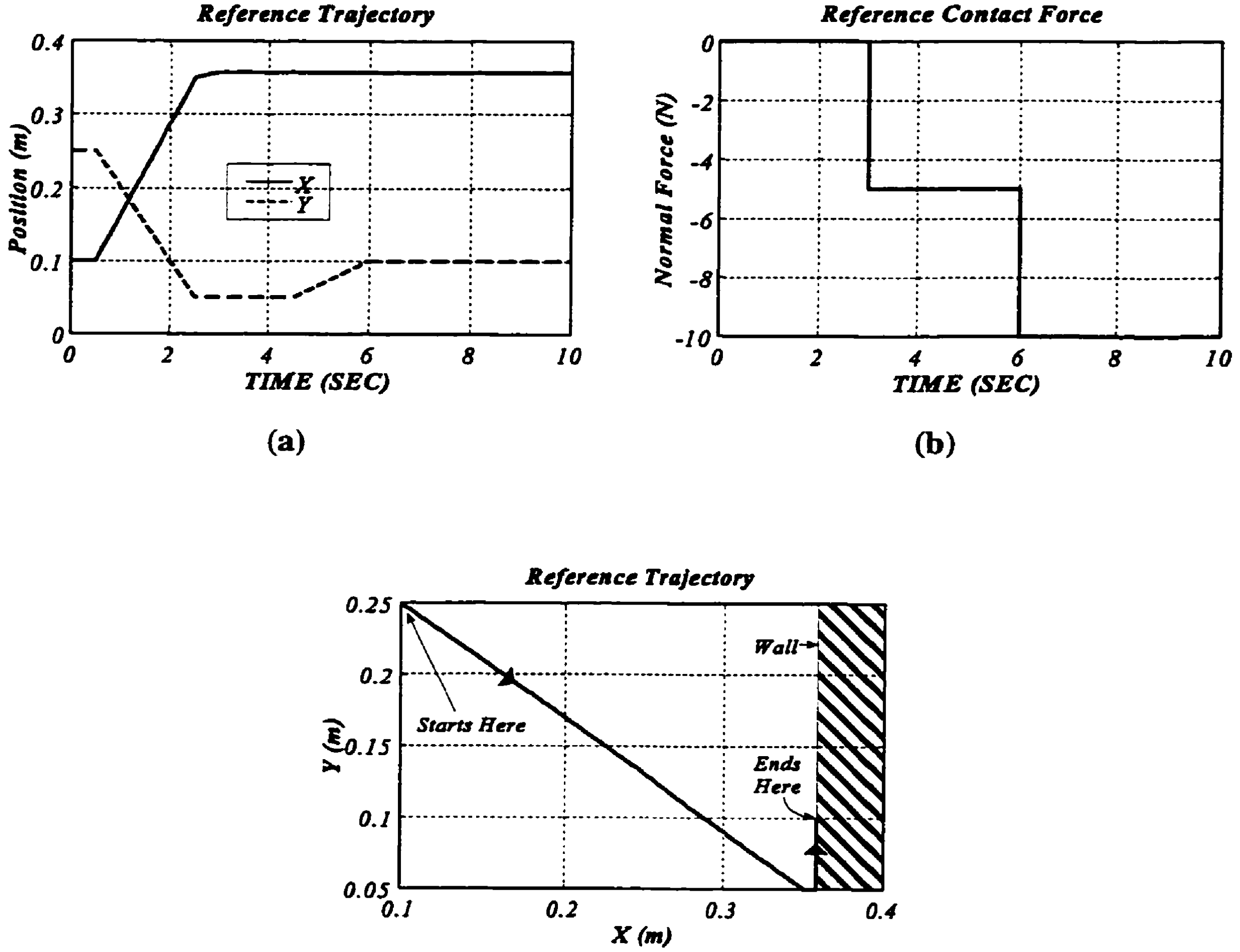


Figure 6.6 - FTS Impulse Response

6.3 - Simulation Results

Few methods to implement nonlinear MPC have been presented in chapter 4 and 5, some of which were clearly stated to have potential to solve the control problem defined in chapter 3. In this section, some of the simulation results obtained applying various possible implementations are presented. The objectives of the simulation analysis were to demonstrate the potential of MPC to solve the problem of impact/contact control and to investigate the real-time applicability the three of the approaches defined in chapter 5. Obviously, with fully deterministic environments, implementing a trajectory generation algorithm adapting the motion for a contact transition is always possible. Therefore, as a part of the simulation, it is shown how nonlinear MPC can deal with roughly defined trajectories and help remove the need for a complex higher level of control. To achieve this, reference trajectories were defined as series of constant velocity segments. For all the simulation results shown in this section, the reference trajectory is as shown in figure 6.7. It was designed to study every aspect of the control problem: free motion control, transition to contact, contact force control and contact motion path/force control.



(a)

(b)

(c)

Figure 6.7-Reference Trajectory

To include modelling errors, the simulation model used the analytical parameters shown in table 5.1, while the control model was set to use the parameters identified experimentally. The 5 to 10% modelling error is common in robot control practice and helps in validating the simulation results. For joint friction, the experimentally identified friction levels were used both for the simulation and control models. Surface friction was modelled using $\mu_k = \mu_s = 0.2$. In all simulations, the integration time step was set to 0.0005 second.

Dealing with contact surface friction is particularly difficult. In the experimental setup, the friction force measurement is not available for control. Two options are available to include its effect in the control law. A surface friction compensation term, based on a static nonlinear state reaction using velocity and normal force measurements, could be added to the feedback linearization component of the control torques. To be practical, very good models of surface friction would be needed for this. A more reasonable approach consists of including the surface friction in the control model used to create the predictor. That way, the apparent compensation of surface friction is dynamic in nature and rely on all feedback signals. This latter approach was selected for all implementations.

6.3.1 - Variational Formulation

The variational formulation leads to the classical solution method for this type of problems. The robot variable structure dynamics result in a Two-Point Boundary Value problem with variable structures as well. It is assumed in this approach that the state vector **w** is

$$w = \{q_1 \;\; q_2 \;\; \dot{q}_1 \;\; \dot{q}_2\}^T \tag{6.15}$$

and the output vector $\mathbf{y}_m$ is

$$y_m = \{q_1 \;\; q_2 \;\; \dot{q}_1 \;\; \dot{q}_2 \;\; [f_e]_N\}^T \tag{6.16}$$

where $[\mathbf{f}_c]_N = (\mathbf{f}_c \cdot \mathbf{n})\mathbf{n}$ represents the normal component of the contact force $\mathbf{f}_c$. If the penalty functional presented in equation 5.27 is used, the state equation of the variable structure system is of the form of equations 5.21 to 5.25 with the details presented in equations 6.1 to 6.12, that is

$$\dot{w} = f(w) + L(w)f_e(w) + G(w)\tau \tag{6.17}$$

with the initial condition defined by

$$w(t_o) = 0 \tag{6.18}$$

and the costate equation is given by

$$\dot{\boldsymbol{\lambda}} = -\left[\frac{\partial(\boldsymbol{y}_m - \boldsymbol{r})}{\partial \boldsymbol{w}}\right] \boldsymbol{Q}_1(\boldsymbol{y}_m - \boldsymbol{r}) - \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}\right]\boldsymbol{\lambda} \qquad \textbf{(6.19)}$$

where

$$\frac{\partial(\boldsymbol{y}_m - \boldsymbol{r})}{\partial \boldsymbol{w}} = \begin{bmatrix} J_{11} & J_{12} & 0 & 0 \\ J_{21} & J_{22} & 0 & 0 \\ -J_{21}\dot{q}_1 - J_{22}\dot{q}_2 & -J_{22}\dot{q}_1 - J_{22}\dot{q}_2 & J_{11} & J_{12} \\ J_{11}\dot{q}_1 + J_{12}\dot{q}_2 & J_{12}\dot{q}_1 + J_{12}\dot{q}_2 & J_{21} & J_{22} \\ \dfrac{\partial[f_e]_N}{\partial q_1} & \dfrac{\partial[f_e]_N}{\partial q_2} & \dfrac{\partial[f_e]_N}{\partial \dot{q}_1} & \dfrac{\partial[f_e]_N}{\partial \dot{q}_2} \end{bmatrix} \qquad \textbf{(6.20)}$$

$$\frac{\partial[f_e]_N}{\partial q_1} = \frac{\partial[f_e]_N}{\partial X_e}\frac{\partial X_e}{\partial q_1} = \begin{cases} 0 & \mathit{if}\ X_e \le 0 \\ -J_{11}K_e & \mathit{if}\ X_e > 0 \end{cases} \qquad \textbf{(6.21)}$$

$$\frac{\partial[f_e]_N}{\partial q_2} = \frac{\partial[f_e]_N}{\partial X_e}\frac{\partial X_e}{\partial q_2} = \begin{cases} 0 & \mathit{if}\ X_e \le 0 \\ -J_{12}K_e & \mathit{if}\ X_e > 0 \end{cases} \qquad \textbf{(6.22)}$$

$$\frac{\partial[f_e]_N}{\partial \dot{q}_1} = 0.0 \qquad \textbf{(6.23)}$$

$$\frac{\partial[f_e]_N}{\partial \dot{q}_2} = 0.0 \qquad \textbf{(6.24)}$$

and where the term $\partial f/\partial w$ is a discontinuous matrix. The stationary condition is given by

$$\tau = -\frac{1}{2}Q_2^{-1}G(w)^T\lambda \tag{6.25}$$

If no extra constraint is applied to the state or to the control signal within the horizon, then the transversality condition becomes

$$\lambda(t_o+T_h)=\mathbf{0} \tag{6.26}$$

Equations 6.17, 6.19 and 6.25, with the conditions expressed in equations 6.18 and 6.26, represent a Two-Point Boundary Value problem. The term $\partial f/\partial w$ in the costate equation can be evaluated numerically such that a numerical solution can be obtained using the algorithm 5.1.

The simulated results for the free motion control are still presented here as a demonstration of the potential for the method. The results are shown in figure 6.8. The sampling frequency was set to 200 Hz and the control parameters were

$$Q_1=\begin{bmatrix} 1.0e5 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0e5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 100.0 \end{bmatrix} \quad Q_2=\begin{bmatrix} 0.03 & 0.0 \\ 0.0 & 0.03 \end{bmatrix} \quad T_h=0.05 \text{ sec} \tag{6.27}$$

A numerical overflow occurred when the impact appeared within the prediction horizon after three (3) seconds of simulation. At that instant, the cost functional grew to values out of the range of numerical stability of the shooting method algorithm, resulting in physical instabilities of the controlled system. The results in free motion still demonstrate the excellent tracking performance that could be achieved with nonlinear MPC.

The difficulties in solving the variational problem appear in the nature of the contact phenomena and the properties of shooting methods. Recalling remark 5 in section 4.3.1, the contact stiffness in the state equations that appears during contact make the unstable costate equations grow more rapidly. The weighting factors could be reduced in an effort to stabilize the costates, but the resulting free motion control would become unsatisfactory. In addition, shooting method algorithms are essentially the application of a Newton zero search method requiring the numerical evaluation of the functions's Jacobian. Such numerical differentiation over wide range of numerical values is numerically ill-conditioned.

Finally, the variational problem was solved in no less than 2.0 seconds using the simulator code on a Pentium 100MHz. This is not sufficient for real-time implementation on common computer platforms.

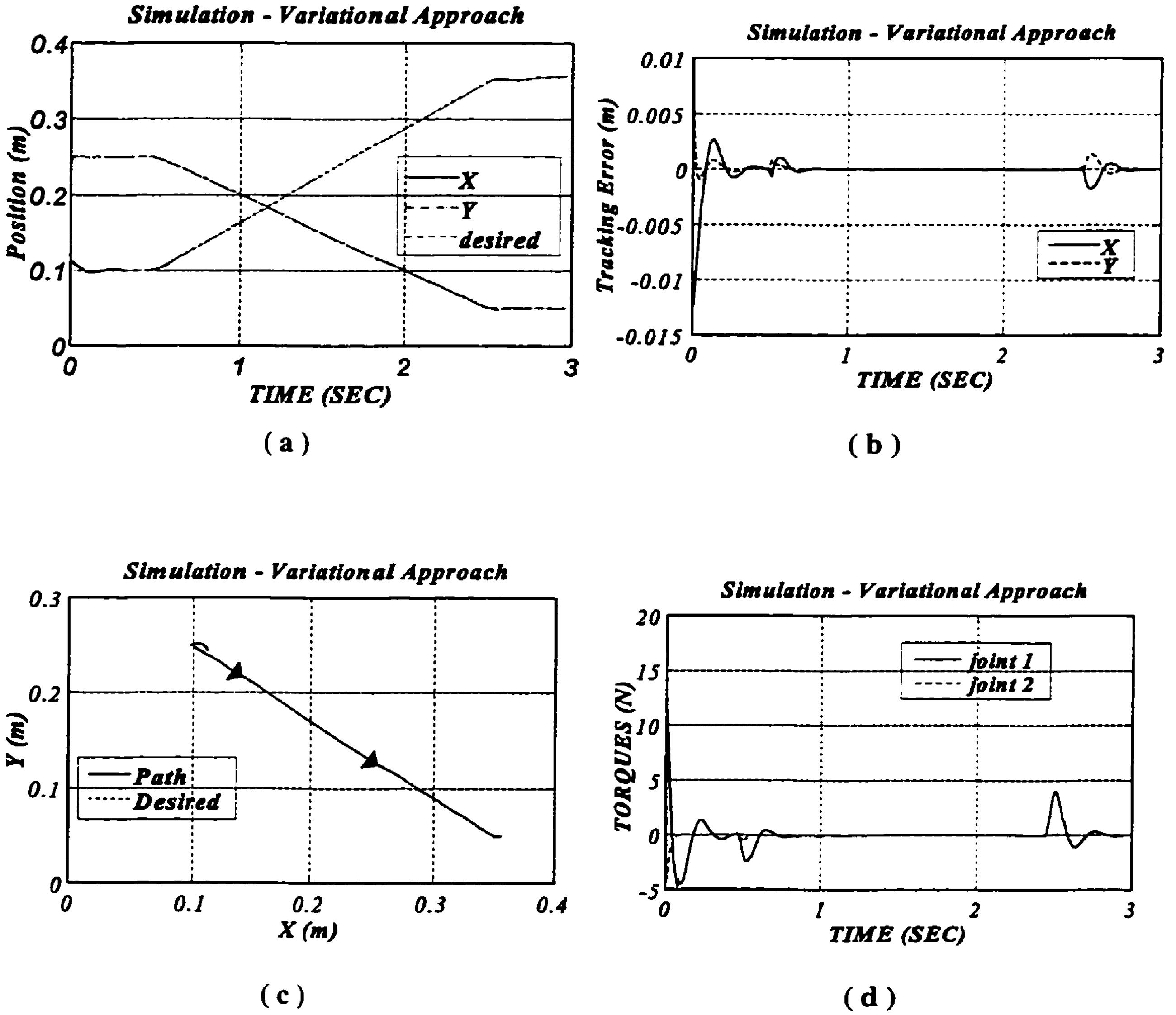


Figure 6.8 - Simulated Response - Variational Approach

6.3.2 - BFGS: A Variable Metric Approach

Using partial linearization(equation 5.44), the equation of motion to be used for the MPC solution can be reduced as in equation 5.45. Using this model and the state vector defined in equation 6.15, a ith-step-ahead nonlinear predictor (as in equation 5.33) can be built and used to compute the cost functional $\Phi_k = \Phi_k(\mathbf{y}_k, \mathbf{R}_k, \mathbf{T}_k)$ where $\mathbf{y}_k$ is the known measurement, $\mathbf{R}_k$ is the known reference to track over the horizon and $\mathbf{T}_k$ is the unknown sequence of MPC torques ($\mathbf{T}_k=\{\tau_{k+1}, \tau_{k+2}, ..., \tau_{k+hp}\}$). The minimization of Φ is the minimization of a single-valued function of $2N_h$ variables. If the mean-level MPC controller is used, that is if the constraint defined by equation 5.29 is applied for T_u=0, then $\Phi_k = \Phi_k(\mathbf{y}_k, \mathbf{R}_k, \tau_k)$ which results in the minimization of a singled-valued function of 2 variables. Algorithm 5.2 presented in chapter 5 was used to solve this optimization problem in real-time. The BFGS algorithm implemented for this application is a modified version of the algorithm provided *Numerical Recipe in C* [97]. The modifications concentrated mainly on dynamic memory allocation and algorithm failure mechanisms for real-time implementation. Within the BFGS algorithm, the line minimization was performed using an approximate line minimization using a line search algorithm with quadratic interpolations [97].

The simulation results are shown in figure 6.9. Excellent tracking results were obtained. Just before 3.0 sec, the upcoming impact with the environment appeared in the prediction horizon and the controller allowed the tracking error to increase momentarily to avoid excessive impact loads. For these results, the control parameters are

$$\boldsymbol{Q}_1 = \begin{bmatrix} 1.0e4 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0e6 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 10.0 \end{bmatrix} \qquad \boldsymbol{Q}_2 = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \qquad T_h = 0.04 \text{ sec} \qquad (6.30)$$

These simulation results alone are sufficient to prove that nonlinear MPC could solve the problem of impact/contact motion control of a robot arm. On the other hand, they also show that although the execution time has been reduced by a factor of twenty with respect to the variational approach, the algorithm is still computationally very demanding and virtually impossible to implement in real-time on a common computer.

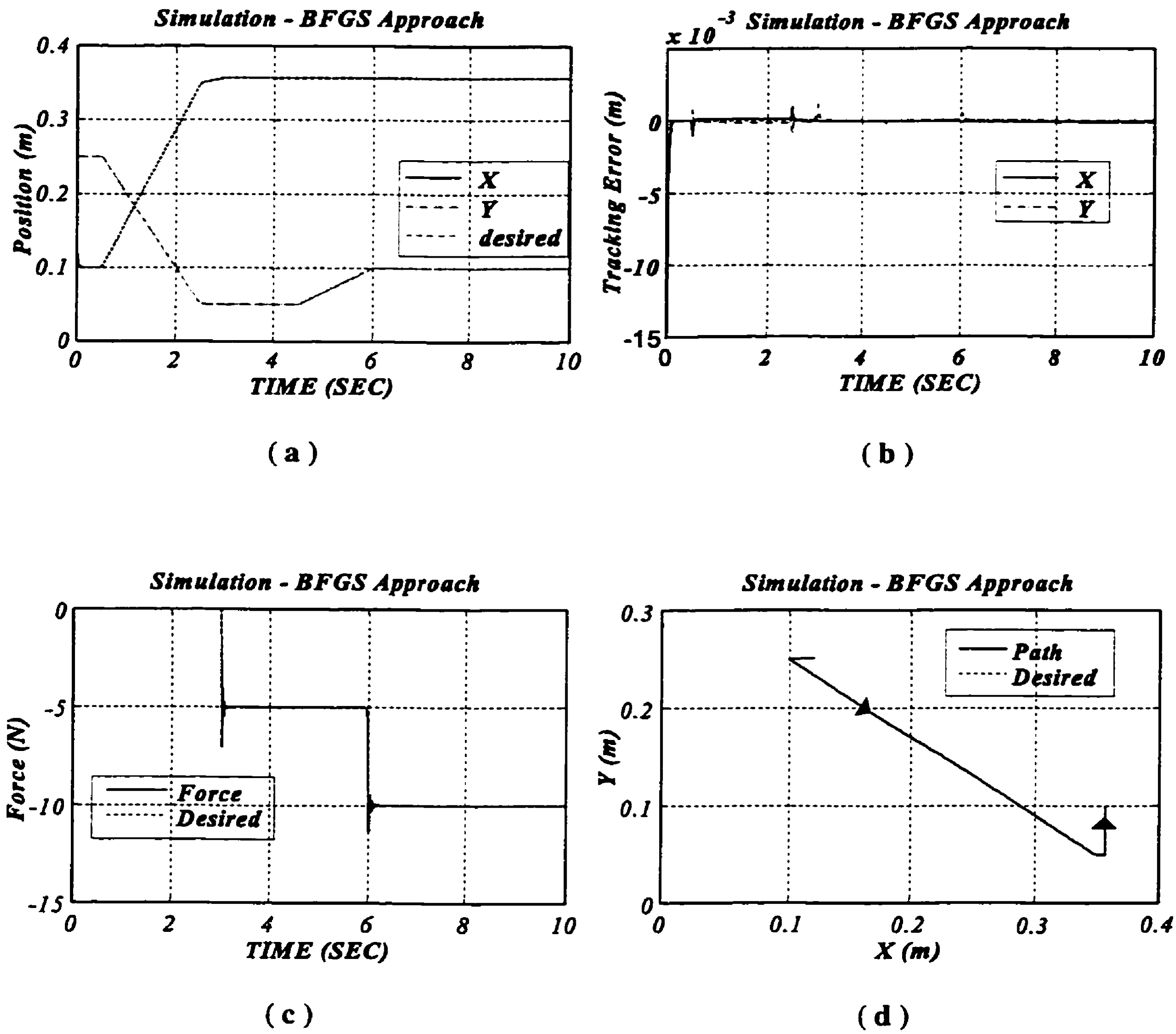


Figure 6.9 - Simulated Response - Variable Metric Approach with Partial Linearization

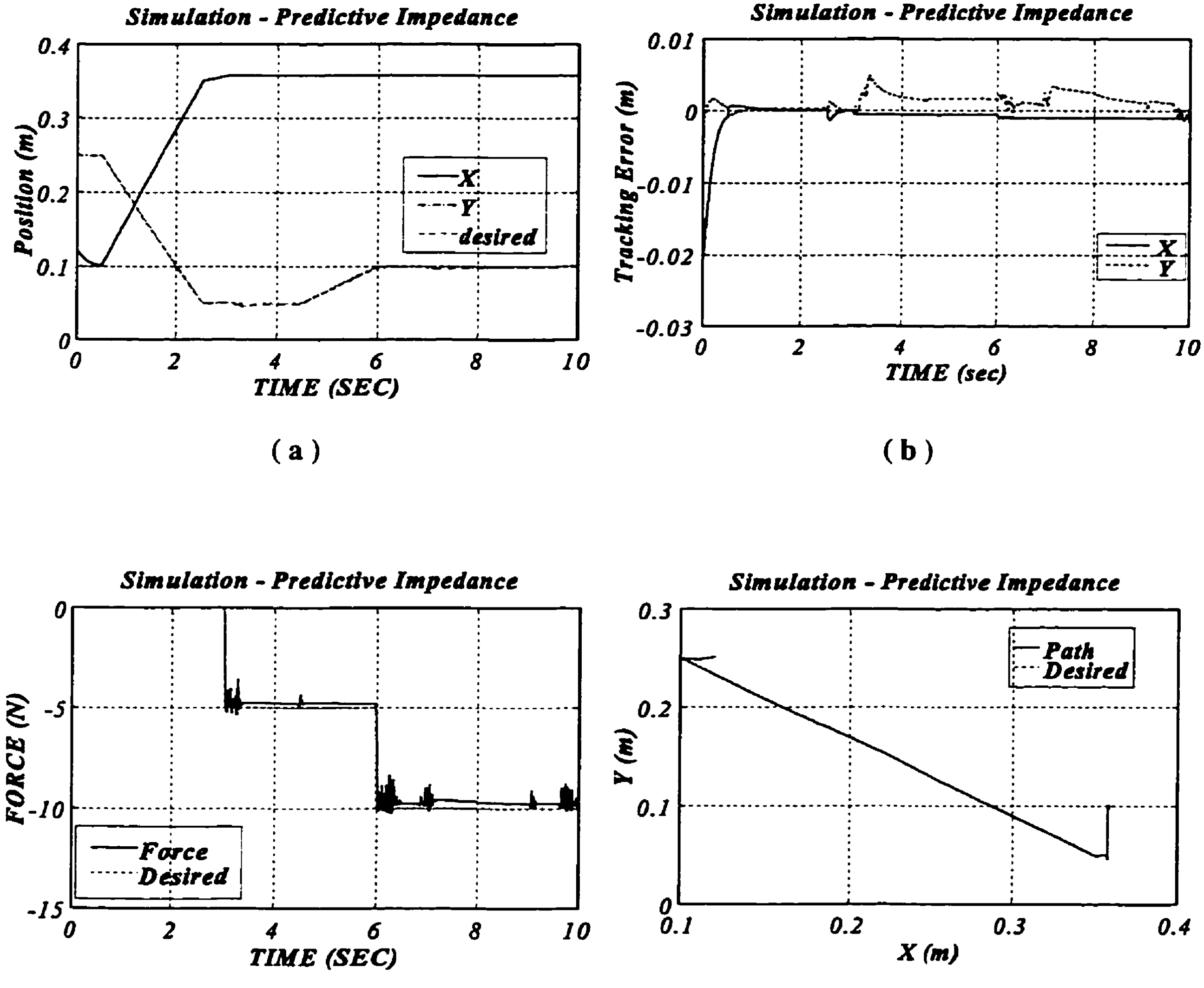
6.3.3 - The Extended Predictive Impedance Approach

The extended predictive impedance approach, with its reduced dynamics and pre-defined search direction, could be a good alternative for a faster realization of MPC. The method is fully detailed in section 5.2.5 and needs no further clarification. Predictive Impedance adds three more control parameters for tuning the control algorithm (search direction). Here, line minimization was achieved using Brent's Golden section method [97]. While a faster approximate minimization was possible with the BFGS approach, a more precise solution was required for the predictive impedance.

The results are shown in figure 6.10. They were obtained for

$$Q_1 = \begin{bmatrix} 2.0e5 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 2.0e5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 200.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 200.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 50.0 \end{bmatrix} \quad Q_2 = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} \quad T_h = 0.02 \text{ sec} \qquad (6.31)$$

and for γ_x=4.0, γ_v=1.0, γ_f=0.03.



(a)

(b)

(c)

(d)

Figure 6.10 - Simulated Response - Extended Model Predictive Impedance

The tracking precision appeared to be reduced by a small factor, but all other properties of the full optimal control approach were maintained. The chattering that occurred during simulation, as visible from the normal contact force curves, is mainly the result of the limited precision of the numerical solution of the MPC interacting with the stiff system obtained during the contact dynamics simulation. In experiment, the small errors in the numerical solution of the MPC problem will have less effect on the real contact force. The algorithm resulted in a reduction by a factor of twenty (20) of the execution time with respect to the BFGS approach. This improvement is sufficient to enable real-time implementation.

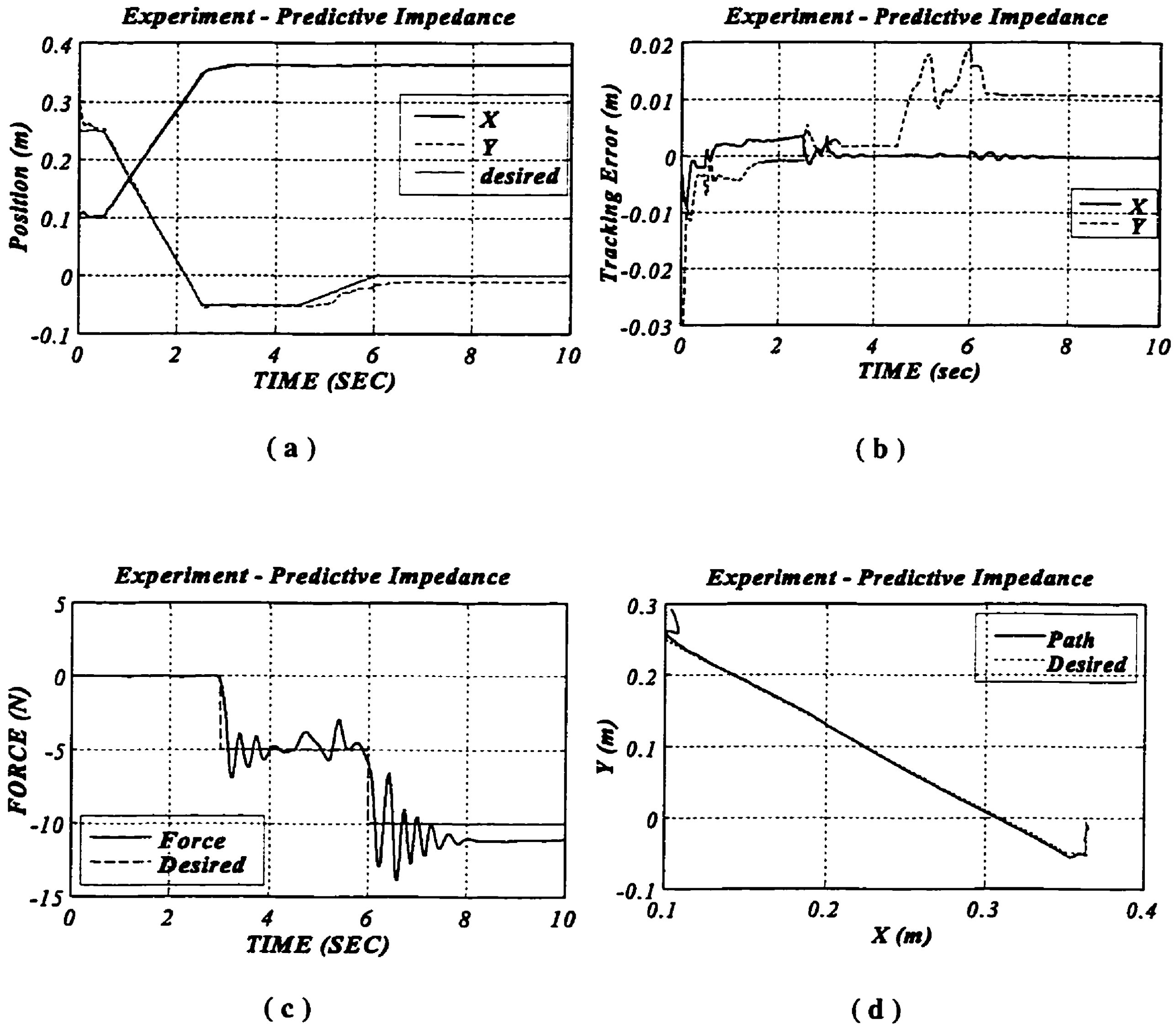
6.4 - Experimental Results

Extended model predictive impedance appeared as the only possible approach for real-time implementation on the dSPACE platform. Obviously, some adjustments were needed to the algorithm and control parameters to deal with experimental issues more difficult to characterize. One of these was the difficulty in precisely evaluating contact surface distance. Here, a manual teaching operation was implemented to provide that information to the controller, but the resulting precision was at best a few tenth of millimetres. For rigid contacts, this could represent a huge offset in the estimated contact force with respect to the measured one. Naturally, this problem could have been solved by using a proximity sensor to provide separation within micrometers. This solution is costly and complicated. Instead, the force sensor was used as a contact detection trigger to adjust the geometric model of the environment. This was sufficient for the particular case of a flat-wall environment. Another major issue was the difficulty in modelling low speed joint and surface friction. Nowadays, this aspect of robot control is a subject of intense research. The experimental results obtained here just emphasize on the need to develop a better comprehension of the friction phenomena at low speed, especially in robots.

As expected from the discussion of section 3.2.4, bandwidth limitations in the force sensor also resulted in implementation problems. Most force sensors, including the one used in this experiment , are rate limited. During a collision, the force sensor response can be momentarily extremely nonlinear. Moreover, to avoid measurement spillovers, the force sensor signal had to be low-pass filtered to remove any content that could destabilize the environment platform (see section 6.2.4). Finally, a bias removal algorithm had to be implemented to remove the offset in the force measurement. This was done by taking five (5) seconds of force sensor data, extracting the mean and subtracting this mean to any other subsequent measures. The results obtained are shown in figure 6.11. They were obtained for

$$Q_1 = \begin{bmatrix} 1.0e7 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0e7 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 50.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 50.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 2.0 \end{bmatrix} \quad Q_2 = \begin{bmatrix} 1.5 & 0.0 \\ 0.0 & 1.5 \end{bmatrix} \quad T_h = 0.02 \text{ sec} \tag{6.32}$$

and for γ_x=6.0, γ_v=1.0, γ_f=0.1.



(a) (b)

(c) (d)

Figure 6.11 - Experimental results - Extended Model Predictive Impedance

Although more weight had to be given to the position error to deal with the increased low-speed joint friction, the results are still good. To avoid excessive position gain, the weight on the requested MPC output had to be increased as well.

More tracking error than in simulation is obtained during the contact motion stage of the trajectory (from 4.5 to 6.5 seconds) due to surface friction. The stick-slip motion in Y during that time period supports this observation. This is **not** a limit inherent to the algorithm presented herein. This type of sticking error is a problem for all control methods dealing with surface friction. Until better models and identification techniques are made available for friction, it is going to be the case. The simulation results presented in the preceding chapter used the same type of model for the dynamic simulation than for the predictor, explaining the good results obtained. The experimental results show that the selected model of friction, although at the leading edge in friction modelling, is still not adequately representing low speed friction.

As expected, the chattering observed in simulation do not appear on the experimental results. On the other hand, oscillations in the force response that were not in the simulation results are observed in experiment. They are essentially the results of the environment platform dynamics. Finally, during the stage where 10 N was requested, the 11.5 N obtained resulted from the effort of the controller to reduce the error in the Y direction.

Chapter VII

Conclusions

A literature review on modelling and control of manipulators doing contact tasks has shown the inability of the existing control approaches to solve the problem in a unified framework. In practice, these methods have to rely on switching laws to match a particular control algorithm with the particular conditions it has been designed for. This discontinuous nature of the solution is the result of non smooth discontinuous terms and, sometimes, variable topological structures in the equations of motion of the close loop system.

A general evaluation of model predictive control (MPC) is enough to understand that the method may be well adapted for the control of dynamic systems undergoing such non smooth evolutions. The model-based predictive nature of the method is indeed an interesting property for such cases. It allows the control law to predict when the discontinuities will occur in time and to modify pro-actively the control command for an upcoming variation. The first contribution of this thesis is to use MPC as a unified solution to the control of nonlinear discontinuous dynamic systems, and in particular the control of robot manipulators doing tasks with intermittent contacts. Because the literature on nonlinear MPC is relatively sparse, this thesis also contributes to the overall understanding of the application of nonlinear MPC. In that respect, the last section of chapter 4 describes the various alternatives for implementing nonlinear MPC.

The theoretical issues regrading the existence of a solution to the general nonlinear MPC problem were discussed in some details. Despite the non-smooth evolution of the dynamics, if a dynamic system is a General Dynamic System (GDS), the solution exists. The notion of GDS thus proves that if a discontinuous system has bounded discontinuities in the state dynamics, the system may still be a GDS with an existing theoretical solution. However, the numerical solution of the global formulation is not prone to real-time implementation, as required in practical control applications. Therefore, this thesis also contributes by proposing two sub-optimal alternatives that reduce how much computation is needed for a solution without losing the interesting features of MPC. Both methods were successfully tested in simulation. Experimental results were also obtained for the fastest option called predictive impedance control.

The applications investigated in this thesis are far less complicated than in real industrial life. Nevertheless, the analyses presented are general enough to encompass every type of application, and the algorithms derived from them are also independent of the complexity of the model. With the belief that computing technology will evolve continuously, the computational aspect of the problem will become obsolete very soon. Nowadays, very complex real-time simulators are being developed to emulate the dynamics of complicated flexible mechanisms with multiple contact points. The time where this technology will become available to the general industrial world is near. At this time in the history, the simple argument of computing time should never be accounted as a drawback to an advanced control method.

References

1. Lewis, F.L., Abdallah, C.T. and Dawson, D.M., "Control of Robot Maniplators", MacMillan, NY, 424 p., 1993.

2. Paul, R.P., "Robot Manipulators", MIT Press, Cambridge, MA, 1981.

3. Youcef-Toumi, K. and Gutz, D.A., "Impact and Force Control: Modelling and Experiments", *ASME Journ. of Dyn. Syst., Measur. and Contr.*, vol 116, no. 1, pp. 89-98, March 1994.

4. Youcef-Toumi, K. and Gutz, D.A., "Impact and Force Control", *Proc. IEEE Conf. on Rob. and Autom.*, pp. 410-416, 1989.

5. Zheng, Y.F. and Hemmani, H., "Mathematical Modelling of Robot Collision with its Enevironment", *Journ. of Rob. Res.*, vol 2, no. 3, pp. 289-307, 1985.

6. Wang, Y. and Mason, M., "Modelling Impact for Robotic Operations", *Proc. of the IEEE Conf. on Rob. and Autom.*, Raleigh, NC, April 1987.

7. Mills, J.K., "Manipulator Transition To and From Contact Tasks: A Discontinuous Control Approach", *Proc. IEEE Conf. on Rob. and Autom.*, Cincinatti, OH, pp. 440-446, 1990.

8. Lokhorst, D.M. and Mills, J.K., "Implementation of a Discontinuous Control Law on a Robot During Collision with a Stiff Environment", *Proc. IEEE Conf. on Rob. and Autom.*, Cincinatti, OH, pp. 56-61, 1990.

9. Gutman, S., "Uncertain Dynamical Systems - A Lyapunov Min-Max Approach", *IEEE Trans. on Autom. Contr.*, vol. AC-24, no. 3, pp. 437-443, 1979.

10. Roxin, E., "On the Generalized Dynamical Systems Defined by Contingent Equations", *Journ. of Diff. Eq.*, vol. 1, pp. 188-205, 1965.

11. Roxin, E., "Stability in General Control Systems", *Journ. of Diff. Eq.*, vol. 1, pp. 115-150, 1965.

12. Walker, I., "Impact Configuration and Measures for Kinematically Redundant and Multiple Armed Robot Systems", *IEEE Trans. on Rob. and Autom.*, vol 10, no. 5, pp. 670-683, October 1994.

13. Whitney, D.E., "An Historical Perspective and State of the Art in Robot Force Control", *Int. Journ. of Rob. Res.*, vol 6, no. 1, pp. 3-7, 1987.

14. Vukobratovich, M. and Tuneski, A., "Contact Control Concepts in Manipulation Robotics - An Overview", *IEEE Trans. on Ind. Electron.*, vol 41, no. 1, pp. 12-24, February 1994.

15. Whitney, D.E. and Nevins, J.L., "What is Remote Center of Compliance (RCC) and What it Can Do?", *Robot sensors vol 2 - Tactile and Non-Vision*, IFS Publ., Springer-Verlag, Berlin, pp. 17-33, 1985.

16. Raibert, M.H. and Craig, J.J., "Hybrid PositionForce Control of Manipulators", *ASME Journ. of Dyn. Syst., Measur. and Contr.*, vol 102, no. 2, pp. 275-281, June 1981.

17. Shin, K.G. and Lee, C.P., "Compliant Control of Robotic Manipulators with Resolved Acceleration", *Proc. IEEE Conf. on Decision and Contr.*, Ft-Lauderdale, Florida, pp. 350-257, December 1985.

18. Yoshikawa, T, "Dynamic Hybrid Position/Force Control of Robot Manipulators - Description of Hand Constraints and Calculation of Joint Driving Force", *Proc. IEEE Conf. on Rob. and Autom.*, San-Francisco, CA, CH2282-2, pp. 1393-1398, 1986.

19. Wen, J.T. and Murphy, S., "Stability Analysis of Position and Force Control for Robot Arms", *IEEE Trans. on Autom. Contr.*, vol AC-36, no. 3, pp. 365-371, March 1991.

20. Khatib, O., "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation", *IEEE Trans. on Rob. and Autom.*, vol RA-3, no. 1, pp. 43-53, 1987.

21. McClamroch, N.H., "Singular Systems of Differential Equations as Dynamic Models for Constrained Robot Systems", *Proc. IEEE Conf. on Rob. and Autom.*, San Francisco, CA, 1986.

22. Wang, D. and McClamroch, N.H., "Position and Force Control for Constrained Manipulator Motion Lyapunov's Direct Method", *IEEE Trans. on Rob. and Autom.*, vol RA-9, no. 3, pp. 308-313, June 1993.

23. Krishnan, H. and McClamroch, N.H., "On the Connection Between Nonlinear Differential-Algebraic Equations and Singularly Perturbed Control Systems in Nonstandard Form", *IEEE Trans. on Autom. Contr.*, vol AC-39, no. 5, pp. 1079-1084, May 1994.

24. Mills, J.K. and Goldenberg, A.A., "Force and Position Control of Manipulators During Constrained Motion tasks", *IEEE Trans. on Rob. and Autom.*, vol RA-5, no. 1, pp. 30-46, 1989.

25. Tabhoud, K.A., Scmidt, T., Schüpphaus, R. and Müller, P.C., "Comparison of Descriptor Models and Reduced Dynamic Models for Constrained Robots", *Proc. IFAC Conf. on Rob. Contr.*, Vienna, Austria, pp. 9-14, 1991.

26. Tabhoud, K.A., "Modelling and Control of Constrained Robots", Ph.D. Dissertation, Universtät-Gesmathochschule Wuppertal, Germany, 112 p., 1993.

27. Kokotovic, P.V., Khalil, H.K. and OReilly, J., "Singular Perturbation Methods in Control: Analysis and Design", Acad. Press, London, 1986.

28. Hogan, N., "Stable Execution of Contact Tasks Using Impedance Control", *Proc. IEEE Int. Conf. on Rob. and Autom.*, Raleigh, N.C., pp. 1047-1054, 1987.

29. Salisburry, J., "Active Stiffness Control of Manipulators in Cartesian Coordinates", *Proc. IEEE Conf. on Decision ad Contr.*, pp. 95-108, December 1980.

30. Fraisse, P., Pierrot, F. and Dauchez, P., "Virtual Environment for Robot Force Control", *Proc. IEEE Conf. on Rob. and Autom.*, Atlanta, Georgia, pp. 219-224, May 1993.

35. Canudas de Wit, C., Olsson, H., Åström, K.J. and Lischinsky, P., " A New Model for Control of Systems with Friction", *IEEE Trans. on Aut. Contr.*, Vol. 40, No. 3, pp. 419-425, March 1995.

36. Gladwell,G.M.L., "Contact Problems in the Classical Theory of Elasticity", Sijthoff & Noordhoff, Alphen aan den Rijn, Netherlands, 1980.

37. Johnson, K.L., "Contact Mechanics", Cambridge University Press, Cambridge, 1985.

38. Greenwood, J.A. and Williamson, J.B.P., "Contact of Nominally Flat Surfaces", *Proc. of Royal Soc.*, London, A295, pp. 300-319, 1966.

39. Chang, W.R., Etsion, I. and Bogy, D.B.. "An Elatsic-Plastic Model for the Contact of Rough Surfaces", *ASME Journ. of Tribol.*, vol 109, no. 1, pp. 257-263, April 1987.

41. Chang, W.R. and Ling, F.F., "Normal Impact Model of Rough Surfaces", *ASME Journ. of Tribol.*, vol. 114, no. 2, pp. 439-447, July 1992.

42. Meriam, J.L., "Statics and Dynamics for Engineers", John Wiley and Sons, New-York, 1980, 540 p.

43 de Carufel, J. and Necsulescu D.S., "Impact/Contact Model for the Control of Mechanical Systems in Contact Motion", *Proc. of CONTACT MECHANICS 95*, Ferrera, Italy,pp. 175-182, July 1995.

44 Shahinpoor, M., "A robot Engineerin textbook", Harper and Row, New-York, 1987.

45 Ma O. and Nahon M., "A Generalized Method for Computing Distances Between two Moving Objects", *ASME, Advances in design Automation*, Vol. 1, 1992, pp 109-117

46 Gotschalk, M.C.and Manocha, D., "OBBTree : A Hierarchical Structure for Rapid Interference Detection", *Proc. Of ACM Siggraph96*, 1996. (http:\\www.cs.unc.edu\~dm\collision.html)

47 Pfeiffer, F., "Dynamical Systems with Time-Varying or Unsteady Structures", ZAMM, vol. 71, no. 4, pp. 6-22, 1991.

48 Jean, M. and Moreau, J.J., "Dynamics in the Presence of Unilateral Contacts and Dry Friction: A Numerical Approach", *Uniletral Problems in Structural Analysis-2, CISM Courses and lectures*, no 304, Springer-Verlag, Wien, New-York, pp. 151-195, 1987.

49 Jean, M. and Moreau, J.J., "Unilaterality and Dry Friction in the Dynamics of Rigid Body Collections", *Proc. Contact Mech. Int. Symp.*, editor A. Curnier, PPUR, pp. 31-48, 1992.

50 Jean, M., "Dynamics of Rigid Bodies with Dry Friction and Partially Elastic Collisions", *Int. Series of Num. Math.*, vol 101, Birkhäuser, Basel, 1991

51 Jean, M. and Pratt, E., "A system of Rigid Bodies with Dry Friction", *Int. Journ. of Eng. Science*, vol 23, no. 5, pp. 497-513, 1985.

52 Alard, P., "A Simple Contact Algorithm Applied to Large Sliding and Anisotropic Friction", Technical Report, Laboratoire de Mécanique et de Génie Civil, Univ. de Montpellier, France, October 1992.

53 Alart, P. and Curnier A., "A Mixed Formulation for Frictional Contact Problems Prone to Newton Like Solution Methods", *Comput. Methods in Appl. Mech. and Eng. 92*, Elsevier Science Publisher. Amsterdam, pp. 353-375, 1991.

54 Curnier, A. and Alart, P.,"A Generalized Newton Method for Contact Problems with Friction", Technical Report, Laboratoire de Mécanique Appliquée, École Polytechnique Fédérale de Lausanne, Switzerland, 1988.

55 Eppinger, S.D and Seering, W.P., "Understanding Bandwidth Limitations in Robot Force Control", *Proc. IEEE Conf. on Rob. and Autom.*, Raleigh, NC, pp. 904-909, April 1987.

56 Fiala, J. and Lumia, R., "The Effect of Time Delay and Discrete Control on the Contact Stability of Simple Position Controllers", *IEEE Trans. on Autom. Contr.*, vol AC-39, no. 4, pp. 870-873, April 1994.

57 Paljung, E., Sugar, T., Kumar, V. and Yun, X., "Some Important Considerations in Force Control Implementation", *Proc. IEEE Conf. on Rob. and Autom.*, Nice, France, pp. 1270-1275, May 1992.

58 Qian, H.P. and De Schutter, J., "The Role of Damping and Low Pass Filtering in the Stability of Discrete Time Implemented Robot Force Control", *Proc. IEEE Conf. on Rob. and Autom.*, Nice, France, pp. 1368-1373, May 1992.

59 Zheng, F.Z. and Fan, Y., "Robot Force Sensor Interacting with Environments", *IEEE Trans. on Rob. and Autom.*, vol RA-7, no. 1, pp. 156-164, February 1991.

60 Jarvis, R.P. and Mills, B., "Vibration Induced by Dry Friction", *Proc. Inst. Mech. Engin.*, vol 178, part I, no 32, pp. 847-857, 1963.

61 Dweib, A.H. and D'souza, A.F., "Self-Excited Vibrations induced by Dry Friction, Part 1 : Experimental Study", *Journ. of Sound and Vibr.*, vol 137, no. 2, pp. 163-175, 1990.

62 Dweib, A.H. and D'souza, A.F., "Self-Excited Vibrations induced by Dry Friction, Part 1 : Stability and Limit-Cycle Analysis", *Journ. of Sound and Vibr.*, vol 137, no. 2, pp. 177-190, 1990.

63 Martin, J.A.C, Oden J.T. and Simoes, F.M.F, "A Study of Static and Kinetic Friction", *Int. Journ. of Engin. Science*, vol 28, no.1, pp. 29-92, 1990.

64 Hess, D.P. and Soom, A., "Normal and Angular Motions at Rough Planar Contacts During Sliding with Friction", *AMSE Journ. of Tribol.*, vol 114, no. 3, pp. 567-578, 1992.

65 Filippov, A.F., "Differential Equations with Discontinuous Right-hand Side", *American Math. Soc. Transl.*, vol 42, series 2, pp. 199-231, 1964.

66 Cutler, C.R. and Ramaker, B.L., "Dynamic Matrix Control - a Computer Algorithm", *AIChE Annual Mtg.*, Houston, Texas, 1979.

67 Richalet, J.A., Rault, A., Testud, J.L. and Papon, J., "Model Predictive Heuristic Control: Application to an Industrial Process", *Automatica*,vol 14, 1978.

68 Soeterboek, R., "Predictive Control: A Unified Approach", Prentice-Hall, New York, 352 p., 1992.

69 Garcia, C.E., Prett, D.M. and Morari, M., "Model Predictive Control: Theory and Practice - a Survey", *Automatica*, vol 25, no. 3, pp. 335-348, 1989.

70 Richalet, J.A., "Pratique de la Commande Prédicitve", Traité des Nouvelles Technologiques, Série Automatique, Hermès, Paris, 349 p., 1993.

71 Garcia, C.E. and Morari, M., "Internal Model Control-1. A Unifying Review and Some New results", *Ind. Eng. Chem. Proc. Des. Dev.*, vol 21, pp. 308-323, 1982.

72 Garcia, C.E. and Morari, M., "Internal Model Control-2. Design Procedure for Multivariable Systems", *Ind. Eng. Chem. Proc. Des. Dev.*, vol 24, pp. 472-484, 1985.

73 Clarke, D.W., Mohtadi, C. and Tuffs, P.S., "Generalized Predictive Control - Part I. The Basic Algorithm", *Automatica*, vol 23, no. 2, pp. 137-148, 1987.

74 Clarke, D.W., Mohtadi, C. and Tuffs, P.S., "Generalized Predictive Control - Part II. Extensions and Interpretations", *Automatica*, vol 23, no. 2, pp. 149-160, 1987.

75 Clarke, D.W., "Application of Generalized Predictive Control to Industrial Processes", *IEEE Contr. Syst. Mag.*, pp. 49-55, 1988.

76 Mothadi, C. and Clarke, D.W., "Generalized Predictive Control, LQ, or Pole-Placement: A Unified Approach", *Proc. IEEE Conf. on Decision and Contr.*, Athens, Greece, pp. 1536-1541, December 1986.

77 Kouvaritakis, B., Rossiter, J.A. and Chang, A.O.T., "Stable Generalized Predictive Control: an Algorithm with Guaranteed Stability", *IEE Proceedings-D*, vol 139, no. 4, pp. 349-362, July 1992.

78 Zhu, M., Warwick, K and Douce, J.L., "Adaptive Generalized Predictive Controller for Nonlinear Systems", *IEE Proceedings*-D, vol 138, no. 1, pp. 33-40, January 1991.

79 M'Saad, M., Dugard, L. and Hammad, SH., "A Suitable Generalized Predictive Adaptive Controller Case Study: Control of a Flexible Arm", *Automatica*, vol 29, no. 3, pp. 589-608, 1993.

80 Clarke, D.W., Mosca, E. and Scattolini, R., "Robustness of an Adaptive Predictive Controller", *IEEE Trans. on Autom. Contr.*, vol AC-39, no. 5, pp. 1052-1056, May 1994.

81 Robinson, B.D. and Clarke, D.W., "Robustness Effects of a Pre-Filter in Generalized Predictive Control", *IEE Proceedings-D*, vol 138, no. 1, pp. 2-8, 1991.

82 Berlin, F. and Frank, P.M., "Design and Realization of a MIMO Predictive Controller for a 3-tank System", *in Advances in Model-Based Predictive Control*, ed. D.W. Clarke, Oxford Science Publications, pp. 446-457, 1994.

83 Necsulescu, D.S. and de Carufel, J., "Predictive Control of Servomotors with Flexible Shaft", Technical Report, University of Ottawa/École Polytechnique Fédale de Lausanne, Seminar presented at EPFL, Switzerland, June 1995.

84 Marquès, D. and Morari, M., "Model predictive Control of Gas Pipeline Network", *Proc. Amer. Contr. Conf.*, Seattle, WA, pp. 349-354, 1986.

85 Garcia, C.E., "Quadratic Dynamic Matrix Control of Nonlinear Processes. An Application to a Batch Reaction Process", *AIChE Annual Meeting*, San Franscisco, CA, 1984.

86 DelRe, L., Chapuis, J. and Nevistić, V., "Predictive Control with Embedded Feedback Linearization for Bilinear Plants with Input Constraints", *Proc. IEEE Conf. on Decision and Contr.*, pp. 2984-2988, 1993.

87 De Nicolao, G. and Scattolini, R., "Stability and Output Constraints in Predictive Control", *in Advances in Model-Based Predictive Control*, ed. D.W. Clarke, Oxford Science Publications, pp. 106-121, 1994.

88 Campo, P.J. and Morari, M, "Robust Model Predictive Control", *Proc. Americ. Contr. Conf.*, Minneapolis, MN, pp. 1021-1026, June 1987.

89 Allwright, J.C., "On Min-Max Model-Based Predictive Control", *in Advances in Model-Based Predictive Control*, ed. D.W. Clarke, Oxford Science Publications, pp. 415-426, 1994.

90 Sanzo, M., Richalet, J. and Prada, C., "Matching the Uncertainty of the Model Given by Global Identification Techniques to the Robustness of a Model Based Predictive Controller" *in Advances in Model-Based Predictive Control*, ed. D.W. Clarke, Oxford Science Publications, pp. 386-401, 1994.

91 Kwon, W.H. and Pearson, A.E., "On Ffeedback Stabilization of Time-Varying Discrete Linear Systems", *IEEE Trans. on Autom. Contr.*, vol AC-23, no. 3, pp. 479-480, June 1978.

92 Lewis, F.L. and Syrmos, V.L. "Optimal Control ", 2nd edition, Jonh Wiley and Sons, New-York, 1995, 560 p.

93 Sage, A.P. and White, C.C., "Optimal Control Systems", Prentice-Hall, Englewood Cliffs, NY, 1977.

94 Isidori, A., "Nonlinear Control Systems", Springer-Verlag, New York, 2nd edition, 1989.

95 Khalil,H.K., "Nonlinear Control", MacMillan Publishing Company, New York, NY, 1992, 564 p.

96 Fossard, A.J. et Norman-Cyrot, D., "Systèmes Nonlinéaires", Masson, Paris, 1993.

97 Necsulescu, D.S., Lonmo, V., Kim, B. and Vukovitch, G. "Autonomous Mobile Robot Control in Operational Space with Torque Saturation, Slippage and Tip-over Avoidance", *1995 IEEE Conf. on Rob. and Autom.*, Orlando, Florida, 1995.

98 Fletcher, R. "Practical Methods of Optimization", Vol.1 - *Unconstrained Optimization*, Vol. 2 - *Constrained Optimization*, John Wiley and Sons, New-York, 1980.

99 "Numerical Recipe in C: The Art of Scientific Computing ", Cambridge University Press, Cambrifge, MA, 1991.

100 Branch, M.A. and Grace A., "Matlab Optimization Toolbox - User's Guide", The MathWorks, Natick, MA, 1996.

101 Jassemi, R. and Kiguchi, J., *Direct Drive Robot - design Documents*, Mechanical Engineering Department, University of Ottawa, Ottawa, Canada, 1991

102 NSK Corp., *MegaTorque Brushless DC Motors - Reference manual*, Japan.

103 Armstrong, K.J. and Wittenmark, B., "Adaptive Control", Addison-Wesley, Reading, MA, 1989, 526 p.

Appendix A

Rigid Multi-body Dynamics

The goal of this Appendix is to provide a quick access to some theoretical background on rigid multi-body dynamics. Some of the analysis presented in chapter 3 makes use of the notion of constrained multibody systems. The material presented here is an introductory lecture in the matter. It is presented with enough detail to allow the understanding of chapter 3 without having to consult another reference on the subject. However, it is not a complete treatise on the subject matter.

Rigid multi-body dynamics is the branch of mechanics studying the dynamics of collections of rigid bodies. The analysis of multibody dynamical systems requires the use of numerous analytical tools, including classical mechanics, analytical mechanics and differential geometry. Good overviews of classical and analytical mechanics are provided in Gantmacher[A.1], Spiegel[A.2] and Woodhouse[A.3]. Most of the material presented here was adpated from Haugh[A.4]. Simeon[A.5] and Tahboud[A.6].

A.1 - Free-Body Dynamics

The dynamics of one body I of a collection of n bodies can be defined using the standard Newton-Euler equations as

$$\begin{aligned} \frac{d}{dt}\left(m_i \dot{x}_i\right) &= F_i + F_i^C + F_i^F \\ \frac{d}{dt}\left(I_i\ \omega_i(\theta_i, \dot{\theta}_i)\right) &= \tau_i + \tau_i^C + \tau_i^F \end{aligned} \qquad \textbf{(A.1)}$$

where m_i is the mass, $x_i \in \mathbb{R}^3$ is the absolute coordinate of the centre of mass, $F_i \in \mathbb{R}^3$ represents the sum of externally applied forces, $F_i^C \in \mathbb{R}^3$ is the sum of the constraint forces, $F_i^F \in \mathbb{R}^3$ is the sum of frictional forces acting at the contact points, $\mathbf{I}_i \in \mathbb{R}^3 \times \mathbb{R}^3$ is the inertia tensor in some body frame, $\omega: \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ defines the angular velocity of the body expressed along the body axis, $\theta \in \mathbb{R}^3$ is a 3 dimensional vector containing the Euler angles, $\tau_i \in \mathbb{R}^3$ is the sum of moment resulting from external efforts, $\tau_i^C \in \mathbb{R}^3$ is the sum of moment resulting from the constraint efforts and $\tau_i^F \in \mathbb{R}^3$ is the sum of moment resulting from frictional efforts. In this description, the $\dot{}$ operator defines the term wise derivative of a vector.

With the Euler angles description, the angular velocity is given by

$$\omega = T(\theta)\dot{\theta} \tag{A.2}$$

where $T: \mathbb{R}^3 \rightarrow \mathbb{R}^3 \times \mathbb{R}^3$ such that by differentiation,

$$\frac{d}{dt}\omega = T(\theta)\ddot{\theta} + \dot{T}(\theta)\dot{\theta} \tag{A.3}$$

In equation A.1, the angular momentum is defined with respect to the non inertial body axes. Consequently, when performing the derivative on the right-hand side of both equations, the Coriolis theorem is applied and equation A.1 becomes

$$\begin{aligned} m_i \ddot{\boldsymbol{x}}_i &= \boldsymbol{F}_i + \boldsymbol{F}_i^C + \boldsymbol{F}_i^F \\ \boldsymbol{I}_i\, \dot{\omega}_i + \omega_i^{\bullet} \boldsymbol{I}_i\, \omega_i &= \tau_i + \tau_i^C + \tau_i^F \end{aligned} \tag{A.4}$$

where $\omega^{\bullet}$ is the skew-symmetric matrix described in general tensor notation as $\omega^{\bullet}_{jk} = \varepsilon_{ijk}\omega_i$ ($\varepsilon_{123}=\varepsilon_{231}=\varepsilon_{312}=1$ and $\varepsilon_{321}=\varepsilon_{132}=\varepsilon_{213}=-1$) and ω_i is the i^th^ component of the velocity vector ω.

Definition A.1. The coordinate vector $\mathbf{q}^f \triangleq \{\mathbf{x}_1^t, \theta_1^t, \mathbf{x}_i^t, \theta_i^t, \ldots, \mathbf{x}_n^t, \theta_n^t\}^t \in \mathbb{R}^{6n}$ is called the *generalized free-body coordinate* vector of the collection of rigid bodies.

Equation A.2 and A.3 can be used to define generalized free-body velocity and acceleration vectors.

Definition A.2. The velocity vector $\dot{\mathbf{q}}_p^{f\,t} \triangleq \{\mathbf{x}_1^t, \omega_1^t, \mathbf{x}_i^t, \omega_i^t, \ldots, \mathbf{x}_n^t, \omega_n^t\}^t \in \mathbb{R}^{6n}$ is called the *generalized free-body velocity vector* of the collection of rigid bodies.

Definition A.3. The acceleration vector $\ddot{\mathbf{q}}_{p}^{f\,t} \triangleq \{\mathbf{x}_1^{t}, \alpha_1^{t}, \mathbf{x}_i^{t}, \alpha_i^{t}, \ldots, \mathbf{x}_n^{t}, \alpha_n^{t}\}^{t} \in \mathbb{R}^{6n}$ is called the *generalized free-body acceleration vector* of the collection of rigid bodies.

The differential relationship between the free-body generalized position, velocity and acceleration is given by

$$\ddot{q}_p^f = \frac{d}{dt}\dot{q}_p^f$$

$$\dot{q}_p^f = \begin{bmatrix} I & 0 & 0 & \cdots & 0 & 0 \\ 0 & T(\theta_1) & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 \\ 0 & 0 & 0 & \cdots & 0 & T(\theta_n) \end{bmatrix} \dot{q}^f \qquad \text{(A.5)}$$

where T(θ) is defined by equation A.2 and **I** defines the unity matrix.

Definition A.4. The equations obtained for each free-body can be grouped into

$$M^f \ddot{q}_p^f = Q + Q^C + Q^F + Q^{nl}(\dot{q}_p^f) \qquad \text{(A.6)}$$

where $\mathbf{M}^f$ is the inertia matrix, **Q** the vector of generalized external forces, $\mathbf{Q}^C$ the generalized constraint force, $\mathbf{Q}^F$ is the generalized friction force and $\mathbf{Q}^{nl}$ is the generalized nonlinear force (from Coriolis effects)

$$M^f \triangleq \begin{bmatrix} m_1 I & 0 & \cdots & 0 & 0 \\ 0 & I_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & m_n I & 0 \\ 0 & 0 & \cdots & 0 & I_n \end{bmatrix} \in \Re^{6n \times 6n} \qquad \text{(A.7a)}$$

$$Q \triangleq \begin{Bmatrix} F_1 \\ \tau_1 \\ \vdots \\ F_n \\ \tau_n \end{Bmatrix} , \; Q^C \triangleq \begin{Bmatrix} F_1^C \\ \tau_1^C \\ \vdots \\ F_n^C \\ \tau_n^C \end{Bmatrix} , \; Q^F \triangleq \begin{Bmatrix} F_1^F \\ \tau_1^F \\ \vdots \\ F_n^F \\ \tau_n^F \end{Bmatrix} , \; Q^{nl} \triangleq \begin{Bmatrix} 0 \\ -\omega_1^{\bullet} I_1 \omega_1 \\ \vdots \\ 0 \\ -\omega_n^{\bullet} I_n \omega_n \end{Bmatrix} \in \Re^{6n} \qquad \text{(A.7b)}$$

is called the *free-body dynamics of the collection of rigid bodies.*

A.2 - Constraints

The motions of each body in a collection of rigid bodies can be related to the motion of other bodies by the mean of constraints. In rigid multi-body dynamics, constraints represents idealized relationship between the motion of neighbouring bodies. In this subsection, the terminology used to describe such rigid constraints is defined. Constraints generally represents kinematic relationships between the free-body coordinates and velocities of the different bodies in a collection of bodies. They are mathematically represented as

$$\Gamma(t,q^f,\dot{q}^f) \geq 0 \qquad \text{(A.8)}$$

where $\Gamma \in \mathbb{R}^c$, c is the number of constraints. Typically, constraints equations are derived recursively using relative variables describing the configuration of one body with respect to the other ones it relates to.

Definition A.5. Constraints for which the relation A.8 is an equation are called *omnilateral constraints* while those with inequality relationships are called *unilateral constraints.*

Definition A.6. Constraints that are not explicitly function of time are called *scleronomic constraints.* Their counterparts, called *rheonomic constraints,* are explicitly function of time.

Throughout the rest of this appendix, only scleronomic omnilateral constraints will be considered and the time variable will be dropped from the notation of the constraints equations.

Definition A.7. Geometrical constraints of the form

$$\phi(q^f) = 0 \tag{A.9}$$

for which rank($\partial\phi/\partial q^f$)=h, are called *holonomic constraints of order h.*

Definition A.8. Kinematic constraints of the form

$$\Omega(q^f)\, \dot{q}^f = 0 \qquad or \qquad \Omega(q^f)\, \dot{q}^f_p = 0 \tag{A.10}$$

for which rank(Ω)=d are called *differential constraints of order d.*

Definition A.9. The matrix Ω is called the *constraint Jacobian.*

Definition A.10. Differential constraints that can be integrated into the geometric form (eq. A.9) are called the *differential form* of the holonomic constraints.

Definition 2.11. Differential constraints that cannot be integrated into the geometric form (eq. A.9) are called the *nonholonomic constraints.*

The nonholonomicity of a differential constraint is typically verified using the Pfaffian test. The problems associated with nonholonomic constraints were not well understood until this century. The non-integrability of the differential constraint invalidates any approach that uses the ideal constraint assumption, such as the Lagrangian and Hamiltonian formulations. Other formulations such as the Gauss principle of least constraint, the Gibbs-Appel method and Kane's variational method, can deal with constrained generalized coordinates and velocities having different dimensions.

A.3 - Constrained Dynamics

The set of equations defining the free-body dynamics is under determined. The set of equations defined by the constraints and free-body dynamics is, however, usually well determined. In general, the following statements hold.

Definition A.12. The set of differential equations defined by the free-body dynamics (eq. A.6) and the differential constraints of order d (eq. A.10) is called the *Set of Constrained Dynamics Equations.*

The set of constrained dynamic equations defines a system of 6n+d differential equations to be solved simultaneously. The d differential constraints can be used to evaluate the d constraint forces in the system. One way to formalize this is by using the Lagrange Multiplier Theorem.

Theorem A.1. (Lagrange Multiplier Theorem) Let $\mathbf{b}\in\mathbb{R}^n$ and $\mathbf{A}\in\mathbb{R}^{m\times n}$, if $\mathbf{s}^t\mathbf{b}=0$ $\forall s\in$ *ker*$\mathbf{A}$, then $\exists\boldsymbol{\lambda}\in\mathbb{R}^m$ s.t. $\mathbf{s}^t\mathbf{b} + \mathbf{s}^t\mathbf{A}^t\boldsymbol{\lambda} = 0$ $\forall s\in\mathbb{R}^n$. Since it is for all admissible $\mathbf{s}$, then $\mathbf{b}=-\mathbf{A}^t\boldsymbol{\lambda}$. Moreover, if $\mathbf{A}$ is full row rank, $\boldsymbol{\lambda}$ has a unique solution.

Proof : The proof of this theorem is done using the implicit function theorem. From the implicit function theorem, one can define $\mathbf{s}=\{\mathbf{u},\mathbf{v}\}$ such that $\mathbf{u}=-\mathbf{B}^{-1}\mathbf{C}\mathbf{v}$ where $\mathbf{B}$ and $\mathbf{C}$ are defined by $\mathbf{A}=[\mathbf{B}\ \mathbf{C}]$. Similarly, one can define $\mathbf{b}=\{\mathbf{e}^t,\mathbf{f}^t\}^t$ such that $\mathbf{u}^t\mathbf{e}+\mathbf{v}^t\mathbf{f}=0$. Substituting $\mathbf{u}$, one obtains $\mathbf{v}^t(-\mathbf{C}^t\mathbf{B}^{-t}\mathbf{e} + \mathbf{f})=0$. Defining $\boldsymbol{\lambda}=-\mathbf{B}^{-t}\mathbf{e}$, one obtains $\mathbf{e}=-\mathbf{B}^t\boldsymbol{\lambda}$ and $\mathbf{f}=-\mathbf{C}^t\boldsymbol{\lambda}$ which proves the theorem. □

Corollary to theorem A.1 The constraint forces $\mathbf{Q}^C$ of the free-body dynamics equation can be expressed as

$$Q^C = -\Omega(q^f)^t\ \lambda \qquad \textbf{(A.11)}$$

if the *constraint Jacobian* $\boldsymbol{\Omega}$ is full row rank.

Definition 2.13. The equation

$$\begin{bmatrix} M^f & \Omega(q^f)^t \\ \Omega(q^f) & 0 \end{bmatrix} \begin{Bmatrix} \ddot{q}^f_p \\ \lambda \end{Bmatrix} = \begin{Bmatrix} Q + Q^F + Q^{nl}(\dot{q}^f_p) \\ \mu \end{Bmatrix} \qquad \textbf{(A.13)}$$

obtained differentiating the differential constraint (A.10) once more

$$\Omega(q^f)\ \ddot{q}^f_p = -\dot{\Omega}(q^f)\ \dot{q}^f_p = \mu \qquad \textbf{(A.12)}$$

and using A.11 is called the *descriptor form of the constrained dynamics.*

The existence of a solution to the descriptor form constrained dynamics is defined by the following theorem.

Theorem A.2. (Constrained Dynamics Existence Theorem) Let **a** be an admissible virtual velocity ($\mathbf{a}\in\{\mathbf{a}|\mathbf{\Omega a}=0\}$), then if $\mathbf{a}^t\mathbf{M}^f\mathbf{a}>0\ \forall\mathbf{a}\in\{\mathbf{a}|\mathbf{\Omega a}=0\}$ and if the constraint Jacobian $\mathbf{\Omega}$ is full row rank, then there exists a unique solution to the constrained dynamics given by

$$\begin{Bmatrix} \ddot{q}_p^f \\ \lambda \end{Bmatrix} = \begin{bmatrix} M^f & \Omega(q^f)^t \\ \Omega(q^f) & 0 \end{bmatrix}^{-1} \begin{Bmatrix} Q + Q^F + Q^{nl}(\dot{q}_p^f) \\ \mu \end{Bmatrix} \qquad \textbf{(A.14)}$$

The proof of this theorem is given in Haugh[A.4]. The full row rank condition of the constraint Jacobian is strong. There may exist configurations in a collection of rigid bodies where two or more constraints become redundant. In these conditions, the constraint Jacobian loses its full row rank condition and the existence of a solution is momentarily lost.

Definition A.14. The set of configuration $C\triangleq\{q^t|\Omega(q^f)$ is not full row rank$\}$ is called the set of *constraint singularities.*

Research is underway to investigate the problem of singularity in mechanical systems. Numerical integration and control system design problems are some of the fields where singularity plays an important role. This is beyond the scope of this appendix.

A.1.4 - Analytical Solution

One global analytical solution consists of analytically inverting the matrix shown in equation A.14. This approach may be extremely time and effort consuming, especially when the number of bodies and constraints are high. The most widely used methods are the substitution methods. Among the most popular , the recursive Newton-Euler consists of recursively solving constraint forces from the extremity going inwards, and substituting the result for the inboard body. All substitution methods makes use of the following theorem.

Theorem A.3. (Coordinate Partitioning Theorem) Let each element of **Ω** with rank d be c times differentiable and let $\mathbf{q}^f(0)$ be the initial condition, then by the implicit function theorem, if **Ω** is full row rank, there exists a partitioning of $\mathbf{q}^f=\{\mathbf{u}^t,\mathbf{v}^t\}^t$ for which rank$\{\mathbf{\Omega}_u(\mathbf{q}^f(0))\}$=d such that there exists a unique c times differentiable solution **u=u(v)** valid in a neighbourhood of $\mathbf{v}_0$.

The proof of this theorem is not presented here. In reality, the theorem is a corollary to the implicit function theorem and, consequently, shares the same proof. It can easily be extended to include the partitioning of the velocity vector, with the result that $\dot{\mathbf{u}}$=**f**(v,$\dot{\mathrm{v}}$) valid in some neighbourhood of $\mathbf{v}_0$ and $\dot{\mathbf{v}}_0$.

Definition 2.15. The coordinates **u** and v obtained from the partitioning are respectively called the *dependent coordinates* and the *independent coordinates*.

The coordinate partitioning theorem can be used to partition the descriptor form as

$$\begin{bmatrix} M_{uu} & M_{uv} & \Omega_u^t \\ M_{vu} & M_{vv} & \Omega_v^t \\ \Omega_u & \Omega_v & 0 \end{bmatrix} \begin{Bmatrix} \ddot{u} \\ \ddot{v} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} Q_u + Q_u^F + Q_u^{nl} \\ Q_v + Q_v^F + Q_v^{nl} \\ \mu \end{Bmatrix} \tag{A.15}$$

From equation A.15, $\ddot{\mathbf{u}}$ may be expressed as

$$\ddot{u} = \Omega_u^{-1}\left[\mu - \Omega_v \ddot{v}\right] \tag{A.16}$$

and substituting equation A.16 in the first two equations of equation A.15, one obtains a reduced order problem for which the solution for $\ddot{\mathbf{v}}$ observe the following equation

$$\hat{M}\ddot{v} = \hat{Q} + \hat{Q}^F + \hat{Q}^{nl} \tag{A.17}$$

where

$$\begin{aligned} \hat{M} &= M_{vv} - M_{vu}\Omega_u^{-1}\Omega_v - \Omega_v^t\Omega_u^{-1}\left[M_{uv} - M_{uu}\Omega_u^{-1}\Omega_v\right] \\ \hat{Q}^i &= Q_v^i - M_{vu}\Omega_u^{-1}\mu - \Omega_v^t\Omega_u^{-t}\left[Q_u^i - M_{uu}\Omega_u^{-1}\mu\right] \end{aligned} \tag{A.18}$$

The constraint efforts are then expressed as

$$\lambda = \Omega_u^{-t}\left[Q_u + Q_u^F + Q_u^{nl} - M_{uv}\ddot{v} - M_{uu}\ddot{u}\right] \qquad \text{(A.19)}$$

As a result, the dynamics part of the constrained dynamics is reduced to the dynamics of the independent coordinate (eq. A.17). Equations. A.16 and A.19 are consequently algebraic expressions of the dependent coordinate and constraint effort respectively as a function of the independent coordinates (and higher derivatives). This generic representation of substitution methods is purely mathematic. In practice, as for the recursive Newton-Euler method, the substitution is applied recursively to single constraint partition instead of to the hole vector of constraints.

Definition A.16. The set of equations given by the dynamics of the independent coordinates (A.17) and the expression of the constraint effort (eq. A.19) is called the *reduced form of the constrained dynamics.*

Another general method to obtain a solution is the method of the orthogonal complement, based on the following definitions and theorem.

Definition A.17. Consider a matrix $A(x):\mathbb{R}^n \rightarrow \mathbb{R}^{n\times m}$, then a matrix $C(x):\mathbb{R}^n \rightarrow \mathbb{R}^{m\times n}$ for which $\mathbf{AC=0}\ \forall x\mathbb{R}^n$ is called the *orthogonal complement* of A.

Theorem A.4. (Orthogonal Complement theorem) Consider a matrix $A\in\mathbb{R}^{n\times m}$ where $n\neq m$ then the matrix $\mathbf{C=I\text{-}A^t(AA^t)^{-1}A}$ is an orthogonal complement of **A**.

proof: $\mathbf{AC=A(I\text{-}A^t(AA^t)^{-1}A)=0}$. □

Definition A.18. Consider a matrix $A\in\mathbb{R}^{n\times m}$ where $n\neq m$, then the matrix $\mathbf{A^{*}=A^t(AA^t)^{-1}}$ is the *pseudoinverse* of A. Note that only m columns of $\mathbf{A^{*}}$ are independent.

The orthogonal complement method makes use of the orthogonal complement to define the null space of the constraint forces. Defining an orthogonal complement of the constraint Jacobian,

$$C = I - \Omega^t(\Omega\Omega^t)^{-1}\Omega \qquad \text{(A.20)}$$

premultiplying equation A.6 by $\mathbf{C}^t$ and using equation A.11, one obtains

$$\boldsymbol{M}^{*}\ddot{\boldsymbol{q}}_{p}^{f} = \boldsymbol{C}^{t}[\boldsymbol{Q} + \boldsymbol{Q}^{F} + \boldsymbol{Q}^{nl}] \tag{A.21}$$

where $\mathbf{M}^{*}=\mathbf{C}^{t}\mathbf{M}^{f}$. Using again the coordinate partitioning theorem, the acceleration vector can be partitioned into dependent and independent coordinates, and the reduced form of the constrained dynamics is obtained. The constraint efforts are obtained by isolating $\boldsymbol{\lambda}$ in equation A.6. The resulting equations are identical to equations A.16, A.17 and A.19.

Practically, the orthogonal complement solution is relatively complex for the general case. However, some relationships between the orthogonal complement and the partitions $\mathbf{\Omega}_u$, $\mathbf{\Omega}_v$ for the case of a three structure simplifies the formulation when relative coordinates are used to define the kinematic.

A.1.5 - Numerical Solution

Numerical solutions can be obtained in a numerous number of ways. Here, an algorithm to solve globally the descriptor form and proposed by Haug [A.4] is presented. The algorithm has been adapted to incorporate the presence of mixed constraints (holonomic and nonholonomic). First, the holonomic constraints are represented by $\boldsymbol{\phi}(\mathbf{q})=0$ and the non-holonomic constraints are given by $\boldsymbol{\Lambda}(\mathbf{q})\dot{\mathbf{q}}=\mathbf{0}$, such that the total differential constraint is $\mathbf{\Omega}(\mathbf{q})\dot{\mathbf{q}}=\mathbf{0}$ where $\mathbf{\Omega}(\mathbf{q})=\{\nabla_q\boldsymbol{\phi}(\mathbf{q})^t,\boldsymbol{\Lambda}(\mathbf{q})^t\}^t$.

step 1 → Partition $\mathbf{q}$ such that $\nabla_u\boldsymbol{\phi}(\mathbf{q})\dot{\mathbf{u}}+\nabla_v\boldsymbol{\phi}(\mathbf{q})\dot{\mathbf{v}}=\mathbf{0}$ using Gaussian elimination.

step 2 → Partition $\dot{\mathbf{q}}$ such that $\mathbf{\Omega}_w(\mathbf{q})\dot{\mathbf{w}}+\mathbf{\Omega}_x(\mathbf{q})\dot{\mathbf{x}}=\mathbf{0}$ using Gaussian elimination.

step 3 → Solve equation A.13 using LU decomposition and back substitution to obtain $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$.

step 4 → Integrate numerically $\ddot{\mathbf{q}}$ to obtain $\dot{\mathbf{q}}$ and $\mathbf{q}$. For example, the Adams-Bashforth-Moulton-predictor can be used. Note that values for $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, $\mathbf{x}$, $\dot{\mathbf{u}}$, $\dot{\mathbf{v}}$, $\dot{\mathbf{w}}$ and $\dot{\mathbf{x}}$ are obtained.

step 5 → Keeping $\mathbf{v}$ constant and using the value of $\mathbf{u}$ obtained in step 4 as initial guess, use a nonlinear Newton-Raphson algorithm to find the values of $\mathbf{u}$ that verifies the holonomic constraints $\boldsymbol{\phi}(\mathbf{u},\mathbf{v})=\mathbf{0}$. Obtain new $\mathbf{q}$ that verifies the holonomic constraints.

step 6 → Evaluate the constraint Jacobian with the new $\mathbf{q}$ and solve $\mathbf{\Omega}_w(\mathbf{q})\dot{\mathbf{w}}+\mathbf{\Omega}_x(\mathbf{q})\dot{\mathbf{x}}=\mathbf{0}$ for $\dot{\mathbf{w}}$ keeping $\dot{\mathbf{x}}$ constant using a Gaussian elimination with back substitution. Obtain a new $\dot{\mathbf{q}}$ that verifies the differential constraints.

step 7 → In the resulting Gaussian elimination done in step 1 or 2, if the new $\mathbf{q}$ and $\dot{\mathbf{q}}$ generates a small pivot condition (invalidate the partition), proceed to next iteration from step 1.

step 8 → until the end, proceed to next iteration from step 3.

A.4 - References

A.1 Gantmacher, F., "Lectures in Analytical Dynamics", MIR, Moscow, 264 p., 1970.

A.2 Spiegel, M.R., "Theoretical Mechanics", Schaum's Outline Series, MacGram-Hill, NY, 368 p., 1990.

A.3 Woodhouse, N.M.J., "Introduction to Analytical Dynamics", Oxford Science Publ., Clarendon Press, Oxford, 169 p., 1987.

A.4 Haugh, E.J., "Intermediate Dynamics", Prentice-Hall, Englewood Cliffs, NJ, 420 p., 1992.

A.5 Simeaon, B., Führer, C. and Rentrop, P., "Differential-Algebraic Equations in Vehicle System Dynamics:, *Surveys on Math. for Industry*, Springer-Verlag, pp. 1-37, 1991.

A.6 Tabhoud, K.A., "Modelling and Control of Constrained Robots", Ph.D. Dissertation, Universtät-Gesmathochschule Wuppertal, Germany, 112 p., 1993.

Appendix B

Tribology

Tribology as a science was born in England at the beginning of this century. However, the study of friction is relatively old. Leonardo da Vinci at the end of the XVth century described in his *Notebooks* the analysis of the motion of a brick on an inclined flat wall (refer to Armstrong [B.1]). His statements reflected what is known today as the fundamental laws of friction. However, it is only some two centuries after and apparently without the knowledge of da Vinci's work that Amontons stated the laws of friction. They are summarized as 1) the friction force is a force opposing the motion, proportional to the normal load, and 2) the friction force is independent of the contact area.

One of the most substantial contribution to tribology is the work of a French army Engineer (Charles Augustin Coulomb) who, stimulated by the French war industry, established scientifically the basic laws of tribology. His statements on the relation that exists between the microstructure of the interfacing bodies in contact and the friction force has survived up until now. In his model, the surface roughness was represented by bristles and the friction force was explained by the effort necessary to move two brushes one relative to another, with their bristles in contact. Using this analogy, the difference between static and kinetic friction could be explained by the fact that, at higher speed, the bristles may have jump over opposing gaps resulting in less effort to oppose the motion. In addition, the reduction of friction due to lubrication could also be explained by the lubricant filling up the gaps. The main weakness of the Coulomb bristle model was its incapacity to explain the dissipative process of friction. However, the effect of surface roughness on friction was found to be a large part of the truth.

In the 1930's and 1940's, and with the intention of understanding wear and lubrication for industrial purposes, Bowden and Tabor [B.2] investigated the microscopic behaviour of solids in contact. Since then, tribology as it is defined today was born. The rate of publication in tribology has been increasing ever since. An impressive amount of publications, treating both experimental and theoretical studies, is available today. The following discussion is a summary of the ideas that emerged from these investigations and that are considered as fundamental to tribology. The discussion is separated into four sections. First, the friction in unlubricated contacts is described. With the understanding of the fundamental process underlying dry friction, the effect of lubrication is presented in a second subsection. Then, for practical purposes, some of the models representing the macroscopic behaviour of friction are presented and analyzed.

B.1 - Unlubricated Friction

In his work, Coulomb used the bristle model to represent the topography of the contact surface. A generalization of the bristle was introduced by Bowden [B.2]: the asperity model. While bristles are long and thin by definition, asperities can have any height and size. This generalization directly lead to the concept of apparent and true surface of contact.

A study of the topography of the contact area is indeed the best approach to explain friction. Consider two bodies A and B in contact over a finite dimension area (fig. A.1a). Let denote A_c as the total area of contact, N the normal force and F the tangential force due to friction. Figure A.1b represents a magnification of one portion of the contact area. The protuberance of each surface are called asperities. At this level, the two bodies are in contact only over a portion of the total area. The sum of those microscopic contact surfaces is called the *true area of contact* A_t as opposed to the *apparent area of contact* A_c. The asperities may have different height, slopes and radius of curvature of the tip. In addition, the slopes may differ in different directions.

With this description of the topography of contact in mind, an intuitive explanation of friction is possible. The apparent stress at the contact surface may appear relatively low. However, at the true contact sites, the stresses may become extremely high. Under these circumstances, two phenomena may occur. For one, if the materials are molecularly compatible, the high compressive stresses may reduce the separation to a level where the intermolecular distance is low enough for the appearance of Van der Waals forces. Therefore, friction at the contact of two clean metallic surface may be the result of a phenomena similar to healing in fatigue crack growth. However, this is not dominant in every frictional contacts and explains exclusively the static friction forces.

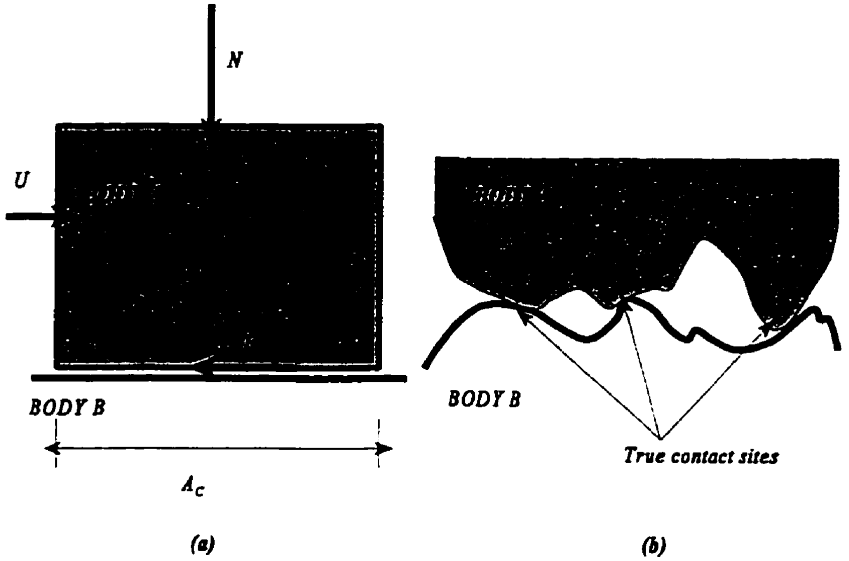


Figure A.1 - The Topography of Contact

The other possible consequence of the high true contact stresses is plastic yielding. The stresses may easily reach the yield strength of one or the other materials in contact. From there, assuming the normal loading constant, any increase in tangential loading would be equivalent to an increase in the number of true contact sites reaching plasticity limits. When all the true contact sites reach that limit, breakaway occurs and sliding is initiated. This observation is in agreement with the work of Rabinowicz [B.3] and Dahl [B.4] who suggested microscopic displacement occurring in static friction called the Dahl effect. In addition, this analysis could explain the dissipative behaviour of friction since plastic deformation is a dissipative process.

The asperity model can explain other important observations on the macroscopic behaviour of friction. For example, when operating under static friction, the limits of plasticity are not reached for all true contact sites. Consequently, assuming that the materials obey Hook's law in the elastic region, a spring-like behaviour is observed (Dahl [B.4]). In can also explain the reduction in friction in the kinetic friction regime. During sliding, some asperities are brought in contact and are deformed up to the plasticity limit. In that process however, some asperities that were plastically deformed are freed, releasing a certain amount of elastic strain energy that can be partly used to deform newly contacting asperities. This may explain the difference that exists between static and kinetic friction. Finally, the asperity model can easily explain the fundamental laws of friction. The increase in normal loading is reflected at the asperity level by an increase in number of contact sites and true contact surface. As a result, the amount of tangential force necessary to bring all contact sites to the plasticity limit is also increased.

The study of the topography of different surfaces was later measured using precise profilometric instruments and demonstrated the validity of the asperity model (Majumbar[B.5], Greenwood and Williamson[B.6], Mann[B.7] and Bailey[B.8]). Throughout the years, researchers have used contact mechanics to study the asperity contact problem. Various models were developed using Elasticity and Elasto-plasticity approaches (Rigney[B.9], Avitzur and al.[B.10,B.11] and Chang and al.[B.12,B.13].

B.2 - Lubricated Friction

Lubrication has been used for a long time for reducing friction and wear. Obviously, lubrication and tribology are very closely related. The analysis of wear in mechanisms demonstrates that wear is essentially a result of friction. Again for industrial and military purposes, the problem of lubrication was studied in great details in the XIXth century. A good approach for the understanding of lubrication in frictional contacts is through the analysis of the Stribeck curves relating the relative velocity between the contacting surfaces and the friction force for constant normal loads.

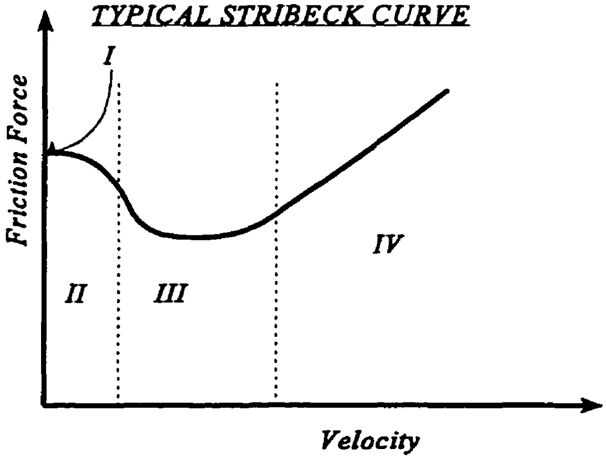


Figure A.2 - Typical Stribeck Curve

A typical Stribeck curve is represented in figure A.2. There exists four distinctive regions to the Stribeck curve (See Armstrong and al. [B.1]). Region I is called the stiction region. In this regime, the lubricant fills up the void at the contact surface without preventing the contact of asperities. Consequently, the friction can be explained using the same argument as those used for static dry contacts. Dahl effects are present in the stiction regime. Region II is the boundary lubrication region. The lubricant, trapped in the gaps between asperities, moves to the contact sites and forms along both surface a boundary layer preventing rubbing between the molecular level asperities. The result is a decrease in friction associated with the flow development of the lubricant. Region II is the partially lubricated region. As the flow develops, the fluid film increases in thickness, pushing apart the two surfaces and reducing the amount of asperities that may come in contact. However, as the flow is developing, the shear stress present in the film becomes the dominant effect of friction. As a result, friction keeps on reducing up to a point where the reduction in asperity contacts becomes insignificant with respect to the increase in shear stress. Finally, region IV is the full fluid lubrication region. The flow is fully developed and the dominant effect on the variation of the friction force is the shear stress in the fluid, proportional to the relative velocity for Newtonian fluids. The residual Coulomb friction force is due to the interaction present between asperities with height greater than the fluid film thickness.

B.3 - Dynamics of Friction

The analysis of friction performed so far is static. However, it was clearly demonstrated that some phenomena associated with friction are dynamic in nature. Some researchers, Rabinowicz[B.3], Dahl[B.4], and Hess and Soom[B.14] in particular, performed experimental studies that suggested some behaviours that clearly demonstrated friction to be dynamic in nature. The most important experiment is the stick-slip experiment of Rabinowicz [B.3]. A schematic representation of the experiment is shown if figure A.3a. It simply consists of a slider attached to a fixed wall with a spring and a damper, and resting on a conveyor. The conveyor motion is then controlled in velocity to provide some measurement of the friction force. Ideally, the damper should not be used to obtain the simple relation $F_f=K_s\Delta x$. However, the vibrations induced by the relative motion of rough surfaces practically make impossible such a simplification. Consequently, a damper is introduced to limit the bandwidth of the slider longitudinal motion. However, great care in the selection of the damper is needed. Indeed, the bandwidth of the mechanical system should not be too restrictive to interfere with phenomena attributed to friction (see Bell and Burdekin[B.15]). The damper should be selected such that 1) the force it generates is negligible when compared to the spring force and 2) such that stick-slip is not avoided.

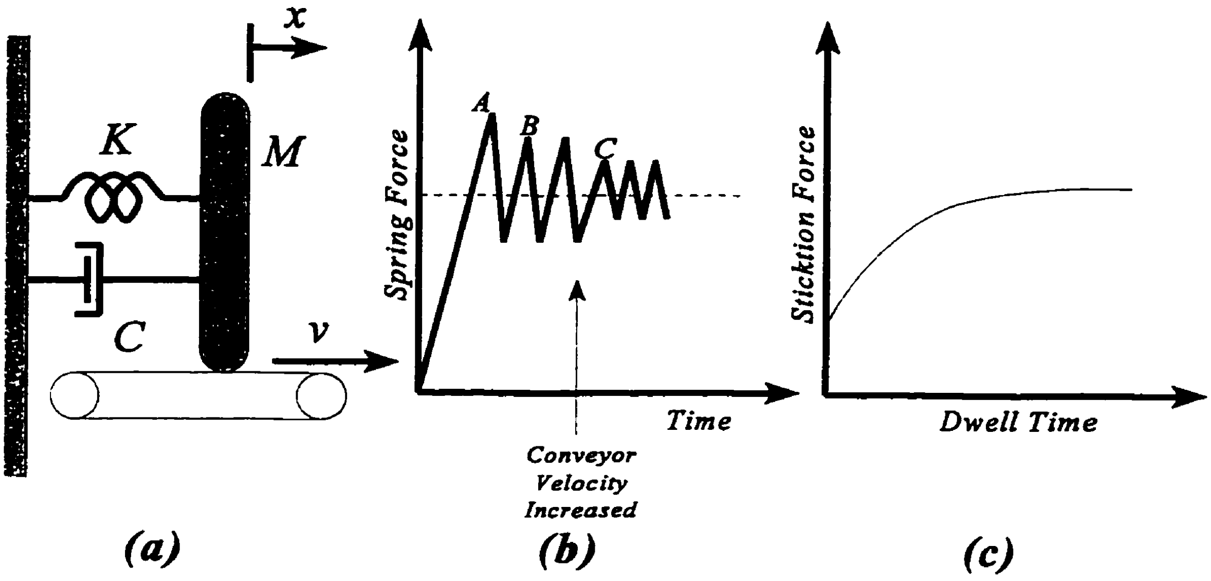


Figure A.3 - Stick-Slip Experiment

A typical spring force response (position) is presented in figure A.4b. At first, the static friction is dominant and the force builds up in the spring. When the breakaway force is reached, the slider starts to move backward due to the reduction in the friction force associated with kinetic friction. When the relative motion between the slider and the conveyor reaches zero, the slider sticks and the process is repeated. If the damper is selected to be such that the position at which stiction reoccurs is near the equilibrium between spring force and kinetic friction, the stick-slip process is avoided. From the static analysis presented in the preceding sections, this explanation makes sense. However, it cannot predict the difference between the point A, B and C. In fact, it has been observed that the maximum stiction force is a function of the dwell time. Consequently, the level of stiction present at the first breakaway (point A) is higher than under steady state operation (point B). Also, when the speed of the conveyor is increased, the stiction force is reduced (point C). If it is increase to a certain level, stick-slip disappears. A similar result is obviously reached if the spring is stiffened. These observations cannot be explained using the static arguments of the preceding section. So far, nobody had succeeded to fully describe the process behind the dwell time dependency of stiction. On the other hand, many lubricant companies had studied empirically and a great deal of empirical data is available.

Another obvious presence of dynamics in friction is called the friction hysteresis. Experimentally, one may modulate the sliding velocity of the form $v=\epsilon+A\sin(\omega t)$ where ϵ is a small number and record the friction force as a function of velocity. A frequency dependent hysteresis occurs near the zero velocity zone. This can be explained only using a dynamic model.

B.4 - Friction Models

There exists two major types of models of friction: the static models and the dynamic models. The most common static model is the Coulomb model described by,

$$F_{friction} = \begin{cases} -F_k \, sgn(v_{rel}) + f_v v_{rel}, & if\ v_{rel} \neq 0 \\ -F_k \, sgn(F_{app} \bullet e_t), & if\ v_{rel} = 0\ and\ |F_{app} \bullet e_t| \geq F_s \\ -F_{app} \bullet e_t, & if\ v_{rel} = 0\ and\ |F_{app} \bullet e_t| < F_s \end{cases} \qquad \text{(A.22)}$$

where v_{rel} defines the relative velocity between the contacting surfaces, f_v defines the viscous damping coefficient, $\mathbf{e}_t$ defines the positive direction of the tangential vector, $\mathbf{F}_{app}$ is the sum of externally applied force to the contacting body, F_s is the static force of friction ($F_s=\mu_s N$), F_k is the kinetic force of friction ($F_k=\mu_k N$),N is the normal contact force, μ_s and μ_k are respectively the static and kinetic coefficient of friction. The friction vector is given by $\mathbf{F}_{friction}=F_{friction}e_t$. Usually, the Coulomb model is presented by the first condition of equation A.22. The addition of the other two conditions makes the model determinate for all speed and force conditions and is useful to explain stick-slip.

Our previous static analysis of friction have shown that in cases of lubricated contact, the Stribeck effect is important to consider. The inclusion of the Stribeck effect in a static model was done in two different ways. First, Hess and Soom[A.20] used a Lorentzian model of the form

$$F_{friction} = \left(F_k + \frac{F_s - F_k}{1 - \left(\frac{v_{rel}}{v_o} \right)^2} \right) sgn(v_{rel}) + f_v v_{rel} \qquad \text{(A.23)}$$

to replace the equation when $v_{rel} \neq 0$ in eq. A.22. In eq. A.23, the parameter v_o is an empirical constant. Second, Bo and Pavelescu[B.16] suggested the use of an exponential model of the form

$$F_{friction} = \left(F_k + (F_s - F_k) e^{-\left(\frac{v_{rel}}{v_o} \right)^{\delta}} \right) + f_v \, v_{rel} \qquad \text{(A.24)}$$

where v_{rel} and δ are empirical constants. These friction models are static and do not introduce extra states. Such models cannot, however, include all aspects of friction.

Dahl[B.4] introduced a bristle model of the form

$$\dot{F}_{friction} = \sigma v_{rel} \left| 1 - \frac{F_{friction}}{F_k} sgn(v_{rel}) \right|^{i} sgn\left(1 - \frac{F_{friction}}{F_k} sgn(v_{rel}) \right) \qquad \text{(A.25)}$$

where σ represents the rigidity of the contact and i is an empirical model. This model represents well the Dahl effect (spring-like behaviour in stiction). Its steady-state solution is the static Coulomb model with F_s=F_k. This model does not capture the Stribeck effect.

Another Dynamic model was proposed by Haessig and Friedland [B.17]. In this model, the state of friction is estimated by an bounded integrator for which the input is 0 if their is a relative velocity and the integrator is saturated, and v_{rel} if the integrator is not saturated.

$$\dot{p} = \begin{cases} 0 & if\ v_{rel} \neq 0\ and\ |p| \geq p_o \\ v_{rel} & if\ |p| < p_o \end{cases} \qquad \text{(A.26)}$$

The friction force is given by $F_{friction}$=K_r p + β $\dot{p}$. The resulting "reset-integrator model" is similar to the Dahl model and captures the same phenomena.

Finally, Canudas and al. [B.18] captured the Stribbeck effect into a bristle model similar to the Dahl model. For their model, the state of contact deformation is given by

$$\dot{z} = v_{rel} - \frac{|v_{rel}|}{g(v_{rel})} z$$

$$F_{friction} = \sigma_0 z + \sigma_1 \dot{z} + f_v v_{rel} \qquad \text{(A.27)}$$

where σ_0 defines the contact rigidity, σ_1 defines the contact damping and $g(v_{rel})$ is set to achieve the desired steady-state solution to the dynamic friction model. For example, to include the Stribbeck effect, one selects

$$g(v_{rel}) = \frac{1}{\sigma_0} \left(F_k + (F_s - F_k) e^{-\left(\frac{v_{rel}}{v_o}\right)^{\delta}} \right) \qquad \text{(A.28)}$$

describing a Bo&Pavelescu exponential model.

B.5 - References

B.1 Armstrong, B, Dupont, P and Canudas de Wit., C., "A Survey of Models, Analysis Tools and Compensation Techniques for the Control of Machines with Friction", *Automatica*, vol 30, no. 7, pp. 1083-1138, 1994.

B.2 Bowden, F.P. and Tabor, D., "The Area of Contact Between Stationary and Moving Surfaces", *Proc. Royal Soc., Series A*, London, vol. 169, pp. 391-413, 1939.

B.3 Rabinowicz, E., "Stick and Slip", *Scientific Americ.*, vol 194, no. 5, pp. 109-118, 1956.

B.4 Dahl, P.R., "A Solid Friction Model", TOR-158, The Aerospace Corp., pp.3107-3118, 1968.

B.5 Majumbar, A. and Bhushan, B., "Fractal Model of Elastic-Plastic Contact Between Rough Surfaces", *ASME Journ. of Tribol.*, vol 113, no. 2, pp. 1-11, July 1991.

B.6 Greenwood, J.A. and Williamson, J.B.P., "Contact of Nominally Flat Surfaces", *Proc. of Royal Soc.*, London, A295, pp. 300-319, 1966.

B.7 Mann, J.B., Farris, T.N. and Chandrasekar, S., "Effects of Friction on Contact of Transverse Ground Surfaces", *ASME Journ. of Tribol.*, vol 116, no. 2, pp. 430-437. July 1994.

B.8 Bailey, D.M. and Sayles, R.S., "Effect of Roughness and Sliding Friction on Contact Stresses", *ASME Journ. of Tribol.*, vol 113, no. 3, pp. 729-738, October 1991.

B.9 Rigney, D.A. and Hirth, J.P., "Plastic deformation and Sliding Friction of Metals", *Wear*, vol 53, pp. 345-370, 1979.

B.10 Avitzur, B., "Interrelation Between the Asperity and the Trapped Lubricant, and Their Effect on friction Behaviour", *ASME Trans. on Mech. Des.*, vol. 10, pp. 1-11, 1987.

B.11 Avitzur, B., Huang, C.K. and Zhu, Y.D., "A Friction Model Based on the Upper-Bound Approach to the Ridge and Sublayer Deformation", *Wear*, vol 95, pp. 59-77, 1984.

B.12 Chang, W.R., Etsion, I. and Bogy, D.B., "An Elatsic-Plastic Model for the Contact of Rough

Surfaces", *ASME Journ. of Tribol.*, vol 109, no. 1, pp. 257-263, April 1987.

B.13 Chang, W.R. and Ling, F.F., "Normal Impact Model of Rough Surfaces", *ASME Journ. of Tribol.*, vol. 114, no. 2, pp. 439-447, July 1992.

B.14 Hess, D.P. and Soom, A., "Friction at a Lubricated Line Contact Operation at Oscillatory Sliding Velocities", *ASME Journ. of Tribol.*, vol 112, no. 1, pp. 147-152, 1990.

B.15 Bell, R. and Burdekin, M., "An Investigation into the Mechanism of Frictional Damping of Machine Tools Slideways", *Proc. Inst. Mech. Eng.*, Applied Mech. Group, vol 184, Part 1, no. 59, pp. 1088-1096, 1969.

B.16 Bo, L.C. and Pavelescu, D., "The Friction-Speed Relation and Its Influence on the Critical Velocity of the Stick-Slip Motion", *Wear*, vol 82, no. 3, pp. 277-289, 1982.

B.17 Haessig, D.A.J. and Friedland, B., "On the Modeling and Simulation of Friction", *Proc. ACC*, Dan Diego, CA, May 1990.

B.18 Canudas de Wit, C., Olsson, H., Åström, K.J. and Lischinsky, P., " A New Model for Control of Systems with Friction", *IEEE Trans. on Aut. Contr.*, To appear.

Appendix C

Proof of Upper Semi-Continuity

The contingent set associated to the dynamic system of equation 5.69 is given by

$$K(w) \triangleq \{z \mid z = f(w) + L(w)\alpha(w) + g(w)u\} \tag{C.1}$$

The proof consists of demonstrating that the separation between contingent sets on both sides of the discontinuity is continuous. This is depicted in figure C.1.

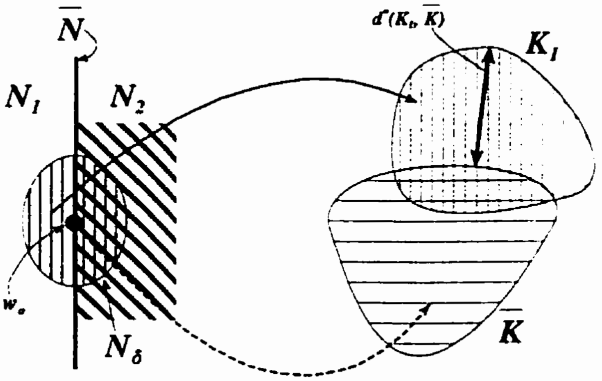


Figure C.1 - Concept of Separation

The points in a δ-neighborhood $N_\delta(w_o)$ can be in N_1 or N_2. Since $\bar{N} \subset N_2$ and $\alpha_2(w)$ is continuous on N_2 only the separation $d^*(K_1, \bar{K})$ needs to be checked for continuity.

The separation is given by

$$d^*(K_1, \bar{K}) = \sup_{u_1, \alpha_1} \inf_{\bar{u}, \bar{\alpha}} \left\| \bar{f} - f_1 + \bar{g}\bar{u} - g_1 u_1 + \bar{L}\bar{\alpha} - L_1 \alpha_1 \right\| \tag{C.2}$$

Using the triangular inequality $\|x+y\| \le \|x\| + \|y\|$ and reorganizing

$$\begin{aligned} d^{*}(K_1,\bar{K}) &\leq \left\| \bar{f}-f_1 \right\| + \sup_{u_1,\alpha_1} \inf_{\bar{u},\bar{\alpha}} \left\| \bar{g}(\bar{u}-u_1)+(\bar{g}-g_1)u_1+\bar{L}\bar{\alpha}-L_1\alpha_1 \right\| \\ &\leq \left\| \bar{f}-f_1 \right\| + \left\| \bar{g}-g_1 \right\| \rho_u + \sup_{u_1,\alpha_1} \inf_{\bar{u},\bar{\alpha}} \left\| \bar{g}(\bar{u}-u_1)+\bar{L}\bar{\alpha}-L_1\alpha_1 \right\| \\ &\leq \left\| \bar{f}-f_1 \right\| + \left\| \bar{g}-g_1 \right\| \rho_u + \left\| \bar{g} \right\| \sup_{u_1} \inf_{\bar{u}} \left\| \bar{u}-u_1 \right\| + \sup_{\alpha_1} \inf_{\bar{\alpha}} \left\| \bar{L}\bar{\alpha}-L_1\alpha_1 \right\| \end{aligned} \tag{C.3}$$

Since, the inferior norm of ($\bar{\mathbf{u}}$ - $\mathbf{u}_1$) is zero (**u** is in a ball B_u for all $\mathbf{w} \in N_1 \cup N_2$) and $\alpha(\mathbf{w})$ is bounded as described in equation 5.71, the following statement also holds.

$$d^{*}(K_1,\bar{K}) \leq \left\| \bar{f}-f_1 \right\| + \left\| \bar{B}-B_1 \right\| \rho_u + \left\| \bar{L}\rho_2-L_1\rho_1 \right\| \tag{C.4}$$

Therefore, since *f*, *g* and *L* are continuous, given any ϵ_1, ϵ_2 and ϵ_3, there exists δ_1, δ_2, δ_3>0 such that

$$\begin{aligned} &\left\| \bar{w}-w \right\| < \epsilon_1 \Rightarrow \left\| \bar{f}-f_1 \right\| < \delta_1 \\ &\left\| \bar{w}-w \right\| < \epsilon_2 \Rightarrow \left\| \bar{B}-B_1 \right\| \rho_u < \delta_2 \\ &\left\| \bar{w}-w \right\| < \epsilon_3 \Rightarrow \left\| \bar{L}\rho_2-L_1\rho_1 \right\| < \delta_3 \end{aligned} \tag{C.5}$$

Thus, given any $\epsilon=\epsilon_1+\epsilon_2+\epsilon_3>0$, there exists $\delta=\min(\delta_1, \delta_2, \delta_3)$ such that

$$\left\| \bar{w}-w \right\| < \epsilon \Rightarrow \left\| \bar{f}-f_1 \right\| + \left\| \bar{B}-B_1 \right\| \rho_u + \left\| \bar{L}\rho_2-L_1\rho_1 \right\| < \delta \tag{C.6}$$

Consequently, the separation is continuous in any δ-neighborhood of any point $w \in \bar{N}$ which implies that the contingent set *K is USC* $\forall w \in \bar{N}$.

Appendix D

Robot Dimensions

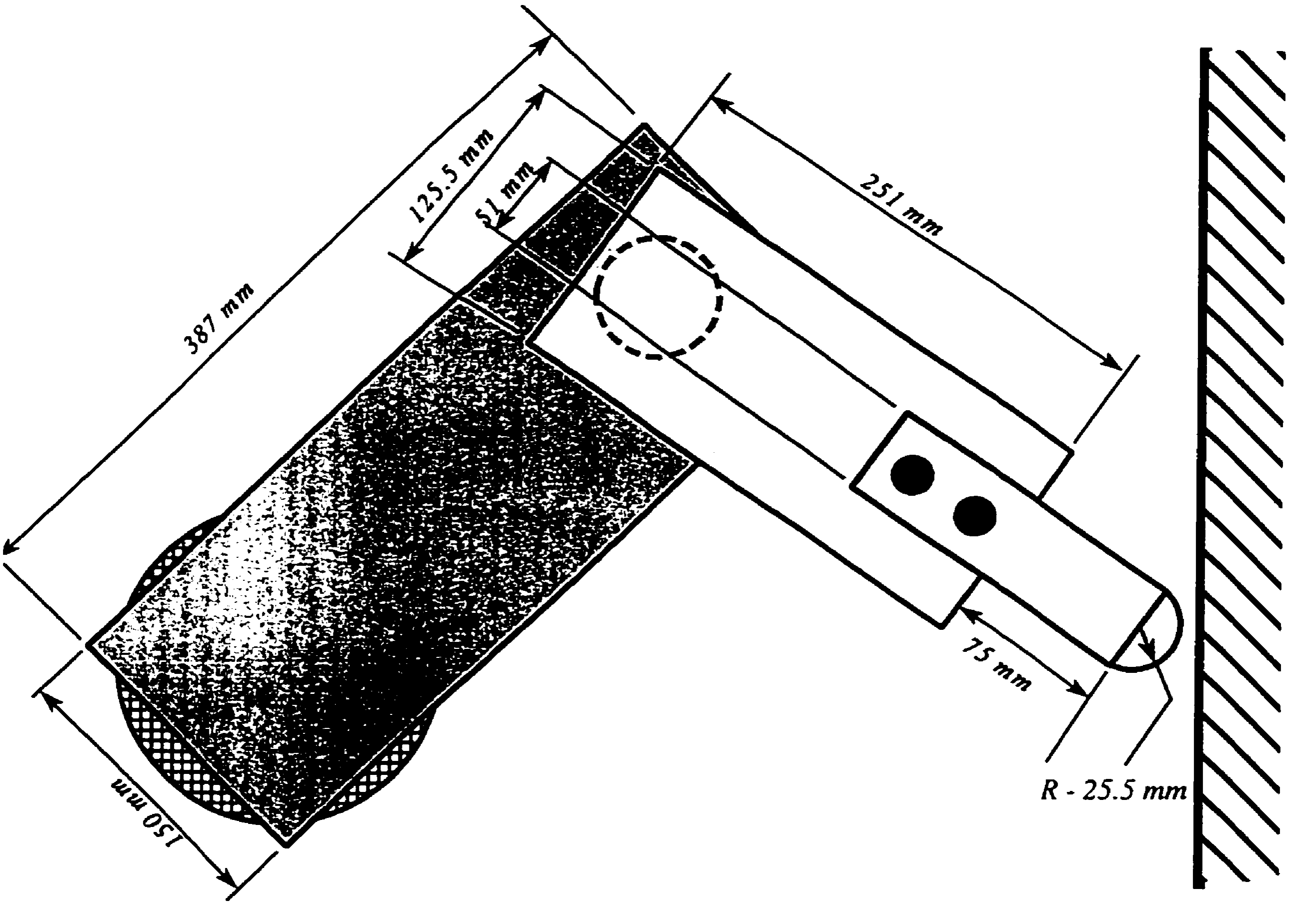


Fig. D.1 - Top View

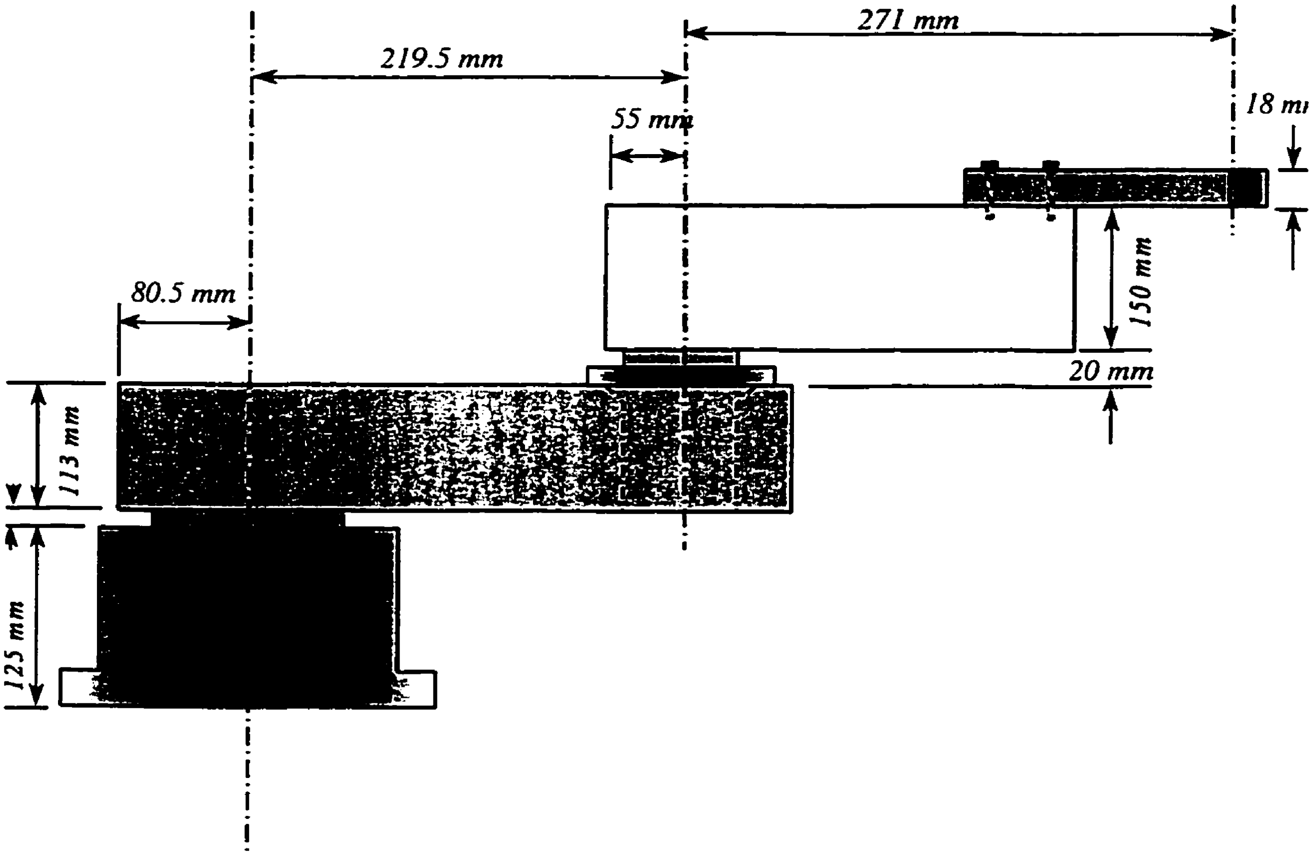
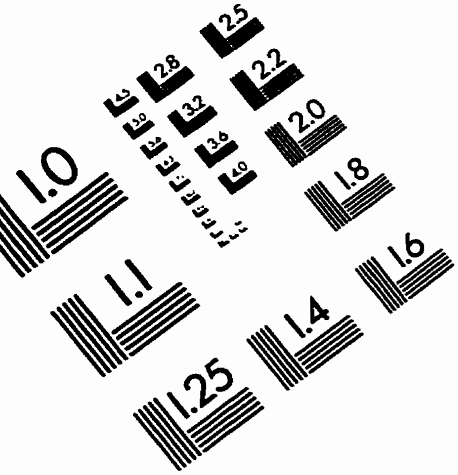
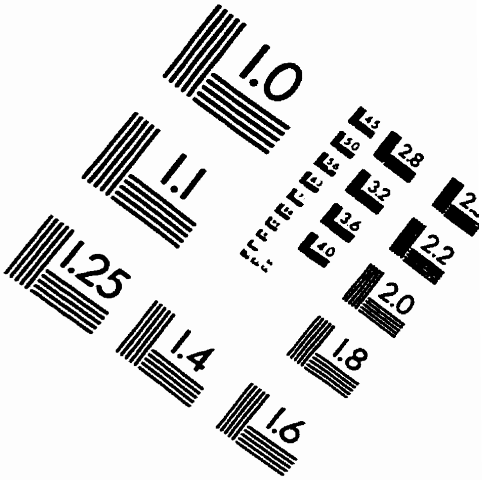
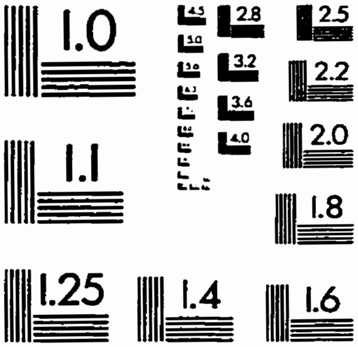


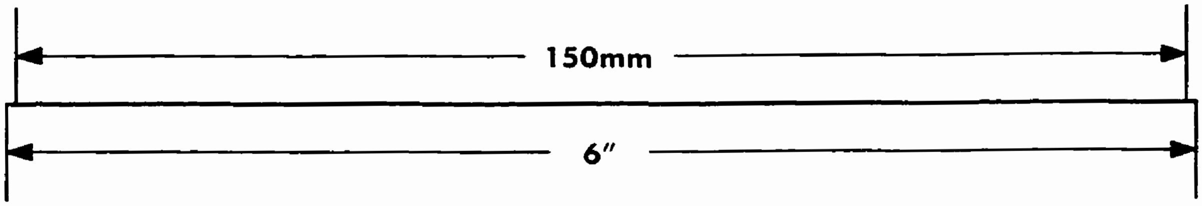
Figure D.2 - Side View

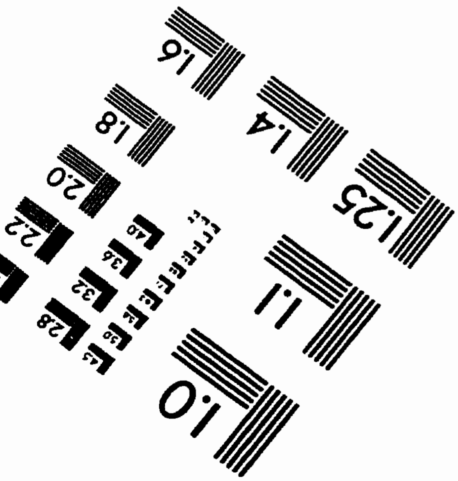
IMAGE EVALUATION TEST TARGET (QA-3)











APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989