ISBN 0-315-56404-0

Analysis and Control of Spring Counterbalanced Robot Arm

Thesis Submitted
to the school of Graduate Studies
as partial fullfilment for the requirements of the
degree of Masters of Applied Science

By Amor Jnifene

at the
Department of Mechanical Engineering
University of Ottawa

*This work is dedicated to my Mother, my Father,
and all members of my caring family*

Abstract

Robot manipulators, operating under the gravity effect are currently balanced by the addition of counterweights. This method, in general leads ˈo a complete static balancing for a specific payload, however the overall inertia of the manipulator increases significantly. This high inertia induces significant effects on the power consumption as well as on robot control.

This work deals with the problem of balancing by springs. These springs, when integrated with the non-linear motion of the manipulator, store a massive amount of energy which is used to counteract the effect of gravity.

Two parts are presented in this study. The first part deals with the analysis of static balancing using spring mechanisms. The second part deals with the control of a three revolute joint manipulator. The advantages of using spring mechanisms over the use of counterweights have been outlined and a comparison between the performance a spring balanced three revolute arm and a weight balanced one has been carried out. The two arms have been tested for two types of control task. The first task is a PTP control task and the second is a trajectory tracking task.

Acknowledgements

The author wishes to express his gratitude to Dr. A. Fahim for his continuous encouragement, invaluable suggestions, and patient guidance throughout the research.

Special thanks to the Tunisian students at the University of Ottawa for their moral support and encouragement during the study.

The financial support received from the Tunisian government through the University Mission of Tunisia in Washington is gratefully acknowledged.

Contents

List of Figures

Nomenclature

AB	=	Elongation of the zero free length spring
a_i	=	Length of link i
b_i^l	=	Lower limit of parameter i
b_i^u	=	upper limit of parameter i
c_i	=	$\cos\theta_i$
c_{i-j}	=	$\cos(\theta_i - \theta_j)$
$E(\theta)$	=	Torque unbalance
$F(X)$	=	objective function (IAE)
T_s	=	Force due to spring
$\mathbf{G}$	=	vector containing the gravity terms
g	=	acceleration due to gravity
$\mathbf{H}$	=	Manipulator inertia matrix
H_{ij}	=	element ij of the $\mathbf{H}$ matrix
$\mathbf{h}$	=	vector of the coriolis and centrifugal terms
h_i	=	element i the $\mathbf{h}$ vector
I_{ix}	=	inertia of link i w.r.to its center of mass about the x axis
I_{iy}	=	inertia of link i w.r.to its center of mass about the y axis
I_{iz}	=	inertia of link i w.r.to its center of mass about the z axis
I_{max}	=	maximum current
I'_{ix}	=	inertia of the counter weight of link i w.r. to its center of mass about the x axis
I'_{iy}	=	inertia of the counter weight of link i w.r. to its center of mass about the y axis

I'_{iz}	=	inertia of the counter weight of link i w.r. to its center of mass about the z axis
J	=	jacobien matrix
K	=	total kinetic energy
K_{ai}	=	current amplification of actuator i
K_{bi}	=	back emf constant of actuator i
K_{di}	=	derivative constant of controller i
K_{mi}	=	torque constant of actuator i
K_{pi}	=	proportionality constant of controller i
K_i	=	constant of spring i
L	=	Lagrangien
L	=	spring length
L_0	=	spring free length
l_c	=	Length of segment c
l_d	=	Length of segment d
L_i	=	inductance of actuator i
l_i	=	distance from joint i to the center of mass of link i
l'_i	=	distance from joint i to the center of mass of counterweight i
L_x	=	component of the spring length along the x axis
L_y	=	component of the spring length along the y axis
m_i	=	mass of link i
m_{lb}	=	load at which the robot is balanced
m'_i	=	mass of the counterweight i
N_i	=	gear ratio of actuator i
OA	=	vertical distance from the joint to one of the spring

OB = distance along the link from the joint to the other end of the spring

Q = generalized force

R_a = Radius of gear A

R_b = Radius of gear B

R_i = armature resistance of actuator i

$\vec{r}$ = position vector of a point w.r.to the reference frame

T_{max} = maximum time to perform the desired trajectory

s_i = $\sin\theta_i$

s_{i-j} = $\sin(\theta_i - \theta_j)$

$\bar{V}_i$ = linear velocity of the center of mass of link i

$\bar{V}_l$ = linear velocity of the center of mass of the payload

$\bar{V}_i'$ = linear velocity of the center of counterweight i

X = design vector for the optimization parameters

β_c = angle between link c and the horizontal

β_d = angle between link d and the horizental

ω = angular velocity

ω_n = natural frequency

θ_i = angular position of joint i

$\dot{\theta}_i$ = angular velocity of joint i

$\ddot{\theta}_i$ = angular acceleration of joint i

ζ	=	damping ratio
τ	=	vector of the actuator's torques
$\tau_{gravity}$	=	Torque due to gravity
τ_i	=	torque at joint i
τ_{ps}	=	peak torque for the spring balanced arm
τ_{pw}	=	peak torque for the weight balanced arm
τ_{spring}	=	Torque due to the spring force

Chapter 1

INTRODUCTION

1.1 General Introduction

A Robot, as defined by the RIA (Robot Institute of America), is a reprogrammable multi-functional manipulator designed to move parts, tools or specialized devices, through variable programmed motion for the performance of a variety of tasks.

Robot arms are constructed from mechanical links assembled together using rotary or linear joints to perform the required motion. Four basic configurations of robot arms are available, rectangular, cylindrical, spherical, and articulated or jointed. Figures 1.a to 1.d show these four basic configurations respectively. In order to be able to completely position

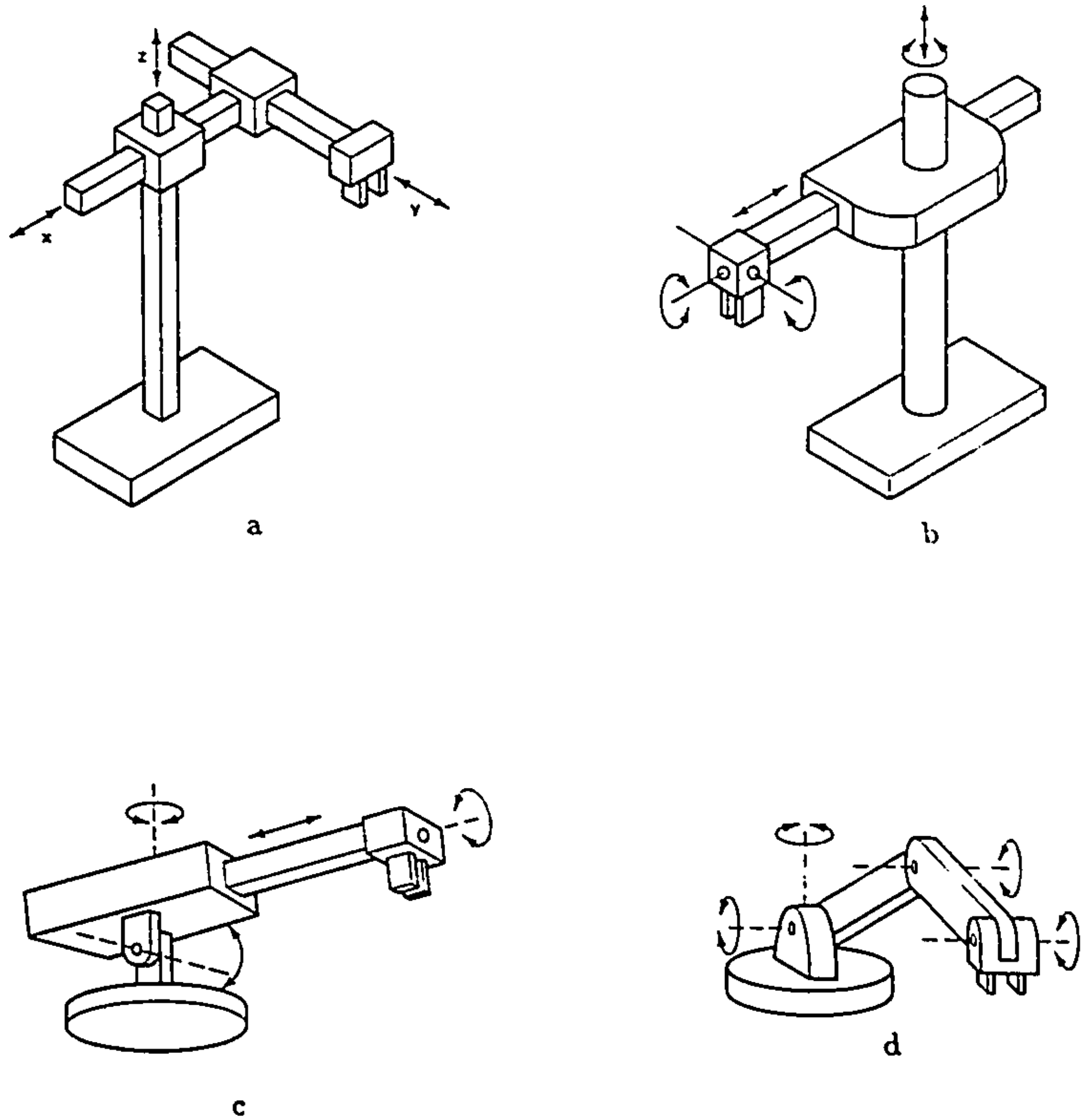

Figure 1.1: Basic Robot configurations

and orientate the hand (end effector) at any point in the work volume, six degrees of freedom (DOFs) are required. These six DOFs can be divided into two groups. The first three position the end effector at a required location in space, and are referred to as the primary DOFs. The remaining three are used to orient the end effector and are referred to as the secondary DOFs. Each DOF is driven by an actuator. Electrical actuators are used for small robots intended to deal with light loads while hydraulic actuators are used for large robots. For robots intended to locate a load at prescribed locations, pneumatic actuators are used, and the locations are programmed using physical stops.

In order for the robot arm to have a useful reach to position the load in space, the links of the primary DOFs are made to be large. The secondary DOFs are constructed using short links and only rotary joints. As a consequence of this type of construction, most of the weight of the robot arm is concentrated in the primary DOFs.

1.2 Balancing Concept

Mechanical systems, working in the gravity field are subjected to external forces due to the weight of each of its components. Industrial robots - as a class of these mechanical systems - consist of an open chain mechanism with several links. Each link is essentially a cantilever beam mounted on a preceding one. When an unbalanced link is attached to the previous one with a rotary joint, and its plane of motion is parallel to the gravity field, it will experience a force due to gravity pull.

A robot must be able to carry its own weight, which is often much larger than the payload, to do this, the joint motors have to provide sufficient torques for the following two situations:
- At rest, the robot must be able to maintain a given position despite the effect of gravity.
- When the robot is moving, the actuators driving each link have to provide torques that vary depending on the configuration and the direction of motion. The torques due to the dynamics of the links are often smaller than those required to counteract the effect of gravity.

In order to reduce the total amount of torque provided by the actuators, the gravity torque needs to be balanced. The balancing can be exact or approximate depending on how the balancing forces vary depending on the mechanism configuration. For a fixed load, exact balancing of a link brings the center of gravity CG to coincide with the joint axis, while approximate balancing moves the CG nearer to the joint axis. In the case of a robot performing a pick and place operation, the mass of the payload varies from one part of the cycle to another and an adaptive or variable balancing system is required.

The most commonly used method of balancing a link consists of the addition of counterweights. Figure 1.2 and 1.3 show two methods of balancing an R-R-R manipulator by means of counterweights.

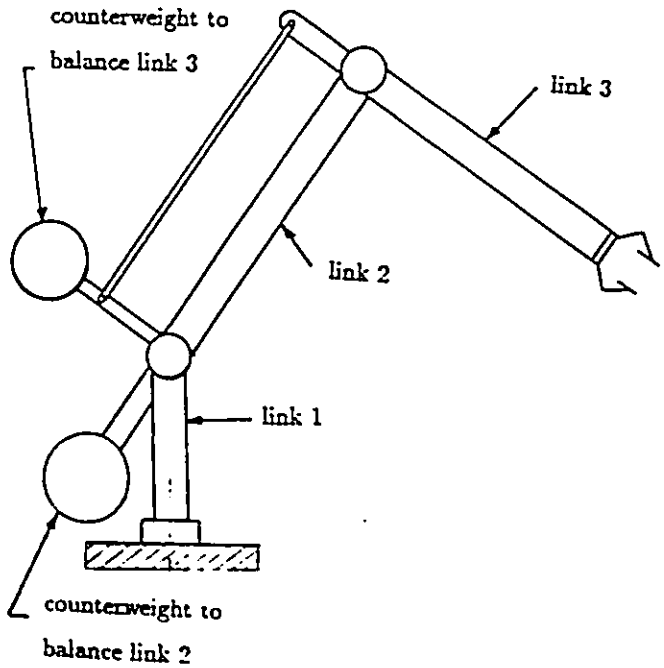

Figure 1.2: Balancing with counterweight in a parallel drive

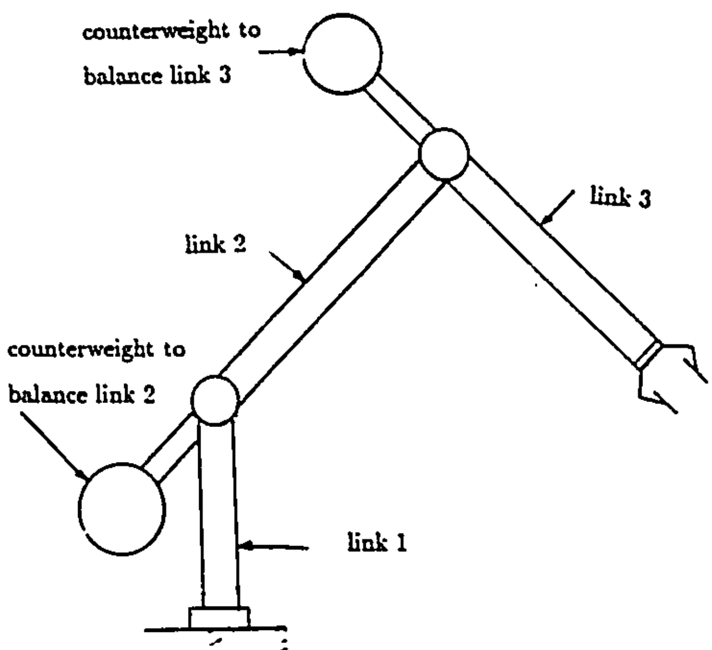

Figure 1.3: Balancing with counterweight in a serial drive

The counterweight approach leads to an increase in the total inertia and weight of the system. Although high inertia robot links are generally undesirable, because of their sluggish dynamic performance, in most industrial applications the benefits of balancing using counterweights outweigh the drawbacks.

Alternative methods have b n proposed and used to overcome the balancing problem. The most used alternative method entails the use of hydraulic piston. A method using spring mechanisms has also been reported [7].

The aim of this work is to achieve balancing without increasing the overall inertia of the system. The method of balancing presented here uses the potential energy storage capabilities of linear springs. When the linearly dependant spring force is transformed with a nonlinear moving mechanism, it results in a force that closely resembles the effect of gravity of a robot link. Balancing using springs results in a better dynamic performance of the manipulator as well as an increase in the load carrying capability. For a given energy consumption, there will be more torque available from the actuators driving the spring-balanced link. This torque can be utilized to carry heavier loads or to enhance the operational speed of the robot.

1.3 Thesis Outline

This thesis deals with the analysis and control of a spring balanced robot arm. The analysis focuses on the static balancing of the links of a robot with 3 revolute joints. Different methods of balancing using springs are analyzed. The performance of a robot and its controller are studied for the two cases: weight counterbalanced and spring counterbalanced links.

A brief review of the use of linear springs as means of balancing of linkages is presented in Chapter 2. Examples of mechanisms which are balanced using springs, are also presented in the chapter.

In Chapter 3 , an analysis of a spring balanced one link manipulator is given. Two mechanisms are presented: A spring-gear mechanism, and a zero free length spring mechanism. The analysis is concentrated on the zero free length spring mechanism since it provides complete balance of the torques due to gravity.

Chapter 4 deals with the balancing of a three revolute joint manipulator using the zero free length spring mechanism described in Chapter 3. The analysis is carried out for serial as well as parallel drive robots. The chapter also presents the relation required to adapt the balancing mechanism to different payloads.

In Chapter 5, a dynamic model of the three revolute joint manipulator has been derived using the Lagrangian formulation. The two methods of balancing using counterweights and spring mechanisms are analyzed. The advantages of using spring mechanisms over counterweights as evident from the dynamic equations have been outlined.

Chapter 6 deals with the different methods of robot control. Two methods are discussed in this chapter. The first uses an individual joint controller approach. The second method uses the computed torque technique for tracking a required trajectory.

Results of a dynamic simulation of a three revolute joint manipulator are presented in Chapter 7. The dynamic model is solved numerically using the Runge-Kutta integration routine. The time responses and the torques developed by the three joint actuators for the weight counterbalanced and the spring counterbalanced manipulator are being compared.

Chapter 8 consists of conclusions drawn based on the analysis as well as recommendations for further research.

Chapter 2

LITERATURE REVIEW

The need for statically balanced systems with low inertia is a major concern of most designers dealing with the dynamics of these systems. The use of linear springs to achieve this goal has been investigated by several authors.

Harmening[1] described a method for balancing a four bar linkages by using a torsion spring. He found that the linkages are operable up to 120° with torque errors of less than 0.2 %, and up 180° with torque errors of less than 2 % of the peak unbalance torque.

G. K. Matthew and D. Tesar [2,3] used a synthesis techniques to select spring parameters in order to satisfy specified energy levels in planar mechanisms. Balancing was done for a specific number of points in the mechanism space. Then they used the same technique in order to balance a general forcing functions acting on a planar mechanism. Their method relies on the potential energy storage capabilities of linear springs when integrated with a nonlinear motion of a given mechanism.

In order to make the work developed by Mathew and Tesar more general, Lu. Zhen [5] presented an algebraic method for dynamic synthesis of spatial mechanisms. The method also uses spring elements to balance the kinetic energy fluctuation or loading fluctuation in mechanisms and/or connected machinery systems. It has been also shown that this method can be used to obtain a desired dynamic response.

R. H. Nathan [4] presented a mechanism that generates a constant and unidirectional force in order to counteract the effect of gravity. He further suggested that for the case of varying load on the mechanism, the geometric parameters may be adjusted to achieve balancing by using a force feedback from the output load. Several mechanisms have been presented. The most practical example is the full sized model of a surgical laser shown in Figure 2.1. With reference to the Figure, a compression spring (1) is placed inside a segment AC, (2) is a cable that transfers the relative motion between B and D to a sliding ring (4) after passing over two pulleys (3). The sliding ring compresses the free end of the spring (1). A second example of this method of balancing is shown in Figure 2.2. The counterweight of a standard record player pick up arm has been replaced by a constant force generation mechanism. It has been shown that this arrangement results in a significant reduction of the total inertia of the pick up arm which leads

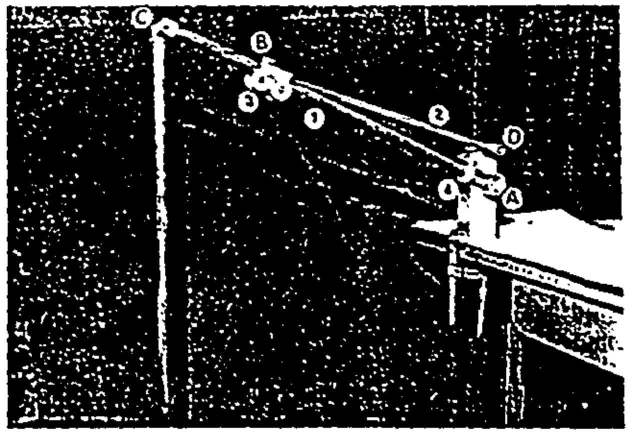

Figure 2.1: Model for a surgical laser

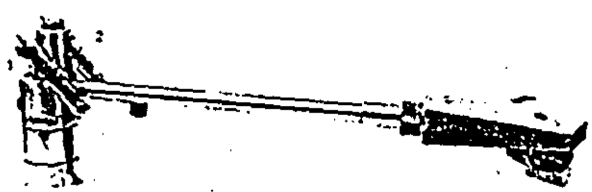

Figure 2.2: Record player pick up arm

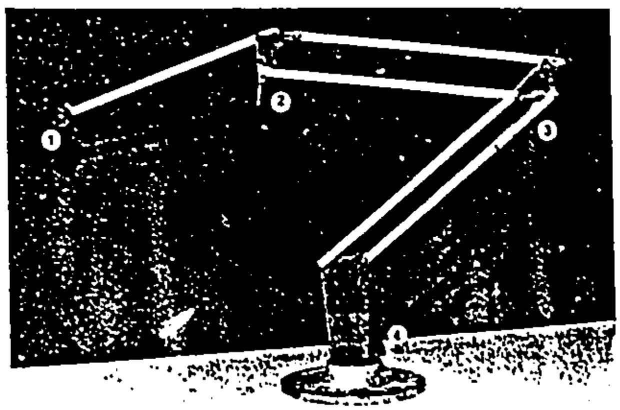

Figure 2.3: General purpose stand

to a corresponding reduction of the stylus force required to keep the needle tracking on the record.

Balancing of a three stage general purpose stand is shown in Figure 2.3. The system is designed to support a desk lamp. At stages (2), (3) and (4), helical springs are used to counteract the gravity effect on the links.

Tokuji Okada [11] proposed link mechanisms for force generation using pulleys and springs. Figure 2.4 shows the basic of the mechanism used in the study. The mechanism is based on a scissor structure having two arms AB and AC with the same length and having free wheels on their ends. One end of each of the two arms are put together to have three wheels totally. θ represents the angle between the two arms, F is the pressing force

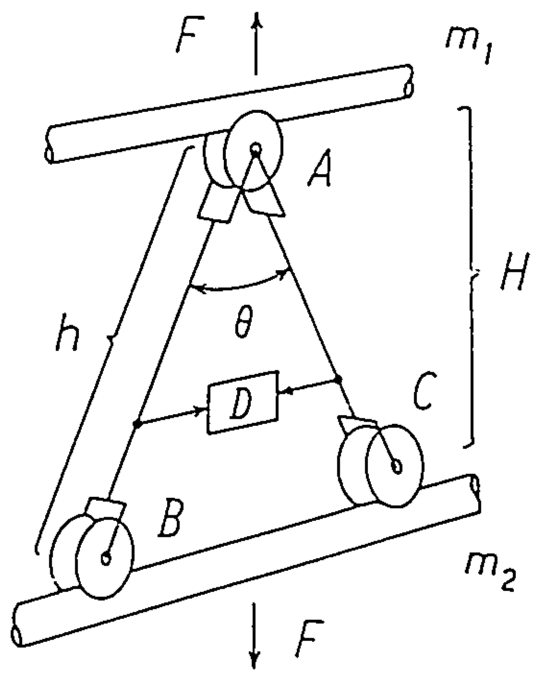

Figure 2.4: Basic force generation mechanism

against probes 1 and 2, D is a linear spring to pull on the two arms and H is distance of A from the line connecting the two points B and C. Balancing is achieved by optimizing the structure and dimension of the links of the mechanism.

H. Hilpert [9] presented different methods for balancing precision mechanical instruments. Both counterweight and spring balancing mechanisms are investigated. Figure 2.5 shows a geared mechanism for the balancing of a lever by means of a spring. Another example of balancing a lever by means of a spring is shown in Figure 2.6. The counterweight used to balance the parallelogram drafting machine shown in Figure 2.7 is replaced by springs, as illustrated in Figure 2.8. Another example showing two ways of balancing a part guided through a curvilinear path is shown in

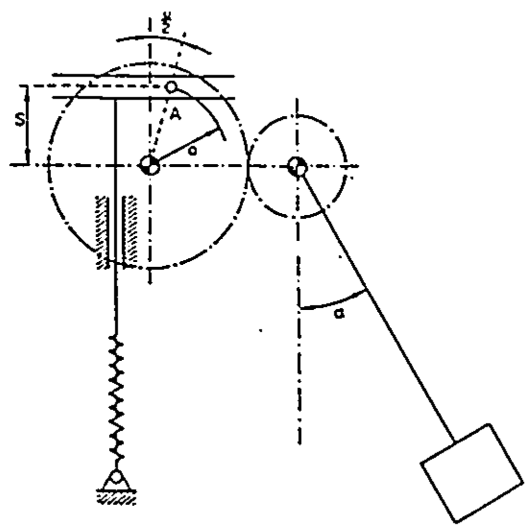

Figure 2.5: Geared mechanism for the balancing of a lever by spring

Figure 2.9 and 2.10.

Balancing of robot manipulators has been also a major concern of few authors. Yu. F. Grozdev[7] showed the advantage gained by balancing of manipulator arms. He presented two new designs of mechanisms to insure complete balancing of a manipulator link. The mechanisms require an external source of hydraulic or pneumatic power in order to operate.

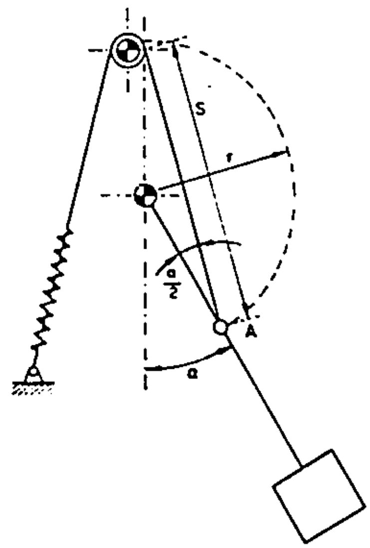

Figure 2.6: Balancing of a lever by spring

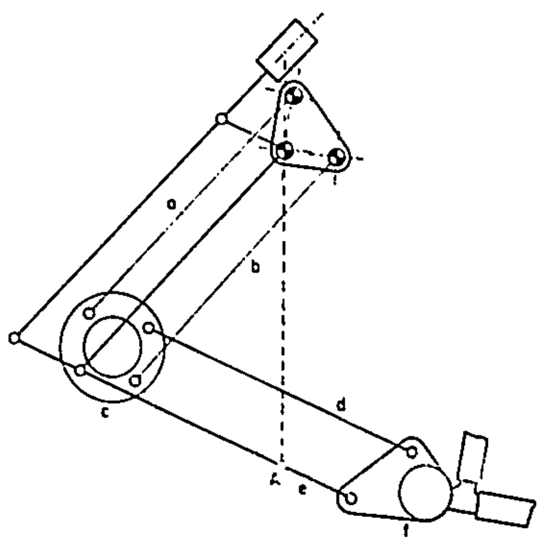

Figure 2.7: Balancing of a drafting machine by counterweight

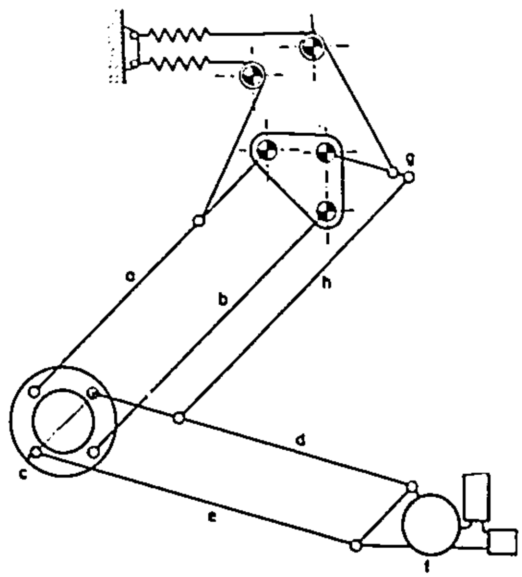

Figure 2.8: Balancing of a drafting machine by springs

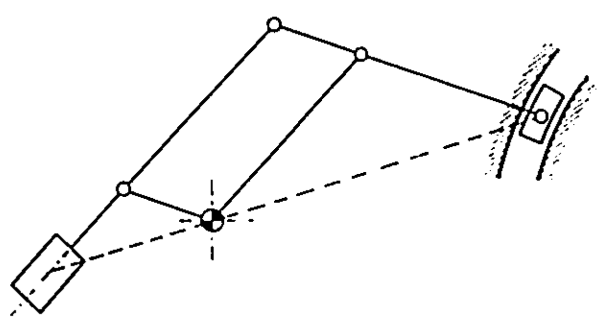

Figure 2.9: Balancing of a part guided through a curvilinear path using counterweights

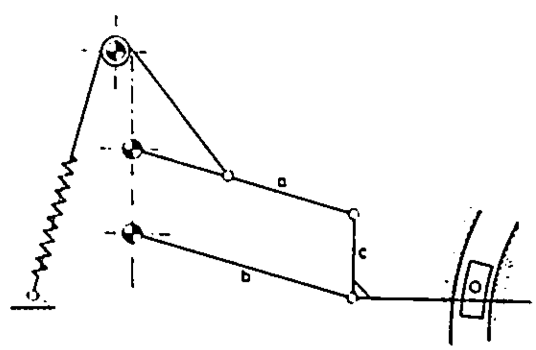

Figure 2.10: Balancing of a part guided through a curvilinear path using springs

Chapter 3

BALANCING OF MANIPULATORS

3.1 Balancing of a one Link Manipulator

In this section, balancing the gravity effect is to be studied considering only one link manipulator. A schematic representation of this manipulator is shown in Figure 3.1. The mass of the link and the load carried are assumed to be lumped at it's free end. If the friction and the deflection of the arm are to be neglected, the dynamic equation can be written as

$$(I + ml^2)\frac{d^2\theta}{dt^2} + mglcos\theta = \tau(t) \tag{3.1}$$

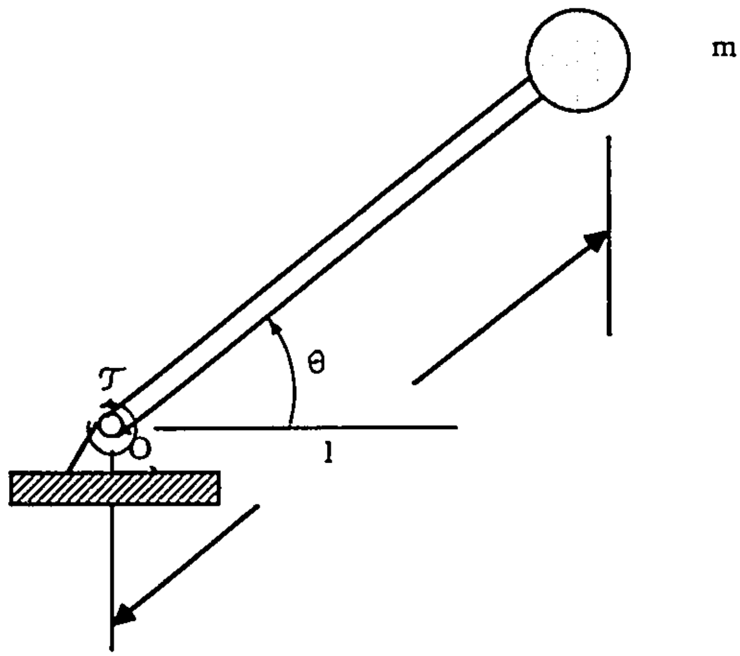

Figure 3.1: One link manipular

Where:

m : Mass of the link and the load carried

θ: The generalized coordinate

l : Length of the link

τ : Applied torque from the actuator

I : total Inertia of the mass about it's center

g : Gravitational acceleration

The term $mglcos\theta$ in the above equation represents the non-linear term caused by the gravity pull. This term is a function of the position angle θ.

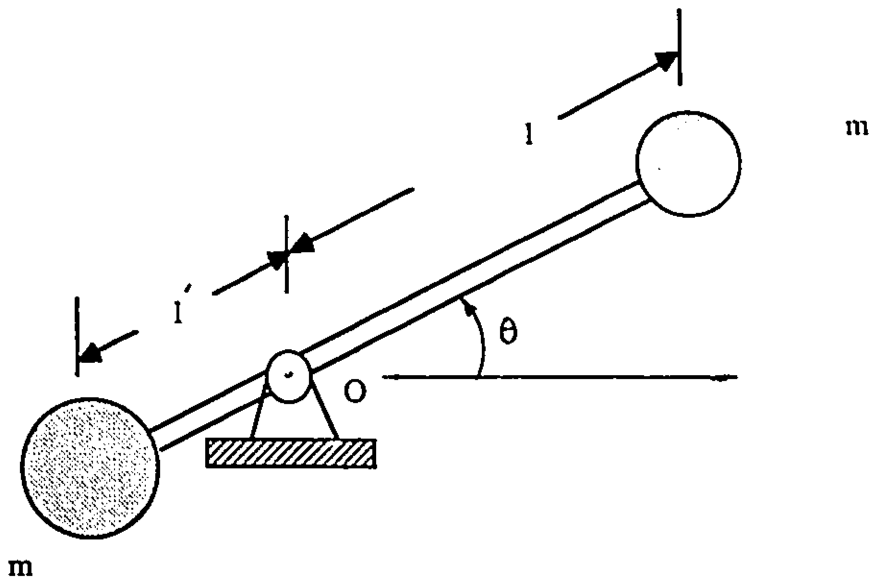

Figure 3.2: Balancing with counterweight

3.1.1 Balancing with a counterweight

One method of balancing the effect of gravity is to add a counterweight m' at a distance l' from point O such that:

$$m'l' = ml \tag{3.2}$$

The equation of motion becomes:

$$(I + ml^2)\ddot{\theta} + mlgcos\theta + (I' + m'l'^2)\ddot{\theta} - m'l'gcos\theta = \tau(t) \tag{3.3}$$

Since m' and l' are to be chosen such that Equation 3.2 is satisfied, the terms representing the gravity will cancel each other and the governing dynamic equation becomes

$$(I + I' + ml^2 + m'l'^2)\ddot{\theta} = \tau(t) \tag{3.4}$$

As shown in Equation 3.4, the effect of gravity is fully counteracted but the overall weight and inertia of the system has been increased. This increase of weight and inertia is undesirable from the control point of view because as the inertia of a system increases, its dynamic behavior becomes sluggish.

3.1.2 Balancing with springs

Balancing of one link manipulator can also be obtained by generating a force of the same magnitude but opposite in sign to the gravitational pull on the link to be balanced. This force can be generated by a spring. Most springs are manufactured as linear. In order to exactly simulate the effect of gravity, a constant force spring has to be used. Such spring is very difficult to produce. A linear spring, however, may be integrated with special mechanisms in order to simulate the non-linear effect of gravity.

Several mechanisms utilizing springs as a means for generating the balancing force have been investigated. In this chapter three of these mechanisms are analyzed

Spring Gear Mechanism (SGM)

The proposed mechanism is shown in Figure 3.3. It consists of two gears A and B, each carrying a link c and d respectively. The two links are spring loaded towards each other by a spring. One gear is attached to

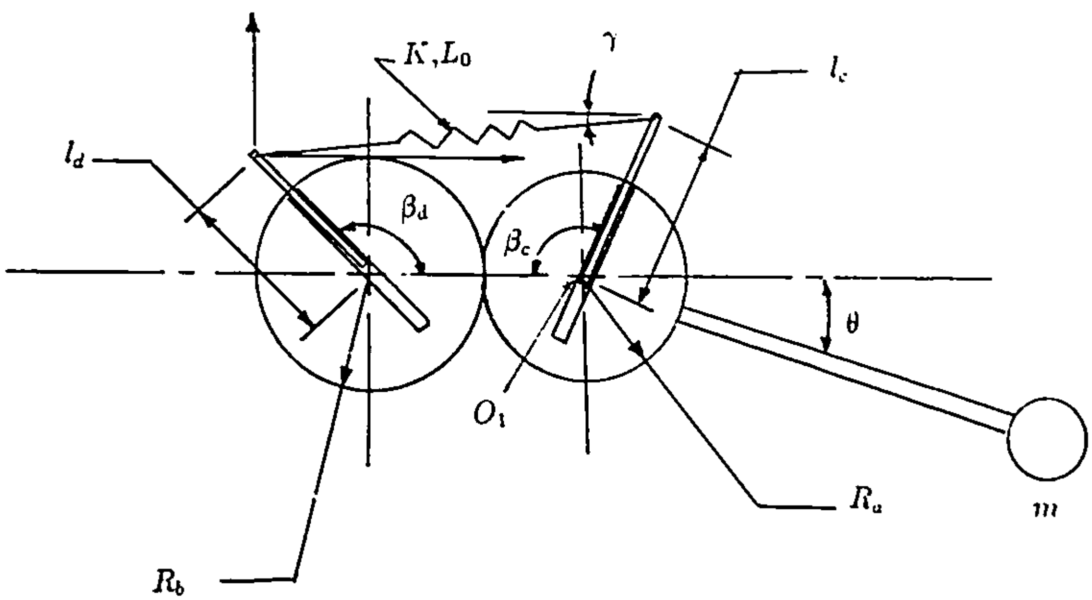

Figure 3.3: Spring Gear Mechanism

ground and the other gear is attached to the link that is to be balanced.

With reference to the figure, the length of the spring can be expressed as:

$$\begin{aligned} L^2 &= {L_x}^2 + {L_y}^2 \\ &= \{(R_a + R_b) - l_c cos\beta_c - l_d cos\beta_d\}^2 \\ &\quad + \{l_c sin\beta_c - l_d sin\beta_d\}^2 \end{aligned} \tag{3.5}$$

the tension in the spring is a function of its elongation and is expressed as:

$$T = K(L - L_0) \tag{3.6}$$

The torque exerted by the spring force at point O_1 is given by:

$$\tau_{spring} = T l_c sin(\alpha) \tag{3.7}$$

where

$$\begin{aligned} \alpha &= \pi - (\beta_c + \gamma) \\ &= \pi - [\beta_c + tan^{-1}(\frac{L_y}{L_x})] \end{aligned} \tag{3.8}$$

Substituting Equations 3.5 , 3.6 and 3.8 into 3.7 results in the expression of the torque exerted by the spring at point O_1 as a function of the parameters of the mechanism as follows:

$$\begin{aligned} \tau_{spring} &= K\{[\{(R_a + R_b) - l_c cos\beta_c - l_d cos\beta_d\}^2 \\ &\quad +\{l_c sin\beta_c - l_d sin\beta_d\}^2]^{1/2} - L_0\} l_c sin(\gamma + \beta_c) \end{aligned} \tag{3.9}$$

This torque counteracts the gravity torque and the resulting amount of unbalance between the torques is given by:

$$\begin{aligned} E(\theta) &= \tau_{spring} - \tau_{gravity} \\ &= K\{(\{(R_a + R_b) - l_c cos\beta_c - l_d cos\beta_d\}^2 \\ &\quad +\{l_c sin\beta_c - l_d sin\beta_d\}^2)^{1/2} - L_0\} l_c sin(\gamma + \beta_c) \\ &\quad -mglcos\theta. \end{aligned} \tag{3.10}$$

Balancing is achieved by choosing the approximate values of the mechanism parameters R_a,R_b,l_c,l_d,K,L_0, β_c and β_d as to minimize $E(\theta)$.

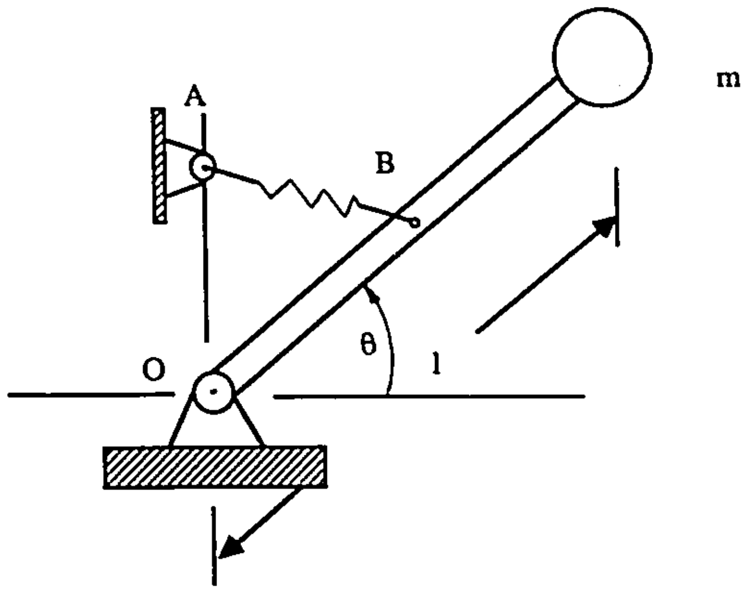

Figure 3.4: Balancing with fixed end spring mechanism

Spring with fixed end

Figure 3.4 shows a schematic diagram of this method of balancing. The mechanism to be studied consists of three links as can be seen from the figure: (a) the link to be balanced; which represents the one link manipulator, (b) a fixed link OA, (c) a linear spring with length AB.

With reference to the figure, for equilibrium at all angular orientations of the arm, the spring has to generate a force T_s such that the moment about point O is zero.

From the geometry of the mechanism, we have:

$$sin\alpha = \frac{OA.sin\theta}{AB} \tag{3.11}$$

and

$$AB = \sqrt{OA^2 + OB^2 - 2.OA.OB.cos(\theta)} \tag{3.12}$$

The tension in the spring is a function of it's elongation and is given by:

$$T_s = K(AB - L_0) \tag{3.13}$$

The unbalance of torques due to the spring torque and due to gravity about point O is given by:

$$\begin{aligned} E(\theta) &= \tau_{spring} - \tau_{gravity} \\ &= T_s \cdot OB \cdot sin\alpha - mgLsin\theta \end{aligned} \tag{3.14}$$

Solving Equations 3.11 to 3.14, we get the balancing equation expressed in terms of the mechanism parameters as follow:

$$E(\theta) = \left[K(1 - \frac{L_0}{\sqrt{(OA)^2 + (OB)^2 - 2 \cdot OA \cdot OBcos\theta}}) \cdot OA \cdot OB - mgl\right] sin\theta \tag{3.15}$$

3.1.3 Optimization of the Mechanisms parameters

An optimization technique can be used to minimize the torque unbalance as the arm moves from θ_i to θ_f.

$$E(\theta) = \tau_{spring} - \tau_{gravity} \tag{3.16}$$

Since the amount of unbalance $E(\theta)$ is to be minimized over an interval of θ, then the objective function to be minimized can be formed by integrating $E(\theta)$ over the interval as follow:

$$F(X) = \int_{\theta_i}^{\theta_f} |E(\theta, X)| d\theta \tag{3.17}$$

Where: X is the design vector containing the mechanism parameters. This objective function represents the integral absolute error (IAE) performance index. The mechanism parameters are physically constrained, and each of them is in fact bounded by two limiting values that can be chosen depending on the nature of the given parameter. The optimization problem; therefore, can be stated as follow: Minimize:

$$F(X) = \int_{\theta_i}^{\theta_f} |E(\theta, X)| d\theta \tag{3.18}$$

Subjected to:

$$b_i^{(l)} \leq x_i \leq b_i^{(u)}$$

Where: $b_i^{(l)}$ and $b_i^{(u)}$ are the lower and upper bounds of the parameter x_i, respectively.

One of the simplest procedures for unconstrained direct-search optimization is the simplex method. This method was originally proposed by Spendly, Hext and Himsworth [35]. The name simplex came from the fact that the method uses a regular simplex to search the parameter space. The method basically consists of choosing an $n+1$ mutually equidistant points in an n dimensional space. During the search, the worst vertex (point with the highest function value) is rejected and replaced by a new point which is obtained by reflecting the worst vertex across the centroid of the remaining points.

In order to deal with constraints, Spendly, Hext and Himsworth and later Nedler and Mead [35] have suggested to introduce large positive function values for all the non-feasible points so that these points will be always rejected, and this will lead the simplex method to take into account the presence of the constraints automatically.

Optimization of SGM parameters

The objective is to find a combination of the parameters of the SGM mechanism that minimizes the IAE criteria defined by equation 3.17 given the upper and lower limits of each parameter.

As an example, we consider a one link manipulator with $m = 4$ kg and $l = 0.75$ m. The limiting values are given in table 3.1

	K	R_a	R_b	l_c	l_d	L_0
b^u	5000	0.5	0.5	0.8	0.8	0.1
b^l	0	0	0	0.0	0.0	0.0

Table 3.1: Constraints on mechanism parameters

By using the optimization technique described earlier, the optimum parameters that minimize the amount of unbalance are listed in table 3.2.

K	R_a	R_b	l_c	l_d	L_0
728	0.15	0.23	0.12	0.17	0.08

Table 3.2: Optimum Mechanism

As can be seen, the mechanism described by the parameters of table 3.2 is optimum only for the case where $m = 4$ kg and $l = 0.75$ m. Figure 3.5, shows the torque due to gravity as well as the torque generated by the SGM mechanism. For a given load, the amount of unbalance can be positive or negative depending wether the load is larger or smaller than $m = 4$ kg.

Optimization of the fixed end spring Mechanism

The optimization parameters of the fixed end spring mechanism are shown in table 3.3

	K	L_0	OA	OB
b^l	0	0.05	0.0	0.0
b^u	5000	0.7	0.5	0.5

Table 3.3: Constraints on mechanism parameters

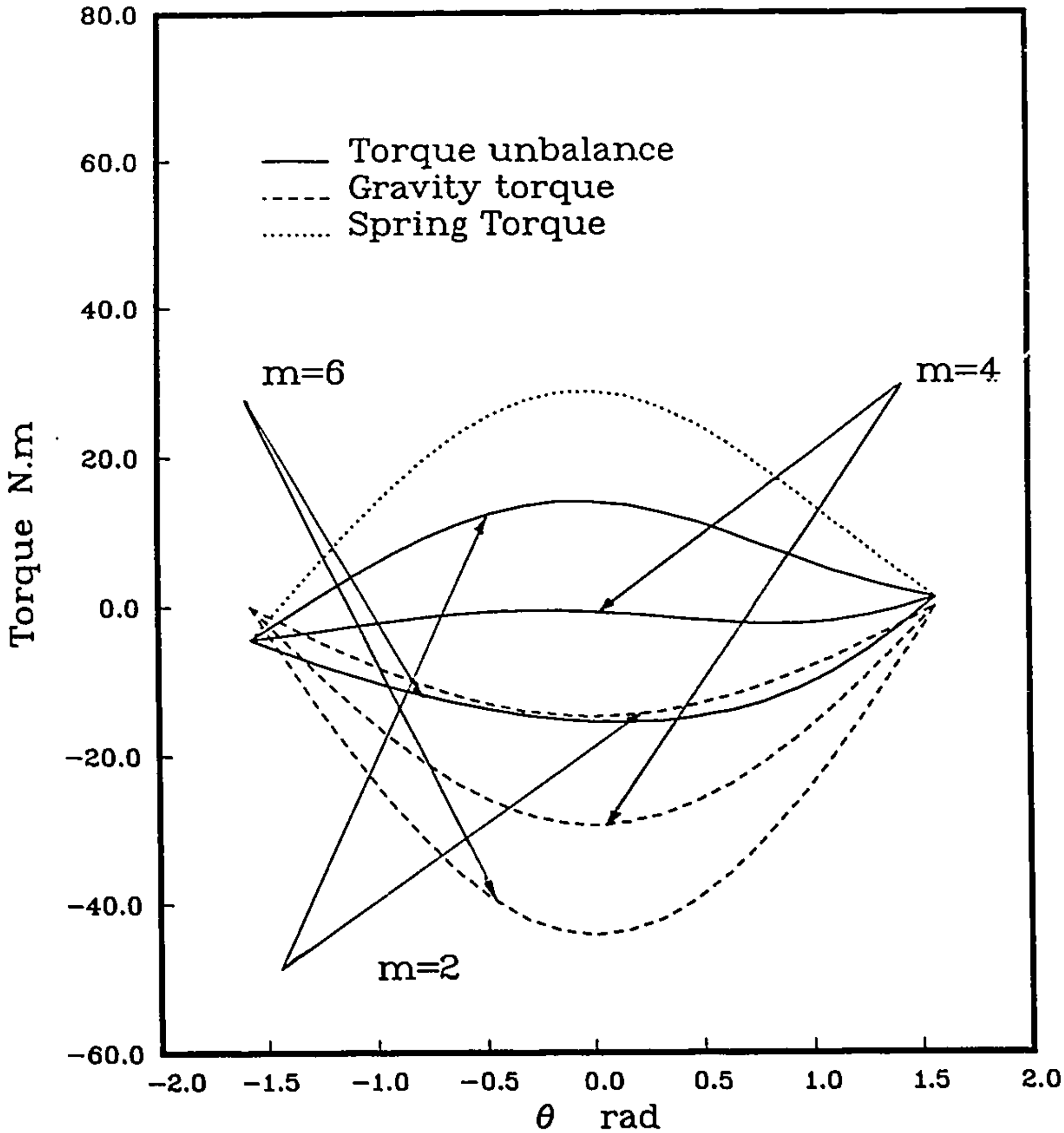

Figure 3.5: Balancing with SGM for $m = 2$, 4 and 6 kg

The optimum values to balance the one link manipulator with $m = 4$ kg and $l = 0.75$ m are given below.

K	L_0	OA	OB
620	0.05	0.1	0.2

Table 3.4: Optimum mechanism

3.1.4 Zero free length spring

For the two mechanisms described previously, complete balancing cannot be achieved due to the difference between the gravitational torque and the balancing torque at different angular positions θ of the manipulator arm. For the mechanism with fixed end, Equation 3.15 shows that if $L_0 = 0$, ie. a zero free length spring, and if OA and OB are chosen appropriately, the $E(\theta)$ can be made to vanish, and perfect balancing is achieved. Several method have been proposed in the literature to achieve a zero free length spring; Mathew and Tesar [2,3], Lu. Zhen [5] and P. Minotti [8]. Figure 3.7 shows one way to achieve this goal. An other method of obtaining a zero free length spring was presented by R. H. Nathan [4] and consists of putting a tension spring inside and coaxially with a compression spring with similar spring rate K. The "positive" free length of the tension spring and the "negative" free length of the compression spring will cancel each other giving an effectively zero free length capsule.

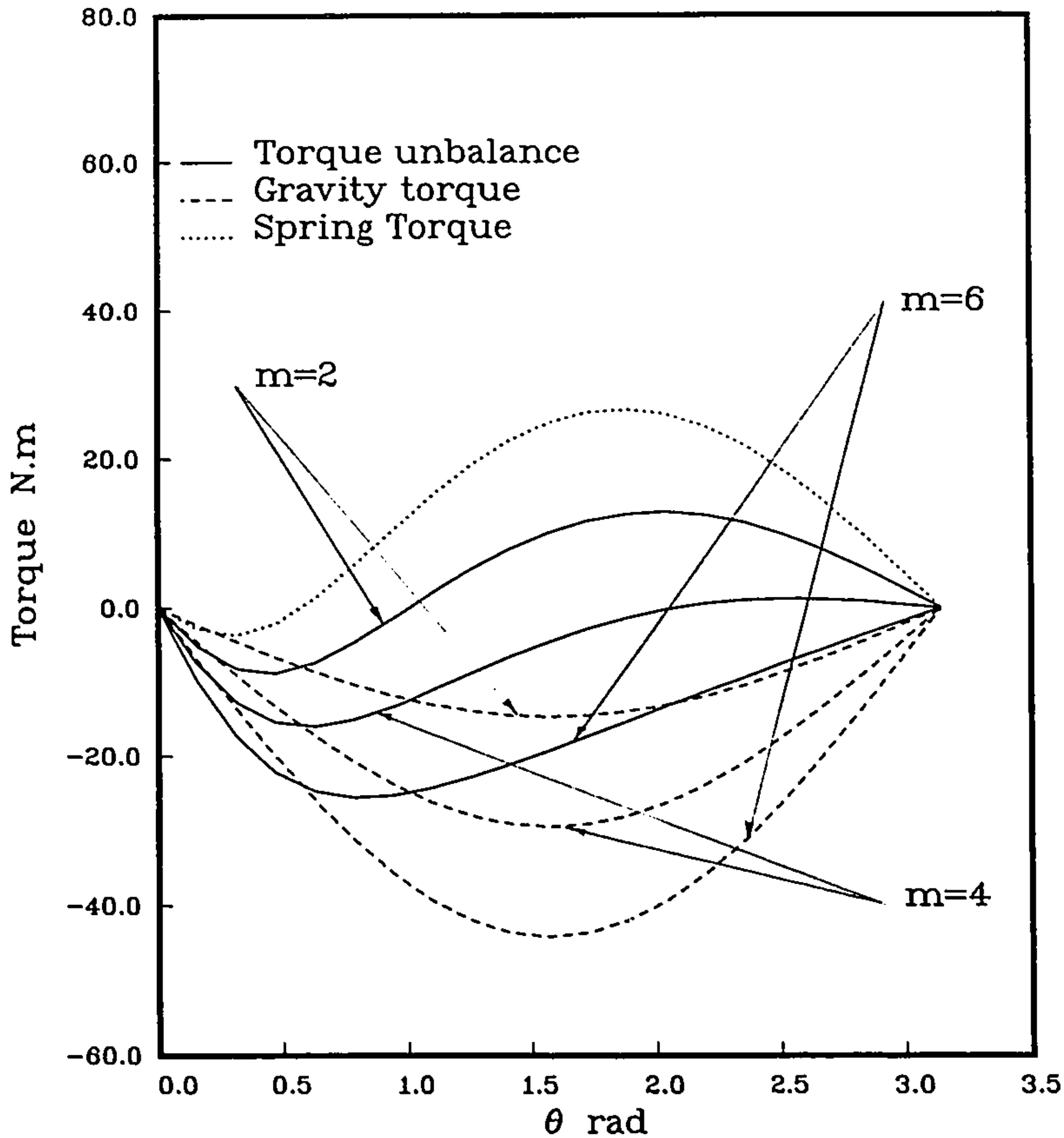

Figure 3.6: Balancing with spring with fixed end for $m = 2$, 4 and 6 kg

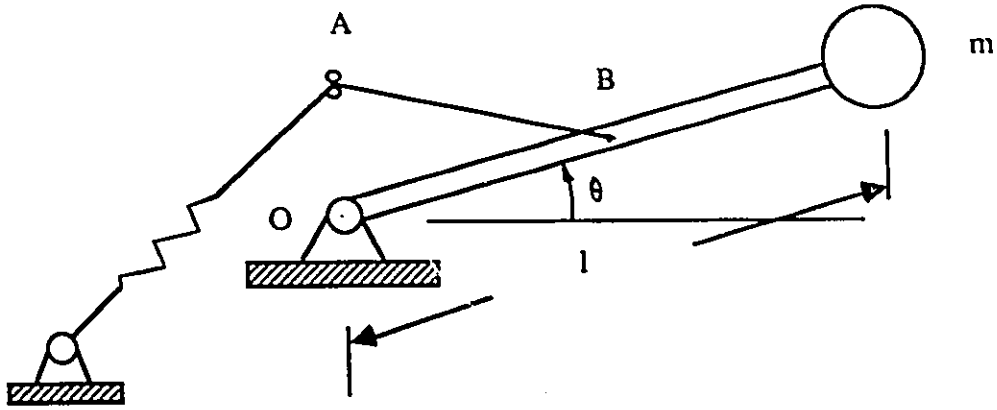

Figure 3.7: One link with a zero free length spring

For perfect balancing,

$$mgl = K \cdot OA \cdot OB \tag{3.19}$$

This equilibrium equation does not depend on θ. Thus with an appropriate selection of OA,OB and K, complete balancing of the gravity torque may be achieved.

Dynamic Analysis of The One Link Manipulator

The dynamic equation of the manipulator shown in Figure 3.7 can be obtained by using the Lagrange method. The Lagrangian is defined as:

$$L = T - P \tag{3.20}$$

Where, T is the total kinetic energy and P is the total potential energy. Assuming that the mass of the link is a point mass lumped at it's free end, T and P can be expressed as follow:

$$T = \frac{1}{2}mv^2 \tag{3.21}$$

$$P = mgl cos\theta + \frac{1}{2}K \cdot (AB)^2 \tag{3.22}$$

Where v is the linear velocity of the mass m, and AB is the spring deflection.

Substituting for T and P from 3.21 and 3.22 into Equation 3.20, the lagrangian L can be written as:

$$L = \frac{1}{2}ml^2\dot{\theta}^2 - mgl cos\theta - \frac{1}{2}K \cdot (AB)^2 \tag{3.23}$$

The dynamic equation is obtained from the lagrangian as follows:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}}) - \frac{\partial L}{\partial \theta} = Q \tag{3.24}$$

Solving for the components of the above equation and rearranging the terms gives:

$$ml^2\ddot{\theta} + (K \cdot OA \cdot OB - mgl)sin\theta = Q \tag{3.25}$$

Dividing by ml^2 both sides of Equation 3.25, the equation of motion may be cast in the standard form as follows:

$$\ddot{\theta} + {\omega_n}^2 sin\theta = \frac{Q}{ml^2} \tag{3.26}$$

ω_n is the natural frequency of the system and is expressed by

$$\omega_n = \sqrt{\frac{\delta}{ml^2}}$$

where

$$\delta = K \cdot OA \cdot OB - mgl$$

In the case when $w_n = 0$, the system becomes a pure inertial one. For $\omega_n = 0$, δ has to vanish, thus

$$K \cdot OA \cdot OB = mgl$$

This is the condition that arises when perfect balancing is achieved.

Effect of L_0 on balancing

Equation 3.15 of balancing for a mechanism having a fixed end shows if the mechanism have a zero free length then the system is in equilibrium at all positions. If the free length of the spring is not exactly zero, then a small amount of unbalance will remain. Figure 3.8 shows the effect of L_0 on the balancing of the one link manipulator as it moves from $-\pi$ to π. The parameters of the mechanism are selected so that the system is completely balanced when $L_0 = 0$. The range of L_0 is from $L_0 = 0$ to $L_0 = OB - OA$.

3.2 Balancing of a 3R Manipulator

In mechanisms with more then one link, the effect of gravity that one link experiences may not be a function of the physical properties of that link alone, the masses and lengths of the other links come into play.

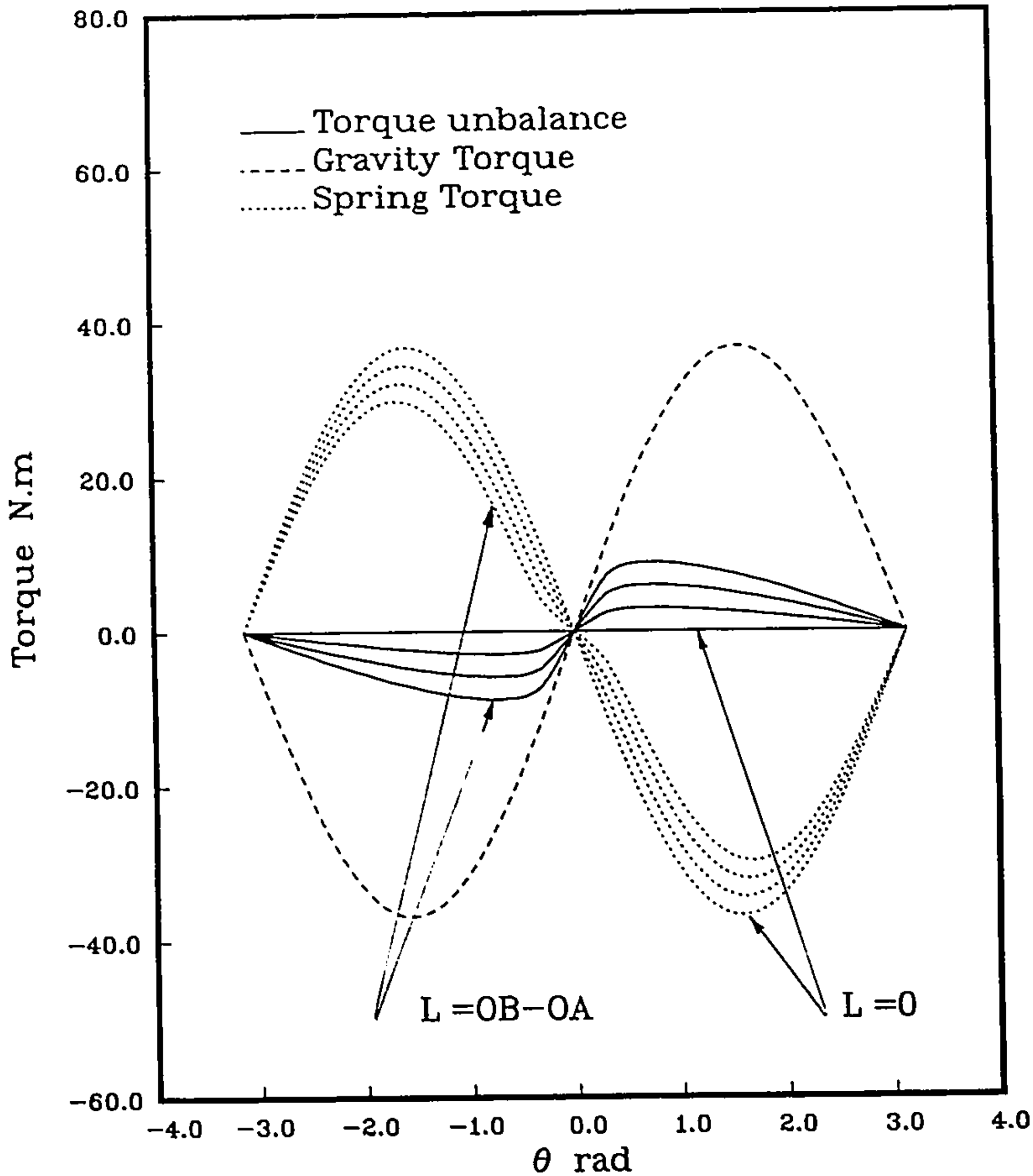

Figure 3.8: Effect of L_0 on a zero free length mechanism

When studying the effect of gravity in a 3R manipulator, the type of drive used should be taken into account when choosing the generalized coordinates. There are two main types of drive used, parallel and serial. The main difference between these two types of drives is that in the case of a parallel drive, the torque driving link 3 acts between the base and link 3. This means that the generalized coordinate for link 3 is absolute ie. taken with respect to ground. However, in the case of a serial drive, an actuator is located at each joint and the torque driving link 3 acts between link 2 and link 3, therefore, the actuator driving link 3 is a load for the actuator driving link 2. and the generalized coordinate system for joint 3 is relative to link 2. The gravity effect for a serial derive varies as a function of an absolute angle $\alpha_3 = \theta_2 + \theta_3$.

3.2.1 Balancing of link 3

Parallel Drive

In the case of a parallel drive, balancing of the upper arm can be achieved by fixing one end of the spring to the base and the other end to the lower side of the parallelogram transmitting the motion to the upper arm. This is illustrated in Figure 3.9. The Equilibrium equation for the balancing of the upper arm is given by:

$$Fl_3' sin\beta = M_{3_{eff}} l_{3_{eff}} cos\theta_3 \tag{3.27}$$

Where:

$$M_{3eff}l_{3eff} = M_l a_3 + m_3 l_3$$

With reference to Figure 3.9, taking the moments arround O_2, and assuming link 2 to be balanced, gives:

$$\begin{aligned} F_s sin\alpha &= F sin\lambda \\ &= F sin\beta \end{aligned} \tag{3.28}$$

Substituting Equations 3.28 into 3.27 gives:

$$F_s O_2 B_3 sin\alpha = M_{eff} l_{eff} cos\theta \tag{3.29}$$

since $l_3' = O_2B_3$. Equation 3.29 is similar to Equation 3.14 for the case of perfect balancing . The equilibrium condition can be derived by considering the triangle $O_2A_3B_3$.

$$M_{3eff}l_{3eff}g = K_3 \cdot O_2A_3 \cdot O_2B_3 \tag{3.30}$$

Serial Drive

In order to be able to incorporate the spring balancing arrangement described earlier with a serial drive, a four bar parallelogram mechanism is used as shown in Figure 3.10. The side O_3A_3 of the parallelogram remains always vertical for all configurations. One end of the spring is fixed to link 3 and the other end is fixed to O_3A_3. The balancing equation is similar to

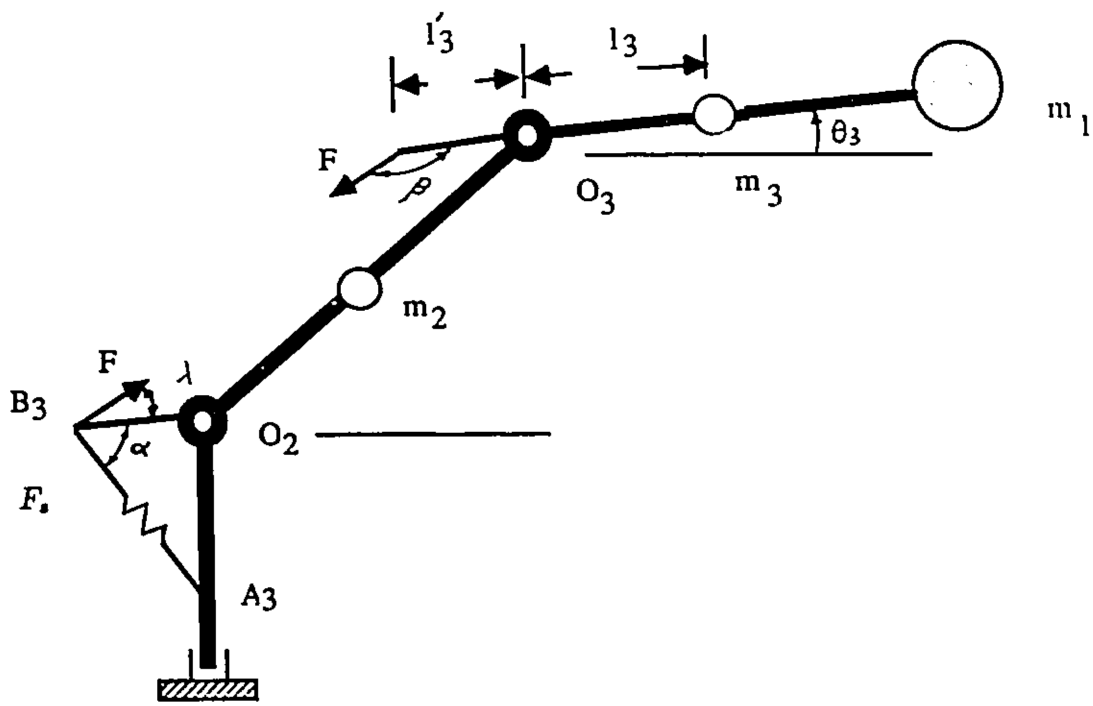

Figure 3.9: Balancing of link 3 in a parallel drive

a one link manipulator and is given by:

$$M_{3eff} l_{3eff} g = K_3 \cdot O_3A_3 \cdot O_3B_3 \tag{3.31}$$

3.2.2 Balancing of link 2

In order to derive the balancing equation of link 2, the principle of virtual work is to be used. With reference to Figure 3.11, link 2 and link 3 are displaced by small displacement $\delta\alpha_2$ and $\delta\alpha_3$, respectively. The work done by all the external forces acting on the link 2 and 3 is give by:

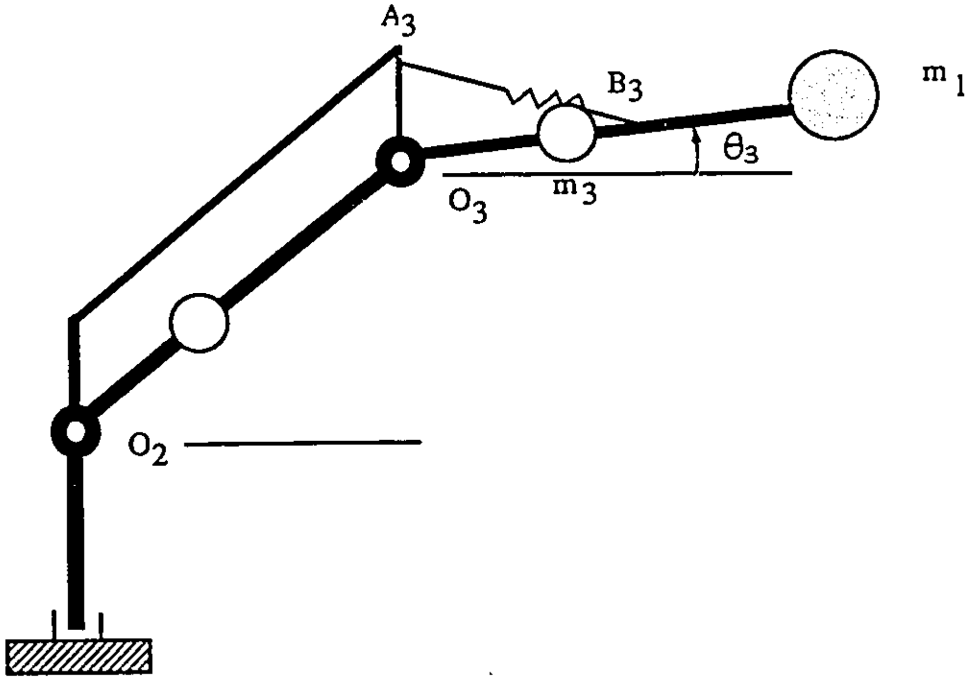

Figure 3.10: Balancing of link 3 in a serial drive

$$\begin{aligned} W = & (m_2 l_2 + m_3 a_2 + m_l a_2) g sin\alpha_2 \delta\alpha_2 \\ & + (m_3 l_3 + m_l a_3) g sin\alpha_3 \delta\alpha_3 \\ & - \tau_2 \delta\alpha_2 + \tau_3 \delta\alpha_3 \end{aligned} \tag{3.32}$$

If link 3 is already balanced, it gives:

$$(m_3 l_3 + m_l a_3) g sin\alpha_3 = \tau_3 \tag{3.33}$$

For balancing, Equation 3.32 has to vanish. Substituting for Equation 3.33 and eliminating $\delta\alpha_2$ from both sides of Equation 3.32 give:

$$(m_2 l_2 + (m_3 + m_l) a_2) g sin\alpha_2 = \tau_2 \tag{3.34}$$

This equation is equivalent to a one link manipulator with a mass M_{2eff} concentrated at a distance l_{2eff} from the joint. In a manner similar to the

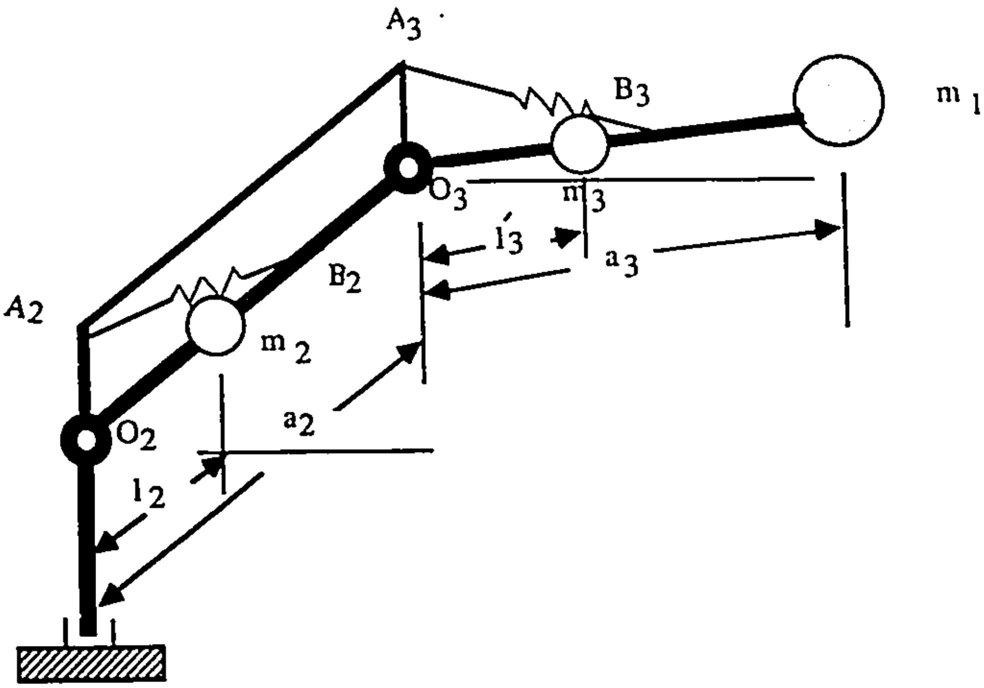

Figure 3.11: Balancing of a Serial drive

Where

$$M_{2eff} l_{2eff} = (m_2 l_2 + (m_3 + m_l) a_2)$$

3.2.3 Adjustment of the mechanism to a change in the payload

The mechanism can be easily adjusted to achieve perfect balancing when the payload changes by adjusting the distances O_2A_2 and O_2A_3 or O_2B_2 and O_2B_3. Rearranging the equilibrium equation for joint 2, (Equation 3.36), the relation between the payload and the distance O_2A_2 is given by:

$$O_2A_2 = \alpha_2 + \beta_2 m_l \tag{3.36}$$

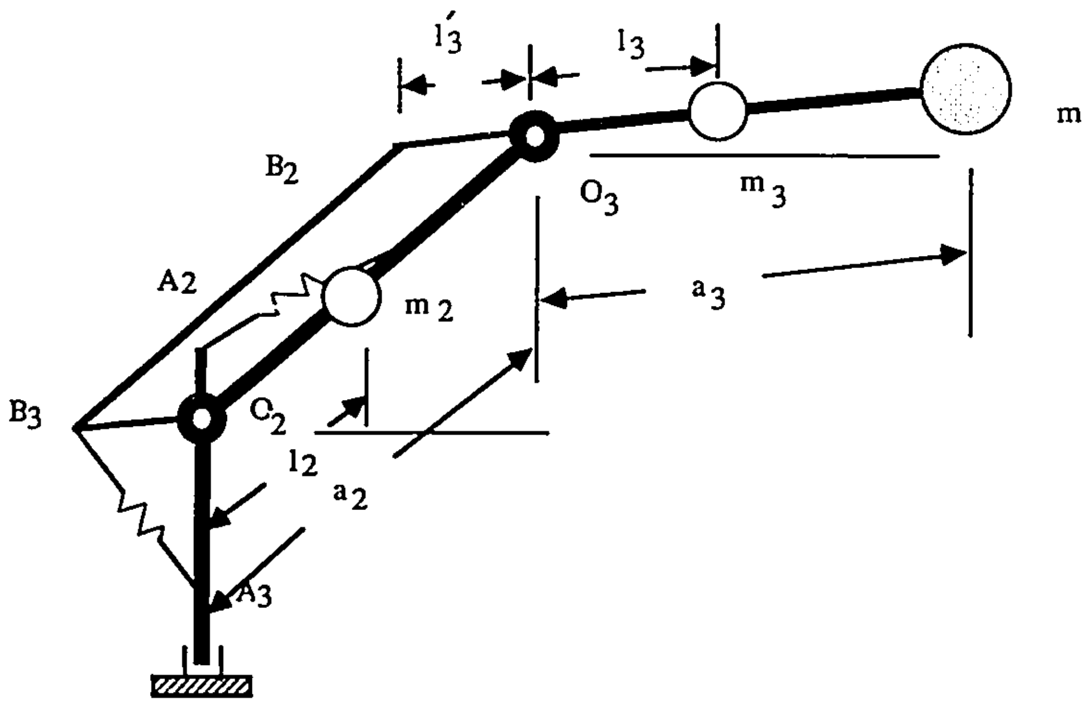

Figure 3.12: Balancing of a parallel drive

Where

$$\alpha_2 = \frac{(m_2 l_2 + m_3 a_2)g}{(K_2.O_2B_2)}$$

and

$$\beta_2 = \frac{l_2 g}{(K_2.O_2B_2)}$$

Similarly, the relation between the payload and O_2A_3 is given by:

$$O_2A_3 = \alpha_3 + \beta_3 m_l \tag{3.37}$$

Where

$$\alpha_3 = \frac{m_3 l_3 g}{(K_3.O_2B_3)}$$

and

$$\beta_3 = \frac{l_3' g}{(K_3.O_2B_3)}$$

Equation 3.38 and 3.39 show that the relation between the mechanisms parameter O_2A_2, O_2A_3 are linear. Furthermore, the intercepts as well as the slopes of the linear relations are themselves inversely proportional to K_2 and K_3 respectively.

O_2B_2 and O_2B_3 are usually limited by the lengths of the links, and their effect is very small compared to that of K_2 and K_3, respectively. If the mechanism is to be designed to balance a given range of payloads, the spring rates can be selected to give practical ranges of the adjustable distances O_2A_2 and O_2A_3.

Figure 3.13 and 3.14 show the effect of the spring rate on the rate of change of the distances O_2A_2 and O_2A_3 as the payload varies.

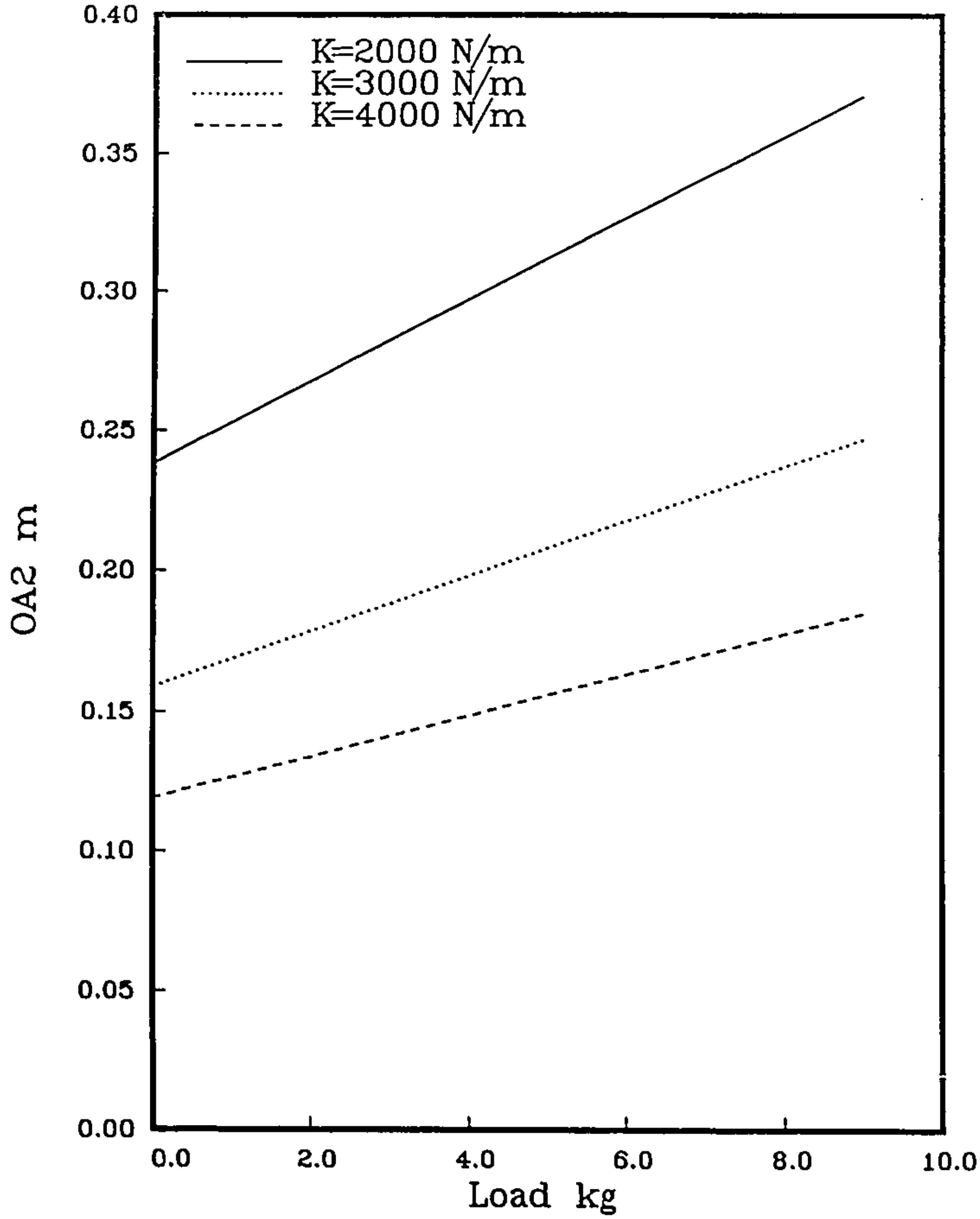

Figure 3.13: Effect of K_2 on the adjustment of the mechanism

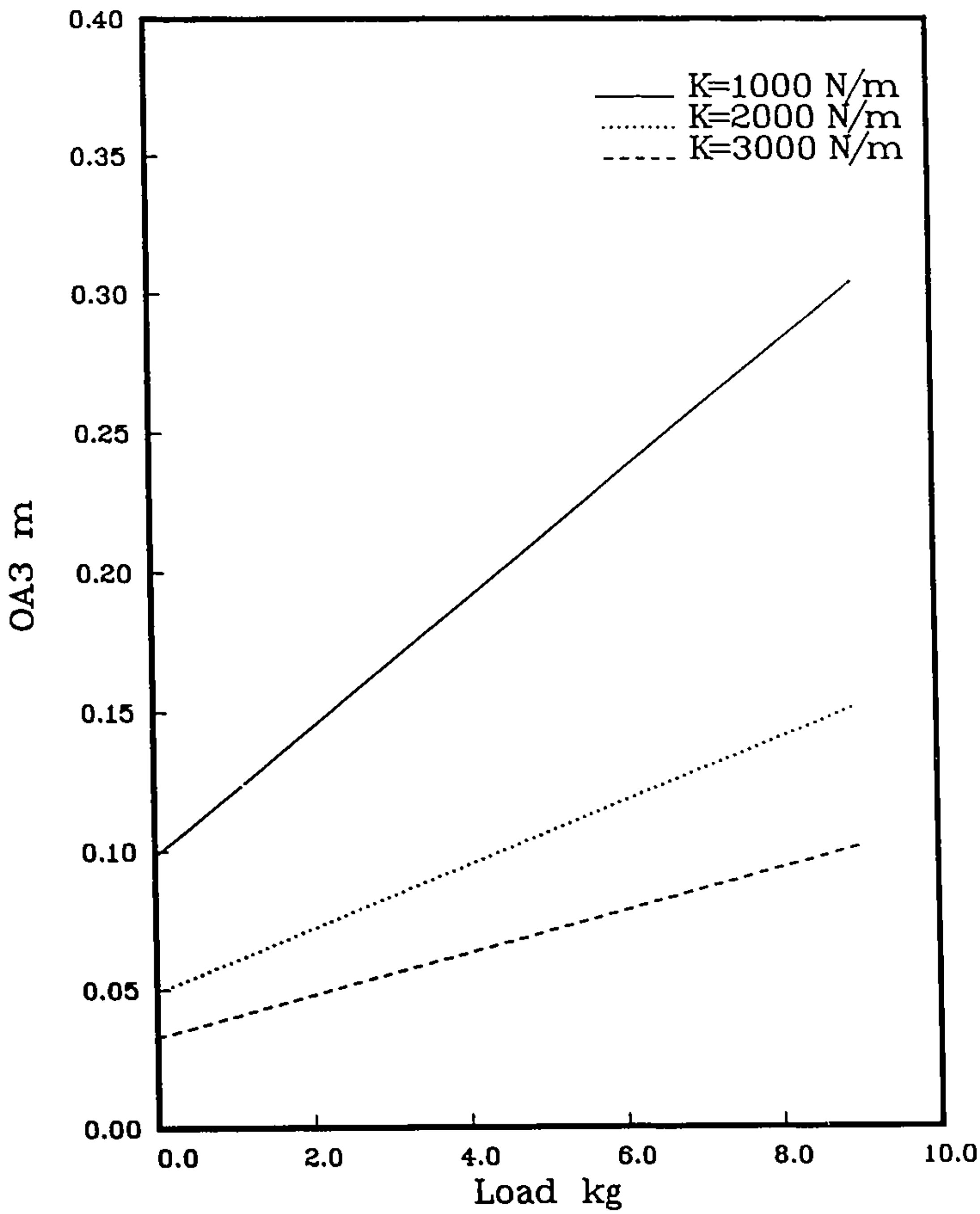

Figure 3.14: Effect of K_3 on the adjustment of the mechanism

Chapter 4

FORMULATION OF MANIPULATOR DYNAMICS

Several methods have been used in the derivation of the dynamics of robot manipulators, the most commonly used of these are the Newton-Euler method and the Lagrangian method.

In the Newton-Euler method, the equations of motion are derived from Newton's second law. These equations are not expressed in terms of the independent variables, and do not include the input joint torques explicitly.

In the Lagrangian method, also known as the work-energy method, the dynamic behavior of a system is described in terms of the work done by all the external forces acting on the system and by the total energy stored in it. Using this approach, all the internal forces are automatically eliminated.

In this chapter, the dynamic equations of a 3 DOF revolute robot will be derived using the Lagrangian method.

4.1 Lagrangian Dynamics

Let $q_1, ..., q_n$ be the generalized coordinates that completely locate a dynamic system with n DOF's

The total kinetic and potential energy stored in the system are given by:

$$\mathbf{K} = \sum_{i=1}^{m} \mathbf{k_i} \tag{4.1}$$

$$\mathbf{U} = \sum_{i}^{m} U_i \tag{4.2}$$

where m is the number of conponents in the system.

The Lagrangian **L** is defined as:

$$\mathbf{L} = \mathbf{K} - \mathbf{U} \tag{4.3}$$

and is a function of the generalized coordinates $q = [q_1, ..., q_n]^T$ and $\dot{q} = [\dot{q}_1, ..., \dot{q}_n]^T$. The equations of motion of the system as defined from the

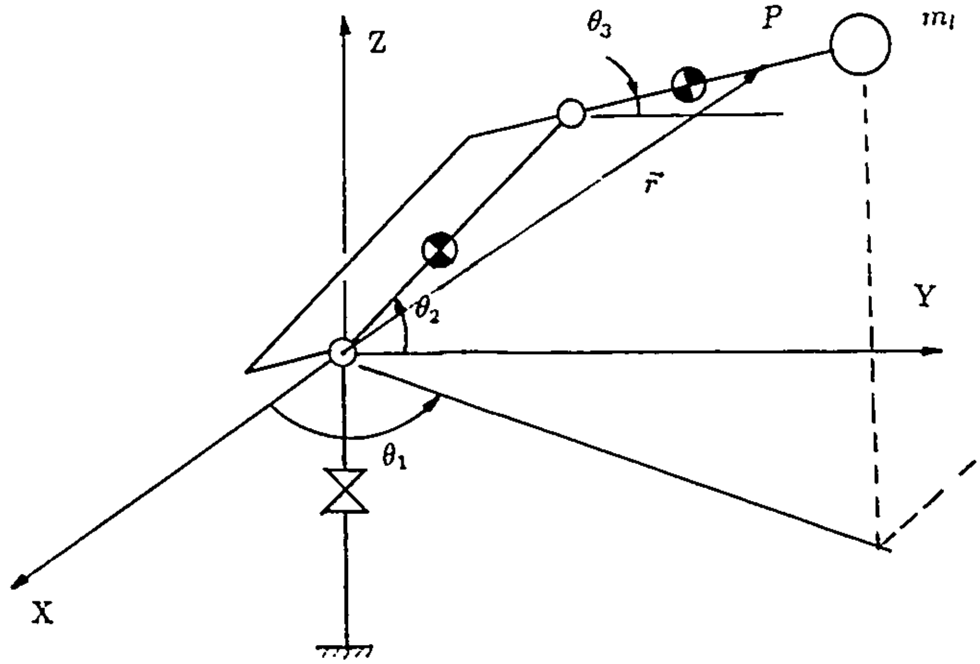

Figure 4.1: R-R-R parallel drive Manipulator

Lagrangian function are as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{4.4}$$

$$i = 1, ..., n$$

where Q_i is the sum of all the external forces acting on the system.

4.2 The Dynamic Equations of a R-R-R Manipulator

The robot used in the analysis is of the parallel drive type. Figure 4.1 shows a schematic representation of this type of robots. Each link of the

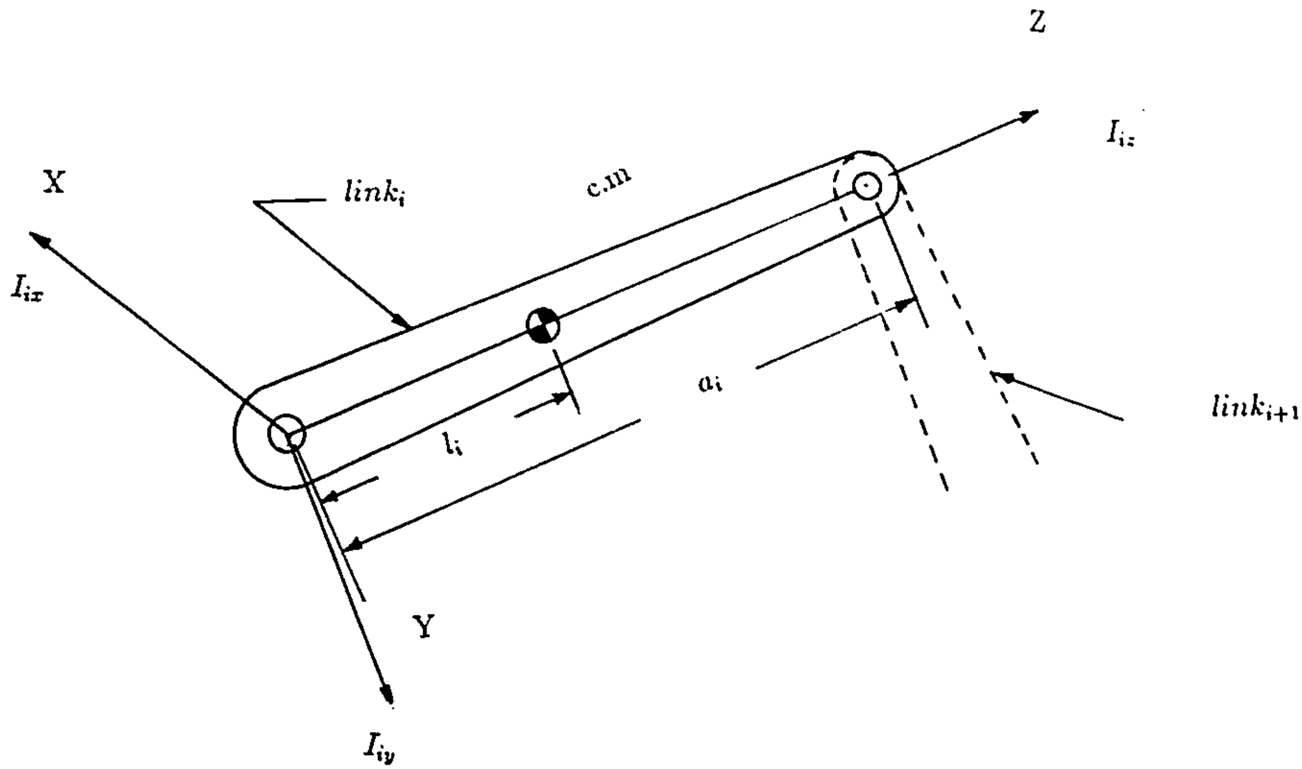

Figure 4.2: Physical constants of link i

manipulator as well as the payload is assumed to have an inertia given by the diagonal matrix

$$I_i = \begin{pmatrix} I_{ix} & 0 & 0 \\ 0 & I_{iy} & 0 \\ 0 & 0 & I_{iz} \end{pmatrix} \tag{4.5}$$

where all off diagonal elements are assumed to be negligible.

With reference to the figure, let P be a point on link i of the manipulator. P is related to the reference frame by a vector $\vec{r} = [r_x \; r_y \; r_z]^T$. As the manipualator moves, the velocity of point P is given by:

$$\vec{v} = \frac{d\vec{r}}{dt} = \mathbf{J}\dot{\theta} \tag{4.6}$$

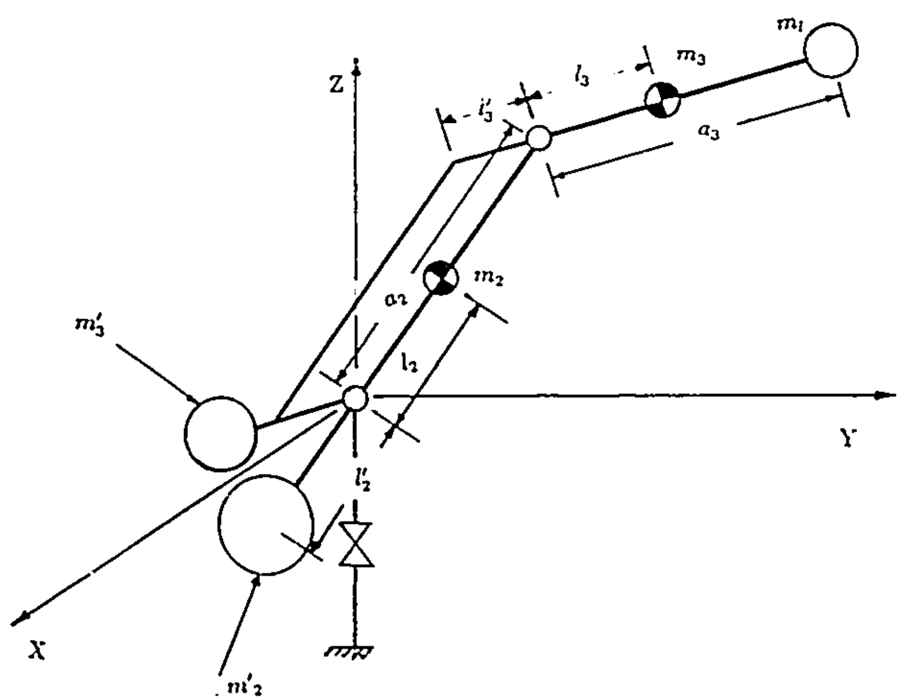

Figure 4.3: Weight counterbalanced R-R-R Manipulator

Where $\mathbf{J}$ is a 3×3 jacobian matrix and is given by:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial r_x}{\partial\theta_1} & \frac{\partial r_x}{\partial\theta_2} & \frac{\partial r_x}{\partial\theta_3} \\ \frac{\partial r_y}{\partial\theta_1} & \frac{\partial r_y}{\partial\theta_2} & \frac{\partial r_y}{\partial\theta_3} \\ \frac{\partial r_z}{\partial\theta_1} & \frac{\partial r_z}{\partial\theta_2} & \frac{\partial r_z}{\partial\theta_3} \end{pmatrix} \tag{4.7}$$

and $\dot{\theta}=[\dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3\ ]^T$ is the generalized velocity vector.

4.2.1 Dynamic equations of weight counterbalanced R-R-R manipulator

Figure 4.3 shows an R-R-R manipulator with weight counterbalanced links. The counterweights m'_2 and m'_3 are to balance link 2 and 3, respectively. The calculation of the kinetic energy of the system requires the evaluation of the absolute angular velocity of each link as well as the velocity of the links center of mass.

The center of mass of link 1 is located on the axis of rotation, hence $\bar{V}_1$=0, and the angular velocity of the link is:

$$\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \tag{4.8}$$

The linear velocities of the lumped masses m_2, m'_2,m_3, m'_3 and m_l are calculated from Equation 4.6 and they are given by:

$$\begin{aligned} \bar{V}_2 &= (-l_2 s_1 c_2 \dot{\theta}_1 - l_2 c_1 s_2 \dot{\theta}_2)\mathbf{i} \\ &\quad +(l_2 c_1 c_2 \dot{\theta}_1 - l_2 s_1 s_2 \dot{\theta}_2)\mathbf{j} \\ &\quad +l_2 c_2 \dot{\theta}_2 \mathbf{k} \end{aligned} \tag{4.9}$$

$$\begin{aligned} \bar{V'}_2 &= (l'_2 s_1 c_2 \dot{\theta}_1 + l'_2 c_1 s_2 \dot{\theta}_2)\mathbf{i} \\ &\quad +(-l'_2 c_1 c_2 \dot{\theta}_1 + l'_2 s_1 s_2 \dot{\theta}_2)\mathbf{j} \\ &\quad -l'_2 c_2 \dot{\theta}_2 \mathbf{k} \end{aligned} \tag{4.10}$$

$$\begin{aligned} \bar{V}_3 &= (-s_1(a_2 c_2 + l_3 c_3)\dot{\theta}_1 - a_2 c_1 s_2 \dot{\theta}_2 - l_3 c_1 s_3 \dot{\theta}_3)\mathbf{i} \\ &\quad +(c_1(a_2 c_2 + l_3 c_3)\dot{\theta}_1 - a_2 s_1 s_2 \dot{\theta}_2 - l_3 s_1 s_3 \dot{\theta}_3)\mathbf{j} \\ &\quad +(a_2 c_2 \dot{\theta}_2 + l_3 c_3 \dot{\theta}_3)\mathbf{k} \end{aligned} \tag{4.11}$$

$$\begin{aligned} \bar{V'}_3 &= (l'_3 s_1 c_3 \dot{\theta}_1 + l'_3 c_1 s_3 \dot{\theta}_3)\mathbf{i} \\ &\quad +(l'_3 c_1 c_3 \dot{\theta}_1 + l'_3 s_1 s_3 \dot{\theta}_3)\mathbf{j} \end{aligned} \tag{4.12}$$

$$
\begin{aligned}
& -l'_3 c_3 \dot{\theta}_3 \mathbf{k} \\
\bar{V}_l \; = \; & (-s_1(a_2c_2 + a_3c_3)\dot{\theta}_1 - a_2c_1s_2\dot{\theta}_2 - a_3c_1s_3\dot{\theta}_3)\mathbf{i} \qquad (4.13) \\
& +(c_1(a_2c_2 + a_3c_3)\dot{\theta}_1 - a_2s_1s_2\dot{\theta}_2 - a_3s_1s_3\dot{\theta}_3\mathbf{j}) \\
& +(a_2c_2\dot{\theta}_2 + l_3c_3\dot{\theta}_3)\mathbf{k}
\end{aligned}
$$

The angular velocities of links 2 and 3, and the load are given respectively by:

$$
\omega_2 = \begin{bmatrix} \dot{\theta}_1 c_2 \\ \dot{\theta}_2 \\ \dot{\theta}_1 s_2 \end{bmatrix} \qquad (4.14)
$$

$$
\omega_3 = \omega_l = \begin{bmatrix} \dot{\theta}_1 c_3 \\ \dot{\theta}_3 \\ \dot{\theta}_1 s_3 \end{bmatrix} \qquad (4.15)
$$

From Equations 4.1, 4.8 to 4.15, the total kinetic energy of the system is given by:

$$
\mathbf{K} = \frac{1}{2}\sum_{i=1}^{m} m_i \bar{V}_i^T \bar{V}_i + \frac{1}{2}\sum_{i=1}^{m} \omega_i^T I_i \omega_i \qquad (4.16)
$$

where m is the number of masses in the system and includes 3 link masses, 2 counterweights and the load.

The total potential energy of the system is given by:

$$
\begin{aligned}
\mathbf{U} \; = \; & (m_2l_2 + (m_3 + m_l)a_2)gs_2 + (m_3l_3 + m_la_3)gs_3 \qquad (4.17) \\
& -m'_2l'_2gs_2 - m'_3l'_3gs_3
\end{aligned}
$$

Substituting Equations 4.16 and 4.17 into 4.4, the equations of motion associated with the generalized coordinates are given by:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}_1}\right) - \frac{\partial K}{\partial \theta_1} + \frac{\partial U}{\partial \theta_1} = \tau_1$$

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}_2}\right) - \frac{\partial K}{\partial \theta_2} + \frac{\partial U}{\partial \theta_2} = \tau_2$$

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}_3}\right) - \frac{\partial K}{\partial \theta_3} + \frac{\partial U}{\partial \theta_3} = \tau_3$$

where τ_1,τ_2 and τ_3 are the driving torques developed by the actuators of joint 1,2 and 3 respectively.

The resulting equation of motion can be presented in matrix form as follow:

$$\mathbf{H}(\theta)\ddot{\theta} + \mathbf{h}(\theta, \dot{\theta}) + \mathbf{G}(\theta) = \tau \tag{4.18}$$

Where $\mathbf{H}$ is a 3×3 matrix, known as the manipulator inertia matrix. The elements of this matrix are:

$$\begin{aligned} H_{11} \;=\;& I_{1z} + I_{2x}{c_2}^2 + I_{2z}{s_2}^2 + m_2{l_2}^2{c_2}^2 \\ & + (I_{3x} + I_{lx}){c_3}^2 + (I_{3z} + I_{lz}){s_3}^2 \\ & + m_3(a_2c_2 + l_3c_3)^2 + m_l(a_2c_2 + a_3c_3)^2 \\ & + I'_{2x}{c_2}^2 + I'_{2z}{s_2}^2 + m'_2 l'^2_2 {c_2}^2 \\ & + I'_{3x}{c_3}^2 + I'^2_{3z} \end{aligned} \tag{4.19}$$

$$+m'_3 l'_3 c_3^2$$

$$H_{12} = 0 \tag{4.20}$$

$$H_{13} = 0 \tag{4.21}$$

$$H_{21} = 0 \tag{4.22}$$

$$H_{31} = 0 \tag{4.23}$$

$$H_{22} = I_{2y} + m_2 l_2^{\,2} + (m_3 + m_l) a_2^{\,2} + I'_{2y} + m'_2 l'^2_2 \tag{4.24}$$

$$H_{32} = H_{23} = (m_3 l_3^{\,2} + m_l a_3^{\,2}) c_{3-2} \tag{4.25}$$

$$H_{33} = m_3 l_3^{\,2} + m_l a_3^{\,2} + I_{3y} + I_{ly} + m'_3 l'^2_3 + I'_{3y} \tag{4.26}$$

$h(\theta,\dot{\theta})$ is a 1×3 vector representing the centrifugal and coriolis forces. The elements of this vector are:

$$\begin{aligned} h_1 = {} & 2\{(I_{2z} - I_{2x}) c_2 s_2 - m_2 l_2^2 c_2 s_2\}\dot{\theta}_1\dot{\theta}_2 \\ & -2\{m_3(a_2c_2 + l_3c_3)(a_2s_2\dot{\theta}_2 + l_3s_3\dot{\theta}_3) \\ & +m_l(a_2c_2 + a_3c_3)(a_2s_2\dot{\theta}_2 + a_3s_3\dot{\theta}_3)\}\dot{\theta}_1 \\ & +2(I_{3z} + I_{lz} - I_{3x} - I_{lx}) s_3 c_3 \dot{\theta}_1\dot{\theta}_3 \\ & +2\{(I'_{2z} - I'_{2x}) c_2 s_2 - m'_2 l'^2_2 c_2 s_2\}\dot{\theta}_1\dot{\theta}_2 \\ & +2(I'_{3z} - I'_{3x} - m'_3 l'^2_3) c_3 s_3 \dot{\theta}_1)\dot{\theta}_3 \\ & \dot{\theta}_1\dot{\theta}_3 \end{aligned} \tag{4.27}$$

$$\begin{aligned} h_2 = {} & \{(I_{2x} - I_{2z}) s_2 c_2 + m_2 l_2^2 s_2 c_2 + m_3(a_2c_2 + l_3c_3) a_2 s_2 \\ & +m_l(a_2c_2 + a_3c_3) a_2 s_2\}\dot{\theta}_1^2 \\ & -(m_3 l_3 + m_l a_3) a_2 s_{3-2} \dot{\theta}_2 \dot{\theta}_3 \\ & -(m_3 l_3^2 + m_l a_3^2) s_{3-2} (\dot{\theta}_3 - \dot{\theta}_2)\dot{\theta}_3 \\ & +\{(I'_{2x} - I'_{2z}) s_2 c_2 + m'_2 l'^2_2 s_2 c_2\}\dot{\theta}_1^2 \end{aligned} \tag{4.28}$$

$$
\begin{aligned}
& -m'_3 l'^2_3 s_{3-2}(\dot{\theta}_3 - \dot{\theta}_2)\dot{\theta}_3 \\
h_3 \quad = \quad & \{(I_{3x} + I_{lx} - I_{3z} - I_{lz})c_3 s_3 + m_3(a_2 c_2 + l_3 c_3)s_3 l_3 \qquad (4.29) \\
& + m_l(a_2 c_2 + a_3 c_3)s_3 a_3\}\dot{\theta}_1^2 + (m_3 l_3 + m_l a_3)a_2 s_{3-2}\dot{\theta}_2^2 \\
& + \{(I'_{3x} - I'_{3z})c_3 s_3 + m'_3 l'^2_2 c_3 s_3\}\dot{\theta}_1^2
\end{aligned}
$$

$$(4.30)$$

and $G(\theta)$ is a 1×3 vector representing the effect of gravity on each of the three links. This vector can be obtained by evaluating the term $\frac{\partial U}{\partial \theta_i}$ as follows:

$$
\begin{aligned}
g_1 &= \frac{\partial U}{\partial \theta_1} = 0 && (4.31) \\
g_2 &= \frac{\partial U}{\partial \theta_2} = (m_2 l_2 + (m_3 + m_l)a_2 - m'_2 l'_2)g c_2 && (4.32) \\
g_3 &= \frac{\partial U}{\partial \theta_3} = (m_3 l_3 + m_l a_3 - m'_3 l'_3)g s_3 && (4.33)
\end{aligned}
$$

If the manipulator links are to be balanced at a load m_{lb}, then the counterweights m'_2 and m'_3 have to satisfy the following relations:

$$m_2 l_2 + (m_3 + m_{lb})a_2 = m'_2 l'_2$$

$$m_3 l_3 + m_{lb} a_3 = m'_3 l'_3$$

4.2.2 Dynamic equations of a spring balanced R-R-R manipulator

The counterweights used for the balancing of link 2 and link 3 are replaced by two springs as shown in Figure 4.4. The springs are assumed

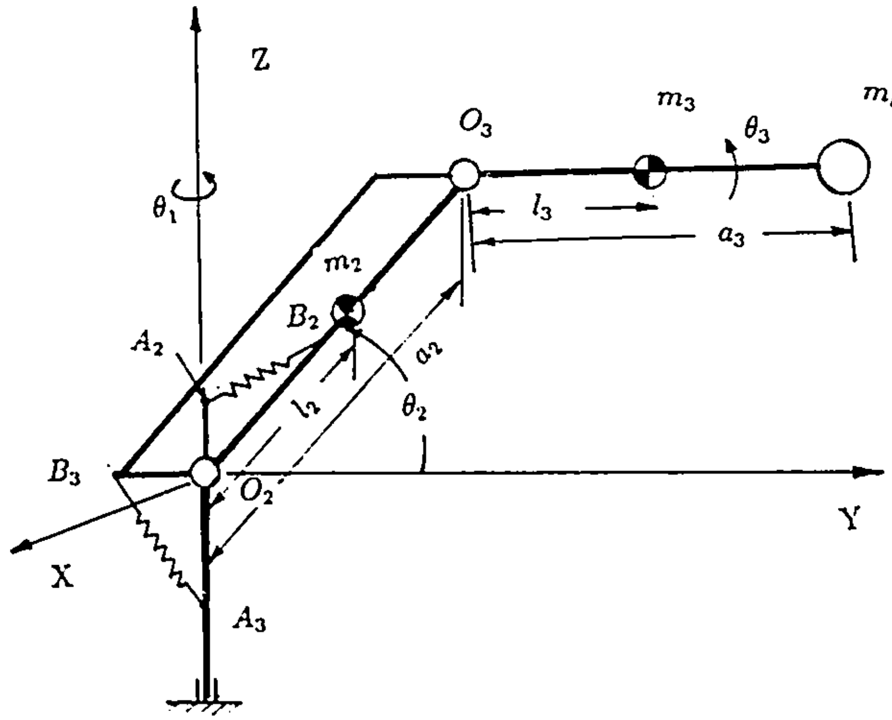

Figure 4.4: Spring counterbalanced R-R-R Manipulator

to have negligible weights in comparaison to the weight of the links. The kinetic energy of this system is similar to that of the weight counterbalanced system with the exeption that all terms countaining m_2', m_3', I_2' and I_3' vanish.

The potential energy of this system is given by:

$$\begin{aligned} U = \; & m_2 g l_2 s_2 + m_3 g(l_3 s_3 + a_2 s_2) + m_l g(a_3 s_3 + a_2 s_2) \\ & + \tfrac{1}{2} K_2 x_2^2 + \tfrac{1}{2} K_3 x_3^2 \end{aligned} \tag{4.34}$$

where x_2 and x_3 are the elongations of spring 2 and spring 3 respectively. For a spring with zero free length, x_2 and x_3 are given by:

$$x_2 = \sqrt{(O_2A_2)^2 + (O_2B_2)^2 - 2 \cdot (O_2A_2) \cdot (O_2B_2) s_2} \tag{4.35}$$

$$x_3 = \sqrt{(O_2A_3)^2 + (O_2B_3)^2 - 2 \cdot (O_2A_3) \cdot (O_2B_3) s_3} \tag{4.36}$$

Substituting the above two equations into Equation 4.34 gives:

$$
\begin{aligned}
U &= m_2 g l_2 s_2 + m_3 g(l_3 s_3 + a_2 s_2) + m_l g(a_3 s_3 + a_2 s_2) \\
&+ \frac{1}{2} K_2 \cdot (O_2A_2)^2 + (O_2B_2)^2 - 2 \cdot (O_2A_2) \cdot (O_2B_2) s_2) \\
&+ \frac{1}{2} K_3 \cdot (O_2A_3)^2 + (O_2B_3)^2 - 2 \cdot (O_2A_3) \cdot (O_2B_3) s_3)
\end{aligned} \tag{4.37}
$$

The equations of motion of this manipulator are given in matrix form by Equation 4.18 with the elements of the manipulator inertia matrix $H(\theta)$ and the vector $h(\theta)$ given by:

$$
\begin{aligned}
H_{11} &= I_{1z} + I_{2x} {c_2}^2 + I_{2z} {s_2}^2 + m_2 {l_2}^2 {c_2}^2 \\
&+ (I_{3x} + I_{lx}) {c_3}^2 + (I_{3z} + I_{lz}) {s_3}^2 \\
&+ m_3 (a_2 c_2 + l_3 c_3)^2 + m_l (a_2 c_2 + a_3 c_3)^2
\end{aligned} \tag{4.38}
$$

$$H_{12} = 0 \tag{4.39}$$

$$H_{13} = 0 \tag{4.40}$$

$$H_{21} = 0 \tag{4.41}$$

$$H_{31} = 0 \tag{4.42}$$

$$H_{22} = I_{2y} + m_2 {l_2}^2 + (m_3 + m_l) {a_2}^2 \tag{4.43}$$

$$H_{32} = H_{23} = (m_3 {l_3}^2 + m_l {a_3}^2) c_{3-2} \tag{4.44}$$

$$H_{33} = m_3 {l_3}^2 + m l {a_3}^2 + I_{3y} + I_{ly} \tag{4.45}$$

$$
\begin{aligned}
h_1 &= 2\{(I_{2z} - I_{2x}) c_2 s_2 - m_2 l_2^2 c_2 s_2\} \dot{\theta}_1 \dot{\theta}_2 \\
&- 2\{m_3 (a_2 c_2 + l_3 c_3)(a_2 s_2 \dot{\theta}_2 + l_3 s_3 \dot{\theta}_3) \\
&+ m_l (a_2 c_2 + a_3 c_3)(a_2 s_2 \dot{\theta}_2 + a_3 s_3 \dot{\theta}_3)\} \dot{\theta}_1 \\
&+ 2(I_{3z} + I_{lz} - I_{3x} - I_{lx}) s_3 c_3 \dot{\theta}_1 \dot{\theta}_3
\end{aligned} \tag{4.46}
$$

$$
\begin{aligned}
h_2 &= \{(I_{2x} - I_{2z})s_2c_2 + m_2l_2^2 s_2c_2 + m_3(a_2c_2 + l_3c_3)a_2s_2 \qquad (4.47)\\
&\quad + m_l(a_2c_2 + a_3c_3)a_2s_2\}\dot{\theta}_1^2\\
&\quad -(m_3l_3 + m_la_3)a_2s_{3-2}\dot{\theta}_2\dot{\theta}_3\\
&\quad -(m_3l_3^2 + m_la_3^2)s_{3-2}(\dot{\theta}_3 - \dot{\theta}_2)\dot{\theta}_3\\
h_3 &= \{(I_{3x} + I_{lx} - I_{3z} - I_{lz})c_3s_3 + m_3(a_2c_2 + l_3c_3)s_3l_3 \qquad (4.48)\\
&\quad + m_l(a_2c_2 + a_3c_3)s_3a_3\}\dot{\theta}_1^2\\
&\quad +(m_3l_3 + m_la_3)a_2s_{3-2}\dot{\theta}_2^2
\end{aligned}
$$

(4.49)

The gravity terms in the equation of motion for this case are given by:

$$
\begin{aligned}
g_1 &= \frac{\partial U}{\partial \theta_1} = 0 && (4.50)\\
g_2 &= \frac{\partial U}{\partial \theta_2} = ((m_2l_2 + (m_3 + m_l)a_2)g - K_2(O_2A_2 \cdot O_2B_2)c_2 && (4.51)\\
g_3 &= \frac{\partial U}{\partial \theta_3} = ((m_3l_3 + m_la_3)g - K_3(O_2A_3 \cdot O_2B_3)c_3 && (4.52)
\end{aligned}
$$

For the links to be completely balanced, two equilibrium conditions have to be satisfied. These two conditions are given by:

$$m_2gl_2 + (m_3 + m_l)ga_2 = K_2 \cdot O_2A_2 \cdot O_2B_2$$

$$m_3gl_3 + m_lga_3 = K_3 \cdot O_2A_3 \cdot O_2B_3$$

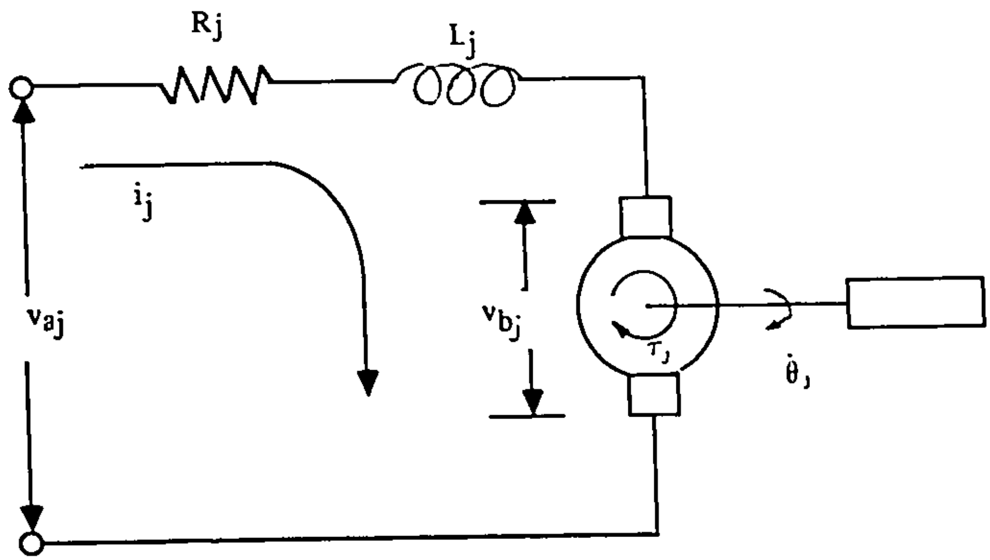

Figure 4.5: Equivalent circuit of a DC motor

4.3 Modelling of the Actuators

The actuators driving the robot links are generally armature controlled dc motors. Figure 4.5 show a schematic diagram of the robot link drive. With reference to the figure, the circuit equations are given by:

$$L_j \frac{di_j}{dt} + R_j i_j + v_{bj} = v_{aj} \tag{4.53}$$

$$v_{bj} = K_{bj}\omega_j \tag{4.54}$$

Where: L_j and R_j are the inductance and resistance of the armature, respectively.

i_j is the current in the circuit

v_{aj} is the armature voltage and is usually supplied by an amplifier.

v_{bj} is an induced voltage due to the rotation of the armature and K_{bj} is the back emf constant.

The torque developed by the motor is a function of the current and it is given by:

$$\tau_j = K_j i_j \tag{4.55}$$

In the case of a parallel drive, the angular velocity of motor j is $\omega_j = \dot{\theta}_j$, where θ_j is the generalized coordinate associated with link j.
Solving Equations 4.53 and 4.54 gives:

$$\frac{L_j}{R_j}\frac{di_j}{dt} + i_j = -\frac{K_{bj}}{R_j}\dot{\theta}_j + \frac{1}{R_j}v_{aj} \tag{4.56}$$

Neglecting the inductance L_j, Equation 4.56 reduces to:

$$i_j = -\frac{K_{bj}}{R_j}\dot{\theta}_j + \frac{1}{R_j}v_{aj} \tag{4.57}$$

The output torque from the actuator j can expressed in terms of its parameters as:

$$\tau_j = K_j i_j = -\frac{K_j K_{bj}}{R_j}\dot{\theta}_j + \frac{K_j}{R_j}v_{aj} \tag{4.58}$$

For the three actuators involved in the system, it is more convenient to express Equation 4.58 in matrix form as follows:

$$\tau = \mathbf{B}\dot{\theta} + \mathbf{D}\mathbf{u} \tag{4.59}$$

Where:

$$\tau = [\tau_1 \ \ \tau_2 \ \ \tau_3]^T \tag{4.60}$$

$$\mathbf{B} = diag[-\frac{K_{b1}K_1}{R_1} \ \ -\frac{K_{b2}K_2}{R_2} \ \ -\frac{K_{b3}K_3}{R_3}] \tag{4.61}$$

$$\mathbf{D} = diag[\frac{K_1}{R_1} \ \ \frac{K_2}{R_2} \ \ \frac{K_3}{R_3}] \tag{4.62}$$

Chapter 5

MANIPULATOR CONTROL

5.1 Introduction

In this chapter, two control algorithms have been used to compare the response of a R-R-R manipulator when it is balanced by counterweights and when it is balanced by springs. The first approach uses an individual joint controller scheme. The control circuit of each link, in this case, does not take into account the interaction between the motion of the link being controlled and that of the other links. The second approach, uses the computed torque technique for the tracking of a pre-programmed trajectory. All the literature dealing with this type of approach emphasize the difficulty of controlling the robot on line due to the highly nonlinear nature and the coupling between the set of differential equations describing the robot

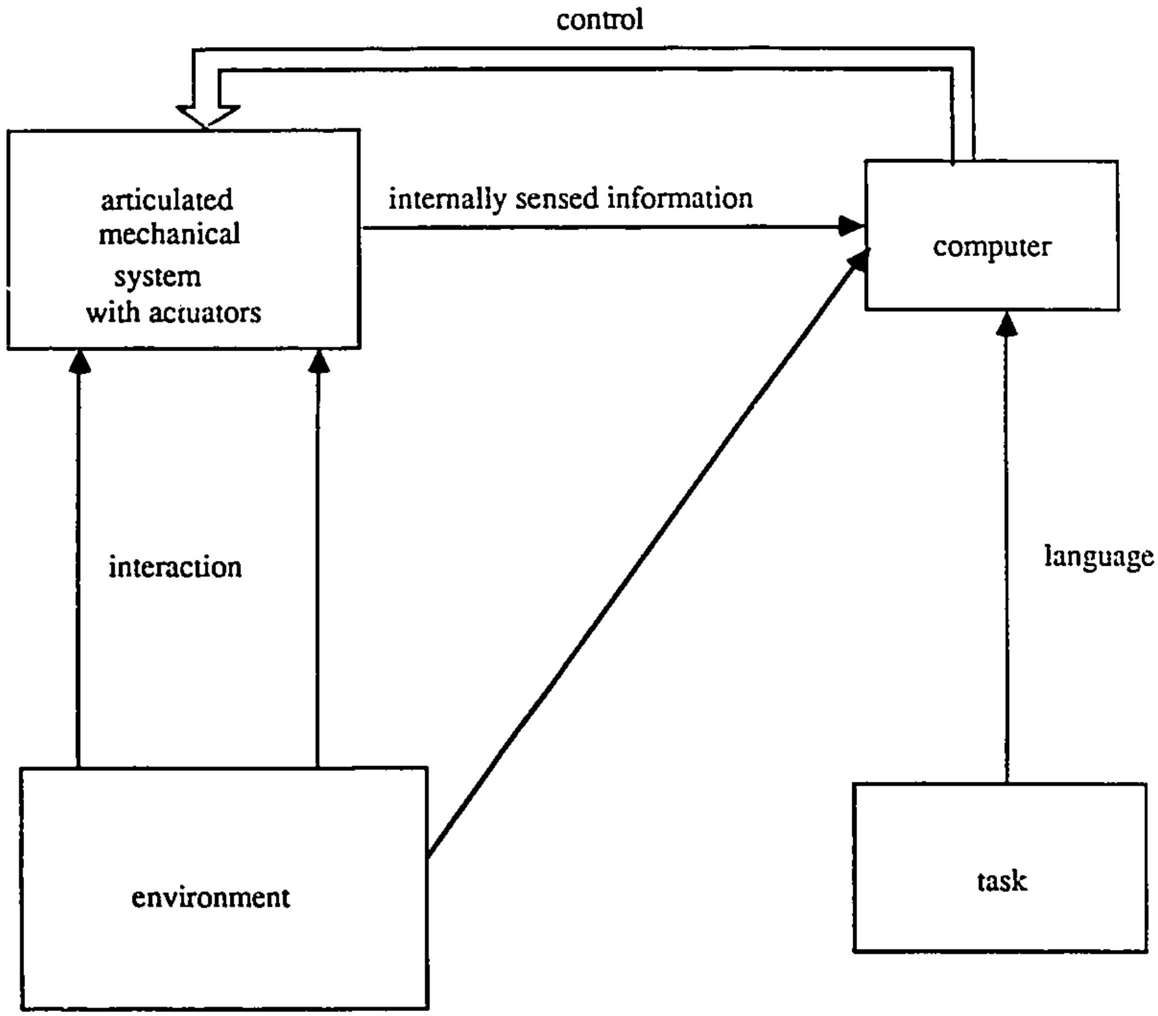

Figure 5.1: Basic Robot Structure

dynamics.

5.2 General Structure of Robot Control

In its most general form Four parts constitute a robot system as indicated by Figure 5.1.

1. Articulated Mechanical system (AMS): This part represents the mechanical part of the robot which consists of several links driven by actuators at each joint.

2: Environment: Is the work volume of the robot and all objects present within that volume.

3. Controller: The robot controller generates approximate signals necessary to guide the end effector to perform a required task.

4. Task: Is defined by the source and the destination of the robot end effector, and may include the definition of the path between these two points.

In order for the controller to keep track of the actual state of the AMS, internal sensors are used to give information about the position and the velocity of each joint. Encoders are usually used as position sensors and tachometers as velocity sensors.

The interaction between the robot controller and the environment is accomplished by using external sensors located at the end effector of the AMS or monitoring the AMS from a fixed point in the world coordinate system. These sensors may be visual like an electronic camera, or tactile sensors, B. Benhabib [34].

In this thesis, it is assumed that there is no feedback from the environment and only internal sensors are used.

A robot task is usually specified in terms of the position of the manipulator's end effector with respect to an inertial reference frame. The required task is fed into the robot controller through an appropriate language that can be interpreted by the robot computer. The task is then transformed by the robot computer from the world coordinates into joint coordinates by solving the inverse kinematic problem. The results of the transformation are used to generates the inputs to the joint controllers required to achieve the desired positioning of the end effector.

There are two types of robot tasks:
1) Point to point (PTP): This task does not care about the path followed and the main goal is to reach the destination point as fast as possible.
2) Trajectory tracking: This task is defined by a pre-programmed trajectory that has to be followed by the robot end effector. This task can be performed by considering the trajectory as a large number of PTP tasks. An other method to perform a trajectory tracking task is known as the computed torque, this method is mainly experimental.

5.3 Individual Joint controller

The vast majority of industrial robot task are of the pick and place type, and do not follow a specified path. The controller for this type of task has to ensure that the robot end effector moves from the source location and reaches the destination. There is no coordination among the motion of the joints, then the path between any two points is arbitrary. This action can be accomplished by using a simple position controller. A velocity feedback signal provides the required damping. A simplified representation of a joint control system is shown in Figure 5.2. At the axis of the joint there are two sensors. An encoder is used to provide a position feedback of the joint by generating a voltage proportional to the angle θ, a tachometer is used

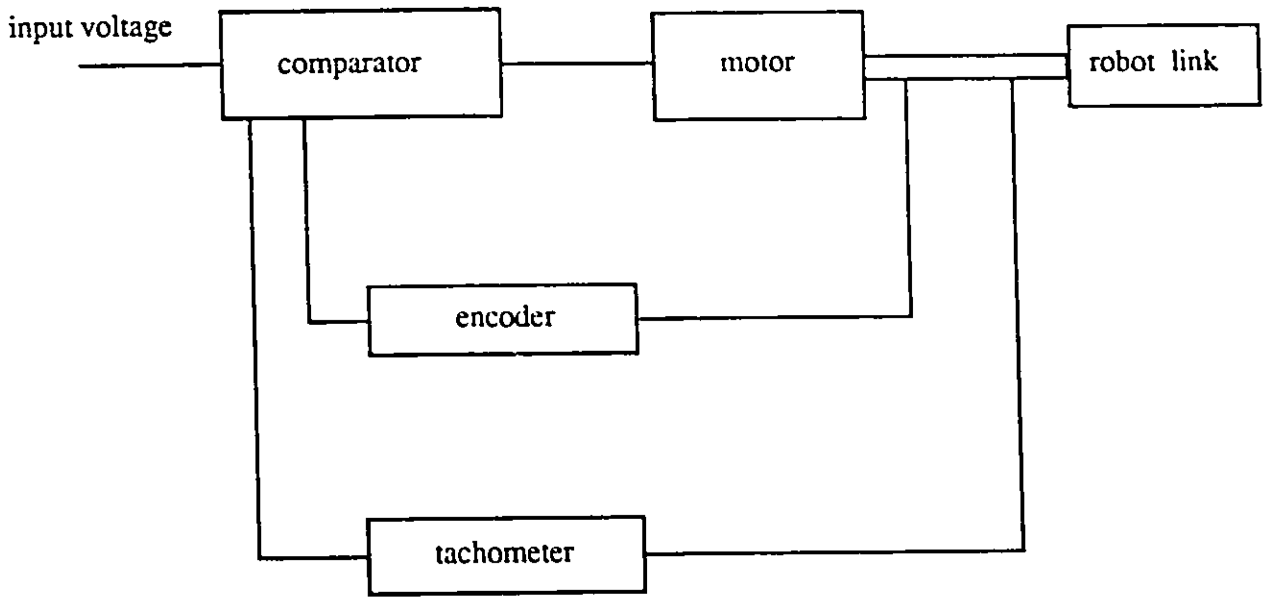

Figure 5.2: Block diagram of a servo-joint

to provide a voltage proportional to the joint velocity; Pradeep K. Khosla [32]. The sum of the two voltages is used to oppose an input voltage and causes the joint to move toward an equilibrium position ($\theta = \theta_d$).

The actuating signal used to energize the actuators is given by:

$$u_i = K_{ip}e_i + K_{id}\dot{e}_i \tag{5.1}$$

Where $e_i = \theta_{id} - \theta_i$ is the position error. K_{ip} and K_{id} are the position, and velocity gains, respectively. The main purpose of K_p is to improve the speed of response. The velocity gain is to enhance the stability and reduces the tendency toward oscillation.

When the system is subjected to an unmodelled disturbances, an offset error may appear as a steady state error. In the case of an unbalanced link, the amount of unbalance usually causes this offset error. To overcome this problem, an integration part is added to the actuating signal as:

$$u_i = K_{ip}e_i + K_{id}\dot{e}_i + K_{iI}\int_0^t e_i d\tau \tag{5.2}$$

The offset error in the case of an unbalanced link can be eliminated by a feedforward conpensation of the amount of unbalance.

5.4 Computed Torque Technique

In some cases, where the robot is required to follow a specified trajectory in space, this trajectory is planned off line and, the position, velocity and acceleration of each joint at each instant of time are specified. The information are fed to the joint controller for execution. The robot controller coordinates the action of all the joint controllers based on these information.

The other approach mentioned earlier is the computed torque technique. Figure 5.3 shows a block diagram of the computed torque algorithm The method utilizes nonlinear feedback loop $h(\theta, \dot{\theta})$ to decouple the manipulator. In effect, the torque required at every instant of time to cause the end effector to follow the path is calculated and the controller/motor arrangement is directed to produce that torque. The required torque τ is calculated based on the inverse dynamic equation given by:

$$\tau = H(\theta)u + h(\theta, \dot{\theta}) + G(\theta) \tag{5.3}$$

Where u is the vector of the actuating signal from the controller.

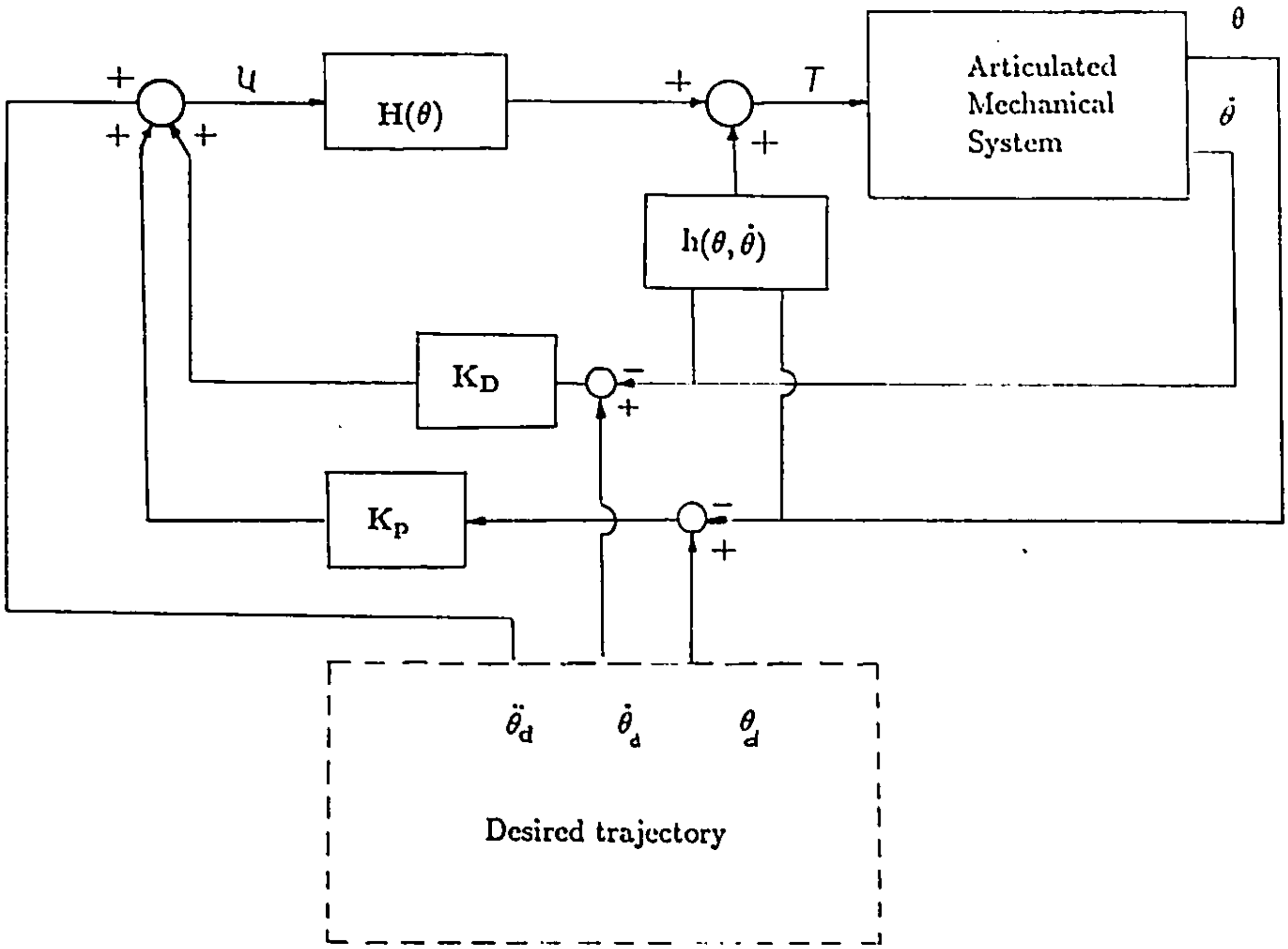

Figure 5.3: Block diagram of the computed torque algorithm

The purpose of using the above equation is to reduce the system to that of a unit mass described by:

$$\ddot{\theta} = u \tag{5.4}$$

When a PD controller is used to control the motion of joint i, the actuating signal is calculated using:

$$u_i = \ddot{\theta}_{id} + K_{ip}e_i + K_{id}\dot{e}_i \tag{5.5}$$

Where $\ddot{\theta}_{id}$ is the acceleration of the desired trajectory associated with the motion of joint i. Substituting Equation 5.5 into Equation 5.3 we get:

$$\tau_i = \sum_{j=1}^{3} H_{ij}(\theta)[\ddot{\theta}_{jd} + K_{jp}e_j + K_{jd}\dot{e}_j] + h_i(\theta, \dot{\theta}) + g_i(\theta) \tag{5.6}$$

5.5 Controller Design

The performance of the control schemes presented in the previous sections is determined by the selection of their parameters. Both Analytical and numerical methods are used to evaluate these parameters, depending on the nature of the system and the control algorithm used.

5.5.1 Design of the gain matrices for the individual joint controller

Analytical method

By using an individual joint controller, the control of a multi-link manipulator becomes the control of each individual link alone with out taking into account the effect of the other links. Figure 5.4 shows a block diagram of the control of a R-R-R manipulator using the individual control approach. Each link has a mass m_i, a moment of inertia about its joint axis and a structural resonant frequency ω_s as a result of its structural stiffness; Paul [20]. The effect of the payload and the other links causes an increase in the effective mass and inertia of the given link which causes the structural frequency ω_s to decrease. Paul [20] has found that this decrease in the frequency is approximated by the square root of the relative increase in the inertia.

$$\frac{\omega}{\omega_s} = \sqrt{\frac{J_s}{J}}$$

Each link is driven by an electric actuator through a gear reduction ratio N_i. The inertia of the link and its actuator can be written as :

$$J_{ii} = H_{ii} + J_{ai}N_i^2$$

Where H_{ii} is the effective inertia of link i (the ii element of the H matrix) and J_{ai} is the inertia of the actuator driving the link i.

If we consider the proportional plus derivative feedback controller in order to control the link having an effective inertia J_{ii}, the control system

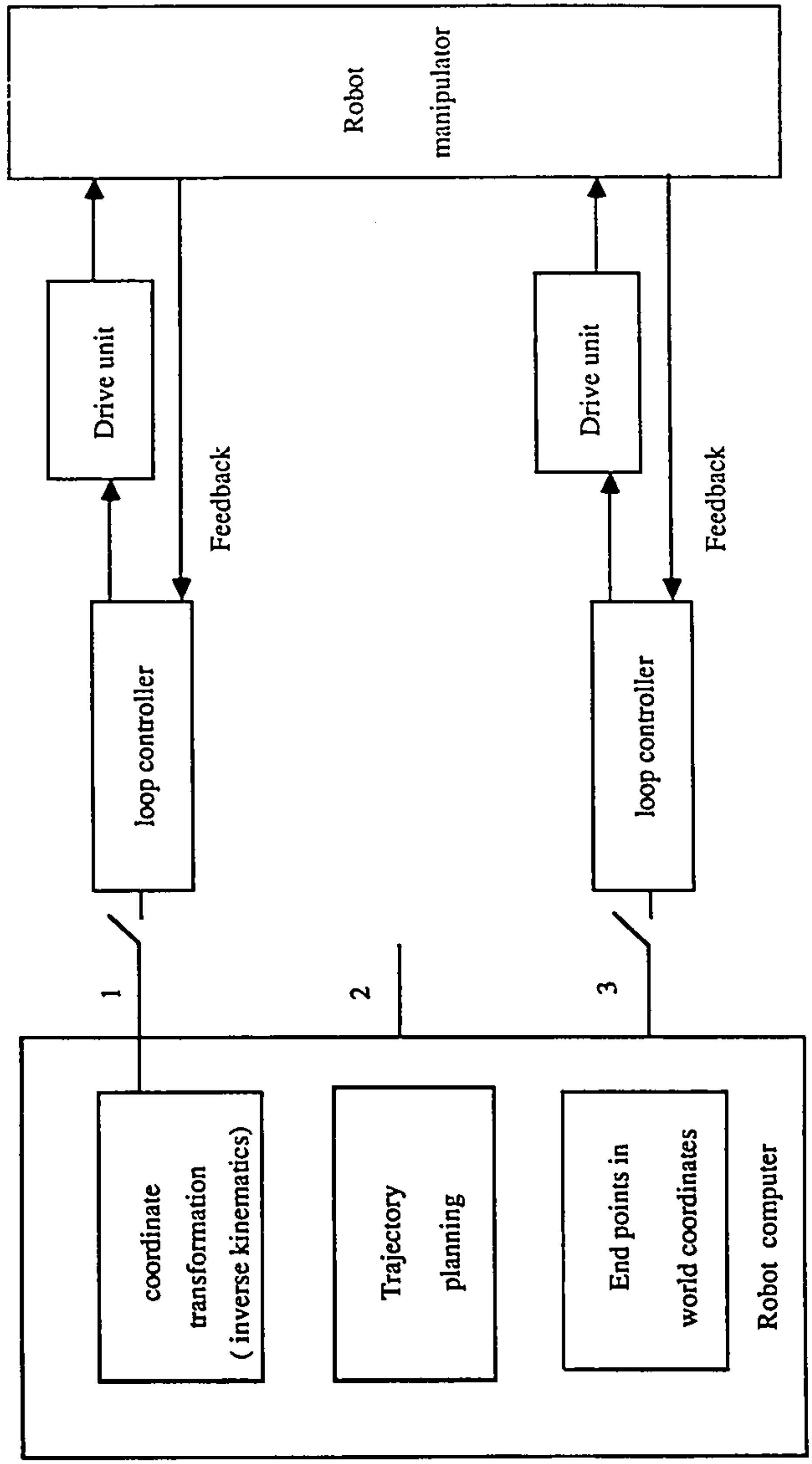

Figure 5.4: Individual joint control of a R-R-R manipulator

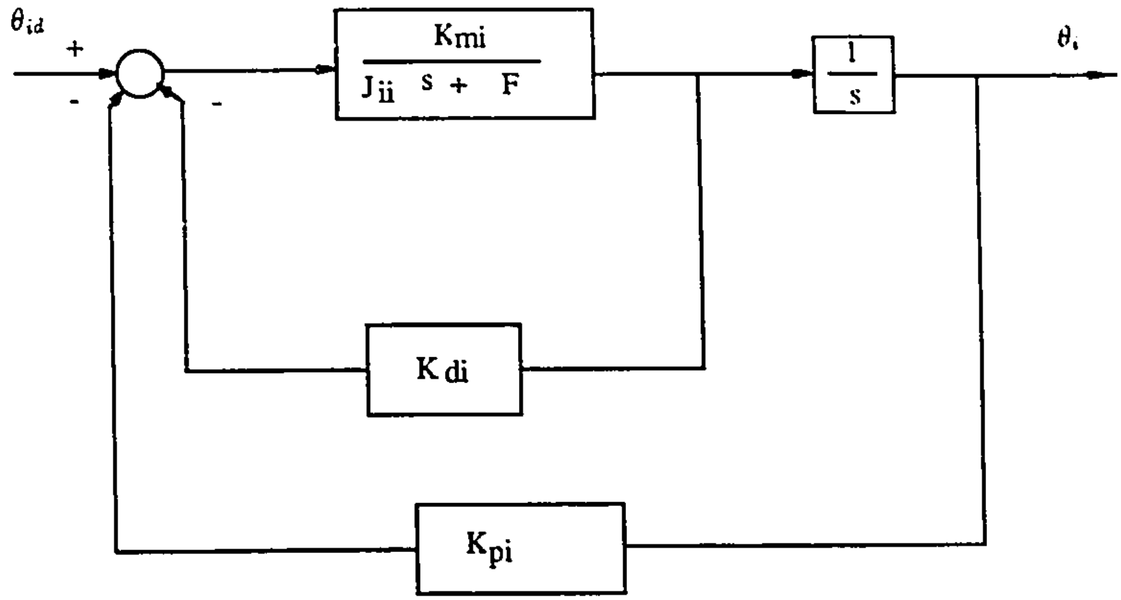

Figure 5.5: Block diagram of a PD joint control

can be described by the block diagram of Figure 5.5. The closed loop transfer function is given by:

$$\frac{\theta(s)}{\theta_d(s)} = \frac{K_{mi}}{J_{ii}s^2 + (F_i + K_{di}K_{mi})s + K_{pi}K_{mi}} \tag{5.7}$$

Where:

$$F_i = \frac{K_{bi}K_{mi}}{R_i}N_i^2$$

Equation 5.7 is the equation of a second order system with characteristic equation in the form:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \tag{5.8}$$

From Equations 5.7 and 5.8 we have

$$\omega_n = \sqrt{\frac{K_{pi}K_{mi}}{J_{ii}}} \tag{5.9}$$

$$\zeta = \frac{F_i + K_{di}K_{mi}}{2\sqrt{J_{ii}K_{pi}K_{mi}}} \tag{5.10}$$

The system is critically damped when $\zeta = 1$ which gives the following relation between K_{pi} and K_{di}.

$$K_{di} = 2\sqrt{J_{ii}\frac{K_{pi}}{K_{mi}}} - \frac{F_i}{K_{mi}}$$

Numerical method

Figure 5.6 show a block diagram of the tuning process. The optimization algorithm uses the simplex method by Nedler and Mead described by D. J. wide [35]. The method is based on the optimization of the parameters of the controllers by minimizing the ITAE performance index. Figure 5.7 shows the details of the algorithm for calculating the ITAE performance index. To take into account the effect of the interaction between the links, the objective function is taken as the sum of the ITAE of each joint.

$$F(X) = \sum_{i=1}^{3} ITAE_i \tag{5.11}$$

Where X is the design vector containing the proportionality, derivation and integration constants of each controller.

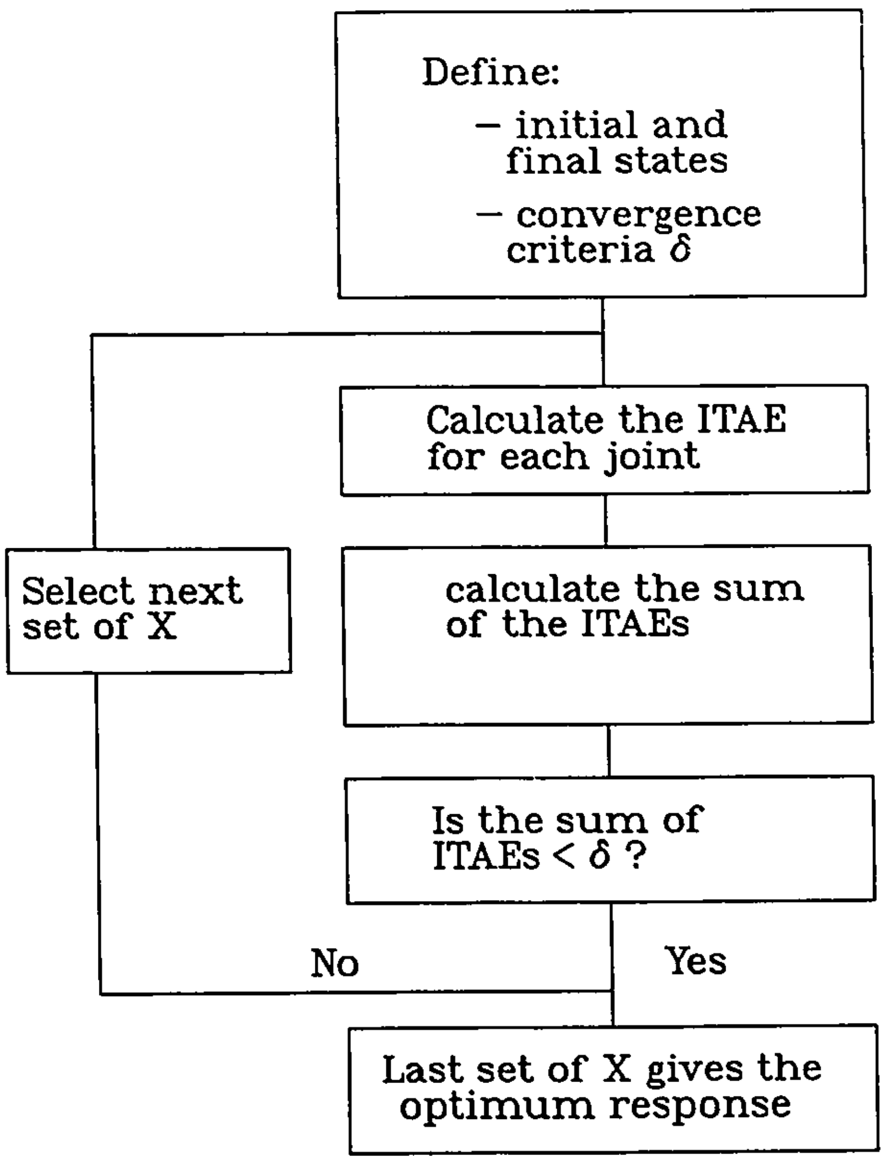

Figure 5.6: Block diagram of the tuning process

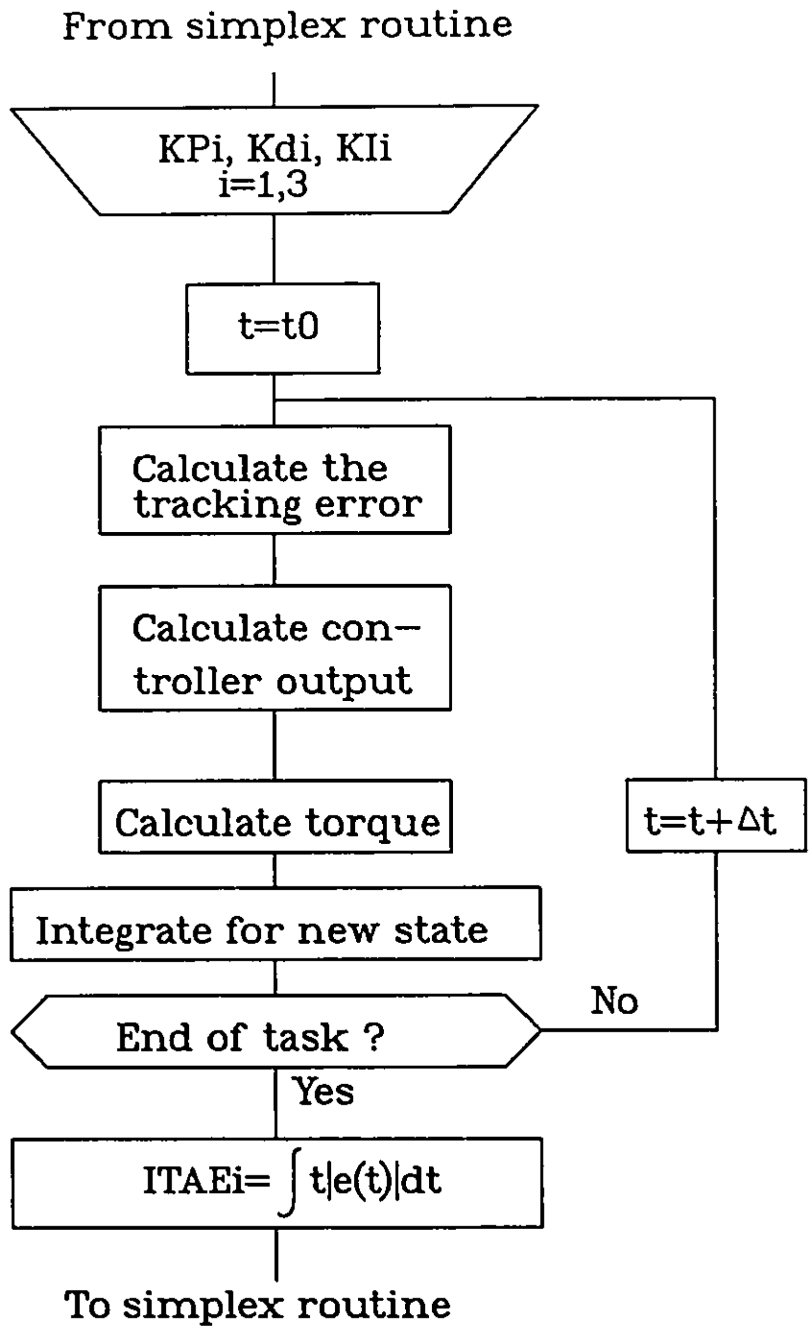

Figure 5.7: Flow chart of the ITAE evaluation

5.5.2 Design of the Gain Matrices for the computed torque scheme

The basic idea of the computed torque scheme is to reduce the system to a unit mass one described by Equation 5.4. Substituting Equation 5.4 into 5.5, gives the closed loop characteristic equation of joint i.

$$\ddot{\theta}_{di} - \ddot{\theta}_i + K_{di}(\dot{\theta}_{di} - \dot{\theta}_i) + K_{pi}(\theta_{di} - \theta_i) = 0 \tag{5.12}$$

Transforming Equation 5.12 into Laplace domain gives:

$$s^2 + K_{di}s + K_{pi} = 0 \tag{5.13}$$

This equation is that of a second order system with

$$2\zeta\omega_n = K_{di}$$

and

$$\omega_n^2 = K_{pi}$$

For typical robot application it is required that the end effector does not overshoot its destination. This is essential to prevent collision in tasks particularly those involving pick and place. Robot systems are hence designed with overdamping i.e, $\zeta \geq 1$. The fastest response will be that for $\zeta = 1$, this condition results in the following relation between K_{di} and K_{pi} [34].

$$K_{di} = 2\sqrt{K_{pi}} \tag{5.14}$$

For both the individual joint controller and the computed torque, the value of K_{pi} is to be chosen as large as possible in order to achieve disturbance rejection. In practice, the value of K_{di} is limited by the noise present in the velocity measuring device, which limits the value of K_{pi} as shown by Pradeep K. [32].

5.6 Gravity compensation in Robot joint controller design

With reference to Equation 4.18, the torque required to hold the robot at a given position is given by:

$$\tau_r = \mathbf{G}(\theta) \tag{5.15}$$

For an unbalanced arm, the torque τ_r has to be provided by the actuators continuously. In order to generate this static torque τ_r, an additional signal has to be added to the actuating signal as shown in Figure 5.8. This approach is known as a feedforward compensation. When The arm is moving, the torque provided by the actuators is:

$$\tau = \tau_r + \tau_d \tag{5.16}$$

Where τ_d is the torque available to move the robot. Due to the limited capacities of the actuators, there is always a maximum torque that cannot be exceeded without damaging these actuators; therefore:

$$\tau_r + \tau_d \leq \tau_{max} \tag{5.17}$$

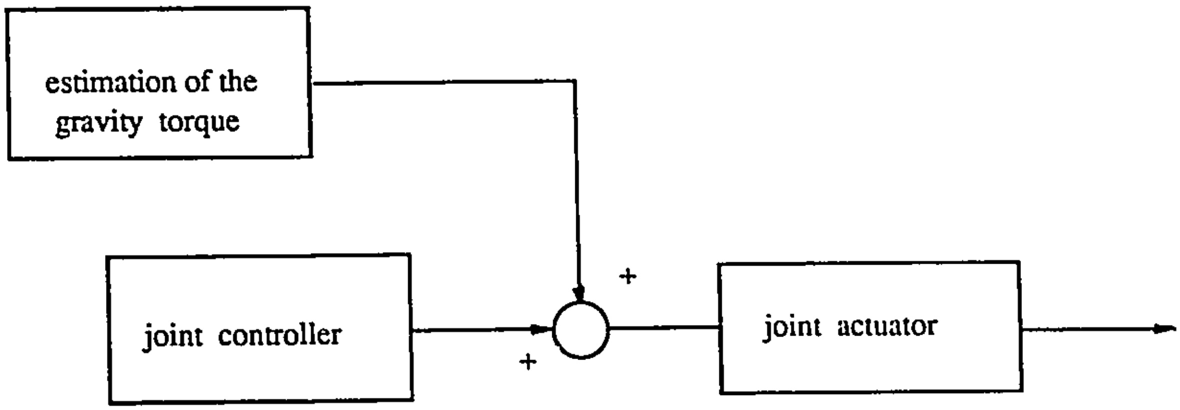

Figure 5.8: Feedforward Compensation

For an unbalanced arm, the static torque τ_r is counteracted by the balancing mechanism, which gives more torque available to move the robot at a higher speed or to carry more heavier load.

Chapter 6

SIMULATION OF THE ROBOT RESPONSE

6.1 Introduction

In the time response analysis, the independent variable is time and the unknowns to be found are the position, velocity and acceleration associated with each link.

To obtain the time response of the manipulator, the differential equations representing the manipulator dynamics are to be solved simultaneously. These equations, as described earlier, are highly nonlinear and coupled set of second order differential equations. Therefore, an analytical solution cannot be achieved and a numerical integration techniques has to be used. The numerical integration method used is the forth order Runge-Kutta. Although this method is not the most efficient, it is among the most stable ones. An integration time step of 0.02 was found to be the largest that can be used, thus the most economic, without resulting in inaccuracies.

6.2 The Robot Model

The Robot used in the simulation has an approximate parameters equivalent to those of the ASEA L6/2. Since the detailed robot links characteristics are not published, the mass and inertia of each link are estimated based on the dimensions and the material used . Table 6.1 is a summary of the physical characteristics of each link.

link	m_i	a_i	l_i	I_{ix}	I_{iy}	I_{iz}
1	-	0.5	-	0.5	-	-
2	16.0	0.9	0.3	0.12	0.12	0.01
3	12.0	0.7	0.2	0.12	0.12	0.01

Table 6.1: Manipulator Characteristics

The robot is equipped with PMI S9M4H servodisc motors with harmonic drive. The pertinent specifications of these motors are given in Table 6.2.

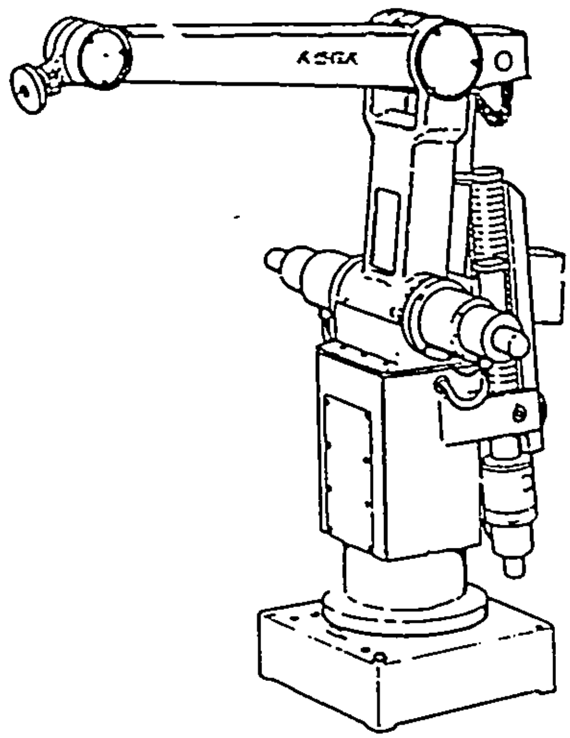

Figure 6.1: ASEA L6/2 Robot

K_i	K_{bi}	I_{max}	R_i	N_i
0.084	0.084	14.6	1.2	120

Table 6.2: Actuators Characteristics

6.3 Simulation Results

6.3.1 Point To Point Control (PTP)

Point to point control is mainly used in pick and place operations and in spot welding. During these operations the end effector of the manipulator has to move from one initial point to a final target point. Since the main goal is to reach the target point, the path followed is not a major concern of the controller. This task is very easy to perform by using an individual joint controller at each joint. Given an initial point $P_1(x_i, y_i, z_i)$ and a final point $P_2(x_f, y_f, z_f)$, a transformation is made to express the position of the manipulator in terms of the joint coordinates θ_1, θ_2 and θ_3. This transformation is necessary since the control is to be performed at the actuators level. Figure 6.2 shows a flow chart of the simulation algorithm based on the point to point control task. With reference to the figure, the initial and final configurations of the robot are used as an input to the program. The error signals are calculated from the differences between the initial and final configurations. These signals are compared to the allowable tolerance representing the steady state error. If the error is larger than ϵ, the error signals are fed to the controller, which calculates the actuating signal. The output is amplified by a current amplifier and the result becomes an input to the joint actuator. Since all actuators have a maximum amount of current that should not be exceeded, a current limiter is being implemented. The actuator produces a torque proportional to the current from the amplifiers. In order to obtain the response of the link to a specified input torque, the equations representing the robot model are solved. The results from the integration routine are compared to the desired configuration, and a new error signals are calculated. The process continues till the error reduces to a small allowable value.

In order to compare the responses of spring balanced and weight

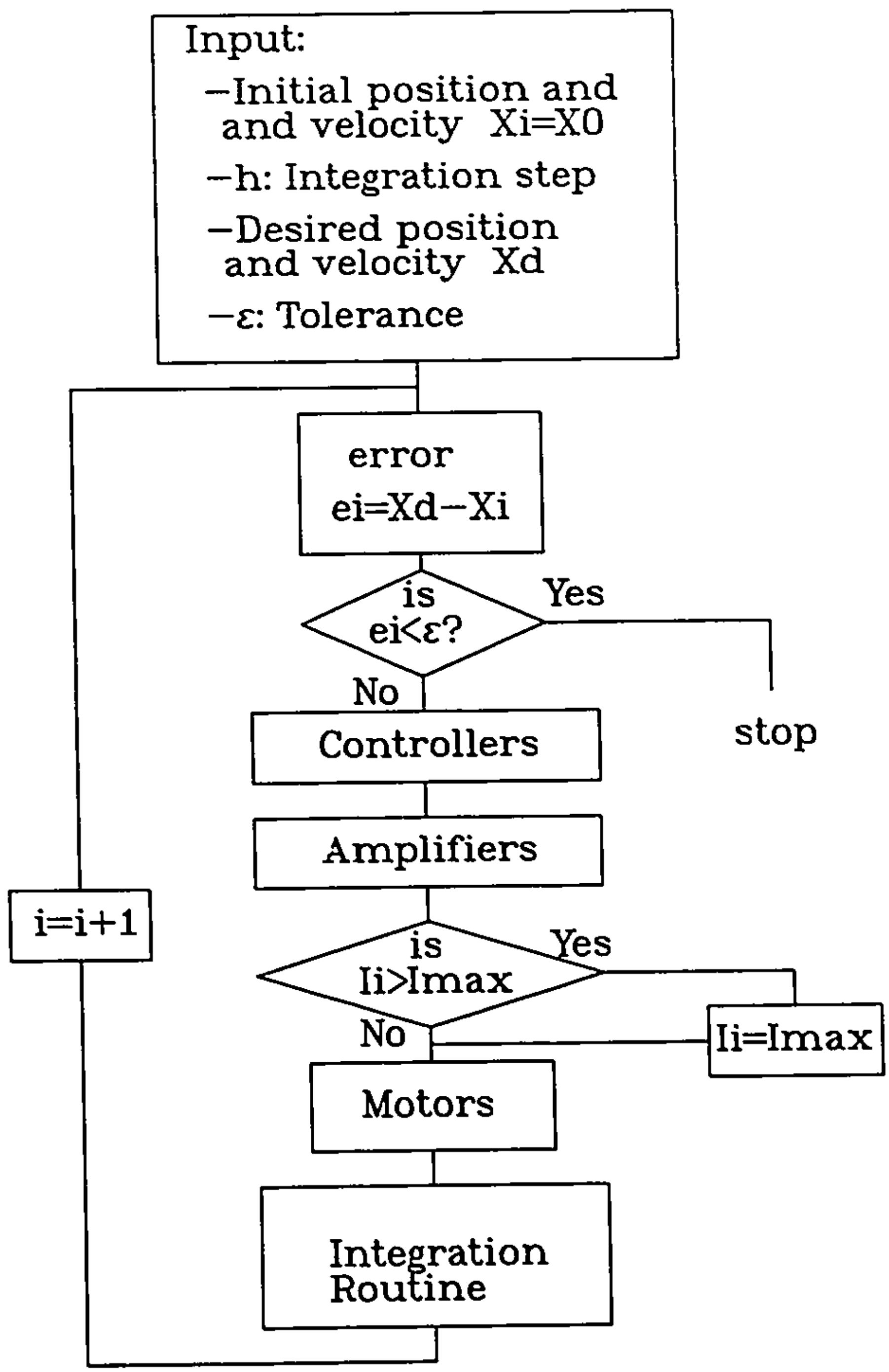

Figure 6.2: Flow chart of the simulation program

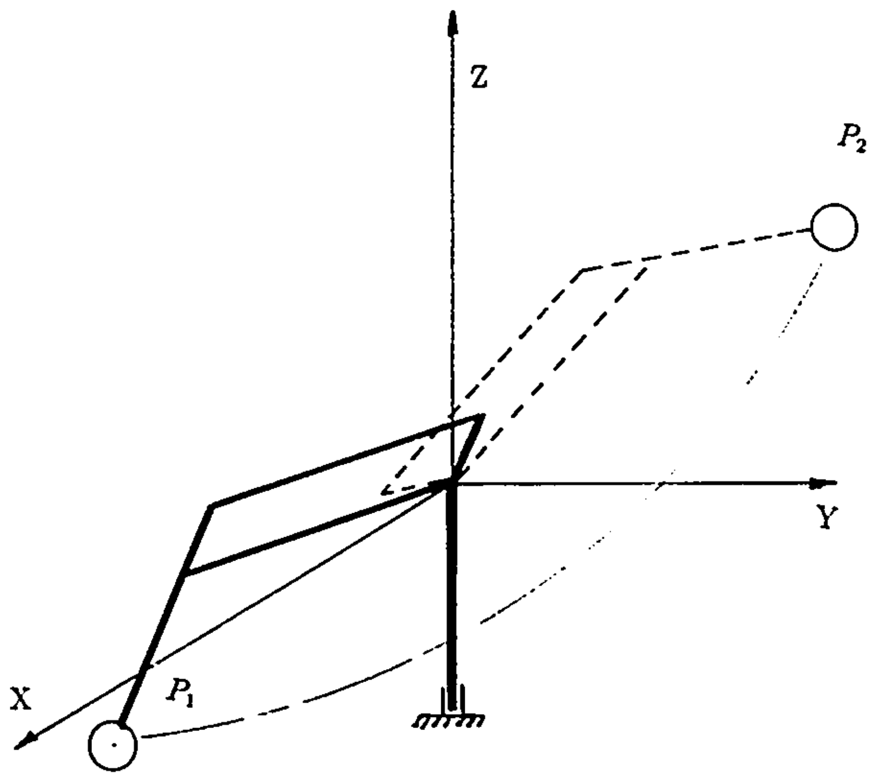

Figure 6.3: Initial and final configuration

balanced arms, two robots with such arms were commanded to move from P_1(1.4,0,-0.35) to P_2(0.0,0.9,0.8). Figure 6.3 shows the initial and final configuration of the robot.

Effect of Tuning

The tuning of the controller parameters is carried out numerically and analytically. For the case of numerical tuning, the joints were found to overshoot their target positions. This is due to the fact that the optimization problem minimizes the overall error of the joint positions without any regards to the resulting response. This case resembles a response of a system with invariant parameters and having a damping ratio of 0.707. Figures 6.4 and 6.5 show the response of the robot as it moves from P_1 to P_2 when the tuning is done numerically. Both, the weight counterbalanced and the spring counterbalanced arms are fully balanced at 5 kg. As it can be seen from figure 6.4, when the controller is tuned numerically, it allows more overshoot because of the high error area between t=0 and the rise time. Figure 6.5 shows that for joint 1, the weight balanced arm is moving at a slower speed then the spring balanced arm because of the higher inertia due to the counterweights. At joint 2, although both systems have similar rise time, the error area is less in the case of a spring balanced arm, furthermore, the spring balanced arm reach steady state faster. The same phenomena can be observed at joint 3.

The analytical tuning, as described earlier in chapter 6, is based on a critically damped response when the inertia reflected at each joint is the highest. The controllers are tuned for the case when the robot is carrying a load of 5 kg at the configuration when θ_2=0 and θ_3=0. ie; both, link 2 and link 3 are fully extended. The result of the analytical tuning is shown in Figure 6.6 and 6.7. With reference to Figure 6.6, the motion of the three joints is over damped since the robot does not pass by the configuration when θ_2=0 and θ_3=0.

Figures 6.6 and 6.7 show that When the spring balanced and the

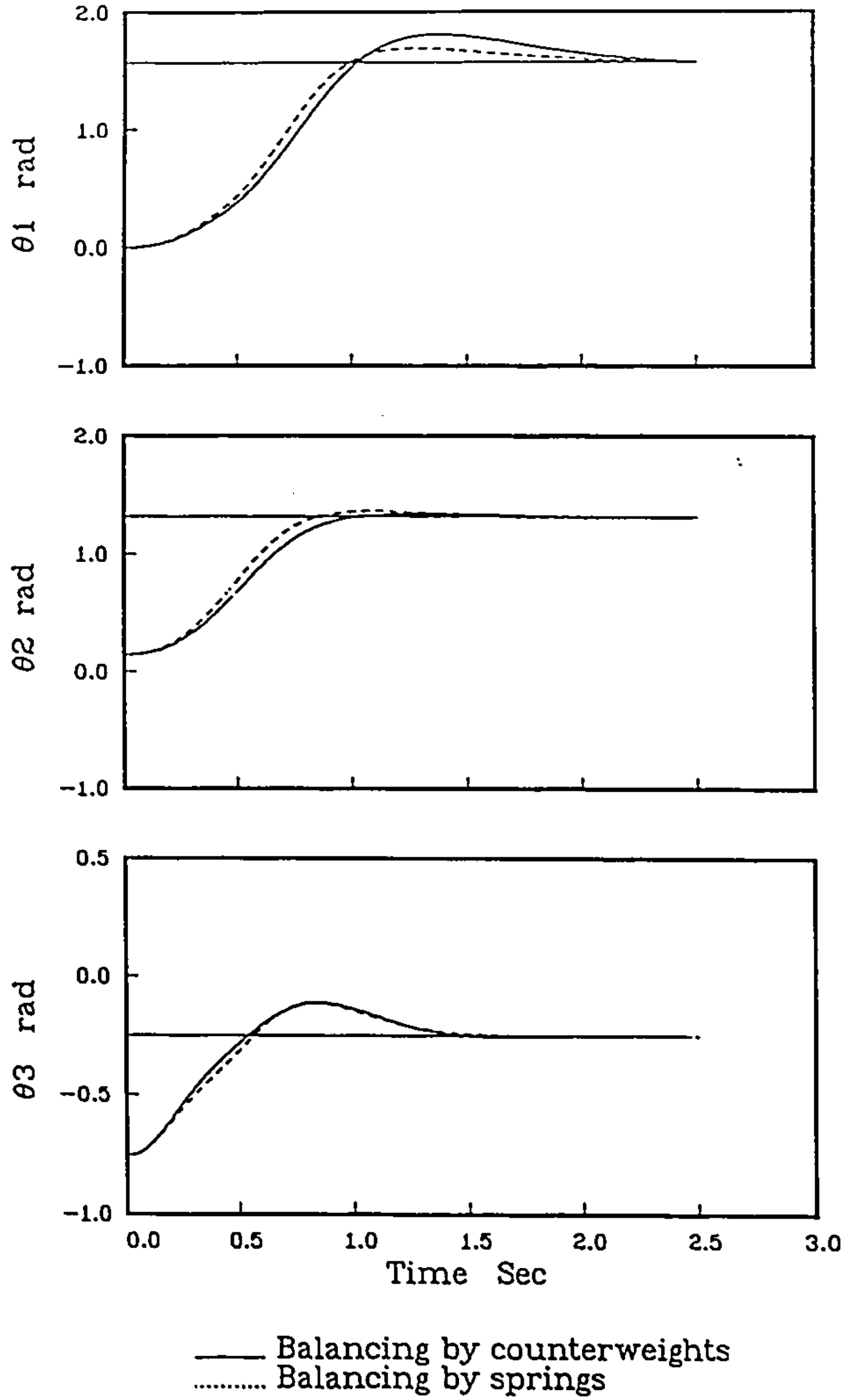

Figure 6.4: Position of joint 1,2 and 3 for numerical tuning using the ITAE criteria, systems balanced for 5 kg

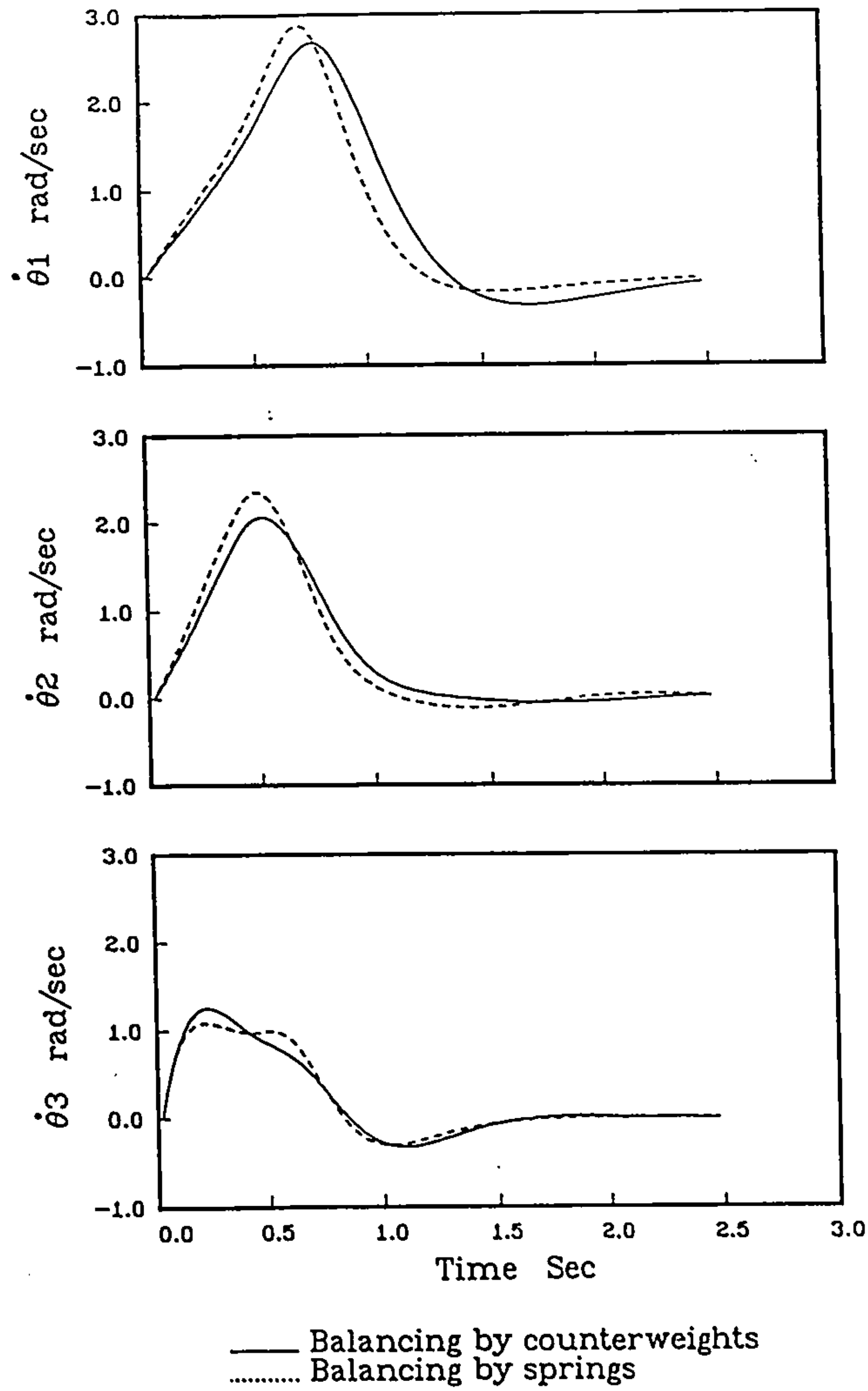

Figure 6.5: Velocity of joint 1,2 and 3 for numerical tuning using the ITAE criteria

weight balanced arms are counter balanced for a specific load, the arm using springs for counter balancing exhibits a faster response as compared to that of the weight counter balanced arm. This slow and sluggish response is due to added inertia of the counterweights. The irregularity in the velocity curve of joint 3 shown in Figure 6.7 is due to the effect of coupling between joint 3 and joint 2. It occurs at the instant when the acceleration of joint 2 changes from positive to negative. The effect of coupling can be seen more clearly in the case of spring counter balanced arm because in this case the velocity and acceleration of joint 2 are larger than in the case of a weight counter balanced arm.

Effect of balancing

When the robot is not carrying the load for which it is balanced, an offset appears as a steady state error in the positions of joints 2 and 3. This error is due to the amount of unbalance experienced at these joints. The offset error is eliminated by introducing an integration part into the control algorithm. In order to study the effect of balancing, the response of the weight balanced and spring balanced arms are obtained for three cases; when the arms are carrying no load, a load of 5 kg, and a load of 10 kg. Both arms are balanced for a load of 5 kg. The responses of the joints are over damped responses for both systems and faster for the case of the spring balanced arm, as shown in Figure 6.8 to 6.13. As the load increases, the response of both systems becomes slower because of the increase in the inertia, but the spring balanced arm is always faster then the weight

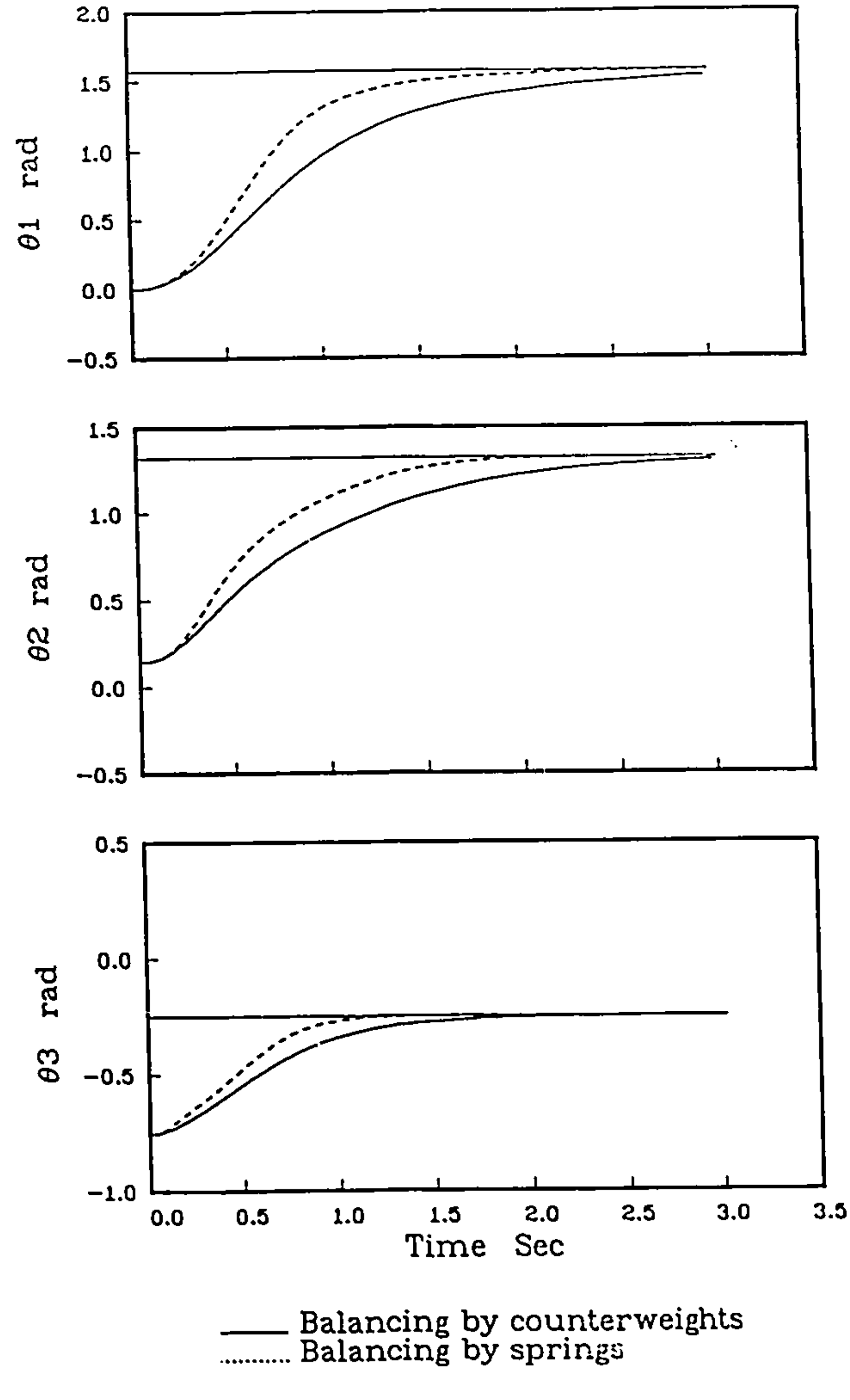

Figure 6.6: Position of joint 1,2 and 3 for analytical tuning based on the highest inertia, systems balanced for 5 kg

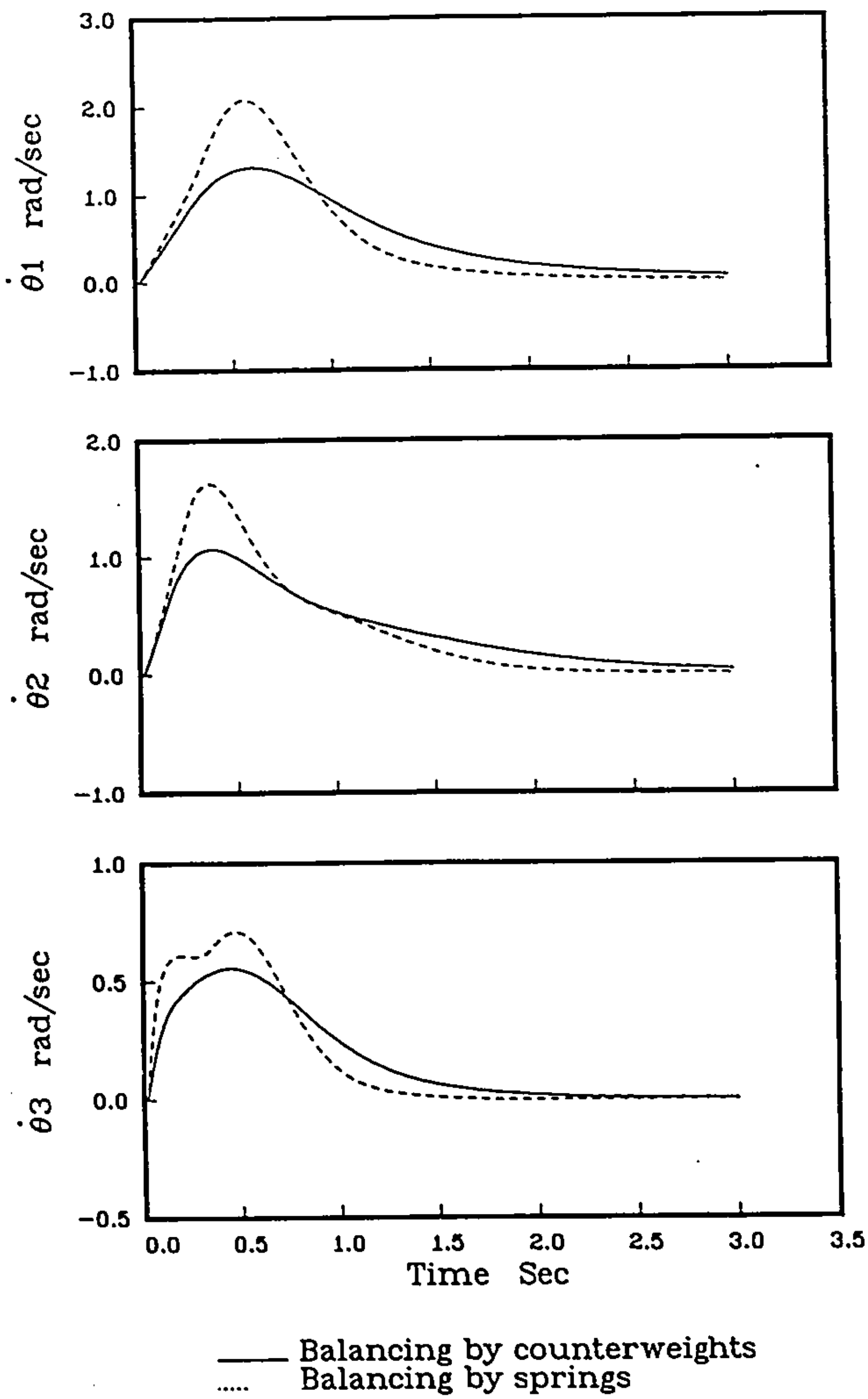

Figure 6.7: Velocity of joint 1,2 and 3 for analytical tuning based on the highest inertia, systems balanced for 5 kg

balanced arm.

In a pick and place cycle, the robot performs two sub-cycles. In the first cycle, the robot carries the load from the source to the destination. In the second sub-cycle, the robot returns while carrying no load to the source in order to start another pick and place cycle. The responses of the joints as the end effector moves from P_2 to P_1 are shown in Appendix A Figures A.1 to A.6. If the robot system was one with an invariant inertia, the responses when the robot is moving according to a PTP scheme from P_2 to P_1 would have been a mirror image of that when the robot is moving from point P_1 to P_2. However, for a robot system the inertia reflected at each joint is configuration dependant. Since in a PTP scheme no constraint is imposed on the trajectory of the robot, the responses when moving from P_2 to P_1 bear no resemblance to those when moving from P_1 to P_2.

The effect of the coupling between joint 2 and joint 3 can be seen very clearly in the velocity profile of joint 3. As the acceleration of joint 2 changes from positive to negative at $t = 0.4$ sec, the shape of the position and the velocity of joint 3 are affected. as can be seen from Figure A.6, the effect of coupling becomes more feasible with the increase of the payload.

6.3.2 Trajectory Control

For the PTP control, the motion of the end effector of the robot originates at a source point and terminates at a destination point, while the path

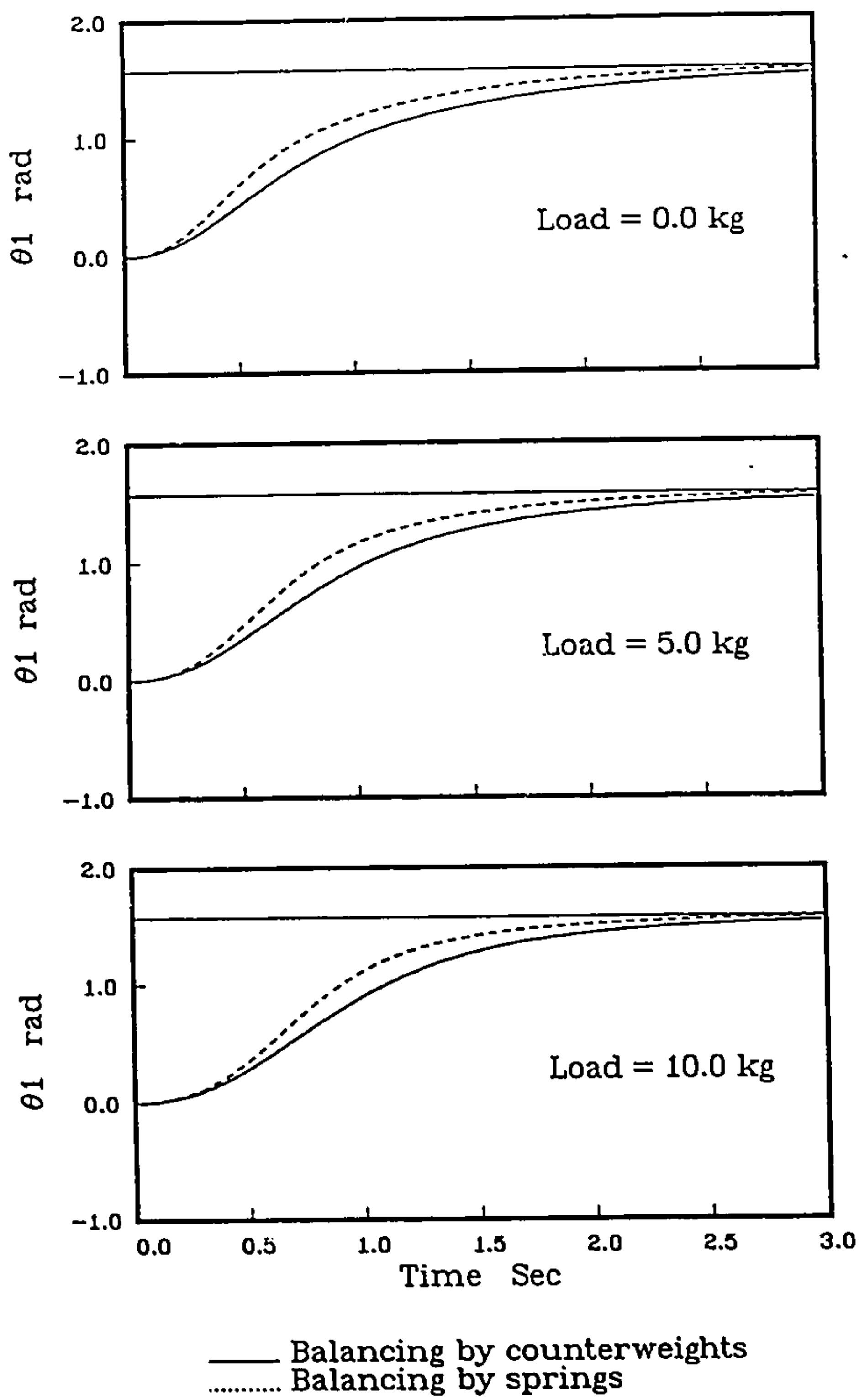

Figure 6.8: Position of joint 1, systems balanced for 5 kg

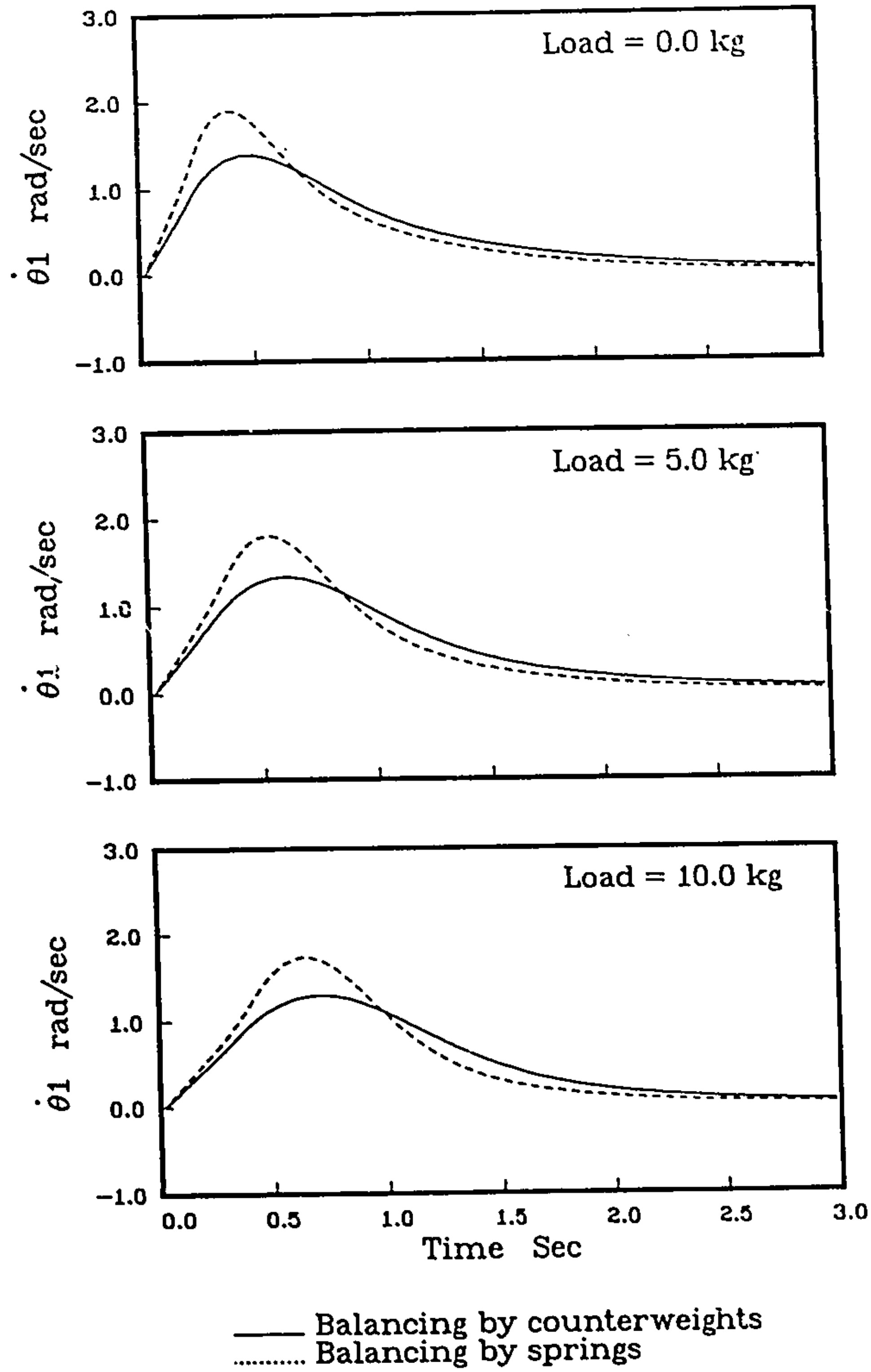

Figure 6.9: Velocity of joint 1, systems balanced for 5 kg

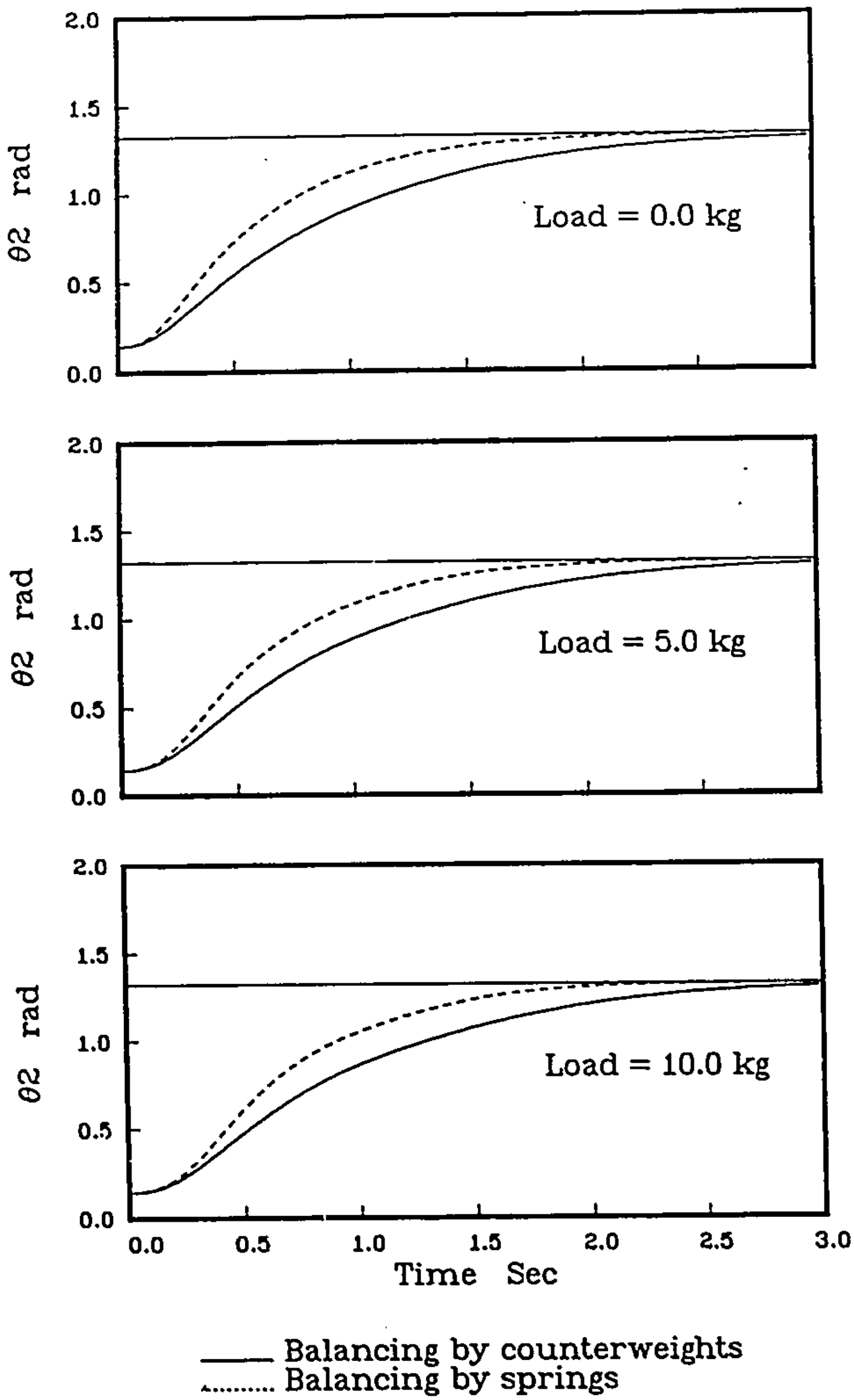

Figure 6.10: Position of joint 2, systems balanced for 5 kg

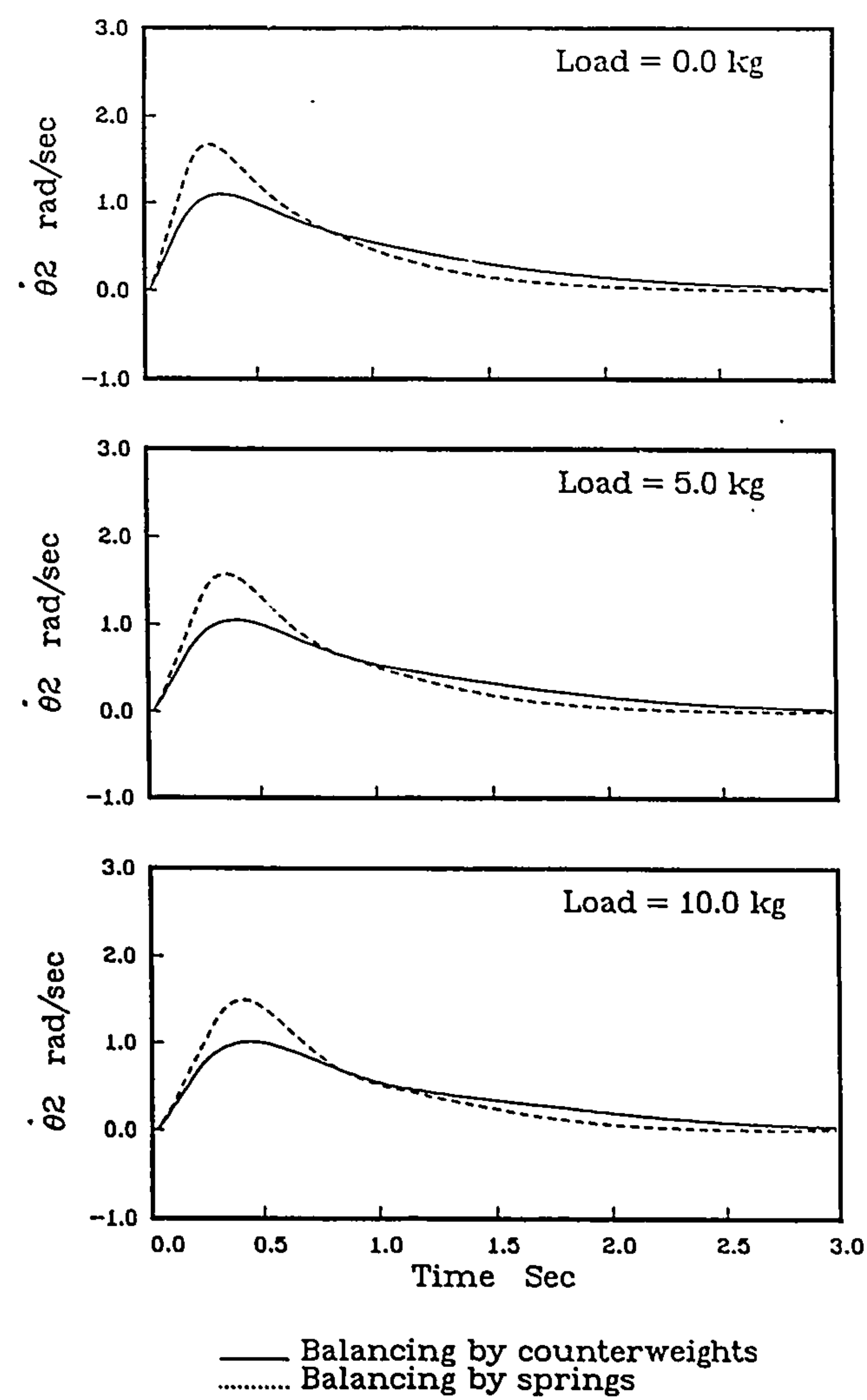

Figure 6.11: Velocity of joint 2, systems balanced for 5 kg

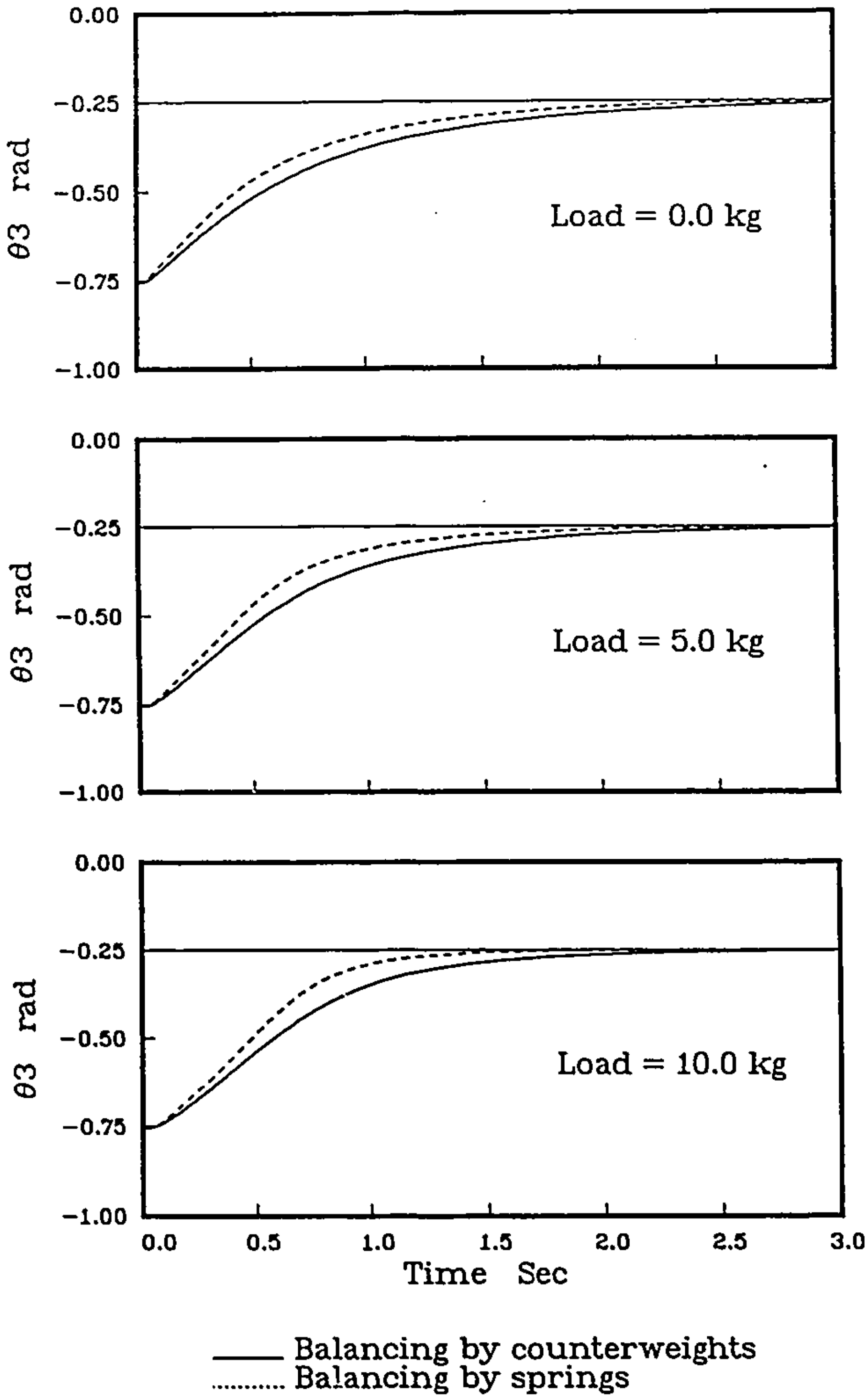

Figure 6.12: Position of joint 3, systems balanced for 5 kg

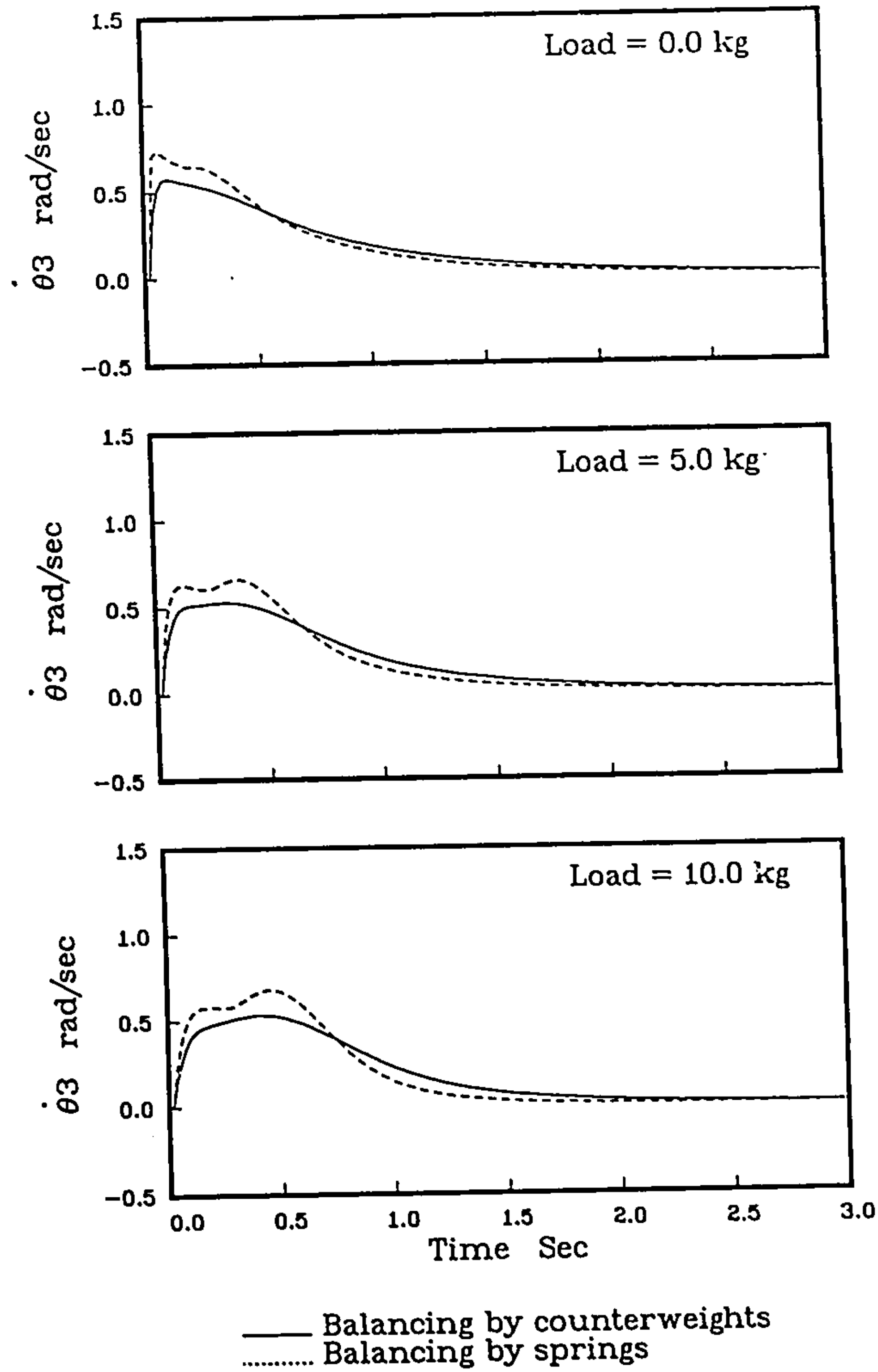

Figure 6.13: Velocity of joint 3, systems balanced for 5 kg

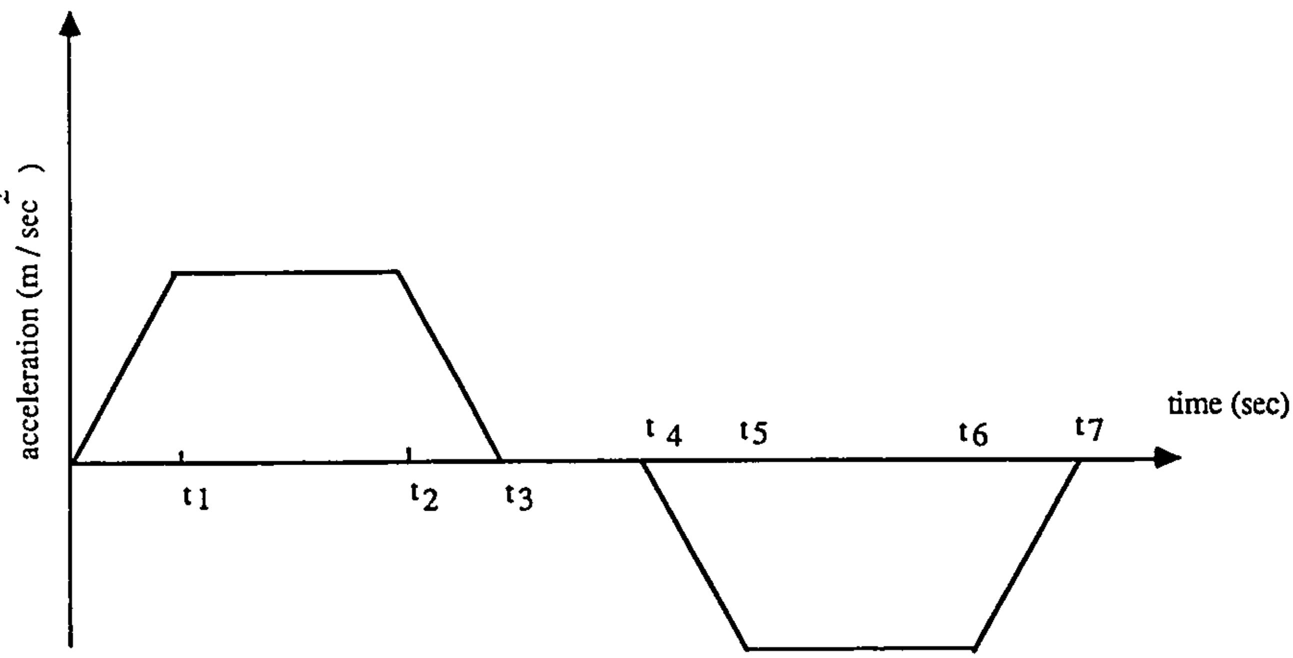

Figure 6.14: Acceleration profile of the trajectory

between these two points is arbitrary, as has been indicated in the previous section. When a specific path is required between the source and destination points, a different method of robot control is required. Several methods may be used to accomplish this task. The one employed here is the computed torque method.

Trajectory used for simulation

Figure 6.14 shows the acceleration profile of the trajectory used for obtaining the dynamic response of the weight balanced and the spring balanced arms. As can be seen from the figure, this trajectory consists of 7 segments: a) an initial segment $[0, t_1]$ in which the acceleration is increased from zero to its maximum. This segment is introduced so as to eliminate the discontinuity of the acceleration at start, ie reduce the jerk. b) An acceleration segment $[t_1, t_2]$ in which the cartesian acceleration is kept constant at its maximum. c) a third segment $[t_2, t_3]$ in which the acceleration decreases from its maximum to zero. d) a constant velocity segment is between t_3 and t_4. (e) to (g) are repeats of (a) to (c) but with negative a sign.

The trajectory is given in the cartesian coordinates by planning the motion along each of the cartesian axis X,Y and Z. The initial and final points are taken to be the same as those used in the PTP control. Figure 6.15 shows the acceleration, velocity and position of the end effector as it moves from P_1 to P_2.

As mentioned earlier, the control is performed at the actuators level, therefore, the inverse kinematic approach is used to transform the desired trajectory from the cartesian coordinates to the joints coordinates. The result of the transformation is shown in Figure 6.16.

The desired velocity and acceleration can be increased by reducing the time needed to perform the desired task. Figure 6.17 displays the results of two completely balanced arms, utilizing the counterweights approach and the spring counter balancing approach, as they move from P_1 to P_2 in T_{max}=4 sec. The figure shows that the spring balanced and weight balanced arms follow very closely the desired trajectory, and the traces in the figure coincide. Figure 6.18 shows that the spring balanced arm requires less torques from the joint actuators.

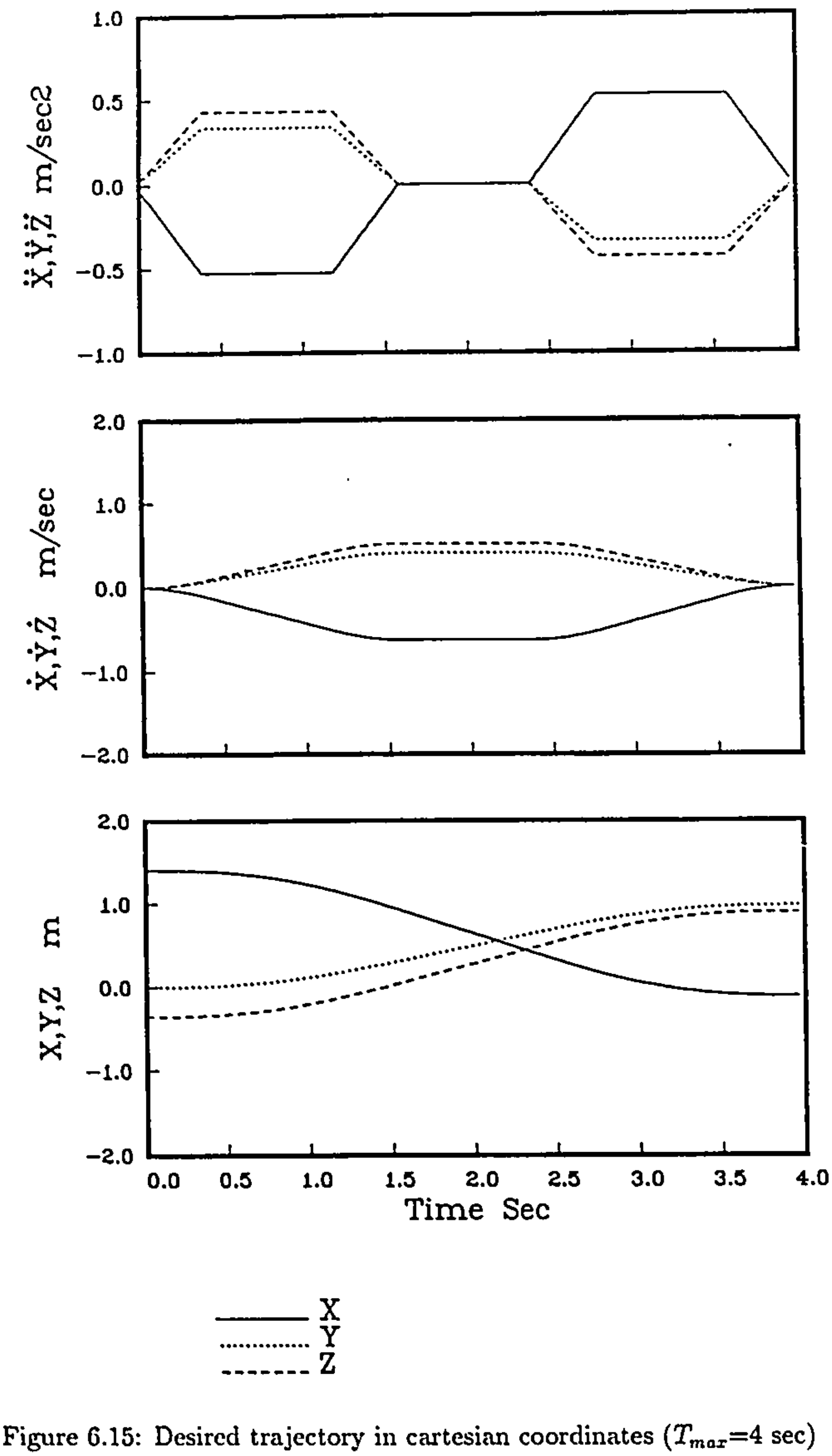

Figure 6.15: Desired trajectory in cartesian coordinates (T_{max}=4 sec)

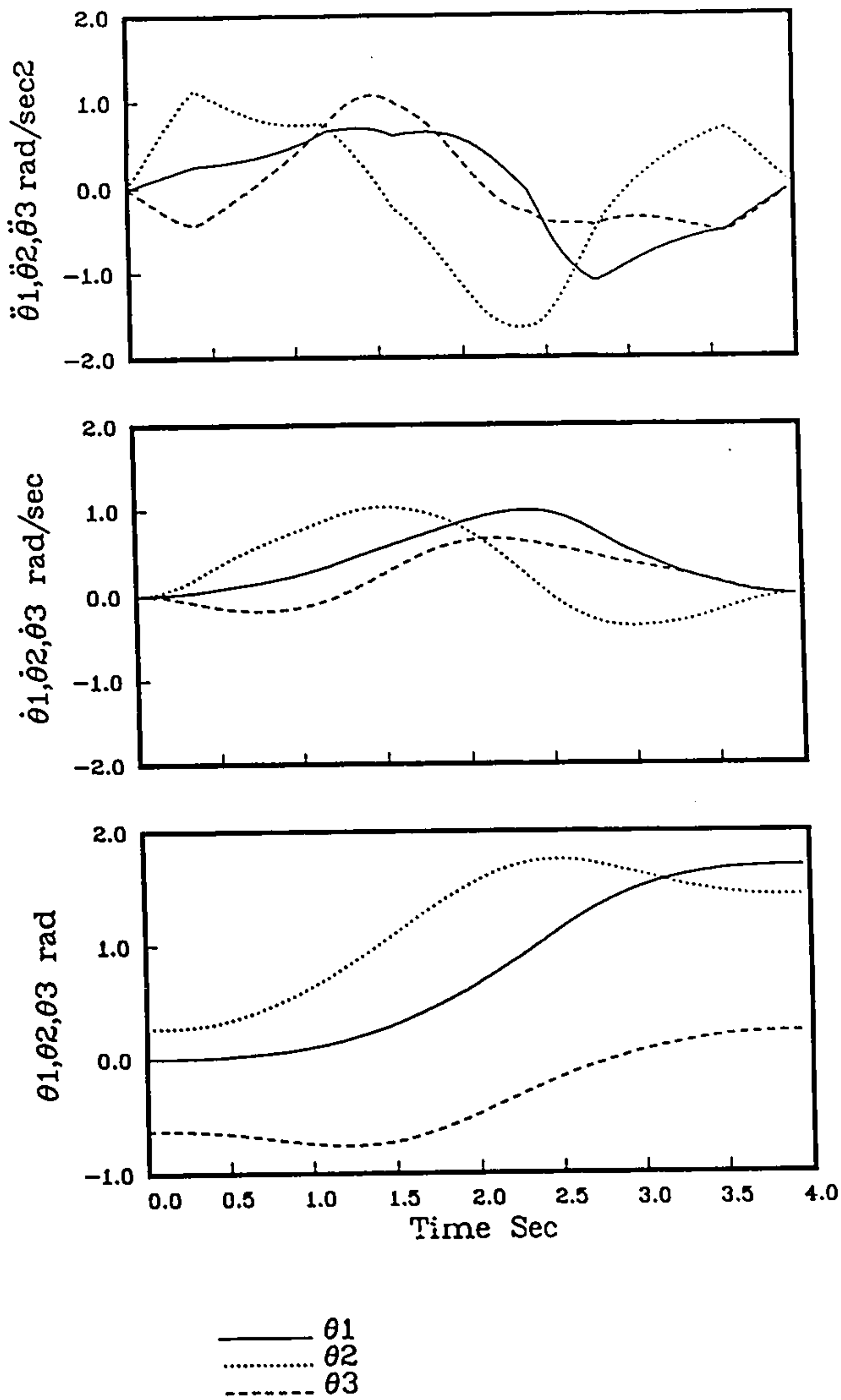

Figure 6.16: Desired trajectory in joint coordinates (T_{max}=4 sec)

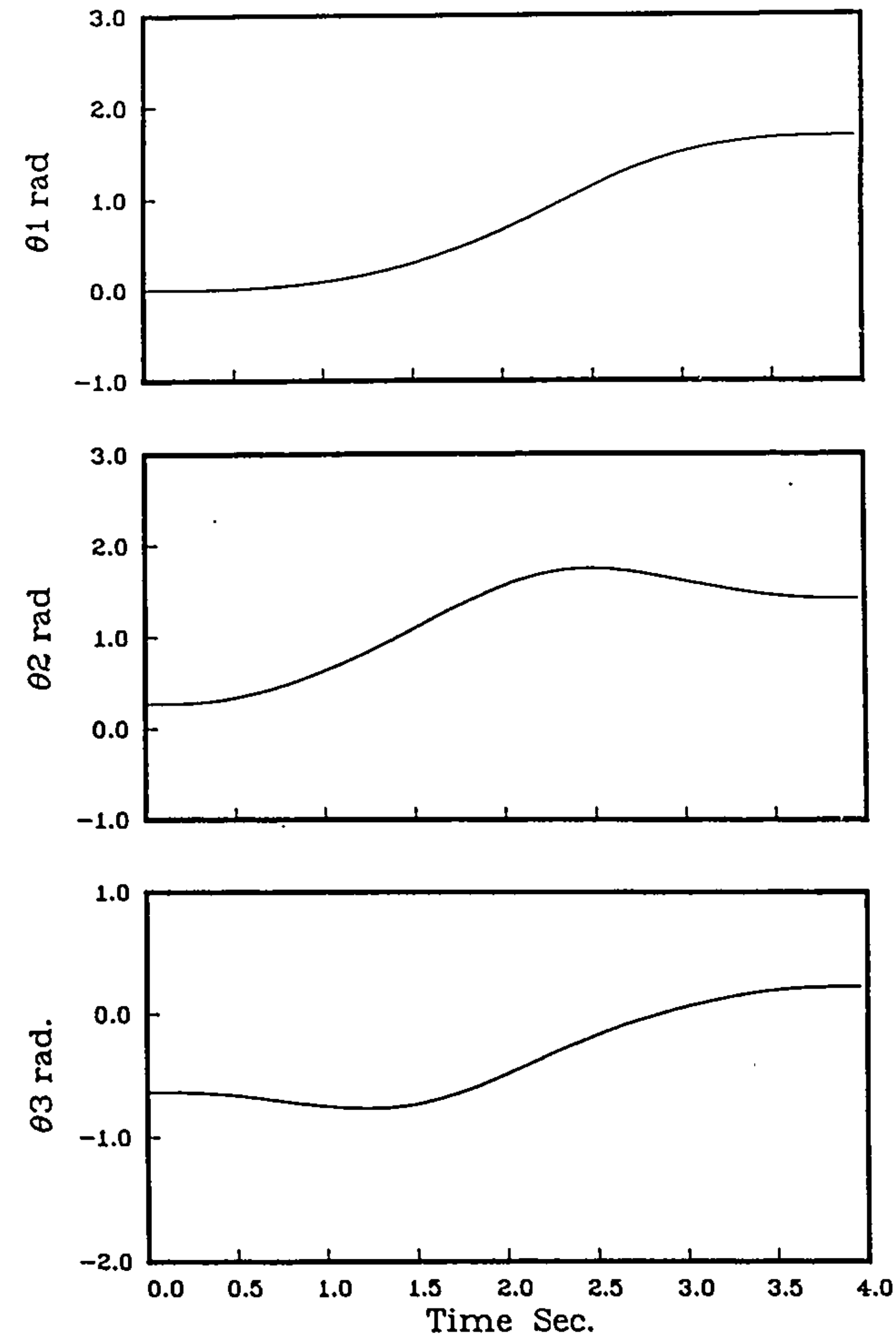

Figure 6.17: Tracking of desired trajectory in joint coordinates when: T_{max}=4 sec, Load=5 Kg and completely balanced

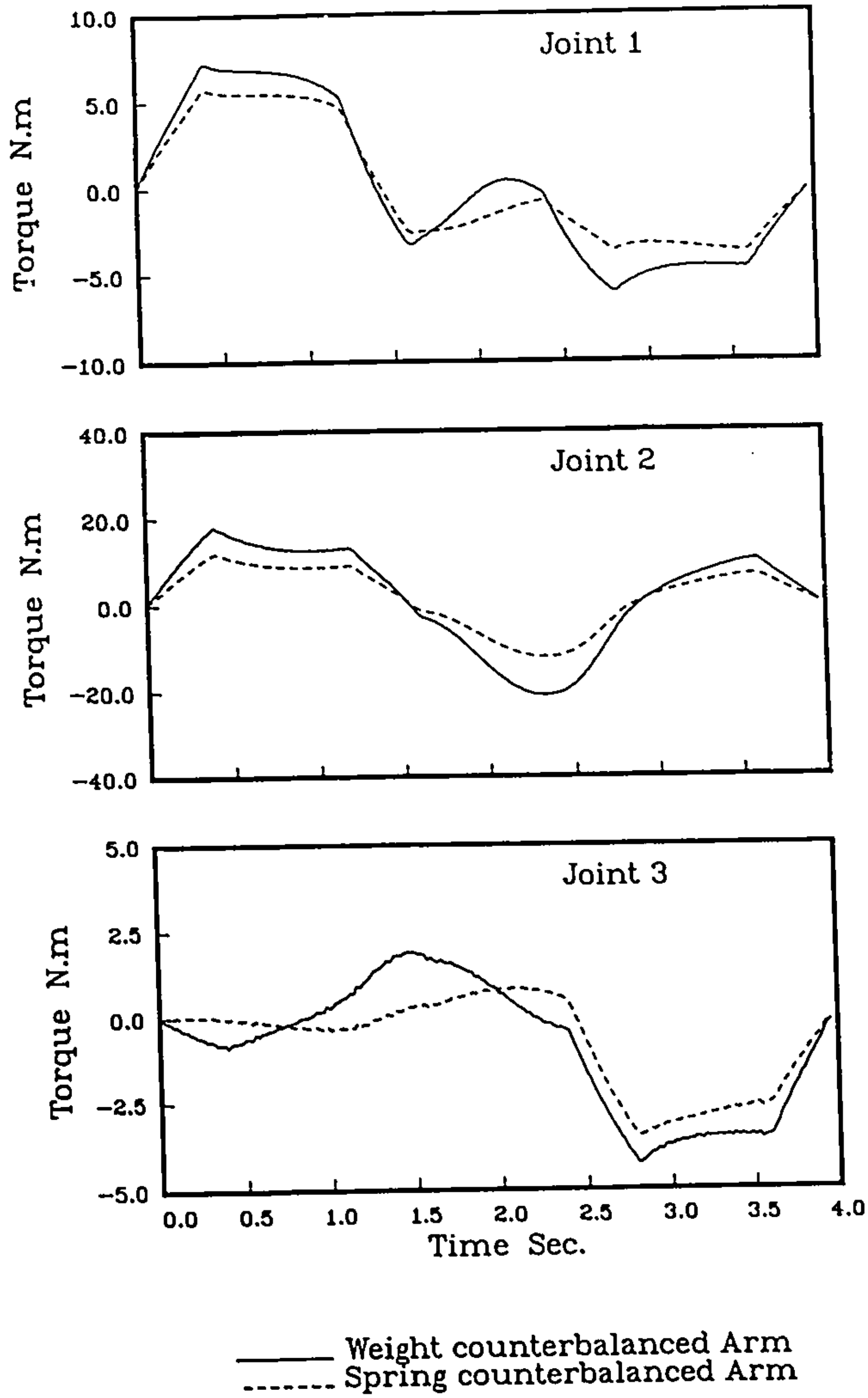

Figure 6.18: Actuators torques to track the desired trajectory when:

T_{max}=4 sec, Load=5 Kg and completely balanced

When the robot is asked to perform the desired trajectories at a higher speed, more torques from the motors are required. At a certain speed the motors cannot deliver any more torque and the desired trajectories can any longer be followed. Figure 6.19 and 6.20 show the trajectories resulting from reducing the maximum time T_{max} from $T_{max} = 4$ sec to $T_{max} = 1.4$ sec. This reduction resulted in an increase in the velocities and accelerations of the robot joints.

The effects of increasing the velocity and acceleration of each joint of the manipulator are shown is Figure 6.21. As can be seen from the figure, The actual trajectories deviate from the desired one. This deviation is much larger in the case of a weight counterbalanced arm.

The deviation of the actual trajectories from the desired ones is caused by the saturation of the joint actuators. As shown in Figure 6.22, the torques provided by the motors to perform the desired trajectories are much larger when the robot is moving at a high speed. Furthermore, the torques at joints 2 and 3 in the case of the weight counterbalanced arm reaches its maximum value, which means that no more torque can be provided by the actuator of this joint.

The actuators torques are affected by an increase in the speed of operation or by an increase in the payload. Wether there is an increase in the speed or an increase in the payload, the torques at each of three joint actuators are higher in the case of a weight counterbalanced arm as compared to a spring counterbalanced arm.

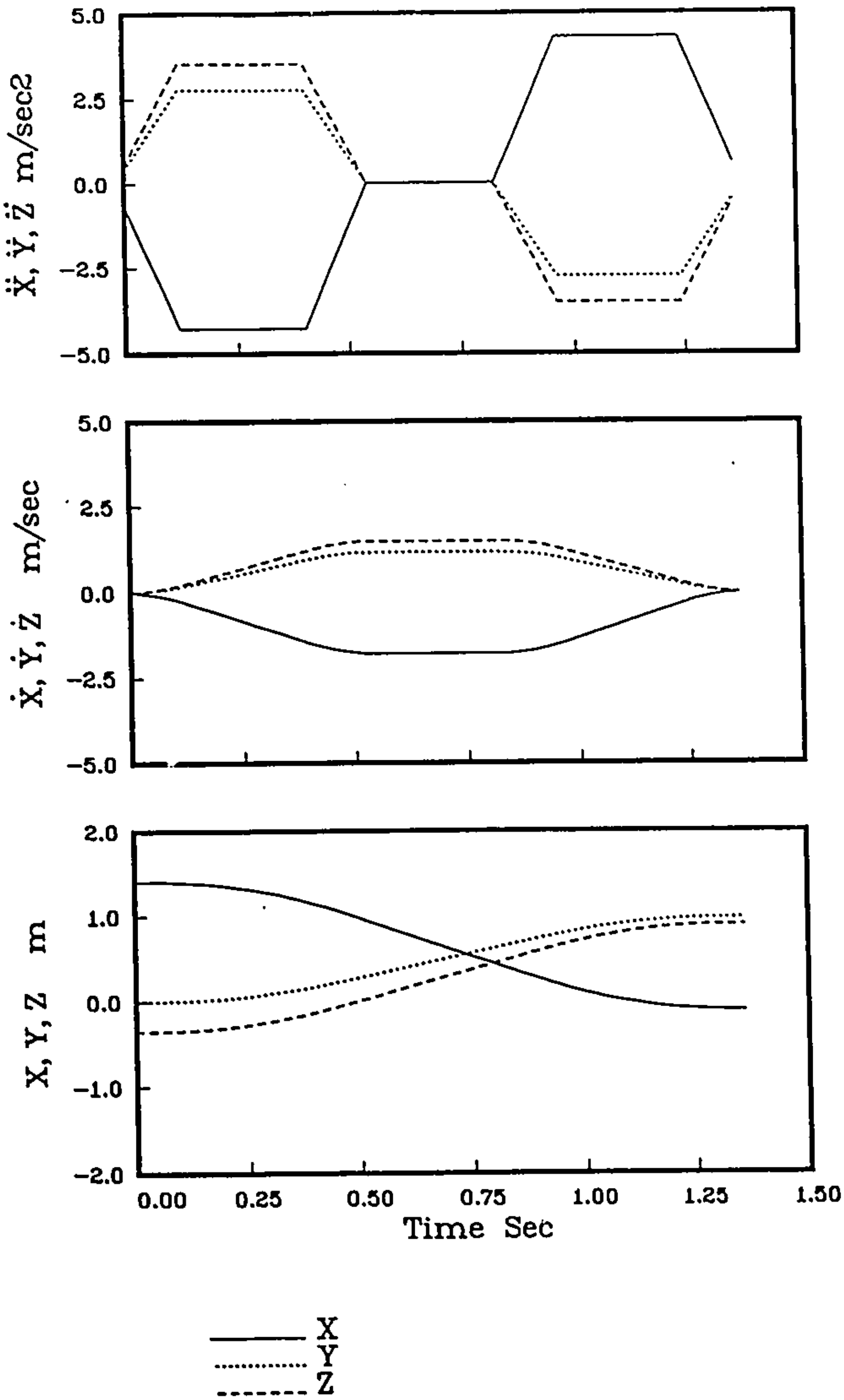

Figure 6.19: Desired trajectory in cartesian coordinates (T_{max}=1.4 sec)

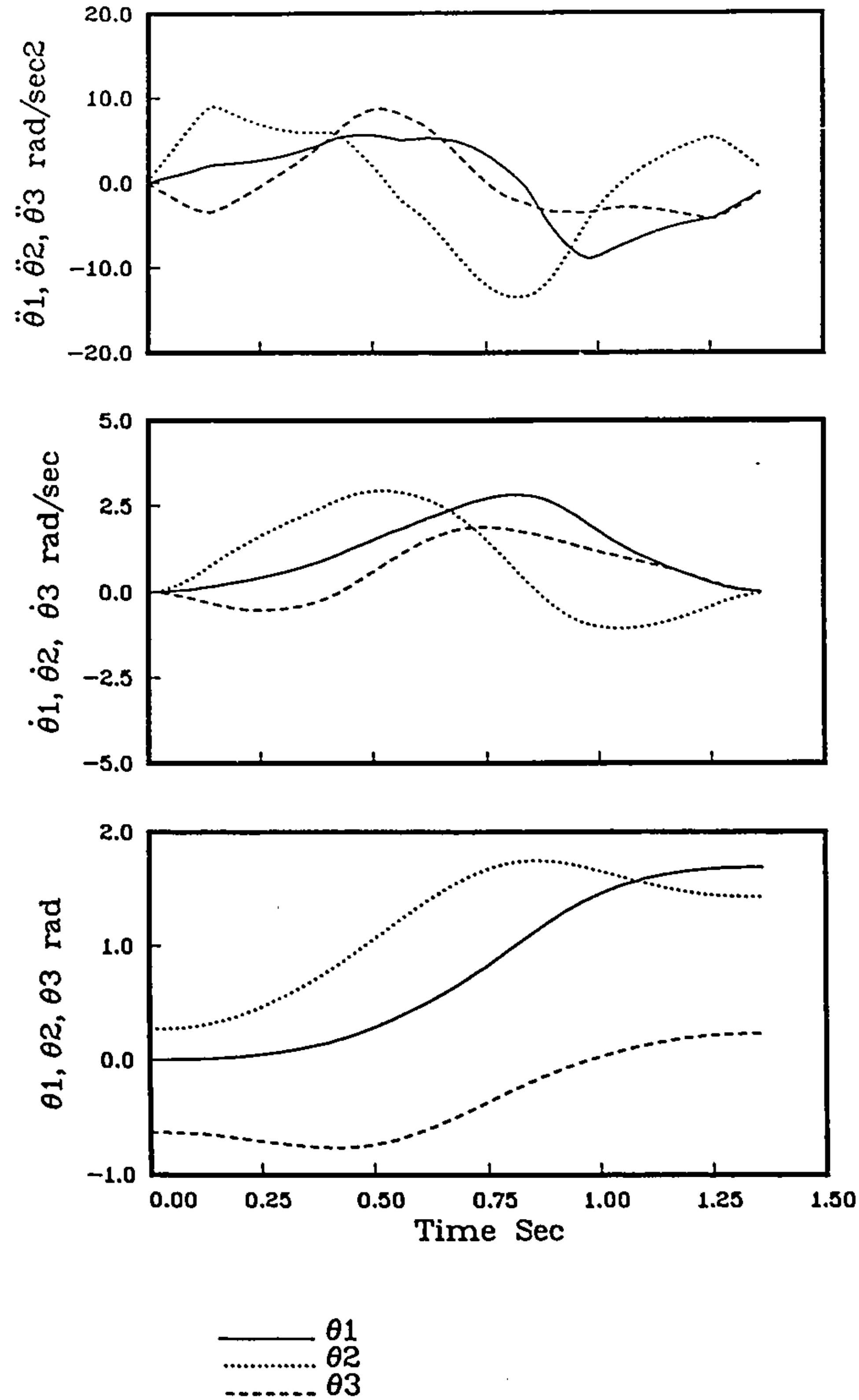

Figure 6.20: Desired trajectory in joint coordinates (T_{max}=1.4 sec)

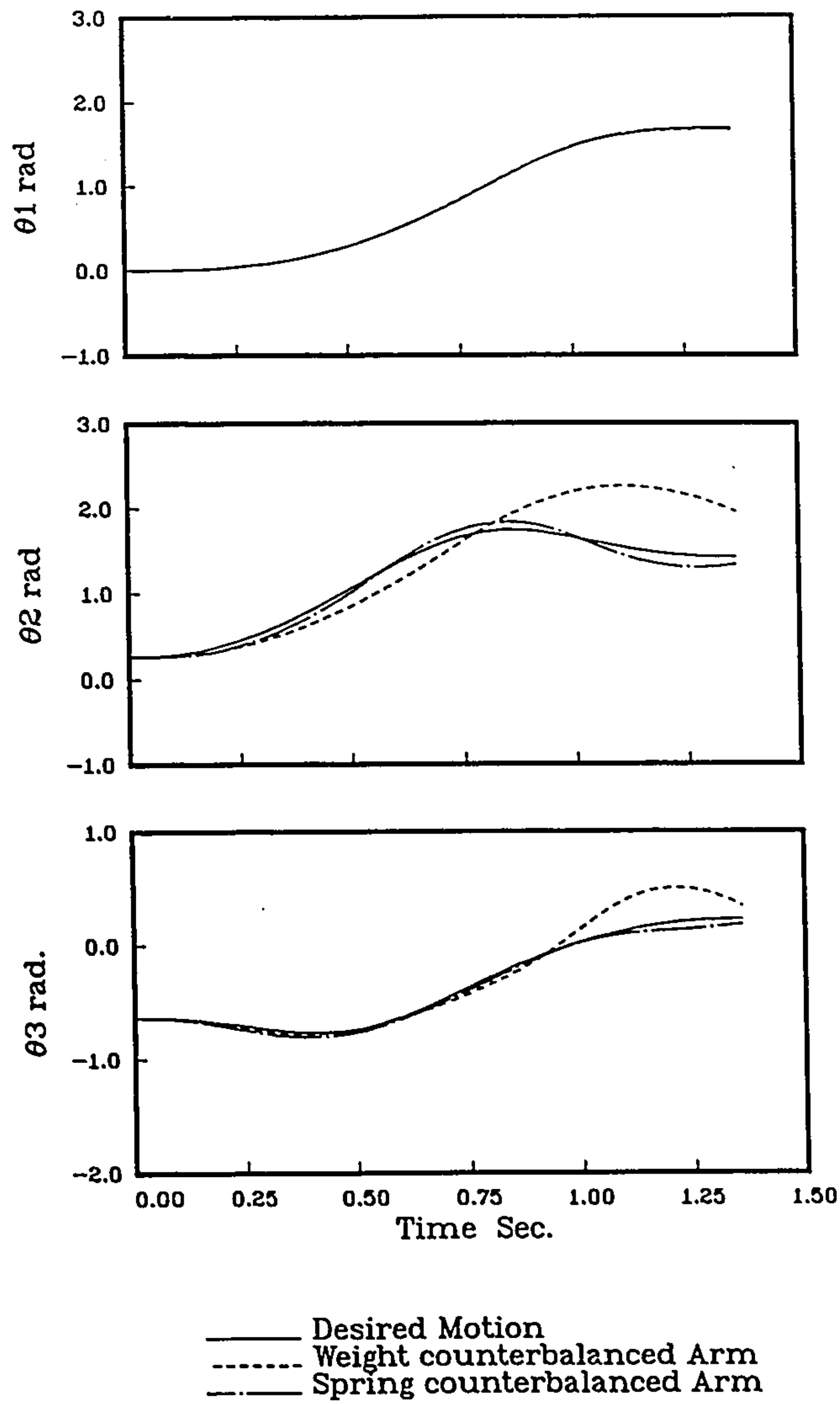

Figure 6.21: Tracking of desired trajectory in joint coordinates when:

T_{max}=1.4 sec, Load=5 Kg and completely balanced

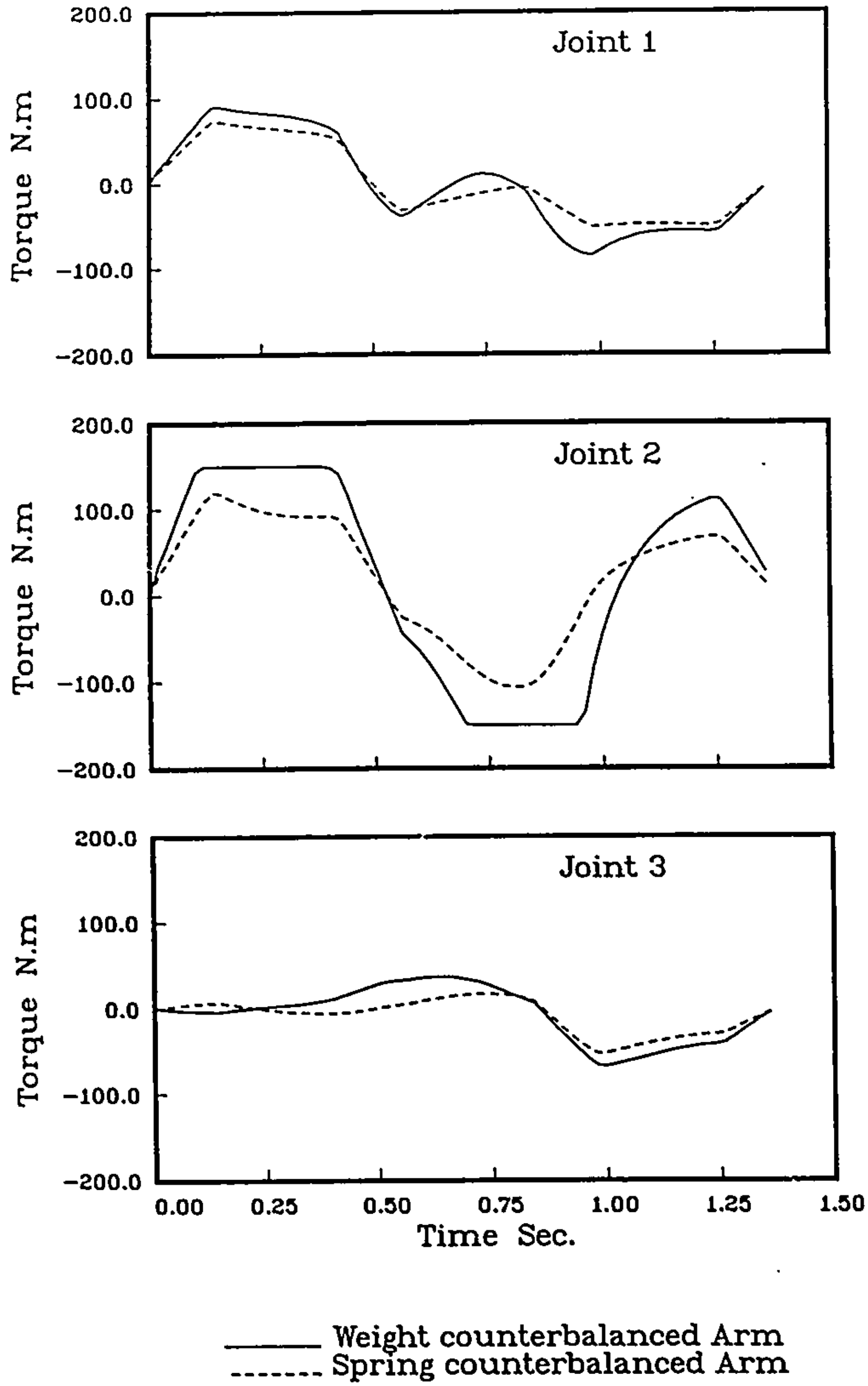

Figure 6.22: Actuators torques to track the desired trajectory when:

T_{max}=1.4 sec, Load=5 Kg and completely balanced

Figure 6.18 shows the actuators torques when the robot is carrying a load of 5 kg an performing the desired trajectories when $T_{max} = 4$ sec. The peak torques for the spring counterbalanced arm and for the weight counterbalanced arm are shown in table 6.3. Reducing the maximum time T_{max} from $T_{max} = 4$ sec to $T_{max} = 2$ sec leads to a higher torques for both spring counterbalanced and weight counterbalanced arms, as well as the advantage of using springs over the use of counterweights in terms of the difference between the peak torques. Figure 6.23 shows the actuators torques when $T_{max} = 2$ sec and a payload of 5 kg.

Increasing the payload from 5 kg to 10 kg results also in an increase in the peak torques for both arms but the amount of torque gained by using springs remain the same as when $T_{max} = 4$ sec and a payload of 5 kg. Table 6.3 presents a summary of the results shown in Figures 6.18, 6.23 and 6.24.

The results shown ir table 6.3 are for the case when the arms are completely balanced for any load, ie. only dynamic torques are present.

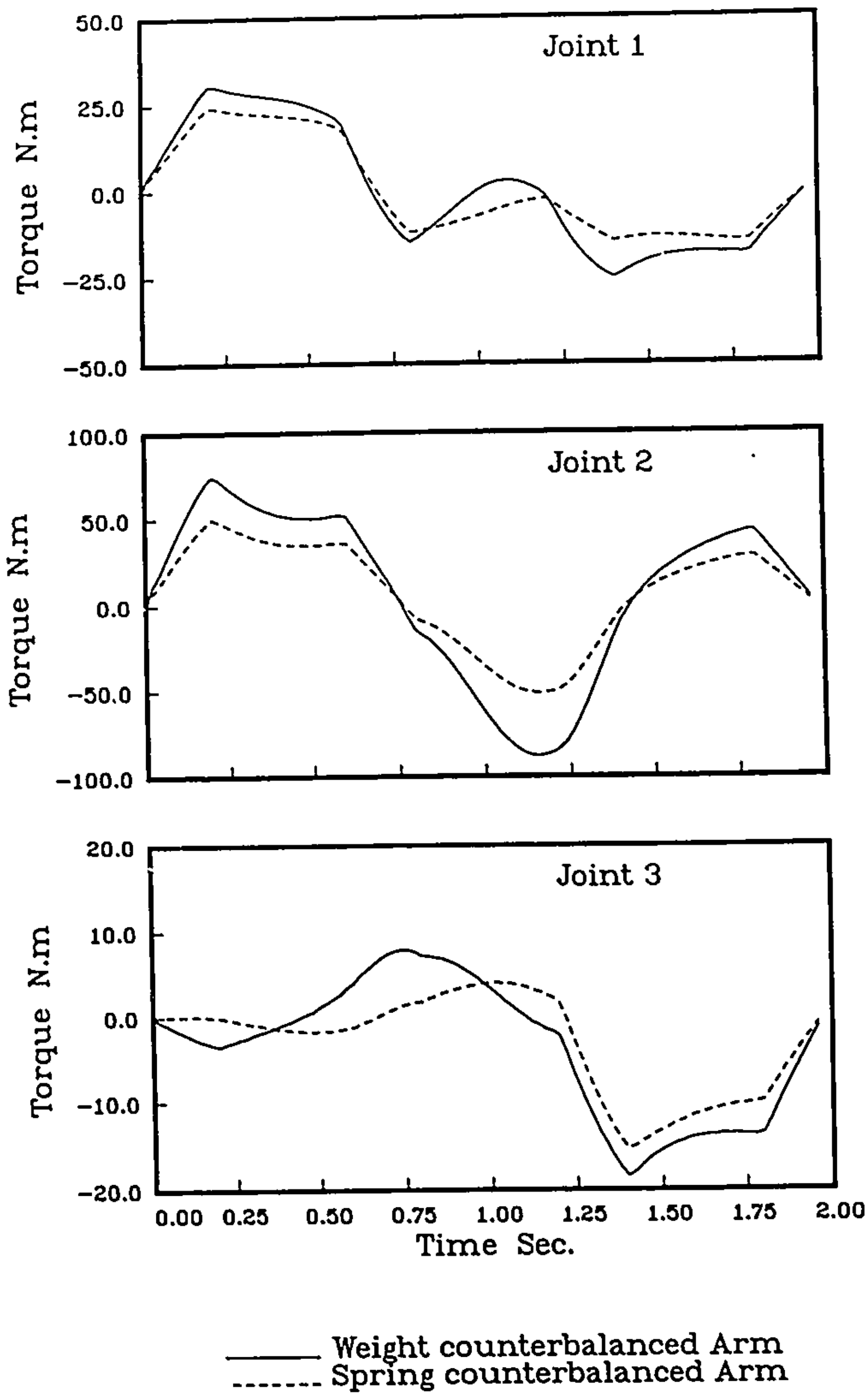

Figure 6.23: Actuators torques to track the desired trajectory when:

T_{max}=2 sec, Load=5 kg and completely balanced

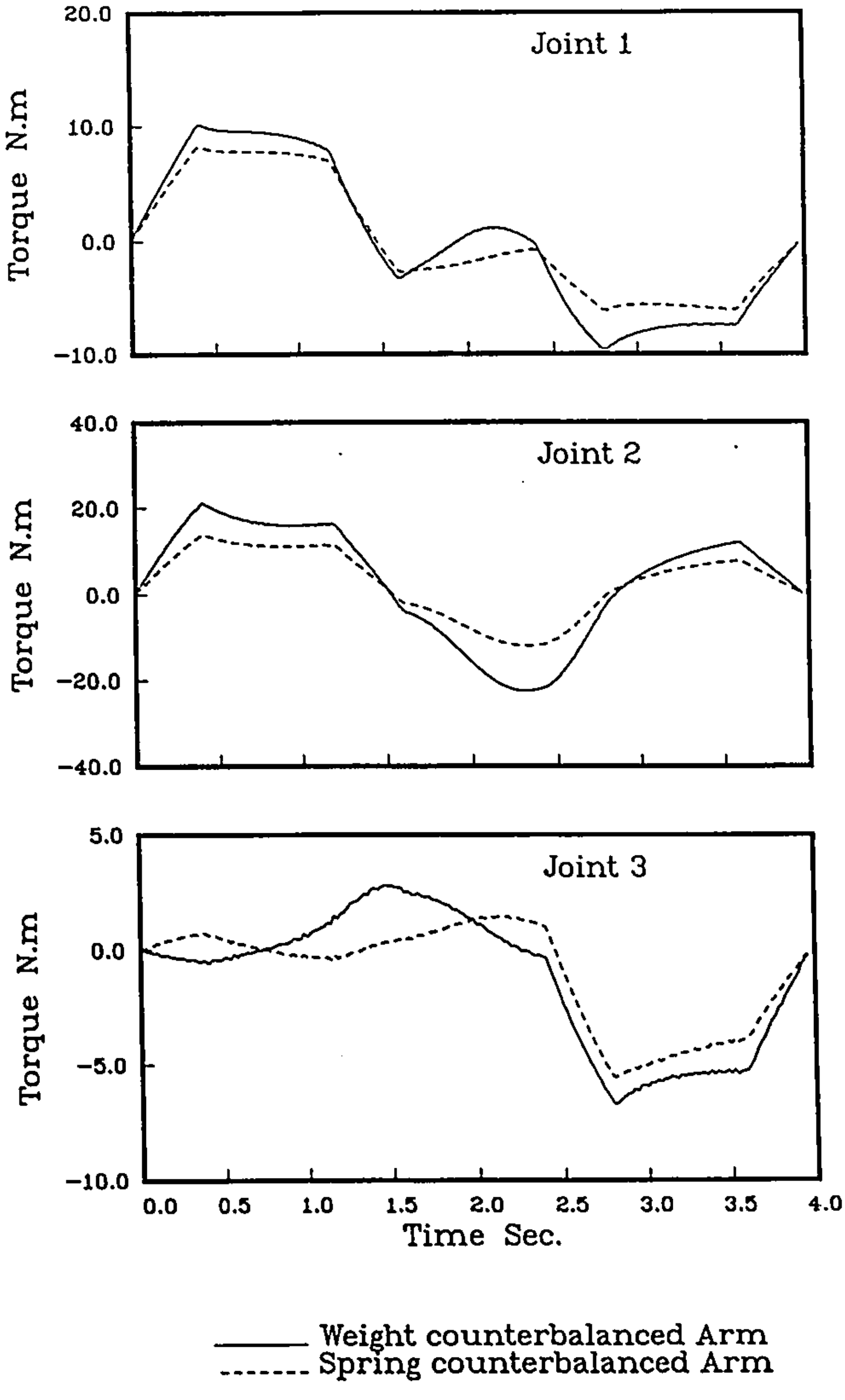

Figure 6.24: Actuators torques to track the desired trajectory when:

T_{max}=4 sec, Load=10 kg and completely balanced

	Load=5 kg $T_{max} = 4$ sec			Load=5 kg $T_{max} = 2$ sec			Load=10 kg $T_{max} = 4$ sec		
	joint1	joint2	joint3	joint1	joint2	joint3	joint1	joint2	joint3
Peak torque weight balanced arm τ_{pw} (N.m)	7.2	18.1	4.3	30.3	88.0	18.5	10.1	20.7	6.7
Peak torque spring balanced arm τ_{ps} (N.m)	5.8	12.0	3.5	24.0	51.4	14.8	8.2	13.9	5.5
$\frac{\tau_{pw}-\tau_{ps}}{\tau_{pw}} \times 100$	19	33	18	21	41	20	19	33	18

Table 6.3: Comparison of the required torques for completely balanced arms

Figure 6.25 and 6.26 show the torques of the joint actuators when the arms are balanced for 5 kg, and carrying a load of 10 kg and 15 kg, respectively. At $t = 0$ and $t = T_{max}$, the torques provided dy the actuators are equal to the amount of unbalance. The added torque due to the gravity effect shifted the curves of joint 2 and joint 3 upward. Since the gravity effect is the same for both spring counterbalanced and weight counterbalanced arms, the advantage gained by using spring is smaller compared to the case of completely balanced arms. Table 6.4 presents the results of the peak torque at each joint for the spring counterbalanced and weight counterbalanced arms when a torque due to gravity is present with the dynamic torque. The advantage gained by using springs is larger when the robot is moving at a higher speed and carrying lighter load.

	Load=10 kg $T_{max} = 1.6$ sec			Load=15 kg $T_{max} = 2$ sec		
	joint 1	joint 2	joint 3	joint 1	joint 2	joint 3
Peak torque weight balanced arm (τ_{pw} N.m)	65.1	163.9	60.6	51.5	156.3	95.8
Peak torque spring balanced arm (τ_{ps} N.m)	55.2	124.2	53.3	45.2	131.3	93.9
$\frac{\tau_{pw}-\tau_{ps}}{\tau_{pw}} \times 100$	15	24	12	12	16	3

Table 6.4: Comparison of the required torques when the arms are balanced for 5 kg

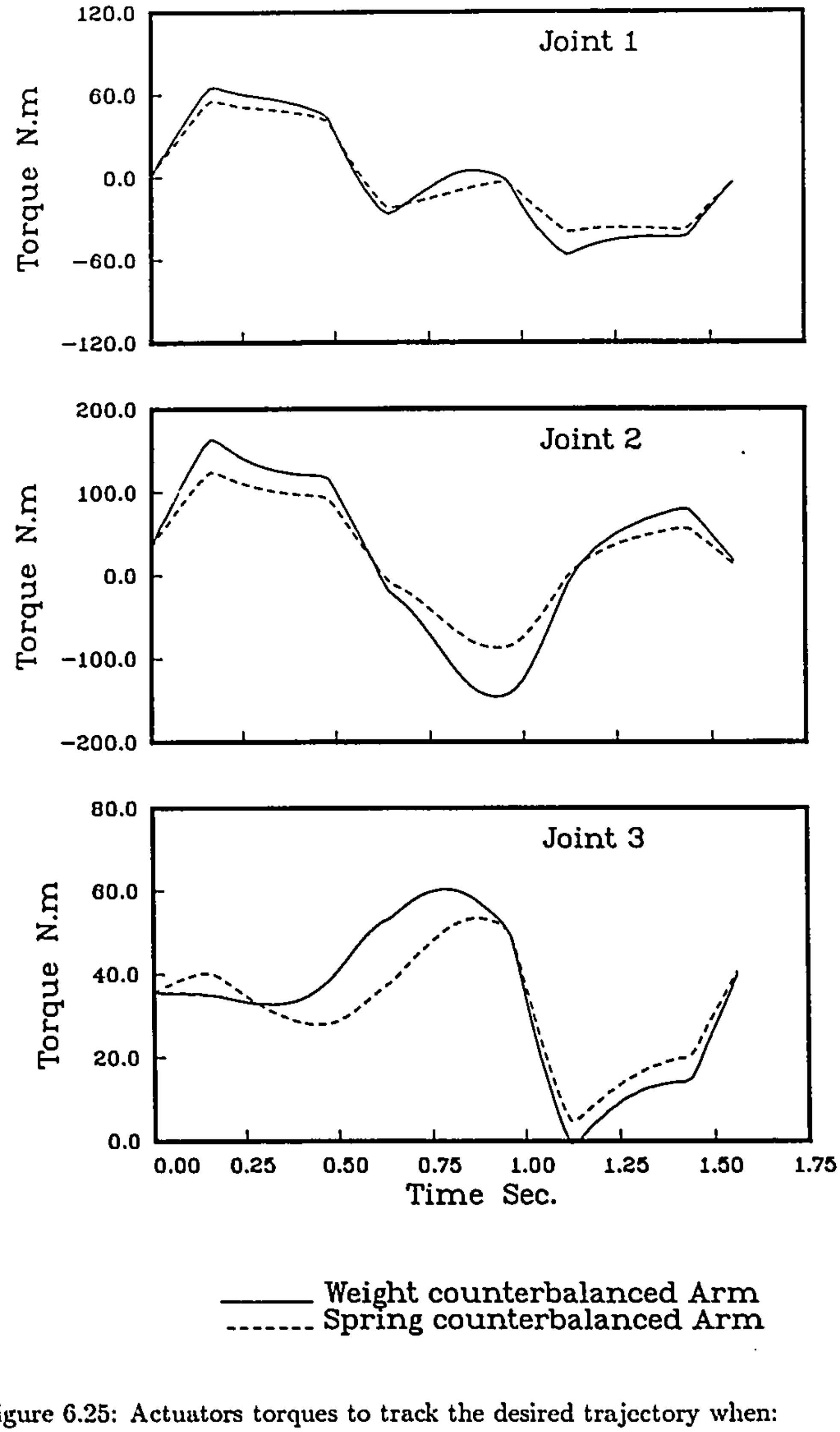

Figure 6.25: Actuators torques to track the desired trajectory when:

T_{max}=1.6 sec, Load=10 kg and balanced for 5 kg

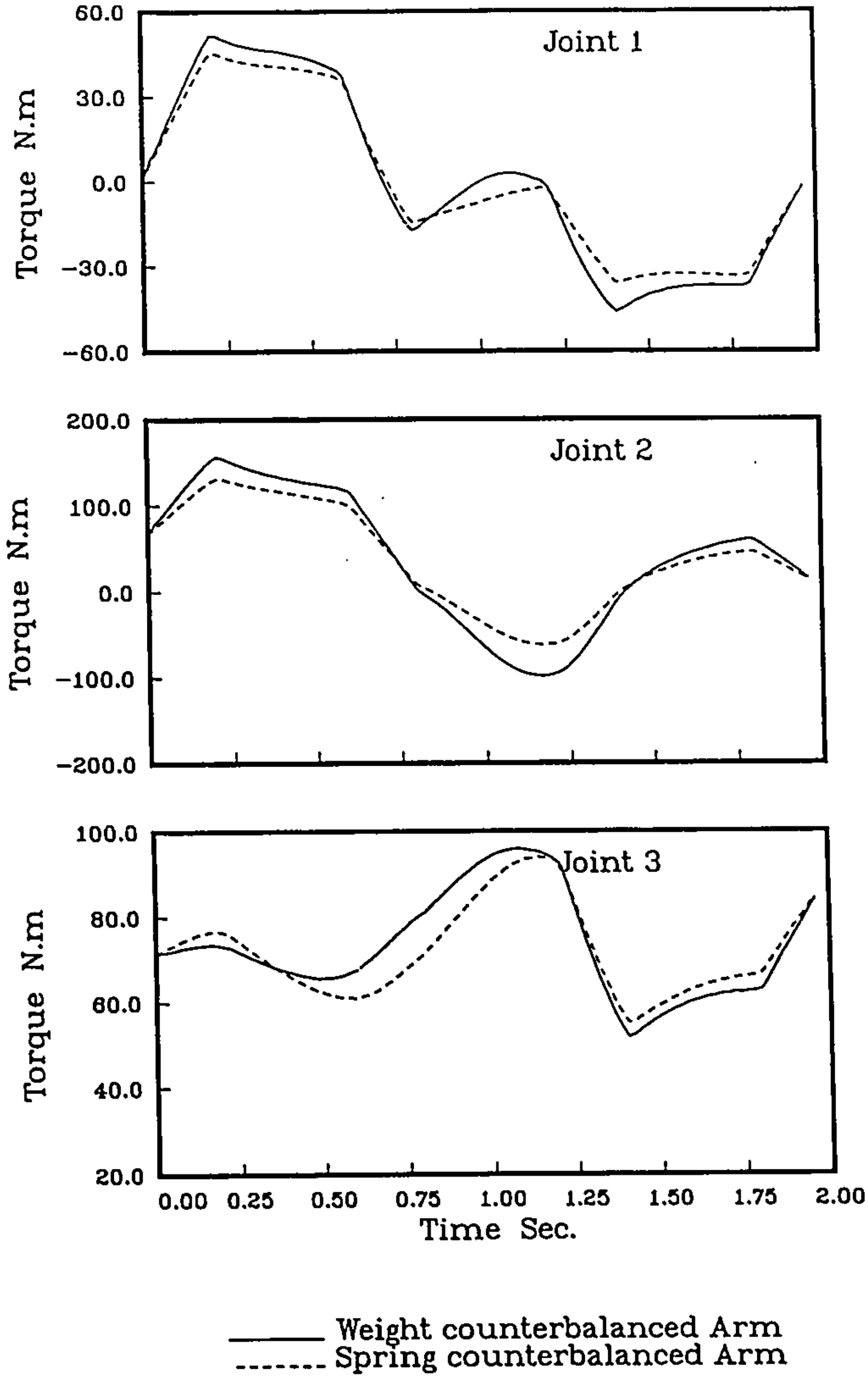

Figure 6.26: Actuators torques to track the desired trajectory when:

T_{max}=2 sec, Load=15 kg and balanced for 5 kg

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

This thesis presents the analysis and control of a spring balanced robot arm. The analysis shows that balancing the gravity effect on robot arms results in enhancing the performance and in increasing the load carrying capabilities of these robots.

Using a counterweight to balance the links of robot manipulators counteracts the effects of gravity and leads to perfect balancing for a specific payload. This method was found satisfactory when the robot is moving at low speed. At high speed, however, the effect of the inertia of the counterweight on the dynamics of the robot become very significant.

Several mechanisms for balancing a one link manipulator have been investigated. The first mechanism uses a spring-gear mechanisms to generate the nonlinear function required to balance the gravity torque. The expression of the balancing torque was found to be position dependent and cannot give complete balancing as the configuration of the link changes. In order to be able to select the parameters of the mechanism that gives the minimum amount of unbalance, an optimization technique was used.

The second mechanism consists of a linear spring with one end fixed to the link to be balanced and the other end is fixed to ground. This arrangement gives an equilibrium equation which depends on the position of the link. With an appropriate choice of the anchor locations of the spring, the spring rate K, and by using a zero free length spring, balancing is achieved regardless of the position of the link. For balancing the links of a three revolute joint manipulator, each link is equipped with a spring mechanism such as the one used to balance a one link manipulator.

In order to study the effect of balancing the gravity by using spring mechanisms, and the advantages of eliminating the counterweights, a dynamic analysis of a spring balanced and a weight balanced three revolute arm has been presented. The performance of the two arms while carrying out a given control task has also been investigated.

Two types of controller were used. The first is an individual joint controller to perform a PTP control task, and the second is the computed torque technique used to perform a trajectory tracking task. It has been found that in the PTP control task, the speed of the weight counterbalanced arm is less than the spring balanced arm. Furthermore, the spring balanced arm reaches steady state faster than the weight counterbalanced arm.

The trajectory tracking task was conducted to evaluate the required input torques for the spring counterbalanced and the weight counterbalanced arms. the inertia torques of the spring balanced arm were smaller than those of the weight balanced arm. As the speed increases, the difference in torque between the two arms becomes larger especially in the accelerating and decelerating zones. The amount of torque available from the actuators of the spring balanced arm can be used to carry heavier loads or to move with a higher speed.

When the arms are balanced for the carried load, the torques due to gravity are absent and only inertia torques are present. For arms balanced for a specific load, torques due to the amount of unbalance are present and add up to the inertia torques.

This work deals only with the balancing of a specific payload. In order to adapt the balancing mechanisms to a change in the payload, a feedback system can be implemented to sense the amount of load carried and adjust the mechanism parameters with respect to the sensed information.

REFERENCES

[1]: W. A. Harmennig, " Static Mass Balancing With a Torsion Spring and Four Bar Linkage ". ASME paper. No 74 DET 29.

[2]: Mathew, G.K, and Tesar, D. " Synthesis of Spring Parameters To Satisfy Specified Energy Levels in Planar Mechanisms ". ASME paper No 76-DET 44.

[3]: Mathew, G.K, and Tesar, D. " Synthesis of Spring Parameters To Balance General Forcing Functions in Planar Mechanisms". ASME paper No 76-DET 45.

[4]: R. H. Nathan, " A Constant Force Generation Mechanism ". Trans. ASME paper No.84-DET 188 vol.107 pp 508-512, 1985.

[5]: Lu. Zhen, " The Analytical Synthesis of Spring Element in Spatial Mechanisms ". Mech. Mach. Theory vol 23, No.1, pp 47-54,1988.

[6]: Neville Hogan. " Mechanical Impedance Control in Assistive Devices and Manipulators " Proceeding of Joint Automatic Control.

[7]: Yu.F. Gvozdv,"New Balancing Mechanisms For The Links of Robot arms". Vestnik Mashinostroeniya. Vol. 63, Issue 8, 1983, pp 13-15.

[8]: P. Minotti, P. Pracht," Ressorts et Mechanismes: Une solution aux problemes d'equilibrage", J. Mech. Mach. Theory, Vol. 23, No. 2, pp. 157-168, 1988.

[9]: H. Hilpert, " Weight balancing of precision Mechanical Instruments", J. Mechanism, Vol. 3, pp. 289-302, Pergamon press, 1968.

[10]: G. G. Lowen, R. S. Berkof, " Survey of Investigations into the balancing of linkages", J. Mechanisms, Vol. 3, pp. 221-231, 1968.

[11]: Tokuji Okada, " Optimization of Mechanisms for Force Generation By Using Pulleys and Spring", Electrotechnical Laboratory 1-1-4 Umezono, Sakura-mura 305 Japan. Journal of robotic research PP. 245 - 252.

[12]: Vukobratovic, M. Kircanski, "A method for Optimal Synthesis of Manipulation Robot Trajectories". Trans ASME, vol 104 1982, pp.188-193.

[13]: Arimoto S., and Miyazaki F., "Stability and Rubustness of PID Feedback control for Robot Manipulators of Sensory Capability". Robotics Research, The first Int. Sym. pp 784-793. Edited by Michael Brady and Richard Paul, 1984.

[14]: Tsuneo Yshikawa, " Analysis and Design of Articulated Robot Arms from The view point of Dynamic Manipulability". Robotics Research, The third International Symposium pp 273-279. Edited by Faugeras and Geoegcs Giralt, 1986.

[15]: George N. Sandor and Arthur .G.Erdman " Advanced Mechanism Design: Analysis and Synthesis. Volume 2.", p. 377.

[16]: Neville Hogan. " Impedence Control: An approach to Manipulation. Part I, Theory". ASME Journal of Dynamic, Measurements, and Control. Vol. 107, March 1985, pp. 1-7.

[17]: Neville Hogan. " Impedence Control: An approach to Manipulation. Part II, Implementations". ASME Journal of Dynamic, Measurement, and Control. Vol. 107, March 1985, pp. 8-16.

[18]: Neville Hogan. " Impedence Control: An approach to Manipulation. Part II, Applications". ASME Journal of Dynamic, Measurement, and

Control. Vol. 107, March 1985, pp. 17-24.

[19]: Roessler J.," A Decentralized Hierarchical Control Concept for Large-Scale Systems", Proc. of the II IFAC Symp. on Large-Scale Systems, pp. 171-179, Toulouse,1980.

[20]: Paul P. R.," Robot Manipulators: Mathematics, Programming and Control, The MIT. Press, Cambridge 1985.

[21]: Vukobratović. K.M, Stokić M.D, "Scientific Fundamentals of Robotics,2, Control of Manipulation Robots: Theory and Application, Monograph, Springer-Verlag, Brlin, 1982.

[22]: Luh Y.S.J.," An Anatomy of Industrial Robots and Their Controls" , TEEE Trans. on Automatic Control, Vol. AC-28, No 2,pp. 133-152, February, 1983.

[23]: Vukobratovic K.M., Stokic M.D, " Is Dynamic Control Needed in Robotic Systems, and If so to What extent ?", International Journal of Robotic Research, vol. 102, June, 1982.

[24]: Takegati M., Arimoto S.," A New Feedback Method for Dynamic Control of Manipulators". Trans. of ASME, Journal of Dynamic Systems, Measurement, and Control, June, pp 119-125, vol. 102, 1981.

[25]: Young K.K.D.," Controller Design for Manipulator Using Theory of Variable Structure Systems", IEEE Trans. On Systems, Man and Cybrenetics, vol. SMC-8,1978.

[26]: Davison J.E.,"The Robust Decentralized Control of a General Servomechanism Problem", IEEE Trans. On Automatic Control AC-21, pp. 14-24, 1976.

[27]: Cvetkovic V.,Vukobratovic K.M.," One Robot Dynamic Control Algorithm for Manipulation Systems", The Int. Journal of Robotic Research, vol.1, No 4, pp.15-28, 1982.

[28]: Vukobratović. K.M, Stokić M.D, "Scientific Fundamentals of Robotics,5,

Non-Adaptive and Adaptive Control of Manipulation Robots", Springer-Verlag, Brlin, 1985.

[29] S. D. Eppinger, D. N. O'Connor, W. P. Seering, D. N. Wormley, "Modelling and experimental evaluation of asymmetric Pantograph Dynamics". ASME Trans., J. of Dynamic systems, Measurement, and Controls, Vol. 110, pp. 168-173, June 1988.

[30] D. C. H. Young, S. W. Tzeng, " Simplification and Linearization of manipulator Dynamics by the design of the inertia Distribution." The international Journal of Robotics Research, Vol. 5,No. 3, pp. 120-128, Fall 1986.

[31] Haruhiko Asada, Kamal Youssef- Toumi, " Analysis and Design of a Direct-Drive Arm with a Five-Bar-link Parallel Drive Mechanism." Journal of Dynamic systems, Measurements, and Control, Vol. 106, pp. 225-230, september 1984.

[32] Pradeep K. Khosla, Tkeo Kanade, " Experimental evaluation of Nonlinear feedback and feedforward control schemes for Manipulators". The international Journal of Robotics Research, Vol. 7, pp. 18-28, February 1988.

[33] Philippe Coiffet, " Robot technology: Modelling and control", Prentice - Hall Inc. 1983; Kogan page Ltd.

[34] B. Benhabib, R. Ristic, J. Sit, J. Wiercienski, A. A. Goldenberg, " An Elector-optical Tactile Sensor", Robotic and Automation Laboratory, Dept. of Mech. Eng., U. of Toronto, Third canadian Universities conference on CAD/CAM, pp. 84-93, July 1987, Ottawa University.

[35] D. J. Wilde, M. Avriel, M. J. Rijckaert, " Optimization and Design", pp. 46-49, Prentice - hall, 1973.

Appendix A

Responses of the systems as they move from P_2 to P_1

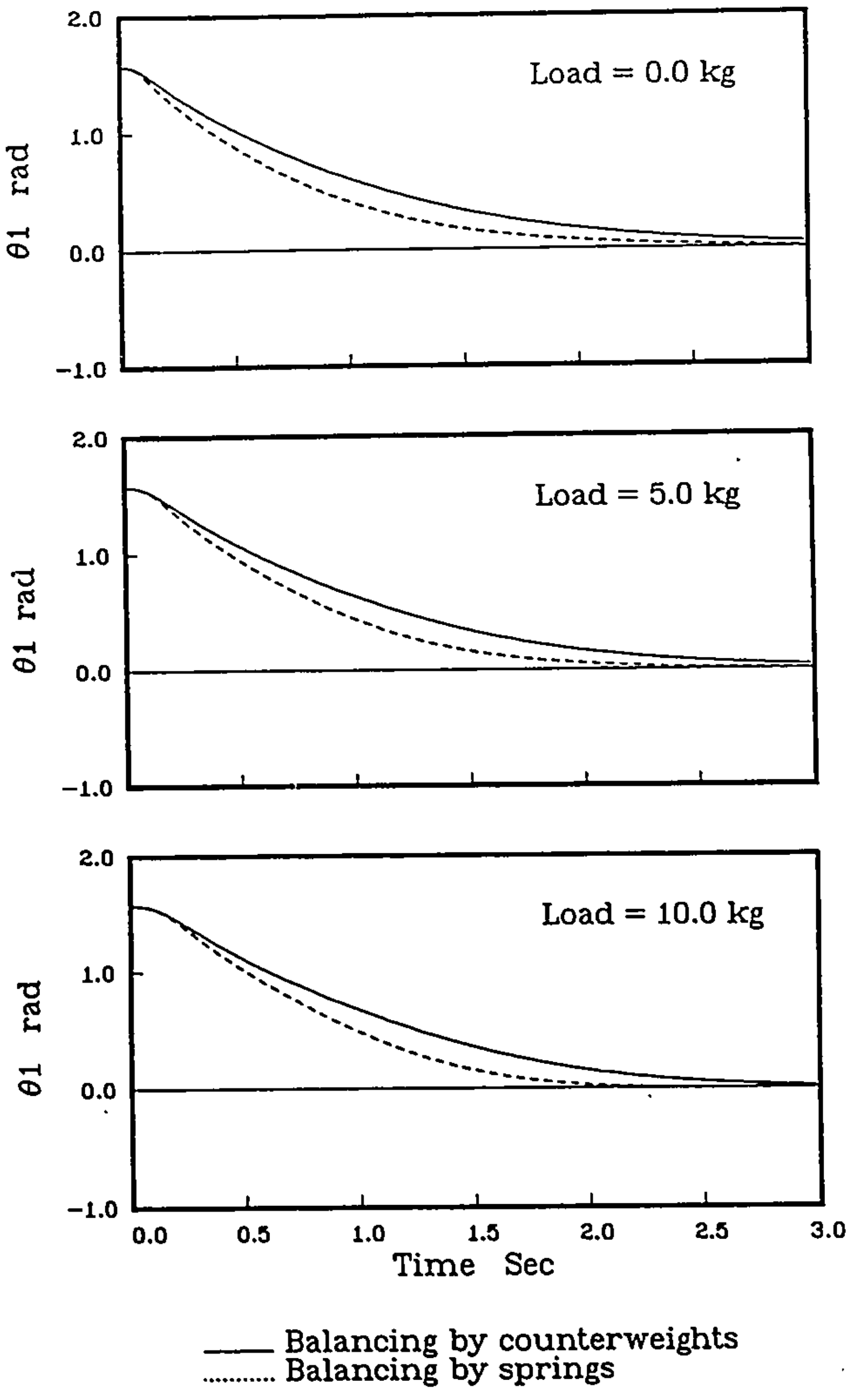

Figure A.1: Position of joint 1, systems balanced for 5 kg

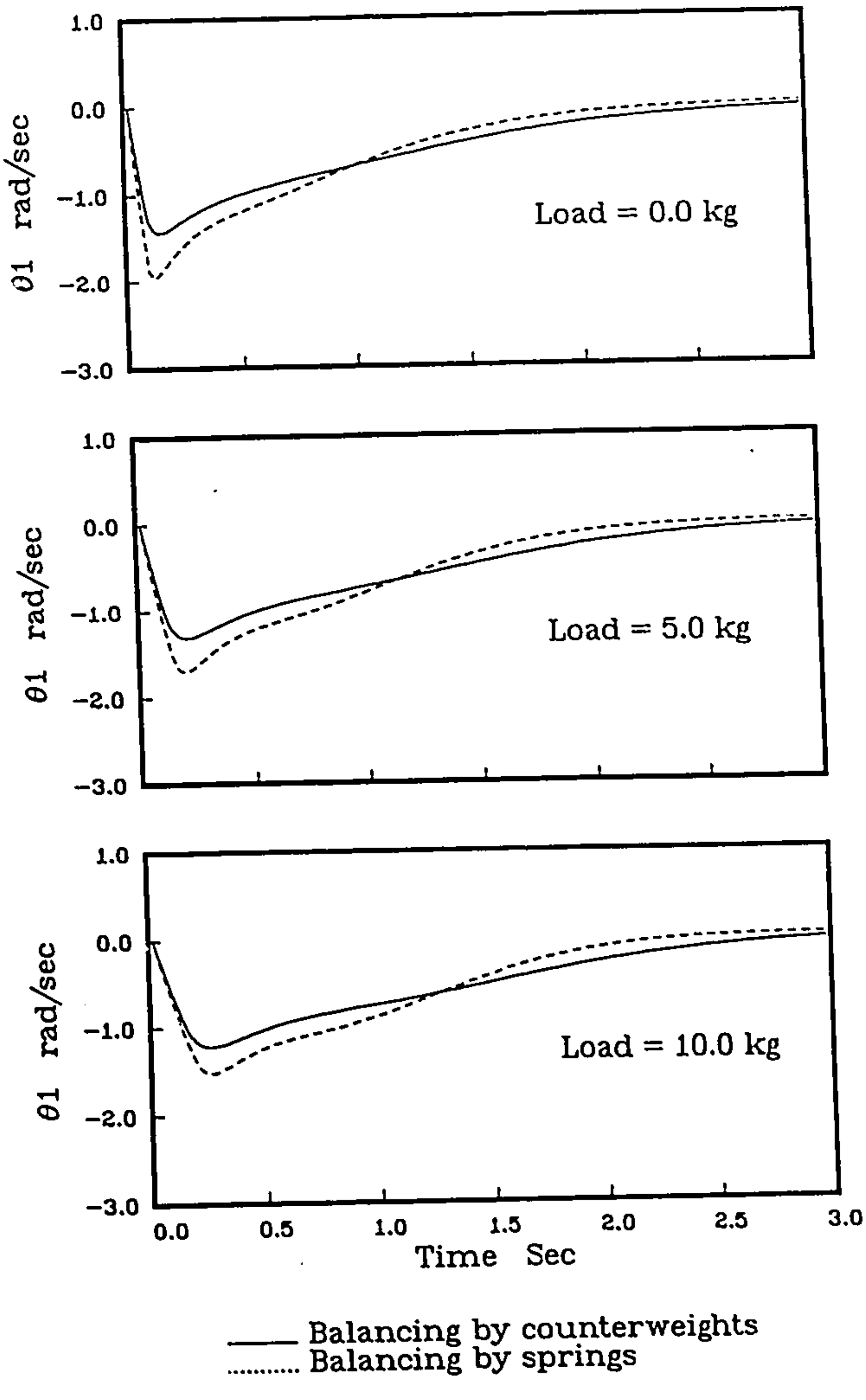

Figure A.2: Velocity of joint 1, systems balanced for 5 kg

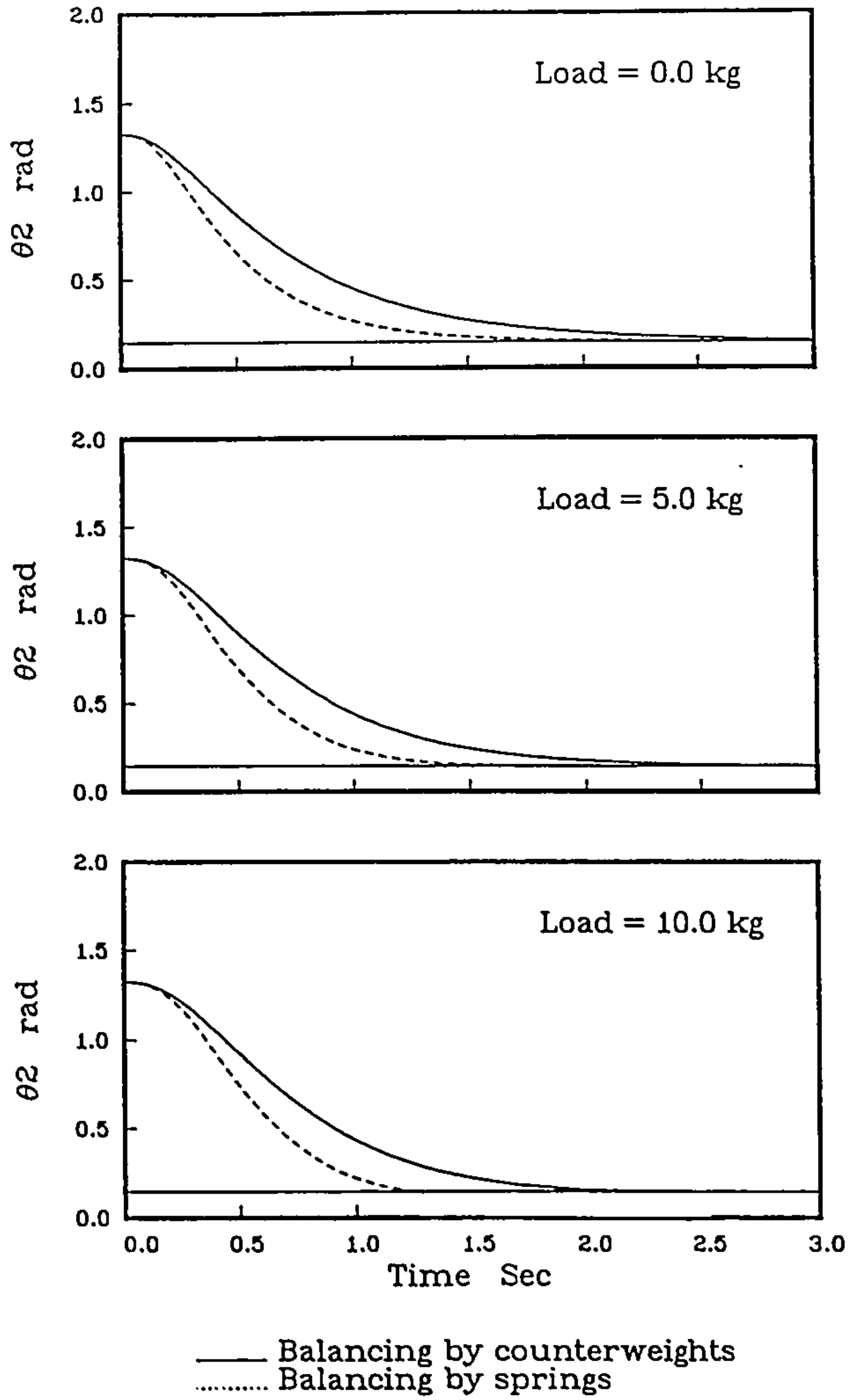

Figure A.3: **Position of joint 2, systems balanced for 5 kg**

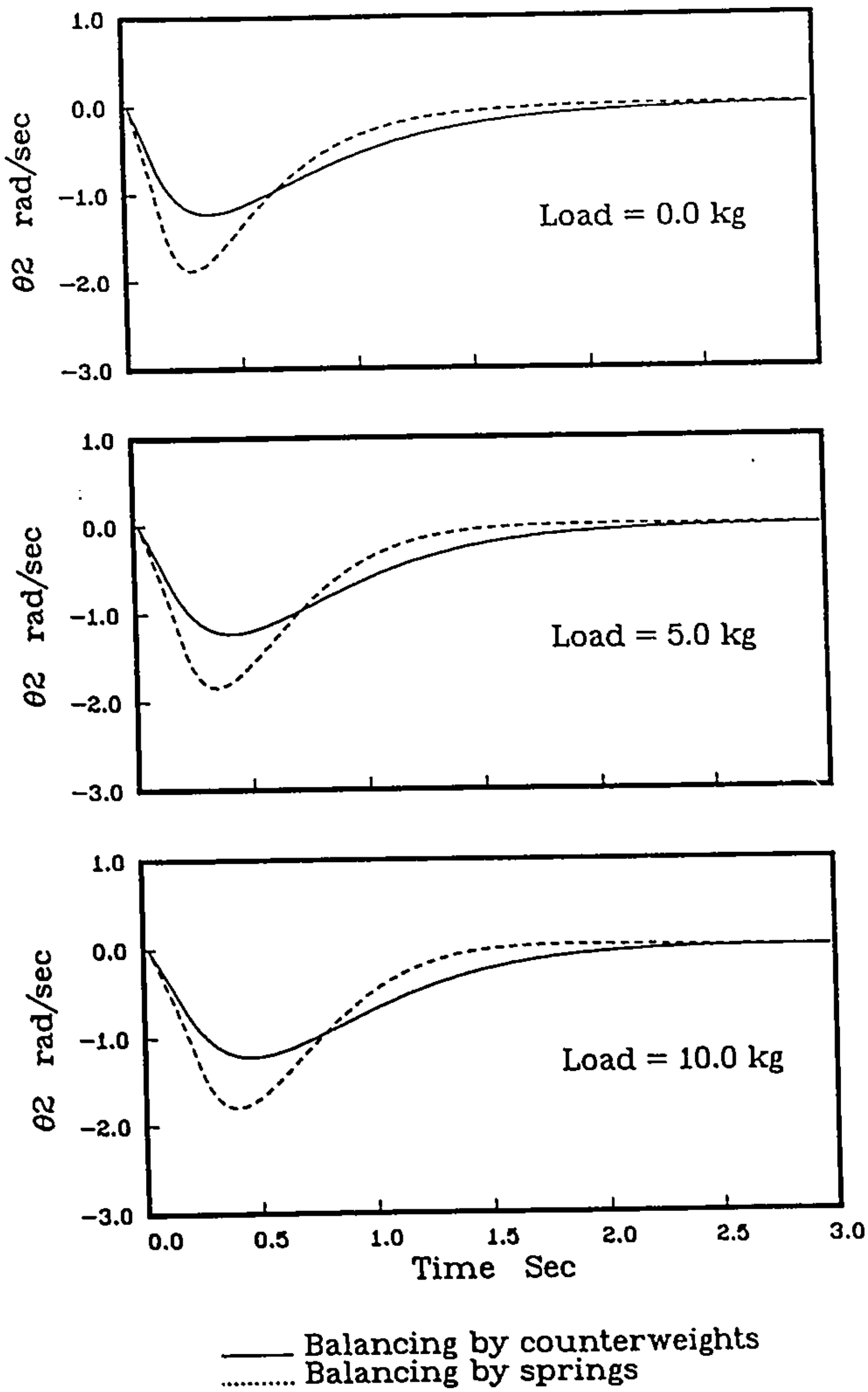

Figure A.4: Velocity of joint 2, systems balanced for 5 kg

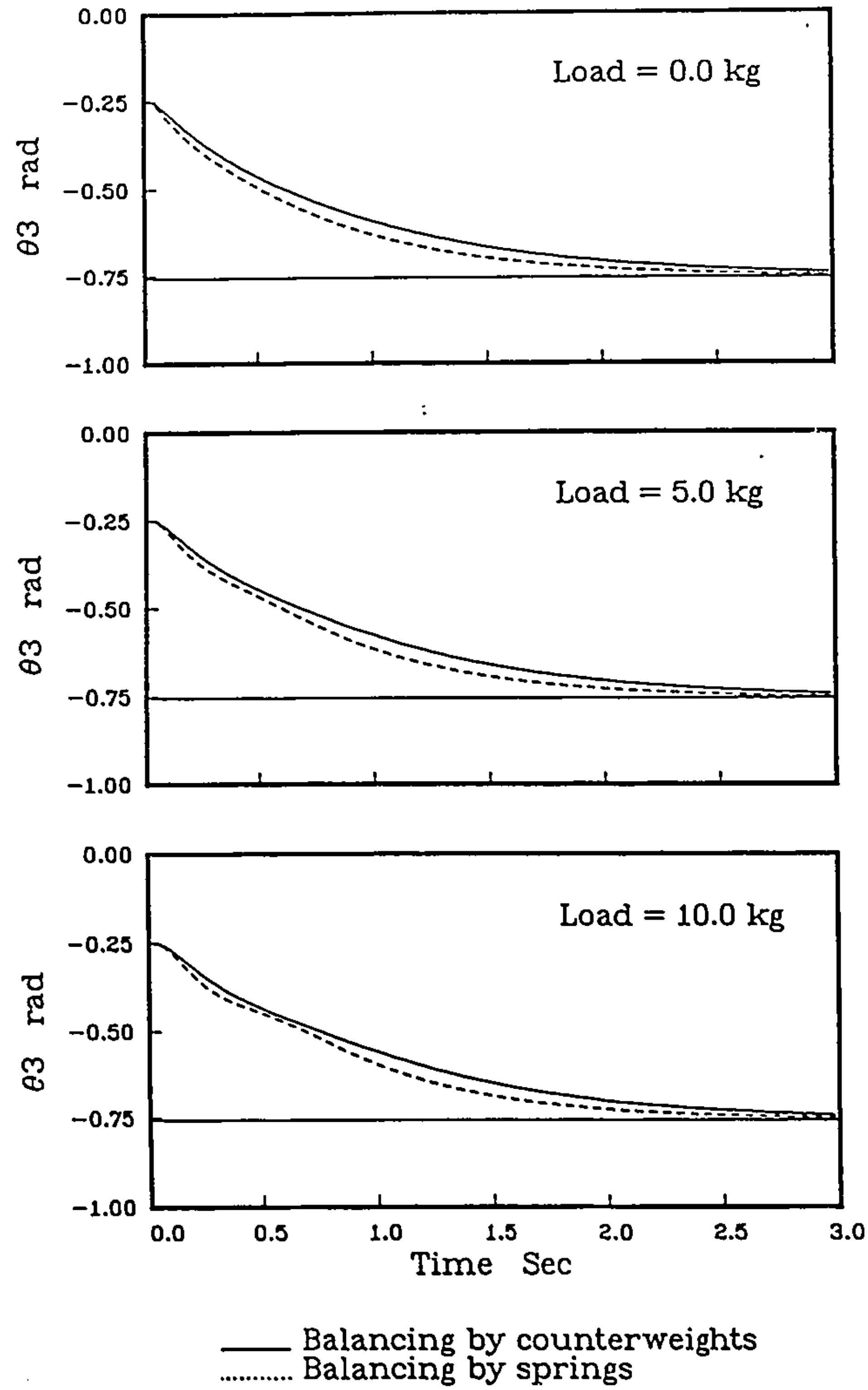

Figure A.5: Position of joint 3, systems balanced for 5 kg

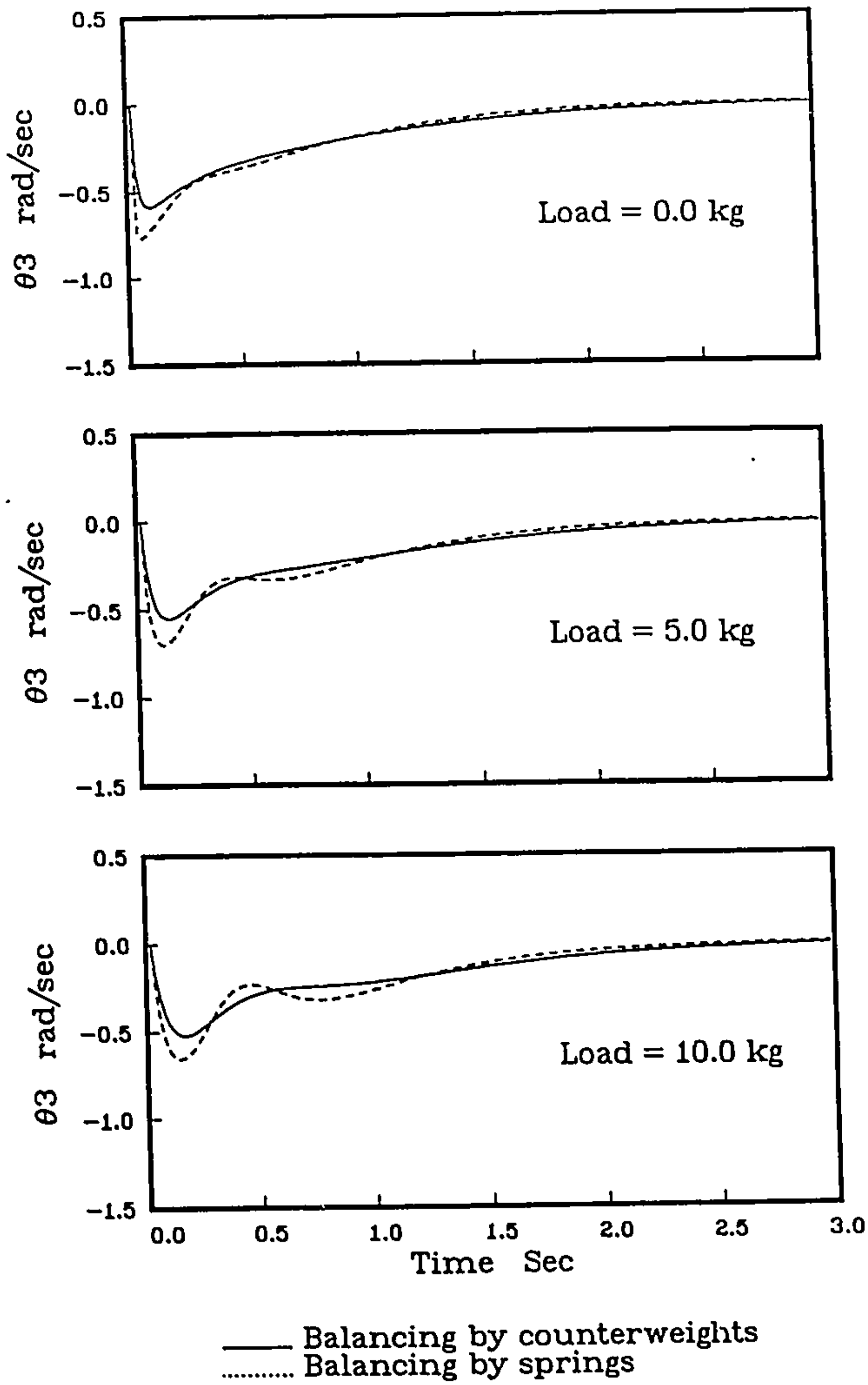

Figure A.6: Velocity of joint 3, systems balanced for 5 kg

Appendix B

Fortran soubroutines used in the simulation

```
C---------------------------------------------------------------
C This program analyzes the effect of replacing the counterweights
C used to balance a three revolute joint robot by spring
C mechanisms.
C The tuning is done numerically using the simples method for
C optimization of the ITAE criteria.
C---------------------------------------------------------------
C ------------------ MAIN PROGRAM -----------------------------
C---------------------------------------------------------------
      IMPLICIT REAL*4 (A-H,L-Z)
C
C  Read the characteristics of the manipulator and the task to
C  be performed by the robot
C
      CALL RMCHA
C
C  Read the characteristics of each actuator
C
      CALL CHMOTR
C
C  Call the coordination subroutine to select the payload and the
C  system to be analyzed.
C
      CALL CCTRL
C
      STOP
      END
C
C ---------------------------------------------------------------
C
C ---------------------------------------------------------------
C     This subroutine is to read the manipulator characteristics
C     and the essential trajectory points.
C
C
C     ....Li is the distance from the center of mass of
C         link i to the origin of frame i.
C     ....Ai is the length of link i
C     ....LiX,LiY,LiZ are the inertias about the axes X,Y and
C         Z of link i respectively,  i=1,2,3,L
C         L indicates the load carried by the robot.
C     ....M2,M3,ML are the masses of link 2,3 and payload
C---------------------------------------------------------------
C---------------------------------------------------------------
C
      SUBROUTINE  RMCHA
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/GGG/L2,L3,A2,A3
      COMMON/LLL/ L1X,L2X,L3X,L1Y,L2Y,L3Y,L1Z,L2Z,L3Z,LLX,LLY,LLZ
      COMMON/MMM/ M2,M3,ML
      COMMON/AAA/ XYZ(40,3),TTT(40,3),NTP
      DIMENSION  TJ(40,3)
      READ(7,*) A2,A3
      READ(7,*) NTP
      DO 110 I=1,NTP
  110 READ(7,*) XYZ(I,1),XYZ(I,2),XYZ(I,3)
      FACT=3.1415927/180.
```

```
      WRITE(2,16)
   16 FORMAT(//10X,'----------------------------------------------',/,
     *       10X,'- DETAILS OF ESSENTIAL TRAJECTORY POINTS --',/,
     *       10X,'----------------------------------------------',//,
     * '      X         Y         Z        T1        T2        T3')
      DO 100 I=1,NTP
      CALL INVKIN(XYZ(I,1),XYZ(I,2),XYZ(I,3),TTT(I,1),TTT(I,2),TTT(I,3))
      DO 120 J=1,3
  120 TJ(I,J)=TTT(I,J)
  100 WRITE(2,216) XYZ(I,1),XYZ(I,2),XYZ(I,3),TJ(I,1),TJ(I,2),TJ(I,3)
  216 FORMAT(2X,6F10.2)
      WRITE(2,700)
 700  FORMAT(//,10X,'-----------------------------------------',/,
     1        10X,'-- INERTIA AND MASS OF LINK 1,2, AND 3 ---',/,
     1        10X,'-----------------------------------------',//)
      READ(7,*) L1X,L1Y,L1Z,M1
      WRITE(2,710) L1X,L1Y,L1Z,M1
  710 FORMAT(//,'INERTIA AND MASS OF LINK 1'/
     2 '      I1X       I1Y        I1Z         M2 '/
     3  ,4F10.4,//)
      READ(7,*) L2X,L2Y,L2Z,M2
      WRITE(2,711) L2X,L2Y,L2Z,M2
  711 FORMAT(//,'INERTIA AND MASS OF LINK 2'/
     2 '      I2X       I2Y        I2Z         M2 '/
     3  ,4F10.4,//)
      READ(7,*) L3X,L3Y,L3Z,M3
      WRITE(2,712) L3X,L3Y,L3Z,M3
  712 FORMAT(//,'INERTIA AND MASS OF LINK 3'/
     2 '      I3X       I3Y        I3Z         M3 '/
     3  ,4F10.4,//)
      READ(7,*) LLX,LLY,LLZ,ML
      WRITE(2,713) LLX,LLY,LLZ,ML
  713 FORMAT(//,'INERTIA AND MASS OF THE PAYLOAD'/
     2 '      ILX       ILY        ILZ         ML '/
     3  ,4F10.4,//)
      READ(7,*) L2,L3
      WRITE(2,702) A2,A3,L2,L3
  702 FORMAT(//,10X,'-----------------------------------------------'
     *       /,10X,'-   LENGTH    LINK  2  AND  3                -'
     *       /,10X,'-----------------------------------------------'
     *       //,'                      A2          A3              '
     * /, 10X,2F10.4,//,
     *       /,10X,'-  LOCATION OF CENTER OF MASS OF LINK 2  AND
3 -'
     *       /,10X,'-----------------------------------------------'
     *       //,'                      L2          L3              ',
     * /,10X,2F10.4,//)
      RETURN
      END
C
C ------------------------------------------------------------
C ------------------------------------------------------------
C     THIS SUBROUTINE EVALUATE THE INVERSE KINEMATIC FOR AN
C     RRR MANIPULATOR.
C
C     - Input X Y Z ref. frame coords
C     - Output T1 T2 T2 joint coords (rad)
C
C------------------------------------------------------------
C------------------------------------------------------------
C
      SUBROUTINE INVKIN(X,Y,Z,T1,T2,T3)
```

```
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/GGG/L2,L3,A2,A3
      PI=3.1415927
      PI2=3.1415927*2
      T1=ATAN2(Y,X)
      C1=COS(T1)
      S1=SIN(T1)
      PX=C1*X+S1*Y
      PY=Z
      C3=(PX**2+PY**2-(A2**2+A3**2))/(2*A2*A3)
      IF(C3.GT.1.OR.C3.LT.-1) GO TO 100
      S3=-SQRT(1-C3**2)
      T3=ATAN2(S3,C3)
      R=C3*A3+A2
      F=S3*A3
      T2=ATAN2(R*PY-F*PX,R*PX+F*PY)
      IF(T2.GE.PI2) GO TO 100
      T3=T3+T2
      RETURN
  100 T1=.0
      T2=.0
      T3=.0
      WRITE(6,16) X,Y,Z
   16 FORMAT(//' POINT WITH FOLLOWING  COORDS IS OUT OF REACH:'/
     1 2X,3F10.2)
      STOP
      END
C
C ---------------------------------------------------------------
C ---------------------------------------------------------------
C     This subroutine evaluates the forward kinematic for an
C     RRR manipulator.
C
C     - Input T1,T2,T3 joint angles
C     - Output X,Y,Z
C
C----------------------------------------------------------------
C----------------------------------------------------------------
C
      SUBROUTINE FORKIN(T1,T2,T3,X,Y,Z)
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/GGG/L2,L3,A2,A3
      C1=COS(T1)
      S1=SIN(T1)
      C2=COS(T2)
      S2=SIN(T2)
      C3=COS(T3)
      S3=SIN(T3)
      X=(A2*C2+A3*C3)*C1
      Y=(A2*C2+A3*C3)*S1
      Z=A2*S2+A3*S3
      RETURN
      END
C
C
C
C ---------------------------------------------------------------
C ---------------------------------------------------------------
C       This subroutine is an integration routine to solve the
C   non-linear set of differential equations representing
C   the robot dynamic.
C       The routine is based on the R A N G E  K U T T A
C   M E T H O D forth order.
C
C----------------------------------------------------------------
C----------------------------------------------------------------
C
C
      SUBROUTINE  MODL(I,TR,NDIM,H)
```

```
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/YYY/ Y(2000,6)
      DIMENSION  F(6),A(6),B(6),C(6),D(6),YY(6)
C
         DO 22 J=1,NDIM
 22        YY(J)=Y(I,J)
       CALL FUNCT(YY(1),YY(2),YY(3),YY(4),YY(5),YY(6),TR,F)
         DO 33 J=1,NDIM
           A(J)=H/2*F(J)
 33        YY(J)=Y(I,J)+A(J)
       CALL FUNCT(YY(1),YY(2),YY(3),YY(4),YY(5),YY(6),TR,F)
         DO 44 J=1,NDIM
           B(J)=H/2*F(J)
 44        YY(J)=Y(I,J)+B(J)
       CALL FUNCT(YY(1),YY(2),YY(3),YY(4),YY(5),YY(6),TR,F)
         DO 55 J=1,NDIM
             C(J)=H/2*F(J)
 55          YY(J)=Y(I,J)+2*C(J)
       CALL FUNCT(YY(1),YY(2),YY(3),YY(4),YY(5),YY(6),TR,F)
         DO 66 J=1,NDIM
 66          D(J)=H/2*F(J)
         DO 77 J=1,NDIM
 77        Y(I+1,J)=Y(I,J)+(A(J)+2*B(J)+2*C(J)+D(J))/3
C
C
C   Set the geometric constraints
C
          IF(ABS(Y(I+1,1)).GE.3.0) THEN
            Y(I+1,1)=Y(I,1)
          ENDIF
C
          IF(ABS(Y(I+1,2)).GT.3.5) THEN
            Y(I+1,2)=3.5
          ENDIF
C
            IF(Y(I+1,3).GE.3.0.OR.Y(I+1,3).LT.-3.0) THEN
              Y(I+1,3)=Y(I,3)
            ENDIF
C
          IF(ABS(Y(I+1,4)).GT.3.5) THEN
            Y(I+1,4)=3.5
          ENDIF
C
            IF(Y(I+1,5).GE.3.0.OR.Y(I+1,5).LT.-3.0) THEN
               Y(I+1,5)=Y(I,5)
            ENDIF
C
          IF(ABS(Y(I+1,6)).GT.3.5) THEN
            Y(I+1,6)=3.5
          ENDIF
      RETURN
      END
C
C ---------------------------------------------------------------
C This subroutine contains the initial conditions of the
C numerical integration algorithm.
C  ....... Yi0: Initial position of joint i
C  ....... YDi0: Initial velocity of joint i
C ---------------------------------------------------------------
C
      SUBROUTINE INCON(Y10,YD10,Y20,YD20,Y30,YD30)
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/AAA/ XYZ(40,3),TTT(40,3),NTP
       Y10=TTT(1,1)
       YD10=0.0
```

```
      Y20=TTT(1,2)
      YD20=0.0
      Y30=TTT(1,3)
      YD30=0.0
      RETURN
      END
C
C ---------------------------------------------------------------
C ---------------------------------------------------------------
C This subroutine evaluates the element of the manipulator
C inertia matrix. HI(i,j) is the element at row i and column j
C ---------------------------------------------------------------
C
      SUBROUTINE HMTRX(T1,T2,T3,HI)
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/LLL/ L1X,L2X,L3X,L1Y,L2Y,L3Y,L1Z,L2Z,L3Z,LLX,LLY,LLZ
      COMMON/MMM/ M2,M3,ML
      COMMON/GGG/L2,L3,A2,A3
      COMMON/BAL/L2B,L3B,MLBAL,M2B,M3B,
     *            L2BX,L2BY,L2BZ,L3BX,L3BY,L3BZ
      DIMENSION HI(3,3)
C
          T32=T3-T2
          C3=COS(T3)
          S3=SIN(T3)
          C2=COS(T2)
          S2=SIN(T2)
          C1=COS(T1)
          S1=SIN(T1)
          C32=COS(T32)
          S32=SIN(T32)
C
C
      HI(1,1)=L1Z+L2X*C2**2+L2Z*S2**2+M3*(A2*C2+L3*C3)**2
     *        +M2*(L2*C2)**2+(L3X+LLX)*C3**2+(L3Z+LLZ)*S3**2
     *        +ML*(A2*C2+A3*C3)**2
C
     *        +M3B*(L3B*C3)**2+L3BX*C3**2
     *        +L3BZ*S3**2+M2B*(L2B*C2)**2
     *        +L2BX*C2**2+L2BZ*S2**2
C
      HI(1,2)=0.0
      HI(1,3)=C.0
      HI(2,1)=0.0
      HI(3,1)=0.0
C
      HI(2,2)=M2*L2**2+(M3+ML)*A2**2+L2Y
     *        +M2B*L2B**2+L2BY
      HI(2,3)=(M3*L3**2+ML*A3**2)*C32
      HI(3,2)=HI(2,3)
      HI(3,3)=M3*L3**2+ML*A3**2+L3Y+LLY
     *        +M3B*L3B**2+L3BY
      RETURN
      END
C
C ---------------------------------------------------------------
C ---------------------------------------------------------------
C This subroutine evaluates the elements of the h vector
C representing the coriolus and centrifugal effects.
C  ... h is a 1 by 3 vector.
C ---------------------------------------------------------------
C
      SUBROUTINE HVECT(T1,T1D,T2,T2D,T3,T3D,HC)
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/LLL/ L1X,L2X,L3X,L1Y,L2Y,L3Y,L1Z,L2Z,L3Z,LLX,LLY,LLZ
      COMMON/MMM/ M2,M3,ML
```

```
      COMMON/GGG/L2,L3,A2,A3
      COMMON/BAL/L2B,L3B,MLBAL,M2B,M3B,
     *           L2BX,L2BY,L2BZ,L3BX,L3BY,L3BZ
      DIMENSION HC(3)
C
           T32=T3-T2
           C3=COS(T3)
           S3=SIN(T3)
           C2=COS(T2)
           S2=SIN(T2)
           C1=COS(T1)
           S1=SIN(T1)
           C32=COS(T32)
           S32=SIN(T32)
C
      HC(1)=2*((L2Z-L2X)*C2*S2-M2*L2**2*S2*C2)*T1D*T2D
     *     +2*(L3Z+LLZ-L3X-LLX)*C3*S3*T1D*T3D
     *     -2*M3*(A2*C2+L3*C3)*(A2*S2*T2D+L3*S3*T3D)*T1D
     *     -2*ML*(A2*C2+A3*C3)*(A2*S2*T2D+A3*S3*T3D)*T1D
C
     *     +2*((L2BZ-L2BX)*C2*S2-M2B*L2B**2*S2*C2)*T1D*T2D
     *     +2*(L3BZ-L3BX)*C3*S3*T1D*T3D
     *     -2*M3B*L3B**2*C3*S3*T3D*T1D
C
      HC(2)=-(M3*L3**2+ML*A3**2)*S32*(T3D-T2D)*T3D
     *     +(M2*L2**2+(L2X-L2Z))*S2*C2*T1D**2
     *     -(M3*L3+ML*A3)*A2*S32*T2D*T3D
     *     +(M3*(A2*C2+L3*C3)+ML*(A2*C2+A3*C3))*A2*S2*T1D**2
C
     *     -(M3B*L3B**2)*S32*(T3D-T2D)*T3D
     *     +(M2B*L2B**2+(L2BX-L2BZ))*S2*C2*T1D**2
C
      HC(3)=(M3*L3+ML*A3)*A2*S32*T2D**2
     *     +(L3X+LLX-L3Z-LLZ)*S3*C3*T1D**2
     *     +(M3*L3*(A2*C2+L3*C3)+ML*A3*(A2*C2+A3*C3))*S3*T1D**2
C
     *     +(L3BX-L3BZ)*S3*C3*T1D**2
     *     +M3B*L3B**2*C3*S3*T1D**2
C
      RETURN
      END
C
C ----------------------------------------------------------------
C This subroutine is to calculate the amount of unbalance
C given the gravity torque and the balancing torque.
C ----------------------------------------------------------------
C
      SUBROUTINE GRVTY(T2,T3,GV)
      IMPLICIT REAL*4 (A-H,L-Z)
      DIMENSION  GV(3)
C
C
      GV(1)=0.0
C
      CALL GVV(T2,T3,G2,G3)
      CALL GSS(T2,T3,TS2,TS3)
C
      GV(2)=G2-TS2
C
      GV(3)=G3-TS3
C
      RETURN
      END
C
C ----------------------------------------------------------------
C This subroutine calculates the effect of the gravity torque
C when no balancing is used.
```

```
C  G2: Gravity effect on joint 2
C  G3: Gravity effect on joint 3
C ----------------------------------------------------------------
      SUBROUTINE GVV(T2,T3,G2,G3)
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/MMM/ M2,M3,ML
      COMMON/GGG/L2,L3,A2,A3
      COMMON/BAL/L2B,L3B,MLBAL,M2B,M3B,
     *            L2BX,L2BY,L2BZ,L3BX,L3BY,L3BZ
C
          G=9.82
          C3=COS(T3)
          C2=COS(T2)
C
C GRAVITY TORQUE AT JOINT 2
C
      G2=((M2*L2+(M3+ML)*A2)-M2B*L2B)*G*C2
C
C GRAVITY TORQUE AT JOINT 3
C
      G3=((M3*L3+ML*A3)-M3B*L3B)*G*C3
C
      RETURN
      END
C ----------------------------------------------------------------
C This subroutine calculates the torque applied by the balancing
C mechanism.
C  TS2: Torque applied by the spring mechanism at joint 2
C  TS3: Torque applied by the spring mechanism at joint 3
C ----------------------------------------------------------------
      SUBROUTINE GSS(T2,T3,TS2,TS3)
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/MMM/ M2,M3,ML
      COMMON/GGG/L2,L3,A2,A3
      COMMON/LEN/ L02,L03,LMAX2,LMAX3,OB2,OB3,SC2,SC3
      COMMON/BAL/L2B,L3B,MLBAL,M2B,M3B,
     *            L2BX,L2BY,L2BZ,L3BX,L3BY,L3BZ
C
          G=9.82
          PI=3.1415927
          C3=COS(T3)
          C2=COS(T2)
C
      IF(OB2.EQ.0.0) GOTO 111
C
C EQUILIBRUIM CONDITIONS
C
      OA2=(M2*L2+(M3+MLBAL)*A2)*G/(OB2*SC2)
      LMAX2=OB2-OA2
C
C SPRING TORQUE AT JOINT 2
C
      TS2=SC2*OA2*OB2*SIN(PI/2-T2)
C
C EQUILIBRUIM CONDITIONS
C
      OA3=(M3*L3+MLBAL*A3)*G/(OB3*SC3)
      LMAX3=OB3-OA3
C
C SPRINT TORQUE AT JOINT 3
C
      TS3=SC3*OA3*OB3*SIN(PI/2-T3)
      GOTO 222
 111  TS2=0.0
      TS3=0.0
C
 222  RETURN
      END
```

```
C ------------------------------------------------------------------
C ------------------------------------------------------------------
C
      SUBROUTINE FUNCT(T1,T1D,T2,T2D,T3,T3D,TR,FF)
      IMPLICIT REAL*4 (A-H,L-Z)
      DIMENSION HI(3,3),HC(3),TR(3),FF(6),GV(3)
      CALL HMTRX(T1,T2,T3,HI)
      CALL HVECT(T1,T1D,T2,T2D,T3,T3D,HC)
      CALL GRVTY(T2,T3,GV)
      FF(1)=T1D
      FF(2)=(TR(1)-HC(1)-GV(1))/HI(1,1)
      FF(3)=T2D
      F2P=TR(2)-HC(2)-GV(2)
      F3P=TR(3)-HC(3)-GV(3)
      DETER=HI(2,2)*HI(3,3)-HI(3,2)*HI(2,3)
      D4=F2P*HI(3,3)-F3P*HI(2,3)
      FF(4)=D4/DETER
      FF(5)=T3D
      D6=F3P*HI(2,2)-F2P*HI(3,2)
      FF(6)=D6/DETER
      RETURN
      END
C
C ------------------------------------------------------------------
C ------------------------------------------------------------------
C     This subroutine reads and writes out the characteristics
C     of the joint actuators.
C
C-------------------------------------------------------------------
C-------------------------------------------------------------------
C
      SUBROUTINE CHMOTR
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/CHRT/ AMP(3),R(3),CMAX(3),TRC(3),EMF(3),RTIO(3)
      DO 55 I=1,3
      READ(7,*) AMP(I),R(I),CMAX(I),TRC(I),EMF(I),RTIO(I)
 55   WRITE(2,710) I,AMP(I),R(I),CMAX(I),TRC(I),EMF(I),RTIO(I)
 710  FORMAT(//,'CHARACTERISTICS  OF MOTOR OF LINK',I5//
     *         ,10X,'CURRENT AMPLIFICATION FACTOR',F10.6,//
     *         ,10X,'ARMATURE RESISTANCE  ',F10.6,2X,'OHMS',//
     *         ,10X,'MAXIMUM CURRENT',F10.6,'AMPS'//
     *         ,10X,'MOTOR CONSTANT ',F10.6,2X,'NM/AMP',//
     *         ,10X,'BACK EMF CONSTANT ',F10.6,2X,'VOLT/RAD',//
     *         ,10X,'GEAR RATIO',F5.3,//)
      RETURN
      END
C|-------------------------------------------------------------------
C|  This subroutine controls the inputs to the control algorithm:
C|  Which system to analyzed ( spring balanced arm or weight
C|  balanced arm as well the load carried by each system.
C|-------------------------------------------------------------------
      SUBROUTINE CCTRL
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/STEP/ KMAX,H
      COMMON/CHRT/ AMP(3),R(3),CMAX(3),TRC(3),EMF(3),RTIO(3)
      COMMON/MMM/ M2,M3,ML
      COMMON/GGG/L2,L3,A2,A3
      COMMON/BAL/L2B,L3B,MLBAL,M2B,M3B,
     *            L2BX,L2BY,L2BZ,L3BX,L3BY,L3BZ
      COMMON/LEN/ L02,L03,LMAX2,LMAX3,OB2,OB3,SC2,SC3
      COMMON/STM/ ISTM
C
C                READ THE  INTEGRATION STEP
C
```

```
      READ(7,*) H
C
C            READ THE MAXIMUM SAMPLING POINTS BETWEEN STEPS
C
      READ(7,*) KMAX
      ISTM=1
      MLBAL=5.0
C
            L2B=0.1
            L3B=0.1
            M2B=((MLBAL+M3)*A2+M2*L2)/L2B
            M3B=(MLBAL*A3+M3*L3)/L3B
C
C
            L2BX=1.0
            L2BY=1.0
            L2BZ=1.0

            L3BX=1.0
            L3BY=1.0
            L3BZ=1.0
C
            SC2=0.0
            SC3=0.0
            OB2=0.0
            OB3=0.0
C
       GO TO 111
C
  222       L2B=0.0
            L3B=0.0
            M2B=0.0
            M3B=0.0
C
            L2BX=0.0
            L2BY=0.0
            L2BZ=0.0
C
            L3BX=0.0
            L3BY=0.0
            L3BZ=0.0
C
            SC2=1000.
            SC3=1000.
            OB2=0.3
            OB3=0.3
C
      L02=0.0
      L03=0.0
 111  CALL CONTL
        ISTM=ISTM+1
        IF(ISTM.EQ.2) GO TO  222
      WRITE(6,830)
 830  FORMAT(///,' ***** EXECUTION ENDED  *****',///)
      RETURN
      END
C
C -------------------------------------------------------------
C  Control algorithm:  The subroutine calls the simplex routine
C  in order to tune the controller parameters
C -------------------------------------------------------------
C
      SUBROUTINE CONTL
      IMPLICIT REAL*4 (A-H,L-Z)
      REAL*4 KP1,KP2,KP3,KI1,KI2,KI3,KD1,KD2,KD3,KOPT
      COMMON/XXX/ X(2000,3,4)
      COMMON/YYY/ Y(2000,6)
      COMMON/VVV/ V(2000,40,4)
      COMMON/STM/ ISTM
      COMMON/TIME/TIME(2000)
      COMMON/STEP/ KMAX,H
      COMMON/TRR/ TOR(2000,3)
```

```
      DIMENSION KOPT(3)
C
C  N :  Number of variables
C
      N=6
C
      CALL SIMPLX(N,KOPT)
C
      DO 11 K=1,KMAX
              CALL FORKIN(Y(K,1),Y(K,3),Y(K,5),X(K,1,ISTM),X(K,2,ISTM),
     *                     X(K,3,ISTM))
              TIME(K)=H*(K-1)
C
C                 ***  POSITION  ***
C
                  V(K,1,ISTM)=Y(K,1)
                  V(K,2,ISTM)=Y(K,3)
                  V(K,3,ISTM)=Y(K,5)
C
C                 ***  VELOCITY  ***
C
                  V(K,4,ISTM)=Y(K,2)
                  V(K,5,ISTM)=Y(K,4)
                  V(K,6,ISTM)=Y(K,6)
C
C                 ***  TORQUE    ***
C
                  V(K,7,ISTM)=TOR(K,1)
                  V(K,8,ISTM)=TOR(K,2)
 11               V(K,9,ISTM)=TOR(K,3)
C
      WRITE(2,*) ISTM
      WRITE(2,335) KOPT(1),KOPT(2),KOPT(3),KOPT(4),KOPT(5),KOPT(6)
 335  FORMAT(//,20X,'RESPONCE FOR A PD CONTROLLER ',
     *          /,10X,'KP1           KP2           KP3',
     *          /,5X,3F10.3,
     *          /,10X,'KD1           KD2           KD3',
     *          /,5X,3F10.3)
      RETURN
      END
C
C |------------------------------------------------------------|
C |  THis subroutine is an optimization routine that uses the  |
C |  simplex method  by Nedler and Mead.
C |------------------------------------------------------------|
C
      SUBROUTINE SIMPLX(N,XC)
      IMPLICIT REAL*4 (A-H,L-Z)
      DIMENSION X(20,20),XO(20),E(20,20),F(20),XJ(20),XH(20),XL(20),
     *          SUM(20),XR(20),XE(20),XN(20),XC(20)
      REAL*4 K
C
C         LMDC= CONTRACTION FACTOR
C
      LMDC=0.5
C
C         LMDE= EXPANSION FACTOR
C
      LMDE=2.0
C
C         CONVERGENCE CRITERIA
C
      DELTA=0.01
      K=10.
      KK=1
      KKM=100
C                            SET THE STARTING POINT (ARBITRARELY)
      DO 5 I=1,N
```

```
        XO(I)=10.
        DO  10  J=2,N+1
          IF(I.EQ.J-1) THEN
            E(J,I)=1.0
          ELSE
             E(J,I)=0.0
          ENDIF
 10     CONTINUE
 5    CONTINUE
C
C           LOCATE THE OTHER N POINTS IN THE N SPACE
C
      DO 11 J=1,N+1
         DO 12 I=1,N
              X(J,I)=XO(I)+K*E(J,I)
 12      CONTINUE
 11   CONTINUE
C
C                 EVALUATE  THE FUNCTION VALUE AT EACH POINT
C
 444  DO 13 J=1,N+1
         DO  14  I=1,N
 14           XJ(I)=X(J,I)
         CALL FUNC(N,XJ,FF)
         F(J)=FF
 13   CONTINUE
C
C                LOCATE XH  ( POINT WITH HIGHEST FUNCTION VALUE)
C
 666  FH=F(1)
      JH=1
      DO 15 J=1,N+1
         IF(FH.LT.F(J)) THEN
             FH=F(J)
             JH=J
             DO  16  I=1,N
                 XH(I)=X(JH,I)
 16          CONTINUE
         ENDIF
 15   CONTINUE
C
C                  LOCATE XL ( POINT WITN LOWEST FUNCTION VALUE
)
C
      FL=F(1)
      JL=1
      DO  17  J=1,N+1
         IF(FL.GT.F(J)) THEN
           FL=F(J)
           JL=J
           DO  18  I=1.N
               XL(I)=X(JL,I)
 18        CONTINUE
         ENDIF
 17   CONTINUE
C
C                           LOCATE THE CENTROID XC
      DO 19 I=1,N
      SM=0
         DO 20  J=1,N+1
            SM=X(J,I)+SM
 20      CONTINUE
         SUM(I)=SM
         XC(I)=(SUM(I)-XH(I))/N
 19   CONTINUE
C                EVALUATE THE FUNCTION VALUE AT XC
      CALL FUNC(N,XC,FC)
C
C                 LOCATE THE REFLECTED POINT XR
```

```
C
      DO 21  I=1,N
         XR(I)=2*XC(I)-XH(I)
 21   CONTINUE
      KK=KK+1
      IF(KK.GE.KKM) GO TO 555
C
C                 EVALUATE THE FUNCTION AT XR                       C
ALL07190
              CALL FUNC(N,XR,FR)
         IF(FR.LT.FL)  THEN
C
C                 EXPAND
C
                   DO 22 I=1,N
                        XE(I)=XC(I)*(1-LMDE)+LMDE*XR(I)
  22               CONTINUE
C
C                EVALUATE THE FUNCTION VALUE AT XE
C
                  CALL FUNC(N,XE,FE)
                  IF(FE.LT.FR) THEN
                       F(JH)=FE
                       DO  23 I=1,N
                         X(JH,I)=XE(I)
 23                   CONTINUE
C
C        CHECK CONVERGENCE
C
                      GO TO 999
                  ELSE
                    F(JH)=FR
                    DO 24 I=1,N
                       X(JH,I)=XR(I)
  24                CONTINUE
C
C      CHECK CONVERGENCE
C
                    GO TO 999
                  ENDIF
         ELSE
           IF(FR.LT.FC) THEN
             F(JH)=FR
             DO 25 I=1,N
               X(JH,I)=XR(I)
               GO TO 999
  25         CONTINUE
           ELSE
             IF(FR.LT.FH) THEN
               F(JH)=FR
               DO 27 I=1,N
                 X(JH,I)=XR(I)
  27           CONTINUE
             ENDIF
C
C       LOCATE XN (CONTRACTION)
C
             DO 28 I=1,N
                 XN(I)=XC(I)*(1-LMDC)+LMDC*XH(I)
 28          CONTINUE
C
C       EVALUATE THE FUNCTION VALUE AT XN
C
             CALL FUNC(N,XN,FN)
             IF(FN.LT.FH) THEN
               F(JH)=FN
               DO 29  I=1,N
                  X(JH,I)=XN(I)
 29            CONTINUE
C
```

```
C       CHECK CONVERGENCE
C
                GO TO 999
              ELSE
                DO 30 J=1,N+1
                  DO 31 I=1,N
                    X(J,I)=(X(J,I)+XL(I))/2
 31               CONTINUE
 30             CONTINUE
               DO 54 I=1,N
               SUMX=0
                 DO 65  J=2,N+1
 65              SUMX=ABS(X(J-1,I)-X(J,I))+SUMX
             IF(SUMX.GT.DELTA)  GO TO 444
 54          CONTINUE
             ENDIF
        ENDIF
C
C        CONVRGENCE   CRITERIA
C
  999   SF=0
        DO 55 J=1,N+1
  55       SF=SF+F(J)
        FMEAN=SF/N
        SUMF=0
        DO  66  J=1,N+1
  66        SUMF=SUMF+(F(J)-FMEAN)**2/N
         SEGMA=SQRT(SUMF)
         IF(SEGMA.GT.DELTA)  GO TO 666
      ENDIF
 555  WRITE(2,100) FC
  100 FORMAT(/,10X,'THE MINIMUM IS F(X)=',F4.1,/,10X,'LOCATED AT:')
      DO 50 I=1,N
  50  WRITE(2,200) I,XC(I)
  200 FORMAT(10X,'X',I1,'=',F6.3)
      RETURN
      END
C
C ---------------------------------------------------------------
C               Evaluation of the prformance index
C ---------------------------------------------------------------
C
C
C ---------------------------------------------------------------
C     This subroutine evaluates the error at each sampling period
C ---------------------------------------------------------------
C
      SUBROUTINE FUNC(N,XOPT,FUN)
      IMPLICIT REAL*4 (A-H,L-Z)
      REAL*4 KP1,KP2,KP3,KD1,KD2,KD3,KOPT
      COMMON/YYY/ Y(2000,6)
      COMMON/AAA/XYZ(40,3),TTT(40,3),NTP
      COMMON/ERR/ ERR(2000,6)
      COMMON/STEP/ KMAX,H
      COMMON/TRR/ TOR(2000,3)
C-----------------------
      DIMENSION GV(3),TR(3),XOPT(20),D(2,6)
      NDIM=6
      EPS=0.001
C
C                SET INITIAL STATE
C
      CALL INCON(Y(1,1),Y(1,2),Y(1,3),Y(1,4),Y(1,5),Y(1,6))
C
      CALL GRVTY(Y(1,3),Y(1,5),GV)
C
      DO 8 J=1,3
          TOR(1,J)=GV(J)
   8      TR(J)=GV(J)
```

```
C
C
C               SET DESIRED STATE
C
      DO 9 I=1,NTP
         D(I,1)=TTT(I,1)
         D(I,2)=0.0
         D(I,3)=TTT(I,2)
         D(I,4)=0.0
         D(I,5)=TTT(I,3)
  9      D(I,6)=0.0
C
      DO 99 I=1,N
 99   IF(XOPT(I).LE.0.0) GO TO 555
          KP1=XOPT(1)
          KP2=XOPT(2)
          KP3=XOPT(3)
C
          KD1=XOPT(4)
          KD2=XOPT(5)
          KD3=XOPT(6)
C
C
          K=1
          DO 15 J=1,NDIM
 15         ERR(K,J)=D(2,J)-Y(K,J)
            GO TO 777
 100        DO 22 J=1,NDIM
C
C        CALCULATE THE ERROR BETWEEN THE INITIAL AND FINAL POSITION
C
  22          ERR(K,J)=D(2,J)-Y(K,J)
              IF(ABS(ERR(K,1)).LE.EPS.AND.ABS(ERR(K,3)).LE.EPS.
     *           AND.ABS(ERR(K,5)).LE.EPS) GO TO 33
              CALL TORQ(K,KP1,KP2,KP3,KI1,KI2,KI3,KD1,KD2,
     *        KD3,TR)
         DO 24 J=1,3
  24          TOR(K,J)=TR(J)
 777          CALL MODL(K,TR,NDIM,H)
              K=K+1
              IF(K.LE.KMAX) GO TO 100
  33          KMAX=K-1
C
C                    EVALUATE THE PERFORMANCE INDEX
C
         CALL INDEX(FUN)
         GO TO 666
 555     FUN=50000.
 666     RETURN
         END
C
C ----------------------------------------------------------------------
C This subroutine evaluates the sum of the ITAEs of the three joints
C ----------------------------------------------------------------------
C
      SUBROUTINE INDEX(FUN)
      IMPLICIT REAL*4 (A-H,L-Z)
      COMMON/ERR/ ERR(2000,6)
      COMMON/STEP/ KMAX,H
      DIMENSION   DIAE(3)
      DO 10 J=1,3
 10     DIAE(J)=0.0
      DO 11 I=2,KMAX
        DIAE(1)=DIAE(1)+H*(H*I)*ABS(ERR(I,1))
        DIAE(2)=DIAE(2)+H*(H*I)*ABS(ERR(I,3))
 11     DIAE(3)=DIAE(3)+H*(H*I)*ABS(ERR(I,5))
```

```
      FUN=DIAE(1)+DIAE(2)+DIAE(3)
      RETURN
      END
C
C ---------------------------------------------------------------
C This subroutine evaluates the torques at each joint besed on
C the PD principle.
C ---------------------------------------------------------------
C
      SUBROUTINE TORQ(K,KP1,KP2,KP3,KI1,KI2,KI3,KD1,KD2,KD3,TR)
      IMPLICIT REAL*4 (A-H,L-Z)
      REAL*4 KP1,KP2,KP3,KD1,KD2,KD3
      COMMON/CHRT/ AMP(3),R(3),CMAX(3),TRC(3),EMF(3),RTIO(3)
      COMMON/STEP/ KMAX,H
      COMMON/YYY/ Y(2000,6)
      COMMON/ERR/ ERR(2000,6)
      COMMON/STM/ ISTM
      DIMENSION U1(2000,3),UP(2000,3),UI(2000,3),UD(2000,3),
     *          U(2000,3),GV(3),OMGD(3),TR(3),CRNT(3)
C
C
       OMGD(1)=Y(K,2)*RTIO(1)
       OMGD(2)=Y(K,4)*RTIO(2)
       OMGD(3)=Y(K,6)*RTIO(3)
C
       U1(1,1)=ERR(1,1)
       U1(1,2)=ERR(1,3)
       U1(1,3)=ERR(1,5)
C
C
C           PROPORTIONAL PART
C
       UP(K,1)=KP1*ERR(K,1)
       UP(K,2)=KP2*ERR(K,3)
       UP(K,3)=KP3*ERR(K,5)
C
C           DERIVATIVE PART
C
       UD(K,1)=KD1*ERR(K,2)
       UD(K,2)=KD2*ERR(K,4)
       UD(K,3)=KD3*ERR(K,6)
C
C
C *** U(K,J) Is The actuating signal of the controller J ***
C *** at the instant k                                     ***
C
        DO 700 J=1,3
               U(K,J)=UP(K,J)+UI(K,J)+UD(K,J)
C
C                  SET CURRENT LIMIT
C
                CRNT(J)=(AMP(J)*U(K,J)-EMF(J)*OMGD(J))/R(J)
                IF(CRNT(J).GE.CMAX(J))  CRNT(J)=CMAX(J)
                IF(CRNT(J).LE.-CMAX(J))  CRNT(J)=-CMAX(J)
 700     CONTINUE
C
C            CALULATE THE OUTPUT TORQUE FROM THE MOTOR
C
        DO 70  J=1,3
 70         TR(J)=TRC(J)*CRNT(J)*RTIO(J)
       RETURN
       END
C
C
```