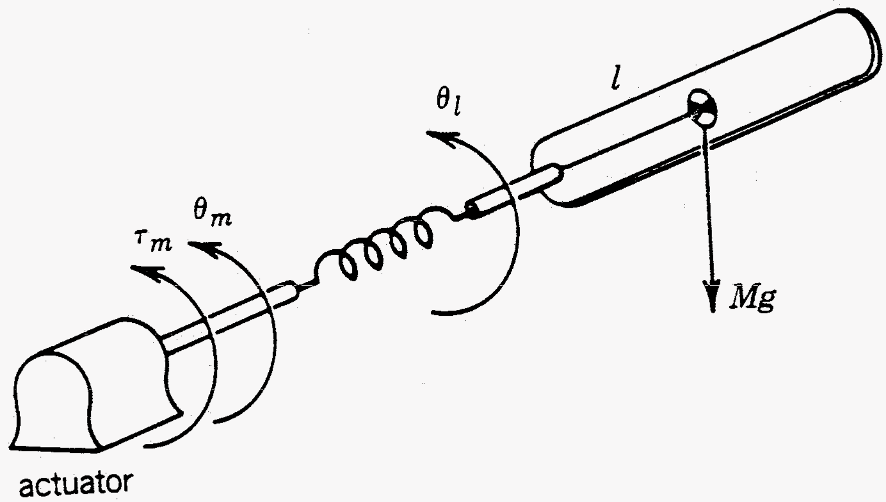
Feedback linearization control of a single link manipulator with joint elasticity

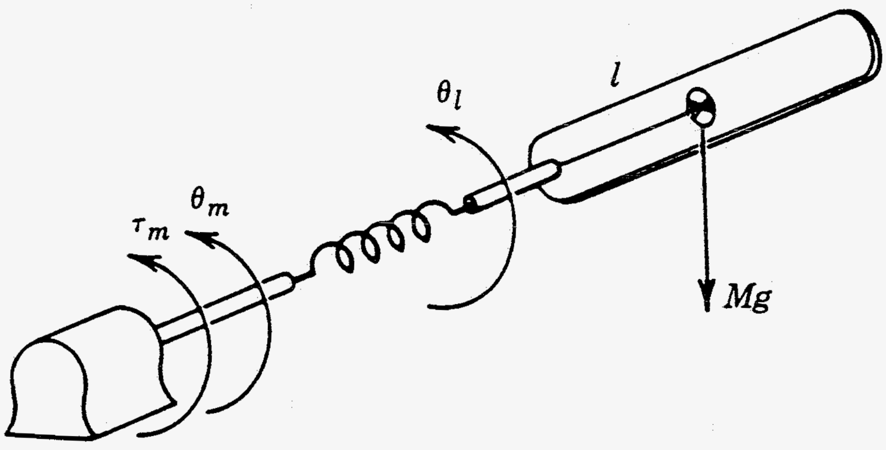
FEEDBACK LINEARIZATION CONTROL
OF A
SINGLE LINK MANIPULATOR
WITH
JOINT ELASTICITY



Ricky Doelman

lecture: 18 Sept. 1991

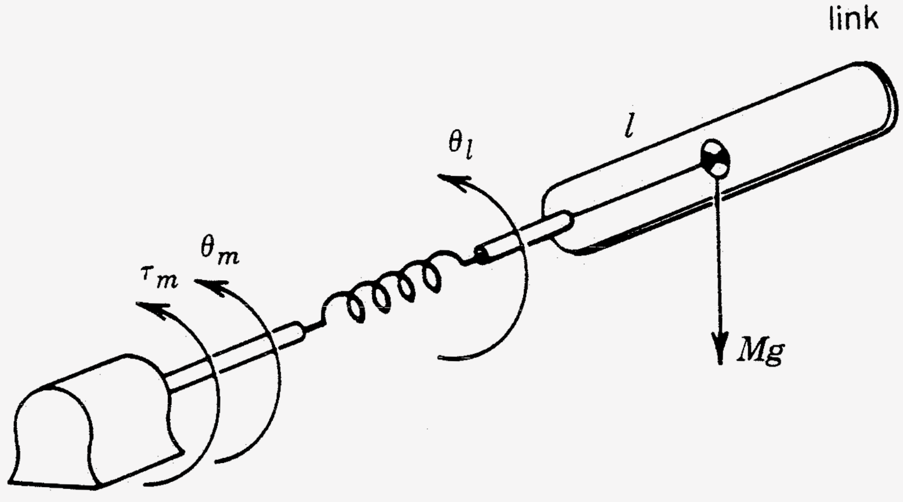
FEEDBACK LINEARIZATION CONTROL
OF A
SINGLE LINK MANIPULATOR
WITH
JOINT ELASTICITY



Ricky Doelman

lecture: 18 Sept. 1991

FEEDBACK LINEARIZATION CONTROL OF A SINGLE LINK MANIPULATOR WITH JOINT ELASTICITY



Ricky Doelman

lecture: 18 Sept. 1991

SUMMARY

Joint elasticity seems to be the main source of compliance in most manipulator designs. **Feedback linearization control** provides us with a method for tackling the problem of controlling these nonlinear dynamical systems. This report deals with the feedback linearization control of a rather simple mechanical system:
a one-link manipulator with joint elasticity.

In order to get a better understanding of the feedback linearization technique we first discuss some basic concepts from differential geometry. With the help of these concepts we derive necessary and sufficient conditions for a system to be feedback linearizable. The key to applying feedback linearization control is finding a (nonlinear) change of coordinates in the state space and a (nonlinear) control law. We show how this problem is reduced to solving a set of partial differential equations. Then, after an accurate modeling of the mechanical system, we put the theory on feedback linearization control into practice and simulate the dynamical behaviour of the control system using MatLab.

On the basis of these simulations we conclude that the feedback linearization technique is sensitive to changes in the systemparameters, and it therefore requires precise modeling. This in strong contrast to the findings of Spong in [1]!

Next, we designed a blueprint for an assignment that can be used during the course "werktuigkundig regelen IV".

In order to avoid the assumption that feedback linearization is some kind of universal remedy for solving any kind of nonlinear control problem, we discuss some restrictions and disadvantages of the control method. We are aware that this treatment on feedback linearization control of a single link manipulator with joint elasticity is far from being complete, but it provides a solid basis for studying and application of advanced feedback linearization control.

CONTENTS

REFERENCES

APPENDIX I: "Modeling and Control of Elastic Joint Robots"
M. W. Spong
Journal of Dynamic Systems, Measurement
and Control, Vol. 109, pp 310 - 319, Dec. 1987

APPENDIX II: Computational Details for the Model with Damping

APPENDIX III: Computer Programs Used for Simulations

APPENDIX IV: Some Computational Details

CHAPTER 1

Introduction

In this report we study the control problem for a single link manipulator with joint elasticity. It has been shown that joint elasticity is the dominant source of compliance in most manipulator designs. This joint elasticity may arise from several sources, such as elasticity in the gears, belts, bearings etc. and limits speed and accuracy achievable by control algorithms designed assuming perfect rigidity at the joints.

As we shall see later on, the differential equation describing the dynamics of such a manipulator is nonlinear. Mostly in the last decade, several techniques were developed concerned with the analysis and design of feedback control laws for nonlinear systems. One of these techniques is the so-called **feedback linearization.**

The basic idea of feedback linearization is to construct a nonlinear control law as an inner loop control, which in the ideal case exactly linearizes the nonlinear system after a suitable state space change of coordinates. The designer can then design an outer loop control in the new coordinates to satisfy the traditional control design specifications.

The problem here is not only the nonlinearity of the robot dynamics (this can for example be compensated by the computed torque technique), but because of the elasticity there are more degrees of freedom to be controlled/stabilized than motor input signals. Probably, the feedback linearization technique can solve this problem too.

Before we go on with the modeling of the plant ,the design and simulation of the control system, we shall first give an analysis of the feedback linearization control technique. We shall also discuss some of the "tools" necessary to understand this technique of transforming a nonlinear system, via change of coordinates in the state space and state feedback, into a linear and controllable system.

CHAPTER 2

Some Background Theory on Feedback Linearization

In the first part of this chapter we shall discuss some "tools" necessary to get a better understanding of the feedback linearization technique. With the help of these basic concepts, like Lie-bracket, dual product, and involutivity we derive the conditions for a system to be feedback linearizable. In this way, we obtain a brief (and certainly not complete!) analysis of the feedback linearization technique.
In the latter part of this chapter we summarize the main results, which will turn out to be usefull when we design and simulate the control system in the next chapter.
But first, we shall start with a restriction to the class of nonlinear systems to which feedback linearization is applicable.

2.1 General Notions

2.1.1 Affine Nonlinear System

In the following we restrict our attention to the class of nonlinear systems of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{i=1}^{m} u_i g_i(\boldsymbol{x})$$

where $\mathbf{f}(\mathbf{x})$, $g_i(\mathbf{x}),\ldots,g_m(\mathbf{x})$ are smooth vectorfields on $\mathbf{R}^n$.
We call such a system an affine nonlinear control system.
By a smooth vectorfield on $\mathbf{R}^n$ we will mean a function $\mathbf{f}: \mathbf{R}^n \rightarrow \mathbf{R}^n$ which is infinitely differentiable. From now on, whenever we use the term function or vector field, it is assumed that the given function or vectorfield is smooth.

2.1.2 Lie - bracket

Let $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ be two vector fields on $\mathbf{R}^n$. The Lie product or Lie bracket of $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$, denoted by $[\mathbf{f},\mathbf{g}](\mathbf{x})$, is a third vectorfield defined by

$$[\boldsymbol{f},\boldsymbol{g}](\boldsymbol{x}) = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\boldsymbol{f} - \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\boldsymbol{g}$$

where

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$$

,respectively

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}$$

denote the nxn Jacobian matrices.

We also denote $[\mathbf{f},\mathbf{g}]$ as $\mathbf{ad_f(g)}$ and define $\mathbf{ad_f}^k(\mathbf{g})$ inductively by

$$\mathbf{ad_f}^k(\mathbf{g}) = [\mathbf{f},\mathbf{ad_f}^{k-1}(\mathbf{g})]$$

with

$$\mathbf{ad_f}^0(\mathbf{g}) = \mathbf{g}$$

2.1.3 Derivative of h($\mathbf{x}$) along $\mathbf{f}(\mathbf{x})$: $L_f h(\mathbf{x})$
Dual product: <**dh**,**f**>

Let h: $\mathbf{R}^n \rightarrow \mathbf{R}^n$ be a scalar function and $\mathbf{f}(\mathbf{x})$ be a vectorfield on $\mathbf{R}^n$, then the gradient of h, denoted by **dh** , is the row vector

$$grad\ h = \boldsymbol{dh} = \nabla h = [\frac{\partial h}{\partial x_1} \cdots \frac{\partial h}{\partial x_n}]$$

For a scalar function h and a vectorfield **f** the dual product of **dh** and **f** is defined as

$$\langle \boldsymbol{dh},\boldsymbol{f}\rangle = \frac{\partial h}{\partial x_1}f_1 + \cdots + \frac{\partial h}{\partial x_n}f_n$$

$$= \sum_{i=1}^{n} \frac{\partial h}{\partial x_i} f_i$$

$$= \frac{\partial h}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})$$

$$= L_f h(\boldsymbol{x})$$

This new function $L_f h(\mathbf{x})$ is sometimes called the derivative of $h(\mathbf{x})$ along $\mathbf{f}(\mathbf{x})$.

remark:

as well as in most current literature as in the trendsetting notes on nonlinear control theory by Isidori, the notation $L_f h(\mathbf{x})$ is used. However, the meaning of $<\mathbf{dh},\mathbf{f}>$ and $L_f h(\mathbf{x})$ is exactly the same.

Repeated use of this differential operation is possible:

example 1:

derivating $h(\mathbf{x})$ first along $\mathbf{f}(\mathbf{x})$ and then along $\mathbf{g}(\mathbf{x})$ yields

$$L_g L_f h(\boldsymbol{x}) = \frac{\partial (L_f h)}{\partial \boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x})$$

example 2:

derivating k times $h(\mathbf{x})$ along $\mathbf{f}(\mathbf{x})$ yields a function recursively defined as

$$L_f{}^k h(\boldsymbol{x}) = \frac{\partial (L_f{}^{k-1} h)}{\partial \boldsymbol{x}} f(\boldsymbol{x})$$

2.1.4 Dual product: $\langle \mathbf{dh}, [\mathbf{f},\mathbf{g}] \rangle$
Derivative of h(x) along [f,g]: $L_{[f,g]}h(\mathbf{x})$

Let h: $\mathbf{R}^n \rightarrow \mathbf{R}^n$ be a scalar function and $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ be vectorfields on $\mathbf{R}^n$. Then we have the following identities:

$$\langle \mathbf{dh}, [\mathbf{f},\mathbf{g}] \rangle = \langle d\langle \mathbf{dh},\mathbf{g}\rangle, \mathbf{f}\rangle - \langle d\langle \mathbf{dh},\mathbf{f}\rangle, \mathbf{g}\rangle$$

which can also be notated as

$$L_{[f,g]}h(\mathbf{x}) = L_f L_g h(\mathbf{x}) - L_g L_f h(\mathbf{x})$$

The proof of this identity is rather straightforward (see: Appendix IV - 1).

2.1.5 Frobenius theorem / Involutivity

The Frobenius theorem can be thought of as an existence theorem for solutions to certain sets of first order partial differential equations. For a thorough treatment of this theorem we refer to [4], as this subject is so delicate that it is beyond the scope of this report. So without any further discussion or proof, we state the following:

Frobenius theorem:
Let $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$ be a set of vector fields that are linearly independent at each point. Then the set of vector fields is completely integrable if and only if it is involutive.

Involutivity:
A linearly independent set of vector fields $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$ is said to be involutive if and only if there are scalar functions $a_{ijk}:\mathbf{R}^n \rightarrow \mathbf{R}$ such that

$$[\mathbf{x}_i, \mathbf{x}_j] = \sum_{k=1}^{m} a_{ijk}\mathbf{x}_k$$

For practical applications, the condition of involutivity simply means that if one forms the Lie bracket of **any** pair of vectorfields from the set $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$, then the resulting vectorfield can be expressed as a linear combination of the original vectorfields $\mathbf{x}_1,\ldots,\mathbf{x}_m$.

2.2 Feedback Linearization for Single Input Systems

2.2.1 Formal Definition of Feedback Linearization

We define the following single-input nonlinear system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u$$

where

* $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are smooth vector fields on $\mathbf{R}^n$
* $\mathbf{f}(\mathbf{0}) = \mathbf{0}$
* $u \in \mathbf{R}^n$

This system is feedback linearizable if there exists

* a region U in $\mathbf{R}^n$ containing the origin
* a diffeomorphism $T: U \rightarrow \mathbf{R}^n$
* a nonlinear feedback $u = \alpha(\mathbf{x}) + \beta(\mathbf{x})v$ with $\beta(\mathbf{x}) \neq 0$ on U

such that the transformed variables

$$\xi = \boldsymbol{T}(\boldsymbol{x})$$

satisfy the system of equations

$$\dot{\xi} = A\xi + \boldsymbol{b}v$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}$$

remarks:

* a diffeomorphism is simply a transformation whose inverse exists. Both $\mathbf{T}(\mathbf{x})$ and $\mathbf{T}^{-1}(\xi)$ are smooth mappings, that is, they have continuous partial derivatives of any order. We can think of $\mathbf{T}(\mathbf{x})$ as a nonlinear change of coordinates in the state space.

* the main ingredients of feedback linearization are coordinate changes in the state space and nonlinear feedback. The idea of feedback linearization is that one first changes to the coordinate system $\xi = \mathbf{T}(\mathbf{x})$, then there exists a nonlinear control law to cancel the nonlinearities in the system.

2.2.2 Necessary and Sufficient Conditions for Feedback Linearization

The nonlinear change of coordinates is described in the form:

$$\xi = \boldsymbol{T}(\boldsymbol{x})$$

differentiating with respect to time yields:

$$\dot{\xi} = \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} := A\xi + \boldsymbol{b}v$$

$$\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} := A\boldsymbol{T}(\boldsymbol{x}) + \boldsymbol{b}v$$

writing out of the above matrix equation gives:

$$\begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \cdots & \frac{\partial T_1}{\partial x_n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \frac{\partial T_n}{\partial x_1} & \cdots & \frac{\partial T_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \cdot & & \\ & & & \cdot & & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} T_1(\boldsymbol{x}) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ T_n(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} v$$

which is the same as

$$\begin{cases} \frac{\partial T_1}{\partial x_1}\dot{x}_1 + \cdots + \frac{\partial T_1}{\partial x_n}\dot{x}_n = T_2 \\ \quad\vdots \\ \frac{\partial T_n}{\partial x_1}\dot{x}_1 + \cdots + \frac{\partial T_n}{\partial x_n}\dot{x}_n = v \end{cases}$$

This set of equations can be renotated as (see: 2.1.3)

$$\begin{cases} \langle \boldsymbol{d}T_1, \ \dot{\boldsymbol{x}}\rangle = \langle \boldsymbol{d}T_1, \ \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u\rangle = T_2 \\ \quad\vdots \\ \langle \boldsymbol{d}T_n, \ \dot{\boldsymbol{x}}\rangle = \langle \boldsymbol{d}T_n, \ \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u\rangle = v \end{cases}$$

$$\begin{cases} \langle \boldsymbol{d}T_1, \ \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{d}T_1, \ \boldsymbol{g}(\boldsymbol{x})\rangle u = T_2 \\ \vdots \\ \langle \boldsymbol{d}T_n, \ \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{d}T_n, \ \boldsymbol{g}(\boldsymbol{x})\rangle u = v \end{cases}$$

We make the following assumptions:

1. $T_1, \ldots, T_n$ are independent of u ⇒

$$\begin{cases} \langle \boldsymbol{d}T_1, \ \boldsymbol{g}(\boldsymbol{x})\rangle = 0 \\ \vdots \\ \langle \boldsymbol{d}T_{n-1}, \ \boldsymbol{g}(\boldsymbol{x})\rangle = 0 \end{cases}$$

   which can also be notated as

$$\begin{cases} L_g T_1 = 0 \\ \vdots \\ L_g T_{n-1} = 0 \end{cases}$$

2. v is dependent of u ⇒

$$\langle \boldsymbol{d}T_n, \ \boldsymbol{g}(\boldsymbol{x})\rangle \neq 0$$

   which can also be notated as

$$L_{\boldsymbol{g}} T_n \neq 0$$

3. this yields ⇒

$$\begin{cases} \langle \boldsymbol{d}T_1, \ \boldsymbol{f}(\boldsymbol{x})\rangle = T_2 \\ \vdots \\ \langle \boldsymbol{d}T_{n-1}, \ \boldsymbol{f}(\boldsymbol{x})\rangle = T_n \\ \langle \boldsymbol{d}T_n, \ \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{d}T_n, \ \boldsymbol{g}(\boldsymbol{x})\rangle u = v \end{cases}$$

which can also be notated as

$$\begin{cases} L_{\boldsymbol{f}}T_1 = T_2 \\ \vdots \\ L_{\boldsymbol{f}}T_{n-1} = T_n \\ L_{\boldsymbol{f}}T_n + (L_{\boldsymbol{g}}T_n)u = v \end{cases}$$

The previous three conclusions can be compactly given in the following way:

$$\begin{cases} \langle \boldsymbol{d}T_1,\ \boldsymbol{ad_f}^k\boldsymbol{g}(\boldsymbol{x})\rangle = 0 \\ \vdots \\ \langle \boldsymbol{d}T_1,\ \boldsymbol{ad_f}^{n-1}\boldsymbol{g}(\boldsymbol{x})\rangle \neq 0 \\ \langle \boldsymbol{d}T_n,\ \boldsymbol{f}(\boldsymbol{x})\rangle + \langle \boldsymbol{d}T_n,\ \boldsymbol{g}(\boldsymbol{x})\rangle u = v \end{cases}$$

$$k = 0,1,\ldots,n-2$$

which can also be notated as

$$\begin{cases} L_{\boldsymbol{ad_f}^k\boldsymbol{g}}T_1 = 0 \\ \vdots \\ L_{\boldsymbol{ad_f}^{n-1}\boldsymbol{g}}T_1 \neq 0 \\ L_{\boldsymbol{f}}T_n + (L_{\boldsymbol{g}}T_n)u = v \end{cases}$$

$$k = 0,1,\ldots,n-2$$

That the above compact notation completely covers all three conclusions is proved in Appendix IV - 2.

The problem of finding a nonlinear transformation and a nonlinear control law is now reduced to the problem of solving

$$\begin{cases} \langle \mathbf{d}T_1, \ \mathbf{ad_f}^k\mathbf{g}\rangle = 0 \\ \langle \mathbf{d}T_1, \ \mathbf{ad_f}^{n-1}\mathbf{g}\rangle \neq 0 \end{cases}$$

$$k = 0,1,\ldots,n-2$$

for T_1. **If** we can find T_1 satisfying the above system of partial differential equations, then $T_2,\ldots,T_n$ are found inductively from

$$\langle \mathbf{d}T_i, \ \mathbf{f}\rangle = T_{i+1}$$

$$i = 1,\ldots,n-1$$

and the control input u is found from

$$\langle \mathbf{d}T_n, \ \mathbf{f}\rangle + \langle \mathbf{d}T_n, \ \mathbf{g}\rangle u = v \quad \Rightarrow$$

$$u = \frac{1}{\langle \mathbf{d}T_n, \ \mathbf{g}\rangle}(v - \langle \mathbf{d}T_n, \ \mathbf{f}\rangle)$$

And it is here, where we need the Frobenius theorem in order to determine whether or not the set of partial differential equations for T_1 has a solution. First we note that the vectorfields **g**, $\mathbf{ad_f}^1\mathbf{g},\ldots,\ \mathbf{ad_f}^{n-1}\mathbf{g}$ have to be linearly independent. Now by the Frobenius theorem the set

$$\langle \mathbf{d}T_1, \ \mathbf{ad_f}^k\mathbf{g}\rangle = 0$$

$$k = 0,1,\ldots,n-2$$

has a solution if and only if the set of vectorfields

$$\{ \mathbf{ad_f}^0\mathbf{g}, \ \mathbf{ad_f}^1\mathbf{g}, \ \ldots, \mathbf{ad_f}^{n-2}\mathbf{g}\}$$

is involutive.

Putting all this together we can formulate necessary and sufficient conditions for a nonlinear system to be feedback linearizable:

The nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u$$

with
* $\mathbf{f(x)}$, $\mathbf{g(x)}$ smooth vectorfields
* $\mathbf{f(0)} = \mathbf{0}$

is feedback linearizable if and only if there exists a region U containing the origin in $\mathbf{R}^n$ in which the following conditions hold

I. the vectorfields

$$\{ \boldsymbol{g},\ \boldsymbol{ad_f}^1\boldsymbol{g},\ \ldots\ ,\ \boldsymbol{ad_f}^{n-1}\boldsymbol{g} \}$$

are linearly independent in U

II. the set

$$\{ \boldsymbol{g},\ \boldsymbol{ad_f}^1\boldsymbol{g},\ \ldots\ ,\ \boldsymbol{ad_f}^{n-2}\boldsymbol{g} \}$$

is involutive in U

2.3.2 Algorithm for Feedback Linearization

STEP 1:

derive a model for the plant to be controlled and write the state equation into the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u$$

STEP 2:

the nonlinear system is feedback linearizable if

I. the vectorfields $\{\mathbf{g}, \mathbf{ad_f g}, \ldots, \mathbf{ad_f}^{n-1}\mathbf{g}\}$ are linearly independent, that is, the matrix P = [g $\mathbf{ad_f g}$ ... $\mathbf{ad_f}^{n-1}\mathbf{g}$] has full rank n. In other words $\det(P) \neq 0$

II. the set $\{\mathbf{g}, \mathbf{ad_f g}, \ldots, \mathbf{ad_f}^{n-2}\mathbf{g}\}$ is involutive. this simply means that the Lie bracket of any pair of vectorfields from the above set is a linear combination of the original vectorfields

So in order to check these two conditions first compute $\mathbf{ad_f g}$, $\mathbf{ad_f}^2\mathbf{g}, \ldots, \mathbf{ad_f}^{n-1}\mathbf{g}$ out of $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ according to the recipe in 2.1.2.

STEP 3

If the necessary and sufficient conditions of step two are satisfied, then by the Frobenius theorem there exists a solution for the partial differential equation

$$\langle \mathbf{d}T_1, \mathbf{ad_f}^k\mathbf{g} \rangle = 0$$

$$\langle \mathbf{d}T_1, \mathbf{ad_f}^{n-1}\mathbf{g} \rangle \neq 0$$

$$k = 0,1,\ldots,n-2$$

Solve this set of equations and compute

$$T_{i+1} = \langle T_i, \mathbf{f} \rangle \qquad i = 1,2,\ldots,n-1$$

inductively.

remark:

the solution for T_1 is not unique, so the transformation $\xi = \mathbf{T}(\mathbf{x})$ is not uniformly determined. However, $\mathbf{T}(\mathbf{x})$ has to be a diffeomorphism, so always compute $\mathbf{T}^{-1}(\xi)$ and check derivativity. Pay attention to the neighbourhood where the transformation is defined (local/global).

STEP 4

Compute the nonlinear control input u from the condition

$$u = \frac{1}{\langle \mathbf{d}T_n, \mathbf{g} \rangle} (v - \langle \mathbf{d}T_n, \mathbf{f} \rangle)$$

Now we have transformed the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) u$$

into a linear and controllable system

$$\dot{\xi} = A\xi + \mathbf{b}v$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}$$

For the control of this system we have many tools at our disposal.

The next blockdiagram schematically depicts the concept of feedback linearization:

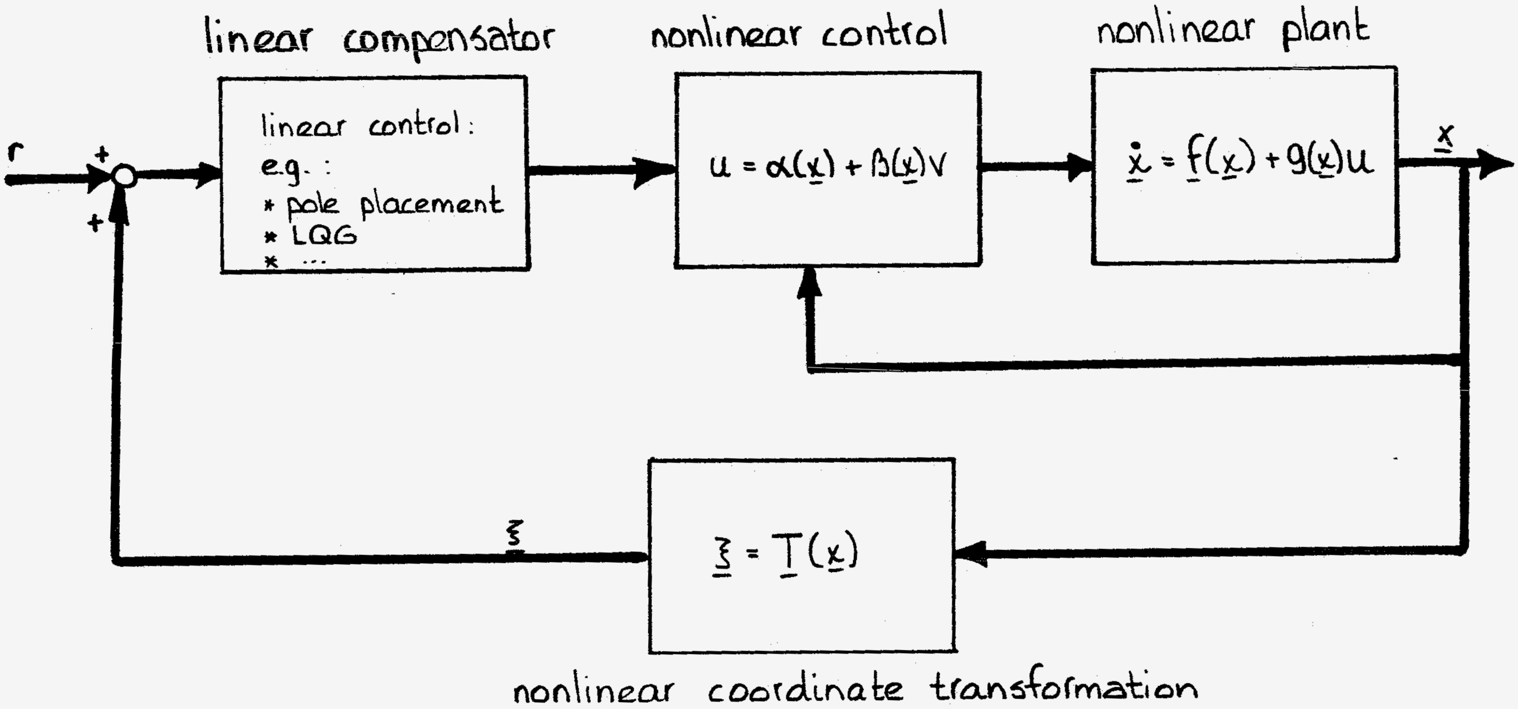


fig 2.1 block diagram for feedback linearization

In the following chapter we shall apply the concept of feedback linearization to the control of a single link manipulator with joint elasticity, according to the just described algorithm.

CHAPTER 3

Modeling, Design and Simulation

In this chapter we derive the equation of motion for a single link manipulator with joint elasticity using the Lagrange equation. We then check the conditions for feedback linearization and compute the required change of coordinates $\xi = \mathbf{T}(\mathbf{x})$.
The feedback linearizing control input u is computed and finally, the results of the simulations using MatLab are presented.

3.1 Modeling

Consider a single link manipulator with revolute joint actuated by a DC - motor. The elasticity is modeled as a torsional spring with known characteristics. For simplicity we consider a linear spring with stiffness k. As generalized coordinates we choose the link angle φ_l and the motor shaft angle φ_m; in other words, we define

$$q_1 = \varphi_l$$
$$q_2 = \varphi_m$$

with I: inertia of the motor shaft
J: inertia of the link about the axis of rotation

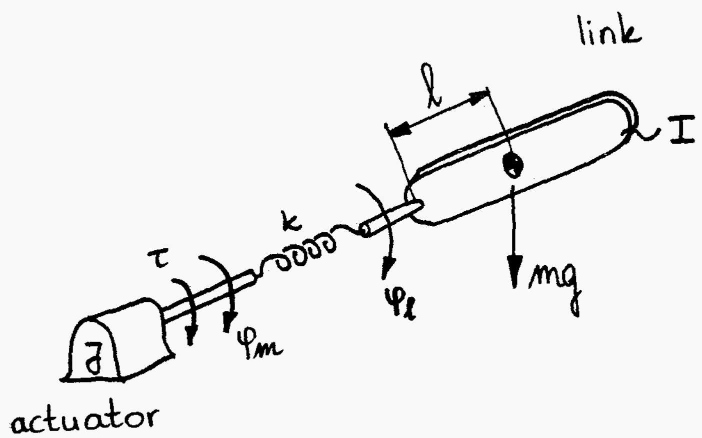


fig. 3.1 single link manipulator with joint elasticity

The total kinetic energy of the system consists of the kinetic energy of the motor and the kinetic energy of the link and is given by

$$T = \frac{1}{2} I\dot{q}_1^{\,2} + \frac{1}{2} J\dot{q}_2^{\,2}$$

The potential energy is given as

$$V = \frac{1}{2} k(q_1 - q_2)^2 + mgl(1-\cos(q_1))$$

where m: total mass of the link
l: distance from the joint axis to the link center of mass
g: gravitation

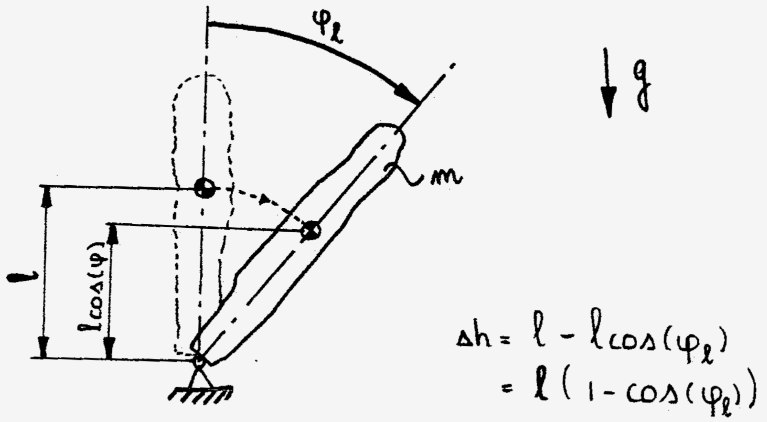


fig. 3.2 computation of the potential energy V

The Lagrangian of the system, that is, the difference of the kinetic energy and the potential energy is given by

$$L = T - V$$

$$= \frac{1}{2} I\dot{q}_1^{\,2} + \frac{1}{2} J\dot{q}_2^{\,2} - \frac{1}{2} k(q_1 - q_2)^2 - mgl(1-\cos(q_1))$$

The Lagrange equation of motion can be written in the form

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = \tau_j$$

where

$$\frac{\partial L}{\partial \dot{q}_j} = \begin{bmatrix} I\dot{q}_1 \\ J\dot{q}_2 \end{bmatrix} \quad \Rightarrow \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) = \begin{bmatrix} I\ddot{q}_1 \\ J\ddot{q}_2 \end{bmatrix}$$

$$\frac{\partial L}{\partial q_j} = \begin{bmatrix} -k(q_1-q_2) - mgl\sin(q_1) \\ k(q_1-q_2) \end{bmatrix}$$

$$\tau_j = \begin{bmatrix} 0 \\ u \end{bmatrix}$$

Substituting the above expressions in the Lagrange equation of motion yields the complete expression for the dynamics of the system. This is given by

$$\begin{cases} I\ddot{q}_1 + mgl\sin(q_1) + k(q_1-q_2) = 0 \\ J\ddot{q}_2 - k(q_1-q_2) = u \end{cases}$$

This set of second order nonlinear differential equations in the generalized coordinates can be written in state space form by setting

$$\begin{aligned} \boldsymbol{x}^T &= [x_1 \; x_2 \; x_3 \; x_4] \\ &= [q_1 \; \dot{q}_1 \; q_2 \; \dot{q}_2] \quad \Rightarrow \\ \dot{\boldsymbol{x}}^T &= [\dot{q}_1 \; \ddot{q}_1 \; \dot{q}_2 \; \ddot{q}_2] \end{aligned}$$

which yields the state equation

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_2 \\ -(\frac{mgl}{I})\sin(x_1) - (\frac{k}{I})(x_1-x_3) \\ x_4 \\ (\frac{k}{J})(x_1-x_3) + \frac{1}{J}u \end{bmatrix}$$

As we only consider continuous time affine nonlinear control systems, that is, control systems which can be written in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{i=1}^{m} g_i(\boldsymbol{x}) u_i$$

we can write for the vector fields **f(x)** and **g(x)** (in our case the system is single input)

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} x_2 \\ -(\frac{mgl}{I})\sin(x_1) - (\frac{k}{I})(x_1 - x_3) \\ x_4 \\ (\frac{k}{J})(x_1 - x_3) \end{bmatrix}$$

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix}$$

3.2 Design

3.2.1 Check for Feedback Linearizability of the System

In order to check whether or not the system

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_2 \\ -(\frac{mgl}{I})\sin(x_1) - (\frac{k}{I})(x_1-x_3) \\ x_4 \\ (\frac{k}{J})(x_1-x_3) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix} u$$

can be transformed into a linear and controllable system via state feedback and coordinates transformation, we have to compute the functions $\mathbf{ad_f g(x)}$, $\mathbf{ad_f^2 g(x)}$ and $\mathbf{ad_f^3 g(x)}$, and test if the following necessary and sufficient conditions are satisfied:

I. rank $\{\mathbf{g}, \mathbf{ad_f g}, \mathbf{ad_f^2 g}, \mathbf{ad_f^3 g}\} = 4$
II. the set $\{\mathbf{g}, \mathbf{ad_f g}, \mathbf{ad_f^2 g}\}$ is involutive

For the 4x4 - Jacobians we find

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -(\frac{mgl}{I})\cos(x_1) - (\frac{k}{I}) & 0 & (\frac{k}{I}) & 0 \\ 0 & 0 & 0 & 1 \\ (\frac{k}{J}) & 0 & -(\frac{k}{J}) & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{g}}{\partial \mathbf{x}} = O$$

and for the Lie brackets $\mathbf{ad_f^0 g}$, $\mathbf{ad_f^1 g}$, $\mathbf{ad_f^2 g}$ and $\mathbf{ad_f^3 g}$ we find

$$\boldsymbol{ad_f^0 g} = \boldsymbol{g} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix}$$

$$\mathbf{ad}_{\mathbf{f}}^{1}\mathbf{g} = [\mathbf{f}, \mathbf{g}]$$

$$= \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{f} - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{g}$$

$$= \mathbf{0} - \begin{bmatrix} * & * & * & 0 \\ * & * & * & 0 \\ * & * & * & 1 \\ * & * & * & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{J} \\ 0 \end{bmatrix}$$

$$\mathbf{ad}_{\mathbf{f}}^{2}\mathbf{g} = [\mathbf{f}, \mathbf{ad}_{\mathbf{f}}^{1}\mathbf{g}]$$

$$= \frac{\partial (\mathbf{ad}_{\mathbf{f}}^{1}\mathbf{g})}{\partial \mathbf{x}}\mathbf{f} - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{ad}_{\mathbf{f}}^{1}\mathbf{g}$$

$$= \mathbf{0} - \begin{bmatrix} * & * & 0 & * \\ * & * & \frac{k}{I} & * \\ * & * & 0 & * \\ * & * & -\frac{k}{J} & * \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{J} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{k}{IJ} \\ 0 \\ -\frac{k}{J^2} \end{bmatrix}$$

$$\mathbf{ad}_{\mathbf{f}}^{3}\mathbf{g} = [\mathbf{f}, \mathbf{ad}_{\mathbf{f}}^{2}\mathbf{g}]$$

$$= \frac{\partial (\mathbf{ad}_{\mathbf{f}}^{2}\mathbf{g})}{\partial \mathbf{x}}\mathbf{f} - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{ad}_{\mathbf{f}}^{2}\mathbf{g}$$

$$= \mathbf{0} - \begin{bmatrix} * & 1 & * & 0 \\ * & 0 & * & 0 \\ * & 0 & * & 1 \\ * & 0 & * & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{k}{IJ} \\ 0 \\ -\frac{k}{J^2} \end{bmatrix} = \begin{bmatrix} -\frac{k}{IJ} \\ 0 \\ \frac{k}{J^2} \\ 0 \end{bmatrix}$$

The matrix

$$P = [\ g \quad ad_f^1 g \quad ad_f^2 g \quad ad_f^3 g\ ]$$

$$= \begin{bmatrix} 0 & 0 & 0 & -\frac{k}{IJ} \\ 0 & 0 & \frac{k}{IJ} & 0 \\ 0 & -\frac{1}{J} & 0 & \frac{k}{J^2} \\ \frac{1}{J} & 0 & -\frac{k}{J^2} & 0 \end{bmatrix}$$

with

$$\det(P) = \frac{1}{J^2}\left(\frac{k}{IJ}\right)^2$$

$$\neq 0$$

as long as $k>0$ and $I,J<\infty$.

Thus, det(P) ≠ 0 which means that rank(P) = 4, that is, full rank and therefore condition I is satisfied.

Condition II prescribes that the set {**g**, $\mathbf{ad_f g}$, $\mathbf{ad_f^2 g}$} is involutive. Involutivity simply means that if one forms the Lie bracket of any pair of vectorfields from the set {**g**, $\mathbf{ad_f g}$, $\mathbf{ad_f^2 g}$}, then the resulting vector field can be expressed as a linear combination of the original vector fields. In our case the check for involutivity is rather simple. The vector fields **g**, $\mathbf{ad_f g}$, and $\mathbf{ad_f^2 g}$ are all constant. This means that the Lie bracket of any two members of the original set of vector fields is zero, which is trivially a linear combination of the vector fields themselves. So also condition II is satisfied.

We may end this section by stating that the system

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_2 \\ -\left(\frac{mgl}{I}\right)\sin(x_1) - \left(\frac{k}{I}\right)(x_1 - x_3) \\ x_4 \\ \left(\frac{k}{J}\right)(x_1 - x_3) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix} u$$

is feedback linearizable.

3.2.2 Computation of the Change of Coordinates

The transformation $\xi = \mathbf{T(x)}$ is computed from

$$\begin{cases} \langle dT_1, \; \mathbf{ad_f}^k \mathbf{g} \rangle = 0 \\ \langle dT_1, \; \mathbf{ad_f}^3 \mathbf{g} \rangle \neq 0 \end{cases}$$

$$k = 0,1,2$$

This can also be written as

$$\begin{cases} L_{\mathbf{ad_f}^k \mathbf{g}} T_1 = 0 \\ L_{\mathbf{ad_f}^3 \mathbf{g}} T_1 \neq 0 \end{cases}$$

$$k = 0,1,2$$

which means that for

$$k = 0 \quad \Rightarrow \quad L_{\mathbf{g}} T_1 = \langle dT_1, \; \mathbf{g} \rangle =$$

$$\begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix} = 0 \quad \Rightarrow$$

$$\frac{\partial T_1}{\partial x_4} = 0$$

$$k = 1 \quad \Rightarrow \quad L_{\mathbf{ad}_f^1 \mathbf{g}} T_1 = \langle dT_1, \; \mathbf{ad}_f^1 \mathbf{g} \rangle =$$

$$\begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{J} \\ 0 \end{bmatrix} = 0 \quad \Rightarrow$$

$$\frac{\partial T_1}{\partial x_3} = 0$$

$$k = 2 \quad \Rightarrow \quad L_{ad_f^2 g} T_1 = \langle dT_1, \; ad_f^2 g \rangle =$$

$$\begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{k}{IJ} \\ 0 \\ -\frac{k}{J^2} \end{bmatrix} = 0 \quad \Rightarrow$$

$$\frac{\partial T_1}{\partial x_2} = 0$$

$$k = 3 \quad \Rightarrow \quad L_{ad_f^3 g} T_1 = \langle dT_1, \; ad_f^3 g \rangle =$$

$$\begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} -\frac{k}{IJ} \\ 0 \\ \frac{k}{J^2} \\ 0 \end{bmatrix} \neq 0 \quad \Rightarrow$$

$$\frac{\partial T_1}{\partial x_1} \neq 0$$

We have derived the following set of partial differential equations:

$$\frac{\partial T_1}{\partial x_1} \neq 0, \qquad \frac{\partial T_1}{\partial x_2} = 0, \qquad \frac{\partial T_1}{\partial x_3} = 0, \qquad \frac{\partial T_1}{\partial x_4} = 0.$$

It is quite easy to see that T_1 can only be a function of x_1 alone, so $T_1 = T_1(x_1)$; therefore we take the simplest solution

$$T_1 = x_1$$

The other components of $\xi = \mathbf{T}(\mathbf{x})$ are computed with

$$T_{i+1} = \langle \mathbf{d}T_i, \ \mathbf{f} \rangle = L_{\mathbf{f}} T_1$$

$$i = 1,2,3$$

which means that for

$$i = 1 \Rightarrow T_2 = L_{\mathbf{f}} T_1 = \langle \mathbf{d}T_1, \ \mathbf{f} \rangle$$

$$= \begin{bmatrix} \dfrac{\partial T_1}{\partial x_1} & \dfrac{\partial T_1}{\partial x_2} & \dfrac{\partial T_1}{\partial x_3} & \dfrac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} x_2 \\ -\dfrac{mgl}{I}\sin(x_1) - \dfrac{k}{I}(x_1 - x_3) \\ x_4 \\ \dfrac{k}{J}(x_1 - x_3) \end{bmatrix}$$

$$= [\,1 \quad 0 \quad 0 \quad 0\,] \begin{bmatrix} x_2 \\ * \\ * \\ * \end{bmatrix}$$

$$= x_2$$

$$i = 2 \Rightarrow T_3 = L_{\mathbf{f}} T_2 = \langle \mathbf{d}T_2, \ \mathbf{f} \rangle$$

$$= \begin{bmatrix} \dfrac{\partial T_2}{\partial x_1} & \dfrac{\partial T_2}{\partial x_2} & \dfrac{\partial T_2}{\partial x_3} & \dfrac{\partial T_2}{\partial x_4} \end{bmatrix} \begin{bmatrix} x_2 \\ -\dfrac{mgl}{I}\sin(x_1) - \dfrac{k}{I}(x_1 - x_3) \\ x_4 \\ \dfrac{k}{J}(x_1 - x_3) \end{bmatrix}$$

$$= [\,0 \quad 1 \quad 0 \quad 0\,] \begin{bmatrix} * \\ -\dfrac{mgl}{I}\sin(x_1) - \dfrac{k}{I}(x_1 - x_3) \\ * \\ * \end{bmatrix}$$

$$= -\frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1 - x_3)$$

$$i = 3 \quad \Rightarrow \quad T_4 = L_f T_3 = \langle \mathbf{d}T_3, \mathbf{f} \rangle$$

$$= \begin{bmatrix} \dfrac{\partial T_3}{\partial x_1} & \dfrac{\partial T_3}{\partial x_2} & \dfrac{\partial T_3}{\partial x_3} & \dfrac{\partial T_3}{\partial x_4} \end{bmatrix} \begin{bmatrix} x_2 \\ -\dfrac{mgl}{I}\sin(x_1) - \dfrac{k}{I}(x_1 - x_3) \\ x_4 \\ \dfrac{k}{J}(x_1 - x_3) \end{bmatrix}$$

$$= \begin{bmatrix} -\dfrac{mgl}{I}\cos(x_1) - \dfrac{k}{I} & 0 & \dfrac{k}{I} & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ * \\ x_4 \\ * \end{bmatrix}$$

$$= -\frac{mgl}{I} x_2 \cos x_1 - \frac{k}{I} x_2 + \frac{k}{I} x_4$$

The transformation $\xi = \mathbf{T}(\mathbf{x})$ has the form

$$\xi = \begin{bmatrix} T_1(\boldsymbol{x}) \\ T_2(\boldsymbol{x}) \\ T_3(\boldsymbol{x}) \\ T_4(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ -\dfrac{mgl}{I}\sin(x_1) - \dfrac{k}{I}(x_1 - x_3) \\ -\dfrac{mgl}{I}x_2\cos(x_1) - \dfrac{k}{I}(x_2 - x_4) \end{bmatrix}$$

A straightforward calculation shows that $\mathbf{x} = \mathbf{T}^{-1}(\xi)$ is

$$\boldsymbol{x} = \begin{bmatrix} T_1^{-1}(\xi) \\ T_2^{-1}(\xi) \\ T_3^{-1}(\xi) \\ T_4^{-1}(\xi) \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_1 + \dfrac{mgl}{k}\sin(\xi_1) + \dfrac{k}{I}\xi_3 \\ \xi_2 + \dfrac{mgl}{k}\xi_2\cos(\xi_1) + \dfrac{k}{I}\xi_4 \end{bmatrix}$$

Both $\xi = \mathbf{T}(\mathbf{x})$ and $\mathbf{x} = \mathbf{T}^{-1}(\xi)$ are well defined in $\mathbf{R}^4$ and are smooth mappings, thus, T_1 defines a global diffeomorphism.

3.2.3 Computation of the Feedback Linearizing Control Input u

The nonlinear control input U is found from the condition

$$L_f T_4 + L_g T_4 u = v \quad \Leftrightarrow$$

$$<\mathbf{d}T_4,\ \mathbf{f}> + <\mathbf{d}T_4,\ \mathbf{g}>u = v \quad \Rightarrow$$

$$u = \frac{1}{<\mathbf{d}T_4,\ \mathbf{g}>}(v - <\mathbf{d}T_4,\ \mathbf{f}>)$$

We find for

$$<\mathbf{d}T_4,\ \mathbf{g}> = \begin{bmatrix} \frac{\partial T_4}{\partial x_1} & \frac{\partial T_4}{\partial x_2} & \frac{\partial T_4}{\partial x_3} & \frac{\partial T_4}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix}$$

$$= \begin{bmatrix} * & * & * & \frac{k}{I} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix} = \frac{k}{IJ}$$

and for

$$<\mathbf{d}T_4,\ \mathbf{f}> = \begin{bmatrix} \frac{\partial T_4}{\partial x_1} & \frac{\partial T_4}{\partial x_2} & \frac{\partial T_4}{\partial x_3} & \frac{\partial T_4}{\partial x_4} \end{bmatrix} \begin{bmatrix} x_2 \\ -\frac{mgl}{I}\sin(x_1) - \frac{k}{I}(x_1 - x_3) \\ x_4 \\ \frac{k}{J}(x_1 - x_3) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{mgl}{I}x_2\sin(x_1) & -\frac{mgl}{I}\cos(x_1) - \frac{k}{I} & 0 & \frac{k}{I} \end{bmatrix} \mathbf{f}$$

$$= \frac{mgl}{I}\sin(x_1)\{x_2^2 + \frac{mgl}{I}\cos(x_1) + \frac{k}{I}\} + \frac{k}{I}(x_1 - x_3)\{\frac{mgl}{I}\cos(x_1) + \frac{k}{I} + \frac{k}{J}\}$$

We may conclude:

$$u = \frac{1}{\langle \boldsymbol{d}T_4, \boldsymbol{g}\rangle}(v - \langle \boldsymbol{d}T_4,\ \boldsymbol{f}\rangle) \quad \Rightarrow$$

$$u = \frac{IJ}{k} * \{(v - \frac{mgl}{I}\sin(x_1)\{x_2{}^2+\frac{mgl}{I}\cos(x_1)+\frac{k}{I}\} - \frac{k}{I}(x_1-x_3)\{\frac{mgl}{I}\cos(x_1)+\frac{k}{I}+\frac{k}{J}\}\}$$

we can also write u as

$$u = \alpha(\boldsymbol{x}) + \beta(\boldsymbol{x})v$$

where

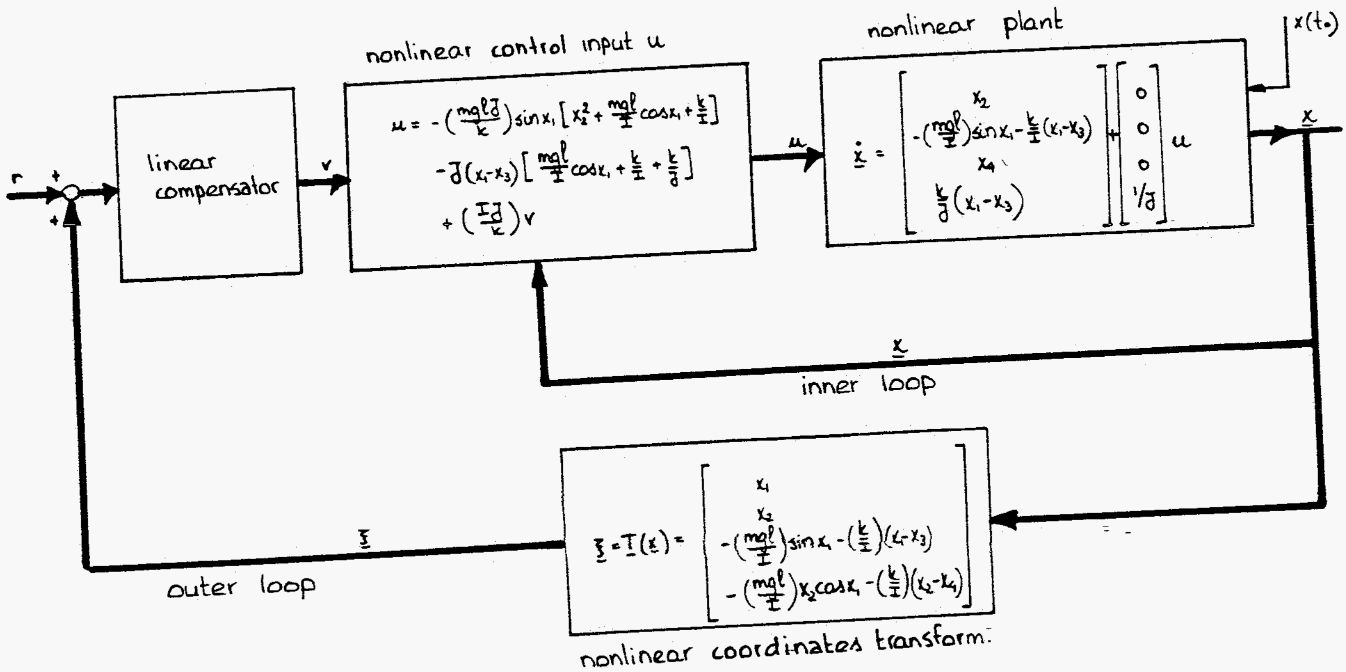
$$\alpha(\boldsymbol{x}) = -\frac{mglJ}{k}\sin(x_1)\{x_2{}^2+\frac{mgl}{I}\cos(x_1)+\frac{k}{I}\} - J(x_1-x_3)\{\frac{mgl}{I}\cos(x_1)+\frac{k}{I}+\frac{k}{J}\}$$

$$\beta(\boldsymbol{x}) = \frac{IJ}{k}$$

The form of the control law $u = \alpha(\mathbf{x})+\beta(\mathbf{x})v$ is called "regular static state feedback".

3.3 Simulation

block scheme of the plant under feedback linearization control (fig 3.3):



Parameters used for Simulation

mass	$m =$	1	[kg]
stiffness	$k =$	100	[Nm/rad]
length	$l =$	1	[m]
gravity	$g =$	9.81	[m/s^2]
inertia	$I =$	1	[kgm^2]
	$J =$	1	[kgm^2]

A simple linear control law for v designed to track a desired trajectory $y_i^d(t)$ can be expressed as

$$v = \xi_4^d - \sum_{i=1}^{4} a_i(\xi_i - \xi_i^d)$$

Applying this control law to the system

$$\dot{\xi} = A\xi + \boldsymbol{b}v$$

$$= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} v$$

yields

$$\begin{cases} \dot{\xi}_1 = \xi_2 \\ \dot{\xi}_2 = \xi_3 \\ \dot{\xi}_3 = \xi_4 \\ \dot{\xi}_4 = v = \xi_4^d - \sum_{i=1}^{4} (\xi_i - \xi_i^d) a_i \end{cases}$$

Writing out this last equation gives

$$\dot{\xi}_4 = \xi_4^d - a_1(\xi_1 - \xi_1^d) - a_2(\xi_2 - \xi_2^d) - a_3(\xi_3 - \xi_3^d) - a_4(\xi_4 - \xi_4^d)$$

$$\xi_1^{(4)} = \xi_1^{d(4)} - a_1(\xi_1 - \xi_1^d) - a_2(\dot{\xi}_1 - \dot{\xi}_1^d) - a_3(\ddot{\xi}_1 - \ddot{\xi}_1^d) - a_4(\dddot{\xi}_1 - \dddot{\xi}_1^d)$$

Defining

$$e(t) = \xi_1(t) - \xi_1{}^d(t)$$

$$= \varphi_1(t) - \varphi_1{}^d(t)$$

results in the fourth order linear error equation

$$e^{(4)} + a_4 e^{(3)} + a_3 \ddot{e} + a_2 \dot{e} + a_1 e = 0$$

This error equation is asymptotically stable if the poles are situated in the left half plane. In mathematical words: $Re(\lambda) < 0$. For computational simplicity we choose the poles to lie in $\lambda = -10$. So we obtain as characteristic equation: $(\lambda+10)^4 = 0$. Using Pascal's triangle gives

$$\lambda^4 + 40\lambda^3 + 600\lambda^2 + 4000\lambda + 10000 = 0 \quad \Rightarrow$$

$$\begin{aligned} a_1 &= 10000 \\ a_2 &= 4000 \\ a_3 &= 600 \\ a_4 &= 40 \end{aligned}$$

So with the control law

$$v = \xi_4{}^d - 10000(\xi_1-\xi_1{}^d) - 4000(\xi_2-\xi_2{}^d) - 600(\xi_3-\xi_3{}^d) - 40(\xi_4-\xi_4{}^d)$$

the link position will closely track a desired trajectory.
We choose the desired trajectory to be

$$\begin{aligned} \xi_1{}^d &= \sin(8t) \\ \xi_2{}^d &= 8\cos(8t) \\ \xi_3{}^d &= -64\sin(8t) \\ \xi_4{}^d &= -512\cos(8t) \\ \xi_4{}^d &= 4096\sin(8t) \end{aligned}$$

Now, the control law v is completely determined:

$$v = 4096\sin(8t) - 10000(\xi_1-\sin(8t)) - 4000(\xi_2-8\cos(8t))$$

$$- 600(\xi_3+64\sin(8t) - 40(\xi_4+512\cos(8t))$$

Further, also the complete blockscheme is determined and is given on the next page (fig 3.4). The computerprogram used for simulating the plant under control is given in Appendix III.

Block scheme for the plant under feedback linearization control (fig 3.4):

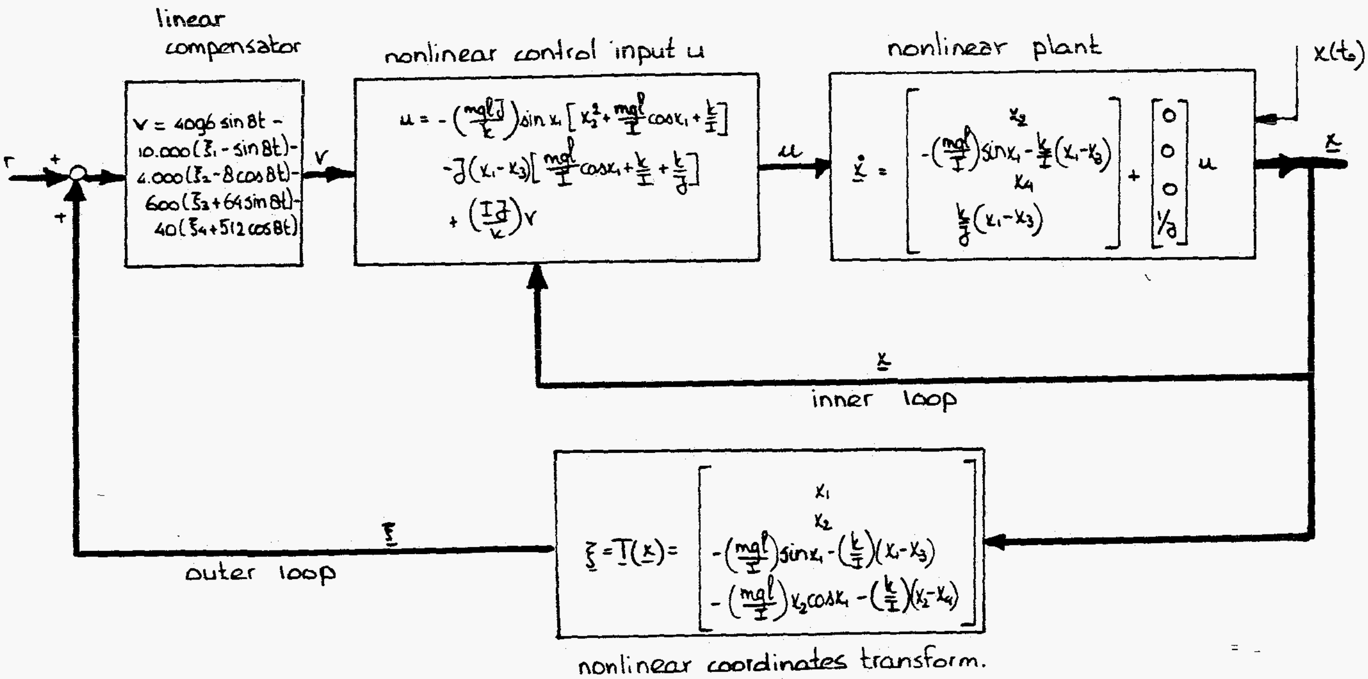


Figure 3.5 shows the result of the simulation. It clearly depicts that after some time the link angle closely tracks its desired trajectory.

We couls ask ourselves what happens if we mistakenly assume that the motorshaft is rigid. In other words, what is the response of the flexible joint system to a rigid control law, that is, a control law designed under the assumption that the motorshaft is rigid. This control law will be computed in the next chapter.

Figure 3.6a shows the result if we replace the stiffness of the spring in the control law u by a stiffness which is 1.5 times its original value. We clearly see that the link angle does not track its desired trajectory.

Figure 3.6b depicts the result when in the control law u the stiffness of the spring has been replaced by 1.75 its original value. Notice that the amplitude grows with time. The system is unstable!

Figure 3.6c clearly displays the unstable character of the system under control. The stiffness of the spring is 2.5 times its original value. The simulation had to be cut off at t = 1.6 [s].

We may conclude that if we design a control law and make the wrong assumption, namely that the stiffness of the spring is larger than in reality, the plant becomes unstable. However, in [1] Spong computes a rigid control law (see form. 5.3: App. I) which he applies to a flexible joint system and presents that result (see fig. 3: App. I). Besides the fact that this control law is wrong (see the calculations carried out in section 4.1.1), our simulation with this control law completely fails! We would like to leave this point open for discussion.

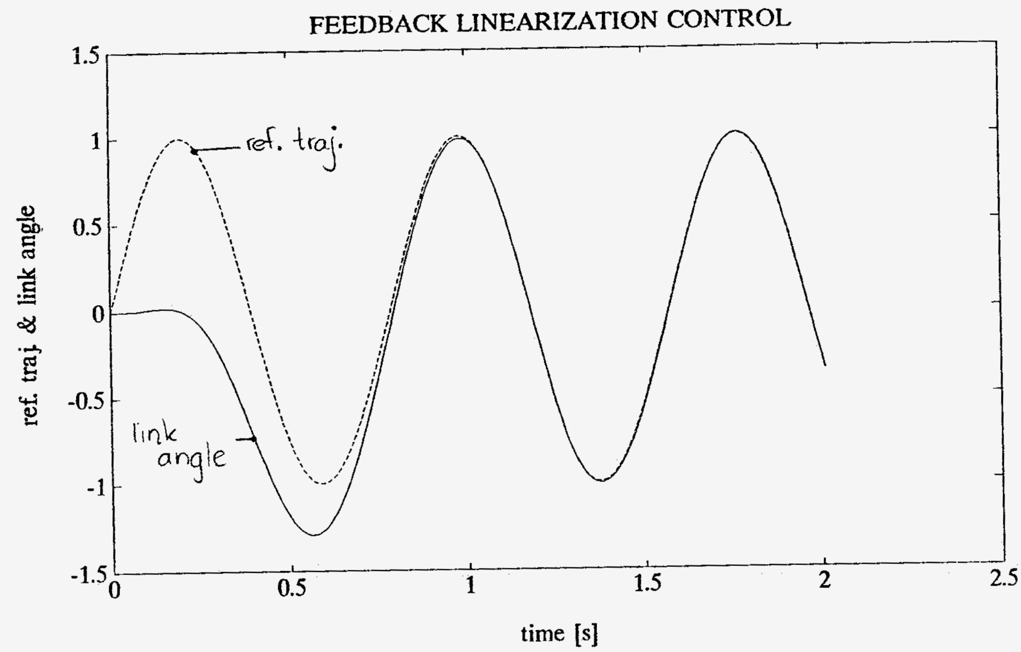


figure 3.5: reference trajectory and link angle
k model = 100
k control law = 100

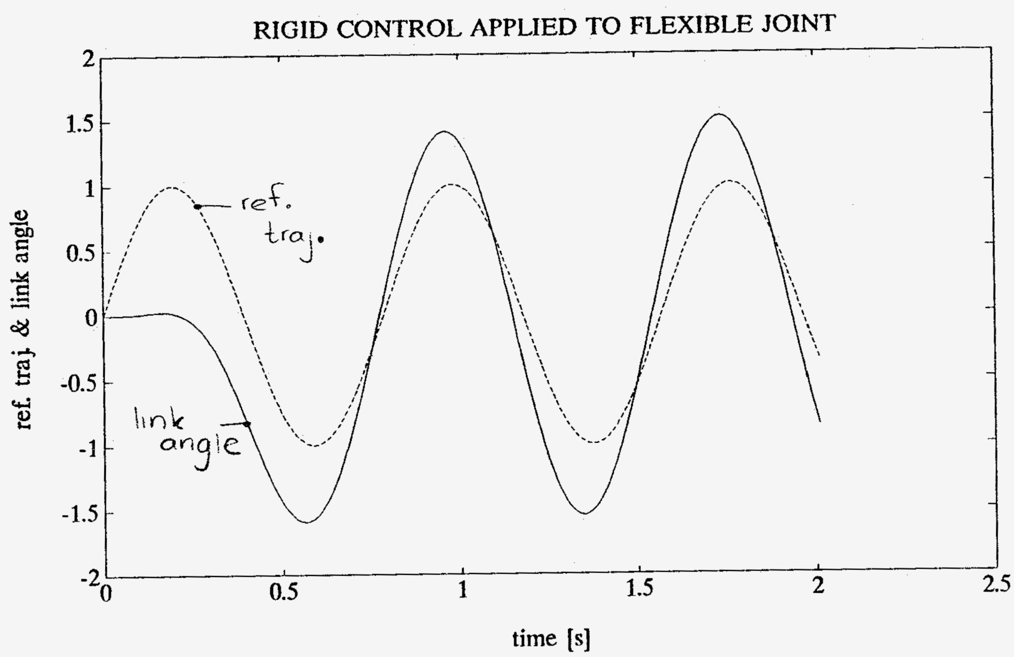


figure 3.6a: reference trajectory and link angle
k model = 100
k control law = 150

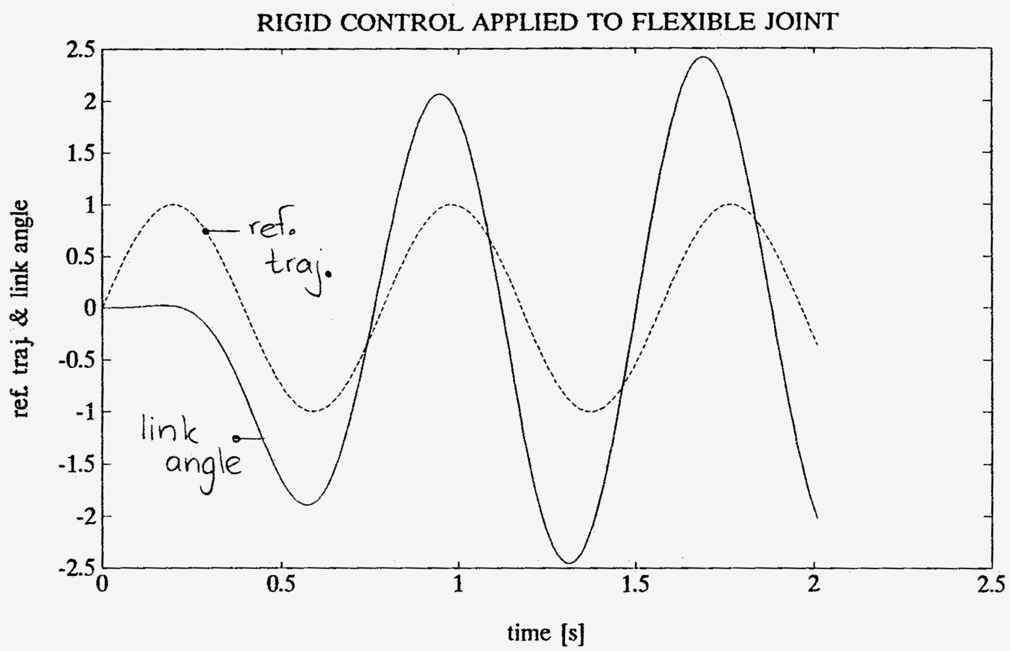


figure 3.6b: reference trajectory and link angle
k model = 100
k control law = 175

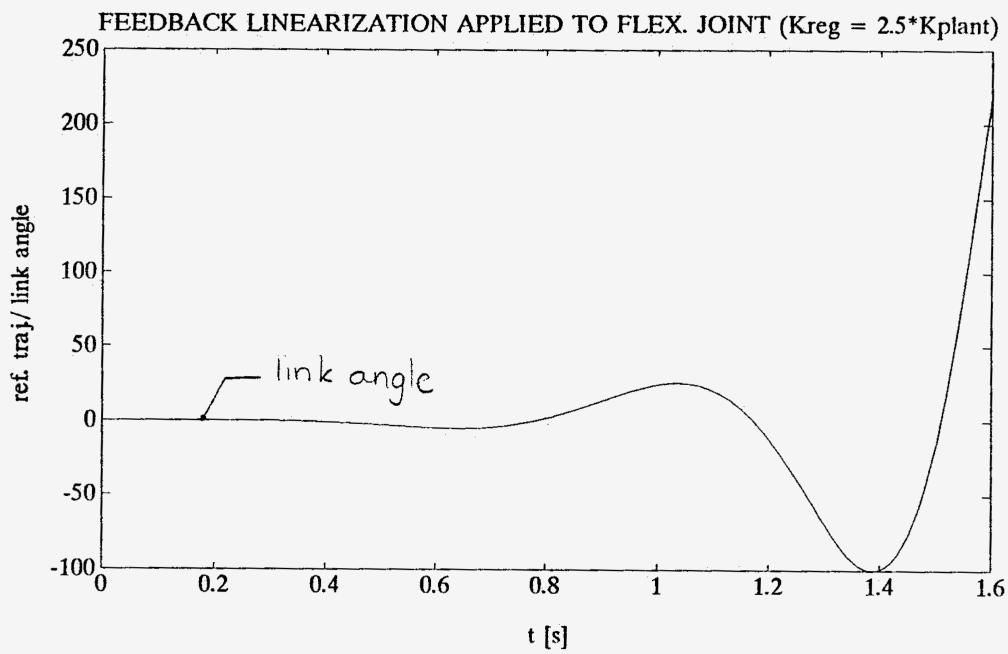


figure 3.6c: reference trajectory and link angle
k model = 100
k control law = 250

CHAPTER 4

Further Developments, Future Developments and Restrictions

In the previous chapters we analysed the feedback linearization control methodology and applied the technique to a simple example. The choice of this example was not arbitrary. On purpose, we used the same model as Spong did in [1] and for some simulations we succeeded in obtaining the same results.
However, this may not be considered as a formal proof for the correctness of our computerprogram, but we may assume that the followed strategy is right. With this model as a base we made some further extensions. First we simplified the model by assuming the stiffness of the spring to be infinitely large. But as we carried out the computations, we noticed that the equations turned out to be relatively simple. Then, the idea arose to use this example for an assignment meant for graduate students in mechanical control engineering. So with this idea in mind, we added a blueprint for a possible assignment which can be used during the course "werktuigkundig regelen IV". We hope it will serve the purpose it is meant for. Next, we extended the original model by adding damping and a nonlinear spring. Furthermore, in order to avoid the assumption that feedback linearization is some sort of remedy for tackling any kind of nonlinear control problem, we discuss some restrictions and disadvantages of the technique.

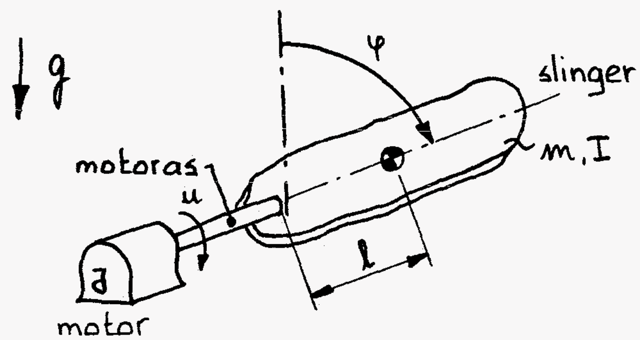
4.1 Further Developments

4.1.1 An Educational Example

This part of our report slightly differs from the rest. When we evaluated the mechanical system from chapter 3 in the case of a rigid motorshaft(that is $k \rightarrow \infty$) we noticed that the equations turned out to be relatively simple. So the application of feedback linearization control theory in this example could well serve as an assignment in the course "werktuigkundig regelen IV", because it highlights some basic, but important notions in nonlinear control theory (like for example Lie-bracket, static state feedback and the differential operation $L_f\lambda(\mathbf{x})$) without being computationally extensive. So, the next part in our report describes (in Dutch) a possible blueprint for such an assignment. We hope it will be usefull.

SYSTEEMBESCHRIJVING

Beschouw het volgende mechanische systeem:



Het systeem wordt als volgt gemodelleerd. Een motor kan via een oneindig stijve motoras een koppel u [Nm] uitoefenen op een slinger. Dit koppel is de ingang van het systeem. Het massatraagheidsmoment van de as bedraagt J [kgm^2]. De slinger is star, heeft massa m [kg] en massatraagheidsmoment I [kgm^2] ten opzichte van de as door het rotatiepunt. De afstand van het rotatiepunt tot het zwaartepunt van de slinger bedraagt l [m]. De rotatiebeweging van de slinger kan uitsluitend plaatsvinden in het verticale vlak. De positie van de slinger wordt vastgelegd met de hoek q tussen de verticaal en de hartlijn van de slinger. Deze hoek q is de te regelen grootheid. Zijn gewenste trajectorie wordt gegeven door $q^d(t) = \sin(8t)$. De hoek q(t) en zijn fluxie d/dt(q) worden op ieder tijdstip en zonder significante fouten gemeten.

FEEDBACK LINEARISERING

Doel van dit praktikum is het bovenstaande mechanische systeem zodanig te regelen dat de beweging van de slinger zijn gewenste trajectorie zo goed mogelijk volgt. Om dit doel te bereiken wordt een techniek toegepast, genaamd "feedback linearization control". De basisgedachte achter deze techniek is door via een (niet-lineaire) transformatie op de toestand en een (niet-lineaire) regelingang u het niet-lineaire systeem te transformeren naar een lineair en regelbaar systeem. Met andere woorden:

Ontwerp een transformatie op de toestand $\xi = \mathbf{T}(\mathbf{x})$, en een terugkoppelwet $u = \alpha(\mathbf{x})+\beta(\mathbf{x})v$ zodanig dat het niet-lineaire systeem

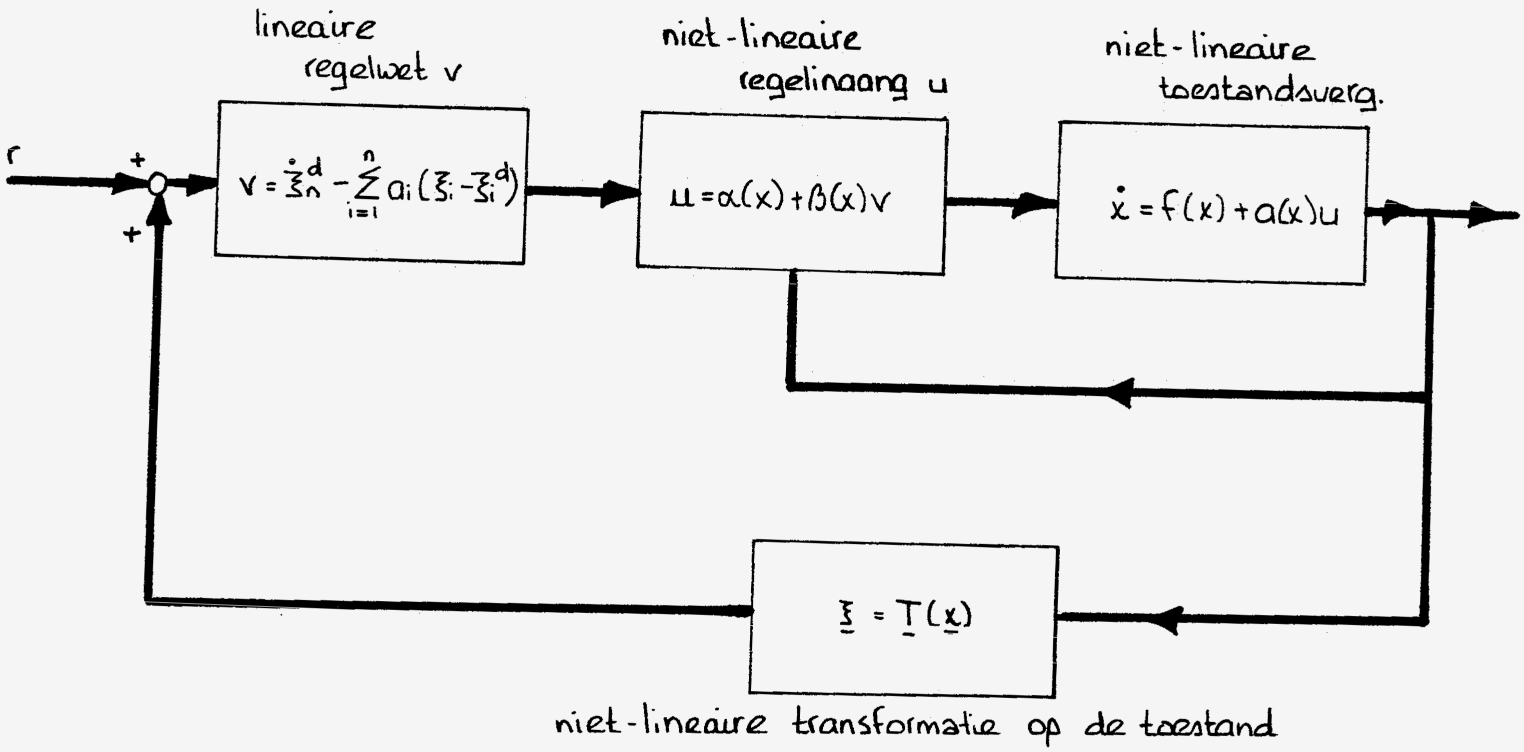
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u$$

wordt omgezet in het lineaire en regelbare systeem:

$$\dot{\xi} = A\xi + \boldsymbol{b}v$$

$$= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} v$$

Voor de regeling van dit systeem hebben we verschillende "gereedschappen" tot onze beschikking. Het bovenstaande wordt schematisch als volgt weergegeven:



VRAGEN EN OPDRACHTEN

1. Bestudeer § 10.1 t/m § 10.4 uit het diktaat "Lectures on advanced control" en de beschrijving van de MatLab tools **ode23/ode45.**

2. Stel de bewegingsvergelijking op van het systeem en herschrijf deze vergelijking naar de toestandsvorm

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u$$

3. Controleer of het systeem "feedback linearizable" is.

4. Bereken de transformatie op de toestandsvector $\xi = \mathbf{T}(\mathbf{x})$, bepaal tevens de inverse en ga na of beide voldoende vaak differentieerbaar zijn.

5. Bereken de regelingang $u = \alpha(\mathbf{x})+\beta(\mathbf{x})v$

6. Verwerk de resultaten van de bovenstaande opgaven in een blokschema. Maak hierbij gebruik van het schema op de vorige bladzijde.

7. De regelwet voor v heeft de vorm

$$v = \xi_2^{\,d} - a_1(\xi_1-\xi_1^{\,d}) - a_2(\xi_2-\xi_2^{\,d})$$

Definieer als volgfout: $e(t) = \xi_1(t)-\xi_1^{\,d}(t)$
Ga uit van de vergelijking:

$$\dot{\xi} = A\xi + \boldsymbol{b}v$$

en stel de foutvergelijking op in de vorm van:

$$\ddot{e} + a_2\dot{e} + a_1 e = \ldots$$

Wanneer is deze dv asymptotisch stabiel? Kies voor het gemak de wortels van de karakteristieke vergelijking in het punt -10 en bereken de waarden van a_1 en a_2. Kies als gewenste trajectorie $z_{1d}(t)= \sin(8t)$. Hoe luidt v?

8. Simuleer het gedrag van het systeem met MatLab.
Geef in één figuur weer als functie van t:
* gewenste trajectorie $z_1^d(t)$
* hoekverdraaiing van de slinger $x_1(t)$

9. Stel dat we de aanname doen dat de hoekverdraaiing van de slinger een kleine uitwijking is rondom de statische evenwichtsstand $\varphi = 0$. Met andere woorden, we ontwikkelen de niet-lineaire termen in een Taylorreeks en beschouwen uitsluitend de 0^e en de 1^e orde term $\Rightarrow$
$\sin(\varphi) = \varphi$
$\cos(\varphi) = 1$
Hoe luidt de regelwet u?
Gebruik nu deze regelwet voor het regelen van het niet-lineaire systeem. We zijn dus in feite een niet-lineair systeem aan het regelen in de veronderstelling dat het lineair is!
Geef in één figuur weer als functie van t:
* gewenste trajectorie $z_1^d(t)$
* hoekverdraaing van de slinger $x_1(t)$

DE ANTWOORDEN OP DE VRAGEN ZIJN UITGEWERKT IN APPENDIX IV - 3

4.1.2 Adding Damping and a Nonlinear Spring

For the state equation of the single link manipulator with joint elasticity we derived:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_2 \\ -(\frac{mgl}{I})\sin(x_1) - (\frac{k}{I})(x_1-x_3) \\ x_4 \\ (\frac{k}{J})(x_1-x_3) + \frac{1}{J}u \end{bmatrix}$$

Notice that the nonlineariry in the equation is given by the potential force $-(mgl/I)\sin(x_1)$; it is significant if the robot arm is to be controlled for **large** displacements from its nominal trajectory.

Another significant nonlinearity in the system will arise if the elastic force due to the spring is modeled by a nonlinear function $\psi = \psi(x_1-x_3)$. A typical example is:

$$\psi = k_1(x_1-x_3) + k_2(x_1-x_3)^3$$

The Lagrange equation gets in this case the form:

$$\begin{cases} I\ddot{q}_1 + mgl\sin(q_1) + k_1(q_1-q_2) + k_2(q_1-q_2)^3 = 0 \\ J\ddot{q}_2 - k_1(q_1-q_2) - k_2(q_1-q_2)^3 = u \end{cases}$$

or in state space form:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_2 \\ -(\frac{mgl}{I})\sin(x_1) - (\frac{k_1}{I})(x_1-x_3) - (\frac{k_2}{I})(x_1-x_3)^3 \\ x_4 \\ (\frac{k_1}{J})(x_1-x_3) + (\frac{k_2}{J})(x_1-x_3)^3 + \frac{1}{J}u \end{bmatrix}$$

Up to now we neglected damping in the system. If we add the damping torques $b_l\dot{q}_1$ and $b_m\dot{q}_2$ the Lagrange equation of motion is given by

$$\begin{cases} I\ddot{q}_1 + b_l\dot{q}_1 + mgl\sin(q_1) + k(q_1-q_2) = 0 \\ J\ddot{q}_2 + b_m\dot{q}_2 - k(q_1-q_2) = u \end{cases}$$

or in state space form:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_2 \\ -(\frac{b_l}{I})x_2 - (\frac{mgl}{I})\sin(x_1) - (\frac{k}{I})(x_1-x_3) \\ x_4 \\ -(\frac{b_m}{J})x_4 + (\frac{k}{J})(x_1-x_3) + \frac{1}{J}u \end{bmatrix}$$

The following question now arises, and that is, is this system feedback linearizable and if so, compute the nonlinear change of coordinates and the nonlinear control input u which transforms the original system into a linear and controllable system of the form

$$\dot{\xi} = A\xi + \boldsymbol{b}v$$

$$= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \cdot & & \\ & & & \cdot & & \\ & & & \cdot & & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} v$$

Here we only present the results of the simulation for the above system:

- figure 4.5: reference trajectory and link angle for the damped system
- figure 4.6: control input u for the damped system
- figure 3.5: reference trajectory and link angle for the undamped system
- figure 4.7: control input u for the undamped system

For computational details we refer to Appendix II.

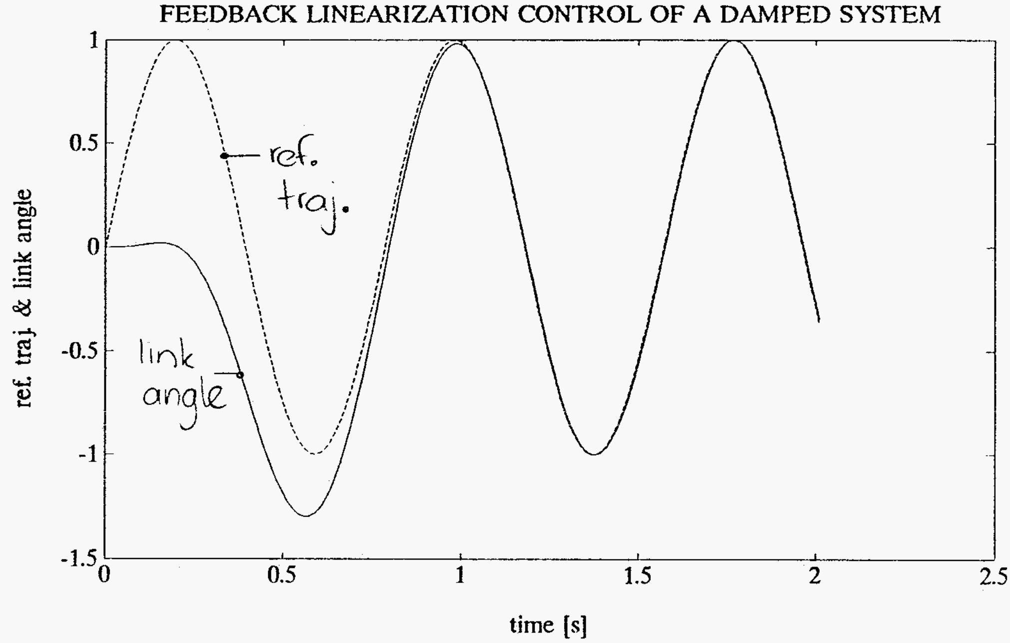


figure 4.5: ref. traj. and link angle
damped system

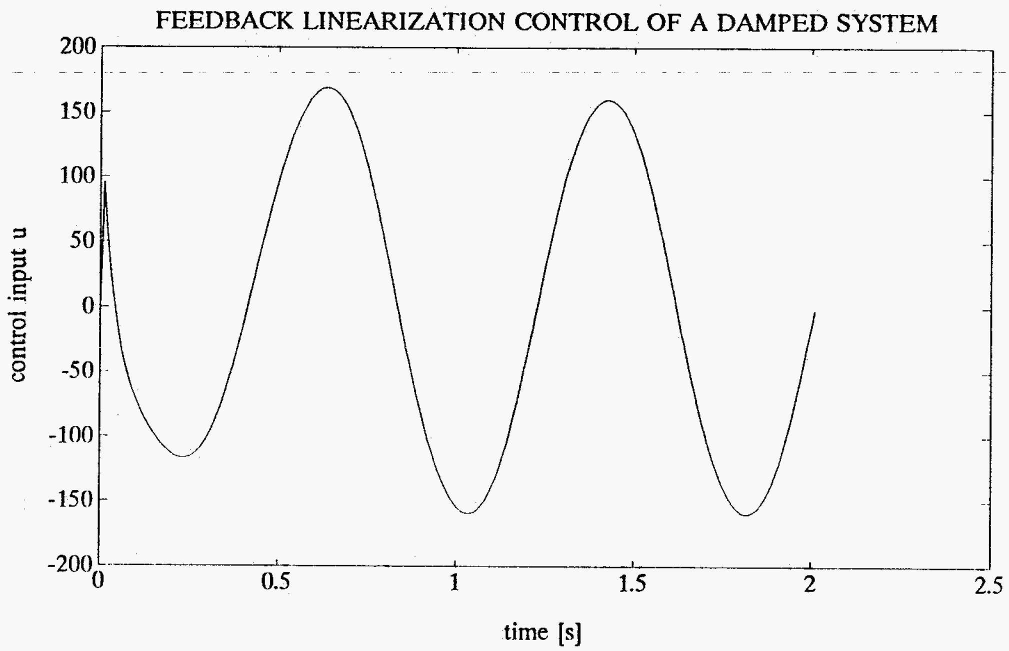


figure 4.6: control input u
damped system

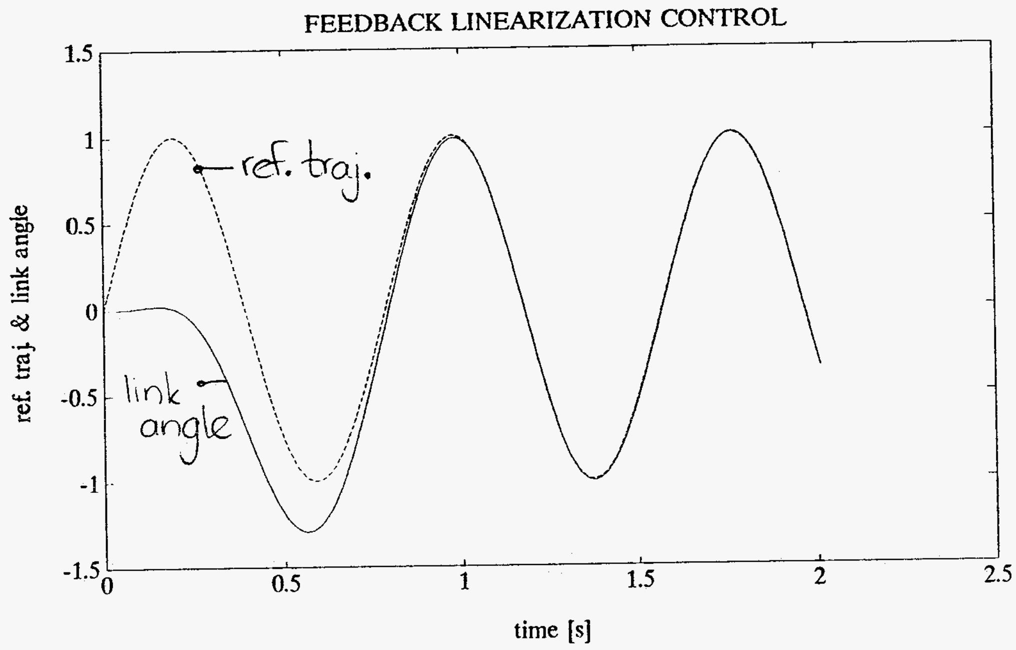


figure 3.5: ref. traj. and link angle
undamped system

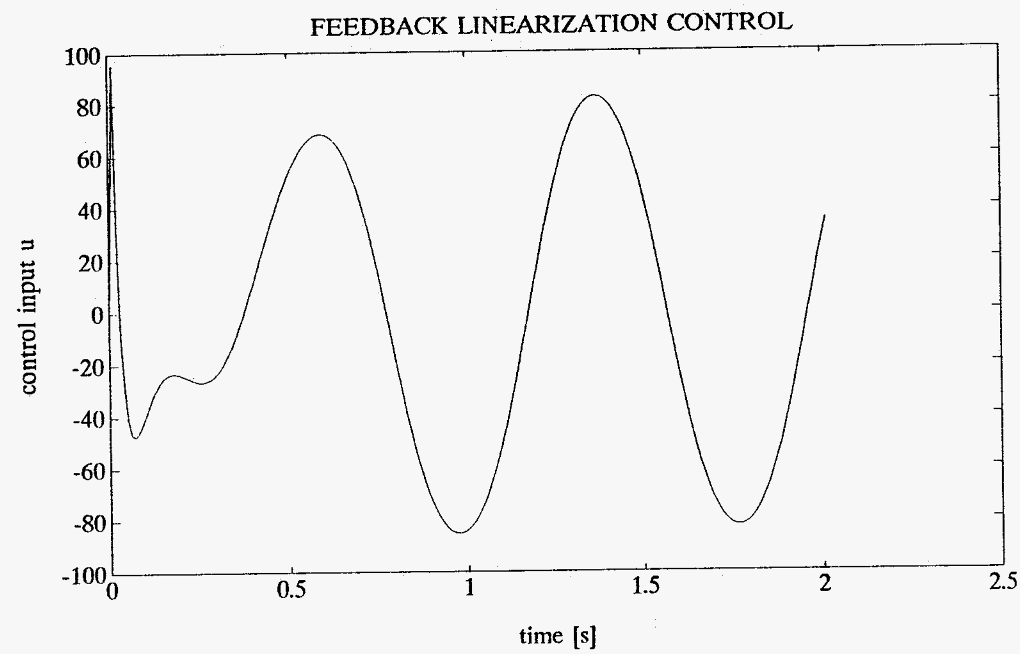


figure 4.7: control input u
undamped system

4.2 Future Developments

In this report we only studied feedback linearization applied to the **state equation**. The total set of equations describing an affine nonlinear control system is given by

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\,u \\ \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \end{cases}$$

One might pose the question of when there exists a feedback and a change of coordinates, transforming the entire description of the system, output function included, into a linear and controllable one. Necessary and sufficient conditions are given in [4].

In this report we only studied a single link manipulator. In the general case of a n-link manipulator the dynamic equations represent a multi-input nonlinear system. The conditions for feedback linearization of multi-input systems are more difficult to state, but the conceptual idea is the same as in the single input case. Derivation of the equations of motion for a n-linked manipulator are given in [3]. However, they neglect damping and use a linear spring. Theory on multi-input multi-output nonlinear systems is given in [5].

4.3 Restrictions and Disadvantages

Up to now we only discussed the possibilities of feedback linearization. The technique of feedback linearization is important in that it leads to a control design methodology for some classes of nonlinear systems. It is most certainly not a universal remedy for tackling any nonlinear control problem.

The system to be feedback linearized must be an affine nonlinear system, that is, the system may be nonlinear in the state, but must be linear in the input. The output equation may be nonlinear:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{i=1}^{m} g_i(\boldsymbol{x})\,u_i \\ y_i = h_i(\boldsymbol{x}) \end{cases}$$

Computation of the coordinates transformation requires solving a set of partial differential equations. Although the Frobenius theorem only speaks of the **existence** of a solution, it does not provide us with that solution. And even if we have it, we still have to compute its inverse. All this may not be as straightforward as it is in our examples.

Feedback linearization control requires full state measurement which in our case are positions and velocities of both the link and the motorshaft. Whenever part of the state is not accesible for measurement problems arise, because theory on nonlinear observers is limited and the nonlinear version of the "separation principle" is not (yet) known.

The feedback linearization technique is computationally expensive in general and requires accurate modeling.

REFERENCES

[1] Spong, M.W.
"Modeling and Control of Elastic Joint Robots"
Trans. ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 109, pp. 310 - 319, Dec. 1987.

[2] Marino, R.W., and Spong, M.W.
"Nonlinear Control Techniques for Flexible Joint Manipulators: A Single Link Case Study",
Proc. 1986 IEEE Conference on Robotics and Automation, San Francisco, pp. 1030 - 1036, April 1986

[3] Spong, M.W., Vidyasagar, M.
Robot Dynamics and Control
John Wiley & Sons, Inc, 1989

[4] H. Nijmeijer, A.J. van der Schaft
Nonlinear Dynamical Control Systems
Springer Verlag New York Inc, 1990

[5] Isidori, A.
Lectures on Nonlinear Control
Notes prepared for a course at the Carl Cranz Gesellschaft
3 - 6 August 1987

[6] Su, R.
"On the Linear Equivalence of Nonlinear Control Systems"
Systems and Control Letters, 2, 48 - 52, 1982

APPENDIX I

Modeling and Control of Elastic Joint Robots

M. W. Spong
Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign,
Urbana, Ill. 61801

In this paper we study the modeling and control of robot manipulators with elastic joints. We first derive a simple model to represent the dynamics of elastic joint manipulators. The model is derived under two assumptions regarding dynamic coupling between the actuators and the links, and is useful for cases where the elasticity in the joints is of greater significance than gyroscopic interactions between the motors and links. In the limit as the joint stiffness tends to infinity, our model reduces to the usual rigid model found in the literature, showing the reasonableness of our modeling assumptions. We show that our model is significantly more tractable with regard to controller design than previous nonlinear models that have been used to model elastic joint manipulators. Specifically, the nonlinear equations of motion that we derive are shown to be globally linearizable by diffeomorphic coordinate transformation and nonlinear static state feedback, a result that does not hold for previously derived models of elastic joint manipulators. We also detail an alternate approach to nonlinear control based on a singular perturbation formulation of the equations of motion and the concept of integral manifold. We show that by a suitable nonlinear feedback, the manifold in state space which describes the dynamics of the rigid manipulator, that is, the manipulator without joint elasticity, can be made invariant under solutions of the elastic joint system. The implications of this result for the control of elastic joint robots are discussed.

1 Introduction

The proper choice of mathematical models for control system design is a crucial stage in the development of control strategies for any system. This is particularly true for robot manipulators due to their complicated dynamics. For simulation purposes one would like as detailed a model as possible, while for control design and implementation one would like to retain only the most significant dynamic effects in the model in order to simplify the analysis and minimize on-line computational requirements.

Because of the extreme complexity of the dynamic equations of motion for n-link manipulators with joint elasticity, most existing results on the control of such manipulators have relied either on computer programs to generate the equations [17] or have treated special configurations [26] and/or single link examples [27]. However, one generally obtains relatively little insight from symbolically generated equations, and an understanding of the physics underlying the model is of prime importance in understanding the control problem. For this reason we first investigate the problem of modeling the dynamics of elastic joint manipulators. For notational simplicity we treat the case of revolute joints driven by DC-motors whose rotors are elastically coupled to the links. It turns out that by making two rather simple approximating assumptions it is possible to derive a model of the system that is much more amenable to analysis and control than previous models.

Specifically, we assume

(A1) That the kinetic energy of the rotor is due mainly to its own rotation. Equivalently, the motion of the rotor is a pure rotation with respect to an inertial frame. We further assume.

(A2) The the rotor/gear inertia is symmetric about the rotor axis of rotation so that the gravitational potential of the system and also the velocity of the rotor center of mass are both independent of the rotor position.

Assumption (A2) hardly needs any justification and Assumption (A1) is easy to justify for a large class of robots, since roughly speaking it amounts to neglecting terms of order at most $1/m$ where m:1 is the gear ratio. In fact, most existing models of rigid manipulators are derived under precisely these same assumptions; see for example Paul [4], equation (6.49). The important point is to model the dynamic effects which are dominant, in this case the joint elasticity.

2 Modeling

We now consider an n-link manipulator with revolute joints actuated by DC-motors, and model the elasticity of the ith joint as a linear torsional spring with stiffness k_i. For notational simplicity we take $k_i \equiv k$ for all i. Because of the additional degrees of freedom introduced by the elastic coupling of the motor shaft to the links we model the rotor of each actuator as a "fictitious link," that is, as an additional rigid body in the chain with its own inertia. Thus the manipulator consists of n "actual" links and n "fictitious" or rotor links.

[1] This research was partially supported by the National Science Foundation under grant DMC-8616091

Contributed by the Dynamic Systems and Control Division for publication in the JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL. Manuscript received by the Dynamic Systems and Control Division July 1986.

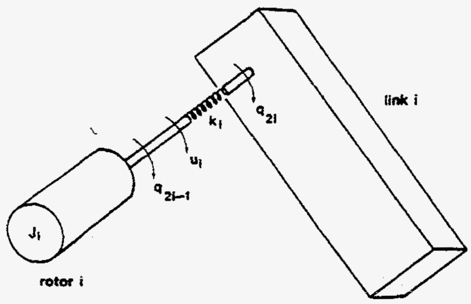


Fig. 1 Elastic Joint

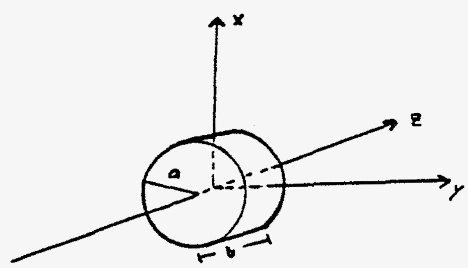


Fig. 2

The specific design of the manipulator will dictate the manner in which the actuators are coupled to the links. For simplicity we discuss the case in which the rotor is directly coupled to the link that it actuates as shown in Fig. 1. Other configurations, for example when the motors are located on link 1 and drive the distal links through cables, etc., can be handled by finding the corresponding transformation between "actuator space" and "joint space" as in [34]. The details are omitted.

Referring to Fig. 1, let $\mathbf{q}=(q_1, \ldots q_{2n})^T$ be a set of generalized of generalized coordinates for the system where

$$q_{2i} = \text{the angle of link } i,\ i=1, \ldots, n \tag{2.1}$$

$$q_{2i-1} = -\frac{1}{m_i}\theta_i,\ i=1, \ldots, n \tag{2.2}$$

where θ_i is the angular displacement of rotor i and m_i is the gear ratio. In this case then $q_{2i}-q_{2i-1}$ is the elastic displacement of link i.

Lagrangian Dynamics The rotor, as an intermediate link, now has its own coordinate frame and inertia tensor associated with it. We shall model the "rotor" link as a right circular cylinder of radius a and length b. From symmetry consideration we may establish the coordinate frame at the center of mass and assume that coordinate axes are principal axes of the cylinder, with the rotor angle θ_i measured about the z_{2i} axis. The inertia tensor of the rotor is then given by

$$I_i = \begin{bmatrix} I_{xx_i} & 0 & 0 \\ 0 & I_{yy_i} & 0 \\ 0 & 0 & I_{zz_i} \end{bmatrix} \tag{2.3}$$

where I_{xx}, I_{yy}, I_{zz} are the moments of inertia of the rotor about the principal axes. The kinetic energy of the rotor is

$$K_{r_i} = \frac{1}{2}M_i v_i^T v_i + \frac{1}{2}\,\omega_i^T I_i \omega_i \tag{2.4}$$

where v_i represents the velocity of the center of mass of the rotor, M_i is the rotor mass, and ω_i is the vector of angular velocities about the principal axes.

Now by the symmetry assumption (A2) the velocity v_i of the center of mass of the rotor can written as a function only of the link variables $q_2, \ldots, q_{2i-2}$. If we therefore include the rotor mass as part of link $2i-2$ for the purposes of calculating the inertia tensor of link $2i-2$ then the first term in (2.4) will be included with the kinetic energy of link $2i-2$.

We now invoke Assumption (A1) and model only the kinetic energy of the rotor about its principal axis of rotation, i.e., we assume that the second term in (2.4) above is given as

$$\frac{1}{2}\omega_i^T I_i \omega_i = \frac{1}{2} I_{zz_i}\dot{\theta}_i^2 \tag{2.5}$$

$$= \frac{1}{2} I_{zz_i} m_i^2 \dot{q}_{2i-1}^2$$

The following example gives a simple illustration of the effect of the above assumption.

Example Consider the cylinder shown in Fig. 2. The rotational kinetic energy is then

$$K = \frac{1}{2}(I_{xx}\omega_x^2 + I_{yy}\omega_y^2 + I_{zz}\omega_z^2) \tag{2.6}$$

The principal moments of inertia of the cylinder with respect to the coordinate system shown are given by

$$I_{xx} = \frac{1}{4}Mb^2 = I_{yy} \tag{2.7}$$

$$I_{zz} = \frac{1}{2}Ma^2 \tag{2.8}$$

Due to the gear ratio m:1 the angular velocity ω_z will generally be a factor of m larger than the angular velocities about the other two axes. If we take therefore $\omega_z = m\omega_x = m\omega_y$ for the purposes of illustration, the kinetic energy becomes

$$K = \frac{1}{4}M\omega_z^2(a^2 + b^2/m^2) \tag{2.9}$$

We now approximate according to (A1) the kinetic energy K as

$$\bar{K} = \frac{1}{2}M\omega_z^2 a^2 \tag{2.10}$$

The percent error in the kinetic energy incurred by using the expression (2.10) instead of the true kinetic energy (2.9) is then

$$Error = \frac{K-\bar{K}}{K} \times 100 \tag{2.11}$$

$$= \frac{b^2}{b^2 + m^2a^2} \times 100 \tag{2.12}$$

For example if $a=1$, $b=1/2$, and $m=100$, the percent error in kinetic energy is 0.01 percent.

Let us now partition the generalized coordinate vector $\mathbf{q}$ as $(\mathbf{q}_1, \mathbf{q}_2)^T$ where

$$\mathbf{q}_1 = (q_2, q_4, \ldots, q_{2n})^T \tag{2.13}$$

$$\mathbf{q}_2 = (q_1, q_3, \ldots, q_{2n-1})^T \tag{2.14}$$

In other words $\mathbf{q}_1$ is the vector of link variables and $\mathbf{q}_2$ is the vector of actuator variables (divided by the gear ratio).

We have shown by the previous discussion then that the kinetic energy of the system under our modeling assumption (A1) is

$$K = \frac{1}{2}\dot{\mathbf{q}}_1^T D(\mathbf{q}_1)\dot{\mathbf{q}}_1 + \frac{1}{2}\dot{\mathbf{q}}_2^T J \dot{\mathbf{q}}_2 \tag{2.15}$$

where $D(\mathbf{q}_1)$ is the inertia of the "rigid" robot

$$D(\mathbf{q}_1) = (d_{ij}(\mathbf{q}_1)) \tag{2.16}$$

which can be calculated using standard techniques, (e.g., formula 6.66 in [4]) once the rotor masses are included as part of the proximal links for the calculation of the latter's inertia tensor. The $n \times n$ matrix J is given by

$$J = diag\left[m_1^2 I_{zz_1}, \ldots, m_n^2 I_{zz_n}\right] \tag{2.17}$$

where the diagonal elements are the motor inertias about their principal axes of rotation multiplied by the square of the respective gear ratios.

We now invoke our second assumption (A2) again that the rotor inertia is symmetric about its axis of rotation. This implies that the gravitational potential is a function only of $\mathbf{q}_1$. Therefore the total potential energy of the system is

$$P = P_1(\mathbf{q}_1) + P_2(\mathbf{q}_1 - \mathbf{q}_2) \tag{2.18}$$

where, as in the case of the kinetic energy, the potential energy term P_1 is found from standard formulae for rigid robots (e.g., formula 6.54 in [4]). The second term above is due to the elastic potential of the spring and is given as

$$P_2 = \frac{1}{2}k(\mathbf{q}_1 - \mathbf{q}_2)^T(\mathbf{q}_1 - \mathbf{q}_2). \tag{2.19}$$

The Lagrangian $L = K - P$ of the system is now given by

$$L = \frac{1}{2}\dot{\mathbf{q}}_1^T D(\mathbf{q}_1)\dot{\mathbf{q}}_1 + \frac{1}{2}\dot{\mathbf{q}}_2 J\dot{\mathbf{q}}_2 - P_1(\mathbf{q}_1) - \frac{1}{2}k(\mathbf{q}_1 - \mathbf{q}_2)^T(\mathbf{q}_1 - \mathbf{q}_2) \tag{2.20}$$

and the equations of motion are found from the Euler-Lagrange equations [4] using (2.20) to be

$$D(\mathbf{q}_1)\ddot{\mathbf{q}}_1 + c(\mathbf{q}_1, \dot{\mathbf{q}}_1) + k(\mathbf{q}_1 - \mathbf{q}_2) = 0 \tag{2.21}$$

$$J\ddot{\mathbf{q}}_2 - k(\mathbf{q}_1 - \mathbf{q}_2) = u. \tag{2.22}$$

The $n \times n$ matrix $D(\mathbf{q}_1)$ is symmetric, positive definite for each $\mathbf{q}_1$. The vector $C(\mathbf{q}_1, \dot{\mathbf{q}}_1)$ contains coriolis, centripetal, and gravitational forces and torques, and can be expressed as

$$c(\mathbf{q}_1, \dot{\mathbf{q}}_1) = \dot{D}\dot{\mathbf{q}}_1 - \frac{1}{2}\dot{\mathbf{q}}_1^T\frac{\partial D}{\partial \mathbf{q}_1}\dot{\mathbf{q}}_1 - \frac{\partial P_1}{\partial \mathbf{q}_1} \tag{2.23}$$

We note however that the gyroscopic forces between each rotor and the other links are not included in this expression as a result of Assumption (A1). It is interesting to compare the simplicity of our model (2.21)-(2.22) with other models of elastic joint manipulators that have been derived in the literature [13, 15, 18, 20, 23, 29]. It turns out that (2.21)-(2.22) is structurally similar to the models used in [26] and [27]. Thus our model can be viewed as the general n-degree-of-freedom extension to the models in the latter references.

Interestingly enough, our model is also the direct extension to the case of elastic joints of the familar rigid models that have become standard in the literature. In fact, we can easily see that the usual rigid model can be recovered from (2.21)-(2.22) as the joint stiffness parameters k_i tend to infinity. To see this we assume that in the limit as $\mathbf{k} \rightarrow \infty$ there is no elastic deformation, so that

$$\mathbf{q}_1 = \mathbf{q}_2, \dot{\mathbf{q}}_1 = \dot{\mathbf{q}}_2. \tag{2.24}$$

On the other hand the force $k(\mathbf{q}_1 - \mathbf{q}_2)$ transmitted through the coupling between the rotor and link remains finite in the rigid case, i.e., as $k \rightarrow \infty$, and it follows that the potential energy P_2 in (2.18) satifies

$$\frac{1}{2}k(\mathbf{q}_1 - \mathbf{q}_2)^T(\mathbf{q}_1 - \mathbf{q}_2) \rightarrow 0 \tag{2.25}$$

as $k \rightarrow \infty$ and $\mathbf{q}_2 - \mathbf{q}_1 \rightarrow 0$. Therefore the Lagrangian of the rigid system L_r is obtained from (2.20) as

$$L_r = \frac{1}{2}\dot{\mathbf{q}}_1^T(D(\mathbf{q}_1) + J)\dot{\mathbf{q}}_1 - P_1(\mathbf{q}_1) \tag{2.26}$$

which leads to the equations of motion for the rigid system by applying the Euler-Lagrange equations to (2.26)

$$D(\mathbf{q}_1 + J)\ddot{\mathbf{q}}_1 + c(\mathbf{q}_1, \dot{\mathbf{q}}_1) = u \tag{2.27}$$

The interesting implication of this is that the usual textbook model of rigid robots is subject to the same assumptions (A1) and (A2) that we use to derive the elastic joint model. Gyroscopic forces due to the rotation of the actuators are thus not considered in most existing rigid models. See [32] for an exception to this statement which does consider the modeling of these gyroscopic terms in the case of the rigid joints. It is of interest to note that [32] concluded that these gyroscopic terms can indeed by neglected in most cases.

3 Feedback Linearization

It is well known that the rigid robot equations (2.27) may be globally linearized and decoupled by nonlinear feedback. This is just the familiar inverse dynamics control scheme which transforms (2.27) into a set of double integrator equations which can then be controlled by adding an "outer loop" control [31].

The above technique of inverse dynamics control is now understood as a special case of a more general procedure for transforming a nonlinear system to a linear system, known as *external* or *feedback linearization.*

Definition 3.1: A nonlinear system

$$\dot{x} = f(x) + \sum_{i=1}^{n} g_i(x)u_i = f(x) + G(x)u \tag{3.1}$$

is said to be *feedback linearizable* in a neighborhood U_o of the origin if there is a diffeomorphism $T:U_o \rightarrow R^n$ and nonlinear feedback

$$u = \alpha(x) + \beta(x)v \tag{3.2}$$

such that the transformed state

$$y = T(x) \tag{3.3}$$

satisfies the linear system

$$\dot{y} = Ay + Bv \tag{3.4}$$

where (A, B) is a controllable linear system.

Necessary and sufficient conditions for a system of the form (3.1) to be feedback linearizable are given in [3]. In the case of elastic joint robots, the feedback linearization property was investigated in [13] using computer generated models of the manipulator dynamics. These models are sufficiently complex, even for two link examples that another computer program was used in [13] to check the conditions for feedback linearization. The answer was negative, i.e., the elastic joint model derived in [13] is not general linearizable in this fashion. In this section we show that the new model (2.21)-(2.22) is always globally feedback linearizable according to Definition 3.1. Moreover we do not need symbolic programs to check linearizability or to compute the required state space change of coordinates or the nonlinear feedback law. These can be found by inspection.

We first write the system (2.21)-(2.22) in state space by setting

$$x_1 = q_1 \qquad x_2 = \dot{q}_1$$
$$x_3 = q_2 \qquad x_4 = \dot{q}_2 \tag{3.5}$$

Then we have from (2.21)-(2.22)

$$\dot{x}_1 = x_2 \tag{3.6}$$

$$\dot{x}_2 = -D(x_1)^{-1}\{c(x_1, x_2) + k(x_1 - x_3)\} \tag{3.7}$$

$$\dot{x}_3 = x_4 \tag{3.8}$$

$$\dot{x}_4 = J^{-1}k(x_1 - x_3) + J^{-1}u \tag{3.9}$$

Since the nonlinearities enter into the second equation above, while the control appears only in the last equation, it is not obvious that the system is linearizable nor can u immediately be chosen to cancel the nonlinearities as in the case of the rigid equations (2.24).

In order to check feedback linearizability of the above system one needs, in principle, to check rank conditions and involutivity of certain sets of vector fields formed by taking Lie brackets of the vector fields defining the state equations (3.6)-(3.9). Our model is simple enough, however, that we can show global feedback linearizabilty by directly computing the required change of coordinates and nonlinear feedback law. Moreover the new coordinates themselves turn out to have physical significance for the control problem at hand.

Consider now the nonlinear state space change of coordinates.

$$y_1 = T_1(x) = x_1 \tag{3.10}$$

$$y_2 = T_2(x) = \dot{T}_1 = x_2 \tag{3.11}$$

$$y_3 = T_3(x) = \dot{T}_2 \tag{3.12}$$
$$= -D(x_1)^{-1}\{c(x_1, x_2) + k(x_1 - x_3)\}$$

$$y_4 = T_4(x) = \dot{T}_3 \tag{3.13}$$
$$= -\frac{d}{dt}D(x_1)^{-1}\{c(x_1, x_2) + k(x_1 - x_3)\}$$
$$-D(x_1)^{-1}\left\{\frac{\partial c}{\partial x_1}x_2\right.$$
$$+\frac{\partial c}{\partial x_2}(-D(x_1)^{-1}(c(x_1, x_2) + k(x_1 - x_3)))$$
$$\left. + k(x_2 - x_4)\right\}$$
$$:= f_4(x_1, x_2, x_3) + D(x_1)^{-1}kx_4$$

where for simplicity we define the function f_4 to be everything in the definition of y_4 above except the last term, which is $D^{-1}kx_4$. Note that x_4 appears only in this last term so that f_4 depends only on x_1, x_2, x_3.

The above mapping is actually a global diffeomorphism. Its inverse is likewise found by inspection to be

$$x_1 = y_1 \tag{3.14}$$

$$x_2 = y_2 \tag{3.15}$$

$$x_3 = y_1 + k^{-1}(D(y_1)y_3 + c(y_1, y_2)) \tag{3.16}$$

$$x_4 = k^{-1}D(y_1)(y_4 - f_4(y_1, y_2, y_3)). \tag{3.17}$$

The linearizing control law can now be found from the condition

$$\dot{y}_4 = \nu \tag{3.18}$$

where ν is a new control input. Computing $\dot{y}_4$ from (3.13) and suppressing function arguments for brevity yields

$$\nu = \frac{\partial f_4}{\partial x_1}x_2 - \frac{\partial f_4}{\partial x_2}D^{-1}(c + k(x_1 - x_3)) \tag{3.19}$$
$$+\frac{\partial f_4}{\partial x_3}x_4 + \left(\frac{d}{dt}D^{-1}\right)kx_4 + D^{-1}k(J^{-1}k(x_1 - x_3) + J^{-1}u)$$
$$:= F(x_1, x_2, x_3, x_4) + D(x_1)^{-1}kJ^{-1}u$$

where $F(x_1, x_2, x_3, x_4) = F(x)$ denotes all the terms in (3.19) but the last term, which involves the input u.

Solving the above expression for u yields

$$u = Jk^{-1}D(x_1)(\nu - F(x)) \tag{3.20}$$

With the nonlinear change of coordinates (3.10)-(3.13) and nonlinear feedback (3.20) the system (3.6)-(3.9) now has the linear block form

$$\dot{y} = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ 0 \\ 0 \\ I \end{bmatrix} \nu \tag{3.21}$$

where $I = n \times n$ identity matrix, $0 = n \times n$ zero matrix, $y^T = (y_1^T, y_2^T, y_3^T, y_4^T) \epsilon R^{4n}$, and $\nu \epsilon R^n$.

The nonlinear control law (3.20) is not completely determined until the function ν is specified. We will detail next one design scheme for ν which guarantees robust tracking for the above system.

Closed Loop Performance and Robustness It is easy to determine from the linear system (3.21) with linear feedback control (3.22) what the response of the system in the y_i coordinate system will be. The corresponding response of the original coordinates x_i is not necessarily easy to determine since the nonlinear coordinate transformation (3.10)-(3.13) must be inverted to find the x_i. However, in this case the transformed coordinates y_i are themselves physically meaningful. Inspecting (3.10)-(3.13) we see that the variables y_1, y_2, y_3, y_4 are n-vectors representing, respectively, the link positions, velocities, accelerations, and jerks (derivative of the acceleration). Since the motion trajectory of the manipulator is typically specified in terms of these quantities [4], they are natural variables to use for control.

The issue of robustness to parameter uncertainty is an important one at this point. In order to control the linear system (3.21) either the y_i coordinates must be physically measurable, or the y_i must be computed from the measured x_i variable according to (3.10)-(3.13), or a robust observer for these variables must be constructed. In the first case, the required measurements may be difficult to obtain, although solid state accelerometers are now available which could greatly simplify the problem. In the second case, that of computing the y_i via (3.10)-(3.130), one needs accurate estimates of the parameters in the manipulator model. In the third case there are results on the design of nonlinear observers which could be applied to this problem [35].

The computation of the overall nonlinear control law (3.20) also requires knowledge of the model parameters. In what follows we assume that both the original variables x_i and the transformed variables y_i may be used for feedback, and we consider the robust tracking problem.

Following [33] we consider the transformed system

$$\dot{y} = y_2 \tag{3.23}$$

$$\dot{y}_2 = y_3$$

$$\dot{y}_3 = y_4$$

$$\dot{y}_4 = F(x) + D^{-1}kJ^{-1}u$$

$$:= -\beta(x)^{-1}\alpha(x) + \beta(x)^{-1}u$$

that is, $\beta(x) = Jk^{-1}D(x)$ and $\alpha = \beta(x)F(x)$.

Now the control law

$$u = \alpha(x) + \beta(x)\nu \tag{3.24}$$

that is, (3.20), which ideally linearizes the system is unachievable in practice due to parameter uncertainty, computational roundoff, unknown disturbances, etc. It is more reasonable to assume a control law of the form

$$u = \hat{\alpha}(x) + \hat{\beta}(x)\nu \tag{3.25}$$

where $\hat{\alpha}(x)$ and $\hat{\beta}(x)$ are estimated or computed values of $\alpha(x)$ and $\beta(x)$, respectively. In addition, the functions α and β are extremely complicated so that $\hat{\alpha}$ and $\hat{\beta}$ may represent intentional model simplification to facilitate real-time computation. In what follows $\|x\|$ denotes the usual L_2-norm or Euclidean norm of a vector $x \epsilon R^n$ and, for any matrix M, $\|M\|$ is the corresponding induced matrix norm, i.e.,

$$\|M\| = \sqrt{\lambda_{\max}(M^TM)}$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of a matrix. We make the following assumptions on the functions α, $\hat{\alpha}$, β, $\hat{\beta}$.

(A3) There exist positive constants $\bar{\beta}$ and β such that

$$\beta \leq \|\beta^{-1}(x)\| \leq \bar{\beta} \tag{3.26}$$

(A4) There is a positive constant $a < 1$ such that

$$\|\beta^{-1}\hat{\beta} - I\| \leq a \tag{3.27}$$

(A5) There is a known function $\phi(x,, t)$ such that

$$\|\hat{a} - \alpha\| \leq \phi < \infty \tag{3.28}$$

We note that (A4) can always be satisfied by suitable choice of $\hat{\beta}$. For example, the choice $\hat{\beta} = 1/cI$, where I is the identity matrix and the constant c is $1/2\,(\bar{\beta} + \beta)$ results in [36]

$$\|\beta^{-1}\hat{\beta} - I\| \leq \frac{\bar{\beta} - \beta}{\bar{\beta} + \beta} < 1.$$

Now we substitute the control law (3.25) into (3.23) which results in

$$\dot{y}_4 = \beta^{-1}\hat{\beta}\nu + \beta^{-1}\Delta\alpha \tag{3.29}$$

$$= \nu + E\nu + \beta^{-1}\Delta\alpha$$

where $\Delta\alpha = \hat{\alpha} - \alpha$, and $E := \beta^{-1}\hat{\beta} - I$. Note that $\|E\| \leq a < 1$ from assumption (A4).

To track a desired trajectory $y_i^d(t)$ we first find K such that $\bar{A} := A + BK$ is stable, where A and B are defined by (3.21), and we set $\nu = \dot{y}_4^d(t) + Ke + \Delta\nu$, where e is the vector tracking error.

$$e(t) = \begin{bmatrix} y_1 - y_1^d \\ y_2 - y_2^d \\ y_3 - y_3^d \\ y_4 - y_4^d \end{bmatrix}$$

The above system may now be written in "error space" as

$$\dot{e} = Ae + B\{\Delta\nu + \Psi\} \tag{3.30}$$

where Ψ is the nonlinear function (hereafter referred to as the ("uncertainty") defined by

$$\Psi = E(\dot{y}_4^d + Ke + \Delta\nu) + \beta^{-1}\Delta\alpha \tag{3.31}$$

The problem of robust trajectory tracking now reduces to the problem of stabilizing the system (3.30) by suitable choice of the additional input $\Delta\nu$. The above formulation is valid for any system that is feedback linearizable as is shown in [33] and any number of techniques can now be used to design the input $\Delta\nu$. However, the problem of stabilizing (3.30) in nontrivial since Ψ is a function of both e and $\Delta\nu$ and hence Ψ cannot be treated merely as a disturbance to be rejected by $\Delta\nu$. A more sophisticated analysis and design is required to guarantee stability of (3.30).

Approaches that can be used to design $\Delta\nu$ in (3.30) to guarantee robust tracking include Lyapunov and sliding mode designs [6], [5], high gain [8]and other approaches. We shall outline one approach to robust stabilization of feedback linearizable systems based on Lyapunov's second method. See [33] for the details and proofs.

First we note that from our assumptions on the uncertainty we have

$$\|\Psi\| \leq a(\|\dot{y}_4^d\| + \|Ke\| + \|\Delta\nu\|) + \bar{\beta}\phi \tag{3.32}$$

$$\leq \bar{\phi} + a\|\Delta\nu\|$$

where $\bar{\phi} := a(\|\dot{y}_4^d\| + \|Ke\|) + \bar{\beta}\phi$. Suppose that we can simultaneously satisfy the inequalities

$$\|\Psi\| \leq \rho(e, t) \tag{3.33}$$

$$\|\Delta\nu\| \leq \rho(e, t) \tag{3.34}$$

for a known function $\rho(e, t)$. The function ρ can be determined as follows. First suppose that $\Delta\nu$ satisfies (3.34). Then from (3.32) we have

$$\|\Psi\| \leq \bar{\phi} + a\rho := \rho \tag{3.35}$$

This definition of ρ is well-defined since $a < 1$ and we have

$$\rho = \frac{1}{1-a}\bar{\phi}. \tag{3.36}$$

It now follows from [33] that the null solution of (3.30) is uniformly asymptotically stable (in a generalized sense) if $\Delta\nu$ is chosen as

$$\Delta\nu = \begin{cases} -\rho\dfrac{B^TPe}{\|B^TPe\|}; & \text{if } \|B^TPe\| \neq 0 \\ 0; & \text{if } \|B^TPe\| = 0 \end{cases} \tag{3.37}$$

where P is the unique positive definite solution to the Lyapunov equation

$$\bar{A}^TP + P\bar{A} = -Q \tag{3.32}$$

for a given symmetric, positive definite Q. The argument is completed by noting that indeed $\|\Delta\nu\| \leq \rho$.

4 Integral Manifold Approach

The above feedback linearizing control scheme requires measurement of the link positions and velocities, the motor positions and velocities as well as the link accelerations and jerks for successful implementation. In this section we present a diffemt approach based on a reformulation of the dynamic equations (2.21)-(2.22) as a singularly perturbed system and the concept of integral manifold. In the case of weakly elastic joints, such as arise in harmonic drive gear elasticity, this approach has the advantage that it may be applied even when only the link position and velocity are available for feedback, provided that the system has a degree of natural damping at the joints. We will make this precise later.

Returning to the original system, we set

$$z = k(\mathbf{q}_2 - \mathbf{q}_1);\ \mu = \frac{1}{k} \tag{4.1}$$

Then z is the elastic force at the joints. If we now choose coordinates z and $\mathbf{q}_1$ we have from (2.21) and (2.22)

$$\ddot{\mathbf{q}}_1 \ = \ -D(\mathbf{q}_1)^{-1}c(\mathbf{q}_1, \dot{\mathbf{q}}_1) - D(\mathbf{q}_1)^{-1}z \tag{4.2}$$

$$:= \ a_1(\mathbf{q}, \dot{\mathbf{q}}) + A_1(\mathbf{q})z$$

where we henceforth drop the subscript on $\mathbf{q}$ for convenience. Likewise,

$$\mu\ddot{z} \ = \ \ddot{\mathbf{q}}_1 - \ddot{\mathbf{q}}_2 \tag{4.3}$$

$$= \ -D(\mathbf{q}_1)^{-1}c(\mathbf{q}_1, \dot{\mathbf{q}}) - (D(\mathbf{q}_1)^{-1} + J^{-1})z - J^{-1}u$$

$$= \ a_2(\mathbf{q}, \dot{\mathbf{q}}) + A_2(\mathbf{q})z + B_2u$$

In this case not that $a_2 = a_1$ and B_2 is constant and invertible. The model (4.2)-(4.3) is singularly perturbed. In the limit as $\mu \rightarrow 0$(4.2)-(4.3) reduces to the rigid equations of motion. In other words, by formally setting $\mu = 0$ in (4.3) and eliminating z from the equations, one obtains the rigid equations (2.24), as we now show.

Setting $\mu = 0$ in (4.3) and solving for z yields

$$z = -(D^{-1} + J^{-1})^{-1}(D^{-1}c + J^{-1}u) \tag{4.4}$$

which, when substituted into (4.2) yields

$$\ddot{q} \ = \ -D^{-1}c + D^{-1}(D^{-1} + J^{-1})^{-1}(D^{-1}c + J^{-1}u) \tag{4.5}$$

$$= \ -D^{-1}c + D^{-1}(D^{-1} + J^{-1})D^{-1}c + D^{-1}(D^{-1}J^{-1})^{-1}J^{-1}u$$

Now a straightforward calculation shows that the first two term in (4.5) above may be combined to yield

$$-D^{-1}c + D^{-1}(D^{-1} + J^{-1})^{-1}D^{-1}c \tag{4.6}$$

$$= -D^{-1}c + D^{-1}J(D+J)^{-1}c$$

$$= (-D^{-1}(D+J) + D^{-1}J)(D+J)^{-1}c$$

$$= -(D+J)^{-1}c$$

Likewise the second term in (4.5) can be simplified as

$$D^{-1}(D^{-1} + J^{-1})^{-1}J^{-1}u \tag{4.7}$$

$$= D^{-1}J(D+J)^{-1}DJ^{-1}u$$

$$= [J\,D^{-1}(D+J)J^{-1}D]^{-1}u = (D+J)^{-1}u$$

and so the reduced order system (4.5) simplifies to

$$\ddot{\mathbf{q}} = -(D+J)^{-1}c + (D+J)^{-1}u \tag{4.8}$$

which is just the rigid sytem (2.24).

Integral Manifold In the $4n$-dimensional state space of (4.2)-(4.3), a $2n$ dimensional manifold M_μ may be defined by the expressions,

$$z = h(\mathbf{q}, \dot{\mathbf{q}}, u, \mu) \tag{4.9}$$

$$\dot{z} = \dot{h}(\mathbf{q}, \dot{\mathbf{q}}, u, \mu) \tag{4.10}$$

The manifold M_μ is said to be an *integral manifold* (4.2)-(4.3) if it is invariant under solutions of the system. In other words, given an admissible input function $t \rightarrow u(t)$, if $\mathbf{q}(t)$, $z(t)$ are solutions of (4.2)-(4.3) for $t > t_0$ with initial conditions $\mathbf{q}(t_0) = \mathbf{q}^0$, $\dot{\mathbf{q}}(t_0) = \dot{\mathbf{q}}^0$, z^0, $\dot{z}(t_0) = \dot{z}^0$ then

$$z^0 = h(\mathbf{q}^0, \dot{\mathbf{q}}^0, u(t_0), \mu) \tag{4.11}$$

$$\dot{z}^0 = \dot{h}(\mathbf{q}^0, \dot{\mathbf{q}}^0, u(t_0), \mu)$$

implies that for $t > t_0$

$$z(t) = h(\mathbf{q}(t), \dot{\mathbf{q}}(t), u(t), \mu) \tag{4.12}$$

$$\dot{z}(t) = \dot{h}(\mathbf{q}(t), \dot{\mathbf{q}}(t), u(t), \mu) \tag{4.13}$$

In otherwords, if the system lies initially on the manifold M_μ, then the solution trajectory remains on the manifold M_μ for $t > t_0$.

The integral manifold M_μ is characterized by the following partial differential equation, formed by substituting the expression (4.9) into the equation (4.3)

$$\mu\ddot{h} = a_2(\mathbf{q}, \dot{\mathbf{q}}) + A_2(\mathbf{q})h + B_2u \tag{4.14}$$

In otherwords, if the system lies initially on the manifold M_μ, then the solution trajectory remains on the manifold M_μ for $t > t_0$.

$$\dot{h} = \frac{\partial h}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial h}{\partial \dot{\mathbf{q}}}(a_1 + A_1z) + \frac{\partial h}{\partial u}\dot{u}$$

and $\ddot{h}$ is to be similarily expanded. Although the p.d.e. (4.14) is seemingly difficult we shall actually find an explicit solution.

Once h is determined from (4.14), the dynamics of the system (4.2)-(4.3) on the integral manifold are given by a reduced order system referred as the *reduced flexible system* formed by replacing z by h in (4.2)

$$\ddot{\mathbf{q}} = a_1(\mathbf{q}, \dot{\mathbf{q}}) + A_1(\mathbf{q})h(\mathbf{q}, \dot{\mathbf{q}}, u, \mu) \tag{4.15}$$

Equation (4.15) is of the same order as the rigid system, but as shown in [18] is a more accurate approximation of the flexible system than is the rigid model (2.24). We leave it to the reader to verify that the reduced flexible system reduces to the rigid system (2.24) as the perturbation parameter μ tends to zero.

We now utilize the concept of composite control [10] and choose the control input u of the form

$$u = u_s(\mathbf{q}, \dot{\mathbf{q}}, \nu, \mu) + u_f(\eta, \dot{\eta}) \tag{4.16}$$

where ν represents a new input to be specified. We also specify $u_f(0,0) = 0$ so that $u = u_s$ on the integral manifold. The variable η represents the deviation of the fast variables from the integral manifold, i.e.,

$$\eta = z - h(\mathbf{q}, \dot{\mathbf{q}}, u_s, \mu) \tag{4.17}$$

$$\dot{\eta} = \dot{z} - \dot{h}(\mathbf{q}, \dot{\mathbf{q}}, u_s, \mu).$$

Since $u = u_s$ on the integral manifold we may combine (4.3) and (4.14) to obtain

$$\mu\eta \ = \ \mu\ddot{z} - \mu\ddot{h}$$

$$\mu\ddot{\eta} \ = \ a_2 + A_2z + B_2u - (a_2 + A_2h + B_2u_s)$$

$$= \ A_2\eta + B_2u_f$$

Therefore, in terms of the variables $\mathbf{q}$ and η, the system (4.2)-(4.3) is rewritten as

$$\ddot{\mathbf{q}} \ = \ a_1 + A_1h(\mathbf{q}, \dot{\mathbf{q}}, u_s, \mu) + A_1\eta \tag{4.18}$$

$$\mu\ddot{d} \ = \ A_2(\mathbf{q})\eta + B_2u_f \tag{4.19}$$

In order to solve the P.D.E. (4.14) defining the integral manifold, we expand the function h in terms of μ as

$$h(\mathbf{q}, \dot{\mathbf{q}}, u_s, \mu) = h_0(\mathbf{q}, \dot{\mathbf{q}}, u_0) + \mu h(\mathbf{q}, \dot{\mathbf{q}}, u_0 + \mu u_1) + \ \ldots \tag{4.20}$$

and we choose u_s as

$$u_s = u_0 + \mu u_1 \tag{4.21}$$

Substituting these expressions into the manifold condition (4.14) yields

$$\mu\{\ddot{h}_0 + \mu\ddot{h}_1 + \ \ldots\ \} = a_2 + A_2(h_0 + \mu h_1 + \ \ldots\ ) \tag{4.22}$$

$$+ B_2(u_0 + \mu u_1).$$

Equating coefficients of μ^k we obtain the sequence of equalities

$$0 = a_2 + A_2 h_0 + B_2 u_0 \tag{4.23}$$

$$\ddot{h} = A_2 h_1 + B_2 u_1 \tag{4.24}$$

$$\ddot{h}_{k-1} = A_2 h_k,\ k > 1. \tag{4.25}$$

Equation (4.23) may be solved for h_0 to yield

$$h_0 = -A_2^{-1}(a_2 + B_2 u_0) \tag{4.26}$$

The derivation now proceeds iteratively. The control u_0 is first computed at $\mu = 0$, that is, based on the rigid model, and can be any one of the many schemes that have been derived for control of rigid manipulators. Given u_0 then h_0 is computable from (4.26). From this, with the given u_0 we can compute $\ddot{h}_0$ and so we can write (4.24) as

$$A_2 h_1 = \ddot{h}_0 - B_2 u_1. \tag{4.27}$$

where the right-hand side contains only known quantities and the control. Since both A_2, and B_2 are invertible, we see that by setting

$$u_1 = B_2^{-1}\ddot{h}_0 \tag{4.28}$$

it follows that

$$h_1 \equiv 0 \text{ and therefore } \ddot{h}_1 \equiv 0 \tag{4.29}$$

From this it follows iteratively from (4.25) and the invertibility of A_2 that $h_k = 0$ for $k > 1$.

We have show therefore that the choice of control input

$$u_s = u_0 + \mu u_1 \tag{4.30}$$

with u_1 given by (4.28) results in $h = h_0$. Thus, on the integral manifold, i.e., when $\eta = 0$, the dynamics of the system are described by the reduced order system.

$$\ddot{\mathbf{q}} = a_1 - A_1 A_2^{-1} a_2 - A_1 A_2^{-1} B_2 u_0 \tag{4.31}$$

$$= -(D(\mathbf{q}) + J)^{-1}\{c(\mathbf{q}, \dot{\mathbf{q}}) + u_0\}$$

which is of course just the rigid system.

We see that we have produced a solution h_0 of the manifold condition (4.14). The fact that h_0, given by (4.26), satisfies (4.14) is significant. What this implies is that by adding the corrective control $\mu\mu_1$ the integral manifold h becomes the rigid manifold h_0. To put it another way, the rigid manifold h_0 is made an invariant manifold for the flexible system by the corrective control.

If the control u_0 is chosen to be the feedback linearizing control for the rigid system

$$u_0 = (D(\mathbf{q}) + J)\nu + c(\mathbf{q}, \dot{\mathbf{q}}) \tag{4.32}$$

we obtain the overall system

$$\ddot{\mathbf{q}} = \nu + A_1(\mathbf{q})\eta \tag{4.33}$$

$$\mu\ddot{\eta} = A_2(\mathbf{q})\eta + B_2 u_f \tag{4.34}$$

Since B_2 is nonsingular the fast subsystem (4.34), which is a linear system in η parmeterized by $\mathbf{q}$, is controllable for each $\mathbf{q}$. Thus there exists a fast control $u_f(\eta, \dot{\eta})$ to place the poles of (4.34) arbitrarily. Note that we have not explicitly included damping in the model. Thus for each $\mathbf{q}$, since $-A_2$ is a positive definite matrix, the open loop poles of (4.34) are on the $j\omega$-axis. This shows clearly the resonance phenomonon whereby the elastic oscillations from (4.34) drive the slow variables through (4.33). Since A_2 is a function of $\mathbf{q}$ the resonant modes will be configuration dependent, a fact that was experimentally verified in [25]. In case the system (4.34) has some inherent natural damping one can show that the fast subsystem is of the form

Table 1 Parameters used for simulation

Mass	$m =$	1
Stiffness	$k =$	100
Length (2L)	$L =$	1
Gravity	$g =$	9.8
Inertias	$I =$	1
	$J =$	1

$$\mu\ddot{\eta} = A_2(\mathbf{q})\eta + \sqrt{\mu}A_3(\mathbf{q})\dot{\eta} + B_2 u_f \tag{4.35}$$

in which the fast variables, represented by η, decay to zero with $u_f = 0$. In other words the integral manifold, which in this case is the rigid manifold becomes an attracting set. Solutions off the manifold rapidily converge to the manifold after which the system equations are just the rigid equations. In this case the control u_s consisting of the rigid control plus the corrective control achieves the desired result. The point to note that this slow control is a function only of $\mathbf{q}$, $\dot{\mathbf{q}}$. Thus the corrective control compensates for the elasticity using a limited set of state measurements.

If there is no damping in the fast variables, or if the damping is insufficient then the fast control u_f must be added. Note that the choice

$$u_f = B_2^{-1}(\zeta - A_2(\mathbf{q})\eta) \tag{4.36}$$

where ζ is a new input, when applied to (4.34) results in

$$\mu\ddot{\eta} = \zeta \tag{4.37}$$

Inspecting (4.33), (4.37) we see that we have for all practical purposes produced an alternate feedback linearization of the original system, which exploits the two-time scale property of the elastic system. A linear control scheme can now be employed, for example

$$\nu = \alpha_1 \cdot \mathbf{q} + \alpha_2 \cdot \dot{\mathbf{q}} + r \tag{4.38}$$

$$\zeta = \beta_1 \cdot \eta + \sqrt{\mu}\beta_2 \cdot \dot{\eta} \tag{4.39}$$

to place the poles of the system arbitrarily. Note that implementation of the above control scheme requires either direct measurement of the fast variables, which in this case are the elastic forces at the joints and their time derivatives or else accurate knowledge of the system paramters in order that η and $\dot{\eta}$ can be computed from (4.17), which is an issue similar to that which arises in the feedback linearization approach of section 3.

5 An Example

For illustrative purposes consider the single link with the flexible joint of Fig. 1 with the parameters shown in Table 1. The equations of motion for this system in state space are easily computed to be

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -MgL/I \sin x_1 - k/I(x_1 - x_3) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= k/J(x_1 - x_3) + 1/Ju \end{aligned} \tag{5.1}$$

where $x_1 = q_1$, $x_3 = q_2$, etc.

In the limit as $k \to \infty$ the resulting rigid system is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -Mgl/(I+J)\sin x_1 + 1/(I+J)u \end{aligned} \tag{5.2}$$

where we take here $x_1 = {}_1 = q_1 = q_2$.

The feedback linearizing control law for (5.2) may be chosen as

$$u = (I+J)(\nu + Mgl \sin x_1) \tag{5.3}$$

with ν given as a a simple linear control term

$$\nu = \dot{x}_2^d - a_1(x_1 - x_1^d) - a_2(x_2 - x_2^d) \tag{5.4}$$

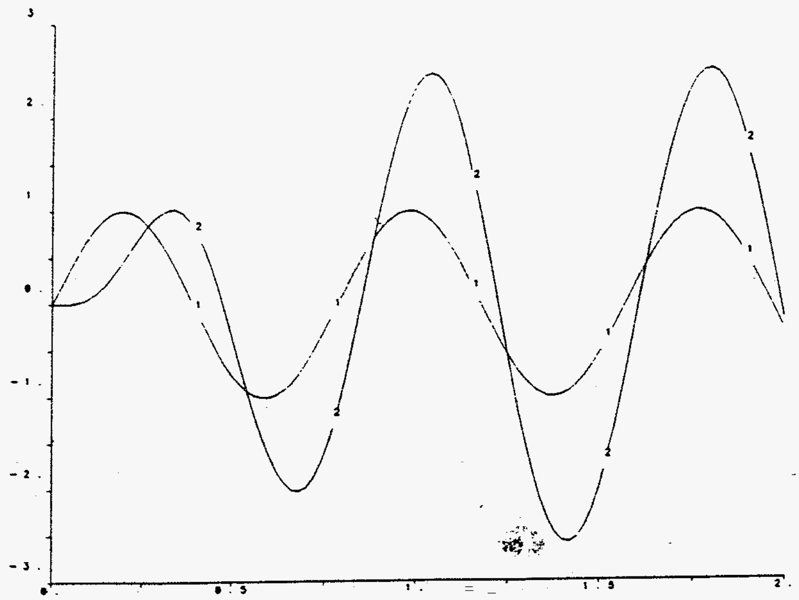


Fig. 3 Rigid control applied to flexible joint. 1 = reference trajectory; 2 = link angle.

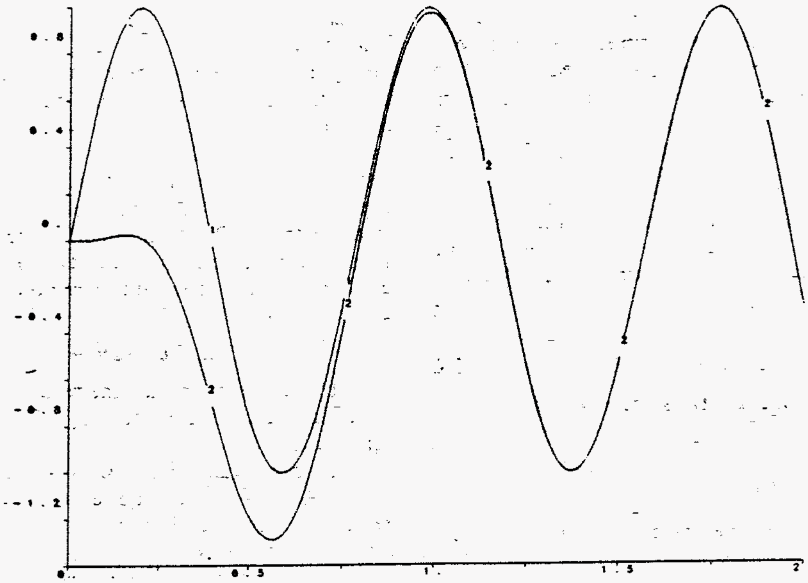


Fig. 4 Feedback linearization control. 1 = reference trajectory; 2 = link angle

designd to track a desired trajectory $t \to x_1^d(t)$.

It is interesting to see the response of the flexible joint system (5.1) to this "rigid control." At this point one must make a choice whether to use the motor variable q_2 or the link variable q_1 in this control law. Figure 3 shows the response of the link variable q_1 in the flexible joint system (5.1) using the motor variable $x_1 = q_2$ in (5.3)-(5.4), with a desired tracjectory $x_1^d = \sin 8t$. It is interesting to note that if one tries to feedback instead the link variable q_1 in (5.3)-(5.4) the system becomes unstable.

Feedback Linearization Control. The feedback linearizing transformation for this system is given by (3.10)-(3.13) as

$$
\begin{aligned}
y_1 &= x_1 \\
y_2 &= x_2 \\
y_3 &= -MgL/I \sin x_1 - k/I(x_1 - x_3) \\
y_4 &= -Mgl/I\cos x_1 \cdot x_2^2 - k/I(x_2 - x_4)
\end{aligned}
\qquad (5.5)
$$

(2)

The feedback linearizing control law computed from (3.19) and (3.20) turns out to be

$$u = \frac{IJ}{k}(v - F(x_1, x_2, x_3, x_4)) \qquad (5.6)$$

where

$$F(x_1, x_2, x_3, x_4) = MgL/I \sin x_1 \cdot x_2^2 \qquad (5.7)$$

$$+ (MgL/I\cos x_1 + k/I)(MgL/I \sin x_1 + k/I(x_1 - x_3))$$

$$+ k^2/IJ(x_1 - x_3)$$

A simple linear control law for v designed to track a desired trajectory $t \to y_i^d(t)$ can be expressed as

(3)

$$v = \dot{y}_4^d - \sum_{i=1}^{3} a_i (y_i - y_i^d) \qquad (5.8)$$

1) $u = (I+J)(v + \frac{Mgl}{(I+J)} \sin x_1)$

2) kwadraat weg

3) $v = \dot{\xi}_4^d - \sum_{i=1}^{4} a_i(\xi_i - \xi_i^d)$

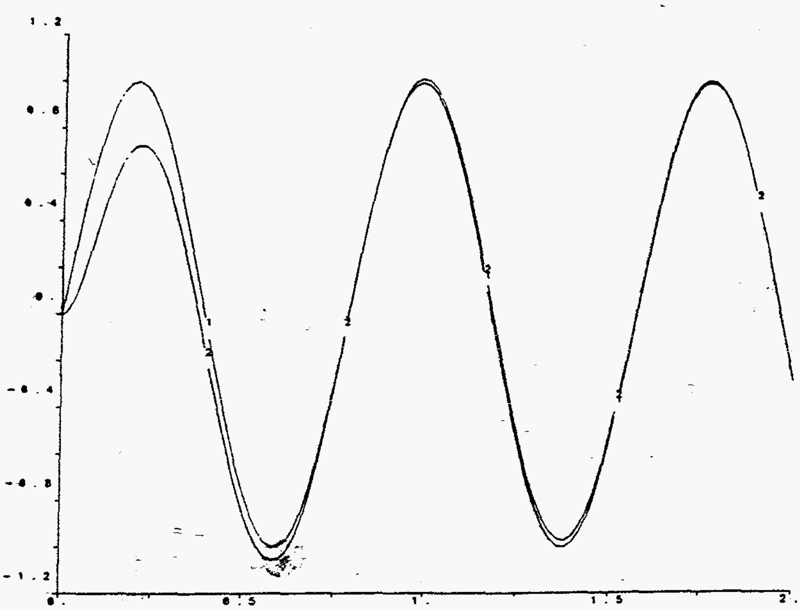


Fig. 5 Corrective control based on the integral manifold. 1 = reference trajectory; 2 = link angle

The zero-state response of the above system with a given desired trajectory $y_1 = \sin 8t$ is shown in Fig. 4. The gains in the control law (5.8) where chosen for simplicity to achieve a closed loop characteristic polynomial for the linearized system of $(s+10)^4$. This response illustrates the improved tracking resulting from basing the control design on the fourth-order flexible joint model rather than on the rigid model. The robust version of this control law, given by (3.37) is omitted.

Integral Manifold Control. In terms of the variables $q = q_1$ and $z = k(q_1 - q_2)$ the equations of motion for the system of Fig. 1 in singularly perturbed form with $\mu = 1/k$ are

$$\ddot{q} = -MgL \sin q - 1/I\, z \tag{5.5}$$

$$\mu \ddot{z} = -MgL \sin q - (1/I + 1/J)z - 1/Ju \tag{5.6}$$

In terms of the variables h and η the system (4.18)-(4.19) is

$$\ddot{q} = -MgL \sin q - 1/I\, h - 1/I\eta \tag{5.8}$$

$$\mu \ddot{\eta} = (1/I + 1/J)\eta - 1/Ju \tag{5.9}$$

where $\eta = z - h$ and h is determined from the manifold condition

$$\mu \ddot{\eta} = -MgL \sin q - (1/I + 1/J)h - 1/Ju \tag{5.10}$$

The detailed calculation of the asymptotic expansion of the function h and the corrective control are carried out in [23]. The interested reader is referred to that paper for the details and also more simulation results. In Fig. 5 the composite and corrective control law thus derived is applied to track the same desired trajectory $q^d = \sin 8t$.

6 Conclusions

In this paper we have rigorously derived a simple and rather intuitive model to represent the dynamics of elastic joint manipulators and presented two attractive control techniques for the resulting system. The first new result that we present is the global feedback linearization of the flexible joint system by nonlinear coordinate transformation and static state feedback. The importance of the property of feedback linearization is not necessarily that the nonlinearities in the system can by computed and exactly cancelled by feedback as this is never achievable in practice. Rather its significance is that once the proper coordinates are found in which to represent the system, the so-called matching conditions are satisfied, which is to say that the nonlinearities are all in the range space of the input. This property allows the design of control laws which are highly robust to parametric uncertainty. The second new result is based on the integral manifold formulation of the equations of motion. We have shown using the corrective control concept that the manifold in state space describing the dynamics of the rigid manipulator can be made invariant under solutions of the flexible joint system, independent of the joint stiffness. This result holds in general only for the model derived here. In previous models of elastic joint manipulators, as shown in [23], the results here hold up to $O(\mu^l)$ by applying a corrective control

$$u_s = u_0 + \mu u_1 + \ldots + \mu^l u_l. \tag{6.1}$$

With the present model the result is exactly achieved to any order in μ and is done so only with a first order correction term μu_1.

There are several interesting research issues that arise at this point. Among are the design of robust state estimators to realize the feedback linearization control using only the joint positions and velocities and also the computational issues associated with computing in real-time what amounts to a very complicated nonlinear control algorithm. Also, the robustness of the integral manifold based corrective control strategy needs to be investigated.

References

1 Bejczy, A. C., "Robot Arm Dynamics and Control," JPL Technical Memo., 33-369, Feb. 1974.

2 Freund, "Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators," *Int. J. Robotics Res.*, Vol. 1, No. 1, 1982, pp. 65–78.

3 Hunt, L. R., Su, R., and Meyer, G., "Design for Multi-input Nonlinear Systems," *Differential Geometric Control Theory Conf.*, Birkhauser, Boston, 1983, pp. 268–298.

4 Paul, R. P., *Robot Manipilators: Mathematics, Programming and Control*, MIT Press, Cambridge, Mass., 1982.

5 Slotine, J. J., "Robust Control of Robot Manipulators," *Int. J. Robotics Res.*, Vol. 4, No. 2, 1985.

6 Spong, M. W., Thorp, J. S., and Kleinwaks, J. W., "The Control of Robot Manipulators with Bounded Control Part II: Robustness and Disturbance Rejection," *Proc. 23nd IEEE CDC*, Las Vegas, NV, Dec. 1984.

7 Tarn, T. J., Bejczy, A. K., Isidori, A., and Chen, Y., "Nonlinear Feedback in Robot Arm Control." *Proc. 23rd IEEE CDC*, Las Vegas, NV, Dec. 1984.

8 Samson, C., "Robust Non Linear Control of Robotic Manipulators," *Proc. 22nd IEEE CDC*, San Antonio, Dec. 1983.

9 Sweet, L. M., and Good, M. C., "Re'Definition of the Robot Motion Control Problem: Effects of Plant Dynamics, Drive System Constraints, and User Requirement," *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, NV, Dec. 1984, pp. 724-731.

10 Chow, J. H., and Kokotovic, P. V., "Two Time Scale Feedback Design of a Class of Nonlinear Systems," *IEEE Trans. on Automatic Control*, AC-23, 1978, pp. 438-443.

11 Kokotovic, P. V., "Applications of Singular Perturbation Techniques to Control Problems," *SIAM Review*, Vol. 26, No. 4, 1984, pp. 396-410.

12 Erzerber, H., "Analysis and Design of Model Following Systems by State Space Techniques," *Proc. JACC*, 1968, pp. 572-581.

13 Cesareo, G. and Marino, R., "On the Controllability Properties of Elastic Robots," *Sixth Int. Conf. on Analysis and Optimization of Systems*, INRIA, Nice, 1984.

14 Fenichel, "Geometric Singular Perturbation Theory for Ordinary Differential Equations," *J. Differential Equations*, Vol. 3, 1979, pp. 53-98.

15 Ficola, A., Marino, R., and Nicosia, S., "A Singular Peturbation Approach to the Control of Elastic Robots," *Proc. 21st. Annual Allerton Conf. on Communication, Control, and Computing*, Univ. of Illinois, 1983.

16 Hoppensteadt, F., "Properties of Solutions of Ordinary Differential Equations with Small Parameters," *Comm. Pure Appl. Math.*, Vol 34, 1971, pp. 807-840.

17 Hunt, L. R., Su, R., and Meyer, G., "Global Transformations of Nonlinear Systems," *IEEE Trans. on Automatic Control*, Vol. Ac-28, 1983, pp. 24-30.

18 Khorasani, K., and Spong, M. W., "Invariant Manifolds and their Applications to Robot Manipulators with Flexible Joints," *Proc. 1985 IEEE Int. Conf. on Robotics and Automation*, St. Louis, Mar. 1985.

19 Kokotovic, P. V., "Control Theory in the 80's: Trends in Feedback Design," *Proc. 9th World Congress of IFAC*, July 1984.

20 Marino, R., and Nicosia, S., "On the Feedback Control of Industrial Robots with Elastic Joints: A Singular Perturbation Approach," University of Rome, R-84.01, 1984.

21 Sobolev, V. A., "Integral Manifolds and Decompositions of Singularly Perturbed Systems," *Systems and Control Letters*, Vol. 5, 1984, pp. 1169-179.

22 Su, R., "On the Linear Equivalents of Nonlinear Systems," *Systems and Control Letters*, Vol 2, 1981, pp. 48-52.

23 Spong, M. W., Khorasani, K., and Kokotovic, P. V., "A Slow Manifold Approach to Feedback Control of Flexible Joint Robots," to appear in *IEEE Journal of Robotics and Automation*, 1986.

24 Rivin, E. I., "Effective Rigidity of Robot Structure: Analysis and Enhancement," *Proc. 1985, ACC*, Boston, June 1985.

25 Good, M., Strobel, K., and Sweet, L. M., "Dynamics and Control of Robot Drive Systems," General Electric Company, Corporate Research and Development, 1983.

26 Forrest-Barlach, M. G., and Babcock, S. M., "Inverse Dynamics Position Control of a Compliant Manipulator," *Proc. IEEE Conference of Robotics and Automation*, San Francisco, Apr. 1986, pp. 196-205.

27 Marino, R. W., and Spong, M. W., "Nonlinear Control Techniques for Flexible Joint Manipulators: A Single Link Case Study," *Proc. 1986 IEEE Conference on Robotics and Automation*, San Francisco, Apr. 1986, pp. 1030-1036.

28 Khorasani, K., and Kokotovic, P. V., "Feedback Linearization of a Flexible Manipulator Near its Rigid Body Manifold," *Systems and Control Letters*, Vol. 6, 1985, pp. 187-192.

29 De Luca, A., Isidori, A., and Nicolo, F., "An Application of Nonlinear Model Matching to the Dynamic Control of Robot Arm with Elastic Joints," preprint, 1985.

30 Bejczy, A. K., personal communication, San Francisco, Apr. 1986.

31 Spong, M. W., and Vidyasagar, M., "Robust Nonlinear Control of Robot Manipulators," *Proc. 24th IEEE Conf. on Decision and Control*, Fort Lauderdale, Dec. 1985, pp. 1767-1772.

32 Springer, H., Lugner, P., and Desoyer, "Equations of Motion for Manipulators Including Dynamic Effects of Active Elements," *Proc. IFAC Symposium and Robot Control*, SYROCO' 85, Barcelona, Nov. 1985.

33 Spong, M. W., "Robust Stabilization For a Class of Nonlinear Systems," *Proc. 7th Int. Symp. on the Math. Theory of Network and Systems*, (MTNS), Stockholm, June 1985, Springer-Verlag.

34 Craig, J., *Introduction to Robotics*, Addison-Wesley, 1985.

35 Krener, A. J., and Isidori, A., "Linearization by Output Injection and Nonlinear Observers," *Systems and Control Letters*, Vol. 3, 1983, pp. 47-52.

36 Shoureshi, R., Roesler, M. D., and Corlis, M. J., "Control of Industrial Manipulators with Bounded Uncertainty," *Japan-USA Symposium of Flexible Automation*, Osaka, Japan. 1986.

Appendix II

For the model with damping we derived the state equation:

$$\underline{\dot{x}} = \begin{bmatrix} x_2 \\ -\frac{b_l}{I}x_2 - \frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1-x_3) \\ x_4 \\ -\frac{b_m}{J}x_4 + \frac{k}{J}(x_1-x_3) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/J \end{bmatrix} u$$

$$= f(\underline{x}) + g(\underline{x})u$$

I. computation of the Lie-brackets:

* $ad_f^0(g) = g = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/J \end{bmatrix}$

* $ad_f^1(g) = [f, g] = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g$

$$= \underline{0} - \begin{bmatrix} * & * & * & 0 \\ * & * & * & 0 \\ * & * & * & 1 \\ * & * & * & -b_m/J \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/J \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1/J \\ b_m/J^2 \end{bmatrix}$$

* $ad_f^2(g) = [f, ad_f^1(g)] = \frac{\partial(ad_f^1 g)}{\partial x} f - \frac{\partial f}{\partial x} ad_f^1 g$

$$= \underline{0} - \begin{bmatrix} * & * & 0 & 0 \\ * & * & \frac{k}{I} & 0 \\ * & * & 0 & 1 \\ * & * & -\frac{k}{J} & -\frac{b_m}{J} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1/J \\ b_m/J^2 \end{bmatrix} = \begin{bmatrix} 0 \\ k/IJ \\ -b_m/J^2 \\ \frac{b_m^2}{J^3} - \frac{k}{J^2} \end{bmatrix}$$

* $ad_f^3(g) = [f, ad_f^2(g)] = \frac{\partial(ad_f^2 g)}{\partial x} f - \frac{\partial f}{\partial x} ad_f^2(g)$

$$= \underline{0} - \begin{bmatrix} * & 1 & 0 & 0 \\ * & -b_l/I & \frac{k}{I} & 0 \\ * & 0 & 0 & 1 \\ * & 0 & -\frac{k}{J} & -\frac{b_m}{J} \end{bmatrix} \begin{bmatrix} 0 \\ k/IJ \\ -b_m/J^2 \\ \frac{b_m^2}{J^3} - \frac{k}{J^2} \end{bmatrix} = \begin{bmatrix} -k/IJ \\ \frac{b_l k}{I^2 J} + \frac{b_m k}{I J^2} \\ -\frac{b_m^2}{J^3} + \frac{k}{J^2} \\ -\frac{2 k b_m}{J^3} + \frac{b_m^3}{J^4} \end{bmatrix}$$

II Computation of $T_1(\underline{x})$:

* $L_g T_1 = 0 \Rightarrow \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/J \end{bmatrix} = 0 \Rightarrow \frac{\partial T_1}{\partial x_4} = 0$

* $L_{ad_f^1 g} T_1 = 0 \Rightarrow \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1/J \\ b_m/J^2 \end{bmatrix} = 0 \Rightarrow \frac{\partial T_1}{\partial x_3} = 0$

* $L_{ad_f^2 g} T_1 = 0 \Rightarrow \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} & \frac{\partial T_1}{\partial x_3} & \frac{\partial T_1}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 \\ k/IJ \\ -b_m/J^2 \\ \frac{b_m^2}{J^3} - \frac{k}{J^2} \end{bmatrix} = 0 \Rightarrow \frac{\partial T_1}{\partial x_2} = 0$

* $L_{ad_f^3 g} T_1 \neq 0 \Rightarrow \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -k/IJ \\ * \\ * \\ * \end{bmatrix} \neq 0 \Rightarrow \frac{\partial T_1}{\partial x_1} \neq 0$

For $T_1(x_1)$ we take the simplest solution : $T_1 = x_1$

III. computation of T_2, T_3 and T_4

* $$T_2 = L_f T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ * \\ * \\ * \end{bmatrix} = x_2$$

* $$T_3 = L_f T_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} * \\ -\frac{b_l}{I}x_2 - \frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1 - x_3) \\ * \\ * \end{bmatrix}$$
$$= -\frac{b_l}{I}x_2 - \frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1 - x_3)$$

* $$T_4 = L_f T_3 =$$
$$= \begin{bmatrix} -\frac{mgl}{I}\cos x_1 - \frac{k}{I} & -\frac{b_l}{I} & \frac{k}{I} & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ -\frac{b_l}{I}x_2 - \frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1 - x_3) \\ x_4 \\ * \end{bmatrix}$$
$$= -\frac{mgl}{I}x_2\cos x_1 - \frac{k}{I}x_2 + \left(\frac{b_l}{I}\right)^2 x_2 + \frac{b_l mgl}{I^2}\sin x_1 + \frac{b_l k}{I^2}(x_1 - x_3) + \frac{k}{I}x_4$$

⇒

$$\xi = T(x) = \begin{bmatrix} x_1 \\ x_2 \\ -\frac{b_l}{I}x_2 - \frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1 - x_3) \\ -\frac{mgl}{I}x_2\cos x_1 - \frac{k}{I}x_2 + \left(\frac{b_l}{I}\right)^2 x_2 + \frac{b_l mgl}{I^2}\sin x_1 + \frac{b_l k}{I^2}(x_1 - x_3) + \frac{k}{I}x_4 \end{bmatrix}$$

IV. computation of the control input u :

$$u = \frac{1}{L_g T_4}\left(v - L_f T_4\right)$$

* $$L_f T_4 = \begin{bmatrix} \frac{\partial T_4}{\partial x_1} & \frac{\partial T_4}{\partial x_2} & \frac{\partial T_4}{\partial x_3} & \frac{\partial T_4}{\partial x_4} \end{bmatrix} \begin{bmatrix} x_2 \\ -\frac{b_l}{I}x_2 - \frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1 - x_3) \\ x_4 \\ -\frac{b_m}{J}x_4 + \frac{k}{J}(x_1 - x_3) \end{bmatrix}$$
$$= \left[\frac{mgl}{I}x_2\sin x_1 + \frac{b_l mgl}{I^2}\cos x_1 + \frac{b_l k}{I^2} ; \; -\frac{mgl}{I}\cos x_1 - \frac{k}{I} + \left(\frac{b_l}{I}\right)^2 ; \; -\frac{b_l k}{I^2} ; \; \frac{k}{I} \right] \begin{bmatrix} x_2 \\ -\frac{b_l}{I}x_2 - \frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1 - x_3) \\ x_4 \\ -\frac{b_m}{J}x_4 + \frac{k}{J}(x_1 - x_3) \end{bmatrix}$$

computation of the above dual product is done during the simulation using MatLab.

* $$L_g T_4 = \begin{bmatrix} \frac{\partial T_4}{\partial x_1} & \frac{\partial T_4}{\partial x_2} & \frac{\partial T_4}{\partial x_3} & \frac{\partial T_4}{\partial x_4} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/J \end{bmatrix} = \frac{k}{IJ}$$

Besides the state-equation $\dot{x} = f(x) + g(x)u$, the transformation $\xi = T(x)$ and the feedback $u = \frac{1}{L_g T_4}(v - L_f T_4)$, nothing has changed compared to the example in Chapter 3. The simulation has been performed with the following program:

```
function xdot=link3(t,x)
%
%       FEEDBACK LINEARIZATION CONTROL OF A
%       SINGLE LINK WITH JOINT ELASTICITY
%
%       inputfile on behalf of main program
%
%*************************************************************************
%       parameters used for simulation
%
m = 1;      % mass
k = 100;    % stiffness
l = 1;      % length
g = 9.81;   % gravity
I = 1;      % inertia
J = 1;      % inertia
bl= 10;     % link damper  (ksi=0.5)
bm= 10;     % motor damper (ksi=0.5)
%
%*************************************************************************
%       state equation describing plant
%
xdot(1)= x(2);
xdot(2)= -(m*g*l/I)*sin(x(1)) - (k/I)*(x(1)-x(3)) - (bl/I)*x(2);
xdot(3)= x(4);
xdot(4)= (k/J)*(x(1)-x(3)) + (1/J)*u - (bm/J)*x(4);
%*************************************************************************
```

```
FEEDBACK LINEARIZATION CONTROL OF A
SINGLE LINK WITH JOINT ELASTICITY

main program

*************************************************************************
set starting values

obal u m g l I J k rij kol LgT4 LfT4 bl bm

 = 0.0;
 = [0 0 0 0]';
 = 0.0;

ut = [t0];
ut = [x0];
ut = [u];

*************************************************************************
begin recursive calculation

tart = 0.00;
tep  = 0.01;
top  = 2.00;

r tel = tstart:tstep:tstop,

    t0 = tel;
    tf = tel + tstep;
    [t,x] = ode45('link3',t0,tf,x0)

    n  = length(t);
    tn = t(n);
    xn = x(n,:)';

    *********************************************************************
    computation of coordinates transformation

    ksi(1) = xn(1);
    ksi(2) = xn(2);
    ksi(3) = -(m*g*l/I)*sin(xn(1)) - (k/I)*(xn(1)-xn(3)) - (bl/I)*xn(2);
    ksi(4) = -(m*g*l/I)*xn(2)*cos(xn(1)) - (k/I)*(xn(2)-xn(4))...
             +(bl/I).^2*xn(2) + ((bl*m*g*l)/(I.^2))*sin(xn(1))...
             +((bl*k)/(I.^2))*(xn(1)-xn(3));

    *********************************************************************
    computation of the control input u

    z1 = sin(8*tn);
    z2 = cos(8*tn);
    v  = 4096*z1 - 10000*(ksi(1)-z1)...
                 -  4000*(ksi(2)-8*z2)...
                 -   600*(ksi(3)+64*z1)...
                 -    40*(ksi(4)+512*z2);
```

```
    rij = [m*g*l/I*xn(2)*sin(xn(1))+(bl*m*g*l/I.^2)*cos(xn(1))+(bl*k)/(I.^2);
           (-m*g*l/I)*cos(xn(1))-k/I+(bl/I).^2;-(bl*k)/I.^2;k/I]';
    kol = [xn(2);-(bl/I)*xn(2)-(m*g*l/I)*sin(xn(1))-(k/I)*(xn(1)-xn(3));
           xn(4);-(bm/J)*xn(4)+(k/J)*(xn(1)-xn(3))];
    LfT4 = rij*kol;
    LgT4 = k/(I*J);


     u = (1/LgT4)*(v-LfT4);

%   *********************************************************************
%   reset initial state and update output

    x0 = xn;
    tout = [tout;tn];
    xout = [xout xn];
    uout = [uout;u];

end

%************************************************************************
% output

z1d = sin(8*tout);
plot(tout,xout(1,:)',tout,z1d),
title('FEEDBACK LINEARIZATION CONTROL OF A DAMPED SYSTEM'),
xlabel('time [s]'),ylabel('ref. traj. & link angle'),pause
meta a:f145g


clg
plot(tout,uout),
title('FEEDBACK LINEARIZATION CONTROL OF A DAMPED SYSTEM'),
xlabel('time [s]'),ylabel('control input u'),pause
meta a:f145k
```

```
unction xdot=link(t,x)

       FEEDBACK LINEARIZATION CONTROL OF A
       SINGLE LINK WITH JOINT ELASTICITY

       inputfile on behalf of main program

**********************************************************************
       parameters used for simulation

 = 1;      % mass
 = 100;    % stiffness
 = 1;      % length
 = 9.81;   % gravity
 = 1;      % inertia
 = 1;      % inertia

2 = 150;   % infinite stiffness: 1.50*k !!!
2 = 175;   % infinite stiffness: 1.75*k !!!
2 = 250;   % infinite stiffness: 2.50*k !!!

**********************************************************************
       state equation describing plant

dot(1)= x(2);
dot(2)= -(m*g*l/I)*sin(x(1)) - (k/I)*(x(1)-x(3));
dot(3)= x(4);
dot(4)= (k/J)*(x(1)-x(3)) + (1/J)*u;
**********************************************************************
```

```
% FEEDBACK LINEARIZATION CONTROL OF A
% SINGLE LINK WITH JOINT ELASTICITY

% main program

%**************************************************************************
% define and initialize parameters

global u m g l I J k

t0 = 0.0;
x0 = [0 0 0 0]';
u  = 0.0;

tout = [t0];
xout = [x0];
uout = [u];

%**************************************************************************
% begin recursive calculation

tstart = 0.00;
tstep  = 0.01;
tstop  = 2.00;

for tel = tstart:tstep:tstop,

    t0 = tel;
    tf = tel + tstep;
    [t,x] = ode45('link',t0,tf,x0)

    n  = length(t);
    tn = t(n);
    xn = x(n,:)';

%   **********************************************************************
%   computation of coordinates transformation

    ksi(1) = xn(1);
    ksi(2) = xn(2);
    ksi(3) = -(m*g*l/I)*sin(xn(1)) - (k/I)*(xn(1)-xn(3));
    ksi(4) = -(m*g*l/I)*xn(2)*cos(xn(1)) - (k/I)*(xn(2)-xn(4));

%   **********************************************************************
%   computation of the control input u

    z1 = sin(8*tn);
    z2 = cos(8*tn);
    v  = 4096*z1 - 10000*(ksi(1)-z1)...
                 -  4000*(ksi(2)-8*z2)...
                 -   600*(ksi(3)+64*z1)...
                 -    40*(ksi(4)+512*z2);

    u  = -(m*g*l*J/k)*sin(xn(1))*(xn(2).^2+(m*g*l/I)*cos(xn(1))+k/I)...
         -J*(xn(1)-xn(3))*((m*g*l/I)*cos(xn(1))+k/I+k/J)...
         +(I*J/k)*v;

%   **********************************************************************
%   reset initial state and update output

    x0 = xn;
    tout = [tout;tn];
    xout = [xout xn];
    uout = [uout;u];

end

%**************************************************************************
% output

z1d = sin(8*tout);
plot(tout,xout(1,:)',tout,z1d),
title('FEEDBACK LINEARIZATION CONTROL'),
xlabel('time [s]'),ylabel('ref. traj. & link angle'),pause
meta a:f135g

clg
plot(tout,uout),
title('FEEDBACK LINEARIZATION CONTROL'),
xlabel('time [s]'),ylabel('control input u'),pause
meta a:f135k
```

III.3

FEEDBACK LINEARIZATION CONTROL OF A
SINGLE LINK WITH JOINT ELASTICITY

main program

```
*************************************************************************
 set starting values

lobal u m g l I J k k2

.0 = 0.0;
:0 = [0 0 0 0]';
  = 0.0;

.out = [t0];
:out = [x0];
iout = [u];

************************************************************************
 begin recursive calculation

:start = 0.00;
:step  = 0.01;
:stop  = 2.00;

for tel = tstart:tstep:tstop,

    t0 = tel;
    tf = tel + tstep;
    [t,x] = ode45('link',t0,tf,x0)

    n  = length(t);
    tn = t(n);
    xn = x(n,:)';

%   ************************************************************************
%   computation of coordinates transformation

    ksi(1) = xn(1);
    ksi(2) = xn(2);
    ksi(3) = -(m*g*l/I)*sin(xn(1)) - (k/I)*(xn(1)-xn(3));
    ksi(4) = -(m*g*l/I)*xn(2)*cos(xn(1)) - (k/I)*(xn(2)-xn(4));

%   ************************************************************************
%   computation of the control input u

    z1 = sin(8*tn);
    z2 = cos(8*tn);
    v  = 4096*z1 - 10000*(ksi(1)-z1)...
                 -  4000*(ksi(2)-8*z2)...
                 -   600*(ksi(3)+64*z1)...
                 -    40*(ksi(4)+512*z2);

    u  = -(m*g*l*J/k2)*sin(xn(1))*(xn(2).^2+(m*g*l/I)*cos(xn(1))+k2/I)...
         -J*(xn(1)-xn(3))*((m*g*l/I)*cos(xn(1))+k2/I+k2/J)...
         +(I*J/k2)*v;

%   ************************************************************************
%   reset initial state and update output

    x0 = xn;
    tout = [tout;tn];
    xout = [xout xn];
    uout = [uout;u];

end

%************************************************************************
% output

z1d = sin(8*tout);
plot(tout,xout(1,:)',tout,z1d),
title('RIGID CONTROL APPLIED TO FLEXIBLE JOINT'),
xlabel('time [s]'),ylabel('ref. traj. & link angle'),pause
meta a:f136g

clg
subplot(211)
plot(tout,xout(1,:)',tout,z1d),
title('RIGID CONTROL APPLIED TO FLEXIBLE JOINT'),
xlabel('time [s]'),ylabel('ref. traj. & link angle'),pause
meta a:f136k
```

Appendix IV - 1

Proof of the identity: $L_{[f,g]}h(x) = L_f L_g h(x) - L_g L_f h(x)$

$$L_{[f,g]}h(x) = \frac{\partial h}{\partial x}[f,g]$$

$$= \frac{\partial h}{\partial x}\left\{\frac{\partial g}{\partial x}f - \frac{\partial f}{\partial x}g\right\}$$

$$= \frac{\partial g}{\partial x}f \cdot \frac{\partial h}{\partial x} - \frac{\partial f}{\partial x}g \cdot \frac{\partial h}{\partial x}$$

$$= L_f g \cdot \frac{\partial h}{\partial x} - L_g f \cdot \frac{\partial h}{\partial x}$$

$$= L_f\left(\frac{\partial h}{\partial x}g\right) - L_g\left(\frac{\partial h}{\partial x}f\right)$$

$$= L_f L_g h(x) - L_g L_f h(x)$$

Appendix IV-2

In this section we show that the compact notation

$$\begin{cases} L_{ad_f^k(g)}T_1 = 0 & k=0,1,2,\dots,n-2 \\ L_{ad_f^{n-1}(g)}T_1 \neq 0 & \\ L_fT_n + (L_gT_n)u = v & \end{cases}$$

completely covers the three conclusions of section 2.2.2. We do this by carrying out the following inductive calculation:

* $k=0 \Rightarrow \underline{L_gT_1 = 0}$

* $k=1 \Rightarrow L_{ad_f^1(g)}T_1 = L_{[f,g]}T_1 = L_fL_gT_1 - L_g\underbrace{L_fT_1}_{=T_2}$
$= L_fL_gT_1 - \underline{L_gT_2}$
$= 0 - 0 = 0$

* $k=2 \Rightarrow L_{ad_f^2(g)}T_1 = 0 = L_{[f,[f,g]]}T_1 = 0 \Rightarrow$
$L_fL_{[f,g]}T_1 - L_{[f,g]}L_fT_1 = 0 \Rightarrow$
$L_f(L_fL_gT_1 - L_gL_fT_1) - (L_fL_g - L_gL_f)L_fT_1 =$
$L_fL_fL_gT_1 - L_fL_gL_fT_1 - L_fL_gL_fT_1 + L_gL_fL_fT_1 =$
$= L_fL_f\underbrace{L_gT_1}_{0} - 2L_fL_gL_fT_1 + L_gL_fL_fT_1 =$
$-2L_f\underbrace{L_gT_2}_{0} + L_gL_fT_2 =$
$+ \underbrace{L_gT_3}_{0} = 0$

⋮

* $k=n-1 \Rightarrow L_{ad_f^{n-1}(g)}T_1 =$
$L_{[f,ad_f^{n-2}(g)]}T_1 = L_fL_{ad_f^{n-2}(g)}T_1 - L_{ad_f^{n-2}(g)}L_fT_1$
$= L_gT_n \neq 0$!

$\Rightarrow L_{ad_f^{n-1}(g)}T_1 \neq 0$

Appendix IV-3 - Antwoorden

1) zie diktaat "werktuigkundig regelen IV"
   MatLab User Guide

2) model van het te lineariseren systeem

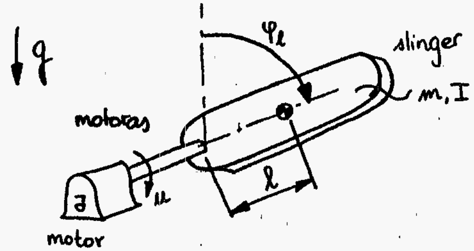


fig. 4.1

totale kinetische energie : $T = \frac{1}{2}(I+J)\dot{q}^2$

totale potentiële energie : $V = mgl(1-\cos q)$

$$\Rightarrow T_{,\dot{q}} = (I+J)\dot{q} \quad ; \; T_{,q} = 0$$

$$\frac{d}{dt}(T_{,\dot{q}}) = (I+J)\ddot{q}$$

$$V_{,q} = mgl \sin q$$

$$Q^* = u$$

Invullen in de vergelijking van Lagrange:

$$\frac{d}{dt}(T_{,\dot{q}}) - T_{,q} + V_{,q} = Q^* \Rightarrow$$

$$(I+J)\ddot{q} + mgl\sin q = u$$

Herschrijven naar toestandsvorm :

$$\underline{x}^T = [\,x_1 \quad x_2\,] \qquad \Rightarrow \qquad \dot{\underline{x}}^T = [\,\dot{x}_1 \quad \dot{x}_2\,]$$

$$[\,q \quad \dot{q}\,] \qquad\qquad [\,\dot{q} \quad \ddot{q}\,]$$

$$(I+J)\ddot{q} + mgl\sin q = u \Rightarrow$$

$$\ddot{q} = -\frac{mgl}{(I+J)}\sin q + \frac{1}{(I+J)}u \Rightarrow$$

$$\dot{\underline{x}} = \begin{bmatrix} x_2 \\ -\frac{mgl}{(I+J)}\sin x_1 + \frac{1}{(I+J)}u \end{bmatrix}$$

$$\dot{\underline{x}} = f(\underline{x}) + g(\underline{x})u \Rightarrow$$

$$f(\underline{x}) = \begin{bmatrix} x_2 \\ \alpha \sin x_1 \end{bmatrix} \qquad g(\underline{x}) = \begin{bmatrix} 0 \\ \beta \end{bmatrix}$$

met $\cdot\ \alpha = -\frac{mgl}{(I+J)}$

$\cdot\ \beta = \frac{1}{(I+J)}$

3) controle of het systeem "feedback linearizable" is.

formeel:

> I. de vectoren $\{ g, ad_f^1(g), \ldots, ad_f^{n-1}(g) \}$ vormen een onafhankelijk stelsel
>
> II. de vectoren $\{ g, ad_f^1(g), \ldots, ad_f^{n-2}(g) \}$ zijn "involutive"

In dit geval: $n=2 \Rightarrow$

* $ad_f^0(g) = g = [0 \quad \beta]^T$
* $ad_f^1(g) = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g$

$$= \underline{0} - \begin{bmatrix} 0 & 1 \\ \alpha \cos x_1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \beta \end{bmatrix} = \begin{bmatrix} -\beta \\ 0 \end{bmatrix}$$

$\Rightarrow$ I. het stelsel vectoren $\left\{ \begin{bmatrix} 0 \\ \beta \end{bmatrix}, \begin{bmatrix} -\beta \\ 0 \end{bmatrix} \right\}$ is onafhankelijk, immers

$P = \begin{bmatrix} 0 & -\beta \\ \beta & 0 \end{bmatrix} \Rightarrow \det(P) = \beta^2 \neq 0 \quad (\beta \neq 0)$

ofwel, $\text{rang}(P) = 2$

II. het stelsel vectoren $\left\{ \begin{bmatrix} 0 \\ \beta \end{bmatrix} \right\}$ is "involutive"

triviaal geval: $\begin{bmatrix} 0 \\ \beta \end{bmatrix}$ is namelijk te schrijven als een lineaire combinatie van $\begin{bmatrix} 0 \\ \beta \end{bmatrix}$ nl. $\begin{bmatrix} 0 \\ \beta \end{bmatrix} = 1 * \begin{bmatrix} 0 \\ \beta \end{bmatrix}$

4) berekening van de niet-lineaire transformatie op de toestandsvector

formeel:

> $L_{ad_f^k(g)} T_1 = 0 \qquad k = 0, 1, \ldots, n-2$
>
> $L_{ad_f^{n-1}(g)} T_1 \neq 0$

In dit geval:

* $k=0 \Rightarrow L_g T_1 = 0 \Rightarrow \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} \end{bmatrix} \begin{bmatrix} 0 \\ \beta \end{bmatrix} = 0 \Rightarrow$

$$\frac{\partial T_1}{\partial x_2} = 0 \qquad (\beta \neq 0)$$

* $k=1 \Rightarrow L_{ad_f^1(g)} T_1 \neq 0 \Rightarrow \begin{bmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} \end{bmatrix} \begin{bmatrix} -\beta \\ 0 \end{bmatrix} \neq 0 \Rightarrow$

$$\frac{\partial T_1}{\partial x_1} \neq 0 \qquad (\beta \neq 0)$$

We moet dus het volgende stelsel partiële dv oplossen : $\frac{\partial T_1}{\partial x_1} \neq 0$ en $\frac{\partial T_1}{\partial x_2} = 0$

Hieraan voldoet de meest eenvoudige oplossing : $T_1 = x_1$

formeel:

$$T_{i+1} = L_f T_i \qquad i = 1, \dots, n-1$$

In dit geval:

* $i = 1 \implies T_2 = L_f T_1 \implies$

$$T_2 = \left[\frac{\partial T_1}{\partial x_1} \quad \frac{\partial T_1}{\partial x_2} \right] \begin{bmatrix} x_2 \\ \alpha \sin x_1 \end{bmatrix}$$

$$= \left[1 \quad 0 \right] \begin{bmatrix} x_2 \\ * \end{bmatrix} = x_2$$

$\implies$ de niet-lineaire transformatie heeft de vorm :

$$\underline{\xi} = T(\underline{x}) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \implies$$

$$\underline{x} = T^{-1}(\underline{\xi}) = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$$

beide zijn differentieerbaar $\implies$ de afbeelding is een diffeomorfisme.

5) berekening van de niet-lineaire regelingang u

formeel:

$$L_f T_n + (L_g T_n) u = v \implies$$
$$u = \frac{1}{L_g T_n} (v - L_f T_n)$$

In dit geval: $n = 2 \implies$

* $L_g T_2 = \left[\frac{\partial T_2}{\partial x_1} \quad \frac{\partial T_2}{\partial x_2} \right] \begin{bmatrix} 0 \\ \beta \end{bmatrix} = \left[0 \quad 1 \right] \begin{bmatrix} 0 \\ \beta \end{bmatrix} = \beta$

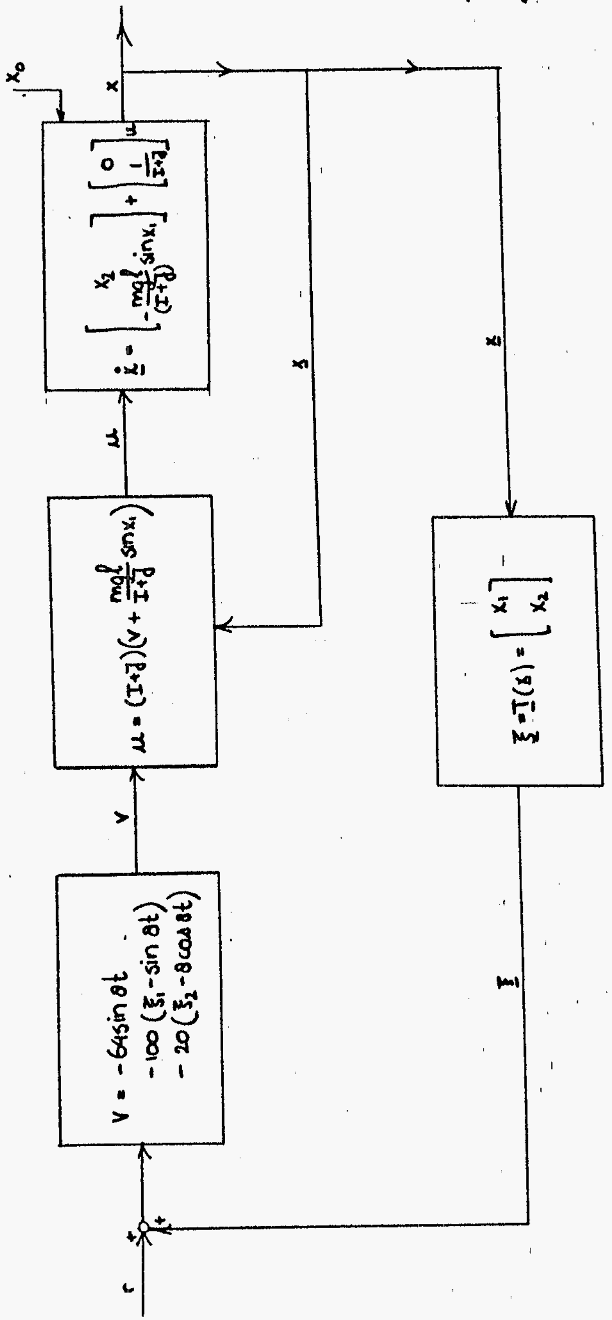
* $L_f T_2 = \left[\frac{\partial T_2}{\partial x_1} \quad \frac{\partial T_2}{\partial x_2} \right] \begin{bmatrix} x_2 \\ \alpha \sin x_1 \end{bmatrix} = \left[0 \quad 1 \right] \begin{bmatrix} x_2 \\ \alpha \sin x_1 \end{bmatrix}$

$$= \alpha \sin x_1$$

$$\implies u = \frac{1}{\beta} (v - \alpha \sin x_1)$$
$$= (I + J)\left(v + \frac{mgl}{(I+J)} \sin x_1\right)$$

<u>nb.</u> vergelijk dit resultaat met de formule die Spong geeft in [1].
(zie Appendix I : formule (5.3)) !

6 blokschema van het geregelde systeem:



7) we kiezen voor v:

$$v = \dot{\xi}_2^d - a_1(\xi_1 - \xi_1^d) - a_2(\xi_2 - \xi_2^d)$$

$$\dot{\underline{\xi}} = A\underline{\xi} + \underline{b}v \Leftrightarrow \dot{\underline{\xi}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \Rightarrow$$

$$\dot{\xi}_1 = \xi_2$$
$$\dot{\xi}_2 = v$$
$$= \dot{\xi}_2^d - a_1(\xi_1 - \xi_1^d) - a_2(\xi_2 - \xi_2^d) \Rightarrow$$

$$(\dot{\xi}_2 - \dot{\xi}_2^d) + a_2(\xi_2 - \xi_2^d) + a_1(\xi_1 - \xi_1^d) = 0 \Rightarrow$$

$$(\ddot{\xi}_1 - \ddot{\xi}_1^d) + a_2(\dot{\xi}_1 - \dot{\xi}_1^d) + a_1(\xi_1 - \xi_1^d) = 0$$

Stel: $e = \xi_1 - \xi_1^d \Rightarrow$

$$\ddot{e} + a_2\dot{e} + a_1 e = 0$$

met als karakteristieke vergelijking: $\lambda^2 + a_2\lambda + a_1 = 0$

Stel we kiezen de ligging van de polen (en dus de wortels van de karakteristieke vergelijking) in het punt $(-10,0) \Rightarrow$

$$a_2 = 20$$
$$a_1 = 100$$

$\Rightarrow$

$$v = \dot{\xi}_2^d - 100(\xi_1 - \xi_1^d) - 20(\xi_2 - \xi_2^d)$$

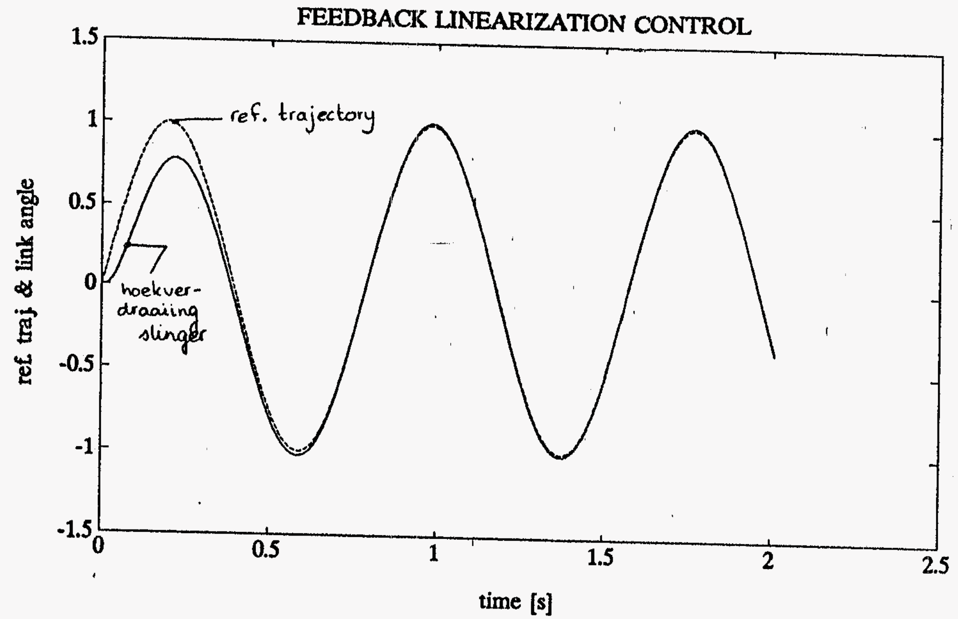
We kiezen als gewenste trajectorie $\xi_1^d = \sin 8t$

$$\Rightarrow \quad \xi_2^d = \dot{\xi}_1^d = 8\cos 8t$$

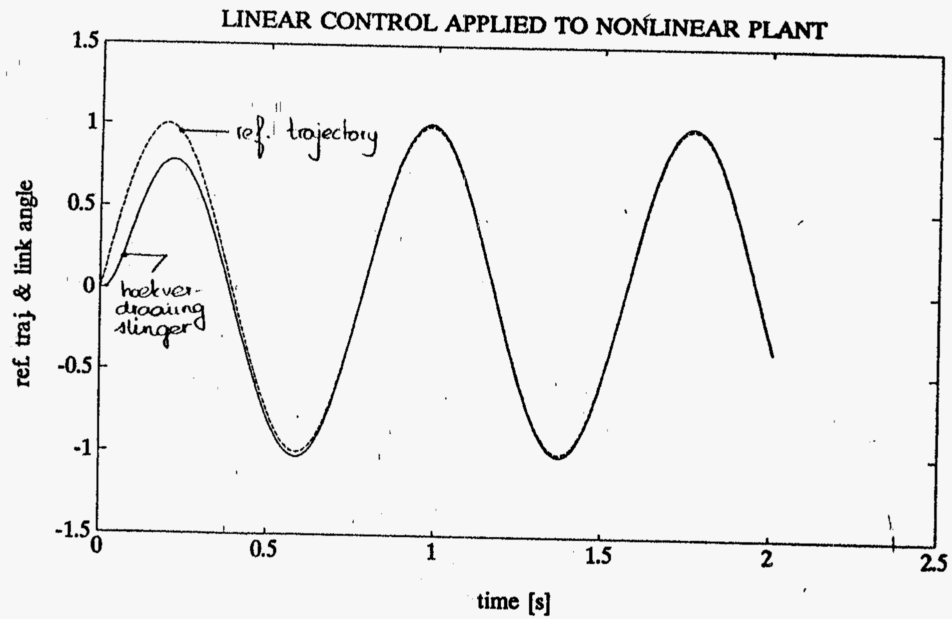
$$\dot{\xi}_2^d = -64 \sin 8t$$

$$\Rightarrow \quad v = -64\sin 8t - 100(\xi_1 - \sin 8t) - 20(\xi_2 - 8\cos 8t)$$

8)



9)



nb de bedoeling van opg. 9) is om te laten zien wat er gebeurt wanneer we een niet-lineair systeem regelen in de veronderstelling dat het systeem lineair is. We hebben daartoe de regelwet

$$u = (I+J)\left(v + \frac{mgl}{I+J}\sin x_1\right)$$

vervangen door

$$u = (I+J)\left(v + \frac{mgl}{I+J}x_1\right)$$

We stellen dus $\sin x_1 = x_1$.
Echter, we zien geen verschil met opg. 8) ???

```
unction xdot=link2(t,x)

       FEEDBACK LINEARIZATION CONTROL OF A
       SINGLE LINK WITH JOINT ELASTICITY

       inputfile on behalf of main program

*************************************************************************
       parameters used for simulation

 = 1;      % mass
 = 100;    % stifness
 = 1;      % length
 = 9.81;   % gravity
 = 1;      % inertia
 = 1;      % inertia

lfa = -(m*g*l)/(I+J);  % constant
eta = 1/(I+J);         % constant

*************************************************************************
       state equation describing plant

dot(1)= x(2);
dot(2)= alfa*sin(x(1)) + beta*u;
*************************************************************************
```

FEEDBACK LINEARIZATION CONTROL OF A
SINGLE LINK WITH JOINT ELASTICITY

IV.9

main program (opgave 8)

```
*********************************************************************
 set starting values

lobal u m g l I J k alfa beta

0 = 0.0;
0 = [0 0]';
  = 0.0;

out = [t0];
out = [x0];
out = [u];

*********************************************************************
 begin recursive calculation

start = 0.00;
step  = 0.01;
stop  = 2.00;

or tel = tstart:tstep:tstop,

    t0 = tel;
    tf = tel + tstep;
    [t,x] = ode45('link2',t0,tf,x0)

    n  = length(t);
    tn = t(n);
    xn = x(n,:)';

    *********************************************************************
    computation of coordinates transformation

    ksi(1) = xn(1);
    ksi(2) = xn(2);

    *********************************************************************
    computation of the control input u

    z1 = sin(8*tn);
    z2 = cos(8*tn);

    v  = -64*z1 - 100*(ksi(1)-z1) - 20*(ksi(2)-8*z2);

    u  = (1/beta)*(v-alfa*sin(xn(1)));

    *********************************************************************
    reset initial state and update output

    x0 = xn;
    tout = [tout;tn];
    xout = [xout xn];
    uout = [uout;u];

nd

 output

ld = sin(8*tout);
lot(tout,xout(1,:)',tout,z1d),
itle('FEEDBACK LINEARIZATION CONTROL'),
label('time [s]'),ylabel('ref. traj. & link angle'),pause
eta a:f143g
```

```
% SINGLE LINK WITH JOINT ELASTICITY

% main program (opgave 9)

%*********************************************************************
% set starting values

global u m g l I J k alfa beta

t0 = 0.0;
x0 = [0 0]';
u  = 0.0;

tout = [t0];
xout = [x0];
uout = [u];

%*********************************************************************
% begin recursive calculation

tstart = 0.00;
tstep  = 0.01;
tstop  = 2.00;

for tel = tstart:tstep:tstop,

     t0 = tel;
     tf = tel + tstep;
     [t,x] = ode45('link2',t0,tf,x0)

     n  = length(t);
     tn = t(n);
     xn = x(n,:)';

%    *******************************************************************
%    computation of coordinates transformation

     ksi(1) = xn(1);
     ksi(2) = xn(2);

%    *******************************************************************
%    computation of the control input u

     z1 = sin(8*tn);
     z2 = cos(8*tn);

     v  = -64*z1 - 100*(ksi(1)-z1) - 20*(ksi(2)-8*z2);

     u  = (1/beta)*(v-alfa*xn(1));

%    *******************************************************************
%    reset initial state and update output

     x0 = xn;
     tout = [tout;tn];
     xout = [xout xn];
     uout = [uout;u];

end

% output

z1d = sin(8*tout);
plot(tout,xout(1,:)',tout,z1d),
title('LINEAR CONTROL APPLIED TO NONLINEAR PLANT'),
xlabel('time [s]'),ylabel('ref. traj. & link angle'),pause
meta a:f144g
```