Modelling and control of IR/EO-gimbal for UAV surveillance applications

Examensarbete utfört i Reglerteknik
av

Per Skoglar

LiTH-ISY-EX-3258-2002

2002-06-06

Modelling and control of IR/EO-gimbal for UAV surveillance applications

Examensarbete utfört i Reglerteknik
vid Linköpings tekniska högskolan
av

Per Skoglar

LiTH-ISY-EX-3258-2002

Supervisor: **Jonas Nygårds**
Morgan Ulvklo
Jonas Gillberg

Examiner: **Mikael Norrlöf**

Linköping, 2002-06-06.

Avdelning, Institution
Division, Department

Institutionen för Systemteknik
581 83 LINKÖPING

Datum
Date
2002-06-06

Språk
Language
Svenska/Swedish
X Engelska/English

Rapporttyp
Report category
Licentiatavhandling
X Examensarbete
C-uppsats
D-uppsats
Övrig rapport

ISBN

ISRN LITH-ISY-EX-3258-2002

Serietitel och serienummer **ISSN**
Title of series, numbering

URL för elektronisk version
http://www.ep.liu.se/exjobb/isy/2002/3258/

Titel Modellering och styrning av IR/EO-gimbal för övervakning med UAV
Title
Modelling and control of IR/EO-gimbal for UAV surveillance applications

Författare Per Skoglar
Author

Sammanfattning
Abstract
This thesis is a part of the SIREOS project at Swedish Defence Research Agency which aims at developing a sensor system consisting of infrared and video sensors and an integrated navigation system. The sensor system is placed in a camera gimbal and will be used on moving platforms, e.g. UAVs, for surveillance and reconnaissance. The gimbal is a device that makes it possible for the sensors to point in a desired direction.

In this thesis the sensor pointing problem is studied. The problem is analyzed and a system design is proposed. The major blocks in the system design are gimbal trajectory planning and gimbal motion control. In order to develop these blocks, kinematic and dynamic models are derived using techniques from robotics. The trajectory planner is based on the kinematic model and can handle problems with mechanical constraints, kinematic singularity, sensor placement offset and reference signal transformation.

The gimbal motion controller is tested with two different control strategies, PID and LQ. The challenge is to perform control that responds quickly, but do not excite the damping flexibility too much. The LQ-controller uses a linearization of the dynamic model to fulfil these requirements.

Nyckelord
Keyword
IR/EO-gimbal, modelling, identification, control, path planning, sensor management, robotics

Abstract

This thesis is a part of the SIREOS project at Swedish Defence Research Agency which aims at developing a sensor system consisting of infrared and video sensors and an integrated navigation system. The sensor system is placed in a camera gimbal and will be used on moving platforms, e.g. UAVs, for surveillance and reconnaissance. The gimbal is a device that makes it possible for the sensors to point in a desired direction.

In this thesis the sensor pointing problem is studied. The problem is analyzed and a system design is proposed. The major blocks in the system design are gimbal trajectory planning and gimbal motion control. In order to develop these blocks, kinematic and dynamic models are derived using techniques from robotics. The trajectory planner is based on the kinematic model and can handle problems with mechanical constraints, kinematic singularity, sensor placement offset and reference signal transformation.

The gimbal motion controller is tested with two different control strategies, PID and LQ. The challenge is to perform control that responds quickly, but do not excite the damping flexibility too much. The LQ-controller uses a linearization of the dynamic model to fulfil these requirements.

Keywords:

IR/EO-gimbal, modelling, identification, control, path planning, sensor management, robotics

Acknowledgment

This thesis is the final work for the degree of Master of Science in Applied Physics and Electrical Engineering at Linköping Institute of Technology.

First of all I would like to thank my supervisors, Morgan Ulvklo and Dr. Jonas Nygårds at the Swedish Defence Research Agency (FOI) in Linköping, Sweden, for their guidance and for involving me in the SIREOS project.

I would also like to express my gratitude to my examiner Dr. Mikael Norrlöf for deep commitment to my work and for always finding time for discussions.

Furthermore, I am very grateful to Jonas Gillberg, Johan Stigwall and Karl-Johan Fredriksson for their valuable comments on this thesis. Also I will mention Frans Lundberg, Paul Nilsen and Andreas Karlsson for their useful contributions to my work.

Linköping, June 2002

Per Skoglar

Notation

Operators and functions

$\prod$	Scalar or matrix product $\prod_{i=1}^{n} A_i = A_1 A_2 ... A_n$
$Rot_{z,\theta}$	θ rad rotation counter-clockwise around the z-axis.
$Trans_{z,d}$	Translation along z-axis with distance d.
$diag[I_1, ..., I_2]$	Diagonal matrix with $I_1, ..., I_2$ as the diagonal elements.
$tr[A]$	The trace of matrix A.
c_θ	$\cos(\theta)$
s_θ	$\sin(\theta)$
$arctan2(y, x)$	Four quadrant arctangent of the real parts of x and y. $-\pi \leq arctan2(y, x) \leq \pi$
$Var[p]$	The variance of p.
$E[p]$	The mean value of p.

Abbreviations

D-H	Denavit-Hartenberg
DOF	Degrees Of Freedom
EO	Electro Optical
GPS	Global Positioning System
IMU	Inertial Measurement Unit
IR	Infrared
LQ	Linear Quadratic
LSE	Least Square Estimate
SIREOS	Signal Processing for Moving IR/EO Sensors
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle

Contents

Chapter 1

Introduction

1.1 The SIREOS project

The project *Signal Processing for Moving IR/EO Sensors* (SIREOS) is a strategic research project at the *Swedish Defence Research Agency* (FOI) in Linköping, Sweden. An experimental sensor system is being developed consisting of an IR sensor, a video camera and an integrated navigation system. The sensor system is placed in a *camera gimbal* and is giving georeferenced image sequences recorded for off-line processing and for development of real-time signal processing for unmanned aerial vehicles (UAVs). The navigation system combines GPS, inertial measurements from IMU and a digital compass to give a pointing accuracy in the same magnitude as the pixel uncertainty on the IR camera.

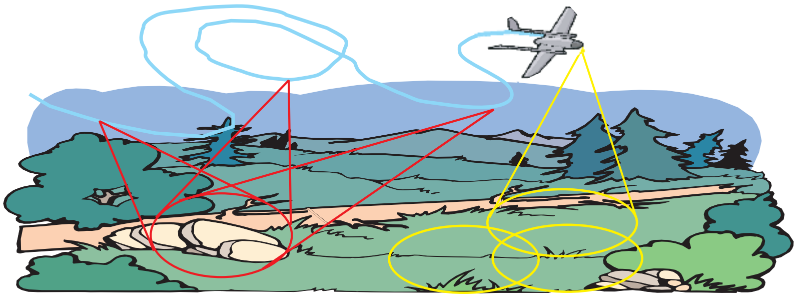


Figure 1.1. UAV surveillance and reconnaissance.

The SIREOS project focus is on developing algorithms for analysis of electro-optical and infrared image sequences from moving platforms for surveillance and reconnaissance purposes. The present system is intended to simultaneously perform wide area coverage, detect and track ground targets, perform detailed analysis of detected regions-of-interest and generate target coordinates by means of multi-view geolocation techniques, Figure 1.1.

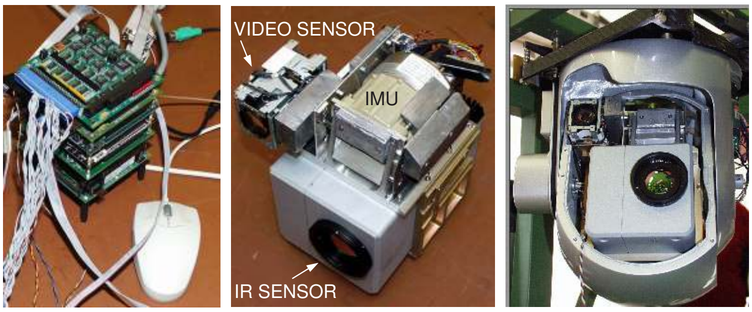


Figure 1.2. *Left:* The system is implemented in a PC-104 stack including dual Pentium III CPUs. *Middle:* The sensor assembly. *Right:* The sensor assembly mounted in the camera gimbal.

1.2 The Gimbal System

The camera gimbal is a modified camera platform from Polytech AB, Sweden, Figure 1.2 (right). The gimbal consists of two outer joints and two inner joints, Figure 1.3 (left). The outer joints, named pan and tilt respectively, are motor controlled and in the original configuration they are also gyro-stabilized. Pan joint has no mechanical constraints and can rotate continuous 360°, but tilt joint is constrained.

Inside the outer axes the cameras are mounted in a two axis inner gimbal. The inner axes have a freedom to move a few degrees and will prevent high frequency rotational vibrations to reach the cameras and causing blur in the images. A magnet provides a weak centering force to the inner axes with respect to the outer.

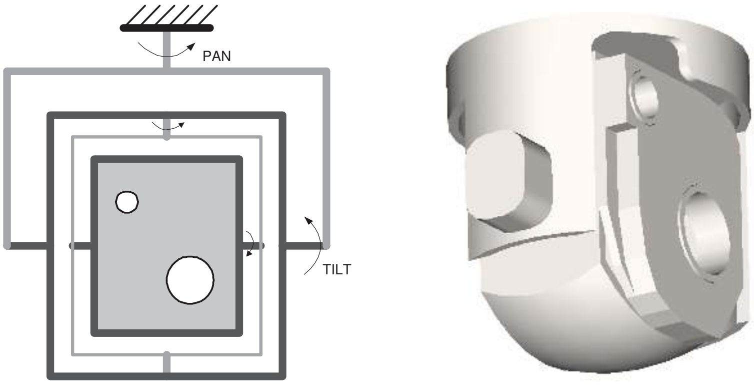


Figure 1.3. *Left:* Gimbal schematics, the four joints are marked. *Right:* CAD-model of the gimbal.

Digital encoders measure the pan and tilt angles and an optical angle sensor measures the angles of the inner gimbal. The navigation system will estimate the position of the camera and the orientation of its optical axes.

From the point-of-view of gimbal control, applications can be divided into two major groups, scanning and tracking. In scan applications the gimbal is controlled to point at desired coordinates, expressed in either a global or a local reference frame. In tracker applications the aim is to simultaneously keep a number of targets in the camera view. To manage these tasks in a satisfying way a detailed knowledge of the gimbal system is required.

The applications generate different reference signals that must be transformed to a new reference signal that can be used by the motion control system. Since the gimbal system has mechanical constraints and a singularity, the reference signals may then again be modified in a adequate way.

A regulator is to be designed to control the gimbal actuators to follow the reference signal. To respond quickly, but not exceed the operational range of the inner gimbal damping mechanism, the regulator must be based on a dynamic model of the gimbal. A dynamic model is also very useful for the purpose of simulations.

Path planning and control of the vehicle, the gimbal platform, is another important issue. In scan applications the vehicle will follow a reference trajectory. In tracker applications one might want the vehicle to be at a particular distance from the target or that the vehicle will keep the gimbal far from constraints and singularities in the joint angles.

1.3 Objectives

The discussion in the section above lead to the following objectives of this thesis.

- Analyze and breakdown the trajectory planning and control problem, with description of the applications as the start-point. Propose a system design and draw attention to difficulties.
- Derive a kinematic model of the gimbal for simulations and trajectory planning.
- Derive a dynamic model of the gimbal for simulations and control design.
- Suggest a solution to the trajectory planning problem.
- Suggest a solution to the gimbal control problem.
- Test and simulate the proposed system design.

1.4 Thesis Outline

The thesis can be divided into four parts, Figure 1.4. Chapter 2 provides some background information on the control system. The kinematics is treated in Chapters 3-5 and the dynamics is treated in Chapters 6-9. Finally Chapter 10 summarizes and evaluates the result.

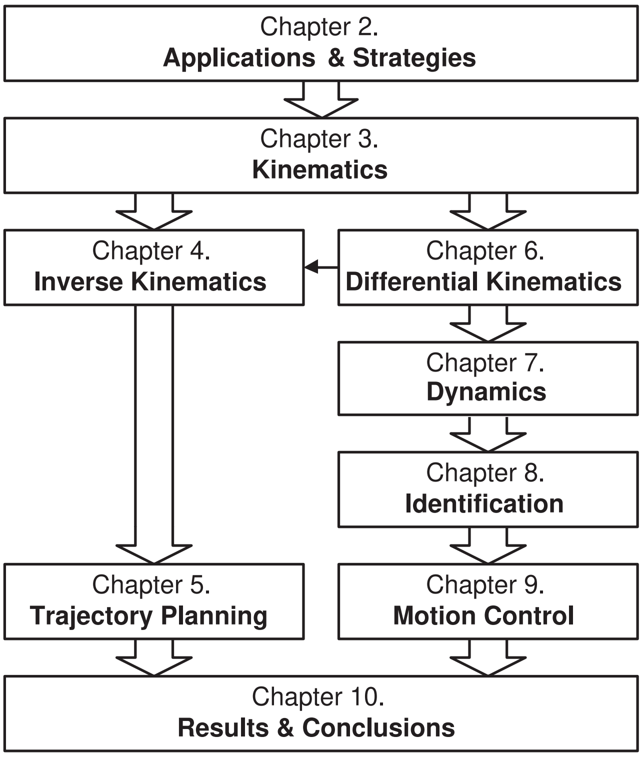


Figure 1.4. Thesis outline.

- Chapter 2 breaks down the trajectory planning and control problem and proposes a system design. This chapter identifies and describes problems that will be treated in the following chapters.

- Chapter 3 analyzes the mechanical structure, the kinematics, of the gimbal. A kinematic model is defined using techniques from robotics. The results in this chapter are used throughout the thesis.

- Chapter 4 treats the inverse kinematic problem and the results will be used in Chapter 5.

- Chapter 5 describes how the reference signals from different applications will be handled and transformed to motion control reference signals.

- Chapter 6 introduces the geometric Jacobian that will be used in Chapter 7.

- Chapter 7 derives a dynamic model of the gimbal system using Lagrange equations. This model will be used in simulations and in the motion control design.

- Chapter 8 proposes a strategy to solve the dynamic parameters identification problem.
- Chapter 9 develops and tests two different motion control strategies.
- Chapter 10 presents the results in larger simulations and experiments where all different parts are put together. The results are evaluated and suggestions to future work are given.

An asterisk (*) after the header of a section indicates that the understanding of the section is of less significance for the understanding of the rest of the thesis. Generally these sections are theoretical backgrounds or digressions. However, notice that an asterisk does not mean that the section is unimportant.

References & Further Readings

A detailed description of the SIREOS sensor system is given in [16]. A real-time tracker developed at FOI is described in [8], [7], [9].

Chapter 2

Applications & Strategies

This chapter serves as an introduction to the gimbal control system that will be developed in this thesis. Some applications are presented and a control system design that will serve these applications is proposed. The parts of this design are roughly described. In particular, the singularity and constraints problem is introduced.

2.1 Applications

Each application has a certain goal and generates one or more types of reference signals. In the section below a number of applications are described. They are divided into two major groups, scanning and tracking.

2.1.1 Scan Applications

Generally scan applications generates two references, a reference that represents the positions to point at and a reference that represents the vehicle trajectory.

Application 1 ("Continuous" scanning relative a global reference frame)
Task: *Scan an area by "continuously" moving the gimbal.*
Output Reference Signals: *Scan reference trajectory and vehicle reference trajectory expressed in global coordinates. Since we want to avoid the gimbal singularity, the trajectories should be synchronized. For instance, the gimbal could keep rotating while the vehicle moves forward which will give images of every area from two directions.*

Application 2 ("Discrete" scanning relative a global reference frame)
Task: *Fix a point for some time, then move the gimbal as quickly as possible and fix a new point, and so on.*
Output Reference Signals: *Reference points expressed in global coordinates and vehicle reference trajectory. As in the previous application the generation of points and vehicle trajectory should take the singularity into consideration.*

Application 3 (Continuous scanning relative a local reference frame)
Task: *Scan an area by "continuously" moving the gimbal. Same as Application 1 except that the trajectory is expressed in local coordinates.*
Output Reference Signals: *Scan reference trajectory expressed in local coordinates. Possibly vehicle reference trajectory.*

2.1.2 Tracker Applications

Application 4 (Track one target)
Task: *Center the target in the view.*
Output Reference Signals: *An error, the distance between the target and the image center in the image view.*

Application 5 (Track multiple targets, targets in same view)
Task: *"Center" targets in the view.*
Output Reference Signals: *An error.*

Application 6 (Track multiple targets, targets not in one view)
Task: *Track multiple targets. Sweep slowly over one target, go quickly to next target and sweep slowly over it, and so on.*
Output Reference Signals: *Reference points, possibly an error.*

The applications will not be treated further in this thesis since they are complex and requires separate treatment. Analysis and a solution to Application 4 can be found in [7], [8] and [9].

2.2 Control System Design

Figure 2.1 shows the proposed control system design. The control system can be divided into two major parts, gimbal and vehicle control. In this thesis the gimbal control part is of main interests. The applications generate different gimbal reference signals that first have to be transformed to a uniform and reasonable reference signal by a trajectory planner. A gimbal regulator controls the gimbal actuators to follow this reference signal. The structure of the vehicle control system is similar.

2.3 Gimbal Trajectory Planning

The gimbal trajectory planner transforms the reference signal from the applications to a reference signal that the gimbal controller can use. An important property of these new references is that they are executable, which means that the references are in the allowed range and are realistic. If the reference signals requires filtering, it too will be performed in this block.

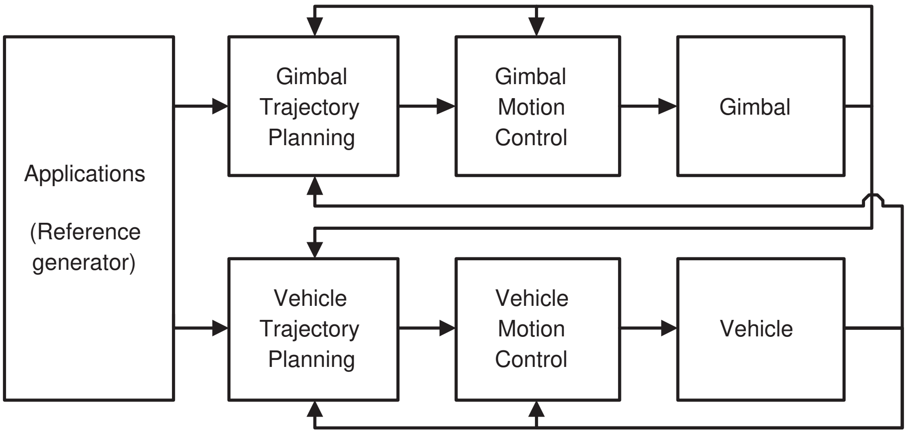


Figure 2.1. The control system design.

2.3.1 Trajectory Reference Types

The applications generate at least five different types of reference signals that we must manage:

1. Path motion, Relative Global Frame.
2. Path motion, Relative Local Frame.
3. Point-to-point, Relative Global Frame.
4. Point-to-point, Relative Local Frame.
5. Error in image view.

First we will transform all different reference signals to a new reference signal which is a path expressed in a local reference frame. The difference between reference type 1 and 2 is a reference frame transformation from global coordinates to local coordinates, the same for 3 and 4. The difference between a ”path” motion and ”point-to-point” motion when the gimbal moves from one point to another is that a ”path” describes how the gimbal should move between the points but in the ”point-to-point” case the gimbal should move along the fastest path between the points. Thus, in reference type 3 and 4 a path between the points must be generated. The error in image views, reference type 5, must be transformed to a path expressed in local coordinates.

If the path is affected by the constraints or the singularity it must be modified before the last transformation to an orientation reference that will be the input to the motion controller, Figure 2.2.

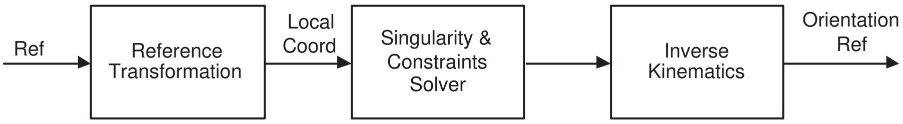


Figure 2.2. Overview of the trajectory planning system.

2.3.2 Singularity & Mechanical Constraints

Let the z-axis denote the extension of the rotational axis of the pan joint, Figure 2.3 (left). Along the z-axis, tilt axis −90°, the gimbal system is singular. This means that small changes of the point direction in the neighbourhood of the z-axis may cause large changes of the pan angle. For instance, when the gimbal is directed to the point $[x \quad y \quad z]^T = [\epsilon \quad 0 \quad -10]^T$ and we want the gimbal to point at $[0 \quad \epsilon \quad -10]^T$. If ϵ is very small this is a very small change in the cartesian frame, but a large change (90°) in the pan angle, Figure 2.3 (right).

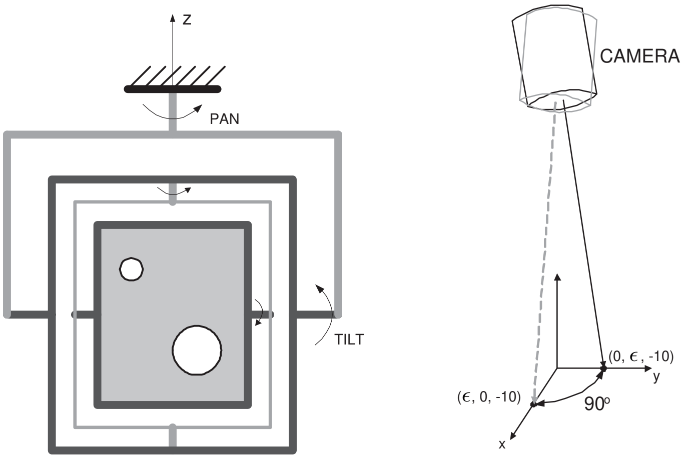


Figure 2.3. *Left:* Gimbal system with pan and tilt joints marked. *Right:* a small change in the cartesian frame may cause a large change in the pan angle.

Tilt angle is 0 in the posture in the figure and negative when faced down. The allowed ranges for pan angle and tilt angle are

$$\begin{array}{ccccc} -\infty & < & \phi_{pan} & < & \infty \\ \phi_{tilt}^{min} & \leq & \phi_{tilt} & \leq & \phi_{tilt}^{max} \end{array} \tag{2.1}$$

where

$$\phi_{tilt}^{max} \approx 10^\circ \tag{2.2}$$

$$\phi_{tilt}^{min} \approx -100^\circ \tag{2.3}$$

Thus pan angle has no constraints but tilt angle do have. Since $\phi_{tilt} = -90°$ when the camera is faced down, we see that the gimbal only can "look back" a few degrees. Hence the gimbal has to rotate 180° along the pan axis to "look back", an operation which takes a lot of time.

If the gimbal had to follow the reference exactly, the singularity and the constraints would have caused disastrous performance problems. Fortunately we work with image views where the target do not have to be in the center. However, the problem remains and we must find solutions that handle the singularity and the constraints in such a way that the risk of losing the target is small.

Assume that we track a target that moves from one side of the singularity to the other side. We now have two alternatives when we keep following the target.

- "Look back", i.e. let the tilt angle be $\phi_{tilt} < -90°$.
- Start rotating the pan angle.

Both alternatives have positive and negative aspects and it is not obvious which alternative to choose. If the gimbal looks back we can track faster targets, but what happens if the target intersect the maximum tilt angle? If the gimbal starts rotating we can only track slow targets, but once we have rotated there are no problems with tilt constraints. A suggestion is to look back when the future path is known, and of course in allowed angle ranges, and rotate in all other cases. An alternative suggestion is always to rotate and never look back.

The gimbal planning problem will be treated in Chapters 4 and 5. These chapters require a detailed knowledge about the kinematics of the gimbal system, a knowledge that we will derive in Chapter 3.

2.4 Gimbal Motion Control

The aim of the motion controller is to steer the gimbal to follow the orientation reference provided by the trajectory planner as well as possible, Figure 2.4. This task is a non trivial task since the gimbal is a nonlinear multivariable system.

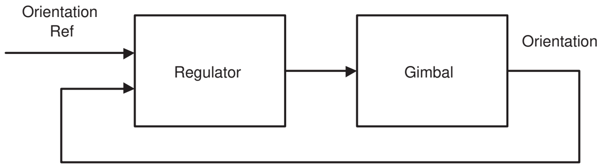


Figure 2.4. Gimbal regulator.

The purpose of the inner damping flexibility is to prevent high frequency rotational vibrations to reach the cameras, causing blur in the images. A drawback that comes with this is that large accelerations or decelerations excite the flexibility, which will cause that the inner camera gimbal will oscillate. While designing the regulator, these aspects have to be taken into account.

The gimbal motion control problem will be treated in Chapter 9 and is based on the dynamic model derived in Chapter 7.

2.5 Vehicle Trajectory Planning & Motion Control

In scan applications the reference trajectory for the vehicle is given. However, in tracker applications it might be desirable to simplify the task of the planner and controller by moving the vehicle in such a way that the gimbal is far from the constraints or the singularity. It may also be desirable to keep a certain distance to the target, i.e. follow the target when it moves.

The vehicle trajectory planning and motion control problems are beyond the scope of this thesis and is left for future work.

Chapter 3

Forward Kinematics

The *Forward Kinematic Problem* is the problem of determining the point direction of the camera when the joint variables are given. First we introduce a compact representation called homogenous transformation for describing rigid motions. Then we present definitions from robotics and compare the gimbal system with a robot manipulator. Finally we will apply a commonly used strategy in robotics, called Denavit-Hartenberg convention, to get a kinematic representation of the gimbal system.

3.1 Homogenous Transformations

The state of a rigid body (in this case a link) is completely described by its position and orientation with respect to a reference frame. Consider an arbitrarily point P in space. Let $o_0x_0y_0z_0$ (frame 0) and $o_1x_1y_1z_1$ (frame 1) be two different coordinate frames. The vectors $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$ are the coordinate of P with respect to frame 0 and frame 1 respectively. Then the relationship between $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$ can be expressed as

$$\boldsymbol{p}^0 = \boldsymbol{d}_0^1 + R_0^1\boldsymbol{p}^1 \tag{3.1}$$

where $\boldsymbol{d}_0^1$ is the vector from o_0 to o_1 and R_0^1 is the rotation matrix, Figure 3.1. The vector $\boldsymbol{d}_0^1$ describes the translation from frame 0 to frame 1 while the matrix R_0^1 describes the orientation of frame 1 with respect to frame 0. It is important that $R_0^1 \in SO(3)$. [1]

A more compact representation of the relationship can be achieved by introducing the *homogenous representation* $\tilde{\boldsymbol{p}}$ of a generic vector $\boldsymbol{p}$ as

$$\tilde{\boldsymbol{p}} = \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} \tag{3.2}$$

[1] $SO(3)$ stands for *Special Orthogonal group of order 3* and is the set of all 3×3 matrices that is orthogonal and whose determinant is $+1$.

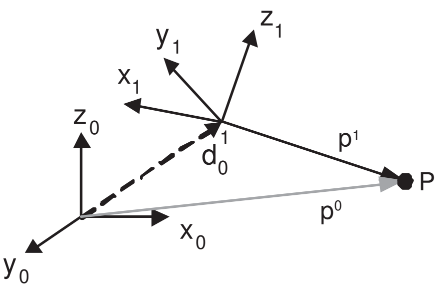


Figure 3.1. Rigid motions, a translation and a rotation.

The coordinate transformation (3.1) can then be compactly rewritten as

$$\tilde{\boldsymbol{p}}^0 = A_0^1 \tilde{\boldsymbol{p}}^1 \tag{3.3}$$

where A_0^1 denotes the 4 × 4 matrix

$$A_0^1 = \begin{bmatrix} R_0^1 & \boldsymbol{d}_0^1 \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{3.4}$$

A_0^1 is termed *homogenous transformation matrix.* It is easy to verify that a sequence of coordinate transformations can be expressed as the product

$$\tilde{\boldsymbol{p}}^0 = \prod_{k=1}^{n} A_{k-1}^k \tilde{\boldsymbol{p}}^n \tag{3.5}$$

3.2 Components and Structure

Robot manipulators are composed of *links* connected by *joints* into a kinematic chain. In this text we assume that the structure is an *open kinematic chain.* A kinematic chain is termed open when there is only one sequence of links connecting the two ends of the chain. One end is connected to a *base* while an *end effector* is mounted on the other end. Joints are typically either *revolute* or *prismatic.* A revolute joint realizes a relative rotation between two links and a prismatic joint realizes a relative translation between two links, Figure 3.2.

In an open kinematic chain, each joint provides the structure with a single *degree of mobility.* If the degrees of mobility of a manipulator are properly distributed along the structure, the number of joints determines the *degrees of freedom* (DOF) of the manipulator. The most general case, arbitrarily position and orientation, require 6 DOF. 3 DOF for positioning a point and 3 DOF for orienting with respect to a reference frame.

The gimbal system consist of two revolute joints that are controllable by two DC-motors. In addition we may consider the variable distance from the camera to

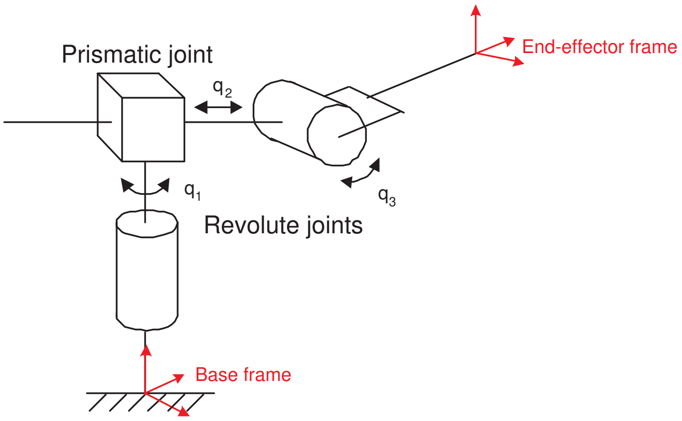


Figure 3.2. Example of an open kinematic chain with prismatic and revolute joints.

the object as a link with a prismatic joint, despite the fact that this link is neither a physically link nor controllable. The length of this artificial link depends on the distance from the camera to the object that the camera points at. In this sense the gimbal system is a 3 DOF system, but with only two direct controllable joints. This means that we are not able to select an arbitrarily position and orientation, but it is sufficient to select an arbitral position which is our goal. Moreover, the gimbal system also has a mechanical damping flexibility which we will model as two revolute joints. These joints are of course not directly controllable.

3.3 Kinematic chain

Consider a robot manipulator whose structure describes an open kinematic chain and whose joints are revolute or prismatic. Suppose the robot has $n+1$ links numbered from 0 to n starting from the base. Let the joints be numbered from 1 to n where joint i connects link $i-1$ and i. The i-th joint variable is denoted by q_i and is the angle of rotation for a revolute joint or displacement in the case of prismatic joint. Then attach a coordinate frame to each link. Let frame 0 be attached to the base and then choose frames 1 through n such that the positions of all points on link i are constant when expressed in the i-th coordinate frame.

Suppose A_i is the homogenous matrix that transforms the coordinates of a point from frame i to frame $i-1$. If all joints are either revolute or prismatic A_i is a function of only q_i. According to (3.3) we can express the position and orientation of the end-effector in the base frame by a homogenous matrix

$$T_0^n = \prod_{i=1}^{n} A_i \tag{3.6}$$

3.4 Denavit-Hartenberg Convention

In general, frames can be arbitrarily chosen as long as they are attached to their respective link, but it is helpful to be systematic in the choice of frames. A commonly used convention in robotic applications is the *Denavit-Hartenberg convention* (D-H), where A_i is a product of four basic transformations

$$\boldsymbol{A}_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

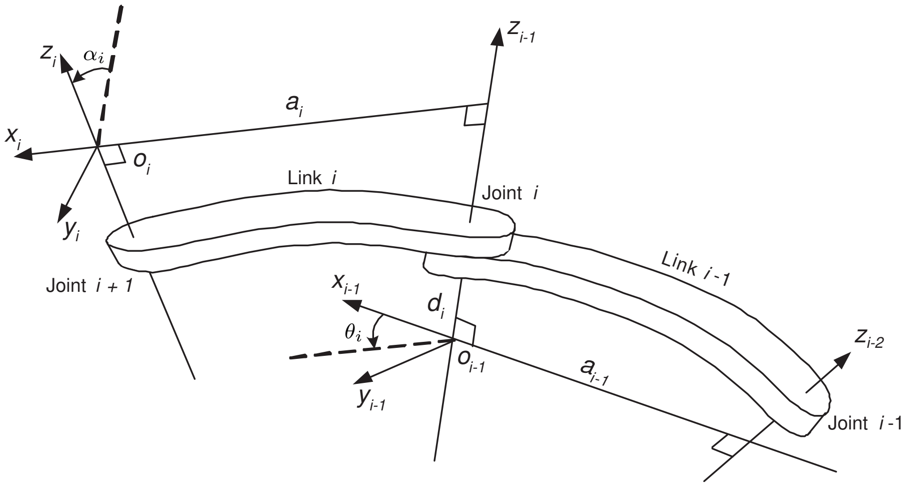


Figure 3.3. Denavit-Hartenberg kinematic parameters.

By the Denavit-Hartenberg convention the procedure for deriving the forward kinematics for a manipulator can be summarized in the following straightforward algorithm [18].

Denavit-Hartenberg algorithm *

Step 1. Locate and label the joint axes z_0,...,z_{n-1}.

Step 2. Establish the base frame. Set the origin anywhere on the z_0-axis. The x_0 and y_0 axes are chosen conveniently to form a right-hand frame.

For i=1,...,n-1, perform Steps 3-5.

Step 3. Locate the origin o_i where the common normal to z_i and z_{i-1} intersects z_i, Figure 3.3. If z_i intersects z_{i-1} locate o_i at this intersection. If z_i and z_{i-1} are parallel, locate o_i at joint i.

Step 4. Establish x_i along the common normal between z_{i-1} and z_i through o_i, or in the direction normal to the z_{i-1}-z_i plane if z_{i-1} and z_i intersect.

Step 5. Establish y_i to complete a right-hand frame.

Step 6. Establish the end-effector frame.

Step 7. Create a table of link parameters α_i, a_i, θ_i, d_i where

α_i = angle between z_{i-1} and z_i measured about x_i, positive direction is counter-clockwise.

a_i = distance along x_i from o_i to the intersection of the x_i and z_{i-1} axes.

θ_i = angle between x_{i-1} and x_i measured about z_{i-1}, positive direction is counter-clockwise. θ_i is variable if joint i is revolute.

d_i = distance along z_{i-1} from o_{i-1} to the intersection of the x_i and z_{i-1} axes. d_i is variable if joint i is prismatic.

Step 8. Form the homogenous transformation matrices A_i (3.7) and T_0^n (3.6).

3.5 D-H Approaches to the Gimbal Kinematics

The structure of the gimbal system appear to be rather simple, but unfortunately it is not. The cameras are not located on the two rotations axes, so they will also translate when an axis rotates. When using the gimbal in short distance surveillance, e.g. on a UGV, the camera offset is significant. Furthermore, we can not suppose that the cameras are pointing right ahead. However, for the purpose of simplicity we suppose that the two rotation axes are perpendicular. The mechanical damping system will be considered as two extra revolute joints.

Below we have some different approaches to describing the gimbal kinematics. Tables 3.1, 3.2, 3.3 are for determining the position and orientation of the end-effector frame. Table 3.4 is for determining the position and orientation of the body that the cameras are attached to. This approach will be used in Chapter 7. Also some of the homogenous transformation matrices are calculated according to (3.6) and (3.7). In the tables below the ”P/R” column shows whether the joint is revolute (R) or prismatic (P). * indicates that the parameter is variable. Note that there are different parameters for each camera.

D-H Approach 1 (Spherical Approximative)

If we ignore the camera offset from the rotations axes and the inner gimbal angles, the resulting approximative model will be a spherical system, Figure 3.4. This model is valid if the distance to the object is very long.

Joint/Link i	α_i [rad]	a_i [m]	θ_i [rad]	d_i [m]	R/P
1	$-\pi/2$	0	θ_1^*	d_1	R
2	$\pi/2$	0	θ_2^*	0	R
3	0	0	0	d_3^*	P

Table 3.1. Parameters for spherical approximative system, D-H Approach 1.

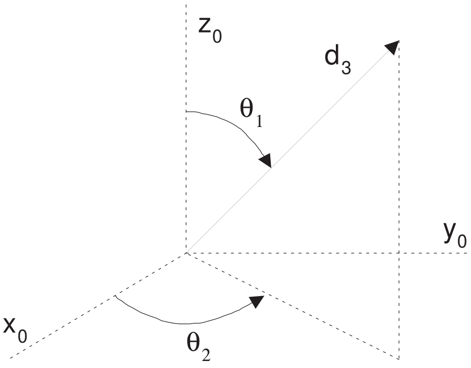


Figure 3.4. Denavit-Hartenberg description in D-H Approach 1. ($d_1 = 0$)

$$T_0^3 = \begin{bmatrix} c_{\theta_1}c_{\theta_2} & -s_{\theta_1} & c_{\theta_1}s_{\theta_2} & d_3c_{\theta_1}s_{\theta_2} \\ s_{\theta_1}c_{\theta_2} & c_{\theta_1} & s_{\theta_1}s_{\theta_2} & d_3s_{\theta_1}s_{\theta_2} \\ -d_3s_{\theta_2} & 0 & c_{\theta_2} & d_1 + d_3c_{\theta_2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Note the first three elements of the fourth column are the end-effector position. Compare these expressions with the spherical coordinates in e.g. [14].

D-H Approach 2 (Exact Minimal Representation Without Flexibility)

Including the camera offset we still can express the kinematics with three joints. However, this representation is not very useful. The camera is not located in a frame origin and the parameters are not so intuitive.

Joint/Link i	α_i [rad]	a_i [m]	θ_i [rad]	d_i [m]	R/P
1	$\pi/2$	0	θ_1^*	d_1	R
2	α_2	a_2	θ_2^*	d_2	R
3	α_3	0	0	d_3^*	P

Table 3.2. Parameters for exact minimal representation without damping flexibility, D-H Approach 2.

D-H Approach 3 (Exact Non-Minimal Representation With Flexibility)

Here we include the damping flexibility. Moreover, we represent the offsets of the camera with three prismatic joints and the "split vision" with two revolute joints, Figure 3.5. In fact these extra joints are constants, which we denote with an extra "C" in the "R/P" column. The elements of the resulting homogenous transformation matrix T_0^{10} can be found in Appendix B.

Joint/Link i	α_i [rad]	a_i [m]	θ_i [rad]	d_i [m]	R/P
1	$-\pi/2$	0	θ_1^*	d_1	R
2	$\pi/2$	0	θ_2^*	0	R
3	$-\pi/2$	0	θ_3^*	0	R
4	$\pi/2$	0	θ_4^*	0	R
5	$-\pi/2$	0	$-\pi/2$	d_5	PC
6	$\pi/2$	0	$-\pi/2$	d_6	PC
7	$\pi/2$	0	π	d_7	PC
8	$-\pi/2$	0	θ_8	0	RC
9	$\pi/2$	0	θ_9	0	RC
10	0	0	$\pi/2$	d_{10}^*	P

Table 3.3. Parameters for more intuitive nonminimal representation, D-H Approach 3.

D-H Approach 4 (Representation With Flexibility for Dynamic Analysis)

In the dynamic analysis in Chapter 7 we are only interested in the kinematics of the body that the cameras are attached to. So we consider the first four links from D-H Approach 3. The distance d_1 is here of no interest so we just set $d_1 = 0$.

The fourth column in the homogenous transformation matrix (3.9) *will be* $[0\ 0\ 0\ 1]^T$, *since we never move the origin of any frame, we just change the orientation. Furthermore, we approximate the flexibility angles θ_3 and θ_4. Since the angles are small we can approximate* $\sin(\theta_i)$ *with* θ_i *and* $\cos(\theta_i)$ *with* 1, $i = 3, 4$.

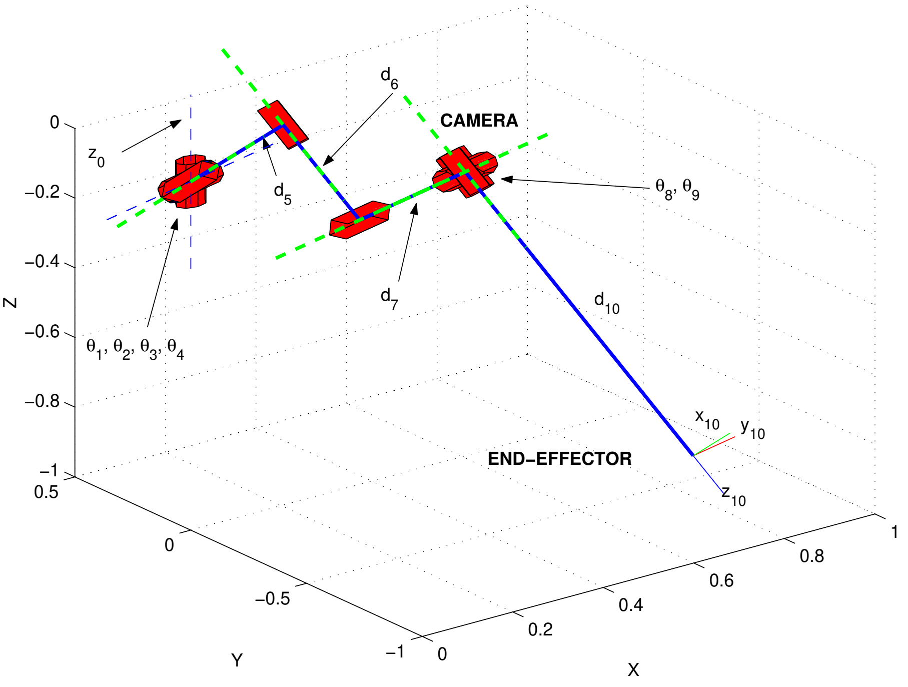


Figure 3.5. Denavit-Hartenberg description in D-H Approach 3.

Joint/Link i	α_i [rad]	a_i [m]	θ_i [rad]	d_i [m]	R/P
1	$-\pi/2$	0	θ_1^*	0	R
2	$\pi/2$	0	θ_2^*	0	R
3	$-\pi/2$	0	θ_3^*	0	R
4	$\pi/2$	0	θ_4^*	0	R

Table 3.4. Parameters with inner flexibility for Dynamic Analysis, D-H Approach 4.

$$T_0^4 = \begin{bmatrix} -\theta_3 s_{\theta_1} + c_{\theta_1}(c_{\theta_2} - \theta_4 s_{\theta_2}) & -\theta_3 c_{\theta_1} c_{\theta_2} - s_{\theta_1} & -\theta_3\theta_4 s_{\theta_1} + c_{\theta_1}(\theta_4 c_{\theta_2} + s_{\theta_2}) & 0 \\ \theta_3 c_{\theta_1} + s_{\theta_1}(c_{\theta_2} - \theta_4 s_{\theta_2}) & c_{\theta_1} - \theta_3 c_{\theta_2} s_{\theta_1} & \theta_3\theta_4 c_{\theta_1} + s_{\theta_1}(\theta_4 c_{\theta_2} + s_{\theta_2}) & 0 \\ -\theta_4 c_{\theta_2} - s_{\theta_2} & \theta_3 s_{\theta_2} & c_{\theta_2} - \theta_4 s_{\theta_2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

In all approaches above θ_1 and θ_2 correspond to the pan and tilt angle respectively and in approaches 3 and 4 the angles of the inner gimbal joints are θ_3 and θ_4. Notice that θ_2 and ϕ_{tilt} (used in Chapter 2) are defined in different ways and the relation is

$$\phi_{tilt} = -\theta_2 \tag{3.10}$$

3.6 Joint Space, Operational Space & Workspace*

In Section 3.3 we have seen that it is possible to describe the posture of a robot manipulator by a number of independent parameters

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\phi} \end{bmatrix} \tag{3.11}$$

where $\boldsymbol{p}$ and $\boldsymbol{\phi}$ are the position and the orientation of the end-effector. The vector $\boldsymbol{x}$ is defined in the space in which the manipulator task is specified. This space is called *operational space.*

We can also form a vector of joint variables

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} \tag{3.12}$$

where $q_i = \theta_i$ for a revolute joint and $q_i = d_i$ for a prismatic joint. The space where the joint variables are defined is called *joint space.* The direct kinematic equations can then be written in a compact form different from (3.6), i.e.

$$\boldsymbol{x} = \boldsymbol{k}(\boldsymbol{q}) \tag{3.13}$$

Thus we can compute the operational space variables from the knowledge of the joint space variables. Note that the rotational part $\boldsymbol{\phi}(\boldsymbol{q})$ of $\boldsymbol{k}(\boldsymbol{q})$ is not usually available in direct form. Fortunately we do not need this part in our case.

The *workspace* of a manipulator is the total space volume of the environment that can be accessed by the origin of the end effector. The workspace can be achieved by considering the position part of $\boldsymbol{x}$, i.e.

$$\boldsymbol{p} = \boldsymbol{p}(\boldsymbol{q}) \qquad q_i^{min} \leq q_i \leq q_i^{max} \qquad i = 1, ..., n \tag{3.14}$$

For the gimbal system the workspace is more interesting than operational space. The workspace of the gimbal system is all points that the cameras can point at and depends on which D-H approach we consider.

D-H Approach 1 (Spherical Approximative)
Approximately the gimbal system can be described by a spherical system where θ_1 and θ_2 are the angles of the first respectively second joint and d_3 is the distance to the target, Figure 3.4. Thus

$$\boldsymbol{q} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ d_3 \end{bmatrix} \tag{3.15}$$

Then the workspace of the gimbal system is defined by

$$\boldsymbol{p} = \boldsymbol{p}(\theta_1, \theta_2, d_{10}) \qquad \begin{array}{ccccc} -\infty & < & \theta_1 & < & \infty \\ \theta_{min} & \leq & \theta_2 & \leq & \theta_{max} \\ d_{10_{min}} & \leq & d_3 & < & \infty \end{array} \tag{3.16}$$

where $\boldsymbol{p}$ *is given by the first three elements of the fourth column in the homogenous transformation matrix* (3.8). *If we consider* $0 \leq \theta_1 < 2\pi$ *there are an infinite number of solutions along the z-axis and two solutions in the area near the z-axis, because* $\theta_{max} > \pi$.

D-H Approach 3 (Exact Non-Minimal Representation With Flexibility)
Joint variables are

$$\boldsymbol{q} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad d_{10}]^T \tag{3.17}$$

Now the workspace of the gimbal system is defined by

$$\boldsymbol{p} = \boldsymbol{p}(\theta_1, \theta_2, \theta_3, \theta_4, d_{10}) \quad \begin{array}{ccccc} -\infty & < & \theta_1 & < & \infty \\ \theta_{2_{min}} & \leq & \theta_2 & \leq & \theta_{2_{max}} \\ \theta_{3_{min}} & \leq & \theta_3 & \leq & \theta_{3_{max}} \\ \theta_{4_{min}} & \leq & \theta_4 & \leq & \theta_{4_{max}} \\ d_{10_{min}} & \leq & d_{10} & < & \infty \end{array} \tag{3.18}$$

$\boldsymbol{p}$ *is given by the first three elements of the fourth column in the homogenous transformation matrix* T_0^{10}. *The elements of the matrix* T_0^{10} *can be found in Appendix B.*

3.7 Parameter Measurements

The Denavit-Hartenberg parameters in the kinematic model need to be computed as exactly as possible. There exist kinematic calibration techniques [17] to find accurate estimates of D-H parameters from measurements on the end-effectors location. However, simple measurements with a measurement-rod will provide sufficient accuracy, since we do not require the same accuracy as in robot applications. Furthermore the significance of the parameters decrease when the distance to the target increase. The angles θ_8, θ_9 have not been measured. We assume that these angles are 0 until further notice. The accuracy of the d_i parameter measurements is estimated to be about $\pm 5mm$.

D-H Parameter	Value
d_1	-260 mm
d_5	127 mm
d_6	55 mm
d_7	20 mm
θ_8	(0)
θ_9	(0)

Table 3.5. D-H parameters (D-H Approach 3) for the IR sensor.

D-H Parameter	Value
d_1	-260 mm
d_5	106 mm
d_6	-57 mm
d_7	-86 mm
θ_8	(0)
θ_9	(0)

Table 3.6. D-H parameters (D-H Approach 3) for the video camera.

References & Further Readings

Sections 3.1, 3.2, 3.3, 3.4 are summaries of the corresponding sections in [15], [17] and [18]. Robotics Toolbox [1] for Matlab was used to plot Figure 3.5.

Chapter 4

Inverse Kinematics

In the forward kinematic problem we determined the position and orientation of the end effector in terms of the joint variables. The *Inverse Kinematic problem* on the other hand is the problem of finding the joint variables given the end-effector position and orientation. In general, this problem is more difficult than the forward kinematic problem.

4.1 Closed Form Solution

In solving the inverse kinematic we prefer a closed form solution of the equations rather than a numerical solution. Thus finding an explicit expression:

$$q_k = f_k(T_{11}, ..., T_{34}), \quad k = 1, ..., n \tag{4.1}$$

where T_{ij} are the elements in $\boldsymbol{T}_0^n$. These equations are the solutions to the 12 equations

$$T_{ij}(q_1, ..., q_n) = h_{ij}, \quad i = 1, ..., 3, \quad j = 1, ..., 4 \tag{4.2}$$

where h_{ij} describes the desired transformation. Generally these equations are nonlinear and much too difficult to solve directly in closed form.

In our case we are only interesting in the position and we will only solve three equations [1]

$$T_{14}(q_1, ..., q_n) = x \tag{4.3}$$

$$T_{24}(q_1, ..., q_n) = y \tag{4.4}$$

$$T_{34}(q_1, ..., q_n) = z \tag{4.5}$$

These equations can easily be solved if we consider D-H Approach 1 from Section 3.5. D-H Approach 3 on the other hand, is much harder to solve. The equations

[1] Compare with the position part of (3.13).

we will solve are then

$$\begin{array}{rcl} x & = & -s_{\theta_1}(d_7 + d_{10}s_{\theta_8}s_{\theta_9}) + c_{\theta_1}(c_{\theta_2}(d_6 + d_{10}c_{\theta_9}) + s_{\theta_2}(d_5 - d_{10}c_{\theta_8}s_{\theta_9})) \\ y & = & c_{\theta_1}(d_7 + d_{10}s_{\theta_8}s_{\theta_9}) + s_{\theta_1}(c_{\theta_2}(d_6 + d_{10}c_{\theta_9}) + s_{\theta_2}(d_5 - d_{10}c_{\theta_8}s_{\theta_9})) \\ z & = & d_1 - (d_6 + d_{10}c_{\theta_9})s_{\theta_2} + c_{\theta_2}(d_5 - d_{10}c_{\theta_8}s_{\theta_9}) \end{array} \tag{4.6}$$

if we assume that $\theta_3 = \theta_4 = 0$. Unfortunately these equations are probably impossible to solve analytically and we must use a numerical method.

4.2 Numerical Solution

There exist different numerical methods to solve the inverse kinematic problem. A common part for all methods is that they use some kind of velocity relationship, a *Jacobian*. We will do a more extensive analysis of the Jacobian in Chapter 6, but we already now need a result called *analytic Jacobian* in our inverse kinematics algorithm.

4.2.1 Analytic Jacobian *

In our algorithms below we need a relationship between joint velocities and the end-effector linear and angular velocity. The translational velocity of the end-effector frame can be expressed as

$$\dot{\boldsymbol{p}} = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = J_P(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.7}$$

and rotational velocity as

$$\dot{\boldsymbol{\phi}} = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = J_\phi(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.8}$$

The matrix

$$J_A(\boldsymbol{q}) = \begin{bmatrix} J_P(\boldsymbol{q}) \\ J_\phi(\boldsymbol{q}) \end{bmatrix} \tag{4.9}$$

is called the *analytic Jacobian*. Thus the differential end-effector location (3.13), expressed with reference to a minimal representation in the operational space can be expressed as

$$\dot{\boldsymbol{x}} = J_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.10}$$

where

$$J_A(\boldsymbol{q}) = \frac{\partial \boldsymbol{k}(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{4.11}$$

4.2.2 Inverse Kinematics Algorithm *

Define the operational space error by the desired and the actual end-effector position and orientation as

$$\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x} \tag{4.12}$$

Consider the time derivative of (4.12) and use (4.10)

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}}_d - J_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.13}$$

We now search a relationship between $\dot{\boldsymbol{q}}$ and $\boldsymbol{e}$ so that (4.13) can describe the error over time. Of course we must choose a relationship that ensures convergence of the error to zero. There exist several methods, but we will use a method that is based on the inverse of the analytical Jacobian.

4.2.3 Jacobian Inverse Method *

Assume that $J_A(\boldsymbol{q})$ is square and nonsingular. Choose the relationship between $\dot{\boldsymbol{q}}$ and $\boldsymbol{e}$ as

$$\dot{\boldsymbol{q}} = J_A^{-1}(\boldsymbol{q})(\dot{\boldsymbol{x}}_d + K\boldsymbol{e}) \tag{4.14}$$

Then (4.13) and (4.14) leads to

$$\dot{\boldsymbol{e}} = -K\boldsymbol{e} \quad \Longleftrightarrow \quad \dot{\boldsymbol{e}} + K\boldsymbol{e} = \boldsymbol{0} \tag{4.15}$$

The system (4.15) is asymptotically stable if K is a positive definite matrix. The speed of the convergence depends on K, the larger eigenvalues, the faster convergence. However, for a discrete time implementation there will be a limit for the maximum eigenvalues under which asymptotical stability is guaranteed. This upper bound depends on the rate of sampling.

4.3 Implementation and Simulation

Implementation of a discrete time version of the Inverse Jacobian Method is quite straight forward. However, there are a number of technical problems that have to be addressed. Firstly, the method requires that the analytical Jacobian is square and nonsingular, which it is not. Secondly, the Jacobian inverse method requires that the iteration starts from a point not too far from the desired solution. Thirdly, the gimbal tilt joint is constrained and we must guarantee that the solution is in the feasible range. Fourthly, in Section 2.3.2 we noticed the difficulties connected with following a path near the singularity. How do we move the gimbal in order to minimize the risk of losing the target? Furthermore, all these problems are not isolated from each other. The first and second problem will be addressed in Section 4.3.1 and 4.3.2 in this chapter. However, the complete solution will be presented in Chapter 5.

4.3.1 Kinematic Singularity

Since we are only interested in the position of the end effector we only consider the position part J_{A_P} of the analytical Jacobian. Furthermore, we set $\theta_3 = \theta_4 = 0$ and search for a solution $\boldsymbol{q} = [\theta_1 \quad \theta_2 \quad d_{10}]$. This result in a 3×3 matrix. Thus the matrix is square, but unfortunately singular when $\theta_2 = \frac{\pi}{2}$ (or $\theta_2 = -\frac{\pi}{2}$).

The Jacobian inverse $J_{A_P}^{-1}$ can be computed only when J_{A_P} has full rank, the system $\dot{\boldsymbol{x}} = J_{A_P}\dot{\boldsymbol{q}}$ contains linearly dependent equations. It is only possible to find a solution $\dot{\boldsymbol{q}}$ if $\dot{\boldsymbol{x}} \in \Re(J)$. [2] If instead $\dot{\boldsymbol{x}} \notin \Re(J)$ the system has no solution and the operational space path cannot be executed. However, even if a solution exists the determinant is small in the neighborhood of a singularity, which will cause large joint velocities. For instance, if the gimbal is faced down (θ_2 near π) very large velocities on pan may be required to make small adjustments in operational space. Recall the discussion in Section 2.3.2.

One solution to the problem with inverting the Jacobian in the neighborhood of a singularity is provided by the *damped least squares inverse* (DLS-inverse)

$$J^* = J^T(JJ^T + k^2 I)^{-1} \tag{4.16}$$

where k is a damping factor.

4.3.2 Finding a Start Point

Considering a spherical system, Figure 4.1, described by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos(\varphi)\sin(\theta) \\ r\sin(\varphi)\sin(\theta) \\ r\cos(\theta) \end{bmatrix} \tag{4.17}$$

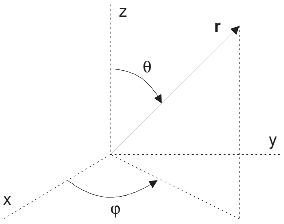


Figure 4.1. Spherical system.

A natural start point can now be calculated by

$$\boldsymbol{q}_{start} = \begin{bmatrix} \varphi \\ \theta \\ r \end{bmatrix} = \begin{bmatrix} arctan2(y,x) + 2\pi n \\ arctan2(\sqrt{x^2+y^2}, z) \\ \sqrt{x^2+y^2} \end{bmatrix} \tag{4.18}$$

where the term $2\pi n$ will guarantee that φ is continuous, n is the number of revolutions.

Once the algorithm has completed its first iteration(s) at the first sample we have two choices of start points for the remaining samples. We can either calculate

[2] $\Re(J)$ is the *range space* of transformation J, the subspace generated by the linearly independent columns of J.

the start point as for the first sample or start from the last point from the previous sample. The last choice requires fewer iterations to compute a solution, but, as we will se in Chapter 5, this solution is sometimes not the one we want.

4.3.3 Simulation Results

The inverse kinematic algorithm was successfully implemented. This pseudo-code shows the instructions that are executed every sample.

```
iter := 0;
q := qstart;
while iter < itermax
    e := xd - k(q);
    qd := inv(JAP(q)) * (xdotd + K * e);
    q  := ts * qd + q;
    iter := iter + 1;
end
```

ts is the sample time and the choice of *qstart* was discussed in 4.3.2. However, the simulation result is better if we set $\dot{\boldsymbol{x}}_d \equiv 0$. The approximately magnitude of error (far from the singularity) is given in Table 4.1.

Iterations	Error 1	Error 2
1	10^{-2}	10^{-6}
2	10^{-5}	10^{-12}
3	10^{-10}	-

Table 4.1. The approximate magnitude of the error in the inverse kinematic algorithm. *Error 1* is when the startpoint is given by (4.18) and *Error 2* is when the startpoint is given by $\boldsymbol{q}$ in the previous sample.

The damped least squares inverse was also implemented and tested. We let the value of damping factor k be a function that depends on a condition number of the Jacobian. Far from the singularity the condition number is low and thus k small and the *DLS*-inverse is approximately the Jacobian inverse. Near the singularity the condition number increase and thus also the damping factor. The simulation result was satisfying.

References & Further Readings

Section 4.2 and 4.3.1 are summaries of corresponding sections in [17]. Besides the Jacobian Inverse Method [17] also suggest another method *Jacobian Transpose* that is not tested here.

Chapter 5

Gimbal Trajectory Planning

The purpose of the gimbal trajectory planner is to transform different reference signals from applications to a camera assembly orientation reference that can be used as the input to the gimbal controller. An important property in this context is that the orientation reference signal is *executable*, which means that the reference signal is realistic and is within the allowed range. We will use the kinematic model in Chapter 3 and the inverse kinematic algorithm from Chapter 4.

5.1 Overview

The trajectory planner consists of three major parts plus two filters, Figure 5.1. First the reference signal from the applications is transformed to local coordinates. With local coordinates we mean coordinates in a vehicle fixed reference frame. After the transformation, the local coordinates are analysed and possibly modified in order to cope with constraints and the singularity. Before applying the inverse kinematic algorithm developed in Chapter 4, it is necessary to filtrate the signal since the algorithm requires a continuous reference signal. An exception to this rule is point-to-point motion where the signal is not filtered. In the inverse kinematic solver the reference is transformed from local coordinates to a desired orientation of the camera assembly. Finally the orientation reference is filtered before being used by the gimbal motion controller, which will be described in Chapter 9.

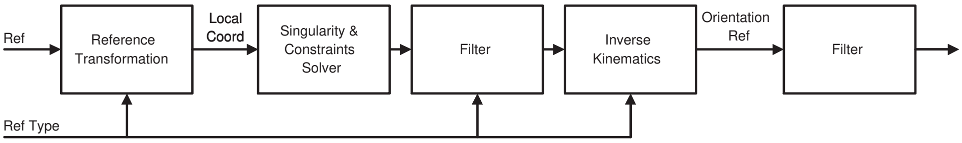


Figure 5.1. Trajectory Planner.

5.1.1 Orientation Reference

The desired orientation of the body that the cameras are attached to is defined by two angles γ_1 and γ_2 in an ordinary spherical system, Figure 5.2. [1]

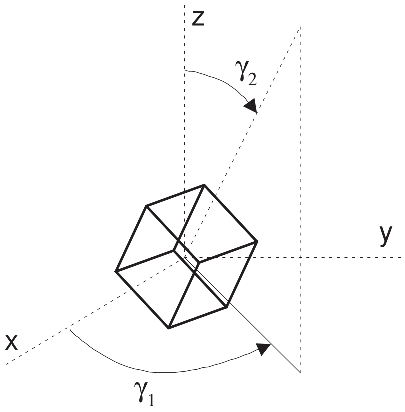


Figure 5.2. Orientation reference definitions.

It is well known that an arbitrary orientation can not be described by a representation of fewer than three parameters, e.g. Euler angles or RPY angles. However, in our case two angles are enough and suitable because of the structure of the gimbal with a pan and a tilt controlled joint.

5.2 Reference Signal Transformations

In this section we derive the transformations that transform the different reference signals from the applications to a local coordinate reference. An overview of the reference signals was presented in Section 2.3.1.

5.2.1 Reference Frames

In this chapter we will use the kinematic description of D-H Approach 3 in Section 3.5, denoted as DH3. As in Chapter 3 we denote T_i^j as the homogenous transformation that describe the position and orientation of frame j with respect to frame i. We define five different reference frames, Table 5.1, and then some homogenous transformation matrices, 5.2.

[1] Note that the orientation is the desired point direction plus a constant orientation bias depending on θ_8 and θ_9 in DH3.

Symbol	Reference frame
G	Global earth fixed reference frame
V	Vehicle fixed (Local) reference frame, the base frame in DH3
C	Camera (IR) fixed reference frame
A	Target fixed reference frame
E	End-effector frame in DH3

Table 5.1. Definitions of different reference frames.

Symbol	Homogenous transformation
T_V^C	Camera relative vehicle, the homogenous transformation in DH3 except the last joint/link (10).
T_V^E	End-effector relative vehicle, the homogenous transformation in DH3.
T_V^A	Target relative vehicle
T_G^A	Target relative a global frame, reference in some scan applications.
T_G^C	Camera relative a global frame, estimated by the navigation system.

Table 5.2. Definitions of different homogenous transformations.

5.2.2 Local coordinates

In Section 5.1 we stated that local coordinates are coordinates relative a vehicle fixed reference frame, i.e. frame V. In this section we will express the target position relative to this frame V. We will see that a useful way of finding this position is to derive T_V^A since the position part of T_V^A is the desired coordinate. From Section 3.1 we know that the position part of a homogenous transformation matrix is the first three elements of the fourth column. Hence the position part $\boldsymbol{p}_V^A$ of T_V^A can be computed as

$$\boldsymbol{p}_V^A = F T_V^A \boldsymbol{v}_p \tag{5.1}$$

where

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.2}$$

and

$$\boldsymbol{v}_p = [0 \quad 0 \quad 0 \quad 1]^T \tag{5.3}$$

5.2.3 Scan Applications

In scan applications the reference signal consists of coordinates relative the global reference frame (G) or coordinates relative the vehicle reference frame (V), denoted as $\boldsymbol{p}_G$ and $\boldsymbol{p}_L$ respectively. Point-to-point and path motion receives an identical treatment.

First consider the case when the reference signal is relative a local reference frame, i.e. frame V. This case is trivial since the reference already is given in the desired coordinates. Thus

$$\boldsymbol{p}_V^A = \boldsymbol{p}_L \tag{5.4}$$

Now consider the next case when the reference is relative a global reference frame, i.e. frame G. Form T_G^A by insert the reference coordinates $\boldsymbol{p}_G$ as the position part, the orientation part is insignificant. Now we can compute T_V^A as

$$T_V^A = T_V^C T_C^G T_G^A \tag{5.5}$$

where $T_C^G = (T_G^C)^{-1}$. Thus

$$\boldsymbol{p}_V^A = F T_V^C (T_G^C)^{-1} \tilde{\boldsymbol{p}}_G \tag{5.6}$$

where $\tilde{\boldsymbol{p}}_G$ is the homogenous representation of $\boldsymbol{p}_G$ according to (3.2).

Notice that in simulations T_G^V is given instead of T_G^C and then T_G^C is

$$T_G^C = T_G^V T_V^C \tag{5.7}$$

Of course it is not required that we explicitly calculate T_G^C in simulations, but we use (5.7) as the output from the simulated navigation system.

5.2.4 Tracker Applications

The camera has a field-of-view of $\alpha_x \times \alpha_y$ and the detector array is composed of $\rho_x \times \rho_y$ pixels. Assume that the error from the tracker is given as a two elements vector $\boldsymbol{e}_{pixel} = [e_x \quad e_y]^T$ describing the pixel error relative the center of the images.

Since the output from the tracker is an error it is tempting to control the gimbal directly by converting the pixel error to an angle error and consider the x-angle error as the pan angle error and y-angle error as the tilt angle error. However, this simple method will only work if the distance to the target is long and the tilt angle is in a neighbourhood of 0. To handle the singularity we first have to transform the error to coordinates relative the local reference frame. Below we will propose two methods.

In Method 1 we assume that the target moves perpendicular to the point direction. Then we can transform the pixel error to coordinates relative the end-effector frame. Consider Figure 5.3. The true lengths L_x and L_y in the real world, corresponding to the width and height of the image, can be expressed as

$$L_i = 2d_{10}\tan(\alpha_i), \quad i = x, y \tag{5.8}$$

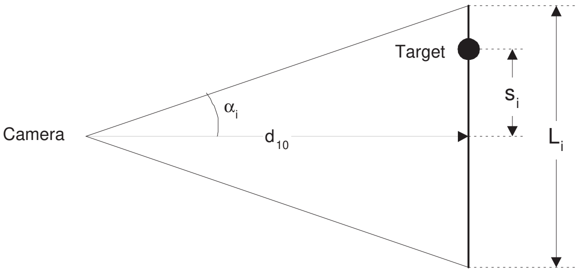


Figure 5.3. Camera view. The target is assumed to move perpendicular to the point direction.

Now the position $[s_x \quad s_y \quad 0]$ of the target relative the position of the center of image easily can be derived from these ratios

$$\frac{s_i}{L_i} = \frac{e_i}{\rho_i}, \quad i = x, y \tag{5.9}$$

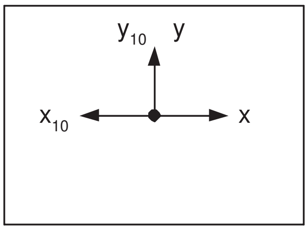


Figure 5.4. The orientation of the end-effector axes x_{10} and y_{10} in the image.

Note that the end-effector frame $o_{10}x_{10}y_{10}z_{10}$ is oriented in a different way, Figure 5.4. Thus the position of the target relative the end-effector frame is

$$\boldsymbol{p}_E^A = \begin{bmatrix} -s_x \\ s_y \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{e_x}{\rho_x} d_{10} \tan(\alpha_x) \\ \frac{e_y}{\rho_y} d_{10} \tan(\alpha_y) \\ 0 \end{bmatrix} \tag{5.10}$$

Finally we can express T_V^A as

$$T_V^A = T_V^E T_E^A \tag{5.11}$$

where the position part of T_E^A is $\boldsymbol{p}_E^A$. Thus

$$\boldsymbol{p}_V^A = F T_V^E \tilde{\boldsymbol{p}}_E^A \tag{5.12}$$

where $\tilde{\boldsymbol{p}}_E^A$ is the homogenous representation of $\boldsymbol{p}_E^A$ according to (3.2).

Note that this method depends on d_{10} and we must be careful when d_{10} is very small or very large.

An alternative method (Method 2) that does not depend on the distance d_{10} is also presented. This methods ignores the camera placement offsets and the damping flexibility.

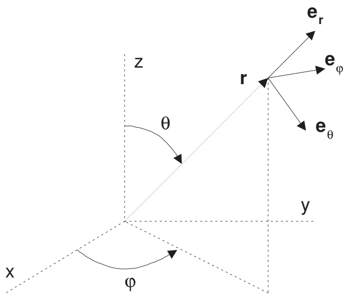


Figure 5.5. Spherical system.

Consider the spherical system in Figure 5.5. Assume that the motor controlled angles θ_1 and θ_2 describes the point direction, thus $\varphi = \theta_1$ and $\theta = \theta_2$. Let $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{\theta}}$ describe a orthogonal frame defined as

$$\begin{array}{rclcl} \hat{\boldsymbol{r}} & = & \boldsymbol{e}_r & = & [\sin\theta\cos\varphi \quad \sin\theta\sin\varphi \quad \cos\theta]^T \\ \hat{\boldsymbol{\theta}} & = & \boldsymbol{e}_\theta & = & [\cos\theta\cos\varphi \quad \cos\theta\sin\varphi \quad -\sin\theta]^T \\ \hat{\boldsymbol{\varphi}} & = & \boldsymbol{e}_\phi & = & [-\sin\varphi \quad \cos\varphi \quad 0]^T \end{array} \tag{5.13}$$

Note that $\hat{\boldsymbol{\varphi}}$ and $\hat{\boldsymbol{\theta}}$ are undefined on the z-axis.

Suppose that $\hat{\boldsymbol{r}}$ is the center of the image, we assume that the distance to the target is 1. The directions of x and y in the image correspond to $-\hat{\boldsymbol{\varphi}}$ and $-\hat{\boldsymbol{\theta}}$ respectively, according to Figure 5.4. If the angles of the image view are small we can approximately express the position of the target in local coordinates as

$$\boldsymbol{p}_V^A = \hat{\boldsymbol{r}} - s_\varphi\hat{\boldsymbol{\varphi}} - s_\theta\hat{\boldsymbol{\theta}} \tag{5.14}$$

where s_φ and s_θ are calculated from (5.8) and (5.9) but $d_{10} = 1$

$$s_\varphi = s_x\big|_{d_{10}=1} \tag{5.15}$$

$$s_\theta = s_y\big|_{d_{10}=1} \tag{5.16}$$

Since we in this method ignore the camera offsets we should not use the inverse kinematic algorithm in Chapter 4, instead we calculate the orientation reference directly as

$$\begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} = \begin{bmatrix} arctan2(y, x) + 2\pi n \\ arctan2(\sqrt{x^2 + y^2}, z) \end{bmatrix} \tag{5.17}$$

where x, y, z are given by the elements in $\boldsymbol{p}_V^A$ in (5.14).

5.3 Inverse Kinematics

The inverse kinematic problem was presented and analyzed in Chapter 4. Since we set $\theta_3 = \theta_4 = 0$ in the algorithm and represent the orientation with the angles γ_1 and γ_2 the result θ_1 and θ_2 from the algorithm is in fact the desired orientation γ_1 and γ_2. Thus inverse kinematic algorithm finds the orientation of the gimbal that makes the camera point at the desired position. The distance (d_{10}) from the camera to the target is also given by the algorithm. However, some problems were introduced and intentionally left unanswered in Section 4.3: A) Selecting a start point, B) Guarantee that result is in the allowed range.

A particular choice of starting point is the result in the previous sample. Usually this choice is good since it is very close to the solution and few (about 1-3) iterations are needed, but it is a bad choice when the reference is discontinuous. Filtering of the reference will solve this problem and since the algorithm use the differential of the reference the filtering also makes the algorithm more robust. This start point is also bad near the singularity since the algorithm then may intersect the tilt constraint. A solution to this problem is to select the start point as in the first sample, see Section 4.3.2, in a neighbourhood of the singularity.

We have now completed the inverse kinematic solution. If we use the DLS-inverse and the proposed strategies above, the singularity and constraints problem are also handled. However, the reference following and the tracking near the singularity are not satisfactory. If the reference goes straight through the singularity the gimbal will move close to the singularity by rotating the tilt angle, then stop rotating the tilt angle, rotate the pan angle 180° and then rotate the tilt angle again. It may take the pan angle about 3 seconds to rotate 180° and the camera runs the risk of losing the target in the image view. It is obvious that there must be better methods since we are working with camera views and it is not necessary for the target to be at the center of the image. The next section will discuss this problem.

5.4 Path Planning Near Singularity

For the sake of simplicity, we assume that the *image view* is circular so that we can ignore its orientation. Since the camera is not placed in the rotational axis we have a circular area, denoted as the *singularity area*, around the z-axis where it is impossible for the *center* of the image view to reach. The size of this area is constant, but the size of the image view depends on the camera zoom and the distance to the target. If we assume that the size of the camera view is larger than the singularity area there always exist a point direction such that the target is in the image view.

It is obvious that we can track faster targets that moves near the singularity if we do not require that the target must be centered in the image view. The problem is to find a path planning system that minimizes the risk of losing the target in the image view. The problem is complex since we do not know the trajectory of the target in advance. It is obvious that we must predict the target motion in some way. We notice that the singularity area affects the problem more when we track targets at a short distance.

We will present two methods for coping with this problems. These methods are generally described as two different mapping functions rather than predictors. However, these algorithms are not intended to be complete and good solutions to the problems, instead the aim is to illustrate some of the difficulties in the path planning problem.

5.4.1 Method 1. Radial mapping on circle.

In this algorithm we define a circle with radius R_1, larger than the singularity area. If the reference (target) is inside this circle we do a radial mapping of the reference on the circle border, Figure 5.6. Possibly we can also introduce a maximum pan angle change limit. Positive qualities of Method 1 are that the method is robust and very simple. However, the result is not very impressive, since the loss of advance planning and the the result reference is rather angular. The result is similar to the result when we just use the DLS-inverse in the inverse kinematic algorithm as described in Section 5.3.

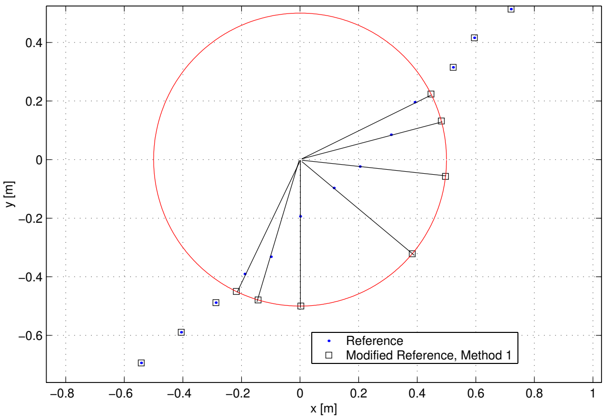


Figure 5.6. Method 1. Radial mapping on circle.

5.4.2 Method 2. Orthogonal mapping on circle tangent.

In this algorithm two circles are defined. When the reference is inside the outer circle the reference is mapped orthogonal, relative a radial vector, on one of the tangents of the inner circle (radius R_2), that intersects the point where the outer circle and the radial vector intersects, Figure 5.7. As the target enters the outer circle, the algorithm predicts which "side" of the singularity the target will move. The tangent that correspond to this side is chosen and this tangent is then used

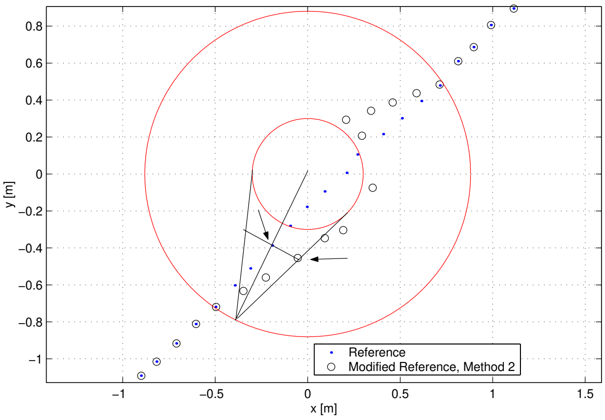


Figure 5.7. Method 2. Orthogonal mapping on circle tangent.

until the target exits the outer circle, to avoid a discontinuous reference. If the reference is inside the inner circle the point is mapped as in Method 1 before the orthogonal mapping is performed.

It is positive that the pan angle starts rotating rather early, but the risk is that the rotating is performed in the wrong direction. Both methods gives different results depending on the direction of the rotating, since the mechanical structure (camera offset) is not symmetric.

5.4.3 Simulations

We tested the methods with a simple reference, a line on a sphere with radius r. We only consider the x and y parts since $r >> |x|$, $r >> |y|$. Thus the reference is a straight line relative x and y that goes past the singularity. The reference is first modified by the two algorithms respectively and these new references are then used in simulations. The results that are presented are the x, y coordinates and the joint angles from the path planning methods compared to the simulated output signals. Also the distance error between the image view center and the target is presented. In simulation 1, Figure 5.8, and simulation 2, Figure 5.9, the circle in Method 1 and the inner circle in Method 2 are both 0.2 [m]. The difference between the simulations is that the target moves 3 times faster in simulation 2 than in simulation 1. In Appendix C two more simulations are presented with the same conditions except that the circles are larger, the radius is 0.5 [m].

Consider the XY-plot in Figure 5.8 (top). The result from Methods 2 ”leaves” the reference earlier than in Method 1. However, Method 2 causes a large ”overshoot” on the other side of the singularity. Thus one may think that Method 1 is better, but we see in the pan plot, Figure 5.8 (middle), and the error plot, Figure 5.8 (bottom), that Method 2 in fact provides the best result. In the error plot the maximum value is of most interest, since if the maximum value is smaller than the image view radius (not defined here) the target was in the image view all the time[2]. In both simulations Method 2 gives the best result. The reason for this is that the pan joint starts rotating earlier in Method 2 than in Method 1. However, there is a risk of start rotating early since the target may swing to the other side. We should remember that a straight reference as in these examples is rather unrealistic.

The methods have some parameters that can be changed, e.g. R_1 and R_2. In simulations 3 and 4, Figures C.1 and C.2 in Appendix C, R_1 and R_2 are larger than in simulations 1 and 2. We see that in the case where the target has less velocity, small values of R_1 and R_2 are preferable. However, when the target is faster it seems to be better with larger R_1 and R_2.

As mentioned before, the purpose of this section was not to present a complete solution. This planning problem must be analyzed further, but hopefully this section has shown the complexity of the problem.

5.5 Orientation Filter

The aim of the orientation filter is to filter γ_1 and γ_2 so that the motion controller problem becomes easier. For instance, it is advantageous to reduce frequencies that we know excite the damping flexibility.

5.6 Point-To-Point

In Section 2.3.1 the difference between "path motion" and "point-to-point" was described. Point-to-point motion is a special case there the path between the points does not matter, the most important is that the elapsed time when going from the first point to the other is as short as possible. If we ignore the difference and consider the two cases as identically, sometimes the point-to-point motion still will perform good, e.g. when the points are close to each other. Now consider a move from the point $[x \quad y \quad z] = [10 \quad 0 \quad -10]$ on one side of the gimbal to $[-10 \quad 0 \quad -10]$ on the other side. Ignoring the difference will cause the gimbal to try to go the shortest way in the cartesian frame, i.e. through the singularity. Intuitively we know that the best and fastest way is to rotate the pan axis 180°.

One simple solution is to *not* filter the reference before the inverse kinematic block and in the inverse kinematic algorithm use the spherical angles as the start point instead of the result in previous sample. The result from the inverse kinematic block will then be a reference from $[\gamma_1 \quad \gamma_2] \approx [0^\circ \quad 45^\circ]$ to $[180^\circ \quad 45^\circ]$. Thus a rotation of pan axis 180° as we wished!

[2] The observant reader may have discovered that the error never goes down to zero, also far away from the singularity. The reason is that we used a regulator in the simulations that could not bring the error to zero.

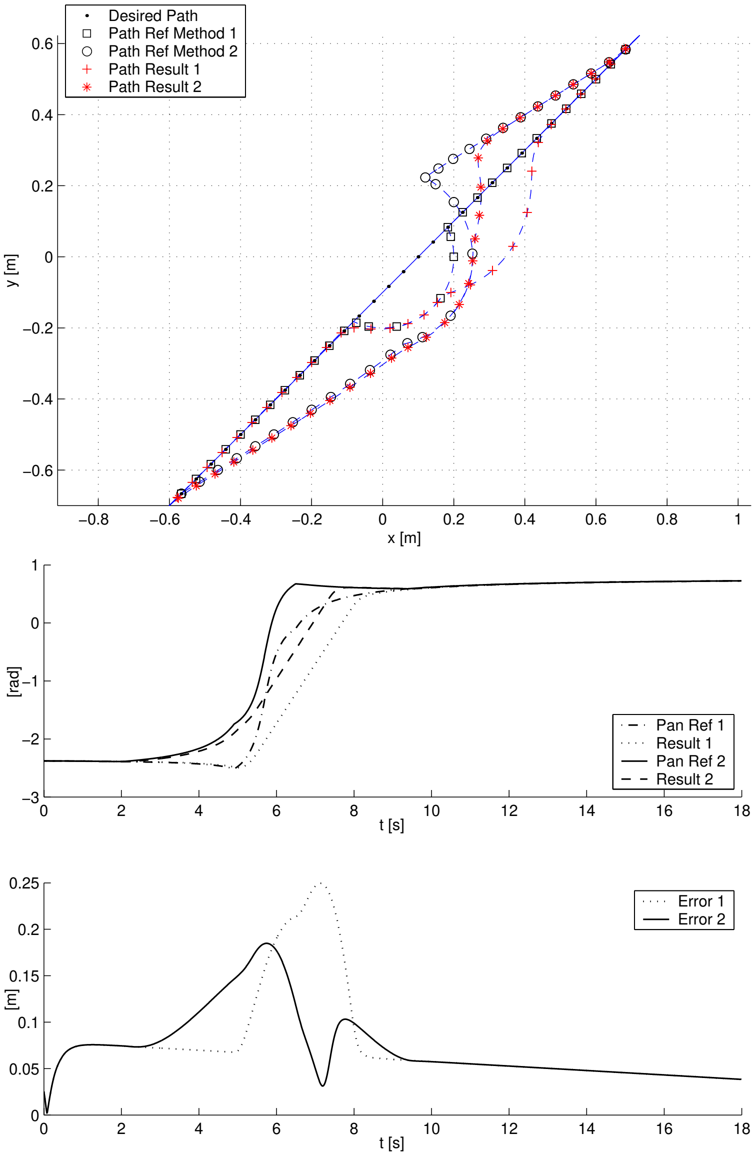


Figure 5.8. Simulation 1. The numbers "1" and "2" are abbreviations for Method 1 and Method 2 respectively. Target velocity ≈ 0.25 [m/s]. $R_1 = R_2 = 0.2$. *Top:* XY references and output results. *Middle:* Pan angle references and output results. *Bottom:* XY error between image view center and target.

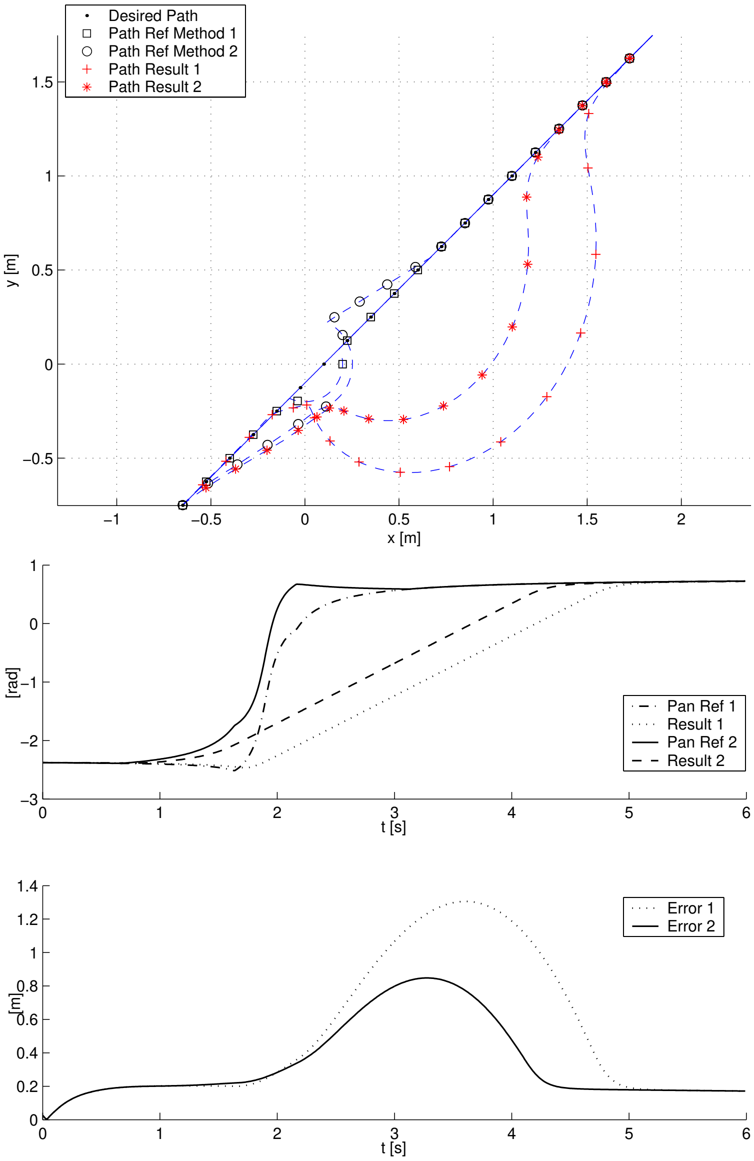


Figure 5.9. Simulation 2. The numbers "1" and "2" are abbreviations for Method 1 and Method 2 respectively. Target velocity ≈ 0.75 [m/s]. $R_1 = R_2 = 0.2$. *Top:* XY references and output results. *Middle:* Pan angle references and output results. *Bottom:* XY error between image view center and target.

Chapter 6

Differential Kinematics

The goal of differential kinematics is to find the relationship between the joint velocities and the end-effector linear and angular velocities. This relationship will be used in the computation of the dynamics of the gimbal systems in the next chapter.

6.1 Geometric Jacobian

We will express the linear velocity $\dot{\boldsymbol{p}}$ and the angular velocity $\boldsymbol{\omega}$ of the end-effector as a function of the joint velocities $\dot{\boldsymbol{q}}$.

$$\dot{\boldsymbol{p}} = J_v(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{6.1}$$

$$\boldsymbol{\omega} = J_\omega(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{6.2}$$

By defining $\boldsymbol{v} = [\dot{\boldsymbol{p}} \quad \boldsymbol{\omega}]^T$ we can write (6.1) and (6.2) in a compact form

$$\boldsymbol{v} = J(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{6.3}$$

where matrix J is the *geometric Jacobian* (or Jacobian for short)

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \tag{6.4}$$

Compare this definition with the analytical Jacobian in Section 4.2.1.

6.1.1 Angular Velocity *

Consider the angular velocity ω_{i-1}^i of link i in the link $i-1$ frame. If joint i is prismatic then

$$\boldsymbol{\omega}_{i-1}^i = 0; \tag{6.5}$$

and if joint i is revolute then

$$\boldsymbol{\omega}_{i-1}^i = \dot{q}_i \boldsymbol{r}_z \tag{6.6}$$

where $\boldsymbol{r}_z = [0 \quad 0 \quad 1]^T$ is the rotational axis. We can express (6.6) in the base frame as

$$\boldsymbol{\omega}_{i-1,0}^{i} = \dot{q}_i R_0^{i-1} \boldsymbol{r}_z \tag{6.7}$$

Considering that angular velocities can be added if they are expressed in the same coordinate frame [18]. Thus we can express the overall angular velocity as

$$\boldsymbol{\omega} \equiv \boldsymbol{\omega}_0^n = \sum_{i=1}^{n} \rho_i \boldsymbol{z}_{i-1} \dot{q}_i \tag{6.8}$$

where

$$\boldsymbol{z}_i = R_0^i \boldsymbol{r}_z \tag{6.9}$$

and $\rho_i = 0$ if joint i prismatic and $\rho_i = 1$ if joint i revolute. J_ω is thus given as

$$J_\omega = [\rho_0 \boldsymbol{z}_0 \quad ... \quad \rho_n \boldsymbol{z}_{n-1}] \tag{6.10}$$

6.1.2 Linear Velocity *

Let $\dot{\boldsymbol{d}}_0^n$ denote the linear velocity of the end-effector. Then the chain rule gives

$$\dot{\boldsymbol{d}}_0^n = \sum_{i=1}^{n} \frac{\partial \boldsymbol{d}_0^n}{\partial q_i} \dot{q}_i \tag{6.11}$$

and we can see that i-th column of J_v is

$$J_{v_i} = \frac{\partial \boldsymbol{d}_0^n}{\partial q_i} \tag{6.12}$$

Thus we can get J_{v_i} from (6.11) if $\dot{q}_i$ equals unity and all other joint velocities are equal to zero. If joint i is prismatic we have

$$\boldsymbol{d}_{i-1}^i = d_i \boldsymbol{r}_z + R_{i-1}^i a_i \boldsymbol{r}_x \tag{6.13}$$

where $\boldsymbol{r}_x = [1 \quad 0 \quad 0]^T$. Fix all joints but the i-th and differentiate d_0^n

$$\dot{\boldsymbol{d}}_0^n = R_0^{i-1} \dot{\boldsymbol{d}}_i^{i-1} = \dot{d}_i \boldsymbol{z}_{i-1} \tag{6.14}$$

Thus

$$J_{v_i} = \boldsymbol{z}_{i-1} \tag{6.15}$$

for a prismatic joint. If joint i is revolute we have

$$\boldsymbol{d}_0^n = \boldsymbol{d}_0^{i-1} + R_{i-1}^0 \boldsymbol{d}_{i-1}^n \tag{6.16}$$

and

$$\dot{\boldsymbol{d}}_0^n = R_{i-1}^0 \dot{\boldsymbol{d}}_{i-1}^n \tag{6.17}$$

if all joints are fix but joint i. Motion of link i can be expressed as

$$\dot{\boldsymbol{d}}_{i-1}^{n} = \dot{q}\boldsymbol{z}_0 \times \boldsymbol{d}_{i-1}^{n} \tag{6.18}$$

and thus

$$\dot{\boldsymbol{d}}_{0}^{n} = \dot{q}\boldsymbol{z}_{i-1} \times (\boldsymbol{d}_0^n - \boldsymbol{d}_0^{i-1}) \tag{6.19}$$

We now see that the i-th column of J_ω can be expressed as

$$J_{\omega_i} = \boldsymbol{z}_{i-1} \times (\boldsymbol{d}_0^n - \boldsymbol{d}_0^{i-1}) \tag{6.20}$$

6.1.3 n-link Manipulator *

From the result above we can now express the geometric Jacobian for an n-link manipulator as

$$J = [J_1 \quad ... \quad J_n] \tag{6.21}$$

where

$$J_i = \begin{bmatrix} \boldsymbol{z}_{i-1} \times (\boldsymbol{d}_0^n - \boldsymbol{d}_0^{i-1}) \\ \boldsymbol{z}_{i-1} \end{bmatrix} \tag{6.22}$$

if joint i is revolute and

$$J_i = \begin{bmatrix} \boldsymbol{z}_{i-1} \\ 0 \end{bmatrix} \tag{6.23}$$

if joint i is prismatic.

Note that all variables in (6.22) and (6.23) can be found in the homogenous transformations matrix.

- $\boldsymbol{z}_{i-1}$ is given by the first three elements of the third column of T_0^{i-1}.
- $\boldsymbol{d}_0^n$ is given by the first three elements of the fourth column of T_0^n.
- $\boldsymbol{d}_0^{i-1}$ is given by the first three elements of the fourth column of T_0^{i-1}.

6.2 Geometric Jacobian for the Gimbal System

We will need the geometric Jacobian in Chapter 7 and therefore we now calculate Jacobians based on the Denavit-Hartenberg parameters in *D-H Approach* 4 in Section 3.5. Following the items above, it is now straight forward to calculate the rotational part of the Jacobians as

$$J_{\omega_1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.24}$$

$$J_{\omega_2} = \begin{bmatrix} 0 & -\sin(\theta_1) & 0 & 0 \\ 0 & \cos(\theta 1) & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.25}$$

$$J_{\omega_3} = \begin{bmatrix} 0 & -\sin(\theta_1) & \cos(\theta_1)\sin(\theta_2) & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1)\sin(\theta_2) & 0 \\ 1 & 0 & \cos(\theta_2) & 0 \end{bmatrix} \tag{6.26}$$

$$J_{\omega_4} = \begin{bmatrix} 0 & -\sin(\theta_1) & \cos(\theta_1)\sin(\theta_2) & -\theta_3\cos(\theta_1)\cos(\theta_2) - \sin(\theta_1) \\ 0 & \cos(\theta 1) & -\sin(\theta_1)\sin(\theta_2) & \cos(\theta_1) - \theta_3\cos(\theta_2)\sin(\theta_1) \\ 1 & 0 & \cos(\theta_2) & \theta_3\sin(\theta_2) \end{bmatrix} \tag{6.27}$$

In this special case, D-H Approach 4, we have placed the origin of each frame at the center of mass for each link. In Chapter 7 we assume that the center of mass for each link is located in one point, the intersection of all rotational axes. This means that the frames do not have any translational movements and therefor

$$J_{v_i} = \mathbf{0} \quad i = 1, 2, 3, 4 \tag{6.28}$$

References & Further Readings

Section 6.1 is a summary of corresponding sections in [18] and [15].

Chapter 7

Dynamics

A *dynamic model* of a manipulator is a description of the relationship between the joint actuator torques and the motion of the structure. The dynamic model is useful for simulation of motion, analysis of structure and design of control strategies.

In this chapter we will use a method based on the *Lagrange formulation* for derivation of the equations of motion for a general robot manipulator. Then we will apply this to the gimbal system.

7.1 Lagrange Equations

By the Lagrange formulation a set of differential equations that describe the motions of a mechanical system can be derived. The procedure is as follows. A set of *generalized coordinates* are chosen which effectively describes the link positions in the mechanical system. Then the *Lagrangian* can be defined as a function of the generalized coordinates

$$L = K - V \tag{7.1}$$

where K and V are the kinetic and potential energy respectively. By using the Lagrange equations the equations of motion can now be derived in a systematic way. The Lagrange equations are expressed as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\lambda}_i} - \frac{\partial L}{\partial \lambda_i} = F_i \qquad i = 1, 2, ..., n \tag{7.2}$$

where F_i is the *generalized force* associated with the generalized coordinate λ_i.

For a robot manipulator an obvious choice of generalized coordinates is the joint variables, i.e. $\lambda_i = d_i$ for prismatic joints and $\lambda_i = \theta_i$ for revolute joints. The generalized forces are given by the nonconservative forces, i.e. the joint actuator torques and joint friction torques.

Lagrange equations are used as a tool in analytical mechanics and can be derived in different ways, but the derivations are out of scope of this thesis. A comprehensive derivation can be found in [18] and an exhaustive analysis can be found in [6].

7.2 Equations of Motion of a Robot Manipulator

In order to derive the equations of motion from Lagrange's equations we need to compute the Lagrangian of a robotic manipulator. Since the Lagrangian is the difference between total kinetic and potential energy we will derive expressions for kinetic and potential energy of a robot manipulator.

7.2.1 Computation of Kinetic Energy *

We want to find the kinetic energy for a rigid object. Assume that we attach a coordinate frame rigidly to the object, with its origin at the center of mass.

The definition of kinetic energy is

$$K = \frac{1}{2}\int_B \boldsymbol{v}^T(x,y,z)\boldsymbol{v}(x,y,z)\,dm \tag{7.3}$$

The velocity of a point on the object can be expressed as [18]

$$\boldsymbol{v} = \boldsymbol{v}_c + \boldsymbol{\omega} \times \boldsymbol{r} \tag{7.4}$$

Using result (A.5) we can express the velocity as

$$\boldsymbol{v} = \boldsymbol{v}_c + S(\boldsymbol{\omega})\boldsymbol{r} \tag{7.5}$$

where $S(\boldsymbol{\omega})$ is a skew symmetric matrix. Then the kinetic energy can be expanded into four terms K_i, $i = 1,2,3,4$

$$\begin{aligned} K &= \tfrac{1}{2}\textstyle\int_B [\boldsymbol{v}_c + S(\boldsymbol{\omega})\boldsymbol{r}]^T[\boldsymbol{v}_c + S(\boldsymbol{\omega})\boldsymbol{r}]\,dm = \\ &= \tfrac{1}{2}\textstyle\int_B \boldsymbol{v}_c^T\boldsymbol{v}_c\,dm + \tfrac{1}{2}\int_B \boldsymbol{v}_c^T S(\boldsymbol{\omega})\boldsymbol{r}\,dm + \\ &+ \tfrac{1}{2}\textstyle\int_B [S(\boldsymbol{\omega})\boldsymbol{r}]^T\boldsymbol{v}_c\,dm + \tfrac{1}{2}\int_B [S(\boldsymbol{\omega})\boldsymbol{r}]^T S(\boldsymbol{\omega})\boldsymbol{r}\,dm \end{aligned} \tag{7.6}$$

Since $\boldsymbol{v}_c$ is independent of the integration variable the first term is

$$K_1 = \frac{1}{2}\int_B \boldsymbol{v}_c^T\boldsymbol{v}_c\,dm = \frac{1}{2}m\boldsymbol{v}_c^T\boldsymbol{v}_c \tag{7.7}$$

This is just the kinetic energy of a particle of mass m with velocity $\boldsymbol{v}_c$ and is called the *translational* part of the kinetic energy.

The second and third term are both 0 because the center of mass is at the origin. Thus

$$K_2 = \frac{1}{2}\int_B \boldsymbol{v}_c^T S(\boldsymbol{\omega})\boldsymbol{r}\,dm = \frac{1}{2}\boldsymbol{v}_c^T S(\boldsymbol{\omega})\int_B \boldsymbol{r}\,dm = 0 \tag{7.8}$$

$$K_3 = \frac{1}{2}\int_B [S(\boldsymbol{\omega})\boldsymbol{r}]^T\boldsymbol{v}_c\,dm = \frac{1}{2}\int_B \boldsymbol{r}^T\,dm S^T(\boldsymbol{\omega})\boldsymbol{v}_c = 0 \tag{7.9}$$

Using identities (A.8) and (A.9) the last term can be expressed as

$$\begin{aligned} K_4 &= \tfrac{1}{2}\int_B [S(\boldsymbol{\omega})\boldsymbol{r}]^T S(\boldsymbol{\omega})\boldsymbol{r}\, dm = \tfrac{1}{2}\int_B tr[S(\boldsymbol{\omega})\boldsymbol{r}\boldsymbol{r}^T S^T(\boldsymbol{\omega})]\, dm = \\ &= \tfrac{1}{2}tr[S(\boldsymbol{\omega})\int_B \boldsymbol{r}\boldsymbol{r}^T\, dm S^T(\boldsymbol{\omega})] = \tfrac{1}{2}\boldsymbol{\omega}_T I\boldsymbol{\omega} \end{aligned} \tag{7.10}$$

where I is the *inertia tensor* defined as

$$I = \begin{bmatrix} \int y^2 + z^2\, dm & -\int xy\, dm & -\int xz\, dm \\ -\int xy\, dm & \int x^2 + z^2\, dm & -\int yz\, dm \\ -\int xz\, dm & -\int yz\, dm & \int x^2 + y^2\, dm \end{bmatrix} \tag{7.11}$$

This fourth term is known as the *rotational* part of the kinetic energy. Thus the kinetic energy for a rigid object is given by

$$K = \frac{1}{2}m\boldsymbol{v}_c^T\boldsymbol{v}_c + \frac{1}{2}\boldsymbol{\omega}_T I\boldsymbol{\omega} \tag{7.12}$$

Finally we examine if the result (7.12) depends on in which coordinate systems $\boldsymbol{v}_c$, $\boldsymbol{\omega}$ and I are calculated. $\boldsymbol{v}_c^T\boldsymbol{v}_c$ is the square of the length and the length is the same irrespective of frame. Hence, the first term is the same irrespective of the frame in which $\boldsymbol{v}_c$ is expressed. Also the second term is the same in all frames. However, it is desirable to express this term in the frame attached to the object, because the inertia matrix then will be constant. So if $\boldsymbol{\omega}_0$ is the angular velocity of the object measured in the inertial frame, then $R^T\boldsymbol{\omega}_0$ is the angular velocity of the object measured in the object frame.

Now consider an n-link robot manipulator. The joint variables are indeed the generalized coordinates and from Chapter 6 we got linear and angular velocity of any point on any link

$$\boldsymbol{v}_i = J_{v_i}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{7.13}$$

$$\boldsymbol{\omega}_i = R_i^T(\boldsymbol{q})J_{\omega_i}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{7.14}$$

Then we can express the kinetic energy for a n-link robot manipulator in terms of Jacobian matrix and generalized coordinates as

$$K = \frac{1}{2}\dot{\boldsymbol{q}}^T\sum_{i=1}^{n}[m_i J_{v_i}(\boldsymbol{q})^T J_{v_i}(\boldsymbol{q}) + J_{\omega_i}(\boldsymbol{q})^T R_i(\boldsymbol{q}) I_i R_i(\boldsymbol{q})^T J_{\omega_i}(\boldsymbol{q})]\dot{\boldsymbol{q}} \tag{7.15}$$

The kinetic energy is of the form

$$K = \frac{1}{2}\dot{\boldsymbol{q}}^T D(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{7.16}$$

where $D(\boldsymbol{q})$ is called *inertia matrix* and is a symmetric positive definite matrix.

7.2.2 Computation of Potential Energy *

Potential energy for a robot manipulator is given by a sum

$$V = \sum_{i=1}^{n} V_i \tag{7.17}$$

where V_i is the potential energy of link i. If all links are rigid then potential energy is only caused by gravity as

$$V_i = \int_{B_i} \boldsymbol{g}^T \boldsymbol{r}_i \, dm = \boldsymbol{g}^T \int_{B_i} \boldsymbol{r}_i \, dm = \boldsymbol{g}^T \boldsymbol{r}_{ci} m_i \tag{7.18}$$

where $\boldsymbol{g}$ is the gravity acceleration vector. Notice that potential energy is a function only of $\boldsymbol{q}$.

7.2.3 Computation of Lagrange Equations *

Having expressions for kinetic and potential energy the Lagrangian for a n-link robot manipulator can be written as

$$L = K - V = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}(\boldsymbol{q}) \dot{q}_i \dot{q}_j - V(\boldsymbol{q}) \tag{7.19}$$

It is now possible to calculate Lagrange equations (7.2). We divide the calculation into some smaller steps. First we have

$$\frac{\partial L}{\partial \dot{q}_i} = \sum_j d_{ij}(\boldsymbol{q}) \dot{q}_j \tag{7.20}$$

and then

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = \sum_j d_{ij}(\boldsymbol{q}) \ddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial d_{ij}}{\partial q_k} \dot{q}_k \dot{q}_j \tag{7.21}$$

Also

$$\frac{\partial L}{\partial q_i} = \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial d_{jk}}{\partial q_i} \dot{q}_k \dot{q}_j - \frac{\partial V}{\partial q_i} \tag{7.22}$$

Lagrange equations can then, after some further calculations [17], be expressed as

$$\sum_j d_{ij}(\boldsymbol{q}) \ddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} c_{ijk}(\boldsymbol{q}) \dot{q}_k \dot{q}_j + g_i(\boldsymbol{q}) = \boldsymbol{\tau}_i, \quad i = 1, ..., n \tag{7.23}$$

where c_{ijk} are termed *Christoffel symbols* and defined as

$$c_{ijk} = \frac{\partial d_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial d_{jk}}{\partial q_i} \tag{7.24}$$

and g_k is defined as

$$g_i(\boldsymbol{q}) = \frac{\partial V}{\partial q_i} \tag{7.25}$$

It is possible to write (7.23) in matrix form as

$$D(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{7.26}$$

where the elements of the matrix C is defined as

$$c_{ij} = \sum_{k=1}^{n} c_{ijk}(\boldsymbol{q})\dot{q}_k \tag{7.27}$$

7.2.4 Nonconservative Forces *

The nonconservative forces $\boldsymbol{\tau}$ are given by *actuator torques* $\boldsymbol{\tau}_a$ minus the *viscous* and *static friction* torques. The viscous friction torques are given by $F_v\dot{\boldsymbol{q}}$ and a simplified model of static torques may be $F_s sgn(\dot{\boldsymbol{q}})$. Thus

$$\boldsymbol{\tau} = \boldsymbol{\tau}_a - F_v\dot{\boldsymbol{q}} - F_s sgn(\dot{\boldsymbol{q}}) \tag{7.28}$$

where F_v and F_s are $(n \times n)$ diagonal matrices.

7.2.5 Equations of Motion

In sum, the equations of motion for a rigid n-link robot manipulator can be expressed as

$$D(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{F}_s sgn(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau_a} \tag{7.29}$$

A physical interpretation of the equations of motion shows that:

- The coefficients d_{ii} represents the moment of inertia at joint i.
- The term $c_{ijj}\dot{q}_j^2$ represent the centrifugal effect on joint i by velocity of joint j.
- The term $c_{ijk}\dot{q}_j\dot{q}_k$ represent the coriolis effect on joint i by velocity of joints j and k.
- The term g_i represent the moment generated at joint i caused by the gravity.
- The coefficients $f_{v_{ii}}$ represent viscous friction coefficients.
- The term $\boldsymbol{\tau}_{a_i}$ is the actuation torque from motor i on link i.

7.2.6 Summary

The following steps can be followed to find the equations of motion for a robot manipulator of open kinematic chain type.

1. Calculate $D(\boldsymbol{q})$
   (a) Solve forward kinematic problem and calculate the rotational matrices R_0^i.
   (b) Solve forward differential kinematic problem and calculate the Jacobian matrices J_{v_i} and J_{ω_i}.
   (c) Calculate the sum in (7.15), i.e. $D(\boldsymbol{q})$. This also require the mass and the inertia matrix for each link.
2. Calculate $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$
   (a) Calculate Christoffel symbols, (7.24).
   (b) Find $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ by calculating its elements according to (7.27).
3. Calculate $\boldsymbol{g}(\boldsymbol{q})$
   (a) Calculate $V(\boldsymbol{q})$ in (7.18). This require knowledge of the center of mass for each link.
   (b) Derive V according to (7.25).
   (c) Let $\boldsymbol{g}(\boldsymbol{q}) = [g_1, ..., g_n]^T$.
4. Finally write down the equations of motion (7.26).

$$D(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

7.3 Equations of Motion of the Gimbal System

The previous sections derived the equations of motion for a general robot manipulator of open kinematic chain type. We will now apply this on the gimbal system. We are interested in finding equations of motion for the body which the cameras are attached to and hence we use *D-H Approach 4* from Chapter 4.

7.3.1 Rigid Structure

In this section we assume that structure is rigid. Hence we can follow the steps given in Section 7.2.6 to find the equations of motion for the (rigid) gimbal system.

First we will discuss and make some approximations to simplify the calculations. Assume that the base frame is in rest (the UAV is not moving). Considering the structure of each link we realize that the center of mass of each link is near its rotational axis, moreover the body that the cameras are attached to, link 4, should be balanced. Thus the center of mass moves a very small distances when the joint variables change. Then it is not a far-fetched assumption to let the center of mass for each link be in the point of intersection between joint 1 and joint 2 axes. This point is the origin for all four link reference frames.

All links are, more or less, asymmetric. This make the inertia matrices I_i non-diagonal and thus we have six unknown parameters for each link. By assuming that I_i are diagonal we can reduce the number of unknown parameters. Also let I_3 be 0, since the mass of link 3 is negligible compared to the other links. We also assume that the joint axes i and $i+1$, $i = 1, 2, 3$, are orthogonal.

Furthermore, we approximate the flexibility angles θ_3 and θ_4. Since the angles are small we can approximate $\sin(\theta_i)$ with θ_i and $\cos(\theta_i)$ with 1, $i = 3, 4$.

Now we are ready to apply the algorithm in Section 7.2.6 on the gimbal system.

Step 1. Calculate $D(\boldsymbol{q})$

The rotational matrices R_i, $i = 1, 2, 3, 4$, and the Jacobian matrices J_{v_i} and J_{ω_i}, $i = 1, 2, 3, 4$, are already calculated in Chapters 3 and 6 respectively. We assume that the inertia matrices I_i, $i = 1, 2, 4$, are diagonal and $I_3 = \mathbf{0}$. Thus

$$I_1 = diag[I_{1_{11}}, I_{1_{22}}, I_{1_{33}}] \tag{7.30}$$

$$I_2 = diag[I_{2_{11}}, I_{2_{22}}, I_{2_{33}}] \tag{7.31}$$

$$I_3 = \mathbf{0} \tag{7.32}$$

$$I_4 = diag[I_{4_{11}}, I_{4_{22}}, I_{4_{33}}] \tag{7.33}$$

Now we can calculate $D(\boldsymbol{q})$ as the sum in (7.15).

$$\begin{aligned} D(\boldsymbol{q}) &= \textstyle\sum_{i=1}^{n} [m_i J_{v_i}(\boldsymbol{q})^T J_{v_i}(\boldsymbol{q}) + J_{\omega_i}(\boldsymbol{q})^T R_i(\boldsymbol{q}) I_i R_i(\boldsymbol{q})^T J_{\omega_i}(\boldsymbol{q})] \\ &= \textstyle\sum_{i=1}^{n} J_{\omega_i}(\boldsymbol{q})^T R_i(\boldsymbol{q}) I_i R_i(\boldsymbol{q})^T J_{\omega_i}(\boldsymbol{q}) \end{aligned} \tag{7.34}$$

Since $J_{v_i} = 0$, $i = 1, 2, 3, 4$, the first term in the sum (7.34) also equals zero for all i. The elements of $D(\boldsymbol{q})$ are

$$\begin{aligned} d_{11} = {} & I_{1_{22}} + (I_{2_{33}} + I_{4_{33}} + I_{4_{11}}\theta_4^2)\cos(\theta_2)^2 + (I_{4_{11}} - I_{4_{33}})\theta_4 \sin(2\theta_2) \\ & + (I_{2_{11}} + I_{4_{11}} + I_{4_{22}}\theta_3^2 + I_{4_{33}}\theta_4^2)\sin(\theta_2)^2 + \\ d_{12} = {} & d_{21} = (-\theta_3)((I_{4_{11}} - I_{4_{33}})\theta_4\cos(\theta_2) + (I_{4_{11}} - I_{4_{22}} + I_{4_{33}}\theta_4^2)\sin(\theta_2)) \\ d_{13} = {} & d_{31} = (I_{4_{33}} + I_{4_{11}}\theta_4^2)\cos(\theta_2) + (I_{4_{11}} - I_{4_{33}})\theta_4\sin(\theta_2) \\ d_{14} = {} & d_{41} = I_{4_{22}}\theta_3(1 + \theta_3^2)\sin(\theta_2) \\ d_{22} = {} & I_{2_{22}} + I_{4_{22}} + \theta_3^2(I_{4_{11}} + I_{4_{33}}\theta_4^2) \\ d_{23} = {} & (-I_{4_{11}} + I_{4_{33}})\theta_3\theta_4 \\ d_{24} = {} & I_{4_{22}}(1 + \theta_3^2) \\ d_{33} = {} & I_{4_{33}} + I_{4_{11}}\theta_4^2 \\ d_{34} = {} & d_{43} = 0 \\ d_{44} = {} & I_{4_{22}}(1 + \theta_3^2)^2 \end{aligned} \tag{7.35}$$

Note that $d_{ij} = d_{ji}$.

Step 2. Calculate $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$

Finding the elements of $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is straight forward. First we compute the Christoffel symbols according to (7.24) and then the C elements according to (7.27). The elements of $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ can be found in Appendix D.

Step 3. Calculate $\boldsymbol{g}(\boldsymbol{q})$

Gravity will not affect the dynamics since the position of the center of mass is constant. Thus we have

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0} \tag{7.36}$$

Step 4. Equations of motion

In sum, we have these equations

$$D(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{7.37}$$

7.3.2 Friction

We neglect the static friction and the viscous friction is represented by a diagonal matrix F_v as

$$F_v = diag[f_{v_1}, f_{v_2}, f_{v_3}, f_{v_4}] \tag{7.38}$$

Then we can express $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = \boldsymbol{\tau_a} - F_v\dot{\boldsymbol{q}} \tag{7.39}$$

where $\boldsymbol{\tau_a}$ are the actuator torques from the motors

$$\boldsymbol{\tau_a} = \begin{bmatrix} \tau_{a1} \\ \tau_{a2} \\ 0 \\ 0 \end{bmatrix} \tag{7.40}$$

7.3.3 Flexible Links

Joints 1 and 2 are controllable by DC-motors, but joint 3 and 4 are an approach to model the known rotational damping flexibility in the gimbal system. In a formal manner the gimbal system only has two joints, joint 1 and 2, since these joints are the controllable, and the damping flexibility of the inner body is a kind of link flexibility. However, since the rotational axes of these flexibilities are known it is advantageous to consider them as two extra joints.

We will model the flexibilities of joint 3 and 4 as a sum of two springs. First consider a linear spring which will model the magnetic force acting on the inner body. The potential spring energy can be expressed as

$$V_{s1} = \frac{1}{2}\alpha q_i^2; \quad i = 3, 4 \tag{7.41}$$

where α is the spring constant and q is the displacement. However, this is not enough to model the constraints. We need another spring with a "spring constant"[1] that is near zero when the displacement is small and with a sharp edge for a certain displacement. One proposal is

$$c(q) = \beta q^{2n_f} \tag{7.42}$$

Approximately we can say that n_f describes how sharp the constraint is and β determines the "position" of the constraint. However, the constraints are not "square" but "circular". Figure 7.1 shows a circular device that is attached to the camera assembly. If the camera assembly is rotated too much, relative to the outer gimbal, this device hits a circular wall that covers the device and is attached to the outer gimbal. Thus we have "circular" flexibility constraints. Furthermore, a magnet provides a centering force.

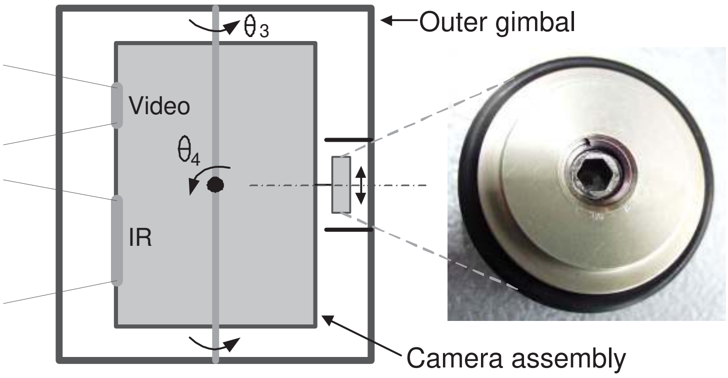


Figure 7.1. Device that prevents the damping flexibility to rotate to much.

Consider the equation of a circle $x^2 + y^2 = r^2$ where r is the radius. Replace q in (7.42) with $q_x^2 + q_y^2$ and we can propose a better model of the spring constant as

$$c(q_x, q_y) = \beta(q_x^2 + q_y^2)^{n_f} \tag{7.43}$$

Hence, let the potential energy of the second spring be

$$V_{s2} = \frac{1}{4(n+1)}\beta(q_i^2 + q_j^2)^{n_f+1}; \quad i,j = 3,4; \quad i \neq j \tag{7.44}$$

and the total potential energy of the flexibilities can be expressed as

$$V_s = \frac{1}{2}\alpha q_3^2 + \frac{1}{2}\alpha q_4^2 + \frac{1}{2}\frac{1}{n+1}\beta(q_3^2 + q_4^2)^{n_f+1} \tag{7.45}$$

Now apply Lagrange equations on V_s.

$$\frac{d}{dt}\frac{\partial V_s}{\partial \dot{q}_i} - \frac{\partial V_s}{\partial q_i} = q_i(\alpha + \beta(q_i^2 + q_j^2)^{n_f}); \quad i,j = 3,4; \quad i \neq j \tag{7.46}$$

[1] Of course this "spring constant" is not a constant.

Let

$$k(\boldsymbol{q}) = \alpha + \beta(q_3^2 + q_4^2)^{n_f} \tag{7.47}$$

and

$$K(\boldsymbol{q}) = diag[0, 0, k(\boldsymbol{q}), k(\boldsymbol{q})] \tag{7.48}$$

and we can express (7.46) in matrix form as

$$K(\boldsymbol{q})\boldsymbol{q} \tag{7.49}$$

7.3.4 Equations of Motion

Finally we complement the equations of motion for a rigid structure (7.37) with (7.39) and (7.49). The equations of motion can then be expressed as

$$D(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + F_v\dot{\boldsymbol{q}} + K(\boldsymbol{q})\boldsymbol{q} = \boldsymbol{\tau_a} \tag{7.50}$$

Since $D(\boldsymbol{q})$ is a symmetric positive definite matrix we can rewrite (7.50) in a nonlinear *state-space form*

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{7.51}$$

where $\boldsymbol{x}$ are the *state variables.* Let the state variables be

$$\boldsymbol{x} = [\boldsymbol{x}_\theta \quad \boldsymbol{x}_\omega]^T \tag{7.52}$$

where

$$\boldsymbol{x}_\theta = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4]^T \tag{7.53}$$

$$\boldsymbol{x}_\omega = [\omega_1 \quad \omega_2 \quad \omega_3 \quad \omega_4]^T \tag{7.54}$$

Then we have

$$\begin{bmatrix} \dot{\boldsymbol{x}}_\theta \\ \dot{\boldsymbol{x}}_\omega \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_\omega \\ D(\boldsymbol{x}_\theta)^{-1}(\boldsymbol{\tau}_a - C(\boldsymbol{x}_\theta, \boldsymbol{x}_\omega)\boldsymbol{x}_\omega - F_v\boldsymbol{x}_\omega - K(\boldsymbol{x}_\theta)\boldsymbol{x}_\theta) \end{bmatrix} \tag{7.55}$$

A state-space model is very useful in simulations of the dynamics.

7.3.5 Actuator Dynamics

The torques τ_{a_1} and τ_{a_2} are produced by two actuators, two DC-motors. Since we are able to control the voltage to the DC-motor instead of the torque directly, we must find a relationship between the voltage and the torque.

A DC-motor basically works on the principle that a current carrying conductor in a magnetic field experience a force. In the DC-motor this is a torque which will cause a rotor (armature) to rotate. The magnitude of the torque is

$$\tau = K_m i \tag{7.56}$$

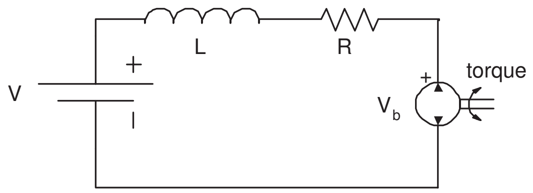


Figure 7.2. Simple model of a armature controlled DC motor.

where K_m is a physical constant and i is the current. In addition a conductor moving in a magnetic field will generate a *back emf* proportional to the velocity and opposing the current flow. Thus the back emf V_b is

$$V_b = K_b \omega \tag{7.57}$$

where K_b is a constant and ω the angular velocity.

Consider the circuit diagram for an armature controlled DC-motor in Figure 7.2. Hence vi can express the current with a differential equation

$$L\frac{di}{dt} + Ri = V - V_b \tag{7.58}$$

However, we ignore the inductance L and then the actuator torque can be expressed as

$$\tau = \frac{K_m}{R}(V - K_b \omega) \tag{7.59}$$

Thus we can express the actuator torques in the gimbal system as

$$\tau_{a_i} = \frac{K_{m_i}}{R_i}(V_i - K_{b_i}\omega_i), \quad i = 1, 2 \tag{7.60}$$

7.3.6 Approximations and Assumptions

We summarize all approximations and assumptions we have done when we calculated the equations of motion for the gimbal system.

- The center of mass for each link is in the point of intersection between the first and second axes.
- I_i, $i = 1, 2, 4$, are diagonal matrices and $I_3 = \mathbf{0}$.
- Joint axes i and $i + 1$, $i = 1, 2, 3$, are orthogonal.
- Static friction is neglected. Viscous friction is linear.
- Link flexibilities are modelled as nonlinear springs.
- Linearization of θ_3 and θ_4. Since θ_3 and θ_4 are small we let $\cos(\theta_i) = 1$ and $\sin(\theta_i) = \theta_i$, $i = 3, 4$.
- Inductances in the DC-motors are assumed to be $L = 0$.

References & Further Readings

The theory of robot manipulator dynamics in Sections 7.1, 7.2 and 7.3.2 is a combination of [17], [18] and [15]. More about analytical mechanics and Lagrange equations can be found in [6]. Section 7.3.5 about actuator dynamics are from [18].

Chapter 8

Identification

In Chapter 7 we derived a dynamic model of the gimbal system. However, the dynamic parameters are not known and this chapter will discuss the problem of dynamic parameter identification.

8.1 Linear Regression Model

The dynamic model is nonlinear, but if we neglect the nonlinear part of the link flexibility, the dynamic model is linear with respect to the dynamic parameters, Figure 8.1. This is an important property that will make the estimation of the dynamic parameters easier.

We will express the dynamics as a *linear regression model*

$$\boldsymbol{y} = \boldsymbol{\Theta}^T \boldsymbol{\phi} \tag{8.1}$$

where $\boldsymbol{\Theta}$ is the *parameter vector* and $\boldsymbol{\phi}$ is the *regression vector*.

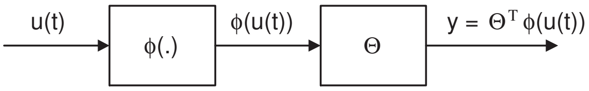


Figure 8.1. The dynamic model is linear with respect to the dynamic parameters and can be expressed as a linear regression model.

Note that $\boldsymbol{u}(t)$ are functions of joint position, velocities and accelerations, i.e. the *output* signals from the gimbal system, and $\boldsymbol{y}(t)$ are the torques, i.e the *input* signals to the gimbal system.

8.2 Least Square Estimate *

Let

$$\hat{y}(t|\Theta) = \Theta^T \phi(t) \tag{8.2}$$

be the prediction of $y(t)$ where

$$\Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_d \end{bmatrix} \tag{8.3}$$

We can express the prediction error as

$$\epsilon(t, \Theta) = y(t) - \hat{y}(t|\Theta) \tag{8.4}$$

Given input and output data over a period $t = 1, ..., N$ we can evaluate the parameter values by a measure, e.g. the *Least Square Criterion*[1]

$$V_N(\Theta) = \frac{1}{N} \sum_{t=1}^{N} \frac{1}{2} \epsilon^2(t, \Theta) \tag{8.5}$$

Thus an obvious choice of Θ is the one that minimize (8.5)

$$\hat{\Theta}_N = \arg \min_{\Theta} V_N \tag{8.6}$$

A positive feature of the choice (8.5) is that (8.6) can be minimized analytically. (8.5) can be expressed as

$$V_N(\Theta) = \frac{1}{N} \sum_{t=1}^{N} [y(t) - \hat{\Theta}^T \phi(t)]^2 = \frac{1}{N} \sum_{t=1}^{N} y^2(t) - 2\Theta^T f_N + \Theta^T R_N \Theta \tag{8.7}$$

where

$$f_N = \frac{1}{N} \sum_{t=1}^{N} \phi(t) y(t) \tag{8.8}$$

$$R_N = \frac{1}{N} \sum_{t=1}^{N} \phi(t) \phi^T(t) \tag{8.9}$$

If matrix R_N is invertible (8.7) can be rewritten as

$$V_N(\Theta) = \frac{1}{N} \sum_{t=1}^{N} y^2(t) - f_N^T R_N^{-1} f_N + (\Theta - R_N^{-1} f_N)^T R_N (\Theta - R_N^{-1} f_N) \tag{8.10}$$

The first two terms are independent of Θ and the last term is always positive, since R_N is positive definite. Hence the V_N is minimized when the last term is zero and thus we have the *Least Square Estimate* (LSE) as

$$\Theta = \hat{\Theta}_N = R_N^{-1} f_N \tag{8.11}$$

[1] Sometimes the factor 1/2 in (8.5) is omitted. However, this factor is unessential to the further analysis.

8.2.1 Matrix Formulation *

For calculations we prefer to express the equations above in matrix form. Define the $N \times 1$ vector

$$Y_N = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \tag{8.12}$$

and the $N \times d$ matrix

$$\Phi_N = \begin{bmatrix} \phi^T(1) \\ \vdots \\ \phi^T(N) \end{bmatrix} \tag{8.13}$$

Then the criterion (8.5) can be written as

$$V_N(\Theta) = \frac{1}{N}(Y_N - \Phi_N\Theta)^T(Y_N - \Phi_N\Theta) \tag{8.14}$$

end the estimate (8.11) as

$$\hat{\Theta}_N = (\Phi_N^T\Phi_N)^{-1}\Phi_N^T Y_N \tag{8.15}$$

(8.15) gives the *pseudoinverse* solution to the overdetermined system[2] of linear equations

$$Y_N = \Phi_N\Theta \tag{8.16}$$

8.2.2 Multivariable Formulation *

The multivariable case is analogues. Let

$$\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix} \tag{8.17}$$

and

$$\boldsymbol{\phi}(t) = \begin{bmatrix} \phi_1(t) \\ \vdots \\ \phi_m(t) \end{bmatrix} \tag{8.18}$$

Now redefine (8.12) and (8.13) as the $mN \times 1$ vector

$$\boldsymbol{Y}_N = \begin{bmatrix} \boldsymbol{y}(1) \\ \vdots \\ \boldsymbol{y}(N) \end{bmatrix} \tag{8.19}$$

[2] The system is overdetermined if $N > d$.

and the $mN \times d$ matrix

$$\mathbf{\Phi}_N = \begin{bmatrix} \boldsymbol{\phi}^T(1) \\ \vdots \\ \boldsymbol{\phi}^T(N) \end{bmatrix} \tag{8.20}$$

The estimate $\hat{\Theta}_N$ is in the multivariable case similar to (8.15).

$$\hat{\Theta}_N = (\mathbf{\Phi}_N^T \mathbf{\Phi}_N)^{-1} \mathbf{\Phi}_N^T \boldsymbol{Y}_N \tag{8.21}$$

8.2.3 ARX *

The *ARX-model* is defined as

$$A(q)y(t) = B(q)u(t) + e(t) \tag{8.22}$$

where A and B polynomial functions of the forward shift operator q. Assume that we have n input signals denoted as $u(t)$. If $A = 1$ and $B = [b_1, b_2, ..., b_n]^T$ the ARX-model is equivalent with the linear regression model.

The function *arx* in the MATLAB toolbox System Identification Toolbox (SITB) estimates the parameters of an ARX-model. The parameters are estimated using a LSE method. Hence we will get the same result as in (8.21).

8.3 Model Properties *

8.3.1 Variance *

Assume that the true parameters are given by Θ_0 and that

$$y(t) - \hat{y}(t|\Theta_0) = e(t) \tag{8.23}$$

is white noise with variance λ. The covariance matrix for the estimate $\hat{\Theta}_N$ is approximately

$$\hat{P}_N = E[(\hat{\Theta}_N - \Theta_0)(\hat{\Theta}_N - \Theta_0)^T] \approx \frac{1}{N}\lambda \bar{R}^{-1} \tag{8.24}$$

where

$$\bar{R} = E[\psi(t,\Theta_0)\psi^T(t,\Theta_0)] \tag{8.25}$$

$$\psi(t,\Theta_0) = \frac{d}{d\Theta}\hat{y}(t|\Theta) \tag{8.26}$$

The parameter variances are the diagonal elements in $\hat{P}_N$ and according to (8.24) the variances are proportional to the noise variance λ and to the inverse of the number of measurements N.

8.3.2 Fit *

We will use a measurement called *fit* to evaluate the model outputs y_m compared to y the true output in simulations or measured output in experiments. *Fit* is the percentage of the output variation that is reproduced by the model. A model that has a fit of 0% gives the same mean square error as just setting the model output to be the mean of the measured output.

$$fit = 100(1 - \frac{||y_m - y||}{||y - \bar{y}||}) \tag{8.27}$$

where $\bar{y}$ is the mean value of y.

8.4 Identifiability

Define Θ^* as

$$\Theta^* = \lim_{N \to \infty} \hat{\Theta}_N \tag{8.28}$$

A certain set of parameters are *identifiable* at Θ^* if

$$\hat{y}(t|\Theta^*) \equiv \hat{y}(t|\Theta) \Rightarrow \Theta = \Theta^* \tag{8.29}$$

One reason that makes the parameters not identifiable is that the data set is not informative enough, i.e. the rank of $\mathbf{\Phi}_N^T \mathbf{\Phi}_N$ is not full and the estimate (8.21) can not be calculated.

Consider the simple model

$$\hat{y}(t|\Theta) = b_1 u(t-1) + b_2 u(t-2), \quad \Theta = [b_1 \quad b_2]^T, \quad \phi = [u(t-1) \quad u(t-2)] \tag{8.30}$$

If the input signal is constant $u(t) \equiv u_0$ then the prediction becomes

$$\hat{y}(t|\Theta) = (b_1 + b_2)u_0 \tag{8.31}$$

The value of the sum $b_1 + b_2$ can be estimated, but it is impossible to know b_1 and b_2. Thus b_1 and b_2 are not identifiable. Note that $\phi^T \phi$ has not full rang.

Obviously the identifiability depends on the trajectory, however there might be problems where no trajectory exist that makes the rank of $\mathbf{\Phi}_N^T \mathbf{\Phi}_N$ full. If this problem occurs, the parameter vector Θ must be reduced by one or more parameters and other methods must be used to identify these parameters.

8.5 Optimal Trajectories for Dynamic Identification

In previous section we realized that the choice of trajectory is very important. It is necessary that $\mathbf{\Phi}_N^T \mathbf{\Phi}_N$ has full rank, however it will not guarantee that we will get a good estimates of the parameters. It is important that the motion excites all aspects of the dynamics of the system and generates data that is informative enough.

Assume that the rank of $\mathbf{\Phi}_N^T\mathbf{\Phi}_N$ is full. We define a condition numbers κ as the ratio of the largest eigenvalue to the smallest eigenvalue of $\mathbf{\Phi}_N^T\mathbf{\Phi}_N$. Thus

$$\kappa = \frac{\max(\lambda_i)}{\min(\lambda_i)} \tag{8.32}$$

where λ_i, $i = 1, 2, ...$ are the eigenvalues of $\mathbf{\Phi}_N^T\mathbf{\Phi}_N$. If $\mathbf{\Phi}$ is scaled, i.e. the elements of $\mathbf{\Phi}$ are of similar size, it is clear that that matrix $\mathbf{\Phi}_N$ is well-conditioned if κ is small. Thus we search for a trajectory that results in a small κ. However, finding a proper input trajectory is not a trivial problem since the signals $\mathbf{\Phi}$ are nonlinear functions of the position, velocities and accelerations. Furthermore we also have to take the constraints into consideration. The gimbal system has not only constraints in joint 2, but also joint 3 and 4 have to be in a limited range.

There exist sophisticated mathematical methods for the optimal trajectory search (e.g. in [10]). However, we will use a simpler Monte Carlo method, which essentially is a trial and error search. First we define input signals $\boldsymbol{\tau}_a$, e.g. a sum of sinus signals

$$\tau_{a_i} = \sum_j a_j \sin(k_j t + f_j) \tag{8.33}$$

For each choice of signal property parameters a_j, k_j, f_j the dynamics are simulated and the condition number κ is calculated. The parameters corresponding to minimum κ is considered as the best input signals. If the simulation results in joint values that are not in the allowed range the set of parameters are discarded. For better efficiency, we ”zoom” in on a neighbourhood of a good set of parameters. The method can be summarized in the following steps.

1. Initialize
2. For $m = 1, 2, ..., m_{stop}$ do steps 3-10.
3. For $n = 1, 2, ..., n_{stop}$ do steps 4-9.
4. Set the signal property parameters to random numbers (within an allowed range) plus the best set of parameters so far in the outer loop.
5. Compute the input signal.
6. Simulate the system dynamics.
7. If joint values are not in allowed range or the rank of $\mathbf{\Phi}_N^T\mathbf{\Phi}_N$ is not full, skip step 8 and 9.
8. Compute the condition number.
9. If the condition number is smaller than previous condition numbers, store the parameters and the condition number.
10. Decrease the allowed range of the random numbers.

8.6 Linear Regression Model of the Gimbal System

We will now apply the LSE method to the identification problem of the gimbal dynamics parameters. First we will express the dynamic model of the gimbal in the linear regression form.

The parameter vector contains all dynamic parameters except n_f that we ignore and the inertia parameters $I_{1_{11}}$ and $I_{1_{33}}$ since the model does not depend on these parameters. Thus we define $\mathbf{\Theta}$ as

$$\mathbf{\Theta} = [I_{1_{22}} \quad I_{2_{11}} \quad I_{2_{22}} \quad I_{2_{33}} \quad I_{4_{11}} \quad I_{4_{22}} \quad I_{4_{33}} \quad f_{v_1} \quad f_{v_2} \quad f_{v_3} \quad f_{v_4} \quad \alpha \quad \beta]^T \tag{8.34}$$

The next problem is to find $\boldsymbol{\phi}(t)$. We start from (7.15) and collect all inertia parameters so we can express K as

$$K = \mathbf{\Theta}^T \boldsymbol{W}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{8.35}$$

Since the potential energy V is zero the Lagrangian is $L = K$. Use Lagrange equations and finally add friction $F_v\dot{\boldsymbol{q}}$ and flexibility $K\boldsymbol{q}$. From this result we can easily express the dynamics on linear regression form (8.1).

8.7 Simulations

Before identification on the physical gimbal system, we first identify the dynamic parameters of our dynamic model. A desired property, mention in Section 8.5, is that the parameters are scaled to similar size. Hence, in the identification simulations we set all dynamic parameters equals one

$$\mathbf{\Theta} = [1 \quad ... \quad 1]^T \tag{8.36}$$

The simulations are executed in MATLAB and SITB and follows these steps:

1. Initialize dynamic parameters, sampling frequency, simulation time and input data.

2. Simulate the dynamics of the gimbal system.

3. Calculate joint accelerations and possibly add noise to joint position, velocities and accelerations.

4. Form $\mathbf{\Phi}_N$ and Y_N.

5. Compute the estimation (8.21). We also estimated a black-box state-space model of order 8, the same order as the dynamic model.

6. Compare the results with the correct answer and simulate the dynamic model twice with a new input data, first with correct dynamic parameters and then with the estimated dynamics parameters.

Sampling time is 0.01 s and the simulation length is 30 s. The *Error* of an estimate is defined as the euclidian vector norm of the difference between the estimate and the correct value.

After the first simulations we realized that the parameters are not identifiable, since the rank of $\mathbf{\Phi}_N^T \mathbf{\Phi}_N$ is not full for any of our choices of input. Thus we must reduce the parameter vector. $I_{1_{22}}$ is the first parameter to remove since we suspect that this parameter can not be identified alone. We assume that we know $I_{1_{22}}$ and subtract the contribution of $I_{1_{22}}$ from the input data. After this modification the rank of $\mathbf{\Phi}_N^T \mathbf{\Phi}_N$ is full. In the simulations below we do not identify β either.

In the easiest case we ignore the mechanical constraints and do not add any noise. In this case it is not necessary to search a trajectory as in Section 8.5, we just use a filtered pseudo-random binary signal. Estimate 1 in Table 8.2 shows the identification result. The condition number is 41.

In the next simulation example we take the constraints into consideration. First a trajectory that results in a minimum condition number and joint values that are in the allowed range, defined in Table 8.1, was searched for. Each input torque is a sum of three sinus signals and a chirp signal and we executed $m_{stop} \times n_{stop} = 10 \times 100$ iterations. The best trajectory, Figure 8.2, resulted in a condition number of 165. Figure 8.3 shows the joint position trajectories from simulation and Estimate 2 in Table 8.2 shows the dynamic identification result. The dynamic model based on the identified parameters is simulated with a new input signal and compared to a simulation based on the correct parameters. The differences between the two simulations are less than $5e^{-3}$ for θ_i, $i = 1, 2, 3, 4$.

Joint	Min	Max
θ_2	$-\frac{\pi}{3}$	$\frac{\pi}{2}$
θ_3	-0.1	0.1
θ_4	-0.1	0.1

Table 8.1. Joint constraints.

At last we add noise to $\boldsymbol{\phi}$ and $\boldsymbol{y}$. The energy of the noise is about 0.5% of the energy of $\boldsymbol{\phi}$ and $\boldsymbol{y}$ respectively. We use the same input signals as in the previous simulation. Estimate 3 in Table 8.2 shows the parameter result. In Figure we compare two simulations, one with the correct parameters and one with the estimated parameters. The input signals are filtered pseudo-random binary signals.

A linear black-box model of order 8 (same order as our dynamic model) is also estimated. The model is then validated against the true nonlinear model, Figure 8.5. The result is fairly good for θ_1 and θ_2, but bad for the flexibility angles, especially θ_3.

Parameter	True Value	Estimate 1	Estimate 2 (constraints)	Estimate 3 (constraints & noise)
$I_{2_{11}}$	1.0000	0.9880	0.9801	0.9954
$I_{2_{22}}$	1.0000	0.9942	0.9908	0.9979
$I_{2_{33}}$	1.0000	0.9940	0.9749	0.9915
$I_{4_{11}}$	1.0000	1.0093	1.0170	1.0051
$I_{4_{22}}$	1.0000	1.0068	1.0093	0.9623
$I_{4_{33}}$	1.0000	1.0076	1.0197	0.9880
α_3	1.0000	1.0061	1.0157	0.9769
α_4	1.0000	1.0010	1.0064	0.9223
f_{v_1}	1.0000	1.0055	1.0038	1.0065
f_{v_2}	1.0000	1.0036	1.0001	1.0038
f_{v_3}	1.0000	1.0102	1.0249	0.9817
f_{v_4}	1.0000	1.0116	1.0140	0.9306
Error	-	0.0269	0.0547	0.1161
Fit θ_1	-	98.03 %	99.47 %	99.66 %
Fit θ_2	-	99.27 %	99.62 %	98.46 %
Fit θ_3	-	99.39 %	99.29 %	98.71 %
Fit θ_4	-	99.59 %	99.45 %	94.39 %
Var$[I_{2_{11}}]$	-	-	$0.369 \cdot 10^{-4}$	$2.512 \cdot 10^{-4}$
Var$[I_{2_{22}}]$	-	-	$0.093 \cdot 10^{-4}$	$0.744 \cdot 10^{-4}$
Var$[I_{2_{33}}]$	-	-	$0.089 \cdot 10^{-4}$	$0.703 \cdot 10^{-4}$
Var$[I_{4_{11}}]$	-	-	$0.319 \cdot 10^{-4}$	$2.169 \cdot 10^{-4}$
Var$[I_{4_{22}}]$	-	-	$0.092 \cdot 10^{-4}$	$0.703 \cdot 10^{-4}$
Var$[I_{4_{33}}]$	-	-	$0.059 \cdot 10^{-4}$	$0.458 \cdot 10^{-4}$
Var$[\alpha_3]$	-	-	$0.087 \cdot 10^{-4}$	$0.677 \cdot 10^{-4}$
Var$[\alpha_4]$	-	-	$0.192 \cdot 10^{-4}$	$1.460 \cdot 10^{-4}$
Var$[f_{v_1}]$	-	-	$0.006 \cdot 10^{-4}$	$0.046 \cdot 10^{-4}$
Var$[f_{v_2}]$	-	-	$0.009 \cdot 10^{-4}$	$0.073 \cdot 10^{-4}$
Var$[f_{v_3}]$	-	-	$0.092 \cdot 10^{-4}$	$0.726 \cdot 10^{-4}$
Var$[f_{v_4}]$	-	-	$0.247 \cdot 10^{-4}$	$1.878 \cdot 10^{-4}$

Table 8.2. Dynamic parameter estimation results.

8.8 Comments on the Experimental Identification

Unfortunately we could not present any identification experiment results on the physical system in this thesis, since the measurement equipment was not ready. However, in this section some comments on the experimental identification are given.

A successful identification result in the simulation does not guarantee that the identification on the physical system will be successful. Our dynamic model must be good enough, which we still do not know if it is, and there are always disturbances and dynamics that are impossible to include in our model. Maybe we have to modify and refine the model, e.g. by a better friction model and/or by introducing a flexible joint model.

It is naive to think that the identification can be performed in one single experiment, as in the simulations. The trajectory planning approach to excite the system in a proper way is possible insufficient. An alternative is to perform experiment in a neigbourhood of a certain θ_2 and identify some of the parameters and then perform new experiments in a neigbourhood of a new choice of θ_2 and identify some other parameters, and so on. Since we know that the data has to be scaled in a proper way, we can re-scale as the estimates of the parameters become more accurate. Hopefully this strategy makes the estimates of the dynamic parameters converge.

In our model we use a very simple friction model which only consider the viscous friction. In Chapter 7 we also mentioned static friction, however, the friction torque is in fact a nonlinear and very complex function of different effects as static friction, viscous friction, stiction, breakaway, slip stick and negative friction. In order to minimize the effects of unmodelled dynamics in the dynamics equations of motion it is important to take these effects into consideration. Sometimes it is advantageous to identify the friction separately from the rest of the dynamic parameters. Once a friction model is estimated the friction torque can easily be subtracted from the generalized torque.

References & Further Readings

The use of LSE in robotics can be found in [10] but the method is also mentioned in [17] and [2]. Sections 8.2 and 8.4 about LSE and identifiability summarizes parts in [11] and [13]. A detailed analysis of system identification and LSE can be found in [11]. The definition of the measurement *fit* is from [12].

The choice of condition number and the idea of the Monte Carlo search in Section 8.5 are from [10], where more information and analysis about optimal trajectory search and robot dynamic identification can be found. An example of friction parameter estimation that is performed separately from the rest of the dynamic parameters can be found in [10].

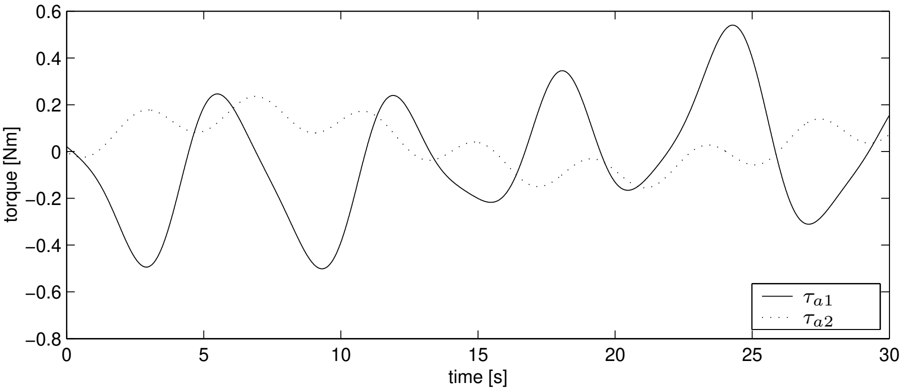


Figure 8.2. Input torque trajectories.

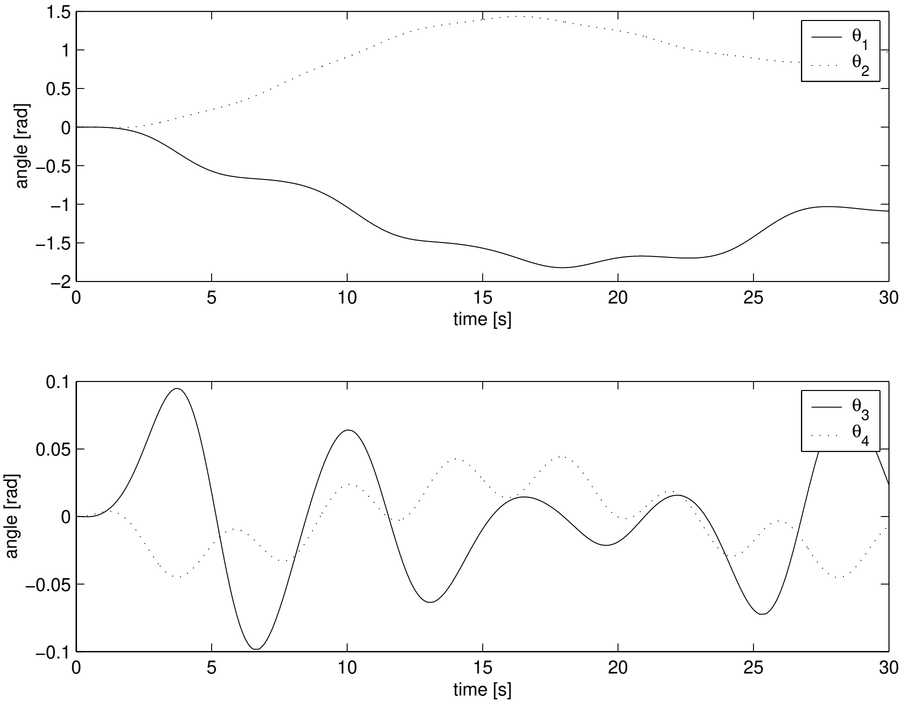


Figure 8.3. The joint position trajectories from simulation with input torque as in Figure 8.2.

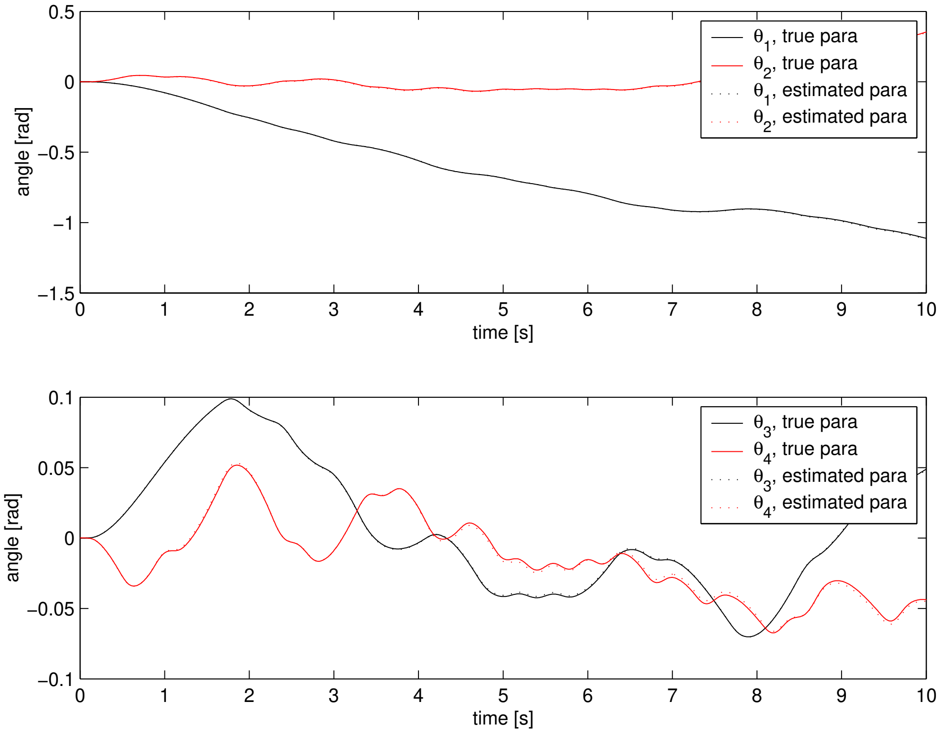


Figure 8.4. Comparison between two simulations, one with the correct parameters and one with the estimated parameters (Estimation 3).

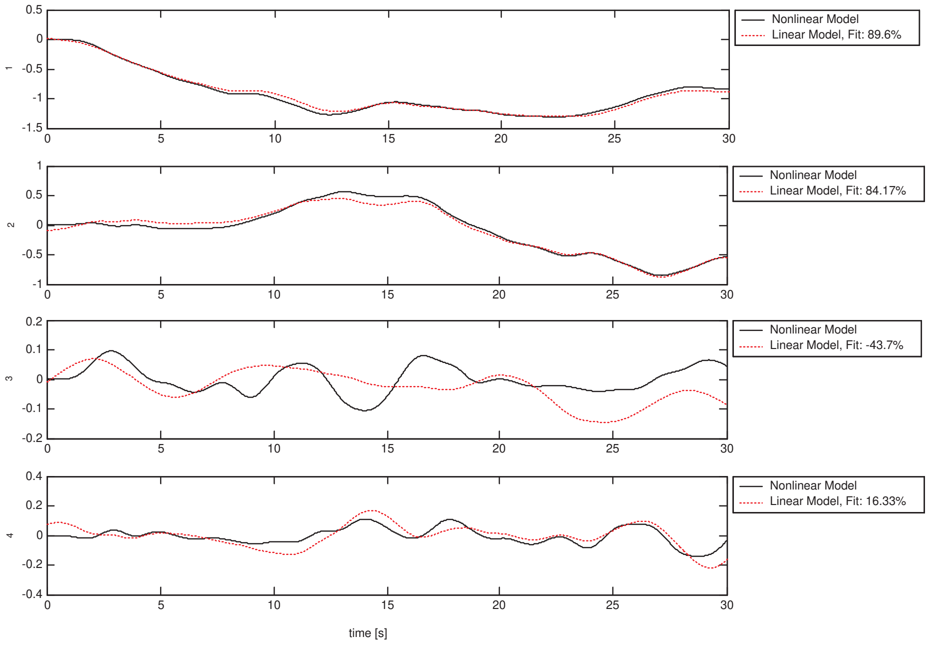


Figure 8.5. Simulated nonlinear model output and simulate linear black-box model output.

Chapter 9

Gimbal Motion Control

The motion controller task is to control the gimbal to follow the orientation reference signal from the trajectory planner as good as possible. This task is not trivial since the gimbal is a nonlinear multivariable system. Our goal is a robust control system that responds quickly and that does not excite the damping flexibility too much, goals that are contradictory.

The orientation reference signals γ_1 and γ_2 are defined in Section 5.1.1. We assume that the trajectory planner generates reference signals that are executable, i.e. the reference signals are in the allowed joint ranges defined by the pan and tilt joints.

9.1 Approximations & Simplifications

9.1.1 Decentralized & Centralized Control Structure

The gimbal system is a multivariable system. The simplest approach to control the system is to assume that the control of the two outer joints are *decoupled*. This means that each actuator torque τ_{a_i}, $i = 1, 2$, influence only the corresponding component θ_i. Thus we can consider the gimbal system as two *single-input/single-output systems* (SISO) and this leads to a *decentralized* control structure. In this approximation the coupling effects are considering as disturbances.

However, large operational speeds causes that the coupling effect strongly influence the system performance. *Centralized* control structures take advantages of knowledge of the dynamics to compensate for coupling effects. Often the coupling effects are reduced by adding a compensating torque based on the knowledge of the dynamics, e.g. *computed torque feedforward control.*

9.1.2 Linearization

A common and often very successful regulator design method is to design the regulator based on the linearization of the nonlinear dynamic model. An expansion of this method is to design a number of regulators based on linearization in different points. The resulting regulator is then calculated by interpolation of these regulators depending on the current state. This method is called *gain scheduling.*

9.1.3 Output Signals

It is not obvious which output signals from the gimbal system that we will control. The reference is the desired orientation, but the the actual orientation is a nonlinear and rather complex function of $\theta_1, \theta_2, \theta_3, \theta_4$.

A first basic approximation would of course be to control θ_1, θ_2 to follow the reference signals and ignoring the influence of the flexibility θ_3, θ_4. Thus

$$\begin{aligned} y_1 &= \theta_1 \\ y_2 &= \theta_2 \end{aligned} \tag{9.1}$$

However, this require that the references do not excite the flexibility too much.

A second approximation of the output signal is

$$\begin{aligned} y_1 &= \theta_1 + \theta_3 \\ y_2 &= \theta_2 + \theta_4 \end{aligned} \tag{9.2}$$

This approximation is good when $\theta_2 \approx 0$, but bad when $\theta_2 \approx \frac{\pi}{2}$.

9.2 Classical Control, PID

PID-regulator is a classical regulator that consists of a proportional (P), an integrating (I) and a differential (D) part

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \tag{9.3}$$

where $e(t) = r(t) - y(t)$ is the error between the reference $r(t)$ and the output $y(t)$ and K_P, K_I and K_D are constants.

A time-discrete PID-regulator can be expressed as

$$u_n = K_P e_n + K_I T_s \sum_{k=0}^{n} e_k + K_D \frac{e_n - e_{n-1}}{T_s} \tag{9.4}$$

where T_s is the sampling time.

9.2.1 Anti-Windup

However, we have to modify (9.4) to avoid a problem called the *reset windup problem.* Reset windup occurs when the output signal takes a long time to reach the reference signal, for instance because of the control input is saturated. During this time the sum in (9.4) becomes very large and causes large overshoots. There exist several *anti-windup* solutions, the one we will use adjusts the sum when the control inputs are saturated. Thus an algorithm is

$$\begin{aligned}
I_n &:= I_{n-1} + K_i T_s e_n \\
v_n &:= K_p e_n + I_n + K_d \tfrac{e_n - e_{n-1}}{T_s} \\
u_n &:= \begin{cases} u_{max}, & if \quad v_n > u_{max} \\ v_n, & if \quad u_{min} < v_n < u_{max} \\ u_{min}, & if \quad v_n < u_{min} \end{cases} \\
I_n &:= I_n + \tfrac{T_s}{T_t}(u_n - v_n)
\end{aligned} \tag{9.5}$$

9.2.2 Simulations

We use a decentralized control structure which results in independent joint control. Thus one PID-regulator to control each actuator. As the output signals we select θ_1 and θ_2 respectively, i.e. the choice (9.1). The choice (9.2) results in large oscillations, since the regulator tries to compensate for the flexibility. In the simulation in this section we use a PD-regulator, thus $K_i \equiv 0$.

Figures 9.1 shows the the simulation result. The regulator tracks the reference good, but the sums $\theta_1 + \theta_3$ and $\theta_2 + \theta_4$ in Figure 9.2 indicate that the result is not so good, the flexibility causes large overshoots. Thus the PID-regulator is

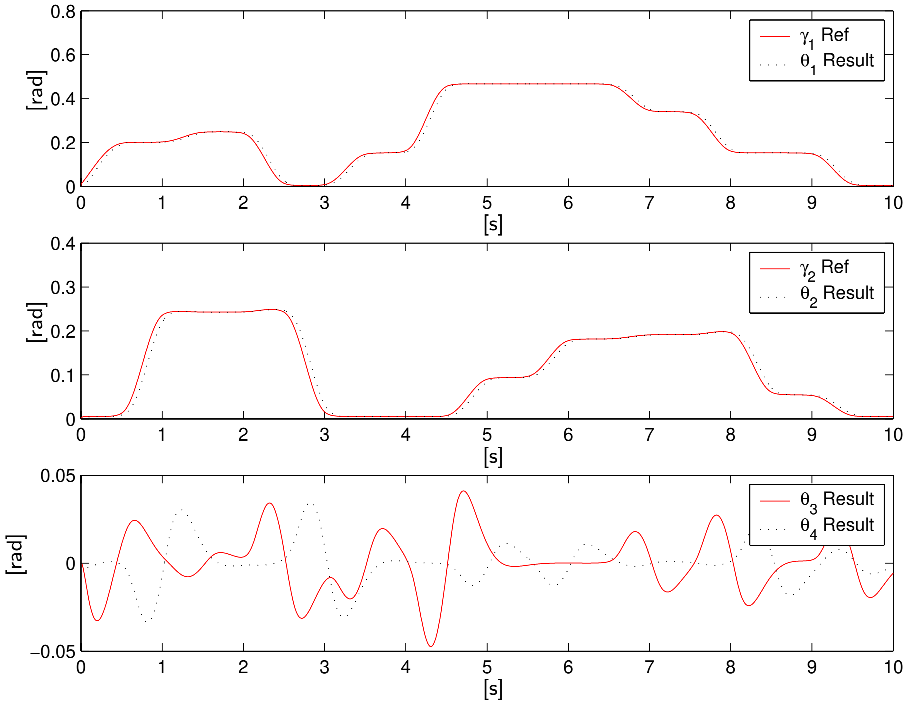


Figure 9.1. Simulation with PD-regulator.

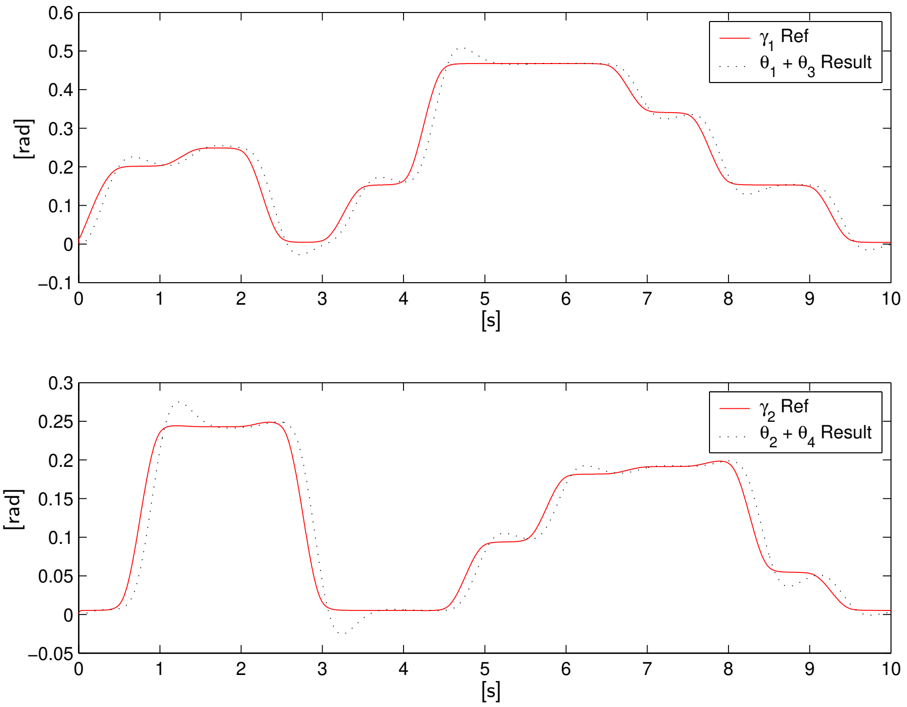


Figure 9.2. Simulation with PD-regulator.

inadequate for control of mechanical structure with flexibility. In the next section we will develop a regulator that can handle the flexibility.

9.3 Linear-Quadratic Control

Consider a linear system on state-space form

$$\begin{aligned} \dot{x} &= Ax + Bu + Nv_1 \\ z &= Mx \\ y &= Cx + v_2 \end{aligned} \tag{9.6}$$

where v_1 and v_2 are white noise. Assume that the reference is $r \equiv 0$. In *Linear Quadratic Control* we calculate the linear feedback $u(t) = F_y y(t)$ that minimize the criterion

$$V = \int (e^T Q_1 e + u^T Q_2 u)dt \tag{9.7}$$

where $e = r - z$ and Q_1 is a positive definite matrix and Q_2 is a positive semidefinite matrix. The optimal linear regulator is then given by [1]

$$u(t) = -L\hat{x}(t) \tag{9.8}$$

[1] If the matrices and noise in (9.6) and (9.7) fulfils some conditions given in [5].

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t)K(y(t) - C\hat{x}(t)) \tag{9.9}$$

where (9.9) is the *kalman filter*. However, in our simulations we assume that all states are measurable. Thus $y = x \Rightarrow \hat{x} = x$ and (9.9) is not needed. The regulator then becomes

$$u(t) = -Lx(t) \tag{9.10}$$

Assume that the number of output signals that we will control equals the number of inputs. In the servo case where r is the reference signal the criterion is

$$V = \int ((r - z)^T Q_1 (r - z) + (u - u^*)^T Q_2 (u - u^*))dt \tag{9.11}$$

where $u^*(t)$ is a constant control signal that gives $z(t) \equiv r$ in stationarity and without noise. The regulator is

$$u(t) = -Lx(t) + L_r r(t) \tag{9.12}$$

where

$$L_r = [M(BL - A)^{-1}B]^{-1} \tag{9.13}$$

ensures that static amplification is 1.

9.3.1 Linearization of the Gimbal Dynamics

The theory in Section 9.3 is valid for linear systems, but our dynamic model is nonlinear. A linear model can be derived by linearization of the nonlinear system. A linearization of

$$\dot{x} = f(x, u) \tag{9.14}$$

in a neigbourhood of the points x_0, u_0 is the system

$$\dot{z} = Az + Bv \tag{9.15}$$

where $z = x - x_0$ and $v = u - u_0$ and the elements of the matrices A and B are

$$\begin{aligned} a_{ij} &= \left.\frac{\partial f_i}{\partial x_j}\right|_{x_0, u_0} \\ b_{ij} &= \left.\frac{\partial f_i}{\partial u_j}\right|_{x_0, u_0} \end{aligned} \tag{9.16}$$

Consider the state-space model of the dynamics in Section 7.3.4 including the actuator dynamics in Section 7.3.5. A first choice of points $\boldsymbol{x}_0$, $\boldsymbol{u}_0$ is

$$\boldsymbol{x}_0 = \begin{bmatrix} \boldsymbol{x}_{\theta_0} \\ \boldsymbol{x}_{\omega_0} \end{bmatrix} = \begin{bmatrix} 0 \\ \theta_{2_0} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{9.17}$$

and

$$\boldsymbol{u}_0 = \begin{bmatrix} V_{1_0} \\ V_{2_0} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9.18}$$

It may be interesting to vary θ_2, since this angle is significant for the damping flexibility. Figures in Appendix E shows the state-space matrices A and B for $\theta_2 = 0$, $\theta_2 = \frac{\pi}{4}$ and $\theta_2 = \frac{\pi}{2}$.

We compare the original nonlinear model with the linear model for $\theta_2 = \frac{\pi}{4}$ in a neigbourhood of $\theta_2 = \frac{\pi}{4}$, Figure 9.3. The result is very satisfying, the model output signals are similar. Also in a neigbourhood of $\theta_2 = 0$ and $\theta_2 = \frac{\pi}{2}$ the model output signals are similar for θ_1, θ_2 and θ_4. Not surprisingly, θ_3 is the variable that differs most, Figure 9.4.

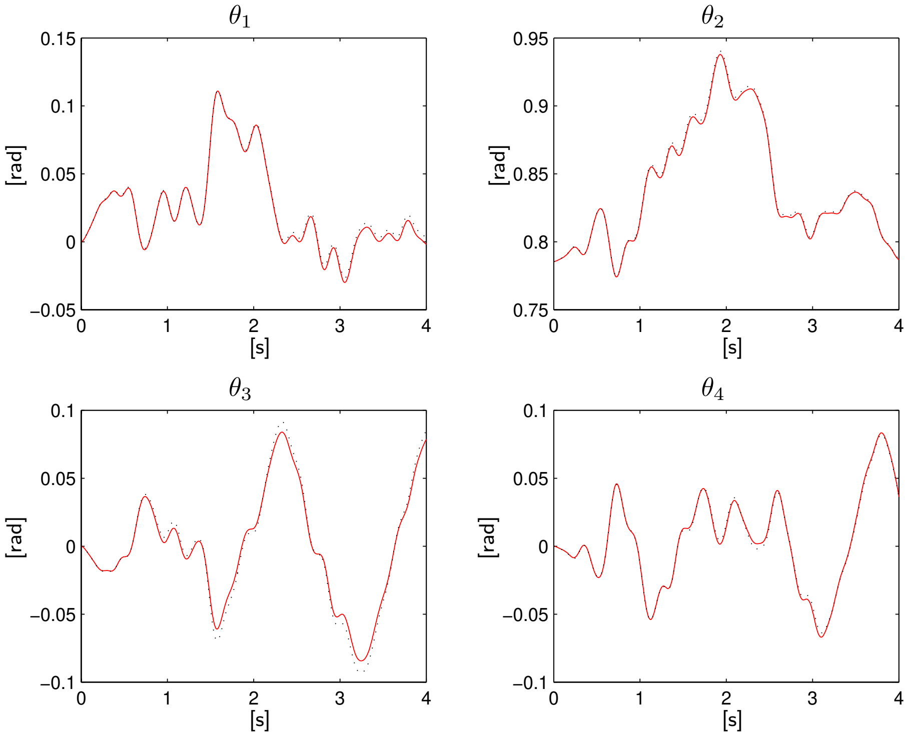


Figure 9.3. Simulated output signals from the original nonlinear system (solid line) and the linear system (dotted line). $\theta_2 \approx \frac{\pi}{4}$

9.3.2 Simulations

It is advantageous to have the same number of output signals as input signals. We want (9.2) as the output signals and therefore it is advantageous to select other

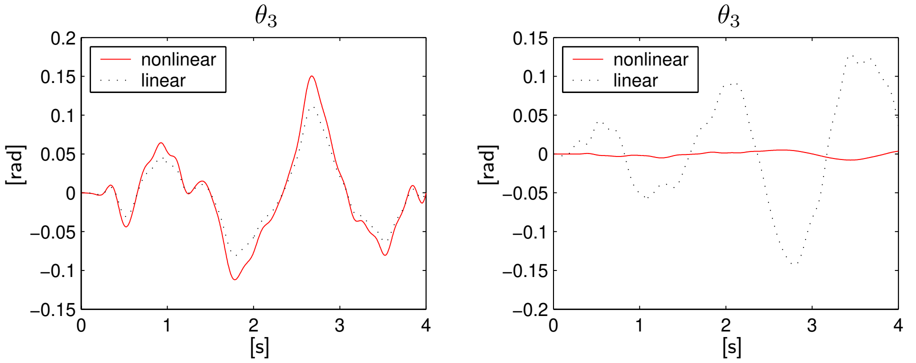


Figure 9.4. Simulated θ_3 output from the original nonlinear system and the linear system. *Left:* $\theta_2 \approx 0$ *Right:* $\theta_2 \approx \frac{\pi}{2}$.

states in our linear state-space description. We select the new states as

$$\boldsymbol{x}_{new} = \begin{bmatrix} \theta_1 + \theta_3 \\ \theta_2 + \theta_4 \\ \theta_3 \\ \theta_4 \\ \omega_1 + \omega_3 \\ \omega_2 + \omega_4 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} x_1 + x_3 \\ x_2 + x_4 \\ x_3 \\ x_4 \\ x_5 + x_7 \\ x_6 + x_8 \\ x_7 \\ x_8 \end{bmatrix} \tag{9.19}$$

or compactly written

$$\boldsymbol{x}_{new} = M\boldsymbol{x} \tag{9.20}$$

where

$$M = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.21}$$

We can now express the new state-space model as

$$\begin{array}{rcl} \dot{\boldsymbol{x}}_{new} & = & AM\boldsymbol{x}_{new} + Bu \\ \boldsymbol{y} & = & CM\boldsymbol{x}_{new} \end{array} \tag{9.22}$$

where A and B is the matrices from the linearization and C is

$$C = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9.23}$$

Next we must define the weighting matrices Q_1 and Q_2. Note that it is the ratios of the elements in the matrices that are significant. For the sake of simplicity, we assume that Q_1 and Q_2 are diagonal, thus the influence of the elements of the matrices are more intuitive, the first elements in Q_1 correspond to the behaviour of the first state x_1, etc.

The design of Q_1 and Q_2 is an iterative procedure; test, change Q_1, Q_2, test, change Q_1, Q_2, and so on until the result is satisfying. It is meaningless to show the exact values of the elements of the weighting matrices, since this is a simulation. Approximately the matrices are

$$Q_1 = diag[a, a, a, a, b, b, b, b] \tag{9.24}$$

$$Q_2 = diag[c, c] \tag{9.25}$$

where $a >> b$. The two first diagonal elements in Q_1 is the output signals and need large elements, but also the third and fourth elements that corresponds to the flexibility are important. The last elements correspond to the velocities and they are relatively small, furthermore the regulator is more sensitive to noise if these elements are to large. Finally c is adjusted so the actuator input signals are realistic and not are unnecessarily saturated.

The simulation results, Figures 9.5 and 9.6 show that the LQ-regulator takes the flexibility into consideration and the output signals $\theta_1 + \theta_3$ and $\theta_2 + \theta_4$ are better than the results with PID-regulator, Figure 9.2.

References & Further Readings

The PID-regulator is well-known and is described in e.g. [4]. The discrete-time PID and different approaches to the reset windup problem are presented in [3].

The LQ-control theory and the conditions that must be fulfilled is given in [5]. LQ-control is used in [18] for control of flexible joints.

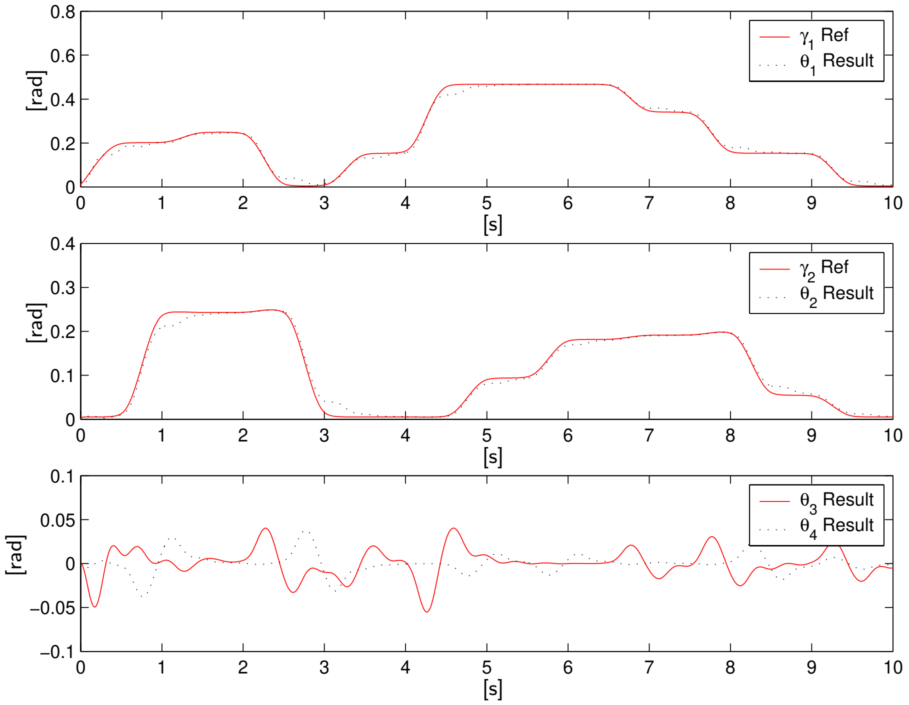
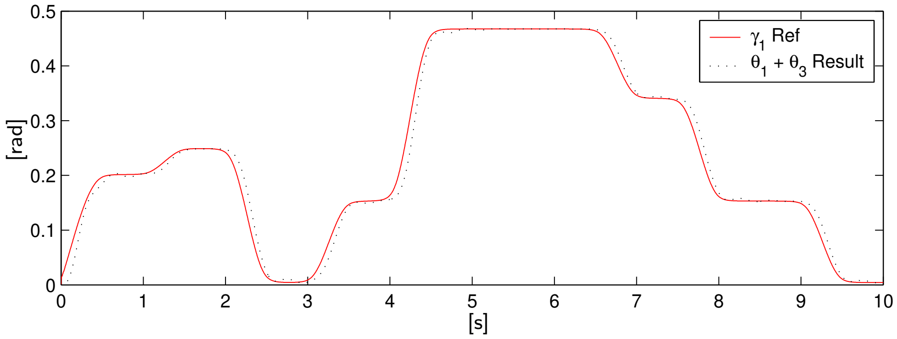


Figure 9.5. Simulations with LQ-regulator.



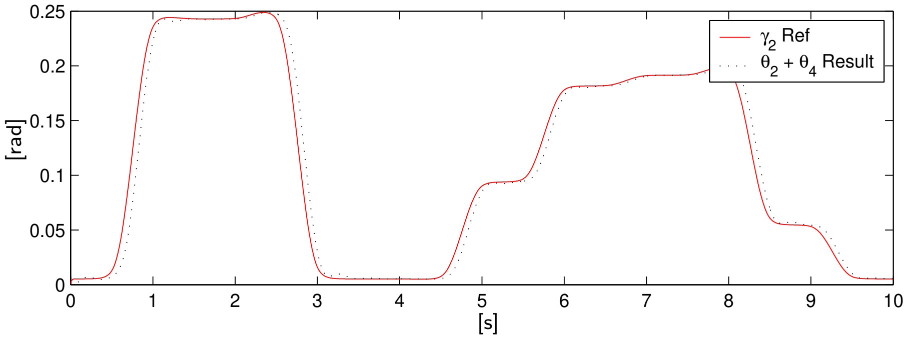


Figure 9.6. Simulations with LQ-regulator.

Chapter 10

Results and Conclusions

In this chapter we summarize the results that are presented in this thesis and we perform some simulations where all different parts are put together. The results are evaluated and suggestions for future work are given.

10.1 Results

The result can be divided into two parts. We have derived models of the gimbal system and we have developed a gimbal control system, based on the gimbal models.

10.1.1 Control System Design

In Chapter 2 a control system design was proposed, Figure 10.1. In this thesis we have focused on the gimbal system, Figure 10.2. The Gimbal Trajectory Planner and the Gimbal Motion Controller have been developed, based on kinematic and dynamic models of the gimbal system.

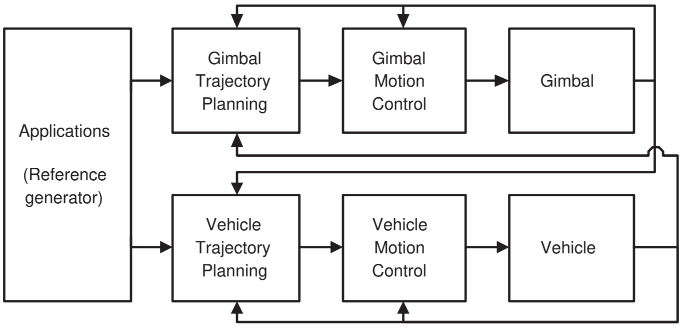


Figure 10.1. The proposed control system design.

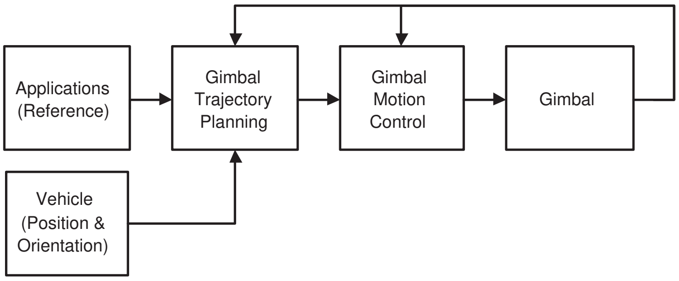


Figure 10.2. The gimbal control system design.

10.1.2 Kinematic Model

In Chapter 3 we derived a kinematic model of the gimbal with two actuated joints and two joints representing the rotational damping flexibility. Especially the fact that the cameras are not placed in the rotational axes have required substantial modelling effort. The kinematic model derivation has been based on techniques from robotics and the description of the model is given in the Denavit-Hartenberg parameters, Table 3.3.

10.1.3 Dynamic Model

In Chapter 7 we used Lagrange equations to derive a dynamic model of the gimbal system

$$D(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + F_v\dot{\boldsymbol{q}} + K(\boldsymbol{q})\boldsymbol{q} = \boldsymbol{\tau_a} \tag{10.1}$$

$D(\boldsymbol{q})$ represent the moment of inertia. $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ represent the centrifugal and the coriolis effects. F_v is the friction in the four joints and the damping flexibility and its constraints were modelled as a nonlinear spring $K(\boldsymbol{q})$. $\boldsymbol{\tau_a}$ are the actuator torques. We ignore the influence of the gravity since the center of mass approximately is not moving.

The dynamic model can be expressed as a nonlinear state-space form

$$\begin{bmatrix} \dot{\boldsymbol{x}}_\theta \\ \dot{\boldsymbol{x}}_\omega \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_\omega \\ D(\boldsymbol{x}_\theta)^{-1}(\boldsymbol{\tau}_a - C(\boldsymbol{x}_\theta, \boldsymbol{x}_\omega)\boldsymbol{x}_\omega - F_v\boldsymbol{x}_\omega - K(\boldsymbol{x}_\theta)\boldsymbol{x}_\theta) \end{bmatrix} \tag{10.2}$$

where $\boldsymbol{x}_\theta$ represents the four joint variables and $\boldsymbol{x}_\omega$ its velocities. The state-space model is used in simulations of the gimbal dynamics.

However, since the measurement equipment for the damping flexibility is not finished when this thesis was written it is difficult to evaluate the dynamic model. The reader should take this into account.

10.1.4 Gimbal Trajectory Planning

First the trajectory planner transforms the input signal to a reference signal relative a vehicle fixed reference frame. Scan mode and tracker mode applications required different solutions. In order to manage singularity and mechanical constraints a path planner has to modify the reference before the inverse kinematic algorithm computes the desired orientation of the body attached to the cameras. The present path planner is a simple mapping function and the inverse kinematic algorithm is based on the inverse of the analytical Jacobian. Finally a filter reduce frequencies that we know excite the damping flexibility.

10.1.5 Gimbal Motion Control

Two different regulator structures were developed. First we proposed a structure with a PID-regulator on each actuated joint. We consider the two joints as the output signals and thus we do not take the measurement of the damping flexibility into consideration. This requires that the reference is filtered so the damping flexibility is not excited too much.

The second proposal was a linear quadratic control (LQ-control). Since the control design requires a linear model of the system, a linearization of the nonlinear dynamic model was performed. The sums $\theta_1 + \theta_3$ and $\theta_2 + \theta_4$ are considered as outputs and since θ_3 and θ_4 are states we take the damping flexibility into consideration.

10.2 Simulations

The simulations were performed in MATLAB and Simulink, Figure 10.3 shows the Simulink-model of the gimbal system. In simulation 1 the reference signals are coordinates, relative the gimbal, which we want the sensors to point at. Figure 10.4 shows the orientation reference and the orientation result. Note that it is the true orientation result, not the sums in (9.2). Figure 10.5 shows the reference coordinates and the coordinates that are calculated from the orientation and the result of d_{10}, the distance between the camera and the end-effector, from the inverse kinematic algorithm.

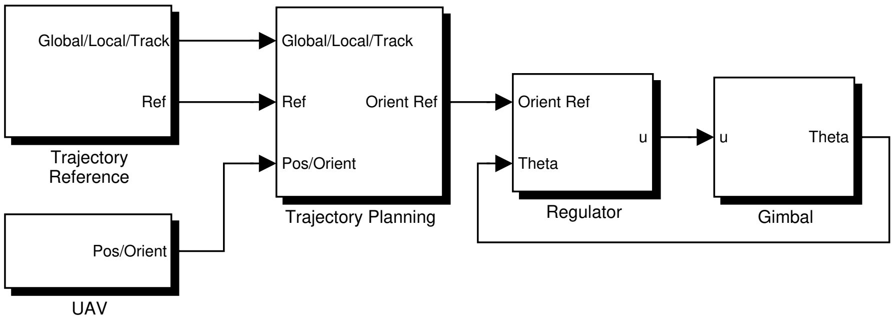


Figure 10.3. Simulink model.

In simulation 2 the reference signals are coordinates relative a global reference frame which we want the gimbal to point at. In this simulation the reference is $[0 \quad 0 \quad 0]^T$. We also define the motion of the UAV. In this simulation we let the UAV position describe a circle 50 m above the global frame. The radius of the circle is 50 m. We also include sinuous UAV orientation motion in two directions with amplitude 0.3 rad and 0.2 rad respectively.

The global reference signal is transformed to coordinates relative the gimbal, Figure 10.7. Figure 10.6 shows the orientation reference and the orientation result and finally 10.8 shows global coordinate reference and the result.

10.3 Experiments

The measurement equipment for the damping flexibility is not finished, but the pan and tilt angles can be measured by digital encoders. Since a MATLAB interface has been developed, the dynamic model of the gimbal system in the Simulink model can be exchanged with a block that receives and sends signals to the physical system. The PID regulator was used and the flexibility was set to $\theta_3 \equiv \theta_4 \equiv 0$. Thus the flexibility was ignored, but it was possible to evaluate parts of the trajectory planner block. The tests were satisfying.

10.4 Conclusions & Future Work

Since the work presented in this thesis is a pilot study, there is much left to do. All parts of the design must be tested and optimized thoroughly. Furthermore, the final design of the trajectory planner and motion controller have to be implemented.

The structure of the kinematic model is satisfying, but the Denavit-Hartenberg parameters, especially θ_8 and θ_9, should be measured or estimated more exactly, maybe with some kinematic parameter calibration method that is used in robotics.

The dynamic model must be evaluated and the dynamic parameters must be identified based on data from the physical gimbal system. Possibly the dynamic model requires some modifications, e.g. a better friction model. For instance static friction and nonlinear viscous friction can be added, the latter to model the constraints of the damping flexibility better.

The trajectory planner must be further developed and tested, especially the path planning problem near the singularity. Probably a satisfying solution is based on an estimator that can predict the target position. Maybe the tracker and the planner can interact.

The transformation methods that transform an error from a tracker to a reference position relative a vehicle fixed frame must be tested further. Is Method 1 robust enough or is Method 2 sufficient? The switching between different applications must also be performed in a smooth and robust way.

When the dynamic model is complete, including the dynamic parameters, the LQ-controller can be tested on the physical system. Hopefully the LQ-controller is satisfying, otherwise one might have to consider gain-scheduling or some nonlinear or feedforward control strategies.

A large and complex problem that was mentioned in Chapter 2 is the trajectory planning and motion control of the vehicle. A first step would be to include an aerospace block in the Simulink simulations in order to get more realistic behavior. A challenge is to perform effective interaction of the vehicle planner and the gimbal planner. Eventually, another question is if the motion of the vehicle affects the gimbal performance, and in what way?

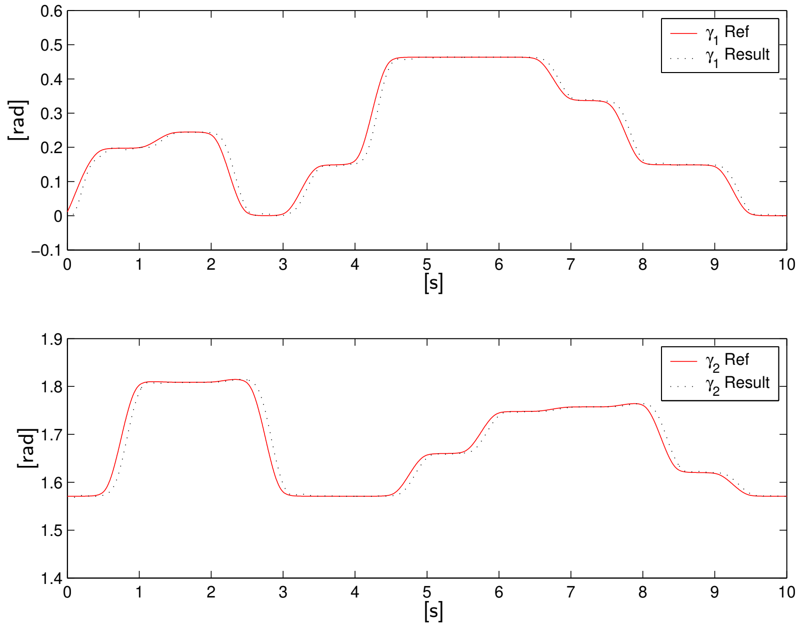


Figure 10.4. Simulation 1. Inner gimbal orientation reference and result.

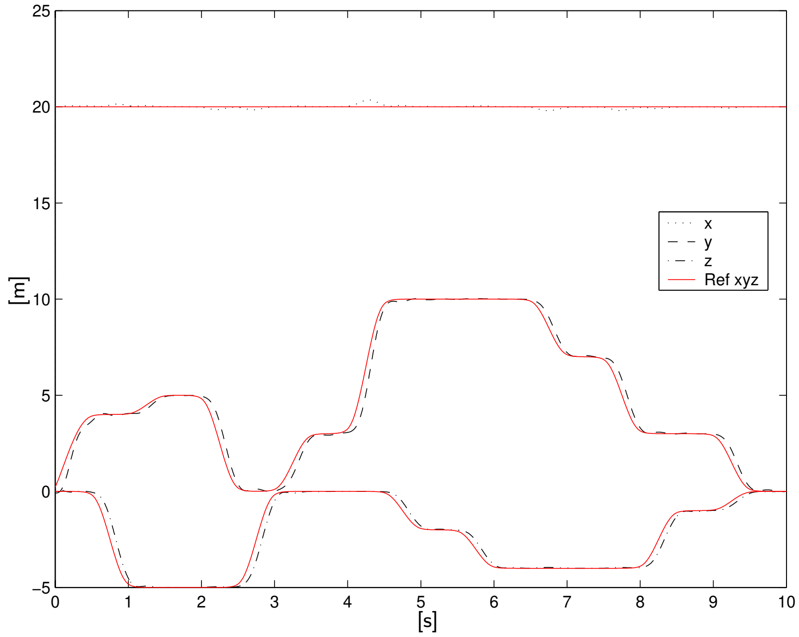


Figure 10.5. Simulation 1. Local coordinate (relative a vehicle fixed reference frame) reference and result.

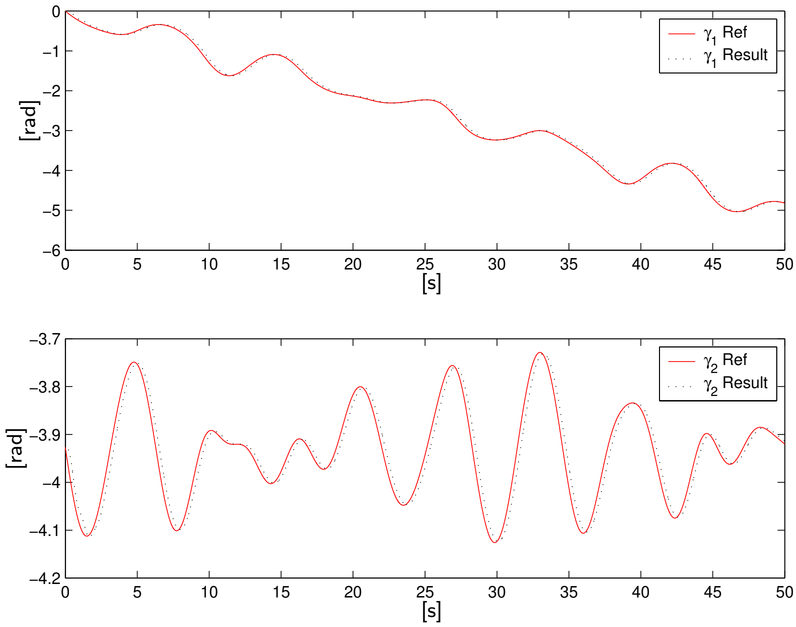


Figure 10.6. Simulation 2. Inner gimbal orientation reference and result.

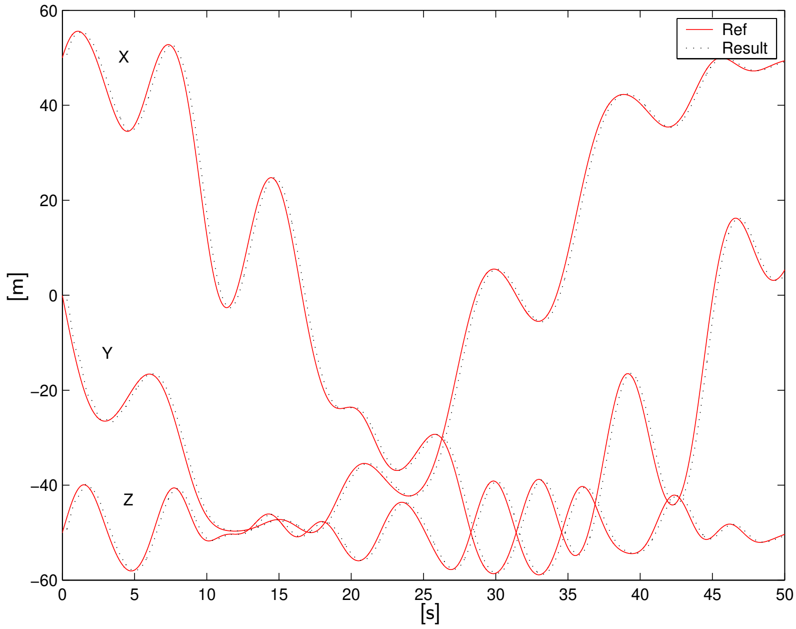


Figure 10.7. Simulation 2. Local coordinate (relative a vehicle fixed reference frame) reference and result.

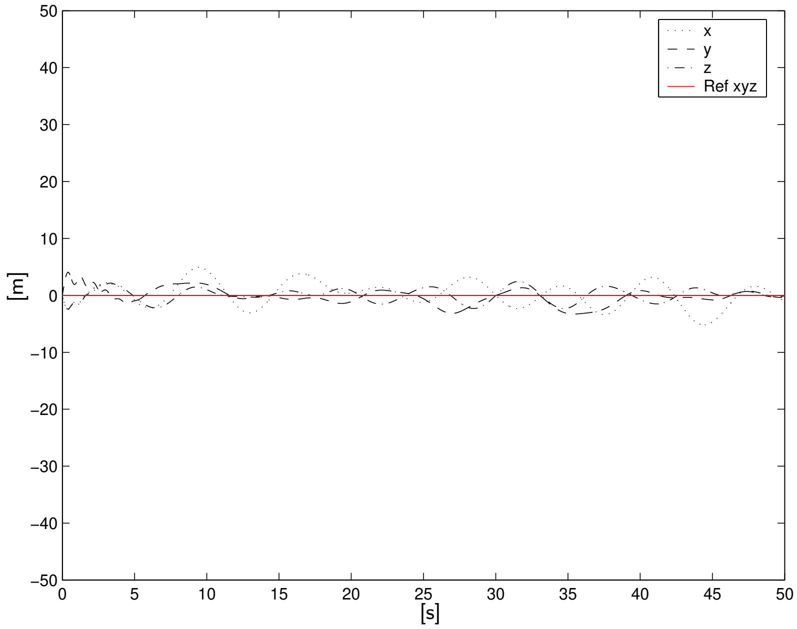


Figure 10.8. Simulation 2. Global coordinate (relative an earth fixed reference frame) reference and result.

Bibliography

[1] P.I. Corke. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3(1):24–32, March 1996.

[2] C. Canudas de Wit, B. Siciliano, and G. Bastin. *Theory of Robot Control.* Springer, 1996.

[3] T. Glad, S. Gunnarsson, L. Ljung, T. McKelvey, and A. Stenman. Digital styrning, August 2000. Compendium in Swedish, Department of Electrical Engineering, Linköping University, Sweden.

[4] T. Glad and L. Ljung. *Reglerteknik. Grundläggande teori.* Studentlitteratur, Lund, Sweden, 2nd edition, 1989. In Swedish.

[5] T. Glad and L. Ljung. *Reglerteori. Flervariabla och olinjära metoder.* Studentlitteratur, Lund, Sweden, 1997. In Swedish.

[6] H. Goldstein. *Classical Mechanics.* Addison-Wesley Publishing Company, Inc, 2nd edition, 1980.

[7] H. Karlsson. Efficient Region Tracking and Target Position Estimation in Images Sequence using Kalman Filters. Master's thesis LiTH-IKP-EX-1936, Department of Mechanical Engineering, Linköping University, Linköping, Sweden, May 2002.

[8] H. Karlsson and J. Nygårds. A Kalman Filter Efficient Region Tracking in Image Sequences using the Affine model. In *Proceedings of the SSAB 02 Symposium on Images Analysis*, pages 89–94, Lund, Sweden, March 2002. SSAB.

[9] P. Knuts and T. Thörn. Robust Effective Target Tracking in Images Sequences. Master's thesis LiTH-ISY-EX-3155, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2001.

[10] K. Kozlowski. *Modelling and Identification in Robotics.* Springer, 1998.

[11] L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, Englewood Cliffs, N.J. USA, 1987.

[12] L. Ljung. *System Identification Toolbox – User's Guide.* The MathWorks, Inc., Sherborn, MA. USA, 1991.

[13] L. Ljung and T. Glad. *Modellbygge och simulering.* Studentlitteratur, Lund, Sweden, 1991. In Swedish.

[14] C. Nordling and J. Österman. *Physics Handbook for Science and Engineering.* Studentlitteratur, 5th edition, 1996.

[15] M. Norrlöf. Modeling of industrial robots. Technical Report LiTH-ISY-R-2208, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1999.

[16] J. Nygårds, T. Högström, and M. Ulvklo. The QWIP/MASP system. An airborne experiment platform with georeferenced EO/IR sensors for research on image processing in UAV applications. Technical report, Swedish Defence Research Agency, Linköping, Sweden, 2002.

[17] L. Sciavicco and B. Siciliano. *Modeling and control of robot manipulators.* The McGraw-Hill Companies, Inc, 1996.

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

Appendix A

Definitions

A.1 Skew Symmetric Matrices

Definition A.1 *A matrix S is said to be* skew symmetric *if and only if*

$$S^T + S = 0 \tag{A.1}$$

It can be shown that a 3×3 skew-symmetric matrix only contains three independent entries and has the form

$$S = \begin{bmatrix} 0 & -s_1 & s_2 \\ s_1 & 0 & -s_3 \\ -s_2 & s_3 & 0 \end{bmatrix} \tag{A.2}$$

Let $\boldsymbol{a} = [a_x \quad a_y \quad a_z]^T$ and then we define the skew symmetric matrix $S(\boldsymbol{a})$ as

$$S(\boldsymbol{a}) = \begin{bmatrix} 0 & -a_1 & a_2 \\ a_1 & 0 & -a_3 \\ -a_2 & a_3 & 0 \end{bmatrix} \tag{A.3}$$

Some important properties of $S(\boldsymbol{a})$ is

$$S(\alpha\boldsymbol{a} + \beta\boldsymbol{b}) = \alpha S(\boldsymbol{a}) + \beta S(\boldsymbol{b}) \tag{A.4}$$

and

$$S(\boldsymbol{a})\boldsymbol{p} = \boldsymbol{a} \times \boldsymbol{p} \tag{A.5}$$

where $\boldsymbol{p} = [p_x \quad p_y \quad p_z]^T$ is a arbitrary vector and × denotes the vector cross product defined in [18].

A.2 Trace

Definition A.2 *The* trace *of a square matrix A is defined as the sum of the diagonal elements and is denoted as $tr[A]$. Thus*

$$tr[A] = \sum_{i=1}^{n} a_{ii} \tag{A.6}$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \tag{A.7}$$

It is easy to show that

$$tr[AB] = tr[BA] \tag{A.8}$$

for any two matrices A and B and furthermore

$$\boldsymbol{a}^T \boldsymbol{b} = tr[\boldsymbol{a}\boldsymbol{b}^T] \tag{A.9}$$

for any two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

Appendix B

T_0^{10}

$$T_0^{10} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where

$$\begin{array}{rcl}
T_{11} & = & -s_{\theta_1} s_{\theta_3} s_{\theta_4} s_{\theta_8} + c_{\theta_2}((-c_{\theta_2})c_{\theta_8} s_{\theta_3} + c_{\theta_4} s_{\theta_2} s_{\theta_8}) + c_{\theta_3}((-c_{\theta_8})s_{\theta_1} + c_{\theta_2} c_{\theta_2} s_{\theta_4} s_{\theta_8}) \\
 & & \\
T_{12} & = & -s_{\theta_1}(c_{\theta_8} c_{\theta_9} s_{\theta_3} s_{\theta_4} - c_{\theta_3} c_{\theta_9} s_{\theta_8} + c_{\theta_4} s_{\theta_3} s_{\theta_9}) + c_{\theta_2}(c_{\theta_2} c_{\theta_9}(c_{\theta_3} c_{\theta_8} s_{\theta_4} + s_{\theta_3} s_{\theta_8}) - \\
 & - & s_{\theta_2} s_{\theta_4} s_{\theta_9} + c_{\theta_4}(c_{\theta_8} c_{\theta_9} s_{\theta_2} + c_{\theta_2} c_{\theta_3} s_{\theta_9})) \\
 & & \\
T_{13} & = & -s_{\theta_1}(c_{\theta_4} c_{\theta_9} s_{\theta_3} + ((-c_{\theta_8})s_{\theta_3} s_{\theta_4} + c_{\theta_3} s_{\theta_8})s_{\theta_9}) - c_{\theta_2}(s_{\theta_2}(c_{\theta_9} s_{\theta_4} + c_{\theta_4} c_{\theta_8} s_{\theta_9}) + \\
 & + & c_{\theta_2}((-c_{\theta_3})c_{\theta_4} c_{\theta_9} + c_{\theta_3} c_{\theta_8} s_{\theta_4} s_{\theta_9} + s_{\theta_3} s_{\theta_8} s_{\theta_9})) \\
 & & \\
T_{14} & = & -s_{\theta_1}(c_{\theta_3}(d_7 + d_{10} s_{\theta_8} s_{\theta_9}) + s_{\theta_3}(c_{\theta_4}(d_6 + d_{10} c_{\theta_9}) + s_{\theta_4}(d_5 - d_{10} c_{\theta_8} s_{\theta_9}))) + \\
 & + & c_{\theta_2}(s_{\theta_2}((-(d_6 + d_{10} c_{\theta_9}))s_{\theta_4} + c_{\theta_4}(d_5 - d_{10} c_{\theta_8} s_{\theta_9})) + c_{\theta_2}((-s_{\theta_3})(d_7 + d_{10} s_{\theta_8} s_{\theta_9}) + \\
 & + & c_{\theta_3}(c_{\theta_4}(d_6 + d_{10} c_{\theta_9}) + s_{\theta_4}(d_5 - d_{10} c_{\theta_8} s_{\theta_9})))) \\
 & & \\
T_{21} & = & c_{\theta_2}(c_{\theta_3} c_{\theta_8} + s_{\theta_3} s_{\theta_4} s_{\theta_8}) + s_{\theta_1}(c_{\theta_4} s_{\theta_2} s_{\theta_8} + c_{\theta_2}((-c_{\theta_8})s_{\theta_3} + c_{\theta_3} s_{\theta_4} s_{\theta_8})) \\
 & & \\
T_{22} & = & c_{\theta_9}(c_{\theta_4} c_{\theta_8} s_{\theta_1} s_{\theta_2} + c_{\theta_2}(c_{\theta_8} s_{\theta_3} s_{\theta_4} - c_{\theta_3} s_{\theta_8}) + c_{\theta_2} s_{\theta_1}(c_{\theta_3} c_{\theta_8} s_{\theta_4} + s_{\theta_3} s_{\theta_8})) + \\
 & + & (c_{\theta_2} c_{\theta_3} c_{\theta_4} s_{\theta_1} + c_{\theta_2} c_{\theta_4} s_{\theta_3} - s_{\theta_1} s_{\theta_2} s_{\theta_4})s_{\theta_9} \\
 & & \\
T_{23} & = & c_{\theta_9}(c_{\theta_2} c_{\theta_3} c_{\theta_4} s_{\theta_1} + c_{\theta_2} c_{\theta_4} s_{\theta_3} - s_{\theta_1} s_{\theta_2} s_{\theta_4}) - (c_{\theta_4} c_{\theta_8} s_{\theta_1} s_{\theta_2} + c_{\theta_2}(c_{\theta_8} s_{\theta_3} s_{\theta_4} - \\
 & - & c_{\theta_3} s_{\theta_8}) + c_{\theta_2} s_{\theta_1}(c_{\theta_3} c_{\theta_8} s_{\theta_4} + s_{\theta_3} s_{\theta_8}))s_{\theta_9} \\
 & & \\
T_{24} & = & c_{\theta_2}(c_{\theta_3}(d_7 + d_{10} s_{\theta_8} s_{\theta_9}) + s_{\theta_3}(c_{\theta_4}(d_6 + d_{10} c_{\theta_9}) + s_{\theta_4}(d_5 - d_{10} c_{\theta_8} s_{\theta_9}))) + \\
 & + & s_{\theta_1}(s_{\theta_2}((-(d_6 + d_{10} c_{\theta_9}))s_{\theta_4} + c_{\theta_4}(d_5 - d_{10} c_{\theta_8} s_{\theta_9})) + c_{\theta_2}((-s_{\theta_3})(d_7 + d_{10} s_{\theta_8} s_{\theta_9}) + \\
 & + & c_{\theta_3}(c_{\theta_4}(d_6 + d_{10} c_{\theta_9}) + s_{\theta_4}(d_5 - d_{10} c_{\theta_8} s_{\theta_9})))) \\
 & & \\
T_{31} & = & c_{\theta_8} s_{\theta_2} s_{\theta_3} + (c_{\theta_2} c_{\theta_4} - c_{\theta_3} s_{\theta_2} s_{\theta_4})s_{\theta_8} \\
 & & \\
T_{32} & = & c_{\theta_2}(c_{\theta_4} c_{\theta_8} c_{\theta_9} - s_{\theta_4} s_{\theta_9}) - s_{\theta_2}(c_{\theta_9} s_{\theta_3} s_{\theta_8} + c_{\theta_3}(c_{\theta_8} c_{\theta_9} s_{\theta_4} + c_{\theta_4} s_{\theta_9})) \\
 & & \\
T_{33} & = & s_{\theta_2} s_{\theta_3} s_{\theta_8} s_{\theta_9} - c_{\theta_2}(c_{\theta_9} s_{\theta_4} + c_{\theta_4} c_{\theta_8} s_{\theta_9}) + c_{\theta_3} s_{\theta_2}((-c_{\theta_4})c_{\theta_9} + c_{\theta_8} s_{\theta_4} s_{\theta_9}) \\
 & & \\
T_{34} & = & d_1 + d_7 s_{\theta_2} s_{\theta_3} + d_{10} s_{\theta_2} s_{\theta_3} s_{\theta_8} s_{\theta_9} + c_{\theta_2}((-(d_6 + d_{10} c_{\theta_9}))s_{\theta_4} + c_{\theta_4}(d_5 - d_{10} c_{\theta_8} s_{\theta_9})) - \\
 & - & c_{\theta_3} s_{\theta_2}(c_{\theta_4}(d_6 + d_{10} c_{\theta_9}) + s_{\theta_4}(d_5 - d_{10} c_{\theta_8} s_{\theta_9}))
\end{array}$$

Appendix C

Path Planning Results

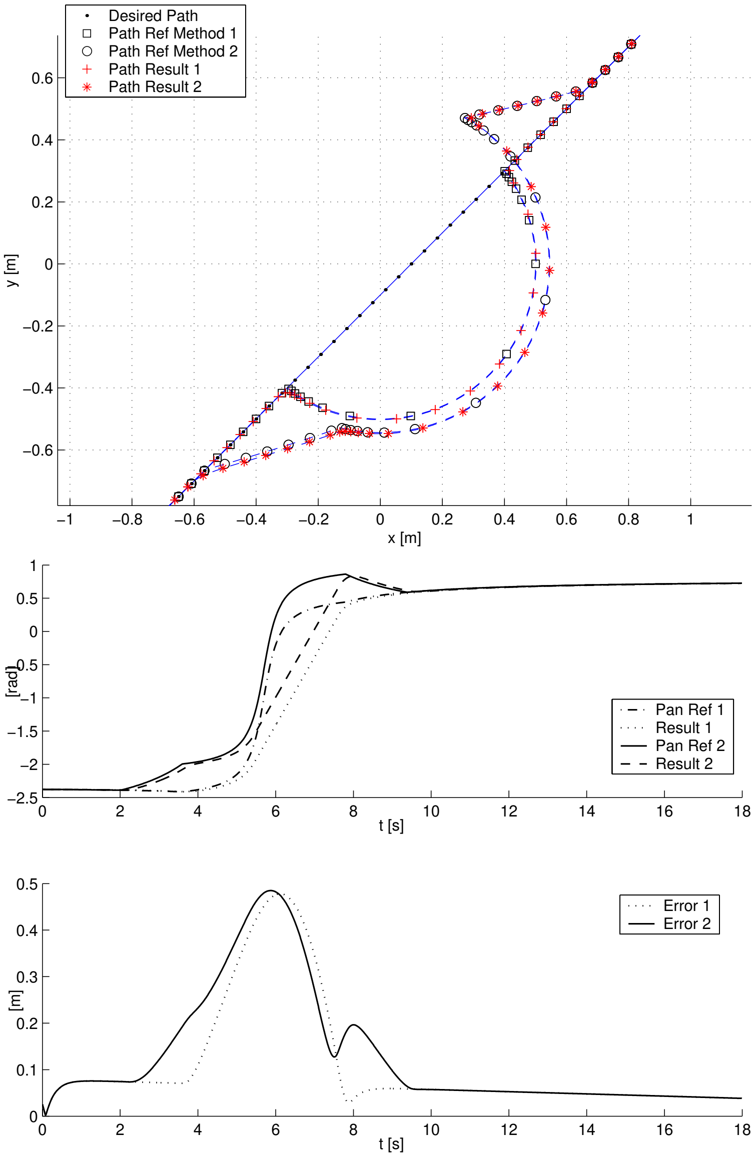


Figure C.1. Simulation 1. Target velocity ≈ 0.25. $R_1 = R_2 = 0.5$. *Top:* XY references and output results. *Middle:* Pan angle references and output results. *Bottom:* XY error between image view center and target.

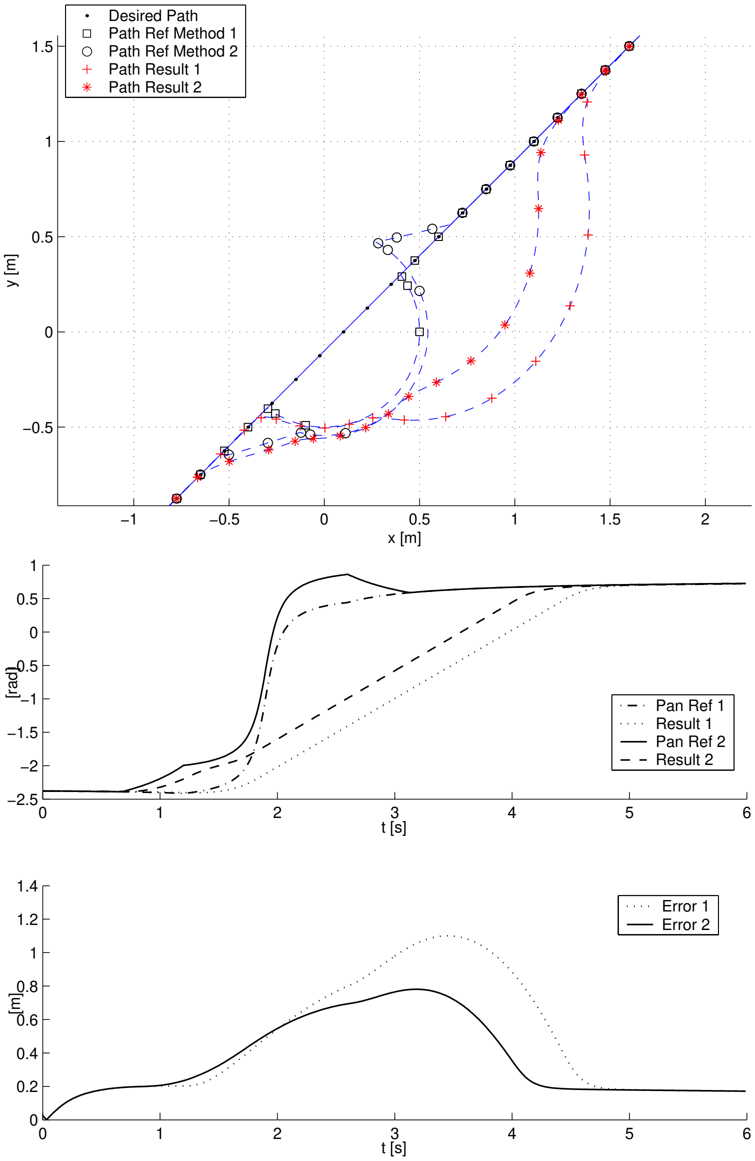


Figure C.2. Simulation 1. Target velocity ≈ 0.75. $R_1 = R_2 = 0.5$. *Top:* XY references and output results. *Middle:* Pan angle references and output results. *Bottom:* XY error between image view center and target.

Appendix D

$C(\theta, \dot{\theta})$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{bmatrix}$$

where

$$\begin{array}{rcl}
C_{11} & = & (1/2)(2I_{4_{11}}\theta_4\omega_4 c_{\theta_2}^2 + 2(I_{4_{11}} - I_{4_{33}})\theta_4\omega_2\cos(2\theta_2) + 2s_{\theta_2}((I_{2_{11}}\omega_2 + \\
 & + & I_{4_{22}}\theta_3^2\omega_2 + I_{4_{33}}\theta_4^2\omega_2 - I_{4_{33}}\omega_4 + I_{4_{11}}(\omega_2 + \omega_4))c_{\theta_2} + (I_{4_{22}}\theta_3\omega_3 + \\
 & + & I_{4_{33}}\theta_4\omega_4)s_{\theta_2}) - (I_{2_{33}} + I_{4_{33}} + I_{4_{11}}\theta_4^2)\omega_2\sin(2\theta_2))
\end{array}$$

$$\begin{array}{rcl}
C_{12} & = & (1/2)(\theta_3((-I_{4_{11}})(2\omega_2 + \omega_4) + I_{4_{33}}(-2\theta_4^2\omega_2 + \omega_4) + I_{4_{22}}(2\omega_2 + \omega_4 + \theta_3^2\omega_4))c_{\theta_2} + \\
 & + & 2(I_{4_{11}} - I_{4_{33}})\theta_4\omega_1\cos(2\theta_2) - (2I_{4_{33}}\theta_3\theta_4\omega_2 - I_{4_{22}}\omega_3 + I_{4_{33}}\omega_3 + I_{4_{33}}\theta_4^2\omega_3 + \\
 & + & I_{4_{11}}(-2\theta_3\theta_4\omega_2 + \omega_3 + \theta_4^2\omega_3) + 2I_{4_{33}}\theta_3\theta_4\omega_4 - 2(I_{2_{11}} - I_{2_{33}} + I_{4_{11}} - I_{4_{33}} + \\
 & + & I_{4_{22}}\theta_3^2 - I_{4_{11}}\theta_4^2 + I_{4_{33}}\theta_4^2)\omega_1 c_{\theta_2})s_{\theta_2})
\end{array}$$

$$\begin{array}{rcl}
C_{13} & = & (1/2)(2I_{4_{11}}\theta_4\omega_4 c_{\theta_2} + s_{\theta_2}(I_{4_{22}}\omega_2 - I_{4_{33}}\omega_2 - I_{4_{33}}\theta_4^2\omega_2 - I_{4_{11}}(\omega_2 + \\
 & + & \theta_4^2\omega_2 - \omega_4) + I_{4_{22}}\omega_4 - I_{4_{33}}\omega_4 + 3I_{4_{22}}\theta_3^2\omega_4 + 2I_{4_{22}}\theta_3\omega_1 s_{\theta_2}))
\end{array}$$

$$\begin{array}{rcl}
C_{14} & = & (1/2)((\theta_3(I_{4_{22}} + I_{4_{33}} + I_{4_{22}}\theta_3^2)\omega_2 + I_{4_{11}}((-\theta_3)\omega_2 + 2\theta_4\omega_3))c_{\theta_2} + \\
 & + & 2I_{4_{11}}\theta_4\omega_1 c_{\theta_2}^2 + s_{\theta_2}(-2I_{4_{33}}\theta_3\theta_4\omega_2 + I_{4_{11}}\omega_3 + I_{4_{22}}\omega_3 - I_{4_{33}}\omega_3 + \\
 & + & 3I_{4_{22}}\theta_3^2\omega_3 + 2(I_{4_{11}} - I_{4_{33}})\omega_1 c_{\theta_2} + 2I_{4_{33}}\theta_4\omega_1 s_{\theta_2}))
\end{array}$$

$$\begin{array}{rcl}
C_{21} & = & (1/2)(((-I_{4_{22}})\theta_3(1 + \theta_3^2)\omega_4 - I_{4_{11}}(2\theta_4\omega_3 + \theta_3\omega_4) + I_{4_{33}}(2\theta_4\omega_3 + \theta_3\omega_4))c_{\theta_2} - \\
 & - & 2(I_{4_{11}} - I_{4_{33}})\theta_4\omega_1\cos(2\theta_2) + (I_{4_{22}}\omega_3 + I_{4_{33}}\omega_3 - I_{4_{33}}\theta_4^2\omega_3 + I_{4_{11}}(-1 + \theta_4^2)\omega_3 - \\
 & - & 2I_{4_{33}}\theta_3\theta_4\omega_4 - 2(I_{2_{11}} - I_{2_{33}} + I_{4_{11}} - I_{4_{33}} + I_{4_{22}}\theta_3^2 - I_{4_{11}}\theta_4^2 + \\
 & + & I_{4_{33}}\theta_4^2)\omega_1 c_{\theta_2})s_{\theta_2})
\end{array}$$

$$C_{22} = \theta_3(I_{4_{11}}\omega_3 + I_{4_{33}}\theta_4(\theta_4\omega_3 + \theta_3\omega_4))$$

$$\begin{array}{rcl}
C_{23} & = & \theta_3(I_{4_{11}} + I_{4_{33}}\theta_4^2)\omega_2 + (-I_{4_{11}} + I_{4_{33}})\theta_4\omega_3 + (1/2)(-I_{4_{11}} + 2I_{4_{22}} + I_{4_{33}})\theta_3\omega_4 + \\
 & + & (1/2)\omega_1(-2(I_{4_{11}} - I_{4_{33}})\theta_4 c_{\theta_2} + (I_{4_{22}} + I_{4_{33}} - I_{4_{33}}\theta_4^2 + I_{4_{11}}(-1 + \theta_4^2))s_{\theta_2})
\end{array}$$

$$\begin{array}{rcl}
C_{24} & = & (-(1/2))\theta_3(-2I_{4_{33}}\theta_3\theta_4\omega_2 + I_{4_{11}}\omega_3 - 2I_{4_{22}}\omega_3 - I_{4_{33}}\omega_3 + (I_{4_{11}} + I_{4_{22}} - I_{4_{33}} + \\
 & + & I_{4_{22}}\theta_3^2)\omega_1 c_{\theta_2} + 2I_{4_{33}}\theta_4\omega_1 s_{\theta_2})
\end{array}$$

$$\begin{array}{rcl}
C_{31} & = & (1/2)(2\theta_4((-I_{4_{33}})\omega_2 + I_{4_{11}}(\omega_2 + \omega_4))c_{\theta_2} - s_{\theta_2}((-I_{4_{11}})\omega_2 + I_{4_{22}}\omega_2 + \\
 & + & I_{4_{33}}\omega_2 + I_{4_{11}}\theta_4^2\omega_2 - I_{4_{33}}\theta_4^2\omega_2 - I_{4_{11}}\omega_4 + I_{4_{22}}\omega_4 + I_{4_{33}}\omega_4 + \\
 & + & 3I_{4_{22}}\theta_3^2\omega_4 + 2I_{4_{22}}\theta_3\omega_1 s_{\theta_2}))
\end{array}$$

$$\begin{array}{rcl}
C_{32} & = & (1/2)((-\theta_3)(2I_{4_{33}}\theta_4^2\omega_2 + 2I_{4_{22}}\omega_4 - I_{4_{33}}\omega_4 + I_{4_{11}}(2\omega_2 + \omega_4)) + 2(I_{4_{11}} - \\
 & - & I_{4_{33}})\theta_4\omega_1 c_{\theta_2} - (I_{4_{22}} + I_{4_{33}} - I_{4_{33}}\theta_4^2 + I_{4_{11}}(-1 + \theta_4^2))\omega_1 s_{\theta_2})
\end{array}$$

$$\begin{array}{rcl}
C_{33} & = & I_{4_{11}}\theta_4\omega_4
\end{array}$$

$$\begin{array}{rcl}
C_{34} & = & (1/2)((-I_{4_{11}})\theta_3\omega_2 - 2I_{4_{22}}\theta_3\omega_2 + I_{4_{33}}\theta_3\omega_2 + 2I_{4_{11}}\theta_4\omega_3 - 4I_{4_{22}}\theta_3\omega_4 - \\
 & - & 4I_{4_{22}}\theta_3^3\omega_4 + 2I_{4_{11}}\theta_4\omega_1 c_{\theta_2} + (I_{4_{11}} - I_{4_{22}} - I_{4_{33}} - 3I_{4_{22}}\theta_3^2)\omega_1 s_{\theta_2})
\end{array}$$

$$\begin{array}{rcl}
C_{41} & = & (1/2)(-2I_{4_{11}}\theta_4\omega_1 c_{\theta_2}^2 + c_{\theta_2}(\theta_3(I_{4_{22}} - I_{4_{33}} + I_{4_{22}}\theta_3^2)\omega_2 + \\
 & + & I_{4_{11}}(\theta_3\omega_2 - 2\theta_4\omega_3) - 2(I_{4_{11}} - I_{4_{33}})\omega_1 s_{\theta_2}) + s_{\theta_2}((-I_{4_{11}} + I_{4_{22}} + \\
 & + & 3I_{4_{22}}\theta_3^2)\omega_3 + I_{4_{33}}(2\theta_3\theta_4\omega_2 + \omega_3) - 2I_{4_{33}}\theta_4\omega_1 s_{\theta_2}))
\end{array}$$

$$\begin{array}{rcl}
C_{42} & = & (1/2)\theta_3(-2I_{4_{33}}\theta_3\theta_4\omega_2 + I_{4_{11}}\omega_3 + 2I_{4_{22}}\omega_3 - I_{4_{33}}\omega_3 + (I_{4_{11}} + I_{4_{22}} - I_{4_{33}} + \\
 & + & I_{4_{22}}\theta_3^2)\omega_1 c_{\theta_2} + 2I_{4_{33}}\theta_4\omega_1 s_{\theta_2})
\end{array}$$

$$\begin{array}{rcl}
C_{43} & = & (1/2)(I_{4_{11}}\theta_3\omega_2 + 2I_{4_{22}}\theta_3\omega_2 - I_{4_{33}}\theta_3\omega_2 - 2I_{4_{11}}\theta_4\omega_3 + 4I_{4_{22}}\theta_3\omega_4 + \\
 & + & 4I_{4_{22}}\theta_3^3\omega_4 - 2I_{4_{11}}\theta_4\omega_1 c_{\theta_2} + (-I_{4_{11}} + I_{4_{22}} + I_{4_{33}} + 3I_{4_{22}}\theta_3^2)\omega_1 s_{\theta_2})
\end{array}$$

$$\begin{array}{rcl}
C_{44} & = & 2I_{4_{22}}\theta_3(1 + \theta_3^2)\omega_3
\end{array}$$

Appendix E

Linearization Results

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & \frac{\text{Kb1Km1 b1R1}}{\text{I122 I211 I411 R1}} & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\text{k4}}{\text{I222}} & 0 & \frac{\text{Kb2Km2 b2R2}}{\text{I222R2}} & 0 & \frac{\text{b4}}{\text{I222}} \\
0 & 0 & \frac{\text{k3}}{\text{I433}} & 0 & 0 & 0 & \frac{\text{b3}}{\text{I433}} & 0 \\
0 & 0 & 0 & \frac{\text{I222 I422 k4}}{\text{I222I422}} & 0 & \frac{\text{Kb2Km2 b2R2}}{\text{I222R2}} & 0 & \frac{\text{b4 I222 I422}}{\text{I222I422}}
\end{bmatrix}$$

$$\begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
\frac{\text{Km1}}{\text{I122 I211 I411 R1}} & 0 \\
0 & \frac{\text{Km2}}{\text{I222R2}} \\
0 & 0 \\
0 & \frac{\text{Km2}}{\text{I222R2}}
\end{bmatrix}$$

Figure E.1. $\theta_2 = \frac{\pi}{2}$.

$$\begin{vmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{\sqrt{2}\ k3}{2\,I122\ I211\ I233\ I411} & 0 & \frac{2\ Kb1Km1\ b1R1}{2I122\ I211\ I233\ I411\ R1} & 0 & \frac{\sqrt{2}\ b3}{2\,I122\ I211\ I233\ I411} & 0 \\ 0 & 0 & 0 & \frac{k4}{I222} & 0 & \frac{Kb2Km2\ b2R2}{I222R2} & 0 & \frac{b4}{I222} \\ 0 & 0 & \frac{2\,I122\ I211\ I233\ I411\ I433\ k3}{2\,I122\ I211\ I233\ I411\ I433} & 0 & \frac{\sqrt{2}\ Kb1Km1\ b1R1}{2\,I122\ I211\ I233\ I411\ R1} & 0 & \frac{b3\ 2\,I122\ I211\ I233\ I411\ I433}{2\,I122\ I211\ I233\ I411\ I433} & 0 \\ 0 & 0 & 0 & \frac{I222\ I422\ k4}{I222I422} & 0 & \frac{Kb2Km2\ b2R2}{I222R2} & 0 & \frac{b4\ I222\ I422}{I222I422} \end{vmatrix}$$

$$\begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{2\,Km1}{2\,I122\ I211\ I233\ I411\ R1} & 0 \\ 0 & \frac{Km2}{I222R2} \\ \frac{\sqrt{2}\ Km1}{2\,I122\ I211\ I233\ I411\ R1} & 0 \\ 0 & \frac{Km2}{I222R2} \end{vmatrix}$$

Figure E.2. $\theta_2 = \frac{\pi}{4}$.

$$\left[\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & \frac{\mathtt{k3}}{\mathtt{I122\ I233}} & 0 & \frac{\mathtt{Kb1Km1\ b1R1}}{\mathtt{I122\ I233\ R1}} & 0 & \frac{\mathtt{b3}}{\mathtt{I122\ I233}} & 0 \\
0 & 0 & 0 & \frac{\mathtt{k4}}{\mathtt{I222}} & 0 & \frac{\mathtt{Kb2Km2\ b2R2}}{\mathtt{I222R2}} & 0 & \frac{\mathtt{b4}}{\mathtt{I222}} \\
0 & 0 & \frac{\mathtt{I122\ I233\ I433\ k3}}{\mathtt{I122\ I233\ I433}} & 0 & \frac{\mathtt{Kb1Km1\ b1R1}}{\mathtt{I122R1\ I233R1}} & 0 & \frac{\mathtt{b3\ \ I122\ I233\ I433}}{\mathtt{I122\ I233\ I433}} & 0 \\
0 & 0 & 0 & \frac{\mathtt{I222\ I422\ k4}}{\mathtt{I222I422}} & 0 & \frac{\mathtt{Kb2Km2\ b2R2}}{\mathtt{I222R2}} & 0 & \frac{\mathtt{b4\ \ I222\ I422}}{\mathtt{I222I422}}
\end{array}\right]$$

$$\left[\begin{array}{cc}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
\frac{\mathtt{Km1}}{\mathtt{I122R1\ I233R1}} & 0 \\
0 & \frac{\mathtt{Km2}}{\mathtt{I222R2}} \\
\frac{\mathtt{Km1}}{\mathtt{I122\ I233\ R1}} & 0 \\
0 & \frac{\mathtt{Km2}}{\mathtt{I222R2}}
\end{array}\right]$$

Figure E.3. $\theta_2 = 0$.