

METHODOLOGICAL FRAMEWORK AND  
DESIGN METHODOLOGY FOR DEVELOPING  
UNDERWATER MANIPULATION SYSTEMS

**Bjørn Solvang**



**Narvik University College**



**The Norwegian University  
of Science and Technology**

**Narvik 2003**

ISBN 82-471-5585-0

NTNU2003:35



## **PREFACE**

This dissertation presents parts of the work to meet the requirements of the Doktor Ingeniør degree at the Norwegian University of Science and Technology.

The study has been carried out at the Narvik Institute of Technology (HIN) and at the Norwegian University of Science and Technology, Institute of Production and Quality engineering (NTNU/IPK).

First of all, my special thanks go to my supervisor Professor Terje K. Lien, for his support, encouragement, and guidance during my study.

Thanks to Professor W. H. Koch for his guidance into the area of non-linear optimisation, and to Dr.ing Nikola Tomac for his guidance in the beginning of my study.

I would like to express my gratitude towards the HIN and NTNU/IPK organisations, giving me the opportunity and the financial support to carry out this study.

Further, I would like to thank my friends and colleagues at HIN, NTNU and SINTEF. These people that makes the organisation alive, prosperous, and the daily life enjoyable.

I am greatly indebted to my family. Their belief, their encouragement, their support, always, has made the fulfilment of this dissertation possible.

Finally, my deepest gratitude goes to my wife Wei and my daughter Isabella. Your smiles makes the hard times vanish and your love has been the key for this work.

**Narvik, Norway**

April, 2003

Bjørn Solvang



## ABSTRACT

Until recently, underwater manipulation tasks have been conducted by human divers alone or in co-operation with ROVs equipped with one or two manipulator arms. Divers are flexible and can perform a lot of different work tasks, but diving operations are expensive and there is a great deal of risk to enter such a hostile environment.

Oil drilling operations around the world are moved into deeper water compared with those only a few years ago, and the humans can not be present in water depths below 600 meters maximum. So to be able to interfere with the sub-sea installations, to conduct inspection, repair and maintenance operations, there is a need to have automatic equipment with the necessary capability to perform both planned and unplanned intervention tasks.

In general there are two different ways to automate the work tasks:

- Either by creating a set of special equipment, specially made for every work task or
- By building/using a robotic systems

The first equipment offers less flexibility, and even a small change in the external conditions of the work task makes it necessary to physically modify the equipment. The robotic solution offers higher flexibility, but is more complicated to design and as well operate.

In this dissertation I have identified those underwater intervention work tasks that exist today and those that will be forced upon us in the near future, due to the strong incentive to “get the man out the water”. The bare amount of different work classes is a clear incentive to choose a flexible instead of a specialised solution when it comes to intervention equipment. Thus, the robotic solution (ROV-manipulator), which offers the necessary flexibility, is selected as the R&D subject of this dissertation.

The increasing rigorous requirements from off-shore oil and gas fields on underwater *unmanned* manipulation in the hostile and unstructured/semi-structured sub-sea environment require constant improvement of the ROV-based underwater manipulation systems. Thus, there is a need to develop methodologies for aiding to optimally and formally develop and design new generation underwater unmanned manipulation systems. For meeting such necessity, this dissertation works to identify an overall concept and methodological framework for underwater manipulator system development.

Further, the serial manipulator is identified to be a natural solution for the workspace constraint environment, and the serial structure is selected for further analysis in this dissertation. I have identified the necessity of developing a precise design

methodology, for aiding system designers to design a precise underwater serial arm manipulator working in a workspace constraint environment, capable of doing high precision work.

During the design stage of a precise manipulator system the necessity of evaluation loops has become clear. The design team needs to evaluate the effect of their design considerations/decisions upon the precision requirement. Thus, an iterative method of “design – evaluation - design-evaluation” is suggested as the precise design methodology. Gaining knowledge of the effect of possible choices enables the design group, before too many ruling decisions has been made, to make an optimum solution.

The precision evaluation or as we say, error analysis methodology, involves a procedure which aims to identify physical errors, find their influence onto the kinematic chain, calculate the error effected real kinematic chain and compare the ideal end-effector pose with the real kinematic pose, so as to identify the single error influence or selected (all) error influence. Both theoretical results and a case study is presented in this dissertation.

As the final step in the general design procedure, a prototype is to be built and tested. The ISO 9283 standard “Manipulating industrial robots, performance criteria and related test methods” describes the test methods for a set of important performances of the manipulator system. However, the outcome from the performance measurements is a numerical value of how big the deviation from the ideal selected reference is. Performance measures does not state **why** and from **where** the errors origin from. This latter is especially important for the designer to have more knowledge about, because he/she may have to make changes to the construction, if some of the performance measurement is out of the specified requirements. In the final part of this dissertation, an experimental error mapping methodology is outlined. With the benefit of giving the designer a possible way to split up the numerical value from the performance measurements and map these numerical values to specific parts or areas of the construction.

# TABLE OF CONTENTS

<b>PREFACE</b> .....	<b>III</b>
<b>ABSTRACT</b> .....	<b>V</b>
<b>TABLE OF CONTENTS</b> .....	<b>VII</b>
<b>LIST OF FIGURES</b> .....	<b>XI</b>
<b>CHAPTER 1 INTRODUCTION AND RESEARCH OBJECTIVES</b> .....	<b>1</b>
1.1 INTRODUCTION .....	1
1.2 RESEARCH MOTIVATION AND PROBLEM STATEMENT.....	2
1.3 RESEARCH OBJECTIVE .....	6
1.4 THESIS STRUCTURE .....	6
<b>CHAPTER 2 METHODOLOGICAL FRAMEWORK FOR DEVELOPING UNDERWATER UNMANNED MANIPULATION SYSTEMS</b> .....	<b>9</b>
2.1 INTRODUCTION .....	9
2.2 A BRIEF HISTORICAL EXCURSION ON DEVELOPING UNDERWATER MANIPULATION SYSTEMS .....	13
2.3 REQUIREMENTS DEFINITION FOR DEVELOPING UNDERWATER MANIPULATION SYSTEMS AND REQUIREMENT DEFINITION MATRIX .....	14
2.3.1 <i>Task requirement identification</i> .....	14
2.3.2 <i>Capability requirement identification</i> .....	15
2.3.3 <i>A requirement definition matrix</i> .....	16
2.4 METHODOLOGICAL FRAMEWORK FOR AIDING THE DEVELOPMENT OF ROV-BASED UNDERWATER MANIPULATION SYSTEMS.....	18
2.4.1 <i>Domain identification</i> .....	18
2.4.2 <i>Methodological framework</i> .....	21
2.5 CONCLUSION OF THIS CHAPTER.....	22
<b>CHAPTER 3 STRUCTURE OF UNDERWATER MANIPULATORS FOR WORKSPACE CONSTRAINT ENVIRONMENT</b> .....	<b>25</b>
3.1 INTRODUCTION .....	25
3.2 VARIOUS CONFIGURATIONS OF UNDERWATER MANIPULATION SYSTEMS .....	27
3.2.1 <i>Configuration of vessel supported ROV-based manipulation systems</i> .....	27
3.2.2 <i>Configuration of platform supported ROV-based manipulation systems</i> .....	28
3.2.3 <i>Configuration of non umbilical supported systems</i> .....	29
3.3 MANIPULATOR STRUCTURES FOR WORKING IN UNDERWATER WORKSPACE CONSTRAINT ENVIRONMENT .....	31
3.4 CONCLUSION OF THIS CHAPTER.....	34

**CHAPTER 4 DESIGN ISSUES AND PRECISE DESIGN METHODOLOGY FOR DEVELOPING WORKSPACE CONSTRAINT UNDERWATER MANIPULATORS... 37**

4.1 INTRODUCTION ..... 37

4.2 DESIGN ISSUES OF UNDERWATER MANIPULATORS ..... 37

    4.2.1 *Design procedure for underwater manipulators* ..... 37

        4.2.1.1 *Conceptual design* ..... 40

        4.2.1.2 *Preliminary design* ..... 41

        4.2.1.3 *Detail design and prototyping* ..... 42

    4.2.2 *A further discussion on design issues of underwater manipulators* ..... 44

4.3 CONTROL STRATEGY AND CONTROL ARCHITECTURE FOR UNDERWATER MANIPULATION 46

    4.3.1 *Master-slave mode* ..... 46

        4.3.1.1 *Master- slave strategy* ..... 46

        4.3.1.2 *Master- slave control architectures* ..... 47

        4.3.1.3 *Summary of conceptual master- slave control design* ..... 52

    4.3.2 *Joystick mode* ..... 53

        4.3.2.1 *Joystick strategy* ..... 53

        4.3.2.2 *Joystick control architectures* ..... 54

        4.3.2.3 *Summary of conceptual joystick control design* ..... 62

    4.3.3 *Computerised high-accuracy trajectory mode* ..... 64

        4.3.3.1 *Computerised high-accuracy trajectory strategy* ..... 64

        4.3.3.2 *Computerised high-accuracy trajectory control architectures* ..... 66

        4.3.3.3 *Summary of conceptual trajectory control design* ..... 69

    4.3.4 *Summary of section 4.3* ..... 70

4.4 METHODOLOGY FOR ANALYSIS OF PRECISE DESIGN ..... 73

    4.4.1 *Introduction* ..... 73

    4.4.2 *Identify the ideal kinematic chain, and the ideal end-effector pose* ..... 77

        4.4.2.1 *Denavit-Hartenberg convention for coordinate frame assignment* ..... 77

        4.4.2.2 *A general 6 parameter description of coordinate relations* ..... 80

    4.4.3 *Identify physical error components in the design and their relation to the kinematic chain* ..... 84

        4.4.3.1 *Identification of physical error components* ..... 84

        4.4.3.2 *Physical errors and their relation to the kinematic chain* ..... 90

    4.4.4 *Identify the real kinematic chain, and the real end-effector pose* ..... 94

    4.4.5 *Calculate influence of single error components to the end-effector* ..... 96

    4.4.6 *Calculate influence of all or a selection of error components to the end-effector* ..... 97

    4.4.7 *Other error modelling methods* ..... 97

    4.4.8 *Summary of section 4.4* ..... 99

4.5 CONCLUSION OF THIS CHAPTER ..... 100

**CHAPTER 5 CASE STUDY OF SERIAL STRUCTURE MANIPULATOR, ACCURACY ANALYSIS..... 103**

5.1 INTRODUCTION ..... 103

5.2 IDENTIFICATION OF THE IDEAL KINEMATIC CHAIN, AND THE IDEAL END-EFFECTOR POSE 104

    5.2.1 *Ideal forward kinematic chain* ..... 105

    5.2.2 *End effector ideal pose* ..... 107



5.3 IDENTIFY PHYSICAL ERRORS; IDENTIFY THEIR RELATION TO THE KINEMATIC CHAIN. ....	107
5.3.1 <i>Error identification of beam element</i> .....	109
5.3.2 <i>Error identification of joint 3 housing element</i> .....	121
5.3.3 <i>Error identification of joint 3 components</i> .....	126
5.4 IDENTIFICATION OF THE REAL KINEMATIC CHAIN, AND THE REAL END-EFFECTOR POSE..	131
5.4.1 <i>Real forward kinematic chain</i> .....	131
5.4.2 <i>End effector real pose</i> .....	136
5.5 CALCULATION OF THE END EFFECTOR ERROR .....	136
5.5.1 <i>Calculation of single error influence</i> .....	137
5.5.2 <i>Calculation of combined error influence</i> .....	138
5.6 CONCLUSION OF THIS CHAPTER.....	139
<b>CHAPTER 6 EXPERIMENTAL ERROR MAPPING METHODOLOGY .....</b>	<b>141</b>
6.1 INTRODUCTION .....	141
6.2 MANIPULATOR TESTBED .....	142
6.2.1 <i>Mechanical design</i> .....	143
6.2.2 <i>Control system design</i> .....	144
6.3 MEASUREMENT SYSTEM .....	147
6.4 ERROR MAPPING METHODOLOGY .....	158
<b>CHAPTER 7 CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK .</b>	<b>169</b>
7.1 CONCLUSION .....	169
7.2 RECOMMENDATIONS FOR FUTURE WORK .....	171
<b>REFERENCES.....</b>	<b>173</b>



## List of figures

- Figure 2.1** An example of ROV-based underwater manipulation system
- Figure 2.2** An example of ROV-manipulator subsystem
- Figure 2.3** A diver is working at underwater structure co-operated with an ROV-manipulator
- Figure 2.4** Water depths in some oil and gas fields
- Figure 2.5** We said diverless, not hopeless
- Figure 2.6** A ROV manipulation system
- Figure 2.7** Italian TM308 system
- Figure 2.8** Japanese marine robot
- Figure 2.9** Block diagram of a sub-sea handling system
- Figure 2.10** Methodological framework for methodology research
- Figure 3.1** Categorization of underwater manipulation systems according to variant supply and surface support systems
- Figure 3.2** Categorization of underwater manipulation systems according to work site reachability
- Figure 3.3** An example of ROV-based underwater manipulation system
- Figure 3.4** A manipulator is working at a subsea structure with a much stronger but less dexterous docking device (rotating and tilting claw)
- Figure 3.5** A manned submergence vehicle DSV4
- Figure 3.6** A manned submarine NR-1
- Figure 3.7** An AUV-based underwater system
- Figure 3.8** A serial structure manipulator
- Figure 3.9** An instance of parallel structure
- Figure 3.10** The MULTICRAFT 560 Robot
- Figure 3.11** A serial arm, HIPRUM, with 3 joints
- Figure 4.1** Flowchart of robot design (Andeen, 1988)
- Figure 4.2** Conceptual design procedure
- Figure 4.3** Preliminary design procedure
- Figure 4.4** Detail design and prototyping procedure
- Figure 4.5** Position architecture
- Figure 4.6** Layout of a position-position control system (Talor, 1993)
- Figure 4.7** Position-Position control architecture
- Figure 4.8** Layout of a position-force control system (Talor, 1993)
- Figure 4.9** Position-Force control architecture
- Figure 4.10** Layout of a bilateral control system (Talor, 1993)
- Figure 4.11** Bilateral control architecture
- Figure 4.12** Control strategy of master-slave mode for underwater manipulators
- Figure 4.13** Joystick joint coordinate rate control (open servo loop) architecture
- Figure 4.14** Joystick joint coordinate rate control (closed servo loop) architecture
- Figure 4.15** Joint coordinate position control (closed servo loop) architectures
- Figure 4.16** Base coordinate rate control (closed servo loop) architecture
- Figure 4.17** Base coordinate position control (closed servo loop) architecture
- Figure 4.18** f.)End-effector coordinate rate control (closed servo loop) architecture

- g.) End-effector coordinate position control (closed servo loop) architecture
- Figure 4.19** h.) External coordinate rate control (closed servo loop)
- i.) External coordinate position control (closed servo loop)
- Figure 4.20** Control strategies of master-slave mode and joystick mode for underwater manipulators
- Figure 4.21** a.) Trajectory control with compensation of a moving base  
b.) Trajectory control with compensation of a moving base and automatic contact force control
- Figure 4.22** Control strategies for underwater manipulators
- Figure 4.23** Reference conceptual control strategy
- Figure 4.24** An iterative process of “design – analysis – design – analysis ...”
- Figure 4.25** An example manipulator
- Figure 4.26** An error analysis procedure for the analysis methodology
- Figure 4.27** A linkage system with coordinate system assignment
- Figure 4.28** Distance vector  $d_{i-1}^i$
- Figure 4.29** Orientation convention, consecutive CBA rotations
- Figure 4.30** A sketch of a manipulator with two joints and an end-effector
- Figure 4.31** Physical error **E1** in link element i-1
- Figure 4.32** A design example of a manipulator joint
- Figure 4.33** A design example, joint housing of a manipulator joint
- Figure 4.34** Maximum orientation error due to parallelism tolerance
- Figure 4.35** Physical error **E1**, **E2** in link element i-1
- Figure 4.36** Coordinate assignment to error sources **E1** and **E2**
- Figure 4.37** Example coordinate assignment
- Figure 4.38** Coordinate assignment to error sources **E1** and **E2**
- Figure 4.39** Frame displacement due to errors,
- Figure 4.40** A summarized sketch of chapter 4
- Figure 5.1** Design case
- Figure 5.2** Arm in staging pose
- Figure 5.3** Assignment of coordinate systems
- Figure 5.4** Coordinate system response to joint activation
- Figure 5.5** Exploded top view of selected area for error analysis
- Figure 5.6** Cut through of selected design with selected internal components
- Figure 5.7** Beam element, top view
- Figure 5.8** Maximum orientation error due to perpendicularity tolerance
- Figure 5.9** Deflection model
- Figure 5.10** Beam load model
- Figure 5.11** Beam load model, single force F
- Figure 5.12** Beam load model, beam weight q
- Figure 5.13** Beam load model, torque T
- Figure 5.14** Side view of beam. Compliance analysis
- Figure 5.15** Coordinate systems for beam element
- Figure 5.16** Joint 3 housing, top view
- Figure 5.17** Joint 3 housing, left view .

- Figure 5.18** Translational movement of coordinate system  $K_D$
- Figure 5.19** a.) Proper mesh. b.) Improper mesh, backlash.
- Figure 5.20** a.) Proper mesh, zero gaps b.) Improper mesh, gap on lagging side.
- Figure 5.21** Deflection torque model
- Figure 5.22** Simplified deflection torque model, harmonic drive gears, (courtesy of Harmonic Drive GmbH)
- Figure 5.23** Coordinate assignment
- Figure 6.1** Manipulator system with three rotary joints
- Figure 6.2** Manipulator system, kinematic layout
- Figure 6.3** Joint layout
- Figure 6.4** Manipulator (2 joint) and peripherals including a force feedback joystick
- Figure 6.5** Manipulator conceptual trajectory control architecture
- Figure 6.6** Step response joint 3. P controller with gain  $K_p=0.8$
- Figure 6.7** LEICA LTD 500 Laser Tracking System (courtesy Leica AG, Switzerland)
- Figure 6.8** 2D measurement set-up
- Figure 6.9** Incremental 1D wire sensor (courtesy ASM, Germany)
- Figure 6.10** Layout of measurement system
- Figure 6.11** Function  $f(\alpha_1, \alpha_2)$ , for a given measured value (sensor 1 and 2)
- Figure 6.12** Function  $f(\alpha_1, \alpha_2)$ , 3D contour plot close to minimum point
- Figure 6.13** NOSYS® flowchart (Koch, 1994)
- Figure 6.14** a.) Result file MATLAB® optimisation  
b.) Result from NOSYS®  
c.) Comparison of functional value  $f(\alpha_1, \alpha_2)$
- Figure 6.15** Reference trajectory
- Figure 6.16** Reference and actual trajectory, derived from external measurements
- Figure 6.17** Manipulator error mapping control architecture
- Figure 6.18** Reference trajectory and calculated trajectory
- Figure 6.19** Reference trajectory and calculated trajectory based on joint measurements
- Figure 6.20** Reference joint trajectory and actual joint trajectory, joint 2
- Figure 6.21** Reference joint trajectory and actual joint trajectory, joint 3
- Figure 6.22** Servo loop error, joint 2
- Figure 6.23** Servo loop error, joint 3



## Chapter 1

### Introduction and Research Objectives

#### 1.1 Introduction

The ocean covers about two-thirds of the earth and has a great effect on the future existence of all human beings. About 37% of the world's population lives within 100 km of the ocean (Cohen, Small, Mellinger, Gallup and Sachs, 1997). The ocean is generally overlooked as we focus our attention on land and atmospheric issues; we have not been able to explore the full depths of the ocean and its abundant living and non-living resources. For example, it is estimated that there are about 2,000 billion tons of manganese nodules on the floor of the Pacific Ocean near the Hawaii Islands. Only recently we have discovered, by using manned submersibles, that a large amount of carbon dioxide comes from the seafloor and extraordinary groups of organisms live in hydrothermal vent areas. Underwater robots (or called underwater manipulators) can help us better understand marine and other environmental issues, protect the ocean resources of the earth from pollution, and efficiently utilize them for human welfare. However, a number of complex issues due to the unstructured, hazardous undersea environment make it difficult to travel in the ocean even though today's technologies have allowed humans to land on the moon and robots to travel to Mars (Yuh, 2000).

Unmanned underwater vehicles are, at the present stage of technology, the key tools for exploring and exploiting resources located at great sea depths. In particular, Remotely Operated Vehicles (ROVs) and Autonomous Unmanned Vehicles (AUVs) are currently being employed to perform several tasks that range from scientific and environmental data gathering to inspection and assembly of submarine installations. ROVs are generally vehicles tethered to a supporting vessel and are being used mainly for inspecting and building underwater structures as pipelines and off-shore platforms, while AUVs exhibit higher maneuverability and do not require the presence of an umbilical; for this reason, the latter are employed for long term missions like exploration and environmental data analysis. The growing interest for the underwater environment motivates a great deal of research efforts aiming at endowing ROVs and AUVs with enhanced capabilities in order to lower the cost of such missions and increase safety and reliability. The basic issue for both ROVs and AUVs is the development of automatic control schemes that guarantee high performances in motion and positioning, independently of the human operator's skill (Conte and Serrani, 1996).

The offshore oil industry has so far provided the major market for underwater manipulator systems (Taylor, 1993). Deepwater oil and gas fields are in various stages of development worldwide. New discoveries are now being made at water depths in excess of 2400 m (8000 feet). The industry has responded to this deepwater rushing with an anticipated delivery of 26 deepwater-drilling units' set for year-end, 2000. This leaves a further 29 deepwater-drilling units still under construction. Most of these

drilling units will be capable of work in water depths between 1800 and 3000 m (6000 and 10000 feet). Therefore, a substantial challenge will now come in the form of creative engineering, development, maintenance and finally the decommissioning of these future deepwater fields. (Garmulewicz, 2000)

To summarise, by using unmanned underwater vehicles for exploring and exploiting resources located at great sea depths, following missions are assigned to

- ROVs: inspecting, maintaining and building of underwater structures,
- AUVs: long term missions like exploration and scientific and environmental data gathering and analysis.

Due to the offshore oil industry has so far provided the major market for underwater manipulator systems, in this thesis, I focus my research in the area of ROV-based underwater unmanned manipulation systems, especially in the subject of *guaranteeing high performances* in positioning and motion of the ROV-based underwater unmanned manipulators.

## **1.2 Research Motivation and Problem Statement**

Until the last decade or two, virtually all work performed below the surface of the sea was accomplished by divers or by men working in submarines. The utilisation of submarines was limited because their physical size prevented easy access to the work area and due to the fact that work had to be accomplished by manipulators operated by the pilot through the pressure hull. Divers, operating in modern diving suits, provided a better alternative because they were capable of exhibiting much of the manual dexterity possessed by men working on the surface. Divers, too, had the limitations in that the maximum depth at which they could perform useful work was less than 300 m (1000 feet) below the surface. Additionally, in all but the shallowest of depths, the divers had to be provided with expensive mixtures of gases to breathe, had to be kept in saturation and had decompression times of about a week after completing the work before they could return to the normal atmosphere. For all of these reasons, the efficiency of divers was low and the costs were high (Langrock and Broome, 1994).

At water depths greater than 200 m it is now more economic to use remote intervention than divers (Marine Tech. Dir. Ltd., 1992). Divers can survive for extended periods at depths of as great as 500 m, but the price paid in time descending and ascending safety is very considerable, such diving is becoming increasingly more difficult to accept as a practical proposition (Hempleman and Lockwood, 1978; Greig, Wang and Broome, 1992).

Current underwater robotic systems typically comprise one or more manipulators mounted on the front of a ROV, equipped with an underwater camera system. They are tele-operated and consequently place a large workload burden on the human operator. The manipulators are operated in a master-slave configuration by an operator on the surface vessel. The movement of the smaller master arm is replicated by the larger



slave arm and they form an approximately spatially correspondent system. However, the operator has a number of handicaps that contribute to the difficulty of performing the task at hand (Lane, Dunnigan, Knightbridge and Quinn, 1991). It is difficult and tiring to **produce straight line motion** of the slave arm end-effector by manually operating the master arm. The operator's perception of the underwater scene from the video camera is two-dimensional and so there is no depth cue. This monochrome image is often of poor quality and there is limited visibility underwater, which can degrade dramatically when operations are performed. The operator is only aware that the slave manipulator has hit an object when the slave manipulator motion is no longer spatially correspondent with the master arm, potentially doing damage to the manipulator, object, or both (Dunnigan, Lane, Clegg and Edwards , 1996).

Kallevik and Hendseth (1991) mentioned also that the existing underwater manipulators still have some considerable drawbacks when **applied in high precision work**. With remote manual joystick rate control it is extremely difficult to **follow a three-dimensional curve**. Even with master/slave control this task is difficult and cumbersome.

Many non-destructive testing (NDT) methods require an inspection probe to be moved across the surface of the object under investigation e.g. ultrasonics, eddy current and alternating current potential drop (ACPD). Some such as alternative current field measurement (ACFM) can operate with a small lift-off (a few millimeters) from the surface, but they still **require close contour following** in order to size cracks. (Greig et al., 1992)

Langrock and Broome (1994) mentioned that while ROVs provide the solution to subsea manipulation, the present designs prevent them from performing many tasks for which divers must still be used. One of these tasks is the non-destructive testing (NDT) inspection of structural welds on the submerged portions of offshore platforms used for the production of hydrocarbons. The available ROV manipulators and their control systems are primitive in design, have little manual dexterity and certainly are not capable of **providing the required accuracy** and repeatability of placement required for NDT. Additionally, the present TV systems do not provide the pilot or manipulator operator with the 'tele-presence' needed to effectively **track welds at the intersection of a structure**.

Langrock and Broome (1994) mentioned that the necessity of designing a special-purpose manipulator developed to place various types of non-destructive testing devices on or near welds with sufficient accuracy to find surface and near-surface fatigue cracks with a high probability of detection. Special computer software must be developed to accurately model the workpiece in situ, define the intersection of the structural members and to **create the trajectory that the manipulator must follow to accurately track the welds**. This is a task that would be virtually impossible if required to be accomplished using conventional master/slave control.

To collect the problems discussed above, they can be categorized as:

- ♣ Problems in meeting high precision work
  - Unable to be applied in high precision work
  - Unable to provide required accuracy
  
- ♣ Problems in meeting the requirement of following a three dimensional curve (trajectory) accurately
  - Unable to produce straight line motion accurately
  - Unable to follow a three dimensional curve
  - Unable to do close contour following
  - Unable to track welds at the intersection of an underwater structure
  - Unable to create the trajectory that the manipulator must follow to accurately track the welds.

I have also made a literature survey into one scientific and engineering publication database, Science Citation Index Expanded Database of ISI (Institute for Scientific Information, Inc.) on the state-of-the-art in the area of underwater manipulation and underwater robotics.

In the survey, I used a keyword statement as ***underwater \* (manipulator + manipulation + intervention + robot + robotic + robotics)***. It resulted in 192 articles and editorial materials. Those articles and editorial materials are categorized as shown in Table 1.1.

In the items 10 and 11 of Table 1.1, it shows that R&D for the ROV design issues and underwater manipulator design issues are less researched with only 10 papers (5.2%) out of 192 papers. I have further read the full papers of those 10 papers and found that none of those papers has studied on the topic of *precise* design issue. In the items of 12 and 13, it shows that deepwater welding, which certainly relates to precise trajectory following, is less researched also with only two papers (1.1%) out of 192 papers, and R&D related to trajectory following and robot motion planning is again less researched with two papers (1.1%) out of 192 papers. Further, I have read the full papers of those 4 papers and found that none of those papers has studied on the precise trajectory following issue for underwater manipulations.

**Table 1.1** Categorization of searched articles and editorial materials from ISI

	<b>Categorization of articles</b>	<b>No. of articles</b>	<b>Percentage (%)</b>	<b>Remarks</b>
1	General discussion of underwater manipulation and underwater robot	24	12.5	
2	Navigation, localization, obstacle avoidance and navigation path planning	31	16.1	
3	Control, simulation and coordination of multi-manipulators	47	24.5	Including master-slave control, adaptive control, stochastic control, navigation motion control, non-linear control, neural control, etc.
4	Other systems other than underwater manipulation systems	28	14.6	Including space flight systems, complex eletromechanical systems, fish robot systems, dolphins cognition systems, nuclear waste retrieval systems, etc.
5	NDT	7	3.6	
6	Various manipulations other than NDT	11	5.7	Including manipulations of heavy load, cable maintenance, reactor dismantlement, oceanographic use, deepwater sampling, etc.
7	Divers and diving	4	2.1	
8	AUV	15	7.8	
9	Sensors and sensing	11	5.7	Including video mosaicking, visual perception, acoustic imaging, sonar sensing, force sensing, laser imaging, etc.
10	<b>ROV design</b> issue	5	2.6	
11	<b>Underwater manipulator</b> (robot) <b>design</b> issue	5	2.6	
12	<b>Trajectory</b> following and robot motion planning	2	1.1	
13	<b>Deepwater welding</b>	2	1.1	

## 1.3 Research Objective

Facing above stated problems, considering the less researched situation, and combining with the research motivation description in sections 1.1 and 1.2, I made up my mind to put my research focus as follows

- ***R&D in the area of ROV-based underwater-unmanned manipulation systems***, due to the offshore oil industry has so far provided the major market for underwater manipulator systems (see the description in section 1.1).
- ***Set up a methodological framework*** for formally aiding the requirement definition for developing ROV-based underwater-unmanned manipulation systems.
- ***Set up a precise design methodology*** for aiding to develop precise underwater manipulators for meeting requirements of high precision work and accurately following three-dimensional curve (trajectory).

## 1.4 Thesis Structure

This dissertation consists of 7 chapters which follow the natural sequence of the scope of research.

### **Chapter 1**

Introduction and problem statement.

### **Chapter 2**

The increasing rigorous requirements from off-shore oil and gas fields on underwater *unmanned* manipulation in the hostile and unstructured/semi-structured sub-sea environment require constant improvement of the ROV-based underwater manipulation systems. Thus, there is need to develop methodologies for aiding to optimally and formally develop and design new generation underwater unmanned manipulation systems. For meeting such necessity, this chapter first defines a requirement definition matrix, which can be referred for aiding to define complete set of requirements for the new development of underwater manipulation systems. Further by combining the requirement definition matrix with the 3<sup>rd</sup> dimension of R&D domains, an overall concept and methodological framework for underwater manipulator system development is achieved.

### **Chapter 3**

Chapter 3 works upon the structure of underwater manipulators for working in a constraint environment. Identification of a suitable structure that can perform a range of work task is conducted.

### **Chapter 4**

In Chapter 4, the design procedure of underwater intervention manipulators is put on order. The goal is to investigate existing design methodologies and revising their capability to achieve the desired objective. Design evaluation technique is therefore a central point of research in this chapter. A new error modeling methodology is

presented, which enables the design to be evaluated with respect to the precision requirement. Control strategies and control architectures of underwater manipulators are studied since they are proved to play a very important role in the design process of underwater manipulators.

### **Chapter 5**

In Chapter 5, a case study to exemplify the usage of the design error evaluation technique is carried out. A selected serial structure is used as a case study, typical component and structure errors are identified and their influences to the end effector pose are calculated. Both single error analysis and combined error analysis is conducted in this part of the dissertation.

### **Chapter 6**

In Chapter 6, I discuss the final step in all design procedures, prototype testing. Performance measures is defined by ISO standards, but performance measures does not state **why** and from **where** the errors origin from. In the final part of this chapter, a experimental error error mapping methodology is outlined. With the benefit of giving the designer a possible way to split up the numerical value from the performance measurements and map these numerical values to specific parts or areas of the construction.

### **Chapter 7**

Conclusion and the recommendations for the future work.

**Parts of this work have been presented at the following international conferences:**

30<sup>th</sup> International Symposium on Robotics, Tokyo, Japan.

Solvang, B., Lien, T.K. and Thomessen, T. (1999), A high precision underwater manipulator, 30<sup>th</sup> International Symposium on Robotics, Tokyo, Japan.

Ocean 2001 MTS/IEEE Conference

Solvang, B., Deng, Z., and Lien, T. K. (2001a), “A Methodological Framework for Developing ROV-manipulator Systems for Underwater Unmanned Intervention”, ISBN: 0-933957-29-7

Ocean 2001 MTS/IEEE Conference

Solvang, B., Deng, Z., and Lien, T. K. (2001b), “Structure of Underwater Intervention Manipulators in Workspace Constraint Environment and Architecture of their Control System”, ISBN: 0-933957-29-7

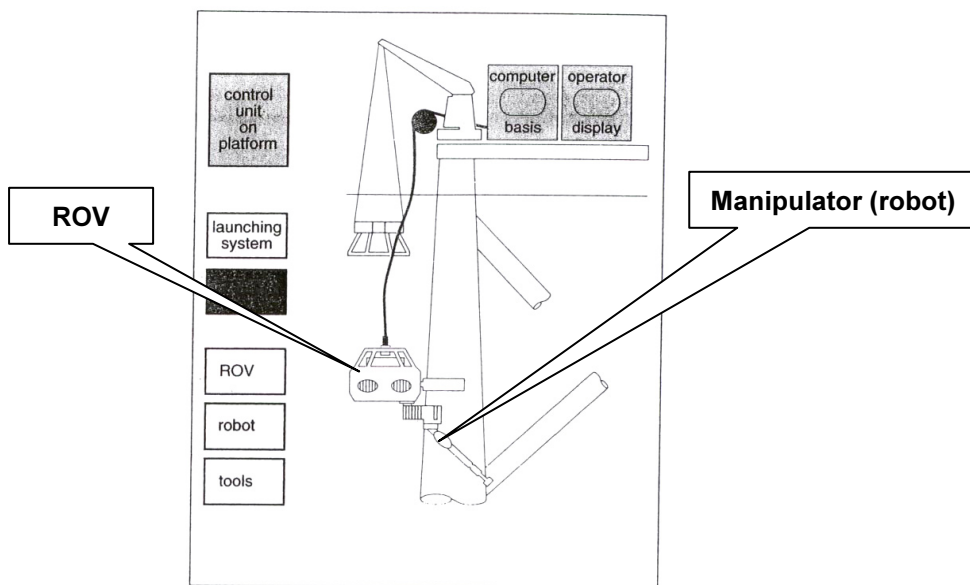
In addition to these international conferences parts of the work have been presented to users of the ROV-manipulator technology, in general oil companies.

## Chapter 2

# Methodological Framework for Developing Underwater Unmanned Manipulation Systems

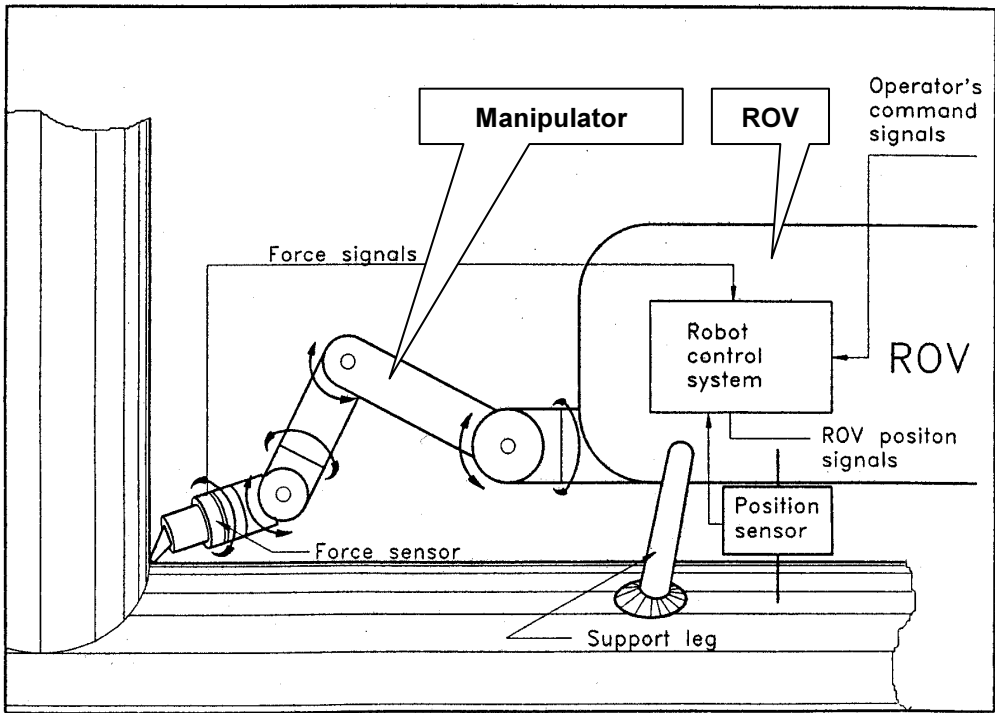
## 2.1 Introduction

In Chapter 1, I have mentioned that in this dissertation I will focus my work in the area of ROV-based underwater-unmanned manipulation systems. An example of ROV-based underwater-manipulation system is shown in Figure 2.1 (Aust, Gustmann, Niemann and Schulhreiss, 1992). This is a platform supported ROV-based underwater-manipulation system where the ROV-manipulator (ROV-robot) subsystem is connected to the control unit on the platform, via the launching system and tether management system.

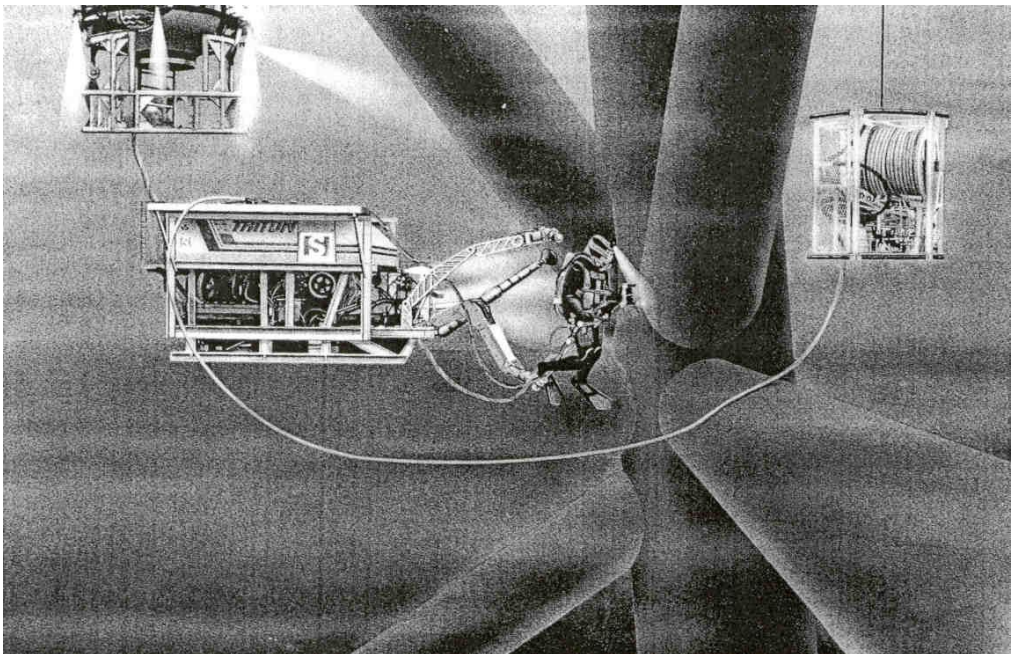


**Figure 2.1** An example of ROV-based underwater manipulation system

An ROV-manipulator subsystem comprises typically of one or more manipulators mounted on the front of a ROV (Dunnigan, Lane, Clegg and Edwards, 1996). An example of ROV-manipulator subsystem is shown in Figure 2.2 (Lien, Aune and Jenssen, 1991) where the ROV is parking at the underwater structure with its support leg (see lower part of Figure 2.2). A six DOF (degree of freedom) manipulator (robot) is mounted on the front of the ROV. The ROV has its local controller (named as robot control system in Figure 2.2). The local controller is connected to the on-surface control units via a tether management system, which are not shown in Figure 2.2.



**Figure 2.2** An example of ROV-manipulator subsystem



**Figure 2.3** A diver is working at underwater structure co-operated with an ROV- manipulator



Until recently, underwater manipulation tasks have been conducted by human divers alone or in co-operation with ROVs (Solvang, Deng and Lien, 2001). Figure 2.3 shows an example where a diver is working at underwater structure co-operated with an ROV-manipulator (Vartdal, 1990).

Divers are flexible and can perform a lot of different work tasks, but diving operations are expensive and there is a great deal of risk to enter such a hostile environment. Oil drilling operations around the world are moved into deeper water compared with those only a few years ago, and the humans can not be present in water depths below 600 meters maximum. So to be able to interfere with the sub-sea installations, to conduct inspection, repair and maintenance operations, there is a need to have automatic equipment with the necessary capability to perform both planned and unplanned intervention tasks (Solvang et al., 2001).

However, even saturated diving in water depths less than 600 m is possible, but not desirable for a number of reasons. The physical and psychic strain to the human body and as well the high costs associated with saturated diving are too apparent and a strong incentive to “get the man out of the water”. Attempts were made to carry out remotely many of the tasks previously only accomplished by divers, and increasing demands were therefore being made on the manipulative capabilities of the ROVs (Taylor, 1993).

**Table 2.1** Comparative costs of ROV and diver operation

(a)	Platform based ROV, Cost	100%
(b)	ROVSV <sup>1</sup> Based ROV, Cost (plus a greater propensity for WOW <sup>2</sup> )	300%
(c)	DSV <sup>3</sup> Based ROV, Cost (plus a greater propensity for WOW)	500%
(d)	DSV Based Diver, Cost	800%

<sup>1</sup> ROVSV - ROV support/supply vessel

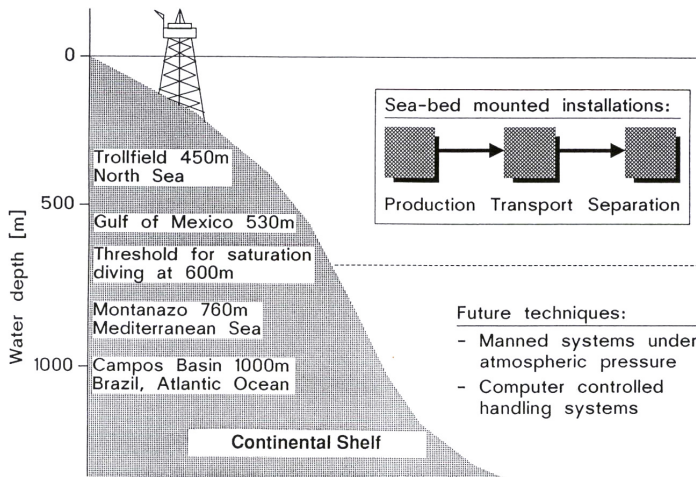
<sup>2</sup> WOW - Waiting on weather

<sup>3</sup> DSV - Dynamic positioning support vessel

On a typical DSV (Dynamic position Support Vessel) operating in the North Sea, a crew of up to seventy men is supporting one man working under water (Vartdal, 1990). Raine & Lugg (1995) gave examples of possible cost as shown in Table 2.1. From Table 2.1, we see that a DSV based diver underwater manipulations are much more expensive than other types of underwater manipulations.

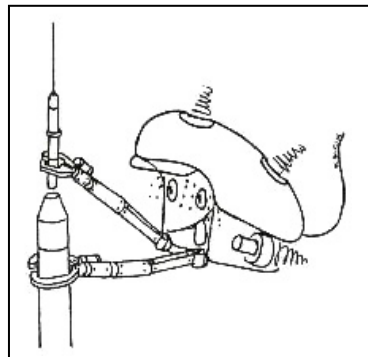
Solvang, Lien and Thomessen (1999) mentioned that sub-sea drilling operations in Norway and the world are nowadays carried out in deeper water compared with those only a few years ago. These operations demand newer technologies and equipment for underwater production, transport and maintenance other than we use today. As known, seabed below water surface approximately 600 meters disables the use of human divers, and therefore all sub-sea operations should be highly automated. The need for

manipulators, mounted on a ROV or placed directly in the underwater production line, is obvious.



**Figure 2.4** Water depths in some oil and gas fields

Aust, Gustmann and Niemann (1994) displayed the water depths of some oil and gas fields as shown in Figure 2.4. They illustrated that in the “Trollfield” (North Sea) and in the Gulf of Mexico (see Figure 2.4) sub-sea work is predominantly done with diver support. However in the ongoing exploitation of oil and gas in the Mediterranean Sea (760 m at Montanazo) and especially in the Brazilian Campos Basin (1000 - 2000 m at Albocora) assistance by divers is no more possible. At the moment, threshold for saturation diving is about 600 m and any new development area exceeding this limit in water depth requires support of highly advanced technical handling systems.



**Figure 2.5** We said diverless, not hopeless

When water depths increase beyond 600 m, all of the sub-sea tasks of offshore oil and gas fields that are currently being undertaken by divers must be totally replaced by ROV-based underwater *unmanned* manipulation systems. Haugvaldstad (1994) mentioned that the conversion from diver to ROV is important because experts in

diving medicine express concerns on the negative effect of deep diving. So field development into deeper and deeper waters, have changed many operators' philosophy into moving away from diving. This concept was depicted as shown in Figure 2.5 (Haugvaldstad, 1994), which shows that a machine-diver (ROV-manipulator) is used to replace the human diver in underwater manipulation. Or say, we want to realise personification by means of machines for underwater manipulation.

To summarize above discussion, we see that

- If water depths are larger than 600 *m*, all sub-sea tasks must be executed by underwater-*unmanned* manipulation systems.
- Saturated diving can be performed in water depths less than 600 *m*, but due to the physical and psychic strain to the human body, as well as the high cost, it is a strong incentive to “get the man out of the water”.
- Thus to develop underwater unmanned manipulation systems is an absolute necessity at the present time.
- Because this dissertation focuses in the case of offshore oil and gas fields, the ROV-based underwater-unmanned systems are taken as the objective research systems, as been discussed in Chapter 1.

## **2.2 A brief historical excursion on developing underwater manipulation systems**

In what follows, let us have a brief historical excursion on the development of ROV-based underwater manipulation systems. Then, from the historical recursion, we may further discuss the design methodology issues for developing the ROV-based underwater manipulation systems.

Keith (1997) mentioned that ROV systems were initially developed and operated by the U.S. Navy to meet specific requirements. Once released by the military, the private sector expanded the concept and purpose of ROVs, as they were no longer constrained by the physical dimensions that dictated the Navy's design. With the luxury of ‘unlimited’ space, vehicle sizes and capabilities grew larger. Original vehicle systems were pretty basic; little more than roving cameras that was hand tended. With the influx of newer models, this resulted in delineating between “Inspection” and “Work class” systems.

The ROV systems that were deemed to be “Inspection Subs” were typically small vehicles with basic capabilities, limited flexibility, and minimal power. The “Work Class” vehicles were fitted with stronger thrusters and correspondingly larger hydraulic systems. Manufacturers were more than happy to make bigger and, not surprisingly more expensive vehicles. This spiral in ROV power resulted in very large “packages”.

ROV usage has gone from a cute little novelty in the late 70s to today's critical path tasks that are essential for the continued expansion into “deepwater”. This is because

the vehicles have metamorphosed from small, barely reliable eyeballs to 100 HP, multi-camera, multiplexed, multi-tasking units that can stay underwater, working, for over a week. This evolution was a combination of necessity and *salesmanship*, but overlooks a basic premise of service oriented business: “What does the customer need?” By continuing to build larger, heavier, and more expensive equipment, the ROV industry is burdening the oil companies and other contractors (Keith, 1997).

Seeing that the evolution mentioned above was a combination of necessity and salesmanship, we would raise questions such as: “Do the existing ROV-based underwater manipulation systems cover the whole scope of requirements of underwater intervention?” and “Are the existing ROV-based underwater manipulator systems optimally task-compliant and cost effective?” From these questions, and for avoiding the salesmanship affecting the development processes, in what follows, there is need to identify the *customer real needs* (or say, the *offshore oil company real requirements*) on ROV-based underwater-unmanned manipulator systems.

Identifying real requirements for developing ROV-based underwater-unmanned manipulation systems requires us first to identify the needed set of underwater manipulation *task requirements*, then to identify the needed set of *capability and performance requirements* upon the ROV-based underwater manipulation systems. Thus in next section, I would

- Firstly identify the needed set of underwater manipulation *task requirements*.
- Then identify the needed set of *capability and performance requirements* upon the ROV-based underwater manipulation systems.
- Finally combine these two needed sets to form a two-dimensional *requirement definition matrix*.

## 2.3 Requirements definition for developing underwater manipulation systems and requirement definition matrix

### 2.3.1 Task requirement identification

There is numerous ROV manipulation tasks required for underwater manipulation in today’s offshore oil field. Vartdal (1990) classified the tasks as IMR (inspection, maintenance and repair) and construction tasks. Raine and Forli (1998) classified the inspection tasks as “inspection for flaws” and “inspection for corrosion”. Based mainly on these articles and taking possible future tasks into account, we sum up the underwater manipulation tasks as shown in Tables 2.2 and 2.3. These may be expressed as the task set,  $T$ , with 27 elements:

$$T = \{t_{mpi}, t_{acp}, t_{edd}, \dots, t_{ccb}, t_{cms}\}$$

Where the meanings of  $t_{mpi}, t_{acp}, t_{edd}, \dots, t_{ccb}, t_{cms}$  can be referred to Tables 2.3 and 2.4.

**Table 2.2** Inspection, maintenance and repair tasks

Inspection for flaws	1. Magnetic particle inspection (MPI), $t_{mpi}$	5. Flooded member detection, $t_{fmd}$
	2. Alternating current potential drop, $t_{acp}$	6. Radiography, $t_{rad}$
	3. Eddy current measurement, $t_{edd}$	7. Ultrasonic inspection, $t_{ult}$
	4. Alternating current field measurement (ACFM), $t_{acf}$	8. General visual inspection, $t_{gvi}$
Inspection for corrosion	9. Close visual inspection, $t_{cvi}$	
	10. Cathodic protection potential measurement, $t_{cpp}$	
Other IMR tasks	11. Wall thickness measurement, $t_{wth}$	
	12. Inspection related cleaning, $t_{icl}$	16. Photography, $t_{pho}$
	13. Inspection related grinding, $t_{igr}$	17. Video recording, $t_{vid}$
	14. Inspection related installing, $t_{iin}$	18. C. P. Reading, $t_{cpr}$
	15. Welding, $t_{wel}$	

**Table 2.3** Construction tasks

19. Trenching, $t_{trn}$	24. Sand bagging and sandbag support, $t_{sbs}$
20. Lifting, $t_{lij}$	25. Connection/disconnection, $t_{cdc}$
21. Pulling, $t_{pul}$	26. Construction related cleaning, $t_{ccl}$
22. Bolt handling, $t_{bol}$	27. Construction related measuring, $t_{cms}$
23. Equipment transport, $t_{etr}$	

**Table 2.4** Capability and performance requirement of ROV-manipulators

1. Swimming and obstacle avoidance, $c_{soa}$	7. Tactile and/or force sense, $c_{ifs}$
2. Worksite identification, positioning and referencing, $c_{wip}$	8. Working accuracy, $c_{wac}$
3. Swimming and positioning process visualization, $c_{spv}$	9. Reliability, $c_{rel}$
4. Worksite reach: device size and workspace constraint, $c_{smc}$	10. Cost efficiency, $c_{cst}$
5. Dexterity of manipulation, $c_{dex}$	11. Adaptivity, $c_{adp}$
6. Manipulation load (weight), $c_{mlw}$	12. Intelligence, $c_{int}$
	13. Working process visualization, $c_{wppv}$
	14. Specialized skill for special work, $c_{ssk}$

### 2.3.2 Capability requirement identification

Obviously, any task in the set,  $T$ , requires certain capabilities of ROV-manipulators to be able to accomplish the task. Thus, for identifying the requirements for underwater-unmanned manipulation, other than identifying task requirements as derived above, we also need to identify a relatively complete set of capability requirements for developing ROV-based underwater manipulation systems. With reference to Tables 2.2 and 2.3, and to Taylor (1993), we may identify the capability requirements for developing ROV-based underwater manipulation systems as shown in Table 2.4.

From Table 2.4, we identify the capability requirement set,  $C$ , with 14 elements as:

$$C = \{c_{soa}, c_{wip}, c_{spv}, c_{smc}, c_{dex}, c_{mlw}, c_{tfs}, c_{wac}, c_{rel}, c_{cst}, c_{adp}, c_{int}, c_{wpv}, c_{ssk}\}$$

### 2.3.3 A requirement definition matrix

For sets  $T$  and  $C$ , we have their Cartesian product  $T \times C$  and it equals to  $\{(t, c) | t \in T, c \in C\}$ . To express these in matrix form, we get a table as shown in Table 2.5, which may be called: matrix of *requirement definition* for developing ROV-manipulator systems.

**Table 2.5** A matrix of requirement definition for developing ROV-manipulators for underwater intervention

		1	2	3	4	...	14
		Swimming and obstacle avoidance $c_{soa}$	Worksite identification positioning and referencing $c_{wip}$	Swimming and positioning process visualization $c_{spv}$	Worksite reach: Device size and workspace constraint $c_{smc}$	...	Specialized skill for special work $c_{ssk}$
1	Magnetic particle inspection $t_{mpi}$						
2	Alternating current potential drop $t_{acp}$						
3	Eddy current measurement $t_{edd}$						
...	...						
27	Construction related measuring $t_{cms}$						

By means of Table 2.5, the requirement definition may first be carried out by establishing a (0,1)-matrix, then further defining detailed requirements for every element, which has a value “1” in the (0,1)-matrix. For example, if one has defined a (0,1)-matrix,  $M_{rd}$ , of requirement definition as:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
2	1	1	1	1	1	1	1	1	1	1	1	0	1	0
3	1	1	1	1	1	1	1	1	1	1	1	0	1	0
...	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Where the numbers of rows 1, 2, ..., 27 and numbers of columns 1 through 14 are identical to the Table 2.5. Any element  $e$  given a value of “1” means that for this task the following requirements are to be considered. A “0” element tells us that we do not find this specific requirement of importance for the task given. For instance, element  $e_{1,7}$  has a value of “1”, it means that one requires to develop an ROV-manipulator system for carrying out magnetic particle inspection task (see 1<sup>st</sup> row of Table 2.5) with the capability of tactile and/or force sense (7<sup>th</sup> column of Table 2.5 and item 7 of Table 2.4).

Taking all values of “1”s in the matrix,  $M_{rd}$ , above into account, we may imagine that one’s intention is to develop a ROV-manipulator system, which can carry out magnetic particle inspection, alternating current potential drop inspection, and eddy current measurement (correspondent to rows 1, 2 and 3 of the matrix,  $M_{rd}$ ). And the capability requirements upon the ROV-manipulator systems are to be the swimming and obstacle avoidance, worksite identification and positioning and referencing, swimming and positioning process visualization, worksite reach (device size and workspace constraint), dexterity of manipulation, manipulation load (weight), tactile and/or force sense, working accuracy, reliability, cost efficiency, adaptivity, and working process visualization (refer to Table 2.4 and matrix  $M_{rd}$ ).

Certainly, one must further define the detailed contents of requirements for every element given the value “1” in the (0,1)-matrix. For example, he/she may define element  $e_{1,8}$  of the matrix,  $M_{rd}$ , (i.e., working accuracy of magnetic particle inspection) as: (1) maximum deflection with 100 kg load in any direction < 8 mm, (2) repeatability, 3 mm, and (3) absolute accuracy in robotic mode, from 2 to 10 mm. As well, he/she may define element  $e_{1,11}$  of the matrix (i.e., adaptivity of magnetic particle inspection) as requirements for motion compensation with: (1) translation: horizontal  $\pm 0.10$  mm, vertical  $\pm 0.12$  mm, and lateral  $\pm 0.15$  mm, and (2) rotation: yaw  $\pm 5$  degrees, roll  $\pm 4$  degrees, and pitch  $\pm 8$  degrees (Ricci and Ellingsen, 1992). After all values of “1”s in the (0,1)-matrix have been further defined in detail, the requirement definition for developing a ROV-manipulator is being figured.

## 2.4 Methodological framework for aiding the development of ROV-based underwater manipulation systems

### 2.4.1 Domain identification

In section 2.3, a requirement definition matrix, which aids for defining requirements for developing ROV-based underwater manipulation systems, is given. In this section, a discussion of the development and design issues is performed. That is, we want to identify how many facets that must be taken into consideration to develop and design the ROV-based underwater manipulation system, which will lead us later to find relevant methodologies to aid the development and design of a ROV-based underwater manipulation system.

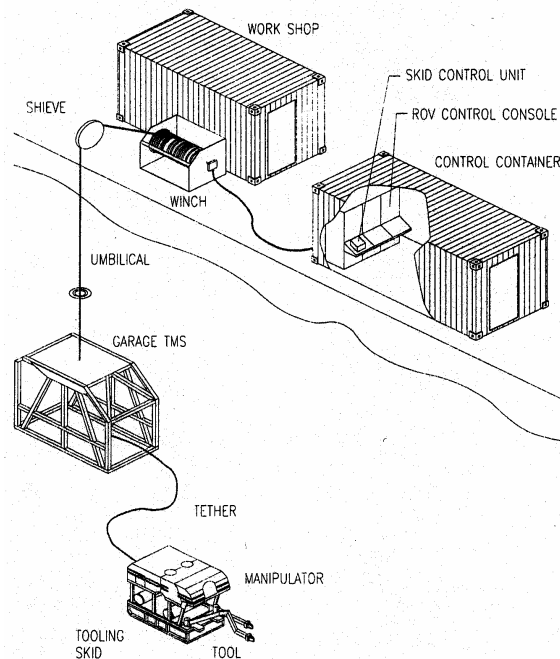


Figure 2.6 A ROV manipulation system

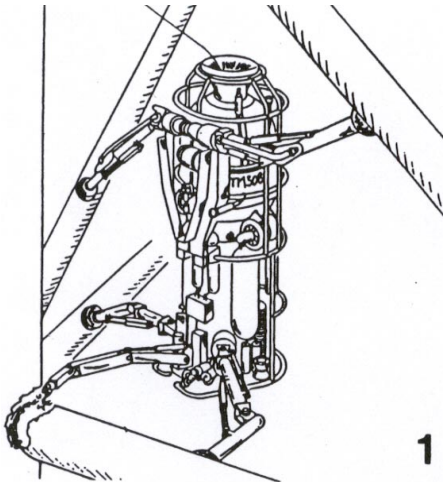
Hallset (1996) described that a ROV-based underwater manipulation system has four basic components (refer to Figure 2.6):

- *ROV and payload.* A ROV is normally equipped with a depth gauge, compass, sonar, and one or more cameras. A work ROV is, as opposed to an observation ROV, usually also equipped with two manipulators: a simple 5-joint gripper for holding the ROV stable while doing work with the more sophisticated 7-joint

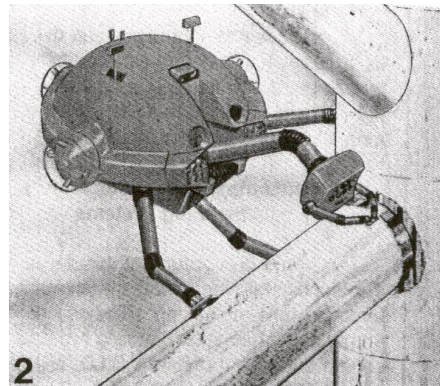


manipulator. The work ROV may have special work packages bolted underneath, called tool skids.

- *Tether management system (TMS)*. This is a garage for protecting the ROV during launch and recovery. The TMS acts as a heavy depressor weight and, thus, decouples the ROV from the surface-vessel motion. A neutrally buoyant tether between the TMS to the ROV is fed out and retrieved as required. The tether provides power and control signals to the ROV. The main disadvantage of the TMS is the limited length of the tether.
- *Surface handling system*. This is a winch for the umbilical and normally an A-frame for launch and recovery of the TMS with the ROV (the A-frame is not shown in Figure 2.6). The umbilical must hold the load of the TMS and carries power and control signals to the ROV and the TMS. An A-frame provides better control than a crane when translating the TMS over the side of a monohull vessel.
- *Surface facilities*. The ROV and the TMS are controlled from within a control container. The ROV has a control console with: a joy-stick for maneuvering the ROV; a joy-stick for controlling the manipulator; push-button panels for power, light, surface winch, and TMS winch; a sonar console; at least one video screen for the ROV pilot camera; one screen for ROV state information such as depth, heading, and alarms. Tool skids are normally equipped with a separate control unit. A separate container functions as a work-shop when modifications and repair are necessary.



**Figure 2.7** Italian TM308 system



**Figure 2.8** Japanese marine robot

Aust et al. (1992) introduced a diverless operated underwater handling system for automated cleaning and NDT-tasks (non-destructive testing) at sub-sea structures (see Figure 2.1). Figure 2.1 shows its main features as follows.

- The control unit, computer and operation base for the ROV and the robot
- The launch system and umbilical for energy, medium, data and information transfer

- The ROV with a rotating and tilting claw to clamp at a structure
- The robot, fixed at the ROV by a moveable arm
- Tools, in a first step for cleaning and NDT-activities.

Aust, Niemann, Boke , Gustmann and Wesche (1995) described sub-sea handling systems with manipulators for IMR-works (inspection, maintenance and repair) at steel structures such as the Italian TM308 system (Figure 2.7) and Japanese marine robot (Figure 2.8). These systems are aimed specially for inspection and maintenance operations at sub-sea steel structures in water depths up to 800 m. The complete task considers the removal of marine growth at welding seams by water jetting or rotating brushes, close visual inspection of the cleaned area by CCD (charge coupled device)-cameras and if necessary the detection of defects by ultrasonic or eddy current sensors. The block diagram (Figure 2.9, Aust et al., 1995) shows in principle the subsystems and components of these concepts.

- The carrier system, which gives mobility and manoeuvrability to the complete system by 4 to 8 thrusters and which allows to fly down to the sub-sea structure.
- Modern sub-systems on board like sonar, gyro and video-systems enable an exact positioning of the vehicle at the working site.
- Energy, signal and information supply is performed by the umbilical, which connects the system with the operating base above the water.
- The actuators and the control units are integrated into the carrier and seawater tight capsulated.
- One of the most important components is the docking system to fix the carrier to the sub-sea structure. The design shows fixing legs with suction type end-effectors or foot-plates.
- Connected to the carrier are 1 to 3 manipulators with 2- finger-grippers, which handles tools and sensors from the integrated magazines. The manipulators with 6 or 7 DOF and high dexterity are handled by the operator with a joystick and monitor information in tele-presence mode. The handling of the manipulators by camera information via the monitor is difficult and exhaustion and concentration problems are obliging.

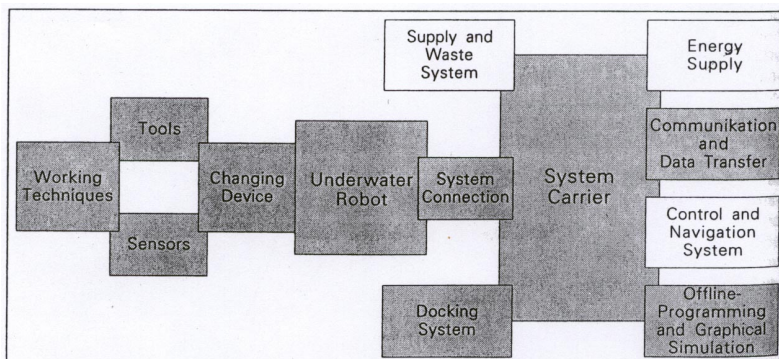


Figure 2.9 Block diagram of a sub-sea handling system

With reference to above discussion, we may category the sub-systems of ROV-manipulators as shown in Table 2.6. They can be taken as R&D domains when developing methodologies and/or advanced technologies in new ROV-manipulator projects.

From Table 2.6, we identify the R&D domain set,  $D$ , with 11 elements as:

$$D = \{d_{osa}, d_{rba}, d_{swm}, d_{man}, d_{tol}, d_{doc}, d_{din}, d_{sen}, d_{pow}, d_{ctl}, d_{com}\}$$

**Table 2.6** R&D domains for ROV sub-sea systems

1. Overall system architecture, configuration and interfacing, $d_{osa}$	6. Docking system, $d_{doc}$
2. ROV body and assembly, $d_{rba}$	7. Database and information system, $d_{din}$
3. Swimming system, $d_{swm}$	8. Sensory system, $d_{sen}$
4. Manipulating system, $d_{man}$	9. Power system, $d_{pow}$
5. Tooling system, $d_{tol}$	10. Control system, $d_{ctl}$
	11. Communication system, $d_{com}$

## 2.4.2 Methodological framework

Combining the above identified eleven R&D domains with the requirement definition matrix discussed in section 2.3, we get a three dimension methodological framework for aiding the development of ROV-based manipulation systems as shown in Figure 2.10.

From Figure 2.10, we may be hinted that firstly one must work in the “problem” plane (task – capability matrix). Then one may work at the “problem solving” dimension to develop subsystems and overall system architecture, configuration and interfacing among subsystems, as shown in Table 2.6.

Certainly, we may define a 4<sup>th</sup> dimension for life cycle based development of ROV-based manipulation systems in addition to Figure 2.10. The life cycle can be composed of following phases: mission, conceptual design, detail design, implementation, operation, maintenance, and retrofitting. As well, a 5<sup>th</sup> dimension for analysis in question space of “what, why, when, where, who and how” can be introduced into the general methodological framework. Introducing this dimension means that while one works at a certain phase of system development life-cycle for certain subsystem, who should work out solutions to answer the “what, why, when, where, who and how” questions. From a methodology-aid point of view, it means that one may generate methods and tools to aid the development. Thus, from above discussion, one may extend the general methodological framework to have five dimensions if he/she thinks it is necessary. However, the scope of this dissertation falls only within the three-dimension methodological framework as shown in Figure 2.10.

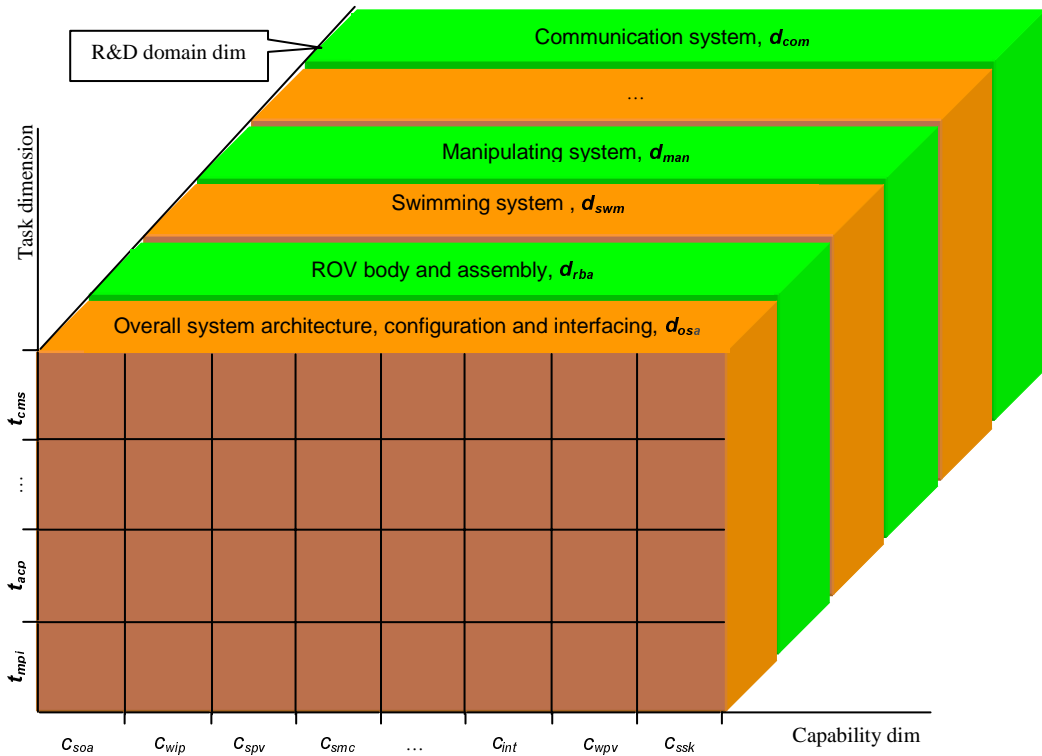


Figure 2.10 Methodological framework for methodology research

## 2.5 Conclusion of this chapter

The increasing rigorous requirements from off-shore oil and gas fields on underwater *unmanned* manipulation in the hostile and unstructured/semi-structured sub-sea environment require constant improvement of the ROV-based underwater manipulation systems. Thus, there is need to develop methodologies for aiding to optimally and formally develop and design new generation underwater unmanned manipulation systems. For meeting such necessity, this chapter first defines a requirement definition matrix, which can be referred for aiding to define complete set of requirements for the new development of underwater manipulation systems. Further by combining the requirement definition matrix with the 3<sup>rd</sup> dimension of R&D domains, an overall concept and methodological framework for underwater manipulator system development is achieved.

Standing on the point of system developers and designers, this methodological framework can be used as a *reference framework* in their developing and designing work, from which they can formally define the requirements and formally link the requirements with the development and design of subsystems.

Standing on the point of methodology researchers, this framework gives methodology researchers a formal overview of what and how many domains they can contribute to develop methodologies to aid system developers and designers in developing and designing new generation of ROV-based underwater manipulation systems.

Recalling what I have defined as my focus in Chapter 1, I would point out that, in the following chapters of this thesis, I would stand at the point of the methodology researchers, focusing my work upon the methodology research for aiding the system developers and designers in developing *manipulating system* (i.e., *manipulator*), which is depicted as one of R&D domains in Figure 2.10.

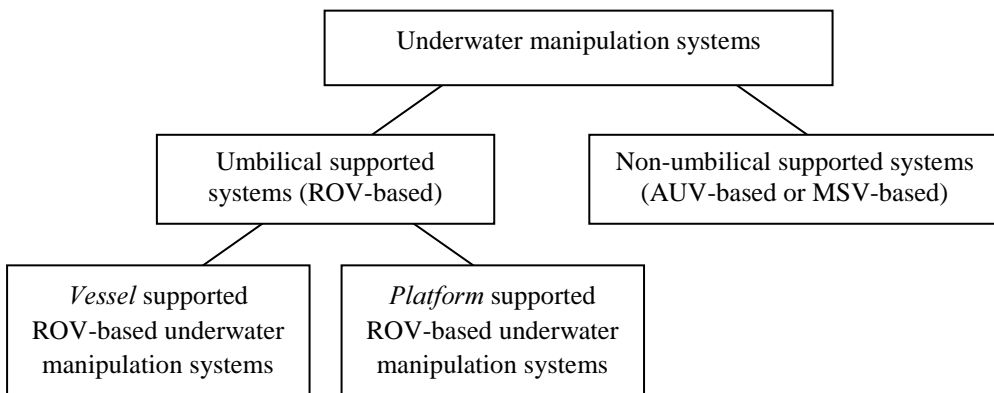


## Chapter 3

### Structure of Underwater Manipulators for Workspace Constraint Environment

#### 3.1 Introduction

Chapter 1 has stated that unmanned underwater vehicles are the key tools for exploring and exploiting resources located at great sea depths. ROVs and AUVs are being employed to perform tasks that range from scientific and environmental data gathering to inspection and assembly of submarine installations. ROVs are generally vehicles tethered to a supporting vessel or platform while AUVs exhibit higher manoeuvrability and do not require the presence of an umbilical. Thus the categorisation of underwater manipulation systems can be categorised into umbilical supported systems (ROV-based underwater manipulation systems) and non-umbilical supported systems (AUV-based underwater manipulation systems or manned submergence vehicle, MSV, based underwater manipulation systems) as shown in the upper part of Figure 3.1 (Solvang, Deng and Lien, 2001).



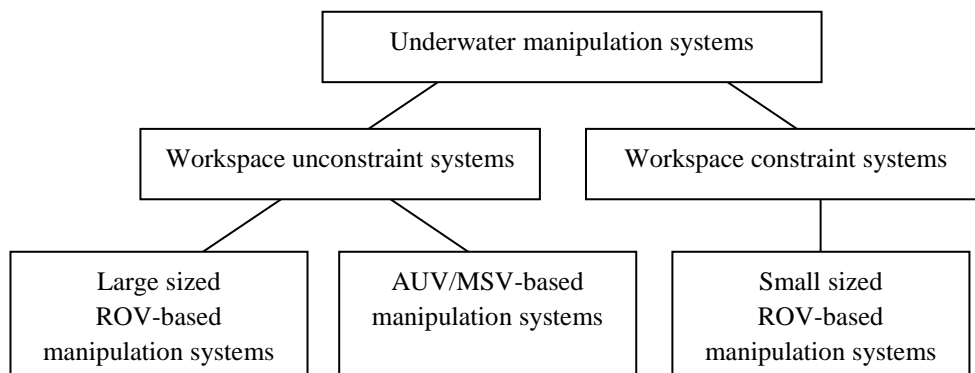
**Figure 3.1** Categorization of underwater manipulation systems according to variant supply and surface support systems

ROV-based underwater manipulation systems can be further categorised into vessel supported (see Figure 2.6) and platform supported (see Figure 2.1), which are shown in the lower left part of Figure 3.1.

Again in Chapter 1, I identified that my R&D preference in this dissertation would be in the area of ROV-based underwater-unmanned manipulation systems. In Chapter 2, I defined a 3D methodological framework (see Figure 2.10). This framework gives methodology researchers a formal overview of what and how many domains they can contribute to develop methodologies to aid system developers and designers in

developing and designing new generation of ROV-based underwater manipulation systems.

Obviously, developing methodologies for all R&D domains shown in Figure 2.10 and Table 2.6 needs huge and enormous efforts by a great number of people to work on. Thus in the conclusion of Chapter 2, I chose only one of the domains, the manipulating system (i.e., *manipulator*) domain, as my methodology research focus.



**Figure 3.2** Categorization of underwater manipulation systems according to work site reachability

The underwater manipulation systems can also be categorized into *workspace constraint* systems and *workspace unconstraint* systems according to the work site reachability (Solvang et al., 2001). This categorization is shown in upper part of Figure 3.2. Such categorization will benefit the study of manipulator structures, because the structures of manipulators are different from each other between the workspace constraint ones and the workspace unconstraint ones.

The *workspace constraint environment* represents an area where there is restriction with regards to available room to conduct the work task. Such areas are typical found in between the construction parts of sub-sea installations. These areas represent special danger since the intervention device (ROV/AUV/MSV) could damage itself or the sub-sea equipment. Entanglement could be crucial, with possible loss of the total equipment or even human life.

*Small sized ROV-based manipulation systems* fall into the category of workspace constraint systems and *large sized ROV-based manipulation systems* fall into the category of workspace unconstraint systems. *AUVs* have for the time being not the capacity to operate in a workspace constraint environment. *MSVs* are also considered as best capable to operate in a workspace unconstraint environment, due to the security issues. This categorization is shown in the lower part of Figure 3.2.

Thus in what follows in this chapter, starting from a discussion on various configurations of underwater manipulation systems, I shall try to identify proper



structures for workspace constraint underwater manipulators and unconstrained ones, with emphasis on workspace constraint underwater manipulators.

## 3.2 Various configurations of underwater manipulation systems

### 3.2.1 Configuration of vessel supported ROV-based manipulation systems

Refer to Figure 2.6, at the upper left of Figure 2.6 (Chapter 2), the umbilical connecting the surface facilities on the vessel with the underwater facilities is shown. So, this is an umbilical supported underwater manipulation system. The umbilical carries power and control signals to the underwater facilities, the garage TMS and the ROV-manipulator (see lower part of Figure 2.6). Notice that the facilities shown in the lowest part of Figure 2.6 is called *ROV-manipulator*, but all facilities shown, including both surface facilities and underwater facilities, is called *ROV-based underwater manipulation system*.

The ROV-manipulator shown in the lowest part of Figure 2.6 is composed of a ROV body part and a manipulator part. The ROV body part consists of a frame equipped with propulsion devices, buoyancy tanks, hydraulic power packs, and one or more cameras. A navigation system including different equipment such as, depth gauge, compass and sonar is mounted onto the ROV frame as well. Sometimes, special purpose tools (dedicated tools for special work tasks) are connected to the ROV frame as a tooling skid. The manipulator part is equipped with one/two manipulators: (1) a 6/7-joint manipulator for doing underwater manipulation work, and (2) a 5-joint manipulator used as a docking device to hold the ROV body steady while doing manipulation work.

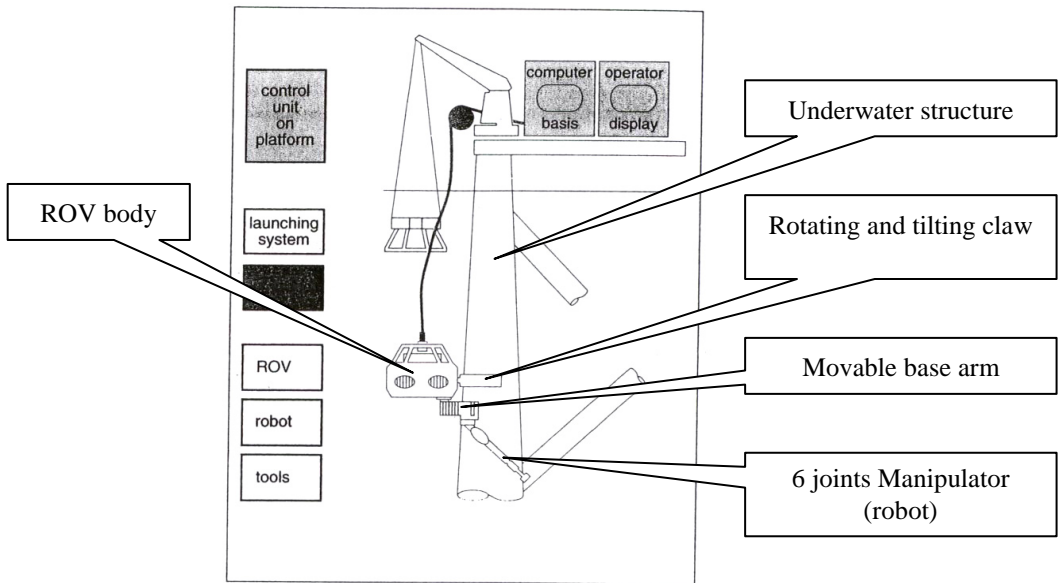
The garage is for protecting the ROV-manipulator during launch and recovery. It acts also as a heavy depressor weight and decouples the ROV-manipulator from the surface-vessel motion. The tether between the TMS and the ROV-manipulator extends from the umbilical to provide power and control signals to the ROV-manipulator. In situation where the work task is to do continuous survey operations (e.g. survey of pipelines) the TMS and the garage is removed. In such a situation the umbilical is connected directly from the supply vessel down to the ROV-manipulator.

At the control container on the vessel, the underwater ROV-manipulator is controlled. The control container consists of (1) a ROV control console, equipped with joy-sticks for maneuvering the ROV-manipulator, (2) push-button panels for power, light, surface winch, and TMS winch for tether, (3) a sonar console, (4) at least one video screen for ROV pilot camera, (5) one screen for ROV state information such as depth, heading, and alarms.

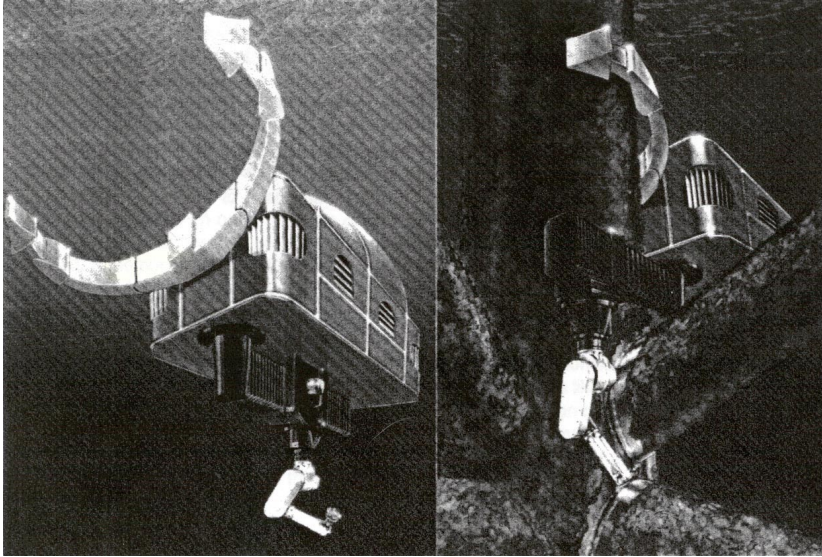
### 3.2.2 Configuration of platform supported ROV-based manipulation systems

Aust, Gustmann, Niemann and Schulhreis (1992) described a platform supported ROV-based manipulation system as shown in Figure 2.1 (Chapter 2). For the sake of convenient illustration, Figure 2.1 is re-depicted here as Figure 3.3. In Figure 3.3, you can see the umbilical (the bold curved line in the middle) connecting the surface facilities on the platform with the underwater facilities. Thus, this is also an umbilical supported underwater manipulation system.

This system is similar to the system described in Figure 2.6, but the 5-joint grabber is exchanged by a rotating and tilting claw (see Figure 3.3), used to clamp the ROV body onto a underwater structure, as a docking device. (A detailed example of the rotating and tilting claw is shown in Figure 3.4.) The system has a 6 joint manipulator attached to a movable base arm (see Figures 3.3 and 3.4) (Aust, Niemann, Boke, Gustmann and Wesche, 1995).



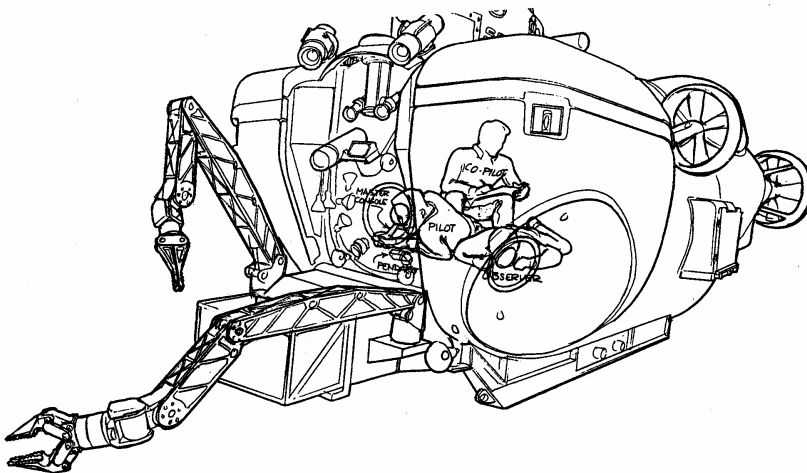
**Figure 3.3** An example of ROV-based underwater manipulation system



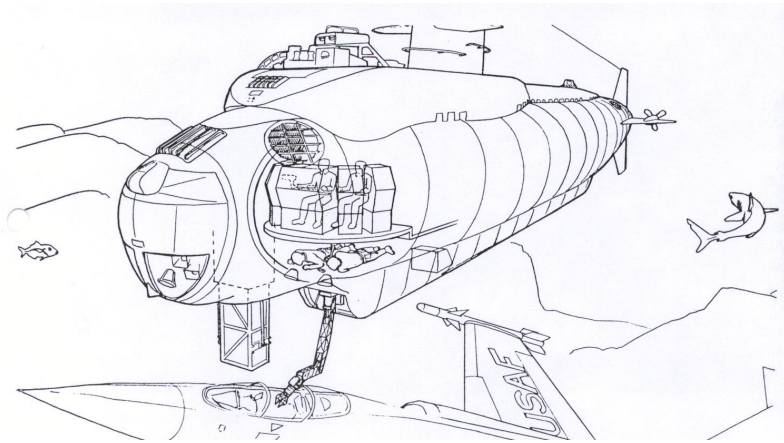
**Figure 3.4** A manipulator is working at a subsea structure with a much stronger but less dextrous docking device (rotating and tilting claw)

### 3.2.3 Configuration of non umbilical supported systems

Regan (1991) described a manned submergence vehicle DSV-4 (Figure 3.5) and a manned submarine NR-1 (Figure 3.6) with manipulators with six degrees of freedom. The manipulators had a master-slave control philosophy. These systems are non-umbilical supported systems according to the categorisation of Figure 3.1.

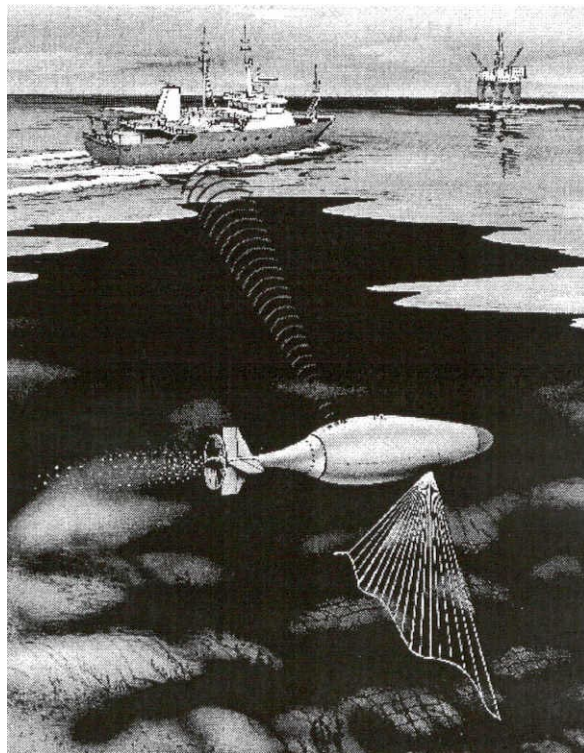


**Figure 3.5** A manned submergence vehicle DSV4



**Figure 3.6** A manned submarine NR-1

The use of manned submersibles is currently limited to a few applications because of very high operational costs, operator fatigue, and safety issues. In recent years, various research efforts have increased autonomy of the vehicle and minimized the need for the presence of human operators. A self-contained, intelligent, decision-making AUV is the goal of current research in underwater robotics (Yuh, 2000).



**Figure 3.7** An AUV-based underwater system

An AUV is an unmanned underwater vehicle that carries its own power source and relies on an on-board computer and built-in machine intelligence to execute a mission consisting of a series of preprogrammed instruction (potentially) modifiable on-line by data or information gathered by the vehicle sensors (Valavanis, Gracanin, Matijasevic, Kolluru and Demetriou, 1997; Kandebo, 1997). Figure 3.7 shows an example of the AUV-based underwater system (Vestgård, Johansen, Klepaker and Størkersen, 2000).

While many underwater ROVs have mechanical manipulators, most of the current AUVs are survey research vehicles without manipulators, only a few of them have performed in deepwater and under ice, so the performance capabilities are still embryonic. Major facts that make it difficult to control underwater robots includes: the highly nonlinear, time-varying dynamic behaviour of the AUV body; uncertainties in hydrodynamic coefficients; the higher order and redundant structure when the manipulator is attached; disturbances by ocean currents; and changes in the center of the gravity and buoyancy due to the manipulator motion which also disturbs the AUV's main body. With the arm attached to the vehicle, the overall system becomes a multi-rigid body system. The vehicle main body continuously moves in water and high performance of arm control, in terms of speed and accuracy, requires highly accurate information about the vehicle position and velocity. Most commercial sensors for vehicle position and velocity do not meet the accuracy requirements of the arm control. Therefore, there are many challenging engineering problems for AUVs with manipulators (Yuh, 2000).

### **3.3 Manipulator structures for working in underwater workspace constraint environment**

In section 3.2, three types of popular underwater manipulation systems is described. From this survey, we may see that the manipulator sub-system plays a central role in the underwater manipulation systems.

The workspace constraint environment manipulating system (i.e. manipulator) is chosen as the research base in this thesis since the total expected work-environment for a manipulator will include areas which are difficult to reach. To gain the necessary flexibility of the manipulator system and to enable successful performance in those areas which are demanding, the chosen manipulator structure must be well adapted to its work environment

Thus, this section includes a discussion of what kinds of manipulator structure is suitable for use in the workspace constraint environment.

The structures of manipulators can be classified as:

- *Serial structure* and
- *Parallel structure*.

The serial structure is the most popular one being used in the underwater manipulation. Figure 3.8 shows one instance of the serial structures (Regan, 1991) where the links and joints are arranged serially. An instance of parallel manipulators is shown in Figure 3.9. This parallel manipulator is composed of a moving platform (MP), a base platform (BP), and four legs. This design provides three degrees of freedom for the MP, namely, heave  $h$ , vertical displacement of MP along  $Z_0$ ; pitch  $\psi$ , rotation of MP about  $y$  axis; and roll  $\phi$ , about  $x$  axis, as shown in Figure 3.9 (Fattah and Kasaei, 2000).

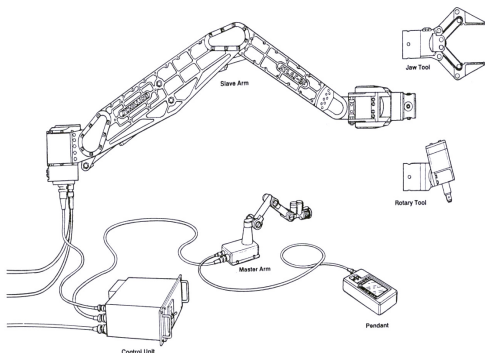


Figure 3.8 A serial structure manipulator

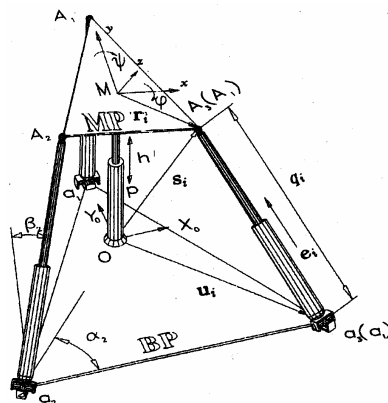
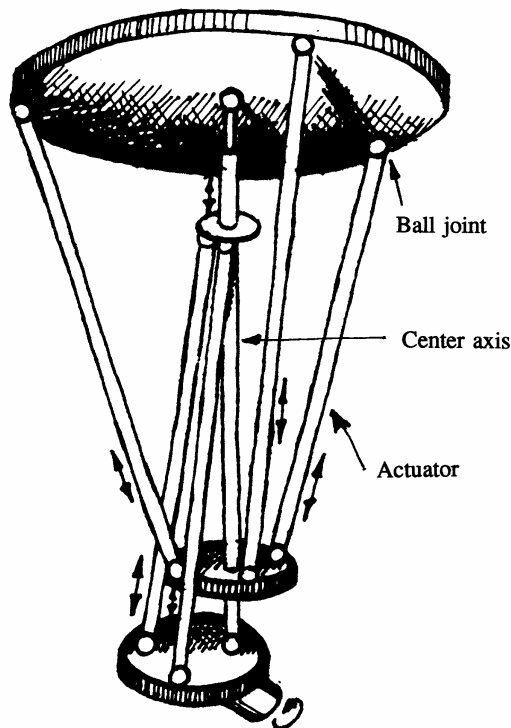


Figure 3.9 An instance of parallel structure

In Figure 3.10, another parallel configuration is shown. This arm is based on the Stewart platform and can among others be found in the Norwegian manipulator system from MULTICRAFT (Thomessen, 1992). This configuration gives an increased strength to the arm. Three legs work in parallel and share the forces from the operation. This arm is strong enough to carry out various types of machining operations as well. From an accuracy point of view, this arm is better than the serial configuration due to an increased stiffness. The stiffness is however not uniform in the whole work area, so deflection will vary within the work envelope. However the main drawback with such a configuration is to require large workspace. The arm is volumetric large compared to its area of operation. A parallel arm has less dexterity compared with a serial arm and its manipulability and reachability is poorer than a similar sized serial construction. In a constraint workspace, the parallel arm is not easy to control in master-slave mode. In the situation where there is a hazard close to one of the parallel joints, it is not so straightforward to predict the movement of the same joint in master-slave mode for preventing collision.

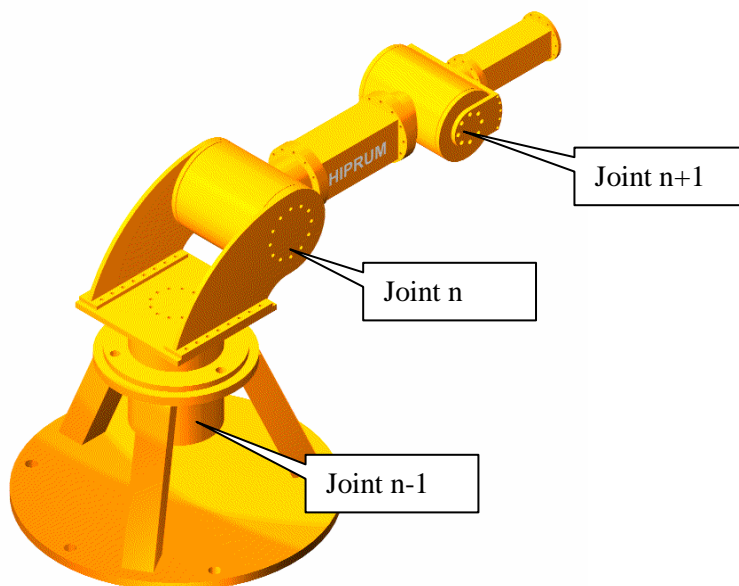


**Figure 3.10** MULTICRAFT 560 Manipulator

A serial design as shown in Figures 3.8 and 3.11 is not as accurate as a parallel-configured arm. This is connected to the decrease in stiffness compared to the parallel arm.

The serial arm is described by that the joint  $n+1$  is connected to joint  $n$  (see Figure 3.11). By movement of joint  $n$ , joint  $n+1$  anchorage point will move as well. An inner joint in the construction has to comply with all the forces induced further out in the arm. The main advantage with such a set-up of the arm is an increased workspace. Its manipulability and reachability is much better compared to the parallel configuration.

The serial arm gives more flexibility regarding the ability to conduct different work tasks, but the serial arm needs to meet the challenge from increased forces and the stiffness must carefully be addressed, especially to meet the challenge for high precision work and accurately following a three-dimensional curve (trajectory) as mentioned in section 1.2.



**Figure 3.11** A serial arm, HIPRUM, with 3 joints

To summarise, the serial arm is selected as our specified structure of workspace constraint underwater manipulators. In the following chapters I shall pay great efforts to solve the serial arm's accuracy problem by developing a precise design methodology for meeting the requirements of high precision work and accurately following a three-dimensional curve (trajectory).

### 3.4 Conclusion of this chapter

In this chapter, the serial arm is chosen as the most flexible one with reference to the work in a workspace constraint environment for performing a variety of different work tasks.

Thus, from Chapter 1 till this chapter, I have

- Identified my research preference in the area of ROV-based underwater-unmanned manipulation systems.
- Developed a three-dimension methodological framework and from which I have selected the manipulating system (i.e. manipulator) domain within the area of ROV-based underwater-unmanned manipulation systems as my methodology research focus.
- Identified the alternatives of workspace constraint manipulators and workspace unconstraint ones. Thus, I further narrow my research focus upon the subject of underwater workspace *constraint* manipulators (i.e. the serial arm underwater manipulators).
- Identified the necessity of developing a precise design methodology for aiding system designers to design a precise underwater serial arm manipulator working in



workspace constraint environment capable of doing high precision work and accurately following three-dimension curve (trajectory).



## Chapter 4

# Design Issues and Precise Design Methodology for Developing Workspace Constraint Underwater Manipulators

### 4.1 Introduction

In the previous chapters, I have identified the necessity of developing a precise design methodology for aiding system designers to design a precise underwater serial arm manipulator, working in a workspace constraint environment capable of doing high precision work and accurately following three-dimension curve (trajectory).

Thus, this chapter starts with a discussion on the design issues of workspace constraint underwater manipulators, and then I shall concentrate my efforts onto the precise design methodology for developing workspace constraint underwater manipulators.

### 4.2 Design issues of underwater manipulators

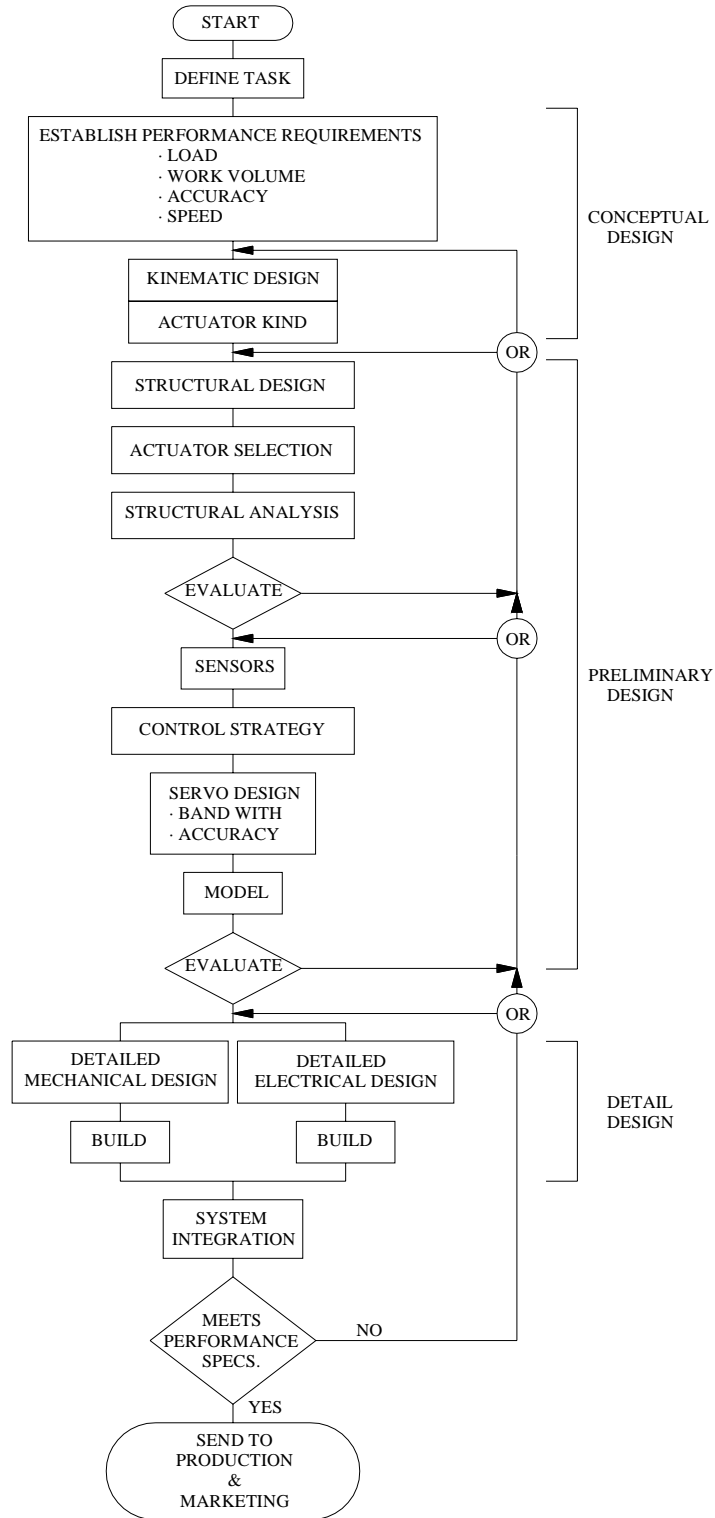
#### 4.2.1 Design procedure for underwater manipulators

Once the underwater manipulation tasks and performance have been defined by means of the requirement definition matrix as described in Chapter 2, design can begin.

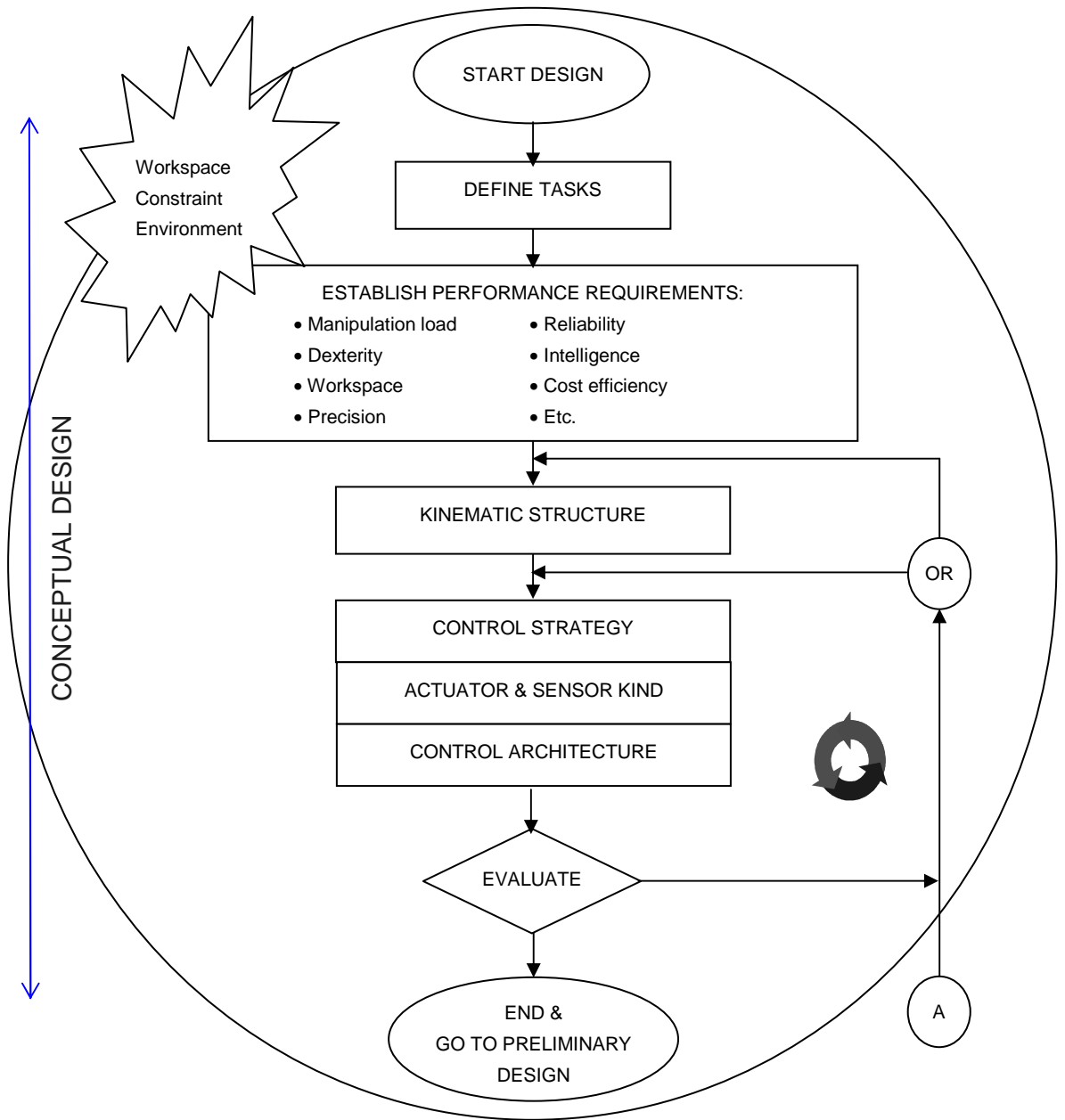
The design process is characterised by the process of bringing together knowledge of available techniques, insight, and the skills of analysis to finally arrive at equipment which satisfy our requirements. A variety of people will usually be a part of the design team; the design will be a result of their background, skills, and interaction. A design procedure is intended to show and guide their interaction. Then a formal procedure can be useful for guiding and managing the design process (Andeen, 1988).

For guiding and managing underwater manipulator design, I will first refer to a flowchart of robot design by Andeen (1988) as shown in Figure 4.1. In this figure a three-phase design procedure with various design tasks is shown. Then a review on this flowchart is given. Based on the review, I will suggest a new flowchart, which I deem is of high relevance in underwater manipulator design.

The three-phase design procedure includes the conceptual design phase, the preliminary design phase, and the detail design phase. In what follows, the tasks of conceptual design, preliminary design, and detail design, given by Figure 4.1, will be reviewed individually in three sub-sections, and some modifications on Figure 4.1 flowchart are given.



**Figure 4.1** Flowchart of robot design (Andeen, 1988)



**Figure 4.2** Conceptual design procedures

#### **4.2.1.1 Conceptual design**

The conceptual design starts with the definition of underwater manipulation tasks (see Figure 4.1). A detailed analysis should determine all manipulation tasks, which are expected to be executed by the manipulator system. These tasks are then the origin of the manipulator performance requirements. Different manipulation tasks create different requirements and it is necessary to identify these requirements and the relationship among them. For aiding the requirement definition, a formal methodology by using the requirement definition matrix has been given in Chapter 2 of this dissertation.

When the requirements are defined, the kinematic design follows. The kinematic design is of high importance in a manipulator project. What kind of structure is best to fit the requirements, parallel or serial? How many degrees of freedom? Linear or rotary joints? Selection of the manipulator layout and configuration? However, the kinematic design is by no means the only task in the conceptual design phase.

Conceptual design, as its name implies, aims to figure out the general concept or overall view of the future system being designed, with emphasis at strategic matters. From such point of view, looking at upper part of Figure 4.1, I find that the design tasks for the conceptual design phase, where the kinematic design and actuator kind are the only design tasks, are not enough to give out a general concept or overall view of the future system being designed.

I deem that after kinematic design, not only *actuator kind* is considered, but also the *sensor kind* must be considered as well, so as to figure out an overall view of both actuating and sensing provisions for the future system. Further, a *control view* of the future system should also be conceived in this phase together with the *kinematic view* so as to let people having a general concept and overall view of the future system. Thus, I would like to define the design tasks for the conceptual design phase as shown in Figure 4.2.

In Figure 4.2, I combine the control strategy task with actuator kind and sensor kind to conceive the system control behaviour of how to collect data from sensors and how to control the actuators. From here a control architecture will be given. Thus, a kinematic structure combining with its control architecture gives out a general concept and overall view of the future system being designed.

In addition, compared with the conceptual design phase shown in Figure 4.1, I add an "EVALUATE" activity in lower part of Figure 4.2 as the final step of the conceptual design phase due to the necessity of evaluating the relevancy of the conceptual design result. If it is not relevant enough after an evaluation, then iteration for conceptual design is required, which is also expressed in Figure 4.2.

Remember, the major decisions are made in the conceptual design phase. Such a strategy is natural since major decisions should not be made when it is too late to make the necessary changes.

In the next paragraph, I will review the tasks of the preliminary design given in Figure 4.1.

#### **4.2.1.2 Preliminary design**

Refer to the middle part of Figure 4.1 where the tasks of the preliminary design are shown. As discussed in section 4.2.1.1, I suggested to move the *control strategy* task from the preliminary design phase to the conceptual design phase. In the preliminary design phase (Figure 4.1) I again deem that the *actuator selection* task is a kind of detail design task and it should be moved to the detail design phase, as a possible sub-task within the detailed *electrical design* task (see the lower part of Figure 4.1).

In middle part of Figure 4.1, the *sensors* tasks can be divided into two subtasks of *sensor kind* and *sensor selection*. The former has been moved to the conceptual design phase as shown in Figure 4.2. And the latter should be moved to the detail design phase as a sub-task of detailed electrical design task with the same reason as that of moving the actuator selection task.

For the *model* task shown in the middle part of Figure 4.1, I agree that it is a necessary task in the preliminary design phase. But I prefer to use the name *servo design analysis* to match its previous task, *servo design*.

To summarise above discussions, I have figured a new flowchart for the preliminary design phase as shown in Figure 4.3.

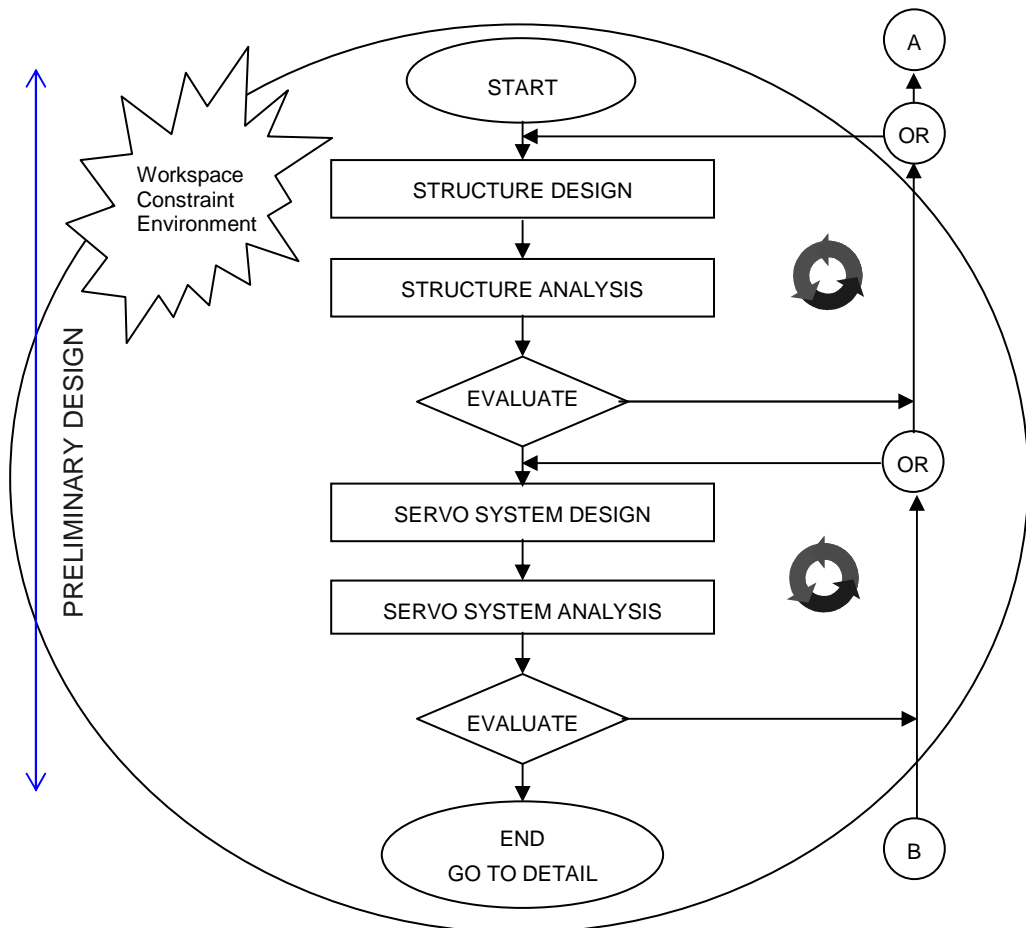


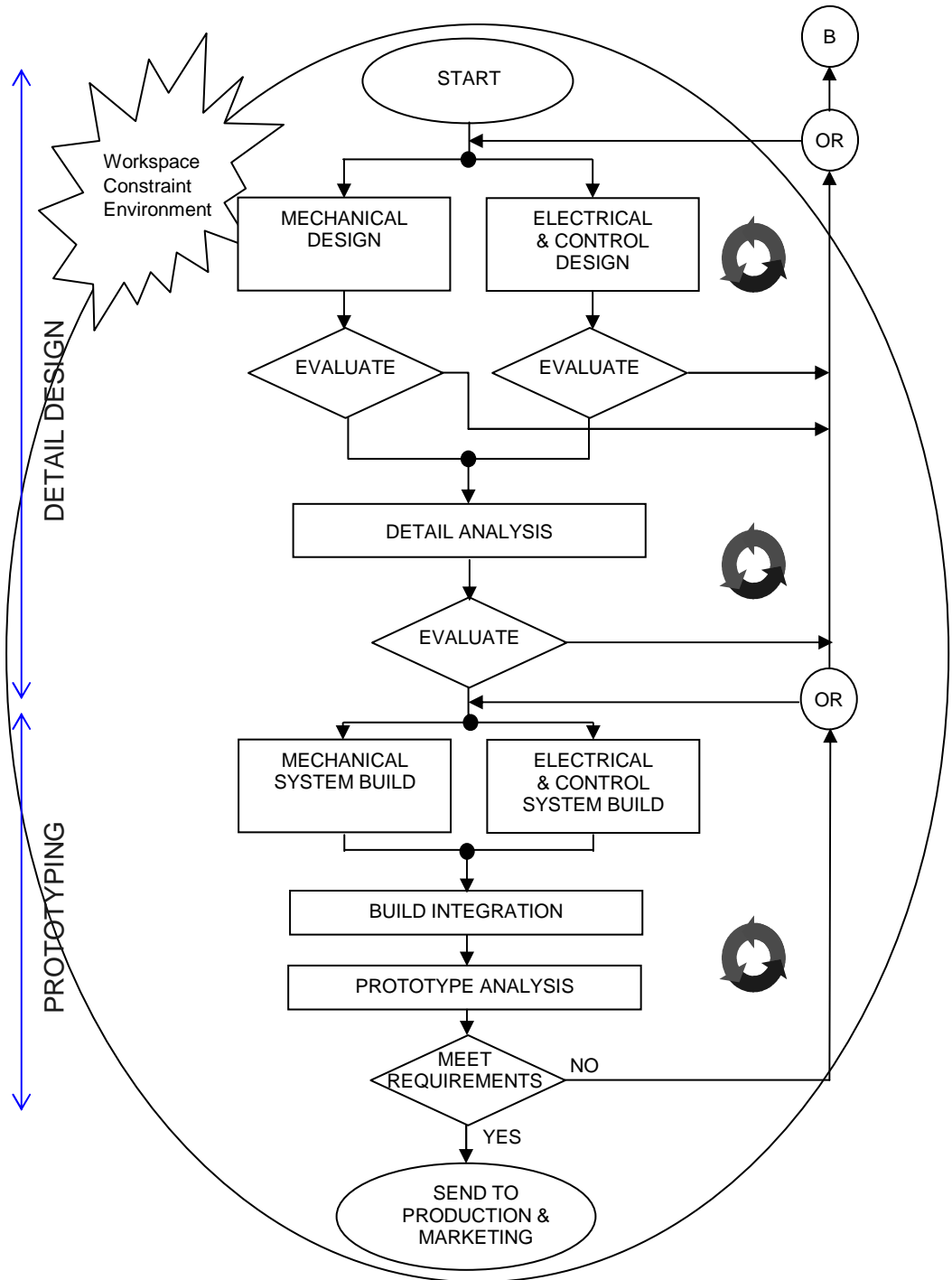
Figure 4.3 Preliminary design procedures

#### 4.2.1.3 Detail design and prototyping

The “detailed design” is the final design phase in the design procedure and in this phase we will fully verify if our ideas made in the conceptual design phase and the preliminary design phase are possible to achieve within our project setting. In this final phase of the design process, all constructional details are made.

For the detail design, I have also recommended a flowchart, other than Figure 4.1, as shown in Figure 4.4. Comparing Figure 4.4 with the lower part of Figure 4.1, an important additional task entitled “*detail analysis*” is added, which is not present in the Figure 4.1





**Figure 4.4** Detail design and prototyping procedure

The reasons of adding this task are as follows. At the preliminary design phase, the *structure design* task (see the upper part of Figure 4.3) is based on temporarily assumed dimensions and presumed cross section of beams with simplified shapes and without exact geometric tolerances and shapes. At this point, we have less details. Thus, the subsequent task of the structure design, *structure analysis* (see upper part of Figure 4.3), which is a task that calculates the end-effector error, is carried out based on *inexact* data. Thus, even if the calculated error of the end-effector is within the precision requirement range specified in the requirement definition task in the conceptual design phase, it does not guarantee that the final real error of the end-effector after the detail design phase can be kept within the precision requirement range.

Looking at the upper part of Figure 4.4, we see that after the detail mechanical design, the exact mechanical assembly, shapes, dimensions including geometrical tolerances of components are finally determined. As well, after the detail electrical and control design, the real actuators and sensors are selected. Thus, the device errors can only at this moment exactly be identified. Therefore, as shown in Figure 4.4, a *detail analysis* task is needed to work out a more exact error analysis for the detail designed manipulator for meeting the precision requirement specified in the conceptual design phase. This is especially necessary while precise design is required for precise underwater serial arm manipulator working in the workspace constraint environment.

In the lower part of Figure 4.4 a new section is introduced, namely the “*prototyping*” section. This section is not specifically named in Figure 4.1 but the tasks *build* and *system integration* indicate such a stage. In the Figure 4.4 the tasks *mechanical, electrical and control system build* and *build integration* replaces the tasks *build* and *system integration* from Figure 4.1.

In addition a new task is added in Figure 4.4 called *prototype analysis*. The *prototype analysis* task is the final stage before manufacturing. This task includes performance measurements. However, in the case where the performance measures are out of our requirements the *prototype analysis* task should also include a methodology which enable the designer to point directly to the error source(s). This methodology will be referred to as an error mapping methodology

#### **4.2.2 A further discussion on design issues of underwater manipulators**

The above discussions have resulted in a new flowchart for designing underwater manipulators as shown in Figures 4.2, 4.3 and 4.4. Naturally, one would think that it is necessary to develop various design methods for aiding the various design tasks covered in these flowcharts. However, many methods already exist and are successfully used in today’s on-shore robot design. Such methods can often directly be transferred into underwater manipulator design. But methodology for some of the task in the new flowcharts of underwater manipulator design is not good enough or does not even exist. So, it requires developing new methods. Thus, in this section, I would follow the sequence of the flowchart to discuss which task that can borrow

methodology from on-shore robot design area and which task may require development of new methods.

Starting the discussion at the top of Figure 4.2, the methodology for requirement definition task for the development of underwater manipulation systems. A methodology has been developed in Chapter 2 where a matrix method is presented.

For the *kinematic structure* identification task, a discussion, especially for the workspace constraint underwater manipulators, has been carried out in the Chapter 3. For other kinematic structures, several existing literature can give the necessary guidance. Refer to literature such as Andeen (1988), Rivin (1988), McKerrow (1991), and Nof (1999).

For the task of identification of *actuator & sensor Kind*, we can find a number of literatures in which the methods are available such as from Andeen (1988), McKerrow (1991), Mooring, Roth and Driels (1991), Ulrich and Yoerger (1991), Yoerger, Schempf and Dipietro (1991) and Nof (1999).

For the *control strategy* and *control architecture* tasks, as shown in Figure 4.2, it is necessary to have a particular consideration, due to the special underwater environment. I will specially discuss control strategy and control architecture in section 4.3.

Looking at Figures 4.3 and 4.4, the methods for *structure design*, *structure analysis*, *servo system design*, *servo system analysis*, *mechanical design* and *electrical & control design* can again be borrowed from the on-shore robot technologies such as from Andeen (1988), Rivin (1988), McKerrow (1991), Lien (1993) and Nof (1999).

However, I did not find a complete methodology existing nowadays for the *detail analysis* task in the detail design phase as shown in the middle of Figure 4.4. Therefore, I have worked for developing a new methodology, which I will discuss in section 4.4. This methodology can actually also be used as the analysis method for the *structure analysis* task in preliminary design phase (see Figure 4.3). I have found that the *structure analysis*, by using this new methodology, possesses a more precise result than those by using conventional methods. This will be further discussed in the coming sections.

The new *prototype analysis* task of Figure 4.4 is divided into two natural parts mainly, performance measurement and error mapping, as described in the previous section. Performance measures are fully described by the ISO 9283 (1998) standard. However the error mapping methodology is not found in literature, so an error mapping methodology is outlined in Chapter 6.

To sum up the above discussions, I have found it natural to further discuss the following two problems in underwater manipulator design.

- Control strategies and control architectures
- Methodology for analysis of precise design

In manipulator prototyping it will be necessary to discuss the ideas of a

- Error mapping methodology

The error mapping methodology is presented in Chapter 6, while the two design issues will be discussed in this chapter.

### **4.3 Control strategy and control architecture for underwater manipulation**

In this part of this dissertation I will discuss the control strategies and control architectures of underwater manipulators.

We may identify an underwater manipulator to act as a “telerobotic” system working in the underwater environment. A telerobotic system is a system that is capable of performing as either a *telemanipulator* (*master-slave* mode or *joystick control* mode) or with the manipulator performing alone as a *robot* (Kress, 2002).

Hence, I would discuss the control strategies and architectures for

- *Master-slave mode*
- *Joystick control mode*
- *Robotic mode (i.e., computerized high-accuracy trajectory)*

#### **4.3.1 Master-slave mode**

##### **4.3.1.1 Master- slave strategy**

Conventionally, the underwater manipulator works according to a *master-slave* control principle. Both the ROV and the manipulator are commanded from the surface. Once the ROV has reached the working site, docking arms are employed to fix it to the underwater construction. The operator then commands the manipulator in *master-slave* mode, making use of the images from remote TV-cameras, as visual feedback devices, to determine the manipulator trajectory. These systems require operator skill, because of the low level of automation introduced. The execution of complex tasks is extremely tiring and time consuming, or even not feasible. The visual feedback is poor and insufficient in most operations (Nicolodi et al., 1990).

In a conventional system, signals from a master arm (a scaled down, kinematic model of the slave manipulator arm) pass directly into the control system electronics. These signals control the manipulator joints directly. For example increasing a joint angle on

the master will proportionally increase the corresponding joint angle on the slave (Broome and Langrock, 1994).

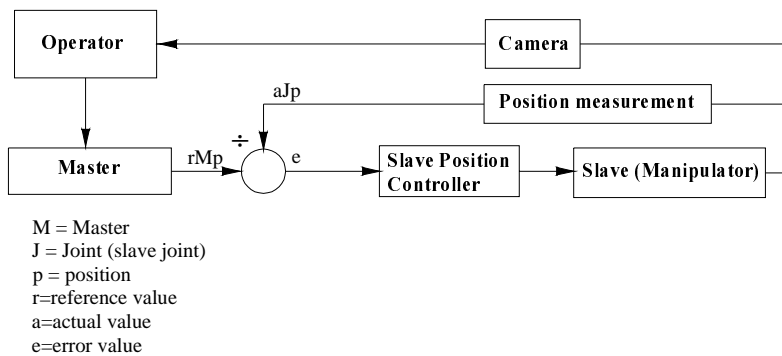
The selected master-slave strategies include:

- a.) *Position control*
- b.) *Position-Position control*
- c.) *Position-Force control*
- d.) *Bilateral control*

These systems will be reviewed in the following sub-sections of master slave control architectures.

#### 4.3.1.2 Master- slave control architectures

##### a.) Position control



**Figure 4.5** Position architecture

In the control architecture in Figure 4.5 the operator activates the master controller and a reference position  $rMp$  is resulting. The actual slave joint position  $aJp$ , arising from measurement inside the slave manipulator joints, is subtracted from the reference position  $rMp$ . The resulting signal  $e$  describing the difference (or error) between the master and the slave position is fed into the slave position controller which drives the slave (manipulator) to the desired joint position.

In this position control architecture the operator watches camera feedback from the work site to decide to which position he will move the slave (manipulator) joint.

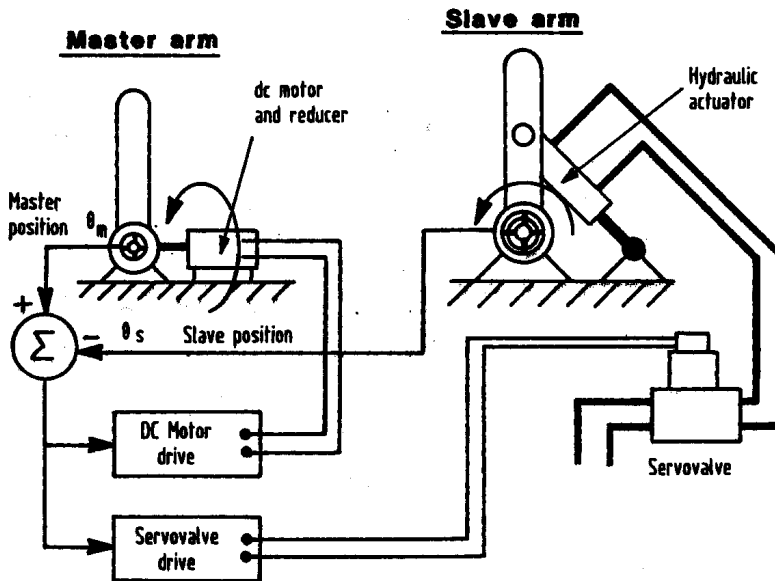
##### b.) Position-Position control

In a Position-Position control system the slave is commanded to the position of the master, as the unilateral position system described in section a.), but now the master is also commanded to drive to the position of the slave. If the actuator forces are

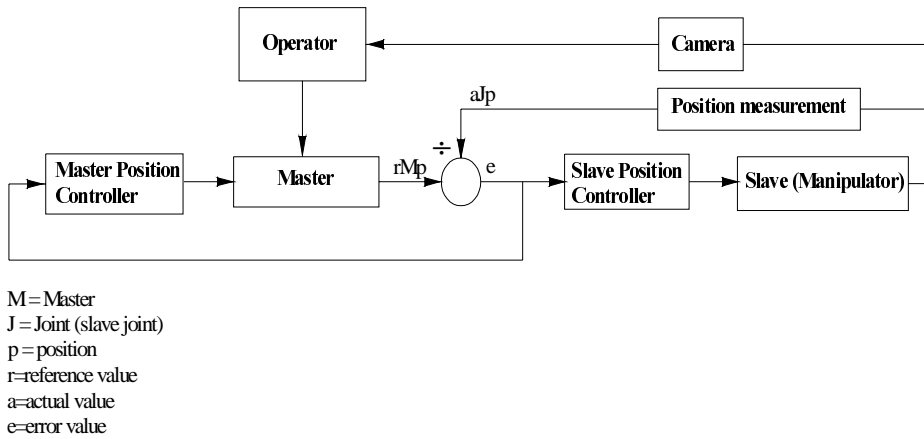
proportional to the position error signal then a representative force reflection is achieved. This is a mechanically simple system requiring no force transducers but has disadvantages of low accuracy and resolution together with the problem that any movement will cause an error signal and hence a force which results in an unwelcome viscous feel to the movements (Taylor, 1993). Figure 4.6 shows the physical layout of such a system.

The Position – Position control system can be realised with the following architecture, as shown in Figure 4.7.

In the control architecture as shown in Figure 4.7 the operator activates the master controller and a reference position  $rMp$  is resulting. The actual slave joint position  $aJp$ , arising from measurement inside the slave manipulator joints, is subtracted from the reference position  $rMp$ . The resulting signal  $e$  describing the difference (or error) between the master and the slave position is fed into the slave position controller which drives the slave (manipulator) to the desired joint position. At the same time, the error signal  $e$  is sent back to the master, via the master position controller. The master is commanded to drive to the position of the slave. This will create a reacting force upon the operator hand. If the master forces is proportional to the position error signal  $e$  a representative force reflection is achieved.



**Figure 4.6** Layout of a position-position control system (Taylor, 1993)



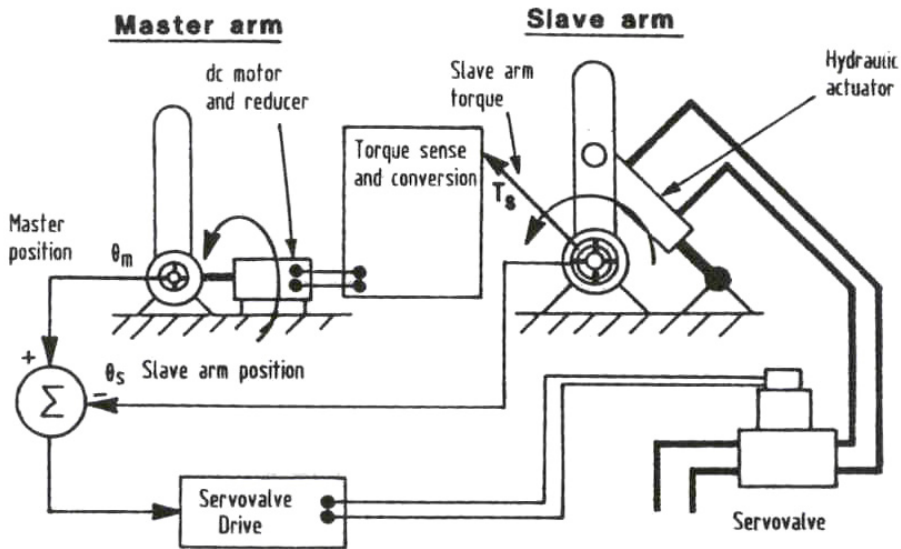
**Figure 4.7** Position-position control architecture

The master arm must, in this control architecture, have a controllable motor located in every joint.

The operator watches camera feedback from the work site to decide to which position he will move the slave (manipulator) joint.

**c.) Position-Force control**

A position/force control system (Figure 4.8) works by driving the slave to the position of the master and driving the master with forces proportional to those encountered at the slave. The “viscous” feel problem is avoided by this method but force transducers are required at the slave and problems of stability can result due to the master and slave being controlled by different methods. System gains and hence overall performance may therefore be limited (Taylor, 1993).



**Figure 4.8** Layout of a position-force control system (Taylor, 1993)

In the Position-Force control architecture, as shown in Figure 4.9, the operator activates the master controller and a reference position  $rMp$  is resulting. The actual slave joint position  $aJp$ , arising from measurement inside the slave manipulator joints, is subtracted from the reference position  $rMp$ . The resulting signal  $e$  describing the difference (or error) between the master and the slave position is fed into the slave position controller which drives the slave (manipulator) to the desired joint position. At the same time, torque measurement from all of the joints  $aJt$  is sent back to the master, via the master torque controller. This will create a reacting force upon the operator’s hand.

The master-arm must, in this control architecture, have a controllable motor located in every joint. The slave or manipulator should have both position and torque sensors.

The special benefit with this control architecture is that the operator will have a sense of feeling when the slave touches any object and the joint torque increase.



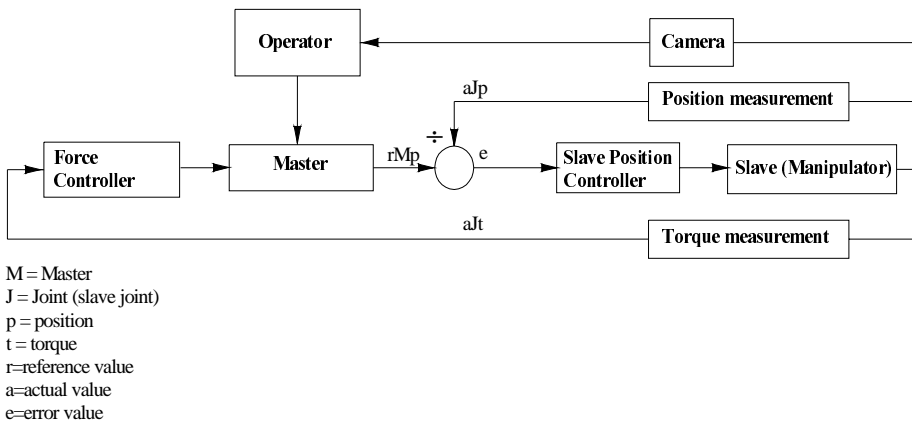


Figure 4.9 Position-Force control architecture

The operator relies on camera feedback from the work site to decide to which position he will move the slave (manipulator) joint.

#### d.) Bilateral control

Bilateral force reflection systems (Figure 4.10) are essentially symmetrical in that a force applied at the slave produces a force at the master and vice versa, with joint positions and velocities also included in the control algorithms. Gravity compensation may also be incorporated to reduce operator fatigue problems. Such systems work very effectively but are complex and expensive (Taylor, 1993).

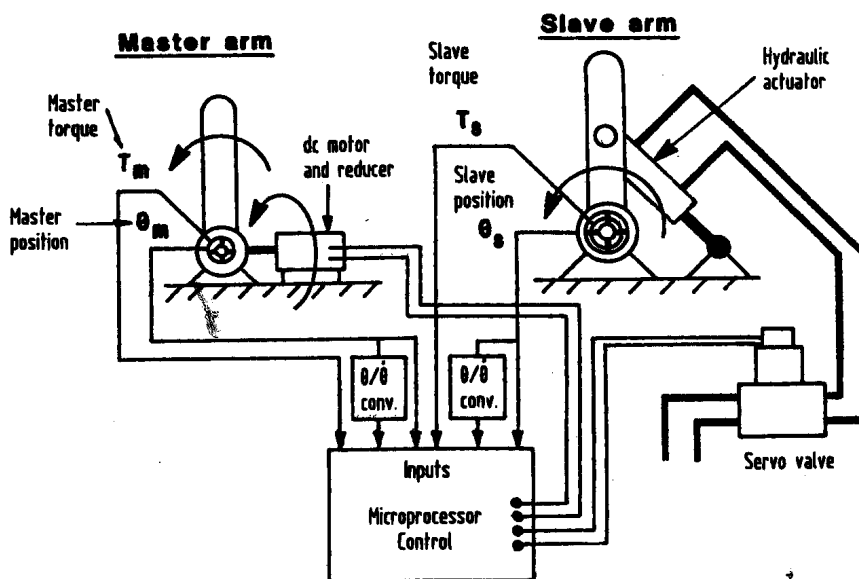


Figure 4.10 Layout of a bilateral control system (Taylor, 1993)

The architecture in Figure 4.11 is the most advanced conceptual architecture of all structures described previously. The idea behind this structure is that the slave will copy the same position  $rMp$ , velocity  $rMv$ , and torque  $rMt$  as the master. At the same time the master gets feedback from the slave position  $aJp$ , velocity  $aJv$ , and torque  $aJt$  is transferred to the operator's hand.

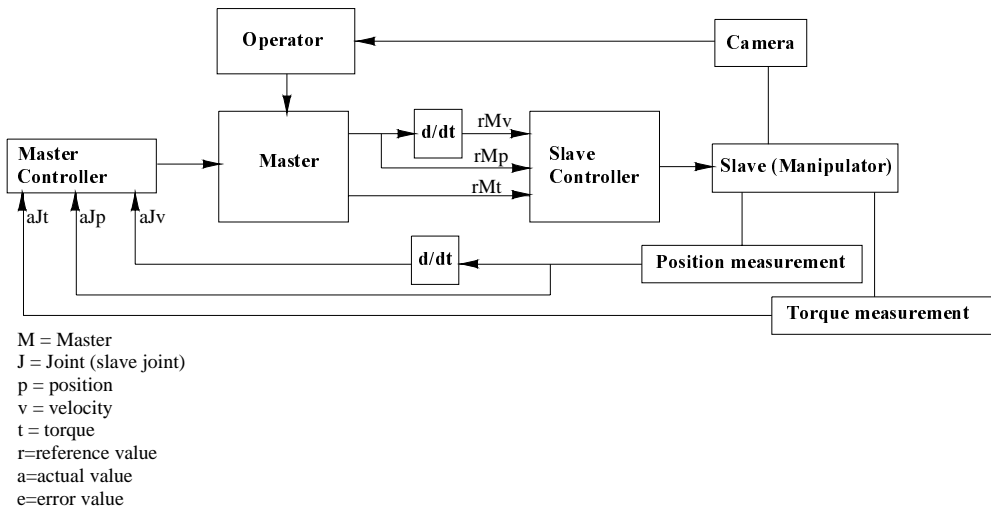


Figure 4.11 Bilateral control architecture

The operator watches camera feedback from the work site to decide to which position he will move the slave (manipulator) joint.

4.3.1.3 Summary of conceptual master- slave control design

From the above discussion, I depict the master-slave control strategy of underwater manipulators as shown in the middle of Figure 4.12. In left and right parts of Figure 4.12, the other control strategies are shown, which will be discusses in following sub-sections.

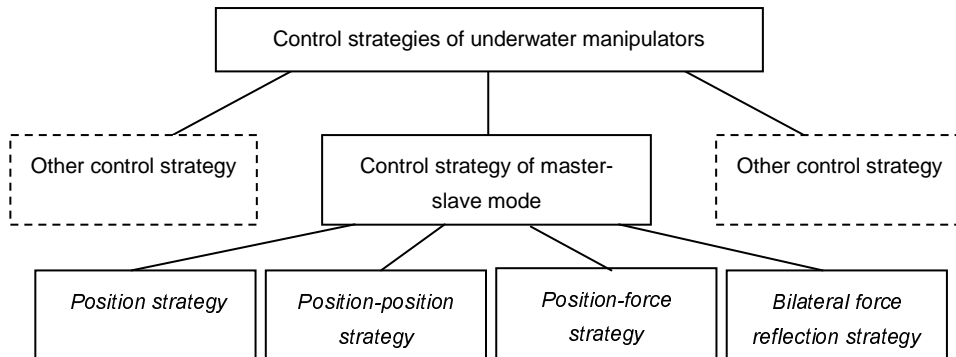


Figure 4.12 Control strategy of master-slave mode for underwater manipulators

By looking into the different control architectures presented I may identify the common software and hardware components which constitute the strategies.

The control architectures a.) b.) c.) and d.) presented in paragraph 4.3.1.2 is realized by introducing a common position loop at the slave (manipulator) side. This will require that every joint has an actuator and a position measurement system. Architecture c.) and d.) require torque measurements in addition to the position measurement. This leads to the conclusion that the slave manipulator should have a position loop inside every joint, if we want to realise the master slave strategy. For more advanced strategies, such as c.) and d.), we need joint torque measurements as well. Another conclusion is that the master will be more advanced with the more advanced control strategies.

All of the strategies have a nature that requires an active operator inside the control loop.

### **4.3.2 Joystick mode**

#### **4.3.2.1 Joystick strategy**

As stated in the last paragraph master-slave control strategies have until now been used to a large extent as the control strategy. Advanced master slave technologies require an advanced and specialised/expensive master system. In the case of joystick technology the advancement in the PC industry, among others, has made the joystick technology commercially available at low cost. However, cost and availability is not the only reasons for the usage of joystick technology, they have a long history from onshore robotic systems.

In onshore robotic systems a classical layout joystick is designed and used as a stick movable in three directions. Such a layout is very suitable for operation in the Cartesian space. Here the three Cartesian directions (X,Y,Z) are coupled to the belonging joystick axes. For maneuvering in the joint space the joystick axes are easily connected to selected joints, in groups of 3 joints, where the selection of groups is done by pushbuttons.

Thus, in the coming sub-section possible joystick architectures is identified and control architectures is developed.

The selected joystick control strategies are:

- a.) *Joint coordinate rate control (open servo loop)*
- b.) *Joint coordinate rate control (closed servo loop)*
- c.1) and c.2.) *Joint coordinate position control (closed servo loop)*
- d.) *Base coordinate rate control (closed servo loop)*

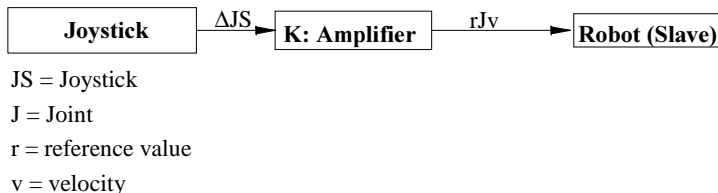
- e.) *Base coordinate position control (closed servo loop)*
- f.) *End-effector coordinate rate control (closed servo loop)*
- g.) *End-effector coordinate position control (closed servo loop)*
- h.) *External coordinate rate control (closed servo loop)*
- i.) *External coordinate position control (closed servo loop)*

These systems will be reviewed in the following sub-sections of joystick control architectures.

### **4.3.2.2 Joystick control architectures**

#### **a.) Joint coordinate rate control (open servo loop)**

The joystick rate control strategy is a low-cost control strategy for underwater manipulator control.



**Figure 4.13** Joystick joint coordinate rate control (open servo loop) architecture

The operation principle of the architecture (Figure 4.13) is as follows:

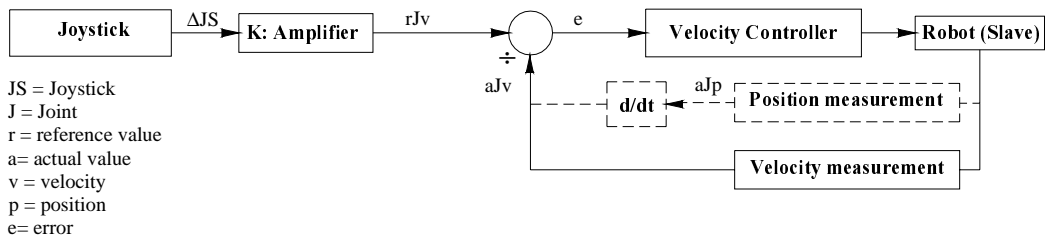
$\Delta JS$  represents the movement of joystick. The amplifier  $K$  transforms the joystick activation into a proper reference joint velocity  $rJv$ . This signal is fed to the joint actuators which start to move (maximum 3 at the same time).

The operator watches camera feedback from the work site and controls the joystick to obtain the desirable joint speed.

The special benefit from this architecture is that it is possible to have an almost direct access to the robot (slave) joints. If other parts of the control system fail (hardware or software errors) it will still be possible to move the manipulator to a safe area.

The system has no servo loop feedback. Thus, it is not necessary with any joint measurement system.

A drawback with this strategy is that the movement of the end-effector will describe circular arcs when the joint is of a rotary type.

**b.) Joint coordinate rate control (closed servo loop)****Figure 4.14** Joystick joint coordinate rate control (closed servo loop) architecture

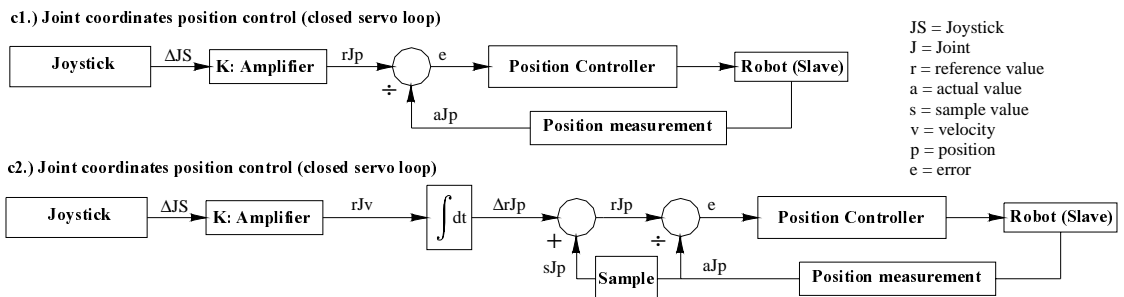
By introducing the position or velocity joint measurement in every joint we will gain the benefit from a closed servo loop. The principle of operation is as follows:

$\Delta JS$  represent the movement of the joystick. The amplifier  $K$  transforms the joystick activation into a proper reference joint velocity  $rJv$ . This signal is compared with the measured actual joint velocity  $aJv$ . The difference between the reference and the actual value is the error signal  $e$ . This signal is fed into the velocity controller for every joint. The robot (slave) joint starts to move.

The operator watches camera feedback from the work site and controls the joystick to obtain the desirable joint speed. However, in this case, helped by the velocity controller.

The servo feedback loop can be realised either with a velocity measuring device or a joint position sensor combined with a differentiator.

The special benefit from this architecture is that the velocity controller will provide more conformity between the joystick activation and the joint velocity. A drawback is the possible unstable feedback loop arising from improper controller parameters.

**c.1) and c2.) Joint coordinate position control (closed servo loop)****Figure 4.15** Joint coordinate position control (closed servo loop) architectures

In both control architectures shown in Figure 4.15 the servo loops are position controlled.

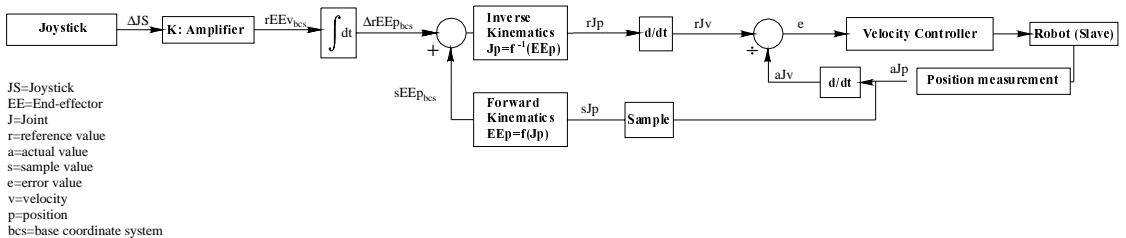
In c1.) architecture the joystick motion is directly coupled to the joint position. Joystick maximum and minimum travel correspond to the actual joint maximum and minimum travel. The joystick activation  $\Delta JS$  is transformed directly into the reference joint position  $rJp$ . The difference  $e$  between the reference and the actual joint position is fed into the servo loop where a position controller activates the joints towards the desired position.

One drawback with this control architecture is that if the joystick has a limited range of operation compared with the joint maximum and minimum travel it will be difficult for an operator to move the joint exactly to the desired angle. This problem may be omitted by introducing the c2.) control architecture.

In c2.) joystick activation  $\Delta JS$  is coupled to the reference joint speed  $rJv$  via the amplifier  $K$ . The joystick velocity is integrated over a time interval and is thereby transformed into a position increment  $\Delta rJp$ . This increment is added to the joint position  $sJp$  ( $s$  = sampled value with a defined sampling rate). The result is a new reference joint position  $rJp$ . This  $rJp$  signal is compared with the measured actual joint positions  $aJp$ . The difference between the reference and the actual value is the error signal  $e$ . This signal is fed into the position controller for every joint. The robot (slave) joint starts to move.

In both control architectures c1.) and c2.) operator watches camera feedback from the work site and control the joystick to obtain the desirable joint position.

**d.) Base coordinate rate control (closed servo loop)**



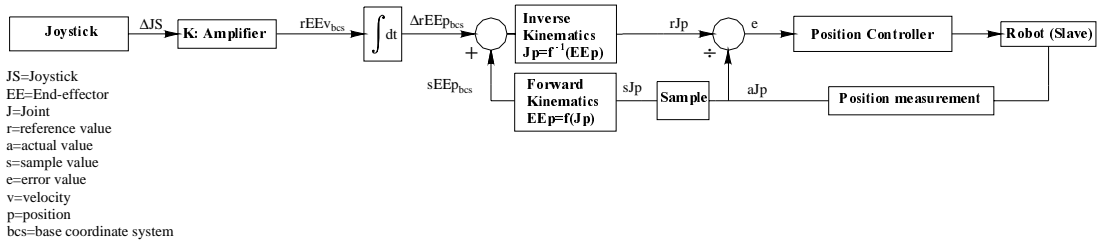
**Figure 4.16** Base coordinate rate control (closed servo loop) architecture

The general idea behind this control architecture is to enable the joystick to control the end effector rate in the base coordinate system. This is especially useful when the work task has an “axis of operation” which is oriented in the same direction as one of the manipulator base coordinate system axes.

In Figure 4.16 the joystick activation  $\Delta JS$  is coupled to the end-effector velocity in the base coordinate system  $rEEv_{bcs}$ . Transformation is done via the amplifier K. The end effector velocity  $rEEv_{bcs}$  is integrated over a time interval and is thereby transformed into a position increment  $\Delta rEEp_{bcs}$ . This increment is added to the sampled position of the end effector  $sEEp_{bcs}$ . The  $sEEp_{bcs}$  is originally from the actual position measurement inside the manipulator joints, actual joint positions  $aJp$ . The  $aJp$  is sampled (with a predefined interval) and fed into the forward kinematic model  $EEp = f(Jp)$  and the end effector actual position with reference to the base coordinate system is calculated  $sEEp_{bcs}$ . The result after the summation of  $\Delta rEEp_{bcs}$  and  $sEEp_{bcs}$  is fed into the inverse kinematics calculations  $Jp = f^{-1}(EEp)$  which result in the reference joint position  $rJp$ . The change of the reference position for a given time increment gives the reference joint velocity  $rJv$ . The  $rJv$  signal is compared with the measured actual joint velocities  $aJv$ . The difference between the reference and the actual value is the error signal  $e$ . This signal is fed into the velocity controller for every joint. The robot(slave) joint starts to move.

In d.) the operator watches camera feedback from the work site and controls the joystick to obtain the desirable end-effector velocity with reference to the base coordinate system.

#### **e.) Base coordinate position control (closed servo loop)**



**Figure 4.17** Base coordinate position control (closed servo loop) architecture

This architecture is similar to the d.) architecture, but in this latter case it is the end-effector position with reference to the base coordinate system which is the object for our control strategy.

In e.), the joystick activation  $\Delta JS$  is coupled to the end-effector velocity given in the base coordinate system  $rEEv_{bcs}$ . Transformation is done via the amplifier K. The end-effector velocity  $rEEv_{bcs}$  is integrated over a time interval and is thereby transformed into a position increment  $\Delta rEEp_{bcs}$ . This increment is added to the position of the end effector  $sEEp_{bcs}$ . The  $sEEp_{bcs}$  is originally from the position measurement inside the manipulator joints, actual joint positions  $aJp$ . When the joystick is activated (or at given time intervals) the actual position  $aJp$  is sampled  $sJp$ . The  $sJp$  is fed into the forward kinematic model  $EEp = f(Jp)$  and the end effector position with reference to the base coordinate system is calculated  $sEEp_{bcs}$ . The result after the summation of

$\Delta rEEp_{bcs}$  and  $sEEp_{bcs}$  is fed into the inverse kinematics calculations  $Jp = f^{-1}(EEp)$  which result in the reference joint position  $rJp$ . The  $rJp$  signal is compared with the measured actual joint position  $aJp$ . The difference between the reference and the actual value is the error signal  $e$ . This signal is fed into the position controller for every joint. The robot (slave) joint starts to move.

In e.), the operator watches camera feedback from the work site and controls the joystick to obtain the desirable end-effector position with reference to the base coordinate system.

### **f.) End-effector coordinate rate control (closed servo loop)**

The architecture f.) is similar to the architecture presented in d.), however the f.) architecture provides rate control of the end effector with reference to the end effector coordinate system. The reason for the need of such a control possibility is if the base coordinate system of the manipulator is not parallel with any of the work task axis of operation it is not possible to move the end effector linear to the work piece operation direction. The solution for this problem is, for example, to use architecture b.) to move the joints of the manipulator so that the end effector coordinate is approximately aligned with the work task, and the switch to architecture f.), described below.

In f.), the joystick activation  $\Delta JS$  is coupled to the end-effector velocity given in the end effector coordinate system  $rEEv_{ecs}$ . Transformation is done via the amplifier  $K$ . The end effector velocity  $rEEv_{ecs}$  is integrated over a time interval and is thereby transformed into a position increment  $\Delta rEEp_{ecs}$ . The  $\Delta rEEp_{ecs}$  signal is transformed into the corresponding end effector position with reference to the base coordinate system  $rEEp_{bcs}$ . This transformation is carried by the transformation matrix  $T_{bcs}^{ecs}$ . This transformation matrix needs angle information for the calculations. Sampled values  $sJp$  are collected from the position measurements  $aJp$ . The result after transformation is fed into inverse kinematics calculations  $Jp = f^{-1}(EEp)$  which result in the reference joint position  $rJp$ . The change of the reference position for a given time increment gives the reference joint velocity  $rJv$ . The  $rJv$  signal is compared with the measured actual joint velocities  $aJv$ . The difference between the reference and the actual value is the error signal  $e$ . This signal is fed into the velocity controller for every joint. The robot (slave) joint starts to move.

In f.), the operator watches camera feedback from the work site and controls the joystick to obtain the desirable end-effector velocity with reference to the end-effector coordinate system.



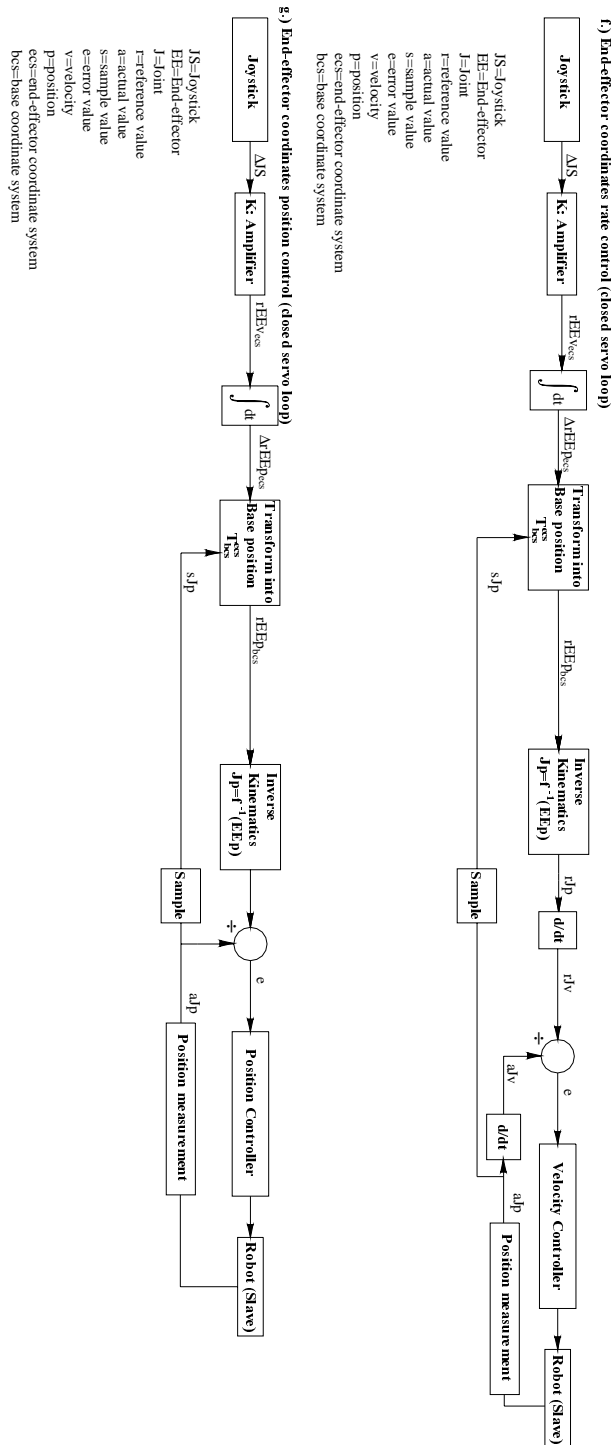


Figure 4.18 f.) End- effector coordinate rate control (closed servo loop) architecture  
 g.) End- effector coordinate position control (closed servo loop) architecture

### **g.) End-effector coordinate position control (closed servo loop)**

The g.) architecture is similar to the f.) architecture, but in the g.) architecture the servo loop is position controlled.

In g.), the joystick activation  $\Delta JS$  is coupled to the end-effector velocity given in the end effector coordinate system  $rEEv_{ecs}$ . Transformation is done via the amplifier K. The end effector velocity  $rEEv_{ecs}$  is integrated over a time interval and is thereby transformed into a position increment  $\Delta rEEp_{ecs}$ . The  $\Delta rEEp_{ecs}$  signal is transformed into the corresponding end effector position with reference to the base coordinate system  $rEEp_{bcs}$ . This transformation is carried by the transformation matrix  $T_{bcs}^{ecs}$ . This transformation matrix needs angle information  $aJp$  and sampled values is collected from the position measurements. The  $rEEp_{bcs}$  is fed into the inverse kinematics calculations  $Jp = f^{-1}(EEp)$  which result in the reference joint position  $rJp$ . The  $rJp$  signal is compared with the measured actual joint position  $aJp$ . The difference between the reference and the actual value is the error signal  $e$ . This signal is fed into the position controller for every joint. The robot (slave) joint starts to move.

In g.), the operator watches camera feedback from the work site and controls the joystick to obtain the desirable end-effector position with reference to the end-effector coordinate system.

### **h.) External coordinate rate control (closed servo loop)**

The control architecture h.) is similar to the architecture presented in f.), however the h.) architecture provides rate control of the end effector with reference to an external coordinate system. If the workpiece coordinate represents a known external coordinate system the h.) architecture enables us to move the end-effector with reference to the external coordinate system. This is a very useful strategy for most work tasks, the challenge is to determine the transformations between the base and external coordinate system. If the robot carrier has a steady base transformation is constant, if the base is moving (unstable docking device) the necessary transformations must be modified with these movements.

In h.), the joystick activation  $\Delta JS$  is coupled to the end-effector velocity given in the end effector coordinate system  $rEEv_{excs}$ . Transformation is done via the amplifier K. The end effector velocity  $rEEv_{excs}$  is integrated over a time interval and is thereby transformed into a position increment  $\Delta rEEp_{excs}$ . The actual position of the manipulator in external coordinates  $sEEp_{excs}$  is calculated from lower feedback loop of the Figure 4.19. Sampled joint positions is transferred into manipulator base coordinates  $sEEp_{bcs}$  which is further transferred into external coordinates  $sEEp_{excs}$  given the transformation between these two systems  $T_{excs}^{bcs}$ . The  $sEEp_{excs}$  is added to the joystick signal  $\Delta rEEp_{excs}$  and the result is transferred into base coordinates given the transformation  $T_{bcs}^{excs}$ . From this point the architecture is similar with the f.) architecture described previously.

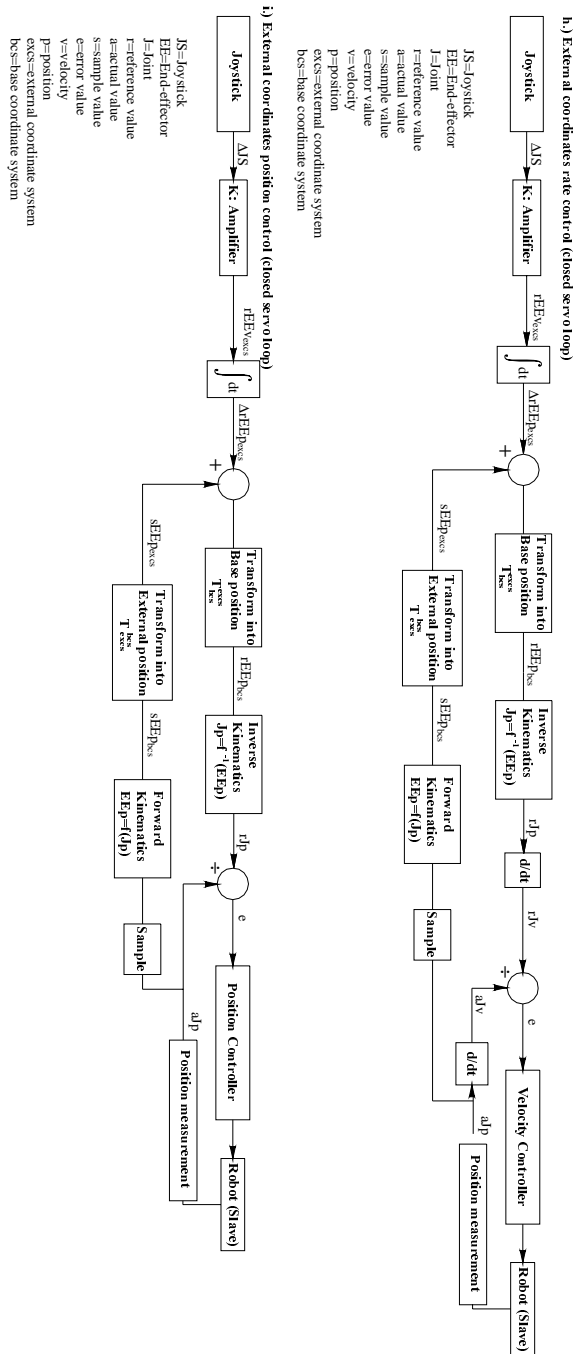


Figure 4.19 h.) External coordinate rate control (closed servo loop)  
 i.) External coordinate position control (closed servo loop)

In h.), the operator watches camera feedback from the work site and controls the joystick to obtain the desirable end-effector velocity with reference to the external coordinate system.

**i.) External coordinate position control (closed servo loop)**

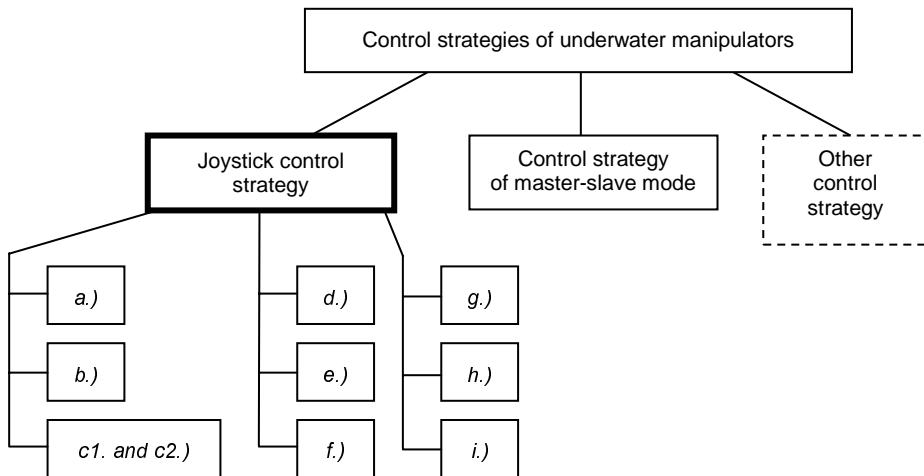
The i.) architecture is similar to the h.) architecture, but in the i.) architecture the servo loop is position controlled as described in f.).

The control architecture i.) provides position control of the end effector with reference to an external coordinate system. If the workpiece coordinate represents a known external coordinate system the i.) architecture enables us to move the end-effector with reference to the external coordinate system. This is a very useful strategy for most work tasks.

In i.), the operator watches camera feedback from the work site and control the joystick to obtain the desirable end-effector position with reference to the external coordinate system.

**4.3.2.3 Summary of conceptual joystick control design**

From the above discussion, I depict the different joystick control strategies into the left part of Figure 4.12, which becomes Figure 4.20.



**Figure 4.20** Control strategies of master-slave mode and joystick mode for underwater manipulators

Where a.), b.), c.), d.), e.), f.), g.), h.), and i.) are the control architectures described in paragraph 4.3.2.2.

For all architectures (from a.) to i.)), the operator is relying on feedback information from the work site and the ongoing intervention operation. The feedback signal is today, normally, realized by a camera system.

The joystick can be coupled to different coordinate systems, from joint coordinate systems to external coordinate systems. In the first case, a joystick axis corresponds to a joint axis movement, while in the latter case a joystick axis corresponds to either the translation or rotation of the end-effector with reference to the external coordinate system.

A standard joystick normally has 3 degrees of freedom and while operating in the joint space only three joints can be operated at the same time. A pushbutton can be activated to be able to move the next group of manipulator joints. Compared with the “master” which is a replica of the slave arm where it is possible to move all joints simultaneously the joystick is less attractive. However, in all other coordinate systems the 3 axis joystick is more attractive than the master (or replica). In these situations the three joystick axes are used to translate/rotate the end-effector in the three orthogonal axes of the selected coordinate frame.

All of the architectures (from b.) to i.) require either a velocity measurement device or a position measurement device in every manipulator joint. In the b.) architecture, the velocity feedback is required alone, but in all other architectures c.) to i.) only position feedback is required or both position and velocity feedback are required. The latter case is represented by architecture d.), f.) and h.). However, in these cases the velocity is computed from the position measurement as the change of position for a given time increment. Thus the velocity feedback can be calculated from the position measurements. This leads us to the conclusion that every joint of the manipulator should have a position measurement device to enable the usage of advanced joystick control. Thus, the actual manipulator mechanical design is dependant on the conceptual control system architecture, as I stated in section 4.2.1.1.

One special joystick has not been investigated throughout the control architectures a.)...i.), namely the force feedback stick. Like the replica master, a force feedback joystick could also receive information from the underwater manipulator to give the operator a feeling of the load onto the equipment. In the replica master technology we used torque feedback from the different joints, see section 4.3.1.2 architecture c.) and d.), and such a strategy may be used in a force feedback joystick system as well.

Since the joystick is an excellent device for control in the Cartesian space (movement of the manipulator end-effector in the base or external coordinate system) a force measurement device situated in the end-effector and coupled back to the stick will give the operator an idea about the real contact forces between the manipulator and the workpiece. Such a system can be used as a indicator of collision. For an operator relying on 2D camera feedback such force feedback system may act as the third depth

dimension. The usage of force measurement and feedback from the end-effector is further described in the next section 4.3.3, trajectory mode.

By studying all above architectures we may identify the common software and hardware components in the joystick control architectures. Summarizing the different blocks in all architectures a.) to i.), including usage of the force feedback joystick, we find the following components: The joystick, the amplifiers K, the integrators and differentiators, the coordinate transformers, the forward and inverse kinematic blocks, the sample blocks, the controllers, the feedback devices (position, velocity, joint torque, end-effector forces), and the robot itself. Here, all of the blocks (except from the robot, the joystick, and the measurement feedback devices) can be realized as software solutions alone. This leads us to the usage of microprocessors. Since most of the architectures include time critical operations like derivations and integrations the microprocessor must address time in a serious manner. This can be done by applying a real time operating system (RTOS) to the control system. Thus the actual manipulator electrical design is dependent on the conceptual control system architecture as well.

### **4.3.3 Computerised high-accuracy trajectory mode**

#### ***4.3.3.1 Computerised high-accuracy trajectory strategy***

Existing underwater manipulators have some considerable drawbacks when applied in high precision work. With remote manual joystick control it is extremely difficult to follow a three dimensional curve, and even with master/slave control this task is difficult and cumbersome. New types of underwater manipulators with computer tools will appear necessary to ensure these facilities in the most cost effective way.

NDT methods such as the use of eddy currents on underwater structural nodes, involve following a three dimensional curve with an accuracy in the millimeter range. Automation of this task will also require an accurate geometric model of the curve (Kallevik and Hendseth, 1991). For those underwater manipulation tasks, which require trajectory operations, like seam welding and NDT tool trajectory generating tasks, the master-slave functionality become inefficient, or even impossible. This leads to additional implementation of a computerised high-accuracy trajectory control strategy, to meet the requirement of these high-accuracy trajectory operations.

In addition, due to the low stiffness of the docking system causing the movement of ROV-manipulator, the difficulties of operating such systems get even worse. In fact, the operator has to manually compensate for the end-effector movements induced by the movements required by the task execution, from the ROV garage via the tether or caused by ocean currents, all by means of the same master arm (Nicolodi et al., 1990).

Then influence of the lack of stiffness in the docking device was examined by Ricci and Ellingsen (1992). They implemented a suction type docking device on a ROV system and measured its movements in relative to the anchor construction and found

that the relative movements of the manipulator tool-centre-point were as follows in Table 4.1.

**Table 4.1** Disturbances on ROV-manipulator via docking device

Translation			Rotation		
Horizontal [mm]	Vertical [mm]	Lateral [mm]	Yaw [deg]	Roll [deg]	Pitch [deg]
± 0.10	± 0.12	± 0.15	± 5	± 4	± 8

Frequencies are expected to be low, due to the large mass of the ROV. Sources among experienced operators have indicated 0.1-0.5 Hz (Lien and Thomessen, 1992).

Driscoll, Lueck and Nahon (2000) found that the most significant forcing of a ROV garage (cage) was in the wave frequency band 0.1-0.25Hz and accounted for over 90% of the variance of vertical acceleration. These disturbances occur mainly from the ship movement, via the umbilical, down to the ROV-garage. Via the tether some elements of these motions will occur at the ROV vehicle itself, influencing the manipulator control.

At some point of the testing Driscoll et al. found that the vertical motion of the cage was larger than the vertical motion of the ship, indicating operation conditions near the natural frequencies. This resulted in so called snap loads in the tether between the garage and the ROV vehicle, which at one instance resulted in a sheared tether, a situation which may lead to the loss of the ROV vehicle itself.

Summing up these results, we find that the influence from the environment onto the manipulator are far too important to be neglected, and has to be controlled in terms of either a active compensation system or a passive strong docking system.

One may argue to that you can design a strong mechanical docking device, similar to the one shown in Figure 3.4, to gain the ROV to be docked as a fixed base. However, in a workspace constraint underwater environment, in some cases, there is no room for large sized docking devices. In such cases, the computer compensation is necessary combined with a relatively less strong, but dexterous docking legs.

Thus, if we can add a compensation function into the master-slave control function to compensate the ROV movement, then it will release the operator by leaving the duty of manually compensating the ROV movement to the computer, and the operations become, from man's point of view, to the ones with a fixed base.

Taylor (1993) states that most sub-sea manipulators operate without force-feedback or contact sensing, so the operator has no way of telling whether the arm is just clear of the surface or is applying full force to it unless there is some visual information such as deflection or deformation. This may of course be too late to prevent damage to the manipulator or the workpiece.

So, the need of a contact sensing device is clear, and such a system may be added to our control architecture. Elle, Thomessen and Lien (1993) suggest implementing a force feedback device at the manipulator end-effector, giving us the opportunity to both detect collisions and as well actively use the force feedback for contour following operations.

Force feedback is of special interest in grinding operations since the grinding process itself requires control of the contact forces with the workpiece to obtain optimal process parameters. The grinding tool wears during the operation, and to maintain correct force against the workpiece the trajectory must be altered according to the change in absolute position.

Contour following, makes a good strategy for following a surface with unknown geometry. This enables us to perform NDT operations, with the tool following a surface with a constant distance relationship.

So by adding a force feedback system we will improve manipulator performance and security.

Thus, in the coming sub-section possible conceptual high accuracy trajectory architectures is identified and their control architectures is developed.

The selected control strategies are:

- a.) Trajectory control with compensation of a moving base*
- b.) Trajectory control with compensation of a moving base and automatic contact force control*

#### **4.3.3.2 Computerised high-accuracy trajectory control architectures**

##### **a.) Trajectory control with compensation of a moving base**

Refer to Figure 4.21 a.)

An operator can interact with the control architecture via an interface, keyboard/screen or similar. From a set of possible commands the operator makes a manipulator program which the administrative unit put into the program storage. For a trajectory control operation the operator gives information of the desired position/orientation/speed of the end effector of the manipulator (often with respect to the manipulator base coordinates). If the programmed position/orientation/speed is given with reference to some other than the base coordinate system, the operator normally must provide the information of the location of this coordinate system with respect to the manipulator base coordinate system, and by this the reference position/orientation/speed can be transformed into manipulator base coordinate system.



When time has come to execute the operator program the administrative unit fetches the new position/orientation/speed reference and transfer it into the trajectory generator. In the trajectory generator the distance to be driven is divided into increments where the increment size is dependent on the desired speed in the given period. The speed is controlled by an own acceleration unit. The calculated increment is added to the previous calculated position and out of the trajectory generator a reference position of the manipulator is given  $rEEp_{bcs}$  (Lien, 1980; 1993)

To compensate for a moving base, as described earlier, a measurement arm for ROV-movement compensation is attached to the sub-sea structure and the measurement arm joint positions is measured  $aJMp$ . The actual end-effector position of the measurement arm with respect to the manipulator base coordinates  $aEEMp$  is calculated by the forward kinematic model  $EEMp = f(JMp)$ . When start of the compensation procedure a reference position is taken by the sample unit  $sEEMp$ . During the operation the start-up position is subtracted from the actual measurement arm position. This results in a compensation signal  $cEEp$  which represents the movement of the base coordinate system. The compensation position  $cEEp$  from the moving base is added to the trajectory reference position  $rEEp_{bcs}$ .

The new compensated reference signal is led to the inverse kinematics module where the cartesian reference is transformed into joint position coordinates  $rJp$ , for every manipulator joint. From the manipulator joints the actual position is measured and subtracted from the references  $rJp$ . The position error signal and also possible joint speed signals are fed into the joint controller which controls the movement (position/speed) of every joint so that the manipulator end-effector follows the desired trajectory.

### **b.) Trajectory control with compensation of a moving base and automatic contact force control**

Refer to Figure 4.21 b.)

The difference between the a.) and b.) structure is that in the latter a work task force control is implemented. This is a very important feature in intervention tasks which requires control of the work tasks forces. This is particularly true in grinding operations where grinding forces are one of the important process parameters. Also when conducting inspection task with sensitive equipment the force controller intends to keep the interaction forces between the object and the manipulator down to a minimum, avoiding damage to the equipment.



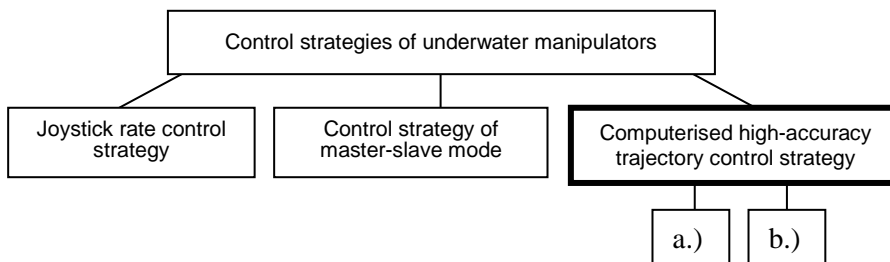
The force controller loop works as follows:

A reference work task force must be given by the operator. This information is given with reference to the work task coordinate system, thus the transformation between the work task and the manipulator must be known. The work task reference force  $rWf_{wcs}$  is compared with the actual work task force  $aWf_{wcs}$  and the difference or error signal  $e$  is fed into a force controller. The actual work task force  $aWf_{wcs}$  is derived from a force sensor normally applied close to the end-effector of the manipulator. The measured forces are called actual work task force with reference to the sensor coordinate system  $aWf_{scs}$ . Since the relationship between the sensor coordinate system and the robot end effector is a known constant, it is possible to calculate the forces with respect to the end effector coordinate system, denoted  $aWf_{ecs}$ . The actual forces related to the manipulator base system  $aWf_{bcs}$  are calculated as a result of the actual joint positions  $aJp$ . Since the relationship between the base (at start-up of operation) and the work task is known the forces can be calculated with reference to the task coordinate system,  $aWf_{wcs}$ . And finally, as mentioned above, these actual forces are compared with the reference forces  $rWf_{wcs}$  and the error is fed into the force controller. The task of the force controller is to create a change in position signal  $\Delta Wp_{wcs}$ , in such a manner that the force error is eliminated or minimized. This new position signal is now transformed into manipulator base coordinates  $\Delta EEp_{bcs}$  and this is added to the reference end effector position  $rEEp_{bcs}$  from the trajectory generator. The base compensation is also summed and after the inverse kinematics module the joint position loop is activated as described in architecture a.)

The b.) architecture expect from the moving base compensation is based on Thomessen (1992). Thomessen (1992) gives a detailed analysis of strategies for both light and heavy grinding operations. All strategies are implemented as an external control loop around the internal position controller of the manipulator joints.

#### 4.3.3.3 Summary of conceptual trajectory control design

From the above discussion, I depict the different trajectory control strategies into the right part of Figure 4.12, which becomes Figure 4.22.



**Figure 4.22** Control strategies for underwater manipulators

Where a.), b.) is the control architectures described in section 4.3.3.2.

For both architectures a.) and b.), the operator is not situated inside the control loop. The operator has a supervisory role during execution of a program.

Both of the control architectures are arranged around a position control loop. This will require a joint position measurement device, installed in every manipulator joint. The contact force measurement requires us to have an end-effector force sensing system.

The compensation of a moving base requires us to develop a suitable measurement system.

The rest of the blocks inside the architectures are realised by software which leads us to the usage of microprocessors, applying a real time operating system (RTOS) to the control system.

From the above discussion I again find that the conceptual control system architecture is of major importance in the design procedure.

#### **4.3.4 Summary of section 4.3**

In the section 4.3 different control strategies and architectures for underwater intervention manipulators working in a constraint environment has been investigated.

Architectures for:

- master-slave control
- joystick control and
- trajectory control

have been studied, refer to section 4.3.1, 4.3.2 and 4.3.3.

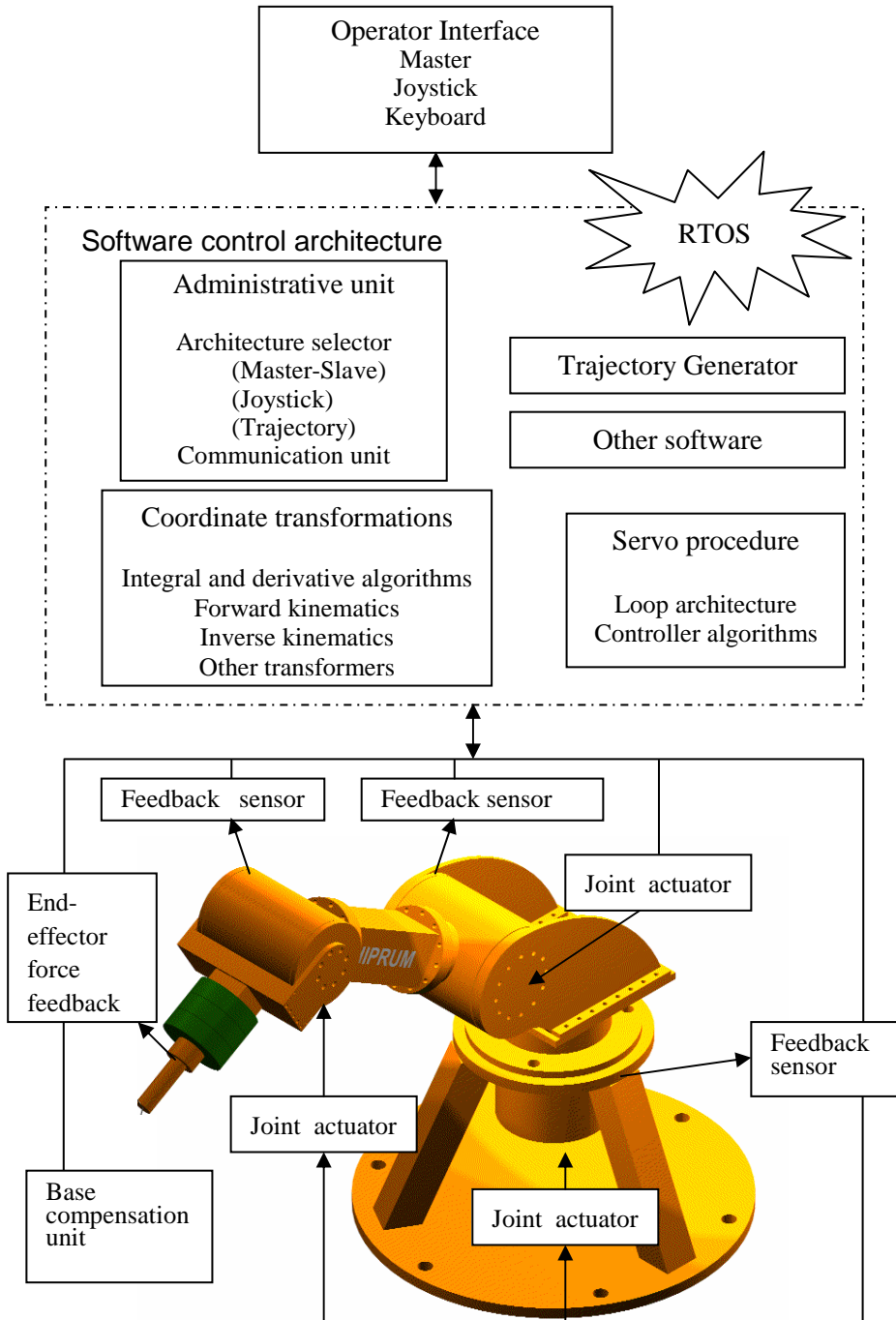
During the development of these conceptual strategies, I have strengthened the idea that the control strategy and architecture is of major importance to the coming design process.

By looking into the different architectures, master-slave, joystick, and trajectory, I would say that they are all good for their special purposes. However, by combining them the designer may create a future control system which is of very high capability addressing a set of special work task and working conditions. The designer may select one or more strategy(s), depending on the requirement of the underwater manipulation tasks.

Combining these architectures is not a very complicated task, since most of the blocks are realised as software solutions. The hardware components will be the operator interface, the selected manipulator feedback system, camera system, and the base movement system. A software realisation requires us to use microprocessors and also a

RTOS when addressing time critical operations. Figure 4.23 visualises the hardware and software components in a conceptual reference control strategy.

The future designer may use section 4.3 as a reference, while selecting her/his control strategies for the equipment to be built.

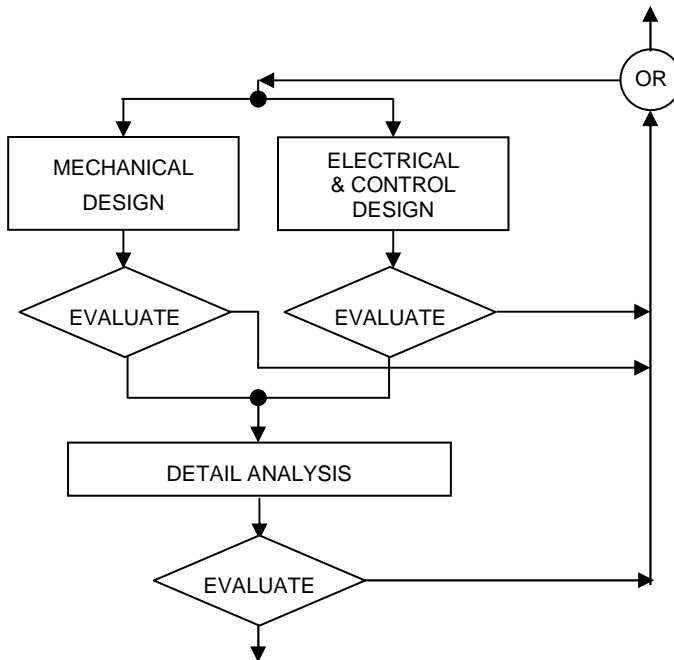


**Figure 4.23** Reference conceptual control strategy

## 4.4 Methodology for analysis of precise design

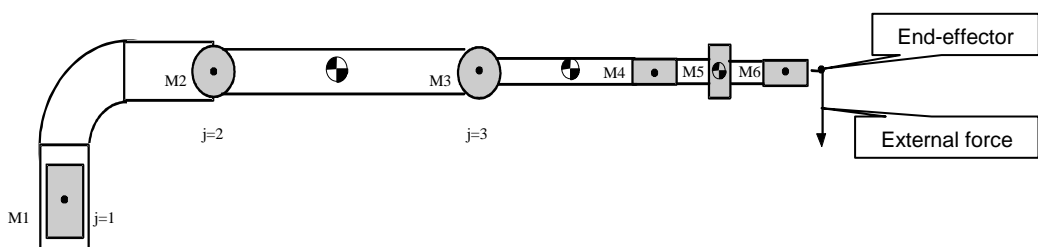
### 4.4.1 Introduction

In the discussion in section 1.2, I have mentioned that one of my research focuses is to “Set up a precise design methodology for aiding the development of precise underwater manipulators for meeting requirements of high precision work and accurately following three-dimensional curve (trajectory)”.



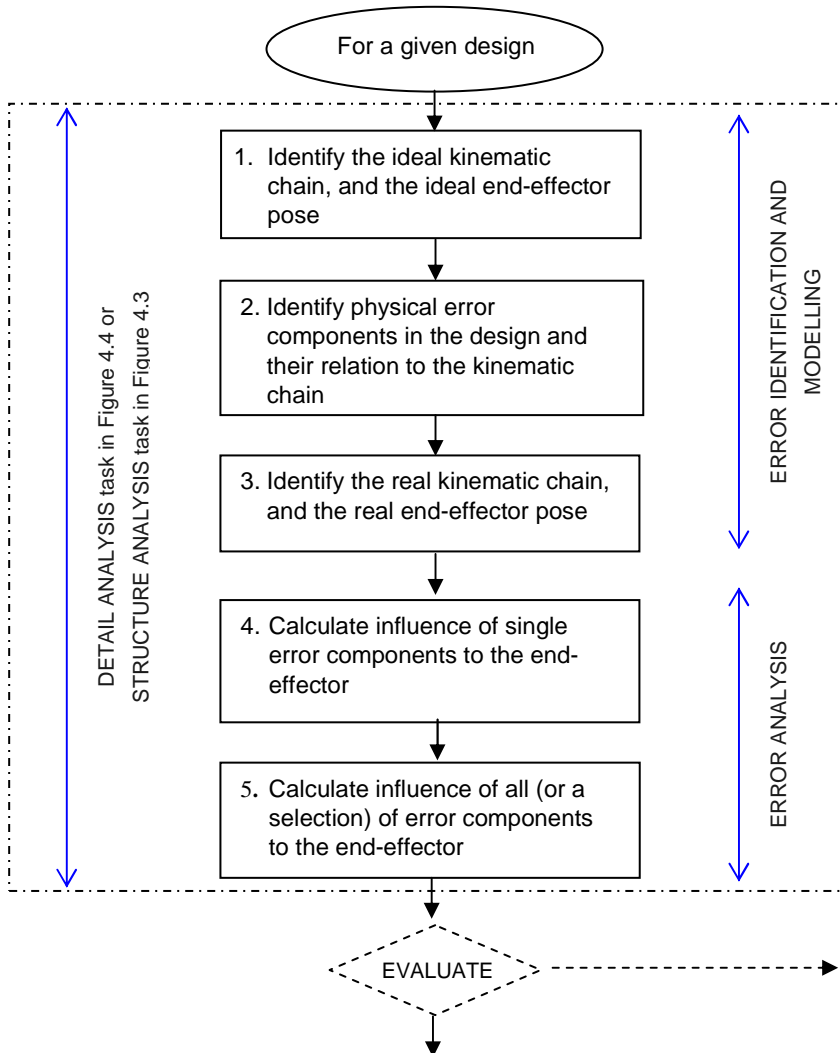
**Figure 4.24** An iterative process of “design-analysis-design-analysis ...”

In the previous sections of this chapter, I have discussed the design issues of underwater manipulators. In Figures 4.2, 4.3 and 4.4, the philosophy of precise design is implicitly contained, i.e., the philosophy is design – error analysis – redesign – error reanalysis ... an iterated procedure.



**Figure 4.25** An example manipulator

For example in Figure 4.4, a *detailed analysis* task follows the *mechanical design* and *electrical and control design* tasks, which are re-depicted as Figure 4.24. From this figure, we can see that the philosophy of precise design is using an iteration process of “design – analysis – design – analysis...”.



**Figure 4.26** An error analysis procedure for the analysis methodology

After the detailed *mechanical design* and detailed *electrical & control design*, manufacturing tolerances and other physical errors either can be identified, or calculated. From here, one may analyze how those errors will influence the precision of the position and orientation of the end-effector of the underwater manipulator. Figure 4.25 shows an example of a manipulator, in which its end-effector is shown. Obviously, all internal physical errors from the joints (M1, M2,..., M6), the links



between joints, will induce the end-effector total error. As well, the external load will induce the end-effector error through the manipulator structure.

Thus, for analyzing how the physical errors influence the end-effector's position and orientation accuracy, an analysis methodology must be conceived. For this purpose, I suggest an error analysis framework as shown in Figure 4.26.

In general, the error analysis strategy proposed leads to the calculation of the ideal and real end-effector pose. Thus, the end-effector error can be found by the difference between the ideal and real pose.

The error analysis framework (Figure 4.26) consists of two analysis parts. The first part is error identification and error modeling. The second part is for the error analysis.

**The error identification and modeling part** includes 3 steps (upper part of Figure 4.26).

In general, there exists no error unless there exists an ideal reference. So the first step is to *identify the ideal kinematic chain, and the ideal end-effector pose*.

The second step is to *identify physical error components in the design and their relation to the kinematic chain*. This second step includes determination of "type" of error. Errors can be distinguished into "repeatable" and "random" errors (Slocum, 1992). Repeatable errors are errors whose numerical value and sign are constant for each manipulator configuration. Random errors are errors whose numerical value or sign changes unpredictably. At each manipulator configuration, the exact magnitude and direction of random errors cannot be uniquely determined, but only specified over a range of values (Mavroidis, Dubowsky, Drouet, Hintersteiner and Flanz, 1997). After manufacturing and prototyping, numerical value of the repeatable errors can be identified and the ideal kinematic chain may be updated to include these parameters, assuming that the formation of the kinematic chain allows for such update. The nature of the random errors excludes them for a similar treatment. Error compensation of these kinematic errors is known as calibration. Calibration techniques will not be treated in this dissertation, interested readers are suggested to survey Hollerbach (1988) or Mooring et al., (1991).

The task of the third step is to identify the real kinematic chain, and the real end-effector pose.

**The error analysis part** includes two steps (lower part of Figure 4.26).

The task of the fourth step is to calculate the influence of single error components to the end-effector. Many physical errors may not play an important role in the total end-effector error, while others are of a critical magnitude. Thus, this analysis step enables

the designer to identify both critical and less critical components, enabling the team to put effort where effort is needed.

The fifth step is to calculate influence of all (or a selection) of error components to the end-effector. From this analysis we can realise the combinatorial effects of a set of errors upon the end-effector, and/or we can study the effect from all errors upon the end-effector, the latter case is known as a worst case analysis.

### **Analysis results**

After conducting all five steps in Figure 4.26, the following *analysis results* related to end-effector error will be given:

1. Error sources, types of error, their numerical max/min value, and their interaction/relation with the kinematic chain are identified.
2. Identified which single physical errors significantly influence the end-effector error.
3. Identified worst case end-effector pose deviation due to the existing design.

### **Interaction with the design procedure**

Look at the bottom of Figure 4.26 where a dotted-line bordered task, EVALUATE, is given. Comparing with Figure 4.4 (detail design procedure), the EVALUATE task in Figure 4.26 corresponds to the EVALUATE task after the DETAIL ANALYSIS task of Figure 4.4. The large stroke-dotted-line bordered box in Figure 4.26 corresponds to the task, DETAIL ANALYSIS, of Figure 4.4.

However, as I have pointed out in final paragraphs of section 4.2.2, the analysis methodology illustrated in Figure 4.26 can also be used in the preliminary design phase for the task of STRUCTURE ANALYSIS. See middle part of Figure 4.3

Thus, we see that the EVALUATE task is actually to evaluate the above-mentioned three analysis results, resulting from the sub-tasks in the large stroke-dotted-line bordered box of Figure 4.26. The design procedure itself is an iterative procedure and allows us to take the necessary step “up” in the procedure, in situations when the analysis result makes it necessary.

To summarize the above discussion on the analysis methodology for the detail analysis task in the detail design phase, or for the structure analysis task in the preliminary design phase, I would conclude that the suggested analysis methodology consists of the following five analysis sub-tasks:

1. Identify the ideal kinematic chain, and the ideal end-effector pose
2. Identify physical error components in the design and their relation to the kinematic chain

3. Identify the real kinematic chain, and the real end-effector pose
4. Calculate influence of single error components to the end-effector.
5. Calculate influence of all (or a selection) of error components to the end-effector.

Obviously, it is necessary to develop methods to carry out the above five tasks. Hence, in the following sections, I will further study those individually. At the end of this chapter, existing methods are reviewed with the purpose of verifying the need for my new error modelling methodology.

#### **4.4.2 Identify the ideal kinematic chain, and the ideal end-effector pose**

As stated in the previous paragraph, if there exists an “error” it follows that there should exist a state (at least imaginary) where there is no error present. Such a state is named “ideal” or “reference” state.

When defining an “ideal kinematic chain” we therefore assume that there are no errors in our kinematic chain. From this “ideal kinematic chain” we can find (calculate) the “ideal end-effector pose”.

To describe a kinematic chain several methods have been developed, one of the most commonly used is the Denavit, Hartenberg and Evanston (1955) (D-H) method which involves a four parameter description of the relationship between two coordinate frames. However, the D-H methodology has some limitations and other methods to describe the kinematic chain have been developed (Lien, 1989). One of these is a 6 parameter description, which I refer to as the “general” convention. As will be seen later most of the other existing error modelling methods involve the D-H convention, while I prefer to use the 6 parameter “general” description, to overcome the shortcomings of the D-H methodology. To clarify the latter point, there is a need to study these two methodologies in the context of defining the “ideal kinematic chain”.

##### **4.4.2.1 Denavit-Hartenberg convention for coordinate frame assignment**

Figure 4.27 shows the principle of Denavit-Hartenberg notation. The parameters in this figure describe how two coordinate systems are connected to a physical structure.

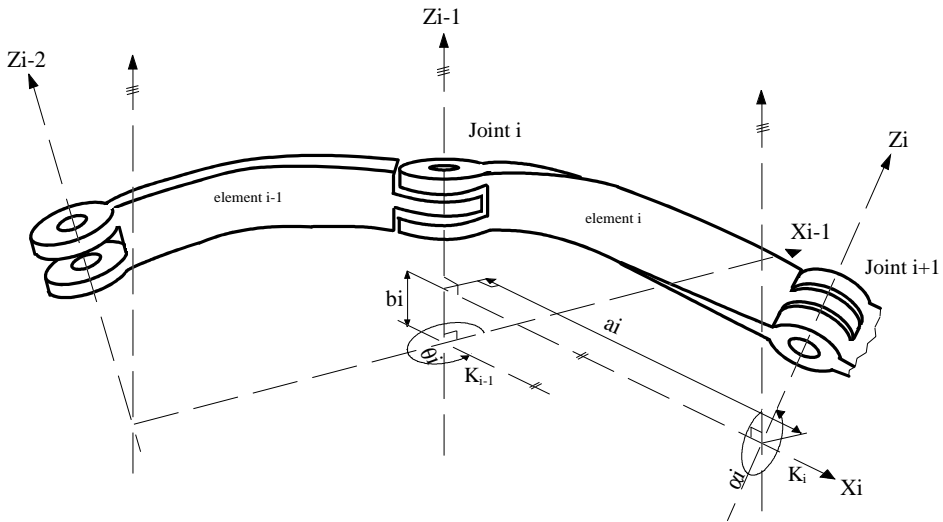
The Denavit-Hartenberg convention defines the formation of the link frame  $K_i$  (see Figure 4.27) as:

- Locate the origin of frame  $K_i$  at the intersection of axis  $z_i$  with the common normal to axes  $z_{i-1}$  and  $z_i$ . (The common normal between two lines is the line containing the minimum distance segment between the two lines.)
- Choose direction of axis  $x_i$  along the common normal to axes  $z_{i-1}$  and  $z_i$  with direction from joint  $i$  to joint  $i+1$ .
- Choose axis  $y_i$  so as to complete a right-handed frame (Sciavicco and Siciliano, 1996).

The formation of the frame  $K_{i-1}$  is done in a similar manner.

The 4 parameters used in the description between frame  $K_{i-1}$  and frame  $K_i$  are :

- ⇒  $\theta_i$  is the angle between  $X_{i-1}$  and  $X_i$ , a rotation around  $Z_{i-1}$  axis. Positive direction of rotation is given by the direction of the closing fingers into your palm while your right hand thumb points in the direction of the axis of rotation.
- ⇒  $b_i$  is the translation between  $X_{i-1}$  and  $X_i$  along  $Z_{i-1}$  axis.
- ⇒  $a_i$  is the translation between  $Z_{i-1}$  and  $Z_i$  along  $X_i$  axis.
- ⇒  $\alpha_i$  is the angle between  $Z_{i-1}$  and  $Z_i$ . This is a rotation around  $X_i$  axis.



**Figure 4.27** A linkage system with coordinate system assignment

Every arm element has a fixed coordinate frame  $K_{i-1}$ . In Figure 4.27, arm element  $i$ , has its main axis of rotation (joint angle variable) around coordinate frame  $Z_{i-1}$ .

To “move forward” from frame  $K_{i-1}$  to frame  $K_i$  the following procedure is followed:

- ① A rotation  $\theta_i$
- ② A translation  $b_i$
- ③ A translation  $a_i$
- ④ A rotation  $\alpha_i$

By using homogenous transformations we can define a  $4 \times 4$  matrix to describe the four steps above. These matrixes are: (Please note:  $c = \text{cosine}$ , and  $s = \text{sine}$  )

$$\begin{aligned}
 T_{\theta_i} &= \begin{bmatrix} c\theta_i & s\theta_i & 0 & 0 \\ -s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_{b_i} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & b_i & 1 \end{bmatrix} \\
 T_{a_i} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a_i & 0 & 0 & 1 \end{bmatrix} & T_{\alpha_i} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & s\alpha_i & 0 \\ 0 & -s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{4.1}$$

The total transformation is given by multiplication:

$$T_{DHi} = T_{\alpha_i} \cdot T_{a_i} \cdot T_{b_i} \cdot T_{\theta_i} \tag{4.2}$$

So,

$$T_{DHi} = \begin{bmatrix} c\theta_i & s\theta_i & 0 & 0 \\ -s\theta_i \cdot c\alpha_i & c\theta_i \cdot c\alpha_i & s\alpha_i & 0 \\ s\theta_i \cdot s\alpha_i & -c\theta_i \cdot s\alpha_i & c\alpha_i & 0 \\ a_i \cdot c\theta_i & a_i \cdot s\theta_i & b_i & 1 \end{bmatrix} \tag{4.3}$$

Matrix 4.3 represents the kinematic model, or say, a kinematic relationship, of the frame  $K_{i-1}$  and  $_{\text{frame}} K_i$ .

A similar approach is used to define the other matrixes for all of the other beam element. The total serial “ideal kinematic model” is then defined by the multiplication of all of these matrixes:

$$T^{ideal} = T_{DHi} \cdot T_{DHi-1} \cdot \dots \cdot T_{DH1} \tag{4.4}$$

Sciavicco and Siciliano (1996) pointed out that the Denavit-Hartenberg convention gives a nonunique definition of the link frame in the following cases:

- For frame  $K_0$ , only the direction of axis  $z_0$  is specified; then origin  $O_0$  and axis  $x_0$  can be arbitrarily chosen.
- For frame  $n$ , only the choice of axis  $x_n$  is constrained (it has to be normal to the axis  $z_{n-1}$ ). Indeed, there is no joint  $n+1$ , and thus  $z_n$  is not defined and can be arbitrarily chosen.

- When two consecutive axes are parallel, the common normal between them is not uniquely defined.
- When two consecutive axes intersect, the direction of  $x_i$  is arbitrary.
- When joint  $i$  is prismatic, only the direction of axis  $z_{i-1}$  is determined

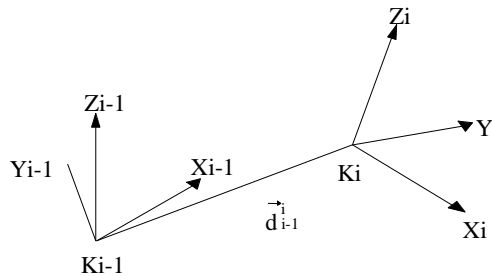
The Denavit-Hartenberg “rules”, or say “restrictions”, for coordinate assignment makes the usage of this approach at least a little cumbersome for the inexperienced newcomer. Lien (1989) pointed out that there might be necessary to include “dummy” coordinate systems to enable a D-H formulation of a robot with an arbitrarily selected convention for the specification of the robot end-effector orientation.

I would like to point to another important fact, the Denavit-Hartenberg assignment of coordinate frames does not necessary relate the assigned frames to the physical construction itself. As seen in figure 4.27, the assigned D-H coordinate frames are situated outside the physical construction. In general, all error modeling methodologies need to relate the physical component error onto the kinematic chain (parameters). The D-H assignment complicates this error modeling process since the relationship between the kinematic chain and the physical construction is not so straightforward to interpret.

From the above discussion I find it desirable to investigate another possibility for coordinate assignment which has fewer restrictions than the D-H methodology.

**4.4.2.2 A general 6 parameter description of coordinate relations**

The position of an object in space, assigned with a coordinate system  $K_i$ , can be described by a 3 parameter representation  $x y z$ .



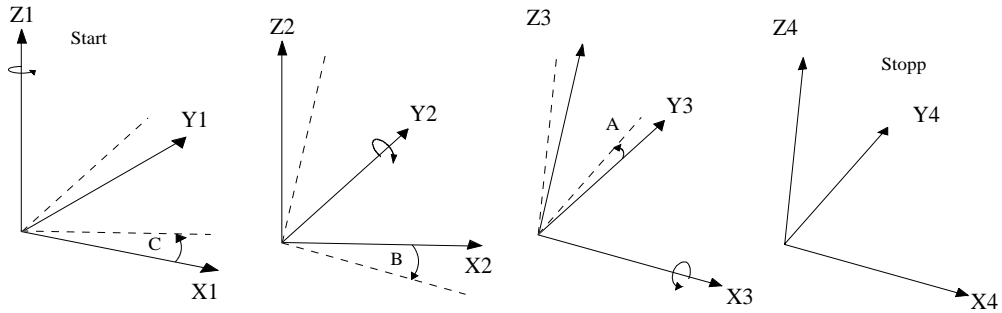
**Figure 4.28** Distance vector  $\vec{d}_{i-1}^i$

In Figure 4.28,  $\vec{d}_{i-1}^i$  is the distance or translation vector from frame  $K_{i-1}$  to frame  $K_i$ .

Coordinate values  $\vec{d}_{i-1}^i = [x \ y \ z]$  given as frame  $K_{i-1}$  coordinates. So, by assigning three parameters to the location of the frame, we have achieved full freedom to decide the exact location of the frame  $K_i$ . If such a methodology is assigned to a manipulator joint we can decide the location of the joint coordinate system anywhere at the joint

axis of rotation, this is not achieved with the usage of the above described D-H methodology.

As will be seen later, an existing error modelling methodology suggest to describe its element physical errors in relationship with the joint coordinate system  $K_i$ , and then it is clearly a benefit to have the full freedom to choose the location of  $K_i$ , along the joint axis of rotation.



**Figure 4.29** Orientation convention, consecutive rotations CBA

To describe the orientation of the object in space, several methods have been developed, in this dissertation I have chosen to use the so-called Euler angles, A, B and C around the x, y, z axis respectively. The rotations are carried out one by one. The sequence of rotation or say the rotation convention must be decided. One possible solution is defined by the three consecutive rotations C, B and A. This convention is depicted in Figure 4.29. C is a rotation about  $Z_1$  axis, while B is the rotation about the  $Y_2$  axis. A is the final rotation conducted about the  $X_3$  axis.

These consecutive rotations can be described each with matrixes as follows:

$$R_{CBA} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cA & sA \\ 0 & -sA & cA \end{bmatrix} \cdot \begin{bmatrix} cB & 0 & -sB \\ 0 & 1 & 0 \\ sB & 0 & cB \end{bmatrix} \cdot \begin{bmatrix} cC & sC & 0 \\ -sC & cC & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{CBA} = \begin{bmatrix} cB \cdot cC & cB \cdot sC & -sB \\ -sC \cdot cA & cC \cdot cA & sA \cdot cB \\ +sA \cdot sB \cdot cC & +sA \cdot sB \cdot sC & \\ sA \cdot sC & -sA \cdot cC & cA \cdot cB \\ +cA \cdot sB \cdot cC & +cA \cdot sB \cdot sC & \end{bmatrix} \quad [4.5]$$

Combining the rotation matrix and the translational part, the transformation between the coordinate system frames  $K_{i-1}$  and  $K_i$  is given by:

$$T_i = \begin{bmatrix} cB \cdot cC & cB \cdot sC & -sB & 0 \\ -sC \cdot cA & cC \cdot cA & sA \cdot cB & 0 \\ +sA \cdot sB \cdot cC & +sA \cdot sB \cdot sC & & \\ sA \cdot sC & -sA \cdot cC & cA \cdot cB & 0 \\ +cA \cdot sB \cdot cC & +cA \cdot sB \cdot sC & & \\ X & Y & Z & 1 \end{bmatrix} \quad [4.6]$$

This equation represents the “general” universal transformation matrix between two coordinate systems in space. Any configuration between two coordinate frames can be described using this equation.

To use the “general” matrix adapted into the coordinate assignment in manipulator systems the following qualities are found:

- We can freely locate the  $K_i$  frame onto the joint axis of rotation (or displacement, in case of linear joints).
- For a rotary joint, the joint activation must be selected to represent one of the consecutive rotations C, B or A. Two of the rotations can be used to orient the joint itself while the third is used to represent the joint activation. In the case where it is necessary to use both C and B (fixed values) rotations to describe the orientation of the joint, the last rotation A (variable) must be used to represent the joint activation. Thus  $x_i$  axis of the  $K_i$  system is decided on the basis of the coordinate frame which arises after the C and B rotation.
- For a linear joint, the joint activation must be selected to represent one of the  $K_{i-1}$  axes.

This third point is a restriction since this means that the joint activation is restricted to move in the direction of only one of the three linear directions of  $K_{i-1}$  (x, y, z). However, compared with the Denavit- Hartenberg convention which only accepts that linear joint operates in the  $z_{i-1}$  direction, this “general” method still is a better alternative.

Thus by using the “general” approach coordinate frames can be assigned to all the joints of the manipulator system and we can formulate the equation of the “ideal kinematic chain”. This is done by the multiplication of the set of  $T_i$  matrixes, as defined in equation 4.7.

$$T^{ideal} = T_n \cdot T_{n-1} \cdot \dots \cdot T_1 = \prod_{i=n}^1 T_i \quad [4.7]$$



Indexing  $n$  in the above equation describes the transformation matrix between coordinate frame  $n-1$  and  $n$ . Numbering of coordinate frames start with the base with index 0 and ends with the end-effector with numbering  $n$ . As an example  $T_1$  describes the relationship between the  $K_0$  and the  $K_1$  frame

To sum up the “general” approach in perspective to the D-H methodology we see that the “general” methodology of coordinate assignment is less restrictive than the D-H methodology. The general methodology allows us to freely locate the  $K_i$  coordinate system inside the physical construction itself, this is a benefit when it comes to error modeling, as will be addressed later. However, due to the above reasoning, I select the “general” 6 parameter methodology as the methodology for the kinematic formation.

End-effector ideal position and orientation,  $(EEp_{bcs})^{ideal}$ , is derived from the structure ideal kinematic chain matrix 4.7 (Lien , 1980):

$$EEp_{bcs}^{ideal} = \begin{bmatrix} X_{bcs}^{ideal} & Y_{bcs}^{ideal} & Z_{bcs}^{ideal} & A_{bcs}^{ideal} & B_{bcs}^{ideal} & C_{bcs}^{ideal} \end{bmatrix} \quad [4.8]$$

The ideal position of the end effector, with respect to the base coordinate system is identified as follows:

$$\begin{aligned} X_{bcs}^{ideal} &= T_{4,1}^{ideal} \\ Y_{bcs}^{ideal} &= T_{4,2}^{ideal} \\ Z_{bcs}^{ideal} &= T_{4,3}^{ideal} \end{aligned} \quad [4.9]$$

Where  $T_{i,j}^{ideal}$  is element of the matrix  $T^{ideal}$  (4.7):

$$T^{ideal} = \begin{bmatrix} T_{1,1} & T_{1,1} & T_{1,1} & 0 \\ T_{2,1} & T_{2,2} & T_{2,3} & 0 \\ T_{3,1} & T_{3,2} & T_{3,3} & 0 \\ T_{4,1} & T_{4,2} & T_{4,3} & 1 \end{bmatrix}$$

The Euler angles A, B and C can be derived:

$$\begin{aligned} A_{bcs}^{ideal} &= \text{arctg2} \left( \frac{T_{2,3}}{T_{3,3}} \right) \\ B_{bcs}^{ideal} &= -\arcsin(T_{1,3}) \\ C_{bcs}^{ideal} &= \text{arctg2} \left( \frac{T_{1,2}}{T_{1,1}} \right) \end{aligned} \quad [4.10]$$

Where  $\text{arctg2}$  is the four quadrante arctan function.

The next step in the presented error methodology is to identify and study those physical errors that will move the end-effector from its ideal position and orientation.

#### **4.4.3 Identify physical error components in the design and their relation to the kinematic chain**

##### **4.4.3.1 Identification of physical error components**

As mentioned above, manipulator end-effector errors are related to physical errors. Therefore, to analyse the end-effector positioning and orientation accuracy, the first step is to identify all possible physical errors.

Physical errors are design dependent and are closely related to component tolerances decided in a manipulator design. During manufacturing of manipulator components, the real dimension and geometry of those components must be kept within the permitted limits given by the tolerances. Thus, the higher rank of tolerance that is assigned, the less physical error of component dimension or geometry will result. However, higher rank of tolerances causes higher production cost. So, tolerances are crucial to manipulator end-effector positioning and orientation accuracy, as well as to production cost. In this dissertation, my focus is related to the problems of how to guarantee the required end-effector positioning and orientation accuracy.

Whitney, Lozinsky and Rourke (1986) divided the sources of manipulator errors into *geometric errors* and *non-geometric errors*. Geometric errors were defined as errors in those parameters that define the geometric relationship between the axes of motion, while non-geometric errors are related to the motion and work load of the manipulator system.















Mooring et al. (1991) stated that the geometric errors usually arises in the construction of the manipulator and is a function of the assigned manufacturing tolerances.

Manufacturing tolerances can be classified into dimension tolerances and geometry tolerances, and geometry tolerances can be further classified into form, orientation, location, and run-out tolerances. These four sub-classes of geometry tolerances can be further decomposed into various geometry tolerance items, which are shown in Table 4.2 (ISO 1101, 1983).

**Geometric errors** arise from dimension-, location-, orientation-, form-, and run out errors of the elements that build the kinematic chain.

**Non-geometric errors** are errors related to robot dynamics errors (such as the compliance of joints and links), controller errors, backlashes of gears etc.

**Table 4.2** Items of form, orientation, location, and run-out tolerances

<b>FORM Tolerances</b>	Name: <i>Straightness</i>	Symbol: 
	Name: <i>Flatness</i>	Symbol: 
	Name: <i>Roundness</i>	Symbol: 
	Name: <i>Cylindricity</i>	Symbol: 
	Name: <i>Profile of any line</i>	Symbol: 
	Name: <i>Profile of any surface</i>	Symbol: 
<b>ORIENTATION Tolerances</b>	Name: <i>Parallelism</i>	Symbol: 
	Name: <i>Perpendicularity</i>	Symbol: 
	Name: <i>Angularity</i>	Symbol: 
<b>LOCATION Tolerances</b>	Name: <i>Position</i>	Symbol: 
	Name: <i>Concentricity and Coaxiality</i>	Symbol: 
	Name: <i>Symmetry</i>	Symbol: 
<b>RUN OUT Tolerances</b>	Name: <i>Circular run-out</i>	Symbol: 
	Name: <i>Total run-out</i>	Symbol: 

To summarise the above discussion, for the purpose of identifying physical errors for a given design, which will be used later as input data for the calculating of the real kinematic chain and the end-effector error, I suggest classifying physical errors into the following three classes:

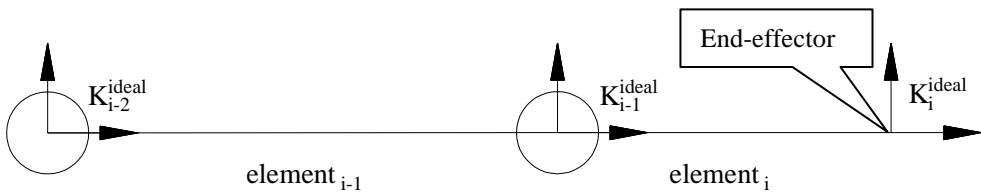
- Joint element related physical errors,
- Link element related physical errors,
- Total assembly related errors.

Certainly, one must further identify physical errors within above three classes of physical errors. For aiding such an identification process, I have worked out a table (Table 4.3), which may be a guide for identification of physical errors within a given design. Namely, while one starts to identify the physical errors upon a given design, the design team may refer to Table 4.3 combined with Table 4.2 to realize what kind of physical error that exists.

**Table 4.3** Serial manipulator (rotation joint) geometric and non-geometric error sources

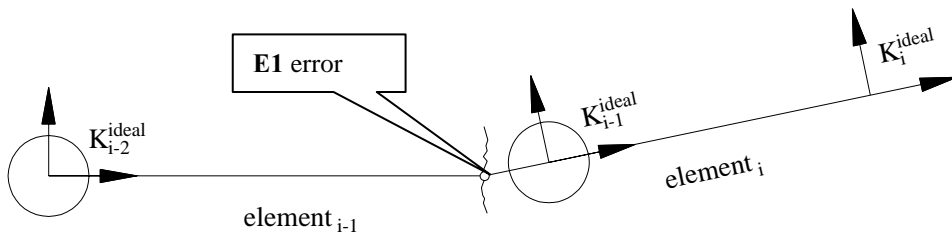
	<i>Error*</i>	<i>Error classification:</i> <i>Geometric</i>	<i>Error classification:</i> <i>Non-Geometric</i>
Joint related errors	Joint housing geometry errors	√	
	Joint housing dimension errors	√	
	Actuator structure errors	√	
	Actuator compliance errors		√
	Gear geometry errors	√	
	Gear transmission errors		√
	Gear backlash		√
	Bearing geometry errors	√	
	Bearing clearance errors		√
	Sensor measuring errors		√
	Computer control errors		√
	Joint sub-assembly errors	√	√
Link related errors	Link dimension errors	√	
	Link geometry errors	√	
	Link compliance errors		√
	Sensor measuring errors		√
	Link sub-assembly errors	√	√
Total assembly errors		√	√

An example is given here to illustrate the identification of physical errors inside the manipulator structure. Figure 4.30 shows a sketch of a manipulator with two joints ( $K_{i-1}$ ,  $K_{i-2}$ ), two links (link  $i-1$ , link  $i$ ), and the end-effector. The configuration of the manipulator is based on the nominally sized position, i.e., it is at “ideal” position without taking physical errors into account. In the Figure 4.31 we see the same manipulator, however this time we can see that there exist at least one physical error source inside the beam. This error is denoted **E1** and is assumed to create a rotational error effect.



**Figure 4.30** A sketch of a manipulator with two joints and an end-effector

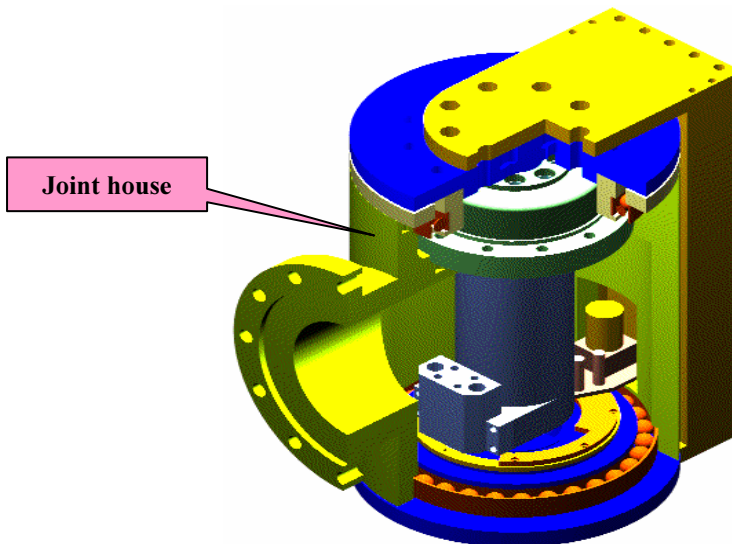
When physical errors are taken into account, the positions of joints, links and the end-effector will deviate from their ideal positions. Figure 4.31 clearly depicts such a deviation.



**Figure 4.31** Physical error **E1** in link element  $i-1$

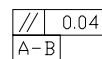
If we look at the detailed design we can identify the origin of the **E1** error, and thereby calculate the numerical value of this physical error. Figure 4.32 shows a detailed view of the link element  $i-1$  and its connecting joint  $K_{i-1}$ .

In the assembly in Figure 4.32, one of the components is the joint house. This is shown in Figure 4.33 as a 2D detailed design drawing. Now, let's take a look at Figure 4.33 as to illustrate the identification of the **E1** physical error.



**Figure 4.32** A design example of a manipulator joint

In the Figure 4.33 the **E1** is identified to arise from the parallelism tolerance,



Parallelism tolerances were identified in Table 4.2 to represent a physical error source.

To calculate the physical error numerical value we take a look at the ISO 1101 (1983) standard which defines that the tolerance plane should lay within two parallel reference planes with a distance  $t$  apart. The reference planes are parallel to the datum line. In Figure 4.34 we see the simplified view of the parallelism tolerance. In this example the tolerance zone  $t = 0.04$  mm, while the diameter  $D_y$  of the flange is 220 mm.

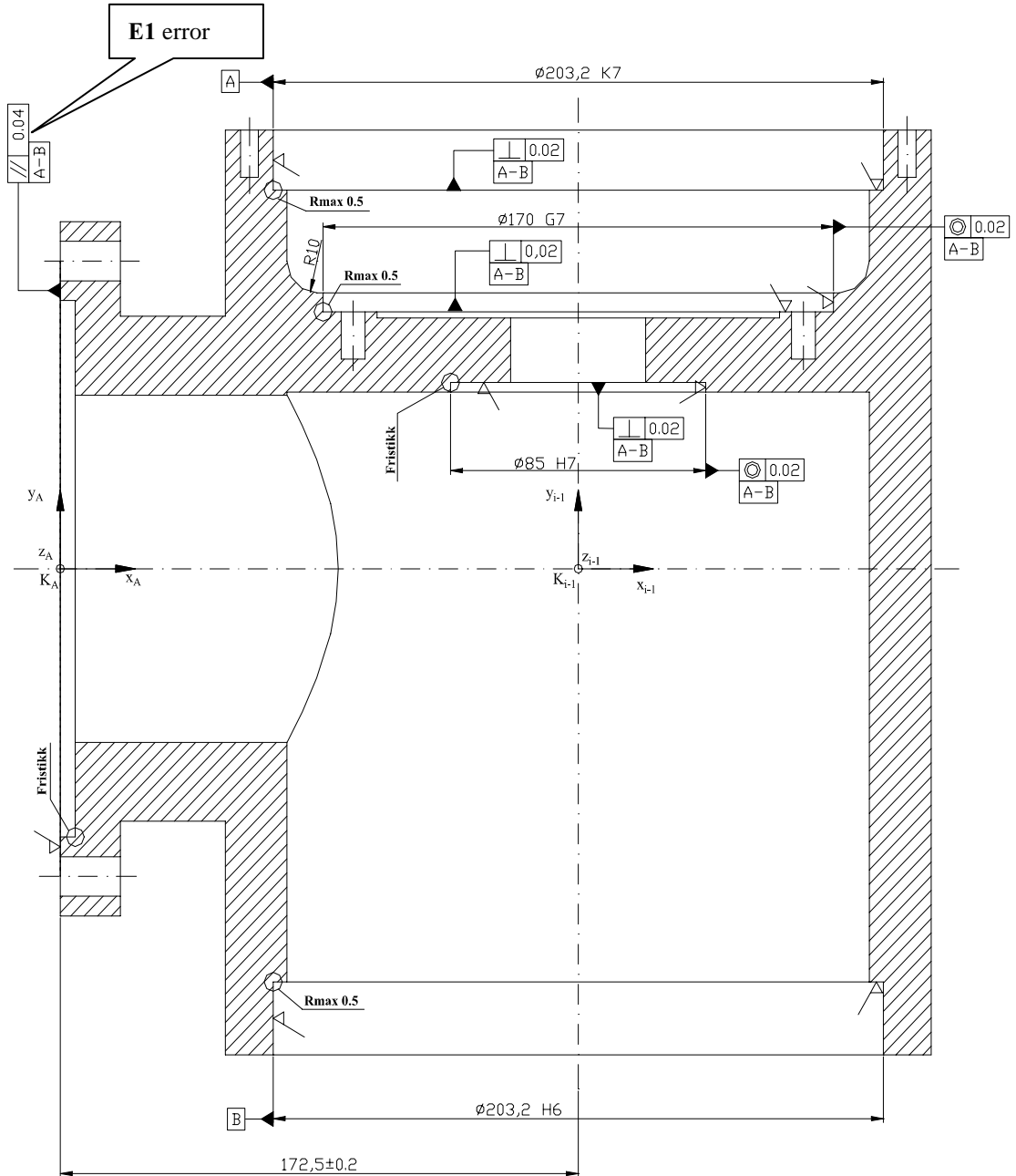


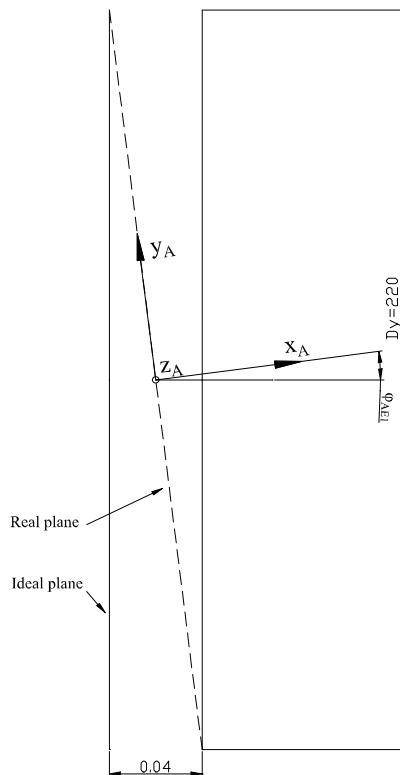
Figure 4.33 A design example, joint housing of a manipulator joint

The maximum orientation error that may arise from this tolerance may either create a rotation around the  $Z_A$  or the  $Y_A$  axis. A combination of both  $Z_A$  and the  $Y_A$  rotation is possible as well but not with maximum values for both rotations.

In both direction the maximum error numerical value of error **E1** is calculated by:

$$\phi_{E1} = \pm \arctg\left(\frac{\varnothing T}{Dy}\right) = \pm \arctg\left(\frac{0.04}{220}\right) \approx \pm 1.04 \cdot 10^{-2} [\text{deg}] \quad [4.11]$$

In this equation the  $Dy$  is the key parameter for the magnitude of  $\phi_{E1}$ . Keeping  $\varnothing T$  as a constant, a larger  $Dy$  gives a less orientation error  $\phi_{E1}$ .



**Figure 4.34** Maximum orientation error due to parallelism tolerance

One may argue that it is possible that the parallelism error will create not a rotary error but a linear error only. Such a situation arises when the manufactured plane is equal to one of the ideal planes. In such a situation we can expect an movement of the  $K_A$  system in the  $x_A$  direction. However, this displacement is incorporated in the location tolerance of the plane. In the lower part of Figure 4.33 we see the location tolerance  $172 \pm 0.2$  which includes the possible linear error arising from the parallelism tolerance.

The next question to be answered is to explain the idea of the coordinate system  $K_A$  which the error is referred to. This will be discussed in the next paragraph.

To sum up, physical errors are identified within the given design, their physical value can be calculated with reference to a selected coordinate frame.

Chapter 5 gives a case study of the error identification /modelling process and many more examples can be studied in this chapter.

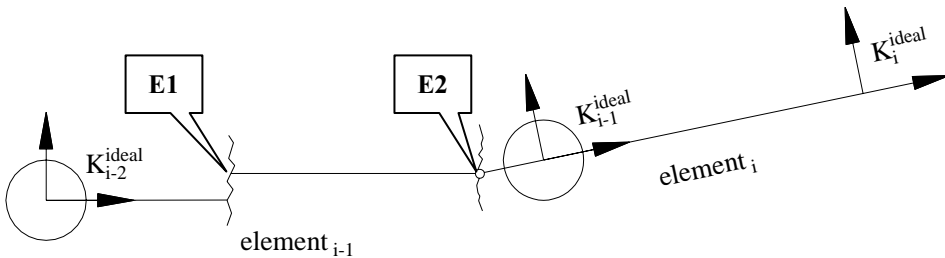
**4.4.3.2 Physical errors and their relation to the kinematic chain**

Vukobratovic and Borovac (1995) state, about mechanisms described by the Denavit-Hartenberg notation, that the parameters used for the mathematical description of the links do not correspond to the measures employed in the manufacturing, thus a question should be addressed. Which parameter should be varied to correspond to the prescribed manufacturing tolerance to be realized in the manufacturing process?

In my discussion about kinematic methodologies in section 4.4.2, I found that a 6 parameter “general” description was less restrictive with respect to location of the mechanism coordinate systems. Thus any error can more easily be related to the kinematic chain.

To further simplify the relationship between kinematic parameters and the physical arm, I suggest increasing the kinematic chain with one extra set of coordinate system for every error identified. With such a methodology I gain the effect of transferring the kinematic chain onto the location of the error source.

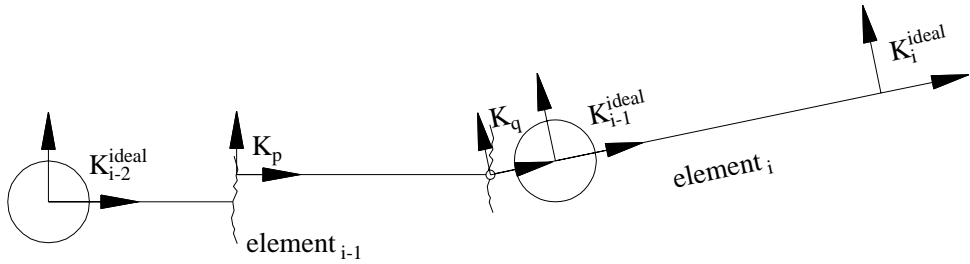
In the Figure 4.35 we again see the simplified structure of Figure 4.30, this time with two errors assigned **E1** and **E2**.



**Figure 4.35** Physical error **E1**, **E2** in link element i-1

According to the demand of assigning kinematics with errors I introduce two new coordinate systems,  $K_p$  at the origin for the **E1** error, and  $K_q$  at the origin for the **E2** error. The idea is that the coordinate systems should incorporate the errors into the kinematic chain. Figure 4.36 shows the new coordinate systems located in the kinematic chain.

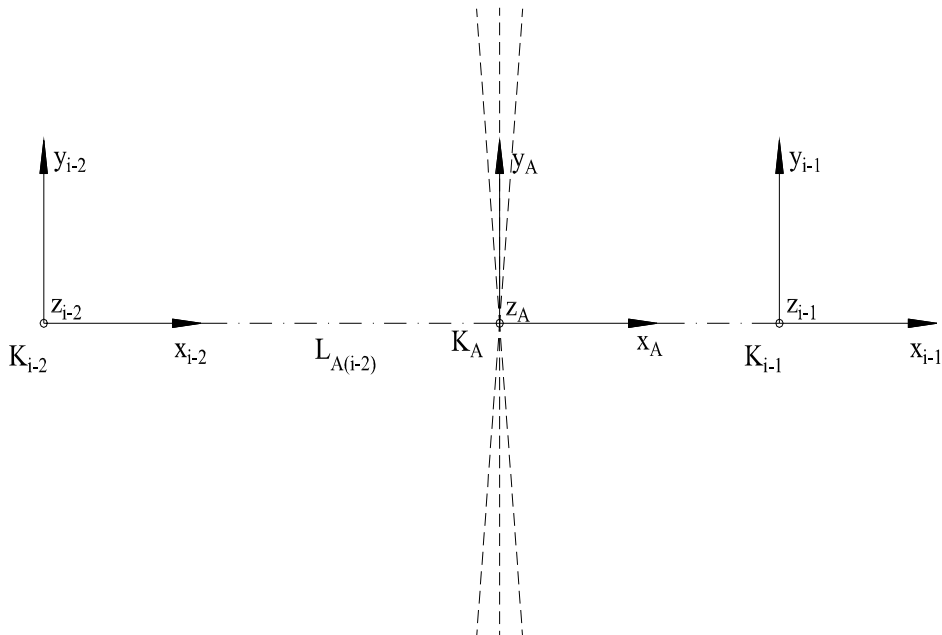




**Figure 4.36** Coordinate assignment to error sources **E1** and **E2**

The “general” notation is used while assigning coordinate systems. The assignment of coordinate systems onto the error sources are made with a similar approach as for the joints systems, as described in section 4.4.2. The error sources may in this context be viewed upon as small joints (linear or rotary) with a joint activation equal to the error numerical value.

If we look back to the parallelism error in the previous section, we now have clarified the usage of the  $K_A$  coordinate system assignment in Figure 4.33 and 4.34. In Figure 4.37 all three coordinate systems representing our example are shown.



**Figure 4.37** Example coordinate assignment

To describe the relationship with the kinematic chain we have to define the transformation matrix between the  $K_{i-2}$  and the  $K_A$  system. Such a transformation contains the linear constant element  $L_{A(i-2)}$  and the variable rotational part  $\phi_{E1}$ . For the

description of the rotational part (the actual error part) there were three possible solutions, either a rotation around  $z_{i-2}$  axis, or a rotation around  $y_{i-2}$  axis, or a combined rotation around both axes.

In order to define the transformation matrix between  $K_{i-2}$  and  $K_A$ , we can use the following three possible transformations:

1. Assuming rotation around axis  $z_{i-2}$ , C rotation
2. Assuming rotation around axis  $y_{i-2}$ , B rotation
3. Assuming a combined rotation around both axes, C followed by B rotation

For all of these cases we can use the general transformation matrix, as described in equation 4.6.

In the first case, rotation around  $z_{i-2}$  axis only, we denote  $\phi_{E1}$  to be  $\phi_{E1C}$  as to say that this is a C rotation. A and B rotations are 0. Thus, by inserting into equation 4.6 we get the transformation matrix:

$$T_{A(i-2)} = \begin{bmatrix} c\phi_{E1C} & s\phi_{E1C} & 0 & 0 \\ -s\phi_{E1C} & c\phi_{E1C} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ L_{A(i-2)} & 0 & 0 & 1 \end{bmatrix} \quad [4.12]$$

In the second case, rotation around  $y_{i-2}$  axis only, we denote  $\phi_{E1}$  to be  $\phi_{E1B}$  as to say that this is a B rotation. C and A rotations are 0. Thus, by inserting into equation 4.6, we get the transformation matrix:

$$T_{A(i-2)} = \begin{bmatrix} c\phi_{E1B} & 0 & -s\phi_{E1B} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E1B} & 0 & c\phi_{E1B} & 0 \\ L_{A(i-2)} & 0 & 0 & 1 \end{bmatrix} \quad [4.13]$$

In the third case, rotation around both axes, consecutive rotation C and B, we still use the same notations of  $\phi_{E1C}$  and  $\phi_{E1B}$ . However, maximum values of  $\phi_{E1}$  could not be assigned to either  $\phi_{E1C}$  or  $\phi_{E1B}$ . A rotation is 0. Thus, by inserting into equation 4.6 we get the transformation matrix:

$$T_{A(i-2)} = \begin{bmatrix} c\phi_{E1B} \cdot c\phi_{E1C} & c\phi_{E1B} \cdot s\phi_{E1C} & -s\phi_{E1B} & 0 \\ -s\phi_{E1C} & c\phi_{E1C} & 0 & 0 \\ s\phi_{E1B} \cdot cc\phi_{E1C} & s\phi_{E1B} \cdot sc\phi_{E1C} & c\phi_{E1B} & 0 \\ L_{A(i-2)} & 0 & 0 & 1 \end{bmatrix} \quad [4.14]$$

In the sense of finding maximum error onto the end-effector, equations 4.12 and 4.13 are more representative than equation 4.14.

In this section we have seen that by adding a coordinate system to the identified error, we can describe the relationship between the error and the kinematic chain, more easily. Only one coordinate system was used to describe each error source. However, if this is not enough to fully relate the kinematic chain onto the identified error as many coordinate systems as needed (or wanted) could be added to the methodology. As an example we could describe the relationship between a reference coordinate system and the error, with one matrix describing the ideal position/orientation of the error source with respect to the reference, and one matrix describing the relationship between ideal position/orientation of the error source and the real position/orientation after the error has taken its effect.

If we apply such an effect to the above described example, for the case of C rotation only, described with equation 4.12:

First a transformation between the reference  $K_{i-2}$  and the ideal location/orientation of the  $K_A$  system. This is a linear transformation where the linear parameter is  $L_{A(i-2)}$ . By inserting  $L_{A(i-2)}$  into equation 4.6, letting  $C=B=A=0$  we get the following matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ L_{A(i-2)} & 0 & 0 & 1 \end{bmatrix}$$

The next step is to carry out the transformation between the ideal location/orientation of the  $K_A$  system and the final location/orientation after the error has taken its effect. This matrix becomes:

$$\begin{bmatrix} c\phi_{E1C} & s\phi_{E1C} & 0 & 0 \\ -s\phi_{E1C} & c\phi_{E1C} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By combining these matrixes together we get the same result as equation 4.12

$$\begin{bmatrix} c\phi_{E1C} & s\phi_{E1C} & 0 & 0 \\ -s\phi_{E1C} & c\phi_{E1C} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ L_{A(i-2)} & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\phi_{E1C} & s\phi_{E1C} & 0 & 0 \\ -s\phi_{E1C} & c\phi_{E1C} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ L_{A(i-2)} & 0 & 0 & 1 \end{bmatrix}$$

So, the designer may choose as many coordinate system as needed, or even wanted for the matter of simplification.

In the case study in Chapter 5, the latter methodology is chosen, describing the relationship between a reference coordinate system and the error:

- With one matrix describing the ideal position/orientation of the error source with respect to the reference, and
- One matrix describing the relationship between ideal position/orientation of the error source and the real position/orientation after the error has taken its effect.

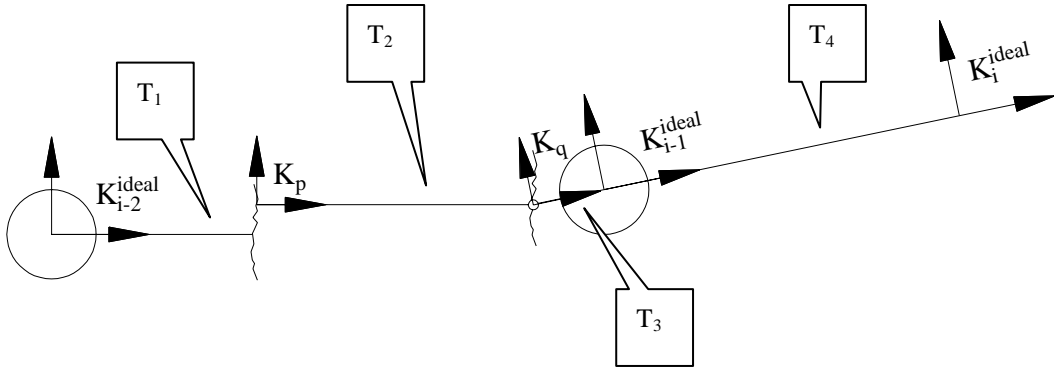
#### 4.4.4 Identify the real kinematic chain, and the real end-effector pose

Results from the error identification and their relationship with the kinematic chain, as described in the section 4.4.3, are used to identify the real kinematic chain. After all error have been identified and the error matrixes established, we simply multiply matrixes together, in the same manner as for the ideal chain, as described in section 4.4.2. The general equation for the description of the real kinematic chain will then be:

$$T^{real} = T_n \cdot T_{n-1} \cdot \dots \cdot T_1 = \prod_{i=n}^1 T_i \quad [4.15]$$

Indexing n in the above equation describes the transformation matrix between coordinate frame n-1 and n. Numbering of coordinate frames start with the base with index 0 and ends with the end-effector with numbering n.

In Figure 4.38 again the simplified manipulator structure is shown. Frames are assigned and the relationship between them determined with the respective transformation matrixes.



**Figure 4.38** Coordinate assignment to error sources **E1** and **E2**

Applying the defined matrixes onto equation 4.15 we find the expression for the real kinematic chain as:

$$T^{real} = T_4 \cdot T_3 \cdot T_2 \cdot T_1$$

The size of the matrix arising from equation 4.15 is dependant of the number of errors identified. Some may argue that this will create a very large work load upon the unit which carries out the numerical calculations, but with today's new microprocessors, large amount of data can be calculated within a time instant. Also, after manufacturing and prototyping the numerical value of error components could be identified, and directly put into the kinematic chain, as constants. The kinematic chain could then be reanalyzed and a simplified structure adapted for real time calculations.

End-effector real position and orientation,  $(EEp_{bcs})^{real}$ , is derived from the structure real kinematic chain matrix 4.15, similar to the ideal end-effector position and orientation,  $(EEp_{bcs})^{ideal}$  as described in section 4.4.2:

$$EEp_{bcs}^{real} = \begin{bmatrix} X_{bcs}^{real} & Y_{bcs}^{real} & Z_{bcs}^{real} & A_{bcs}^{real} & B_{bcs}^{real} & C_{bcs}^{real} \end{bmatrix} \quad [4.16]$$

The real position of the end effector, with respect to the base coordinate system is identified as follows:

$$\begin{aligned} X_{bcs}^{real} &= T_{4,1}^{real} \\ Y_{bcs}^{real} &= T_{4,2}^{real} \\ Z_{bcs}^{real} &= T_{4,3}^{real} \end{aligned} \quad [4.17]$$

Where  $T_{i,j}^{real}$  is elements of the matrix  $T^{real}$  (4.15)

$$T^{real} = \begin{bmatrix} T_{1,1} & T_{1,1} & T_{1,1} & 0 \\ T_{2,1} & T_{2,2} & T_{2,3} & 0 \\ T_{3,1} & T_{3,2} & T_{3,3} & 0 \\ T_{4,1} & T_{4,2} & T_{4,3} & 1 \end{bmatrix}$$

The Euler angles A, B and C can be derived:

$$\begin{aligned} A_{bcs}^{real} &= \arctg2\left(\frac{T_{2,3}}{T_{3,3}}\right) \\ B_{bcs}^{real} &= -\arcsin(T_{1,3}) \\ C_{bcs}^{real} &= \arctg2\left(\frac{T_{1,2}}{T_{1,1}}\right) \end{aligned} \quad [4.18]$$

Where  $\arctg2$  is the four quadrante arctan function.

According to Figure 4.26 we have now completed the three steps in the ERROR IDENTIFICATION AND MODELLING process. Thus, in what follows I will continue with the second part of the error analysis methodology, namely the ERROR ANALYSIS with its two steps.

For both of the steps of the error analysis part of Figure 4.26 we use the same equation 4.19.

$$dEEp_{bcs} = EEp_{bcs}^{real} - EEp_{bcs}^{ideal} \quad [4.19]$$

The end- effector error  $dEEp_{bcs}$  is found by subtracting the ideal pose from the real pose. The equation from the ideal position/orientation was deducted earlier in section 4.4.2 as equations 4.9 and 4.10. The equation from the real position/orientation was deducted above in section 4.4.4 as equations 4.17 and 4.18.

#### 4.4.5 Calculate influence of single error components to the end-effector

In this part the designer may use equation 4.19 to calculate the effects of the error sources, one by one. In such a situation, only one error component is given a numerical value, while the other errors are assigned with zeroes only. The effect of the zero assignment is that these error matrixes will become unit matrixes and no transformation is taking place.

The designer can test error by error and even experience with different values of the selected error, so to decide what tolerance should/could be assigned to this physical

part of the construction and at the same time having full control of the effect of the tolerance assignment onto the end-effector. Such a capability that enables an easy access to changing values of physical error components and see the resulting end-effector error create a very good environment for the designer for experimenting with his design.

#### **4.4.6 Calculate influence of all or a selection of error components to the end-effector**

Calculation of multiple error components are as easily as for the single error calculations. Just select the errors to be analyzed, all other components are zeroed, then the equation 4.19 gives the demanded result. The equation 4.19 therefore allows us to study the combinatorial effect from several errors onto the end effector. Such a result is not the same as just summing up the single errors calculations carried out in the previous section. The reason for this is that the influence that one error has upon the end-effector is dependant onto the other errors closer to the base of the manipulator. In the serial structure chain different errors are dependant with each other in most situations.

Mavroidis et al. (1997) developed a methodology where the calculated repeatable errors were summed together based on single error components, such calculations will result in loss of precision in the calculations of the end-effector error.

By selecting worst case numerical errors to all error components we get the worst case end-effector position/orientation error. If the calculated result is still within our precision requirement we can keep the design as it is. If the result is out of our requirements, a redesign will occur according to the methodology presented earlier in this chapter.

In general, we may say that repeatable errors can, if possible to exactly identify after manufacturing and prototyping, be used to update the ideal kinematic chain so that the effect of the repeatable error is eliminated. Random errors are due to their nature impossible to remove in such a manner, component redesign is a natural step to deal with random errors.

#### **4.4.7 Other error modelling methods**

As stated in the section 4.4.1, it is important to identify and study other error analysis methodologies to clearly see the necessity of developing the presented error methodology.

Hollerbach (1988) stated that many error modelling methodologies have been using a four parameter Denavit–Hartenberg presentation for kinematic description and error modelling. However, as identified in section 4.4.2 this is a difficult approach due to the restrictions of the Denavit-Hartenberg notation. From a design point of view it is difficult to relate the D-H kinematic parameters to the physical chain. Thus the usage

of four parameter D-H notation in the error modeling methodology clearly has some weakness.

Hollerbach (1988) further stated that several authors, starting with Hayati (1983), identified the weakness with the 4 parameter D-H presentation, thus an extra parameter was added to overcome the weakness from the four parameter D-H notation. Veitchegger and Wu (1986) state that the D-H four parameter will not be adequate to define parallel or near parallel joints of a robot with kinematic errors, an extra term is added to the D-H notation to compensate for the errors in parallel or near parallel joints. Thus, in the methodology presented by Veitchegger and Wu (1986) the ability of the manipulator to reach a specific pose is dependant onto the accuracy of these five parameters (4 D-H and one extra rotational term). Again, such a methodology supposes that the physical errors can be related to these five parameters.

Some of the newest research in the error modelling area has been presented by Mavroidis et al. (1997), Mavroidis, Flanz, Dubowsky, Drouet, Goitein (1998). They have been working with error modelling of a high precision medical robot.

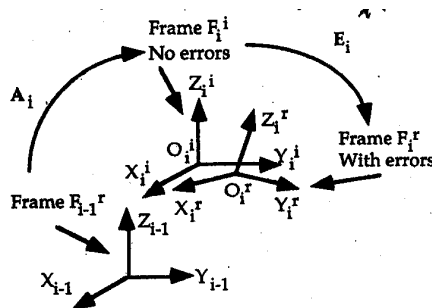


Figure 4.39 Frame displacement due to errors

The main idea behind their methodology is to use the D-H notation to describe the relationship of a beam/joint element, assuming that this element has no error. This will represent the ideal position of frame  $F_i^i$  in Figure 4.39.

But physical errors change the geometric properties of a manipulator. As a result, the frames defined at the manipulator joints are slightly displaced from their expected, ideal locations. In Figure 4.39, frame  $F_i$  is shown in the ideal location  $F_i^i$  and in its real location  $F_i^r$  due to errors.

The position and orientation of a frame  $F_i^r$  with respect to its ideal location  $F_i^i$  is represented by a 4x4 homogeneous matrix  $E_i$ . The rotation part of matrix  $E_i$  is the results of the product of three consecutive rotations  $e_{si}$ ,  $e_{ri}$ ,  $e_{pi}$  around the Y, Z and X axes respectively. (These are the Euler angles of  $F_i^r$  with respect to  $F_i^i$ ). The subscripts s, r, and p represent spin (yaw), roll, and pitch, respectively. The translational part of matrix  $E_i$  is composed of the 3 coordinates  $e_{xi}$ ,  $e_{yi}$ ,  $e_{zi}$ . So the  $E_i$  matrix actually has a



six parameter representation. Thus, this methodology combines both D-H notation and a 6 parameter “general” notation.

The 6 parameters  $e_{xi}$ ,  $e_{yi}$ ,  $e_{zi}$ ,  $e_{si}$ ,  $e_{ri}$ , and  $e_{pi}$  are called here “generalized error” parameters. For a six degrees of freedom manipulator, there are 36 generalized errors which can be written in vector form as  $\varepsilon = [\dots, e_{xi}, e_{yi}, e_{zi}, e_{si}, e_{ri}, e_{pi}, \dots]$ , with  $i$  ranging from 1 to 6 (Mavroidis et al., 1997).

So, all of the possible error sources inside one joint will contribute to the transformation from the ideal to the real position, and this transformation should be described with the 6 generalized parameters. However, the physical errors do not directly relate to these parameters, so it forces the designer to transform every physical error into the 6 parameter description. For a given error source these 6 generalized parameters must be changed whenever the error numerical value changes. For a designer who wants to test the effect of different tolerances this will be very cumbersome process. Also, all combinations of errors will create different set of generalized parameters within one joint/beam system and the designer must identify these relationships.

The error methodology presented in this dissertation assigns 6 parameters to every error inside the chain, with the goal to describe every error as a natural part of the kinematic chain. This gives a clear relationship between the different error sources and the kinematic parameters. After the chain has been identified, the designer has finished his job, and he/her can put their effort to the error analysis part.

Thus, from the above reasoning I find that the presented methodology in section 4.4 will be helpful for the designer and his/her work with error modelling and analysis.

#### 4.4.8 Summary of section 4.4

The purpose of section 4.4 is to set up a method relevant for calculating numerical values of the end-effector error. For this purpose, a review on Denavit-Hartenberg convention combined with a survey on exiting error modelling methods has been carried out. Based on these, I found that a new unified general method for calculating numerical values of the end-effector error is needed.

The presented error analysis methodology has the following aspects. The first aspect is that by using the “general” convention of assigning reference frames for each joints, end-effector, and base frame the “ideal kinematic chain” can be located within the manipulator physical structure itself.

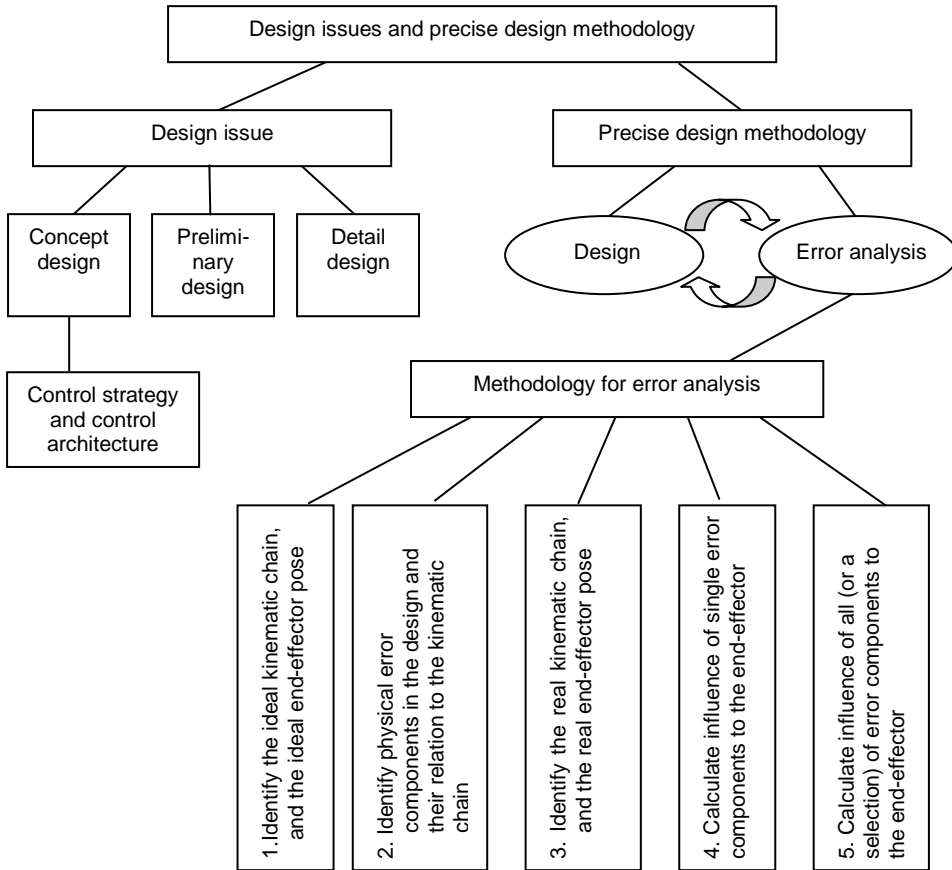
The second is that by assigning a transformation matrix (one or several) to every error, all errors are modelled into the kinematic chain. Thus, the relationship between the physical construction and the kinematic parameters become apparent.

The third aspect is that the presented method allows for combinatorial error analysis without losing precision or having to make cumbersome calculations.

The fourth is that the presented methodology will give the designer a tool which is easy to handle and this will promote the very important design process of evaluation.

### 4.5 Conclusion of this chapter

The content of this chapter can be summarized with the Figure 4.40.



**Figure 4.40** A summarized sketch of Chapter 4

The purpose of this chapter has been to discuss the design issues and conceive a precise design methodology for workplace constraint underwater manipulators.

The discussion starts from a review on design procedure of underwater manipulators. A discussion on procedures of the conceptual design phase, the preliminary design phase and the detail design phase is conducted (see upper left of Figure 4.40), and two suggestions are raised.

The first suggestion is that the tasks of CONTROL STRATEGY and CONTROL ARCHITECTURE are recommended to be put in the conceptual design phase (see the upper left of Figure 4.40 and Figure 4.2), because the results of the control strategy design and control architecture design, together with the KINEMATIC STRUCTURE design (see Figure 4.2) will give out a conceptual picture (or say, an overall outline) of the future manipulator to be designed. Control strategies and architectures are studied in detail in section 4.3.

The second suggestion is that a detail analysis (i.e., error analysis) task is inserted into the detail design phase after the tasks of MECHANICAL DESIGN and ELECTRICAL & CONTROL DESIGN (see Figure 4.4) for the purpose of precise design. Thus, an iterative method of “design – error analysis – design – error analysis...” is suggested for the precise design (see the upper right of Figure 4.40).

The section 4.4 of this chapter has continued discussing and developing a methodology for error analysis purpose, which has been summarized in section 4.4.8.

Figure 4.40 gives an overview of my theoretical research results on design issues and precise design methodology for developing workspace constraint underwater manipulators. In the next chapter, a case study will be given to show how the theoretical results are to be used in the area of error modelling and analysis.

Finally, in Chapter 6 an experimental error mapping methodology is presented. In section 4.2.1.3 and section 4.2.2 this activity was pointed to as one of the two tasks within the PROTOTYPE ANALYSIS task.



## Chapter 5

### Case study of serial structure manipulator, accuracy analysis

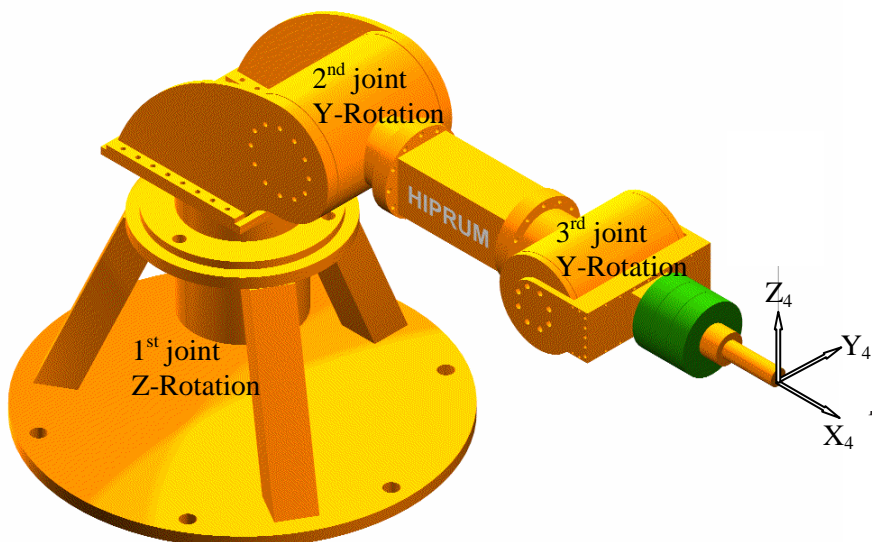
#### 5.1 Introduction

In the previous chapter, a method for accuracy analysis of a serial manipulator arm was given. In this chapter, a case study of a serial manipulator design follows, aiming towards practical usage of the theory previously presented.

The error analysis methodology, as stated in the Chapter 4, is a 5 step procedure as follows:

- 1<sup>st</sup> Identify the ideal kinematic chain, and the ideal end effector position,
- 2<sup>nd</sup> Identify physical error components in the design and their relation to the kinematic chain,
- 3<sup>rd</sup> Identify the real kinematic chain, and the real end-effector position,
- 4<sup>th</sup> Calculate influence of single error components to the end-effector,
- 5<sup>th</sup> Calculate influence of all (or a selection) of error components to the end effector.

In the following, the methodology above is carried out, step by step.

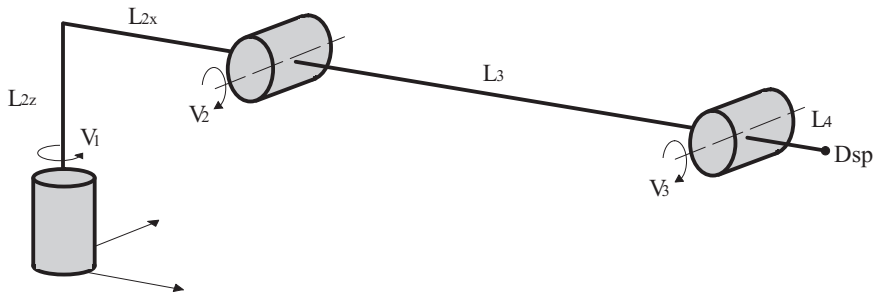


**Figure 5.1** Design case

To do this we need a real design, and Figure 5.1 shows the chosen design, used in the forthcoming analysis.

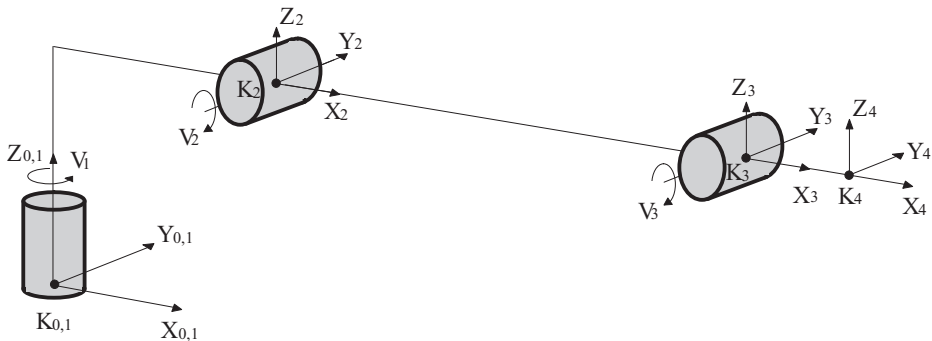
## 5.2 Identification of the ideal kinematic chain, and the ideal end-effector pose

The arm structure, as shown in Figure 5.1, is selected as our case study arm. This arm consists of three rotational joints. Figure 5.2 shows a simplified principle drawing of the same arm.



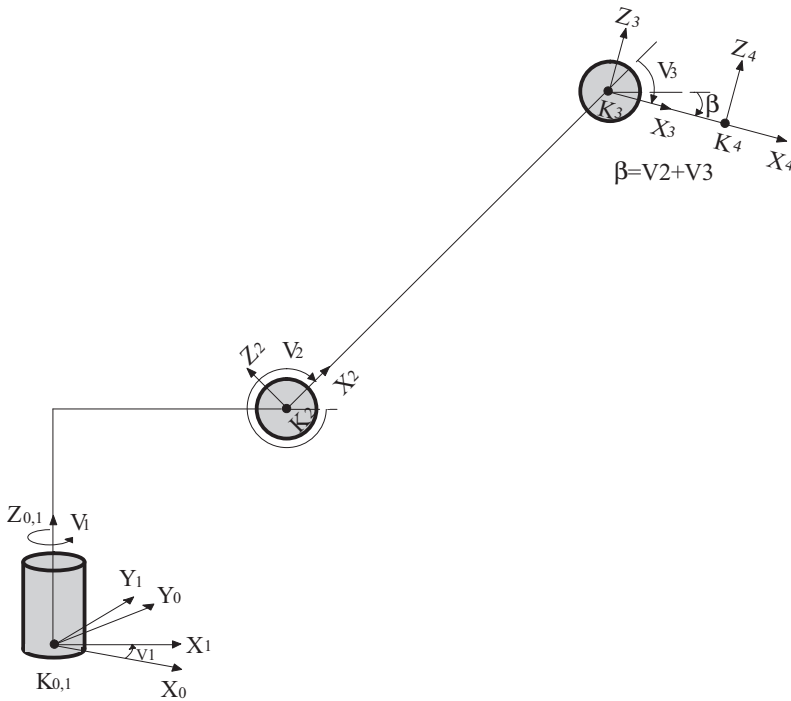
**Figure 5.2** Arm in staging pose

To define the ideal kinematics, firstly a coordinate system is attached to every rotating joint. This is shown in Figure 5.3.



**Figure 5.3** Assignment of coordinate systems

The coordinate systems rotate with the joint activation. Figure 6.3 shows the “movement” of assigned coordinate systems with the joint activation.



**Figure 5.4** Coordinate system responses to joint activation

### 5.2.1 Ideal forward kinematic chain

Based on the Figure 5.1, 5.2 and 5.3, the ideal kinematic parameters are identified and summarized in the Table 5.1.

**Table 5.1** Kinematic ideal parameters

Coordinate Systems	X	Y	Z	A	B	C	Component	
							Fixed	Variable
$K_0 \rightarrow K_1$	0	0	0	0	0	$V_1$	*	$V_1$
$K_1 \rightarrow K_2$	$l_{2x}$	0	$l_{2z}$	0	$V_2$	0	$l_{2x}, l_{2z}$	$V_2$
$K_2 \rightarrow K_3$	$l_3$	0	0	0	$V_3$	0	$l_3$	$V_3$
$K_3 \rightarrow K_4$	$l_4$	0	0	0	0	0	$l_4$	*

The universal transformation matrix, as given in Chapter 4, is used to calculate the transformation matrixes. Table 5.1 gives the necessary input for each matrix

$$T = \begin{bmatrix} cB \cdot cC & cB \cdot sC & -sB & 0 \\ -sC \cdot cA & cC \cdot cA & sA \cdot cB & 0 \\ +sA \cdot sB \cdot cC & +sA \cdot sB \cdot sC & & \\ sA \cdot sC & -sA \cdot cC & cA \cdot cB & 0 \\ +cA \cdot sB \cdot cC & +cA \cdot sB \cdot sC & & \\ X & Y & Z & 1 \end{bmatrix} \quad [5.1]$$

The ideal transformations:

$$\underline{K_0 \rightarrow K_1} \quad T_1 = \begin{bmatrix} cV_1 & sV_1 & 0 & 0 \\ -sV_1 & cV_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \underline{K_1 \rightarrow K_2} \quad T_2 = \begin{bmatrix} cV_2 & 0 & -sV_2 & 0 \\ 0 & 1 & 0 & 0 \\ sV_2 & 0 & cV_2 & 0 \\ l_{2x} & 0 & l_{2z} & 1 \end{bmatrix} \quad [5.2]$$

$$\underline{K_2 \rightarrow K_3} \quad T_3 = \begin{bmatrix} cV_3 & 0 & -sV_3 & 0 \\ 0 & 1 & 0 & 0 \\ sV_3 & 0 & cV_3 & 0 \\ l_3 & 0 & 0 & 1 \end{bmatrix}, \quad \underline{K_3 \rightarrow K_4} \quad T_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l_4 & 0 & 0 & 1 \end{bmatrix}$$

Structure total ideal forward transformation matrix is given by:

$$T^{ideal} = \prod_{i=n}^1 T_i = T_4 \cdot T_3 \cdot T_2 \cdot T_1 \quad [5.3]$$

where  $n$  equals number of transformation matrixes

Matrix multiplication gives the total ideal forward transformation:

$$T^{ideal} = \begin{bmatrix} cV_1 \cdot c\beta & sV_1 \cdot c\beta & -s\beta & 0 \\ -sV_1 & cV_1 & 0 & 0 \\ cV_1 \cdot s\beta & sV_1 \cdot s\beta & c\beta & 0 \\ l_4 \cdot cV_1 \cdot c\beta & l_4 \cdot sV_1 \cdot c\beta & -l_4 \cdot s\beta & \\ +l_3 \cdot cV_1 \cdot cV_2 & +l_3 \cdot sV_1 \cdot cV_2 & -l_3 \cdot sV_2 & 1 \\ +l_{2x} \cdot cV_1 & +l_{2x} \cdot sV_1 & +l_{2z} & \end{bmatrix} \quad [5.4]$$



$$\beta = V_2 + V_3$$

Where:  $c = \text{cosinus}$   
 $s = \text{sinus}$

### 5.2.2 End effector ideal pose

End-effector ideal position and orientation,  $(EEp_{bcs})^{ideal}$ , often called the ideal forward kinematic model is derived from the structure ideal transformation matrix, as described in Chapter 4:

$$EEp_{bcs}^{ideal} = \begin{bmatrix} X_{bcs}^{ideal} & Y_{bcs}^{ideal} & Z_{bcs}^{ideal} & A_{bcs}^{ideal} & B_{bcs}^{ideal} & C_{bcs}^{ideal} \end{bmatrix} \quad [5.5]$$

#### Position:

$$\begin{aligned} X_{bcs}^{ideal} &= (l_4 \cdot c\beta + l_3 \cdot cV_2 + l_{2x}) \cdot cV_1 \\ Y_{bcs}^{ideal} &= (l_4 \cdot c\beta + l_3 \cdot cV_2 + l_{2x}) \cdot sV_1 \\ Z_{bcs}^{ideal} &= -l_4 \cdot s\beta - l_3 \cdot sV_2 + l_{2z} \end{aligned} \quad [5.6]$$

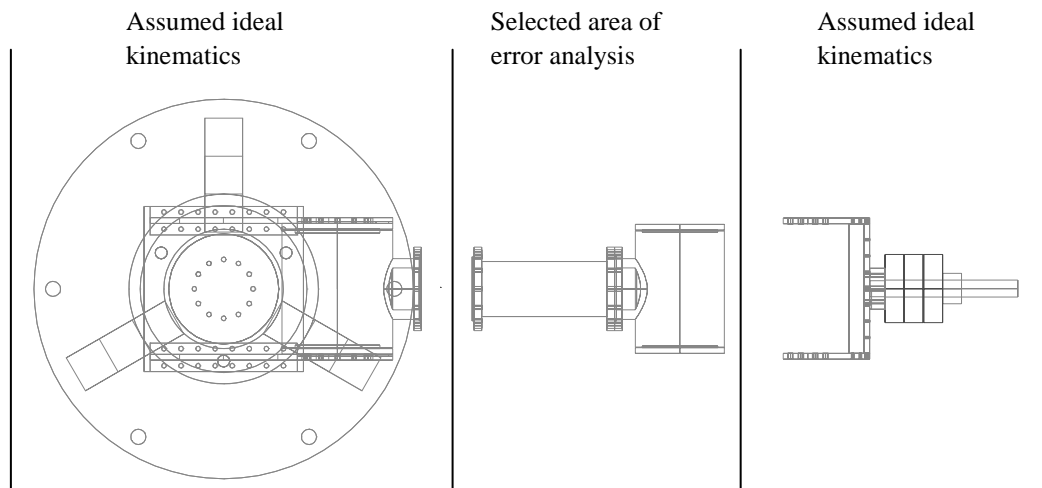
#### Orientation:

$$\begin{aligned} B_{bcs}^{ideal} &= -\arcsin(-s\beta) = \beta = V_2 + V_3 \\ C_{bcs}^{ideal} &= \arctg\left(\frac{sV_1 \cdot c\beta}{cV_1 \cdot c\beta}\right) = \arctg(\tg V_1) = V_1 \quad c\beta \neq 0 \\ A_{bcs}^{ideal} &= \arctg\left(\frac{0}{c\beta}\right) = \arctg(0) = 0 \end{aligned} \quad [5.7]$$

### 5.3 Identify physical errors; identify their relation to the kinematic chain.

In the design of a manipulator system, the outlook of the kinematic chain is influenced by various error sources within the design itself. These error sources were in general analyzed in Chapter 4. In this paragraph we will exemplify these error sources and show their influence to the kinematic chain.

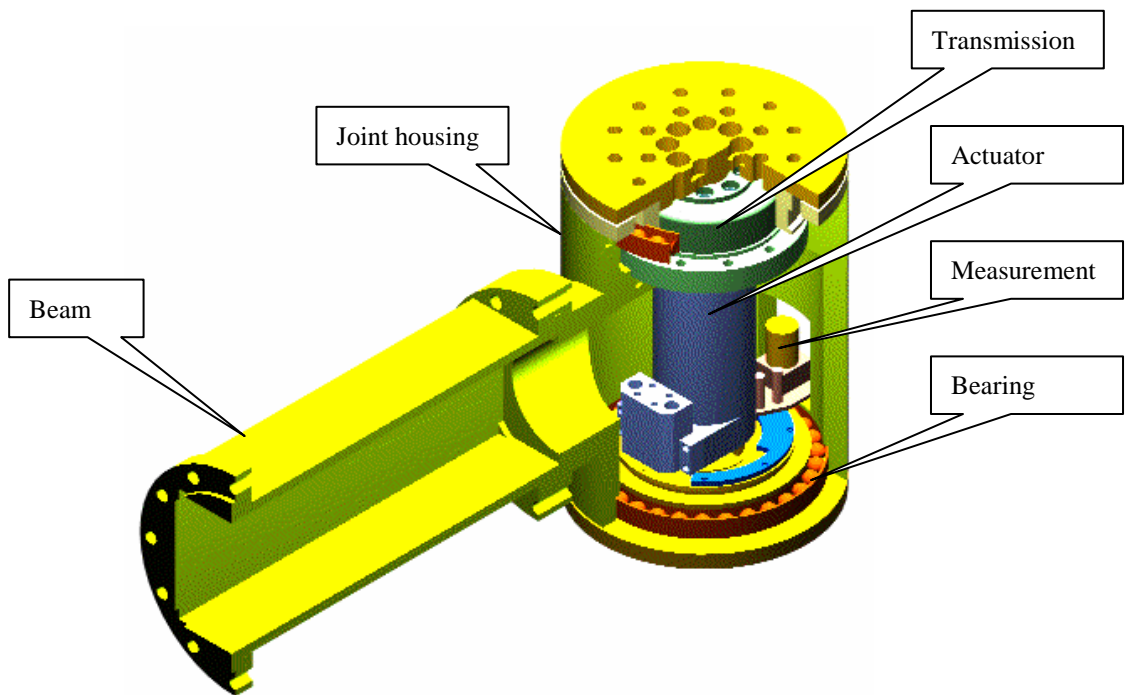
It this study, we select a designated area as our focal point. We select and carry out an analysis of the beam between joint 2/3 and the joint 3 itself. These components are shown in Figure 5.5.



**Figure 5.5** Exploded top view of selected area for error analysis

Referring to Figure 5.3, our selected parts lay within the coordinate transformation  $K_2 \rightarrow K_3$ . All other coordinate transformations  $K_0 \rightarrow K_1$ ,  $K_1 \rightarrow K_2$  and  $K_3 \rightarrow K_4$  are assumed ideal. Naturally, these transformations also consist of errors, but their analysis is similar to the selected system. Thus, the transformations  $K_0 \rightarrow K_1$ ,  $K_1 \rightarrow K_2$  and  $K_3 \rightarrow K_4$  are unchanged, but the transformation  $K_2 \rightarrow K_3$  must be analyzed.

To carry out such an error analysis, we need to look at a detailed design level of the chosen area. Figure 5.6 shows our selected arm/joint element in top view, while Figure 5.6 shows a cut-through of the beam/joint.



**Figure 5.6** Cut-through of selected design with selected internal components

The selected manipulator area is divided into three parts, where typical errors are identified and analyzed with respect to the kinematic chain. The three different parts are:

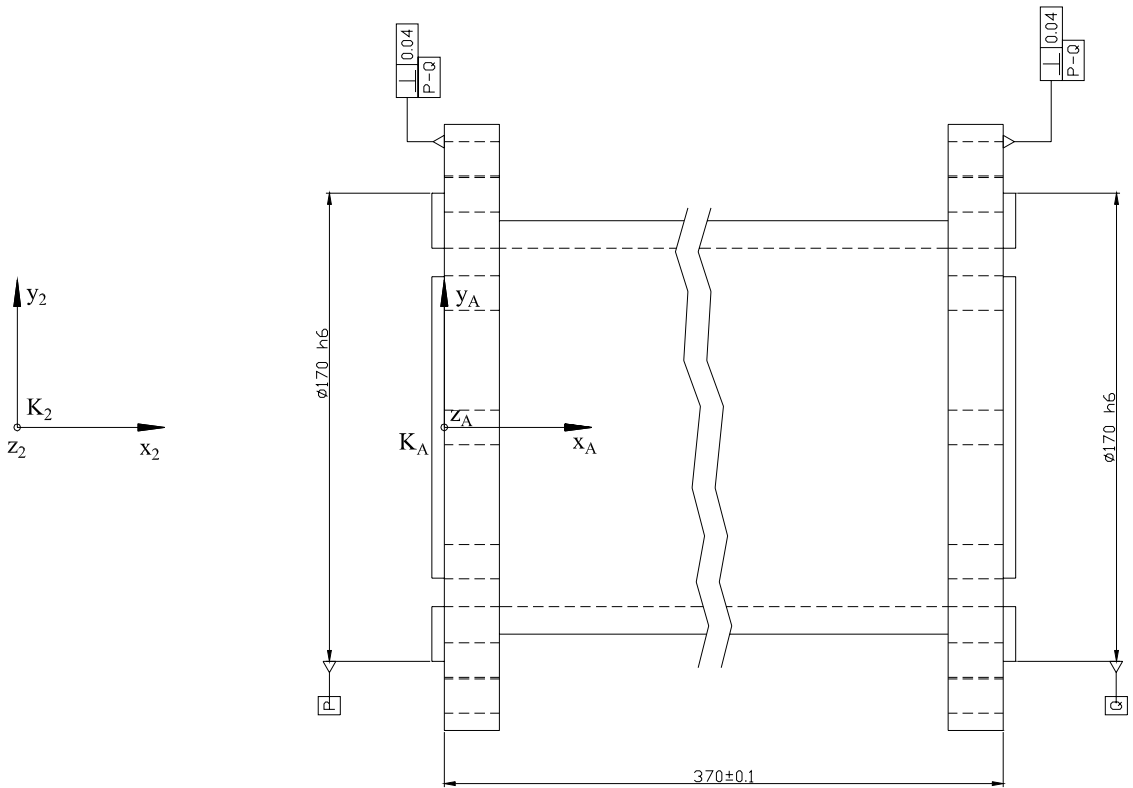
- Beam,
- Joint 3 housing,
- Joint 3 components

### 5.3.1 Error identification of beam element

The error analysis of the beam starts in the left part of Figure 5.6. From this point the first error is identified and a transformation matrix is made to describe this specific error influence to the kinematic chain. Then, moving to the right in Figure 5.6, the next error is found and its contribution to the kinematic chain is identified. When all significant sources are taken into consideration a real kinematic chain of the beam element is gained. Thus, the sequence of appearance of the error sources is important to define the correct kinematic chain.

In the selected design case the beam error sources are identified and their influence to the kinematic chain is calculated.

In Figure 5.7, a top view of the beam element is shown.



**Figure 5.7** Beam element, top view

The  $K_2$  coordinate system is from the previous link/joint system. The transformation from  $K_2$  to  $K_A$  is a translation only determined by  $L_{2-A}$ . The centre line of the beam  $P-Q$  is the design datum line, this reference line is assumed to coincide with the ideal kinematic line in the origin point of  $K_A$ .

The first error in the beam element is denoted **E1**, appearing from the perpendicularity tolerance.

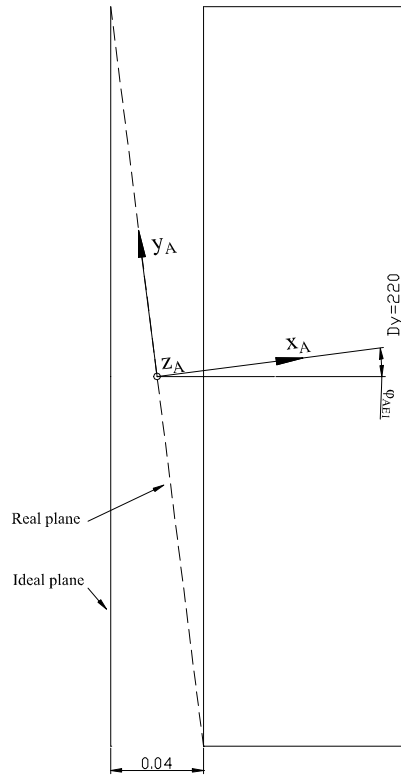


### **The E1 error**

The perpendicularity tolerance is defined according to International standard ISO 1101 (1983).

The tolerance zone (0.04) is defined as the maximum distance between two perpendicular planes which our real plane is expected to lay within.

This principle is shown in the simplified view of Figure 5.8.



**Figure 5.8** Maximum orientation error due to perpendicularity tolerance

The maximum orientation error that may arise from this tolerance may either create a rotation around the  $Z_A$  or the  $Y_A$  axis. A combination of both  $Z_A$  and the  $Y_A$  rotation is possible as well but not with maximum values for both rotations.

In both directions the error numerical value **E1** is calculated by:

$$\phi_{E1} = \pm \arctg\left(\frac{\mathcal{O}T}{Dy}\right) = \pm \arctg\left(\frac{0.04}{220}\right) \approx \pm 1.04 \cdot 10^{-2} [\text{deg}] \quad [5.8]$$

In this equation the  $Dy$  is the key parameter for the magnitude of  $\phi_{E1}$ . Keeping  $\mathcal{O}T$  as a constant, a larger  $Dy$  gives less orientation error  $\phi_{E1}$ .

The transformation of the kinematic chain will be decided by the one of the two rotation matrixes.

$$E1 = \begin{bmatrix} c\phi_{E1} & s\phi_{E1} & 0 & 0 \\ -s\phi_{E1} & c\phi_{E1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ as the rotation around } Z_A \text{ axis or} \quad [5.9]$$

$$E1 = \begin{bmatrix} c\phi_{E1} & 0 & -s\phi_{E1} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E1} & 0 & c\phi_{E1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ as the rotation around axis } Y_A \quad [5.10]$$

Both matrixes will be used in the final calculations of the end effector position, to verify which is the most significant.

The perpendicularity tolerance may also create a linear movement of the coordinate system  $K_A$ , with a maximum value of  $\frac{\phi T}{2}$ . This is not taken into consideration at this point since there is a length tolerance of the beam itself which includes the possible translational error from the perpendicularity tolerance. The beam length tolerance will be analysed later.

The second error in the beam element is denoted **E2**, appearing from the beam compliance.

### **The E2 error**

Compliance analysis may be carried out in both the preliminary and the detailed design stage.

#### **At detailed design stage:**

A compliance analysis of the beam element is a complex task when geometry is advanced combined with a complex load picture. This is a task for finite element method analysis (FEM analysis). FEM analysis in manipulator beam structures is not made a topic in this dissertation. Interested readers may find literature like the Andeen (1988) as a gateway to this topic.

#### **In preliminary design stage:**

In the preliminary design phase, FEM modelling may be too time/cost consuming, or not even possible due to the lack of detail in design. In this stage of the project we will test different solutions to gain an overall idea of what design will work according to our requirements. At this point in the design procedure a simplified worst case analysis of the beam compliance may be enough to determine the effect of the compliance.

In many situations the simplified methodology will be used as a first step analysis in the detailed design as well, to gain understanding of what magnitude we expect our error to lay within. This will give the design team an idea of the importance of the compliance and thereby the necessity to go on with a more detailed FEM analysis. In this chapter the latter methodology is selected as the prior first analysis of the beam compliance.

### Simplified beam analysis:

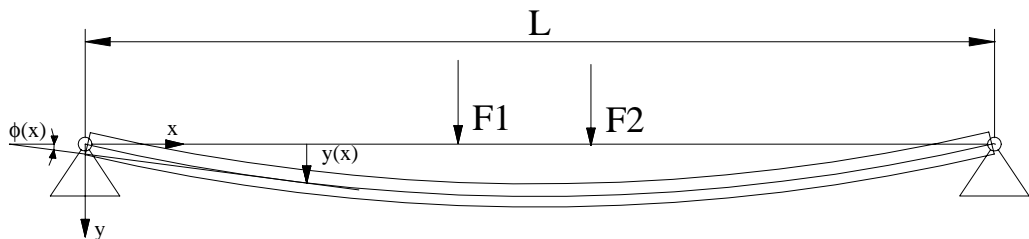
The forthcoming deflection analysis is deduced from the elementary beam theory. This theory is based on 4 hypotheses (Irgens, 1999):

1. Plane cross-sections during deformation,
2. Normal stresses in plane parallel with the beam axis is neglected,
3. Beam material is linearly elastic,
4. The gradient of deformation  $\phi = dy / dx \ll 1$ .

These hypotheses lead to the differential equation for the elastic line. (The deduction itself could be found in various books on beam theory, such as Irgens (1999)).

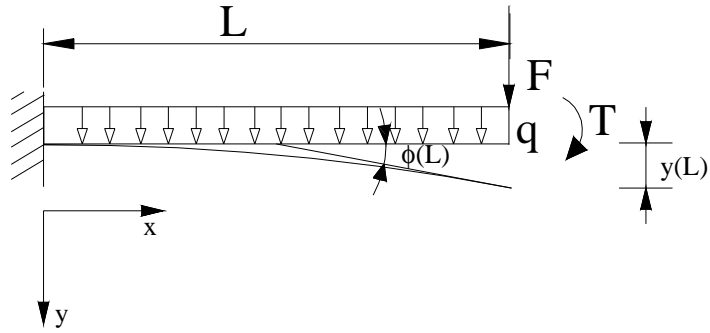
$$\frac{d^2 y}{dx^2} = -\frac{M}{EI} \quad [5.11]$$

where  $E$  is the material elasticity module, while  $I$  is the cross sectional moment of inertia.  $M$  is the bending moment arising from the load model. Figure 5.9 shows parameters in the elastic line model.



**Figure 5.9** Deflection model

In the manipulator design phase the load picture itself could be difficult to predict exactly. However a simplified load picture will also give the design crew important information on the error contribution from the deflection of the manipulator beams. A simplified load picture for initial calculation was suggested by Elle, Johnsen and Lien (1994). This idea is depicted in Figure 5.10.

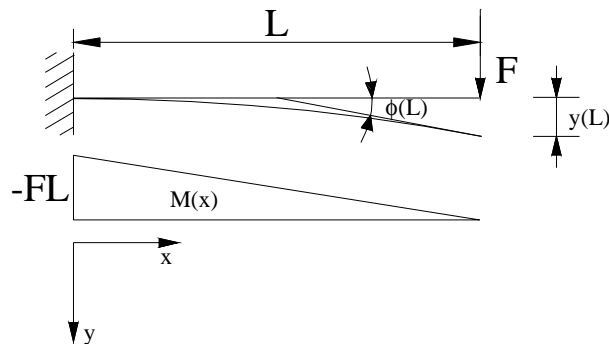


**Figure 5.10** Beam load model

Equation 5.11 is linear with reference to the bending moment  $M(x)$ , this leads to the possible use of the superposition principle (Irgens, 1999). This enables us to calculate the deflection arising from each of the load components  $F$ ,  $q$  and  $T$  separately and then finally sum all three together.

In Figure 5.10 the selected load picture consists of a vertical point force  $F$ , arising from the weight of the connecting motor and gear system. Further  $q$  is defining the weight of the beam itself, while  $T$  is a torque originated from weight of the beam/joint further out in the manipulator construction.

To calculate the total deflection, all three members of the deflection model must be analyzed, starting with single force  $F$  as seen in Figure 5.11.



**Figure 5.11** Beam load model, single force  $F$

The momentum arising from the force  $F$  is calculated by

$$M(x) = -F(L - x) \quad [5.12]$$

which is inserted into the general equation 5.11.



$$\frac{d^2y}{dx^2} = -\frac{M}{EI} = \frac{F(L-x)}{EI} \quad [5.13]$$

This result is integrated as follows

$$\frac{dy}{dx} = \phi(x) = \frac{F}{EI} \left( Lx - \frac{x^2}{2} \right) + C_1 \quad [5.14]$$

In the case of  $x = 0$ ,  $\phi(0) = 0$ , which leads to  $C_1 = 0$ . The end point angle, due to the force  $F$  is then identified by using equation 5.14 with  $x = L$ .

$$\phi_F(L) = \frac{FL^2}{2EI} \quad [5.15]$$

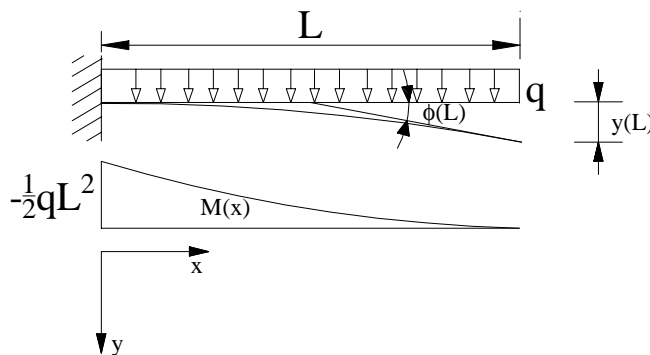
To calculate the deflection  $u(x)$  equation 5.14 is further integrated

$$y(x) = \frac{F}{EI} \left( \frac{Lx^2}{2} - \frac{x^3}{6} \right) + C_2 \quad [5.16]$$

In the case of  $x = 0$ ,  $y(0) = 0$ , which leads to  $C_2 = 0$ . The end point deflection, due to the force  $F$  is then identified by using equation 5.16 with  $x = L$ .

$$y_F(L) = \frac{FL^3}{3EI} \quad [5.17]$$

The analysis of the effect from the beam weight  $q$  is the next step in the deflection analysis. Figure 5.12 identifies the load picture.



**Figure 5.12** Beam load model, beam weight  $q$

The momentum arising from the weight  $q$  is calculated by

$$M(x) = -\frac{q}{2}(L-x)^2 \quad [5.18]$$

Which is inserted into the general equation 5.11

$$\frac{d^2y}{dx^2} = -\frac{M}{EI} = \frac{q(L-x)^2}{2EI} \quad [5.19]$$

This result is integrated as follows

$$\frac{dy}{dx} = \phi(x) = \frac{q}{2EI}(L^2x - Lx^2 + \frac{x^3}{3}) + C_1 \quad [5.20]$$

In the case of  $x = 0$ ,  $\phi(0) = 0$ , which leads to  $C_1 = 0$ . The end point angle, due to the weight  $q$  is then identified by using equation 5.20 with  $x = L$ .

$$\phi_q(L) = \frac{qL^3}{6EI} \quad [5.21]$$

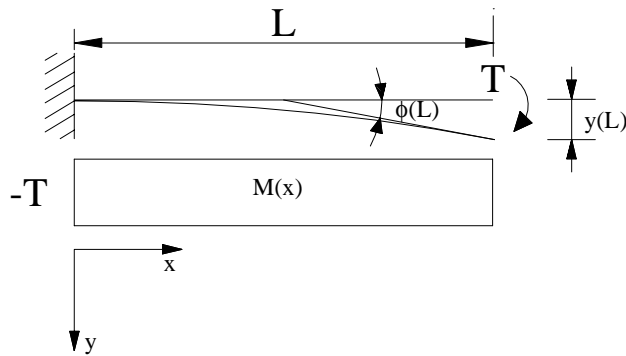
To calculate the deflection  $u(x)$  equation 5.20 is further integrated

$$y(x) = \frac{q}{2EI}(\frac{L^2x^2}{2} - \frac{Lx^3}{3} + \frac{x^4}{12}) + C_2 \quad [5.22]$$

In the case of  $x = 0$ ,  $y(0) = 0$ , which leads to  $C_2 = 0$ . The end point deflection, due to the weight  $q$  is then identified by using equation 5.22 with  $x = L$ .

$$y_q(L) = \frac{qL^4}{8EI} \quad [5.23]$$

The analysis of the effect from the torque  $T$  is the next step in the deflection analysis. Figure 5.13 identifies the load picture.



**Figure 5.13** Beam load model, torque  $T$

The momentum arising from the torque  $T$  is a constant

$$M(x) = -T \quad [5.24]$$

Which is inserted into the general equation 5.11

$$\frac{d^2 y}{dx^2} = -\frac{M}{EI} = \frac{T}{EI} \quad [5.25]$$

This result is integrated as follows

$$\frac{dy}{dx} = \phi(x) = \frac{T}{EI} x + C_1 \quad [5.26]$$

In the case of  $x = 0$ ,  $\phi(0) = 0$ , which leads to  $C_1 = 0$ . The end point angle, due to the torque  $T$  is then identified by using equation 5.26 with  $x = L$ .

$$\phi_T(L) = \frac{TL}{EI} \quad [5.27]$$

To calculate the deflection  $y(x)$  equation 5.26 is further integrated

$$y(x) = \frac{T}{2EI} x^2 + C_2 \quad [5.28]$$

In the case of  $x = 0$ ,  $y(0) = 0$ , which leads to  $C_2 = 0$ . The end point deflection, due to the torque  $T$  is then identified by using equation 5.28 with  $x = L$ .

$$y_T(L) = \frac{TL^2}{2EI} \quad [5.29]$$

The total angle error is found by summing equations 5.15, 5.20 and 5.27.

$$\phi(L) = \phi_F(L) + \phi_q(L) + \phi_T(L) = \frac{FL^2}{2EI} + \frac{qL^3}{6EI} + \frac{TL}{EI} = \frac{L}{EI} \left( \frac{FL}{2} + \frac{qL^2}{6} + T \right) \quad [5.30]$$

The total deflection is found by summing equations 5.17, 5.23 and 5.29.

$$y(L) = y_F(L) + y_q(L) + y_T(L) = \frac{FL^3}{3EI} + \frac{qL^4}{8EI} + \frac{TL^2}{2EI} = \frac{L^2}{EI} \left( \frac{FL}{3} + \frac{qL^2}{8} + \frac{T}{2} \right) \quad [5.31]$$

Equation 5.30 and 5.31 is applied to our beam structure as shown in Figure 5.14. The analysed part of the beam are selected to lay within the area between the two flanges at each end, where we have a uniform cross-section of the beam. The flanges, at each end of the beam, has much larger dimensions and they are assumed to contribute far less to the compliance calculations.

The point force F is removed from these calculations since its not existing in this particular set-up because there is no motor and gear system at exactly this point. These components is located a bit further away from  $K_C$ , and their influence to the beam is incorporated in the calculation of the torque T.

The torque T is calculated by summing up the influence of the weight of the joint 3 (including all components)  $T_{J_3}$ , the weight of arm 4  $T_{A_4}$  and the selected workload  $T_{WL}$ .

$$T_{J_3} \approx 47.7 \quad [Nm] \quad [5.32]$$

$$T_{A_4} \approx 50.4 \quad [Nm] \quad [5.33]$$

$$T_{WL} \approx 39.4 \quad [Nm] \quad [5.34]$$

Summing up the total torque T :

$$T = T_{J_3} + T_{A_4} + T_{WL} = 137.5 \quad [Nm] \quad [5.35]$$

The beam weight distribution q is calculated for the given square aluminium beam. Beam cross-section dimensions are: 145x145x10 [mm] (outer width x outer height x thickness)

$$q \approx 143 \quad [N / m] \quad [5.36]$$

According to the equation 5.29 the torque T creates at deflection at the end of the beam (movement of  $K_B$  in the  $Z_B$  direction)

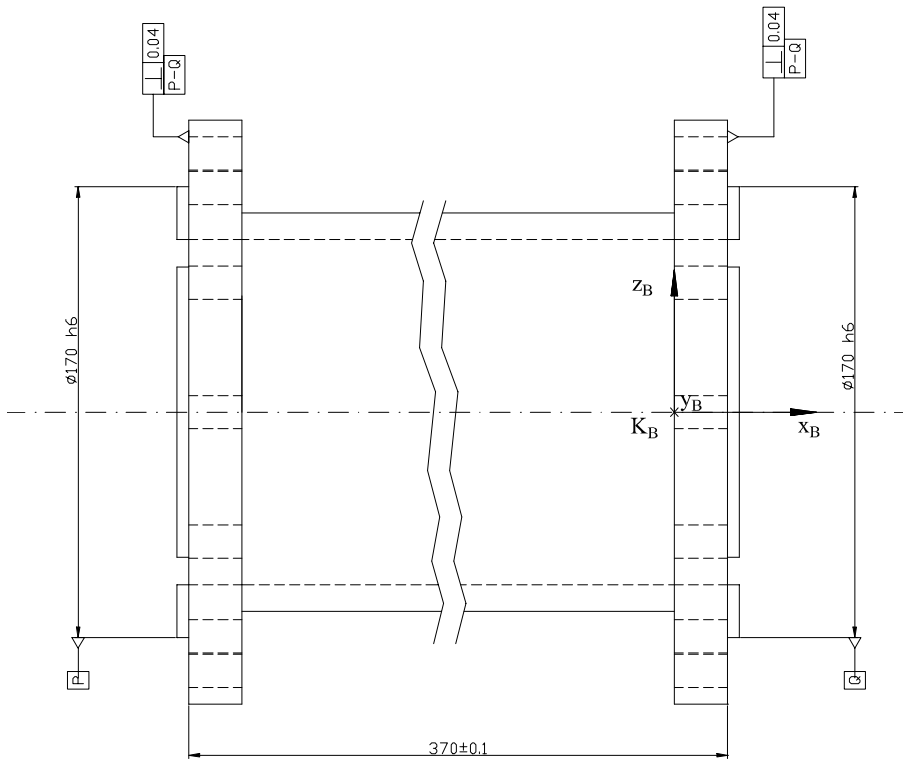
$$y_T(L) = \frac{TL^2}{2EI} \approx 0.008 \quad [mm] \quad [5.37]$$

According to the equation 5.23 the distributed weight creates a deflection at the end of the beam (movement of  $K_B$  in the  $Z_B$  direction)

$$y_q(L) = \frac{qL^4}{8EI} \approx 0.0003 \quad [\text{mm}] \quad [5.38]$$

Summing together, the total deflection is found

$$z_{E2} = -(y_q(L) + y_T(L)) = -\left(\frac{qL^4}{8EI} + \frac{TL^2}{2EI}\right) = -\left(\frac{L^2}{EI} \left(\frac{qL^2}{8} + \frac{T}{2}\right)\right) = -0.0083 \quad [\text{mm}] \quad [5.39]$$



**Figure 5.14** Side view of beam. Compliance analysis

Calculation of the angle error arising from the torque and the distributed load is done according to equation 5.27 and 5.20, respectively.

$$\phi_T(L) = \frac{TL}{EI} \approx 4.408 \cdot 10^{-5} \quad [\text{deg}] \quad [5.40]$$

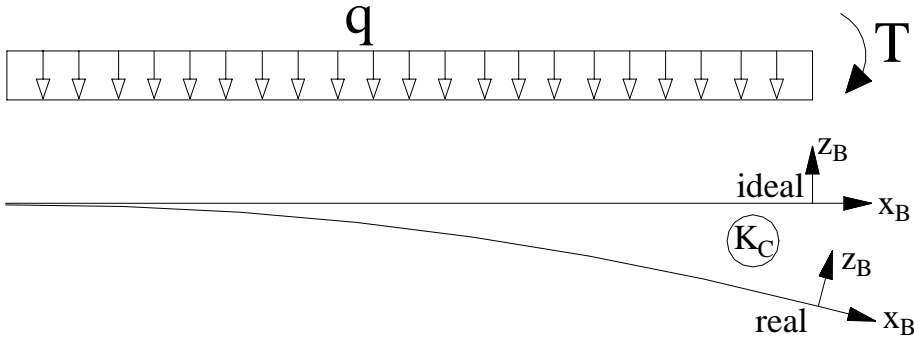
$$\phi_q(L) = \frac{qL^3}{6EI} \approx 1.046 \cdot 10^{-6} \quad [\text{deg}] \quad [5.41]$$

Summing together, the total angle error is found

$$\phi_{E2} = \phi_q(L) + \phi_T(L) = \frac{qL^3}{6EI} + \frac{TL}{EI} = \frac{L}{EI} \left( \frac{qL^2}{6} + T \right) \approx 4.513 \cdot 10^{-5} \quad [\text{deg}] \quad [5.42]$$

This angle error is representing a rotation of the coordinate system  $K_B$  around the  $Y_B$  axis which appears after the translation  $Z_{E2}$  is carried out.

Thus, the total transformation of the coordinate system  $K_B$  due to the beam compliance is given by a two step procedure. First a transformation  $Z_{E2}$  along the ideal  $Z_B$  direction, followed by a rotation  $\phi_{E2}$  around the new  $Y_B$  axis which is formed after the transformation  $Z_{E2}$  carried out . Figure 5.15 shows the coordinate systems  $K_B$  before (ideal) and after (real) transformation.



**Figure 5.15** Coordinate systems for beam element

The transformation matrix between the ideal and real system, or say the influence of the beam compliance to the kinematic chain, is given by:

$$E2 = \begin{bmatrix} c\phi_{E2} & 0 & -s\phi_{E2} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E2} & 0 & c\phi_{E2} & 0 \\ 0 & 0 & z_{E2} & 1 \end{bmatrix} \quad [5.43]$$

**Comments on the simplified beam analysis**

Equations 5.30 and 5.31 are made on the basis of a horizontal beam with a given work load (F, q and T). The beam cross-section is assumed to be designed such that the moment of inertia I is constant. Material elasticity (the E-modulus) is also assumed to be constant. This is not the case in a real design.

In the real design, we would expect to find complex beam structures where the above parameters would vary with the length of the beam structure. To analyze these structures we need complex CAD modeling and computer calculations.

However, the usage of equation 5.30 and 5.31 with a worst-case load scenario enables us to be sure that our calculations are conservative and reach maximum values. Thus, we get a quite clear picture of the maximum error. Further calculations into end-effector error will give the design team a possibility to see how well this matches our precision requirement. In the preliminary and the detailed design phase, this methodology will give important knowledge to the team of designers.

No more error sources than the two above mentioned errors are analyzed for the beam element. Of course, in a real design, we may identify several other beam errors than the above analyzed, but in this case study we seek to gain an understanding of the principle of error handling only.

In the next paragraph we move forward to the joint 3 housing element, working with other selected error sources.

### **5.3.2 Error identification of joint 3 housing element**

For the joint housing element we further select two sources of errors in our example calculation. Figure 5.16 shows the joint housing with some selected dimensional and geometrical tolerances applied. One of the selected error sources, denoted **E4**, is shown at the lower end of Figure 5.16.

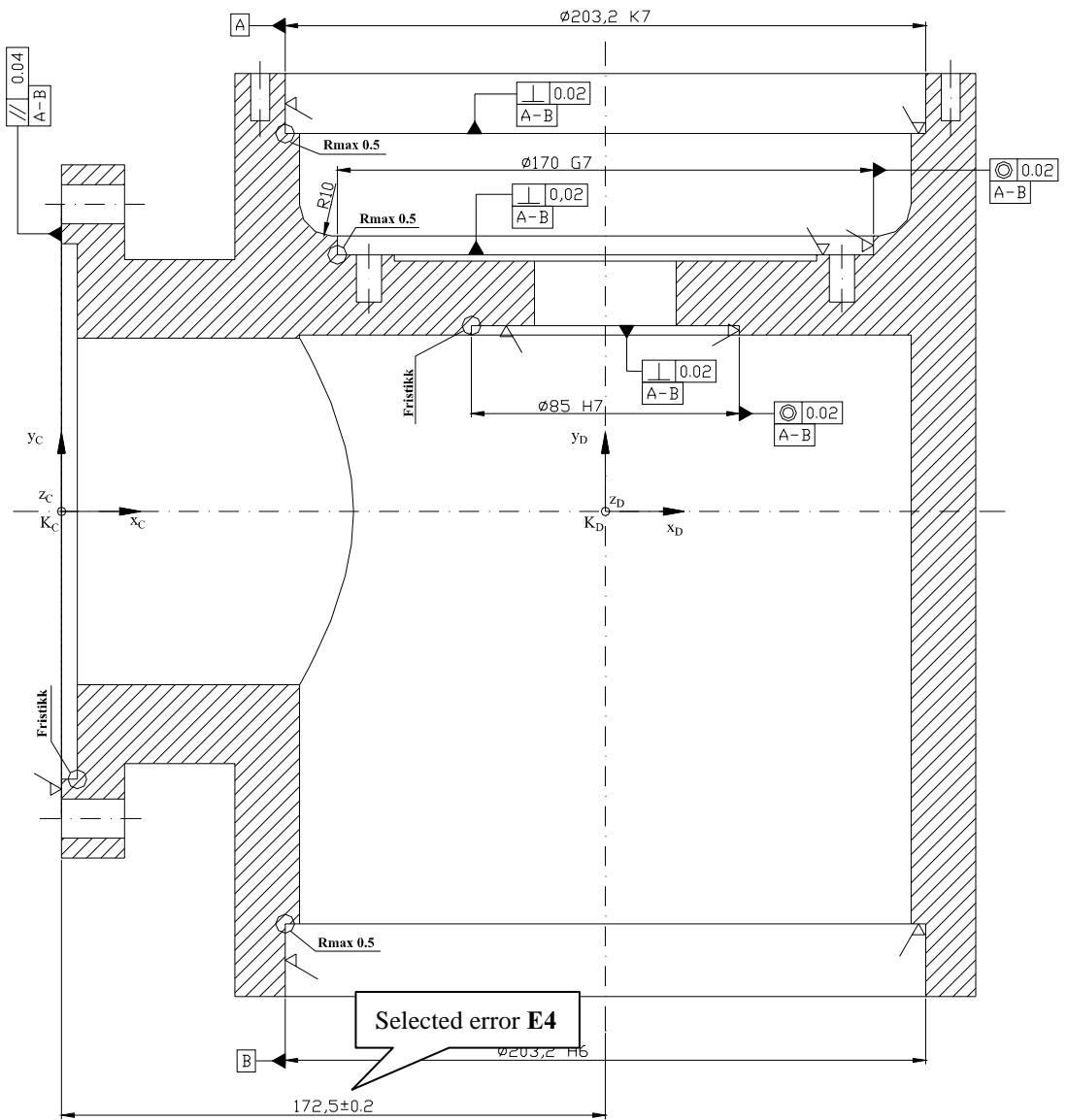
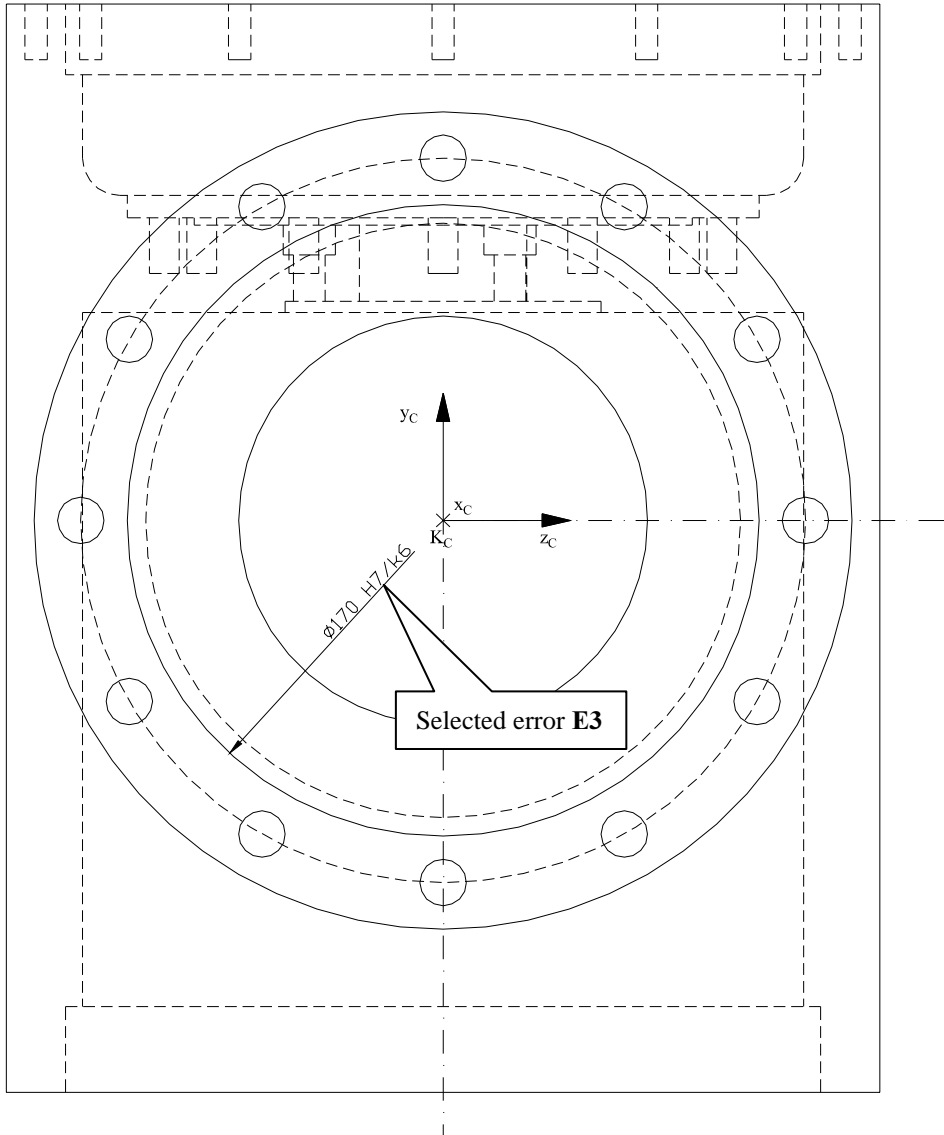


Figure 5.16 Joint 3 housing, top view

Figure 5.17 shows another view of the joint. In this view the second selected error source is visible, denoted **E3**.





**Figure 5.17** Joint 3 housing, left view

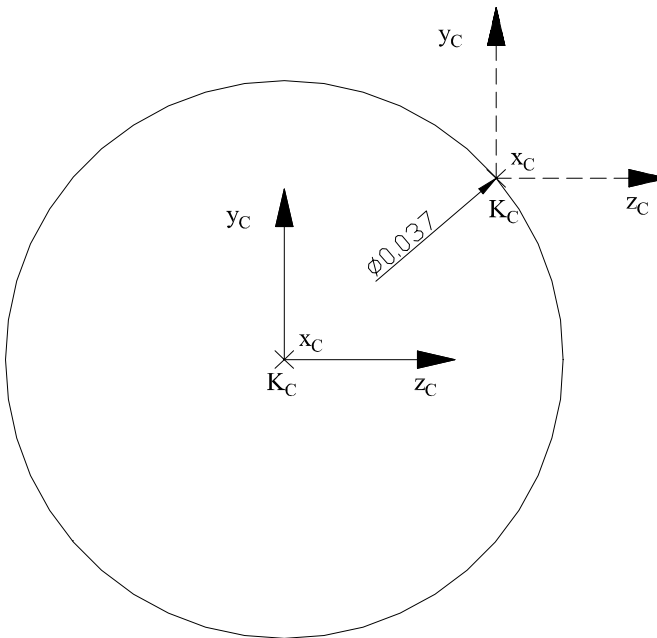
### **The E3 error**

The E3 error arises from the mating tolerance H7/k6 in the area of assembly of the beam and the joint housing component. The tolerance allows a translational movement of the  $K_C$  coordinate system. The H7 symbol is equivalent to an allowable manufacturing zone of  ${}^{+40}_0 [\mu\text{m}]$  while the mating axis symbol k6 is  ${}^{+28}_{+3} [\mu\text{m}]$ . When the hole, defined by H7, is manufactured at the maximum tolerance the dimension becomes  $\phi 170.04 [\text{mm}]$ . A mating beam with a corresponding minimum manufacturing

tolerance will have the minimum dimension of  $\phi 170.003 [mm]$ . This gives a physical maximum possible free movement error of

$$p = \phi 170.04 - \phi 170.003 = \phi 0.037 [mm] \quad [5.44]$$

Applying this to the kinematic chain, the  $K_D$  system is allowed to translate (move) around the circle with a defined diameter of  $\phi 0.037 [mm]$ , as depicted in Figure 5.18.



**Figure 5.18** Translational movement of coordinate system  $K_D$

The tolerance  $\phi 0.037 [mm]$  will contribute with a possible maximum displacement of  $K_C$  in the  $Z_C$  direction of

$$z_{E3} = \pm \frac{0.037}{2} \approx \pm 0.019 [mm] \quad [5.45]$$

If all possible displacement is focused instead into the  $Y_C$  direction

$$y_{E3} = \pm \frac{0.037}{2} \approx \pm 0.019 [mm] \quad [5.46]$$

Any combination of displacement into the  $Z_C$  and  $Y_C$  direction is possible as long as the resulting point will be at the circumference. Equation 5.47 allows for translations in both directions.

$$(z_{E3})^2 + (y_{E3})^2 = (0.019)^2 \quad [5.47]$$

The transformation matrixes arising from the above equations are as follows:

For equation 5.45:

$$E3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & z_{E3} & 1 \end{bmatrix} \quad [5.48]$$

For equation 5.46:

$$E3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & y_{E3} & 0 & 1 \end{bmatrix} \quad [5.49]$$

For equation 5.47:

$$E3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & y_{E3} & z_{E3} & 1 \end{bmatrix}, \text{ while } (z_{E3})^2 + (y_{E3})^2 = (0.019)^2 \quad [5.50]$$

### **The E4 error**

The dimensional tolerance of the length  $172.5 \pm 0.2$  is the next error source to be analysed.

According to the Figure 5.16 above the tolerance  $\pm 0.2$  will contribute with a possible displacement of the  $K_D$  system in the  $X_D$  direction.

$$x_{E4} = \pm 0.2 \quad [5.51]$$

This transformation matrix of the  $K_D$  system, originated from this displacement tolerance is:

$$E4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_{E4} & 0 & 0 & 1 \end{bmatrix} \quad [5.52]$$

### 5.3.3 Error identification of joint 3 components

In the Figure 5.6 the following joint 3 components can be viewed:

- One motor (actuator),
- One gear or transmission system,
- Two bearings,
- One joint measurement system

In this part we will select the transmission system as our elements for the error analysis.

Mechanical transmissions are introduced into manipulators for a number of reasons:

- To reduce joint speed,
- To enhance joint torque.

#### **As a speed reducer**

For example, using a direct driven motor in a joint close to robot-base one has to consider the minimum uniform speed of the motor. For an axial plunger hydraulic motor of a given size the minimum uniform speed is approximately 5 rpm. Operating below the minimum 5 rpm makes the motor move in a stick-slip manner and thus decrease control system resolution.

If such a motor is connected to an arm with a length of 1.8 meter, the tool-centre-point will move with a minimum linear speed of 942 mm/s, based on 5 rpm minimum uniform speed. By using a harmonic drive as a speed reducer and given a gearing ratio (1:120) of the drive the linear speed will be approx. 7.85 mm/s (Solvang, Lien and Thomessen, 1999).

This is obviously not good enough for most applications. By introducing a gearing system, as a speed reducer, a uniform end-effector speed and resolution is achieved.

#### **As a joint torque enhancer**

Serial structure manipulator inner joints must comply with significant torques. To deal with such a load, a gearing unit will give an increase in capacity proportional to the

gearing ratio. Torques from inertial mass forces is even reduced by the square of the gearing ratio. Motor physical sizes and thereby their weight is dependent on their rated torque. So, gearing units allow for the usage of smaller motors which are lighter and thus decrease the load onto the construction.

In the joint 3 a harmonic drive unit is used as a transmission system. A harmonic drive gear has the following capabilities (see Table 5.2) according to Rivin (1988).

**Table 5.2** Harmonic drive capabilities

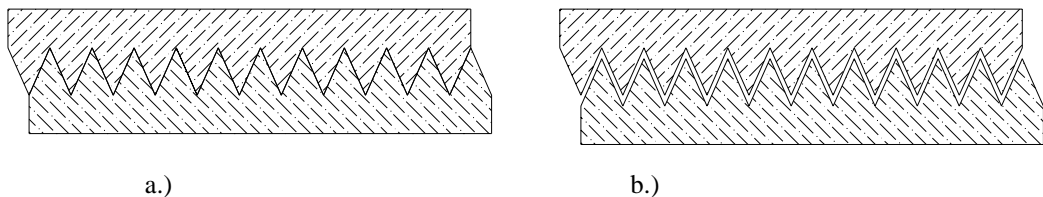
Transmission system	Special usage	Advantage	Disadvantage
Harmonic drive	Widespread usage	Large velocity ratio and high torque capacity, in a very compact space. Lightweight. Small backlash because of the preloaded mesh.	Not perfectly smooth operation due to deformations (compliance). Variable stiffness with load. Limited back drivability. Limited input speed due to fatigue endurance problem with flexispline.

From this table we further identify two sets of error sources which must be further studied, namely the **backlash** (however small) and the **compliance** due to the lack of gear stiffness.

### The E5 error

#### **Backlash:**

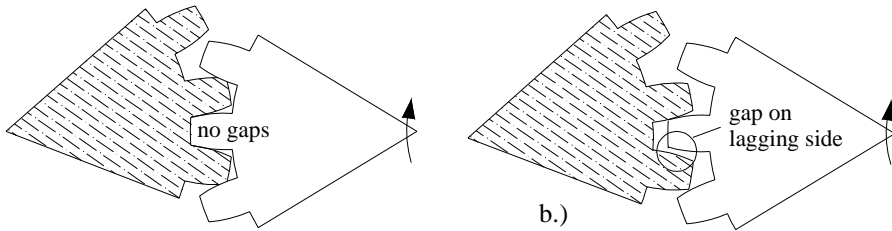
Backlash can be defined as the maximum angular motion of a gear while its mating gear is fixed, but no or low torque is applied.



**Figure 5.19** a.) Proper mesh. b.) Improper mesh, backlash.

In conventional power transmissions backlash has to be provided to accommodate the thermal expansion of teeth as well as machining and assembly tolerances. Accordingly, in conventional gearing systems reduction or elimination of backlash has to be done with either enhancing the accuracy of the mating parts or introduce some compensation system (Rivin, 1988). Figure 5.19 and 5.20 illustrate typical backlash occurrence in gearing systems.

As stated above backlash error arises from three different sources: manufacturing tolerances, assembly tolerances and thermal effects. To exactly calculate the numerical value of the backlash, or say physical error, it is necessary to look to the manufacturing unit, the unit who does the final assembly and finally study the impact from the working conditions. However, most manufactures provide a numerical value for the maximum backlash (maximum physical error), provided that the assembly tolerance and the operating conditions are within specified values. The sign of this numerical value is unknown, and any value less than maximum could not be predicted. Thus, backlash is a random error type, which is dependent on the manipulator configuration.



**Figure 5.20** a.) Proper mesh, zero gaps b.) Improper mesh, gap on lagging side.

As seen on Figure 5.20 b.), the “play” (i.e., backlash) is found on the lagging side of the gear, while when the joint is driving towards the opposite direction maximum backlash error is found on what is now the lagging side. While the gear rotation is changing direction or the gear is at rest, the numerical value and its sign is somewhere between the max/min value.

From the manufacturer of the actual harmonic drive unit used in joint 3, we find that there is no backlash in the region of the tooth engagement, but instead there exist a backlash in the coupling (so called oldham coupling) between the HD (Harmonic Drive) gear and the motor shaft.

The numerical value of this error source is: (Harmonic Drive Germany, manufacturer)

$$\phi_{E5} = \pm 6 [\text{arcmin}] \approx \pm 1.67 \cdot 10^{-3} [\text{deg}] \quad [5.53]$$

The transformation matrix between the ideal and real system, or say the influence of the beam compliance to the kinematic chain, is given by:

$$E5 = \begin{bmatrix} c\phi_{E5} & 0 & -s\phi_{E5} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E2} & 0 & c\phi_{E5} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [5.54]$$

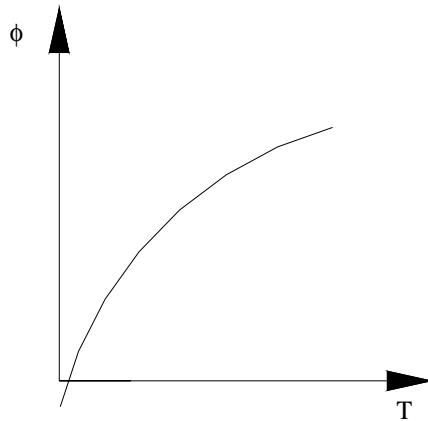
The error arising from the backlash is assumed to interfere with the kinematic chain and in particular coordinate system  $K_D$  in a rotational manner, meaning that it creates a rotational error around the  $y_D$  axis. See Figure 5.6 and 5.16.

### **The E6 error**

#### **Compliance:**

The backlash error does not include gear compliance, which also is an (angular) motion component. Compliance arises from deformation due to external forces.

Compliance characteristics for a gear system are normally provided by the manufactures. Here we often find a functional description between the deflection  $\phi$  and the applied torque  $T$ . Figure 5.21 shows this principle.



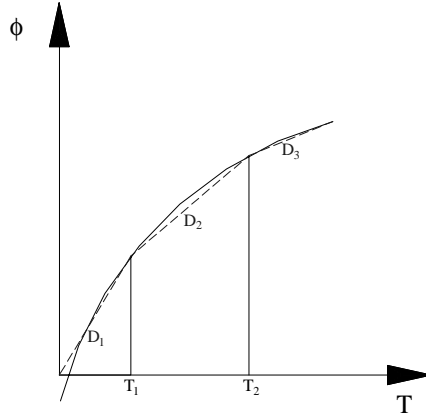
**Figure 5.21** Deflection torque model

The curvature of the deflection is found by measurements of the gearing system. The curvature is either made on the basis of one gear or, more likely, the curvature is found by measurements of a whole range of gears within the same production. In the latter case the worst case scenario should be picked out to represent the hole range of gearing system to ensure a conservative formula for the further calculation of the total system deflection.

The functional relationship:

$$\phi = f(T) \quad [5.55]$$

is found based on empirical data. Figure 5.22 shows the functional description of the deflection of the actual HD unit. The manufacturer applies a “linear zone” approach. In their model, the curvature is linearized into three zones, separated by  $T_1$  and  $T_2$ .  $D_1$ ,  $D_2$  and  $D_3$  are the corresponding linear constants (spring stiffness).



**Figure 5.22** Simplified deflection torque model, harmonic drive gears  
(Courtesy of Harmonic Drive GmbH)

The calculation of the deflection is done by

$$\phi_{E6} = \frac{T}{D_1} \text{ when } T \leq T_1 \quad [5.56]$$

$$\phi_{E6} = \frac{T}{D_1} + \frac{T - T_1}{D_2} \text{ when } T_1 < T < T_2 \quad [5.57]$$

$$\phi_{E6} = \frac{T}{D_1} + \frac{T_2 - T_1}{D_2} + \frac{T - T_2}{D_3} \text{ when } T \geq T_2 \quad [5.58]$$

To calculate the numerical value of the physical error, in the actual joint design, the following constants is gained from the manufacturer:

$$T_1 = 108 [Nm] \quad [5.59]$$

$$T_2 = 382 [Nm] \quad [5.60]$$

$$D_1 = 2 \cdot 10^5 [Nm / rad] \quad [5.61]$$

$$D_2 = 2.7 \cdot 10^5 [Nm / rad] \quad [5.62]$$

$$D_3 = 3.4 \cdot 10^5 [Nm / rad] \quad [5.63]$$

The actual torque  $T$  is to be found from the arm static and or dynamic load analysis. For the simplicity of this case study a static load calculation is used by summing up the influence of the weight of arm 4  $T_{A4}$  and the selected workload  $T_{WL}$ .

$$T_{A4} \approx 28.8 [Nm] \quad [5.64]$$

$$T_{WL} \approx 22.5 [Nm] \quad [5.65]$$



Summing up the total torque  $T$  :

$$T = T_{A4} + T_{WL} = 51.3 \quad [Nm] \quad [5.66]$$

Since  $T \leq T_1$  the equation 5.56 is applied and the deflection is calculated

$$\phi_{E6} = \frac{51.3}{2 \cdot 10^5} \cdot \frac{180}{\pi} \approx 1.47 \cdot 10^{-2} \quad [\text{deg}] \quad [5.67]$$

The transformation matrix between the ideal and real system, or say the influence of the transmission compliance to the kinematic chain, is given by:

$$E6 = \begin{bmatrix} c\phi_{E6} & 0 & -s\phi_{E6} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E6} & 0 & c\phi_{E6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [5.68]$$

The error arising from the transmission compliance is assumed to interfere with the kinematic chain and in particular coordinate system  $K_D$  in a rotational manner, meaning that it creates a rotational error around the  $y_D$  axis. See Figure 5.6 and 5.16.

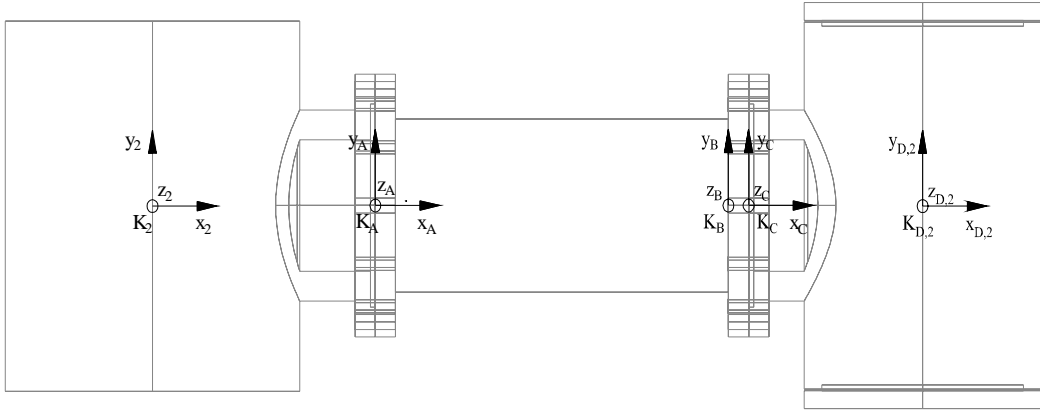
## 5.4 Identification of the real kinematic chain, and the real end-effector pose

### 5.4.1 Real forward kinematic chain

In the previous paragraph some selected error components were chosen and their influence to the kinematic chain were identified by setting up the error matrixes **E1..E6**. All of the error matrixes were identified inside what is denoted as the  $K_2 \rightarrow K_3$  system.

A set of new coordinate systems  $K_A, K_B, K_C, K_D$  were distributed inside the  $K_2 \rightarrow K_3$  system at the positions where the different errors origin. See Figure 5.23

- $K_A$  is situated at the area of the error **E1**
- $K_B$  is situated at the area of the error **E2**
- $K_C$  is situated at the area of the error **E3**
- $K_D$  is situated at the area of the error **E4, E5, E6**



**Figure 5.23** Coordinate assignment

The distances (x direction) between the different coordinate systems are:

$$l_{A2} = 221 [mm] \quad [5.69]$$

$$l_{BA} = 350 [mm] \quad [5.70]$$

$$l_{CB} = 20 [mm] \quad [5.71]$$

$$l_{DC} = 172.5 [mm] \quad [5.72]$$

The transformation between  $K_2 \rightarrow K_A$  is decided in a two step procedure, first a transformation  $l_{A2}$  along the  $X_2$  axis followed by the rotation defined as the **E1** error. The **E1** error was dual, thus represented by two error matrixes in equation 5.9 and 5.10. The first rotation was defined as a rotation around  $Z_A$  axis while the other possible solution was a rotation around the  $Y_A$  axis. The maximum rotational error was found by introducing the calculated  $\phi_{E1}$  into one of the matrixes, defined by 5.9 and 5.10:

$$E1 = \begin{bmatrix} c\phi_{E1} & s\phi_{E1} & 0 & 0 \\ -s\phi_{E1} & c\phi_{E1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$E1 = \begin{bmatrix} c\phi_{E1} & 0 & -s\phi_{E1} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E1} & 0 & c\phi_{E1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, to calculate the effect from the E1 error in both directions we need to be able to use both the matrixes inside the kinematic chain, without having to change the kinematic chain in between the calculations. This is possible to achieve by using the universal transformation matrix, as defined in Chapter 4.

$$T = \begin{bmatrix} cB \cdot cC & cB \cdot sC & -sB & 0 \\ -sC \cdot cA & cC \cdot cA & sA \cdot cB & 0 \\ +sA \cdot sB \cdot cC & +sA \cdot sB \cdot sC & & \\ sA \cdot sC & -sA \cdot cC & cA \cdot cB & 0 \\ +cA \cdot sB \cdot cC & +cA \cdot sB \cdot sC & & \\ X & Y & Z & 1 \end{bmatrix}$$

Here we simply loose the effect from the  $A$  rotation by letting  $A=0$ , the  $B$  rotation will be denoted  $\phi_{E1B}$ , while the other possible  $C$  rotation will be denoted  $\phi_{E1C}$ . While the matrix is used in numerical calculations the  $\phi_{E1C}$  or the  $\phi_{E1B}$  will be equal to zero, and the transformation matrix will be reduced to the dual **E1** matrix. In addition by introducing the length parameter  $l_{A2}$  as the  $X$  in the general transformation matrix, we finally achieve the transformation matrix between coordinate system  $K_2 \rightarrow K_A$

$$T_{A2} = \begin{bmatrix} c\phi_{E1B} \cdot c\phi_{E1C} & c\phi_{E1B} \cdot s\phi_{E1C} & -s\phi_{E1B} & 0 \\ -s\phi_{E1C} & c\phi_{E1C} & 0 & 0 \\ s\phi_{E1B} \cdot c\phi_{E1C} & s\phi_{E1B} \cdot s\phi_{E1C} & c\phi_{E1B} & 0 \\ l_{A2} & 0 & 0 & 1 \end{bmatrix} \quad [5.73]$$

Naturally this is the same results as if we had carried out the transformation, step by step.

$$T_{A2} = \begin{bmatrix} c\phi_{E1B} & 0 & -s\phi_{E1B} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E1B} & 0 & c\phi_{E1B} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\phi_{E1C} & s\phi_{E1C} & 0 & 0 \\ -s\phi_{E1C} & c\phi_{E1C} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l_{A2} & 0 & 0 & 1 \end{bmatrix}$$

The transformation between  $K_A \rightarrow K_B$  is decided in a three step procedure, first a transformation  $l_{BA}$ , followed by a new transformation  $Z_{E2}$ , and finalised by a rotation  $\phi_{E2}$ . The two latter transformations are included in the **E2** matrix, defined by equation 5.43. The total transformation is found by the multiplication:

$$T_{BA} = \begin{bmatrix} c\phi_{E2} & 0 & -s\phi_{E2} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E2} & 0 & c\phi_{E2} & 0 \\ 0 & 0 & z_{E2} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l_{BA} & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\phi_{E2} & 0 & -s\phi_{E2} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E2} & 0 & c\phi_{E2} & 0 \\ l_{BA} & 0 & z_{E2} & 1 \end{bmatrix} \quad [5.74]$$

The same result is also achieved by using the universal matrix by letting  $C = 0, A = 0, Y = 0$  while  $B = \phi_{E2}, X = l_{BA}, Z = z_{E2}$ .

The transformation between  $K_B \rightarrow K_C$  is decided in a two step procedure, first a transformation  $l_{CB}$ , followed by a new transformation. This new transformation is defined by equations 5.48, 5.49 and 5.50. The matrix in 5.50 includes both information from 5.49 and 5.48 and thereby represents all possible error configurations. The total transformation is found by the multiplication:

$$T_{CB} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & y_{E3} & z_{E3} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l_{CB} & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l_{CB} & y_{E3} & z_{E3} & 1 \end{bmatrix} \quad [5.75]$$

The transformation between  $K_C \rightarrow K_D$  is decided in a four step procedure, first a transformation  $l_{DC}$ , followed by a new transformation defined as the **E4** error [5.52]. The next step is a rotation defined as the **E5** error [5.54] and finally a new rotation defined as the **E6** error [5.68]. The total transformation is found by carrying out the multiplication:

$$T_{DC} = \begin{bmatrix} c\phi_{E6} & 0 & -s\phi_{E6} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E6} & 0 & c\phi_{E6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\phi_{E5} & 0 & -s\phi_{E5} & 0 \\ 0 & 1 & 0 & 0 \\ s\phi_{E5} & 0 & c\phi_{E5} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_{E4} & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l_{DC} & 0 & 0 & 1 \end{bmatrix}$$

The calculation of the  $T_D$  matrix is simplified by summing both angle errors  $\phi_{E5}$  and  $\phi_{E6}$ .

$$T_{DC} = \begin{bmatrix} c(\phi_{E5} + \phi_{E6}) & 0 & -s(\phi_{E5} + \phi_{E6}) & 0 \\ 0 & 1 & 0 & 0 \\ s(\phi_{E5} + \phi_{E6}) & 0 & c(\phi_{E5} + \phi_{E6}) & 0 \\ x_{E4} + l_{DC} & 0 & 0 & 1 \end{bmatrix} \quad [5.76]$$

The final transformation to describe the complete movement from  $K_2 \rightarrow K_3$  is to include the joint angle movement  $V_3$  itself. Thus, the final transformation from  $K_D \rightarrow K_3$  is:

$$T_{3D} = \begin{bmatrix} cV_3 & 0 & -sV_3 & 0 \\ 0 & 1 & 0 & 0 \\ sV_3 & 0 & cV_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [5.77]$$

The total real transformation within the  $K_2 \rightarrow K_3$  system is now given by:

$$T_3 = T_{3D} \cdot T_{DC} \cdot T_{CB} \cdot T_{BA} \cdot T_{A2} \quad [5.78]$$

While the total new real structure transformation matrix  $K_0 \rightarrow K_4$  is given by:

$$T^{real} = T_4 \cdot T_3 \cdot T_2 \cdot T_1 = T_4 \cdot T_{3D} \cdot T_{DC} \cdot T_{CB} \cdot T_{BA} \cdot T_{A2} \cdot T_2 \cdot T_1 \quad [5.79]$$

Where  $T_4, T_2$  and  $T_1$  is the ideal transformations defined in equation 5.2.

In a real design with many error sources included in the analysis calculation of the equation for transformation  $T^{real}$  needs computing capacity. However, simplifications could be made by summing subsequent equivalent angles in the chain. A special case of the equation 5.79 where we assume that the  $T_{A2}$  contributes with a B rotation only in equation 5.73 is such a situation and the  $T^{real}$  will become:

$$T^{real} = \begin{bmatrix} c\beta \cdot cV_1 & c\beta \cdot sV_1 & -s\beta & 0 \\ -sV_1 & cV_1 & 0 & 0 \\ s\beta \cdot cV_1 & s\beta \cdot sV_1 & c\beta & 0 \\ X & Y & Z & 1 \end{bmatrix} \quad [5.80]$$

Where  $\beta = V_3 + \phi_{E6} + \phi_{E5} + \phi_{E2} + \phi_{E1} + V_2$

However, the expressions for the position, given by  $X$ ,  $Y$  and  $Z$  are more extensive:

$$\begin{aligned} X &= (l_4 \cdot c\beta + (X_{E4} + l_{CB} + l_{DC}) \cdot c(V_2 + \phi_{E1} + \phi_{E2}) + Z_{E3} \cdot s(V_2 + \phi_{E1} + \phi_{E2})) \\ &+ l_{2x} + l_{BA} \cdot c(V_2 + \phi_{E1}) + Z_{E2} \cdot s(V_2 + \phi_{E1}) + l_{A2} \cdot cV_2 \cdot cV_1 - Y_{E3} \cdot sV_1 \\ Y &= (l_4 \cdot s\beta + (X_{E4} + l_{CB} + l_{DC}) \cdot c(V_2 + \phi_{E1} + \phi_{E2}) + Z_{E3} \cdot s(V_2 + \phi_{E1} + \phi_{E2})) \\ &+ l_{2x} + l_{BA} \cdot c(V_2 + \phi_{E1}) + Z_{E2} \cdot s(V_2 + \phi_{E1}) + l_{A2} \cdot cV_2 \cdot sV_1 - Y_{E3} \cdot cV_1 \\ Z &= l_4 \cdot s\beta + Z_{E2} \cdot c(V_2 + \phi_{E1}) - (l_{DC} + X_{E4} + l_{CB}) \cdot s(V_2 + \phi_{E1} + \phi_{E2}) - l_{A2} \cdot sV_2 \\ &+ Z_{E3} \cdot c(V_2 + \phi_{E1} + \phi_{E2}) - l_{BA} \cdot s(V_2 + \phi_{E1}) + l_{2z} \end{aligned} \quad [5.81]$$

To solve larger systems and even in this particular case when  $T_{A2}$  is freely allowed to contribute in one or the other direction a analytical software is used for calculation of the  $T^{real}$ . Maple® has been used for these analytical calculations.

### 5.4.2 End effector real pose

End-effector real position and orientation,  $(EEp_{bcs})^{real}$ , notified as the real forward kinematic model is derived from the structure real transformation matrix in equation 5.79. As described in Chapter 4 the position/orientation is found:

**Position:**

$$\begin{aligned} X_{bcs}^{real} &= T_{4,1}^{real} \\ Y_{bcs}^{real} &= T_{4,2}^{real} \\ Z_{bcs}^{real} &= T_{4,3}^{real} \end{aligned} \quad [5.82]$$

**Orientation:**

$$\begin{aligned} A_{bcs}^{real} &= \arctg\left(\frac{T_{2,3}^{real}}{T_{3,3}^{real}}\right) \\ B_{bcs}^{real} &= -\arcsin(T_{2,3}^{real}) \\ C_{bcs}^{real} &= \arctg\left(\frac{T_{1,2}^{real}}{T_{1,1}^{real}}\right) \end{aligned} \quad [5.83]$$

Where the  $T_{n,n}^{real}$  is collected as the matrix elements within 5.79.

By grouping both position and orientation data inside a vector we find an expression for the end-effector real pose:

$$EEp_{bcs}^{real} = \left[ X_{bcs}^{real} \quad Y_{bcs}^{real} \quad Z_{bcs}^{real} \quad A_{bcs}^{real} \quad B_{bcs}^{real} \quad C_{bcs}^{real} \right] \quad [5.84]$$

## 5.5 Calculation of the end effector error

The end- effector error  $dEEp_{bcs}$  is found by subtracting the ideal pose from the real pose. The equation from the ideal pose was deducted earlier in paragraph 5.2.2 as equation 5.5.

$$dEEp_{bcs} = EEp_{bcs}^{real} - EEp_{bcs}^{ideal} \quad [5.85]$$

The following parameters are held constant in the forthcoming calculations:

**Constants:**

Manipulator in stretched out (staging) position

$$V_1 = V_2 = V_3 = 0.$$

Ideal kinematic parameters

$$l_{2x} = 300 [mm], l_{2z} = 200 [mm], l_3 = 763.5 [mm], l_4 = 553 [mm].$$

Parameters related to the location of the error sources **E1...E6**

$$l_{A2} = 221 [mm], l_{BA} = 350 [mm], l_{CB} = 20 [mm], l_{DC} = 172.5 [mm].$$

**5.5.1 Calculation of single error influence**

In this paragraph the effect of the single error sources **E1...E6** onto the end effector pose  $dEEp_{bcs}$  is calculated by the usage of equation 5.85.

**E1 analysis:**

a.) Assuming C rotation and  $\phi_{E1} = \pm 0.0104$  [deg], all other error components are ZERO.

$$dEEp_{bcs} = [-0.000018 \quad \pm 0.198849 \quad 0 \quad 0 \quad 0 \quad \pm 0.0104]$$

b.) Assuming B rotation and  $\phi_{E1} = \pm 0.0104$  [deg], all other error components are ZERO.

$$dEEp_{bcs} = [-0.000018 \quad 0 \quad \pm 0.198849 \quad 0 \quad \pm 0.0104 \quad 0]$$

**E2 analysis:**

Assuming  $\phi_{E2} = 4.513 \cdot 10^{-5}$  [deg],  $Z_{E2} = -0.0083$  [mm], all other error components are ZERO.

$$dEEp_{bcs} = [0 \quad 0 \quad -0.008887 \quad 0 \quad 0.00004513 \quad 0]$$

**E3 analysis:**

a.) Assuming  $Y_{E3} = \pm 0.019$  [mm], all other error components are ZERO.

$$dEEp_{bcs} = [0 \quad \pm 0.019 \quad 0 \quad 0 \quad 0 \quad 0]$$

b.) Assuming  $Z_{E3} = \pm 0.019$  [mm], all other error components are ZERO.

$$dEEp_{bcs} = [0 \quad 0 \quad \pm 0.019 \quad 0 \quad 0 \quad 0]$$

**E4 analysis:**

Assuming  $X_{E4} = \pm 0.2 [mm]$ , all other error components are ZERO.

$$dEEp_{bcs} = [\pm 0.2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

**E5 analysis:**

Assuming  $\phi_{E5} = \pm 1.67 \cdot 10^{-3} [deg]$ , all other error components are ZERO.

$$dEEp_{bcs} = [-0.0000002 \quad 0 \quad \pm 0.0161183 \quad 0 \quad \pm 0.00167 \quad 0]$$

**E6 analysis:**

Assuming  $\phi_{E6} = 1.47 \cdot 10^{-2} [deg]$ , all other error components are ZERO.

$$dEEp_{bcs} = [-0.0000182 \quad 0 \quad -0.141879 \quad 0 \quad 0.0147 \quad 0]$$

**5.5.2 Calculation of combined error influence**

In this paragraph the effect of all error sources **E1...E6** onto the end effector pose  $dEEp_{bcs}$  are calculated. Equation 5.85 is used for the calculation.

The selected parameters represent a **worst case** situation where every source is contributing in the same direction

**Combined E1...E6 analysis:**

The following error components are selected in this analysis:

$$\begin{aligned} \phi_{E1} &= 0.0104 [deg], \text{ B rotation} \\ \phi_{E2} &= 4.513 \cdot 10^{-5} [deg], Z_{E2} = -0.0083 [mm] \\ Z_{E3} &= -0.019 [mm], Y_{E3} = 0 \\ X_{E4} &= -0.2 [mm] \\ \phi_{E5} &= 1.67 \cdot 10^{-3} [deg] \\ \phi_{E6} &= 1.47 \cdot 10^{-2} [deg] \end{aligned}$$

A combination of only positive B rotations, negative Z displacement and negative X displacement. Inserting into equation 5.85 gives the following worst case end effector error.

$$dEEp_{bcs} = [-0.200075 \quad 0 \quad -0.384697 \quad 0 \quad 0.026815 \quad 0]$$



## 5.6 Conclusion of this chapter

In this chapter a case study of the manipulator pose error has been conducted. Within the kinematic chain a set of physical errors has been identified and their relation to the kinematic chain has been established. The real kinematic chain has, as a result of the error analysis, emerged. The real end effector position has been compared with the ideal end effector position for all error sources alone, or as a combination of error sources.

All calculations of the end effector error has been carried out with the error equation  $dEEp_{bcs}$  [5.85]. Maple<sup>®</sup> has been used as the analytical tool to form the equations, and performing the numerical calculations, as well.

As stated in section 4.4.8 the error modelling based on forming of the real kinematic chain is more intuitive and easy than other methods. The designer is not forced to do any calculations into, lets say generalised errors, like methods presented by other authors. The real kinematic chain method also allows for combinatorial effects from several error sources.



## Chapter 6

### Experimental error mapping methodology

#### 6.1 Introduction

After the manipulator detailed design is finished, a prototype is to be built and tested. Activities connected to prototyping are described in the lower part of Figure 4.4. The final step of prototyping is “*prototype analysis*”. This task can be divided into two sub-tasks:

1. Performance measures
2. Error mapping

Performance measures are defined and necessary methodology is described in the ISO 9283 (1998) standard “*Manipulating Industrial robots, Performance criteria and related test methods*”. The outcome of such performance measurements are a numerical value of how large the deviation from the ideal selected reference is. If the prototype meets the requirements, the design is finished and manufacturing can begin.

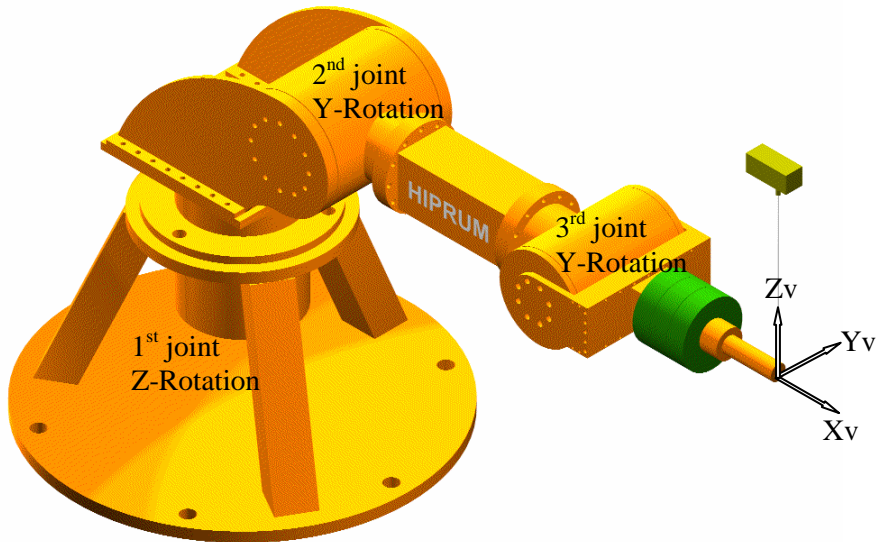
However, if the performance measurements are out of the requirements the second sub-task in the “*prototype analysis*”, namely the error mapping task, must proceed. In this chapter, an experimental error mapping methodology is outlined. With the benefit of giving the designer a possible way to split up the numerical value from the performance measurements and map these numerical values to specific parts or areas of the construction. This methodology gives the designer a tool to evaluate the existing design and guide his/her effort into the specific part where effort is most needed.

The outlined methodology of error mapping is based on the real situation where the designer has full access to, knowledge of and documentation of all parts of his prototype. Full openness is difficult to achieve by using commercial available manipulator systems. So, to help outline the experimental error mapping methodology an experimental manipulator and an experimental measurement system is used. This experimental manipulator and measurement system has the necessary openness.

To start with, this chapter gives a description of the open architecture test equipment while the outlined methodology is presented in paragraph 6.4

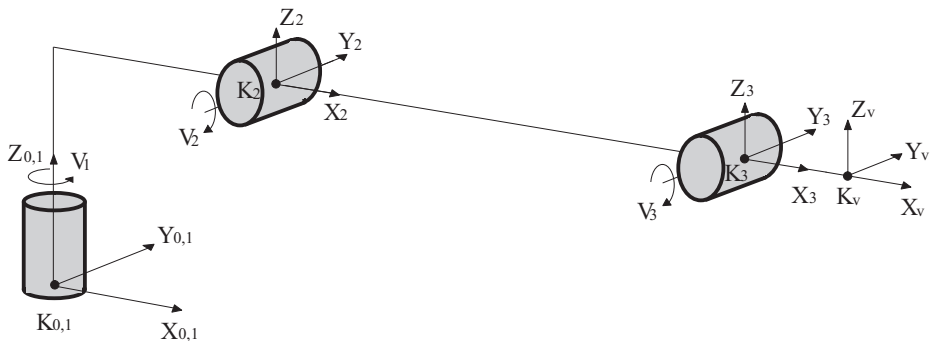
## 6.2 Manipulator testbed

A manipulator system of 3 joints as shown in the Figure 6.1 has been built, during my study, to gain knowledge/experience on the design of sub-sea manipulators.



**Figure 6.1** Manipulator system with three rotary joints

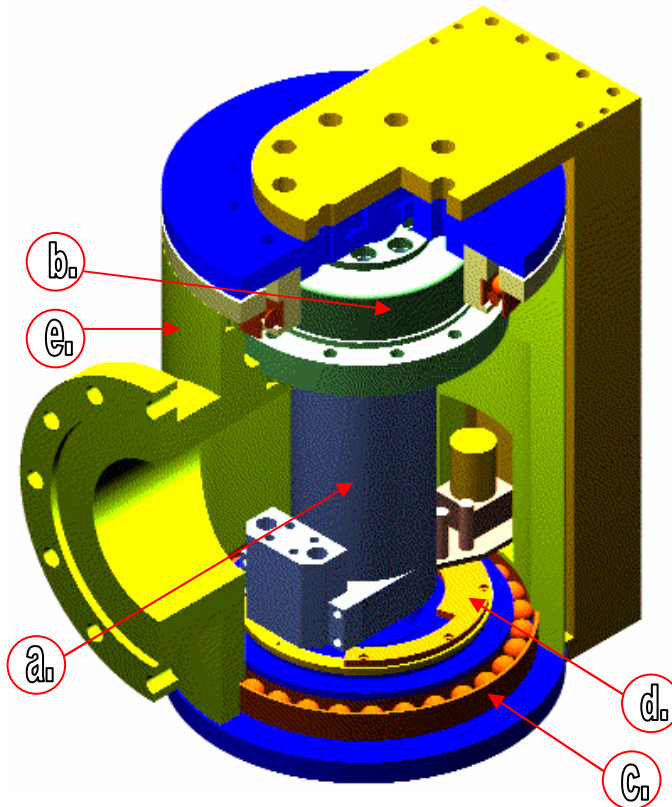
The simplified kinematic layout of the above manipulator system is shown in Figure 6.2. Here we clearly can identify the direction of the joint rotations. Joint 1 rotate around the axis  $Z_1$ , while joint 2 and 3 rotate along respectively  $Y_2$  and  $Y_3$  axis.



**Figure 6.2** Manipulator system, kinematic layout

### 6.2.1 Mechanical design

Every joint in the manipulator consists of the same elementary building blocks necessary to ensure smooth operation. Figure 6.3 shows a cut-through of joint 3 with its basic components.



**Figure 6.3** Joint layout

**a.)**

A hydraulic motor with a connecting servo valve. Fluid power components are chosen as drive machinery due to their ability to resist penetration of water. System pressure has to be kept above the surround water pressure to ensure that the leakage is running the correct way. During operation this is normally the situation.

Electric components could as well be used as drive elements but special attention to pressure compensation will be of outermost importance to prevent any short-circuiting. Such system will be more expensive than hydraulic drive units.

Most important is however that hydraulic components shows a low size/force ratio compared with electric components. Allowing us to keep the joint size to a minimum reducing its weight and thereby reduce the total deflection.

**b.)**

Gearing between motor and joint is necessary for two main reasons:

1. To be able to transfer the necessary torque.
2. To ensure a uniform minimum speed.

The latter point is of special interest in hydraulic drive systems due to some characteristic behaviour of fluid power engines. For example using a direct driven hydraulic motor in a joint close to robot-base one has to consider the minimum uniform speed of the motor. For an axial plunger hydraulic motor of a given size the minimum speed is approximately 5rpm. Connect this motor to a arm with a length of 1.8 meter and the tool-centre-point will move with a minimum linear speed of 942 mm/s. This is obviously not good enough for most applications. By using a harmonic drive as a speed reducer and given a ratio 1:120 of the drive the linear speed will be approx. 7.85 mm/s.

**c.)**

Two high precision thin section bearings allowing smooth operation between moving parts.

**d.)**

An angle measurement system, consisting of one resolver, and one gray code scale. In joint 2 the resolver is geared with a ratio of 1/10. This gives a unique resolver signal for only  $360/10=36$  degrees. The 3 bits gray code gives the sector which the resolver lay within. This total system gives an absolute measurement in a sector of 8 (3 bits) times 36 degrees = 288 degrees.

Resolver to digital conversion is done with a total resolution of 19 bits giving us an angle resolution of  $288/2^{19} = 5.49E-4$  degrees

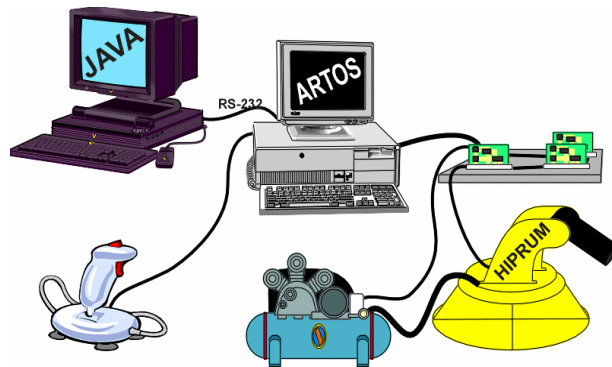
**e.)**

A precision machined joint in a lightweight aluminium alloy. A lightweight material is a preferable choice due to that this will reduce the total deflection of the manipulator. From the ROV system point of view a lightweight manipulator arm will interfere less with the buoyancy and gravity balance of the vessel.

Between the manipulator joints, a beam element is added. This beam holds the hydraulic servo valves and all the necessary electric and hydraulic connections.

### **6.2.2 Control system design**

All communication between the robot and the operator is carried out with a PC-based control system as shown in Figure 6.4.



**Figure 6.4** Manipulator (2 joint) and peripherals including a force feedback joystick

Two personal computers (PC) are used in the control system:

One with a JAVA based operator system, mainly built for the future manipulator operator, and with a functionality which is made with the commercial market in mind. Some operator functionality within this JAVA based system includes: Send and receive manipulator programs (cls –files), receive log files from the manipulator, analysing tools for log files.

On another PC the detailed control system is implemented including a laboratory user interface with detailed low level control system access. In general it has all functionality of the JAVA system, but with a more open architecture giving access to much more raw data used for testing and system start-up.

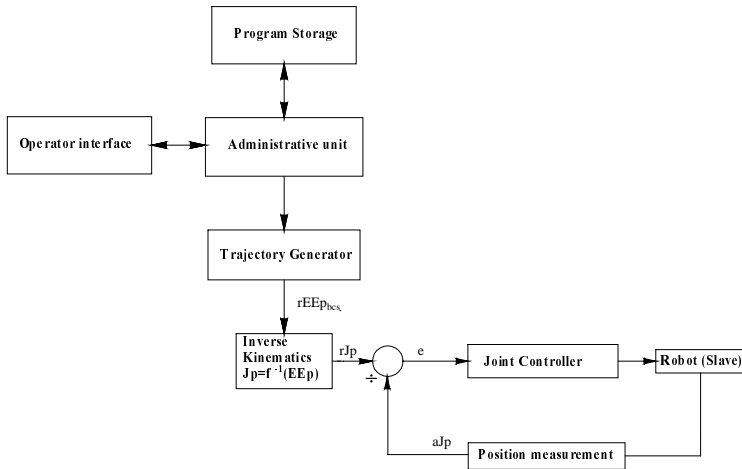
This PC-based control system of the manipulator is supported by a real time operating system ARTOS (Advanced Real Time Operating System). The application code of the control system is written in Borland C++.

The architecture of the manipulator conceptual trajectory control system is shown in Figure 6.5.

### **Trajectory control operating principle:**

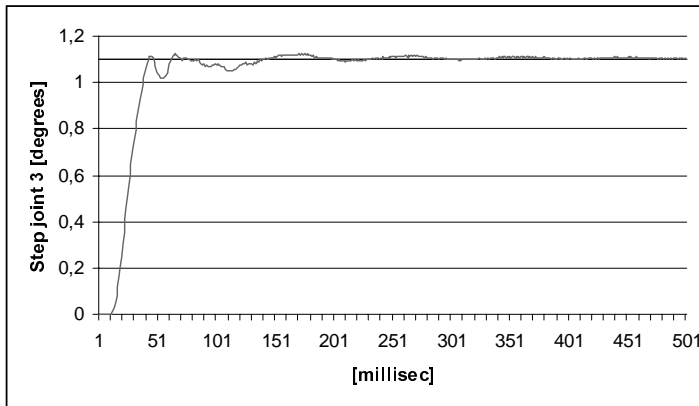
The operator can interact with the control architecture via an interface, keyboard/screen. From a set of possible commands the operator makes a manipulator program which the administrative unit put into the program storage. The operator program (cutter location source (CLS) file) may consists of a n (number) of the following commands:

FEDRAT/velocity	End-effector speed
GOTO/ X,Y,Z,A,B,C	Position and orientation of end- effector in base coordinates
WAIT/time	Wait a specified time
LOG/filename.log	Start logging. Logging of joint positions
LOG/off	Stop logging



**Figure 6.5** Manipulator conceptual trajectory control architecture

When time has come to execute the operator program the administrative unit fetch the new position/orientation/speed reference and transfer it into the trajectory generator. In the trajectory generator the distance to be driven is divided into increments where the increment size is dependent on the desired speed in the given period. The speed is controlled by an own acceleration unit. The calculated increment is added to the previous calculated position and out of the trajectory generator a reference position of the manipulator end-effector position with respect to the base coordinates is given  $rEEp_{bcs}$ .



**Figure 6.6** Step response joint 3. P controller with gain  $K_p=0.8$

The reference signal  $rEEp_{bcs}$  is led to the inverse kinematics module where the cartesian reference is transformed into joint position coordinates  $rJp$ , for every manipulator joint. From the manipulator joints the actual position is measured and subtracted from the references  $rJp$ . The position error signal is feed into the joint controller which control



the movement position of every joint so that the manipulator end-effector follow the desired trajectory.

The trajectory generator operates at a rate of 100Hz. while the servomodule operates at the rate of 1000Hz. When datalogging is activated the servomodule sends jointposition information to the computer memory and finally to a logging file.

Experiments were carried out to tune the servoloop controller parameters. Suitable parameters were found by step response tests. Figure 6.6 shows step response testing for the joint 3. Here a P regulator was programmed with a gain of 0.8.

### 6.3 Measurement system

To be able to identify the manipulator end-effector absolute position/orientation (pose) errors, while moving along a trajectory, an absolute measurement of the end effector pose is needed.



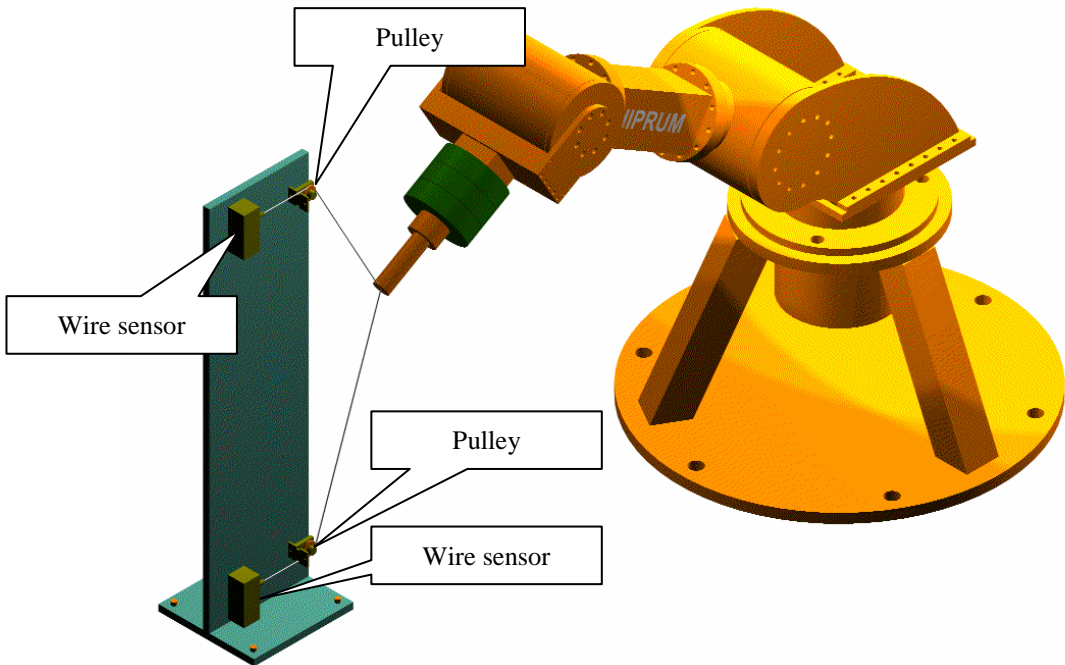
**Figure 6.7** LEICA LTD 500 Laser Tracking System (courtesy Leica AG, Switzerland)

The manipulator pose holds information of both position and orientation, given as six parameters  $\{X \ Y \ Z \ A \ B \ C\}$ . The three first parameters represent the end effector position while the final three represent the end effector orientation. The measurement system should be capable to measure all of these six components at the same time.

Usually, 3D laser measurement systems are used to carry out measurement of the moving manipulator end-effector. However, they are very expensive. In Figure 6.7 a 3D laser measurement system from LEICA AG, Switzerland is shown.

This system, a LEICA LT/LTD 500 contains a freely movable target reflector, which has to be attached to the manipulator end-effector, and a laser tracker, which consist of a mirror system with two motors and two encoders, a laser interferometer and a precision distance sensor. A PC is used as the controller for the measurement system and the device is capable of recording the target reflector with velocities up to 4m/s and end-effector accelerations of more than 2g. To receive position information, one target reflector is sufficient, but to determine the total end-effector pose (six parameters), at least three targets should be recorded simultaneously (Nof, 1999).

In my study, the costs of laser measurements became to high, and therefore it has not been possible to carry out full pose measurements. A 2D measurement system for identification of the errors in two dimensions (plane) has instead been adapted to help outline the error mapping methodology.



**Figure 6.8** 2D measurement set-up

Figure 6.8 shows the manipulator testbed with the 2D measurement system attached.

The 2D measurement system enables for an external measurement of only two parameters  $\{X Z\}$  out of the total 6 parameters  $\{X Y Z A B C\}$ . The parameter  $Z$  is defined as the vertical axis measurement and is very important in the error analysis. The reason for this is that the manipulator total compliance, in the  $g$ -force direction, is equal to the difference between the calculated reference of the  $Z$  parameter and the actual measured value of  $Z$ .

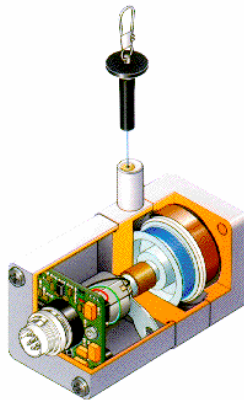
In the 3-joint manipulator arm, shown in the Figure 6.8 both joint 2 and joint 3 in the arm is rotational joints, which enable movement of the tool centre point (TCP) in Z-direction (gravitational). Since both joints operate in the Z direction, the joints rational compliance will contribute to an Z-axis error as well as the beam compliance. Thus, the Z axis measurement is of high importance.

By conducting measurements for the arm in different positions, from the worst case with the arm fully stretched out, to the situation were the manipulator arm is retracted in such a manner that the compliance is minimised, it is possible to retrieve quantitative information about the influence of the compliance of the arm in the given direction. These measures could be used to determine if the manipulator is within its constructional positional accuracy in the important ZX plane.

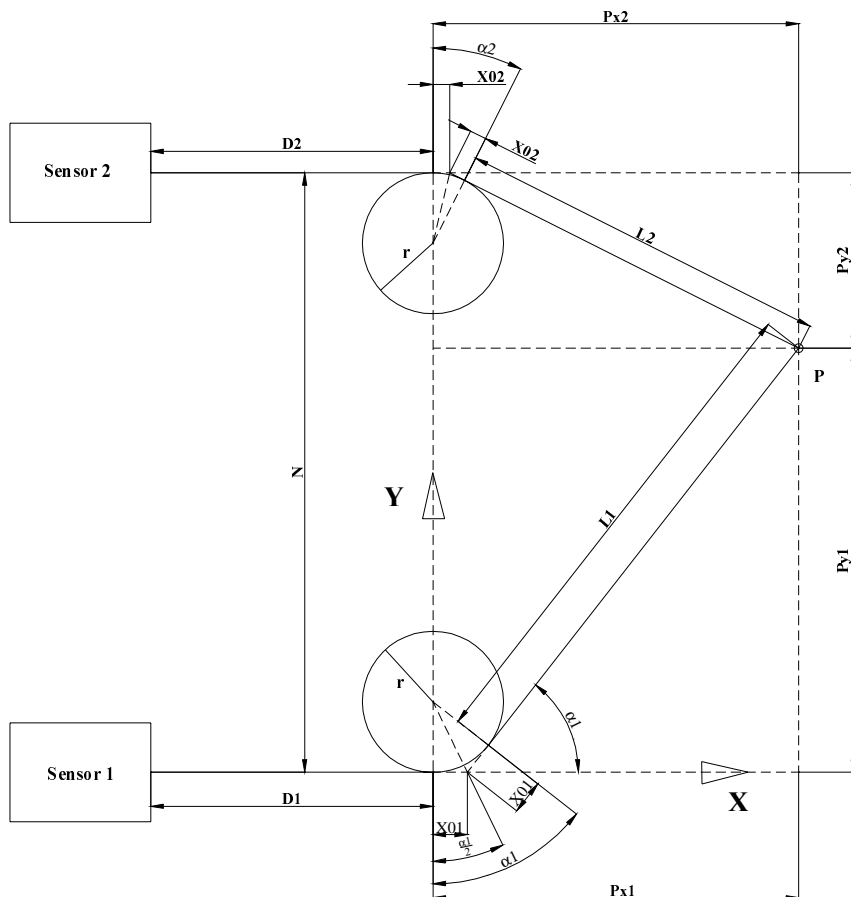
The 2D measurement system presented on the coming few pages, is a low cost solution, which enables to measure in a freely selected manipulator base plane ( to measure in a different plane than the ZX requires a reorientation of the measurement system itself). This measurement system can be used as a first step measurement equipment to indicate if the design is within the requirements.

### **Two dimensional measurement system**

Two wire sensors are used as the measurement devices in the measurement set-up shown in Figure 6.8. The linear wire sensor is an incremental rotary encoder with a wire arrangement transforming the linear movement into rotation of an incremental code scale. In Figure 6.9 an incremental wire sensor from ASM, Germany is shown.



**Figure 6.9** Incremental 1D wire sensor (courtesy ASM, Germany)



**Figure 6.10** Layout of measurement system

Two guide-wheels (pulleys) are applied to the measurement set-up to ensure linear travel of the wire in/out of the sensor housing. This is a precautionary measure, enabling increased lifetime due to secure operation.

Both wire sensors are attached to the manipulator end effector. When the end-effector moves the extension of the two wires are measured and a position is calculated. The calculation of the position is done with reference to a freely selected coordinate system  $XY$ , as shown in Figure 6.10.

The position to be deduced (2 dimensional) will be addressed to as:

$$P = [Px1 \quad Py1] \quad [6.1]$$

Based on the sensor readings this position P can be calculated.

From Figure 6.10 we can derive the following equations based on sensor 1.

$$La1 = RS1 - D1 \quad [6.2]$$

Where  $RS1$  = Sensor 1 reading. The quantity  $La1$  represents the length of the wire from the origin in the reference coordinate system  $[X,Y]$  to the point  $P$ , including the arc length around the pulley.

$$L1 = La1 - Lb1 \quad [6.3]$$

Where  $Lb1$  is the arc length of sensor 1 around the pulley.

$$Lb1 = r \cdot \alpha1 \quad [6.4]$$

$\alpha1$  [rad] is the angle of the wire arc length around the pulley.

From a geometrical analysis the following equations are derived.

$$\tan\left(\frac{\alpha1}{2}\right) = \frac{X01}{r} \quad [6.5]$$

$$\cos(\alpha1) = \frac{Px1 - X01}{L1 + X01} \quad [6.6]$$

$$\sin(\alpha1) = \frac{Py1}{L1 + X01} \quad [6.7]$$

For sensor 2 a similar analysis is conducted and the following equations arise.

$$La2 = RS2 - D2 \quad [6.8]$$

$$L2 = La2 - Lb2 \quad [6.9]$$

$$Lb2 = r \cdot \alpha2 \quad [6.10]$$

$$\tan\left(\frac{\alpha2}{2}\right) = \frac{X02}{r} \quad [6.11]$$

$$\cos(\alpha2) = \frac{Px2 - X02}{L2 + X02} \quad [6.12]$$

$$\sin(\alpha_2) = \frac{Py_2}{L_2 + X_{02}} \quad [6.13]$$

By rearranging eq. 6.6, 6.12, 6.7 and 6.13:

$$Px_1 = (L_1 + X_{01}) \cdot \cos(\alpha_1) + X_{01} \quad [6.14]$$

$$Px_2 = (L_2 + X_{02}) \cdot \cos(\alpha_2) + X_{02} \quad [6.15]$$

$$Py_1 = (L_1 + X_{01}) \cdot \sin(\alpha_1) \quad [6.16]$$

$$Py_2 = (L_2 + X_{02}) \cdot \sin(\alpha_2) \quad [6.17]$$

In expression 6.14 and 6.15 we insert 6.5 and 6.11 for  $X_{01}$ ,  $X_{02}$  while  $L_1$  and  $L_2$  can be derived from respectively eq. 6.2, 6.3, 6.4 and 6.8, 6.9, 6.10.

$$Px_1 = \left( RS_1 - D_1 - r \cdot \alpha_1 \cdot \frac{\pi}{180} + r \cdot \tan\left(\frac{\alpha_1}{2}\right) \right) \cdot \cos(\alpha_1) + r \cdot \tan\left(\frac{\alpha_1}{2}\right) \quad [6.18]$$

$$Px_2 = \left( RS_2 - D_2 - r \cdot \alpha_2 \cdot \frac{\pi}{180} + r \cdot \tan\left(\frac{\alpha_2}{2}\right) \right) \cdot \cos(\alpha_2) + r \cdot \tan\left(\frac{\alpha_2}{2}\right) \quad [6.19]$$

Similar substitutions of  $X_{01}$ ,  $X_{02}$  and  $L_1$ ,  $L_2$  are done for eq. 6.16 and 6.17.

$$Py_1 = \left( RS_1 - D_1 - r \cdot \alpha_1 \cdot \frac{\pi}{180} + r \cdot \tan\left(\frac{\alpha_1}{2}\right) \right) \cdot \sin(\alpha_1) \quad [6.20]$$

$$Py_2 = \left( RS_2 - D_2 - r \cdot \alpha_2 \cdot \frac{\pi}{180} + r \cdot \tan\left(\frac{\alpha_2}{2}\right) \right) \cdot \sin(\alpha_2) \quad [6.21]$$

The following two equations describe the interconnection between  $Px_1$ ,  $Px_2$  and  $Py_1$ ,  $Py_2$ .

$$Py_1 + Py_2 = N \quad [6.22]$$

$$Px_1 = Px_2 - (D_1 - D_2) \quad [6.23]$$

Eq. 6.18, 6.19, 6.20, 6.21, 6.22 and 6.23 are six equations with 6 unknown parameters:  $Px_1$ ,  $Px_2$ ,  $Py_1$ ,  $Py_2$ ,  $\alpha_1$ ,  $\alpha_2$ . However, these equation system are not solvable without using numerical methods.

Equation 6.22 and 6.23 are rearranged.

$$f1 = Py1 + Py2 - N = 0 \quad [6.24]$$

$$f2 = Px1 - Px2 - (D1 - D2) = 0 \quad [6.25]$$

The solution of this equality system can be done by treating the minimisation problem 6.26 based on the sum of squared  $f1$  and  $f2$ .

$$f(\alpha1, \alpha2) = f1^2 + f2^2 = (Py1 + Py2 - N)^2 + (Px1 - (Px2 - (D1 - D2)))^2 \rightarrow Min \quad [6.26]$$

By inserting the relations derived above, the complete objective function is formed as follows:

$$\begin{aligned}
 f(\alpha1, \alpha2) = & \\
 & \left( \left( RS1 - D1 - r \cdot \alpha1 \cdot \frac{\pi}{180} + r \cdot \tan\left(\frac{\alpha1}{2}\right) \right) \cdot \sin(\alpha1) + \right. \\
 & \left. \left( RS2 - D2 - r \cdot \alpha2 \cdot \frac{\pi}{180} + r \cdot \tan\left(\frac{\alpha2}{2}\right) \right) \cdot \sin(\alpha2) - N \right)^2 + \\
 & \left( \left( RS1 - D1 - r \cdot \alpha1 \cdot \frac{\pi}{180} + r \cdot \tan\left(\frac{\alpha1}{2}\right) \right) \cdot \cos(\alpha1) + r \cdot \tan\left(\frac{\alpha1}{2}\right) - \right. \\
 & \left. \left( RS2 - D2 - r \cdot \alpha2 \cdot \frac{\pi}{180} + r \cdot \tan\left(\frac{\alpha2}{2}\right) \right) \cdot \cos(\alpha2) + r \cdot \tan\left(\frac{\alpha2}{2}\right) + (D1 - D2) \right)^2 \\
 & \rightarrow Min
 \end{aligned} \quad [6.27]$$

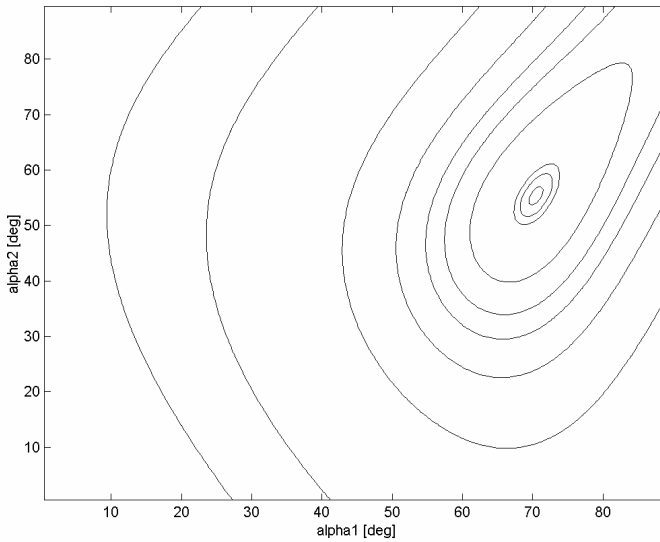
The mathematical expression 6.27 contains only to unknown variables  $\alpha1, \alpha2$ . The measurement physical layout is designed in such a manner that these variables  $\alpha1, \alpha2$  is to be kept within the following feasible area.

$$0.1 \leq \alpha1 \leq 89.9 \quad [6.28]$$

$$0.1 \leq \alpha2 \leq 89.9 \quad [6.29]$$

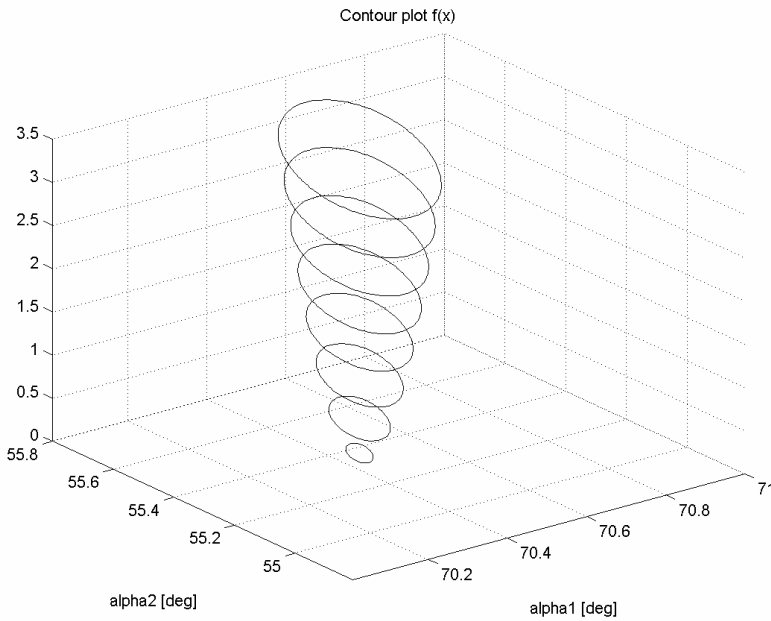
By keeping both angles less than 90 degrees, we decrease the possible measurement area, by not including the area in between the sensors. This is a precaution, reducing the risk for entanglement of the sensor and the robot. It is also necessary to keep both angles above 0 degrees to ensure that the sensors wires is not moved away from the pulleys.

Figure 6.11 shows a contour plot for function  $f(\alpha1, \alpha2)$ , for a given set of sensor readings.



**Figure 6.11** Function  $f(\alpha_1, \alpha_2)$ , for a given measured value (sensor 1 and 2)

In Figure 6.12  $f(\alpha_1, \alpha_2)$  is drawn for the variables  $\alpha_1, \alpha_2$  closer to minimum solution, indicating a solution  $\alpha_1 \approx 70.4$   $\alpha_2 \approx 55.3$ .



**Figure 6.12** Function  $f(\alpha_1, \alpha_2)$ , 3D contour plot close to minimum point



To solve the minimisation task in 6.27 Matlab® programming language was firstly selected.

In Matlab optimisation toolbox the root finding function “fsolve” was used to derive the optimal values for the unknown parameters  $\alpha_1, \alpha_2$ . The “fsolve” algorithm solves nonlinear equations by a least squares method. Equations have to be in the form  $f(x)=0$  where f and x may be vectors or matrices.

To control the performance of the “fsolve” algorithm there are several optional parameters. The function call “fsolve(f(x),initial guess, options)” allows a vector of optional parameters to be defined. Options(2) is a measure of the precision required for the values of  $\alpha_1, \alpha_2$  at the solution. Options(3) is a measure of the precision required of the objective function f(x) at the solution. The solving algorithm used by “fsolve” is a Levenberg-Marquardt method. A complete description of the “fsolve” function can be found in Matlab® reference.

The listing below shows the structure of the “postprocessor” build in Matlab®. (The numerical calculations with the “fsolve” algorithm is carried out **after** the sensor measurements, thereby the name postprocessor)

Postprocessor Algorithm:

```

Define global variables and constants;
Read input file, consisting of RS1 and RS2 in a text file from the sensor logging
system;
Calculate size of input file (eg. number of rows j);
Preallocate memory of variables to speed up calculations;
Main Loop
For(i=1 to j)
    Set options for “fsolve” function;
    Define function to be solved in the form f(x)=0;
    Initial guess G for a solution of unknown variables;
    Call “fsolve” function with the input parameters (f(x),G,options)
    Outputs parameters  $\alpha_1$  and  $\alpha_2$ 
    Calculate Py1 from eq. 6.20;
    Calculate Py2 from eq. 6.21;
    If Py1 not equals (N-Py2) then
        Error message;
    Calculate Px1 from eq. 6.18;
    Calculate Px2 from eq. 6.19;
    If Px1 not equals (Px2 - (D1 - D2)) then
        Error message;
    Save results from Px1, Py1,  $\alpha_1$  and  $\alpha_2$ ;
End Main Loop
Write data Px1, Py1,  $\alpha_1$  and  $\alpha_2$  to output file –ascii format for storage

```

To verify the calculations from Matlab® a non-linear optimisation system NOSYS® has been applied. NOSYS® is a user-friendly and engineering oriented problem solution tool for applied nonlinear optimisation. It includes the following components (see Figure 6.13) (Koch, 1994):

**DINOS:**

User interface where all necessary input/output files is generated.

**MENOS:**

Package of mathematical optimisation methods.

**GRANOS:**

Graphical solver of 2-dimensional optimisation tasks.

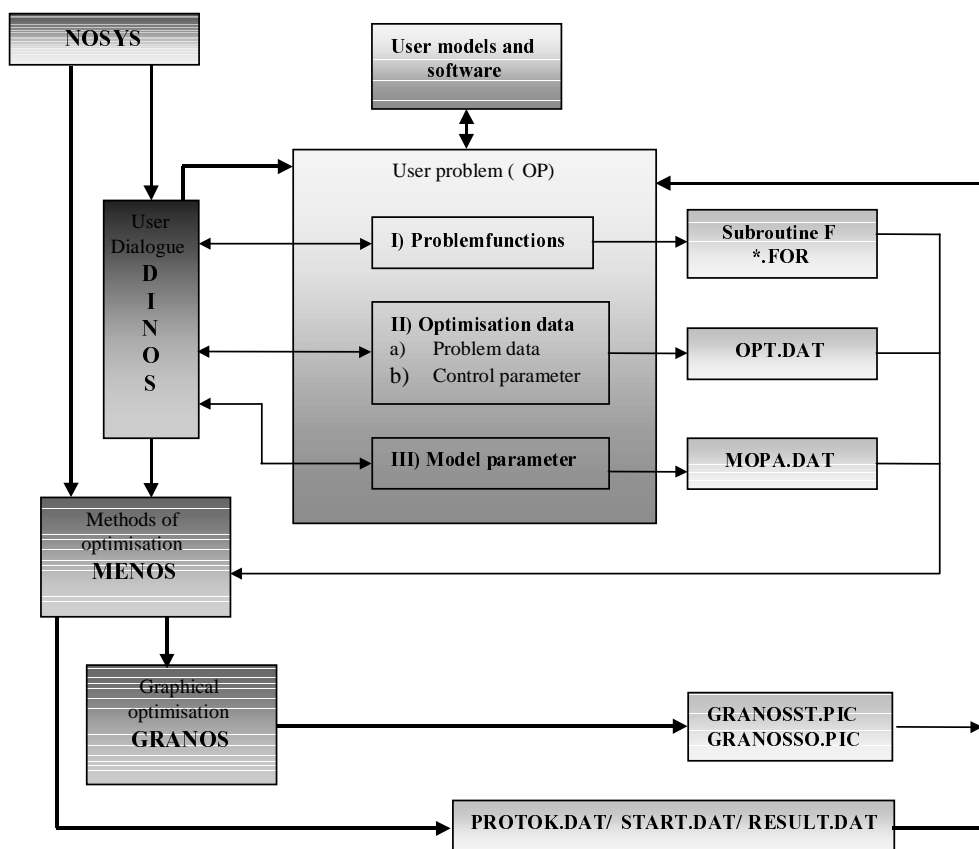
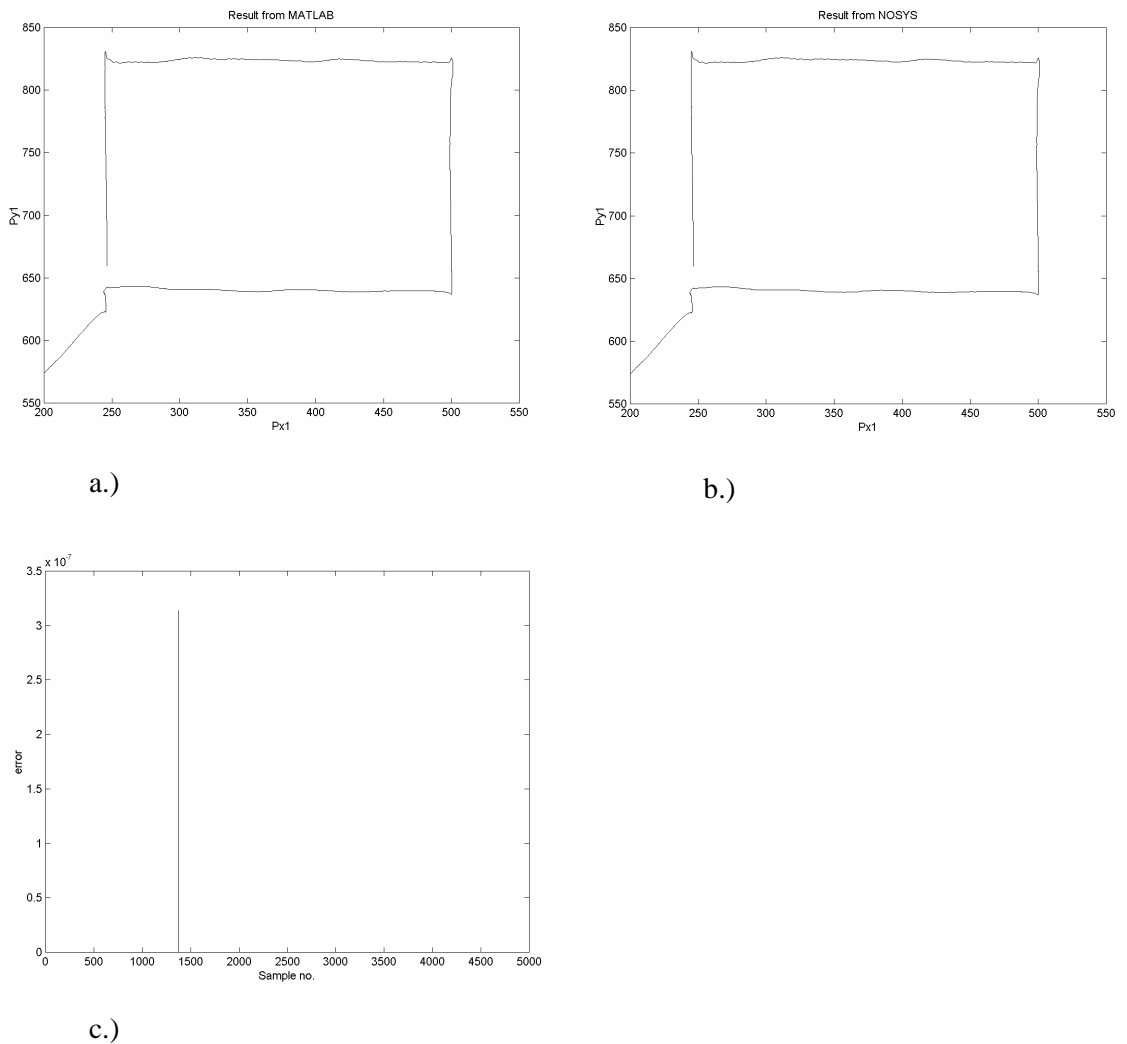


Figure 6.13 NOSYS® flowchart (Koch, 1994)

A modified LEVITIN/POLYAK algorithm (Koch, 1983) is used for solving problem 6.27, also considering constraints defined in 6.28 and 6.29. The algorithm uses a successively quadratic approximation of  $f(\alpha_1, \alpha_2)$  from 6.27 for determining the iteration directions and a suitable ARMIJO step length procedure. The result  $(\alpha_1, \alpha_2)$  from the NOSYS® optimisation calculations is to be fed into equations 6.18 and 6.20, from where the position  $P = [Px1 \quad Py1]$  can be calculated.



**Figure 6.14** a.) Result file MATLAB® optimisation. b.) Result from NOSYS®. c.) Comparison of functional value  $f(\alpha_1, \alpha_2)$  .

NOSYS® and MATLAB® based postprocessor units gave “close to identical results” for the  $(\alpha_1, \alpha_2)$  calculations. Figure 6.14.a,b shows a plotting of  $[Px_1, Py_1]$ , for a large series of input sensor readings, calculated with both optimisation tools. Figure 6.14.c shows the comparing error function:

$$e = f(\alpha_1, \alpha_2)_{MATLAB} - f(\alpha_1, \alpha_2)_{NOSYS}$$

The error function is equal to zero for all calculations.

The input sensor reading (-ascii) files is normally very large. The total calculation time used by the two different postprocessor systems, with its connected solver algorithms is depended on the size of the input file. If we want to use our postprocessor for real time calculations in a robot control system, the time elapsed by computations will be of critical interest. Both postprocessor made were not optimised for minimum processing time, however the following quantitative remarks could be of interest for future analysis.

- Optimisation with the NOSYS® selected optimisation algorithm was faster than MATLAB®. In fact, the calculation time with NOSYS® was only around 20% of the MATLAB® processing time.
- MATLAB® calculations was unstable (crashed) at some tests.

## 6.4 Error mapping methodology

After the design of a robot is finished and a prototype has been built, performance measures are undertaken to see how well our system perform with respect to our requirements. The ISO 9283 (1998) standard “Manipulating Industrial robots, Performance criteria and related test methods” describes the test methods for the following:

- Pose accuracy and repeatability
- Multi directional pose accuracy variation
- Distance accuracy and distance repeatability
- Position stabilization time
- Position overshoot
- Drift of pose characteristics
- exchangeability
- Path accuracy and path repeatability
- Path accuracy and reorientation
- Cornering deviations
- Path velocity characteristics
- Minimum posing time
- Static compliance
- Weaving deviations

The “test” team may carry out a set of necessary tests according to the ISO standard, and by this identify the deviations or errors from the ideal performance. However, these performance measures does not state **why** or from **where** these errors origin from.

From a design point of view **why** and **where** are important questions to be answered to be able to change the existing design, and as well gain knowledge for the future design projects.

In Chapter 4 an error model was developed for the manipulator pose, this model can be compared with the performance measurements of the pose accuracy. If the calculated errors are identical or “close” to identical, the designer will have a good knowledge about both **why** and from **where** the errors are arising from. If the design needs to be modified, calibration of the kinematic model may be done, with respect to the repeatable errors. If this is still not enough the random errors must be corrected, this is normally achieved with a redesign of the physical manipulator itself. After the modifications are finished, new performance tests could be carried out, to see if the new expected accuracy is achieved.

If the performance measurements, at some point, are not in accordance to the expected calculated error, the designer has not been precise enough in the error modelling process, and he/she actually does not have any idea were to put his effort when improvement is necessary. The design team has lost track of **why** and from **where** the error origin from. However, as will be seen, experiments will guide the designer back on the track again.

In general, the error mapping methodology presented in this chapter requires:

- Full access to the manipulator control architecture
- Complete 6 parameter manipulator pose measurement system

In the following description of the error mapping methodology, the 3 joint testbed and external measurement system, presented in the beginning of this chapter, are used. The experimental manipulator has a control architecture which have the necessary openness, but the measurement system is **not** capable to measure all the necessary six parameters  $\{X \ Y \ Z \ A \ B \ C\}$  of the pose. However, the lack of capability of the measuring device is not to important in the context of this chapter since the focal point is to present an experimental error mapping methodology. The presented measurements are only used to help outline the methodology itself.

In Chapter 4, control architectures for manipulators were discussed, and it was shown that the joint position control loop was the central part of the architecture. Such a control architecture includes a position measurement of every joint movement. Both joint measures and measures from an external measurement system will together give the design team an idea of which part of the manipulator system that contribute to the error and by how much.

A trajectory, as shown in Figure 6.15, is selected as the reference path. The corresponding input operator file (CLS file) is shown at the right hand side of Figure 6.15. In the CLS file the square corner coordinates is given in manipulator base-coordinates. The programmed velocity of the end-effector is 50 mm/s.

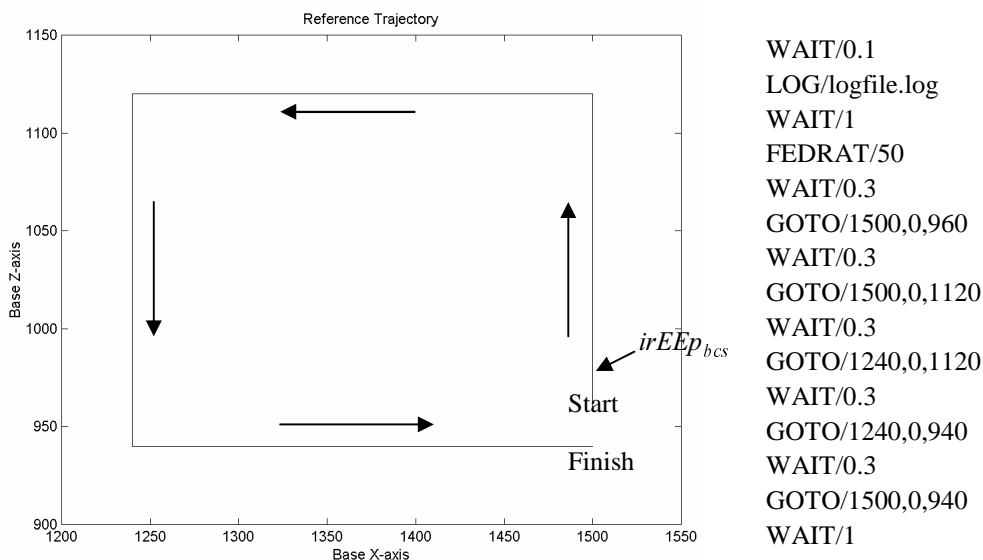
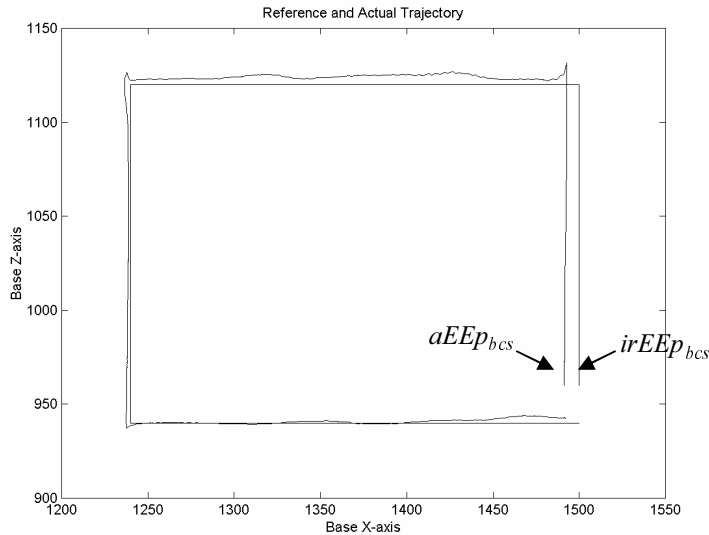


Figure 6.15 Reference trajectory

The input trajectory from the operator is denoted  $irEEp_{bcs}$  since it describes the ideal input reference end-effector position with respect to the base-coordinate system. While carrying out performance measurement for the path accuracy, the  $irEEp_{bcs}$  is compared with the actual obtained path  $aEEp_{bcs}$ , made from the external measurement system. Figure 6.16 shows the reference and the actual obtained trajectory.

In the Figure 6.16 we clearly see the deviations between the reference and the actual path, but the performance test hold no information about why and from where the error origin from. For the designer this is not enough, he/she must now from where the error origin from, to enable hardware or software modifications.



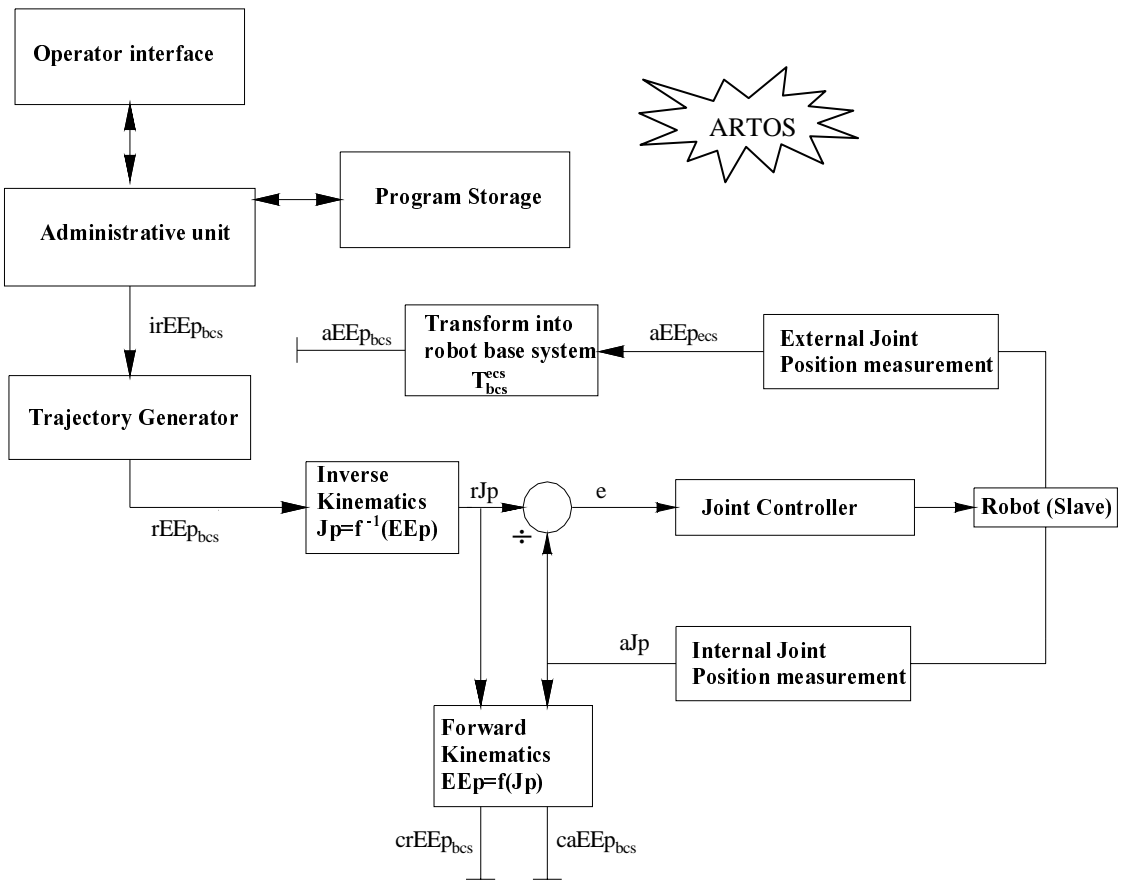
**Figure 6.16** Reference and actual trajectory, derived from external measurements

In this process of error mapping, we start with the reference trajectory and finally end up with the actual trajectory, both shown in Figure 6.16. The steps in between is explained with reference to the control architecture of the manipulator system. A control architecture is shown in Figure 6.17. This architecture was described in section 6.1.2. The external measurement set-up was described in section 6.3.

The first error source that contributes to the total system error for the given trajectory is the trajectory generator and the inverse kinematic calculation. The trajectory generator, described in Chapter 4, divides the distance to be driven into increments, while the kinematic model transfer the Cartesian reference into joint coordinates. These numerical calculations may suffer from round off and/ or truncation errors. In addition to these, digitisation errors arise when numbers are transferred between floating point and integers.

To identify how much these computing errors contributes to the total error, the ideal reference  $irEEp_{bcs}$  is plotted against the calculated reference  $crEEp_{bcs}$ . The calculated reference  $crEEp_{bcs}$ , as shown in the architecture, is the  $irEEp_{bcs}$  after it has passed through the trajectory generator  $rEEp_{bcs}$ , then through the inverse kinematic  $rJp$ , and finally through the dummy forward kinematic model  $crEEp_{bcs}$ . The dummy forwards kinematics is necessary to enable a comparison in Cartesian coordinates.

In Figure 6.18, the  $irEEp_{bcs}$  and the calculated reference  $crEEp_{bcs}$  is printed.



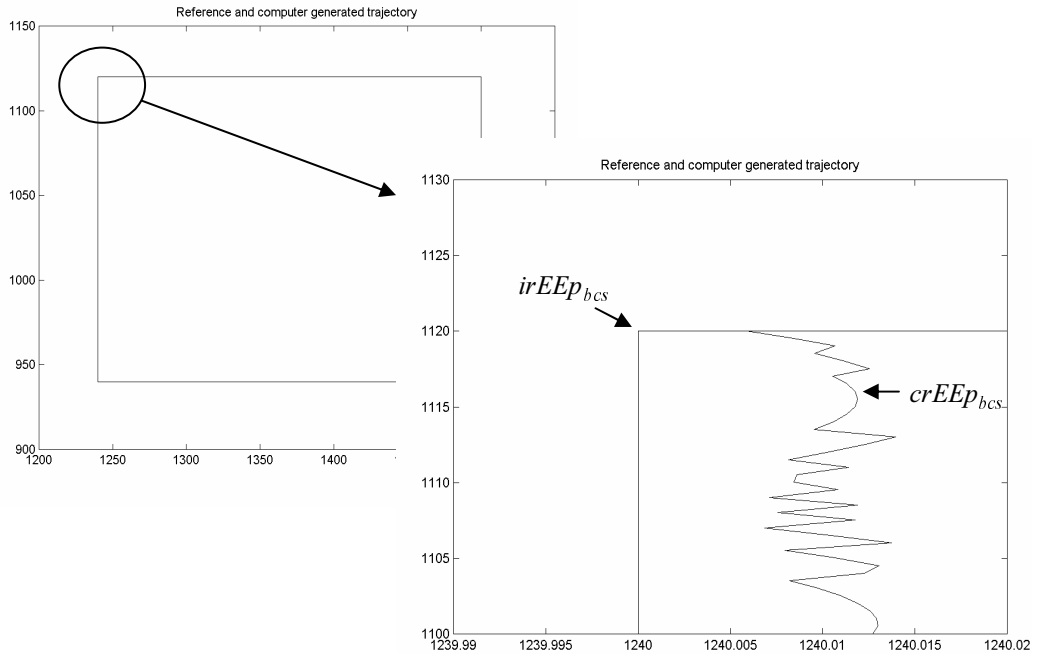
**Figure 6.17** Manipulator error mapping control architecture

From the extract in Figure 6.18, we find that the error arising from calculations are very small. Normally, truncation and round-off errors are made very small in modern (micro) computer system, and normally they may be neglected.

However, the digitisation errors are more critical and must be addressed carefully in the manipulator design phase.

In the experimental manipulator testbed control architecture, the reference joint position  $rJp$  is digitised into integer format with a resolution corresponding to the joints measurement system. Thus, the resolution of the joint measurement system, will be of great importance to the calculated reference.





**Figure 6.18** Reference trajectory and calculated trajectory

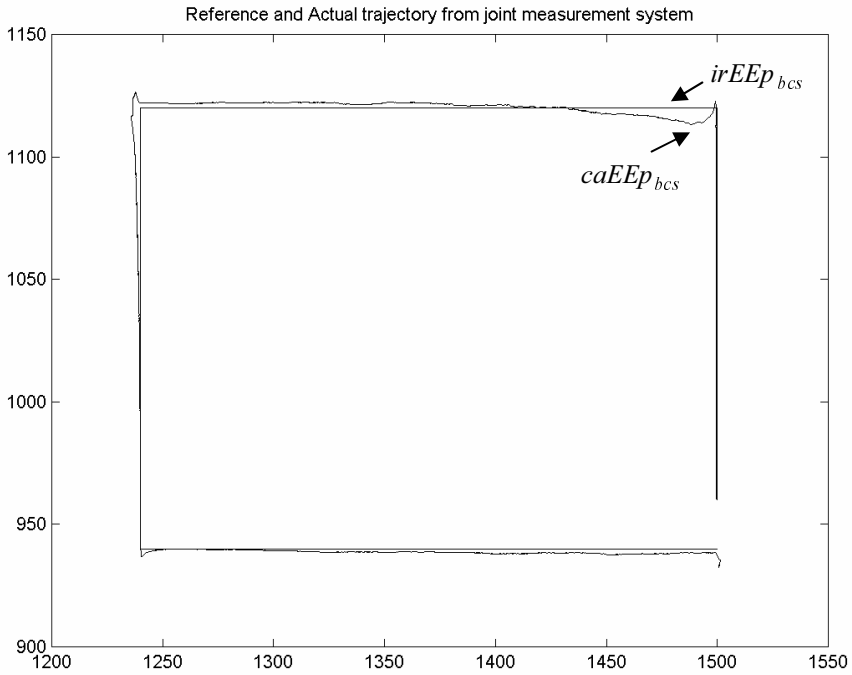
The next step in the error analysis will be to look at the actual joint positions  $aJp$  and how these are compared with the ideal Cartesian reference  $irEEp_{bcs}$ . The dummy forward kinematic unit is again used to transform the measured joint positions  $aJp$  into the Cartesian equivalent  $caEEp_{bcs}$ . This is depicted in the Figure 6.19.

Figure 6.19 is an interesting picture, since it gives information of, how well it would be possible to follow a path if the kinematic model is 100% correct. The errors in the above Figure 6.19 are mainly caused by manipulator joint dynamics.

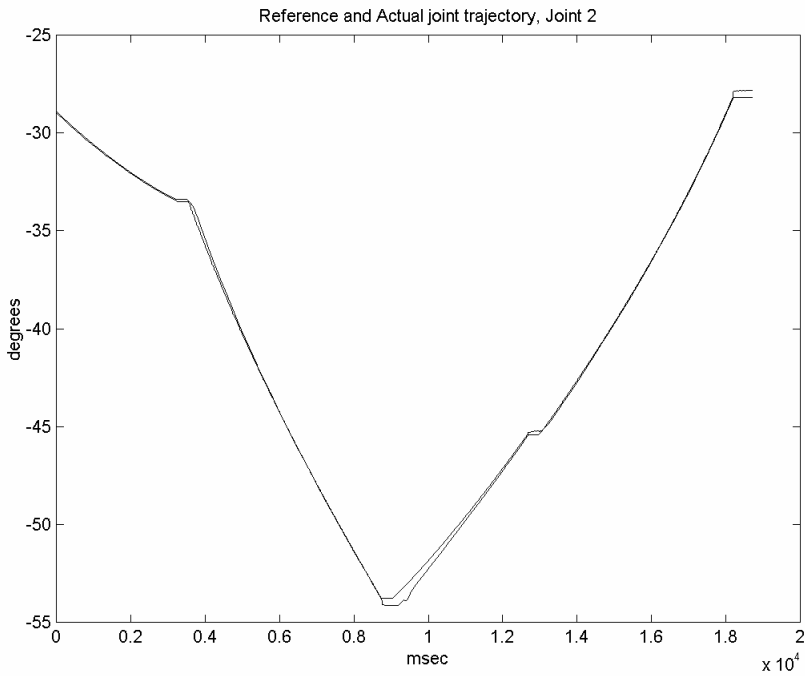
The joint controller tries to eliminate the deviations, but may not be able to follow the path accurate enough because of the incorrect setting of controller parameters. This leads to a non-optimal response of the joint movement.

Since the calculated actual reference  $caEEp_{bcs}$  arise from a number of actual joint positions  $aJp$  it is natural to identify how the different joint contributes to the error between the  $caEEp_{bcs}$  and the ideal reference  $irEEp_{bcs}$ . This is possible by investigation of the reference joint position  $rJp$  and actual measured position  $aJp$ , both given in joint coordinates. Figure 6.20 and 6.21 shows the reference and actual joint position for both joint 2 and joint 3.

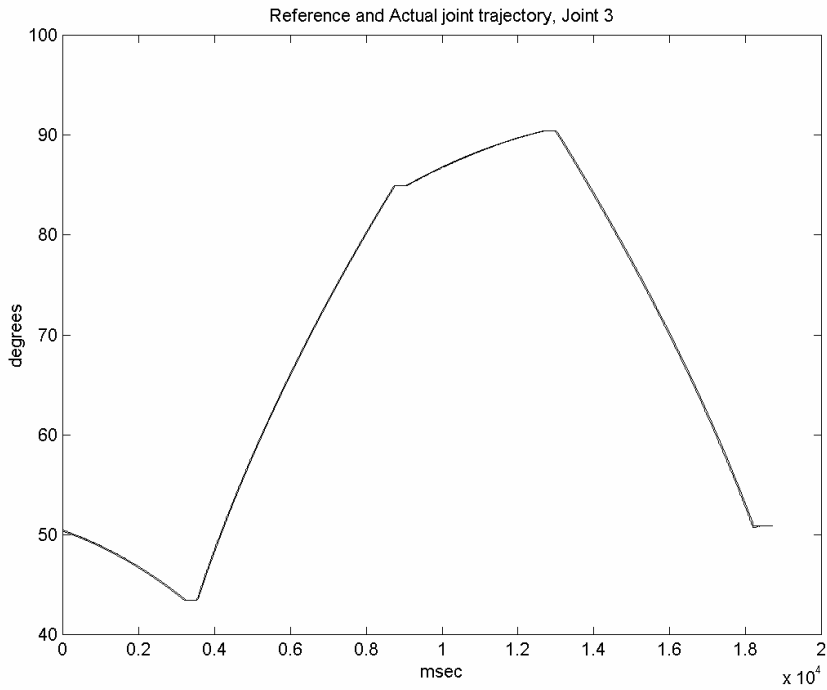
From these two figures, we find that joint three has much better path following capability as joint two, thus effort for improvement should be addressed to joint number two.



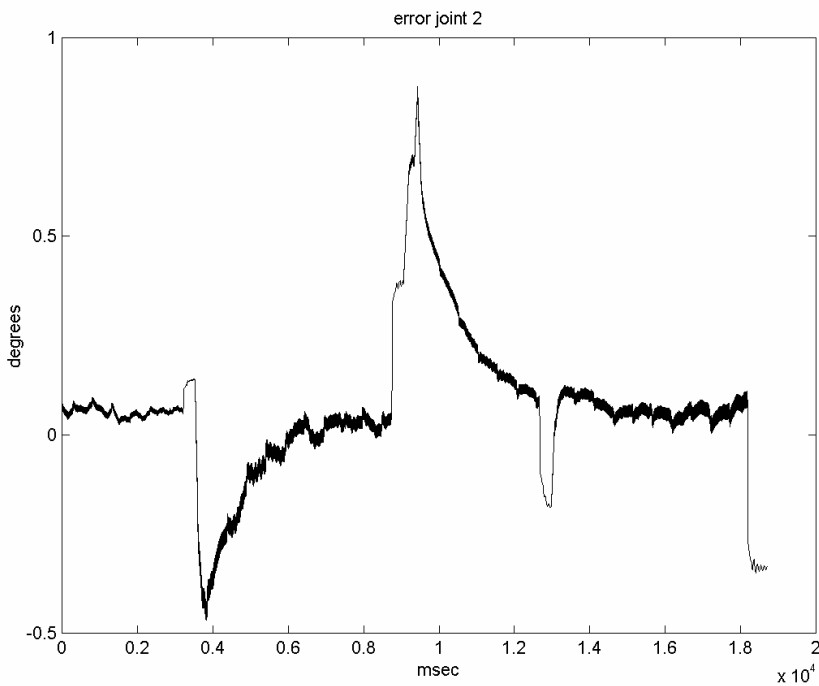
**Figure 6.19** Reference trajectory and actual trajectory based on joint measurements



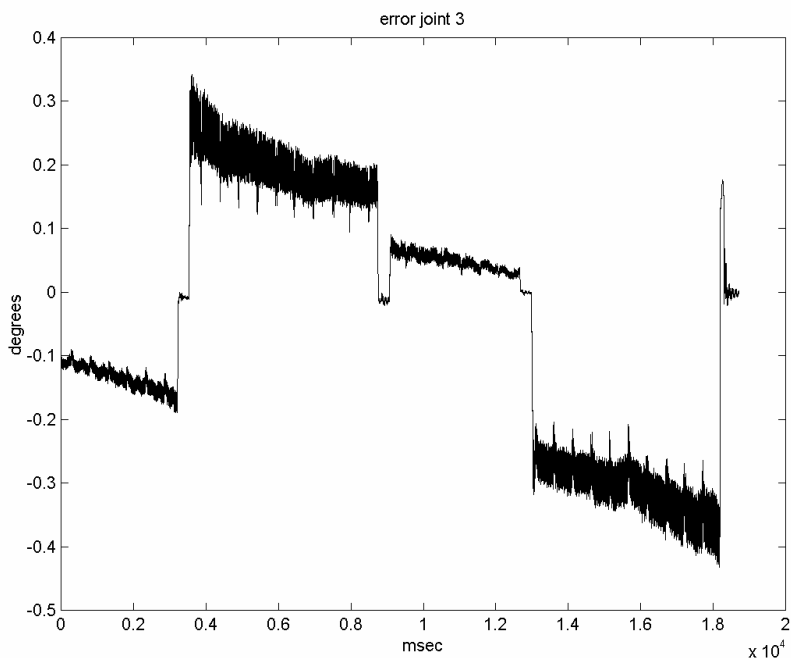
**Figure 6.20** Reference joint trajectory and actual joint trajectory, joint 2



**Figure 6.21** Reference joint trajectory and actual joint trajectory, joint 3



**Figure 6.22** Servo loop error, joint 2



**Figure 6.23** Servo loop error, joint 3

Figure 6.22 and 6.23 prints the error function  $e$  with respect to the passed time, this is just another way to present the result in Figure 6.20 and 6.21 since  $e = rJp - aJp$ .

Finally the external measurement system is used to identify the total error, as shown in the Figure 6.16. This measurement holds information of all errors involved.

Compared with joint measurements as described above, the external measurement holds information of the error in the kinematic model and the compliance of the beam structures.

To sum up the experiments the following error mapping can be done based on experimental results where the three errors are identified:

**E computing** (as described with reference to Figure 6.18)  
Computing/ digitisation error

**E joint dynamic** (as described with reference to Figure 6.19, 6.20, 6.21, 6.22, 6.23)  
Errors in the joint servo system, if the feedback measurement encircle the gearing system, then joint compliance is also included in this measurement

**E total** (as described with reference to Figure 6.16)

The E total error includes all error sources in the set-up. In addition to the above mentioned errors here we find error in the kinematic model and also the beam compliance.

Error in the external measurement system and the transformation between external measurement system and the robot base- coordinates are also included in the error chain. Thus, the measurement errors must be kept to a minimum.

The E total can be summed together by the following components, neglecting measurement errors:

$$E_{\text{total}} = E_{\text{computing}} + E_{\text{joint dynamic}} + E_{\text{kinematics}} + E_{\text{beam compliance}} \quad [6.30]$$

The two late elements of this chain (E kinematics + E beam compliance) is identified by rearrangement of the equation 6.30

$$E_{\text{kinematics}} + E_{\text{beam compliance}} = E_{\text{total}} - (E_{\text{computing}} + E_{\text{joint dynamic}}) \quad [6.31]$$

Thus, we have gained knowledge of from **where** the total error origin from, and we have managed to get course numerical values for some cluster of errors sources. This is important information when the designer wants to look back into the detailed design again in a situation when necessary to make changes. In the process of redesign the designer can use the numerical values gained from the experiments combined with the error analysis methodology presented in Chapter 4 to understand **why** the errors arise.



## Chapter 7

### Conclusion and recommendations for future work

#### 7.1 Conclusion

The importance of an increased research into the area of deep sea intervention has been clearly identified in this study.

Most intervention work tasks, which previously were conducted by the human divers, are going to be automated, and specialised or flexible intervention tools will be built. This dissertation gives the designer a methodology for aiding the design procedure of such equipment.

A methodology for aiding to optimally and formally develop and design new generation underwater unmanned manipulation systems is presented. First a requirement definition matrix is defined, which can be referred for aiding to define a complete set of requirements for the new development of underwater manipulation systems. Further, combining the requirement definition matrix with the 3<sup>rd</sup> dimension of R&D domains, we get an overall concept and methodological framework.

Standing on the point of system developers and designers, this methodological framework can be used as a reference framework in their developing and designing work, from which they can formally define the requirements and formally link the requirements with the development and design of subsystems.

Standing on the point of methodology researchers, this framework gives methodology researchers a formal overview of what and how many domains they can contribute to develop methodologies to aid system developers and designers in developing and designing new generation of ROV-based underwater manipulation systems.

Further, the serial manipulator is identified to be a natural solution for the workspace constraint environment, and the serial structure is selected for further analysis in this dissertation. The different work task will influence the serial manipulator design in many ways like, payload, precision, speed, reach, dexterity, robustness etc. Many existing manipulator systems are well adjusted for most of these requirements, but the precision term has been less investigated. So, I have identified the necessity of developing a precise design methodology, for aiding system designers to design a precise underwater serial arm manipulator working in workspace constraint environment, capable of doing high precision work.

During the design stage of a precision manipulator system the necessity of evaluation loops has become clear. The design team can evaluate the effect of their design considerations/decisions upon the precision requirement. Thus, an iterative method of

“design – evaluation - design-evaluation” is suggested as part of the precise design methodology. To gain knowledge of the effect of your possible choices enables the design group, before too many ruling decisions has been made, to make the optimum solution.

The suggested, error analysis methodology involves a 5 step procedure:

- 1<sup>st</sup> Identify the ideal kinematic chain, and the ideal end effector pose
- 2<sup>nd</sup> Identify physical error components in the design and their relation to the kinematic chain
- 3<sup>rd</sup> Identify the real kinematic chain, and the real end-effector pose
- 4<sup>th</sup> Calculate influence of single error components to the end-effector
- 5<sup>th</sup> Calculate influence of all (or a selection) of error components to the end effector

Compared with existing error analysis methodologies the presented method demands less calculations as user input, and thus are more intuitive. The methodology can handle combined error source analysis. The methodology is easily transferred into software architecture.

During the early steps of design of underwater manipulators, conceptual control strategies and architectures plays an important role. In this dissertation I have identified and studied architectures for:

- master-slave control
- joystick control and
- trajectory control

During the development of these conceptual strategies I have strengthened the idea that the control strategy and architecture is of major importance to the coming design process.

By looking into the different architectures, master-slave, joystick, and trajectory, I would say that they are all good for their special purpose. However, by combining them the designer may create a future control system which is of very high capability addressing a set of special work task and working conditions. Combining these architectures is not a very complicated task, since most of the blocks are realised as software solutions. The future designer may use section 4.3 as a reference, while selecting her/his control strategy for the equipment to be built.

The ISO 9283 standard “Manipulating industrial robots, performance criteria and related test methods” describes the test methods for a set of important performances of the manipulator system. However, the outcome of the performance measurements is a numerical value of how big the deviation from the ideal selected reference is. Performance measures does not state **why** and from **where** the errors origin from. This



latter is especially important for the designer to have more knowledge about, because he/she may have to make changes to the construction, if some of the performance measurement is out of the specified requirements. In the final part of this dissertation, an experimental error mapping methodology is outlined. With the benefit of giving the designer a possible way to split up the numerical value from the performance measurements and map these numerical values to specific parts or areas of the construction. Experiments to help outline the methodology is presented.

## **7.2 Recommendations for future work**

The research in this dissertation has provided a methodology for aiding the designer to create subsea manipulator systems where the precision requirement was selected as the main focus. The selection of precision manipulators was made from the methodology framework for developing underwater unmanned manipulating systems. A further future research could be selected from this methodology cube, were the research topic will be to investigate how the other different parts of the ROV system, must be adapted to new working conditions and new high precision work tasks. All system components must be adapted to the new depth and the requirements of the working conditions.

In the same scope as the dissertation itself I would suggest to follow up with a error analysis software development, making the error analysis as easy as possible for the user, normally for the designer or his/her team.

Case studies of existing manipulator systems should be carried out to investigate how well they are physically built, to perform the desired work task. Such a methodology could prove handy, acting as a qualifying method of the ROV- manipulator contractors.

Finally, the design methodology and error evaluation technique has mainly been developed based on the demand from the oil industry, however the methodology is universal and it would be interesting to transfer the ideas to other operating environments, like onshore, space or say hazardous areas.



## References

- Andeen, B. A. (1988). "Robot design handbook". McGraw-Hill Book Company.
- ASM GmbH. WS Position sensors. Product information. KAT-WS-E-96.
- Aust, E., Gustmann, M., Niemann, H.-R., Schulhreis, G. F. (1992). "1100 m subsea robot start wet working". Intervention ROV. 1992. San Diego, California, USA.
- Aust, E., Gustmann, M., Niemann, H.-R. (1994). "A six-axes robot for deep water applications". Underwater Intervention, 1994. San Diego, California, USA.
- Aust, E., Niemann, H.-R., Boke, M., Gustmann, M., Wesche, A. (1995). "Six-years development in subsea robotics", Underwater Intervention 1995. Houston, Texas, USA.
- Broome, D. R., Langrock, D. (1994). "Development of a Telerobotic Underwater Manipulator". INSIGHT. 36(9).
- Cohen, J.E., Small, C., Mellinger, A., Gallup, J., Sachs, J. (1997). "Estimates of Coastal Populations". Science. 278(5341): pp. 1211-1212.
- Conte, G., Serrani, A. (1996). "Modelling and Simulation of Underwater Vehicles". Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design. Dearborn, MI, USA.
- Denavit, J., Hartenberg, B. S., Evanston, I. (1955). "A Kinematic Notation for Lower-Pair Mechanism Based on Matrices". J. Applied Mechanics.
- Dunnigan, M. W., Lane, D. M., Clegg, A. C., Edwards, I. (1996). "Hybrid Position/force Control of a Hydraulic Underwater Manipulator". IEE Proc.-Control Theory Appl. 143(2).
- Elle, O. J., Johnsen, K., Lien, T. K. (1994). "Mechanical Analysis of High Precision Manipulator". Scandinavian Symposium on robotics. October 4-6, 1994.
- Elle, O. J., Thomessen, T., Lien, T. K. (1993). "Tele-manipulation using guided contour tracking", Proceedings International Offshore and Polar Engineering Conference (3:1993: Singapore).
- Fattah, A., Kasaei, G. (2000). "Kinematics and dynamics of a parallel manipulator with a new architecture". Robotica (2000) vol.18. Cambridge University Press.
- Garmulewicz, J.A. (2000). "The Application of the Auto-ROV System in Deepwater Field Developments". Proceedings of UI 2000.

Greig, A. R., Wang, Q., Broome, D. R. (1992). "Weld Tracking with a Manipulator Fitted with a Compliant Wrist Unit". Proceedings of ROV 92.

Hallset, J. O. (1996). "ROV systems and operations". Oceaneering a.s. report: 36073/TEC/05/DOC/236.

Harmonic, Drive. Harmonic drive gear component sets and units HFUC series Manufacturer leaflet HFUC series.

Haugvaldstad, J. (1994). "Converting from diver to ROV", Underwater Intervention 1994. San Diego, California, USA.

Hayati, S. A. (1983). "Robot Arm Geometric Link Parameter Estimation". IEEE Conf. Decision and Control.

Hempleman, H. V., Lockwood, A. P. M. (1978). "The Physiology of Diving in Man and Other Animals". The Institute of Biology's Studies in Biology. No. 99: pp. 33-57.

Hollerbach, J. (1988). "A survey of kinematic calibration" Robotics Review. MIT press.

Irgens, F. (1999). "Fasthetslære" 6 utg. Tapir Forlag. Trondheim 1999.

ISO 1101. (1983). "Geometrical tolerancing".

ISO 9283. (1998). "Manipulating Industrial robots, Performance criteria and related test methods".

Kallevik, V., Hendseth, S. (1991). "A Teleoperated Trajectory Generation System for 3D Welding Curves on Subsea Structures". Proceedings of ROV 91.

Kandebo, S.W. (1997). "AW&ST and AUVSI 1997-98 international guide to unmanned vehicles". New York: Mcgraw-Hill, Inc.

Keith, R.J. (1997). "The requirement for mid-size ROVs and their place in the industry". Underwater Intervention 1997. Houston, Texas, USA.

Koch, W.H. (1994). "Creative Engineering work via applied nonlinear optimisation" Lecture Notes 94-004 PT. Narvik Institute of Technology.

Koch, W.H.: Adaptiv genaue LEVITIN/POLJAK-Typ-Verfahren zur Behandlung linear restringierter Optimierungsaufgaben ohne Verwendung von Ableitungen. Fakultät Mathematik und Naturwissenschaften der TU Dresden, 1983.

- Kress, R. L. (2002). "Teleoperation and Telerobotics", <http://www.engr.utk.edu/maes/ff/rlk/ieec/>
- Lane, D. M., Dunnigan, M. W., Knightbridge, P. J., and Quinn, A. W. (1991). "Planning and Control for Co-ordination of Underwater Manipulators". Proc. IEE Control 91, Edinburgh. Vol. 1: pp. 493-498.
- Langrock, D., Broome, D. R. (1994). "Development of a Telerobotic Underwater Manipulator". INSIGHT 36(9).
- Lien, T. K. (1980). "Banestyring for universelle handteringsautomater" Doctoral dissertation. NTH A8002.
- Lien, T. K. (1989). "Efficient inverse transformations for robot arms". SINTEF report; STF17 A89015.Trondheim.
- Lien, T. K. (1993). "Industrirobotteknikk". Tapir Forlag. ISBN 82-519-1134-6.
- Lien, T. K., Aune, D., Jenssen, H. P. (1991). "High Performance Robot for Subsea Machining Tasks". Proceedings of ROV 91.
- Lien, T. K., Thomessen, T. (1992). "Force feedback controlled grinding in underwater operations". Proceedings of Intervention/ROV'92.
- Matlab®. Software for technical computing. The Mathworks Inc.
- Marine Technology Directorate Ltd. (1992). "TUUV Technology for Unmanned Underwater Vehicles: Technical Brief", London UK: MTD Ltd.
- Mavroidis, C., Dubowsky, S., Drouet, P., Hintersteiner, J., Flanz, J. (1997). "A Systematic Error Analysis of Robotic Manipulators: Application to a High Performance Medical Robot". Proceedings of the 1997 IEEE International Conference on Robotics and Automation.
- Mavroidis, C., Flanz, J., Dubowsky, S., Drouet, P., Goitein, M. (1998). "High Performance Medical Robot Requirements and Accuracy Analysis". Robotics and Computer-Integrated Manufacturing. Elsevier Science Ltd.
- McKerrow, P. J. (1991). "Introduction to Robotics". Addison-Wesley Publishers Ltd. ISBN 0-201-18240-8.
- Mooring, B. W., Roth, Z. S. Driels, M. R. (1991). "Fundamentals of Manipulator Calibration". John Willy & Sons, Inc., ISBN 0-471-50864-0.

Nicolodi, S., Visentin, A., Rotella, A. (1990). "Supervisory Controlled Telemanipulation with Automatic Compensation of Manipulator base Movements". Proceedings of ROV 90.

Nof, S. Y. (1999). "Handbook of Industrial Robotics". John Wiley & Sons, Inc. ISBN 0-471-17783-0.

NOSYS®. Non-linear optimisation system software.

Raine, G.A., Lugg, M.C. (1995). "ROV inspection of welds – a reality", Underwater Intervention 1995, Houston, Texas, USA.

Raine, G. A., Forli, O. (1998). "A new guidance document on underwater non-destructive testing". Underwater Intervention 1998. New Orleans, Louisiana, USA.

Regen, B. (1991). "ATLAS 8F bilateral force feedback remote manipulator system". Proceedings ROV 91.

Ricci, F., Ellingsen, P. B. (1992). "The REMO project". Intervention ROV 1992. San Diego, California, USA.

Rivin, E. I. (1988). "Mechanical Design of Robots, McGraw-Hill Co.". ISBN 0-07-052992-2.

Sciavicco, L., Siciliano, B. (1996). "Modeling and Control of Robot Manipulators", McGraw-Hill.

Slocum, A. H. (1992). "Precision Machine Design". Englewood Cliffs.

Solvang, B., Lien, T.K., Thomessen, T. (1999). "A high precision underwater manipulator". 30<sup>th</sup> International Symposium on Robotics. Tokyo, Japan.

Solvang, B., Deng, Z., Lien, T. K. (2001). "A Methodological Framework for Developing ROV-manipulator Systems for Underwater Unmanned Intervention". Proceedings of Ocean 2001 MTS/IEEE Conference. ISBN: 0-933957-29-7.

Solvang, B., Deng, Z., Lien, T. K. (2001). "Structure of Underwater Intervention Manipulators in Workspace Constraint Environment and Architecture of their Control System". Proceedings of Ocean 2001 MTS/IEEE Conference. ISBN: 0-933957-29-7.

Taylor, A. J. (1993). "Design Issues for Underwater Manipulator Systems". *Mechatronics*. 3(4): pp. 419-432. Pergamon Press Ltd.

Thomessen, T. (1992). "Robot grinding strategies based on contact force measurements". Doctoral Dissertation. NTH 1992:64. ISBN 82-7119-418-6.

- Ulrich, N., Yoerger, D. R. (1991). "Design Optimization of an Electric Underwater Manipulator". Proceedings of ROV 91.
- Valavanis, K.P., Gracanin, D., Matijasevic, M., Kolluru, R., Demetriou, G.A. (1997). "Control architecture for autonomous underwater vehicles". IEEE Control Systems. 0272-1708/97, December 1997.
- Vartdal, K. (1990). "The diving assistance vehicle DIVEROV", Intervention ROV 1990. Vancouver, British Columbia, Canada.
- Veitschegge, W. K., Wu, C. (1986). "Robot Accuracy Analysis Based on Kinematics". IEEE Journal of Robotics and Automation. RA-2(3).
- Vestgård, K., Johansen, K. A., Klepaker, R. A., Størkersen, N. (2000). "Surveying and Inspection with the HUGIN Unthetered Underwater Vehicle (UUV/AUV)". Proceedings of UI 2000.
- Vukobratovic, M., Borovac, B. (1995). "Accuracy of the Robot Positioning and Orientation Assessed via its Manufacturing Tolerances". Mech. Mach. Theory. 30(1) Elsevier Science Ltd.
- Whitney, D.E., Lozinski, C.A., Rourke, J.M.(1986). "Industrial robot forward calibration method and results". Transactions of ASME. Journal of Dynamic Systems.
- Yoerger, D. R., Schempf, H., Dipietro, D. M. (1991). "Design and Performance Evaluation of an Actively Compliant Underwater Manipulator for Full-Ocean Depth". Journal of Robotic Systems. 8(3).
- Yuh, J. (2000). "Design and Control of Autonomous Underwater Robots: A Survey". Autonomous Robots 8: pp.7-24. Kluwer Academic Publishers.