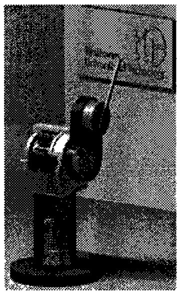
RRR-robot : design of an industrial-like test facility for nonlinear robot control



RRR-robot:

Design of an industrial-like test facility for nonlinear robot control

ir. A.M. van Beek

Supervisors: Prof. dr. ir. J.J. Kok
dr. ir. A.G. de Jager
ir. L. Kodde
ir. J.P.A. Banens

Eindhoven University of Technology (TUE)
Faculty of Mechanical Engineering
Division of Fundamental Mechanical Engineering
Section Systems & Control
May 1998
WFW rapport 98.014

ISBN-GEGEVENS

A.M. van Beek
'RRR-robot: Design of an industrial-like test facility
for nonlinear robot control' / Beek, A.M. van;
- Eindhoven: Stan Ackermans Instituut. -Ill.
Ontwerpers opl. Computational Mechanics
Met lit. opg.
ISBN 90-5282-858-X
Trefw.: Robot-design / Robot control /

Summary

Improving performance In industrial robots, several (non)linear effects may reduce the performance: configuration dependent inertia and conservative terms, Coriolis and centrifugal torques, friction, flexibilities in joints and/or links, backlash, motor dynamics, and sensor dynamics. By counteracting these unwanted effects, manipulators can become faster and/or more accurate. Various control schemes have been developed to deal with one or more of the nonlinear effects. For high-speed tracking of complex trajectories, the Coriolis and centrifugal torques form an essential part of the occurring nonlinear effects. Therefore, these terms play an important role in many of the proposed control strategies.

Experimental validation The RRR-robot (3×Rotation) has been designed and build to facilitate further development, i.e., to test and compare these concepts in an experimental setup with a complexity comparable to industrial manipulators with multiple degrees of freedom. To achieve a resemblance with conventional industrial robots (e.g., Puma-type robots), the system has a chain structure and three rotational degrees of freedom.

RRR-robot The main feature of this experimental facility is the ability to enhance the Coriolis and centrifugal torques relative to other nonlinear effects. This was achieved by eliminating two limitations of conventional robots. First, sliprings were applied to realize unconstrained rotation of all joints. Thus enabling a combination of low accelerations (low inertia), and large velocities (large Coriolis and centrifugal torques). Second, direct-drive servos (without transmissions) were used to reduce the influence of motor inertia originating from gear boxes with high reduction ratios, and to eliminate transmissions as a source of additional friction, backlash, and elasticity.

Another feature of the RRR-robot is a modular setup, i.e., the last link can be replaced by a flexible link, and the last joint can be transformed into a flexible joint. This makes it suitable to evaluate control schemes for both rigid-robots and flexible joint and/or link robots.

To control the robot, and to evaluate its performance, a PC based control system was selected. This system facilitates "rapid prototyping". Control algorithms can be implemented by means of graphical block scheme manipulations where the resulting diagram is used to generate real-time code.

Contents

Summary **iii**

1 **Introduction** **1**

2 **Design specifications** **3**
- 2.1 Problem formulation 3
- 2.2 Problem analysis 3
- 2.3 RRR-robot specifications 6

3 **RRR-robot subsystems** **9**
- 3.1 Subsystems 9
  - 3.1.1 Interactions 10
  - 3.1.2 Design strategy 10
- 3.2 Joint-Actuation System 11
  - 3.2.1 Transmissions 11
  - 3.2.2 Servo motors 11
  - 3.2.3 Servo sizing 14
- 3.3 Measurement System 17
  - 3.3.1 Joint-angle sensors 17
  - 3.3.2 Link sensors 19
  - 3.3.3 Cartesian position end-effector 20
- 3.4 Control System 22
  - 3.4.1 Control implementation 22
  - 3.4.2 Safety provisions 26
- 3.5 Signal-Transfer System 27

4 **Synthesis of subsystems** **33**
- 4.1 Manipulator frame 33
  - 4.1.1 Motor load 33
  - 4.1.2 Sliprings 37
  - 4.1.3 Eigenfrequencies 37
- 4.2 Measurement System: Cartesian measurement system 37
- 4.3 Evaluation 38
  - 4.3.1 Real-time performance 38
  - 4.3.2 Servo performance 40
  - 4.3.3 Design specifications 42
  - 4.3.4 Design goals 43

5 Conclusions and recommendations **45**
5.1 Conclusions 45
5.2 Recommendations 46

A Modeling the RRR-robot **47**
A.1 Introduction 47
A.2 Rigid manipulator model 47
A.2.1 Direct kinematics 47
A.2.2 Dynamics 49
A.2.3 Inertia parameters 51
A.3 Flexible model 51
A.3.1 Flexible-arm kinematics 52
A.4 Implementation 52

B RRR-robot components **55**
B.1 Dynaserv servos 55
B.2 MultiQ plug-in and terminal board 56
B.3 Sliprings 57
B.3.1 Power sliprings 57
B.3.2 Signal sliprings 58

C Budget (in Dutch) **59**

D Time schedule **61**

E Terminology **63**

Bibliography **67**

Chapter 1

Introduction

Industrial robots Robots[1] are applied in a great variety of fields, of which industrial automated manufacturing is the most important one. Objectives such as reducing the manufacturing costs, increasing the productivity, and improving (or maintaining) the product quality standards, represent the main factors that have promoted an increasing use of robotics technology. Nowadays, typical robot applications include: materials handling (e.g., palletizing or packaging), manipulation (e.g., arc welding or spray painting), and measurement (e.g., object inspection).

Not surprisingly, there is an increasing demand from industry for systems which can achieve these tasks faster and/or more accurately. If the increase of performance can be achieved at the same price, both improvements will result in lower price per product since increased accuracy may result in a reduction of post processing steps, and the increase of velocity obviously increases throughput.

In order to achieve fast response times in combination with an acceptable effort (i.e., driving torque[2]), industrial manipulators should be lightweight constructions. However, lightweight robot arms are also flexible. Especially at high velocities, deformations and vibrations can occur in both the joints and the manipulator's links. These unwanted dynamics seriously reduce a robot's accuracy. In applications which require precise tracking of a moving position reference, such as arc welding, laser cutting or gluing, this is a serious problem.

One way to avoid such unwanted dynamics is realizing a stiff construction, i.e., stiff joints and stiff arms. Designing both stiff and lightweight manipulators, though, is a difficult engineering task.

Stiff behavior by control An alternate approach is letting the construction "behave stiffly" by using advanced control strategies to deal with the flexibilities in manipulator joints and/or links. Unlike conventional control designs, these strategies do not assume stiff and rigid behavior but are based on models describing the flexible components. For fast industrial robots, velocity dependent torques — like Coriolis and centrifugal torques — are expected to play a decisive role in these models.

Due to rapid developments in computing hardware, the application of more complex and nonlinear control algorithms has become a viable option. However, implementations of control strategies based on flexible models are still rare in industrial robots [43].

[1] According to the definition of the Robot Institute of America, *a robot is a reprogrammable multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks.*

[2] The term "torque" is used in a general sense, and denotes both torques and forces.

Objective RRR-robot The main objective of the RRR-robot[3] design project, is to realize a manipulator-like system to test these advanced nonlinear control strategies. To achieve a resemblance with conventional industrial robots (e.g., Puma-type robots), the system should have a chain structure and at least three rotational degrees of freedom.

Although several other nonlinear effects (e.g., friction) may also reduce the performance, in this project the focus is on highlighting the Coriolis and centrifugal torques. Therefore, the main requirement of the robot is the ability to maintain these torques at a significant level.

The project is financed by the Eindhoven University of Technology (EUT) in the context of the participation in the Dutch Institute for Systems and Control (DISC). One of its goals is to tighten the bonds between the various research groups which constitute the research school DISC. Therefore, the project involves an array of input from Mechanical Engineering, Electrical Engineering, and Control Engineering.

Outline of this report In Chapter 2, the problem which the RRR-robot aims to solve is analyzed, and all relevant design specifications are summarized. Next, in Chapter 3, the robot is conceptually divided into four subsystem: a Joint-Actuation System, a Measurement System, a Control System, and a Signal-Transfer System. Each of these subsystems is discussed in detail, and working solutions, or even components are chosen. In Chapter 4, the chosen working principles are combined to form a complete system, and the design specifications are evaluated. Finally, in Chapter 5, some conclusions are presented and recommendations are made for future work.

In the Appendices, technical details and additional information are gathered. The used rigid-robot model is discussed in Appendix A. In Appendix B, the RRR-robot components and their main features are listed. The budget and time schedule regarding the design and construction are presented in Appendix C and D. In Appendix E, the used terminology is summarized.

Detailed information with regard to the use and assembly of the robot is presented in the RRR-robot instruction manual [41]. For a survey on 3D measurement systems for robot manipulators see [40].

[3] The triple R stands for three rotational degrees of freedom.

Chapter 2

Design specifications

The main problem in evaluating (non)linear control laws designed for high speed tracking is to separate Coriolis and centrifugal torques from other nonlinear effects. In this Chapter, it is explained why this evaluation requires a new experimental facility. By analyzing the problem and thus exploring the main requirement, two important features of the RRR-robot are identified. Finally, all the RRR-robot specifications are summarized.

2.1 Problem formulation

In any (industrial) manipulator, several (non)linear effects may reduce the performance: configuration dependent inertia and conservative terms, Coriolis and centrifugal torques, friction, flexibilities in joints and/or links, backlash, motor dynamics, and sensor dynamics. Various control schemes have been developed to deal with one or more of these effects.

For high-speed tracking of complex trajectories, the Coriolis and centrifugal torques form an essential part of the occurring nonlinear effects. Therefore, these terms play an important role in many of the proposed control strategies [12, 23, 24]. For further development it is necessary to test and compare these concepts in an experimental setup with a complexity comparable to industrial manipulators with multiple degrees of freedom [5].

Problem statement The main problem is to enhance the Coriolis and centrifugal torques relative to other linear and nonlinear effects. In an industrial manipulator these effects may obscure the influence of the velocity dependent torques. The objective of the RRR-robot is to solve this problem, i.e., to highlight the influence of Coriolis and centrifugal torques in an experimental facility with an industrial-like complexity.

2.2 Problem analysis

Maintaining the Coriolis and centrifugal torques at a significant level during a considerable period meets with two fundamental limitations in most existing manipulators:

- the rotation of each joint is constrained to a certain angle due to the cables used for energy and data transport throughout the manipulator;

- the presence of gears with high transmission reduction ratios (e.g. harmonic drives) tends to linearize the system dynamics. Furthermore, gear boxes are a source of additional friction, backlash and elasticity.

To illustrate these limitations, consider a simple rigid manipulator model[1] (rigid joints and links). Without flexibility, friction, backlash, motor dynamics or, sensor dynamics, the equations of motion are given by

$$M(\underline{q})\;\ddot{\underline{q}} + C(\underline{q},\dot{\underline{q}})\;\dot{\underline{q}} + \underline{g}\,(\underline{q}) = \underline{\tau} \tag{2.1}$$

where $\underline{q}$ denotes the vector with rotation angles, $\dot{\underline{q}}$ the angular velocities, and $\ddot{\underline{q}}$ the angular accelerations. So the torques $\underline{\tau}$ acting on each link consist of inertia terms, $M(\underline{q})\;\ddot{\underline{q}}$, Coriolis and centrifugal torques, $C(\underline{q},\dot{\underline{q}})\;\dot{\underline{q}}$, and conservative torques due to gravity $\underline{g}\,(\underline{q})$.

Constrained rotations In order to highlight the Coriolis and centrifugal torques for any link i, the contribution $\{C(\underline{q},\dot{\underline{q}})\;\dot{\underline{q}}\}_i$ should be "substantial" during a "sufficiently" long period of time. To quantify this requirement, the following ratio is defined:

$$r = \frac{\int_0^\infty \mid \{C(\underline{q},\dot{\underline{q}})\;\dot{\underline{q}}\}_i \mid d\xi}{\int_0^\infty \mid \{M(\underline{q})\;\ddot{\underline{q}}\}_i + \{C(\underline{q},\dot{\underline{q}})\;\dot{\underline{q}}\}_i + \{\underline{g}\,(\underline{q})\}_i \mid d\xi}. \tag{2.2}$$

Highlighting the Coriolis and centrifugal torques now means achieving a large ratio. As can be seen in (2.1), this can be achieved by a combination of high velocities and low accelerations. If the joint rotation angles are limited, such a combination can be achieved only with considerable energy (using powerful, expensive servos) and only over a short period of time, since shortly after the acceleration to a certain velocity, the de-acceleration must begin in order to avoid breaking any cables. Furthermore, the large injection of energy will excite additional vibrational modes, thus clouding the contribution of the Coriolis and centrifugal torques.

Transmissions To understand the linearizing effect of transmissions, first, they must be included in the model. Assume the transmissions are rigid and without backlash. Let $\underline{q}_m$ and $\underline{\tau}_m$ denote the vectors of joint actuator displacements and actuator driving torques respectively; the transmissions then establish the following relationships:

$$\left\{ \begin{array}{l} \underline{q}_m = K_r\,\underline{q} \\ \underline{\tau}_m = K_r^{-1}\underline{\tau} \end{array} \right. \tag{2.3}$$

where K_r is a diagonal matrix containing the gear reduction ratios k_{r_i}. The central moments of inertia, I_{m_i}, of each motor with respect to its rotor axis are collected in the diagonal matrix I_m.

The *mass matrix* $M(\underline{q})$ can be split into a constant diagonal matrix, $K_r I_m K_r$ and a configuration-dependent (dependent on functions of $\underline{q}$) matrix $\Delta M(\underline{q})$, i.e.,

$$M(\underline{q}) = K_r I_m K_r + \Delta M(\underline{q}). \tag{2.4}$$

Substituting (2.3) and (2.4) into (2.1) yields

$$\underline{\tau}_m = I_m\,\ddot{\underline{q}}_m + \underline{d} \tag{2.5}$$

[1] In Appendix A, the derivation of models for rigid robots is considered in more detail.

where

$$\begin{aligned} \underline{d} &= K_r^{-1}\,\Delta M(\underline{q})K_r^{-1}\,\underline{\ddot{q}}_m + K_r^{-1}\,C(\underline{q},\underline{\dot{q}})\,K_r^{-1}\,\underline{\dot{q}}_m + K_r^{-1}\underline{g}\,(\underline{q}) \\ \underline{q} &= K_r^{-1}\underline{q}_m \end{aligned} \tag{2.6}$$

represents the contribution depending on the configuration. As illustrated by the block scheme of Fig. 2.1, the system described by (2.1) actually consists of two subsystems:

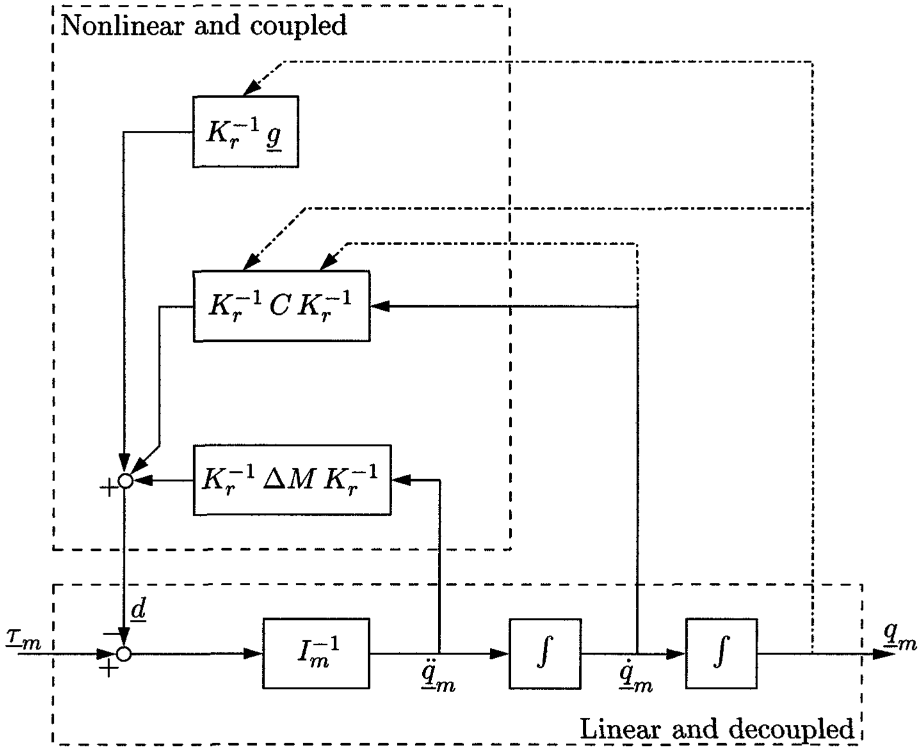


Figure 2.1: Block scheme of a rigid manipulator with transmissions.

- a linear and decoupled subsystem in which each component of $\underline{\tau}_m$ influences only the corresponding component of $\underline{q}_m$, and
- a nonlinear and coupled subsystem.

In case of high reduction gears ($k_{r_i} \gg 1$), the contribution of the nonlinear interaction term $\underline{d}$ is small, and can be considered as a disturbance for each joint servo. As a result the ratio of (2.2) becomes small, since the torque τ_i is almost completely determined by the motor inertia.

The price to pay, however, is the occurrence of joint friction, elasticity, and backlash that may limit system performance more than the nonlinear terms ($\underline{d}$ in (2.6)). The use of drives without transmissions, *direct-drives* ($K_r = I$), could eliminate these drawbacks, but the influence of nonlinearities and coupling between the joints then becomes relevant. Aside from the cost and size of this actuator type this is the main reason that these servos are not yet very popular in industrial manipulators [34].

Conclusion Analyzing the problem of highlighting the Coriolis and centrifugal torques, two limitations of conventional manipulators were identified:

- the joint rotations are constrained, and
- the transmissions have high reduction rates.

In the RRR-robot, these limitations should be overcome, leading to:

- the removal of constraints on the rotation of each joint, and
- the use of transmissions with small, unity (direct-drive servos) or even inverse reduction rates ($k_{r_i} < 1$).

2.3 RRR-robot specifications

In the funding application [42] several specifications are mentioned throughout the text. Based on this application and the preceding section, a complete list of specifications was made.

The list is divided into three categories: design requirements (quantitative specifications), wishes (qualitative specifications), and boundary conditions (determined by the environment).

Design requirements

The RRR-robot must have:

(1) the ability to maintain the Coriolis and centrifugal torques on at least one link i sufficiently large:

$$\frac{\int_0^\infty \mid \{C(\underline{q},\underline{\dot{q}})\,\underline{\dot{q}}\}_i \mid d\xi}{\int_0^\infty \mid \{M(\underline{q})\,\underline{\ddot{q}}\}_i + \{C(\underline{q},\underline{\dot{q}})\,\underline{\dot{q}}\}_i + \{\underline{g}\,(\underline{q})\}_i \mid d\xi} \geq 0.5$$

(2) the ability to control the robot, using command signals proportional to the actuator torques;

(3) three or more rotational degrees of freedom;

(4) a static end-effector accuracy for each Cartesian coordinate of 0.1 [mm];

(5) an end-effector tracking accuracy of 0.2 [mm] for reference inputs up to 5 [Hz], while a gradual degradation down to 1 [mm] at 25 [Hz] inputs is allowed;

(6) a third link which is exchangeable with a flexible one;

(7) a third joint which is expandable to a flexible drive;

(8) a control interface integrated with the control development environment (Matlab/Simulink);

(9) a characteristic dimension of 0.5 [m];

(10) a technical lifespan of at least 10 year;

(11) a total realization cost less than fl 155.000,-.

In addition, the system should be:

(12) suitable for demonstration for a group of at least 10 people in a lab environment;

(13) safe for operators and spectators.

Compared to the funding application there are two significant omissions:

- maximal angular velocities of $1 \times 2\pi$, $5 \times 2\pi$ and $25 \times 2\pi$ (rad/s) for the first, second and third link respectively;
- maximal accelerations based on reaching the maximal velocities within one revolution or within 0.5 (s), whichever results in the largest acceleration.

Considering the objective, these requirements can be replaced by the more relevant requirement (1), regarding torques instead of velocities and accelerations.

Design goals

If possible, the system should fulfill the following goals (in descending importance):

(a) modern, state-of-the-art techniques should be applied;

(b) future expansion (e.g. more sensors) should be possible;

(c) a recognizable relation with industrial manipulators should exist;

(d) maintenance should be easy;

(e) insight into the robot's working principles should exist.

Boundary conditions

Finally, the RRR-robot project is subject to certain boundary conditions with respect to its environment and users:

- the system will be operating in a laboratory environment;
- within the research school DISC, the RRR-robot is to be used as a research object for several research groups.

Chapter 3

RRR-robot subsystems

In this chapter, the RRR-robot is conceptually divided into *subsystems* for *joint actuation*, *measurement*, *control*, and *signal transfer.* Each subsystem is discussed, and appropriate components or design solutions are chosen.

3.1 Subsystems

To tackle the design problem, a choice is made to conceptually divide the RRR-robot into several subsystem based on their function. To a certain extent, these subsystems can be considered independently. In the following, all subsystems, and their interactions are defined.

- The Joint-Actuation System consists of
  - transmissions, and
  - servo motors with appropriate power supplies and power amplifiers.
- The Measurement System consists of
  - joint sensors (e.g., to measure joint-angles),
  - link sensors (e.g., strain gauges), and
  - a 3D measurement system to determine the Cartesian position of the end-effector independently from the joint sensors.
- The Control System consists of
  - hardware components for data acquisition, command output, and overload protection;
  - software components to implement the control algorithm, and user interface; and
  - a hardware platform to execute the software.

  Connections between these hardware components are also viewed as part of the Control System.
- The Signal-Transfer System consists of components for the transport of
  - power, and
  - measurement signals, across parts which move relative to each other.

This includes transport between power amplifier and servo motor; between Measurement System and Control System; and between Control System and power amplifier. In some cases the Signal-Transfer System is reduced to a simple stationary connection (e.g., an electric cable, or hydraulic duct).

3.1.1 Interactions

In Fig. 3.1, a possible functional lay-out of the RRR-robot is shown.

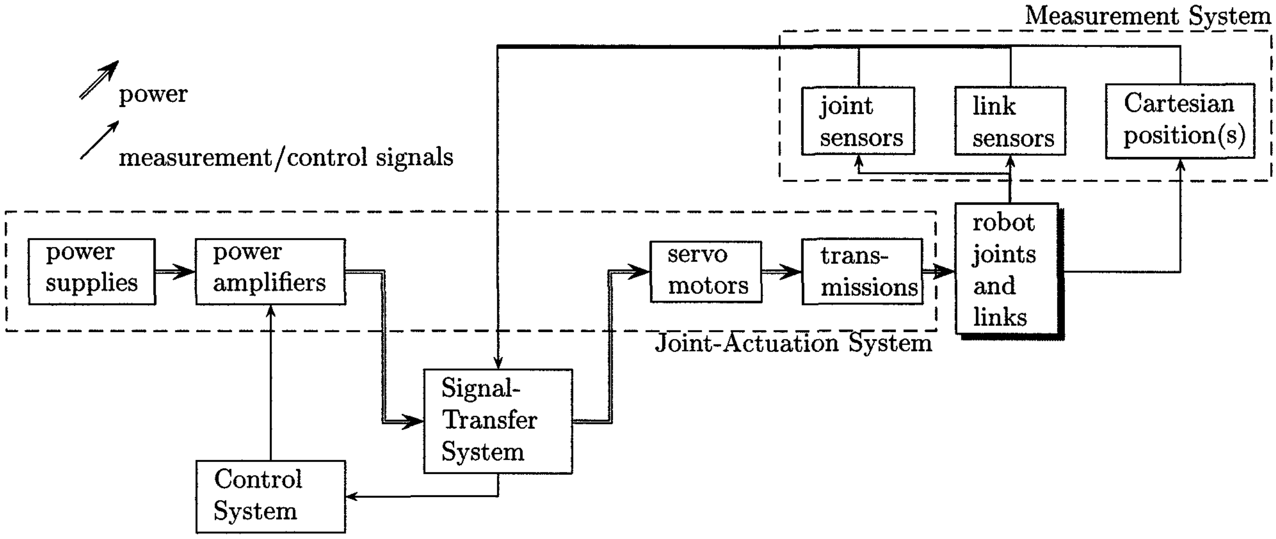


Figure 3.1: A possible functional lay-out of the RRR-robot's subsystems, and their interaction. Note that, especially the position of the Signal-Transfer System relative to the Joint-Actuation System may vary (e.g., depending on the type and placement of the power amplifier).

The choice of the Joint-Actuation System determines the type of power transported by the Signal-Transfer System, e.g., electrical, hydraulic or pneumatic. The Measurement System, together with the Joint-Actuation System, completely determines the requirements of the Signal-Transfer System, i.e., the amount and type of transport channels. The Control System mainly interacts with the Measurement System. It must be able to process measurements from all sensors. The interaction with the servos is restricted to three (usually electrical) command signals.

3.1.2 Design strategy

In designing the RRR-robot, each subsystem is considered separately, and an initial design solution or component is chosen. Because of the interactions, the order in which the subsystems are discussed is not unique.

The chosen starting point is the Joint-Actuation System since this determines the performance of the robot in terms of the design requirement (1) (Section 2.3).

Next the Measurement System and the Control System are addressed. These subsystems are mainly interdependent. Any additional link sensors, but also the components of the 3D measurement system which are attached to the end-effector can be considered as black boxes with certain mass and (electrical) interface to the Signal-Transfer System.

Finally, the Signal-Transfer System is discussed based on the transport specifications imposed by the Joint-Actuation System and the sensors of the Measurement System. Obviously, this is an iterative procedure. If in any stage no satisfactory design solution can be found, it is necessary to return one or more steps.

3.2 Joint-Actuation System

In the following sections, the components of the Joint-Actuation System are discussed. A choice is made to use brushless DC motors as direct-drive servos (without transmissions). Finally, a rigid-robot model is used to aid servo sizing.

3.2.1 Transmissions

Since servo motors typically provide high speeds with low torques, transmissions are often necessary to transform these quantities into the required opposite combination: low speeds with high torques. On the other hand, transmissions are also an important (additional) source of joint friction, elasticity, and backlash.

To eliminate these drawbacks, attempts have been made to develop actuation systems which allow direct connection of the motor to the joint without the use of any transmission element. The use of such *direct-drive* actuation systems is not yet popular for industrial manipulators, mainly because of the control complexity; due to the absence of reduction gears, the nonlinear terms in the dynamic model can no longer be neglected (see section 2.1).

For the RRR-robot, however, this complexity is exactly what we want. Furthermore, the application of direct-drives also complies with the first design goal: modern state-of-the-art techniques (design goal (a), Section 2.3)

Although the control complexity could be increased even more by using transmissions with inverse reduction rates ($k_{r_i} < 1$), such "inverse transmissions" present several problems and disadvantages. First of all there are technical problems finding appropriate, powerful enough, actuators to drive transmissions which *reduce* the output torque. Secondly, those transmission have all the disadvantages of "regular" transmissions: additional friction, elasticity and backlash (or, if not, are rather expensive). Finally, with inverse reduction rates the relation with *future* industrial manipulators (design goal (c)) is lost.

3.2.2 Servo motors

Motor types used for the actuation of joint motions can be classified into three groups:

1. Pneumatic motors,
2. Hydraulic motors, and
3. Electrical motors.

Of this groups the following servos are typically used for direct-drive applications:

- hydraulic motors, and
- electrical motors:

- Variable Reluctance Stepper (VRS) motors,
- Direct-Current (DC) motors, and
- Brushless Direct-Current (BDC) motors.

Pneumatic motors are not suitable for applications where continuous motion control is of concern due to their unavoidable air compressibility errors. They are mainly used in simple pick-and-place applications and for opening and closing motions of the jaws in a gripper tool.

Asynchronous induction motors are typically suited to deliver high power (> 1 kW). Due to the control complexity, they are not yet popular for servos.

Hydraulic motors Compared to electrical motors, hydraulic motors have several advantages. They:

- can achieve much higher torques at low velocities,
- have an excellent power-to-weight ratio, and
- are self-lubricated and self-cooled by the circulating fluid.

However, they also present the following drawbacks:

- need for a hydraulic power station,
- temperature dependent dynamics,
- need for regular maintenance, and
- pollution of the working environment due to oil leakage.

The last drawback is a significant problem, especially in combination with the specified unconstrained rotation (from requirement (1)) in each joint. Since the hydraulic lines (conducting the power signals) must pass through a signal transfer system with relative motion, leakage is difficult to avoid.

Mainly because of this last problem, a choice was made to use electrical motors.

Variable-Reluctance Stepper motors The VRS motor is an induction type stepping motor. Both stator and rotor consist of laminated steel with a high number of opposing teeth; only the stator has (phase) windings. Torque is generated from the variation of the air gap due to the difference in teeth pitch on the rotor and the stator. The (electrical) dynamics (see, e.g., [44]) are given by

$$u = Ri + L(g)\frac{di}{dt} + i\frac{dL(g)}{d\theta}\dot{\theta} \tag{3.1}$$

where u is the voltage applied to the windings in the stator, R and L are the resistance and the inductance of the windings respectively, i is the current, and $g(\theta)$ is the width of the air gap. The exerted torque can be derived from the magnetic co-energy $W_m' = \frac{1}{2}Li^2$,

$$T = \frac{\partial W_m'}{\partial \theta} = \frac{1}{2}i^2\frac{dL(g)}{d\theta}. \tag{3.2}$$

These actuators are controlled by suitable excitation sequences, and their operation principle does not require measurement of the motor-shaft angle. Together with the absence of permanent magnets, this results in simple and cheap actuators. However, the stepping nature of this motor type restricts the position accuracy, causes vibrations, and introduces a significant torque ripple. Such inconveniences confine the use of stepper motors to applications, where low-cost implementation prevails over the need for accuracy and high dynamic performance.

With the use of a controller (and motor-shaft position feedback), the mentioned problems can be partly solved; torque control however, is difficult to realize.

Direct-Current motors Like the regular DC motor, the direct drive version consists of a stator[1] which generates a permanent magnetic field (by ferromagnetic ceramics or rare earths), and a winded rotor which is powered via brushes and commutators.

The design goal of a direct-drive DC motor is to maximize output torque rather than power. The torque T is the product of the torque constant K_t and the rotor current i;

$$T = K_t\, i. \tag{3.3}$$

Since the torque constant is determined by the motor inductance L ($K_t \sim L$), two important features of direct-drive DC motors emerge [2]:

- a bulky or sometimes "pancake" like appearance because a large inductance requires a considerable motor volume (and mass): $L \sim l_r d_r^2$ where l_r and d_r are the rotor length and the rotor diameter respectively;
- a considerable electric time constant $\tau_e = \frac{L}{R}$ which is often no longer negligible.

Maximizing the current meets with an important limitation of the DC motor in general: mechanical commutation. At the brush and commutator, large sparks can occur especially in combination with a large inductance. These sparks cause the brush to wear quickly and cause unwanted noise. Furthermore, the brush mechanism increases mechanical friction.

Note that the torque of (3.3) is not constant due to the effect of the discrete commutation. The (maximum) deviation from the average torque is called the torque ripple.

Brushless Direct-Current torque motors In a brushless DC motor, mechanical commutation has been replaced with electronic commutation. Unlike conventional DC motors, the rotor consists of permanent magnets, while the stator consists of several phase windings. The commutation of currents is accomplished by measuring the rotor position using a position sensor.

The elimination of brush and commutator allows an improvement of motor performance in terms of higher torques and less material wear, and is therefore the main reason for using BDC motors.

The inversion between the functions of stator and rotor leads to further advantages. The presence of a winding on the stator instead of the rotor, facilitates heat disposal. The absence of a rotor winding allows construction of more compact rotors, which in turn have a lower moment of inertia. As a result, the size of a BDC motor is smaller than that of a regular DC motor of the same power.

The superiority of the brushless motor, though, comes at a cost: the rotor position must be detected or measured, and fed back to the power amplifier (typically a DC-to-AC converter). However, if such an internal sensor is accurate enough for robot control purposes, no additional joint position sensors are necessary, and the construction becomes more compact.

[1] Brushed DC motors come in a wide range of configurations, this configuration (permanent magnet stator and winded rotor) is most common in direct drive applications [1].

Conclusion Considering the obvious advantages of BDC motors, a choice is made to use brushless direct-drive motors for all joints. Preferably, the internal sensors should be used for the robot control.

3.2.3 Servo sizing

Choosing appropriate servos for the RRR-robot is an iterative problem, since, even for brushless direct-drive servos, the torque-to-mass ratio is small (see Fig. 3.2). So, as a result of the required torques, each motor has also a significant mass. To accelerate these masses, even larger motors may be necessary.

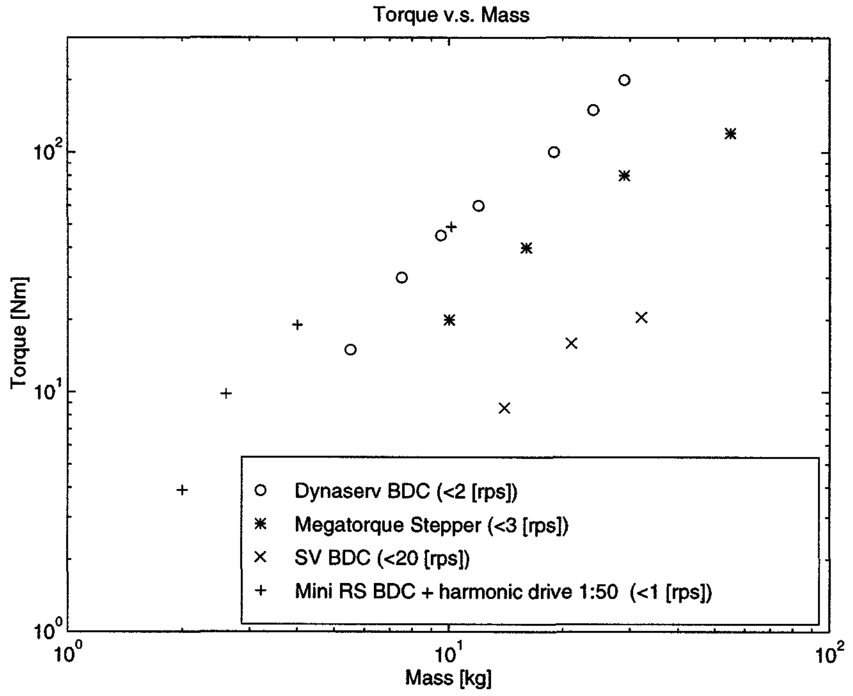


Figure 3.2: Maximum torque versus mass for several commercial servo series: the Dynaserv brushless DC motors [26], the Megatorque stepper motors [29], the SV brushless DC motors [9], and the mini-RS BDC motors with integrated harmonic drive [17]. The first two are specifically designed for direct-drive applications.

To aid the motor choice, a rigid-manipulator model of the RRR-robot is used. The derivation of this model is described in Appendix A. The aim is to find three motors with the right combination of mass, torque and angular velocity; at least one link should have a performance ratio r of 50 % or more at the largest possible angular velocity.

Based on an initial design sketch, a rough estimate was made for the link parameters (i.e., dimensions and material) leading to the main system parameters shown in the upper part of Table 3.1. Because of its low mass density, aluminum is chosen as construction material. To illustrate this initial design, the kinematic relations (section A.2.1) were used to draw the simplified wire-frame picture of Fig. 3.3.

Table 3.1: Main system parameters used in evaluation model. The links are assumed to be hollow rectangular aluminum beams. The motor data are taken from the Dynaserv series of Litton Precision Products.

parameters link/joint		1	2	3
link length	[m]	0.5	0.2	0.3
link mass	[kg]	9.86	1.21	0.73
motor mass	[kg]	12	7.5	5.5
motor inertia	[kg m^2]	0.023	0.015	0.012
max. torque	[Nm]	60	30	15

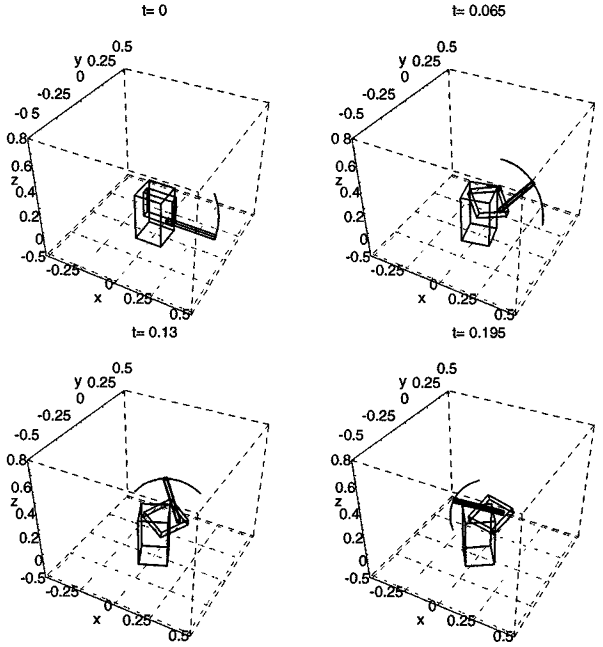


Figure 3.3: A wire-frame "animation" of the initial design. All quantities are in SI-units: t in [s], and x, y, z in [m].

After choosing an appropriate motion pattern, i.e., with a high ratio r, a commercial motor can be evaluated by substituting its mass and rotor inertia.

Selection

With the Dynaserv series of BDC motors of Litton Precision Products (see the motor data in the lower part of Table 3.1) the main design requirement, maintaining high Coriolis and centrifugal forces, was met. This is illustrated in Fig. 3.4 with on the left the torques required to realize a stationary motion pattern (without inertia effects because $\underline{\ddot{q}}_i = 0$, so with large ratios r_i) $\underline{q}_{\text{stat}} = [\pi\, t, 2\pi\, t, 3\pi\, t]^T$ [rad] ; on the right are the torques required to realize a motion pattern with accelerations: $\underline{q}_{\text{dyn}} = [\sin \pi/2\, t, \sin 3\pi/2\, t, \sin 3\pi\, t]^T$ [rad]. Both motion patterns are chosen to achieve the maximum torque of one or more motors.

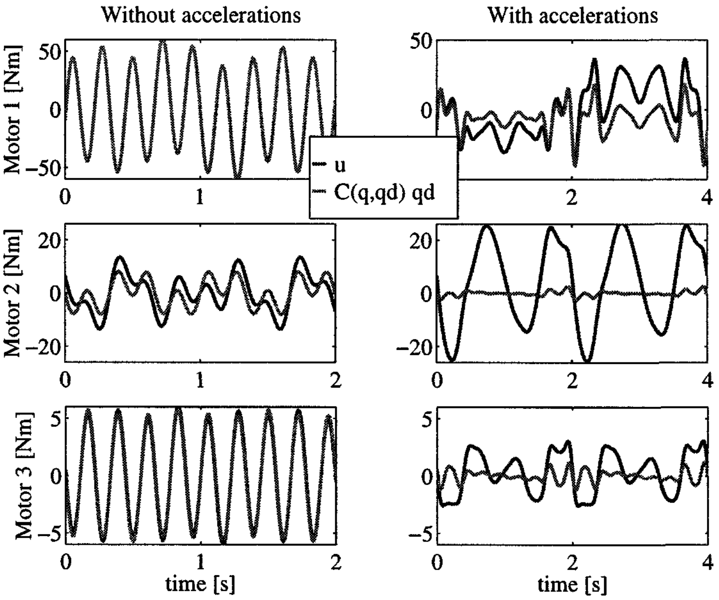


Figure 3.4: Required motor torques (thick black lines) to realize a chosen motion pattern without accelerations (on the left) and with accelerations (on the right). The contribution of the Coriolis and centrifugal torques is shown with a thick grey line.

Evaluation of (2.2) for each link yields $r = [1, 0.39, 1]$ for $\underline{q}_{\text{stat}}$ and $r = [0.51, 0.03, 0.20]$ for $\underline{q}_{\text{dyn}}$. So for both motion patterns, the Coriolis and centrifugal component of motor 1 is above the required 50% of the total torque.

In addition, these servos can be used as complete self-contained joints with internal bearings and sensors. Thus simplifying the design of the manipulator. In Section B.1, the main features and specifications are summarized.

Risk analysis

In their function of compact joint-units, the use of the selected Dynaserv servos involves a number of potential risks; damage to the servos may occur due to thermal or mechanical overload.

- Thermal overload: due to heat development the allowed motor temperature (45 °C) is exceeded.

  Action: correspondence with the supplier guaranteeing that heat development in closed environments is not a problem (Litton 4-11-96).

- Mechanical overload: the allowed static or dynamic moment load (see Table B.1) may be exceeded.

  Action: designing the manipulator frame based on minimizing the moment loads on the servos (see Chapter 4).

3.3 Measurement System

In this Section, sensors are chosen for the measurement of the joint angles, and the Cartesian position of the end-effector. Additional link sensors will be realized later, so they are discussed only briefly with the emphasis on their possible interaction with the manipulator design (e.g., mass) and other subsystems, such as the Signal-Transfer System and the Control System (e.g., data transport and conditioning).

3.3.1 Joint-angle sensors

First the specifications for the static and tracking accuracy of the end-effector are translated into joint-angle specifications for each joint. After discussing two types of suitable sensors, a choice is made to use incremental encoders built-in with the motors.

Specifications

Assuming that the end-effector position is determined only by the joint angles $\underline{q}$ (see [36]), the sensitivity of the end-effector to variations Δq_i is maximal when link 1 and 2 are perpendicular and link 2 and 3 are aligned (see Fig. 3.5).

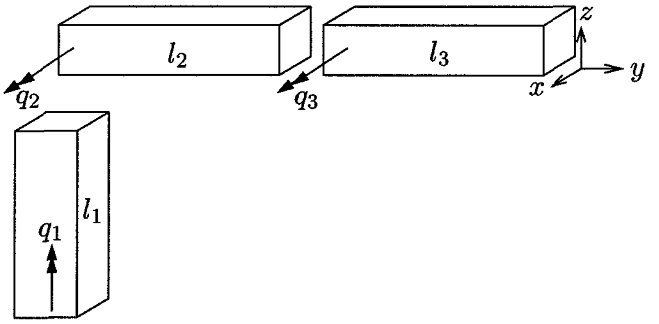


Figure 3.5: Sensor specifications

In this configuration, the angles Δq_2 and Δq_3 both influence the position error in the z-direction. To simplify the design by using the same sensor for joint 2 and 3, both errors are chosen equal, i.e., $\Delta q_2 = \Delta q_3$. Then the relation between the position error of the end-effector, $\underline{e} = [\Delta x, \Delta y, \Delta z]^T$, the link length l_i, and the joint angle deviation Δq_i is given by:

$$\begin{aligned} \Delta q_1 &\approx \frac{\Delta x}{l_2 + l_3} = \frac{\Delta y}{l_2 + l_3} \\ \Delta q_2 = \Delta q_3 &\approx \frac{\Delta z}{l_2 + 2l_3} \end{aligned} \tag{3.4}$$

With the specified static and tracking end-effector accuracy for each Cartesian coordinate (see Section 2.3) and a first choice of the link length l_i the joint-angle specifications are given in Table 3.2.

Table 3.2: Joint-angle specifications. The required bit resolution is determined by the static error. Note that, joint-angle sensor 2 and 3 have identical specifications because the allowable errors Δq_2 and Δq_3 (not their contributions to the end-effector error Δz) were chosen equal.

quantity			link 1	link 2	link 3
length l_i		[m]	0.5	0.2	0.3
static error	$\Delta q_i \leq$	$\times 10^{-3}$ [rad]	0.20	0.125	0.125
		[*arc-sec*]	41.3	25.8	25.8
tracking error	$\Delta q_i \leq$	$\times 10^{-3}$ [rad]	0.40–2.00	0.25–1.25	0.25–1.25
(5–25 [Hz])		[*arc-sec*]	82.5–412.5	51.6–257.8	51.6–257.8
required sensor resolution		[bit]	15	16	16

Sensor types

The most common transducers used in robot applications are resolvers (or synchros) and (absolute or incremental) encoders because of their precision, robustness, and reliability. Both types have operating principles which enable an accuracy within the specifications (up to 20 [bit] for encoders, and up to 16 [bit] for resolvers) to. Other sensors to measure angular displacement are either not accurate enough to meet the required joint specifications (e.g., potentiometers, up to 10 [bit] resolution), or have accuracies well above these specifications (e.g., inductosyn, 18 to 22 [bit]) [13].

Resolver The operating principle of the resolver is based on the mutual induction between two electric circuits which allow continuous transmission of angular position without mechanical limits. From a construction viewpoint, the resolver is a small electric machine with a stator and a rotor fed by a sinusoidal voltage. The information on the angular position is associated with the magnitude of the supply and induced sinusoidal voltages, which are treated by a suitable resolver-to-digital converter (RDC) to obtain the digital data corresponding to the position measurement. In the process also analog velocity measurements become available.

Encoder An encoder consists of an optical-glass disk on which concentric tracks are made with alternating transparent and opaque sectors. A light beam is emitted perpendicularly to the disk; for each track a photodiode is used to determine whether a sector is transparent or not. In an absolute encoder the pattern of sectors is unique for each position on the disk. The number of tracks determines the resolution of the encoder, e.g., a 12 [bit] absolute encoder requires 12 tracks.

Simpler from a construction viewpoint (and thus cheaper) is the incremental encoder which in its most basic form has only one track with equally distributed sectors. The absolute position is determined by means of suitable counting and storing circuits. Often a second track is added to determine the sign of the rotation and improve the resolution. To define an absolute mechanical zero as reference, sometimes a third track with one opaque sector is added. Note that in contrast with the absolute encoder, the position information is more sensitive to disturbances.

Velocity measurements can be reconstructed by using a voltage-to-frequency converter (with analog output), or by (digitally) measuring the frequency of the pulse train.

Selection

Although 15 or even 16 [bit] absolute encoders exist, they are rare and extremely expensive. So two options remain: the resolver and the incremental encoder. By construction the resolver type sensor with winded stator and rotor coils is heavier then the encoder with only an optical disk.

The chosen Dynaserv motor is available both with a resolver (DR-series: accuracy ±45 [*arc-sec*]), or with an incremental encoder (DM-series: accuracy ±15 [*arc-sec*]). Based on the joint-angle specifications (Table 3.2), a servo from the DM-series with an incremental encoder is selected for all three joints. Note that the DR-series servos also have a 30%–60% smaller torque-to-mass-ratio then the DM-series.

Risk analysis

The main risk using incremental encoders is the sensitivity for (electronic) disturbances. Every missed pulse reduces the position accuracy. See Section 3.5 for a more detailed discussion.

3.3.2 Link sensors

Several possible sensors to determine deflection or acceleration of the link are discussed with respect to their interaction with the rest of the system.

Strain-gauge

Strain-gauges can be used to measure the local strain of a link. The most used type is the bonded metal-foil gauge. The strain, $\frac{dL}{L}$ is determined from the relative change in resistance of the metal conductor, i.e.,

$$dL/L = \frac{dR/R}{\text{Gauge factor}} \tag{3.5}$$

The Gauge factor depends on the material properties of the strain-Gauge and can not be measured directly, but is determined by sample testing with a typical accuracy of ± 1 %.

To measure the deflection of a beam, a Wheatstone bridge with 4 strain-Gauges can be used. When using a battery for the supply voltage (0.5–5 [V] with a typical gauge current of 5–40 [mA]), two signal lines are needed for each deflection measurement. Since the output voltage is quite small (a few microvolts to a few millivolt) amplification is needed.

Any loading effect of the Signal-Transfer System can be counteracted by (regular) calibration provided the resistance of the supply lines changes only slowly in time. To improve the signal-to-noise ratio, the strain-gauge output should be amplified as early as possible.

Inertial sensors

Inertial measurement sensors such as gyroscopes and accelerometers measure rotation rate, and acceleration, respectively. The measurements can be integrated to obtain the change in orientation (yaw, pitch and roll) and double-integrated to determine the change in position (Δx,Δy, and Δz). In [40], the properties of several types of (commercial) accelerometers, and gyroscopes are addressed.

When used for tracking the position and/or orientation, inertial sensors have the advantage that they consist of self-contained, small, and light (< 0.5 [kg]) units. No receiver/transmitter set-up is required. The possible update rate is limited only by sensor bandwidth. Another advantage of these sensors is a basically unlimited range, independent of the tracking accuracy. However, integrating the data means that only relative orientation or position is measured; a known starting or reference point is required. Moreover, any constant error increases without bound after integration. Therefore, inertial sensors alone are generally inadequate for periods of time that exceed a few seconds. Often, they are used in conjunction with other technologies (so-called sensor fusion) to provide periodical updates of the absolute position. Then the inertial measurements can be used to increase the update rate of the entire system.

For the application in the RRR-robot several properties are important:

- *Frequency range*
  To determine the change in position and orientation, the transducer should be able to measure constant, DC, values. So the frequency range should start at 0 [Hz]. However, in sensor fusion setups this requirement can be relaxed if other technologies can be used to provide updates with a complementary frequency range (e.g., stationary updates).
- *Measurement range*
  Based on the initial design parameters and the nominal velocity of the servos, inertial sensors should be able to measure and/or withstand peak values of 6 [m/s], 150 [degrees/s] and 50 [g].
- *Signal conditioning/amplification*
  To avoid problems with the Signal-Transfer System the transducer output should be amplified or conditioned as early as possible.
- *Connections*
  The number of connections should be as low as possible.

3.3.3 Cartesian position end-effector

An independent measurement of the Cartesian position of the end-effector is required for two, different, applications:

1. Kinematic calibration, i.e., to estimate and compensate the deviations from the nominal construction parameters. The robot kinematic structure is often represented by DH (Denavit Hartenberg) parameters. Using four parameters per joint, the position and orientation of each joint with respect to the previous joint is described: link length, a, distance offset along the rotation axis, d, the joint axis orientation angle, α, and the joint rotation angle, q. With accurate (static) measurements, of the end-effector position in the Cartesian space,

$$[x, y, z]_{\text{end-effector}} = \mathcal{F}\left(a_1 \dots a_n,\ d_1 \dots d_n, \alpha_1 \dots \alpha_n, q_1 \dots q_n\right)$$

   with n joints/links, the deviation from the nominal parameters can be estimated using a least-squares technique [34]. To avoid an ill-conditioned estimation problem, it is advisable to choose enough ($\gg 4n$) measurements, distributed over the entire workspace.
2. Dynamic tracking of one or more points on an intentional elastic link and/or joint.

Ideally, the 3D measurement system for the RRR-robot should be suitable for both high-speed dynamic tracking, and static kinematic calibration. However, in all Cartesian measurement systems, a trade-off exists between accuracy and tracking velocity. Therefore, evaluation of the tracking performance – the main function of the RRR-robot – has priority.

Specifications

The design requirements of the Cartesian measurement system are determined by the specifications of the RRR-robot as a whole (section 2.3); the implementation of the Joint-Actuation System, the Control System, the Signal-Transfer System; and the design of a future flexible link and/or joint. The latter is not a priori known and can therefore be used to provide extra freedom in designing the Cartesian measurement system. So the design of the flexible components can be adapted to the limitations of a specific measurement system.

- *Accuracy*
  The end-effector tracking accuracy is defined in Cartesian coordinates :
  $$[x \pm \epsilon, y \pm \epsilon, z \pm \epsilon],$$
  with ϵ is 0.1 [mm] for static measurements, and 0.2 [mm] for reference inputs up to 5 [Hz], allowing for a gradual degradation down to 1 [mm] at 25 [Hz] inputs.
- *Tracking volume*
  The tracking volume is a sphere with a diameter of approximate 1 [m]. In this volume of 0.52 [m^3] the end-effector position should be measured without interruptions.
- *Update rate*
  To measure flexible modes up-to 500 [Hz] the update rate of the measurement system should be 1000 [Hz].
- *Dynamic bounds*
  Based on the nominal design and velocities the end-effector has a maximal linear tracking velocity of 6 [m/s] and a maximum acceleration of 50 [g].
- *Interface*
  The output of the measurement system, should be compatible with the chosen PC based control system.

Selection

In [40], a survey is presented of systems for independent measurement of the Cartesian 3D position of the end-effector of a robot manipulator.

The design specifications of the 3D measurement system for the RRR-robot, especially the combination of high accuracy, high update rate and uninterrupted tracking, severely restrict the possible options. Since the RRR-robot is designed to enable unconstrained joint rotation, it is inevitable that the line-of-sight from one single observer (camera or sensor) to the tip of the end-effector is lost during normal operation. Therefore, optical tracking systems and mechanical contact measurements can be ruled out.

Because acoustic system are not accurate enough, only one technology remains: vision based ranging. A combination of both a high accuracy and a high update rate can be achieved with measurement systems based on line-array cameras. By combining multiple viewpoints also the line-of-sight problem can be (partly) solved. In principle, also CCD Area cameras can be used, but only in combination with an inertial measurement system to improve the update rate. The inertial measurements also allow for a short (< 0.1 [s]) loss of the line(s) of sight.

Suitable commercial systems exist, but exceed the available budget by a factor 4 or more. Therefore, the remaining option to realize a 3D vision based measurement system is building a customized system. This can be either a CCD-area camera based system combined with inertial sensors, or a CCD-line camera based system. The latter is preferred mainly because with one single technology both accuracy and update rate requirements can be satisfied.

Risk analysis

A customized system requires more effort than setting up an of-the-shelf system. A firm, such as Schäfter & Kirchhoff can supply components, experience and partial development. In general, a larger involvement of an outside party increases the cost, but it may also reduce the (financial) risk (depending on the particular agreement).

3.4 Control System

The Control System is a key component of the robot since it executes the various control algorithms under evaluation. But it is also responsible for controlling high-level functions such as enabling the servos and provides safety features (protection against mechanical or electrical overload and provisions for emergency stopping). It consists of several hard- and software components:

- hardware for data acquisition (e.g., filtering, AD-conversion), command output (e.g., DA conversion), and overload protection;
- one or more hardware platforms to execute the controller and the user interface; and
- software for the controller and the user interface.

In this section, first, the low level function – the actual control implementation – is discussed, mainly from the point of view of the hardware platform, and a choice is made to use a PC based controller. Then, some safety features are addressed.

3.4.1 Control implementation

Although the software for the controller is equally important, its choice is largely predetermined: the focus is on hardware platforms running software which is closely integrated with the control development environment, i.e., Matlab/Simulink (design requirement (8), Section 2.3).

Specifications

The design requirements of the Control System are determined by the specifications of the RRR-robot as a whole, and the chosen joint angle sensors:

- *User interface*
  The user interface of the freely programmable Control System should be integrated with the control development environment (Matlab).
- *Additional inputs*
  Input facilities for at least 3 encoders should exist. In addition, the Control System should have extra analog and/or digital input facilities for future link state sensors and the Cartesian measurement system.
- *Sampling rate*
  Based on the update rate of the Cartesian measurement system: 1000 [Hz].

Hardware platform

There are several hardware platforms commercially available, each with its own capability, application and price [43]. For reasons of cost-effectiveness, the choice is limited to PC related solutions:

- PC hosted controllers, using an embedded DSP, and
- PC based controllers, relying on the PC CPU.

For both platforms computer aided controller design software based on Matlab and Simulink exists, enabling quick implementation, and a large degree of interaction with the control algorithm while running. This so called Rapid Prototyping is illustrated in Fig. 3.8.

However, compared to a well written hand-coded implementation in C, these automatic code generating tools are less efficient. A degradation by a factor two is not uncommon [43]. In addition, due to the desired generality of the code generation tools, dedicated software libraries are required to exploit special properties of a specific hardware platform.

PC hosted controllers PC hosted controllers consist of a general purpose PC, and an embedded DSP[2] computer with its own program and data memory. In addition, the DSP board is equipped with hardware for I/O: analog and digital inputs and outputs (see Fig. 3.6).

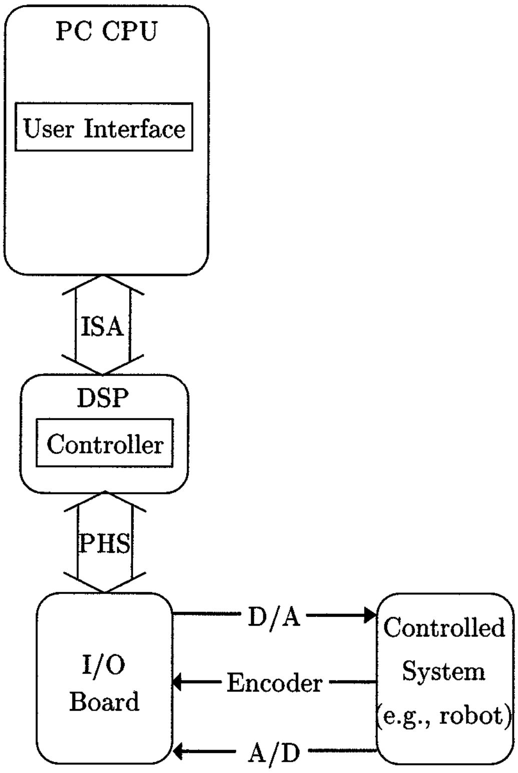


Figure 3.6: PC hosted controller configuration.

[2] Originally, Digital Signal Processors were designed to perform relatively simple operations on large amount of I/O. However, because of their speed, they quickly found their way to control applications were they are used for the opposite: to perform complex operations on a much smaller amount of data [16].

PC hosted systems with embedded DSP are reliable and guarantee real-time performance and deterministic behavior regardless of the CPU specifications and the operating system of the host. However, they are expensive, mainly because of the high cost of static RAM. A basic DSP control-board, the dSpace DS1102 with a Texas Instruments TMS320C31 digital signal processor, 4 analog inputs, and 4 analog outputs costs NLG 15K [14].

A potential problem when programming DSPs using a high level language like C is the lack of support of some of the special capabilities of DSPs. For example, the Texas Instruments TMS320C4X user's guide [20], makes explicit recommendations to (partly) implement the (control) algorithm in assembly language if large sample rates are required. Obviously, the use of automatic (C) code generating tools further complicates matters.

Van der Linde [43] uses a 4-processor C40 system to control an industrial hydraulic RRR-robot. The I/O time of his system is 64.6 [μs] (reading three encoders: 38 [μs], reading three pressures with 16 bits ADC: 17.6 [μs] ; and actuating three hydraulic servos with 16 bits DAC: 9 [μs]). For a computed torque controller, hand-coded in C, a sampling frequency of 1 [kHz] could be achieved (with an actuator pressure loop at 5 [kHz] needed because of the fast, 1 [kHz], pressure dynamics).

PC based controllers PC based controllers use one or more PC CPUs (central processing units) for both the controller and the user interface. The I/O is provided by plug-in cards (see Fig. 3.7).

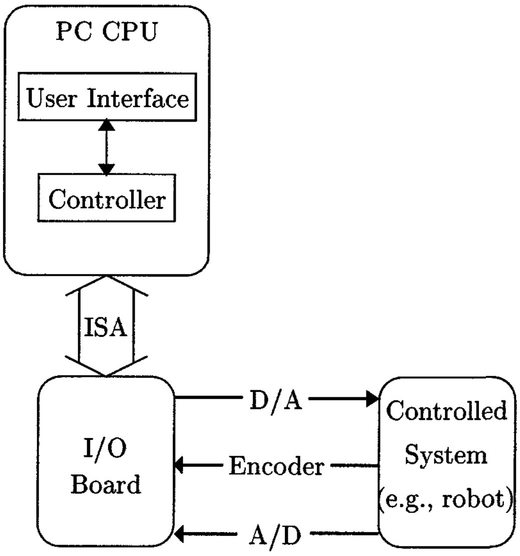


Figure 3.7: PC based controller configuration (using one CPU).

Although the specifications of the PC CPU now determine the performance, a single PC based controller is cheaper when compared to the PC hosted controller: only a PC and I/O hardware is needed, not an embedded DSP. Because of the rapid advance in PC technology and the large market, the performance of modern CPUs increases rapidly while the price of a new system remains at a relatively constant level. In addition, partly upgrading the PC from one generation to the next is possible without requiring the purchase of a new system.

Costescu and Dawson [11] compared the performance of a benchmark controller (written in C) on a TMS320C30 to that of a Pentium/Pentium Pro running various operating systems. They concluded that modern CPUs can outperform this DSP board by a factor 30 when running complex control algorithms. However, from their comparison it is not clear whether the difference in performance is caused by the hardware alone, or by the combination of hardware and software (e.g., due to their market-share, compilers for Pentium processors may be better optimizing then those for DSPs).

Further enhancement of the performance can be achieved by using multiple PC CPUs, e.g., one CPU for the user interface and one for the controller. These CPUs can be connected using an Ethernet network interface or by using a dual processor board within one PC.

An important aspect using PC based controllers is the operating system (OS). A real-time system should ensure fast, predictable CPU access for critical real-time tasks. For many operating systems, e.g., the popular Microsoft products, this is a problem since the host CPU not only performs real-time tasks but also system services, such as controlling the video display and disk drives, and responding to input devices. When the CPU is providing system services, hardware interrupts are frequently disabled, and for an indeterminate amount of time. As a result, critical I/O devices may not be able to gain access to the CPU for extended periods.

A *hard real-time* OS, such as QNX [27], has a fixed time delay (latency) for operations. *Soft real-time* implies this delay has a distribution, and a range. This range is determined by the various device drivers (e.g., video, keyboard and mouse) and their interaction, and therefore not known in advance. Although Windows 95 and NT are better than Windows 3.11, because they have a priority based scheduler which can preempt low priority processes, they can not guarantee hard real-time capabilities [11].

IA-SPOX developed by Spectron [37] is an extension of Windows 95 which enhances its soft real-time capabilities (amongst others by measuring and optimizing the latency of the device drivers) to achieve a bounded maximum latency (1 [ms]) with a narrow distribution. But even then, any device driver (e.g., when using keyboard or mouse) can shut-off hardware interrupts for a non-deterministic period of time.

If the use of a soft real-time system is still preferred, there are several ways to reduce latency to an "acceptable" level:

- minimize the number of active window applications (the demand on the memory), avoid DOS applications,
- minimize display updates,
- minimize the use of mouse and keyboard,
- use memory for disk caching, and
- careful select the PC system components (if latency characteristics are specified).

An example of a commercial soft real-time system with Simulink and Matlab integration (see Fig. 3.8) is Wincon from Quanser Consulting [10]. In combination with their MultiQ I/O board for control purposes (8 × 13 bits ADC, 8 × 12 bits DAC, 8 digital I/O, 6 encoder inputs, and 3 hardware timers), this Windows 3.1 package costs USD 2K. Without specifying the controller, it claims sample frequencies of 2 [kHz] on a Pentium using an external clock, or 1 [kHz] using the Windows 3.1 clock. The new Windows 95 version of Wincon (due March 1998) can use multiple PCs and claims an even higher performance: using two PCs, and an Ethernet connection, a controller with 64 Simulink blocks, 2 analog inputs, 3 analog outputs and 3 encoder inputs runs at a sampling frequency of 15 [kHz] on a Pentium 200 system (maximum interrupt latency < 20 [μs]).

With a different PC operating system, the same hardware (MultiQ I/O board + PC) can also be used to realize a hard real-time system. Costescu and Dawson [11] developed such a system using the QNX OS. However, up to now their system offers no Simulink or Matlab integration.

Selection

In summary, when high level programming (in C/Fortran or even using a auto-code generating tool) is required, PC based controllers can offer a performance at least equal to a PC hosted solution at a much lower price. In general, PC hardware not older than one generation can be easily upgraded. Furthermore several interface [21], and even special control plug-in boards are available.

However, the reliability of a PC based controller is not up to industrial standards like the PC hosted controller. Especially, if one PC CPU is used, unpredictable behavior can occur when the OS system crashes. Since no commercial hard real-time solutions exist, the user is responsible for reducing the latency, and verifying the real time execution of the controller.

In a research environment, this is an acceptable handicap, and based on the performance and price, a choice is made to use a PC based controller: the MultiQ I/O board and a fast Pentium Pro 200 running Wincon.

Risk analysis

For any control system, a lack of computational power to implement a specific control algorithm is fatal. However, assessing this risk is difficult. Because of the interaction of hardware and software, a hardware performance measure such as the number of floating points operations breaks down.

When using the chosen PC based controller a number of modifications is possible:

- changing the operating system, e.g., to Windows 95 or QNX,
- updating the hardware: using a faster CPU, or
- expanding the hardware: using a second PC CPU.

3.4.2 Safety provisions

Safety provisions are necessary to provide safety for the operator and spectators. In a laboratory environment also provisions must be made to prevent damage due to operator mistakes or unstable controllers.

Electrical shielding All motor drivers and high voltage connections must be shielded in order to comply with all legal requirements.

Hardware emergency stop To stop the robot in case of a crash of the controller software or the operating system, a hardware emergency switch is necessary to cut open the control loop and shut down the power to the servos. This hardware switch is also the preferred stopping method for al other emergencies because it can provide a fail-safe stop.

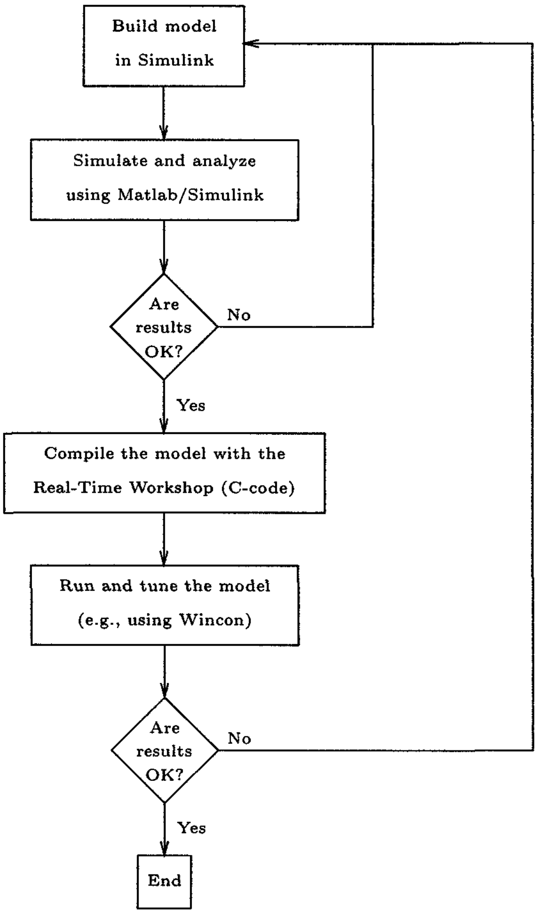

Build model in Simulink
Simulate and analyze using Matlab/Simulink
Are results OK?
No
Yes
Compile the model with the Real-Time Workshop (C-code)
Run and tune the model (e.g., using Wincon)
Are results OK?
No
Yes
End

Figure 3.8: Rapid prototyping using Matlab, Simulink and the Real-Time Workshop (products from The MathWorks Inc. [19]). Here in combination with Wincon.

Electro-mechanical brake(s) For both human safety and to prevent mechanical overload of the motor, electro-mechanical brakes are chosen to bring the robot to a controlled stop. The brake(s) should be automatically engaged in case of a hardware emergency stop or a power shutdown. It should also be accessible from the controller, e.g., in case of a software shutdown.

Software shutdown To prevent mechanical overload the state variables should be monitored by a high-level controller. If pre-set values are exceeded, the controller outputs should be reduced or nullified. Alternatively, the brake can be engaged.

3.5 Signal-Transfer System

The Signal-Transfer System transports power and measurement signals across parts which move relative to each other. Various transfer technologies exist; some only suited for data transport, other suited for both power and measurements signals. In the RRR-robot, power can only be transported serially, i.e., power from the base to the end-effector servo must pass across two barriers. Measurement data can also be transmitted to the base directly.

In the following sections several transfer technologies and their properties are addressed. Although in principle various combinations are possible, the implementation of the Signal-Transfer System is almost completely determined by the physical characteristics of the chosen Joint-Actuation System. Without modifications to the drivers, the only practical option is to implement serial connections using sliprings for all motor-driver leads.

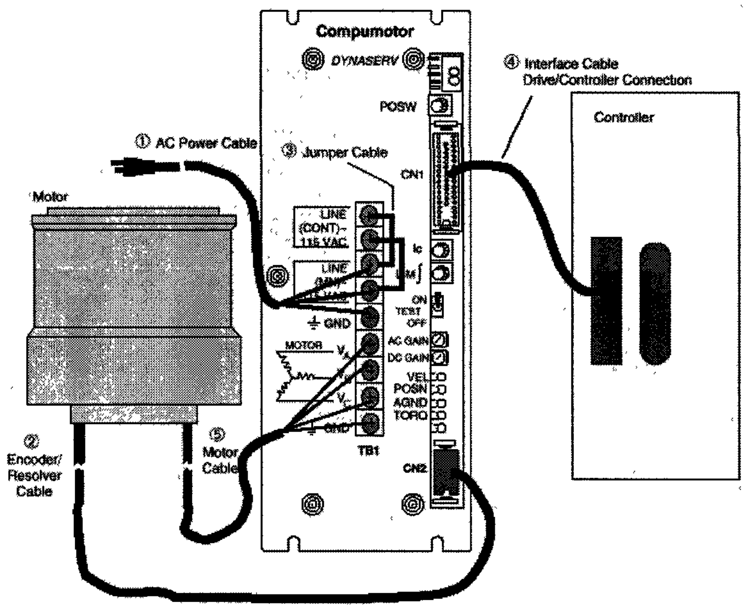


Figure 3.9: All connections to and from the *Dynaserv* driver.

Signal transfer methods

Contact rotary signal transfer The most basic transfer method to transport power or data through a rotary joint is the use of sliprings. They consist of brushes rotating over conducting rings. The construction is simple and modular, i.e., by stacking shielded rotary conductors and brush pairs, multiple (power and/or data) channels can be created. However, mechanical contact introduces friction (causing wear and noise especially at higher currents) and electrical resistance.

Contactless rotary signal transfer By use of an intermediate medium, mechanical contact can be avoided, thus making the connection more robust to wear and aging. Compared to sliprings, the construction is more complex since at least two conversions are required, i.e., electrical currents to the used medium, and back. Therefore, the number of channels is often limited, requiring a multiplexing solution.

A magnetic connection consists of two coils, placed opposite to each other. The efficiency (typically 90%) depends on the size of the air-gap, the resistance of the coils and the amplitude of the power source. However, combining multiple independent channels in one rotary joint is impossible without interference.

Capacitive connections consist of two conducting rings (plates) placed opposite to each other and shielded by two isolated rings on each side. Because of the low efficiency (10–20%) power transfer is difficult.

Optical rotary joints are suited for data transfer only. They consist of one or more LEDs opposite to photo-transistors or diodes. Because the number of channels is limited (e.g., the dual channel FOR000P2S0 from Schleifring [33]), a multiplexing solution is necessary.

Wireless data transfer Wireless data transfer is the only direct, non-serial transfer method, e.g., data from the last link does not have to pass three joints, but is transmitted directly to the Control System. In general the number of channels is restricted, so multiplexing is necessary.

Radio data transfer does not require a direct line-of-sight, but in an indoor environment with reflections and echos, special modulation techniques are required. With the use of infrared serial modems [35] a line-of-sight must be maintained.

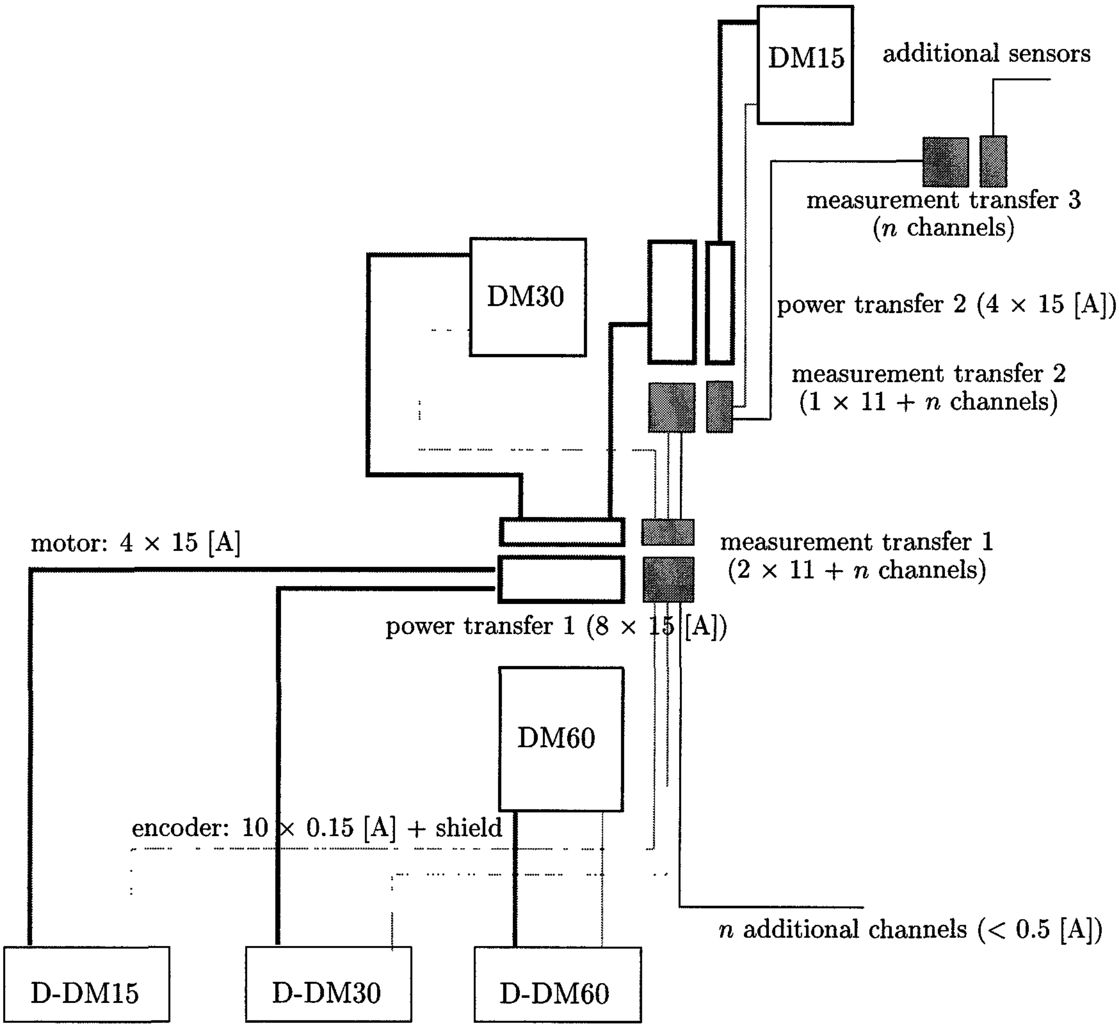


Figure 3.10: Schematic of a Signal-Transfer System with complete serial power and measurement connections. D-DMn are the motor drivers, DMn the servos.

Selection

In Fig. 3.9 all connections to and from the Dynaserv driver are shown. Since the drivers are too large to integrate in the robot, connection ⑤ (4 × 15 [A], max. power 2 [kW]) and connection ② (11 × 0.15 [A] DC) are part of the Signal-Transfer System.

So for the transfer of power, a serial connection (i.e., the power to each motor passes through all previous links) of all three motor phases and the ground is the only possible configuration (see Fig. 3.10).

For the transfer of measurement data a multiplexing system, either wireless or by using a bus system, could greatly simplify the Signal-Transfer System. Compared with the complete serial connections of Fig. 3.10, the number of measurement channels for each links could be eliminated or at least reduced (e.g., to one twisted pair connections). Commercial device(bus) networks such as Profibus-DP [30] and DeviceNet [3] provide complete solutions for connecting actuators and sensors in twisted pair networks. However, the cycle time of these networks is too large ($>$ 0.2 [ms]) to achieve a sampling rate of 1 [kHz] [31, 15].

Moreover, it is difficult to predict whether the motor drivers will accept demultiplexed feedback from the encoders. Therefore, a choice is made to implement the serial connections of Fig. 3.10 with a number of additional free channels.

These free channels can be used to connect additional sensors on the last (flexible) link. Power to these sensors ($<$ 0.5 [A]) must be provided either by a battery or by sacrificing one or more of the same channels. Alternatively, these channels could be used to facilitate a bus system for additional sensors.

The only practical technology which facilitates modular combination of multiple (serial) connections is brushed power and slipring transfer. The design specifications of these slipring units are determined by the chosen Joint-Actuation System and given in Table 3.3.

Table 3.3: Design specifications slipring units. The number of additional channels, n should sufficient for at least 3 additional link sensors on the end-effector, i.e., $n \geq 8$

System	# of channels	current [A]
power transfer 1	8	≥ 15
power transfer 2	4	≥ 15
signal transfer 1	$22 + n$	≥ 0.5
signal transfer 2	$11 + n$	≥ 0.5
signal transfer 3	n	≥ 0.5

Because sliprings are available in various shapes and sizes, the choice for specific components is made in conjunction with the synthesis of the subsystems in Chapter 4.

Risk analysis

The need for a Signal-Transfer System in general, and more specific the use of sliprings introduces a difficult to assess risk of increased sensitivity for external disturbances. In addition sliprings for power transfer may be damaged due to "welding effects".

- Disturbance sensitivity: especially, the encoder feedback is sensitive to disturbances: every missed pulse reduces the position accuracy.

Actions:

- the use of additional cable shields, and
- the use of redundant circuits to cancel out possible disturbances, as illustrated in Fig. 3.11.

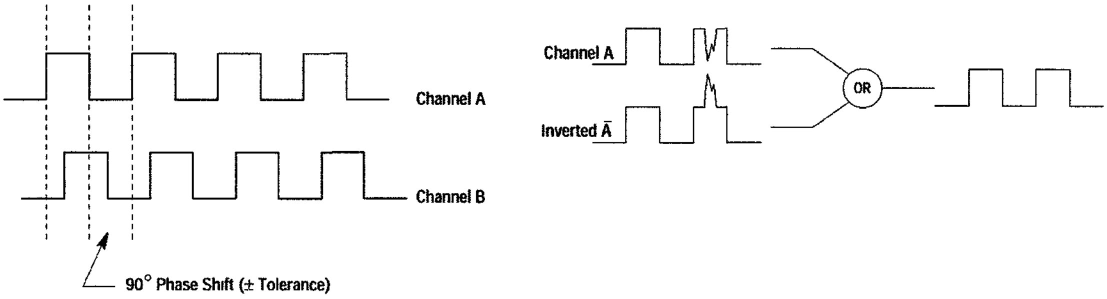


Figure 3.11: The Dynaserv servos use 10 circuits (and a shield) for the encoder feedback: 2 for power, and 8 for data. The latter consist of 4 twisted pairs transmitting the pulse trains A, $\overline{A}$ (A inverse), B, and $\overline{B}$. Assuming a disturbance acts both on A (or B) and its inverse, it can be canceled out.

- Slipring welding effects: the power sliprings may be damaged if they are operated at continuous standstill, or if a power surge occurs. Note that in most applications, sliprings are continuously rotating while transferring power.
  Actions:
  - Dimensioning the power sliprings based on maximum power generated by the motor drivers (limited by fuses).
  - Operational warnings not to operate the sliprings, and thus the robot at continuous standstill of the first two joints (see the RRR-robot instruction manual [41]).

Chapter 4

Synthesis of subsystems

After the selection of appropriate components and design solutions in the previous chapter, now, all subsystems are combined. First, the design of the *manipulator frame* is discussed. Next, the *Control System* is added, and an evaluation of the system is performed. Finally, the RRR-robot specifications are discussed.

4.1 Manipulator frame

Obviously, the design of the manipulator frame can not be separated from the previously discussed subsystems. Especially, the Joint-Actuation System, the Measurement System, and the Signal-Transfer System are closely interconnected with the frame design. In Section 3.2, a specific series of servos was found which can satisfy the main requirement: maintaining high Coriolis and centrifugal torques. These so called "Dynaserv" motors are self-contained units with internal bearings and suitable joint angle-sensors (see Appendix B).

Although restricting the designers freedom, the use of these complete servo-units can greatly simplify and speed-up the design process. Therefore, a choice was made to design the manipulator frame starting with the three Dynaserv motors from Table 3.1. The components of the slipring based Signal-Transfer System are selected in conjunction with the frame design based on the available space and available commercial products.

In the following sections various design aspects are discussed. Leading to the conceptual design of Fig. 4.1 and Fig. 4.2. Based on this design the CTD manufactured all non-standard components (see Table 4.2).

4.1.1 Motor load

The limiting factor in using the Dynaserv servos is the bending moment their bearings can handle: 200 [Nm] for static load, and 60 [Nm] for alternating load. However, these limits are calculated for a lifetime of 5 years, and continuous pick-and-place applications with a trapezium shaped velocity profile. In the RRR-robot, the motors will be used for much shorter periods, but with larger accelerations. Therefore, it is preferable to design the robot for smaller loads.

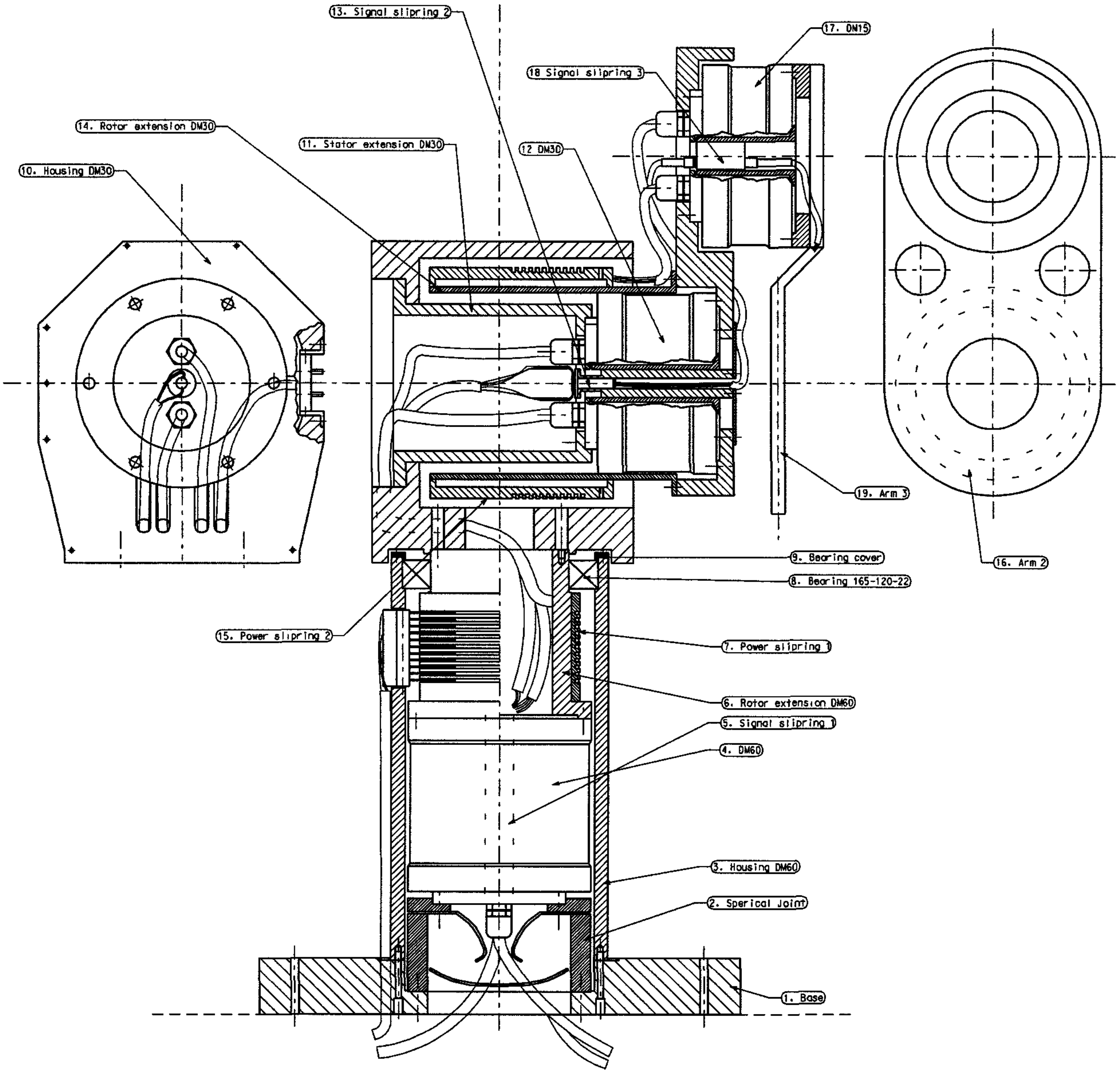


Figure 4.1: RRR-Robot conceptual design, used as a basis for the realization of the various non-standard components. Note that some dimensions were slightly altered during the manufacturing

Table 4.1: Moving parts: joints and links

part	subcomponents	mass [kg]
Joint 1	DM60	12
Link 1 (vertical)	Rotor extension DM60, PSR-1, Housing DM30, and Stator extension DM30	20.6
Joint 2	DM30	7.5
Link 2	Rotor extension DM30, PSR-2 and Arm 2	8.9
Joint 3	DM15	5.5
Link 3	Arm 3 (not present)	-

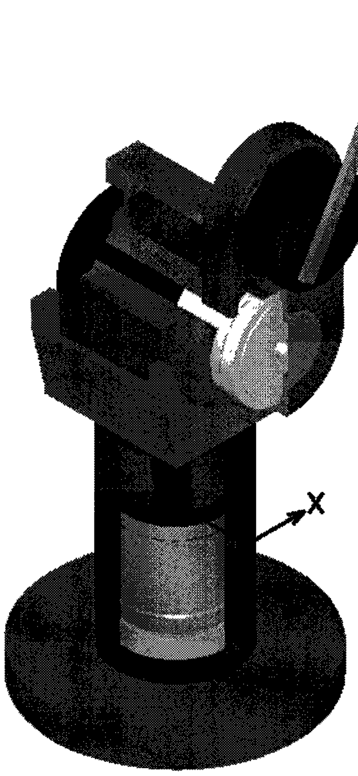


Figure 4.2: 3D view of the conceptual design.

Table 4.2: RRR-Robot main components (see Fig. 4.1). With the exception of part 8, all parts manufactured by the CTD are made of Aluminum. The masses are either measured, estimated, calculated (indicated with a *), or supplied by the manufacturer.

No.	Part	Source	Mass [kg]
1.	Base	CTD	15*
2.	Spherical joint	CTD	2*
3.	Housing DM60	CTD	6*
4.	DM60	Litton (Appendix B.1)	12
5.	Signal slipring 1 (SSR-1)	Litton (Appendix B.3.2)	< 0.1
6.	Rotor extension DM60	CTD	3.504
7.	Power slipring 1 (PSR-1)	Litton (Appendix B.3.1)	
8.	Bearing 165-120-22	CTD	< 0.5
9.	Bearing cover	CTD	< 0.1
10.	Housing DM30	CTD	15.5
11.	Stator extension DM30	CTD	1.620
12.	DM30	Litton (Appendix B.1)	7.5
13.	Signal slipring 2 (SSR-2)	Litton (Appendix B.3.2)	< 0.1
14.	Rotor extension DM30	CTD	1.482
15.	Power slipring 2 (PSR-2)	Litton (Appendix B.3.1)	3.464
16.	Arm 2	CTD	3.908
17.	DM15	Litton (Appendix B.1)	5.5
18.	Signal slipring 3 (SSR-3)	Litton (Appendix B.3.2)	< 0.1

The moment loads are caused by Coriolis and centrifugal forces acting perpendicular and parallel to the rotation axes. Motor 1 (4. DM60 in Fig. 4.1) is submitted to the largest forces, caused by its own rotation velocity, and acceleration; and those of motor 2 and 3 (12. DM30 and 17 DM17). Since otherwise the specified load is exceeded, an additional (radial) bearing is needed to reduce the moment load on motor 1.

However, adding a second bearing causes Link 1 (defined in Table 4.1) to be kinematicly overdetermined, and thus imposes severe demands on its centering and alignment. The shaft metal of the motor may get damaged if the centering offset remains 10 [μm] or more (see the Dynaserv manual). To bring down the kinematic degrees of freedom, a choice was made to partly "disable" the internal motor bearings. This can be achieved by mounting the motor on a membrane plate, or more practical on a spherical joint (2. Spherical joint in Fig. 4.1). With this solution, Motor 1 is completely discharged from moment load.

Motor 2 (12. DM30 in Fig. 4.1) is now subject to the largest Coriolis and centrifugal forces. In Fig. 4.3 the moment load on Motor 2 is plotted for two different loading directions. The accelerations are chosen to represent the largest values possible based on the inertia matrices and the maximum motor torque. As can be seen, the load remains well below 60 [Nm] if the chosen nominal motor velocities of 0.5, 1 and 1.5 [rps] are not exceeded.

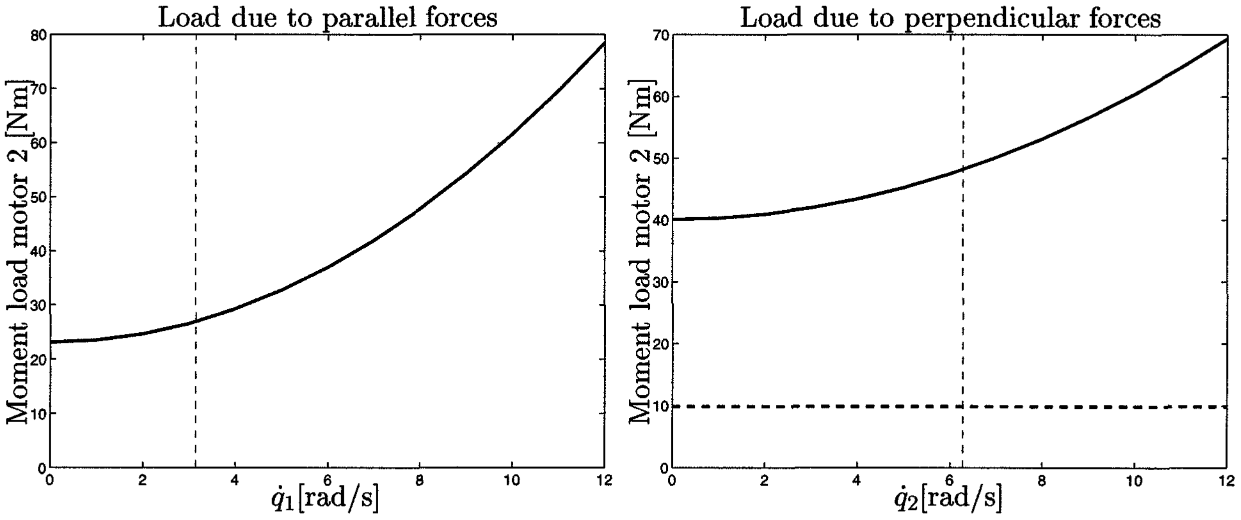


Figure 4.3: Moment load on motor 2 due to forces parallel and perpendicular to the rotation axis. This load should remain below 60 [Nm]. Unless otherwise specified the velocities are $\underline{\dot{q}} = [\pi, 2\pi, 3\pi]^T$. Rotation angles and accelerations are set to represent worst-case values (the maximum accelerations are based on the inertia matrices).

Bearing type For the additional radial bearing a large, and thus stiff ball bearing is selected (ball bearing CTD 165-120-22) based on the available space. The calculated worst-case life span of this bearing is 1.25 10^{12} revolutions. This is more than adequate considering the required technical lifespan of 5 years:

$$10 \text{ year} \times 40 \text{ weeks/year} \times 5 \text{ days/week} \times 1 \text{ hour/day} \times 3600 \times 0.5 \text{ rps} = 3.6\ 10^6 \text{ revolutions}$$

4.1.2 Sliprings

Based on the specifications derived in Section 3.5, and the chosen nominal motor velocities of 0.5, 1 and 1.5 [rps], sliprings are selected for data and power transfer.

Data

Although, sliprings suitable for both data and power transfer exist, the combination of high and low power circuits complicates the selection. To satisfy the combined specifications, completely custom made and thus expensive slipring units are required. Therefore, exploiting a special feature of the Dynaserv motors – the hollow core (Fig. B.1) – small signal sliprings are placed inside the motors.

In Table B.4, the main specifications of the selected sliprings are summarized. Signal slipring 3 (SSR-3) has $n = 12$ (> 8) circuits. For SSR-1, a sliprings with 36 circuits is selected ($> 22 + 12$). Although less circuits are required, SSR-2 is chosen equal to SSR-1 because it is cheaper to order two identical sliprings than one with 36, and one with 24 circuits.

Note that the maximum sliprings velocities (1.67, 1.67, and 2 [rps]) are below the maximum individual motor velocities (2.4 [rps]). Apart from the motor load, this is another reason to maintain strict limitations on the motor velocities: 0.5, 1 and 1.5 [rps] for motor 1, motor 2, and motor 3 respectively.

Power

With the use of an additional bearing (8. Bearing 156-120-22), the total length of Link 1 is no longer important. Therefore Power Slipring 1 (7.) can be placed above Motor 1. Based on the specifications two off-the-shelve sliprings are selected (see Table B.3).

However, the length of Link 2, and especially, the offset distance between Motor 2, and 3 (parallel to their rotation axes) is relevant; it determines the moment load on Motor 2. To minimize this distance, Power slipring 2 is placed around and behind motor 2. In Table B.3, the main specifications of a modified, large diameter, slipring are summarized.

4.1.3 Eigenfrequencies

The magnitudes of several eigenfrequencies were estimated using simple mass-spring models of the design. The lowest eigenfrequency, 79 [Hz], of the system emanates from the motor bearings of the largest motor (the DM60). All other eigenfrequencies – emanating from the spherical joint, Link 1, or Link 2 – are higher.

4.2 Measurement System: Cartesian measurement system

In Section 3.3.3, the results of a survey of 3D tracking systems are summarized, and a recommendation is made to use a CCD-line camera based system. Since commercial systems are too expensive, this should be a custom-made, or even homebuilt system.

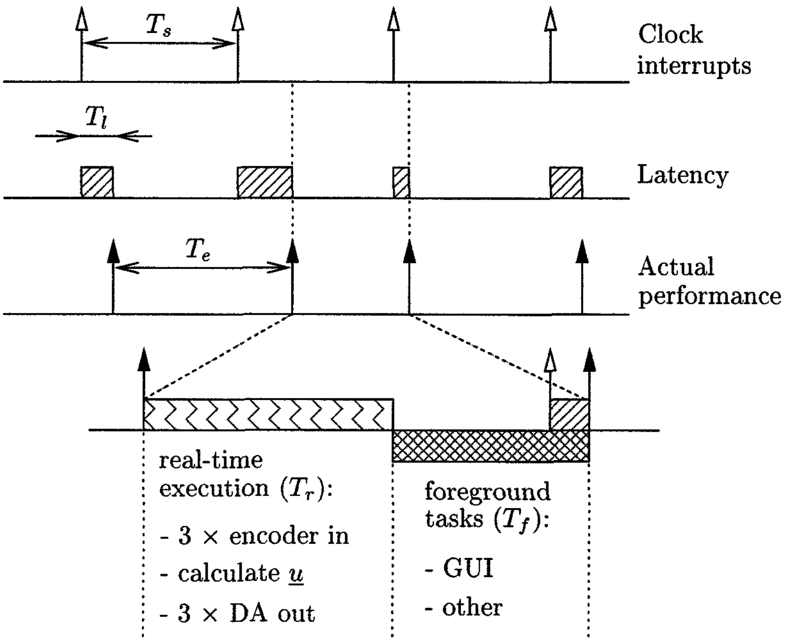
Designing, and building this system, however requires considerable time, and effort; and therefore has not yet been realized.

4.3 Evaluation

In this Section, the manipulator frame is combined with the Control System, and a model based control algorithm is implemented in Simulink to test the real-time performance of the controller. Next, several experimental results are discussed to evaluate the performance of the chosen servos. Finally, all RRR-robot specifications are discussed.

4.3.1 Real-time performance

In Section 3.4, a PC based controller was selected: the MultiQ I/O board in combination with a soft real-time operating system (Windows 3.11). Although, this controller offers Matlab and Simulink integration, it may cause a variation in the effective sample frequency. In Fig. 4.4, the parameters determining the performance are defined.



T_s: desired sampling period (with the internal windows clock).

T_l: latency, the time to stop the foreground tasks and start the real-time section.

T_e: effective sampling time, with an average $\overline{T_e}$, and variance $\sigma(T_e)$.

T_r: time to execute the real-time code, e.g., generated from the block diagram shown in Fig. 4.5: read out the encoders, calculate the control law, and send the calculated values to the DA converter.

T_f: time remaining for foreground tasks, e.g., the graphical user interface (GUI) of the controller.

Figure 4.4: Parameters defining the real-time performance.

Using an internal or external timer, interrupts are generated with a desired sampling period T_s. When an interrupt occurs there is a latency T_l before the foreground tasks are stopped, and the real-time controller is executed, resulting in a varying effective sampling period T_e. The time T_r to execute the real-time code should be less than the desired sampling period minus the latency, i.e.,

$$T_r \leq T_s - T_l. \tag{4.1}$$

Note that T_r includes the time needed for analog and digital input and output (see Table 4.3). If (4.1) is satisfied, $\overline{T_e} = T_s$. Else, interrupts are missed, and the controller will perform differently than expected. Ideally, T_l should be constant, then $T_s = T_e$, and only the initialization of the controller is delayed by T_l.

Unfortunately, most of the real-time parameters depend on the combined use of hardware and software (see Section 3.4), and are difficult to predict. Therefore, the real-time performance has to be verified experimentally.

By comparing the Wincon real-time display with a hand-held chronometer, it can be checked whether (4.1) is satisfied. If after several minutes, the Wincon display is lagging behind, interrupts have been missed. From the delay, the average sampling period can be estimated:

$$\overline{T_e} = \frac{t_{\text{chronometer}}}{t_{\text{Wincon}}} T_s \tag{4.2}$$

For a more accurate measure of the real-time performance, both the average and the variation of the effective sampling period T_e can be determined by monitoring an additional test output, e.g., by switching one of the DA channels between two voltages at every sample. The output monitored on an oscilloscope should be a square wave with a period $T_s/2$.

The achievable sampling period T_e is not only limited by (4.1), but also by the clock used to generate the interrupts every T_s [s]. The internal Windows 3.11 timer limits the clock frequency to $\frac{1}{T_s}$ = 1000 [Hz]. Higher clock frequencies (up to 2 [MHz]) require the use of one of the on-board clock timers, and have not yet been tested.

Table 4.3: Several I/O times and noise levels of the MultiQ board (measured using the separate test software supplied by Quanser Consulting, see Appendix B for a summary of all characteristics).

Input/Output	I/O times [μs]	Noise level bounds
Encoder read (24 [bit] counter)	5.5	
ADC (13 [bit])	20	±3 LSB ≙ ±3.7 [mV]
DAC (12 [bit])	5	±2 LSB ≙ ±4.9 [mV] ≙ ± 0.06% max. motor torque

Passivity-based Computed Torque To put the Control System to the test, a computational involved Passivity-based Computed Torque Controller [24] is implemented, i.e.,

$$\underline{u} = \underline{\tau} = M(\underline{q})\underline{\ddot{q}}_d + C(\underline{\dot{q}}, \underline{q})\underline{\dot{q}}_d + \underline{g}(\underline{q}) + K_D\underline{\dot{e}} + K_P e \tag{4.3}$$

with $\underline{e} = \underline{q}_d - \underline{q}$ the tracking error, and K_D and K_P positive definite feedback matrices. This method exploits the passivity property of mechanical systems to guarantee asymptotic stability of the closed loop system.

The controller is implemented in Simulink, as shown in Fig. 4.5. Pulses from the encoders are read in, and voltages – proportional to the desired torque – are send to the motor drivers. Based on this block scheme, the real-time code is generated.

Equation (4.3) is contained in the "Controller u [Nm]" block and requires 185 flops (determining a lower bound for the computational load). The Simulink input for M, C, and $\underline{g}$ was generated by Mathematica based on the rigid-robot model. This model has been refined using the inertia data generated by the CAD program Microstation (see Appendix A).

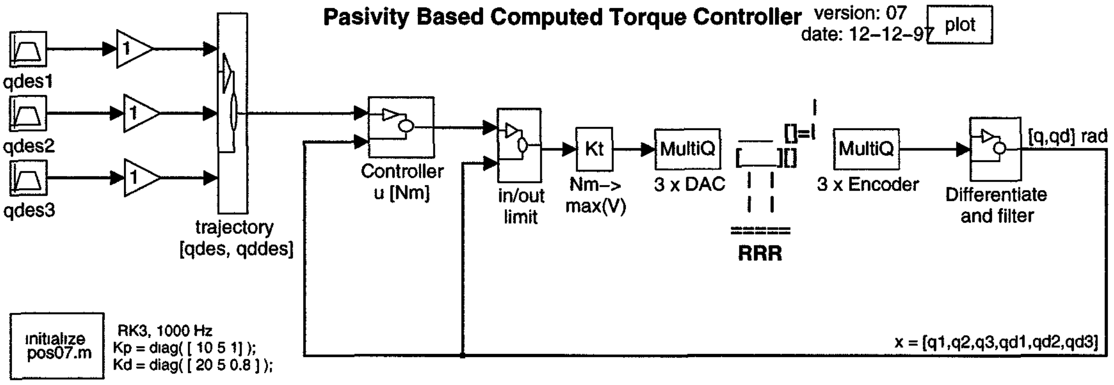


Figure 4.5: The (top-level) Simulink block scheme of a Passivity Based Computed Torque Controller.

Without AD conversions, the resulting I/O time is 31.5 [μs] (see Table 4.3). By monitoring a test signal (a block wave), it was verified that an effective sampling frequency of 1 [kHz] can be achieved without missing interrupts, i.e., $\overline{T_e} = T_s$. Furthermore, the variation of T_e is smaller than 0.05 [ms]. So, for the implementation of this computational involved (model based) control algorithm, the real-time performance seems adequate.

4.3.2 Servo performance

Torque vs command voltage Ideally, the relation between the command voltage and the torque exerted by the motor is linear, i.e., $T_{\text{output}} = f(U_{\text{command}}) = K_t\, U_{\text{command}}$. However, the exact value of K_t, or function $f(U_{\text{command}})$ is unclear. In Fig. 4.6, three motor output relations are shown for motor 2: based on experiments and data supplied by two different suppliers of Dynaserv servos.

In the experiment, arm 2 is rotated at a very low ($<$ 1[rpm]), but constant (controlled) velocity. The torque needed for this rotation can be determined from the combined gravity and friction load, i.e., $u_2 = 14.52\sin(q_2) + T_{f_2}$ [Nm], where T_{f_2} is the (constant) coulomb friction torque. The resulting torque constant $K_t = 6.7$. The friction torque $T_{f_2} = u_2(q_2 = 0) \approx 3$ [Nm]. The other two motor output relations are based on the (relative) saturation curve of Fig. B.2 from Compuserve, and the torque constant stated by Litton (see Table B.1).

The differences between these input-output relations are considerable, so in order to achieve high performance tracking using model based control laws, more experiments (over the full output range) are necessary to determine the correct functions $f(U_{\text{command}})$ for all three motors.

Furthermore, the friction torque of the Dynaserv servos ($\approx$ 3 [Nm] for all three) is larger than expected (Litton stated a friction torque of 5% of the maximum torque). So for high performance tracking, friction compensation is necessary as well.

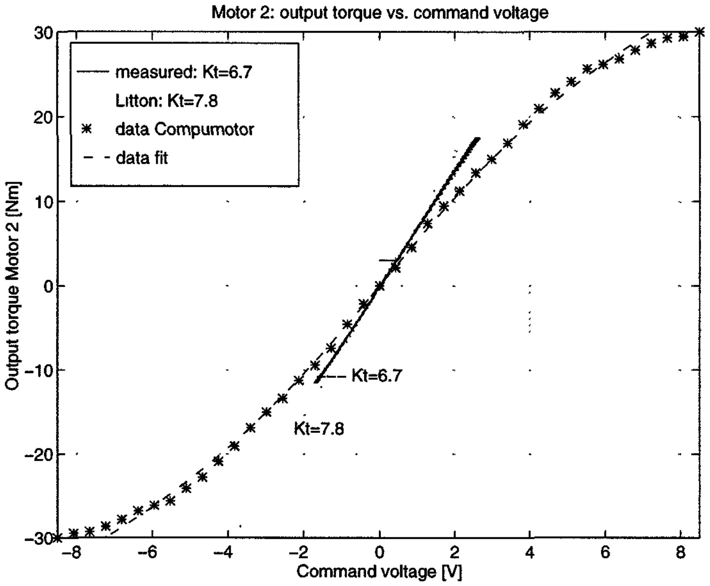


Figure 4.6: Output torque motor 2 as a function of the command voltage based on experiments, and data from two suppliers of Dynaserv servos (Litton [26] and Compumotor [9]).

Model quality Figure 4.7 shows a comparison for both experiments and simulation for the computed torque controller. The desired trajectory is $\underline{q}_d = [\pi\ t, 2\pi\ t - \pi/2, 3\pi\ t]^T$ with initial conditions $\underline{\dot{q}}(0) = \underline{0}$ [rad/s] are shown.

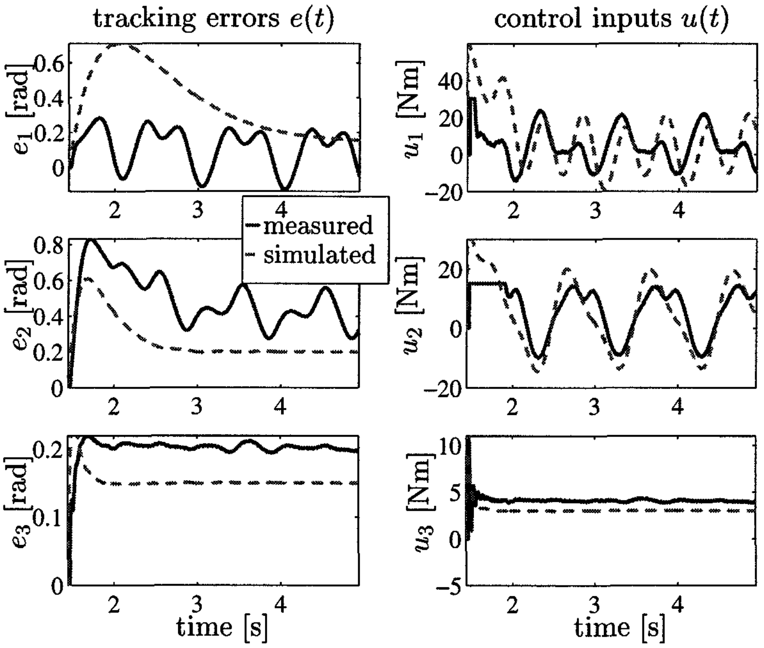


Figure 4.7: CTC controller, simulated and measured position errors and control inputs. With trajectory $\underline{q}_d = [\pi\ t, 2\pi\ t - \pi/2, 3\pi\ t]^T$, and initial conditions $\underline{q}(0) = \underline{\dot{q}}(0) = \underline{0}$ [rad]

The simulations are performed with the enhanced rigid-robot model implemented in Mathematica with an non compensated friction torque of 3 [Nm] for each motor.

The controller parameters were set for a slow response. Motor constant 2 is set to 6.7 [Nm/V]. As can be seen, there are some appreciable differences between simulation and experiments, probably due to incorrect model parameter values.

Contribution Coriolis and centrifugal torques To obtain an indication of the performance, ratio r (2.2) is evaluated by substituting measured positions, velocities and accelerations into the model. Two links have a ratio of 0.5 or more: $r = [0.76, 0.32, 0.83]$.

The presence of friction obviously reduces the contribution of the Coriolis and centrifugal torques. But even if the friction torque is included into ratio r, the performance ratio of link 1 remains above 0.5.

4.3.3 Design specifications

In this Section the RRR-robot specifications (Section 2.3), and thus the main features of the combined subsystems, are discussed. Although the robot is designed to satisfy all requirements of Section 2.3, currently, the requirements with regard to the static, and dynamic end-effector accuracy (requirements (4) and (5)) are difficult to evaluate because of the absence of a Cartesian measurement system for the end-effector.

Design requirements

(1) By using sliprings to facilitate unconstrained rotation, and direct-drive servos, the Coriolis and centrifugal torques on at least one link can be made sufficiently large (see the experimental results of Section 4.3.2).

(2) The actuator torques are a direct function of the command signals. However, this function is possibly non-linear (see Fig. 4.6), and remains to be measured.

(3) The RRR-robot has three rotational degrees of freedom.

(4) The accuracy of joint-angle sensors (±15 [arc-sec]) should – after kinematic calibration – enable a static end-effector accuracy of $\underline{e} = [\Delta x, \Delta y, \Delta z]^T = [0.036, 0.036, 0.058]^T$[mm].

(5) Assuming all joint and links are rigid (no unintentional flexibilities), a lower bound for the absolute end-effector $|e(t)|$, could be determined by observing the tracking behavior of a well tuned controller. However, this kind of fine tuning has not yet been performed.

(6) The third link is exchangeable with a flexible one.

(7) A torsionally flexible element (see, e.g., [32]) can be attached to the third servo (or joint), thus transforming it into a flexible drive.

(8) The control interface is based on Simulink (and Matlab).

(9) The characteristic dimension – the length of Link 1 – is 0.5 [m];

(10) The technical lifespan of all components is at least 10 year.

(11) If the Cartesian measurement system is be realized with NLG 50K, the (expected) total realization cost is about fl 160.000,-, slightly exceeding the original budget (see Appendix C).

(12) Because of its size, the robot is suitable for demonstrations for a group of at least 10 people.

(13) The system is made safe for operators and spectators by means of electrical shielding and restricting the access to the robot's workspace.

4.3.4 Design goals

With respect to the design goals, the RRR-robot has the following features:

(a) Modern, state-of-the-art techniques and components are used: compact direct-drive servos, miniaturized signal sliprings, and automated real-time code generation.

(b) On Link 3, 12 channels are available for additional sensors.

(c) The RRR-robot has the same structure as the familiar "Puma"-type industrial manipulator (without a wrist).

(d) Maintenance is restricted to cleaning the contacts of the power sliprings. Both units can be reached by simply removing the brushblocks.

(e) For demonstration purposes, each link has a separate color. In addition, the power slipring brushblock terminals have Plexiglas covers.

Chapter 5

Conclusions and recommendations

5.1 Conclusions

Result An industrial-like experimental facility, the RRR-robot, has been realized that offers the ability to enhance Coriolis and centrifugal torques relative to other nonlinear effects. Two important features of the RRR-robot were introduced in order to reach this design goal: unconstrained rotation of each link by using sliprings, and the use of direct-drive servos. Although, the friction torques of these servos are larger than expected, the contribution of the Coriolis and centrifugal torques on one link can be made sufficiently large. A rigid-robot model has been implemented in *Mathematica* to aid servo sizing. Using the CAD package *Microstation*, this model has been refined for simulation purposes, and to aid in the implementation of controllers by generating Simulink input.

Frame design The robot has a modular setup, i.e., it has been designed starting from a specific type of servo with internal bearing, and angular sensors. The last link can be replaced by a flexible link, and the last joint can be transformed into a flexible joint. Thus making it suitable to evaluate control schemes for both rigid-robots and flexible joint and/or link robots.

Control System The RRR-robot is equipped with a Simulink based graphical user interface, offering optimal flexibility to the user. The computational power of this Control System has been demonstrated by implementing a model based control algorithm. Several provisions have been made to provide safety for operators and spectators, but also to prevent damage due to operator mistakes or controllers that destabilize the closed loop system.

Cartesian measurement system Considering the combination of high accuracy, high update rate and uninterrupted tracking only one separate technology is suitable for tracking the end-effector of the RRR-robot: vision based ranging using multiple line-array cameras. Since available commercial system exceed the available budget (NLG 50K) by a factor 4, this should be a custom-made or even homebuilt system. Designing and building this system requires considerable effort. Although, several (engineering) firms were contacted, a Cartesian measurement system has not yet been realized. As a consequence, the required static and dynamic position accuracy has not yet been validated.

Budget Comparing the expenses, and the original budget (Appendix C), the Control System (posts 4 and 5), and the Joint-Actuation System together with the Signal-Transfer System (post 2) both turned out to be cheaper. The construction of the frame parts, especially the shielding of the motor drivers (required by law) resulted in considerable larger expenses than anticipated. If the budget for the Cartesian measurement system remains unchanged, the total budget will be exceeded by approximately 5%.

5.2 Recommendations

In its present state, the RRR-robot can be used to evaluate rigid-robot controllers in the joint space. To evaluate control algorithms for flexible joint and/or link robots, and to calibrate the robot, additional components are required:

- an optical measurement system (using line-array cameras) to determine the Cartesian position of the end-effector,
- a flexible third link, and
- a flexible third joint.

Furthermore, a number of parameters must be determined experimentally:

- the exact (Denavit-Hartenberg) construction parameters (kinematic calibration),
- the motor input-output characteristics (K_t) and motor dynamics (e.g., torque ripple),
- the motor (and joint) friction ($\underline{T}_f$), and
- the rigid-robot model parameters in M, C, and $\underline{g}$.

Future research may involve the systematic evaluation of several control algorithms for rigid, and flexible joint and/or link robots. Besides encoder feedback, non-colocated actuation/sensing (using additional link sensors) and vision feedback schemes can be implemented.

Appendix A

Modeling the RRR-robot

A.1 Introduction

A model of the RRR-robot is needed for a number of different reasons: as a design tool (e.g., choosing the right servos), as a simulation model for control design, and for control synthesis and implementation. Each application imposes different requirements, and for all specialized software packages exist.

In this appendix, however, a general purpose and conceptually straightforward modeling approach is described. This approach can be easily implemented in a symbolic manipulation package, and results in an explicit model. Although the computational efficiency of this general purpose model is low, it can be used for all applications mentioned above.

First a rigid body model is derived using the Denavit-Hartenberg convention to describe the kinematics in a systematic manner. Assuming small deflections of the links, this convention can be extended to a generalized flexible link model (section A.3). Finally, in section A.4 some implementation issues are discussed.

A.2 Rigid manipulator model

The first step in modeling a manipulator is formulating the *kinematics*, the relation between the joint-angle space and the position and orientation of each arm with respect to a inertial reference frame.

Depending on the formulation of the kinematic relationships, a distinction is made between the direct kinematics problem and the inverse kinematics problem. To derive the dynamic model the direct kinematic problem is solved, i.e., the position and orientation of the end-effector is expressed as a function of the joint coordinates. The inverse problem is important to transform the desired motion of the end-effector in the workspace into the required joint motion. However, it is not dealt with here.

Based on the (direct) kinematic model, various formulations (such as Euler-Lagrange or d'Alembert) can be used to derive the *dynamics*.

A.2.1 Direct kinematics

A computational convenient method to describe the kinematic relationships is a vector-matrix description. The direct kinematic problem is then to find a transformation matrix that relates a body-attached coordinate frame to the reference coordinate frame.

To include both rotations and translations (and, if desired, scaling), a 4 × 4 homogeneous transformation matrix can be used [25]. This transformation matrix, A, maps an augmented position vector p expressed in homogeneous coordinates ($p = [p_x, p_y, p_z, 1]$) from one coordinate system to another. For robotic applications this homogeneous transformation matrix is given by:

$$A = \begin{bmatrix} 3 \times 3 \text{ rotation matrix} & & & 3 \times 1 \text{ translation vector} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

A minimum of four parameters is needed to describe this transformation: two distances, a and d, and two angles, α and q. If coordinate frames are attached to each rigid link of a manipulator, the position and orientation of frame O_i with respect to a previous frame O_{i-1} is given by:

$$A^i_{i-1}(q_i) = \begin{bmatrix} \cos(q_i) & -\cos(\alpha_i)\sin(q_i) & \sin(\alpha_i)\sin(q_i) & a_i\cos(q_i) \\ \sin(q_i) & \cos(\alpha_i)\cos(q_i) & -\sin(\alpha_i)\cos(q_i) & a_i\sin(q_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2}$$

For a kinematic chain, the coordinate transformation $T^n_0(\underline{a}, \underline{d}, \underline{\alpha}, \underline{q})$ from any link frame O_n to the base coordinate frame O_0 is given by the product of homogeneous transformation matrices, i.e.,

$$T^n_0(\underline{q}) = A^1_0 A^2_1 \ldots A^n_{n-1} \tag{A.3}$$

As a result the kinematic configuration of a rotational manipulator is completely described by three constant parameter vectors ($\underline{a}$, $\underline{d}$ and $\underline{\alpha}$) depending on the geometry of the robot, and a vector with the joint rotations $\underline{q}$. Using this transformation all link properties (such as the center of gravity and the link or motor inertia) defined in their own link coordinates, can be expressed in base coordinates. Thus in terms of the fixed inertial coordinate of the base, the position of a point on link i is given by:

$$h^i_0 = T^i_0 \, h_i \tag{A.4}$$

Denavit and Hartenberg proposed a convention to choose the link coordinate frame systematicly. For the RRR-robot this is shown in Fig. A.1. In Table A.1 are the *DH-parameters* for an ideal robot with perfect parallel or perpendicular axes.

Table A.1: Denavit-Hartenberg parameters for the RRR-robot based on the coordinate frames of Fig. A.1; l_i is the length of link i measured from O_{i-1} to O_i, d_i is the offset of each link along the z-axis needed for unconstrained rotation, and q_i is the rotation angle of motor i.

parameters link	1	2	3
$a_i \quad = D(O_{i-1}, O_i)$ along x_i	0	l_2	l_3
$d_i \quad = D(O_{i-1}, O_i)$ along z_{i-1}	l_1	d_2	d_3
$\alpha_i \quad = \angle(z_{i-1}, z_i)$ about x_i	$\frac{\pi}{2}$	0	0
$q_i \quad = \angle(x_{i-1}, x_i)$ about z_{i-1}	q_1	q_2	q_3

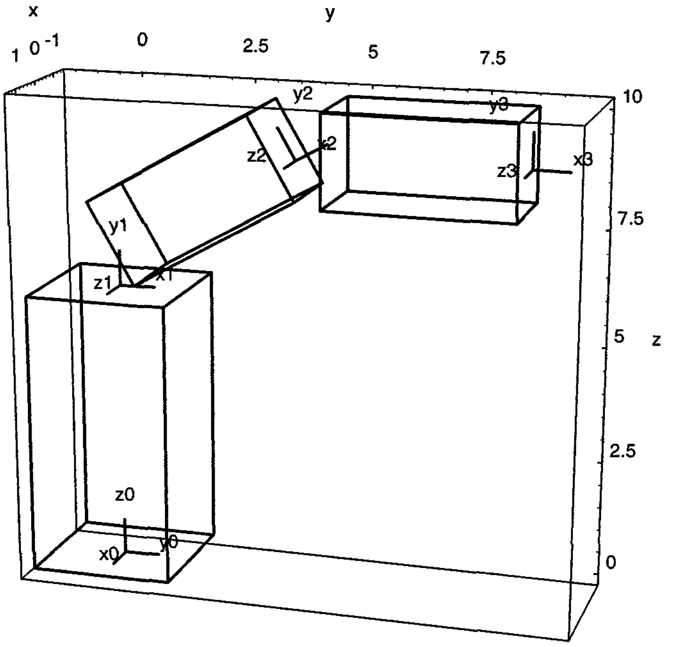


Figure A.1: Orientation of the coordinate frames: O_0 is placed at the center-base of link 1, O_i at the end of link i. The center of each servo motor (not shown) is located at O_0, O_1 and O_2 parallel to the z-axis.

A.2.2 Dynamics

Several approaches can be used to derive the dynamics, e.g., Euler-Lagrange, Newton-Euler, the recursive Euler Lagrange and the generalized d'Alembert formulations. These methods vary greatly in computational effort [25]. Computing the applied torque from the Euler-Lagrange equations is of the order $O(n^4)$, while the generalized d'Alembert formulation is of the order $O(n^3)$, and the Newton-Euler method is of the order $O(n)$ (where n is the number of DOF).

Although computationally inefficient, the Euler-Lagrange formulation is chosen here because it straightforward to compute, and from a control viewpoint results in a very convenient set of (closed-form) equations:

$$M(\underline{q})\,\underline{\ddot{q}} + C(\underline{q},\underline{\dot{q}})\,\underline{\dot{q}} + \underline{g}(\underline{q}) = \underline{\tau} \tag{A.5}$$

where, M is the *inertia matrix*, $C(\underline{q},\underline{\dot{q}})\,\underline{\dot{q}}$ accounts for the Coriolis and centrifugal torques, $\underline{g}$ are the conservative torques due to gravity, and $\underline{\tau}$ accounts for all non-conservative torques (such as control input or friction).

The *dynamic coefficients* M, C, and $\underline{g}$ can be determined from the Lagrangian $L = U_k - U_p$ (see [24, 25]),

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i; \quad i = 1 \ldots n \text{ links} \tag{A.6}$$

In the following paragraphs the expressions for the kinetic energy, U_k, and the potential energy, U_p, are addressed briefly. Finally the resulting dynamic coefficients are stated.

Kinetic Energy The kinetic energy of a point on the i-th link is

$$dU_{k_i} = \frac{1}{2}\, dm\, \mathrm{Tr}(\dot{h}_0^i\, \dot{h}_0^{iT}) \tag{A.7}$$

where dm is the differential mass, and Tr(.) the trace operator. Substituting (A.4), integrating each link over its entire mass and summing over all, n, links results in the total kinetic energy, U_k with respect to the inertial frame:

$$U_k = \sum_{i=1}^{n} \int dU_{k_i} = \sum_{i=1}^{n} \left\{ \frac{1}{2} \sum_{j=1}^{i} \sum_{k=1}^{i} \mathrm{Tr}\left\{ \frac{\partial T_0^i}{\partial q_j}\, J_i \left(\frac{\partial T_0^j}{\partial q_k} \right)^T \dot{q}_j\, \dot{q}_k \right\} \right\} \tag{A.8}$$

where J_i is the total inertia matrix of link i with respect to the base frame.

Potential Energy Expressed in the base coordinate frame the potential energy is given by

$$U_p = \sum_{i=1}^{n} U_{p_i} = \sum_{i=1}^{n} -m_i\, \underline{G}\, T_0^i\, \underline{r}_i \tag{A.9}$$

where $\underline{r}_i$ is the center of mass of link i in the link coordinate frame and $\underline{G}$ is the gravity vector.

Dynamic coefficients

$$M_{ik} = \sum_{j=\max(i,k)}^{n} \mathrm{Tr}\left\{ \frac{\partial T_0^j}{\partial q_k}\, J_j \left(\frac{\partial T_0^j}{\partial q_i} \right)^T \right\} \tag{A.10}$$

$$C_{ik} = \sum_{m=1}^{n} \sum_{j=\max(i,k,m)}^{n} \mathrm{Tr}\left\{ \frac{\partial^2 T_0^j}{\partial q_k \partial q_m}\, J_j \left(\frac{\partial T_0^j}{\partial q_i} \right)^T \right\} \tag{A.11}$$

$$g_i = \sum_{j=i}^{n} -m_j\, \underline{G} \left(\frac{\partial T_0^j}{\partial q_i} \right) \underline{r}_j \tag{A.12}$$

for $i, k, m = 1 \ldots n$ links.

A.2.3 Inertia parameters

After completion of the robot, *Microstation Modeler* was used to generate a 3D model of the robot in order to calculate the inertia tensors of each joint-link pair expressed in the appropriate link coordinates. In Table A.2, all moving components are grouped to form joints (the motors) and links with appropriate link coordinate frames.

Table A.2: Moving parts (joints and links) and the placement of the coordinate frames.

part	subcomponents	mass [kg]
Joint 1	DM60	12
Link 1 (vertical)	Rotor extension DM60, PSR-1, Housing DM30, and Stator extension DM30	20.6
Joint 2	DM30	7.5
Link 2 (green)	Rotor extension DM30, PSR-2 and Arm 2	8.9
Joint 3	DM15	5.5
Link 3	Arm 3 (not present)	-

Placement of the coordinate frames (see Fig. A.1 for the orientations):

- O_0 at the center-base of link 1.
- O_1 at he intersection of the DM60 and DM30 rotation axes.
- O_2 at the intersection of the DM15 rotation axis and the line parallel to the length axis of Arm 2, through its center of gravity.
- O_3 at the end of the line parallel to the length axis of Arm 3 through its center of gravity. Note that Arm 3 is not yet present, so O_3 is located at the end of a virtual link with length 0.3 [m].

The combined inertia tensor of Link i and Joint i with respect to frame i is given by J_i:

$$J_1 = \begin{pmatrix} 2.2334 & -2.6560\ 10^{-7} & 2.7755\ 10^{-8} & 0 \\ -2.6560\ 10^{-7} & 0.2213 & -0.01456 & -6.1425 \\ 2.7755\ 10^{-8} & -0.01456 & 2.2623 & -0.2270 \\ 0 & -6.1425 & -0.2270 & 32.9305 \end{pmatrix} \tag{A.13}$$

$$J_2 = \begin{pmatrix} 0.1921 & 1.9029\ 10^{-4} & -0.2068 & -2.8439 \\ 1.9029\ 10^{-4} & 0.7340 & 0 & 0 \\ -0.2068 & 0 & 0.6474 & -1.0222 \\ -2.8439 & 0 & -1.0222 & 16.3123 \end{pmatrix} \tag{A.14}$$

$$J_3 = \begin{pmatrix} 0.01238 & 0 & 0 & 0 \\ 0 & 0.01190 & 0 & 0 \\ 0 & 0 & 0.012 & 0 \\ 0 & 0 & 0 & 5.4763 \end{pmatrix} \tag{A.15}$$

A.3 Flexible model

Many studies of flexible manipulators have concentrated on the single-joint, single link example, e.g., [22, 38]. Link flexibility is modeled by a Bernoulli-Euler beam with a truncated number of modal coordinates.

Although, some researchers [5] stress the importance to move into more realistic 2 and 3 joint experiments where flexibility is representative of real applications or at least scaled to those applications, much less literature deals with modeling and control of these multi-link manipulators. Mainly because of the highly complicated algebraic manipulations involved.

Dynamic modeling of multi-link manipulators can be divided into two groups:

1. finite element based methods, and
2. assumed modes based models.

Finite element based methods [28, 18] can be applied to complex shaped systems. However they do not give much insight to the dynamic structure of the system.

Book [4, 6] and Cetinkunt [7, 8] developed a general Lagrangian-assumed modes based method. It uses 4 × 4 homogeneous transformation matrices to include both joint and deflection motion. The resulting dynamic model has a form similar to the previous discussed rigid manipulator model. In the following subsection the basic idea behind flexible kinematics is discussed. The implementation of flexible-arm dynamics is analogue to the rigid case.

A.3.1 Flexible-arm kinematics

The transformation from a point on a flexible arm to the base coordinates can be separated into two steps: a transformation, A, due to the joint rotation, and a transformation, E, due to the (small) link deflection:

$$h_0^i = T_0^i\, h_i = A_0^1 E_0^1\, A_1^2 E_1^2 \ldots A_{n-1}^n E_{n-1}^n\, h_i \tag{A.16}$$

The joint transformation matrix $A_{i-1}^i(\underline{q})$ is identical to (A.2).

To incorporate the deflection of the link, the approach of modal analysis is used. This approach is valid for small deflection of the link. A coordinate system $O_{e_i} = [x_i, y_i, z_i]$ is chosen at the base of the link with the x-axis coincident with the neutral axis. With η as a coordinate along this neutral axis a point on the link is given by

$$h_i(\eta) = \begin{bmatrix} \eta \\ 0 \\ 0 \\ 1 \end{bmatrix} + \sum_{j=1}^{m_i} \delta_{ij} \begin{bmatrix} x_{ij}(\eta) \\ y_{ij}(\eta) \\ z_{ij}(\eta) \\ 0 \end{bmatrix} \tag{A.17}$$

where δ_{ij} is the time varying amplitude of mode j of link i; x_{ij}, y_{ij}, and z_{ij} are the displacement components; and m_i is the number of modes of link i.

If the deflections are small so are the rotation components $[\theta_{xij}, \theta_{yij}, \theta_{zij}]$, and the transformation $E(\underline{\theta})$ is given by

$$E_{i-1}^i(\underline{\theta}) = \begin{bmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \sum_{j=1}^{m_i} \delta_{ij} \begin{bmatrix} 0 & -\theta_{zij} & \theta_{yij} & x_{ij} \\ \theta_{zij} & 0 & \theta_{xij} & y_{ij} \\ -\theta_{yij} & \theta_{xij} & 0 & z_{ij} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.18}$$

A.4 Implementation

To derive a simulation model of the robot several multi-body packages exist. Some commercial products like Adams can be integrated with CAD packages and even with Matlab/Simulink.

However, none of them is directly suitable for control purposes [39]. The output consists of a implicit model including differential algebraic equations.

For optimal flexibility Mathematica [45] was used to generate the resulting equations of motion. Implementing for example (A.10) in this symbolic manipulation package is straightforward:

```
Mik[i_,k_] :=  Sum[ Tr[
D[T[j],q[[k]]] .J[j]. Transpose[ D[T[j],q[[i]]] ]],
{j,Max[i,k],n}  ]
```

Appendix B

RRR-robot components

In this Appendix, The main features of the used components are listed. For more detailed information (e.g., contact information) refer to the RRR-robot instruction manual [41].

B.1 Dynaserv servos

The *Dynaserv* [26] motors are self-contained units containing precision ball bearings, magnetic components, and integral feedback. The rotor is placed around the stator (outer-rotor), providing direct motion to the outside of the motor. On the inside, the motor has a hollow core (diameter 25 [mm]) which is part of the rotor. With regard to the stator, only the base part is visible.

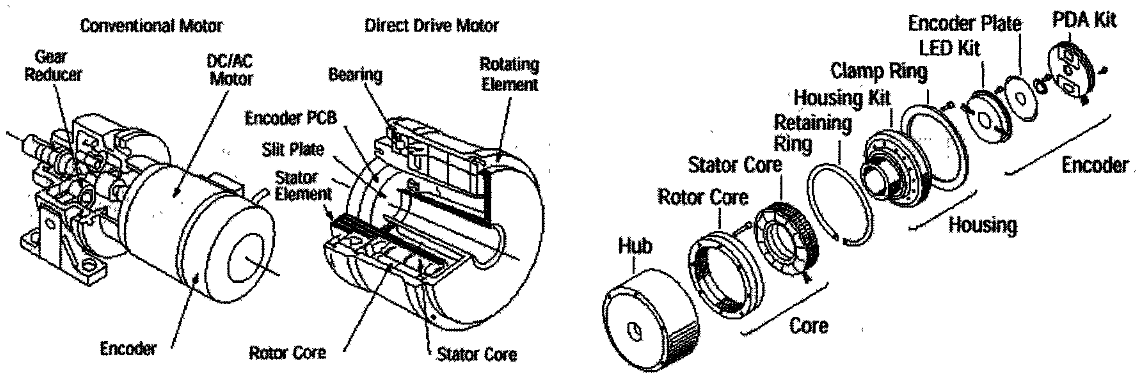


Figure B.1: On the left, Dynaserv motor v.s. a conventional motor; and on the right, an exploded view of the Dynaserv servo.

In Table B.1, the main specifications of the motors are summarized. The listed maximum torque and maximum velocity cannot be achieved simultaneously, as illustrated in Fig. B.2.

Table B.1: Main specifications of the Dynaserv motors. Source: Dynaserv manual and personal communications with Litton (marked by *).

property	unit	DM60 Motor 1	DM30 Motor 2	DM15 Motor 3
max. output torque	[Nm]	60	30	15
max. velocity	[rps]	2.4	2.4	2.4
mass	[kg]	12	7.5	5.5
inertia	[kg m^2]	0.023	0.015	0.012
allowed axial load	[N]	3 10^4 (compression), 3 10^4 (tension)		
allowed moment load	[Nm]	200 (static load), 60 (alternating load*)		
allowed velocity	[rps]	0.5	1	1.5
torque constant*	[Nm/V]	15.6	7.8	3.9
max. power	[kW]	2.2	2.0	1.6

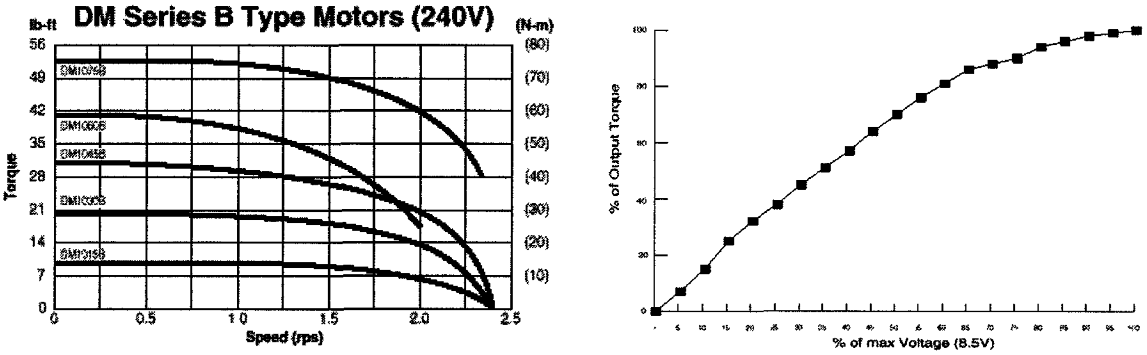


Figure B.2: On the left, the DM series motor performance (torque v.s. velocity), on the right the saturation of the motor output torque as as a function of the command voltage send to the driver (source: Compumotor [9]).

B.2 MultiQ plug-in and terminal board

The MultiQ I/O board from Quanser Consulting [10] and its terminal board are shown in Fig. B.3. Its main characteristics are summarized in Table B.2.

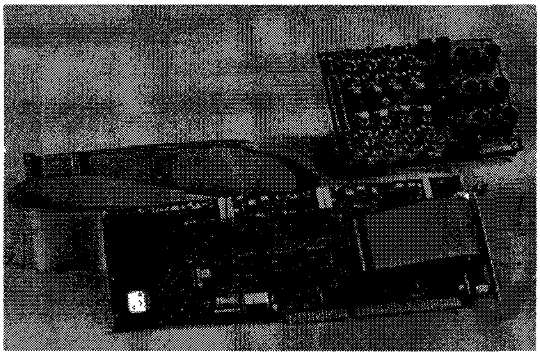


Figure B.3: MultiQ plug-in board, placed inside the PC (bottom), and terminal board (top right)

Table B.2: Main characteristics MultiQ board (I/O times measured using the separate test software supplied by Quanser Consulting).

Input/Output	Range	I/O $[\mu s]$	Noise level bounds	Remarks
6 × Encoder read	2^{24} pulses	5.5		5 [V] encoders only
8 × ADC (13 [bit])	-5...5 [V]	20	±3 LSB	Overload voltage 5.3 [V]
8 × DAC (12 [bit])	-5...5 [V]	5	±2 LSB	
8 × Digital input	GND/$\overline{\text{GND}}$	5		
8 × Digital output	0/5 [V]	2		
3 × timer	30.52 [Hz]... 2 [MHz]			

B.3 Sliprings

All slipring units were supplied by Litton [26].

B.3.1 Power sliprings

Power slipring 1 (PSR-1) is made up out of two off-the-shelve sliprings, the SM140-6 and the SM140-2. Power slipring 2 is a modified (M) version of the SM204 with 12 circuits. For each motor phase, three parallel circuits are used.

Table B.3: Main specifications of power sliprings PSR-1 and PSR-2.

property	PSR-1 SM140-6 and -2	PSR-2 M-SM204-12
number of circuits	8	12
maximum current/circuit	15 [A]	6 [A]
maximum total load	15 [A] × 50 [V]/circuit	-
maximum velocity	250 [rpm]	150 [rpm]
circuit resistance	< 0.1 [Ohm]	-

Appendix C

Budget (in Dutch)

In de onderstaande tabel is weergegeven wat de stand van zaken is met betrekking tot de gerealiseerde en voorgenomen uitgaven per eind maart 1998. Voor een bedrag van NLG 94.947,15 zijn rekeningen binnen. Een afrekening van de GTD voor de vervaardiging van afscherming van de servoversterkers i.v.m. veiligheid moet nog binnen komen (geschat op NLG 15K).

Het meetsysteem voor de eindpositie moet nog gerealiseerd worden en zou dus ca. NLG 45K mogen kosten.

Table C.1: Overzicht van gerealiseerde, voorgenomen en begrote uitgaven per eind maart 1998. De posten zij ontleend aan de oorspronkelijke Stimuleringsfonds aanvraag [42]. Alle bedragen zijn in NLG en inclusief BTW. Bedragen gemarkeerd met ± zijn nog niet definitief.

	Post	gerealiseerd	gerealiseerd	verwacht	subtotaal	begroot	
		fase 1	fase 2		rij	fase 1	fase 2
1.	a. detail ontwerp, constructie,						
	in bedrijf stellen					13.000	4.000
	b. constructie, frame, ophang.					4.000	1.000
	kolom subtotaal	14.646,00	8.434,00	± 15.000,-	± 38080,00	17.000	5.000
2.	a. Motoren, servoversterkers,						
	hoekmetingen	30.540,97	3.459,20			30.000	15.000
	b. vermogenssleepringen	9.361,23					
	dataoverdrachtssysteem	9.685,52				10.000	5.000
	kolom subtotaal	49.586,75	3.459,20	0	53.046,74	40.000	20.000
3.	Eindpositie meting	0	0	± 45.000,-	± 45.000,-	0	50.000
4.	Besturingssysteem:						
	a. PC	6.696,33					
	b. DAQ-board	3.540,88					
	c. printer	611,51					
	kolom subtotaal	10.848,72	0	0	10.848,72	18.000	0
5.	Software:						
	a. Matlab	5.934,02					
	b. Watcom C		699,13				
	kolom subtotaal	5.934,02	699,13		6.633,15	5.000	0
		fase 1	fase 2	verwacht			
	Totaal	81.009,49	12.593,33	± 60.000,-	± **154.947,15**	80.000	75.000

B.3.2 Signal sliprings

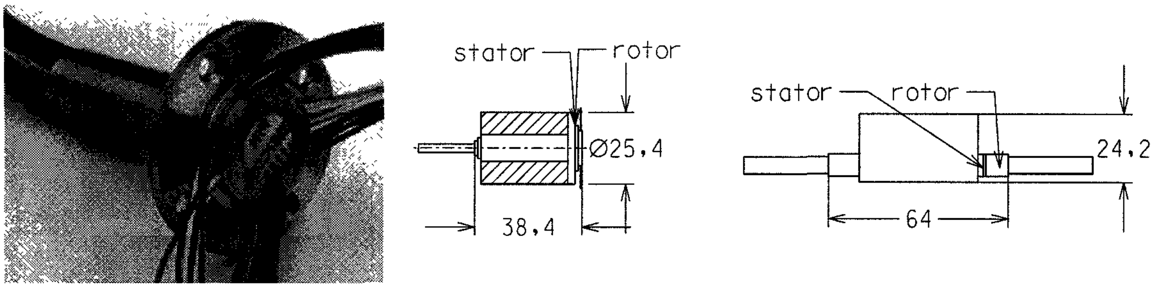


Figure B.4: Signal sliprings: on the left, a photograph of the original AC-267 slipring; in the middle, the AC-267 inside a hollow mounting cylinder for gluing it inside the motors (used for SSR-1 and SSR-2); and on the right, SSR-3.

Table B.4: Main specifications of the signal sliprings (see Fig. B.4).

property	SSR-1 and 2	SSR-3
	Capsule AC-267 modified	Litton 12-ring capsule
number of circuits	36	12
maximum current/circuit	1.2 [A]	1 [A]
maximum total load	25 [A]	50 [W]
maximum velocity	100 [rpm]	120 [rpm]
total friction torque	0.0054 [Nm]	0.0023 [Nm]

Appendix D

Time schedule

Main activities and decision moments:

January 1996

- Start of the RRR-robot design project.

April 1996

- Implementation of a simple rigid-robot model in *Mathematica.*
- Evaluation of the original specifications.

July 1996

- Decision to order a PC-based control system: PC, DAQ board and software.

October 1996

- Selection of direct-drive servos using a refined rigid-robot model.
- Start concept design of the manipulator frame.

November 1996

- Decision to use sliprings for both power and data transfer.

December 1996

- Decision to order 3 servos.
- Decision to order 5 sliprings units for power and data transfer.

February 1997

- Order of slipring units for data transfer was "lost", and had to be reiterated.
- Decision to skip the detail design by the CTD and start of manufacturing of frame parts based on the concept design.

March/April 1997

- Servos received and start of individual servo testing.
- Decision to start the selection of a measurement system for the Cartesian position of the end-effector.

May 1997

- Control problems with servo testing solved.

June 1997

- Signal transfer sliprings received.
- Start of the frame assembly.

July-August 1997

- Testing and further assembly of frame and Signal-Transfer System.
- Decision to order dynamics brakes.

September 1997

- Integration and evaluation of the Control System.

October 1997

- First tests with controlling all three degrees of freedom.

January/February 1998

- Decision to postpone the design, and realization of the 3D measurement system.

March/April 1998

- Completion of the design assignment:
  - 3D measurement systems for robot manipulators, WFW report 98.011
  - RRR-robot instruction manual, WFW report 98.012, and
  - this report.

Progress and decisions were discussed with:

- Prof. dr. ir. J.J. Kok
- Dr. ir. A.G. de Jager
- ir. J.P.A. Banens
- ir. L. Kodde
- ir. A.M. van Beek

Appendix E

Terminology

When applicable, parts are numbered from the base to the end-effector, i.e., the stationary base is link 0, the rotating octagonal housing is the top of link 1, and the end-effector is link 3.

Table E.1: Terminology

Short	Synonym	Description
A/D	AD(C)	Analog-to-Digital (Converter)
AC		Alternating Current
BDC	Brushless DC	Brushless Direct Current (motor)
BE60	Brake 1	Electro-mechanical brake BE1060B
BE30	Brake 2	Electro-mechanical brake BE1030B
CTD	GTD	"Centrale/Gemeenschappelijke Technische Dienst"
D/A	DA(C)	Digital-to-Analog (Converter)
DC		Direct Current
DM60	Motor/servo 1	Dynaserv Motor DM1060B50*1, max. torque 60 [Nm]
DM30	Motor/servo 2	Dynaserv Motor DM1030B50*1, max. torque 30 [Nm]
DM15	Motor/servo 3	Dynaserv Motor DM1015B50*1, max. torque 15 [Nm]
Driver cabinet		Shielded cabinet for drivers, brakes and line filters
MultiQ	PC plug-in board	board with ADC, DAC, encoder inputs, and digital I/O
PSR-1	Power Slipring 1	SM140-6 and -2 combined
PSR-2	Power Slipring 2	M-SM204-12
RTW	Real Time Workshop	Matlab toolbox to generate C-code
SSR-1	Signal Slipring 1	Capsule AC-267 modified
SSR-2	Signal Slipring 2	Capsule AC-267 modified
SSR-3	Signal Slipring 3	Litton 12-ring capsule
SD60	Driver 1	Driver SD1060B52-2, driving DM60
SD30	Driver 2	Driver SD1030B52-2, driving DM30
SD15	Driver 3	Driver SD1015B52-2, driving DM15
Terminal board		Board with connectors to the internal MultiQ board
Watcom	Watcom C/C++	Watcom C/C++ compiler version 10.6
Wincon	WinCon V2.0	Software to interact with real-time running C-code

Bibliography

[1] Gerry B. Andeen. *Robot design handbook.* McGraw-Hill, London, 1988.

[2] Haruhiko Asada and Kamal Youcef-Toumi. *Direct-drive robots.* MIT press, London, 1987.

[3] Open DeviceNet Vendor Association. DeviceNet. http://www.odva.org, 1998.

[4] W. J. Book. Recursive Lagrangian dynamics of flexible manipulators. *International Journal of Robotics Research*, vol. 3(no. 3):87–101, 1984.

[5] W. J. Book. Modeling, design, and control of flexible manipulator arms: A tutorial review. In *Proceedings of the 29th Conference on Decision and Control*, volume 3, pages 500–506. IEEE, 1990.

[6] W. J. Book. Controlled motion in an elastic world. *Dynamic Systems, Measurements, and Control*, vol. 115:252–261, 1993.

[7] S. Cetinkunt and W. J. Book. Symbolic modeling of flexible manipulators. In *Proceedings of 1987 IEEE International Conference on Robotics and Automation*, volume 3, pages 2074–2080, 1987.

[8] S. Cetinkunt and W. J. Book. Symbolic modeling and dynamic simulation of robotic manipulators with compliant links and joints. *Robotics & Computer-Integrated Manufacturing*, vol. 5(no. 4):301–310, 1989.

[9] Compumotor. Electronic motion control. http://www.compumotor.com, 1998.

[10] Quanser consulting. Wincon and MultiQ. http://www.quanser.com, 1998.

[11] N. Costescu and D. Dawson. QMotor 2.0 - A PC based real-time multitasking graphical control environment. *Submitted to IEEE Control Systems*, 1997. Available at http://ece.clemson.edu/crb/research/qmotor/qmotor2.pdf.

[12] B. de Jager. *Practical evaluation of robust control for a class of nonlinear mechanical dynamic systems.* PhD thesis, Eindhoven University of Technology, 1992.

[13] E. O. Doebelin. *Measurement systems: Application and design.* McGraw-Hill, London, 1990.

[14] dSpace. DS1102 DSP controller board. http://www.dspace.de, 1997.

[15] J. R. Gyorki. Making connections with factory-floor networks. *Machine Design*, 6:56–61, 1996.

[16] Herbert Hanselmann. Guest editorial: Introduction to the special issue on digital signal proessors in control. In *IEEE Transactions on Control Systems Technology*, volume 5, pages 277–278, 1994.

[17] Inc HD Systems. Harmonic drives (actuators). http://www.hdsystemsinc.com, 1998.

[18] S. K. Ider and F. M. L. Amirouche. Nonlinear modeling of flexible multibody systems dynamics subjected to variable constraints. *Journal of Applied Mathematics*, vol. 56: 444–450, 1989.

[19] The MathWorks Inc. Matlab, simulink and the real-time workshop. http://www.mathworks.com, 1998.

[20] Texas Instruments. *TMS320C4X user's guide*, 1993.

[21] Gerald Jacob. Revisiting data acquisition configurations. *EE-Evaluation Engineering*, pages 22–28, 1987.

[22] T. R. Kane, R. R. Ryan, and A. K. BanerJee. Dynamics of a canterlever beam attached to a moving base. *Journal of Guidance*, vol. 10(no. 2):139–151, 1987.

[23] I. M. M. Lammerts, F. E. Veldpaus, J. J. Kok, and M. J. G. Van de Molegraft. Adaptive computed torque computed reference control of flexible robots. In *ASME 1993 Winter annual meeting*, New Orleans, Louisiana, USA, December 1993.

[24] Ivonne M. M. Lammerts. *Adaptive computed reference computed torque control of flexible manipulators.* PhD thesis, Eindhoven University of Technology, 1993.

[25] C. S. G. Lee. Robots kinematics, dynamics and control. *IEEE Computer*, vol. 15(no. 12): 62–80, 1982.

[26] Litton. Dynaserv sevos, μFORS-36 gyroscope, and various sliprings. http://www.litton.com, 1997. No online datasheet.

[27] QNX Software Systems Ltd. QNX realtime OS. http://www.qnx.com, 1997.

[28] F. Melzer. Symbolic computations in flexible multibody systems. *Nonlinear Dynamics*, (no. 9):147–163, 1996.

[29] NSK. Megatorque stepper motors. http://www.nsk.com, 1998. No online datasheet.

[30] Profibus. Process field bus. http://www.profibus.com, 1998.

[31] T. M. Rosenberg. Industrial networks: The options and rewards are many. *I&CS*, 3: 61–66, March 1996.

[32] P.C.J.N. Rosielle and E.A.G. Reker. *Constructieprincipes 1: bedoeld voor het nauwkeurig bewegen en positioneren.* Technische Universiteit Eindhoven, Vakgroep Precision Engineering, Dictaat 4007, 1997.

[33] Schleifring. Contactless optical and electrical transmission systems. http://www.schleifring.de, 1998.

[34] Lorenzo Sciavicco and Bruno Siciliano. *Modeling and control of robot manipulators.* McGraw-Hill series in electrical engineering. McGraw-Hill, London, 1996.

[35] Scientific Technologies Inc. Wireless data communication. http://www.sti.com, 1998.

[36] S. E. Skolnik. Transport of data in the RRR-testrobot. Technical report, WFW 95.147, Eindhoven University of Technology, 1995.

[37] Spectron. IA-Spox. http://www.spectron.com, 1998.

[38] W. W. Spong. Modeling and control of elastic joint robots. *Dynamic systems, Measurements, and Control*, vol. 109:310–319, 1987.

[39] Michael Valasek. Efficient modelling of multibody systems with respect to control. In *Notes of the workshop at the IEEE International Conference on Control Applications, Hartford, CT*, 1997.

[40] Bert van Beek. 3D measurement systems for robot manipulators. Technical report, WFW 98.011, Eindhoven University of Technology, 1998.

[41] Bert van Beek. RRR-robot: Instruction manual. Technical report, WFW 98.012, Eindhoven University of Technology, 1998.

[42] P. P. J. Van den Bosch, M. L. J. Hautus, J. J. Kok, and H. A. Preisig. Stimuleringsfonds aanvraag voor RRR-robot. Aanvraag TUE-participatiesubsidie t.b.v. onderzoeksgroepen aan de TUE die deelnemen aan de onderzoeksschool DISC, Technische Universiteit Eindhoven, 1995.

[43] G.-W. van der Linden. *High performance robust manipulator motion control: Theory, tools and experimental implementation.* PhD thesis, Delft University of Technology, 1997.

[44] A. Vandenput. *Electromechanica en vermogenselectronica.* Technische Universiteit Eindhoven, Vakgroep EMV, dictaat 5765, 1994.

[45] S. Wolfram. *Mathematica, a System for Doing Mathematics by Computer.* Addison-Wesley Publishing Company, 1993.