

Design of a New Joint Mechanism and a Simulator for a Climbing Robot

Martin Wiig

Master of Science in Engineering Cybernetics

Submission date: June 2007

Supervisor: Geir Mathisen, ITK

Co-supervisor: Pål Liljebäck, Sintef

Problem Description

This master thesis will investigate a new joint mechanism for flexible light-weight robotic manipulators. This joint mechanism enables one single rotary motor (either electric or hydraulic) to power all the joints in one arm, or even all the joints of all the arms of a climbing robot. The thesis will also develop a simulation program that will enable important design parameters of a climbing robot to be simulated.

1. Joint mechanism

A new joint mechanism that will potentially reduce the weight of the arm has been proposed. This design will enable one actuator to drive several joints, and can potentially reduce the weight of the robot arm significantly. The candidate should study this proposed design. The study should consist of:

- A study of existing solutions. The existing solutions should be assessed with respect to this assignment, and parts of the solution that are relevant for the new flexible joint design should be pointed out.
- Identification of challenges of the proposed design
- Delve deeper into one or more of these challenges and propose and evaluate possible solutions
- Evaluate the suitability of this joint mechanism for a climbing robot

2. Simulation

A detailed design of a climbing robot will include the choice which degrees of freedom the robot needs, placement of weights such as actuators, choice of materials, movement strategy etc. To choose these parameters, it is useful to study simulations of the robot. To facilitate this, the candidate should develop a simulation program.

The design of the simulation program should start with an exploration of existing solutions for simulation of robots. The existing solutions should be assessed with respect to this assignment. Also, existing work on modelling robots should be explored.

The candidate will then have to develop a mathematical model of the robot. The model should be able to take in different parameters, such as material density or points of weight representing actuators, instrumentation etc. This will make it possible to use the model as a design tool. The model should be used in a simulation, where it should be possible to control the robot and measure joint torques, velocities and positions.

Assignment given: 08. January 2007

Supervisor: Geir Mathisen, ITK

Abstract

This report covers the development of a new joint mechanism and a robotic simulator. These are intended to be used in the further development of a climbing robot that has been previously worked on. The climbing robot is briefly described in this report. It is a robot with four arms that each has 6 joints with 1 degree of freedom each. At the end of each arm is a gripper that will serve as the robots main tool of interaction with its immediate environment, for instance a ladder the robot is climbing in. The weight of this robot is estimated and used to find torque requirements on the new joint mechanism.

A lightweight joint mechanism is of interest not only for a climbing robot, but for many if not all other kinds of robots as well. Such a joint will not need to be as strong as its heavier counterparts, as less power is required to support the joint itself. The new joint mechanism is an attempt of creating a joint that is lighter than traditional joints. The main idea is to have a single powerful actuator driving several joints. The joints are mechanically powered by a rotating shaft that runs through all of them. Each joint uses a clutching mechanism to connect to the shaft and thus transfer torque. the clutching mechanism has been explored in detail, and a suitable clutch has been selected. This clutch is then used in four design proposals. The last of these proposals is assembled into a prototype, and the prototype is qualitatively tested. The tests demonstrate the concept, but it is not yet shown that this joint mechanism will be lighter than a traditional joint mechanism.

The simulator is developed to be a design tool for further development of the climbing robot. A list of specifications for the simulator is presented, however the simulator is not able to meet all these requirements at the current stage of development. The remaining work on the simulator is discussed, as well as an evaluation of the software as it is at the time of writing. The report describes the kinematic modeling that is used to represent the translational and rotational relationship between the different elements of the robot. The kinematic relationships is then put into a mathematical representation of the robot which represents the robots as a chain of elements. Each element has several properties, among which are its relationship to other elements and a graphical representation of the element. The elements, and thus the robot, is represented graphically in a virtual reality environment. The simulator allows the user to specify several parameters of the robot, such as number and dimensions of the joints and the dimensions of the body. The user is also able to control the velocity of each joint.

Both the simulator and the joint has been developed to be general enough for use on other kinds of robots, with small modifications in the case of the simulator.

Preface

This Master's thesis concludes my studies at the Norwegian University of Science and Technology. There are many people that have, in ways great and small, helped me both in the work on this thesis and other parts of my studies.

First of all I would like to thank my supervisor Pål Liljebäck at Sintef for his support and constructive feedback. As with the project work last semester, his undying enthusiasm and perpetual good mood has given me much inspiration and motivation. As has his sometimes disturbingly late working hours.

The report has included work on actual hardware in the real world, a subject much feared by the more theoretical of the engineering kind. This fear has proved to be unfounded, much thanks to Terje Haugen and Hans Jørgen Berntsen at the workshop of the Department of engineering here at NTNU. They have been most service-minded, and their patience with what I'm sure is an endless supply of nagging students is astonishing.

I would also like to thank my friends at the office: Martin (the other one), Stig Are, John, Frode and Einar. These guys have made my life significantly easier and more enjoyable while at work, and have shown a great interest in my thesis. Their inputs have always been appreciated, and the additional benefit of having some like-minded fellows to take a night off with even more so.

Lastly, I take the liberty of thanking my girlfriend Marie, even though this might be somewhat of a cliché. She has been very patient in listening to my endless descriptions of the tiniest detail concerning robotic simulations, while still managing to show interest in my work.

I sincerely hope that the work in this report will be carried further, and that both the joint and the simulator will be of help in the development of a climbing robot. It will be very exciting to see what Sintef is able to make of it.

Martin Wiig

Contents

1	Introduction and motivation	1
1.1	Joint Mechanism	1
1.2	Simulator	3
2	The climbing robot	5
2.1	Description	5
2.2	Weight estimation	7
2.2.1	Weight of the joint	8
2.2.2	Weight of the arm	9
2.2.3	Weight of the body	9
2.3	Torque requirements	10
2.3.1	The DC motor	12
3	Joint mechanism	13
3.1	General issues of joint design	13
3.1.1	Joint Components	13
3.1.2	Joint analysis and control	15
3.1.3	Examples of existing joint designs	15
3.2	Challenges of the proposed design	17
3.3	The clutching mechanism	17
3.3.1	The SO15 clutch	19
3.4	Joint design 1	20
3.5	Joint design 2	23
3.6	Joint design 3	26
3.7	Joint design 4	28
3.8	Testing of the prototype	30
3.8.1	Motor and main shaft test	32
3.8.2	Worm gear back-drivability	34
3.8.3	Clutch functionality.	34
3.8.4	Potentiometer	36
3.8.5	Maximum and minimum rotation angle	36
4	Simulator	38
4.1	Simulator specifications	38
4.2	Modeling of robot kinematics: The Denavit-Hartenberg convention	40
4.3	Representation of the robot	42
4.3.1	Graphical representation	42
4.3.2	A mathematical chain representation of the robot	44
4.3.3	Chain elements used in the simulator	45
4.3.4	Chain structure	47
4.4	Simulator description	48

4.4.1	Running the simulator	49
4.4.2	Unfulfilled specifications	52
4.5	Software structure	52
4.5.1	Alphabetic list and description of files	54
5	Evaluation	56
5.1	Evaluation of the joint design	56
5.2	Evaluation of the simulator	57
6	Further Work	60
6.1	Further work on the joint	60
6.2	Further work on the simulator	61
7	Conclusion	65
7.1	Joint	65
7.2	Simulator	66
	Appendix	70
A	The SO series of electromagnetic friction clutches	70
B	The DCB-2 wrap spring clutch	73
C	The HKD series of magnetic clutches	75
D	Digital Appendix	76

1 Introduction and motivation

This report will look at a new kind of robotic joint mechanism that can be used in the climbing robot of [28]. Challenges of the new joint mechanism will be looked at, and some of them will be explored in detail in the design of a joint prototype. The report will also cover the development of a simulator that can simulate the climbing robot. The required mathematical theory of the simulator will be explained, and the developed software will be described and evaluated. This section contains the introduction and motivation to each of these two parts. Since both subjects covered by this report is intended to be used on the climbing robot, a description of this robot is given in Section 2.

1.1 Joint Mechanism

When designing a robot, it is often important to keep the weight of the robot as low as possible. This will reduce the power consumption of the robot and will also reduce the need for large, expensive and heavy actuators. Since a climbing robot will continuously be lifting itself, low weight is particularly important for this class of robots.

One way to lower the weight of a robot is to reduce the number of actuators, as actuators often constitute a large part of the total weight. As the robot becomes lighter the remaining actuators do not need to lift as much, and can be replaced with lighter and cheaper ones. However, less actuators may also result in fewer degrees of freedom (DOF), and thus in a less flexible robot. A compromise must often be made between weight and flexibility. In the design proposed in [28], the robot has four arms with 6 DOF each. This robot needs to be quite flexible, and it is not desirable to cut down on the degrees of freedom.

To counter this, a new joint mechanism has been proposed. The idea is to have a powerful actuator that drives several joints. The actuator can for instance be placed in the body of the robot. Mechanical power is transferred to each of the joints via a rotating shaft that runs through the arm. To allow each joint to move separately from the others, they will each have small clutch (or clutch-like device) that controls the transfer of power from the rotating shaft to the joint, thus serving as a local actuator. It is possible that such a mechanism would make the overall weight of a robot with several joints less than if each joint was equipped with one or more DC motor.

An example of how the joint can be envisioned is displayed in Figure 1.1. The numbered parts are:

1. *Main shaft* - this is the rotating shaft that is driven by the main actuator.
2. *Bevel gears* - these gears enables the shaft to translate its rotation through the joint independently of the joint angle.
3. *Clutch* - the clutch (or clutch-like device) controls how much torque is transferred from the rotating shaft to the joint. There are one clutch for each direction of movement of the joint (clockwise and anti-clockwise).

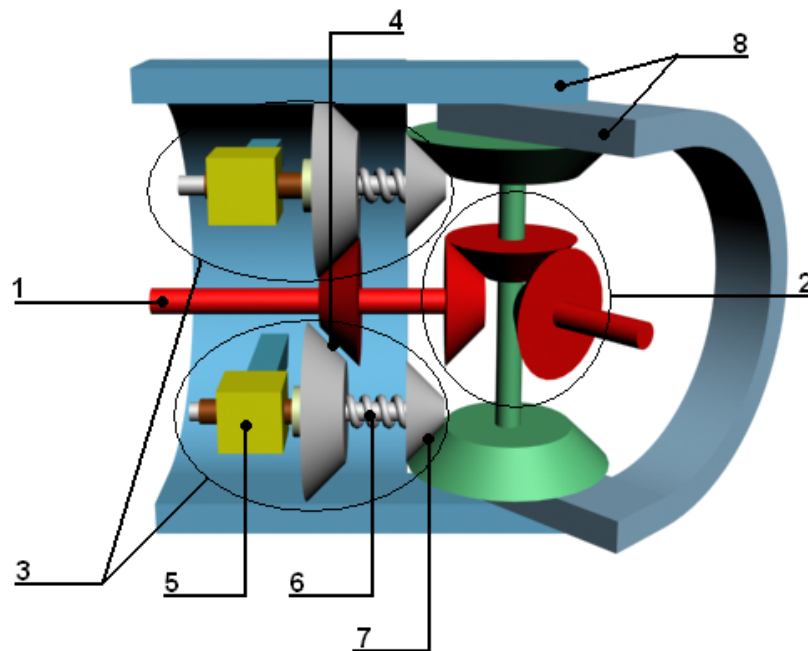


Figure 1.1: Exemplification of the proposed joint design

4. *Friction plates* - transmission of torque happens through friction plates. The harder the friction plates are pressed against each other, the more torque is transmitted.
5. *Solenoid* - the small actuator pushing on the clutch is envisioned to be a solenoid in this exemplification.
6. *Spring* - when the solenoid is no longer pushing the friction plates together, this spring will push them away from each other.
7. *Gears* - these are the bevel gears that move the joint itself.
8. *Surface* - these are the surfaces of the two adjacent links that are connected by the joint.

The design of such a joint mechanism can be very beneficial, not only to climbing robots but to other kinds of robots (such as snake robots) as well. Section 3 of the paper will delve deeper into the concept. It will start out by exploring what solutions exist today in Section 3.1, to make a foundation for the investigation of the proposed design. The main challenges of the proposed design are then pointed out in Section 3.2. These are

- The clutching mechanism
- Transmission of power
- Braking

- Weight
- Modeling, simulation and control

The clutching mechanism is treated in detail in Section 3.3, where three clutches are evaluated and one clutch, the SO15 clutch from Inertia Dynamics[10], was selected as the best alternative for the joint design.

Sections 3.4 - 3.7 covers 4 stages of the design process for creating a prototype of the joint. In these sections, both transmission of power and braking are solved in different ways. The fourth and final design has been assembled to a prototype and tested in Section 3.8. The tests demonstrated the concept of using clutches as local actuators and was evaluated along with the joint design in Section 5.1. Some areas where further work on the joint design and prototype might be beneficial are pointed out in Section 6.1.

1.2 Simulator

While simulation never will remove the need for real world testing completely, a thorough evaluation of the robot in a simulated environment is will make the overall testing task considerably easier. The effort of testing robot behavior and specifications in the real world is much higher than running the equivalent experiments in a simulated environment. Further, the starting conditions of a simulated environment can be accurately re-created, thus giving the exact same starting conditions when testing different motion planning algorithms or configurations. This opens the possibility of automatically creating and running a large number of randomized test cases for evaluation of a proposed design [5]. Alternatively, a set of test cases can be made that evaluates several configurations or designs of a robot. In addition, it is possible to efficiently train adaptive algorithms, such as the resolved motion adaptive control mentioned in [28] and [8], and other algorithms based on learning such a neural networks in simulations[5]. Efficient training of such algorithms are, of course, dependent of the accuracy of the kinematic and dynamic models used in the simulator.

Many simulation packages are restricted to specific types of robots, such as industrial manipulators. Some commercial software include packages for several types of robots, but prices are quite high and they tend to have a very restricted flexibility when it comes to new robot models[21] (such as the climbing robot developed in [28] that is to be simulated in this paper). This gives a clear need for a more general simulation tool that eases the task of modeling and simulation of robots with different geometric, kinematic and dynamic parameters. Such a simulation tool would be very useful in the further design of the climbing robot, since it would allow the user to experiment with parameters such as degrees of freedom, joint strength and dimensions, sensor placement etc. An interesting example of such a simulation tool was developed in [21], and this is used as a basis for the simulator developed in this paper.

It is possible to equip the virtual robot with perfect sensors and perfect control, or to add specified models of sensor and control errors. This would enable testing of control strategies under different conditions, and various error sources could be determined[11]. Such modular design of the simulation software also makes it easy to evaluate different sensors and actuators,

and can for instance be useful in the design of a new joint mechanism such as the one developed in this report.

The simulator developed for this report is described in Section 4. It is a purely kinematic simulator, with no dynamic models involved. The main task of the simulator at the current level of development is visualization of robot designs, so that it can be a design tool for further design on the climbing robot. It can also be used to test different high-level control algorithms for the arms. The simulator has been developed to be modular and general enough for dynamic simulations and more advanced high and low level control algorithms to be relatively easy to implement.

The specifications for the simulator are given in Section 4.1. Not all of these specifications are fulfilled by the simulator at the current stage of development. In particular, fulfilling the requirements regarding simulation of dynamics, environment and the ability of the robot to move around is left for further work. This is discussed in sections 5.2 and 6.2.

An overview of the graphical and mathematical representation of the robot is given in Section 4.3. The graphical representation uses the Virtual Reality Modeling Language which is supported by the Virtual reality toolbox of MATLAB[26]. Mathematically the robot is represented as a chain (or an array in MATLAB) of elements, where each element represents a joint, a base or an end effector. Each element contains information about its:

- Name
- Relationship to other elements in the chain
- Limitations on joint movement (if applicable)
- Type (base, joint or end effector)
- Graphical representation
- Dimensions
- Position and orientation

These representations are brought together with the kinematic model by the software. A brief overview of the software is given in Section 4.5.

2 The climbing robot

This section gives a brief overview of the climbing robot designed in [28]. Since the joint mechanism of Section 3 and the simulator of Section 4 both are developed for this robot, it is of interest to describe its constraints and requirements. The weight of the different parts of the robot are estimated in this section, and these weights are used to produce torque requirements on the clutching mechanism and on the DC motor that will actuate the robot.

2.1 Description

The purpose of the climbing robot described in [28] was to have the ability bring instruments for inspection in areas that requires climbing, but where human presence for some reason (such as safety) is unwanted. One such area that was particularly explored was the flare tower of an oil platform. If a flare tower is inspected today, it was argued, the platform must be shut down. This is a very expensive operation, and thus this is an area where there could be use for a climbing inspection robot.

In order to climb a flare tower, it was concluded that the robot should be able to transverse:

- Areas where humans can climb by getting foot- and hand-hold on each step (e.g. ladders)
- Stairs
- Horizontal surfaces

A set of physical constraints on the robot that would allow it to transverse these areas was found and is here given again in Table 2.1. The general requirements of the robot were summed up as:

- Four arms with grippers.
- As few actuators as possible.
- The ability to turn in the roll pitch and yaw axis.
- The ability to carry enough sensors for inspection.
- The sensors must be robust and waterproof.
- Communication with the operator.
- Enough power to climb and to power the sensors.
- Safety measures so that it does not fall down in case of power failure.

In addition, since the design of the arm was of such importance to the overall design of the robot, the requirement of the arms were stated:

- The arm must have a robust gripper at the tip.

Table 2.1: Physical constraints on the robot

Feature	Constraint
Retracted length	Between 50cm and 200cm
Extended length	Between 100cm and 250cm
Width	Not more than 70cm, extendable by at least 50cm
Thickness	Not more than 40cm
Grasping diameter	From 0 to 15cm
Weight	As low as possible, not above 100kg

- The arm must be at least 50cm long.
- The arm must be able to lift at least half the weight of the robot.
- The arm must have enough degrees of freedom to control the position and orientation of the gripper sufficiently.
- The arm must have enough degrees of freedom to move the robot in a desired direction during climbing or walking.
- The arm must not have too many degrees of freedom.
- The arms must be symmetrical to each other.
- The gripper must be able to carry at least half the weight of the robot.
- The gripper must be able to remain locked without power.
- The gripper must be able to be used as feet.
- The control system must be sufficiently accurate and not too complex.
- The user must be able to control the arm either by direct Cartesian control or by sending predefined procedures.

Figure 2.1 shows one of the preliminary design proposals from [28]. It shows the robot with four arms with 7 degrees of freedom each. The arms are anthropomorphic (i.e. humanlike, the arm has a shoulder joint and an elbow joint [3]). It was decided that the arms should be anthropomorphic to minimize the intrusion into the workspace, but that there was no need for kinematic redundancy so that 6 degrees of freedom was enough.

When estimating weights and joint torque requirements in this paper, the arm will be assumed to consist of 6 joints with no links between them (and with a gripper at the end of the arm). While this is not an anthropomorphic arm, it can be argued that as long as there are no static links between the joints, it still minimizes the intrusion into the workspace. To ensure that the arm fulfills the requirement of being at least 50cm long, it must be verified that the joints are large enough to give this length. It is difficult to estimate the size of the joint before the clutch is

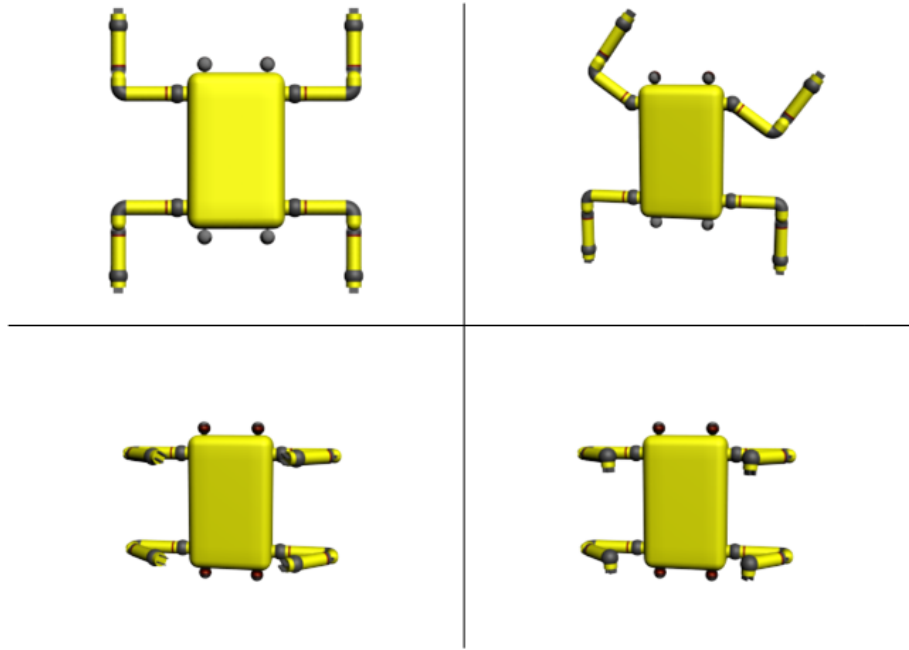


Figure 2.1: Preliminary design of the climbing robot from [28]

chosen. From the sizes of the clutches explored in appendices A, B and C however, it can seem reasonable to assume that the joint will be approximately the same size as the joint designed in [2], or about 10 cm in diameter and length. For simplicity, the joint will be assumed to be roughly cubic with sides 10 cm long and the rotation axis at the end. The arm will then be 60cm long, plus the length of the gripper.

The body of the robot will contain the DC motor, a motherboard with an CPU for control purposes and shafts going from the DC motor and to each of the arms. The size of the body will much depend on the size of the DC motor and of the electronics. The most powerful brushless DC servomotor of the Faulhaber Group [7], the 4490 048 BS, is 11.5cm high and has a diameter of 4.4cm. To ensure that the body is large enough to contain both the DC motor and the motherboard, it is estimated that it will be 20cm wide, 20cm thick and 30cm high. A body this size can also easily fit the arms if they are approximately 10cm thick. A schematic figure of this design is shown in Figure 2.3.

2.2 Weight estimation

In this section the weight of the robot and of the different parts of the robot is estimated. These estimations will in section 2.3 be used to specify a torque requirement of the clutches, as a function of the weight of the clutches the DC motor driving the main shaft.

2.2.1 Weight of the joint

The arms of the robot will have 6 joints and a gripper as described in Section 2.1. For simplicity in the weight estimation, the gripper is assumed to weigh the same as a normal joint. The parts of the joint and their estimated weight are given Table 2.2. These parts come from the design in Section 3. A more detailed derivation of the weight estimates is given below.

Table 2.2: Joint parts and weight estimation

Part	Number	Estimated weight
Clutch	2	m_{clutch}
Brake	1	m_{clutch}
Shell and supporting structures	1	300g
Electronics	1	50g
Gearing	3	150g
Shaft	1	20g
Total		$3m_{\text{clutch}} + 520\text{g}$

Each of the clutches will weigh m_{clutch} , which will be clear when the clutch is decided. Since the brake must be (at least) as powerful as the clutch, it is estimated to weigh the same as the clutch, i.e. m_{clutch} .

If the shell is made of a 1mm thick layer of aluminum in the shape of a box without top and bottom, shown in Figure 2.2, then the weight can be estimated as

$$m_{\text{shell,joint}} = \rho_{\text{aluminum}}(l \times h_e \times w_e - l \times h_i \times w_i) \quad (2.1)$$

where ρ_{aluminum} is the density of aluminum (2.7g/cm^3), l is the length of the joint (10cm), h_e and h_i is the height of the exterior and the interior of the shell (10cm and 9.8cm), and w_e and w_i is the width of the exterior and interior of the shell (10cm and 9.8cm). This gives an estimated weight of 107g. To include supporting structures for the clutches, gearing, and shaft the weight estimation is increased to 300g, as this was the total weight of these structures for the joint designed in [2].

The electronics of the joint, which is likely to mainly include wires and a joint angle sensor, is estimated to weigh 50g. The gearing will depend on the chosen gear ratio, and was also estimated to weigh 50g. It is assumed that there will be one set of gearing for each clutch, and one for the brake.

The shaft going through the joint needs to be quite solid, and is thus likely to be made of steel. Hence the weight of the shaft is estimated using a 10cm long steel rod with a diameter of 5mm. With a mass density of $\rho_{\text{steel}} = 7.8\text{g/cm}^3$ the weight $m_{\text{shaft,joint}}$ is estimated as

$$m_{\text{shaft,joint}} \approx \rho_{\text{steel}}(\pi \times r^2 \times l) = 15.3\text{g} \approx 20\text{g} \quad (2.2)$$

The weight estimation is rounded upwards to 20 g to include gears transferring shaft rotation between joints.

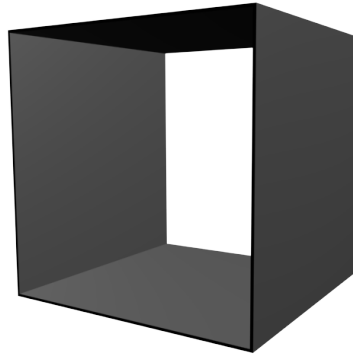


Figure 2.2: The shell of the joint

2.2.2 Weight of the arm

The arm will consist of 6 joints and a gripper, giving a total weight estimate of $m_{\text{arm}} = 21m_{\text{clutch}} + 3640\text{g}$ as shown in Table 2.3. The gripper is assumed to weigh the same as a normal joint.

Table 2.3: Arm parts and weight estimation

Part	Number	Estimated weight
Joint	6	$3m_{\text{clutch}} + 520\text{g}$
Gripper	1	$3m_{\text{clutch}} + 520\text{g}$
Total (m_{arm})		$21m_{\text{clutch}} + 3640\text{g}$

2.2.3 Weight of the body

The parts of the robot and their estimated weights are listed in Table 2.4. The shell and the

Table 2.4: Arm parts and weight estimation

Part	Number	Estimated weight
Shell and supporting structures	1	1000g
DC motor	1	m_{motor}
Electronics	1	1000g
Shafts	1	150g
Gearing	1	50g
Total (m_{body})		$m_{\text{motor}} + 2200\text{g}$

supporting structures will carry the parts and protect them from the environment. The weight of the shell is estimated by assuming that the body is a box $20\text{cm} \times 20\text{cm} \times 30\text{cm}$ (as done in Section 2.1). If the outermost millimeter of this box is made of aluminum, it would weigh:

$$m_{\text{shell,body}} = \rho_{\text{aluminum}}(20\text{cm} \times 20\text{cm} \times 30\text{cm} - 19.8\text{cm} \times 19.8\text{cm} \times 29.8\text{cm}) = 856\text{g} \approx 1000\text{g} \quad (2.3)$$

The approximation to 1000g is done to include supporting structures in the same way as in the estimation of $m_{\text{shell,joint}}$. It is likely that the shell itself will not weigh as much as 856g, as lighter materials such as plastics may be used for most of it. This is countered by the fact that the supporting structure is likely to weigh more than 144g.

The DC motor is set to weigh m_{motor} , which will be clear when the motor is decided. The gearing from the optimal rpm of the motor to the desired input rpm of the joints is estimated to weigh about 50g. The electronics of the robot will include several cameras and a motherboard with an CPU[28]. It is estimated to weigh about 1000g.

It is assumed that the length shaft in the body will be as if there were four shafts going from the center of the body and to each corner (i.e. to each arm). This gives a total length of $l_{\text{shaft}} = 4\sqrt{10^2 + 15^2} \approx 72\text{cm}$. Assuming that the shaft has a diameter of 5mm and is made of steel the weight estimate becomes:

$$m_{\text{shaft,body}} = \rho_{\text{steel}} \times 72\text{cm} \times \pi \times 0.25^2 = 110\text{g} \approx 150\text{g} \quad (2.4)$$

The approximation to 150g is done to include gears between the shafts of the body and the shafts of the arms.

2.3 Torque requirements

The torque requirements of the DC motor and of the clutches will be calculated based on the weight estimates of Section 2.2 and on the model from Section 2.1. A schematic view of the robot is shown in Figure 2.3. In this figure, the two upper arms are holding on to a supporting structure (such as a ladder), while the two lower ones are loose. Thus only the upper arms are lifting the robot. This corresponds to the requirement in [28] that each arm must be powerful enough to lift half the robot. The joints and motor must be powerful enough to hold the robot at this position.

It is assumed that the joints are configured so that just the elbow joints (joint_e in the figure) and the shoulder joints (joint_s in the figure) needs to produce torque when the robot is in this position. Further, it is assumed that the mass center of the body is at the geometrical center of the body, and that the mass centers of the arms is at the centers of the arms. The two shoulder joints must then carry the weight of the body, the lower arms and half of the two upper arms. The mass centers for the two lower arms are merged into one mass center weighing $2m_{\text{arm}}$, and the mass centers for the lower halves of the two upper arms are merged into one mass center weighing m_{arm} .

The elbow joint must thus lift half the weight of 3 arms and 1 body at an distance l_{elbow} of 40cm, which corresponds to a torque of:

$$\tau_{\text{elbow}} = F_{\text{elbow}}l_{\text{elbow}} = [0.5g(3m_{\text{arm}} + m_{\text{body}}) \times 0.4] \text{Nm} = [6m_{\text{arm}} + 2m_{\text{body}}] \text{Nm} \quad (2.5)$$

The shoulder joint has to lift approximately the same weight as the elbow joint but at a distance l_{shoulder} of 15cm, giving the torque:

$$\tau_{\text{shoulder}} = F_{\text{shoulder}}l_{\text{shoulder}} = [0.5g(3m_{\text{arm}} + m_{\text{body}}) \times 0.15] \text{Nm} = [2.25m_{\text{arm}} + 0.75m_{\text{body}}] \text{Nm} \quad (2.6)$$

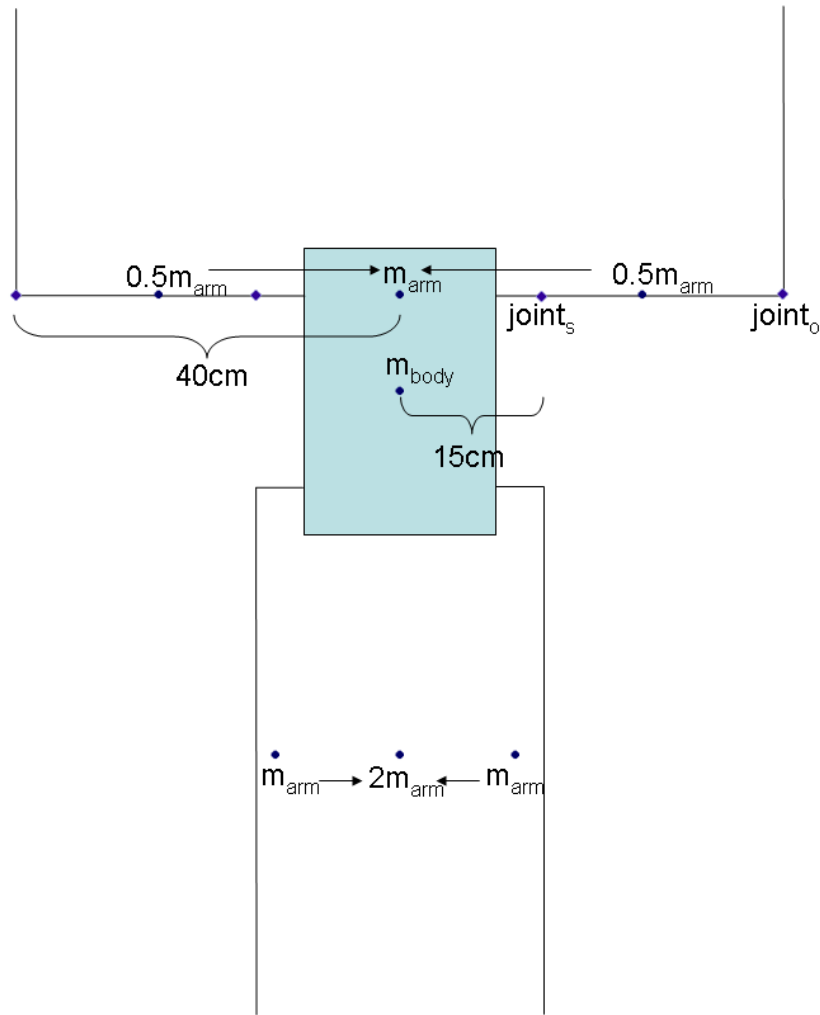


Figure 2.3: Schematic view of the robot

In both equations g is the gravitational acceleration, approximated to 10m/s^2 .

The torque requirements for the clutch equals the torque required by the elbow joint. The DC motor must be powerful enough to power both the elbow joints and the shoulder joints. This gives the torque requirements:

$$\tau_{\text{clutch}} = [6m_{\text{arm}} + 2m_{\text{body}}] \text{Nm} \quad (2.7)$$

$$\tau_{\text{motor}} = 2(\tau_{\text{elbow}} + \tau_{\text{shoulder}}) = [16.5m_{\text{arm}} + 5.5m_{\text{body}}] \text{Nm} \quad (2.8)$$

Inserting the equations for the weight of the arm and motor from tables 2.3 and 2.4 (in kg) into (2.7) and (2.8) the requirements becomes:

$$\tau_{\text{clutch}} = [126m_{\text{clutch}} + 2m_{\text{motor}} + 26.24] \text{Nm} \quad (2.9)$$

$$\tau_{\text{motor}} = [346.5m_{\text{clutch}} + 5.5m_{\text{motor}} + 72.16] \text{Nm} \quad (2.10)$$

It is clear that the weight of the clutch is more critical than the weight of the motor. The clutch should therefore be kept as light as possible.

2.3.1 The DC motor

Choosing and implementing the final DC motor of the robot is beyond the scope of this paper, as the complete robot will not be designed. However, it is necessary with an estimated weight of the motor in order to choose the clutch. The 4490 048 BS brushless DC servomotor from the Faulhaber Group[7] weighs 0.75kg and produces a maximum of 0.2024Nm at 10000rpm. If this is converted to the required 15 rpm of the joint, the motor has a torque of $0.2024 * 10000/15 = 134.9\text{Nm}$.

When the weight and torque of the motor is inserted into equation (2.10) an upper bound on the clutch weight is produced:

$$m_{\text{clutch,max}} = \frac{\tau_{\text{motor}} - 5.5m_{\text{motor}} - 72.16}{346.5} \text{kg} = 0.169\text{kg} \quad (2.11)$$

Preferably, the clutch should be as light as possible, to account for loss of efficiency e.g. in gearing, which can be quite severe. If no clutch is found that is light enough, another, more powerful motor will have to be used. The extra weight of a more powerful motor should not be a problem since the weight of the motor has much less effect on the torque requirements than the weight of the clutch.

Inserting the weight and torque of the motor into equation (2.9) gives new equation for the torque requirement of the clutch:

$$\tau_{\text{clutch}} = [126m_{\text{clutch}} + 27.74] \text{Nm} \quad (2.12)$$

3 Joint mechanism

This section looks at the new joint concept that has been proposed and identifies and briefly describes challenges of the concept. Some of these challenges are covered more deeply during development of a prototype that demonstrates the concept of the joint design. The design process involves several design proposals made in Autodesk Media & Entertainment's 3ds Max[1], and the 3ds files can be found in Appendix D. The final design is used to assemble a prototype of the joint. The prototype goes through a few qualitative tests that are evaluated together with the joint design in Section 5.1.

3.1 General issues of joint design

The proposed design concept will give a modular joint well suited for modular robots. A modular structure of the robot has advantages such as easy maintenance and manufacture, as well as the possibility for reconfiguration[19]. However, analysis done in the design of the humanoid robot LOLA[13] shows that it can collide with the demand for minimal weight. For humanoid robots, they figured that approximately 31% of the weight of the robot comes from the drive chains (gears and actuators). Thus the need of compact and lightweight joints are apparent. While trying to keep to a principle of modularity, they still ended up with 7 different drives for the actuated 22 degrees of freedom (DOF). This shows that it can be necessary not only to design one modular joint, but rather several different kinds of joints that still can be mounted to each other.

In many cases, it might be desirable for a joint to have more than one degree of freedom. For instance, a joint mimicking the human hip joint needs 3 DOF. Such joints can be very complex to design and manufacture. An alternative to designing such complex joints can be to place several 1-DOF joints together. This was done in the design of the hip joint of LOLA, where three serial drives compose the hip joint. The axes of the drives meet at one point, and thus they work as a 3-DOF spherical joint. While it might very well be possible to implement multiple DOF joints with the proposed design principle of this paper, the climbing robot in [28] (which this joint is likely to be tested on) has not been designed with such joints. Further, it is advantageous to design the new joint with one DOF before trying to include more DOF's. Thus multiple DOF solutions of this principle will not be explored.

3.1.1 Joint Components

A typical revolute joint usually consists of actuators, gearing systems, feedback components (such as encoders) and various other sensors and equipment[19]. Of these components the actuators and gearing system are of most interest in the design of new, light-weight joints, as they tend to take up the most space and weight.

For the actuators, significant design choices will be which type of actuator to use (e.g. some kind of electrical DC motor) and where to place the actuator. To reduce the size and weight of the main joint structure, it can be desirable to place the actuator some distance away. Some sort of transmission of power from the actuator to the joint must then be implemented. This is

done in [16], where actuators for adjusting damping, compliance and position in a finger joint are placed away from the joint and power is transmitted using cables and outer conduits. Another example is the knee joint of the humanoid LOLA robot. In this joint, shown in Figure 3.1, a ballscrew was chosen as the method of transmission due to its high efficiency. With the actuator placed closer to the hip, the moment of inertia of the hip was reduced. In addition, this type of transmission gave a more muscle-like actuator, which might be an advantage in the design of such an anthropomorphic robot.

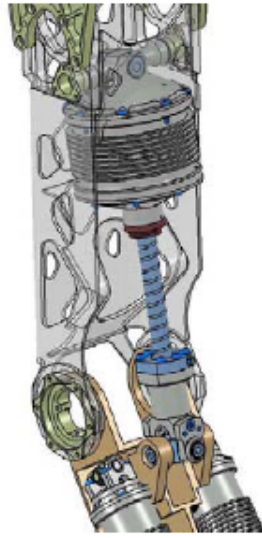


Figure 3.1: A knee joint using a ballscrew as transmission method[13]

Requirements that need to be taken into consideration when selecting the actuators are[13]:

- High dynamic response
- High output axis speed
- High output axis torque over a large speed range

For the new joint design proposed in this paper, it is likely possible to have the actuator running at the most efficient speed at all times. However, the demands on the actuators for traditional joints are easily transferred as demands on the clutch or transmission mechanisms in the new joint.

The output speed of the actuators (typically electrical DC motors) tends to be much higher than what is desired of the joint. Thus a gearing system is needed to reduce this speed. The gearing system can divide modular joints into four main groups, according to [19]. The groups are defined as joints using:

- Planetary gears, which are heavy and requires high assembling accuracy.

- Worm gears, which are lighter than the planetary gears, offer a high reduction rate at compact size, has the advantage of not being backdrivable[16]. However, it is less efficient than the planetary gears.
- Cup-type harmonic gears, which has a small volume, a high gear ratio, a steady transmission, and no backlash [14].
- Pancake-type harmonic gears, another variety of harmonic gears

In [19], the cup type harmonic gear was chosen as the gearing of a modular robot joint. This gave a simple and compact construction. Harmonic gears was also used in the revolute joints of LOLA robot. In the finger joint in [16] however, a worm gear was used to enforce a desired non-backdrivability of the actuator. In addition to the four kinds of gearing systems mentioned in [19], it is of course possible to use normal gears. While this can be less effective and heavier than the other solutions, it is simple and thus useful for preliminary designs.

3.1.2 Joint analysis and control

Any robotic joint is likely to be position and/or velocity controlled. It is therefore important to perform an analysis of possible error sources (e.g. for positional errors). In [19], three kinds of positional error were pointed out, namely axiality error, perpendicularity error and length error (i.e. error in the three orthogonal spatial axes). For high precision control, these and other errors must be taken account for in the control system.

For certain tasks, such as grasping, it can be beneficial for the robot to have force or compliance control of its joint in stead of (or in addition to) traditional position or velocity control. The joint mechanism in [16] is designed with this in mind, as it is to be used as a finger joint. Such control can be implemented using active force control, with feedback from force sensors like strain gauges, or passively using springs with controllable spring constants.

According to [16] passive control is more suitable for grasping motions requiring multiple motions points. The passive compliance control is implemented using a mechanical string with adjustable damping and compliance. The spring constant can be adjusted using a mechanical impedance adjuster. This is implemented using a brake disk and a brake rod on each of the moving sides of the joint. Thus, the more force that is applied in pushing the brake rod against the disk, the more damping is achieved.

Another issue that might be of interest in the analysis of a joint design, is stress analysis. The designers of the modular joint in [19] used finite-element analysis on the outer shell as this is the part of the joint, according to their analysis, that comes under most stress. Such a simulation is important if the joint will be pushed to its limits and/or there's doubt as to whether or not it will hold.

3.1.3 Examples of existing joint designs

A simple and common joint design is the actuated revolute joint. Joints of this design usually have one or two revolute degrees of freedom. If they have more than one degree of freedom, the

axes of rotation are typically orthogonal to each other. These joints have their actuators within themselves, one for each DOF. Several joints of this type will typically be connected together to form a chain of joints constituting an robot arm or a snake robot [24].

An alternative to placing the axes of rotation orthogonal to each other is to place the axes of orientation at a non-orthogonal angle. This is often used when the robot should mimic motion of animals (such as snakes) or humans. One way to do this is to use bevel gears as shown in Figure 3.2

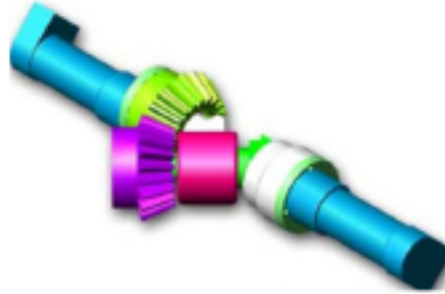


Figure 3.2: A revolute joint with 2 DOF where the rotational axes are not orthogonal[24]

Another kind of joint uses parallel mechanisms to connect adjacent links. This joint design provides rigidity, accuracy and strength. In this design, linear actuators are placed in a non-planar, parallel structure, as shown in Figure 3.3. A disadvantage with this kind of joint is that they tend to be bulky since several actuators are stacked side by side[24]. Further, this design principle seems to be unsuitable for use along with the principle with one actuator and multiple joints explored in this paper.

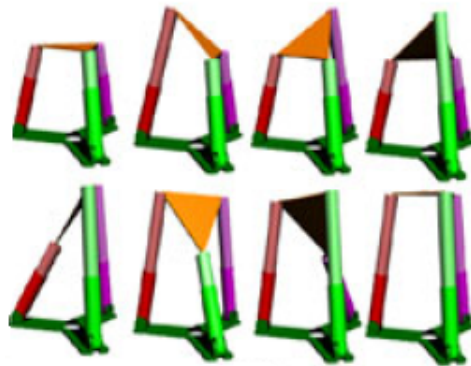


Figure 3.3: A joint using parallel, linear actuators[24]

3.2 Challenges of the proposed design

This section briefly describes four of the main challenges of the new joint design concept. Three of these challenges, the clutching mechanism, transmission of power and braking, are looked at more deeply in this report during the development of a prototype of the joint.

The four main challenges are:

- *The clutching mechanism* - A major challenge of the proposed design will be to find a suitable clutch mechanism. The mechanism must have the ability to transfer a variable amount of torque to the joint rotation. The device must also be resilient to wear and tear, as this might be a problem when friction is induced over a prolonged period of time. The clutching mechanism is treated in detail in section 3.3
- *Transmission of power* - When one motor is powering several joints like in the proposed design, there must be a method of mechanically transmitting power from the main shaft of one joint to the main shaft of the next. Two possible ways to do this that both are looked at in this report is to use bevel gears, as is done in the exemplification of the design concept in Section 1.1 and in the first and second designs in sections 3.4 and 3.5, or to use a universal joint as done in the third and fourth designs in sections 3.6 and 3.7.
- *Braking* - There must be a way of preventing the joint from moving due to external forces such as gravity. If both the clutching mechanisms are turned on at the same time, this would stop such unwanted movement, however it would also either block the rotation of the main shaft or induce unnecessary wear and tear on the constantly slipping clutch devices. In the designs of the following section, a braking clutch that is disconnected from the main shaft has been used. Worm gears have also been used, as they have the nice property of being easy to rotate from the screw part to the wheel part but difficult to rotate the other way.
- *Weight* - As with all joints used in climbing robots (and indeed joints used in other robots as well) the joint need to be as light as possible while still being strong enough to carry the robot. This gives requirements on what materials to use and on the size and weight of clutches, gears and sensors.
- *Modeling, simulation and control* - A control system is necessary for the new joint design. A dynamic model of the clutch can be of advantage when developing the control system, as well as to use in simulators as the one developed in Section 4.

3.3 The clutching mechanism

A principal component of the new joint design in Figure 1.1 will be the torque-transmitting mechanism using friction plates. Since this essentially is a clutch, it was decided to implement this mechanism using a commercial miniature clutch. The clutch was chosen before the design of the joint was finished, and must the choice must then be made based on the weight estimations of Section 2.2, which might be incorrect. The reason that the clutch had to be decided at such an early stage was that the clutch might have long delivery time.

Three types of miniature clutches have been considered for use in the new joint mechanism; the wrap spring clutch, the electromagnetic friction clutch and the magnetic clutch. The wrap spring clutch uses a spring wrapped around the input and output shaft to generate friction and thus transfer torque. The amount of torque that is transferred is adjusted by adjusting the tightness of the spring. Dynacorp[4] offers 2 kinds of wrap spring clutches, the DCB series and the SC series. Only the DCB series is electrically actuated, using a solenoid to tighten the spring. The dimensions of the smallest DCB clutch, the DCB-2, is shown in Appendix B and is summarized in Table 3.1.

Another type of clutch that uses friction as the method of torque transfer is the electromagnetic friction clutch. These clutches uses an electromagnetic force to clamp two objects together. One of the objects rotate along with the input shaft, while the other is connected to the output shaft. When these objects are clamped together, friction forces are induced and torque is transmitted from the input shaft to the output shaft. Inertia Dynamics[10] provides several series of these clutches. The most suitable series was decided to be the SO line of clutches, which is a shaft to shaft clutch coupling. In Appendix A the mechanical and electrical data of these clutches is provided. An iterative process using the equations from Section 2.3 showed that the SO15 type of clutch was suitable. Technical data for this clutch is shown in Table 3.1.

Magnetic clutches are the third type of clutches that were looked into. These type of clutches, provided by Rimtec Corporation[20], uses permanent magnets with opposing poles to provide torque transmission. Since they do not rely on friction, these clutches are likely to suffer much less wear and tear than the friction based clutch. However, when the clutches are slipping, heat will still be generated that can damage the clutch. Further, magnetic clutches are quite weak when compared to friction clutches. Lastly, in order to get a variable torque from the clutch, some customization of the clutch must be done as Rimtec corporation does not offer this feature. This reduces the suitability of this clutch, even though it remains an interesting option of the joint prototype shows that friction damage is a problem. Data on the magnetic clutch HKD 2 can be found in Appendix C.

Table 3.1: Wrap spring, electromagnetic friction and magnetic clutches

	Wrap spring clutch (DCB-2)	Electromagnetic friction clutch (SO15)	Magnetic clutch (HKD 2)
Torque(Nm)	2.825	1.13	0.1
Width/diameter(mm)	50.8	47.5	31
Height(mm)	86.1	-	-
Length(mm)	63.5	40	55
Weight (g)	566	107.54	129
Max rpm	1800	5000	10000

The wrap spring clutch is heavier than the maximum clutch weight stated in Section 2.3, and would require an heavier and more powerful motor. Further it is quite bulky and has a relatively low torque to weight ratio. The magnetic clutch is within the weight requirement, but has only a fraction of the output torque of the electromagnetic friction clutch which is also light enough.

Thus the electromagnetic friction clutch was chosen to be tested in a demonstrator consisting of two joints.

While Inertia Dynamics suggest a maximum input rpm of 5000 for their clutches, it is likely that such speed will generate much heat and friction damage to the clutch. Further, it is a possibility that power lost due to inefficiency will be much more than what is gained by the high gear ratio from the clutch rotation to the joint movement that is made possible by such speeds. The main shaft is not expected to hold more than 1000 rpm, and to avoid having to gear the speed up before the clutch, 1000 rpm was used as the input speed when calculating torque requirements. With 15 rpm (or 90 degrees per second), this gives a gear ratio of 66.7 : 1. Ignoring loss due to inefficiency in gearing (which might be considerable), this gives the joint an output torque of $66.7\tau_{SO15} = 75.37\text{Nm}$. Inserting the weight of the SO15 clutch into equation (2.12) gives a torque requirement of 41.29Nm. The output of the clutch is thus well within the limits, and gives room for gearing inefficiency. The estimated weight of the joint with this clutch is 842.62g.

3.3.1 The SO15 clutch

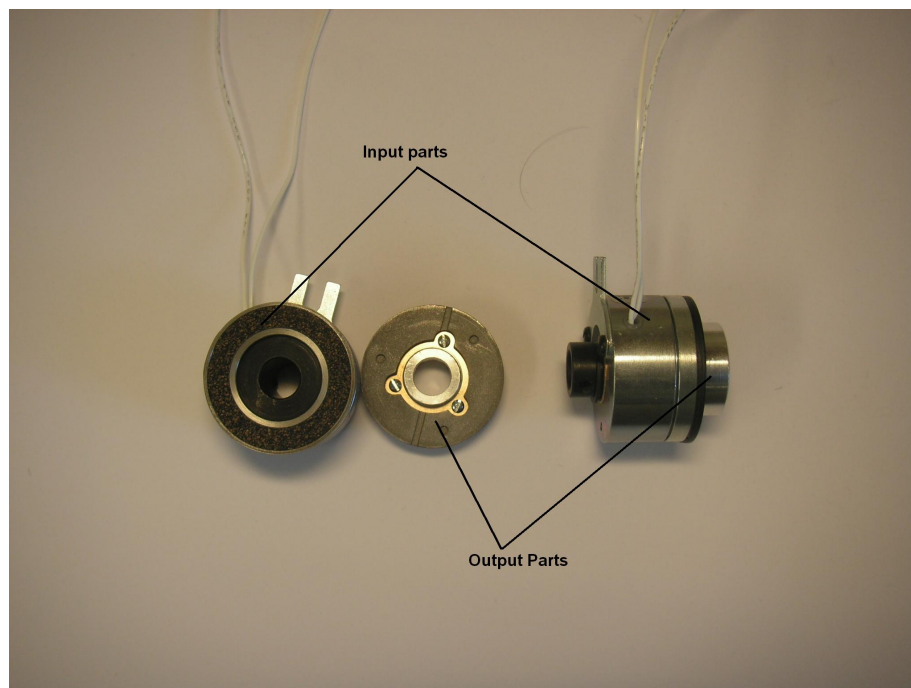


Figure 3.4: The SO15 clutch

Figure 3.4 shows the SO15 clutch. It consists of an input part and an output part, as noted in the figure. The input part contains an electromagnet that will, when powered, exercise a pulling force on the output part. The higher the voltage applied to the electromagnet, the stronger the

pulling force, and hence more friction is induced by slipping of the clutch. The clutch can be applied a maximum of 24V. The output part of the clutch is equipped with a small spring that pulls it back when the clutch is turned off.

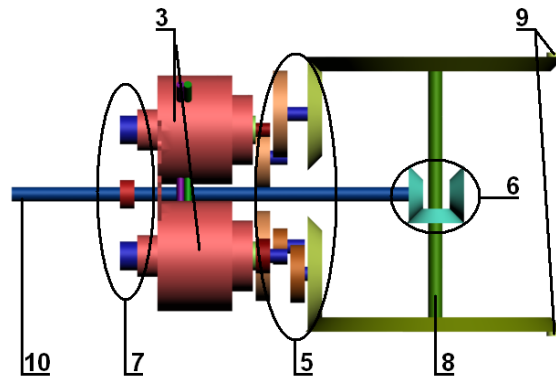
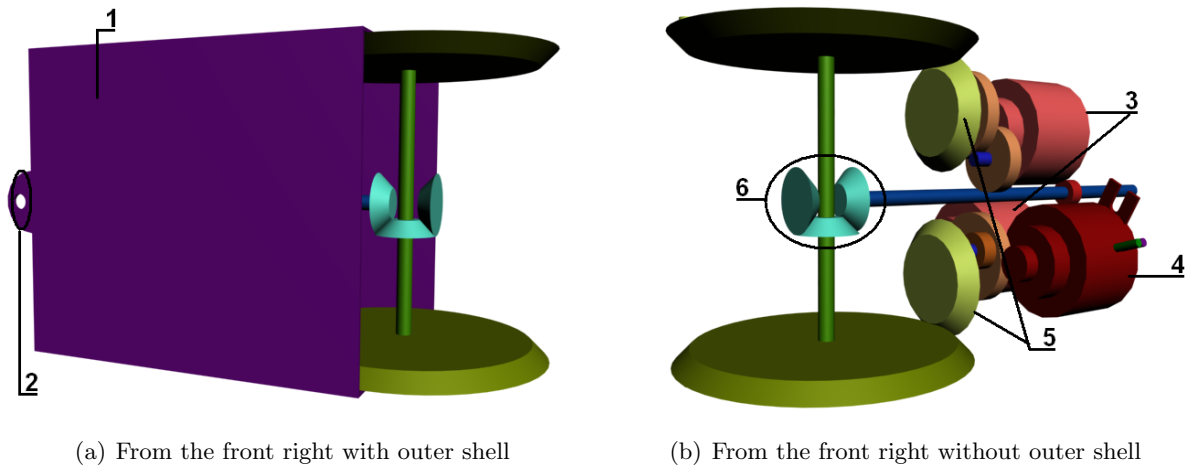
Before use, the clutches must be burnished. This is done by applying 30-40% of the maximum voltage (around 8V) until the output part cones in contact with the friction surface on the input part. The clutches should be mounted with a small air gap between the input and output part, about 0.1-0.5 mm thick.

3.4 Joint design 1

This section describes the first joint design made in 3ds Max[1]. The 3d model is shown in Figure 3.5, and important parts of the design are numbered and explained. A discussion of the design is then given, along with an evaluation that led to the second joint design described in Section 3.5.

The parts of notice in the first joint design are:

1. *Outer shell* - The outer shell protects the joint from the environment, and also serves as a connection points between two joints. The inner joint parts can be fastened to the outer shell. The outer shell is 3mm thick and made of aluminum, which should be enough to support the joint while still light-weight.
2. *Fastening points on the shell* - This is where this joint connects to a previous joint. These points are on all four sides of the shell, giving two possible connection configurations.
3. *Drive clutches* - The clutches that drives the rotation of the joint, one clutch drives the rotation clockwise while the other clutch drives it counterclockwise.
4. *Brake clutch* - This clutch is of the same type as the driving clutches (i.e. the SO15 electromagnetic clutch), however the input is fastened to a static structure such as the outer shell rather than the main shaft. Thus, if this clutch is turned on, it will brake the movement of the joint.
5. *Gearing system* - This is the gearing system that gears the speed from the output of the clutches down to the desired joint speed. The actual gearing of this design has a total gear ratio of 63.2 : 1. The last gear of the gearing system is a bevel gear that is used to change the axis of rotation to the output rotation axis of the joint.
6. *Bevel gears* - These gears transfer the rotation, and thus torque, from the main shaft of this joint to the main shaft of the connected joint (or other connected device). The rotation of these bevel gears are independent of the rotation of the joint.
7. *Transmission belt gears* - The 1:1 gearing from the main shaft to the input of the driving clutches are implemented using a transmission belt that is wrapped around these gears. The belt itself has, for simplicity, not been modeled.



(c) From the left

Figure 3.5: The first joint design from different angles, with and without outer shell

8. *Joint rotation shaft* - The joint axis of rotation goes through this shaft, and the bevel gears fastened to the top and bottom of it. If another joint is connected to the fastening points on top of the bevel gears, it will follow this rotation. It is made of steel and has a diameter of 5mm.
9. *Fastening points* - This is where the fastening points on the shell of an attached joint are fastened.
10. *Main shaft* - This is the main shaft that transfers torque from the DC motor of the robot. It is made of steel and has a diameter of 5mm.

To ensure that the joint does not move when unactuated (due to gravity, other joints movement or other disturbances) a third electromagnetic clutch was added as a brake. This clutch will have a non-moving input and the output mounted to the output shaft of the joint. In choosing this brake it was reasoned that since the SO15 clutch should be powerful enough to move the joint, it should also be powerful enough to brake it and hold it still.

It is also possible to brake the joint by turning on both the clutches, however this would either stop the main shaft from rotating and thus stop power transmission to other joints, or it would severely wear on the clutches.

The gearing system used normal gears, and not one of the more advanced gear systems mentioned in Section 3.1. This is to keep the design as simple as possible. This makes the joint heavier than with for instance harmonic gears, however for a prototype that will test the concept of using a clutch mechanism this way, it is not necessary to fulfill the weight requirements. The gearing ratio of the design is somewhat, but not much, lower than the ideal ratio of 66.7 : 1 from Section 3.3. As with the weight requirements, it can be argued that the gearing ratio is of less importance to a prototype like this.

The shafts of the joint are all made of steel, which is stronger and less flexible than aluminum. The main shaft and the joint rotation shaft are 5mm in diameter, the gearing shafts and the shafts going to the clutches are all 10mm thick, which is the diameter of the bores in the clutches.

The three clutches of this design are placed in an equilateral triangle, to have as much symmetry around the main shaft as possible. Similarly, the gearing systems is the same above and below the main shaft to give symmetry. Symmetry makes the joint easier to model mathematically, and is convenient when the joints can be mounted on each other with different orientations. It is however possible to make the clutch more symmetric by placing one of the clutches around the main axis, as was done in the second design presented in Section 3.5.

The elasticity of the transmission belt can be a drawback, as it makes modeling and control of the joint more difficult. While it is convenient to use a transmission belt, it can therefore be better to use gears, as was done in the second design.

Another disadvantage of this design is that it is difficult to see the inner mechanisms of the joint since it is covered by the outer shell. While this is advantageous for the joint when it is used on the climbing robot of [28], when the joint must be protected from the environment, it is better to be able to observe the mechanisms of the joint (especially the clutching mechanism) in the case of a prototype. It will then be easier to point out errors and potentials for improvements.

This model has not included any fastening points for the clutches and gears, nor is there any support for the main shaft or the shafts running through the clutches. This is an obvious drawback that must be corrected before a prototype can be built.

No sensors for measurements of the joint angle has been included in this design. Such sensors are necessary to have a decent control of the joint, and are thus included in the second design.

The gears used in the design are represented in 3ds Max as cylinders and cones and are not based on real gears. For prototype implementation, it is necessary to find real gears for the gearing system.

3.5 Joint design 2

This section describes the second joint design made in 3ds Max. The 3d model is shown in Figure 3.6, and important parts of the design are numbered and explained, as with the first joint design. A discussion of the design is then given, along with an evaluation that led to the third joint design described in Section 3.6.

The parts of notice in the second joint design are:

1. *Fastening plates* - There is an octagonal fastening plate on each end of the joint. When connecting two joints, the front plate of one joint and the back plate of the other are fastened together (e.g. with screws).
2. *Bevel gears* - As with the first design, these gears transfer torque from the main shaft of this joint to the main shaft of a connected joint. Note that one has been mounted on the end of the other shaft, and that a bevel gear has been added to the top of the structure.
3. *Joint rotation shaft* - Functions similarly to the joint rotation shaft of the first design. It is made of steel and has a diameter of 5mm.
4. *Potentiometer* - This box represents a potentiometer used to measure the joint angle.
5. *Support plates* - These plates provide support for the main shafts, the shafts going through the driving clutches and the gearing system. They also provide fastening points for the clutches. They are 3mm thick and made of aluminum.
6. *Connection clutch* - This is the clutch that transfers torque from the main shaft to the transmission gears, which transfer it to the driving clutches. When this clutch is turned on, torque is transmitted, when it is off torque is not transmitted and the main shaft and joint rotate independently of each other. The clutch is of the same type as the driving clutches.
7. *Outer support structure* - This structure, made of 5mm thick aluminum, is the main support structure of the joint. The support plates are attached to this structure, as are the potentiometer, the joint output axis and the back fastening plate.
8. *Driving clutches* - Like in the first design, there is one clutch for each rotational direction.

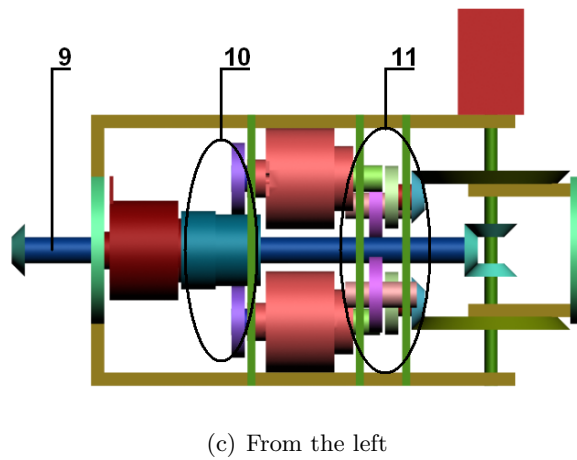
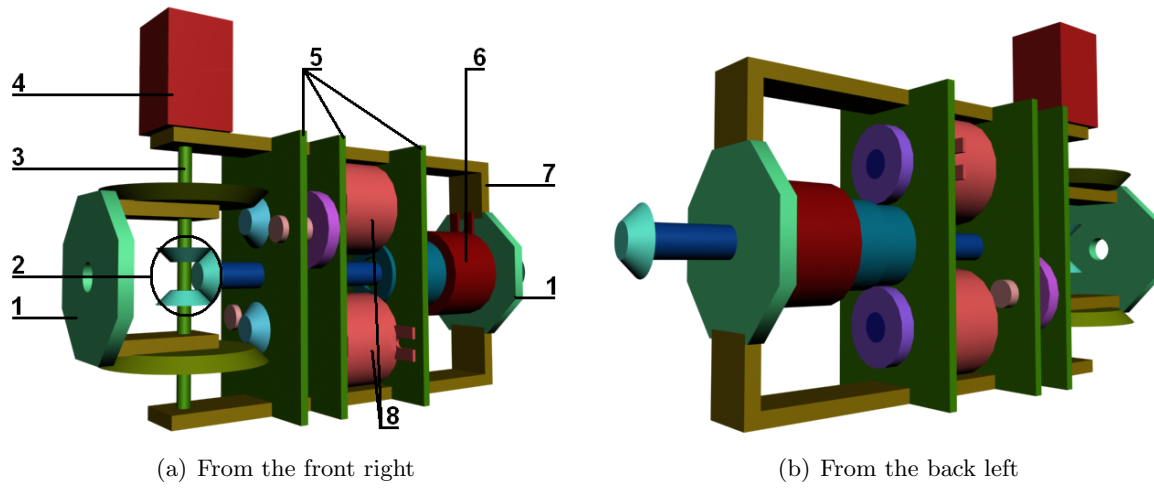


Figure 3.6: The second joint design from different angles

9. *Main shaft* - Like in the first design, this shaft carries the torque from the DC motor of the robot. It is made of steel and has a diameter of 10mm.
10. *Transmission gears* - These are two normal gears and one custom made gear providing torque transmission from the connection clutch to the driving clutch. The gearing ration is 1 : 1.
11. *Gearing system* - Similar to the gearing system of the first design, but with one gear less. The gear ratio is 25.8 : 1

The major change made in this design is the removal of the brake clutch and the addition of the connection clutch. The connection clutch is of the same type as the driving clutch, i.e. SO15. Having a clutch around the main shaft ensures more symmetry than the first design, and this solution also avoids the use of an elastic transmission belt. The main shaft runs through this clutch and continuously rotates the input (i.e. back) part of it. When the clutch is turned on, the output part will ensure that torque is transfered from the main shaft and to the driving clutch, via the transmission gears. The driving gears can then rotate the joint rotation shaft.

If the connection clutch is turned off, the rotation of the joint shaft (and thus of the shafts through the driving clutches) are independent of the rotation of the main shaft. If both the driving clutches are then turned on, the joint rotation shaft will be locked in position, eliminating the need for a brake. Since the connection clutch is turned off, this will not lock the main shaft. Torque can thus be transfered to the next joint.

The transmission mechanism from the connection clutch to the driving clutches consists of two normal gears and and one custom made gear. The custom made gear is fastened to the output part of the joint and has support from one of the support plates. The cogs are in between the fastening and support, where the normal gears are connected to it as seen in Figure 3.6.

To keep the symmetry of the joint, the driving clutches has been moved so that one is directly above the main shaft, while the other is directly below. This lead to changes in the gearing system. The largest change was the removal of one of the gears. This gives a gear ratio of 25.8 : 1, which is considerably less than the ideal gear ratio of 66.7 : 1. However, as was done in the first joint design, the gear ratio is deemed less important for these designs as they will only serve as a demonstrator of the concept. This decision becomes particularly clear in the fourth design, where there is no gearing system at all. If the output speed turns out to be to high, the input speed of the main shafts can be lowered.

The gearing system is otherwise similar to the one of the first design in that it uses normal gears for simplicity and that the last gear is a bevel gear that is used to change the axis of rotation to the output rotation axis of the joint. As with the first design, the 3ds Max model of the gearing system used cylinders and cones as models for the gears, and are not based on actual gears.

Support plates are necessary to avoid unwanted swing movement of the shafts. The bores for the shafts in the support plates and the connection plate are fitted with ball-bearings (these are not included in the 3ds Max model). Ball-bearings will provide support for the shafts while at the same time minimizing friction.

The box labeled potentiometer shows the location of the potentiometer and is not intended to be part of the implementation of the design.

There is one more bevel gear for torque transfer between joints than in the first design. This is to make the mechanism more robust. Further, one of the gears has been placed on the other end of the main shaft. This is for illustration, to better show how the joints will be connected together. How the gears would be placed in an implementation of the design would depend on hardware considerations.

All the supporting structures are made of aluminum, which is light but should be strong enough to carry the clutches, shafts and gears. The supporting plates are 3mm thick, which should be enough to support different mechanisms. The outer support structure is 5mm thick, to ensure that it is strong enough as it has to carry the support plates as well.

The main shaft has a 10mm diameter which is larger than the main shaft of the first joint design. This is because the main shaft must carry not only the torque required by this joint but also torque required by other joints connected to this one. To allow the output part of the connection clutch to rotate independently from the main shaft, the shaft must either be slightly thinner where it goes through the output bore, or the bore must be slightly larger.

By using connecting plates to attach to joints, the axis of rotation of the joints can be rotated as much as desired around the main shaft axis. This gives much more room when choosing the degrees of freedom of the overall robot.

The second design is much more open than the first design. This is a major advantage, as it is now possible to observe the behavior of the joint directly. Another advantage is the addition of support plates, which makes the design more realistic and closer to an actual implementation. The supporting plates are however too thin to be fastened to the outer support structure with screws. To allow the plates to be screwed in place, they should be at least 5mm thick[9].

The removal of the transmission belt is also an advantage of the second design. However, the custom-made gear used to transmit torque from the main shaft to the driving clutches are an unwanted complexity. In addition, having three clutches to control two directions of rotation seems intuitively to be one to many. If the number of clutches was reduced to two, it would significantly reduce the weight and complexity of the joint.

The bevel gears that are used to transfer torque from one joint to the next requires a high degree of precision when assembling the joint[9]. One possible solution to this problem is to create a separate module (i.e. a box) containing the bevel gears. Such a module could be assembled separately from the rest of the joints, and could also contain bores where the shafts could be fitted. This would reduce the assembly complexity. Another, and simpler, solution is to use a universal joint, a cardan joint, to transmit the rotation.

3.6 Joint design 3

This section gives a brief discussion of the third joint design, shown in Figure 3.7. This design is largely similar to the second design, the differences are:

- The cones and cylinders used by 3ds Max to model the bearings has been replaced by models of actual gears from Mekanex[15]. The gear ratio of the design is now 38.6 : 1, which is

slightly higher than the gear ratio of the second design. The gears are marked 1 in Figure 3.7. Note that the custom made gear used to transmit torque from the connection clutch to the driving clutches is still not based on a real gear.

- The bevel gears used to transmit torque from one joint to the next is replaced by a universal joint (or cardan). The universal joint gives limits the joint value because there will be a loss of transmitted effect over the universal joint, dependent on the rotational speed of the main shaft and of the rotation angle of the joint. This limitation is acceptable, especially for a demonstrator such as this. The universal joint is marked 2 in Figure 3.7.
- The joint rotation axis has been separated in two parts to make room for the universal joints.

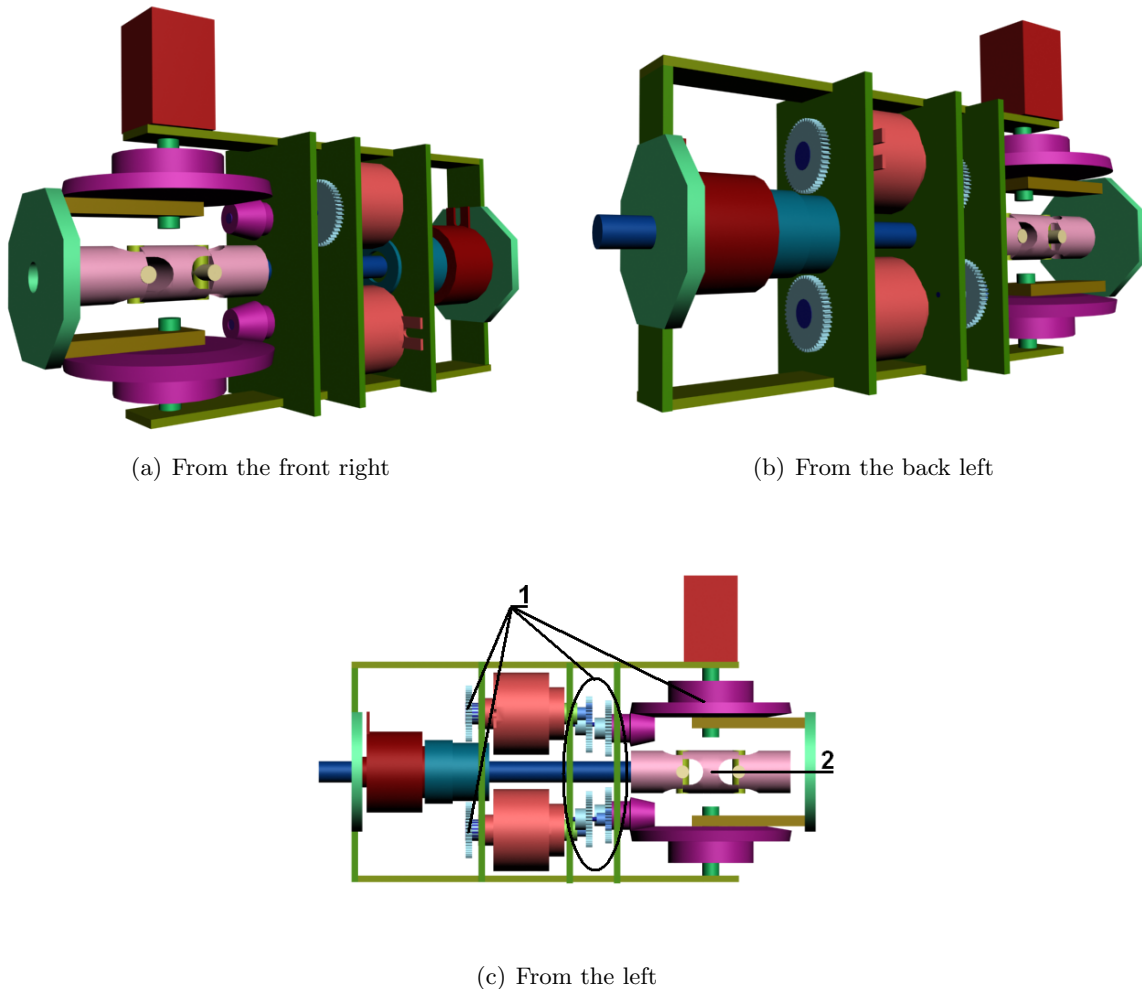


Figure 3.7: The third joint design from different angles

This design removes the complexity of the bevel gears transmitting torque from one joint to the next. It is also closer to implementation as the gears used in the design are actual gears that can be ordered from Mekanex. The improvement of the gear ratio is a side effect from using models of real gears in stead of cones and cylinders. This was not intentional as the gearing ratio of the demonstrator has been decided to be less important.

As with the second design, it is would still be advantageous to remove the custom made gear and to reduce the number of clutches to two. This is done in the fourth design.

3.7 Joint design 4

This section describes the fourth and currently final design of the joint prototype, shown in Figure 3.8. The design is based on the previous three designs, and much of it was done by [12]. Different parts are numbered in Figure 3.8 and listed below, and a more detailed description and explanation of the joint design is then given.

The parts of notice in the fourth joint design are:

1. *The main shaft* - As in the previous designs, this is the main shaft that carries torque from the DC motor.
2. *Input Gearing* - These transfers the rotation of the main shaft to the input of the upper driving clutch. Note that the inputs of the driving clutches will rotate in opposite directions.
3. *Driving clutches* - These are the clutches that drives the rotation of the joint. One of the clutches will, if turned on, rotate the cogwheels driving the transmission belt one way, while the other will turn them the other way. This way, the clutches controls different directions of joint shaft rotation.
4. *Transmission belt* - This is the belt that transfers torque from the driving clutches to the worm gear. To cogwheels driving the transmission belt is shown in Figure 3.8(d).
5. *Joint rotation shaft* - The joint axis of rotation goes through this shaft. At the top of this shaft, there is a bore where the potentiometer can be fitted.
6. *Worm gears* - These gears transfers changes the axis of rotation to the output axis of rotation and also provides a gearing ratio of 11.25 : 1. The screw part of the worm gears can rotate the wheel part with much less force than the other way around. Thus, it is likely that external forces such as gravity cannot rotate the joint, and the need of a brake clutch may be removed.
7. *Connection plate* - This is the plate that a connected joint will be fastened to.
8. *Universal joint* - This is the universal joint that allows the torque of the main shaft of this joint to be transfered to the main shaft of the next, even when the joint is rotated.
9. *Support plates* - These plates supports the shafts and gears and provide fastening points for the screws. They are 5mm thick and made of aluminum.

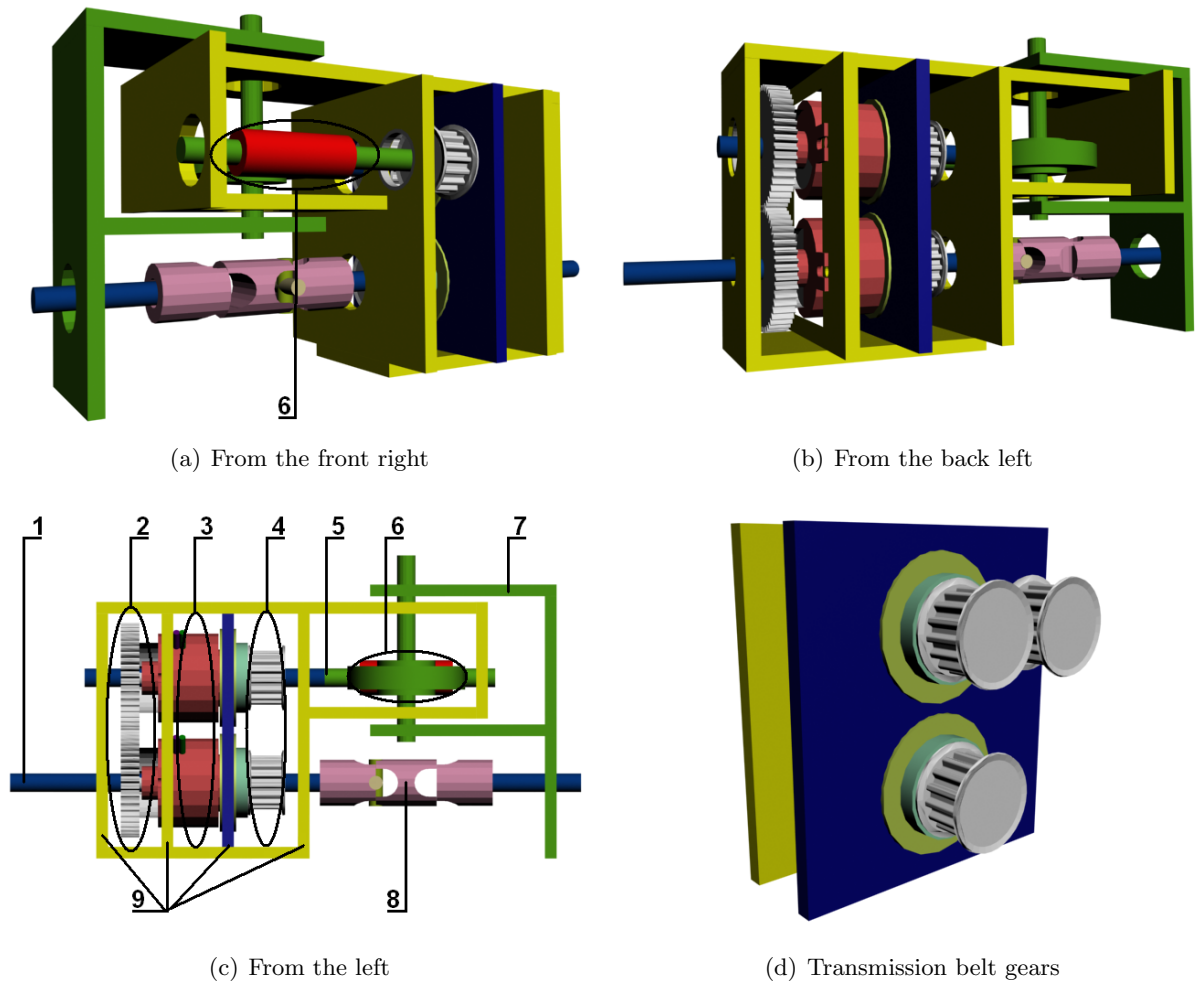


Figure 3.8: The fourth joint design from different angles

The major change in this design is the removal of a clutch. This comes from the addition of the worm gears, which are likely to eliminate the need of a clutch-driven brake mechanism by not being back-drivable. The input part of the two clutches that are left will rotate continuously in directions opposite to each other. When one of the clutches are turned on, it will transfer the rotation of the input part to the output and thus to the joint rotation shaft. Since the input parts of the clutches rotate in opposite directions, each clutch controls a different direction of rotation. Both of the clutches should not be turned on at the same time, as this would block the rotation of the main shaft.

Another major change in the design is the removal of most of the gearing system from the output of the clutches to the joint rotation shaft. The only gearing that is on the output of the clutches is the worm gears, which provide a gearing ratio of 11.25 : 1, this is far less than the ideal ratio of 66.7 : 1. As argued in Section 3.5, the gear ratio need not be as high as the ideal gear ratio for the prototype, as it will only demonstrate the concept of the joint design and will not be required to meet the strict torque demands of the joints that will be used in the climbing robot.

The diameter of the main shaft of the joint has been reduced from 10mm to 8mm. This allows the shaft to run freely through the bottom clutch, so that the rotation of the shaft can be transferred to the next joint. The shaft is fastened to the input part of the bottom clutch, so that this part rotates when the main shaft rotates. The gears on the clutch inputs ensures that the top clutch also rotates along with the main shaft, albeit in opposite direction of the input of the bottom clutch. The other shafts of the joint has also been designed to be 8mm in diameter, this is for simplicity.

The support plates will be fitted to the outer support structure with screws that are 3mm in diameter. Because of this, the thickness of the support plates has been increased from 3mm to 5mm.

There is a hole on the top of the joint rotation shaft for the potentiometer. The details of mounting the potentiometer has not been included in the design, and will be done in the assembly of the joint. The potentiometer can be seen in Figure 3.9, which shows the assembled joint.

While it was argued in the first joint design that the transmission belt introduced unwanted elasticity to the system, it was a quite convenient and light-weight solution for transferring torque from the output of the clutches to the worm gears. For this reason, the transmission belt was reintroduced.

This is the final design of the joint prototype. The joint will be qualitatively tested in Section 3.8. The tests and design will and evaluated in section in Section 5.1.

3.8 Testing of the prototype

This section describes the test setup and procedure for testing of the joint prototype which was assembled by [9]. The testing is for a large part qualitative in nature, and is intended to demonstrate the concept of using clutches as the local actuators of a joint. The tests that will be done are:

1. Motor and main shaft test - tests if the motor is able to drive the main shaft, and if the main shaft is able to rotate at the output of the joint even when the joint angle is not zero.

2. Worm gear back-drivability - tests if the worm gear is back-drivable.
3. Clutch functionality - test slipping of the clutches and of they are able to rotate the joint.
4. Potentiometer - calibrates the potentiometer. The potentiometer will be useful if feedback is introduced in further work, as discussed in Section 6.1.
5. Maximum and minimum rotation angle - uses the potentiometer to measure the maximum and minimum rotation angle of the joint.

The test results are evaluated along with the joint design in Section 5.1

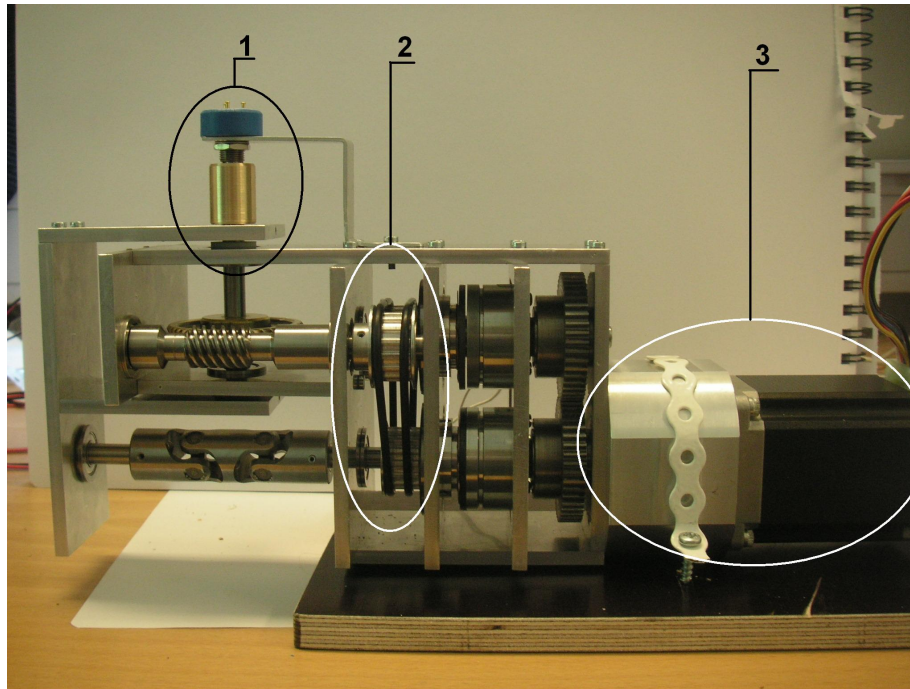


Figure 3.9: The assembled joint, assembled by [9]

The assembled joint is shown in Figure 3.9. Points of notice in the joint are marked. These are:

1. *Potentiometer* - This is the potentiometer used to measure the joint value.
2. *O-rings* - The cogged transmission belt turned out to be too short, so o-rings were used as a temporary solution. This is a major drawback, as the o-rings slips more easily than the clutches, thus transferring much less power and preventing the resistance on the clutches to be large enough for them to slip.
3. *Motor* - The motor was connected to the joint. The structure was also fastened to a wooden board to prevent the joint from moving too much. The motor produced a maximum torque of 1.27 Nm, according to the data sheet[22].

The components used by the test setup are listed in Table 3.2.

Table 3.2: Components used in testing

Component	Type	Manufacturer	Reference
Motor controller	8SMC1-USBh	Standa	[25]
Input/Output unit	NI USB-6009	National Instruments	[18]
Stepper motor	Step-Syn 103H7126-1941	Sanyo Denki	[22]
Gears	10045 - S08	Mekanex	[15]
2 clutches	SO15	Inertia Dynamics	[10]
3 Cogwheels	21-T5/15-2	Mekanex	[15]
2 O-rings	-	-	-
Universal joint	KRD10-N	Mekanex	[15]
Worm gears	14H 10 04 & 141-A 10 04	Mekanex	[15]
Potentiometer	357-2-1-1S22-103	Vishay	[27]
3 1k Ω resistors	-	-	-
2 NPN transistors	2N4401	Fairchild Semiconductor	[6]

The clutches were controlled using the circuit shown in Figure 3.10. The 5 volt supply was taken from two digital outputs of the Input/Output unit, using the Measurements and Automation Explorer that came with LabVIEW[17]. In this software, it was possible to set the each output high or low, and thus turn each of the clutches on and off. The resistors are both 1k Ω

The potentiometer was connected in series to a resistor of 1191 Ω , as shown in Figure 3.11. The voltage v_R was measured using the analog input of the Input/Output unit. The voltage v_{in} was set to 5V using the analog output of the same unit. This was done in LabVIEW, and the diagram can be found in Appendix D. The measured voltage v_R is given by

$$v_R = \frac{Rv_{in}}{R + R_{pot}} \quad (3.1)$$

This gives

$$R_{pot} = R \left(\frac{v_{in}}{v_R} - 1 \right) \quad (3.2)$$

3.8.1 Motor and main shaft test

The motor was controlled using the software SMCVieW which came with the motor controller. This software allows the user to set the speed of the motor and the direction of rotation. When the stepper motor was started, the main shaft rotated as it should. However, the gears at the input of the clutches were mounted too close to each other, making it heavy for the motor. At times, the rotation stopped all together. Some oil was applied to the gear to make it easier. This helped, but the motor still stopped from time to time. As the main shaft turned the gears became worn down. This too made it easier for the motor, and the rotation worked better and

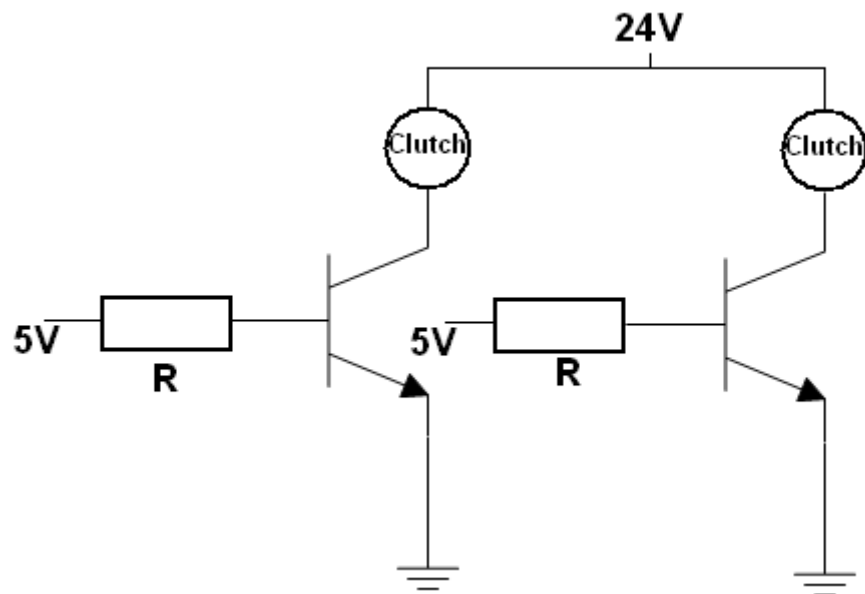


Figure 3.10: The clutch circuit

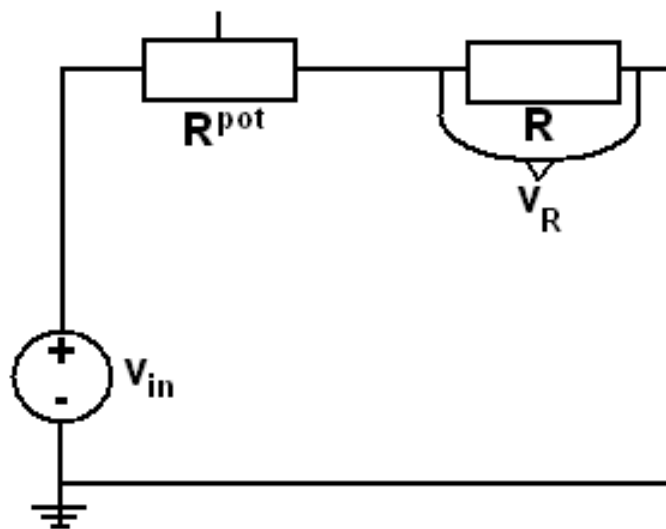


Figure 3.11: The potentiometer circuit

better. A video of the rotation can be found in Appendix D. When the motor stopped, it usually helped to switch the direction of rotation of the motor. The gears were not completely circular, and it was noticed that they tended to be in the same place when the motor stopped.

The maximum speed the motor ran relatively smoothly was about 600 steps per second, which is $60 \times \frac{600 \text{ steps} \times 1.8^\circ / \text{step}}{360^\circ} = 180 \text{ rpm}$. At speeds higher than 700 steps or 210 rpm, the gears turned slower at often stopped, and the motor was quite noisy.

The cogwheels was turned manually to rotate the joint. The universal joint ensured that the main shaft rotated also at the output, even when the joint was rotated. A video found in Appendix D shows this. The motor stopped more often when the joint was rotated, indicating that the loss of effect in the universal joint induced a higher work load for the motor.

The bottom clutch was mounted a bit too tight, and tended to move the joint even when it was not powered. It became worn as the main shaft rotated, but apparently not enough to stop the rotation. It is likely the metal parts of the clutch that became worn. Figure 3.12 shows that the input part of the bottom clutch is touching the input part, while there is a gap between the input and output part of the top clutch.

3.8.2 Worm gear back-drivability

An external force was applied to the output of the joint, to check if it was possible to rotate it. It was discovered that even though it was heavier to rotate the joint by using an external force on the connection plate than by turning the worm gear, it was still very much possible.

3.8.3 Clutch functionality.

The normal operation of the joint was tested by turning on each of the clutches. The motor was set to rotate at 60 rpm, as this gave a convenient output speed.

When fully powered (i.e. when powered with 24V), the clutches did turn the joint. The bottom clutch worked quite well, but the top clutch turned the joint very slowly and sometimes not at all. When the top clutch was applied, the o-rings slipped over the cogwheels, thus giving less joint rotation.

By applying a smaller voltage to the clutch it was attempted to induce slipping. When the voltage was slowly increased from 0V and the motor was not running, the top clutch did not connect until 19V. When the voltage then was slowly decreased, the clutch disconnected at 4.5V. The bottom clutch was always connected.

Because the o-rings tended to slip more easily than the clutches, slipping was induced by stopping the cogwheel connected to the clutch by hand. This way, when then top clutch was connected, it was possible to induce slipping up to 16V when the shaft rotated at 60rpm. The bottom joint slipped even when it was not powered at all, and it was not possible to induce slipping by holding the cogwheel with the fingers at more than 6V.

The clutches became worn during this process, especially the bottom clutch, and fine metallic dust could be seen under them.

The universal joint tended to fall out when the joint moved. This was because it was not fastened hard enough to the main shaft, and moved backwards (toward the clutches) during

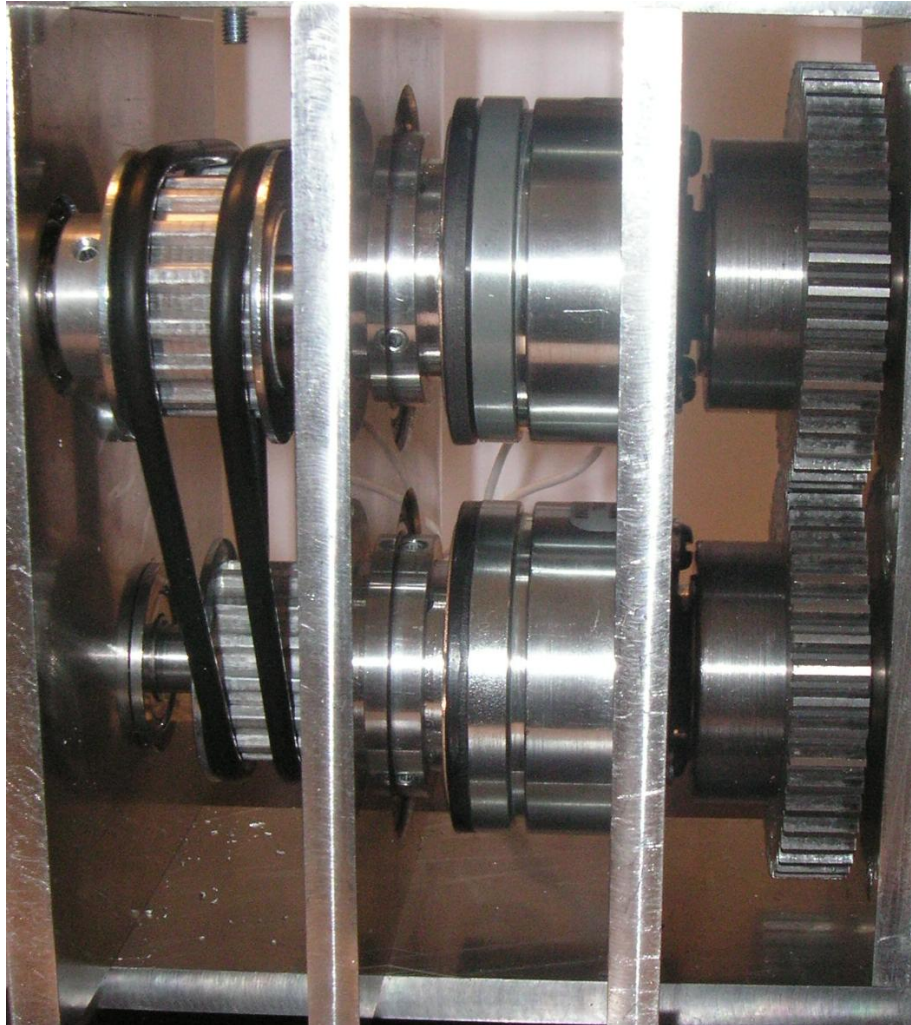


Figure 3.12: The assembled clutches, note that the bottom clutch is assembled a bit too tight

movement. When the universal joint reached the support wall next to the transmission belt cogwheels it stopped falling out.

Appendix D contains a video that shows the joint as it moves when the bottom clutch is turned on and, as well as a video that shows how the o-rings slip when the top clutch is turned on.

3.8.4 Potentiometer

The potentiometer was mounted on top of the joint rotation shaft as seen in Figure 3.9. To calibrate the potentiometer, the voltage was measured at different angles, and the resistance of the potentiometer calculated according to equation (3.2). Table 3.3 shows the measured and calculated values. Note that the angles are measured clockwise around the joint axis of rotation.

Table 3.3: Potentiometer calibration: Angles, v_R and R_{pot}

Angle	60°	45°	30°	15°	0°	-15°	-30°	-45°	-60°
v_R	0.63V	0.67V	0.71V	0.76V	0.83V	0.90V	0.98V	1.08V	1.19V
R_{pot}	8261Ω	7697Ω	7196Ω	6645Ω	5984Ω	5426Ω	4886Ω	4323Ω	3813Ω

The 'Basic Fitting' tool in MATLAB were used to fine a linear approximation to the values. The expression found was

$$\text{Angle} = 0.0266R_{\text{pot}} - 160.56 \quad (3.3)$$

Figure 3.13 shows the plot of angles and resistances along with the linear approximation.

The voltage over R were read using LabVIEW[17], and the LabVIEW diagram can be found in Appendix D. The voltage is low-pass filtered using a butterworth filter with cut-off frequency 0.05. This removed noise. The resistance of the potentiometer and the angle of the joint rotation was then calculated, and the angle was displayed in a graph and in a numeric indicator.

3.8.5 Maximum and minimum rotation angle

The potentiometer was used to find the maximum angle of rotation for the joint. The maximum angle was said to be the angle at which the motor was no longer able to rotate the main shaft. The test was repeated for maximum and minimum angles, and the results are given in Table 3.4.

Table 3.4: Minimum and maximum angles

Test	Minimum angle	Maximum angle
1	-76°	52°
2	-62°	62°
3	-72°	72°
4	-65°	66°

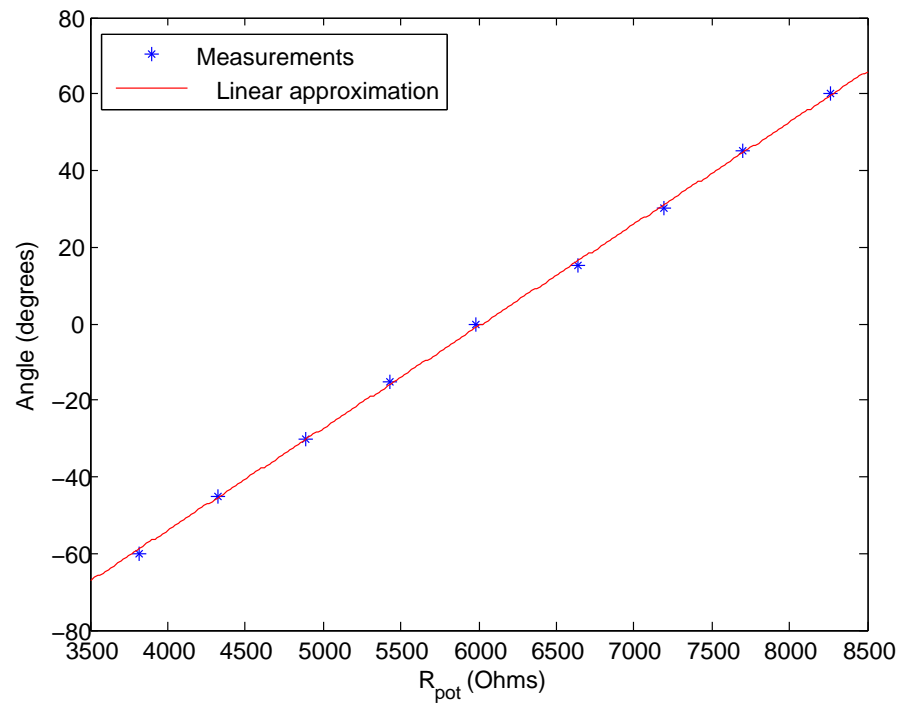


Figure 3.13: Potentiometer calibration: Angles vs R_{pot} and linear approximation

4 Simulator

This section describes the simulator developed in this report. The specifications of the simulator are derived from the assignment text and from literature and are listed in Section 4.1. The kinematic model used is then explained in Section 4.2, and the mathematical and graphical representation of the robot is explained in Section 4.3. Finally a description of the developed simulator is given in Section 4.4, and an overview of the files used are given in Section 4.5.

The simulator is evaluated in Section 5.2, where good sides and shortcomings are discussed. Further work that can be done on the simulator is discussed in Section 6.2. The further work is for a large part based on the evaluation, and if this is done the specifications of the simulator will be fulfilled.

4.1 Simulator specifications

This section contains the specifications of the simulator. The foundation of these specifications is derived from the assignment text, which states that a mathematical model of the robot should be developed. It further states that the model should be used in a simulator, and the model and simulator should have the following specifications:

- The model should be able to take in different parameters, such as material density or points of weight representing actuators, instrumentation etc.
- The simulator should give the user the ability to control the robot.
- The simulator should provide the user with measurements of joint torques, velocities and positions(angles).

In [21] a set of specifications for a general simulations tool is suggested. These are:

- Basic specification for the parameters of manipulators.
- Easy integration of standard CAD models describing the geometric parameters.
- Visualization of the robots movements to evaluate its performance.
- Flexible algorithms for kinematic and dynamic simulation, allowing calculations for several robotic systems.
- Modular approach for a combination of different kinds of robots and grippers.
- Simple definitions of mounting points (points where tools are mounted), tool center points and manipulator bases.
- Interfaces to secure the functionality of the system in various software applications.

The ability to give basic specifications the manipulator parameters coincides with the specification that the user should provide material density, points of weights etc. It is natural to extend

this to the specifications of dimensions of the robot's parts, and of the kinematic relationship between the robot's joints.

Flexible algorithms for kinematic and dynamic simulation gives the simulator the ability to not only provide data on joint torque and kinematics, but also to be used on several different robotic systems. This is in accordance with the rule of thumb of software design that everything should be as general as possible. The modular approach further strengthens the generality of the simulator. So does the requirement of the systems functionality in different software applications, however this is of no direct consequence to the behavior of the simulator and was thus deemed outside the scope of this report.

While the assignment text does not mention visualization of the robot, it is clear that such a visualization will significantly improve the usefulness of the simulator, and ease user control. The advantages of a 3d visualization is also pointed out in[11], where it is stated that by showing the simulation in a 3d view, the simulator can provide a testbed for the development and evaluation of high level control software, as it gives an easy and intuitive feedback of the robots behavior.

It is clear that the simulator should provide a 3d visualization of the robot, providing the user with easy overall information of the robot's performance. The visualization can also serve as an interface where the user can enter commands. However, as is pointed out in [5], for more detailed performance analysis it is necessary also to log data in a text-based observer, where it can be stored for further inspection.

The simulation environment can separate high-level and low-level control, thus allowing separate testing and tuning of these[11]. This would be advantageous as the high level control can be changed between direct user input, reading joint values from a file or other alternatives, all without affecting the lower level control software.

In addition to these requirements, the simulation system will have to include possibilities for the robot to be mobile and to move about in and interact with an environment (such as a ladder).

The final specifications of the robot are listed below:

- The simulator must be able to simulate:
 - The kinematics of each arm.
 - The dynamics of each arm.
 - The movement of the entire robot.
 - The immediate environment of the robot, e.g. a ladder the robot is climbing in.
- The simulator must be able to visualize the robot in a 3d visualization.
- The simulator must be able to log to robots movements and torques in a text-based observer.
- The simulator must implement high and low level control. The control algorithm should be of a modular structure so that they are easily interchangeable.
- The user should be able to specify:
 - The number of the joints in each arm.

- The size of and kinematic relationship between the joints in the arms.
- The weight and location of mass centers (representing actuators, sensors or other payload) in the robot.
- The mass density of the shell of the robot.
- Whether or not a gripper is holding on to its environment (i.e. it is unmovable).
- A reference velocity or angle for each joint.
- A reference velocity or position of the end effector of a chosen arm.

4.2 Modeling of robot kinematics: The Denavit-Hartenberg convention

To represent the kinematic (i.e. rotational and translational) relationship between two links of a robotic serial manipulator, the Denavit-Hartenberg convention can be used[3][8][23]. There are several equivalent notations for the Denavit-Hartenberg convention, differing for instance in joint numbering. This report will use the notation in [8] and [23].

The Denavit-Hartenberg convention establishes coordinate systems (or frames) for each link, described by three orthonormal vectors ($\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i$) so that:

- the \mathbf{z}_{i-1} axis points along the axis of motion of the i th joint
- the \mathbf{x}_i axis is normal to the \mathbf{z}_i axis and pointing away from it
- the \mathbf{y}_i axis completes the right-hand coordinate system

From these coordinate systems four parameters, θ_i , d_i , \mathbf{z}_{i-1} and α_i are obtained. These parameters are known as the Denavit-Hartenberg parameters and completely describe the kinematic relationship between two adjacent links:

- θ_i is the angle from the \mathbf{x}_{i-1} axis to the \mathbf{x}_i about the \mathbf{z}_{i-1} using the right hand rule
- d_i is the distance from the origin of the $(i-1)$ th coordinate frame to the intersection of the \mathbf{z}_{i-1} with the \mathbf{x}_i axis along the \mathbf{z}_{i-1} axis
- a_i is the length of the common normal between the \mathbf{z}_{i-1} axis and the \mathbf{z}_i axis
- α_i is the angle between the \mathbf{z}_{i-1} axis and the \mathbf{z}_i axis about the \mathbf{x}_i axis

Figure 4.1 shows the coordinate systems and Denavit-Hartenberg parameters of two adjacent links. Note that the parameter θ_i is called ϑ_i in the figure.

If the links are connected by a revolute joint, then the parameter θ_i is variable and dependent on the joint value. If the links are connected by a prismatic joint, then the parameter d_i is dependent on the joint value.

Once these parameters are obtained, it is possible to move from one coordinate system to the next by the following steps:

1. Rotate θ_i radians about the \mathbf{z}_{i-1} axis using the right-hand rule.

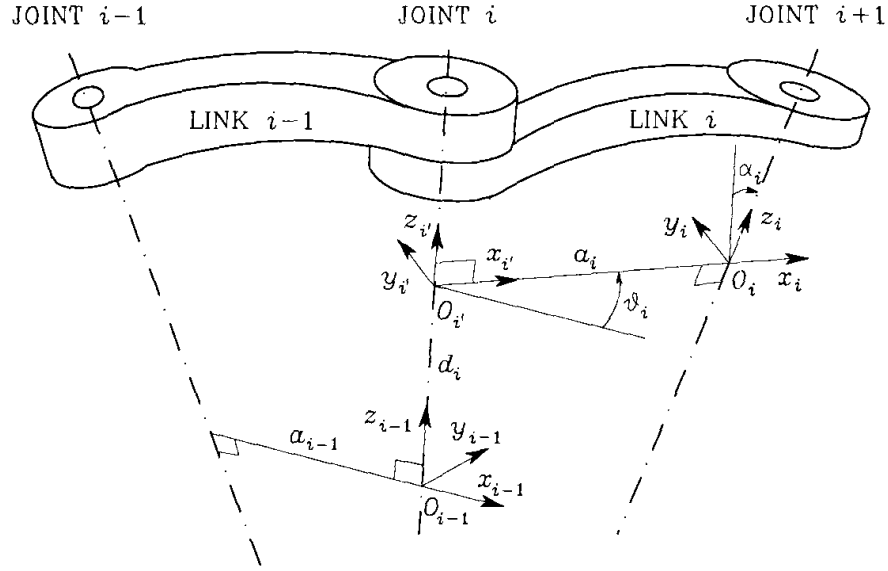


Figure 4.1: Link coordinate systems and Denavit-Hartenberg parameters for two connected[23]. Note that θ_i here is called ϑ_i

2. Translate a distance d_i along the \mathbf{z}_i axis.
3. Translate a distance a_i along the \mathbf{x}_i axis.
4. Rotate α_i radians about the \mathbf{x}_i axis using the right-hand rule.

These movements are represented by a homogeneous transformation matrix T_{i-1}^i as follows:

$$T_{i-1}^i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

In this matrix the upper left 3×3 matrix is a rotation matrix describing the rotational relationship between link $i - 1$ and link i , while the upper right 3×1 vector describes the translational relationship, i.e. the position of link i in the coordinate system of link $i - 1$. Note that

$$T_i^{i-1} = \text{inv}(T_{i-1}^i) \quad (4.2)$$

T_{i-1}^i describes the kinematic relationship between link $i - 1$ and link i . $T_{i-2}^i = T_{i-2}^{i-1} T_{i-1}^i$ describes the relationship between link $i - 2$ and link i . Thus

$$T_0^i = T_0^1 T_1^2 \cdots T_{i-1}^i = \prod_{k=1}^{k=i} T_{k-1}^k \quad (4.3)$$

describes the relationship between the origin and the i th link. If the i th link is the end effector, T_0^i gives the position and rotation of the end effector with respect to the origin.

Figure 4.2 illustrates how the Denavit-Hartenberg convention is used on a PUMA robot. The PUMA robot only has revolute joints, and it is clear from the figure how the parameters θ_i depends on the joint values.

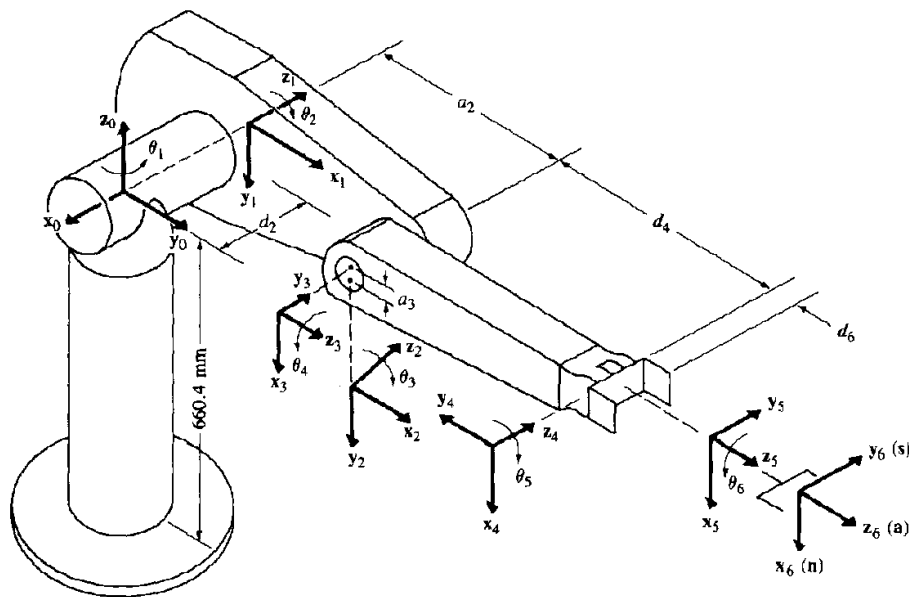


Figure 4.2: Link coordinate systems and Denavit-Hartenberg parameters for a PUMA robot[8]

4.3 Representation of the robot

4.3.1 Graphical representation

The graphical representation of the robot in the simulator was kept simple and general. It would be premature with a complex graphical design in the simulator, as it is not yet clear how the final design of the robot will be. With a simple graphical representation the simulator still shows the overall behavior of the robot. Further, it makes the programming process significantly easier.

The graphical representation consists of two main elements; the body and the joint. Both were made in Autodesk Media & Entertainment's 3ds Max[1], the same program used in [28]. The body was represented as a box called 'Body' with sides $10 \times 10 \times 10$ units, with one unit representing 1 cm in the real world. The joint was represented as a sphere with radius 1 half merged with a cylinder with radius 1 and height 1. The sphere was called 'Base' and the cylinder 'Link', and they were put in a group called 'Joint'. Figure 4.3 shows the two elements.

The 3ds models was exported to VRML (Virtual Reality Modeling Language) which is used by the virtual reality toolbox of MATLAB[26]. The VRML files can be found in Appendix D, where the body is in the file `body.wrl` and the joint in `joint.wrl`. Each 3ds object is

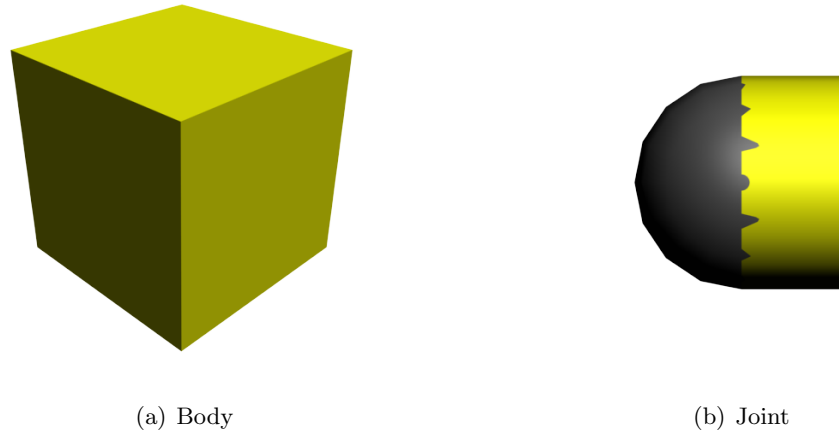


Figure 4.3: Graphical representation of the body and joint

represented as a `transform` object in these files. The `Joint` group, for instance, begins with `'DEF Joint Transform'` and includes a `Base Transform` and a `Link Transform`. Note that the translation of the `Link Transform` (which moves the cylinder relative to the `Joint Transform`) was changed manually from `0 0 0` to `0 0.5 0` to move the link so that it cover the last half of the sphere and not the middle part.

When opening a `.wrl` file using the command `w=vrworld('filename')` in MATLAB a virtual reality world is created with the handle `w`. This object contains several nodes, called `vrnodes`. These are obtained by writing `nodes(w)`. A `vrnode` can represents a graphical object in the virtual world (e.g. a `Transform`), and has fields describing its position, rotation and scale. There are several other fields, and other kinds of `vrnodes`, but they are not used in this report.

The virtual world obtained by opening the `joint.wrl` file includes the nodes `Joint`, `Base` and `Link`. Since the `Base Transform` and the `Link Transform` are part of the `Joint Transform`, they will move and rotate as the `Joint` node is moved and rotate.

To create a world that contains the body and all the joints of the robot's arms, a MATLAB function was written that makes a copy of the `Body Transform` from the `body.wrl` file to the file `robot.wrl`. It then adds N copies of the `Joint Transform` from in `joint.wrl`. N is the total number of joints of the robot. The joints are numbered, so that the `Joint Transform`'s are called `Joint1` to `JointN`, the `Base Transforms` and `Link Transforms` are numbered similarly. The MATLAB function can be found in the file `makeRobot.m` (see Section 4.5 and Appendix D).

By creating a virtual world from `robot.wrl`, MATLAB has access to one `vrnode` for the body and one for each joint. This joint can be positioned, rotated and scaled according to the robot's parameters and current kinematic configuration.

4.3.2 A mathematical chain representation of the robot

A way to mathematically represent a complex robot is as a chain of links connected by joints. A rigorous definition of such chains are given in [21] where they are used as a foundation for the development of a general simulation software for robots. The chain is said to consist of elements, which represent a base (a non-moving part of the robot typically connected to the ground) or a link. The chain can be

- a simple open chain, such as the common robotic manipulator used in industry
- a general open chain where one joint can be connected to several links
- a simple closed chain, such as a parallell robot
- a partly closed chain, which contains both open and closed subchains

Figure 4.4 shows these different types of chains.

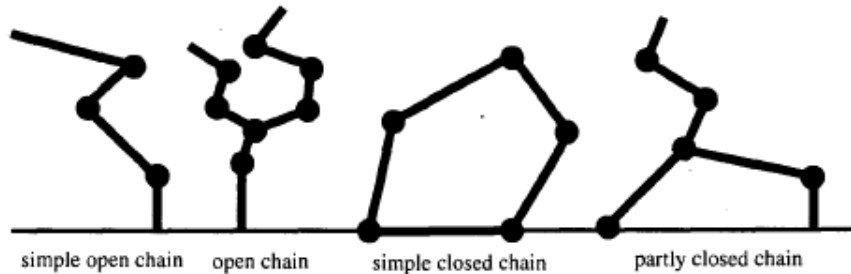


Figure 4.4: Different types of chains [21]

The elements of the chain have certain attributes that describes their relationship to the other elements:

- Every link is connected to a joint, whose rotation around or translation along the joint axis is represented by the attribute j_{value} .
- A set \mathcal{S} that lists all the successors of a link and the relationship between the link and its successors. The Denavit-Hartenberg parameters can for instance be used to describe this relationship.
- To separate the links they an attribute l_{name} for the unique name of each link.
- If a link is part of a closed or partly closed chain, the joint movement of that link might be dependent on other joint movements. Such dependencies are described in the set \mathcal{D} .
- Limitations on the link, such as minimum and maximum joint values, are described in the set \mathcal{E} .

4.3.3 Chain elements used in the simulator

The chain described in the previous section is well suited to implement in an object oriented programming language, as was done in [21]. However it is also possible to implement it in a procedure oriented language like MATLAB, which was used in this simulation. In the simulation, three different types of elements were used, namely base, joint and end effector. The joint element is much the same as the link element of [21].

To implement each element of the chain, a struct was created with fields representing the different attributes. The chain was then implemented as an array of these structs. The fields of the structs were:

- **name** - The name of the element, e.g. "Joint1" or "Body". This is the equivalent of l_{name} in [21]. Note that all elements in the chain, not just the joints, are given a name. This is for convenience, as the robot may have more than one base and end effector.
- **dependents** - If the movements of this joint is dependent on the joint movements of other links it is described here. This is the equivalent of \mathcal{D} in [21]. Currently, it is not used by the simulator, however it might become useful for further work (see Section 7.2)
- **limits** - This field represents the limits on the joint movement. In the simulator, it has the format

$$[\min_{\text{pos}} \quad \max_{\text{pos}} \quad \min_{\text{vel}} \quad \max_{\text{vel}}]$$

where \min_{pos} and \max_{pos} is the minimum and maximum position in radians, and \min_{vel} and \max_{vel} is the minimum and maximum position in radians/second. This is the equivalent of the attribute \mathcal{E} in [21].

- **denavit** - This is a $n \times 5$ matrix that holds the Denavit-Hartenberg parameters for the kinematic relationship with the n direct successors of this element. The format of each 1×5 row vector is

$$[index \quad \theta \quad d \quad a \quad \alpha]$$

where $index$ is the index of the succeeding element in the chain, and the other parameters are as described in Section 4.2. This field is the equivalent of the attribute \mathcal{S} in [21]

- **type** - This is an integer that tells what type of element this is. The integer is
 - 1 if the element is a base
 - 2 if the element is a normal joint
 - 3 if the element is an end effector
- **transmat** - This is the translational matrix from origo to element.
- **vrNode** - This is a struct with the elements `moveNode` and `scaleNode`. `moveNode` is an array of MATLAB `vrnode` objects that must be moved as this element moves. `scaleNode` is an array of MATLAB `vrnode` objects that must be scaled according to the size of this element.

For example, the normal joints of the robots arm has a MATLAB `vrnode` object for the `Joint Transform` that represents the entire joint as the only element of the `moveNode` array, while it has one MATLAB `vrnode` object representing the spherical part (base) and one representing the cylindrical part (link) of the joint in the `scaleNode`. When scaling the joint, the radius of both the base and the link are increased or decreased to the value specified by the user. The length of the link is then increased to be

$$L_{\text{joint}} - r \quad (4.4)$$

where L_{joint} is the total length of the joint and r is the radius. This is because the first r of the joint will be the first half of the sphere, and the rest will be the cylinder. Because of this, length of the joint must be at least twice the radius.

If this element represents an imaginary joint or an end effector it has no graphical representation, and `vrNode` is empty.

- **dimensions** - This is an array containing the dimensions of the element. If the element is the body of the robot, the array contains the width, height and thickness of the body, if it is a joint it contains the length and the radius of the joint. This field is empty if the element represents an imaginary joint or end effector.
- **changed** - Used to see if this element has changed from one time step to the next. 1 if this element has changed 0 if not. If the element has changed, the kinematics of the element and of all the following elements in the arm must be updated. This variable makes it unnecessary to update elements that has not changed, thus saving computation time.

The attributes of the implemented elements and of the chain elements in [21] are different. The differences are:

- **$jvalue$** - To avoid representing the same value twice, the simulator inserts the joint value directly into the Denavit-Hartenberg parameters, and thus it does not need a field for the value.
- **vrNode and dimensions** - These elements are used for the graphical representation of the robot and is not mentioned in the mathematical representation of [21].
- **transmat** - The translational matrix of an element can be calculated using the Denavit-Hartenberg parameters of the previous elements. Strictly speaking, this means that the position and orientation of the element is already represented. However, to avoid having to calculate the translational for all joints in an arm if just the last joint is moving, the translational matrix is stored. This saves computation time.
- **type** - In [21] the bases and links are separated by being stored in two different sets. While it was possible to do this for bases, joints and end effectors in the simulator, it was deemed more convenient to simply separate them by this value. This way it is also convenient to change the type of an end effector to a base, which can be useful in further work when the body of the robot should be able to move. This is discussed in Section 7.2.

- **changed** - Since this variable is used for pure time-saving purposes, no such variable is mentioned in [21].

4.3.4 Chain structure

The robot's body is represented as a chain with five elements, see Figure 4.5. The central element represents the body itself, and contains the dimensions of the body as well as the translational matrix of the center. The four outer elements are virtual joints with zero dimensions and no `vrnode`. Their purpose is to provide a constant rotation around the axis pointing out of the paper, so that the arms of the robot point horizontally outwards at zero joint values as seen in Figure 4.8. Thus they have a purely mathematical function.

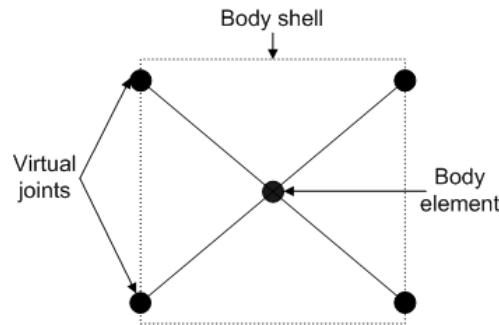


Figure 4.5: Chain representation of the robot body

The Denavit-Hartenberg parameters connecting the elements show the kinematic relationship from the beginning of one element to the beginning of the next (e.g. from the beginning of one joint to the beginning of the next). Thus it is necessary with an element with zero dimensions and no `vrnode` at the end of each arm, representing the end effector. These elements provide the translational matrices for the tip of the arms, and can be useful for inverse kinematics in further work.

The order of the elements within chain is:

1. The first element is the central body element
2. The next 4 elements are the virtual joints, starting from the top right and proceeding clockwise
3. The $N_1 + 1$ next are the N_1 elements representing the joints of the first arm, starting with the innermost joint and ending with the outer. The last element in this group is the end effector.
4. Similarly the $N_2 + 1$, $N_3 + 1$ and $N_4 + 1$ next elements represents the joints and end effectors of the second, third and fourth arm. As with the virtual joints, the first arm is the one to the top right, and the counting then proceeds clockwise.

In summary, the elements of the chain are ordered like this:

$$[\text{Body} \quad [4 \text{ Virtual joints}] \quad [\text{Arm } 1] \quad \dots \quad [\text{Arm } 4]]$$

In general, the placement of the different elements within the chain should be of no consequence, as the relationship between them are described in attributes such as `denavit` and `dependents`. However, it was convenient to place the elements in the order described above. The files that uses this order are `simulateDynamics.m` and `updateKinematics.m`. Further, the file `createChain.m` uses the structure since the chain is created here. These files can be found in Appendix D, and a description of the functions can be found in Section 4.5.

4.4 Simulator description

This section gives a brief overview of the functionality of the simulator as it is at the time of writing. A signal flow chart of the simulator is shown in Figure 4.6. It shows a modular structure, where high level control, low level control, dynamic simulation and kinematic simulation are separated. Such a modular structure makes it possible to focus on one part at a time. It also enables testing of different control algorithms etc. without unnecessary software complications. Note that the sending of signals is done by storing a global variable called `simulator` containing the relevant data. This variable is described in Section 4.4.1

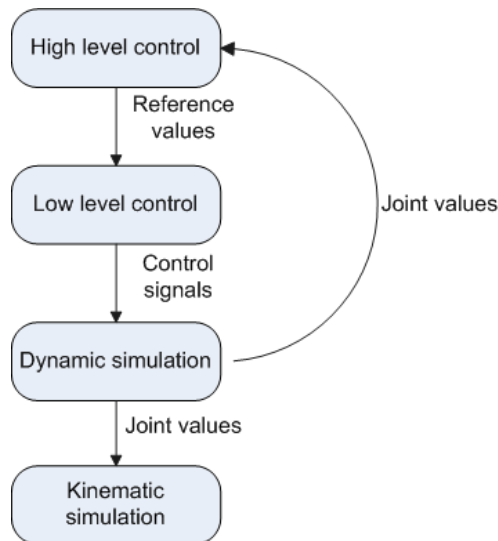
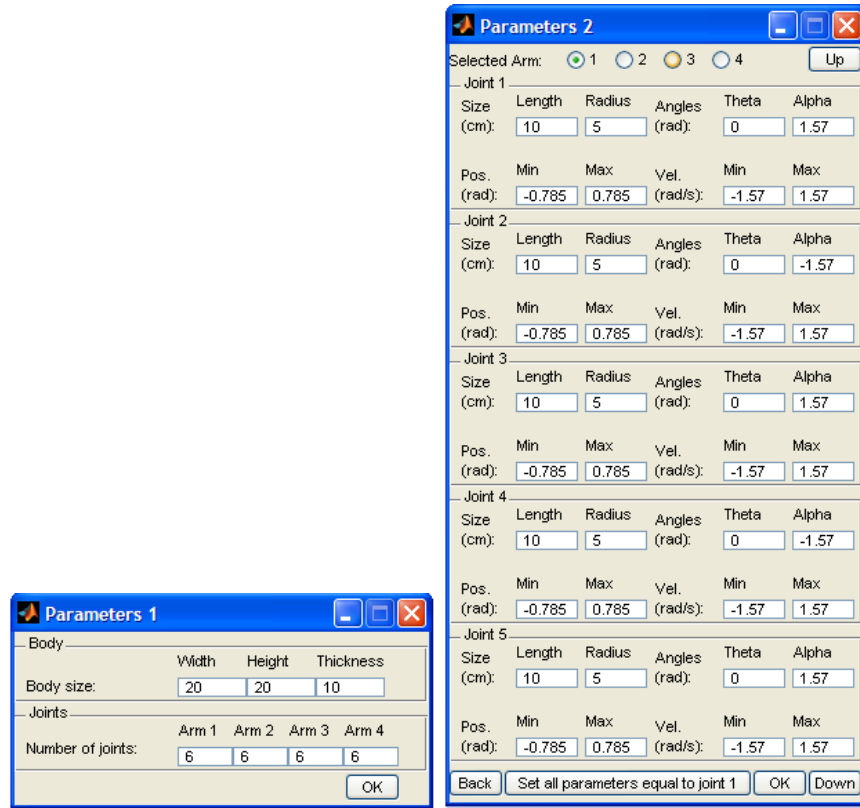


Figure 4.6: Signal flow chart of the simulator

The simulator uses MATLAB [26] as this language both provides a virtual reality toolbox and an extensive mathematical library. The main disadvantage of using MATLAB is that it is not object oriented. The structure of the simulator makes it well suited for implementing in an object oriented language. Another possible drawback is that a more low-level language like C might execute the program be faster. It was decided that the advantage of having MATLAB's mathematical library outweighed the drawbacks, and hence MATLAB was used for the simulator.

4.4.1 Running the simulator

To start the simulator, the file `main.m` is run. This opens the first dialog window in Figure 4.7 where the user can specify the size of the body (in cm) and the number of joints in each arm. Arm 1 is defined as the upper right arm, and the counting then proceeds clockwise.



(a) Dialog window 1

(b) Dialog window 2

Figure 4.7

When the OK button is pressed, the second dialog window in Figure 4.7 opens. In this window the user specifies which arm to edit by selecting the corresponding radio button at the top. Each joint of the selected arm is given the following parameters:

- The size of the joint. The joint are modeled as cylinders with a round end (see Section 4.3.1), and the user can specify the length of the joint and the radius of the cylinder in cm. Note that the length must be at least twice as much as the radius. This limitation comes in part because the rotation of the joint would be limited if the length was too short, and mainly because the graphical representation demands it.
- The θ and α parameters of the Denavit-Hartenberg convention, as described in Section 4.2. These angles are given in radians. The θ value is regarded as an offset to the joint value.

- The minimum and maximum angle of the joint with respect to the θ parameter. If θ is $\frac{\pi}{2}$, then the actual upper limit of the joint is $\frac{\pi}{2}$ plus the angle written in the maximum angle field, and similarly for the actual minimum value. These values are in radians
- The minimum and maximum joint velocities, in radians per second.

The default values of the first and second dialog window corresponds roughly to the robot designed in [28].

The back button of the second dialog window takes the user back to the first. The 'Set all parameters equal to joint 1' sets parameters of all the joints in that arm to that of the first joint. The up and down button (which only appears if there are more than 5 joints in the selected arm) scrolls the list of joints up and down.

The Ok button starts the simulation and opens a window showing the 3d visualization of the robot and a dialog window for velocity control. These are shown in Figure 4.8. Each of the sliders in the velocity control window sets the velocity of a joint. The velocity can also be set in the text field next to the slider. The minimum and maximum velocity of that joint are as specified previously. As with the second dialog window, the user can choose which arm to control using the radio buttons at the top. The stop button sets all the velocities of the current arm to zero. Closing this window will stop the simulator.

The simulation creates a struct called `simulation` with the following fields:

- `sampleTime` - The sampling time of the simulator, by default set to 0.1.
- `logLimit` - The number of time steps the log remembers. By default set to 600.
- `logStep` - Which step the log currently is on. Cycles between 1 and `logLimit` during the simulation, so that if `logStep` is 20 the last `logLimit` steps are 19, 18 , \dots , 1, `logLimit`, `logLimit`-1, \dots 21.
- `log` - A struct that contains the angles for each joint at the last `logLimit` timesteps.
- `jReferences` - Contains the current reference signals of the joints. Used by the low level control.
- `controlSignals` - Contains the current control signals sent from the low level control to the dynamic simulator. Currently, neither low level control nor the dynamics are simulated, so this field holds the same values as `jReferences`.
- `numberOfJoints` - An array containing the number of joints of each arm. Used by the software at different levels.
- `referenceType` - An integer that equals 1 if the joint references are position (angle) references, 2 if they are velocity references and 3 if they are torque references. Currently, this value is always 2 as it is the velocities of the joint that are controlled.
- `dialog` - A handle to the velocity control dialog window. Used by the software at different levels.

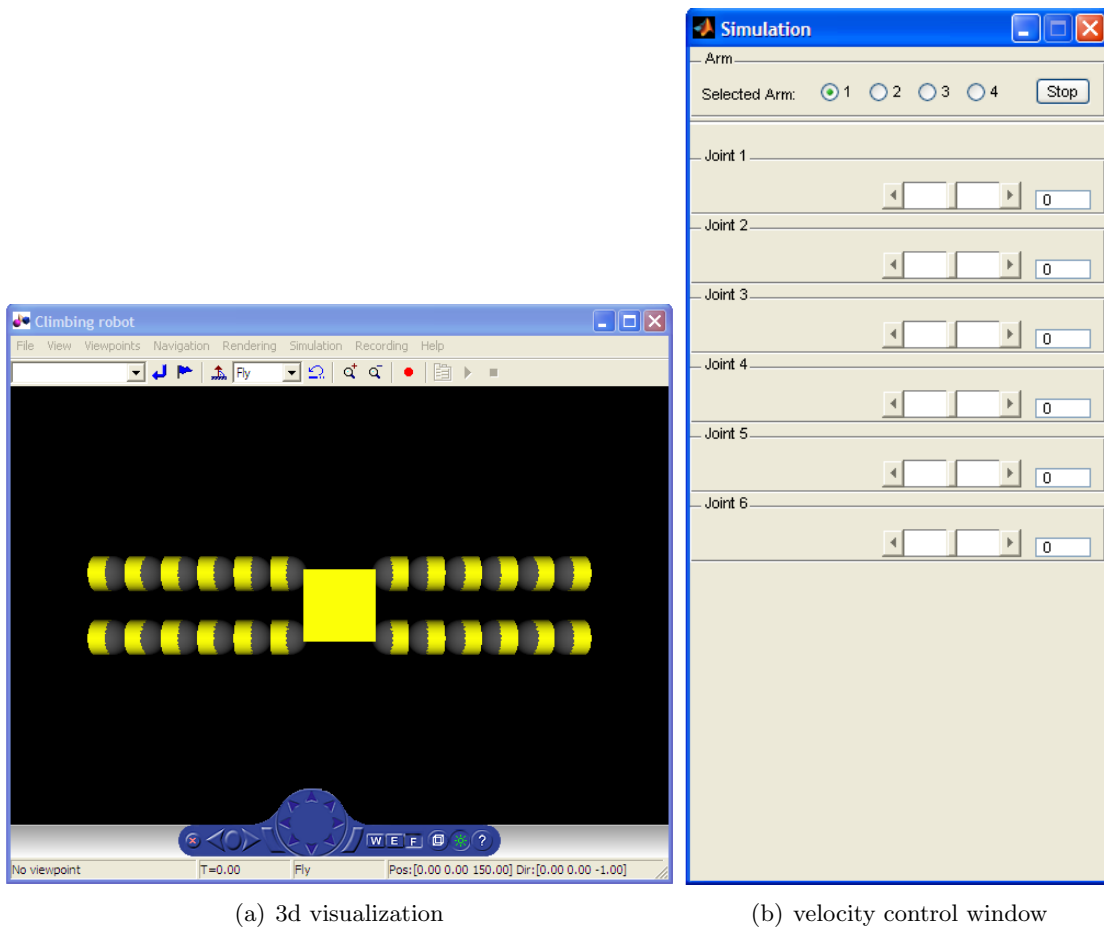


Figure 4.8

- `running` - Equals 0 if the simulation is not running and 1 if the simulation is running. Used to stop the simulation loop.

The variables `sampleTime` and `logLimit` can be set in the file `main.m`.

4.4.2 Unfulfilled specifications

Section 4.1 gave a list of specifications of the simulator. Some of these were achieved, while others are left for further work. The currently unfulfilled specifications are:

- The simulator does not simulate:
 - The dynamics of the arm.
 - The movement of the entire robot.
 - The immediate environment of the robot.
- The simulator is not able to log the torques of the joints.
- The simulator does not implement low level control.
- The user cannot specify:
 - The weight and location of mass centers (representing actuators, sensors or other payload) in the robot.
 - The mass density of the shell of the robot.
 - Whether or not a gripper is holding on to its environment.
 - A reference velocity or position of the end effector of a chosen arm.

This is discussed in Section 7.2. The simulator has room for low level control in the file `lowLevelControl.m`, and for dynamical simulations in the file `simulateDynamics`. Section 4.5 provides a more in depth description of these and other files used by the simulator.

4.5 Software structure

This section gives an overview of the software structure of the simulator, and also a brief description of each MATLAB file that is used. This section can be useful as a reference when further work on the simulator is to be done.

Figure 4.9 shows an overview of the files and their dependencies. A file in the diagram runs the files immediately to its right, so that `main.m` runs `chooseParameters.m`, `createChain.m` and so on. The file `updateFromElement.m` runs itself recursively.

The files are run from the top down, so that `main.m` first runs `chooseParameters.m`, then `createChain.m` (which again runs `createElement.m`) and then `makeRobot.m`. Loops are not included in the diagram, but it is worth noting that the main simulation loop goes from `updatePositions.m` to `simulateDynamics.m`.

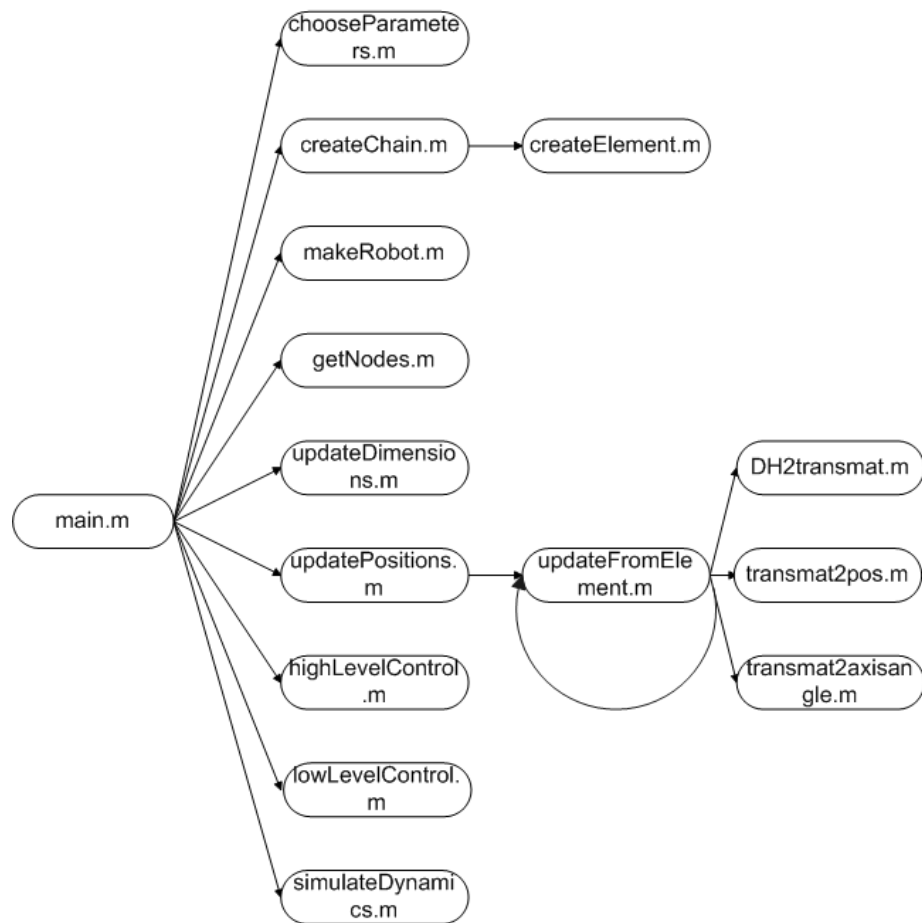


Figure 4.9: Overview of the files and their dependencies

4.5.1 Alphabetic list and description of files

A description of each of the MATLAB files that is used is given below. The code itself can be found in Appendix D.

- `chooseParameters.m` - This file opens the first and second dialog window seen in Figure 4.7, and stores the parameters entered by the user in these.
- `createChain.m` - This file creates the chain as described in Section 4.3.4, using the parameters stored by `chooseParameters.m`.
- `createElement.m` - This file creates a single element of the chain and is used by `createChain.m`. It creates the element by making a struct as described in Section 4.3.3.
- `DH2transmat.m` - This file creates a translational matrix from Denavit-Hartenberg parameters described in Section 4.2
- `getNodes.m` - This file gives a `vrnode` to the body element and each of the joint elements (i.e. the elements with nonzero dimensions). The `vrnodes` are made by `main.m`. Each element that receives a node stores it in the `vrNode` field described in Section 4.3.3
- `highLevelControl.m` - If the window where the user inputs joint velocities, shown in Figure 4.8, is not open, then running this file will create and open it. When the window has been opened, or if it was already opened, the file will read the input and store them as reference signals used by `lowLevelControl.m`. The joint reference signals are stored in the `jReferences` field of the `simulator` struct described in Section 4.4.1.
- `lowLevelControl.m` - In this file it is possible to implement a low level control algorithm, such as a PID regulator. The file is intended to use the joint reference signals produced by `highLelevControl.m` and update the control signals stored in the `controlSignals` field of the `simulator` struct described in Section 4.4.1. Currently, this files copies the values from the `jReferences` field of the `simulator` struct to the `controlSignals` field.
- `main.m` - This is the main file that runs the files for creating the chain representation of the robot and the VRML file containing the virtual world, it opens the virtual world and starts the simulation loop.
- `makeRobot.m` - This file create a VRML file (`.wrl`) with one `Body Transform` from the file `body.wrl` and N `Joint Transforms` copied from the file `joint.wrl`. See section 4.3.1 for more on VRML.
- `simulateDynamics.m` - This file is intended to simulate the dynamics of the robot, and future code for this should be put here. Currently, the file just increases the joint values by the values stored in the control signal field of the `simulator` struct (which is the joint velocities) multiplied with the timestep. This file also updates the log.

- `transmat2axisangle.m` - This file makes an axis-angle representation (which is used by MATLAB's Virtual Reality Toolbox) from the orientation described by a transformation matrix. The axis-angle representation is a vector with four elements, where the first three describes an axis of length one and the last is the rotation angle around this axis in radians, using the right-hand rule.
- `transmat2pos.m` - This file makes a position vector from a transformation matrix.
- `updateDimensions.m` - This file scales the `vrnodes` of the body and joints so that the size of the 3d representation of the robot is in accordance with the parameters given by the user. The scaling process is described in Section 4.3.1.
- `updateFromElement.m` - This file will update the position and rotation of the `vrnode` of an element, if the element has changed (the `changed` field is 1). It then updates the translational matrices of the direct successors of the element according to the Denavit-Hartenberg parameters stored in the `denavit` field, and calls itself on each of the successors. The fields of the struct making up the elements is described in Section 4.3.3
- `updatePositions.m` - This file finds all bases in the chain (currently just one) and calls `updateFromElement.m` on these.

5 Evaluation

This section evaluates both the joint prototype and design developed in Section 3, as well as the simulator developed in Section 4.

5.1 Evaluation of the joint design

This section gives an evaluation of the joint design and prototype developed in Section 3. Drawbacks as well as advantages in the design is pointed out, and the results of testing on the prototype are evaluated. The discussions are used as a foundation for recommending areas of further work, as is done in Section 6.1.

The gear ratio of the prototype is 11.25 : 1, which is lower than the ideal ratio of 66.7 : 1. This reduced the output strength of the joint, and the motor must to be run at a lower speed to reduce the output speed of the joint. However, for the prototype it was acceptable with a lower gear ratio, as it was only intended to demonstrate the concept.

The prototype is quite heavy (3420g) and the weighs is much more than the estimated weight of the joint from section 2.2 and 3.3 (842.62g). The weights are not really comparable, as the prototype has not been designed with focus on low weight and includes a stepper motor. As with the gear ratio, the weight of the prototype was not a concern, and the only part of the design process where weight considerations where made was in the selection of the clutch in Section 3.3. However, if the weight is not reduced in future designs it will be a prohibiting disadvantage when it comes to using the joint in a climbing robot.

A drawback with the prototype was the heavy load put on the motor by the input gears. If these gears are slightly reduced in size, or moved a bit further from each other, the loss of efficiency in these gears are likely to be severely reduced. Wearing the gears down a bit is also likely to reduce the work load, indeed it was noticed that the gears went smoother and smoother during the short testing that was done on the prototype.

A major advantage of the final design is the reduction of the number of clutches from three to two. In the final joint design, when the weight of other parts of the joint have been significantly reduced, it is likely that the weight of the clutches will constitute a large part of the total weight of the joint. This reduction in the number of clutches can only be justified if the need for a braking clutch is removed. The worm gears which were supposed to remove this need in the prototype did not provide enough resistance, and some other solution must be found.

The bottom clutch was mounted a bit too tight, as the input and output part were in contact even when the clutch was not powered. This made the joint move unintentionally. The situation improved as the clutch became more worn.

A possible disadvantage of using clutches is the wear and tear they might be subject to when the joint is rotating. It is possible that the clutches must be replaced from time to time, which is both expensive and possibly complex. Further, as the clutches becomes more worn it is possible that the dynamic characteristics will change. This can reduce the efficiency of a control system, unless the system has the ability to adapt to these changes or are robust enough to ignore them.

The final joint design is unsymmetric. This is a disadvantage when mounting several joints together, as it can make the dynamics of the robot less smooth and more difficult to model (it

is for instance incorrect to assume that the centers of mass lies on the center of the main shaft when the joints are unsymmetric).

The use of o-rings in stead of a cogged transmission belt for transferring torque from the clutches to the worm gears severely limits the performance of the joint. The o-rings slips quite easily and the output strength of the joint is thus reduced. This makes it difficult to induce slipping in the clutches, and also to burnish the clutches.

The potentiometer was used only to measure the maximum and minimum angles of the joint. It will become more useful when feedback is introduced. The maximum and minimum angle was measured as the point when the motor was no longer able to rotate the main shaft. This was because the loss of effect over the universal joint gave a too heavy work load on the motor when the joint rotation shaft was rotated too much. The maximum and minimum angles varied a bit. To be on the safe side, the limits of the joint should be set some degrees less than the maximum angle and more than the minimum angle, to avoid blocking the motor. How much will depend on the strength of the motor, but $\pm 45^\circ$ should be a reasonable value, at least for the prototype.

A drawback of the design proposal is that the robot becomes dependent a single motor to work. If the motor fails, the robot effectively loses all power. If the robot was equipped with one or more DC motor for each joint, as is the common solution today, it would be more robust to motor failure (if not to power failure). It is possible that this drawback must be accepted as part of the design concept, unless redundancy is somehow introduced.

The joint is quite general, which makes it possible for the joint to be used in many different robotic applications. It is for instance well suited for use in a robotic manipulator connected to the ground, as the weight of the motor then becomes insignificant (since the motor does not have to be moved). It can also be suitable for use on snake robots, which traditionally consists of many equal joints connected together.

5.2 Evaluation of the simulator

This section gives an evaluation of the simulator developed in Section 4. Advantages, disadvantages and unfulfilled specifications are pointed out and discussed. Possible improvements are discussed in Section 6.2.

The simulator is implemented in the mathematical programming language MATLAB[26] because of the extensive mathematical library it provides. This works quite well, despite the fact that the chain representation of Section 4.3.2 is suited for a more object oriented language. Indeed, the language used to implement the representation by its developers was object oriented[21]. Still, the heavy use of structs makes the implementation work well, even though it might have been more 'elegant' with an object oriented approach.

The chain representation of the robot is an intuitive and general way to represent a robot. By using this representation, it is possible to represent several kinds of robots in the same framework[21]. While the chain representation is not designed for mobile robots, it is possible to modify it to cover these kinds of robots as well, for instance by allowing elements to switch between being end effectors and bases.

The placement of elements within the chain is of no consequence for most of the files. This makes the simulator quite general, and only small adjustments to the code will be necessary

if the simulator is to be modified to simulate other kinds of robots, like robotic manipulators or snake robots. The exceptions are the files `simulateDynamics.m` and `updateKinematics.m`, which is dependent on the structure of the chain as described in Section 4.3.4, and thus make the simulator somewhat less general.

The graphical representation of the robot is quite simple. It is possible to create a more complex and realistic graphical representation of the robot, however for this report it was deemed premature as no part of the robot has a finished design. By displaying a simple and general representation of the robot, the simulator is still able to show the overall behavior, and the complexities of the software becomes considerably less. Further work such as collision detection might become overly complex with a more realistic graphical representation of the robot.

The visualization of the robot can immediately tell the user how the current design parameters will effect the overall look and behavior of the robot. However, while the simulator provides a useful visualization, it is clear that there is much potential for further work on other fields. Several of the simulator specifications remains unfulfilled by the current design. These are listed in Section 4.4.2, and are reproduced here for convenience:

- The simulator does not simulate:
 - The dynamics of the arm.
 - The movement of the entire robot.
 - The immediate environment of the robot.
- The simulator is not able to log the torques of the joints.
- The simulator does not implement low level control.
- The user cannot specify:
 - The weight and location of mass centers (representing actuators, sensors or other payload) in the robot.
 - The mass density of the shell of the robot.
 - Whether or not a gripper is holding on to its environment.
 - A reference velocity or position of the end effector of a chosen arm.

The movement of the entire robot and the reference velocity or position of the end effector of a chosen arm where not included because of the complexities involved with inverse kinematics, such as singularities and non-unique solutions. When the body of the robot moves, it is likely that it is mathematically represented as the end effector of the arms that are currently holding on to the environment, which are likely to be represented as bases. Another possibility is for the body to be represented as a non-moving joint that connects the arms holding on to the environment and the arms that are loose. Either way, inverse kinematics are necessary for controlled robot movement.

Another problem with the movement of the entire robot is that the end effectors currently holding on to the environment (e.g. a ladder) must be represented as bases. When an end

effector of an arm becomes a base, then each element must know the kinematic relationship to the previous element (i.e. the element closer to the body), not the next element as before. This is currently not implemented in the simulator.

Simulation of the immediate environment of the robot was postponed due to time considerations. It is clear that this specification must be fulfilled in order to provide a good testbed for high level control algorithms. To fulfill this specification, it is also required to simulate a gripper. The simulation of the environment must not become too complex and take too much focus and computational time away from the simulation of the robot.

Execution speed is particularly important when it comes to simulation of the robot dynamics. The robot proposed in [28] is quite complex with at least 24 degrees of freedom. It might be problematic to have an extensive dynamic simulation along with an on-line visual representation. For this reason, it was decided to focus primarily on the kinematic simulation, and postpone the dynamics for further work.

Without any simulation of the robot's dynamics, there is no need for the user to specify points of weight or mass densities, and so this too was postponed. Further, without dynamic simulations it is not possible to provide the user with torque measurements.

The simulator provides a good changeability of the high- and low-level control. This makes it convenient to test different high- and low-level algorithms, without having to change the rest of the code. As the simulator is today, there is no low-level control at all, as this becomes meaningless without dynamic simulation. There is room for low-level control algorithms in the file `lowLevelControl.m` described in Section 4.5

The simulator has a log storing the joint values at the N last time steps, where N is a parameter that can be set in `main.m`. This is useful both for retrieval of results and for future dynamic simulations, as these often require knowledge of at least some of the history of the joint movements. It is however possible that the joint values alone is not enough joint data, and that control signals, joint velocities, joint references and joint torques should be logged as well. In addition, it might be confusing that the simulator does not distinguish between the θ parameter and the joint value in the log, so that a joint with a θ value of $\frac{\pi}{2}$ will log $\frac{\pi}{2}$ at zero joint value.

The velocity control of each joint is easy and intuitive to understand and use. However, for complex movements of the arm it is quite cumbersome, as it is difficult to repeat tests accurately. The other dialog windows function well, and can easily be extended to include other parameters. It is not possible to store the parameters of a previously designed robot in the simulator. If the robot differs from the default parameters, this will result in the repetitive and cumbersome task of writing all the parameters for each time the simulation is run.

6 Further Work

This section contains recommendations on further work that can be done on both the joint design and the simulator developed in this report. This work will bring both much closer to a level of completeness where they can be of actual use for the climbing robot. It is possible that the simulator can be of help in designing the joint, and that the dynamic model that can be developed for the joint will be of use in the simulator. Thus, the two subjects covered in this report can ultimately be joined as parts of the overall robot design process.

6.1 Further work on the joint

There is much further work that can be done on the joint design, both to improve the prototype and to make a final design that can be used in a climbing robot. This section recommends some areas where further work can be done, much based on the evaluation of the joint in Section 5.1.

To achieve the desired output strength of the joint, it is clear that a higher gear ratio will be needed than the 11.25 : 1 offered by the worm gears. It is possible to use worm gears with a higher ratio, as for instance Mekanex[15] offers a set of worm gears with ratio 47 : 1. To achieve additional gearing, it is possible to use either normal cylindrical gears, planetary gears or harmonic gears as described in Section 3.1.

A high gear ratio allows the rotational velocity of the main shaft (the input speed of the gearing system) to be increased, thus increasing the power transmitted by the clutches and ultimately the strength of the joint. It is, however, important that the gears can transmit the torque required by the joint. Also, it is important to ensure that the efficiency of the gear is as high as possible, to reduce the amount of power that is used for driving the gears themselves.

The weight of the joint must be reduced, as the prototype is quite heavy (3420g). The only area where weight has been taken into consideration when designing the prototype is in the selection of the clutch in Section 3.3. Thus, there is much potential for weight reduction. This can be achieved by:

- Removing the motor from the joint. The prototype includes the stepper motor and a casing for the stepper motor. Since the main idea behind the joint was to remove the motor, the final joint design will not include this.
- Reducing the dimensions of the support structures. The support structure of the prototype is much larger than necessary, mostly for convenience in assembly. Weight can be reduced both by reducing the thickness of these as well as cutting them to fit the joint more smoothly
- Using lighter gears. The gears currently used by the joint is quite heavy compared to for instance a harmonic gear. The 1 : 1 gearing on the input part of the clutches, which are made of steel, are particularly heavy.
- Using lighter materials. The joint is for a large part made of aluminum, with the shafts and the input gears being made by steel. If a lightweight yet tough material like carbon fiber is used, the weight of the joint may be reduced. A drawback of this is that the joint may become quite expensive.

When reducing the weight of the joint, it is important to ensure that the joint remains sturdy enough to meet the strength requirements of the joint. Some sort of stress analysis (like for instance in [19]) should be done on future designs to ensure that the joint will be able to support itself, to deliver the required output torque and to support the climbing robot without breaking. In the joint design that will be used in the climbing robots, it is necessary for the different parts to fulfill the additional requirements that comes from working in a more or less hazardous environment as well[28]. When the weight of the joint has been reduced, it needs to be compared with the weight requirements and torque requirements on the joint. It is possible that the clutch must be replaced with a more powerful, and thus heavier, one. It is also possible that the clutch can be replaced with a lighter and less powerful one.

The maximum and minimum angles of the joint was determined by the universal joint. This can be avoided by using bevel gears as the transmission mechanism for rotation of the main shaft, as in the designs of sections 3.4 and 3.5. While this solution is more complex when it comes to assembly, it is still likely to be preferable as the loss of effect is not dependent of joint rotation (giving a more predictable behavior) and the limits on the joint rotation decided by the transmission mechanism, but by the structure of the joint or the control system. It is likely that even when the joint rotates to its maximum or minimum value using bevel gears, the motor will not be stopped like with the universal joint.

Control algorithms needs to be made for the clutch, in order to control output angle, velocity or torque. This requires a model of the clutch dynamics, as well as feedback sensors like the potentiometer used in the prototype. This model needs to take the wear and tear of the clutches into account, and must at the same time be simple enough for use in a fast control system. Such a model would also be useful for the simulator developed in Section 4.

For the dynamic model to be correct, it is likely that the positioning of the clutches must be highly accurate. This is particularly so for the air gap between the input and output part. In the assembled joint, one of the clutches was mounted too tight, which shows that this can be difficult to achieve. This error source, and other sources of errors[19], must be taken into account when designing the control system.

As pointed out in Section 5.1, it may be a disadvantage for the joint that it is unsymmetrical. For this reason, it should be attempted to make future designs more symmetrical than the prototype, for instance by ensuring that the main shaft runs through the middle of the joint.

The lifetime of the clutches when used in a joint like this is unknown, and needs to be tested. So must the effect of wear and tear on the dynamic characteristic of the clutches. Finally, the induced friction might induce a high temperature which may damage the clutch or surrounding equipment, and this temperature must also be measured so that the design of the joint and control systems can counter this as far as possible. If wear and tear is too much, it might be an alternative to use the magnetic clutches described in Section 3.3

6.2 Further work on the simulator

The simulator does not fulfill the specifications stated in Section 4.1 at the current level of development. To achieve these requirements, further work needs to be done. This section points out areas where this further work must be focused, and recommends possible approaches to the

problems.

To simulate the dynamics of the robot, it is possible to use either the Newton-Euler or Lagrange-Euler method[8][23], which give equivalent dynamics but have very different implementation. The Newton-Euler method has the advantage that it is well suited for implementation on a general robot. However, it does not provide a closed form solution of the robot dynamics, as the Lagrange-Euler method does. When simulating such a complex robot as the climbing robot suggested in [28], it might save valuable computational time to have a closed form solution. This would require the closed form solution to be calculated off line, i.e. before the simulation loop, as part of the creation of the robot representation. The difficulty of implementing a general way to derive the dynamics in a closed form is the main drawback of the Lagrange-Euler method. Since the simulator will simulate a robot with variable number of degrees of freedom, it is not possible (or very inconvenient) to do this in a non-general way, or indeed by hand. Using a symbolic math package in MATLAB[26] might be helpful in meeting this challenge.

Once the dynamics of the robot has been simulated, the simulator can be modified to provide the user with joint torques. The log can be modified to include the joint torques, as well as joint references, control signals, robot position and other data that the user finds useful. It might also be beneficial to add torque limits to the `limits` field of the `element` struct (the struct describing each element in the chain representation of the robot).

To simulate the immediate environment of the robot, it is necessary with a model of this environment. For simplicity, this model might well be static, so that computational time is not spent calculating the dynamics of the environment. Further, to implement interaction between the robot and the environment it is necessary with some sort of collision detection. This will not only signal when the robot's end effectors are grasping for instance the ladder, but will also prevent the robot to move the arms inside itself, which is possible in the current simulator. The collision detection can be implemented as a feature of VRML, or it might be done in MATLAB. A compromise must likely be made between the quality of the collision detection and the computational requirements of the simulation.

It is necessary to fit the simulated robot with a gripper, as it is the grippers that will interact most with the immediate environment. The gripper can be an element of a special type, or it can be an extension of the current end effector element. It is also possible to implement the gripper as a series of joints with kinematic relations that gives them a gripping function. How the gripper of the climbing robot in [28] will look has not been decided, but it might be advantageous to design a general gripper for the simulator, much in the same way a general joint and body has been used in the current simulator. It might also be beneficial to implement the gripper as a general tool, so that it is possible at a later stage to exchange it with another kind of end effector. This could for instance be inspection equipment, a welding torch or other tools the robot needs to bring to the working area.

To implement robot movement, it is argued in Section 5.2 that the end effectors that holds on to the environment must change their type to base (effectively inverting the arm by making the first element the last and vice versa). The body can be represented as an end effector for the arms holding on, while at the same time serve as a link between the arms holding on and the arms that are loose. The body can also be represented as a non-moving joint. Either way, this

functionality must be implemented in the simulator.

To invert an arm, it is possible for each element to have a field not just for the Denavit-Hartenberg parameters describing the kinematic relationship to the next element in the arm, but also for the previous one. This would make it significantly easier perform the inversion task as it would only be just a matter changing which set of Denavit-Hartenberg parameters to use. Another method is to create a function that uses the Denavit-Hartenberg parameters from one element to the next to create the Denavit-Hartenberg parameters the other way, and then inserts them into the `denavit` field (where the parameters for the kinematic relationship to the next element is stored). The first method saves computational time, while the second saves storage space. Since computational time is likely to be more critical than storage space, it is recommended that the first method is used.

To make the body serve both as an end effector and as a link between the arms holding on and the arms that are loose it is possible to make the body a fourth type of element that is both end effector and joint. It is also possible to make the body consist of two elements in the same place, one that is an imaginary joint and one that is an end effector. Lastly, it is possible to have the body serve as a normal joint and not an end effector at all. This way, the movement of the body is not directly controlled, but rather the movement of the loose grippers. The last option might be both more intuitive and simple.

When the mobility of the robot is implemented, it is likely that part of the robot becomes a closed chain. This will for instance happen if the lower arms of the robot holds on to the environment and must cooperate to lift the body. In this case, the `dependents` field of the `element` struct might become useful. Finding a way for the software to automatically create the constraints put on the joints in this field is a big challenge.

There is much further work that could be done that does not directly help fulfilling the specifications of the simulator, but which will improve it. The following is a list of recommendations to improvements that could be made:

- As described in [5] it is possible to generate a large number of test cases to test different designs. This can for instance be done by having a script generate a large number of text files with joint references, and let the high level control module read from these. Reading from pre-generated text files is much less cumbersome than controlling each joint directly, as is done in the current simulator. Further, this approach allows the complex inverse kinematics problem involved in controlling the end effectors to be solved off line.
- It should be possible to save a designed robot to avoid having to enter robot parameters each time the simulator is run. The parameters can for instance be stored in a text file, or the MATLAB function `save` can be used[26].
- In the high level control it is also possible to train adaptive algorithms, an advantage of simulations mentioned in [5].
- As there can be advantageous to have a library of several different high- and low-level control modules, it might be convenient to have a dialog window where the user selects which high and low level control she or he wants to use. This dialog window can in turn open windows for specification of parameters etc. for the selected algorithms.

- The simulator can be modified to simulate other kinds of robots, as mentioned in Section 5.2. This would require at least the high-level control algorithm to be modified. Since both the files `simulateDynamics` and `updateKinematics` currently is dependent on a chain structure as described in Section 4.3.4, these files must also be updated for a simulation of other kinds of robots to work.
- By adding actuators and sensors to each element, it is possible to simulate different models of actuator and sensor dynamics and error dynamics[11]. An example of an actuator that could be added is the clutch used in the joint mechanism developed in Section 3. Actuators might be the clutch used in Section 3, and sensors might be potentiometers reading the joint value or cameras providing visual feedback. The implementation can for instance be done by letting the fields for actuators and sensors hold transfer functions for the dynamic responses of the actuators or sensors. Such dynamics will have to be included in the overall dynamic simulation of the robot.
- When implementing computational heavy work such as dynamic simulations or collision detection, it might be an advantage to re-assess MATLAB as the programming language. It can be beneficial to use more low-level languages such as C to do the complex dynamic simulations, as C might be faster. The MATLAB program could then call on the low-level program from the respective functions. The drawback of this choice is that the extensive mathematical library of MATLAB becomes unavailable.

7 Conclusion

Both a joint prototype and a simulator has been developed in this report. These are intended to be used in the further design of the climbing robot is described in [28]. Both has also been developed to be general enough to be used on other kinds of robots as well. This section contains the main conclusions drawn from the development work.

7.1 Joint

Part of this report has been concerned with the design and development of a prototype for a new joint concept. This part covered the general issues that needs to be fdjfkø in the design of a joint in Section 3.1, and in Section 3.2 challenges of the particular design proposed in this report are pointed out, namely:

- The clutching mechanism
- Transmission of power
- Braking
- Weight
- Modeling, simulation and control

The clutching mechanism were explored more deeply in Section 3.3, where the SO15 clutch produced by Inertia Dynamics[10] where chosen as the clutch most suitable for the joint. A joint prototype was then developed, and sections 3.4 - 3.7 covered the steps in the design process by showing 4 design proposals made in Autodesk Media & Entertainment's 3ds Max[1]. The fourth and final design proposal was assembled into a joint prototype which was qualitatively tested in Section 3.8. These tests were for the most part qualitative in nature and were evaluated in Section 5.1.

The prototype was mainly intended to demonstrate the concept of the proposed design. The tests thus concerned issues as how it rotated the joint, how the rotation of the main shaft of the joint was transferred past the joint even when the joint was rotated and how resistant the joint was to external forces trying to rotate the output.

It was demonstrated that the motor was able to rotate the main shaft, even when the joint was at an angle. This was because of a universal joint, or cardan, that was used to transfer the rotation past joint rotation axis. A drawback with the universal joint was that the loss of effect over it increased with increasing joint rotation.

The prototype used two clutches, one for each direction of rotation. It was demonstrated that each of the clutches was able to rotate the joint.

The braking mechanism that should enable the prototype to resist rotation due to external forces was implemented using worm gears. The worm gears was not strong enough to stop such unintended movement, and it was concluded that the braking mechanism must be improved.

A potentiometer was calibrated to measure the angle of the joint rotation. This was used to find the maximum and minimum angles of rotation, which was determined as the point where the

motor was no longer able to turn the joint. To avoid reaching these values, and thus stopping the motor, it is recommended to use a limit of $\pm 45^\circ$ for the joint rotation in future control systems.

There were some hardware problems with the joint prototype. Two of the gears of the assembled joint was slightly too large, and was thus quite heavy to rotate. In the beginning, this often blocked the motor, but as the gears were oiled and became worn down it became less of a problem. Further, a cogged transmission belt that was supposed to be used to transfer power from the main shaft of the joint to the output was too small. There was no time to replace it with another, larger transmission belt, so o-rings were used as a temporary solution. These slipped much more easily than a transmission belt would, and severely limited the output power of the joint. Lastly, one of the clutches was mounted slightly too tight, making the motor rotate slowly even when the clutch was not powered. This problem was gradually reduced as the clutch became more worn in.

Overall, the joint prototype demonstrated the principle, despite the hardware problems. Several areas where improvement can be done were revealed by the prototype, and these were discussed in Section 6.1. In addition to solving the hardware problems mentioned above, a major area of improvement is weight reduction.

7.2 Simulator

In addition to a new joint design, this report has also covered the development of a robot simulator. Specifications were given for this simulator based on the assignment text and literature, and these were listed in Section 4.1. The theoretical background necessary for developing the simulator was then given in sections 4.2 and 4.3. Section 4.2 explained the Denavit-Hartenberg convention [3][8][23], which were used to model the kinematics of the robot. Section 4.3 explained how the robot was represented in the simulator, using a chain representation for the mathematical representation [21] and VRML [26] for the graphical representation.

A description of the simulator as it is at the current stage of was given in Section 4.4. The structure of the software was presented in Section 4.5, where an overview and brief explanation of each file that has been used were given. The simulator described was evaluated in Section 5.2, and it was clear that the specifications of the simulator was not all fulfilled. Suggestions were given in Section 6.2 on how to fulfill these specifications in further work, and how to generally improve the simulator.

The chain representation used by the simulator is well suited for implementation in an object oriented programming language. When it was implemented in MATLAB, a heavy use of structs ensured that the programming environment was 'object oriented-like', so that the implementation worked well even in a non-object-oriented language.

The graphical representation of the robot used the Virtual Reality Modeling Language, or VRML, and was kept simple to give the user a quick overview of the simulated robot design. The simulator also provides dialog windows where the user can enter design parameters regarding the number of joints in each arm, the dimensions of the body and joints, position and velocity limits on the joints and kinematic parameters of the joints. When the simulator is running, the current high-level control provides the user with a dialog window where she or he can enter joint velocities.

The simulator saves all joint values of the last N timesteps in a log. N is a parameter that can be set by the user.

The simulator fails to fulfill several specifications at the current level of design. As discussed in Section 5.2, the unfulfilled specifications concern:

- The simulation of dynamics, which was postponed due the heavy requirements on computational time that such simulations would entail.
- The mobility of the robot, which was postponed due to complexities around inverse kinematics and collision detection.
- Simulation of the immediate environment of the robot, which was postponed due to time considerations. There was also no need to simulate this when the mobility of the robot was not simulated.

It is quite possible to modify the simulator to fulfill these specifications as well, as was suggested in Section 6.2. This section also suggested modifications to the simulator which would improve the overall performance without necessarily fulfill any specificationa.

It is hoped that the simulator will be developed to be a useful tool in the further design of the climbing robot developed in [28]. If completed, the simulator can be useful both to test different physical design, as well as in designing and testing control software. The simulator can also be useful as a demonstrator of the climbing robot, for early presentations for the industry.

References

- [1] Autodesk - Media & Entertainment Solutions. Autodesk - Media & Entertainment Solutions. <http://www.autodesk.com/me>, 2007.
- [2] Sverre Brovoll. Design og implementasjon av slangerobot aktuert med DC-motorer. Master's thesis, Norwegian university of science and technology, 2006.
- [3] J.J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, 3 edition, 2005.
- [4] Dynacorp. Dynacorp. <http://www.idicb.com/dynacorp/index.html>, 2007.
- [5] A. Ettl, P. Büchler, and H. Bleuler. A Simulation Environment for Robot Motion Planning. *Robot Motion and Control, 2005. RoMoCo'05. Proceedings of the Fifth International Workshop on*, pages 277–282, 2005.
- [6] Fairchild Semiconductor. Fairchild Semiconductor. <http://www.fairchildsemi.com/>, 2007.
- [7] Faulhaber Group. Faulhaber Group.
- [8] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee. *Robotics: control, sensing, vision, and intelligence*. McGraw-Hill, 1 edition, 1987.
- [9] Terje Haugen and Hans Jørgen Berntsen. NTNU Engineering Cybernetics Workshop. Oral communication, 2007.
- [10] Inertia Dynamics. Inertia Dynamics. <http://www.idicb.com/>, 2007.
- [11] JJ Kuffner Jr, S. Kagami, M. Inaba, and H. Inoue. Graphical simulation and high-level control of humanoid robots. *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, 3, 2000.
- [12] Pål Liljebäck. Sintef. Oral communication, 2007.
- [13] Sebastian Lohmeier, Thomas Buschmann, Heinz Ulbrich, and Friedrich Pfeiffer. Modular Joint Design for Performance Enhanced Humanoid Robot LOLA. *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pages 88–93, 2006.
- [14] Gears Manufacturers. Harmonic drives. <http://www.gears-manufacturers.com/harmonic-drives.html>, 2007.
- [15] Mekanex. Mekanex. <http://www.mekanex.se/>, 2007.
- [16] T. Morita and S. Sugano. Design and development of a new robot joint using a mechanical impedance adjuster. *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, 3:2469–2474, 1995.
- [17] National Instruments. LabVIEW. <http://www.ni.com/labview/>, 2007.

- [18] National Instruments. National Instruments. <http://www.ni.com/>, 2007.
- [19] Jia Qing-xuan, Sun Han-xu, Song Jingzhou, Cheng Tao, and Zheng Ping. Development of Modular Robot Joint. *Industrial Informatics, 2006 IEEE International Conference on*, pages 827–832, 2006.
- [20] Rimtec Corporation. Rimtec Corporation. <http://www.rimteccorporation.com/>, 2007.
- [21] O. Rogalla, K. Pohl, and R. Dillmann. A general approach for modeling robots. *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, 3, 2000.
- [22] Sanyo Denki. Sanyo Denki America, Inc. <http://www.sanyo-denki.com/>, 2007.
- [23] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 3 edition, 2000.
- [24] E. Shammas, A. Wolf, HB Brown Jr, and H. Choset. New joint design for three-dimensional hyper redundant robots. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 4, 2003.
- [25] Standa. Standa. <http://www.standa.lt/>, 2007.
- [26] The MathWorks. The MathWorks - MATLAB and Simulink for Technical Computing. <http://www.mathworks.com/>, 2007.
- [27] Vishay. Vishay. <http://www.vishay.com/>, 2007.
- [28] Martin Wiig. A climbing robot for inspection of infrastructures. Norwegian university of science and technology, 2006.

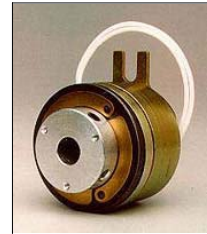
A The SO series of electromagnetic friction clutches



[Products](#)
[Inertia News](#)
[About ID](#)
[Conversion Charts](#)
[Request Information](#)
[Design Assistance](#)
[Home](#)

Shaft Mounted Clutch Couplings – Type SO

SO series power-on clutch couplings are used to couple two in-line shafts. The armature hub assembly is mounted to the load shaft and the rotor assembly is mounted on the input shaft. The field assembly is mounted on the input shaft and retained by a loose-fitting pin or bracket through the anti-rotation tab.



[Click here for ordering information](#) (PDF, 56k). For larger sizes, please see [Dynacorp's Clutches and Clutch Couplings](#).

[RETURN TO ELECTROMAGNETIC CLUTCHES & BRAKES](#)

Type SO: [Mechanical Specs](#) | [Electrical Specs](#) | [Dimensions](#) | [Notes](#) | [Ordering](#)

CUSTOMER SHALL MAINTAIN

A loose-fitting pin through the anti-rotation tab to prevent pre-loading the bearings.
Concentricity between the shafts within .005" T.I.R. initial air gap setting of .005" - .020".

Mechanical				
MODEL NO.	STATIC TORQUE LB. - IN.	INERTIA LB. - IN. ²		WEIGHT OZ.
		ROTOR	ARM & HUB	
SO08	2.5	.002	.0011	2
SO11	6	.0058	.0024	3.2
SO15	10	.060	.026	3.8
SO17	15	.061	.031	11
SO19	25	.082	.042	12
SO22	50	.215	.070	20
SO26	80	.362	.320	28
SO30	125	.610	.561	45
SO42	250	2.50	2.30	80

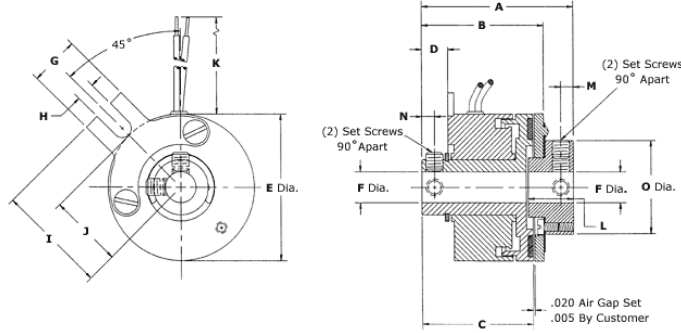
[TOP OF PAGE](#)

Electrical						
MODEL NO.	90 V.D.C.		24 V.D.C.		12 V.D.C.	
	AMPS	OHMS	AMPS	OHMS	AMPS	OHMS
SO08	.046	1977	.117	205	.246	48.8
SO11	.047	1930	.198	121	.447	26.8
SO15	.042	2150	.183	132	.380	31.6
SO17	.066	1369	.289	83	.561	21.4
SO19	.074	1213	.294	81.6	.574	20.9
SO22	.079	1140	.322	74.6	.628	19.1
SO26	.092	980	.374	64.2	.760	15.8
SO30	.091	988	.378	65.3	.729	16.5
SO42	.124	722	.468	51.2	.934	12.84

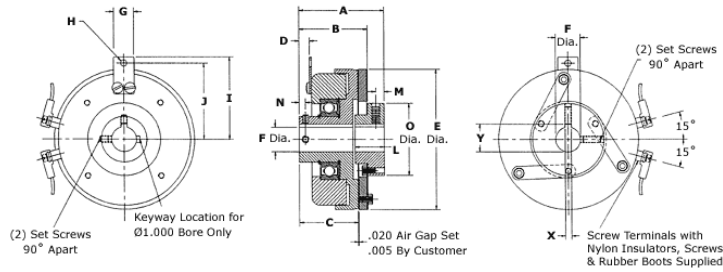
Lead wire is U.L. recognized style 1213, 1015 or 1430, 22 gage.
Insulation is .050" O.D. on 08, 11, 15 units; .064" or .095" O.D. on all other units.

[TOP OF PAGE](#)

Model SO08 through SO26



Model SO30 through SO42



MODEL NO.	A MAX.	B NOM.	C NOM.	D NOM.	E MAX.	F NOM.	G MAX.	H MIN.	I NOM.	J NOM.	K ± .500	ROTOR KEYWAYS		L MAX.	M NOM.	N NOM.	O MAX.	
												BORE	NOMINAL KEYWAY					
																		X
SO08	1.059	.875	.763	.191	.903	1/8 3/16 1/4	.305	.094	.625	.445	12.00	N.A.	SET SCREWS ONLY	.237	.070	.080	.500	
SO11	1.168	.933	.777	.147	1.160	3/16 1/4 5/16	.380	.122	.875	.585	12.00	N.A.	SET SCREWS ONLY	.307	.093	.087	.687	
SO15	1.575	1.255	1.075	.275	1.500	1/4 5/16 3/8	.520	.180	1.120	.750	12.00	N.A.	SET SCREWS ONLY	.475	.125	.125	.965	
SO17	1.605	1.311	1.060	.270	1.780	1/4 5/16 3/8	.505	.184	1.325	.975	12.00	1/4 5/16 3/8	.0625 -.0655 .0625 -.0655 .094 -.097	.285 - .290 .347 - .352 .417 - .427	.460	.115	.125	1.190
SO19	1.609	1.314	1.060	.270	2.000	5/16 3/8 1/2	.505	.184	1.325	.975	12.00	5/16 3/8 1/2	.0625 -.0655 .094 -.097 .125 -.128	.347 - .352 .417 - .427 .560 -.567	.455	.115	.125	1.190

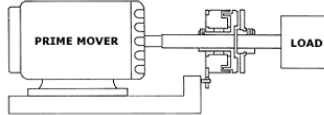
A THE SO SERIES OF ELECTROMAGNETIC FRICTION CLUTCHES

SO22	1.989	1.578	1.273	.281	2.260	3/8 1/2	.442	.170	1.515	1.160	18.00	3/8 1/2	.094 - .097 .125 - .128	.417 - .427 .560 - .567	.510	.115	.117	1.005
SO26	2.115	1.754	1.444	.277	2.645	3/8 1/2 5/8	.510	.190	1.750	1.465	18.00	3/8 1/2 5/8	.094 - .097 .125 - .128 .1885 - .1905	.417 - .427 .560 - .567 .709 - .716	.610	.150	.187	1.440
SO30	2.151	1.815	1.403	.265	3.268	1/2 5/8 3/4	.442	.170	2.050	1.695	SCREW TER- MINALS	1/2 5/8 3/4	.125 - .128 .1885 - .1905 .1885 - .1905	.560 - .567 .709 - .716 .836 - .844	.680	.150	.135	1.825
SO42	2.570	2.050	1.625	.320	4.270	1/2 5/8 3/4 7/8 1	.645	.190	2.500	2.312	SCREW TER- MINALS	1/2 5/8 3/4 7/8 1	.125 - .128 .1885 - .1905 .1905 - .1885 - .1905 - .251 - .253	.560 - .567 .709 - .716 .836 - .844 .962 - .970 1.113 - 1.121	.890	.250	.187	2.195

[TOP OF PAGE](#)

Notes:

1. 30 and 42 units have a single ball bearing between the field and rotor.
2. 08 units have set screws 120° apart.
3. 08 and 19 units have retaining collar.



[TOP OF PAGE](#)

[RETURN TO ELECTROMAGNETIC CLUTCHES & BRAKES](#)



Inertia Dynamics, LLC
 100 Franklin Drive, P.O. Box 1277, Torrington, CT 06790
 Toll Free: 1-800-800-6445 | Phone: (860) 482-4444 | Fax: (860) 482-4378 | sales@idicb.com

[Products](#) | [Inertia News](#) | [About Inertia Dynamics](#) | [Conversion Charts](#) | [Site Directory](#) | [Home](#)
[Request More Information](#) | [Request Design Assistance](#)


[Electromagnetic Clutches & Brakes](#) | [Spring Applied Brakes](#) | [Wrap Spring Clutches & Brakes](#)
[Motor Brakes](#) | [Clutch/Brake Controls](#) | [Moment of Inertia Measurement Instruments](#)

[Dynacorp® Products](#)

B The DCB-2 wrap spring clutch

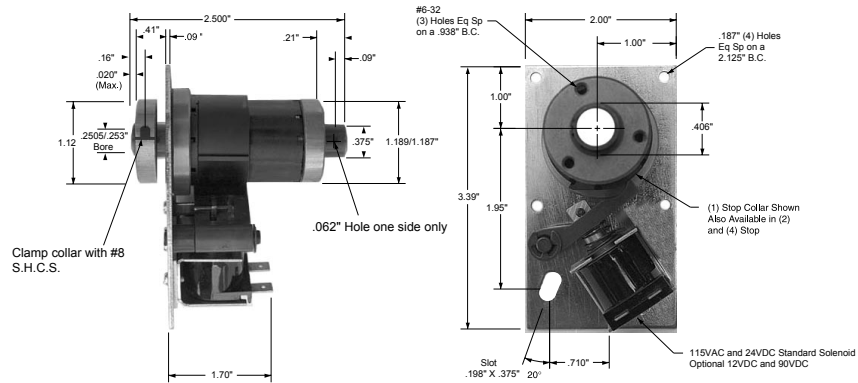
Wrap Spring Clutches

DCB-2 Dimensions

Product Function: Start-Stop 

Specifications

Maximum Speed:	1800 RPM
Static Torque:	25 in.-lb
Max. Anti-overrun Torque:	10 in.-lb.
Max Anti-back Torque:	10 in.-lb.
Positioning Accuracy:	$\pm 1/2^\circ$
Weight:	1.25 lbs.




Unit Shown is CW Rotation, CCW is Available

DCB-2 Part Numbers

Part Number		Description			
CW Rotation	CCW Rotation	Model	Bore	Voltage	Stops
D302-17-001	D302-27-001	DCB2	1/4"	24VDC	1
D302-17-002	D302-27-011	DCB2	1/4"	24VDC	2
D302-17-003	D302-27-003	DCB2	1/4"	24VDC	4
D302-17-007	D302-27-007	DCB2	1/4"	115VAC	1
D302-17-008	D302-27-008	DCB2	1/4"	115VAC	2
D302-17-009	D302-27-029	DCB2	1/4"	115VAC	4

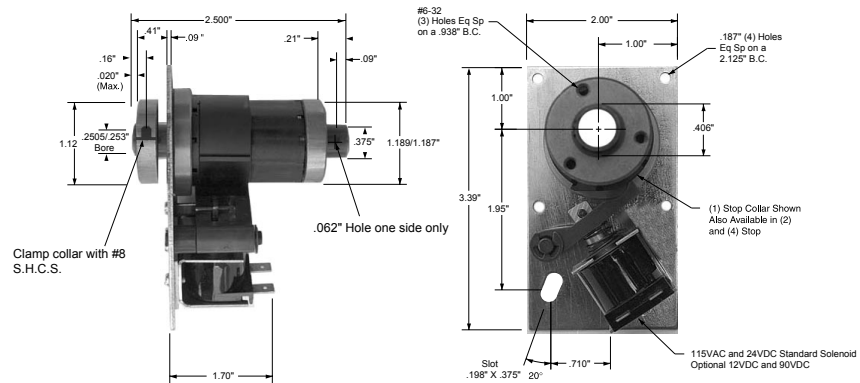
Wrap Spring Clutches

DCB-2 Dimensions

Product Function: Start-Stop 

Specifications

Maximum Speed:	1800 RPM
Static Torque:	25 in.-lb
Max. Anti-overrun Torque:	10 in.-lb.
Max Anti-back Torque:	10 in.-lb.
Positioning Accuracy:	± 1/2°
Weight:	1.25 lbs.



Unit Shown is CW Rotation, CCW is Available

DCB-2 Part Numbers

Part Number		Description			
CW Rotation	CCW Rotation	Model	Bore	Voltage	Stops
D302-17-001	D302-27-001	DCB2	1/4"	24VDC	1
D302-17-002	D302-27-011	DCB2	1/4"	24VDC	2
D302-17-003	D302-27-003	DCB2	1/4"	24VDC	4
D302-17-007	D302-27-007	DCB2	1/4"	115VAC	1
D302-17-008	D302-27-008	DCB2	1/4"	115VAC	2
D302-17-009	D302-27-029	DCB2	1/4"	115VAC	4

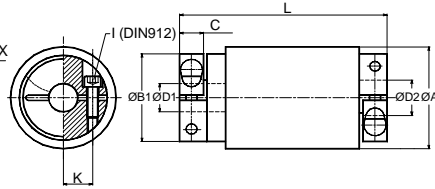
C The HKD series of magnetic clutches

HKD SERIES COLLET TYPE

PERMANENT- MAGNET HYSTERESIS CLUTCH



ORDER CODE: HKD 18-14-17-18-17-XX
 Type _____
 Size(Nm) _____
 Bore D1 _____
 Bore D2 _____
 Keyways, material, etc _____



DIMENSIONS: (mm)

Type HKD	ØA	ØB	C	ØD1 and ØD2 Min - Max H7	I DIN 912	K	L	M Magnet Length (mm)
HKD 2	31	25	8.2	3 - 10	M3	9	55	20
HKD 4	38	32	10	6 - 16	M4	11.5	58	20
HKD 10	46	40	10	6 - 19	M4	15.5	57	20
HKD 18	51	45	12	10 - 20	M5	17.5	78	30
HKD 30	56	47	15	10 - 20	M6	16	88	30
HKD 60	69	57	19.5	14 - 23	M8	20	107	40
HKD 150	84	68	21.5	20 - 28	M10	24	130	50

TECHNICAL DATA:

Type HKD	Moment of Inertia (10 ⁻³ kgm ²)		Overload Torque (Nm)	Mass (kg)		Torque to Tighten Clamps (Nm)	Max. Operating Speed (rpm)	Max. Power Dissipation (W)
	Inside	Outside		Inside	Outside			
HKD 2	0.005	0.018	0.1	0.07	0.12	2	10000	4.0
HKD 4	0.02	0.04	0.2	0.11	0.15	3	9000	5.0
HKD 10	0.04	0.07	0.4	0.16	0.18	3	8000	7.0
HKD 18	0.08	0.14	0.9	0.25	0.28	6	7000	12.0
HKD 30	0.11	0.20	1.2	0.27	0.34	12	6000	14.0
HKD 60	0.57	0.87	2.5	0.51	0.68	30	5000	20.0
HKD 150	1.30	1.80	5.0	1.6	1.7	50	4000	30.0

- 1) Samarium Cobalt (SmCo) magnets available for higher temperature applications, up to 300°C
- 2) Size 2-60 clamping hubs made of aluminum, size 150 hubs made of steel.
- 3) In the case of bores of less than C_{min}, transmission of nominal torque Mn of the clutch is no longer guaranteed. Versions with bores of less than C_{min} can be supplied.
- All hub bores are supplied in accordance with customer specifications to tolerance fit H7.
- 4) Keyways according to standard DIN 6885 or American on request. Clearance of keyway, standard JS 9
- 5) Larger radial misalignment possible
- 6) See below for calculation of HKD heat dissipation

MISALIGNMENT:		
Type	Lateral (mm)	Axial (mm)
HKD	0.2	Variable

Power Dissipation Calculation

Hysteresis type clutches and brakes slip if an overload occurs. The losses due to the slip rotation and torque are converted into heat. If the power to be dissipated exceeds the heat dissipation capabilities of the clutch, the clutch (brake) will superheat, damaging the magnets. The following formula should be used to insure that the maximum power loss of the selected clutch (brake) is sufficient for the desired operation mode

$$P_v = \frac{T \cdot n_s \cdot d}{9.55}$$

P_v = Maximum power loss (in W)

T = Applied torque (in Nm)

n_s = Slip rotation (in rpm)

d = Duty cycle (in %)

RIMTEC CORPORATION
 211 S. Lombard Road, Addison, IL 60101
 Toll Free: 877-4RIMTEC (877-474-6832)
 Tel: 630-628-0036, Fax: 630-628-0041
 E-mail: info@RimtecCorporation.com
 Web Site: http://www.RimtecCorporation.com

D Digital Appendix

The digital appendix is enclosed on a CD containing 4 folders:

- **3ds models** - contains the 3ds models of the four designs presented in sections 3.4-3.7.
- **LabVIEW diagram** - contains the LabVIEW file used in Section 3.8.
- **MATLAB files** - contains the MATLAB files used by the simulator of Section 4
- **Testing videos** - contains videos of the testing in Section 3.8. The file `Joint rotation.mov` shows how the joint rotated, the file `Main shaft rotation 1.mov` shows the rotation of the main shaft, the file `Main shaft rotation 2.mov` shows how the universal joint transfers this rotation past a rotated joint and the file `Orings slip.mov` show how the o-rings slipped over the cogwheels.