University of Twente

EEMCS / Electrical Engineering
Control Engineering

Analysis and development of a 2D walking machine

Niels Beekman

M.Sc. Thesis

Supervisors
prof.dr.ir. J. van Amerongen
dr.ir. S. Stramigioli
ir. V. Duindam, dr. ir. H. van der Kooij

december 2004

Report nr. 033CE2004

Control Engineering
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

For Vincent Duindam's promotional research a biped walker has to be constructed that can be used to test certain control and modeling techniques for 3D walking. With this thesis a small start has been made to achieve this. Because 3D walking is very complex a choice was made to first analyse and design a 2D walker. One of the main aspects is that the design should be energy efficient and should respect the natural dynamics of the system.

In this report 2D walking is analyzed by modeling and simulation. A controller was designed that respects the natural dynamics of the system by keeping it in a stable oscillation. Modeling and simulation also gave some hints on practical dimensions of the walker and requirements on the subsystems.

With these results a mechanical and electrical design was made. At the time of writing a start has been made in building this design.

Preface

In September 1999 I started a new live by studying Electrical Engineering at the University of Twente. This thesis represents the final stage in obtaining a Master of Science-level degree, which in my case is still the 'ir.'-title.

I would like to thank Stefano and Vincent for their support and guidance not only during this thesis, but also during my IOO-assignment and my internship. I am very happy that the two of you supervised my work with so much enthusiasm, and with useful hints and ideas when needed.

I would also like to thank Edwin and Gijs. During the finalization of my thesis they were very cooperative and will continue my work and build the robot. The help of Edwin made it possible that in a relatively short time a design has been made that will be constructed. In this context I would also like to thank prof. Soemers for giving some very useful mechanical design tips.

But most of all I would like to thank my parents since they made it possible for me to focus on my education without having to worry much about financial issues.

Enschede, November 19, 2004

Niels Beekman

Contents

Chapter 1

Introduction

Figure 1.1: John Cleese in the Ministry of silly walks.

For long time walking has been studied by many researchers because of its complex nature. Walking is being studied to be able to help paralyzed people, or to create robots that can cover rough terrain. Even in cinema walking has been studied (figure 1.1). Different approaches can be found in literature; one approach is to build a humanoid like the Honda P2 ([3]). From an engineering point of view it is very advanced, it has many actuators to create human like motion and it also uses some very interesting control techniques to control the balance of the robot. However, it uses lots of energy while walking, up to 2 [KW] where a human being of the same size would only consume 100 [W].

For this reason people have been interested in walking in a way that is more energy efficient. The complete opposite to the Honda solution is passive dynamic walking.

Passive dynamic walking was first researched by Tad McGeer [5]. He showed that it is possible for a machine to perform a stable gait down a slope without actuation or control. He constructed a fully passive model with knees that create the required ground clearance. His research opened new possibilities for very simple walkers that can walk on a level ground with only minimal actuation.

The simplest example of a 2D passive dynamic biped can be seen in figure 1.2. It has two pairs of legs to constraint sideways motion. When given the proper initial speed the walker can walk a few steps down a slope. Because it has no knees it needs the steps to create ground clearance. While a very simple model it shows the power of passive dynamic walking: it is intrinsically stable and very energy efficient.

In this project we try to extend this passive walker with actuator(s) to create an active 2D walker that can walk on a level floor. By restricting us to a 2D walker we do not have to pay attention to sideways motion (e.g. roll and yaw). Of course the principles of dynamic walking should be regarded; the active part of the walker should respect the dynamics of the walker in stead of suppressing them.

This thesis consists of two parts: a theoretical study of 2D walking (Part 1) and a design proposal of a 2D walker (Part 2).

The theoretical part is organized as follows:

- Chapter 2 presents the simulation model that is used to study a 2D walker with knees.

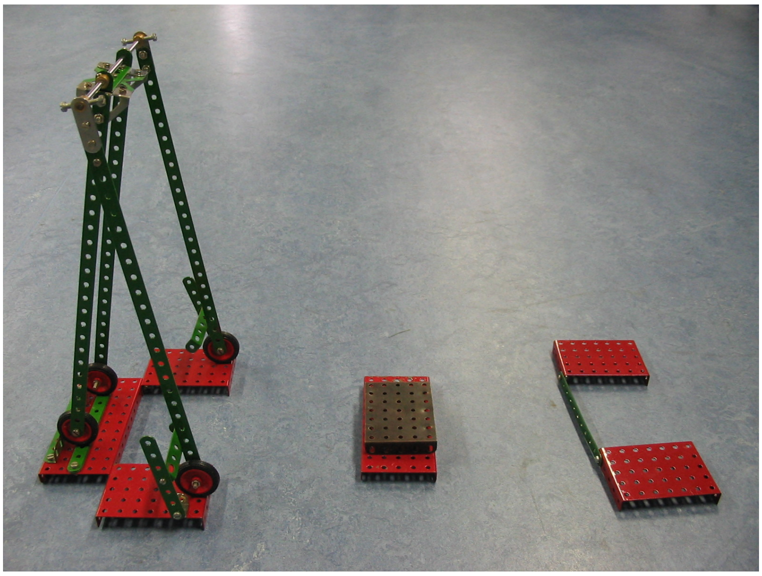


Figure 1.2: Simplest passive dynamic walker by Gijs van Oort.

- Chapter 3 presents the control techniques that are studied for the walker.
- Chapter 4 presents a method of increasing the efficiency of the walker.
- Chapter 5 presents the requirements on the mechanical design that are found from the simulations.

The practical part is organized as follows:

- Chapter 6 discusses various design choices that have to be made.
- Chapter 7 proposes the design.

After that, we present our conclusions and recommendations in chapter 8.

Part I

Analysis of 2D passive dynamic walking

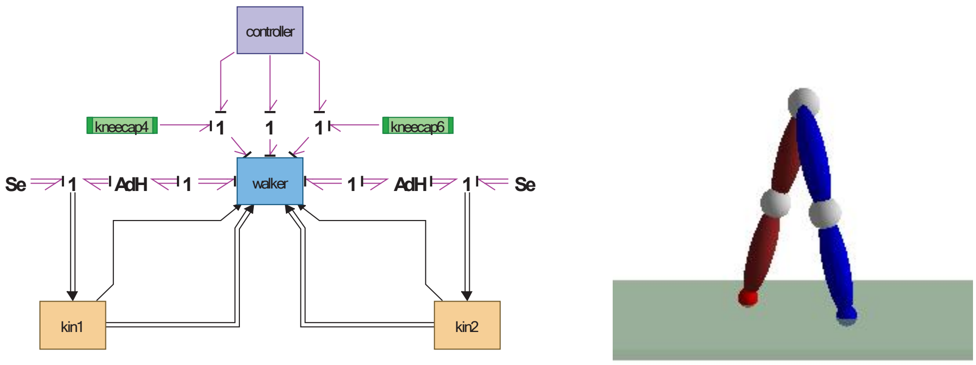
Chapter 2

2D simulation model with knees

All simulations done in this thesis are based on the 2D model presented in this chapter. It is a 20Sim model [10] made by Vincent Duindam. In this section we will present this model and discuss the theory of it.

2.1 The basic model

The basic model can be seen in figure 2.1(a). The walker it models looks like 2.1(b).



(a) The basic 20sim model

(b) Graphical view of the model

Figure 2.1: The basic walking model.

At the top we see the controller. It is a simple PID controller of which the setpoint switches from -0.4 [rad] to 0.4 [rad] once every step is made.

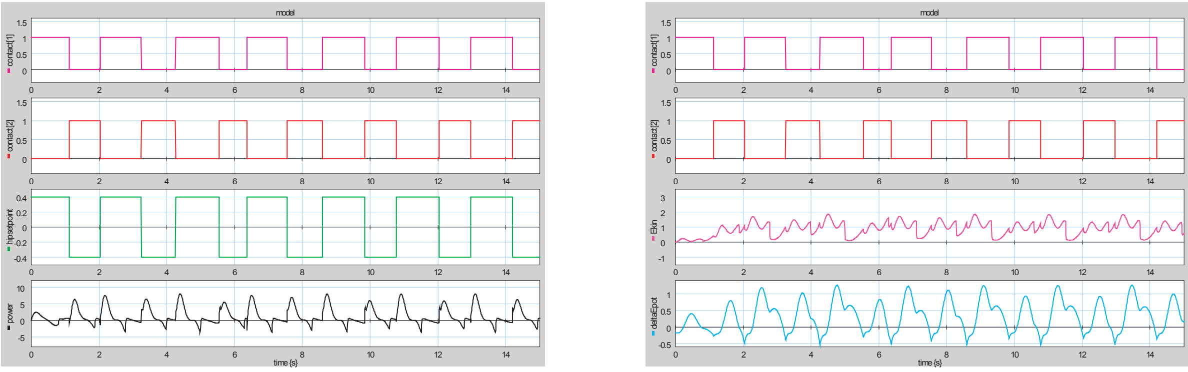
Each knee has an endcap that is modeled as a (very stiff) spring and a damper. It is critically damped. The end cap is only active when the leg is fully stretched. The angle of the knee is at 0 [rad] in that case.

The block labeled 'walker' models the kinematics of the walker. It is generated with the body editor of 20sim. The walker of the basic model has links of unity length and unity mass.

The two 'kin' blocks are used to calculate the contact force of a foot on the ground. The foot to ground contact is assumed to be inelastic. The force that is calculated here is applied on the walker model.

Simple walking The robot can already walk when the simple walking algorithm of this controller is used. See figure 2.2(a) for a simulation run of this model. We see that the hip setpoint is switched from

-0.4 [rad] to 0.4 [rad]. Looking at the power plot in the figure we see that the negative part is quite big. This means that the controller has to absorb energy. This energy comes from the controller overshoot, the leg has to much kinetic energy and this has to be absorbed by the controller. In reality this will mean that this energy is converted into heat. In chapter 4 we will look at this in more depth and a solution will be presented to use this energy effectively.



(a) Simple walking results

(b) Kinetic and potential energy

Figure 2.2: The basic walking model.

2.2 An energetic view of the walker model

In the walker there is energy storage as kinetic energy (while moving) and potential energy (because of gravity). These energies are exchanged between each other, potential energy is at its highest when the swing leg passes the stance leg whereas the kinetic energy is minimal at that point (and maximal right after foot impact). A good controller will try to use this fact and will control the walker so that this energy exchange is fluid.

2.2.1 Kinetic and potential energy equations

The following equations can be used to calculate the kinetic and the potential energy of the walker.

Kinetic energy For a point mass of mass m the kinetic energy is given as:

$$E_{kin} = \frac{1}{2m}p \tag{2.1}$$

with m the mass and p the momentum of the mass.

This can be generalized for our walker (and other rigid body structures) to:

$$E_{kin} = \frac{1}{2}P^T M^{-1} P \tag{2.2}$$

where P is the momentum of the walker and M^{-1} is the inverse of the mass matrix.

Potential energy The potential energy of the walker is also easy to compute when all the masses of the bodies (which are in fact the links of the walker) are known and the heights of their centers of mass with respect to ground. See 2.3.

$$E_{pot} = \sum_i m_i g h_i \tag{2.3}$$

where m_i is the mass of body i, g is the constant of gravity and h_i is the distance of the center of mass of body i to the ground.

2.2.2 Kinetic and potential energy simulation plots

A plot of the kinetic and the potential energy in the walker can be found in figure 2.2(b). They are calculated using equations 2.2 and 2.3.

The two plots in the top of the figure show when contact with the ground is made for each foot. The lower two plots are the kinetic energy (third plot) and the potential energy (fourth plot).

Here we can see what we already expected: kinetic energy is maximal after impact. It is not maximal directly after impact because the foot to ground impact results in some kinetic energy being lost. It is minimal when the swing leg passes the stance leg (but not zero because there is some kinetic energy storage in the swing leg). We see that the potential energy is maximal at this point.

It is striking that it is not at the exact same point as when kinetic energy is minimal, as would be expected.

The reason for this is that the big drop in kinetic energy does not come from the energy exchange between potential and kinetic energy, but it is due to dissipation in the knees when the end cap is active.

So in this figure we already see two of the main causes of energy dissipation in a walker. They will be discussed in more depth in chapter 4.

Chapter 3

Walking control algorithms

Control techniques for walking biped robots can roughly be divided into two groups. On the one hand there are control techniques that are used when both legs of the robot are in contact with the ground. We will refer to this as the double support phase. On the other hand there are control techniques that control the walker when there is only one leg in contact with the ground. This is the single support phase. For this project only the single support phase is relevant because we focus on walking only but it is useful to investigate the double support phase also for possible future use.

In here two control techniques will be discussed, one for each phase. For the double support phase we will use virtual model control, which was first presented in [8]. This technique and our results applying this will be presented in section 3.1. For the single support phase we will use a method called 'foot placement' [9].In section 3.2 we will describe this method and extend it. It will be shown that foot placement is a powerful method for controlling a walking robot while still respecting the natural dynamics of the walker.

3.1 Double support: Virtual Model Control

Virtual model control is a control method that makes the total system act as if there are external components attached to it. This is why it is called *virtual* model control, because the external components are not real. To do this it has to calculate the joint torques that are required to let the system behave as if the virtual components were present. It can be used to stabilize a walking robot. One approach is to construct a virtual granny walker, named after the structure that stabilizes the gait of millions of elderly people.

In here we will discuss how virtual model control can be implemented for a biped walking robot to stabilize the dual support phase. It requires full actuation of the joints to function.

3.1.1 Walker simulation model

The simulation model used looks like figure 3.1. Although it looks like a multiple branch structure, it is just a serial chain. One of the joints in the figure that is connected to the floor is not a real joint, it is only a constraint. [1] The constraint acts as the end effector of the serial chain. This approach makes it easy to simulate the walker in the case that one leg is fixed to the ground (stance) and the other swings freely. The fixed leg is connected to the ground with a joint. The constraint on the end effector is only active when the swing leg hits the ground. This simplification makes it possible to simulate a multiple branch structure as a serial chain.

Two equations are very important to describe the kinematics of a robot:

$$T_n^{0,0} = J_n(\vec{q})\dot{\vec{q}} \tag{3.1}$$

$$\vec{\tau} = J_n^T(\vec{q})W_n^0 \tag{3.2}$$

[1] Of course a joint is nothing else than a set of constraints. For modeling purposes here we make the distinction between a joint as an element from the body editor and a constraint as an line of code in the model.

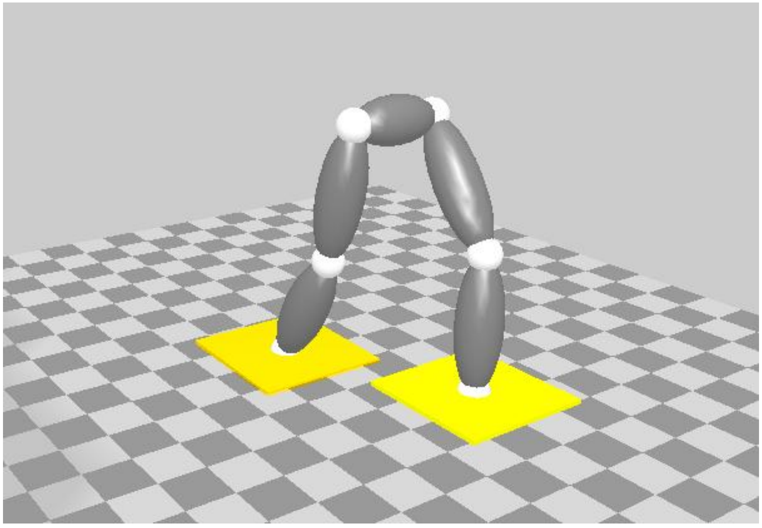


Figure 3.1: Two legged walker simulation model

where $T_n^{0,0}$ is the twist of body n with respect to body 0 expressed in the frame of body 0, W_n^0 is the wrench (generalized force) applied on n and expressed in the frame of body 0, $J_n(\vec{q})$ is the manipulator Jacobian of body n, $\dot{\vec{q}}$ are the joint-velocities and $\vec{\tau}$ are the torques on the joints. A good introduction into screw theory can be found in [12].

So, equation 3.1 describes the twist that results from actuation of the joints, and 3.2 describes what joint torques are necessary to apply a certain wrench to the body.

The body editor generates 5 Jacobians for this model: one for each joint/body. The first will have only one column because motion of that body is only influenced by the first joint. The last Jacobian will have all the columns filled because every joint influences the motion of the last body. The floor is the reference body (body 0).

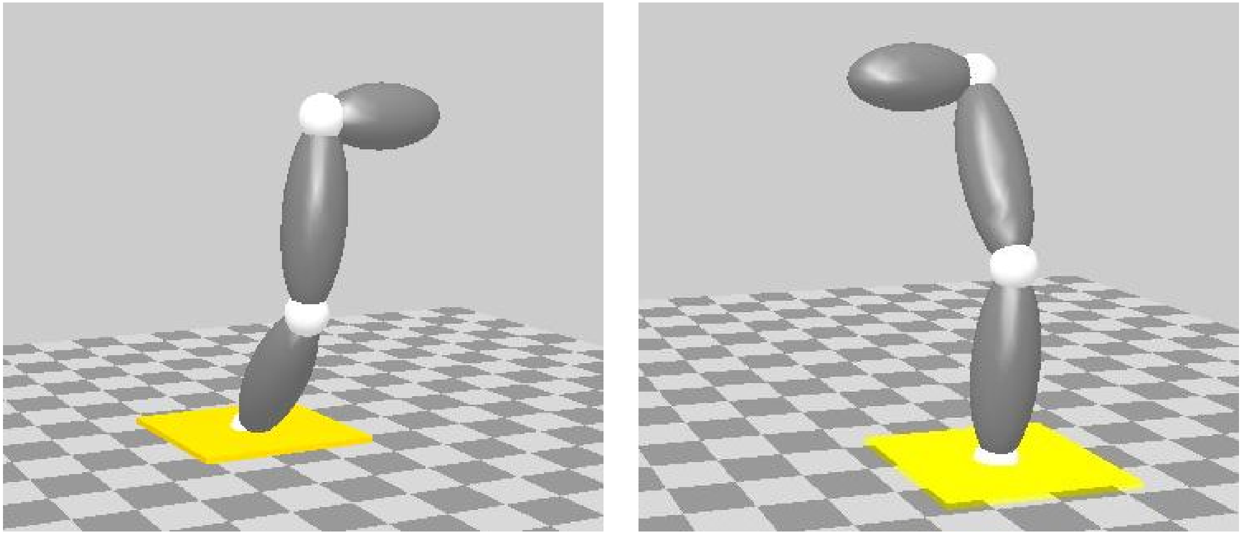
$$\begin{aligned}
J_1 &= \begin{bmatrix} T_1^{0,0} & 0 & 0 & 0 & 0 \end{bmatrix} \vec{q} \\
J_2 &= \begin{bmatrix} T_1^{0,0} & T_2^{0,1} & 0 & 0 & 0 \end{bmatrix} \vec{q} \\
&\vdots \\
J_5 &= \begin{bmatrix} T_1^{0,0} & T_2^{0,1} & T_3^{0,2} & T_4^{0,3} & T_5^{0,4} \end{bmatrix} \vec{q}
\end{aligned} \tag{3.3}$$

The Jacobians are as in equation 3.3, where $T_c^{a,b}$ is the unit twist of body c with respect to body b expressed in the frame of body a. In other words, this twist describes a joint.

3.1.2 Virtual Model Control simulation difficulties

To simulate the kinematics of the biped walker we use the model of figure 3.1. For virtual model control a mapping is needed that maps the virtual wrench generated by the virtual model to joint torques. However, we cannot use the before mentioned model with the Jacobians of equation 3.3. Say for example a virtual model is attached to the hip. The hip corresponds with J_3. But J_3 only has the first three columns non-zero and so equation 3.2 will only compute three joint torques (of the 5 joint torques available). So we need a model that uses the fact that the biped walker consists of two parallel branches.

The solution is to split the walker model of figure 3.1 into two separate leg models as is done in figure 3.2. Both legs are identical, they both consist of three joints: an ankle joint, a knee joint and an hip



(a) Left leg (b) Right leg

Figure 3.2: Two seperate legs to calculate the joint torques required for VMC.

joint. Per leg you get a Jacobian of 3 columns giving 3 joint-torques when equation 3.2 is calculated. The trick now is to split the wrench generated by the virtual model into two wrenches: one for the left leg and one for the right leg. A logical choice would be an even load distribution, but this is arbitrary. You get two sets of joint torques that can then be used to stabilize the model of figure 3.1. An important issue is that the two separate leg models should have the same configuration as the serial chain model. The joint angles of the separate leg models will need to be derived from the serial chain model. This is of importance because otherwise the wrong joint torques will be applied to the serial chain model.

The next section will discuss the mathematics involved to calculate the required joint torques given the wrench applied by the virtual model.

3.1.3 Virtual Model Controller torque derivation

We assume that the virtual model wrench gets distributed between the two legs:

$$W^0 = W_L^0 + W_R^0 \tag{3.4}$$

where W^0 is the virtual model wrench, W_L^0 is the virtual model wrench applied to the left leg and W_R^0 the virtual model wrench applied to the right leg.

This can be written in matrix notation as:

$$W^0 = \begin{bmatrix} I & I \end{bmatrix} \begin{bmatrix} W_L^T \\ W_R^T \end{bmatrix} \tag{3.5}$$

From equation 3.2 we get the joint torques:

$$\vec{\tau_L} = J_L^T W_L^T \tag{3.6}$$
$$\vec{\tau_R} = J_R^T W_R^T \tag{3.7}$$

where J_L^T is the manipulator Jacobian of the left leg, J_R^T is the manipulator Jacobian of the right leg and $\vec{\tau_L}$, $\vec{\tau_R}$ are the torques to be applied to the serial chain model.

Equations 3.6 and 3.7 written in matrix notation gives:

$$\begin{bmatrix} \vec{\tau_L} \\ \vec{\tau_R} \end{bmatrix} = \begin{bmatrix} J_L^T & 0 \\ 0 & J_R^T \end{bmatrix} \begin{bmatrix} W_L^T \\ W_R^T \end{bmatrix} \tag{3.8}$$

We want to be able to express equation 3.8 as a function of W^0, because we get W^0 from our virtual model. So, we need to rewrite equation 3.5. This cannot be done just like that because that equation has 12 degrees of freedom (six for each wrench) and only 6 equations (the six joint torques). This would require 6 constraints. Since our walker will be a 2D walker we will limit ourselves to the case where there is no motion in the y-direction.

In the general 3D case the wrenches look like:

$$W^T = \begin{bmatrix} m_x \\ m_y \\ m_z \\ f_x \\ f_y \\ f_z \end{bmatrix}$$

The wrenches are much simpler in our 2D-case, there can only be a moment around the x-axis and force along the y and z axes. m_y, m_z and f_x are therefore all zero. We will use a star (*) to denote a 2D wrench from now on, so:

$$W^{*,T} = \begin{bmatrix} m_x \\ f_y \\ f_z \end{bmatrix}$$

To get an expression in $W^{*,T}$ we will rewrite equation 3.5 into:

$$\begin{bmatrix} W^{*,T} \\ 0 \end{bmatrix} = \begin{bmatrix} I & I \\ constraint & functions \end{bmatrix} \begin{bmatrix} W_L^{*,T} \\ W_R^{*,T} \end{bmatrix} \tag{3.9}$$

We now have 6 degrees of freedom and 3 equations. So 3 constraints are needed to perform the matrix inversion necessary to express 3.9 as a function of $W^{*,T}$. Lets say the ankle torques are all zero because we use point feet:

$$\tau_{LA} = 0 \tag{3.10}$$
$$\tau_{RA} = 0 \tag{3.11}$$
$$\tau_{LH} = -\tau_{RH} \tag{3.12}$$

where τ_{LA} and τ_{RA} are the left and right ankle torques, τ_{LH} and τ_{RH} are the left and right hip torques.

Substitution of the constraints into equation 3.9 gives:

$$\begin{bmatrix} m_x \\ f_y \\ f_z \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ & J_L^{*,T}[1] & & 0 & 0 & 0 \\ 0 & 0 & 0 & & J_R^{*,T}[1] & \\ & J_L^{*,T}[3] & & & J_R^{*,T}[3] & \end{bmatrix} \begin{bmatrix} m_{x,L} \\ f_{y,L} \\ f_{z,L} \\ m_{x,R} \\ f_{y,R} \\ f_{z,R} \end{bmatrix} \tag{3.13}$$

where [1] denotes the first row of that matrix and again the star indicates the 2D-case.

Lets call the matrix CM (for Constraint Matrix). Calculating the inverse of CM leads to:

$$\begin{bmatrix} m_{x,L} \\ f_{y,L} \\ f_{z,L} \\ m_{x,R} \\ f_{y,R} \\ f_{z,R} \end{bmatrix} = [CM^{-1}] \begin{bmatrix} m_x \\ f_y \\ f_z \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.14}$$

Finally, after substituting equation 3.14 into equation 3.8 we get to the end result:

$$\begin{bmatrix} \vec{\tau_L} \\ \vec{\tau_R} \end{bmatrix} = \begin{bmatrix} J_L^{*,T} & 0 \\ 0 & J_R^{*,T} \end{bmatrix} [CM^{-1}] \begin{bmatrix} m_x \\ f_y \\ f_z \end{bmatrix} \tag{3.15}$$

3.1.4 20sim implementation

The model used is a modified version of the original model presented in chapter 2. Both knees and the hip can be actuated. See figure 3.3. The goal is to stabilize a 2D walker using a virtual spring and damper between the hip of the walker and the ground.

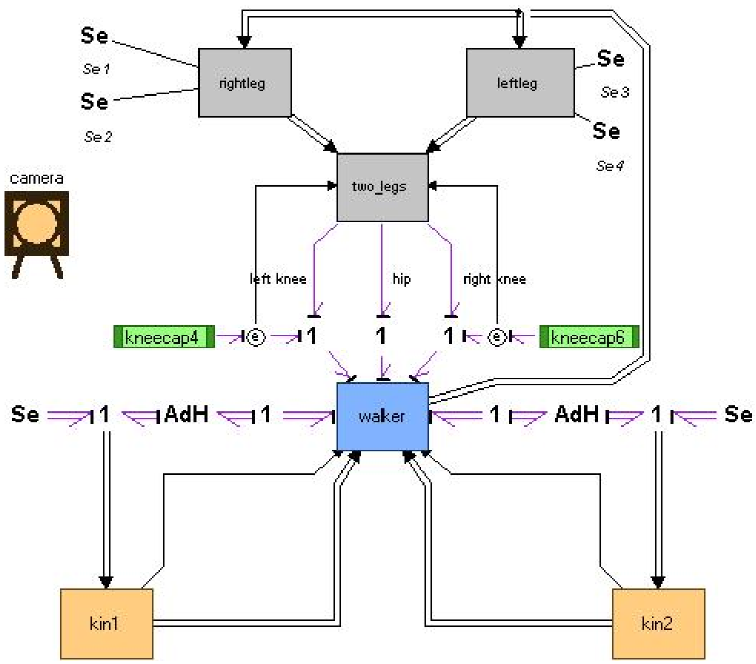


Figure 3.3: 2D controller model

The top part is the virtual model controller. There are three important blocks there: ‘leftleg’, ‘rightleg’ and ‘two_legs’. The first two are used to receive the jacobians from, as described in section 3.1.2. The ‘two_legs’ block needs these jacobians to calculate the required knee and hip torques using equation 3.15. The big arrow from the ’two_legs’ block to the other two blocks depict the joint angles vector. This vector is used to calculate the joint angles of each separate leg. The two leg submodels then calculate their jacobian and output it to the 'two_legs' block.

Two springs and dampers have been implemented in the 'two_legs' submodel. There is one spring and damper that controls the y-position of the walker and one spring and damper to control the z-position. To calculate the forces from these components we need some information from the walker submodel. The springs need distance information and the dampers need speed information.
The distance is easily determined from the absolute homogeneous matrix of the hip. The translation vector in this matrix (vector p^0_{hip} in equation 3.16) holds the displacement between the hip and the reference frame, which is the ground. The y-component and z-component of this vector are the required distances.

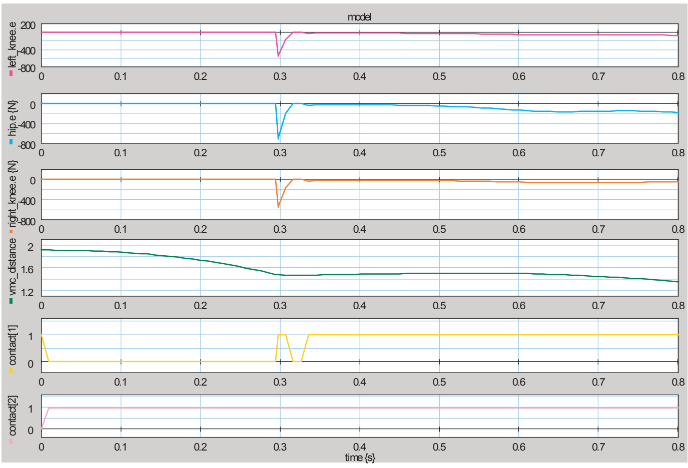
$$H^0_{hip} \quad := \quad \begin{bmatrix} R^0_{hip} & p^0_{hip} \\ 0^T_3 & 1 \end{bmatrix} \tag{3.16}$$

For the speed information the derivative of the distance is needed. This can be done by calculating the derivative numerically, but this will take a lot of computations. There is a way to calculate the derivative directly since the twist of the hip with respect to the ground is known. It can then be calculated as:

$$\dot{H}^0_{hip} \quad = \quad \tilde{T}^{0,0}_{hip} H^0_{hip} \tag{3.17}$$

3.1.5 Results

See figure 3.4 for the result of applying virtual model control with the parameters mentioned there.



Parameter	Value	Unit
k_{fy}	2	-
k_{fz}	50	-
$setpoint_z$	2.1	[m]
$setpoint_y$	0	[m]
d_y	5	-
d_z	100	-

Figure 3.4: Results of the controller and controller parameters

At a first glance it looks as if the controller does not function. This is the case in the beginning of the simulation because at that moment only one foot is in contact with the ground.
Because the virtual model controller is only used for the dual support phase a choice was made that the controller only calculates the torques when both feet are on the ground. This is because when only one foot is on the ground it is impossible to generate the virtual forces since a point foot cannot generate torque.
So in this case the joint torques are set to zero and the walker will collapse, until both feet are on the ground. We see that the controller starts to control when both feet are on the ground; a damping

effect is visible. Also it is visible that the spring constant of the z-spring is not big enough to raise the walker again.

Increasing this constant will not have the desired result due to the fact that when only one foot is on the ground, the virtual spring gets more and more compressed (since the distance to the ground decreases) and when both feet are on the ground this spring then tries to raise the walker with a very large force. With a bigger spring constant this effect will be enlarged.

Changing the initial joint angles does not result in a more firm contact with the ground. A configuration as in figure 3.5 was tried, with no effect.

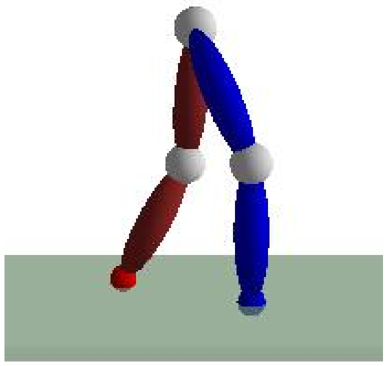


Figure 3.5: A different initial configuration

Only for small spring values the contact with the ground is firm in the beginning, but because the spring values are so low the walker collapses (as if there is no spring).

No more effort was put into further study to solve these problems because in this project we focus on the single support phase.

3.2 Single support: Foot Placement

In [9] an elegant solution to control the lateral stability of a two legged walker has been presented that still respects the natural dynamics of the walker.

They call the method 'foot placement', which means that with each step it is tried to place the foot of the swing leg such that the kinetic energy of the walker is completely converted into potential energy when the body moves over the support foot.

A nice feature of this method is that it is very energy conservative and stable. The simple controller presented in chapter 2 uses fixed setpoints. It can happen that the kinetic energy of the walker is too big or too small for that fixed setpoint. In that case it will fall forwards or backwards. Other controllers, like many humanoids use, try to mimic the human motion by controlling the legs to follow a certain position path. Effectively this suppresses the natural dynamics of the walker because the controller has to absorb all the kinetic energy that is stored in the legs when the desired position is reached.

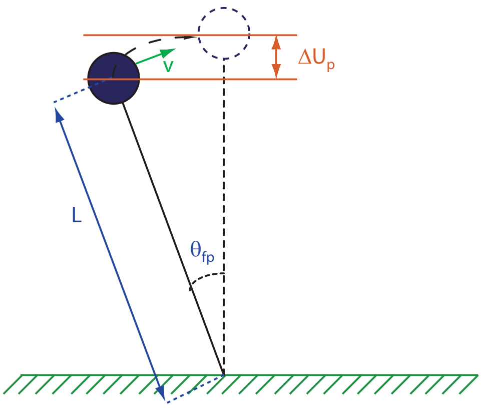
In here we demonstrate the foot placement technique in combination with the 2D walker model of

3.3 2dmodel

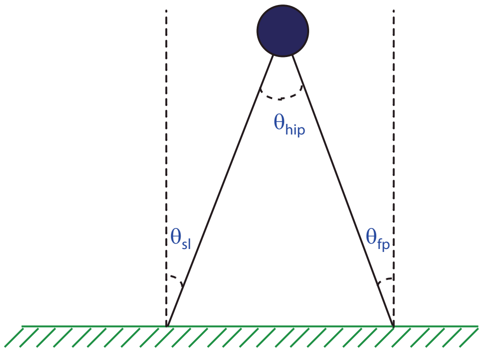
. It will be shown that foot placement can be used to get the walker into a stable gait. Also it will be shown that with foot placement it is possible to regulate the velocity of the gait.

3.3.1 Foot placement theory

To determine the required angle for the foot placement technique the very simple inverted pendulum model of figure 3.6(a) is used.



(a) Inverse pendulum model.



(b) Foot placement angle.

Figure 3.6: Determining the foot placement angle.

In figure 3.6(a) θ_{fp} is the angle that needs to be calculated. L is the length of the total leg. The vector v is the speed of the walker, which says something about the amount of kinetic energy present in the walker. ΔU_p is the change in potential energy when the walker is in its up most position.

The kinetic energy U_{kin} of the walker is:

$$U_{kin} = \frac{1}{2}mv^2 \tag{3.18}$$

When the mass transfers to the highest point the potential energy U_p increases with:

$$\Delta U_p = mgl(1 - \cos\theta_{fp}) \tag{3.19}$$

We want that the mass is in rest when the highest point is reached, which means that the change in potential energy is equal to the kinetic energy. This gives:

$$\cos\theta_{fp} = 1 - \frac{v^2}{2gl} \tag{3.20}$$

Small angle approximation results in:

$$\theta_{fp} = \frac{v}{\sqrt{gl}} \tag{3.21}$$

The result of equation 3.21 is used to determine the required angle. We assume that the knee of the stance leg is locked. We also assume that the knee of the swing leg is locked just before it gets in contact with the floor. In that case we can control the swing leg angle with only one actuator, located in the hip. The angle that the hip has to have can be found from figure 3.6(b):

$$\theta_{hip} = \theta_{sl} + \theta_{fp} \tag{3.22}$$

Where θ_{sl} means the angle between the stance leg and the normal of the ground. At each new step the new θ_{fp} is calculated and remains constant during that same step.

A smart reader might ask why the walker will move beyond the point where the potential energy is maximum, since all kinetic energy was converted into potential energy it cannot move further and it would just be a stabilized inverted pendulum. This is obviously true when the legs are completely massless. But in reality the legs will also have mass (this is also assumed in the following simulations). Because of this the center of mass of the robot will be in front of it when the swing leg passes the stance leg and the potential energy will be converted into kinetic energy again.

3.3.2 Extended foot placement controller

It can be useful if we can control exactly the speed of the walker. This means that we need to be able to control the kinetic energy of the walker. We therefore have modified the original controller from [9] with a new expression for the kinetic energy:

$$U_{kin} = \frac{1}{2}mv^2 + K_{fp}(\frac{1}{2}mv_{des}^2 - \frac{1}{2}mv^2) \tag{3.23}$$

where v_{des} is the desired velocity and K_{fp} is the feedback gain.

Now if we again want $\Delta U_p = U_{kin}$ we get:

$$\begin{aligned} \cos\theta_{fp} &= 1 - \frac{v^2 + K_{fp}(v_{des}^2 - v^2)}{2gl} \Rightarrow \\ \theta_{fp} &= \arccos(1 - \frac{v^2 + K_{fp}(v_{des}^2 - v^2)}{2gl} \end{aligned} \tag{3.24}$$

What this does is that the controller is fooled to believe that it is walking at a higher speed than in fact is the case. To sustain this higher speed a bigger foot placement angle is required, as can be seen in equation 3.24. When the desired velocity is reached the second term in (3.23) disappears.

So the control law is simple: increase the foot placement angle if the desired velocity is bigger than the actual velocity and decrease it when the desired velocity is smaller than the actual velocity.

It has to be realized that the K_{fp} cannot be very big since the argument of the arccos operation in equation 3.24 has to be between -1 .. 1. Also, a too big angle can result in the walker falling forwards.

3.3.3 Implementation

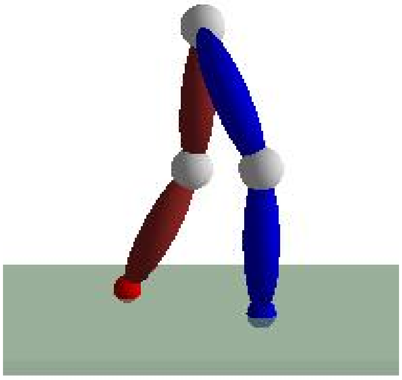
In figure 3.7 we can see the walker and its implementation in 20sim. The bottom part of the 20sim model is as described in chapter 2.

The block named 'fp_controller' is the foot placement controller. It calculates the required angle of the swing leg.

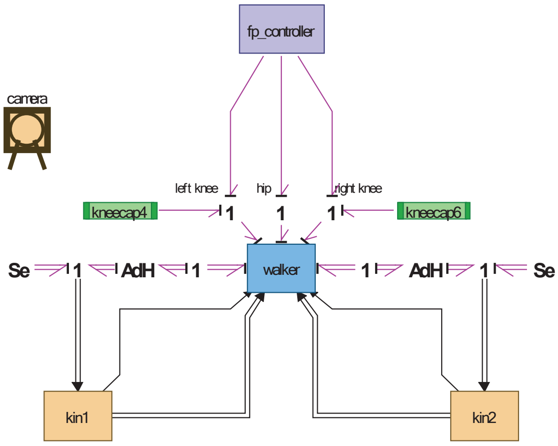
3.3.4 Results

A plot of the most interesting variables during simulation can be found in figure 3.8(a). During the first 3 seconds the walker is controlled by a simple PID controller to give it a starting velocity. After 3 seconds the foot placement controller starts to function. It is clearly visible that the controller is capable of sustaining the gait started by the PID controller. From the plot of 'theta_fp' we can see that each step the controller makes some small adjustments to the required foot placement angle. From the 'v_kin' plot we see that the speed of the walker is nearly constant (at each new step).

Using the extended foot placement controller it is also possible to control the speed of the walker. This can be seen in figure 3.8(b). At first the desired speed is 1.2 $[\frac{m}{s}]$. After 8 seconds this is changed into 0.8 $[\frac{m}{s}]$. It is clear that the walking speed converges first to 1.2 $[\frac{m}{s}]$ and then to 0.8 $[\frac{m}{s}]$. This can also be seen from the plots of the kinetic and the potential energy; they are much lower after 8 seconds. For the kinetic energy this is of course expected. Because the potential energy plot is a relative plot (it is the change in potential energy with respect to the initial potential energy) this can also be expected since the foot placement angle is smaller.

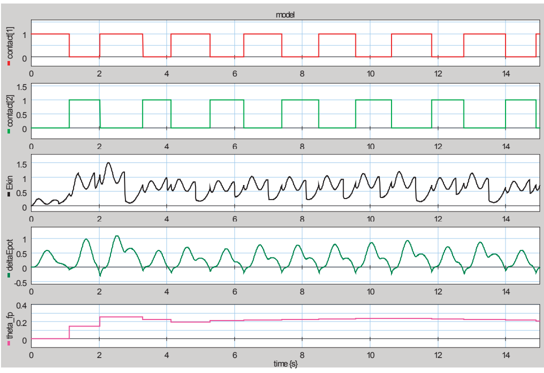


(a) The walker

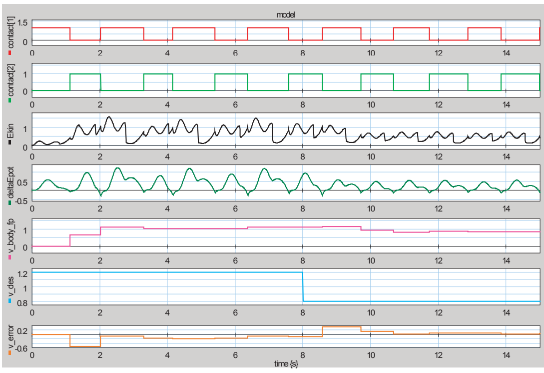


(b) 20sim model

Figure 3.7: 20sim implementation.



(a) Results of the basic foot placement controller



(b) Results of the extended foot placement controller.

Figure 3.8: Foot placement results.

Chapter 4

Energy efficient walking

4.1 Energy losses overview

During walking there is energy loss mainly due to energy dissipation when the lower leg collides with the knee end cap and because energy will be lost when the foot hits the ground. Of course also energy will be lost because of friction in joints and drive system, but we will neglect that here.

To compensate for this loss of energy we need to inject energy into the system, for example using motors. An elegant solution would be the use of adaptive springs that maintain the oscillation of the walker and only inject energy (for example using motors) when absolutely necessary (and the exact right amount). For this we have to know something about the losses.

Foot to ground impact loss When two rigid objects collide with each other in a purely inelastic way the momentum of the objects will change instantaneous. Vincent Duindam found a solution for this impact change. The solution calculates the force that makes the momentum in the direction of the normal to the ground zero.

The change of momenta can be expressed as:

$$P_+ \quad = \quad P_- + J_k^T \widehat{W}_k f_k \tag{4.1}$$

where P_+ is the momenta after the collision, P_- is the momenta before the collision, J_k^T the Jacobian relating Euclidean coordinates to internal coordinates, $\widehat{W}_k$ is a basis wrench depending on the point of contact of the collision force and f_k is the actual magnitude of this force.

The point of contact depends on the shape of the feet and ground. When a foot of a certain shape makes contact with the ground the point of contact will follow the contour of this foot. In here we will assume point feet, so that the point of contact does not change. Some extra calculations are required when the feet are not point feet, but it can be done as is presented in [1].

The change of momenta can be determined with equation 4.1. With equation 2.2 the kinetic energy can be calculated, so it is easy to calculate the difference between the kinetic energy after and before the impact:

$$Ekinloss = \Delta Ekin \quad = \quad \frac{1}{2} P_+^T M^{-1} P_+ - \frac{1}{2} P_-^T M^{-1} P_- \tag{4.2}$$

where $Ekinloss$ is the kinetic energy lost during contact.

With this the energy loss of a single impact is determined.

Dissipation in the knees Since the knee is modeled as a spring and a damper we can assume energy to be dissipated in the knees by the damper. Because the spring is quite stiff it can also be modeled with an impact model similar to how the foot to ground impact is modeled but we will stay with the spring and damper here.

4.1.1 Quantifying the losses

The model of figure 3.7 is used to determine the energy losses in the walker.

Dissipation in the knees In figure 4.1 we see quite clearly the influence that the dissipation in the knees has on the total kinetic energy of the walker. We see that the kinetic energy of the walker drops a lot each time that a knee dissipates energy.

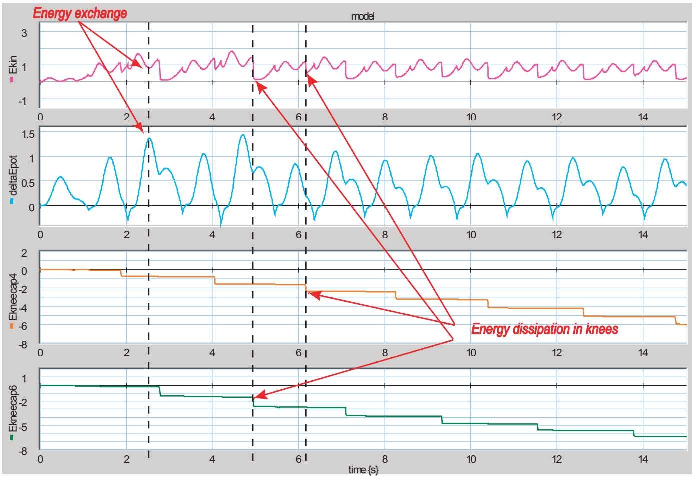


Figure 4.1: Energy loss in the kneecaps

Loss of kinetic energy because of foot impact The energy that is lost each time when a foot hits the ground can be seen in figure 4.2(a). Here we see a plot that looks like a series of impulses and a plot that shows the loss of kinetic energy because of the foot impacts.

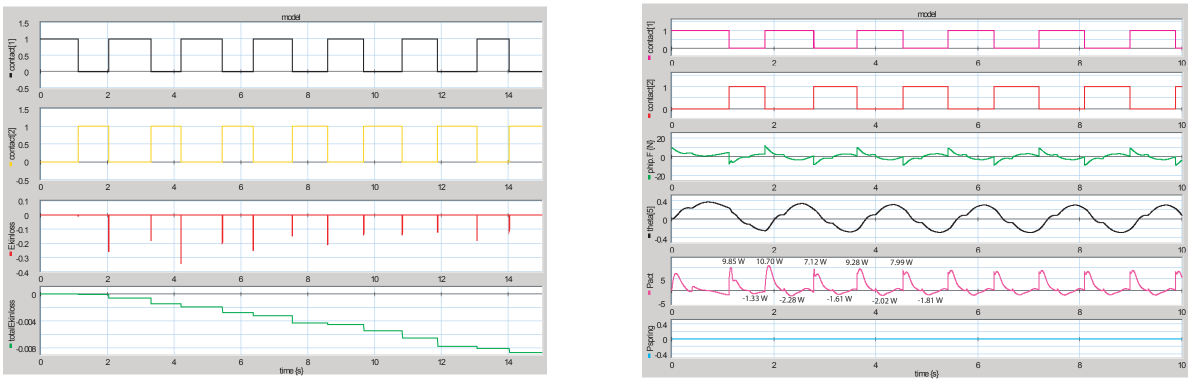
The reason why the first plot is a series of impulses is because of the way it was implemented in the simulator. Every simulation step the momenta of the walker are calculated. When there is an impact detected the new momenta are calculated. The old momenta are just the momenta one simulation step ago, so there will only be a difference in momenta at the moment that an impact is made.

So to see the loss of kinetic energy because of the foot impact the previous result is integrated. This results in the lower plot of figure 4.2(a). In the plots we can see very clear that with every impact some energy is lost.

It is visible that the loss of energy is not always the same. Some impacts result in a big loss of energy whereas others are small. This is caused by the fact that the speed of the walker is not completely constant. So the momenta of the walker are also not completely constant before each impact. The variations in walking speed are caused by the calculation of the required foot placement angle. This uses just a very crude inverted pendulum model which does not represent the walker very accurate.

4.2 Energy injection with an ideal power source

We now know why and where energy is lost. Energy has to be injected to sustain the walking. In figure 4.2(b) we see how this is done with an ideal power source.



(a) Plot of the energy lost when a foot makes contact with the ground

(b) Walker simulation plot showing the energy absorbtion.

Figure 4.2: Energy plots of the walker.

Here we see that the actuator power is at certain points below zero and thus it has to absorb energy. For an ideal power source this means that the total applied energy decreases. This only holds in theory since in reality the energy that has to be absorbed will be converted in heat by the actuator. We will now present a method that overcomes this problem by storing this energy locally in a (non-linear) spring.

4.3 Increasing walking efficiency with a spring

The purpose of this section is to study the effect that a (non-linear) spring can have on the energy consumption of the walking robot. In previous simulations we have seen that at a certain point the hip servo has to absorb energy instead of inject energy in the leg to make sure the hip does not overshoot the desired set point. This will result in a loss of energy because the absorbed energy will be converted into heat. In here we will try to store this energy in a spring so that it can be used to inject energy later on. The goal is only to study the effect of a spring and to see if it would be useful. A physical implementation is not yet needed.

4.3.1 Determining the spring characteristic equation

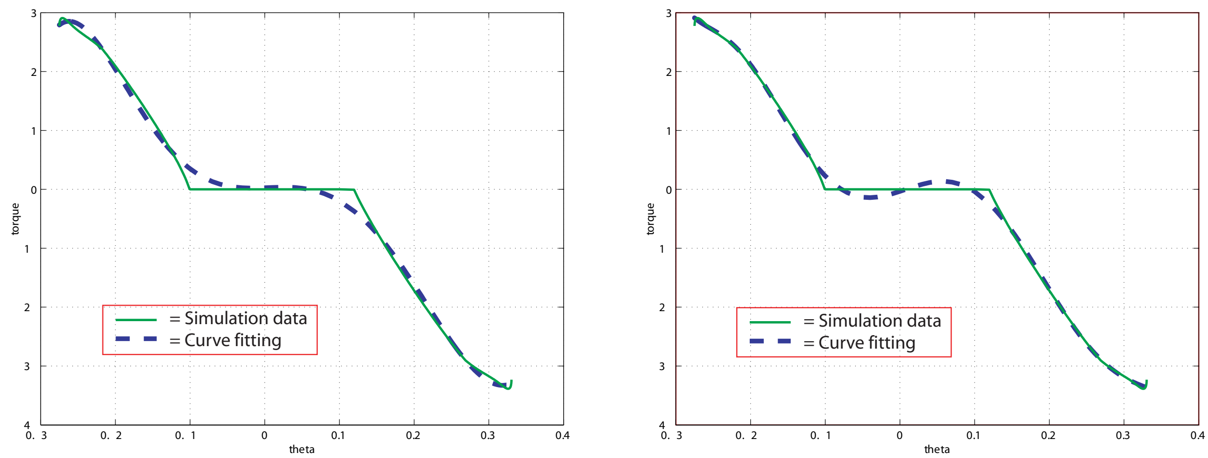
In figure 4.2(b) we have seen that the actuator power is at certain points below zero and thus it has to absorb energy.

We also see there that each step about the same power has to be absorbed. If this would not be the case it would probably be hard to store this energy. So we now need to see if it is possible to find a relation between hip torque and hip angle at the energy absorption regions.

Such a relation can be found when a plot is made of the force of the actuator on the y-axis and the angle of the hip on the x-axis. By restricting the dataset to the region where the actuator has to absorb energy, and applying a curve fitting algorithm on the dataset we get an expression that relates hip angle to spring torque. This is done twice, one time to determine the relation when the left leg is the stance leg and one time for the other leg. The results can be found in figure 4.3. In here we see two curve fitting results, one 6th order polynomial and one 8th order polynomial.

The 6th order polynomial of figure 4.3(a) is:

$$\tau = -2504.2\theta_{hip}^6 + 2664.3\theta_{hip}^5 + 113.57\theta_{hip}^4 - 346.66\theta_{hip}^3 + 4.0391\theta_{hip}^2 + 0.46599\theta_{hip} + 0.027617 \quad (4.3)$$



(a) 6th order poly

(b) 8th order poly

Figure 4.3: Plots of 6th and 8th order polynomial curve fitting.

The 8th order polynomial of figure 4.3(b) is:

$$\tau = 922293\theta_{hip}^{8} - 534190\theta_{hip}^{7} - 129170\theta_{hip}^{6} + 107340\theta_{hip}^{5} + 422.03\theta_{hip}^{4} - 719.36\theta_{hip}^{3} + 0.48393\theta_{hip}^{2} + 5.4264\theta_{hip} + 0.10162 \tag{4.4}$$

Of course the 8th order polynomial has a better fit, but it looks a bit wavy around zero. We would like this region to be almost zero so that the spring is not active in this region (most of the energy is injected in this region). So it has to be investigated if this waviness around zero is problematic.

4.3.2 Applying the polynomial spring

Figure 4.2(b) acts as a benchmark to which we compare the results of adding a spring to the walker. The goal is of course to reduce the below zero part of the actuator power.

In figure 4.4 we see the model that is used to simulate the effects of the spring on the walker.

The red C is the model of the spring. It evaluates the polynomial functions of equations (4.3) and (4.4). Since it is a C-type buffer, the input (flow in) is the angular velocity of the hip and the output is the torque of the spring (effort out). The first 1.5 seconds of the simulation are used to give the walker a start-up speed, so the spring is disabled during that period.

Comparing figure 4.2(b) with figures 4.5(a) and 4.5(b) we see that the spring does decrease the energy absorbed by the controller. In the figures the peak power values are written. Table 4.1 shows an overview of the performance of the two springs compared to walking without a spring.

We see that indeed the springs improve energy efficiency, because the peak applied power is reduced (the spring returns the stored energy at this point) and also the peak power that has to be absorbed is reduced (now the spring stores this braking energy). Between the two springs only a slight different performance can be seen. The 6th order polynomial performs better in reducing the peak power absorbed, but the 8th order polynomial performs better in reducing the peak power applied.

4.3.3 Optimization of the spring

We have seen that it is possible to reduce energy consumption by using a spring. We will now investigate if it is possible to further improve this by the use of optimization. To make this possible an optimization factor is added to the spring equation:

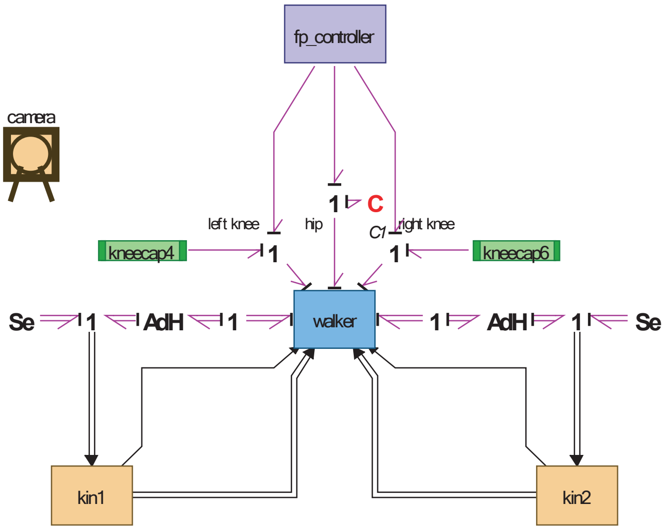
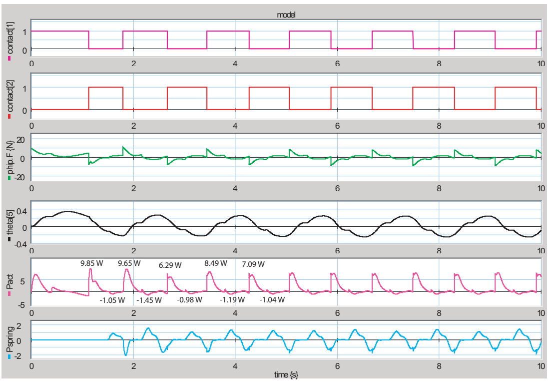
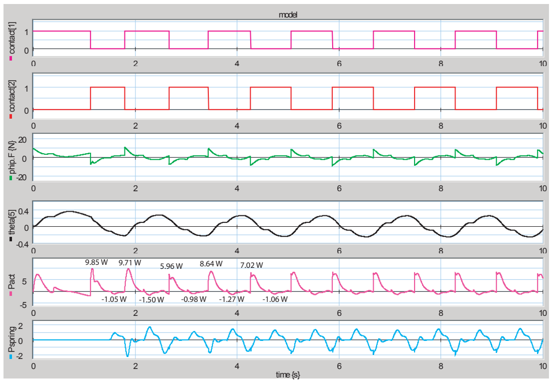


Figure 4.4: The simulation model used to simulate the walker with spring.



(a) Plots of a simulation run with 6th order polynomial spring.



(b) Plots of a simulation run with 8th order polynomial spring.

Figure 4.5: Results of applying a 6th and 8th order polynomial spring.

	Peak power applied [W]	Decrease [%]	Peak power absorbed [W]	Decrease [%]
No spring	8.99	0	1.41	0
6th order spring	8.27	7.94	1.14	19.01
8th order spring	8.24	8.37	1.17	16.88

Table 4.1: Peak power values.

$$\tau = K_{optimize}(-2504.2\theta_{hip}^6+2664.3\theta_{hip}^5+113.57\theta_{hip}^4-346.66\theta_{hip}^3+4.0391\theta_{hip}^2+0.465999\theta_{hip}+0.027617) \tag{4.5}$$

where $K_{optimize}$ is a multiplication factor.

In table 4.2 a parameter sweep of $K_{optimize}$ is shown to find a good starting point for an optimization run. The optimization criterion is to try to minimize $E_{neg,act}$, which is the total absorbed actuator energy after 5 [s], by changing $K_{optimize}$.

$K_{optimize}$	$E_{neg,act}$
1	0.8581099196842
2	0.704388814258
3	0.6403101746123
4	0.5938490325467
5	**0.5726871319131**
6	0.5948029102804
7	0.6499553147989
8	0.6555224141655
9	0.6682932556591
10	0.6804506884764

Table 4.2: Parameter sweep

$K_{optimize}$	$E_{neg,act}$
5	0.5806375274473
5	0.5718805645479
4.9998	0.5745250765674
5.0002	0.5724215442359
5.0002	0.5734867560312
5	**0.5715699212476**

Table 4.3: Optimization results of Broydon Fletcher Goldfarb Shanno gradient search method, tolerance 0.001

According to the parameter sweep of table 4.2 we see that a factor of 5 results in the lowest $E_{neg,act}$.

To see if there are more optimal values around 5 an optimization run was done. See 4.3 for the results of this. This result also shows that an optimization factor of 5 is the most optimal.

The optimal $K_{optimize}$ was used to do a simulation run. This is shown in figure 4.6.

We see that even less negative energy has to be absorbed and also that the peak power output is lower than in the previous cases.

4.3.4 Other possible approaches

The approach used here assumes that it is possible to create a spring structure that satisfies the polynomial. This does not have to be the case. A more practical solution can then be to use an existing spring structure and try to find the optimal values for the parameters of this structure.

An interesting spring structure that can be used is presented in [2]. Using an optimization run that tries to minimise $E_{neg,act}$ by changing the length of the links and the spring stiffness it can be possible to achieve a similar result as with the polynomial spring.

A benefit of this approach is that it can be implemented physically.

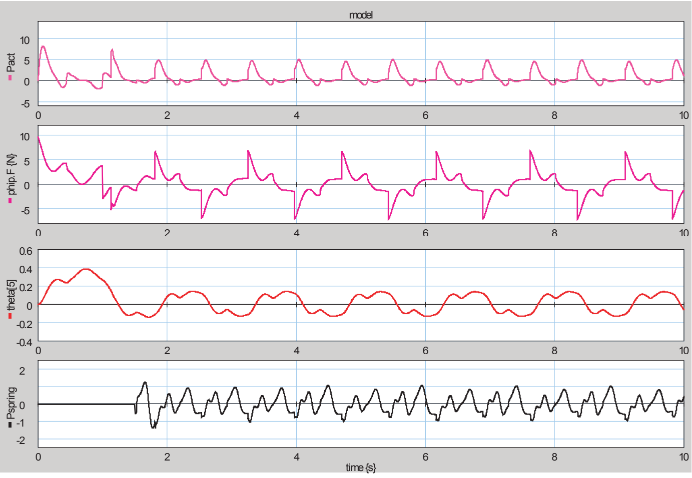
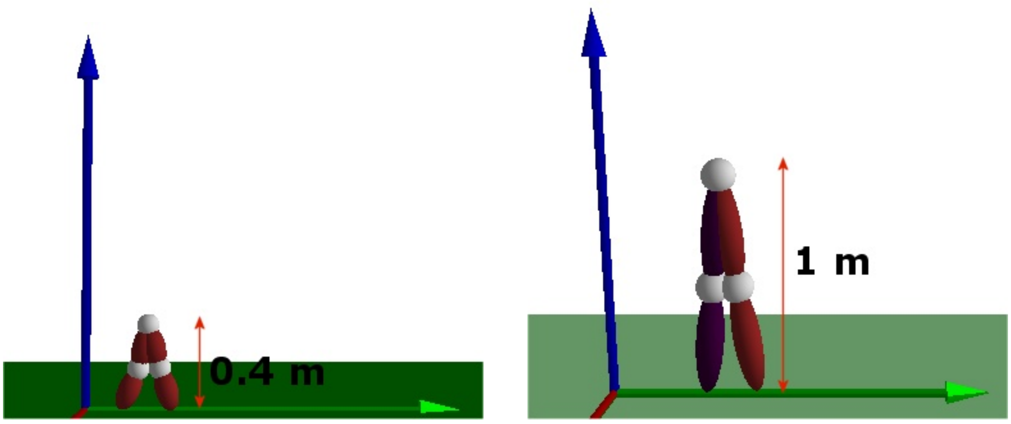


Figure 4.6: Plots of a simulation run with the optimization results used.

Chapter 5

Derivation of high level design requirements

In chapter 3.2 we have seen that it is possible to use foot placement to control a walking biped. The dimensions of the model of the walker there are somewhat unrealistic: it is two meters high. In this chapter we will derive high level design choices for the walker design. We will compare simulations of two different sized walker models (see figure 5.1) to make a sensible choice on the height of the walker. Also some robustness tests will be presented to see if the model can handle a sudden step in the floor height.



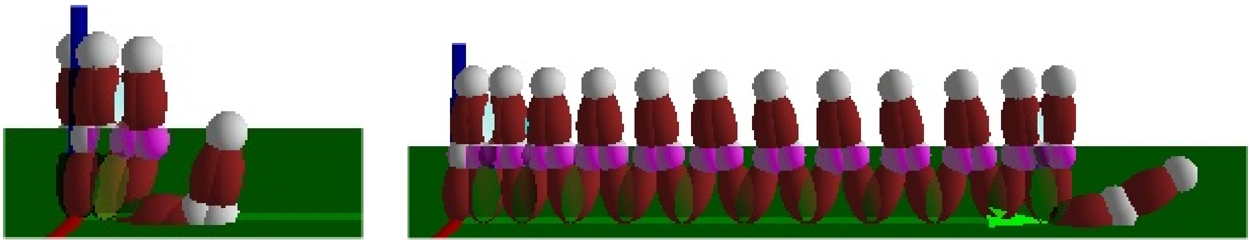
(a) Small walker (b) Medium walker

Figure 5.1: Simulation models of smaller walkers.

5.1 Small walker

In figure 5.1a the simulation model of a small walker is shown. This walker is 0.4 [m] high. Figure 5.2a shows elapsed time snapshots of a simulation run with the initial setpoint of the hip at 0.2 [rad]. We see that the walker falls almost immediately, after only one or two steps. Other values for the initial setpoint of the hip have similar results. The foot placement controller is not yet used because it requires an already walking walker.

The reason that the walker falls so quickly is because the swing leg is not fully stretched at the moment it hits the ground. This is because of the lower inertia value of the lower leg (because it is smaller). To solve this a rotational spring is added in the knee joint. A too stiff knee spring reduces ground clearance too much, because the leg then behaves as a leg without a knee. If the knee spring is too compliant it cannot stretch the leg completely before the leg hits the ground. After experimenting it was found that a spring value of 5 $[\frac{N}{rad}]$ is the best (in terms of number of successful steps). Compare figure 5.2b with 5.2a to see the performance gain.



(a) not spring assisted; 0.5s time interval

(b) spring assisted, K=5; 1 step time interval

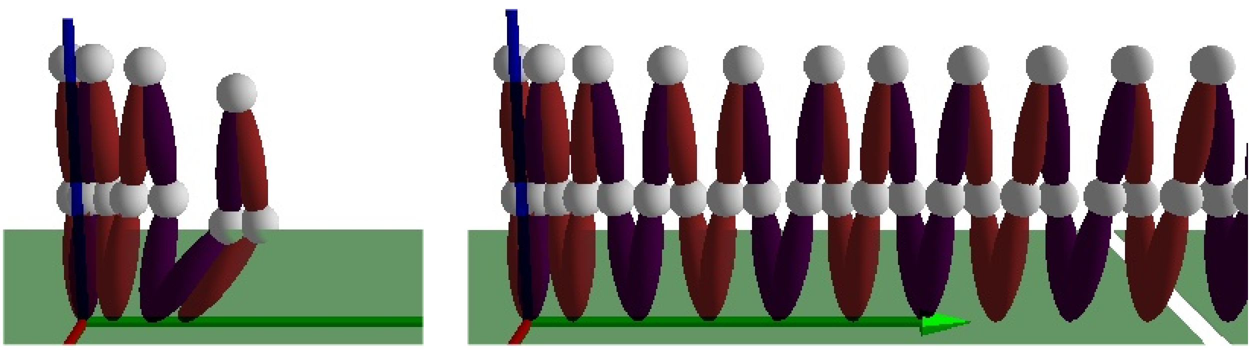
Figure 5.2: Elapsed time snapshots of the small walker walking.

We see that eventually the walker still falls, even with spring assisted knee joints. The cause of this can also be found in figure 5.2b: the knee of the stance leg bends a bit right before the swing leg hits the ground. A stiffer knee spring would solve this, but then the ground clearance is too small for walking.

For a small walker like this it might be necessary to use active control in the knees, for example a device that locks the knee when it is in the stance phase. But first we will see if increasing the dimensions of the walker helps.

5.2 Medium size walker

In figure 5.1b the medium size walker is shown. This one is 1 m high. In 5.3 we see that the walker will fall without the springs in the knees. But when a spring with a value of 2 $[\frac{N}{rad}]$ is used the walker does walk stablely.



(a) not spring assisted; 1 step time interval

(b) spring assisted, K=2; 1 step time interval

Figure 5.3: Elapsed time snapshots of the medium walker walking.

In 5.4 a 120s simulation run is shown. For the first 5 seconds the simple PID controller is used, the rest of the run is with the foot placement controller.

The simulation parameters for this run are in Table 5.1. Some parameters deserve more attention. First of all the walker length, we see in the table that it is 2 [m]. This is the length that is used in the foot placement controller. When a length of 1 [m] is used the controller the steps that are made by the walker are too big and it will fall. By choosing it to be 2 [m] the controller functions properly.

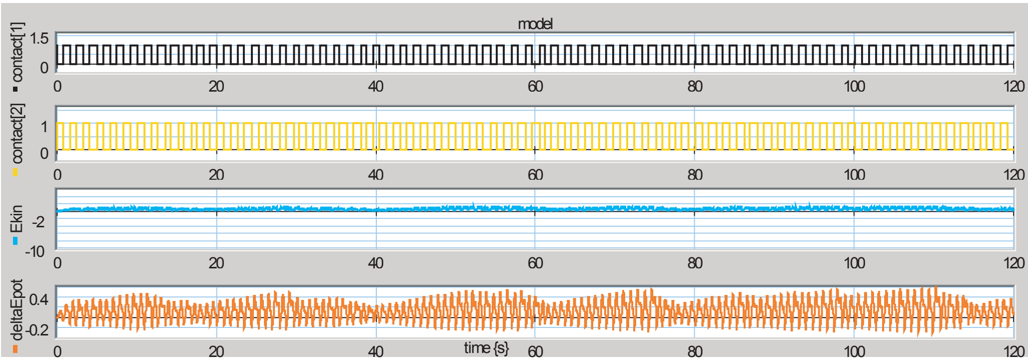


Figure 5.4: 120s simulation run with foot placement controller

Parameter	**Symbol**	**Value**	**Unit**
Hip servo gain	K_{hip}	12	-
Hip servo damping	D_{hip}	1	-
Hip setpoint	SP_{hip}	0.25	[rad]
Knee spring constant	$K_{kneespring}$	2	$[\frac{N}{rad}]$
Kneecap spring value	$K_{kneecap}$	100	$[\frac{N}{rad}]$
Kneecap damper value	$D_{kneecap}$	10	$[\frac{Ns}{rad}]$
Desired velocity	v_{des}	0.3	$[\frac{m}{s}]$
Walker length	L	2	[m]

Table 5.1: Simulation parameters

Also there is a parameter Desired velocity of the extended foot placement controller. This is required because the step size decreases each time a step is made (by the foot placement controller). The desired velocity value makes sure that the walker has a certain basis speed.

5.3 Robustness against ground variations

By introducing a (negative) step in the height of the floor the robustness against ground variations of the gait of the walker can be tested. This is important to know because it gives a requirement on how smooth the floor has to be.

This is implemented in the model by lowering the contact point of the foot with the ground at certain points. It appears that without modifications of the controller it can cope with step up to -4 [mm]. This is mainly due to limited foot to ground clearance: the swing leg will hit the ground before it is stretched.

The walker has a quite low ground clearance because the knees are not actuated. This happens also in human walking: when a person is strolling, often the foot will hit the ground before a step is made. During strolling the knees are actuated only little and this results in low ground clearance.

The low ground clearance is not a big problem because normal quality floors do not have steps bigger than 4 [mm]. It is however something that has to be taken into account when designing the feet: they should be small so that they do not affect the ground clearance.

5.4 Recommendations for the mechanical design

From simulation we found 'bigger is better'. A bigger walker requires less 'tricks' for walking. It is difficult to get the small walker to walk, probably active knees are needed. The medium size walker also needs some help to stretch the legs fully, but a simple spring and some modifications on the foot placement controller suffice. We already found that the big walker can walk without knee springs and with the simple foot placement controller.

A wise choice is to design the walker to be approximately 1 [m] high, since with this height it is still possible to walk and is also not to big to be impractical. The knee should have a spring integrated and it is useful if this spring can easily be adjusted to tune walking.

Feet should be kept small so that they do not affect the ground clearance.

Part II

Development of the 2D walking robot

Chapter 6

Design options

In this project we want to make a walker that can be used to test the control algorithms mentioned above. A very important aspect is that the walker has to be designed in such a way that the natural dynamics of it are preserved. This imposes some requirements on the mechanics and on the control of the walker. We will discuss this in more detail later.

Because walking is very complex a choice was made to first start with a 2D walker. This constraints yaw and roll and therefore increases the chances of success.

We will divide the design in two parts: the mechanical design and the electrical design.

Here we will first present the various options that have to be evaluated and problems that have to be taken care of. Then, we make a high level choice for a certain design and create requirements for the parts the design will consist of. In the next chapter this will lead to a detailed design proposal.

6.1 Mechanical design options

Some considerations can be made before starting to design the walker. Some important issues are

- the necessity for feet
- the implementation of the 2D constraints

6.1.1 Feet or no feet

The 2D simulation model that was used to verify the effectiveness of foot placement has point feet. With this it is possible to achieve a stable gait, in simulation. On the other side, [13] showed that a foot with a large radius is advantageous for a high disturbance rejection.

An extra benefit from using feet is that it is possible to measure the angle that the legs makes with the ground from a sensor mounted on the ankle joint. This angle is required when foot placement is used for control.

But the foot placement method uses an inverted pendulum model to calculate the swing leg angle. For this model it is required that the contact with the ground is only on one point. This would require point or flat feet. Curved feet cannot be used since the contact point between foot and ground moves along the curve of the foot.

For now a probably safe choice would be to use small feet that are neglectablely small with respect to the length of the walker, but are still sufficiently big to be able to determine the angle of the leg with respect to the ground.

6.1.2 Limiting sideways motion

The walker has to be constraint in the sideways direction. This can be done by guiding the walker so that sideways motion is impossible. Another possibility is to build a walker with two pairs of legs.

Guiding

One solution can be to use guide lines that make sure that the walker does not fall sideways. This is not a very strict constraint because guide lines are not very stiff. But it can be effective if the walker is not very unstable sideways.

Another solution can be to make some sort of tunnel without a ceiling in which the walker can walk. This is a very stiff sideways constraint, but is more difficult to make than the simple guide line solution.

A benefit of the guiding approach is that the walker still looks like a 'normal' biped, because it has two legs. A big disadvantage is that it is not autonomous. It can only walk where the guiding is.

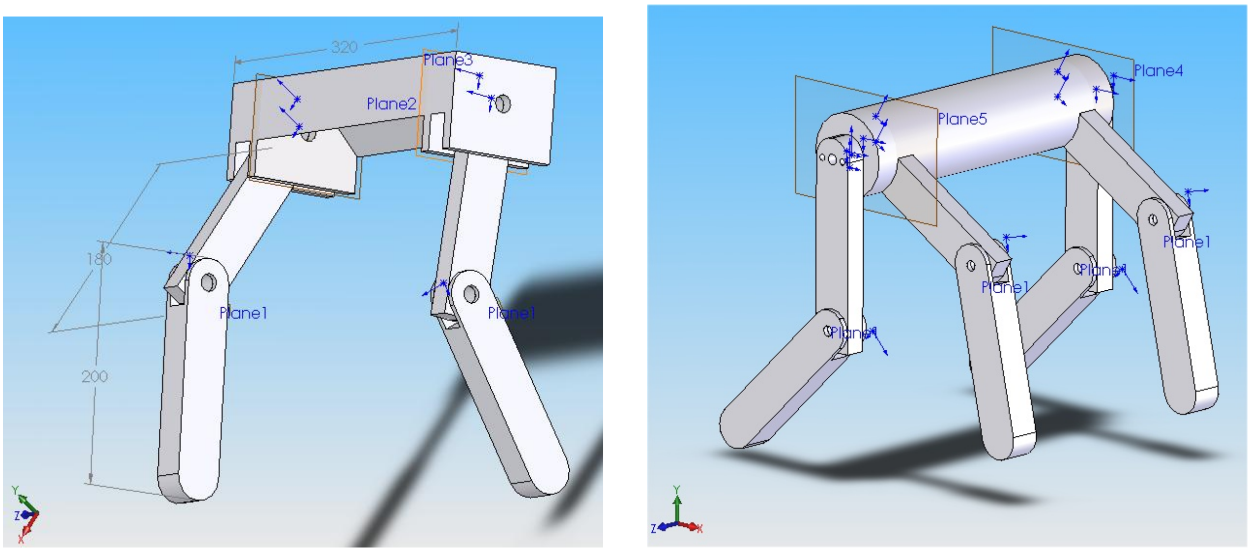
See figure 6.1(a) for a possible walker design that requires guiding.

Autonomous

An autonomous walker will require a 2D constraint that is integrated in the design itself. This can be done by placing small wheels at each side of the robot, like the way how kids learn to ride a bike. It will be difficult to implement for a walker because the height of the walker changes during each step. This will require variable length connections between the walker and the wheels.

A more elegant solution can be to use two pairs of legs. The two legs of a pair behave like each other, so that when seen from the side there is not a difference between a two legged and a four legged walker. This would mean an outer set of legs, and an inner set of legs.

See figure 6.1(b) for a possible design with two pairs of legs.



(a) Mechanical design of a walker which needs guiding

(b) Mechanical design of an autonomous walker

Figure 6.1: Different walker designs.

6.2 Electrical (control) design options

The control part of the walker can be divided in three parts:

- Actuation
- Sensing
- Control

These will be discussed in the next sections to formulate a first method of approach for designing the walking robot.

6.2.1 Actuation

A couple of questions arise when thinking of actuation of a walking robot. How should actuation be done (servo, direct drive, McKibben muscles etc.), where should it be done (in the hip, in the legs, in the joints etc.) and also very important, should it even be actuated? It does not make sense to add a degree of freedom (DOF) when it is not used or not necessary. This section will try to answer some of these questions. From chapter 5 we know that walking is possible with only one actuator (in the hip) and locking devices in the knees. That is also what will be used as a starting point for the design, but here we will also show some possible solutions when more than one actuator is desired.

Actuation using motors located in the hip

A design decision can be to have the actuators, and thus most of the mass of the robot located in the hip. As a result the mass of the legs will be low. This is desired because then it will be easy to swing them.

This method also requires energy transfer over quite a big distance. A method for this is to use cable on a pulley. Using the pulley a high reduction ratio can be achieved with low losses because there is only one reduction stage needed. The flexible cable also has the advantage that it is very backdriveable. This is especially important if the dynamics of the walker should be preserved. Using two pulleys that rotate in opposing directions both swing directions can be controlled. This is much like the biceps and triceps in the human body.

A (major) disadvantage of this method is that accurate position control will be quite difficult because of the large elasticity of the cables. It will probably not be as accurate as a direct transmission. This is not necessarily a problem because accuracy is not the most important design criterion of a walker.

However a big advantage of this method is that it can be build using very common components, and therefore will be quite cheap.

Actuation using linear actuators located in the legs

This option will make for a lighter hip but heavier legs. From a stability point of view this is good, because the center of gravity is more close to the ground and so the robot will be more stable. It requires more power to swing the legs because they are heavier than the former option, but it will require less power to move the hip mass over the stance leg.

Another benefit is that position control will be more accurate because the linear actuators have a direct connection to the legs. This direct connection however has the drawback of low backdriveability. Especially when using some kind of screw spindle, which is hardly backdriveable. If this is to be used something has to be made that can decouple the actuator from the leg, so to enable a free moving leg. Otherwise it will not be possible to make use of the dynamics of the walker.

If it is possible to decouple the actuator from the legs than springs can be used for energy storage. This will make it possible to use less powerful actuators by exciting the legs at the resonance frequency of the springs.

Actuators located in the legs will complicate the mechanical design of the legs because they will need to be able to move through the legs, or they need to be mounted on the sides of the legs.

Actuation using servos in the joints

It is obvious that in a discussion about different possibilities for actuation servo joint actuators cannot be omitted. Servos offer direct control over the joint angles because they are actuator and joint at the same time. This is very useful if for example virtual model control is used to control the robot. The virtual model control gives the joint torques that are required for a certain task, so servos do not require a mapping from joint torques to actuator forces because the joint torque equals the servo torque.

A drawback of a normal servo joint actuator is that the reduction stage has some friction, which makes it less efficient when trying to preserve the dynamic nature. Also, the controller of the servo should be disconnected when the angle of the joint is not important to make sure that the joint can move freely. A big disadvantage of a motor with gearbox is that the gearbox transforms the motor resistance with a factor that is quadratically proportional to the gearbox ratio. So external forces applied to the gearbox shaft will see a lot of resistance. Obviously this is disastrous for backdriveability.

This problem can be escaped when direct drive torque motors are used. These motors offer a very high torque compared to normal motors, at low rotational speeds. A reduction stage is not necessary anymore. This kind of motors is very efficient because there is only friction in the bearings inside the motor. Using direct drive motors it is very easy to ensure backdriveability because the joint can rotate freely when the motor is electrically disconnected.

However, direct drive motors this small and that operate on a low voltage are difficult to find (and very expensive). Another possibility to ensure backdriveability with a motor with gearbox is to measure the torque between the gearbox shaft and the object that has to be actuated. If an external force is applied to the object this will generate a torque on the gearbox shaft because of the motor resistance that is transformed over the gearbox. This torque can be controlled to zero by measuring it and applying a certain motor power in the right direction. This will effectively make the motor turn along with the direction that the shaft is being turned. The external force will no longer see a big resistance so that the actuator is now backdriveable.

Other alternatives

One can look into McKibben muscles [4], series elastic actuators [7], upper body actuation [6] etc, but in this project we want to use only electrical actuators.

6.2.2 Control

In chapter 3 we have seen two methods for controlling a biped walker. Virtual model control can be used to control the static case (both legs in contact with the ground) while foot placement is a good technique to control the dynamic (walking) case.

Of course other alternatives are available. [13] applies power to a leg in an on/off-fashion depending on contact with the ground of the opposite leg. [3] uses an advanced control algorithm called 'posture control' that changes the body's position or makes an extra step when a tipping force cannot be resisted anymore.

Also a choice has to be made between local control or remote control using an umbilical cord. If the robot has to be fully autonomous obviously control has to be done locally. This requires a local energy source (batteries) and a local CPU. This puts some limits on actuator power and computing power.

6.2.3 Sensing

The same questions as in section 6.2.1 can be asked when regarding sensing. How to do sensing, where to do sensing and why to do sensing to see if it is even necessary.

To make a sensible estimation on where sensors are needed one needs to know roughly how the robot will be controlled. In section 6.2.2 some control methods were discussed. The foot placement requires the measuring of the angle of the leg with respect to the ground normal to see if the foot is placed correct. This cannot be derived from the joint angles if the robot has point feet, because what is the angle of the foot to the ground? *Note: the same problem exists when a 'normal' foot is not in contact with the ground.* Therefore the angle of the leg with respect to the ground has to be measured differently.

Also the forward robot speed has to be known for foot placement. This is required to be able to calculate how much kinetic energy the robot has.

For virtual model control all the joint angles have to be known. These are relative to the frame of the link on which the joint is fixed and therefore quite easy to measure. Less easy to measure is the angle of the body (the hip) with respect to the reference frame which is the ground. Note that when this angle is known also the angle of the leg with respect to the ground is known.

Now a short enumeration will be given of various sensing options that are available.

Joint angles

The joint angles are not so difficult to measure. Probably the most easy and most used option for this is to use (optical) encoders. A contact free measurement is preferred (because low resistance is of utmost importance) so potentiometers are out of the question. This leaves us with sensors based on optical (e.g. optical encoders) or electromagnetic principles (e.g. resolvers). Optical encoders are then preferred because they make for a digital read out and do not require analog to digital conversion.

Body angle

If it is assumed that the robot is walking on a level surface it is possible to use a gyroscope to measure the angle between the body and the normal of the ground. This is a nice way because it is contact free.

Another contact free solution would be to use optical reflection. Then the distance between the reflection and the light source on a photo sensitive surface, together with the time of flight of the light will give the angle of the body to the ground. This will probably be quite difficult to get working reliably.

Also cameras can be used to determine the body angle but that will be a difficult technique to master.

An easy solution is to use small, non-point feet with a flat sole. The angle of the ankle joint can then be referenced to the ground. The contact point of flat feet also does not change during a step, just like point feet so for foot placement it does not matter a lot. A problem is that some parts of the foot are in contact with the ground more early than others so it is hard to define the exact time of contact.

Robot speed

Because the previous measurement techniques will uniquely determine the configuration of the robot it will be possible to calculate the robot speed. This can be done by calculating the twist of the hip with respect to the ground. All H-matrices that are needed for this are known, and can be differentiated to give the required twist. The horizontal component of the linear speed of this twist will give the robot speed.

It will be quite difficult to try to measure the speed because there is no direct connection to the ground (or the speed of some wheel that is continuously pushed to the ground has to be used).

6.3 The influence of mechanical non-idealities

The mechanical design is the physical reflection of the simulation model. The goal is to approach the simulation model as closely as possible. But one has to realize that a simulation model is often based on idealized components, which are not available in real life. For example, the links of the legs of the walker are considered as ideal rigid bodies during simulation, but in reality they of course have a certain (limited) stiffness.

It is important to look at the non-ideal nature of a physical implementation when designing a mechanical structure from a simulation model. This will prevent strange, unforeseen problems to occur when the 2d walker has been constructed. The most prominent issues that deserve attention are:

- The stiffness of the links/axles and joints will not be infinite;
- The links have mass;
- The hip cannot be considered as a point mass;
- There will be play and friction in the drive system

Finite stiffness The stiffness of the links depends on the choice of material and the structure of the design, like the shape or the thickness of the link. A high stiffness is important because this is more easy to control since end effector position will be more directly related to the joint angle. The same holds for the stiffness of the axles and joints. Also, because a walker can be seen as an inverted pendulum, the links will have to support a quite substantial mass over a large distance. This is another reason that the links should be stiff enough.

The stiffness of the links itself is quite fixed. The most obvious choice is to use aluminum tube profiles because they are light weight and stiff. Larger diameter tubing or tubes with higher wall thickness can be used if the stiffness is not high enough.

To keep the torsional stiffness of the axles sufficiently large, steel axles can be used of a large enough diameter. Bending stiffness of the axle should be minimized by designing proper hinges. By supporting the axle on two points the axle will be much less loaded and this will make the joint stiffer.

Links have mass Heavy legs require a lot of motor power to move. It is relevant to keep the mass of the legs as low as possible to keep energy consumption low. One difficulty is that the legs should have quite a high stiffness and therefore will have some weight.

Also, in the inverted pendulum model it is assumed that all the mass is at the top of the model. If this is not the case the foot placement algorithm might fail. By placing all the motors, batteries and computing power in the hip it will be very likely that the weight of the hip is much larger than the weight of the legs and therefore the inverted pendulum assumption will probably still be valid.

The hip is not a point mass The inverted pendulum assumes that the mass of the hip is a point mass. This will certainly not be the case. Because it will need to hold all the motors and the gears etc it will have quite a big size. This is less a problem if the weight is divided over the hip in a uniform way. Also the inertia of the hip should be kept low so that the dynamics of the physical walker resemble the simulation model.

Non ideal drive system / transmission At certain points there will be non idealities in the drive system. Very unwanted is play in the gearbox, also known as backlash because this value gives an upper limit to the accuracy of the actuator. The output shaft cannot be positioned more accurately than the value specified as the average backlash value since the output gear can move freely without contacting the other gear in this backlash zone. See figure 6.2 for a simple sketch that further illustrates this. Play can also exist in the bearings of the hip, knees and ankles. Good quality bearings are essential.

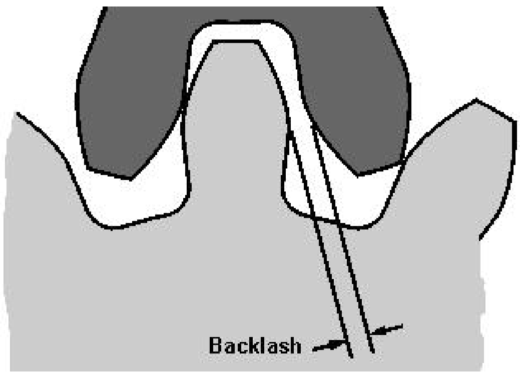


Figure 6.2: Backlash in two gears.

Also, the drive system will have some friction that should be kept low. Friction will exist in the motors, gearbox and bearings of the walker. This could be reduced a lot by using direct drive motors because they do not require gearboxes.

Since we want to exploit the natural dynamics of the structure good bearings will be necessary that have low friction. The system has to be highly backdriveable because then the drive system does not influence the dynamics.

Part	Estimated weight	Unit
Actuator	0.6	[kg]
Batteries	1.0	[kg]
Total hip	1.9	[kg]
Leg	1.47	[kg]
Total walker	7.75	[kg]

Table 6.1: Estimated weights

6.4 High level design choice

From chapter 5 we have seen that it is the most practical when the walker is approximately 1 [m] high. Before we can determine the requirements on the actuator and sensors etc., we need to have a basic idea on how the walker will look like. Also we need some rough estimations on the weight of the design.

The design will look like figure 6.1(b), only it will be 1 [m] high. We want it to be autonomous so to constraint sideways motion it will have two pairs of legs. It will have small feet to measure the angle of the ankles. Actuation will be done with an actuator in the hip.

Using SolidWorks weights were estimated of a 4 legged walker of 1 [m] high looking like figure 6.1(b). The walker is constructed from aluminum to keep weight low. See table 6.1 for an overview of the estimated weights of the parts of the walker.

6.5 Requirements of the subsystems

The control system consists of three parts: sensing, actuation and the controller itself. For foot placement it is necessary to determine the angle of the stance leg with respect to ground and the angle of the hip joint. So it requires three sensors that can measure these angles.

The only actuation is done in the hip, so just one actuator is required. We will now determine the requirements for these components.

6.5.1 Sensor requirements

The foot placement method requires that the swing leg is servoed to a certain angle with respect to ground. This angle can be measured when the angles of the hip joint and the stance leg are known. Then the swing leg angle is:

$$\theta_{swingleg} = \theta_{stanceleg} + \theta_{hip} \tag{6.1}$$

In simulations we have seen that a very accurate angle of the swing leg is not required. It is possible to multiply the angle with a factor of 0.9 .. 1.1 and still the walker walks stablely.

So we can say that the swing leg angle has to be measured with a maximum deviation of 10%.

$$0.9 \leq \frac{\theta_{swingleg} \pm \Delta\theta_{swingleg}}{\theta_{swingleg}} = \frac{(\theta_{stanceleg} \pm \Delta\theta_{stanceleg}) + (\theta_{hip} \pm \Delta\theta_{hip})}{\theta_{stanceleg} + \theta_{hip}} \leq 1.1 \tag{6.2}$$

Now if we assume that both angle sensors have the same accuracy we can rewrite this in:

$$\begin{aligned}
\Delta\theta_{sensorerror} &= \Delta\theta_{stanceleg} = \Delta\theta_{hip} \\
0.9 &\leq \frac{(\theta_{stanceleg} + (\theta_{hip}) \pm 2\Delta\theta_{sensorerror}}{\theta_{stanceleg} + \theta_{hip}} = 1 \pm \frac{2\Delta\theta_{sensorerror}}{\theta_{stanceleg} + \theta_{hip}} \leq 1.1 \\
-0.05 &\leq \frac{\Delta\theta_{sensorerror}}{\theta_{stanceleg} + \theta_{hip}} \leq 0.05
\end{aligned} \tag{6.3}$$

We now need to determine the smallest possible value of the swing leg angle that still needs to be measured accurately. From simulations we have seen that the minimum walking speed is about 0.2 [m/s]. Below that value the walker collapses. For this speed the foot placement angle is about 0.045 [rad]. A 5% deviation around 0.045 [rad] would mean 0.00226 [rad] deviation. If we use an incremental encoder for measurement then each pulse should represent maximally 0.00226 [rad]. This would mean an encoder with at least

$$PPR \geq \frac{2\pi}{0.00226} = 2783 \tag{6.4}$$

where PPR means Pulses Per Revolution.

This is quite a big number because the angle that has to be measured is small. But there are commercially available encoders with a PPR of 2000 that are not expensive. Although the required accuracy is not met at low speeds of 0.2 [m/s], at the speed where the walker functions best (at about 1.0 [m/s]) the accuracy is more than enough. Higher speeds require bigger angles and so the absolute deviation allowed can be bigger.

It is also possible to use a gearbox to enable the full use of the range of the encoder. At 1.0 [m/s] the foot placement angle is only 0.2257 [rad]. This is only a small value compared with the full range of the encoder (2π). With the gearbox the input angle can be multiplied with a certain factor before it is measured with the encoder. This drastically decreases the required number of pulses per revolution. Say we only want to measure angles of up to 0.5 rad (which is an angle that will never be reached since it is when walking faster than 2.2 [m/s]). We can then amplify with a factor of $\frac{2\pi}{0.5} = 12.6$ so the required PPR drops to $\frac{2783}{12.6} = 221$.

6.5.2 Actuator requirements

As was seen earlier it is advantageous if direct drive motors are used, because they offer the lowest possible friction of all types of electric drives.

First we need to determine some of the requirements for the actuator. Most important are the output RPM of the actuator and the maximum (and continuous) torque that can be supplied.

The 20sim model is updated with the estimations of table 6.1 to determine the requirements on the actuator.

From simulations (see figure 6.3 we see that when walking at 1.0 [m/s] the hip joint rotates at speeds of about 1.6 [rad/s]. Say that a rotational speed of at least 2.0 [rad/s] is desired. We then need an output RPM of at least:

$$RPM \geq 60\frac{\omega}{2\pi} = 19.1 \tag{6.5}$$

In 6.3 also the torque is plotted (please note that the units in the plot are not correct, they should be [Nm] and [rad/s]). The highest torque value is more than 12 [Nm]. So the actuator must have a maximum torque of at least 12 [Nm], which is high. Continuous torque is much less, around 3 [Nm].

Figure 6.4 shows a plot of the actuator torque versus the actuator angular speed. It shows that most of the actuator power is positive, since most of the points are in the upper right and lower left quadrant. But there also some values in the upper left and lower right quadrants. In those quadrants the actuator power is negative. A way to handle this was already presented in chapter 4. The power in the upper right quadrant is not exactly the same as in the lower left quadrant because of the start-up conditions. In the first seconds of the simulation the hip set point is larger than normal. This happens to be in the upper right quadrant.

6.5.3 Controller requirements

The controller will consist of two parts: a (PID)-controller for the hip joint angle and the foot placement controller that calculates the required hip joint angle to achieve a stable gait.

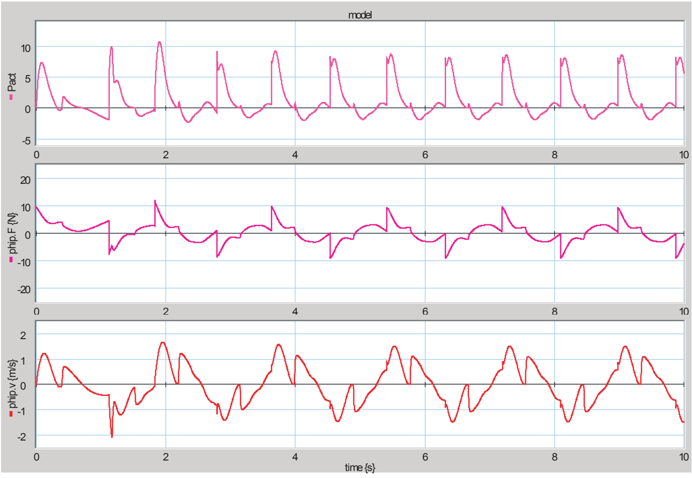


Figure 6.3: Simulation plots to determine the actuator requirements.

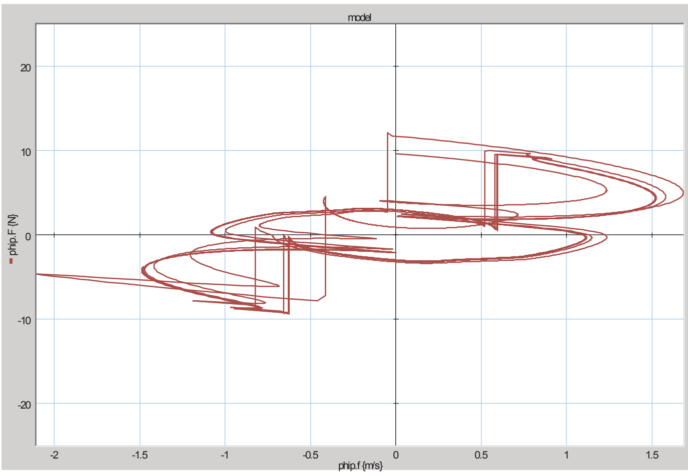


Figure 6.4: A plot of the actuator torque versus actuator angular speed.

The dynamics are not very fast so a very fast controller is probably not needed. The foot placement algorithm requires the need for division and square root operations so a controller with a good ALU (Arithmetic Logic Unit) is required.

Chapter 7

Design proposal

To see if virtual model control and foot placement are good control techniques for a two legged walker with knees, a physical model has to be designed and constructed. In the previous chapter we have seen some problems that have to be solved for this. Here we will propose a design based on the considerations of the last chapter.

7.1 Electrical (control) design proposal

7.1.1 Actuation

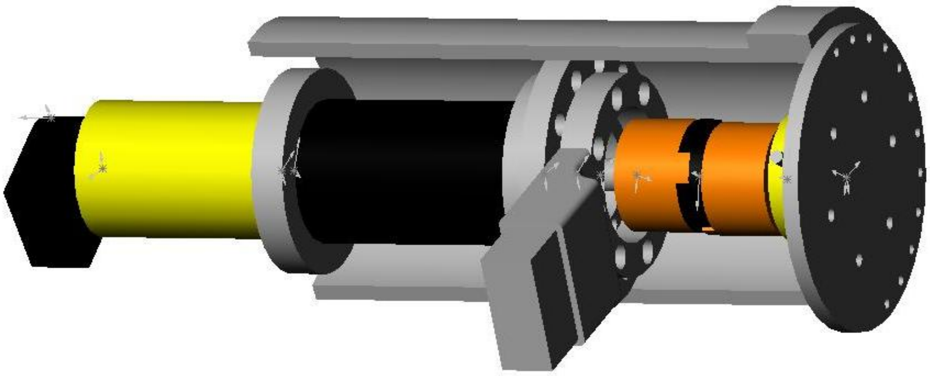


Figure 7.1: A 3D CAD image of the actuator setup (by Edwin Dertien).

In section 6.5.2 we have seen the requirements in terms of maximum torque and output speed. Since it is very difficult to find direct drive motors that are: capable of these requirements, are small and can operate on low voltage, a choice was made to use normal DC servo motors with a reduction stage. Of course good quality motors should be used to minimize friction, but also a technique will be used to 'emulate' backdriveability, which was already presented in 6.2.1.

See figure 7.1 for a 3D CAD image of the actuator setup as it designed in cooperation with Edwin Dertien. He has to be credited for most of the work of designing the actuator. In the figure we see the encoder that measures the motor position (far left). To the right of the encoder is the motor. Next to the motor is the gearbox. To overcome alignment problems an Oldham coupling is placed between the gearbox and the torque sensor (far right). Because the torque sensor is a very important part of the actuator design we discuss it here and not in the sensor section. We will now discuss the subsystems.

Motor

The motor used is the Maxon RE40-24V. This is a motor of the permanent magnet DC motor type. Some specifications are given in table 7.1. For more specifications see the data sheet in appendix B.

Motor data	Value	Unit
Power rating	150	[W]
Nominal voltage	24.0	[V]
No load speed	7580	[rpm]
Stall torque	2290	[mNm]
Terminal resistance	0.316	[Ω]
Max. continuous current	6.00	[A]
Max. continuous torque	181	[mNm]
Max. efficiency	91	[%]
Torque constant	30.2	$[\frac{mNm}{A}]$
Speed constant	317	$[\frac{rpm}{V}]$
Weight	480	[g]

Table 7.1: Some specifications of the Maxon RE40-24V motor

This motor was chosen because of its high power, high efficiency and low weight and since it can be easily connected to a gearbox. Also, the manufacturer has different encoders that can be mounted. With a suitable gearbox (which will be discussed in the next section) it matches the requirements of section 6.5.2. Furthermore, 20sim has a library consisting of Maxon motors so that it is very easy to simulate the walker if this motor is used.

The mechanical design of the motor is also very suitable because it has low axial and radial play on the output shaft, which is of course desired.

Gearbox

We have seen that the maximum torque of the actuator should be more than 12 [Nm] and the continuous torque has to be about 3 [Nm]. If we do not want to allow the motor to have to generate torques larger than the maximum continuous torque allowed, a reduction ratio of at least $\frac{12}{0.181} = 66 : 1$ is needed. Of course it is not a problem if for short moments in time the motor torque exceeds the maximum continuous torque value. A lower ratio reduction could be used, but a high reduction reduces the current draw of the motor, which is less demanding on the motor amplifier and batteries.

The actuator output speed should be at least 19 [rpm]. This means that a reduction ratio of less than $\frac{7580}{19} = 102 : 1$ has to be used. Maxon produces a wide range of planetary gearboxes that satisfy these limits but only two of them are in the stock program: the 43:1 and the 74:1. A choice was made for the 74:1 version because it reduces torque demand on the motor the most and still the output speed is high enough.

Some specifications are presented in table 7.2.

Gearbox data	Value	Unit
Reduction	74:1	[-]
Max. motor shaft diameter	10	[mm]
Max. continuous torque	15	[Nm]
Intermittently permissible torque	22.5	[Nm]
Average backlash no load	0.00873	[rad]
Weight	460	[g]

Table 7.2: Some specifications of the Maxon GP42C gearbox

The backlash of this gearbox is very low, which is a very important factor when deciding on a gearbox.

Torque sensor

The torque sensor is used to measure the torque that is applied by the motor to the leg or generated by an external force which acts on the leg. The main reason to do this was already mentioned: to emulate backdriveability. But it also creates the possibility of actuator force control. Now not only the output position of the actuator can be controlled, but also the torque that is applied. With this it is possible to give the actuator any desired behaviour (within the limits of the motor and gearbox).

This is of course very useful when spring structures as in section 4.3 are investigated because with it the desired structure can be tested before it is even build. A simple look-up table in the motor controller can map actuator positions to desired actuator torques found from the characteristic equation of the spring structure under investigation.

Figure 7.2: Transducer Techniques TRT-50 reaction torque sensor.

The torque sensor used in this project is the Transducer Techniques TRT-100, similar to the one showed in figure 7.2. The most important specifications are given in table 7.3 and the full specifications can be found in appendix B. Its range is 100 [in. lb.] which corresponds to 11.3 [Nm] in metric units. This range is big enough even though momentarily torques slightly bigger than this can occur because the safe overload factor is 150 [%], which corresponds to 16.95 [Nm]. This means that torques of up to this value do not damage the sensor. The specifications of the sensor are of course not guaranteed anymore in this range.

Torquesensor data	**Value**	**Unit**
Rated Output (R.O.) @ F.S.	2	[mV/V]
Nonlinearity	0.1	[% of R.O.]
Hysteresis	0.1	[% of R.O]
Nonrepeatability	0.05	[% of R.O.]
Zero Balance	1.0	[% of R.O.]
Compensated Temp. Range	15.5 to 71.1	[^{o}C]
Safe Temp. Range	-53.9 to 93.3	[^{o}C]
Temp. Effect on Output	0.00278	[% of Load/^{o}C]
Temp. Effect on Zero	0.00278	[% of R.O./^{o}C]
Terminal Resistance	350	[Ω]
DC Excitation Voltage	10	[V]
Safe Overload	150	[% of R.O.]

Table 7.3: Specifications of the Transducer Techniques TRT-100 reaction torque sensor.

Because the measurement principle of the sensor is a bridge configuration of strain gages it has good thermal stability since the temperature effects on the strain gages compensate for each other. This is one of the basic features of a bridge configuration and more information can be found in any good textbook on measurement techniques, for example [11].

7.1.2 Control

The control part of the design consists of two parts: a power stage that acts as an amplifier to drive the motor, and an intelligent part that uses the foot placement algorithm to calculate the required hip setpoint for a PID controller that controls the hip angle. Foot placement is chosen because from chapter 3.2 we found it to be very powerful to control our walker.

Amplifier implementation

Because we want the robot to be autonomous, a small amplifier is desired. Most commercial amplifiers are big boxes that cannot be used inside the robot so a small amplifier that can fit in a 60 [mm] diameter cylinder of length 150 [mm] has to be designed. The requirements of the amplifier can be found in table 7.4.

Amplifier requirement	Value	Unit
Input DC voltage	15 - 30	[V]
Continuous current capability	5	[A]
Peak current capability	10	[A]
Max. length	150	[mm]
Max. diameter	60	[mm]

Table 7.4: Motor amplifier requirements.

Although the motor is a 24V type it is desirable if the amplifier can also operate at higher or lower voltages, since a fully charged battery pack has a higher voltage than 24V and an almost empty battery pack has a much lower voltage than 24V. The current requirements are conservative choices and more than enough for the walker. A proof of concept design for an amplifier that satisfies this is presented in appendix A. It uses so called 'smart FETs' that have built in short circuit and thermal protection.

It is desirable that the torque sensor interface electronics and the motor amplifier are integrated on one PCB. Using a powerful micro controller the torque control and position control can be done locally.

Controller implementation

As was mentioned in the preceding section there are local control loops in the amplifier electronics that control position and torque of the motor. On a higher level there has to be a controller that calculates the foot placement algorithm and sends the desired hip setpoint to the lower level position controller. This higher level controller should also generate the desired torque setpoints for the torque controller.

This higher level controller will eventually be a PC/104 board with an Anything-I/O card added, located in or on the robot. For first experiments a normal PC can be used with a normal serial link to the electronics of the robot, or with a wireless connection if a fully free moving set-up is desired.

7.1.3 Sensing

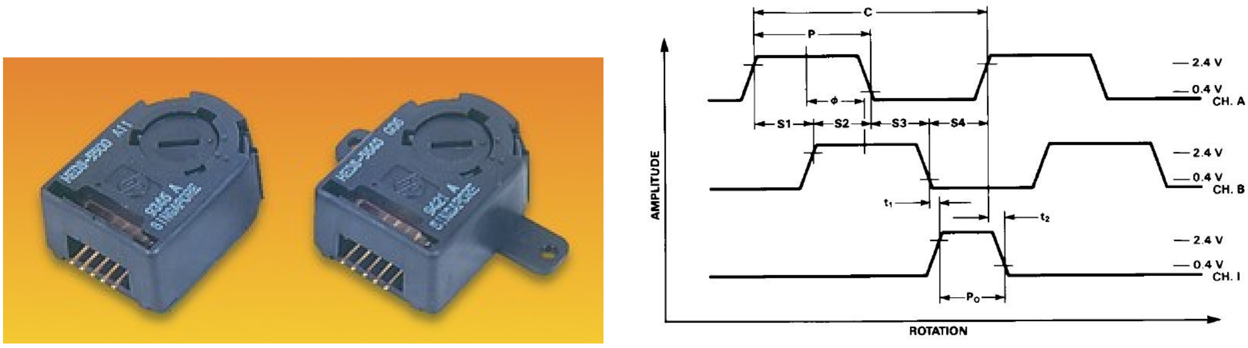
Angle measurements

All the angle measurements will be done with optical encoders. The are chosen because they:

- Are easy to interface since the signal is already digital;
- Have no rolling resistance because it is a contact free measurement;
- Are very accurate (up to 2000 PPR with a standard encoder, using 4X decoding this results in 8000 PPR);
- Are cheap.

A major disadvantage is that the output of an optical encoder is not an absolute value. By counting the output pulses of the encoder an angle can be measured, but it is relative to the angle that the encoder had when it was powered up. For this reason an encoder with a third channel has to be used. This third channel generates an index pulse when the encoder is at a certain angle. So this index pulse can be used to measure absolute angles. A slight disadvantage of this method is that during initialization the axle on which the encoder is mounted has to be moved until the index pulse is detected. For the unactuated knee and ankle joints this requires some extra effort from the operator of the robot.

The Agilent HEDS5540-A11 optical encoder will be used (see figure 7.3(a). This is a standard encoder with a 500 PPR resolution. Typical output waveforms of this encoder are shown in figure 7.3(b).



(a) Picture of the encoder

(b) HEDS5540 output waveforms

Figure 7.3: The Agilent HEDS5540 optical encoder.

Because four different output states can be separated in figure 7.3(b) it is possible to quadruple the resolution of the encoder. Each state corresponds to a quarter of the angular value of 1 slot of the code wheel in the encoder. For this 500 PPR encoder this means that it is possible to measure angles as small as $\frac{2\pi}{500*4} = 0.00314$ [rad]. Since this is smaller than the actuator backlash it is not useful to use an encoder that is even more accurate.

Impact measurements

To be able to determine the configuration of the walker it is necessary to know which leg is in contact with the ground. This can be done by placing a micro switch in the sole of each foot. The micro switch will be active when the foot is in contact with the ground.

7.2 Mechanical design proposal

Figures 6.1(a) and 6.1(b) already gave a preview on how the walker would look. In this section we will present the final design that tries to cope with the problems mentioned in chapter 6. A full view of the design is shown in figure 7.4.

In the following sections the important subparts of the design will be highlighted.

7.2.1 Hip design

A more detailed view can be found in figure 7.5.

The empty space will be filled with the batteries and the electronics. In the right side the actuator assembly is mounted, as was already explained in section 7.1.1.

The hip has to be (rotationally) very stiff because it has to cope with very big torques and because both legs in a pair should move the same. Since the distance between the legs of a pair is quite big this

Figure 7.4: Mechanical design of the 2D walker.

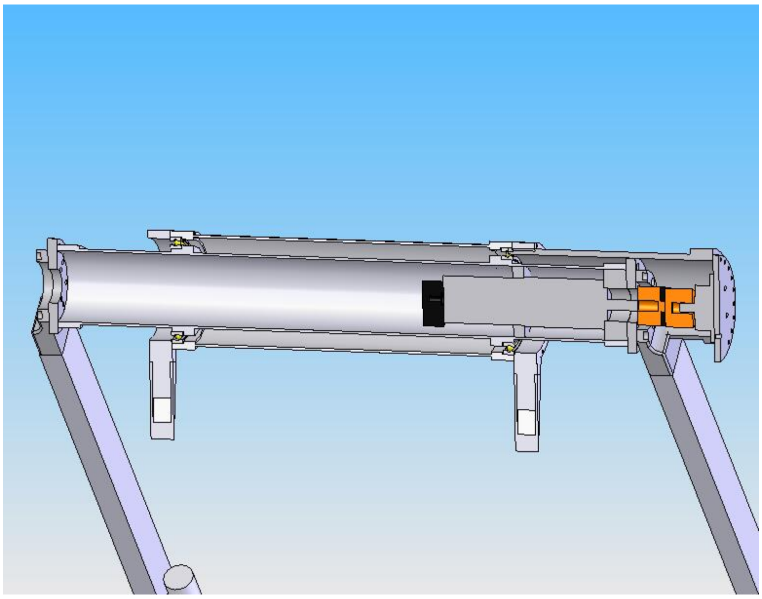


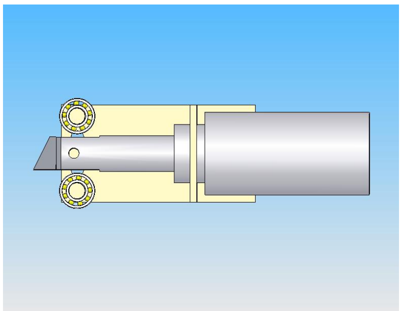
Figure 7.5: A detailed (cross-sectional) view of the proposed 2d autonomous walker.

can have a big influence. To assure a high stiffness a design was made consisting of two tubes since tubes have the highest stiffness to weight ratio of all structures.

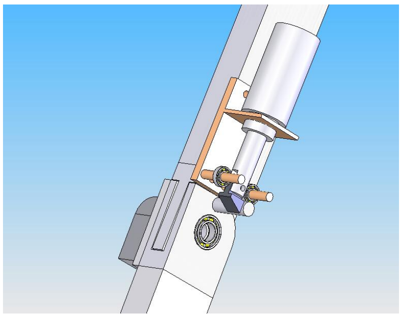
To assure low friction movement the two tubes are connected to each other with ball bearings. The bearing guidances are designed so that the bearings are not statically overdetermined.

The half tube on the right is used to transfer the power from the motor shaft to the outer tube. The outer tube has the two inner legs connected to it. The motor and gearbox housing is fixed to the inner tube so that the two pairs of legs move relative to each other.

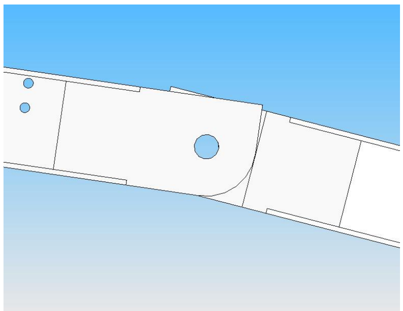
7.2.2 Knee and knee locking mechanism



(a) Side view of the knee locking mechanism



(b) The knee locking mechanism in place



(c) An inner view of the knee showing the mechanical end stop.

Figure 7.6: The knee locking mechanism.

In figure 7.6 the knee locking mechanism can be seen in detail. It is used to lock the knee to assure a fully stretched leg, which is necessary for foot placement when the leg has to be positioned to the right angle but also when the leg is in stance to make sure it does not collapse.

The mechanism consists of a solenoid that can retract a locking pin which holds the knee in place when the leg is stretched. Locking happens automatically: when the leg swings to become stretched, the pin on the lower leg will push the locking pin back, until the pin is lower than the locking pin. Then a spring (not shown in the figure) will push the locking pin outwards and the knee will be locked, as in figure 7.6b. The spring should be stiff enough to push the locking pin outwards, but it should be not too stiff because otherwise it can happen that the swinging leg cannot push the locking pin back.

The lower leg cannot move back again because of the locking pin, and it cannot move forward because the mechanical design of the knee does not allow angles beyond this point. This is shown clearly in figure 7.6(c).

The mechanism has two bearings that are used to guide the locking pin, which is flat at the top and bottom to prevent rotation of the locking pin. The bearings are necessary because the locking pin will be subject to large forces, which could make the friction forces so big that the solenoid cannot retract the pin. The same holds for the pin on the lower leg, it should also be able to rotate. To make sure that alignment of the wheels is not very critical a piece of rubber has to be glued on top and on the bottom of the locking pin. This will give it a bit compliance.

The knee has bearings (one of them can be seen in 7.6b) for low friction movement. In this figure also the optical encoder can be seen, which will measure the angle of the knee.

Some specifications of the solenoid can be found in table 7.5. This solenoid should only used intermittently, with a 10 % duty rating. This means that it should only be energized 10 % of an operation cycle. This is the case since it is only on for short periods compared to the time between two activations.

7.2.3 Feet

The feet are only for the measurement of the angle with respect to ground. They have to be small so that the foot to ground clearance is not affected. One foot can be seen in figure 7.7. Visible is the optical

Solenoid specification	**Value**	**Unit**
Operating voltage (DC)	24	[V]
Magnetic force @ 0 mm	35.6	[N]
Magnetic force @ rated stroke	1.2	[N]
Rated stroke	25	[mm]
Weight	161	[g]
Duty rating	10	[%]

Table 7.5: Emessem R16X16,24V15% solenoid specifications (intermittent version).

encoder that will measure the angle of the ankle. Not visible is a micro switch in the sole of the foot that will do the impact detection. Also not visible are the bearings which are located in the ankle.

If the grip on the floor is too low some pieces of friction material like rubber should be added on the sole of the foot.

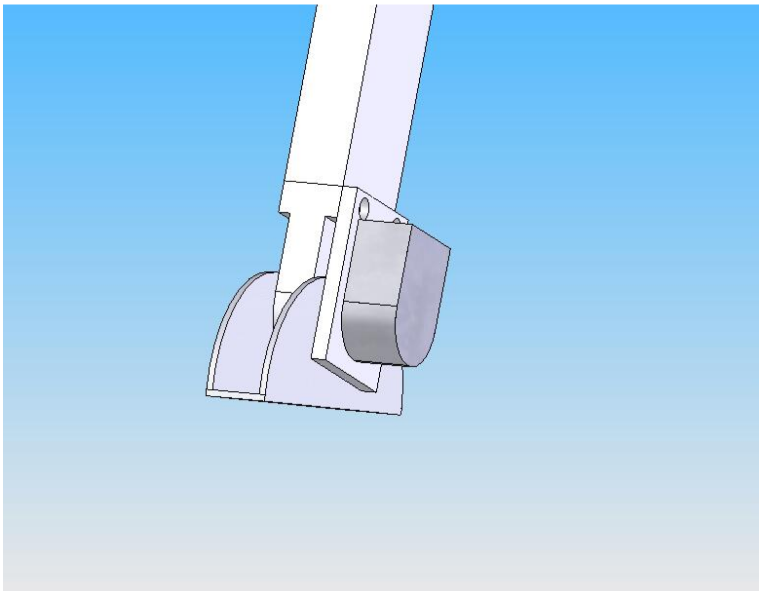


Figure 7.7: One of the feet of the walker.

7.2.4 Dimensions and estimated weights

See table 7.6 for the dimensions of the walker. The weights are estimated using the Mass Properties tool of Solid Works. The robot is constructed from aluminum.

Dimension of	**Length/Depth** [mm]	**Height/Diameter** [mm]	**Width** [mm]	**Weight** [kg]
Hip (without actuator)	-	100	500	2.374
Actuator (with mounting aids, without coupling/torque sensor)	-	60	180	0.94
Knee locking mechanism	35	40	100	0.229
Leg (fully streched from foot, to centre of hip connection)	1010	40	20	1.111
Total walker	-	1060	500	6.949
Total walker weight with batteries and actuator	-	-	-	**9.6**

Table 7.6: Dimensions of the walker

Chapter 8

Conclusion / Recommendations (Future work)

8.1 Conclusion

The goal of this project was to analyse and develop a 2D walker. During analysis some interesting control methods were modeled and simulated for a two legged walker. It was found that virtual model control can be a good control method to control the double support phase of a walker. Foot placement was investigated and an extended foot placement controller was proposed that makes it possible to control the speed of the walker. The foot placement method proved to be powerful enough to control a biped walker with only one actuator located in the hip and knee locking mechanisms.

A study was done to see if the walking efficiency could be increased. A non-linear spring was used to store the kinetic energy that otherwise would be converted into heat by the controller. Using this spring decreased both the maximum power applied by the actuator and the negative part of the actuator power.

By simulation we found requirements for the design of the 2D walker. With these results a mechanical and an electrical design was made. An actuator design was proposed that is: capable of generating the required torque, lightweight and relatively cheap. Using shaft torque feedback it is "‘fully backdriveable"’ and can emulate any given behaviour, which makes it possible to test the effect of a non-linear spring without actually building the spring. A lot of effort was put in the mechanical design of the hip of the walker. A hip design was proposed that is both stiff and lightweight. A control system was designed to use the foot placement method.

The design proposed will be used to actually build the walker. Edwin Dertien and Gijs van Oort will continue this work.

8.2 Recommendations

Because we were only able to propose a design and not finalize building it the most important recommendation is of course to finalize the building of the design. This will be done by Edwin Dertien and Gijs van Oort.

Further recommendations are:

- Virtual model control was studied in this project but was found to be difficult to implement because of the impact models. Both feet have to be in contact with the ground but often one of the feet comes loose from the ground. With only one foot on the ground it is not possible to create the virtual model control forces. More study is needed to overcome this simulation problem.

- During simulation it was found that the foot to ground clearance is quite low. This is a drawback of unactuated knees. It can be useful to have small actuators in the knees that can help improve foot to ground clearance. These do not have to be powerful since they only have to retract the lower leg a little bit.

- The principles of foot placement can also be used to stabilize the sideways motion of a 3D walker. A controller can be designed that uses this foot placement method to calculate the correct sideways angle that the leg of a 3D walker has to have so that all sideways kinetic energy will be converted into potential energy.
- Using a non-linear spring appeared to increase efficiency. A physical design has to be made that can implement this spring.
- From literature (for example [13]) it appears that curved feet increase the stability of the walker. A design could be made that uses this fact and can still measure the angle of the ankle.
- It can be useful to have an upper body to increase stability of the walker. Using a bisecting hip mechanism (see [13]) this body will always be in the middle of the two legs.
- Using the results obtained of building this walker a new design could be made of a 3D autonomous walker.

Bibliography

[1] Vincent Duindam and Stefano Stramigioli. Modeling the kinematics and dynamics of compliant contact. *International Conference on Robotics and Automation*, pages 4029 – 4034, September 2003.

[2] Eric Hale, Nathan Schara, Joel Burdick, and Paolo Fiorini.

[3] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda robot. *Proceedings IEEE International Conference on Robotics and Automation*, pages 1321–1326, May 1998.

[4] Glenn K. Klute, Joseph M. Czerniecki, and Blake Hannaford. Mckibben artificial muscles: pneumatic actuators with biomechanical intelligence. *Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 221–226, September 1999.

[5] Tad McGeer. Passive walking with knees. *Proceedings of 1990 IEEE International Conference on Robotics and Automation*, 3:1640–1645, May 1990.

[6] Chandana Paul, Hiroshi Yokoi, and Kojiro Matsushita. Design and control of humanoid robot locomotion with passive legs and upper body actuation. *International Symposium on Robotics*, 2003.

[7] Gill A. Pratt and Matthem M. Williamson. Series elastic actuators. *Proceedings of IROS '95*, 1995.

[8] Jerry Pratt, Peter Dilworth, and Gill Pratt. Virtual model control of a bipedal walking robot. *Proceedings of the 1997 International Conference on Robotics and Automation*, 1997.

[9] Jerry E. Pratt and Gill E. Pratt. Exploiting natural dynamics in the control of a 3d bipedal walking simulation. *International Conference on Climbing and Walking Robots*, September 1999.

[10] Control Lab Products. 20sim version 3.4. http://www.20sim.com/, 2004.

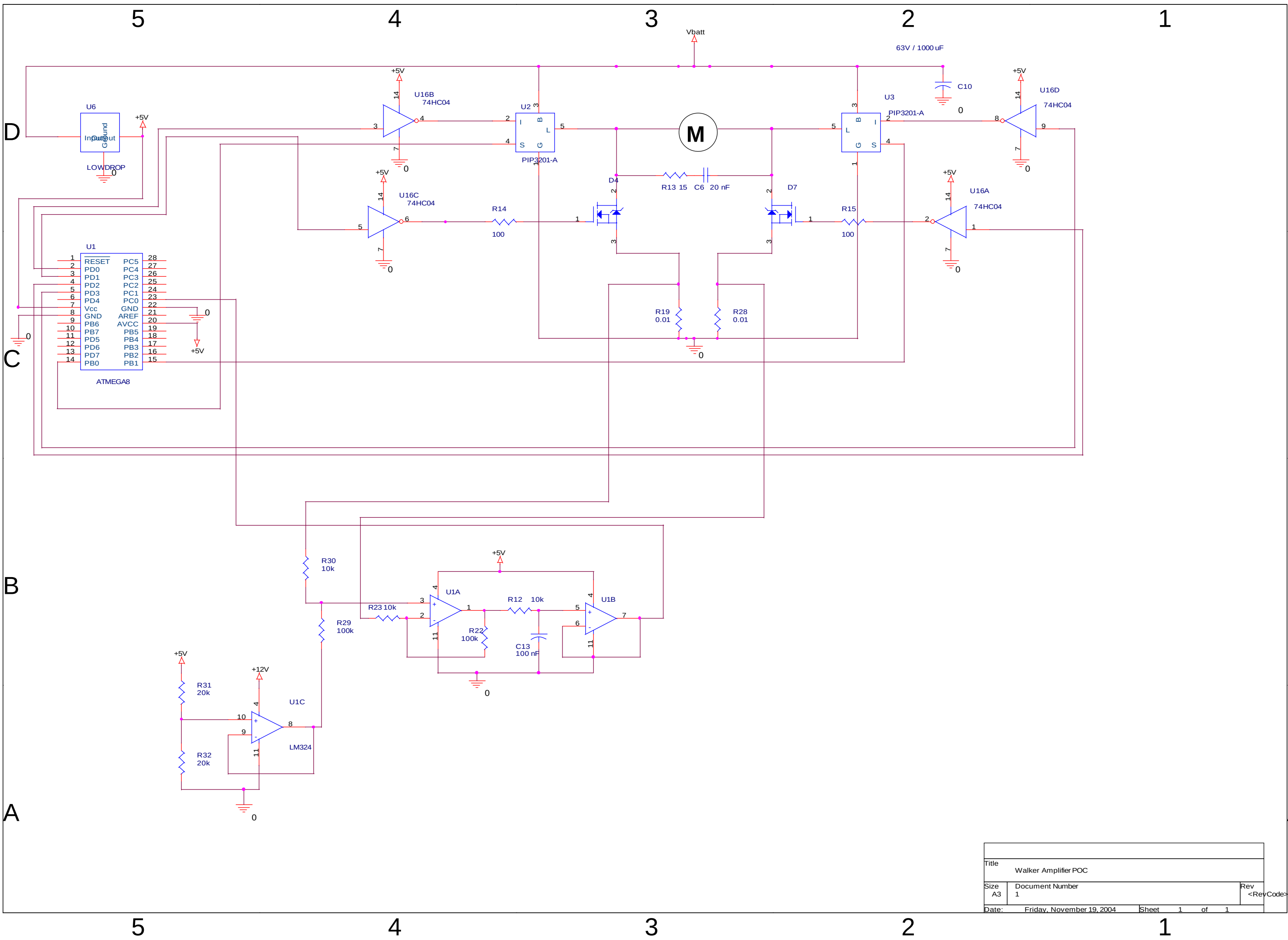
[11] P.P.L. Regtien. *Sensors for mechatronic systems course book*. Laboratory for measurement and instrumentation, University of Twente, Enschede, the Netherlands, 2003.

[12] Stefano Stramigioli and Herman Bruyninckx. Geometry and screw theory for robotics. Tutorial during ICRA 2001, 2001.

[13] M. Wisse and J. van Frankenhuyzen. Design and construction of mike: A 2d autonomous biped based on passive dynamic walking. *International Conference on Adaptive Motion of Animals and Machines*, 2003.

Appendix A

Electronic designs

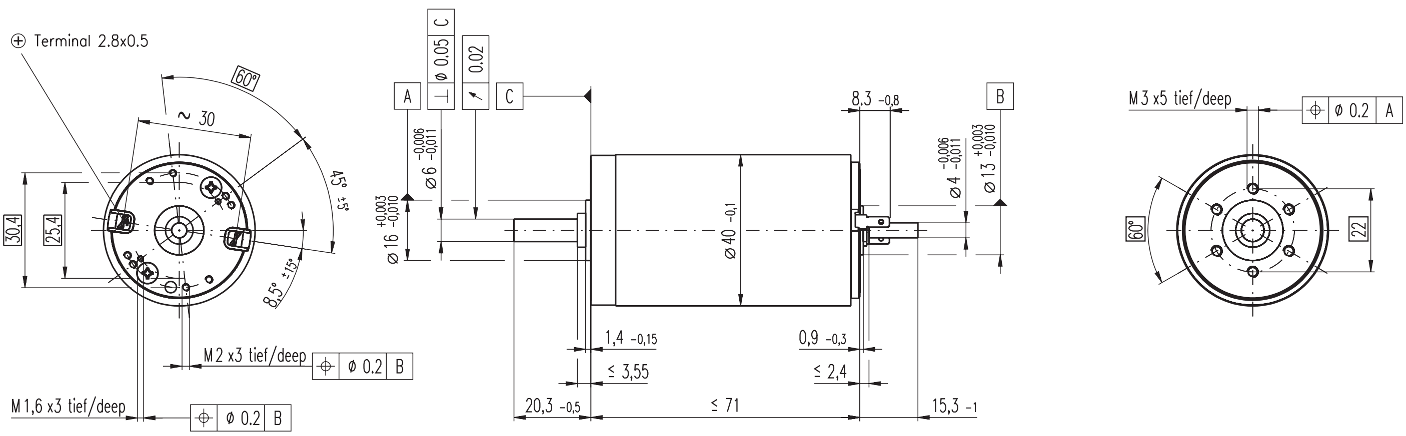


Vbatt
63V / 1000 uF
C10
U6
Input
Output
Ground
LOWDROP
+5V
U16B
74HC04
U16C
74HC04
U16D
74HC04
U16A
74HC04
U2
PIP3201-A
U3
PIP3201-A
M
D4
D7
R13 15
C6 20 nF
R14
100
R15
100
R19
0.01
R28
0.01
U1
RESET
PD0
PD1
PD2
PD3
PD4
Vcc
GND
PB6
PB7
PD5
PD6
PD7
PB0
PC5
PC4
PC3
PC2
PC1
PC0
GND
AREF
AVCC
PB5
PB4
PB3
PB2
PB1
ATMEGA8
R30
10k
R29
100k
R23 10k
U1A
R12 10k
R22
100k
C13
100 nF
U1B
+12V
U1C
LM324
R31
20k
R32
20k
Title
Walker Amplifier POC
Size
A3
Document Number
1
Rev
<RevCode>
Date:
Friday, November 19, 2004
Sheet 1 of 1

Appendix B

Datasheets

RE 40 ∅40 mm, Graphite Brushes, 150 Watt



M 1:2

Stock program
Standard program
Special program (on request!)

Order Number

		148866	148867	148877	218008	218009	218010	218011	218012	218013	218014	218015
Industrial version		263065	263066	263067	263068	263069	263070	263071	263072	263073	263074	263075
Motor Data												
1 Assigned power rating	W	150	150	150	150	150	150	150	150	150	150	150
2 Nominal voltage	Volt	12.0	24.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0
3 No load speed	rpm	6920	7580	7580	6420	5560	3330	2690	2130	1710	1420	987
4 Stall torque	mNm	1690	2290	2500	1990	1580	996	796	641	512	415	289
5 Speed / torque gradient	rpm / mNm	4.11	3.32	3.04	3.23	3.53	3.36	3.39	3.35	3.37	3.44	3.45
6 No load current	mA	241	137	69	54	44	22	17	13	10	8	5
7 Starting current	A	103	75.9	41.4	28.0	19.2	7.26	4.69	3.00	1.92	1.29	0.627
8 Terminal resistance	Ohm	0.117	0.316	1.16	1.72	2.50	6.61	10.2	16.0	24.9	37.1	76.6
9 Max. permissible speed	rpm	8200	8200	8200	8200	8200	8200	8200	8200	8200	8200	8200
10 Max. continuous current	A	6.00	6.00	3.33	2.75	2.41	1.41	1.13	0.904	0.725	0.594	0.414
11 Max. continuous torque	mNm	98.7	181	201	196	198	193	192	193	193	191	190
12 Max. power output at nominal voltage	W	285	440	491	332	255	86.5	55.7	35.6	22.9	15.3	7.40
13 Max. efficiency	%	88	91	92	91	91	89	88	87	86	85	83
14 Torque constant	mNm / A	16.4	30.2	60.3	71.3	82.2	137	170	214	266	321	461
15 Speed constant	rpm / V	581	317	158	134	116	69.7	56.2	44.7	35.9	29.8	20.7
16 Mechanical time constant	ms	6	5	4	4	4	4	4	4	4	4	4
17 Rotor inertia	gcm²	135	134	134	125	127	118	117	118	117	114	114
18 Terminal inductance	mH	0.02	0.08	0.33	0.46	0.61	1.70	2.62	4.14	6.40	9.31	19.20
19 Thermal resistance housing-ambient	K / W	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7
20 Thermal resistance rotor-housing	K / W	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9
21 Thermal time constant winding	s	41	40	40	38	38	36	35	35	35	34	34

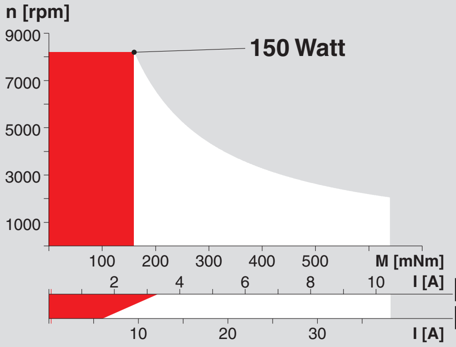
Specifications

- Axial play 0.05 - 0.15 mm
- Max. **ball bearing** loads
  axial (dynamic)
  not preloaded 5.6 N
  preloaded 2.4 N
  radial (5 mm from flange) 28 N
  Force for press fits (static) 110 N
  (static, shaft supported) 1200 N
- Radial play **ball bearing** 0.025 mm
- Ambient temperature range -20 ... +100°C
- Max. rotor temperature +155°C
- Number of commutator segments 13
- Weight of motor 480 g
- 2 pole permanent magnet
- Values listed in the table are nominal.
  For applicable tolerances see page 43.
  For additional details please use the maxon selection program on the enclosed CD-Rom.

Operating Range

Comments

Details on page 49



Recommended operating range

Continuous operation
In observation of above listed thermal resistances (lines 19 and 20) the maximum permissible rotor temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

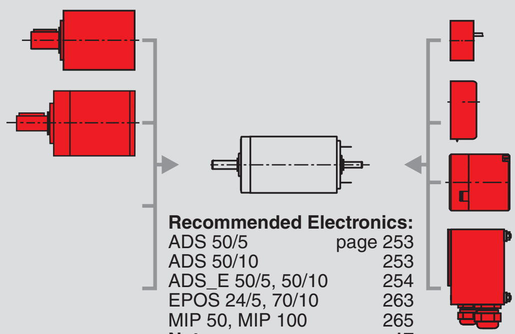
148877 Motor with high resistance winding
148866 Motor with low resistance winding

maxon Modular System

Overview on page 17 - 21

Planetary Gearhead
∅42 mm
3 - 15 Nm
Details page 218

Planetary Gearhead
∅52 mm
4 - 30 Nm
Details page 221



Recommended Electronics:

ADS 50/5	page 253
ADS 50/10	253
ADS_E 50/5, 50/10	254
EPOS 24/5, 70/10	263
MIP 50, MIP 100	265
Notes	**17**

Digital MR Encoder
256 - 1024 CPT,
3 channels
Details page 233

Digital Encoder HP HED_ 55__
500 CPT, 3 channels
Details page 236 / 238

Brake
∅40 mm,
24 VDC, 0.4 Nm
Details page 269

Industrial version Encoder
Details page 241

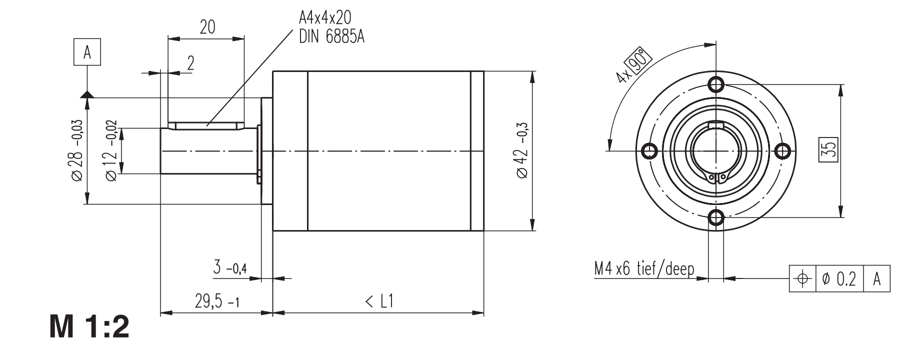
Brake
Details page 270

July 2004 edition / subject to change

Planetary Gearhead GP 42 C ∅42 mm, 3 - 15 Nm

Ceramic Version

maxon gear



M 1:2

Technical Data

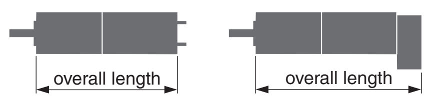
Planetary Gearhead	straight teeth
Output shaft	stainless steel
Bearing at output	ball bearing
Radial play, 12 mm from flange	preloaded
Axial play	preloaded
Max. permissible axial load	150 N
Max. permissible force for press fits	300 N
Sense of rotation, drive to output	=
Recommended input speed	< 8000 rpm
Recommended temperature range	-20 ... +100°C

Number of stages	1	2	3	4
Max. radial load 12 mm from flange	120 N	150 N	150 N	150 N

Stock program
Standard program
Special program (on request!)

Gearhead Data

			Order Number						
			203113	203115	203120	203125	203128	203134	203139
1	Reduction		3.5 : 1	12 : 1	43 : 1	91 : 1	150 : 1	319 : 1	546 : 1
2	Reduction absolute		$^{7}/_{2}$	$^{49}/_{4}$	$^{343}/_{8}$	91	$^{2401}/_{16}$	$^{637}/_{2}$	546
3	Mass inertia	gcm²	14	15	15	15	15	15	14
4	Max. motor shaft diameter	mm	10	10	10	10	10	10	10
	Order Number		203114	203116	203121	203126	203130	203135	203140
1	Reduction		4.3 : 1	15 : 1	53 : 1	113 : 1	186 : 1	353 :1	676 : 1
2	Reduction absolute		$^{13}/_{3}$	$^{91}/_{6}$	$^{637}/_{12}$	$^{338}/_{3}$	$^{4459}/_{24}$	$^{28561}/_{81}$	676
3	Mass inertia	gcm²	9.1	15	15	9.4	15	9.4	9.1
4	Max. motor shaft diameter	mm	8	10	10	8	10	8	8
	Order Number			203117	203122	203127	203131	203136	203141
1	Reduction			19 : 1	66 : 1	126 : 1	230 : 1	394 : 1	756 : 1
2	Reduction absolute			$^{169}/_{9}$	$^{1183}/_{18}$	126	$^{8281}/_{36}$	$^{1183}/_{3}$	756
3	Mass inertia	gcm²		9.4	15	14	15	15	14
4	Max. motor shaft diameter	mm		8	10	10	10	10	10
	Order Number			203118	203123	203129	203132	203137	203142
1	Reduction			21 : 1	74 : 1	156 : 1	257 : 1	441 : 1	936 : 1
2	Reduction absolute			21	$^{147}/_{2}$	156	$^{1029}/_{4}$	441	936
3	Mass inertia	gcm²		14	15	9.1	15	14	9.1
4	Max. motor shaft diameter	mm		10	10	8	10	10	8
	Order Number			203119	203124		203133	203138	
1	Reduction			26 : 1	81 : 1		285 : 1	488 : 1	
2	Reduction absolute			26	$^{2197}/_{27}$		$^{15379}/_{54}$	$^{4394}/_{9}$	
3	Mass inertia	gcm²		9.1	9.4		15	9.4	
4	Max. motor shaft diameter	mm		8	8		10	8	
5	Number of stages		1	2	3	3	4	4	4
6	Max. continuous torque	Nm	3.0	7.5	15	15	15	15	15
7	Intermittently permissible torque	Nm	4.5	11.3	22.5	22.5	22.5	22.5	22.5
8	Max. efficiency	%	90	81	72	72	64	64	64
9	Weight	g	260	360	460	460	560	560	560
10	Average backlash no load	°	0.3	0.4	0.5	0.5	0.5	0.5	0.5
11	Gearhead length L1	mm	40.9	55.4	69.9	69.9	84.4	84.4	84.4



Combination

+ Motor	Page	+ Tacho / Encoder	Page	+ Brake	Page	**Overall length [mm]** = Motor length + gearhead length + (tacho / encoder / brake) + assembly parts						
RE 35, 90 W	80					111.9	126.4	140.9	140.9	155.4	155.4	155.4
RE 35, 90 W	80	MR Encoder	233			123.3	137.8	152.3	152.3	166.8	166.8	166.8
RE 35, 90 W	80	Digital Encoder HED_ 55__	236/238			132.9	147.4	161.9	161.9	176.4	176.4	176.4
RE 35, 90 W	80	DC Tacho 22	246			130.0	144.5	159.0	159.0	173.5	173.5	173.5
RE 35, 90 W	80			Brake 40	269	148.0	162.5	177.0	177.0	191.5	191.5	191.5
RE 36, 70 W	81					112.2	126.7	141.2	141.2	155.7	155.7	155.7
RE 36, 70 W	81	MR Encoder	233			123.6	138.1	152.6	152.6	167.1	167.1	167.1
RE 36, 70 W	81	Digital Encoder HED_ 55__	236/238			133.2	147.7	162.2	162.2	176.7	176.7	176.7
RE 36, 70 W	81	DC Tacho 22	246			130.3	144.8	159.3	159.3	173.8	173.8	173.8
RE 40, 150 W	82					112.0	126.5	141.0	141.0	155.5	155.5	155.5
RE 40, 150 W	82	MR Encoder	233			123.4	137.9	152.4	152.4	166.9	166.9	166.9
RE 40, 150 W	82	Digital Encoder HED_ 55__	236/238			132.7	147.2	161.7	161.7	176.2	176.2	176.2
RE 40, 150 W	82	Digital Encoder HEDL 9140	241			166.1	180.6	195.1	195.1	209.6	209.6	209.6
RE 40, 150 W	82			Brake 40	269	148.1	162.6	177.1	177.1	191.6	191.6	191.6
RE 40, 150 W	82			Brake 28	270	156.1	170.6	185.1	185.1	199.6	199.6	199.6
RE 40, 150 W	82	Digital Encoder HED_ 55__	236/238	Brake 40	269	165.2	179.7	194.2	194.2	208.7	208.7	208.7
RE 40, 150 W	82	Digital Encoder HEDL 9140	241	Brake 28	270	176.6	191.1	205.6	205.6	220.1	220.1	220.1
EC 40, 120 W	159					111.0	125.5	140.0	140.0	154.5	154.5	154.5
EC 40, 120 W	159	Digital Encoder HED_ 55__	237/239			129.4	143.9	158.4	158.4	172.9	172.9	172.9
EC 40, 120 W	159	Resolver 26	247			137.6	152.1	166.6	166.6	181.1	181.1	181.1
EC 40, 120 W	159			Brake 40	269	141.8	156.3	170.8	170.8	185.3	185.3	185.3
EC 45, 150 W	160					152.2	166.7	181.2	181.2	195.7	195.7	195.7
EC 45, 150 W	160	Digital Encoder HEDL 9140	241			167.8	182.3	196.8	196.8	211.3	211.3	211.3
EC 45, 150 W	160	Resolver 26	247			152.2	166.7	181.2	181.2	195.7	195.7	195.7
EC 45, 150 W	160			Brake 28	270	159.6	174.1	188.6	188.6	203.1	203.1	203.1
EC 45, 150 W	160	Digital Encoder HEDL 9140	241	Brake 28	270	176.6	191.1	205.6	205.6	220.1	220.1	220.1

July 2004 edition / subject to change

710 - 8084

emessem
SOLENOID COMPANY LTD
ELECTROMAGNETIC EQUIPMENT

QUALITY SINCE 1912

D.C. Cylindrical Solenoid

Robust enclosed coil design with nose mounting

Rectifier for A.C. supply
Stroke up to 25mm
Pushing version available (RP)

10
Product group

Type R16 x 16 (RP16 x 16)

- Long stroke design with integral clevis end (fig. 1)
- Pull operation (version with threaded pushrod available)
- Increasing force characteristic (fig. 2)
- Threaded mounting nose with anti-rotation feature, locknut and shakeproof washer
- Coil with insulation to class B, for voltages up to 250 volts
- Protection classification - DIN VDE0470 / EN60529
  Flying leads - IP00
- UL rated materials of construction
- Zinc / nickel plated iron parts
- Suitable for operation in any attitude
- Modifications and special designs on request
- Increased protection solenoid for arduous service on:
  Machine tools, Office machines, Motor vehicles, Remote control, Automation, Medical equipment, Packaging and coin equipment, Textile machinery

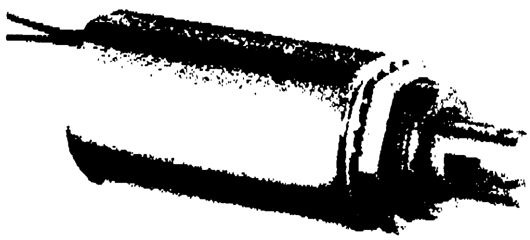


Fig. 1
R16 x16

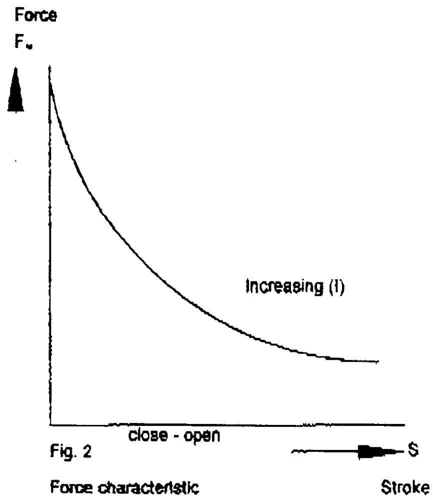


Fig. 2
Force characteristic

emessem
A MEMBER OF THE MAGNET-SCHULTZ GROUP OF COMPANIES
3 & 4 Capital Park
High Street, Old Woking
Surrey GU22 9LD.
Phone
01483 794700 Main
01483 794701 Sales
Fax 01483 757298
E-mail
sales@emessem-solenoid.co.uk

emessem
London

Product group 10

Performance and dimensional data for type R16 x 16 and RP16 x 16

		R16 x 16 - Pull			RP16 x 16 - Push		
Duty Rating		Continuous	Intermittent	Impulse	Continuous	Intermittent	Impulse
Stroke s	(mm)	Magnetic force F_M (N)					
	0	25.3	35.6	42.3	25.2	39.0	46.4
	2	6.9	19.2	28.6	5.6	14.2	30.3
	4	3.8	13.7	23.1	2.7	7.3	21.8
	8	1.8	8.3	17.0	0.8	2.8	13.3
	12	1.1	5.6	12.9	0.2	1.8	8.3
	16	0.76	4.0	10.0			
	20	0.45	2.6	7.4			
	25	0.11	1.2	3.5			
Power Consumption P_{25}	(Watts)	5.5	38	96	5.5	38	96
Armature Weight m_A	(kg)	0.043			0.038		
Solenoid Weight m_M	(kg)	0.161					

0mm = completion of energised stroke.

TABLE BASIS
24V / Continuous - Intermittent - Impulse duty
Mounted on steel plate 152 x 152 x 3mm
Horizontal working
Tolerance +/- 10% (inherent and manufacture)

Ambient temperature 25°C
Free air mounted
Pull arrangement

Conversion Factors
1 N = 0.102 kp ≈ 0.1 kp
1 Ncm = 0.102 kpcm ≈ 0.1 kpcm
1 kp = 1 kg = 2.2 lbs
1 mm = 0.039 in.
1 cmkp = 0.856 in. lbs

MAGNETIC FORCE (F_M)
is listed in HOT condition at RATED voltage
Adjust for armature weight

POWER CONSUMPTION (P_{25})
is listed with a 25°C coil temperature (decrease / HOT)

DUTY RATING
The proportion of time that the solenoid is energised per operation cycle, normally shown in %.

$$\text{Proportion (\%)} = \frac{t_{on}}{t_{on} + t_{off}} x100$$

For each coil type: maximum energised (proportion) time/cycle -
Continuous: (100%) Intermittent: (10%) 60secs. Impulse: (5%) 0.1secs.

Order Example

Type	R16 x 16
Voltage	24v DC
Duty rating	Continuous

SUPPLY VOLTAGE
Standard DC: 24V, 97V, 205V (for AC 50 / 60Hz rectified, 110, 230V)
Rectifier can be provided for AC supply.

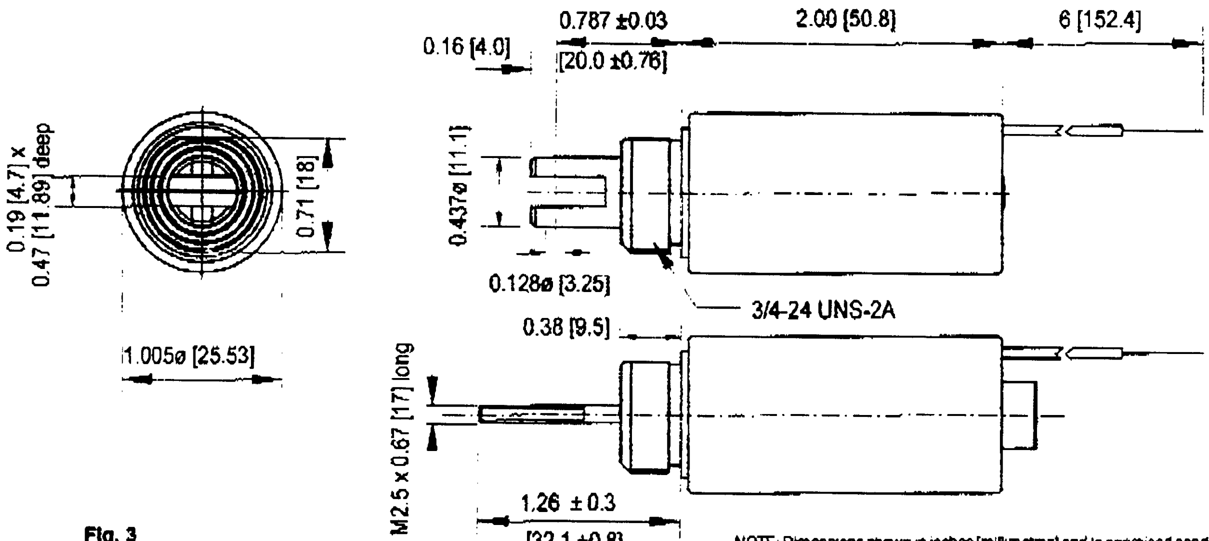


Fig. 3
R16 x 16 and RP16 x 16

NOTE: Dimensions shown in inches [millimetres] and in energised condition.

Quick Assembly Two and Three Channel Optical Encoders

Technical Data

HEDM-550X/560X
HEDS-550X/554X
HEDS-560X/564X

Features

- **Two Channel Quadrature Output with Optional Index Pulse**
- **Quick and Easy Assembly**
- **No Signal Adjustment Required**
- **External Mounting Ears Available**
- **Low Cost**
- **Resolutions Up to 1024 Counts Per Revolution**
- **Small Size**
- **-40°C to 100°C Operating Temperature**
- **TTL Compatible**
- **Single 5 V Supply**

Description

The HEDS-5500/5540, HEDS-5600/5640, and HEDM-5500/5600 are high performance, low cost, two and three channel optical incremental encoders. These encoders emphasize high reliability, high resolution, and easy assembly.

Each encoder contains a lensed LED source, an integrated circuit with detectors and output circuitry, and a codewheel which rotates between the emitter and detector IC. The outputs of the HEDS-5500/5600 and HEDM-5500/ 5600 are two square waves in quadrature. The HEDS-5540 and 5640 also have a third channel index output in addition to the two channel quadrature. This index output is a 90 electrical degree, high true index pulse which is generated once for each full rotation of the codewheel.

The HEDS series utilizes metal codewheels, while the HEDM series utilizes a film codewheel allowing for resolutions to 1024 CPR. The HEDM series is nont available with a third channel index.

These encoders may be quickly and easily mounted to a motor. For larger diameter motors, the HEDM-5600, and HEDS-5600/5640 feature external mounting ears.

The quadrature signals and the index pulse are accessed through five 0.025 inch square pins located on 0.1 inch centers.

Standard resolutions between 96 and 1024 counts per revolution are presently available. Consult local Hewlett-Packard sales representatives for other resolutions.

Applications

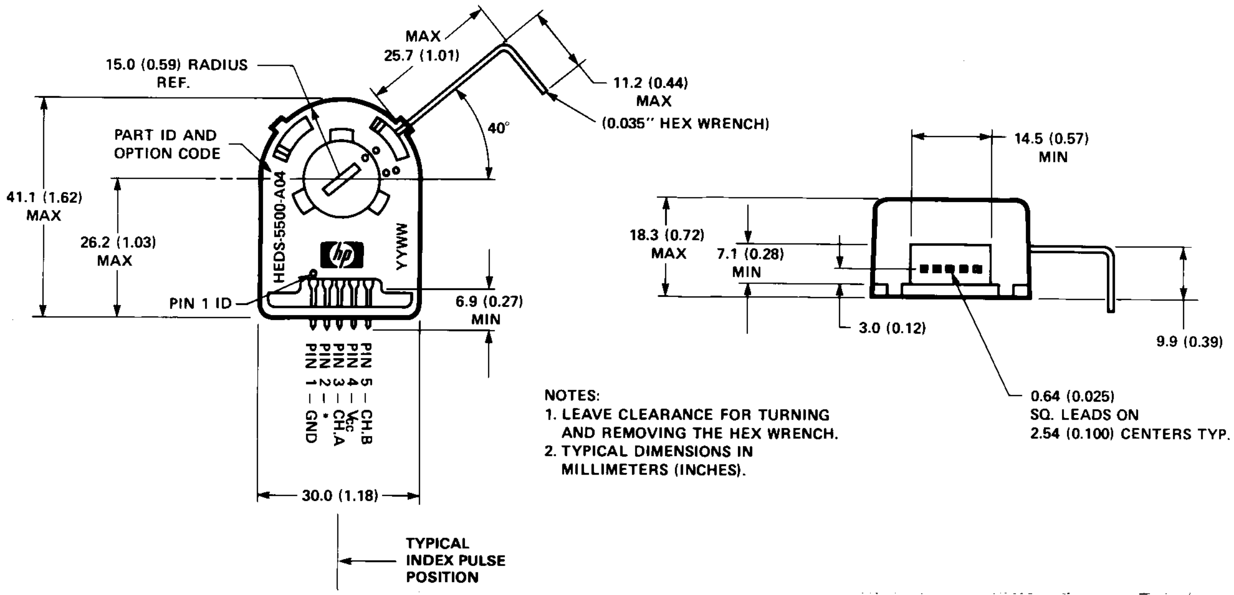
The HEDS-5500, 5540, 5600, 5640, and the HEDM-5500, 5600 provide motion detection at a low cost, making them ideal for high volume applications. Typical applications include printers, plotters, tape drives, positioning tables, and automatic handlers.

ESD WARNING: NORMAL HANDLING PRECAUTIONS SHOULD BE TAKEN TO AVOID STATIC DISCHARGE.

5965-5875E

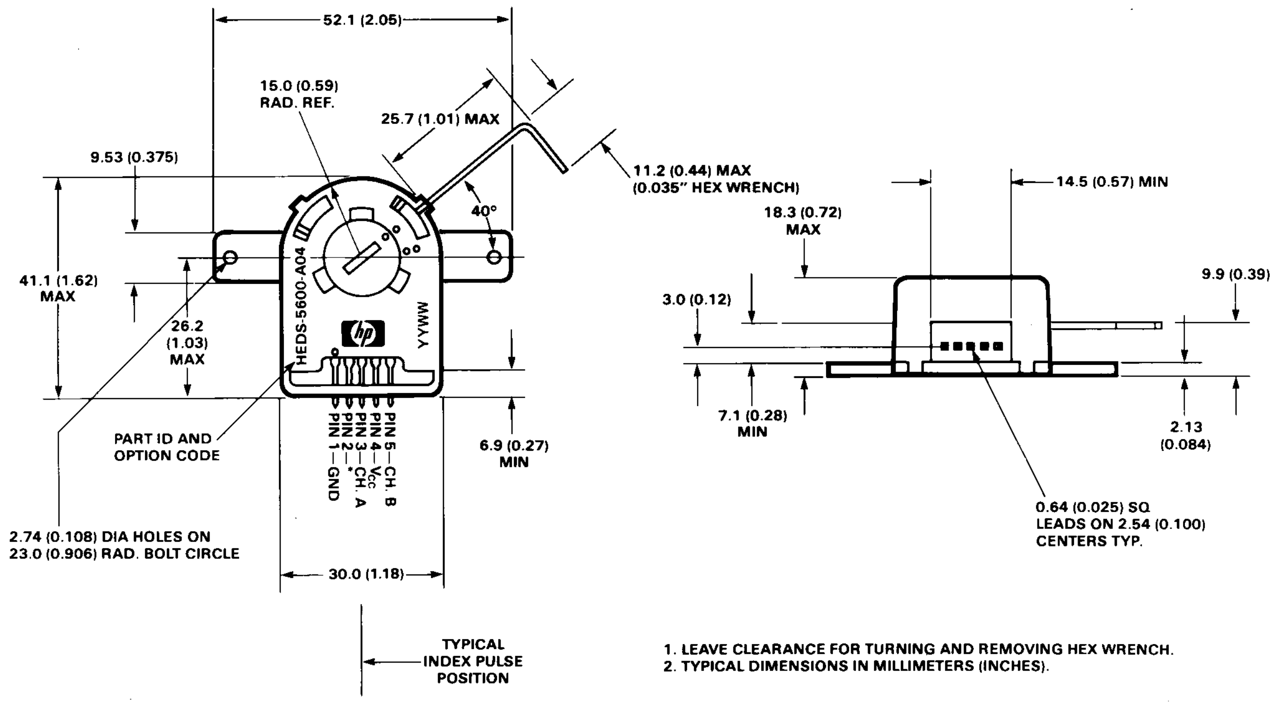
Package Dimensions

HEDS-5500/5540, HEDM-5500



***Note:** For the HEDS-5500 and HEDM-5500, Pin #2 is a No Connect. For the HEDS-5540, Pin #2 is CH. I, the index output.

HEDS-5600/5640, HEDM-5600



***Note:** For the HEDS-5600 and HEDM-5600, Pin #2 is a No Connect. For the HEDS-5640, Pin #2 is CH. I, the index output.

Theory of Operation

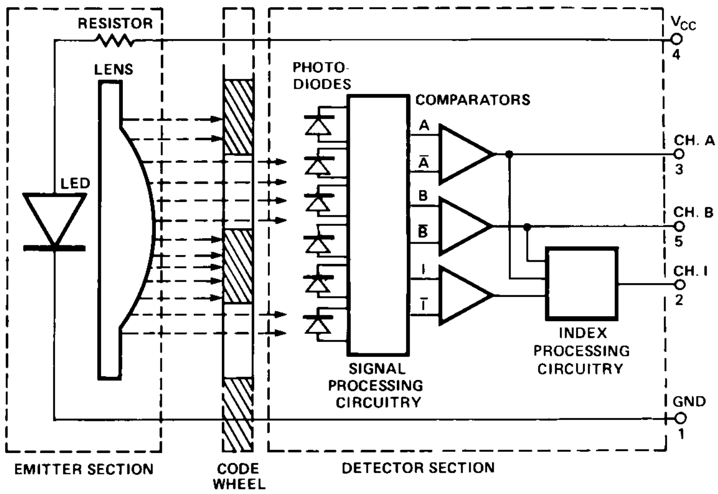
The HEDS-5500, 5540, 5600, 5640, and HEDM-5500, 5600 translate the rotary motion of a shaft into either a two- or a three-channel digital output.

As seen in the block diagram, these encoders contain a single Light Emitting Diode (LED) as its light source. The light is collimated into a parallel beam by means of a single polycarbonate lens located directly over the LED. Opposite the emitter is the integrated detector circuit. This IC consists of multiple sets of photodetectors and the signal processing circuitry necessary to produce the digital waveforms.

The codewheel rotates between the emitter and detector, causing the light beam to be interrupted by the pattern of spaces and bars on the codewheel. The photodiodes which detect these interruptions are arranged in a pattern that corresponds to the radius and design of the codewheel. These detectors are also spaced such that a light period on one pair of detectors corresponds to a dark period on the adjacent pair of detectors. The photodiode outputs are then fed through the signal processing circuitry resulting in A, $\overline{A}$, B and $\overline{B}$ (also I and $\overline{I}$ in the HEDS-5540 and 5640). Comparators receive these signals and produce the final outputs for channels A and B. Due to this integrated phasing technique, the digital output of channel A is in quadrature with that of channel B (90 degrees out of phase).

In the HEDS-5540 and 5640, the output of the comparator for I and $\overline{I}$ is sent to the index processing circuitry along with the outputs of channels A and B.

Block Diagram



NOTE: CIRCUITRY FOR CH. I IS ONLY IN HEDS-5540 AND 5640 THREE CHANNEL ENCODERS.

The final output of channel I is an index pulse P_O which is generated once for each full rotation of the codewheel. This output P_O is a one state width (nominally 90 electrical degrees), high true index pulse which is coincident with the low states of channels A and B.

Definitions

Count (N): The number of bar and window pairs or counts per revolution (CPR) of the codewheel.

One Cycle (C): 360 electrical degrees (°e), 1 bar and window pair.

One Shaft Rotation: 360 mechanical degrees, N cycles.

Position Error (ΔΘ): The normalized angular difference between the actual shaft position and the position indicated by the encoder cycle count.

Cycle Error (ΔC): An indication of cycle uniformity. The difference between an observed shaft angle which gives rise to one electrical cycle, and the nominal angular increment of 1/N of a revolution.

Pulse Width (P): The number of electrical degrees that an output is high during 1 cycle. This value is nominally 180°e or 1/2 cycle.

Pulse Width Error (ΔP): The deviation, in electrical degrees, of the pulse width from its ideal value of 180°e.

State Width (S): The number of electrical degrees between a transition in the output of channel A and the neighboring transition in the output of channel B. There are 4 states per cycle, each nominally 90°e.

State Width Error (ΔS): The deviation, in electrical degrees, of each state width from its ideal value of 90°e.

Phase (ϕ): The number of electrical degrees between the center of the high state of channel A and the center of the high state of channel B. This value is nominally 90°e for quadrature output.

Phase Error (Δϕ): The deviation of the phase from its ideal value of 90°e.

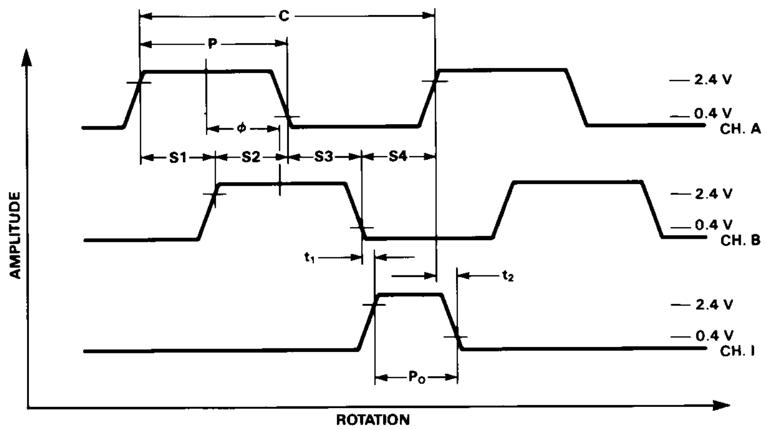
Absolute Maximum Ratings

Parameter	HEDS-55XX/56XX	HEDM-550X/560X
Storage Temperature, T_S	-40°C to 100°C	-40°C to +70°C
Operating Temperature, T_A	-40°C to 100°C	-40°C to +70°C
Supply Voltage, V_{CC}	-0.5 V to 7 V	-0.5 V to 7 V
Output Voltage, V_O	-0.5 V to V_{CC}	-0.5 V to V_{CC}
Output Current per Channel, I_{OUT}	-1.0 mA to 5 mA	-1.0 mA to 5 mA
Vibration	20 g, 5 to 1000 Hz	20 g, 5 to 1000 Hz
Shaft Axial Play	± 0.25 mm (± 0.010 in.)	± 0.175 mm (± 0.007 in.)
Shaft Eccentricity Plus Radial Play	0.1 mm (0.004 in.) TIR	0.04 mm (0.0015 in.) TIR
Velocity	30,000 RPM	30,000 RPM
Acceleration	250,000 rad/sec^2	250,000 rad/sec^2

Direction of Rotation: When the codewheel rotates in the counter-clockwise direction (as viewed from the encoder end of the motor), channel A will lead channel B. If the codewheel rotates in the clockwise direction, channel B will lead channel A.

Index Pulse Width (P_O): The number of electrical degrees that an index output is high during one full shaft rotation. This value is nominally 90°e or 1/4 cycle.

Output Waveforms



Recommended Operating Conditions

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Temperature HEDS Series	T_A	-40		100	°C	
Temperature HEDM Series	T_A	-40		70	°C	non-condensing atmosphere
Supply Voltage	V_{CC}	4.5	5.0	5.5	Volts	Ripple < 100 mV_{p-p}
Load Capacitance	C_L			100	pF	2.7 kΩ pull-up
Count Frequency	f			100	kHz	Velocity (rpm) x N/60
Shaft Perpendicularity Plus Axial Play (HEDS Series)				± 0.25 (± 0.010)	mm (in.)	6.9 mm (0.27 in.) from mounting surface
Shaft Eccentricity Plus Radial Play (HEDS Series)				0.04 (0.0015)	mm (in.) TIR	6.9 mm (0.27 in.) from mounting surface
Shaft Perpendicularity Plus Axial Play (HEDM Series)				± 0.175 (± 0.007)	mm (in.)	6.9 mm (0.27 in.) from mounting surface
Shaft Eccentricity Plus Radial Play(HEDM Series)				0.04 (0.0015)	mm (in.) TIR	6.9 mm (0.27 in.) from mounting surface

Note: The module performance is guaranteed to 100 kHz but can operate at higher frequencies. 2.7 kΩ pull-up resistors required for HEDS-5540 and 5640.

Encoding Characteristics

Encoding Characteristics over Recommended Operating Range and Recommended Mounting Tolerances unless otherwise specified. Values are for the worst error over the full rotation.

Part No.	Description		Sym.	Min.	Typ.*	Max.	Units
HEDS-5500 HEDS-5600 (Two Channel)	Pulse Width Error		ΔP		7	45	°e
	Logic State Width Error		ΔS		5	45	°e
	Phase Error		Δϕ		2	20	°e
	Position Error		ΔΘ		10	40	min. of arc
	Cycle Error		ΔC		3	5.5	°e
HEDM-5500 HEDM-5600 (Two Channel)	Pulse Width Error		ΔP		10	45	°e
	Logic State Width Error		ΔS		10	45	°e
	Phase Error		Δϕ		2	15	°e
	Position Error		ΔΘ		10	40	min. of arc
	Cycle Error		ΔC		3	7.5	°e
HEDS-5540 HEDS-5640 (Three Channel)	Pulse Width Error		ΔP		5	35	°e
	Logic State Width Error		ΔS		5	35	°e
	Phase Error		Δϕ		2	15	°e
	Position Error		ΔΘ		10	40	min. of arc
	Cycle Error		ΔC		3	5.5	°e
	Index Pulse Width		P_O	55	90	125	°e
	CH. I rise after CH. A or CH. B fall	-25°C to +100°C	t_1	10	100	250	ns
		-40°C to +100°C	t_1	-300	100	250	ns
	CH. I fall after CH. B or CH. A rise	-25°C to +100°C	t_2	70	150	300	ns
		-40°C to +100°C	t_2	70	150	1000	ns

Note: See Mechanical Characteristics for mounting tolerances.
*Typical values specified at V_{CC} = 5.0 V and 25°C.

Electrical Characteristics

Electrical Characteristics over Recommended Operating Range.

Part No.	Parameter	Sym.	Min.	Typ.*	Max.	Units	Notes
HEDS-5500	Supply Current	I_{CC}		17	40	mA	
HEDS-5600	High Level Output Voltage	V_{OH}	2.4			V	I_{OH} = -40 µA max.
	Low Level Output Voltage	V_{OL}			0.4	V	I_{OL} = 3.2 mA
	Rise Time	t_r		200		ns	C_L = 25 pF
	Fall Time	t_f		50		ns	R_L = 11 kΩ pull-up
HEDS-5540	Supply Current	I_{CC}	30	57	85	mA	
HEDS-5640	High Level Output Voltage	V_{OH}	2.4			V	I_{OH} = -200 µA max.
HEDM-5500	Low Level Output Voltage	V_{OL}			0.4	V	I_{OL} = 3.86 mA
HEDM-5600	Rise Time	t_r		180		ns	C_L = 25 pF
	Fall Time	t_f		40		ns	R_L = 2.7 kΩ pull-up
HEDM-5500	Supply Current	I_{CC}	30	57	85	mA	
HEDM-5600	High Level Output Voltage	V_{OH}	2.4			V	I_{OH} = -40 µA max.
	Low Level Output Voltage	V_{OL}			0.4	V	I_{OL} = 3.86 mA
	Rise Time	t_r		180		ns	C_L = 25 pF
	Fall Time	t_f		40		ns	R_L = 3.2 kΩ pull-up

*Typical values specified at V_{CC} = 5.0 V and 25°C.

Mechanical Characteristics

Parameter	Symbol	Dimension	Tolerance[1]	Units
Codewheel Fits These Standard Shaft Diameters		2 3 4 5 6 8	+0.000 -0.015	mm
		5/32 1/8 3/16 1/4	+0.0000 -0.0007	in
Moment of Inertia	J	0.6 (8.0×10^{-6})		g-cm^2 (oz-in-s^2)
Required Shaft Length[2]		14.0 (0.55)	± 0.5 (± 0.02)	mm (in.)
Bolt Circle[3]	2 screw mounting	19.05 (0.750)	± 0.13 (± 0.005)	mm (in.)
	3 screw mounting	20.90 (0.823)	± 0.13 (± 0.005)	mm (in.)
	external mounting ears	46.0 (1.811)	± 0.13 (± 0.005)	mm (in.)
Mounting Screw Size[4]	2 screw mounting	M 2.5 or (2-56)		mm (in.)
	3 screw mounting	M 1.6 or (0-80)		mm (in.)
	external mounting ears	M 2.5 or (2-56)		mm (in.)
Encoder Base Plate Thickness		0.33 (0.130)		mm (in.)
Hub Set Screw		(2-56)		(in.)

Notes:
1. These are tolerances required of the user.
2. The HEDS-55X5 and 56X5, HEDM-5505, 5605 provide an 8.9 mm (0.35 inch) diameter hole through the housing for longer motor shafts. See Ordering Information.
3. The HEDS-5540 and 5640 must be aligned using the aligning pins as specified in Figure 3, or using the alignment tool as shown in "Encoder Mounting and Assembly". See also "Mounting Considerations."
4. The recommended mounting screw torque for 2 screw and external ear mounting is 1.0 kg-cm (0.88 in-lbs). The recommended mounting screw torque for 3 screw mounting is 0.50 kg-cm (0.43 in-lbs).

Electrical Interface

To insure reliable encoding performance, the HEDS-5540 and 5640 three channel encoders require 2.7 kΩ (± 10%) pull-up resistors on output pins 2, 3, and 5 (Channels I, A, and B) as shown in Figure 1. These pull-up resistors should be located as close to the encoder as possible (within 4 feet). Each of the three encoder outputs can drive a single TTL load in this configuration.

The HEDS-5500, 5600, and HEDM-5500, 5600 two channel encoders do not normally require pull-up resistors. However, 3.2 kΩ pull-up resistors on output pins 3 and 5 (Channels A and B) are recommended to improve rise times, especially when operating above 100 kHz frequencies.

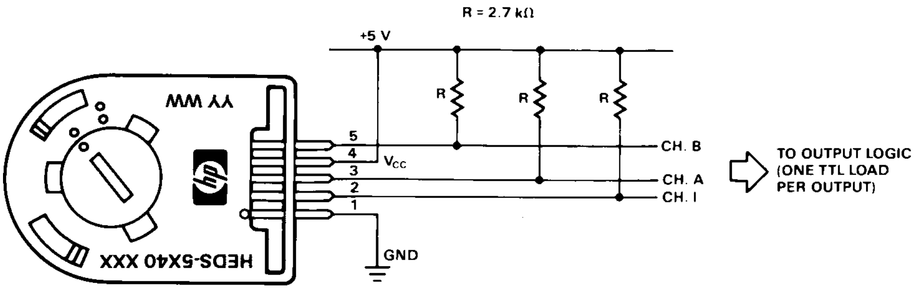


Figure 1. Pull-up Resistors on HEDS-5X40 Encoder Outputs.

Mounting Considerations

The HEDS-5540 and 5640 three channel encoders and the HEDM Series high resolution encoders must be aligned using the aligning pins as specified in Figure 3, or using the HEDS-8910 Alignment Tool as shown in Encoder Mounting and Assembly.

The use of aligning pins or alignment tool is recommended but not required to mount the HEDS-5500 and 5600. If these two channel encoders are attached to a motor with the screw sizes and mounting tolerances specified in the mechanical characteristics section without any additional mounting bosses, the encoder output errors will be within the maximums specified in the encoding characteristics section.

The HEDS-5500 and 5540 can be mounted to a motor using either the two screw or three screw mounting option as shown in Figure 2. The optional aligning pins shown in Figure 3 can be used with either mounting option.

The HEDS-5600, 5640, and HEDM-5600 have external mounting ears which may be used for mounting to larger motor base plates. Figure 4 shows the necessary mounting holes with optional aligning pins and motor boss.

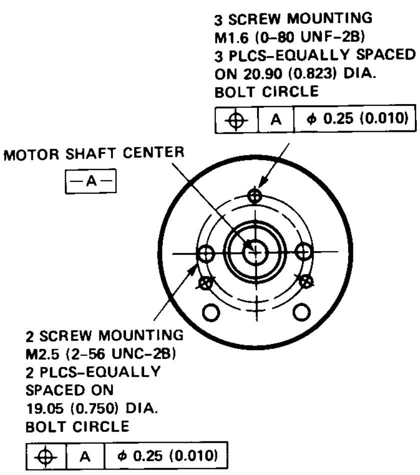


Figure 2. Mounting Holes.

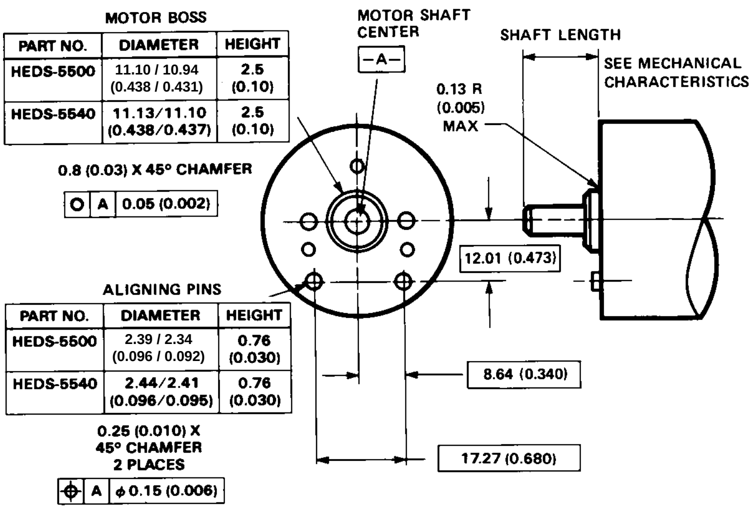
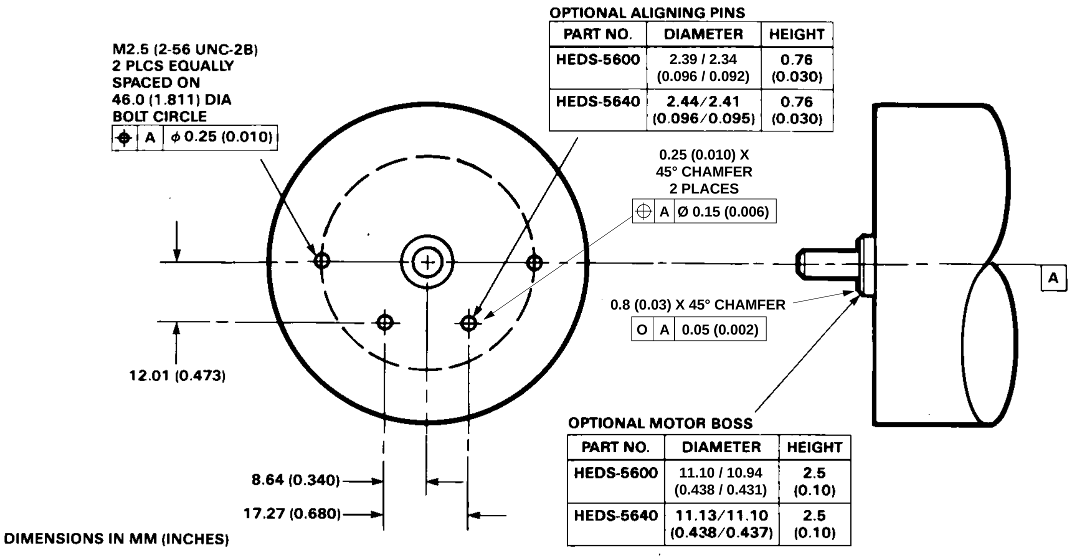


Figure 3. Optional Mounting Aids.



OPTIONAL ALIGNING PINS

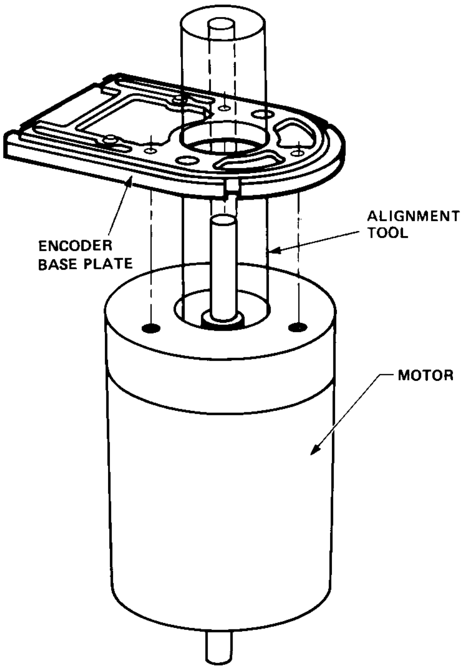
PART NO.	DIAMETER	HEIGHT
HEDS-5600	2.39 / 2.34 (0.096 / 0.092)	0.76 (0.030)
HEDS-5640	2.44/2.41 (0.096/0.095)	0.76 (0.030)

OPTIONAL MOTOR BOSS

PART NO.	DIAMETER	HEIGHT
HEDS-5600	11.10 / 10.94 (0.438 / 0.431)	2.5 (0.10)
HEDS-5640	11.13/11.10 (0.438/0.437)	2.5 (0.10)

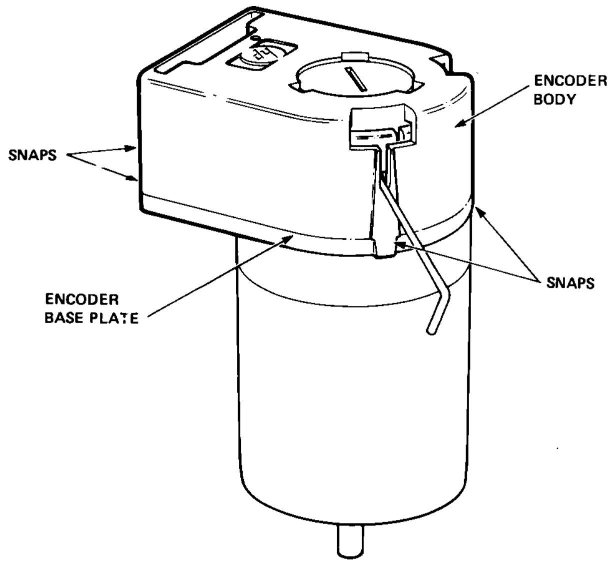
Figure 4. Mounting with External Ears.

Encoder Mounting and Assembly

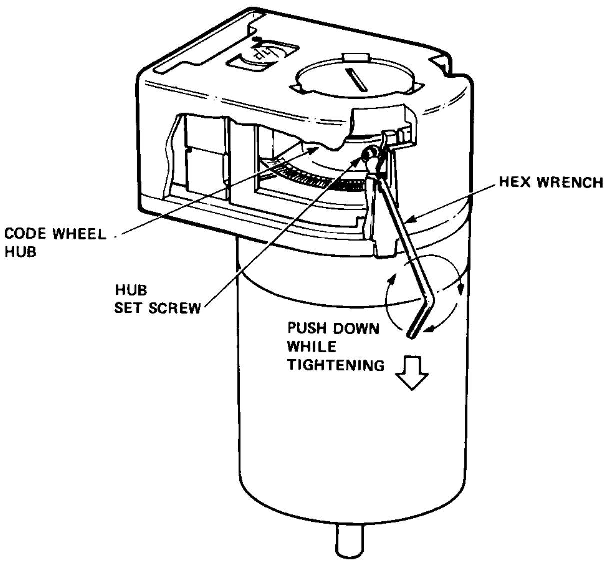


1. For HEDS-5500 and 5600: Mount encoder base plate onto motor. Tighten screws. Go on to step 2.

1a. For HEDS-5540, 5640 and HEDM-5500, 5600: Slip alignment tool onto motor shaft. With alignment tool in place, mount encoder baseplate onto motor as shown above. Tighten screws. Remove alignment tool.



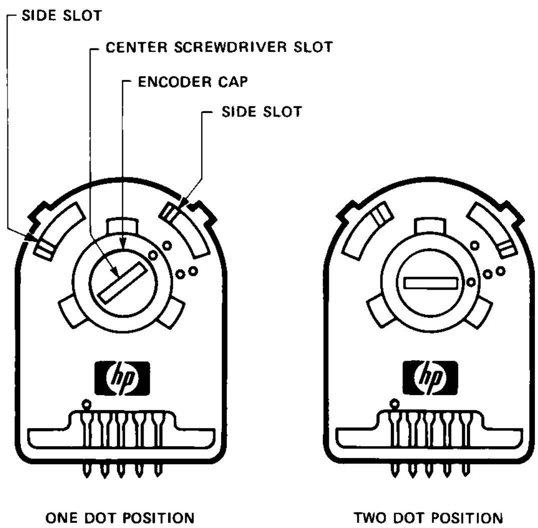
2. Snap encoder body onto base plate locking all 4 snaps.



3a. Push the hex wrench into the body of the encoder to ensure that it is properly seated into the code wheel hub set screws. Then apply a downward force on the end of the hex wrench. This sets the code wheel gap by levering the code wheel hub to its upper position.

3b. While continuing to apply a downward force, rotate the hex wrench in the clockwise direction until the hub set screw is tight against the motor shaft. The hub set screw attaches the code wheel to the motor's shaft.

3c. Remove the hex wrench by pulling it straight out of the encoder body.

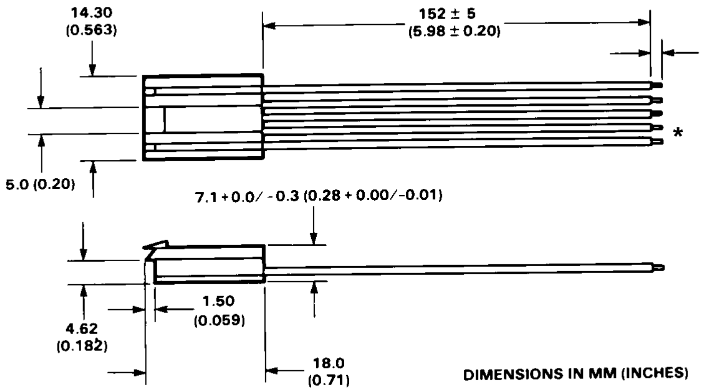


4. Use the center screwdriver slot, or either of the two side slots, to rotate the encoder cap dot clockwise from the one dot position to the two dot position. Do not rotate the encoder cap counterclockwise beyond the one dot position.

The encoder is ready for use!

Connectors

Manufacturer	Part Number
AMP	103686-4 640442-5
Dupont/Berg	65039-032 with 4825X-000 term.
HP (designed to mechanically lock into the HEDS-5XXX, HEDM-5X0X Series)	HEDS-8902 (2 ch.) with 4-wire leads
	HEDS-8903 (3 ch.) with 5-wire leads
Molex	2695 series with 2759 series term.

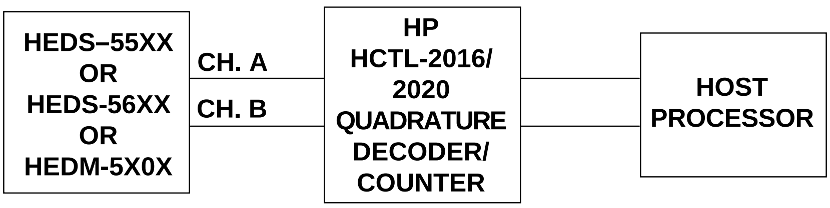


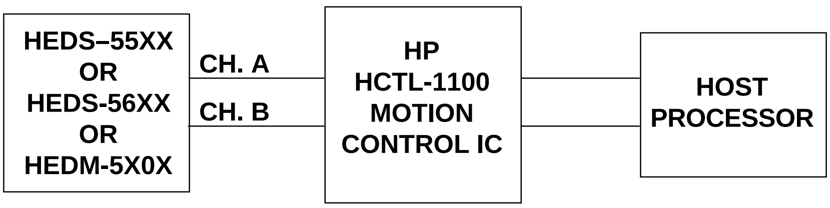
PIN NUMBER	PARAMETER	HEDS-8902 COLORS	HEDS-8903 COLORS
1	GROUND	BLACK	BLACK
2	CH. I	N/A*	BLUE
3	CH. A	WHITE	WHITE
4	V_{CC}	RED	RED
5	CH. B	BROWN	BROWN

*THIS WIRE IS NOT ON HEDS-8902.

Figure 5. HEDS-8902 and 8903 Connectors.

Typical Interfaces





Ordering Information

Encoders with Metal Codewheels

HEDS-5 □□□ Option □ □□

Mounting Type
- 5 - Standard
- 6 - External Mounting Ears

Outputs
- 0 - 2 Channel
- 4 - 3 Channel

Through Hole
- 0 - None
- 5 - 8.9 mm (0.35 in.)

Resolution (Cycles/Rev)

(HEDS-550X, 560X 2 Channel)

S - 50 CPR	F - 256 CPR
K - 96 CPR	G - 360 CPR
C - 100 CPR	H - 400 CPR
D - 192 CPR	A - 500 CPR
E - 200 CPR	I - 512 CPR

(HEDS-554X, 564X 3 Channel)

S - 50 CPR	G - 360 CPR
K - 96 CPR	H - 400 CPR
C - 100 CPR	A - 500 CPR
E - 200 CPR	I - 512 CPR
F - 256 CPR	

Shaft Diameter

01 - 2 mm	06 - 1/4 in.
02 - 3 mm	11 - 4 mm
03 - 1/8 in.	14 - 5 mm
04 - 5/32 in.	12 - 6 mm
05 - 3/16 in.	13 - 8 mm

HEDS-8910 0 □□ Alignment Tool

(Included with each order of HEDS-554X/564X three channel encoders)

Encoders with Film Codewheels

HEDM-5 □ 0 □ Option □ □□

Mounting Type
- 5 - Standard
- 6 - External Mounting Ears

Outputs
- 0 - 2 Channel

Through Hole
- 0 - None
- 5 - 8.9 mm (0.35 in.)

Resolution (Cycles/Rev)
- B - 1000 CPR
- J - 1024 CPR

Shaft Diameter

01 - 2 mm	06 - 1/4 in.
02 - 3 mm	11 - 4 mm
03 - 1/8 in.	14 - 5 mm
04 - 5/32 in.	12 - 6 mm
05 - 3/16 in.	13 - 8 mm

HEDS-8910 0 □□ Alignment Tool

(Included with each order of HEDM-550X/560X two channel encoders)