Cheetah - compliant quadruped robot

Simon Rutishauser

Responsible professor: Auke Ijspeert
Assistant: Alexander Spröwitz

January 11, 2008

Abstract

In this report the mechanical development and implementation of a new quadruped robot "Cheetah" is described in detail. "Cheetah" features three-segment legs with passive compliant knee joints. Each leg has two degrees of freedom - knee and hip joint can be actuated using RC servo motors.

Simple electronics to command the actuators from a desktop computer have been designed in order to test the robot. Ideas for more complete autonomous electronics are sketched.

Preliminary tests were performed to verify the correct functioning of the mechanical design.

Contents

List of Tables

List of Figures

1 Introduction

The objective of this project is to design a compliant quadruped robot leg. It should be similar to the legs of Puppy II [8, 17, 18] with the additional feature of active leg retraction in order to allow for several gaits. The leg must be as lightweight as possible in order to support high step frequencies - the idea here is to be able to create different gaits: walk, trot and pace. Additionally the knee joint should remain passive compliant, such as it was for Puppy II.

These requirements have consequences: The actuators have to be placed proximal in order to minimize leg inertia. The number of actuated degrees of freedom per leg is two - less actuated degrees of freedom such as for Puppy II would limit the robot to hopping gaits since the leg can't be retracted[1], more degrees of freedom would increase the overall weight of the robot and it's energy consumption. The leg should consist of three segments since this has been found to be energetically advantageous [7, 15].

In a first step, different cinematics and actuation principles will be drafted and evaluated with respect to criterions such as weight, energy consumption, feasibility and robustness.

Next the best design is to be dimensioned, using lots of approximations. Continuous-signal force sensors must be included in the design. Geometrical, mechanical and energy limits have to be evaluated and considered. Then the parts have to be drawn using CAD software, manufactured and assembled.

Finally simple test electronics have to be created, to allow testing of the mechanics and possibly some experiments.

---

[1]Puppy II has a passively compliant knee joint. Since the leg can't be actively retracted only hopping gaits are possible. The new design should be able to do walk, trot, pace and other gaits.

2 Task description

This semester project (12 h per week, 17 weeks) aims at an active compliant leg design for a robot structure similar to the quadruped robot Puppy I and II[2] [8, 17, 18].

The task is to design an extremely lightweight, three-segment leg with two actuated joints; the hip joint and the knee joint. With the additional actuation in the knee joint this robot should be able to actively retract its legs, ultimately resulting in several different dynamic gaits for the quadruped legged robot structure.

Both the actuators for the hip joints and the actuators for the knee joints should be mounted proximal, to decrease the weight of the legs and increase the leg dynamics. Concerning the overall dimensions: we are aiming at a slightly smaller size than Puppy II, main restriction are the available torque delivered by the RC servo motors. The weight of the construction (structure, electronics, possibly including the batteries) should not exceed the weight of the actuators (therefore all parts in sum about 1 kg). Similar to Puppy II the new robot has to feature robust, non-binary touch sensors. These can be implemented as pressure sensors in the feet or by sensing the deflection of the compliance in the joints, a redundant system is preferred. The foot design of the robot should be modular, to test several different compliant designs. For the same reason the compliance in the l_2-segment [7, 24] (possibly actuated as an in-series-elastic actuation system) should be exchangeable, possibly also into a non-compliant segment.

A complex body of the aimed quadruped robot is not the primary task of this project, however a very simple body frame needs to be included in the construction, to mount on the legs, actuators, batteries, and the electronics.

The project should result in a report, a project presentation, a web page summarising the work done and a CD with developed software and CAD drawings.

---

[2]Puppy II features a compliant, passive, two-segment leg, each leg is actuated at the hip joint by a strong RC servo motor. Puppy (I) features a compliant three-segment leg design with an actuated knee joint, however the knee actuator is attached at the knee and not proximal, as aimed in this project.

3 Project definition and objective list

3.1 Objective list

C: compulsory
O: objective
W: wish

Functions

Importance	Objective
C	compliant mammal-like 3-segment leg design
O	minimal inertia, proximal motors as sketched in Fig. 4
C	retractable in order to enable different gaits
C	compliance through spring in series with retraction mechanism
C	actuated hip, 2 actuators per leg
C	leg can be disassembled and replaced
O	good gait robustness
O	mechanically robust and lightweight (total robot weight should not exceed 1 kg)
O	modular foot design such that different compliant feet may be tested
C	force sensor with gradual output in X and Y direction, see Fig. 4
C	sensing redundancy by measuring φ_{34}, see Fig. 4
W	include possibility of sagittal oscillations in robot
W	adjustable spring constant (through spring replacement)

Manufacturing

Importance	Objective
O	can be easily manufactured at EPFL, preferred is PCB material (FR4), POM or Al/AlMg alloys and/or off-the-shelf components
O	dimensioned for use with Kondo KRS-2350 or Dynamixel AX-12+ servo motors, therefore similar to the Puppy II legs

3.2 Timeplan

Figure 1 shows a gantt-chart of the project timeplanning.
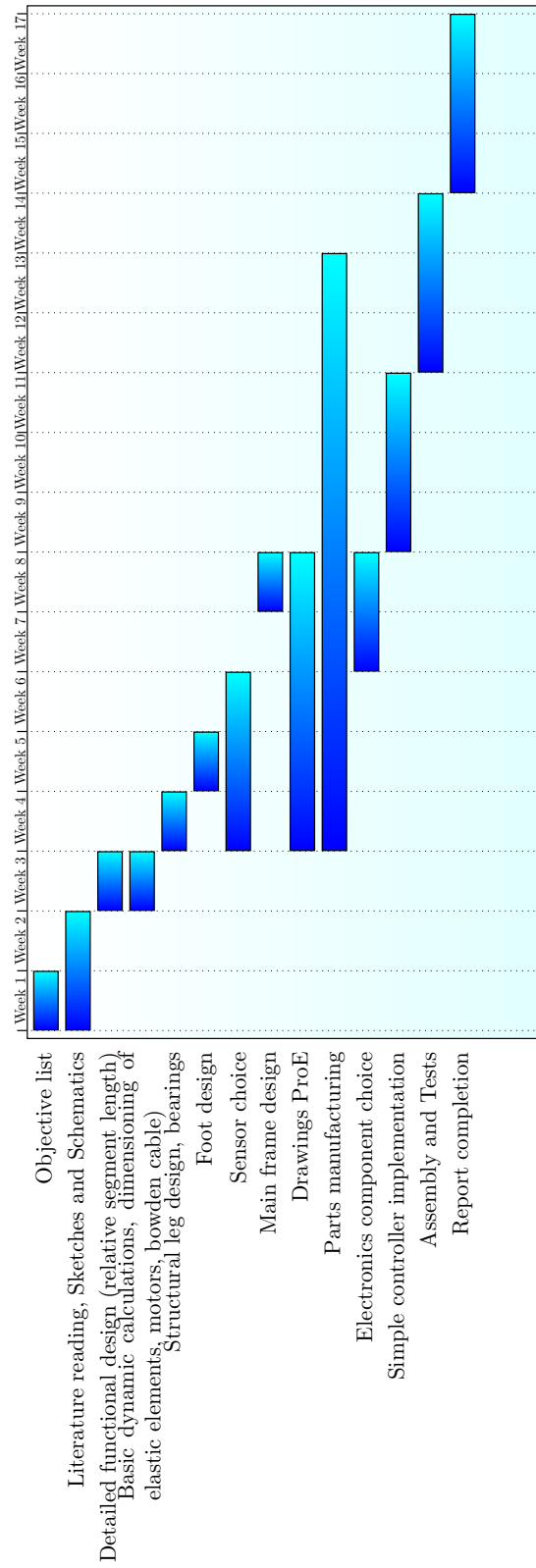
Figure 1: Project timeplan

3.3 Journal

Week	Progress
1	Detailed Objective list, Paper reading
2	Hardware sketches and Timeplan
3	Start construction variant evaluation (leg length actuation), calculations to evaluate leg length and compliance.
4	Calculations on spring-mass running model, equivalence to the three segment leg.
5	Choice of elastic element and motor M_2.
6	Mechanical design, bearings
7	Pro/Engineer leg drawings
8	Pro/Engineer leg drawings - alternative construction, bowden cable and knee actuation motor
9	Midterm presentation, construction of feet, motors
10	2D construction drawings (see Appendix) and IGS files, autonomous electronics meeting with Pierre-André and Alessandro
11	Small changes to 2D drawings, handing in
12	Report writing, looking for nonautonomous electronics for first tests
13	Programming PWM on PIC - did not work due to hardware bug
14	Programming PWM on PIC - second variant
15	Writing host software to send setpoints over serial bus, CPG
16	Leg Assembly
17	Complete assembly, some tests, slight modifications

3.4 Hardware sketches
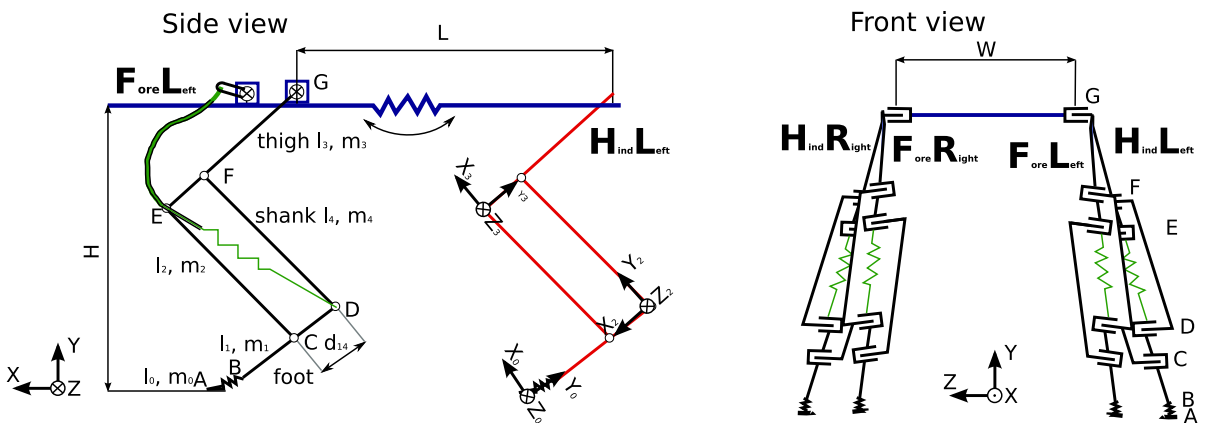
Figure 2: Hardware sketch labeling the different points and segment lengths

3.4.1 Relative segment lengths

In addition to the absolute segment lengths as defined in Fig. 2, segment lengths relative to the maximum leg length are defined as following (Fig. 3):

The subtraction of the relative distance between segments l_2 and l_4, called $\frac{\lambda_p}{2}$ in the terms for λ_1 and λ_3 is due to the fact that the studies on relative segment lengths [6, 7, 22, 24] don't consider pantographic legs specifically, but rather work with only one middle segment. This subtraction is a mean way between considering leg segment l_2 as the mid segment and neglecting segment l_4 and the other way around.

$$\lambda_0 = \frac{l_0}{l_0 + l_1 + l_2 + l_3} \tag{1}$$

$$\lambda_1 = \frac{l_1}{l_0 + l_1 + l_2 + l_3} - \frac{\lambda_p}{2} \tag{2}$$

since in many studies for three-segmented legs l_0 and l_1 are one segment, we also introduce

$$\lambda_{01} = \frac{l_1 + l_0}{l_0 + l_1 + l_2 + l_3} - \frac{\lambda_p}{2} = \lambda_0 + \lambda_1 \tag{3}$$

$$\lambda_2 = \frac{l_2}{l_0 + l_1 + l_2 + l_3} \tag{4}$$

$$\lambda_3 = \frac{l_3}{l_0 + l_1 + l_2 + l_3} - \frac{\lambda_p}{2} \tag{5}$$

$$\lambda_p = \frac{|CD|}{l_0 + l_1 + l_2 + l_3} = \frac{|EF|}{l_0 + l_1 + l_2 + l_3} \tag{6}$$
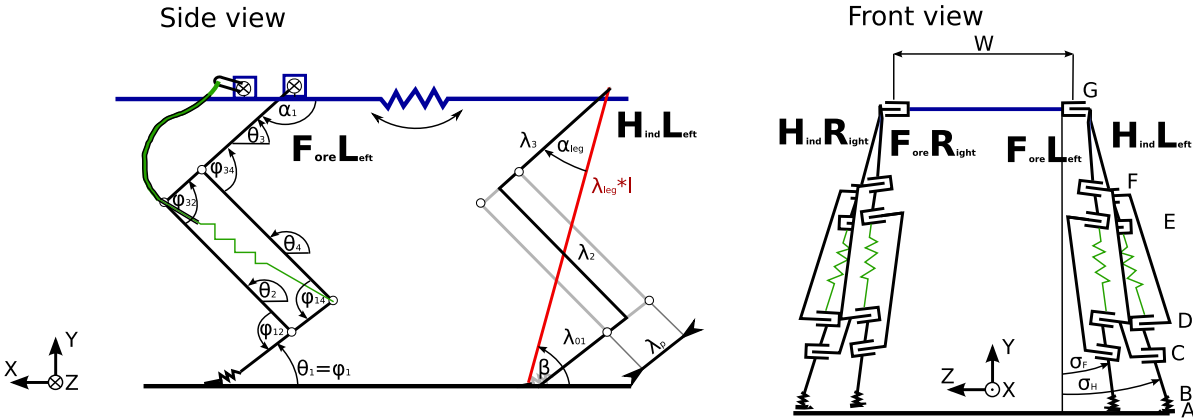
Figure 3: Hardware sketch labeling the angles
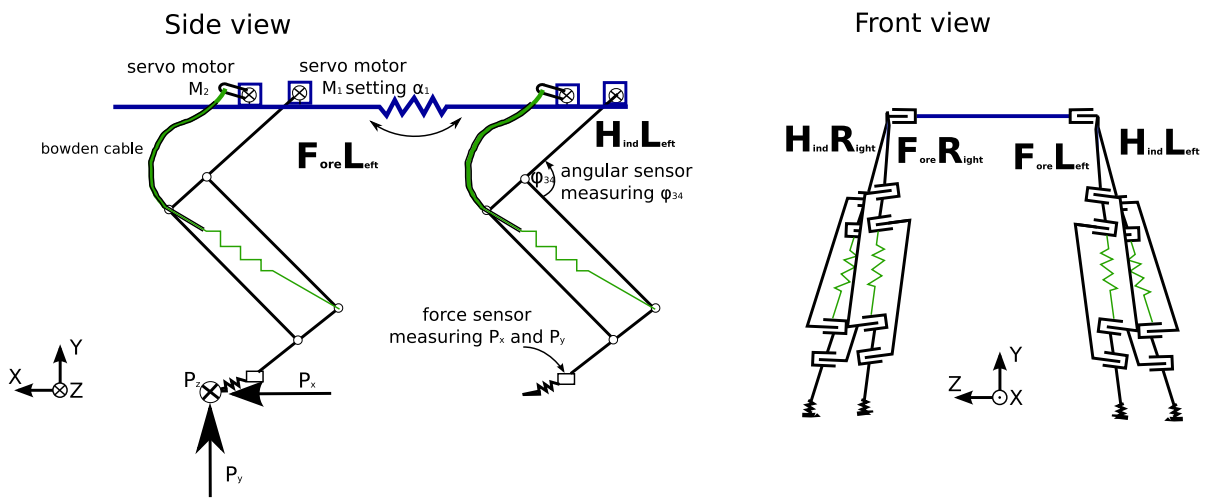
Figure 4: Hardware sketch showing actuators, external forces and sensors

4 Construction variant evaluation

4.1 Actuation of leg length using bowden cables

4.1.1 Sketches and explanations

u<v<w

a) flight

b) touchdown

c) midstance

F_{spring} F_{servo}

a)

F_{spring} F_{ext}

b)

F_{spring} F_{ext}

c)
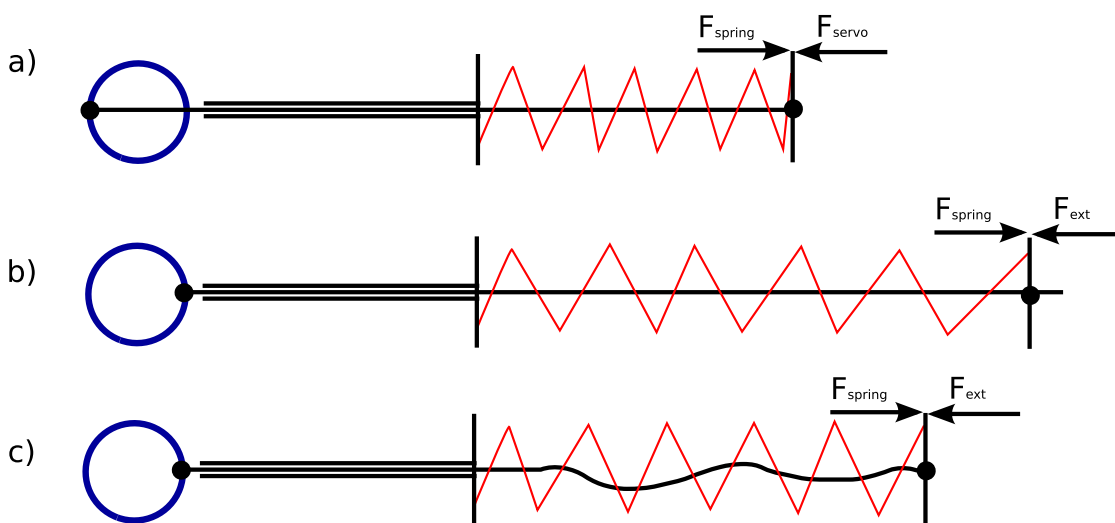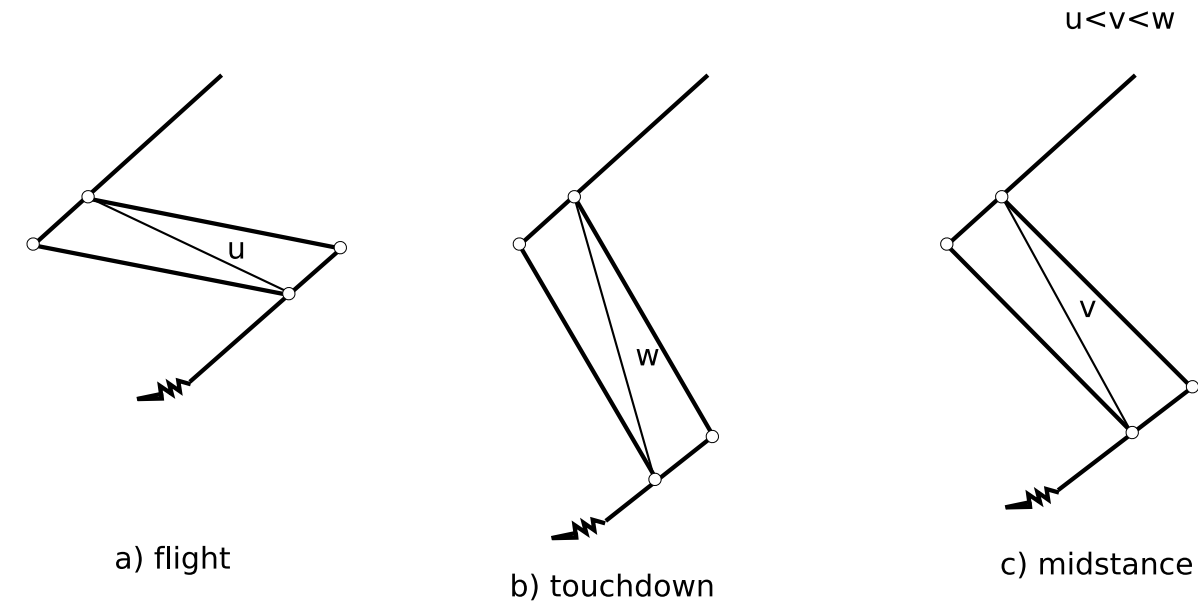
Figure 5: Sketch of bowden cable actuation where the external forces act on the spring

There are two main ways of modifying the leg length. The first way is by connecting points F and C (points are labelled in Fig. 2) as shown in Fig. 5. The alternative to this is to connect points E and D as sketched in Fig. 6.
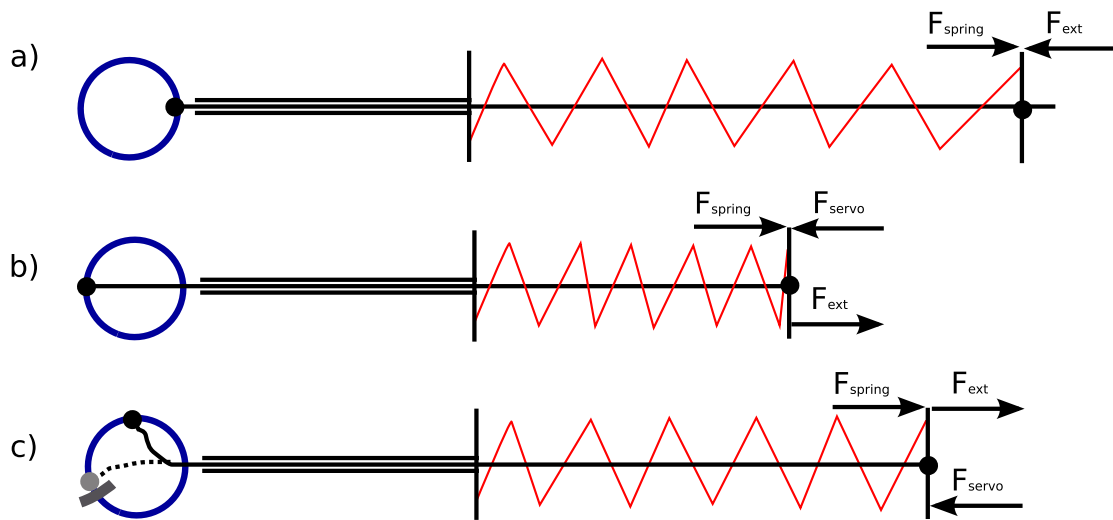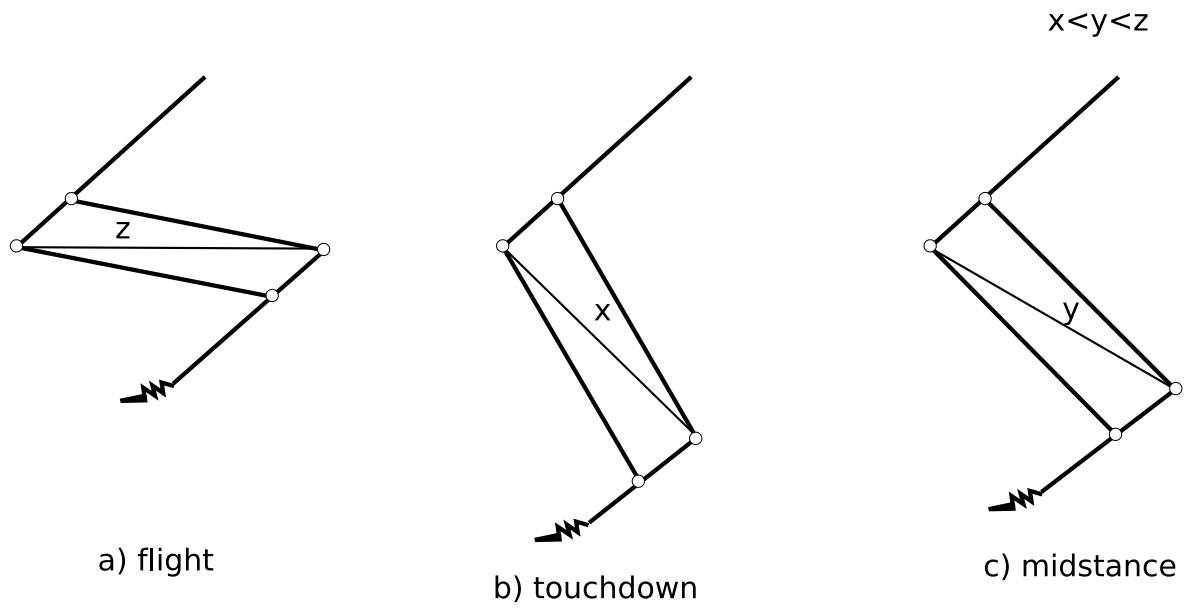
Figure 6: Sketch of bowden cable actuation where the external forces act as tension on the bowden cable and thereby on the motor

In the first case, shown in Fig. 5 the compressive spring extends the leg before touch-down and becomes then gravity loaded. The spring may or may not be precharged. The motor only serves to retract the leg during flight phase and to support leg retraction during the first part of the stance phase, loading thereby the spring.

The disadvantage of this system is that if the spring is not preloaded, the force at touchdown is very low which leads to a large leg retraction until midstance. If however the spring is preloaded, the motor torque required for leg retraction increases significantly, especially when compared to the solutions sketched in Fig. 6 and 7.

As sketched the bowden cable is not tense during stance, if this is a problem the motor can tense the cable based on the angular sensor feedback.

In the second case (Fig. 6) a relaxing compressive spring retracts the leg during flight phase. It only needs strength sufficient for leg retraction. Before touchdown the servo motor loads the spring, extending thereby the leg. During stance the motor may either remain in the position it took at touchdown, or it may let the spring (and thus the leg) extend slightly until midstance, as shown in Fig. 6.

The disadvantage of this solution is that there's no compliance in the system, at least not at this level - this results in higher control demands. The external force at touchdown is applied directly through the bowden cable to the motor, which requires either energy for keeping the motor position or alternatively complying a little until midstance or makes the introduction of a mechanical stop necessary, as sketched by the dotted line and the gray areas.

A possible solution to the lacking in series compliant actuator in Fig. 6 is shown in Fig. 7. The actuation principle is the same, except for the added tensile spring which is precharged due to the green solid cylinder, as sketched in the figure.

However such a solution is rather complicated: The two springs have to be tuned, the system takes a lot of space and a mechanical stop has to be introduced somehow, taking space and adding weight to the system.

4.1.2 Evaluation table

For each criterion a rating $R \in [0, 4]$ is awarded, 0 being the worst and 4 the best rating.

The variants rated in Table 1 are:

- Variant 1: as in Fig. 5 with and without precharged spring

- Variant 2: as in Fig. 6 with motor fixed and compliant during stance

Furthermore a weight W is introduced, to represent the importance of each criterion.

Multiplying weights W with rating R and summing over all criterions a number representing the quality of the solution is obtained (the larger the better).
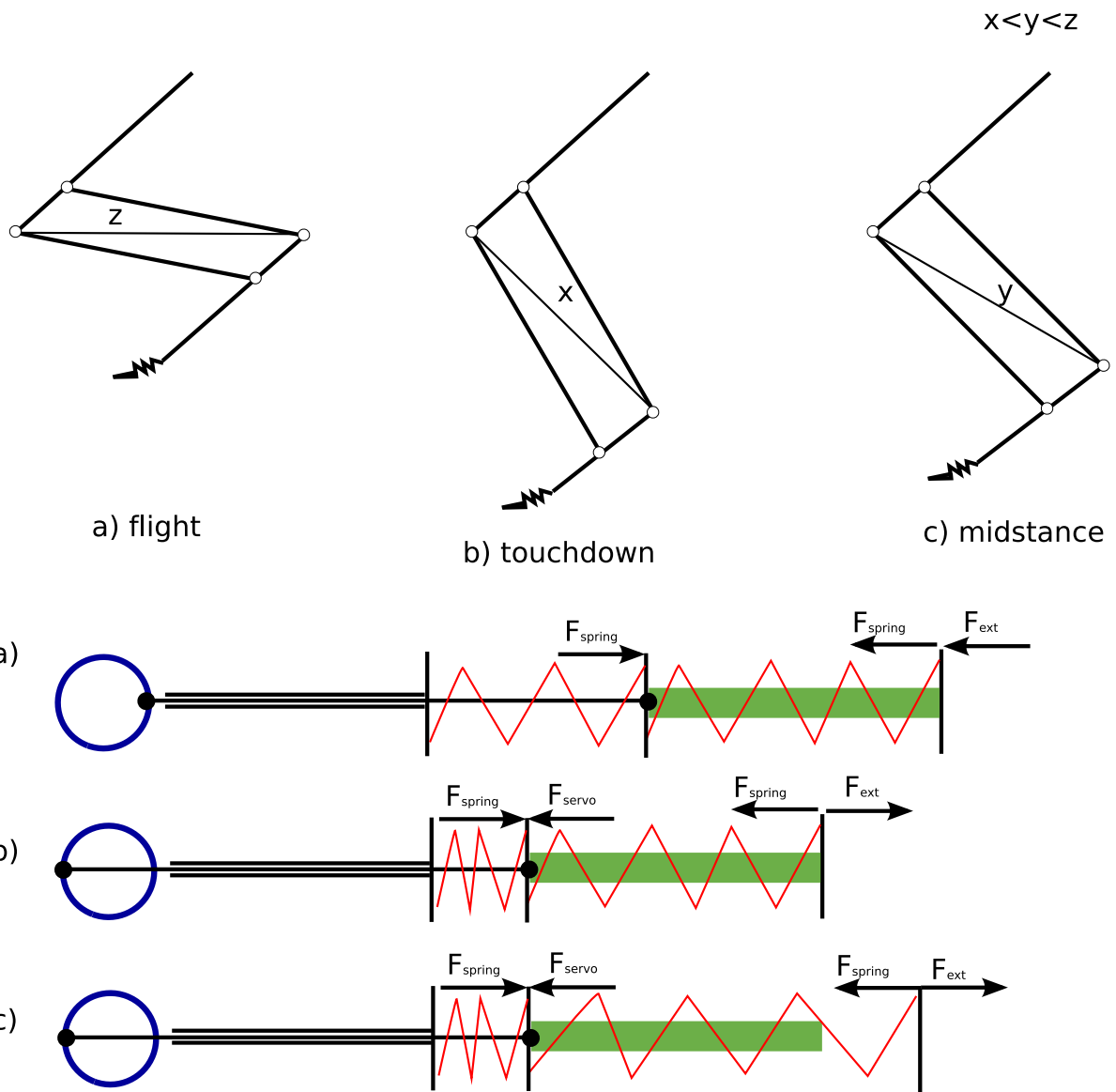
Figure 7: Sketch of bowden cable actuation where the external forces act as tension on the bowden cable and the precharged in-series tensile spring

Criterion	Weight W	Variant 1		Variant 2	
		R	$R \cdot W$	R	$R \cdot W$
Motor power and size	3	2, 3	6, 8	4, 2	12, 6
Command simplicity	1	4	4	4, 0	4, 0
Compliance	2	4, 2	8, 4	0, 2	0, 4
Required Spring length/force	1	3	3	4	4
Spring easily replaced	2	3	6	3	6, 6
Sum			27, 25		26, 20

Table 1: Evaluation table for different leg actuation schemes

5 Morphology and spring-mass modelling

5.1 Relative leg segment length

X-ray studies [12] show, that for small mammals the leg segments l_3 and l_1 (Fig. 2) are indeed nearly parallel during walking and running, which justifies the use of a pantograph mechanism.

There exist only a few recent studies [6, 7, 22, 24] on the relative segment lengths of three-segmented legs and their influence on energy efficiency, achievable acceleration.

Of these studies one [24] focuses mainly on stability, the risk of having the leg leave the "Z"-configuration. This leads to the suggestion of using a relative middle segment length λ_2 of around 0.4 for symmetrical operation ("Z"-configuration, both joint angles as defined in Fig. 3 are equal $\varphi_{14} = \varphi_{34}$) in order to prevent the leg configuration from becoming unstable and leaving the symmetrical configuration in favour of an asymmetrical one with non-equal segment angles $\varphi_{14} \neq \varphi_{34}$.

However these considerations are not important when a pantographical leg design is used - the leg segments l_1 and l_2 are always parallel, the angles φ_{14} and φ_{34} always equal.

Therefore we can concentrate on other considerations than stability when selecting the relative segment lengths. In terms of energy [7] a relative shank length $\lambda_2 = 0.45$ and a ratio between foot and thigh length of $\frac{\lambda_{01}}{\lambda_3} = 0.4$ which results in relative segment lengths of $\lambda_{01} = 0.16$ and $\lambda_3 = 0.39$ seems to be optimal. However this is in nature only seen for humans and large quadruped animals and comes at the cost of rather low acceleration.

For smaller mammals equal segment lengths $\lambda_{01} = \lambda_2 = \lambda_3 = \frac{1}{3}$ with relatively equal segment angles $\varphi_{14} = \varphi_{34}$ seem to be optimal for locomotion in crouched posture [6, 15, 21]. The advantage of this configuration is a large working range and good acceleration properties, since less torque is required for a certain acceleration. Interestingly enough similar relative leg segment lengths can be observed for different cats [10], though usually featuring a slightly shorter foot segment

Now the interesting question is to which group small cats such as the robot under construction belong. Considering the posture of small cats and an article on animal gaits [3] which considers dogs and cats in general as cursorial mammals it is decided to use the relative segment lengths (Fig. 3) proposed by [7].

$$\lambda_{01} = 0.16 \tag{7}$$

$$\lambda_2 = 0.45 \tag{8}$$

$$\lambda_3 = 0.39 \tag{9}$$

In addition a total leg length for the fore legs is fixed to

$$l = l_{fore} = l_0 + l_1 + l_2 + l_3 = 150\ mm \tag{10}$$

while the hind legs are going to be 20% longer as observed for several cat species such as the cheetah and lions [10]

$$l_{hind} = \frac{120}{100} l_{fore} = 180 \ mm \tag{11}$$

Admittedly the observed length difference for domestic cats which are of similar size as the robot is around 38% rather than 20%, but such a huge leg length difference would induce quite some construction problems and is therefore currently not realistic.

For the following calculations only the fore leg length l from Eq. 10 will be used.

The robot mass is estimated to:

$$m = \approx 4 \cdot 100 \ g[leg] + 8 \cdot 58 \ g[motors] + 100 \ g[battery] + 100 \ g[electronics and back] \approx 1 \ kg \tag{12}$$

Using the local coordinate system X_0, Y_0, Z_0 (Fig. 2) the relative leg length $l_\lambda = \frac{|AG|}{l_0 + l_1 + l_2 + l_3 - d_{14}}$ can be expressed in function of the angles $\varphi = \varphi_{14} = \varphi_{34}$ and the relative leg segment lengths (Fig. 3, Eq. 7, 8 and 9):

$$\mathbf{l_\lambda} = \lambda_{\mathbf{01}} + \lambda_{\mathbf{2}} + \lambda_{\mathbf{3}} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \lambda_{01} + \begin{pmatrix} \sin\varphi \\ -\cos\varphi \end{pmatrix} \lambda_2 + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \lambda_3$$

The modulus of the leg vector gives the relative leg length

$$l_\lambda = |\mathbf{l_\lambda}| = \sqrt{\lambda_2^2 \sin^2\varphi + (\lambda_{01} - \lambda_2 \cos\varphi + \lambda_3)^2} \tag{13}$$

whereas the inverse equation $\varphi(l_\lambda)$ is:

$$\cos(\varphi) = \frac{\lambda_{01}^2 + \lambda_2^2 + \lambda_3^2 + 2\lambda_{01}\lambda_3 - l_\lambda^2}{2\lambda_2(\lambda_{01} + \lambda_3)} \tag{14}$$

while the angle α_{leg} (Fig. 3) between the whole leg $\mathbf{GA}$ and the thigh segment l_3 amounts to

$$\cos\alpha_{leg} = \frac{\mathbf{l_\lambda} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}}{l_\lambda} = \frac{\lambda_{01} - \lambda_2 \cos\varphi + \lambda_3}{l_\lambda} \tag{15}$$

5.2 Expected gaits and corresponding velocities

In order to estimate the forces in play it is useful to know what gaits will be used at which speeds. It has been shown [3] that for the same Froude number, most quadruped animals show similar gaits.

The Froude number is defined as:

$$F = \frac{v^2}{gh} \sim \frac{hf^2}{g} \tag{16}$$

where $g = 9.81 \frac{m}{s^2}$ is the gravitational acceleration, h the leg length, v the velocity and f the leg stroke frequency.

Froude number	Associated gait	Duty factor D	Corresponding velocity $[\frac{m}{s}]$
0.5	amble or walk	0.7	0.8
1	trot or pace (symmetric running gaits)	0.5	1.1
2-3	canter or gallop (asymmetric gaits)	0.4	1.6

Table 2: Froude numbers (Eq. 16) and associated gaits

		amble		trot		canter		transverse gallop		rotary gallop	
LF	RF	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0.3	0	0.1	0	0.1
LH	RH	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	0	0.7	0	0.5	0.6	0.6	0.5

Table 3: Phase differences for the most common gaits, normalised to 1

Another important data is the duty factor, defined as the ratio between stance time and total step time:

$$D = \frac{T_{stance}}{T_{step}} \tag{17}$$

The same study [3] also mentions some typical Froude numbers and the associated gaits as listed in Table 2. The approximate velocities corresponding to these froude numbers for a leg length of $h = l_{leg} \cdot \lambda_{touchdown} = 136 \ mm$ (Eq. 24 and 10) and the corresponding duty factors D are listed as well.

However the velocities suggested by mammals of similar size [3] can in practise not be realised, the Kondo motor limits the leg stroke frequency to about $2.67 Hz$ (Eq. 21) and thereby the velocity to $v_{max} = 1.04 \frac{m}{s}$ (Eq. 22).

The gaits are defined [3, 14] by the phase difference between the four legs as listed in Table 3.

In order to verify that the velocities mentioned in Table 2 and the velocity calculated based on the motor limitations (Eq. 22) are realistic, they were compared to the maximum velocities of comparable robots:

- Tekken II [19] achieves $v_{max} = 0.7 \ \frac{m}{s}$

- Tekken IV [20] achieves $v_{max} = 1.5 \ \frac{m}{s}$

So our estimation in Table 2 and Eq. 22 is not unrealistic.

5.3 Geometrical restrictions for compliance

In order to dimension the compliant parts of the leg it is important to estimate how much it should retract during stance.

The maximal leg retraction is limited by the geometry of the system: point D (Fig. 2) must not touch the ground, which would imply $\beta = \alpha_{leg}$ (Fig. 3). Since the dynamics

of the systems are complicated, a worst-case scenario is evaluated: It is assumed that the leg has it's minimal length at touchdown (and not during midstance).

The angle of attack β_{attack} (β in Fig. 3 at touchdown) of the system can be estimated[3] [11] as:

$$\beta_{attack} \geq 55° \tag{18}$$

whereas the opposite angle is the half-angle swept during stance:

$$\alpha_{stance} \leq 35° \tag{19}$$

Now the limiting conditions can be found quite easily. It is known that $\beta_{attack} = 55° = \alpha_{leg}$ defines the limit of the system, where α_{leg} is defined in Eq. 15 and 13. Solving these equations for $\varphi = \varphi_{12} = \varphi_{14}$ the minimal allowed angle φ_{min} is found

$$\varphi_{min} = 19.43° \tag{20}$$

even if we add some margin this problem can be neglected.

5.4 Estimation of the required spring constant in the spring-mass running model

Replacing the complex three-segmented leg with a simple spring-mass model as it is common for modelling of (mostly) running gaits [7, 13, 23], the required equivalent spring constant of the leg can be estimated.

Since the eigenfrequency of the system can not be changed in function of the velocity, it will be calculated at maximum velocity, which is limited by the frequency the Kondo motor can provide[4] (Eq. 19):

$$f_{max,kondo} = \frac{1}{\omega_{kondo} \cdot 2\alpha_{stance} \cdot \frac{1}{D}} = \frac{1}{\frac{0.16}{60} 2 \cdot 35 \frac{1}{0.5}} = 2.67 \ Hz \tag{21}$$

The half-angle swept during stance $\alpha_{stance} = 35°$ (Eq. 19) permits also the calculation of how much the robot advances during the stance part of one step cycle (Eq. 10 and 24):

$$\Delta X_{stance} = 2 \cdot \sin(\alpha_{stance}) \cdot l_{\lambda,touchdown} \cdot l = 0.156 \ m$$

Knowing the duty factor ($D = 0.4$ for gallop is assumed), we also know how much the robot advances during one complete step cycle:

$$\Delta X_{step} = \frac{\Delta X_{stance}}{D} = 0.39 \ m$$

---

[3]The interesting diagram Fig. 2 in [11] was plotted for a dog with leg length $h = 0.5 \ m$, assuming a maximal Froude number (Eq. 16) of 2 (corresponding to gallop, Table 2) the maximal half-angle swept during stance θ amounts to $\theta = 35° = 90° - \beta_{attack}$ which results in the angle of attack $\beta_{attack} \geq 55°$

[4]Kondo KRS-2350 ICS motors have a maximum angular velocity of $0.16sec/60°$ at no load

Since the maximum step frequency is known, the corresponding velocity can be calculated:

$$v_{max} = \Delta X_{step} \cdot f_{max.kondo} = 0.39 \ m \cdot 2.67 \ Hz = 1.04\frac{m}{s} \tag{22}$$

Since for gallop gaits, the spring nearly always works in parallel with another spring of same spring constant (Tables 3 and 2), the eigenfrequency of the system is:

$$\omega = 2\pi f = \sqrt{\frac{2k_{leg}}{m}}$$

where the robot mass is assumed as $m = 1 \ kg$ (Eq. 12).

Solving for the leg spring constant it is found that for $f_{max,kondo} = 2.67 \ Hz$ the spring constant is:

$$k_{leg} = \frac{1}{2}m(2\pi f_{max,kondo})^2 = 140\frac{N}{m} \tag{23}$$

5.5 Minimal leg length at midstance in spring-mass model

A leg angle at touchdown of $\varphi_{touchdown} = 130°$ is assumed, which corresponds to the mean value of scapula, elbow, knee and ankle joint angles of different sized cats [10]. Smaller mammals on the contrary use often more crouched postures and therefore have slightly smaller leg segment angles along with the different relative leg segment lengths [12].

This results in a relative leg length (Eq. 13) of

$$l_{\lambda,touchdown} = \sqrt{(\lambda_2 \sin(\varphi))^2 + (\lambda_{01} - \lambda_2 \cos(\varphi) + \lambda_3)^2} = 0.91 \tag{24}$$

and a length of the spring assembly of

$$l_{s,touchdown} = |CF| = sqrt(l_2^2 + d_{14}^2 - 2 \cdot d_{14} \cdot l_2 \cos(\varphi)) = 75.8 \ mm \tag{25}$$

Assuming furthermore a velocity ($v_{max} = 1.04\frac{m}{s}$ for running, Eq. 22), mass ($m' = \frac{m}{2} = \frac{1}{2} \ kg$ (Eq. 12, where it is assumed that two legs share the robot weight - an assumption justified by the small phase difference for gallop, Table 3), spring constant $k_{leg} = 140 \ \frac{N}{m}$ (Eq. 23) and an angle of attack of $\beta_{attack} = 55°$ (Eq. 18), the minimal leg length during stance can be calculated for a simple spring-mass model [13] (using equations 1 to 12).

The following minimal relative leg length is obtained (a and b as defined in [13])

$$l_{\lambda,min} = l_{\lambda,touchdown} \cdot (1 + a - b) = 0.62 \tag{26}$$

and a stored elastic energy of (Eq. 10 and 24)

$$E = \frac{k_{leg}}{2} \cdot (l \cdot (l_{\lambda,touchdown} - l_{\lambda,min}))^2 = 0.13 \ J \tag{27}$$

5.6 Finding the spring constant of the leg through the static equilibrium equations

Writing the static equilibrium equations, where $\varphi = \varphi_{12} = \varphi_{14}$:

$$\sum F_{X_0} = P \sin(\alpha_{leg}) - F_2 \sin \varphi - F_4 \sin \varphi - F_s \sin \gamma = 0 \tag{28}$$

$$\sum F_{Y_0} = P \cos(\alpha_{leg}) + F_2 \cos \varphi + F_4 \cos \varphi - F_s \cos \gamma = 0 \tag{29}$$

$$\sum M_A = (l_{01} - d_{14})(F_2 \sin \varphi + F_s \sin \gamma) + l_{01} F_4 \sin \varphi = 0 \tag{30}$$

It is assumed that the ground reaction force is applied along the leg, that is in direction **AG** (Fig. 2) which results in an angle α_{leg} (Eq. 15) between ground reaction force and the lower leg segment l_{01}.

This has been shown as being valid at midstance, where we expect the largest leg contraction amplitude and force, for small quadruped mammals [5].

Furthermore the spring length amounts to

$$l_s(\varphi) = \sqrt{l_2^2 + d_{14}^2 - 2d_{14}l_2 \cos \varphi}$$

and the angle γ is

$$\gamma = \arcsin(\frac{l_2}{l_s} \sin \varphi)$$
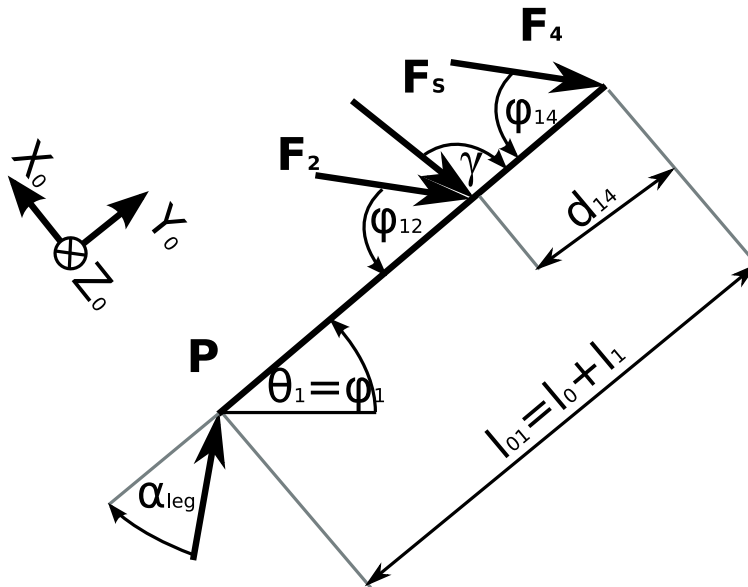
Figure 8: Sketch of forces on the lower leg segment l_{01}

The spring force is given by

$$F_s = k(l_{s,0} - l_s)$$

The unknowns are: F_2, F_4, P, φ (and F_s which depends on φ)

We can solve the equation system for different leg angles φ, thus obtaining a function that describes the nonlinear spring constant in function of leg angle φ or relative leg length (Eq. 13).

A plot of this function (Fig. 9) shows clearly the nonlinearity of the relationship. For low external forces, the obtained length and angle change is quite large. From a leg angle of $\varphi = 60°$ and a relative leg length of about $\lambda_{leg} = 0.58$ the leg becomes stiffer. If the external force is increased further the leg becomes unstable, once the force peak is passed.
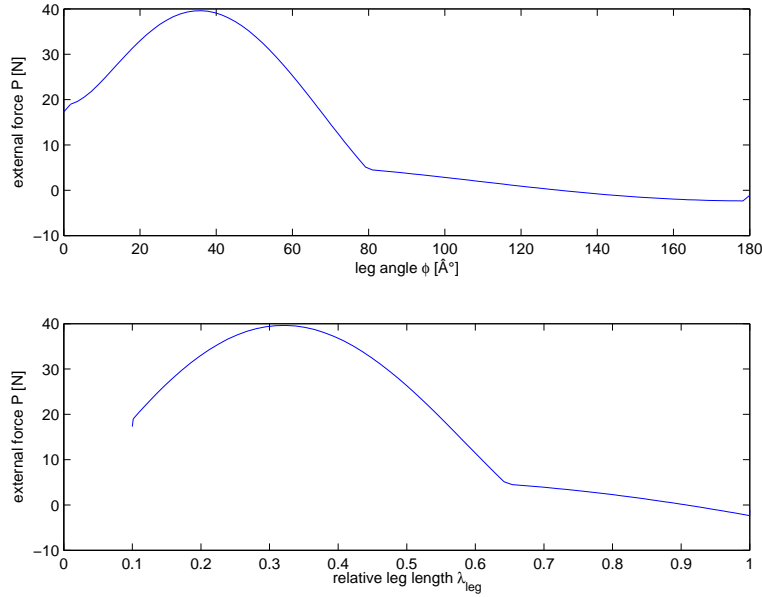
Figure 9: Plot of the relationship between leg length or angle and external force

Solving the equilibrium equations furthermore results in the internal leg forces along segments l_2 and l_4 (Fig. 10). During a normal stance cycle the forces are very low - in the order of ten newton at midstance (Eq. 26).

5.7 Leg length range

The spring choice is also influenced by the required leg length range - the maximal leg length is $l_{\lambda,touchdown} = 0.91$ (Eq. 24), at which the spring is assumed to be completely relaxed. The maximal spring contraction amounts occurs during the swing phase, where the motor has to retract the leg more than it retracts itself during stance (Eq. 26).

It is assumed that a leg rectraction exceeding the retraction during stance by one third is sufficient. Thus a minimal relative leg length of

$$l_{\lambda,min} = 0.50 \tag{31}$$

is required.

Since the length of the spring assembly at touchdown $l_{s,touchdown}$ is known (Eq. 25 and the length of the spring assembly during retraction amounts to

$$l_{s,retracted} = |CF| = sqrt(l_2^2 + d_{14}^2 - 2 \cdot d_{14} \cdot l_2 \cos(\varphi)) = 62.2 \; mm \tag{32}$$

the spring has to have a working range of at least (Eq. 25 and 32)

$$s_n = l_{s,touchdown} - l_{s,retracted} = 13.6 \; mm \tag{33}$$

5.8 Energetic equivalence between real-world leg and spring-mass model

The problem at this point is to translate the requirements on the linear spring in the spring-mass model into requirements on the real-world three segment leg, which can be considered as a nonlinear spring.

The easiest way to match the two models is to consider elastic energy and leg length. The elastic energy that has to be stored is the same in both models.
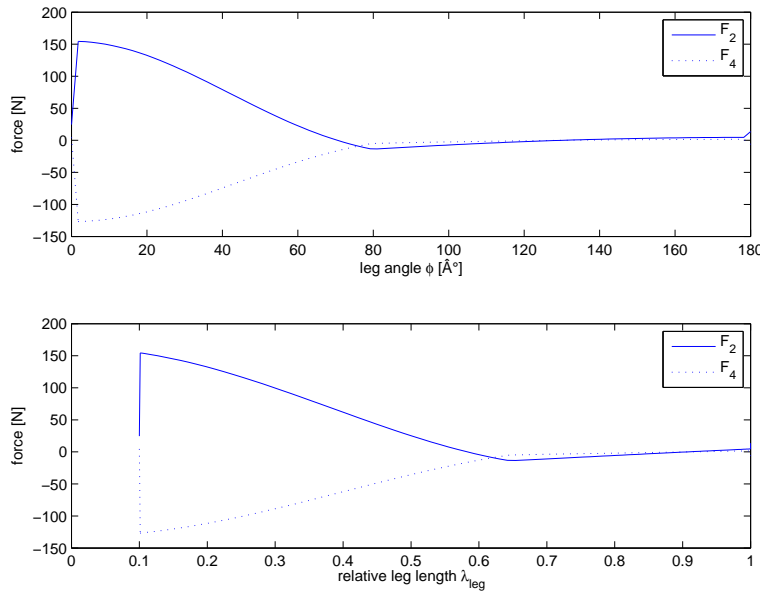
Figure 10: Plot of the relationship between leg length or angle and internal forces in segments l_2 and l_4

Knowing the force-displacement function for the real world leg (Fig. 9), the leg length for a certain stored elastic energy can be calculated by integrating the force F_{leg} over the leg length change Δl_{leg}.

Now different springs have to be tested, looking for one that results in a relative leg length as given by Eq. 26 at a stored elastic energy of Eq. 27.

A spring according to DIN 2098-1 is used. The data is shown in Table 4 and the energy graph is shown in Fig. 11, which corresponds closely to the spring-mass model used before. It is assumed that the spring is not precharged at touchdown.

wire diameter d $[mm]$	1
mean diameter D_m $[mm]$	9.3
hole diameter into which spring fits D_a $[mm]$	10.3
pin diameter on which spring fits D_d $[mm]$	8
relaxed length l_0 $[mm]$	25
maximal displacement s_n $[mm]$	17
number of working turns n	5.5
spring constant $k[\frac{N}{mm}]$	2.3

Table 4: Din 2098-1 spring data
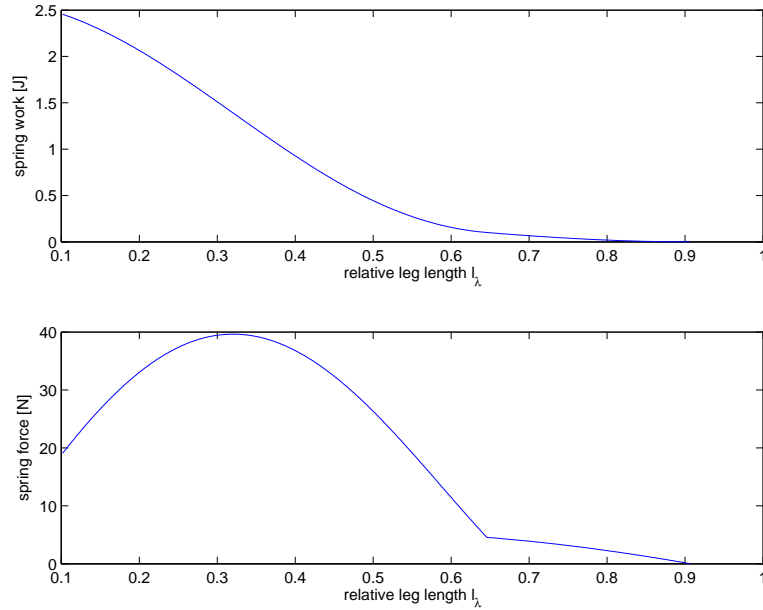
Figure 11: Plot of the relationship between leg length and stored energy

The force the motor has to provide in order to retract the spring to it's minimal length amounts to (Eq. 33 and Table 4)

$$F_{retract} = k_{spring} \cdot \Delta l_{spring} = 2.3 \cdot 10^3 \ [\frac{N}{m}] \cdot 13.6 \cdot 10^{-3} \ [m] = 31.6 \ [N] \qquad (34)$$

5.9 Estimating friction in the wire guiding cable

On the robot, the motor does not directly actuate the spring but rather does so by means of a cable. Friction in the cable adds to the required motor force. However estimating the amount of friction is a difficult task - to start with, the exact cable trajectory would have to be known, which is obviously not the case.

A rough estimation can nevertheless be performed, replacing the cable guiding spring with two friction points that force the cable to a Z-shaped trajectory (Fig. 12).
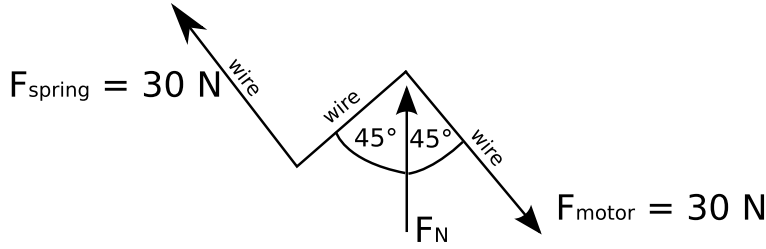
Figure 12: Sketch of forces assumed for friction estimation

The angles of $45°$ are randomly assumed - as mentioned before the wire trajectory as function of time is unknown. The force creating the friction is F_N - as long as the wire follows a straight trajectory in it's guiding spring there's almost no friction, only when the guiding spring is bent, friction increases.

Calculating F_N is straightforward - projecting all the forces on the vertical axis:

$$F_N = F_{motor} \cdot \sqrt{2} + F_{motor} \cdot \sqrt{2} = 2\sqrt{2}F_{motor} \approx 85 \; N \tag{35}$$

The friction force along the wire is therefore:

$$F_{friction} = \mu F_N = 0.1 \cdot 85 \; N = 8.5 \; N \tag{36}$$

Knowing that a lubricant is used, the friction coefficient is assumed to be $\mu = 0.1$. This is the upper bound of the range given by [9]: $0.04 < \mu < 0.1$ for friction of steel on steel with grease.

In order to achieve the Z-shaped trajectory, two friction points are needed. Therefore the total friction force amounts to:

$$F_{friction,tot} = 2 \cdot F_{friction} = 17 \; N \tag{37}$$

The total the motor needs to provide is thus (Eq. 34):

$$F_{retract,tot} = F_{retract} + F_{friction,tot} = 31.6 \; N + 17 \; N \approx 50 \; N \tag{38}$$

5.10 Dynamics in the spring-mass model using a nonlinear spring

As in [13] the Lagrangian is:

$$L = \frac{m}{2}(\dot{r}^2 + r^2\dot{\varphi}^2) - \frac{k(r)}{2}(l_0 - r)^2 - mgr\sin\varphi \tag{39}$$

This results in the following equations of motion for the system:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\varphi}}\right) - \frac{\partial L}{\partial \varphi} = \frac{d}{dt}mr^2\dot{\varphi} - mgr\cos\varphi = 0 \tag{40}$$

and

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{r}}\right) - \frac{\partial L}{\partial r} = \frac{d}{dt}(m\dot{r}) + mr\dot{\varphi}^2 + k(r)(l_0 - r) - mg\sin\varphi - \frac{dk(r)}{dr}\frac{(l_0 - r)^2}{2} = 0 \tag{41}$$

which is equivalent to:

$$m\ddot{r} = k(r)(l_0 - r) - \frac{dk}{dr}\frac{(l_0 - r)^2}{2} + mr\dot{\varphi}^2 - mg\sin\varphi \tag{42}$$

Equations 40 and 42 can be transformed into a set of ordinary differential equations, substituting $x_1 = \varphi$, $x_2 = \dot{\varphi}$, $x_3 = r$ and $x_4 = \dot{r}$:

$$\dot{x}_1 = x_2 \tag{43}$$
$$\dot{x}_2 = \frac{g\cos x_1}{x_3} - 2\frac{x_2 x_4}{x_3} \tag{44}$$
$$\dot{x}_3 = x_4 \tag{45}$$
$$\dot{x}_4 = \frac{k(r)}{m}(l_0 - x_3) - g\sin(x_1) + x_3 x_2^2 - \frac{dk}{dr}\frac{1}{2m}(l_0 - x_3)^2 \tag{46}$$

these equations can be solved to find the spring-mass running behaviour when using a spring that is equivalent to the three-segment pantograph leg.

The solution for given initial conditions ($v = 1.04$ m/s, $m = 0.5$ kg) is shown in Fig. 13 - this results in a step duration of $T = 0.09$ s and thus a frequency of 5.6 Hz.

The body bounces back a bit faster than we'd like - the knee motor will have work a bit here to prolong step duration. Alternatively using a weaker spring might be a solution - however this induces a lot of changes: the leg length variation during a stance phase increases, therefore the leg must be retracted more during the swing phase as well, which is quite hard to achieve mechanically.
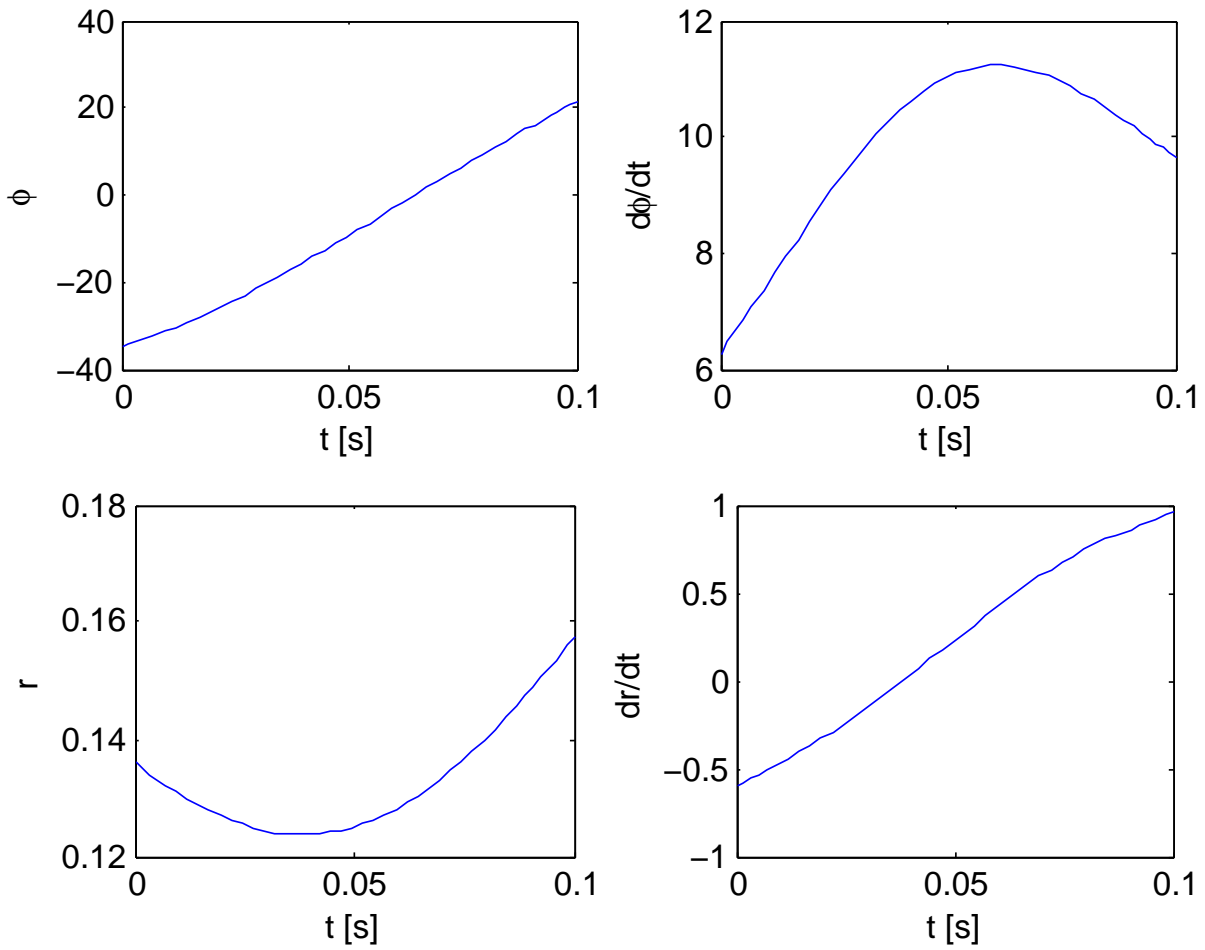
Figure 13: Plot of the leg length in a stance cycle

6 Dimensioning and mechanical construction

6.1 Motor dimensioning

The spring length change amounts to $s_n = 13.6 \ mm$ (Eq. 33) which thus also is the length over which the motor must be able to displace the bowden cable.

6.1.1 First dimensioning approach
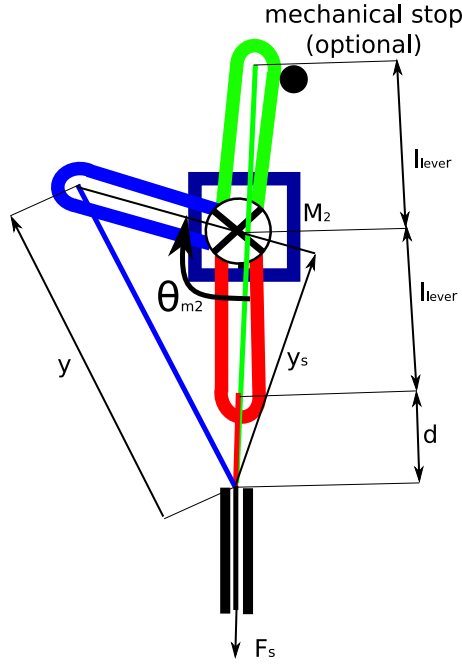
Figure 14: Hardware sketch showing motor M_2, lever and bowden cable

Assuming that the motor should be in an equilibrium situation both during swing and stance phase (the most plausible alternative being continuous operation in one rotation direction) the lever has necessarily a length of $l_{lever} = \frac{s_n}{2} \approx 7 \ mm$. Based on simple geometrical calculations, the required motor torque can be calculated - all the lengths are named in Fig. 14.

The spring force is

$$F_s = k \cdot (y(\Theta_{M2}) - d)$$

where $y - d$ is the spring length change and the length y is given by

$$y^2 = l_{lever}^2 + (l_{lever} + d)^2 - 2l_{lever}(l_{lever} + d) \cos \Theta_{M2}$$

and

$$y_s = (l_{lever} + d) \cdot \sin \Theta_{M2}$$

Thus the projection of F_s on the normal to the lever amounts to

$$F_{s,\perp} = F_s \frac{y_s}{y}$$

which means that the required motor moment is

$$M_{M2} = l \cdot F_{s,\perp} = l \cdot F_s \frac{y_s}{y}$$

Solving these equations for a given distance $d = 10\ mm$ gives a graph like Fig. 15. In order to find the optimal distance d, the maximal value for different distances $d \in [0\ 50]\ mm$ is plotted in Fig. 16.

In any case the distance d does not strongly influence the required motor moment - it amounts to

$$M_{motor} = 0.13\ Nm \tag{47}$$
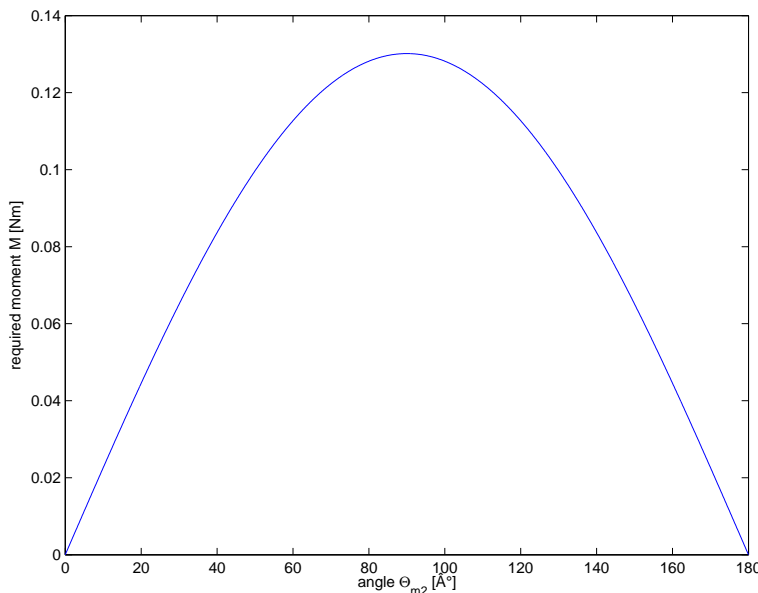
Figure 15: Plot of motor moment depending in function of angle for $d = 10\ mm$

In addition the dynamic requirements (inertia and time) should be considered. However the total leg mass is very small - less than $50\ g$ and the acceleration does not have to be very high, so the dynamic properties of the system may be neglected; even more so since the spring contraction force in the beginning when the system has to be accelerated is very small.

The demanded torque can easily be delivered by a Dynamixel AX-12. However the angular velocity is a problem since the Kondo KRS-2350 ICS used as hip motor provides a angular of $0.16s/60°$. During one step cycle this motor only sweeps twice an angle of $70°$ - in contrast the knee motor has to move twice from $0°$ to $180°$ and

back, the required velocity is 5.1 times as high: $0.031s/60°$ (see Fig. 18 for a sketch of the motor trajectories over time).

This is not possible with common servos, since they all have a rather high reduction factor and therefore a low maximum velocity - possible solutions to this problem are:

- instead of using a Kondo or Dynamixel servo use a motor with a custom-made gear and a reduction ratio well fit for the problem

- use a Kondo or Dynamixel servo but add a reductor with reduction factor $r < 1$ at the output - this is quite obviously everything but optimal

- use a servo but with a longer lever arm - the servo moves only by about $30°$ to completely contract the spring

The first option has the advantage that an optimal motor-gear combination can be selected; however this requires a lot of work for the gear design, a motor controller has to be implemented manually and the supply voltage of the motor might differ from the Kondo servos. This choice would increase the complexity of the electric and mechanical system.

In this context the third option is the most favourable: The same Kondo servo motor will be used as for the hip, using a long lever arm. The advantage of this choice is that the command signals and the required supply voltage are exactly the same as for the hip motors.
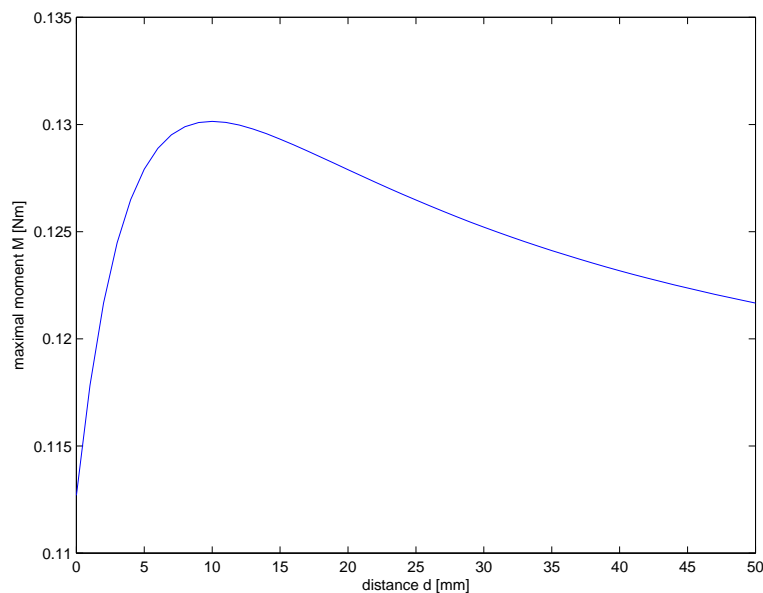
Figure 16: Plot of maximum required motor moment in function of distance d

6.1.2 Second dimensioning approach

As mentioned before, a long lever arm will be used. Since it is simplest to use the same command signals everywhere and the $360°$ capabilities of the Dynamixel motors are not required, Kondo KRS-2350 motors will be used.

Since using cables without cladding (for instance over distance d in Fig. 14) do not provide a relevant advantage with respect to the required motor torque, a configuration with $d = 0$ (definition of d: Fig. 14) such as shown in Fig. 17 will be used. This has the advantage of greater robustness - even if someone touches the cables, the system is not disturbed.
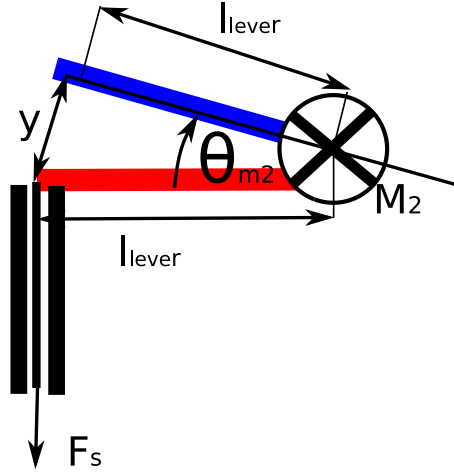
Figure 17: Hardware sketch showing motor m_2, long lever and bowden cable

The relation between the given geometric variables is:

$$\frac{y}{2} = l_{lever} \cdot \sin(\frac{\alpha}{2}) \tag{48}$$

Assuming that the motor provides a torque of $M_{motor} = 1\ Nm$ (the Kondo motor can provide up to $2\ Nm$), the required length l_{lever} is (Eq. 34) [5]

$$l_{lever} = \frac{M_{motor}}{F_{retract}} = \frac{1\ Nm}{31.6\ N} \approx 32\ mm \tag{49}$$

and the angle the motor has to move amounts to ($y = s_n$ from Tab. 4)

$$\Delta\theta_{M2} = 2 \cdot \arcsin(\frac{y}{2l_{lever}}) = 2 \cdot \arcsin(\frac{13.6\ mm}{2 \cdot 32\ mm}) = 24.5° \tag{50}$$

A sketch of the trajectories for motor M_1 and M_2 (Fig. 4) is shown in Fig. 18. The limiting factors (as far as one can tell from the given motor data) is the velocity of

---

[5]Note that at this point, the friction forces in the cable had not yet been estimated - they were simply assumed to be negligible, the force having a torque twice as large as required.

motor M_1, it defines the upper limit for the step cycle time T to (see also Eq. 21 and 19)

$$T_{step} = \frac{0.16\ s}{60°} \cdot 2 \cdot \alpha_{stance} = \frac{1}{f_{max,kondo}} = 0.37\ s \tag{51}$$

Motor M_2 has to move by (Eq. 50) $24.5°$ - the minimal time required for this movement is:

$$T_{\Delta\theta_{M2}} = \frac{0.16\ s}{60°} \cdot \Delta\theta_{M2} = 65\ ms < \frac{T}{4} = 92\ ms \tag{52}$$

It seems that the limiting factor is motor M_1 - however it has to be noted that motor M_2 has to accelerate and decelerate four twice as often as M_1 so it is a good thing to have some margin for M_2.
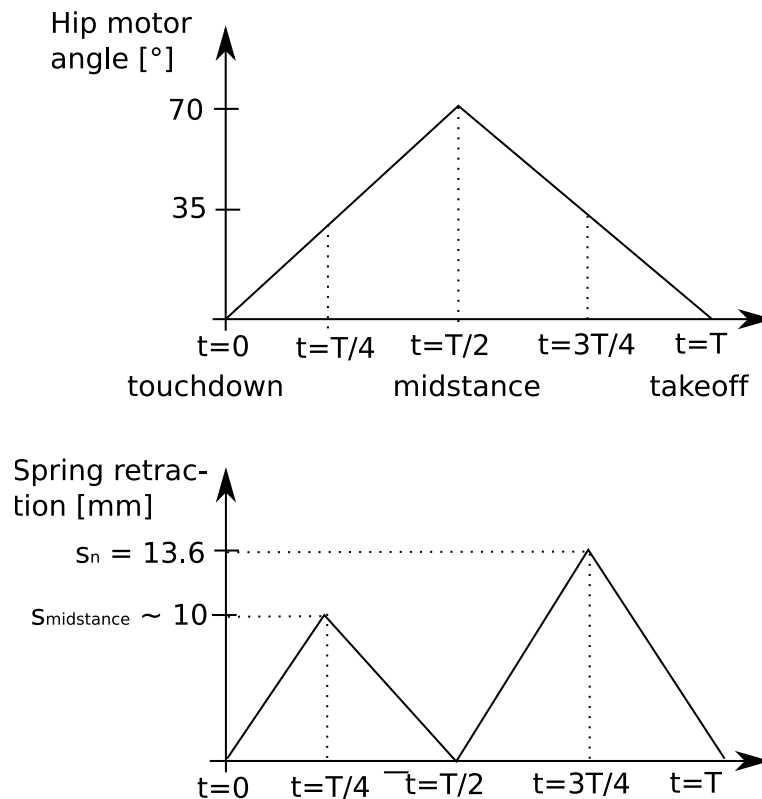
Figure 18: Hardware sketch showing the trajectories of motor M_1 and M_2 over time

Modifications

After having constructed the first prototype it turned out that the torque of the Kondo KRS-2350 ICS motors was not sufficient to contract the legs. Once the leg was contracted it could, however, keep it in that position.

There are several reasons for this problem: It turned out that the torque given for the motor was the stall torque. Once the motor turns, the torque decreases linearly - at usable velocities it is in the order of four times lower.

In addition, the effect of friction in the wire guiding cable was completely neglected - an approximate calculation (section 5.9) shows that friction increases the required by around 50%, it is not easily neglected.

In consequence to these problems, the servo horn was replaced by a much shorter one - leading to a decrease in required torque and an increase in movement angle. The motor now moves by about $120°$ to completely contract the leg.

6.2 Mechanical design

6.2.1 Fr4 Material properties

Fr4 has approximately the mechanical properties shown in Tab. 5 (exact values depend on the fabricate, the material is anisotropic).

Density	$\rho = 1.9 \cdot 10^3 \frac{kg}{m^3}$
Yield strength	$\sigma_{yield} = 276\ MPa$
Young's modulus	$E = 17\ GPa$
Flexural modulus	$G = 16.5\ GPa$
Poisson's Ratio	$\mu = 0.118$

Table 5: Fr4 material properties

6.2.2 Material thickness and width

In order to reduce bending along the z-axis (Fig. 2) the strongest available material thickness $e = 3.2\ mm$ will be used - this is also useful since it increases the length of the plain bearings used to connect the leg segments.

The width of segment 1 and 3 (Fig. 2) was selected to be $w = 15\ mm$ - this is quite a lot and could probably be reduced. However this would require a careful stress analysis, knowing the amplitude of the shocks that act on the leg, which is hard to estimate.

Segment 2 and 4 are a bit smaller $w_{1,4} = 10\ mm$, since at this point two segments work in parallel - though sometimes segment 2 is under compression while segment 4 is under traction.

The distance between segment 2 and 4 was chosen arbitrarily (Fig 2) $d_{14} = 12\ mm$. From a mechanical point of view it would be interesting to increase this distance in order to reduce the stresses - however increasing it too much would leave no space for the foot.

Already with the current design, the relative leg segment length of $\lambda_{01} = 0.14$ which corresponds to $l_{01} = 21\ mm$ had to be increased to satisfy space constraints.

6.2.3 Buckling, Torsion and Bending

For the following approximate estimations the leg is assumed to be a straight bar made of Fr4 with thickness $e_{bar} = 3.2\ mm$, width of $w_{bar} = 15\ mm$ and a length of $l_{bar} = 180\ mm$ (Eq. 11). The joints are neglected.

Buckling

The critical buckling load for a column (both ends are hinged $K = 1$) is:

$$F = \frac{\pi^2 E I}{(Kl)^2} = \frac{\pi^2 17 \cdot 10^9 \cdot 4.096 \cdot 10^{-11}}{3.24 \cdot 10^{-2}} = 212\ N \tag{53}$$

where the area moment of inertia is

$$I = \frac{BH^3}{12} = \frac{w_{bar} \cdot e_{bar}^3}{12} = 4.096 \cdot 10^{-11}\ m^4 \tag{54}$$

Such a high load might be achieved at touchdown under some conditions - but only during very short periods of time.

Torsion and Bending

Considering torsion, bending and other simple load cases of the legs is of limited use at this point, since the most probable source of lacking stability or rigidity are the joints between the leg segments and not the segments themselves.

6.2.4 Joints and Bearings

Several types of bearings could be used to link the leg segments:

Bearing type	Advantages	Disadvantages
Thin flexion blade/Leaf spring	no friction (but eventually hysteresis depending on the material)	limited flexion angle
	lightweight	not purely an angular joint if too long
	easy to integrate	limited stability under torsion
	intrinsic elasticity	
Ball bearing	very low friction	hard to integrate in system, results in complicated and heavy construction
	360° freedom	rather heavyweight itself
		low resistance to shocks
Plain bearing	easy to integrate in system	more friction than the other two systems
	360° freedom	
	cheap, can be made of standard parts	
	robust	

Using the simplest solution - plain bearings - seems advantageous in this case. Leaf springs don't support a sufficiently large angular displacement while remaining short, while a ball bearing construction tends to get heavy - a non neglectable disadvantage since each leg requires at least 4 bearings.

The bearings will be made of brass and are fixed in the leg segments 1 and 3, while the axes are made of steel and fixed in leg segments 2 and 4, as recommended by André Guignard.

6.2.5 Sensor choice

Two different sensors are required for each leg: an angular sensor at one of the bearings to measure the angle φ_{14} (Fig. 3) and a force sensor in the foot.

Angular sensor

Different types of angular sensors and their advantages and disadvantages are listed in Tab. 6. Both Potentiometers and Hall sensors are good choices for this application.

It is intended to use a potentiometer due to the low price and because the potentiometer sensors used on Puppy II worked reliably, according to Jonas Buchli. Mechanical lifetime might turn out to be a problem - however the Piher PT-10 LV potentiometer that is going to be used exists also in a long-life version, guaranteeing 100'000 mechanical cycles.

Angular sensor type	Advantages	Disadvantages
Potentiometer	high resolution	friction
	gives absolute position values	limited mechanical lifetime
	no initialisation required	initial calibration required
	price	
Optical encoder	high resolution	very expensive
	no friction	only incremental information
Hall angular sensors	no friction	expensive
	absolute position information	rather low resolution (though interpolation is common)

Table 6: Different angular sensor types

Foot force sensor and toe design

Finding a good foot contact and force sensor is more difficult. Several completely different sensor types can be imagined - depending on the foot design Tab. 7.

All of the mentioned solutions have their disadvantages: The point force sensor, such as used in Puppy II does not work reliably; the infrared sensor is heavily dependant

Sensor type	Foot design/sensor mounting	Advantages	Disadvantages
Angular sensor	requires a rotational bearing between leg segments 0 and 1, including an elastic element	continuous output	constrains foot design
			output not proportional to the applied force (depending on stance angle)
Point force sensor	similar as in Puppy II, on the lower side of the foot with direct ground contact	binary/continuous output	not reliable - signal only at some stance angles
			high mechanical strain on sensor
Infrared sensor	mounted pointing to the floor	price	binary output
		does not heavily constrain foot design	sensitive to floor material; may require adjustments
			does not give actual contact information but distance
Strain gauge	mounted on a flat spring	continuous output	signal requires amplification/noise problems
		small and lightweight	constrains foot design
			temperature sensitivity
Distance sensor	measures deformation of a flat spring	continuous output	high resolution signal required, thus expensive
			constrains foot design

Table 7: Different force sensor types and measurement principles

on ambient light and floor material and gives only a binary output signal. Both strain gauges and distance measuring to determine the strain in a flat spring are critical with respect to noise and required resolution.

An angular sensor in combination with a helical torsion spring and a plain bearing was finally chosen. The only fundamental problems of this solution is the fact that the measured output depends on the current leg angle. The angle will be measured by the same potentiometer used to measure the leg angle.

This choice in practice also determines the toe design (see drawings "klaue") - it

is rounded and made of Fr4, coupled to leg segment l1 by a rotational joint. The elasticity is provided by a helical torsional spring. A consequence of this joint is that the length of foot and toe l_{01} has to be slightly increased in order to create enough space for toe and potentiometer.

6.2.6 Toe helical torsion spring dimensioning

For a helical torsion spring the angular displacement is given by [9]

$$\theta = \frac{64MnD_m}{Ed^4} \tag{55}$$

and the maximal stress is

$$\sigma_{max} = \frac{32M}{\pi d^3} q \tag{56}$$

where M is the applied moment, n the number of turns, D_m the mean spring diameter, d the wire diameter and $q = 1.19$ is a constant depending on $\frac{D_m}{d}$ [9].

   Both the choice of n and D_m are quite limited by geometrical constraints - the more turns there are the less length is available for the bearing, D_m can't be increased infinitely either since it's limited by the foot size.

   Assuming that applied moment is $M = 0.06 \ Nm$ (based on the applied force at midstance and the mean lever length given by the foot geometry)[6], $n = 3$, $D_m = 6 \ mm$, $d = 0.9 \ mm$, $E = 210 \ GPa$ an angle of

$$\theta = \frac{64 \cdot 0.06 \cdot 3 \cdot 6 \cdot 10^{-3}}{210 \cdot 10^9 (0.9 \cdot 10^{-3})^4} \frac{180}{\pi} = 28.9° \tag{57}$$

and a stress of

$$\sigma_{max} = \frac{32 \ 0.06}{\pi (0.9 \cdot 10^{-3})^3} \approx 1 GPa \tag{58}$$

The angular displacement corresponds to what is required; the stress is quite high even for spring steel, but then again the assumed moment is an overestimation of what can be expected in reality.

---

[6]A force of about $F = 12$ N and a lever length of $l = 5$ mm, based on the foot geometry, are assumed - this results in a torque of $M = F \cdot l = 12 \cdot 5 \cdot 10^{-3} = 0.06$ Nm

7 Test electronics

In order to be able to test the system without building a complex setup of autonomous electronics, battery power supply, communications and sensory feedback as planned for the future, a simpler setup was used.

Fig. 19 shows the schematics of the test electronics.

At the first sight the servo angle setpoint update bandwidth seems to be limited by the serial bus - a data rate of 115 kpbs with one byte per setpoint and eight servos to command results in a refresh rate of 1.8 kHz and a setpoint refresh cycle time of $0.5\ ms$.

The cycle time of the PWM signals to be generated by the PIC microcontroller is at least $8\ ms$ according to the datasheets. So this is what limits the setpoint updates to about $125Hz$ or 50 setpoints per step when running at $f_{step} = 2.3\ Hz$.
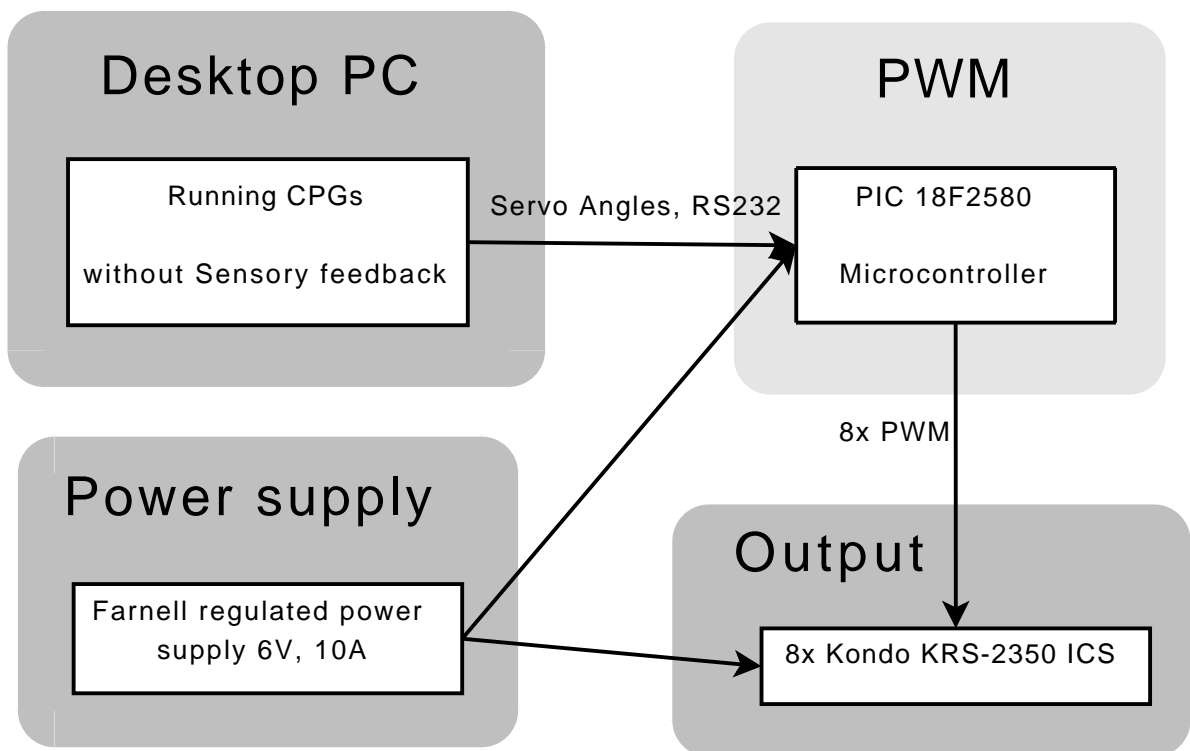

Figure 19: Schematics of test electronics

7.1 PWM

The PWM controller has to generate eight independent PWM signals. The cycle time is $8000\ \mu s$. The pulse width can vary between $700\ \mu s$ and $2400\ \mu s$ (without considering the special modes supported by the servo).

A PIC 18F2580 testboard made on 24.6.2005 by Alessandro Crespi was used to generate the PWM signals. The complete code of the PWM implementation can be found in section 15.3.

7.1.1 Implementation on a PIC 18f2580 - first try

Programming the PIC 18f2580 such as to produce the eight PWM signal was in reality not as easy as it may seem in theory.

The first idea was to generate an interrupt on timer2 every $6.4\ \mu s$ and to check on every interrupt if one of the eight output channels had to be set to zero. In addition a separate counter would be maintained to ensure that every $8000\ \mu s$ all outputs are set to one and a new PWM cycle is started.

However timer2 is not able to generate an interrupt every $6.4\ \mu s$ even if set up to do so - interrupts only occur about every $18\ \mu s$.

7.1.2 Implementation on a PIC 18f2580 - second try

Since timer2 can't produce interrupts fast enough, an alternative way had to be found. Now all the servo setpoints are sorted into an array - the first element being the earliest output to be reset. Separately an array in the same order, containing the corresponding output pins, is kept[7].

Timer2 is set up as to increment every $1.6\ \mu s$ - interrupts are thus generated at least every $400\ \mu s$ which is obviously a lot lower than the $8000\ \mu s$ cycle time. Therefore a separate 16 bit counter variable is used for timekeeping, comparing to the setpoint-time array. This counter is also for calculating the time until the next interrupt, which is adapted in function of the next event - $400\ \mu s$ if nothing is to happen within this timespan, less if one of the outputs has to be set to zero before.

Note that also with this implementation one might run into the timer2-bug as seen with the first trial implementation: If two servos require for instance a pulse of $700\ \mu s$ and $707\ \mu s$ a timer interrupt will occur at $700\ \mu s$. At this time the next interrupt is scheduled to be within $7\ \mu s$ which is faster than what timer2 can do in reality - thus the next interrupt will happen only after at least $20\ \mu s$ which will result in a systematic delay on all later setpoints.

In order to avoid this problem, setpoints that are that close in time to each other are considered to be at the same time. Both PWMs would output a cycle time of $7000\ \mu s$. One might also consider using active waiting during timespans lower than $20\ \mu s$, but such high precision doesn't seem necessary in this case.

---

[7]Why not use one single array containing a structure? Well, the compiler allows doing so but the binary it produces does not work at all as it should. This is why a specifically adapted quicksort and two arrays is used - one array contains the setpoints and the other the output pin addresses. Both are sorted at the same time according to the data in the first one.

7.2 RS232 Communication Protocol

The setpoints are sent over a serial bus to the PWM controller. The bus settings are listed in Tab. 8 whereas the protocol is described in Tab. 9.

Baud rate	115200 bps
Data bits	8
Flow control	Off
Parity bits	None
Stop bits	1

Table 8: RS232 Bus settings

255	Reset - the next value will be the setpoint for servo nr. 1
252-4	Change operation mode of Kondo motor (exact meaning unknown)
251	Free mode - servos apply no torque
0-250	RC servo motor position 0 corresponds to the extreme position in counter-clockwise direction, 250 corresponds to the extreme position in clockwise direction

Table 9: PWM RS232 communication commands

For example send the sequence in Tab. 10 to first set all the eight servos to the extreme position in counter-clockwise direction, then immediately afterwards - in the next PWM cycle - to the extreme position in clockwise direction:

Sequence part 1	255	0	0	0	0	0	0	0	0
Corresponding motor		FLH	FLK	FRH	FRK	HLH	HLK	HRH	HLK
Sequence part 2	[255]	250	250	250	250	250	250	250	250
Corresponding motor		FLH	FLK	FRH	FRK	HLH	HLK	HRH	HLK

Table 10: PWM RS232 communication commands

As listed in the table, the first value transferred is the setpoint for the FLH motor (Front Left Hip), the second is for the FLK motor (Front Left Knee) and so - the last one is for the HLK (Hind Left Knee) motor.

Note that sending the reset-signal [255] between the two sequences shown in Tab. 10 is optional.

Hard limits on the setpoint values are enforced by the PWM software on the PIC microcontroller for the knee motors, in order to avoid damaging. The exact values have to be found experimentally and vary slightly from one leg to the next.

7.3 Power supply board

As power supply a Farnell lab power supply was used - it should provide a current of 10 A at 6 V. In order to distribute the power to the motors, a simple PCB board was developed - thanks to Alessandro Crespi. It's schematics and connections are explained in the Annex (Section 14).

7.4 Command on the PC

In order to command the PWM board over the serial line, a simple Qt application was developed, featuring eight sliders to allow controlling the motors manually. This application is also included on the CD-ROM that was handed in with this report.

8   Designated autonomous electronics

8.1   Overview

Fig. 20 shows the schematics of designated autonomous electronics.
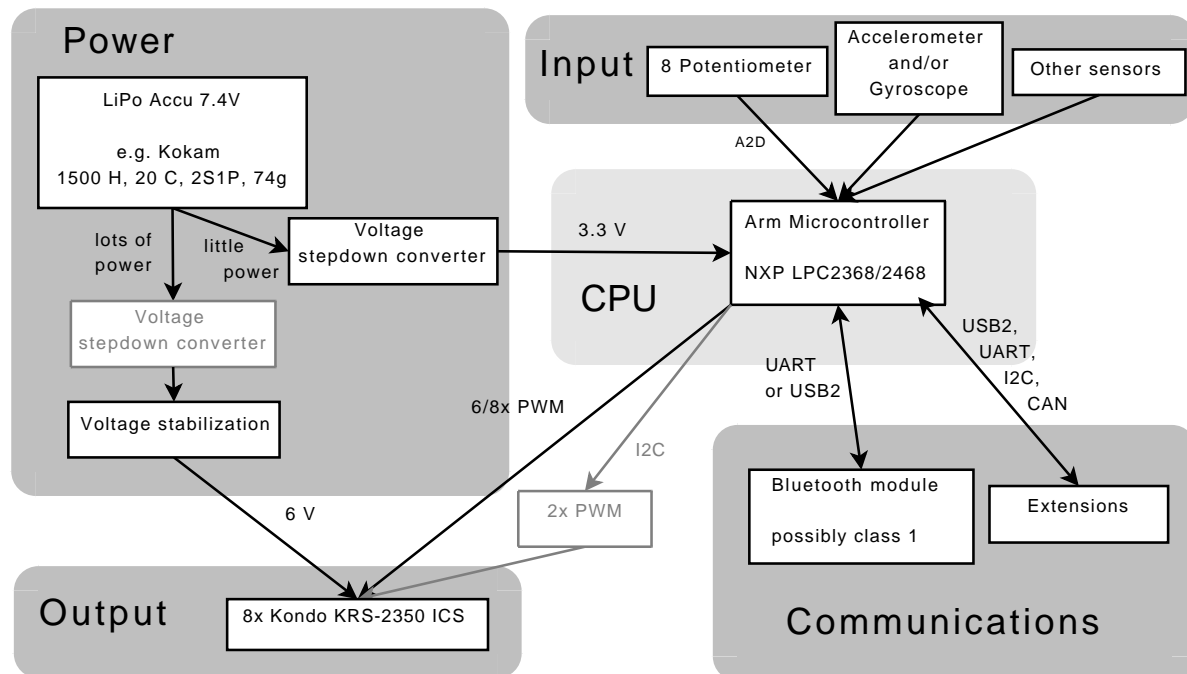

Figure 20: Schematics of designated autonomous electronics

8.2   Microcontroller

The system can be designed using one or multiple microcontrollers. With regard to communications overhead, system complexity and the different programming setups the easiest way is probably using a single microcontroller fro running the CPGs and treating sensory information.

The requirements are the following:

- "fast enough" to run four CPGs and functions for the hip-motors in realtime

- low power consumption (not very important, considering how much current the eight motors draw)

- one single supply voltage

- possibly eight PWM outputs (else one or several would have to be realised in software)

- RS232, I^2C and USB communication channels (CAN, Ethernet and others are nice to have too)

- one fast or several ADC fo treating sensor information

Using a microcontroller with ARM architecture seems to be the most straightforward solution. For instance an NXP LPC2468 or a Atmel AT91SAM7A3 - both fulfill all of the above stated requirements, as listed in Tab. 11.

Other manufacturers offer comparable products. However it turns out that the most critical feature are the required eight PWM outputs - only few ARM microcontrollers provide that many independant PWM outputs.

There exists evaluation boards featuring a NXP LPC2468 ARM microcontroller [4]. This might be an interesting option, but it does not contain any form of wireless communication support (Bluetooth, WLan or other).

8.3 Power supply

Due to the very favourable energy per weight ratio (around 130 Wh/kg) Lithium Polymer batteries seem the most interesting ones. The main problem is the low maximum current that can be drawn from most Lithium Polymer battery.

But there exist such batteries - in the model aircraft scene. Instead of the usual 2C[8], such batteries can be charged by currents up to 30C.

A maximum charge test for a Kokam $1500\ mAh$, specified by the importer as supplying up to 10C[9] has been performed [16]. It shows that such a battery can supply permanent currents up to $8.5\ A$ - somewhat more than 5C - without being damaged.

Fig. 21 shows the resulting voltage curves. The most interesting ones are the pink one (discharge at 8.5 A, 1418 mAh, temperature $45°$) which the battery survived without being damaged and the brown one (discharge at 12 A, 1275 mAh, temperature $70°$) which damaged the battery permanently.

There exist several suppliers for such high-current LiPo batteries, Kokam being the most renowned one. Considering what is known about the system - the motors should draw up to 1 A each summing up to 8 A, using a 1800 mAh 7.4 V Kokam battery pack seems to be a good choice, it should be sufficient for a runtime of fifteen to twenty minutes. The battery data is summarised in Tab. 12.

8.4 Communications

A reliable highspeed communication channel is needed to transfer sensor and actuator data for logging purposes and to transfer highlevel commands to the CPGs running onboard. Most larger ARM microcontrollers support ethernet and USB - these are viable options for highspeed communication.

---

[8]The maximum current that can be drawn from a LiPo battery is usually given in relation to it's capacity in Ah. If a maximum current of 2C is specified, one can draw currents up to $2 \cdot 1800\ mA = 3.6\ A$ from a $1800\ mAh$ battery.

[9]Note however that the importer specifications usually differ from what the manufacturer specifies.

Microcontroller	**NXP** **LPC2368**	**NXP** **LPC2468**	**Atmel** **AT91SAM7A3**	**Atmel** **AT32UC3A....**
Architecture	ARM7ATDMI-S	ARM7	ARM7TDMI	AVR32
Clock Frequency	$\leq$ 72 MHz	$\leq$ 72 MHz	$\leq$ 60 MHz	$\leq$ 66 MHz
Internal Flash	$\leq$ 256 KByte	512 KByte	256 KByte	128/256/512 KByte
Internal SRAM	$\leq$ 32 KByte	98 KByte	32 KByte	32/64 KByte
Supply voltage	3.3 V	3.3 V	3.3 V	3.3 V
PWM Outputs	6, 32 bit	2 · 6, 32 bit	8, 20 bit	7, 16 bit
UART	4	4	3	4
SPI	1	1	2	2
CAN	2	2	2	☐
I²C	3	3	1	1
I²S	☑	☑	☑	☑
Ethernet	10/100 Mbit/s	10/100 Mbit/s	10/100 Mbit/s	10/100 Mbit/s
USB	2.0 Full Speed	2.0 Full Speed	2.0 Full Speed	2.0 Full Speed
SD/MMC Card Interface	☑	☑	☑	☑
ADC	10-bit, 6 Pin MUX	10-bit, 8 Pin MUX	10-bit, 16 Pin MUX	10-bit, 7 Pin MUX
FPU	☐	☐	☐	☐

Table 11: NXP LPC2368 and LPC2468 ARM microcontroller features

Manufacturer	Kokam
Capacity	1800 mAh
Voltage	7.4 V
Dimensions	104 x 16 x 34 mm
Weight	102 g
Constant discharge current	30C (54A)
Max. charge current	2C
Supplier	Slowflyer [2], Mamo Modelltechnik [1] and others

Table 12: Kokam 1800 mAh 2S1P battery data

However wireless communication is much more comfortable in most cases - so including a bluetooth device (possibly class 1 with up to 100 m coverage) seems a good idea. Other alternatives might be Zigbee or standard 802.11 WLan - though both have their disadvantages: Zigbee has a low data rate and WLan consumes lots of power.
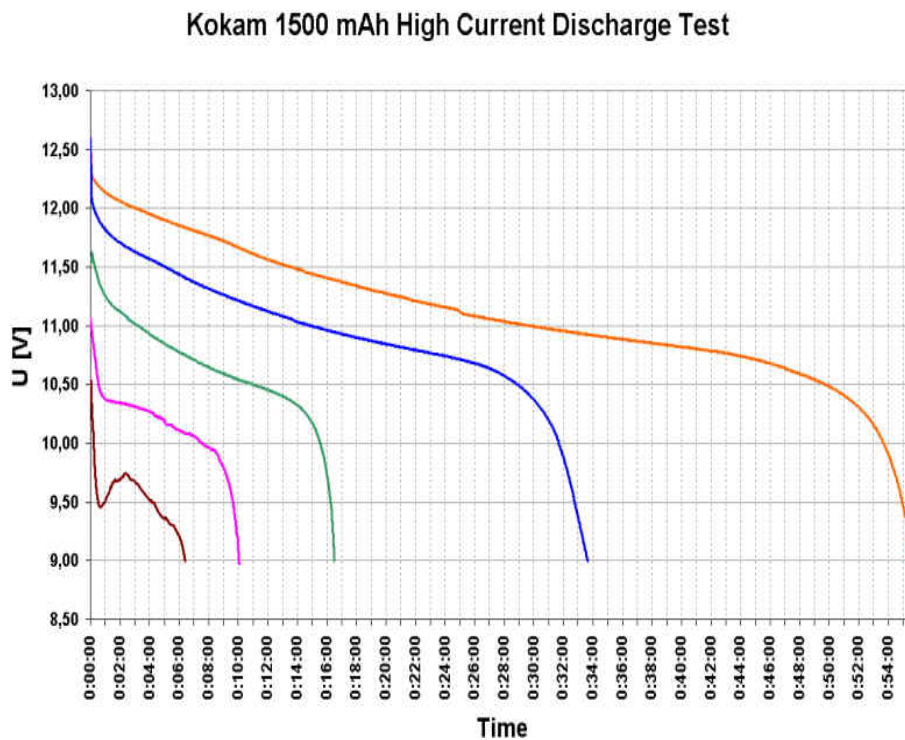
Figure 21: Kokam 1500 mAh high current discharge test curves, © Helitron [16]

9   Assembly

9.1   Overview

Fig. 22 shows all the parts (with exception of the motors and the springs) before assembly.

Figure 22: Parts before assembly

9.2   Feet

The feet are produced in two parts - these two have to be soldered together using solder paste. The alignment of the parts proved to be a difficulty in the process. Finally the hole in the foot (71) has to be re-drilled manually and the corresponding pin is pressed into the foot.

The pins turned out to be slightly too short (due to the division of the foot into two pieces it got thicker which was not properly reported to the pin length) - they had to be pressed further than planned to allow the circlip to be mounted. This has been corrected in the drawings included in the appendix.

The prismatic parts of some pins were slightly bent due to the pressing. Applying force on these parts has to be avoided - the same holds unfortunately for the chamfer (where the circlip is to be put later on).

The same procedure is applied to the second and fourth leg segments (74). Note that for these segments special care has to be taken in order to ensure that the pins are as rectangular as possible to the leg segments. Slight misalignments here increase friction quite heavily, since the leg is a pantographic one and therefore overconstrained. A completely assembled leg (without springs) should extend and contract easily under it's own weight.
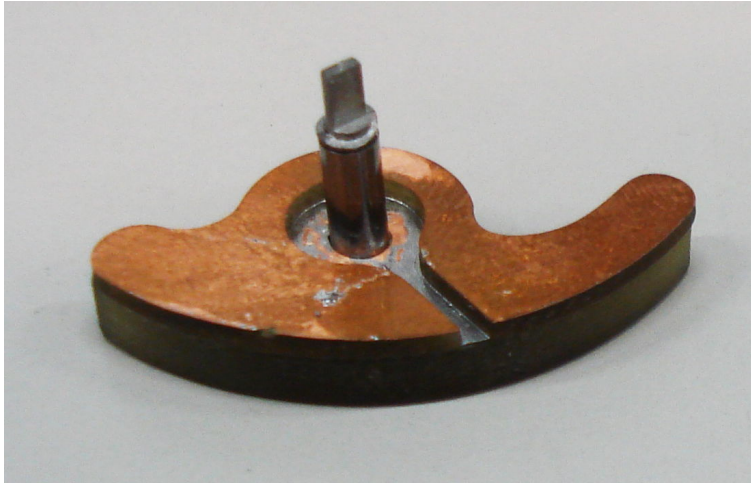

Figure 23: Completed foot assembly

9.3 Leg segments 1 and 3

The first leg segment (part number 73), the corresponding reinforcement (67) and the messing bearing (44) are assembled using a vise (Fig. 24). In addition they are glued together by hot-melt adhesive.

The same procedure is applied to assemble the third leg segment (75) and it's reinforcement (69) - there the glue is more important since the bearing where the foot and spring are to be mounted later on does not connect the two pieces (Fig. 25).

9.4 Spring assembly

Assembling the spring, tube and it's supporting Fr4 pieces is straightforward: *spring_l1* (61), the white guiding tube (46) and the *spring_washer* (49) are all glued together using hot-melt glue.

In order to increase stability in the direction perpendicular to the leg plane (z direction Fig. 4), lots of hot-melt glue are used and two small metal pins are put between the guiding tube (46) and *spring_l1* (61).

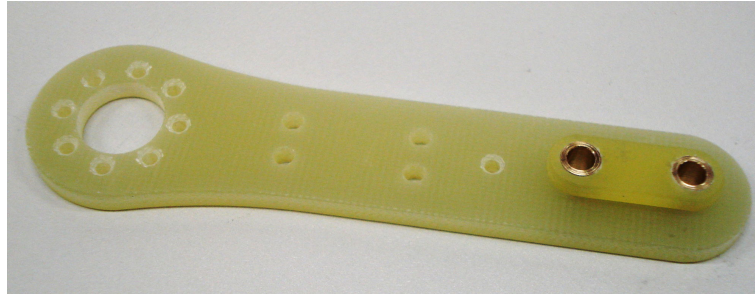The other pieces are simply put in place - the spring ensures their position.
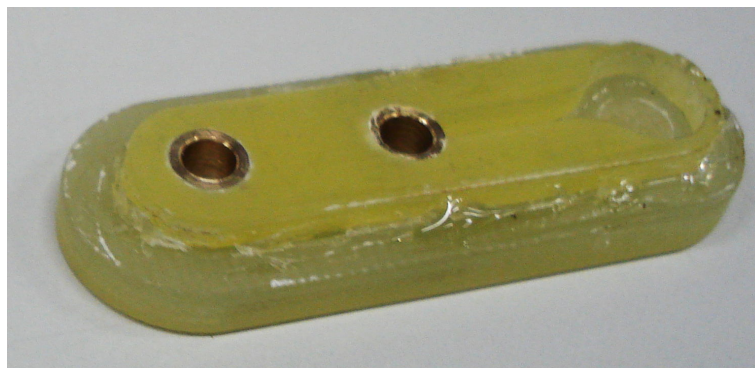
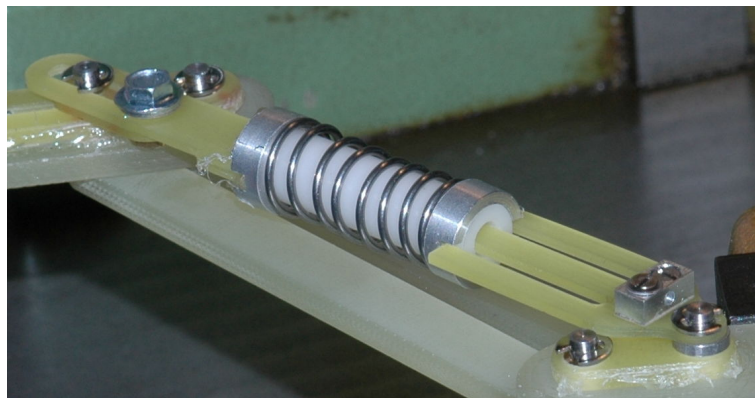Figure 24: First leg segment

Figure 25: Third leg segment

Figure 26: Spring assembly

9.5 Complete leg assembly

A complete assembled leg is shown in Fig. 27. It is important that there is enough room between leg segments two/four and one/three, such as to ensure low friction in the overconstrained pantograph mechanism. Rigidity turns out to be quite high - due to the low tolerances on the messing bearings the guiding is very stable.
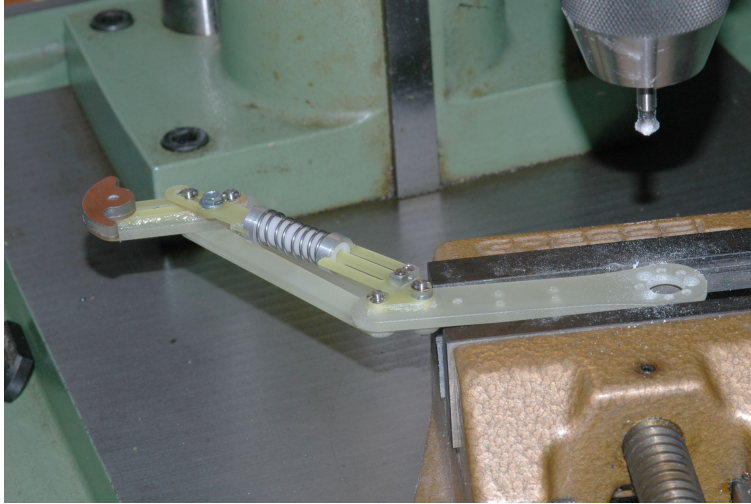

Figure 27: A complete leg

9.6 Back

The completed back assembly can be seen in Fig .28. Only two small operations apart from mounting screws were necessary:

The motor fixation blocks (21) do *not* have holes on top and bottom - these have to be drilled manually, $\varnothing = 1.5$ mm - the M1.6 screw creates the thread itself (Fig 29).

Apart from that, screwing the four servos that are mounted directly on the base plate without using fixation blocks deformed the plastic motor casings. Thus 3mm thick distance pieces made of Fr4 were added between the motors and base plate.

9.7 Bowden cables

Assembling the bowden cables is rather slow and tiresome. The best way is to first fix it on the spring assembly side, since that's the harder part (the spring assembly consists of lots of parts that are not fixed relatively to one another).

Then the wire is pulled through the guiding cable - it should be oiled or greased to reduce friction.

In order to get enough cable length to fix the cable on the servo horn it might be useful to contract the leg.
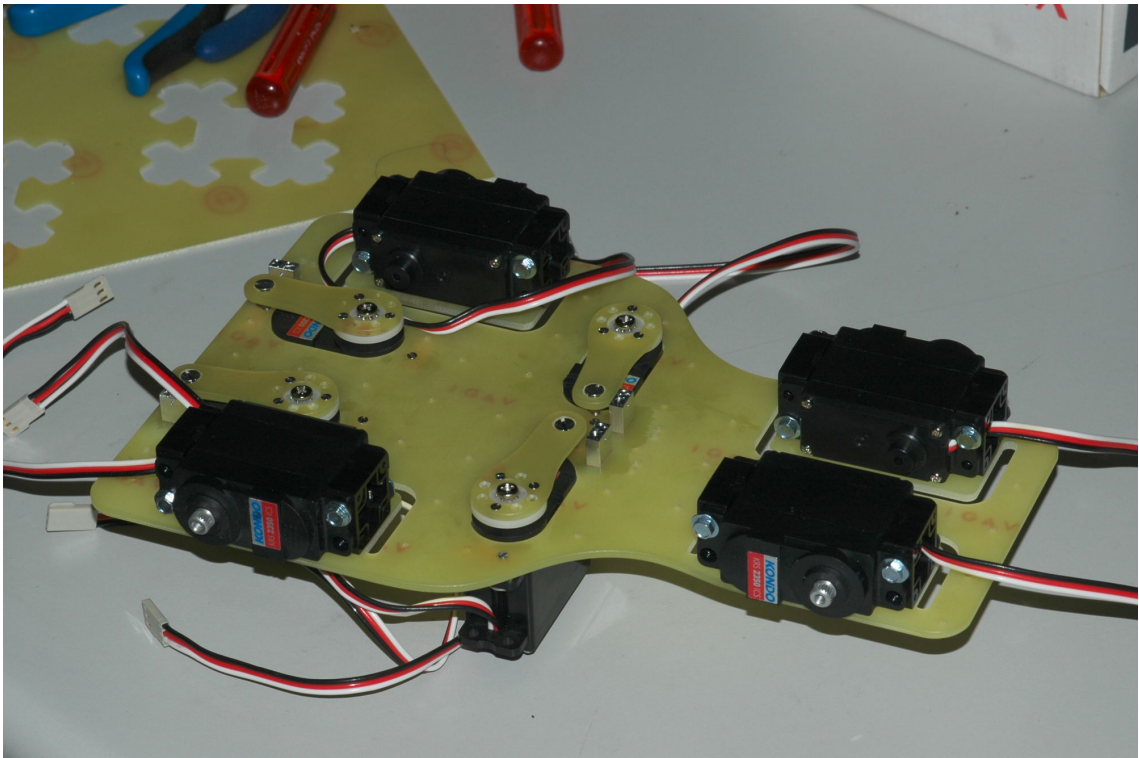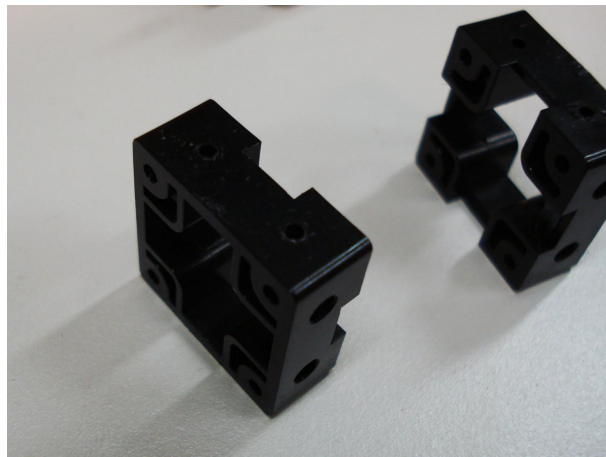
Figure 28: Back assembly

Figure 29: Motor fixation blocks

After assembling the tips of the cable should be fixated using highly viscous glue in order to ensure it's long-term stability.

9.8 Bowden Horns

During the first tests it turned out that the Kondo motors, using the long *kondo_bowden_horn* (70) servo horns, did not provide enough torque to completely contract the legs.

In a first series of tests, the supply voltage was increased up to 9 V. This led to two times stronger leg contraction - however the motors became quite warm, thus this was not an optimal solution.

Next the Fr4 bowden horns were replaced by shorter standard aluminum horns from Conrad[10]. Using these horns the motor has to turn further to contract the legs: around $120°$ instead of only $30°$. During the last part of the movement the wire is actually rolled on the servo horn axis - the very short lever arm allows the motor torque to generate lots of force.

In order to use the new shorter servo horns, the *spring_l3_bowden* bowden cable guiding spring fixations for two of the motors had to be displaced, while the two other motors had to be turned by $180°$ relatively to what was planned (Fig 30 shows the whole new arrangement).
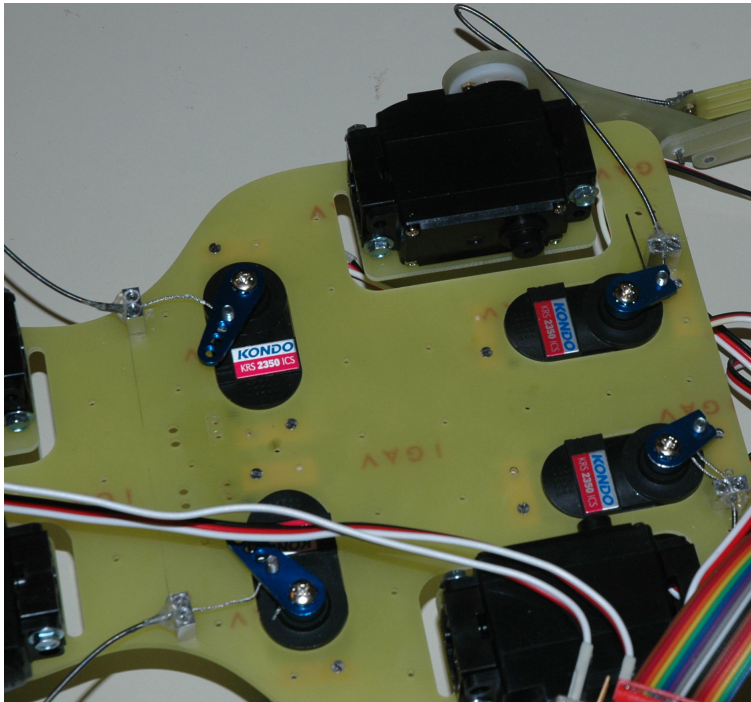

Figure 30: Bowden horns

---

[10]http://www.conrad.de

10 Functionality tests and measurements

10.1 Weights

The weights of some sub-assemblies were measured after assembly (Tab. 13) and compared to the weights as calculated by Pro/Engineer.

Part	Measure	Pro/Engineer
fore leg	21 g	34 g
hind leg	23.5 g	37 g
back with motors	630 g	670 g
batteries	102 g	
robot without batteries or electronics	720 g	741 g
complete robot (estimation) - including batteries, electronics, cables and sensors	850 g	

Table 13: Measured and calculated weights

10.2 Tests

There was not enough time to do intensive testing of the whole robot, such as trying to achieve walk, trot and other gaits. Measuring the speed that can be achieved and the precise power consumption peaks during movement are important tests that remain to be done. Especially the latter has to be known before the autonomous electronics can be designed.

But of course the basic functionalities were tested - the robot is able to start in a crouched position (Fig. 31) and stand up quickly (Fig. 32), swing it's legs, retract and extend them.
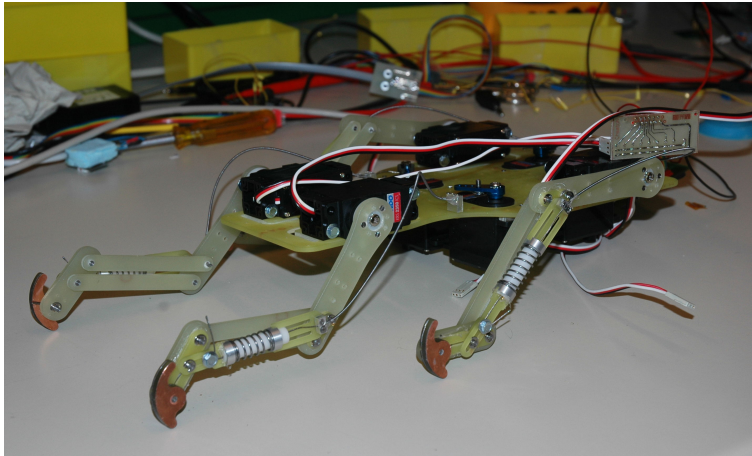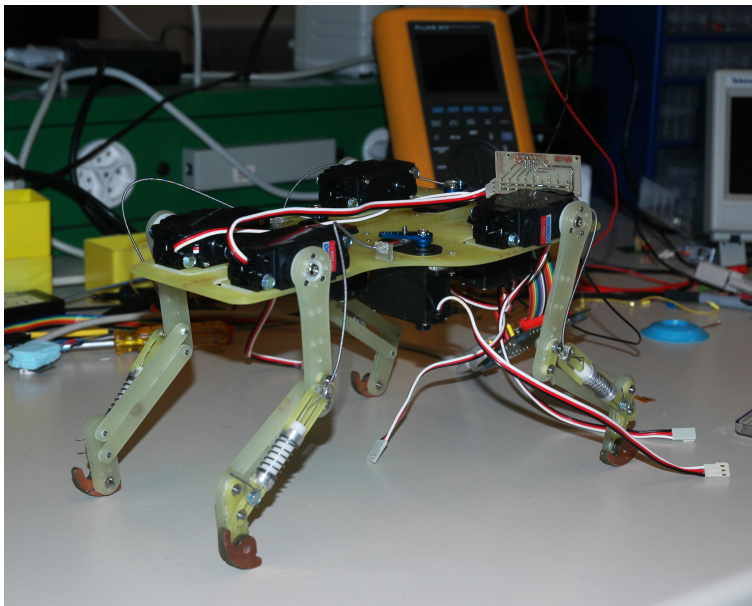
Figure 31: Cheetah - ready to stand up

Figure 32: Cheetah - standing

11 Proposed improvements

11.1 Pressing

The size of the holes is inconsistent:

- the holes for the messing bearings (44, 45) in the first and third leg segment are have the same nominal diameter as the bearings themselves. When manufacturing them they are always somewhat smaller (due to the fast use of the cutting tool by the glass fibers in Fr4)

- the holes for the pins (40-43) in the second and fourth leg segment and the foot are drilled smaller than the pins during PCB production. They have to be re-drilled to slightly less than the pin diameter (about $\frac{1}{10}$ mm)

Since the hole size in both cases corresponded to the requirements, using the first variant is preferable since it does require less manual work.

11.2 Spring assembly

As noted above the rigidity of the connection between *spring_l1* (61) and *spring_tube* (46) is not sufficient. An instant solution was found using pins and hot-melt glue. Introducing half-round pieces between the two would solve the issue definitely - though some room still has to be left for the bowden cable to pass.

11.3 Feet

Spring stiffness of the foot spring is way too high. It could be that the calculations largely overestimated the external forces on the feet - this would have to be simulated to get more complete information.

Assuming the calculations were not completely wrong, there still remain other sources of error: It is very hard to actually manufacture these springs with precision, but already small deviations create huge differences in force. In addition it was noted during assembly that the springs fit exactly into the holes (for some even force had to be applied to get them mounted) - some play there might help in decreasing the forces.

It was noted that the alignment precision between the two parts constituting a foot (*klaue* and *klaue_spring*) was not optimal. In order to ease the soldering process and alignment, introducing two small pins passing through both these parts might be a good idea.

11.4 Bowden cables

There was observed quite a lot of friction in one of the cables - making the wire guiding springs slightly larger would help avoiding such problems. It also remains to be seen if the cable is strong enough to withstand fatigue and wear.

The fixation of the cables is not optimal either - they are held by screws on both ends. This setup is not easy to mount - using the current system to fix one end and some kind of quick-mount system on the other end[11].

The fixation of the wire guiding spring on the back *back_bowden* (47) appears rather weak. Two M1.6 screws are enough to hold the fixation in place, but this creates quite some strain on the back itself. Modifying the fixation such that the two screws are not mounted in a plane perpendicular to the cable, but rather in a plane along the cable would help to reduce the stresses.

11.5 Servo horns

As mentioned in section 13 the custom-made Fr4 servo horns have to be replaced by shorter ones (Fig. 30). The replacement are aluminum servo horns from Conrad (part number 226636 - no longer available; part 233159-62 looks very similar).

The reason for this replacement is the insufficient motor torque: The given torque value is apparently the stall torque, though this is not mentioned in the documentation. At nominal velocity the torque is significantly lower. Additionally, friction in the wire guiding cables was neglected, since it is very difficult to estimate (Section 6.1.2).

11.6 Back compliance

It remains to be verified if the degree of compliance that the current back provides should be increased, decreased or be kept as it is. Varying compliance in the frontal plane is quite simple: If the point of connection of the hip servo motors to the rest of the back is moved towards the sagittal plane, compliance increases. If the thickness of the connections or of the back plate is increased, compliance decreases.

An interesting idea might also be to increase compliance in the sagittal plane, where the robot is not very compliant at the moment.

---

[11]Looking at bicycles, one cable end is a metal clod. For bicycles this end is mounted first - for the Cheetah robot a similar system could be used. Mounting the other (screw-mounted) end first is a possibility, the leg spring would ensure that the cable stays in position.

12 Conclusion

Before starting to design the hardware a detailed objective list was created, to state clearly the required and desired features of the leg: active compliant knee joint, low weight, possibly modular design and manufacturability using the means at disposal.

In a following step, several actuation principles for pantographic three-segment legs were drafted and compared. Using a pantographic design is advantageous since it completely avoids structural instability [7,24] without deviating significantly from the leg segment configurations observed in nature [12]. The most promising design - using a compression spring to extend the leg and a bowden cable to contract it during flight phase - was investigated further.

The leg segment lengths were chosen such as to be energetically advantageous [7]. In order to dimension the leg stiffness and actuators, a simple spring-mass model for walking and running was introduced [13] and solved for the given leg dimensions and estimated stroke frequencies for different gaits [3]. These calculations permitted an estimation of the leg length during a complete stance cycle - thus defining the required leg length range - and the forces in play.

In the following the actual hardware was designed. It was found that both hip and knee joints should best be actuating using the same Kondo KRS-2350 ICS motors. Plain bearings are used as joints, since they are simple to manufacture but still provide good performance and shock resistance.

Simple test electronics were designed using a PIC test board an a simple custom-made power supply board. Ideas for more powerful autonomous electronics and power supply have been drafted as well.

Finally all the parts were assembled and the leg was tested.

Designing hardware is always a difficult task. Lots of assumptions have to be made, some of which can be verified only after a prototype is built while others can't be verified at all. This is put simply why the prototype "Cheetah" is not without flaws. Section 11 mentions some of the obvious improvements - especially the foot design needs further work.

Most other things however, work very well: The pantographic leg itself and the passive compliant knee joint.

Looking back at the task description in section 2 it can be concluded that, as far as tests and verification have progressed up to now, the mentioned goals have been met.

A working lightweight, three-segment leg with two actuated joints (the hip joint and the knee joint) has been designed. Whether different gaits can be achieved has not yet been verified, due to lack of time. The weight of the system is estimated to 850 g including batteries, electronics, sensors and cables - 150 g less than the specified upper limit.

Force- and other sensors have not yet been included or tested, but the leg is prepared for mounting potentiometers both at the knee and at the foot.

There remains a lot of work to be done in continuation:

Implementation a CPG to test different gaits, measure of energy consumption, improvement and optimisation of the mechanical design through use of simulation software, implementation of battery-powered autonomous electronics, introduction of sensors and sensory feedback just to mention some topics.

Lausanne, January 11, 2008

References

[1] Mamo modelltechnik. `http://www.mamo-modelltechnik.com`.

[2] Slowflyer. `http://www.slowflyer.ch`.

[3] R. McN. Alexander. The Gaits of Bipedal and Quadrupedal Animals. *The International Journal of Robotics Research*, 3(2):49–59, 1984.

[4] Embedded Artists. Lpc2468 oem board with oem base board basic.

[5] AA Biewener. Locomotory stresses in the limb bones of two small mammals: the ground squirrel and chipmunk. *J Exp Biol*, 103(1):131–154, 1983.

[6] R. Blickhan and Martin S. Fischer. The tri-segmented limbs of therian mammals: kinematics, dynamics, and self-stabilization: a review. *Journal of Experimental Zoology Part A: Comparative Experimental Biology*, 305A(11):935–952, 2006.

[7] R. Blickhan, A. Seyfarth, H. Geyer, S. Grimmer, H. Wagner, and M. Günther. Intelligence by mechanics. *Philosophical Transactions of the Royal Society*, 365:199–220, 2007.

[8] J. Buchli, F. Iida, and A.J. Ijspeert. Finding resonance: Adaptive frequency oscillators for dynamic legged locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3903–3909. IEEE, 2006.

[9] Reymond Clavel. *Composants de la microtechnique*. Polycopié. EPFL, 2005.

[10] Lisa M. Day and Bruce C. Jayne. Interspecific scaling of the morphology and posture of the limbs during the locomotion of cats (Felidae). *J Exp Biol*, 210(4):642–654, 2007.

[11] CT Farley, J Glasheen, and TA McMahon. Running springs: speed and animal size. *J Exp Biol*, 185(1):71–86, 1993.

[12] Martin S. Fischer, Nadja Schilling, Manuela Schmidt, Dieter Haarhaus, and Hartmut Witte. Basic limb kinematics of small therian mammals. *J Exp Biol*, 205(9):1315–1338, 2002.

[13] H. Geyer, A. Seyfarth, and R. Blickhan. Spring-mass running: simple approximate solution and application to gait stability. *J. Theor. Biol.*, 232(3):315–328, 2004.

[14] Martin Golubitsky and Ian Stewart. *From Equilibrium to Chaos in Phase Space and Physical Space: The Symmetry Perspective*. Birkhäuser Verlag, 2000.

[15] Michael Günther, Valentin Keppler, André Seyfarth, and Reinhard Blickhan. Human leg design: optimal axial alignment under constraints. *Journal of Mathematical Biology*, 48(6):623–646, 2004.

[16] Helitron. Lipo: Lithium-polymer akkus der hochlast-test. `http://flyheli.helitron.eu/lipo_last.htm`.

[17] F. Iida. Cheap design approach to adaptive behavior: Walking and sensing through body dynamics. In *International Symposium on Adaptive Motion of Animals and Machines*, 2005.

[18] Fumiya Iida and Rolf Pfeifer. *Self-Stabilization and Behavioral Diversity of Embodied Adaptive Locomotion*, volume 3139 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004.

[19] Hiroshi Kimura, Yasuhiro Fukuoka, and Avis H. Cohen. The quadruped robot tekken-ii. http://www.kimura.is.uec.ac.jp/research/Quadruped/photo-movie-tekken2-e.html, October 2007.

[20] Hiroshi Kimura, Yasuhiro Fukuoka, and Avis H. Cohen. Tekken4. http://www.kimura.is.uec.ac.jp/research/Quadruped/photo-movie-tekken4-e.html, October 2007.

[21] Nadja Schilling and Alexander Petrovitch. Postnatal allometry of the skeleton in tupaia glis (scandentia: Tupaiidae) and galea musteloides (rodentia: Caviidae) - a test of the three-segment limb hypothesis. *Zoology*, 109(2):148–163, 2006.

[22] A. Seyfarth, H. Geyer, R. Blickhan, S. Lipfert, J. Rummel, Y. Minekawa, and F. Iida. *Running and walking with compliant legs*. Springer Verlag, 2006.

[23] Andre Seyfarth, Hartmut Geyer, Michael Gunther, and Reinhard Blickhan. A movement criterion for running. *Journal of Biomechanics*, 35(5):649–655, 2002.

[24] André Seyfarth, Michael Günther, and Reinhard Blickhan. Stable operation of an elastic three-segment leg. *Biological Cybernetics*, 84(5):365–382, 2001.

13 Annex I - Drawings

13.1 Part list - as used for the prototype

For norm parts shop article numbers are given:
   BAN: Bossard article number, `http://www.bossshop.ch`

Position	Quantity	Description	Material	Shop number
1	40	Screw conic head ISO 2009-M1.6x6	Ac	BAN 1246658
3	8	Screw conic head ISO 2009-M1.6x12	Ac	BAN 1246682
(1, 2)	64	Washer ISO 7089-1.6	Ac	BAN 1421417
(1, 2)	64	Hex nut ISO 4032-M1.6	Ac	BAN 1168843
4	4	Screw conic head ISO 2009-M3x8	Ac	BAN 1246739
5	4	Screw hex head ISO 4014-M3x6	Ac	BAN 1345893
(4, 5)	20	Washer ISO 7089-3	Ac	BAN 1229796
(4, 5)	8	Hex nut ISO 4032-M3	Ac	BAN 1195670
6	20	Circlip DIN 6799	Ac	BAN 1254499
20	8	Kondo KRS-2350 ICS		
21	8	Kondo KRS-2350 Fixation blocks		
22	8	Piher PT-10Lv (preferably longlife -E option)		
23	4	Compressive spring ISO 2098 $d = 1, D_m = 9.3, l_0 = 25$		
(45a-d)	4	Helical torsion spring $\varnothing_{moyen} = 6, \varnothing_{fil} = 0.9, n_{spires} = 3$ (plus ≈ 2 spires pour fixation debut/fin), pas tel que les spires se touchent, deux des ressorts enroulés dans une et deux dans l'autre direction		
25	4	Helical spring for wire guiding $\varnothing_{ext} = 1.3, \varnothing_{fil} = 0.3, l_0 = 180$		
26	4	Flexible steel wire $d = 0.63, l \approx 220$ (Carl Stahl Technocables)		
40	8	pin_iso_2338_3x12	Ac	
41	4	pin_iso_2338_3x14	Ac	
42	4	pin_iso_2338_3x18	Ac	
43	4	pin_iso_2338_3x14_FOOT	Ac	
44	16	bearing_laiton_3x4_5x5_6	Lt	
45	4	bearing_laiton_3x4_5x3_2	Lt	
46	4	spring_tube	POM	
47	4	back_bowden	Al	
48	4	spring_l3_bowden	Al	

Position	Quantity	Description	Material	Shop number
49	8	spring_washer	Al	
65	4	spring_l3_intermediate	Alx1.2	
61	2	spring_l1	Fr4x1.2	
62	2	hind_spring_l1	Fr4x1.2	
63	4	spring_l1_intermediate	Fr4x1.2	
64	4	spring_l3	Fr4x1.2	
65	4	klaue_spring (2 pcs cuivrées d'un et 2 de l'autre côté)	Fr3x1.2	
66	1	back	Fr4x2.4	
67	2	segment_l1_reinforcement	Fr4x2.4	
68	2	hind_segment_l1_reinforcement	Fr4x2.4	
69	4	segment_l3_reinforcement	Fr4x2.4	
70	4	kondo_bowden_horn	Fr4x3.2	
71	4	klaue (2 pcs cuivrées d'un et 2 de l'autre côté)	Fr3x3.2	
73	2	segment_l1	Fr3x3.2	
74	4	segment_l2	Fr3x3.2	
75	2	segment_l3	Fr3x3.2	
76	2	hind_segment_l1	Fr3x3.2	
77	4	hind_segment_l2	Fr3x3.2	
78	2	hind_segment_l3	Fr3x3.2	

13.2 Part list - modifications

13.2.1 Parts to remove

Position	Quantity	Description	Material	Shop Partnumber
70	4	kondo_bowden_horn	Fr4x3.2	

13.2.2 Parts to add

Position	Quantity	Description	Material	Shop Partnumber
70	4	Conrad Alu Servohorn 4 BOH JR Blau GEW NR 226636, similar to 233159-62	Al	Conrad 226636

13.3 Drawings - as used for the prototype

⌀0.9

pas 0.9

137°

5

R3

⌀6

5
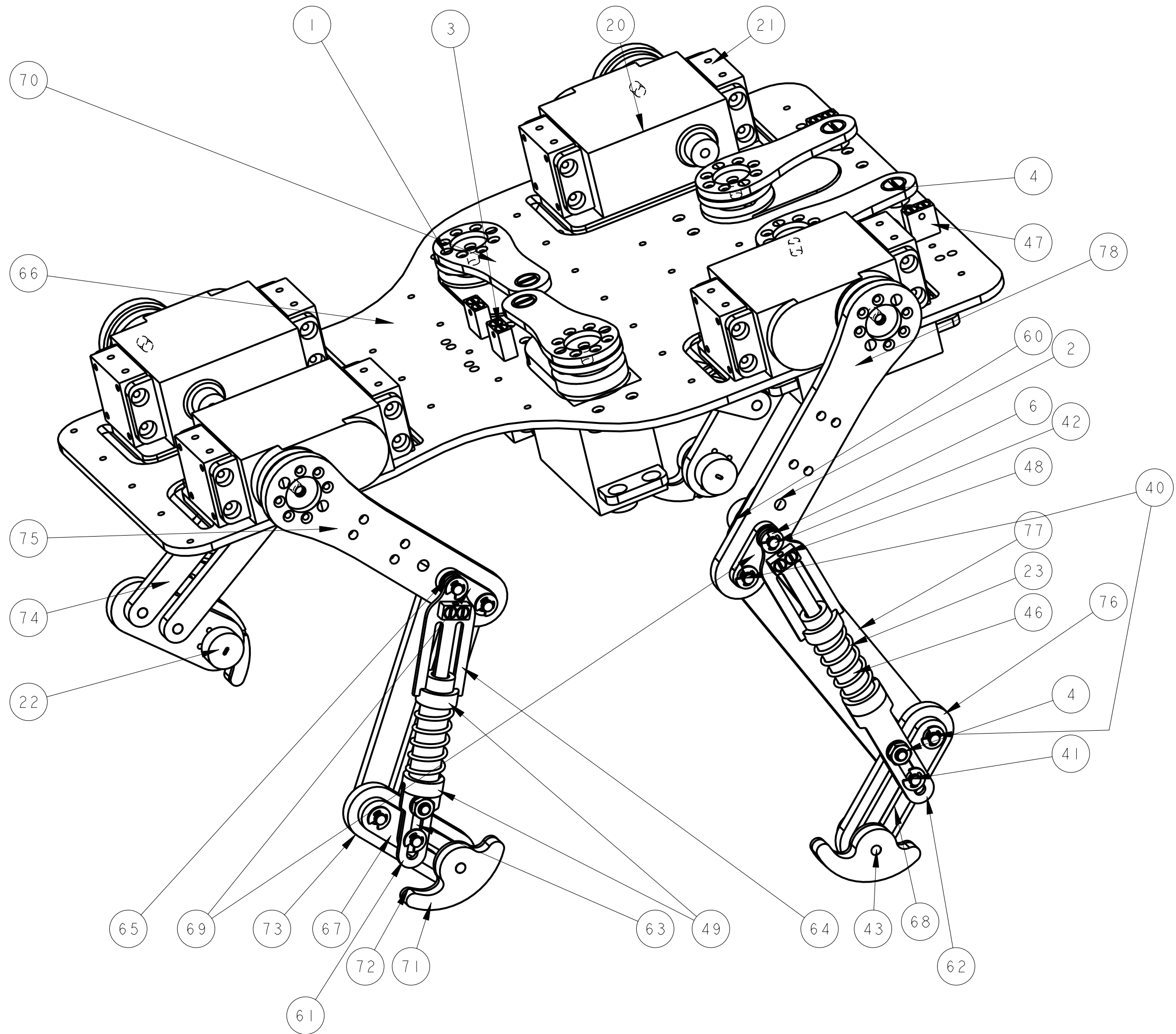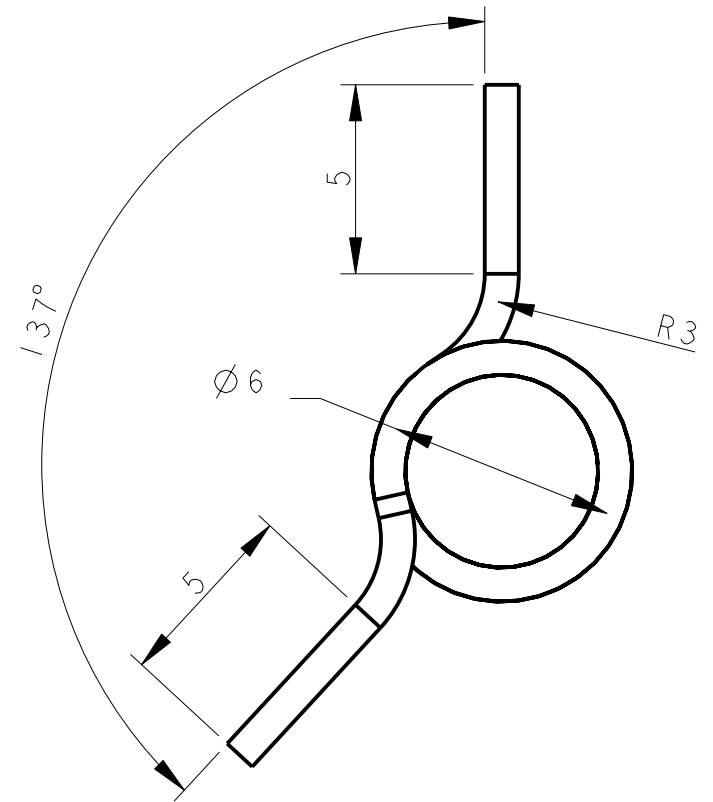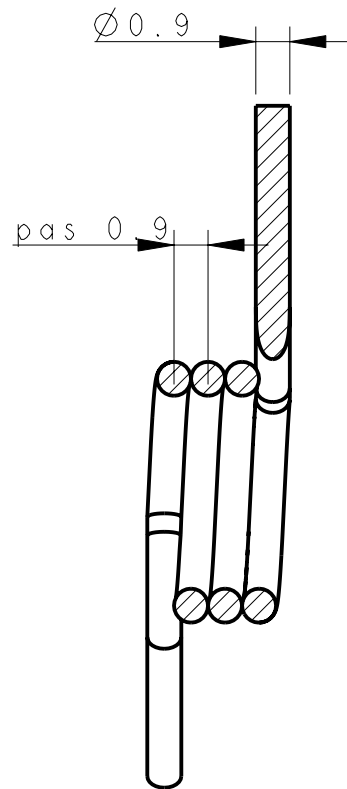
SECTION A-A

45a	I			Fil ressort		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod				Mod			Dessine	Simon Rutishauser		Echelle
							Controle			5:1
							Conf normes			
							Bon execution			
Sans nomenclature separee							N? commande			
Nomenclature sep de meme N?							Origine			Nb feuilles / Feuille N°
Nomenclature sep de N? different		N? ident					Remplace			1 / 1

N° de dessin

LEG_ASSEMBLY_FOOT_SPRING

Ø0.9

pas 0.9

SECTION A-A

137°

5

R3

5

Ø6

A

A

SCALE 5:1

∅0.9

pas 0.9

5

5

∅6

SECTION A-A

A

A

45c	I			Fil ressort		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		
Mod				Mod		Dessine Simon Rutishauser
						Controle
						Conf normes
						Bon execution
Sans nomenclature separee						N? commande
Nomenclature sep de meme N?						Origine
Nomenclature sep de N? different		N? ident				Remplace

Echelle
5:1

Nb feuilles
1

Feuille N°
1

N° de dessin

HIND_LEG_FOOT_SPRING

⌀0.9

pas 0.9

SECTION A-A

⌀6

165°

5

5

45d	I			Fil ressort		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod			Mod		Dessine	Simon Rutishauser	Echelle
					Controle		5:1
					Conf normes		
					Bon execution		
Sans nomenclature separee					N? commande		
Nomenclature sep de meme N?					Origine		Nb feuilles
Nomenclature sep de N? different	N? ident				Remplace		1

Feuille N°
1

N° de dessin

HIND_LEG_LEFT_FOOT_SPRING

180

Ø1

∅1

(Ø1.3)

(Ø0.7)

pas0.3

∅0.3

25	4			Fil ressort		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		
Mod				Mod		Dessine Simon Rutishauser	Echelle
						Controle	20:1
						Conf normes	
						Bon execution	
Sans nomenclature separee						N? commande	
Nomenclature sep de meme N?						Origine	Nb feuilles / Feuille N°
Nomenclature sep de N? different		N? ident				Remplace	1 / 1

SPRING_WIRE_GUIDING

N° de dessin
25

12

1.2

0.5

0.64 $^{+0.2}_{0}$

5°

$\varnothing$ 3 h 9

$\varnothing$ 2.3

5°

0.5

40	8			Ac		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod			Mod		Dessine	Simon Rutishauser	Echelle
					Controle		10:1
					Conf normes		
					Bon execution		
Sans nomenclature separee					N? commande		
Nomenclature sep de meme N?					Origine		Nb feuilles
Nomenclature sep de N? different	N? ident				Remplace		1

Feuille N°
1

N° de dessin

PIN_ISO_2338_3X12

14

0.5

15°

0.64 $^{+0.2}_{0}$

1.1

15°

$\emptyset$ 3h9

$\emptyset$ 2.3

0.5

41	4			Ac		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod			Mod		Dessine	Simon Rutishauser	Echelle
					Controle		10:1
					Conf normes		
					Bon execution		
Sans nomenclature separee					N? commande		
Nomenclature sep de meme N?					Origine		Nb feuilles
Nomenclature sep de N? different	N? ident				Remplace		1

Feuille N°
1

N° de dessin

EPFL

PIN_ISO_2338_3X14

18

4

Ø 3h9

0.75

0.64 $^{+0.2}_{0}$

1

15°

Ø 2.3

0.5

Ø 2

Tolerances generales ISO 2768-1 - fH

42	4			Ac			
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques			

Mod			Mod		Dessine	Simon Rutishauser		Echelle
					Controle			10:1
					Conf normes			
					Bon execution			
Sans nomenclature separee					N? commande			
Nomenclature sep de meme N?					Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident				Remplace		1	1

N° de dessin

PIN_ISO_2338_3X18

14

0.5

5°

Ø3h9
Ø2.3
0.75
0.6

0.6
2.5
0.64 +0.2 0

Ø2

Tolerances generales ISO 2768-1 - fH

43	4			Ac			
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques			

Mod			Mod		Dessine	Simon Rutishauser	Echelle
					Controle		10:1
					Conf normes		
					Bon execution		
Sans nomenclature separee					N? commande		
Nomenclature sep de meme N?					Origine		Nb feuilles
Nomenclature sep de N? different	N? ident				Remplace		1

Feuille N°
1

N° de dessin

EPFL

PIN_ISO_2338_3X14_FOOT

5.6 ±0.1

Ø4.5 ±0.05

Ø3H8

44	I6			Laiton		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod			Mod		Dessine	Simon Rutishauser		Echelle
					Controle			10:1
					Conf normes			
					Bon execution			
Sans nomenclature separee					N? commande			
Nomenclature sep de meme N?					Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident				Remplace		1	1

N° de dessin

BEARING_LAITON_3X4_5X5_6

3.2 ±0.1

⌀4.5 ±0.05
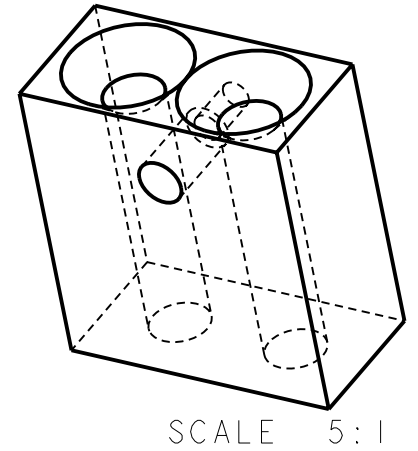
⌀3H8

45	4			Laiton		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod				Mod			Dessine	Simon Rutishauser		Echelle
							Controle			15:1
							Conf normes			
							Bon execution			
Sans nomenclature separee							N? commande			
Nomenclature sep de meme N?							Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different		N? ident					Remplace		1	1

EPFL

BEARING_LAITON_3X4_5X3_2

N° de dessin

33

0.5

30°

Ø8

Ø4

46	4			Pom ou autre plastique	
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques	

Mod			Mod		Dessine	Simon Rutishauser		Echelle
					Controle			5:1
					Conf normes			
					Bon execution			
Sans nomenclature separee					N? commande			
Nomenclature sep de meme N?					Origine			Nb feuilles / Feuille N°
Nomenclature sep de N? different	N? ident				Remplace			1 / 1

EPFL

SPRING_TUBE

N° de dessin

SECTION B-B

SECTION A-A

SCALE 5:1

8

3.6

∅3.6

90°

8.6

∅1.7

A

A

4

2.5

2

∅0.8

∅1.3

B

B

BACK_BOWDEN

8

Ø3.6

90°

3

Ø1.7

SECTION A-A

4

.5

2.5

Ø0.8

3

Ø1

SECTION C-C

SCALE 5:1

Tolerances generales ISO 2768-1 - fH

48	4			A1		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod			Mod		Dessine	Simon Rutishauser		Echelle
					Controle			10:1
					Conf normes			
					Bon execution			
Sans nomenclature separee					N? commande			
Nomenclature sep de meme N?					Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident				Remplace		1	1

N° de dessin

SPRING_L3_BOWDEN

SECTION A - A

⌀11

⌀8.1

4

2

.2 +0.1 0

Feuille N°
1

N° de dessin

SPRING_WASHER

1.2

⌀6

⌀3.1

26

4

14

11

8.05

3.95

⌀2.05

⌀3.1

⌀8

1.2

SECTION A-A

61	2			Fr4 x 1.2		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		
Mod				Mod		Dessine Simon Rutishauser	Echelle 5:1

Controle	
Conf normes	
Bon execution	

Sans nomenclature separee — N? commande
Nomenclature sep de meme N? — Origine
Nomenclature sep de N? different | N? ident — Remplace

Nb feuilles 1

Feuille N° 1

N° de dessin

EPFL

SPRING_L1

39.4

Ø3.1

Ø2.05

11

8.05

3.95

A

A

1.2

SECTION A-A

18

8

∅3.2

∅3.1

SECTION A-A

63	4			Fr4 x 1.2	
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques	

Mod			Mod			Dessine	Simon Rutishauser		Echelle
						Controle			5:1
						Conf normes			
						Bon execution			
Sans nomenclature separee						N? commande			
Nomenclature sep de meme N?						Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident					Remplace		1	1

N° de dessin

SPRING_L1_INTERMEDIATE

40.5

6.5

Ø2.15

Ø1.7

Ø6

Ø2.8

14.1

8.1

3.8

A

A

A

A

1.2

6

SECTION A-A

64	4			Fr4 x 1.2		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		
Mod				Mod		Dessine Simon Rutishauser

			Controle	
Conf normes	
Bon execution	

Sans nomenclature separee

Nomenclature sep de meme N?

Nomenclature sep de N? different N? ident

N? commande	
Origine	
Remplace	

Echelle
5:1

Nb feuilles
1

Feuille N°
1

N° de dessin

SPRING_L3

SECTION A-A

1.2

18.02

R1.99

R19.05

Ø12

Ø7

1.5

27.42

A

A

R3.5

R2.84

R2.31

2x ? face dessus cuivr?e et
2x ? face dessous cuivr?e

Pos.	Quantite	Unite	N? d'identification		Denomination / Caracteristiques		
65	4				Fr4 x 1.2		

Mod				Mod			Dessine				Echelle
							Controle				5:1
							Conf normes				
							Bon execution				
Sans nomenclature separee							N? commande				
Nomenclature sep de meme N?							Origine			Nb feuilles	Feuille N°
Nomenclature sep de N? different		N? ident					Remplace			1	1

N° de dessin

KLAUE_SPRING

235

R2

150

90

Ø20.5

SECTION B-B

2.4

Ø1.7 Ø1.7

Ø3.1

Ø2.5 Ø1.7

4

90°

90°

Ø3.2

66	I				Fr4 x 2.4		
Pos.	Quantite	Unite	N? d'identification		Denomination / Caracteristiques		

Mod				Mod			Dessine	Simon Rutishauser		Echelle
							Controle			1:1
							Conf normes			
							Bon execution			
Sans nomenclature separee							N? commande			
Nomenclature sep de meme N?							Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident						Remplace		1	1

N° de dessin

BACK

37

2.4

SECTION A-A

1.5

Ø4.5

Ø7

9

A

A

67	2			Fr4 x 2.4		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod			Mod		Dessine	Simon Rutishauser	Echelle
					Controle		5:1
					Conf normes		
					Bon execution		
Sans nomenclature separee					N? commande		
Nomenclature sep de meme N?					Origine		Nb feuilles
Nomenclature sep de N? different	N? ident				Remplace		1

Feuille N°
1

N° de dessin

SEGMENT_L1_REINFORCEMENT

Ø4.5

Ø7

2.4

SECTION A-A

42.6

1.5

9

Feuille N°
1

N° de dessin

HIND_SEGMENT_L1_REINFORCEMENT

19

7

A — A

Ø4.5

2.4

SECTION A-A

69	4			Fr4 x 2.4		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod			Mod		Dessine Simon Rutishauser			Echelle
					Controle			8:1
					Conf normes			
					Bon execution			
Sans nomenclature separee					N? commande			
Nomenclature sep de meme N?					Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident				Remplace		1	1

EPFL

SEGMENT_L3_REINFORCEMENT

N° de dessin

SECTION A-A

Ø6.72
90°
2.4
Ø3.2

Ø3.
90°
Ø10.2
Ø1.7

47.78

R33.71

2

A   A

Ø12

Ø19.56

70	4			Fr4 x 3.2		Echelle
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

N° de dessin

KONDO_BOWDEN_HORN

R1.99

∅12

R2.84

A   A

R19.05

R2.31

∅2.8

3.2

SECTION A-A

2x ? face dessus cuivr?e et
2x ? face dessous cuivr?e

71	4				Fr4 x 3.2	
Pos.	Quantite	Unite	N? d'identification		Denomination / Caracteristiques	

Mod				Mod			Dessine	Simon Rutishauser	Echelle
							Controle		4:1
							Conf normes		
							Bon execution		
Sans nomenclature separee							N? commande		
Nomenclature sep de meme N?							Origine		Nb feuilles / Feuille N°
Nomenclature sep de N? different	N? ident						Remplace		1 / 1

N° de dessin

KLAUE

Ø1.4

Ø4.5

42.5

A   A

15

Ø1.4   Ø4.5   Ø4.5

3.2

SECTION A-A

73	2			Fr4 x 3.2		
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques		

Mod			Mod		Dessine	Simon Rutishauser	Echelle
					Controle		4:1
					Conf normes		
					Bon execution		
Sans nomenclature separee					N? commande		
Nomenclature sep de meme N?					Origine		Nb feuilles / 1
Nomenclature sep de N? different	N? ident				Remplace		Feuille N° / 1

EPFL

SEGMENT_L1

N° de dessin

75.5

8

3.2

Ø2.8

SECTION A-A

74	4			Fr4 x 3.2	
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques	

Mod			Mod			Dessine	Simon Rutishauser		Echelle
						Controle			2:1
						Conf normes			
						Bon execution			
Sans nomenclature separee						N? commande			
Nomenclature sep de meme N?						Origine			Nb feuilles
Nomenclature sep de N? different		N? ident				Remplace			1

SEGMENT_L2

N° de dessin

Nb feuilles 1   Feuille N° 1

75.5

8

3.2

Ø2.8

SECTION A-A

74	4			Fr4 x 3.2	
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques	

Mod			Mod			Dessine	Simon Rutishauser		Echelle
						Controle			2:1
						Conf normes			
						Bon execution			
Sans nomenclature separee						N? commande			
Nomenclature sep de meme N?						Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident					Remplace		1	1

EPFL

SEGMENT_L2

N° de dessin

48.6

15

A

Ø1.4

3.2

Ø4.5

SECTION A-A

76	2			Fr4 x 3.2	
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques	

Mod			Mod			Dessine	Simon Rutishauser		Echelle
						Controle			4:1
						Conf normes			
						Bon execution			
Sans nomenclature separee						N? commande			
Nomenclature sep de meme N?						Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident					Remplace		1	1

EPFL

HIND_SEGMENT_L1

N° de dessin

89

A   A

8

⌀2.8

3.2

SECTION A-A

77	4			Fr4 x 3.2	
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques	

Mod			Mod		Dessine	Simon Rutishauser		Echelle
					Controle			2:1
					Conf normes			
					Bon execution			
Sans nomenclature separee					N? commande			
Nomenclature sep de meme N?					Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different		N? ident			Remplace		1	1

EPFL

HIND_SEGMENT_L2

N° de dessin

89

8

A — A

Ø2.8

3.2

SECTION A-A

77	4			Fr4 x 3.2	
Pos.	Quantite	Unite	N? d'identification	Denomination / Caracteristiques	

Mod			Mod		Dessine	Simon Rutishauser		Echelle
					Controle			2:1
					Conf normes			
					Bon execution			
Sans nomenclature separee					N? commande			
Nomenclature sep de meme N?					Origine		Nb feuilles	Feuille N°
Nomenclature sep de N? different	N? ident				Remplace		1	1

N° de dessin

HIND_SEGMENT_L2

104

14 Annex II - Test electronics

The important part of the datasheet of the PIC test board used is included here. Fig. 33 shows the schematics of the connector board used to connect the PIC Testboard to the Kondo KRS-2350 ICS RC servo motors and the power supply.

A 14-pin flat cable connector is used to connect the output pins PIN_B0, PIN_A0 - PIN_A5 and PIN_C0 from the PIC board to the corresponding pins P2-P9 on the connector board[12] (Fig. 33).

Pin TP1 on the connector board should be connected to the "7..30 V" supply voltage pin on the PIC connector board. GND is connected through the 14-pin flat cable.

A 6 V power supply is to be connected to pins TP2 (GND) and TP3 (VCC) - exactly how much power is required is not sure at the moment, at least 8 A current should be provided to power all the motors.
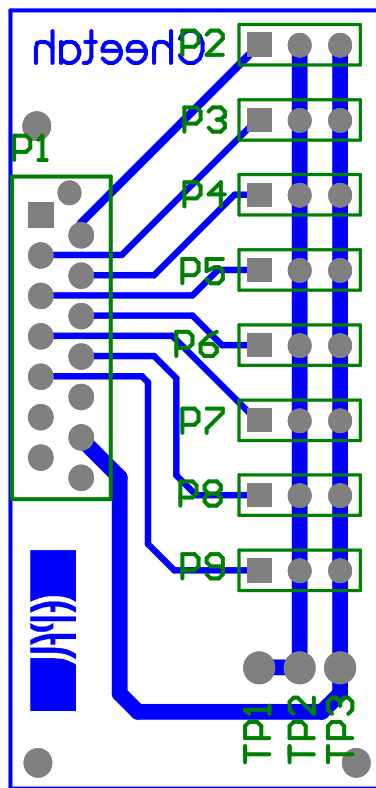
Figure 33: Schematics of the power supply connector board

---

[12]Note that if the connectors are both on the same side of the cable, either the connector on the PIC or the connector on the connector board has to be mounted the wrong way around (turned by $180°$)

RESET

SW1

progpgm

R3
47k

U3
NO

PIC18F2580ML

26 MCLR/VPP/RE3

C4
22p
C5
22p

NKS7

10MHZ
XT1

6 OSC1/CLKI/RA7

7 OSC2/CLKO/RA6

ra0 27 RA0/AN0
ra1 28 RA1/AN1
ra2 1 RA2/AN2/VREF−
ra3 2 RA3/AN3/VREF+
ra4 3 RA4/T0CKI
ra5 4 RA5/AN4/SS/HLVDIN

17 VDD

C1
100n

5 VSS1
6 VSS2

RB0/INT0/AN12 18 rb0
RB1/INT1/AN8 19 rts
RB2/INT2/CNTX 20 rb2
RB3/CANRX 21 canrx
RB4/KBI0/AN9 22 rb4
RB5/KBI1/PGM 23 rb5
RB6//KBI2/PGC 24 progpcl
RB7/KBI3/PGD 25 progpdt

RC0/T1OSO/T13CKI 8 rc0
RC1/T1OSI 9 rc1
RC2/CCP1 10
RC3/SCK/SCL 11 scl
RC4/SDI/SDA 12 sda
RC5/SDO 13 rc5
RC6/TX/CK 14 txd
RC7/RX/DT 15 rxd

METAL_PAD 29

SMDLED
VERT

D2

R4
220

C2
100n
C3
100n

SMD1
U1
MAX232
+5+10
+10-10

C6
100n

C7
100n

txd_out 14
cts_out 7
rxd_in 13
rts_in 8

out
232
in
232

in
ttl
out
ttl

11 txd
10 cts
12 rxd
9 rts

R1
4k7
R2
4k7

W1
W2

sda
scl

Pull-up

cantx

W3
JUMP3

cts

Rxx
10k

rb2

Alim.
5 VDC

+

TP1

TP2

−

D3
5.1V

C8
68u
6.3V

D1
SM4001

C9
100n

U2

NO
SOIC

MCP2551-I/SN

cantx 1 TDX
8 Rs
canrx 4 RXD
5 Vref

VDD 3
CANH 7
CANL 6
VSS 2

TP3

TP4

H

CAN BUS

L

Alim.
7-30 VDC

TP5

TP6

C11
100n

MC78M05CDT
IN   OUT
0.5A
U4

C12
100n

EXTERNAL

rb0	1	J5	MM14
ra0	2	J5	MM14
ra1	3	J5	MM14
ra2	4	J5	MM14
ra3	5	J5	MM14
ra4	6	J5	MM14
ra5	7	J5	MM14
rc0	8	J5	MM14
rc1	9	J5	MM14
rc5	10	J5	MM14
	11	J5	MM14
rb4	12	J5	MM14
rb5	13	J5	MM14
	14	J5	MM14

RS 232

9PFCI	J1	1	txd_out
9PFCI	J1	2	rxd_in
9PFCI	J1	3	
9PFCI	J1	4	
9PFCI	J1	5	
9PFCI	J1	6	rts_in
9PFCI	J1	7	cts_out
9PFCI	J1	8	
9PFCI	J1	9	
9PFCI	J1	10	
9PFCI	J1	11	

C10
100n
0805_G

I2C-old

MM6	J2	1	
MM6	J2	2	sda
MM6	J2	3	
MM6	J2	4	
MM6	J2	5	scl
MM6	J2	6	

I2C-new

1	J3	MM6	
4	J3	MM6	
3	J3	MM6	
5	J3	MM6	
2	J3	MM6	
6	J3	MM6	

A

Program

MM6	J4	1	progdbg
MM6	J4	2	progpcl
MM6	J4	3	progpdt
MM6	J4	4	
MM6	J4	5	
MM6	J4	6	progpgm

EXTERNAL

I2CNEW

GND

7..30V

RESET

I2COLD

PUP

H

CAN

GND

L

CAN

PROGR

RS232

CTS

+5V

EPFL
24.06.05
PIC18DEV

IC-LSL A.Crespi
ACORT gV
TOP

15 Annex III - Code

15.1 Matlab

The matlab files used to produce the plots shown troughout the report are on the CD-ROM that was handed in with this report.

15.2 Manual control program

The manual control program used to send simple commands over the serial bus in order to test the robot is on the CD-ROM that was handed in with this report.

It requires Qt 4.3.3 (`http://www.trolltech.com`) and QExtSerialPort (`http://qextserialport.sf.net`) and should work both on Windows and Linux/Posix systems.

15.3 Pic 18F2580 PWM

This code compiles with the CCS C Compiler - Version 4.062 was used. It is also included on the CD-ROM that was handed in with this report.

15.3.1 main.h

```
 1  /*
 2   * Author: Simon Rutishauser <simon.rutishauser@epfl.ch>
 3   * Date: 09-JAN-08
 4   */
 5
 6  #ifndef include_main
 7  #define include_main
 8
 9  #include <18F2580.h>
10  #device adc=8
11
12  #FUSES NOWDT                //No Watch Dog Timer
13  #FUSES WDT128               //Watch Dog Timer uses 1:128 ⇒
        ⇒Postscale
14  #FUSES H4                   //High speed osc with HW enabled 4X⇒
        ⇒ PLL
15  #FUSES NOPROTECT            //Code not protected from reading
16  #FUSES BROWNOUT             //Reset when brownout detected
17  #FUSES BORV21               //Brownout reset at 2.1V
18  #FUSES NOPUT                //No Power Up Timer
19  #FUSES NOCPD                //No EE protection
20  #FUSES STVREN               //Stack full/underflow will cause ⇒
        ⇒reset
```

```
21  #FUSES NODEBUG                     //No Debug mode for ICD
22  #FUSES NOLVP                       //No low voltage prgming, B3(PIC16)⇒
        ⇒ or B5(PIC18) used for I/O
23  #FUSES NOWRT                       //Program memory not write ⇒
        ⇒protected
24  #FUSES NOWRTD                      //Data EEPROM not write protected
25  #FUSES IESO                        //Internal External Switch Over ⇒
        ⇒mode enabled
26  #FUSES FCMEN                       //Fail−safe clock monitor enabled
27  #FUSES PBADEN                      //PORTB pins are configured as ⇒
        ⇒analog input channels on RESET
28  #FUSES BBSIZ2K                     //2K words Boot Block size
29  #FUSES NOWRTC                      //configuration not registers write⇒
        ⇒ protected
30  #FUSES NOWRTB                      //Boot block not write protected
31  #FUSES NOEBTR                      //Memory not protected from table ⇒
        ⇒reads
32  #FUSES NOEBTRB                     //Boot block not protected from ⇒
        ⇒table reads
33  #FUSES NOCPB                       //No Boot Block code protection
34  #FUSES LPT1OSC                     //Timer1 configured for low−power ⇒
        ⇒operation
35  #FUSES MCLR                        //Master Clear pin enabled
36  #FUSES NOXINST                     //Extended set extension and ⇒
        ⇒Indexed Addressing mode disabled (Legacy mode)
37
38  #use delay(clock=40000000)
39  #use rs232(baud=115200,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
40
41  #endif include_main
```

15.3.2 main.c

```
1   /*
2    * Author: Simon Rutishauser <simon.rutishauser@epfl.ch>
3    * Date: 09−JAN−08
4    *
5    * This program reads values from the serial bus and outputs eight ⇒
        ⇒PWM signals
6    * (may be changed by tuning PWM_COUNT) fit for Kondo KRS 2350 ICS ⇒
        ⇒servos on
7    * the pins set in pwm.c, after all eight PWM signals are received.
8    *
9    * Value    Action
10   * ————————————————————
11   * 0xff     Reset − the next value will be the setpoint for servo nr⇒
        ⇒. 1
```

```
12  * 252-4    Change operation mode of Kondo motor (exact meaning ⇒
       ⇒unknown)
13  * 251       Free mode - servos apply no torque
14  * 0-250     Position servo (0 means extreme position in counter-⇒
       ⇒clockwise direction,
15  *                      250 is the extreme in clockwise ⇒
       ⇒direction). Limits may
16   *          be enforced by setting minima/maxima in ⇒
        ⇒pwm_setpoints_min/max below.
17  */
18
19  #include "main.h"
20  #include "pwm.c"
21
22  unsigned int8 pwm_setpoints[PWM_COUNT]; // buffer for pwm ⇒
       ⇒setpoints read from serial bus
23
24  // pwm_setpoint value limits enforced by validate_setpoints()
25  unsigned int8 pwm_setpoints_min[PWM_COUNT] = {   0,   0,   0,  40,⇒
       ⇒   0,  20,   0,   0 };
26  unsigned int8 pwm_setpoints_max[PWM_COUNT] = { 250, 125, 250, 250,⇒
       ⇒ 250, 250, 250,  90 };
27
28  /* Timer2 timeout, resolution was set to 1.6 us,
29   but only a resolution of 4*1.6us = 6.4us is actually
30   used
31   SET_TIMER2(0) == 409 us
32   SET_TIMER2(255 - 4); == 6.4 us
33   */
34  #int_TIMER2
35  void  TIMER2_isr(void)
36  {
37    pwm();
38  }
39
40  /*
41  * enforce limits on setpoints in pwm_setpoints
42  */
43
44  void validate_setpoints()
45  {
46    unsigned int8 i = 0;
47    for (i = 0; i < PWM_COUNT; i++)
48    {
49      if (pwm_setpoints[i] <= 250) // special values
50      {
```

```
51          if (pwm_setpoints[i] > pwm_setpoints_max[i])
52            pwm_setpoints[i] = pwm_setpoints_max[i];
53          else if (pwm_setpoints[i] < pwm_setpoints_min[i])
54            pwm_setpoints[i] = pwm_setpoints_min[i];
55      }
56    }
57 }
58
59 /*
60 * infinite loop, read from serial bus
61 *
62 * if 0xff is read the following value is a setpoint for the servo ⇒
         ⇒on the first
63 * channel (sending this is not necessary − after having received ⇒
         ⇒PWM_COUNT values
64 * the loop restarts with reading a setpoint for the first servo)
65 */
66 void read_serial() {
67    static int1 led = 0;
68    unsigned char c;
69    static unsigned int8 i = 0;
70
71    while(TRUE) {
72      c = getchar();
73
74      if (c == 0xff)
75        i = 0xff;
76      else
77      {
78        pwm_setpoints[i] = c;
79
80        if (i == PWM_COUNT−1)
81        {
82          validate_setpoints();
83          pwm_set_values(pwm_setpoints);
84        }
85      }
86
87      i++;
88      i %= PWM_COUNT;
89
90      led=~led;
91      output_bit(PIN_C2, led);
92    }
93 }
94
```

112

```
 95  void main()
 96  {
 97      setup_adc_ports(NO_ANALOGS|VSS_VDD);
 98      setup_adc(ADC_OFF|ADC_TAD_MUL_0);
 99      setup_spi(SPI_SS_DISABLED);
100      setup_wdt(WDT_OFF);
101      setup_timer_0(RTCC_INTERNAL);
102      setup_timer_1(T1_DISABLED);
103      setup_timer_2(T2_DIV_BY_16,255,1);
104
105      setup_vref(FALSE);
106
107      pwm_init();
108
109      enable_interrupts(INT_TIMER2);
110      enable_interrupts(GLOBAL);
111
112      read_serial();
113  }
```

15.3.3 pwm.h

```
 1  /*
 2   * Author: Simon Rutishauser <simon.rutishauser@epfl.ch>
 3   * Date: 09−JAN−08
 4   *
 5   * This file implements all the functions necessary for creating ⇒
        ⇒the PWM signals.
 6   * It requires Timer2 to be set up correctly (1.6 us steps) and the⇒
        ⇒ interrupt routine
 7   * must call the function pwm().
 8   */
 9
10  #ifndef include_pwm
11  #define include_pwm
12
13  #include "main.h"
14
15  // Number of PWM channels
16  const unsigned int8 PWM_COUNT = 8;
17
18  /*
19  Initialize PWM to default values (Robot is standing)
20  */
21  void pwm_init();
22
23  /*
```

```
24  ISR for TIMER2 − do all the PWM logics
25  */
26  void pwm();
27
28  /*
29   * Set PWM values
30   * IN:     Pointer to unsigned int8[PWM_COUNT] containing servo ⇒
          ⇒angles 0...250
31   *        or special values 251..254
32   */
33  void pwm_set_values(unsigned int* values);
34
35
36  #endif include_pwm
```

15.3.4 pwm.c

```
1  /*
2   * Author: Simon Rutishauser <simon.rutishauser@epfl.ch>
3   * Date: 09−JAN−08
4   */
5
6  #include "pwm.h"
7
8  #include <stdio.h>
9
10 // multiply time counter value with time_to_us to get us
11 const double TIME_TO_US = 6.4;
12
13 // output pins
14 const unsigned int16  pins[] = { PIN_B0, PIN_A0, PIN_A1, PIN_A2, ⇒
       ⇒PIN_A3, PIN_A4, PIN_A5, PIN_C0 };
15
16 // kondo characteristics
17 const unsigned int16 PWM_PULSE_BASE = 700/TIME_TO_US;     // ⇒
       ⇒minimum pulse length
18
19 // special pulse widths
20 const unsigned int16 PWM_PULSE_FREE = 50/TIME_TO_US;      // switch⇒
       ⇒ to free mode
21 const unsigned int16 PWM_PULSE_100 = 100/TIME_TO_US;
22 const unsigned int16 PWM_PULSE_150 = 150/TIME_TO_US;
23 const unsigned int16 PWM_PULSE_200 = 200/TIME_TO_US;
24
25 // time from PWM pulse start to next one
26 const unsigned int16 PWM_CYCLE_TIME = 8000/TIME_TO_US;
27
```

```
28  const unsigned int16 PWM_MIN_DELTA_T = 50/TIME_TO_US; // PWM does ⇒
        ⇒not resolve outputs with time differences below this due to ⇒
        ⇒hardware limitations
29
30  // time counter, one step == TIME_TO_US = 6.4 us
31  static unsigned int16 time = 0;
32
33  // time counter values at which we have to set pins to zero
34  static unsigned int16 pwm_values_values[PWM_COUNT];
35  // corresponding pins
36  static unsigned int16 pwm_values_pins[PWM_COUNT];
37  // array index − which is the next value to treat in pwm_values_*
38  static unsigned int8 pwm_values_pos = 0;
39
40  void pwm_sort_values();
41  signed int pwm_values_compare(char *arg1, char *arg2);
42
43  /*
44   * initialize pwm values − robot should be standing
45   */
46  void pwm_init() {
47      unsigned int8 val[PWM_COUNT] = { 90, 0, 170, 250, 170, 250, 90, ⇒
            ⇒0 };
48      pwm_set_values(val);
49  }
50
51  /*
52   * TIMER2 overflow interrupt treatment − do all the PWM logics
53   *
54   * How it works: We keep our own 16−bit timer "time". All the time ⇒
        ⇒variables used
55   * work with a resolution of 6.4 us (whereas TIMER2 has a ⇒
        ⇒resolution of 1.6 us).
56   *
57   * All the PWM output durations are sorted and stored in ⇒
        ⇒pwm_values_values, the
58   * program starts by setting all output pins (to 1) at time = 0
59   *
60   * It looks at pwm_values_values[0] to see if it has to set ⇒
        ⇒pwm_values_pins[0] to
61   * zero. If not it checks how long it has to wait:
62   *   − longer than a complete TIMER2 cycle (−> increment time by 64⇒
        ⇒ (*6.4us))
63   *   − shorter than a complete TIMER2 cycle (−> set TIMER2 ⇒
        ⇒accordingly to get the
```

```
64   *        next timeout when we have to reset pwm_values_values[0] and ⇒
          ⇒increment time)
65   * If it has to reset pwm_values_pins[0] it does so, checks if it ⇒
          ⇒also, increments
66   * pwm_values_pos, and checks if it has to reset pwm_values_pins[1,⇒
          ⇒ 2, 3, i]
67   * Then the whole restarts, only that the program now looks at ⇒
          ⇒pwm_values_values[i].
68   */
69   void pwm() {
70     unsigned int8 i = 0;
71     unsigned int16 dt = 0;
72
73     // reset all the PWM output pins that have to be reset now
74     while( pwm_values_pos < PWM_COUNT && (time + PWM_MIN_DELTA_T >= ⇒
          ⇒pwm_values_values[pwm_values_pos] ) ) {
75       output_bit(pwm_values_pins[pwm_values_pos], 0x00);
76       pwm_values_pos++;
77     }
78
79     // set all the pins and start new PWM cycle
80     if ( time >= PWM_CYCLE_TIME) {
81       for(i = 0; i < PWM_COUNT; i++) {
82         output_bit(pins[i], 0xFF);
83       }
84       time = 0;
85       pwm_values_pos = 0;
86     }
87
88     // all outputs are zero, waiting for time to reach ⇒
          ⇒PWM_CYCLE_TIME
89     if (pwm_values_pos == PWM_COUNT) {
90       time += 64;
91     }
92     else {
93       // difference between our 16-bit timer and the next PWM value ⇒
          ⇒change
94       dt = pwm_values_values[pwm_values_pos] - time;
95
96       if (dt >= 64) {
97         // let TIMER2 restart at 0 for another cycle
98         time += 64;
99       }
100      else {
101        // next PWM output change is due faster than next TIMER2 ⇒
          ⇒overflow,
```

116

```
102        // set TIMER2 accordingly
103        // left shift due to 1.6us resolution of TIMER2, "our" timer⇒
             ⇒−variables
104        // only use 6.4us = 4∗1.6us resolution
105        SET_TIMER2(256 − (dt<<2) );
106        time += dt;
107      }
108    }
109  }
110
111  /*
112   * change setpoints of the PWM
113   */
114  void pwm_set_values(unsigned int8∗ values) {
115    unsigned int8 i;
116
117    for (i = 0; i < PWM_COUNT; i++) {
118      pwm_values_pins[i] = pins[i];
119
120      if(values[i] <= 250) {
121        pwm_values_values[i] = values[i] + PWM_PULSE_BASE;
122      }
123      else {
124        switch(values[i]) {
125          case 251:
126            pwm_values_values[i] = PWM_PULSE_FREE;
127            break;
128          case 252:
129            pwm_values_values[i] = PWM_PULSE_100;
130            break;
131          case 253:
132            pwm_values_values[i] = PWM_PULSE_150;
133            break;
134          case 254:
135            pwm_values_values[i] = PWM_PULSE_200;
136            break;
137          case 255:
138            pwm_values_values[i] = 0;
139            break;
140        }
141      }
142    }
143
144    pwm_sort_values();
145  }
146
```

```
147  /*
148   * compare function for the pwm_sort_values() quicksort algorithm
149   */
150  signed int pwm_values_compare(char *arg1, char *arg2) {
151    int16 val1, val2;
152
153    val1 = *((unsigned int16 *) arg1);
154    val2 = *((unsigned int16 *) arg2);
155
156    if (val1 < val2)
157      return −1;
158    else if (val1 == val2)
159      return 0;
160    else
161      return 1;
162  }
163
164  /*
165   * Adapted quicksort, sorts pwm_values_values in ascending order;
166   *
167   * the same move operations are also applied to pwm_values_pins ⇒
          ⇒such that
168   * the pin address corresponding to a timeout in pwm_values_values ⇒
          ⇒is at the
169   * same array index in pwm_values_pins
170   */
171  void pwm_sort_values() {
172      unsigned int m,j,i,l;
173      int1 done;
174      //BYTE t[16];
175      unsigned int16 t;
176
177      m = PWM_COUNT/2;
178      while( m > 0 ) {
179        for(j=0; j<(PWM_COUNT−m); ++j) {
180            i = j;
181          do
182          {
183              done=TRUE;
184              l = i+m;
185              if( pwm_values_compare(pwm_values_values+i, ⇒
                    ⇒pwm_values_values+l) > 0 ) {
186                // switch values in pwm_values_values
187                t = pwm_values_values[i];
188                pwm_values_values[i] = pwm_values_values[l];
189                pwm_values_values[l] = t;
```

```
190
191                       // also switch corresponding values in ⇒
                              ⇒pwm_values_pints
192                       t = pwm_values_pins[i];
193                       pwm_values_pins[i] = pwm_values_pins[l];
194                       pwm_values_pins[l] = t;
195
196                       if(m <= i)
197                           i −= m;
198                           done = FALSE;
199                   }
200             } while(!done);
201         }
202     m = m/2;
203     }
204 }
```

16 Annex IV - Datasheets

16.1 Kondo Red Version technical manual

There is no official Kondo servo manual available. The following unofficial and incomplete translation from the Japanese manual can be found at `http://www.robotshop.ca/PDF/rbixs03-KRSmanual.pdf`

16.1.1 Findings about the Kondo Servos

PWM modulation

Short pulses of $700 \ \mu s$ mean that the servo turns in counterclockwise direction, the longer the pulse the more it turns in clockwise direction.

The $100 \ \mu s, 150 \ \mu s$ and $200 \ \mu s$ allow apparently to switch between different servo configurations (PID parameters and other settings). However these configurations can only be tuned using a proprietary interface, they are therefore not very useful in practice.

Motor and Gear

Since Dynamixel, producing servo motors with performance slightly inferior to the Kondo servos, officially states that they include Maxon motors it can be assumed that Kondo also uses motors made by Maxon, Faulhaber, Escap or a similar high-performance motor brand.

Diameter and length of the included motor are both $17 \ mm$, the nominal voltage is 6 V. Checking in the catalogues of the forementioned fabricants, only Faulhaber has a motor of these dimensions (it is not sure which brand of motors is used by Kondo, though).

The exact gear reduction is not known - assuming $r = 170$ (commonly found in servo motors) it is found from the Faulhaber motor data that:

$$\omega_{out} = 2\pi \frac{n_{c,max}}{60} = 1047 \text{ rad/s} \tag{59}$$

where $n_{c,max} = 14'000 \text{tr/min}$. This gives a maximum velocity in the usual servo-units ($s/60° = s/rad$) of

$$v_{max} = \frac{1}{\frac{\omega}{r}} = 0.1623 \text{ s/rad} \tag{60}$$

Which corresponds exactly to the known maximum velocity of a KRS-2350 ICS motor.

The stall torque of $M_H = 5.34$ mNm would result in an output stall torque of

$$M_0 = r \cdot M_H = 170 \cdot 5.34 \text{ mNm} = 0.9 \text{ Nm} \tag{61}$$

which is about half of what is given for the KRS-2350 ($20 \ Kg \cdot cm \approx 2 \ Nm$).

Thus it seems that Kondo does use a motor which, though similar in dimensions, has characteristics that differ quite a bit from those of a Faulhaber 1717 SR motor.

KRS Red Version Series technical manual

Thank you for choosing KONDO KRS series servo. Please make sure this instruction manual using KRS series servo. This instruction describes the mutual contents for a KRS Red Version, please refer the each specification to each manual.

Specification
Dynamical Specification Setting Function

It requires the ICS PC interface cable (as an option) for connecting with PC. You can download this application from KONDO website.

Connection with PC available!!

Advanced interactive communication system inside. You can connect with PC using ICS PC interface cable (No.01018 6,000yen). You can set the servo dynamics parameters as you like using GUI interface.

<Setting parameters>

Pulse stretch	Holding characteristic
Speed	Max speed
Punch	Initial response
Dead band	Neutral band
Damping	Damping parameter
Pulse operation timer	Time to free mode
Limit	Operation angle
Reverse	Reverse direction

Specification
Advanced Function

New function
     Characteristic change function
Changeable setting parameter from the specific signal
     Position capture function
You can obtain current position from servo.
     Power reduction function
You can cut the power by command.

Danger !   It is postulated dangers such as death or heavy injury unless you keep in mind.

Do not touch servos during operation

This servo outputs high torque, therefore, you will fracture and cutoff when you intervene between the mechanism.

Warning !   It is postulated dangers such as injury or damage something unless you keep in mind.

Please make sure the input signal and input voltage

Do not input irregular signal and voltage. It causes malfunction or breakage.

Please make sure the direction of connector and insert correctly.

It causes malfunction of servo.

Do not use on wet place.

IT cause malfunction of servo.

Do not decompose and modify.

Operating remark

This is only for robot, you can not use for radio control car model and so on.

Please make sure a servo operate smoothly. A life of servo will be shorter if mechanism has deflection and gap.

Servo might be cause sufficient output torque if you select dry cell.

Operation angle range depends on an input signal.

This servo accepts the signal from 700 usec to 2300 usec, and these corresponds the minimum angle and maximum angle. Avoiding misoperation by noise and wrong input signal, you can set the limit value using ICS PC interface cable which is option.

Common Specifications

Input voltage : DC 6V   Maximum operation angle : 180 degree

Control signal

You can connect this servo to general receiver for radio control model
(neutral position is 1.5 msec,   signal period is from 8 to 20 msec)

Technical instruction for advanced functions

Characteristic change function (changing parameters)

Input signal for general servos is from 0.7 to 2.3 msec signal and from 8 to 20 msec signal period. Servos control position corresponding to this input signal, furthermore, red version servos can change the pulse stretch, speed parameters.
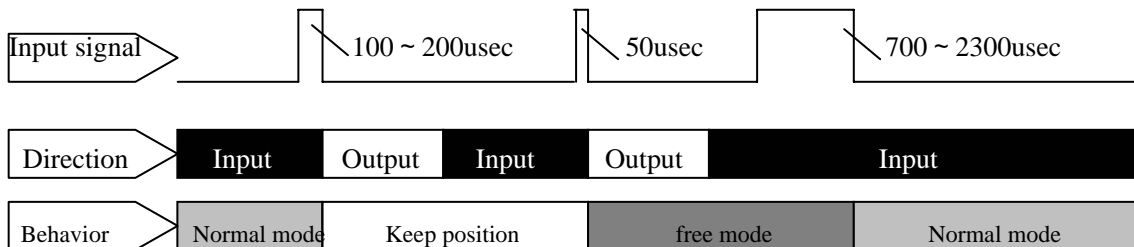
Input signal : 100 usec	➡	change to parameter 1
Input signal : 150 usec	➡	change to parameter 2
Input signal : 200 usec	➡	change to parameter 3

Power reduction function (free mode)

If you send 50 usec signal, servos becomes free mode until input signal from 700 to 2300 usec will be caused. When you send characteristic change function during free mode, a servo keeps free mode.

Position capture function (output a current position)

When a servo obtains 50, 100, 150 and 200 usec signal, it outputs a position signal corresponding to current position within 100 usec. You can calculate a current position monitoring this pulse on outside PC. After sending a position signal from a servo, it changes input mode. During output mode, a servo keeps current position.

| Input signal | | 100 200usec | 50usec | 700 2300usec |

| Direction | Input | Output | Input | Output | Input |

| Behavior | Normal mode | Keep position | free mode | Normal mode |

Notice for using advanced function

3 characteristics are same initially.

Please make sure the parameter number when you setup it.

Please make sure the battery voltage and noise, they may cause irregular behavior.

CPU boards are required pull up resister and change direction as input/output in order to use position capture function.

How to use with double holding axis (This is only for servo with bottom axis)

M2.6 x 6 tapping screw

Washer

Free hone

KRS-700 series has a axis on bottom case, you can use double holding axis using free hone.

M2.6 x 6 tapping screw

Washer

Bearing

Free hone

KRS-2300 series has a axis on bottom case, you can use double holding axis using free hone with bearing No.01025).

---
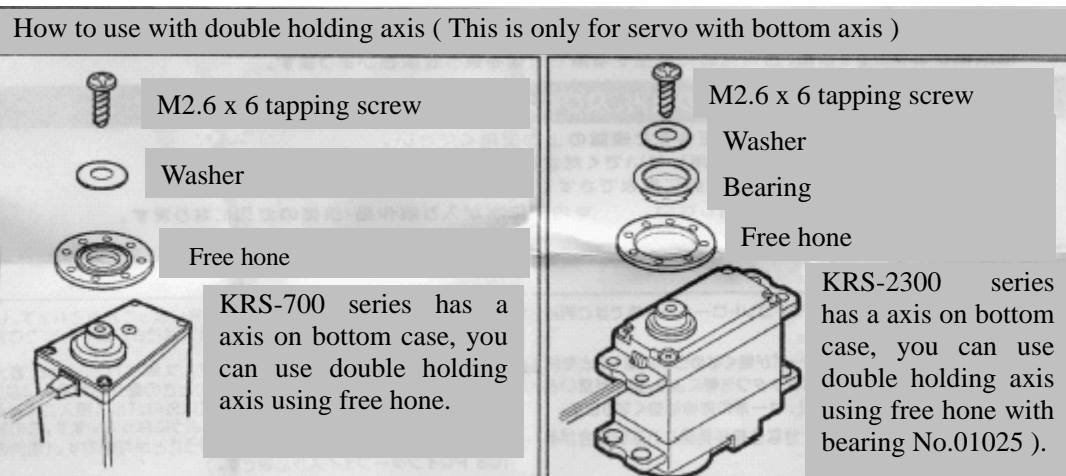
for repair

Please send defect products with following message.

(1) Status of your problem
(2) Situation of your usage
(3) Product number and quantity
(4) Your name, address, telephone number

Contact

DC-Micromotors

Precious Metal Commutation

2 mNm

For combination with (overview on page 14-15)
Gearheads:
15A, 16A, 16/7
Encoders:
IE2 – 16 ... 512

Series 1717 ... SR

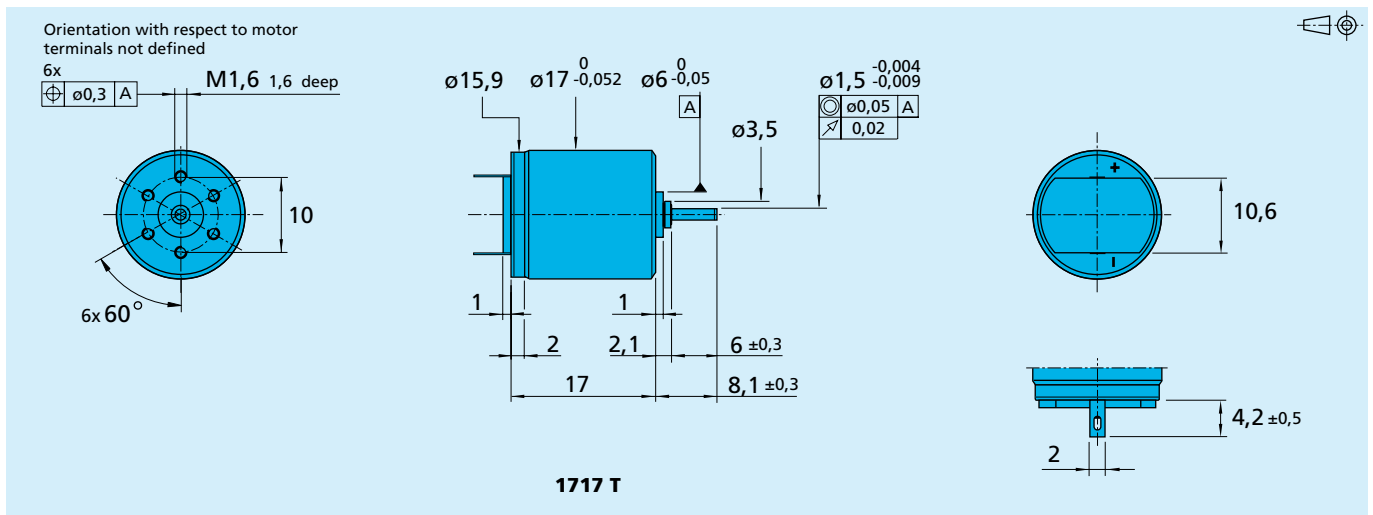|		1717 T		003 SR	006 SR	012 SR	018 SR	024 SR	
1	Nominal voltage	U_N		3	6	12	18	24	Volt
2	Terminal resistance	R		1,07	4,30	17,1	50,1	68,8	Ω
3	Output power	$P_{2\ max.}$		1,97	1,96	1,97	1,50	1,96	W
4	Efficiency	$\eta_{max.}$		69	69	70	68	70	%
5	No-load speed	n_o		14 000	14 000	14 000	12 300	14 000	rpm
6	No-load current (with shaft ø 1,5 mm)	I_o		0,091	0,046	0,023	0,013	0,011	A
7	Stall torque	M_H		5,37	5,34	5,38	4,66	5,36	mNm
8	Friction torque	M_R		0,18	0,18	0,18	0,18	0,17	mNm
9	Speed constant	k_n		4 820	2 410	1 210	709	602	rpm/V
10	Back-EMF constant	k_E		0,207	0,414	0,829	1,410	1,660	mV/rpm
11	Torque constant	k_M		1,98	3,96	7,92	13,50	15,90	mNm/A
12	Current constant	k_I		0,505	0,253	0,126	0,074	0,063	A/mNm
13	Slope of n-M curve	$\Delta n/\Delta M$		2 610	2 620	2 600	2 640	2 610	rpm/mNm
14	Rotor inductance	L		17	65	260	760	1 040	µH
15	Mechanical time constant	τ_m		16	16	16	16	16	ms
16	Rotor inertia	J		0,59	0,58	0,59	0,58	0,59	gcm²
17	Angular acceleration	$\alpha_{max.}$		92	92	92	80	92	·10³rad/s²
18	Thermal resistance	$R_{th\ 1}$ / $R_{th\ 2}$	4,5 / 27						K/W
19	Thermal time constant	τ_{w1} / τ_{w2}	2,0 / 210						s
20	Operating temperature range:								
	– motor		– 30 ... + 85 (optional – 55 … + 125)						°C
	– rotor, max. permissible		+125						°C
21	Shaft bearings		sintered bronze sleeves (standard)		ball bearings (optional)		ball bearings, preloaded (optional)		
22	Shaft load max.:								
	– with shaft diameter		1,5		1,5		1,5		mm
	– radial at 3 000 rpm (3 mm from bearing)		1,2		5		5		N
	– axial at 3 000 rpm		0,2		0,5		0,5		N
	– axial at standstill		20		10		10		N
23	Shaft play:								
	– radial	≤	0,03		0,015		0,015		mm
	– axial	≤	0,2		0,2		0		mm
24	Housing material		steel, black coated						
25	Weight		18						g
26	Direction of rotation		clockwise, viewed from the front face						

Recommended values - mathematically independent of each other

				003 SR	006 SR	012 SR	018 SR	024 SR	
27	Speed up to	$n_{e\ max.}$		10 000	10 000	10 000	10 000	10 000	rpm
28	Torque up to	$M_{e\ max.}$		2	2	2	2	2	mNm
29	Current up to (thermal limits)	$I_{e\ max.}$		1,20	0,60	0,30	0,18	0,15	A

Orientation with respect to motor terminals not defined

6x
⊕ ø0,3 | A

M1,6 1,6 deep

ø15,9   ø17 0/-0,052   ø6 0/-0,05   A

ø3,5

ø1,5 -0,004/-0,009
⊚ ø0,05 | A
⟋ 0,02

10

6x 60°

1   1

2   2,1   6 ±0,3

17   8,1 ±0,3

+

10,6

4,2 ±0,5

2

1717 T

For details on technical information and lifetime performance refer to pages 28-34.
Edition 2006-2007

For options on DC-Micromotors refer to page 64.
Specifications subject to change without notice.
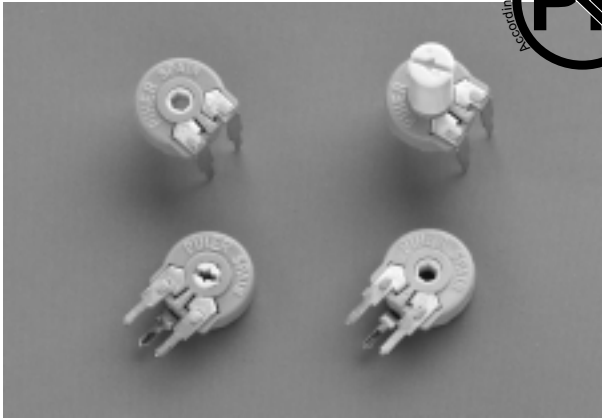
www.faulhaber-group.com

16.2 Piher PTC 10 LV

This is the datasheet for the Piher PTC 10 LV potentiometers to be used for angle detection. Using a potentiometer with "E" option is advisable since these have higher mechanical lifetime. The pin drawings fit to the "L" rotors.

PIHER

PTC-10 | 10 mm Cermet Potentiometer

According to ELV (2000/53/EC) / RoHS (2002/95/EC) — Pb

FEATURES

– Cermet resistive element.
– Dust proof enclosure.
– Plastic material according to UL94V-0
– Alumina substrate.
– Also upon request:
  • Wiper positioned at 50% or fully clockwise.
  • Supplied in magazines for automatic insertion.
  • Long life model for low cost control pot. applications
  • Special tapers
  • Mechanical detents
  • Low & extra low torque versions
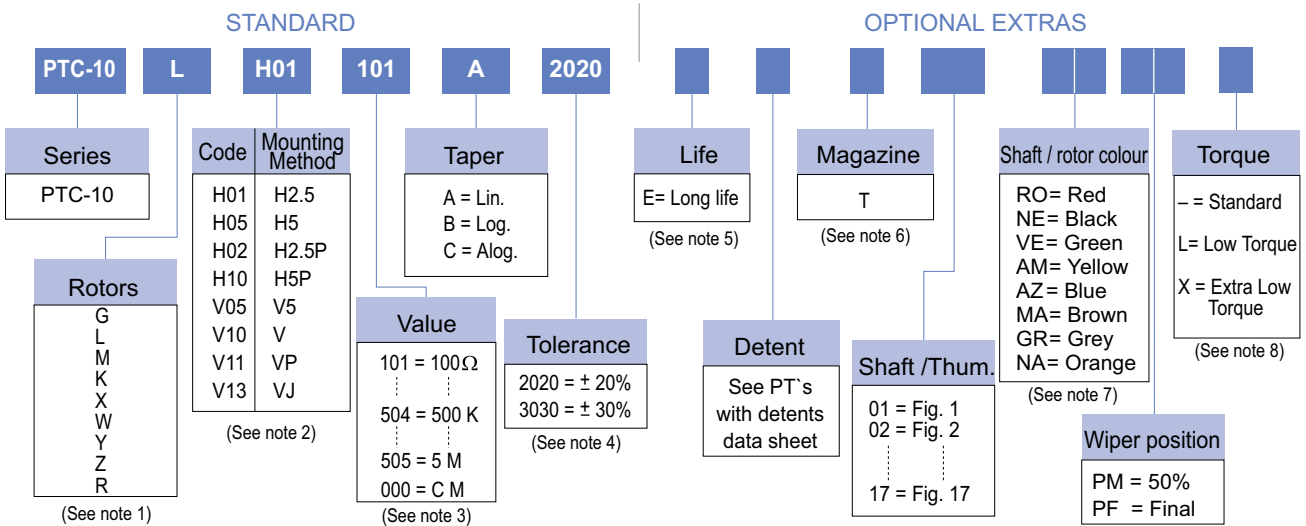 • Available as SPDT switch
  • Laser trimming for tighter tolerances

MECHANICAL SPECIFICATIONS

– Mechanical rotation angle: $235° \pm 5°$
– Electrical rotation angle: $220° \pm 20°$
– Torque: 0.4 to 2 Ncm.
  (0.6 to 2.7 in-oz)
– Stop torque: > 5 Ncm. (>7 in-oz)

ELECTRICAL SPECIFICATIONS

– Range of values (*)
  $100\Omega \leq Rn \leq 5\ M$ (Decad. 1.0 - 2.0 - 2.2 - 2.5 - 4.7 - 5.0)
– Tolerance (*): $100\Omega \leq Rn \leq 1M\ \Omega$ ------ $\pm 20\%$
  $1M\Omega < Rn \leq 5M\ \Omega$ ------ $\pm 30\%$
– Max. Voltage: 200 VDC (lin) 100 VDC (no lin)
– Nominal Power 70°C (158°F) (see power rating curve)
  0.33 W (lin) 0.17 W (no lin)
– Taper (*) (Log. & Alog. only $Rn \geq 1K$) Lin ; Log; Alog.
– Residual resistance: $\leq 5.10^{-3}$ Rn (2 Ω min.)
– Equivalent Noise Resistance: $\leq 3\%$ Rn (3 Ω min.)
– Operating temperature: –40°C + 90°C (–40°F + 194°F)

(*) Others upon request

HOW TO ORDER

STANDARD | OPTIONAL EXTRAS

PTC-10 | L | H01 | 101 | A | 2020 | | | | |

Series					Taper		Life	Magazine	Shaft / rotor colour	Torque
PTC-10					A = Lin. B = Log. C = Alog.		E= Long life	T	RO= Red NE= Black VE= Green AM= Yellow AZ= Blue MA= Brown GR= Grey NA= Orange	– = Standard L= Low Torque X = Extra Low Torque

Code	Mounting Method
H01	H2.5
H05	H5
H02	H2.5P
H10	H5P
V05	V5
V10	V
V11	VP
V13	VJ

(See note 2)

Rotors
G
L
M
K
X
W
Y
Z
R

(See note 1)

Value
101 = 100Ω
:
504 = 500 K
:
505 = 5 M
000 = C M

(See note 3)

Tolerance
2020 = ± 20%
3030 = ± 30%

(See note 4)

Life — E= Long life (See note 5)

Magazine — T (See note 6)

Detent
See PT`s with detents data sheet

Shaft /Thum.
01 = Fig. 1
02 = Fig. 2
17 = Fig. 17

Shaft / rotor colour (See note 7)

Wiper position
PM = 50%
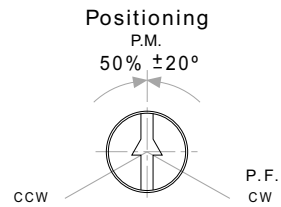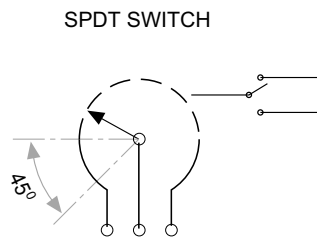PF = Final

Torque (See note 8)

NOTES:

(1) "Z" adjustment only available on "H" versions. Rotor "G" only available in purple (shaft/rotor colour "VI").
(2) Terminal styles: "P" & "J" are crimped terminals. V=Vertical adjust; H=Horizontal Adjust
(3) Value Example: Code: 10 1 = 100 Ω
  → Numb of zeros
  → First two digits of the value. 000 = CM: SPDT switch 45°
(4) Non standard tolerance, upon request. Example: +7% Code: 07 05
  -5% → negative tolerance
  → positive tolerance
(5) Life • Standard 100 cycles
  • Long life 10000 cycles
(6) Magazines: not available with the H10, V05 and V13 models, nor with adjustment types X, W, Y, Z.
  Non flammable: housing, rotor and shaft.
(7) Colour shaft/rotor: • Potentiometer without shaft: only rotor • Potentiometer with shaft: only shaft
(8) Low Torque: 0.25 to 1 Ncm (per pot.)
  Extra Low Torque: 0.1 to 0.4 Ncm (per pot.). Only available on "H" models without crimping
  No detent option available for low and extra low torque models. No shaft or thumbwheel option for extra low torque models.

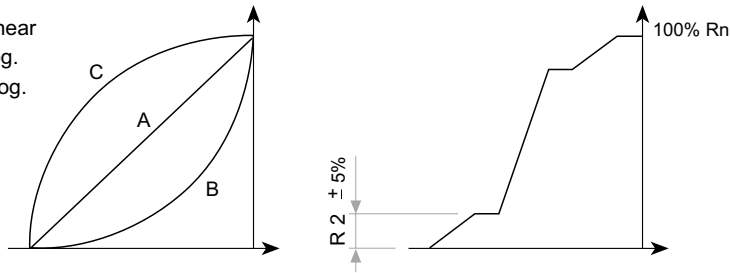NOTE: The information contained here should be used for reference purposes only.

MEGGITT

HOW TO ORDER CUSTOM DRAWING

PTC-10 LH 01 + DRAWING NUMBER (Max. 16 characters)

This way of ordering should be used for options which are not included in the "How to order" standard and optional extras.

STANDARD OPTIONS

Mechanical Life ------------------------------- 100 cycles
Detents -------------------------------------- None
Packing -------------------------------------- Bulk
Rotor colour --------------------------------- Natural
Shaft colour --------------------------------- Natural
Wiper position ------------------------------- Initial
Torque --------------------------------------- Standard

ROTORS

Without shaft

L = Screwdriver thru hole
M = Hexagonal thru hole
G = Hexagonal thru hole
K = Cross slot thru hole
R

With shaft

X = Adjustable from collector side
W = Adjustable from terminal side

Wipers positioned at 50%

With thumbwheel

Y = Adjustable from terminal side

Z = Adjustable from collector side

MOUNTING METHODS

v = horizontal mount – vertical adjust

v (10)   **A** = Initial   **S** = Wiper   **E** = Final

v (5)

NOTE = Please note relative terminal positions when ordering non linear tapers.

h = vertical mount – horizontal adjust

h (2,5)   **A** = Initial   **S** = Wiper   **E** = Final

h (5)

Crimped terminals

Mod. **J**   Mod. **P**   Detail

SPDT SWITCH

Positioning
P.M.
50% ±20°

CCW

P.F.
CW

Std. Position = CCW

TAPERS

A = Linear
B = Log.
C = Alog.

Standard

100% Rn

R 2 $\pm$ 5%

Special example

POWER RATING CURVE

W (%)

100 %

40 %

70°   90°   °C
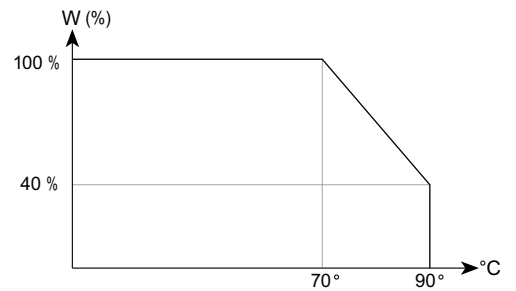
NOTE = Please note relative terminal positions when ordering non linear tapers.

TESTS		TYPICAL VARIATIONS
ELECTRICAL LIFE	1.000 h. @ 70°C; 0.33 W	±5 %
MECHANICAL LIFE (CYCLES)	100 @ 10 CPM ...15 CPM	±3 % (Rn < 1 MΩ)
TEMPERATURE COEFFICIENT	−40°C; +90°C	±100 ppm (Rn <100 ±K)
THERMAL CYCLING	16 h. @ 90°C; 2h. @ −40°C	±2.5 %
DAMP HEAT	500 h. @ 40°C @ 95% HR	±5 %
VIBRATION (for each plane X,Y,Z)	2 h. @ 10 Hz. ... 55 Hz.	2 %

NOTE: Out of range values may not comply these results.

PACKAGING

BOXES

Model	Units
Without shaft	500 (40 x 85 x 185 mm.)
With thumbwheel	400 (40 x 85 x 185 mm.)
With shaft	200 (40 x 85 x 185 mm.)

7.6

18.9

540 $^{+1.5}$
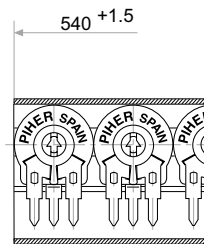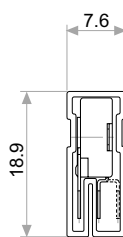
Magazines for PTC-10 h 2.5; h 5

Also crimped term. h 2.5 P

AUTOMATIC INSERTION

Magazines	Units
PTC-10H & PTC-10V	50 Pieces

540 $^{+1.5}$

13.5

12
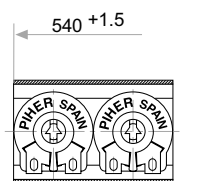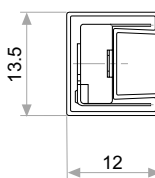
Magazines for PTC-10 V

Also crimped term. VP
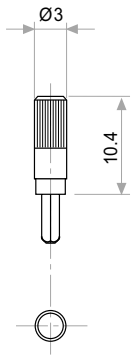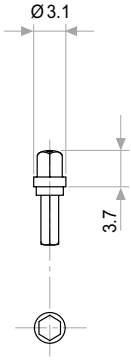
Fig. 1 / Ref. 5016

Fig. 2 / Ref. 5053
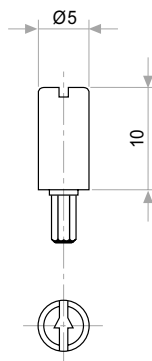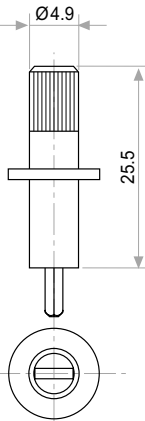
Fig. 3 / Ref. 5012
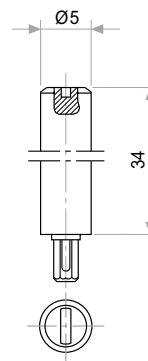
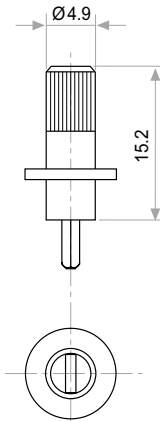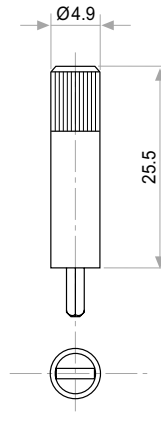Fig. 4 / Ref. 6053
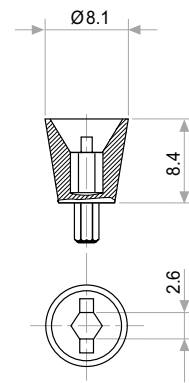
Fig. 6 / Ref. 5035
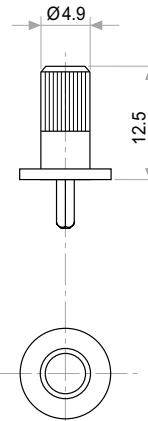
Fig. 7 / Ref. 5115

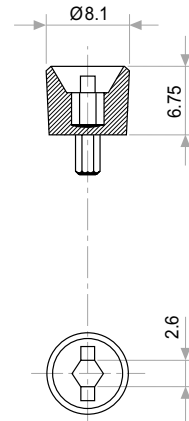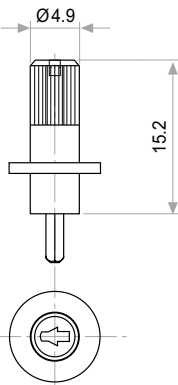Fig. 8 / Ref. 5116

Fig. 9 / Ref. 5119
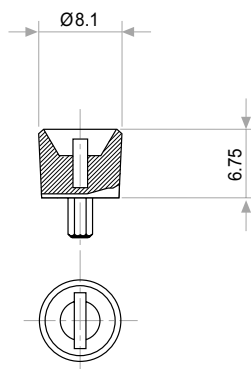
Fig. 10 / Ref. 5120
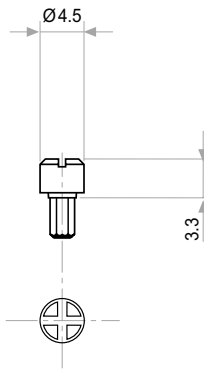
Fig. 11 / Ref. 5027

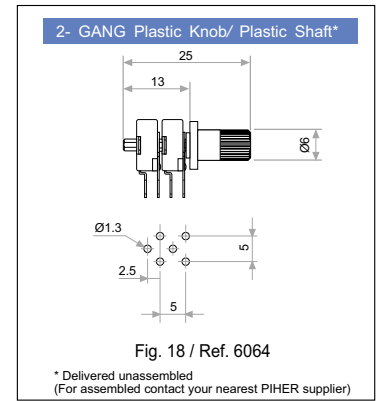Fig. 12 / Ref. 6052

Fig. 13 / Ref. 5121

Fig. 14 / Ref. 5055

2- GANG Plastic Knob/ Plastic Shaft*

Fig. 18 / Ref. 6064

* Delivered unassembled
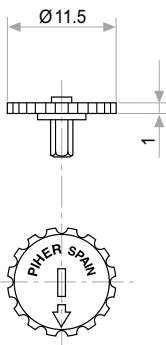(For assembled contact your nearest PIHER supplier)
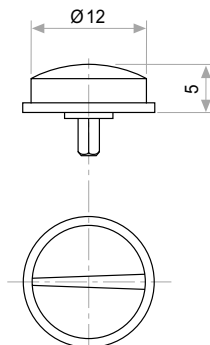
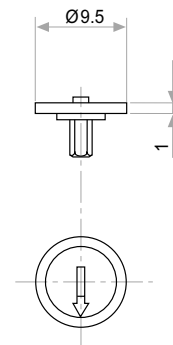## THUMBWHEELS
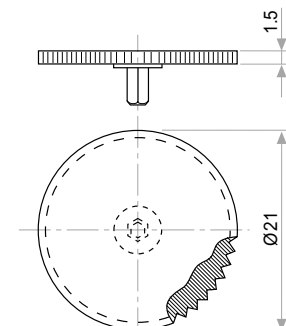
PIHER SPAIN

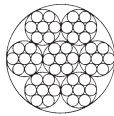Fig. 5 / Ref. 5034

Fig. 15 / Ref. 6008

Fig. 16 / Ref. 5039

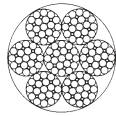Fig. 17 / Ref. 5062

16.3 Cables

Cables used for force transmission from motors to the legs - part number 26 ($d = 0.63, l \approx 220$): Câbles Inox AISI 316, Construction 7x7 (page 19)

Edelstahlseil 1.4401
Stainless steel wire rope AISI 316
Câbles Inox AISI 316

Konstruktion 7 x 7
Construction 7 x 7

Nenn-⌀ / Nom. dia / Dia nom. (mm)	Mindestbruchkraft / Min. break. load / Charge rupt. min. (N)	Gewicht / Weight / Poids (kg/1000 m)	Best.-Nr. / Ref. No. / Réf.
0,12	10	0,060	CG 077012
0,18	15	0,130	CG 077018
0,27	59	0,290	CG 077027
0,36	103	0,520	CG 077036
0,45	162	0,800	CG 077045
0,54	235	1,200	CG 077054
0,63	324	1,600	CG 077063
0,72	422 **	2,100	CG 077072
0,81	440 **	2,600	CG 077081
0,90	647	3,200	CG 077090
1,00	785	3,900	CG 077100
1,20	1075	5,000	CG 077120
1,35	1274	7,200	CG 077135
1,50	1800	9,200	CG 077150
1,80	1820 *	12,7	01551018
2,0	2260 *	15,7	01551020
2,5	3360 *	24,6	01551025
3,0	5060 *	35,4	01551030
4,0	9010 *	62,9	01551040
5,0	14100 *	98,3	01551050
6,0	20300 *	142,0	01551060

* Festigkeitsklasse 1570 N/mm². **nur in Werkstoff 1.4301 lieferbar.
* Tensile strength 1570 N/mm². **only available in AISI 304
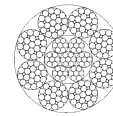* Fils de résistance 1570 N/mm². **Existe seulement en AISI 304

Edelstahlseil 1.4401
Stainless steel wire rope AISI 316
Câbles Inox AISI 316

Konstruktion 7 x 19
Construction 7 x 19

Nenn-⌀ / Nom. dia / Dia nom. (mm)	Mindestbruchkraft / Min. break. load / Charge rupt. min. (N)	Gewicht / Weight / Poids (kg/1000 m)	Best.-Nr. / Ref. No. / Réf.
0,45	152	0,810	CG 719045
0,60	270	1,440	CG 719060
0,75	417	2,100	CG 719075
0,90	613	3,500	CG 719090
1,00	765	4,400	CG 719100
1,20	976	5,600	CG 719120
1,35	1227	7,100	CG 719135
1,50	1590	9,000	CG 719150
1,75	2099	13,500	CG 719175
2,00	2768	17,000	CG 719200
2,50	3551	24,500	CG 719250
3,0	4690 *	34,200	01601030
4,0	8340 *	60,900	01601040
5,0	13000 *	95,200	01601050
6,0	18800 *	138,000	01601060
8,0	33300 *	243,000	01601080

* Festigkeitsklasse 1570 N/mm²
* Tensile strength 1570 N/mm²
* Fils de résistance 1570 N/mm²

Edelstahlseil 1.4401
Stainless steel wire rope AISI 316
Câbles Inox AISI 316

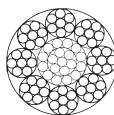### Konstruktion 8 x 19 + 7 x 7
Construction 8 x 19 + 7 x 7

Nenn-⌀ / Nom. dia / Dia nom. (mm)	Mindestbruchkraft / Min. break. load / Charge rupt. min. (N)	Gewicht / Weight / Poids (kg/1000 m)	Best.-Nr. / Ref. No. / Réf.
0,57	247	1,170	CG 819057
0,76	473	2,120	CG 819076
0,95	687	3,200	CG 819095
1,14	946	4,720	CG 819114
1,33	1279	6,400	CG 819133
1,52	1847	8,340	CG 819152
1,76	2100	10,360	CG 819176

Diese und andere Sonderkonstruktionen fertigen wir auf Anfrage. Mindestmengen erforderlich.
This and other special constructions are available on request. Minimum production quantities apply.
Cette construction ainsi que d'autres constructions spéciales sont fabriquées à la demande.

Edelstahlseil 1.4401
Stainless steel wire rope AISI 316
Câbles Inox AISI 316

Konstruktion 8 x 7 + 1 x 19
Construction 8 x 7 + 1 x 19

Nenn-⌀ / Nom. dia / Dia nom. (mm)	Mindestbruchkraft / Min. break. load / Charge rupt. min. (N)	Gewicht / Weight / Poids (kg/1000 m)	Best.-Nr. / Ref. No. / Réf.
0,44	172	0,760	CG 087044
0,55	270	1,200	CG 087055
0,66	387	1,700	CG 087066
0,77	495	2,300	CG 087077
0,88	696	3,000	CG 087088
0,99	824	3,900	CG 087099
1,10	1010	4,700	CG 087110
1,21	1226	5,700	CG 087121
1,32	1471	6,800	CG 087132
1,43	1716	8,000	CG 087143
1,54	1962	9,200	CG 087154

Diese und andere Sonderkonstruktionen fertigen wir auf Anfrage. Mindestmengen erforderlich.
This and other special constructions are available on request. Minimum production quantities apply.
Cette construction ainsi que d'autres constructions spéciales sont fabriquées à la demande.

Alle Litzen und Drahtseile werden, falls nicht anders angegeben, in der Festigkeitsklasse von 1770 N/mm² geliefert.
Unless otherwise specified, all strands and ropes are manufactured with a tensile strength of 1770 N/mm².
Tous les torons et câbles sont réalisés avec des fils de résistance 1770 N/mm² sauf mention contraire.