# The eXperimental Robot Project

Felix Schneider Norbert Braun
{felix,norbert}@xrpbot.org

EHSM 2014
2014-06-27

# Dingfabrik Köln

- Fablab, maker-/hackerspace
- Founded 2010 in Cologne
- $\sim$ 90 members
- Wood workshop, metal workshop
- Moved in 2013 to 450m$^2$ cellar

# Wood Workshop

- Professional circular saw
- Mitre saw
- 1200x600mm lasercutter
- Small, cheap 500x250x70mm CNC portal router
- Drill press
- All kind of handtools

# Metal Workshop

- Still in the making
- MIG, TIG, stick welding, gas axe
- Professional drill press
- Professional conventional universal mill
- TODO: sheet metal
- TODO: bandsaw
- TODO: plan table
- TODO: move lathe to the new dingfabrik

# Deckel FP2

- Built in 1978
- Donated by SGL Carbon in 2013
- Completly overhauled in 2014
- 400x200x500mm
- Digital readout
- Good results

# Misc

Electronics

- Small but fully featured
- Professional soldering iron, hot air
- 4-Ch 200MHz digital phosphor scope

3D printing

- Orcabot
- Prusa-Mendel

# Secret Underground Facility

## The lab in the lab

- Small 20m$^2$ room in Dingfabrik
- Project space granted for some longer time
- Home of the XRPBot team
- Fully featured electronic workbench
- Scope/pcb-making/parts

# XRP: Goals

The e**X**perimental **R**obot **P**roject

- Life-size humanoid robot
- Focus on legs (walking), arms and hands will come (much) later

- Fully free (open source, open hardware), transparent development process
- Goal: state-of-the-art software, hardware optimized for cost/manufacturability

# Why humanoids?

- Wheels ideal in dedicated environment (streets), otherwise fairly limited
- Human environments made for humans, wheels are really limiting (wheelchair!)
- Service robots
- Disaster recovery

- The real reason: they are cool...

# Other projects

- Progress on humanoids appears to be heating up
- Big company players (Boston Dynamics, Schaft) extremely secretive
- University projects more, but still not fully, open
- Exisiting robots cost $\geq$ 100 k€ (our goal: few k€)
- Physics-based character animation is a hot topic at SIGGRAPH (but usually not on physical hardware)

# Simulation: Introduction

- Simulate robot using simplified physics models
- Goal: develop controllers
- Goal: evaluate actuation requirements
- Goal: inform design choices

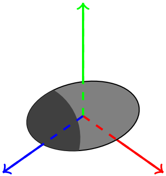
- Use Open Dynamics Engine (ODE, http://www.ode.org/) plus dedicated algorithms

# Rigid Body Dynamics

How to simulate a robot?

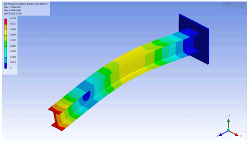
**Rigid body:**

- Non-deformable (no flexing, vibration, etc.)
- Details of mass distribution condensed into 10 parameters



Next step up in realism: **soft body**

- Complete details of mass distribution/stiffness/etc. matter
- Infinitely many degrees of freedom
- Simulation by finite element method



Wikipedia

# Rigid Body Dynamics (2)

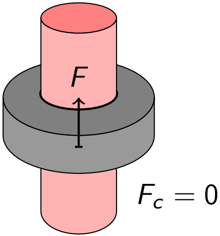
- 6 degrees of freedom:
  - Rotation (3 DoFs)
  - Translation (3 DoFs)
- 10 parameters:
  - Total mass $m$ (1 parameter)
  - Center of gravity $c$ (3 parameters)
  - Moment of inertia $I$ (6 parameters)

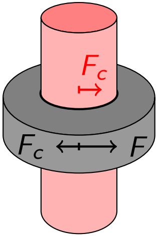
Newton-Euler equation: link between force $(T, F)$, velocity $(w, v)$ and acceleration $(\alpha, a)$.

$$\begin{pmatrix} T \\ F \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & m\mathbf{1} \end{pmatrix} \begin{pmatrix} \alpha \\ a \end{pmatrix} + \begin{pmatrix} \omega \times I\omega \\ \omega \times mv \end{pmatrix}$$

# Joints

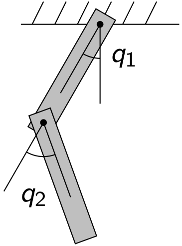
- Joints enforce constraints between rigid bodies.
- Motion respecting constraint unaffected
- Otherwise: constraint force occurs such that constaint remains fulfilled
- Actuated joint: force in active direction can be chosen



# Joint space dynamics

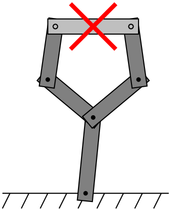
- Typical physics engine: simulate all 6 DoFs per body
- Alternative: consider only active degrees of freedom for each joint
- Question: equations of motion?



# Recursive Newton-Euler algorithm

Kinematic tree:

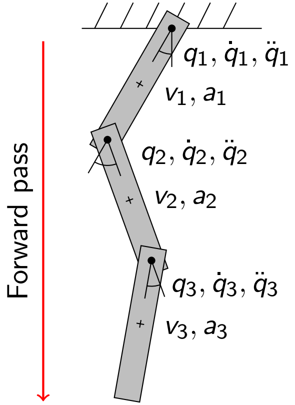
- Root body has joint to inertial (fixed) frame
- No loops



Recursive Newton-Euler algorithm (RNE):
inverse dynamics for kinematic trees
(given joint space velocity and acceleration $\dot{q}$, $\ddot{q}$, calculate joint space forces $\tau$)
addition: allow external forces $F^{(ext)}$

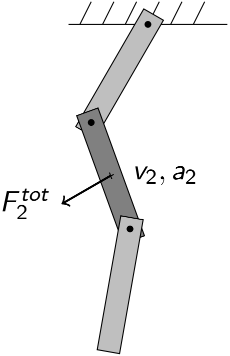
# RNE: forward pass

calculate $v_i$, $a_i$ from $q$, $\dot{q}$, $\ddot{q}$ for each body

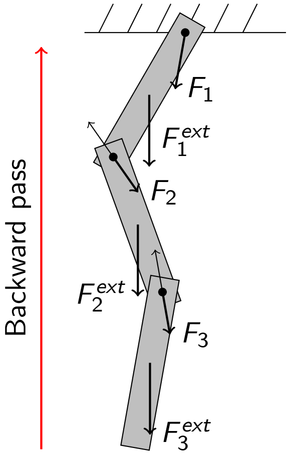


# RNE: local pass

use Newton-Euler equation to calculate total force on body from velocity and acceleration



# RNE: backward pass




- $F^{(ext)}$ given
- Solve

$$
\begin{aligned}
F_3^{tot} &= F_3^{(ext)} + F_3 \\
F_2^{tot} &= F_2^{(ext)} + F_2 - F_3 \\
F_1^{tot} &= F_1^{(ext)} + F_1 - F_2
\end{aligned}
$$

- project $F_i$ to get joint space forces $\tau$

# RNE properties

- Run-time: $O(n)$
- Constraint forces can be calculated

Analysis shows: $\tau$ is linear in $\ddot{q}$

$$\tau = M(q)\ddot{q} + C(q, \dot{q})$$

- $M(q)$: mass matrix (symmetric, positive definite, hence invertible)
- $C(q, \dot{q})$: Coriolis terms

# RNE properties (2)

Inverse dynamics:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})$$

Forward dynamics:

$$\ddot{q} = M(q)^{-1}(\tau - C(q, \dot{q}))$$

Note: forward dynamics requires matrix inversion, hence $O(n^3)$. Use Articulated Rigid Body algorithm if this is a problem.

**Reference:** R. Featherstone: Rigid Body Dynamics Algorithms (Springer 2008)

# Trajectory tracking

Track joint space trajectory $q_{des}(t)$ ($q_{des}(t), \dot{q}_{des}(t), \ddot{q}_{des}(t)$ given).

Control: $\tau$. Add small PD controller to correct modeling errors.

$$
\begin{aligned}
\ddot{q} &= \underbrace{\ddot{q}_{des}}_{feedforward} + \underbrace{k_p(q_{des}(t) - q(t)) + k_d(\dot{q}_{des}(t) - \dot{q}(t))}_{PD\ \ control} \\
\tau &= M(q)\ddot{q} + C(q, \dot{q})
\end{aligned}
$$

Remember: kinematic trees only!

# Walking with magnetic boots

- Idea: turn robot into kinematic chain by considering magnetic boots
- Above algorithms apply
- Design joint space trajectories, track them

**Demo #1**

- Simulation with ODE

# Simulation results

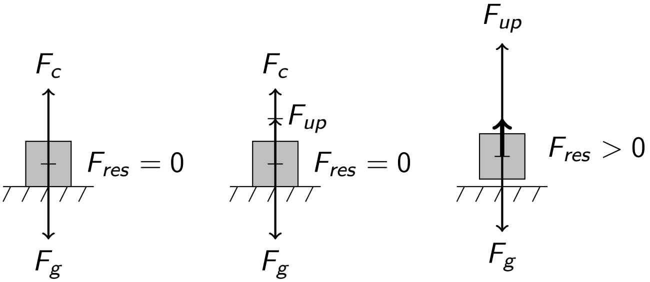
Works!
However, we have really only shown that RNE and ODEs algorithm agree.
Do we need the magnetic boots?

Demo #2

# Contact: normal component

Contacts are (usually) non-sticky!

Normal component of contact force: $F_c^{(n)} \geq 0$.

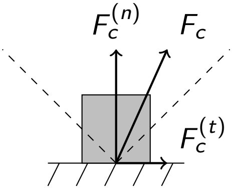


# Contact: tangential component

- Contact is a complicated microscopic phenomenon
- Commonly used model: Coulomb friction

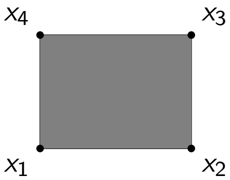
$$\left|F_c^{(t)}\right| \leq \mu F_c^{(n)}$$

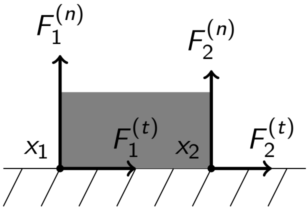
- Rubber soles on structured ground: $\mu \sim 1$
- Limited relevance in practice



# The center of pressure

Consider multiple contact points $x_i$:




Define:

$$x_c = \frac{\sum_i x_i F_i^{(n)}}{\sum_i F_i^{(n)}}$$

# Center of pressure (2)

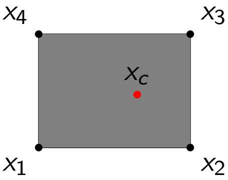
CoP:

$$x_c = \frac{\sum_i x_i F_i^{(n)}}{\sum_i F_i^{(n)}}$$

is weighted sum of contact points:

$$x_c = \sum_i \alpha_i x_i \ , \quad \alpha_i = \frac{F_i^{(n)}}{\sum_i F_i^{(n)}}$$

$F_i^{(n)}$ implies $0 \leq \alpha_i \leq 1$: convex sum!




$x_c$ must lie inside rectangle!

# Center of pressure (3)

Sum all contact forces into total contact force and pressure:

$$F = \sum_i F_i \,, \quad T = \sum_i x_i \times F_i$$

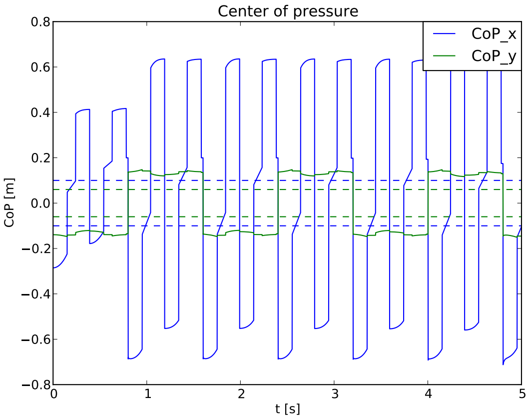
Let $n$ be the normal vector and coordinate origin in the contact plane. Then:

$$x_c = \frac{n \times T}{n \cdot F}$$

# Center of pressure (4)

- "Magnetic boots" can transfer arbitrary contact forces
- Necessary conditions for real contact:
  - $F^{(n)} \geq 0$
  - $x_c$ inside foot
- sufficient for $\mu \to \infty$
- usually sufficient in practice

# Walking with magnetic boots revisited



# Respecting the CoP constraint

- Cartwheel3d
- Buschmann

# Cartwheel 3d

- Physics-based character animation framework
- by S. Coros, P. Beaudoin and M. van de Panne
- Paper: S. Coros, P. Beaudoin and M. van de Panne. Generalized Biped Walking Control. SIGGRAPH 2010
- Open source (Apache 2.0)
- Originally indended for interactive authoring, **not** hardware control
- **https://code.google.com/p/cartwheel-3d/**

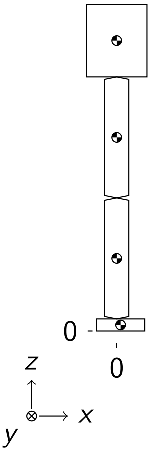
# Cartwheel 3d biped

- 6 DoF per leg
- Foot position and rotation fully controllable
- Analytical inverse kinematics
- Kinematic singularity for fully extended leg

- originally additional DoF in upper body

# Cartwheel 3d biped (2)

**Side view**




Pelvis/torso
Hip (HZ, HY, HX)

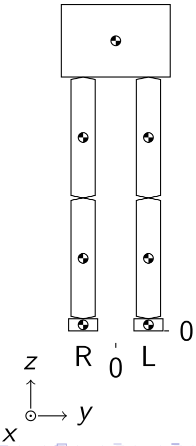
Thigh

Knee (KY)

Shank

Ankle (AY, AX)
Foot

**Front view**



# Approach: Cartwheel 3d

- Regulate CoM velocity with simple PD controller
- Clamp virtual CoM force using CoP constraint
- $\Rightarrow$ poor control over CoP trajectory, but
- use swing foot position on impact as additional control input

# Demo: Cartwheel 3d

- Simplified version of Cartwheel 3d controller
- Clean separation of controller and physics engine
- Physics engine: ODE

Demo time

# Demo summary

- Works. Looks realistic.

Drawbacks:

- trying to keep CoM velocity constant wastes control effort (minor)
- lost control over swing foot positioning (needed by higher-level controller, e.g. climbing stairs, rough terrain)
- Performance on physical robot unclear

# Buschmann controller

- Controller for physical robot (Lola, TU Munich)
- T. Buschmann. Simulation and Control of Biped Walking Robots. PhD thesis, TU Munich, 2010.
- No code, but reasonably complete description
- Our implementation work in progress

# Linear and angular momentum

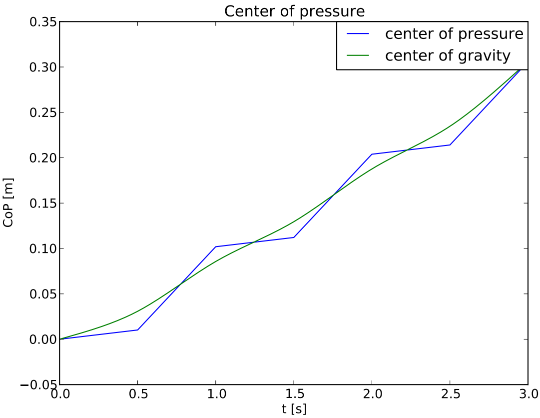
- Imagine: robot floating in space
- Linear and angular momentum conserved
- Conservation of linear momentum implies that center-of-mass trajectory cannot be influenced
- No similar result for angular momentum (can reorient!)

- Robot on ground: Total linear and angular momentum only changed through contact forces
- ... but we can control the contact forces through the legs!

# Buschmann: approach

- **Choose** CoP trajectory
- assume $L = \text{const.}$
- Solve BVP to obtain CoM trajectory
- design rest of robot movements around CoM trajectory

# Buschmann: demo



# Long term prospect: optimization

- Hand-crafted controllers OK for simple walking
- approach breaks down for complicated movements
- design movements by large-scale numerical optimization
- good way to use (still) increasing computational power
- many interesting results in simulation (SIGGRAPH)
- few results on physical robots: why?

# Gear Requirements

Ballpark estimates:

- Peak joint torque in order of 100 Nm
- Motor torque $\sim$ 1 Nm
- Needed reduction $\sim$1:100

Options left:

- Gearing: Harmonic Drives, Planetary Gears
- Linear actuators: Ball screws, Planetary Roller Screws

# Harmonic Drive

## Overview

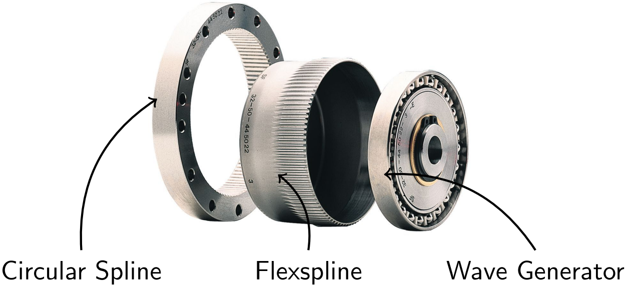
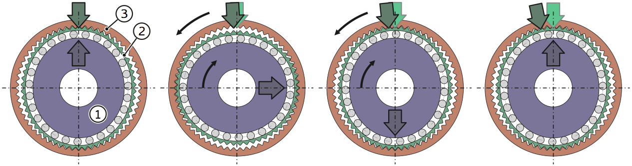



Image: Harmonic Drive AG

# Harmonic Drive

## How it works




1 Wave Generator | 2 Flexspline | 3 Circular Spline

- Reduction ratio:

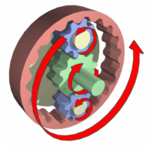
$$\frac{\text{Number of Flexspline Teeth}}{\text{Number of Flexspline Teeth - Number of Circular Spline Teeth}}$$

- E.g. $\frac{200}{200 - 202} = -\frac{1}{100}$
- Usual ratios: $\frac{1}{50}$ - $\frac{1}{200}$

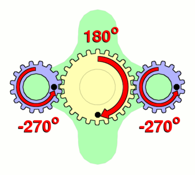
# Planetary Gear

## Overview

- Three main parts: Sun (green), Planet (blue), Annular Gear (red)
- Multiple Stages in a single Annular Gear possible
- $\frac{1}{1}$ - $\frac{1}{500}$



Wikipedia, Chris 73




Wikipedia, Guam

# Linear Actuators

Ball Screw

- works like a normal srew
- bearing balls are used to reduce friction
- no self-locking

Roller Screw

- order of magnitude more expensive
- increases contact area ->heavier load
- very shock resistent
- planetary roller screw combines planetary gear principle ->reduction

superiorballscrewrepair.com

servo-drive.com

## Comparsion

| | Planetary Gear | Harmonic Drive |
|---|---|---|
| Speed | - | + |
| Efficiency | 3% loss per stage | 87% |
| Backlash | - | ++ |
| Costs | + | - - |
| Weight | - | ++ |

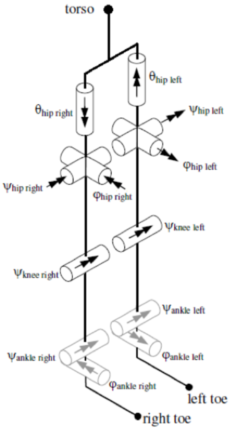
# TUlip

- Humanoid robot, realized at Eindhoven/Delft/Twente university
- 120cm, 15kg
- Uses *series elastic actuation* (resulting bandwidth: 5-10 Hz)
- Brushed motors (Maxon RE30, 60W)
- Planetary gears (Maxon GP32)
- Predecessor named *Flame*


TU Eindhoven

# TUlip: Kinematic concept

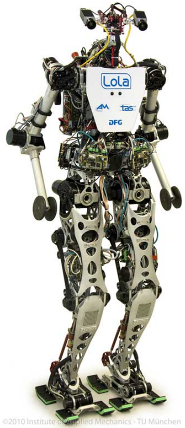
- 6 DoFs per leg: 3 hip, 1 knee, 2 ankle
- Hip Joint has 2 axis in 1 plane
- Third axis is in the torso
- Ankle roll axis is passive (spring)




TU Eindhoven

# Lola

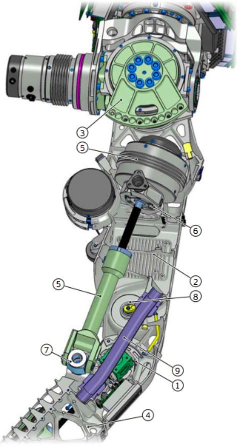
- Humanoid robot, realized at TU Munich
- 180cm, 55kg
- 25 DoF total, 7 DoFs per leg
- Predecessor named *Johnny Walker*




TU Munich

# Lola: Actuation concept

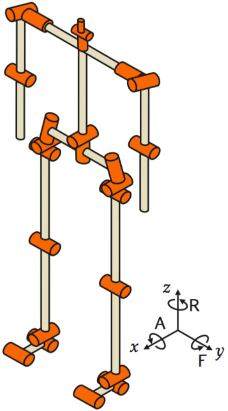
- Brushless motors (PMSM)
- Harmonic Drives (hip joint, toe joint)
- Planetary Roller Screws used as linear actuator (knee, ankles)




TU Munich

# Lola: Kinematic concept

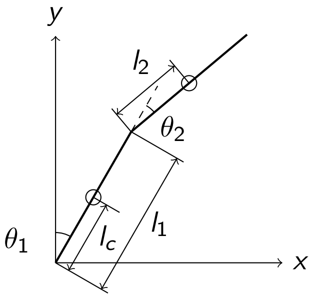
- 7 DoFs per Leg
- Comparable to TUlip
- Additional toe joint
- All joints are active
- Hip $z$ axis is tilted against $xy$ plane




TU Munich

# Acrobot: Introduction

- Double pendulum
- Only middle joint is actuated
- Task: swing up from hanging down
- Famous toy system from control



# Acrobot: Trajectory generation

- Black-box approach
- Insert:
  - Equations of motion
  - Start and goal position
  - Cost function
- out comes: feasible, locally optimal trajectory

- based on large scale, constrained, non-linear optimization
- Software: psopt (**http://www.psopt.org/**)
- Optimizer: ipopt (**https://projects.coin-or.org/Ipopt**)

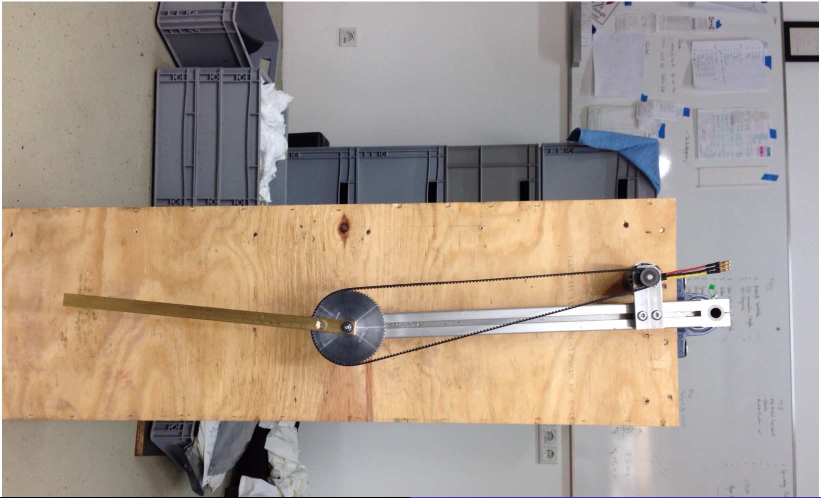
# Acrobot: Trajectory tracking

- Open-loop execution of trajectory will fail
- Feedback: complicated because underactuated system (2 DoFs, 1 control)
- Solution: linearize around nominal trajectory, use linear time-varying linear quadratic regulator (LTV-LQR)

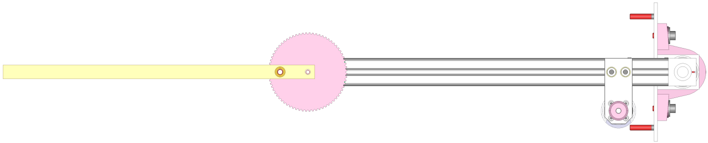
# Acrobot: references

- Details in upcoming blog post
- **Optimization-based control:** J. T. Betts: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming: SIAM, 2010
- **LTV-LQR:** R. Tedrake: Underactuated Robotics: Lecture series, MIT OpenCourseWare, http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-832-underactuated-robotics-spring-2009/video-lectures/

# Acrobot Hardware (1)



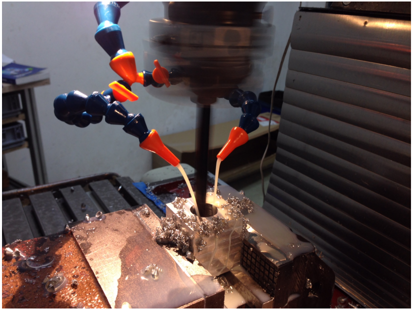
# Acrobot Hardware (2)

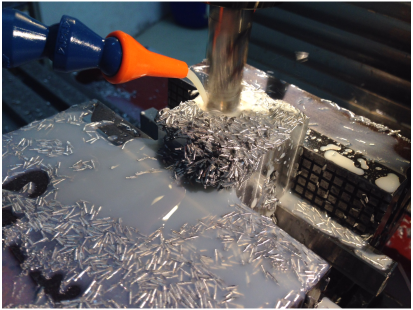


- Pulleys and extrusion profile purchased
- All other parts manufactured at Dingfabrik
- Complete STEP files on github

# Acrobot Hardware (3)

Manufacturing

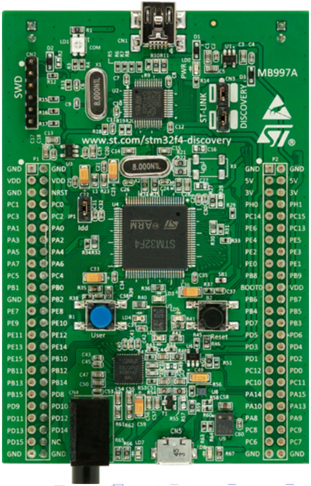




# $\mu C$ - Hardware

STM32F407

- ARM Cortex M4
- 164MHz, 1MB flash, 192kb RAM (newer models have even more)
- Huge set of peripherals
- Evaluation Board is about 15$
- Cheaper than a the chip alone
- Very well designed, probing is a breeze




ST Microelectronics

# $\mu C$ - Software

Library

- LibOpenCM3 (http://libopencm3.org)
- Good support for STM32
- More lightweight than original ST Library
- usually just works, but isn't stable

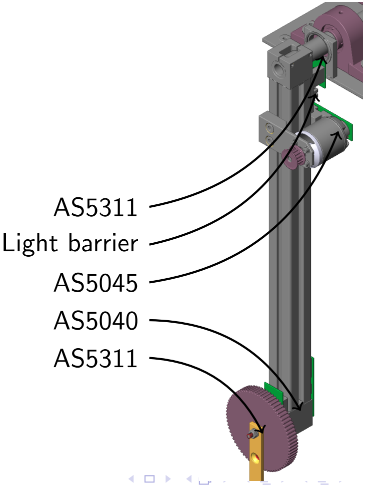
Toolchain

- arm-none-eabi gcc (precompiled by ARM)
- gdb over ST-Link (JTAG/SWD)

# Rotation Sensors

Austria Microsystems AS504x/AS5311

- Magnetic hall effect sensors
- absolute (AS504x) or incremental (AS5311)
- 12 bit (4096 steps/rev)
- about 10$ each
- magnets are about 5$
- quadrature output
- incremental ring sensor resolution: 0.0007°



# Electronic Speed Controller (1)

First approach:

- *simonk* compatible ESC
- 40A ESC with Atmel ATMega is 20$
- Caveat: fw is in assembly

Next approach:

- Copy known to work chinese ESC
- Own layout, own controller (STM32)
- Caveat: original layout mulitlayered. Custom board will be huge

# Electronic Speed Controller (2)

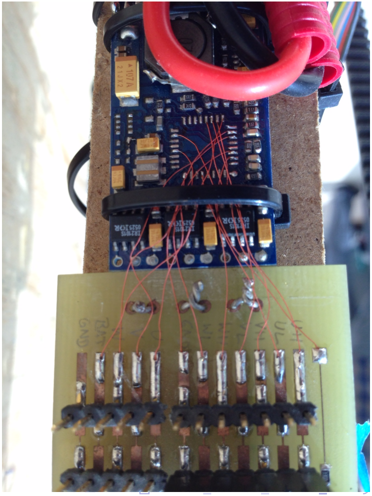
Conclusion:

**Decaptiate the Chinese ESC**

- Cheaper than the needed FETs alone
- Benefits of the newer ARMs (highres Timers, PWM)

Next step:

- *Space Vector Modulation*
- Think of it like *Microstepping*
- Finish a integrated PCB



# Motor

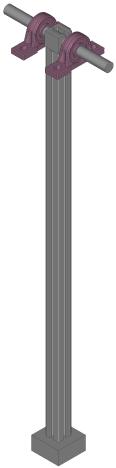
- Cheap 30$ 2kW BLDC RC Motor
- Weight: $\sim$ 500g
- Slightly overpowered but has only 270KV
- $\rightarrow$ 849 rad/s @ max. voltage
- Torque: 3.15 Nm @ max. current (calculated)

# Gears again

- Gears are expensive
- Idea: Use cheap gears from cordless screwdrivers
- Caveat: No exact, guaranteed specs
- backleash is a big uncertainty
- Solution: Motor test bed

# Motor Test Bed

- Static torque (point mass in plane): $M = Fr$
- Inertia $I = mr^2$
- Pendulum with 1m radius and 10kg point mass



# Back to Robots: Design Goals

- Size: 120cm (with torso as small as needed)
- Weight: 30kg

- Dynamic Walking
- Speed comparable to a human at same leg size

# Current status

- Preparatory phase: simulation, study exisiting designs
- Workshop mostly set up
- Toy project: Acrobot
- Next step: find suitable motor/gear solution
- Ready to start construction after gear question is solved

# Thank you!

http://xrpbot.org

...or meet us in the hall!