EUROPEAN
SPALLATION
SOURCE

Steps and motors in EPICS:
how to interface modern motion controllers

Torsten Bögershausen

Motion Control and Automation Group

Outline

- Motion control situation @ ESS
- Case: Brushed DC motor
- Case: Servo motor (common in industry)
- Case: Stepper motor (the working horse for science)
- Current motor abstraction in EPICS
- Difficulties & potential improvements
- Questions

Motion control situation at ESS

- Neutron instruments in an ”early phase”
- Motion control: look years into the future
- Focus on todays high-end motion controllers
- Integrate into EPICS
- Available integration into EPICS not ideal
- But first: Explanations …

Note: complex information is simplified

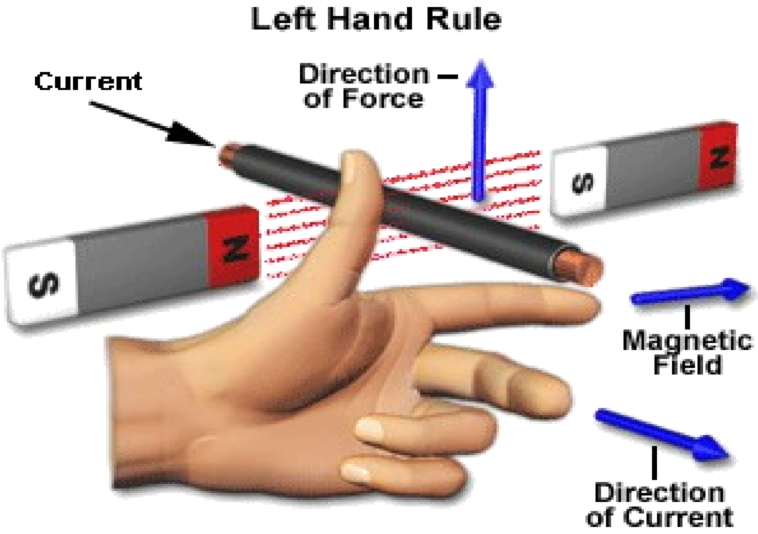
DC motor

- Magnetic field
- Current through a wire
- Lorentz force

-Wire is a "winding"

-Winding is rotating

-Brushes revert current ("rotor")



http://www.electrical4u.com/dc-motor-or-direct-current-motor/

DC motor characteristics

- More current -> more torque

- No current -> no torque

- Good in driving things

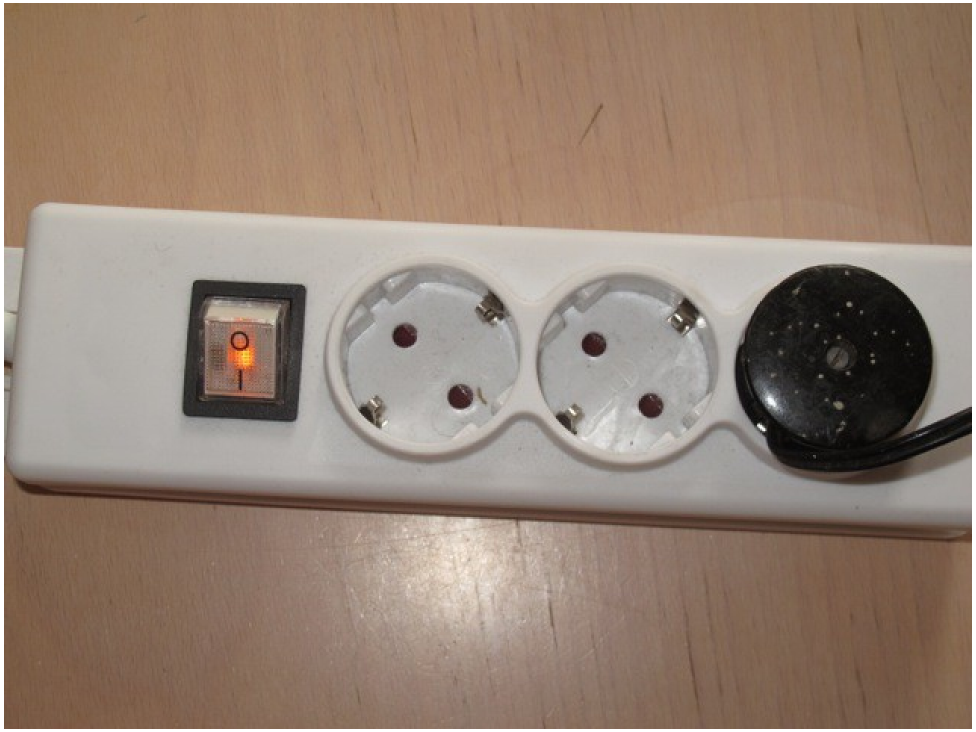
A driver motor

Is it controllable ?

Control an DC motor

Is it controllable ?
  Yes, on/off

In EPICS ?
  bo record



Position control

Assume we want to control a blade

Position control ?

Servo control

Motor stops where it stops - unless

Add a fine adjustable amplifier	$$$
Add a readback (encoder)	$$$
Add an advanced controller	$$$

Servo motor

Now we have a servo motor: DC + closed control loop

- Powerful machinery
- If applicable it's worth the money

[skip 100 pages discussion stepper vs servo]

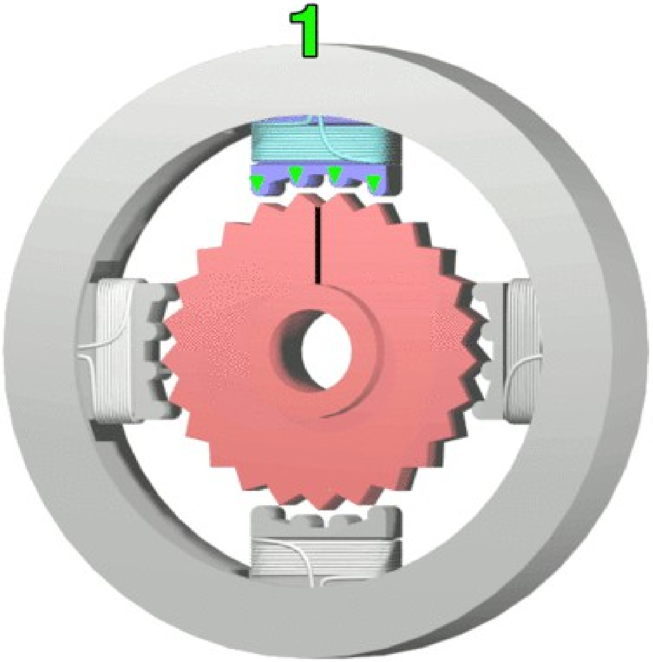
Lets look at a stepper motor

Stepper: Fundamentals of operation

Rotor has teeth with permanent magnets

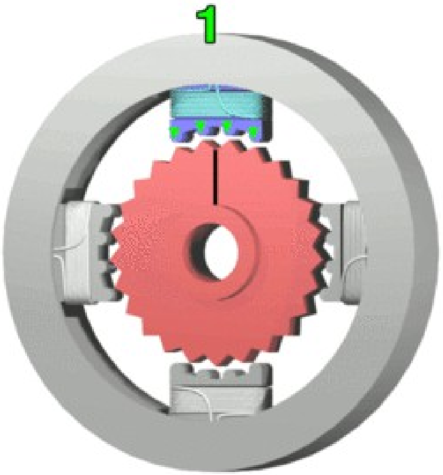
Stator has windings with teeth

Only at one winding the teeth align
  Exactly: either 1,2,3 or 4



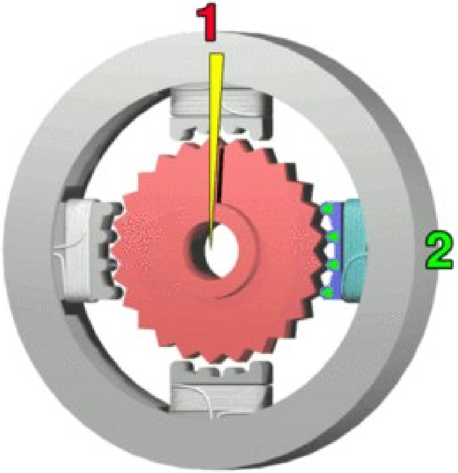
From wikipedia

Stepper: Step 1



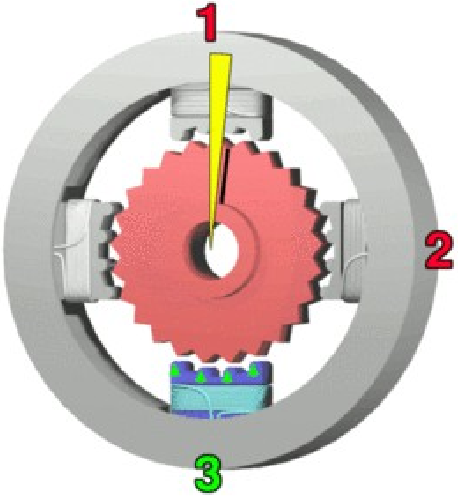
From wikipedia

Stepper: Step 2



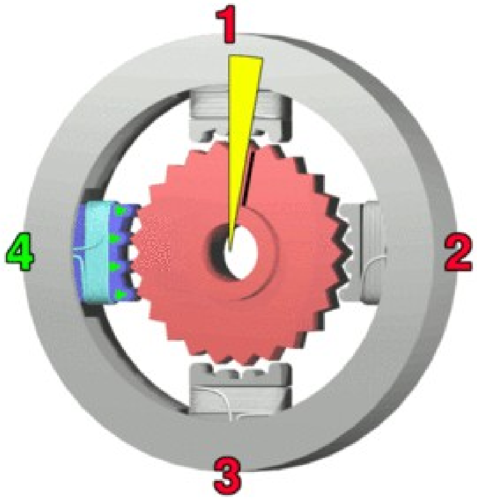
From wikipedia

Stepper: Step 3



From wikipedia

Stepper: Step 4



From wikipedia

Stepper motor characteristics

+ Holds position when powered off

+ Stops at a known position

+ Digital interface to the amplifier

+ Fits to computers: PDP 8...Arduino, RaspberryPI

+ Needs no sensor, no advanced electronics,
not so expensive

Stepper motor: limitations

- Teethish-Hackish movements

- when mis-configured:

”Resonances” at certain speeds

May loose steps or stall

+ Otherwise: runs reproducable in many years

Working horse (in scientif applications)

Modern motion controllers

- Fast feedback with *advanced controller* and encoder:
  - smoother driving (sinus/cosinus)
  - half step, step/4..step/64 == micro stepping

- Interface to the user (servo and micro-stepper)
  step size does not make sense: engineering units,

SW abstraction: motorRecord

+ All motors look the same to the user

+ Interface to user in engineering units (e.g. mm)

+ Lots of controllers are supported

- Always assumes that positions are a multiple of a step
- Servo motors are never exactly where they should be: they stay within a ”deadband”

Potential improvements

- Today: It's too complicated:

  a change in the motorRecord for one controller breaks another

+ Tomorrow: some code needs to move from record into driver(s)
+ Automated tests
+ Simulator

+ (pre) work started / ongoing
+ Focus on model 3

Future work

Anybody in the same boat?

Village People: ”There's no need to feel down -
there's a place you can go”

EPICS motor working group

Questions

- Thank you

- Questions ?