Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany

CONTROL FEATURES OF THE PLUNGING PICK-UP ELECTRODES WITH REAL TIME DIGITAL DATA PROCESSING

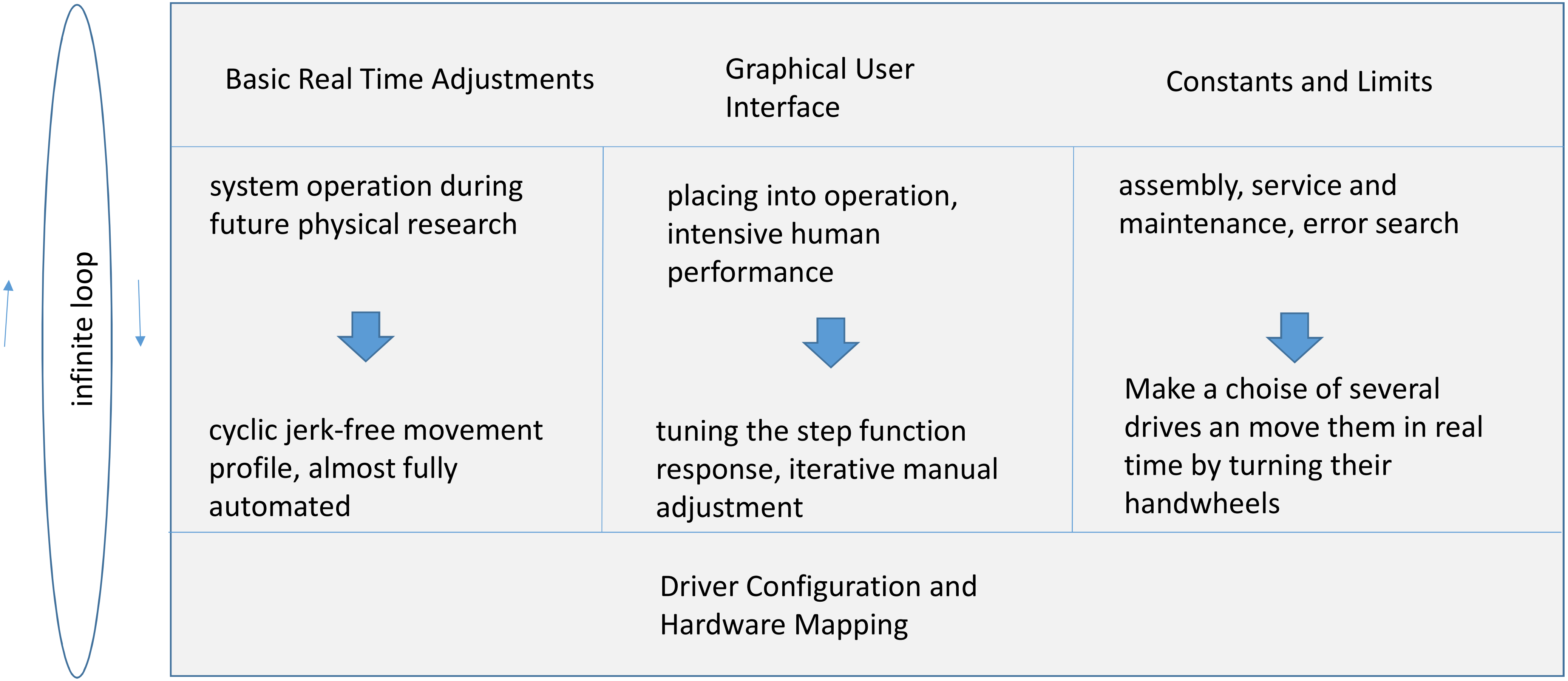
R. Hettrich, C. Peschke, R. Böhm, C. Dimopoulou

Abstract

The Pick-Up electrodes of the CR Stochastic cooling can be positioned very precisely and fast.
In normal operating state a function without jerk provides the set values for an underlying position control loop. Merging the electrodes however with the drive parts within a narrow tank is expected to be very challenging. For installation and service it might need a manual control facility, which allows to steer the mobile drive rods slowly to the connecting electrodes.
Hence eight hand wheels, one at each drive, are to be expected a manual positioning of each.
A star-shaped network from several wheel-controllers to a central computer was implemented.
A smooth and data saving transmission is intended to be achieved by the application of approved techniques from real time data processing. The equipment of analog drive systems with digital regulation and control systems allows to change the proportion between drive distance and angle of rotation of a hand wheel only by means of software.

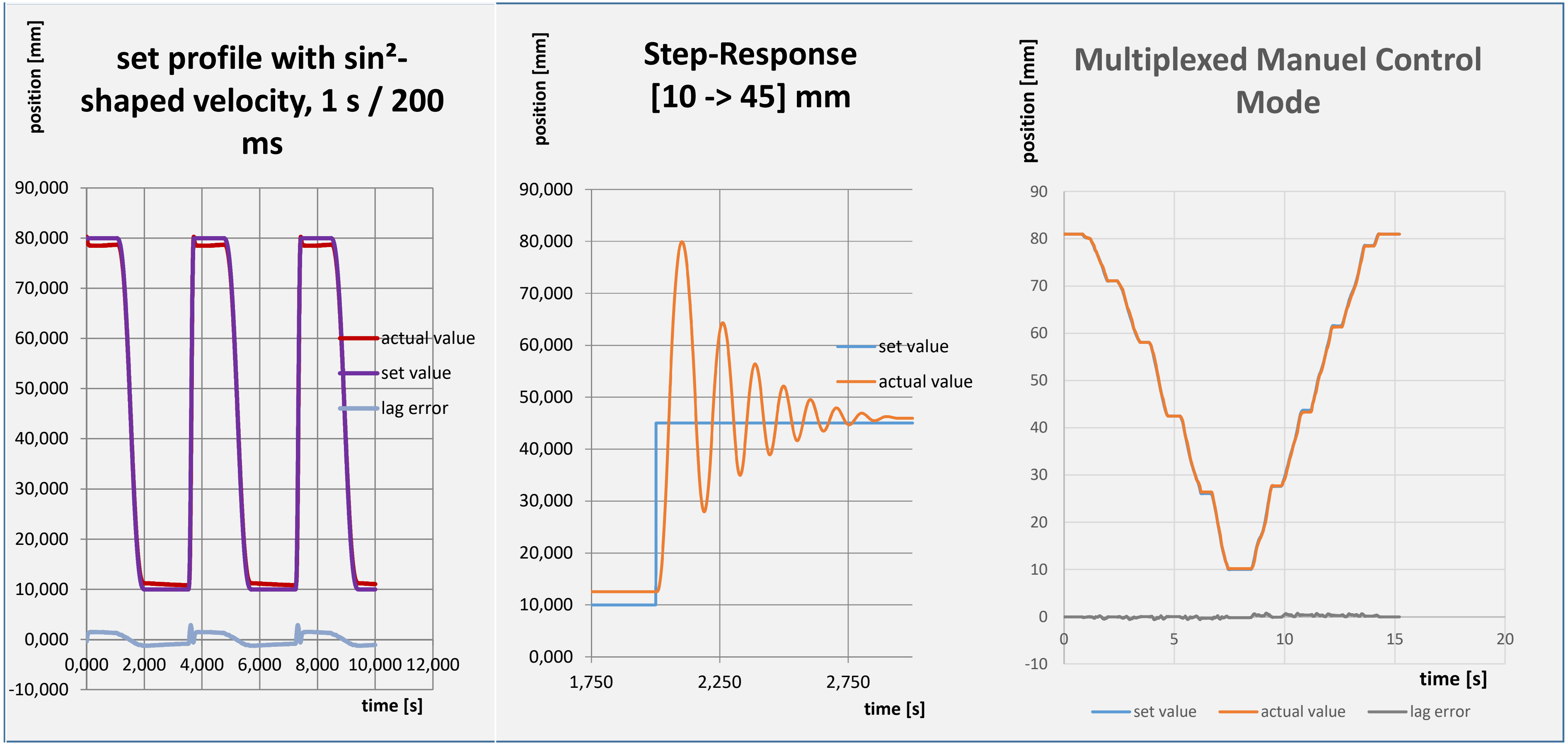
Concept of Operations for the CR Plunging Pickup Electrodes

endless Loop Framework with GUI



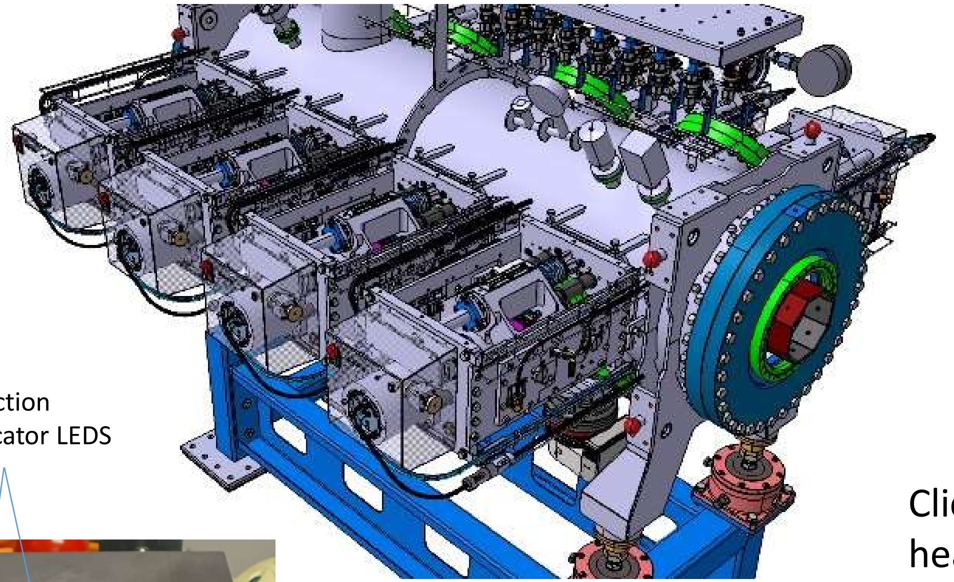
The Illustration Of Operation Modes

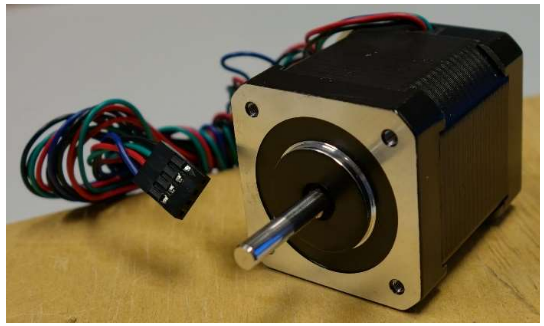
- Typical time responses illustrate the expected situations



The Mechanical Design Overview

The CR Pick-Up Tank with open Drive Enclosures

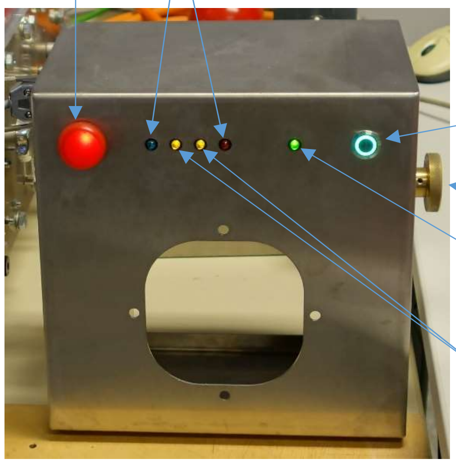




a stepping motor, misused as rotary encoder serves as convenient interface

blinking caution light

direction indicator LEDS



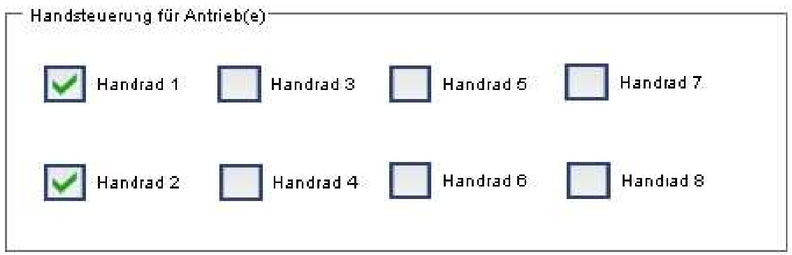
release button

steering wheel

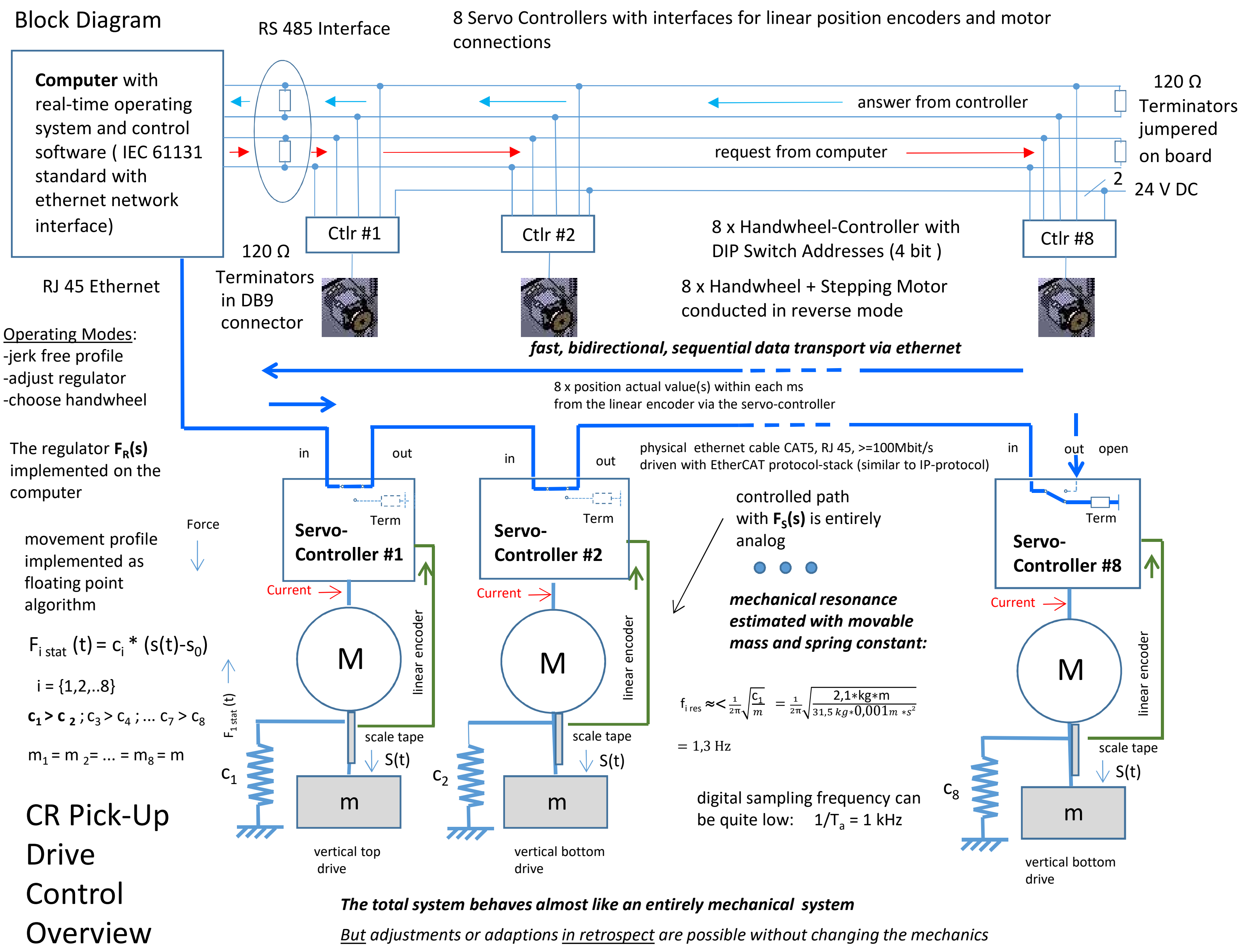
power LED

transmission control LEDs for each direction

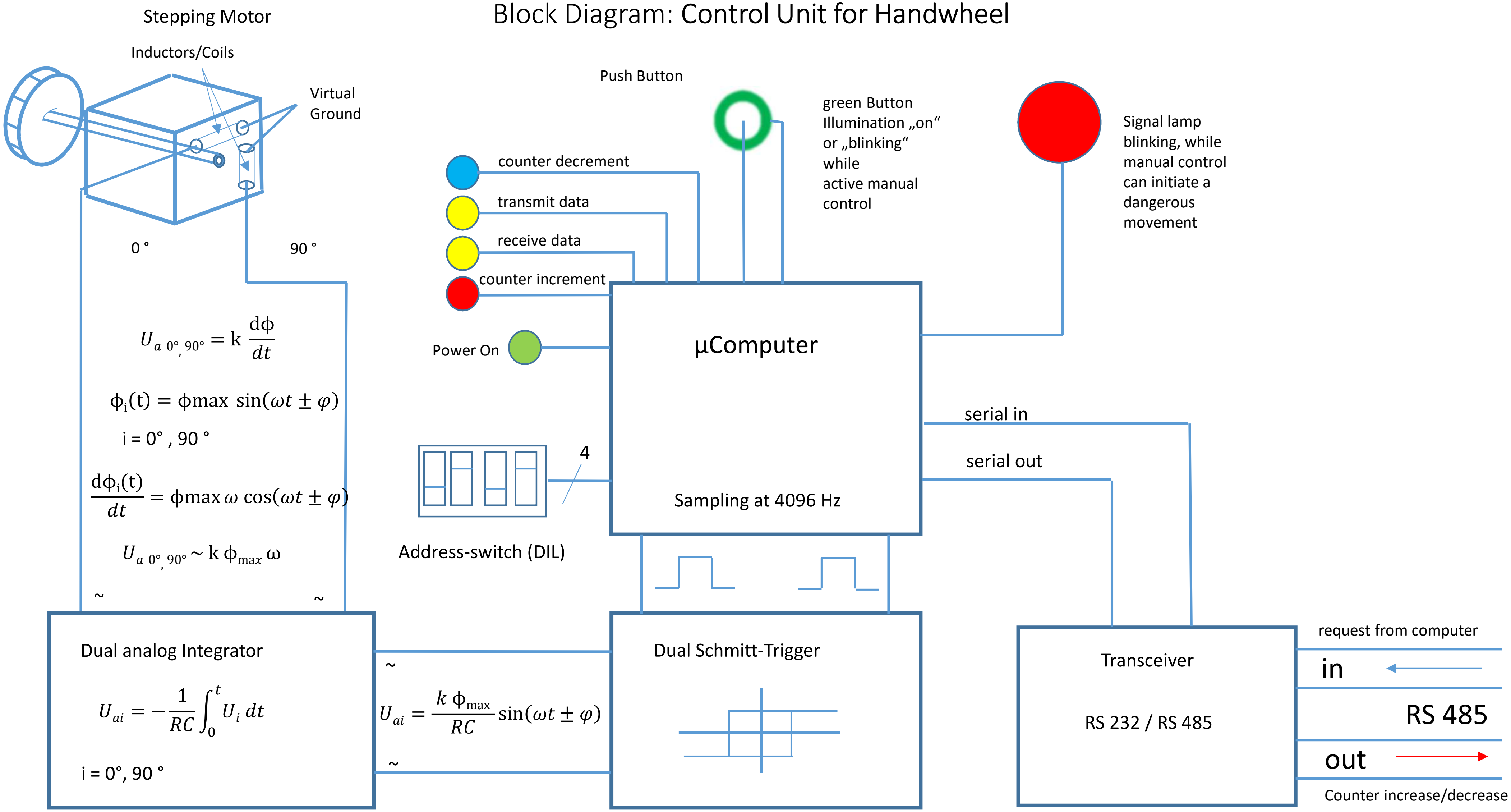
Click up to two drives of your choice and easily slide the heavy payloads just with the force out of your fingers!



Snapshot of the Graphical User Interface

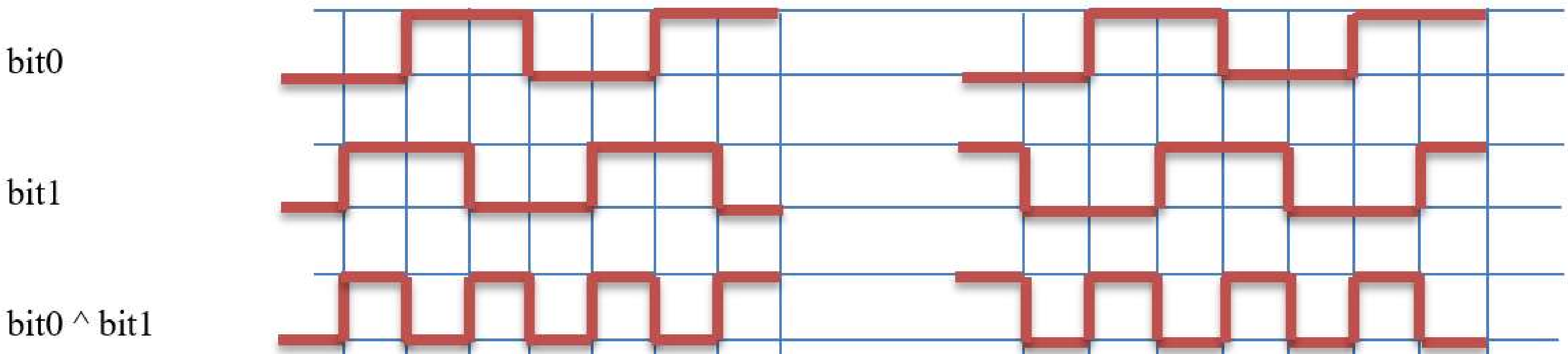

Block Diagram
RS 485 Interface
8 Servo Controllers with interfaces for linear position encoders and motor connections
Computer with real-time operating system and control software (IEC 61131 standard with ethernet network interface)
answer from controller
request from computer
120 Ω Terminators jumpered on board
2
24 V DC
Ctlr #1
Ctlr #2
Ctlr #8
120 Ω Terminators in DB9 connector
8 x Handwheel-Controller with DIP Switch Addresses (4 bit)
8 x Handwheel + Stepping Motor conducted in reverse mode
RJ 45 Ethernet
fast, bidirectional, sequential data transport via ethernet
8 x position actual value(s) within each ms from the linear encoder via the servo-controller
Operating Modes:
-jerk free profile
-adjust regulator
-choose handwheel
The regulator F_R(s) implemented on the computer
in
out
in
out
in
out
open
physical ethernet cable CAT5, RJ 45, >=100Mbit/s driven with EtherCAT protocol-stack (similar to IP-protocol)
Term
Servo-Controller #1
Servo-Controller #2
Servo-Controller #8
controlled path with F_S(s) is entirely analog
movement profile implemented as floating point algorithm
Force
Current
linear encoder
M
mechanical resonance estimated with movable mass and spring constant:
F_i stat (t) = c_i * (s(t)-s_0)
i = {1,2,..8}
c_1 > c_2 ; c_3 > c_4 ; ... c_7 > c_8
m_1 = m_2 = ... = m_8 = m
F_1 stat (t)
scale tape
S(t)
c_1
c_2
c_8
m
f_i res ≈< 1/2π √(c_1/m) = 1/2π √(2,1*kg*m / 31,5 kg*0,001m *s²)
= 1,3 Hz
digital sampling frequency can be quite low: 1/T_a = 1 kHz
CR Pick-Up Drive Control Overview
vertical top drive
vertical bottom drive
vertical bottom drive
The total system behaves almost like an entirely mechanical system
But adjustments or adaptions in retrospect are possible without changing the mechanics

Block Diagram: Control Unit for Handwheel



Evaluation of the Stepping Motor Pulses to reveal the Sense of Rotation

<u>forward</u> (red LED) <u>back</u> (blue LED)



	00	11	10	01	00	11	10	01	11	00	01	10	11	00	01	10
bit1|(bit0^bit1)	0	3	2	1	0	3	2	1	3	0	1	2	3	0	1	2
bit1|(bit0^bit1) $_z$	1	0	3	2	1	0	3	2	2	3	0	1	2	3	0	1
difference to previous value	-1	3	-1	-1	-1	3	-1	-1	1	-3	1	1	1	-3	1	1

Hence the evaluation algorithm arises from the last line of the above-mentioned table:

difference = (bit1(1) | (bit0(1) ^ bit1(1))) – (bit1(0) | (bit0(0) ^ bit1(0))) *

*: The first bit1 in the expression (bit1 OR (bit0 XOR bit1)) uses the digital value, left shifted by one, the second uses the pure value of the digit.

increment/decrement : = { +1 for difference = -1 ,
+1 for difference = +3 ,
-1 for difference = +1 ,
-1 for difference = -3 ,
0 for difference = 0 ,
0 for difference = 2 ,
0 for difference = -2 }

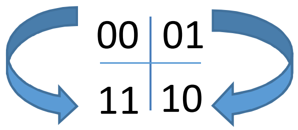
Hardware sampling frequency for the wheel signals equates to 4096 Hz, T_{sample} = 0,2441 milliseconds

´^´ corresponds to XOR, this means the logical ´exclusive OR´ operation.

´|´ corresponds to OR operation.

<u>forward</u>:

difference < 0
decrement counter



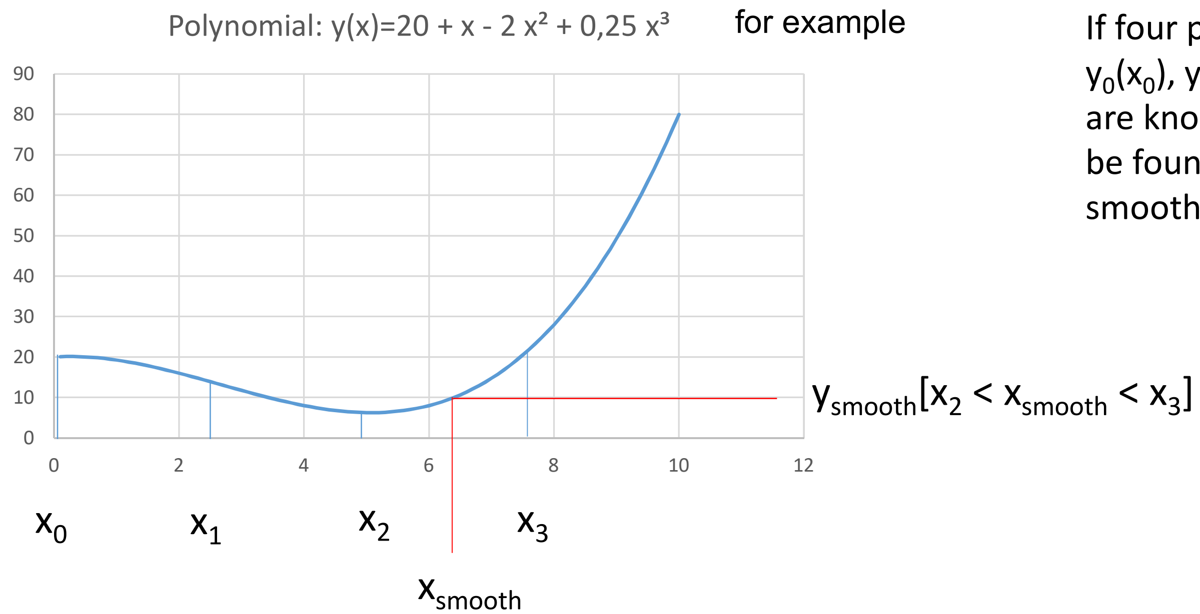
<u>backward</u>:

difference > 0
increment counter

New increments are read out every 0.25 milliseconds. This corresponds to a differential fetch rate of 4 kHz. This is preset within the real time based software on the controlling computer.

After transmission of increments to a central computer, the programmed calculator sums up them to a central Position:
counter: = counter + increment or counter: = counter – decrement or
counter: = counter + difference

Suppressing the Step Function of the Digital Sampling Process by Cubic Spline Interpolation



If four pairs of
$y_0(x_0)$, $y_1(x_1)$, $y_2(x_2)$, $y_3(x_3)$
are known, a three-order polynomial can be found, which connects all points in a smooth, jerk-free way.

Provided that the four consecutive sampling points are not sampled fast enough to avoid a stair-shaped function with a scraping noise, the past first points $y_0(x_0)$, $y_1(x_1)$, $y_2(x_2)$ and the endpoint $y_3(x_3)$ can be used to calculate a function, which delivers a continuous sequence between $[x_2, x_3]$. This sequence fits seamlessly to the previous ones and avoids an ugly sound. This method can be applied in real time by exploiting the four past sampling points to calculate the coefficients of a three order polynomial just before the next sequence starts. But the limitation is a latency time $(x_3 - x_2)$ until the drive follows.

$s_i(x)=a_i + b_i (x-x_i) + c_i (x-x_i)^2 + d_i(x-x_i)^3$

$s_i(x_2)=s_{i-1}(x_3)$	points fit
$s_i'(x_2)=s_{i-1}'(x_3)$	slopes fit
$s_i''(x_2)=s_{i-1}''(x_3)$	bendings fit
$s_i'(x_0)=b_0=\alpha$	incoming slope given
$s_i''(x_3)=c_3=0$	no output bending

Desired: $\mathbf{s_2(x)=a_2 + b_2 (x-x_2) + c_2 (x-x_2)^2 + d_2(x-x_2)^3}$ with
$c_1, c_2, c_0, d_2, d_1, d_0, b_2, b_1, \alpha, b_0$ where $a_2=y_2(x_2)$=known

The Algorithm As Final Result

Memorize 4 consecutive values of [position, timestamp]: $[y_0, x_0]$, $[y_1, x_1]$, $[y_2, x_2]$, $[y_3, x_3]$

calculate the 3 distances between the 4 points : $h_0 = x_1 - x_0$, $h_1 = x_2 - x_1$, $h_2 = x_3 - x_2$, let the slope be $\alpha = 0$ at start

After a convenient intervall of time, a new value of the position counter with the latest increment from the handwheel is sampled: $[y_{3new}, x_{3new}]$:

(entry point): set $y_0 := y_1$, $y_1 := y_2$, $y_2 := y_3$, $y_3 := y_{3new}$
set $h_0 := h_1$, $h_1 := h_2$, $h_2 := x_{3new} - x_3$ (ignore if equally spaced) , calculate :

$$c_1 = \frac{3\left[(-3)\frac{(a_1-a_0)}{h0}(h_1+h_2) + 2\frac{(a_2-a_1)}{h_1}\left(\frac{3}{2}h_1+h_2\right) - \frac{(a_3-a_2)}{h_2}h_1 + \alpha\,(h_1+h_2)\right]}{2\,(2\,h_1 + \frac{3}{2}h_0)\,(h_1+h_2) - h_1^2}$$

$$c_2 = \frac{3\left[\frac{3}{2}\frac{(a_1-a_0)}{h0}h_1 - 3\,\frac{(a_2-a_1)}{h_1}\left(h_1+\frac{1}{2}h_0\right) + \frac{(a_3-a_2)}{h_2}\left(2\,h_1+\frac{3}{2}h_0\right) - \frac{1}{2}\alpha\,h_1\right]}{2\,(2\,h_1 + \frac{3}{2}h_0)\,(h_1+h_2) - h_1^2}$$

Software Loop

The further coefficients ...

$c_0 = -\frac{1}{2}\, c_1 + \frac{1}{2\, h_0}\, [3\, \frac{(a_1 - a_0)}{h_0} - 3\alpha]$ $\qquad c_3 = 0;$

$d_2 = -\frac{c_2}{3\, h_2}$ $\qquad d_1 = -\frac{c_2 - c_1}{3\, h_1}$ $\qquad d_0 = -\frac{c_1 - c_0}{3\, h_0}$

$b_2 = \frac{(a_3 - a_2)}{h_2} - \frac{2}{3}\, h_2\, c_2$

$b_1 = \frac{(a_2 - a_1)}{h_1} - \frac{h_1}{3}\, (c_2 + 2\, c_1)$ $\qquad \alpha = b_1$ incoming slope for the following polynomial

$b_0 = \frac{(a_1 - a_0)}{h_0} - \frac{h_0}{3}\, (c_1 + 2\, c_0)$

$a_2 = y_2$ is already known

calculate $\mathbf{y(x) = s_2(x) = a_2 + b_2\, (x - x_2) + c_2\, (x - x_2)^2 + d_2\, (x - x_2)^3}$ in $[x_2 < x < x_3]$ and apply to drive(s)
The sliding carriage will follow without jerk or unpleasant sound.

return to the „entry point“ and repeat the loop as long as needed ..

Software Loop

Reference

R. Hettrich, Latest news from stochastic cooling developments for the collector ring at FAIR. https://doi.org/10.18429/JACoW-COOL2017-TUP16