ISBN: 978-3-95450-243-1 ISSN: 2226-0374 doi:10.18429/JACoW-COOL2021-P2003

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

R. Hettrich†, R. Böhm, C. Dimopoulou, C. Peschke,
GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany

Abstract

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

OPERATIONAL CONCEPT FOR THE CR PLUNGING PICK-UP ELECTRODES

The CR plunging Pick-Up electrodes are moved at cryogenic temperatures within a vacuum tank. They are propelled by linear motors outside this tank, vacuum sealed by bellows. Pre-compressed springs slightly overcompensate the vacuum force, thus avoid dropping the electrodes into the beam axis in case of a power failure. But using this drive construct for fast and precise movements requires damping the inherent resonance of a heavy mass with a strong spring. The solution is a control loop consisting of the motor, the moving payload, the motor controller, a position sensor and a regulator, whose set values can be forced due to almost any desired function [1]. A movement profile can be defined as a periodic function. This can be considered as the future "**automatic mode**". The digital design of the motor controller and the position sensor allowed to implement a digital regulator completely in software. This offers the opportunity to feed the regulator's set values for a desired position by any actuator with a convenient digital interface to the controlling computer. Thus the control software had been supplemented by two additional branches. The first can be considered as "**adjustment mode**" in order to set up the positon control loop. The second can be called "**manual mode**" and provides the input of a man-machine interface for a real time control of the drives. A stepping motor with a special interface hardware was equipped with a hand wheel, linked to the control computer and used as high resolution incremental encoder. While the hand wheel gets twisted, the pulses are counted and transferred to a controlling computer, which accumulates them to an absolute position set value for the underlying control loop. A change of the ratio between the turned wheel angle and the advance of the carriage is carried out classically by a gear transmission. The fully digital implementation allows the change of it without any additional mechanics – just by multiplying the motor pulses with another factor. Thus, each counted pulse has to be assigned to an actual distance, let's assume 100 µm. A continuous turn of the wheel results in a stairway-shaped function, whose steps have a minimum height and whose length lasts always one millisecond. The basic sampling frequency of 1 kHz is given by the motor controller and resides in the audible range. Decreasing the step size down to a few microns can avoid visible jumps, but due to the logarithmic sense of the human ear, the discrete steps remain audible. Thus, this raw kind of counting procedure ends up in a very unpleasant sound, approximately like a rusted machine without any grease. In order to overcome this, an equidistant choice of sampling points was connected by consecutive cubic spline interpolations. Then the ugly sound turned into gentle clicks.

This initial three-mode concept was enhanced from one to eight control channels by digital addressing and transmission techniques, shown on top of Fig. 1. It is proven with up to two manually controlled drives working virtually in parallel. The manual control feature was designed to facilitate all kinds of assembling, service and maintenance tasks. It is not intended for operation with beam.

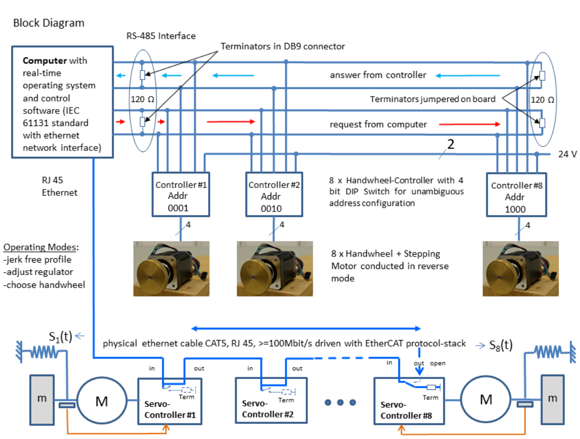

Figure 1: Linear drive control scheme for CR stochastic cooling pickup tanks, supplemented by eight addressable rotary encoders, attached to a four-wire RS-485 bus.

† r.hettrich@gsi.de

ISBN: 978-3-95450-243-1 ISSN: 2226-0374 doi:10.18429/JACoW-COOL2021-P2003

THE USE OF A STEPPING MOTOR AS ROTARY ENCODER

Turning the cross section of a coil within the flux of a permanent magnet induces an alternating voltage between both ends of the coil. Its amplitude and frequency grow proportional with the rotational speed. Therefore, the turning angle can be determined by counting the alterations of the voltage. To alleviate the counting process, a simple analogue electronic integrator can be connected to the coil, thus eliminating the dependence of the amplitude on the rotational frequency. The result is a sine wave with a constant amplitude and varying frequency. Provided a simple OP integrator is used, the circuit can be matched empirically by decreasing the characteristic capacitor until the generated signal just covers the full range of the supply voltage. Putting a full range sinewave signal to a Schmitt trigger, generates a full range square wave. An edge-triggered flip-flop can count the number of pulses. A cyclic read request from the controlling computer fetches the values and resets the flip-flop. The accumulation is done by the controlling computer. This explanation is simple, but it does not yet cover the sense of rotation. To resolve the direction information, an additional signal from a further coil is needed, which is oriented exactly 90° twisted in relation to the motor spindle. The outgoing voltages of a 2-phase stepping motor conform to this condition. The inductor magnet of the motor consists of twisted toothed wheels, one step of the motor corresponds to 1.8°. Provided that the logical evaluation is done by a microcontroller, the preceding hardware had just to be doubled. Two µC-inputs sample the information. A block diagram for the controller hardware is shown in Fig. 2. Its occurrence is eightfold per Pick-Up tank.

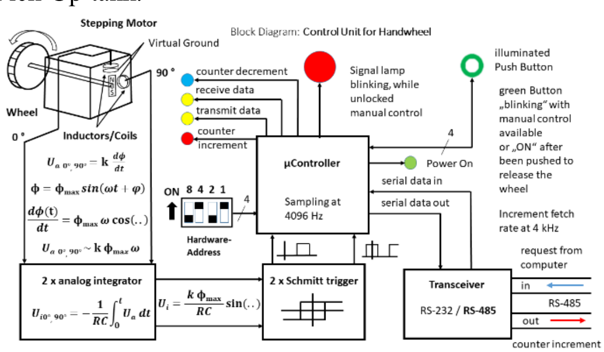

Figure 2: The structural components of a single manual control hardware with their distinctive signals.

The transmission to the controlling computer is done by a bus transceiver, which aligns the digital signals to the RS-485 standard. A four-bit hardware address can be adjusted on the controller board. It will be read by the microcontroller immediately after "switch on". Henceforward the controller only responds to requests directed to its own address. The microcontroller additionally serves as controller for some basic functions: Optical warning about dangerous movement, indicating several states via LEDs for power, for transmission, for wheel rotation direction, locking the manual control option via a dedicated release button. Figure 3 illustrates the full evaluation of the conditioned motor signals.

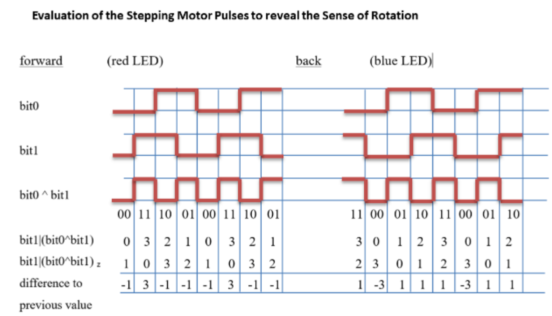

Figure 3: Evaluation of the stepping motor pulses to reveal the sense of rotation.

Assuming that the "bit0" signal is derived from the 0°-coil and the "bit1" signal is derived from the 90°-coil of the motor, a turn forward generates "bit1" 90° before "bit0" and a turn backward generates it 90° after "bit0". Combining bit0 with bit1 via an XOR (or ^) operation delivers a signal, that has twice the frequency as the one from the original coil. When this signal is added to the original one from bit1 by a logical OR-function, a two-digit binary number in the range [0, 3] arises. Turning forward generates a falling sequence [3, 2, 1, 0] and turning backward generates a rising sequence [0, 1, 2, 3]. The line "bit1|(bit0^bit1)" shows the possible values of the actual sampling. The line "bit1|(bit0^bit1) $_z$" shows the corresponding values taken one sample before. Subtracting the previous from the actual values always delivers: -1 or 3 while the wheel turns forward. It delivers 1 or -3 when the wheel turns backward. Equations (1), (2) and (3) show the algorithm, which derives itself from the last line of the above-mentioned table.

$$\text{difference} := (\text{bit1}(1) \mid (\text{bit0}(1)\ \text{XOR}\ \text{bit1}(1))) - (\text{bit1}(0) \mid (\text{bit0}(0)\ \text{XOR}\ \text{bit1}(0)))\ . \quad (1)$$

$$\text{increment} := \{ +1 \text{ for difference} = -1 \text{ or } +3, \; -1 \text{ for difference} = +1 \text{ or } -3, \; 0 \text{ for difference} = 0 \text{ or } 2/-2 \}\ . \quad (2)$$

$$\text{counter} := \text{counter} + \text{increment}\ . \quad (3)$$

The collection of digits is outlined in Fig. 4.

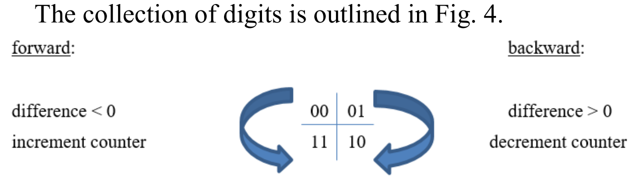

Figure 4: Illustration of the encoder algorithm.

The hardware sampling frequency is set to 4096 Hz in the µcontroller firmware. New increments are fetched every 0.25 millisecond. This corresponds to a transmission rate of 4 kHz and is set up as basic clock within the real time software on the superior control computer.

ISBN: 978-3-95450-243-1 ISSN: 2226-0374 doi:10.18429/JACoW-COOL2021-P2003

SUPPRESSING THE STEP FUNCTION OF THE DIGITAL SAMPLING PROCESS BY CUBIC SPLINE INTERPOLATION

A facility of a manual control of each single drive with its dedicated wheel was developed, so that a preferably high number of drives can be used virtually in parallel. Each wheel generates 200 pulses per turn. These pulses are collected by a control computer via a serial bus. Eight wheel controllers and the superior computer are attached to two twisted pair lines. The computer consecutively sends addressed requests to the wheel controllers, which answer with their increments. Between two answered requests from different wheels, the prior controller has to free the bus by switching its output into a high impedance state and the following controller has to occupy the bus by switching its output on. Due to the limited transfer rate of the serial RS-485 interface to 115200 bits/s, a rate of 4000 bus transmissions per second could not be exceeded. A virtual parallel operation would need 8 bus transmissions for sampling two drives. Thus the parallel update rate for each drive drops to (4000/8) s^{-1}=500 per second and below the basic sampling frequency of the motor controller of 1 kHz. The application of these values to the position control loop would result in a stairway-shaped function with higher and longer steps than described above for the "manual control mode". To avoid the scratching sound, only every forty-fifth accumulated value was transferred to the underlying position control loop. The missing values in between were calculated by a consecutive cubic spline interpolation in such a way, that each section between the remaining nodes fits seamlessly to the next. The drives follow with a time lag of 45 milliseconds, which appears just to be in real time. Provided that four pairs of $y_0(x_0)$, $y_1(x_1)$, $y_2(x_2)$, $y_3(x_3)$ are known with an incoming slope $s_0'(x_0) = b_0 = \alpha$, a three order polynomial with no output bend $s_3''(x_3) = 2\ c_3 = 0$ can be found, which connects all nodes in a smooth, jerk-free way. Figure 5 shows a graphical representation.

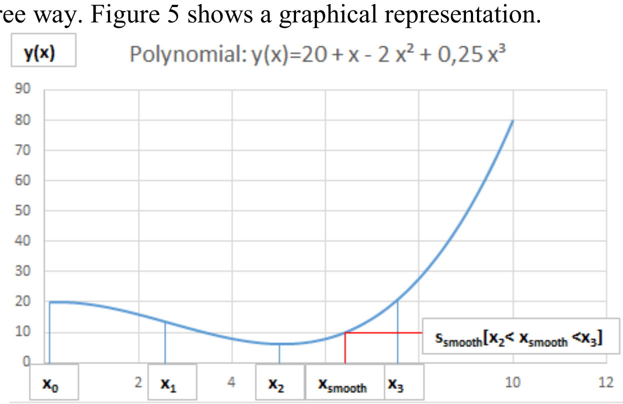

Figure 5: Example of a three-order polynomial.

Assuming that the first three nodes are stored in the past, the polynomial

$$s_2(x) = a_2 + b_2\,(x - x_2) + c_2\,(x - x_2)^2 + d_2\,(x - x_2)^3. \quad (4)$$

with c_1, c_2, c_0, d_1, d_0, b_2, b_1, $b_0=\alpha$ and $a_2=y_2(x_2)$ can be used to calculate a continuous sequence between $[x_2, x_3]$, which fits seamlessly to the previous ones. This method can be applied in real time by exploiting the four past samples to calculate the coefficients of a three order polynomial just before the next sequence starts. But the limitation is a latency time $(x_3\text{-}x_2)$ until the drive follows.

The algorithm is derived from the following equations and their known conditions with i=[0..2], j=[0..3]:

$$s_i(x) = a_i + b_i\,(x - x_j) + c_i\,(x - x_j)^2 + d_i\,(x - x_j)^3. \quad (5)$$

$$s_i(x_j) = s_{i+1}\,(x_{j-1}), \quad j \neq 0. \quad \text{points fit} \quad (6)$$

$$s_i'(x_j) = s_{i+1}'\,(x_{j-1}), \quad j \neq 0. \quad \text{slopes fit} \quad (7)$$

$$s_i''(x_j) = s_{i+1}''\,(x_{j-1}), \quad j \neq 0. \quad \text{bends fit} \quad (8)$$

$$s_i'(x_0) = b_0 = \alpha. \quad \text{entrance slope given} \quad (9)$$

$$s_i''(x_3) = 2\ c_3 = 0. \quad \text{no output bend} \quad (10)$$

The Algorithm as final Result

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

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

After a convenient interval of time [e. g. 45 milliseconds], a new position value with the latest increment from the wheel has been accumulated: $[y_{3new}, x_{3new}]$:

(**entry point**) set $y_0 = y_1$, $y_1 = y_2$, $y_2 = y_3$, $y_3 = y_{3new}$
set $h_0 = h_1$, $h_1 = h_2$, $h_2 = x_{3new}\text{-}\ x_3$,
ignore if equally spaced,
set $a_0=y(x_0) =y_0$, $a_1=y(x_1) =y_1$, $a_2=y(x_2) =y_2$, $a_3=y(x_3) =y_3$
calculate:

$$c_1 = \frac{3\left[(-3)\frac{(a_1-a_0)}{h0}(h_1+h_2) + 2\frac{(a_2-a_1)}{h_1}\left(\frac{3}{2}h_1+h_2\right) - \frac{(a_3-a_2)}{h_2}\,h_1 + \alpha\,(h_1+h_2)\right]}{2\,(2\,h_1+\frac{3}{2}h_0)\,(h_1+h_2) - h_1^2} \quad . (11)$$

$$c_2 = \frac{3\left[\frac{3}{2}\frac{(a_1-a_0)}{h0}\,h_1 - 3\,\frac{(a_2-a_1)}{h_1}\left(h_1+\frac{1}{2}h_0\right) + \frac{(a_3-a_2)}{h_2}\left(2\,h_1+\frac{3}{2}h_0\right) - \frac{1}{2}\alpha\,h_1\right]}{2\,(2\,h_1+\frac{3}{2}h_0)\,(h_1+h_2) - h_1^2} \quad . (12)$$

$$d_2 = -\frac{c_2}{3\,h_2}\ . \quad (13)$$

$$b_2 = \frac{a_3-a_2}{h_2} - \frac{2}{3}h_2 c_2\ . \quad (14)$$

$$b_1 = \frac{a_2-a_1}{h_1} - \frac{h_1}{3}(c_2 + 2\,c_1)\ . \quad (15)$$

$$a_2 = y_2\ . \quad \text{(already known)} \quad (16)$$

use $\alpha = b_1$ as input slope for the following iteration.

Calculate $y(x) = s_2(x)$ in $[x_2 < x < x_3]$ due to Eq. (4) and apply to the drives. The sliding carriage will follow without jerk or unpleasant sound.

RETURN to (**entry point**) and repeat the loop as long as needed …

REFERENCES

[1] R. Hettrich *et al.*, "Latest News from Stochastic Cooling Developments for the Collector Ring at FAIR", in *Proc. 11th Workshop on Beam Cooling and Related Topics (COOL'17)*, Bonn, Germany, Sep. 2017, pp. 64-67. doi:10.18429/JACoW-COOL2017-TUP16

[2] Arndt Brünner, "Kubische Splines", https://www.arndt-bruenner.de/mathe/scripts/kubspline.htm