LinuxCNC

Spindel synchronisieren

Meine verwendeten Komponenten.

Lenze Vector 8200

VEM Drehstrommotor

Initiator + Encoder

PC

mit Parallelport

Theorie - Konfiguration der Spindel

Ich konfiguriere in der HAL eine Spindel die das PWM Signal an den Regler gibt.

Ein Initiator bzw. der Encoder auf der Motorwelle soll die Drehzahl in LinuxCNC geben.

Ziel eine synchronisierte Spindel um z.B. G33.1 verwenden zu können.

Aber wie

HAL/INI - Konfiguration der Spindel (synchronisieren)

Als erstes erstelle ich in der INI eine Sektor [SPINDLE_9] und einige Variablen die dann von der HAL gelesen werden.
Damit kann man später die Spindel einfacher parametrieren.

```
[SPINDLE_9]
PWM_FREQ        = 1000          (Die PWM Frequenz soll hier mal 1000 Hz sein.
OUTPUT_SCALE    = 3600          (bei  ±10V die maximale Drehzahl)
ENCODER_SCALE   = 64            (Mein Encoder liefert 64 mpulse pro Umdrehung )
```

In der HAL erstelle ich mit den Komponenten pwmgen und encoder die Ausgabe und das Feedback.

```
################################################################################
# Spindel mit PWM über Parallelport mit  Synchronisation

loadrt pwmgen output_type=1
addf pwmgen.update servo-thread
addf pwmgen.make-pulses base-thread

setp pwmgen.0.pwm-freq  [SPINDLE_9]PWM_FREQ                 (hier wird z.B. der Wert aus der INI übernommen)
setp pwmgen.0.scale     [SPINDLE_9]OUTPUT_SCALE              (hier wird z.B. der Wert aus der INI übernommen)
setp pwmgen.0.offset 0
setp pwmgen.0.dither-pwm true

loadrt encoder num_chan=1                                    (Encoder mit Encoder.0 weiter verwenden)
addf encoder.update-counters base-thread
addf encoder.capture-position servo-thread

setp encoder.0.counter-mode     false                        (Countermode deaktivieren )
setp encoder.0.x4-mode          false                        (nicht mal 4 nehmen)
setp encoder.0.position-scale   [SPINDLE_9]ENCODER_SCALE     (hier wird z.B. der Wert aus der INI übernommen)

net spindle-sig-a   encoder.0.phase-A <= parport.0.pin-10-in     (PIN10 auf Encoder Signal A)
net spindle-sig-b   encoder.0.phase-B <= parport.0.pin-11-in     (PIN11 auf Encoder Signal B)
net spindle-sig-z   encoder.0.phase-Z <= parport.0.pin-12-in-not (PIN12 auf Encoder Signal Z)

net spindel-rps     motion.spindle-speed-in <= encoder.0.velocity                    (Istwert vom Encoder zum Planer)
net sollwert        motion.spindle-speed-out=> pwmgen.0.value                        (Sollwert vom Planer an das PWM geben)
net spindle-an      pwmgen.0.enable <= motion.spindle-on => parport.0.pin-17-out     (PIN17 soll ein Relais schalten damit der FU los legt )
net spindle-dr      motion.spindle-forward => parport.0.pin-14-out                   (PIN14 soll  Analogsignal  negieren = Drehrichtungswechsel)
net spindle-pwm     pwmgen.0.pwm => parport.0.pin-16-out                             (PIN16 soll das PWM Signal  rauskommen)
net spindle-sync    motion.spindle-index-enable => encoder.0.index-enable            (Planer sagt Encoder das  die Synchronisation  startet)
net spindel-pos     motion.spindle-revs <= encoder.0.position                        (Encoder teilt dem Planer die Spindelposition mit)

################################################################################
```

LinuxCNC

Spindel Konfiguration

http://talla83.de/linuxcnc/config.htm

http://linuxcnc.org/docs/2.7/html/man/man9/motion.9.html

http://linuxcnc.org/docs/2.7/html/man/man9/pwmgen.9.html

http://linuxcnc.org/docs/2.7/html/man/man9/encoder.9.html

http://linuxcnc.org/docs/2.7/html/gcode/g-code.html#gcode:g33.1