

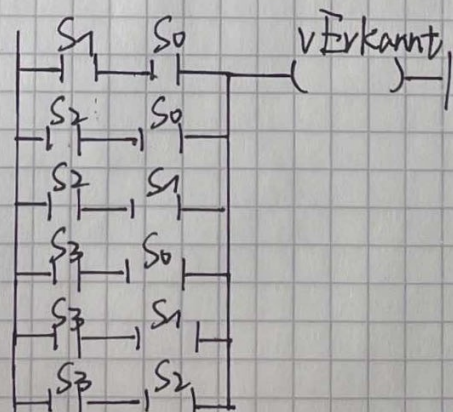
Aufgabe 1

7/9/2011 - Youtan Wang - RAS

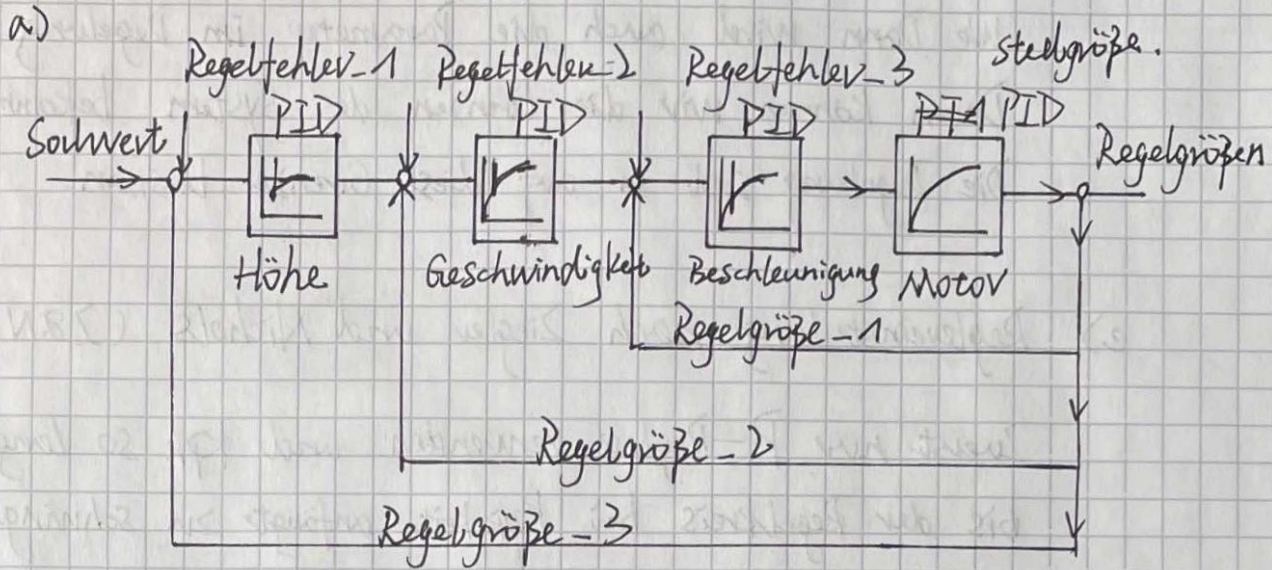
a)

S_3	S_2	S_1	S_0	vErkannt
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

	S_3, S_2			
S_1, S_0	0	0	1	0
	0	1	1	1
	1	1	1	1
	0	1	1	1



$$= \cancel{1}x = \cancel{x_1}x_0 + \cancel{x_2}x_0 + = S_1 S_0 + S_2 S_0 + S_2 S_1 + S_3 S_0 + S_3 S_1 + S_3 S_2$$



Sollwert: Ziel-Höhe.

Regelgröße:

Stellgröße: Befehl zum Motor

Regelgröße-1: Beschleunigung vom Sensor

Regelgröße-2: Geschwindigkeit vom Sensor

Regelgröße-3: Höhe vom Sensor

- b)
1. Es kann schnell reagieren.
 2. Es kann den Regelabweichung in Null regeln
 3. ~~Es ist~~ Einsatz bei hohen Anforderungen an Regelgeschwindigkeit und Genauigkeit.

c) Es ~~wird~~ ^{hat} Regelabweichung. Dann gibt es auch Differenz der Höhe. D.h. kann es nicht zum Ziel erreichen.

d) Wir können auch ein Sensor für System ersetzen, um die Zustand zu messen.

Wie Dann wird auch die Parameter im Regelung.
Dann können wir die Grenzen des System bekommen.
Die Regelung soll in auf diese Grenze basieren.

e) Reglereinstellung nach Ziegler und Nichols (Z&N)

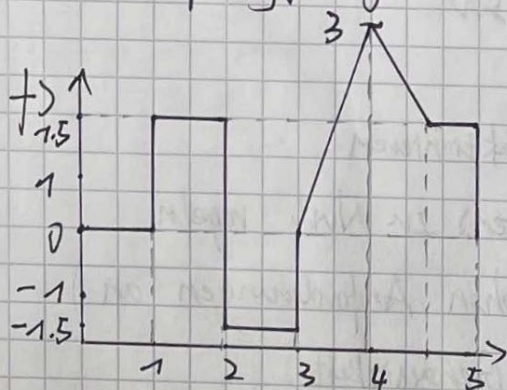
Zuerst nur P-Regler verwenden und KPR so lange erhöhen, bis der Regelkreis bei KPR krit anfangt zu schwingen

P-Regler : $K_{PR} \approx 0.5 K_{PR\text{ krit}}$

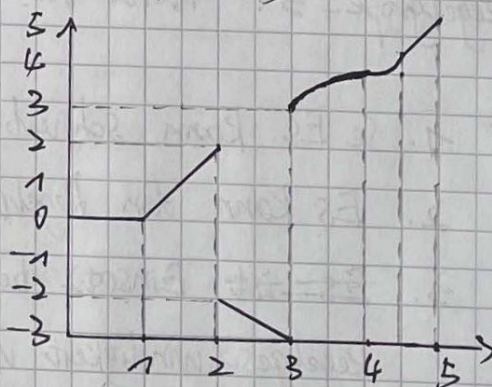
PI-Regler : $K_{PR} \approx 0.45 K_{PR\text{ krit}}$, $T_n = 0.85 T_p$

PII-Regler : $K_{PR} \approx 0.6 K_{PR\text{ krit}}$, $T_n = 0.5 T_p$, $T_v \approx 0.12 T_p$

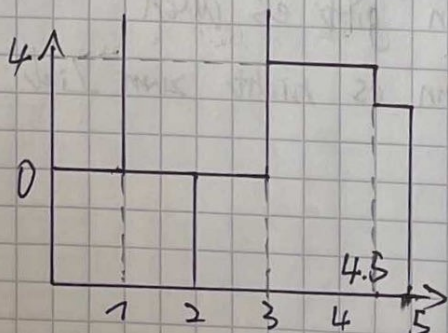
Es geht gutes Störverhalten, aber starke Schwingungsreizung bei sprungförmiger Änderung der Führungsgröße.



P-Anteil



I-Anteil



D-Anteil

Aufgabe 3

719511 - Youwan Wang - RAS

a) var STOP
var REORIENT
var GO
var BACK
var TURN
var AVOID
var TURNBACK

b) var state = STOP
var speed = 200
var obstacle = 1000
var Motorwerte = 0
timer . period [0] = 0

c) ~~on~~ onevent buttons
when button . forward == 1 and state == STOP do
state = REORIENT
end
when button . center == 1 do
state = STOP
end

d) onevent prox
if prox . horizontal [2] > obstacle then
state = BACK
end

e) onevent timer 0

if state == BACK then

motor . left . target = - Speed

motor . right . target = - Speed

timer . period [0] = 500

~~end~~

State = TURN

end

if state == TURN then

motor . left . target = speed

motor . right . target = 0

timer . period [0] = 1200

State = AVOID

end

if state == AVOID then

motor . left . target = Speed

motor . right . target = Speed

timer . period [0] = 2000

State = TURNBACK

end

if state == TURNBACK then

timer . period [0] = 1200

State = REORIENT

end

f) onevent acc