|  |  |  |
| --- | --- | --- |
| Übung Nr.: 1  Jahrgang: 5BHME  Gruppe: 3  Betreuer: SR |  | Protokollabgabe:  Solldatum: 16.12.2024  Ist-Datum:  Note:  Note Deutsch: |
|  |  |  |
| Protokoll | | |
| PID- Niveauregelung | | |
| THEMA: Regelungstechnik | | |

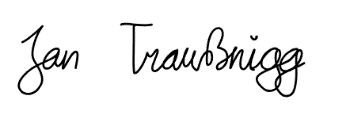
|  |  |
| --- | --- |
| Tag: | Montag, 09.12.2024 |
| Zeit: | 10:30-13:15 |
| Ort: | HTBLA Kaindorf, Messlabor |
| Anwesend: | Iris Bauer, Luca Pansi, Stefan Perr, Jan Traußnigg, Bowen Wang, Lukas Weiß |
| Abwesend: | Nico Hütter, Elias Gottsbacher |
| Schriftführer | Bauer Iris |

Aufgabenstellung

Auf Basis der Informationen aus dem Dokument *Niveauregelung\_2024.*doc soll eine grundsätzlich funktionierende PID-Niveauregelung mit der vorhandenen B&R-SPS realisiert werden.

Resümee

Der Source-Code für die Niveauregelung wurde fertiggestellt.



|  |  |  |
| --- | --- | --- |
| Jan Traußnigg | Ein Bild, das Handschrift, Kalligrafie, Schrift, Unterschrift enthält.  Automatisch generierte BeschreibungIris Bauer | Datum  29.12.2024 |

Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc187156336)

[Abbildungsverzeichnis 2](#_Toc187156337)

[1. Zeitlicher Ablauf 3](#_Toc187156338)

[2. Theorie 3](#_Toc187156339)

[2.1. PID-Anteile 3](#_Toc187156340)

[3. Aufgabenstellung 4](#_Toc187156341)

[4. Ergebnisse 4](#_Toc187156342)

[5. Verwendete Betriebsmittel 5](#_Toc187156343)

[6. Resümee 5](#_Toc187156344)

# Abbildungsverzeichnis

# Zeitlicher Ablauf

11:00 – 11:50: Grundlagen Niveauregelung

12:00 – 13:15: An Aufgabenstellung Arbeiten

# Theorie

Ein PID-Niveauregelungssystem ist ein Steuerungssystem, das dazu verwendet wird, den Füllstand einer Flüssigkeit in einem Behälter auf einem gewünschten Sollwert zu halten. Die Regelung erfolgt durch einen PID-Regler, der aus drei Komponenten besteht: dem proportionalen (P), dem integralen (I) und dem differenziellen (D) Anteil.

Das Prinzip basiert auf:

* Ein Sensor misst den aktuellen Füllstand der Flüssigkeit im Behälter (Istwert).
* Der gemessene Istwert wird mit dem gewünschten Sollwert verglichen, um die sogenannte Regelabweichung (Fehler) zu bestimmen:
  + e(t)=Sollwert−Istwerte
* Der PID-Regler berechnet auf Basis der Regelabweichung ein Regelsignal, das an den Aktor (z. B. ein Ventil oder eine Pumpe) gesendet wird, um den Flüssigkeitszufluss entsprechend anzupassen.

## PID-Anteile

P: Der proportionale Anteil erzeugt ein Regelsignal, das direkt proportional zur Regelabweichung ist:

* P=Kp⋅e(t)
* Kp ist Proportionalitätsfaktor

Der P-Anteil reagiert schnell auf Veränderungen, aber es kann eine bleibende Abweichung (Restfehler) entstehen.

I: Der integrale Anteil summiert die Regelabweichung über die Zeit auf:

* I=Ki⋅∫e(t) dt
* Ki ist Integrationsfaktor

Der I-Anteil korrigiert den Restfehler, indem er auch auf langfristige Abweichungen reagiert, aber kann zu Instabilität führen, wenn er zu stark ist.

D: Der differentiale Anteil berücksichtigt die Änderungsrate der Regelabweichung:

* D=Kd⋅de(t)dt
* Kd ist Differenzierungsfaktor

Der D-Anteil reagiert auf schnelle Änderungen und hilft, Überschwingungen zu reduzieren.

# Aufgabenstellung

Eine PID-Regelung zur Niveausteuerung in einem Behältersystem (Pumpenanlage im Messlabor) soll erstellt werden. Der Flüssigkeitsstand soll durch einen Drucksensor gemessen und auf einen Sollwert geregelt werden. Die Initialisierung, die zyklische Berechnung der Regelausgabe sowie die Begrenzung des Ausgangssignals sollen implementiert sein. Reglerparameter die gebraucht werden müssen erstellt und optimiert werden um ein stabiles Regelverhalten zu gewährleisten.

# Ergebnisse

#include <bur/plctypes.h>  
  
#define KONST 0.1/16\*100000\*1000/(1000\*9.81)  
#define SOLLNIVEAU\_W 300  
#define YH 10.  
#define XH 700.  
#define RICHT 1.  
  
#define KPR 10.  
#define TN 0.5  
#define TV 0  
#define DTC 0.1  
  
#ifdef \_DEFAULT\_INCLUDES  
 #include <AsDefault.h>  
#endif  
  
void \_INIT pidReglerInit(void)  
{  
 ton\_dt.PT = DTC\*1000;  
 ton\_dt.IN = 1;  
   
 istniveau\_x=0;  
 e=e1=0;  
 y=0;  
 summe\_e=0;  
}  
  
void \_CYCLIC pidReglerCyclic(void)  
{  
 TON(&ton\_dt);  
   
 if(ton\_dt.Q)  
 {  
 ton\_dt.Q=0;  
 ton\_dt.IN=0;  
 ton\_dt.M=0;  
   
 istniveau\_x = ((REAL)druck\_io\*20/32767-4)\*KONST;  
 e = SOLLNIVEAU\_W-istniveau\_x;  
 summe\_e = summe\_e+e;  
 y = RICHT\*YH/XH\*KPR\*(e+(DTC\*summe\_e/TN))+(TV/DTC\*(e-e1));  
   
 if(y<0)  
 y = 0;  
 if(y>YH)  
 y = YH;  
   
 e1 = e;  
   
 pu1\_io = (INT) (y\*32767.0/10.0);  
   
 ton\_dt.IN = 1;  
   
   
 }  
}

Der Code implementiert eine PID-Regelung für den Flüssigkeitsstand in einem Behälter. Der gemessene Druck wird in ein Niveau umgerechnet und mit einem Sollwert verglichen. Der PID-Regler berechnet basierend auf der Abweichung ein Ausgangssignal, das über ein Stellglied (z. B. Pumpe oder Ventil) den Füllstand anpasst. Der Timer sorgt dafür, dass die Berechnung in festen Zeitintervallen ausgeführt wird.

# Verwendete Betriebsmittel

* PC (Automation Studio)
* Netzteil
* B&R SPS

# Resümee

Das Programm für die PID-Niveauregelung wurde fertiggestellt. Theoretische Hintergründe zur Realisierung einer digitalen PID-Regelung wurden gelernt. In der nächsten Einheit werden wir die Regelung aufbauen und testen.

Die Regelparameter wurden bei der Entwicklung des Programmes angenommen und müssen bei aufgebauter Regelung noch optimiert werden.