

Praktikum Teil 1/3

Mit ★ gekennzeichnete Aufgaben sind Vorbereitungsaufgaben. Diese sind vor dem Praktikum schriftlich zu erledigen und Voraussetzung für die Teilnahme. Nicht vorbereitete Gruppen, oder Teilnehmer, dürfen nicht am Versuch teilnehmen. Eine MATLAB-Lizenz können Sie kostenlos über Ihren URZ-Account beziehen.

1 Regelungstechnik

Für die Bewegung eines Roboters sind Aktoren zuständig. Damit der Roboter definierte Trajektorie abfahren kann, oder eine bestimmte Pose einnimmt, müssen die Gelenke bezüglich ihrer Position, Geschwindigkeit oder ihrem Drehmoment geregelt werden.

1.1 Standardregelkreis

In Abbildung 1 ist ein Standardregelkreis dargestellt. Mit der Führungsgröße w , auch Sollgröße genannt, und der Regelgröße y wird die Regelabweichung $e = w - y$ gebildet, die dem Regler zugeführt wird. Zur Minimierung der Regelabweichung liefert der Regler die Stellgröße u , die zur Steuerung der Regelstrecke verwendet wird.

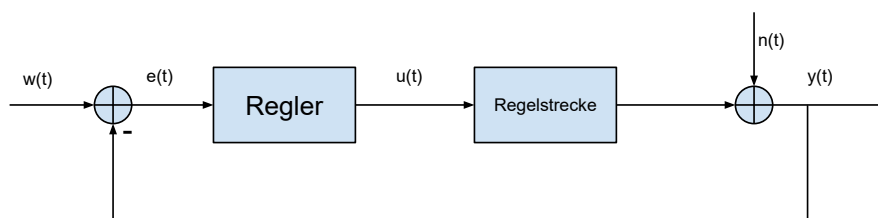


Abbildung 1: Standardregelkreis

1.2 PID-Regler

Ein in der Industrie oft verwendeter Regler ist der PID-Regler (Abbildung 2).

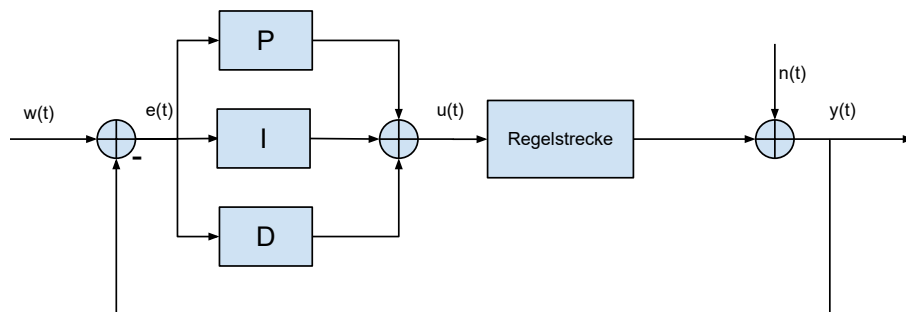


Abbildung 2: PID-Regler

P: Der Ausgang des P-Reglers ist proportional zu e .

$$u(t) = K_p \cdot e(t) \quad (1)$$

Pseudocode: $u = k_p \cdot e$

I: Beim I-Anteil wird die Regelabweichung e aufsummiert und verstärkt. Zudem stellt der I-Anteil die stationäre Genauigkeit des Reglers sicher. Bei einem digitalen Regler sollte e_{sum} begrenzt werden, da die Gefahr eines Überlaufs besteht.

$$u(t) = K_I \cdot \int_0^t e(\tau) d\tau \quad (2)$$

Pseudocode: $e_sum = e_sum + e;$

$u = k_i \cdot e_sum \cdot T_a;$

D: Der D-Anteil reagiert auf Änderung von e , sodass Sprünge in w oder y schnell ausgeregelt werden.

$$u(t) = K_D \cdot \frac{d}{dt}e(t) \quad (3)$$

```
Pseudocode:  e_delta = e - e_old;
               e_old = e;
               u = kd · e_delta/Ta
```

1.3 Reglereinstellung nach Ziegler-Nichols

Häufig ist die Regelstrecke unbekannt, sodass die Parameter des PID-Reglers ohne ein Modell durch “Ausprobieren und Erfahrung” eingestellt werden müssen. Ein geeignetes Verfahren zur Bestimmung der Reglerparameter liefert die Methode nach Ziegler-Nichols. Voraussetzung ist, dass das System mit den nicht-optimierten Reglerparametern schon stabil läuft oder eine Instabilität keine Schäden am System verursachen wird.

Schritt 1: P-, I- und D-Verstärkung auf 0 setzen.

Schritt 2: P-Verstärkung und K_P solange erhöhen bis $K_{P,Krit}$ das System zum Erreichen seines Endwerts mit der Periode T_{Krit} einschwingt.

| Regler | Parameter |
|--------|--|
| P | $K_P = 0,5 \cdot K_{P,Krit}$ |
| PI | $K_P = 0,45 \cdot K_{P,Krit}; \quad K_I = \frac{K_P}{T_n} = \frac{K_P}{0,85 \cdot T_{Krit}}$ |
| PD | $K_P = 0,55 \cdot K_{P,Krit}; \quad K_D = K_P \cdot T_v = K_P \cdot 0,15 \cdot T_{Krit}$ |
| PID | $K_P = 0,6 \cdot K_{P,Krit}; \quad K_I = \frac{K_P}{0,5 \cdot T_{Krit}}; \quad K_D = K_P \cdot 0,125 \cdot T_{Krit}$ |

1.4 Kaskadenregelung

Die Kaskadenregelung mit ihren unterlagerten Regelkreisen erzielt ein wesentlich besseres dynamisches Regelungsverhalten als ein einzelner Regler an sich. Hierbei dient die Ausgangsgröße des vorherigen Reglers als Führungsgröße des nachfolgenden Reglers. Am Beispiel des Positionsreglers (Abb.: 3) ist zu sehen, dass neben der Position auch Geschwindigkeits- und Beschleunigungsverhalten mit beeinflusst werden. Eine geringe Änderung der Position ist mit einem Positionsregler alleine nur schwer zu detektieren. Eine geringe Änderung der Position führt allerdings zu einer großen Änderung der Geschwindigkeit. Die Änderung in der Beschleunigung sind umso größer. Daher sind die Geschwindigkeits- und Beschleunigungsregler in der Lage viel schneller auf Änderungen in der Position zu reagieren. Die Kaskadierung erlaubt zudem eine Begrenzung der Geschwindigkeit und Beschleunigung auf die maximal erreichbaren Werte der Maschine.

Sind Änderungen an der Führungsgröße w bekannt, so kann mit einer Vorsteuerung direkt in die unterlagerten Regelkreise eingegriffen werden. Das dynamische Verhalten wird hiermit nochmals verbessert, da schon in die Regler eingegriffen werden, bevor es zu einer Regelabweichung durch Änderung der Führungsgröße w kommt.

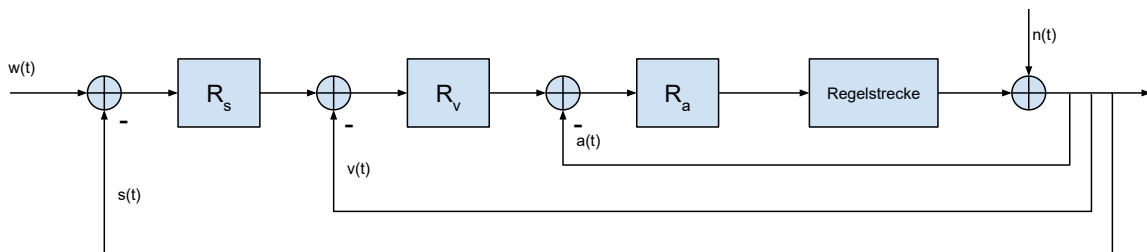


Abbildung 3: Kaskadenregelung

2 Hinweise zum Umgang mit dem Roboter

1. Nach dem Einschalten des Roboters muss dieser zunächst mit dem Handgerät in die Home-Position gefahren werden. Dazu muss die gelbe Taste gedrückt und gehalten werden, bis der Roboter die Home-Position erreicht.
2. Der Roboter besitzt keine Bremsen im Gelenk. Wenn der Roboter vom Strom getrennt wird, dann fällt dieser in sich zusammen. Beim Ausschalten sollte daher darauf geachtet werden, dass sich der Roboter in der Ausschaltposition befindet. Dazu muss der Roboter in die Home-Position gefahren werden (langes Drücken der gelben Taste). Aus dieser Position fährt der Roboter durch nochmaliges langes Drücken der gelben Taste in die Ausschaltposition.
3. Während sich der Roboter bewegt, solle sich immer eine Person in der Nähe des Roboters befinden, die Notfalls den Strom abschalten und den Roboter stützen kann.
4. Nach dem Praktikum sollte der Roboter vom Stromnetz getrennt werden.
5. Die maximale Gelenkgeschwindigkeit für die ersten drei Gelenke dürfen $36^\circ/s$ nicht überschreiten. Für die letzten drei Gelenke beträgt der Wert $48^\circ/s$.
6. Für die Gelenke gelten folgende maximalen Positionswerte:

| Gelenk | Min. | Max. |
|--------|-----------------|----------------|
| 1 | -10.000° | 10.000° |
| 2 | 50° | 310° |
| 3 | 19° | 341° |
| 4 | -10.000° | 10.000° |
| 5 | -10.000° | 10.000° |
| 6 | -10.000° | 10.000° |

3 Roboterprogramm

3.1 Kinova Development Center

Mit dem Kinova Development Center können Sie den Roboter steuern und Daten auslesen. Das Programm bietet eine übersichtliche Benutzerschnittstelle in der alle aktuellen Sensordaten aufgelistet sind. Des Weiteren lassen sich alle Gelenke mit der Maus bedienen. Im Praktikum können Sie dieses Programm zur Überprüfung Ihrer Programmierung nutzen. Dazu folgendes in das Terminal Programm eintippen:

```
cd /opt/kinova/GUI
./DevelopmentCenter
```

3.2 Roboter Programmierung

Der Roboter wird mit der Programmiersprache C++ programmiert. Schauen Sie sich dafür die entsprechenden Grundlagen vor dem Praktikum an. Als Programmierumgebung wird Eclipse verwendet. Für das Praktikum stehen fertige Programmbibliotheken für die Ansteuerung des Roboters bereit, die für die Implementierung der Regelung notwendig sind. Die wichtigsten Funktionen sind der folgenden Tabelle aufgeführt.

| Klasse | Beschreibung |
|------------------------|---|
| RobotThread | Erstellt einen Thread mit einer Schleife im Regeltakt |
| RobotIO::Update() | Schreibt Stellgrößen in den Regler und ließt Sensorwerte aus. |
| RobotIO::Init() | Stellt eine Verbindung mit dem Roboter her. |
| RobotIO::Close() | Trennt die Verbindung mit dem Roboter. |
| RobotIO::Home() | Bewegt den Roboter zur Home Position. |
| RobotIO::setVelocity() | Setzt die Geschwindigkeitswerte für das nächste Update. |
| RobotIO::get...() | Gibt die Messwerte aus dem letzten Update zurück. |

4 Aufgaben

1. a) ★ Was versteht man unter dem Begriff stationäre Genauigkeit? Welche Eigenschaft muss der Regler oder die Regelstrecke dafür haben?
- b) ★ Gegeben ist ein Regler mit folgender Übertragungsfunktion:

$$G_R(s) = K_P + \frac{K_I}{s} \quad (4)$$

Um welchen Regler handelt es sich?

- c) ★ Häufig wird für einen PD-Regler folgende Implementierung verwendet:

$$u = K_P \cdot e - K_{D,y} \cdot \frac{d}{dt}y + K_{D,w} \cdot \frac{d}{dt}w \quad (5)$$

Leiten Sie diese Formel mithilfe von Gl.:3 und Gl.: 1 her.

- d) ★ Wieso ist die Trennung des D-Anteils in dieser Implementierung sinnvoll?
 - e) ★ Schreiben Sie in Pseudocode einen Positionsregler mit unterlagerter Geschwindigkeitsregelung (Kaskadenregelung). Die max. und min. Geschwindigkeit soll begrenzt werden und als Regler sollen P-Regler verwendet werden.
2. Für die nachfolgenden Programmieraufgaben werden Musterdateien zur Verfügung gestellt, die Sie bitte vervollständigen.
 - a) Initialisieren Sie den Roboter und fahren Sie ihn anschließend zur Home Position.
 - b) Das erste (unterste) Gelenk soll mit konstanter Geschwindigkeit mit $5^\circ/s$ um seine eigene Achse rotieren.
 - c) Programmieren Sie eine Reglerklasse, die den Algorithmus für einen Positionsregler mit unterlagerter Geschwindigkeitsregelung enthält. Beachten Sie dabei, dass die max. Geschwindigkeiten nicht überschritten werden dürfen.

Zunächst soll ein P-Regler mit Verstärkung $K_P = 5$ verwendet werden. Die Reglerfunktion soll in der Hauptschleife ausgeführt werden.

- d) Erweitern Sie den P-Regler um einen I- und D-Anteil.
- e) Wenden Sie die Ziegler-Nichols Methode zur Einstellung der PID-Parameter an. Die gemessenen Positionswerte müssen hierzu in eine TXT-Datei abgespeichert werden. Anschließend können die Daten mit MATLAB geplottet werden.
- f) Vergleichen Sie die Regelabweichung zwischen P-Regler und PI-Regler. Was fällt auf und wie kommt das Ergebnis zustande?
- g) Programmieren Sie eine Stop-Funktion, die den Regler beendet, sobald die Endposition einschließlich einer Einschwingphase erreicht wird.