

## Reglerentwurf

In dem folgenden Abschnitt werden verschiedene Ansätze zur Regelung des Systems erläutert und bewertet. Die Aufgabe des Reglers besteht darin aus der Ausgangsgröße des Systems, dem Winkel des Würfels, die erforderliche Eingangsgröße, das Motormoment, zu ermitteln. Mit Hilfe eines Simulinkmodelles können die verschiedenen Regler konfiguriert und erprobt werden.

### Simulation in Simulink

Das dynamische Verhalten des Würfels kann mit den folgenden Differentialgleichungen bestimmt werden.

$$(\Theta_K^{(A)} + m_w \cdot l^2) \cdot \ddot{\phi}_K = (m_K l_K + m_w \cdot l) \cdot g \cdot \sin(\phi_K) - C_K \cdot \dot{\phi}_K - T_M$$

$$\ddot{\phi}_r = \frac{(\Theta_K^{(A)} + \Theta_R^{(B)} + m_w \cdot l^2) \cdot (T_M - C_R \cdot \dot{\phi}_r)}{\Theta_R^{(B)} \cdot (\Theta_K^{(A)} + m_R \cdot l^2)} - \frac{(m_K \cdot l_K + m_R \cdot l) \cdot g \cdot \sin(\phi_K) - C_K \cdot \dot{\phi}_K}{\Theta_K^{(A)} + m_R \cdot l^2}$$

Folglich kann ein Simulinkmodell entworfen werden, welches dieses Verhalten simuliert. Bei dem Sollwert  $\phi_{Soll} = 0$  handelt es sich um einen instabilen Gleichgewichtspunkt. Um dieses Verhalten auf das Modell zu übertragen muss eine zufällige Störgröße zur Winkelbeschleunigung des Körpers addiert werden. Der maximale Störfaktor entspricht dem Gravitationsmoment bei einem Ausfallwinkel von fünf Grad.

Der DC-Motor als Stellglied wird als PT2-Glied modelliert. Die Streckenverstärkung wird unter der Voraussetzung, dass der Regler ein Drehmoment berechnet, auf eins gesetzt. Die Zeitkonstanten entsprechen der mechanischen und elektrischen Zeitkonstante des Motors und können dem Datenblatt entnommen werden.

### Bewertung eines Reglers

Die entworfenen Regler müssen nach qualitativen Merkmalen bewertet werden. Bei der Wahl solcher Merkmale ist auf die speziellen Bedingungen des Anwendungsfalles zu achten. Da das Aufspringen des Würfels näherungsweise deterministischer und zeitinvarianter Vorgang ist und für die Ermittlung des richtigen Sprungmomentes lediglich ein eindimensionaler Raum durchsucht werden muss, kann für die Regelung angenommen werden, dass diese bereits in der unmittelbaren Nähe des Sollwertes startet. Außerdem ist dieser Sollwert konstant, somit ist das Verhalten des Regelsystem auf einen Sollwertsprung irrelevant. Folglich besteht die Aufgabe der Regelung im Normalbetrieb lediglich darin, Ungleichgewichte und kleine Störeinflüsse zu kompensieren. Dies kann simuliert werden in dem ein zufälliges Störmoment auf den Körper gegeben wird. Das qualitative Merkmal der Regelung ist das Integral der quadratischen Regelabweichung über einen fixen Zeitraum.

Außerdem muss die Regelung in der Lage sein Störimpulse, wie z.B. das Anstoßen des Würfels, auszuregulieren. Hierfür wird das Modell zusätzlich durch zyklische Störimpulse beeinflusst. Zur Beurteilung der Regelung wird wiederum das Integral der quadratischen Regelabweichung verwendet.

## PID-Regler

Der erste Ansatz zur Kontrolle des Würfels besteht darin einen PID-Regler zu entwerfen. Da die Regelstrecke bekannt und somit simulierbar ist, bietet es sich an Methoden des maschinellen Lernen zu verwenden um die Regelparameter zu bestimmen.

## Evolutionäre Algorithmen

Evolutionäre Algorithmen gehören zu den Methoden des maschinellen Lernens. Das Ziel besteht darin einen optimalen Lösungskandidaten  $k_i$  zu ermitteln in dem Prinzipien der Evolution, wie z.B. Mutation oder Selektion eingesetzt werden. Für einen evolutionären Algorithmus muss zuerst die Kodierung der Lösungskandidaten festgelegt werden. In diesem Anwendungsfall setzt sich ein Lösungskandidat aus jeweils einem Wert für die Verstärkung des proportionalen, des differenzierenden und des integrierenden Anteils zusammen.

$$k_i = \{K_P^{(i)}; K_D^{(i)}; K_I^{(i)}\}$$

Zu Beginn des Verfahrens wird eine Anfangspopulation  $P_0$  der Größe  $p$  von Lösungskandidaten erzeugt.

$$P_0 = \{k_1, k_2, \dots, k_p\}$$

Mit Hilfe einer Fitness-Funktion werden die einzelnen Lösungskandidaten bewertet. Hierbei soll die Fitness-Funktion so gestaltet werden, dass gute Kandidaten eine höhere Fitness erhalten. In diesem Fall wird das Integral der quadratischen Regelabweichung  $e_{k_i}$  verwendet.

$$f_{fit}(k_i) : k_i \rightarrow fitness_{k_i} \quad | \quad fitness_{k_i} \in \mathbb{R}$$

$$f_{fit}(k_i) = \left( \int_0^{t_{end}} e_{k_i}^2 dt \right)^{-1}$$

Der Algorithmus basiert darauf die aktuelle Population zu verändern und dadurch die nächste Generation zu erzeugen. Dies erfolgt durch die Manipulation und das Ersetzen von Kandidaten. Hierfür werden so genannten genetische Operatoren verwendet. Einerseits wird mit einer Selektionsmethode bestimmt welche Kandidaten aus der alten Population in die neue übernommen werden. Der Selektionsparameter  $r$  bestimmt den Anteil der zu ersetzenden Kandidaten.

$$f_{sel}(P_i) : P_i \rightarrow P_{i,selected} \quad | \quad \|P_{i,selected}\| = p \cdot r$$

In dieser Anwendung wird eine Selektionsmethode verwendet, die mit gewichteter Wahrscheinlichkeit Kandidaten auswählt. Zuerst wird die relative Fitness eines Kandidaten ermittelt.

$$fitness_{rel}(k_i) = \frac{f_{fit}(k_i)}{\sum_{j=1}^p f_{fit}(k_j)}$$

Die relative Fitness eines Kandidaten wird als Wahrscheinlichkeit interpretiert. Somit werden zufällige Kandidaten aus der alten Population ausgewählt. Allerdings erhöht eine hohe Fitness die Chance eines Kandidaten in die nächste Generation übernommen zu werden. Dieser Selektionsprozess wird so oft wiederholt bis die gewünschte Anzahl von Kandidaten in der Folgepopulation erreicht wurde.

Anschließend wird die selektierte Population mit Hilfe der Crossover-Funktion auf die ursprüngliche Größe  $p$  gebracht.

$$f_{cross}(P_{i,selected}) : P_{i,selected} \rightarrow P_{i,crossed} \quad | \quad \|P_{i,crossed}\| = p$$

Die Crossover-Funktion wählt zwei Kandidaten und erzeugt aus diesen einen neuen. Die Parameter des neuen Kandidaten werden bestimmt, in dem der jeweilige Wert des zufällig gewählten Elternkandidaten übernommen wird.

$$f_{cross}(k_i, k_j) : \{k_i, k_j\} \rightarrow k_{new} \quad | \quad k_i, k_j \in \mathbb{R}$$

Zuletzt wird die Population manipuliert. Hierfür werden die Parameter der Lösungskandidaten mit der Mutation-Funktion verändert. Die Mutationsrate  $m$  bestimmt hierbei die Wahrscheinlichkeit, dass ein Kandidat mutiert wird.

$$f_{mut}(P_{i,crossed}) : P_{i,crossed} \rightarrow P_{i,mutated}$$

$$P_{i,mutated} = P_{i+1}$$

Die Mutation-Funktion addiert zu den Parametern eines Kandidaten einen zufälligen Wert, der in einem vorgegebenen Bereich liegt.

$$f_{mut}(k_i) = \{K_P^{(i)} + M_P^{(i)}; K_D^{(i)} + M_D^{(i)}; K_I^{(i)} + M_I^{(i)}\} \quad | \quad M_{min} < M^{(i)} < M_{max}$$

Dieser Ablauf wird wiederholt bis entweder ein Kandidat die gewünschte Fitness erreicht hat oder eine vorgegebene Anzahl von Generationen durchlaufen worden ist.

## Parameterbestimmung des Algorithmus

Der Erfolg eines evolutionären Algorithmus ist stark von den gewählten Parametern abhängig. Deshalb müssen die Auswirkungen der einzelnen Werte zuerst analysiert werden. Daraufhin werden verschiedene Konfigurationen getestet und miteinander verglichen.

Parameter	Auswirkung
Populationsgröße $k$	Eine ausreichend große Population ist die Grundvoraussetzung für eine flächendeckende Durchforstung des Suchraumes
Selektionsparameter $r$	Bestimmt den Anteil der zu ersetzenden Population. Ein zu niedriger Wert führt zu einer lokalen Suche, ein dagegen zu großer Wert resultiert in einer reinen Zufallssuche.
Mutationsrate $m$	Bestimmt die Wahrscheinlichkeit, dass ein Kandidat mutiert wird. Ein zu großer Wert verhindert das finden von lokalen Optima, ein zu kleiner Wert führt zu einer Versteifung auf lokale Optima.
Mutationswert $M_{min} \Leftrightarrow M_{max}$	Bestimmt den maximalen Wert der zu einem Parameter addiert wird. Ein zu großer Wert führt zu einer reinen Zufallssuche. Ein zu kleiner Wert verhindert das Durchsuchen des gesamten Raumes.

Die Rechenzeit hängt linear von der Populationsgröße ab, bei der Simulation eines Modelles ist dies vernachlässigbar. Bei dem Reglerentwurf an einem reellen System kann dies allerdings zu einem großen zeitlichen Aufwand führen. Deswegen wird bei diesen versuchen die maximale Größe von 20 Lösungskandidaten nicht überschritten.