



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Pascal Stubel

**Untersuchung der Eignung von Reinforcement Learning
Algorithmen zur Regelung einer Lüftungsanlage im Vergleich zu
Standardreglern**

*Fakultät Technik und Informatik
Studiendepartment Informations-
und Elektrotechnik*

*Faculty of Engineering and Computer Science
Department of Information and Electrical Engi-
neering*

Pascal Stubel

**Untersuchung der Eignung von Reinforcement Learning
Algorithmen zur Regelung einer Lüftungsanlage im Vergleich zu
Standardreglern**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Elektro- und Informationstechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Michael Erhard
Zweitgutachter: Prof. Dr. Klaus Jünemann

Eingereicht am: 11.04.2022

Pascal Stubel

Thema der Arbeit

Untersuchung der Eignung von Reinforcement Learning Algorithmen zur Regelung einer Lüftungsanlage im Vergleich zu Standardreglern

Stichworte

Maschinelles Lernen, Reinforcement Learning, Soft-Acor-Critic Algorithmus, Gebäudeautomation

Kurzzusammenfassung

In dieser Arbeit wird untersucht, inwiefern sich Reinforcement Learning Algorithmen dazu eignen einen Regler zu trainieren, dessen Aufgabe es ist, eine Lüftungsanlage zu regeln. Der trainierte Regler wird mit Standardreglern verglichen, die über unterschiedliche Verfahren ausgelegt werden. Die Untersuchung der Eignung erfolgt anhand mehrere Aspekte. Es werden das Regelverhalten bei Stör- und Führungsgrößensprüngen sowie der Zeitaufwand und die Zuverlässigkeit der Trainingsergebnisse untersucht.

Pascal Stubel

Title of the paper

Examination of the suitability of reinforcement learning algorithms for controlling a ventilation system in comparison to standard controllers

Keywords

machine learning, reinforcement learning, soft-actor-critic algorithm, building automation

Abstract

This thesis examines the extent to which reinforcement learning algorithms are suitable for training a controller with the aim of controlling a ventilation system. The trained controller is compared with standard controllers that are designed using different Automation engineering methods. The suitability is investigated on the basis of several aspects. The control behaviour for setpoint and disturbance jumps, as well as the time required to design and train the controller and the reliability of the training results are examined.

Inhaltsverzeichnis

Tabellenverzeichnis	vi
Abbildungsverzeichnis	vii
1 Einleitung	1
2 Ausgangssituation	2
2.1 Versuchsaufbau	2
2.2 Software	4
2.3 Regelstrecke	4
2.3.1 Physikalisches Modell	5
2.3.2 Kontinuierliche Betrachtung	6
3 Reinforcement Learning	8
3.1 Terminologie	8
3.1.1 Trajektorie und Episode	9
3.1.2 Belohnung und Belohnungsfunktion	9
3.1.3 Strategie	11
3.1.4 Value-Funktionen	12
3.1.5 Entropie	12
3.2 Auswahl des Reinforcement Learning Algorithmus	13
3.3 Soft Actor-Critic-Algorithmus	16
4 Auslegung der Standardregler	18
4.1 Bestimmung des Arbeitspunkts	18
4.2 Dimensionierung nach Ziegler-Nichols	21
4.3 Reglersynthese mit Polkompensation	23
4.3.1 Auswahl einer geeigneten Dämpfung	25
4.3.2 Bestimmung der Verstärkung des Reglers	27
4.4 Dimensionierung über Tune-Funktion	28
5 Training des Soft Actor-Critic Agenten	29
5.1 Implementierung in Matlab/Simulink	29
5.1.1 Matlab	30

5.1.2	Simulink-Modell	31
5.1.3	Lernprozess	31
5.2	Training am mathematischen Modell	33
5.2.1	Kompensation der Totzeit	34
5.2.2	Zustandsvektor	41
5.2.3	Aktionsvektor	51
5.2.4	Belohnungsfunktion	52
5.2.5	Wahl der Hyperparameter	55
5.3	Training am Versuchsaufbau	56
5.3.1	Lernen am mathematischen Modell und Anlage	58
5.3.2	Lernen an der Anlage	62
6	Vergleich des trainierten Agenten mit den dimensionierten PI-Reglern	64
6.1	Messung der Führungssprungantworten	64
6.1.1	Reglerentwurf Ziegler-Nichols-Einstellregeln	64
6.1.2	Reglerentwurf mittels Polkompensation	65
6.1.3	Reglerentwurf mittels Tune-Funktion eines Industriereglers	67
6.1.4	Regelung mittels Soft-Actor-Critic Agent	68
6.1.5	Auswertung	69
6.2	Messung der Störsprungantworten	71
6.2.1	Reglerentwurf Ziegler-Nichols-Einstellregeln	71
6.2.2	Reglerentwurf mittels Polkompensation	72
6.2.3	Reglerentwurf mittels Tune-Funktion eines Industriereglers	73
6.2.4	Regelung mittels Soft Actor-Critic Agent	74
6.2.5	Auswertung	75
6.3	Vergleich der Stellgröße	77
6.4	Vergleich des Zeitaufwands	79
7	Fazit	81
A	Inhalt der DVD	83
	Literaturverzeichnis	84

Tabellenverzeichnis

3.1	Liste der Reinforcement Learning Algorithmen mit ihren möglichen Einsatzbereichen [9]	14
4.1	Reglereinstellwerte nach Ziegler-Nichols ([14] ,S.208)	22
5.1	Ergebnisse der untersuchten Agenten mit verschiedenen Zustandsvektoren . .	51
6.1	Kenngrößen der Führungsgrößensprungantworten der Regelkreise	71
6.2	Kenngrößen der Störgrößensprungantworten der Regelkreise	77

Abbildungsverzeichnis

2.1	Versuchsaufbau der Temperatur-Regelstrecke	3
3.1	Beziehung zwischen Agent und Umwelt	9
4.1	Streckenantwort auf einen 4 V Führungsgrößensprung	19
4.2	orange: Führungsgrößensprungantwort der Regelstrecke; blau: Führungsgrößensprungantwort des mathematischen Modells der Regelstrecke	21
4.3	Vergleich der genäherten und der tatsächlichen Übertragungsfunktion der Regelstrecke	24
4.4	e_{\max} in % als Funktion des Dämpfungsgrad in Bezug auf $h_{w,\infty} = 100\%$	26
5.1	Vergleich der Auswirkung der Totzeit-Kompensation auf die mittlere Belohnung	34
5.2	Regelverhalten ohne Totzeitgliedkompensation bei einem Sollwertsprung, rot: Führungsgröße, gelb: Regelgröße, blau: Stellgröße	36
5.3	Blockschaltbild des Smith-Prädiktors [14] S.291	37
5.4	Regelverhalten des mit Smith-Prädiktors trainierten Agenten bei einem Führungsgrößensprung, rot = Führungsgröße, gelb = Regelgröße, blau = Stellgröße	40
5.5	Regelverhalten des im Frequenzbereich dimensionierten Reglers bei einem Sollwertsprung, rot = Führungsgröße, gelb = Regelgröße, blau = Stellgröße	41
5.6	Verlauf der mittleren Belohnung über das Training mit dem Regelfehler als Zustand	42
5.7	Verlauf der Regelgröße (orange) und Führungsgröße (blau) bei einem Zustandsvektor $s_t = (e_t)$	43
5.8	Verlauf der Stellgröße (blau) bei einem Sollwertsprung bei der Regelgröße als Zustand	44
5.9	Verlauf der mittleren Belohnung über das Training bei Agenten mit Regelfehler und Führungsgröße als Zustand	46
5.10	Verlauf der Regelgröße(orange) auf einen Führungsgrößensprung(blau) bei Agent mit Regelfehler und Führungsgröße als Zustand	47
5.11	Verlauf der Stellgröße auf einen Führungsgrößensprung bei Agent mit Regelfehler und Führungsgröße als Zustand	48
5.12	Verlauf der mittleren Belohnung über das Training bei Agenten mit Regelfehler und Regelgröße als Zustand	49

5.13	Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau) bei Agent mit Regelfehler und Regelgröße als Zustand	50
5.14	Verlauf der Stellgröße auf einen Führungsgrößensprung bei Agent mit Regelfehler und Regelgröße als Zustand	50
5.15	Verlauf der Belohnungsfunktion (blau) und ihrer Bestandteile R_1 (rot) und R_2 (gelb) in Abhängigkeit vom Betrag des Regelfehlers	53
5.16	Verlauf der Regelgröße (gelb) bei einem Führungsgrößensprung (blau)	54
5.17	Verlauf der Stellgröße bei einem Führungsgrößensprung	55
5.18	Mittlere Belohnung während des Training bei unterschiedlichen Mini-Batch-Sizes	56
5.19	Mittlere Belohnung während des Training von sechs Agenten unter den gleichen Bedingungen	58
5.20	Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau)	59
5.21	Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau) am Versuchsaufbau, bei der Regelung des am mathematischen Modell trainierten Agenten	60
5.22	Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau) am Versuchsaufbau, bei der Regelung des weiter trainierten Agenten	61
5.23	Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau) am Versuchsaufbau, bei der Regelung des Versuchsaufbau trainierten Agenten	62
6.1	Sprungantwort (blau) des Regelkreises mit dem nach Ziegler-Nichols ausgelegten Regler auf einen Führungsgrößensprung (orange)	65
6.2	Sprungantwort (blau) des Regelkreises mit dem mittels Polkompensation ausgelegten Regler auf einen Führungsgrößensprung (orange)	66
6.3	Sprungantwort (blau) des Regelkreises mit dem mittels Tune-Funktion ausgelegten Regler auf einen Störgrößensprung (orange)	68
6.4	Sprungantwort (blau) des Regelkreises mit dem Agenten auf einen Störgrößensprung (orange)	69
6.5	Sprungantwort (blau) des Regelkreises mit dem nach Ziegler-Nichols ausgelegten Regler auf einen Störgrößensprung (orange)	72
6.6	Sprungantwort (blau) des Regelkreises mit dem mittels Polkompensation ausgelegten Regler auf einen Störgrößensprung (orange)	73
6.7	Sprungantwort (blau) des Regelkreises mit dem mittels Tune-Funktion ausgelegten Regler auf einen Störgrößensprung (orange)	74
6.8	Sprungantwort (blau) des Regelkreises mit dem Agenten auf einen Störgrößensprung (orange)	75
6.9	Verlauf der Stellgröße des Agenten auf einen Sollwertsprung	78
6.10	Verlauf der Stellgröße des Reglers mit Polkompensation auf einen Sollwertsprung	78

1 Einleitung

In einem modernen Gebäudekomplex sind eine Vielzahl von unterschiedlichen Lüftungsanlagen verbaut. Die unterschiedlichen Anforderungen an die Raumluft von Sanitäreinrichtungen, Küchen und Büroräumen erfordern, dass ein Gebäude mehrere Lüftungsanlagen besitzt. In großen Firmensitzen kann es dazu kommen, dass Lüftungsanlagen im zweistöckigen Bereich verbaut sind. Jede dieser Anlagen hat variierende Streckenparameter. Damit die Regler der einzelnen Anlagen passend an die jeweiligen Anforderungen dimensioniert werden können, müssen die Streckenparameter der einzelnen Lüftungsanlagen bestimmt werden. Das Bestimmen der Streckenparameter jeder einzelnen Lüftungsanlage würde eine erhebliche Menge an Zeit und Geld kosten, weil ein qualifizierter Techniker für die Aufgabe bereitgestellt werden müsste. Aufgrund der hohen Kosten wird normalerweise darauf verzichtet, die Regler anhand der Streckenparameter zu dimensionieren. Oft werden die Regelparameter aufgrund von Erfahrungswerten gewählt oder von Reglern ähnlicher Anlagen übernommen.

In den letzten Jahren gab es große Fortschritte im Bereich des Reinforcement Learning. Es gibt Beispiele von Robotern, die sich selbst über Reinforcement Learning Algorithmen das Laufen beigebracht haben. Es stellt sich nun die Frage, ob ein Computer sich selbst beibringen kann, eine Lüftungsanlage zu regeln. Das hätte den entscheidenden Vorteil, dass sich kein Techniker mit der Dimensionierung der Regler beschäftigen muss.

In dieser Arbeit wird untersucht, ob sich ein moderner Reinforcement Learning Algorithmus dazu eignet, einen Regler zu trainieren, der in der Lage ist, eine Lüftungsanlage zu regeln. Im Idealfall ist der trainierte Regler besser als ein Regler, der auf Grundlage der Streckenparameter für die Lüftungsanlage dimensioniert ist.

2 Ausgangssituation

Wie eben beschrieben, soll in dieser Arbeit untersucht werden, inwiefern sich der Einsatz von modernen Reinforcement Learning Algorithmen eignet, um Lüftungsanlagen zu regeln. Der trainierte Regler wird dann mit mehreren PI-Reglern verglichen, welche in der Industrie weit verbreitet sind. Die zu untersuchenden PI-Regler werden mithilfe von verschiedenen Verfahren dimensioniert. Für die Auslegung werden folgende Verfahren in Betracht gezogen:

- empirische Dimensionierung nach Ziegler-Nichols-Einstellregeln
- Reglersynthese mit Polkompensation
- Autotuning mit Industrieregler

Damit ein passender Reinforcement Learning Algorithmus ausgewählt werden kann und damit die Standardregler dimensioniert werden können, muss zuerst die zu regelnde Anlage analysiert werden.

2.1 Versuchsaufbau

Um die Lüftungsanlage in einer Laborumgebung simulieren zu können, wird als Modell für die Lüftungsanlage eine Temperatur-Regelstrecke des Regelungstechnik-Labors verwendet.

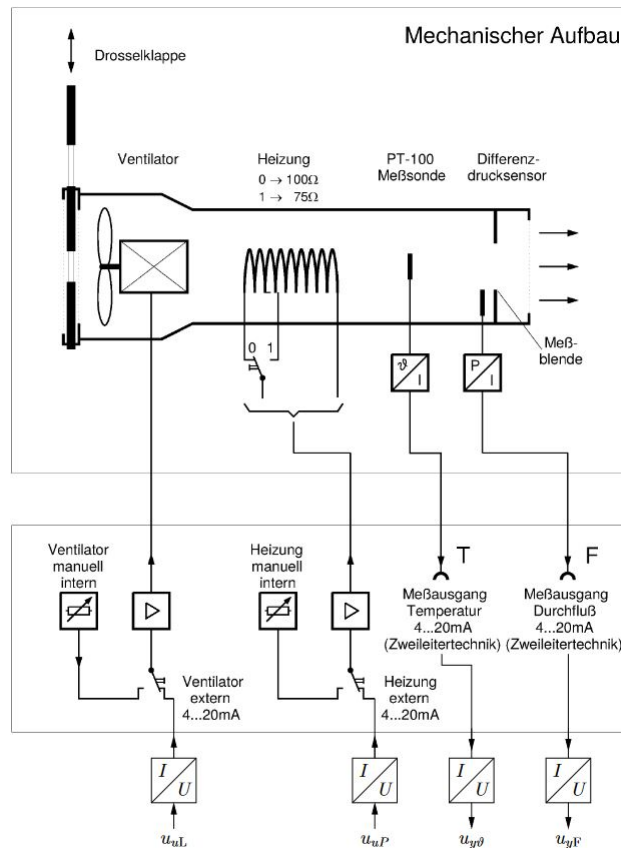


Abbildung 2.1: Versuchsaufbau der Temperatur-Regelstrecke

Quelle: [2](S.1)

Der Versuchsaufbau besteht aus einem Heizwiderstand, der zwischen 75Ω und 100Ω variiert werden kann. Für diesen Versuch wird ausschließlich der Heizwiderstand mit 100Ω eingesetzt. Der Heizwiderstand kann entweder intern direkt am Versuchsstand über einen Regler mit $0 W$ bis $100 W$ betrieben werden oder extern über ein $0 V - 10 V$ Stellsignal. Das Stellsignal wird dann über einen Strom-Spannungswandler in ein $4 mA - 20 mA$ Signal gewandelt. Am Ausgang der Regelstrecke befindet sich ein PT-100-Messwiderstand, über den die Austrittstemperatur gemessen wird. Die gemessene Temperatur wird als $4 mA - 20 mA$ Messsignal über einen Strom-Spannungswandler in ein $0 V - 10 V$ Messsignal umgewandelt, welches dann über Anschlussbuchsen abgegriffen werden kann. Der Ventilator wird mit einer konstanten

Drehzahl betrieben, weil in dieser Arbeit nur die Temperatur-Regelung über den Heizwiderstand untersucht werden soll. Die externen Kontakte für das Heizungs-Stellsignal und für das Temperatur-Messsignal werden an die Humusoft 624 Multifunktions I/O Karte angeschlossen, die im Labor-Rechner eingebaut ist. Das ermöglicht die Messung der Austrittstemperatur und die Regelung der Heizleistung über den Labor-Rechner. In den Versuchsaufbau ist ein Bürkert 1110 Industrieregler eingebaut, mit dem ein Regler über Autotuning dimensioniert werden kann.

2.2 Software

Die am weitesten verbreitete Form der Umsetzung von Reinforcement Learning Algorithmen geschieht über die Skriptsprache Python und den entsprechenden Paketen, die von Spining Up Open AI veröffentlicht werden. Der Nachteil ist, dass diese offiziell nur auf Linux oder MAC basierenden Betriebssystemen unterstützt werden. Die im Versuchsaufbau verwendete Humusoft 624 Multifunktions I/O Karte soll nicht ausgebaut werden und der entsprechende Computer hat als Betriebssystem Windows. Deswegen muss eine andere Möglichkeit der Implementierung gewählt werden. Matlab bietet mehrere Toolboxen an, über die Reinforcement Learning Algorithmen in Matlab und Simulink umgesetzt werden können. Als Software für die Auswertung des Messsignals und Ausgabe des Stellsignals wird Matlab in der Version 2021a verwendet, welche mit der eingesetzten Multifunktions I/O Karte von Humusoft kompatibel ist. Über Matlab-Simulink können die Ein- und Ausgänge der Karte in Echtzeit gelesen und geschrieben werden. Eine Liste der benötigten Matlab Toolboxen kann im Anhang gefunden werden.

2.3 Regelstrecke

Damit die PI-Regler passend ausgelegt werden können und ein Modell für das Training abseits der Anlage entwickelt werden kann, muss ein mathematisches Modell der Regelstrecke entworfen werden. Dafür wird das stationäre und dynamische Verhalten der physikalischen Strecke betrachtet.

2.3.1 Physikalisches Modell

Zuerst wird das stationäre Verhalten betrachtet. Befindet sich die Anlage im stationären Zustand, wird die zugeführte elektrische Leistung P_{el} komplett in thermische Leistung P_{th} umgewandelt. Soll ein Stoff um die Temperaturdifferenz ϑ erwärmt werden, muss ihm nach [2](S.2) die Energie

$$E_{th} = c \cdot m \cdot \vartheta = c \cdot \rho \cdot V \cdot \vartheta \quad (2.1)$$

zugeführt werden. Mit dem physikalischen Zusammenhang $P = \frac{dE}{dt}$ ([2], S.2) und der Bedingung, dass es bei der Wandlung von elektrischer in thermische Energie keine Verluste gibt, erhält man nach ([2], S.2) die benötigte thermische Leistung, um einen bestimmten Volumenstrom zu erwärmen:

$$P_{th} = \frac{dE_{th}}{dt} = c_L \cdot \rho_L \cdot \frac{dV}{dt} \vartheta = c_L \cdot \rho_L \cdot \frac{dV}{dt} \cdot \vartheta = c_L \cdot \rho_L \cdot \dot{V} \cdot \vartheta = P_{el} \quad (2.2)$$

Für die spezifische Wärmekapazität von Luft c_L und die Dichte von Luft ρ_L gilt ([2], S.2):

$$c_L = 1,01 \frac{W \cdot s}{g \cdot K}; \quad \rho = 1,293 \frac{kg}{m^3}$$

Der Volumenstrom $\dot{V} = A \cdot v$ ist definiert als das Produkt der durchströmten Fläche und der Strömungsgeschwindigkeit der Luft. Stellt man Gleichung 2.2 nach ϑ um und ersetzt \dot{V} gilt für die Temperaturdifferenz:

$$\vartheta = \frac{1}{c_L \cdot \rho_L \cdot A} \cdot \frac{1}{v} \cdot P_{th} \quad (2.3)$$

ϑ beschreibt hier die Temperaturdifferenz bezüglich der Raumtemperatur.

Bei der Betrachtung des dynamischen Verhaltens muss nach [2](S.3) berücksichtigt werden, dass die zugeführte elektrische Leistung nicht sofort als thermische Leistung abgegeben wird. Stattdessen muss das Heizelement selbst erwärmt werden. Die für die Erwärmung des Heizelements benötigte Energie ist nach [2](S.3):

$$-W_H = C_H \cdot \vartheta_H = \int (P_{el} - P_{th}) dt. \quad (2.4)$$

2 Ausgangssituation

Bildet man die Ableitung der Temperaturänderung gegenüber der Zeit t erhält man die Gleichung

$$C_H \cdot \frac{\vartheta_H}{dt} = P_{el} - P_{th} \quad (2.5)$$

Gleichung 2.5 kann mit Gleichung 2.3 nach P_{th} umgestellt entsprechend vereinfacht werden. Daraus folgt:

$$c_L \cdot \varrho_L \cdot A \cdot \vartheta_H + C_H \cdot \frac{d\vartheta_H}{dt} = P_{el} \quad (2.6)$$

2.3.2 Kontinuierliche Betrachtung

Betrachtet man die zugeführte elektrische Leistung P_{el} als zeitabhängige Eingangsgröße $P_{el}(t)$ und die Temperaturänderung ϑ_H als zeitabhängige Ausgangsgröße $\vartheta_H(t)$, erhält man die lineare Differenzialgleichung 1. Ordnung:

$$\vartheta_H(t) + \frac{C_H}{c_L \cdot \varrho_L \cdot A} \cdot \dot{\vartheta}_H(t) = \frac{1}{c_L \cdot \varrho_L \cdot A} \cdot P_{el}(t) \quad (2.7)$$

Das zugehörige Ergebnis der Laplace-Transformation in den Bildbereich ist:

$$\vartheta_H(s) + \frac{C_H}{c_L \cdot \varrho_L \cdot A} \cdot \dot{\vartheta}_H(s) = \frac{1}{c_L \cdot \varrho_L \cdot A} \cdot P_{el}(s) \quad (2.8)$$

Um Gleichung 2.8 zu vereinfachen, werden die Zeitkonstante

$$T_s = \frac{C_H}{c_L \cdot \varrho_L \cdot A} \quad (2.9)$$

und der Proportionalbeiwert

$$K_S = \frac{1}{c_L \cdot \varrho_L \cdot A} \quad (2.10)$$

eingeführt. Mit den Gleichungen 2.8, 2.9 und 2.10 folgt daraus die Übertragungsfunktion des Heizelements

$$G_H(s) = \frac{\vartheta_H(s)}{P_{el}(s)} = \frac{K_S}{1 + T_s \cdot s} \quad (2.11)$$

2 Ausgangssituation

welche aber noch nicht die komplette Regelstrecke beschreibt. Es wird die Abkühlung der transportierten Luft vom Heizelement zum Streckenausgang vernachlässigt, aber die Zeit, die für den Transport der erwärmten Luft über die Strecke l vom Heizelement zum Streckenausgang vergeht, muss in der Übertragungsfunktion berücksichtigt werden ([2], S. 3). Die Verzögerung wird als Totzeitglied im Bildbereich modelliert. Die daraus resultierende Übertragungsfunktion

$$G_T(s) = \frac{\vartheta(s)}{\vartheta_H(s)} = e^{-T_t \cdot s} \quad (2.12)$$

hat die Totzeit:

$$T_t = \frac{l}{v} \quad (2.13)$$

Die Zusammenführung der zwei Übertragungsfunktionen $G_H(s)$ und $G_T(s)$ zu einer Übertragungsfunktion, die das System-Modell für einen konstanten Volumenstrom beschreibt, ergibt:

$$G_S(s) = G_H(s) \cdot G_T(s) = \frac{\vartheta_H(s)}{P_{el}(s)} \cdot \frac{\vartheta(s)}{\vartheta_H(s)}$$
$$G_S(s) = \frac{\vartheta(s)}{P_{el}(s)} = \frac{Y(s)}{U(s)} = \frac{K_S \cdot e^{-T_t \cdot s}}{1 + T_S \cdot s} \quad (2.14)$$

Die Übertragungsfunktion der Strecke kann nun für das Matlab/Simulink-Modell eingesetzt werden.

3 Reinforcement Learning

Reinforcement Learning verbindet die wissenschaftlichen Bereiche des maschinellen Lernens, der künstlichen Intelligenz und der neuronalen Netzwerke ([13], S. xvii). Beim Reinforcement Learning lernt die Maschine, indem sie direkt mit dem zu regelnden System interagiert. Um Reinforcement Learning zu verstehen, müssen zuerst einige Begrifflichkeiten geklärt werden.

3.1 Terminologie

Der Prozess des Reinforcement Learning kann als Kreislauf aus zwei miteinander interagierenden Teilnehmern dargestellt werden. Die Rolle des Reglers wird beim Reinforcement Learning vom Agenten übernommen und das zu regelnde System wird als Umwelt bezeichnet ([1], S. 32-33). Der Agent nimmt über Aktionen a_t Einfluss auf die Umwelt. Die Menge an möglichen Aktionen wird als Aktionsraum \mathcal{A} bezeichnet ([1], S. 39). Im Gegenzug bekommt der Agent in jedem Zeitschritt die Information, in welchem Zustand s_t sich das System aktuell befindet. Die Menge an möglichen Zuständen wird als Zustandsraum \mathcal{S} bezeichnet ([1], S. 39). Der Agent bekommt außerdem eine Rückmeldung von der Umwelt, wie gut oder schlecht die Aktion für die Umwelt war. Die Rückmeldung erfolgt im nächsten Zeitschritt und wird als Belohnung r_{t+1} bezeichnet ([13], S. 48). Anhand der Zustands-Information entscheidet der Agent seine nächste Aktion. Aus diesem Zusammenspiel entsteht dann eine Reihe von Zuständen, Aktionen und Belohnungen ([13], S. 48)

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, s_3, a_3, r_4, \dots \quad (3.1)$$

In der Regel braucht der Agent nicht den kompletten Zustand der Umwelt, sondern nur den Teil, der für den Entscheidungsprozess des Agenten wichtig ist. Bekommt der Agent nur einen Ausschnitt des kompletten Zustands, wird das als Beobachtung (eng. observation)

bezeichnet ([1],S.32). Die Auswirkung der Zusammensetzung des an den Agenten übergebenen Teil-Zustands beziehungsweise der übergebenen Beobachtung wird in Abschnitt 5.2.2 näher untersucht.

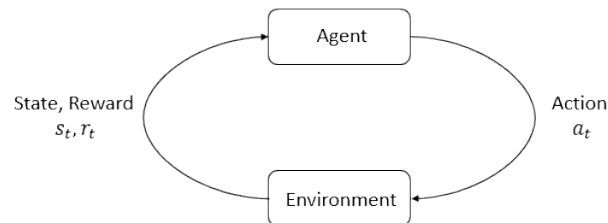


Abbildung 3.1: Beziehung zwischen Agent und Umwelt

Quelle: [1], S. 32

3.1.1 Trajektorie und Episode

Eine Reihe von aufeinander folgenden Zuständen und Aktionen wird als Trajektorie oder auch als Episode $\tau = (s_0, a_0, s_1, a_1, \dots, s_n, a_n)$ definiert([1], S. 35) und kann entweder eine endliche Zahl oder unendlich viele Zeitschritte haben. Die Bezeichnung Episode wird in dieser Arbeit verwendet, wenn die komplette Folge der Zustände und Aktionen eines Trainingszyklus betrachtet wird, also von Zeitschritt $t = 0$ bis zum Zeitschritt $t = T$, an dem die Abbruchbedingung des Trainingszyklus erfüllt ist. Die Trajektorie bezeichnet hier eine Reihe von Zuständen und Aktionen variabler Länge innerhalb einer Episode.

3.1.2 Belohnung und Belohnungsfunktion

Die Belohnung und Belohnungsfunktion sind für das Reinforcement Learning von essentieller Bedeutung, wie in diesem Abschnitt erklärt wird. Jedem Paar aus Zustand und Aktion s_t, a_t folgt die um einen Zeitschritt verzögerte Belohnung r_{t+1} ([13], S. 48). Daraus folgt, dass aus der Episode beziehungsweise Trajektorie $\tau = (s_0, a_0, s_1, a_1, \dots, s_n, a_n)$ die Reihe

$$r_1, r_2, \dots, r_{n+1} \tag{3.2}$$

von Belohnungen resultiert ([13], S. 48). Die Belohnung $r_{t+1} = R(s_t, a_t, s_{t+1})$ ([1], S. 36) signalisiert dem Agenten, wie gut oder schlecht seine letzte Aktion war ([13], S.28) und wird von der Belohnungsfunktion $R(s_t, a_t, s_{t+1})$ bestimmt. Die Entwicklung einer passenden Belohnungsfunktion ist für den Erfolg des Lernprozesses entscheidend. Über die Belohnung wird dem Agenten sein Ziel mitgeteilt ([13], S. 28). Daraus folgt, dass eine schlecht definierte Belohnungsfunktion dazu führen kann, dass der Agent ein Verhalten lernt, das ihn niemals zum eigentlichen Ziel führt. Die Belohnung ist bedeutend, denn das Ziel des Agenten ist in der Regel, die Summe der Belohnungen über einen Zeitraum beziehungsweise über eine Episode zu maximieren ([1], S. 36). Die Summe der Belohnungen über eine Episode mit $T \in [0, \infty]$ Zeitschritten wird als Return bezeichnet und kann nach [13](S. 55) wie folgt definiert werden:

$$G_t = \sum_{t=0}^T r_{t+1} = \sum_{t=0}^T R(s_t, a_t, s_{t+1}) \quad (3.3)$$

t legt das Zustands-Aktions-Paar (s_t, a_t) und damit die Trajektorie $\tau = (s_0, a_0, s_1, a_1, \dots, s_{T-t}, a_{T-t})$ mit $s_0 = s_t, a_0 = a_t$ innerhalb der Episode fest, für welche der Return bestimmt werden soll. Ist $t = 0$, dann wird die komplette Episode betrachtet. Daraus folgt, dass der Return aus Gleichung 3.3 auch als Funktion von τ formuliert werden kann, wie es [1](S.36) vorschlägt:

$$G(\tau) = \sum_{k=0}^{T-t} r_{k+1} = \sum_{k=0}^{T-t} R(s_k, a_k, s_{k+1}) \quad (3.4)$$

Episoden sind nicht immer endlich, wie das oft bei Regelprozessen der Fall ist. So auch bei der hier untersuchten Regelung einer Lüftungsanlage. Würde man in Gleichung 3.4 eine nicht endende Episode mit $T = \infty$ Zeitschritten betrachten, wäre es nicht möglich, den Return zu maximieren, da dieser möglicherweise nicht konvergiert. Damit sichergestellt ist, dass der Return auch bei nicht endlichen Episoden immer gegen ein Maximum konvergiert, wird ein Abzugsfaktor $\gamma \in [0, 1)$ eingeführt ([13], S. 55). Daraus ergibt sich für den Return bei $T = \infty$:

$$G(\tau) = \sum_{k=0}^{T-t} \gamma^k \cdot r_{k+1} = \sum_{t=0}^{\infty} \gamma^t \cdot R(s_k, a_k, s_{k+1}) \quad (3.5)$$

Der Abzugsfaktor γ bestimmt nach [13](S. 55), wie weit- oder kurzfristig der Agent in Bezug auf den Return ist. Je kleiner γ gewählt wird, desto kleiner wird der Anteil der Belohnungen am Return, die in Bezug auf den Zeitpunkt t weit in der Zukunft liegen. Andersherum gilt, je größer γ gewählt wird, desto länger haben die Belohnungen, die weit entfernt vom Ausgangszustand

liegen, einen signifikanten Anteil am Return. Bei dem Sonderfall, dass $\gamma = 1$ gewählt wird, würden alle Belohnungen gleich gewichtet werden, egal ob sie sehr nah oder weit weg vom Beobachtungszeitpunkt t liegen. $\gamma = 1$ kann aber nur unter der Bedingung $T \neq \infty$ gewählt werden, weil sonst der Fall auftreten kann, dass der Return nicht konvergiert ([13], S. 55).

3.1.3 Strategie

Die Strategie ist die Grundlage, auf der der Agent seine nächste Aktion basiert. Strategien werden in zwei Kategorien einsortiert, die davon abhängen, welche Umwelt betrachtet wird. Nach [1](S. 33-34) gelten die in diesem Abschnitt folgenden Definitionen und Zusammenhänge. Eine deterministische Strategie definiert mit μ

$$a_t = \mu(s_t) \tag{3.6}$$

wird der Zustand des Systems übergeben und gibt als Antwort die eine Aktion zurück, die als am besten gewertet wird. Die Strategie weist jedem Zustand die beste Aktion zu. Deterministische Strategien werden in Systemen eingesetzt, wenn sicher ist, dass der Zustand des Systems $s_{t+1} = f(s_t, a_t)$ ist. Ist der Zustand $s_{t+1} \sim P(\cdot | s_t, a_t)$, der auf die Aktion a_t aus dem Zustand s_t folgt, nicht sicher, also stochastisch, wird auch eine stochastische Strategie

$$a_t \sim \pi(\cdot | s_t) \tag{3.7}$$

benötigt, um die nächste Aktion zu bestimmen. Die stochastische Strategie gibt auf einen Zustand eine Wahrscheinlichkeitsverteilung über die möglichen Aktionen zurück. Ob deterministisch oder stochastisch, das Ziel des Agenten ist es, die Strategie zu lernen, bei der der erwartete Return maximal wird. Der erwartete Return, wenn die aktuelle Strategie angewendet wird, ist wie folgt definiert:

$$J(\pi) = \mathbb{E}_\pi[G(\tau)] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k \cdot R_k(s_k, a_k, s_{k+1}) \right] \tag{3.8}$$

Daraus folgt, dass wenn der erwartete Return $J(\pi)$ einer Strategie maximal ist, die Strategie optimal ist. Die optimale Strategie kann also wie folgt definiert werden:

$$\pi^* = \operatorname{argmax}_\pi J(\pi) \tag{3.9}$$

3.1.4 Value-Funktionen

Die zwei Value-Funktionen sind sehr weit verbreitet und werden in fast allen Reinforcement Learning Algorithmen auf irgendeine Art und Weise geschätzt ([13], S. 58). Die Zustand-Value-Funktion gibt einen Rückschluss darauf, ob der Zustand, in dem sich die Umwelt gerade befindet, langfristig gesehen gut oder schlecht für den Agenten ist. Je höher der erwartete Return eines Zustands ist, desto besser ist der Zustand für den Agenten und infolgedessen steigt der Value des Zustands. Der Value eines Zustands ist der erwartete Return unter der Bedingung, dass man in einem Zustand s_t der Episode startet und von da aus über die Trajektorie τ der aktuellen Strategie π folgt [1](S.37). Der Value eines Zustands ist laut [1](S.37) wie folgt definiert:

$$v_{\pi}(s_t) = \mathbb{E}_{\tau \sim \pi}[G(\tau) | s_0 = s_t] \quad (3.10)$$

Die zweite Value-Funktion bewertet, wie gut es ist, eine bestimmte Aktion a_t in Zustand s_t auszuführen und wird als Zustands-Aktions-Value-Funktion $q_{\pi}(s_t, a_t)$ bezeichnet. Eine Aktion $a_{t,1}$ in s_t ausgeführt, ist besser als die Aktion $a_{t,2}$ in demselben Zustand ausgeführt, wenn der erwartete Return von diesem Zustands-Aktions-Paar größer ist als der erwartete Return des anderen Zustands-Aktions-Paars. Die Zustands-Aktions-Value-Funktion nach [1](S. 37)

$$q_{\pi}(s_t, a_t) = \mathbb{E}_{\tau \sim \pi}[G(\tau) | s_0 = s_t, a_0 = a_t] \quad (3.11)$$

ist definiert als der erwartete Return, wenn man in Zustand s_t eine Aktion $a_t \sim \mathcal{A}$ ausführt und dann über die Trajektorie τ der aktuellen Strategie π folgt. Die Zustands-Aktions-value-Funktion wird in der Regel als Q-Value-Funktion bezeichnet ([1](S. 37))

3.1.5 Entropie

Die Entropie ist ein Konzept, das bei dem im nächsten Abschnitt behandelten Soft-Actor-Critic-Algorithmus eine hohe Bedeutung hat. Die Entropie beschreibt, wie zufällig die Wahrscheinlichkeitsverteilung einer Variable X in einem stochastischen Prozess ist. Nach [4](S.14-17) gelten die folgenden Zusammenhänge und Definitionen dieses Kapitels. X ist eine zufällige Variable mit der Wahrscheinlichkeitsfunktion $p_X(x) = Pr(X = x)$, $x \in U$, wobei U eine diskrete Zahlenmenge ist, die x enthält. Die Entropie der zufälligen Variable X ist definiert als:

$$H(X) \doteq - \sum_{x \in U} p_X(x) \cdot \log_2 p_X(x) \quad (3.12)$$

Es wird ein Logarithmus zur Basis 2 verwendet und das Ergebnis in Bits angegeben, weil die Definition aus der Informationstechniktheorie kommt. Ist $X \sim p(x)$, dann gilt für den erwarteten Wert einer zufälligen Variable $g(X)$:

$$\mathbb{E}_p[g(X)] = \sum_{x \in U} g(x) \cdot p(x) \quad (3.13)$$

Bestimmt man in Gleichung 3.13 anstelle des erwarteten Werts von $g(X)$ den erwarteten Wert für $\log_2 p(X)$, erhält man die Gleichung

$$\mathbb{E}_p[\log_2 p_X(X)] = \sum_{x \in U} \log_2 p_X(x) \cdot p_X(x) \quad (3.14)$$

mit der man die Gleichung 3.12 wie folgt umformen kann:

$$H(X) \doteq -\mathbb{E}_p[\log_2 p_X(X)] \quad (3.15)$$

3.2 Auswahl des Reinforcement Learning Algorithmus

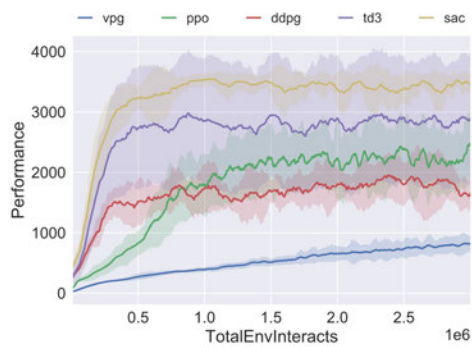
Für das Training des Agenten muss ein geeigneter Reinforcement Learning Algorithmus gefunden werden. Es gibt mittlerweile eine Vielzahl von Reinforcement Learning Algorithmen, die alle Vor- und Nachteile haben. Entscheidend ist, nicht jeder Reinforcement Learning Algorithmus lässt sich in jeder Situation einsetzen. Für die Auswahl des geeigneten Reinforcement Learning Algorithmus müssen die Eigenschaften des betrachteten Systems sowie die Anforderungen an den Regler in Betracht gezogen werden.

Tabelle 3.1: Liste der Reinforcement Learning Algorithmen mit ihren möglichen Einsatzbereichen [9]

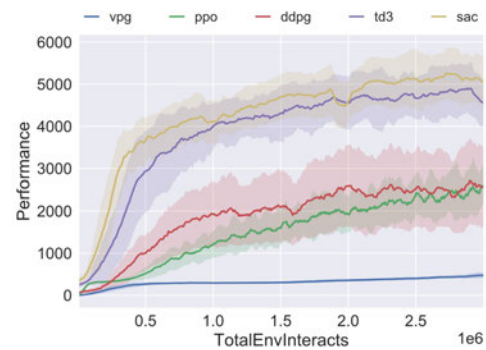
Algorithmus	Diskreter Aktionsraum und Zustandsraum	Diskreter Aktionsraum und kontinuierlicher Zustandsraum	Kontinuierlicher Aktionsraum und Zustandsraum
SARSA	Ja	Nein	Nein
Q-Learning	Ja	Nein	Nein
DQN	Ja	Ja	Nein
PPO	Ja	Ja	Ja
TRPO	Ja	Ja	Ja
DDPG	Nein	Nein	Ja
TD3	Nein	Nein	Ja
SAC	Nein	Nein	Ja

Tabelle 3.1 welche Algorithmen bei welcher Variation von Zustands- und Aktionsraum gewählt werden können. Der Zustandsraum hängt in diesem Versuch davon ab, welche Werte das Messsignal annehmen kann. Das analoge Messsignal kann theoretisch alle Werte zwischen $0\text{ V} - 10\text{ V}$ annehmen. Daraus folgt, dass der Zustandsraum des Algorithmus kontinuierlich sein soll. Das gleiche gilt für den Aktionsraum, der in diesem Versuch vom Stellsignal abhängt und auch kontinuierlich ist. Das Stellsignal muss kontinuierlich sein, weil gefordert ist, dass der Regelfehler minimiert werden soll, also idealerweise 0 annimmt. Das wäre nicht möglich, wenn das Stellsignal nur in festen Schritten geändert werden könnte. Betrachtet man Tabelle 3.1 und berücksichtigt die Anforderungen an den Zustands- und Aktionsraum, können nur drei Algorithmen für die Regelung in Betracht gezogen werden. Es muss nun zwischen Deep-Deterministic-Policy Gradient (DDPG), Twin-Delayed-Deep-Deterministic-Policy Gradient (TD3) - der, wie der Name schon sagt, eine Weiterentwicklung von DDPG ist - und Soft Actor-Critic (SAC) entschieden werden.

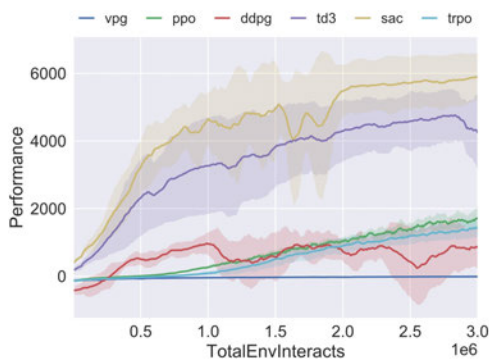
3 Reinforcement Learning



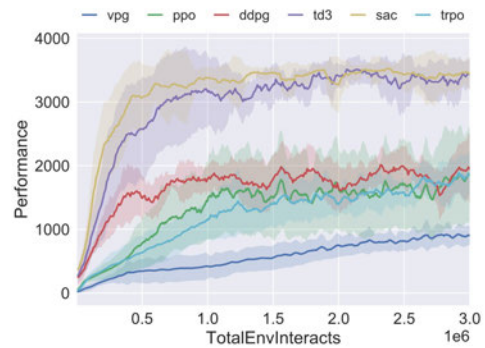
(a) Hopper-v3 Benchmark mit PyTorch implementierung.



(b) Walker2d-v3 Benchmark mit PyTorch implementierung.



(c) Ant-v3 Benchmark mit Tensorflow implementierung.



(d) Hopper-v3 Benchmark mit Tensorflow implementierung.

Quelle: <https://spinningup.openai.com/en/latest/spinningup/bench.html>

Die von OpenAi Spinning Up durchgeführten Benchmarks der am weitesten verbreiteten Algorithmen zeigen, dass Soft Actor-Critic in allen Benchmarks die besten Ergebnisse erzielt hat, wobei TD3 im Vergleich zu DDPG auch besser abschneidet. Für diesen Versuchsaufbau wird die Regelung mit dem Soft Actor-Critic-Algorithmus realisiert, weil die Performance durchgehend besser ist. Außerdem hat TD3 laut [1] das gleiche Problem wie sein Vorgänger DDPG. Beide reagieren sehr sensibel auf Änderungen der Hyperparameter. Soft Actor-Critic hat dieses Problem nicht.

3.3 Soft Actor-Critic-Algorithmus

Soft Actor-Critic(SAC) ist ein Reinforcement Learning Algorithmus, der zuerst in [5] vorgestellt wurde.

Das in Abschnitt 3.1 beschriebene Ziel der Maximierung des erwarteten Returns $J(\pi) = \mathbb{E}_{\tau \sim \pi}[G(\tau)]$ wird von [5](S. 3) bei diesem Algorithmus um die Maximierung der erwarteten Entropie erweitert. Dazu wird in jedem Zeitschritt auf die Belohnung die Entropie der aktuellen Strategie für den aktuellen Zustand s_t addiert. Um die Gewichtung der Entropie im Bezug zur Belohnung ändern zu können, wird von [5](S. 3) der Temperaturparameter $\alpha \in [0, \infty)$ eingeführt. Daraus resultiert nach [5](S. 3) ein neuer Erwartungswert

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \cdot \left(R(s_t, a_t, s_{t+1}) + \alpha \cdot H(\pi(\cdot|s_t)) \right) \right] \quad (3.16)$$

der den Return, die Entropie sowie den bereits bekannten Abzugsfaktor γ , der das Konvergieren bei Prozessen mit unendlichen Zeitschritten garantiert, enthält. Über den Temperaturparameter kann angepasst werden, wie stark der Entropie-Term gewichtet werden soll. Ein hoher Temperaturwert führt zu einer hohen Gewichtung der Entropie und erhöht damit die Stochastizität der zu lernenden optimalen Strategie π^* [5](S. 3). Das ursprüngliche Ziel der alleinigen Maximierung des erwarteten Returns kann für $\alpha = 0$ wiederhergestellt werden. Nach [5] (S. 3) hat die Einbindung der Entropie in den erwarteten Return mehrere Vorteile. Zum einen hat die Entropie den Effekt, dass die Strategie im Lernprozess mehr erkundet. Also, dass die Strategie aus ihrer Sicht weniger vorteilhafte Aktionen auswählt, mit dem Hintergrund, dass dadurch neue Trajektorien mit einem höheren Return gefunden werden können. Gleichzeitig werden Aktionen und daraus resultierende Episoden, die definitiv als schlecht gewertet werden, schneller verworfen. Der zweite Vorteil ist, durch das erhöhte Maß an Erkundung werden eher mehrere optimale Strategien beziehungsweise Verhaltensweisen gefunden. Nach [5](S. 3) ist die Folge, dass die Lerngeschwindigkeit gegenüber ähnlich modernen Algorithmen besser ist. Mit dem erweiterten erwarteten Return können nun nach [1](S.120-121) die Value-Funktionen aus Gleichung 3.10 und 3.11 entsprechend um die Entropie erweitert werden. Die Value-Funktion $v_{\pi}(s_t)$ enthält nun den Entropiewert bei jedem Zeitschritt. Es gilt nun:

$$v_{\pi}(s_t) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t, s_{t+1}) + \alpha \cdot H(\pi(\cdot|s_t)) \right) \middle| s_0 = s_t \right] \quad (3.17)$$

3 Reinforcement Learning

Die erweiterte Q-Value-Funktion - beziehungsweise Aktions-Value-Funktion - enthält nun den Entropiewert für jeden Zeitschritt außer dem ersten. Es gilt:

$$q_{\pi}(s_t, a_t) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) + \alpha \sum_{t=0}^{\infty} \gamma^t \cdot H(\pi(\cdot | s_t)) \middle| s_0 = s_t, a_0 = a_t \right] \quad (3.18)$$

Die beiden Value-Funktionen sind laut [1](S.121) wie folgt voneinander abhängig:

$$v_{\pi}(s_t) = \mathbb{E}_{a_t \sim \pi} [Q(s_t, a_t)] + \alpha \cdot H(\pi(\cdot | s_t)) \quad (3.19)$$

4 Auslegung der Standardregler

Wie bereits in Abschnitt 2 beschrieben, werden die Regler, die mit dem trainierten Agenten verglichen werden sollen, mithilfe mehrerer Verfahren dimensioniert. In der Regel werden an den zu dimensionierenden Regler mehrere Anforderungen gestellt. Laut [14](S.185) gehören zu den klassischen Anforderungen:

1. Der Regelkreis soll stabil sein
2. Der Einfluss eventuell vorhandener Störgrößen auf die Regelgröße soll minimiert werden
3. Der Regelfehler soll minimiert werden, idealerweise zu null

4.1 Bestimmung des Arbeitspunkts

Um den Arbeitspunkt zu bestimmen, wird der Ausgang der Regelstrecke für einen Stellgrößen-sprung zum Zeitpunkt $t = 9,8 \text{ s}$ von 3 V auf 7 V aufgezeichnet .

4 Auslegung der Standardregler

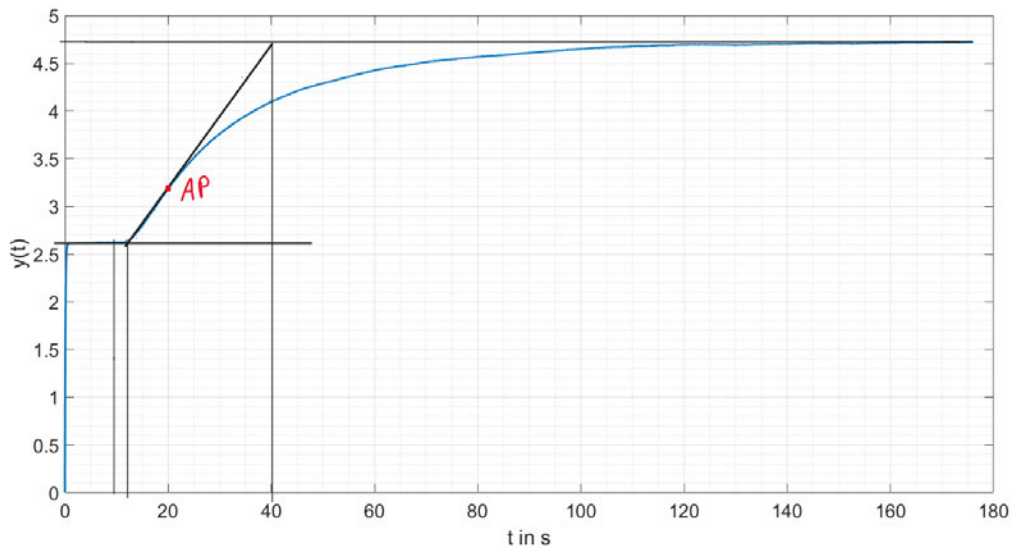


Abbildung 4.1: Streckenantwort auf einen 4 V Führungsgrößensprung

Abbildung 4.1 zeigt die Sprungantwort der Regelstrecke. Aus Abbildung 4.1 können nun die Streckenparameter graphisch bestimmt werden. Der Arbeitspunkt liegt im linearen Verlauf der Übertragungsfunktion bei ungefähr $y_{AP} = 3,25 \text{ V}$. Graphisch bestimmt ergeben sich nach [14](S. 207) die folgenden Streckenparameter: Die Verstärkung beträgt:

$$K_s = \frac{4,72 \text{ V} - 2,62 \text{ V}}{7 \text{ V} - 3 \text{ V}} = 0,525 \quad (4.1)$$

Die Verzugszeit T_u entspricht der Totzeit T_t und beträgt:

$$T_u = 2,3 \text{ s} = T_t \quad (4.2)$$

Die Ausgleichzeit T_g entspricht der Zeitkonstante T_s und beträgt:

$$T_g = 28 \text{ s} = T_s \quad (4.3)$$

Der Nachteil der graphischen Bestimmung ist, dass es durch Fehler beim Ablesen zu größeren Abweichungen der Streckenparameter kommen kann. Hat man eine Übertragungsfunktion für die Regelstrecke, kann man die Sprungantwort von Matlab analysieren lassen. Dafür gibt es

4 Auslegung der Standardregler

das Curve-Fitting-Tool. Dem Curve-Fitting-Tool übergibt man die Funktion der Strecke, die wie folgt bestimmt wird. Gleichung 2.14 wird nach $Y(s)$ umgestellt:

$$Y(s) = \frac{Y(s)}{U(s)} = G_S(s) \cdot U(s) \quad (4.4)$$

mit dem Sprung $U(s) = \frac{a}{s}$ der Höhe a folgt dann

$$Y(s) = \frac{K_S \cdot e^{-T_t \cdot s}}{1 + T_S \cdot s} \cdot \frac{a}{s} = \frac{a \cdot \frac{K_S}{T_S} \cdot e^{-T_t \cdot s}}{\left(\frac{1}{T_S} + \cdot s\right) \cdot s} \quad (4.5)$$

Mit der inversen Laplace Transformation ([15], S.466) für :

$$\frac{1}{s(\alpha + s)} \circ \bullet \frac{1}{\alpha} \cdot (1 - e^{-\alpha t}) \quad (4.6)$$

und dem Verschiebungssatz ([15], S. 465) kann die Antwort der Strecke $y(t)$ im Zeitbereich bestimmt werden. Es folgt:

$$y(t) = \mathcal{L}^{-1}\{Y(s)\} = a \cdot K_S \cdot \left(1 - e^{-\frac{(t-T_t)}{T_S}}\right) \quad (4.7)$$

Damit das Curve-Fitting-Tool die Sprungantwort analysieren kann, wird die Gleichung

$$y(t) = Y_0 + a \cdot K_S \cdot \left(1 - e^{-\frac{(t-T_t-T_0)}{T_S}}\right) \cdot \sigma(t - T_t - T_0) \quad (4.8)$$

implementiert. Vor der Analyse muss der Zeitpunkt $T_0 = 9,8 \text{ s}$, an dem der Sprung auftritt, sowie die Höhe $a = 4$ des Sprungs angegeben werden. Die Analyse der Sprungantwort aus Abbildung 4.1 ergibt die folgenden Streckenparameter:

$$K_S \approx 0.528 \quad (4.9)$$

$$T_S = 23,06 \quad (4.10)$$

$$T_t \approx 2,4 \quad (4.11)$$

Abbildung 4.2 zeigt die Sprungantwort der tatsächlichen Regelstrecke (orange) und des mathematischen Modells (blau) auf einen Sollwertsprung von 3 V auf 7 V bei $t = 9,8$ s. Der Verlauf der Sprungantwort des Versuchsaufbaus (orange) deutet darauf hin, dass es sich bei der tatsächlichen Übertragungsfunktion der Regelstrecke um ein PT_2 -Glied mit Totzeit handelt. Es ist erkennbar, dass die in Abschnitt 2.3 entwickelte Übertragungsfunktion der Regelstrecke eine gute Näherung der Übertragungsfunktion des Versuchsaufbaus ist. Das bestätigt die in [14](S. 206-207) getroffene Annahme, über aperiodische Prozesse, deren Verhalten durch ein PT_n -Glied beschrieben werden kann. Demnach können diese ausreichend gut durch ein PT_1 -Glied genähert werden.

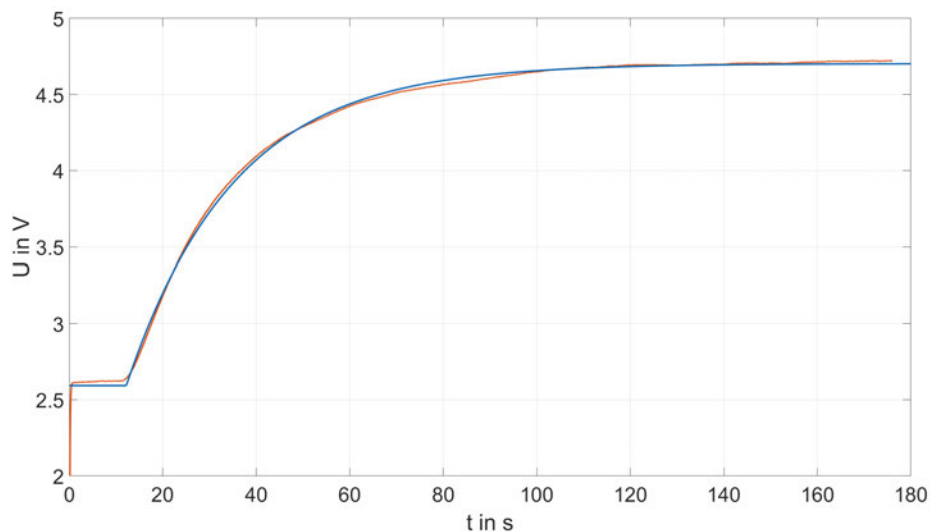


Abbildung 4.2: orange: Führungsgrößensprungantwort der Regelstrecke; blau: Führungsgrößensprungantwort des mathematischen Modells der Regelstrecke

4.2 Dimensionierung nach Ziegler-Nichols

Das erste und einfachste der hier untersuchten Verfahren für die Auslegung eines Reglers sind die Einstellregeln nach Ziegler-Nichols ([14], S. 207-208). Es werden zwei Methoden zur Bestimmung der Einstellwerte vorgestellt. In dieser Arbeit wird die Methode der Übergangsfunktion verwendet. Bei dieser Methode werden die Streckenparameter aus der Sprungantwort

4 Auslegung der Standardregler

der Strecke bestimmt ([14], S. 207). Der Vorteil in diesem Fall ist, dass die Streckenparameter bereits in Abschnitt 4.1 ermittelt wurden. Mit den Streckenparametern

$$K_s \approx 0,528 \quad (4.12)$$

$$T_s = 23,06 = T_a \quad (4.13)$$

$$T_t \approx 2,4 = T_u \quad (4.14)$$

und der Tabelle 4.1 wird der PI-Regler dimensioniert. Man erhält die Verstärkung

$$K_R = \frac{0,9 T_a}{K_s T_u} = \frac{0,9}{0,528} \frac{23,06}{2,4} = 16,63 \quad (4.15)$$

und für die Zeitkonstante

$$T_I = T_n = 3,33 \cdot T_u = 7,2 \text{ s} \quad (4.16)$$

Tabelle 4.1: Reglereinstellwerte nach Ziegler-Nichols ([14] ,S.208)

	Reglertypen	Reglereinstellwerte		
		K_R	T_I	T_D
Methode 1	P	$0,5 \cdot K_{R,\text{krit}}$	-	-
	PI	$0,45 \cdot K_{R,\text{krit}}$	$0,85 \cdot T_{\text{krit}}$	
	PID	$0,6 \cdot K_{R,\text{krit}}$	$0,5 \cdot T_{\text{krit}}$	$0,12 \cdot T_{\text{krit}}$
Methode 2	P	$\frac{1}{K_s} \frac{T_a}{T_u}$	-	-
	PI	$\frac{0,9 T_a}{K_s T_u}$	$3,33 \cdot T_u$	-
	PID	$\frac{1,2 T_a}{K_s T_u}$	$2 \cdot T_u$	$0,5 \cdot T_u$

4.3 Reglersynthese mit Polkompensation

Um die Dimensionierung des PI-Reglers für die in Abschnitt 2.14 bestimmte Übertragungsfunktion $G_s(s)$ der Strecke zu vereinfachen, wird die in [14](S. 206) vorgeschlagene Approximation eines PT_n -Glied als $PT_1 - T_t$ -Glied umgekehrt angewendet. Für die Approximation des Totzeitglieds wird die Potenzreihe

$$e^{-x} = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad (4.17)$$

der e-Funktion angewendet. Mit Gleichung 4.17 kann das Totzeitglied wie folgt genähert werden:

$$G_T(s) = e^{-T_t \cdot s} = \left(e^{T_t \cdot s} \right)^{-1} \quad (4.18)$$

$$G_T(s) = \left(\sum_{n=0}^{\infty} \frac{(T_t \cdot s)^n}{n!} \right)^{-1} = \left(1 + T_t \cdot s + \frac{(T_t \cdot s)^2}{2!} + \dots \right)^{-1} \quad (4.19)$$

Die Potenzreihe wird nach dem zweiten Element abgebrochen. Daraus folgt für das approximierte Totzeitglied

$$G_T(s) = e^{-T_t \cdot s} \approx \frac{1}{1 + T_t \cdot s} \quad (4.20)$$

Ersetzt man das Totzeitglied aus Gleichung 2.14 mit Gleichung 4.20, ergibt sich folgende neue Übertragungsfunktion der Strecke:

$$G_s(s) = \frac{K_s}{(1 + T_s \cdot s)(1 + T_t \cdot s)} \quad (4.21)$$

Um zu überprüfen, ob die Näherung gut ist, werden die Sprungantworten der beiden Übertragungsfunktionen aus den Gleichungen 2.14 und 4.21 in Matlab mit den in diesem Kapitel bestimmten Parametern für $K_s = 0,553$, $T_t = 2,4$ und $T_s = 23,06$ simuliert.

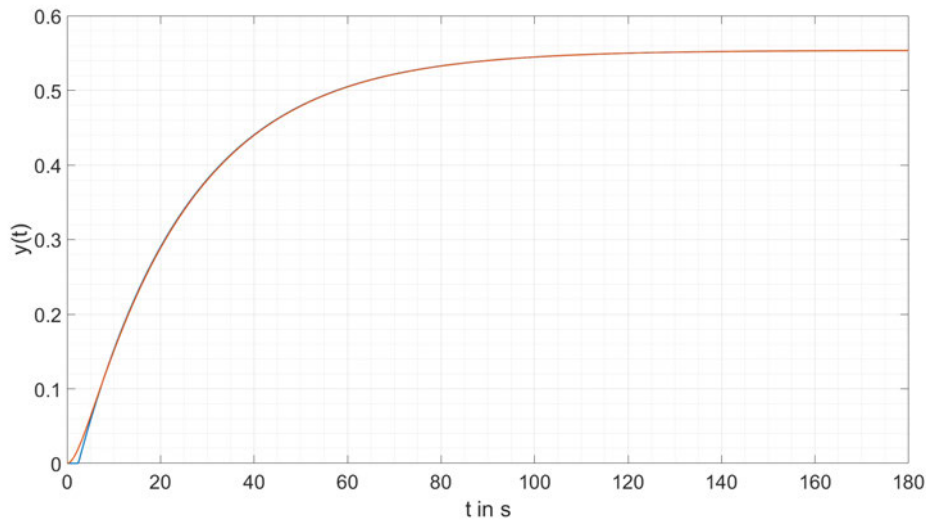


Abbildung 4.3: Vergleich der genäherten und der tatsächlichen Übertragungsfunktion der Regelstrecke

Abbildung 4.3 zeigt, dass bereits nach 7 Sekunden der Verlauf der beiden Übertragungsfunktionen nahezu identisch ist. Mit Gleichung 4.20 und der Übertragungsfunktion

$$G_R(s) = K_p \cdot \left(1 + \frac{1}{T_n \cdot s}\right) = K_p \cdot \frac{T_n \cdot s + 1}{T_n \cdot s} \quad (4.22)$$

des PI-Reglers wird nun die Übertragungsfunktion

$$G_0(s) = G_R(s) \cdot G_s(s) = K_p \cdot \frac{T_n \cdot s + 1}{T_n \cdot s} \cdot \frac{K_s}{(1 + T_s \cdot s)(1 + T_t \cdot s)} \quad (4.23)$$

des offenen Regelkreis bestimmt. Es wird nach [14](S. 241) der dominierende Pol durch eine Nullstelle im Regler kompensiert. Um den dominierenden Pol bei $s = -\frac{1}{T_s}$ der Regelstrecke zu kompensieren, wird $T_n = T_s$ gesetzt. Daraus folgt für den offenen Regelkreis:

$$G_0(s) = K_p \cdot \frac{1}{T_s \cdot s} \cdot \frac{K_s}{(1 + T_t \cdot s)} \quad (4.24)$$

Mit Gleichung 4.24 ergibt sich für den geschlossenen Regelkreis:

$$G_w(s) = \frac{G_0(s)}{1 + G_0(s)} = \frac{K_p \cdot K_s}{T_s \cdot s \cdot (1 + T_t \cdot s) + K_p \cdot K_s} \quad (4.25)$$

4.3.1 Auswahl einer geeigneten Dämpfung

[14](S. 188) führt mehrere Kenngrößen ein, die den Verlauf der Regelgröße $y(t)$ für eine sprungförmige Erregung der Führungsgröße charakterisieren. Zu den genannten Kenngrößen gehören:

- Der Betrag der maximalen Regelabweichung nach erstmaligem Erreichen der Führungsgröße wird als maximale Überschwingweite e_{max} bezeichnet.
- Der Zeitpunkt, an dem die maximale Überschwingweite auftritt, wird als t_{max} deklariert.
- $T_{a,50}$ bezeichnet die resultierende Anstiegszeit, wenn man die Tangente an $h_w(t)$ genau dann anlegt, wenn 50% der des Sollwerts erreicht sind.
- Die Verzugszeit T_u ist der Schnittpunkt von T_a mit der t-Achse.
- Der Zeitpunkt, ab dem die Regelabweichung $e(t)$ einen definiert Grenzwert unterschreitet, zum Beispiel 5% des Sollwerts, wird als Ausregelzeit t_ϵ deklariert.
- Die Anregelzeit t_{an} ist der Zeitpunkt, ab dem die Regelgröße erstmalig den Sollwert erreicht
- e_∞ kennzeichnet den bleibenden Regelfehler.

Nach [14](S. 188) können diese Kenngrößen in drei Kategorien einsortiert werden. e_{max} und t_ϵ bestimmen die Dämpfung des Regelverhaltens. Die Schnelligkeit des Regelverhaltens wird von $T_{a,50}$, t_{max} , t_{an} bestimmt. Das stationäre Verhalten wird vom bleibenden Regelfehler e_∞ gekennzeichnet. Bei der Auslegung des Reglers kann man sich auf die Einstellung von e_{max} , t_ϵ und $t_{a,50}$ limitieren. Es muss zwischen den drei Kenngrößen ein Kompromiss gefunden werden, weil sich die Kenngrößen gegenseitig beeinflussen. [14](S. 210-214) zufolge können die eben genannten Kenngrößen jeweils als Funktion der Dämpfung bestimmt werden. Im Folgenden wird die Dämpfung des Reglers von der maximalen Überschwingweite abhängig gemacht. [14](S. 211-212) zufolge gilt für ein PT_2 -Glied mit der Übertragungsfunktion

$$G_w(s) = \frac{\omega_0^2}{s^2 + 2 \cdot D \cdot \omega_0 \cdot s + \omega_0^2} = \frac{1}{\frac{1}{\omega_0^2} \cdot s^2 + \frac{2 \cdot D}{\omega_0} \cdot s + 1} \quad (4.26)$$

4 Auslegung der Standardregler

und der dazugehörigen Übergangsfunktion

$$h_w(t) = \left\{ 1 - e^{-D\omega_0 t} \left[\cos \sqrt{1 - D^2} \omega_0 t + \frac{D}{\sqrt{1 - D^2}} \sin \sqrt{1 - D^2} \omega_0 t \right] \right\} \cdot \sigma(t) \quad (4.27)$$

der folgenden Zusammenhang. Die maximale Überschwingweite e_{max} kann als Funktion von D ausgedrückt werden:

$$e_{max} = f_1(D) = e^{-\frac{D\pi}{\sqrt{1-D^2}}} \quad (4.28)$$

Daraus folgt die in Abbildung 4.4 gezeigte Kennlinie, die e_{max} als Funktion der Dämpfung zeigt.

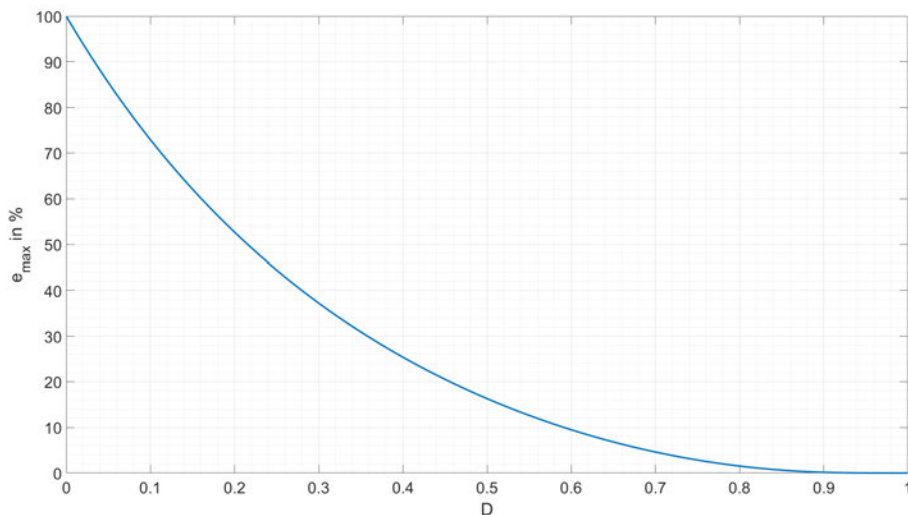


Abbildung 4.4: e_{max} in % als Funktion des Dämpfungsgrad in Bezug auf $h_{w,\infty} = 100\%$

Für den zu dimensionierenden Regler wird eine maximale Überschwingweite von 5% ausgewählt, was nach Abbildung 4.4 einer Dämpfung $D = 0,69$ entspricht.

4.3.2 Bestimmung der Verstärkung des Reglers

Mit der im letzten Abschnitt bestimmten Dämpfung kann nun die geforderte Verstärkung bestimmt werden. Dafür wird Gleichung 4.25 in dieselbe Form wie Gleichung 4.26 gebracht. Die umgestellte Übertragungsfunktion

$$G_w(s) = \frac{1}{\frac{T_s \cdot T_t}{K_p \cdot K_s} \cdot s^2 + \frac{T_s}{K_p \cdot K_s} \cdot s + 1} \quad (4.29)$$

ermöglicht nun einen Koeffizientenvergleich mit Gleichung 4.26. Der Koeffizientenvergleich ergibt:

$$\frac{1}{\omega_0^2} = \frac{T_s \cdot T_t}{K_p \cdot K_s} \quad (4.30)$$

und

$$\frac{2 \cdot D}{\omega_0} = \frac{T_s}{K_p \cdot K_s} \iff \frac{1}{\omega_0^2} = \left(\frac{T_s}{K_p \cdot K_s \cdot 2 \cdot D} \right)^2 \quad (4.31)$$

$\frac{1}{\omega_0^2}$ aus Gleichung 4.31 wird mit Gleichung 4.30 substituiert. Daraus folgt

$$\frac{T_s \cdot T_t}{K_p \cdot K_s} = \left(\frac{T_s}{K_p \cdot K_s \cdot 2 \cdot D} \right)^2 \quad (4.32)$$

was nach K_p umgestellt die folgende Lösung ergibt:

$$K_p = \frac{T_s}{4 \cdot D^2 \cdot K_s \cdot T_t} \quad (4.33)$$

Mit den für den Arbeitspunkt von $y_{AP} = 3,25V$ bestimmten Parametern $K_s = 0,528$, $T_t = 2,4$, $T_s = 23,06$, sowie der Dämpfung $D = 0,69$ wird der Regler mit einer Regelverstärkung von

$$K_p = \frac{23,06}{4 \cdot 0,69^2 \cdot 0,528 \cdot 2,4} = 9,53 \quad (4.34)$$

und der Zeitkonstante

$$T_n = T_s = 23,06 \text{ s} \quad (4.35)$$

dimensioniert.

4.4 Dimensionierung über Tune-Funktion

Der Industrie-Regler Bürklert 1110 hat eine Tune-Funktion, über die er die Reglerparameter selbstständig bestimmt. Der genaue Algorithmus für die Berechnung der Parameter ist unbekannt, aber laut [3](S. 73) wird ein modifiziertes Ziegler-Nichols-Verfahren angewendet. Die Anwendung der Tune-Funktion auf den Arbeitspunkt $y_{AP} = 3,25 \text{ V}$ liefert die folgenden Ergebnisse:

$$K_p = 4,6 \quad (4.36)$$

$$T_n = 5,628 \quad (4.37)$$

5 Training des Soft Actor-Critic Agenten

Die Umsetzung des Soft Actor-Critic Algorithmus und des dazugehörigen Trainings wird in zwei Schritte aufgeteilt. Die Entwicklung eines passenden Modells wird nicht direkt am Versuchsaufbau durchgeführt. Das hat mehrere Gründe: Aufgrund der nicht vermeidbaren Trägheit der Regelstrecke und der großen Anzahl der nötigen Testdurchläufe kann man nicht davon ausgehen, dass passende Ergebnisse innerhalb eines tragbaren Zeitraums gefunden werden können. Die Messungen für den Arbeitspunkt in Abschnitt 4.1 haben ergeben, dass die Regelstrecke ungefähr nach 120 Sekunden stationär ist. Das bedeutet für den Trainingsprozess, dass mindestens 120 Sekunden vergehen müssen, bevor eine Episode beginnen kann. Das hat folgenden Grund: Bevor der Agent anfangen kann, muss die Regelstrecke zuerst in eine Ausgangsposition gebracht werden. Geht man davon aus, dass ein gutes Ergebnis innerhalb von 200 Episoden erreicht werden kann, würde das zu einer Trainingsdauer von 800 Minuten bzw. 13 Stunden und 20 Minuten führen. Berücksichtigt man, dass an der Anlage nicht mehrere Trainingsläufe gleichzeitig laufen können und das für die Entwicklung des Modells viele Trainingsprozesse abgeschlossen werden müssen, folgt daraus, dass eine Lösung gefunden werden muss, die das Training verkürzt.

5.1 Implementierung in Matlab/Simulink

Um den Soft Actor-Critic Agenten in Matlab und Simulink trainieren zu können, werden zwei Dateien benötigt. Es wird Matlab-Script verwendet, um den Agenten, die Umwelt und eine Reset-Funktion zu definieren. Die zweite Datei ist ein Simulink-Modell, das die Interaktionen des im Matlab initialisierten Agenten und der Umwelt simuliert.

5.1.1 Matlab

Die Matlab-Datei wird auf Grundlage der Matlab-Dokumentation [11] implementiert und an das verwendete Modell angepasst. Zu Beginn müssen der Beobachtungsraum und der Aktionsraum definiert werden. Beide Räume sind kontinuierlich und der Aktionsraum wird zusätzlich noch auf einen Bereich von 0 V bis 10 V begrenzt, da dies vom Versuchsaufbau begrenzt wird. Dann wird die Umwelt initialisiert. Dafür wird das Simulink-Modell ausgewählt, da es die Interaktion zwischen Agent und Regelstrecke simuliert. Es wird auch noch eine Rücksetz-Funktion definiert, die zu Beginn jeder Episode ausgeführt wird und so das System initialisieren kann. In dieser Arbeit wird die Rücksetz-Funktion verwendet, um einen zufälligen Startpunkt, den zufällig ausgewählten Sollwert sowie die Raumtemperatur in das Simulink-Modell zu schreiben. Außerdem wird ein Zähler zurückgesetzt und initialisiert, dessen Funktion später erklärt wird.

Im nächsten Schritt wird der Agent initialisiert. Der Agent wird aus drei Komponenten zusammengesetzt. Zum einen muss der stochastische Spieler (eng. actor) initialisiert werden, der die zu lernende Strategie darstellt. Zum anderen gibt es zwei Kritiker, die jeweils über eine Zustands-Aktions-Value-Funktion dargestellt werden. Alle drei Teilnehmer werden als neuronales Netz implementiert. Der Agent wird erstellt, indem er als SAC-Agent initialisiert wird. Diesem SAC-Agenten werden der Spieler, die Kritiker sowie vorher festgelegte Optionen übergeben. Zuletzt müssen noch die Trainingsoptionen festgelegt werden, die dem „train“-Objekt zusammen mit dem Agenten und der Umwelt zum Start des Trainings übergeben werden. Ist das Training abgeschlossen, werden der Agent und zusätzliche Trainingsinformationen gespeichert. Um den Agenten als Regler einzusetzen, muss zuerst über einen Befehl die Strategiefunktion aus dem Agenten erzeugt werden. Der Befehl erzeugt eine Funktions-Datei und eine weitere Datei, in der die gelernte Strategie als neuronales Netz gespeichert ist. Der Funktions-Datei wird ein Vektor übergeben, der vom gleichen Typ sein muss wie der im Training definierte Zustandsvektor. Im Regelprozess wird der Funktion, die die Strategie auswertet, der Vektor übergeben. Der Vektor wird dann als Eingangsdaten an das neuronale Netz übergeben, das dann eine Vorhersage für die nächste Aktion macht.

5.1.2 Simulink-Modell

Das Simulink-Modell für das Training wird wie das Modell eines Regelkreises aufgebaut. Die zu regelnde Strecke wird als Übertragungsfunktion implementiert. An der Stelle, an der normalerweise der Regler ist, wird der Agent eingesetzt. Der Agent hat drei Eingänge und zwei Ausgänge. Am ersten Eingang wird dem Agenten der Zustandsvektor übergeben. Am zweiten Eingang wird dem Agenten die Belohnung übergeben, die wiederum über einen Funktionsblock bestimmt wird. Der dritte Eingang ist ein „isdone“-Flag, welches dem Agenten signalisiert, dass die Episode beendet werden soll. Das „isdone“-Flag kann dafür eingesetzt werden, um die laufende Episode frühzeitig abubrechen, für den Fall, dass ein zuvor definiertes Ereignis eintritt. Zum Beispiel, wenn das System eine festgelegte Grenze überschreitet. Soll die Episode nicht frühzeitig abgebrochen werden, kann das „isdone“-Flag durchgehend auf null gesetzt werden. Zu Beginn jeder Episode wird der Regelkreis zuerst auf einen bestimmten Sollwert geregelt. Das erfolgt durch einen PI-Regler. Während der PI-Regler aktiv ist, wird das Subsystem mit dem Agenten deaktiviert. Der Grund dafür ist, dass der Agent sonst schon während des Initialisierungsvorgangs anfangen würde zu lernen, auch wenn er selbst das System noch nicht beeinflussen kann. Nach einer im Zähler festgelegten Zeit wird das Subsystem mit dem Agenten aktiviert und der Agent übernimmt die Regelung.

5.1.3 Lernprozess

Wie bereits im vorherigen Abschnitt 5.1.1 erwähnt, lernt der Agent, drei Funktionen möglichst optimal zu nähern. Dazu gehören die zwei Q-Value-Funktionen, die möglichst optimal über ein neuronales Netz genähert werden sowie die stochastische Strategie, die auch über ein neuronales Netz approximiert wird. Nach [10] hat der stochastische Spieler die Parameter θ . Dem Spieler wird der Zustand s_t des Systems übergeben. Der Spieler $\pi(s_t|\theta)$ gibt dann die Wahrscheinlichkeitsdichtefunktion der Aktionen zurück, anhand derer der Agent die nächste Aktion zufällig entscheidet. Die zwei Q-Value-Funktionen sind nach [10] wie folgt definiert: Die erste Q-Value-Funktion hat die Parameter ϕ_k . Der Q-Value-Funktion $q_k(s_t, a_t|\phi_k)$ wird der Zustand s_t und die Aktion a_t übergeben und gibt dann - wie in Abschnitt 3.3 erläutert - den erwarteten Wert zurück, der sich aus Return und Entropie zusammensetzt. Die zweite Q-Value-Funktion $q_{tk}(s_t, a_t|\phi_{tk})$ ist für den Ziel-Kritiker. Die Aufgabe des Ziel-Kritikers ist es, die Stabilität und Optimierung des Agenten zu verbessern. Um das zu erreichen, werden die

Parameter ϕ_{tk} der Q-Value-Funktion des Ziel-Kritikers in festgelegten Zeitabständen mit den aktuellen Parametern ϕ_k des anderen Kritikers gleichgesetzt.

Die in Matlab implementierte Variante des Soft Actor-Critic Algorithmus läuft nach [10] wie folgt ab: Bevor das Training startet, werden die Parameter des Spielers θ und die Parameter des Kritikers ϕ_k mit Zufallswerten initialisiert. Es wird außerdem $\phi_{tk} = \phi_k$ gesetzt. Bevor das Lernen anfängt, wird außerdem eine Reihe von n Aktionen ausgeführt, die auf Basis der zufällig initialisierten Startstrategie ausgewählt werden. Für jede Aktion wird der zugehörige Vektor $(s_t, a_t, r_{t+1}, s_{t+1})$ als Erfahrungswert im Erfahrungspuffer gespeichert. Die n Aktionen, die zu Beginn ausgeführt werden, entsprechen standardmäßig der Mini-Batch Größe. Das tatsächliche Lernen läuft dann wie folgt ab: In jedem Trainingsschritt wird zunächst die Aktion a_t auf Basis der aktuellen Strategie und des ihr übergebenen Zustands s_t bestimmt. Es wird dann ein Zeitschritt gewartet, bis dem Agenten die Belohnung r_{t+1} für das letzte Zustand-Aktions-Paar (s_t, a_t) sowie die nächste Beobachtung s_{t+1} übergeben wird. Der Zustand s_{t+1} , der aus dem Zustand-Aktions-Paar (s_t, a_t) resultiert, wird im Weiteren als s'_t bezeichnet. Diese vier Werte werden dann als Vektor in der Form $(s_t, a_t, r_{t+1}, s'_t)$ im Erfahrungspuffer gespeichert. Ein Mini-Batch der Größe n wird im nächsten Schritt aus n zufällig gewählten Erfahrungswerten (s_i, a_i, r_i, s'_i) des Erfahrungspuffers erstellt. Mithilfe der Erfahrungswerte, die im Mini-Batch gespeichert sind, werden dann die Parameter der Kritiker und des Spielers aktualisiert. Die Anzahl der Zeitschritte zwischen den Aktualisierungen kann geändert werden. Standardmäßig wird bei jedem Zeitschritt aktualisiert.

Nach [10] laufen die Aktualisierungen wie folgt ab: Um die Parameter ϕ_k des Kritikers zu aktualisieren, wird die Verlustfunktion

$$L_k = \frac{1}{n} \cdot \sum_{i=1}^n (y_i - q_k(s_i, a_i | \phi_k))^2 \quad (5.1)$$

über die Summe der im Mini-Batch gespeicherten Erfahrungswerte minimiert. y_i wird als das Ziel der Value-Funktion bezeichnet und ist definiert als die Summe der Belohnung, dem Minimum des mit γ reduzierten erwarteten Return des Ziel-Kritikers und der mit α gewichteten Entropie.

$$y_i = r_i + \gamma \cdot \min_k (q_{tk}(s'_i, a'_i | \phi_{tk})) - \alpha \cdot \ln \pi(s'_i | \theta) \quad (5.2)$$

Die Parameter des Spielers werden aktualisiert, indem die Funktion

$$J_\pi = \frac{1}{n} \sum_{i=1}^n \left(-\min_k (q_{tk}(s_i, a_i | \phi_{tk}) + \alpha \ln \pi(s_i | \theta)) \right) \quad (5.3)$$

minimiert wird. Zuletzt wird die Gewichtung α der Entropie aktualisiert, indem die Verlustfunktion

$$L_\alpha = \frac{1}{n} \sum_{i=1}^n (-\alpha \ln \pi(s_i | \theta) - \alpha H) \quad (5.4)$$

über die Summe der im Mini-Batch gespeicherten Erfahrungswerte minimiert wird. H ist hier die Ziel-Entropie, die vor dem Training definiert werden kann. Die jeweilige Minimierung der hier genannten Funktion erfolgt, indem die entsprechenden Parameter über den stochastischen Gradientenabstieg optimiert werden. Für jede Aktualisierung wird jeweils für einen Schritt des stochastischen Gradientenabstiegs die Optimierung durchgeführt.

5.2 Training am mathematischen Modell

Die in Kapitel 2.3 bestimmte diskrete und kontinuierliche Übertragungsfunktion

$$G_s(s) = \frac{K_s}{T_s \cdot s + 1} \cdot e^{-T_t \cdot s} \quad (5.5)$$

$$G_s(z) = \frac{b_1 \cdot z + b_2}{a_1 \cdot z + a_2} \cdot z^{-d} \quad (5.6)$$

der Regelstrecke ist die Grundlage für das mathematische Modell. Das Ziel des Agenten ist es, die Ausgangsgröße der Regelstrecke auf den gewünschten Sollwert zu regeln. Das Verhalten des Reglers bzw. Agenten hängt, wie in 2.3 beschrieben, von mehreren Faktoren ab. Die entscheidenden Fragestellungen, die man beantworten muss, sind:

- Wie soll der Agent das System beeinflussen?
- Welche Informationen über das System braucht der Agent?
- Was sind die Anforderungen an die Regelung?

5.2.1 Kompensation der Totzeit

Um ein Modell der Regelstrecke in Simulink zu erstellen, das möglichst nah am Verhalten des Versuchsaufbaus liegt, bleibt das Totzeitglied im Simulink Modell erhalten. Das hat aber gravierende Nebenwirkungen auf den Trainingsprozess und das daraus resultierende Regelverhalten. Das zeigt der Vergleich der beiden Kurven aus Abbildung 5.1. Beide Agenten wurden auf dieselbe Regelstrecke unter denselben Bedingungen trainiert. Es wurden 150 Episoden trainiert. Jede Episode hatte einen Simulations-Zeitraum von 120 Sekunden bei einer Abtastzeit von 0,2 Sekunden. Das bedeutet, eine Episode hat 600 Zeitschritte. Ohne Kompensation braucht der Agent länger bis es zu einer Verbesserung der der Belohnung kommt, die aber trotzdem immer noch nicht gut ist. Im Gegensatz zum Training mit Kompensation der Totzeit, bei dem der Agent nicht nur schneller in den Bereich der Belohnung vom Agenten ohne Totzeitkompensation kommt, sondern den anderen Agenten sogar um Weiten übertrifft. Der Agent kommt in einen Bereich zwischen drei- und viertausend.

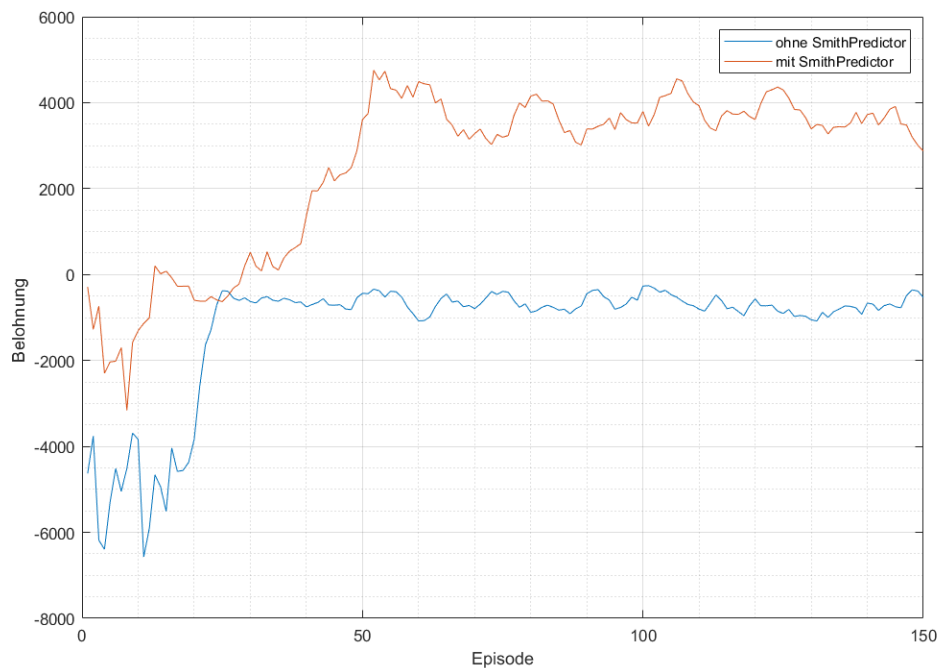


Abbildung 5.1: Vergleich der Auswirkung der Totzeit-Kompensation auf die mittlere Belohnung

Die in diesem Versuch eingesetzte Belohnungsfunktion

$$r_{t+1} = 10 \cdot \left(1 - \frac{|e_t|}{0,05}\right) \cdot I(|e_t|) - 10 \cdot |e_t| \quad (5.7)$$

mit

$$I(e_t) = \begin{cases} 1 & \text{falls } |e_t| \leq 0,05 \\ 0 & \text{sonst} \end{cases} \quad (5.8)$$

hat ihren maximalwert bei $|e_t| = 0$.

$$\max(r_t) = 10 \quad (5.9)$$

Die Belohnungsfunktion kann theoretisch einen maximalen Return pro Episode von

$$\max(G_t) = \sum_{k=0}^{600} \max(r_t) = 6000 \quad (5.10)$$

erreichen. Praktisch wird dieser Fall niemals eintreten, weil die Regelgröße aufgrund des PT_1 -Glieds mit Totzeit nicht sofort den Wert der Regelgröße annehmen kann. Die Belohnungsfunktion wird in Abschnitt 5.2.4 näher untersucht.

Die niedrige mittlere Belohnung erzeugt die Vermutung, dass das Regelverhalten wahrscheinlich nicht gut sein wird. Abbildung 5.2 bestätigt die Vermutung. Wie erwartet, ist der Verlauf der Stellgröße sowie der Regelgröße nicht optimal. Es kommt zu einem Sägezahn-Verlauf der Regel- und Führungsgröße. Betrachtet man den Verlauf der Stellgröße, erkennt man, dass der Agent gelernt hat, das System über ein Rechteck-Impuls-ähnliches Signal zu regeln. Vergleicht man den Verlauf der Regelgröße und der Kenngröße fällt auf, dass der Regler erst reagiert, nachdem der Sollwert über- oder unterschritten wird. Er ändert dann die Stellgröße. Das führt dazu, dass die Regelgröße eine Sägezahnform um die Führungsgröße annimmt. Der Grund für das schlechte Verhalten ist, dass die Temperaturänderung, die aus der Änderung der Stellgröße folgt, erst nach 2,4 Sekunden, was der Totzeit entspricht, am Messpunkt ankommt.

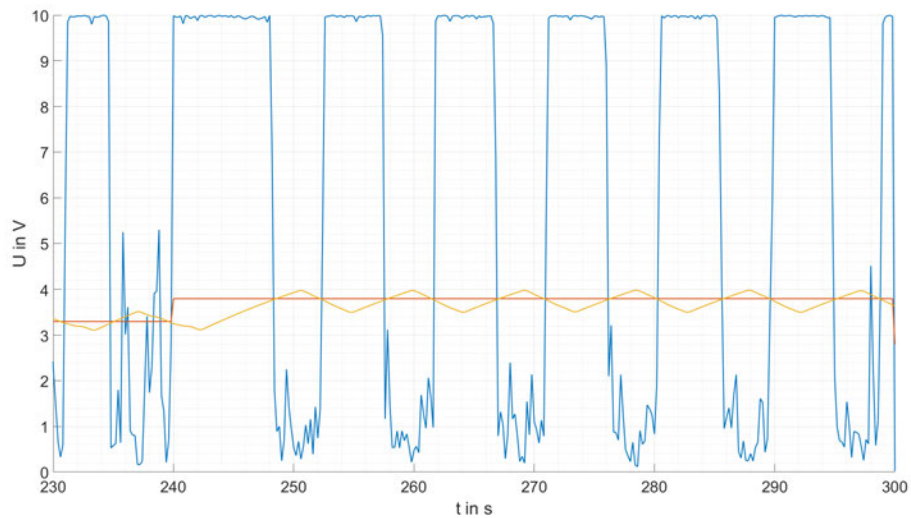


Abbildung 5.2: Regelverhalten ohne Totzeitgliedkompensation bei einem Sollwertsprung, rot: Führungsgröße, gelb: Regelgröße, blau: Stellgröße

Der Agent müsste also schon früher gegensteuern. Dieses suboptimale Verhalten entsteht dadurch, dass der Agent im Trainingsprozess den aktuellen Zustand, die aktuelle Aktion und die daraus resultierenden Belohnung in Zusammenhang bringt. Der übergebene Zustand des Systems basiert auf dem am Ende der Strecke angebrachten Messwert des PT100. Das daraus folgende Problem ist, dass der Zustand und die Belohnung, die der Agent mit der aktuellen Aktion verbindet, nicht von eben dieser Aktion abhängt, sondern von der Aktion, die vor T_t Sekunden ausgeführt wurde. Anders formuliert: Die aktuelle Aktion hat erst in T_t Sekunden eine Auswirkung auf das System und der Agent müsste in die Zukunft blicken, um die Aktion bewerten zu können. Es muss also ein Möglichkeit gefunden werden, um den Zustand des Systems in T_t Sekunden vorherzusagen.

Smith-Prädiktor

Hier setzt der Smith-Prädiktor aus Abbildung 5.3 an ([14], S.290-291).

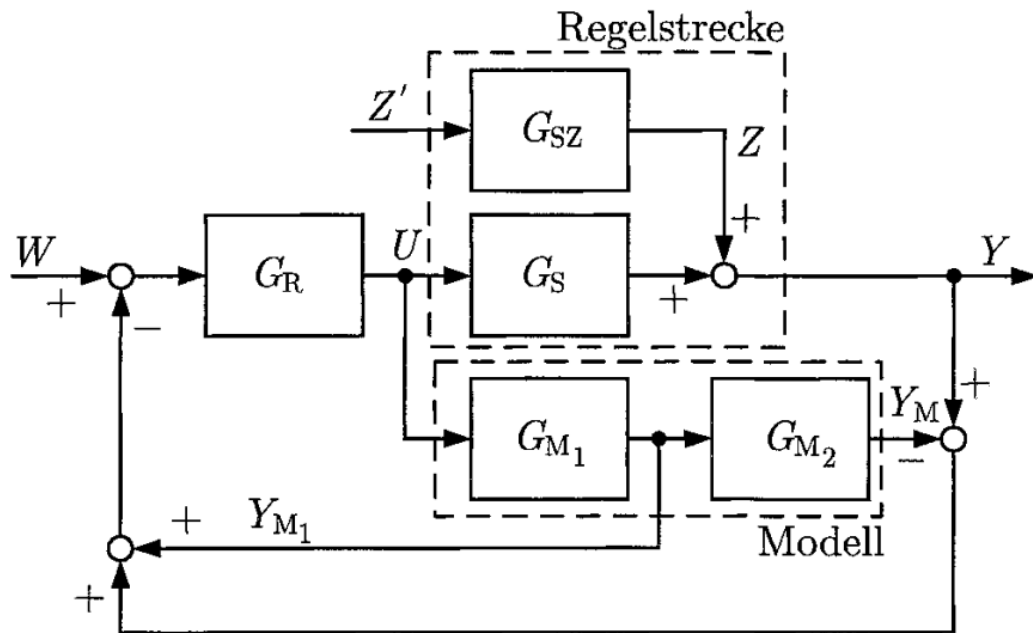


Abbildung 5.3: Blockschaltbild des Smith-Prädikors [14] S.291

Der Smith-Prädiktor erweitert nach ([14], S.290-291) den geschlossenen Regelkreis um ein Modell, welches das Verhalten der Regelstrecke möglichst genau wiedergibt. Das Modell besteht aus zwei in Reihe geschalteten Übertragungsfunktionen $G_{M1}(s)$ und $G_{M2}(s)$, welche zusammen die Übertragungsfunktion $G_s(s)$ der Regelstrecke nachbilden. Es gilt:

$$G_s(s) = G_{M1}(s) \cdot G_{M2}(s) \quad (5.11)$$

Die Übertragungsfunktion $G_{M1}(s)$ entspricht der Übertragungsfunktion der Regelstrecke $G_s(s)$ ohne das Totzeitglied und $G_{M2}(s)$ besteht wiederum nur aus dem Totzeitglied. Das Modell bekommt wie die Regelstrecke die Stellgröße des Reglers als Eingangssignal. Im Gegensatz zur Standard-Variante wird hier nicht die Regelgröße $Y(s)$ der Strecke zurück gekoppelt, sondern ein Signal, das zwei Bestandteile hat. Teil 1 ist die Antwort des Modells $Y_{M1}(s)$, die das Verhalten der Strecke vorhersagt. Teil 2 bildet die Differenz aus der Antwort der Regelstrecke und dem Modell mit Totzeitglied, um eventuell Abweichungen durch Störgrößen oder Ungenauigkeiten des Modells zu berücksichtigen. Die Summe der beiden Werte wird dann dem Regler über den Regelfehler übergeben.

Für die Umsetzung des Modells der Regelstrecke, welches die eigentliche Vorhersage der Regelgröße bestimmt, werden zwei Varianten betrachtet. Die erste Variante ist ein Funktionsblock, in dem über die aus der z-Transformation gebildeten Rekursivformel die Regelgröße bestimmt wird. Für die zweite Variante wird, wie in [14] (S. 290) vorgeschlagen, die Übertragungsfunktion der Strecke ohne Totzeitglied als Simulink Block implementiert.

z-Transformation

Für die Transformation der Übertragungsfunktion $G_{M1}(s)$ aus dem kontinuierlichen Zeitbereich in den diskreten Zeitbereich wird die in [12](S. 274) beschriebene Approximation der diskreten Übertragungsfunktion mithilfe der bilinearen-z-Transformation verwendet. Es gilt:

$$s = \frac{1}{T_f} \cdot \ln z \quad (5.12)$$

Die Entwicklung der Potenzreihe von Gleichung 5.12

$$s = \frac{1}{T_f} \cdot \ln z = \frac{2}{T_f} \cdot \left\{ \left(\frac{z-1}{z+1} \right) + \frac{1}{3} \left(\frac{z-1}{z+1} \right)^3 + \frac{1}{5} \left(\frac{z-1}{z+1} \right)^5 + \dots \right\} \quad (5.13)$$

und dem Verwerfen der Potenzreihe nach dem ersten Element ergibt nach [12](S.275) den folgenden Substitutionsausdruck für s:

$$s = \frac{2}{T_f} \cdot \frac{z-1}{z+1} \quad (5.14)$$

Nun wird die Übertragungsfunktion der Strecke aus Gleichung 2.14 ohne Totzeit durch die Substitution von s mit Gleichung 5.14 in den diskreten-Zeitbereich transformiert, wobei T_f die Abtastzeit ist:

$$\begin{aligned} G_S(s) &= \frac{\vartheta(s)}{P_{el}(s)} = \frac{K_S}{1 + T_S \cdot s} \Rightarrow G_S(z) = \frac{K_S}{1 + T_S \cdot \frac{2}{T_f} \frac{z-1}{z+1}} \\ G_S(z) &= \frac{K_S}{1 + T_S \cdot \frac{2}{T_f} \frac{z-1}{z+1}} = \frac{\frac{K_S \cdot T_f}{2 \cdot T_S} \cdot z + \frac{K_S \cdot T_f}{2 \cdot T_S}}{\left(\frac{T_f}{2 \cdot T_S} + 1 \right) \cdot z + \left(\frac{T_f}{2 \cdot T_S} - 1 \right)} = \frac{\frac{K_S \cdot T_f}{2 \cdot T_S + T_f} \cdot z + \frac{K_S \cdot T_f}{2 \cdot T_S + T_f}}{z + \frac{T_f - 2 \cdot T_S}{T_f + 2 \cdot T_S}} \end{aligned} \quad (5.15)$$

Mit

$$\begin{aligned}
 a_1 &= 1 \\
 a_2 &= \frac{T_f - 2 \cdot T_s}{T_f + 2 \cdot T_s} \\
 b_1 &= \frac{K_s \cdot T_f}{2 \cdot T_s + T_f} \\
 b_2 &= \frac{K_s \cdot T_f}{2 \cdot T_s + T_f}
 \end{aligned}$$

erhält man:

$$G_S(z) = \frac{Y(z)}{U(z)} = \frac{b_1 + b_2 \cdot z^{-1}}{a_1 + a_2 \cdot z^{-1}} \quad (5.16)$$

Theoretisch kann die Übertragungsfunktion in dieser Form als $G_{M1}(s)$ in den Smith-Prädiktor des Simulink Modells implementiert werden. Simulink erkennt dann einen algebraischen Kreis, der zu Problemen beim Kompilieren des Modells führt. Aus diesem Grund wird $G_S(z)$ in eine Differentialgleichung zurückgeführt, mit der $Y(z)$ bestimmt werden kann. $G_S(z)$ wird in die Differentialgleichung.

$$a_1 \cdot Y(z) + a_2 \cdot Y(z) \cdot z^{-1} = b_1 \cdot U(z) + b_2 \cdot U(z) \cdot z^{-1} \quad (5.17)$$

umgeformt. Mit

$$Y(z) = y_z \quad (5.18)$$

$$Y(z) \cdot z^{-1} = y_{z-1} \quad (5.19)$$

$$U(z) = u_z \quad (5.20)$$

$$U(z) \cdot z^{-1} = u_{z-1} \quad (5.21)$$

folgt die rekursive Gleichung

$$y_z = \frac{b_1 \cdot u_z + b_2 \cdot u_{z-1} - a_2 \cdot y_{z-1}}{a_1} \quad (5.22)$$

mit der die Regelgröße y_z zu einem beliebigen Abtastzeitpunkt z als Funktion des aktuellen Werts der Stellgröße sowie dem vorherigen Wert der Stell- und Regelgröße bestimmt werden kann. Gleichung 5.22 kann dann als Funktionsblock implementiert werden. Der Funktion werden die folgenden Werte übergeben. Sie erhält Faktoren a_i und b_i jeweils als Vektor, sowie die aktuelle Regelgröße. Außerdem erhält sie einen Vektor mit den Werten, die die Stellgröße in den letzten $d = \frac{T_L}{T_f}$ Zeitschritten angenommen hat. Aus dem aktuellen Wert der Regelgröße und den letzten d Werten der Stellgrößen wird die Regelgröße in d Zeitschritten iterativ bestimmt. Der Agent, der in einem Modell mit Smith-Prädiktor trainiert wird, weist im Vergleich zum Modell ohne Smith-Prädiktor ein besseres Regelverhalten auf, wie Abbildung 5.4 bestätigt. Abbildung 5.4 zeigt den Verlauf der Regelgröße und der Stellgröße auf einen Sollwertsprung. Der Verlauf der Regel- und Stellgröße ähnelt, wie ein Vergleich mit Abbildung 5.5 zeigt, dem Verlauf der Regel- und Stellgröße eines PI-Reglers.

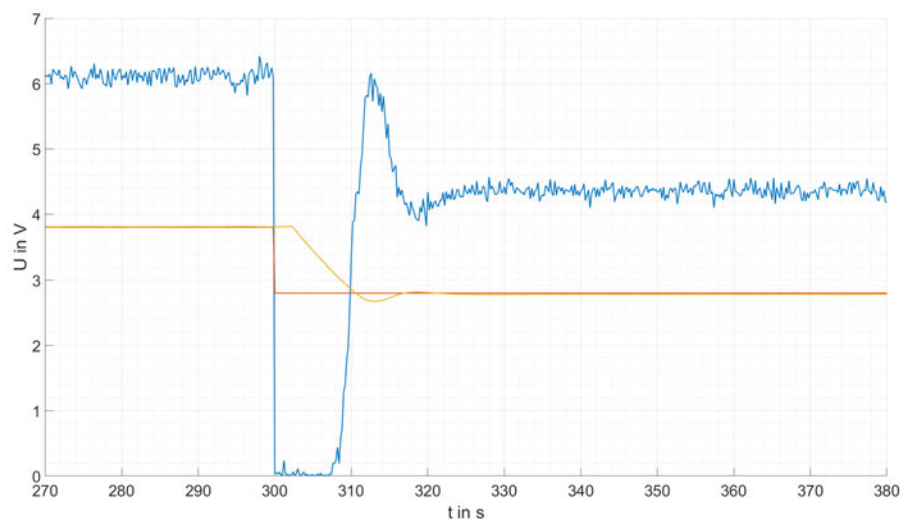


Abbildung 5.4: Regelverhalten des mit Smith-Prädiktor trainierten Agenten bei einem Führungsgrößensprung, rot = Führungsgröße, gelb = Regelgröße, blau = Stellgröße

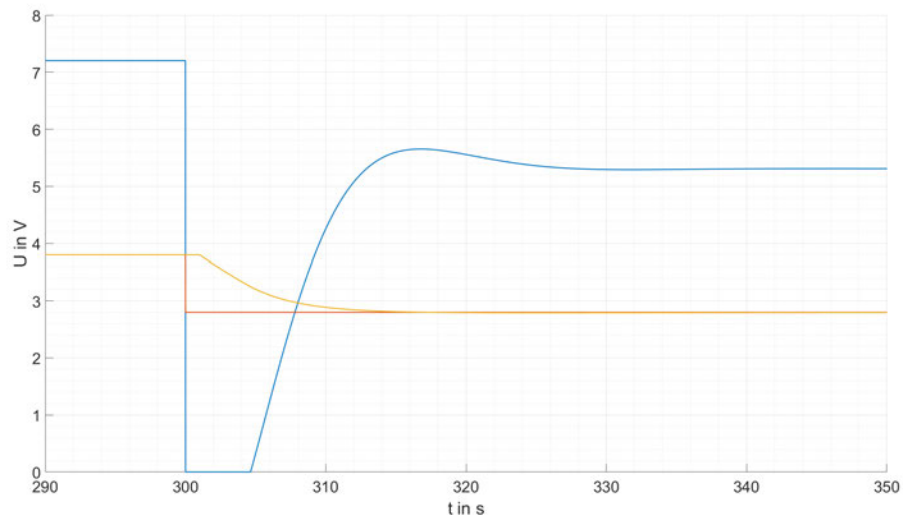


Abbildung 5.5: Regelverhalten des im Frequenzbereich dimensionierten Reglers bei einem Sollwertsprung, rot = Führungsgröße, gelb = Regelgröße, blau = Stellgröße

5.2.2 Zustandsvektor

Die zweite Fragestellung bestimmt, wie der Zustandsvektor des Systems aufgebaut sein soll. Geht man von einem Standardregler aus, erhält dieser nur eine Eingangsgröße und zwar den Regelfehler.

Trainings- und Simulationsbedingungen

Die in diesem Kapitel untersuchten Agenten werden jeweils für 150 Episoden trainiert. Eine Episode wird für eine Länge von 120 Sekunden simuliert mit einer Abtastzeit $T_f = 0,2$ s. Daraus folgt, dass eine Episode $n = 600$ Zeitschritte umfasst. Der Startpunkt einer Episode wird zufällig als 2,5 V oder 5 V initialisiert. Es wird dann zufällig ein Wert zwischen 0 V und 2,5 als Sollwertsprung gewählt. Liegt der Startpunkt bei 2,5 V, ist der Sollwertsprung positiv. Liegt der Startpunkt bei 5 V, ist der Sollwertsprung negativ. Es werden jeweils vier Agenten trainiert und der beste für die Simulation ausgewählt. Für die Simulation wird die Regelstrecke mit einem Standard-PI-Regler auf einen Sollwert von 3 V geregelt. Bei $t = 150$ s kommt ein Führungsgrößensprung auf 3,5 V und der Agent übernimmt die Regelung.

Regelfehler als Zustandsvektor

Das Training des Agenten sowie die Simulation des trainierten Agenten mit einem Zustandsvektor $s_t = (e_t)$ liefert die folgenden Ergebnisse. Wie Abbildung 5.6 zeigt, ist die mittlere summierte Belohnung, die der Agent innerhalb von 150 Belohnungen und bei 3 unterschiedlichen Seeds erreicht, maximal ungefähr 1000 - bleibt dann aber in einem Bereich zwischen 0 und 500. Es gibt zwar zu Beginn eine starke Verbesserung, aber das Ergebnis konvergiert dann relativ schnell und ändert sich nicht mehr stark. Aus den erreichten Belohnungen lässt sich schon vermuten, dass der Agent kein gutes Regelverhalten gelernt haben wird.

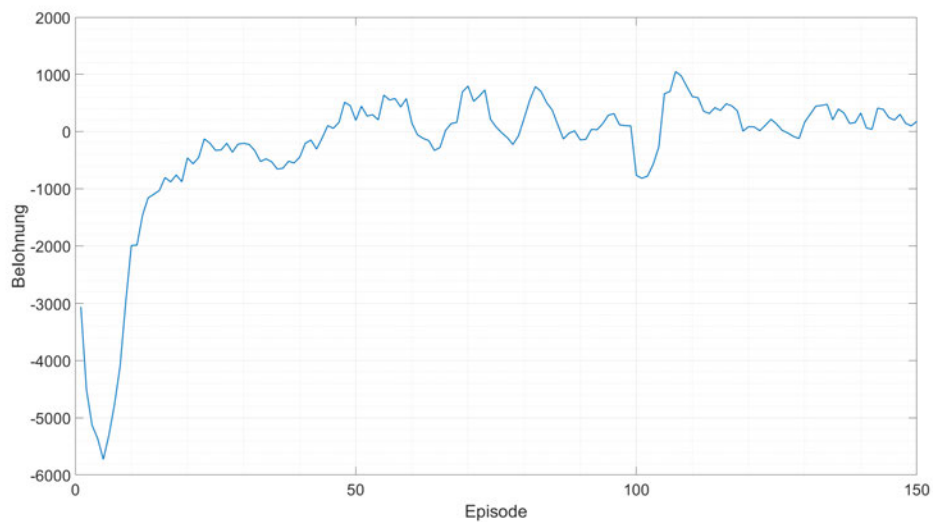


Abbildung 5.6: Verlauf der mittleren Belohnung über das Training mit dem Regelfehler als Zustand

Abbildung 5.7 zeigt den Verlauf der Regelgröße bei einem Führungsgrößensprung und Abbildung 5.8 zeigt den Verlauf der Stellgröße auf den in Abbildung 5.7 gezeigten Sollwertsprung. Die Betrachtung der beiden Abbildungen bestätigt die Vermutung, dass das erlernte Regelverhalten des Agenten nicht die Anforderungen aus Kapitel 4 erfüllt. In beiden Abbildungen übernimmt der Agent ab $t = 150$ s die Regelung der Regelstrecke. Aus Abbildung 5.7 erkennt man, dass das Stellsignal ungefähr 60 Sekunden nach dem Führungsgrößensprung eingeschwungen ist und ab dann nahezu periodisch um die Führungsgröße schwingt. Der Agent

schaft es, das Temperatursignal in den Bereich des Sollwertes zu regeln, aber es kommt zu einem schwingenden Verhalten und somit auch zu einem bleibenden Regelfehler. Der aus Abbildung 5.7 bestimmte, maximale bleibende Regelfehler beträgt:

$$e_{\max, \infty} = \frac{3,5 \text{ V} - 3,4 \text{ V}}{3 \text{ V} - 3,5 \text{ V}} = 0,2 = 20\% \quad (5.23)$$

Desweiteren ist die maximale Überschwingweite

$$e_{\max} = \left| \frac{3,5 \text{ V} - 3,331 \text{ V}}{3 \text{ V} - 3,5 \text{ V}} \right| \approx 0,338 = 33,8\% \quad (5.24)$$

des Sollwertsprungs.

Das Stellsignal des Agenten ist dementsprechend auch schlecht. Nach dem Einschwingvorgang kommt es zu einer bleibenden Schwingung des Stellsignals in einem Bereich zwischen 4 V und 8 V.

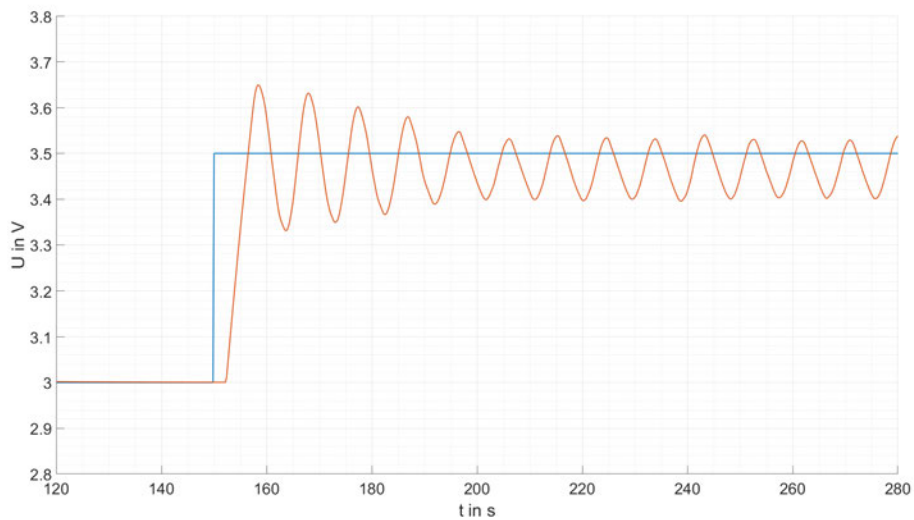


Abbildung 5.7: Verlauf der Regelgröße (orange) und Führungsgröße (blau) bei einem Zustandsvektor $s_t = (e_t)$

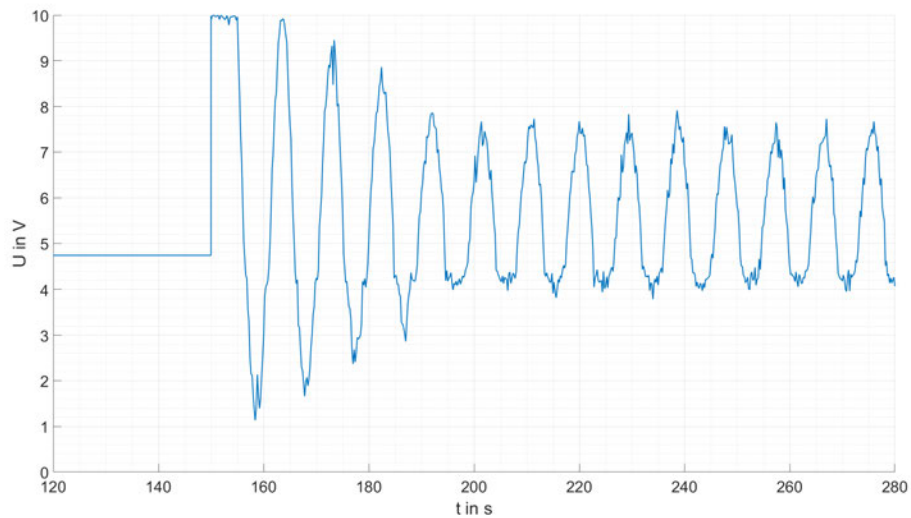


Abbildung 5.8: Verlauf der Stellgröße (blau) bei einem Sollwertsprung bei der Regelgröße als Zustand

Der Grund für dieses schlechte Regelverhalten ist der gewählte Zustandsvektor. Das Ziel des Agenten ist es, den Regelfehler möglichst schnell zu minimieren. Im Idealfall wird der Regelfehler null. Bekommt der Agent nur den Regelfehler, also die Differenz zur Führungsgröße, entsteht daraus folgendes Problem. Der Regelfehler, der dem Agenten als Zustand bei einem Sollwertsprung von 3 V auf 4 V und einem Sollwertsprung von 5 V auf 6 V zu Beginn übergeben wird, ist in beiden Fällen gleich. Die Stellgröße, die im eingeschwungenen Zustand benötigt wird, um die Regelgröße auf 4 V zu Regeln, ist nicht dieselbe wie für 6 V. Angenommen, der Agent befindet sich im Training. Die Episode wird mit einer Führungsgröße $w_t = 4$ initialisiert. Zu Beginn der Episode hat die Regelgröße den Wert $y_t = 3$. Daraus folgt für den Zustand im Zeitpunkt 0

$$s_0 = e_0 = w_0 - y_0 = 4 - 3 = 1 \quad (5.25)$$

Zum Ende der Episode hat der Agent gelernt, dass im eingeschwungenen Zustand das Stellsignal sechs betragen muss, wenn die Regelgröße den Wert vier annehmen soll. Damit wäre der Regelfehler 0.

Wie bereits erläutert, ist der Zustand $s_t = 1$ nicht einzigartig, denn $s_t = 1$ gilt auch für $w = 1.4$ und $y = 0.4$ oder für $w = 10$ und $y = 9$. Würde der Agent nun bei $w = 10$ und $y = 9$ die Aktion $a_t = 6$ wählen, weil er gelernt hat, dass diese Aktion bei $s_t = 1$ den Regelfehler minimiert, kommt es dazu, dass die Regeldifferenz nicht kleiner sondern größer wird. Das würde bedeuten, dass die Belohnung, die der Agent bekommt, niedrig ist, was wiederum seinem bisherigen Wissen widerspricht. Aus dem Test folgt, dass die Regeldifferenz alleine nicht ausreicht, um den Zustand des Systems genau genug zu beschreiben, um ein gutes Regelverhalten zu lernen.

Für die Erweiterung des Zustandsvektors, um den Zustand des Systems eindeutig beschreiben zu können, bieten sich zwei Kenngrößen an. Zum einen kann der Zustandsvektor um die Führungsgröße w erweitert werden. Zum anderen kann der Zustandsvektor um die Regelgröße y erweitert werden. Theoretisch führen beide Varianten zu einer eindeutigen Zuweisung.

Regelfehler und Führungsgröße als Zustandsvektor

Als nächstes wird der Zustandsvektor mit der Führungsgröße w_t erweitert. Die Trainingsbedingungen des Agenten sind identisch mit denen aus dem vorherigen Abschnitt. Mit der Ausnahme des Zustandsvektors. Der Verlauf der mittleren Belohnung über den Trainingszeitraum wird in Abbildung 5.9 dargestellt. Ein Vergleich mit der mittleren Belohnung aus Abbildung 5.6 für $s_t = (e_t)$ zeigt, dass die Erweiterung des Zustandsvektors zu einem besseren Ergebnis führt. Die mittlere Belohnung erreicht zum Ende des Trainings einen Bereich zwischen zwei- und dreitausend. Daraus lässt sich vermuten, dass das Regelverhalten des Agenten besser sein sollte als im vorherigen Abschnitt.

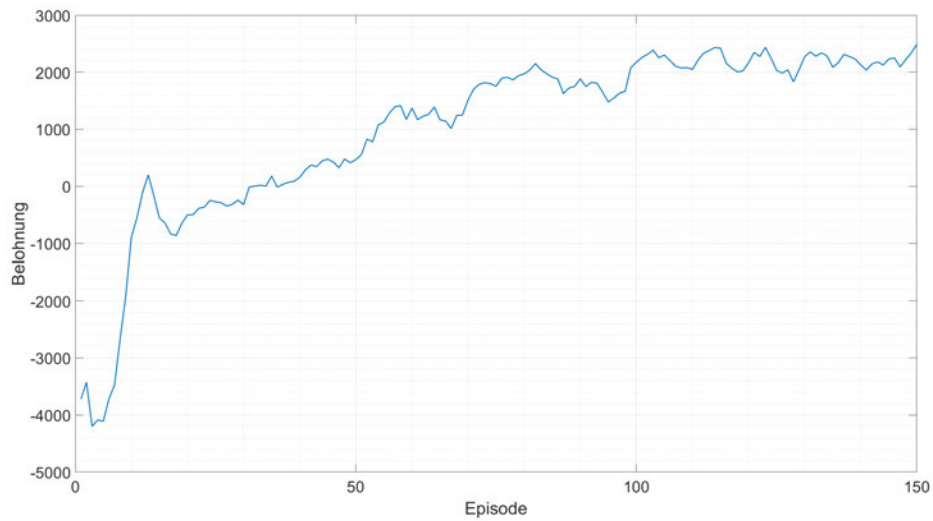


Abbildung 5.9: Verlauf der mittleren Belohnung über das Training bei Agenten mit Regelfehler und Führungsgröße als Zustand

Die Simulation des Agenten an der Regelstrecke bestätigt die Vermutung, dass das Regelverhalten besser ist. Betrachtet man den Verlauf der Führungsgröße aus Abbildung 5.10, erkennt man, dass sich die Regelgröße bereits nach dem Überschwingen nicht mehr stark verändert. Es bleibt ein leicht schwingender Regelfehler, der sein Maximum bei

$$e_{\max, \infty} = \left| \frac{3,5 \text{ V} - 3,5093 \text{ V}}{3; \text{ V} - 3,5 \text{ V}} \right| \approx 0,0186 = 1,86\% \quad (5.26)$$

des Sollwertsprungs hat. Der Regelfehler des ersten Überschwingens beziehungsweise die maximale Überschwingweite, e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,5 \text{ V} - 3,59 \text{ V}}{3 \text{ V} - 3,5 \text{ V}} \right| \approx 0,18 = 18\% \quad (5.27)$$

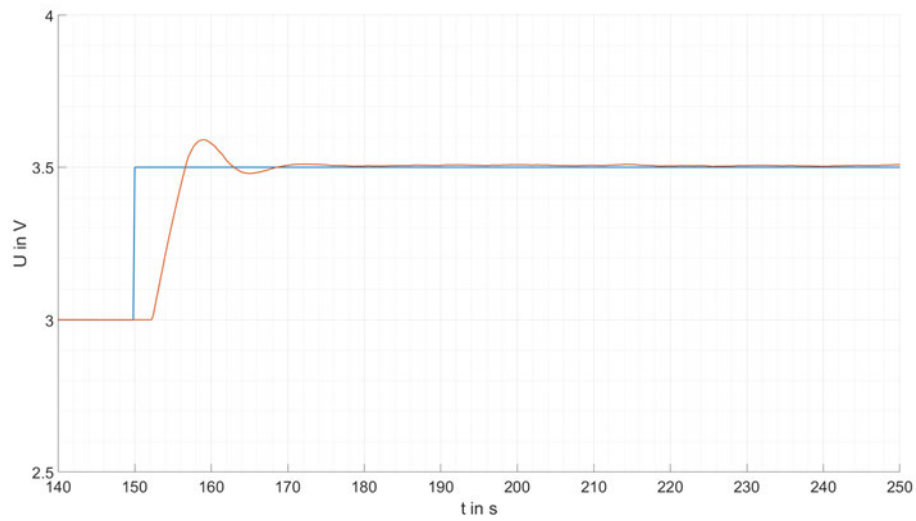


Abbildung 5.10: Verlauf der Regelgröße(orange) auf einen Führungsgrößensprung(blau) bei Agent mit Regelfehler und Führungsgröße als Zustand

Abbildung 5.11 zeigt, dass auch der Verlauf der Regelgröße besser ist als in Abbildung 5.8. Die Stellgröße erreicht 30 Sekunden nach dem Führungsgrößensprung einen näherungsweise stationären Wert, auch wenn das Stellsignal ein hochfrequentes Rauschen hat.

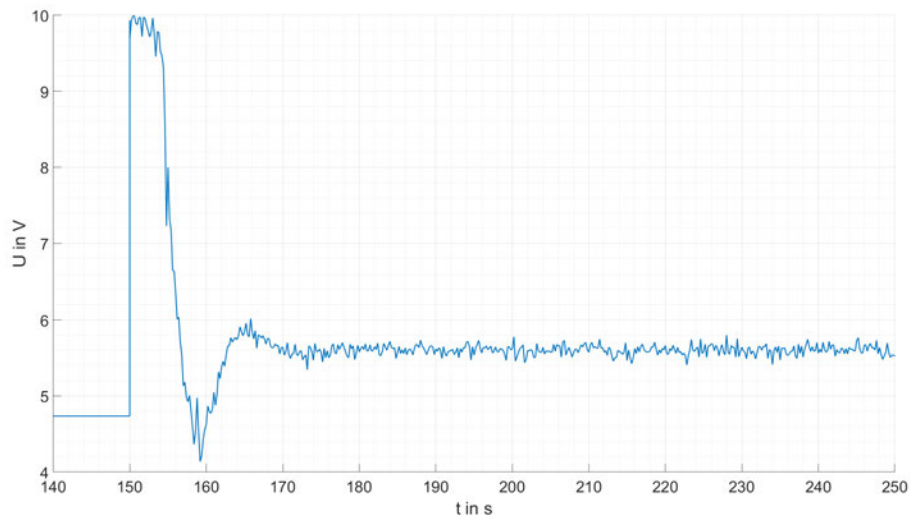


Abbildung 5.11: Verlauf der Stellgröße auf einen Führungsgrößensprung bei Agent mit Regelfehler und Führungsgröße als Zustand

Regelfehler und Regelgröße als Zustandsvektor

Die letzte Variante erweitert den Zustandsvektor $s_t = (e_t)$ mit der Regelgröße y_t . Wie schon zuvor wird auch hier der Agent vier mal trainiert und die mittlere Belohnung der vier Trainingsdurchläufe in 5.12 dargestellt. Ein Vergleich mit 5.9 zeigt, dass beide Varianten des Zustandsvektors ähnliche Belohnungswerte im Training erreichen. Der Vergleich könnte vermuten lassen, dass die beiden Agenten ein ähnliches Regelverhalten gelernt haben.

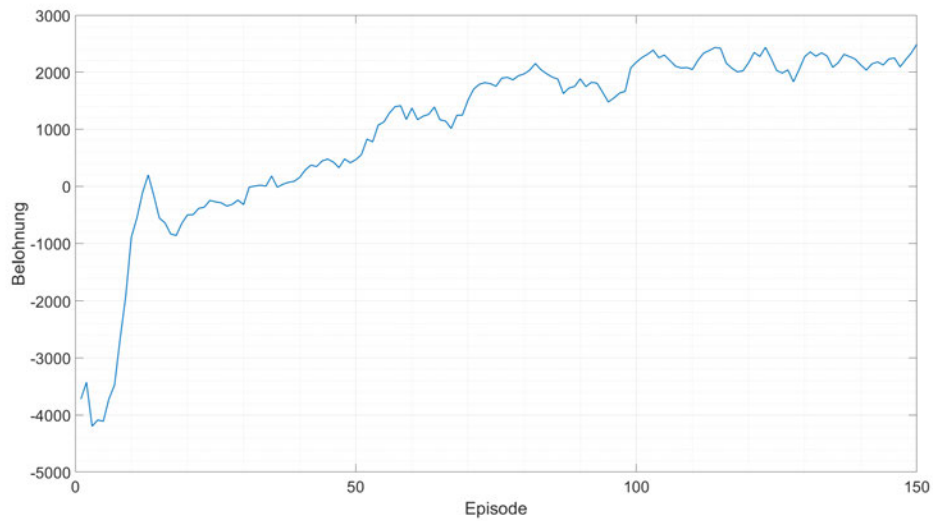


Abbildung 5.12: Verlauf der mittleren Belohnung über das Training bei Agenten mit Regelfehler und Regelgröße als Zustand

Die Untersuchung des Verlaufs der Regelgröße aus Abbildung 5.13 ergibt eine maximal bleibende Regelabweichung

$$e_{\max, \infty} = \left| \frac{3,5 \text{ V} - 3,495 \text{ V}}{3 \text{ V} - 3,5 \text{ V}} \right| \approx 0,1 = 1\% \quad (5.28)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,5 \text{ V} - 3,64 \text{ V}}{3 \text{ V} - 3,5 \text{ V}} \right| \approx 0,28 = 28\% \quad (5.29)$$

Außerdem braucht der Regler länger, bis er ein näherungsweise stationäres Verhalten aufweist. Der Regler erreicht erst ungefähr 45 Sekunden nach dem Führungsgrößensprung ein stationäres Verhalten. Wie auch der vorherige Agent, hat das Stellsignal des Agenten aus Abbildung 5.13 eine hochfrequente Komponente

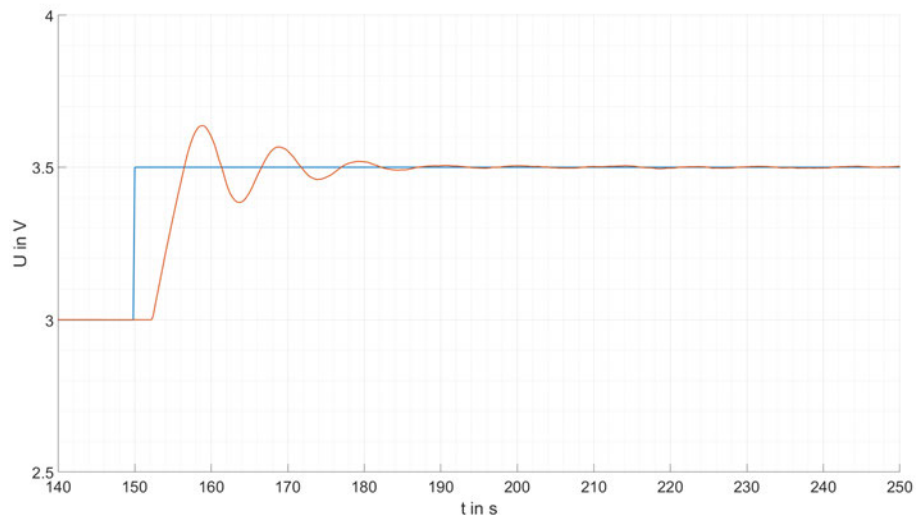


Abbildung 5.13: Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau) bei Agent mit Regelfehler und Regelgröße als Zustand

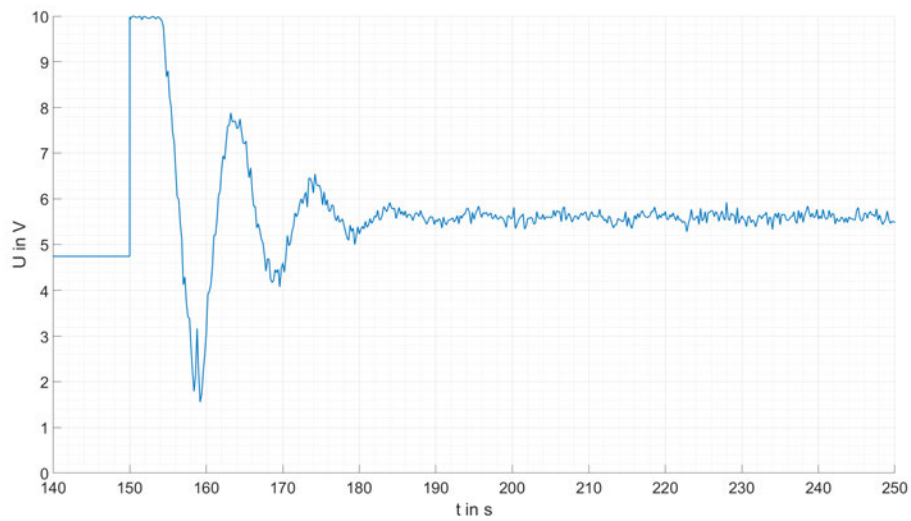


Abbildung 5.14: Verlauf der Stellgröße auf einen Führungsgrößensprung bei Agent mit Regelfehler und Regelgröße als Zustand

Auswahl des Zustandsvektors

Wie bereits in den vorherigen Abschnitten erläutert, ist der Agent mit dem Zustandsvektor $s_t = (e_t)$ ungeeignet, um die Strecke zu regeln. Im Vergleich sind die Werte der maximalen Regelabweichung $e_{\max} = 16,9\%$ sowie der maximal bleibenden Regelabweichung $e_{\max,\infty} = 3\%$ weit über den Ergebnissen der anderen beiden Zustandsvektoren. Außerdem braucht der erste Agent mit einer Dauer von ungefähr 60 Sekunden am längsten, bis das Stellsignal eingeschwungen ist.

Tabelle 5.1: Ergebnisse der untersuchten Agenten mit verschiedenen Zustandsvektoren

Zustandsvektor	Einschwingzeit	e_{\max}	$e_{\max,\infty}$
$s_t = (e_t)$	60 s	33,8%	20%
$s_t = \begin{pmatrix} e_t \\ w_t \end{pmatrix}$	30 s	18%	1,86%
$s_t = \begin{pmatrix} e_t \\ y_t \end{pmatrix}$	45 s	28%	1%

Der Vergleich der letzten beiden Zustandsvektoren aus 5.1 zeigt, dass der Agent, der die Führungsgröße im Zustandsvektor hat, eine niedrigere maximale Überschwingweite e_{\max} sowie eine geringere Einschwingzeit aufweist. Dahingegen hat der Agent, der die Regelgröße im Zustandsvektor hat, eine niedrigere maximal bleibende Regelabweichung. Für die spätere Implementierung am Versuchsaufbau werden beide Varianten untersucht.

5.2.3 Aktionsvektor

Die Beantwortung der ersten Fragestellung ist für diesen Versuchsaufbau relativ simpel. Theoretisch kann der Agent an der Anlage zwei Kenngrößen regeln. Zum einen kann der Agent über ein Stellsignal die Drehzahl des Lüfters ändern, um den Volumenstrom zu regeln. Zum anderen kann der Agent über ein Stellsignal die elektrische Leistung des Heizelements ändern und damit die Lufttemperatur beeinflussen. Der Volumenstrom wird konstant gehalten, da die Temperaturregelung nur über das Heizelement erfolgen soll. Daraus folgt, dass der Aktionsvektor nur aus einem Element besteht. Aus dem 0 V - 10 V Stellsignal des Heizelements. Damit der Agent alle Werte zwischen 0 V und 10 V als Stellsignal wählen kann, wird ein kontinuierlicher Aktionsraum gewählt.

5.2.4 Belohnungsfunktion

Die Belohnungsfunktion und die daraus bestimmte Belohnung sind, wie in Abschnitt 3 bereits erläutert, die Basis, auf der die meisten Reinforcement Learning Algorithmen aufbauen. Deswegen ist die Entwicklung einer geeigneten Belohnungsfunktion so entscheidend für den Erfolg des Lernprozesses. Das Belohnungssignal kann unterschiedlich aufgebaut sein. [7] nennt drei Arten von Belohnungsfunktionen:

- Kontinuierliche Belohnungsfunktion
- Diskrete Belohnungsfunktion
- Gemischte Belohnungsfunktion

Diese sind nach [7] wie folgt aufgebaut. Kontinuierliche Belohnungsfunktionen geben dem Agenten durchgehend eine Belohnung $r_t \neq 0$, die vom Zustand des Systems abhängt. Diskrete Belohnungsfunktionen unterscheiden sich dadurch, dass $r_t = 0$ gilt, außer für bestimmte Ereignisse, die den Agenten belohnen ($r_t > 0$) oder bestrafen ($r_t < 0$) sollen. Die letzte Art sind gemischte Belohnungsfunktionen. Hier setzt sich die Belohnungsfunktion aus diskreten und kontinuierlichen Belohnungen zusammen. In dieser Arbeit wird eine kontinuierliche Belohnungsfunktion $R(e_t) = -R_1(e_t) + R_2(e_t)$ implementiert, die aus zwei Komponenten zusammengesetzt wird.

Kontinuierliche Strafe

Der erste Teil der Belohnungsfunktion ist eine kontinuierliche Strafe, die wie folgt bestimmt wird:

$$R_1(e_t) = \beta \cdot |e_t| \quad (5.30)$$

Die Strafe ist das Produkt des Betrags der Regelabweichung mit einem Gewichtungsfaktor $\beta > 0$, der frei gewählt werden kann. Die Strafe ändert sich proportional zur Regelabweichung. Die Idee ist, dass der Agent durch die Strafe dazu verleitet wird, die Regelgröße in Richtung Sollwert zu bewegen.

Ereignisabhängige kontinuierliche Belohnung

Der zweite Teil der Belohnungsfunktion ist eine ereignisabhängige kontinuierliche Belohnung, die wie folgt aufgebaut ist:

$$R_2(e_t) = \epsilon \cdot \left(1 - \frac{|e_t|}{0,05}\right) \cdot I(e_t) \quad (5.31)$$

mit

$$I(e_t) = \begin{cases} 1 & \text{falls } e_t \leq 0,05 \\ 0 & \text{sonst} \end{cases} \quad (5.32)$$

Wie bei Gleichung 5.30 wird auch hier ein Gewichtungsfaktor ϵ eingeführt, mit dem das Verhältnis der zwei Belohnungen an der Gesamtbelohnung angepasst werden kann. $R_2 > 0$ tritt nur auf, wenn der Betrag der Regelabweichung kleiner als 0,05 ist. In dem Bereich $\pm 0,05$ um den Sollwert nimmt die Belohnung proportional zu. Abbildung 5.15 zeigt den Verlauf der Belohnung bei einem Regelfehler zwischen null und eins. Es wurde $\epsilon = \beta = 10$ gewählt. Man erkennt, dass ungefähr ab einem Regelfehler $|e_t| \leq 0,48$ die Gesamtbelohnung $R(e_t)$ einen Wert über 0 annimmt. Ob diese Belohnungsfunktion ein gutes Ergebnis beim Training erzeugt, werden die nächsten Versuche zeigen.

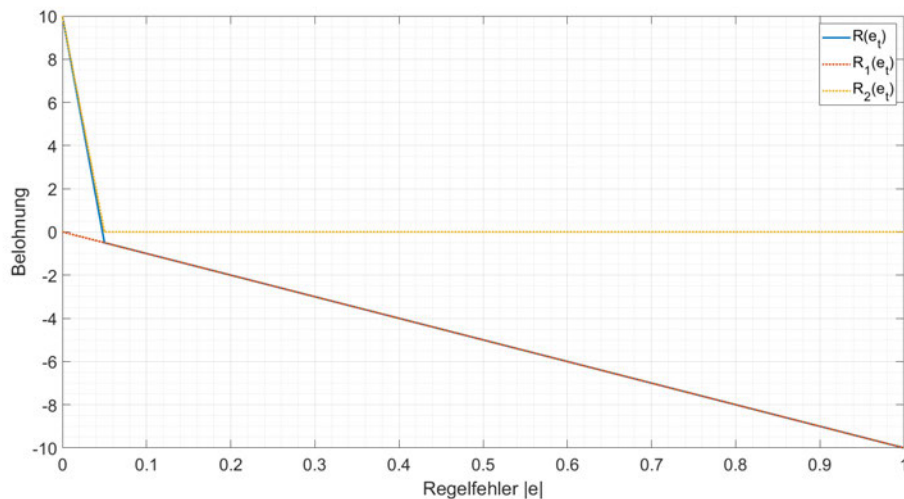


Abbildung 5.15: Verlauf der Belohnungsfunktion (blau) und ihrer Bestandteile R_1 (rot) und R_2 (gelb) in Abhängigkeit vom Betrag des Regelfehlers

5 Training des Soft Actor-Critic Agenten

Der in Abschnitt 5.2.2 trainierte Agent verwendet im Lernprozess die in diesem Abschnitt beschriebene Belohnungsfunktion mit $\epsilon = \beta = 10$. Wie die Verläufe der Stell- und Führungsgröße auf einen Sollwertsprung aus Abbildung 5.16 und Abbildung 5.17 zeigen, ist die entwickelte Belohnungsfunktion für das Training geeignet.

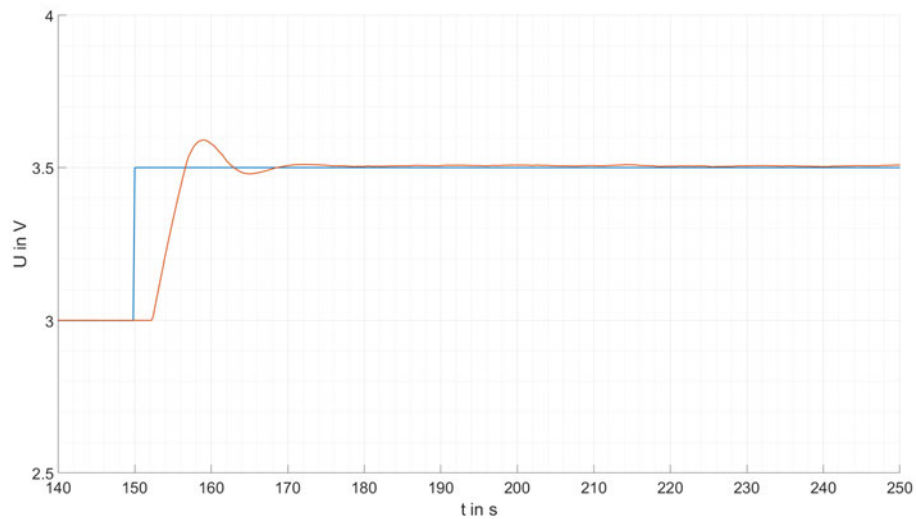


Abbildung 5.16: Verlauf der Regelgröße (gelb) bei einem Führungsgrößensprung (blau)

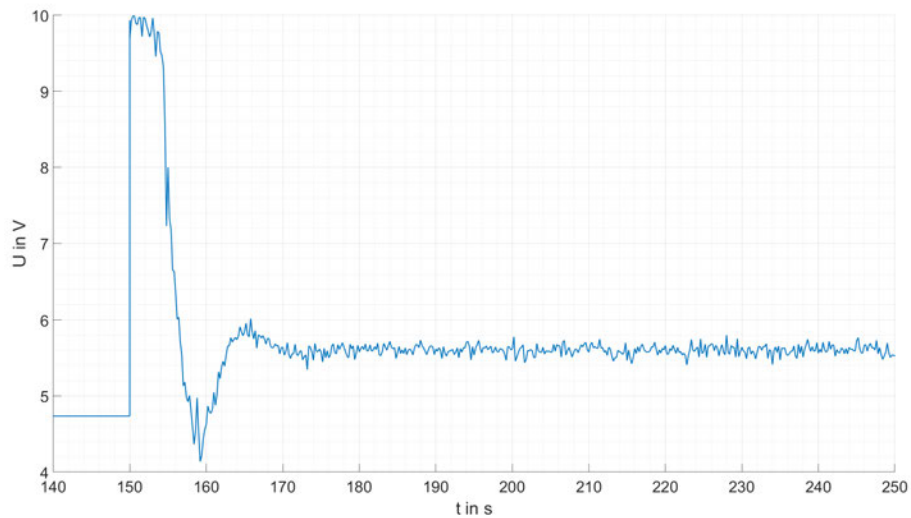


Abbildung 5.17: Verlauf der Stellgröße bei einem Führungsgrößensprung

5.2.5 Wahl der Hyperparameter

Bei Implementation des Soft Actor-Critic Algorithmus besteht die Möglichkeit, eine Vielzahl von Hyperparametern einzustellen. Eine vollständige Liste der Hyperparameter des Agenten findet man bei [8]. Die Optimierung der möglichen Hyperparameter wäre sehr zeitaufwendig. Deswegen wird im Folgenden beispielhaft untersucht, wie sich die Größe des Mini-Batch vom Agenten auf den Lernprozess auswirkt. Für die restlichen Hyperparameter werden die Standardwerte gewählt beziehungsweise die von [11] in Beispiel „Create SAC Agent from Actor and Critics“ vorgeschlagenen Werte. Die Größe des Mini-Batch bestimmt die Anzahl der Erfahrungswerte, die zufällig aus dem Erfahrungspuffer gewählt werden, mit denen dann die Kritiker und die Strategie aktualisiert werden.

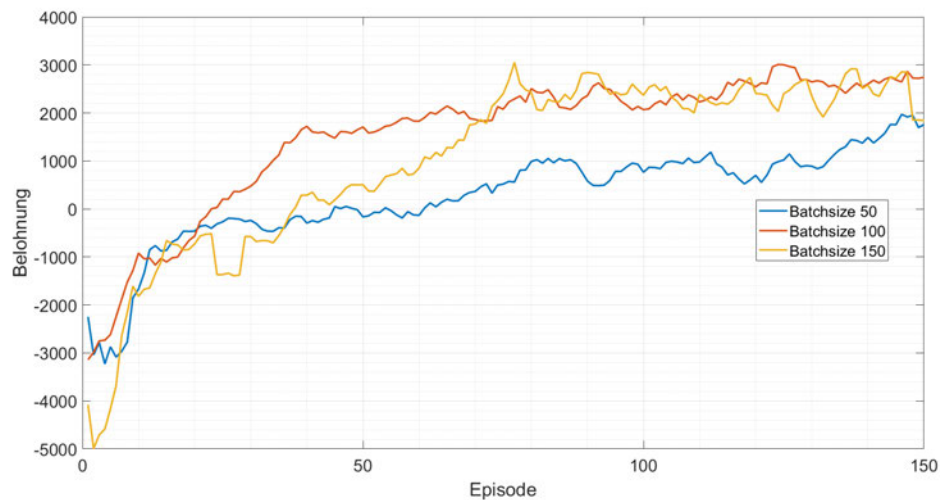


Abbildung 5.18: Mittlere Belohnung während des Training bei unterschiedlichen Mini-Batch-Sizes

Für den Versuch wurde das Training für einen Mini-Batch jeweils sechs mal durchgeführt. Aus den Ergebnissen wurde dann die mittlere Belohnung gebildet. 5.18 zeigt, dass die Agenten bei einem Mini-Batch von 100 und 150 ab ungefähr Episode 75 einen ähnlichen Verlauf haben. Der Verlauf bei einer Mini-Batch-Size von 50 liegt durchgehend darunter. Daraus folgt, es reicht, wenn ein Mini-Batch der Größe von 100 gewählt wird. In diesem Fall ist es besser, den niedrigeren der beiden Werte zu nehmen, weil laut [8] eine steigende Mini-Batch-Size die benötigte Rechenleistung erhöht.

5.3 Training am Versuchsaufbau

Mit dem Wissen aus Abschnitt 5.2 werden nun zwei Agenten für den Vergleich mit den Standardreglern aus Abschnitt 4 trainiert. Der erste Agent wird ausschließlich an der Anlage trainiert. Der zweite Agent wird zuerst am Simulink Modell trainiert und danach am Versuchsaufbau. Das Ziel ist es zu prüfen, ob es einen Vorteil gibt, den Agenten zuerst an einem mathematischen Modell zu trainieren. Das Teilen des Trainings in zwei Schritte hat in der Theorie mehrere Vorteile gegenüber dem alleinigen Training am Versuchsaufbau. Im Vergleich

zum Lernen an der Anlage benötigt der Agent für das Training am mathematischen Modell nur einen Bruchteil der Zeit, um ein ähnliches Ergebnis zu erreichen. Am verwendeten Rechner werden für zwei gleichzeitig laufende Trainingsprozesse mit jeweils 300 Episoden einer Länge von 90 Sekunden ungefähr 53 Minuten vergehen. Dasselbe Ergebnis würde am Versuchsaufbau 35 Stunden dauern. Unter Berücksichtigung, dass am Versuchsaufbau nur ein Trainingsprozess zur Zeit laufen kann.

Außerdem vergehen vor jeder Episode 120 Sekunden, um die Strecke in ihren Ausgangspunkt zu regeln. Es muss außerdem berücksichtigt werden, dass nicht jeder Trainingsprozess erfolgreich abläuft. Abbildung 5.19 zeigt die mittlere Belohnung während des Trainings. Die gezeigten Graphen stammen von den Ergebnissen aus Abschnitt 5.2.2, die um zwei weitere Trainingsdurchläufe unter denselben Bedingungen erweitert wurden. Abbildung 5.19 zeigt, dass die mittlere Belohnung von zwei, türkis und grün, der sechs trainierten Agenten weit unter den restlichen Ergebnissen liegt. Bei den Ergebnissen mit einer hohen Belohnung, orange, blau und gelb, besteht die Möglichkeit, dass das erlernte Regelverhalten nicht den Anforderungen erfüllt und das erlernte Verhalten dem aus Abbildung 5.13 ähnelt. Die lange Trainingszeit und die Unsicherheit bei der Qualität des gelernten Regelverhaltens bestätigt, dass in der Theorie das geteilte Training dem alleinigen Training an der Anlage vorzuziehen ist.

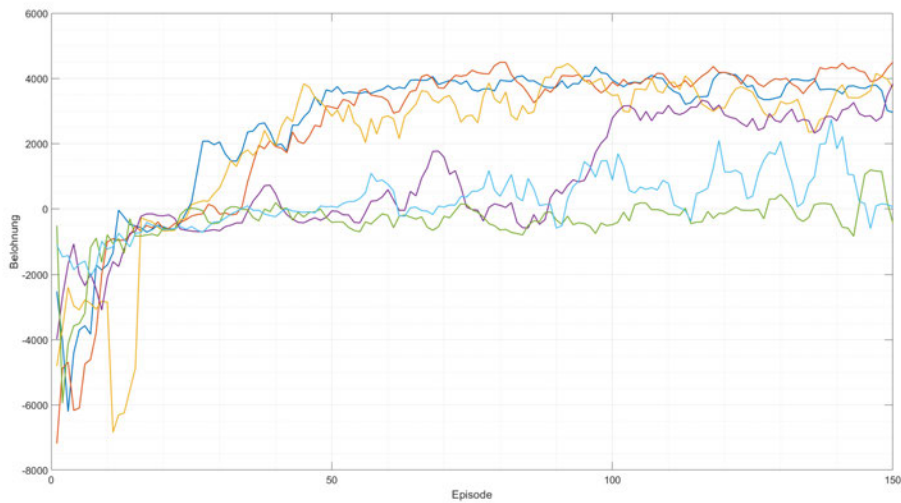


Abbildung 5.19: Mittlere Belohnung während des Training von sechs Agenten unter den gleichen Bedingungen

5.3.1 Lernen am mathematischen Modell und Anlage

Beim Lernen am mathematischen Modell und Anlage wird der Agent zuerst am mathematischen Modell der Regelstrecke trainiert. Das Wissen aus dem ersten Lernprozess wird dann eingesetzt, um das Lernen am Versuchsaufbau zu beschleunigen. Der Grundgedanke ist, dass der Agent im ersten Schritt am mathematischen Modell das gewünschte Regelverhalten lernt. Im zweiten Schritt am Versuchsaufbau lernt der Agent dann, wie er seine Strategie beziehungsweise sein Verhalten anpassen muss, um eventuelle Ungenauigkeiten des Modells auszugleichen. Hierfür wird der Agent für 300 Episoden einer Länge von jeweils 90 Sekunden trainiert. Anschließend wird der Agent für weitere 150 Episoden einer Länge von jeweils 90 Sekunden am Versuchsaufbau trainiert.

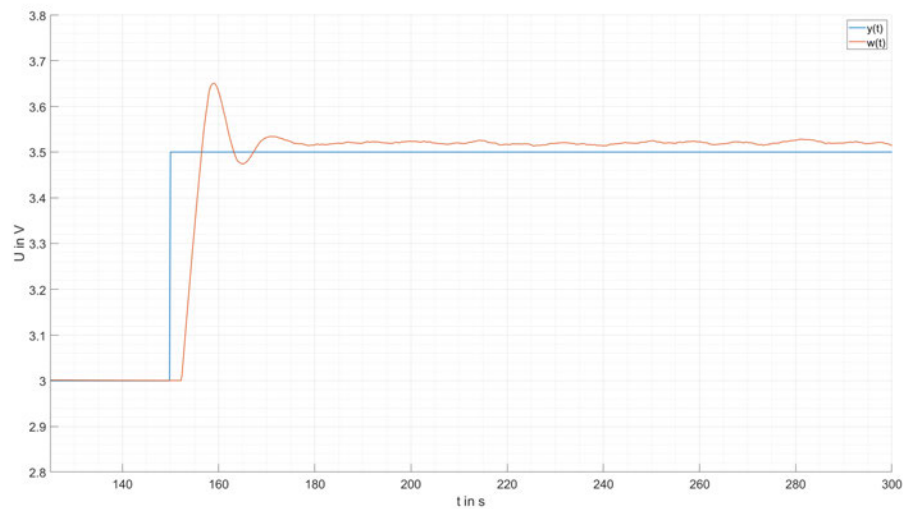


Abbildung 5.20: Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau)

Abbildung 5.20 zeigt den resultierenden Verlauf der Regelgröße auf eine Sollwertsprung von 3 V auf 3,5 V, wenn der Agent das mathematische Modell regelt. Die Untersuchung der Sprungantwort ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,5 \text{ V} - 3,528 \text{ V}}{3 \text{ V} - 3,5 \text{ V}} \right| \approx 0,056 = 5,6\% \quad (5.33)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,5 \text{ V} - 3,65 \text{ V}}{3 \text{ V} - 3,5 \text{ V}} \right| \approx 0,3 = 30\% \quad (5.34)$$

Die Regelgröße ist nach ca. 30 Sekunden eingeschwungen. Ein Vergleich mit Tabelle 5.1 zeigt, dass die Ergebnisse der Regelabweichungen und Einschwingzeit in einem ähnlichen Bereich liegen.

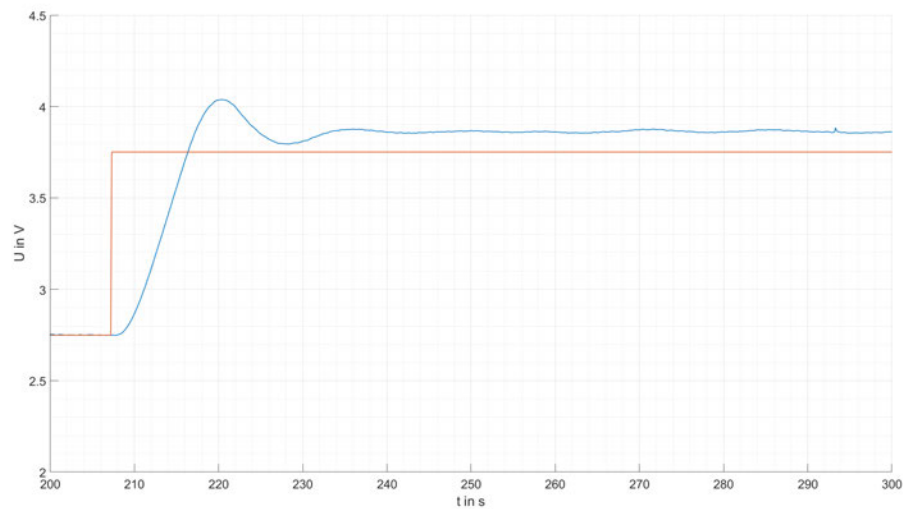


Abbildung 5.21: Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau) am Versuchsaufbau, bei der Regelung des am mathematischen Modell trainierten Agenten

Abbildung 5.21 zeigt die Sprungantwort des Regelkreis auf einen 1 V Sollwertsprung von 2,75 V auf 3,75 V. Die Untersuchung der Sprungantwort ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,75 \text{ V} - 3,875 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,125 = 12,5\% \quad (5.35)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,5 \text{ V} - 4,04 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,29 = 29\% \quad (5.36)$$

Die Regelgröße ist nach ungefähr 40 Sekunden eingeschwungen. Ein Vergleich der beiden Sprungantworten zeigt, dass der Verlauf der Regelgröße ähnlich ist und nur um einen Offset nach oben verschoben ist. Der Offset kann dadurch entstanden sein, dass die Raumtemperatur, die im Training als Basis angenommen wurde, nicht exakt mit der tatsächlichen Raumtemperatur während des Versuchs übereinstimmt. In der Theorie ist das eine Modellungenauigkeit, die durch das weitere Training ausgeglichen werden sollte.

Der im Folgenden eingesetzte Agent ist der beste aus drei trainierten Agenten, die jeweils als Grundlage das erlernte Wissen des oben vorgestellten Agenten haben.

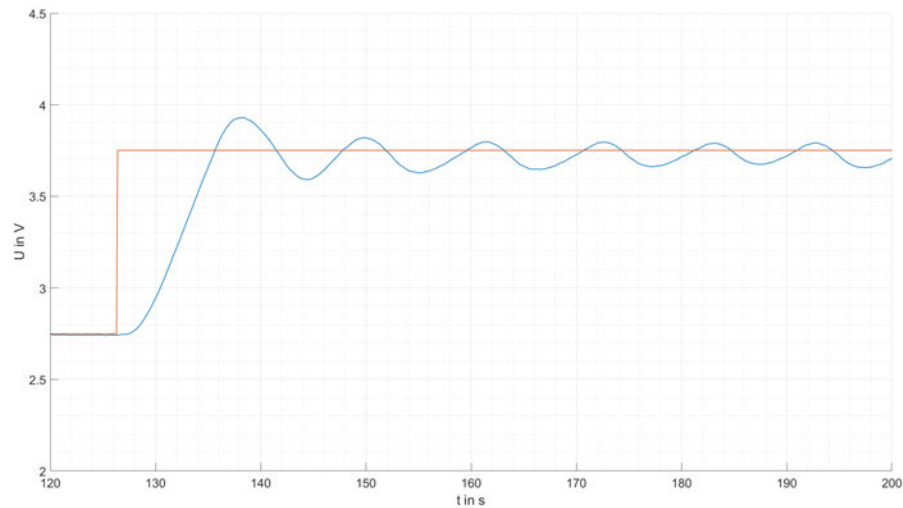


Abbildung 5.22: Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau) am Versuchsaufbau, bei der Regelung des weiter trainierten Agenten

Abbildung 5.22 zeigt die Sprungantwort des Regelkreises auf einen 1 V Führungsgrößensprung von 2,75 V auf 3,75 V. Die Untersuchung der Sprungantwort ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,75 \text{ V} - 3,66 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,09 = 9\% \quad (5.37)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,75 \text{ V} - 3,93 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,18 = 18\% \quad (5.38)$$

Die Regelgröße ist nach ungefähr 35 Sekunden eingeschwungen. Die Theorie, dass durch die Fortsetzung des Trainings Ungenauigkeiten im mathematischen Modell ausgeglichen werden

können, wurde bestätigt. Aus Abbildung 5.22 erkennt man, dass die Regelgröße um den Sollwert schwingt und nicht - wie vorher - weit über dem Sollwert liegt. Der Vergleich mit Abbildung 5.21 zeigt auch, dass die maximale Überschwingweite sowie die maximale bleibende Regelabweichung geringer geworden sind. Der Nachteil ist, dass durch die Fortsetzung des Training eine bleibende Schwingung in der Regelgröße und Stellgröße bleibt, auch nachdem der Regelkreis eingeschungen ist. Es tritt ein ähnliches Regelverhalten auf wie schon in Abschnitt 5.2.2 beim trainierten Agenten mit dem Regelfehler und der Regelgröße im Zustandsvektor.

5.3.2 Lernen an der Anlage

Für das Lernen an der Anlage wird der Agent für 250 Episoden direkt an der Anlage trainiert. Die gesamte Trainingszeit beläuft sich auf ungefähr 14 Stunden und 35 Minuten. Der trainierte Agent wird in das Simulink-Modell als Regler implementiert. Für den Vergleich mit dem im vorherigen Abschnitt untersuchten Agenten wird die Sprungantwort des Regelkreises auf einen 1 V Führungsgrößensprung untersucht.

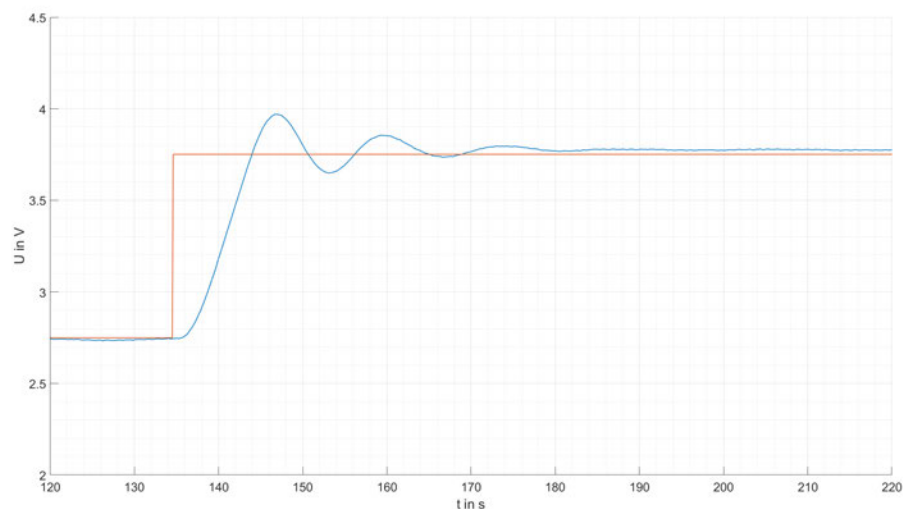


Abbildung 5.23: Verlauf der Regelgröße (orange) auf einen Führungsgrößensprung (blau) am Versuchsaufbau, bei der Regelung des Versuchsaufbau trainierten Agenten

Abbildung 5.23 zeigt die Sprungantwort des Regelkreises auf einen 1 V Führungsgrößensprung von 2,75 V auf 3,75 V. Die Untersuchung der Sprungantwort ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,75 \text{ V} - 3,78 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,03 = 3\% \quad (5.39)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,75 \text{ V} - 3,969 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,219 = 21,9\% \quad (5.40)$$

Der Regelkreis ist nach ungefähr 50 Sekunden eingeschwungen. Ein Vergleich mit dem Regelverhalten des Agenten aus Abschnitt 5.3.1 zeigt, dass das gelernte Regelverhalten besser ist. Es kommt zu keiner bleibenden Schwingung der Regelgröße. Auch die maximale bleibende Regelabweichung beträgt mit 3% nur ein Drittel der maximale bleibende Regelabweichung des anderen Agenten. Dafür ist sie nahezu konstant über dem gewünschten Sollwert. Ein ähnliches Ergebnis wie auch schon in Abschnitt 5.2.2 bei den beiden Agenten mit erweitertem Zustandsvektor, auch wenn hier $e_{\max, \infty}$ größer ist. Eine mögliche Ursache für das schlechtere Ergebnis könnte die vorhergesagte Regelgröße des Smith-Prädiktors sein. Für die Vorhersage wird die in Abschnitt 2.3.2 bestimmte Übertragungsfunktion verwendet. Diese ist wiederum nur eine gute Näherung der realen Regelstrecke. Dadurch könnte eine Abweichung bei der Vorhersage entstehen.

6 Vergleich des trainierten Agenten mit den dimensionierten PI-Reglern

In diesem Kapitel wird untersucht, wie das Regelverhalten des trainierten Soft Actor-Critic Agenten im Vergleich zu den ausgelegten PI-Reglern abschneidet. Für den Vergleich werden zuerst die Sprungantworten des geschlossenen Regelkreises auf einen Führungsgrößensprung von 1 V untersucht. Die Standardregler werden jeweils im Arbeitspunkt betrieben. Der zweite Teil besteht aus dem Vergleich der Sprungantworten des geschlossenen Regelkreises für einen Störgrößensprung von $-0,5$ V am Ausgang der Regelstrecke. Für die PI-Regler werden die in Abschnitt 4 bestimmten Werte für die Verstärkung und Zeitkonstante verwendet. Es wird der in Abschnitt 5.3.2 vorgestellte Agent für die Untersuchungen eingesetzt.

6.1 Messung der Führungssprungantworten

6.1.1 Reglerentwurf Ziegler-Nichols-Einstellregeln

Für die Untersuchung der Führungssprungantwort des Regelkreises wird ein PI-Regler mit einer Verstärkung von

$$K_R = 16,63 \quad (6.1)$$

und einer Zeitkonstante

$$T_u = 7,2 \text{ s} \quad (6.2)$$

im Simulink-Modell implementiert. Abbildung 6.1 zeigt den Verlauf der Regelgröße für einen Führungsgrößensprung bei $t = 200,4$ s. Die Untersuchung von Abbildung 6.1 ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,75 \text{ V} - 3,758 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,008 = 0,8\% \quad (6.3)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,75 \text{ V} - 4,39 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,64 = 64\% \quad (6.4)$$

Die Einschwingzeit, die benötigt wird bis die bleibende Regelabweichung durchgehend kleiner oder gleich $e_{\max, \infty}$ ist, beträgt ungefähr 85 Sekunden.

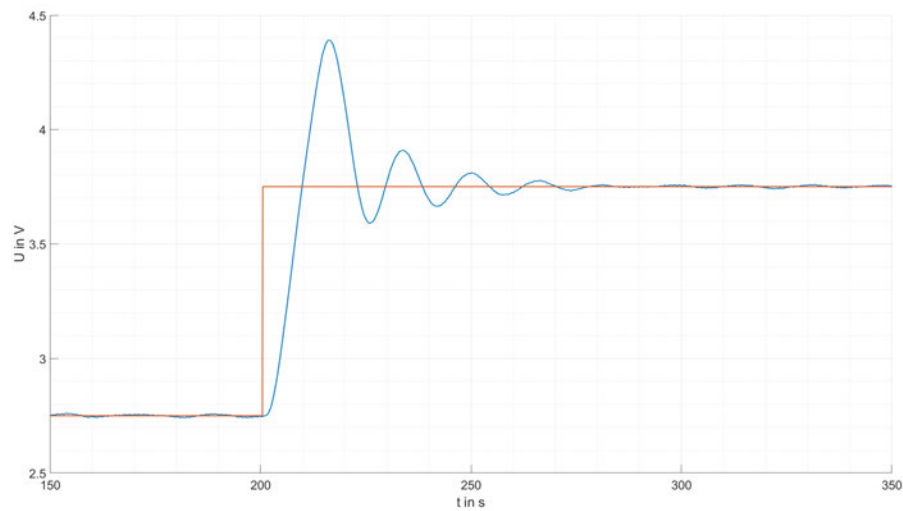


Abbildung 6.1: Sprungantwort (blau) des Regelkreises mit dem nach Ziegler-Nichols ausgelegten Regler auf einen Führungsgrößensprung (orange)

6.1.2 Reglerentwurf mittels Polkompensation

Für die Untersuchung der Führungssprungantwort des Regelkreises wird ein PI-Regler mit einer Verstärkung von

$$K_R = K_P = 9,53 \quad (6.5)$$

und einer Zeitkonstante

$$T_u = 23,06 \text{ s} \quad (6.6)$$

6 Vergleich des trainierten Agenten mit den dimensionierten PI-Reglern

im Simulink-Modell implementiert. Abbildung 6.2 zeigt den Verlauf der Regelgröße für einen Führungsgrößensprung bei $t = 102,6 \text{ s}$. Die Untersuchung von Abbildung 6.2 ergibt eine maximale bleibende Regelabweichung von

$$e_{\max,\infty} = \left| \frac{3,75 \text{ V} - 3,756 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,01 = 1\% \quad (6.7)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,75 \text{ V} - 3,9 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,15 = 15\% \quad (6.8)$$

Die Einschwingzeit, die benötigt wird bis die bleibende Regelabweichung durchgehend kleiner oder gleich $e_{\max,\infty}$ ist, beträgt ungefähr 35 Sekunden.

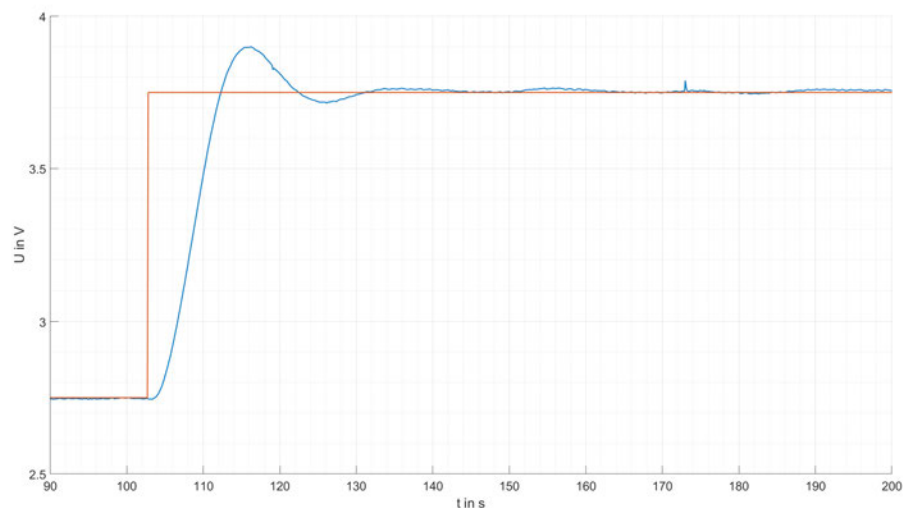


Abbildung 6.2: Sprungantwort (blau) des Regelkreises mit dem mittels Polkompensation ausgelegten Regler auf einen Führungsgrößensprung (orange)

6.1.3 Reglerentwurf mittels Tune-Funktion eines Industriereglers

Für die Untersuchung der Führungssprungantwort des Regelkreises wird ein PI-Regler mit einer Verstärkung von

$$K_R = K_P = 4,6 \quad (6.9)$$

und einer Zeitkonstante

$$T_u = 5,628 \text{ s} \quad (6.10)$$

im Simulink-Modell implementiert. Abbildung 6.3 zeigt den Verlauf der Regelgröße für einen Führungsgrößensprung bei $t = 134 \text{ s}$. Die Untersuchung von Abbildung 6.3 ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,75 \text{ V} - 3,744 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,006 = 0,6\% \quad (6.11)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,75 \text{ V} - 4,18 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,43 = 43\% \quad (6.12)$$

Die Einschwingzeit, die benötigt wird bis die bleibende Regelabweichung durchgehend kleiner oder gleich $e_{\max, \infty}$ ist, beträgt ungefähr 70 Sekunden.

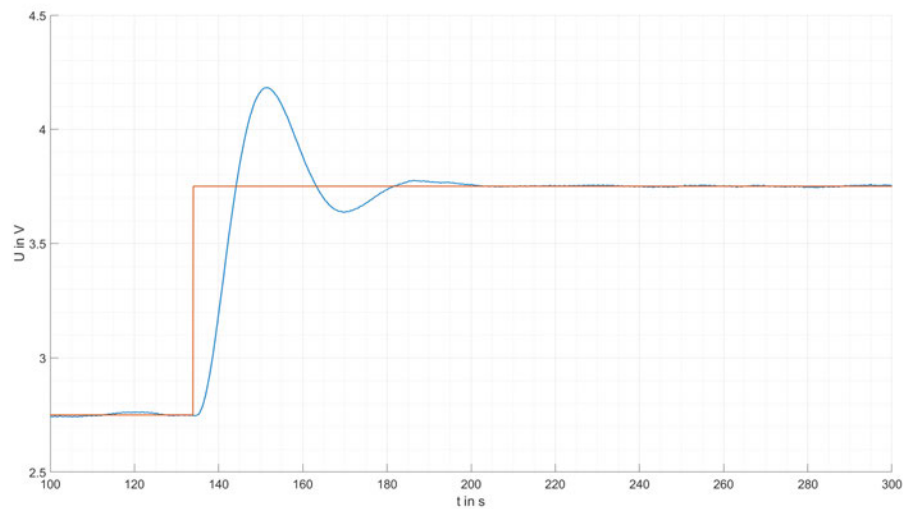


Abbildung 6.3: Sprungantwort (blau) des Regelkreises mit dem mittels Tune-Funktion ausgelegten Regler auf einen Störgrößensprung (orange)

6.1.4 Regelung mittels Soft-Actor-Critic Agent

Für die Untersuchung der Führungssprungantwort des Regelkreises wird der Agent aus Abschnitt 5.3.2 im Simulink-Modell implementiert. Abbildung 6.4 zeigt den Verlauf der Regelgröße für einen Störgrößensprung bei $t = 134,5$ s. Die Untersuchung von Abbildung 6.4 ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,75 \text{ V} - 3,78 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,03 = 3\% \quad (6.13)$$

des Sollwertsprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,75 \text{ V} - 3,969 \text{ V}}{2,75 \text{ V} - 3,75 \text{ V}} \right| \approx 0,219 = 21,9\% \quad (6.14)$$

Die Einschwingzeit, die benötigt wird, bis die bleibende Regelabweichung durchgehend kleiner oder gleich $e_{\max, \infty}$ ist, beträgt ungefähr 50 Sekunden.

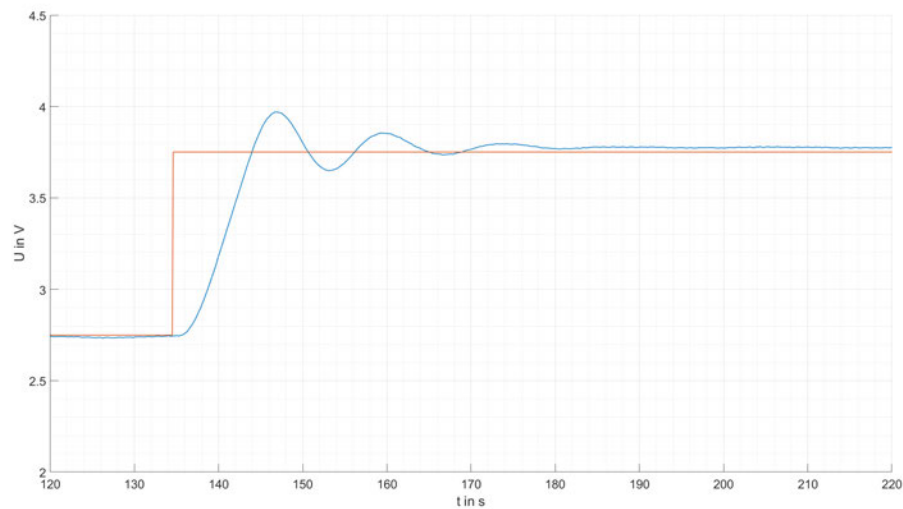


Abbildung 6.4: Sprungantwort (blau) des Regelkreises mit dem Agenten auf einen Störgrößen-sprung (orange)

6.1.5 Auswertung

Die Messungen der Führungsgrößen-sprungantworten des Regelkreises unter Einsatz der verschiedenen Regler hat die in Tabelle 6.1 gelisteten Ergebnisse erzeugt. Die Ergebnisse der mit Standardmethoden der Regelungstechnik dimensionierten Regler zeigen, dass der Regler mit Polkompensation die besten Ergebnisse bei einem Führungsgrößen-sprung liefert. Die maximale bleibenden Regelabweichungen, die bei den drei Reglern entstehen, liegen sehr dicht beieinander und können als gleich gut angenommen werden. Die leichten Abweichungen lassen sich durch Messungenauigkeiten sowie Materialtoleranzen bei den verwendeten mechanischen und elektrischen Komponenten erklären.

Die gemessene maximale Regelabweichung e_{\max} ist sowohl bei dem über Ziegler-Nichols-Einstellregeln als auch bei dem über die Tune-Funktion eingestellten Regler vergleichsweise groß. Das ist aber zu erwarten, denn bei beiden Verfahren handelt es sich um eine Art der Ziegler-Nichols Einstellregeln, welche wiederum zu den empirischen Einstellregeln gehören.

Bei beiden Verfahren liegt keine mathematische Beschreibung der Regelstrecke vor, sondern die Regelstrecke wird als PT_1 -Glied mit Totzeit genähert und die Streckenparameter werden aus Messungen geschätzt ([15], S. 224-225). Die gemessene maximale Regelabweichung e_{\max} bei der Regelstrecke mit einem PI-Regler, der über Polkompensation ausgelegt wurde, liefert mit 15% das beste Ergebnis. Das Ergebnis ist nicht überraschend, denn im Gegensatz zu den anderen beiden Verfahren wird hier für die Dimensionierung des Reglers eine mathematische Beschreibung der Regelstrecke verwendet. Der Regler überschreitet die maximale Regelabweichung um $e_{\max} = 5\%$, für die er dimensioniert ist. Das hat mehrere Ursachen. Zum einen wird bei der Auslegung des Reglers über Polkompensation nicht berücksichtigt, dass die Stellgröße auf einen Bereich von 0 V bis 10 V begrenzt ist. Zum anderen ist es praktisch nicht umsetzbar, einen Pol der Regelstrecke durch eine Nullstelle beim Regler vollständig zu kompensieren. Das liegt daran, dass bei der mathematischen Auslegung des Reglers nur die genäherten Werte für die Parameter der Regelstrecke verwendet werden ([14], S. 242). In der Regel ist es nicht üblich, dass man die exakten Parameter der Regelstrecke kennt beziehungsweise die Parameter können sich im Laufe des Betriebs innerhalb bestimmter Grenzen verändern.

Vergleicht man nun die Ergebnisse des trainierten Agenten mit den Ergebnissen der drei PI-Regler, fällt Folgendes auf: Man kann anhand der Kenngrößen nicht eindeutig sagen, ob der Agent besser oder schlechter ist als die PI-Regler. Die Einschwingzeit bei dem Regelkreis mit Agenten beträgt ungefähr 85 Sekunden. Das ist im gleichen Bereich wie bei den Regelkreisen mit Tune-Funktion oder Ziegler-Nichols ausgelegten Reglern. Die Einschwingzeit ist aber zwei mal so groß wie bei dem Regelkreis mit Polkompensation dimensionierten Reglern. Ob die Einschwingzeit ein Ausschlusskriterium ist, hängt davon ab, welche Prozesse die Lüftungsanlage mit Luft versorgt. Wird die Lüftungsanlage eingesetzt, um die Raumlufttemperatur in einem Büro zu regeln, kann eine längere Einschwingzeit eher toleriert werden, als wenn ein Labor versorgt wird, in dem durchgehend bestimmte klimatische Bedingungen herrschen müssen. Gleiches gilt für die maximale Überschwingweite. Der Vergleich der maximalen Überschwingweite ergibt, dass der Regelkreis mit Agent bei der maximalen Überschwingweite das zweitbeste Ergebnis hat. Betrachtet man nur e_{\max} , ist das Ergebnis des Agenten annähernd auf dem gleichen Niveau wie der Regler mit Polkompensation. Die maximale bleibende Regelabweichung beim Regelkreis mit Agent liegt bei 3%. Das ist mehr als drei mal so viel wie bei den Standard-PI-Reglern. Das eigentliche Ziel ist es, dass die Regelabweichung gegen 0 geht.

Schaut man sich den Verlauf der Regelgröße der drei PI-Regler an nachdem sie eingeschwen- gen sind, erkennt man, dass die Regelgröße nicht exakt den Sollwert erreicht. Sie schwingt aber sehr nah um den Sollwert. Das ist akzeptabel, wenn man berücksichtigt, dass es Messun- genauigkeiten gibt und die Temperatur auf dem Weg zum Ausgang der Strecke leicht durch Isolationsverluste verringert wird. Dahingegen ist die bleibende Regelabweichung bei dem Regelkreis mit Agenten nahezu konstant 3% über dem Sollwert. Der Agent ist also, wenn man die bleibende Regelabweichung betrachtet, schlechter als die Standard-PI-Regler. Betrachtet man die eingesetzte Belohnungsfunktion des Agenten aus Abschnitt 5.2.4, hat der Agent das Potential, ein ähnliches Verhalten zu erlernen. Das zeigen auch die simulierten Ergebnisse aus Abschnitt 5.2.2 , bei denen der Regelkreis mit Agent einen bleibenden Regelfehler von 1,86% hat. Eine Fortsetzung des Trainings könnte also ein besseres Endergebnis liefern. Das hätte aber den Nachteil, dass eine große Menge an Zeit für das Training investiert werden muss.

Betrachtet man alle Aspekte zusammen, kann man sagen, dass das Regelverhalten des Agenten für Führungsgrößensprünge in seinem aktuellen Zustand schlechter ist als das der PI-Regler.

Tabelle 6.1: Kenngrößen der Führungsgrößensprungantworten der Regelkreise

Auslegungsart	Führungsgrößensprung		
	$e_{\max, \infty}$	e_{\max}	Einschwingzeit
Ziegler-Nichols	0,8%	64%	85 s
Polkompenstaion	1%	15%	35 s
Tune-Funktion	0,6%	43%	70 s
Soft-Actor-Critic -Agent	3%	21,9%	85 s

6.2 Messung der Störsprungantworten

6.2.1 Reglerentwurf Ziegler-Nichols-Einstellregeln

Für die Untersuchung der Störsprungantwort des Regelkreises wird derselbe PI-Regler wie in Abschnitt 6.1.1 eingesetzt. Abbildung 6.5 zeigt den Verlauf der Regelgröße für einen

Störgrößensprung bei $t = 140,3 \text{ s}$. Die Untersuchung von Abbildung 6.5 ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,25 \text{ V} - 3,324 \text{ V}}{3,25 \text{ V} - 2,75 \text{ V}} \right| \approx 0,02 = 2\% \quad (6.15)$$

des Störgrößensprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,25 \text{ V} - 3,6 \text{ V}}{3,25 \text{ V} - 2,75 \text{ V}} \right| \approx 0,7 = 70\% \quad (6.16)$$

Die Einschwingzeit, die benötigt wird, bis die bleibende Regelabweichung durchgehend kleiner oder gleich $e_{\max, \infty}$ ist, beträgt ungefähr 75 Sekunden.

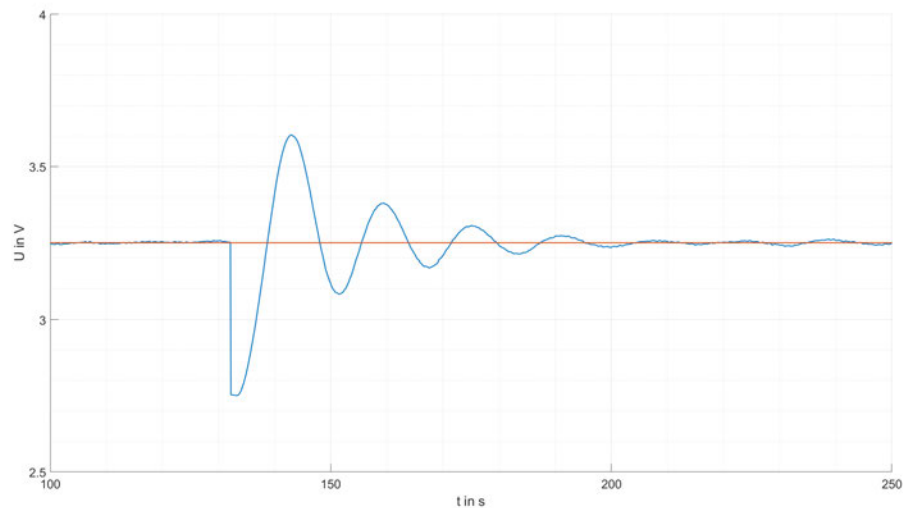


Abbildung 6.5: Sprungantwort (blau) des Regelkreises mit dem nach Ziegler-Nichols ausgelegten Regler auf einen Störgrößensprung (orange)

6.2.2 Reglerentwurf mittels Polkompensation

Für die Untersuchung der Störsprungantwort des Regelkreises wird derselbe PI-Regler wie in Abschnitt 6.1.2 eingesetzt. Abbildung 6.6 zeigt den Verlauf der Regelgröße für einen

Störgrößensprung bei $t = 130$ s. Die Untersuchung von Abbildung 6.6 ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,25 \text{ V} - 3,24 \text{ V}}{3,25 \text{ V} - 2,75 \text{ V}} \right| \approx 0,02 = 2\% \quad (6.17)$$

des Störgrößensprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,25 \text{ V} - 3,33 \text{ V}}{3,25 \text{ V} - 2,75 \text{ V}} \right| \approx 0,16 = 16,0\% \quad (6.18)$$

Die Einschwingzeit, die benötigt wird, bis die bleibende Regelabweichung durchgehend kleiner oder gleich $e_{\max, \infty}$ ist, beträgt ungefähr 35 Sekunden.

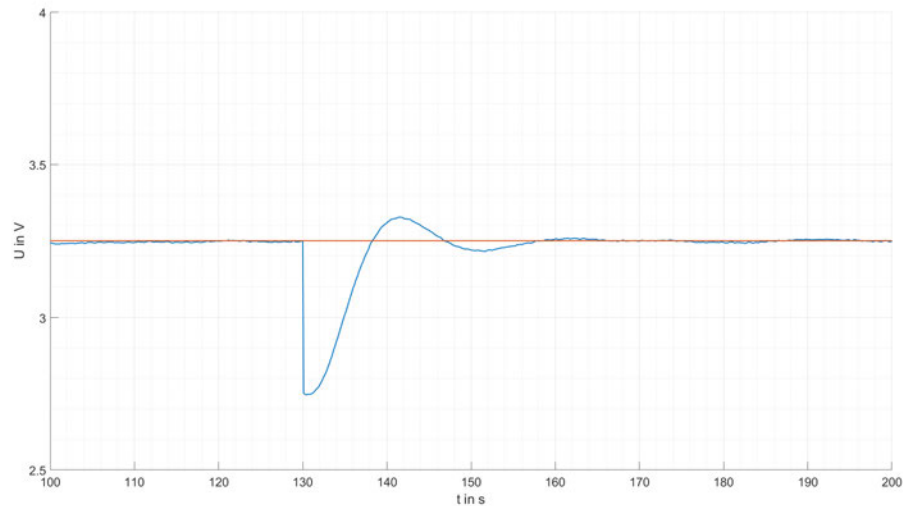


Abbildung 6.6: Sprungantwort (blau) des Regelkreises mit dem mittels Polkompensation ausgelegten Regler auf einen Störgrößensprung (orange)

6.2.3 Reglerentwurf mittels Tune-Funktion eines Industriereglers

Für die Untersuchung der Störsprungantwort des Regelkreises wird derselbe PI-Regler wie in Abschnitt 6.1.3 eingesetzt. Abbildung 6.7 zeigt den Verlauf der Regelgröße für einen

Störgrößensprung bei $t = 124,4$ s. Die Untersuchung von Abbildung 6.7 ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,25 \text{ V} - 3,24 \text{ V}}{3,25 \text{ V} - 2,75 \text{ V}} \right| \approx 0,02 = 2\% \quad (6.19)$$

des Störgrößensprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,25 \text{ V} - 3,47 \text{ V}}{3,25 \text{ V} - 2,75 \text{ V}} \right| \approx 0,44 = 44\% \quad (6.20)$$

Die Einschwingzeit, die benötigt wird, bis die bleibende Regelabweichung durchgehend kleiner oder gleich $e_{\max, \infty}$ ist, beträgt ungefähr 70 Sekunden.

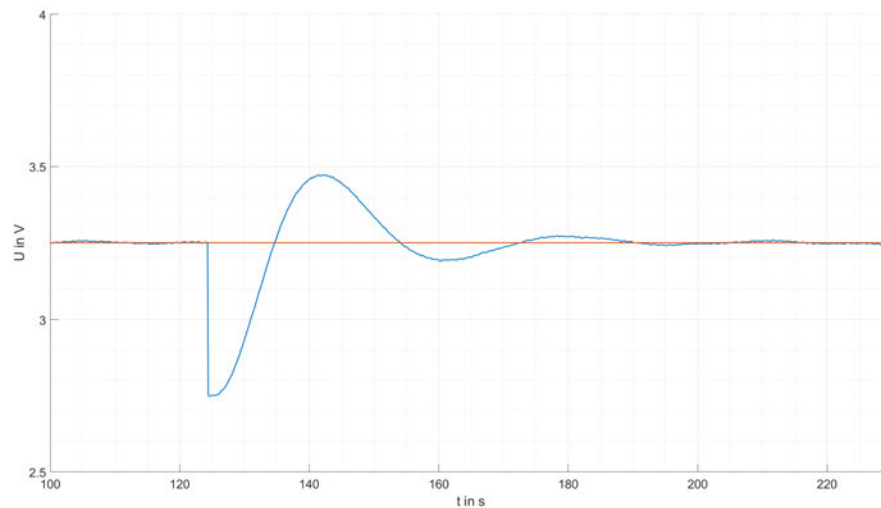


Abbildung 6.7: Sprungantwort (blau) des Regelkreises mit dem mittels Tune-Funktion ausgelegten Regler auf einen Störgrößensprung (orange)

6.2.4 Regelung mittels Soft Actor-Critic Agent

Für die Untersuchung der Stör sprungantwort des Regelkreises wird derselbe Agent wie in Abschnitt 6.1.4 eingesetzt. Abbildung 6.8 zeigt den Verlauf der Regelgröße für einen Störgrößensprung bei $t = 134$ s. Die Untersuchung von Abbildung 6.4 ergibt eine maximale bleibende Regelabweichung von

$$e_{\max, \infty} = \left| \frac{3,25 \text{ V} - 3,22 \text{ V}}{3,25 \text{ V} - 2,75 \text{ V}} \right| \approx 0,06 = 6\% \quad (6.21)$$

des Störgrößensprungs. Die maximale Überschwingweite e_{\max} beträgt:

$$e_{\max} = \left| \frac{3,25 \text{ V} - 3,43 \text{ V}}{3,25 \text{ V} - 2,75 \text{ V}} \right| \approx 0,36 = 36\% \quad (6.22)$$

Die Einschwingzeit, die benötigt wird, bis die bleibende Regelabweichung durchgehend kleiner oder gleich $e_{\max, \infty}$ ist, beträgt ungefähr 45 Sekunden.

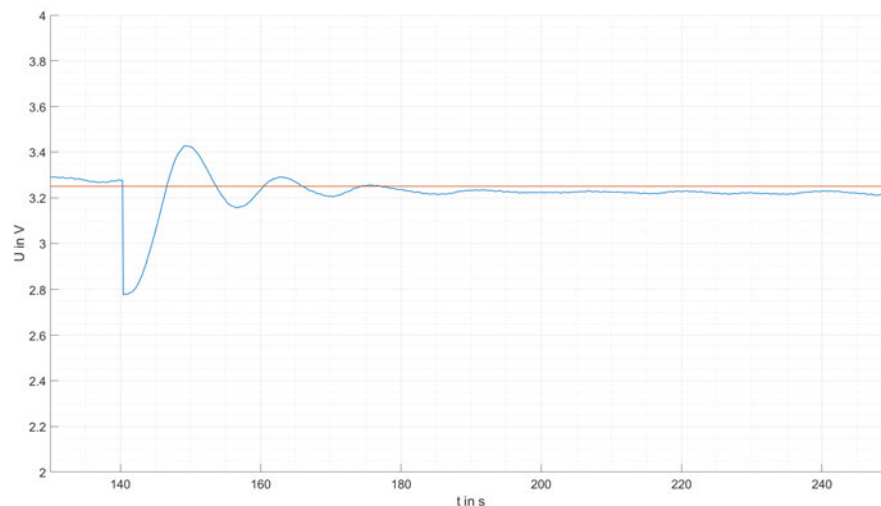


Abbildung 6.8: Sprungantwort (blau) des Regelkreises mit dem Agenten auf einen Störgrößensprung (orange)

6.2.5 Auswertung

Tabelle 6.2 zeigt die gemessenen Kenngrößen der Regelkreise für einen Störgrößensprung von $-0,5 \text{ V}$. Zuerst werden die drei PI-Regler untereinander verglichen und danach werden die Ergebnisse des Agenten mit betrachtet. Alle drei PI-Regler wurden für einen Führungsgrößensprung ausgelegt. Betrachtet man die Einschwingzeiten, hat sich bei den Werten im Vergleich zu den Ergebnissen beim Führungsgrößensprung nicht wirklich etwas verändert. Nur die Einschwingzeit beim über die Tune-Funktion ausgelegten Regler ist 10 Sekunden schneller.

Auch bei der maximalen Überschwingweite hat sich bei den PI-Reglern nicht viel verändert. Die Ergebnisse bei allen drei PI-Reglern hat sich um ein paar Prozentpunkte verschlechtert, denn das Verhältnis der maximalen Überschwingweiten ist größer geworden. Die maximale bleibende Regelabweichung bei den drei PI-Reglern ist gleich, aber im Vergleich zu dem Ergebnis bei dem Führungsgrößensprung hat sich die maximale bleibende Regelabweichung verdoppelt beziehungsweise mehr als verdoppelt. Die leicht schlechteren Ergebnisse resultieren daraus, dass die PI-Regler für die Übertragungsfunktion der Regelstrecken bei einem Sollwertsprung ausgelegt wurden und nicht für einen Störgrößensprung am Ausgang der Strecke.

Das Verhalten des Regelkreises mit Agenten bei einem Störgrößensprung ist im Vergleich zum Verhalten beim Führungsgrößensprung schlechter. Ausnahme ist die Einschwingzeit. Die hat sich um 30 Sekunden verkürzt. Die maximale bleibende Regelabweichung hat sich verdoppelt und die maximale Überschwingweite ist 63% größer. Betrachtet man den Verlauf der Regelgröße (blau) aus Abbildung 6.4 erkennt man, dass die Störgröße nicht vollständig kompensiert wird. Im Gegensatz dazu wird bei den PI-Reglern die Störgröße vollständig kompensiert. Es gibt mehrere Gründe, warum der Agent die Störgröße nicht vollständig kompensieren kann. Zu den Gründen gehört, dass der Agent über den Zustandsvektor nicht mitbekommt, wenn eine Störgröße auftritt und die auftretende Störgröße verändert die Ausgangslage der Umwelt. Das hat folgende Auswirkung: Im Training lernt der Agent die Ausgangstemperatur der Strecke zu regeln. Die Temperatur der Luft, die in die Anlage eintritt, bestimmt, um wie viel Grad die Luft erwärmt werden muss, um den Sollwert zu erreichen und entspricht hier der Raumtemperatur des Labors. Beispiel: Die Eintrittstemperatur soll von 21 °C auf 40 °C erhöht werden. Die Lufttemperatur soll also um 19 °C gegenüber der Raumtemperatur erwärmt werden und dafür wird im eingeschwungenen Zustand die Stellgröße 6 V benötigt. Der Agent lernt jetzt, in dieser Umwelt, in der die Raumtemperatur 21 °C beträgt, die Strecke zu regeln. Im optimalen Fall lernt der Agent, dass wenn die Luft auf 40 °C geregelt werden soll, die Stellgröße im eingeschwungenen Zustand 6 V betragen muss. Der trainierte Agent wird nun zum Regeln eingesetzt. Es tritt nun im laufenden Prozess die Störgröße auf und senkt die Raumtemperatur um 5 °C, weil das Fenster geöffnet wurde. Der Agent würde dann - wie die normalen PI-Regler - versuchen, die Störgröße zu kompensieren, um wieder eine Austrittstemperatur von 40 °C zu erreichen. Der Agent hat im Training gelernt, dass die Stellgröße für eine Austrittstemperatur von 40 °C im eingeschwungenen Zustand 6 V betragen muss und er würde die Stellgröße auf diesen Wert bringen. Das würde aber jetzt nur zu einer Austrittstemperatur von 35 °C führen,

da die Stellgröße von 6 V nur zu einer Temperaturerhöhung von 19 °C gegenüber der Raumtemperatur führt. Zu einem gewissen Grad kann der Agent die Änderung der Bedingungen der Umwelt durch sein restliches Wissen über die Umwelt ausgleichen, wie der Verlauf der Regelgröße aus Abbildung 6.4 zeigt. Um Störungen dieser Art vollständig kompensieren zu können, müsste der Agent diese während des Lernprozesses trainieren. Dafür müsste er aber auch über den Zustand die Information bekommen, dass diese Störung aufgetreten ist. Das in dieser Arbeit nicht möglich ist, weil es keinen Temperaturmesspunkt am Eingang der Strecke gibt. Außerdem kann die Eingangstemperatur beziehungsweise die Raumlufttemperatur nicht geändert werden.

Zusammenfassend kann gesagt werden, dass das Regelverhalten des Agenten auf einen Störgrößensprung am Ausgang der Strecke schlechter ist als das Regelverhalten der PI-Regler. Es gibt keinen Grund, warum man nicht einen PI-Regler mit Polkompensation dem Agenten vorziehen sollte.

Tabelle 6.2: Kenngrößen der Störgrößensprungantworten der Regelkreise

Auslegungsart	Störgrößensprung		
	$e_{\max, \infty}$	e_{\max}	Einschwingzeit
Ziegler-Nichols	2%	70%	75 s
Polkompenstaion	2%	16%	35 s
Tune-Funktion	2%	44%	70 s
Soft-Actor-Critic -Agent	6%	36%	45 s

6.3 Vergleich der Stellgröße

In diesem Abschnitt wird untersucht, inwiefern der Verlauf der Stellgröße des Agenten auf einen Sollwertsprung mit dem Verlauf der Stellgröße eines Standardreglers vergleichbar ist. Es wird geprüft, ob das vom Agenten gelernte Regelverhalten Komponenten besitzt, die normalerweise unerwünscht sind. Abbildung 6.9 zeigt den Verlauf der Stellgröße auf den Sollwertsprung aus Abbildung 6.2. Abbildung 6.10 zeigt den Verlauf der Stellgröße auf den in Abbildung 6.4 gezeigten Sollwertsprung. Beide Stellgrößen wurden direkt am Ausgang des Reglers aufgezeichnet.

6 Vergleich des trainierten Agenten mit den dimensionierten PI-Reglern

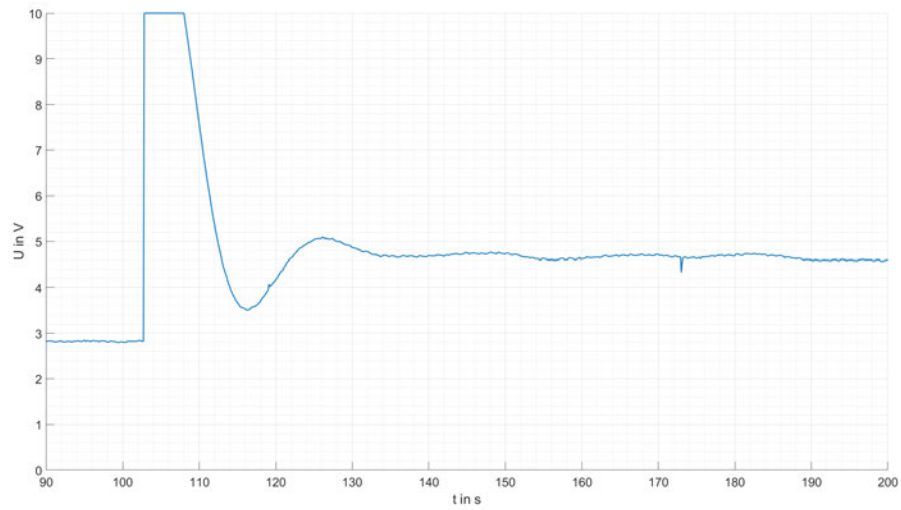


Abbildung 6.9: Verlauf der Stellgröße des Agenten auf einen Sollwertsprung

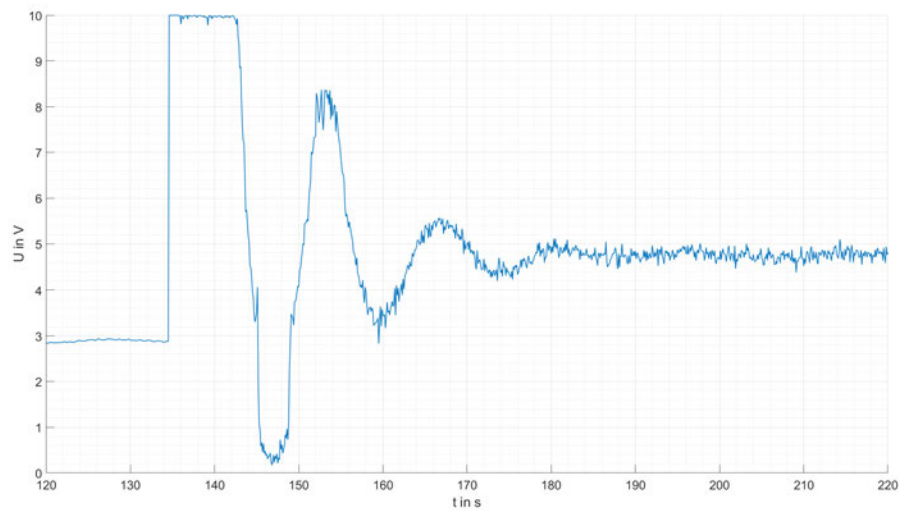


Abbildung 6.10: Verlauf der Stellgröße des Reglers mit Polkompensation auf einen Sollwertsprung

Der Vergleich von Abbildungen 6.9 und 6.10 zeigt, dass der Verlauf der Stellgröße des Agenten einen hochfrequenten Teil hat. Das hochfrequente Verhalten ist unerwünscht, weil es die

verwendeten Komponenten unnötig belastet. Beispiel: In großen Lüftungsanlagen wird die Lufttemperatur oft durch Heizregister erwärmt. Die Wassertemperatur und damit die Wärmeenergie, die an die Luft abgegeben werden kann, wird durch Misch-Ventile geregelt. Die Ventile ändern die Austrittsstemperatur, indem das Verhältnis von Warm- und Kaltwasser geändert wird. Die vielen Schwankungen in der Stellgröße - wie es beim Agenten der Fall ist - würde bedeuten, dass der Antrieb das Ventil in kurzen Zeitabständen leicht öffnet und schließt. Das führt zu einem erhöhten Verschleiß und verkürzt die Lebenszeit des Antriebs und des Ventils. Das ist ein Verhalten, das der Agent im Training gelernt hat, weil es nicht in der Belohnung berücksichtigt wird. Das hat aber folgenden Grund - wie es in [6] beschrieben wird: Würde der Agent dafür belohnt werden, dass die Stellgröße nicht stark in kurzen Zeitabständen schwankt, könnte es den Agenten von seinem ursprüngliche Ziel abbringen. Eine Auswirkung könnte sein, dass sich der Agent im Training mehr auf die Maximierung dieser Belohnung fokussiert, weil sie leichter zu erreichen ist, als die Regelgröße zu minimieren. Es müsste dann Zeit dafür investiert werden, einen passenden Kompromiss zwischen den Zielen zu finden.

6.4 Vergleich des Zeitaufwands

In diesem Abschnitt soll untersucht werden, inwiefern der Zeitaufwand, der benötigt wird, um die PI-Regler zu dimensionieren, beziehungsweise den Agenten zu trainieren, die Entscheidung beeinflusst, ob der Agent für die Regelung von Lüftungsanlagen geeignet ist. Das schnellste Verfahren zur Auslegung eines PI-Reglers ist über die Tuning-Funktion des PI-Reglers. Am Industrieregler muss nur ein Sollwertsprung ausgeführt werden und dann bestimmt der Industrieregler die entsprechenden Parameter für den PI-Regler. Die benötigte Zeit liegt unter 30 Minuten. Die Auslegung mittels der vorgestellten Methode der Ziegler-Nichols Einstellregeln ist ähnlich schnell. Es muss nur eine Sprungantwort aufgezeichnet werden und dann kann man graphisch die Parameter bestimmen. Sollen die Parameter genauer bestimmt werden, so wie es in dieser Arbeit über das Curve-Fitting Tool geschehen ist, wird ein wenig länger benötigt, da zuerst die Sprungantwort der Regelstrecke im Zeitbereich bestimmt werden muss. Die Auslegung eines PI-Reglers mittels Polkompensation ist von den untersuchten Verfahren das, welches am meisten Zeit in Anspruch nimmt. Für die Dimensionierung wurden weniger als zwei Stunden benötigt. Wie bereits in den vorherigen Abschnitten gezeigt, ist das resultierende Regelverhalten mit Abstand das Beste, wenn die Einschwingzeit und maximale Überschwingweite betrachtet wird.

Der Zeitaufwand für die Entwicklung und das Training des Agenten ist enorm. Der zum Vergleichen eingesetzte Agent hat ungefähr 14 Stunden und 35 Minuten benötigt, um das hier vorgestellte Regelverhalten zu lernen. Alleine die für das Training benötigte Zeit übersteigt das 7fache der Zeit, die für die Auslegung des PI-Reglers mit Polkompensation benötigt wurde und das gelernte Regelverhalten ist deutlich schlechter, als das des PI-Reglers. Außerdem muss berücksichtigt werden, dass die 14 Stunden und 35 Minuten für nur einen Trainingsdurchlauf waren, der beim ersten Versuch ein im Vergleich zu den anderen Agenten gutes Ergebnis geliefert hat. Wie bereits in Abschnitt 5.3 erläutert, ist ein Trainingserfolg nicht garantiert, was die benötigte Trainingszeit nochmal vervielfachen kann. Wird direkt an der Anlage trainiert, kann die Anlage für den Trainingszeitraum für nichts anderes verwendet werden, was die Einsatzmöglichkeiten einschränkt, weil nur außerhalb von Geschäftszeiten trainiert werden kann. Es muss auch bedacht werden, dass das für diese Arbeit verwendete Modell einer Lüftungsanlage vergleichsweise schnell ist. Der Regelfehler nach einem Sollwertsprung ist zum Teil innerhalb von 90 Sekunden minimiert. Große Lüftungsanlagen, die ganze Gebäude mit Luft versorgen, benötigen, je nach dem wie sie dimensioniert wurden, einen zweistelligen Minutenbereich bis sie eingeschwungen sind und ihren neuen Sollwert erreicht haben. Das würde die Trainingszeit auf mehrere Tage erhöhen, ohne das garantiert ist, dass der trainierte Agent das gewünschte Verhalten gelernt hat. Betrachtet man den Zeitaufwand, bietet eine Regelung über Reinforcement Learning Agenten keinen Vorteil gegenüber Standardreglern.

7 Fazit

Das Ziel dieser Arbeit war es zu untersuchen, inwiefern sich Reinforcement Learning Algorithmen eignen, um einen Agenten zu trainieren, der in der Lage ist, eine Lüftungsanlage zu regeln. Der Agent wurde anhand unterschiedlicher Kriterien bewertet. Dazu gehörte ein Vergleich mit mehreren PI-Reglern, die über verschiedene Verfahren ausgelegt wurden. Die Simulation hat gezeigt, dass der Agent lernen kann, ein mathematisches Modell einer Lüftungsanlage zu regeln. Das in der Simulation gelernte Regelverhalten für ein mathematisches Modell der Lüftungsanlage hat eine ähnliche Qualität wie das Regelverhalten der PI-Regler. Die Simulationen haben aber auch gezeigt, dass die Trainingsergebnisse stark variieren können. Nicht aus jedem abgeschlossenen Training resultiert ein Agent, der ein gutes Regelverhalten aufweist.

Die in der Simulation erzielten Ergebnisse konnten nicht auf das reale Modell der Lüftungsanlage übertragen werden. Um zu testen, ob ein trainierter Agent eine Alternative zu den Standardreglern ist, wurden mehrere Messungen durchgeführt. Es wurde das Verhalten des Agenten bei Führungsgrößensprüngen sowie bei auftretenden Störgrößen am Ausgang der Regelstrecke untersucht. Das Regelverhalten des Agenten wurde mit dem Regelverhalten der Standardregler verglichen. Die Messungen am realen Modell der Lüftungsanlage haben gezeigt, dass der in dieser Arbeit implementierte Algorithmus nicht in der Lage ist, einen Regler zu trainieren, dessen Regelverhalten am Versuchsaufbau ein ähnliches Ergebnis erzielt wie ein PI-Regler. Die Untersuchung bei einem Führungsgrößensprung hat gezeigt, dass beim Agenten im stationären Zustand eine konstante bleibende Regelabweichung bleibt. Die Messungen haben außerdem gezeigt, dass eine auftretende Störgröße vom Agenten nicht komplett ausgeregelt werden kann. Ein weiterer Nachteil ist, dass die Stellgröße in kurzen Abständen stark schwankt.

Wie zu Beginn der Arbeit beschrieben, war der ursprüngliche Gedankengang, dass durch den Einsatz von Reinforcement Learning Zeit und damit Geld gespart werden kann, weil kein qualifizierter Techniker die Streckenparameter bestimmen muss. Wie aber im Laufe der Arbeit

gezeigt wurde, müssen die Streckenparameter bestimmt werden, damit die Auswirkung der Totzeit auf das Trainingsergebnis kompensiert werden kann. Ein weiterer Punkt, der gegen die Eignung spricht, ist der benötigte Trainingsaufwand. Wie bereits im Laufe der Arbeit erläutert, kann es mehrere Tage dauern, bis ein Agent trainiert ist, der das gewünschte Regelverhalten gelernt hat. In der Zeit, in der trainiert wird, kann die Anlage nicht normal betrieben werden.

Diese Arbeit hat gezeigt, dass Reinforcement Learning Algorithmen theoretisch das Potential haben, einen Agenten zu trainieren, der ähnlich gutes Regelverhalten hat wie ein entsprechend dimensionierter PI-Regler. Die Unzuverlässigkeit des Trainingserfolges sowie der benötigte Zeitaufwand sprechen aktuell gegen den Einsatz von Reinforcement Learning Algorithmen zum Regeln von Lüftungsanlagen.

A Inhalt der DVD

Der Anhang zur Arbeit befindet sich auf einer DVD und kann beim Erstgutachter Prof. Dr. Michael Erhard eingesehen werden. Die folgende Liste zählt den Inhalt der DVD auf:

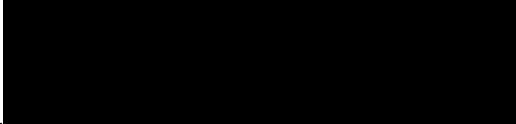
- **Matlab_Code**
Ordner mit verwendetem Matlab Code und Simulink Modellen
- **Messdaten**
Ordner mit aufgenommenen Messreihen als .fig Datei
- Bachelorarbeit_PascalStubel.pdf

Literaturverzeichnis

- [1] ACHIAM, Joshua: *Spinning Up Documentation*. Spinning Up OpenAi, 2020
- [2] ATP2: *Praktikum Grundlagen der Regelungstechnik*. 2021
- [3] BÜRKERT: *Digitaler Industrieregler Digital Industrial Controller Type 1110, Bedienungsanleitung*
- [4] COVER, Thomas M. ; THOMAS, Joy A.: *Elements of Information Theory*. Wiley-Interscience, 2006 (An Introduction). – ISBN 78-0-262-19398-6
- [5] HAARNOJA, Tuomas ; ZHOU, Aurick ; ABBEEL, Pieter ; LEVINE, Sergey: *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. 2018. – URL <https://arxiv.org/abs/1801.01290>
- [6] IRPAN, Alex: *Deep Reinforcement Learning Doesn't Work Yet*. <https://www.alexirpan.com/2018/02/14/rl-hard.html>. 2018
- [7] MATHWORKS: *Define Reward Signals*. – URL <https://ch.mathworks.com/help/reinforcement-learning/ug/define-reward-signals.html>
- [8] MATHWORKS: *Options for SAC agent*. – URL <https://ch.mathworks.com/help/reinforcement-learning/ref/rlsacagentoptions.html>
- [9] MATHWORKS: *Reinforcement Learning Agents*. – URL <https://www.mathworks.com/help/reinforcement-learning/ug/create-agents-for-reinforcement-learning.html>
- [10] MATHWORKS: *Soft Actor-Critic Agents*. – URL <https://de.mathworks.com/help/reinforcement-learning/ug/sac-agents.html>
- [11] MATHWORKS: *Soft actor-critic reinforcement learning agent*. – URL <https://ch.mathworks.com/help/reinforcement-learning/ref/rlsacagent.html>

- [12] MEYER, Martin: *Signalverarbeitung*. Vieweg+Tuebner, 2011 (Analoge und digitale Signale, Systeme und Filter). – ISBN 978-3-8348-0897-4
- [13] SUTTON, Richard S. ; BARTO, Andrew G.: *Reinforcement Learning*. The MIT Press, 2011 (An Introduction). – ISBN 78-0-262-19398-6
- [14] UNBEHAUEN, Heinz: *Regelungstechnik 1*. Vieweg+Tuebner, 2008 (Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme, Fuzzy-Regelsysteme). – ISBN 978-3-8348-0497-6
- [15] ZACHER, Serge ; REUTER, Manfred: *Regelungstechnik für Ingenieure*. Springer Vieweg, 2017 (Analyse, Simulation und Entwurf von Regelkreisen). – ISBN 978-3-658-17631-0

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach § 16 (5) APSO-TI-BM/APSO-INGI ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 11.04.2022  _____
Pascal Stubel