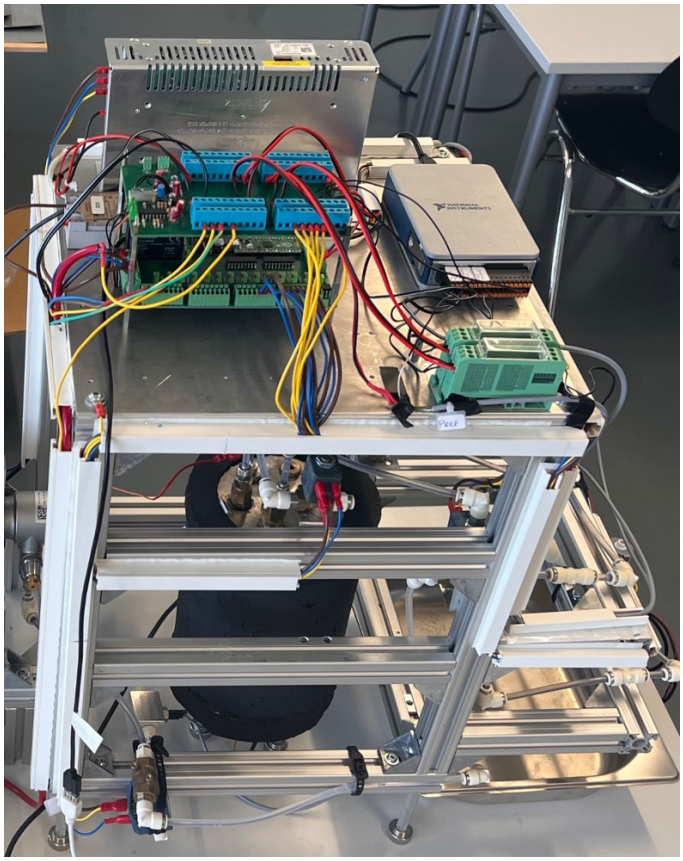
Updates am Pumpenprüfstand

Optimierung der Bedienung und Implementierung der Kalibrierprozesse
SoSe 2023

Patricia Viebke



Inhaltsverzeichnis

Mit einer Projektarbeit wurden die Grundfunktionen des Pumpenprüfstands in Betrieb genommen [95]. Zu dem Zeitpunkt hat der Prüfstand eine GUI, über welche dieser gesteuert wird. Durch den Einsatz eines Microcontrollers und einer NI-Messkarte findet die Kommunikation von Befehlen bzw. Messwerten statt. Die zuverlässige Bedienung und Funktion des Prüfstands wird gewährleisteten.

Der nächste Schritt umfasst die Entwicklung einer Temperatur- und Druckregelung. Die zwei Regler stellen die Grundlage für den späteren Kalibrierprozess von Sensoren da. Im Laufe der Entwicklung der Kalibrierprozesse werden weitere Updates und Änderungen vorgenommen, um den Bedienkomfort und den Automatisierungsgrad zu erweitern.

1. GUI-Benutzeroberfläche

Der Grundaufbau der GUI hat sich nicht verändert. Die Benutzeroberfläche lässt sich in drei Teile einteilen: Prüfstands Überwachung, Pico Ausgabe und TabGroup für die Interaktion.

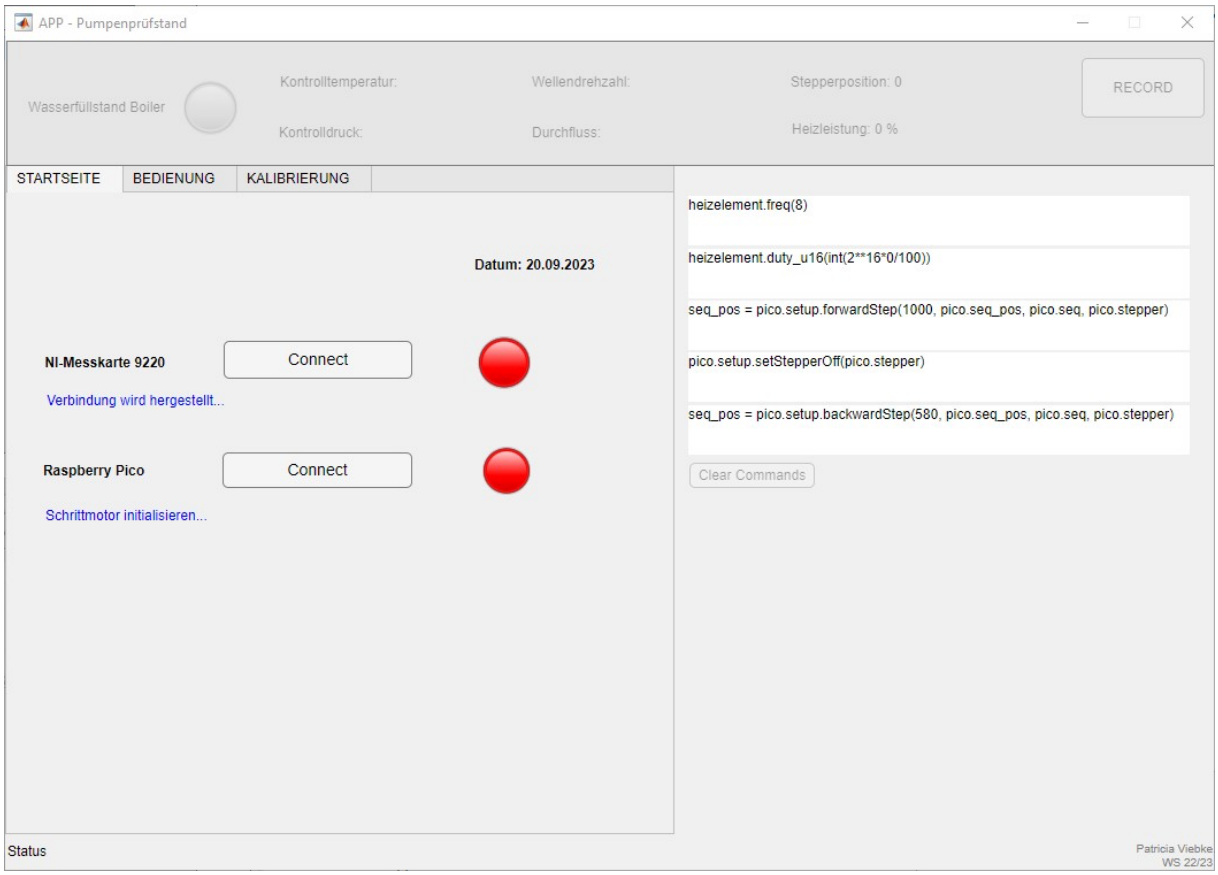


Abbildung 1: neue GUI-Startseite

Wie in Abbildung 1 zu sehen ist, wurden auf der Startseite Labels für die Anzeige der Verbindungsprozesse in blauer Schrift hinzugefügt. Da das Herstellen einer Verbindung teilweise etwas länger dauert. Durch die Labels wird das doppelte Betätigen der Buttons vermieden, was zu Fehlermeldungen führt.

Beim Herstellen der Verbindung mit dem Raspberry Pico wurden bisher nur die benötigten MicroPython Skripte importiert. Nun wird neben dem Import auch das Heizelement und das Dosierventil initialisiert, so dass ein definierter Ausgangszustand gegeben ist.

1.1. Prüfstands Überwachung

Bei der Prüfstands Überwachung hat sich das Erscheinungsbild etwas verändert. Der ursprünglich geplante Drehmomentsensor wird aus dem Prüfstand entfernt, da die Drehmomentmessung vorerst keine Priorität hat. Anstatt von der Drehmoment Anzeige in der

Überwachung, werden Labels für die Überwachung von Schrittmotor und Heizelement hinzugefügt. Durch ein fachliches Telefonat mit AVS Römer zum verwendeten elektrischen Dosierventils, werden Informationen über die Initialisierung bekannt. Die Spindel in dem Dosierventil, welche für das Öffnen und Schließen zuständig ist, kann sich bei zu weitem Öffnen aushängen. Das Schließen ist jedoch als fest Zustand definiert. Das Schließen kann also mit mehr Schritten angesteuert werden als notwendig, das Öffnen nicht.

Für die Initialisierung wird eine feste Vorgehensweise definiert. Das Dosierventil wird durch die Ansteuerung des Raspberry Picos um 1000 Halbschritte geschlossen, anschließend um 580 Halbschritte geöffnet. Bei der Ansteuerung des Schließens ist zuerst eine leichte Vibration, anschließend ein Brummen wahrzunehmen. Das Brummen ist ein Zeichen für das Erreichen des Endzustands. Durch eine fest definierte Initialisierung ist es möglich einen Ausgangszustand mit 0 Schritten festzulegen. Durch Addition beim Schließen bzw. Subtraktion beim Öffnen des Dosierventils, wird die aktuelle Stepperposition ermittelt.

Die Anzeige dient vor allem zur Übersicht bei der Druckkalibrierung. Der Prüfstand ist bei maximalem Verschließen bei einer Drehzahl von 1.400 1/min auf 13 bar ausgelegt.

Zusätzlich zum Dosierventil wird die aktuelle Heizleistung des Heizelements im Boiler prozentual angegeben. Die Überwachung erfüllt vor allem bei der Durchführung einer Kalibrierung von Temperatursensoren die Funktion. Beim Übergeben der Heizleistung von MATLAB als PWM-Signal an das XSSR und das Heizelement, erscheint in der Pico Ausgabe der verwendete Befehl.

```
writeline(app.pico, ['heizelement.duty_u16(int(2**16*50/100))'])
```

Hier wird eine Heizleistung von 50 % übergeben. Da diese Befehle in der Pico Ausgabe nicht sehr übersichtlich sind für die Nachverfolgung, wird die Heizleistung in der Prüfstands Überwachung angezeigt.

1.2. TabGroup

Auf der Startseite sind die Buttons für den Verbindungsaufbau mit der NI-Messkarte und dem Raspberry Pico vorzufinden. Bei Betätigen der Buttons erscheint in blauer Schrift der Prozess, welcher gerade bei der Verbindung durchgeführt wird.

Startseite
Bei der NI-Messkarte dauert der Verbindungsaufbau nach Neustarten von MATLAB etwas länger. In diesem Fall darf der Button ebenfalls nur einmal betätigt werden, da sonst eine Fehlermeldung erscheint. Beim Verbinden mit dem Raspberry Pico wurden einige Schritte hinzugefügt.

Abbildung 2: Startseite mit Zusatzinfo beim Verbindungsprozess

Neben dem Importieren der benötigten MicroPython Skripte, wird zusätzlich eine Initialisierung des PWM-Pins für das Heizelement und eine Initialisierung des Dosierventils durchgeführt. Durch Ausprobieren hat sich gezeigt, dass nach Erstellen des Ports für die serielle Schnittstelle etwas Zeit benötigt wird, bevor die MicroPython Befehle ausgeführt werden können. In dem Code der GUI sind daher Pausen von 1 Sekunde zwischen den Schritten zu finden. Damit erkennbar ist, ob der Verbindungsaufbau zu dem Pico und der Messkarte gestartet ist, werden Labels hinzugefügt (Abbildung 2), welche Informationen darüber ausgeben.

Bedienung

Weiter Änderungen gibt es auf dem Tab Bedienung. Die GUI ist hier unterteilt in zwei Hälften, wobei die obere Hälfte die automatisierten Vorgänge im Prüfstand darstellt. Die Grundfunktionen Befüllen, Entleeren, Entlüften lagen in Form von State Buttons vor. Da der Einsatz der State Buttons zu Problemen führte, wird hier auf normale Buttons umgestellt. Die Buttons schalten bei Betätigen die definierten Magnetventile des jeweiligen Vorgangs. Lediglich die Pumpe muss selbstständig eingeschaltet werden. Das Deaktivieren eines Vorgangs muss manuell durch den RESET Button ausgelöst werden. Hier wird das MicroPython Skript *reset.py* durchgeführt, welches die Magnetventil in den Ausgangszustand versetzt.

Der Befüllvorgang ist der einzige dieser Grundfunktionen, welcher selbstständig beendet wird, bei Erreichen des Boilerfüllstands. Hier werden die Pumpe und der Button ausgeschalten, wenn das Wasser mit dem Stab in Berührung kommt. Da die Füllstandsanzeige nur qualitativ und nicht quantitativ misst, verhindert die Automatisierung des Befüllvorgangs ein Überfüllen des Boilers.

Kalibrierung

Zu den vorhandenen Tabs wird ein Weiterer für die Kalibrierung programmiert. Hier muss der Benutzer mit der ButtonGroup auswählen, ob ein Druck- oder ein Temperatursensor für die Kalibrierung vorliegt. Die Voraussetzung für die Auswahl der Druckregelung ist, dass die Wassertemperatur 30 °C nicht überschreitet. Ein Temperatursensor hingegen kann nur dann kalibriert werden, wenn der Kontrolldruck kleiner 1 bar ist.

Für die eindeutige Zuweisung der Daten, bietet es sich an die Bezeichnung des zu kalibrierenden Sensors einzugeben. Im nächsten Schritt müssen die Parameter Zielwert und Anzahl der Kontrollpunkte eingegeben werden. Der Zielwert bzw. Zieldruck wird durch die im Pumpenprüfstand verbauten Komponenten beschränkt.

Für den Druck ist ein maximaler Wert von 12 bar möglich, während die Eingabe einer Temperatur bis 100 °C möglich ist. Bei der Anzahl der Kontrollpunkte muss ein Wert sinnvoll abhängig vom Kalibrierbereich gewählt werden. Je höher die Anzahl der Kontrollpunkte ist, desto präziser wird die Kalibrierkurve für den Sensors. Es besteht jedoch das Risiko, dass bei einer zu hohen Anzahl die Werte nicht genau genug geregelt werden können. Für die Absicherung der Druckregelung in einem Bereich von 2 bis 12 bar wurden 7 Kontrollpunkte verwendet. Bei der Temperaturkalibrierung Absicherung in einem Bereich von 25 bis 95 °C wurden 8 Kontrollpunkte angegeben.

In Abbildung 3 ist ein Kalibriervorgang für Drucksensoren aktiv. Unter den Eingabefeldern sind vier graue Textfelder mit Informationen zu der Kalibrierung vorzufinden. Die Informationen in den Labels werden nach dem ersten Durchlauf der jeweiligen Reglerfunktion angezeigt. Die Kontrollwerte zeigen die Zwischenpunkte an, welche sich aus dem Kalibrierbereich und der Anzahl der Kontrollpunkte errechnen.

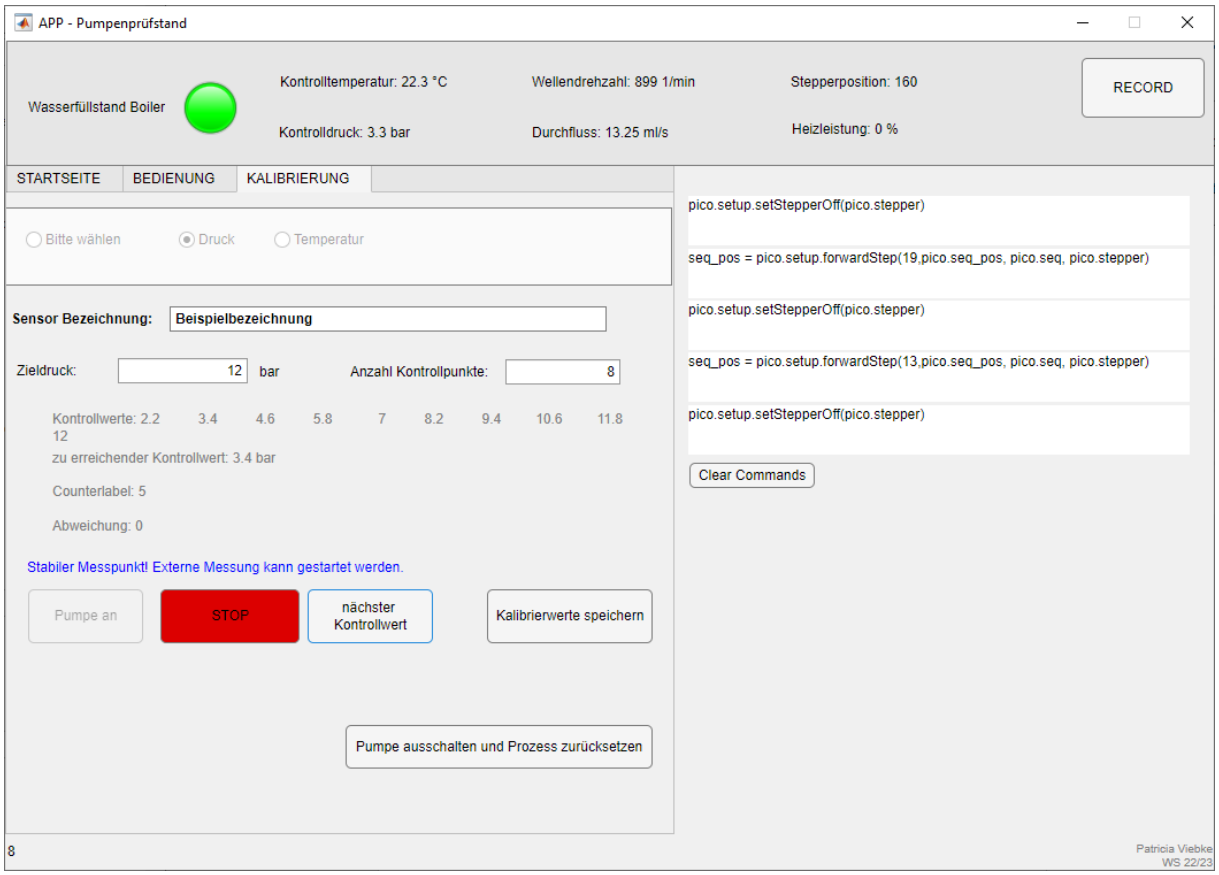


Abbildung 3: GUI bei aktivem Kalibrierprozess

Das nächste Label zeigt den zu erreichenden Kontrollpunkt an. Damit hat der Benutzer die Möglichkeit die Druckänderung während des Prozess zu kontrollieren und bei zu starken Abweichungen einzugreifen. In diesem Fall wird der zweite Kontollpunkt von 3,4 bar angesteuert. In der Prüfstandsüberwachung ist ein Druck von 3,3 bar abzulesen, was innerhalb der Toleranz eines Kontrollpunkts liegt. Das Counterlabel ist die Anzeige für eine Zählervariable, welche aktiviert wird, wenn der Kontrolldruck innerhalb der Toleranz liegt. Bei jedem weiteren Durchlauf der Reglerfunktion zählt der Counter um eins hoch. Bei Erreichen des Werts von 5, wird der Kontrollpunkt als stabiler Punkt definiert. Bei der Druckregelung muss der Counter gleich 5 sein, was nach 10 s der Fall ist. Da die Temperaturregelung etwas verzögert abläuft, wird hier erst ab einem Counter 10 der Punkt als stabil definiert. Zusätzlich wird noch die Abweichung zum aktuellen Kontrollpunkt angezeigt.

Unter den Info Labels befinden sich 5 Buttons, die die Kalibrierung steuern. Bevor die Kalibrierung gestartet werden kann, muss die Pumpe eingeschaltet werden. Die Pumpe muss manuell über den Drehknopf des 24V-Leistungsnetzteils auf die entsprechende Drehzahl eingestellt werden. Es wird eine Drehzahl von 1200 1/min bis 1400 1/min empfohlen. Anschließend kann die Kalibrierung gestartet werden. Der Start Button wird nach Betätigung zum Stopp Button. Bei Erreichen eines Kontrollpunkts, kann der Benutzer selbst entscheiden, wann zum nächsten Kontrollpunkt hochgeschaltet wird. Dieser Schritt wird nicht automatisiert, da für die Kalibrierung von Sensoren ein zweites System benötigt wird, in welchem die zu kalibrierenden Sensoren verbaut sind. Über dieses System muss ebenfalls eine Messdatenaufnahme des Drucks im Prüfstand stattfinden. Erst durch Gegenüberstellung des Referenzwerte des Prüfstands mit den Messdaten des externen Systems kann eine Kalibrierkurve bzw. Kennlinie ermittelt werden.

Die Funktion des STOP Buttons ist als Übergangslösung geplant. Dieser kann entfernt werden, da durch den letzten Button der gesamte Kalibrierprozess zurückgesetzt wird. Der letzte Button

schaltet die Pumpe aus und je nachdem welcher Sensor kalibriert wurde, wird entweder das Dosierventil initialisiert oder das Heizelement ausgeschaltet. Mit einem Button „Kalibrierwerte speichern“ werden die aufgezeichneten Messwerte in einem MATLAB-Struct gespeichert. Das Struct ist dem aktuellen MATLAB Ordner zu finden, versehen mit einem Zeitstempel zum Zeitpunkt des Speichervorgangs. Im Appdesigner sind weitere Zeilen auskommentiert. Bei Entkommentieren werden die Messwerte direkt in das MATLAB workspace gesendet. Von dort aus kann eine direkte Überprüfung stattfinden. Diese Vorgehensweise wurde bei der Entwicklung bevorzugt verwendet. Die Variablen können vom workspace aus direkt weiterverarbeitet bzw. graphisch dargestellt werden.

1.3. Log File

Für die Nachverfolgbarkeit der Bedienung und der Zuverlässigkeit des Microcontrollers wird beim Starten der GUI ein Logfile „log.txt“ erstellt. In dem Logfile werden bei Benutzung der GUI alle Befehle erfasst, die über den Raspberry Pico laufen. Beim Starten der GUI wird ein Erkennungsbefehl geschrieben, dass die StartUp Funktion aufgerufen wurde.

```
app.logfile = fopen('log.txt', 'w');
txt = '# GUI STARTUPFCN';
fprintf(app.logfile, '%s\n', txt);
```

Das Logfile dient für die Ursachenuntersuchung bei auftretenden Fehlern. Auch die Korrektheit der Bedienung kann hier überprüft werden. In dem Logfile sind Erkennungsbefehle immer mit einem # gekennzeichnet. Diese werden ohne Zeitstempel in das Logfile geschrieben. Die Erkennungsbefehle dienen der Orientierung im Logfile. Die Erkennungsbefehle kommen nicht vom Pico, sondern werden direkt von MATLAB als string in die Datei geschrieben. Nur die Befehle mit Zeitstempel sind vom Pico ausgeführte Kommandozeilen.

```
# GUI STARTUPFCN
# VERBINDUNG MIT PICO
2023-Sep-20 10:32:56     import connect
2023-Sep-20 10:32:56     import pico
2023-Sep-20 10:32:56     import reset
2023-Sep-20 10:32:56     connect.led_pico.on()
2023-Sep-20 10:32:56     heizelement = machine.PWM(machine.Pin(3))
2023-Sep-20 10:32:57     heizelement.freq(8)
2023-Sep-20 10:32:57     heizelement.duty_u16(int(2**16*0/100))
2023-Sep-20 10:32:58     seq_pos = pico.setup.forwardStep(1000, pico.seq_pos, pico.seq, pico.stepper)
2023-Sep-20 10:33:00     pico.setup.setStepperOff(pico.stepper)
2023-Sep-20 10:33:00     seq_pos = pico.setup.backwardStep(580, pico.seq_pos, pico.seq, pico.stepper)
2023-Sep-20 10:33:01     pico.setup.setStepperOff(pico.stepper)
2023-Sep-20 10:33:01     connect.skript()
2023-Sep-20 10:33:01     connected
# DRUCKKALIBRIERUNG WIRD DURCHGEFÜHRT
2023-Sep-20 10:33:09     pico.setup.ssr_on(pico.ventile,3)
2023-Sep-20 10:33:09     pico.setup.ssr_off(pico.ventile,5)
```

Abbildung 4: Erzeugtes Logfile

Die vom Raspberry Pico aufgerufenen Befehle werden in den fünf Zeilen bei der Pico Ausgabe in der GUI angezeigt. Da diese nicht permanent zu sehen sind, werden diese ebenfalls in das Logfile aufgenommen. Die Befehle vom Raspberry Pico starten alle mit „writeline“. Bei dem Schreiben der Pico Befehle in das Logfile, wird ein Zeitstempel zum Zeitpunkt der Ausführung dazugeschrieben. Das Logfile wird automatisch in dem aktuell geöffneten MATLAB Pfad als ‚log.txt‘ gespeichert und kann in direkt geöffnet werden.

2. PID-Reglerfunktionen

Der Prüfstand wird primär für die Kalibrierung von Sensoren entwickelt. Durch ein Magnetventil Y03, wird zwischen den beiden Messbalkonen unterschieden. Je nachdem, ob Temperatur oder Druck gewählt wird, kann ein Sensor des jeweiligen Typs kalibriert werden.

Für die Temperatur- und Druckregelung wird jeweils eine Funktion programmiert, welche für die Regelung zuständig ist. Die beiden Funktionen werden in der Funktion für die Messdatenverarbeitung aufgerufen, wenn die Voraussetzungen erfüllt sind. Bei einer Abtastrate von 40.000 S/s und refresh = 4, wird die Datenverarbeitung jede viermal pro Sekunde ausgeführt.

2.1. Druckregelung

Damit die Druckregelung aktiviert wird, muss in der Button Group der Druck ausgewählt sein und der Kalibriervorgang muss gestartet sein. Anschließend wird mit der Variable *app.mess_counter* hochgezählt. Die Variable ist eine Zählvariable, die das Intervall für das Ausführen der Druckregelung bestimmt. Mit aktueller Parametrierung wird die Funktion *druckregelung* bei einem Messcounter von 10, also alle 2.5 s, ausgelöst. Das Intervall kann durch Ändern der Bedingung variiert werden. Da der Druck jedoch eine sofortige Änderung auf das Verhalten des Dosierventils zeigt, kann das Intervall kurzgehalten werden. Während jedes Durchlaufs der Datenaufbereitung werden 10.000 Messwerte pro Messkanal erzeugt, woraus anschließend der RMS-Wert gebildet wird. So ist pro Durchlauf nur ein Wert vorhanden. Dieser RMS-Wert wird in den RMS-Buffer geschrieben. Nach 10 Durchläufen wird ein Mittelwert des RMS-Buffers erzeugt. Dieser Mittelwert wird an die Funktion *druckregelung* übertragen.

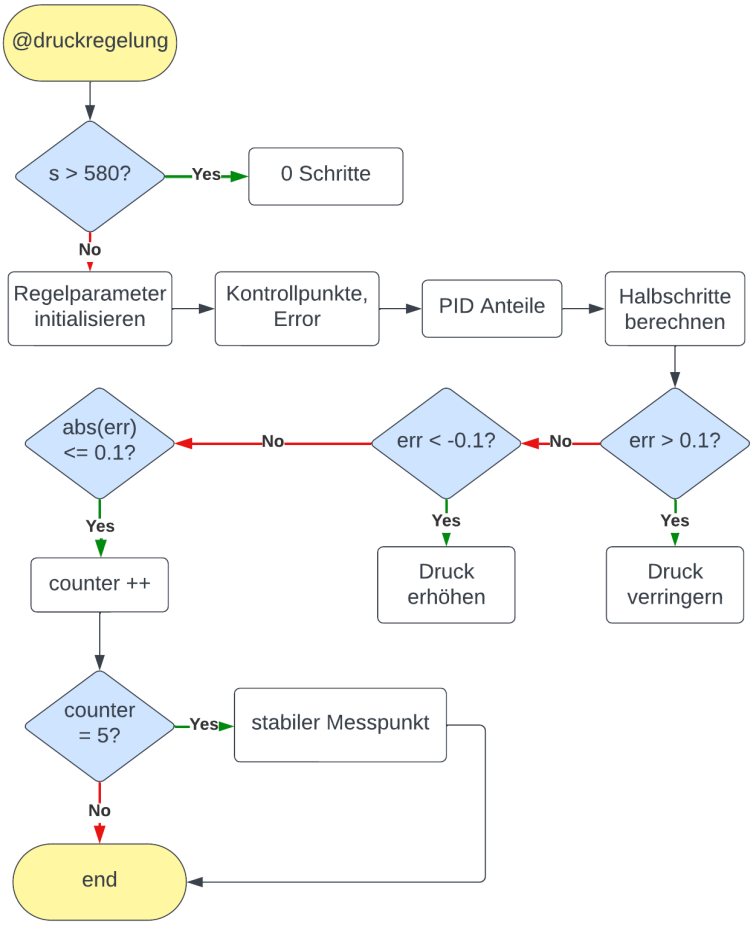


Abbildung 5: Flowchart für die Funktion der Druckregelung

Bei jedem Durchlauf der Funktion wird überprüft, ob die aktuelle Schrittzahl vom Dosierventil 580 Halbschritte erreicht hat. Ist dies der Fall, so wird keine weitere Regelung durchgeführt, da das Dosierventil bereits vollständig geschlossen ist. Wenn die Schrittzahl noch nicht erreicht

ist, wird im ersten Durchlauf der Funktion die Initialisierung der Variablen und der Regelparameter durchgeführt. Hier wird bei Beginn der Kalibrierung der IST-Druck einmalig als Startdruck definiert, welcher für die Berechnung der Kontrollpunkte notwendig ist. Durch Subtraktion des Startdrucks vom Zieldruck wird die maximale Abweichung errechnet. Aus der maximalen Abweichung und der Anzahl der Kontrollpunkte wird der Abstand der Kontrollpunkte ermittelt.

Tabelle 1: Beispielberechnung der Kontrollpunkte mit p_start = 2.2 bar, p_soll = 12 bar

```
anzahl = app.AnzahlKontrollpunkteEditField.Value
# anzahl = 8
z = round((p_soll-p_start)/anzahl,1)
# z = round(1.225,1) -> z = 1.2
punkte = [p_start:z:p_soll,p_soll]
# punkte = [2.2  3.4  4.6  5.8  7  8.2  9.4  10.6  11.8  12]
```

Der Abstand der einzelnen Kontrollpunkte z zueinander errechnet sich durch Division der maximalen Abweichung mit der Anzahl der Kontrollpunkt. Der Abstand wird auf eine Nachkommastelle gerundet. Für die Berechnung der Kontrollpunkte wird ein Vektor *punkte* erstellt. Dieser beginnt mit dem Startdruck und im Abstand z werden die weiteren Punkte bis zum Solldruck erzeugt. Die Berechnung der Punkte geht nicht immer glatt auf, so dass der letzte Wert den Solldruck ergibt. Deswegen wird der Solldruck zusätzlich angefügt. Hier kann es vorkommen, dass bei gerade Aufteilung der Kontrollpunkte der Solldruck zweimal erscheint.

Im nächsten Schritt werden die Verstärkungen für die PID-Anteile definiert. Iteratives Vorgehen hat gezeigt, dass der Regler am besten mit $k_P = 30$, $k_I = 0.5$ und $k_D = 15$ funktioniert.

Damit die zu steuernden Halbschritte des Dosierventils errechnet werden können, sind die PID-Anteile zu ermitteln. Einfachheitshalber werden die Halbschritte im Code *steps* genannt.

$$steps = k_P \cdot P + k_I \cdot I + k_D \cdot D$$

$$P = err$$

$$I = err \cdot dt$$

$$D = \Delta err/dt$$

Der proportionale Anteil P ergibt sich aus der Fehlerabweichung vom Ist zum Soll-Zustand. Hier ist der Sollzustand jedoch nicht der Zieldruck, sondern der Kontrollpunkt. Der Integralanteil ergibt sich aus dem Integral des Fehlers über die Zeit. Der Diffentialanteil D wird durch Derivation des Fehlers nach der Zeit errechnet.

Nach Ermittlung der Schritte muss die aktuelle Fehlerabweichung als alter Fehler definiert werden, um beim nächsten Durchlauf das Fehlerdelta zu berechnen.

Die Anzahl der Schritt ist ermittelt, jedoch nicht die Richtung, in welche das Dosierventil die Schritte fahren muss. Dies wird mithilfe von definierten Toleranzen entschieden. Liegt der Istwert über dem Sollwert, so ist der Fehler größer als 0.1 bar. In dem Fall muss der Druck gesenkt werden, durch öffnen des Dosierventils. Liegt der Istwert unter dem Sollwert ist der Fehler größter als -0.1 bar und das Dosierventil muss etwas geschlossen werden für eine Druckerhöhung. Liegt der Wert innerhalb der Toleranz von 0.1 bar um den Sollwert, so wird eine Countervariable erhöht. Da der Druck keine Verzögerung bei Änderung des Dosierventils zeigt, reicht hier ein kurzer Counter um einen Messpunkt als stabil zu erkennen. Bei Counter = 5 wird der Messpunkt als stabil gekennzeichnet.

Am Ende jedes Durchlaufs der Funktion für die Druckregelung wird die Variable *mess_counter* und der RMS-Buffer zurückgesetzt.

2.2. Temperaturregelung

Die Temperaturregelung wird analog zur der Druckregelung ausgelöst. Hier muss in der ButtonGroup die Temperatur ausgewählt und zusätzlich der Kalibriervorgang gestartet sein. Trifft der Fall zu, so wird hier ein Messcounter aktiviert. Durch den iterativen Prozess in der Entwicklung der Temperaturregelung, löst der Messcounter ebenfalls bei Wert 10 die Temperaturregelung aus. Aufgrund des verzögerten Verhaltens der Temperatur ist ein Wert von 20 ursprünglich definiert. Es hat sich gezeigt, dass das Intervall zu hoch ist und das Heizelement zu lange auf einer Leistungsstufe heizt, was zum Überschreiten der Zieltemperatur führt.

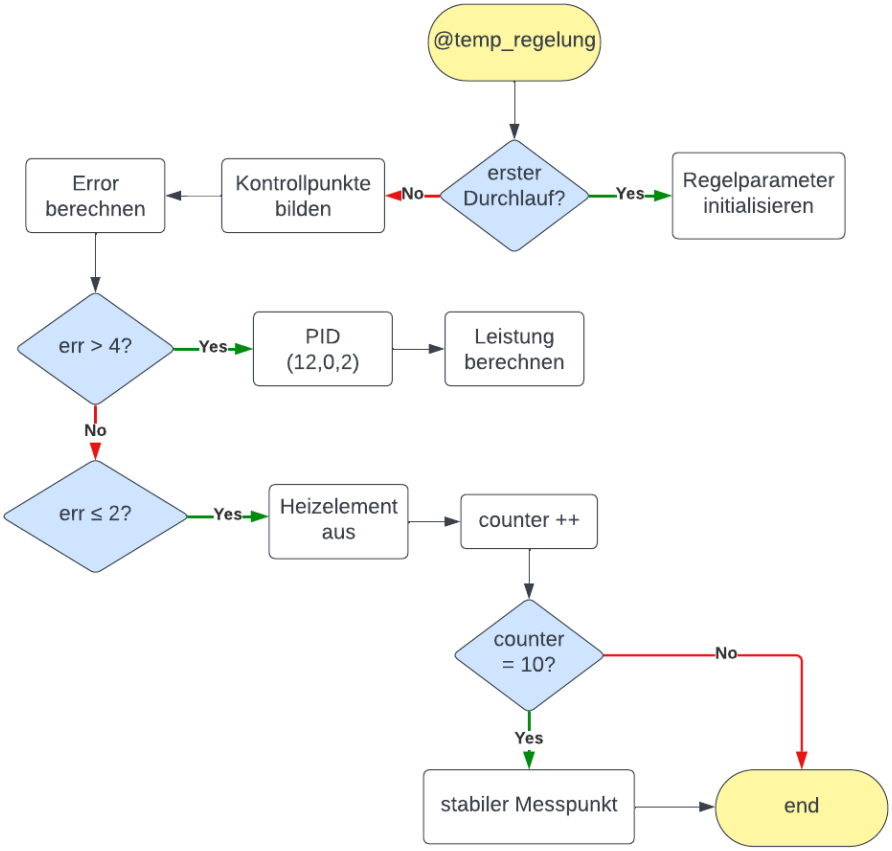


Abbildung 6: Flowchart für die Funktion der Temperaturregelung

In der Datenverarbeitung wird die Temperaturregelung analog zur Druckregelung vorbereitet. Da die Druck- und Temperaturregelung nicht parallel ausgeführt werden, wird bei einer Kalibrierung von Temperatursensoren ebenfalls der RMS-Buffer gebildet.

und Wert der Kontrolltemperatur gebildet. Dieser wird bei Erreichen des Messcounters gleich 10 an die Funktion übergeben.

Zu Beginn der Funktion *druckregelung* wird durch den Einsatz einer Flag geprüft, ob die Regelung das erste Mal aufgerufen wird. Für diesen Fall werden Parameter initialisiert und die Flag wird umgeschaltet, so dass die Initialisierung beim nächsten Aufrufen nicht erneut stattfindet. Anschließend werden die Kontrollpunkte mit einem analogen Vorgehen zur Druckregelung ermittelt.

Im nächsten Schritt wird die Temperaturabweichung zum Solldruck berechnet. Der Solldruck ist hier der erste Kontrollpunkt. Bei der Temperaturregelung lässt sich das Abkühlen des Wassers schwer realisieren, daher wird diese so aufgebaut, dass an die Solltemperatur langsam angeschmiegt wird. Dies wird durch zwei Fälle erreicht. Für den ersten Fall, dass der Error mindestens 4 °C beträgt, sind die PID-Parameter $k_P = 12$, $k_I = 0$, $k_D = 2$. Anhand dieser Parameter wird aus der PID-Gleichung die prozentuale Heizleistung berechnet mit einer oberen Grenze bei 100 %.

$$heizleistung = k_P \cdot P + k_I \cdot I + k_D \cdot D$$

Die PID-Anteile werden analog zur Druckregelung berechnet, bezogen auf die Fehlerabweichung in der Temperatur. Mit zunehmendem Anschmieden der Ist-Temperatur an den Sollwert, verringert sich die Fehlerabweichung. Ist die Fehlerabweichung maximal 2 °C, wird das Heizelement ausgeschalten. Die Restwärme führt dazu, dass die Temperatur weiter steigt. Durch Ausschalten des Elements hält sich die Temperatur für ca. 30 Sekunden stabil. Weiter wird bei Erreichen der 2 °C Grenze ein Counter ausgelöst, welcher bei Wert 10 den Messpunkt als stabil definiert.

Im letzten Schritt des Durchlaufs der Regelfunktion wird der Messcounter für das Aktivieren der Regelungsfunktion zurückgesetzt und der RMS-Buffer wird geleert.

Literatur

Quellen Nummerierung ist dem Institut für Kaffeetechnologie zu entnehmen auf:

[95] Patricia Viebke, Finalisierung der Inbetriebnahme des Pumpenprüfstands, Projektarbeit, 2023