4.6 Zusatzaufgaben

4.6.1 Reale Implementierung der 3-Tank Regelungsstrategie

Testen Sie den entworfenen Regler am realen 3-Tank System. Auf den Echtzeitrechner können Sie Ihre Embedded MATLAB Function gemäß Abbildung 4.2 sowie ihre Parameter `struct`-Objekte `parSys` und `parInit` übertragen. Kopieren Sie dazu die Embedded MATLAB Function in ein eigenes SIMULINK- Model und speichern Sie dieses unter dem Namen `Regler_XX.slx` ab. Beachten Sie gegebenenfalls die Abwärts-Kompatibilität zur MATLAB Version `2014b`. Speichern Sie des Weiteren die beiden `struct`-Objekte in einer Datei mit dem Namen `Parameter_XX.mat` mithilfe der `save` Funktion. XX sei dabei Ihre Gruppennummer.

4.6.2 Trajektorienplanung 2-Tank

Das in Aufgabe 4.6 entworfene Sollwertfilter wurde konservativ ausgelegt um die Stellgrößenbegrenzungen nicht zu verletzen. Die daraus resultierenden Trajektorien führen dazu, dass das geschlossene System nicht optimal betrieben werden kann. Daher soll in dieser Aufgabe dieses Sollwertfilter mit einem *Rate-Limiter* kombiniert werden, um die mögliche Dynamik des Systems besser auszunutzen. In einem ersten Schritt soll die Wirkung der Stellgrößenbeschränkung auf die möglichen Änderungsraten der Füllhöhen untersucht und anschließend eine geeignete Trajektorienplanung umgesetzt werden.

Aufgabe 4.13 (Trajektorienplanung).

- Die maximale Anstiegsgeschwindigkeit der Füllhöhe des zweiten Tanks $\dot{h}_{2,max}$ kann einfach anhand der Simulation abgeschätzt werden. Das System wird dazu, ausgehend von der Ruhelage $h_{2,R} = 0.1\,\mathrm{m}$, mit dem maximalen Zufluss $q_{Z1} = 4.5\,\mathrm{l/min}$ beaufschlagt. Ermitteln Sie anhand der Steigung des zeitlichen Verlaufes von h_2 näherungsweise $\dot{h}_{2,max}$.

- Die maximale Sinkgeschwindigkeit der Füllhöhe des zweiten Tanks $\dot{h}_{2,min}$ kann analog zur Anstiegsgeschwindigkeit abgeschätzt werden. Ausgehend von der Ruhelage $h_{2,R} = 0.3\,\mathrm{m}$ wird das System mit dem minimalen Zufluss $q_{Z1} = 0\,\mathrm{l/min}$ betrieben. Bestimmen Sie $\dot{h}_{2,min}$ näherungsweise anhand der Steigung des zeitlichen Verlaufes von h_2 .

- Basierend auf den ermittelten maximalen Anstiegs- und Sinkgeschwindigkeiten des Füllstands h_2 können Sie die Änderungsrate der Sollvorgabe mittels einem *Rate-Limiter-Block* begrenzen. Mithilfe eines darauf folgenden Sollwertfilters wird wiederum die hinreichend glatte Solltrajektorie mit den entsprechenden Zeitableitungen generiert.
  Passen Sie die Pole dieses Sollwertfilters in Kombination mit dem *Rate-Limiter* so an, dass für den Arbeitspunktwechsel $\Delta h_2(t) = 0.1\,\mathrm{m}(\sigma(t-1) - \sigma(t-500))$ der absolute Regelfehler stets kleiner $0.2\,\mathrm{mm}$ ist. Vergleichen Sie die Ergebnisse mit dem in Aufgabe 4.6 entworfenen Sollwertfilter.

Stellgröße bleibt länger nahe der Beschränkung
-> schneller

Hinweis:

- Führen Sie alle Simulation mit dem Regler ohne Integralanteil, deaktivierten Rauschen und aktiver Stellgrößenbeschränkung des Models durch.
- Ignorieren Sie für diese Aufgaben etwaige Verletzungen der Stellgrößenbeschränkungen.

4.6.3 Anti-Windup

In Aufgabe 4.12 wurde eine einfache Anti-Windup-Strategie für das 3-Tank-Modell entworfen. Hierbei werden beide Integratoren eingefroren, sobald eine der Stellgrößenbeschränkung aktiv ist. Diese Vorgehensweise stellt nur eine Möglichkeit dar, wie der Integrator bei erreichen der Stellgröße behandelt wird. In der folgenden Aufgabe soll die Notwendigkeit und der Einfluss von unterschiedlichen Anti-Windup Maßnahmen veranschaulicht werden. Dies erfolgt anhand des SISO Systems $G(s)$

$$G(s) = \frac{s+1}{s^2 - 0.5s + 1} \quad \text{-> instabil} \tag{4.19}$$

welches mit dem PI-Regler $R(s)$

$$R(s) = 5\left(1 + \frac{1}{s}\right) \tag{4.20}$$

stabilisiert wird. Das Systemverhalten wird im Folgenden anhand des Referenzsignals $r = 0.9\sigma(t - 0.1)$ analysiert. Des weiteren gilt für die Stellgröße die Beschränkung $u_{\min} \leq u \leq u_{\max}$ mit $u_{\min} = 0$ und $u_{\min} = 1$.

Aufgabe 4.14 (Anti-Windup). Anhand des Systems (4.19) mit dem Regler (4.20) wird die Notwendigkeit und der Einfluss von Anti-Windup Maßnahmen veranschaulicht. Bearbeiten Sie dafür die folgenden Punkte:

1. Verifizieren Sie das Systemverhalten ohne als auch mit aktiver Stellgrößenbeschränkung. Wie verändert sich das Systemverhalten aufgrund der Stellgrößenbeschränkung? kann nicht mehr stabilisiert werden
2. Implementieren Sie eine Anti-Windup Strategie bei der der Integrator eingefroren wird sobald die Stellgrößenbeschränkung verletzt wird, vgl. Aufgabe 4.12.
3. Eine weitere Anti-Windup Strategie ist das dynamische Rücksetzen des Integralanteils in Abhängigkeit der Stellgrößenbeschränkung. Geben Sie dazu das Regelgesetz als Funktion des Regelfehlers $e(t)$ an und spalten Sie die Stellgröße $u(t)$ in einen Integralanteil $u_I(t)$ und einen Proportionalanteil $u_P(t)$ auf. Beim dynamischen Anti-Windup-Reset wird der integrale Fehler $e_I(t) = \int e(\tau)\mathrm{d}\tau$ überschrieben, sobald die berechnete Stellgröße die Beschränkung verletzt. Implementieren Sie in einer Embedded MATLAB Function diese Anti-Windup Strategie und berücksichtigen Sie folgende Punkte:

- Beim Überschreiten der oberen Schranke $u_{\max}$ muss $e_I(t)$ so gesetzt werden, dass $u(t) = u_P(t) + u_I(t) = u_{\max}$ gilt.
- Beim Unterschreiten der unteren Schranke $u_{\min}$ muss $e_I(t)$ so gesetzt werden, dass $u(t) = u_P(t) + u_I(t) = u_{\min}$ gilt.
- Andernfalls gilt $e_I(t) = \int_{t_1}^{t} e(\tau)\mathrm{d}\tau + e_I(t_1)$, wobei t_1 jenen Zeitpunkt beschreibt ab der die Stellgröße wieder innerhalb der Beschränkung liegt.

Hinweis:

- Wenn Sie den Regler als zeitdiskrete Funktion implementieren ($T_a = 1\,\mathrm{ms}$), können die einzelnen Regelanteile einfach beeinflusst werden. Die Integration kann dabei über ein explizites Euler-Verfahren erfolgen.
- Für einen direkten Vergleich der Anti-Windup Strategien stellen Sie die Ausgangsgröße aller betrachteten Fälle in einer gemeinsamen Darstellung dar.
- Zur Überprüfung der korrekten Implementierung betrachten Sie den integralen Fehler und die Stellgröße in einer gemeinsamen Darstellung.

4.6.4 Elektrisches Netzwerk

In weiterer Folge soll ein elektrisches Netzwerk gemäß Abbildung 4.3 betrachtet werden. Dieses dynamische System vierter Ordnung ist linear und zeitinvariant und kann daher mit dem Zustand $\mathbf{x} = \begin{bmatrix} i_{L_1} & u_{C_1} & i_{L_2} & u_{C_2} \end{bmatrix}^{\mathrm{T}}$ in der Form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \tag{4.21}$$

mit

$$\mathbf{A} = \begin{bmatrix} -\frac{R_1}{L_1} & -\frac{1}{L_1} & 0 & 0 \\ \frac{1}{C_1} & 0 & -\frac{1}{C_1} & 0 \\ 0 & \frac{1}{L_2} & -\frac{R_2}{L_2} & -\frac{1}{L_2} \\ 0 & 0 & \frac{1}{C_2} & -\frac{1}{C_2 R_L} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

dargestellt werden. Die zugehörigen Parameterwerte sind in Tabelle 4.2 angegeben.

Aufgabe 4.15 (Eingangs-Ausgangslinearisierung). In dieser Aufgabe soll der Strom i_{L_2} geregelt werden. Führen Sie dazu eine Eingangs-Ausgangslinearisierung durch. Dabei sind folgende Punkte zu bearbeiten:

1. Bestimmen Sie den relativen Grad r des Systems. r=3
2. Bestimmen Sie eine mögliche Transformationsmatrix $\mathbf{T}$, die das System in Byrnes-Isidori-Normalform transformiert und geben Sie das transformierte System an.
3. Ist die auftretende Nulldynamik stabil?

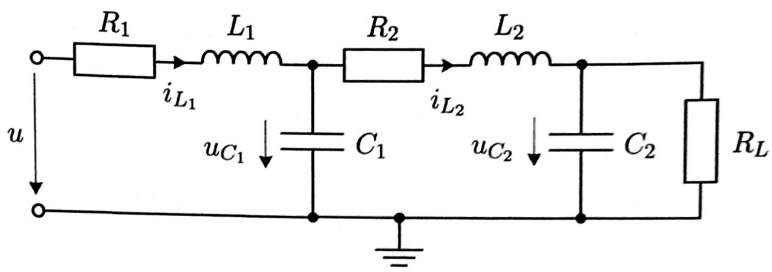

Abbildung 4.3: Elektrisches Netzwerk.

Größe	Wert	Einheit
L_1	0.3	H
L_2	0.1	H
C_1	0.1	F
C_2	0.2	F
R_1	0.5	Ω
R_2	0.1	Ω
R_L	500	Ω

Tabelle 4.2: Parameter des elektrischen Netzwerks.

4. Implementieren Sie die Eingangs-Ausgangslinearisierung in einer zeitdiskreten Embedded MATLAB Function. Wählen Sie hierbei eine geeignete Abtastzeit gemäß der Systemdynamik und achten Sie auf sinnvolle Solltrajektorien.

Hinweis:

- Wählen Sie die freien Einträge der Transformation so, dass die Nulldynamik unabhängig vom Eingang u wird.
- Setzen Sie das Regelgesetz aus den einzelnen Teilen der Systemmatrizen $\mathbf{A}$, $\mathbf{b}$ und $\mathbf{c}$ zusammen.
- Nutzen Sie den Trajektoriengenerator aus Aufgabe 4.4 zur Erzeugung hinreichend glatter Solltrajektorien.
- Sollten Schwierigkeiten beim Erreichen des stationären Endwertes auftreten, verringern Sie die Abtastzeit des Trajektoriengenerators und des Reglers bzw. implementieren Sie letztgenannten als zeitkontinuierliche Funktion.

Aufgabe 4.16 (Eingangs-Zustandslinearisierung). Für das elektrische Netzwerk (4.21) soll nun eine Eingangs-Zustandslinearisierung durchgeführt werden. Bearbeiten Sie dazu folgende Punkte:

1. Finden Sie eine mögliche Parametrierung $y = \mathbf{c}^{\mathrm{T}}\mathbf{x}$, sodass das System (4.21) vollständig eingangs-zustandslinearisierbar ist.
2. Implementieren Sie die Eingangs-Zustandslinearisierung in einer zeitdiskreten Embedded MATLAB Function

Hinweis: Aufgrund der Ähnlichkeit zur Eingangs-Ausgangslinearisierung können Sie auf den Quellcode aus Aufgabe 4.15 zurückgreifen.