Name: Valentin Nimmervoll Matr.-Nr.: 11811502

Angaben zur Matlab Version:

Die Hausübung wurde von mir allein, mit Matlab R2022a ausgearbeitet.

Aufgabe 1:

Die in der ersten Aufgabe geforderte Funktion myisimpl_extended() wurde auf Basis des gegebenen Flussdiagramms implementiert. Die Abfragen im Flussdiagramm wurden mittels If und Else im Code implementiert.

Zur Überprüfung der BIBO Stabilität von T(s) wurde die Funktion myisstable aus der Übung verwendet.

Die zweite Abfrage in der Funktion ist das Prüfen ob alle instabile Nullstellen von P(s) Nullstellen von T(s) sind. Dazu werden alle Nullstellen mit folgender Abfrage geprüft:

```
real(nu(i)) > 0) && ~ismember(nu(i), nut)
```

Falls dies nicht erfüllt ist wird T(s) modifiziert und die Fehlenden Nullstellen hinzugefügt.

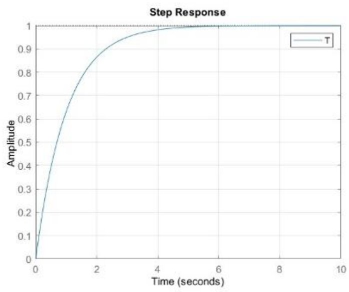
Bei der letzten Abfrage wird kontrolliert, ob der Polüberschuss von T(s) ausreichend groß ist. Ansonsten wird T(s) für den nötigen PÜ modifiziert.

Abschließend werden abhängig von T(s) die Flags isimp_ext_flag und modify_flag sowie das Resultierende T(s) zurück übergeben, sowie im Falle von Implementierbarkeit die Sprungantworten von T(s) und falls vorhanden T(s) ausgegeben.

- Testbeispiel 1

$$P(s) = \frac{s+3}{s(s-1)} \text{ und } T(s) = \frac{1}{s+1}$$

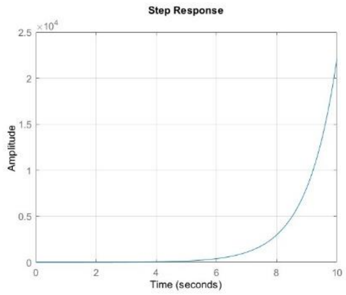
```
isimp_ext_flag = 1
modify_flag = 0
Tresult = 1/(s+1)
```



- Testbeispiel 2

$$P(s) = \frac{s+3}{s(s-1)} \text{ und } T(s) = \frac{1}{s-1}$$

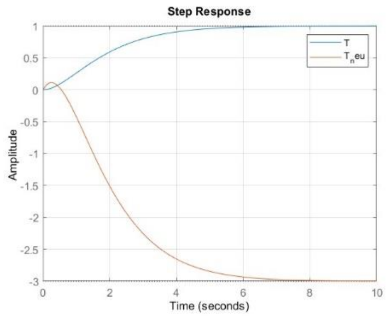
```
isimp_ext_flag = 0
modify_flag = 0
Tresult = []
```



- Testbeispiel 3

$$P(s) = \frac{(s-3)}{s(s-1)} \text{ und } T(s) = \frac{1}{(s+1)^2}$$

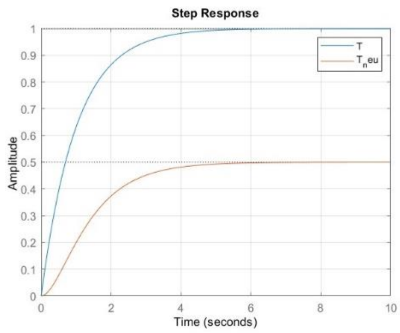
```
isimp_ext_flag = 1
modify_flag = 1
Tresult = (s-3)/((s+1)*(s+1))
```



- Testbeispiel 4

$$P(s) = \frac{1}{s(s-1)} \text{ und } T(s) = \frac{1}{(s+1)}$$

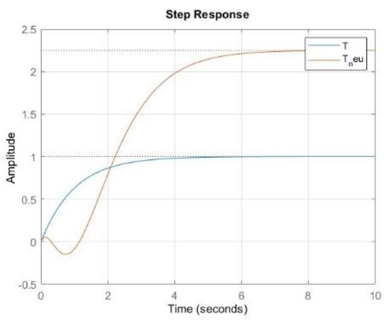
```
isimp_ext_flag = 1
modify_flag = 1
Tresult = 1/((s+1)*(s+2))
```



- Testbeispiel 5

$$P(s) = \frac{(s-3)^2}{s(s-1)^2} \text{ und } T(s) = \frac{1}{(s+1)}$$

```
isimp_ext_flag = 1
modify_flag = 1
Tresult = ((s-3)*(s-3))/((s+1)*(s+2)*(s+2))
```



Aufgabe 2

In der zweiten Aufgabe sollte für eine gegebene Strecke P(s) ein Standardregelkreis mittels Polvorgabe ermittelt werden, bei der eine gegebene harmonische Störung unterdrückt werden kann.

Dazu wurde auf Basis der Funktion polvorgabe aus der Übung die Funktion polvorgabe_HStörung programmiert. Diese führt folgende Berechnungsschritte durch:

1. Ermittlung der Regler Ordnung
2. Überprüfung, ob $\nu_T(s)$ konsistent ist
3. Entwurf des Reglers

Die Einwirkung einer Störung auf das geschlossene System wird durch folgende Übertragungsfunktion beschrieben:

$$S(s) = \frac{1}{1 + R(s) * P(s)}$$

Die gegebene harmonische Cosinus Störfunktion muss nun in den Bildbereich transformiert werden.

$$\mathcal{L}\{\mathrm{d(t)} = \cos(\omega * \mathrm{t})\} = \mathrm{d(s)} = \frac{\mathrm{s}}{\omega^2 + \mathrm{s}^2}$$

Um die Störung zu unterdrücken, wird nun der Beitrag der Störung am Endwert null gesetzt.

$$\lim_{s \to 0} S(s) d(s) = \lim_{s \to 0} \frac{\nu(s) * a(s)}{\nu(s) * a(s) + \mu(s) * b(s)} \frac{s}{\omega^2 + s^2} \overset{!}{\Leftrightarrow} 0$$

Da Omega ungleich null ist, und $\nu(s)$ durch die Strecke gegeben ist, muss $a(s) = 0$ gelten.

Dies wird durch folgende Zusatz Gleichungen gewährleistet:

$$a(s = j\omega) = \sum_{i}^{\rho} (j * \omega)^i * a_i \overset{!}{\Leftrightarrow} 0$$

Da $(j * \omega)^i$ eine Periode von 4 besitzt ergeben sich beim Koeffizienten vergleich zwei zusätzliche Bedingungen an den Regler.

$$\begin{bmatrix} \omega^0 & 0 & -\omega^1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \omega^1 & 0 & -\omega^2 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_\rho \\ b_0 \\ b_1 \\ \vdots \\ b_\rho \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Als Vereinfachung kann noch die zweite Zeile durch ω dividiert werden.

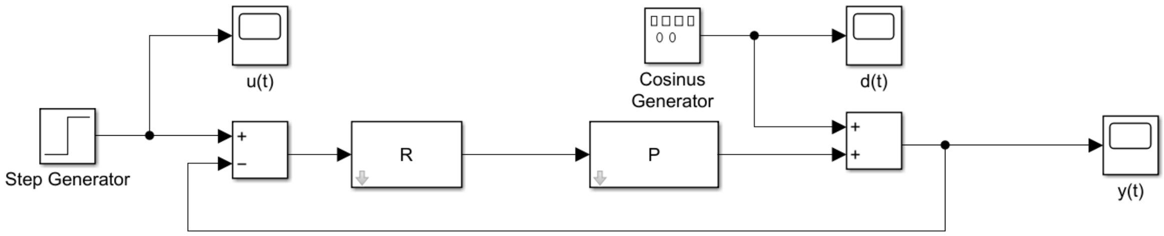
$$\begin{bmatrix} \omega^0 & 0 & -\omega^1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \omega^0 & 0 & -\omega^1 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_\rho \\ b_0 \\ b_1 \\ \vdots \\ b_\rho \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Diese zwei Zusatzbedingungen werden an die Resultante angehängt und erfordern eine Regler Ordnung von $\rho = n + 1$. Für die Implementierung in Matlab wurden zwei for Schleifen verwendet worin mit einer if Abfrage durch Überprüfen, ob der Index gerade oder ungerade ist jeder Eintrag beschrieben. Für das Vorzeichen der Koeffizienten wurde die Periodizität von $(i)^n$ ausgenutzt.

Erfüllt das gegebene Nenner Polynom v_t dies nicht wird es mit zusätzlichen Nullstellen bei s = -4 erweitert.

Die Restliche Berechnung der Polvorgabe gleicht, der in der Übung erstellten, einfachen Polvorgabe.

Zur Simulation wurde ein Simulink Modell erstellt.



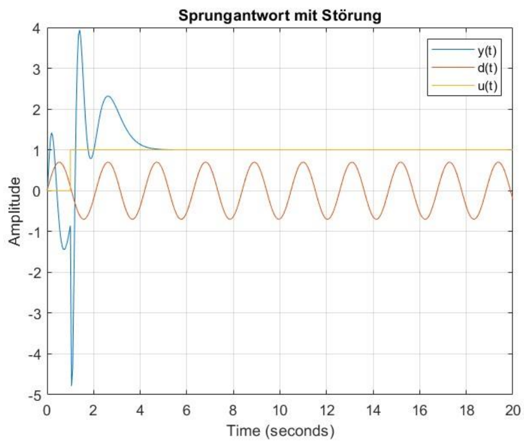
Hierbei wurden der Regler R und die Strecke P aus dem Matlab Workspace übernommen und die Ergebnisse der Simulink für u(t), d(t) und y(t) mittels Scopes als Variablen in den Matlab Workspace zur Weiterverarbeitung zurück übergeben.

In Matlab werden daraus dann plots generiert.

<u>Simulationsergebnisse der geforderten Wunschpolynome:</u>

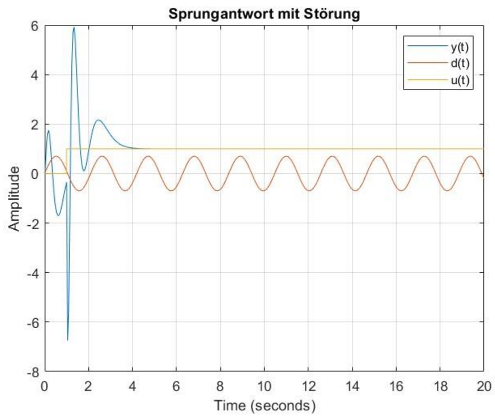
$v_{t1}(s) = s + 5$

Das Polynom $v_{t1}(s)$ wurde bei der Berechnung mehrfach erweitert.



$v_{t2}(s) = (s + 5)^2$

Das Polynom $v_{t2}(s)$ wurde bei der Berechnung mehrfach erweitert.



$v_{t3}(s) = (s + 5)^5$

Es konnte ohne Modifikation ein Regler mit $v_{t3}(s)$ entworfen werden.