Aufgabe 1: Überprüfung auf Implementierbarkeit

Erstellen Sie eine Matlab-Funktion, die bei gegebener Strecke $P(s)$ eine gewählte Führungsübertragungsfunktion $T(s)$ hinsichtlich ihrer *Implementierbarkeit* bewertet. Der Funktionskopf soll dabei folgenden Aufbau haben

```
function flag = myisimp(T,P).
```

Hierbei sind `P` und `T` Übertragungsfunktionen (Matlab-Datentyp `tf`), für den Rückgabeparameter `flag` soll gelten:

`flag=0`	T ist nicht implementierbar,
`flag=1`	T ist implementierbar.

Die Routine soll auf zwei selbstgeschriebene Funktionen

```
function flag  = myisproper(G)
function flag  = myisstable(G)
```

zurückgreifen. Diese überprüfen zeit-kontinuierliche Übertragungsfunktionen auf *Realisierbarkeit* (`myisproper`) und kontrollieren, ob ihr Nennerpolynom „stabil" ist (`myisstable`). Testen Sie Ihre Funktion anhand des folgenden Beispiels.

Testbeispiel: $$P(s) = \frac{(s+2)(s-1)}{s\,(s^2+2s+1)}$$

$$i)\quad T(s) = 1 \qquad ii)\quad T(s) = \frac{s+2}{(s+1)(s+3)} \qquad iii)\quad T(s) = \frac{s-1}{(s+1)^2(s+3)}$$

Zur Erinnerung: $T(s)$ ist genau dann implementierbar, wenn die Übertragungsfunktionen, $T(s)$ und $T_u(s) = \dfrac{T(s)}{P(s)}$ beide BIBO stabil und realisierbar sind.

Aufgabe 2: Automatische Erzeugung der Resultante

Es soll eine Matlab-Funktion

```
Res=resultante(nu,mu,rho)
```

erstellt werden, die aus den Polynomen `nu` und `mu` und der natürlichen Zahl `rho` die Resultante generiert

$$\begin{bmatrix} \nu_0 & 0 & \dots & 0 & \mu_0 & 0 & \dots & 0 \\ \nu_1 & \nu_0 & \ddots & 0 & \mu_1 & \mu_0 & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots & \vdots & \ddots & 0 \\ \nu_{n-1} & \nu_{n-2} & \dots & \nu_0 & \mu_{n-1} & \mu_{n-2} & \dots & \mu_0 \\ \nu_n & \nu_{n-1} & \dots & \nu_1 & 0 & \mu_{n-1} & \ddots & \vdots \\ 0 & \nu_n & \dots & \nu_2 & 0 & 0 & \ddots & \mu_{n-2} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \mu_{n-1} \\ 0 & \dots & 0 & \nu_n & 0 & 0 & \dots & 0 \end{bmatrix}.$$

Für die beiden Polynome gilt hierbei

$$\begin{aligned}\nu(s) &= \nu_n s^n + \nu_{n-1} s^{n-1} + \ldots + \nu_1 s + \nu_0, \\ \mu(s) &= \mu_m s^m + \ldots + \mu_1 s + \mu_0 \qquad \text{mit} \qquad m \leq n.\end{aligned}$$

Zur Erinnerung: Die Resultante hat die Dimension $(n + \rho + 1) \times (2\rho + 2)$.

Aufgabe 3: Analytische Synthese für den Standardregelkreis - "Polvorgabe"

Der einfachste algebraische Entwurf für den *Standardregelkreis* (d.h. $\rho = n - 1$) soll implementiert werden. Die entsprechende Matlab-Funktion soll folgenden Funktionskopf besitzen:

```
[R,T]=polvorgabe(P,nut)
```

Hierbei ist **`P`** die Streckenübertragungsfunktion (Datentyp **`tf`**), **`nut`** ist das gewünschte Nennerpolynom der Führungsübertragungsfunktion des Regelkreises. Die Variable **`R`** ist die gesuchte Reglerübertragungsfunktion, **`T`** ist die resultierende Führungsübertragungsfunktion. Die Funktion soll folgendermaßen aufgebaut sein:

1. Aufstellen der Resultante mittels der Funktion **`resultante`** (siehe Beispiel 2) und Lösung des zugehörigen linearen Gleichungssystems.
2. Ermittlung von **`R`** und **`T`**.

Testbeispiel: Lösen Sie bei gegebener Strecke

$$P(s) = \frac{s-2}{s\,(s-1)}$$

und einem gewählten Polynom

$$\nu_T(s) = s^3 + 3s^2 + 3s + 1$$

das Polvorgabeproblem. Wie lautet die Führungsübertragungsfunktion $T(s)$? Zeichnen Sie die Sprungantwort des Regelkreises.

Relevante Matlab-Befehle

- `eye`
- `fliplr`
- `flipud`
- `for`
- `help`
- `length`
- `minreal`
- `order`
- `real`
- `roots`
- `step`
- `sum`
- `tf`
- `tfdata`
- `toeplitz`
- `zeros`
- `zpk`
- `zpkdata`