

1 Identifikation

Ziel dieser Übung ist es, nicht-parametrische und parametrische Verfahren zur Identifikation linearer dynamischer Systeme zu vertiefen. Untersucht wird dabei ein Einmassenschwinger, welcher zunächst mit Hilfe der Frequenzbereichsanalyse (ETFE) identifiziert werden soll. Im zweiten Teil dieser Übung wird das Least-Squares Verfahren zur direkten Parameterschätzung des Einmassenschwingers angewandt. Dabei werden zwei Wege zur Formulierung des Schätzproblems aufgezeigt und erarbeitet. Schlussendlich werden mit Hilfe rekursiver Verfahren die Parameter einer zeitveränderlichen Strecke geschätzt.

Es wird davon ausgegangen, dass Sie das folgende Skript beherrschen:

- Skriptum zur VU Regelungssysteme 1 (WS 2019/2020) [1]
 - Kapitel 1



Alle Dateien, die zum Bearbeiten dieser Übung benötigt werden, finden Sie in `uebung1.zip` auf der Homepage der Lehrveranstaltung.



Bei Fragen oder Anregungen zu dieser Übung wenden Sie sich bitte an

- René Lenz <lenz@acin.tuwien.ac.at> oder
- Julian Landauer <landauer@acin.tuwien.ac.at>.

1.1 Nicht-parametrische Identifikation

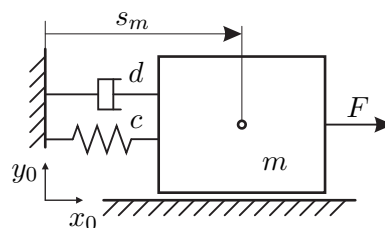


Abbildung 1.1: Translatorischer Einmassenschwinger.

Im Folgenden wird der in Abbildung 1.1 skizzierte translatorische Einmassenschwinger betrachtet. Der Starrkörper mit der Masse m gleitet in x_0 -Richtung reibungsfrei auf einer Unterlage und ist über eine lineare Feder mit der Federsteifigkeit c und einem viskosen

Dämpfer mit dem Dämpfungskoeffizienten d mit einer Wand verbunden. Die Position der Masse wird mit der Koordinate s_m bezeichnet. Die Geschwindigkeit wird mit $v_m = \dot{s}_m$ benannt und der Stelleingang u entspricht der Kraft F . Als Ausgang y wird die Position s_m gewählt.

Erarbeiten Sie die folgenden Aufgaben zur Frequenzbereichsanalyse in MAPLE bzw. in einem m-File mit MATLAB.

Aufgabe 1.1 (Modellierung).

1. Berechnen Sie das mathematische Modell und die zeitkontinuierliche Übertragungsfunktion $G(s)$ vom Eingang $u = F$ zum Ausgang $y = s_m$ des Einmassenschwingers nach Abbildung 1.1. Diskretisieren Sie das Streckenmodell exakt mit einer Abtastzeit $T_a = 25$ ms unter Verwendung eines Haltegliedes nullter Ordnung (Zero-Order-Hold). Geben Sie sowohl die analytische Lösung in MAPLE wie auch die numerische Lösung in MATLAB an. Verwenden Sie hierzu die in Tabelle 1.1 angegebenen Parameter. Bestimmen Sie weiterhin die Resonanzfrequenz ω_0 bzw. f_0 des Einmassenschwingers für $d = 0$.
2. Berechnen Sie die Frequenzauflösung $\Delta\omega$ bzw. Δf und die höchste messbare Frequenz ω_{max} bzw. f_{max} für $N = 4096$ Messwerte und $T_a = 25$ ms.

| Bezeichnung | Variable | Wert |
|-------------|----------|-----------|
| Masse | m | 0.5 kg |
| Dämpfung | d | 0.7 N s/m |
| Steifigkeit | c | 100.0 N/m |

Tabelle 1.1: Nominelle Parameter des Einmassenschwingers.

Für die nachfolgenden Identifikationsaufgaben werden zwei Testsignale verwendet:

1. *Chirp-Signal mit trapezförmiger Fensterung*: Ein Chirp-Signal mit trapezförmiger Fensterung kann durch

$$u_k = U_0 + r_k \sin\left(\omega_s k T_a + \left(\frac{\omega_e - \omega_s}{N T_a}\right) \frac{(k T_a)^2}{2}\right) \quad (1.1a)$$

$$r_k = \hat{U} \text{sat}\left(\frac{10k}{N}\right) \text{sat}\left(\frac{10(N-k)}{N}\right) \quad (1.1b)$$

mit

$$\text{sat}(x) = \begin{cases} 1 & \text{für } x \geq 1 \\ x & \text{für } -1 < x < 1 \\ -1 & \text{für } x \leq -1 \end{cases} \quad (1.2)$$

und $k = 0, 1, \dots, N-1$, berechnet werden. Darin bezeichnet T_a die Abtastzeit, N die Anzahl der Abtastpunkte, \hat{U} die Amplitude, ω_s die Startfrequenz und ω_e die

Endfrequenz des Chirpsignals sowie U_0 ein Offset zur Kompensation des Signalmittelwerts.

2. *PRBS-Signal mit Überabtastung*: Sogenannte PRBS (Pseudo Random Binary Signal) Verläufe werden häufig in Identifikationsaufgaben verwendet, da sie ähnliche Eigenschaften wie weißes Rauschen aufweisen. Ein PRBS-Signal der Ordnung O_p kann durch die Differenzengleichung

$$p_k = \text{mod}(a_1 p_{k-1} + a_2 p_{k-2} + \dots + a_{O_p} p_{k-O_p}, 2) \quad (1.3)$$

mit den Koeffizienten $a_j \in \{0, 1\}$, $j = 1, \dots, O_p$, welche für unterschiedliche Ordnungen in Tabelle 1.2 dargestellt sind, ermittelt werden. Man beachte, dass sich das PRBS-Signal alle $2^{O_p} - 1$ Abtastschritte wiederholt.

| Ordnung O_p | $a_j \neq 0$ für folgende j |
|---------------|-------------------------------|
| 2 | 1, 2 |
| 3 | 2, 3 |
| 4 | 1, 4 |
| 5 | 2, 5 |
| 6 | 1, 6 |
| 7 | 3, 7 |
| 8 | 1, 2, 7, 8 |
| 9 | 4, 9 |
| 10 | 7, 10 |
| 11 | 9, 11 |

Tabelle 1.2: Koeffizienten des PRBS-Signals.

Um die Frequenzeigenschaften des PRBS-Signals zu beeinflussen ist häufig eine Überabtastung des Signals p_k mit dem Faktor $P_p \geq 1$ sinnvoll. Ist eine Abtastzeit T_a gegeben, dann werden die Werte von p_k entsprechend P_p -mal aufgeschaltet, d. h. es gilt

$$u((P_p k + j)T_a) = 2\hat{U}p_k, \quad j = 0, \dots, P_p - 1, \quad k = 0, \dots, 2^{O_p} - 2. \quad (1.4)$$

Es kann gezeigt werden, dass damit im Frequenzbereich eine Tiefpassfilterung des PRBS-Signals erfolgt. In der Literatur wird typischerweise eine Überabtastung von $P_p = 4$ empfohlen. Bei einer Ordnung O_p und einer Überabtastung P_p steht damit eine Anzahl von $N = (2^{O_p} - 1)P_p$ Signalwerten zur Verfügung.

Aufgabe 1.2 (Testsignale).

1. Implementieren Sie das Chirp-Signal nach (1.1) und (1.2) in Form einer **function** in MATLAB. Die Funktion soll als Parameter die Amplitude \hat{U} , die Startfrequenz ω_s , die Endfrequenz ω_e , die Anzahl der Abtastpunkte N sowie die Abtastzeit T_a bekommen. Als Rückgabewerte sollen der Zeitvektor \mathbf{t} sowie der Signalvektor \mathbf{u} ermittelt werden. Die Konstante U_0 soll dabei so gewählt werden, dass der Signalvektor \mathbf{u} mittelwertfrei ist.
2. Implementieren Sie das PRBS-Signal nach (1.3) und (1.4) in Form einer **function** in MATLAB für eine Ordnung $O_p = 10$ sowie einer frei vorgebbaren Überabtastung $P_p \geq 1$. Verwenden Sie für die Initialisierung des PRBS-Signals den Vektor $[p_{-O_p} \ p_{-O_p+1} \ \dots \ p_{-1}] = [0 \ 0 \ \dots \ 1]$. Die Funktion soll als Parameter die Amplitude \hat{U} , die Überabtastung P_p sowie die Abtastzeit T_a bekommen und den Zeitvektor \mathbf{t} sowie den mittelwertfreien Signalvektor \mathbf{u} ermitteln.
3. Stellen Sie den Zeitverlauf des Chirp-Signals für $N = 4096$, $\omega_s = 2\pi 1 \text{ rad/s}$, $\omega_e = 2\pi 10 \text{ rad/s}$ sowie die Zeitverläufe des PRBS-Signals für $P_p = 4$ und $P_p = 1$ dar. Verwenden Sie dazu eine Amplitude von $\hat{U} = 10 \text{ N}$ ($[\hat{U}] = \text{N}$) und eine Abtastzeit $T_a = 25 \text{ ms}$. Vergleichen Sie weiterhin die Amplitudenspektren der Signale.

Hinweis:

- Wählen Sie für die korrekte Darstellung der Zeitverläufe den MATLAB-Befehl **stairs**.
- Achten Sie bei der Darstellung der Frequenzgänge auf eine richtige Skalierung der Frequenzachse und stellen Sie nur den physikalisch sinnvollen Bereich dar. Um einen sinnvollen Vergleich zweier Signale zu erlauben, müssen beide Signale normiert über die Anzahl der Messpunkte in einer gemeinsamen Abbildung dargestellt werden. Beachten Sie weiterhin die in der Regelungstechnik übliche Darstellung von Betrags- und Phasengang in logarithmischen Achsen über die Kreisfrequenz ω und verwenden Sie für den Betrag eine Darstellung in dB sowie für die Phase eine Darstellung in Grad.

Mit Hilfe der beiden obigen Testfunktionen soll nun der diskrete Frequenzgang

$$\hat{G}(\omega_l) = \frac{Y_n(\omega_l)}{U_n(\omega_l)} = \frac{\text{DFT}((y_k))}{\text{DFT}((u_k))} \quad \text{mit} \quad \omega_l = \frac{2\pi l}{NT_a}, \quad l = 1, 2, \dots, \frac{N}{2} - 1 \quad (1.5)$$

in MATLAB geschätzt werden.

Aufgabe 1.3 (ETFE).

1. Simulieren Sie den Einmassenschwinger mit den von Ihnen definierten Chirp-

und PRBS-Signalen als Eingang. Wählen Sie dazu geeignete Werte für ω_s und ω_e , sowie eine Amplitude von $\hat{U} = 10 \text{ N}$ ($[\hat{U}] = \text{N}$), eine Abtastzeit $T_a = 25 \text{ ms}$, eine Überabtastung $P_p = 4$, eine Ordnung $O_p = 10$ und $N = 4096$ Abtastpunkte. Mit dem MATLAB/SIMULINK-Modell `Einmassenschwinger.slx` steht Ihnen dazu ein Simulationsmodell des Einmassenschwingers zur Verfügung.

2. Ermitteln Sie für die Messung ohne Messrauschen eine Schätzung des Frequenzgangs mit Hilfe der ETFE. Vergleichen Sie die Ergebnisse der ETFE mit Chirp-Signal und der ETFE mit PRBS-Signal mit dem Bodediagramm der zeitdiskreten Strecke aus Aufgabe 1.1.
3. Wiederholen Sie die obigen Schätzungen für die Messungen mit Messrauschen.

Hinweis:

- Die Simulationsparameter werden mit Hilfe der MATLAB-struct `parSim` vorgegeben. Definieren Sie für das Eingangssignal den Zeitvektor `parSim.t_in`, den Vektor der Eingangswerte `parSim.u_in` und wählen Sie geeignete Anfangswerte für $s_{m,0}$ und $v_{m,0}$, die Sie dem Parameter `parSim.xx0` zuweisen. Achten Sie darauf, dass hier jeweils Zeilenvektoren erwartet werden. Setzen Sie zusätzlich den Parameter `parSim.Methode` gleich `null` und definieren Sie die Abtastzeit `parSim.T_a`.
- Sie können die Simulation des MATLAB/SIMULINK-Modells direkt aus dem workspace oder aus einer m-Datei mit Hilfe des Befehls `sim('Einmassenschwinger',simtime)`, wobei `simtime` für die notwendige Simulationszeit steht, starten.
- Die Simulationsergebnisse stehen Ihnen nach erfolgter Simulation im workspace in den MATLAB-structs mit den Namen `sim_kein_rauschen` und `sim_mit_rauschen` zur Verfügung, wobei die Simulation einmal ohne und einmal mit simulierten Messrauschen durchgeführt wurde.
- Verzichten Sie während der gesamten Übung auf die Benutzung des MATLAB-Befehls `etfe`.

1.2 Parametrische Identifikation

Die Least-Squares Identifikation soll in dieser Übung anhand zweier möglicher Formulierungen des Schätzproblems diskutiert werden. In einem ersten Schritt wird die Schätzung der Parameter des Einmassenschwingers mit Hilfe einer zeitdiskreten Darstellung des Systems erarbeitet. In einem zweiten Schritt wird eine Schätzung der Parameter auf Basis der zeitkontinuierlichen Darstellung ermittelt.

1.2.1 Parameterschätzung aus zeitdiskreter Darstellung

Die erste Möglichkeit zur Schätzung der Parameter besteht in der Diskretisierung der kontinuierlichen Übertragungsfunktion $G(s)$ und der anschließenden Least-Squares Schätzung der Parameter der diskreten Übertragungsfunktion $G(z)$. Für eine diskrete Übertragungsfunktion $G(z)$ der Form

$$G(z) = \frac{B(z)}{A(z)} = z^{-d} \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}, \quad m \leq n, d = n - m \quad (1.6)$$

mit Zählergrad m und Nennergrad n berechnet sich der Wert der Ausgangsfolge (y_k) zum k -ten Abtastzeitpunkt zu

$$y_k = \underbrace{\begin{bmatrix} -y_{k-1} & -y_{k-2} & \dots & -y_{k-n} & u_{k-d} & u_{k-d-1} & \dots & u_{k-d-m} \end{bmatrix}}_{\mathbf{s}_k^T} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_0 \\ \vdots \\ b_m \end{bmatrix}}_{\mathbf{p}} \quad (1.7)$$

mit dem Datenvektor \mathbf{s}_k und dem Parametervektor \mathbf{p} . Diese Darstellung stellt die Basis für die Anwendung des Least-Squares Verfahrens dar, da nunmehr der Parametervektor \mathbf{p} so geschätzt wird, dass der verallgemeinerte Gleichungsfehler

$$e_k = y_k - \hat{y}_k = y_k - \mathbf{s}_k^T \hat{\mathbf{p}} \quad (1.8)$$

mit dem Schätzwert $\hat{y}_k = \mathbf{s}_k^T \hat{\mathbf{p}}$ nach Gleichung (1.7) und dem geschätzten Parametervektor $\hat{\mathbf{p}}$ minimiert wird. Für $j = 0, \dots, N$ Messungen erhält man

$$\underbrace{\begin{bmatrix} e_0 \\ \vdots \\ e_N \end{bmatrix}}_{\mathbf{e}_N} = \underbrace{\begin{bmatrix} y_0 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{y}_N} - \underbrace{\begin{bmatrix} \mathbf{s}_0^T \\ \vdots \\ \mathbf{s}_N^T \end{bmatrix}}_{\mathbf{S}_N} \hat{\mathbf{p}}_N, \quad (1.9)$$

wobei der Index N andeutet, dass $N + 1$ Messungen zur Schätzung von \mathbf{p} herangezogen werden. Löst man die Minimierungsaufgabe

$$\min_{\mathbf{p}} \mathbf{e}_N^T \mathbf{e}_N, \quad (1.10)$$

so folgt die Lösung des Schätzproblems im Sinne der kleinsten Fehlerquadrate zu

$$\hat{\mathbf{p}}_N = \left(\mathbf{S}_N^T \mathbf{S}_N \right)^{-1} \mathbf{S}_N^T \mathbf{y}_N. \quad (1.11)$$

Die exakte Diskretisierung der Übertragungsfunktion $G(s)$ führt in der Regel auf nichtlineare Zusammenhänge zwischen den Parametern des kontinuierlichen und des

diskreten Modells, siehe Aufgabe 1.1. Um die physikalischen Parameter des Systems (m , c , d) zu erhalten, muss daher eine nichtlineare Umrechnung zwischen den Ersatzparametern der diskreten Übertragungsfunktion und den realen Parametern durchgeführt werden. Um diese Problematik zu umgehen, wird im Folgenden zusätzlich eine Approximation der Differentiation durch eine numerische Differentiation mittels der Untersumme (Euler–Vorwärts–Verfahren) und der Trapezregel untersucht.

Ausgehend von der kontinuierlichen Übertragungsfunktion $G(s)$ können die zugehörigen Approximationen $G^E(z)$ mittels Untersumme bzw. $G^T(z)$ mittels Trapezregel ermittelt werden, indem man die Laplace-Variable s in $G(s)$ für das Euler–Verfahren durch

$$s \approx \frac{z-1}{T_a} \quad (1.12)$$

ersetzt und für die Trapezregel

$$s \approx \frac{2}{T_a} \frac{z-1}{z+1} \quad (1.13)$$

verwendet.

Aufgabe 1.4 (Numerische Diskretisierung). Diskretisieren Sie die Übertragungsfunktion $G(s)$ des Einmassenschwingers mit Hilfe der Näherungsformeln (1.12) und (1.13) und vergleichen Sie das Ergebnis mit der exakten Diskretisierung anhand der Pole und der Nullstellen sowie der Sprungantworten der Übertragungsfunktionen. Untersuchen Sie hier insbesondere den Einfluss der Abtastzeit T_a im Bereich $T_a = 1$ ms bis $T_a = 50$ ms.

Die obigen Analysen werden zeigen, dass die Diskretisierung mittels Euler–Verfahren nicht sinnvoll ist. Aus diesem Grund wird in den folgenden Aufgabenstellungen nur mehr die Diskretisierung mit Hilfe der Trapezregel nach (1.13) verwendet. Bei dieser Diskretisierung entstehen zusätzliche Nullstellen in der diskreten Übertragungsfunktion $G^T(z)$. Man erkennt, dass die Koeffizienten b_0, b_1, b_2 des Zählerpolynoms von $G^T(z)$ in der Form

$$2b_0 = b_1 = 2b_2 \quad (1.14)$$

linear abhängig sind und damit die Lösung der Identifikationsaufgabe erschwert wird. Führt man einen neuen virtuellen Eingang \bar{u}_k in der Form

$$\bar{u}_k = u_k + 2u_{k-1} + u_{k-2} \quad (1.15)$$

für die Identifikation ein, so kann die Schätzaufgabe auf die Ermittlung von drei Ersatzparametern reduziert werden.

Aufgabe 1.5 (Formulierung der diskreten Identifikationsaufgabe).

1. Führen Sie den Eingang \bar{u}_k nach (1.15) ein und überführen Sie die diskrete Übertragungsfunktion $G^T(z)$ in eine neue Form $\bar{G}^T(z)$ mit dem neuen Eingang \bar{u} und dem Ausgang y .
2. Berechnen Sie die zu $\bar{G}^T(z)$ zugehörige Differenzengleichung und geben Sie diese in Vektorschreibweise $y_k = \mathbf{s}_k^T \mathbf{p}$ mit dem Datenvektor \mathbf{s}_k und dem Parametervektor \mathbf{p} an.

3. Drücken Sie die physikalischen Parameter des Einmassenschwingers als Funktion der Parameter der diskreten Übertragungsfunktion $\tilde{G}^T(z)$ aus.

Aufgabe 1.6 (Diskrete Parameteridentifikation).

1. Generieren Sie durch Simulation mit Hilfe eines PRBS-Signals eine Messung \mathbf{y}, \mathbf{u} für eine Abtastzeit $T_a = 25$ ms und eine Messdauer von $T_{sim} = 100$ s. Verwenden Sie dazu das MATLAB/SIMULINK-Modell **Einmassenschwinger.slx** und ein PRBS-Signal mit $\hat{U} = 10$ N ($[\hat{U}] = \text{N}$), $O_p = 10$ sowie $P_p = 4$.
2. Führen Sie die Least-Squares Schätzung für die Parameter des diskreten Modells des Einmassenschwingers aus Aufgabe 1.4 und 1.5 durch. Verwenden Sie zum Lösen des Schätzproblems den MATLAB-Backslashoperator.
3. Variieren Sie die Überabtastung P_p und die Messdauer und vergleichen Sie Ihre geschätzten Parameter.
4. Wiederholen Sie Ihre Identifikation mit den verrauschten Signalen und vergleichen Sie die Ergebnisse mit den Ergebnissen der ungestörten Signale. Untersuchen Sie insbesondere den Einfluss der Messdauer T_{sim} .
5. Überlegen Sie sich, warum im Fall von verrauschten Signalen keine korrekte Schätzung der Parameter möglich ist.

Hinweis:

- Definieren Sie für die Simulation den Zeitvektor `parSim.t_in`, den Eingangsvektor `parSim.u_in`, den Vektor der Anfangswerte `parSim.xx0` sowie die Abtastzeit `parSim.T_a` und setzen Sie den Parameter `parSim.Methode` gleich null.
- Der Eingang ist nach der Simulation im MATLAB-struct mit dem Namen `sim_eingang`, der unverrauschte Ausgang in `sim_kein_rauschen` und der verrauschte Ausgang in `sim_mit_rauschen` abgespeichert.
- Beachten Sie, dass Sie einige Abtastpunkte (ca. 10) am Anfang der Messung herausnehmen sollten. Es ist mindestens so lange zu warten, bis ein sinnvoller und vollständiger Vektor \mathbf{s}_k gebildet werden kann. Anderenfalls entstehen ungültige Einträge im Datenvektor, die große Fehler im Gütekriterium der Least-Squares Schätzung ergeben und zu unbrauchbaren Schätzergebnissen führen.

1.2.2 Parameterschätzung aus der Differentialgleichung

Jedes lineare zeitinvariante SISO-System n -ter Ordnung kann in der allgemeinen Form

$$y^{(n)} = -a_0 y - a_1 \dot{y} - \dots - a_{n-1} y^{(n-1)} + b_0 u + b_1 \dot{u} + \dots + b_m u^{(m)}, \quad (1.16a)$$

bzw. im Laplacebereich als

$$s^n \hat{y} = -a_0 \hat{y} - a_1 s \hat{y} \dots - a_{n-1} s^{n-1} \hat{y} + b_0 \hat{u} + b_1 s \hat{u} + \dots + b_m s^m \hat{u}, \quad (1.16b)$$

mit $n \geq m$, dem Eingang u , dem Ausgang y sowie den konstanten Koeffizienten $a_i, i = 0, \dots, n-1$ und $b_i, i = 0, \dots, m$, dargestellt werden.

Hat man nun neben der Messung von y und u auch die zeitlichen Ableitungen $y^{(k)}, u^{(k)}$, bzw. $s^k \hat{y}, s^k \hat{u}$ zur Verfügung, kann man die Parameter des Systems direkt mit Hilfe eines Least-Squares Schätzproblems nach (1.7) ermitteln. Das Problem hierbei ist, dass man einen idealen Differenzierer benötigt, um die Ableitungen zu erzeugen. Dieser ist jedoch bekanntlich nicht realisierbar.

Sei nun

$$F(s) = \frac{f_0}{s^{n+1} + f_n s^n + \dots + f_0} \quad (1.17)$$

ein Realisierungsterm, mit dem die Gleichung (1.16b) multipliziert wird ergibt sich

$$F(s) s^n \hat{y} = -a_0 F(s) \hat{y} - a_1 F(s) s \hat{y} \dots - a_{n-1} F(s) s^{n-1} \hat{y} + b_0 F(s) \hat{u} + b_1 F(s) s \hat{u} + \dots + b_m F(s) s^m \hat{u}. \quad (1.18)$$

Man erkennt nun, dass mit den gefilterten Signalen $s^k F(s) \hat{y}$ und $s^k F(s) \hat{u}$ wieder dasselbe Schätzproblem für die Parameter a_k, b_k erzeugt werden kann. Somit können auch kausale Filter $s^k F(s)$ (für $k \leq n$) benutzt werden.

Es gilt also, einfach die Signale

$$y_f^{(k)} = \mathcal{L}^{-1}[s^k F(s) \hat{y}] \quad \text{und} \quad u_f^{(k)} = \mathcal{L}^{-1}[s^k F(s) \hat{u}] \quad (1.19)$$

zu berechnen und wie in (1.7) weiter zu verarbeiten.

Durch Lösen des Realisierungsproblems (vgl. [2]) ergibt sich dann ein sogenanntes Zustandsvariablenfilter (ZVF) in der Form

$$\frac{d}{dt} \mathbf{x}_f = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -f_0 & -f_1 & -f_2 & \dots & -f_n \end{bmatrix} \mathbf{x}_f + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} y \quad (1.20a)$$

$$\begin{bmatrix} y_f \\ \vdots \\ y_f^{(n)} \end{bmatrix} = f_0 \mathbf{x}_f \quad (1.20b)$$

für den Ausgang y . Da für das Eingangsfilter und das Ausgangsfilter der selbe Realisierungsterm notwendig ist wird das Filter nach (1.20) auch für den Eingang u verwendet.

Aufgabe 1.7 (Zustandsvariablenfilter).

1. Stellen Sie das Modell des Einmassenschwingers in der Form (1.16a) dar.
2. Berechnen Sie das Zustandsvariablenfilter für den Eingang u und den Ausgang y . Beachten Sie, dass hierzu (mindestens) eine Filterordnung von $n + 1 = 3$ notwendig ist. Wählen Sie die Filterkoeffizienten f_j so, dass die Pole der Übertragungsfunktion $F_k(s)$ bei $s_j = -\frac{1}{T_f}$ liegen. Die Filterzeitkonstante T_f sei dabei noch unbestimmt.
3. Überlegen Sie sich einen sinnvollen Wertebereich für die Filterzeitkonstante T_f und wählen Sie T_f entsprechend. Implementieren Sie anschließend das kontinuierliche Zustandsvariablenfilter für u und y in Form eines state-space Objektes (MATLAB-Befehl `ss`).
4. Diskretisieren Sie das Filter mit dem MATLAB-Befehl `c2d()` und wählen Sie eine passende Abtastzeit $T_{a,ZVF}$ für das Filter. Die Abtastzeit $T_{a,ZVF}$ am Zustandsvariablenfilter muss dabei nicht zwingend gleich der Abtastzeit T_a an der Strecke sein. Denken Sie daran, dass die durch T_f festgelegte Filterdynamik mit der Abtastzeit $T_{a,ZVF}$ des diskretisierten Filters auch dargestellt werden können muss. Fügen Sie schließlich das Filter in das Subsystem ZVF ein, was Sie im SIMULINK-Modell `Einmassenschwinger.slx` im varianten Subsystem `Zustandsvariablenfilter` finden und setzen Sie den Parameter `parSim.Methode` auf `eins`, um das Subsystem ZVF zu aktivieren.

Aufgabe 1.8 (Kontinuierliche Parameteridentifikation).

1. Generieren Sie durch Simulation eine Messung \mathbf{y}_f , $\dot{\mathbf{y}}_f$, $\ddot{\mathbf{y}}_f$ und \mathbf{u}_f für eine Abtastzeit $T_a = 25$ ms und einer Messdauer von $T_{Sim} = 20$ s unter Zuhilfenahme des Zustandsvariablenfilters und schätzen Sie die Parameter der Differentialgleichung des Einmassenschwingers. Verwenden Sie dazu wiederum ein PRBS-Signal mit $\hat{U} = 10$ N ($[\hat{U}] = \text{N}$), $O_p = 10$ sowie $P_p = 4$. Berechnen Sie anschließend die physikalischen Parameter des Einmassenschwingers.
2. Vergleichen Sie die Ergebnisse der Parameterschätzung der verrauschten Signale mit den Ergebnissen der ungestörten Signale und untersuchen Sie den Einfluss der Messdauer T_{sim} .

Hinweis:

- Definieren Sie für die Simulation den Zeitvektor `parSim.t_in`, den Eingangsvektor `parSim.u_in`, den Vektor der Anfangswerte `parSim.xx0` sowie die Abtastzeit `parSim.T_a` und setzen Sie den Parameter `parSim.Methode` gleich `eins`.
- Im Simulationsmodell `Einmassenschwinger.slx` ist ein variantes Subsystem mit dem Namen `Methoden` vorhanden. In diesem finden Sie das Subsystem ZVF, indem Ihr Zustandsvariablenfilter implementiert werden soll. Über den

Parameter `parSim.Methode` gleich eins wird dieses aktiviert.

- Der gefilterte Eingang ist nach der Simulation im MATLAB-struct mit dem Namen `sim_ZVF_u`, der unverrauschte gefilterte Ausgang in `sim_ZVF_y` und der verrauschte Ausgang in `sim_ZVF_y_rausch` abgespeichert.

1.2.3 Rekursive Parameterschätzung

In der SIMULINK-Datei `Einmassenschwinger.slx` unter `Methoden/RLS` ist eine Embedded MATLAB Function mit dem Namen `RLS` implementiert, die die Struktur zur Programmierung des rekursiven Least-Squares Verfahrens zeigt.

Aufgabe 1.9 (Rekursive Least-Squares Identifikation).

1. Programmieren Sie das rekursive Least-Squares Verfahren mit exponentiell abklingendem Gedächtnis auf Basis des Näherungsmodells (Trapezregel) aus Aufgabe 1.4–1.6. Beachten Sie hierbei, dass die Schätzfehlerkovarianzmatrix \mathbf{P} symmetrisch ist und daher nur $(n^2 + n)/2$ Einträge der Matrix berechnet werden müssen, wobei n die Anzahl der zu schätzenden Parameter ist.
2. Wählen Sie den Nullvektor als Anfangsbedingung für die Parameter und den Gedächtnisfaktor q zu eins. Führen Sie eine Schätzung durch. Verwenden Sie dazu eine Abtastzeit von $T_a = 25$ ms, eine Messzeit von $T_{sim} = 100$ s und ein PRBS-Signal mit $\hat{U} = 10$ N ($[\hat{U}] = \text{N}$), $O_p = 10$ und $P_p = 40$. Variieren Sie die Anfangsbedingungen der Schätzfehlerkovarianzmatrix \mathbf{P} und vergleichen Sie die Konvergenzgeschwindigkeit des Verfahrens.
3. Ändern Sie die Steifigkeit c der Feder rampenförmig mit der bereitgestellten Funktion. Passen Sie den Gedächtnisfaktor q so an, dass die Schätzung dem Parametersollverlauf folgen kann.
4. Beaufschlagen Sie ihre Messung mit dem vorgegebenen Rauschprozess und überprüfen Sie den Einfluss des Gedächtnisfaktors auf die Rauschunterdrückung und das Parameterfolgeverhalten.

Hinweis:

- Definieren Sie für die Simulation den Zeitvektor `parSim.t_in`, den Eingangsvektor `parSim.u_in`, den Vektor der Anfangswerte `parSim.xx0` sowie die Abtastzeit `parSim.T_a` und setzen Sie den Parameter `parSim.Methode` gleich zwei.
- Für die Simulation wird zusätzlich die Parameterstruktur `parRLS` mit den Elementen für die Abtastzeit `T_a`, dem Gedächtnisfaktor `q` und die Initialisierung der Kovarianzmatrix `alpha` benötigt, die Sie durch eine geeignete Wahl initialisieren müssen.
- Das rekursive Least-Squares Verfahren kann sehr rechengünstig implementiert

werden, indem der Term $\mathbf{s}^T \mathbf{P}$ wiederverwendet wird.

- Im Vergleich zu den vorigen Aufgaben wird in dieser Aufgabe das PRBS-Signal noch mit einem Filter 1. Ordnung gefiltert. Dies führt zu einem besseren Verhalten des RLS-Schätzers. Vergleichen Sie dazu die Ergebnisse ohne Filter mit jenen mit Filter, indem Sie den **Manual Switch** umschalten.

1.3 Verfahren nach Clary

Im Folgenden soll die Übertragungsfunktion des Einmassenschwingers aus Abschnitt 1.1 um die bekannte Dynamik des Aktors G_a erweitert werden. Es wird angenommen, dass die Dynamik des Aktors näherungsweise in der Form

$$G_a(s) = \frac{\hat{F}(s)}{\hat{F}_i(s)} = \frac{10}{s^2} \quad (1.21)$$

mit der Kraft F und dem Eingang F_i , gegeben ist. Die gesamte Übertragungsfunktion vom neuen Eingang F_i zum Ausgang s_m kann damit in der Form

$$G_g(s) = G_a(s)G(s) = \frac{10}{s^2} \frac{1}{ms^2 + ds + c} \quad (1.22)$$

dargestellt werden. Mit dem Verfahren nach Clary soll eine ETFE der Strecke $G(s)$ vorgenommen werden.

Erarbeiten Sie die folgenden Aufgaben in einem m-File mit MATLAB.

Aufgabe 1.10 (Clary).

1. Berechnen Sie mit der Abtastzeit $T_a = 25\text{ms}$ die z -Übertragungsfunktion $G_g(z)$ zur neuen Strecke $G_g(s)$. (Diese dient als Referenz für die Ergebnisse der ETFE und steht bei realen Problemstellungen natürlich nicht zur Verfügung!)
2. Ermitteln Sie geeignete Vorfilter zur Berücksichtigung des bekannten Teils der Strecke $G_a(s)$ mit Hilfe des Verfahrens nach Clary. Bedenken Sie dabei, dass für die ETFE keine Annahmen über die Strecke getroffen werden, d.h. dass nur $G_a(s)$ bekannt ist. Da jedoch die Pole einer Übertragungsfunktion eindeutig vom s -Bereich in den z -Bereich transformiert werden, können trotzdem sinnvolle Vorfilter für $G_a(s)$ erstellt werden.
3. Führen Sie eine Simulation mit dem MATLAB/SIMULINK-Modell **Einmassenschwinger.slx** durch und setzen Sie dafür den Parameter **parSim.Methode** auf drei. Verwenden Sie weiters das Chirp-Signal aus Aufgabe 1.3 mit $\omega_s = 2\pi 0.3 \text{ rad/s}$, $\omega_e = 2\pi 10 \text{ rad/s}$, $\hat{U} = 10 \text{ N}$ ($[\hat{U}] = \text{N}$) und $N = 4096$.
4. Führen Sie die ETFE einmal mit und einmal ohne die Vorfilter des Clary-Verfahrens durch und vergleichen Sie beide Ergebnisse mit den exakten Ergebnissen der Strecke $G_g(z)$ aus Punkt 1.

Hinweis:

Zur Berechnung der über die Vorfilter gefilterten Ein- und Ausgangssignale steht Ihnen der MATLAB-Befehl *lsim* zur Verfügung.

5. Wiederholen Sie Ihre Identifikation mit verrauschten Messsignalen.

Hinweis:

Definieren Sie für die Simulation den Zeitvektor `parSim.t_in`, den Eingangsvektor `parSim.u_in`, den Vektor den Anfangswerte `parSim.xx0` sowie die Abtastzeit `parSim.T_a` und setzen Sie den Parameter `parSim.Methode` gleich drei.

1.4 Adaptive Signalrekonstruktion

Im Folgenden soll der Least-Mean Squares (LMS) Algorithmus zur Rekonstruktion eines periodischen Nutzsignals bzw. eines Audiosignals verwendet werden. Man betrachte dazu die in Abbildung 1.2 dargestellte Übertragungsstrecke mit den unbekannten Übertragungsfunktionen $G_1(\delta)$ und $G_2(\delta)$. An dieser werden die beiden Signale (d_k) und (n_k) gemessen. Die gemessene Signalfolge (d_k) setzt sich aus dem zu identifizierenden Nutzsignal (r_k) und dem Störanteil $G_1(\delta)m_k$ zufolge des Störsignals (m_k) zusammen. Die Signalfolge (n_k) hingegen stellt eine indirekte Messung der Störung (m_k) dar, gemäß $n_k = G_2(\delta)m_k$.

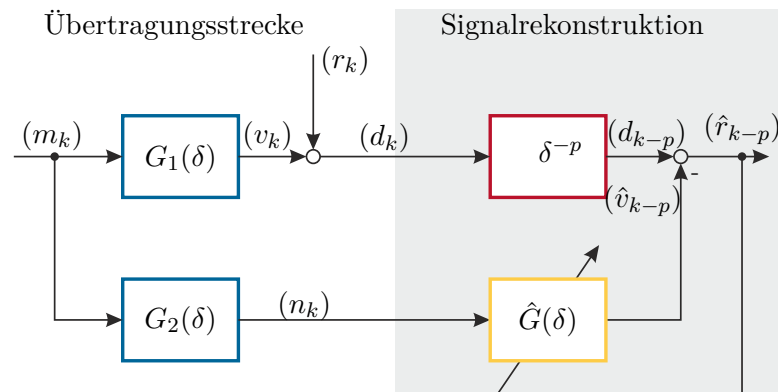


Abbildung 1.2: Zur adaptiven Signalfilterung mit dem LMS-Algorithmus.

Die Rekonstruktion des Nutzsignals (r_k) erfolgt nun nach Abbildung 1.2 mit Hilfe eines MA-Modell (vgl. [1]) der Form

$$\hat{G}(\delta) = \hat{C}(\delta) = \hat{c}_0 + \hat{c}_1\delta^{-1} + \dots + \hat{c}_q\delta^{-q}. \quad (1.23)$$

Der Schätzwert des Signals \hat{r}_{k-p} errechnet sich damit zu

$$\hat{r}_{k-p} = d_{k-p} - \underbrace{\begin{bmatrix} n_k & n_{k-1} & n_{k-2} & \dots & n_{k-q} \end{bmatrix}}_{\mathbf{s}_k^T} \underbrace{\begin{bmatrix} \hat{c}_0 \\ \hat{c}_1 \\ \vdots \\ \hat{c}_q \end{bmatrix}}_{\hat{\mathbf{p}}} \quad (1.24)$$

mit der Signalverzögerung p , dem Schätzwert $\hat{\mathbf{p}}$ des Parametervektors und dem Datenvektor \mathbf{s}_k^T . Durch Anwendung des LMS-Algorithmus folgt der Parameterschätzer zu

$$\hat{\mathbf{p}}_j = \hat{\mathbf{p}}_{j-1} + \mu_j \mathbf{s}_j \underbrace{\left(d_{j-p} - \mathbf{s}_j^T \hat{\mathbf{p}}_{j-1} \right)}_{\hat{r}_{j-p}}. \quad (1.25)$$

Der zeitvariante Parameter μ_j wird hierbei in der Form

$$\mu_j = \frac{\bar{\mu}}{l_j} \quad \text{mit} \quad l_{j+1} = l_j + \bar{\mu} (\mathbf{s}_j^T \mathbf{s}_j - l_j) \quad (1.26)$$

mit dem Anfangswert $l_{-1} = \mathbf{s}_{-1}^T \mathbf{s}_{-1} > 0$ gewählt.

Erarbeiten Sie die nachfolgenden Aufgaben in einem m-File mit MATLAB.

Aufgabe 1.11 (Adaptive Signalrekonstruktion).

1. Entwerfen Sie einen on-line Algorithmus nach der LMS-Methode und extrahieren Sie aus den in der Datei `data_sinus.mat` bereit gestellten Signalverläufen (d_k) und (n_k) das periodische Nutzsignal (r_k) = ($\sin(8 \times 10^3 \pi t)$). Wählen Sie hierfür eine Signalverzögerung $p = 1$, für das MA-Modell eine Modellordnung $q = 40$ und für den Parameter $\bar{\mu} = 0.02$. Zur Kontrolle stellen Sie das rekonstruierte Signal (\hat{r}_k) dem Nutzsignal (r_k) gegenüber.
2. Ändern Sie die Ordnung q des MA-Modells bzw. die Signalverzögerung p und untersuchen Sie die Auswirkungen auf das Ergebnis.
3. Verwenden Sie in weiterer Folge den entwickelten Algorithmus und rekonstruieren Sie aus den im Ordner `Audio` zur Verfügung gestellten Signalverläufen (d_k) und (n_k) das gesuchte Audiosignal (r_k). Wählen Sie hierfür wiederum eine Signalverzögerung $p = 1$, für das MA-Modell eine Modellordnung $q = 40$ und für den Parameter $\bar{\mu} = 0.02$.

Hinweis: Das gemessene Signal (d_k) bzw. das rekonstruierte Audiosignal (r_k) kann mit dem MATLAB-Befehl `sound` über die Soundkarte an einem Lautsprecher abgespielt werden. Alternativ kann der Befehl `audiowrite` verwendet werden. Dieser speichert die Daten in einer `.wav`-Datei.

1.5 Literatur

- [1] W. Kemmetmüller und A. Kugi, *Skriptum zur VU Regelungssysteme 1 (WS 2018/2019)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2018. Adresse: <https://www.acin.tuwien.ac.at/master/regelungssysteme-1/>.
- [2] A. Kugi, *Skriptum zur VU Automatisierung (WS 2018/2019)*, Institut für Automatisierungs- und Regelungstechnik, TU Wien, 2018. Adresse: <http://www.acin.tuwien.ac.at/?id=42>.