Übung 11
Simulation

Benötigt: -
Befehle: ode23, fxdot, @(...)

Aufgabe 1: Simulation mit Differenzenapproximation

Gegeben ist die Differentialgleichung eines dynamischen Systems mit dem Eingangssignal u und dem Ausgangssignal y zu:

$$4\ddot{y} + 2\dot{y} + y = 3u$$

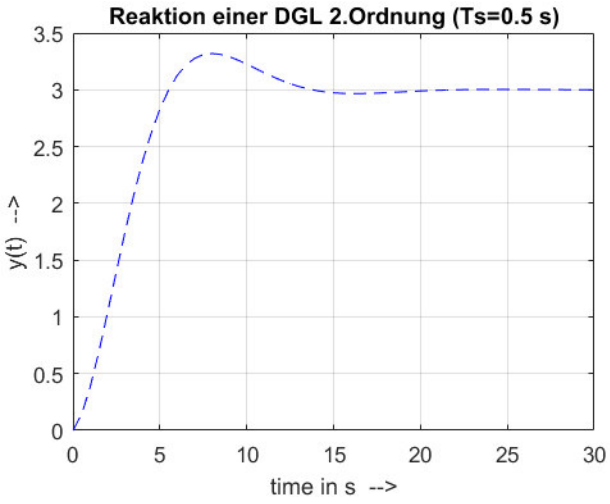
Mit der Anfangsbedingung $y(0) = \dot{y}(0) = 0$ simuliere man mit Ts = 0.5 s über 30 s die Reaktion des Systems, wenn zum Zeitpunkt t=0 der Eingang einen Einheitssprung von 0 auf 1 macht und stelle die Reaktion des Ausgangs grafisch dar.

- öffne neues Skript „Simulation.m“
- wandle die Differentialgleichung in eine Differenzengleichung um, indem man die Ableitungen durch die Differenzenquotienten

$$\dot{x}(k) \approx \frac{x(k) - x(k-1)}{T_s}$$

$$\ddot{x}(k) \approx \frac{x(k) - 2x(k-1) + x(k-2)}{T_s^2}$$

approximiert und nach dem neuesten Ausgangssignal y(k) auflöst. Dabei belasse man die Abtastzeit Ts allgemein, da diese später variiert werden wird

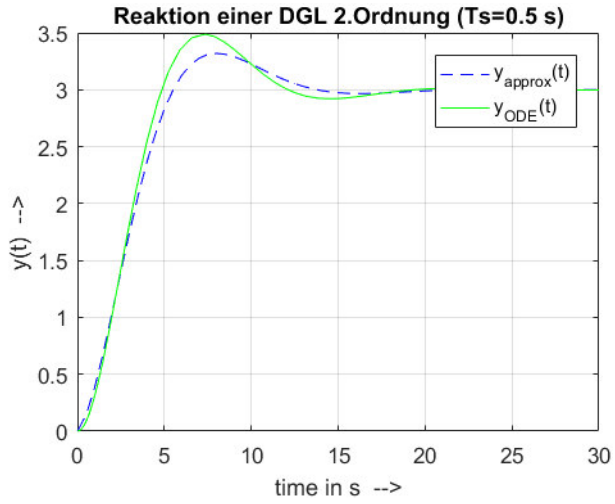


- man realisiere mit der gefundenen Differenzengleichung 2. Ordnung die Simulation der Sprungantwort und stelle das Ausgangsverhalten in einem Diagramm über der Zeit dar (gestrichelt, blau).
- sehe vor, dass noch weitere Kurvenverläufe in das Diagramm gezeichnet werden sollen

Aufgabe 2: Simulation mit ODE-Solver

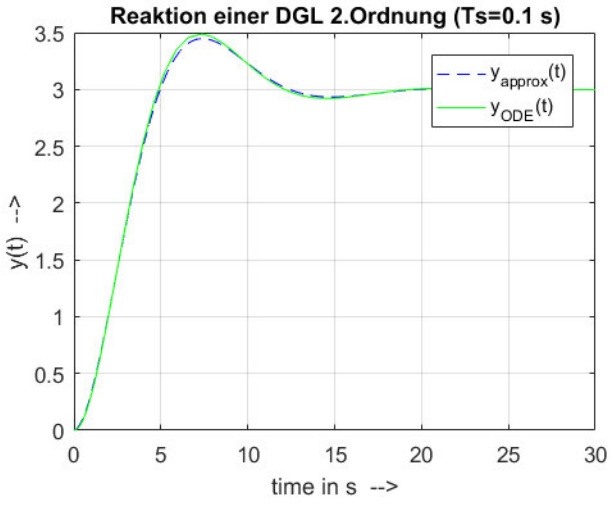
- man wandle das Skript „Simulation.m“ in eine Funktion um, indem man in die erste Zeile „**function** Simulation“ eingibt, da wir sonst keine lokale Unterfunktion realisieren können
- wandle die DGL in die sog. ODE-Form um, d.h. zerlege sie in DGLs 1. Ordnung
- bestimme hieraus die lokale Unterfunktion „fxdot(t,x,u)“ und realisiere sie am Ende des M-Files
- simuliere für einen Zeitraum von 30 s und dem Eingangssignal u=1 das Verhalten mit dem ODE23-Solver

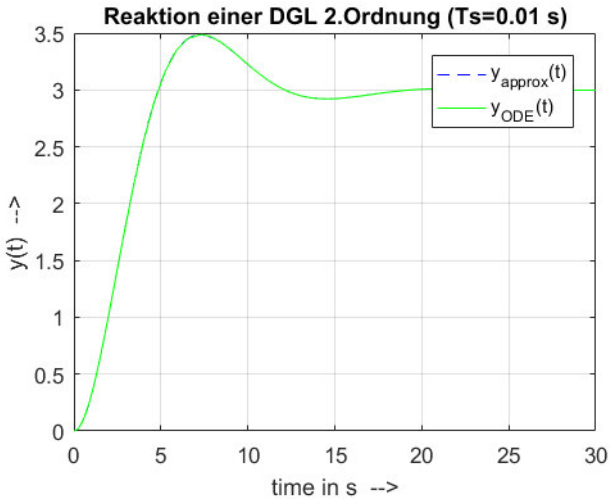
- stelle das Ausgangssignal über der Zeit dar (durchgezogen,grün) und füge eine passende Legende dazu



Aufgabe 3: Einfluss der Abtastzeit auf die Genauigkeit der Approximation

- führe die Funktion „Simulation.m“ nochmals mit einer Abtastzeit Ts=0.1 s und Ts=0.01 s aus und bewerte die Ergebnisse im Vergleich





Grundlagen - Differenzenapproximation

Auszug aus: Billmann, L.: Systemtheorie. 1st Edition ISBN 978-1-291-46832-8, Lulu Press Morrisville USA, 2013

0.0.1 Ansatz über Differenzenapproximation

Es ist klar, dass wir ein dynamisches System nicht nur mit einer Differentialgleichung beschreiben können, sondern auch mit einer sog. Differenzengleichung, sofern uns das Verhalten nur zu diskreten Zeitpunkten interessiert. So liegt die Frage nahe, wie man die passenden Koeffizienten der Differentialgleichung ermitteln kann.

Eine gangbare Lösung hierfür ist der Ansatz über die *Differenzenapproximation*, die die zeitlichen Ableitungen von Signalen durch Differenzen-Ansätze zu nähern versucht. Es muss daher klar sein, dass es sich hierbei nur um eine Näherung und keine exakte Lösung handeln kann, die aber in der Praxis sehr gern und häufig wegen ihrer Einfachheit verwendet wird.

Als Näherungsformel für die erste Ableitung eines Signals verwendet man

$$\boxed{\dot{x}(t) = \frac{d\,x(t)}{dt} \quad \rightarrow \quad \Delta x(k) = \frac{x(k) - x(k-1)}{T_0}} \quad , \tag{0.0.1.1}$$

was der Sekantensteigung als Näherung für die exakte Tangentensteigung entspricht.

Für die zweite Ableitung eines Signals lässt sich dann analog

$$\boxed{\begin{aligned} \ddot{x}(t) = \frac{d^2 x(t)}{dt^2} \quad \rightarrow \quad \Delta^2 x(k) &= \frac{\Delta x(k) - \Delta x(k-1)}{T_0} \\ &= \frac{x(k) - 2x(k-1) + x(k-2)}{T_0^2} \end{aligned}} \tag{0.0.1.2}$$

angeben, jedoch höhere Ordnungen verbieten sich durch die dann zu hohen Ungenauigkeiten.

Beispiel

Gegeben ist die Differentialgleichung eines dynamischen Systems

$$T \cdot \dot{y}(t) + y(t) = 2 \cdot u(t) \tag{0.0.1.3}$$

Die Aufgabe besteht nun darin, die zugehörige Differenzengleichung näherungsweise zu ermitteln. Hierzu nähern wir die erste Ableitung des Ausgangs durch Gl. (0.0.1.1) an und ersetzen die Signale selbst durch ihre diskreten Ausdrücke.

$$\begin{aligned} T \cdot \frac{y(k) - y(k-1)}{T_0} + y(k) &= 2\,u(k) \\ (1 + T/T_0)\,y(k) - T/T_0\,y(k-1) &= 2\,u(k) \\ y(k) &= \frac{1}{1 + T/T_0} \cdot \left[T/T_0\,y(k-1) + 2\,u(k) \right] \\ &= \frac{T/T_0}{1 + T/T_0} \cdot y(k-1) + \frac{2}{1 + T/T_0} \cdot u(k) \end{aligned}$$

Wir erhalten somit die Differenzengleichung in Berechnungsform zu

$$\boxed{\boldsymbol{y(k) = \frac{T}{T_0 + T} \cdot y(k-1) + \frac{2T_0}{T_0 + T} \cdot u(k)}} \quad . \tag{0.0.1.4}$$

Fazit

Mit dem beschriebenen Ansatz haben wir die Möglichkeit, eine Differentialgleichung durch eine Differenzengleichung anzunähern und zumindest für kleine Zeitschritte $\mathbf{T_0}$ ansprechend genaue Ergebnisse zu erzielen.

Ist das Vorgehen zunächst auf Ordnungen bis maximal 2 beschränkt, so lässt sich dies dadurch umgehen, dass man einer Differentialgleichung höherer Ordnung in mehrere Differentialgleichungen niedrigerer Ordnung (1. oder 2.) umwandelt und diese dann durch mehrere Differenzengleichungen annähert.

Grundlagen – Merkblatt ODE-Solver

Ansatz:
Die numerische Lösung gewöhnlicher Differentialgleichungen (**O**rdinary **D**ifferential **E**quations) kann man mit sog. ODE-Solver durchführen, wie sie etwa in MATLAB implementiert sind. Hierzu ist aber die Differentialgleichung meist höherer Ordnung in ein Gleichungssystem von DGLs 1.Ordnung umzuwandeln, damit etwa der ODE23-Solver damit arbeiten kann.

Die ODE-Form:
Haben wir ein dynamisches System etwa mit einem Eingang u(t) und einem Ausgang y(t), so werde dies durch eine gewöhnliche DGL n-ter Ordnung etwa beschrieben.
Sind nur zeitliche Ableitungen des Ausgangs y(t) enthalten und nur das Eingangssignal u(t) selbst, so spricht man von einem Verzugssystem, das sich recht einfach in die ODE-Form umstellen lässt.

- man Löse die DGL nach der höchsten Ableitung des Ausgangs auf

$$\frac{d^n y(t)}{dt^n} = f\left(y, \dot{y}, \ddot{y}, \ldots., u, t\right) \quad (1)$$

- man definiere einen Zustandsvektor X (Spaltenvektor) mit n Elementen, die sich aus dem Ausgangssignal und seinen Ableitungen zusammensetzt

$$X(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \\ \ddot{y}(t) \\ \vdots \\ \frac{d^n y(t)}{dt^n} \end{bmatrix} \quad (2)$$

- hiermit kann man das Verhalten des dynamischen Systems (die DGL) in folgender Form (die ODE-Form) beschreiben

$$\boxed{\dot{\boldsymbol{X}}(\boldsymbol{t}) = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{u}, \boldsymbol{t})} \quad (3)$$

- Umgesetzt wird dies in MATLAB etwa mit der Funktion „xdot = fxdot(t,x,u)“, die abhängig von den Zuständen X dem Eingangssignal(en) u und der Zeit t die zeitliche Ableitung aller Zustände berechnet

$$\dot{X}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} \dot{y}(t) \\ \ddot{y}(t) \\ \dddot{y}(t) \\ \vdots \\ \frac{d^n y(t)}{dt^n} \end{bmatrix} = \begin{bmatrix} x_2(t) \\ x_3(t) \\ x_4(t) \\ \vdots \\ \text{Gl.(1)} \end{bmatrix} \quad (4)$$

- Die Simulation selbst erfolgt dann durch den Aufruf des ODE-Solvers in der Form:

```
[t,x] = ode23(@(t,x) fxdot(t,x,u),[0 Tsim],[0 0]);
```

`@(t,x) fxdot(t,x,u)` mit einer sog. anonymen Funktion teilen wir dem ODE-Solver die Funktion mit, die unsere Dynamik beschreibt Gl.(4)
`[0 Tsim]` dies ist die Zeitspanne für die das System simuliert wird, d.h. hier etwa von 0 bis Tsim Sekunden. Welche Zeitschritte wirklich dazwischen genommen werden entscheidet der Solver selbst und gibt es in dem Vektor t zurück.
`[0 … 0]` dies sind die Startwerte für alle n Zustände, der sog. Anfangszustand.

- Als Ergebnis erhalten wir neben dem Zeitvektor t die Matrix x, die in jeder Zeile alle Zustandswerte zu diesem Zeitpunkt auflistet. Interessieren wir uns nur für das Ausgangssignal, welches ja der 1. Zustand ist, so kann man dies leicht mit y=x(:,1) extrahieren.