Modellierung und Simulation | **Übung 02**
WS 2016/17 | **Abgabefrist: Siehe elearning bzw. exchange**

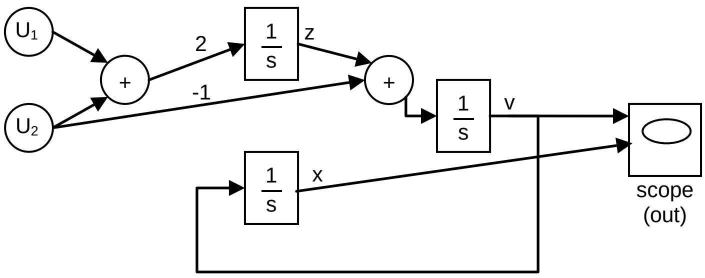
Name: ________________________ **Aufwand (h):** __________

Punkte: __________

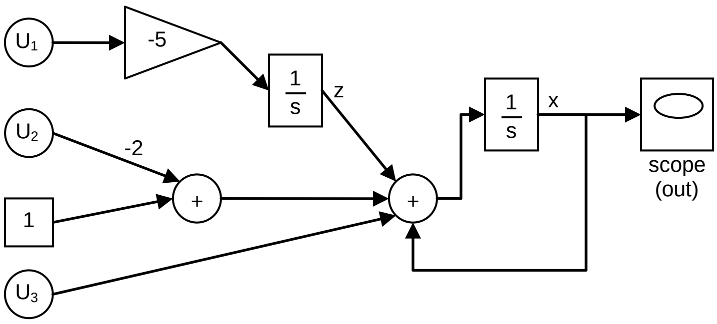
Aufgabe 1 (4 + 4 + 3 + 3 = 14 Pkt): Kontinuierliche Modellierung

(a) Beschreiben Sie folgende (als Blockschaltform gegebenen) Systeme in der (A,B,C)-Form:

(I)



(II)



(b) Beschreiben Sie die folgenden (in (A,B,C)-Form gegebenen) Systeme als Blockschaltbild:

(I) $A=\begin{bmatrix} 0 & -1 & 0 \\ 2 & -8 & 0 \\ -2 & 0 & 1 \end{bmatrix}, B=\begin{bmatrix} 1 & 2 \\ 3 & -2 \\ -1 & 1 \end{bmatrix}, C=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 1 \\ 0 & 3 & 0 \end{bmatrix}$

(II) $A=\begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}, B=\begin{bmatrix} 0 & 2 \\ 1 & -2 \end{bmatrix}, C=\begin{bmatrix} 0 & 1 \end{bmatrix}$

(c) Was ist an der folgenden Angabe eines in (A,B,C)-Form gegebenen Systems nicht korrekt?

$$A=\begin{bmatrix} 0 & -1 & 0 \\ 2 & -1 & 0 \\ -2 & 0 & 1 \end{bmatrix}, B=\begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix}, C=\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

(d) Beschreiben Sie folgende Systeme in (A,B,C)-Form und als Blockschaltbild:

$$
\begin{aligned}
a' &= 2a + 4b - 2c + 2i_1 & \qquad y_1 &= 3a \\
b' &= 4b - c + i_2 \qquad ; & y_2 &= -b \qquad (i : \text{Input},\ y : \text{Output}) \\
c' &= 3i_1 - c & y_3 &= a - c
\end{aligned}
$$

Aufgabe 2 (10 Pkt): Kontinuierliche Simulation

Beschreiben Sie in eigenen Worten, was man unter „Numerischer Integration“ im Zusammenhang mit Computer-Simulation versteht.
Gehen Sie dabei auch darauf ein, in welchem Kontext man sie benötigt, wozu man sie verwendet; Graphiken können dabei hilfreich sein.

Hinweise: Geben Sie Ihre Ausarbeitung gedruckt auf Papier ab.
Abgegebene Beispiele müssen in der Übungsstunde präsentiert werden können.

1 Kontinuierliche Modellierung

Dieser Abschnitt beschäftigt sich mit der Aufgabenstellung 1 der zweiten Übung.

1.1 Erstes Blockdiagramm für 1a 1

Dieser Abschnitt beschäftigt ich mit dem Aufstellen der Gleichungen in A, B, C Normalform, die aus den gegebenen Blockdiagramm aus Aufgabenstellung 1a abgeleitet wurden.

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, X = \begin{bmatrix} z \\ v \\ x \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$\begin{aligned}
z' &= 0*z + 0*v + 0*x + 2*u_1 + 2*u_2 &\equiv\;& 2*u_1 + 2*u_2 \\
v' &= 1*z + 0*v + 0*x + 0*u_1 - 1*u_2 &\equiv\;& z - u_2 \\
x' &= 0*z + 1*v + 0*x + 0*u_1 + 0*u_2 &\equiv\;& v
\end{aligned}$$

$$\begin{aligned}
y_1 &= 0*z + 0*v + 0*x &\equiv\;& 0 \\
y_2 &= 0*z + 1*v + 0*x &\equiv\;& v \\
y_3 &= 0*z + 0*v + 1*x &\equiv\;& x
\end{aligned}$$

$$X' = A*X + B*U$$

$$X' = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} z \\ v \\ x \end{bmatrix} + \begin{bmatrix} 2 & 2 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$Y = C*X$$

$$Y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} z \\ v \\ x \end{bmatrix}$$

1.2 Zweites Blockdiagramm für 1a 2

Dieser Abschnitt beschäftigt ich mit dem Aufstellen der Gleichungen in A, B, C Normalform, die aus den gegebenen Blockdiagramm aus Aufgabenstellung 1b abgeleitet wurden.

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, X = \begin{bmatrix} z \\ x \end{bmatrix}, Y = \begin{bmatrix} y1 \\ y2 \end{bmatrix}$$

$$\begin{aligned} z' &= 0*z + 0*x + 0*u_1 - 5*u_2 + 0*u_3 && \equiv -5*u_2 \\ x' &= 1*z + 1*x + 0*u_1 + (-2*u_2+1) + 1*u_3 && \equiv z + x + (-2*u_2+1) + u_3 \end{aligned}$$

$$\begin{aligned} y_1 &= 0*z + 0*x && \equiv 0 \\ y_2 &= 0*z + 1*x && \equiv x \end{aligned}$$

$$X' = A*X + B*U$$

$$X' = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} z \\ x \end{bmatrix} + \begin{bmatrix} 0 & -5 & 0 \\ 0 & (-2*u_2+1) & 1 \end{bmatrix} * \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$Y = C*X$$

$$Y = C = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} z \\ x \end{bmatrix}$$

1.3 Erstes System als Blockschaltbild für 1b 1

Dieser Abschnitt beschäftigt sich mit der Aufgabenstellung 1b 1. Die angeführten Vektoren wurden angenommen.

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, X = \begin{bmatrix} x \\ v \\ z \end{bmatrix}$$

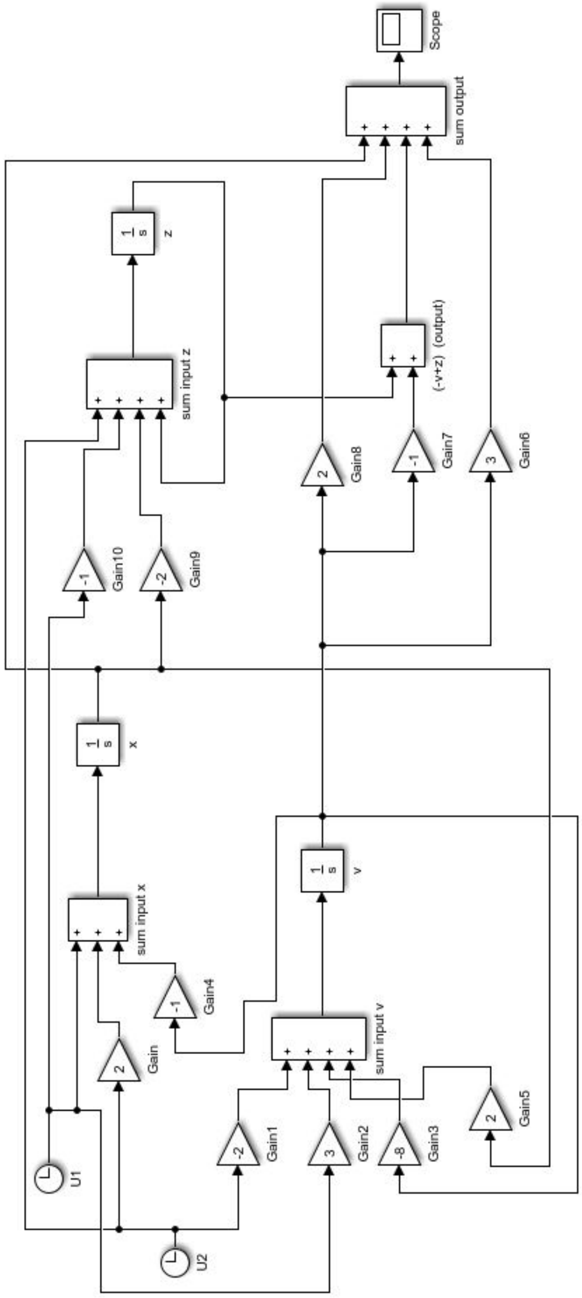


Abbildung 1: Blockschaltbild zum System aus Aufgabenstellung 1b 1

1.4 Zweites System als Blockschaltbild für 1b 2

Dieser Abschnitt beschäftigt sich mit der Aufgabenstellung 1b 2. Die angeführten Vektoren wurden angenommen.

$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, X = \begin{bmatrix} x \\ v \end{bmatrix}$

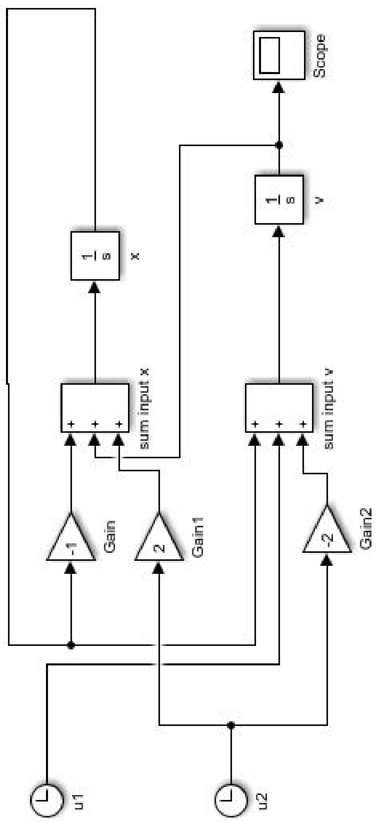


Abbildung 2: Blockschaltbild zum System aus Aufgabenstellung 1b 2

1.5 Fehlerhaftes System in A, B, C Normalform für 1c

Dieser Abschnitt beschäftigt sich mit der Aufgabenstellung 1c. Die für das System aufgestellte C-Matrix ist falsch, da dieses System drei Systemzustände besitzt und daher die C-Matrix drei Spalten haben muss, nähmlich so viele Spalten wie es Systemzustände gibt.

$$Y = C * X \quad wobei \quad X = \begin{bmatrix} x \\ v \\ z \end{bmatrix} \quad und \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 0 \end{bmatrix}$$

1.6 System als Blockschaltbild und in A, B, C Normalform für 1d

Dieser Abschnitt beschäftigt sich mit der Aufgabenstellung 1d.

$$X = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, U = \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}$$

$$X' = A * X + B * U$$

$$X' = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 4 & -1 \\ 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 3 & 0 \end{bmatrix} * \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}$$

$$Y = C * X$$

$$Y = \begin{bmatrix} 3 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} (3 * a) \\ (-b) \\ (a - c) \end{bmatrix}$$

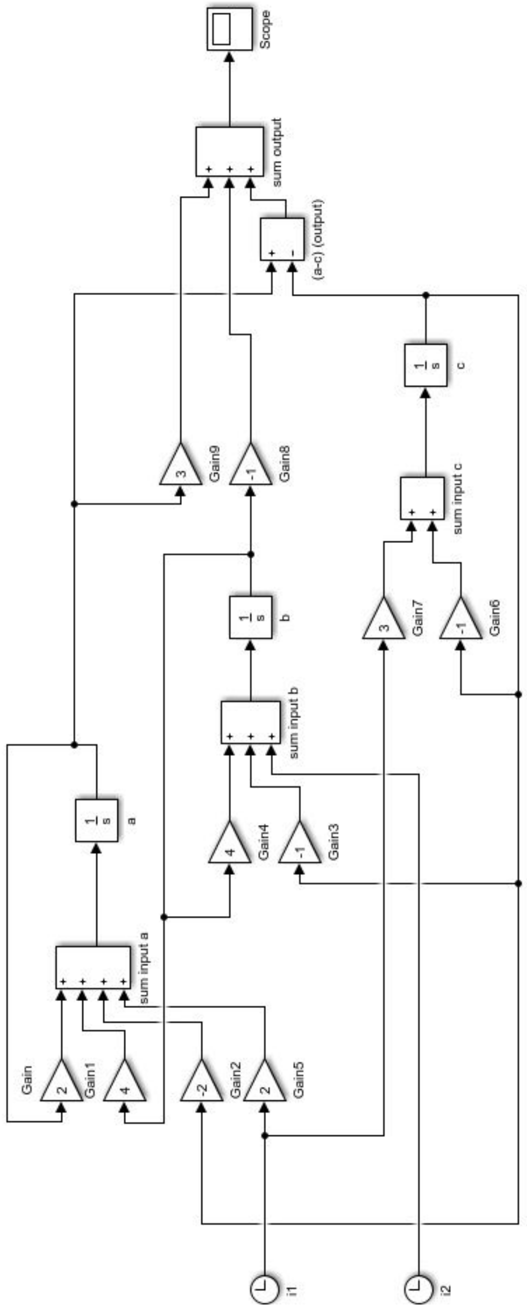


Abbildung 3: Blockschaltbild zum System aus Aufgabenstellung 1c

2 Kontinuierliche Simulation

Im Gegensatz zu linearen System können bei nicht linearen Systemen die Werte nicht zu jeden Zeitpunkt exakt berechnet werden. Daher wird bei nicht linearen Systemen (kontinuierliche Systeme) die Werteveränderung zum Zeitpunkt t annäherungsweise (numerische Integration) berechnet. Mit der numerischen Integration können alle kontinuierlichen Systeme, die durch Differenzialgleichungen dargestellt werden können, linear oder nicht, simuliert werden. Es können mit der numerischen Integration die Wertänderung zum Zeitpunkt t nur annäherungsweise berechnet werden, wodurch zwangsweise Fehler gemacht werden. Die Fehler hängen ab von der Schrittweite h und der gewählten Methode der numerischen Integration.

Ideal wäre die Schrittweite $h = 0$, da dadurch überhaupt keine Fehler gemacht werden würden, es würde aber unendlich viel Laufzeit in Anspruch nehmen und ist daher nicht in endlicher Zeit lösbar. Eine zu große Schrittweite wie z.B. $h = 10$ würde zu einen zu großen Fehler führen und ein Teil des Verhaltens des Systems könnte nicht erfasst werden. Auch ist eine über die Simulationsdauer gewählte Schrittweite h vielleicht auch nicht immer passend, wenn es z.B. wenig Veränderungen im System gibt, eine zu kleine Schrittweite unnötig ist und bei starken Veränderungen eine zu kleine Schrittweite zu wenig ist. Daher bitten einige Methoden der numerischen Integration wie z.B. die Methode nach *Runge Kutta* die Möglichkeit der Anpassung der Schrittweite während der Simulation an, wodurch die Simulation effizienter durchlaufen werden kann.

Wie schon erwähnt ist es nicht möglich eine Simulation eines nicht linearen Systems zu realisieren, ohne Fehler vollständig zu vermeiden. Es ist nur möglich den Fehler in einem Bereich zu halten, der für die Simulation akzeptabel ist. Die beiden Fehlertypen lokale Fehler und globale Fehler beschreiben den Fehler der entweder pro berechneten Schritt oder über die gesamte Simulationsdauer gemacht wird. Für eine Simulation ist der globale Fehler von größerem Interesse, da die pro Schritt gemachten Fehler bei einem akzeptablen globalen Fehler nicht mehr ins Gewicht fallen.

Die folgende Auflistungen zeigen die Methoden der numerischen Integration, die sich in die beiden Gruppen *Einschrittverfahren* und *Mehrschrittverfahren* aufteilen.

Einschrittmethoden:

1. Euler Integration
2. Methode von Heun
3. Runge Kutta

Mehrschrittmethoden:

1. Adams Bashford
2. Adams Moulton

Die beiden Gruppen *Einschritt-* und *Mehrschrittverfahren* unterscheiden sich im Bezug auf die einbezogenen Werte bei der Berechnung, wobei bei Einschrittverfahren immer nur einen Wert verwenden und die Mehrschrittverfahren vergangene und zukünftige Werte miteinbeziehen können. Die Wahl der richtigen Methode für die Simulation hängt ab von der Simulation selbst, der benötigten Fehlertoleranz und dem Laufzeitverhalten. Allgemein wird das Einschrittverfahren nach *Runge Kutta* als angemessen angesehen.

2.1 Euler Integration

Mit der Euler Integration $y_{i+1} = y_i + h * f(y_i)$, die ein Einschrittverfahren ist, wird die Wertänderung über ein rechtwinkliges Dreieck ermittelt. Dieses Verfahren ist das schnellste der Verfahren, da pro Schritt nur einmal die Funktion $f(y_i)$ aufgerufen werden muss. Der globale Fehler ($O(h)$) verhält sich linear zur gewählten Schrittweite.

Bsp.: $h = 10 \rightarrow O(h), h' = h/2 \rightarrow O(h') = O(h/2) =$

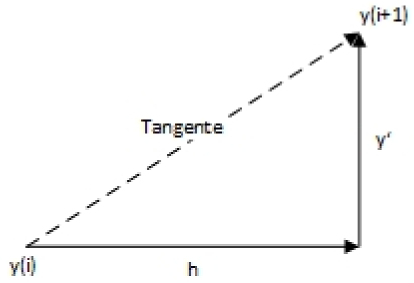


Abbildung 4: Dreieck zur Berechnung der Steigung der Wertänderung

2.2 Methode von Heun

Mit der Methode von Heun $y_{i+1} = y_i + h/2 * (f(y_i) + f(y_i + h * f(y_i)))$, die auch ein Einschrittverfahren ist, wird der nächste Wert im Gegensatz zur Euler Integration über ein Trapez ermittelt. Der globale Fehler ($O(h^2)$) verhält sich quadratisch zur gewählten Schrittbreite h.

Bsp.: $h = 10 \rightarrow O(h^2), h' = h/2 \rightarrow O(h') = O((h/2)^2)$

Der globale Fehler nimmt daher bei geringerer Schrittbreite deutlich mehr ab als bei der Euler Integration.

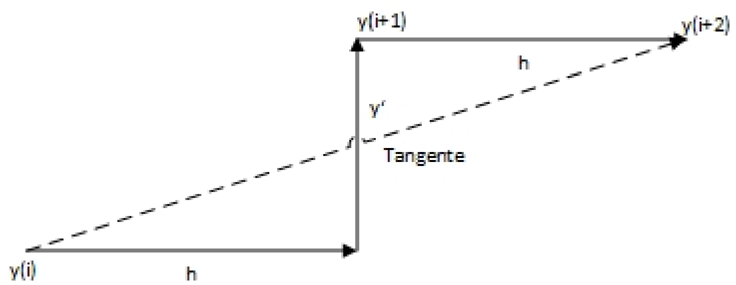


Abbildung 5: Trapez zur Berechnung der Steigung der Wertänderung

2.3 Runge Kutta

Die Methode von Runge Kutta ist ein vierstufiges Einschrittverfahren, das mit Schrittweitenanpassung arbeitet. Es wird ein *Threshold* des erlaubten Fehlers und die neue Schrittweite für den Fehlerfall definiert, die angewendet wird, wenn der *Threshold* überschritten wurde. Mit der dynamischen Anpassung der Schrittweite kann bei einer Simulation der Rechenaufwand bei wenig Änderungen vermindert werden. Nehmen die Änderungen zu, wird der Fehler bei der Berechnung größer, wodurch der *Threshold* überschritten wird und die Schrittweite auf den definierten Wert verkleinert wird. Damit wird der Fehler wieder kleiner, was mit einem Rechenaufwand einhergeht.

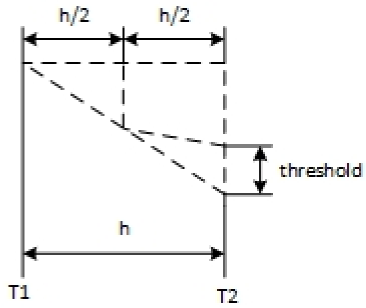


Abbildung 6: *Runge Kutta* Verfahren mit Schrittweitenanpassung

2.4 Mehrschrittverfahren

Bei dem Mehrschrittverfahren nach *Adams Bashford* werden nur vergangene Werte bei der Berechnung des nächsten Werts miteinbezogen. Bei dem Mehrschrittverfahren nach *Adams Moulton* werden zusätzlich zu den vergangen Werten auch angenommene Zukunftswerte bei der Berechnung des nächsten Werts miteinbezogen. Bei beiden Verfahren werden empirische Faktoren verwendet, die sich über die Zeit als gute Faktoren bewährt haben. Die Mehrschrittverfahren haben ein besseres Laufzeitverhalten sind aber schwerer zu implementieren.