

Aufgabe 1

Teilaufgabe 1.1

1.1 a

(1) Schnelligkeit (2) Genauigkeit (3) Stabilität

1.1 b

$$X_1 = G_1 \cdot G_2 \quad (1)$$

$$X_2 = G_3 \quad (2)$$

$$X_3 = G_4 + G_5 \quad (3)$$

1.1 c

$$L(2\ddot{x} + 3\dot{x} - 4x) = L(\dot{w} + w) \quad (4)$$

$$\Leftrightarrow 2 \cdot L(\ddot{x}) + 3L(\dot{x}) - 4L(x) = L(\dot{w}) + L(w) \quad (5)$$

$$\stackrel{?}{\Leftrightarrow} 2(s^2 X(s) - X(0) - x(0)) + 3(X(s) - x(0)) - 4X(s) = sW(s) - w(0) + W(s) \quad (6)$$

$$\Leftrightarrow X(s)(2s^2 - 1 - 2X(0) - 2x(0)) + 3(X(s) - x(0) - 4X(s)) = W(s)(s + 1) - w(0) \quad (7)$$

Annahme: $x(0) = 0, X(0) = 0, w(0) = 0$

Ist das
notwendig
/ gerecht-
fertigt?

$$X(s)(2s^2 - 1) = W(s)(s + 1) \quad (8)$$

$$\Leftrightarrow \frac{X(s)}{W(s)} = \frac{s + 1}{2s^2 - 1} = G(s) \quad (9)$$

1.1 d

Hurwitz-Kriterium

1.1 e

- Das D-Glied **differenziert** die **Regeldifferenz**.
- Durch Betrachtung der Änderung des Signals wird ein zukünftiger Trend berücksichtigt. Die D-Reglerstrategie ist: Je stärker die Änderung der Regelabweichung ist, desto stärker muss das Stellsignal verändert werden.
- D-Glieder **verbessern** gewöhnlich die **Regelgeschwindigkeit** und die dynamische Regelabweichung.
- D-Glieder verstärken besonders hochfrequente (verrauschte) Anteile des Eingangssignals. Dies erhöht die Neigung zu Schwingungen.

Teilaufgabe 1.2

1.2 a

- Einmaliger Hardware-Entwicklungsaufwand, dann sind parametrisierte Systemänderungen per Software möglich.
- Realisierung komplexer Reglerstrukturen

1.2 b

Bild mit Sollwert, Istwert, Regler $R(s)$, Taster T , Stracke $G(s)$ (TODO)

1.2 c

(Bild, TODO)

- 0T - 1T: 1
- 1T - 2T: 7
- 2T - 3T: 1
- 3T - 4T: 4
- 4T - 5T: 2

1.2 d

Differenzialgleichung (DGL): $2\ddot{x}(t) + 3\dot{x}(t) - x(t) = w(t)$

Differenzengleichung:

$$2 \frac{x(k) - 2x(k-1) + x(k-2)}{T_A^2} + 3 \frac{x(k) - x(k-1)}{T_A} - x(k) = w(k) \quad (10)$$

$$x(k) \left(\frac{2}{T_A^2 + \frac{3}{T_A}} - 1 \right) + x(k-1) \left(-\frac{2}{T_A^2} - \frac{3}{T_A} \right) + \frac{2}{T_A^2} x(k-2) = w(k) \quad (11)$$

Z-Transformierte

$$X(z)\left(\frac{2}{T_A^2} + \frac{3}{T_A} - 1 + z^{-1}\left(-\frac{2}{T_A^2} - \frac{3}{T_A} + z^{-2}\frac{2}{T_A^2}\right)\right) = W(z)$$

1.2 e

Z-Transformierte der DGL:

$$(z^{-1} + 1)X(z) = W(z) + Z(-w_{k+1}) \quad (12)$$

$$\stackrel{\text{Linksverschiebung}}{\Leftrightarrow} (z^{-1} + 1)X(z) = W(z) - z(W(z) - w_0 z^0) \quad (13)$$

$$= W(z) - zW(z) \quad (14)$$

$$= W(z)(1 - z) \quad (15)$$

$$\Leftrightarrow \frac{X(z)}{W(z)} = \frac{1 - z}{1 + z^{-1}} \quad (16)$$

1.2 f

Für die Polstellen $p_1 = 0.7$ und $p_2 = -0.25$ gilt: $|p_1| < 1$ und $|p_2| < 1 \Rightarrow$ Es ist stabil.

Eine mögliche zugehörige Übertragungsfunktion ist:

$$G(z) = \frac{(z + 0.5)(z + 1.5)}{(z - 0.7)(z + 0.25)}$$

Diese kann mit beliebigen, von 0 verschiedenen, Konstanten multipliziert werden und würde immer noch im selben Pol-Nullstellendiagramm resultieren.

Aufgabe 2

Teilaufgabe 2.1

2.1 a

- Datenspeicher
- Programm- und Konstantenspeicher
- nonvolatile RAM
- Wie kann nichtflüchtiger Speicher weiter unterschieden werden? **Einmal beschreibbar / Wiederbeschreibbar**

2.1 b

Folgende Techniken beschleunigen die Programmausführung in Mikroprozessoren: (1) Pipelining erhöht Befehlsdurchsatz (2) Spekulation (3) Speicherhierarchien

Kennzahlen und Argumentation (je weiter best und worst case auseinanderliegen, desto weniger echtzeitfähig):

- ohne Pipeline: best = worst = 15 Taktzyklen - echtzeitfähig
- einfache Pipeline: best = worst = 12 Taktzyklen - echtzeitfähig
- Pipeline + Spekulation: best = 9, worst = 15 - weniger echtzeitfähig
- Pipeline + Cache + Spekulation: best = 5, worst = 15 - noch weniger echtzeitfähig

2.1 c

Zur Wahrung der Echtzeitfähigkeit von System mit mehr als einem Busmaster müssen folgende Eigenschaften erfüllt sein:

1. Prioritäten: Jedem Busmaster muss eindeutig eine Priorität zugeordnet werden.
2. Preemption: Verlangt ein Busmaster mit höherer Priorität den Bus, so wird der mit niedriger Priorität unterbrochen.
3. Unterbrechbarkeit von Blocktransfers: Zur Vermeidung von Prioritäteninversion müssen Blocktransfers, d.h. Datentransfers in langen Blöcken unterbrochen werden können.
4. Busmonitor: Dient der Überwachung der Buszuteilungsregeln. Überschreiten ein Master das ihm zugeteilte Zeitkontingent, so muss der Busmonitor dies feststellen und Gegenmaßnahmen (z.B. Bus Error) einleiten.

2.1 d

Der PCI-Bus ist Echtzeitfähig, Synchron (Takt: 33 MHz–66 MHz), hat einen gemultiplexten Adress- und Datenbus und erlaubt Burst-Transfers.

2.1 e

Da die PCI-Bridge wie ein normales PCI-Device behandelt wird, können PCI-Bussysteme hierarchisch kaskadiert werden.

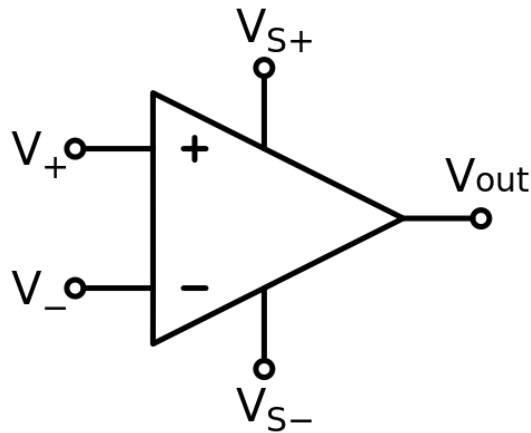


Abbildung 1: Quelle: commons.wikimedia.org/wiki/File:Op-amp_symbol.svg

Teilaufgabe 2.2

Teilaufgabe 2.2 a

In Abbildung 1 ist ein Operationsverstärker mit dem invertierenden Eingang V_- , dem nicht-invertierenden Eingang V_+ , sowie der positiven und negativen Versorgungsspannung (V_{S+} und V_{S-}) und der Ausgangsspannung V_{out} dargestellt.

(i) Eingangswiderstand $\rightarrow \infty$ (ii) Ausgangswiderstand $\rightarrow 0$ (iii) Verstärkung $\rightarrow \infty$

Teilaufgabe 2.2 b

Für nichtinvertierende Operationsverstärker gilt: $U_a = \frac{(R_e + R_n)}{R_e} U_e$ das Verhältnis soll $1,75 = \frac{7}{4}$ sein. Somit könnte $R_e = 4$ und $R_n = 3$ sein.

Zeichne also einen nichtinvertierenden Operationsverstärker mit 3 Widerständen (für R_n) in der Rückführung und 4 Widerstände für R_e .

Teilaufgabe 2.2 c

Differenzierer. TODO Zeichnung

Berechnung der Kapazität des Kondensators:

- Von $0 - 1s$ gilt $U_e = 2t$
- Von $1 - 2s$ gilt $U_e = 2$
- Von $2 - 3s$ gilt $U_e = -2t$

aus $U_a = -R_n C \frac{dU_e}{dt}$ folgt (Punkt 2 und 3 nur zur Verifizierung des Ergebnisses):

- $-1 = 1k\Omega C \frac{d^2t}{dt} \Rightarrow -1 = 2k\Omega C \Rightarrow C = -\frac{1}{2}10^{-3}F$
- $0 = 1k\Omega C \frac{d^2}{dt} \Rightarrow 0 = 0k\Omega C$
- $1 = 1k\Omega C \frac{d-2t}{dt} \Rightarrow 1 = -2k\Omega C \Rightarrow C = -\frac{1}{2}10^{-3}F$

Teilaufgabe 2.2 d

Zunächst wird die Eingangsspannung n_1 Takte lang integriert. Anschließend wird die negative Referenzspannung solange aufintegriert bis $U_i = 0$. Dabei wird die benötigte Anzahl an Takten n_2 gemessen. Schließlich gilt: $U_e = \frac{n_1}{n_2}U_{ref}$

Aufgabe 3

Teilaufgabe 3.1

Teilaufgabe 3.1 a

- Token-Passing
- Polling
- CSMA/CD
- CSMA/CA
- TDMA

Teilaufgabe 3.1 b

Maximale MMAT bei Token Passing: Anzahl der Teilnehmer + maximale Ausführungszeit jedes Teilnehmers

Teilaufgabe 3.1 c

Dekodiertes Signal: 0011 1010 10

Teilaufgabe 3.1 d

(11) 2 (12) 3 (13) 4 (14) 2 (15) 3 (16) sendet (17) 5

Teilaufgabe 3.1 e

Maximaler Füllstand des Paketpuffers bei TCP/IP-Kommunikation: TODO

Teilaufgabe 3.2

Teilaufgabe 3.2 a

Asynchrone und Synchrone Programmierung. Synchrone Programmierung wird für rein zeitbasierte Systeme eingesetzt.

- Vorteil Synchroner Programmierung:
 - Festes, vorhersagbares Zeitverhalten
 - Einfache Tests und Analyse des Systems
 - Rechtzeitigkeit und Gleichzeitigkeit können leicht garantiert werden.
- Nachteil Synchroner Programmierung:
 - keine Behandlung aperiodischer Ereignisse vorgesehen
 - geringe Flexibilität gegenüber Änderungen der Aufgabenstruktur

Teilaufgabe 3.2 b

Ein Schedulingverfahren heißt *optimal*, wenn es einen gültigen Schedule findet, falls dieser existiert.

- Optimal auf Einprozessorsystemen: GPS (Guaranteed Percentage Scheduling), LLF (Least Laxity First), EDF (Earliest Deadline First)
- Nicht optimal: FPP (Fixed Priority Preemptive) bei $H_n > n \cdot (2^{\frac{1}{n}} - 1)$, FPN (Fixed Priority Non-Preemptive)

also nicht
Mehrpro-
zessorsys-
teme?

Teilaufgabe 3.2 c

Nicht-Echtzeitsysteme: Logische Korrektheit

Echtzeitsysteme: Zusätzlich Zeitliche Korrektheit

Teilaufgabe 3.2 d

$$H_1 = 3/12 = 1/4 = 25\%$$

$$H_2 = 7/14 = 1/2 = 50\%$$

$$H_3 = 5/25 = 1/5 = 20\%$$

$H_{ges} = 95\% > 78\% = 3(n^{1/3} - 1)$, somit kann nicht garantiert werden, dass FPP ein Schedule findet

Teilaufgabe 3.2 e

- (1) Eingabedaten rechtzeitig holen
- (2) Ausgabedaten rechtzeitig berechnen und zur Verfügung stellen

Teilaufgabe 3.2 f

(1) Gleichzeitigkeit (2) Verfügbarkeit

Aufgabe 4

Teilaufgabe 4.1

Teilaufgabe 4.1 a

(1) Elementare Taskverwaltung (2) Interprozesskommunikation (3) Synchronisation

Teilaufgabe 4.1 b

RT-Linux

Teilaufgabe 4.1 c

TODO: Ablauf Sperrsynchronisation / Reihenfolgensynchronisation

Teilaufgabe 4.1 d

Middleware ist Software oberhalb des Betriebssystems. Sie sorgt in heterogenen und/oder verteilten Systemen für Transparenz.

Teilaufgabe 4.1 e

- 32 bit Prozessor \Rightarrow Logische Adresse ist 32 bit lang.
 - 1 GB = 2^{30} Byte \Rightarrow Physikalische Adresse ist 30 bit lang.
 - 4 kB = 2^{12} Byte \Rightarrow Beide Offsets sind 12 bit
- \Rightarrow 20 bit Seitenadresse, 18 bit Basisadresse

Teilaufgabe 4.2

4.2 a

Anzahl Gatter: 2

Ist das ein Betriebssystem-aufsatz? Was ist ein Betriebssystem-aufsatz?


```

      +---+
E0.2 +-----0+ |
      | & +---+
E0.1 +-----+ | | +-----+
      +---+ +---+ |
                        | ≥1 +-----+ A1.0
E0.3 +-----0+ |
      +---+

```

4.2 b

```

      E0.2      E0.1      A1.0
      | / |      | |      | - - |
+-----| / |-----| |-----+-----| - - |
      | / |      | |      | - - |
                        |
      E0.3      |
      | / |      |
+-----| / |-----+
      | / |

```

4.2 c

$A1.0 := ((E0.1 \ \&NOT \ E0.2) \ OR \ NOT \ E0.3)$

4.2 d

- Einlesen der Sensorsignale in Eingangsspeicher
- Aufrufen des Automatisierungsprogramms; Abarbeiten der Anweisungen
- Eingangssignale mit internen Zuständen und Ausgängen zu Ausgangssignalen verknüpfen