**ESS Section 4**

**Contents:**

**4. Control Loops**

4.1 *LTI* Feedback Loops 4.1.1 Linearity and Time-Invarianz (LTI) 4.1.1.1 Linearity

 4.1.1.2 Time-Invariance

 4.1.1.3 Causality

 4.1.1.4 Stability

 4.1.2 Time-Continuous LTI Systems 4.1.2.1 *Laplace* variable *s* → *Matlab:* tf( *av, bv* ) *→ Simulink*

 4.1.2.2 *Laplace* models → *Matlab*: bodeplot(…) *→ Simulink*

 4.1.3 Time-Discrete LTI Systems 4.1.3.1 Sampling Criteria of *Nyquist / Shannon*

 4.1.2.2 Phasor *z → Matlab: tf( dv, cv, Ts ), c2d, d2c*

 **4.1.3.**3 Time-Discr. Filters in 1st + 2nd Canonic Direct Structure

 4.1.4 Linear Feedback Loops 4.1.4.1 Signal and Noise Transfer Functions: *STF*, *NTF*

 4.1.4.2 *STF*, *NTF* Modeling Using *Matlab*

 4.1.4.3 The *Simulink* Using the Control Loop Template

4.2 *Fuzzy Logic* as Non-Linear Control Loop Example

**Procedure.** In class, this MS Word document is written using a tablet PC. Page breaks and/or subsection headings can be moved. After class, students are offered both the modified Word document and a PDF printout of it.

# Control Loops

## LTI Feedback Loops

### Linearity and Time Invariance (LTI)

#### Linearity

**y[ c1⋅x1(t) + c2⋅x2(t) ] = c1⋅y[ x1(t) ] + c2⋅y[x2(t)]. (2.1)**



**Fig. 2.1.1:** **(a)** linear superposition of two signals, **(b)** equivalent system.

Linearity for signal processing systems is defined according equation (2.1), illustrated by Fig. 2.1.1.

**Proportionality Implication.**

Setting c2=0 in equation (2.1) shows: Linearity implies proportionality:

**y[ c⋅x(t)] = c⋅y[x(t)] (2.2)**

as illustrated in Fig. 2.1.2. Proportionality allows to shift constants over LTI systems and therefore to combine several constants within the circuit mathematically to a single constant.



**Fig. 2.1.2:** Proportionality: Systems (a) and (b) are equivalent for linear circuits.

**Zero-Offset Implication.**

Setting c=0 in equation (2.2) shows: Proportionality implies zero offset:

**y[0] = 0⋅y[x(t)] = 0 (2.3)**

Conclusion:

The resistive divider in Fig. 2.1.3(a) is linear, because *U2* = constant ⋅ *U1*.

The circuit with operational amplifier in Fig. 2.1.3(b) is non-linear, as *U2*≠0 when *U1*=0.



**Fig. 2.1.3:** **(a)** resistive divider, **(b)** circuit using OpAmp with offset voltage *Uoff*≠0.

Remark: Linearity according to Eq. (2.1) is a signal processing definition. From a mathematical point of view a system *Uout = a⋅Uin + b* with constants *a, b* is linear.

#### Time Invariance

A system is time invariant when its impulse response *h*(*t*) is not a function of time:

**h(t) = h(t-τ) (2.4)**

|  |  |
| --- | --- |
| **Fig. 2.1.4:**Time-variant system when *Uctrl* varies with time |  |

Most systems we use are time-invariant. An example for a time-variant system is shown in Fig. 2.1.4, where response *Uout* to the impulses at *Uin* depends on control voltage *Uctrl*, which varies with time.

#### Causality: y(t) = f(x(τ)) with τ≤t

The present state of a system, *y*(*t*), is a function of the past and present state of its input, but not of future inputs.

#### Stability Definition : Bounded Input Bounded Output (BIBO)

Definition: There exist constants *M* and *K*, so that from |*x*(t)|≤*M* follows **|***y*[*x*(*t*)]**|** ≤ *K⋅M*.

**Question:** Is an ideal integrator BIBO stable? (Hint: consider *f*→0!)

### Time-Continuous LTI Systems

#### Laplace variable s → Matlab: tf( av, bv ), feedback(F,B)

#### Functions *step* and *impulse*

#### *Laplace* Models → *Matlab*: bodeplot(…) → Simulink

### Time-Discrete LTI Systems

#### Sampling Criteria of *Nyquist / Shannon*



**Fig. 4.1.3.1-A:** Sampling sinusoidal waveforms of with frequency *f#*, #=a, b, c, at different rates *fs*, then try to interpolate the samples. **(a)** *Fa= fa/fs*=¼, **(b)** *Fb= fb/fs*/ =½, **(c)** *Fc=fc/fs*=1.

Fig. 2.1.1 illustrates sinusoidal waveforms with different frequencies *f#*, sampled with rate *fS* corresponding to sampling interval and relative frequency, respectively:

*Ts = 1 / fs and F = f / fs*.

**Some public sampling rates:**

Old telephone: 8 KHz (±3dB band 300...3400Hz)

Audio data on CD: 44.1 KHz

Digital Video Broadcasting - Terrestrial (DVB-T): 13.3 MHz

**Exercise:**

Up to which relative frequency *F=f/fS* can curves in Fig. 2.1.1 be reconstructed from samples, when samples are interpolated with lowest possible frequency and amplitude?

According to the Nyquist-Shannon criterion, a sinusoidal wave can be reconstructed from samples, when the sampling frequency is at least twice the sinusoidal waveform's frequency. Consequently, the bandwidth that can be can be transmitted in sampled form, is

 ***fB ≤ ½ fS ⬄ fS ≥ 2 fB .*** (1)

During digital signal processing (DSP) we are rather interested in frequencies relative to the sampling rate ***fs*** than in physical frequencies. Consequently, we do not compute the true frequencies *f* and *ω* but relative frequencies

 ***F = f / fs = f∙Ts and Ω = ω / fs = ω Ts = 2π F.*** (2)

According to Shannon and Nyquist correct signal reconstruction requires

***F ≤ ½*** *<=>* ***Ω ≤ π*** *.* (3)

**Exercise 4.1.3.1-A:**

The figure below shows three sinusoidal waveforms. Vertical dashed lines are sampling time points. Draw sampled pulses and then reconstruct curves by interpolating with lowest-amplitude, lowest-frequency sinusoidal waves. Also compute relative frequencies *F* and Ω.



**Fig.:** Exercise: Sampling of different sinusoidal waveforms (solution see → Fig. 4.1.3.1-A).

**Problems with the Rule of Shannon and Nyquist**

Even when the Nyquist criterion is fulfilled, some problems may occur. In Fig. 2.1.1(b) the sampler picks samples exactly at the extrema of the curves. This cannot be guaranteed as illustrated in Fig. 2.1.2:

1. **Fig. part (a)** shows a waveform with frequency *fB(a)*=½*fS*, however, the sampler does not get the curve's extrema. Reconstruction delivers a signal with lower amplitude and phase shift.
2. **Fig. part (b)** shows a waveform with frequency *fB(b)*=½*fS*, however, the sampler gets the curve's zeros. Reconstruction delivers a zero signal.
3. **Fig. part (c)** illustrates sampling of a sinusoidal wave with *fB(c)<½fS*, the *Nyquist* criterion is fulfilled. Reconstruction delivers a signal with beats. This can be considered composed of the original frequency ½*fS*‑Δ*f* and a second signal with frequency½*fS*+Δ*f*. An extremely good filter could remove the upper signal, but this is costly.



**Fig. 4.1.3.1-B:** Problems close to Nyquist bandwidth *fB=fS*/2.

In practical applications, these problems can be overcome with increased sampling rate, e.g.

***fS* = 4...10 *fB***

corresponding to an „**O**ver **S**ampling **R**atio“

***OSR = fS/2fB* = 2...5** (4)

**Exercise 4.1.3.1-B:** the figure below shows three sinusoidal waveforms and vertical dashed lines as sampling time points. Draw the sampled pulses and then reconstruct curves from them interpolating with lowest-amplitude, lowest-frequency waves. Also compute the relative frequencies *F* and Ω.



**Exercise 4.2.3.1-B:** Sampling of frequencies close to *fS*/2. (Solution → Fig. 4.1.3.1)

Sampling is no lowpass filtering! Frequencies >½*fs* are not removed but appear at frequencies in the baseband, i.e. in range 0 ... ½*fs* according to

***falias = fin ‑ N⋅fs*** with *N* = round(*fin / fS*).

Do not worry about negative frequencies; they might indicate a phase shift, which does not matter when we have a frequency shift. (Remark: All real sinusoidal waves have their half energy at a positive and the other halt a negative frequency: cos(ωt)=½(ejωt+e‑jωt) and sin(ωt)=(ejωt‑e‑jωt)/(2j). If the positive frequency part becomes negative, then the corresponding negative part becomes positive.)

**Exercise 4.1.3.1-C:** You record music at a rate of *fS*=10KHz. The sampled accord contains 3 KHz, 6 KHz, 12 KHz and 24 KHz. Which frequencies do you hear when you hear your sampled data?

Solutions: 3 → 3, 6 → |6-10|=4, 12 → |12-10|=2, 24 → |24-2x10|=4

#### Phasor z → Matlab: tf( dv, cv, Ts ), c2d, d2c

While it is very difficult to time-delay continuous-time signals (e.g. for the PAL-TV system) without distortion, this is one of the easiest tasks in digital technology.

A delay *T* in the time domain appears as a linear phase shift φ=ω*T* in the frequency domain. The Fourier transform of a signal delayed by T is multiplied by the phase shifter:



In digital signal processing we work with the rotation factor , where T is the sampling interval and fs=1/T is the sampling frequency. Each delay of one clock period T corresponds to a multiplication by z 1 in the z domain. The relationship between the s of the Laplace transform and z is

  <=> .

These functions are uniquely invertible: each *s* value corresponds to exactly one *z* value and vice versa.



**Fig. 4.1.3.2:** Mapping between *s* an *z* plane.

#### Time-Discrete Filters in 1st and 2nd Canonic Direct Structure

FIR: Finite Impulse Response (non-recursive)

IIR: Infinite Impulse response (recursive)

canonic: Number of memory elements equals filter order

|  |  |
| --- | --- |
| 1. FIR filter in 1. canonic direct structure
 |  |

**(b)** FIR Filter in 2. canonic direct structure

**(c)** IIR Filter in 1. canonic direct structure



|  |  |
| --- | --- |
| 1. IIR filter in 2. canonic direct structure
 |  |

**Fig. 4.1.3.3:** Digital Filter structures

**Realizing digital filtering in software:**

Ansatz c0 yn + c1 yn-1 + … + cR yn-R = d0 xn + d1 yn-1 + … + dR yn-R

yields: **yn = ( d0 xn + d1 yn-1 + … + dR yn-R - c1 yn-1 - … - cR yn-R ) / c0**

with xn = x(n∙Ts), yn = y(n∙Ts), sampling interval *Ts* = 1 / *fs* , sampling frequency *fs*.

### Linear Feedback Loops

#### Signal and Noise Transfer Functions: *STF, NTF*

#### *STF, NTF* Modeling Using *Matlab*: feedback( F, B )

Matlab statement

**>> STF\_s = feedback(F, B)**

computes the *STF*(*s*) while

**>> NTF\_s = feedback(1, F\*B)**

computes the *NTF*(*s*).

#### The *Simulink* Control Loop Template

Type

**>> simulink**

in *Matlab R2022b*, click on *show more* and then on *Feedback Controller*. You should get the feedback control template as shown below.


## Fuzzy Logic as Non-Linear Control Loop Example

### Einleitung

Das englische Wort „fuzzy“ steht für „kraus, unscharf, verschwommen“. Fuzzy ist die Erweiterung der Logik auf unscharfe Zustände. In der Digitaltechnik kennt nur die Zustände „wahr“ (true) und „nicht wahr“ (false), die sich auch durch logisch 1 und logisch 0 ausdrücken und im dualen Zahlensystem verarbeiten lassen. Wie unterscheidet man jedoch ein handschriftlich unscharf geschriebenes „a“ von einem „d“? Das selbe Schriftzeichen wird unterschiedlich interpretiert, je nachdem, ob es in dem Wort „durch“ oder in dem Wort „auch“ gefunden wird. Beim Erblicken des Zeichens weisen wir ihm automatisch eine gewisse Wahrscheinlichkeit μ(Zeichen=a) zu, ein „a“ zu sein und die Wahrscheinlichkeit μ(Zeichen=d), ein „d“ zu sein. Die Wahrscheinlichkeit μ, Mitglied einer bestimmten Klasse zu sein, nennt man Fuzzy-Zugehörigkeitsfunktion (engl.: membership function or membership value). Statt des griechischen Buchstaben μ findet man bisweilen auch andere Buchstaben, typischerweise *W* (Wahrscheinlichkeit), *P* (Probability) oder A. Fuzzy-Experten legen Wert auf die Tatsache, daß sich nicht um Zufallsereignisse, nicht um eine Statistik, handelt. Die Fuzzy-Unschärfe bezieht sich auf die Überlappung der Zugehörigkeitsfunktionen.

Beispiel: Auf einer flachen Wiese befinden sich zwei Hügel und ein großer Felsbrocken. Wie beschreibt man diese Wiese auf einer linearen Skala mit den Kategorien: *flach, wellig, felsig, bergig*? Fuzzy kann sie beschreiben mit *μ(Wiese=flach)=0,7, μ(Wiese=wellig)=0,2, μ(Wiese=felsig)=0,1*, alle anderen *μ(\*=\*)* = 0.

**Fuzzy erlaubt die linguistische Darstellung von Expertenregeln.** Das macht es auch für Nichttechniker so interessant.

**Fuzzy-Entscheidungen fallen dreistufig:**

1. Fuzzyfizierung
2. Regelauswertung
3. Defuzzyfizierung

### Fuzzyfizierung (engl. Fuzzification)

Die Logik des Systematikers Aristoteles geht davon aus, daß Gegensätzliches nicht gleichzeitig möglich ist: Die Flamme einer Kerze ist entweder an oder aus, aber nie beides gleichzeitig. Dies ist die Grundlage der False/True-Logik, einem Geisteskind des von Aristoteles geprägten, abendländischen Denkens. Fuzzy-Logik dagegen ermöglicht die Behandlung eigentlich widersprüchlicher Aussagen und ist damit der klassisch indischen Denkweise sehr viel näher, nach welcher Gegensätze durchaus nebeneinander existieren können. Wann ist ein Auto noch blau? Ab welchem Farbton nennt man es grün? Warum sollte es nicht gleichzeitig grün und blau sein können? Als Beispiel betrachten wir die Frage, ob ein Glas halbvoll oder halbleer ist. Es sind zwei Zustände, die sich aufgrund der Fuzzyfizierung „halb“ nicht widersprechen, obwohl ein Glas nicht gleichzeitig voll und leer sein kann. Wir beantworten die Frage nach dem Füllstand des Glases mit Fuzzy-Zugehörigkeitsfunktionen, auch Klassen genannt. (In anderen Bereichen der Technik würde man solche Basisfunktionen als „Shape Functions“ bezeichnen.)

Eine Fuzzy-Klasse ist immer eindimensional, es können jedoch mehrere Klassen über der Abszisse dargestellt werden. Meistens sind es Rampen oder Dreiecks­funktionen. Fuzzy-Klassen sind im Beispiel des teilweise gefüllten Glases die Zugehörigkeitsfunktionen μ(Glas=voll) und μ(Glas=leer).



**Fig. 4.2.1:** Fuzzy-Zugehörigkeitsfunktionen bzw. Klassen definieren Zugehörig­keitsgrade als Antwort auf die Fragen: (i) wie voll ist ein Glas und (ii) wie leer ist es?

Als Fuzzy-Zugehörigkeitsfunktionen sind Rampenfunktionen wie μniedrig und Dreiecks­funktionen wie μmittel in Fig. 4.2.2 üblich, welche Kesseltemperaturen beschreiben. Diese Funktionen können auch durch Tabellen (Look-up-Tables) realisiert werden.



**Fig. 4.2.2:** Fuzzy-Zugehörigkeitsfunktionen bzw. Klassen definieren Zugehörigkeits­grade einer Kesseltemperatur mit Rampen- und Delta-Funktionen.

### Regel-Definition und Regel-Auswertung

#### Definition linguistischer IF- Regeln

Die Definition der Regeln muß vor dem Durchlaufen der Fuzzy-Logik erfolgt sein. Die Regeln definieren, wie die Fuzzy-Variablen zusammenhängen.

Nehmen wir als Beispiel einige Regeln für den Wasserkessel eines Kraftwerkes:

1. *IF temp = high AND pressure = high THEN energy\_input = small AND outlet = open*
2. *IF temp = medium AND pressure = high THEN energy\_input = high AND outlet = half*
3. *IF temp = low AND pressure = medium THEN energy\_input = high AND outlet = closed*
4. *....*

Es können beliebig viele Regeln definiert werden. Fuzzy-Logik schließt nicht einmal aus, daß diese Regeln widersprüchlich sein können. Kennzeichnend ist, daß sich Regeln überlappen. In eindeutig deterministischen Systemen müßten solche Situationen zu numerischen Problemen führen, in der Fuzzy-Logik werden sie akzeptiert.

Die Fuzzy-Eingabevariablen im Beispiel oben sind *temp* und *pressure*, Ausgabewerte sind *energy\_input, outlet*. Die zugeordneten Klassen sind *high, medium* und *low* für die Fuzzy-Variablen *temp* und *pressure*, sowie *small, high* für *energy\_input* und *open, half, closed* für *outlet*. *IF, AND, THEN* sind Fuzzy-Operatoren. Neben dem explizit ausgeschriebenen Operator *AND*, der innerhalb einer Regel angewendet wird, gibt es den Fuzzy-Operator *OR*, der implizit zwischen den Regeln gilt. Dies wird im Folgenden deutlich.

#### Regelauswertung (engl. Rule Inference)

Das englische Wort „inference“ läßt sich mit Schlußfolgerung übersetzen. Zuvor müssen die Fuzzy-Zugehörigkeitsgrade bekannt sein. Es gelte

 *μ(temp=low) = 0.3, μ(temp=medium) = 0.5, μ(temp=high) = 0.2*

 *μ(pressure=low) = 0.0, μ( pressure=medium) = 0.2, μ(pressure=high) = 0.8*

Damit entwickeln wir die erste Regel:

*IF temp = high AND pressure = high THEN ...*

 *temp = 0.2 pressure = 0.8 => min{0.2, 0.8} = 0.2*

 *... THEN energy\_input = small AND outlet = open*

 *=> μ(energy\_input=small) = 0.2*

 *=> μ(outlet=open) = 0.2*

Die Fuzzy-Operation *AND* in den *IF*-Abfragen wird meistens als Minimum-Funktion ausgewertet. Schlußfolgerung: Der Zugehörigkeitsgrad der gesamten *IF*-Abfrage ist gleich dem minimalen Zugehörigkeitsgrad der Teilabfragen. Die Operatoren *AND, OR, IF THEN* können auch anders definiert werden, z.B. das *AND* als Multiplikation. (Die Multiplikation mehrerer Werte ≤1 liefert ein Ergebnis ≤1.)

Wenn eine Anzahl von *N* Regeln (entsprechend N Zeilen) aufgestellt wurde, können für die Fuzzy-Variable *μ(outlet=open)* eine Anzahl von *M≤N* Zugehörigkeitswerten entwickelt werden. Diese *M* Werte werden *ODER* verknüpft, was einer Maximum-Funktion entspricht.

Beispiel: Zugehörigkeitsfunktion *μ(outlet=open)* werden in 4 Regeln (also in 4 veschiedenen Zeilen) die Erfüllungsgrade 0.2, 0.5, 0.1, 0.3 zugeordnet. Diese Zeilen werden nun vertikal mit der *ODER*-Funktion ausgewertet, wobei *ODER* hier als Maximum realisiert wird:

 *μ(outlet=open) = max{0.2, 0.5, 0.1, 0.3} = 0.5 .*

### Defuzzyfizierung (engl. Defuzzyfication)

Zu ermitteln sind die Ausgangswerte *energy\_input* und *outlet*. Gegeben sind die Erfüllungs­grade

 *μ(energy\_input=low) = 0.3, μ( energy\_input=high) = 0.5*

 *μ(outlet=closed) = 0.3, μ(outlet=half) = 0.6, μ(outlet=open) = 0.2*

Die entsprechenden Gewichtsfunktionen *μ*(*<Ausgangswert> = <Klasse>*) werden nun auf der ihnen zugeordneten Höhe abgeschnitten und die unter dem Schnitt verbleibenden Flächen gemäß Fig. 4.2.3 zu einer einzigen Fläche vereint. Der Flächenschwerpunkt auf der Abszisse (hier y-Achse) bildet das gesuchte Ergebnis für den Ausgangswert.

Neben dieser Standardversion der Defuzzyfizierung kann man auch andere Techniken wählen. Statt die Gewichtsfunktionen bei dem Wert *μ(...)* abzuschneiden, kann man ihre Ordinate auch mit diesem Wert multiplizieren, wie dies in Fig. 4.2.4 gezeigt wird. Ferner kann man den Gewichtsfunktionen noch eine weitere Dimension (entsprechend einer Höhe(x,y) ) zuordnen und den Schwerpunkt des Volumens durch Integration ermitteln. Der Wissenschaft sind hier viele Ansatzpunkte gegeben. In der Praxis haben sich vor allem die beiden unten graphisch dargestellten Verfahren durchgesetzt.

Vergleicht man einen Standardregler mit einem Fuzzy-Regler, dann sind die Unterschiede so groß, daß ein direkter Vergleich schwerfällt. Der Versuch eines Vergleichs ist daher notwendigerweise hinkend und unvollkommen.

1. Der Fuzzy-Regler ermöglicht eine linguistische Eingabe der Regeln für die Regelauswertung. Beispielsweise nutzen Förster Fuzzy-Logik, um mit Hilfe linguistischer Geländebeschreibungen die optimale Bepflanzung zum Schutz einer Futterstelle oder Anbaufläche vor Wind und Schnee zu ermitteln. Eine solche Optimierung wäre mit einem klassischen, mathematischen Verfahren zumindest problematisch.
2. Die nichtlineare Handhabung vieler Ein- und Ausgangsgrößen ist mit einem Fuzzy-Regler sehr leicht und natürlich machbar.
3. Lineare Regler setzen voraus, daß der Ausgang des Reglers nicht durch seine Aussteuer­grenzen (z.B. Betriebsspannungen) beeinträchtigt wird. Dadurch ist für Sprünge in der Führungsgröße eine Steuerreserve nötig. Nimmt man Nichtlinearitäten durch die Aussteuergrenzen in Kauf, dann muß mit aufwendigen, nichtlinearen Algorithmen gearbeitet werden. Diese Probleme kennt der Fuzzy-Regler nicht.
4. Der Vorteil der exakten Ermittelbarkeit der Stabilität des Regelkreises ist für den klassischen Regler nur dann gegeben, wenn ein geeignetes Modell für die Regelstrecke vorliegt. Ansonsten gilt: Die Dynamik des Reglers sollte kleiner sein als die Dynamik der Strecke.



**Fig. 4.2.3:** Fuzzy-Ausgangswerte werden durch Ermittlung des Flächen­schwerpunktes S der abgeschnittenen Gewichts­funktionen ermittelt.



**Fig. 4.2.4:** Fuzzy-Ausgangswerte werden in dieser Variante durch Ermittlung des Flächenschwerpunktes *S* der skalierten Gewichtsfunktionen ermittelt.

Ob ein PID-Regler oder ein Fuzzy-Regler vorteilhaft ist, kann letztlich nur situativ beantwortet werden. Zur Regelung von Kraftwerken, wo jedes Prozent Energieausbeute wertvoll ist, wird typischerweise noch mit klassischen Reglern gearbeitet. Entworfen werden diese von hoch spezialisierten Experten.

## References

1. *Wikipedia, Fuzzy Logic,*  available: <https://en.wikipedia.org/wiki/Fuzzy_logic>