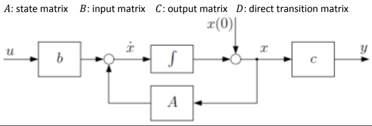
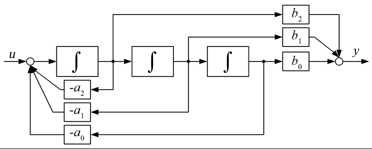
1. State Space Models (SSM)

$\dot{x}(t) = Ax(t) + Bu(t)$ (State Equation)
$y(t) = Cx(t) + Du(t)$ (Output Equation)
A: state matrix B: input matrix C: output matrix D: direct transition matrix



Chain of Integrators

Für $G(s) = \dfrac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$ gilt folgendes Schaltbild



Transfer Function -> State Space Model

$Y(s) = G(s)U(s)$

$$G(s) = \frac{b(s)}{a(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0}$$

$$G(s) = d + \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

Controller Canonical Form

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}; \qquad b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$c = [b_0 \quad b_1 \quad b_2]$; $d = d$

Observer Canonical Form

$$A = \begin{bmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{bmatrix}; \qquad b = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

$c = [0 \quad 0 \quad 1]$; $d = d$

State Space Model -> Transfer Function

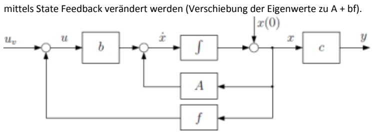
Laplace Transformation des SSM:
$sX(s) = AX(s) + bU(s)$
$Y(s) = cX(s) + dU(s)$
State Equation nach $X(s)$ umstellen und in Output Equation einsetzen:
$Y(s) = (c(sI - A)^{-1}b + d)U(s) = G(s)U(s)$

$$G(s) = c(sI - A)^{-1}b + d = \frac{c\ adj(sI - A)b}{det(sI - A)}$$

Static Gain $s = 0 \rightarrow G(0)$
Charakteristisches Polynom $a(s) = det(sI - A) = 0$
Die Pole, die das dynamische Verhalten des Systems charakterisieren, sind die Eigenwerte von A.
Charakteristisches Polynom Open Loop
$a(s) = \det(sI - A) = s^2 + 2\zeta\omega_n s + {\omega_n}^2$ (2. Ordnung)
$K = G(0)$ $\omega_n = \sqrt{a_0}$ $\zeta = \frac{a_1}{2\omega_n}$
$T_s \cong \frac{4{,}6}{\omega_n \zeta}$ Settling Time $T_r = \frac{1{,}7}{\omega_n}$ Rise Time
$M_P \cong 1 - \frac{\zeta}{0{,}6}$ Peak Overshoot

Changing Eigenvalues via State Feedback

Dynamische Eigenschaften des Systems (Eigenwerte des Systemmatrix A) können mittels State Feedback verändert werden (Verschiebung der Eigenwerte zu A + bf).



Der Eingang wird zu:
$u(t) = fx(t) + u_v(t)$ $f = [f_1 \quad f_2 \quad \ldots \quad f_n]$
Das SSM wird zu:
$\dot{x}(t) = Ax(t) + b(fx(t) + u_v(t)) = (A + bf)x(t) + bu_v(t)$
$y(t) = cx(t) + du(t)$

Non-Uniqueness of SSM and Similarity Transformation

Ein gegebenes SSM kann mittels Koordinatentransformation in ein anderes SSM überführt werden. Die SSM beschreiben dabei weiterhin das gleiche System.
Transformation T mit $\det T \neq 0$
$x(t) = T\tilde{x}(t)$
$(A, b, c, d) \xrightarrow{T} (\tilde{A}, \tilde{b}, \tilde{c}, d)$
(d unverändert weil keinen Bezug zu den State Variables)
$\tilde{A} = T^{-1}AT$ $\tilde{b} = T^{-1}b$ $\tilde{c} = cT$
Diagonal Form
Wenn in T die Eigenvektoren von A stehen und kein Eigenwert doppelt auftritt, dann wird A auf Diagonalform gebracht. Auf der Diagonalen stehen dann die Eigenwerte.

Solution of State Equation and Matrix Exponentials

Zur Lösung der Zustandsgleichung ist eine Matrixexponentialfunktion notwendig:

$$e^{At} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \cdots$$

$$\frac{d}{dt}e^{At} = Ae^{At}$$

Solution to State Equation:

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}bu(\tau)d\tau$$

mit Transition Matrix $\Phi(t) = e^{At}$:

$$x(t) = \Phi(t)x_0 + \int_0^t \Phi(t-\tau)bu(\tau)d\tau$$

$x(t) = free\ response + forced\ response$
Non-Zero Initial Condition
$X(s) = \Phi(s)x_0 + \Phi(s)bU(s)$ Laplace von $x(t)$
$\Phi(s) = \mathcal{L}[\Phi(t)\sigma(t)] = (sI - a)^{-1}$ Laplace Transition Matrix
Es gilt:
$\Phi^{-1}(t) = e^{-At} = \Phi(-t)$
$A\Phi(t) = \Phi(t)A$

Stability

Ein System ist stabil wenn sich alle Eigenwerte der Systemmatrix A in der linken Halbebene befinden.

Cayley-Hamilton Theorem

Wenn die Charakteristische Gleichung der Matrix A
$\det(sI - A) = a(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 = 0$
ist, dann folgt:
$A^n + a_{n-1}A^{n-1} + \cdots + a_1 A + a_0 I = 0$
Die Transition Matrix kann geschrieben werden als:
$\Phi(t) = \alpha_0(t)I + \alpha_1(t)A + \alpha_2(t)A^2 + \cdots + \alpha_{n-1}(t)A^{n-1} = e^{At}$
Durch Einsetzen der Eigenwerte lassen sich die α berechnen.

$$\begin{bmatrix} \alpha_0(t) \\ \alpha_1(t) \\ \vdots \\ \alpha_{n-1}(t) \end{bmatrix} = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} e^{\lambda_1 t} \\ e^{\lambda_2 t} \\ \vdots \\ e^{\lambda_n t} \end{bmatrix}$$

2. Controllability & Pole Placement

Controllability Gramian

$$W_C(t) = \int_0^t e^{A\tau}bb^Te^{A^T\tau}d\tau$$

System ist controllable, wenn $W_C(t)$ positiv definit für alle t > 0 ist. Wenn $W_C(t)$ für ein t > 0 positiv definit ist, dann ist $W_C(t)$ für alle t > 0 positiv definit.

Controllability Matrix

Controllability Matrix entsprechend Dimension von A. Hat $C(A, b)$ vollen Rang, dann ist das System controllable.
$C(A, b) = [A^0b \quad A^1b \quad A^2b \quad \ldots \quad A^{n-1}b]$ n=n
$C(A, b) = [b \quad Ab]$ n=2
$C(A, b) = [b \quad Ab \quad A^2b]$ n=3
Controllable Subspace
Der Controllable Subspace ist die Menge, die ein uncontrollable System erreichen kann. Diese Menge wird von den linear unabhängigen Spalten der C Matrix aufgespannt. Für z.B. $[1\ 2]^T$ ist es eine Gerade mit Steigung 2. Bei vollständig steuerbaren Systemen ist dies der gesamte Zustandsraum. Der Uncontrollable Subspace liegt senkrecht dazu.
Stabilizability
Ein nicht stabiles System ist stabilisierbar, wenn ein f existiert, so dass mittels State Feedback $u(t) = fx(t)$ das resultierende System stabil ist.
Wenn das System (A, b, c, d) instabil ist und die Controllability Zerlegung vorliegt, dann ist das System nur stabilisierbar, wenn $\bar{A}_{\bar{c}}$ keine Eigenwerte in der rechten Halbebene hat.

PBH-Test

Das System (A, B) ist genau dann vollständig steuerbar, wenn die Bedingung $Rang(\lambda I - A \quad B) = n$ für alle komplexen Werte λ erfüllt ist.Wenn λ kein Eigenwert der Matrix A ist, so ist das Kriterium erfüllt. Die Bedingung muss also nur für die Eigenwerte der Matrix A überprüft werden. Somit zeigt das Kriterium direkt, welche Eigenwerte ggf. nicht controllable sind. Wenn voller Rang für $\lambda > 0$ gegeben ist, dann ist das System stabilisierbar.

Pole Placement

Wenn ein System controllable ist, dann können mittels State-Feedback die closed-Loop Pole an beliebige Stellen gelegt werden. State-Feedback ändert die Nullstellen nicht.
Die vorgegebenen Pole e_1, e_2, e_3 erzeugen ein Polynom
$P(s) = (s - e_1)(s - e_2)(s - e_3)$
$P(s) = s^3 + \bar{a}_2 s^2 + \bar{a}_1 s + \bar{a}_0$
$\bar{a}_0 = e_1 e_2 e_3$
$\bar{a}_1 = e_1 e_2 + (e_1 + e_2)e_3$
$\bar{a}_2 = -e_1 e_2 e_3$
Für zwei vorgegebene Pole (System 2. Ordnung):
$\bar{a}_0 = e_1 e_2$
$\bar{a}_1 = -(e_1 + e_2)$
Aus A ist das charakteristische Polynom bestimmt worden:
$P(A) = s^3 + a_2 s^2 + a_1 s + a_0$
Es entsteht ein Zeilenvektor:
$p = [\bar{a}_2 - a_2 \quad \bar{a}_1 - a_1 \quad \bar{a}_0 - a_0]$
Controllability Matrix und Toeplitzmatrix lauten:

$$C = [b \quad Ab \quad A^2b] \qquad T_a = \begin{bmatrix} 1 & a_2 & a_1 \\ 0 & 1 & a_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Danach lässt sich der benötigte Vektor f berechnen:
$f = -pT_a^{-1}C^{-1}(A, b)$ (Bass-Gura)
Für Pole Placement eines Observers gilt:
$l = -\left(pT_a^{-1}C^{-1}(A^T, c^T)\right)^T$

Uncontrollable Systems

Ist das System (A, b, c, d) uncontrollable, also $C(A, b)$ hat nicht vollen Spalten-Rang $rank[C(A, b)] = r < n$, dann gibt es eine Transformationsmatrix T_c die das System in ein controllable und ein uncontrollable Subsystem zerlegt:
$T_c = [b \quad Ab \quad \ldots \quad A^{r-1}b \quad q_{r+1} \quad q_n]$
T_c enthält in den ersten Spalten die linear unabhängigen Spalten von $C(A, b)$ und wird mit $n - r$ linear unabhängigen Vektoren (orthogonal zu den anderen) aufgefüllt.
Die Transformation auf das System angewandt liefert:
$\bar{A} = {T_c}^{-1}AT_c$;
$\bar{b} = {T_c}^{-1}b$
$\bar{c} = cT_c$
$C(\bar{A}, \bar{b}) = C({T_c}^{-1}AT_c, {T_c}^{-1}b) = {T_c}^{-1}C(A, b)$
Die Controllability Zerlegung ergibt nach der Transformation:

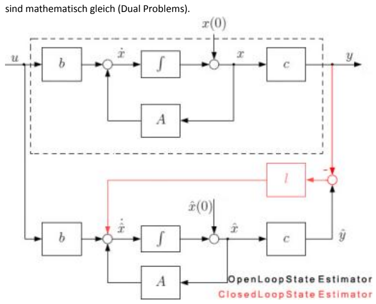
$$\begin{bmatrix} \dot{\bar{x}}_c \\ \dot{\bar{x}}_{\bar{c}} \end{bmatrix} = \begin{bmatrix} \bar{A}_c & \bar{A}_{12} \\ 0 & \bar{A}_{\bar{c}} \end{bmatrix} \begin{bmatrix} \bar{x}_c \\ \bar{x}_{\bar{c}} \end{bmatrix} + \begin{bmatrix} \bar{b}_c \\ 0 \end{bmatrix} u$$

$$y = [\bar{c}_c \quad \bar{c}_{\bar{c}}] \begin{bmatrix} \bar{x}_c \\ \bar{x}_{\bar{c}} \end{bmatrix}$$

3. Observability and State Estimation

State Estimation

Für State Feedback müssen alle State Variablen bekannt sein. Können diese nicht alle gemessen werden, dann können sie mithilfe eines State Estimators abgeschätzt werden. Berechnung eines State Estimators und die Berechnung von State Feedback sind mathematisch gleich (Dual Problems).



Open-Loop estimated State Vector:
$\dot{\hat{x}}(t) = A\hat{x}(t) + bu(t)$ $\hat{x}(0) = \hat{x}_0$
estimation error dynamics ($\tilde{x}(t) = x(t) - \hat{x}(t)$):
$\dot{\tilde{x}}(t) = A\tilde{x}(t)$, $\tilde{x}(0) = x(0) - \hat{x}(0)$
Closed-Loop (State Observer) estimated State Vector:
$\dot{\hat{x}}(t) = A\hat{x}(t) + lc(\hat{x}(t) - x(t)) + bu(t)$ $\hat{x}(0) = \hat{x}_0$
estimation error dynamics ($\tilde{x}(t) = x(t) - \hat{x}(t)$):
$\dot{\tilde{x}}(t) = (A + lc)\tilde{x}(t)$, $\tilde{x}(0) = x(0) - \hat{x}(0)$
Wenn das System observable ist, können mittels l die Eigenwerte des State Estimators beliebig plaziert werden. Berechnung von l identisch zu Pole Placement bis auf einige Änderungen:
$A \rightarrow A^T$, b→ c^T, f→ l^T
$l = -\left(pT_a^{-1}C^{-1}(A^T, c^T)\right)^T$
Zur p Vektor Bestimmung wird das characteristische Polynom des Observers anstatt des angestrebten Systems benutzt:
$\bar{a}_o(s) = det(sI - A - lc)$

Observability Matrix

Observability Matrix entsprechend Dimension von A. Hat $O(c, A)$ vollen Rang, dann ist das System observable.

$$O(c, A) = C^T(A^T, c^T) = \begin{bmatrix} c \\ cA \\ \vdots \\ cA^{n-1} \end{bmatrix} \quad \text{n=n}$$

$$O(c, A) = \begin{bmatrix} c \\ cA \end{bmatrix} \quad \text{n=2}$$

$$O(c, A) = \begin{bmatrix} c \\ cA \\ cA^2 \end{bmatrix} \quad \text{n=3}$$

Äquivalente Aussagen zur Observability
- Das System (c, A) ist observable
- Das System (A^T, c^T) ist controllable
- Das Observability Gramian ist positive definit für $t > 0$

$W_O(t) = \int_0^t e^{A^T\tau}c^Tce^{A\tau}d\tau$
- Die Observability Matrix hat vollen Rank.
- Die Matrix hat $\begin{bmatrix} sI - A \\ c \end{bmatrix}$ vollen Rank für alle s (Dual of PBH Test)

Unobservable Systems

Ist das System A,b,c,d nicht observable, also $O(c, A)$ hat nicht vollen Zeilen-Rang $rank[O(c, A)] = r < n$, dann gibt es eine Transformationsmatrix T_o die das System in ein observable und ein unobservable Subsystem zerlegt:

$$T_o = \begin{bmatrix} c \\ cA \\ \vdots \\ cA^{r-1} \\ q_{r+1} \\ q_n \end{bmatrix}$$

T_o enthält in den ersten Zeilen die linear unabhängigen Zeilen von $O(c, A)$ und wird mit n-r linear unabhängigen Vektoren (orthogonal zu den anderen) aufgefüllt.
Die Transformation auf das System angewandt liefert:
$\bar{A} = {T_o}^{-1}AT_o$; $\bar{b} = {T_o}^{-1}b$; $\bar{c} = cT_o$
$O(\bar{c}, \bar{A}) = O(cT_o, {T_o}^{-1}AT_o) = O(c, A){T_o}^{-1}$
Die Observability Zerlegung ergibt nach der Transformation:

$$\begin{bmatrix} \dot{\bar{x}}_o \\ \dot{\bar{x}}_{\bar{o}} \end{bmatrix} = \begin{bmatrix} \bar{A}_o & 0 \\ \bar{A}_{21} & \bar{A}_{\bar{o}} \end{bmatrix} \begin{bmatrix} \bar{x}_o \\ \bar{x}_{\bar{o}} \end{bmatrix} + \begin{bmatrix} \bar{b}_o \\ \bar{b}_{\bar{o}} \end{bmatrix} u$$

$$y = [\bar{c}_o \quad 0] \begin{bmatrix} \bar{x}_o \\ \bar{x}_{\bar{o}} \end{bmatrix}$$

Observable Subspace
Der Observable wird von den linear unabhängigen Zeilen der O Matrix aufgespannt. Bei vollständig beobachtbaren Systemen ist dies der gesamte Zustandsraum. Der NullSpace der Observability Matrix ist der unobservable Subspace, der RowSpace ist der observable Subspace. Der unobservable Subspace liegt senkrecht auf dem observable Subspace.
Detectability
Ein System heißt detectable wenn es einen Vektor l gibt, so dass alle Eigenwerte von $A + lc$ in der linken Halbebene liegen. Wenn das System unobservable ist, dann darf $\bar{A}_{\bar{o}}$ keine Eigenwerte in der rechten Halbebene haben.

Minimales System

Ein System ist minimal wenn es controllable und observable ist!

Kalman Canonical Decomposition

Wenn ein System uncontrollable und unobservable ist, dann existiert eine Transformation die zu folgendem System führt:

$$\begin{bmatrix} \dot{\bar{x}}_{co} \\ \dot{\bar{x}}_{c\bar{o}} \\ \dot{\bar{x}}_{\bar{c}o} \\ \dot{\bar{x}}_{\bar{c}\bar{o}} \end{bmatrix} = \begin{bmatrix} \bar{A}_{co} & 0 & \bar{A}_{13} & 0 \\ \bar{A}_{21} & \bar{A}_{c\bar{o}} & \bar{A}_{23} & \bar{A}_{24} \\ 0 & 0 & \bar{A}_{\bar{c}o} & 0 \\ 0 & 0 & \bar{A}_{43} & \bar{A}_{\bar{c}\bar{o}} \end{bmatrix} \begin{bmatrix} \bar{x}_{co} \\ \bar{x}_{c\bar{o}} \\ \bar{x}_{\bar{c}o} \\ \bar{x}_{\bar{c}\bar{o}} \end{bmatrix} + \begin{bmatrix} \bar{b}_{co} \\ \bar{b}_{c\bar{o}} \\ 0 \\ 0 \end{bmatrix} u$$

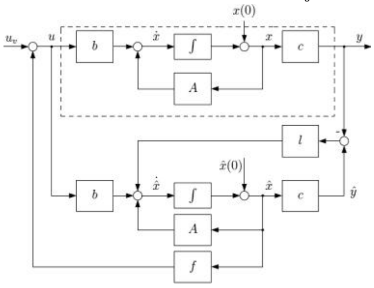
$$y = [\bar{c}_{co} \quad 0 \quad \bar{c}_{\bar{c}o} \quad 0] \begin{bmatrix} \bar{x}_{co} \\ \bar{x}_{c\bar{o}} \\ \bar{x}_{\bar{c}o} \\ \bar{x}_{\bar{c}\bar{o}} \end{bmatrix}$$

Das Input-Output Verhalten des Systems wird allein durch das Subsystem $(\bar{A}_{co}, \bar{b}_{co}, \bar{c}_{co})$ bestimmt. Die Transfer Funktion lautet:
$G(s) = c(sI - A)^{-1}b = \bar{c}_{co}(sI - \bar{A}_{co})^{-1}\bar{b}_{co}$

4. Observer-Based Control

State Estimation Feedback

Feedback wird mit dem estimated Statevektor $\hat{x}$ des Observers durchgeführt



Die State Variablen des Closed-Loop Systems sind die State Variablen von Plant und Observer, somit ergibt sich das StateSpace Model zu:

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & bf \\ -lc & A + bf + lc \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} b \\ b \end{bmatrix} u_v, \quad \begin{bmatrix} x(0) \\ \hat{x}(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ \hat{x}_0 \end{bmatrix}$$

$$y = [c \quad 0] \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

Ersetzen des Observer State Vektors $\hat{x}$ durch den Estimation Error $\tilde{x}$ durch Transformation des Systems mit

$$\begin{bmatrix} x \\ \hat{x} \end{bmatrix} = T \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}, \quad T = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$$

führt zu folgendem System:

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A + bf & -bf \\ 0 & A + lc \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} u_v, \quad \begin{bmatrix} x(0) \\ \hat{x}(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ \tilde{x}_0 \end{bmatrix}$$

$$y = [c \quad 0] \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}$$

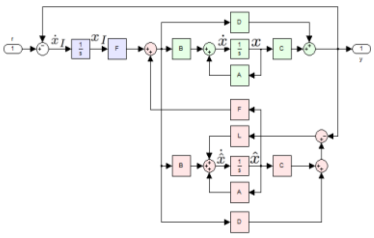
Die Eigenwerte des Closed-Loop System sind die Eigenwerte des State-Feedback $(A + bf)$ und des Observers $(A + lc)$. State-Feedback und Observer können also unabhängig voneinander entworfen werden (Separation Principle). $\tilde{x}$ ist uncontrollable von u_v, x ist controllable wenn (A, b) controllable ist.

Spezialfall mit Integrator und Gain F_I vorm Eingang u:

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \\ \dot{x}_I \end{bmatrix} = \begin{bmatrix} A & bf & bF_I \\ -lc & A + bf + lc & bF_I \\ -C & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \\ x_I \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b \end{bmatrix} r$$

$$y = [C \quad 0 \quad 0] \begin{bmatrix} x \\ \tilde{x} \\ x_I \end{bmatrix}$$

Spezialfall mit Durchgriff:



$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \\ \dot{x}_I \end{bmatrix} = \begin{bmatrix} A & BF & BF_I \\ -LC & A + BF + LC - LDF & BF_I - LDF_I \\ -C & -DF & -DF_I \end{bmatrix} \begin{bmatrix} \dot{x} \\ \hat{x} \\ x_I \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r$$

$$y = [C \quad DF \quad DF_I] \begin{bmatrix} \dot{x} \\ \hat{x} \\ x_I \end{bmatrix}$$

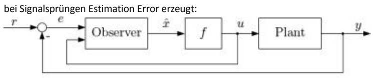
Übertragungsfunktion Closed-Loop
$G_{cl}(s) = c(sI - A - bf)^{-1}b$

Choice of Observer Eigenvalues

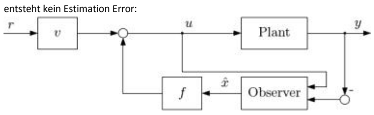
- Eigenwerte bestimmen Abklingzeit des Estimation Errors
- Eigenwerte sollten links von den Plant Eigenvalues liegen um deren Entwicklung zu verfolgen
- Zu weit links verstärkt das Rauschen zu sehr

Reference Tracking

Verschiedene Möglichkeiten ein Referenzsignal ins System zu geben, wobei die erste bei Signalsprüngen Estimation Error erzeugt:



Die zweite Möglichkeit gibt die Signale gleichzeitig an Plant und Observer, dadurch entsteht kein Estimation Error:



Zeros of Transfer Functions
Für eine Nullstelle (Transmission Zero) gilt

$$G(s) = \frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_1)} = 0$$

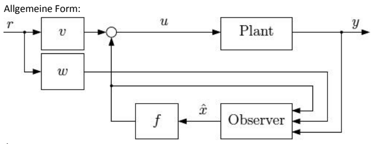
Zeros of State Space Models
Eine komplexe Zahl s ist eine Nullstelle (Invariant Zero) wenn gilt:

$$det \begin{bmatrix} sI - A & -b \\ c & d \end{bmatrix} = 0$$

Ist ein State Space Modell minimal, dann sind Invariant und Transmission Zeros gleich, hat ein Modell uncontrollable und unobservable Modes, existieren Invariant Zeros die nicht als Transmission Zeros auftauchen.

Closed Loop Transfer Function

Allgemeine Form:



$\dot{\hat{x}} = (A + bf + lc)\hat{x} - ly + wr$
$u = f\hat{x} + vr$
w und v haben keinen Einfluss auf die Closed-Loop Eigenvalues, aber auf die Closed-Loop Zeros. Bedingung für Zeros ($y = 0$ wegen Interesse an Zeros von r nach u):

$$det \begin{bmatrix} sI - A - bf - lc + \frac{1}{v}wf & -\frac{1}{v}w \\ 0 & 1 \end{bmatrix} = 0$$

$$det\left(sI - A - bf - lc + \frac{1}{v}wf\right) = 0$$

Nachdem f und l festgelegt sind können die Closed-Loop Zeros mit v und w festgelegt werden wenn $\left(-\frac{w}{v}, A + bf + lc\right)$ observable ist.
Closed-Loop Übertragungsfunktion

$$G_{cl}(s) = K\frac{det\left(sI - A - bf - lc + \frac{1}{v}wf\right)b(s)}{det(sI - A - bf)det(sI - A - lc)}$$

Symmetric Root Locus Design

Linear Quadratic Regulators for SISO Systems
Ein optimaler Controller minimiert die Cost Function
$V = \int_0^\infty (y^2(t) + \rho u^2(t))\,dt$, $\rho > 0$
Der erste Term repräsentiert den Control Error, der zweite den Control Effort, ρ ist ein Gewichtungsfaktor um den Schwerpunkt festzulegen, ein großes ρ verringert den Anteil von Control Effort.
Das Optimal Control Law hat die Form von State Feedback
$u_{opt}(t) = fx(t)$
und minimiert die Cost Function und plaziert die Eigenwerte von $A + bf$ auf den stabilen Nullstellen von

$$p(s) = a(-s)a(s) + \frac{1}{\rho}b(-s)b(s)$$

Die Nullstellen des Polynoms $p(s)$ liegen in der komplexen Ebene symmetrisch zur Im-Achse. Links die stabilen, rechts die instabilen, die stabilen Nullstellen bilden die gewünschten Eigenwerte des State-Feedback Vektors f.
Optimal Controller Design
Teilen von $p(s) = 0$ durch $a(-s)a(s)$ führt zu

$$1 + \frac{1}{\rho}G(-s)G(s) = 0$$

Die optimalen Eigenwerte von s liegen in der rechten Halbebene. Die Verwendung von standart Root-Locus Verfahren können mit folgenden Ersetzungen verwendet werden:
$L(s) \rightarrow G(-s)G(s)$, $K \rightarrow \frac{1}{\rho}$
$\Rightarrow\ 1 + KL(s) = 0$
Vorgehensweise zusammengefasst:
-Root Locus Plot von $1 + \frac{1}{\rho}G(-s)G(s) = 0$ zeichnen
-Optimale Closed-Loop Eigenvalues auswählen (Gewichtung zwischen Performance und Control Effort festlegen)
-Optimal State Feedback Gain berechnen durch Lösen eines Pole Placement Problems
Der entstandene State Feedback Controller heißt Linear Quadratic Regulator LQR.
Optimal State Estimation
Plant Model um Rauschen erweiter:
$\dot{x} = Ax + bu + b_n n_n$
$y = cx + n_y$
Observer Equation ist somit:
$\dot{\hat{x}} = A\hat{x} + bu + lc(\hat{x} - x) - ln_y$

Vom Plant Modell subtrahiert ergibt den Estimation Error:
$\dot{\tilde{x}} = (A + lc)\tilde{x} + b_n n_x + ln_y$
Mit der Transferfunktion von Process Noise zu Measured Output $G_n(s) = c(sI - A)^{-1}b_n$ und Rauschverhältnis $q = \frac{S_x}{S_y}$ ergibt sich, dass der Observer Gain l, welcher $\lim_{t\to\infty} E[\tilde{x}^T(t)\tilde{x}(t)]$ (Measure of Estimation Error Size) minimiert, die Eigenwerte von $A + lc$ auf die stabilen Werte in der linken Halbebene von $1 + qG_n(-s)G_n(s) = 0$ abbildet.
So ein Observer heisst Kalman Filter. Ein LQR Controller mit einem Kalman Filter kombiniert heißt Linear Quadratic Gaussian (LQG) Controller.

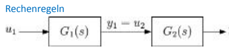
5. Multivariable Systems

Transfer Function Model

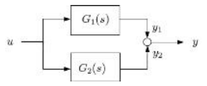
Von jedem Eingang zu jedem Ausgang gibt es eine Übertragungsfunktion, besteht keine Verbindung von Eingang j zu Ausgang i, so ist die entsprechende Übertragungsfunktion $G_{ij} = 0$. Ein Modell mit zwei Ein- und Ausgängen sieht wie folgt aus:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}$$

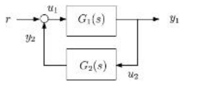
$Y(s) = G(s) * U(s)$
Rechenregeln



$G(s) = G_1(s)G_2(s)$



$G(s) = G_1(s) + G_2(s)$



$$G(s) = \frac{G_1(s)}{1 - G_1(s)G_2(s)}$$

State Space Model

MIMO State Space Model:
$\dot{x}(t) = Ax(t) + Bu(t)$
$y(t) = Cx(t) + Du(t)$

State Space Model -> Transfer Function

$G(s) = C(sI - A)^{-1}B + D$

Transfer Function -> State Space Model

Zu jeder einzelnen Übertragungsfunktion $G_{ij}(s)$ der Übertragungsfunktionsmatrix $G(s)$ die Controller Canonical Form bestimmen und zu einem System zusammenfassen:

$$\dot{x} = \begin{bmatrix} A_{11} & 0 & 0 & 0 \\ 0 & A_{12} & 0 & 0 \\ 0 & 0 & A_{21} & 0 \\ 0 & 0 & 0 & A_{22} \end{bmatrix} x + \begin{bmatrix} b_{11} & 0 \\ 0 & b_{12} \\ b_{21} & 0 \\ 0 & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & 0 & 0 \\ 0 & 0 & c_{21} & c_{22} \end{bmatrix} x$$

Diese Realisierung ist nicht zwangsläufig minimal, es können also uncontrollable und unobservable State Variablen auftreten.

The Gilbert Realization

Nur möglich, wenn die Eigenwerte jeder skalaren Übertragungsfunktion G_{ij} einfach sind. Ist die Übertragungsfunktionsmatrix $G(s)$ gegeben wird sie wie folgt zerlegt:
$G(s) = \frac{1}{d(s)}N(s)$ Bsp: $G(s) = \frac{1}{(s-\lambda_1)(s-\lambda_2)}N(s)$
$d(s)$ ist das kleinste gemeinsame Vielfache der Nenner aus $G(s)$ ($\deg d(s) = r$) und enthält die Polstellen, $N(s)$ wird bruchfrei. Dann wird Partialbruchzerlegung durchgeführt:

$$G(s) = \frac{1}{(s - \lambda_1)(\lambda_1 - \lambda_2)}N_1(\lambda_1) + \frac{1}{(\lambda_2 - \lambda_1)(s - \lambda_2)}N_2(\lambda_2) + \cdots$$

Mit der Definition $\rho_i = rankN_i$ lässt sich jede Matrix N_i faktorisieren als:
$N_i = C_i B_i$ mit $C_i \in \mathbb{R}^{l \times \rho_i}$, $B_i \in \mathbb{R}^{\rho_i \times m}$
C_i und B_i werden durch Ausprobieren gefunden.
Die Partialbruchzerlegung lässt sich umschreiben zu:

$$\frac{N_i}{s - \lambda_i} = C_i\frac{1}{s - \lambda_i}B_i = C_i\left(sI_{\rho i} - \lambda_i I_{\rho i}\right)^{-1}B_i$$

Somit erzeugt jeder Term eine State Space Realisierung mit $A_i = \lambda_i I_{\rho i}$. Alle Systeme mittels paralleler Verschaltung zusammengefasst ergibt das (minimale) MIMO State Space Model:

$$\dot{x} = \begin{bmatrix} \lambda_1 I_{\rho 1} & & 0 \\ & \ddots & \\ 0 & & \lambda_r I_{\rho r} \end{bmatrix} x + \begin{bmatrix} B_1 \\ \vdots \\ B_r \end{bmatrix} u$$

$y = [C_1 \quad \cdots \quad C_r]x$

MIMO Controllability and Observability

Controllability
Folgende Aussagen sind äquivalent:
- The System is controllable
- The controllability Gramian

$W_c(t) = \int_0^t e^{A\tau}BB^Te^{A^T\tau}d\tau$ is positive definite for any $t > 0$
- The controllability matrix $C(A, B) = [B \quad AB \quad \cdots \quad A^{n-1}B]$ has full row rank
- The matrix $[sI - A \quad B]$ has full row rank for all $s \in \mathbb{C}$

Observability
Folgende Aussagen sind äquivalent:
- The System is observable
- The observability Gramian

$$W_o(t) = \int_0^t e^{A^T\tau}C^TCe^{A\tau}d\tau$$

Is positive definit for any $t > 0$
- The observability matrix

$$O(C, A) = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

Has full column rank
- The matrix $\begin{bmatrix} sI - A \\ C \end{bmatrix}$ has full column rank for all $s \in \mathbb{C}$

Minimales System
Ein Realisierung System ist minimal wenn es die geringste Ordnung hat, also die geringstmögliche Anzahl an State Variablen. Eine Realisierung ist nur minimal, wenn (A, B) controllable und (C, A) obersvable ist. Eine Gilbert Realisierung ist immer minimal, weil sie immer controllable und observable ist.

Multivariable Poles and Zeros

- Der Grad der minimalen Realisierung bestimmt die Anzahl der Pole des Systems
- Bei einer minimalen Realisierung sind die Eigenwerte von A die Pole von $G(s)$
- Wenn Pole-Zero-Cancellation auftritt ist das System entweder unobservable oder uncontrollable
- Verliert $G(s)$ an der Stelle s_0 Rang, dann ist s_0 eine Nullstelle
- $G(s)$ hat eine invariante Nullstelle bei z_i wenn gilt: $det\begin{bmatrix} z_i I - A & -B \\ C & D \end{bmatrix} = 0$

$- G(s)$ hat eine transmission Zero bei z_i wenn gilt: $G(z_i)u_0 = 0$ mit $u_0 \neq 0$
(Wenn $G(s)$ eine Matrix ist, dann $\det(G(z_i))$ bilden)

- Auch hier sind bei minimalen Systemen Invariant und Transmission Zeros gleich. Ist das System nicht minimal, kommen zusätzliche Invariant Zeros dazu, die jedoch keinen Einfluss auf das Input-Output-Verhalten des Systems haben.

Augmented System

$A = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \bar{x} = \begin{bmatrix} x \\ x_I \end{bmatrix}$

State Feedback (positiv)

$u = \bar{F}\bar{x} = [F \quad F_I]\begin{bmatrix} x \\ x_I \end{bmatrix}$

Optimal Control of MIMO Systems

System mit (A, B) stabilizable und (C, A) detectable und $x_0 \neq 0$. Gesucht wird ein Eingangssignal, welches das System zurück zu $x_0 = 0$ bringt.

$V = \int_0^\infty \left(y^T(t)y(t) + u^T(t)Ru(t)\right) dt\,,\ R > 0$ MIMO Cost Function

$V = \int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) dt\,,\ R > 0, Q \geq 0$ more general Form

$Q = C^T C, \qquad R = \begin{bmatrix} \rho_1 & & 0 \\ & \ddots & \\ 0 & & \rho_m \end{bmatrix}$ ρ_i: tuning parameter

$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \quad H = \begin{bmatrix} h_i \\ g_i \end{bmatrix} = \lambda \begin{bmatrix} h_i \\ g_i \end{bmatrix} \qquad i = 1, \ldots, 2n$

Das optimal State Feedback $u_{opt}(t) = Fx(t)$ welches die Cost Function V minimiert, positioniert die Eigenwerte von $(A + BF)$ auf den stabilen Eigenvalues von H. Die Eigenvektoren von $(A + BF)$ sind die partitionierten h_i der Eigenvektoren von H, welche den stabilen Eigenwerten zugeordnet sind.

Optimal State Estimation

Das State Space Model wird um Rauschen erweitert

$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$

$y(t) = Cx(t) + v(t)$

Observer Gain L wird auf die gleiche Weise bestimmt wie Optimal Pole Placement mit folgenden Änderungen:

$A \to A^T, \qquad B \to C^T, \qquad Q \to Q_e, \qquad R \to R_e$

$Q_e = BB^T, \qquad R_e = diag(r_1, \ldots, r_l) \qquad r_i$: speed of estim. tuning parameter

6. Digital Control

Discrete-Time System: arbeitet nur mit diskreten Signalen

Sampled-Data System: arbeitet mit diskreten und kontinuierlichen Signalen

Discrete-Time Systems – z-Transform and State Space Model

Discrete-Time Signals

$x(k) = 0\,,\ k < 0 \qquad k \in \mathbb{Z} \Rightarrow k: -\infty \to +\infty$

Aanlog zu Continous-Time Systemen ist $x(k) = 0$ für $k < 0$.

Discrete-Time Unit Impuls (Dirac Impuls):

$\delta(k) = \begin{cases} 1, & k = 0 \\ 0, & sonst \end{cases}$

Discrete-Time Unit Step:

$\sigma(k) = \begin{cases} 1, & k \geq 0 \\ 0, & k < 0 \end{cases}$

Discrete-Time Exponential Function:

$x(k) = \begin{cases} e^{-ak}, & k \geq 0 \\ 0, & k < 0 \end{cases}$

$x(k) = e^{-ak}\sigma(k)$

Linear Difference Equation:

$y(k) + a_1 y(k-1) + \cdots + a_n a(k-n) = b_0 u(k-1) + \cdots + b_n u(k-n)$

Wenn $b_0 \neq 0$ ist, hängt der Ausgang zu jedem Zeitpunkt direct vom Eingang ab, das System antwortet unverzögert. Physikalische Systeme können das nicht, deshalbt ist normalerweise $b_0 = 0$, Nur wenn ein System sehr schnell antwortet (in Bezug auf die Sampling-Rate), ist $b_0 \neq 0$ zur Approximation legitim.

The z-Transform

Die Laplace-Transformation führt leicht abgewandelt zur z-Transformation, dabei wird stets vorrausgesetzt dass $x(0) = 0$.

$\mathcal{L}[\dot{x}(t)] = sX(s) \Rightarrow Z[x(k-1)] = z^{-1}X(z)$

Die z-Transformation termweise auf die Differenzen-Gleichung angewendet liefert:

$Y(z) + a_1 z^{-1} Y(z) + \cdots + a_n z^{-n} Y(z) =$
$b_1 z^{-1} U(z) + \cdots + b_n z^{-n} U(z)$

Dies führt zur Discrete-Time Transfer Function

$Y(z) = G(z)U(z)$

$G(z) = \dfrac{b_1 z^{-1} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}} = \dfrac{b_1 z^{n-1} + \cdots + b_n}{z^n + a_1 z^{n-1} + \cdots + a_n}$

Unit Impulse

$Z[\delta(k)] = 1$

Unit Step

$Z[\sigma(k)] = \dfrac{1}{1 - z^{-1}} = \dfrac{z}{z - 1}$

Exponential Function

$Z[x(k)] = \dfrac{1}{1 - e^{-n}z^{-n}} = \dfrac{z}{z - e^{-n}}$

Final Value Theorem and Steady State Gain

Der Final-Value eines Discrete-Time Systems kann berechnet werden mit:

$\lim_{k\to\infty} x(k) = \lim_{z\to 1}(z-1)X(z)$

Damit lässt sich der Static Gain eines Systems $G(z)$ berechnen. Die Sprungantwort ist:

$Y(z) = G(z)\dfrac{z}{z-1}$

Und führt mit der vorherigen Formel zum Static Gain $G(1)$:

$y(\infty) = \lim_{z\to 1}(z-1)G(z)\dfrac{z}{z-1} = G(1)$

Initial Value Theorem

$x(0) = \lim_{z\to\infty} X(z)$

Impulse Response

$Y(z) = G(z)U(z) = G(z) = \sum_{k=0}^{\infty} g(k)z^{-k}$

$g(k)$ heißen Markov-Parameter & stellen die Impulsantwort eines Systems dar mit:

$g(k) = \begin{cases} 0, & k < 0 \\ D, & k = 0 \\ C\Phi^{k-1}\Gamma, & k > 0 \end{cases}$

Discrete-Time State Space Model

Discrete-Time MIMO System:

$x(k+1) = \Phi x(k) + \Gamma u(k)$

$y(k) = Cx(k) + Du(k)$

Pulse Transfer Function

$G(z) = C(zI - \Phi)^{-1}\Gamma + D$

$D = 0$: strictly proper; $D \neq 0$: bi-proper

Steady State Gain, Static Gain

$z = 1 \to G(1)$

Solution of State-Equation

$x(k) = \Phi^k x(0) + \sum_{t=0}^{k-1} \Phi^{k-l-1}\Gamma u(l)$

$x(k) = free + forced\ response$

Stability of Discrete-Time Systems

Ein Discrete-Time System ist stabil wenn alle Eigenwerte von Φ innerhalb der Unit Disc sind.

Controllability

$C_d = [\Gamma \quad \Phi\Gamma \quad \cdots \quad \Phi^{n-1}\Gamma]$

Es existiert eine Sequenz $\{u(0), u(1), \cdots, u(n-1)\}$ die das System in n Schritten zum Final State bringt, wenn C_d Rank n hat.

Observability

$O_d = \begin{bmatrix} C \\ C\Phi \\ \cdots \\ C\Phi^{n-1} \end{bmatrix}$ Das System ist observable wenn O_d Rang n hat.

Stabilizability, Detectibility, Kalman-Decomposition, Minimal Realization

können genau wie für Continuous-Time Systems berechnet werden.

Sampled Data System

Ein Continuous-Time Signal wird durch einen Zug von Delta-Impulsen zu den Zeiten kT repräsentiert. Vergleicht man die Laplace-Transformierte des Delta-Zuges mit der z-Transformierten des Discrete-Time Signals, so erhält man die Beziehung

$z = e^{Ts} = e^{T(\sigma - jw)} = e^{T\sigma}(cos(wT) + jsin(wT))$

Die linke Halbebene wird in den Kreis $e^{T\sigma}$ projiziert, die rechte außerhalb.

Time Constant

Ein Pol, oder ein Polpaar mit $Re\ s < -\sigma$ korrespondiert mit einem exponentiellen Abfall mit der Zeitkonstanten $\tau < \frac{1}{\sigma}$

Berechnung von Sprungantwort $y(k)$

$Y(z) = G(z)\dfrac{z}{z-1} = \dfrac{a(z)}{b(z)}$

Polynomdivision durchführen: $a(z): b(z) = c + dz^{-1} + \cdots$

Dann gilt: $y(0) = c\,, \quad y(1) = d\,, \quad y(2) = \cdots$

Pole Location and Stability

Wie bei den Diskrete-Time Systemen müssen auch hier die Pole innerhalb der Unit-Disc liegen, damit das System stabil ist.

Die Pole des second order Continuous-Time Systems

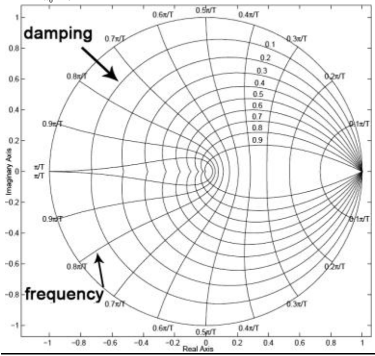
$\dfrac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

werden zu den folgenden Roots gemappt

$z^2 = a_1 z + a_0$

Mit $a_1 = -2e^{-\zeta\omega_n T}cos\left(\sqrt{1-\zeta^2}\,\omega_n T\right)$

$a_0 = e^{-2\zeta\omega_n T}$



Sampled Data Controller Design

Zwei Möglichkeiten:

1. Einen Continuous-Time Controller mit Standart Methoden entwerfen und danach den Controller $C(s)$ durch eine Discrete-Time Transfer-Function $D(z)$ approximieren
2. Zuerst den Continuous-Time Plant diskretisieren und dann einen Discrete-Time Controller $D(z)$ entwerfen.

Continuous-Time Design – Tustin Approximation

Folgende Ersetzung der s in der Continuous-Time Funktion führt auf die Discrete-Time Funktion

$C(s) \Rightarrow s = \dfrac{2}{T}\cdot\dfrac{1 - z^{-1}}{1 + z^{-1}} = \dfrac{2}{T}\cdot\dfrac{z-1}{z+1} \Rightarrow D(z) \qquad$ Rücktrafo: $z = \dfrac{2 + Ts}{2 - Ts}$

Die Sampling-Frequenz $\omega_s = {}^{2\pi}/_{T}$ sollte höher als die Bandbreite ω_b sein, ein vernünftiger Wert ist $\omega_s = 20 \sim 30\omega_b$.

Exact Discretization (Zero Order Hold)

Erzeugen eines Discrete-Time Systems durch

$A \Rightarrow \Phi = e^{AT}; \quad B \Rightarrow \Gamma = \int_0^T e^{At}\,Bdt$

Während der Sampling Augenblicke beschreibt das diskrete Modell das Continuous-Time Modell exakt, im Ggensatz zur Tustin-Approximation. Liegt ein Dicrete-Time Modell vor, so kann mit den Discrete-Time Versionen der Desgin Techniken, wie Root Locus, Pole Placement, Optimal Feedback und State Feedback, ein Controller entworfen werden.

Bei ZOH konvergieren die Pole zu 1 bei kleinem Sampling Time: $\lim_{T_s\to 0} e^{p_i T_s} = 1$

Deadbeat Control

Deadbeat Control ist rein theoretisch und bringt jedes System in n Schritten zurück zum Ursprung (0). Closed Loop Matrix von einem System in Controller Canonical Form:

$\Phi + \Gamma F = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \qquad (\Phi + \Gamma F)^n = 0 \to x(n) = (\Phi + \Gamma F)^n x(0) = 0$

Poles of Sampled-Data Systems

Pole des kontinuierlichen Systems: $s = p_i$

Pole des diskreten Systems: $z = e^{p_i T}$

Zeros of Sampled-Data Systems

Der Diskretisierungsprozess fügt neue Nullstellen hinzu (Sampling Zeros). Bei Tustin hat das neue System $n - m$ Zeros mehr, bei der Exact Discretization hat es $n - m$ Zeros mehr wenn $D \neq 0$ ist, ansonsten $n - m - 1$.

$G(s) = \dfrac{m\ x\ NS}{n\ x\ PS} \xRightarrow{Tustin} \dfrac{n\ x\ NS}{n\ x\ PS} = H(z)$

$D \neq 0$: $G(s) = \dfrac{m\ x\ NS}{n\ x\ PS} \xRightarrow{Exact\ Disc.} \dfrac{n\ x\ NS}{n\ x\ PS} = H(z)$

$D = 0$: $G(s) = \dfrac{m\ x\ NS}{n\ x\ PS} \xRightarrow{Exact\ Disc.} \dfrac{(n-1)\ x\ NS}{n\ x\ PS} = H(z)$

Conrete-Time SSM → DiscreteTime SSM (ZeroOrderHold / Exakt Discretization)

Soll eine Übertragungsfunktion $G(s)$ in eine diskrete Übertragungsfunktion $H(z)$ umgewandelt werden, so ist der Umweg über das Discrete-Time SSM notwendig:

$A \Rightarrow \Phi = e^{AT} = I + AT + \dfrac{A^2T^2}{2!} + \dfrac{A^3T^3}{3!} + \cdots$

$\Psi = A^{-1}(e^{AT} - I)T^{-1} = I + \dfrac{AT}{2!} + \dfrac{A^2T^2}{3!} + \cdots$

$B \Rightarrow \Gamma = \int_0^T e^{At}Bdt = \Psi TB$

$x(kT + T) = \Phi x(kT) + \Gamma u(kT)$ DiscreteTime SSM

$y(k) = Cx(k) + Du(k)$

$H(z) = G(z) = C(zI - \Phi)^{-1}\Gamma + D$ Transfer Funktion

Time Delay Systems

$G'(s) = G(s)e^{-sTd} \Rightarrow H'(z) = H(z)z^{-d} \qquad d = \dfrac{T_a}{T}$

Choice of Sampling Frequency

Daumenregel: Samplingfrequenz mindestens 5-10 mal so groß wie Signalbandbreite.

7. System Identification

Least Square Estimation

Ein unbekannter Prozess ist zu bestimmen, von welchem die Ausgänge $y(k)$ und lineare Faktoren $m_1(k), m_2(k), \cdots, m_n(k)$ bekannt sind. Ein lineares Prozessmodell lautet:

$y(k) = m_1(k)p_1 + m_2(k)p_2 + \cdots + m_n(k)p_n + e(k)$

Die $p(k)$ beschreiben die Abhängigkeit von den gemessenen Variablen und $e(k)$ enthält mögliche Fehler. Das Model lässt sich umschreiben in:

$y(k) = [m_1(k) \quad \cdots \quad m_n(k)]\begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} + e(k) = m^T(k)p + e(k)$

Wenn $y(l)$ und $m(l), l = 0, 1, \cdots, k$ gegeben sind, dann lautet das Least Square Estimation Problem: Finde einen Parametervektor p, so dass die Summe quadratischer Fehler minimal wird: $V(p) = \sum_{l=0}^k e^2(l)$

Für die Möglichkeit einer Berechnung, muss M vollen Spaltenrang besitzen.

Bsp:

Ein unbekanntes System erster Ordnung mit Transferfunktion

$G(z) = \dfrac{bz^{-1}}{1 + az^{-1}}$

soll identifiziert werden. Dazu wird das unbekannte und das bestimmte System mit dem selben $u(k)$ angeregt und die Differenz der Ausgänge ergibt

$e(k) = y(k) - \hat{y}(k)$

Umschreiben der Differenzengleichung führt zum linearen Regressionsmodel

$y(k) = -ay(k-1) + bu(k-1) + e(k) = [-y(k-1) \quad u(k-1)]\begin{bmatrix} a \\ b \end{bmatrix} + e(k)$

$\quad = m^T(k)p + e(k) = \hat{y}(k) + e(k)$

Solving the Least Square Estimation Problem

Mit den verfügbaren $y(l), u(l), l = 0, 1, 2, \cdots$ wird das Problem in die folgende Form gebracht: $Y = Mp + E$

$Y = \begin{bmatrix} y(0) \\ \vdots \\ y(k) \end{bmatrix}, \qquad M = \begin{bmatrix} m^T(0) \\ \vdots \\ m^T(k) \end{bmatrix}, \qquad E = \begin{bmatrix} e(0) \\ \vdots \\ e(k) \end{bmatrix}$

Die Summe der quadratischen Fehler $V(p)$ ist minimal, wenn der Parametervektor folgende Gleichung erfüllt: $M^T Mp = M^T Y$

Geometrische Interpretation

Funktion von oben so umschreiben:

$E = Y - \varphi_1 p_1 - \cdots - \varphi_n p_n \qquad$ mit $\varphi_i = [m_i(0) \cdots m_i(k)]$

Man sucht hier eine lineare Kombination von $\varphi_1 \cdots \varphi_n$, die am nächsten an Vektor Y heran kommt. Dann stehen alle φ_i orthogonal zu E also $\varphi_i^T E = 0$

Über $M^T E = 0$ gelangt man zu: $M^T(Y - Mp) = 0$

Estimation of Transfer Function Models

Zur Vereinfachung enthält folgendes System keine autoregressiven Komponenten a_i

$\hat{y}(k) = b_1 u(k-1) + b_2 u(k-2) + \cdots + b_n u(k-n)$

$\hat{y}(k) = [u(k-1) \quad \cdots \quad u(k-n)]\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = m^T(k)p$

Die Measurement Matrix hat die Form

$M = \begin{bmatrix} u(n-1) & u(n-2) & \cdots & u(0) \\ u(n) & u(n-1) & \cdots & u(1) \\ \vdots & \vdots & \ddots & \vdots \\ u(k) & u(k-1) & \cdots & u(k-n+1) \end{bmatrix}$

↳ ↳ ↳

Empirical Autocorrelation der Datensequenz $u(k)$

$c_{uu}(l) = \lim_{k\to\infty} \dfrac{1}{k}\sum_{i=1}^{k} u(i)u(i-l)$

führt zu

$C_{uu}(n) = \begin{bmatrix} c_{uu}(0) & c_{uu}(1) & \cdots & c_{uu}(n-1) \\ c_{uu}(1) & c_{uu}(0) & \cdots & c_{uu}(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ c_{uu}(n-1) & c_{uu}(n-2) & \cdots & c_{uu}(0) \end{bmatrix}$

und dem Zusammenhang

$\lim_{k\to\infty} \dfrac{1}{k} M^T M = C_{uu}(n)$

Für ausreichend lange Datensequenzen kann die Matrix $M^T M$ als eine skalierte Version der empirischen Kovarianzmatrix $C_{uu}(n)$ interpretiert werden.

Persistent Excitation

Die Excitation Condition besagt dass die Matrix $M^T M$ vollen Spaltenrang haben muss, d.h. das Eingangssignal muss so reichhaltig sein, dass alle dynamischen Mode des Systems angeregt werden. Ein Signal $u(k)$ wird persistently exciting of order n genannt, wenn die Matrix $C_{uu}(n)$ positiv definit ist.

Eine weitere, zwingend notwendige Bedingung ist:

$\lim_{k\to\infty} \frac{1}{k}\sum_{l=0}^{k}\left(a(z)u(l)\right)^2 > 0 \qquad \forall a(z): \deg a(z) \leq n$

$a(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_{n-1}z^{n-1}$

$a(z)u(l) = a_0 u(l) + a_1 u(l+1) + \cdots + a_{n-1}u(l+n-1)$

$\lim_{k\to\infty} \dfrac{1}{k}\sum_{l=0}^{k}\left(a(z)u(l)\right)^2 = a^T C_{uu}(n)a$

$a^T = [a_{n-1} \quad a_{n-2} \quad \cdots \quad a_0]$

Bsp

Bsp	Impulse:	PE of order 0
	Step:	PE of order 1
	Sinus:	PE of order 2
	White Noise:	PE of any order

ARX Models (AutoRegressive with eXogenous Input)

Modell ohne Vereinfachung, also mit autoregressiven Komponenten im Ausgang

$\hat{y}(k) = -a_1 y(k-1) - \cdots - a_n y(k-n) \quad + b_1 u(k-1) + \cdots + b_n u(k-n)$

$\quad = [-y(k-1) \quad \cdots \quad -y(k-n) \quad u(k-1) \quad \cdots \quad u(k-n)]\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$

$M = \begin{bmatrix} u(n-1) & u(n-2) & \cdots & u(0) & y(n-1) & y(n-2) & \cdots & y(0) \\ u(n) & u(n-1) & \cdots & u(1) & y(n) & y(n-1) & \cdots & y(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u(k) & u(k-1) & \cdots & u(k-n+1) & y(k) & y(k-1) & \cdots & y(k-n+1) \end{bmatrix}$

Mit der empirical CrossCovariance

$c_{uy}(l) = \lim_{k\to\infty} \dfrac{1}{k}\sum_{i=1}^{k} u(i)y(i-l)$

folgt: $\lim_{k\to\infty} \frac{1}{k} M^T M \begin{bmatrix} C_{yy} & -C_{uy} \\ -C_{uy} & C_{uu} \end{bmatrix}$

Subspace Identification of State Space Models

State Space Model

$x(k+1) = \Phi x(k) + \Gamma u(k)$

$y(k) = Cx(k) + Du(k)$

Wird mit Unit Impulse auf Eingang i erregt

$u_{\delta i}(k) = \begin{bmatrix} 0 \\ \vdots \\ \delta(k) \\ \vdots \\ 0 \end{bmatrix} \quad \to \quad y_{\delta i}(k) = \begin{bmatrix} g_{1i}(k) \\ \vdots \\ g_{li}(k) \end{bmatrix}$

Ein Eintrag $g_{ji}(k)$ in dem Output Vektor beschreibt den Ausgang am Channel j wenn dem Eingang i ein Unit Impulse zugeführt wurde. Die komplette Impulse Response Matrix beinhaltet die Impulse Responses von jedem Eingang zu jedem Ausgang:

$g(k) = \begin{bmatrix} g_{11}(k) & \cdots & g_{1m}(k) \\ \vdots & & \vdots \\ g_{l1}(k) & \vdots & g_{lm}(k) \end{bmatrix}$

Mit der Schreibweise $\Gamma = [\Gamma_1 \quad \Gamma_2 \quad \cdots \quad \Gamma_m], \quad D = [d_1 \quad d_2 \quad \cdots \quad d_m]$

Lassen sich alle Antworten auf Sprünge an allen Eingängen schreiben als:

$g(k) = [y_{\delta 1}(k) \quad \cdots \quad y_{\delta m}(k)] = \begin{cases} 0, & k < 0 \\ D, & k = 0 \\ C\Phi^{k-1}\Gamma, & k > 0 \end{cases}$

Constructing a Model from the Impulse Response

Measured Impulse Response Data in Matrixform (Hankel Matrix):

$H_k = \begin{bmatrix} g(1) & g(2) & g(3) & \cdots & g(k) \\ g(2) & g(3) & \cdots & & g(k+1) \\ g(3) & \vdots & & & \\ \vdots & & & & \vdots \\ g(k) & g(k+1) & & \cdots & g(2k-1) \end{bmatrix} \qquad H \in \mathbb{R}^{k\cdot Outputs \times k\cdot Inputs}$

Mit der Schreibweise von $g(k)$:

$H(k) = \begin{bmatrix} C\Gamma & C\Phi\Gamma & C\Phi^2\Gamma & \cdots & C\Phi^{k-1}\Gamma \\ C\Phi\Gamma & C\Phi^2\Gamma & \cdots & & C\Phi^k\Gamma \\ C\Phi^2\Gamma & \vdots & & & \\ \vdots & & & & \vdots \\ C\Phi^{k-1}\Gamma & C\Phi^k\Gamma & & \cdots & C\Phi^{2k-2}\Gamma \end{bmatrix}$

Faktorisierung:

$H_k = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{k-1} \end{bmatrix}[\Gamma \quad \Phi\Gamma \quad \cdots \quad \Phi^{k-1}\Gamma] = O_k C_k$

O_k und C_k sind die erweiterten Observability und Controllability Matrizen, weil die Anzahl an Samples k höher ist als die erwartete Order n des Systems. Da ein minimales System bestimmt werden soll gilt

rank O_k = rank $C_k = n \quad \Rightarrow \quad$ rank $H_k = n$

Allgemein gilt: rank $H_k = \min(\text{rank}C_k, \text{rank}O_k) = n$

Wenn man also die Impulsantwort eines Systems in die Matrixform H_k schreibt, lässt sich am Rang der Ordnung des Systems erkennen

The Ideal Case

$g(0) = D$ wenn $D \neq 0$

$H_k \in \mathbb{R}^{mk\,x\,lk}; \quad H_k = ML; \qquad M \in \mathbb{R}^{lk\,x\,n}; \qquad L \in \mathbb{R}^{n\,x\,mk}$

$rank(H_k) = n = Order\ of\ System$

$O_k = M; \ C_k = L$

Die ersten l Zeilen von M sind die Systemmatrix C; die ersten m Spalten von L sind die Systemmatrix Γ.

$\bar{O}_k = \begin{bmatrix} C\Phi \\ \vdots \\ C\Phi^k \end{bmatrix} = O_k\Phi$

$\Phi = (O_k^T O_k)^{-1} O_k^T \bar{O}_k$

Single Value Decomposition

Eigenwerte von σ die deutlich kleiner sind als andere werden zu Null gesetzt. Die ersten x Zeilen von O_k ergeben C und die ersten y Spalten von C_k ergeben Γ.

$H_k = Q\Sigma V^T = [q_1 \quad \cdots \quad q_{kl}]\begin{bmatrix} \sigma_1 & & 0 & \cdots & 0 \\ & \ddots & & & 0 \\ 0 & & \sigma_p & \cdots & 0 \end{bmatrix}\begin{bmatrix} v_1^T \\ \vdots \\ v_{km}^T \end{bmatrix}$

$O_k = Q_s\Sigma_s^{1/2} = \begin{bmatrix} u_{1.1} & \cdots & u_{1.m} \\ \vdots & & \vdots \\ u_{n.1} & \cdots & u_{n.m} \end{bmatrix}\begin{bmatrix} \sqrt{\sigma_1} & & \\ & \ddots & \\ & & \sqrt{\sigma_n} \end{bmatrix}$

$C_k = \Sigma_s^{1/2}V_s^T = \begin{bmatrix} \sqrt{\sigma_1} & & \\ & \ddots & \\ & & \sqrt{\sigma_n} \end{bmatrix}\begin{bmatrix} v_{1.1} & \cdots & v_{1.m} \\ \vdots & & \vdots \\ v_{n.1} & \cdots & v_{n.m} \end{bmatrix}$

Allgemeine Rechenregeln

$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \qquad A^{-1} = \frac{1}{\det(A)}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{\det(A)}\cdot \text{adj}(A)$

$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, \qquad A^{-1} = \frac{1}{\det(A)}\begin{bmatrix} ei - fh & ch - bi & bf - ce \\ fg - di & ai - cg & cd - af \\ dh - eg & bg - ah & ae - bd \end{bmatrix}$

Transformationen

$X(s)$	$x(t)$	$x(k)$	$X(z)$
		$\delta(k)$	1
$\frac{1}{s}$	$\sigma(t)$	$\sigma(k)$	$\frac{1}{1-z^{-1}}$
$\frac{1}{s+a}$	$e^{-at}\sigma(t)$	$e^{-akT}\sigma(k)$	$\frac{1}{1-e^{-aT}z^{-1}}$
$\frac{1}{s^2}$	$t\sigma(t)$	$kT\sigma(k)$	$\frac{Tz^{-1}}{(1-z^{-1})^2}$
$\frac{a}{s(s+a)}$	$(1-e^{-at})\sigma(t)$	$(1-e^{-akT})\sigma(k)$	$\frac{(1-e^{-aT})z^{-1}}{(1-z^{-1})(1-e^{-aT}z^{-1})}$
$\frac{b-a}{(s+a)(s+b)}$	$(e^{-at}-e^{-bt})\sigma(t)$	$(e^{-akT}-e^{-bkT})\sigma(k)$	$\frac{(e^{-aT}-e^{-bT})z^{-1}}{(1-e^{-aT}z^{-1})(1-e^{-bT}z^{-1})}$
$\frac{1}{(s+a)^2}$	$te^{-at}\sigma(t)$	$kTe^{-akT}\sigma(k)$	$\frac{Te^{-aT}z^{-1}}{(1-e^{-aT}z^{-1})^2}$
$\frac{s}{(s+a)^2}$	$(1-at)e^{-at}\sigma(t)$	$(1-akT)e^{-akT}\sigma(k)$	$\frac{1-(1+aT)e^{-aT}z^{-1}}{(1-e^{-aT}z^{-1})^2}$
$\frac{b}{s^2+b^2}$	$\sin bt\ \sigma(t)$	$\sin bkT\ \sigma(k)$	$\frac{z^{-1}\sin bT}{1-2z^{-1}\cos bT+z^{-2}}$
$\frac{s}{s^2+b^2}$	$\cos bt\ \sigma(t)$	$\cos bkT\ \sigma(k)$	$\frac{1-z^{-1}\cos bT}{1-2z^{-1}\cos bT+z^{-2}}$
		$a^k\sigma(k)$	$\frac{1}{1-az^{-1}}$

Matlab

System und Transferfunktion	
A = [1 2 ; 3 4];	Matrix $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ erzeugen
A = eye(n); A = eye(m,n);	m*n Matrix mit 1 Diagonale: 2,3 -> $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
d = size(A); [m,n] = size(A);	Rückgabe der Dimensionen der Matrix als Vektor d oder direkt in die Elemente m (Zeilen) und n (Spalten)
a=size(A,x);	Ausgabe an Zeilen (x=1) od. Spalten (x=2)
B = A(n:m,j:k);	Matrix B erzeugen mit den Zeilen n bis m und Spalten j bis k von A.
B = A(n:m,:);	Matrix B erzeugen mit den Zeilen n bis m und allen Spalten von A.
a = A(1,2);	Aus der Matrix A den Wert von A_{12} entnehmen
A_inv = inv(A);	Inverse A Matrix erzeugen
A_trans = A‘;	A transponieren
d = eig(A); [V,d] = eig(A);	Gibt einen Vektor d mit den Eigenwerten von A zurück. V ist eine Matrix mit den Eigenvektoren von A in den Spalten.
p = poly(A);	Charakteristisches Polynom von A
k = rank(A);	Rang von A
C = conv(A, B);	Ausmultiplizieren der Vektoren A & B wie Polynome
Sys = ss(A,b,c,d); Sys = ss(G); Sysd = ss(A,b,c,d,Ts);	StateSpace Model erzeugen aus einzelnen Matrizen oder einer Transferfunktion. Digitales System mit Samplingzeit Ts.
G = tf(Sys);	Transferfunktion aus StateSpace Model
[A,B,C,D] = tf2ss(b,a);	StateSpace Model aus Transferfunktion mit Numerator b und Denumerator a.
SysT = ss2ss(sys,T);	Similarity Transformation with T
[a,b,c,d] = ssdata(sys); [a,b,c,d,Ts] = ssdata(sys)	Matrizen A,B,C und D aus dem System extrahieren, für diskrete Systeme mit Samplingzeit Ts.
Co = ctrb(sys); Co = ctrb(A,B);	Controllability Matrix Co erzeugen
Ob = obsv(sys); Ob = obsv(A,C);	Observability Matrix Ob erzeugen
[Abar,Bbar,Cbar,T,k] = ctrbf(A,B,C);	Controllability Zerlegung erzeugen und ausgeben des neuen Systems, Transformationsmatrix und Zeilenvektor k, wessen Länge die Ordnung des Systems A ist.
[Abar,Bbar,Cbar,T,k] = obsvf(A,B,C);	Observability Zerlegung
sysr = minreal(sys);	Minimale Realisierung des Systems sys
[sysr,u] = minreal(sys,tol);	Minimales System mit orthogonaler Matrix U so dass (U*A*U',U*B,C*U') eine Kalman decomposition von (A,B,C) ist
sysd = c2d(sys,Ts,method);	Concrete Time (SSM oder TF) in Discrete Time Model umwandeln mit Abtastzeit Ts. Method entweder ‘tustin’ oder ‘zoh’.
sysc = d2c(sysd,method);	Discrete Time in Concrete Time wandeln. Method entweder ‘tustin’ oder ‘zoh’.
[K,S,e] = LQR(A,B,Q,R,N); [K,S,e] = LQR(A,B,Q,R);	Linear-quadratic (LQ) state-feedback regulator für ein StateSpace System. K: optimal Gain Matrix. S: Solution of Riccati Equation. E: Closed-Loop Eigenvalues.
F = -lqr(A,B,Q,R);	Optimal Direkt State Feedback Vektor F ausgeben
L = -lqr(A’,B’,Qe,Re);	Optimal Observer Gain L ausgeben
F = -place(A,B,p);	State Feedback Vektor F zu vorgegebenen Polen in Zeilenvektor [p1 p2 p3 …]
L = -place(A‘,B‘,p);	Observer Gain L zu vorgegebenen Polstellen
[U,S,V] = svd(X);	Single Value Decomposition. S ist diagonal und positiv, U & V sind unitär wobei gilt: USV=X
Übertragungsfunktionen	
G = tf(num,den);	Transferfunktion aus numerator und denumerator
G = tf(num,den,Ts);	Diskrete Transferfunktion mit Samplingzeit
k = dcgain(G);	Statische Verstärkung von G bestimmen
P = pole(sys);	Polstellen berechnen
z = zero(sys);	Nullstellen bestimmen
sys = series(sys1,sys2);	Zwei Systeme in Reihenschaltung zusammenführen
sys = feedback(sys1,sys2);	Zwei Systeme in negativer Feedbackschaltung zusammenführen
Plotfunktionen	
step(sys);	Sprungantwort vom System plotten
Impulse(sys,3);	Impulsantwort bis 3sec bzw. bis k=3 falls diskret
ltiview(sys);	Sprungantwort vom System im LTI Viewer
bode(sys);	Bodeplot vom System
plot(Y); plot(X,Y);	Zeichnet Vektor Y über X
nyquist(sys);	Nyquistplot vom System erstellen
rlocus(sys);	RootLocus Plot vom System
Funktionen	
for k=initval:endval, statements, end	For Schleife mit k als Zählvariable

Beispiele

- Kalman decompisition of StateSpace Model

```
G = ss(A,B,C,D);
[Gmin,T] = minreal(G);
Gkalman = ss2ss(G,T);
```

- System Identification:

Bestimmung eines unbekanntes StateSpace Models. Nur die Hankel Matrix H_k ist im Matlab Workspace:

```
n = rank(Hk);          % 1 rank(Hk) = system order
[m,r] = size(Hk);      % 2 dimensions
[U,S,V] = svd(Hk);     % 3 decompose
Us = U*sqrt(S);        % 4 split the singular values evenly
Vs = sqrt(S)*V';       % 5 sqrt zieht die Wurzel aus Elementen von S
Ck = Vs(1:n,1:r);      % 6 extended controlability
Ok = Us(1:m,1:n);      % 7 extended observability (Hk = Ok*Ck)
Gamma = Ck(:,1);       % 8 1 input -> 1st column of Ck
C = Ok(1:2,:);         % 9 2 % outputs -> 1st and 2nd rows of Ok
```

- MIMO Plant Observer-Based Control mit Integrator (Augmented Plant)

Für einen Plant $G(s)$ soll für einen LQR Controller F und F_I bestimmt werden. Im Workspace liegen die Plant Matrizen A, B, C, D. Die Gewichtungsmatrizen müssen gesetzt werden.

```
A_aug = [A zeros(size(A,1),size(D,2)); -C 0];
B_aug = [B; 0];
Q_aug = [C'*C zeros(size(A,1),size(D,2));
         zeros(size(D,1),size(A,2)) 1];
R = eye(2);       % for 2 inputs
F_aug = lqr(A_aug,B_aug,Q_aug,R);
F = -F_aug(:,1:size(A,1));
F_I = -F_aug(:,size(A,1)+1: size(A,1)+ size(B,2));
```

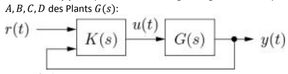
- Testen der Controllability

Für ein System von Grad n. A und B sind bereits im Workspace

```
rank(ctrb(A,B)) < n
```

- LTI System Controller erzeugen

Controller $K(s)$ entsprechend Abbildung erzeugen. Im Workspace befinden sich A, B, C, D des Plants $G(s)$:




```
r  = 0.1;
re = 0.1;
[A,B,C,D] = ssdata(G);
Q  = C’*C;                % output weighting
R  = r*eye(2);            % input weighting
Qe = B*B’;                % process noise
Re = re*eye(2);           % output noise
F  = lqr(A,B,Q,R);        % state feedback
L  = lqr(A’,C’,Qe,Re);    % observer
A_cl = [A-B*F B*F; zeros(3,3) A-L’*C];
B_cl = [B ; zeros(3,2)];
C_cl = [C   zeros(2,3)];
G_cl=ss(A_cl,B_cl,C_cl,0);
```

- System Identification

Berechnung von p wenn U und Y im Workspace sind.

```
n  = length(u);
M  = [u(2:n-1) u(1:n-2)];
Yn = Y(n+1:length(y));
p  = ((M’)*M)/M’*Yn);

function M = mkm(y,u,n)
ny = length(y);
nu = length(u);
ndata = min(nu,ny)-1;

for i=n:data
   M(i-n+1 , 1:n) = -y(I:-1:i-n-1)’;
   M(i-n+1 , n+1:2*n) = u(i:-1:i-n+1)’;
end
```