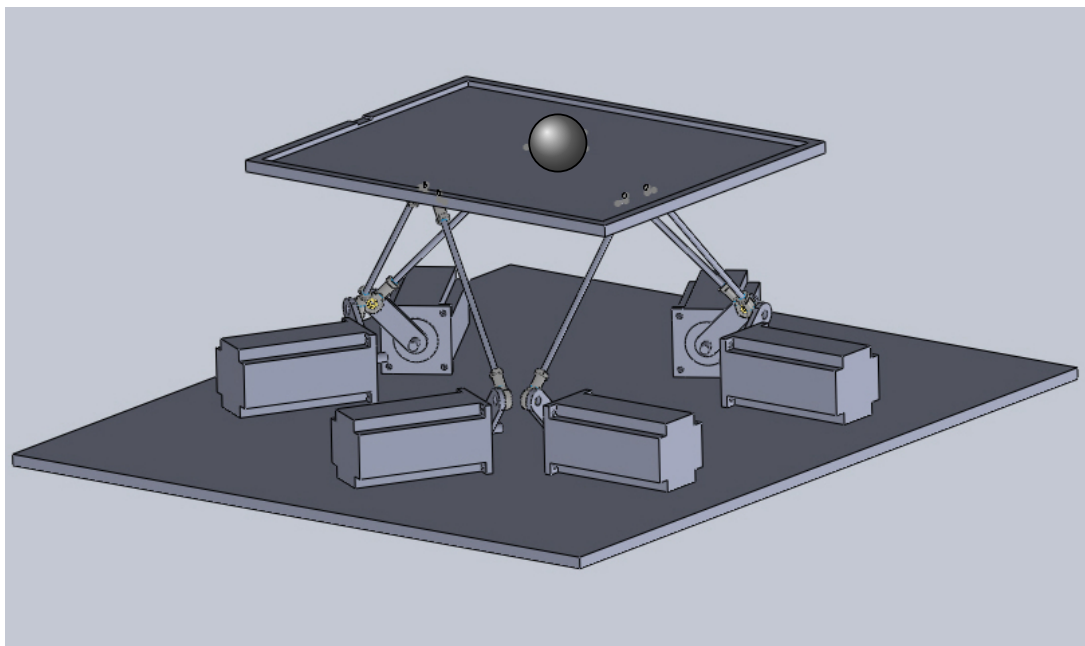


# Praktikum Automatisierungstechnik

- Sommersemester 2022-

## Versuch 3: Ball auf Platte

Version 1.9



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Versuchsaufbau</b>	<b>4</b>
<b>3</b>	<b>Modellierung des Systems</b>	<b>5</b>
3.1	Modellierung des Balls auf der Platte . . . . .	5
3.1.1	Koordinatensysteme . . . . .	6
3.1.2	Herleitung der Bewegungsgleichungen . . . . .	6
3.1.3	Darstellung des Systems im Zustandsraum . . . . .	8
3.1.4	Modellverifikation . . . . .	8
3.1.5	Linearisierung der Systemgleichungen . . . . .	9
3.1.6	Modellanalyse . . . . .	11
3.2	Modellierung der Motor/Plattendynamik . . . . .	11
<b>4</b>	<b>Reglerentwurf</b>	<b>13</b>
4.1	Stabilisierender Regler . . . . .	13
4.2	Erweiterung um eine flachheitsbasierte Vorsteuerung . . . . .	15
4.2.1	Grundlagen . . . . .	15
4.2.2	Flachheitsbasierte Vorsteuerung . . . . .	17
4.3	Trajektorienplanung . . . . .	18
4.3.1	Arbeitspunktwechsel . . . . .	18
4.3.2	Periodische Schwingung . . . . .	19
<b>5</b>	<b>Versuchsdurchführung</b>	<b>20</b>
5.1	Implementierung in den Simulationsumgebungen . . . . .	20
5.1.1	Implementierung der berechneten Modelle und des Kaskadenreglers . . . . .	20
5.1.2	Implementierung Flachheitsbasierte Vorsteuerung . . . . .	21
5.1.3	Implementierung der Trajektorienplanung für den Arbeitspunktwechsel (2D) . . . . .	21
5.1.4	Implementierung der Trajektorienplanung für die Kreisbahn (2D) . . . . .	22
5.1.5	Erweiterung der Regler auf die 2. Dimension . . . . .	22

# 1 Einleitung

Im Praktikumsversuch *Ball auf Platte* werden Methoden zur Regelung von nichtlinearen Systemen entworfen. Diese Methoden werden daraufhin in Matlab/Simulink implementiert und experimentell am Versuchsaufbau erprobt. Ziel der jeweiligen Regelung ist es den Ball zu stabilisieren und einer vorgegebenen Solltrajektorie folgen zu lassen.

Die zentralen Lehrinhalte des Praktikums beinhalten:

- Methoden der Modellierung dynamischer Systeme
- Lineare und nichtlineare Systemanalyse und Steuerungsentwurf
- Implementierung der Algorithmen in Matlab/Simulink
- Portierung der Algorithmen auf eine Industriesteuerung

## Hinweis:

Das Praktikumskript enthält zwei Arten von Aufgaben welche unterschiedlich gekennzeichnet sind.

### **Vorbereitungsaufgabe – V3.0 –**

Alle Vorbereitungsaufgaben sind selbstständig vor dem Praktikumstermin durchzuarbeiten. Falls nicht anders angegeben, wird eine schriftliche Lösung der Aufgaben verlangt. Diese sind auf gesonderten Blättern, welche mit Name, Studiengang und Matrikelnummer versehen sind, zu lösen. Die angefertigte Ausarbeitung kann gegebenenfalls zur Bewertung des Praktikums genutzt werden.

### **Praktikumsaufgabe – P3.0 –**

Die Praktikumsaufgaben führen Sie durch den experimentellen Teil des Praktikums und werden am Tag der Versuchsdurchführung bearbeitet.

## 2 Versuchsaufbau

Im Folgenden wird der für das Praktikum zur Verfügung stehende Versuchsaufbau beschrieben. Dieser besteht, wie in Abbildung 1

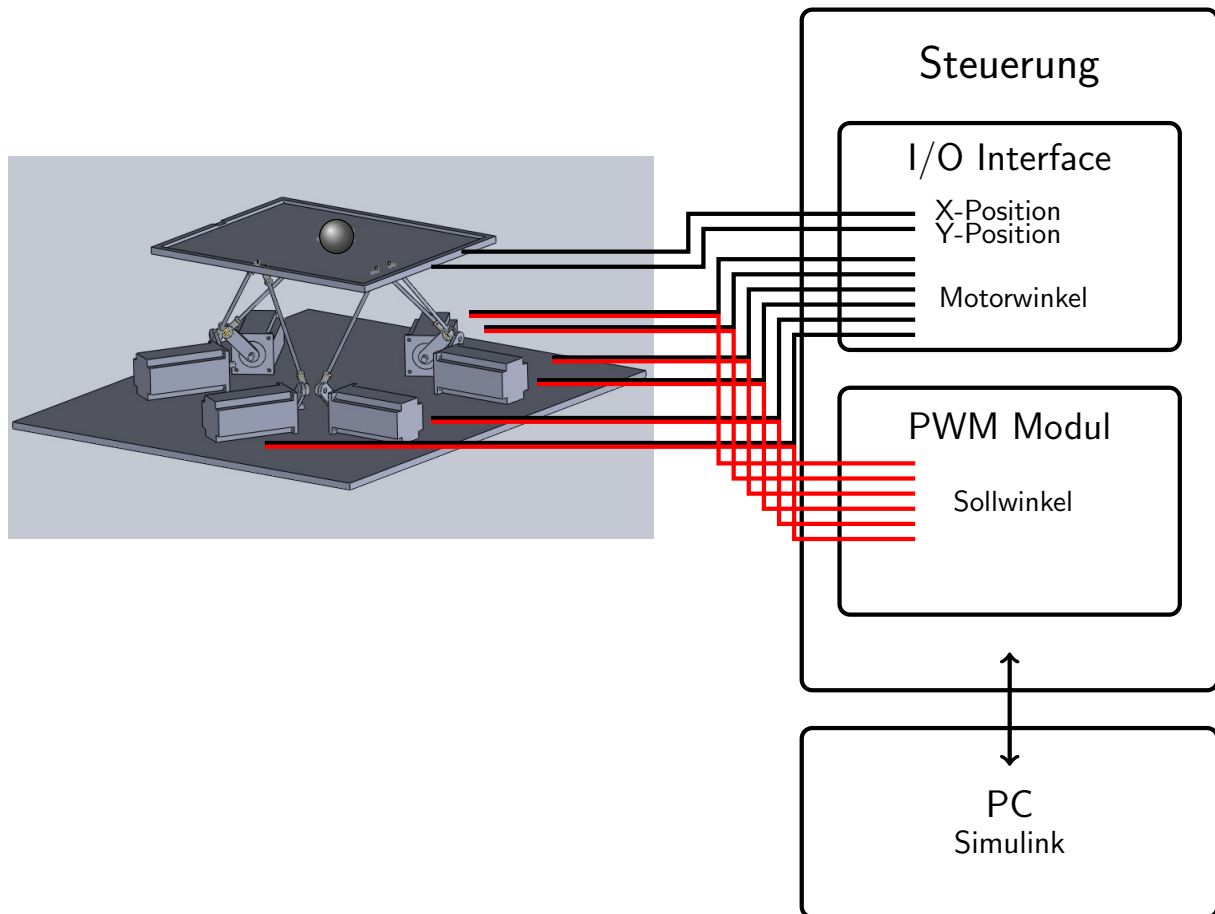


Abbildung 1: Komponenten des BaP

dargestellt, aus einem resistiven Touch-Display, Hexapoden, 6 Schrittmotoren, einer Industriesteuerung von Bachmann und einem handelsüblichen PC. Das Display kann mit dem Schrittmotoren und dem Hexapod in 6 Freiheitsgraden (3 Translationen und 3 Rotationen) bewegt werden. Die Stell-signale werden über die Steuerung mit einem PWM Signal angetrieben. Für das Nachfahren einer Bahn oder zur Stabilisierung der Kugel wird die Kugelposition mit dem Display detektiert und die zugehörigen Steuersignale mit einem geeigneten Regelgesetz für die Schrittmotoren generiert. Das Regelgesetz wird über einen PC mit Simulink realisiert und auf die Steuerung geladen. Die Moto-ren verfügen über interne Schrittzähler und erlauben so nach einer anfänglichen Referenzierung das Auslesen ihrer absoluten Position.

In diesem Versuch wird der Aufbau auf zwei Rotationen beschränkt, mit denen die Kugel bewegt wird.

### 3 Modellierung des Systems

#### 3.1 Modellierung des Balls auf der Platte

Im Folgenden soll ein dynamisches Modell des Balls auf der Platte hergeleitet werden. Hierzu ist in Abbildung 2 der Versuchsaufbau skizziert. Oben dargestellt ist die dreidimensionale Ansicht der Verkipfung der Platte um die  $y$ - bzw.  $x$ -Achse in Abhängigkeit der Winkelposition der Stellmotoren.

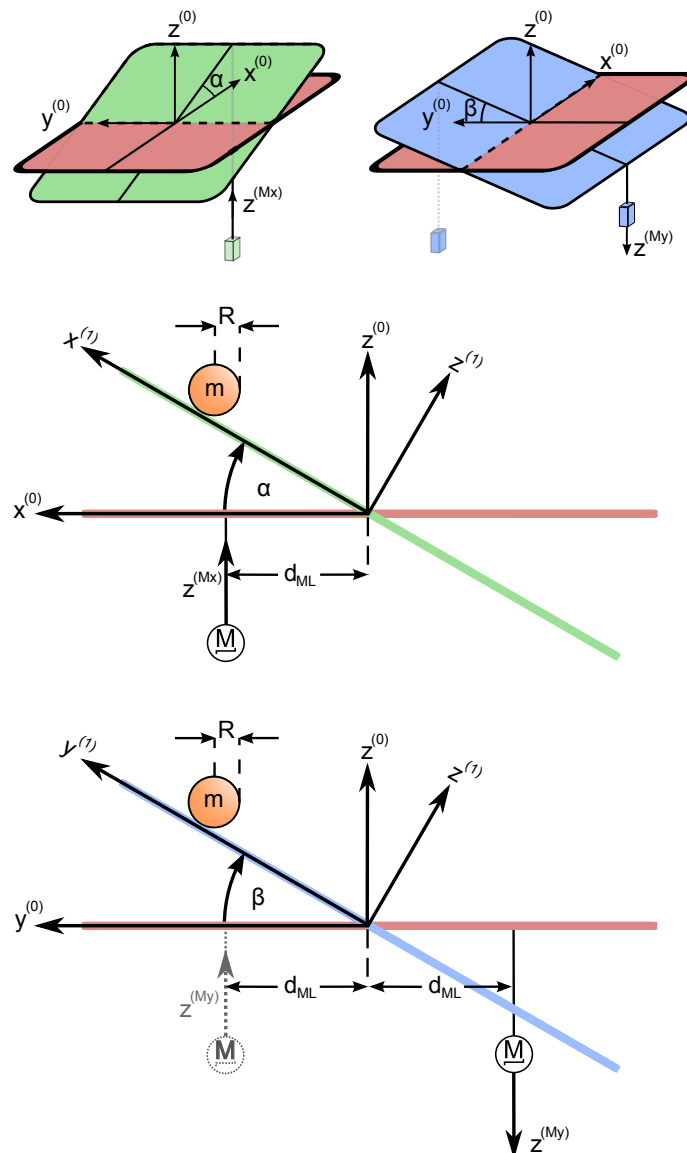


Abbildung 2: Skizze des Versuchsaufbaus. Oben: dreidimensionale Ansicht mit Verkipfung um die  $y$ - (links) bzw.  $x$ -Achse (rechts). Mitte/Unten: Projektion auf  $y$ - bzw.  $x$ -Ebene. Die Punktierung entspricht der virtuellen Spiegelung des Motors.

Die entkoppelte Betrachtungsweise des zweidimensionalen Ball-Platte Versuchs in nur je einer Dimension führt auf den eindimensionalen Ball-Balken Versuch. Das MIMO<sup>1</sup> System mit zwei Eingängen und zwei Ausgängen wird somit vereinfachend als zwei SISO<sup>2</sup> Systeme mit jeweils einem Eingang und Ausgang aufgefasst.

Der Ball-Balken Versuch entspricht einem rollenden Körper auf einer schiefen Ebene. Das Ersatzbild des freigeschnittenen mechanischen Systems ist in Abbildung 3 dargestellt. Aufgrund des analogen Vorgehens für die Bestimmung der Bewegungsgleichung der beiden Achsen, wird im Folgenden lediglich auf die Balldynamik entlang der x-Achse eingegangen.

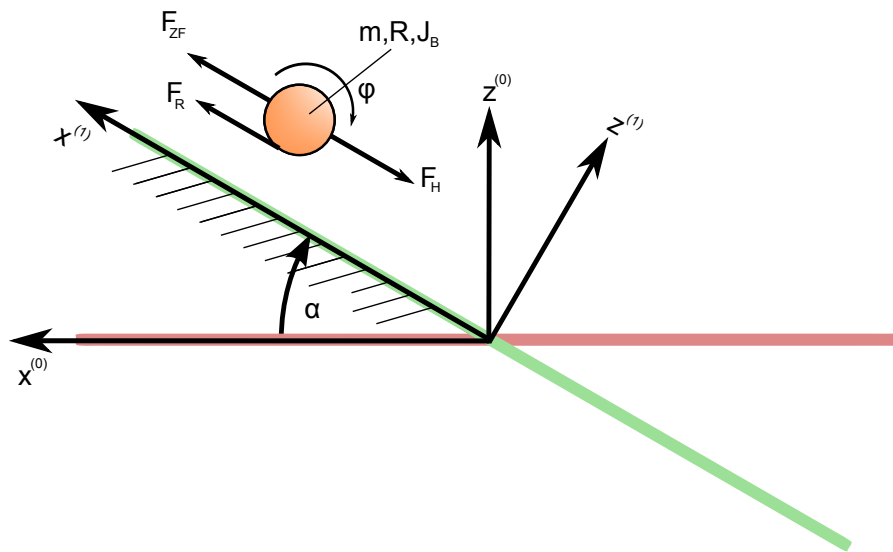


Abbildung 3: Ersatzbild des projizierten mechanischen Systems. Dabei handelt es sich um einen Ball auf einer schiefen Ebene. Am Ball greifen Rollreibungs-, Hangabtriebs- und Zentrifugalkraft an.

### 3.1.1 Koordinatensysteme

Das körperfeste Koordinatensystem  $e^{(1)}$  hat seinen Ursprung im Gelenkpunkt des Balkens und führt mit diesem eine Rotation bezüglich des raumfesten Koordinatensystems  $e^{(0)}$  aus. Die beiden Koordinatensysteme sind durch eine Drehung um die x- bzw. y-Achse mit den Winkeln  $\alpha$  und  $\beta$  miteinander verknüpft.

### 3.1.2 Herleitung der Bewegungsgleichungen

Bei dem Körper handelt es sich um eine Kugel der Masse  $m$  mit dem Radius  $R$  und dem Trägheitsmoment  $J_B$ . Zwischen Ball und Platte wirkt die Rollreibungskraft  $F_R$ , welche ein schlupffreies Rollen

<sup>1</sup>Multiple Input Multiple Output

<sup>2</sup>Single Input Single Output

der Kugel bewirkt. Die Gewichtskraft verursacht die Hangabtriebskraft  $F_H$ . Unter Betrachtung der Neigewinkelgeschwindigkeit ergibt sich die in radialer  $x^{(1)}$ -Richtung wirkende Zentrifugalkraft  $F_{ZF}$ . Zur Bestimmung der Bewegungsgleichung kann das zweite Newtonsche Gesetz bezüglich des körperfesten Koordinatensystems angewandt werden.

### **Vorbereitungsaufgabe – V3.1 –**

Herleitung der Bewegungsgleichung des Balls mit Hilfe newtonscher Mechanik:

- Stellen Sie den Impulssatz in x-Richtung sowie den Drallsatz um den Kugelschwerpunkt auf.
- Stellen Sie die kinematischen Beziehungen unter der Annahme eines schlupffrei rollenden Rades zwischen  $x$  und  $\varphi$  auf.
- Berechnen Sie die Rollreibungskraft zwischen Ball und Platte  $F_R$ .
- Stellen Sie die Formeln für die Zentrifugalkraft und Hangabtriebskraft auf.
- Stellen Sie die Bewegungsgleichung des Balls in x-Richtung auf.

Neben den Methoden der newtonschen Mechanik können die Bewegungsgleichungen eines dynamischen Systems auch mit Hilfe des Lagrange-Formalismus hergeleitet werden. Dabei kann die Dynamik eines Systems mit einer einzigen skalaren Gleichung, der Lagrange-Funktion, beschrieben werden. Der Formalismus ist auch in bewegten Bezugssystemen gültig und invariant gegenüber Koordinatentransformationen.

Die Lagrange Funktion lautet

$$L = T - V \quad (1)$$

wobei hier  $T$  die kinetische Energie und  $V$  die potentielle Energie des Systems darstellen. Mit Hilfe der Lagrange-Gleichungen zweiter Art und der Lagrange-Funktion lassen sich die Bewegungsgleichungen des Systems herleiten. Die Lagrange-Gleichungen zweiter Art lauten

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i^*, \quad i = 1, \dots, n, \quad (2)$$

wobei

- $q_i$  ... die i-te verallgemeinerte Koordinate,
- $\dot{q}_i$  ... die zugehörige verallgemeinerte Ableitung,
- $Q_i$  ... die i-te generalisierte nicht-konservative Kraft,
- $n$  ... die Anzahl der Freiheitsgrade

ist. Nach der Berechnung dieser Gleichungen erhält man als Ergebnis die Bewegungsgleichungen des Systems.

### **Vorbereitungsaufgabe – V3.2 –**

Herleitung der Bewegungsgleichung des Balls mit Hilfe des Lagrange Ansatzes:

- Stellen Sie die Geschwindigkeit des Balls in x-, y- und z-Richtung im Koordinatensystem  $e^{(1)}$  auf.
- Berechnen Sie die kinetische Energie des Balls. Drücken Sie diese in Abhängigkeit von  $\dot{x}$  aus.
- Berechnen Sie die potentielle Energie des Balls.
- Stellen Sie die Lagrange-Funktion auf und benennen Sie die generalisierten Koordinaten. Sind nicht-konservativen Kräfte vorhanden?
- Setzen Sie die Lagrange-Funktion in die Lagrange Gleichung ein und berechnen Sie die einzelnen Terme.
- Stellen Sie die Bewegungsgleichung des Balls in x-Richtung auf.

### **3.1.3 Darstellung des Systems im Zustandsraum**

Aus der eindimensionalen Bewegungsgleichung kann im Folgenden das System in eine nichtlineare Zustandsraumdarstellung der Form

$$\dot{x} = f(x, u) \quad (3)$$

$$y = g(x, u) \quad (4)$$

gebracht werden. Dabei soll als Eingangsgröße  $u$  die Winkelgeschwindigkeit  $\dot{\alpha}$  der Platte um die y-Achse dienen. Der Ausgang  $y$  ist die Position des Balls in x-Richtung. Als Zustandsvektor soll  $x = [x, \dot{x}, \alpha]^T$  verwendet werden.

### **Vorbereitungsaufgabe – V3.3 –**

- Bestimmen Sie die nichtlineare Zustandsraumdarstellung des Balls auf der Platte.
- Bestimmen Sie den relativen Grad des Ausgangs.  
Anmerkung: siehe Relativer Grad - Kap. 4.2.1
- Ist das System flach?  
Anmerkung: siehe Byrnes-Isidori-Normalform für Flache Systeme - Kap. 4.2.1

### **3.1.4 Modellverifikation**

Um sicher zu stellen, dass das hier hergeleitete Modell das reale System ausreichend genau beschreiben kann, soll nun eine Modellverifikation durchgeführt werden. Hierfür steht Ihnen eine Simulink Umgebung zur Verfügung, in der Sie Ihr Modell implementieren können. Zudem steht Ihnen eine



Messung des realen Systems für einen Eingangssprung in  $\dot{\alpha}$  zur Verfügung. Die notwendigen Parameter finden Sie in Tabelle 1. **Hinweis:** Durch geschicktes Einsetzen von  $J_B$  kann der Term  $\frac{1}{1 + \frac{J_B}{mR^2}}$  vereinfacht werden.

Symbol	Beschreibung	Wert
$R$	Radius der Kugel	15 mm
$m$	Masse des Kugel (Stahl)	111 g
$J_B$	Trägheitsmoment der Kugel	$\frac{2}{5} mR^2$

Tabelle 1: Parameter des Ball-Platte Versuchsaufbaus

### Vorbereitungsaufgabe – V3.4 –

- Implementieren Sie das nichtlineare Modell in die Simulink Vorlage Praktikumsvorbereitung\ModellVerifikation.slx in den Block nlModell\_Ord3. Verwenden Sie hierfür die Modellparameter aus dem File ModellInit.m.
- Vergleichen Sie das Modell mit der Messung. Sind Abweichungen festzustellen? Worauf können diese zurückgeführt werden?
- Lösen Sie die Bewegungsgleichung aus Abschnitt 3.1.2 nach  $\ddot{x}$  auf. Berechnen Sie in Ihrem Modell die einzelnen Terme der Gleichung von  $\ddot{x}$  und begutachten Sie diese. Ein Term ist dabei von der Schwerkraft abhängig und der andere von der Zentrifugalkraft. Ist der Betrag einer der Terme signifikant größer als der Betrag des Anderen?
- Kann ein Term vernachlässigt werden? Wenn ja, vernachlässigen Sie im zweiten Schritt einen der beiden Terme.
- Stellen Sie für das vereinfachte Modell die nichtlineare Zustandsraumdarstellung auf. Reduzieren Sie nun das Modell auf Ordnung 2 indem Sie als Eingang  $u = \alpha$  verwenden. Der Zustandsvektor ist somit  $\mathbf{x} = [x, \dot{x}]^T$  und der Ausgang bleibt  $y = x$ .
- Bestimmen Sie den relativen Grad des Ausgangs. Was fällt auf?
- Implementieren Sie das reduzierte Modell in die Simulink Vorlage ModellVerifikation.slx in den Block nlModell\_Ord2 und vergleichen es mit dem Modell aus Block nlModell\_Ord3. Rechnen Sie im Folgenden mit diesem System weiter.

### 3.1.5 Linearisierung der Systemgleichungen

Jedes nichtlineare System kann im Arbeitspunkt linearisiert werden, falls dieses stetig differenzierbar ist. Für die Linearisierung eines nichtlinearen Systems werden verschiedene Ableitungsnotationen verwendet. Die partielle Ableitung einer skalaren Funktion  $f(\mathbf{x})$  an der Stelle  $\mathbf{x} = \mathbf{x}_A$  wird als

Zeilenvektor (Gradient)

$$\left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_A} = \left[ \frac{\partial f}{\partial x_1} \quad \cdots \quad \frac{\partial f}{\partial x_n} \right]_{\mathbf{x}=\mathbf{x}_A} \quad (5)$$

definiert. Für eine vektorwertige Funktion  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]^T$  ist die *Jacobi-Matrix* an der Stelle  $\mathbf{x} = \mathbf{x}_A$  definiert durch

$$\left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{\mathbf{x}=\mathbf{x}_A} \quad (6)$$

Zur Linearisierung wird nun zu Abweichungen des Zustands, des Ausgangs und der Stellgrößen vom Arbeitspunkt übergegangen:

$$\mathbf{x}(t) = \mathbf{x}_A + \Delta \mathbf{x}(t) \quad (7)$$

$$\mathbf{y}(t) = \mathbf{y}_A + \Delta \mathbf{y}(t) \quad (8)$$

$$\mathbf{u}(t) = \mathbf{u}_A + \Delta \mathbf{u}(t) \quad (9)$$

wobei hier  $\mathbf{x}_A$  der Arbeitspunkt und  $\mathbf{u}_A$  die Stellgröße im Arbeitspunkt ist. Setzt man diese Gleichungen in die allgemeine Systemgleichung eines nichtlinearen Systems aus Gleichung (3) und (4) ein, so erhält man

$$\dot{\mathbf{x}}_A + \Delta \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}_A + \Delta \mathbf{x}(t), \mathbf{u}_A + \Delta \mathbf{u}(t)) \quad (10)$$

$$\mathbf{y}_A + \Delta \mathbf{y}(t) = \mathbf{g}(\mathbf{x}_A + \Delta \mathbf{x}(t), \mathbf{u}_A + \Delta \mathbf{u}(t)) \quad (11)$$

Mit einer Taylorentwicklung erster Ordnung der rechten Seite um den Arbeitspunkt ergibt sich

$$\dot{\mathbf{x}}_A + \Delta \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}_A, \mathbf{u}_A) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_A}^{\mathbf{u}_A} \Delta \mathbf{x} + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{x}_A}^{\mathbf{u}_A} \Delta \mathbf{u} \quad (12)$$

$$\mathbf{y}_A + \Delta \mathbf{y}(t) = \mathbf{g}(\mathbf{x}_A, \mathbf{u}_A) + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}_A}^{\mathbf{u}_A} \Delta \mathbf{x} + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \right|_{\mathbf{x}_A}^{\mathbf{u}_A} \Delta \mathbf{u} \quad (13)$$

$$(14)$$

Setzt man nun  $\mathbf{x}_A$  und  $\mathbf{u}_A$  in Gleichung (3) und (4) ein, so erkennt man, dass

$$\dot{\mathbf{x}}_A = \mathbf{f}(\mathbf{x}_A, \mathbf{u}_A) \quad (15)$$

$$\mathbf{y}_A = \mathbf{g}(\mathbf{x}_A, \mathbf{u}_A) \quad (16)$$

Es ergibt sich damit das linearisierte System

$$\Delta \dot{\mathbf{x}}(t) = \underbrace{\left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_A}^{\mathbf{u}_A}}_{=: \mathbf{A}} \Delta \mathbf{x} + \underbrace{\left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{x}_A}^{\mathbf{u}_A}}_{=: \mathbf{B}} \Delta \mathbf{u} = \mathbf{A} \Delta \mathbf{x} + \mathbf{B} \Delta \mathbf{u} \quad (17)$$

$$\Delta \mathbf{y}(t) = \underbrace{\left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}_A}^{\mathbf{u}_A}}_{=: \mathbf{C}} \Delta \mathbf{x} + \underbrace{\left. \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \right|_{\mathbf{x}_A}^{\mathbf{u}_A}}_{=: \mathbf{D}} \Delta \mathbf{u} = \mathbf{C} \Delta \mathbf{x} + \mathbf{D} \Delta \mathbf{u} \quad (18)$$

mit

$$\Delta \mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_A \quad \Delta \mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_A \quad \Delta \mathbf{y}(t) = \mathbf{y}(t) - \mathbf{y}_A \quad (19)$$

### Vorbereitungsaufgabe – V3.5 –

- (a) Berechnen Sie die Ruhelage des nichtlinearen Systems Ball auf Platte. Was fällt auf?
- (b) Linearisieren Sie das nichtlineare Modell um die Ruhelage  $\mathbf{x}_A = [0, 0]^T$  und  $\mathbf{u}_A = 0$ .
- (c) Bringen Sie das System in die lineare Zustandsraumdarstellung.
- (d) Implementieren Sie das Modell in die Simulink Vorlage ModellVerifikation.slx in den Block lModell\_Ord2.
- (e) Vergleichen Sie die in Abschnitt 3.1.3 implementierten Modelle mit dem linearisierten Modell. Sind Abweichungen festzustellen?

### 3.1.6 Modellanalyse

Im Folgenden soll das linearisierte Modell analysiert werden.

### Vorbereitungsaufgabe – V3.6 –

- (a) Bestimmen Sie die Eigenwerte des Systems. Ist es stabil?
- (b) Ist das linearisierte System steuerbar?
- (c) Ist das linearisierte System beobachtbar?
- (d) Gilt das Ergebnis global für alle Ruhelagen?

## 3.2 Modellierung der Motor/Plattendynamik

Im Folgenden soll die Motordynamik modelliert werden.

Die Linearaktuatoren besitzen eine Schnittstelle zur Vorgabe einer Sollgeschwindigkeit jedes Motors. Somit kann die Sollwinkelgeschwindigkeit der Platte in beide Richtungen vorgegeben werden. Die Linearaktuatoren stellen diese Sollgeschwindigkeit aufgrund ihrer guten Performanz sehr schnell ein. Der Beschleunigungsvorgang bis zum Erreichen der vorgegebenen Sollgeschwindigkeit kann daher bei der Modellierung der Motor- bzw. Plattendynamik vernachlässigt werden. Als Eingang des Motormodells können somit direkt die Winkelgeschwindigkeiten der Platte  $\mathbf{u} = [\dot{\alpha}, \dot{\beta}]^T$  verwendet werden.

Im Folgenden wird lediglich die Modellierung einer Achse betrachtet. Die Modellierung der anderen Achse ist analog dazu. Gesucht ist nun ein Motormodell mit dem Eingang  $u = \dot{\alpha}$  und dem Ausgang  $y = \alpha$ .

**Vorbereitungsaufgabe – V3.7 –**

- (a) Stellen Sie die Differentialgleichung für den Plattenwinkel  $\alpha$  auf.
- (b) Stellen Sie das Motormodell in Zustandsraumdarstellung auf. Wie lautet das regelungstechnische Blockschaltbild dazu?

## 4 Reglerentwurf

Im Folgenden sollen Regler entworfen werden, die in der Lage sind, das System zu stabilisieren und einer Trajektorie zu folgen. Dabei sollen die hier vollzogenen Betrachtungen auf eine Dimension der Platte im Raum, die  $x$ -Richtung, beschränkt sein. Die Erweiterung auf die 2. Dimension passiert in Abschnitt 5.

### 4.1 Stabilisierender Regler

Zunächst soll das linearisierte Modell aus Abschnitt 3.1.5 für einen linearen Reglerentwurf dienen. Das Ziel ist es zunächst, den Ball in seinem Arbeitspunkt  $x = 0$  zu stabilisieren. Später soll der Ball der in Abschnitt 4.3 berechneten Trajektorie folgen.

Betrachtet man das zu regelnde lineare System und die gegebenen Randbedingungen, so lassen sich diese wie in Tabelle 2 zusammenfassen. Hinzu kommen die Steuer- und Beobachtbarkeits-

Randbedingung	Beschreibung	Wert
Verfügbare Messungen	Plattenwinkel, Ballposition	$\alpha, x$
Verfügbare Stellgrößen	Plattenwinkelgeschwindigkeit	$\dot{\alpha}$
Stellgrößenbeschränkung	Plattenwinkelgeschwindigkeit	$\dot{\alpha}_{\max} = \pm 30^\circ/\text{s}$
Stellgrößenbeschränkung	Plattenwinkel	$\alpha_{\max} = \pm 5.3^\circ$
Ballmodell	2. Ordnung, linear, Eingang: $\alpha$	
Motormodell	1. Ordnung, linear, Eingang: $\dot{\alpha}$	

Tabelle 2: Randbedingungen zum Reglerentwurf

eigenschaften die Sie in Abschnitt 3.1.6 berechnet haben. Das Gesamtsystem besteht aus einer Motordynamik sowie einer Balldynamik. Die Position des Balls sowie der Winkel der Platte kann direkt gemessen werden.

Aus diesen Randbedingungen heraus bietet sich eine Kaskadenregelung zur Regelung der Ballposition an. Die Struktur ist in Abbildung 4 dargestellt. Dadurch, dass der Plattenwinkel direkt gemessen werden kann, kann dieser für den inneren Regelkreis zur Regelung des Plattenwinkels  $\alpha$  verwendet werden. Die Stellgröße hierfür ist die Plattenwinkelgeschwindigkeit  $\dot{\alpha}$ , wie in Tabelle 2 beschrieben. Hierfür eignet sich ein P-Regler. Durch diesen stellt sich vom Eingang des inneren Regelkreises  $\alpha_d$  zum Ausgang des inneren Regelkreises  $\alpha$  eine geeignete Dynamik ein, die mit Hilfe des Verstärkungsfaktors des P-Reglers eingestellt werden kann. Hierbei ist es möglich die Dynamik so schnell einzustellen, dass der echte Plattenwinkel  $\alpha$  dem vorgegebenen Plattenwinkel  $\alpha_d$  gut folgt und diese Dynamik daher für den äußeren Kreis vernachlässigt werden kann. Das ist dann der Fall, wenn die Dynamik des inneren Regelkreises signifikant schneller ist als die Dynamik des äußeren Regelkreises.

Als Regler für den äußeren Regelkreis eignet sich ein PD-Regler. Dieser ist für die Regelung der Ballposition zuständig und nutzt hierfür als Stellsignal den Plattenwinkel  $\alpha$ . Als Modell zur Auslegung des Reglers soll hier das lineare Modell 2. Ordnung verwendet werden. Die Auslegung des PD-Reglers kann zum Beispiel mit Hilfe der Matlab Funktion `[C, Info] = pidtune(Sys, 'pd',  $\omega_c$ )` erfolgen,

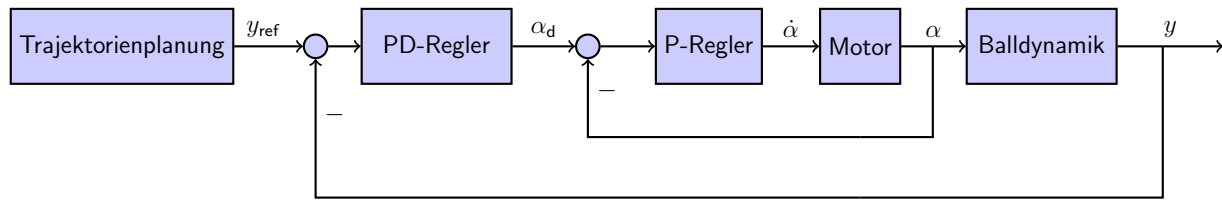


Abbildung 4: Struktur der Kaskadenregelung. Innerer Regelkreis für den Plattenwinkel  $\alpha$  und äußerer Regelkreis für die Ballposition

wobei hier Sys das Modell und  $\omega_c$  die Durchtrittsfrequenz der Amplitude der Übertragungsfunktion bei 0 db des offenen Regelkreises darstellt. Umso größer  $\omega_c$  gewählt wird, umso schneller wird der Regelkreis. Umso kleiner  $\omega_c$  gewählt wird umso größer wird in der Regel die Stabilitätsreserve.

Zur Auslegung des Reglers kann der geschlossene Regelkreis zum Beispiel für einen Anfangswertfehler in der Ballposition simuliert werden. Um die Gleichungen für den geschlossenen Regelkreis (unter Vernachlässigung des inneren Regelkreises, also  $\alpha_d = \alpha$  aufzustellen, muss das Regelgesetz des PD-Reglers

$$\begin{aligned} u &= k_p (y_{\text{ref}} - y) + k_d (\dot{y}_{\text{ref}} - \dot{y}) \\ &= k_p (y_{\text{ref}} - x_1) + k_d (\dot{y}_{\text{ref}} - x_2) \end{aligned} \quad (20)$$

in die Modellgleichungen des Balldynamik-Modells eingesetzt werden. Dabei kann  $y_{\text{ref}} = 0$  und  $\dot{y}_{\text{ref}} = 0$  gesetzt werden, da hier lediglich der Regelvorgang eines Anfangswertfehlers betrachtet werden soll. Die Simulation des Regelvorgangs kann mit der Matlab Funktion `initial()` erfolgen.

### Vorbereitungsaufgabe – V3.8 –

- Berechnen Sie die Übertragungsfunktion des inneren geschlossenen Regelkreises vom Eingang  $\alpha_d$  zum Ausgang  $\alpha$ . Wie wird diese Übertragungsfunktion genannt?
- Berechnen Sie die Verstärkung  $K_\alpha$  des P-Reglers so, dass die Anstiegszeit  $T$  bei 0.1 s liegt.
- Legen Sie in dem vorgefertigten m-File `ModelInit.m` den PD-Regler für den äußeren Regelkreis mit Hilfe der Matlab Funktion `pidtune()` aus.
- Simulieren Sie dazu den geschlossenen Regelkreis mit einem Anfangsfehler in der Ballposition von 20 cm. Der Anfangswert des Zustands ist somit  $x_0 = [0.2, 0]^T$ .
- Plotten Sie den Verlauf der Ballposition, der Ballgeschwindigkeit und der Stellgröße Plattenwinkel. Werden die Stellgrößenbeschränkungen im Plattenwinkel eingehalten? Legen Sie Ihren PD-Regler eventuell neu aus.

## 4.2 Erweiterung um eine flachheitsbasierte Vorsteuerung

Nun soll unter Ausnutzung der Eigenschaften flacher Systeme eine flachheitsbasierte Vorsteuerung entworfen werden.

Im Folgenden beziehen sich die Formeln auf ein nichtlineares Zustandsraummodell in der allgemeinen eingangsaffinen Form mit skalarem Ein- und Ausgang

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u, \\ y &= h(\mathbf{x}), \\ \text{mit } \mathbf{x}(t) &\in \mathbb{R}^n, u(t) \in \mathbb{R}, y(t) \in \mathbb{R}, \mathbf{f}(\mathbf{0}) = h(\mathbf{0}) = \mathbf{0}, \\ \mathbf{x}(0) &= \mathbf{x}_0\end{aligned}\tag{21}$$

Zum besseren Verständnis werden die Berechnungen beispielhaft an einem einfachen linearen System 2. Ordnung vorgestellt. Dieses stellt einen Spezialfall des allgemeinen eingangsaffinen nichtlinearen SISO-Systems dar:

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} x_2 \\ x_1 + x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= x_1\end{aligned}\tag{22}$$

### 4.2.1 Grundlagen

**Relativer Grad** Eine charakteristische Kenngröße nichtlinearer Systeme ist der relative Grad  $r$ . Dieser beschreibt, die wievielte zeitliche Ableitung des Ausgangs  $y$  vom Eingang  $u$  des Systems direkt beeinflusst wird.

Anhand des Beispielsystems aus Gl. (22) berechnen sich die Ableitungen des Ausgangs zu

$$\begin{aligned}y &= x_1 \\ \dot{y} &= \dot{x}_1 = x_2 \\ \ddot{y} &= \dot{x}_2 = x_1 + x_2 + u.\end{aligned}\tag{23}$$

Damit ist der relative Grad des System  $r = 2 = n$  und das System ist flach.

Eine mathematische Definition erfolgt unter Verwendung der Lie-Ableitung des Ausgangs, welche die Ableitung der skalaren, zustandsabhängigen Funktion  $h(\mathbf{x})$  in Richtung des Vektorfeldes  $\mathbf{f}(\mathbf{x})$  darstellt. Die Ableitungen des Ausgangs  $y$  nach der Zeit können dann mithilfe der Lie-Ableitungen

$$\begin{aligned}\dot{y} &= \mathcal{L}_f h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}), \\ {}^{(k)}y &= \mathcal{L}_f^k h(\mathbf{x}) = \frac{\partial \mathcal{L}_f^{k-1} h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}), \quad \mathcal{L}_f^0 h(\mathbf{x}) = h(\mathbf{x}), \\ \mathcal{L}_g \mathcal{L}_f h(\mathbf{x}) &= \frac{\partial \mathcal{L}_f h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x})\end{aligned}$$

dargestellt werden. Der relative Grad  $r$  am Arbeitspunkt  $\mathbf{x} = \mathbf{0}$  kann als diejenige natürliche Zahl  $r$  beschrieben werden, für die

$$(i) \mathcal{L}_g \mathcal{L}_f^k h(\mathbf{x}) = 0, \text{ für } k = 0, 1, \dots, r-2$$

$$(ii) \mathcal{L}_g \mathcal{L}_f^{r-1} h(\mathbf{x}) \neq 0$$

gilt. Ist  $\mathcal{L}_g \mathcal{L}_f^{r-1} h(\mathbf{0}) = 0$ , aber  $\mathcal{L}_g \mathcal{L}_f^{r-1} h(\mathbf{x}) \neq 0$  in einer Umgebung von 0, spricht man von einem *nicht wohldefinierten* relativen Grad  $r$ . Die Ableitungen des Ausgangs  $y = h(\mathbf{x})$  lassen sich damit als

$$\begin{aligned} {}^{(k)}y &= \mathcal{L}_f^k h(\mathbf{x}) \\ {}^{(r)}y &= \mathcal{L}_f^r h(\mathbf{x}) + \mathcal{L}_g \mathcal{L}_f^{r-1} h(\mathbf{x}) u \end{aligned}$$

darstellen, wobei die Ableitung  ${}^{(r)}y$  des Ausgangs erstmals direkt vom Eingang  $u$  abhängt.

**Byrnes-Isidori-Normalform für flache Systeme** Falls der relative Grad des Ausgangs  $r$  gleich der Systemordnung ist, also  $r = n$ , dann ist das System flach und der Ausgang wird als flacher Ausgang bezeichnet. Mit Hilfe des Transformationsvektors

$$\mathbf{z} = \Phi(\mathbf{x}) = \begin{bmatrix} y \\ \dot{y} \\ \vdots \\ {}^{(r-1)}y \end{bmatrix} = \begin{bmatrix} h(\mathbf{x}) \\ \mathcal{L}_f h(\mathbf{x}) \\ \vdots \\ \mathcal{L}_f^{r-1} h(\mathbf{x}) \end{bmatrix}. \quad (24)$$

kann das System in die *Byrnes-Isidori-Normalform*<sup>3</sup> gebracht werden. Der transformierte Zustandsvektor  $\mathbf{z}$  ist nur abhängig vom Ausgang und dessen zeitlichen Ableitungen. Mit Hilfe von  $\mathbf{z}$  lassen sich  $\mathbf{x}$ ,  $y$  und  $u$  parametrieren. Das flache System lässt sich dann in der BINF durch die Integratorkette

$$\begin{aligned} \dot{z}_1 &= z_2 \\ &\vdots \\ \dot{z}_{n-1} &= z_n \\ \dot{z}_n &= a(\mathbf{z}) + b(\mathbf{z})u \end{aligned} \quad (25)$$

$$\begin{aligned} y &= z_1 \\ \text{mit } a(\mathbf{z}) &= \mathcal{L}_f^n h \circ \Phi^{-1}(\mathbf{z}) \text{ und } b(\mathbf{z}) = \mathcal{L}_g \mathcal{L}_f^{n-1} h \circ \Phi^{-1}(\mathbf{z}) \end{aligned} \quad (26)$$

darstellen, wobei der Operator  $\circ \Phi^{-1}(\mathbf{z})$  die Anwendung der differentiellen Parametrierung in Form der inversen Koordinatentransformation auf  $\mathbf{x}$  beschreibt.

Für das Beispielsystem aus Gl. (22) ergibt sich die Transformation

$$\begin{aligned} z_1 &= y = x_1 \\ z_2 &= \dot{y} = \dot{x}_1 = x_2. \end{aligned}$$

<sup>3</sup>kurz: BINF



Folglich liegt das System bereits in der Byrnes-Isidori-Normalform vor. Die Zustands- und Ausgangsparametrierung ergibt sich direkt zu

$$\begin{aligned}x_1 &= z_1 \\x_2 &= z_2 \\y &= x_1 = z_1.\end{aligned}$$

Die Eingangsparametrierung wird aus der dritten Zeile aus Gl. (23) berechnet

$$\begin{aligned}u &= \ddot{y} - x_2 - x_1 \\&= \ddot{y} - \dot{y} - y.\end{aligned}$$

### Vorbereitungsaufgabe – V3.9 –

- (a) Führen Sie die nichtlineare Zustandsraumdarstellung des Balls auf der Platte mit vernachlässigter Zentrifugalkraft auf eine eingangsaffine Form nach Gleichung (21) zurück. Betrachten Sie hierzu das System 3. Ordnung und das System 2. Ordnung.  
Anmerkung: Betrachten Sie im Folgenden nur noch das System 2. Ordnung.
- (b) Ist das System flach?
- (c) Berechnen Sie die Zustandstransformation  $\Phi(\mathbf{x})$  für das Ball auf Platte System.
- (d) Schreiben Sie das Modell in Byrnes-Isidori-Normalform um.

### 4.2.2 Flachheitsbasierte Vorsteuerung

Um ein noch besseres Führungsverhalten des Balls bezüglich der vorgegebenen Trajektorie zu erzielen, soll eine flachheitsbasierte Vorsteuerung konzipiert werden. Die Stabilisierung um die Referenztrajektorie soll mit der bereits verwendeten Kaskadenregelung aus Abschnitt 4.1 erfolgen.

Der Entwurf der Vorsteuerung erfolgt gemäß einer Zwei-Freiheitsgrade-Struktur unabhängig vom eingesetzten Regler. Die Struktur aus Abbildung 4 wird um diese erweitert, wie in Abbildung 5 anhand des grünen Pfads dargestellt. Bei der Bestimmung einer geeigneten Vorsteuerung wird Nutzen aus der differentiellen Parametrierung der Zustände durch den flachen Ausgang gezogen. Das inverse Modell wird bestimmt, indem das flache System (25) in BINF betrachtet wird. Die Differentialgleichung für den  $n$ -ten Zustand

$$\dot{z}_n = \overset{(n)}{y} = \mathcal{L}_f^n h(\mathbf{x}) + \mathcal{L}_g \mathcal{L}_f^{n-1} h(\mathbf{x}) u$$

nach der Stellgröße  $u$  aufgelöst ergibt

$$u(\mathbf{x}) = \frac{\overset{(n)}{y} - \mathcal{L}_f^n h(\mathbf{x})}{\mathcal{L}_g \mathcal{L}_f^{n-1} h(\mathbf{x})}.$$

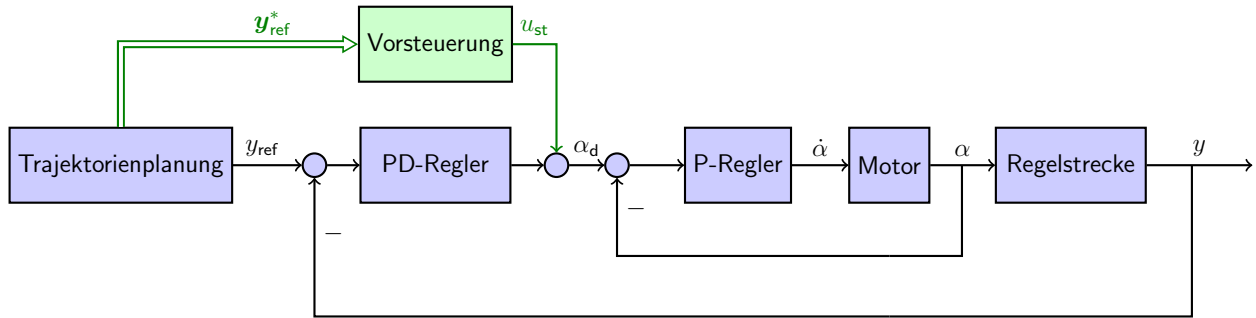


Abbildung 5: Kaskadierte Struktur mit zwei Freiheitsgraden sowie einer flachheitsbasierten Vorsteuerung mit Hilfe des inversen Modells der Strecke. Die Vorsteuerung berechnet den für  $y_{\text{ref}}^*$  benötigten Stellgrößenverlauf  $u_{\text{st}}$ .

Mit der Parametrierung von  $x$  durch den Ausgang und dessen Ableitungen kann die Vorsteuerung  $u_{\text{st}}(t)$  für die Solltrajektorie  $y_{\text{ref}}(t)$  mit

$$u_{\text{st}}(t) = \frac{{}^{(n)}y_{\text{ref}}(t) - \mathcal{L}_f^n h(x) \circ \Phi^{-1} \left( y_{\text{ref}}(t), \dot{y}_{\text{ref}}(t), \dots, {}^{(n-1)}y_{\text{ref}}(t) \right)}{\mathcal{L}_g \mathcal{L}_f^{n-1} h(x) \circ \Phi^{-1} \left( y_{\text{ref}}(t), \dot{y}_{\text{ref}}(t), \dots, {}^{(n-1)}y_{\text{ref}}(t) \right)} \quad (27)$$

berechnet werden, wobei der Operator  $\circ \Phi^{-1}(\cdot)$  die Anwendung der differentiellen Parametrierung in Form der inversen Koordinatentransformation auf  $x$  beschreibt. Der Zustand kann also durch die inverse von  $\Phi$  berechnet werden:

$$x = \Phi^{-1}(z) \quad (28)$$

### Vorbereitungsaufgabe – V3.10 –

- Berechnen Sie die inverse Koordinatentransformation  $x = \Phi^{-1}(z)$ .
- Berechnen Sie die flachheitsbasierte Vorsteuerung in Abhängigkeit der Referenztrajektorie und deren Ableitungen.

## 4.3 Trajektorienplanung

Zur Verwendung der Vorsteuerung werden im Folgenden Trajektorien offline geplant. Dazu werden die zwei Fälle *Arbeitspunktwechsel* und *periodische Schwingung* betrachtet.

### 4.3.1 Arbeitspunktwechsel

Der Wechsel zwischen zwei Sollpositionen soll für ein besseres Führungsverhalten mit einer flachheitsbasierten Vorsteuerung realisiert werden. Die Solltrajektorie  $y_{\text{ref}}(t)$  wird mit einem Polynom

geeigneter Ordnung ermittelt.

#### **Vorbereitungsaufgabe – V3.11 –**

- (a) Bestimmen Sie die Anfangs- und Endbedingungen des Polynoms. Von welcher Ordnung ist das Polynom? Verwenden Sie hierzu das angehängte Hilfsblatt auf Seite 23.
- (b) Berechnen Sie die notwendigen Zeitableitungen von  $y_{\text{ref}}$ .
- (c) Für welche Transitionszeit  $T$  ist der Vorsteuerungsentwurf realisierbar?

#### **4.3.2 Periodische Schwingung**

Im Folgenden soll einer periodischen Solltrajektorie  $y_{\text{ref}}(t) = f(R_p, \omega, t)$  gefolgt werden, wobei  $R_p$  dessen Amplitude und  $\omega$  deren Kreisfrequenz ist.

#### **Vorbereitungsaufgabe – V3.12 –**

- (a) Schreiben Sie die Funktion für eine periodische Schwingung auf. Achten Sie dabei darauf, dass die Ableitung der Funktion zum Zeitpunkt  $t = 0$ , also die Soll-Geschwindigkeit des Balles 0 ist. Die Soll-Position des Balles zum Zeitpunkt  $t = 0$  muss nicht 0 sein.
- (b) Berechnen Sie die erste und zweite Zeitableitung von  $y_{\text{ref}}$ .

## 5 Versuchsdurchführung

Das folgende Kapitel führt Sie durch den experimentellen Teil des Praktikums.

### 5.1 Implementierung in den Simulationsumgebungen

Nun haben Sie die Möglichkeit Ihre zu Hause ausgelegten Algorithmen in der Simulation zu testen. Anschließend werden sie auf den realen Versuch übertragen.

#### 5.1.1 Implementierung der berechneten Modelle und des Kaskadenreglers

Im Folgenden soll Ihr Systemmodell aus Abschnitt 3.1.5 in die vorhandene Vorlage implementiert werden. Darüber hinaus soll Ihr stabilisierender Regler aus Abschnitt 4.1 implementiert und mit Simulink simuliert werden.

#### Praktikumsaufgabe – P3.1 –

- (a) Starten Sie Matlab 2013a und öffnen Sie den Ordner \Praktikumsversuch\Simulation\1D.
- (b) Implementieren Sie in das vorgefertigte m-File KaskadeModellInit.m Ihre Auslegung des P-Reglers sowie das lineare Ballmodell und die Routine zur Auslegung des PD-Reglers. Nutzen Sie dabei die bereits angegebenen Variablennamen.
- (c) Implementieren Sie in die Vorlage KaskadeControlled.slx das Motormodell in den Block Motor 1in, das lineare Ballmodell in den Block Ball 1in, den P-Regler, sowie den PD-Regler anstatt der Konstante Constant und simulieren Sie den Anfangswertfehler aus der vorigen Aufgabe. Simulieren Sie 5s. Tipp: verwenden Sie  $x_1$  und  $x_2$  als Rückführung für den P- und D-Anteil des PD-Reglers.
- (d) Legen Sie den PD-Regler so aus, dass Sie mit dem Ergebnis zufrieden sind. Beachten Sie dabei, dass die Stellgrößenbeschränkungen in  $\alpha$  und  $\dot{\alpha}$  eingehalten werden.  
Anmerkung: siehe Tabelle 2 - Randbedingungen zum Reglerentwurf
- (e) Implementieren Sie die Solltrajektorie und deren Ableitungen, welche in Abschnitt 4.3 hergeleitet wurden, innerhalb des Blocks Trajektorienplaner. Nutzen Sie für die periodische Solltrajektorie die vorhandenen Variablen aus dem m-File KaskadeModellInit.m. Gestalten Sie das Modell so, dass Sie zwischen der Solltrajektorie und dem Sollarbeitspunkt  $x = 0$  umschalten können. Simulieren Sie nun den geschlossenen Regelkreis für die Solltrajektorie aus Abschnitt 4.3. Was stellen Sie fest? Vergleichen Sie die Solltrajektorie, welche Sie aus dem Ansatz des Arbeitspunktwechsels bestimmt hatten, mit dem der periodischen Schwingung.
- (f) Probieren Sie unterschiedliche Auslegungen Ihres PD-Reglers aus. Wie verändert sich das Folgeverhalten? Probieren Sie mit den verschiedenen Auslegungen auch das Ausregeln des Anfangsfehlers aus.

- (g) Am realen System kann  $x_2$  nicht gemessen werden. Welche Möglichkeiten kennen Sie um den PD-Regler in dieser Weise auch am realen System verwenden zu können? Bedenken Sie, dass Sie im realen System Messrauschen haben.

### 5.1.2 Implementierung Flachheitsbasierte Vorsteuerung

Im Folgenden soll der Regler aus Abschnitt 5.1.1 durch die flachheitsbasierte Vorsteuerung unterstützt werden.

#### Praktikumsaufgabe – P3.2 –

- (a) Implementieren Sie die flachheitsbasierte Vorsteuerung in Ihr bereits modifiziertes Modell `KaskadeControlled.slx` zusätzlich zu Ihrem Regler in den Block `Flachheitsbasierte_Vorsteuerung`.
- (b) Simulieren Sie das System mit flachheitsbasierter Vorsteuerung und Regelung für die Trajektorie der periodischen Schwingung.
- (c) Führt der Ball die vorgegebene Trajektorie besser ab als ohne Vorsteuerung? Warum?
- (d) Vergleichen Sie den Regleranteil und den Vorsteueranteil an der Stellgröße mit und ohne Vorsteuerung. Was fällt Ihnen auf?
- (e) Können Sie nun Ihren PD-Regler weniger schnell auslegen ohne große Einbußen im Trajektorienfolgeverhalten zu haben?
- (f) Was spricht in der Praxis gegen eine schnelle PD-Regler Auslegung?

### 5.1.3 Implementierung der Trajektorienplanung für den Arbeitspunktwechsel (2D)

Im Praktikumsversuch soll der Ball neben der x-Richtung, auch eine Bewegung in y-Richtung ausführen. Nun soll der Ball auf einen beliebigen Punkt  $(x_p, y_p)$  der Platte bewegt werden.

#### Praktikumsaufgabe – P3.3 –

- (a) Erweitern Sie Ihre bisher verwendete Solltrajektorie auf die zweite Dimension in y-Richtung, sodass der Ball auf dem Punkt  $(x_p, y_p)$  platziert wird.
- (b) Bestimmen Sie die zwei Zeitableitungen der Solltrajektorie in y-Richtung.

#### 5.1.4 Implementierung der Trajektorienplanung für die Kreisbahn (2D)

Im Praktikumsversuch soll der Ball auch entlang einer zweidimensionalen Trajektorie bewegt werden. Eine geeignete Trajektorie stellt dabei eine Kreistrajektorie dar. Im Folgenden soll eine Referenztrajektorie für eine Kreisbahn aufgestellt werden.

##### Praktikumsaufgabe – P3.4 –

- (a) Erweitern Sie Ihre bisher verwendete Solltrajektorie auf die zweite Dimension in y-Richtung, sodass eine Kreistrajektorie entsteht.
- (b) Bestimmen Sie die zwei Zeitableitungen der Solltrajektorie in y-Richtung.

#### 5.1.5 Erweiterung der Regler auf die 2. Dimension

Nachdem Sie nun Ihren Regler anhand des eindimensionalen linearen Modells ausgelegt und getestet haben, soll dieser anhand eines komplexen verkoppelten zweidimensionalen Modells des Balls auf der Platte auf 2 Dimensionen erweitert werden. Hierbei soll für die zweite Dimension in y-Richtung der selbe Regler wie für die x-Richtung verwendet werden.

##### Praktikumsaufgabe – P3.5 –

- (a) Öffnen Sie den Ordner Praktikumsversuch\Simulation\2D\
- (b) Kopieren Sie den Inhalt Ihres m-Files `KaskadeModelInit.m` aus Abschnitt 5.1.1 in das in diesem Ordner befindliche File. Achten Sie darauf, dass keine Doppelbenamungen mit den Variablen im Skript `Control_parameter.m` auftreten.
- (c) Implementieren Sie im Modell `KaskadeControlled_2D` Ihre Referenztrajektorie in den Block `Trajektorienplaner` für die x- und y-Richtung.
- (d) Implementieren Sie Ihren P- und PD-Regler in die Blöcke `P-Regler` und `PD-Regler`. Fügen Sie dabei jeweils die x- und y-Richtung einer Größe zu einem Vektor zusammen. Nutzen Sie diese Vektoren für Ihre Regler.
- (e) Implementieren Sie Ihre flachheitsbasierte Vorsteuerung in den Block `Flachheitsbasierte_Vorsteuerung`.
- (f) Simulieren und testen Sie Ihre Regler ohne Vorsteuerung für den Anfangswertfehler und die Kreistrajektorie. Zur Auswertung der Simulation kann das Skript `Auswertung_simu` verwendet werden.
- (g) Schalten Sie nun Ihre Vorsteuerung hinzu und vergleichen Sie das Ergebnis. Stellen Sie die Regler so ein, dass die Stellgrößen nicht zu groß sind.

## Polynomial Reference Trajectories for Set Point Transitions

The flatness-based tracking control of a set point transition in a presettable transition time  $T > 0$  requires the planing and calculation of a sufficiently smooth reference trajectory  $z_d(t)$ ,  $t \in [0, T]$  for the flat output  $z$ . In the case of  $n$ -th order SISO-systems, the reference trajectory has to be  $n$ -times differentiable, for the input trajectory to be continuous. The coefficients  $a_i$ ,  $i = n + 1, \dots, 2n + 1$  of the polynomial reference trajectory

$$z_d(t) = z_0 + (z_T - z_0) \sum_{i=n+1}^{2n+1} a_i \left(\frac{t}{T}\right)^i, \quad t \in [0, T] \quad (1)$$

are set by  $2(n + 1)$  conditions for both (steady state) set points

$$z_d(0) = z_0, \quad z_d(T) = z_T, \quad z_d^{(i)}(0) = z_d^{(i)}(T) = 0, \quad i = 1, \dots, n.$$

The transition time  $T > 0$  is chosen depending on the system dynamics and possible restrictions of the input trajectory. The following table shows the coefficients  $a_i$  for the system order  $n = 1$  to  $n = 7$ ; due to  $z_d(T) = z_T$  the coefficients satisfy the easily verifiable condition  $\sum_{i=n+1}^{2n+1} a_i = 1$ .

$n$	$a_{n+1}$	$a_{n+2}$	$a_{n+3}$	$a_{n+4}$	$a_{n+5}$	$a_{n+6}$	$a_{n+7}$	$a_{n+8}$
1	3	-2						
2	10	-15	6					
3	35	-84	70	-20				
4	126	-420	540	-315	70			
5	462	-1980	3465	-3080	1386	-252		
6	1716	-9009	20020	-24024	16380	-6006	924	
7	6435	-40040	108108	-163800	150150	-83160	25740	-3432

In the following plot, reference trajectories  $z_d(t)$  according to (1) are displayed for the orders  $n = 1$  to  $n = 7$  in the time range  $t \in [0, 2]$ . As the plot shows, the trajectories are getting steeper for increasing system order  $n$ .

