



# Einführung in die Robotik

Vorlesung 8

08 Januar 2008

Dr. Mohamed Oubbati  
Institut für Neuroinformatik

WS 2007/2008

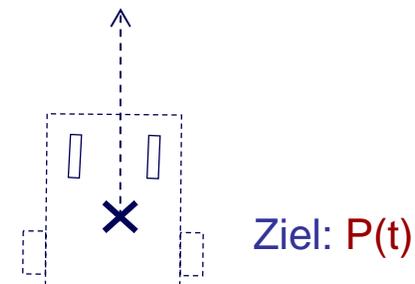
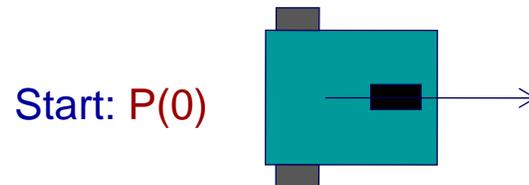
**Wiederholung vom letzten Mal !**

# Rückwärtskinematik

If I want this to happen, what should I do?

**Gegeben:** start und ziel Positionen:  $P(0) = [0,0,0]$ ,  $P(t) = [x(t),y(t),\theta(t)]$

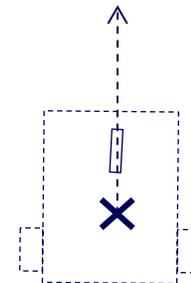
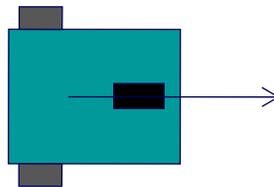
**Aufgabe:** Finde Steuersequenz  $V=[v_r, v_l]$ , die die Positionsänderung  $P(0) \rightarrow P(t)$  realisieren.



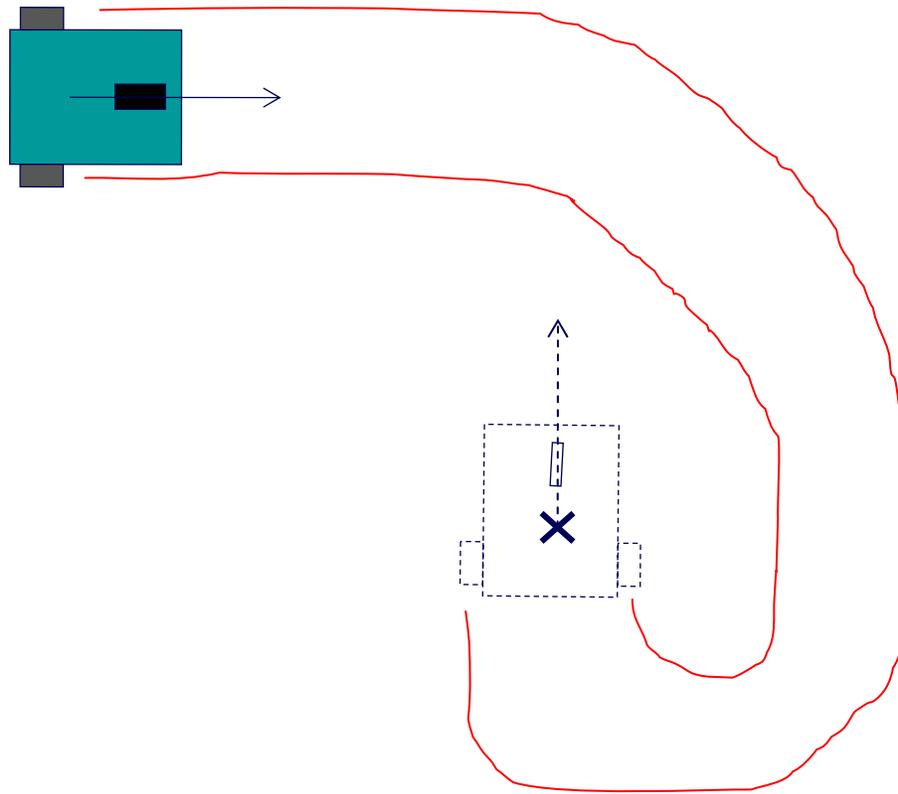
Ziel:  $P(t)$

## Lösung

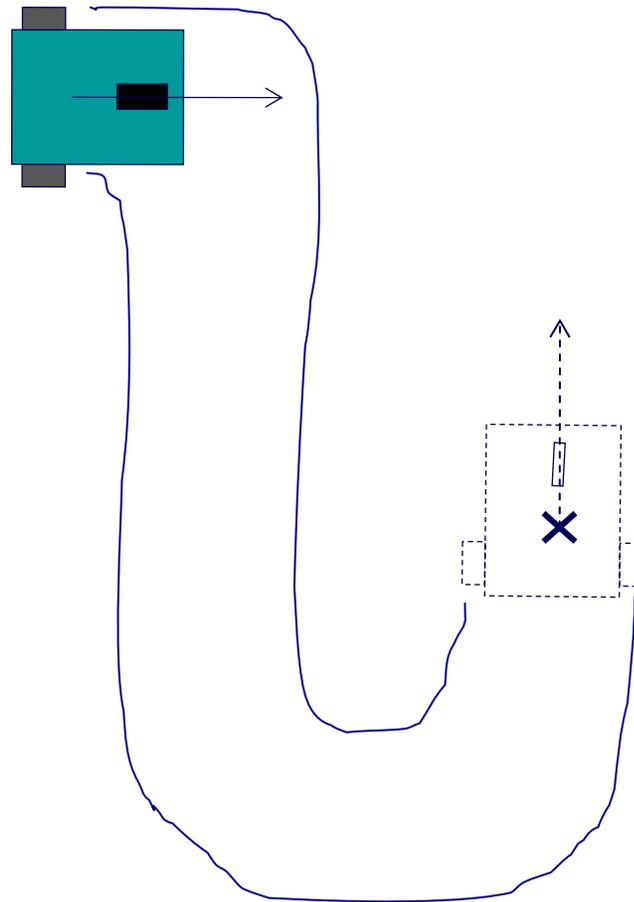
Es gibt mehrere Lösungen!



## Beispiel 1



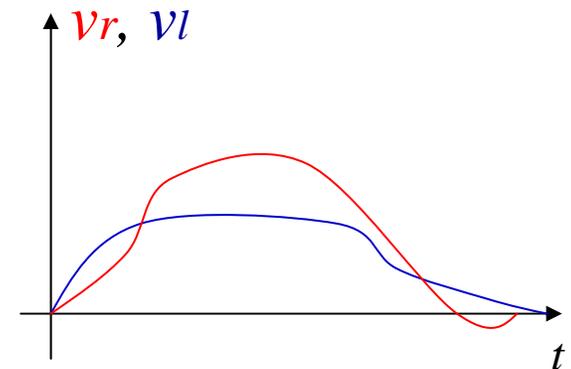
## Beispiel 2



Eine Lösung zu finden ist nicht ein Problem, aber die beste zu finden...

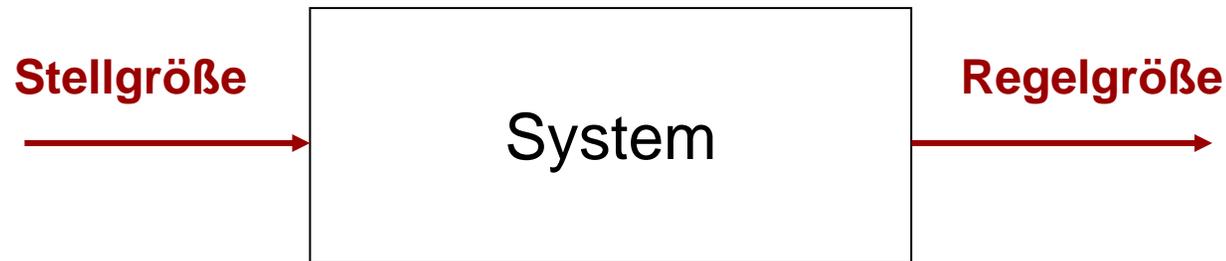
Es hängt von der Definition „beste“ ab:

- schnell
- Energie effizient
- smoothest velocity



# Was ist Regelung?

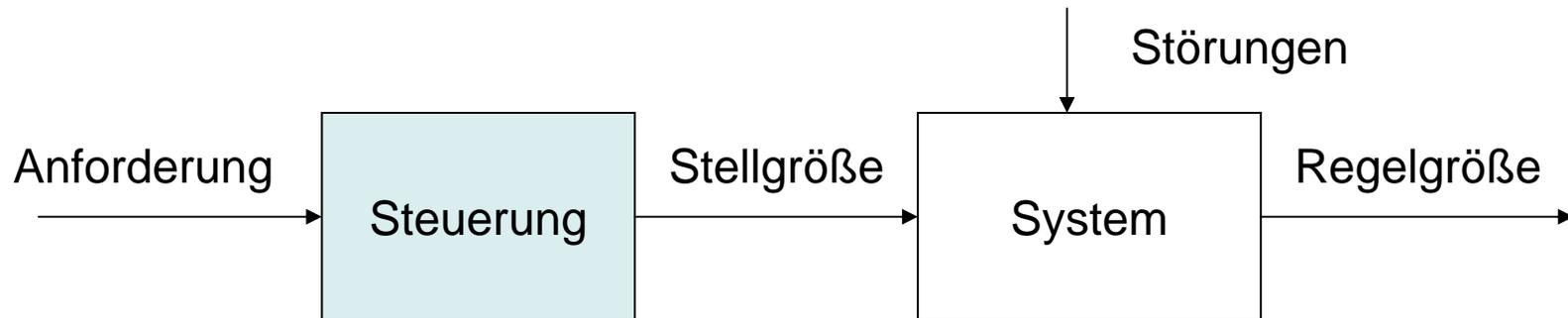
**Regelung** ist eine gezielte Beeinflussung (Stellgröße) dynamischer Systeme, so dass eine gewünschte Betriebsart (Regelgröße) eingestellt wird.



# Unterschied Steuerung - Regelung

## Steuerung (open loop control)

Die Steuerung wirkt auf das Eingangssignal und beeinflusst damit das Ausgangssignal.



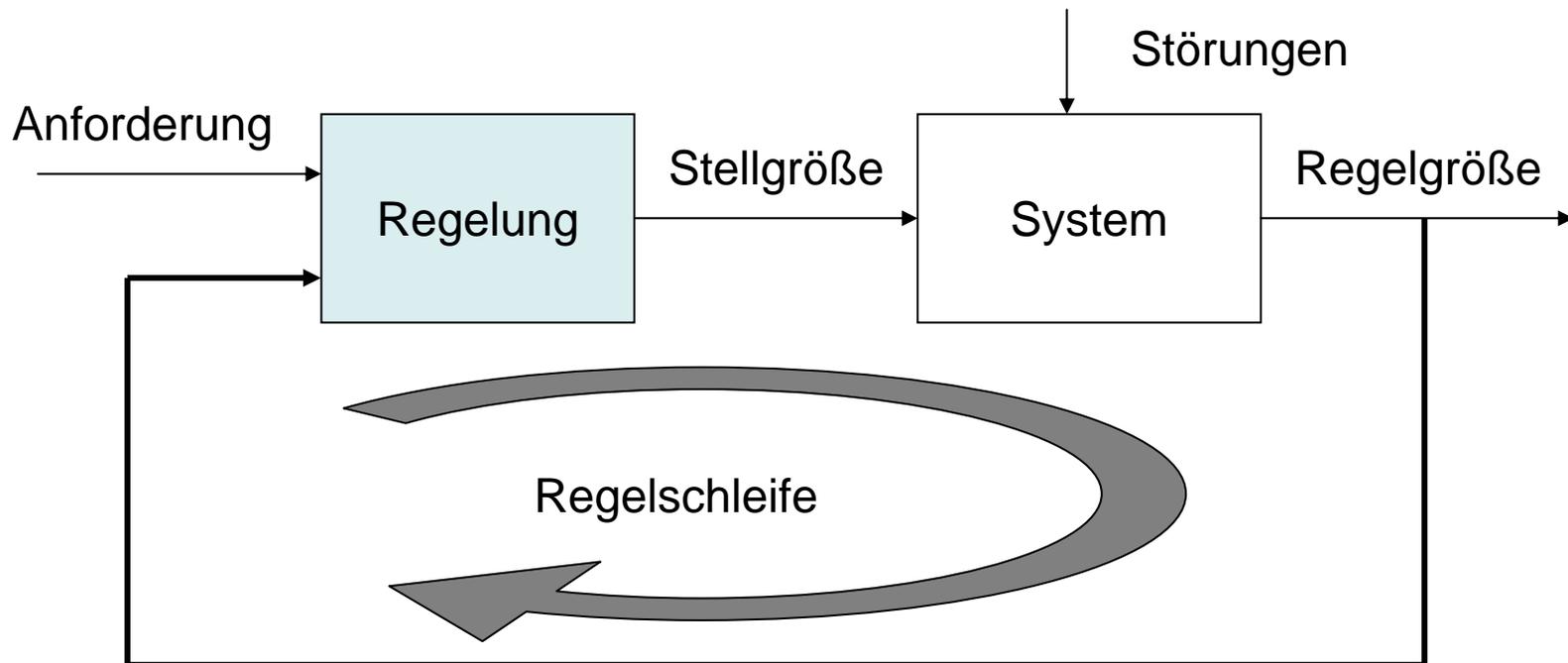
## Problem

Die Steuerung **“weiß” nicht**, ob die Regelgröße den gewünschten Wert hat!

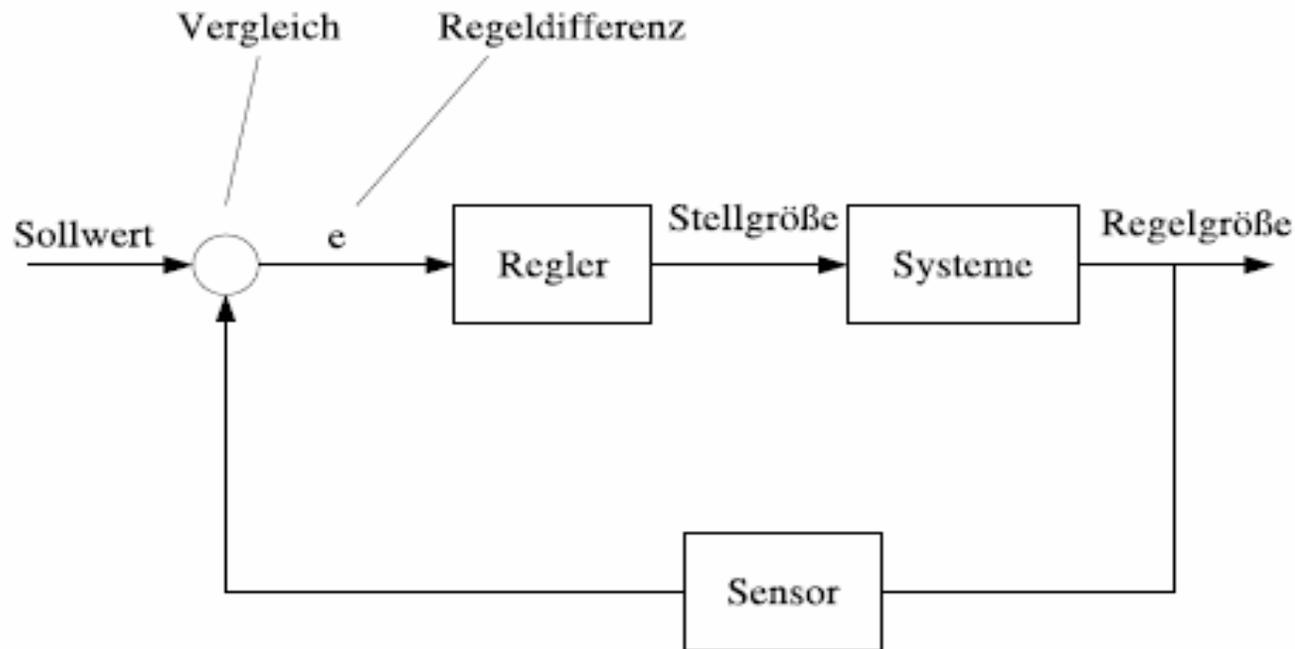
# Unterschied Steuerung - Regelung

## Regelung (closed loop control)

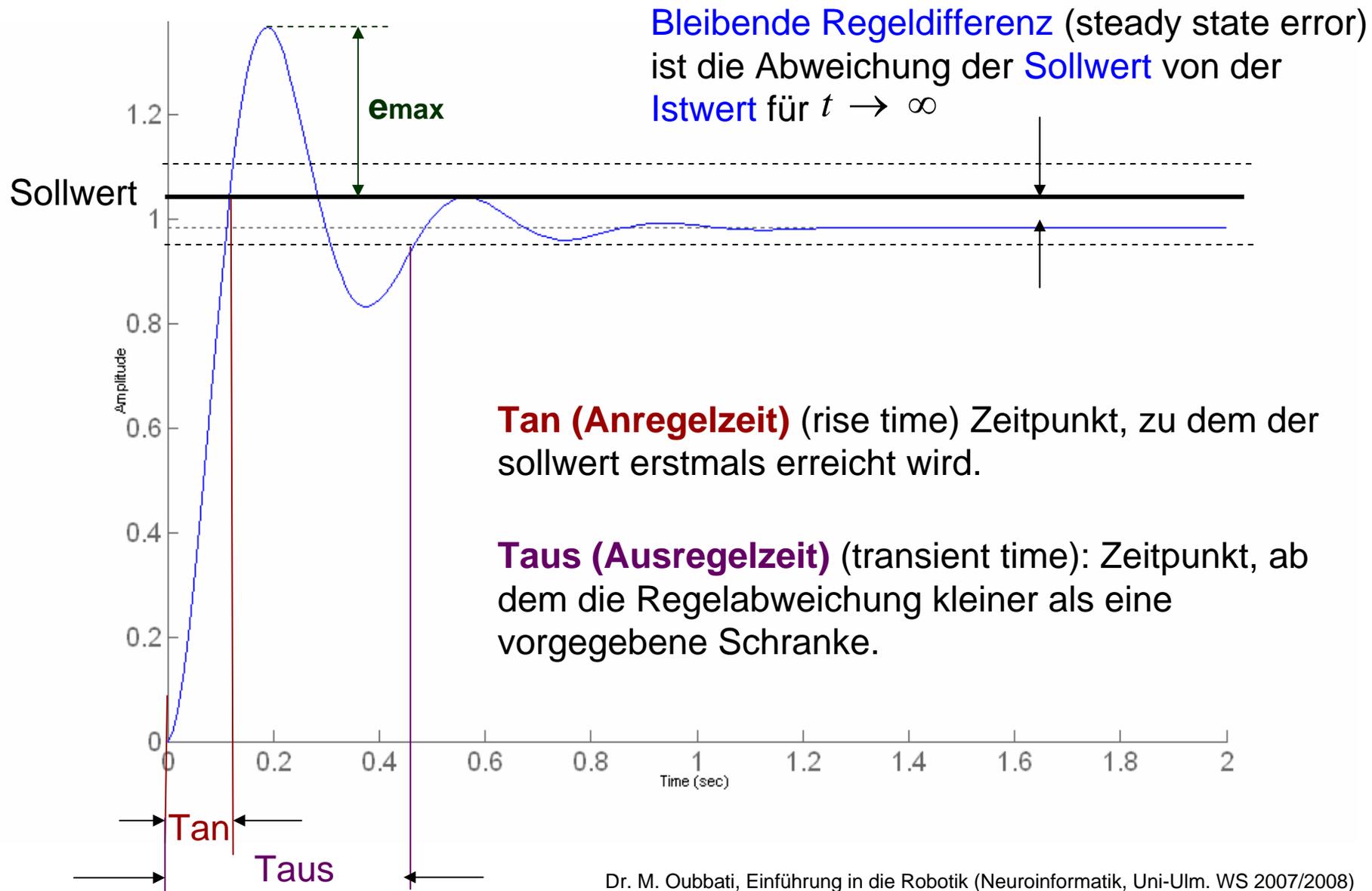
Die Regelung wirkt auch auf das Eingangssignal um das Ausgangssignal zu beeinflussen, aber in diesem Vorgehen "weiß" man, ob das Ausgangssignal die gewünschte Wert hat.



# Grundsätzlicher Aufbau von Regelkreisen



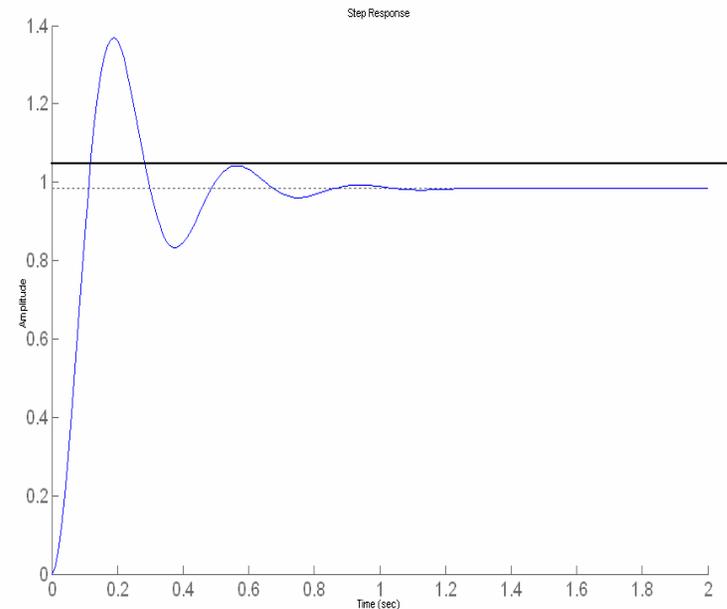
**$e_{max}$  (Maximale Überschwingweite) (overshoot)** gibt die maximale Regelabweichung an, die nach dem erstmaligen Erreichen des Sollwertes auftritt.



# Allgemeine Anforderungen an einen Regelkreis

An eine Regelung werden vier Anforderungen gestellt:

- Kleine bleibende Regeldifferenz.
- geringes Überschwingen.
- Kleine Anregelzeit.
- Kleine Ausregelzeit.

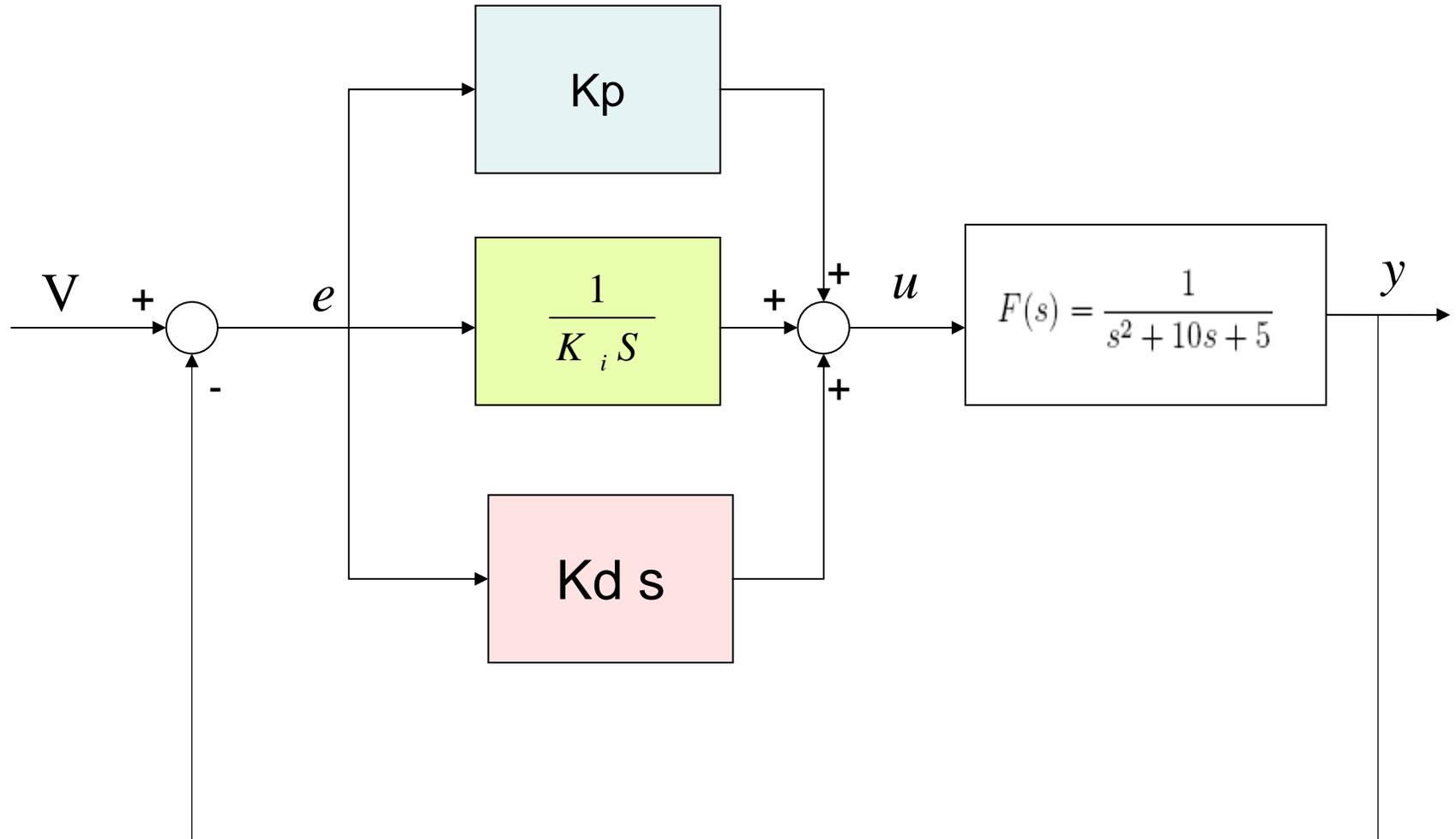


# PID-Regler

PID-Regler besteht aus folgenden Anteilen:

- Proportionalen Anteil (**P-Regler**)
- Integralen Anteil (**I-Regler**)
- Differentialen Anteil (**D-Regler**)

# PID-Regler



# *PID*-Regler

Regler	Anregelzeit	Überschwinger	Ausregelzeit	Regelabweichung
$K_p$	sinkt	wächst	geringe Änderung	sinkt
$K_I$	sinkt	wächst	wächst	0
$K_D$	geringe Änderung	sinkt	sinkt	geringe Änderung

Heutiges Thema:

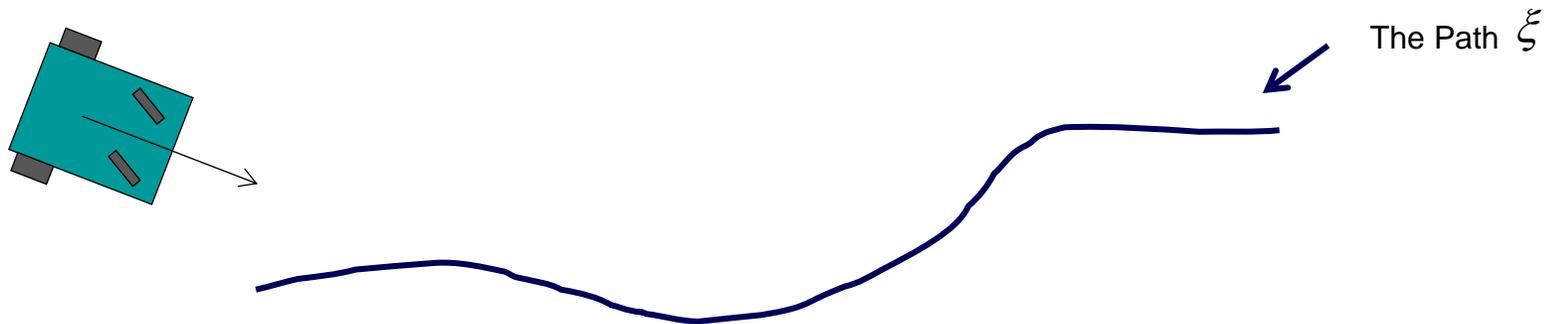
Bewegungsregelung

Motion Control has three basic problems:

➔ **Path following**

- Tracking
- Point stabilization

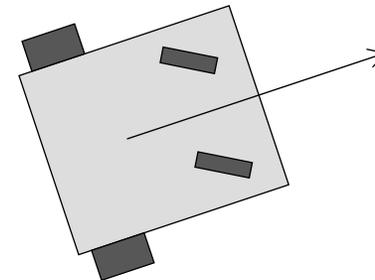
Find a smooth control input in the mobile robot, to minimise the distance between the mobile robot and the path  $\xi$



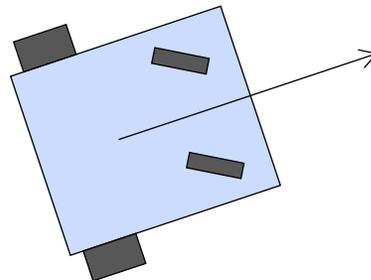
Motion Control has three basic problems:

- Path following
- ▪ **Tracking**
- Point stabilization

reference (robot,  
object,...)



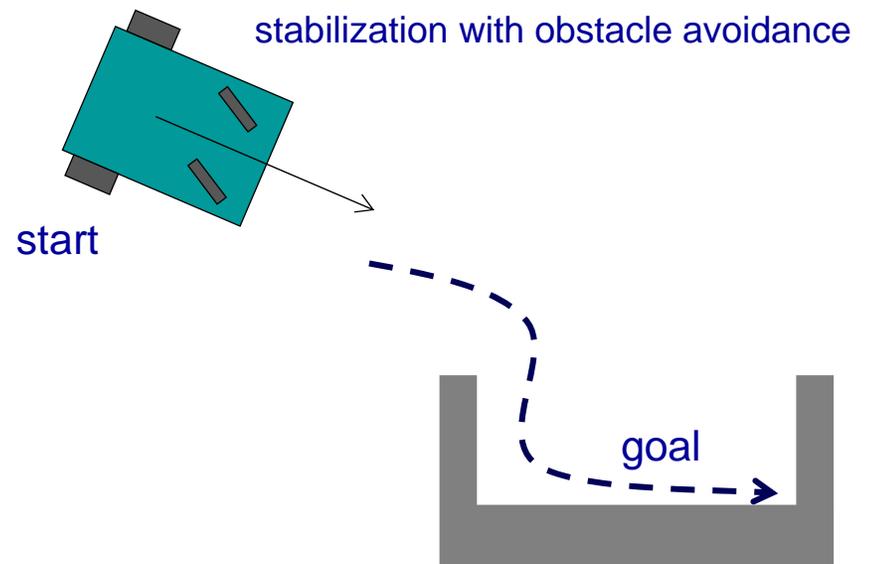
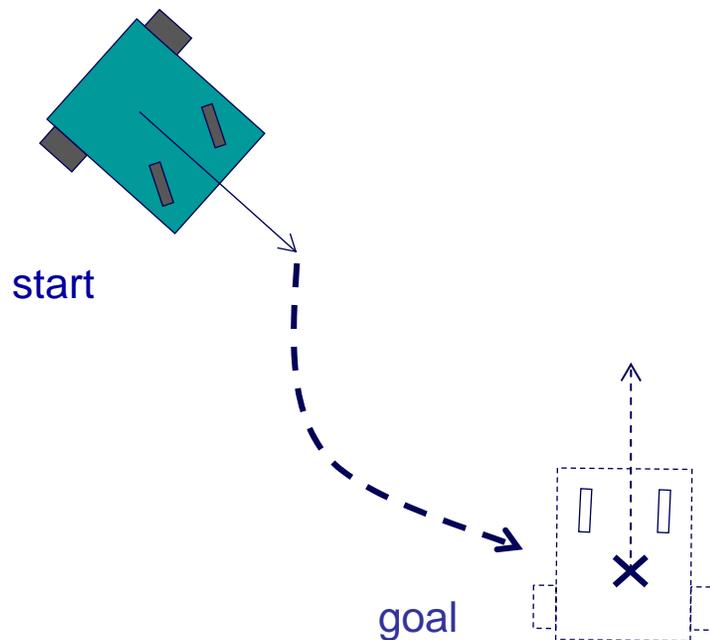
actual robot



Motion Control has three basic problems:

- Tracking
- Path following

➔ ▪ **Point stabilization**



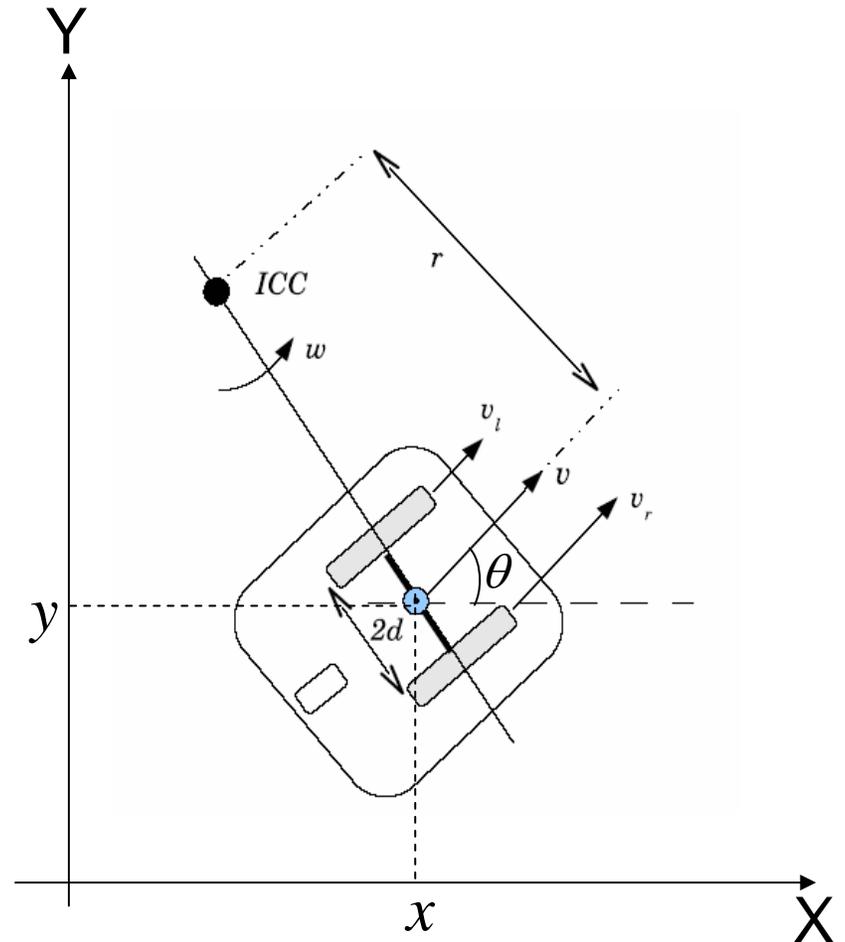
# Positionsregelung (Kinematics Control)

robot pose: **Position + Orientierung**  
des Roboters zu einem Zeitpunkt  $t$ .

## Kinematik Model

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

Das Problem zu lösen ist, welche Werte die **Translationsgeschwindigkeiten**  $v$  und die **Rotationsgeschwindigkeit**  $\omega$  annehmen müssen, damit der Roboter eine **bestimmte Pose erreichen kann**.

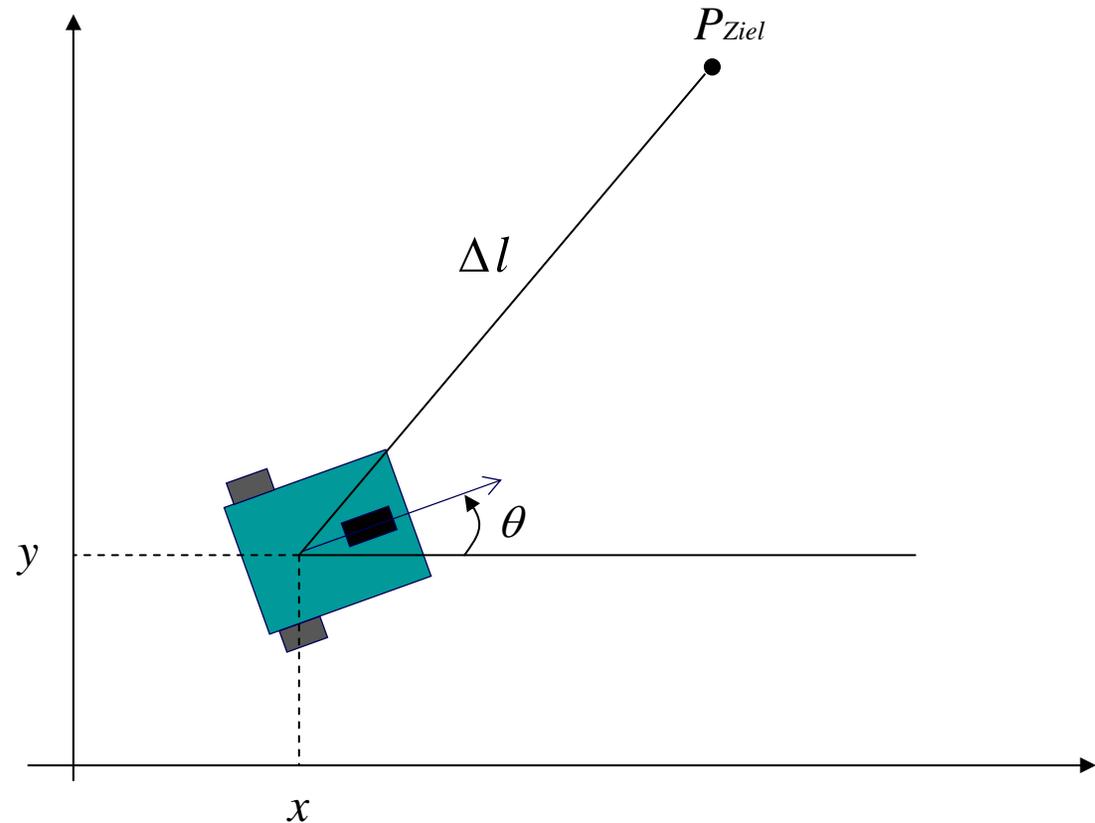


→ Dafür existiert aber **kein allgemeines Verfahren**.

## Point stabilization

## Kinematik Model

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$



**Problem:** Find a control law so that:  $\Delta l \rightarrow 0$

→ This is not trivial since the  $l$  variable do not appear in the kinematics model  
(**Nonholonomic constraint**)

# Point stabilization

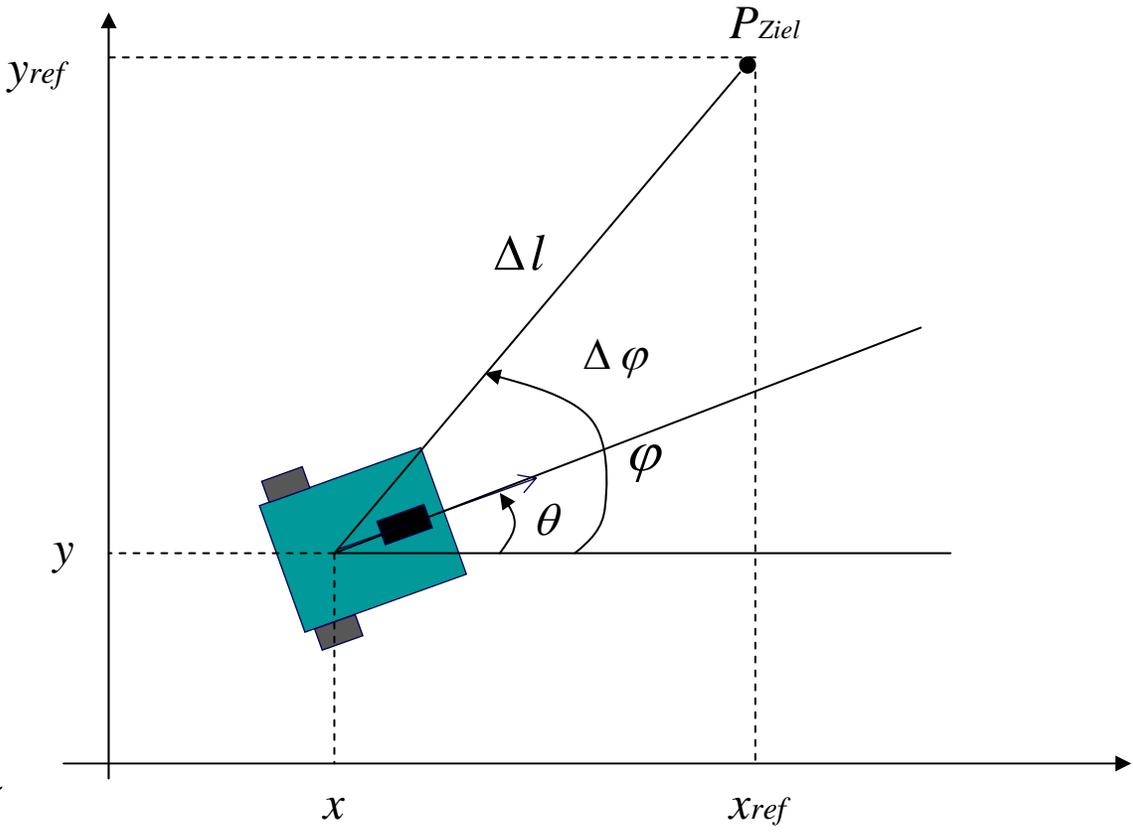
## Kinematik Model

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

To overcome this problem, we can define a new variable  $\Delta\varphi$

$$\Delta\varphi = \varphi - \theta$$

$$\Delta\varphi = \tan^{-1}\left(\frac{y_{ref} - y}{x_{ref} - x}\right) - \theta$$



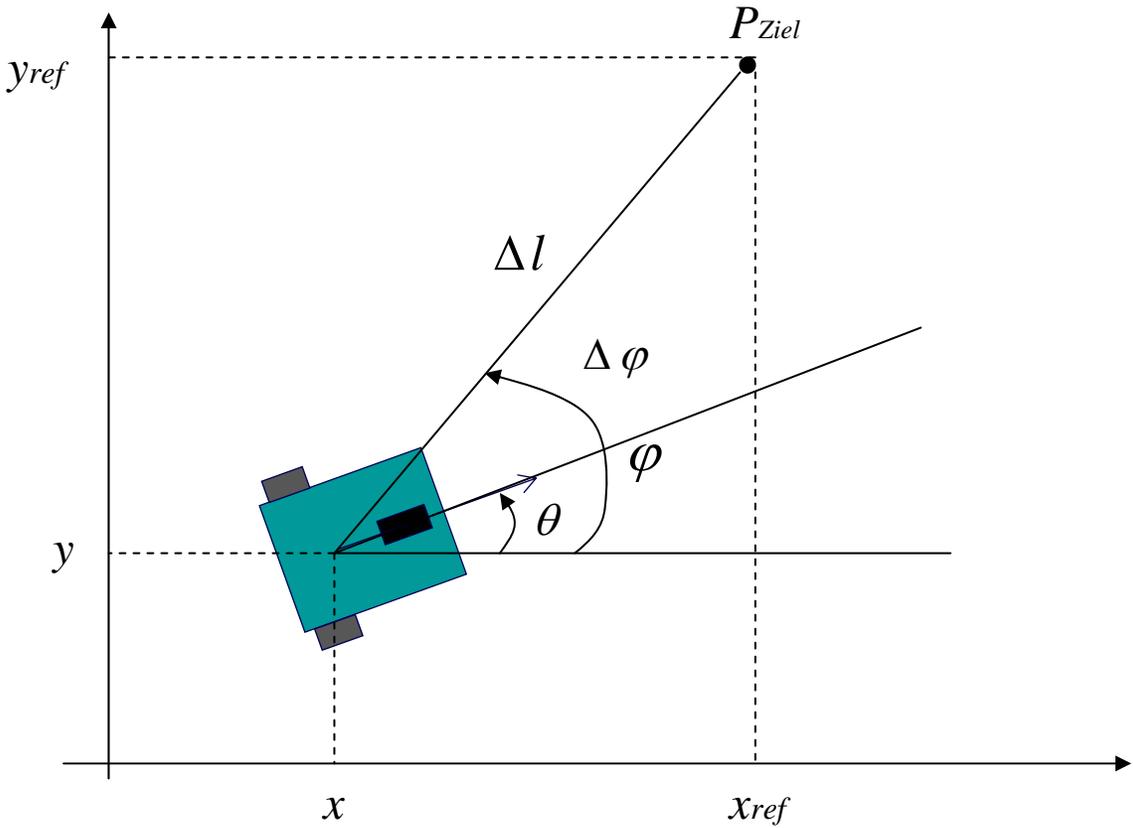
# Point stabilization

## Kinematik Model

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

Wenn  $\theta$  Zeitvariant ist, das Kinematik-Model ist Nichtlinear  $\rightarrow$  Braucht man einen nichtlinearen Kontroller um das Problem zu lösen.

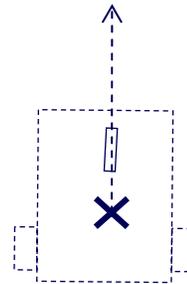
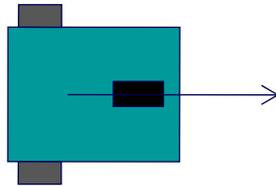
$$\Delta \varphi \rightarrow 0 \quad \Delta l \rightarrow 0$$



*Die Tafel !*

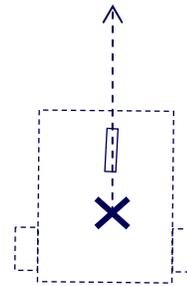
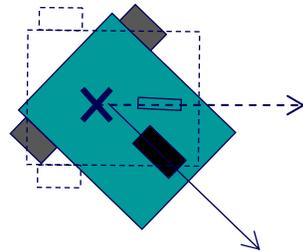
# Point stabilization

**Lösung 1:** die Position  $\Delta l$  und die Orientierung  $\Delta \varphi$  werden **getrennt** und **nicht gleichzeitig** geregelt.



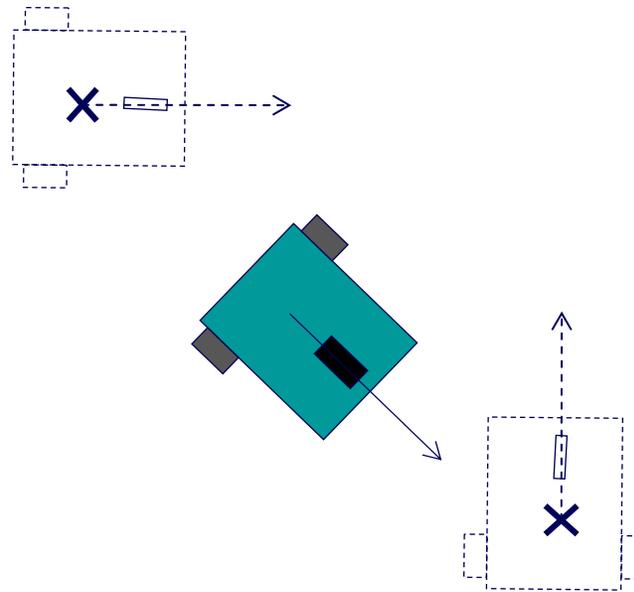
# Point stabilization

**Lösung 1:** die Position  $\Delta l$  und die Orientierung  $\Delta\varphi$  werden **getrennt** und **nicht gleichzeitig** geregelt.



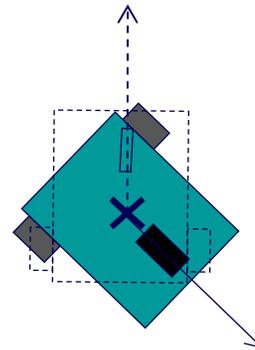
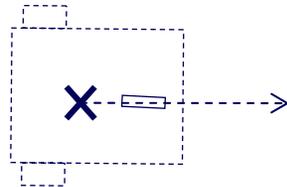
# Point stabilization

**Lösung 1:** die Position  $\Delta l$  und die Orientierung  $\Delta \varphi$  werden **getrennt** und **nicht gleichzeitig** geregelt.



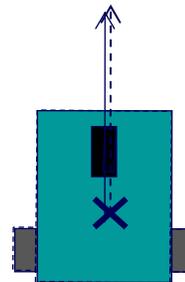
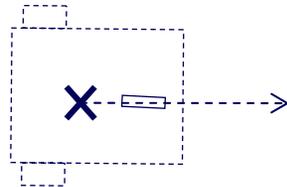
# Point stabilization

**Lösung 1:** die Position  $\Delta l$  und die Orientierung  $\Delta \varphi$  werden **getrennt** und **nicht gleichzeitig** geregelt.



# Point stabilization

**Lösung 1:** die Position  $\Delta l$  und die Orientierung  $\Delta \varphi$  werden **getrennt** und **nicht gleichzeitig** geregelt.



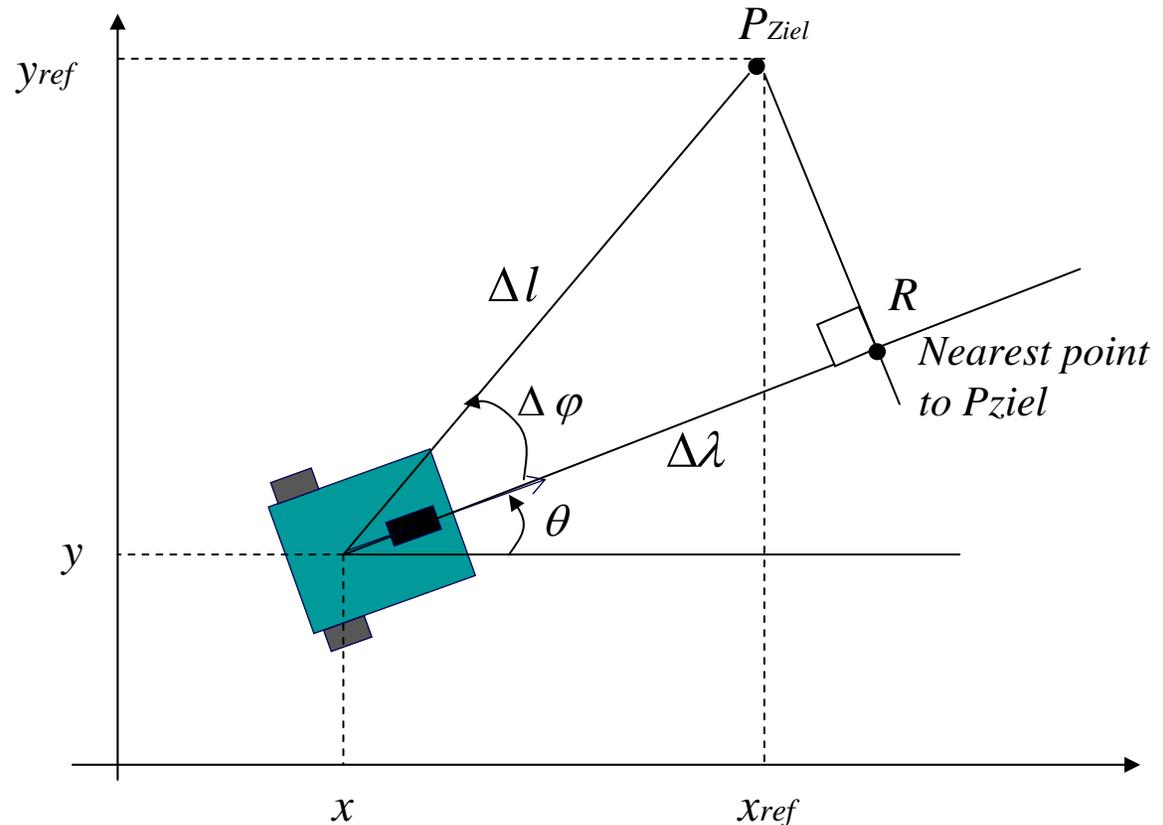
# Point stabilization

**Lösung 2:** die Position  $\Delta l$  und die Orientierung  $\Delta \varphi$  werden **getrennt** aber **gleichzeitig** geregelt.

$$\Delta l = \frac{\Delta \lambda}{\cos \Delta \varphi}$$

If  $\Delta \varphi \rightarrow 0$  and  $\Delta \lambda \rightarrow 0$

then  $\Delta l \rightarrow 0$



# Point stabilization

**Lösung 2:** die Position  $\Delta l$  und die Orientierung  $\Delta \varphi$  werden **getrennt** aber **gleichzeitig** geregelt.

$$e_{\theta} = \tan^{-1} \left( \frac{y_{ref} - y}{x_{ref} - x} \right) - \theta$$

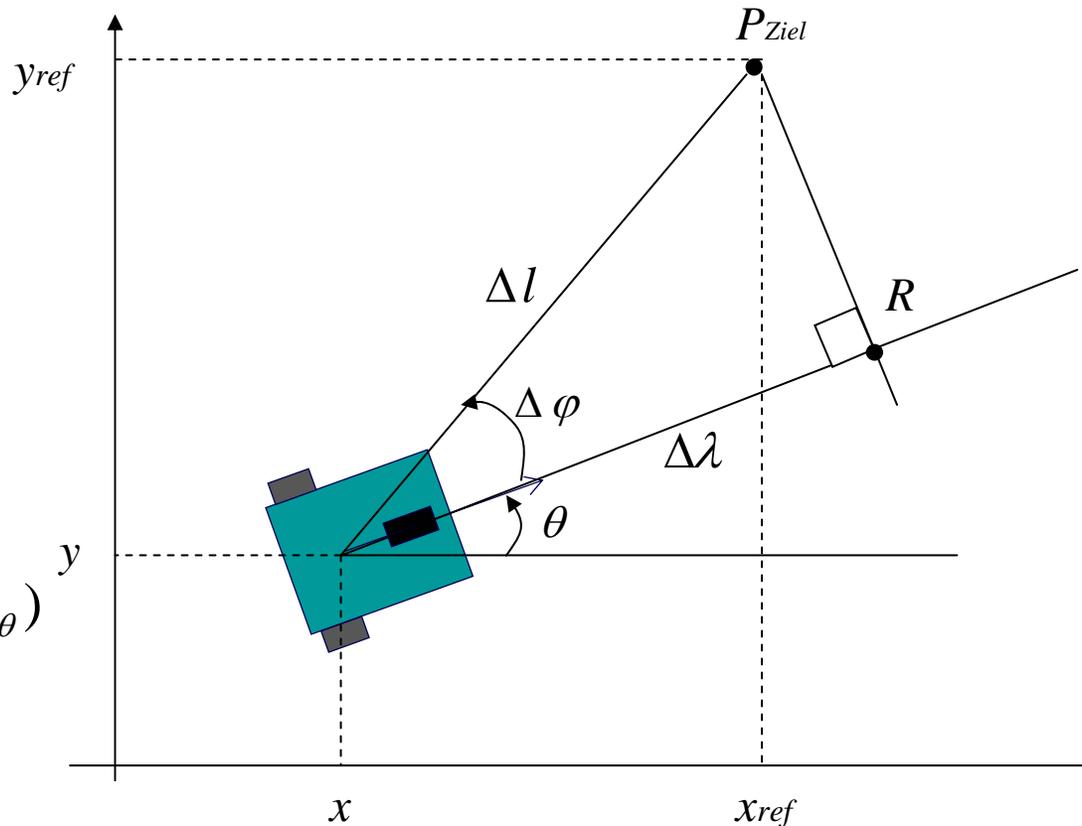
$$e_p = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2} \cos(e_{\theta})$$

$$\text{if } e_p \rightarrow 0 \quad e_{\theta} \rightarrow 0$$

$$\text{then } \Delta l \rightarrow 0$$

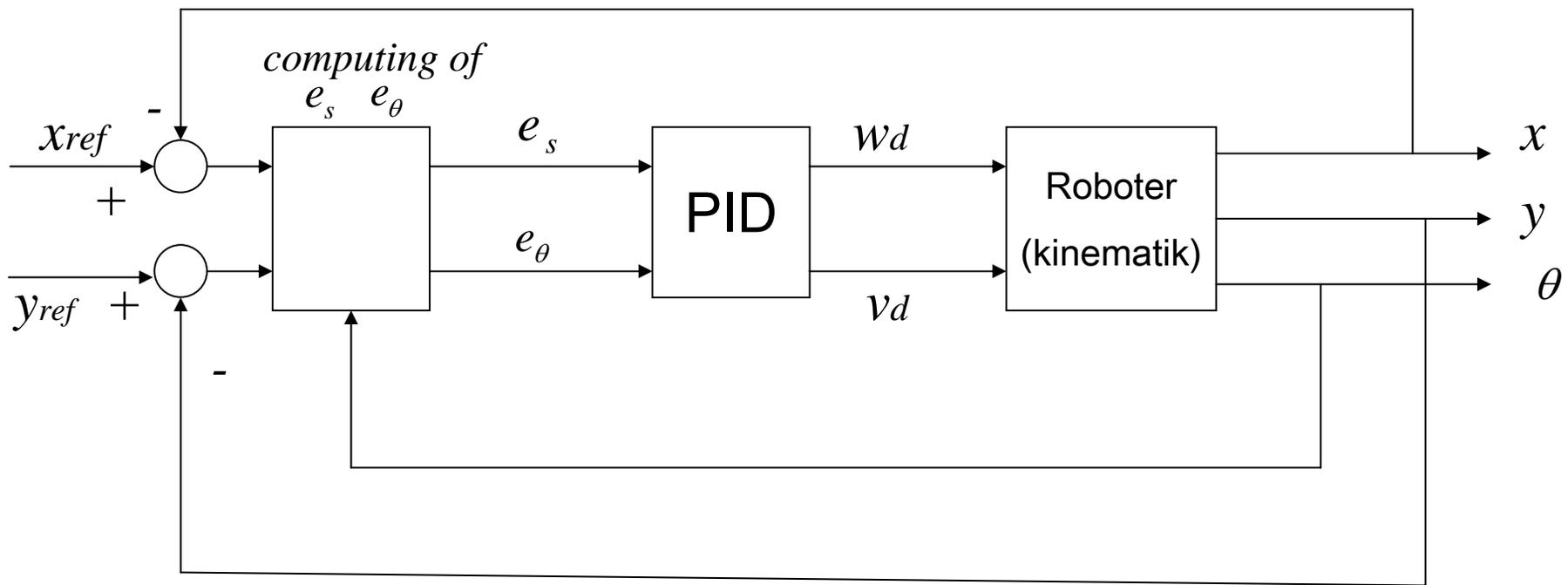
*This could be reached by  
a PID controller.*

*Die Tafel!*

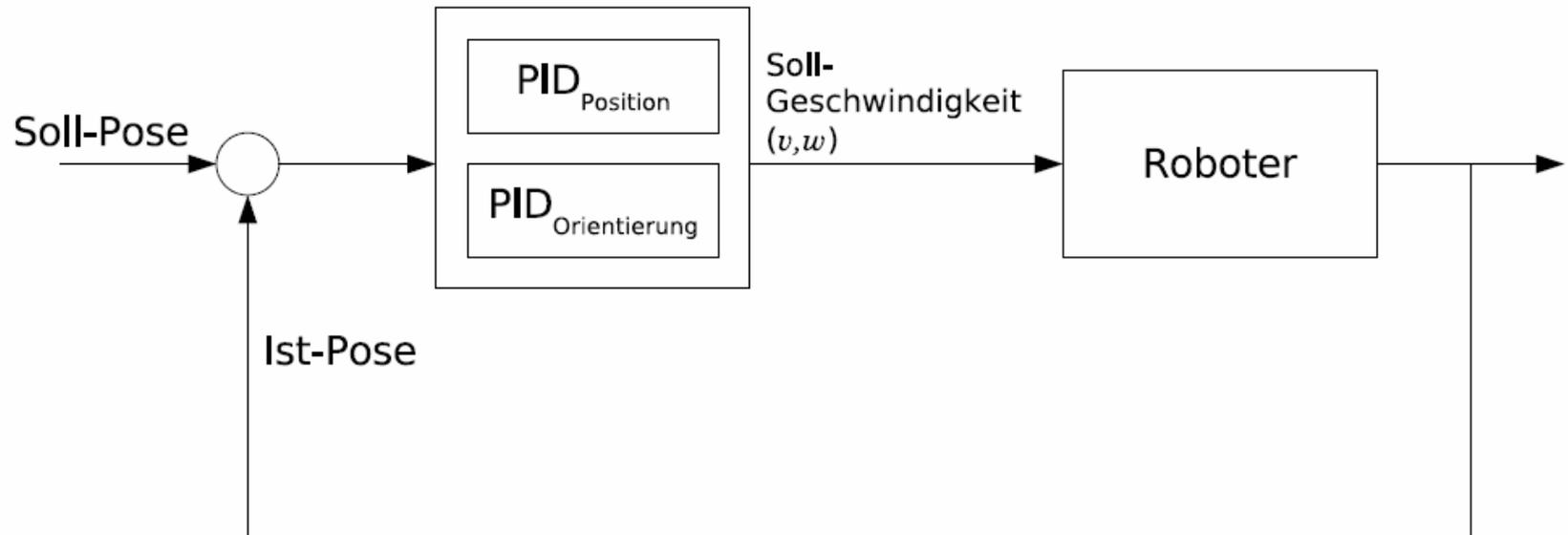


# Point stabilization

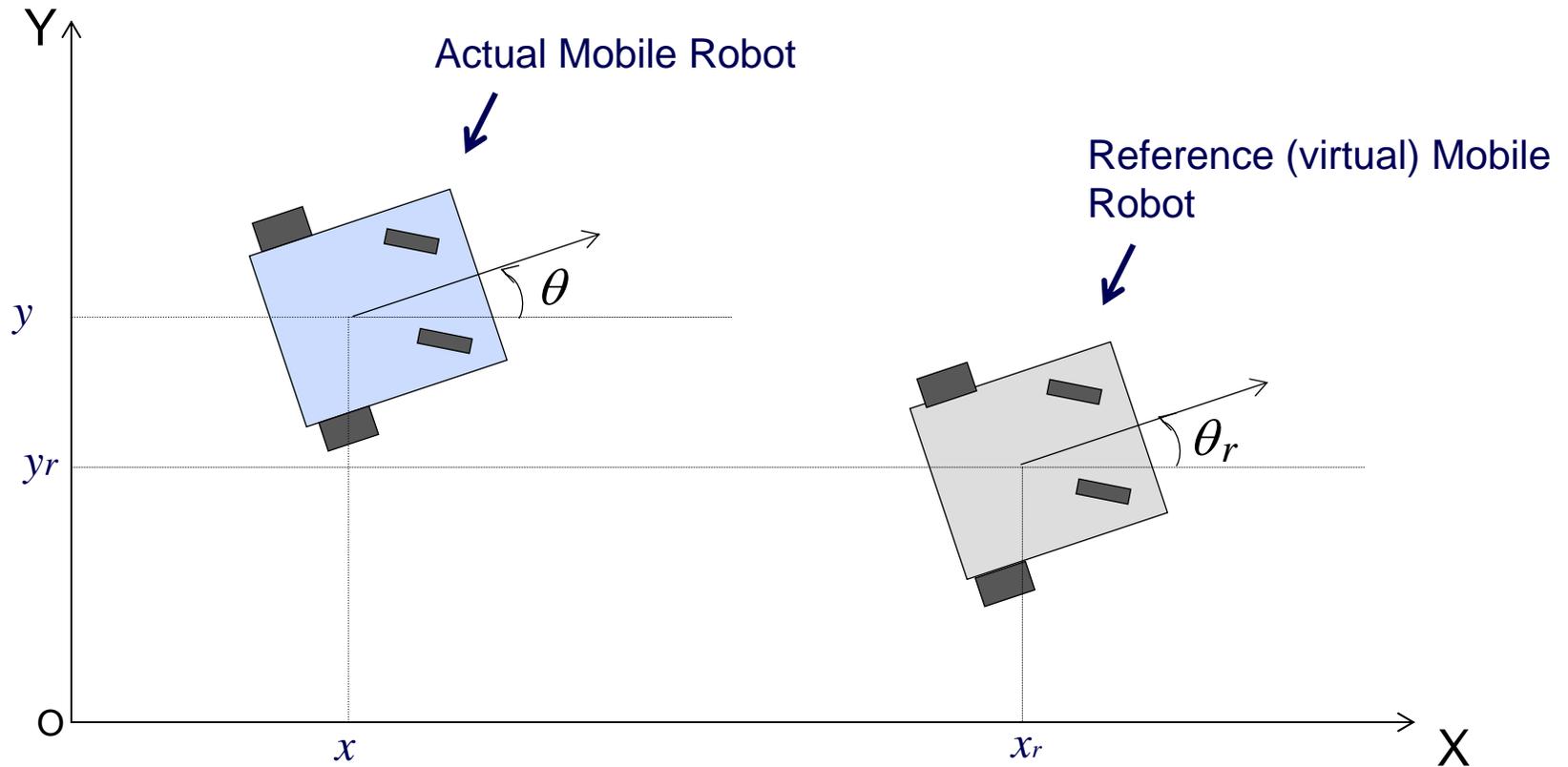
## Die gesamte Positionsregelung



## Positionierung mit PID-Regler



## Tracking



**Objective:**  $\lim_{t \rightarrow \infty} |x(t) - x_r(t)| = 0$ ,  $\lim_{t \rightarrow \infty} |y(t) - y_r(t)| = 0$  and  $\lim_{t \rightarrow \infty} |\theta(t) - \theta_r(t)| = 0$

## Tracking

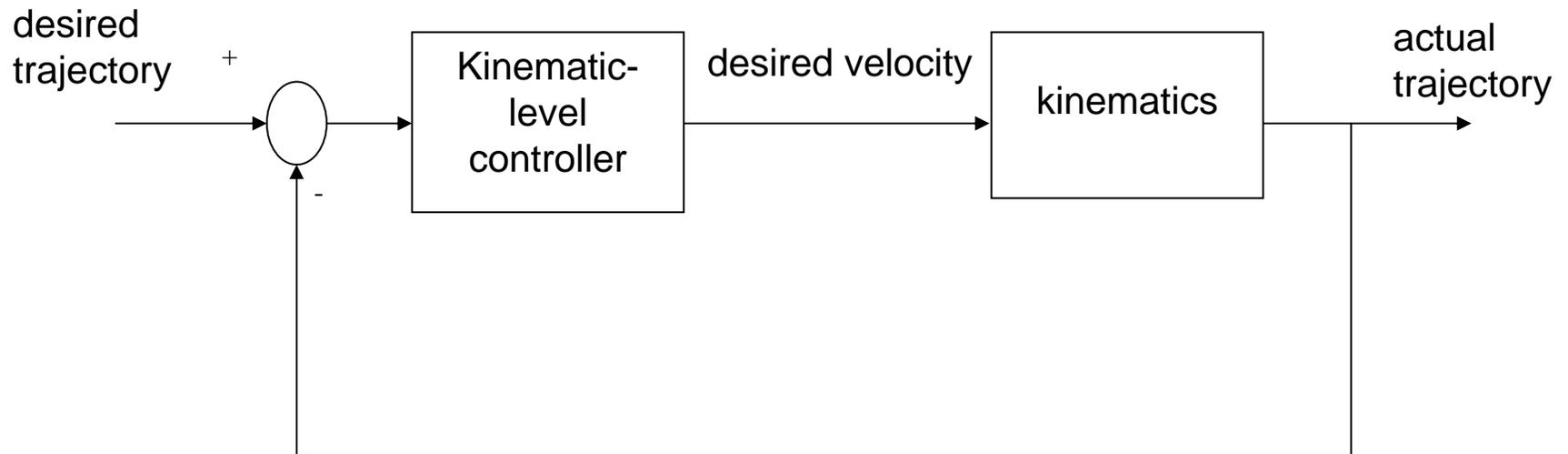
$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi}_r \end{bmatrix} = \begin{bmatrix} \cos(\phi_r) & 0 \\ \sin(\phi_r) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} \quad \text{Kinematics model of a reference robot:}$$

It is defined that

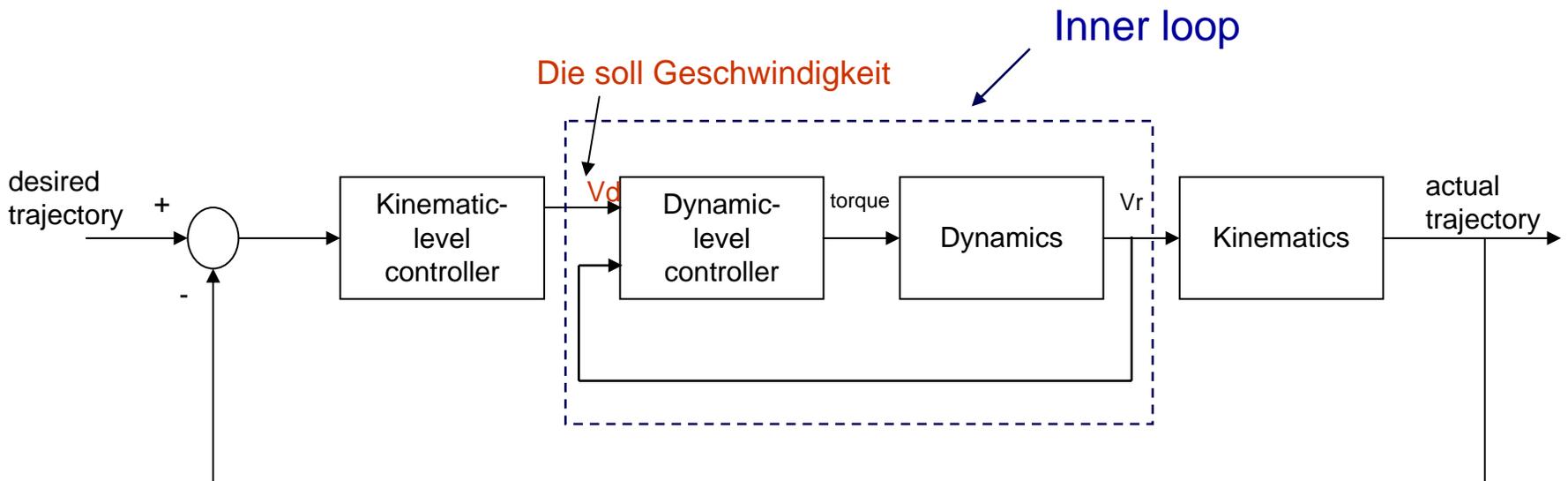
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \phi_r - \phi \end{bmatrix}$$

The desired linear and angular velocities  $(v_d, w_d)$  which make  $e_1, e_2, e_3$  converge to zero are given by

$$\begin{cases} v_d = v_r \cos e_3 + K_1 e_1 \\ w_d = w_r + v_r K_2 e_2 + K_3 \sin e_3 \end{cases} \quad \text{where } K_1, K_2, K_3 \text{ are positive constants.}$$

Control of an Ideal Mobile Robot

## Control of a Real Mobile Robot



outer loop → position control

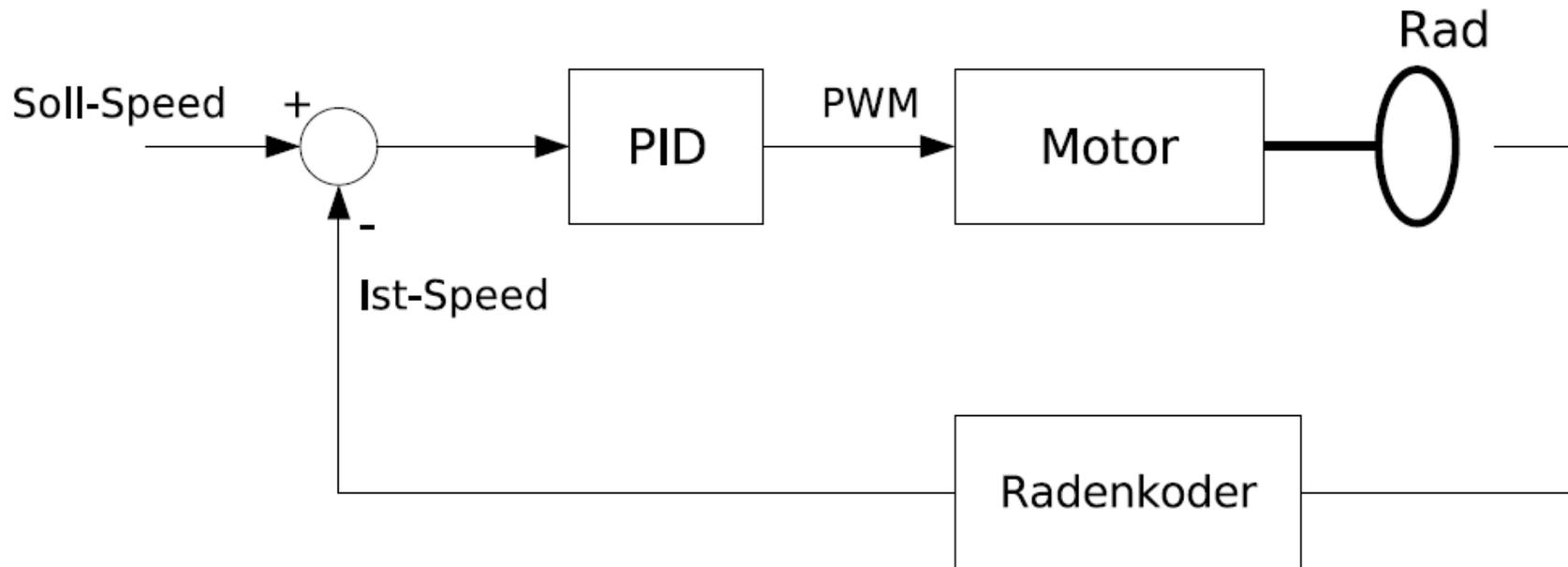
Inner loop → Velocity control (motors)

outer loop

## Geschwindigkeitsregelung (Dynamics Control)

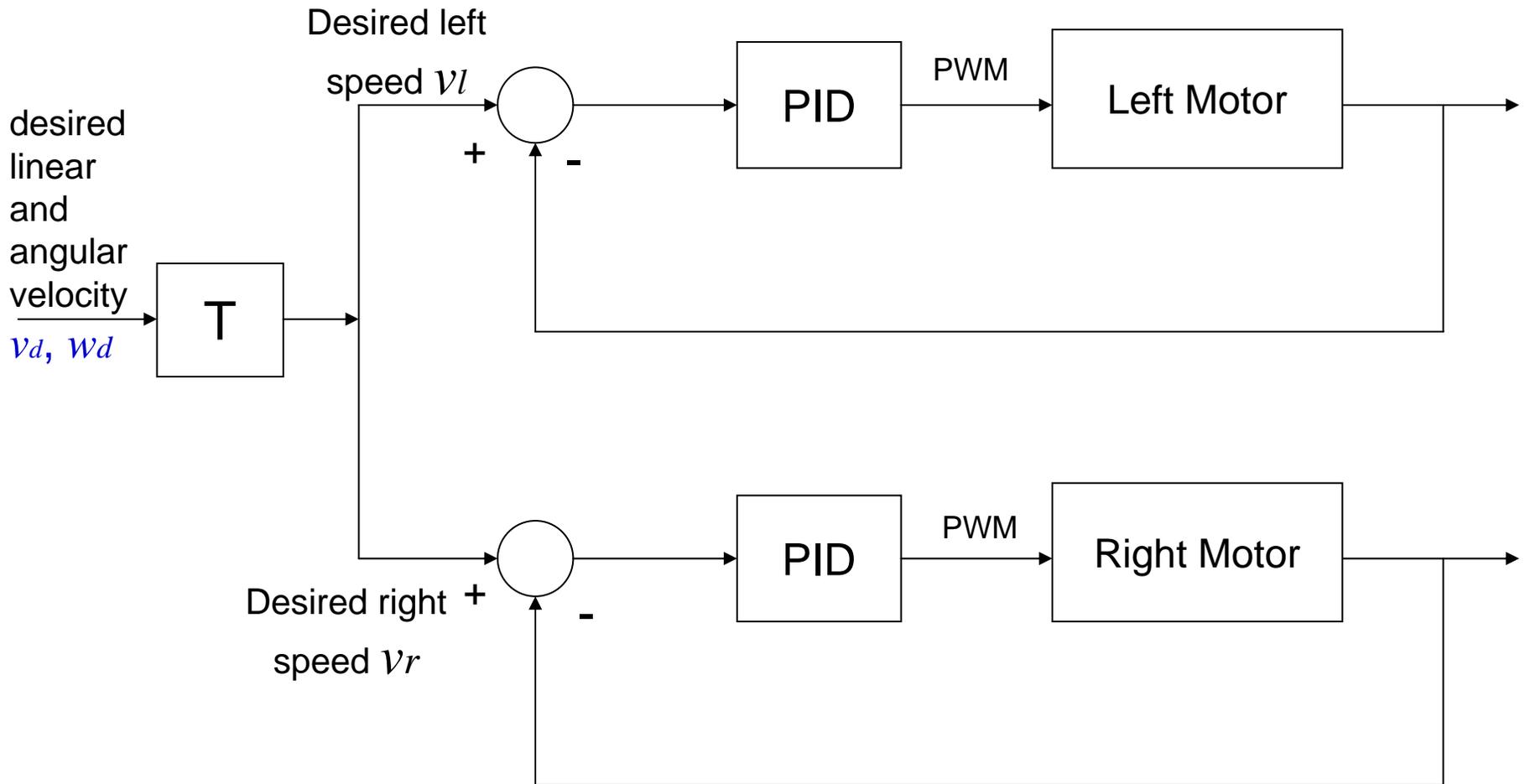
Um ein stabiles und sauberes Fahren zu ermöglichen, muss die Drehgeschwindigkeit der Motoren geregelt werden. Dies erfolgt Z.b. mit Hilfe eines PID-Reglers.

# Geschwindigkeitsregelung



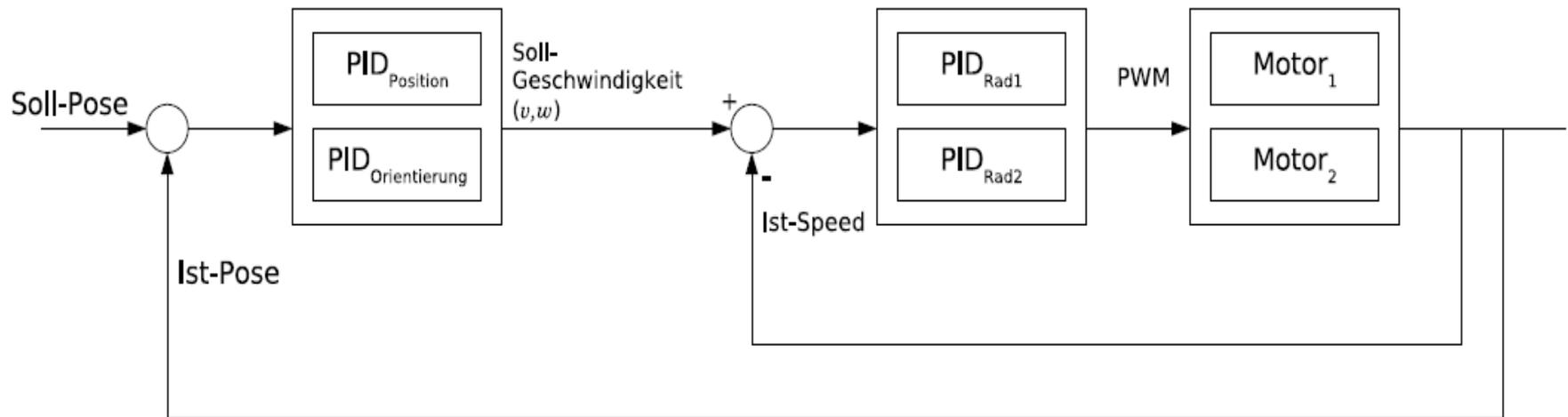
Der Radenkoder gibt dem PID-Regler Informationen über die Aktuelle Geschwindigkeit des Rads. Durch die Stellgröße, hier die Frequenz und das Tastverhältnis der PWM, wird die Regelgröße also die Rad-Geschwindigkeit verändert (geregelt).

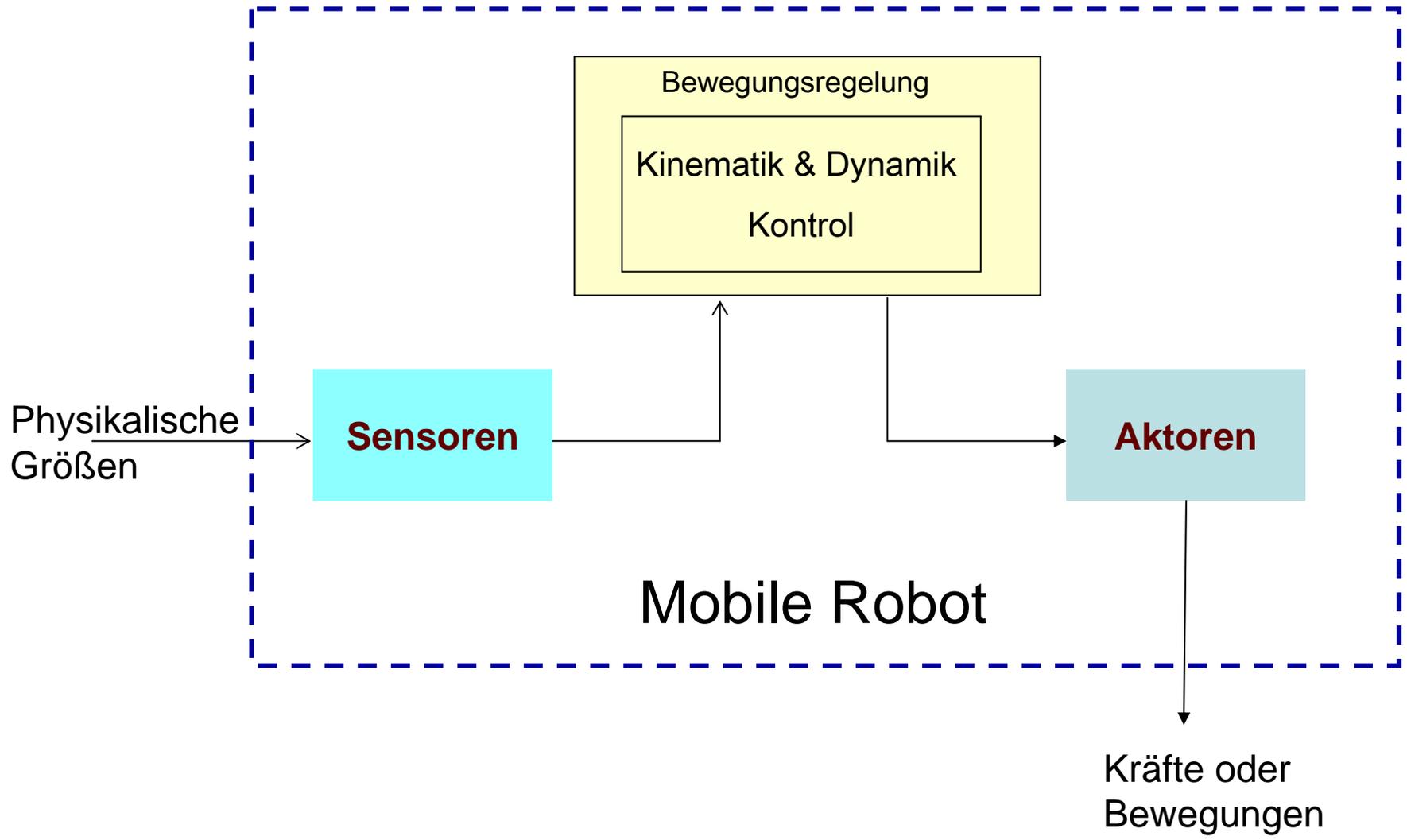
# Geschwindigkeitsregelung



T: Transformationsfunktion

# Die gesamte Bewegungsregelung des Roboters.





# Signalverarbeitungskette

