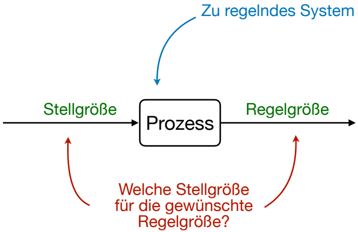
Der balancierende Roboter

# PID-Regler für einen Minseg

# Lernziele

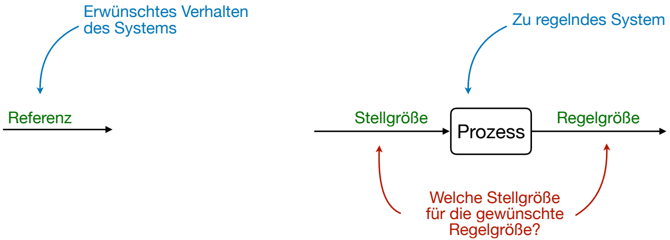
- Im Detail verstehen wie Rückkopplung funktioniert
- Verstehen was ein PID-Regler ist und wie jedes seiner Teile funktioniert

# Rückkopplung



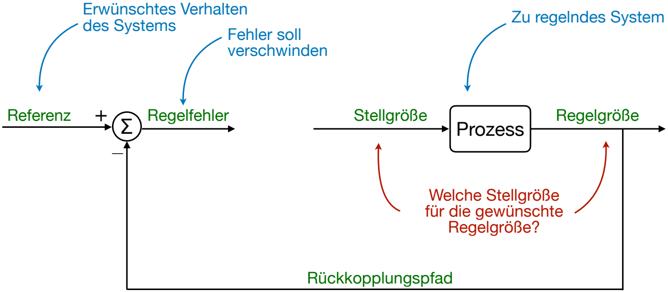

- **Regelungsaufgabe:** Wie errechnen wir die passende Stellgröße, sodass wir die gewünschte Regelgröße erhalten?

# Rückkopplung



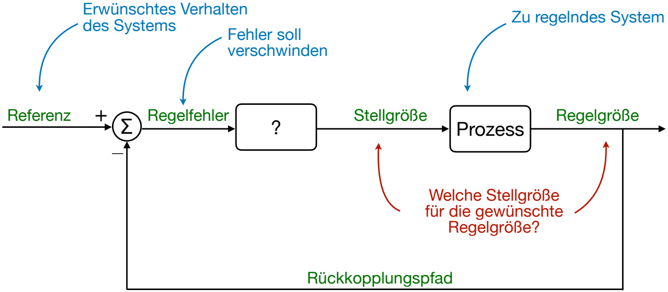

- **Regelungsaufgabe:** Wie errechnen wir die passende Stellgröße, sodass wir die gewünschte Regelgröße erhalten?
- Die Regelgröße soll der Referenz folgen.

# Rückkopplung




- **Regelungsaufgabe:** Wie errechnen wir die passende Stellgröße, sodass wir die gewünschte Regelgröße erhalten?
- Die Regelgröße soll der Referenz folgen.
- Im Rückkopplungskreis wird die Regelgröße zurückgeführt und mit der Referenzwert verglichen. Die Differenz der beiden bildet den Regelfehler.

# Rückkopplung



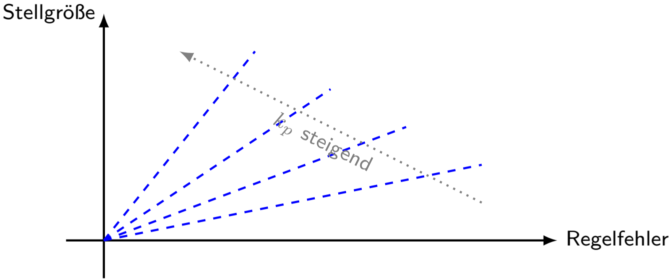

- **Regelungsaufgabe:** Wie errechnen wir die passende Stellgröße, sodass wir die gewünschte Regelgröße erhalten?
- Die Regelgröße soll der Referenz folgen.
- Im Rückkopplungskreis wird die Regelgröße zurückgeführt und mit der Referenzwert verglichen. Die Differenz der beiden bildet den Regelfehler.
- Frage: Wie kommen wir vom Regelfehler zu einer passenden Stellgröße, sodass der Regelfehler zu null wird?

# Proportional-Regler

- Intuitiver Ansatz: multipliziere den Regelfehler mit $k_p$
- Ein größerer Regelfehler führt also zu einer größeren Stellgröße und damit zu einem größeren Effekt im System.

# Proportional-Regler

- Intuitiver Ansatz: multipliziere den Regelfehler mit $k_p$
- Ein größerer Regelfehler führt also zu einer größeren Stellgröße und damit zu einem größeren Effekt im System.

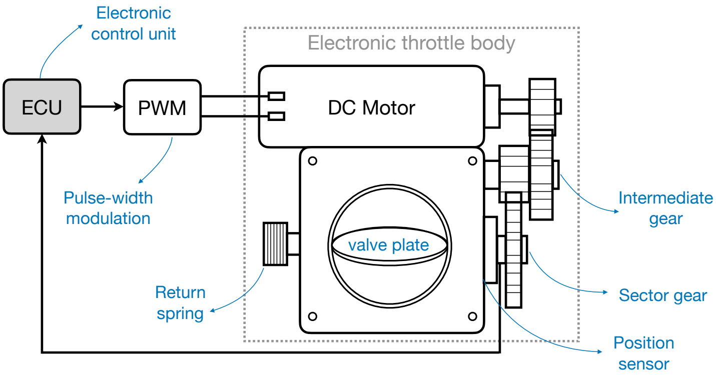


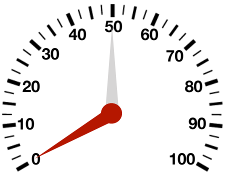

- Die Stellgröße errechnet sich durch:

$$u(t) = k_p \times e(t)$$

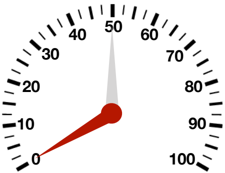
# Proportional-Regler: Beispiel

- Als Beispiel betrachten wir einen einfachen **Tempomaten**.
  - Die Drosselklappe / Ventilplatte dreht sich im Drosselklappengehäuse und öffnet den Drosseldurchgang, um mehr Luft in den Ansaugkrümmer zu lassen.

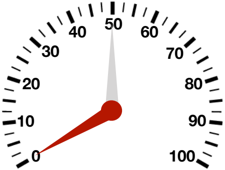




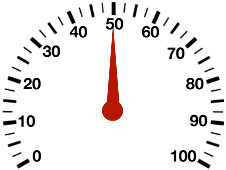

- Am Anfang ist die Geschwindigkeit gleich null. (Der Fehler ist also 50 km/h.)



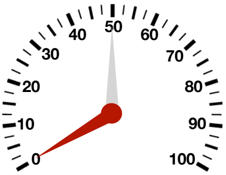

- Am Anfang ist die Geschwindigkeit gleich null. (Der Fehler ist also 50 km/h.)
- Die Drosselklappe öffnet sich und das Auto beginnt sich zu bewegen, wodurch der Fehler verringert wird.



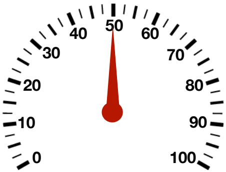

- Am Anfang ist die Geschwindigkeit gleich null. (Der Fehler ist also 50 km/h.)
- Die Drosselklappe öffnet sich und das Auto beginnt sich zu bewegen, wodurch der Fehler verringert wird.



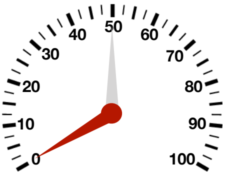

- Was passiert, wenn die Geschwindigkeit bereits 50 km/h ist?



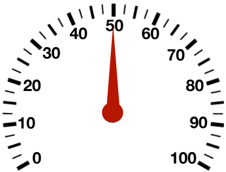

- Am Anfang ist die Geschwindigkeit gleich null. (Der Fehler ist also 50 km/h.)
- Die Drosselklappe öffnet sich und das Auto beginnt sich zu bewegen, wodurch der Fehler verringert wird.




- Was passiert, wenn die Geschwindigkeit bereits 50 km/h ist?
- Der Regelfehler wäre dann null und damit würde sich die Drosselklappe schließen, worauf das Auto an Fahrt verlieren würde.
- In diesem Fall steigt der Fehler und die Drosselklappe öffnet wieder.




- Am Anfang ist die Geschwindigkeit gleich null. (Der Fehler ist also 50 km/h.)
- Die Drosselklappe öffnet sich und das Auto beginnt sich zu bewegen, wodurch der Fehler verringert wird.




- Was passiert, wenn die Geschwindigkeit bereits 50 km/h ist?
- Der Regelfehler wäre dann null und damit würde sich die Drosselklappe schließen, worauf das Auto an Fahrt verlieren würde.
- In diesem Fall steigt der Fehler und die Drosselklappe öffnet wieder.
- Es gibt einen Drosselklappenwinkel, so dass die Geschwindigkeit genau bei 50 km/h bleibt.

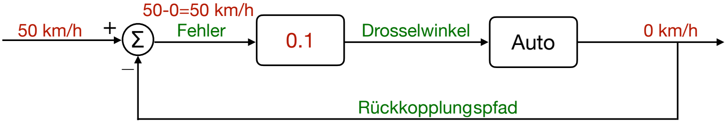
# Wie sorgt der Proportionalregler dafür, dass das Auto bei dieser Geschwindigkeit bleibt?

# Wie sorgt der Proportionalregler dafür, dass das Auto bei dieser Geschwindigkeit bleibt?

- Nehmen wir an, dass wir einen Drosselwinkel von $20^{\mathrm{o}}$ brauchen um die Geschwindigkeit zu halten.
- Der Drosselwinkel ist außerdem wie folgt gegeben:

$$\text{Drosselwinkel} = \text{Verstärkungsfaktor} \times \text{Fehler}$$

- Wenn der Verstärkungsfaktor $0.1$ ist,



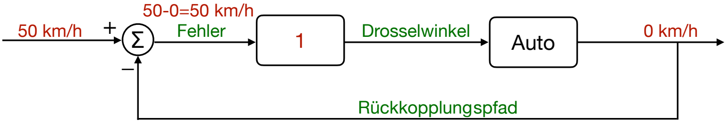

dann ist der Drosselwinkel $5^{\mathrm{o}}$, sodass der Fehler zwar reduziert wird, die Geschwindigkeit aber nicht 50 km/h erreicht.

# Wie sorgt der Proportionalregler dafür, dass das Auto bei dieser Geschwindigkeit bleibt?

- Nehmen wir an, dass wir einen Drosselwinkel von $20^{\mathrm{o}}$ brauchen um die Geschwindigkeit zu halten.
- Der Drosselwinkel ist außerdem wie folgt gegeben:

$$\text{Drosselwinkel} = \text{Verstärkungsfaktor} \times \text{Fehler}$$

- Wenn der Verstärkungsfaktor $1$ ist,



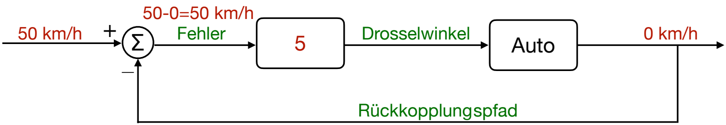

dann ist der Drosselwinkel $50^{\mathrm{o}}$ und das Auto beschleunigt. Wenn der Fehler genau 20 km/h ist (wenn wir also 30 km/h erreicht haben), dann beschleunigt das Auto nicht weiter sondern hält seine Geschwindigkeit konstant, da der Winkel $20^{\mathrm{o}}$ ist.

# Wie sorgt der Proportionalregler dafür, dass das Auto bei dieser Geschwindigkeit bleibt?

- Nehmen wir an, dass wir einen Drosselwinkel von $20^{\circ}$ brauchen um die Geschwindigkeit zu halten.
- Der Drosselwinkel ist außerdem wie folgt gegeben:

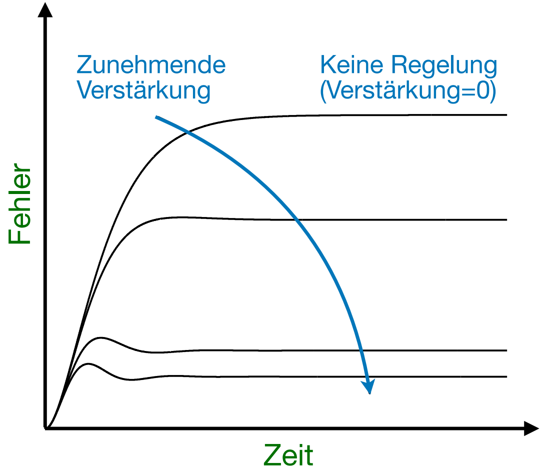
$$\text{Drosselwinkel} = \text{Verstärkungsfaktor} \times \text{Fehler}$$

- Wenn der Verstärkungsfaktor $5$ ist,




beträgt der Fehler nur 4 km/h, weil das Auto seine Beschleunigung stoppt, wenn es 46 km/h erreicht hat und damit der Drosselwinkel $20^{\circ}$ ist.

# Der Fehler verschwindet nicht. Er wird nur kleiner!

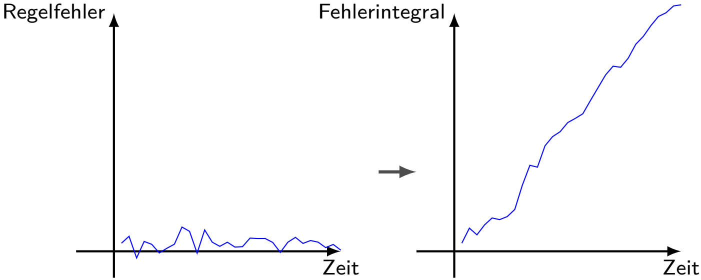



- Dies wird auch konstante Regelabweichung oder konstanter Regelfehler genannt.
- Wie können wir das verhindern?

# Proportional-Integral (PI) Regler

- Die Idee ist den Fehler der Vergangenheit zu betrachten.

- Kleine Fehler, die über einen langen Zeitraum bestehen bleiben, führen nur zu kleinen Werten für den Proportionalregler. Aber ihr Integral, also die Fläche unter der Kurve, wird über die Zeit größer:

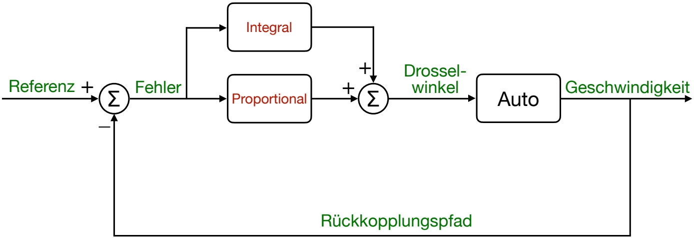



- In einem System mit konstanter Regelabweichung, wird auch dieser Fehler über die Zeit integriert, sodass das Fehlerintegral mit der Zeit ansteigt.

- Das Integral wächst erst dann nicht mehr, wenn der Fehler gleich null ist.

# Proportional-Integral (PI) Regler

- Das nutzen wir aus, sodass unser Regler nun zwei Komponenten hat:

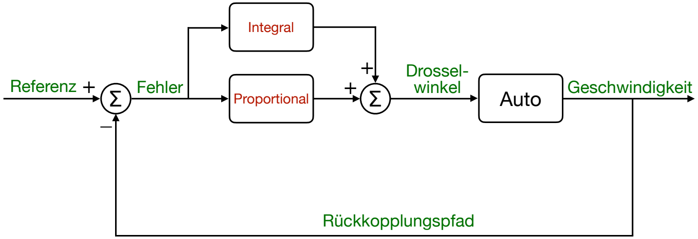



- Das Stellsignal (hier: der Drosselwinkel) wird dann zu:

$$u(t) = k_p \times e(t) + k_i \times \int_0^t e(t)dt$$

# Proportional-Integral (PI) Regler

- Das nutzen wir aus, sodass unser Regler nun zwei Komponenten hat:



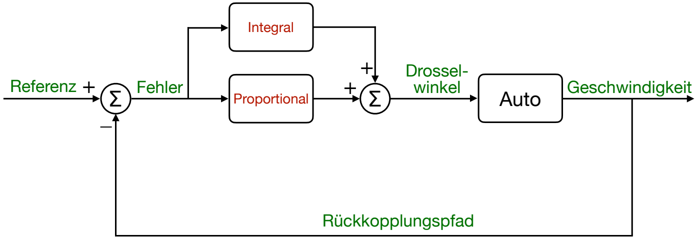

- Das Stellsignal (hier: der Drosselwinkel) wird dann zu:

$$u(t) = k_p \times e(t) + k_i \times \int_0^t e(t)dt$$

- Wenn das System einen stationären Fehler aufweist, wird dieser Fehler über die Zeit integriert, wodurch der Integralterm und damit der Drosselklappenwinkel erhöht werden.

# Proportional-Integral (PI) Regler

- Das nutzen wir aus, sodass unser Regler nun zwei Komponenten hat:



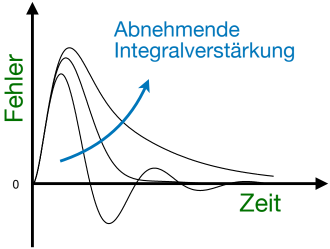

- Das Stellsignal (hier: der Drosselwinkel) wird dann zu:

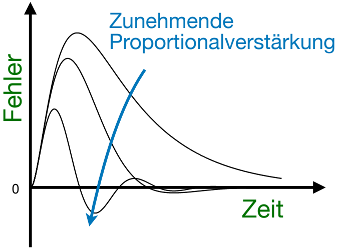
$$u(t) = k_p \times e(t) + k_i \times \int_0^t e(t)dt$$

- Wenn das System einen stationären Fehler aufweist, wird dieser Fehler über die Zeit integriert, wodurch der Integralterm und damit der Drosselklappenwinkel erhöht werden.
- Das Integral wächst nicht nur für große Fehler sondern auch durch langanhaltende kleine Fehler. Damit können konstante Regelfehler vermieden werden.

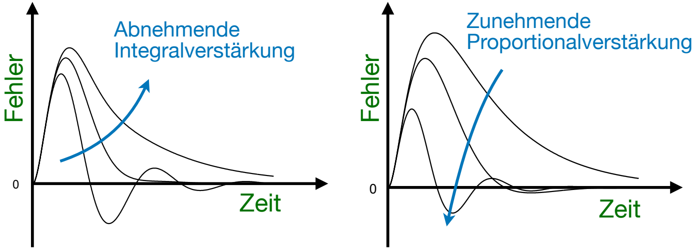
- Vorteil: Wir erhalten keinen konstanten Regelfehler.

- Vorteil: Wir erhalten keinen konstanten Regelfehler.
- Nachteil: Für kleine Werte von $k_i$ reagiert der Regler langsam (weil das Integral mit der Zeit wächst), und wenn $k_i$ groß ist, kann das System überschwingen, da der Integralteil nur dann gleich bleibt, wenn der Fehler null ist:



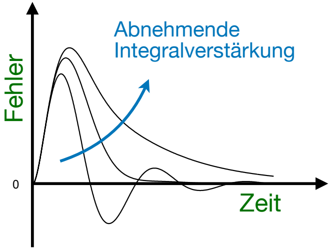



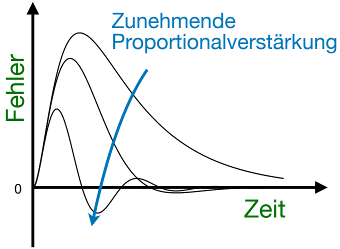
- Vorteil: Wir erhalten keinen konstanten Regelfehler.
- Nachteil: Für kleine Werte von $k_i$ reagiert der Regler langsam (weil das Integral mit der Zeit wächst), und wenn $k_i$ groß ist, kann das System überschwingen, da der Integralteil nur dann gleich bleibt, wenn der Fehler null ist:




- Dieses Überschwingen wollen wir vermeiden!

- Vorteil: Wir erhalten keinen konstanten Regelfehler.
- Nachteil: Für kleine Werte von $k_i$ reagiert der Regler langsam (weil das Integral mit der Zeit wächst), und wenn $k_i$ groß ist, kann das System überschwingen, da der Integralteil nur dann gleich bleibt, wenn der Fehler null ist:



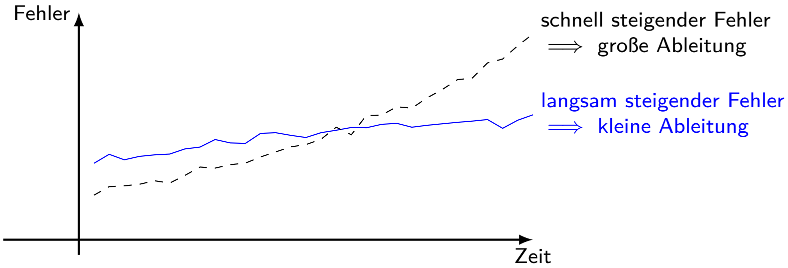

- Dieses Überschwingen wollen wir vermeiden!
- Aber wie?

# Proportional-Integral-Derivative (PID)-Regler

- Nun wollen wir auch in Betracht ziehen, wie der Fehler in der Zukunft sein wird.

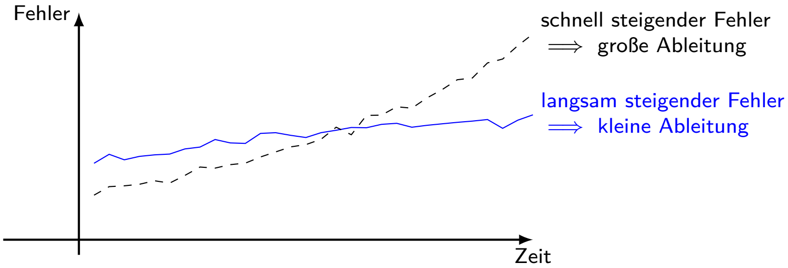
# Proportional-Integral-Derivative (PID)-Regler

- Nun wollen wir auch in Betracht ziehen, wie der Fehler in der Zukunft sein wird.
- Dazu betrachten wir wie schnell sich der Fehler ändert:



# Proportional-Integral-Derivative (PID)-Regler

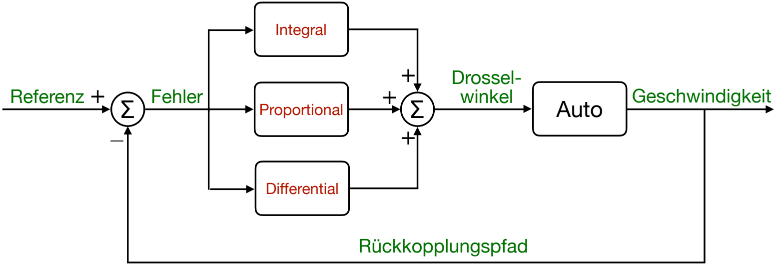
- Nun wollen wir auch in Betracht ziehen, wie der Fehler in der Zukunft sein wird.
- Dazu betrachten wir wie schnell sich der Fehler ändert:




- Der Ableitungsteil
  - betrachtet die aktuelle Änderungsrate des Fehlers (je schneller der Fehler wächst, desto größer wird die Ableitung),
  - ermittelt, wie schnell wohl der gewünschte Wert erreicht werden wird, und
  - reduziert die Stellgröße rechtzeitig.

# Proportional-Integral-Derivative (PID)-Regler

- Der Regler hat nun drei Teile:

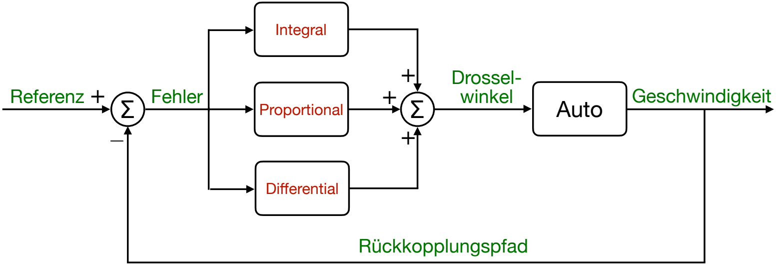



- Das Stellsignal ist dann:

$$u(t) = k_p \times e(t) + k_i \times \int_0^t e(t)dt + k_d \times \frac{de(t)}{dt}$$

# Proportional-Integral-Derivative (PID)-Regler

- Der Regler hat nun drei Teile:




- Das Stellsignal ist dann:

$$u(t) = k_p \times e(t) + k_i \times \int_0^t e(t)dt + k_d \times \frac{de(t)}{dt}$$

**Wie können wir diese Konzepte bei der Regelung des Minsegs anwenden?**

# PID-Regler für den Minseg-Roboter

Schauen Sie in die Bibliothek "SEG_CONTROL"!
Wie ist die Gleichung für die Stellgröße implementiert?

$$u(t) = k_p \times e(t) + k_i \times \int_0^t e(t)dt + k_d \times \frac{de(t)}{dt}$$

# PID-Regler für den Minseg-Roboter

Schauen Sie in die Bibliothek "SEG_CONTROL"!
Wie ist die Gleichung für die Stellgröße implementiert?

$$u(t) = k_p \times e(t) + k_i \times \int_0^t e(t)dt + k_d \times \frac{de(t)}{dt}$$

Proportionalteil:

```
Vout1 = Kp * error + Kd * dTerm + integralTerm;
```

# PID-Regler für den Minseg-Roboter

Schauen Sie in die Bibliothek “SEG_CONTROL”!
Wie ist die Gleichung für die Stellgröße implementiert?

$$u(t) = k_p \times e(t) + k_i \times \int_0^t e(t)dt + k_d \times \frac{de(t)}{dt}$$

Proportionalteil:

```
Vout1 = Kp * error + Kd * dTerm + integralTerm;
```

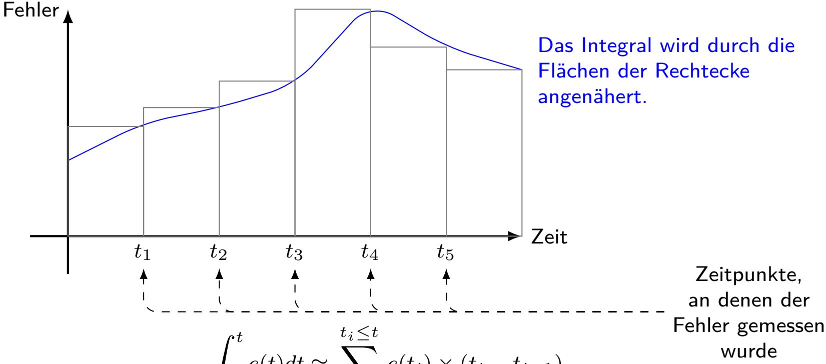
Wie werden der Integral- und Ableitungsteil berechnet?

# Integral (I) Regler für den Minseg-Roboter

- Wir müssen das Integral numerisch berechnen.
- Wir kennen den Fehler nicht immer, sondern nur für einige Momente, in denen wir ihn gemessen haben.

# Integral (I) Regler für den Minseg-Roboter

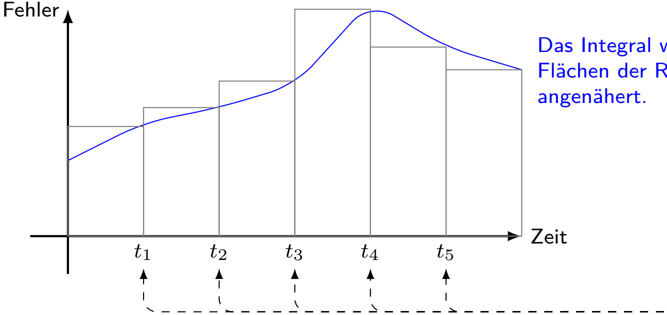
- Wir müssen das Integral numerisch berechnen.
- Wir kennen den Fehler nicht immer, sondern nur für einige Momente, in denen wir ihn gemessen haben.




$$\int_0^t e(t)dt \approx \sum_{t_i=t_1}^{t_i \leq t} e(t_i) \times (t_i - t_{i-1})$$

# Integral (I) Regler für den Minseg-Roboter

- Wir müssen das Integral numerisch berechnen.
- Wir kennen den Fehler nicht immer, sondern nur für einige Momente, in denen wir ihn gemessen haben.




$$\int_0^t e(t)dt \approx \sum_{t_i=t_1}^{t_i \leq t} e(t_i) \times (t_i - t_{i-1})$$

- Der Code berechnet (für den Verstärkunsfaktor $k_i$):

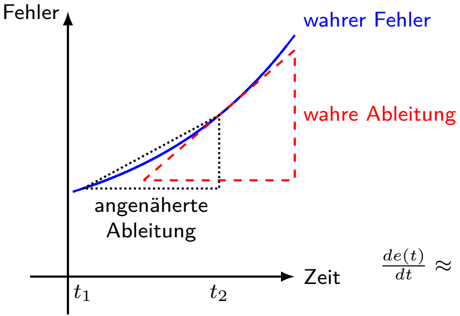
```
integralTerm += error * actualDt * Ki;
```

# Differential (D) Regler für den Minseg-Roboter

- Wir müssen die Ableitung numerisch berechnen.
- Wir kennen den Fehler nicht immer, sondern nur für einige Momente, in denen wir ihn gemessen haben.

# Differential (D) Regler für den Minseg-Roboter

- Wir müssen die Ableitung numerisch berechnen.
- Wir kennen den Fehler nicht immer, sondern nur für einige Momente, in denen wir ihn gemessen haben.
- Dies führt zu einer Annäherung der Ableitung als Steigung im Dreieck:

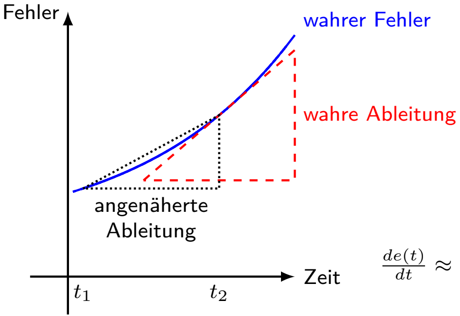



$$\frac{de(t)}{dt} \approx \frac{e(t_i) - e(t_{i-1})}{t_i - t_{i-1}}$$

Die Steigung ergibt sich as der Höhe $e(t_i) - e(t_{i-1})$ und Breite $t_i - t_{i-1}$ des Dreiecks.

# Differential (D) Regler für den Minseg-Roboter

- Wir müssen die Ableitung numerisch berechnen.
- Wir kennen den Fehler nicht immer, sondern nur für einige Momente, in denen wir ihn gemessen haben.
- Dies führt zu einer Annäherung der Ableitung als Steigung im Dreieck:




$$\frac{de(t)}{dt} \approx \frac{e(t_i)-e(t_{i-1})}{t_i-t_{i-1}}$$

Die Steigung ergibt sich as der Höhe $e(t_i) - e(t_{i-1})$ und Breite $t_i - t_{i-1}$ des Dreiecks.

- Damit ergibt sich im Code:

```
dTerm = (error - lastErr) / actualDt;
```