

Python Übung Blatt II - Simulation und Stabilität eines Traktors mit Anhänger

Prof. Dr. Moritz Diehl, Jochem De Schutter, Peter Hofmeier, Fabien Jenne

Auf diesem Blatt soll ein Traktor (schwarz) mit Anhänger (grün) sowohl beim Vorwärts- als beim Rückwärtsfahren simuliert werden sowie auf Stabilität geprüft werden.

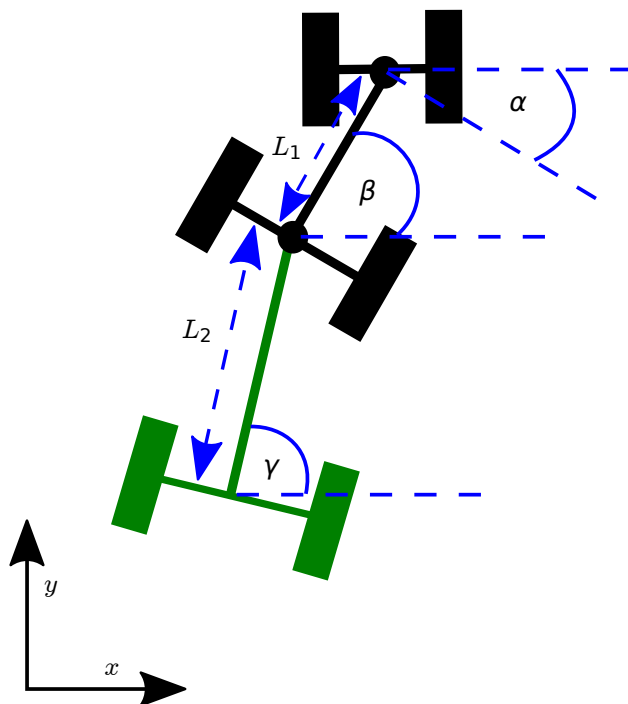


Abbildung 1: Skizze des Traktors mit Anhänger

In dieser Aufgabe betrachten wir hierzu ein reduziertes Modell des Systems, das nur die Orientierung des Traktors und des Anhängers beschreibt, aber nicht deren Translation.

$$x(t) = \begin{bmatrix} \beta(t) \\ \gamma(t) \end{bmatrix}, \quad u(t) = \alpha(t), \quad y(t) = [\gamma(t)]$$
$$\dot{x} = f(x, u) = \begin{bmatrix} \frac{V}{L_1} \tan(\alpha) \\ \frac{V}{L_2} \cos(\beta - \gamma - \frac{\pi}{2}) \end{bmatrix}$$

Dabei ist $L_1 = 6$ m und $L_2 = 4$ m. Die Geschwindigkeit ist $V = 5$ m/s beim Vorwärtsfahren und $V = -5$ m/s beim Rückwärtsfahren.

Das `python-control`-Paket stellt Funktionen zum Entwurf sowie zur Analyse linearer Steuerungs- und Regelungssysteme zur Verfügung. In dieser Aufgabe untersuchen Sie die Stabilität des Traktor-Anhänger Systems.

1. Linearisieren Sie das System zunächst um den Gleichgewichtspunkt $x_{ss} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $u_{ss} = [0]$. Berechnen Sie die Matrizen A , B , C , D des linearisierten Zustandsraummodells.
2. Erstellen Sie nun mithilfe des `python-control`-Pakets das Zustandsraummodell. Nutzen Sie hierzu den Befehl `ss(...)`.
3. Überprüfen Sie das erhaltene System auf Stabilität.
Hinweis: Berechnen Sie zunächst die Polstellen des Systems. (`pole(...)`).
4. Wiederholen Sie die Schritte (1) - (3) für den rückwärts fahrenden Traktor aus Aufgabe 1.

5. Simulieren Sie über einen Zeitraum von 10 s die Systeme für Vorwärts- und Rückwärtsfahrt mithilfe des `python-control`-Pakets. Nutzen Sie den Eingang $u(t) = 0.3$ rad und verwenden Sie einen Zeitschritt von $\Delta t = 0.01$ s.
Hinweis: `forced_response(...)`.
6. Plotten Sie die Zustände $\beta(t)$ und $\gamma(t)$ der simulierten Systeme. Erscheinen die Ergebnisse sinnvoll?
7. Simulieren Sie nun das nichtlineare System über den gleichen Zeitraum und mit gleichem Steuerungsinput. Nutzen Sie zur Simulation die Funktion `nlsim()`, die bereits von Übungsblatt 3 bekannt ist. `nlsim()` ist keine Python-eigene Funktion und kann auf der Kursseite heruntergeladen werden als Teil des Moduls `toolbox_sr1.py` (siehe Übungsblatt 3). `nlsim()` benötigt außerdem die Funktion `rk()`, die sich im gleichen Modul befindet.
Hinweis: Erstellen Sie zunächst die beiden Funktionen $f(x, u)$ und $y(x, u)$. Erstellen Sie dann ein Skript, in welchem Sie x_0 , Δt sowie u definieren und anschließend `nlsim()` aufrufen.
8. Vergleichen Sie die Ergebnisse der linearen Simulation mit den Zuständen aus der nichtlinearen Simulation. Was fällt Ihnen auf?