

## 6. Übung

In dieser Übung soll ein modellprädiktiver Regler für den in Abbildung 1 dargestellten 3D-Laborkran entworfen werden. Die Zustände

$$\mathbf{x} = [x_T \quad \dot{x}_T \quad y_T \quad \dot{y}_T \quad l \quad \dot{l} \quad \theta_X \quad \dot{\theta}_X \quad \theta_Y \quad \dot{\theta}_Y]^T \quad (1)$$

bezeichnen die Position der Laufkatze, die Länge des Seils und die Auslenkwinkel des Seils, sowie deren Ableitungen. Als Stellgrößen

$$\mathbf{u} = [a_X \quad a_Y \quad a_l]^T \quad (2)$$

werden die Beschleunigungen der Laufkatze in X- und Y-Richtung sowie die Beschleunigung des Seils verwendet. Durch den Einsatz von unterlagerten Geschwindigkeitsreglern für die Motoren kann das vereinfachte

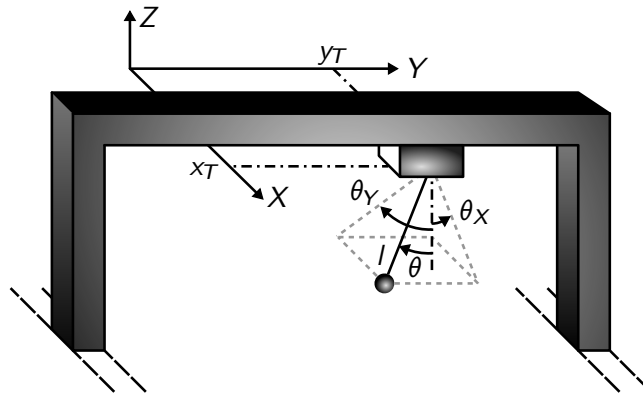


Abbildung 1: Schematische Darstellung des Laborkrans [1].

Modell der Krandynamik [1]

$$\ddot{x}_T = a_X, \quad \ddot{y}_T = a_Y, \quad \ddot{l} = a_l \quad (3a)$$

$$\ddot{\theta}_X = -\frac{1}{l \cos \theta_Y} (g \sin \theta_X + a_X \cos \theta_X + 2\dot{\theta}_X (\dot{l} \cos \theta_Y - l \dot{\theta}_Y \sin \theta_Y)) \quad (3b)$$

$$\ddot{\theta}_Y = -\frac{1}{l} (2\dot{l}\dot{\theta}_Y + \sin \theta_Y (g \cos \theta_X - a_X \sin \theta_X) - a_Y \cos \theta_Y + l\dot{\theta}_X^2 \sin \theta_Y \cos \theta_Y) \quad (3c)$$

hergeleitet werden. Die Stellgrößen unterliegen dabei den Beschränkungen

$$a_X \in \left[-2 \frac{\text{m}}{\text{s}^2}, 2 \frac{\text{m}}{\text{s}^2}\right], \quad a_Y \in \left[-2 \frac{\text{m}}{\text{s}^2}, 2 \frac{\text{m}}{\text{s}^2}\right], \quad a_l \in \left[-2 \frac{\text{m}}{\text{s}^2}, 2 \frac{\text{m}}{\text{s}^2}\right]. \quad (4)$$

### **Aufgabe 6.1**

Formulieren Sie ein geeignetes quadratisches Kostenfunktional für einen Positionswechsel der Last. Stellen Sie die zugehörigen Optimalitätsbedingungen auf und berechnen Sie die dafür benötigten Jacobi-Matrizen  $\frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}$  und  $\frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}$ . Skizzieren Sie den Ablauf der numerischen Lösung mit dem Gradientenverfahren.

### **Aufgabe 6.2**

Implementieren Sie einen modellprädiktiven Regler für den Laborkran in MATLAB unter Verwendung des vorgefertigten Templates für das Gradientenverfahren. Wählen Sie für den Zeithorizont  $T = 1.5\text{ s}$ , 40 Stützstellen für die Integration, eine Abtastzeit von  $\Delta t = 2\text{ ms}$  und 2 Iterationen pro Zeitschritt. Testen Sie den Regler für verschiedene Positionswechsel der Last.

### **Aufgabe 6.3**

Installieren Sie die MPC-Toolbox GRAMPC und machen Sie sich anhand der Dokumentation [2] sowie den Artikeln [3] und [4] mit der Benutzung vertraut. Implementieren Sie die modellprädiktive Regelung des Laborkrans unter GRAMPC und vergleichen Sie die Rechenzeit sowie die Trajektorien mit der MATLAB-Implementierung. Untersuchen Sie außerdem den Einfluss verschiedener Parameter wie beispielsweise des Zeithorizonts  $T$  oder der Anzahl der Iterationen pro Zeitschritt.

### **Aufgabe 6.4**

Für die Anwendung der unterlagerten Geschwindigkeitsregler sollen die Maximalgeschwindigkeiten des Wagens und des Seils begrenzt werden. Solche Beschränkungen können in GRAMPC mit äußeren Straffunktionen oder der erweiterten Lagrange-Methode berücksichtigt werden, wobei in beiden Varianten der Strafparameter durch eine Heuristik angepasst werden kann. Skizzieren Sie den Ablauf der numerischen Lösung mit der erweiterten Lagrange-Methode. Implementieren Sie die Beschränkungen unter GRAMPC und vergleichen Sie die Ergebnisse mit dem unbeschränkten Fall.

### **Aufgabe 6.5 Zusatzaufgabe**

Untersuchen Sie den Einfluss von normalverteiltem Messrauschen auf die modellprädiktive Regelung. Modifizieren Sie dazu den Anfangswert  $\mathbf{x}_0$  in jedem Abtastschritt des Reglers, wobei die Simulation des realen Systems jedoch mit dem tatsächlichen Wert erfolgen soll.

## **Literatur**

- [1] B. Käpernick. Gradient-based nonlinear model predictive control with constraint transformation for fast dynamical systems. Shaker Verlag, Aachen, 2016.
- [2] T. Englert, A. Völz, F. Mesmer, S. Rhein, K. Graichen. GRAMPC documentation. [sourceforge.net/projects/grampc](https://sourceforge.net/projects/grampc/), 2018.
- [3] B. Käpernick, K. Graichen. The gradient based nonlinear model predictive control software GRAMPC. In *Proc. European Control Conference (ECC)*, Seiten 1170-1175, Straßburg, Frankreich, 2014.
- [4] T. Englert, A. Völz, F. Mesmer, S. Rhein, K. Graichen. A software framework for embedded nonlinear model predictive control using a gradient-based augmented Lagrangian approach. *Optimization and Engineering*, doi.org/10.1007/s11081-018-9417-2, 2019.