① Laborversuch Tempomat

Zoltán Zomotor

Versionsstand: 18. Oktober 2016, 13:29

Inhaltsverzeichnis

1 Laborarbeit

Die Laborarbeit gibt Ihnen die Gelegenheit, sich individuell und praktisch mit der Regelungstechnik zu beschäftigen. Sie bearbeiten die Aufgaben alleine oder in einer Zweiergruppe. Die Aufgaben sollten sich während der vorgegebenen Vorlesungszeit lösen lassen.

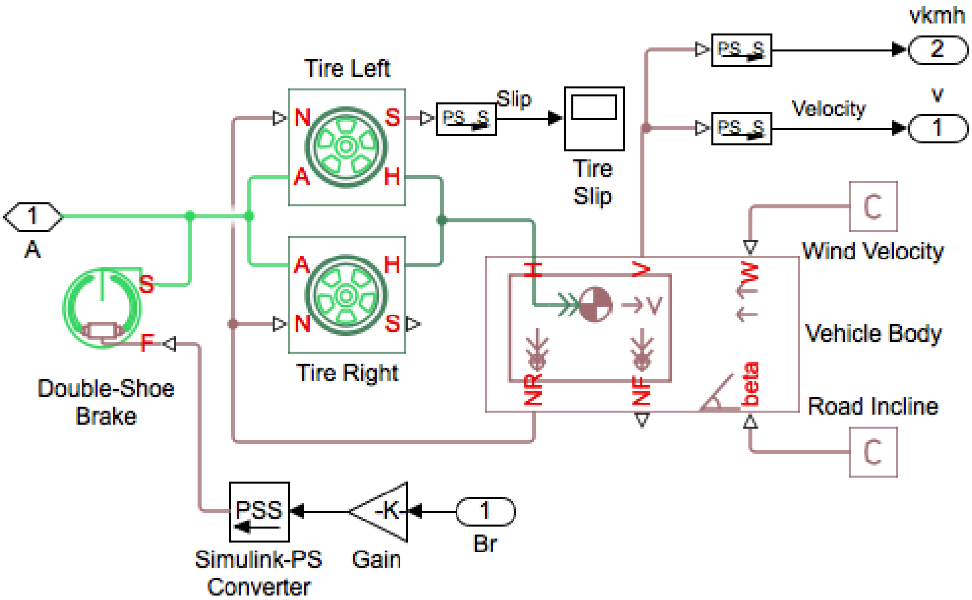
2 Laborbericht

Dokumentieren Sie Ihre Arbeit mit Screenshots und den Antworten auf die gestellten Fragen in einem PDF. Dieser und die folgenden Berichte werden bei Vollständigkeit mit jeweils einem Punkt bewertet, die Sie zusätzlich zu den Punkten in der Klausur gutgeschrieben bekommen. Den Laborbericht laden Sie bitte in den entsprechenden Moodle-Kurs hoch.

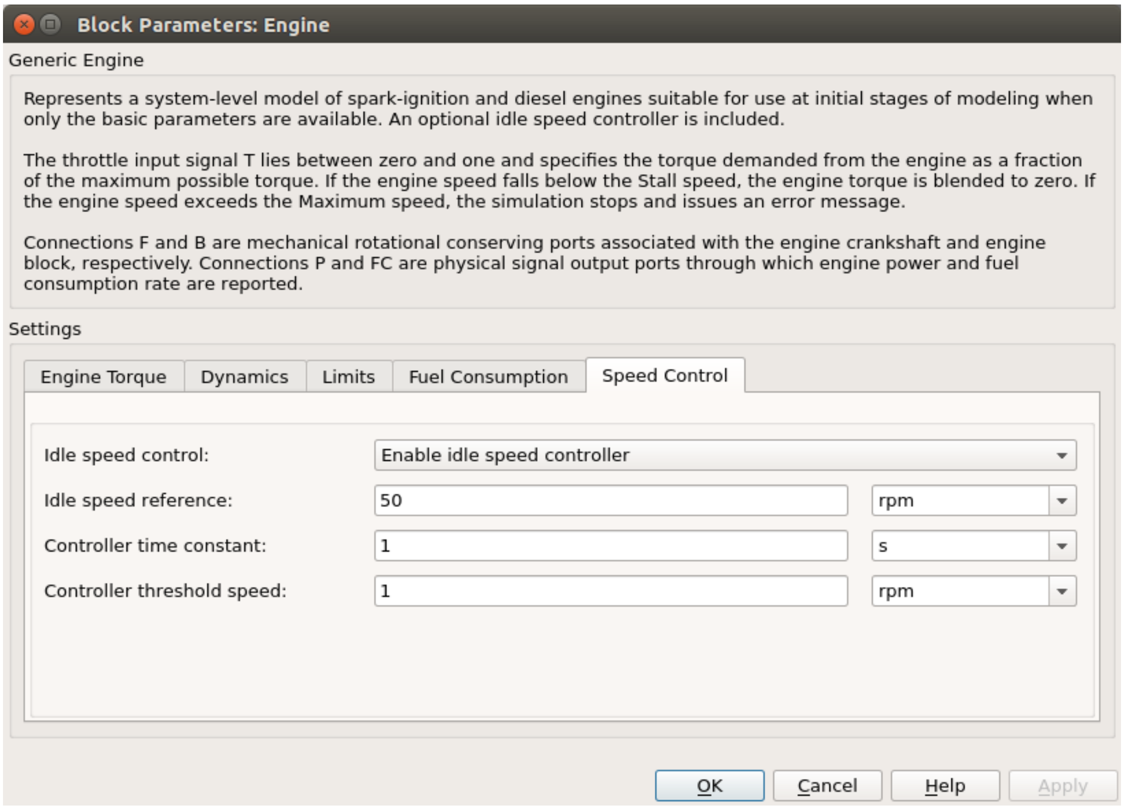
3 Vorbereitungen

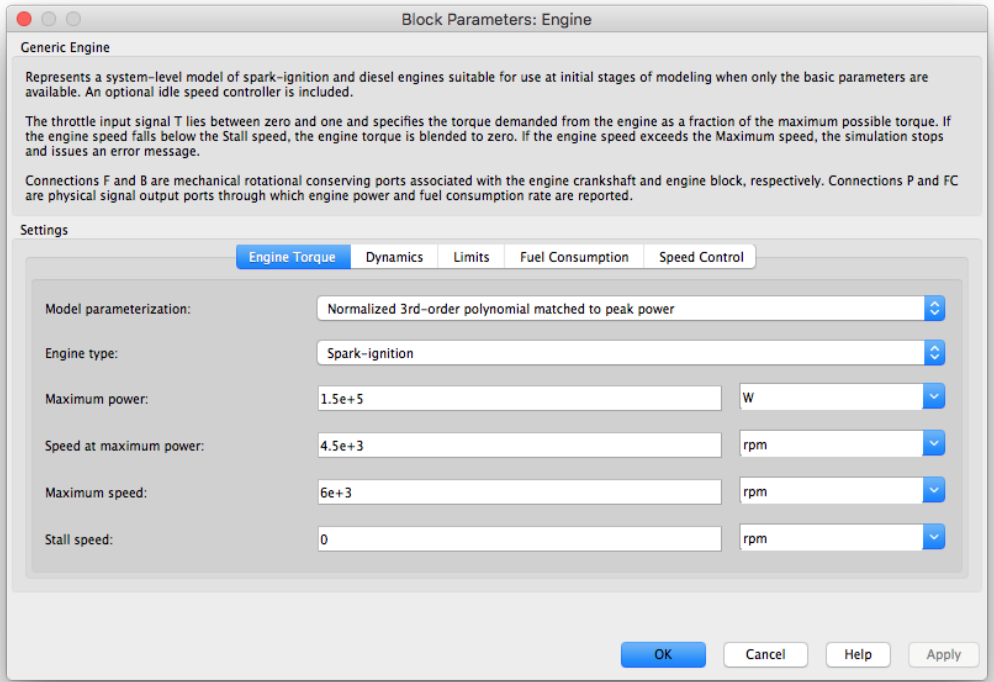
Geben Sie im Matlab-Fenster `sdl_car` ein und speichern das Modell lokal ab. Simulieren Sie das Modell mit unterschiedlichen Driver Inputs. Falls alles wie erwartet funktioniert, ändern Sie dann Folgendes:

- `sdl_car ▶ Vehicle Body`: PS-Simulink Converter duplizieren und Einheit km/hr setzen. Neuen Out-Port `vkmh` mit dem Converter und den Converter mit dem V-Signal des Vehicle Body Blocks verbinden, , *Road incline* auf 0.01 setzen (*PS Constant*-Block rechts unten).



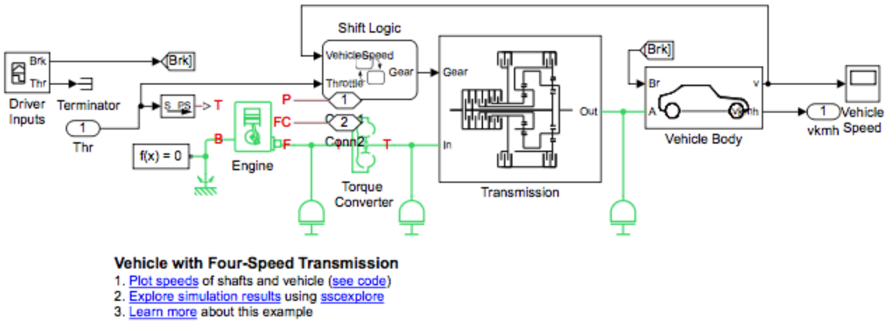
- `sdl_car ▶ Engine`: *Enable idle speed controller* setzen, *Idle speed reference:* auf 50, *Stall speed:* auf 0

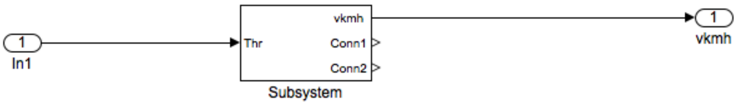




- Verknüpfung des Ausgangs `Thr` des Blocks `Driver Inputs` trennen und mit einem neuen In-

Port `Thr` verbinden, den offenen Ausgang mit einem Terminator verbinden (oder einen manual switch einbauen, so dass man zwischen äußerem Eingang und dem Driver-Inputs-Block umschalten kann), dann alles zu einem Subsystem zusammenfassen.

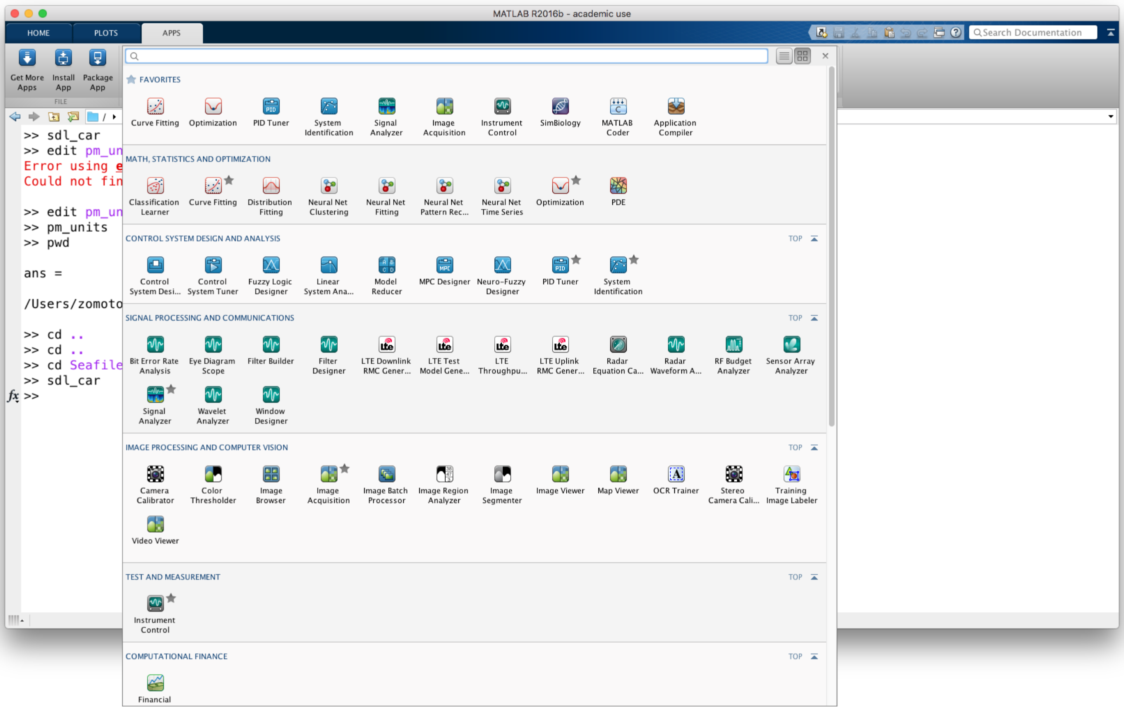




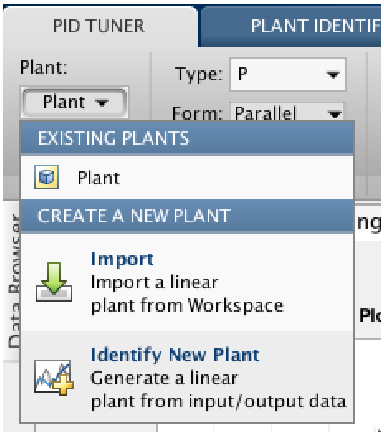
4 Modellidentifikation

Simulieren Sie einen Step-Input. Das Modell sollte sich erst bei Thr=0 einschwingen (zum Beispiel durch step time 100), so dass der Sprung in einem stationären Zustand auf das Modell wirkt. Nutzen Sie den *To Workspace*-Block, um die resultierende Geschwindigkeit in den Matlab-Workspace zu sichern. Nutzen Sie das Save-Format *Timeseries*. und setzen Sie die *Sample time* auf 0.04.

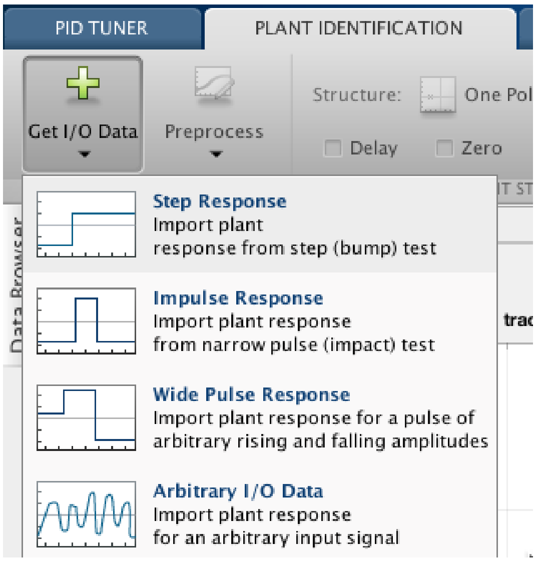
Starten Sie die Matlab-App *PID-Tuner*. Sie befindet sich in der Zeile *Control System Design and Analysis* Importieren Sie die Sprungantwort:



- Reiter PID TUNER, `Plant▾ Identify new Plant`



- Reiter PLANT IDENTIFICATIOM `Get I/O Data▾ Step  Response`



- Nutzen Sie Auto Estimate und verstellen Sie die Streckenparameter manuel (Button *Edit Parameters*), um die Übertragungsfunktion anzupassen. Probieren Sie alle vier vorgegebenen Übertragungsfunktionen aus (One Pole, Two Real Poles, Underdamped Pair, Underdamped Pair + Real Pole), die am besten zur Sprungantwort passt. Wichtig ist, dass die Überhöhung und der Stationärwert getroffen werden. Beim Anstieg darf die identifizierte Strecke auch daneben liegen. Wenn Sie fertig sind mit identifizieren klicken Sie auf Apply. Das linke Fenster zeigt dann die Sprungantwort des geschlossenen Regelkreises mit dem aktuellen Regler.

5 Reglerentwurf

Im Reiter PID TUNER ist als Regler-Typ *Type: P* und *Form: Standard* voreingestellt. Wählen Sie dieselbe Form wie im PID-Tuner Block, also jeweils *Parallel* oder *Standard.* Sie können den Wert mit dem Schieberegler *Response Time (seconds)* tunen. Exportieren Sie mit dem Button Export für Ihr System den Regler. Im Workspace gibt es dann die Variable C, die die Verstärkung des entworfenen P-Reglers enthält. Erweitern Sie Ihr Simulink-Modell, so dass Sie eine Referenzgeschwindigkeit vorgeben können, die dann vom entsprechenden Regler eingeregelt wird.

Ziel: Entwerfen Sie einen Regler, so dass die Referenzgeschwindigkeit genau eingeregelt wird, ohne dass die Geschwindigkeit schwingt oder überschwingt. Ein Sprung von 0 km/h auf 100 km/h des Referenzwerts sollte in weniger als 150 s eingeregelt werden.