

Bedienungsanleitung Fahrspurverfolgung TUCar

Leopold Mauersberger

16. Oktober 2018

1 Gültigkeitsbereich&Voraussetzungen

Diese Anleitung beschreibt den Start der im Rahmen der Bachelorarbeit „Kamerabasierte Navigation eines Modellfahrzeugs in einem Straßenverkehrsszenario“ implementierten Fahrspurverfolgung. Für ein weiterführendes Verständnis der hier beschriebenen Vorgehensweisen sei auf das ROS-Wiki ¹ sowie die MATLAB-Dokumentation, speziell die Abschnitte der Robotics-System-Toolbox ², verwiesen. Zudem wird die Fähigkeit mit einem Linux-Terminal zu arbeiten, vorausgesetzt.

2 Notwendige Verbindungen

Bevor die entsprechenden Applikationen gestartet werden können, muss der PC sich auf dem vom Raspberry PI bereitgestellten WLAN-Hotspot einloggen. Das benötigte Passwort liegt dem Fahrzeug bei oder ist beim Admin zu erfragen. Zudem empfiehlt es sich, kurz ein LAN-Kabel am Raspberry PI anzustecken, um die Zeit des Einplatinencomputers ohne Echtzeituhr einzustellen. Einige generierte Debugging-Daten besitzen andernfalls keine Aussagekraft.

3 Raspberry PI

Ist die Wlan-Verbindung zwischen PC und PI hergestellt, kann sich in einer neuen Konsole via

```
ssh lemau@10.0.0.1
```

auf diesem eingeloggt werden. Das Passwort lautet *tucar2018*. Nun kann das Launchfile zum Starten aller notwendigen Nodes aufgerufen werden:

```
roslaunch tucar tucar_bringup.launch use_sensors:=true
```

Beendet werden diese durch die Tastenkombination STRG + C. Der PI kann mittels

```
Poweroff
```

heruntergefahren werden.

¹<http://wiki.ros.org/>

²<https://de.mathworks.com/products/robotics.html>

3.1 bashrc

Der Einfachheit halber wurden die Befehle

```
export ROS_HOSTNAME=10.0.0.1
source ~/tucar_ws/catkin/devel/setup.bash
```

zum Festlegen IP-Adresse des Netzwerkinterface, welches ROS verwenden soll und zum Sourcen der Nodes des TUCars schon in die *bashrc* des PI eingetragen.

4 PC

4.1 Matlab

Am PC kann nun MATLAB gestartet werden. Hierbei ist es wichtig, dies aus dem Ordner *Riverflow* zu tun, da sich hier das Skript *startup.m* befindet, welches die benötigten Custom-Messages des TUCars lädt, so wie alle relevanten Unterordner dem Matlab-Pfad hinzufügt:

```
cd .../Riverflow
matlab&
```

Auch das von ROS am PC zu nutzende Netzwerk-Interface muss festgelegt werden. Dies geschieht beim Start des Matlab-Skriptes automatisch. Es kann jedoch notwendig sein, diese IP anzupassen:

```
ifconfig
```

in einem weiteren Terminal aufgerufen liefert Informationen zu allen vorhandenen Netzwerkadaptern, der WLAN-Stick sollte hier ausfindig zu machen sein. Die gefundene Adresse kann nun gegebenenfalls in der Datei *function_init_parameters.m* im Ordner *Initialisierungen* unter Punkt *ROS-Parameter* bei *params.RosIP* eingetragen werden.

Zum Start der Fahrspurverfolgung muss lediglich die Datei *main.m* ausgeführt werden. Die Debuggingmöglichkeiten können von hier aus eingeschaltet werden, genaue Erläuterungen finden sich in den Kommentaren. Zum Auswerten dieser sind die Skripte im Ordner *Debug* zu nutzen. Die Arbeitsweise des Quellcodes kann nach lesen der zugehörigen Bachelorarbeit anhand der Funktionsbeschreibungen durch rekursives abarbeiten dieser ergründet werden.

4.2 Konsole

Soll nicht nur via Matlab mit dem TUCar kommuniziert werden, so muss ein weiteres Terminal geöffnet werden, in dem die IP des ROS-Masters (PI) bekannt gemacht und die richtige Netzwerkschnittstelle eingestellt wird.

```
export ROS_MASTER_URI=http://10.0.0.1:11311 # IP PI-Hotspot
export ROS_IP=x.x.x.x # IP Wlan-Stick -> siehe ifconfig
```

Eine kleine Liste nützlicher Befehle ist:

```
# verfügbare Topics anzeigen
rostopic list
# Topicdaten auf Konsole ausgeben
rostopic echo topicname
# Kamerabild anzeigen
rqt_image_view
```

5 Parameteranpassung TUCar

5.1 Fahrplattform

Die Konfiguration der Fahrplattform kann mittels der Datei

```
tucar_controller.yaml
```

im Ordner

```
~/tucar_ws/catkin/src/tucar/tucar/config/platform
```

erfolgen.

5.2 Kamera

Die Datei

```
camera.xml
```

im Ordner

```
~/tucar_ws/catkin/src/tucar/tucar/launch/sensor_drivers
```

ist für die Konfiguration der Kamera zuständig.