User Manual

Precisionsreglering av gaffeltruck

December 5, 2019

Version 1.0

Reviewed	Jonatan Gustafsson	2019-12-02
Approved	Erik Hedberg	2019-12-03

Course name:	Reglerteknisk projektkurs	Group E-mail:	jongu926@student.liu.se
Course code:	TSRT10	Responsible for the document:	Jonathan Carlin
Project:	Precisionsreglering av gaffeltruck	Name of document:	Design specification

Project Identity

Group E-mail:	jongu926@student.liu.se
Orderer:	Erik Hedberg, Linköpings universitet Phone: +46 13 28 13 38 E-mail: erik.hedberg@liu.se
Supervisor:	Olov Holmer Phone: +46 13-28 16 17 E-mail: olov.holmer@liu.se
Course Responsible:	Daniel Axehill Phone: +46 13 28 40 42 E-mail: daniel.axehill@liu.se

Participants of the group

Name	**Responsibility**	**E-mail**
Jonatan Gustafsson	Project leader (PL)	jongu926@student.liu.se
Jonathan Carlin	Responsible for the documentation (DOC)	jonca673@student.liu.se
Sebastian Haglund	Responsible for the testing (TEST)	sebha953@student.liu.se
Jennie Bäcklin	Responsible for the design (DES)	jenba497@student.liu.se
Carl Hynén Ulfsjöö	Responsible for the software (SOFT)	carhy681@student.liu.se
Henrik Johansson	Responsible for the quality (Q)	henjo648@student.liu.se
Linus Roos	Responisble for the public relations (PR)	linro231@student.liu.se

CONTENTS

DOCUMENT HISTORY

Version	Datum	Changes made	Sign	Review
0.1	2019-11-29	First Draft	All members	JG, HJ, SH
0.2	2019-12-02	Second Draft. Fixed comments from Supervisor and Orderer.	HJ	JG
1.0	2019-12-05	First Version	JG	EH

1 INTRODUCTION

This document is a user manual for the project *Precionsreglering av gaffeltruck* given in the course *TSRT10 Reglerteknisk projektkurs*. The user manual provides the necessary information for users and developers to run the model in simulations and in real-time on the forklift. The user manual is made up of two parts; one describes how to run the model as a simulation using MATLAB, Simulink and Simscape, and the second describes how to run the model on the forklift using Speeadgoat, a real-time target computer, and Simulink.

2 SIMULATIONS

The software and add-ons listed below are required to run the simulation in Simulink.

2.1 Software Requirements

- Microsoft Windows
- MATLAB 2019a
- Simulink
- Simscape
- C++ compiler for MATLAB, MinGW-w64 or Visual C++ 2017

2.2 Add-On Requirements

The listed add-ons can be installed via the MATLAB Add-On explorer.

- MATLAB Coder
- Simulink Real-Time
- Simulink Test
- Simulink Coder
- Simscape Driveline
- Simscape Electrical
- Simscape Fluids
- Simscape Multibody
- Simscape Multibody Contact Forces Library
- Simscape Multibody Multiphysics Library

2.3 Open the Model

1. Open the project *model_based_design.prj*.
2. Open the model *top_model.slx* in the folder *models*.

2.4 Start Simulation from Simulink

Open *controller.slx* and go to Model Configuration Parameters and make sure that solver is set to Variable-step. To start the simulation press Run. Note that this gives open loop control with control signals defined in the Signal builder found in the block *top_model/controller (controller)/Variant Subsystem/open_control* in Simulink. It is possible to change the control signals in the signal builder. To enable the controller, define signals in MATLAB or use signals from an old test, see the section below.

2.5 Setup Simulation in MATLAB

This method makes it possible to use the controller, defining your own signals or use signals from old tests.

1. Open *simulation_setup.m* found under *cdio/simulink_setup*.
2. Open *controller.slx* and go to Model Configuration Parameters and make sure that solver is set to Variable-step.
3. Setup your simulation according to Table 1 and then run the script.
4. Before starting the simulation make sure the current working directory is *work*, otherwise you will get a simulation error.
5. Start the simulation in Simulink.
6. After the simulation all logged signals can be found under Simulation Data Inspector in Simulink.
7. Export the logged data to a *.mat* file to save them.

Parameter	Description	Unit
use_controller	Set to 1 to use controller. All control parameters will automatically be set up and reference trajectories will be calculated based on the inputs in this table.	[-]
controller_in_real_time	Set to 1 to set a 5 s offset to trajectories. This is needed when using Simulink real-time to switch the control from the forklift to Simulink through Speedgoat.	[-]
goal_height	Set goal height. This is the total lift height for the trajectory.	[mm]
mass_load	Set the nominal load mass. This parameter changes the weight of the load that the optimal trajectory will be based on. Note that the weight has to be changed manually in the Simscape model.	[kg]
use_designed_signals	Set to 1 to use designed signals. This allows the user to design control signals in MATLAB that can be sent to Simulink as time series. This enables open loop control of the model from MATLAB.	[-]
use_measured_signals	Set to 1 to use measured signals from a real test. This option will read the measured signals from a real test and use them as input signals to Simulink. If this is set to 1, then the *test_file_name* must be specified.	[-]
opt_mode	Set to 'Energy' or 'Time'. This makes it possible to choose which mode the truck will be in. *Time* will give a fast lift with low oscillations. *Energy* will give an efficient, with respect to energy, lift with low oscillations.	[-]
test_file_name	String with the name of test file. This string will define from which file the input signals will come from if *use_measured_signals* is set to 1.	[-]
do_plot	Set to 1 to plot the trajectory or the input signals. This flag will provide plots of the calculated trajectories if *use_controller* is set to 1 or plots of the input signals if *use_measured_signals* or *use_designed_signals* is set to 1.	[-]

Table 1: The setup choices that the user has to make in simulation_setup.

3 SPEEDGOAT

To run the controller on the forklift using the Speedgoat real-time hardware, a setup is required for the software environment.

3.1 Setup

1. Download Speedgoat library and Speedgoat documentation.
2. Follow the instructions in the Speedgoat quick start guide.
   - Note that the compiler must be Visual C++ 2017.

3.2 Run Real-time Target

1. Make sure the Simulink environment works, see Section 2.
2. Connect Ethernet cable.
3. Open *top_model_rel_time_target.slx* under *models/rrt_interface*
4. Open *simulation_setup.m*, set it up according to Table 1 and then run it.
5. Go to the controller block in *top_model_rel_time_target.slx* and make sure that the variant is set to controller.
6. Open the model *controller.slx* and change the step size to Fixed-step under Model Configuration Parameters.
7. Change the setting of Target in Simulink to external.
8. Log the desired output signals in Simulink.
9. Run the Simulink controller using the following steps:
   - Build the model
   - Connect to target
   - Press run
10. When test is done press stop.
11. The logged signals can be seen under Simulation Data Inspector where it is also possible to export the signals to desired file format.

3.3 General Tips for Simulink Real-time

- Remember to check the input signals when using open loop control. *do_plot* flag described in Table 1 is useful.
- Time delay on input signals might be useful. *controller_real_time* in Table 1 is useful.
- It is possible to build the model without having the computer connected to the forklift. If the build is successful, the error message should be: *Could not find target*.

3.4 Adding New Model Under Controller

When adding a new model in *controller.slx* go to Model Configuration parameters and fix the following things:

- Go to Code generation then set System target file to *slrt.tlc*
- Change the step size to Fixed-step.
- Set "Allow tasks to execute concurrently on target" to off.