Supervisory Control of a Warehouse Robots System

Final Assignment 2018-2019 (4CM30)

February 19, 2019

Introduction

The purpose of this final assignment is to allow you to demonstrate that you master the different subsequent steps encountered in model-based engineering of supervisory controllers as proposed in the course Supervisory Control (4CM30). These steps are

1. modelling an uncontrolled system (plant) by means of a network of automata,
2. use of the simulation capabilities of CIF to validate this model,
3. formulation of safety requirements, both informally and formally,
4. synthesis of a supervisory controller that satisfies these requirement,
5. simulation-based validation of the obtained controlled system,
6. translation to the verification tool mCRL2,
7. formulation of some progress properties, both informally and formally, and
8. verification of the controlled system against the safety requirements and the progress properties.

Warehouse systems - background information

Currently, the forth industrial revolution is taking place, called Industry 4.0, where the physical world and the digital word are intertwined resulting in cyber-physical system [1]. This industrial revolution is especially visible in warehouses and distribution centers within (global) supply chains [2]. Automation has already been used for decades within warehouses and distribution centers, but often required standardized trailers, products, and containers. With the rapid growth of e-commerce (in 2017 a growth of 16% [3]), over 10 billion parcels have been shipped in the USA alone [3], each parcel having a different size and different destination. To cope with this increasing demand and product variability, several technological advances are deployed nowadays, like vision picking, adaptive connected automated guided vehicles, and fully automated quality assurance [2].

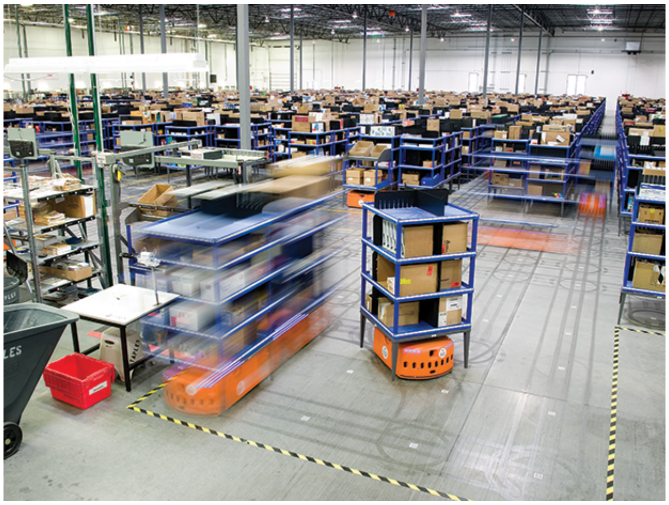

Figure 1: Amazon Robotics, or formerly known as Kiva System. Picture from [8]

In this project, we will focus on connected automated guided vehicles in warehouses and distribution centers. Several examples of these systems include Amazon Robotics (formerly known as Kiva Systems) [4], Symbotic (formerly known as CasePick Systems) [5], Adapto [6], and Fleet [7]. There are numerous advantages of a fleet of these vehicles, all in order to cope with changing demands of warehouse and distribution centers [8]. First, the design of these vehicles allow for storing, transporting, and picking of products in any shape, size, and weight. Second, the system provides flexibility to ensure just-in-time arrival of the products at the next (processing) station. Third, moving products by robots to the warehouse workers is faster rather then vice versa. Forth, the space is used more efficiently, as vehicles can move close to each other (e.g., the Amazon Robotics can move within 15 centimeters of each other). Finally, there is no single point-of-failure in the system making it robust. Due to the succes of these connected automated guided vehicles, Amazon has reported in 2018 that it uses more than 100,000 robots in its warehouses [9].

The Amazon Robotics system consists of numerous au-

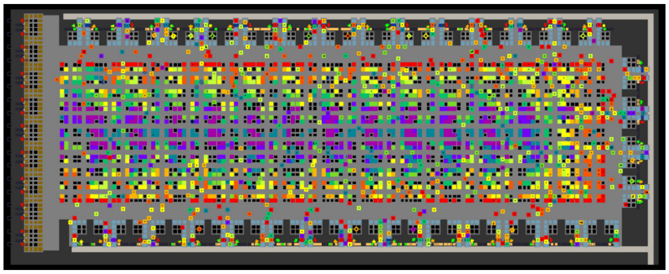

Figure 2: Simulation environment of the Amazon Robotics system used to design and validate the system. Picture from [10]

tonomous agents moving along a grid in a warehouse and a centralized server. The centralized server communicates with the robots over a secure WiFi network and solves the resource allocation problem, i.e., it determines which robot should pick which product rack, which picking station drive to, and where to store the rack after service by the warehouse worker. The drive units (agents) are differential-drive two wheeled robots equipped with numerous sensors [8, 10]. It is able to drive with two brushless DC motors along a grid and making turns of 90 degrees. Each robot can senses its location by reading barcode stickers on the floor with previously a barcode laser and nowadays a camera. Each barcode represents a coordinate in the grid at which the drive units are allowed to turn. Arriving at a location in the storage area, the robot is able to lift or lower a product rack from the ground with a screw mechanism. The drive unit is able to validate the correct product rack as it uses a camera to read bar codes underneath the racks. As these drive units move autonomously, several infrared proximity sensors are present to observe the nearby area around itself in order to prevent collision with robots or humans. Furthermore, touch-sensitive bumpers are used to detect collisions.

The proper functioning of these connected automated guided vehicle systems requires solving several control challenges on different abstraction levels [10]. On the low level, proper feedback loops needs to be designed to drive the vehicles with high precision with cheap equipment (motors, encoders). A level higher, each robot needs to operate safely in a warehouse environment, ensuring no collisions with other vehicles, humans, or other equipment, and a safe execution order of the different tasks (like raising the rack, turning 90 degrees). Again, one level higher a path planning problem needs to be solved to calculate the best route from the current position to the desired destination. Finally, at the highest level a dynamic resource allocation problem is solved constantly to assign to each drive unit a goal it autonomously needs to fulfill. Simulations are used to verify and validate these control systems [10].

In the final assignment of the course Supervisory Control, we will only focus on developing a supervisory controller for connected automated guided vehicles to ensure safe behavior of these vehicles. We assume that all other control systems are in place and designed correctly (which does not imply that no faults may happen).

Problem setting

In this assignment, we consider a warehouse system where m connected autonomous guided vehicles move around to transport products from a storage area to picking stations and back. The number m is not specified beforehand, as the complexity of designing and synthesizing supervisors increases when m increases. The warehouse area is partitioned in a grid along which the vehicles may move. Each point (or coordinate) in the grid is a location where the vehicle may perform special actions, like rotating, lifting a product rack, or charging the battery. The locations can be grouped into four types: storage, driving, picking, and charging. The layout as shown in Figure 2 may be used: in the middle there is a rectangular area containing $2 \times n$ blocks of storage locations separated by driving lanes, this rectangular area is surrounded by driving lanes, on three sides of the warehouse there are picking stations, and on one side there are charging stations.

Safe behavior of these vehicles needs to be guaranteed, especially when working together with humans. Eventually (or ideally), it is desired to have a supervisory control system ensuring safe behavior in an environment with faults. Some examples of faults include, but not limited to, observing a product rack as a vehicle, vehicle stuck due to empty battery, unknown obstacles along the way, and slippery surfaces (because a product has fallen of the reck) resulting in encoder errors.

Design and simulation of the supervisory controller

In this project, we simulate a typical design process where some specifications of the used hardware and desired control system are given initially, but they are incomplete and vague, they change over time, and as the design progress, more functionality needs to be incorporated. The specifications in this document act as a starting point.

Supervisor architecture

While a distributed control architecture is used for the Amazon Robotics system, no specific supervisor architecture is specified for our warehouse system. A centralized supervisor may be designed to coordinate all vehicles in the system to ensure safe behavior. While choosing this architecture may ease formulating the control problem, limits of computational power may be encountered during synthesis.

If a modular architecture is chosen, it should be shown that the set of supervisors do not conflict with each other, as blocking is not desired in this warehouse system.

Interface specifications

There are several interfaces to other (sub)systems. Below several specifications are provided about these interfaces.

Low-level controllers. Several low-level feedback controllers are present in the vehicle to actuate its behavior. The supervisor is able to set the references for these controllers. For the DC motors, the supervisor is able to command it to move the vehicle forward, to stop its forward movement, to rotate 90 degrees clockwise, and to rotate 90 degrees counter clockwise. It receives back from the encoders whether the vehicle is moving forward or not, and it receives a signal indicating the rotation has finished. Furthermore, the product stack lifting mechanism can be commanded to raise, lower, or stop. Two limit switches indicate whether the lifting mechanism is in its upper or lower position; no signal from both switches indicates it is somewhere between its limits. Finally, the battery controller may signal to the supervisor when the battery needs to be recharged and when the battery is critically low.

Resource allocation controller. The vehicle is able to ask the resource allocation controller for a new destination. It can ask for a location to pick up a new product rack, a picking station to drive to, a storage location to store the current product rack, and a battery charging location. Depending on the load of this controller, a destination location is provided shortly after the request. After requesting a picking location, it may be possible that no location is provided when no picking location needs one of the product on the current lifted product rack.

Path planning controller. We assume that the path planning controller both calculates a new path (for example with Dijkstra's Algorithm) and executes this path. The path planning controller has the same interface to the low-level controllers as the supervisor. When the supervisor disables certain low level commands, the path planning controller adheres to this disablement (i.e., the supervisor can overrule the path planning controller). The supervisor is able to start the path planning with the latest received destination, to start the path execution, and to stop the path execution. After starting the path planning, the supervisor receives a signal when it has finished planning; and after starting the path execution, the supervisor receives a signal when the destination has been reached (the path execution is automatically stopped).

Location barcode camera. From the location barcode camera, located at the bottom of the vehicle, the supervisor receives the following information: whether a location barcode is visible, and the type of location (storage, driving, picking, and charging).

Product stack barcode camera. From the product stack barcode camera, located at the top of the vehicle, the supervisor receives the signal whether a barcode is visible.

Communication. A WiFi network is available to communicate with other vehicles. It should be specified clearly when what is communicated, and this communication should be minimized in order to prevent congestion on the WiFi network.

Safety specifications

For safety purposes, the vehicle is equipped at the front of the vehicle with two sensors: an infrared proximity sensor measuring the presence of other object directly in front of the vehicle, and a touch-sensitive sensor indicating whether something touches the vehicle. If the vehicle needs to sense in other directions, it needs to rotate.

In this early stage of the design process, the system engineers have formulated already the following safety specifications.

Safe movement.

M1 Special actions, like rotating, may only be performed when the vehicle is at a grid point as indicated by the location barcodes.

M2 The vehicle may not move when it is raising or lowering its lifting mechanism.

M3 The vehicle may not collide with other vehicles.

M4 If a product rack is lifted, it may not collide with other product racks.

M4 When the infrared proximity sensor is activated, the vehicle should stop immediately.

M5 When the touch-sensitive sensor is activated (for whatever reason), the vehicle should stop immediately.

M6 When the battery is critically low, the vehicle should stop moving and wait for human assistance.

M7 After the battery being recharged, the vehicle may only start moving again when the battery is fully charged.

Path planning and execution.

P1 Path planning should only be started when no path execution is in progress.

P2 Path execution may only be started after calculating a new path.

P3 The current execution of the path needs to be terminated when a low battery signal is received.

Location requests.

L1 The order of requesting for types of locations is, under normal circumstances, pick up location, picking location (one or more times), drop location, pick location, etc.

L2 When the vehicle is at a picking location, it should first ask for a new picking location. A new storage location can only be requested when the resource allocation controller has returned no location for the picking location request.

L3 A new destination may only be requested when it has reached its previous destination.

L4 The vehicle should wait with performing other actions until a new destination is received.

L5 A battery location may only be requested when the battery is low.

Obstacle avoidance. When an obstacle is detected in the near proximity, the vehicle should stop immediately (requirement M4). Unfortunately, the systems engineers do not know how to proceed after identifying an obstacle. The obstacle may be another moving vehicle just crossing your path. In this case, the sensor would go off eventually, so the vehicle can proceed moving again. But the obstacle may also be a vehicle running out of power and therefore stopped moving. This obstacle is likely not to move in the near feature, so the vehicle needs to avoid this obstacle (and not wait indefinitely). The systems engineers assigned the supervisor design team to come up with an appropriate control strategy.

Other remarks

Several details and design choices are not specified here to endow the project with a significant level of flexibility also on the formulation side. An important part of the project is therefore to make assumptions under which the proposed methods will function and make a significant progress towards automated warehouse robots shaped by the expertise of the members of the group. Moreover, the students are highly encouraged to propose other functionalities to the system and changes to the present project.

Handing in

You are required to hand in a final report in CANVAS. With the final report also provide your CIF and mCRL2 models electronically.

Your final report contains

- clear informal description of the uncontrolled system (including system layout),
- models of the uncontrolled system, with explanation,
- list of informal safety requirements and their models used for synthesis,
- model of the supervisor obtained from synthesis,
- list of informal additional progress properties and their mCRL2 formulations,
- results of the verification of the safety requirements and the progress properties on the controlled system,
- collection of traces (in the form of a CIF model for each trace) that may be used to test the controlled system for relevant required behaviour as well as absence of relevant forbidden behaviour (these traces can best be provided as automata models that dictate which events occur in which order).

Evaluation

Your final project will be evaluated on the following aspects:

1. level of ambition of the proposed concrete case study,
2. correctness, clarity and elegance of the CIF model of the uncontrolled plant and its explanation,
3. correctness, clarity and elegance of the informal requirements and their models in CIF,
4. correctness of the formalization of the properties in mCRL2,
5. correctness of informal and formal progress properties,
6. level of ambition of the proposed progress properties,
7. correctness of formalization of the progress properties in mCRL2.

Notes:

ad 1) If your uncontrolled system is too simple (and in this sense avoids problems in the uncontrolled system) this is not valued very much.

ad 2,3) Elegance and clarity are much helped by having a clear relationship between concepts from the informal description and their formal counterparts.

ad 6) One can consider very simple progress requirements or more interesting ones. This will be taken into account in the evaluation.

References

[1] Heiner Lasi, Peter Fettke, Hans-George Kemper, Thomas Feld, Michael Hoffmann (2017). Industry 4.0. Business & Information Systems Engineering 6 (4) pp. 239-242.

[2] Alan Taliaferro, Charles-Andre Guenette, Ankit Agarwal, Mathilde Pochon, Industry 4.0 and distribution centers - Transforming distribution centers through innovation, 2016. Available at https://www2.deloitte.com/insights/us/en/focus/industry-4-0/warehousing-distributed-center-operations.html

[3] Kleiner Perkins, Internet Trends Report, 2018. Available at https://www.kleinerperkins.com/perspectives/internet-trends-report-2018/

[4] https://www.amazonrobotics.com/

[5] https://www.symbotic.com/

[6] https://www.vanderlande.com/warehousing/innovative-systems/storage-asrs/adapto

[7] https://www.vanderlande.com/airports/evolutions/fleet

[8] Erico Guizzo (2008). Kiva Systems: Three Engineers, Hundreds of Robots, One Warehouse. IEEE Spectrum. Available at https://spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-one-warehouse

[9] https://www.dailymail.co.uk/sciencetech/article-5808319/Amazon-100-000-warehouse-robots-company-insists-replace-humans.html

[10] Raffaello D'Andrea, Peter Wurman (2008). Future challenges of coordinating hundreds of autonomous vehicles in destribution facilities. IEEE Conference on Technologies for Practical Robot Applications.

[11] Compositional Interchange Format, available at http://cif.se.wtb.tue.nl/