

Autonomous Robot Supervision using Fault Diagnosis and Semantic Mapping in an Orchard

Blanke, Mogens; Blas, Morten Rufus; Hansen, Søren; Andersen, Jens Christian; Caponetti, Fabio

Published in:
Fault Diagnosis in Robotic and Industrial Systems

Publication date:
2012

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Blanke, M., Blas, M. R., Hansen, S., Andersen, J. C., & Caponetti, F. (2012). Autonomous Robot Supervision using Fault Diagnosis and Semantic Mapping in an Orchard. In G. Rigatos (Ed.), Fault Diagnosis in Robotic and Industrial Systems (Chapter 1, pp. 1-22). iConcept Press Ltd.

DTU Library

Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Autonomous Robot Supervision using Fault Diagnosis and Semantic Mapping in an Orchard

Mogens Blanke^{1,4}, Morten Rufus Blas³, Søren Hansen^{1,5}, Jens Christian Andersen¹, Fabio Caponetti^{1,2}

¹*Department of Electrical Engineering
Technical University of Denmark, Denmark*

²*Vestas Technology R & D
Vestas Wind Systems A/S, Denmark*

³*Research & Development Section
CLAAS Agrosystems, Denmark*

⁴*CeSOS, Centre for Ships and Ocean Structures
Norwegian University of Science and Technology, Norway*

⁵*Danish Forces Joint UAV Team
Danish Defence, Denmark*

1 Introduction

Classification and perception for autonomous vehicles in natural environments is notoriously difficult. One of the problems is that high dimensional data is often involved like those coming from video streams. With stochastic sensor signals that have limitations in accuracy and range, it is hard to model and generalise objects and contexts. Metric localisation, which for outdoor applications is commonly based on position from Global Positioning System (GPS) Satellites combined with motion sensing from Inertial Measurement Units (IMU), provides very useful information, but needs to be supplemented by environment sensing and perception to obtain safe autonomous operation. Semantic mapping can be used to extract non-metric features from sensors such as cameras and laser scanners by dimensionality reduction and can be used to provide a way of supervising metric localisation approaches. In a changing natural environment, autonomous operation is a challenge because GPS availability can be irregular or prone to outliers, and feature extraction from advanced sensors frequently suffers from artifacts. Robust methods to recognise objects and scenes are, therefore, a key to assure that a robot shows correct and safe behaviour, even in faulty conditions or unforeseen situations (Bouguerra et al., 2008; Chandrababu & Christensen, 2009). Semantic mapping will in this chapter be defined as a function mapping a high dimensional input down into a set of semantic features such as locations, obstacles or features of the environment. Safe autonomous operation will require robust semantic mapping and object recognition, and the technology needed must reach far

beyond conventional robotic motion planning, (LaValle, 2006; Mettler et al., 2010).

Recent developments in outdoor robots and sensor technology have made autonomous field operations a realistic aim (Blas & Blanke, 2011). The challenge is to develop and add functionality so that vehicles will behave in a safe and reliable manner under unmanned operation. Safe behaviour is crucial if outdoor robots are to become acceptable to authorities and society. High reliability is also required if robots are to be attractive to farmers and other professional users. The biggest technological challenge in such autonomous outdoor systems is ability to sense the environment, as well as to classify and use perceptual information for control.

This chapter will discuss diagnosis, monitoring and classification of current conditions in the environment and supervision at different levels in an autonomous robot. Based on stochastic automata (Lunze, 2001), a recent approach to create and use outdoor semantic maps (Caponetti et al., 2011) is demonstrated to provide autonomous navigation in an orchard. The role of supervision based on conventional diagnostic methods (Blanke et al., 1997; Blanke et al., 2006; Isermann, 2006; Noura et al., 2009) is shown in the context of localisation. Supervision based on representation of semantic characteristics of the environment (Galindo et al., 2005a; Galindo et al., 2005b; Galindo et al., 2008) is introduced in a framework of stochastic automata, the states of which conveniently correspond to different characteristic environments, hence giving models an intuitive interpretation. The states are determined from probabilistic analysis of the complex sensor signals from laser scanner and stereo vision, using perception patterns and vehicle motion history to determine when to switch state. It is discussed how signal quantisation can enhance robustness against noise and achieve fault-tolerant performances, and how quantisation of signal spaces can be optimised according to the probability distributions of observations. Combining well-known task planning and re-planning methods (Choset et al., 2005) with robust semantic mapping, it is shown how safe operation can be achieved.

1.1 Autonomous Operation - Illustrated by an Orchard Case

To diagnose whether behaviours are normal or not-normal is instrumental to obtain true autonomy with dependable operation (Avizienis et al., 2000). Fault diagnosis technology offers well established methods to obtain such information at levels where information is metric. At a higher level of abstraction, detailed metrics become impossible in a natural environment and is supplemented by semantic classification. Semantic mapping is of extreme importance for mobile robots (Galindo et al., 2005a). With a good state estimate of a semantic map, this information can be used to supervise plan execution, redefine controllers, or aid the localisation process, helping to assure safety and reliability. Semantic mapping in indoor environments is commonly done by matching the metric position of the robot in a known map. This is still viable outdoors but falls short when evaluated in terms of robustness. Missions have to continue, regardless of weather or season, leading to the need for a sensing system robust against vegetation shape, environment changes or localisation faults. Fig. (1) illustrates the use of an automaton model to represent semantic information of an orchard.

To validate robustness of solutions, data used in this chapter were recorded during autonomous operations of a tractor in an experimental orchard owned by Copenhagen University (Griepentrog et al., 2009). The area is characterised by geometric features: trees or plants, set along straight and parallel lines as visible in Fig. (2). Orchard maintenance is an application, which is of high commercial interest to automate, because tasks are repetitive and labour intensive. What makes it difficult to automate is that the trees in the orchard tend to occlude satellite reception for GPS systems. While this does not mean complete loss of localisation it has the consequence that it can be hard to localise a tractor to a specific row of trees. This is crucial when accurate and safe navigation in the orchard is required. The accuracy needed calls for a GPS receiver based on real-time kinematics (RTK) technology, which uses carrier frequency phase tracking, but these are sensitive to shadowing from trees with leaves, and their readout may suffer from artifacts. Conventional GPS receivers in the 1-2 m class are less prone to artifacts but do not have the required accuracy unless used in a sensor fusion solution with other instruments.

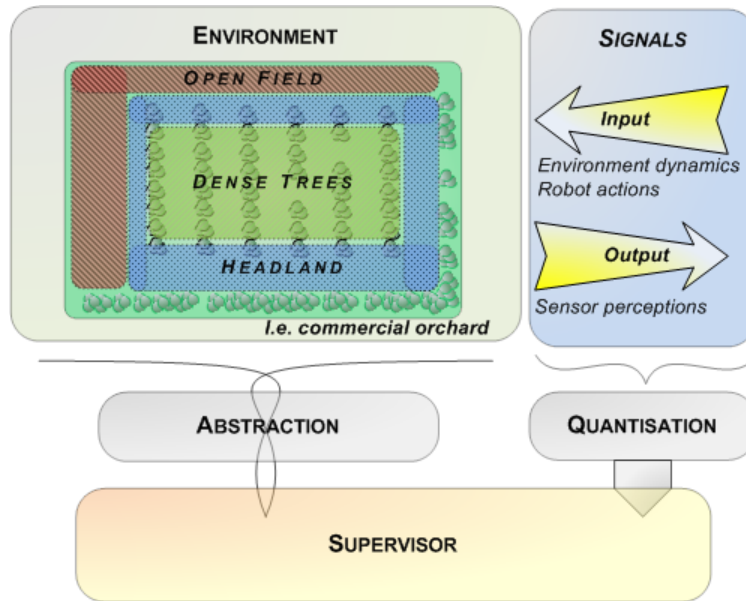


Figure 1: Environment-distinctive areas and their topological relations are modelled by states and transitions of a stochastic automaton. To identify the model probabilities, the real-valued, measured input and perceived output are quantised and are tuned through frequency count probabilities. Robust semantic localisation is achieved in real time by recursive evaluation in an observer.

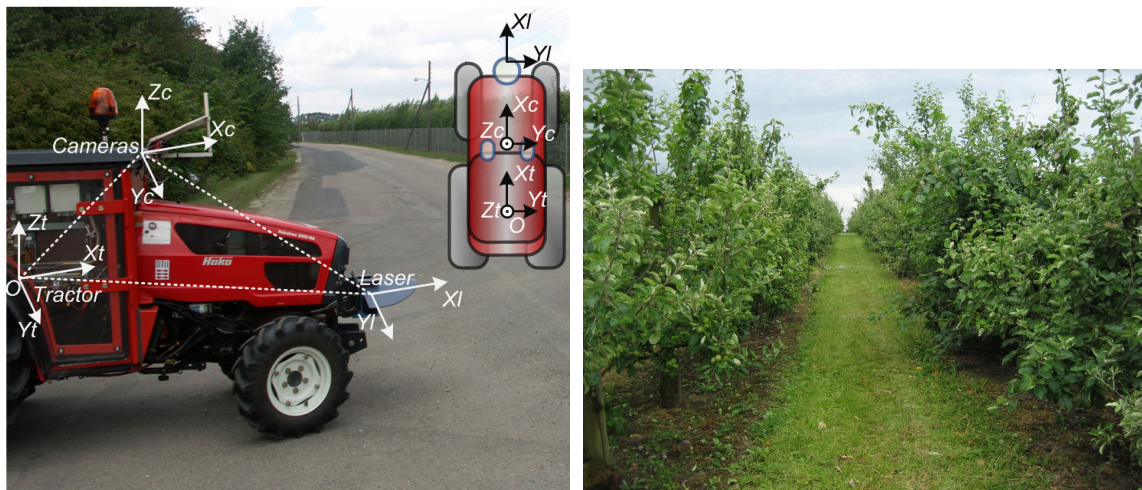
Semantic mapping provides the means to supervise navigation based on metric localisation. Typical mission plans for the tractor involve either spraying or mowing. Errors in metric localisation, and/or an operator based plan, can have consequences for the environment. Spraying, as an example, should only be activated when the autonomous vehicle is located next to the tree rows. Semantic information can create context aware supervision of the process so attempts to spray in open field or headland could be detected as faults. As spraying is potentially harmful to humans it is critical that such faults are detected. A rule-based planning system was demonstrated to work well in (Andersen et al., 2010). A system formally based on semantic mapping has a wider generality.

1.1.1 Autonomous Tractor

The autonomous robot in this application was a standard orchard tractor, build by Hako Werke, that was retrofitted with additional sensors and computing power (Griepentrog et al., 2009). A SICK[®] LMS-200 laser scanner is mounted on the robot. The scanner measures distances to objects in a two-dimensional 180° fan in front of the robot. The maximal range of the SICK scanner is 8 m in the configuration used. A stereo vision camera, which makes a 3D point cloud of the field of view, was also mounted on the robot, see Fig. (3). Standard software coming with the stereo device was used to extract the 3D points. The tractor was further equipped with an RTK-GPS which gives a position estimate with a theoretical precision of 2 cm. To provide this precision, 6 satellites or more must be in the field of view, otherwise the precision drops. Quadrature encoders that measure the wheels' rotation, a steering angle transducer and a rate gyro complete the instrumentation package. The low level interfacing to these sensors and actuators is done using the Mobotware software (Beck et al., 2010), where also the odometry based on the travelled distance and gyro data is calculated. Calibration of the odometry is done before the mission, by completing a number of manoeuvres and optimise the sensor models to best fit the



Figure 2: Google Earth® view of orchard. The numbers denote some of the zones for semantic mapping: 1. Open field (Road) 2. Headland 3. Dense trees 4. Sparse trees.



(a) Autonomous tractor and coordinate definitions

(b) Dense trees part of orchard during Summer

Figure 3: Autonomous tractor and the dense trees orchard environment.

trajectory measured by the RTK-GPS.

1.1.2 Supervision

At the highest level of abstraction the robot should try to complete its given tasks while assuring safe operation. In the case of faults it will have to work under reduced capabilities but may still be able to finish its mission. At the highest level of abstraction the role of the supervisor is largely to coordinate functionalities of lower levels to obtain the overall goals, while at the same time preventing the robot from turning local faults or signal artifacts into failure at the overall level.

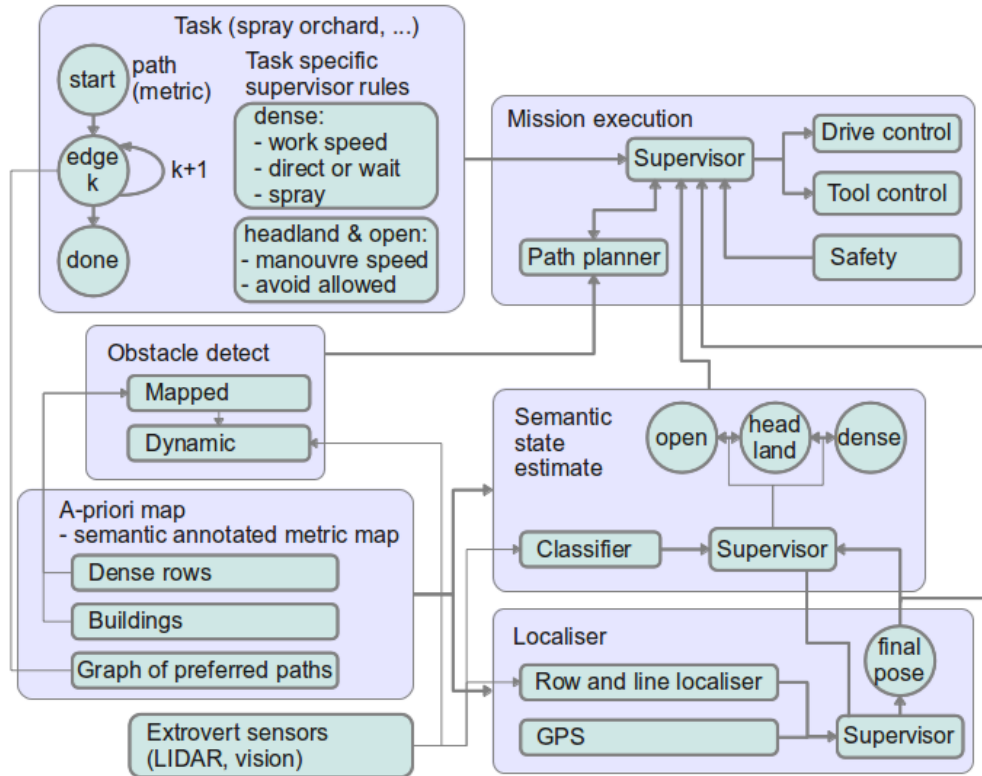


Figure 4: Overview of the architecture of the robot and how supervision is used at component levels. Semantics form an important part of the supervision process.

At lower levels of abstraction, individual modules should try to provide the information (service) needed by other parts of the robot. Fault diagnostic methods and hypothesis testing are used at the lower levels to ensure fault-tolerant operation, where full performance of a module is traded for possibly lower performance but uninterrupted output when faults occur.

The architecture implemented in the orchard application has three different levels of supervision. At the lowest level the metric localisation is based on interpreted environment features from the LIDAR, position from the GPS, turn rate from a rate gyro and odometry using wheel angle and wheel speed sensors. The instruments are supervised using industry accepted methods for achieving fault-tolerance as detailed in (Blanke et al., 2006). Localisation is detailed in Section 2. Above this level, a semantic supervisor is used to evaluate any inconsistencies between an a-priori semantic annotated metric map, the metric localisation, and the semantic mapping of inputs from extrovert sensors (laser and vision). The higher level supervisor is then used to take decisions on which operations to perform (driving, spraying, mowing, ...) based on the current metric position, current mission tasks, metric semantic supervisor and information on any detected obstacles. An overview of the architecture is given in Fig. (4). Splitting the supervision into a number of subsystems allows for individual unit tests and it also significantly reduces the complexity of the automata for supervision. This in turn makes system development simpler, modular, easier to maintain and presumably also more reliable due to reduced complexity.

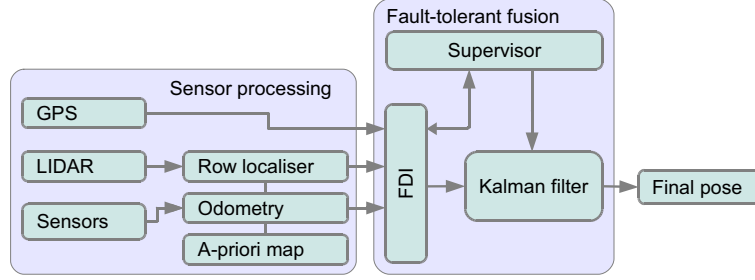


Figure 5: Fault tolerant localisation using a diagnostic module (FDI) to check validity of signals, the sensor-fusion Kalman type of filter and a local supervisor to handle faults.

1.2 Chapter Outline

The chapter is organised according to the levels of supervision. First, metric localisation is discussed and how it can be self-supervised making use of metric-based fault diagnosis. The pitfalls with metric localisation for the robot are also discussed in the context of augmenting the supervision processes with semantic mapping. Perception for semantic mapping is introduced using a stochastic automaton framework, signal space quantisation is discussed and an optimisation is suggested to minimise the false alarm probability. It is then outlined how the semantic mapping can be exploited in a supervisory architecture to enhance overall safety and reliability for the autonomous robot. Lastly, supervision of the mission execution is discussed for enhancing such supervision with output of a semantic state machine. A section showing results from autonomous operation and a conclusion finalises the chapter.

2 Supervision of Metric Localisation

The robot's pose relative to the objects in its environment is essential for carrying out its objectives. In this section the robots pose is kept in the Universal Transverse Mercator (UTM) coordinate system also used by the on-board GPS receiver. The pose is denoted $\mathbf{z} = [\mathbf{R}, \psi]^T$ where the position is \mathbf{R} and the heading angle is ψ . Metric information about position is available from sensors like GPS, \mathbf{R}_{gps} , and through odometry, \mathbf{R}_{odo} , where steering wheels' angle θ and average wheel speed δL are utilised for estimating the pose. This is done by integrating the robots velocity \mathbf{u}_g in the GPS coordinate system. The transformation matrix between the robots body coordinate system and its ground system is denoted \mathbf{A}_{gb} .

True locations of objects in the environment are represented in a map $\mathcal{M}^k(o_i)$ of known objects where o_i are the objects. The laser range finder gives information about objects in view $\mathbf{p}_{las}^{o_i}(\mathbf{z})$. There will also be unknown objects in the environment. These are denoted $\mathcal{M}^u(o_i)$.

The constraints describing the kinematics of the tractor are then given in Table 1, with δL being the translational distance driven in each time step, θ being the steering angle, f^{odo} an odometry relation and ρ_{gyro} measured turn rate and ρ_b bias on the rate measurement. The odometry relation is split up into translational $f^{odo,tran}$ and rotational $f^{odo,rot}$ parts to accommodate easy comparison with measurements from a rate gyro. Finally, by processing the range scan r from the laser-scanner in different ways, it is possible to estimate parts of the pose. The methods used in this text are discussed in more detail in section 2.1. The overall behavioural models of these processes will be denoted \mathcal{L}_1 , \mathcal{L}_2 and \mathcal{L}_3 . These different ways of estimating the pose can formally be written as a set of constraints, where c_i are physical constraints of the system and m_i the measurements.

Following the procedure for analysis of system topology properties, referred to as structural analysis,

System Constraints						Measurement constraints							
c_1	:	$\mathbf{u}_b = f^{odo,tran}(\delta L, \theta)$				m_1	:	$\Delta n_{enc} = \frac{\delta L}{\gamma}$					
c_2	:	$\mathbf{u}_g = \mathbf{A}_{gb}\mathbf{u}_b$				m_2	:	$\theta_{meas} = \theta$					
c_3	:	$\rho_b = f^{odo,rot}(\delta L, \theta)$				m_3	:	$\mathbf{R}_{gps} = \mathbf{R}$					
c_4	:	$\mathbf{R} = \int_0^t \mathbf{u}_g(\tau) d\tau + \mathbf{R}_0$				m_4	:	$\rho_{gyro} = \rho$					
c_5	:	$\Psi = \int_0^t \rho_b(\tau) d\tau + \Psi_0$				m_5	:	$\mathbf{R}_{m1} = \mathcal{L}_1(r, \mathbf{R}, \Psi, \mathcal{M}^k(o_i))$					
c_6	:	$r = \mathbf{g}_3(R, \Psi, \mathcal{M}^k(o_i))$				m_6	:	$\mathbf{R}_{m2} = \mathcal{L}_2(r, \mathbf{R}, \Psi, \mathcal{M}^k(o_i))$					
						m_7	:	$\Psi_m = \mathcal{L}_3(r, \mathbf{R}, \Psi, \mathcal{M}^k(o_i))$					

Table 1: Constraints that define the FDI problem for localisation

	c_1	c_2	c_3	c_4	c_5	c_6	m_1	m_2	m_3	m_4	m_5	m_6	m_7
r_1	:	1	1	1	1		1	1	1	1			
r_2	:		1				1	1		1			
r_3	:				1	1			1	1	1		
r_4	:				1	1			1	1		1	
r_5	:				1	1			1	1			1
<i>detectable</i>		d_1	d_1	i	d_1	d_2	i	d_3	d_3	d_2	i	i	i
<i>isolable</i>													

Table 2: Residuals' dependency on constraints

see (Blanke et al., 2006) and references herein for details, the set of known variables in Table 1 are $\mathcal{K} = \{\mathbf{R}_0, \Psi_0, \mathcal{M}^k(o_i)\}$ and the set of unknowns, $\mathcal{Z} = \{\delta L, \theta, \mathbf{R}, \Psi, \mathbf{u}_b, \mathbf{u}_g, \rho_b, r\}$. In this setting the robots starting pose \mathbf{R}_0, Ψ_0 and the map \mathcal{M} are considered known. Analytic redundancies are obtained by matching of a structure graph resulting from the constraints in Table 1 and these redundancies form the residuals used for diagnosis. With 13 constraints and 8 unknowns, 5 constraints become available for sensor fusion, diagnosis and / or supervision of the metric localisation. Despite describing normal behaviours, constraints m_5 to m_7 are dependent on the objects in view. In certain areas of the orchard, in between rows of trees, individual objects can not always be distinguished, trees need be grouped as rows etc. Hence, a higher level of abstraction is needed to supplement metric localisation. This abstraction level is semantic maps and there is a dependency between the semantic state and the computations for FDI within the localisation module and its supervisor.

Using the system equations listed above between three and five parity relations can be created in different areas of the orchard. The parity relations use different constraints for their evaluation. This is listed in Table 2, which shows that faults related to constraints c_3, c_6, m_4, m_5, m_6 and m_7 are structurally isolable, indicated by i in the last row, while others are group-wise detectable, denoted d_1, d_2, d_3 in the table.

An example of the residual during a turn from one row to the next is shown in Fig. (6). The Figure shows two passages: the solid lines are the development of $r_2 - r_4$ for a fault-free case and the dashed lines for a case where an artifact occurs on the GPS position. The fault is not affecting r_3 .

2.1 Localisation Based on Extended Kalman Filter

An orchard is a semi-structured natural environment, where fruit trees are planted in straight rows, and headland is available to drive the tractor to consecutive rows. Hence, the map \mathcal{M}^k is easily defined. Therefore, using distance measurements between the robot and the tree rows for localisation is obvious when the robot is in between rows.

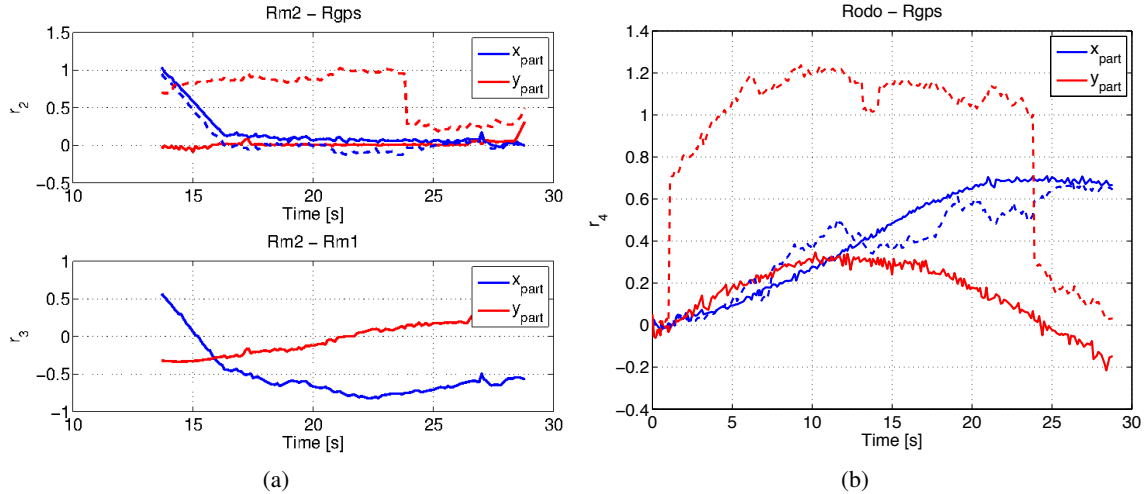


Figure 6: Residuals formed from the structural analysis of the system constraints in c_1 to m_7 .

In the headland, the robot must drive in a particular pattern due to physical limitations in turning capability. It must skip at least one row, and reversing need be avoided when a fungicide sprayer is in a trailer behind the tractor.

The task of keeping track of the robots pose as it executes its mission in the orchard, is not as easy as it might appear. Because of the issues with sensor dependency on objects in view, supervision is needed. An example of the different sensors pose estimates is shown in Fig. (7).

The high precision RTK-GPS gives good position indications when a clear view to GPS satellites is present, but when driving between fruit trees, which cover the path to satellites with their foliage, erroneous behaviour is often experienced (Hansen et al., 2009). This is illustrated in Fig. (7).

2.2 Fault-tolerant Sensor Fusion

With redundant information available, gross artifacts or faults in signals from one sensor are commonly dealt with by sensor fusion and excluding sensors with detected faults. In an orchard, 2D laser scans can be used to aid the GPS in estimating the robot's pose utilising the prior knowledge about of the structure of this environment. The laser scanner gives a field of distance measurements in a 180° fan in front of the robot. The scanner is mounted as seen in Fig. (3). Using a feature extracting algorithm on these data, lines representing the orchard rows are created.

2.2.1 Row Feature Extraction Algorithm

The feature extraction is relatively simple because of the sparse data coming from the laser scanner. Each set of range data points $r_i = [x_i, y_i]^T$, is transformed into the GPS coordinate system based on the tractor's pose estimate.

$$q_i = \mathbf{A}_{gb} p_i \quad (1)$$

The range data are then grouped into groups Q_j according to there proximity to the mapped tree rows o_j from the known map $\mathcal{M}^k(o_j)$.

$$Q_j = \{q_j \mid \text{dist}(o_j, q_i) < D\} \quad (2)$$

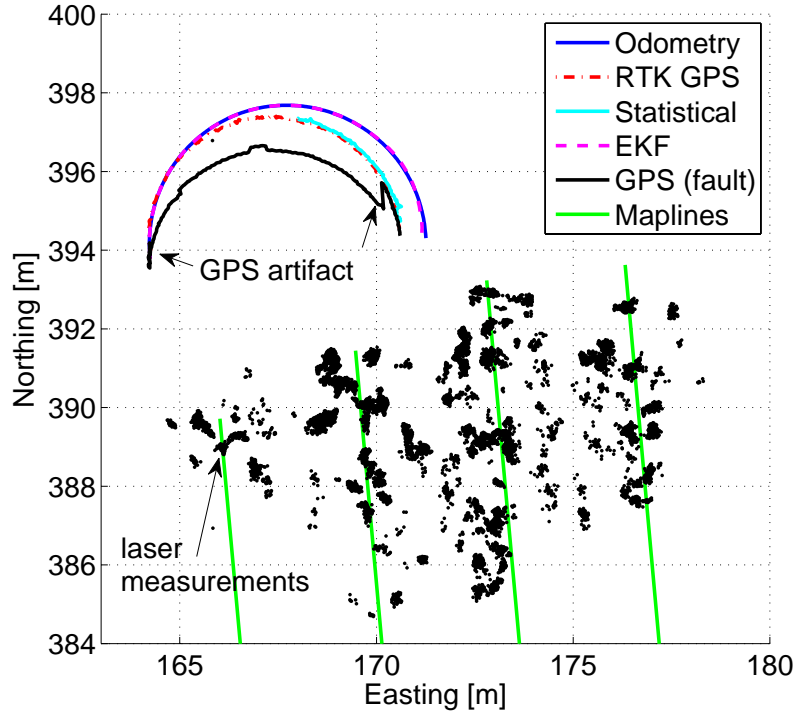


Figure 7: Trajectory of robot in a headland turn given by the GPS and the most likely position according to a statistical estimate. The robot moves clockwise from the leftmost row to the middle one. A case where the RTK-GPS has temporal faults is also plotted.

A straight line of the description $A_jx + B_jy + C_j = 0$ of each Q_j is then estimated using least squares estimation.

2.2.2 Robot Pose Estimation

The estimated lines are then compared with regard to distance and angle from the robot pose to the mapped lines and this difference is used as innovation in the filter. A well known way to unify this information is the extended Kalman filter (EKF). The state information is the robots pose, \mathbf{z} , which is predicted by feeding the wheel encoder measurements to a model of the robots steering system, c_1, c_2 in Table 1,

$$\mathbf{z}_{k+1} = \mathbf{f}(\mathbf{z}_k, \mathbf{u}_k, \boldsymbol{\varepsilon}_k) \quad (3)$$

where $\mathbf{u} = [v, \alpha]^T$ is input and $\boldsymbol{\varepsilon}$ process noise. Using the 2D laser scanner information to update the pose by comparing the extracted lines to the map, constraint m_1 . With measurement noise $\boldsymbol{\varepsilon}$,

$$\mathbf{z}_k = \mathbf{h}(\mathbf{p}_{las}^{o_i}(\mathbf{z}), r_m, \boldsymbol{\varepsilon}_k) \quad (4)$$

Both process and measurement noises \mathbf{w}_k and \mathbf{v}_k are assumed white and Gaussian in the Kalman filter formulation. Deviations from this assumption can make the Kalman filter less accurate. The sometimes crude linearisation done by the EKF can have quite negative effects and solutions using higher order derivative free filters have been tested on this orchard case, see (Hansen et al., 2011).

2.2.3 Sensor Fusion

The localisation solution with extended Kalman filter in Eqs. (3) and (4) alone has a shortcoming in headland manoeuvres. When the orchard rows are not visible from the laser scanner, no updates arrive to the pose estimate Eq. (4), which then solely relies on odometry. This makes it drift because of wheel slippage, especially while turning. This is seen in Fig. (7) where GPS artifacts are also apparent when passing from the foliage GPS signal shadows to full satellite visibility in the headland.

The constraints $c_1 - m_7$ provide several complete matchings of the unknown variables. Unmatched constraints of one matching used by the Kalman filter for pose estimation would leave information that should be evaluated for correctness. One intuitive result is that the pose should be validated through the vision system,

$$\mathbf{z}_k^{(2)} = \mathbf{h}(\mathbf{p}_{vis}^{o_i}(\mathbf{z}), r_m, \epsilon_k) \quad (5)$$

The structural analysis approach result in a complete fault diagnosis scheme for validating the estimates of the pose. The pose is useful for real-time driving and track control but it does not possess sufficient information when upper-level fault handling and re-planning are needed.

The use of an EKF filter solution is common in robotics but since it is relying on fault-free measurements, may have accuracy issues with wheel slippage or field's slope, that has not been modelled, overall supervision of performance is needed to enhance reliability in an autonomous operation. When different algorithms need to be employed in different sectors of the environment, it is natural to let the supervisor comprise a model that attempts to recognise each sector where the robot operates and utilise this knowledge in its supervision of correct performance and function. Semantic maps are well suited for abstraction for this purpose and supplementary to the metric localisation.

2.3 Supervisor for the Localisation Module

The supervisor for the localisation must be able to operate with temporal faults or artifacts occurring at random instants and in random sequence. The localiser should be able to operate under conditions of multiple simultaneous faults, continuing to provide a pose estimate, or at least a position, perhaps down to last sensor available.

The supervisor needs to discard sensors isolated as *faulty* (\mathcal{H}_1) from the sensor fusion filter and include a sensor again when the hypothesis *normal* (\mathcal{H}_0) has been confirmed. Events e_i are either $e_i : \mathcal{H}_1 \rightarrow \mathcal{H}_0$ or $\neg e_i : \mathcal{H}_1 \rightarrow \mathcal{H}_0$.

The set of residual generators that need be employed after one fault has occurred is generally different from the set used for the non-faulty system. Given that one behaviour is not-normal, the associated constraint need be removed from the analysis of structure, and the graph-matching and back-tracing be done along the matched path of the graph to generate the parity relations valid for the particular case. The supervisor needs to handle this complexity.

2.3.1 Automaton for state supervision

An automaton has the following form, where $e_k = \{v_k, w_k\} \in \mathcal{E}$ denote event k , $s_k \in \mathcal{S}$ is the state when this event happens and s_{k+1} is the consecutive state, $\{v_k, w_k\}$ are the input and output associated with the event. With \mathcal{A} being a deterministic supervision automaton, which is here distinguished from the stochastic diagnostic automaton in Eq. (22) and $u \in \mathcal{U}$ the action performed by the supervisor,

$$\begin{aligned} s_{k+1} &= \mathcal{A}(e_k, s_k) \\ u_k &= \mathcal{U}(s_k) \end{aligned} \quad (6)$$

The plane architecture in Eq. (6) easily leads to considerable combinational complexity when both normal and not normal operation need be dealt with.

An implementation as parallel automata (Blanke et al., 1997) gives the possibility of a significantly reduced complexity. Following a notation derived from (Izadi-Zamanabadi, 1999), let a hierarchy of automata consist of an automaton in each layer that is coupled to other layers and in particular is coupled with the semantic state s_{sem} . The associated, layered automaton,

$$\begin{aligned} s_{k+1}^{ij} &= \mathcal{A}_{\mathbb{S}^{\square}}(e_k, s_k^i, s_{sem}^j) \\ u_k^{ij} &= \mathcal{U}(s_k^{ij}) \end{aligned} \quad (7)$$

usually has much less states and significantly less state transitions compared to Eq. (6).

2.3.2 Localisation supervision

The supervisor automaton consists of states s_k , $s \in \mathcal{S}$, each of which represent the combination of sensors κ available, among m sensors, $s_k = \{\kappa_m, \kappa_{m-1}, \dots, \kappa_1\}$ and is formally described by the sets of states \mathcal{S} and actions \mathcal{U} , leaving the supervisor automaton formally defined by,

$$\begin{aligned} \text{Let : } \kappa_l &\equiv \begin{cases} 1 & \text{sensor}_l = \text{valid} \\ 0 & \text{sensor}_l \neq \text{valid} \end{cases} \quad \forall l \\ s_k^i &= \{\kappa_m, \kappa_{m-1}, \dots, \kappa_1\} \\ \mathcal{A} : s_{k+1}^{ij} &\leftarrow \mathcal{A}(e_k, s_{k-1}^i, s_{sem}^j) \\ \mathcal{U} : \{\mathbf{r}^{ij} &= \pi(s_k^{ij}, s_{sem}^j), \mathbf{K}(s_k^i, s_{sem}^j)\} \end{aligned} \quad (8)$$

where $\pi(s_k)$ is the parity vector available from analysis of the system of constraints corresponding to available sensors and $\mathbf{K}(s_k)$ is the fusion filter used in this state. The parity vector is generated from the list of parity relations $\pi(s_k)$ obtained automatically from the structure graph matching and backtracking. The overall diagnostic procedure is hence provable correct for residual generation and for the actions taken when faults occur. It is noted that the supervisor Eq. (8) depends on upper level information because some calculations are dependent on the semantic state.

3 Supervision of Semantic Mapping

Previous work on semantic mapping focused on online building of semantic maps using perceptual information. Starting from classification techniques based on boosting algorithms (Mozos et al., 2005), several approaches were proposed in literature to improve performances by taking advantage of object recognition (Nuchter et al., 2005) and probabilistic environment models (Mozos et al., 2007). Separation of semantic maps into spatial and conceptual hierarchies was done for indoor environments in (Galindo et al., 2005a). The present approach uses a non-hierarchical approach for the mission supervision, which makes it possible to utilise the intuitively appealing modelling of semantics as states in an automaton. Staying in a state or shift to another is determined by classification of the input received from available sensors and characterisation of sensor input as confirming the presence in a particular state or decision that a change has happened, and the latter event should trigger a change of state. An equivalent functionality as that of a hierarchy is obtained using different supervisors, and hence automata, for each of the main modules.

3.1 Semantic State

Consider a semantic state as a geographical area, or context, which the tractor must recognise. Referring to Fig. (2), a typical orchard can be divided into five interconnected zones, enumerated and described as follows:

1. *Open field* Few obstacles and freely traversable space in field of view of the tractor, i.e. a road or a low vegetation field.
2. *Headland* Defines the start/end of the agricultural area. Delimited by fences, markers or open space.
3. *Dense trees* Distance between trees is tight. Limited manoeuvring possibilities. Robot may need to push through branches to get by.
4. *Sparse trees* Sparse vegetation, manoeuvre possible between the trees without physical contact.
5. *End of row* End of row of trees.

3.2 Indicators used for Semantic Mapping

The semantic states in the orchard example has $N = 5$ states,

$$\mathcal{N}_c = \{\text{Open field, Headland, Dense trees, Sparse trees, End of Row}\} = \{1, 2, 3, 4, 5\}. \quad (9)$$

When dealing with a complex natural environment, laser range scanner and stereo vision information need be processed into a form easily digestible for perception. The indicators used for semantic mapping in the orchard were those listed above. The algorithms to extract the indicators from signals are listed below, adopted from (Caponetti et al., 2011). These were designed to be computationally efficient for real-time application. More expensive probabilistic models could also be used to take into account variations in the sensor readings. Instead the stochastic automaton is later used to compensate for this by learning the signal distributions. With the onboard processing powers of the tractor, these signals can be generated at approximately 4Hz.

3.2.1 Algorithm for Ground Plane

A signal y_{gp} representing the amount of ground plane visible is extracted using a method similar to those shown in (Konolige et al., 2009). Given a 3D point cloud a RANSAC technique (Fischler & Bolles, 1981) is used to construct ground plane hypotheses. This is done by: (a) choosing three non-collinear points at random from the point cloud; (b) constructing a plane estimate from the three points; (c) ranking the plane estimates based on number of inliers.

A 2D grid map is then constructed by projecting the 3D point cloud to the ground plane. The sum of grid cells that get assigned to the ground plane are summed to give the amount of ground plane visible.

Let the found ground plane be written in the Hessian normal form of a plane with unit normal vector $\hat{\mathbf{n}}$ and distance p along the line. Let \mathcal{P} be the set of points in the point cloud considered. If a point falls within a grid cell c_{xy} , defined by a quantisation interval Q_{xy} , Eq. (10) and its distance to the plane is less than D_{\max} then the cell belongs to the ground. For $\mathbf{x} \in \mathcal{P}$,

$$c_{x,y} = \left| \left\{ \mathbf{x} \mid \mathbf{x}_x \in Q_{xy} \wedge \hat{\mathbf{n}} \cdot \mathbf{x} + p < D_{\max} \right\} \right| > 0 \quad (10)$$

$$y_{gp} = \sum_{x=n_0}^{n_1} \sum_{y=m_0}^{m_1} c_{x,y}. \quad (11)$$

3.2.2 Algorithm for Laser Free Space

The free space y_{fs} observed by the laser is taken as the area spanned by the measurements. Given two adjacent range measurements s_i, s_{i+1} $i = 1, \dots$ a triangle is formed and its area can be easily evaluated using Heron's formula,

$$d_i = \|l_i - l_{i+1}\|_2 \quad (12)$$

$$s_i = \frac{l_i + l_{i+1} + d_i}{2} \quad (13)$$

$$A_i = \sqrt{s_i(s-l_i)(s-l_{i+1})(s-d_i)}. \quad (14)$$

Having a single laser scan, by summing the area of each triangle defined by adjacent readings, an estimate of the free space is,

$$y_{fs} = \sum_{i=0}^{n_l-1} A_i. \quad (15)$$

3.2.3 Algorithm for Obstacles

An obstacle signal y_o is constructed by creating a 3D grid map. Each point from the stereo vision as well as the laser scanner are projected into a grid map. A grid cell is then labelled as occupied if a stereo point or laser point lands in it.

A 3D grid cell c_{xyz} is given the value of 1 if a point falls inside the cell and its height above the ground plane is larger than D_{\max} . The number of occupied grid cells is then counted and used as a measure of the amount of obstacles seen from the pose,

$$c_{xyz} = \left| \left\{ \mathbf{x} \mid \mathbf{x} \in Q_{xyz} \wedge \hat{\mathbf{n}} \cdot \mathbf{x} + p \geq D_{\max} \right\} \right| > 0 \quad (16)$$

$$y_o = \sum_{x=n_0}^{n_1} \sum_{y=m_0}^{m_1} \sum_{z=k_0}^{k_1} c_{xyz}. \quad (17)$$

3.2.4 Algorithm for Detecting Linear Structures

A signal y_{ls} indicates the presence of linear structures in the environment such as walls, fences, or hedges. The 3D grid map of obstacles is now collapsed into a 2D grid map on the ground plane by summing the number of occupied cells along the vertical component, see Fig. (8).

A RANSAC line-fitting algorithm is then run on the 2D grid map to extract the strongest line. With $g_{x,y}$ denoting 2D grid map cells formed by summing data along the z dimension of the 3D grid map used in calculating the obstacles signal,

$$g_{x,y} = \sum_{z=k_0}^{k_1} c_{x,y,z}. \quad (18)$$

A function $l(a,b)$ returns the grid cells that intersect with the line parameterised by a and b ,

$$l(a,b) = \{j \mid j \in g_{x,y}, j_y = aj_x + b\}. \quad (19)$$

The RANSAC algorithm finally attempts to maximise the sum of grid cells that intersect with the line,

$$y_{ls} = \max_{a,b} \sum_{i=1}^{|w|} l(a,b). \quad (20)$$

3.2.5 Algorithm for Detecting Row End

The row end detection is based on the intensity of laser scanner reflections per area unit. The detection density $\rho(\theta_i)$ is calculated as number of detections in the 16 cm row slice θ_i along the tree row (maplines in Fig. 7) and within the laser scanner coverage of the row, up to the scanners max range, M in Eq. (21). Change in mean density was done in (Andersen et al., 2010) using a generalised likelihood ratio like test on the log-likelihood ratio of a change in density down to zero, under Gaussian assumption,

$$k = \arg \max_{1 < k < M} \sum_{i=0}^k \left(\frac{\mu_{row}}{\sigma_z^2} (\rho(\theta_i) - \mu_{row}) \right) \quad (21)$$

where k is the index pointing to the end of row distance, μ_{row} is the test quantity for change in mean and σ_z^2 the variance of $\rho(\theta_i)$ along the row.

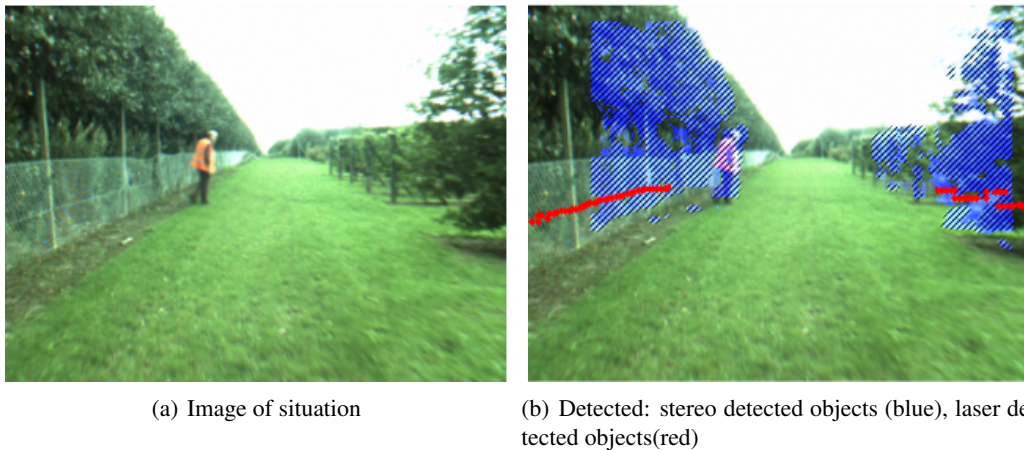


Figure 8: Detection of obstacles by stereo and laser in headland.

3.3 Validation of Indicator Algorithms in a Natural Environment

The validation data-set was recorded during a run which covered the track shown in Fig. (2). The run was made in summer time to catch one extremum of the scenario. Full grown foliage, bushes and tree branches hanging increased the variability of each zone. To stress more the robustness of the methods, people were moving or standing in the tractor field of view during the data recording. Collecting the signals described in Sec.3.2, the observed output consist of a vector $\mathbf{y} = [y_{gp}, y_{fs}, y_{ls}, y_o, y_{vp}]$ of continuous signals. The procedure introduced in section 3.5.1 was then used (Caponetti et al., 2011) to design quantisers for the perceptual data. A training set was used to determine distributions in different areas of the orchard. The state-conditioned probability distributions were obtained by kernel density estimation and shown in Fig. (9). Perceptual aliasing is recognisable in the probability space as an overlap of the distribution curves. This problem was handled by creating quantisation levels for each region of interest and by combining all the signals in the automaton and using the state transition model.

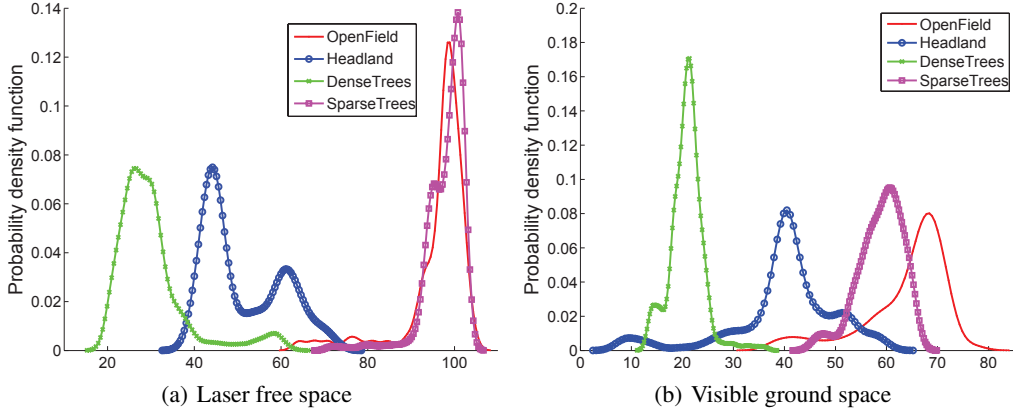


Figure 9: State conditional probability density functions for two features.

3.4 Stochastic Automaton

A robot operating in an environment can be modelled as a hybrid system in which the states are given by the situations or places that it could experience, a discrete-event system subject to input that evolves through states, which generate output. One input is actuation of the robot causing it to potentially move from one environment to another. Other input would be time, which affects the season and cause the environment to change. Measured output are signals from the robots perception system that allows it to observe a change of state.

Raw sensor signals from extrovert sensors (LIDAR and vision) are real-valued and in order to be fed to the model, a discretisation has to be performed, (Schroder, 2003; Supavatanakul, 2004).

Suppose that all signals are discrete and let events be changes in the discrete signals values. Let the states be a set of semantic locations, which it is desired that the robot can distinguish between. Let the system discrete input be $v \in \mathcal{N}_v \subset \mathbb{Q}$, $\mathcal{N}_v = \{1, 2, \dots, M\}$, state $z \in \mathcal{N}_z \subset \mathbb{Q}$, $\mathcal{N}_z = \{1, 2, \dots, N\}$ and output $w \in \mathcal{N}_w \subset \mathbb{Q}$, $\mathcal{N}_w = \{1, 2, \dots, R\}$ where M , N and R are finite.

The automaton behaviour is timed by the successive events but like all probability based models, a stochastic automaton, at time k , is not in a unique state z_k but has a probability of being within a set of possible states. This is described by a discrete probability distribution $P(z_k)$,

$$P(z_k) = \{P(z_{k=1}), P(z_{k=2}), \dots, P(z_{k=N})\}.$$

The probability distribution $P(z_k)$ is a function that associates to all the states $z \in \mathcal{N}_z$ the probability which the automaton assumes at time k .

Using the notation of (Schroder, 2003), an initialised stochastic automaton \mathcal{S} for diagnosis is fully described by the *five – tuple*:

$$\mathcal{S} = \langle \mathcal{N}_z, \mathcal{N}_v, \mathcal{N}_w, L, P(z_k) \rangle \quad (22)$$

L is a behavioural function, the law that governs the stochastic process underlying the automaton. The behavioural function is defined as,

$$L : \mathcal{N}_z \times \mathcal{N}_w \times \mathcal{N}_z \times \mathcal{N}_v \rightarrow [0, 1] \subset \mathbb{R} \quad (23)$$

$$L(z', w, z, v) = P(z_{k+1} = z', w_k = w | z_k = z, v_k = v), \quad (24)$$

which has the properties

$$0 \leq L(z', w|z, v) \leq 1, \forall z', z \in \mathcal{N}_z, v \in \mathcal{N}_v, w \in \mathcal{N}_w \quad (25)$$

$$\sum_{z' \in \mathcal{N}_z} \sum_{w \in \mathcal{N}_w} L(z', w|z, v) = 1, \forall z \in \mathcal{N}_z, v \in \mathcal{N}_v. \quad (26)$$

Fig. (10) shows a simplified automaton with five states, equivalent to the semantic states defined earlier. Transitions between states are triggered by probabilities, Eq. (24). Transition probabilities are abbreviated L_{hd} for transition from headland to a row of dense trees, L_{dr} from dense trees to row end, L_{rh} from row end to headland, L_{sr} from sparse trees to row end, L_{oh} from open field to headland, etc.

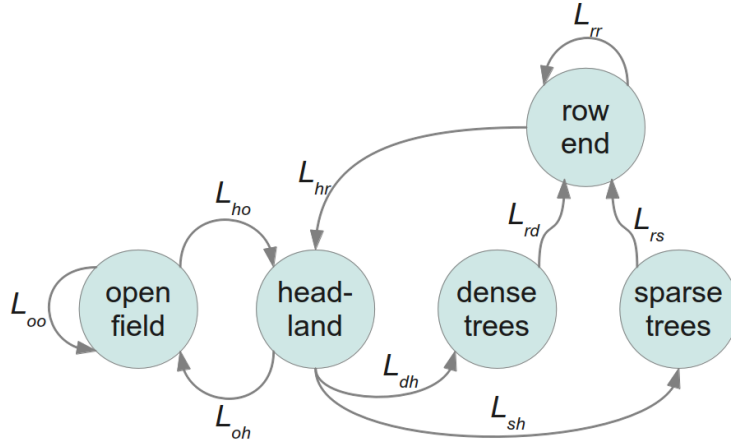


Figure 10: Simplified illustration of stochastic automaton. Behavioural relations L in the form of Eq. (23) are functions of observed features. Probabilities determine change of state but this is not shown in the figure.

3.5 Classification

Being able to classify the current state among a set of possible choices using the local available information is equivalent to solve an observation problem for the corresponding stochastic automaton. Defining the input and output sequences till the current time sample k_h ,

$$\text{input sequence: } V(0..k) = (v_0, v_1, ..v_k)$$

$$\text{output sequence: } W(0..k) = (w_0, w_1, ..w_k),$$

the state observation problem can be formulated as the situation in which the I/O pair sequence

$$\{V(0..k), W(0..k)\}$$

is known up to present time k and the current state z_k of the stochastic automaton is to be determined.

Given an input and output sequence and an initialised automaton \mathcal{S} in the form of definition 22, the solution to the observation problem is obtained by determining the conditional probability distribution (Blanke et al., 2006),

$$P(z_k|k) = P(z_k|V(0..k_h), W(0..k)). \quad (27)$$

The solution of the observation problem is in general not unique and given by the set

$$\mathcal{Z}(k|k) = \{z_k : P(z_k|V(0\dots k), W(0\dots k)) > 0\} \quad (28)$$

which includes all states z_k to which the automaton may move with non zero probability while accepting the input sequence V and generating the output sequence W .

Considering a stochastic automaton with the initial state distribution $P(z_0)$, if the I/O pair (V, W) is consistent with the stochastic automaton, the a-posteriori state probability distribution is obtained by recursive application of Eqs. (29) and (30), (Schroder, 2003; Blanke et al., 2006):

$$P(z_k|k) = \frac{\sum_{z_{k+1}} L(k)P(z_k|k-1)}{\sum_{z_k, z_{k+1}} L(k)P(z_k|k-1)}, \quad (29)$$

where $k_h \geq 0$,

$$P(z_k|k-1) = \frac{\sum_{z_{k-1}} L(k-1)P(z_{k-1}|k-2)}{\sum_{z_k, z_{k-1}} L(k-1)P(z_{k-1}|k-2)}$$

$$P(z_0|-1) ::= P(z(0)), \quad (30)$$

and input v_k and output w_k are arguments in

$$L(k) = L(z_{k+1}, w_k|z_k, v_k).$$

Eq. (29) describes how the prediction is obtained from the previous time point, Eq. (30), have to be corrected after the new measurements $v(k)$ and $w(k)$ become available.

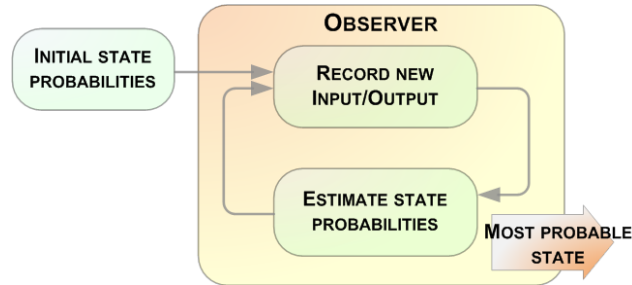


Figure 11: Given the current discrete value input output couple, the probabilities of the states can be estimated recursively by prediction and correction. The current state is then chosen as the one with the highest probability among the possible states.

3.5.1 Quantisation of Signal Spaces

Modelling through a stochastic automaton requires that all signals are discrete variables in a finite set. When input are continuous-variables, a quantisation procedure is needed and quantisation is also very useful in terms of reducing the computational efforts. Quantisation of the extrovert signals in the orchard was dealt with (Caponetti et al., 2011) where an optimised quantisation procedure was suggested that would minimise the false detection probability for semantic classification. This procedure is an essential element in obtaining a fast computational procedure and a reliable estimate of the semantic state.

3.5.2 Training of classifier

Regarding semantic map building for mobile robots, it is common to have a set of already classified examples from which a model can be built (Mozos et al., 2005; Nuchter et al., 2005; Triebel et al., 2008). Such examples can be automatically recorded by the robot in a first tour in an environment or it can be build gradually from previous exploration. Such learning procedure is also feasible for abstracting a model and to define and tune quantisers. The training procedure adopted for semantic classification was detailed in (Caponetti et al., 2011). Here, N samples are drawn independently from the training data set according to a uniform probability distribution to define the set $\gamma = \{x_i, i = 1 \dots N\}$. A label $\theta \in \mathcal{N}_c$ is associated with each sample. After sampling, the training data consist of a finite sequence of independent pairs $(x_i, \theta_i), i = 1, \dots, N$ for the individual features. The training probabilities are then quantised and merged and a procedure ensures that the conditional probabilities of the merged distributions have a manageable low number of levels. A maximisation of the reconstruction performance was also achieved (Bowman & Azzalini, 1997).

It is possible to use these results to estimate $\hat{F}(x|\theta = z)$, where z is the semantic state, for each qualitative state in \mathcal{N}_c . This procedure makes it computationally efficient to sample signals of very complex nature in order to obtain a very significant data reduction while choosing the classification such that a best possible misclassification rate is obtained. The performance of this and other classification algorithms was also evaluated in (Caponetti et al., 2011).

3.6 Supervision of the Semantic State Estimator Module

The use of the a-priori semantic annotated metric map allows us to reduce complexity of operations in the pose estimation by selecting relevant parts of the map when calculating the likely pose from Eqs. (4) and (5) to make it faster and more reliable to extract information from relevant parts of the metric localisation.

The output of the stochastic automaton can be readily compared to the a-priori map to identify faults in either the semantic mapping, localisation, or a-priori map. Ideally, the localisation should be fairly fault-tolerant considering its own supervisor but this may not be the case (e.g. under multiple sensor faults). Given that a fault is detected, this information is used to switch off the row and line localiser (to avoid incorrect input to the sensor fusion module) as well as the mission execution supervisor. The mission execution will then switch off or reduce services that can negatively affect safe mission behaviour under the knowledge that important system components are malfunctioning.

4 Supervision of Mission Execution

At the highest level of abstraction the robot should try to complete its given tasks while assuring safe operation. In the case of faults it will have to work under reduced capabilities but may still be able to finish its mission.

Under normal operation the robot has a set of tasks that it must execute and has a number of services that define what are allowed actions while trying to complete a specific step of the plan. Planning and tasks to conduct are defined using abstractions, and command primitives can activate different services.

In the case a fault is detected in the semantic supervisor, the mission execution supervisor will switch to a reduced set of allowed services it may use. For example, in the case of a dense orchard being detected as a sparse orchard it is still ok to allow spraying. However, if the a-priori map says we are in the dense orchard and the sensors say we are in the headland then we are not allowed to spray for safety reasons.

The mission supervisor uses a short term destination pose some 10-15 m ahead along the current pose to ask the task planner for an obstacle avoidance route to the short term destination pose, if an obstacle is detected. This results in the possible states: With no obstacles: a direct path or no path; with obstacles: an obstacle

avoidance path or no path. The combination of the semantic state and the task specific state determines additional parameters for the behaviour: maximum speed, obstacle behaviour and whether a particular tool should be engaged. Route-replanning was done using methods suggested by (Choset et al., 2005). A final set of rules combines the navigation behaviour and sends the appropriate commands to the real-time drive control layer - specifying the behaviour for the next 0.5-1m.

The experience is that implementation of the supervisor and action tables as a rule-based supervisory control, such as reported in (Andersen et al., 2010), is indeed a feasible way to implement supervision. It is flexible when rules are interpreted and re-compilation can be avoided when rule changes are made. However, the complexity of this approach is that rules need to alter other rules depending on the context and events, and the implementation reliability is at risk both during implementation and later, when changes are made. The logics and the actions to be taken are the same for the rule-based implementation as with an automaton, but it appears easier to specify, implement and test parallel automata with the tools that are readily available from computer science. Automata can and should be implemented as data, such that logics can be changed without re-compilation, which is possible if the state transition and action matrices are data instead of hard-coded statements. The problems of diagnosis of defects related to specification, software implementation and robot functionality are essential issues that were discussed in (Chandrababu & Christensen, 2009).

5 Results

The autonomous tractor, the localisation algorithms and supervision were implemented and tested in a cherry tree orchard. Fig. (12) shows an autonomous passage of the orchard. Colours in the picture are: blue is driven path, purple is mapped preferred path graph and red lines are the tree rows. Fig. (12) shows a detail of an autonomous mission in the cherry orchard. The rows with cherry trees are marked in the a-priori map to assist the localisation, the cherry area is framed by lines, marking the area where cherry treatment is allowed. The actual path follows the mapped preferred rules, and the semantic state - headland or dense trees - changes the behaviour. In the headland obstacle avoidance is allowed, and higher speed is allowed. In the dense orchard no obstacle avoidance manoeuvres are allowed (but pause for an obstacle is allowed), and speed should not exceed treatment speed.

Fig. (13) shows a situation from an autonomous mission in a dense cherry orchard. An obstacle is detected about 5 m in front of the tractor. The path planner suggests an avoidance path (red curve), but the rules for dense tree areas requires that obstacles should be avoided by pausing until the obstacle has gone, so the tractor stops and reduces engine RPM.

6 Conclusions

This chapter discussed the roles of fault-diagnostic technique and semantic mapping as means to achieve robust or even fault-tolerant performance of an autonomous robot conducting maintenance tasks in an orchard. Defining an architecture with local supervision in the main modules: metric localisation, semantic state estimation and mission execution, it was shown how supervision was conducted at each level.

At the metric localisation level, fault-diagnosis was achieved using structural analysis on the mixture of conventional instruments and a virtual instrument detecting tree rows from feature extraction from LIDAR signals. Fault-diagnosis was shown to be dependent on the semantic state, since the diagnostic relations would change according to the environment in the orchard. Automata with rules or algorithmic control actions were shown to be efficient in handling this problem.

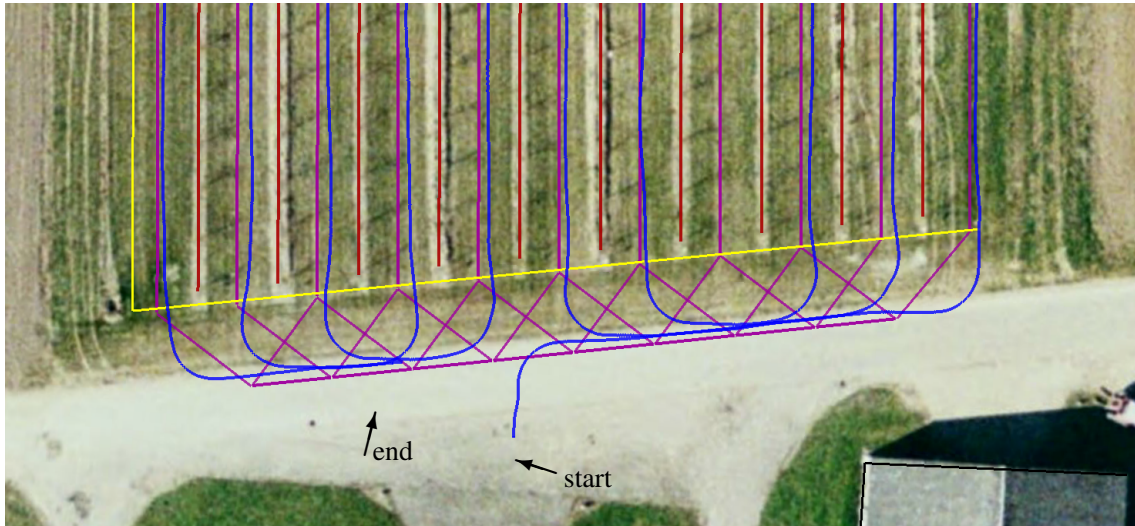


Figure 12: Part of a mission in the southern part of a cherry orchard. The tree rows are the isolated red lines, the purple lines are the preferred paths, the blue curved line is the actual path from the autonomous driven mission, the yellow square marks the extend of the cherry orchard. The background is from Google Earth® (winter image with rather young trees). Colour legend refer to the pdf version of this chapter.

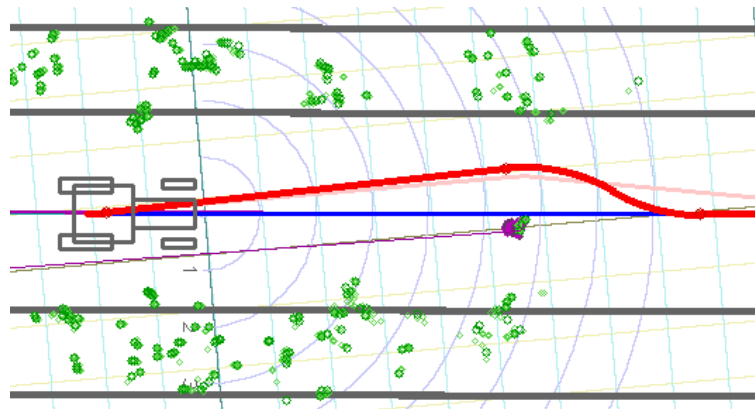


Figure 13: Zoom-in on event with obstacle avoidance re-planning. Manoeuvre is, however, avoided by the supervisor, because the semantic state is *dense trees* and a halt is instead commanded while the obstacle is in view. Small green circles are the laser scanner detection of the cherry trees, gray lines are the mapped row positions of trees. Colours refer to the pdf version of this chapter.

At the semantic mapping level, feature extraction from exogenous instruments (laser range scanner and stereo vision) was shown to provide information that could estimate a semantic state. A stochastic automaton was suggested for this level to estimate the semantic state. Training was used to obtain an a priori model of the statistical distributions one should observe in each semantic state. Quantisation of signal spaces was an essential ingredient to obtain an efficient implementation.

At the mission execution level, supervision was shown to be essential to avoid consequences of erroneous

pieces of information, originating from artifacts in signals, instrument drop outs or from errors in maps or in the prior planning. The supervision at this stage was very dependent on the semantic state estimated, as this determined the feasibility of overall actions to be taken when obstacles were met or in case of unexpected events. Results from fully autonomous driving through the orchard completed the chapter.

The approach presented is indeed applicable to a wide range of supervision tasks, also beyond the agricultural domain. Semantic mapping for fault diagnosis is on-going research and shows great promise in a variety of robotic systems.

7 Acknowledgements

This research was supported by the Danish Ministry of Food Agriculture and Fisheries, contract 3412-06-01729, by the Danish Ministry of Science, Technology and Innovation and by CLAAS Agrosystems. Our colleagues from Copenhagen University, Department of Life Sciences, Mr. J. Resting-Jeppesen and Dr. H. W. Griepentrog (now Professor at University of Hohenheim) are gratefully acknowledged for allowing access to equipment and data. Hako Werke is acknowledged for providing the tractor used for the orchard experiments.

References

- Andersen, J. C., Ravn, O., & Andersen, N. A. (2010). Autonomous rule-based navigation in orchards. In *Proc. IFAC Symposium on Intelligent Autonomous Vehicles*.
- Avizienis, A., Laprie, J., & Randell, B. (2000). Fundamental concepts of dependability. *Information Survivability Workshop*, (pp. 7–12).
- Beck, A. B., Andersen, N. A., Andersen, J. C., & Ravn, O. (2010). Mobotware - a plug-in based framework for mobile robots. In *IFAC Symposium on Intelligent Autonomous Vehicles*.
- Blanke, M., Izadi-Zamanabadi, R., S.A..Bgh, & Lunau, C. (1997). Fault-tolerant control systems - a holistic view. *Control Engineering Practice*, 5, 693–702.
- Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2006). *Diagnosis and fault tolerant control*. Springer, second edition.
- Blas, M. R. & Blanke, M. (2011). Automatic baling using stereo vision and texture learning. *Computers and Electronics in Agriculture*, 75, 159–168.
- Bouguerra, A., Karlsson, L., & Saffiotti, A. (2008). Monitoring the execution of robot plans using semantic knowledge. *Robotics and autonomous systems*, 56, 942–954.
- Bowman, A. W. & Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis*. Oxford University Press.
- Caponetti, F., Blas, M. R., & Blanke, M. (2011). Stochastic automata for outdoor semantic mapping using optimised signal quantisation. *Control Engineering Practice*, 19, 223–233.
- Chandrababu, S. & Christensen, H. I. (2009). Adding diagnosis to intelligent robot systems. In *Proc. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3961–3967).: IEEEExplore.
- Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., & Thrun, S. (2005). *Principles of Robot Motion*. Massachusetts Institute of Technology.
- Fischler, M. A. & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6).

- Galindo, C., Fernández-Madrigal, Juan-Antonio, González, J., & Saffiotti, A. (2008). Robot task planning using semantic maps. *Robotics and autonomous systems*, 56(11), 955–966.
- Galindo, C., Gonzalez, J., & J.A.Fernandez-Madrigal (2005a). An architecture for cognitive human-robot integration. application to rehabilitation robotics. In *Proc. IEEE International Conference on Mechatronics & Automation*.
- Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernandez-Madrigal, J., & Gonzalez, J. (2005b). Multi-hierarchical semantic maps for mobile robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*. (pp. 2278 – 2283).
- Griepentrog, H. W., Andersen, N. A., Andersen, J., Blanke, M., Heinemann, O., Madsen, T., Pedersen, S., Ravn, O., & Wulfsohn, D. (2009). Safe and reliable - further development of a field robot. In *Proc. 7th European Conference on Precision Agriculture (ECPA)* Wageningen: Academic Publishers.
- Hansen, S., Bayramoglu, E., Andersen, J. C., Ravn, O., Andersen, N., & Poulsen, N. K. (2011). Orchard navigation using derivative free kalman filtering. In *Proc. American Control Conference*.
- Hansen, S., Blanke, M., & Andersen, J. C. (2009). Autonomous tractor navigation in orchard - diagnosis and supervision for enhanced availability. In *Proc. 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFE-PROCESS'2009)* (pp. 360–365). Barcelona: IFAC.
- Isermann, R. (2006). *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*. Springer.
- Izadi-Zamanabadi, R. (1999). *Fault-tolerant Supervisory Control - System Analysis and Logic Design*. PhD thesis, Aalborg University, Department of Control Engineering.
- Konolige, K., Agrawal, M., Blas, M. R., Bolles, R. C., Gerkey, B., Solà, J., & Sundaesan, A. (2009). Mapping, navigation, and learning for off-road traversal. *J. of Field Robotics*, 26(1), 88–113.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press. Available at <http://planning.cs.uiuc.edu/>.
- Lunze, J. (2001). Diagnosis of quantised systems. *Fault Detection, Supervision and Safety for Technical Processes 2000*, 1(1), 29–40.
- Mettler, B., Dadkhah, N., & Kong, Z. (2010). Agile autonomous guidance using spatial value functions. *Control Engineering Practice*, 18(7), 773–788.
- Mozos, O., Stachniss, C., & Burgard, W. (2005). Supervised learning of places from range data using adaboost. *Proceedings of the 2005 IEEE, International Conference on Robotics and Automation*.
- Mozos, O. M., Jensfelt, P., Zender, H., Kruijff, G. J., & Burgard, W. (2007). From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots. In *Workshop "Semantic information in robotics" at the IEEE International Conference on Robotics and Automation*.
- Noura, H., Theilliol, D., Ponsart, J.-C., & Cham, A. (2009). *Fault-tolerant Control Systems - design and Practical Applications*. Springer.
- Nuchter, A., Wulf, O., Lingemann, K., Hertzberg, J., Wagner, B., & Surmann, H. (2005). 3d mapping with semantic knowledge. *RoboCup International Symposium*, (pp. 335–346).
- Schroder, J. (2003). *Modeling, state observation and diagnosis of quantised systems*. Number 282 in Lecture notes in control and information science. Springer.
- Supavatanakul, P. (2004). *Modelling and diagnosis of timed discrete-event systems*. PhD thesis, Technische Universitat Hamburg.
- Triebel, R., Mozos, O. M., & Burgard, W. (2008). Collective classification for labeling of places and objects in 2d and 3d range data. *Data Analysis, Machine Learning and Applications*, (pp. 293–300).