Information Fusion for Autonomous Robotic Weeding

Stefan Ericson
School of Technology and Society
University of Skövde
stefan.ericson@his.se

Klas Hedenberg
School of Technology and Society
University of Skövde
klas.hedenberg@his.se

Ronnie Johansson
Informatics Research Centre
University of Skövde
ronnie.johansson@his.se

Abstract: Information fusion has a potential applicability to a multitude of different applications. Still, the JDL model is mostly used to describe defense applications. This paper describes the information fusion process for a robot removing weed in a field. We analyze the robotic system by relating it to the JDL model functions. The civilian application we consider here has some properties which differ from the typical defense applications: (1) indifferent environment and (2) a predictable and structured process to achieve its objectives. As a consequence, situation estimates tend to deal with internal properties of the robot and its mission progress (through mission state transition) rather than external entities and their relations. Nevertheless, the JDL model appears useful for describing the fusion activities of the weeding robot system. We provide an example of how state transitions may be detected and exploited using information fusion and report on some initial results. An additional finding is that process refinement for this type of application can be expressed in terms of a finite state machine.

1 Introduction

1.1 Precision Agriculture

Farmers have to make many decisions concerning what and when to sow, how to add nutrient and pesticide, and when to harvest. Measurements of soil properties, weed pressure and crop nutrient are often made once in a field and action is then performed on the entire field. This approach is suboptimal since the field has local variations of the measured properties. Modern navigation technology has made it possible to treat each part of the field according to the specific demand, saving both money and environment. This approach is commonly known as *precision agriculture*.

More advanced tasks can be performed by using better sensors for positioning and identification, e.g. precision spraying [TRH99, DGS04], mechanical weeding [ÅB02, ÅB05] or automated harvesting [PHP$^+$02]. These systems are very information intense and there are several levels of decision making.

In this article, we analyze the design of an automated weeding robot from an information fusion perspective. The robot exists but all parts of the hardware (e.g., the weeding tool) and software (e.g., obstacle detection) are not implemented yet. However, this would not inhibit the analysis of the ideas described in this article.

1.2 Weeding Robot

The main task of the weeding robot is to remove weed on a field of plants. The robot is autonomous and has cameras and GPS (Global Positioning System) as main sensors. Software modules which process sensor data and provide information for decision making we refer to as *services*.[1]

This article is limited to the control of one robot. It has a home position where it can recharge its batteries and seek shelter in bad weather. The home position is typically in a garage or a mobile transport vehicle. This will be the start position for a mission to remove weed. The home position and the field are connected by a road. The road and field are limited by natural objects such as ditches or fences that are possible to detect with a camera.

Some a priori information is needed for the robot, such as a rough map of the field or some waypoints, so that the robot knows where to find the field. It should also approximately know the end positions of the rows.

1.3 Motivation and Contribution

The JDL model originates from a military context where the focus has been on describing objects (typically vehicles), their relations and impacts of situations. Recently, however, there has been an ambition to generalize the model to fit generic fusion problems (including civilian ones). So far, though, few attempts to discuss the applicability of the JDL model from a civilian applications' perspective have appeared.

The main objective of this article is to explore the utility of applying the JDL model [SBW99] to analyze the weeding robot application. There are at least two interesting differences between the weeding robot application and the typical defense application. First, unlike the defense case, the weeding robot has an indifferent and rather static environment (unlike the defense case where a hostile opponent responds to actions), and sensing its internal state and mission progress becomes more of an issue than estimating the intentions of hostile agents. Second, the mission of the robot is highly structured, i.e., it has a start state and proceeds to the goal state through the completion of a number of sub-tasks. The structure of a defense mission is typically much less certain. These two properties are shared by many civilian applications, e.g., manufacturing assembly [HM97].

What we end up with is a system with level 1 information concerning features of the field and the internal state of the robot and level 2 aspects reflecting the weeding robot's mission progress. Process refinement, level 4, is an important part of this application as the fusion and the use of sources and processing of data change considerably with different parts of

[1] Some of the services mentioned in the rest of the article have been implemented while others are envisioned.

the mission. A simple fusion method for detecting mission state transitions is implemented and tested.

1.4 Overview

In Section 2, we describe and decompose the weeding robot mission into a number of sub-tasks. In Section 3, we present the weeding robot platform and in Section 4 our experiments are described. Section 5 discusses the experiment results. In Section 6, we summarize and conclude the article.

2 The Weeding Mission and Fusion

In this section, we decompose the weeding robot problem into a number of sub-tasks and describe the transitions between sub-tasks, and suggest that fusion methods could be used to detect the transitions.

2.1 Mission Decomposition

A successful mission for a weeding robot involves completing a number of sub-tasks. The sub-tasks are (1) Navigate to field, (2) Weeding, and (3) Navigate to home. Mission and sub-tasks are illustrated in Figure 1. The weeding mission can be described by an event-driven finite state machine as shown in Figure 2. The event-driven machine consists of states (the rounded boxes) and transitions (the directed arcs). The states represent the different sub-tasks of the mission and the transition events denote conditions that have to be fulfilled for the weeding robot to change sub-tasks. Here, we have added a fourth sub-task, which corresponds to activities that the robot undertakes while in its home position. The filled circle with the arrow pointing to the maintenance sub-task indicates the typical start state for the weeding robot.

We call the activity the robot system engages in to complete a sub-task a *mode of operation* (or mode for short). Each mode also involves the detection of transition events. Furthermore, to detect transition events, we need to establish an estimation of the mission situation, i.e., the state of the robot and the environment, using the sensors of the robot platform. The modes of operation are further discussed in the next section.

A formal description of this finite state machine is the tuple $A = (T, t_0, E, M, \delta, \beta)$, where

- T is the set of sub-tasks
- t_0 is the initial sub-task
- E is the set of (transition) events

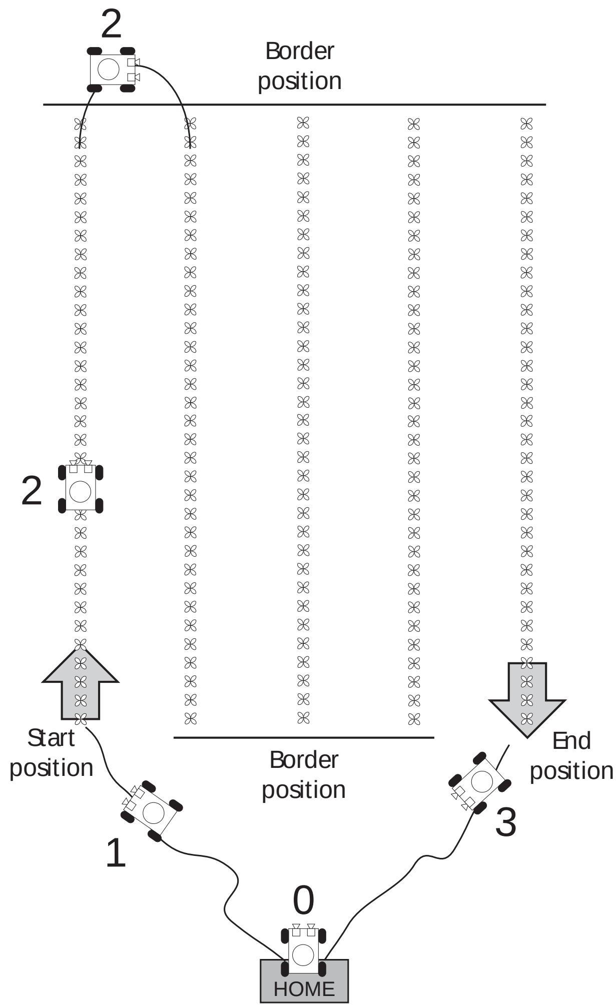

Figure 1: Sub-tasks of the robotic weeding mission

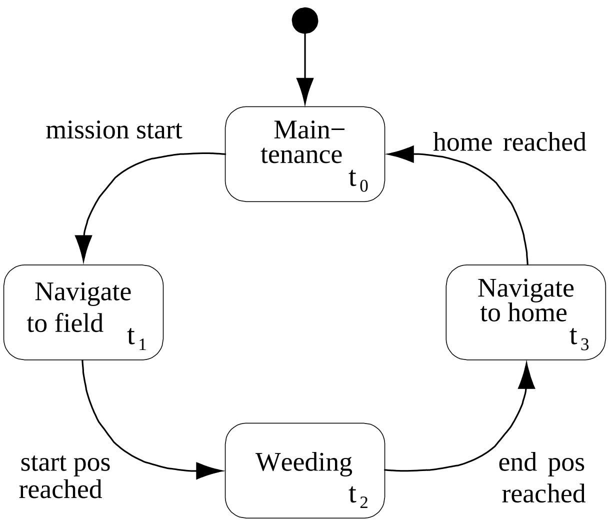

Figure 2: Event-driven finite state machine for the weeding mission

- M is the set of modes of operation
- δ is the state transition function: $\delta : T \times E \to T$
- β is the mode selection function: $\beta : T \to M$

Hence, the mission starts in sub-task t_0. Detected events in E results in the change of sub-tasks, in a manner described by the transition function δ, and initiating a new sub-task results in the invocation of a new mode of operation, specified by β, to deal with the situation.

2.2 Modes of Operation

Each mode involves collecting information from platform information services and building a situation representation (or here simply called 'situation') which is used to complete the sub-task and to detect transition events. Transition events are typically issued by the mode itself by applying fusion algorithms that combine information from sources (e.g., to determine that the start position has been reached).

2.2.1 Maintenance

The maintenance mode focuses on detecting the mission start transition event. For this mode, the situation consists of information about battery level, environmental conditions, weeding need and the inferred information about whether a weeding mission should be initiated.

2.2.2 Navigate to Field

The navigate to field sub-task involves maneuvering the robot to the start position (where it should begin to weed) while avoiding obstacles. The mode has to estimate the platform's global position to determine its progress towards the start position. The transition event is triggered if the estimated distance to the start position is small and if the Row estimation service detects rows.

2.2.3 Weeding

The weeding mode is the most complex of the four modes. It arbitrates between three behaviors: Find row, Weeding and Change row. All of these behaviors employ the motor actuator, but Weeding also uses the weed tool actuator.

Behavior Weeding uses the Row following service to follow a row of plants. The mode uses the Weed tool service to remove weed and the End of row detection service to switch to the Change row behavior if the robot's position is close to a field border position.

The Change row behavior uses the Local position estimation service to turn around and Row estimation to change behavior to Weeding.

The transition event end position reached is triggered if the position (using the global position estimate) is close enough to a field end position (given by the field map).

2.2.4 Navigate to Home

The navigate to home mode is identical to the navigate to field mode except for the different transition event. In this case, the transition event, home reached, is triggered by the position estimate (provided by the Global position estimation service) together with the known home position and possibly the a Landmark recognition service.

2.3 Relation to the JDL model

The purpose of the Joint Directors of Laboratories (JDL) data fusion model is to define and highlight essential functions of fusion processes and to facilitate the communication between researchers, engineers and users [HM04].

In its latest revision [SBW99], the JDL model consists of five functions:

- level 0 - sub-object assessment (e.g., signal processing)
- level 1 - object assessment (e.g., estimation of observed entity properties, such as position and type)

- level 2 - situation assessment (e.g., estimation of the context and relations of entities)
- level 3 - impact assessment (e.g., estimation of future events given the current situation), and
- level 4 - process refinement (e.g., adaptation of the fusion process in light of changing environment state and mission objectives).

From a JDL model perspective, most of the information generated by our robot system belong to level 1, e.g., robot position estimate, and obstacle detection. Level 2 information typically refers to relations between entities generated by level 1 fusion functions. Some of our transition event estimates are of this type, e.g., start position reached which is based on a relation between the own global position estimate, its relation to a map and the detection of rows. In our current analysis, level 3 is not considered, but could occur if the state of the robotic platform is compared to external circumstances (e.g., to anticipate and avoid collisions and platform breakdown).

It is interesting to note how the situation representation changes (and therefore also the use of services) with different modes. There are some pieces of information which are irrelevant for the decision-making (and hence the situation representation) in some modes, but relevant in others. Row detection, which is an important service during weeding but not while navigating to the home position, is one example. Hence, not all services have to be active all the time; some can be inhibited during some modes while others are activated. The activity just described, i.e., selecting focus of attention, is in some sense a part of a JDL model function that is rarely discussed, namely, level 4 process refinement.

2.4 Fusion to Detect Transition Events

In this article, we focus on the fundamental estimation problem of detecting state transitions. Our initial approach to this problem is the probabilistic model:

$$P(ST) = \sum_{P,A,R} P(ST|P, A, R)P(P)P(A)P(R) \tag{1}$$

Where ST is a binary variable (with values True/False) representing that a state transition is imminent; P (Close/Not Close) represents the position of the robot relative to the end position of the sub-task; A (Close/Not Close) is the heading angle of the robot relative to the end-position angle of the sub task; and R (True/False) represents row detection. For simplicity, we assume that each individual occurrence of $P = Far$, $A = NotClose$ and $R = False$, will result in $ST = False$. With this assumption, $P(ST|P, A, R)$ can be expressed as a noisy-And gate [GD00]. Furthermore, from the noisy-And gate assumption follows $P(ST = \text{True}|P, A, R) > 0$ only for $P = \text{Close}$, $A = \text{Close}$ and $R = \text{True}$, Eq. (1) reduces[2] to the simple expression

$$P(ST = \text{True}) = P(P = \text{Close})P(A = \text{Close})P(R = \text{True}), \tag{2}$$

[2]Assuming $P(ST = \text{True}\ |P = \text{Close}, A = \text{Close}, R = \text{True}) = 1$

which we use in our experiments in Section 4.1 to estimate the degree of certainty that a state transition is imminent.

3 Application

In this section, we present a setup for the weeding robot including a short overview of the sensors and some software services.

3.1 Sensors

The hardware configuration is presented in Figure 3. The robot is primarily constructed from a electric wheelchair. Encoders are placed on the wheel axis to measure rotational position of each wheel. This is a cheap and simple sensor that provides good positioning under the assumption that the shape of the wheel is uniform and there is no wheel-slip. Since this sensor only measures relative movement, any introduced error is accumulated. Camera is the primary sensor for the system. It is a very powerful sensor since a lot of information can be extracted from images. Different information is provided depending on the algorithm applied to the image. The hardware is quite simple and consists of an image acquisition system (camera) and a computer.

Three front-mounted cameras for obstacle end row detection are selected to capture a view of the area in front of the robot. These are used in a trinocular stereo setup, where algorithms for measuring distances can be applied. There are two cameras looking down to the field. The advantage of using two cameras is that epipolar geometry algorithms can be applied to measure distance to objects in the stereo image. The cameras can be used for both plant detection and visual odometry. A drawback of using cameras is their sensitivity to different light conditions. Hence, light sources are mounted under the robot to illuminate the ground for the down-looking cameras. This setup creates a controlled light condition, which enhances the result from the vision algorithms applied on this camera.

3.2 Vision Algorithms

The three onboard computers run Ubuntu GNU/Linux which incorporates the Player/Stage architecture. Two computers are dedicated for computer vision algorithms written with the open source package OpenCV. The third computer, mission computer, has an onboard display.

There are several algorithms for the machine vision system, depending on what information that is needed. Some algorithms require more computational resources and take longer time than others to complete. Algorithms suitable for this task are: Hough transform for a row-following system, Visual odometer to calculate traveled distance from consecutive

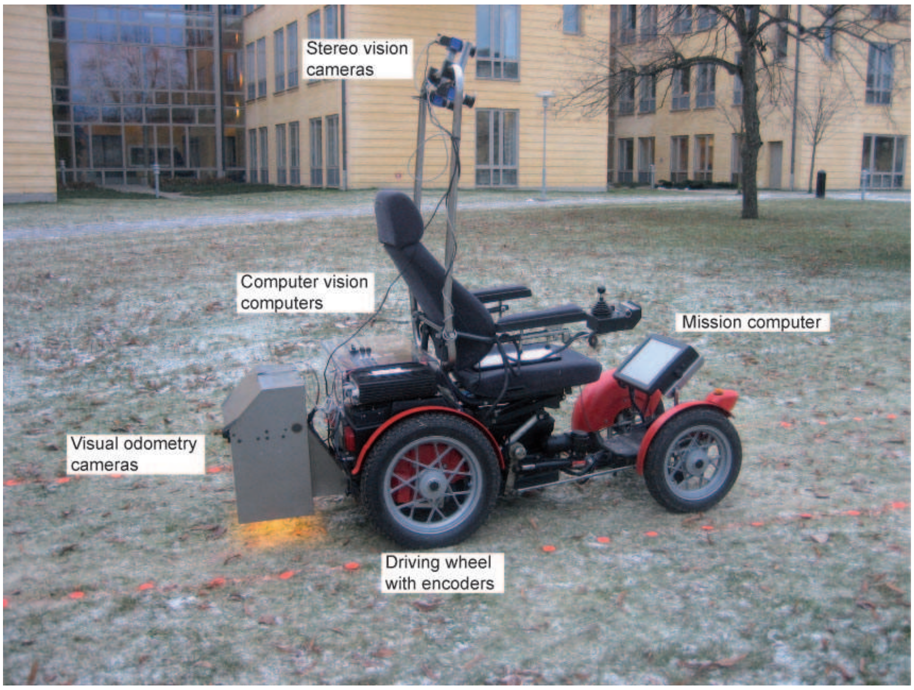

Figure 3: The agricultural robot prototype

images, Object recognition to identify objects by comparing points of interest to a database and Epipolar geometry to measure the distance to objects using stereo cameras.

4 Evaluation

Some, but not all, parts of the weeding robot system have been implemented. In the following experiment, we focus on the sensors and algorithms needed to detect the state transitions of the weeding mission.

4.1 Experiments

Our robotic system is evaluated by first collecting data from real experiments, and then using Matlab to analyze the data offline. In this way, different methods can be evaluated using the same dataset. Two test-runs are performed where one is used for setting up the decision rules and the other for evaluation.

The experiments are performed on an artificial field constructed on a green lawn. The agricultural field is simulated using white poker chips placed in two rows. Each row is approximately 10 meter long with five chips each meter. The distance between the rows is 1 meter. The reason for using white color to represent plants is that it contrasts to the green lawn in a similar way as green plants contrasts to black soil. It is also easy to introduce noise by adding white objects to the field. In this way, a false row is constructed close to the start-point of the first row. It is placed with an angle of about 90° from the required heading in the start-point.

The robot is manually controlled to simulate realistic driving behaviors of an autonomous system. With the manual control, each state transition is emphasized by leaving the robotic platform immobile for about ten seconds. Data is recorded from the encoders which gives position and heading estimates. Data is also recorded from the row-following system which provides a row detect signal, perpendicular distance to row and angle to the row.

A manual test-run for data collection consists of:

- Start at home position
- Head toward false row
- Turn and place the robot in the beginning of real row
- Follow the row and stop at end of row
- Turn the robot around and stop at the beginning of the next row
- Follow the row and stop at end of the row
- Drive to home position

The two test-runs are performed on the same field. Figure 4 shows the reference field and estimated position from encoder data (odometry shown as a dashed green trajectory). The start and end position of each row is used as *reference points* with a known position. Since the robot is manually controlled, we know that the robot passes all reference points in a correct way, but the estimate of its own position contains an accumulated error.

When a state transition is detected, the robot is assumed to be at the reference point with an expected heading. In this way, accumulated error can be removed. During the row-following, the heading is assumed to be in the direction of the row. The data that are used for decision is the distance to next reference point, heading and the row detection signal (compare to Eq. (2)). This compensated odometry is also shown in Figure 4 (the dotted red line).

5 Results

The test-run in Figure 4 shows the estimated trajectory of the robot when only relying on odometry (green dashed line) one trajectory that uses estimates the state transitions and

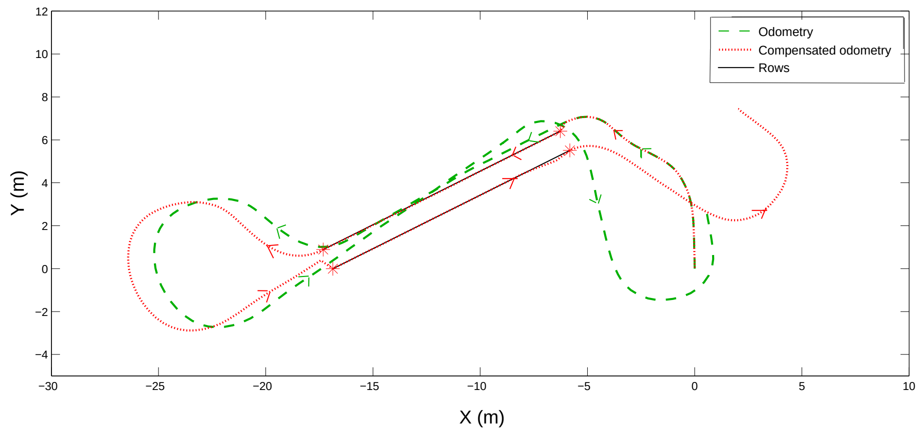

Figure 4: Test field and plot of position estimations from encoder data

exploits the known reference points (red dotted line) to try improve the trajectory. The compensated trajectory appears (for the most part) more correct.

Figure 5 shows a plot of all individual data that is used for the transition decision (state 2a, 2b, and 2c are different parts of the weeding state). The first plot shows distance to reference point, the second shows heading error to reference and the third plot shows the signal from the row detection system (note that in states 2a and 2c, where the end of the row should be detected, the probability for row not detected instead of detected is shown). The fourth plot shows the result of the fusing the three aforementioned estimates using the approach described in Section 2.4 (note that for some state transitions a row should be detected and for others not). The solid red vertical lines indicate at what time the robotic system has detected a state transition and the dashed green lines shows the actual time the state transition occurred during the manual drive of the robot.

As can be seen, the position probability is decreasing when the robot is approaching the first reference point (i.e., the first row). The false row is detected, but since the heading is wrong the state transition estimation remains low. During the turn to the right row, the probability of row detection decreases for a while until the real row is detected. At this point, the decision to change mode is made and the estimated position is corrected to the known reference position of the point.

Figure 4 also shows that compensated odometry requires carefully designed estimations of row detection, position and angle, as the compensated odometry results in a deviating path when the robot returns to the home position. This is explained by the time differences between the estimated and actual state transitions in Figure 5.

6 Summary and Conclusion

In this article, we describe the proposed design of a weeding robot system including a robotic platform, sensors, and software. The software is a collection of services which should be tailored to utilize the robotic sensors and actuators effectively.

From an information fusion perspective, the fusion process and the generation of information (i.e., decision support) for the weeding robot is of essence. The JDL model has a design which should appeal to diverse applications, but has for the most part only been used for defense applications. In this article, we test the applicability of the JDL model to the weeding robot system.

The result of this study is that the JDL model is applicable, but the information generated is somewhat different than the typical defense information. The level 1 information, here, concerns, e.g., the robot's own position estimate and obstacle detection. The level 2 information relates mainly to the transition event estimates, e.g., start position reached which is based on a relation between the own global position estimate, its relation to a map and the detection of rows. Compared to many defense applications, the generated information here mostly refers to the state of the robotic system and mission progress rather than external agents. An approach to estimate state transitions was implemented and tested. State transition information was further used to improve the trajectory estimation of the robot. Initial results indicate both advantages and disadvantages.

Another interpretation of the JDL model in the weeding robot system is level 4, process refinement. Given that the mission of the robot can be described with a finite state machine, process adaptation can simply be described with state transitions and mode selection. The reason is that mode selection results in a change of focus of attention which is reflected in the change of the type of software services used and information processed.

7 Acknowledgments

This work was supported by the Information Fusion Research Program (www.infofusion.se) at the University of Skövde, Sweden, in partnership with the Swedish Knowledge Foundation under grant 2003/0104, and participating partner companies.

References

[ÅB02] Björn Åstrand and Albert-Jan Baerveldt. An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots*, 13(1):21–35, 2002.

[ÅB05] Björn Åstrand and Albert-Jan Baerveldt. A vision based row-following system for agricultural field machinery. *Mechatronics*, 15(2):251–269, 2005.

[DGS04] D. Downey, D. K. Giles, and D. C. Slaughter. Pulsed jet micro-spray applications for high spatial resolution of deposition on biological targets. *Atomization and Sprays*,

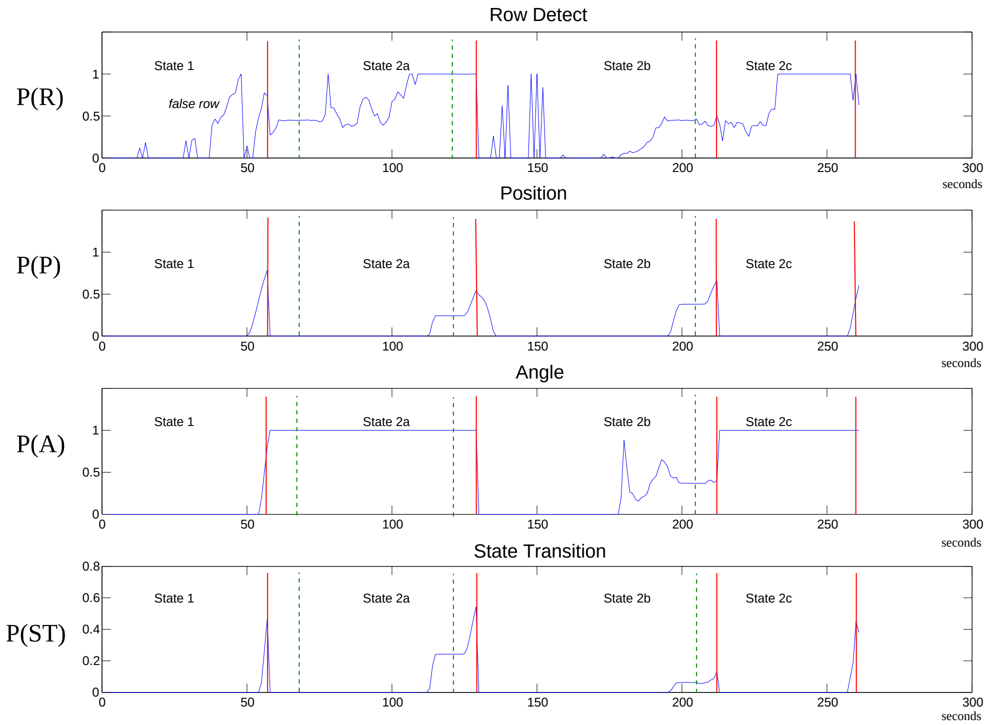

Figure 5: Result of the state transition estimation

14(2):93–110, 2004.

[GD00] Severino F. Galán and Francisco J. Díez. Modelling dynamic causal interactions with Bayesian networks: Temporal noisy gates. In *Proceedings of the 2nd International Workshop on Causal Networks (CaNew'2000)*, pages 1–5, August 2000.

[HM97] Geir E. Hovland and Brenan J. McCarragher. Dynamic sensor selection for robotic systems. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA)*, pages 272–277. IEEE, 1997.

[HM04] David L. Hall and Sonya A. H. McMullen. *Mathematical techniques in multisensor data fusion*. Artech house, 2 edition, 2004.

[PHP$^+$02] Thomas Pilarski, Michael Happold, Henning Pangels, Mark Ollis, Kerien Fitzpatrick, and Anthony Stentz. The demeter system for automated harvesting. *Auton. Robots*, 13(1):9–20, 2002.

[SBW99] Alan N. Steinberg, Christopher L. Bowman, and Franklin E. White. Revisions to the JDL data fusion model. In *SPIE Conference on sensor fusion: architectures, algorithms, and applications III*, volume 3719, April 1999.

[TRH99] Lei Tian, John F. Reid, and John W. Hummel. Development of a precision sprayer for site-specific management. *Transactions of the ASAE*, 42(4):893–900, 1999.