View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Southern Denmark Research Output

Pure

Syddansk Universitet

Decomposition of Agricultural tasks into Robotic Behaviours

Fountas, Spyros; Blaskmore, Benjamin Simon; Vougioukas, Stavros; Tang, Lie; Sørensen, Claus Aage Grøn; Jørgensen, Rasmus Nyholm

Published in:
E-journal CIGR

Publication date:
2007

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

Citation for pulished version (APA):
Fountas, S., Blaskmore, B. S., Vougioukas, S., Tang, L., Sørensen, C. A. G., & Jørgensen, R. N. (2007). Decomposition of Agricultural tasks into Robotic Behaviours. E-journal CIGR, IX.

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 ?

Take down policy
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.

Download date: 15. Oct. 2017

Decomposition of Agricultural tasks into Robotic Behaviours

S. Fountas[1], B. S. Blackmore[1], S. Vougioukas[2], L. Tang[3], C. G. Sørensen[4], R. Jørgensen[4]

[1] University of Thessaly, Department of Crop Production, Volos, Greece
[2] Aristotle University, Department of Agricultural Sciences, Thessaloniki, Greece
[3] Iowa State University, Ag & Biosystems Engineering, USA
[4] Aarhus University, Research Centre Bygholm, Denmark
simon@unibots.com

ABSTRACT

A new method is described that can be used to decompose human controlled agricultural operations into robotic behaviours embedded in an autonomous tractor. Four main levels have been identified: Operation, Task, Optimisation/Behaviour and Primitive actions where each level is subsumed by the level above. Tasks were further classified into two distinctive types, deterministic tasks that can be planned and optimised before the operation begins and reactive tasks that have associated behaviours to deal with unknown conditions whilst in the field. Both deterministic and reactive tasks can be further decomposed into primitive actions, which in turn are converted into the tractor directrix. Examples of this method are given for exploring an unknown area and ploughing a field. Results of a simulation of the explore operation are presented.

Keywords: Autonomous vehicles, route planning, autonomous tractor

1. INTRODUCTION TO AUTONOMOUS VEHICLES IN AGRICULTURE

Autonomous vehicles have been widely used in industrial production and warehouses, where a controlled environment can be guaranteed. In agriculture, research into driverless vehicles has always been a dream but serious research started in the early 1960's. These projects mainly involved automatic steered tractors and Wilson (2000) published a review. Furthermore, Hollingum (1999) reviewed the agricultural robotic developments around the world and Kondo and Ting (1998) elaborated on robotics for bio production systems, including open fields. However, there is a limited number of references on fully autonomous agricultural vehicles, such as the Demeter system for automated harvesting equipped with a video camera and GPS for navigation **(**Pilarski *et al.*, 2002) as well as semiautonomous agricultural vehicles (Freyberger and Jahns, 2000; Billingsley, 2000)**.** Nevertheless, in recent years the development of autonomous vehicles in agriculture has experienced an increased interest. This development has led many researchers to start developing more rational and adaptable vehicles based on a behavioural approach. Blackmore *et al.* (2002a) argued that autonomous vehicles for agricultural operations should behave sensibly in a semi-natural environment, over long periods of time, unattended, whilst carrying out a useful task. These and other specifications of requirements are needed for a new system of autonomous agricultural vehicles. These vehicles would be small, light weight, capable of receiving instructions and communicating information, able to be co-ordinated with other machines, collaborate with other machines and behave in a safe manner, even when partial system failures occur, while carrying out a range of useful tasks

In the case of autonomous vehicles, much effort has been put on developing control systems architectures to deal with the complexity and the interaction with the environment. The aim has been to make the vehicles more capable of operating in a partial or unknown environment. Yavuz and Bradshaw (2002) carried out a detailed analysis of 43 control systems architectures and they concluded that there is not an optimal systems architecture that covers all aspects of robotics, while a combination of architectures would probably be a sensible approach. Blackmore *et al.* 2002b proposed an object-oriented system architecture customized to tractors. The need to establish a unified systems architecture in field robotics for the American military has led to the development of the Joint Architecture for Unmanned Ground Systems (JAUGS) (Torrie *et al.*, 2002).

The approach, and hence the systems architecture, that has gained increased attention is the hybrid system. This hybrid approach combines both deterministic control and reactive behaviour. The deterministic control is hierarchical and is usually used for very structured and known environments (e.g. a field). The reactive behaviour is used to respond directly to a stimulus and is used for unstructured or unknown environments and conditions (e.g. a tree fallen over in a field). The advantage of reactive behaviour is that it can deal with uncertainty in perception. It need not have to recognize an unknown object or situation but be able to classify it in terms of how to react to it. This approach dramatically increases the robustness of the behaviour. On the other hand, the main disadvantage of purely reactive behaviour is that it takes into consideration only the current state of sensory information and not any overall goal-oriented targets as in deterministic control (Yavuz and Bradshaw 2002). In contrast, a hybrid system combines adaptive and goal-oriented control. These hybrid approaches, based on behaviour are therefore the centre of focus for many researchers and there have been many control architectures proposed. (e.g. Na and Oh, 2003; Yavuz and Bradshaw, 2002**)**.

For agricultural operations, the environment is usually semi-structured and difficult to control. Deterministic tasks can be optimised based on known structures within the field. Static and mobile objects that can become obstacles are also present in this type of environment. Static obstacles could be trees or stones, while mobile obstacles could be people, cattle or sheep. An autonomous tractor would need to classify the type of obstacle and then react in a safe and rational manner according to the context.

1.1 Behaviour-based Robotics

Behaviour based robotic systems are used for reactive control. Such systems provide the means for a robot to navigate in an uncertain and unpredictable world without planning, by endowing the robot with behaviours that deal with specific goals independently and coordinating them in a purposeful way (Arkin, 1998). Figure 1 illustrates the traditional way of hierarchical control system and Figure 2 the novel approach of behaviour-based control, is where behaviour is defined as a stimulus-action pair. Behaviour-based robotics have many common and divergent aspects. The common aspects are the emphasis on coupling perception and execution; decomposition to contextually meaningful units; and avoidance of representational symbolic knowledge. The differences lie in the granularity of behavioural decomposition; the basis of behaviour specification (ethological, situated-activity or experimental); the response encoding method (discrete or continuous); the coordination methods used (competitive or cooperative); and the programming languages employed (Arkin, 1998). Figure 3 and 4 illustrate the two different behaviours' coordination methods in behaviour-based robotics, competitive and cooperative. The different response encoding methods, discrete and continuous are demonstrated in the experiments of this paper.

Subsumption architecture is the most widely used and applied architecture in behaviour-based robotics (Brooks, 1986; Brooks, 1999). Subsumption architecture is a layered architectural control system that directly links perception to execution. Each layer specifies a behaviour pattern and is implemented with augmented finite state machines. The output of layers can be used as inputs to others, suppressing and inhibits messages. As Brooks (1999) argued, this architecture has been implemented by many mobile vehicles that have led to many robust and flexible systems. However, subsumption architecture has received some criticism. Gat (1998) argued that it is not sufficiently modular and as a consequence, it cannot deal with complexity. This is due to the fact that the upper layers interfere with the lower layers and cannot be designed independently.

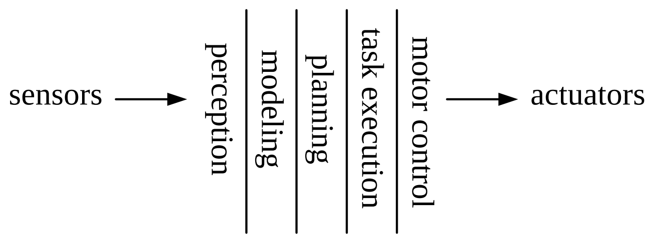

Figure 1. A traditional decomposition of a mobile robot control system into functional modules (Brooks, 1986)

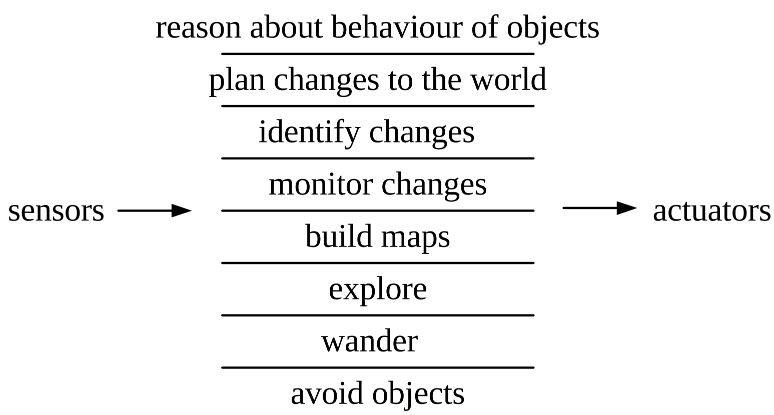

Figure 2. A decomposition of a mobile robot control system based on task achieving behaviours (Brooks, 1986)

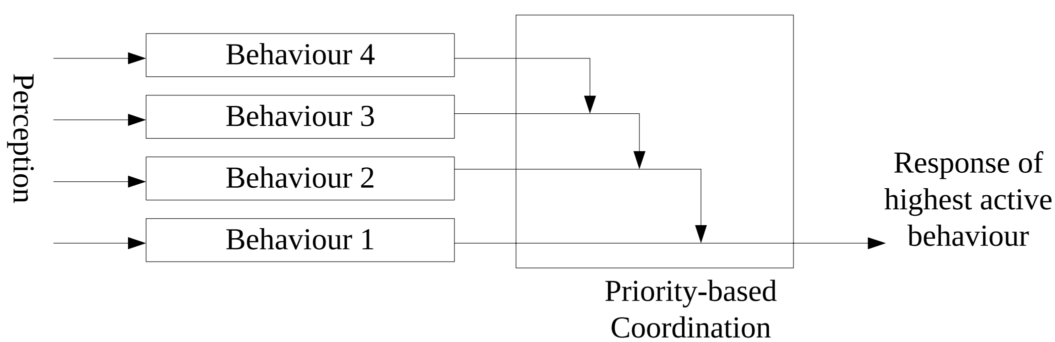

Figure 3. Competitive coordination of behaviours (Arkin, 1998)

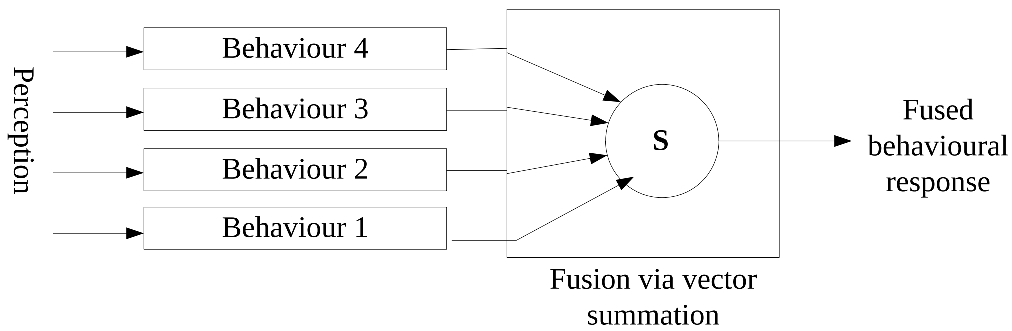

Figure 4. Cooperative coordination of behaviours (Arkin, 1998)

1.2 Behaviour

The word "**behaviour**" in robotics has been used in many different ways. Arkin (1998) refers to behaviour as a response to a stimulus, associated only with reactive control architectures. He identified two types of behaviours. **Reactive behaviour**, which is a reactive behaviour, created by direct coupling between perception and execution; and **emergent behaviour**, which is the (desired or otherwise) consequence of the interaction of the active individual behaviours with the environment. Rzevski (1995) referred to **behaviour** as a particular interaction of the machine with its environment, defined by a set of inputs and outputs, similar to what Arkin referred to as **emergent behaviour**. Pfeifer and Scheier (1999) argued that **behaviour** is what an autonomous agent is observed doing, always in interaction with the environment. They referred to **emergent behaviour** as not a programmed behaviour that is derived by the interaction of the agent with the environment, but usually when many processes are assembled to derive a single behaviour. He also referred to **desired behaviour** as similar to the task that the autonomous agent would have to accomplish. Additionally, Brooks (1989) argued that there are two types of behaviour, **higher-level** (macro) such as following people that control **lower-level** (micro) behaviours, such as leg lifting and force balancing. Furthermore, Gat (1998) referred to **Behaviour** as a piece of code that produces a **behaviour** when it is running. He also distinguished between **primitive behaviours**, which can be composed to produce more complex **task-achieving behaviours**. Finally, Konolige and Myers (1998) argued that there are another two types of behaviours: **reactive and goal-oriented behaviours**. Reactive behaviours are event-driven behaviours that exist while an unexpected event occurs, while goal-oriented behaviours are produced to satisfy individual tasks using artefacts (*a priori* information, perceptual features and user commands).

Apart from the differences in the definition of behaviour, there is also to some degree confusion on the use of behaviour, primitive behaviour and process in applications. Obstacle avoidance, for example, has been referred to as behaviour [Gat, (1998); (Murphy, 1998); Ridao, et al. (2002)], reactive process (Preifer and Scheir, (1999)), as well as agent with a number of modes (Shyu *et al.* (1998). Hassan *et al.* (2001) identified a number of reactive and deliberative processes for their autonomous vehicle, where "obstacle avoidance behaviour" was denoted a reactive process. Kosecka *et al.* (1997), identified a number of macro behaviours for their autonomous vehicles, such as bumper behaviour, GoTo behaviour, Detect behaviour, Servo behaviour, path planning process, localize behaviour, initialization behaviour. When they explained the bumper behaviour, though, they explained it to be a low-level process, responsible for stopping the robot. However, ‘bumper’ in other cases is an input to a process or behaviour, and not behaviour itself.

1.3 The Need to Develop a Methodology to Decompose Agricultural Operations

In order to be able to define behaviours for an autonomous tractor, a decomposition of the various tasks into behaviours or process has to be carried out. Pfeifer and Scheier (1999) mentioned that identifying processes needed to achieve a desired behaviour in accomplishing a task is a basic research issue. They also mentioned that, even though in some cases, it is straightforward, in other cases it is very complicated and no methodology existed to achieve that. Kosecka *et al.* (1997) supported the idea of representing a task as a network of processes, where each process is a Finite State Machine (FSM) and the transitions between the states are modelled by events, which show the initiation, termination, interruption or change. Summarising, Arkin (1998) argued that there are three methods to design robotic behaviours. Ethologically guided, (representing animal behaviours), situated activity-based, (generated by the situation that it has to handle); and experimental driven, (a bottom-up approach), testing a limited number of behaviours and then add more. In this paper, the exploratory experiments follow the experimental-driven methods, but the overall strategy of our research group is the activity-based approach method.

The objective of this paper was to establish a methodology for decomposing agricultural tasks into primitive actions and providing definitions for the different components.

2. METHODS AND MATERIALS

2.1 Description of the Decomposition Methodology

A method is required to define, understand and decompose the intelligent behaviour of a human in a certain context into the sensible behaviour of a machine in the same context. To achieve this, the physical actions of a person were analysed and then defined in a number of different logical representations, semantics and a lexicon.

Operation is the field operation that the vehicle should carry out. (e.g. ploughing a field) Each operation can have a number of tasks.

Tasks are the main activities that the vehicle should execute while carrying out an operation. They include the main predetermined actions (e.g. ploughing) and reactions (e.g. obstacle avoidance) that the vehicle should carry out. Two task groupings have been identified: deterministic and reactive.

Deterministic tasks are those tasks that can be planned before the operation starts (e.g. route plan). Deterministic tasks can be optimised in terms of best utilising existing resources based on the prior knowledge about the tractor, field and conditions.

Reactive tasks are those tasks that are carried out when uncertainty is encountered. These tasks react in real-time to local conditions that were not known before the operation started. Reactive tasks can be defined by their behaviour to certain classes of situation (e.g. stopping when approached, obstacle avoidance).

Deterministic task optimisations are the **way** in which the deterministic task is carried out. These are a set of equations (in the form of linear programming rules) that should be optimised within the final result (e.g. .plough straight, minimize route).

Reactive task behaviours are the **way** in which the reactions should be carried out. These behaviours are defined in terms of reaction to stimuli and context (e.g. turn to the right when encountering an unknown obstacle)

Primitive actions are the simplest natural language descriptions of the vehicle functions (e.g. stop, go ahead, back up, turn right, turn left).

Directrix is the command of what the vehicle should do. It can be translated from the primitive actions and is machine dependant (e.g. velocity, trajectory)

Figure 5 shows a Venn diagram (or finite state diagram) depicting the relationship between the different elements and how the inner functions are nested within the outer ones.

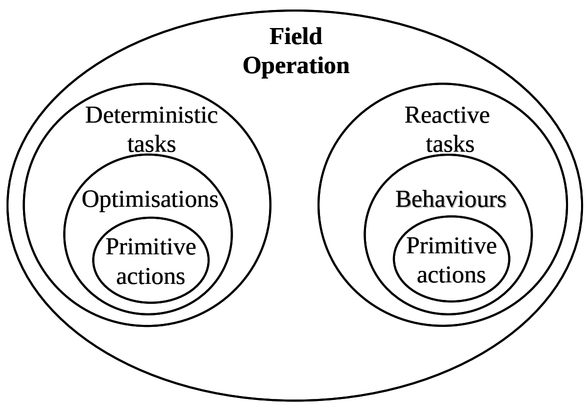

Figure 5. Generic behavioural subsumption diagram

2.2 Operation

The operation is the highest level of operational activity that the vehicle will carry out in order to realise the cultural practices of the crop system (Sørensen, 1999). It describes the main agronomic purpose of what the vehicle will have to achieve and has an immediately intelligible; meaning to the manager. Agricultural operations that have been identified are: ploughing a field, cultivating a seedbed, seeding a field, fertilising a plot, etc.

2.3 Task

To be able to carry out the operation, some tasks will have to be performed involving relevant implements. These tasks comprise the actions and reactions that are required to make something happen. To plough a field, the tasks are to identify the resources (tractor, plough, field etc), develop a method of ploughing (direction, width etc). Both of these tasks can be optimised before the ploughing starts, so they are called deterministic tasks. There are a number of parameters that can be taken into account so that the actions can be optimised under certain conditions (minimise distance travelled, keep straight and parallel to previous rows etc). Different tasks will require different machine configurations or modes, as the tractor will have the plough in the ground while ploughing and have it raised during transport.

Some tasks cannot be planned beforehand but can be foreseen. A typical reactive task would be to avoid an unknown object. We do not know what the object may be and may have difficulty in recognising it but we can decide what would be sensible behaviour in such a situation. An expert system can be used to resolve the vehicle context into a set of suitable reactions based on human behaviour.

2.4 Optimisation

A deterministic task can be optimised given a set of technical and temporal constraints as well as operational preference guidelines that should be met, maximised or minimised. This priory optimisation predetermines operational parameters like working speed, driving pattern, transport logistics, etc. The modelling approaches include simulation, linear programming, and other scheduling techniques (Elderen & Kroeze, 1994; Sørensen, 2003). In the example of ploughing a field, we can optimise the route that the tractor should take by identifying the characteristics that we want the route to take. There are many ways to plough a field, but we can identify one particular way by giving a set of criteria such as to turn the soil opposite to the previous ploughing operation, minimise distance travelled, keep the ploughing in a straight line parallel to the previous row, etc. For example, linear programming can then suggest a route that may meet the criteria.

2.5 Behaviour

Reactive tasks can be defined by they way in which the task is carried out or the behaviour of the task. When encountering an unknown situation, it can be classified into a set reaction that exhibits a defined behaviour suitable for the context. Some contexts and their associated reactive tasks and behaviours have been identified in Blackmore *et. al.* (2002b) and are listed here: Avoiding, Threat, Assessing, Skid, Slip, Stuck, Sink, Tilt, Weather extreme and Theft.

3. RESULTS AND DISCUSSION

3.1 Decomposition and Simulation of the 'Explore' Operation

This method was applied to one of the basic operations for a mobile robot, that of being able to explore and record its environment. Descriptive English was used to describe the required operation, tasks and behaviours before being decomposed into structured English and machine code in a robotic simulator called MobotSim (Gonzalo 2005). The Explore operation was to survey an unknown field with an unknown closed boundary that may have a number of different obstacles within it. At the end of this operation all obstacles and the field boundary should have been recorded.

Structured English was then used to define the primitive actions and coded into the simulator. An example of the logic used is presented in Table 1.

The Explore operation was made up of four tasks. The first was a simple deterministic task to move the vehicle straight forward until it reaches an object (Part (a) in Table 1.). It then turned right and switched into a reactive task to follow the edge of the object at a set distance (Part (b)). If the path around the object closed in an anti-clockwise manner then this was defined as an obstacle. If the path closed in a clockwise manner then this was the field boundary (Part(c)). The third task was to plan a route so that the entire field within the field boundary was surveyed (Part (d)) and the fourth task was to deterministically follow the route plan (Part (e)). If at any time an unknown obstacle was encountered, the vehicle could switch into the reactive follow edge task again and re plan the route.

Given that an object could be static or dynamic there should be different behaviours for each. Firstly, when an unknown object was encountered the vehicle stopped and switched into the 'Watch and Wait' task. If the object appeared to be stationary then it would switch into follow edge task. If the object appeared to be mobile then the vehicle could remain in the Watch and Wait task until the mobile object moved out of the way and then continue. These behaviours have not yet been implemented in the simulator.

Table 1. Example of Structured English used in the simulation

```
Sub Main
Initialise
Do
        Do
        Forward                                                  (a)
        Loop Until Close
        IdentifyObject
        If Not KnownObstacle Then ExploreObject                  (b)
Loop Until Boundary                                              (c)
PlanRoute                                                        (d)
For WayPoint = FirstWayPoint To LastWayPoint                     (e)
```

```
    CalcIdealRoute
    NavigateTo(WayPoint)
    If ObjectNear Then
        IdentifyObject
    If Not KnownObstacle Then ExploreObject Else AvoidObstacle(CurrentObstacle)
    End If
Next WayPoint
End
End Sub
```

Figure 6 shows the results of this simulation on three fields of increasing complexity.

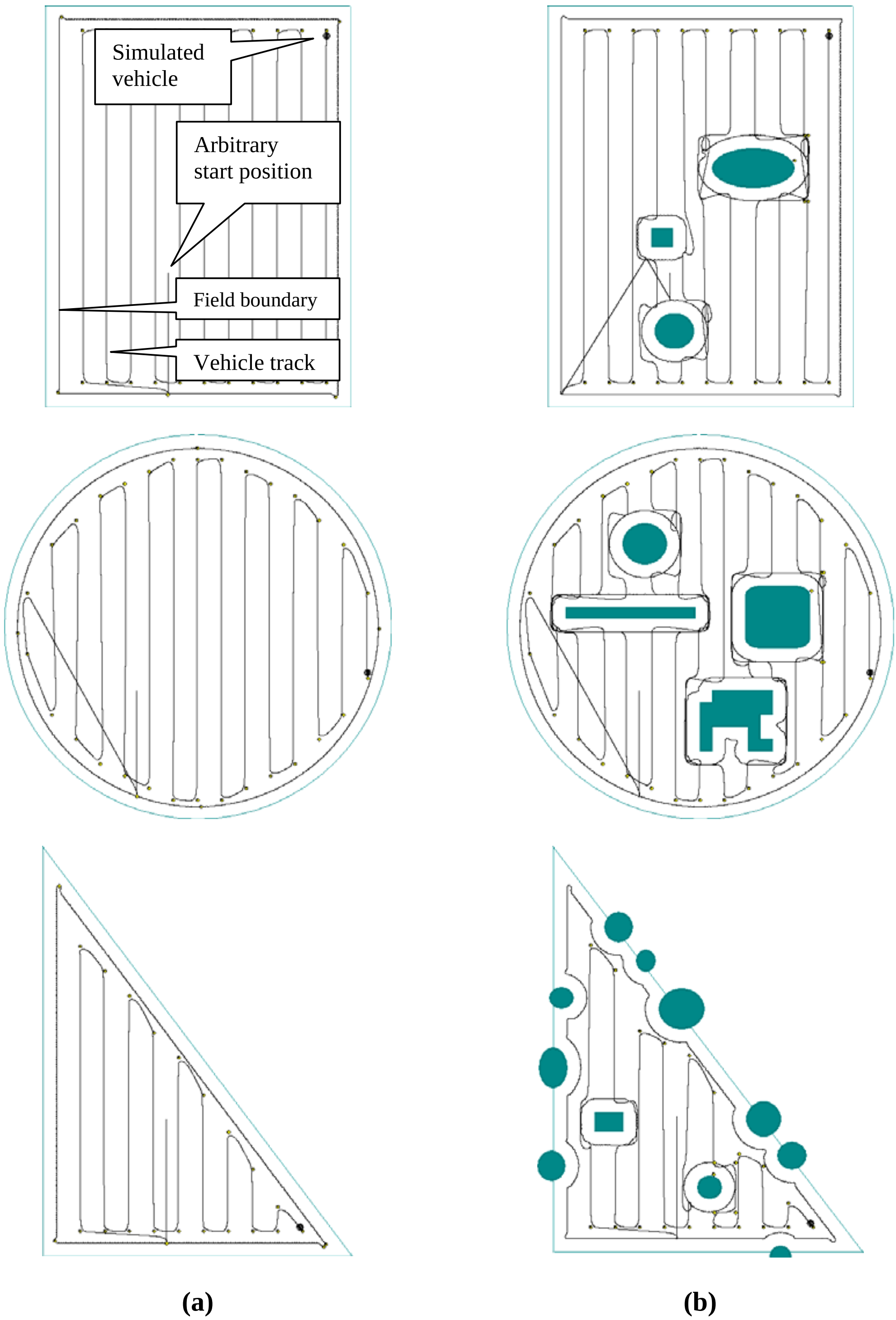

Figure 6. Results from simulation. (a) Simple field boundary.
(b) Boundary with static obstacles

3.2 Decomposition of the ‘plough a field’ Operation

Another example to demonstrate the decomposition procedure was to decompose the “plough a field” operation. The main tasks were identified and further decomposed but the operation has not yet been trialled.

Deterministic tasks identified were, ‘Plan to plough a field’, ‘plough’ and Navigate. Reactive tasks identified are ‘Deal with unknown objects’, ‘Threat’ and ‘React to internal changes’.

3.2.1 Deterministic task: Plan to plough a field

Resources	Tractor and plough details, field boundary, topography, soil type, depth, start date and time
Optimise	Width of lands and headlands Minimise distance travelled Cover whole area once Plough in a straight line Rates of turn, speed Minimise cost
Result	Route plan, treatment map, estimated finish time, fuel usage and cost

3.2.2 Deterministic task: Plough

Resources	Tractor, plough, desired and actual working depth, draft force, slip.
Optimise	Minimise deviation from route plan Keep parallel to previous row Constant depth / draft / slip
Result	Ploughed field

3.2.3 Deterministic task: Navigate

Resources	Tractor, Route plan
Optimise	Shortest distance to next waypoint
Result	Arrive at waypoint within specifications

3.2.4 Reactive task: Deal with unknown objects

Behaviour	Avoid obstacles
Sub behaviours	Watch and wait Wait for dynamic objects to move out of the way Explore stationary obstacles and record outline Return to route plan as soon as possible

3.2.5 Reactive task: Threat

Behaviour	Stop when objects move towards tractor
Sub behaviours	Watch and wait

3.2.6 Reactive task: React to internal changes

Behaviour	Graceful degradation due to partial systems failures
Sub behaviours	Nominal safe operation
	Safe operation with warnings
	Partial systems shutdown – mobile
	Partial systems shutdown – immobile
	Stopped – still communicating
	'Dead'

4. DISCUSSION

The key element in the future of mobile robots, is to find out how to embed enough intelligence into a machine to allow it to work by itself. No new hardware is needed, what is needed is a way to define what we want the tractor to do in predefined contexts. These machines will never be intelligent but if they can exhibit sensible behaviour over long periods of time, unattended, while carrying out a useful task, then they will have a place in future agricultural operations. This paper presents a new approach to develop a methodology that may allow us to define what an autonomous tractor should do in terms of how we (as humans) can define its behaviour. As the method shows the decomposition from top level processes, right down to primitive actions that can be defined in machine language, it can be used to build truly autonomous vehicles. It combines the advantages of determinism and relativism into a hybrid system that can be actually implemented in a vehicle.

Defining a common lexicon and semantics that can be used both in agriculture and robotics has significantly helped the collaboration between the authors in different institutions and countries. The use of functional descriptions, system diagrams and structured English that define the same process but from different perspective has also helped improve the clarity.

This method seems to make sense intuitively and works well for the simple Explore operation. The next step will be to apply this method in more detail to 'ploughing a field' or another typical agricultural operation.

5. CONCLUSIONS

A method for decomposing hierarchical agricultural Operations, Tasks, Optimisations and Behaviours into Primitive actions was presented. Definitions of these words have been suggested and their usage outlined as part of an overall method to better define what an autonomous tractor should actually do. It has taken both a top down as well as a bottom up approach that has resulted in a method that is understandable from an agricultural point of view as well as being able to be programmed into a robot.

The separation of deterministic tasks that can be optimised beforehand and reactive tasks that have real time behaviours in certain contexts is a novel approach especially when they can be recombined into a hybrid system that should allow them to work well in reality as it is close to what people do in reality.

6. ACKNOWLEDGEMENTS

We would like to thank Michael Noerremark for his help in driving a tractor demonstrating, filming and explaining the ploughing process.

7. REFERENCES

Arkin, R.C., 1998. *Behaviour based robotics.* The MIT Press, USA, 491 pp.

Billingsley, J., 2000. Automatic guidance of agricultural mobiles at the NCEA. *Industrial Robot* 27 (6), 449-457.

Blackmore, S., H. Have and S. Fountas. 2002a. A specification of behavioural requirements for an autonomous tractor. In: Automation technology for off-road equipment, edited by Zhang, Q. ASAE Publication, 33-42.

Blackmore, S., S. Fountas and H. HaveH. 2002b. A proposed system architecture to enable behavioural control of an autonomous tractor. In *Proc: Automation technology for off-road equipment,* edited by Zhang, Q. ASAE Publication, 13-23.

Brooks, R. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* RA-2, 14-23.

Brooks, R. 1999. *Cambrian Intelligence: The early history of the new AI.* The MIT Press, Cambridge, Massachusetts, 199 pp.

Elderen, E. van and G.H. Kroeze. 1994. *Operational decision masking models for arable and grassland farms*. Report no. 94-3. IMAG-DLO; Wageningen, the Netherlands, 74 pp.

Freyberger, F. and G. Jahns. 2000. Symbolic course description for semiautonomous agricultural vehicles. *Computers and Electronics in Agriculture* 25, 121-132.

Gat, E. 1998. Three-layer architectures. In *Proc: Artificial intelligence and mobile robots*, edited by Kortenkamp, D., Bonasso, R.P., and Murphy, R. The MIT Press, Massachusetts, USA, 195-210 pp.

Gonzalo, R. M. 2005. *Mobotsim, Mobile Robot Simulator*. www.mobotsoft.com.

Hassan, H., J. Simo and A. Crespo. 2001. Flexible real-time mobile robotic architecture based on behavioural models. *Engineering Applications of Artificial Intelligence* 14, 685-702.

Hollingum, J., 1999. Robots in agriculture. *Industrial Robot* 26 (6), 438-445.

Kondo, N. and K.C. Ting. 1998. *Robotics for Bioproduction Systems.* The Society for Engineering in Agricultural, food and biological systems ASAE Publication, 325 pp.

Konolige, K. and K. Myers. 1998. The Saphira architecture for autonomous mobile robots. In *Proc: Artificial intelligence and mobile robots*, edited by Kortenkamp, D., Bonasso, R.P., Murphy, R. The MIT Press, Massachusetts, USA, 211-242pp.

Kosecka, J., H.I. Christensen and R. Bajcsy. 1997. Experiments in behaviour composition. *Robotics and Autonomous Systems* 19, 287-298.

Murphy, R., 1998. Coordination and control sensing for mobility using action-oriented perception. In *Proc: Artificial intelligence and mobile robots*, edited by Kortenkamp, D., Bonasso, R.P., Murphy, R. The MIT Press, Massachusetts, USA, 142-157 pp.

Na, Y.-K., S.-Y. Oh. 2003. Hybrid control for autonomous mobile robot navigation using neural network based bahaviour modules and environment classification. *Autonomous Robots* 15, 193-206.

Pfeifer, R., Scheier, C., 1999. *Understanding intelligence.* The MIT Press, Massachusetts, USA, 697 pp.

Pilarski, T., M. Happold, H. Pangels, M. Ollis, K. Fitzpatrick and A. Stentz. 2002. The Demeter system for automated harvesting. *Autonomous Robots* 13, 1

Ridao, P., J. Batlle and M. Carreras. 2002. O2CA2, a new object oriented control architecture for autonomy: the reactive layer. *Control Engineering Practice* 10, 857-873.

Rzevski, G., 1995. *Mechatronics: Perception, Cognition and Execution*. Butterworth-Heinemann, Oxford, UK, 332 pp.

Sørensen, C.G., 1999. *A Bayesian Network Based Decision Support System for the Management of Field Operations. Case: Harvesting Operations*. Ph.D.-Thesis, Technical University of Denmark, 193 pp.

Sørensen, C.G. 2003. A Model for Field Machinery Capability and Logistics: the case of Manure Applications. *Agricultural Engineering Internationa.: the CIGR Journal of Scientific Research and Development*, Manuscrip PM 03 004, Vol 5, October 2003

Torrie, M.W., D.L. Cripps and J.P. Swensen. 2002. Joint architecture for unmanned ground vehicles (JAUGS) applied to autonomous agricultural vehicles. In *Proc: Automation technology for off-road equipment,* edited by Zhang, Q. ASAE Publication, 1-12.

Wilson, J.N., 2000. Guidance of agricultural vehicles - a historical perspective. *Computers and Electronics in Agriculture* 25, 1-9.

Yavuz, H., Bradshaw, A., 2002. A new conceptual approach to the design of hybrid control architecture for autonomous mobile robots. *Journal of Intelligent and Robotic Systems* 34, 1-26.