GLOBAL CONTROL FOR ROBOTIC EXCAVATION USING FUZZY LOGIC AND STATECHARTS

M Santos, Q P Ha, DC Rye and HF Durrant-Whyte
Australian Centre for Field Robotics
The Rose Street Building J04
The University of Sydney 2006 NSW, Australia.

{*Miguel|Quang.Ha|Rye|Hugh*}*@acfr.usyd.edu.au*

Abstract: This paper presents global strategies for the computer control of autonomous backhoe-type excavators. The control structure is divided into low and high levels. The low level control utilises fuzzy logic to encapsulate expert experience for capturing soil in many excavation scenarios. UML statecharts are used at the higher level for mapping environment and machine sensor data to actuator control signals. This mapping is based on a deep understanding of excavation performed by a skilful operator and is coded into rule sets. Transition between states or behaviours is accomplished via associated task characteristic functions that not only switch among tasks but also enable/disable rules according to digging phases. Typical excavation tasks are decomposed into statecharts and task elements. The control schemes are illustrated by autonomous trench digging. Field test results are provided to verify the validity of the proposed control architecture.

Keywords: robotic excavation, global control, fuzzy logic, UML, statecharts

1. INTRODUCTION

Earth moving is common at construction and mining sites. In moving toward the control of autonomous robotic excavators, appropriate strategies are required to control the machine actuators at both lower and higher levels. Lower level control requires the development of controllers that are robust to uncertainties associated with excavation operations [1]. For excavator dynamics control, kinematic and dynamic models that assume the hydraulic actuators act as infinitely powerful force sources are well presented in the literature [2]. Position control with a conventional PD controller is used in [3] for simulation of the digging process with limited contact information. Force control and impedance control has been addressed in [1,4] using a force model for the soil reaction.

Due to the complex interactions between the bucket and the environment, there exist no practical, accurate models of tool-soil contact. In fact, the machine reactions during digging change in real time depending on the soil characteristics. Cognitive control schemes are believed to be able to tackle the problem. Fuzzy logic control has been successfully used for automatic digging of a wheel loader [5]. In this work, fuzzy controllers are developed for low-level control in accordance with an algebraic architecture using statecharts at the higher level. The proposed architecture then involves behavioural-hierarchical control schemes that are based on the decomposition of typical excavation tasks into states or state elements.

Here, every task will be represented by a state in a statechart [6]. Unlike finite state machines [7], statecharts allow one to reuse components, to use concurrent states or to build complex states (nested states or superstates) if required. Approximate reasoning with fuzzy logic is incorporated in this mapping to encapsulate human expert knowledge in earthmoving operations. Task characteristic functions depending on state entry and exit conditions and rule sets are used for transition between task elements. Experimental results obtained through a number of field tests conducted on a mini-excavator verify the validity of the proposed global control strategies in robotic excavation.

2. EXCAVATION TASK DECOMPOSITION

In general, task decomposition for a robotic machine can be represented as a combination of task elements that form a basis for executing autonomous operations of the machine. Practically, excavation tasks can be decomposed into behaviours that activate an appropriate set of suitable controllers. The description of a particular behaviour is based mainly on observation and study of how excavators react when digging. The machine motion is composed of the motion of each actuator driving its working attachments. In excavation, task decomposition based on a task action space is proposed in [8] using task characteristic functions for transition. As soil is inhomogeneous and difficult to model, it is easier to decompose digging tasks into behaviour sets, and a behaviour is decomposed into its activity set by using finite state machines, as proposed in [5]. A task element base can contain one or more of the following feasible operations of the machine actuators:

- τ_1: Adjust the engine throttle to maintain a constant speed.
- τ_2 : Keep the current position for a certain time.
- τ_3 : Curl the bucket inward.
- τ_4 : Curl the bucket outward.
- τ_5 : Rotate the arm inward
- τ_6 : Rotate the arm outward
- τ_7 : Luff the boom up.
- τ_8 : Luff the boom down.
- τ_9 : Swing the boom to the right
- τ_{10} : Swing the boom to the left
- τ_{11}: Crawl the tracks forward.
- τ_{12} : Crawl the tracks backward
- τ_{13} : Lift the blade up.
- τ_{14} : Lower the blade down.
- τ_{15} : Crawl the tracks right.
- τ_{16} : Crawl the tracks left.

For example, a trenching task can be identified as having the following states:

1. Position the bucket over the trench start.
2. Lower the boom to the ground surface.
3. Penetrate the ground by curling the bucket.
4. Drag the bucket teeth in a straight line by moving simultaneously the arm and the boom. Notice that the error will be within a tolerable limit if the bucket tip does not find any obstacle or hard soil.
5. Curl the bucket to collect soil into the bucket.
6. Raise the boom out of contact with ground.
7. Dump the bucket contents at the side of the trench.
8. Check the necessity of doing another dig cycle according to the specified trench dimensions.
9. If necessary then repeat steps 1 to 8, else stop the task.

All axis movements involved in a particular state are known as the *main working axes* that are driven by the state when digging a trenching. Other axis movements can be used to deal with hard soils or obstacles. The bucket movement is of crucial importance in that particular task. States 2 to 6 are known as the digging portion of the process in direct relations to bucket-soil interactions that are very difficult to model. Mimicking the control of a human expert in these phases, a behaviour-based control approach combined with fuzzy controllers is believed to achieve with good performance autonomous excavation.

3. LOW-LEVEL FUZZY CONTROLLERS

Fuzzy control was mentioned in [8] as having promise in autonomous excavation for construction machines. The work [5] reported the use of fuzzy logic for automated digging by a Caterpillar wheel loader. Fuzzy rules were provided only for some limited cases of digging in these papers. Furthermore, it is not clear on how to adjust weighting schemes to combine multiple outputs and how to handle hard soil, obstacles or the cases when the bucket becomes stuck. Our design of fuzzy controllers is based on expert experience in excavation with a focus on modularity and compatibility of the controllers within global control schemes to achieve full autonomy of higher-level excavation tasks. Fuzzy logic controllers, **FLCi**, have been developed and implemented relating directly to feasible task elements τ_i. For example:

FLC3 and **FLC4** use the bucket cylinder pressures and bucket angular speed as inputs and the bucket spool valve opening area as the output to implement the task element “Curl the bucket inward” (τ_3) and “Curl the bucket outward” (τ_4) respectively.

FLC5 and **FLC6** use the arm cylinder pressures and arm angular speed as inputs and the arm spool valve opening area as the output to implement the *atomic states* (or task elements) “Rotate the arm inward” (τ_5) and “Rotate the arm outward” (τ_6).

FLC7 and **FLC8** use the boom cylinder pressures and boom angular speed as inputs and the boom spool valve opening area as the output to implement the atomic states “Luff the boom up ” (τ_7) and “Luff the boom down” (τ_8) respectively.

In addition to controllers relating to single task elements, supplementary controllers should be developed for compound (conditional or parallel) tasks. For example, to cause the bucket tip to follow a prescribed path, the bucket, arm and boom controllers should be activated simultaneously.

If the main working axis stalls because the bucket encounters very hard soil or rock, a modified controller **FLC7.1** based on **FLC7** is used.

The rule sets are in general heuristically formulated from observing skilful operators of earthmoving machines. For instance, the element task τ_5 “rotate the bucket inwards” is controlled by **FLC5** using the following rule set:

If (pressure is PL and angular_speed is PL) then (spool is PM)

If (pressure is PL and angular_speed is PS) then (spool is PL)

If (pressure is PL and angular_speed is ZR) then (spool is PL)

If (pressure is PM) then (spool is PL)

If (pressure is PS) then (spool is PL)

If (pressure is ZR) then (spool is PM),

where the linguistic labels are defined as PL=positive large, PM=positive medium, PS=positive small, ZR= zero, NS=negative small, NM=negative medium, and NL= negative large, and the pressure, angular speed and spool position are referred to the bucket axis. The centre of gravity method [9] is used for defuzzification. Weights can be incorporated within each fuzzy rule to allow for enabling or disabling, and for further adjusting rule activation if required.

4. HIGH-LEVEL BEHAVIOUR-BASED CONTROL WITH STATECHARTS

The basic philosophy behind the proposed control architecture is summarised as follows. A task algorithm is used to describe how the machine should attempt to reach a specified goal. In task planning, the task algorithm is decomposed into a number of (perhaps hierarchical) states by combining task elements (or atomic states) that do not need to further be decomposed. It follows that a task can also be composed from sub-states to form a superstate (a state of nested states). This principle allows for component reuse, and for building complex states if required.

Each action phase in earthmoving operations is represented as a state, and the states are represented as objects within the framework of the Unified Modelling Language (UML). UML statecharts [6] are chosen because of their advantages over finite state machines in terms of reuse, concurrency and flexible transition capabilities.

A statechart represents a state machine consisting of states and transitions between states, together with synchronisation states and pseudostates. A *state* is a representation of machine behaviour or a particular activity or strategy to follow and can be considered as behaviour or reactive task [10]. Every state object has entry and exit actions, and executes the particular behaviour or activity until a transition is set and the state is exited. *Synchronisation states* are state vertices that model the synchronisation of compound tasks or concurrent states. *Pseudostates* are used as initial, terminal or transient states that are visited for short periods of time and correspond to actions with near-zero execution times. *Transitions* are triggered by the receipt of events, and give a flow of execution control around a statechart by changing states. Transition conditions can be refined to have event priorities if more than one condition is true at the same time, or can be seen as if/else or switch/case structures. For transition between state elements, a c*haracteristic function* γ_i associated with the state S_i is defined here as follows:

$$\gamma_i = \begin{cases} 1, & \text{transition from } S_i \text{ to } S_j \\ 0, & S_i \text{ is active}, i = 1,2,\ldots,r \end{cases}$$

Activities are actions or behaviours that take a significant amount of time to execute, and must be interrupted by some condition (transition). Here, an activity instantiates the required fuzzy logic controller(s) described in section 4. *Actions* are non-interruptible behaviours executed instantaneously at particular points in the state machines. These actions may take place at the entry or exit of a state, or during a transition. *Atomic states* correspond to task elements that command a set of controllers based on feasible movements of the machine. Given a task decomposition, the associated states will be driven by a state machine using UML statecharts. A state is linked in real-time directly with the machine controllers using information obtained from sensors and estimators by activating the corresponding controllers.

To illustrate this structure, consider the task of digging a trench to a certain depth. Associated with each phase of the task decomposition (section 2), are states S_i, *i*=1,…,5 shown in figure 1. These states include LowerBoom (S_1), Penetrate (S_2), Drag (S_3), Capture (S_4) and LoadToTruck (S_5). The digging portion of the excavation work cycle and the dump cycle are considered as sub-machines of this statechart. The digging sub-machine is presented in figure 2. Transition between task elements is determined mainly by estimating if the digger has reached a position predetermined according to excavator expert heuristics by measuring the Cartesian error.

5. FIELD TEST RESULTS

Joint angle encoders and hydraulic cylinder pressure transducers allow for mapping task elements to spool position set-points for the boom, boom swing, arm, and bucket. In this way, the robotic excavator can execute a variety of digging tasks. In this section, the proposed global control architecture for autonomous operation is demonstrated through trench forming.

The goal is to dig autonomously a straight trench of one bucket width and a certain depth. It is assumed that the undisturbed soil surface is almost flat. A dig pass is considered “successful” if it results in a bucket that is at least 80% full.

Figure 3 shows the robotic excavator executing the task of digging a trench. The average task duration for the digging part of the trajectory one-pass loading is 15 seconds, an average time for a human operator to do this task.

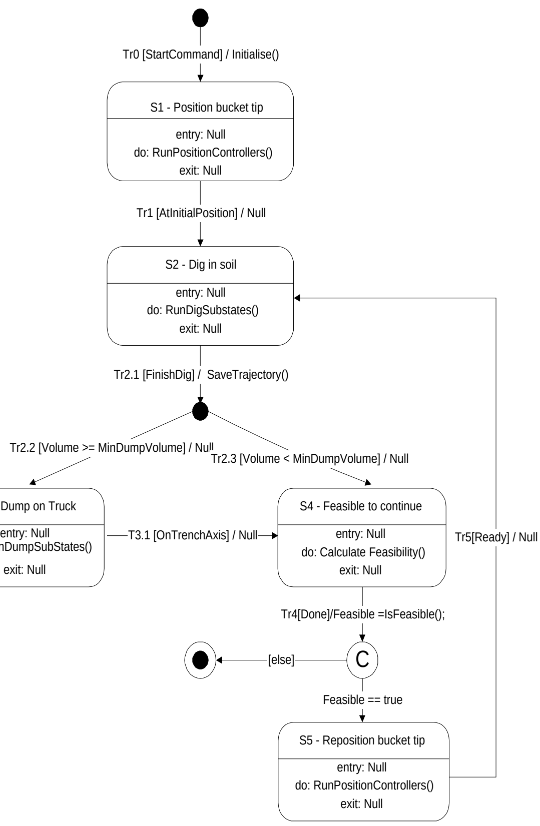


Figure 1. A statechart for trench forming.

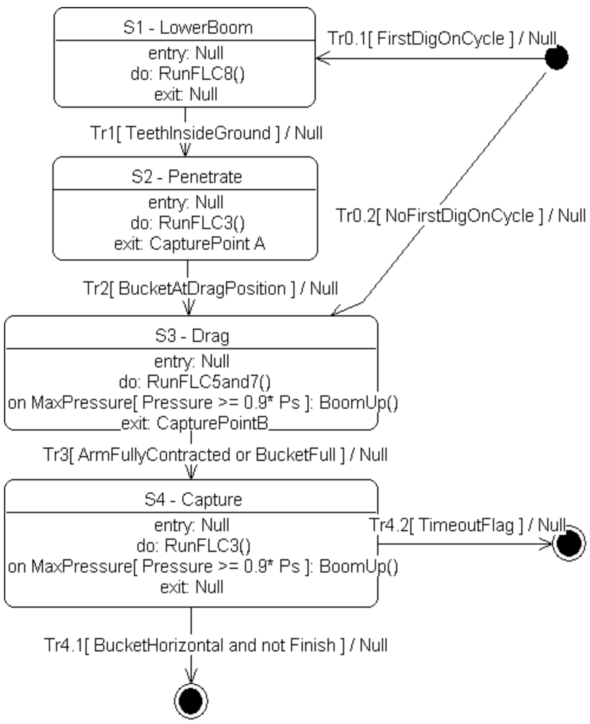


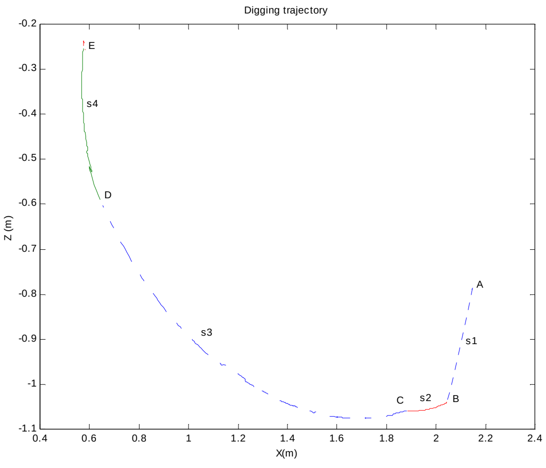
Figure 2. Digging submachine for trench forming.

Assume that the digging portion (figure 2) of the excavation work cycle is to be executed with the statechart shown in figure 1. This part of the work cycle involves the states LowerBoom (S_1), Penetrate (S_2), Drag (S_3) and Capture (S_4).

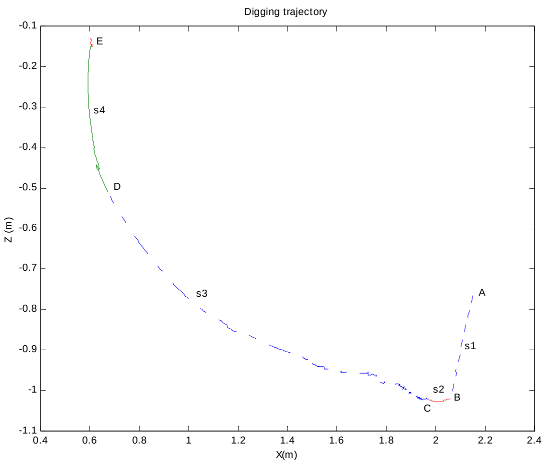
As an example, the implementation of state S_4 Capture is described here. The entry conditions stated in transition Tr_3 must be true. That is, if the arm is fully contracted or the bucket is full or the bucket tip is out of the soil then state S_4 will be executed by running the associated fuzzy logic controller **FLC3**. If the bucket is horizontal in the case of full bucket filling, then transition Tr_{41} is true and the next state DumpToTruck (S_5) will execute. Additionally, a time-out transition (Tr_{42}) is implemented so that if the bucket gets stuck or cannot finish the dig in an allocated time, the rule base will provide a new strategy to remedy the situation. This strategy could be discarding the contents if the captured volume is less than 50 % or else going to state S_5 .

Figure 3. Digging a trench.

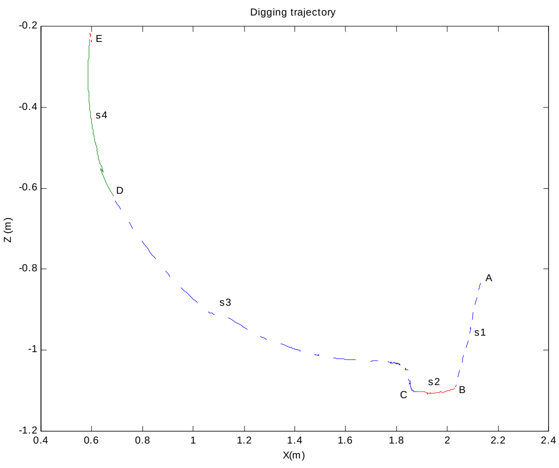
The recorded data for the entry and exit point of the dragging phase are employed to generate the desired trajectory in joint space for the next digging cycle. Field tests for executing the task of digging a trench have been conducted with the soil stiffness categorised as “soft”, “medium”, and “hard”. Figure 4 shows the measured Cartesian trajectory of the bucket tip corresponding to each phase for these categories. In the figure, AB, BC, CD, and DE correspond respectively to the phases S_1 , S_2 , S_3 , and S_4 . Observe that the bucket tip is slightly raised at the beginning of S_4 . This occurs because the boom differential pressure reading is high, so **FLC7** is activated for a short time. Without any obstacles or when the soil is not “hard”, the tip during the dragging phase is observed to satisfy a desired tolerance of 5 centimetres (figure 4a,b). The bucket tip cannot, however, penetrate "hard" soil smoothly (figure 4c) because the soil reaction force is too large to allow smooth penetration. The tip must scratch the surface in order to loosen the soil underneath.



(a)



(b)

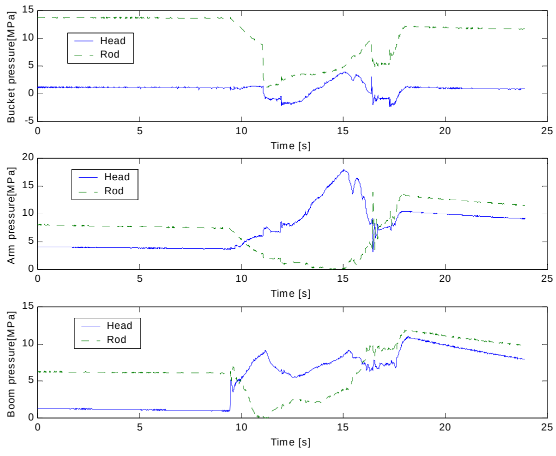


(c)

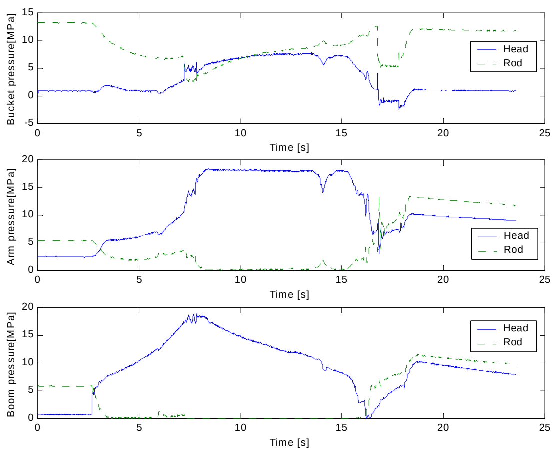
Figure 4. Bucket tip trajectory, digging (a) "soft", (b) "medium" and (c) "hard" soil.

Figure 5 shows the hydraulic pressures measured at the head side (solid line) and rod side (dashed line) of the bucket (top), arm (middle) and boom (bottom) cylinders for the cases of digging soft, medium, and hard soil respectively. The force generated by each cylinder can be estimated approximately from the pressure difference across the cylinder. It is observed that digging medium and hard soil requires large forces, as one would expect.

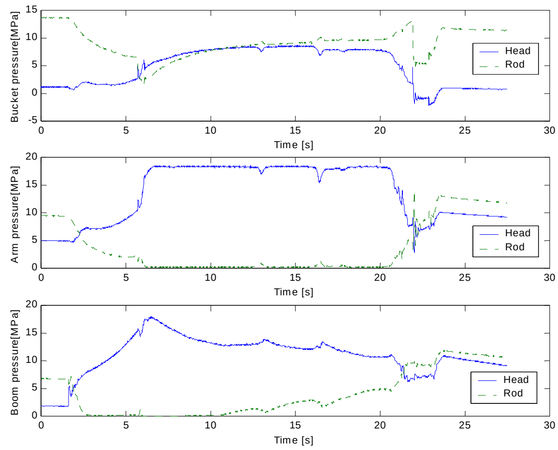
Figure 6 presents the percentage spool opening area of the bucket (top), arm (middle), and boom (bottom) as the outputs of the corresponding fuzzy logic controllers activated under the sub-machine shown in figure 2 during digging in soft, medium, and hard soil. The figure shows that the opening time of the spool is an important factor in dealing with hard soil, and can be used to handle obstacles or to decide whether the task can be executed.



(a)

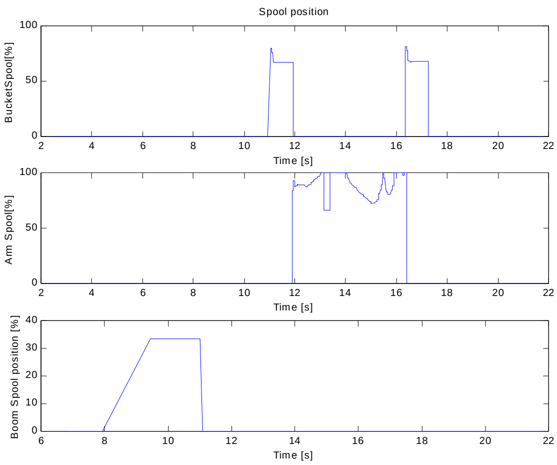


(b)

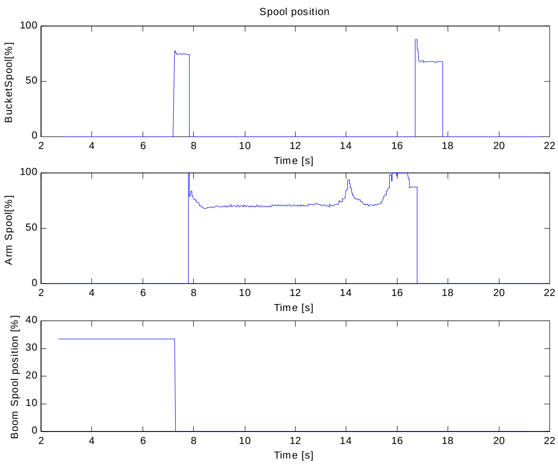


(c)

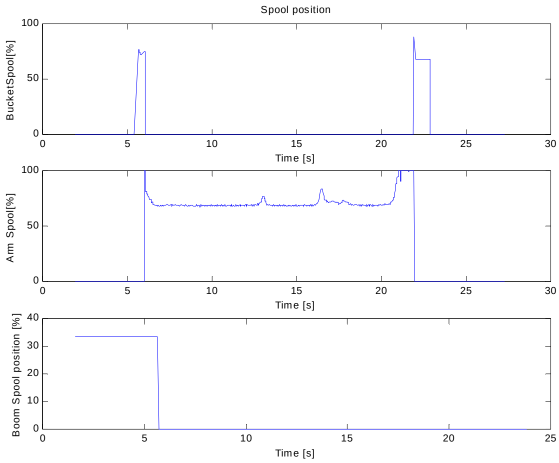
Figure 5. Pressure measurements during digging a trench in (a) "soft", (b) "medium" and (c) "hard" soil.



(a)



(b)



(c)

Figure 6. Spool commands during digging a trench in (a) "soft", (b) "medium" and (c) "hard" soil.

Experimental results obtained confirm the possibility of achieving fully autonomous execution of robotic excavation tasks using the proposed control architecture. It is believed that this methodology can be extended to any autonomous excavator in the context of coordinated control at all scales and with a variety of distinct operating regimes.

6. CONCLUSION

This paper has proposed a global control architecture for autonomous operations of robotic excavators. Fuzzy logic and statecharts have been applied to the autonomous execution of some excavation tasks of the robotic digger developed in our laboratory. Field tests of digging various types of soil show promising results. At this stage very hard soil or difficult shaped obstacles cannot be successfully handled. Further research will focus on the design and implementation of customised modules to handle hard soil and obstacles, execution of full dig cycles and the integration of behavioural and AI-based techniques with robust controllers that have been developed [1,4] to retain both operating flexibility and verifiable system integrity as well as high performance in the low-level control of the robotic machine.

Acknowledgments

Support from the Australian Research Committee, NS Komatsu Pty Ltd, and the Cooperative Research Centre for Mining Technology and Equipment is gratefully acknowledged.

REFERENCES

[1] Ha, Q.P., Nguyen, Q.H., Rye, D.C., & Durrant-Whyte, H.F. "Force and position control for electrohydraulic systems of a robotic excavator". *Proc. IAARC/IFAC/IEEE Int. Symp. on Automation and Robotics in Construction*, Madrid Spain, pp. 483-489, 1999.

[2] Vaha, P.K. and Skibniewski, M.J. "Dynamic model of an excavator". *J. Aerospace Engineering*, vol. 6, no. 2, pp. 148-158, 1993.

[3] Koivo, A.J., Thoma, M., Kocaoglan, E., & Andrade-Cetto, J. "Modeling and control of excavator dynamics during digging operation". *J. of Aerospace Engineering*, vol. 9, no. 1, pp. 10-18, 1996.

[4] Ha, Q.P., Nguyen, Q.H., Rye, D.C., & Durrant-Whyte, H.F. "Robust impedance control of excavator dynamics". *Proc. Int. Conf. on Field and Service Robotics*, Pittsburgh PA, pp. 226-231, 1999.

[5] Shi, X., Wang, F.Y., & Lever, P.J.A. "Task and behaviour formulations for robotic rock excavation". *Proc. IEEE Int. Symp. on Intelligent Control*, pp. 248-253, 1995.

[6] Douglas, B.P. *Real-time UML: developing efficient objects for embedded systems.* Addison-Wesley, Reading MA, 1999.

[7] Fok, K.Y. & Fabuka, M.R. "A automatic navigation system for vision-guided vehicles using a double heuristic and a finite state machine". *IEEE Trans. Robotics and Automation*, vol. 7, no. 1, pp. 181-188, 1991.

[8] Singh, S. "Synthesis of tactical plans for robotic excavation". PhD thesis, Carnegie Mellon University, Pittsburgh PA, 1995.

[9] Pedrycz, W. *Fuzzy control and fuzzy systems.* John Wiley & Sons, New York NY, 1993.

[10] Arkin, R.C. *Behaviour-based robotics.* MIT Press, Cambridge MA, 1998.