Agent-Based Control for Fuzzy Behavior Programming in Robotic Excavation

Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—This paper discusses the concept, formulation, and implementation of the agent-based control for fuzzy behavior programming in robotic excavation. Petri net transducers are introduced to describe excavation control agent coordination and specification, while fuzzy control rules are used to implement primitive motions. A prototype laboratory excavation system is built with PUMA robotic manipulators and a force/torque sensor. Extensive experiments have been conducted and the results have demonstrated that the proposed control method is capable of continuously adapting and replanning its actions based on sensory feedback, and completing its excavation tasks in dynamic and unstructured environments.**

***Index Terms*—Agent-based control (ABC), agent coordination, behavior programming, fuzzy logic, Petri net transducers, robotic excavation.**

I. Introduction

EARTHMOVING plays an essential role in activities such as mining, construction, and hazardous waste treatment. The recent trend toward greater automation of earthmoving machines, such as backhoes, loaders, shovels, and dozers, reflects a larger movement in related industries to improve productivity, efficiency, and safety. However, as pointed out in [19], automation of fieldworthy earthmovers offers great potentials but is a difficult problem since these machines must operate with limited computing and sensing powers in unstructured, dynamic, outdoor environments, often in poor visibility conditions and inclement weather.

Over the past decade, a wide range of research efforts have been taken in modeling, sensing, actuation, control strategies, and system architectures for automated earthmoving operation, and several enabling technologies relevant to earthmoving automation have been developed [3]–[5], [7], [8], [12]–[14], [16]–[20], [22]. For the control of automated excavators, much of the research work has been concentrated on backhoe type excavators [15], [21] and the active control methods proposed for the cutting blade have been the basis for many studies [1], [2], [6]. The major problem for such works is the requirement of a mathematical model for interactions of machines and environments. However, for excavation, especially rock excavation, it is impossible to predict what exactly will happen to machines because the characteristics of the materials at the excavation site cannot be predetermined, and the digging conditions change continuously as machines disturb the material. This complexity implies that automated excavation machines might not be able to utilize conventional control methods that are based on mathematical equations, since it is impractical or infeasible to develop analytical model to specify interaction between an excavation machine and its environment [18]. Thus, new control methods must be developed for such applications.

On the other hand, skilled human operators can achieve sophisticated control of excavation machines in dynamic, unstructured and unpredictable environments. This has lead to our work on using behavior programming, fuzzy control and neural networks for automated excavation since early 1990s [9]–[11], [16]–[18], [22]. The motivation for this type of approach is quite simple: Since the human control process of excavation tasks requires no analytic models of machine/environment interactions, it is advantageous to build an excavation control system using experience and knowledge from skilled human operators. So far, both laboratory testing with prototype excavation machines and field implementation in industrial wheel loaders and other type of vehicles have produced successful results and demonstrated great potentials for our proposed control method [18], [25], [26], [28].

This paper describes the first part of our work in robotic excavation, fuzzy agent algorithms and their laboratory experiments. The actual implementation of those algorithms in an industrial wheel loader and the corresponding field testing results will be discussed later. The paper is focused on the agent-based control for fuzzy behavior programming in robotic excavation. The basic idea is to modify and encapsulate various behavior programs we have developed for excavation into control agents so that they can travel and run in a networked environment. This will enable us to implement a robotic excavation control system of high performance and intelligence with limited computing power and memory space. The paper is organized as follows: Section II explains the basic concept of agent-based control; Section III introduces Petri net transducers for agent coordination and specification; Sections IV and V presents various control agents and primitive motions for excavation and their fuzzy logic implementations; several lab experiments conducted by using a PUMA robotic manipulator are described in details in Section VI, and Section VII concludes the paper with final remarks.

Manuscript received November 7, 2003; revised June 2, 2004. This work was supported in part by the Outstanding Young Scientist Research Fund from the National Natural Science Foundation (60125310), the 973 Key Project (2002CB312200) from the Ministry of Science and Technology, Shandong 863 Project (030335), and Key Project (60334020) from the NNSF, China, Caterpillar, Peoria, IL, and by the National Science Foundation.

The author is with the Key Laboratory of Complex Systems and Intelligent Science (LCSIS), Institute of Automation, the Chinese Academy of Sciences, Beijing 100080, China, and also with the Program for Advanced Research in Complex Systems (PARCS), Systems and Industrial Engineering Department, the University of Arizona, Tucson, AZ 85721 USA (e-mail: feiyue@sie.arizona.edu).

Digital Object Identifier 10.1109/TFUZZ.2004.832522

II. Agent-Based Control for Excavation

In the conventional behavior control approach, a robotic excavation is generally carried out through the three steps of *goal specification, task organization, behavior coordination*, and *action execution* [16], [24]. This approach is based on the constraint that the robotic excavation control system must be hosted and supported entirely by the in-vehicle computing facility on an operational site, normally at remote mining pits. In our new agent-based control approach, however, we assume that excavation vehicles are connected to a central operation center through a wireless network, normally via satellite communication as used currently by some Caterpillar wheel loaders but so far only for the purpose of service and maintenance. The availability of the wireless connectivity enables us to design a new control architecture that can utilize the additional networked computing power to execute excavation tasks more effectively and intelligently without actually increasing the cost of system implementation.

In the agent-based control for behavior programming in robotic excavation, the robotic excavation control system is divided into two parts: A local control system for task execution and a remote supervisor for dispatching and learning. The local excavation control system is supported by an on-site mining vehicle and conducts its tasks with a set of *default control agents* that are based locally and a set of *active control agents* that are provided by the remote supervisor via a network. When the communication is available and real-time responses are permitted, active control agents will perform excavation tasks and normally lead to better results in terms of productivity and damage to vehicles. If the network is off or a real-time response is not allowed, then either the current active control agents or the default control agents will conduct excavation tasks, since they are the only available control agents for task execution in those situations. The remote supervisor is hosted by a remote central operation center and connected to the on-site vehicle via a wireless network. The major function of the supervisor is to maintain, modify, and even design control agents for excavation, select and send a particular control agent or a set of control agents as active control agents for the optimal performance to the local control system based on the site characteristics, vehicle capacities, and on-line task information. The two key issues for the remote supervisor are effective learning algorithms for improving the performance of control agents and dispatching algorithms for determining when and which control agents should be selected and scheduled to be downloaded for local excavation tasks [27]. In this paper, however, we will not address those issues related to the remote supervisor. Instead, we will focus on the design, specification and implementation of the agent-based local excavation control systems.

For the agent-based control approach, a robotic excavation is conducted through *goal specification, task organization, agent coordination*, and *motion execution*. Generally, an excavation goal is either assigned by a human operator or selected by the control system from a set of predetermined goals. A goal is accomplished by organizing appropriate excavation tasks into sequences of executions. An excavation task is conducted by coordinating active or default control agents based on sensory feedback and vehicle conditions. Finally, a control agent executes its functionality through a sequence of primitive motion commands that are predesigned and specified by fuzzy logic decision rules.

III. Agent Coordination and Specification

As for design of intelligent machines [22], [23], Petri nets are used to formalize the process of agent coordination and specification. This is accomplished by modeling tasks and agents using transducers (PNTs). A PNT for an excavation task specifies all the feasible sequences of agents for its completion, while a PNT for a control agent defines all the possible sequences of primitive motions for its execution. Another commonly used model for agent coordination and specification is the finite-state machine (FSM). The major advantage of Petri nets over finite state machines is the capability of Petri nets to represent cooperation, parallelism, and conflict in shared resources and environments [29].

A PNT, M M, is a six-taple [22]

$$M \equiv (N,\ \Sigma,\ \Delta,\ \sigma,\ \mu,\ F)$$

where

i) $N \equiv (P,\ T,\ I,\ O)$ is a Petri net with initial marking μ;
ii) Σ is a finite input alphabet;
iii) Δ is a finite output alphabet;
iv) σ is a translation mapping from $T \times (\Sigma \cup \{\lambda\})$ to finite set of Δ^*;
v) $F \subseteq R(\mu)$ is a set of final markings;

and λ represents the empty string, Δ^* the set of all strings over Δ, and $R(\mu)$ the reachable set of markings from μ.

In the actual modeling, Σ represents the set of resources or inputs to PNT, Δ the set of commands or outputs by PNT, Petri net N and mapping σ determine the state transition and task execution, μ and F specify the initial state of and the set of the final states to be reached by PNT, respectively.

Using PNT, the process of agent coordination for an excavation task TC can be specified as

$$\mathrm{TC} \equiv (N_t, \Sigma_a, \Delta_a, \sigma_t, \mu_t, F_t)$$

where

i) $N_t \equiv (P_t, T_t, I_t, O_t)$ is a Petri net with initial marking μ_t underlying all the feasible sequences of control agents for the task completion;
ii) Σ_a is the set of control agents applicable to TC;
iii) Δ_a is the set of active or default control agents applicable to TC;
iv) $\sigma_t\ :\ T_t \times (\Sigma_a \cup \{\lambda\}) \rightarrow \Delta_a^*$ is a mapping that $\sigma_t(t, \mathrm{CA}) = \mathrm{CA}$ when CA is an active or default agent otherwise $\sigma_t(t, \mathrm{CA})$ specifies an alterative agent or agent sequence for executing CA;
v) $F_t \subseteq R(\mu_t)$ is the set of terminal markings indicating the task completion.

Fig. 1 illustrates this modeling process for excavation task: *unearth-an-oversize-particle*. The Petri net in Fig. 1 specifies

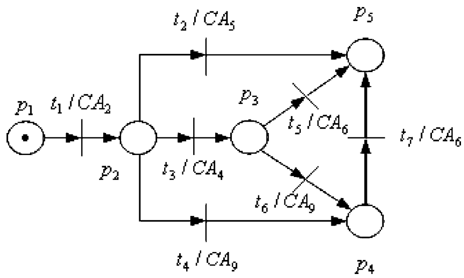

Fig. 1. PNT for task of "*unearth-an-oversized-particle*."

TABLE I
TYPICAL CONTROL AGENTS FOR EXCAVATION TASKS

Name	*CA₁ : Entry-point-mover*
Move the bucket to the initial loading location as instructed by the higher control level or other modules in the control system;	
Name	*CA₂: Horizontal-digger*
Make the bucket dig forward horizontally along a preset elevation, and small deviations above or below the preset elevation is allowed;	
Name	*CA₃: Over-particle-follower*
Move the bucket to load over the top of encountered over-sized particles by digging in parallel to the surface of the particles;	
Name	*CA₄:Under-particle-follower*
Make the bucket to dig down or underneath encountered over-sized particles within a preset limit on the rotating down angle;	
Name	*CA₅: Bucket-lifter*
Lift the bucket from the current position until it clears the pile, and make it hoist forward and up whenever possible to load more materials;	
Name	*CA₆:Bucket-extractor*
Recover from failures of executing other control agents by extracting the bucket back to a preset position without considering the lose of loaded materials;	
Name	*CA₇:Floor-follower*
Make the bucket to clear the excavation site by loading forward while maintaining the contact with the floor of the excavation site;	
Name	*CA₈:Down-digger*
Make the bucket to excavate in a downward direction while keeping its orientation at a preset attack angle, and small deviations from the preset attack angle are allowed;	
Name	*CA₉:Up-digger*
Make the bucket to extract over-sized particles within the material pile or from the excavation ground and an special effort is made to prevent the particles from rolling out of the bucket;	
Name	*CA₁₀:Pusher*
Make the bucket to push over-sized particles away from the loading path to a location determined by other modules of the control system.	

the coordination process of control agents for this specific task, where

$$\begin{aligned}\Sigma_a = \Delta_a = \{&\mathrm{CA}_2 : \text{hortizontal-digger}\\ &\mathrm{CA}_4 : \text{under-particle-follower}\\ &\mathrm{CA}_9 : \text{up-digger}\\ &\mathrm{CA}_6 : \text{bucket-extractor}\\ &\mathrm{CA}_5 : \text{bucket-lifter}\}\\ F_t = \{&(0,0,0,0,1)\}.\end{aligned}$$

A detailed description of control agents for robotic excavation is given by Table I.

Note that at places p_2 and p_3, decision conflicts occur and a selection or specification of the most appropriate control agent for the current situation must be decided based on the sensory feedback or external input information.

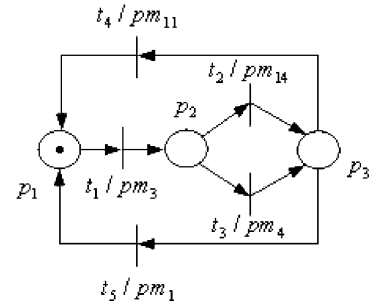

Fig. 2. PNT for control agent *under-particle-follower*.

Similarly, the execution process of primitive motions for an excavation control agent can be specified by a PNT as

$$\mathrm{CA} \equiv (N_a, \Sigma_p, \Delta_p, \sigma_a, \mu_a, F_a)$$

where

i) $N_a \equiv (P_a, T_a, I_a, O_a)$ is a Petri net with initial marking μ_a underlying all the possible sequences of primitive motions for the agent execution;
ii) $\Sigma_p = \Delta_p$ is the set of primitive motions applicable to control agent CA;
iii) $\sigma_a : T_a \times (\Sigma_p \cup \{\lambda\}) \rightarrow \Delta_p^*$, a simple mapping $\sigma_a(t,p) = p$ is assumed here;
iv) $F_a \subseteq R(\mu_a)$ is the set of terminal marking indicating the completion of agent execution.

Fig. 2 is an example of PNT for excavation control agent: *under-particle-follower*. The Petri net in Fig. 2 specifies the execution process of primitive motions for the control agent, where

$$\begin{aligned}\Sigma_p = \Delta_p &= \{\mathrm{pm}_1, \mathrm{pm}_3, \mathrm{pm}_4, \mathrm{pm}_{11}, \mathrm{pm}_{14}\}\\ F_a &= \{(1,0,0)\}.\end{aligned}$$

Table II in the next section presents the complete list of primitive motions for robotic excavation used in this paper.

Clearly, the PNT in Fig. 2 defines four possible sequences in one execution cycle for control agent *under-particle-follower*

i) $\mathrm{pm}_3 \rightarrow \mathrm{pm}_4 \rightarrow \mathrm{pm}_1$;
ii) $\mathrm{pm}_3 \rightarrow \mathrm{pm}_4 \rightarrow \mathrm{pm}_{11}$;
iii) $\mathrm{pm}_3 \rightarrow \mathrm{pm}_{14} \rightarrow \mathrm{pm}_1$;
iv) $\mathrm{pm}_3 \rightarrow \mathrm{pm}_{14} \rightarrow \mathrm{pm}_{11}$.

Online and real-time force/torque feedback information must be used to determine which sequence to be used to carry out the function of the control agent in the actual excavation.

IV. CONTROL AGENTS AND PRIMITIVE MOTIONS

Since 1992, we have developed a large collection of behavior programs for a robotic excavation experimental setup at the University of Arizona, Tucson [17], [18]. Those excavation behaviors have been constructed and formulated based on onsite and/or video observation of skilled human operators of wheel loaders at several open pit mines. Recently, we have modified and encapsulated those behaviors into control agents so that they can travel and run in a networked environment.

TABLE II
PRIMITIVE MOTIONS FOR EXCAVATION CONTROL AGENTS.

Symbol	*Name*
pm_1	*move forward parallel to bucket bottom*
pm_2	*move upward perpendicular to bucket bottom*
pm_3	*move backward parallel to bucket bottom*
pm_4	*move downward perpendicular to bucket bottom*
pm_5	*move forward up*
pm_6	*move backward up*
pm_7	*move forward down*
pm_8	*move backward down*
pm_9	*dig forward and rotate up*
pm_{10}	*dig backward and rotate down*
pm_{11}	*dig forward down and rotate down*
pm_{12}	*dig backward down and rotate up*
pm_{13}	*rotate up*
pm_{14}	*rotate down*

Table I presents some control agents for robotic excavation frequently used in our laboratory experiments that demonstrate the versatile abilities achieved by a typical excavation machine. A more comprehensive list of control agents for autonomous truck loading can be found in [28].

From the functions described in Table I, it is clear that those control agents have the following features: 1) the functional specification of an individual control agent is too broad to be accomplished by executing a single simple robotic motion; and 2) many terms used in those descriptions are too ambiguous or at least not specific enough for precise specification, lest accurately predesigned and preprogrammed robotic motions for their completion. To implement those control agents for excavation, primitive motions designed for accomplishing simple excavation tasks are introduced.

Table II lists primitive motions applicable for control agents to utilize for their task execution. Those primitive motions are developed based on the analysis of basic bucket digging actions of front-end-loaders. Note that the bucket motion is planar since both its translation and rotation are confined to the vertical plane parallel to the main shaft of the loading vehicle.

Those primitive motions are control commands that would translate the available sensory feedback information, mainly force/torque signals, to bucket motions. Since it is very difficult to interpret those force/torque signals precisely, it is quite natural to pursue fuzzy logic for their representation and implementation, as described in the following section.

V. FUZZY LOGIC CONTROL RULES FOR PRIMITIVE MOTIONS OF ROBOTIC EXCAVATION

In our experimental robotic excavation, only force/torque sensory data are actually used to infer the bucket/environment interaction. However, since it is not possible to determine precisely the status of interaction and very difficult to specify accurately the useful patterns for digging conditions from the force/torque sensory information, fuzzy-logic based control rules are used to express primitive motions for excavation.

Fig. 3 shows the coordinate system for the robotic arm, bucket, and wrist force/torque sensor. Let F and ΔF be the force/torque vector and the force/torque variation during the two consecutive sampling points, and M be the motion control vector for bucket, that is

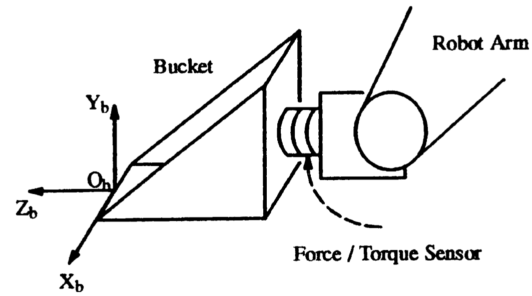

Fig. 3. Coordinate system for robotic excavation.

$$\begin{aligned} F &= (F_x, F_y, F_z, M_x, M_y, M_z)^{\mathrm{T}} \\ \Delta F &= (\Delta F_x, \Delta F_y, \Delta F_z, \Delta M_x, \Delta M_y, \Delta M_z)^{\mathrm{T}} \\ M &= (\Delta_y, \Delta_z, \Delta_r, BV)^{\mathrm{T}} \end{aligned}$$

where Δ_y and Δ_z are the translational change in the bucket position along y and z axes, Δ_r the rotational change in the bucket orientation along x axis and BV is bucket velocity. Variation ΔF in force/torque readings are used to detect collisions between the bucket and rock particles since it generally is very difficult or even impossible to differentiate normal resistance from bucket motion by using instant force/torque information alone.

As an example, primitive motion *dig-forward-and-rotate-up* (pm_9) is specified by the following set of fuzzy decision rules:

$$\begin{aligned}
&\textbf{Rule 1} - \text{IF } M_x \text{ is PL, THEN } \Delta_y \text{ is ZR}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PS, and } BV \text{ is PM} \\
&\textbf{Rule 2} - \text{IF } M_x \text{ is PS, THEN } \Delta_y \text{ is PM}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PL, and } BV \text{ is PL} \\
&\textbf{Rule 3} - \text{IF } M_x \text{ is ZR, THEN } \Delta_y \text{ is PM}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PL, and } BV \text{ is PL} \\
&\textbf{Rule 4} - \text{IF } M_x \text{ is NS, THEN } \Delta_y \text{ is PM}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PL, and } BV \text{ is PL;} \\
&\textbf{Rule 5} - \text{IF } M_x \text{ is NL, THEN } \Delta_y \text{ is PL}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PL, and } BV \text{ is PL;} \\
&\textbf{Rule 6} - \text{IF } F_z \text{ is NL, THEN } \Delta_y \text{ is PL}, \Delta_z \text{ is PS} \\ &\qquad \Delta_r \text{ is PS, and } BV \text{ is PS;} \\
&\textbf{Rule 7} - \text{IF } F_z \text{ is NS, THEN } \Delta_y \text{ is PL}, \Delta_z \text{ is PM} \\ &\qquad \Delta_r \text{ is PS, and } BV \text{ is PM;} \\
&\textbf{Rule 8} - \text{IF } F_z \text{ is ZR, THEN } \Delta_y \text{ is PL}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PS, and } BV \text{ is PL;} \\
&\textbf{Rule 9} - \text{IF } F_z \text{ is PS, THEN } \Delta_y \text{ is PL}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PS, and } BV \text{ is PL;} \\
&\textbf{Rule 10} - \text{IF } F_z \text{ is PL, THEN } \Delta_y \text{ is PL}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PS, and } BV \text{ is PL;} \\
&\textbf{Rule 11} - \text{IF } \Delta F_y \text{ is PL, THEN } \Delta_y \text{ is PL}, \Delta_z \text{ is PL} \\ &\qquad \Delta_r \text{ is PL, and } BV \text{ is PL;}
\end{aligned}$$

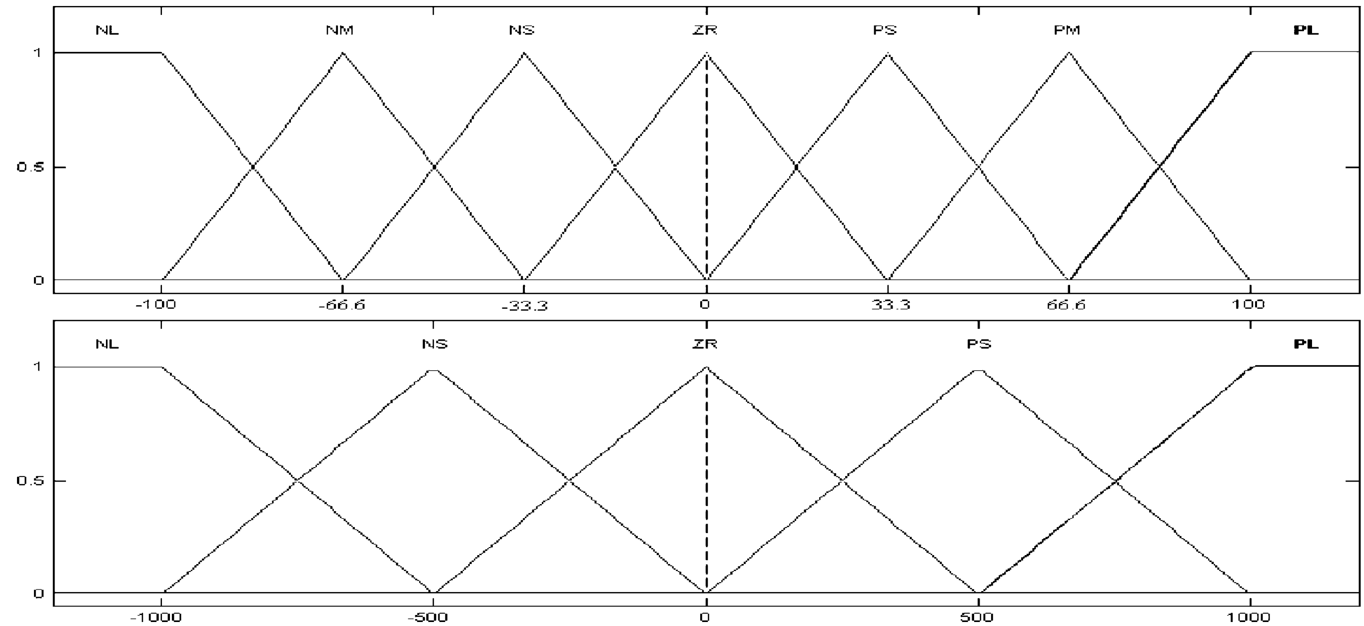

Fig. 4. Membership functions for fuzzy linguistic terms.

$$\begin{aligned}
&\textbf{Rule 12} - \text{IF } \Delta F_y \text{ is PS, THEN } \Delta_y \text{ is PM}, \Delta_z \text{ is PL} \\
&\qquad \Delta_r \text{ is PM, and } BV \text{ is PL;} \\
&\textbf{Rule 13} - \text{IF } \Delta F_y \text{ is NS, THEN } \Delta_y \text{ is PS}, \Delta_z \text{ is PL} \\
&\qquad \Delta_r \text{ is PL, and } BV \text{ is PL} \\
&\textbf{Rule 14} - \text{IF } \Delta F_z \text{ is NL, THEN } \Delta_y \text{ is ZR}, \Delta_z \text{ is ZR} \\
&\qquad \Delta_r \text{ is PS, and } BV \text{ is PS.}
\end{aligned}$$

where (PL, PM, PS, ZR, NS, NM, NL) is a set of fuzzy linguistic terms for positive large, positive medium, positive small, zero, negative small, negative medium, and negative large, respectively. Fig. 4 illustrates membership functions for those terms. Note that in the actual computation, the universe of discourse of the input and output signals are linearly normalized to intervals $[-1000, 1000]$ and $[-100, 100]$, respectively. A large scale factor is used for input signals in order to achieve high resolution, while a small scale factor is applied to evaluate output controls since normally their values are quite small in our laboratory experiments.

Using the standard procedure for fuzzy reasoning and defuzzification, a crisp value for conducting primitive motion *dig-forward-and-rotate-up* can be obtained from the previous decision rule set once the current force/torque reading and its variation are available.

In general, a primitive motion pm_i can be implemented as a set of fuzzy decision rules as

$$\begin{aligned}
&\textbf{Rule 1} - \text{IF } F \text{ is } Q_{i1} \text{ and } \Delta F \text{ is } \Delta Q_{i1}, \text{ THEN } M \text{ is } U_{i1}; \\
&\qquad\qquad \vdots \\
&\textbf{Rule } \mathbf{n_i} - \text{IF } F \text{ is } Q_{in_i} \text{ and } \Delta F \text{ is } \Delta Q_{i1}, \text{ THEN } M \text{ is } U_{in_i}
\end{aligned}$$

where n_i is the number of rules for pm_i, and Q_{i1}, ΔQ_{i1} and U_{i1} are linguistic terms for bucket position and orientation, force/torque signals and their variation, and motion control commands, respectively.

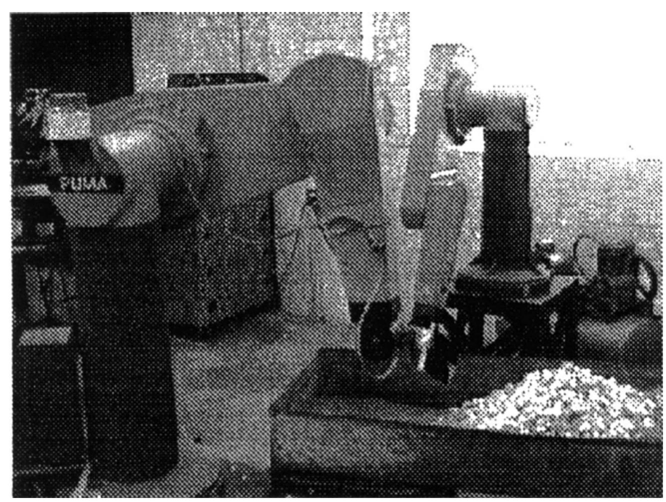

Fig. 5. Prototype experimental setup for robotic excavation.

As we have pointed out in Section III, at a particular marking for a PNT of a control agent, more than one primitive motion might be enabled and compete for task execution. Therefore, an arbitration process must be conducted to determine which primitive motion will be selected for actual action (or firing in terms of the language of Petri nets). A reasonable rule of arbitration is to select primitive motions based on the levels at which the preconditions of their fuzzy decision rules are satisfied. In this paper, the average value of input membership functions of a primitive motion is used for arbitration, i.e., the primitive motion with the highest average input membership function value, which is enabled under the current marking of an excavation control agent, will be selected for task execution. When the highest average input membership function value of a control agent is below a threshold, a new control agent must be called upon locally or over the network to replace the current executing control agent to continue the task.

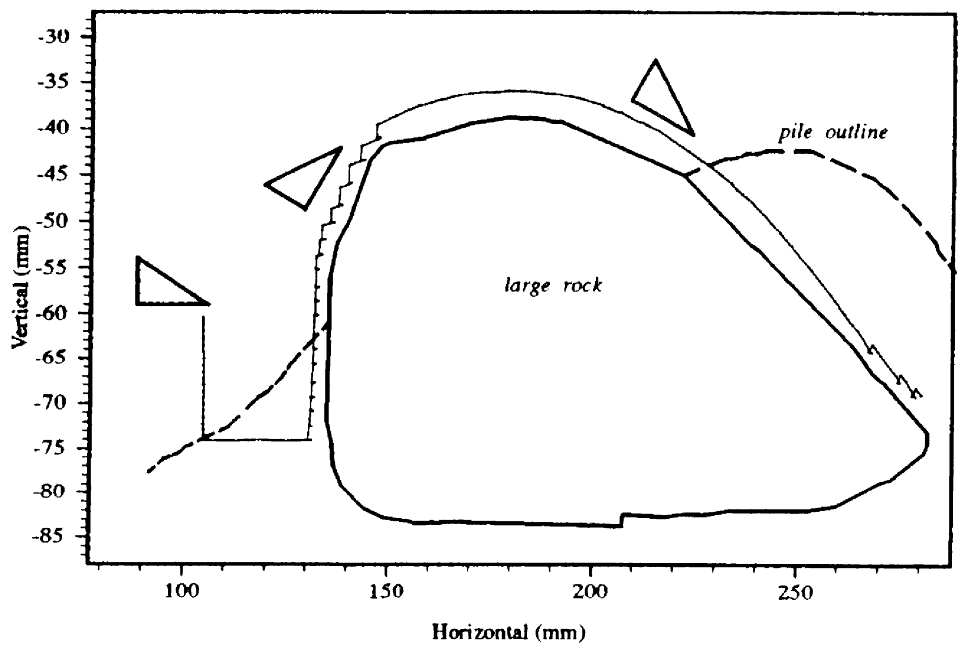

Fig. 6. Bucket trajectory of running *over-particle-follower*.

A similar rule of arbitration is also introduced to resolve the competition for execution by control agents in a PNT for an excavation task. Define the highest average value of input membership functions of all enabled primitive motions of a control agent as its firing strength; the control agent with the highest firing strength, which are enabled under the current marking of the excavation task, will be selected for execution. Various experimental results conducted in the laboratory have demonstrated the effectiveness of those simple arbitration rules, as one can see in the following section.

VI. Laboratory Experiments

To test and demonstrate the effectiveness of the proposed agent-based control method for robotic excavation, a prototype experimental setup has been developed at the Robotics and Automation Laboratory of the University of Arizona, Tucson, with a joint effort in control hardware design at the Key Laboratory for Complex Systems and Intelligence Science of the Chinese Academy of Sciences, Beijing, China. The setup consists mainly of an excavation site, a force/torque sensor, a PUMA robotic arm with excavation bucket, an interface board for receiving and hosting control agents, and a remote supervisory control that maintain all excavation control agents and communicate excavation tasks and agents to the interface board and PUMA arm via Internet (see Fig. 5).

A. Simple Excavation Tasks

The first set of robotic experiments was designed to test and evaluate the performance of individual control agents for simple excavation tasks. Fig. 6 shows the results of excavation control agent *over-particle-follower*. The function of this control agent is to make the orientation of the bucket follow the outline of the particle by rotating the bucket tip upwards when passing over the front of the rock particle and rotating the tip downwards while moving down along the back of the rock. In this way, the bucket will try to dig parallel to the rock particle surface. This feature has enabled the bucket to reduce the resistive forces encountered during the excavation, which leads to a quick climb and/or descend in the front and/or back of the particle. This agent is very useful for various robotic excavation tasks since it can be used to uncover buried (invisible) objects of any shapes without damaging both buried objects and the bucket. Note that in Fig. 6 the bucket tip trajectory at the initial digging is not very smooth. Fig. 7 illustrates the performance of control agent *under-particle-follower* when it is used to excavate a large rock. This experimental run shows the bucket has to contact the rock twice before it can proceed underneath the rock; a behavior mimics the actions of a human using a shovel to dig under a rock. The actual executing record indicates that in this case the control agent first repeats the sequence of primitive motions: *move-backward* (pm_3), *move-down* (pm_4), and *move-forward down-and-rotate-up* (pm_{11}) until the bucket tip reaches the point vertically below the initial contact point, and then starts a sequence of *rotate-up* (pm_{14}) and *move-forward*(pm_1) until the bucket is horizontal. Note that in the last sequence primitive motion *move-backward* (pm_3) is called upon but not activated.

The results of those two experiments, as well as others, have demonstrated the adaptability of robotic excavation control agents to unknown conditions and their effectiveness for simple tasks in unstructured environments.

B. Complex Excavation Tasks

The second set of experiments was designed to validate and evaluate the capability and performance of coordinating control agents for completing complex robotic excavation tasks. Figs. 8–14 presents the experimental results of achieving the excavation task of "*loading along a horizontal plane and removing*

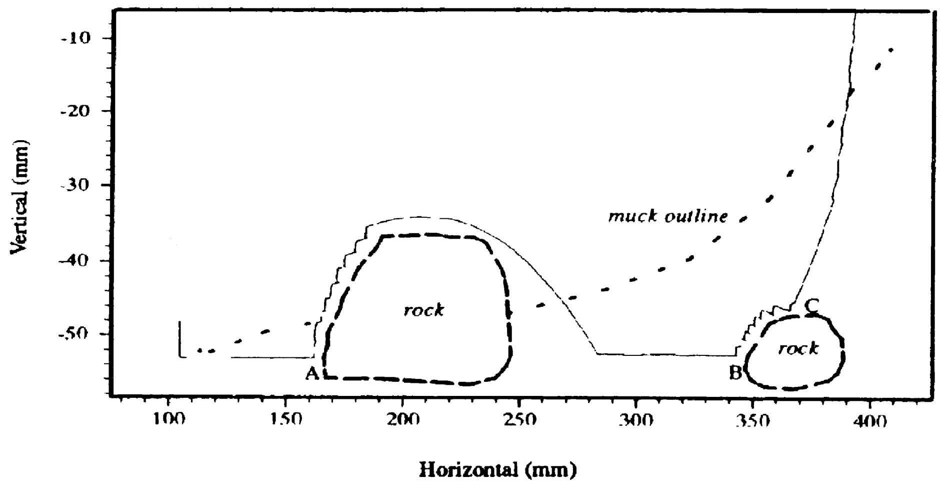

Fig. 8. Bucket trajectory during task execution: One.

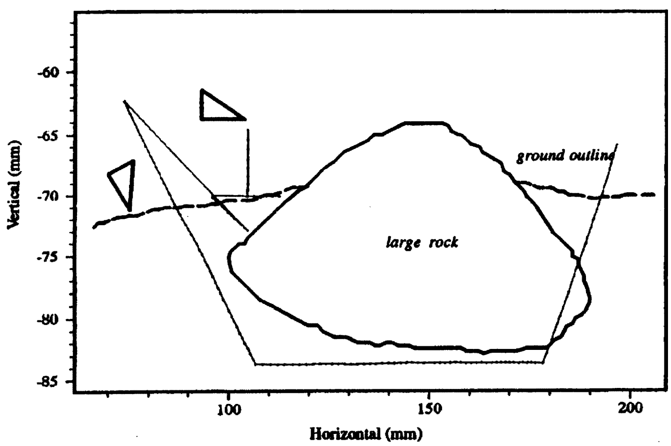

Fig. 7. Bucket trajectory of running *under-particle-follower*.

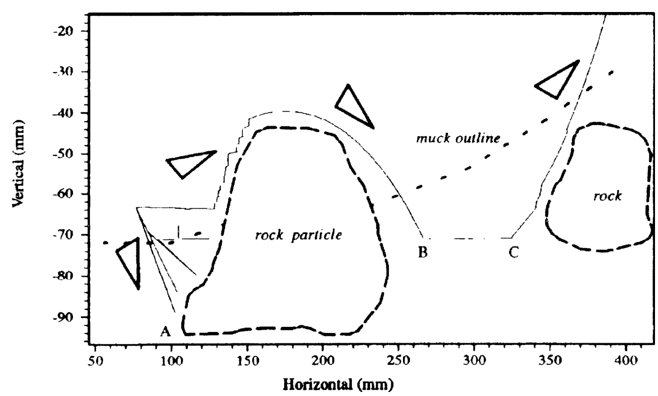

Fig. 9. Bucket trajectory during task execution: Two.

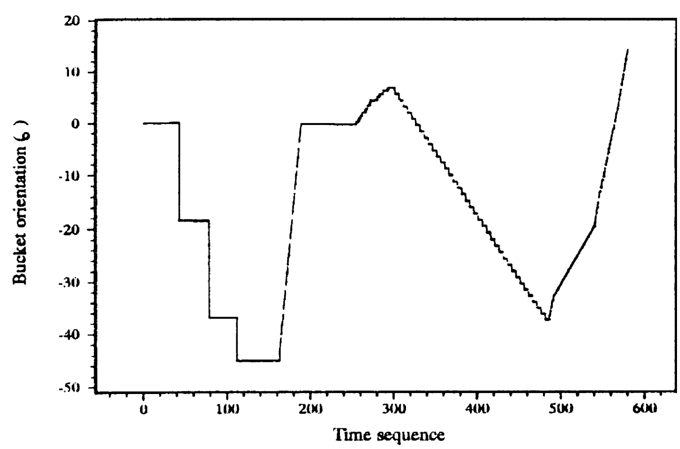

Fig. 10. Bucket elevation during task execution.

buried rocks when encountered, or loading over the rocks to expose its surface if they cannot be removed." Two rocks were buried in the excavation site for those experiments.

Fig. 8 illustrates the tip trajectory of the bucket during one experiment. At the beginning, control agent *entry-point-mover* takes the bucket to the selected entry point and when the bucket contacts the rock pile, agent *horizontal-digger* is activated until a large rock is encountered at point A. At this moment, the force/torque feedback infers a surface slopping upwards and away from the bucket, indicating that it may be easier to load over the top of the rock, and thus agent *over-particle-follower* is called for action. The bucket passes successfully over the top and down the back of the rock until it reaches its initial excavation elevation. Since the bucket is not perceived to be full, agent *horizontal-digger* is selected again to continue the excavation task. At point B, agent *over-particle-follower* is called up once again since another rock is detected and the bucket has loaded some materials already. Finally, at point C the bucket is perceived to be full and control agent *bucket-lifter* is employed to complete the loading task.

Another experiment is given in Fig. 9. In this case, the bucket contacts the surface of a large particle and digs down to remove it with control agent *under-particle-follower*. This agent is selected since the bucket is almost empty and the forces F_y and F_z indicates the rock is not facing upwards steeply which usually means the rock may not be buried deeply. Executing this agent until point A, the arbitration rule decides to abort it since the bucket can not rotate down any further, and agent bucket-extractor is selected to recover the bucket and followed by agent *over-particle-follower* until reaching point B where the bucket comes back to its original digging elevation and agent *horizontal-digger* is then activated. The bucket is full at point C and is removed by agent *bucket-lifter*. Figs. 10 and 11 present the

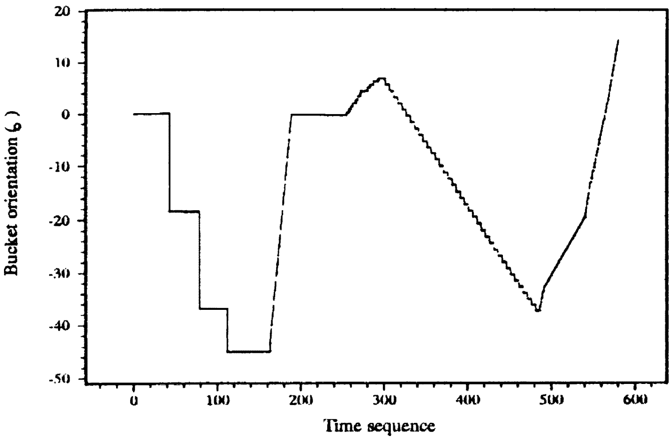

Fig. 11. Bucket orientation during task execution.

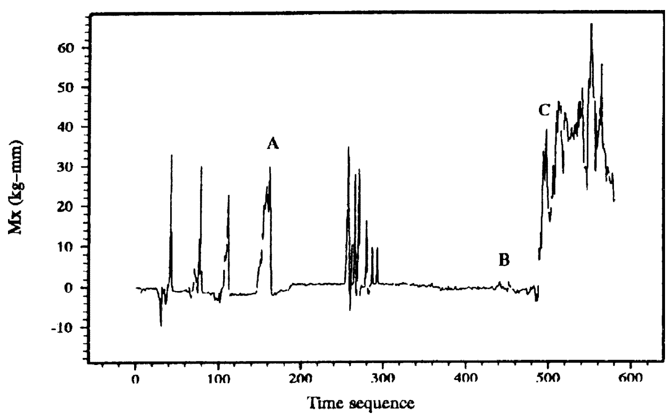

Fig. 12. Torque M_x during task execution.

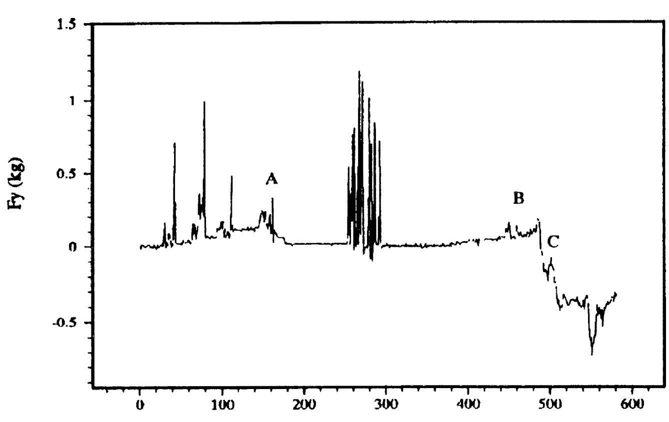

Fig. 13. Force F_y during task execution.

bucket tip elevation and orientation, and Figs. 11–14 display the force/torque information during this task execution, where the feedback at points A, B, and C are marked.

These experimental results indicate that the agent-based control for robotic excavation can emulate the basic abilities and behaviors of human operators for excavators to achieve complex excavation tasks under unknown environments. Note that some of experimental results have been published in [17], similar but new experimental results cannot be published at this time due to our research contract with Caterpillar Corporation, Peoria, IL.

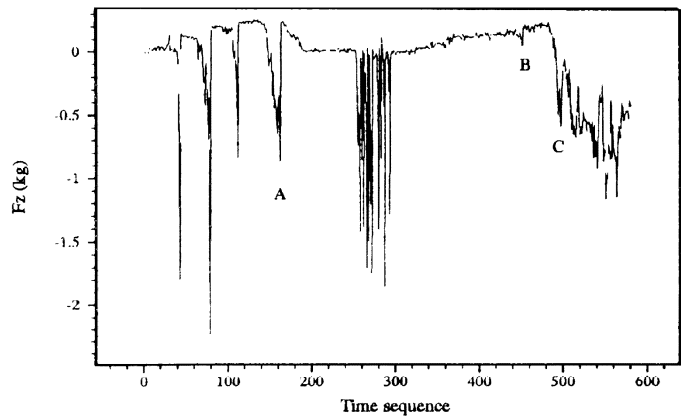

Fig. 14. Force F_z during task execution.

VII. Conclusion

This paper presents an agent-based control for fuzzy behavior programming in robotic excavation and the corresponding experimental results in a simulated excavation environment using a PUMA robotic manipulator. In this approach, the control problem for robotic excavation is solved by decomposing complex tasks into various simple actions that can be implemented by control agents through primitive motions. Petri net transducers and fuzzy logic based decision rules are used to specify control agents and primitive motions for excavation tasks. Extensive laboratory experiments validate this approach and demonstrate its effectiveness.

The control technique developed is an initial step toward autonomous robotic excavation in real-world applications. Related efforts for applying this technique to industrial wheel loaders for open pit mining operations have been conducted and will be reported in our future works.

Acknowledgment

The author would like to thank Drs. X. Shi, D. Chen, and P. Lever for their help in laboratory experiments.

References

[1] T. V. Alekseeva *et al.*, *Machines for Earthmoving Work*. New Delhi, India: Amerind, 1985.
[2] L. E. Bernold, "Motion and path control for robotic excavation," *ASCE J. Aerospace Eng.*, vol. 6, no. 1, pp. 1–18, 1993.
[3] H. Cannon and S. Singh *et al.*, "Models for automated earthmoving," in *Experimental Robotics VI, Lecture Notes in Control and Information Science*, P. Corke *et al.*, Eds. New York: Springer-Verlag, 2000, pp. 183–192.
[4] P. Corke *et al.*, "3D perception for mining robotics," in *Proc. Int. Conf. Field Service Robotics*, Pittsburgh, PA, 1999, pp. 41–47.
[5] ——, "Experiments and experiences in developing a mining robotic system," in *Experimental Robotics VI, Lecture Notes in Control and Information Science*, P. Corke *et al.*, Eds. New York: Springer-Verlag, 2000.
[6] C. S. Desai *et al.*, "Mechanics of 3D soil structure interaction," *ASCE J. Eng. Mech. Division*, vol. 108, no. 5, pp. 731–747, 1982.
[7] Q. P. Ha *et al.*, "Force and position control of electrohydraulic position systems of a robotic excavator," in *Proc. IAARC/IFAC/IEEE Int. Symp. Automation and Robotics in Construction*, Marid, Spain, 1999, pp. 483–489.
[8] ——, "Robust impedance control of excavator dynamics," in *Proc. Int. Conf. Field Service Robotics*, Pittsburgh, PA, 1999, pp. 226–231.
[9] P. J. A. Lever, F.-Y. Wang, and D. Q. Chen, "A fuzzy control system for an automated mining excavator," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 4, San Diego, CA, 1994, pp. 3284–3289.

[10] P. J. A. Lever and F.-Y. Wang, "Intelligent excavator control system for a lunar mining system," *ASCE J. Aerospace Eng.*, vol. 8, no. 1, pp. 16–24, 1995.

[11] P. Lever, F.-Y. Wang, X. Shi, and D. Chen, "Autonomous robotic mining excavation using fuzzy logic and neural networks," *J. Intell. Fuzzy Syst.*, vol. 3, no. 1, pp. 31–42, 1995.

[12] O. Luengo, S. Singh, and H. Cannon, "Modeling and identification of soil-tool interaction in automated excavation," presented at the Proc. IEEE/RSJ Int. Conf. Intelligent Robotic Systems, Victoria, BC, Canada, 1998.

[13] J. Roberts *et al.*, "Autonomous control of underground mining vehicles using reactive navigation," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 2000, pp. 3790–3795.

[14] ——, "3D imagining for a very large excavator," in *Proc. Int. Conf. Field ervice Robotics*, Helsinki, Finland, 2001, pp. 255–261.

[15] H. Romero-Lois *et al.*, "A strategic planner for robotic excavation," in *Proc. 6th Symp. Automation Robotics in Construction*, San Francisco, CA, 1989, pp. 553–560.

[16] X. Shi, P. J. A. Lever, and F.-Y. Wang, "Fuzzy behavior integration and action fusion for robotic excavation," *IEEE Trans. Ind. Electron.*, vol. 43, pp. 395–402, June 1996.

[17] ——, *Autonomous Rock Excavation: Intelligent Control Techniques and Experimentation*. River Edge, NJ: World Scientific, 1998.

[18] X. Shi, F.-Y. Wang, and P. J. A. Lever, "Experimental results of robotic excavation using fuzzy behavior control," *Control Eng. Pract.*, vol. 4, no. 2, pp. 145–152, 1996.

[19] S. Singh, "The state of the art in automation of earthmoving," *ASCE J. Aerospace Eng.*, vol. 10, no. 4, pp. 179–188, 1997.

[20] S. Singh and H. Cannon, "Multi-resolution planning for earthmoving," in *Proc. IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, 1998, pp. 2555–2561.

[21] P. K. Vaha and M. J. Skibniewski, "Congitive force control of excavators," *ASCE J. Aerospace Eng.*, vol. 6, no. 2, pp. 159–166, 1993.

[22] F.-Y. Wang and G. N. Saridis, "A coordination theory for intelligent machines," *Automatica*, vol. 26, no. 5, pp. 833–844, 1989.

[23] ——, "Task translation and integration specification in intelligent machines," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 257–271, Mar. 1993.

[24] F.-Y. Wang and P. J. A. Lever, "An intelligent robotic vehicle system for lunar/martian resource assessment," in *Recent Trends in Mobile Robots*, Y. F. Zheng, Ed. River Edge, NJ: World Scientific, 1993, pp. 293–313.

[25] F.-Y. Wang, Z. X. Wang, and P. B. Mirchandani, "The VISTA project and its applications," *IEEE Intell. Syst. Mag.*, vol. 17, pp. 72–75, June 2002.

[26] Y. T. Lin *et al.*, "Implementing adaptive driving systems for intelligent vehicles using neuro-fuzzy networks," *Transport. Res. Rec.*, no. 1774, pp. 98–105, Nov. 2001.

[27] F.-Y. Wang and C.-H. Wang, "Agent-based control systems for operation and management of intelligent network-enabled devices," in *Proc. of IEEE Int. Conf. Systems, Man, and Cybernetics*, vol. 2, Crystal City, VA, 2003, pp. 1123–1129.

[28] F.-Y. Wang, "A hierarchical intelligent control system for robotic excavation in autonomous truck loading," Univ. Arizona, Tucson, AZ, PARCS Working Rep. 01-08-03, 2003.

[29] M. C. Zhou, M. C. , and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Norwell, MA: Kluwer, 1993.

Fei-Yue Wang (S'87–M'89–SM'94–F'03) received the B.S. degree in chemical engineering from Qingdao University of Science and Technology, Qingdao, China, the M.S. degree in mechanics from Zhejiang University, Hangzhou, China, and the Ph.D. degree in electrical, computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, in 1982, 1984, and 1990, respectively.

He joined the University of Arizona, Tucson, in 1990, and became a Full Professor of Systems and Industrial Engineering in 1999. Currently, he is the Director of the Program for Advanced Research in Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences, Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program. Since 2002, he has been the Director of the Key Laboratory of Complex Systems and Intelligence Science at the Chinese Academy of Sciences. His current research interests include modeling, analysis, and control mechanism of complex systems; agent-based control systems; intelligent control systems; real-time embedded systems, application specific operating systems (ASOS); applications in intelligent transportation systems, intelligent vehicles and telematics, web caching and service caching, smart appliances and home systems, and network-based automation systems. He has published more than 200 book, book chapters, and papers in those areas since 1984 and has received substantial funding from various organizations and institutions.

Dr. Wang received Caterpillar Research Invention Award with Dr. P.J.A. Lever in 1996 for his work in robotic excavation and the National Outstanding Young Scientist Research Award from the National Natural Science Foundation of China in 2001, as well as various industrial awards for his applied research from major corporations. He was the Editor-in-Chief of the *International Journal of Intelligent Control and Systems* from 1995 to 2000, the Editor-in-Charge of *Series in Intelligent Control and Intelligent Automation* from 1996 to 2004, and currently is the Editor-in-Charge of *Series in Complex Systems and Intelligence Science*, Editor for the Intelligent Transportation Systems Department, an Associate Editor of IEEE INTELLIGENT SYSTEMS, and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and several other international journals. He is an elected member of IEEE SMC Board of Governors and IEE ITSC AdCom, the Secretary and Vice President of IEEE Intelligent Transportation Systems Council, and Chair the Technical Committee on System Complexity of the Chinese Association of Automation. He was the Program Chair of the 1998 IEEE International Symposium on Intelligent Control, the 2001 IEEE International Conference on Systems, Man, and Cybernetics, Chair for Workshops and Tutorials for 2002 IEEE International Conference on Decision and Control (CDC), the General Chair of the 2003 IEEE Internation Conference on Intelligent Transportation Systems, Co-Program Chair of the 2004 IEEE International Symposium on Intelligent Vehicles, and will be the General Chair of the IEEE Internation Symposium on Intelligent Vehicles and the IEEE International Conference on Networked Control and Sensing Systems in 2005. He was the Vice President and one of the major contributors of the American Zhu Kezhen Education Foundation, and a Member of the Boards of Directors of five companies in information technology and automation. He is also a member of Sigma Xi, ACM, AMSE, ASEE, and the International Council of Systems Engineering (INCOSE).