# Intelligent Sub-coach Construction in a Fuzzy Coach-Player System for Controlling a Robot Manipulator

Chandimal Jayawardena, Keigo Watanabe and Kiyotaka Izumi
Department of Advanced Systems Control Engineering,
Graduate School of Science and Engineering, Saga University, Japan
chandimal@ieee.org, {watanabe, izumi}@me.saga-u.ac.jp

*Abstract*— **This paper presents a method of controlling robot manipulators by fuzzy-voice commands. Recently, there have been some researches on controlling robots using information rich fuzzy voice comands such as "go little slowly" and learning from such commands. However, the scope of all those works was limited to basic fuzzy-voice motion control commands. In this paper, we introduce a method for controlling the posture of a manipulator using complex fuzzy-voice commands. The proposed idea is demonstrated with a PA-10 redundant manipulator.**

## I. Introduction

The inception of the first real-world applications of robots was at the heavy industry. However, the scope of the robotic applications is steadily improving and gradually spreading into new areas. One of the most important such areas is, human-friendly robots. The research activities in this area are strongly motivated by the aging of the population. A personal robot would increase the functional capacity of the aged and possibly even improve their state of health. In addition to that, a successful human-friendly robot will help disabled people, be conversant in several languages and watch over babies [1].

In the area of human-friendly robots, the human-robot interaction is very important. A true interactive robot should be capable of showing human-like behaviour when dealing with a human. Natural language communications, gestures, showing feelings, etc. are the important behaviours [2].

The ability to communicate using voice-commands is a must for a human-friendly robot. In natural language communication, encountering words and phrases with fuzzy implications is inevitable. On the otherhand, words with fuzzy implications are useful in machine control because they can fine tune the performance of the machine. For example, a command like "*move slowly*" will contain many information regarding the nature of the terrain, distance to obstacles, etc.[3].

Lin and Kan[4] proposed an adaptive fuzzy command aquisition method to control machines by natural language commands such as "*move forward at a very high speed*". In [5] robot control using such information rich voice commands has been studied. In a more practical approach adopted in [6], controlling a redundant manipulator by such commands to perform a complex assembling task has been demonstrated.

However, in all those systems, although the interpretation of fuzzy voice commands has been considered well, the importance of learning from such commands has not been addressed. As an attempt to realize the learning from fuzzy voice commands, the concept of a sub-coach was introduced in [7] and was further developed in [8]. A brief description of this concept is as follows.

Consider the process of controlling a robot to achieve a complex task using voice commands. If the robot does not have any prior knowledge of the task and if it cannot learn from the previous commands, the user has to issue a series of commands at each step such as:

- "move forward slowly"
- "turn to right"
- "move far"
- "stop", etc.

This type of a system can be seen as a coach-player system, since a coach (or a human user) commands a robot (or a player) by observing its performance at each step. The robot acts accordingly and the user will stop commanding when the performance of the robot comes to a satisfactory level (the work at hand is completed). Although this kind of coach-player systems is very useful in many situations, their disadvantage is the need of issuing commands at each small step. This can be eliminated by introducing the concept of a sub-coach. Sub-coach is a software process which stands in between the user (coach) and the robot (player). It can learn from the fuzzy voice commands issued by the user and can use that knowledge to control the robot without the help of the coach. However, sub-coach can consult the coach in situations where it does not have sufficient knowledge to handle a particular situation.

Systems discussed in [7] and [8] used the fuzzy-voice commands shown in Table I to control the tip position of a manipulator. User could guide the tip from a starting point to a target point within the working space. A sub-coach process was designed in a way such that it could learn from user's commands. After few training movements, the sub-coach could directly control the manipulator consulting the user only when he faced difficulties.

However, the commands shown in Table I are basic fuzzy motion commands. Therefore, to complete any task, either the user or the sub-coach had to guide the manipulator through several steps. For example, to change the posture of the manipulator from its home posture to a desired posture, several basic commands had to be issued.

The system discussed in [6] also had the same drawback because it too used a set of basic fuzzy motion commands somewhat similar to the commands shown in Table I. In

TABLE I

BASIC FUZZY-VOICE MOTION COMMANDS.

| Action | Action Modification |
|---|---|
| go up | |
| go down | very little |
| go right | little |
| go left | medium |
| go forward | far |
| go backward | |

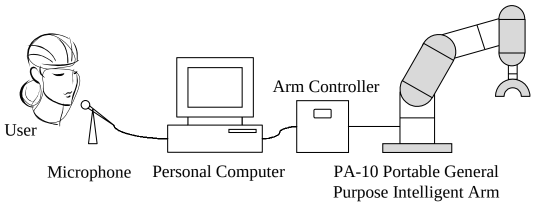



Fig. 1. The experimental setup.

addition, since learning had not been considered, a user had to command the robot over and over with similar commands in similar situations.

In the system presented in this paper, in addition to the basic fuzzy motion commands given in Table I, complex fuzzy-voice posture controlling commands are also used. A sub-coach is used to learn such commands from a human user. This will improve the efficiency of a coach-player system for controlling a robot manipulator by eliminating the drawback discussed above.

## II. SYSTEM OVERVIEW

In the framework of conventional robot control, it needs to control the joint angles and the joint velocities of the robot using the joint input torques in order to achieve the final control objective. However, in a fuzzy coach-player system, the control input for a manipulator will compose of voice commands. Naturally, they will be very vague or fuzzy. For example, if controlling the posture of a manipulator is considered, a human user might say, "*turn to right*", "*bend your elbow little*", "*bend to left a little and turn your tip downwards*", etc.

In the implemented system, the tip of a manipulator could be moved to any desired position within its 3-D working space using fuzzy-voice commands. The most efficient way to do this is,

1) to command the manipulator to change to a more convenient posture to reach the desired point and then,
2) to command the manipulator to reach the exact target from that convenient posture.

For the step (1) above, one complex posture controlling command can be used. Step (2), can be realized with a series of basic motion commands.

The proposed concept was implemented using a PA-10 redundant manipulator. The experimental setup is shown in Fig. 1.

It consists of a microphone, a personal computer, a PA-10 portable general purpose intelligent arm and the arm controller. The speech recognition software, the software modules for the implemented system and the operational control program for PA-10 are hosted in the personal computer whose operating system is Windows XP. The speech recognition was performed using IBM Via Voice.

### *A. Learning to Execute Complex Commands*

Fuzzy-voice posture controlling commands issued to a manipulator can be categorized into two.

1) Fuzzy-voice joint control commands: These are the commands which affect a single joint.
2) Complex fuzzy-voice posture control commands: These affect more than one joint resulting in complex manoeuvring of the manipulator.

We assumed that any complex fuzzy-voice posture control command can be broken down to a set of fuzzy-voice joint control commands. Therefore, if the ability to interpret and execute the former is built-in, the later can be learned using the algorithm shown in Fig. 2.

Functional modules referred to in the algorithm are shown in Fig. 3. Definitions of the entities used in the algorithm are given below.

| | |
|---|---|
| $P$ : | User visualized final state of the manipulator; |
| $S$ : | User's perception of the current state of the manipulator; |
| $usrcmd$ : | User's voice command; |
| $C$ : | Set of available fuzzy-voice joint control commands, where $C = \{c_1, ..., c_i, ..., c_n\}$; |
| $u_{c_i}^{(k)}$ : | Crisp joint control input corresponding to $c_i$ at time $k$; |
| $W$ : | Set of complex commands learned by the sub-coach, where $W = \{w_1, ..., w_q, ..., w_r\}$; |
| $C^{(w_q)}$ : | The series of fuzzy-voice joint control commands equivalent to $w_q$, where $C^{(w_q)} = \{c_1^{(w_q)}, ..., c_p^{(w_q)}, ..., c_{N_{w_q}}^{(w_q)}\}$ and $\forall c_p^{(w_q)} \in C$. |

In $u_{c_i}^{(k)}$, time label $k$ is used merely to indicate that the crisp joint control input corresponding to same $c_i$ is different at different times.

This algorithm works as follows. The user can issue any arbitrary command. If it is a joint control command, FI interprets the crisp value of the joint control input and the command is executed. If it is a complex command, and if it is in the knowledge-base of the SC (i.e. previously learned), the sequence of joint control commands which is equevalent to the complex command is sent to the FI by the SC. Once that sequence of commands is executed, the action referred by the complex command is realized. If the user issued a complex command and if it is not in its knowledge-base, SC will consult the user. Then, the user has to use a series of basic commands instead of the complex command to obtain the same work done. Meanwhile, SC learns the complex command and it will be added to its knowledge-base.

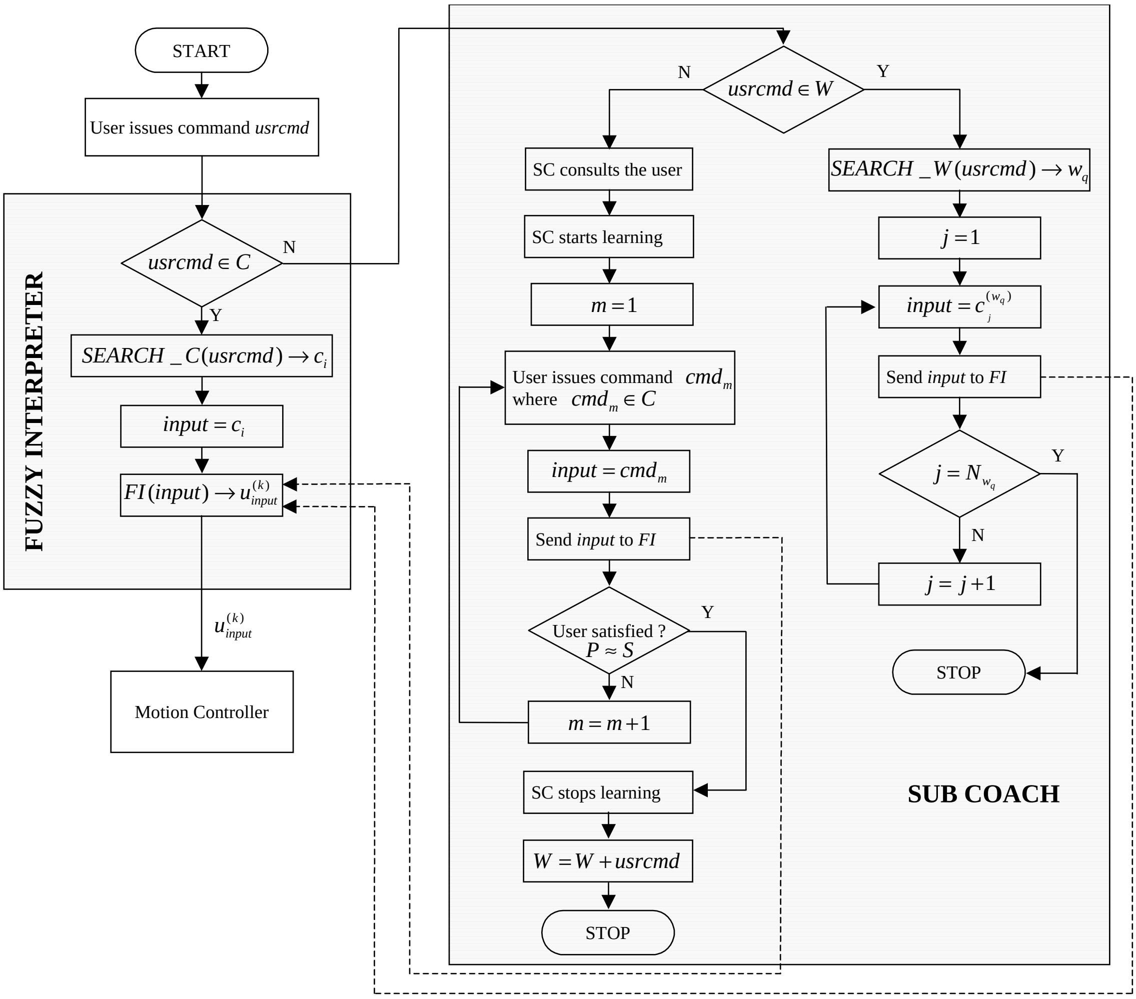



Fig. 2. Algorithm for learning and executing complex commands.

## III. IMPLEMENTATION

In Fig. 3, both SC and FI are software modules. MC consists of the operation control program of PA-10 manipulator and the arm controller.

As shown in Fig. 2, FI module consists of the knowledge-base and the inference procedure for interpreting fuzzy-voice joint control commands. SC module consists of the knowledge-base and the learning procedure for complex posture controlling commands.

In this implementation, the posture controlling was performed directly using joint angles. In the case of PA-10, it is possible to control any joint angle using high-level commands if the required angular deviation for that joint is known.

### *A. Fuzzy-voice joint control commands*

These are the elements of set $C$ defined in the section II-A. When controlling a robot manipulator with voice commands, there is no way to control joint angles using conventional methods. Instead, it would be much more convenient for a user to issue a command such as "*rotate your wrist little right*".

In the implemented system, only three axes of the PA-10 manipulator are considered for the simplicity. However, the same concept can be extended to all seven axes.

TABLE II

FUZZY-VOICE JOINT CONTROL COMMANDS.

| Action | Action Modification |
|---|---|
| bend your wrist forward | |
| bend your wrist backward | very little |
| bend your lower arm forward | little |
| bend your lower arm backward | medium |
| rotate your lower arm right | far |
| rotate your lower arm left | |

The voice commands that a human user can use to control these three joints are shown in the Table II. User can use any action command modified by any action modification command. The exhaustive set of such combinations makes the set $C$.

The joint configuration, axis nomenclature, and axis motion of PA-10 manipulator are shown in Fig. 4. 'Wrist forward' and 'wrist backward' motions referred to by the voice commands in the Table II are equavelent to the positive and negative rotations about the axis no. 6. Similarly, 'lower arm forward' and 'lower arm backward' are equavalent to the rotations about the axis no. 4. 'lower arm left' and 'lower arm right' are equavelent to the rotations about the axis no. 3.

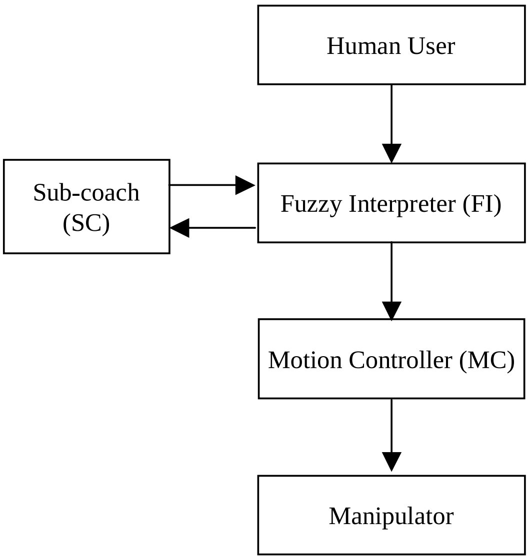



Fig. 3. Functional modules of the architecture.

For the interpretation of these fuzzy commands, a method similar to the fuzzy-motion command interpretation method used in [6] is used. The similarity between the two approaches comes from the fact that in both systems, the actual response to the previous command is used as an input when interpreting the present command. However, in the system discussed in this paper, simple fuzzy reasoning is used while in [6], a fuzzy neural network was used.

The membership functions used are shown in Fig. 5. Once a command is issued, the angle to be rotated ($\alpha$) is determined using the angle rotated in response to the immediately previous command ($\theta$) as an input. Altogether there are sixteen (16) rules. They are shown below.

$R^1$ : If action modification is 'very little' and $\theta$ is $L$ then $\alpha$ is $VVL$.
$R^2$ : If action modification is 'very little' and $\theta$ is $M$ then $\alpha$ is $VL$.
$R^3$ : If action modification is 'very little' and $\theta$ is $H$ then $\alpha$ is $L$.
$R^4$ : If action modification is 'little' and $\theta$ is $L$ then $\alpha$ is $VL$.
$R^5$ : If action modification is 'little' and $\theta$ is $M$ then $\alpha$ is $L$.
$R^6$ : If action modification is 'little' and $\theta$ is $H$ then $\alpha$ is $M$.
$R^7$ : If action modification is 'medium' and $\theta$ is $L$ then $\alpha$ is $L$.
$R^8$ : If action modification is 'medium' and $\theta$ is $M$ then $\alpha$ is $M$.
$R^9$ : If action modification is 'medium' and $\theta$ is $H$ then $\alpha$ is $H$.
$R^{10}$ :If action modification is 'far' and $\theta$ is $L$ then $\alpha$ is $H$.
$R^{11}$ :If action modification is 'far' and $\theta$ is $M$ then $\alpha$ is $VH$.
$R^{12}$ :If action modification is 'far' and $\theta$ is $H$ then

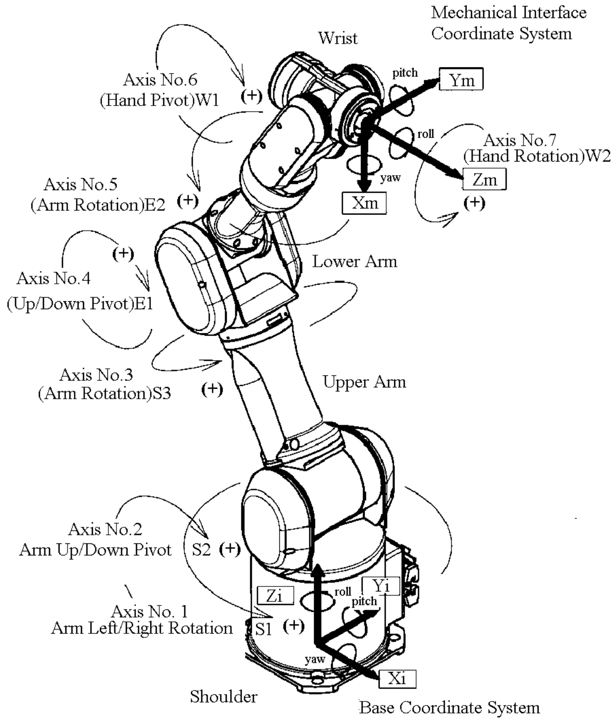



Fig. 4. Arm nomenclature of PA-10 manipulator.

TABLE III
COMPLEX FUZZY COMMANDS LEARNED BY THE SUB-COACH.

| Complex command | Fuzzy-voice joint control command sequence |
|---|---|
| bend towards left a little | (i) rotate the lower arm medium left<br>(ii) bend the lower arm medium forward<br>(iii) bend the lower arm little forward<br>(iv) bend the wrist medium forward |
| bend forward and incline to right | (i) bend the lower arm little forward<br>(ii) rotate the lower arm medium right<br>(iii) bend the lower arm medium forward |

$\alpha$ is $VVH$.

In the algorithm discussed in the section II-A, $u_{input}^{(k)}$ consists of both angular deviation $\alpha$ and the axis around which to be rotated. As a result, the motion controller will command the manipulator to rotate around the relevant axis by an amount of $\alpha$.

### *B. Complex fuzzy-voice posture control commands*

As explained in the section II-A, these commands affect more than one joint of the manipulator. According to our assumption, one such complex command can be represented by a series of basic commands discussed in the above section.

As the basic commands, these complex commands cannot be directly interpreted by FI; instead, they shoud be learned by SC. Initially, there are no complex commands in the

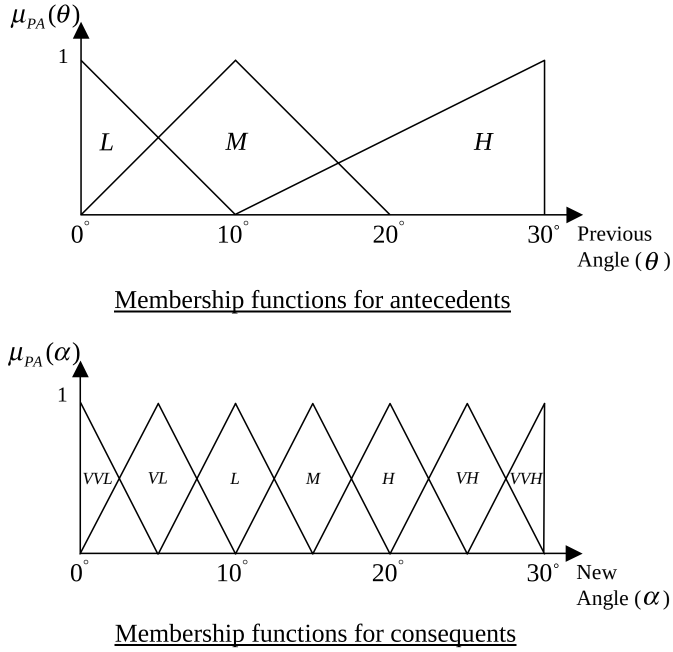



Fig. 5. Membership functions.

knowledge-base of SC. However, as and when user issues commands, SC learns them and adds to its knowledge base.

As an example, Table III shows two complex commands learned by SC. Second column shows the sequence of basic commands equivalent to the complex command.

## IV. Results and Conclusion

Fig. 6 shows four test results obtained with the implemented system.

In each graph in the figure, A was the home position of the manipulator tip. User desired to move the tip to the point C. Thus, first user issued a complex posture controlling command. At the command, the tip travelled from A to B. Then, user did the fine adjustments to the tip position to reach the exact point C. For these fine adjustments, basic motion commands given in the Table I were used.

In each graph, complex commands used to move the tip from A to B are as follows:

Test 1 : "bend far down left"
Test 2 : "bend towards far right and turn your tip down"
Test 3 : "reach far upper back"
Test 4 : "bend little back inclined to right"

It was observed that the implemented system could be used for complex maneouvering of the manipulator. Any user desired point within the working space could be accessed. The number of complex commands that a user can use is not fixed while only the number of basic commands is fixed. As and when user issues new commands, they are added to the knowledge base of the sub-coach and they can be used later.

It can be observed that the user can move the manipulator tip to the target point with fewer number of commands when compared with the systems discussed in [6], [7], and [8] in which only the basic commands were used.

The dexterity of redundant manipulators can be further improved by this method. It is well suited for complex human-robot cooperative tasks. Capabilities of redundant manipulators can be used together with the capabilities of a human such as fast processing of visual information and intelligent decision making.

## References


[1] P. Menzel and F. D'Aluisio, *Robosapiens*, The MIT Press, England, 2000.

[2] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, pp. 143–166, 2003.

[3] K. Pulasinghe, K. Watanabe, K. Kiguchi, and K. Izumi , "Voice controlled modular fuzzy neural controller with enhanced user autonomy," *Artificial Life and Robotics*, vol. 7, pp. 40–47, 2003.

[4] C.T. Lin and M.C. Kan, "Adaptive fuzzy command aquisition with reinforcement learning," *IEEE Trans. on Fuzzy Systems*, vol. 6, no. 1, pp. 102–121, Feb. 1998.

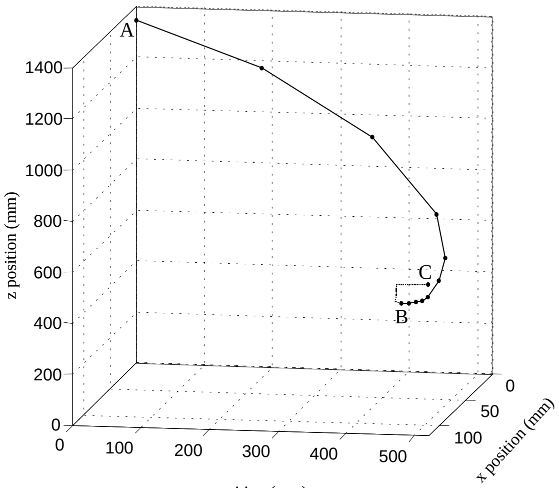
[5] K. Pulasinghe, K. Watanabe, K. Izumi, and K. Kiguchi, "A modular fuzzy-neuro controller driven by spoken language commands," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 34, no. 1, pp. 293–302, 2004.

[6] K. Pulasinghe, K. Watanabe, K. Izumi, and K. Kiguchi, "Control of redundant manipulators by fuzzy linguistic commands," in *Proc. of the SICE Annual Conference 2003*, Fukui, Japan, Aug. 2003, pp. 2819–2824.

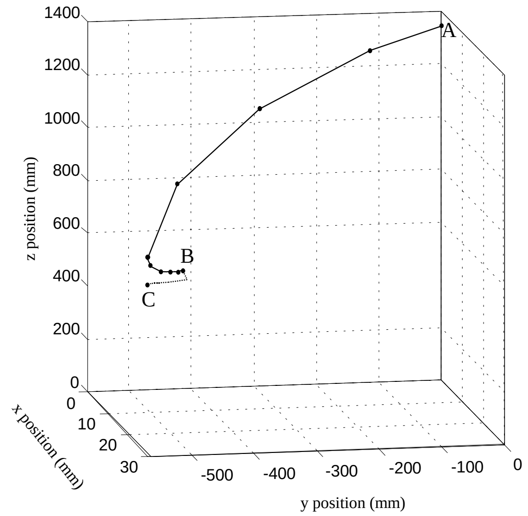
[7] C. Jayawardena, K. Watanabe, and K. Izumi,"Knowledge acquisition by a sub-coach in a coach-player system for controlling a robot," *The 4th International Conference on Advanced Mechatronics*, Hokkaido, Japan, Oct. 2004, To be published.

[8] C. Jayawardena, K. Watanabe, and K. Izumi, "Probabilistic neural network based learning from fuzzy voice commands for controlling a robot," *International Conference on Control, Automation, and Systems*, Bangkok, Thailand, Aug. 2004, Accepted for publishing.

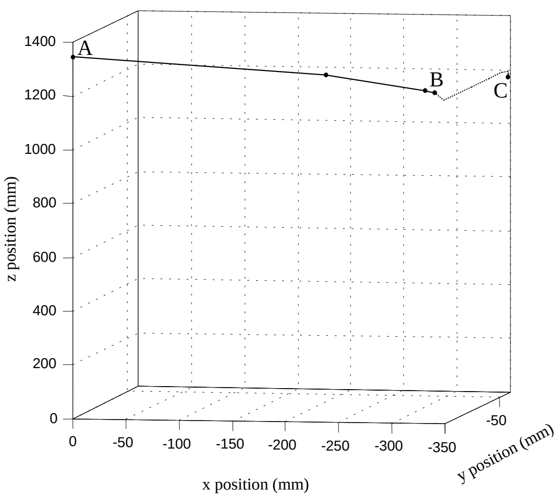
[9] K. Pulasinghe, K. Watanabe, K. Kiguchi and K. Izumi, "A noval modular neuro-fuzzy controller driven by natural language commands," in *Proc. of the 40th SICE Annual Conference*, 312C-4 (CD ROM), Nagoya, Japan, Jul. 2001.



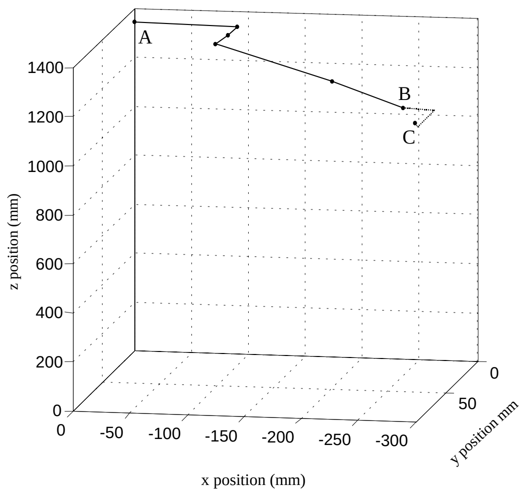

(a) Test 1




(b) Test 2




(a) Test 3




(b) Test 4

Fig. 6. Execution of complex commands.