8-8-2006

Fuzzy Mouse Cursor Control System for Computer Users with Spinal Cord Injuries

Tihomir Surdilovic

Fuzzy Mouse Cursor Control System
For Computer Users with Spinal Cord Injuries

A Thesis

Presented in Partial Fulfillment of Requirements for the

Degree of Master of Science

in the College of Arts and Sciences

Georgia State University

2005

by

Tihomir Surdilovic

Committee:

Dr. Yan-Qing Zhang, Chair

Dr. Rajshekhar Sunderraman, Member

Dr. Michael Weeks, Member

Dr. Yi Pan, Department Chair

Date

July 21st 2005

Abstract

People with severe motor-impairments due to Spinal Cord Injury (SCI) or Spinal Cord Dysfunction (SCD), often experience difficulty with accurate and efficient control of pointing devices (Keates et al., 02). Usually this leads to their limited integration to society as well as limited unassisted control over the environment. The questions "How can someone with severe motor-impairments perform mouse pointer control as accurately and efficiently as an able-bodied person?" and "How can these interactions be advanced through use of Computational Intelligence (CI)?" are the driving forces behind the research described in this paper. Through this research, a novel fuzzy mouse cursor control system (FMCCS) is developed. The goal of this system is to simplify and improve efficiency of cursor control and its interactions on the computer screen by applying fuzzy logic in its decision-making to make disabled Internet users use the networked computer conveniently and easily. The FMCCS core consists of several fuzzy control functions, which define different user interactions with the system. The development of novel cursor control system is based on utilization of motor functions that are still available to most complete paraplegics, having capability of limited vision and breathing control. One of the biggest obstacles of developing human computer interfaces for disabled people focusing primarily on eyesight and breath control is user's limited strength, stamina, and reaction time. Within the FMCCS developed in this research, these limitations are minimized through the use of a novel pneumatic input device and intelligent control algorithms for soft data analysis, fuzzy logic and user feedback assistance during operation. The new system is developed using a reliable and cheap sensory system and available computing techniques. Initial experiments with healthy and SCI subjects have clearly demonstrated benefits and promising performance of the new system: the FMCCS is accessible for people with severe SCI; it is adaptable to user specific capabilities and wishes; it is easy to learn and operate; point-to-point movement is responsive, precise and fast. The integrated sophisticated interaction features, good movement control without strain and clinical risks, as well the fact that quadriplegics, whose breathing is assisted by a respirator machine, still possess enough control to use the new system with ease, provide a promising framework for future FMCCS applications. The most motivating leverage for further FMCCS development is however, the positive feedback from persons who tested the first system prototype.

Acknowledgments

I would like to thank all people that guided me over the years. I thank my wife and my parents for their support. I would especially like to thank my advisor, Dr. Yan-Qing Zhang and committee members Dr. Rajshekhar Sunderraman and Dr. Michael Weeks, for useful discussions and recommendations that were very helpful for completion of this thesis. I also thank Mr. Milan Rasic, from the Electrotechnical Faculty of University in Nis, Serbia and Montenegro, for his help in implementing the flowmeter board, a part of the input device used within the fuzzy mouse cursor control system.

Finally, my deepest thanks go to Mr. Kalmani, my colleague and friend, for his help and patience in testing the new prototype application. His strong will to fight against misfortune after his injury, motivated me greatly to start and accomplish this research.

Table of Contents

List of Figures

List of Tables

1. Introduction

According to the National Spinal Cord Injury Association (NSCIA) approximately 300,000 people live with SCI in the USA, with an annual incidence of 12000 new cases each year (NSCI Statistical Center, 2004). Most of these injuries result from motor vehicle accidents, violence, and sport injuries, affecting primarily young adults (Push America source, http://www.pushamerica.org/Physical.asp). SCI causes loss of function, such as mobility and feeling, loss of independence, low quality of life and high cost of care. SCI is usually classified into two types: complete or incomplete (Louis C. Memorial Library, source http://calder.med.miami.edu/pointis/causes.html). A complete injury results in no function below the level of the injury; no sensation and no voluntary movement, whereas an incomplete injury allows partial motion and/or feeling below the level of the injury. According to the American Spinal Cord Association, a cure for SCI does not exist, and only a very small fraction of individuals sustaining SCI recover all functioning lost. For people with SCI the use of computers play a significant role in their post-injury life, whether used for work, pleasure or environmental control, such as bed positioning, television, telephone, lights etc. For these people computers can provide a whole new realm of independence. However, computer access for disabled users commonly requires special hardware, software and/or assistive technology (AT) devices[1] to help with basic computer tasks such as typing or using a mouse. Although, multitudes of AT devices for computer access are available on the market today, they are limited in their ability to optimally convey user’s intents and needs. Nevertheless, there is not one assistive input device that will meet the needs of all motor-impaired people, since everyone’s needs and abilities are different.

1.1 Problem statement

Research presented in this thesis focuses on improving the human-computer interaction (HCI) for people suffering severe SCI, where the areas of lesion encompasses the most

[1] According to the definition (Public Law 105-17: Individuals with Disabilities Education Act, IDEA Sec. 602), AT denotes any item, piece of equipment, or product system that is used to increase, maintain or improve the functional capabilities of individuals with disabilities.

critical cervical or neck region consisting of seven vertebrae numbered C1-C7 . If the spine is injured below the level of the neck a person is said to be a *paraplegic* and will be paralyzed to some degree in the legs and abdomen, otherwise if the injury is in the cervical region, a person is said to be quadriplegic and may experience difficulties breathing (according to Apparelyzed.com source, www.apparelyzed.com/quadriplegia.html).

Complete quadriplegics require considerable levels of assistance to operate a computer. Since only their head and neck muscles can still move voluntarily, proper HCI AT-devices utilize various motor functions within these areas, such as head movements, eye-gaze, lips and/or tongue touch control, respiration effects (puff/sip), voice, chin motion etc. All of these “sources” exhibit specific advantages and drawbacks. Some of them are subject to sanitation problems (puff/sip switches), while other may physically intrude upon the user (chin-operated joysticks), or maybe socially objectionable (head-mounted accelerometers, inclinometers, or infrared dots). Some AT-devices can unduly restrict user’s gaze (eye-gaze computer interface), while voice-control is often inadequately responsive for continuous control.

1.2 Approach

This thesis focuses on development of an improved HCI system for cursor, i.e. screen pointer, control (Mouse Cursor Control – MCC). MCC represents the predominant computer interface, and is an integral part of interaction with GUI tools.

A major step in designing AT devices and systems is to determine its “target audience”. In this research, the hypothetical users of the Fuzzy Mouse Cursor Control System (FMCCS) are severely motor-impaired people (e.g. quadriplegics), with the capacity of producing input signals to the system by use of their breath. Other abilities to produce inputs are not considered and are assumed to fall under the user’s expression deficiencies. The most commonly used assistive input device for such users is Sip’N Puff, a dual-action pneumatic switch device capable of sensing airflow direction through an easily accessible piece of tubing build similarly to a drinking straw. These types of devices cannot sense airflow intensity. Therefore, the input device to the FMCCS described in this work is assumed to be a dual-action pneumatic switch such as Sip'N Puff, with the addition of an airflow sensor, which is able to measure the intensity of the air flowing through the input device, allowing for two distinct signals to the FMCCS, one signal describing the direction of airflow, and the other its intensity.

The second major step in designing assistive technology devices and systems is to determine how the user will interact with the system (describe the human-computer interactions). Human-computer interactions in the described FMCCS are based on two selection operations, namely Pointing, or the positioning of the cursor at the desired location on the computer screen, and Clicking, the executions of mouse up/down functions which are interpreted by the operating system as an indication to complete the selection process associated to the previously pointed area icon (Hwang, 2003).

In order to understand pointing, or cursor movement, best suited for motor-impaired users, we must understand the difference of their interaction behaviors compared with able-bodied users (Keates et al, 02). Cursor measure studies for motor-impaired users (Hwang, 2002) have revealed several difficulties experienced by motor-impaired users in performing pointing tasks. FMCCS described in this paper tries to minimize these difficulties by limiting the need and amount of user input as much as possible in order to achieve the most accurate positioning of the cursor on the target object on the screen. Fuzzy logic is used to aid the making of these decisions, assisting the user in efficient and brief cursor positioning. It is assumed that through the assistance of fuzzy logic, and minimized user input, we could significantly decrease the performance time of the pointing task performed by motion-impaired users. Mouse button execution studies (Price and Cordova, 83) have revealed that multiple buttons are more preferable for different clicking tasks compared with a single mouse button approach. Multiple button execution can be achieved in a system using a pneumatic dual-action switch as input device, by interpreting the input as Morse code. Different Morse code sequences could trigger different clicking functions. Within the FMCCS described in this work, we focus on the three predominate functions: single click (both mouse buttons), double click, and drag. The implementation of more advanced clicking functions is part of future developments goals.

1.3 Outline of the thesis

This thesis is organized as follows. Related works are discussed in Chapter 2. Chapter 3 presents the new fuzzy cursor control system development involving fuzzy input scanning and mouse pointer control algorithms. Several of these algorithms are tested and evaluated using MATLAB/SIMULINK as well as a GUI setup. The developed hardware system is described in Chapter 4. Implemented software system is presented in Chapter 5. Initial experimental results with people with different levels of SCI are presented in Chapter 6. Finally, conclusions and future development topics are given in Chapter 7.

2 AT techniques for cursor control

A multitude of AT devices for computer access targeting mouse cursor control for handicapped individuals are available on the market or are under development in several research centers. For example, for MCC only, there are worldwide more than 160 commercial products available (according to ABLEDATA source, www.abledata.com). Developing novel or improving existing systems for cursor control requires a deeper critical knowledge of present solutions. Therefore, this chapter provides a review of current MCC solutions and considers representative devices within specific groups, discussing their accessibility features and practical suitability for user suffering severe SCI. Both state-of-art technology, as well as recent research developments will be discussed.

2.1 Classification of AT devices

AT devices can be classified based on characteristics that are relevant for their operations. For users with SCI the most relevant classification criterion are *accessibility features* encompassing a variety of motor functions and gesture controls required to operate a system. Accessibility features define required skills and efforts, as well as physical effects used for sensing of user intents and commands. From the communication theory viewpoint, these features define communication, i.e. interaction channels between the user and the system. Principal communication channels can be distinguished into puff/sip switches, tongue switches, optical switches, head-controlled mouse, chin-controlled assistive devices, eye-tracking systems, facial-muscle sensors, voice-controlled devices, lips-controlled joysticks, gesture-based mice, as well as neural interfaces (brain-computer interfaces based on EEG signals).

2.2 Switch related systems

Switch-related AT systems have been applied for controlling computer and environmental devices for a considerable amount of time now, and they and they are still popular and subject of several current investigations (Marler, 04). Due to their relatively low costs and simple use (activated by inhaling or exhaling), puff/sip switches are frequently applied in AT systems for environmental control. Their working principle is quite simple: blowing and sucking on the mouthpiece activates a pneumatic dual switch. It requires little or no movement and offers an easy and unobtrusive way to operate an AT device. For computer access, puff/sip technology is usually combined with Morse code systems, such as “Morse 2000” developed at the University of Wisconsin (see http://www.uwec.edu/ce/Morse2000.htm) to allow different mouse command operations. Switches can be quite useful when there is a need to replace mouse buttons; however pointer control, when using Morse code, is quite inefficient and not responsive.

Tongue touch keypads are dental retainers with several buttons embedded in it. For example, the UCS 1000 produced by newAbilities Systems Inc, includes 9 buttons (see http://www.newabilities.com/ for more details). A wireless radio transmitter provides communication with the control unit. Users of these types of systems use their tongue to move through a series of menus that assist in controlling various devices. These kinds of operations appear to be quite useful for computer use and environmental control including powered mobility access (e.g. wheel chair control). However for continuous and smooth dynamic cursor control they appear to be very tedious. In addition, the price of tongue touch keypad system is quite high, e.g. the USC 1000 is priced from $10,000 to $18,000.

The same conclusion may be drawn for other low-tech switch based interfaces, such as chin or lips operated joysticks, and optical switches. Augmentative joysticks (e.g. “powered easel”, see Madison et al. 1996) are sophisticated mechanical interfaces mounted on an adjustable arm requiring fine short displacements (less than one inch) or lip pressure (e.g. USB Integra Mouse needs ca. 1/3 ounce, about 10 g) in order to trip switches. Optical switches (e.g. Self-calibrating Auditory Tone Infrared Switch SCATIR developed at Michigan State University, see also http://www.tashinc.com/) work with infrared beam light that can be broken by a variety of control gestures, including eye-blink, eyebrow movement, finger movement, head movement, and facial muscle movement. Because they are optical devices, switches can be activated at a distance. The user's controlling body part therefore does not need be in physical contact with the switch sensor, which is an advantage in comparison to tongue touch keypads and augmentative mechanical joysticks. The cost of optical switches is also considerably lower (SCATIR costs about $1,000). There are also several general-purpose Software (SW) and Hardware (HW) interfaces supporting mouse emulation in MS Windows environments for arbitrary physical switch structures (a review of commonly used switch concepts in augmentative computer communication is given in http://okabletech.okstate.edu/at/switches.html). However, the main problem with switch-based systems remains their responsiveness to user inputs. Moreover, the critical problem causing operation disturbances is related to user's involuntary body movements. In spite of recent AT producers' claims that their sophisticated SW improvements can distinguish voluntary and involuntary movements, this remains the decisive problem for people with profound disabilities.

2.3 Head pointing systems

Wireless infrared technology was also applied in various head pointing systems to convert head motions to mouse pointer movements. Commonly these systems use an infrared emitter that is attached to user's glasses, headband or cap. Some systems place transmitter over the user's monitor and point an infrared reflector (reflective dot) at the user's forehead or glasses (e.g. HeadMouse, Tracker 2000). The mouse pointer movement on the screen is then proportional to the user's head movement, which are used to trigger a switch through which the user can control various system functions. A few head pointing system products offer integrated augmentative solutions involving communications tools that feature both a static keyboard and a dynamic color display (e.g. PRC's Pathfinder, see http://www.prentrom.com/), as well as voice recognition abilities. A specific problem with head mouse systems is the required motor skills. A possible problem of patients with sever SCI that use these type of systems is fatigue and muscle pain. As demonstrated in (LoPresti *et al.*, 03) patients with severe multiple sclerosis and SCI have reduced range of neck motion causing difficulties during computer use through these type of devices. Subjects with disabilities were also found to have longer reaction time, and spend more time trying to make fine adjustments to cursor position. A number of head-control systems available in the market today include filtering and gain adjustment options that might improve usability for some people with neck movement impairments. However, practical experiments have demonstrated limitations of these systems and have presented needs for more adaptive techniques to allow head control to automatically be adjusted to the needs and abilities of a particular user. More severe problems with head control were mentioned in (Ortega, 2004). Head mouse systems, operating on the principle of a single switch, allow the user to give single commands at the appropriate time; which should reduce the amount of user's head movements. However, a critical issue with this approach is its exact timing requirement, which often leads to increased head movement and spasticity; especially when the user is trying to work relative fast. Head movements indeed require considerable muscles and ligaments efforts and their overuse can cause injuries to the users.

2.4 Vision-based tracking

A further improvement of switch related augmentative devices is achieved by using video cameras, image processing and visual tracking algorithms. Vision-based advanced mouse emulators, e.g. Camera Mouse developed at Boston College (Betke *et al.* 02, http://www.bc.edu/schools/csom/eagleeyes/cameramouse/), track users' movements with a camera focusing on various body features as target, such as tip of the user's nose, eyes, lips or fingers. Sophisticated pattern recognition SW algorithms recognize the target pattern, determine motion parameters, and translate this information into motion of the mouse pointer on the screen. Initial experiments with the Camera Mouse (Betke *et al.* 2002) have given encouraging results for subjects with relatively good muscle control abilities. It has proven to be user friendly because it requires no calibration or body attachments before and during its use. The Camera mouse presents an AT approach, that is easily adaptable to serve specific needs of various disabilities, and it is especially suitable for children (e.g. with cerebral palsy). However, several problems were also observed during its experiments, such as drifts, loss of communication, slow communication rates etc. For people with insufficient muscles control, the Camera Mouse becomes quite ineffective. Despite of these limitations, advanced image processing provides an efficient basis for development of sophisticated adaptable dynamic video-based mouse emulators and general augmentative computer interfaces with reliable communication rates. However, the overall cost of such systems is very high. Ongoing research focuses on intelligent algorithms capable of learning the user's performance and habits to adjust lower level image-processing based mouse control reaction. A concept of future cognitive vision-based mouse emulator referred to as IMouse was recently proposed at Yang's Scientific Research Institute (http://www.yangsky.com/IMouse.htm).

2.5 Eye tracking systems

Another group of commercial AT systems for computer access is based on the estimation of eye gaze direction. Communication through the direction of eye is one of fastest modes of human interaction with a computer system, making this approach very attractive for users with severe disabilities. Eye tracking systems do not require head movements and can be applied for people with insufficient muscle control. Through a directing gaze towards various fixation points on screen the user can express his/her own intention, e.g. to position the mouse pointer to a selected target. There are several physical principles, which have been utilized for eye-gaze measuring. The most common approach utilizes effects of light reflection (e.g. from an infrared source) on the surface of the cornea (e.g. EyeGaze Computer System from LC Technologies-http://www.eyegaze.com). This principle, patented by Mason (69), is based on the observation that reflected light produces a bright spot (glint) on the cornea, which position can vary according to the change of eye-gaze direction. These changes can be estimated using a vector from the glint to the center of the pupil assuming analytic relationships between glint vector and the gaze (Cleveland, 93). Parameterization of these polynomial functions requires quite cumbersome calibration procedures, requiring the user to fixate his/her gaze on numerous target points on screen one by one. Although head movement's range in people with cervical SCI is minute compared to the subjects without disabilities (according to LoPresti et al. 03, neck range of motion is halved), unpredicted head displacements can jeopardize the performance of eye-gaze systems and must be considered. Users with severe disabilities are often not able to keep their head still, causing problems with their interactions with the system. In some systems, uncomfortable chin rests are applied to keep the user's head still, making the interactions with the system less user-friendly, and cumbersome.

Considerable research efforts were recently directed towards overcoming these limitations. Ohno and Mukawa (2003) developed an eyeball model requiring only two points for calibration. This model was applied in (Ohno *et al*., 03) to develop a head-free eye-gaze system using three cameras and one IR LED. 3D vision techniques involving multiple cameras and multiple point light sources were applied in (Shi *et al*, 03) in order to completely eliminate the cumbersome calibration procedure and to compensate for head movement limitations of users with severe disabilities. However, these systems appear to be quite complex and expensive. Yoo and Chung (2004) proposed a simpler system consisting of five IR LEDs attached to the monitor screen and only one pan-tilt CCD camera. This system is capable of determining eye-gaze direction without computing geometric relations between the eyes, the camera and the monitor. Zhu and Ji (2004) have proposed an IR LEDs ring instead of screen-mounted sensors. For the learning of user's specific data, the authors have developed a neural-network algorithm with the aim to improve robustness of the systems. Initial experiments demonstrated that these systems work well under relatively large head movements, though the accuracy did not reach the one commercially used eye-gaze systems. In spite of these developments, eye-gaze-tracking systems are still comparatively expensive and require great user attention and efforts to achieve proper cursor control.

Electrical signals generated by the permanent cornea-retina electrical potential difference, referred to as Electro-OculoGram (EOG), provide an alternative principle to develop cheaper eye-tracking systems. EOG activities may be registered by means of skin electrodes placed close to the eyes (Chen and Newman, 04). The obtained signals provide templates correlated with the eye-movement patterns, such as gaze fixation, blinks, saccades (i.e. fast conjugate changes of eye positions between fixations) etc. Chen and Newman (2004) have recently presented sophisticated algorithms for feature extraction from EOG gaze patterns that distinguish 2D eye-gaze displacements (i.e. coordinates defined by elevation and azimuth angles) that is in principle sufficient for cursor control, as well as various blink features (e.g. unconscious and intentional, normal and strong, intentionally strong blinks etc.). These features were applied to control 2D displacements of a robot manipulator, which is similar to controlling a mouse cursor. Compared to optical systems, EOG based systems provide favored possibilities for mouse pointer control, and are practical and valuable for people with SCI. However, their complex learning and calibration procedures present the main limitations and require further development.

2.6 Brain controlled systems

Brain computer interfaces (BCI) that allow computers to receive input signals directly from a user's brain, certainly represent one of the most challenging approaches towards developing HCI systems. Brain signals may be obtained by invasive means, by recording single neuron-activities within the brain. Recorded activities have a good spatial resolution (offer “fine control”) and may provide signals with many degrees of freedom supporting multidimensional movement control of a robot arm or a neuromotor prosthesis. However, invasive methods require recording electrodes to be implanted in motor and premotor areas of the cortex, which is not only risky, but also doesn’t ensure their long-term stable operations. Therefore these methods are being studied mainly in human primates with a surprising success, after appropriately trained monkeys were able to move cursor to different targets (Serruya *et al*., 02).

Current non-invasive (i.e. minimally-invasive) brain-controlled methods use electroencephalographic (EEG) waves recorded from surface-mounted electrodes (within an electrode cap) on the users scalp. These methods are safe and inexpensive, but they have limited bandwidth and spatial resolution. Continuum EEG brain waves spectrum involves various waves usually grouped by frequency (amplitudes are about 100 mV maximum) (Seabrock, 94), which can be processed in time- or frequency-domain. Frequency-domain techniques use spectral analysis and focus on specific frequencies on specific scalp locations.

The BCI developed at the Wadsworth Center (Wolpaw *et al.*, 1991) uses scalp-recorded μ-waves, i.e. μ rhytms with a frequency of 9-11 Hz, to control one-dimensional (1-D) cursor movements. Frequency bands of μ-waves are associated with the frontal motor cortex and their amplitude is diminished with a movement or with an intention to move. The conceptual model of a μ-rhytm control based BCI system is that users would only have to "think" about voluntary movement and the corresponding μ-rhythm would be suppressed in the motor cortex. Subjects can learn to control the amplitude of this waveform by trial and error when visualizing various motor activities. Thus, with a certain amount of feedback training, users can learn to move the cursor with the appropriate mental effort (in Wolpaw experiments, four out of five subjects acquired impressive control over their mu-rhytm during 10-45 minute sessions over a period of two months). Recently Wolpaw and McFarland (2004) extended BCI cursor control to 2-D by using a combination of μ and β rhytms. B-waves represents the second most common waveform occurring in electroencephalograms of the adult brain, characteristically having a frequency of 13-30 Hz and associated with an alert waking state (can also occur as a sign of anxiety or apprehension). In initial experiments a promising performance: hit rate of 92% (of a target appearing at one of 8 possible locations on screen) in average 2 sec with a cursor moving sampling rate of 50, was achieved. This result is comparable with the invasive BCI experiments.

An alternative approach to EEG-based computer interfaces utilizes exogenous electrophysiological activities, such as evoked electrical potential, referred to as P300

(Polikoff et al., 95). P300 potential may be produced in response to a significant but law probability event, i.e. various task-relevant stimuli such as letter or asterisk flashes. P300 has been utilized for cursor control (Polikoff et al, 95) as well as robotic arm movement (Vora, 04). In comparison to exogenous (internal, signals obtained inside users brain) methods, endogenous (external, signals obtained usually on users' scalp) BCI's provide a significantly better fit for a control model, which is used for training. Furthermore they are considerably faster. On the other hand, exogenous BCI's may not require extensive training like the endogenous ones, but do require a somewhat structured environment (Wolpaw and Vaughan, 00).

A novel BCI approach (Millan *et al.*, 03) aims to discover mental EEG patterns embedded in continuous EEG signals, rather than to look for predefined EEG phenomena. It relies on the fact that different mental tasks, such as imagination of movements, activate cortical areas at different extent producing different EEG patterns. Using a neural network algorithm, a computer learns after a few days of training to distinguish among different mental states, and may be programmed to perform a specific command (mental control) based on the mental pattern it detects (e.g. to move a mobile robot).

In spite of their encouraging results and newly acquired understanding of brain functions, it must be recognized that the current state of exciting and emerging BCI technology is still largely experimental. The BCI technology is still in the stages of laboratory research, early clinical trials and experiences (Fabiani et al., 04). Consequently, most BCI systems are expensive and require skilled personnel to operate them. The current research focuses on various still unresolved problems related to both invasive and non-invasive methods, such as practical/ethical issues, adaptation to users, increasing bandwidth, generalization of learning and training procedures, etc. A recent intermediate BCI methodology (Leuthardt et al., 04) uses electrocorticographic activity (EcoG) signals, recorded from cortical surface by means of subdural electrode arrays that do not penetrate into user's cortex, and could provide a powerful and practical alternative to both invasive and non-invasive methods.

2.7 Future research

Concluding this review of available AT HCI techniques and state of the art research, it may be stated that numerous results in this rapidly evolving field have greatly contributed to removing access barriers of human-computer interactions for people with disabilities, permitting them to operate computers for learning, communication and leisure. Nevertheless, there is not one single AT input device that will meet the needs of all motor-impaired people, since everyone's needs and abilities are different. Therefore, a major step in designing assistive technology devices and systems is to determine its "target audience". In spite of numerous developed AT systems, this challenge requires persistent improvements of existing devices, adaptation to specific user's requirements and needs, and developments of new, more intuitive, easy to operate, robust, user friendly, comfortable and responsive devices. More complex systems developed recently combine several user input channels (Marler, 04) in order to improve system performance, adaptability and robustness (e.g. Cyberlink-Brainfingers system, www.brainfingers.com, combines EMG, EOG and EEG channels). The development of standardized modular HCI units and their combination to optimally meet specific subject's needs also presents challenging future research goals.

Even fundamental mouse pointer control problems, used as benchmarks in numerous development efforts, requires further investigations and improvements. That presents the main research goal of this thesis.

3 Development of FMCC algorithms

This chapter defines new fuzzy logic-based algorithms for mouse pointer control. A key role for these algorithms plays the measurement of puff and sip signals through a novel airflow sensor providing information about airflow direction (puff or sip) as well as its intensity. It will be demonstrated how these relatively small improvements to puff-sip switch systems can be utilized to develop sophisticated cursor control methods.

3.1 User physical capabilities considerations – user requirements

When developing an assistive system, available physical abilities of targeted users must be considered, since they affect several parts of system design and implementation. These issues ultimately determine the requirements on system development (Buxton, 1986). Several factors have to be considered for successful identification of system access and operation methods. Key questions imposing specification requirements of FMCCS and their corresponding hypothetical answers are given in Table 3.1.

This table indicates that FMCCS does not impose high demands on user capabilities, nevertheless appears to be sensitive to involuntary head, eye or facial muscles motion, as well on user's fatigue and other environmental disturbances. This promises possible development of a robust application which can have a large potential user group. Specifically, FMCCS appears to be a feasible and realistic solution for target users sustaining severe SCI.

The specific problem of all puff-sip based systems is that injuries at the cervical level of the spinal cord, depending on the completeness of the lesion, can lead to paralysis of respiratory muscles. According to the Agency for Healthcare Research and Quality (AHRQ), patients with paralyzed respiratory muscles (specifically the diaphragm) may not have the ability of unassisted breathing and often must be placed on mechanical ventilators. Management of ventilatory insufficiencies implies specific breathing performance, characterized for example by good puff intensity, but difficulty to control the sip intensity and duration. This specific breathing performance needs to be especially considered.

Consideration factors	FMCCS hypothetical user
Does the user have controlled voluntary movement?	Good to limited breathing control (also assisted respiration) is most important required operation; Voluntary movement of face, mouth and tongue, are not relevant.
How accurate is the user's movement?	Moderate accuracy of breathing through the mouth (both inhale and exhale)
What is the user's range of movement?	Not relevant, i.e. normal range of movement of tongue and mouth.
How repeatable is the user's movement?	Not relevant for operation, even repeatability of respiratory functions is not required. Required is ability to be able to repeat operation codes by means of elemental puff-sip sequences. Exhaustion is not critical.
What strength does the user have when performing the movement?	Weak to moderate breathing strength, compared with able-bodied people.
What is the speed of user's movement?	Relatively slow puff and sip, compared with able-bodies people. Specific breathing performance in case of management of ventilatory insufficiency.
Is there more than one movement that can be utilized?	Not relevant, movements of neck, or facial muscles, are not required.

Table 3.1 User requirements

3.2 User inputs

The acquisition and interpretation of user input signals defines the input block of FMCCS, which regards the user as a system that produces input signals (expressions) via an assistive input device. Signals useful to interactions between the user and the system are kept and evaluated, while others are ignored. Accidentally caused input signals (e.g. involuntary puff or sip, cough etc.) are considered in a input signal processing Fuzzy

Inference System, a computing framework based on fuzzy set theory, if-then rules and fuzzy reasoning (Jang *et al.*, 97). As previously mentioned, the adaptive input device providing the interface between hypothetical users and the FMCCS consists of only one airflow sensor providing all information which is required: puff and sip signals resolved from airflow direction (positive or negative) and air mass flow, describing the intensity of the input, which represents the second input signal to be used for cursor control. There are several possibilities for processing of these relatively simple input signals. The preprocessing is based on A/D conversion and low-pass filtering suppressing noise. We use a digital filter for noise suppression and anti-aliasing effects. The obtained digital input is then normalized by mapping directional input to 1 (puff) and -1 (sip) and by scaling airflow intensity signals to the range [-10,10]. The time-continuous airflow signals provide possibilities to obtain more sophisticated input information, such as frequency spectrum of input signals or time derivatives describing rates of change of airflow mass (i.e. puff and sip velocity). These instrumented input channels increase system dimensionality, but impose complex requirements on user's capabilities and complicate his/her interactions with the system.

The processing of input signals is realized by an input fuzzy filter which performs all needed user-specific non-linear signal conditioning and filtering operations. These operations include scaling and weighting over the entire range, adaptive adjustment of dead-zone and saturation function. Fuzzy filters are especially effective for applications targeting people with disabilities, since they combine numerical and linguistic input, involving expert knowledge about specific user abilities.

Input-data Fuzzy Inference System (FIS) of the FMCCS interprets airflow input signals as a linguistic variables taking one of the following linguistic values:

- "*Weak Positive*", produced by a weak and positive (puff) airflow into the input device
- "*Moderate Positive*", produced by a moderate and positive (puff) airflow into the input device

- "*Strong Positive*", produced by a strong and positive (puff) airflow into the input device
- "*Weak Negative*", produced by a weak and negative (sip) airflow into the tube.
- "*Moderate Negative*", produced by a moderate and negative (sip) airflow into the tube.
- "*Strong Negative*", produced by a strong and negative (sip) airflow into the tube.
- "*Null*", special case of a not existing or insignificantly small airflow (e.g. caused by a reflex).

3.3 Mouse control demands – task requirements

The next system consideration involves task requirements, especially the desired mouse pointer control demands. This task requires 2D movement and positioning of the pointer on the screen. As previously mentioned, the input system provides two degrees of freedom, which represent two system input channels from the point of view of information technology. These channels offer a basis for the realization of cursor control. However, these signals are correlated and difficult to apply for independent x-y pointer positioning. Furthermore, mouse control requires several clicking operations to be successful. As previously mentioned, in systems using a pneumatic dual-action switch as the input device, multiple button switch functions are commonly realized by Morse code inputs. FMCCS focuses on three basic clicking functions: single click (both mouse buttons), double click (both mouse buttons), and drag, which can be coded by positive or negative airflow directions sequences.

When using Morse codes within pneumatic dual-switch input devices each sip is interpreted as a dot, and each puff by a dash (or vice versa). Different Morse code sequences of dots and dashes, defined by the assistive software system, are used to perform different control functions. The advantage of Morse code input interpretations is that every system control function can be described by one distinct Morse code sequence. However, given a large number of these functions, their corresponding input sequences tend to become very long. Because of this, Morse code interpretation of the input signals tends to create a very inefficient user experience, especially to users with limited breathing and/or learning capabilities. FMCCS implements relatively simple Morse code sequences of mostly three dot/dash combinations. Additionally, fuzzy-filtered input deals with specific user effects (i.e. involuntary breathing, or coughing) that ensure robust code interpretation without mistakes.

3.4 Interpretation of input signals – mouse scanning algorithms

Once user input has been interpreted, it can be used in the selection stage of the FMCCS. The selection technique, in general, describes a way for the user to point to desired items on the screen by moving the cursor to them. Two selection techniques most frequently used by people with severe motor impairments are Direct Selection, and Scanning (Hwang, 2003):

Direct selection refers to selection techniques in which the cursor points to target items directly (without cursor motion). Because it is fast and simple, direct selection is the most conductive selection approach, however, often requires motor-skills such as head, hand, and arm movement, not possessed by people with severe expression deficiencies. For this reason, indirect selection methods such as scanning have been developed.

Scanning selection involves presenting items until the user indicated that the desired item has been reached, therefore requiring far less motor skills than the direct selection technique. Items in the selection set are displayed through a predetermined or user-triggered item-selection process. Users have the option to either accept or decline the item presented by the facilitator. Scanning selection can include one or combination of the following scanning pattern techniques (Hwang, 2003):

- *Circular Scanning* (one-dimensional clockwise or counterclockwise scanning)
- *Linear Scanning* (one-dimensional row or column scanning)
- *Group Item Scanning* (two-dimensional row-columns or circular scanning)
- *Directed Scanning* (user-controlled direction scanning)

- *Selection Control Scanning* (user item selection through input encoding, e.g. Morse Code)

Even though scanning selection, compared with direct selection, significantly reduces the number of motor-skills required for its use, it still requires the user to have several cognitive abilities such as focusing on several items at once, memory, coordination and reflexes, as well as concentration.(Hwang, 2003).

Different scanning pattern techniques can be combined, in order to lessen the amount of required cognitive abilities and to achieve a more accurate and efficient user selection process. In addition, fuzzy logic can be used to further assist the selection process, specifically adding intelligent reasoning and rules to its decision-making. In the following section, developed novel fuzzy selection techniques will be briefly presented. For comparison purposes, conventional (traditional) linear and circular scanning methods are presented first.

3.4.1 Traditional scanning algorithms

Traditional Linear and Circular Scanning Technique consists of two steps, namely the Linear/Circular Selection Scan Line Move, and the Cursor Position Move. These are described in more detail in the following section.

3.4.1.1 Traditional Linear Scanning Technique

Traditional Linear Scanning Technique consists of two steps, namely the Selection Scan Line move, and the Cursor positioning more along the Scan Line. During the Selection Scan Line Move, an easily recognizable scan line moves horizontally across the screen. The user has a one-time option to select the movement to be top-down or down-up. Once the scan line has covered the desired object on the screen, the user stops the scan line movement. At this point cursor motion starts from the very left side of the scan line, moving to the right across the scan line. Once the cursor has covered the desired object, the user, again, performs the stop operation. Then the user can perform clicking operation, or can choose to restart the scanning process. Figure 3.1 shows a graphical representation of the traditional linear scanning technique.

There are several problems with the conventional line scanning technique. These include its requirements of good timing, no user-driven scan-speed control availability, no user cursor speed control availability, a high learning rate due to Morse Code inputs, and a small coverage area presentation.

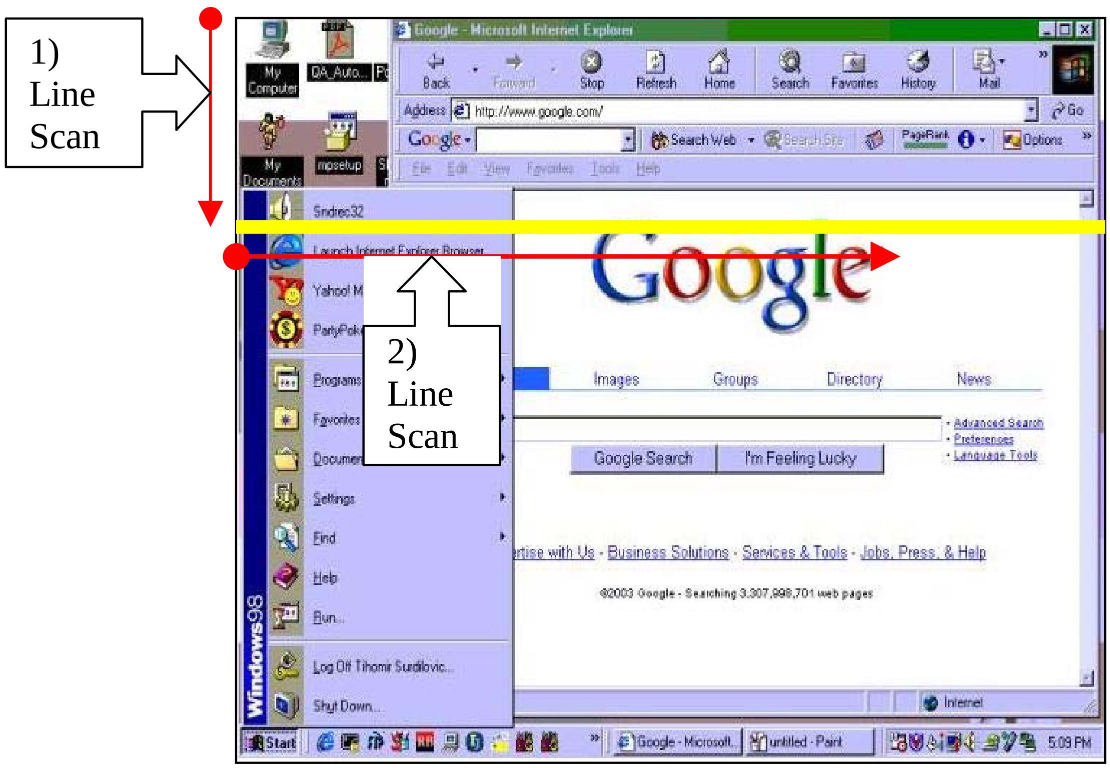


Fig. 3.1 Traditional Line Scanning Technique

3.4.1.2 Traditional Circular Scanning

Conventional Circular Scanning consists of following two steps:

- Circular Scan Line Move
- Cursor Position Move along the line

During Circular Scan Line Move, an easily recognizable scan line moves clockwise or counterclockwise, spanning the screen, starting at its center. The user has first option to select a desired movement direction of the scan line. Once the user stops the scan line, the circular scanning process pursues with the linear scanning process, where the cursor movement starts at the origin of the scan line, following it until stopped by the user, when a certain object on the screen has been reached. Circular Scanning has directional control comparable with Linear Scanning, and experiences similar problems to the above mentioned ones. Figure 3.2 sketches a graphical illustration of this method.

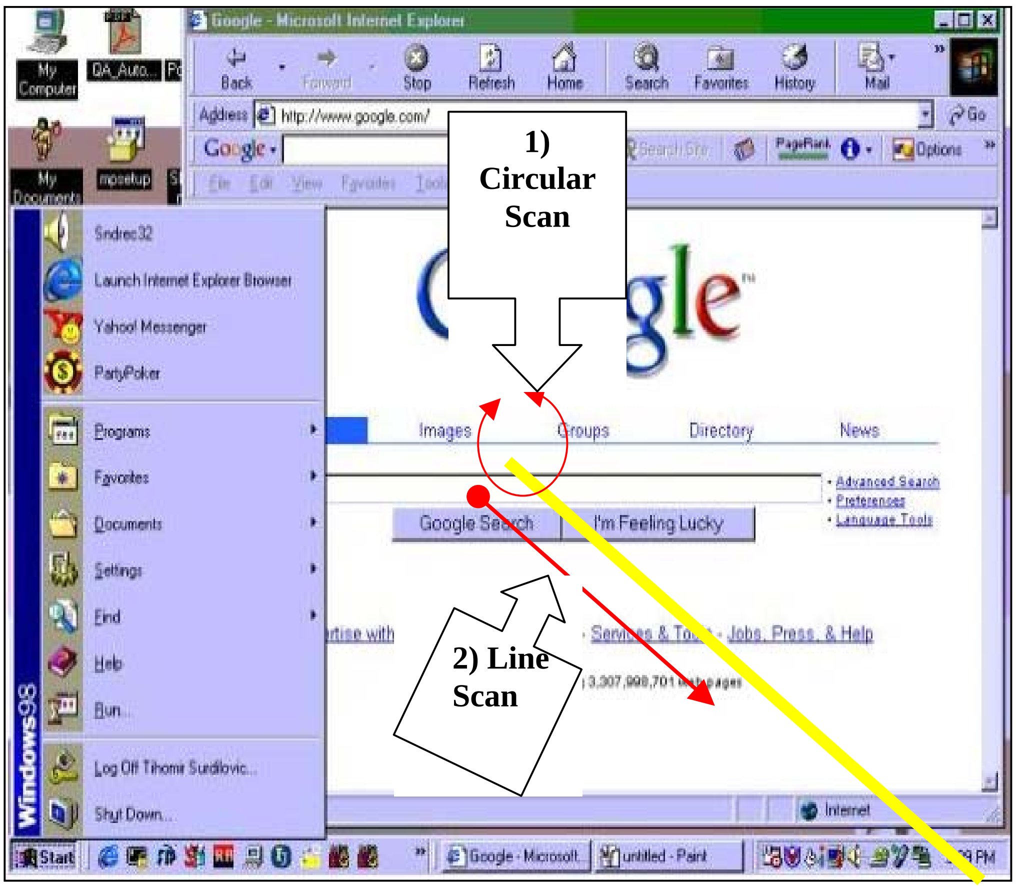


Fig. 3.2 Conventional Circular Scanning Technique

3.4.2 Fuzzy Scanning Methods

Based on the two presented traditional scanning methods, the following Fuzzy Scanning Methods can be defined:

3.4.2.1 Fuzzy Object Selection Scanning

Fuzzy Object Selection Scanning consists of three stages, namely the Fuzzy Circular Scan, Fuzzy Selection Area Scan, and the Fuzzy Object Selection Scan Stage. During the Fuzzy Circular Scan stage, the user controls the speed and direction of the selection scan area. A FIS defines the scan speed based on the fuzzy input airflow intensity signal, while the input direction signal defines the scan direction. Once the proper scan area position has been reached (the intensity becomes zero for a short prescribed time period), the user can perform the next Fuzzy Selection Area Scan operation. During this operation, the user is able to control the width of the scan area. Input intensity and direction define an increase and decrease of the scan area. The input intensity defines the rate at which the increase or decrease of the scan area is performed, and is defined by a FIS in the FMCCS. The goal of this step is to increase efficiency of pointing to larger objects, such as screen icons. When the user has defined the width of the scan area, she/he can pursue with the Fuzzy Object Selection operation. Based on the inclusion rate of a particular object within the selection area, which is described by a FIS, its selection priority can be determined. Also, the user can control the selection process, through the intensity of her/his input to the system. A "Strong positive" input would indicate a selection of an object further away from the scanning line origin; a "Weak negative" input would indicate the selection of an object near this origin. Figure 3.3 shows a graphical representation of the fuzzy object selection scanning process.

3.4.2.2 Fuzzy Selection Area Circular Scanning

Fuzzy Selection Area Circular Scanning consists of four stages, namely the Fuzzy Circular Scan, Fuzzy Selection Area Scan, Fuzzy Selection Area Circular Scan, and the Fuzzy Cursor Movement Scan stage.

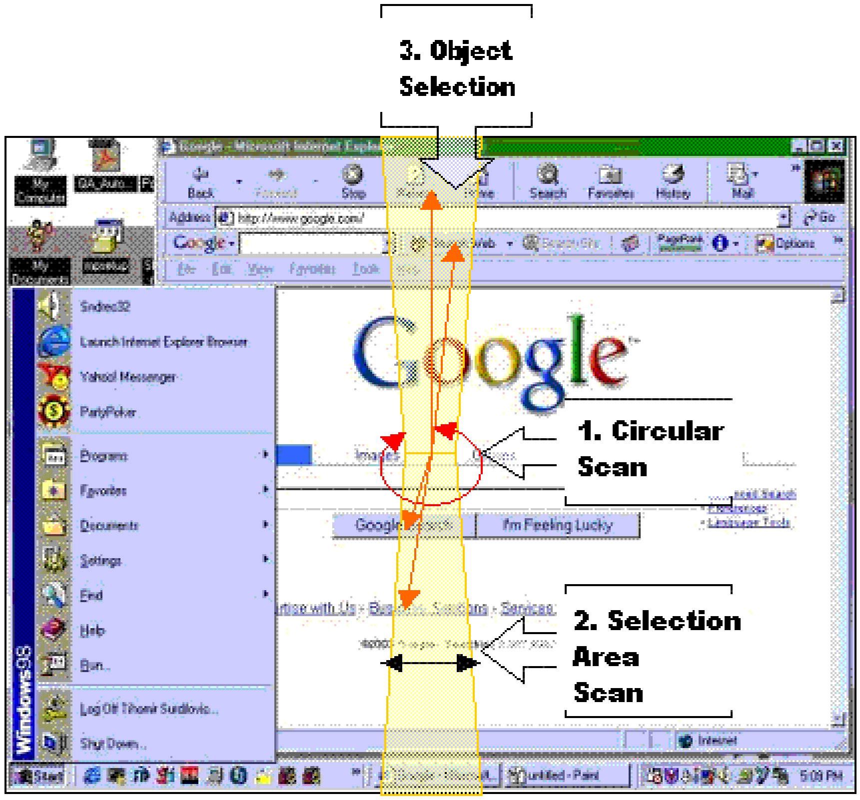


Fig. 3.3 Fuzzy Object Selection Scanning

The flow of this scanning technique is very similar to the fuzzy object selection technique, with the major difference that once the width of the scan area is determined, an additional circular scan is performed on the selection area only. The user, again, is able to control the scan speed and direction through her/his input. Once this secondary circular scan is performed, the user can control the cursor to the desired object covered by the secondary scan line. The user controls direction and speed of the cursor movement along this secondary scan line. For example, a “Weak negative” user input would indicate slow downward motion of the cursor along the secondary scan line. Figure 3.4 provides a graphical representation of this fuzzy scanning technique.

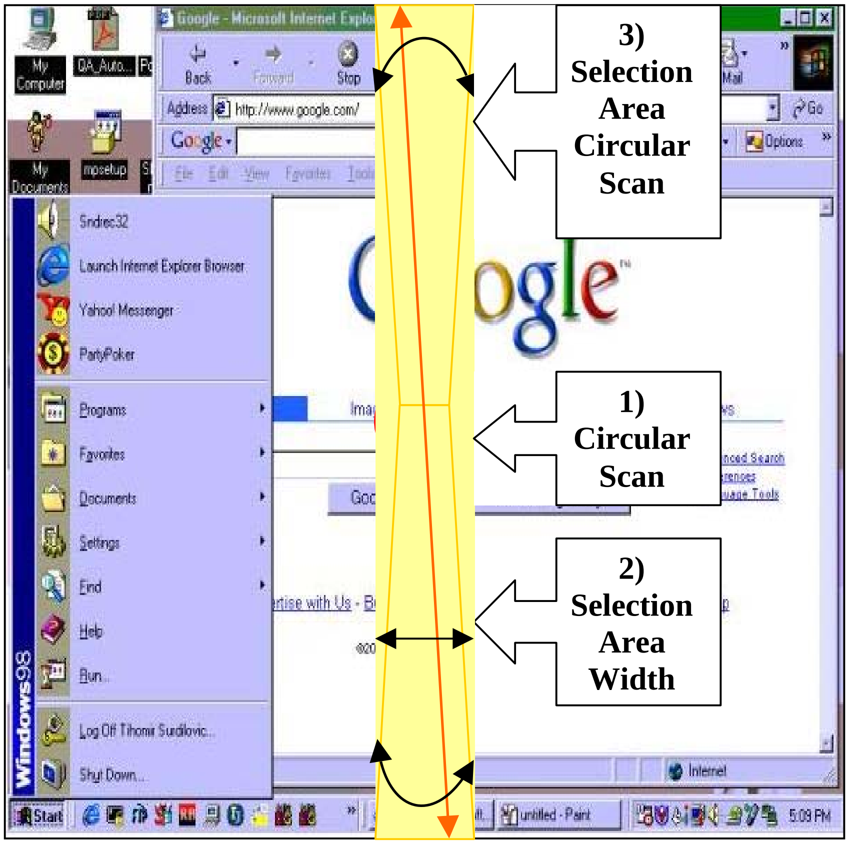


Fig. 3.4 Fuzzy Selection Area Selection Scanning

3.4.2.3 Fuzzy Selection Area Zoom

Fuzzy selection area zoom consists of four stages, namely the Fuzzy Circular Scan, Fuzzy Selection Area Scan, Fuzzy Selection Area Zoom, and the Fuzzy Object Selection Scan stage.

The flow of this scanning technique is very similar to the fuzzy object selection technique, with the addition of the scan area zoom. Within this operation the user is able to control the zoom rate of the scan area in which the desired object is included. To illustrate this, a “Medium Positive” user input defines a medium zoom-in of the scan area. The actual zoom-in value in this case is determined by the corresponding fuzzy control function, and it’s FIS that defines the “Medium Zoom-in” linguistic variable. Figure 3.5 shows this technique combined with the object selection technique. Figure 3.6

represents the fuzzy selection area zoom technique combined with fuzzy selection area circular scanning.

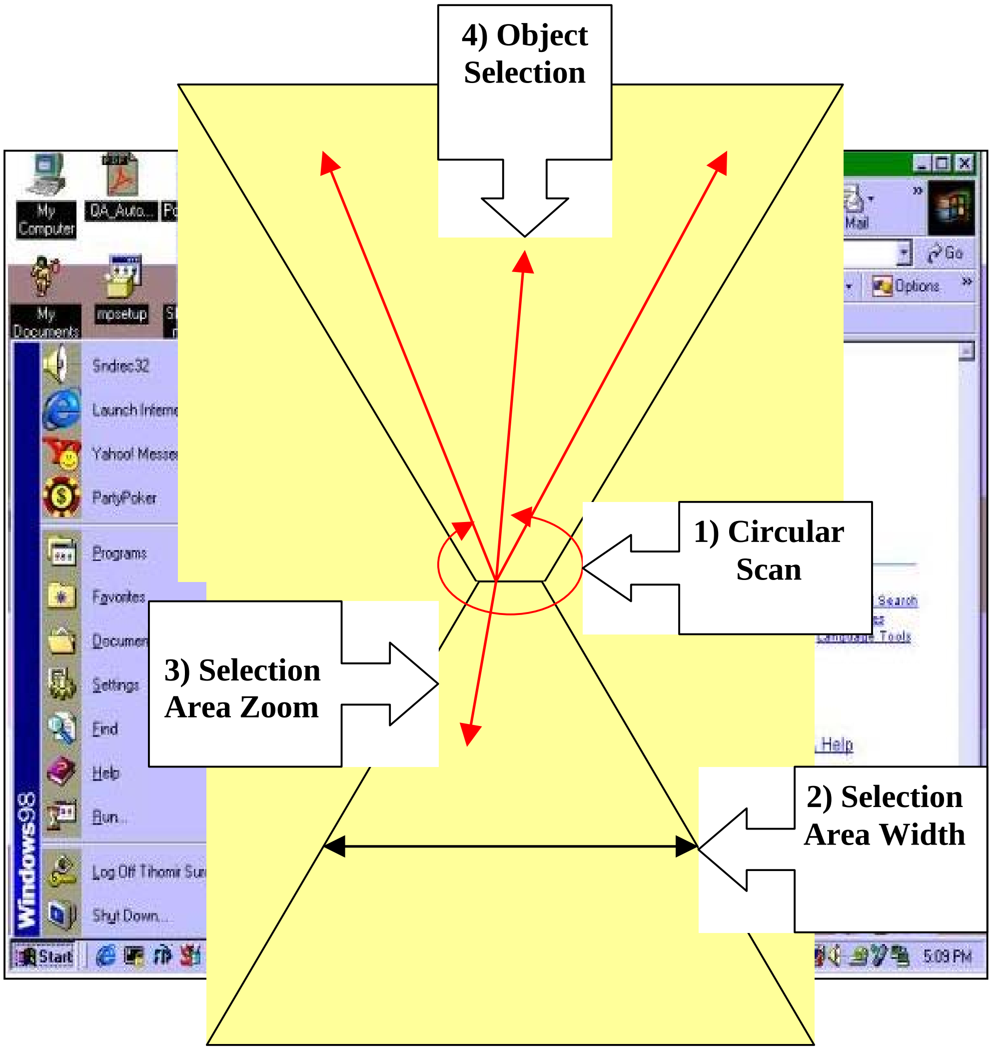


Fig. 3.5 Fuzzy Selection Area Zoom with Object Selection

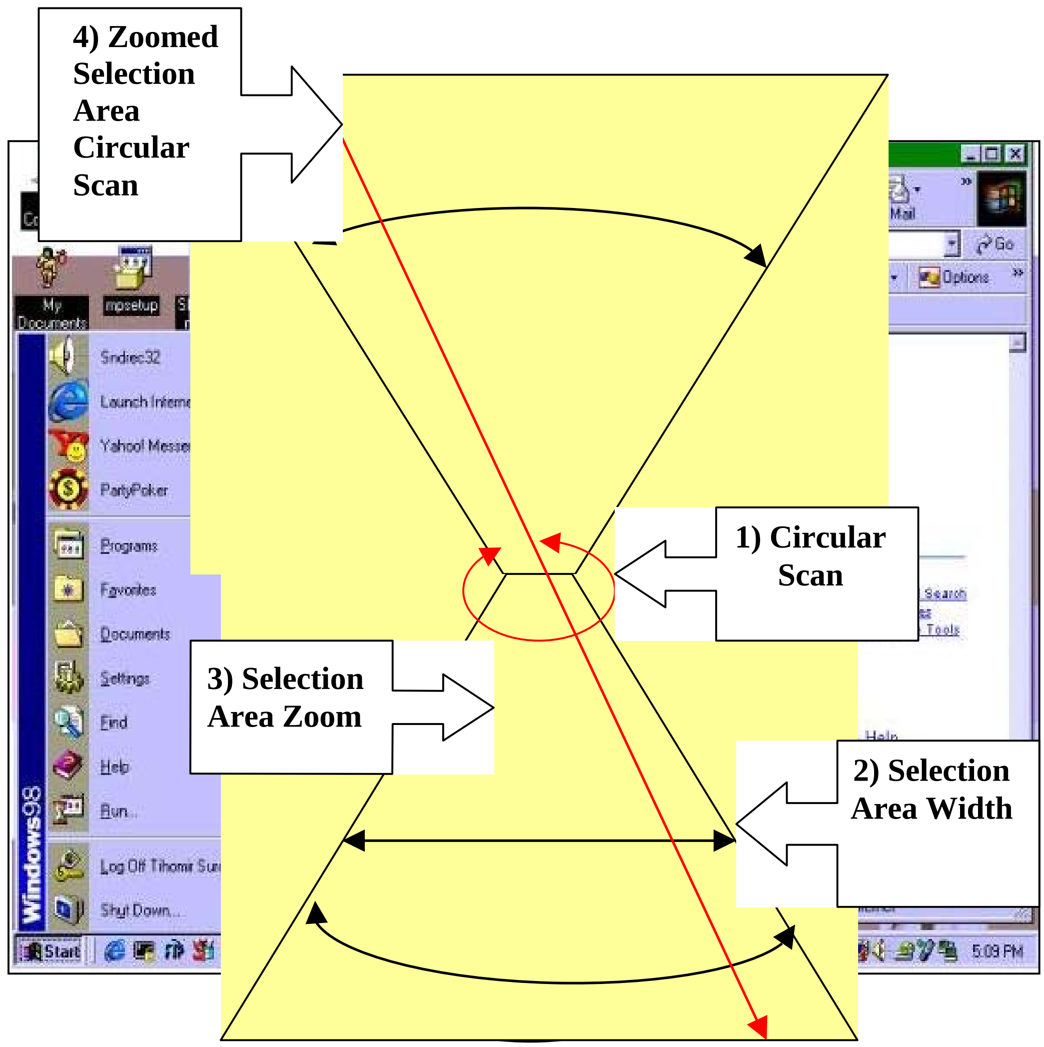


Fig. 3.6 Fuzzy Selection Area Zoom with Fuzzy Selection Area Circular Scanning

3.4.2.4 Fuzzy Cursor Zoom

The fuzzy cursor zoom technique is a quite simple fuzzy scanning approach intended to assist visually impaired users to make better selection by increasing the cursor size. The user has control of the cursor zoom rate, similarly to the zoom rate of the scan area described earlier. Figure 3.7 shows the graphical representation of this fuzzy scanning technique.

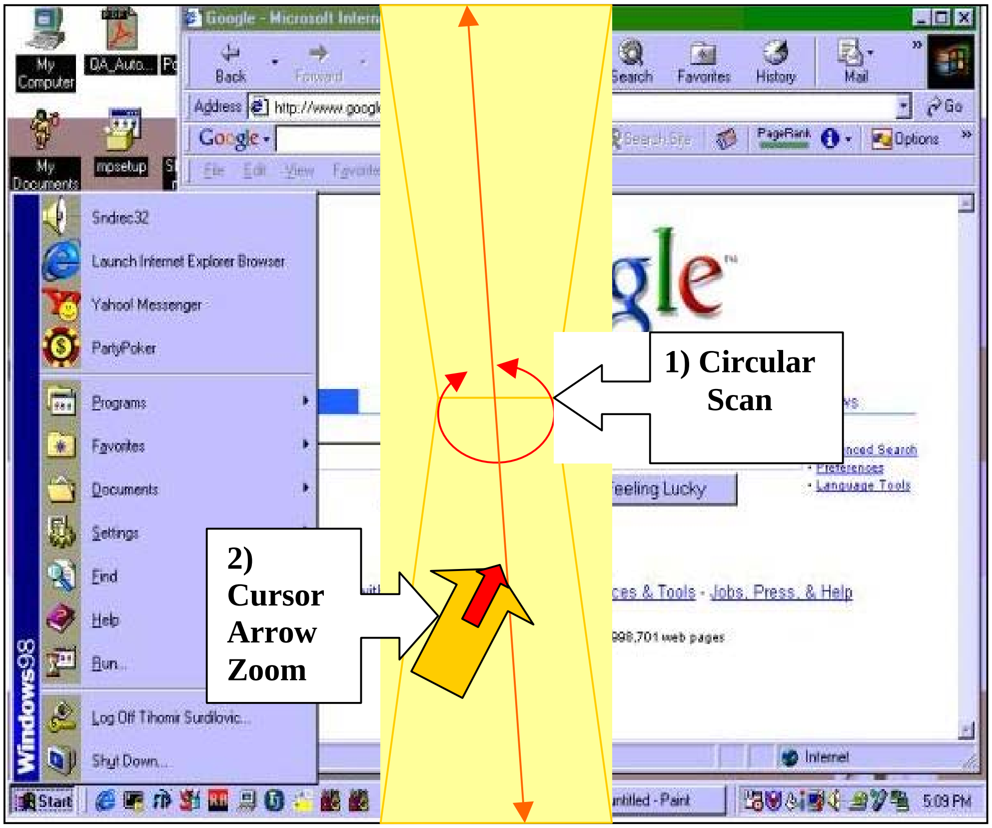


Fig. 3.7 Fuzzy Cursor Zoom

Many other modifications of fuzzy scanning techniques can be developed based on the ones presented above. Given each user's unique expression deficiencies, as well as voluntary movement abilities, unique scanning techniques can be created for a particular user, not only by fine-tuning FIS rules and parameters, as well by adding specific algorithms based on fuzzy methods. This can contribute to a more realistic and transparent utilization of the FMCCS.

3.5 FMCCS Control Architecture

Architecture of the FMCCS is based on the Finite-State machine model. An Input-Data FIS first interprets user input. The crisp output of this FIS, which is in the range (-10, 10) and corresponds to one of the seven previously mentioned linguistic variables, becomes the single input to the FMCCS, and is fed into the System State logic unit. This unit includes a state flow, which represents the systems selection process states. Depending on the fuzzy scanning techniques chosen for a particular user, the system states and their flow may vary. Given user input and the current state of the system, a corresponding fuzzy control function is executed. This function can use system input its corresponding FIS to determine different outputs, i.e. user's action parameters, such as cursor or scan-line speed, or selection area width. A command is send to the underlying control module to execute the determined decision on the computer screen. Once the command has been executed, the system can choose to stay in its current state, or move to another state, based on the defined state flow.

In order to achieve high scalability of the system, the state flow, fuzzy control functions, and the inference systems are divided into three separate and independent parts of the FMCCS. By this means each parts of the system can easily be added to, adapted to a specific user, removed, or replaced, which allows the system to be targeted towards unique user's motor-impairments, as well as personal preferences. The FMCCS thus becomes a flexible framework for the creation of fuzzy selection procedures. Figure 3.8 shows a graphical representation of the FMCCS. In the following design and realization of each specific FIS will briefly be described.

3.5.1 Input-Data Fuzzy Inference System (FIS)

As previously mentioned, the FMCCS is based on two input signals produced by the assistive input device: airflow direction and intensity. A Mamdani model (Jang *et al.*, 97) FIS is used to define the mapping of input parameters to a single output, using fuzzy logic. The output of this input data processing FIS is then used as the single input to the FMCCS. The input signal describing the direction is divided into three overlapping fuzzy sets, called “Negative”, “Positive”, and “Null”. A “Negative” input represents users' sip (inhale) operation, whereas a “Positive” input represents users' puff (exhale) operation. “Null” input describes users' lack of input, as well as small involuntary inputs provided by user or the environment. The membership functions of each of the three fuzzy sets are defined as "Negative (trapmf(-1 -1 -0.2 0)), "Positive" (trapmf(0 0.2 1 1)), and "Null(trimf(-0.1 0 0.1)), where trapmf() and trimf() stand for Trapezoidal, and Triangular Membership Function, respectively. The range of this input data signal is (-1, 1). By means of the Matlab FIS Editor, the membership functions of the input data FIS is sketched and presented in (Fig. 3.9).

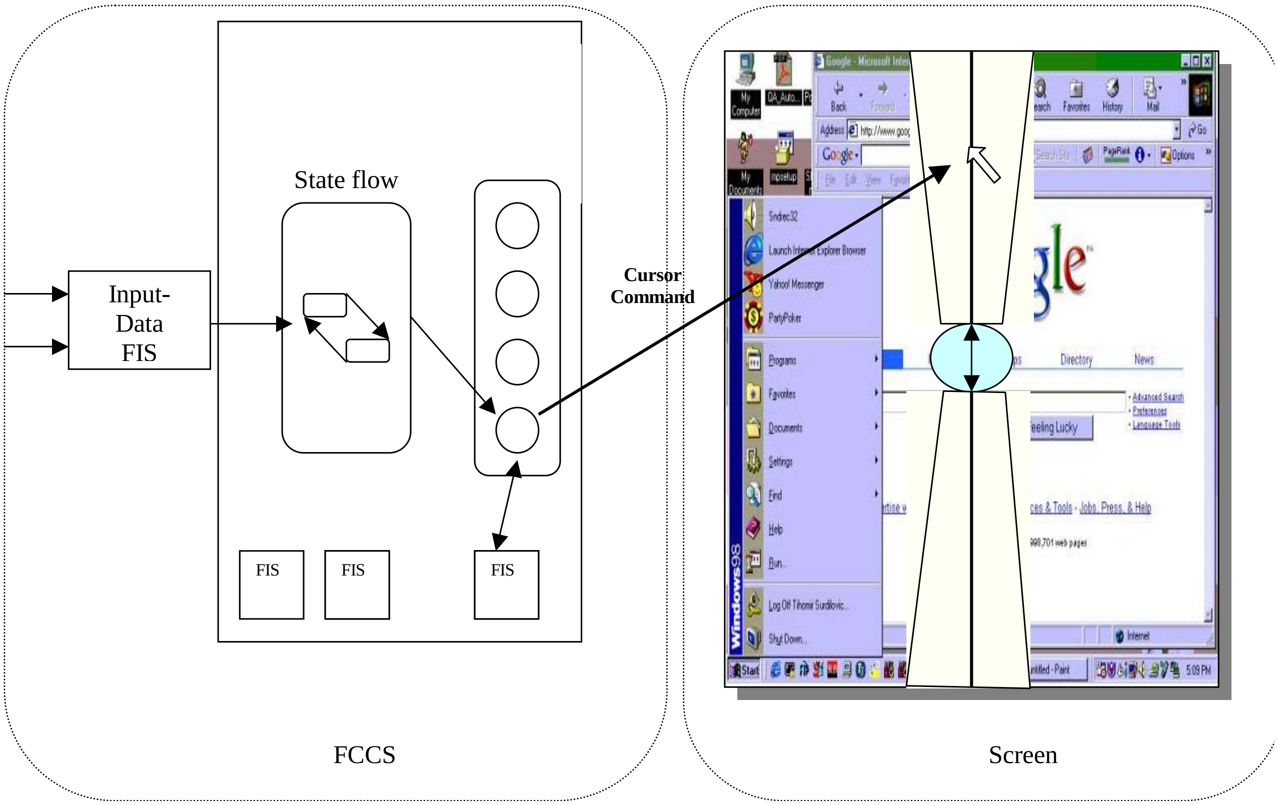


Fig. 3.8 FMCCS architecture

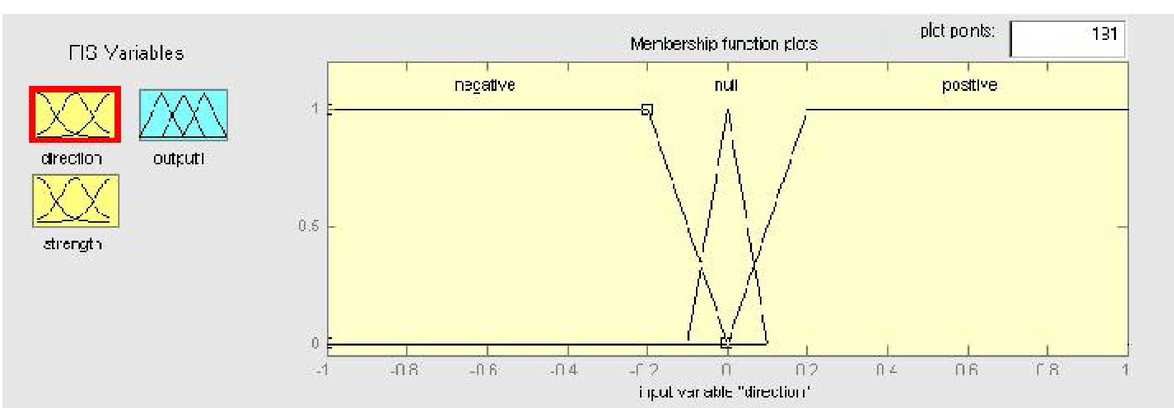


Fig. 3.9 Input direction FIS membership function

Similarly, we define the input signal describing the strength as with fuzzy sets as "Weak" (gaussmf(0.2 0.025)), "Moderate" (gaussmf(0.1269 0.5)), and "Strong" (gaussmf(0.2 0.994)), where gaussmf() describes the Gaussian membership function. The range of the input, in this case, is (0,1), and the graphical representation of the membership functions of each of the fuzzy sets has been given in (Fig. 3.10).

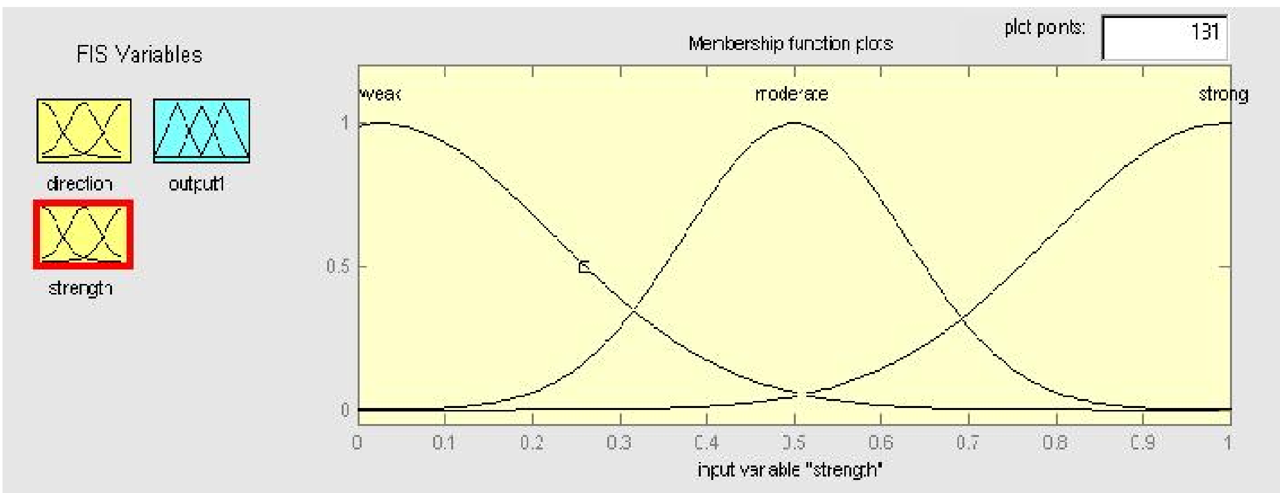


Fig. 3.10 Input intensity FIS membership function

3.5.2 Output FIS

The output of the FIS, before the defuzzification step, is described by seven overlapping fuzzy sets, namely "Strong Negative" (gaussmf(0.1845 -0.95)), "Moderate Negative" (gaussmf(0.1213 -0.5)), "Weak Negative" (gaussmf(0.07078 -0.25)), "Null" (gaussmf(0.04247 1.735e-018)), "Weak Positive" (gaussmf(0.04247 1.735e-018)), "Moderate Positive" (gaussmf(0.1213 0.5)), and "Strong Positive" (gaussmf(0.185 0.95)), where the range of the FIS output is (-1, 1), and the output membership functions is illustrated in (Fig. 3.11).

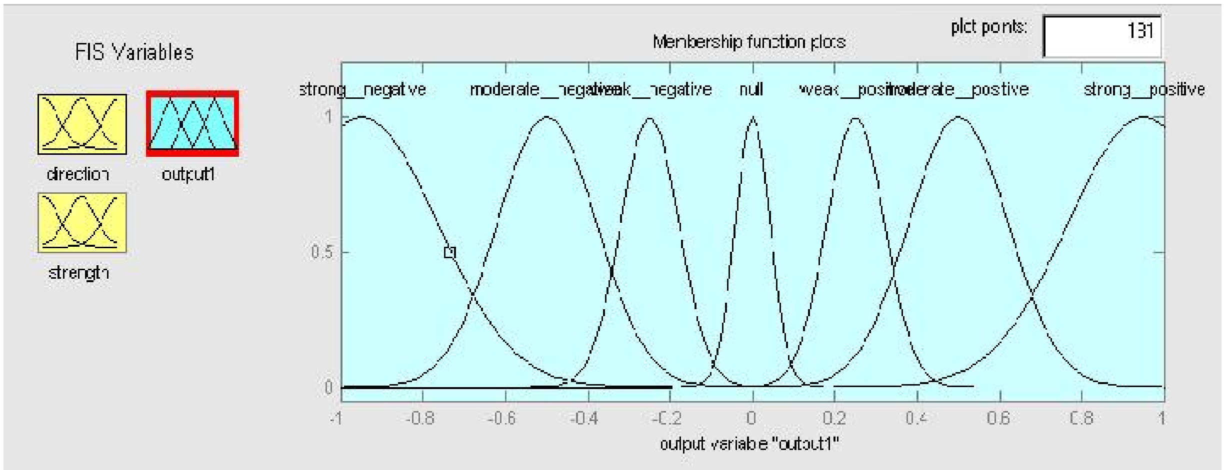


Fig. 3.11 Output FIS membership function

The fuzzy reasoning step of the FIS is accomplished through the following nine fuzzy if-then rules as shown in following MATLAB-generated image:

1. If (direction is null) and (strength is weak) then (output1 is null) (1)
2. If (direction is null) and (strength is moderate) then (output1 is null) (1)
3. If (direction is null) and (strength is strong) then (output1 is null) (1)
4. If (direction is negative) and (strength is weak) then (output1 is weak__negative) (1)
5. If (direction is negative) and (strength is moderate) then (output1 is moderate__negative) (1)
6. If (direction is negative) and (strength is strong) then (output1 is strong__negative) (1)
7. If (direction is positive) and (strength is weak) then (output1 is weak__positive) (1)
8. If (direction is positive) and (strength is moderate) then (output1 is moderate__positive) (1)
9. If (direction is positive) and (strength is strong) then (output1 is strong__positive) (1)

Table 3.2 demonstrates the design of the FIS engine specifying the implemented operations (i.e. methods) of the Mamdani FIS. The detailed definition of applied operators is given in (Jang *et al*., 97). The overall input-output surface for this FIS is sketched in Fig. 3.12. This surface defines is based on the mapping of input values (strength and direction) based on the implemented fuzzy if-then rules. Defuzzification method used within this fuzzy inference system was Centroid of Area (COA). This defuzzification method is based on finding the center of gravity of the geometrical shape representing the overall output membership function (MF) of the system. The described FIS produces a crisp output in the range of (-1, 1), which defines the single input to the FMCCS.

It is worth mentioning that presented membership functions were adjusted based on simulations described bellow, and the experience obtained through user-input (see the next chapter). Advantage of this fuzzy approach is that filtering and control functions can easily be adjusted to meet specific subject needs and abilities. A more sophisticated approach, based on neuro-fuzzy methods, capable of more systematically learning user abilities and intents, is a subject of future investigations.

Design Method (Operators)	FIS Realization
AND	T-norm (MIN operator)
OR	T-conorm (MAX operator)
Implication	Product operator
Aggregation	Sup operator
Defuzzification	Center of Area (COA)

Table 3.2 FIS engine design

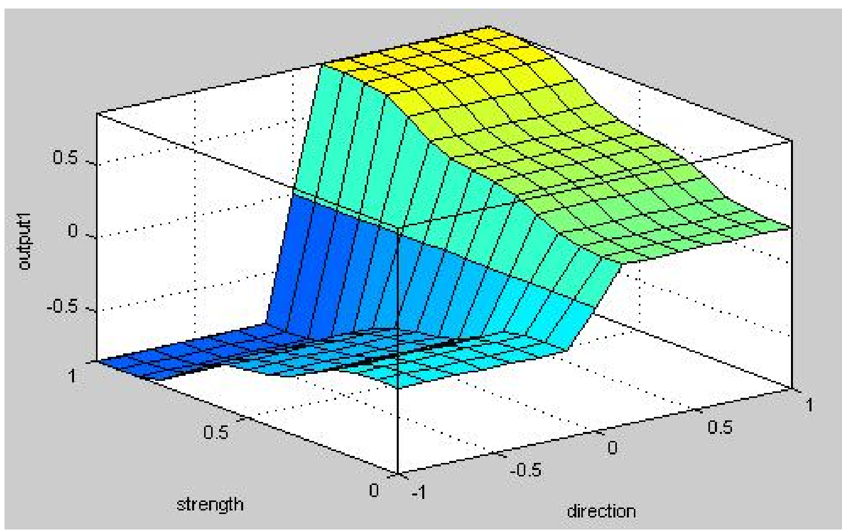


Fig. 3.12 FIS input/output surface

3.5.3 FMCCS States

Depending on the scanning strategy used for a particular user, the FCCS state flow may consist of different states and transition conditions. Overall, we define following four major states of the FMCCS, namely the Initial (Idle), Scanning, Clicking, and the Error states

Each of the major FMCCS states can include one or more sub-states, describing an implemented scanning strategy. All sub-states within initial implementation of the FMCCS are:

- Initial State (I.S.)
  - No sub-states defined
- Scanning State (S.S.)
  - Directional Arrow Scan (D.A.S.)
  - Area Width (A.W.)
  - Object Selection (O.S.)
  - Area Scan (A.S.)
  - Cursor Movement (C.M.)
- Zooming State (Z.S.)
  - Area Zoom (A.Z.)
  - Cursor Zoom (C.Z.)
- Clicking State (C.S.)
  - No sub-states defined
- Error State (E.S.)
  - No sub-states defined

Figure 3.13 shows a representation of FMCCS state-flow transition diagram for the Selection Area Zoom operation within Selection Area Circular Scanning approach. In this diagram, the above abbreviations are used to denote final state blocks.

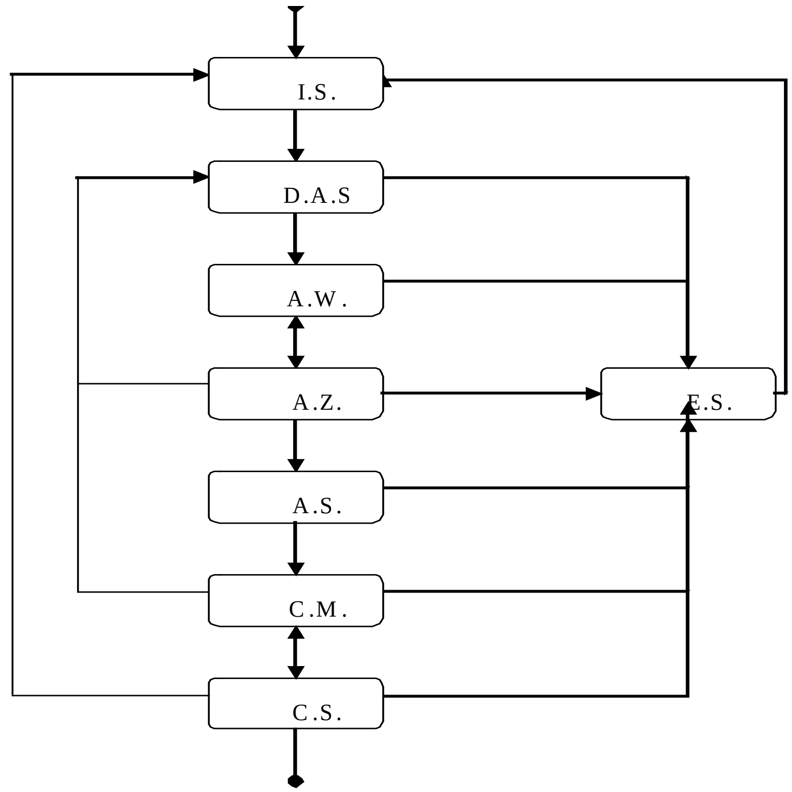


Fig. 3.13 FIS state-flow diagram

3.5.4 Fuzzy Control Functions

Fuzzy control functions (FCF) play a major part of the FMCCS. Each state consists of one or more fuzzy control functions to accomplish the scanning logic of that state. Once the system moves to a particular state, it executes corresponding fuzzy control function, passing the input from the Input-FIS and receiving output fuzzy value. For example, the cursor speed fuzzy control functions action submits intensity input to Cursor Speed FIS and receives output that determines the actual speed-up or slow-down of the cursor movement. Depending on the underlying operating system that the FMCCS runs on, the fuzzy control functions include OS-specific objects for controlling graphics environment, i.e. mouse pointer motion and commands. In case of an error found during the execution of a particular fuzzy control function, an error handler is activated moving the actual state to the Error State and providing corresponding error code to the user with recovery actions to be pursued. The Table 3.3 sums up fuzzy control functions and their corresponding FIS implemented in the current FMCCS:

System State	Fuzzy Control Function	Fuzzy Inference System
Initial	Initial Positioning FCF	None
Scanning	Directional Arrow FCF Object Selection FCF	Directional Arrow Speed FIS Object Priority FIS
Zooming	Area Zoom FCF Cursor Zoom FCF	Area Zoom FIS Cursor Zoom FIS
Clicking	Single-Click FCF Double-Click FCF	Cursor-Click FIS
Error	Error Message FCF	None

Table 3.3 FIS engine design

Table 3.3 shows that each state has its own fuzzy control function, which is executed once the system is in that particular state. Each of these control functions then uses its defined FIS, and uses its crisp defuzzified value as input. The only exceptions are the Initial and the Error states, which do not have a corresponding FIS.

3.5.5 Pseudo-code implementation

This sections show the pseudo-code implementation of two defined fuzzy control functions and their FIS description:

3.5.5.1 Directional Arrow Scan FCF

The implemented pseudo-code has the following form:

```
function directionArrowScan(userInput)
  var arrowDirection = defaultDirection
  var arrowSpeed = none
  boolean validPositive = userInput > 0 + inputThreshold
  boolean validNegative = userInput < 0 + inputThreshold
  if(validPositive || validNegative)
          if(validPositive)
```

```
                arrowDirection = clockwise
        else if(validNegative)
                arrowDirection = counterclockwise

    arrowSpeed = DirectionArrowFIS(userInput)

moveDirectionArrow(arrowDirection, arrowSpeed)
```

The *directionArrow* function takes as input the *userInput* parameter which describes input coming into the system. Depending if *userInput* is larger or smaller than a system defined input threshold, which describes the "Null" input to the system, the function determines the cursor arrow direction. The speed with which the cursor is to be moves in the identified direction is computed by receiving the crisp defuzzified value from the Directional Arrow FIS through the *DirectionArrowFIS* function. Finally the *moveDirectionArrow* function performs the actual cursor movement in the specified direction and speed. The implementation of this function may vary according to the underlying operating system.

The I/O membership functions of the directional Arrow Scan FIS are presented in Figures 3.14 and 3.15 respectively

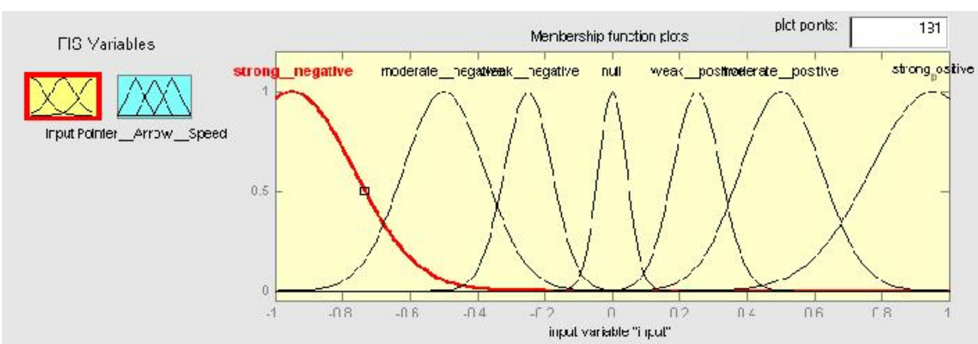


Fig. 3.14 Input functions (output of the Input-FIS)

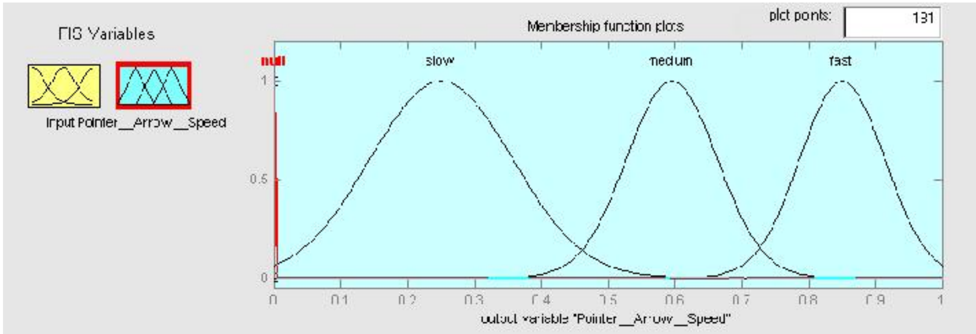


Fig. 3.15 Output functions (Directional Arrow Speed defined as “null”, “slow”, “medium”, and “fast”)

The implemented fuzzy If-Then Rules of the FIS have the following form as shown in following MATLAB-generated image:

1. If (input is null) then (Pointer__Arrow__Speed is null) (1)
2. If (input is strong__negative) then (Pointer__Arrow__Speed is fast) (1)
3. If (input is moderate__negative) then (Pointer__Arrow__Speed is medium) (1)
4. If (input is weak__negative) then (Pointer__Arrow__Speed is slow) (1)
5. If (input is weak__positive) then (Pointer__Arrow__Speed is slow) (1)
6. If (input is moderate__positive) then (Pointer__Arrow__Speed is medium) (1)
7. If (input is strong_positive) then (Pointer__Arrow__Speed is fast) (1)

3.5.5.2 Object Selection Fuzzy Control Function (FCF)

The implemented pseudo-code can be written in the following meta-form:

```
function objectSelection(userInput, includedObject[])
  boolean validPositive = userInput > 0 + inputThreshold
  boolean validNegative = userInput < 0 + inputThreshold
  if(validPositive || validNegative)
        for(i=0;<includedObject.size;i++)
              includedObjects[i].priority =
              ObjectSelectionFIS(userInput,
                    includedObjects[i].inclusionRate)
```

sortIncludedObjectsByInclusionRate(includedObjects[])

selectBestObject(includedObjects[])

The *objectSelection* function takes two parameters, namely *userInput* describing the input to the system, as well as *includedObjects*[], an array describing all screen objects and their information that are included within the current selection area. Depending on the user input being “positive” (puff) or “negative” (sip), each included object’s “priority” is calculated. Using the *ObjectSelection FIS* and information about each individual object such as distance from cursor and inclusion rate, the priority describes, given user input, which object would be most likely selected. Finally the *selectBestObject* performs the cursor movement to the object with the highest priority value.

The input membership functions from the Input FIS, as well as input for an object inclusion within the selection area, defined as: “Small Far”, “Medium Far”, “Large Far”, “Total Far”, “Small Near”, “Medium Near”, “Large Near” and “Total Near**”,** are presented in Figures 3.16 and 3.17 respectively.

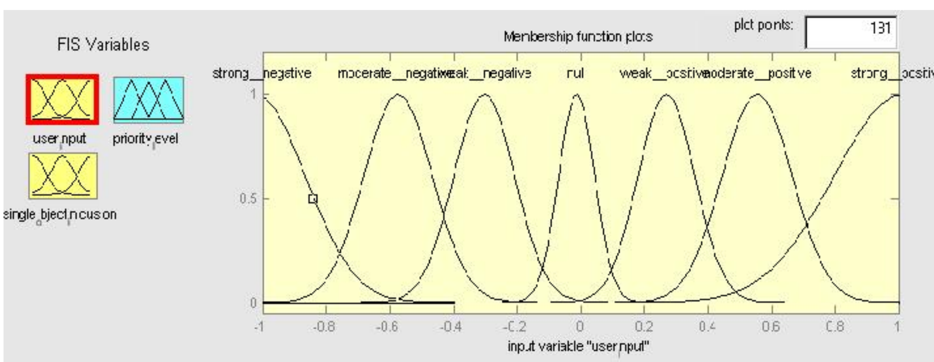


Fig. 3.16 Input 1: output of the Input-FIS

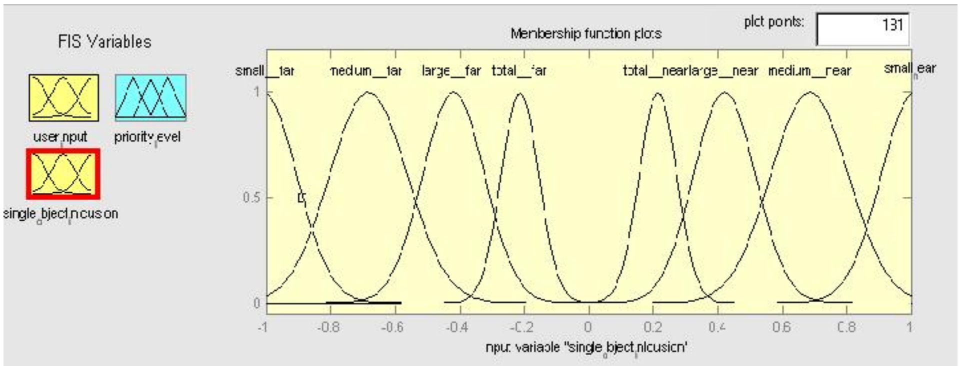


Fig. 3.17 Input 2: object Inclusion within the selection area

The output functions providing Object Selection Priority, defined as: “Low”, “Medium”, “High” and “Very High”, are sketched in Fig. 3.18.

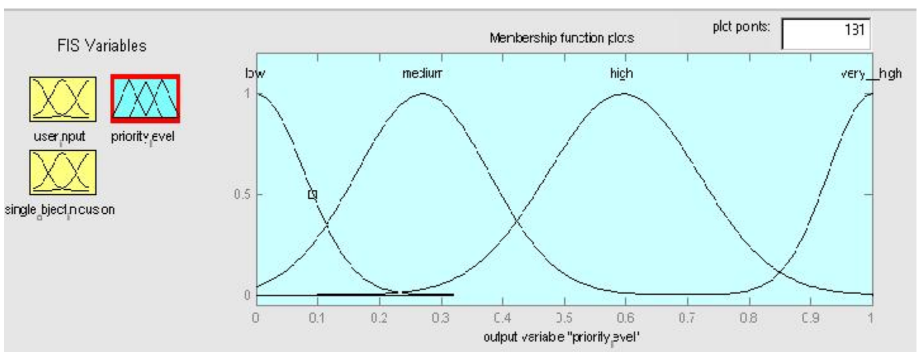


Fig. 3.18 Output FIS

3.5.6 FMCCS Simulation environment

Simulation of the FMCCS provides a functional environment for tuning and optimizing the FIS models, as well as for identification of specific user parameters. The entire FMCCS is implemented in MATLAB and SIMULINK environment providing ready-to use functions and interfaces for simulation of FIS, state flow transition, as well environment, i.e. GUI based on the Virtual Reality Modeling Language (VRML) model. The implemented SIMULINK model (Fig. 3.19) describes:

o Simulation of the assistive input device through a Spirometer (device for measuring lung capacity, e.g. airflow realized during breathing) model (Fig. 3.20)

o Fuzzy Logic Controller implementation using the Input-FIS membership functions and fuzzy if-then rules previously shown (Fig. 3.21)

o Inline directional arrow scan FCF implementation

o Simulation of the computer screen (Fig. 3.22)

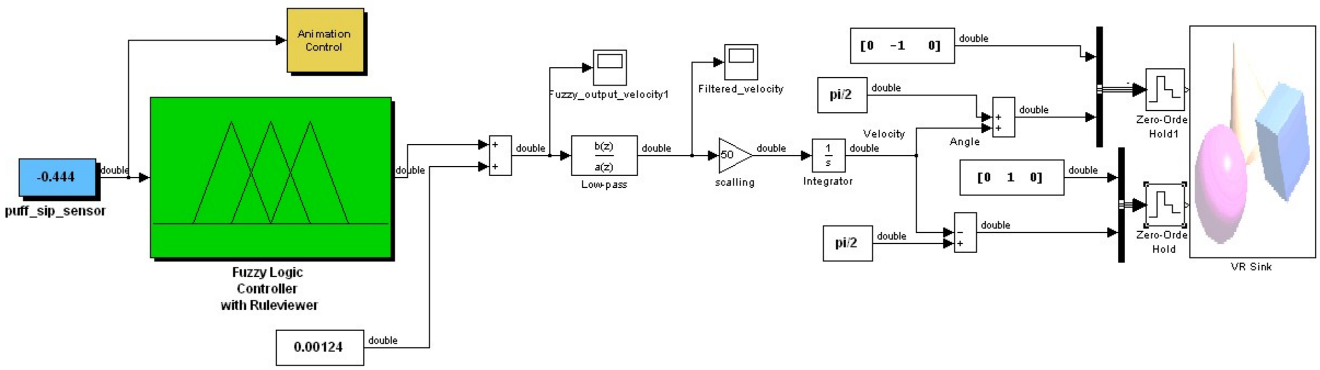


Fig. 3.19 Simulink FMCCS implementation

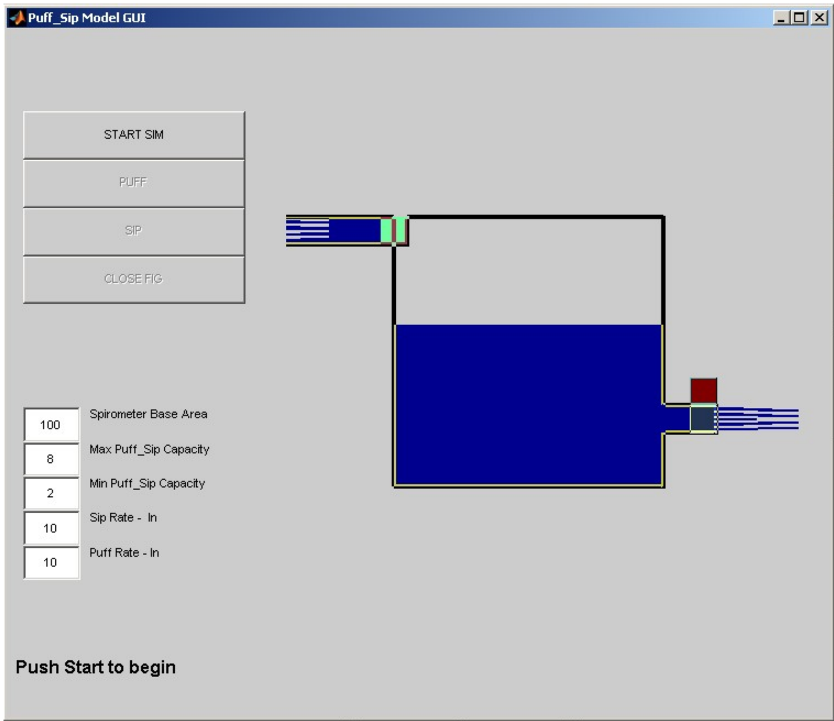


Fig. 3.20 Simulation of airflow through an assistive input device (Spirometer)

Figure 3.21 shows the simulation of the assistive input device. It is symbolic representation of the fluid level which is given through a mouse clock to simulate puff/sip operations.

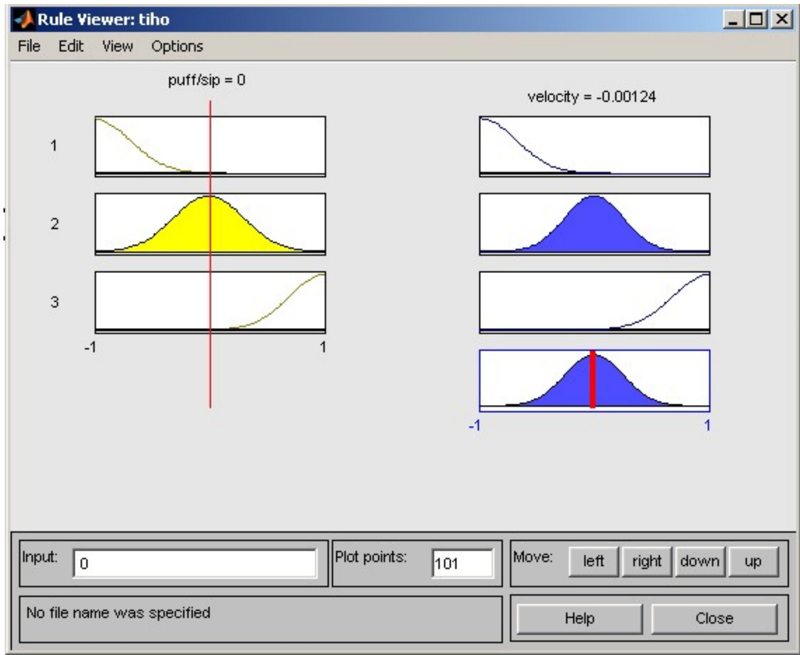
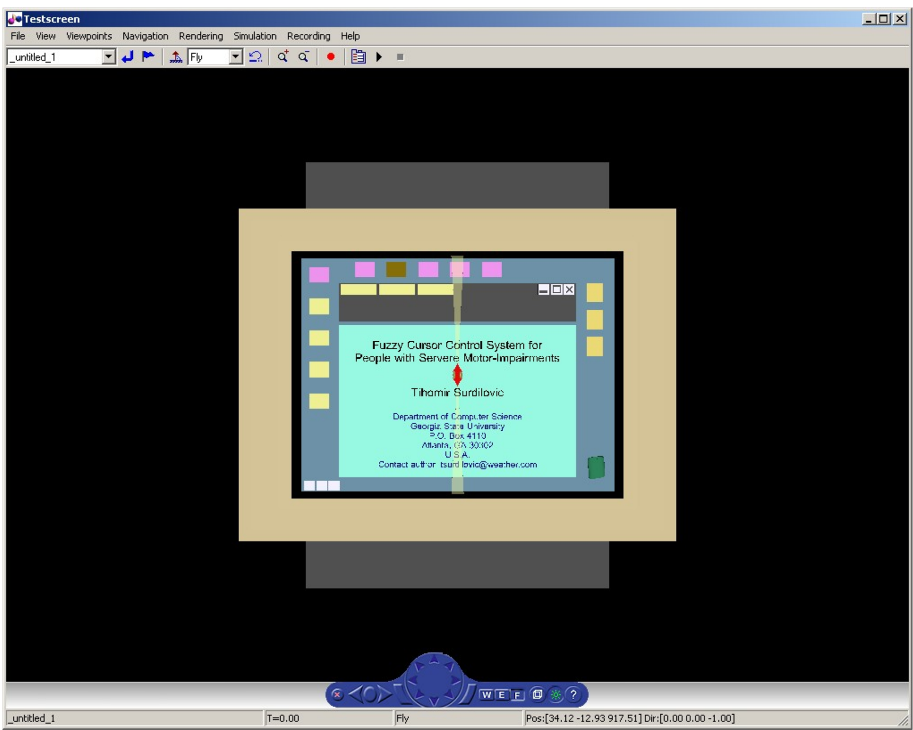


Fig. 3.21 Fuzzy rule viewer



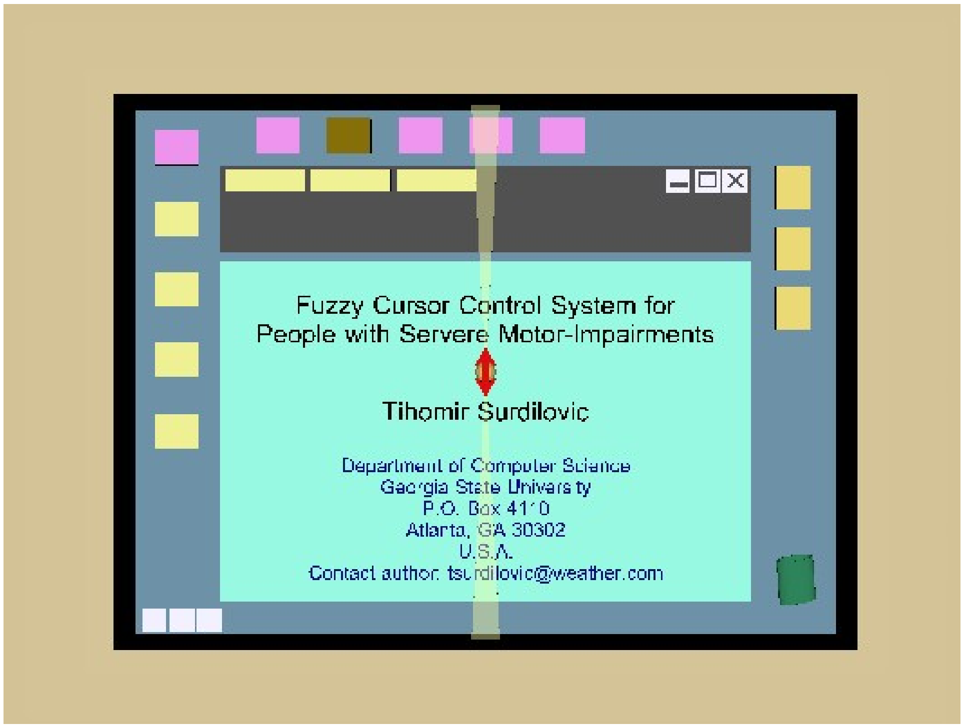


Fig. 3.22 Screen VRML model integrated in SIMULINK

Figure 3.22 shows the simulated computer screen using VRLM. This screen allows users to perform fuzzy circular scan, as well as the fuzzy area scanning method.Some typical simulation results are presented in Figures 3.23 and 3.24. Figure 3.23 shows the unfiltered input signals realized by user puff/sip operations. Figure 3.24 shows the filtered continuous signal which can be mapped into continuous mouse movements. Without this filtering process, the mouse movement using only input signals from Figure 3.23 would be very noisy and choppy. It should be mentioned that the implemented simulator contributed significantly to gaining experience with fuzzy control and mouse cursor controlling on the simulated screen. It was easy to implement and test various ideas in the developed environment, what represents the main benefit of the simulation model. The obtained knowledge was useful during system implementation and testing.

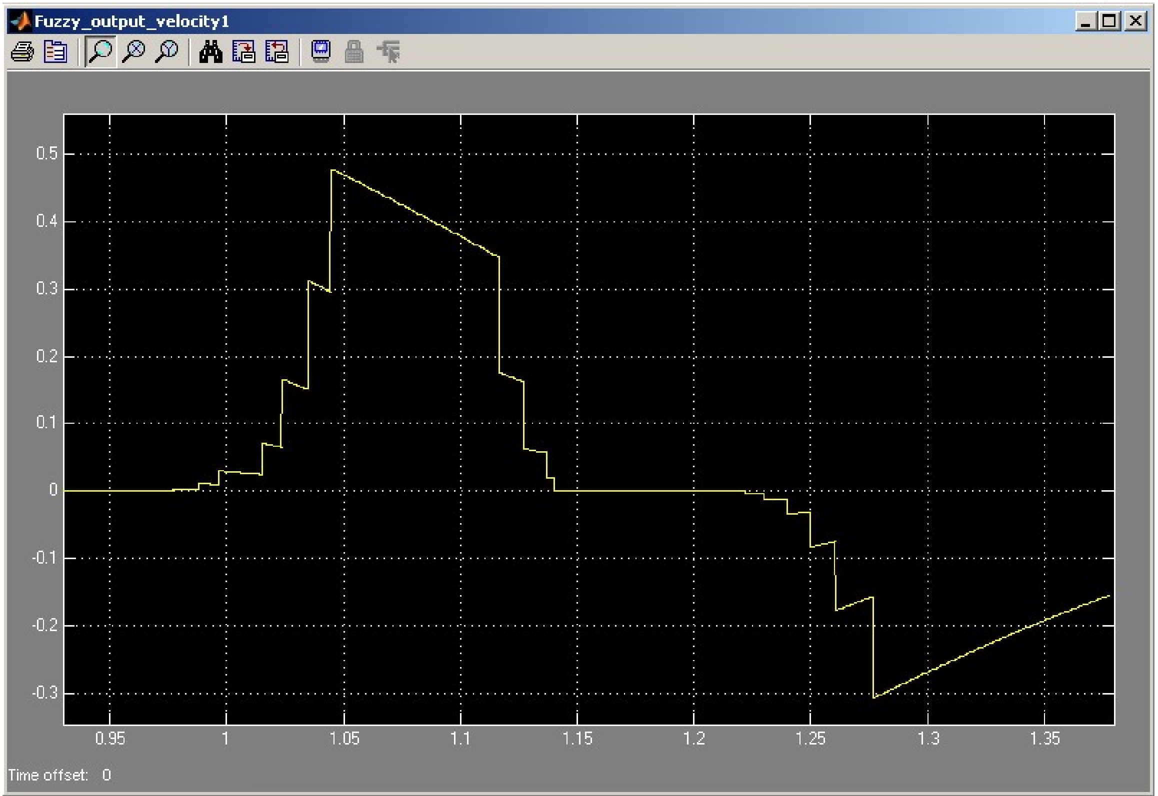


Fig. 3.23 Unfiltered output of the Fuzzy Control System during a sample use

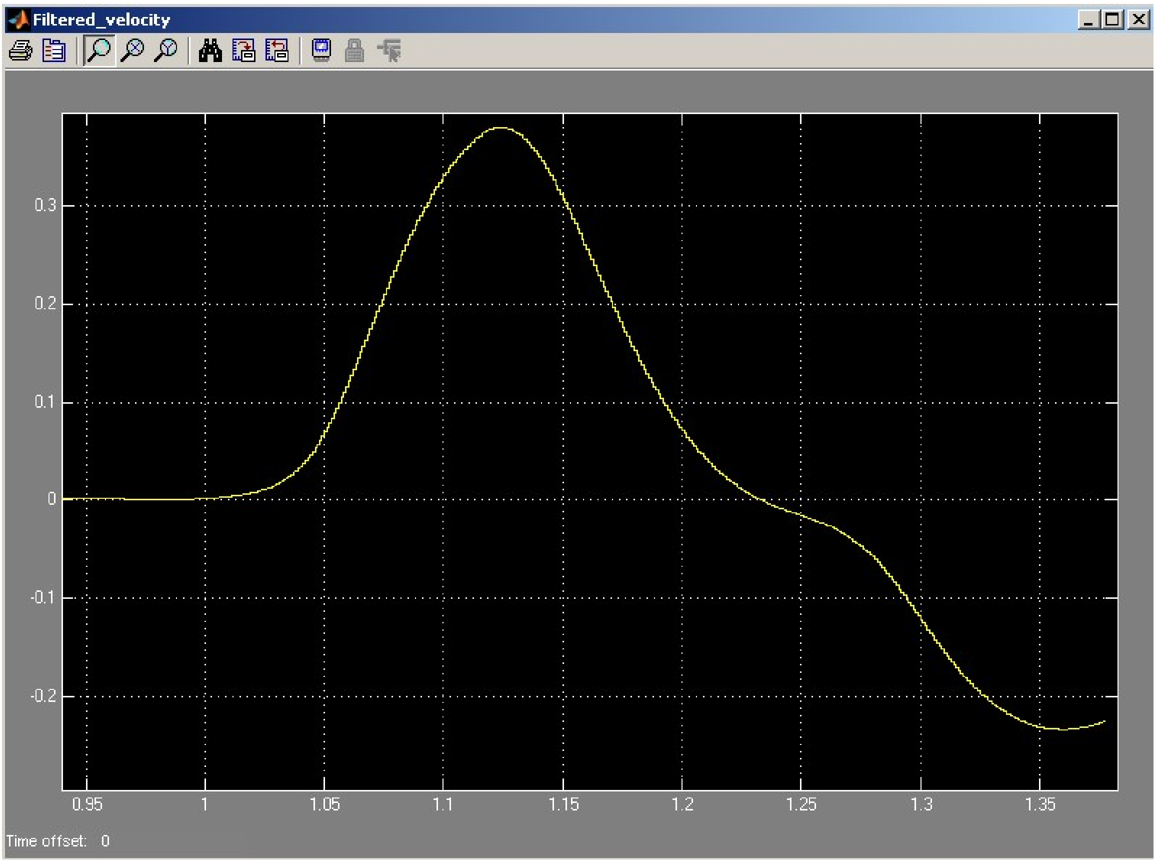


Fig. 3.24 Filtered output of the Fuzzy Control System during a sample use.

4 FMCC system hardware development

This chapter describes the development of the hardware module of the FMCCS. The hardware module (Fig. 4.1) consists of following components: flow-meter board, a bare-bone computer case used as chassis and supply box, A/D converter, biomedical disposable filter (saliva trap), connection tubes, gooseneck and air tube straw (disposable mouth straw).

Fig. 4.1 FMCCS Prototype

4.1 Flow-meter board

The key element of the flow meter board is Honeywell's microbridge mass airflow sensor AWM 2100V with bi-directional sensing capabilities. This cost-effective sensor (price about $80) provides precise, fast and repeatable low flow rate sensing. Its high performance (average 1.0 msec response time) makes the sensor applicable for high demanding requests, such as anesthesia control, in medical respiratory systems, etc. The mass airflow sensor operates on the principle of heat transfer: the more mass flow passes the sensor, the more heat is being transferred. A specially designed housing precisely directs mass airflow across the surface of a micromachined silicon sensing element. The sensor is integrated in a unique silicon chip based on advanced microstructure technology. It consists of a thin film, thermally isolated bridge structure containing heater and temperature sensing elements. Dual temperature sensing resistors, deposited in silicon thin film, are positioned on both sides of central heating elements in order to indicate flow direction as well as flow rate. Special technology ensures good thermal isolation for the heather and sensing thermal resistors. The small size and weight (approx 10.8 grams) as well as a good thermal isolation provide fast response and high sensitivity to flows. Figure 4.2 presents the output curves of the sensor within measuring range of ± 200 sccm (standard cubic centimeters per minute).

Two Wheatstone bridges control airflow measurements. The first circuit provides closed loop control of the temperature relative to the environment, while the second one is the sensing bridge. These two circuits, as well as output amplifier are not on the sensor board package and are realized based on design reference given in sensor data sheets. The design of the measuring board is sketched on (Fig. 4.3). The created board is presented in (Fig. 4.4). The board includes the following features: sensor power supply adjustment to 10V (by means of trimmer variable resistor), adjustments of output range (two standard outputs are possible 0-5 V and 0-10 V) with zero flow corresponding to the half of selected range and 0.5 V full scale flow safety margins ensuring that flow meter works in linear mode (no output saturation), offset compensation and measuring bridge balance adjustment. The prototype printed circuit board is etched and assembled manually and the adjustment procedures are performed using a digital voltmeter.

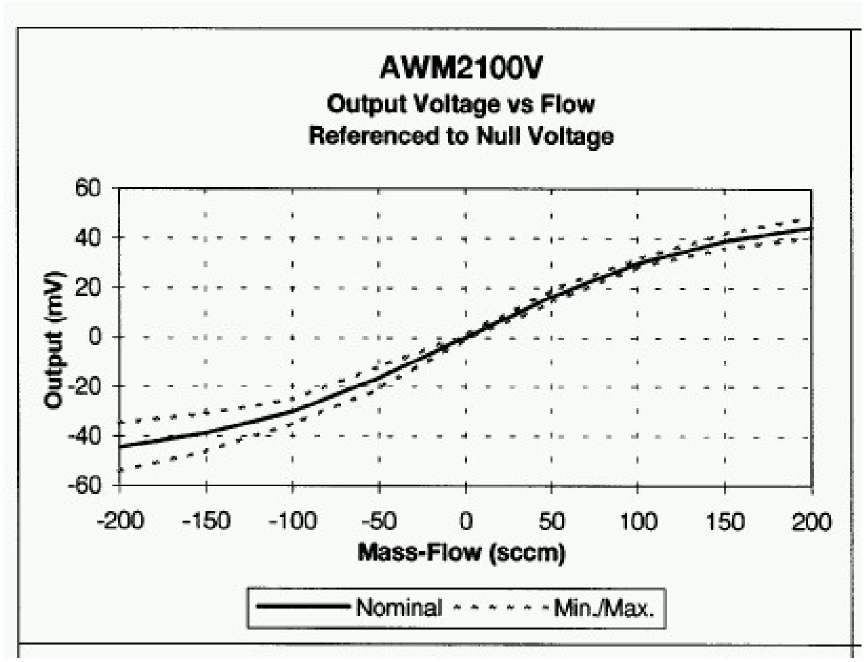


Fig. 4.2 Flowmeter characteristic curve

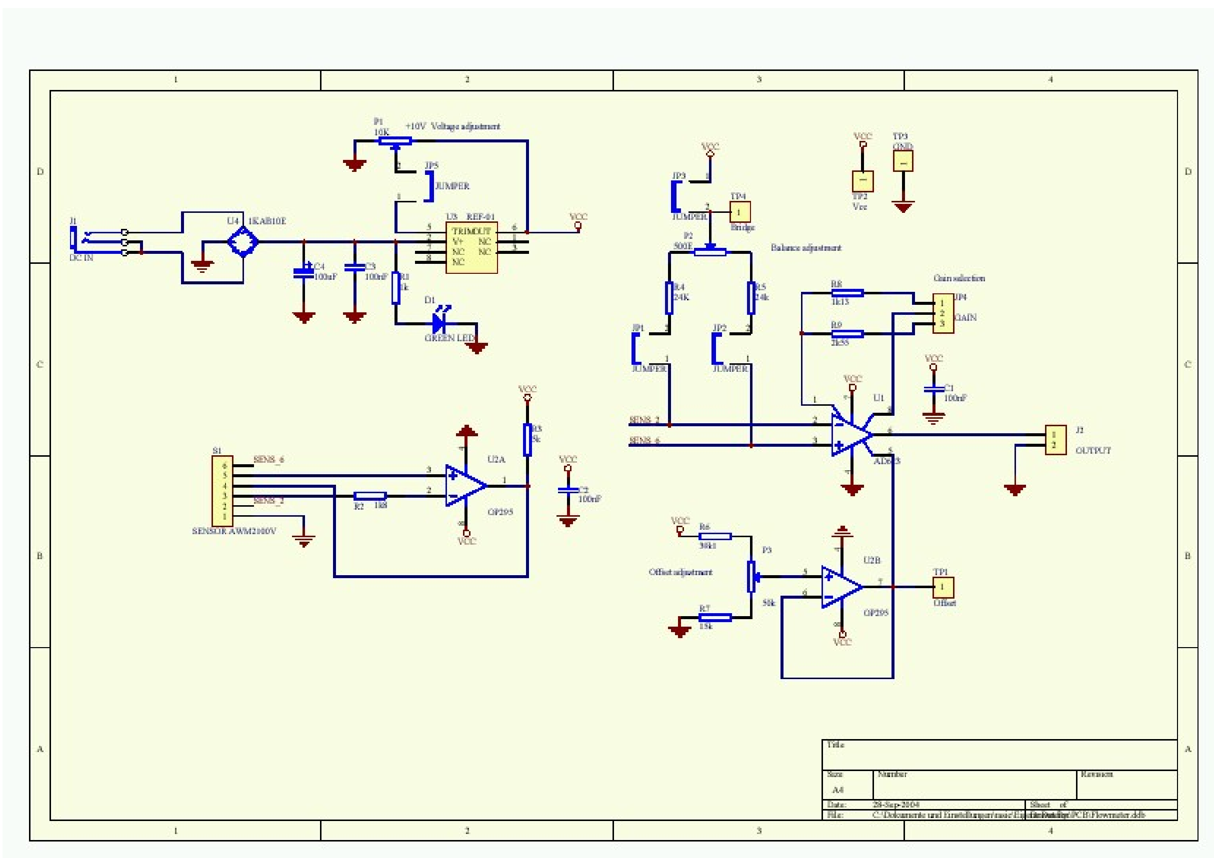


Fig. 4.3 Flowmeter circuits design

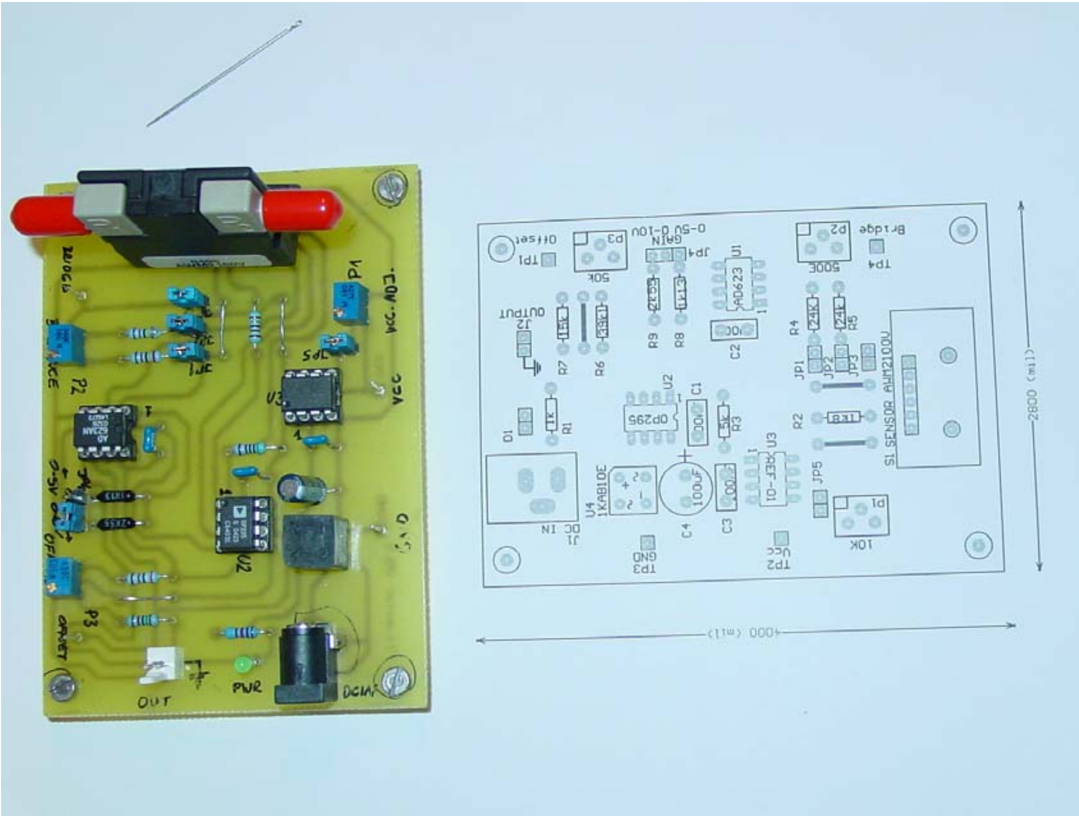


Fig. 4.4 Created circuit board

4.2 A/D conversion unit

As I/O computer measurement unit the Meilhaus PMD-1208FS USB data acquisition device has been applied (Fig. 4.5). This "personal measurement device" features eight analog inputs, two 12 bit analog outputs, 16 digital I/O connections, and one 32-bit external event counter. For initial testing of the FMCC prototype only A/D conversion of the flowmeter board has been utilized. The remaining channels provide a good base for future extensions and implementing a stand-alone product-oriented system. The plug-and-play module is powered by 5V USB supply from a computer. The module includes a Data Acquisition (DAQ) driver library and a basic measurement and testing software (SW) environment.

Fig. 4.5 Meilhaus PMD 1208FS

4.3 Prototype assembly and tests

The airflow sensing printed circuit board is integrated in a bare bone computer case providing 12V DC supply. On the front plate quick tube plug-in bulkhead connectors (for input and output flow) are installed and sensor is connected using silicon tubes (Fig. 4.6). In order to prevent contamination of sensing system a disposable medical filter (saliva trap) is mounted in the input flow branch (Fig. 4.7). Finally, the connection tubes with a disposable straw with a molded tip are added (Fig. 4.8). A versatile mounting system with a stiffened gooseneck (Fig. 4.1) is used in order to create an adjustable and stable positioning of the straw to the patient mouth.

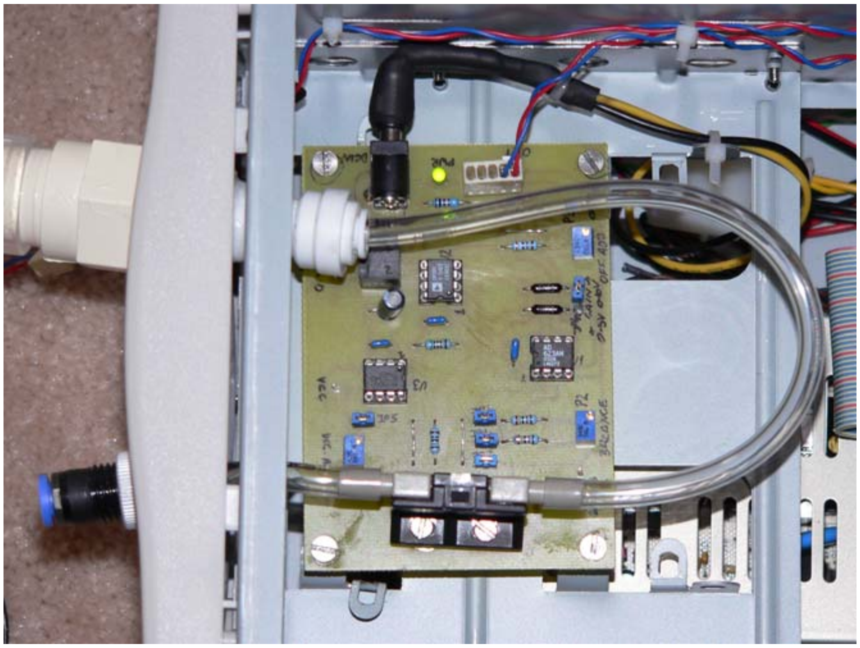


Fig. 4.6 Prototype airflow board mounted in computer case with plug-in front plate connectors

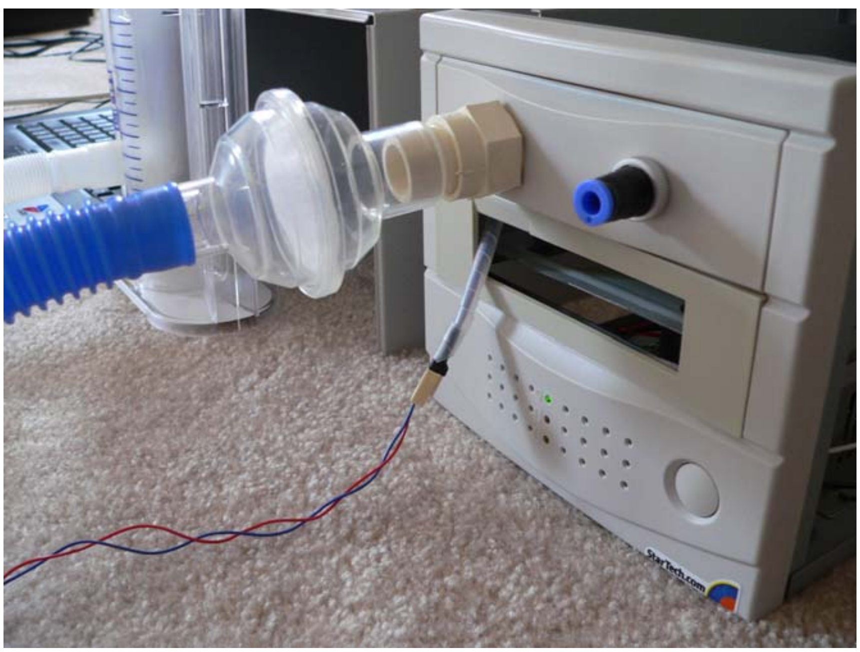


Fig. 4.7 Saliva trap biological filter (airflowmeter I/Os)

Fig. 4.8 Disposable silicon tube with mouth straw

The assembled system was tested using basic Meilhaus SW (InstaCal) providing data acquisition of raw output voltage data. This acquired data is visualized and analyzed in MATLAB. The analysis has validated proper sensor selection and correspondently expected good system performance for the foreseen cursor control application, such as: stable zero point with negligible drift, high sensitivity to the airflow by relatively small puff and sip, as well as full range output control with an amount of airflow produced without stress. A small sensitivity to mechanical tube displacements or pressings is resolved by stable fixation of tubes.

5 Cursor control implementation

5.1 Application Architecture and Design

The first FMCCS prototype is implemented in C, C++ and JAVA. The JAVA application has been used to simulate cursor control mechanisms described in the previous text. Figure 5.1 shows the high-level Architecture of FMCCS prototype.

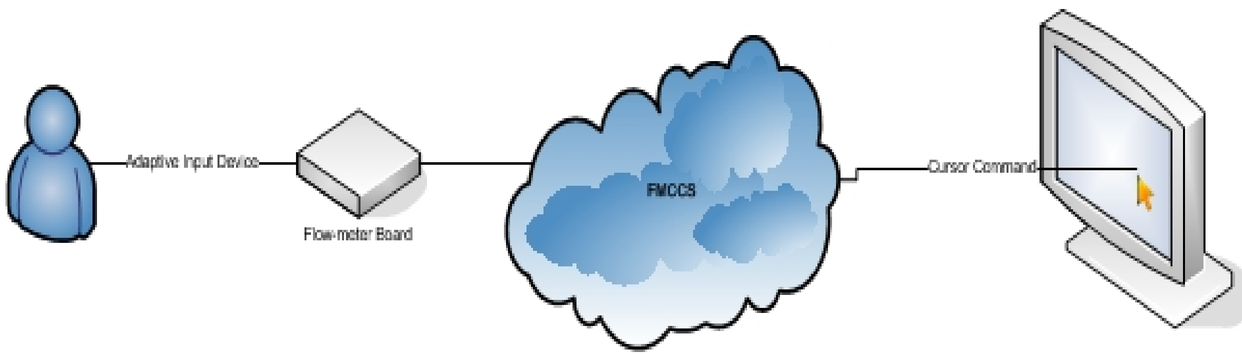


Fig. 5.1 High-level FMCCS architecture

The architecture as shown in Figure 5.1 consists of the user interacting with the system through the assistive input device. This device then sends signals to FMCCS. According to user inputs, the systems performs OS-related functions in order to realize different mouse functions on the computer screen.

As already mentioned above, internally FMCCS consists of two separate system layers, a Flow-meter Board data acquisition and processing module implemented in C/++, and the core cursor control system layer written in JAVA. Communication between the two systems is done via JNI (Java Native Interfaces). Figure 5.2 shows the two systems layers within FMCCS.

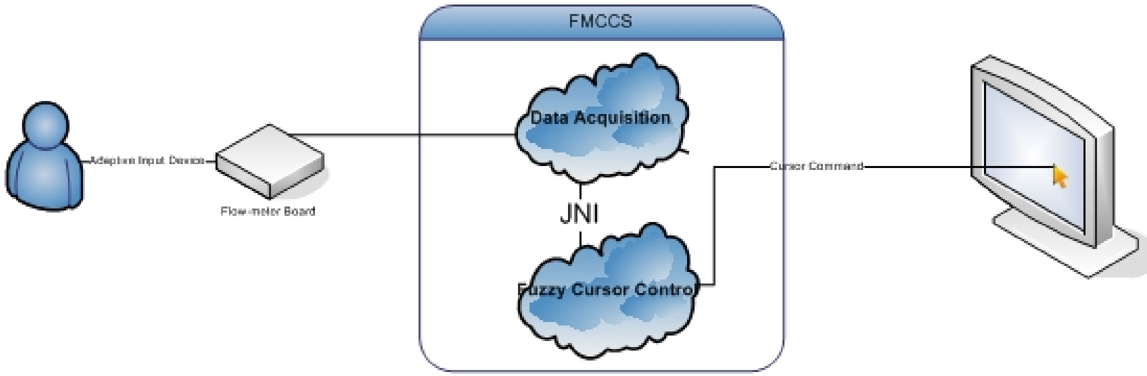


Fig. 5.2 System Layers

At system startup, the data acquisition layer is responsible to create the JNI communication bridge between Data Acquisition and the Fuzzy Cursor Control layers. The Data Acquisition layer, implemented as an finite data acquisition loop that acquires raw user input and makes it available to the rest of the system, uses A/D objects (provided by the Meilhouse Universal Library) to continuously acquire data from the Flow-Meter Board and asynchronously proceed them to the fuzzy cursor control layer. Raw data preprocessing is also implemented within this module.

Preprocessed input signals are passed to the fuzzy cursor control layer, and further fed to the Data Input FIS. This FIS transforms the raw data (input voltage in the range 0-9 V, with 4.5 V stable puff/sip zero point) to a crisp value in the range [-10, -10]. Figure 5.3 shows implemented Fuzzy Data Input FIS Membership functions.

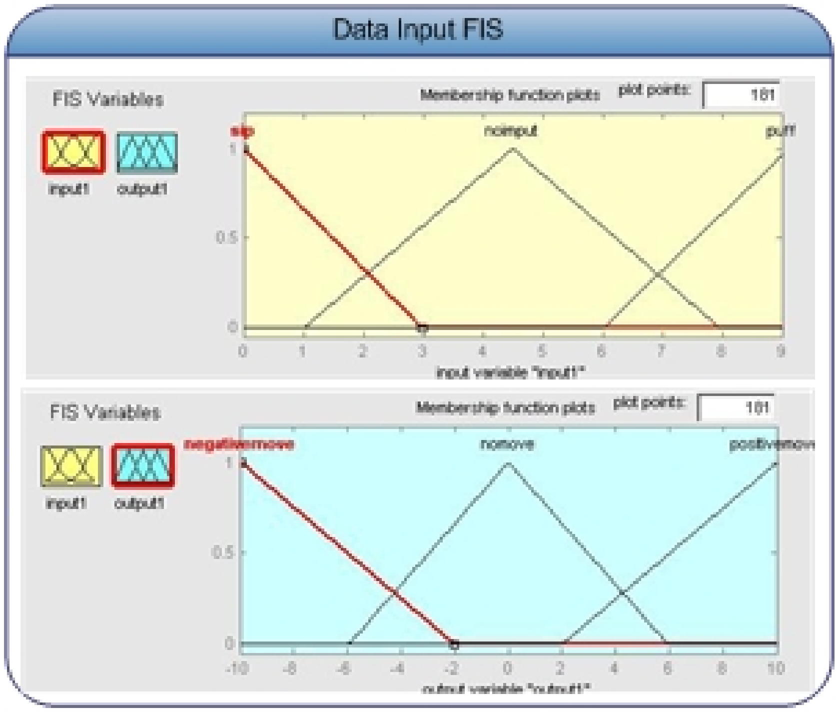


Fig. 5.3 Data Input FIS membership functions

All FISs within FMCCS are first created in MATLAB. Custom JAVA code was then written to read the corresponding saved MATLAB fis-file, describing function parameters, in order to create a programmatic representation of the inference functions. Currently this code only supports triangular and trapezoidal membership functions, but the remaining functions are under implementation. Depending on severity of motor deficiencies, all FISs within FMCCS can easily be adjusted to individual users of the systems by graphically changing them in MATLAB and generating new cross reference data files. No coding changes are necessary. The FIS creation process is illustrated in Fig. 5.4

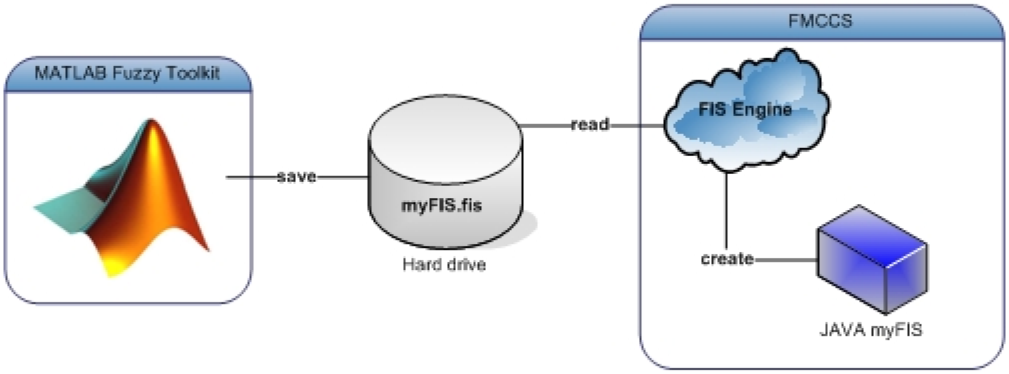


Fig. 5.4 FIS creation process

As an example, the communication with the myFIS Object created within FMCCS can be described with following JAVA Code snippet:

```
FIS myFIS = new FIS("<PATH TO myFIS.fis>");
double defuzzAnswer = myFIS.evaluate(rawinputData);
```

All input data that goes through the Data Input FIS is stored in the application input Queue. Because the Flow-meter Board's sampling rate of 1kHz, storage and acquisition of data to/from this input Queue had to be synchronized to prevent user data from being lost during the data acquisition process. Data stored in the Queue is then used by the system to create Morse Code patterns describing user commands, as well as input to other fuzzy inference systems within FMCCS responsible for cursor rotation and movement. A special "Idle" command can also be extracted from the input Queue, and describes a defined period of time during which the users did not give any input to the system.

5.2 State Flow engine

A very important part of FMCCS is its internal state-flow engine. It defines eight unique system states, namely the Idle, Mouse Movement, Mouse Rotation, Single Click, Double Click, Right Click, Drag Start, and Drag End states. Initially, the system enters the Idle state. Depending on the Morse Code command extracted from user input, the system can switch states. Once entering a state, the system stays in that state until the users gives a new command to transit to a different state, or the Data Input Queue issues the "Idle" command, in which the system sets itself back to the initial state. Upon each state transition or change, the user is notified of the change by means of the FMCCS synthetic speech engine (based of FreeTTS). The state-flow engine diagram implemented in the first FMCCS prototype is presented in Fig. 5.5.

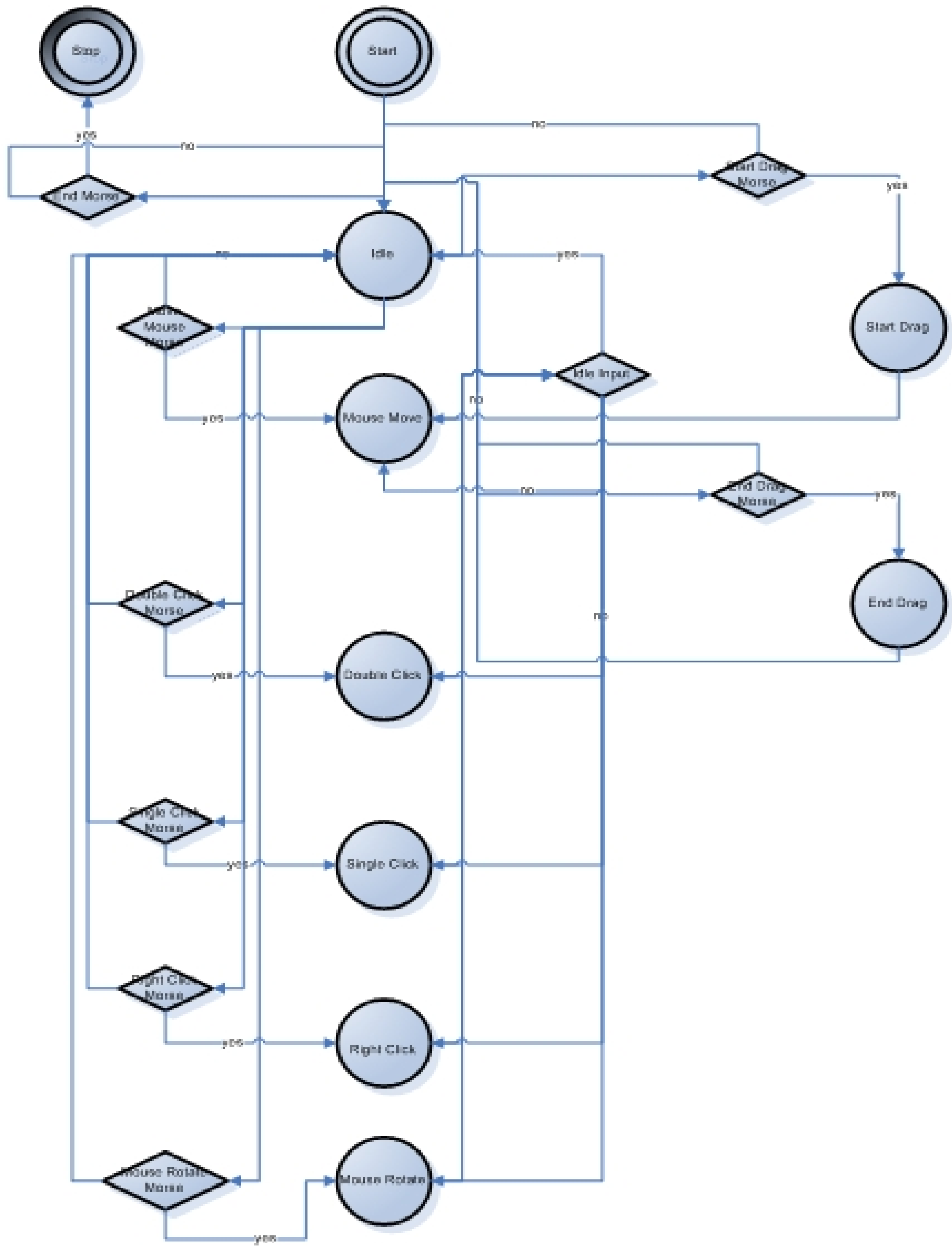


Fig. 5.5 State-flow Engine diagram

FMCCS uses the Morse Code patterns presented in Table 5.1 for state transitions, where the Puff operation is represented by a dash (-) and the Sip operation is represented by a dot (.).

State	**Morse Code**
Mouse Rotation	-.-
Single Click	-..
Double Click	…
Right Click	..-
Drag End	--.
Drag Start	.--
Mouse Movement	---
Idle	none

Table 5.1 Morse code patterns

Each State, if active, defines specific cursor and voice commands within the system. Each cursor command is accompanied with its corresponding voice command confirmation, as well as next-action and actual state information. FMCCS uses synthetic speech processing to guide the user through system use. Voice guidance can be turned off and on upon user preference. Table 5.2 presents these voice feedback confirmation and following-on commands.

5.3 User Interface and Functionality

FMCCS is implemented to run under two Operation Modes: The first one is Training mode intended to let the user practice with entering Morse Code commands. In Training Mode, the user is asked to enter Morse Code commands that correspond to state changes within the system. The user is asked to repeat a certain code if it was entered incorrectly. A Morse Code Legend is displayed in bottom-right corner of the screen that can be used as a guide.

In Mouse Control Mode, user can apply learned commands to freely move the cursor on the screen to perform different tasks. The screen shots presented in Fig. 5.6 and Fig. 5.7

show the system user interface during Training Mode and Operational Mode, respectively.

State	Voice Communication
Mouse Rotation	1) Notify user Mouse Rotation state is entered 2) Change Cursor Icon to “Double-Arrow” 3) Draw Cursor Scan-Line extension on screen 4) Rotate cursor and Scan-Line according to user input.
Single Click	1) Simulate OS Mouse left button click 2) Notify user single click was performed
Double Click	1) Simulate OS Mouse double left button click 2) Notify user double click was performed
Right Click	1) Simulate OS Mouse Right button click 2) Notify user right click was performed
Drag End	1) Simulate OS Mouse Right button release 2) Notify user drag operation ended
Drag Start	1) Simulate OS Mouse Right button hold 2) Notify user drag operation started
Mouse Movement	Move Cursor along Scan-Line depending on user input.
Idle State	1) Change Cursor Icon to “Default Arrow” 2) Notify user to enter new command

Table 5.2 Voice command user support

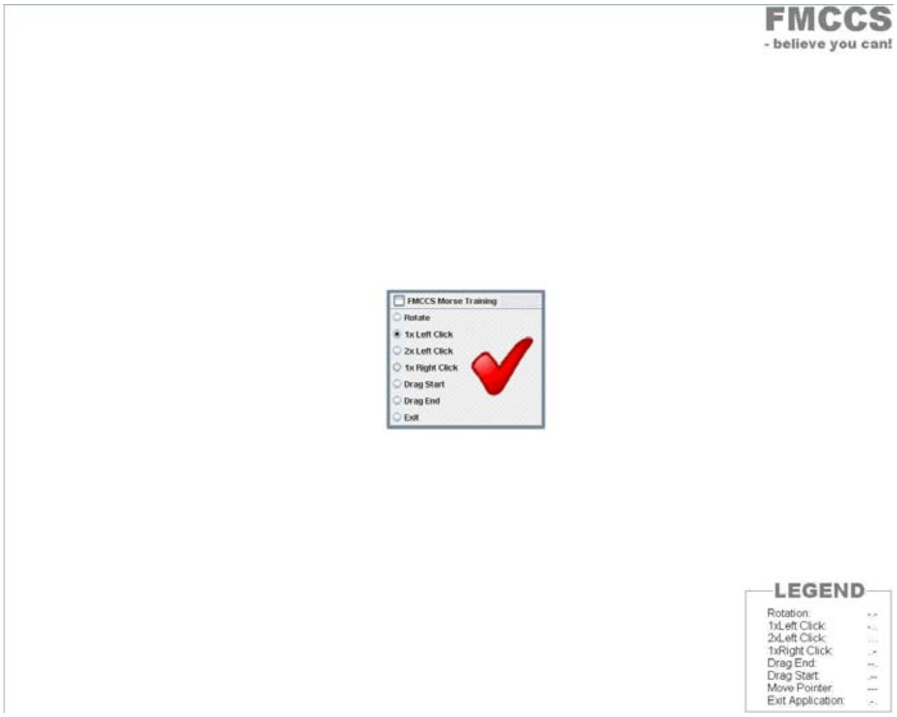


Fig. 5.6 FMCCS Training Mode Screen

Figure 5.6 shows the FMCCS Training Mode screen. This screen includes a window which presents the users with Morse Code commands she/he needs to execute. A Legend containing all Morse Code patterns is displayed in the bottom-right corner of the screen. Users' task is to perform the given Morse Code pattern. If successful, user is asked to perform the next pattern in the list. In case of failure, the user is asked to repeat the particular pattern.

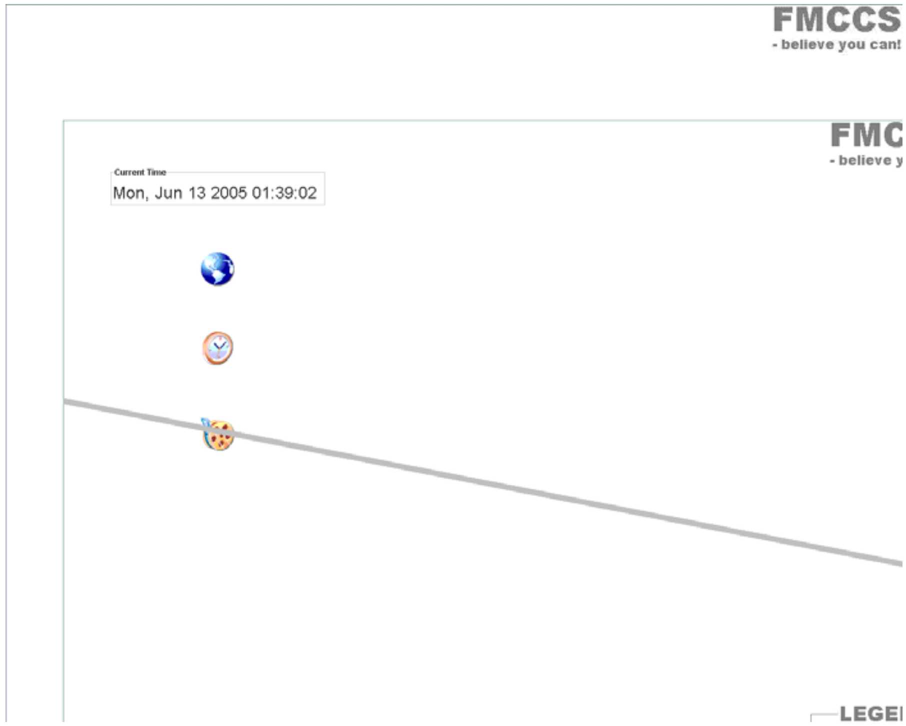


Fig. 5.7 Operational Mode Screen

Figure 5.7 shows the Operational Mode FMCCS screen. Within this screen, the user is able to enter learned Morse Code commands to perform various mouse functions described. Three buttons on the left side of the screen allow the user to play a drag/drop game, turn the current time presentation on and off, as well as show a screen which includes the latest movies played in theaters. A Morse Code input Legend is also presented in the lower-right corner of the screen, helping the user in remembering the different input commands.

6 FMCCS initial experiments

Two people were used for initial tests with the FMCCS prototype: a man age 30 (user A) and a man age 32 (user B), both quadriplegics with cervical region injury, were the first FMCCS users in initial tests. User A has relatively satisfactory diaphragm control to allow unassisted breathing, while the user B (Fig. 6.1) requires mechanical assistance to breathe. Both users are well experienced with puff-sip AT devices for controlling wheel chairs, as well as operate computers and several environmental objects (i.e. telephone).

Fig. 6.1 FMCCS prototype testing with user B

6.1 Experiments Protocol

Experiments with user A, and B were divided into two sections, representing the two modes of FMCCS (Training Mode, and Operational Mode).

6.1.1 Training Mode

During FMCCS operation, users sat facing a laptop screen. The trials started with the Training Mode. Subject A had little difficulties operating the system under Training Mode and, similarly to healthy persons, learned quickly to utilize simple Morse codes for mouse operation commands.

However, subject B had significantly more difficulties and spent higher efforts operating the system, especially to repeat multiple sip sequences. For him the actual tuning of FIS functions was too sensitive, mainly since assisted breathing produced a higher intensity of input air flow causing the selected sensor to operate close to saturation (by puffing). In addition his sip sequences were very short, immediately followed by uncontrolled puff sequence, which produced complexity to realize specific Morse codes.

6.1.2 Operation Mode

In Mouse Control Mode the goal was to operate the mouse pointer in the implemented fuzzy circular scanning mode, to point it to selected icons on the screen, to select one and drag/drop it to other given position. More sophisticated operations involved a puzzle-like game that required multiple drag/drop screen operations.

Again user A performed all these task easily and quickly, after a small amount of training he needed about 2 seconds to point the mouse to different targets. Patient A demonstrated significant benefits of using novel velocity control features, as well as controlled direction and fine velocity settings. Within these tests with user A, the FMCCS demonstrated considerable improvements compared with conventional mouse cursor control systems. User A was very impressed by the velocity control features, but voiced out that the user interface would need to be improved to allow more tasks to be performed, such as email, word processing as well as internet browsing capabilities.

However, patient B continued having difficulties in controlling the cursor in the Mouse Control Mode. To facilitate his needs, the initial Morse code commands for the cursor circular and linear scan in screen polar coordinates have be changed and coded using puff sequences only (Fig. 6.2). After these changes, user B succeeded to generate commands and control the mouse pointer. However, he did this at a high velocity due to increased assisted airflow input. He was not able to get experience with the FMCCS novel features, since he was not capable of realizing and using them. Several trials to adjust data input and control FIS did not provided any major improvements, while the sensor worked close to saturation point. Obviously another sensor providing higher flow rate would be unconditionally needed to allow a more efficient operation to user B. User B was also not able to control the mouse for a relatively longer distance in the sip direction, due to his ability to only realize very short sip durations.

Fig. 6.2 Adjustment of Morse Code and FIS parameters in order to support user B operations

6.2 Discussion

Obtained initial results show that people with severe disabilities experienced with puff-sip switch systems can rapidly learn to operate the FMCCS. Their control within the system develops gradually over the use of training and operation system modes. Their ability to adapt to fuzzy control functions has been demonstrated to be very useful. However, to utilize improved mouse control functions, especially in fine control manipulation, relatively good breathing control is required. Considering mouse positioning efficiency and good responsiveness, especially in fine control applications, the FMCCS demonstrated a considerable step forward in comparison to state of the art switch-based mouse control systems.

For users with assisted breathing, however, a more appropriated sensor must be applied, as well as more sophisticated and complicated control algorithms, reflecting specific breathing abilities, must be implemented in order to benefit from the novel FMCCS functions. This, as well further testing and improvements, represent short-term goals of future investigations.

7 Conclusion and future work

The developed FMCCS provides a feasible and promising solution for development of sophisticated and adaptive HCI systems that can be adjusted to specific user's abilities and needs. The implemented system also provides a good platform for further development and improvements.

System usage results suggest that people with severe motor disabilities could use FMCCS with ease and very effectively. This encourages further improvements of the FMCCS as well as other applications of intelligent soft computing-based AT systems.

There are many possible techniques, which can improve the current FMCCS implementation. These include:

- Creation of a hybrid user-input interpretation method based on linguistic variables and Morse code sequences. This hybrid interpretation scheme could be very useful in the state-transition process of the FCCS, as well as for performing more advanced clicking operations such as drag/drop.
- Inclusion of Neural Network techniques in scanning procedure. The system could benefit from learning about user actions in different parts of the selection process. The FMCCS could use neural networks to learn about the most commonly used object on the screen, and make prediction-based selection during the scanning process. This technique could improve the efficiency and performance of the overall selection process within in FCCS.
- Use of Genetic Algorithms for optimization of fuzzy control functions and the FMCCS Stateflow. It would be very helpful to optimize cursor control, allowing for more efficient movement and object selection. Also, given different possible implementations of the scanning processes within the FMCCS, GA can be used to optimize the stateflow based on certain user needs. Through the execution of the GA, different, possibly new, scanning

techniques can be determined, that give the user more transparent access to the system.

- User Knowledge Discovery through usage logs. Extraction of intelligent information from user system-usage logs can be very beneficial in creating a more efficient system. The extraction of associative rules from usage logs could give ideas on creation of new fuzzy scanning procedures, or indicate certain flaws within the system.
- Implementation of more efficient fuzzy control functions and their corresponding fuzzy inference systems.
- Improvement of fuzzy scanning procedures through user feedback and testing.
- More robust and complete system implementation in Matlab and Windows.

Some of these topics will be addressed in future research efforts.

8 Bibliography

Betke M., Gips J. and Flemeing P, “The camera Mouse: Visual Tracking of Body Features to Provide Computer acess for People With Severe Disabilities”, IEEE Transaction on Neural Systems and Rehabilitation Engineering, Vol. 10, No.1, March 2002, pp. 1-10.

Buxton W., “Human Interface Design And The Handicapped User”, Proceedings ACM CHI'86 Conference, New York, 1986, pp. 291-297.

Chen Y. and Newman W.S., “A Human-Robot Interface Based on Electro0culagraphy”, Proceedings of the 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA, April 2004, pp. 243-248.

Chen Y.L., Tang F.T., Wong M.K., Shih Y.Y. and Kuo S.T., “The New Design of Infrared-Controlled Human-Computer Interface for the Disabled”, IEEE Transaction on Neural Systems and Rehabilitation Engineering, Vol. 7, December 1999, pp. 474-481.

Cleveland D., “Method and Apparatus for Locating Image Features”, U.S. Patent 5,231,674, 1993.

Fabiani G., G.E., McFarland J.R., Wolpaw J.D., and Pfurtscheller G., “Conversion of EEG Activity Into Cursor Movement by a Brain-Computer Interface (BC)”, IEEE Transaction on Neural Systems and Rehabilitation Engineering, Vol. 12, No. 3, September 2004, pp. 331-338.

Hwang F., “A Study of Cursor Trajectories of Motion-Impaired Users” in Proc. ACM CHI 2002, Extended Abstracts, Minneapolis, Minnesota, 2002, pp. 842-843.

Hwang F., “Partitioning Cursor Movements in Point and Click Tasks” In Extended ACM CHI’2003 Abstracts, Fort Lauderdale, Florida, 2003, pp. 682-683.

Jang J-S.R, Sun C-T., Mizutani E., Neuro-Fuzzy and Soft Computing : A Computational Approach to Learning and Machine Intelligence, Matlab Curiculum Series, Prentice Hall, NJ, 1997.

Keates, S., Hwang F., Langdon, P., Clarkson, P. J., and Robinson, P., “Cursor measures for motion-impaired computer users. Proceedings of ACM SIGCAPH Conference of Assistive Technologies – ASSETS 2002, 135-142.

Leuthardt E. C., Shalk G., Wolpaw J.R., Ojemann J.G. and Moran D. W., “ A Brain-Computer Interface Using Electrocorticographic Signals in Humans”, Journal of Neural Engineering, No. 1, 2004, pp. 63-71.

LoPresti E.F., Brienza D.M., Angelo J., and Gilbertson, “Neck Range of Motion and Use of Computer Head Control”, Journal of Rehabilitation Research and Development, Vol 40, N0. 3, May/June 2003, pp. 199-212.

Madison K., Neil Wasylewski N. and Ruthenberg B., “Powered Easel”, Proceedings RESNA Annual Conference 1996, Salt Lake City, Utah.

Marler D.M., “Cyberlink - Computer Access for Persons Identified With Multiple Disabilities”, Ms.A. Thesis, California State University, Northridge, 2004, (http://www.icap.nu/pdf/cl_uppsats.pdf)

Mason K.A., “Control Apparatus Sensitive to Eye Movement”, U.S.Patent 3,462,604,1969.

Millan JdR., Renkens F., Mourino J., Gerstner W., „Non-Invasive Brain-Actuated Control of a Mobile Robot“, Proc. Of the 18th Joint International Conference on Artificial Intelligence, 2003, Acapulco, Mexico, (Article available at ftp://ftp.idiap.ch/pub/reports/2003/millan_2003_ijcai.pdf).

NSCI Statistical Center, “Spinal Cord Injuries : Facts and Figures at a Glance”, Publication of NSCI-SC Birmingham, Alabama, August 2004 (www.spinalcord.uab.edu).

Ohno T. and Mukawa N., “Gaze Based Interaction for Anyone, Anytime”, Proceedings of International Conference on Human Computer Interfaces, 2003, pp. 1452-1456, (online availably from http://www.brl.ntt.co.jp/people/takehiko/papers/hcii2003.pdf).

Ohno T., Mukawa N., and Kawato S., "Just blink your eyes: A head-free gaze tracking system ", *Extended abstract of the ACM Conference on Human Factors in Computing Systems, CHI2003,* Apr 2003, pp.950-951.

Ortega R., “Unusal Acess Methods”, Proceedings CSUN's 19th Annual International Conference "Technology and Persons with Disabilities", Los Angeles, 2004, (online available on http://www.csun.edu/cod/conf/2004/proceedings/143.htm)

Polikoff J.B., Bunell H.T., and Borkowski W.J., “Toward a P-300-based Computer Interface”, Proc. Rehabilitation Eng. And Assistive Technology Society of North America (RESNA’95), Arlington, 1995, pp. 178-180.

Price L.A. and Cordova C. A., “Use of Mouse Buttons”, Proceedings of the ACM CHI Conference on Human Factors in Computing Systems, December 12-15, Boston, Massachusetts, 1983, pp.262-266

Seabrock R.H.C., The Brain Computer Interfaces : Techniques for Controlling Machines, Technical Report, University of Maryland, 1994, Available from http://www.lclark.edu/~alec/finalproject/dbi/dbi.html .

Serruya M.D., Hatsopoulos N.G., Paninski L., Fellows M.R., Donoghue J.P., "Instant Neural Control of a Movement Signal", Nature, Vol. 416, March 2002, pp. 141-142.

Shi S-W., Wu Y-T. and Liu J., "A calibration-free gaze tracking technique", Proceedings of 15th International Conference on Pattern Recognition, Spain, 2000, Vol. 4, pp.201-204.

Walpaw J.R., Vaughan T.M., "Brain-Computer Interface Technology: A Review of the First International meeting", IEEE Transactions on Rehabilitation Engineering, Vol. 8, No. 2, June 2000.

Walpaw J.R., MCFarland D.J., Neat G.W., and Forneris C.A., "An EEG-based Brain-Computer Interface for Cursor Control", Electroencephalography and Clinical Neuropshysiology, Vol. 78, 1991, pp. 252-259.

Walpaw JR, McFarland DJ., "Control of a Two-Dimensional Movement Signal by a Noninvasive Brain-Computer Interface in Humans", Proceedings of National Academy of Sciences, 2004, Dec 21, Vol. 101(51), pp. 17849-54.

Yoo D.H. and Chung M.J., "Eye-Mouse Under Large Head Movement for Human-Computer Interface", Proceedings of the 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA, April 2004, pp. 237-242.

Zhu Z. and Ji Q., "Eye and Gaze Tracking for Interactive Graphic Display", Machine Vision and Applications, Vol. 15, 2004, pp. 139-148.

Janki Vora, "*A P300 Signal Controlled Robotic Arm Movement Brain Computer Interface*", 2004, Master's Thesis, Georgia State University