DEVELOPMENT OF AN ELASTIC PATH CONTROLLER FOR COLLABORATIVE ROBOT

LONG BO

(B.Eng, Huazhong University of Science and Technology)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
SINGAPORE
2005

Acknowledgements

I would like to thank Dr. Teo Chee leong and Dr. Etienne Burdet, my supervisors, for their many valuable suggestions and constant support during this research.

Suggestions from Dr. Yu Haoyong made this work moving forward faster.

My friend and collaborator, Rebsamen Brice, who gave me a lot of nice suggestions on the programming, and helped me to overcome my laziness and encouraged me to pursuit a higher degree.

And many thanks to Dr. J.Edward Colgate and Dr. Michael Peshkin, their kindness and warm heart made us possible to test the elastic path controller on the Scooter cobot in Laboratory for Intelligent Mechanical Systems(LIMS). Eric Faulring gave me selfless help during my stay at the LIMS.

I dedicate this thesis to my parents, you gave me *love* when I felt lost in the life. You are the only reason why I gonna be better.

Without the help of the people mentioned above, this work would never have come into existence.

Finally, I wish to thank the following: He Cong (for her encouragement, pressure and porridge when I got sick); Hu Jiayi (for changing my life from worse to bad); Wang Fei, Liu Zheng, Ganesh Gowrishankar, Ankur Dhanik (for all the good and bad times we had together).

Table of Contents

Summary

This thesis describes the development of an elastic path controller for assistive robotic devices. This controller combines the functionalities of path tracking and modification of trajectory. It is able to compensate for changes in the environment such as when there is a new obstacle or there are errors in position sensing. The controller is tested on two such devices: the Cobot (**Co**llaborative Ro**bot**) invented in the Laboratory for Intelligent Mechanical Systems (LIMS), Northwestern University and a Collaborative Wheelchair Assistant developed in the Control and Mechatronics Laboratory, National University of Singapore.

Cobots are robotic devices intended for direct interaction with a human worker. It is passive, i.e. it will not move without power provided by the user, and is thus intrinsically safe. It potentially is well-suited to safety-critical tasks such as computer-assisted surgery, or to tasks where conventional robots would be too dangerous for direct contact with a person, such as automobile assembly [1]. Cobots operate in two modes: a "free mode" and a "guided mode". In free mode, the cobot is free to move without constraints while in guided mode, the cobots are constrained to move along pre-defined paths to facilitate maneuvering. Cobots implement these pre-defined paths via software.

The Collaborative Wheelchair Assistant (CWA) is another assistive device designed to give the user freedom of movement. Users can decide when, where and how he/she wants to move according to his/her needs and users operate the wheelchair in a collaborative fashion. It is based on a commercial wheelchair with minimal extra sensors added. The CWA also implements "software-defined" path constraints (similar to the cobots) to facilitate operation of the wheelchair.

To realize a more effective collaboration between the user and the assistive robotic devices,

we design an elastic path controller for the devices. As the name "elastic path controller" implies, it gives users more freedom when they work with the robotic devices. The elastic path controller not only supplies a "guiding" path, it also let the users have the autonomy to modify this guiding path dynamically. In this way, the elastic path controller integrates the functions of path following and obstacle avoidance. The system makes use of the inference ability of humans to complete the obstacle avoidance task easily without the need for expensive obstacle detectors. When the device is working in the constraint mode, it will follow the pre-defined guiding path. If the user sees the obstacle along the path, he/she can decide to activate the elastic mode to avoid the possible collision.

The elastic path controller is tested in simulations on the CWA and Scooter cobot and implemented on the Scooter cobot at LIMS, Northwestern University. The simulations are done in the simulation environment written in MATLAB. The experiments on the Scooter cobot demonstrated that users can learn to use this novel tool in order to modify and design guiding paths in a relatively simple way. The results also suggest that the users may feel the attraction from the guiding path which help them to maneuver the cobot.

List of Figures

List of Tables

Chapter 1

Introduction

COBOT (for **Co**llaborative Ro**bot**) was invented by Edward Colgate and Michael Peshkin from Northwestern University. "Cobots" are intended for direct interaction with a human worker, handling a shared payload [3]. Cobot is a device activated by operator's movement such as pushing or pulling(Figure 1.1). It is passive, i.e. will not move without power provided by the user, and is thus intrinsically safe.

Figure 1.1. Industrial prototype of the Scooter cobot used at General Motors [6]

Cobots interact with people by producing software-defined "virtual surfaces" which constrain and guide the motion of the shared payload. Ergonomic as well as productivity benefits

result from combining the strength of the cobot with the sensing and dexterity of the human worker [3]. Cobot can follow a pre-defined path stored in the outfitted computer using a "Path following" control. Path following drives an object along a geometric path without a timing law assigned to it. Path following is a useful motion control approach when maneuvering mobile robots from one area to another[4]. Since cobot depends less on the sensors or other localization devices from which the moving error usually comes, it is supposed to complete the task with a higher efficiency and accuracy.

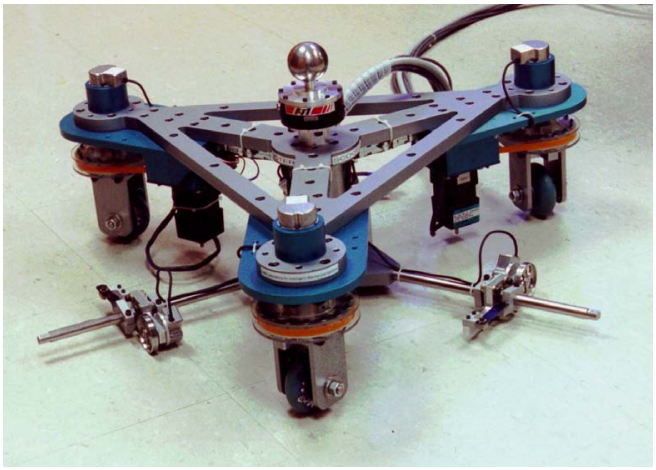


Figure 1.2. Scooter cobot at LIMS, Northwestern University

The Scooter cobot on which experiments have been performed for this thesis is a mobile platform moving in the two-dimensional plane: (x, y, θ)(Figure 1.2). This prototype was conceived to facilitate the placement/removing of car doors in the assembly line [5].

Cobots have two basic motion modes: free mode (FM) and guided mode (GM). In free mode, the cobot behaves like a chair with casters. In guided mode it is constrained along a virtual guideway which is defined in software. Eng Seng et al. could show in experiments [14] that less effort is required to move in guided than in free motion. Further, movements in GM are faster, smoother, and require less back and forth correction than in FM. Simple

and efficient methods to define ergonomic guiding paths were also developed in [10].

A problem arises with guided motion when an obstacle or a person is standing on the guiding path, the obstacle had to be removed before the cobot can proceed on. A solution to this problem may be to provide the operator means to avoid obstacles. In common mobile robotics, obstacle avoidance is achieved by using sensors and heavy sensor processing to detect the obstacles, and by modifying the path planning correspondingly. However, cobots work with a human operator who is equipped with natural sensors and powerful sensor processing, in particular vision. Our idea is thus to provide the operator an Elastic Path Controller with which he or she can avoid obstacles, by pushing the cobot when he or she detects the obstacle.

We envision that with this Elastic Path Controller (EPC) the cobot can follow the predefined guiding path when no obstacle is detected and the operator wants to keep moving. The EPC enables the operator to deform the guiding path when the obstacle is detected, and bring the cobot back to the guiding path when the obstacle is passed. Using it, the user will be able to go through narrow passages, what may be difficult and even dangerous with autonomous navigation systems if odometry is not sufficiently accurate. The user can use his own judgement to perform necessary correction during movement.

A collaborative wheelchair is developed at NUS[9], which uses virtual guideways to help disabled to maneuver their wheelchair according to their needs. Our Collaborative Wheelchair Assistant (CWA) was built on a commercial wheelchair YAMAHA JW-1 (Figure 1.3). Previous attempts with robotic wheelchairs have shown that disabled are generally not satisfied with fully autonomous wheelchairs. Despite heavy computation to recognize the environment and perform motion planning, an automatic wheelchair frustrates the disabled from their freedom to control the motion, to stop for observing something or to chat with a friend.

This thesis develops such an Elastic Path Controller and tests it in simulations and in

Figure 1.3. Collaborative Wheelchair Assistant at the National University of Singapore

experiments performed on the Scooter cobot(Figure 1.2) and the Collaborative Wheelchair Assistant. The CWA[9] has an unicycle-type kinematics. The Scooter is a triangular vehicle moving on a plane, with a steerable wheel at each corner. However for simplicity a two-steering-type vehicle kinematics model was adopted to control two of three steering wheels in our experiments, and the third wheel was controlled to go through the intersection formed by axes of two steering wheels.

Simulations have been performed to develop and test the EPC. Unicycle and two-steering-wheels type kinematics corresponding to the CWA and Scooter were considered. The simulation environment consisted of a joystick connected to a computer with graphical user interface controlled by a MATLAB program. Several controllers were tested for each kinematics model, including the feedback linearization based controller proposed by Samson [15, 19, 17] and a nonlinear Lyapunov-oriented controller from Micaelli and Samson [16, 18] in 1993. These controllers were adapted to realize the elastic characteristic.

Experiments have been performed on the Scooter to investigate the performance of the elastic path controller, using the feedback linearization based version. One experiment

investigated whether and how the users can train the cobot to avoid obstacles using the EPC, and examined its efficiency and accuracy. Another experiment investigated which strategies the users use to work with the EPC. The results suggest that the EPC is easy to learn and an efficient mean of modifying the desired path for collaborative robots.

The kinematics and the elastic path planners for the CWA and Scooter cobot are described in chapters 2 and 4. Simulation of the elastic path controller are presented in Chapter 5. Chapter 6 presents experiments performed on the Scooter Cobot to investigate performance with the Elastic Path Planner. Conclusions and suggestions for further research are given in Chapter 7.

Chapter 2

The Collaborative Wheelchair Assistant

In this chapter, a review of past research projects about robotics wheelchair which have been applied to assist people with disabilities has been introduced. Furthermore, the kinematics model of our wheelchair platform and its path controller has been derived.

2.1 Research on Robotic Wheelchairs

> *A* person's control of his/her personal space is an important component of human dignity and the quality of life [20].

Robotics technology has been applied to assist people with disabilities. Robotics wheelchair is an important part in this broad field.

Figure 2.1 shows the block diagram of a standard powered wheelchair. The user interacts with the wheelchair using an access method such as joystick or sip-and-puff system. The commands given through the access method are passed to the wheelchair controller as motor commands consisting of a direction component and a speed component.

Research in the field of robotic wheelchairs seeks to address issues such as safe navigation, splitting control between the user and the wheelchair, and creating systems that will be usable by the target population. Robotic wheelchairs are usually built with standard powered

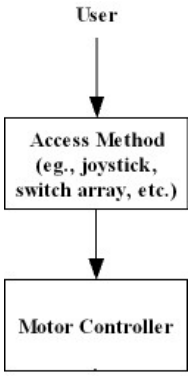


Figure 2.1. Block diagram of a standard powered wheelchair.

wheelchairs for their bases as the research focus is not on improving the mechanical design of the standard powered wheelchair. [22] presents a literature review covering many aspects of powered mobility and [23] discusses issues for engineering both powered and manual wheelchairs.

Figure 2.2 shows the block diagram of common autonomous wheelchair systems. The user gives commands to the user interface using an access method. The command from the user interface is passed to the navigation system along with sensor readings and information from the vision system. Sensor readings are also used for mode detection, which determines the proper navigation code to use for the current environment. The navigation system computes the correct motor commands and passes them to the motor control.

The OMNI project[24, 26, 27, 25] uses a custom-designed omnidirectional wheelchair as its base. Some ultrasonic and infrared sensors provide assistance through obstacle avoidance, wall following and door passage. The wheelchair can rotate around its center point, allowing it to move in tighter spaces than a standard powered wheelchair base.

Another custom designed omnidirectional wheelchair was built in the Mechanical Engineering department at MIT[28]. Semi-autonomous control and autonomous control were assisted by ultrasonic sensors. Horseback riding strategy was used in semi-autonomous control. A

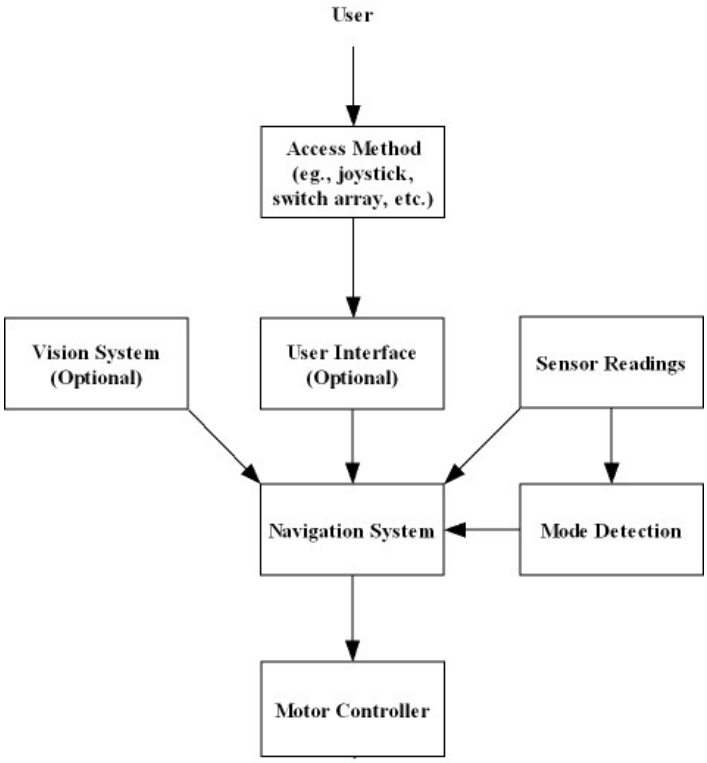


Figure 2.2. Block diagram of control of common prototypes of autonomous wheelchairs.

horse will follow its rider's commands, but not if they put the horse in danger.

A system built by Connell[29] also follows a horseback riding analogy. The user would sit on a chair on a mobile robot base. A joystick was used for driving the system. A bank of toggle switches were used to turn on or off the ability of the robot to perform some tasks autonomously. These behaviors include obstacle avoidance, hallway traversal, turning at doors and following other moving objects. The robot is equipped with ultrasonic, infrared and bump sensors.

An autonomous robotic wheelchair was developed at Arizona State University[30]. The purpose of the system was to transport its user to a specified room in a building using a map of the environment. The wheelchair has been equipped with an on-board microcomputer, a digital camera, and a scanning ultrasonic rangefinder for obstacle avoidance. The system used only a restricted amount of vision processing to locate and verify known objects such as room numbers, look at elevator lights and keep the wheelchair centered in the hallway.

Wheelesley[32] project is based on the platform built by KISS Institute for Practical Robotics. Wheelesley consists of an electric wheelchair outfitted with a computer and infrared, bump and ultrasonic sensors and a laptop that is used for the user interface. The user interface developed allows the user to operate in three modes: manual, joystick and user interface. In manual mode, the wheelchair functions as a normal electric wheelchair. In joystick mode, the user issues directional command through the joystick while the robot will avoid objects in the requested path. In user interface mode, the user interacts with the robot solely through the user interface. The robot can travel semi-autonomously in an indoor environment. This allows the user to issue general directional commands and to rely upon the robot to carry out the low level routines such as object avoidance and wall following.

Hephaestus, the greek god of fire, craftsmen and smiths was the only Olympian with a disability. To compensate for his disability Hephaestus built two robots, one silver and one gold, to transport him. The Hephaestus Smart Wheelchair System[34] aims to be a

navigation assistant that can be added to any powered wheelchair. The system would be installed between the wheelchair's joystick and motor controller. The first prototype has been tried with one powered wheelchair base.

The NavChair navigates in indoor office environments using ultrasonic sensors, and an interface module interposed between the joystick and power module of the wheelchair. The NavChair has three operating modes: general obstacle avoidance, door passage, and automatic wall following. The system can select a mode automatically based on the environment[36]. The NavChair has application to the development and testing of"shared control" systems where a human and machine share control of a system and the machine can automatically adapt to human behaviors.

Senario[38, 39]can be operated in a semi-autonomous or fully autonomous mode. In semi-autonomous mode, the system accepts commands through a voice-activated or joystick interface and supports robot motion with obstacle/collision avoidance features. Fully autonomous mode is a superset of semi-autonomous mode with the additional ability to execute autonomously high-level go-to-goal commands. The user can override in semi-autonomous mode. The wheelchair will stop moving if an emergency situation is detected. The system uses 13 ultrasonic sensors, split into navigation sensors and protection sensors. Two encoders provide a rough orientation estimate. Two infrared range finders mounted at 192cm (above the user's head) are also used for calculating positioning information.

A deictic navigation system has been developed for shared control of a robotic wheelchair[40]. Shared control approach divides task responsibilities between the user(high level) and the robot(low level). The user of the wheelchair tells the robot where to go by clicking on a landmark in the screen image from the robot's camera and by setting parameters for motion, where the target should be at the end of motion, what the distance between the robot and the target at the end of the motion and the desired speed in a computer window. The robot then extracts the region around the mouse click to determine to which landmark the

user wishes to travel. It then uses the parameters to plan and execute the route to the landmark.

Wakaumi[41]developed a robotic wheelchair that drove along a magnetic ferrite marker lane. A magnetic lane is preferable to other nonmagnetic materials due to its ability to continue to work in the presence of dirt on the line. Two infrared sensors in front of the wheelchair have been added for obstacle detection. This type of system is useful for a nursing home environment to allow people to drive around without the need for being pushed by a care giver.

A wheelchair developed at Notre Dame[42] provides task-level supervisory control; the user can select the nominal speed, stop and select a new destination or stop and take over control. The system is taught 'reference paths' during set up which are stored in memory. Visual assistance from two cameras are used to correct errors. The system does not include obstacle avoidance function. If an obstacle is put on the path, the operator needs to take over control to maneuver around it and can then pass control back to the system.

The VAHM project[43, 44] operates in an assisted manual mode and an automatic mode. The philosophy of this project is the person supervises the robot in automatic mode, overriding robotic commands that are unwanted, and the robot supervisees the person in assisted manual mode, overriding commands that put the user in danger.

The Intelligent Wheelchair Project[45] is developed at University of Texas. The wheelchair is enabled with active vision and other sensing modes, spatial knowledge representation and reasoning. The environment is learned through local observations. The system uses stereo color vision, in addition to ultrasonic and infrared sensors to assist movement.

A pushrim-activated power-assisted wheelchair (PAPAW) that use a combination of human power and electric power has been developed[46, 47]. The human power is delivered by the arms through the pushrims while the electric power is delivered by a battery through two

electric motors. The peak torque used to push the rim was significantly reduced. Intuitive control reduces the strain on the upper extremities commonly associated with secondary disabling conditions among manual wheelchair users.

2.2 Definition of the Collaborative Wheelchair Assistant

The Collaborative Wheelchair Assistant (CWA) implements path constraints to facilitate manoeuvering of a wheelchair. The current prototype is based on a commercial wheelchair with a laptop providing control and a graphical user interface. This platform enables development of human-machine interface strategies[9], in particular the elastic path controller, which will enable operators to deform the desired path and so avoid obstacles and modify the path when necessary. These path modifications are controlled by the user via some interface, currently a joystick, so rely on the capabilities of the user and do not require external sensors or sensor processing. Figure 2.3 is the block diagram of this new application of cobot.

While the members of the target community may have different disabilities, we assume that they have some common abilities. We expect that any potential user can see and give high-level commands to the wheelchair through some access method. We also assume that potential users have the cognitive ability to learn to operate the system. Finally we require that the system be able to navigate in indoor and outdoor environments.

2.3 Kinematics

2.3.1 Kinematics model of a moving point

Following the exposition of [18], we will first look at the kinematics model of a moving point, corresponding to Figure 2.4.

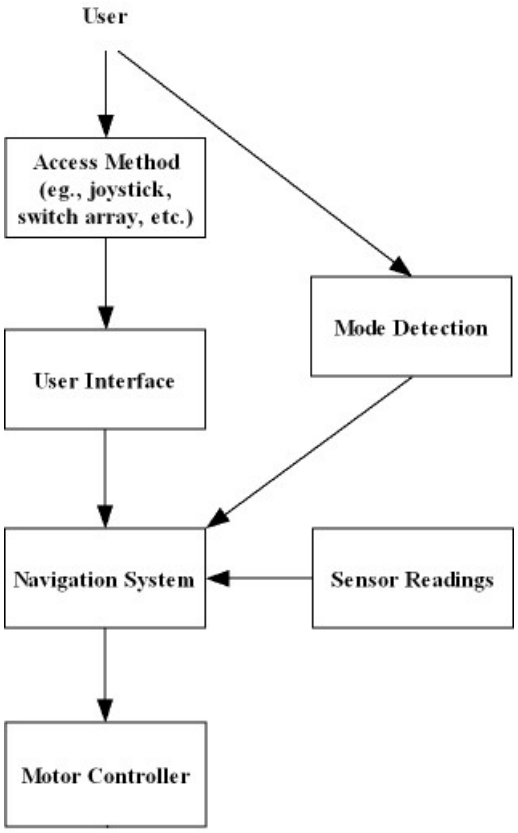


Figure 2.3. Block diagram of Collaborative Wheelchair Assistant. The user gives movement commands to the wheelchair through an access method. The signals from the access method are passed to user interface. Information from User Interface, Positioning Sensors Readings and Mode Detection dictated by the user will help the navigation system to give the correct commands which will be translated into motor commands that are passed to the motor controller.

Definition 2.3.1. $\mathcal{M}$ is a point which is moving to the curve $\mathcal{C}$ defined in the Frenet frame $\mathcal{T}$. The point $\mathcal{P}$ is the orthogonal projection of the point $\mathcal{M}$ onto the curve $\mathcal{C}$. And $\mathcal{O}$ is the origin of the global frame $\mathcal{R}$.

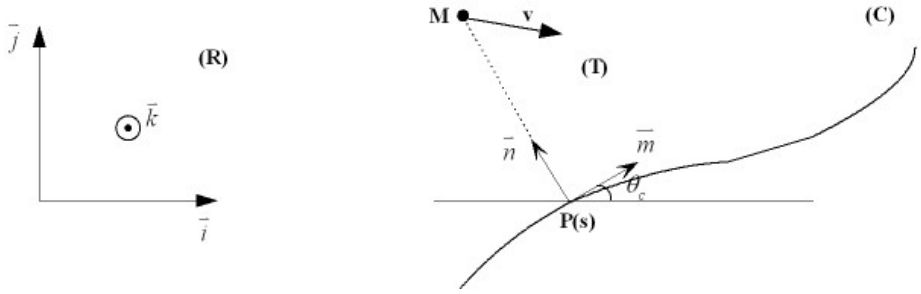


Figure 2.4. Frames and Notations.

A classical law of Mechanics gives:

$$\left(\frac{d\overrightarrow{OM}}{dt}\right)_R = \left(\frac{d\overrightarrow{OP}}{dt}\right)_R + \left(\frac{d\overrightarrow{PM}}{dt}\right)_T + \vec{w_c} \times \overrightarrow{PM} \tag{2.3.1}$$

With

$$\left[\overrightarrow{PM}\right]_T = \begin{bmatrix} 0 \\ y \\ 0 \end{bmatrix} \quad \left[\overrightarrow{W_c}\right] = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_c = c_c(s)\dot{s} \end{bmatrix} \tag{2.3.2}$$

and

s, y: Curvilinear coordinate of a point (M) along the guiding path and its normal distance

θ_c: The angle of the tangent to the guiding path relative to the fixed frame (x,y)

$c_c(s)$: The changing curvature of the guiding path

$\left(\frac{d\overrightarrow{OM}}{dt}\right)_R$: The velocity of point M measured on the reference frame(R)

$\left(\frac{d\overrightarrow{OP}}{dt}\right)_R$: The velocity of point T to the frame(R)

$\left(\frac{d\overrightarrow{PM}}{dt}\right)_T + \vec{w}_c \times \overrightarrow{PM}$: The velocity of point M to the frame(T)

$[w_c]_R$: the rotation velocity vector of frame(T) w.r.t frame (R)

$\left(\frac{d}{dt}\right)_R$: time derivation w.r.t the frame(R), $c_c(s)$ is the path's curvature at frame(T).

Then the system equations of a point relative to a given curve are (For details, please refer to Appendix A):

$$\begin{cases} \dot{s} = (\cos\theta_c \quad \sin\theta_c) \cdot \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} / [1 - c_c(s)y] \\ \dot{y} = (-\sin\theta_c \quad \cos\theta_c) \cdot \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} \end{cases} \tag{2.3.3}$$

and

$\left(\dot{X}, \dot{Y}\right)$: The velocities of the point along the abscissa and ordinate of the fixed frame (x, y).

This set of equations can also be regarded as the transformation relationship from frame(R) to frame(T) of a point.

2.3.2 Kinematics model of Collaborative Wheelchair Assistant

Our wheelchair platform has two actuated wheels on a common axis and the reference point $\mathcal{M}$ at mid-distance of these two wheels (see Figure 2.5), so the kinematic equations of this unicycle-type vehicle are as follows:

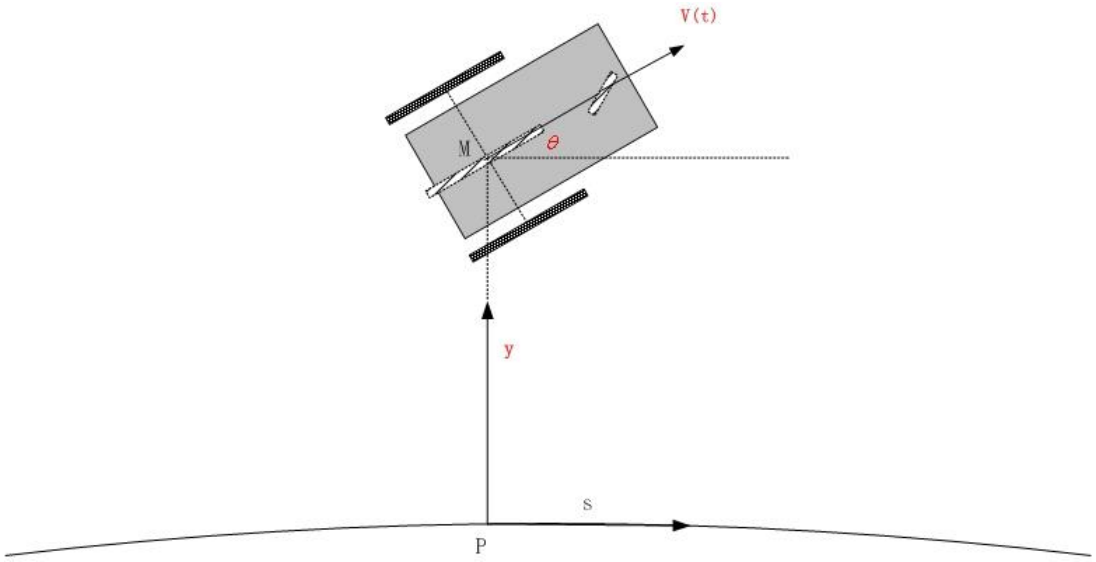


Figure 2.5. Schematic diagram of kinematics model of CWA.

$$\begin{cases} \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = v \cdot \begin{pmatrix} \cos\theta_m \\ \sin\theta_m \end{pmatrix} \\ \dot{\theta}_m = w \end{cases} \tag{2.3.4}$$

and

v: The moving speed of CWA in the direction of normal to its common axis

$\dot{\theta}_m$: The orientation of the CWA w.r.t the fixed frame (x, y).

From the above two functions, we have the following expression of unicycle expressed in coordinate $\{s, y\}$:

$$\begin{cases} \dot{s} = v\cos(\theta_m - \theta_c)/(1 - c_c y) \\ \dot{y} = v\sin(\theta_m - \theta_c) \\ \dot{\theta}_m = w \end{cases} \tag{2.3.5}$$

For simplicity, we make $\theta = \theta_m - \theta_c$, so we have the kinematics function of CWA expressed as:

$$\begin{cases} \dot{s} = v\cos\theta/(1 - c_c y) \\ \dot{y} = v\sin\theta \\ \dot{\theta}_m = w \end{cases} \tag{2.3.6}$$

2.4 Path controller

The control variable chosen for this system is the angular velocity w. To derive the control variable w, modify the kinematics model of unicycle-type vehicle in terms of the distance travelled by the vehicle along the desired path instead of the time-index t. After easy calculation, we get the expression below (Please refer to Appendix B for details):

$$\begin{cases} s' = sign(v\dfrac{\cos\theta}{1 - c_c y}) \\ y' = \tan\theta(1 - c_c y)sign(v\dfrac{\cos\theta}{1 - c_c y}) \\ \theta' = \dfrac{w|1 - c_c y|}{|v\cos\theta|} - c_c sign(v\dfrac{\cos\theta}{1 - c_c y}) \end{cases} \tag{2.4.1}$$

The control objective is to stabilize the output y to zero. Since the control does not explicitly appear in the expression of y', a second derivation is needed. After lengthy but straight calculation,we can get the second derivation of y.

$$y'' = \frac{w}{v\cos^3\theta}(1 - c_c y)^2 - c_c(1 - c_c y)\frac{1 + \sin^2\theta}{\cos^2\theta} - g_c y\tan\theta \tag{2.4.2}$$

This equation is linearized by setting:

$$w = v\frac{\cos\theta}{1-c_c y}\left[u\frac{\cos^2\theta}{1-c_c y} + c_c(1+\sin^2\theta) + g_c y\frac{\sin 2\theta}{2(1-c_c y)}\right] \tag{2.4.3}$$

which results in:

$$y'' = u \tag{2.4.4}$$

The auxiliary control u must be calculated so as to fall upon a stable closed-loop system. Here we choose the following PD control law:

$$u = -k_{py}y - k_{vy}y'; \quad k_{py} > 0, k_{vy} > 0 \tag{2.4.5}$$

Chapter 3

Elastic path controller

The idea of the Elastic Path is to deform the actual path by pushing it perpendicular to the guiding path. You can think of the actual path as a rubber string. The shape of rubber band will be changed when the user give a force perpendicular to it. When the user releases the force, the rubber band will recover to its original shape. Boy et al. developed an elastic path controller which directs the cobot by generating a path curvature necessary to track the ideal path and transforms it from the task space to the wheel space[10]. The individual wheels will then steer to realize this curvature. Unfortunately this controller has a singularity when the tangent vector is normal to the guiding path. This condition does not occur frequently in normal following mode, but can be encountered easily and frequently in elastic path mode. Therefore, the development of a singularity-free Elastic Path Controller (EPC) becomes necessary. In this chapter, such a brand new EPC will be introduced. In our project, the cobot can move on such shape-alterable path when the user activates the elastic mode by pushing or pulling the cobot in order to escape from the guiding path.

3.1 Design requirements

The Elastic Path Controller should meet following requirements:

- In guiding mode the cobot will track the guiding path.
- The EPC enables cobots to deviate from the guiding path according to inputs given by the operator through an interface, such that the deviation is a monotonic function of the input. This means that an input of larger magnitude will lead to larger deviation.

- To avoid undesired deviation from the path, the elastic mode will be activated only when the input from the operator is above a threshold.
- No maximum diviation from the guiding path is specified by the EPC, hence allowing the user to deviate a large amount if necessary, for example to avoid a large obstacle.
- However the ability to deform the path will decrease with the distance to the guiding path, such that the user should not deviate more than necessary from the guiding path and be able to feel a gradient in the direction of this path.

3.2 Elastic Path Controller for the Collaborative Wheelchair

Corresponding to these needs, we propose modifying the control law of Equation 2.4.5 as follows:

$$u = -(1-\alpha)(\underbrace{k_{py}y + k_{vy}y'}_{restoring\ force}) - \underbrace{\alpha F_{\perp}}_{user's\ input} \quad ; \quad k_{py} > 0, k_{vy} > 0 \tag{3.2.1}$$

where $F_{\perp}$ is a function of the input normal to the desired path, as described in Figures 3.1 and 3.2. We use here input but not force because Collaborative Wheelchair Assistant uses a joystick as control interface. The input used in the elastic mode is orientation displacement measured by the joystick. As we will see in next chapter, the scooter cobot uses force/torque as input.
In the first method (Fig. 3.1), the input normal to the current direction of the cobot is used to compute $F_{\perp}$. The user can deform the path independently on the cobot's orientation, as long as he or she is using enough force. With the second method (Fig. 3.2), the normal input relative to the current cobot direction is projected onto the normal to the guiding path. This prevents a large change of orientation relative to the guiding path and limits it to 90^o. If the normal to the guiding path would be used directly, the deformation would be larger when the cobot is normal to the path than when it is almost parallel to it. Therefore the user may not feel where the guiding path is.

In Equation 3.2.1, the elasticity term is composed of the constant elasticity parameter α and $F_{\perp}$ which is a function of the normal input signal. To realize the conditions listed under section 3.1 we use an elastic factor α computed as follows:

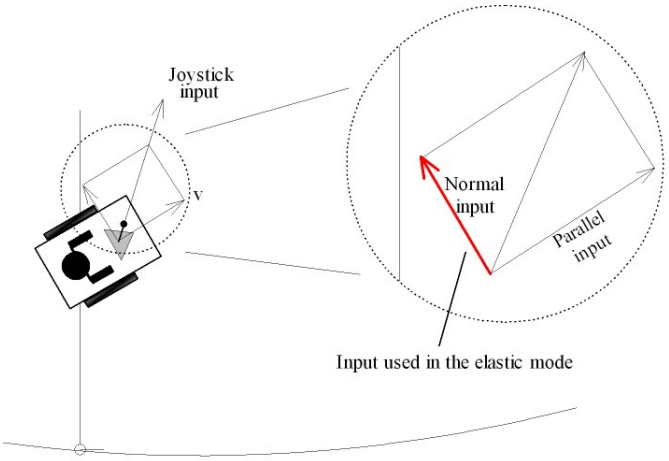


Figure 3.1. Input normal to the current cobot's direction used as to deviate from the prescribed path.

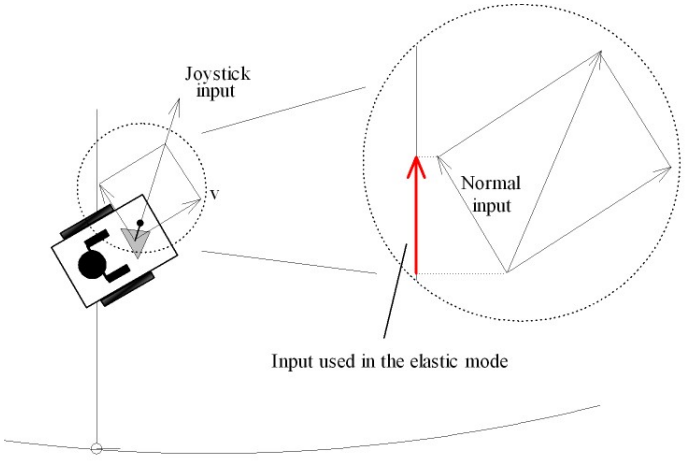


Figure 3.2. Projection of normal input (relative to a local cobot frame) on the normal to the guiding path used to deviate from this guideway.

$$\alpha = \left[\frac{1}{2}\left(\left(\frac{F_\perp}{F_m}\right)^2 - \left(\frac{D}{D_m}\right)^2 + 1\right)\right]_{0.1}^{0.9} I_{\{F_\perp > 5N\}} \tag{3.2.2}$$

where α is an elastic factor which weighs the influence of the user's input $F_\perp$ on the control, $\mathcal{F}_\perp$ is the normal input/force to steer the elastic mode, $\mathcal{F}_m$ is the maximal normal input/force, $\mathcal{D}_{\mathcal{CP}}$ is the distance between the cobot and the desired path and $\mathcal{D}_m$ is the maximum distance.

To make sure the cobot always can follow the guiding path even in elastic mode, we set an upper limit of the elastic factor at 0.9. A lower limit of elastic factor set at 0.1 insures that the user can deform the trajectory even when the normal distance is large. This is realized through the function $[\cdot]_\nu^\mu \equiv \min\{\max\{\nu, \cdot\}, \mu\}$. $I_{\{F_\perp > 5N\}}$ (where $I_{\{\text{condition}\}}$ is the Kronecker function equal to 1 when the condition is fulfilled and 0 otherwise) ensures that no deformation occurs for $\{|F_\perp| < 5N\}$.

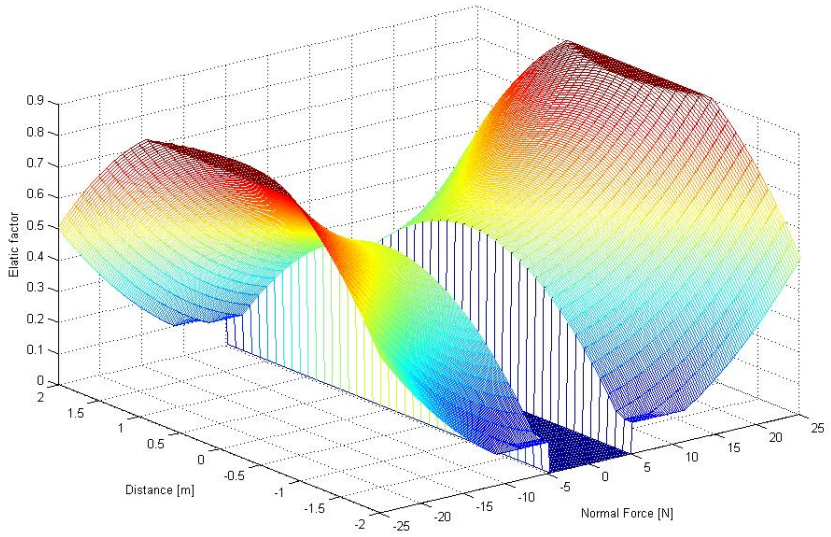


Figure 3.3. Elastic Factor as a function of the elastic force and distance to the guiding path.

Figure 3.3 displays the elastic factor α as a function of $F_\perp$ and distance y. A threshold of $\pm 5N$ is implemented on $F_\perp$ in order to avoid unwanted oscillations around the guiding path.

From

$$\begin{cases} y' = \tan\theta(1 - c_c y) sign(v\dfrac{\cos\theta}{1 - c_c y}) \\ w = v\dfrac{\cos\theta}{1 - c_c y}\left[u\dfrac{\cos^2\theta}{1 - c_c y} + c_c(1 + \sin^2\theta) + g_c y\dfrac{\cos\theta\sin\theta}{1 - c_c y}\right] \\ u = -(1-\alpha)(k_{py}y + k_{vy}y') - \alpha F_\perp \end{cases} \tag{3.2.3}$$

The resulting control is:

$$w = v\frac{\cos\theta}{1 - c_c y}\left\{y\frac{\cos\theta}{1 - c_c y}(g_c\sin\theta - (1-\alpha)k_{py}\cos\theta) + \sin\theta\left[c_c\sin\theta - (1-\alpha)k_{vy}\cos\theta sign(\frac{v\cos\theta}{1 - c_c y})\right]\right.$$
$$\left. -\alpha F_\perp\frac{\cos^2\theta}{1 - c_c y} + c_c\right\} \tag{3.2.4}$$

So the closed-loop system with elastic function is:

$$\begin{cases} \dot{s} = v\cos\theta/(1 - c_c y) \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \dot{\theta_m} - \dot{\theta_c} = v\dfrac{\cos\theta}{1 - c_c y}\left\{y\dfrac{\cos\theta}{1 - c_c y}(g_c\sin\theta - (1-\alpha)k_{py}\cos\theta) + \sin\theta\Big[c_c\sin\theta - \cdots\right. \\ \left. -(1-\alpha)k_{vy}\cos\theta sign(\dfrac{v\cos\theta}{1 - c_c y})\Big] - \alpha F_\perp\dfrac{\cos^2\theta}{1 - c_c y}\right\} \end{cases} \tag{3.2.5}$$

Figure 3.4 is the block diagram of the whole system with elastic characteristic combined.

3.3 Discussion

We have designed an Elastic Path Controller for the wheelchair cobot, which fulfills the requirements listed in Section 3.1:

- The cobot will track the guiding path in guided mode as the elastic path controller is reduced to a path following controller when no elasticity is used.

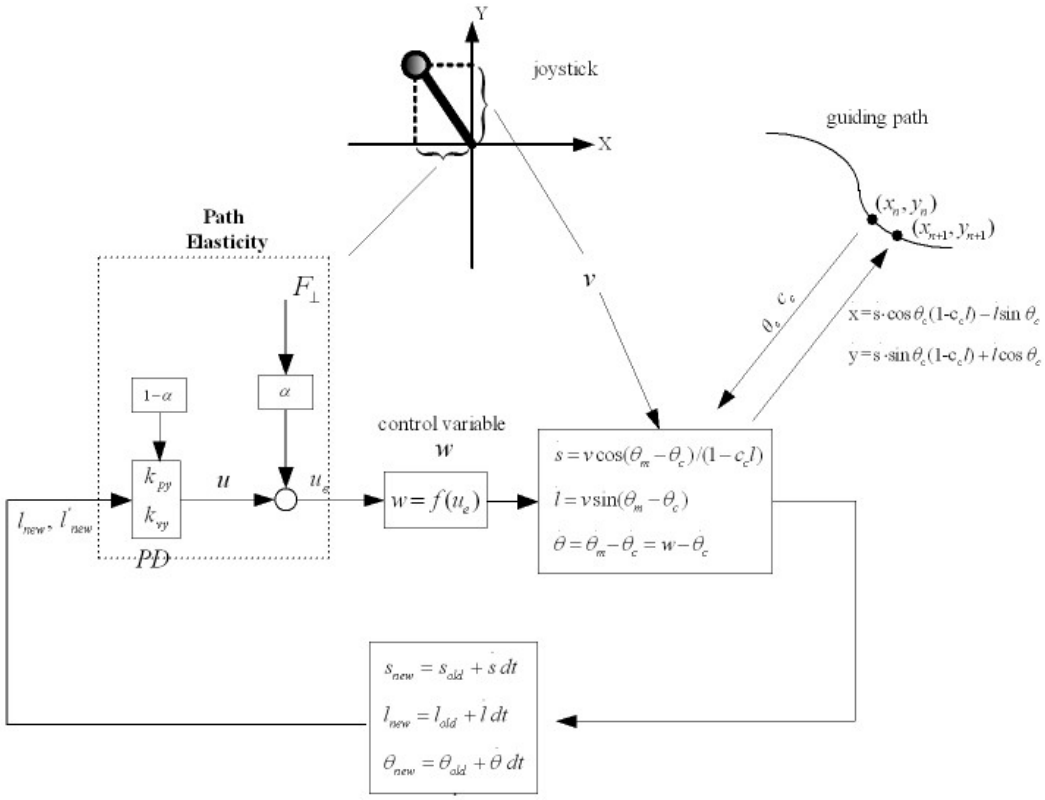


Figure 3.4. Block diagram of Elastic path controller for Collaborative Wheelchair

- Inputs normal to the cobot's path force the linear control to alter its original tracking of the guiding path and deform the trajectory as desired by the user. As the inputs from the operator influence the control following a monotonic rule of the distance to the path, a larger input will be lead to a larger deviation. Please refer to Section 5.1 and Figure 3.3 for details.
- A threshold avoids undesired deviation triggered by unvolunteer input by the operator from eliciting undesired deviation.
- The distance away from the guiding path is not limited by the EPC. However the influence of the normal input signal decreases with the distance to the guiding path. This should help the user to avoid deviating too much from the guiding path and returning to it as soon as the deviation is no longer needed.

Chapter 4

The Scooter Cobot

4.1 Scooter

The Scooter(Figure 4.1) is a triangular vehicle moving on a plane with a steerable wheel at each corner. In Free Mode(FM) each wheel turns like a caster to align with the force exerted by the operator and behaves as if it had 3 DOF (i.e. planar position and orientation). Operator's force is measured by a force-sensor mounted on the handle. In Guided Mode(GM) and Elastic Mode(EM), each wheel is steered by a motor to follow a guiding path coded in software. The Scooter velocity and position are measured using three glide wheels, which are plastic wheels with an encoder mounted at fixed angles of the Scooter Cobot. Encoders measure the rotation of each wheel. The Scooter is controlled by a Pentium Pro 200 MHz 80 MBRAM PC computer operating under QNX system. All programs are written in C language.

The Scooter cobot was developed by Witaya Wannasuphoprasit at the Laboratory for Intelligent Mechanical Systems(LIMS), Northwestern University as a platform to do research on cobots. Eric Faulring converted it into a warehousing "Pallet Jack Cobot" by mounting a freely pivoting handle equipped with an encoder, in order to facilitate smooth transition between free and constrained modes[48].

In contrast to the wheelchair (or unicycle), the Scooter can use the three degrees of freedom (x, y, θ) of planar motion. In guided mode, it is reducing these three degrees of freedom to only one degree of freedom.

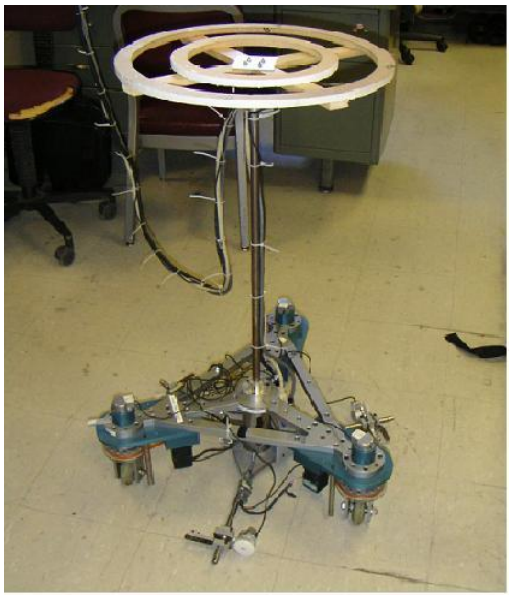


Figure 4.1. Scooter cobot

4.2 Kinematics

We follow the derivation of kinematics and path control of [18]. Figure 4.2 shows a geometric model of two-steering type mobile robot. The wheel's orientation angles are denoted as α and β. The distance between the two wheels is equal to l. As long as the steering wheels are not parallel, the instantaneous motion of the vehicle's body is a pure rotation about the point I_{CR}, termed **I**nstantaneous **C**enter of **R**otation, located at the intersection of the wheel's axes.

The kinematics model of Scooter cobot can be simplified and modified as the two-steering type vehicle when only two wheels among three are considered as the steering wheels. A low-level controller aligns the third wheel to the intersection of the two steering wheels. The cobot position and orientation are described relative to a frame consisting of a curvilinear coordinate s along the guiding path, its normal l and the angle θ_m relative to a fixed frame (x, y):

$$\begin{cases} \dot{s} = v\cos(\theta_m - \theta_c + \alpha)/(1 - c_c y) \\ \dot{y} = v\sin(\theta_m - \theta_c + \alpha) \\ \dot{\theta}_m = v\sigma \end{cases} \tag{4.2.1}$$

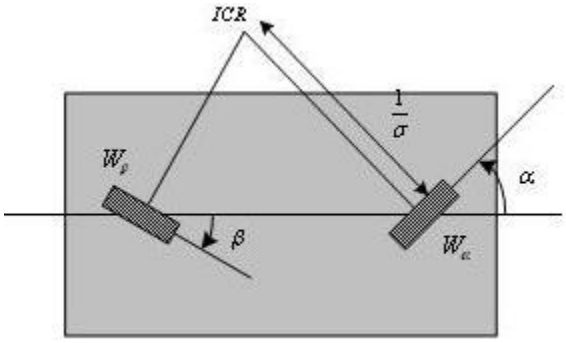


Figure 4.2. Kinematics Model of Scooter cobot

α denotes the orientation of the front wheel relative to the line through the two steering wheels, σ the reciprocal of the length from the leading wheel to the intersection of the normals to the two steering wheels, $v = \sqrt{\dot{x}^2 + \dot{y}^2}$ is the translational speed, c_c the guiding path's curvature, and θ_c the angle of the tangent to the guiding path relative to (x, y).

4.3 Derivation of control variable

Following the same derivation of the unicycle, the kinematics model of two-steering type vehicle can be expressed as below in terms of the distance travelled by the vehicle along the path

$$\begin{cases} s' = sign\left[v\dfrac{\cos(\theta+\alpha)}{1-c_c y}\right] \\ y' = \tan(\theta+\alpha)(1-c_c y)sign\left[v\dfrac{\cos(\theta+\alpha)}{1-c_c y}\right] \\ \theta' = \sigma\left|\dfrac{1-c_c y}{\cos(\theta+\alpha)}\right| sign(v) - c_c sign\left[v\dfrac{\cos(\theta+\alpha)}{1-c_c y}\right] \end{cases} \tag{4.3.1}$$

In this case, we choose two control variables α and σ to linearize the equations of two system outputs,chosen as y and $\tilde{\theta}$. ($\tilde{\theta} \equiv \theta_m - \theta_c - \theta_d$ represents the orientation error, where θ_d is the desired orientation.) Follow the same approach as in unicycle case: Deriving y'and $\tilde{\theta}'$ a second time, then obtains:

$$y'' = \left(\sigma + \frac{\dot{\alpha}}{v}\right)\frac{(1-c_c y)^2}{\cos^3(\theta+\alpha)} - c_c(1-c_c y)\frac{1+\sin^2(\theta+\alpha)}{\cos^2(\theta+\alpha)} - g_c y\tan(\theta+\alpha) \tag{4.3.2}$$

$$\tilde{\theta}'' = \dot{\sigma}\frac{(1-c_c y)^2}{v\cos^2(\theta+\alpha)} - (g_c+g_d) - \sigma\frac{1-c_c y}{\cos(\theta+\alpha)}\left\{\frac{\cos(\theta+\alpha)}{1-c_c y}y\left[g_c\cos(\theta+\alpha)+\cdots\right.\right.$$
$$\left.\left.+k_{py}\sin(\theta+\alpha)\right]+\sin(\theta+\alpha)\left[c_c\cos(\theta+\alpha)+k_{vy}\sin(\theta+\alpha)sign\left[\frac{v\cos(\theta+\alpha)}{1-c_c y}\right]\right]\right\} \tag{4.3.3}$$

Two equations (4.3.2) and (4.3.3) are linearized by setting:

$$\begin{cases} \dot{\alpha} = v\dfrac{\cos(\theta+\alpha)}{1-c_c y}\left\{u_y\dfrac{\cos^2(\theta+\alpha)}{1-c_c y}+c_c[1+\sin^2(\theta+\alpha)]+\cdots\right. \\ \left.+g_c y\dfrac{\sin(\theta+\alpha)\cos(\theta+\alpha)}{1-c_c y}\right\}-v\sigma \\ \dot{\sigma} = v\dfrac{\cos(\theta+\alpha)}{1-c_c y}\left\{\dfrac{\cos(\theta+\alpha)}{1-c_c y}[u_\theta+(g_c+g_d)]+\cdots\right. \\ +\sigma\left\{\dfrac{\cos(\theta+\alpha)}{1-c_c y}y[g_c\cos(\theta+\alpha)+k_{py}\sin(\theta+\alpha)]+\cdots\right. \\ \left.\left.+\sin(\theta+\alpha)\left[c_c\cos(\theta+\alpha)+k_{vy}\sin(\theta+\alpha)sign\left[\dfrac{v\cos(\theta+\alpha)}{1-c_c y}\right]\right]\right\}\right\} \end{cases} \tag{4.3.4}$$

Now choosing the auxiliary control variables u_y and u_θ as:

$$u_y = -k_{py}y - k_{vy}y' \tag{4.3.5}$$

$$u_\theta = -k_{p\theta}\tilde{\theta} - k_{v\theta}\tilde{\theta}' \tag{4.3.6}$$

4.4 Elastic path controller

To realize the elastic mode, we modify the control variables as follows:

$$u_{ye} = -(1-\alpha_1)(\underbrace{k_{py}y+k_{vy}y'}_{restoring\ force}) - \underbrace{\alpha_1 F_\perp}_{input\ force} \quad ; k_{py}>0, k_{vy}>0 \tag{4.4.1}$$

$$u_{\theta e} = -(1-\alpha_2)(\underbrace{k_{p\theta}\tilde{\theta}+k_{v\theta}\tilde{\theta}'}_{restoring\ torque}) - \underbrace{\alpha_2\tau}_{input\ torque} \quad ; k_{p\theta}>0, k_{v\theta}>0 \tag{4.4.2}$$

The scooter has two linear controls because of its two-steering type kinematics. $F_\perp$ and τ in Equation 4.4.1 and 4.4.2 are the force and torque used to activate elastic mode respectively. $F_\perp$ is similar to the one defined for CWA in section 3.2. τ is a circular torque to enable the elasticity of scooter in rotary.

The elastic factor α_1 in Equation 4.4.1 is computed as in Equation **??**. The elastic factor α_2 for torque τ in Equation 4.4.2 is computed in a similar way, as:

$$\alpha_2 = \frac{(\tau/\tau_m)^2 - (D_{CP}/D_m)^2}{2} + 0.5 \tag{4.4.3}$$

where α_2 is the rotary elastic factor weighting the influence of input τ on the restoring force/torque, τ is the input/torque to steer the elastic mode in rotary, τ_m is the maximum input/torque, $\mathcal{D}_{\mathcal{CP}}$ is the distance between the cobot and the desired path, and $\mathcal{D}_m$ is the maximum distance.

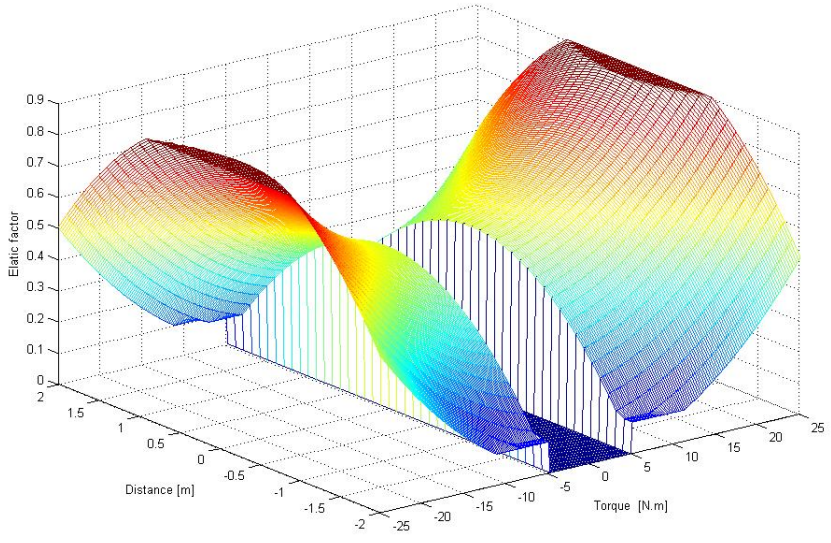


Figure 4.3. Relationship among Elastic Factor in rotary, Torque and Distance

Figure 4.3 displays the relationship between the elastic factor α_2, torque τ and normal distance y. We set an upper limit of the elastic factor α_2 of 0.9 and a lower limit of 0.1. A threshold of $\pm 5Nm$ is implemented on τ to avoid unwanted oscillations in orientation.

The force and torque signals are measured by the sensor mounted on the shaft of the cobot. which translates the user intention. The closed-loop system function with elastic properties becomes :

$$
\begin{cases}
\dot{s} = v\cos(\theta+\alpha)/(1-c_c y)] \\
\dot{y} = v\sin(\theta+\alpha) \\
\dot{\theta} = v\left[\sigma - c_c\dfrac{\cos(\theta+\alpha)}{1-c_c y}\right] \\
\dot{\alpha} = v\dfrac{\cos(\theta+\alpha)}{1-c_c y}\left\{ y\dfrac{\cos(\theta+\alpha)}{1-c_c y}[g_c\sin(\theta+\alpha) - (1-\alpha_1)k_{py}\cos(\theta+\alpha)] + \cdots \right. \\
\quad + \sin(\theta+\alpha)\left[c_c\sin(\theta+\alpha) - (1-\alpha_1)k_{vy}\cos(\theta+\alpha)sign\left[\dfrac{v\cos(\theta+\alpha)}{1-c_c y}\right]\right] + \cdots \\
\quad \left. -\alpha_1 F_\perp\dfrac{\cos^2(\theta+\alpha)}{1-c_c y} + c_c\right\} - v\sigma \\
\dot{\sigma} = v\dfrac{\cos(\theta+\alpha)}{1-c_c y}\left\{\dfrac{\cos(\theta+\alpha)}{1-c_c y}\left[-(1-\alpha_2)k_{p\theta}\tilde{\theta} + (g_c+g_d)\right] + \cdots \right. \\
\quad + \sigma\left\{\dfrac{\cos(\theta+\alpha)}{1-c_c y}y[g_c\cos(\theta+\alpha) + (1-\alpha_1)k_{py}\sin(\theta+\alpha)] + \cdots \right. \\
\quad + \sin(\theta+\alpha)\left[c_c\cos(\theta+\alpha) + (1-\alpha_1)k_{vy}\sin(\theta+\alpha)sign\left[\dfrac{v\cos(\theta+\alpha)}{1-c_c y}\right]\right]\Big\} + \cdots \\
\quad \left. -(1-\alpha_2)k_{v\theta}\left[\sigma - (c_c+c_d)\dfrac{\cos(\theta+\alpha)}{1-c_c y}\right]sign\left[\dfrac{v\cos(\theta+\alpha)}{1-c_c y}\right] - \alpha_2\tau\dfrac{\cos(\theta+\alpha)}{1-c_c y}\right\}
\end{cases}
\tag{4.4.4}
$$

Compared to the CWA, the Scooter cobot uses force/torque as input instead of the joystick orientation. The elasticity was implemented in a similar way than with the CWA and should similarly fulfill the requirements of section 3.1.

Figure 4.4 is the block diagram of the whole system.

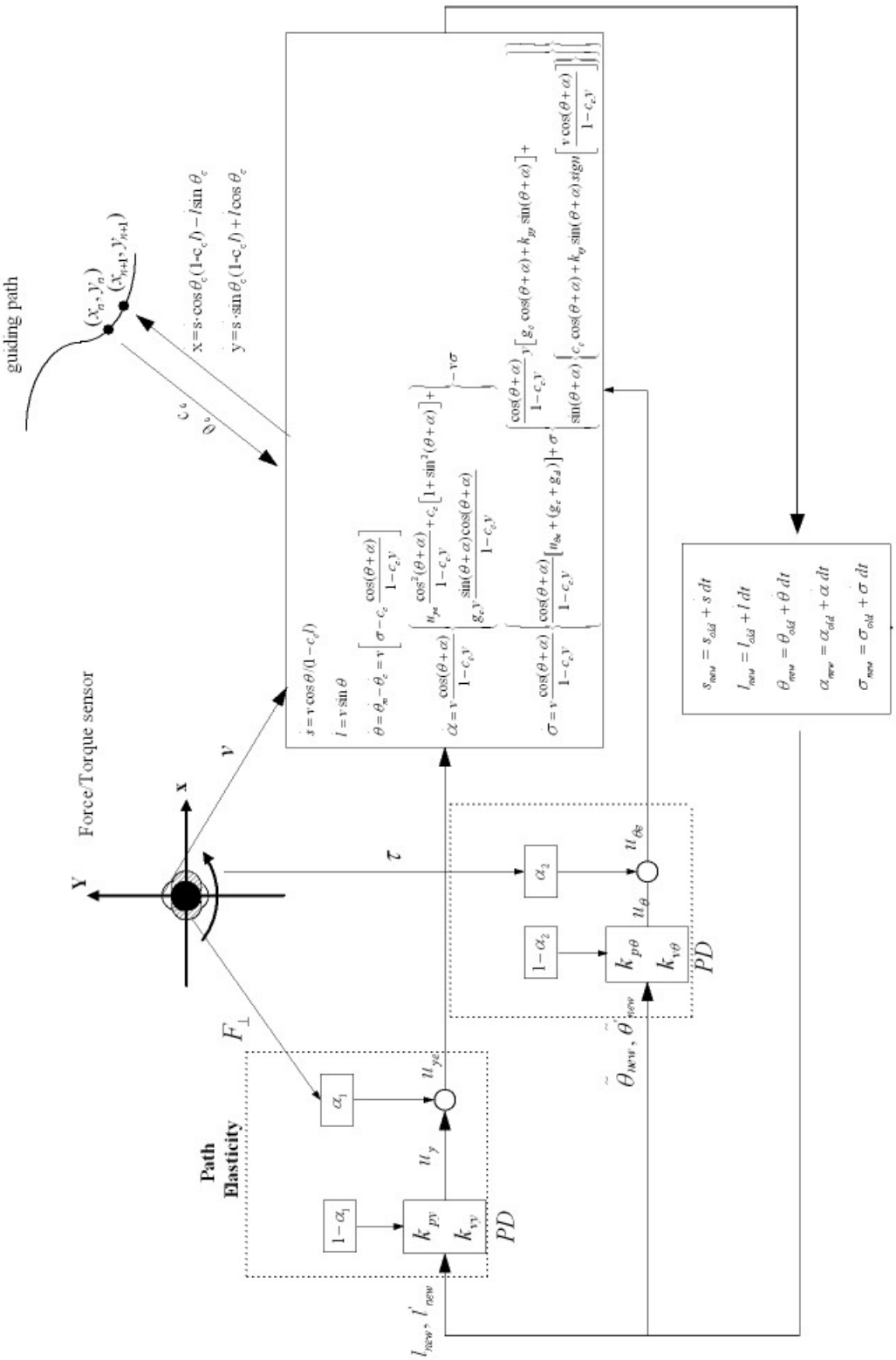


Figure 4.4. Block diagram of Elastic path controller for Scooter cobot

Chapter 5

Simulation on Elastic Path Planner for wheelchair Cobot

5.1 Simulation Environment

A simulation environment was realized in MATLAB to develop and test the elastic path controller. Figure 5.1 presents the Graphical User Interface (GUI). The controller can be selected from the drop down list at the top of the panel area. The wheelchair is represented by a rectangle. The top right panel is where all initial settings for the simulation is controlled:

- Path No. - select between up to ten different paths. If a simulation is currently in progress, it will be aborted and the new selected path will be redrawn.
- Vehicle Initial Orientation - the wheelchair's initial orientation is set with respect to the global frame.
- Vehicle Initial Distance - the initial distance between the guiding path and the wheelchair can be set here.
- Diagram - to display the relationship between two different parameters. For example, if you want to know how the orientation of wheelchair is changed w.r.t time, you can choose the corresponding entry in the dropping list. The function will be plotted during the simulation..
- Show in new window - show the simulation in a separated window with only the animation displayed.

- Draw obstacles - to test the elastic path controller, you can add arbitrary obstacles which represented by filled circles in the central display panel.

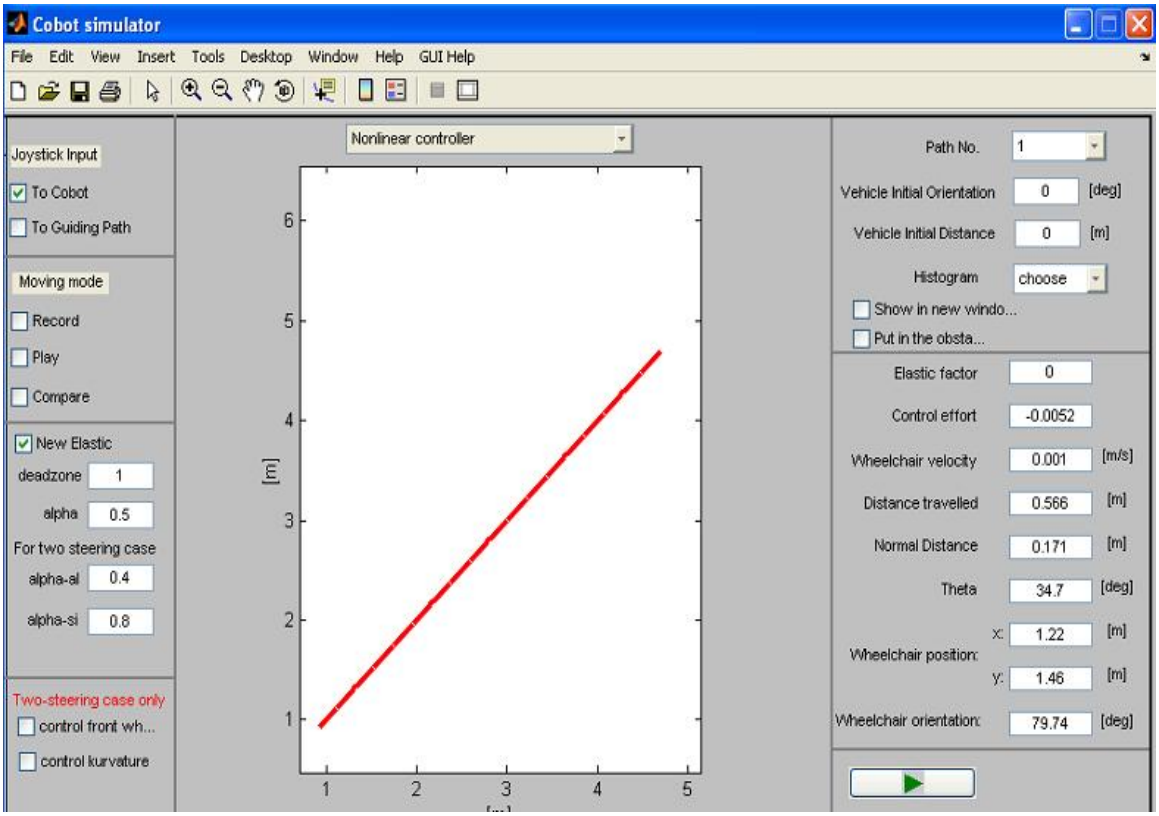


Figure 5.1. Graphical User Interface for Cobot Simulator

The center right area displays information regarding the joystick, condition of the vehicle and the parameters used in the elastic path planner. *Wheelchair orientation*, *Wheelchair position(x,y)* and *Wheelchair velocity* are explicit. *Theta* is the angle between the wheelchair's orientation (globe frame) and the tangent orientation of the projection of the wheelchair's reference point (center) onto the guiding path. *Distance* between the reference point on the wheelchair and the projection of the wheelchair's reference point on the guiding path. *Distance travelled* tells how much the wheelchair has moved. *Elastic factor* and *Wheelchair velocity* both have relationship with the input from joystick. *Wheelchair velocity* is totally decided by the joystick input in the forward direction. *Elastic factor* is decided by both the force given by the user normal to the guiding path and how

far the wheelchair is from the guiding path. It will be zero if no elasticity is used. For more about elastic factor, please refer to 3.2 and Figure 3.3. *Control effort* is the price that directs the cobot towards the guiding path, the further the wheelchair is away from the guiding path, the larger will be this control effort[10].

The top left area *Joystick Input* is setting the way $F_{\perp}$ is computed, either as the local normal or its projection onto the normal to the guiding path.

The area below *Joystick Input* supplies functions to compare different settings. Using *Record*, the program can record every input from joystick and save them to a file. *Play* can load the input data from saved file to repeat joystick's performance. *Compare* is used to compare the performance between different controllers with same settings or same controller with different settings.

The area above the bottom includes entries to control the elasticity of the path controller. In the wheelchair case, you can set the threshold of the joystick's normal input which can activate the elastic mode (Please refer to Figure 5.2 and Table 5.1 for details). *Alpha* is a factor weighing the effect between the *restoring force* and *elastic force* (See section 3.2). For the Scooter two parameters weight the relationship between restoring and elastic force/torque. Please refer to section 4.4 for more information.

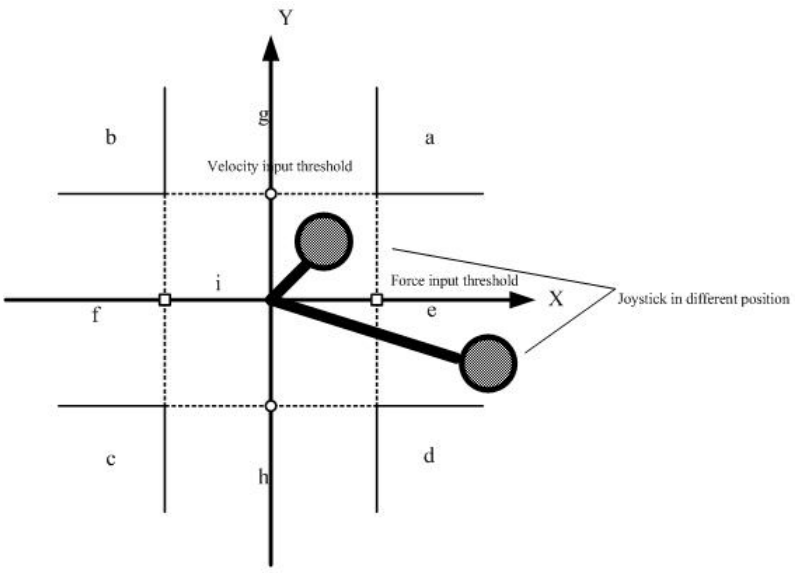


Figure 5.2. Joystick Frames and Settings Illustration

Table 5.1: Table of functionality of joystick mapping

Position	X	Y	Movement condition	Elastic Mode or not
a	> 0	> 0	forward	Yes
b	< 0	> 0	forward	Yes
c	< 0	< 0	backward	Yes
d	> 0	< 0	backward	Yes
e	> 0	$= 0$	stop	Yes
f	< 0	$= 0$	stop	Yes
g	$= 0$	> 0	forward	No
h	$= 0$	< 0	backward	No
i	$= 0$	$= 0$	stop	No

The bottom left area include three options: *Control front wheel*, *Control curvature* to research how the scooter performs in the elastic mode if only one control variable was modified to realize the elasticity. And what is the result when both of two control variables were modified. Modification of front wheel variable will bring elasticity in parallel movement , while controlling curvature will bring the Scooter elasticity in rotation.

The bottom right area controls the starting or stopping of the simulation. Pressing the green arrow button will start a new simulation with the current settings. A new red rectangle button will appear that aborts the simulation and redraws the previous simulation window with the current settings.

5.1.1 Hardware Settings

We use the LOGITECH *WingMan EXTREME* Joystick as an input device with basic trim control as X-axis(left/right) and Y-axis(up/down). The computer has an AMD 1.7GHz Processor, 512MB RAM for Memory, 64MB shared memory Graphics card and the Hard drive with capacity of 40GB(5400rpm) are running under Microsoft Windows 2000 operating system.

5.2 Simulation results of Guided Mode

A flowchart of the program is shown is Figure 5.3. The simulation results below use the controller described in section 2.3.2 and 2.4 (Also can refer to the schematic diagram in Figure 3.4), which

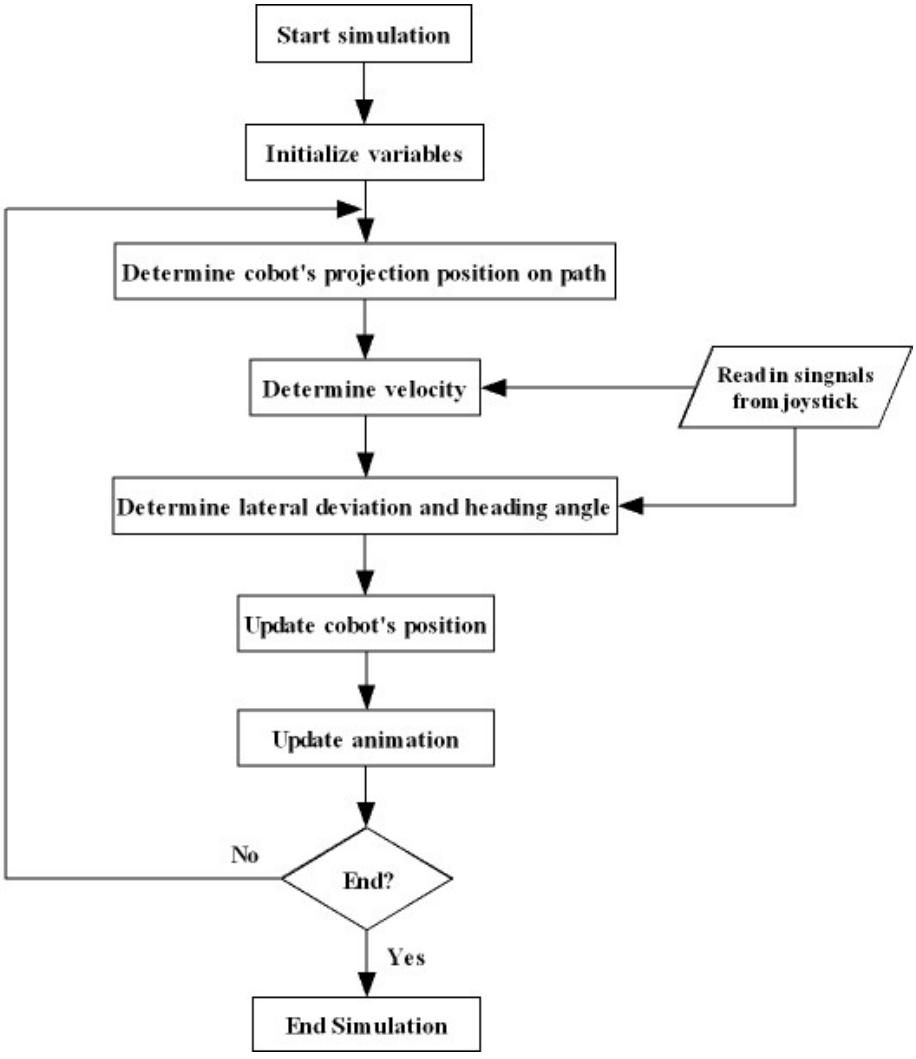


Figure 5.3. Simulation flowchart

was also used for the experiments presented in next chapter. In the figures below, the direction in which the cobot is moving is represented by the black arrow, and the joystick in grey.

5.2.1 Performance of Collaborative Wheelchair Assistant

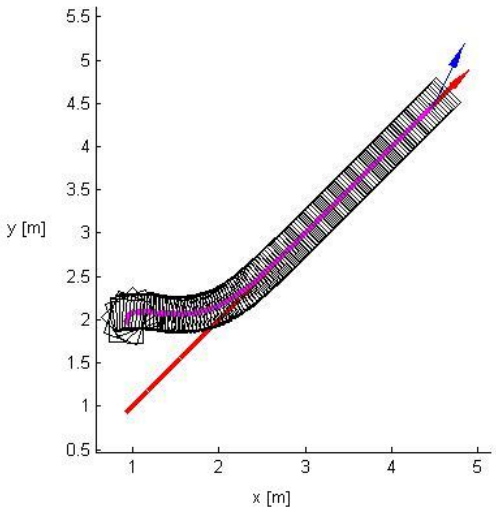


Figure 5.4. Wheelchair in guided mode.

Figure 5.4 and 5.5 present the simulation of the wheelchair along a straight line and a sine wave, respectively. The initial orientation of the guiding path is in both cases $\theta = 45^o$. We see that the path controller is able to track the desired trajectory well.

5.3 Simulation of Elastic Mode

To compare the effect of computing the input to the elastic path controller listed below, we used the same force profile, the same initial orientation -45 degrees with respect to the global frame and all start from the guiding path in one of three situations listed below.

- the normal to the guiding path. (Method 1)
- the normal to the cobot direction (Please refer to Figure 3.1). (Method 2)

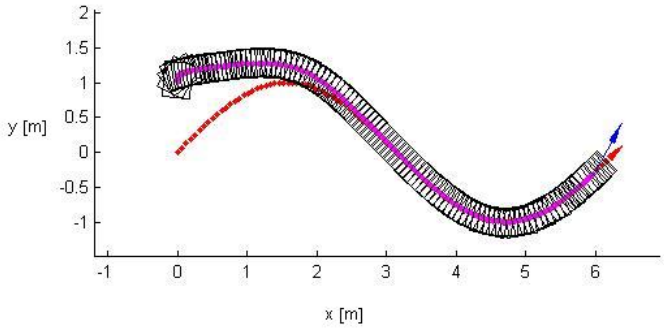


Figure 5.5. Guided mode performance with the wheelchair on a sinusoidal wave

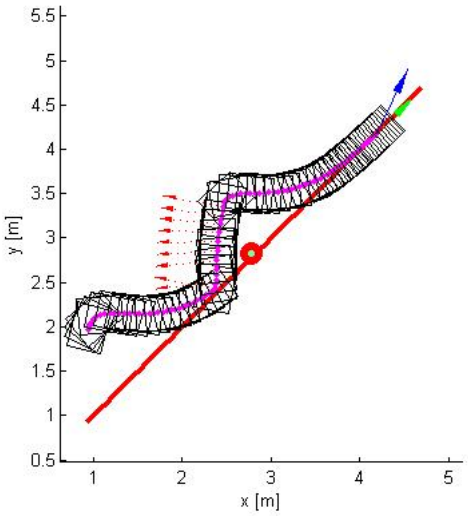


Figure 5.6. Elastic mode to avoid an obstacle with the CWA (Filled circle on the path is the obstacle).

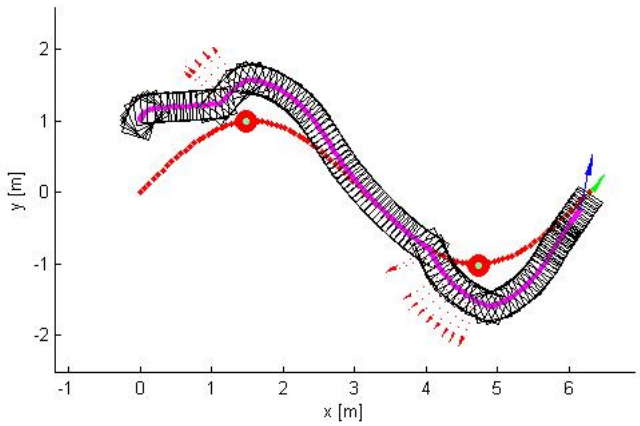


Figure 5.7. Elastic mode performance on sine wave.

- the normal to the cobot direction projected on the normal to the guiding path (Please refer to Figure 3.2). (Method 3)

First, we recorded the performance dominated by the second method which tries to deviate and remain far from the guiding path as the reference inputs for other two methods. In Figure 5.8, the reference inputs are represented by a series of arrows. We found that the cobot deviates more and more if the first method is adopted and use the reference inputs as control flows. And the deviation on the increase happened in first method will be corrected by using third method.

Figures 5.6 and 5.7 demonstrate that the elastic planner enables the user to deform the actual path in order to avoid obstacles. After passing the obstacle, the user releases the normal input and the controller tracks the guiding path. In these simulations the user input to the elastic controller is computed from the second method for $F_{\perp}$ (Figure 3.2). The experiments presented in next chapter show that similar properties hold with the Scooter cobot.

One limitation of the simulation is that the vehicle can turn from $+40^{o}$ to -40^{o} in one time step. In reality the turning rate depends on the moving velocity and on the length of one time step. This may influence the way users avoid obstacles.

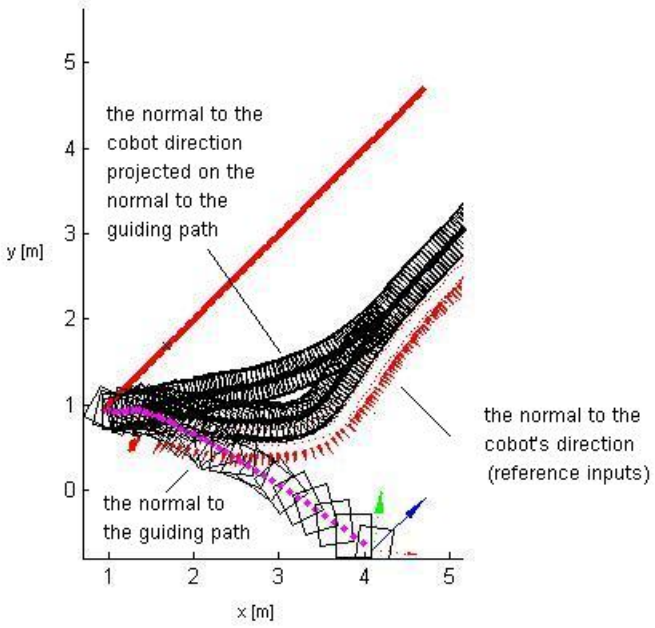


Figure 5.8. Effect of three different methods of computing the input to the elastic path controller

A feature that has not been implemented is measurement noise, i.e. perfect positioning was assumed in the simulation.

The experiments presented in next chapter show that similar properties hold with the Scooter cobot.

Chapter 6

Elastic Guiding Motion Experiments on Scooter

The two experiments reported in this chapter were approved by the Institutional Review Board of Northwestern University, and performed by seven students (mean age 23.5, with standard deviation 3.2) without known motor disability. These subjects were informed about the experiments, and gave their consent prior to participation.

The first experiment investigated how the operators learned to avoid obstacles using the elastic path controller. The subjects were asked to follow a straight path and avoid an obstacle in elastic mode. The result shows that the high frequency content generally decreased with learning. The high frequency content was evaluated as the integral of the Fourier transform from a cutoff frequency of 10Hz to the maximum frequency of 50Hz normalized by the full frequency content. A t-test confirmed a significant difference between the mean value of the first five trials and the last five trials proportion of high frequency content.

However, this does not prove that users will able to perform well in other (possibly unknown) guiding paths, nor does it show the strategy used by the operator. To infer this, we designed a second experiment with paths unknown to the user. The experiment result suggests that the subjects use a similar strategy independent of the direction. They first push or pull against the guiding path by a suitable amount to avoid the obstacle. After the obstacle is passed, they release the normal force and trust the path controller, which leads them to the guiding path. They may be able to feel the gradient of attraction from the given guiding path which can help them manoeuver the scooter cobot. In turn, these results with randomly chosen path demonstrate that the users have learned to

work with the elastic path controller independent of the path.

Subjects were first trained with the cobot to move in a comfortable way in free mode. The training was completed when the subject could pass through the passage (Figure 6.1C) between two styrofoam boards easily. This passage is narrower than the maximum diameter of scooter. The training has a duration of maximally 5 and minimally 3 minutes for the subjects (mean 4 minutes, with standard deviation 1 minute). The subject was asked to follow a virtual straight line and an eight-shape path in guided mode (Figure 6.1A and 6.1B). Subjects were asked to push and pull the scooter around in the training environment and try to feel the guidance provided by the scooter in guided mode.

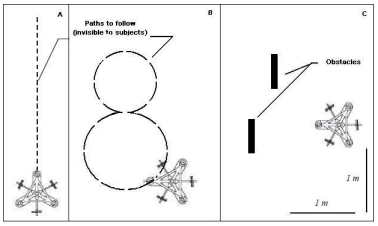


Figure 6.1. Environment to learn moving Scooter cobot

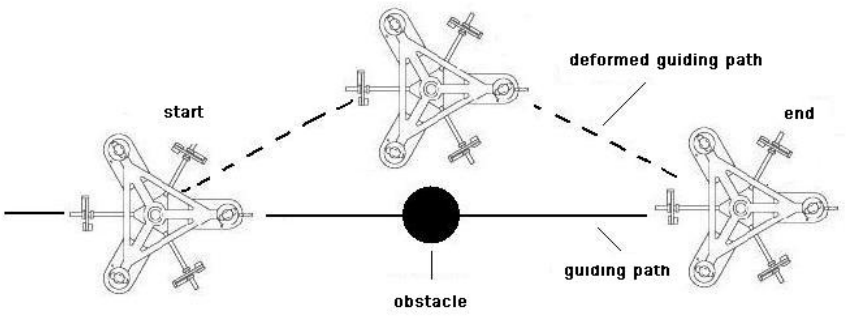


Figure 6.2. In first experiment, we test how users can avoid an obstacle placed along a straight line using the elastic path controller.

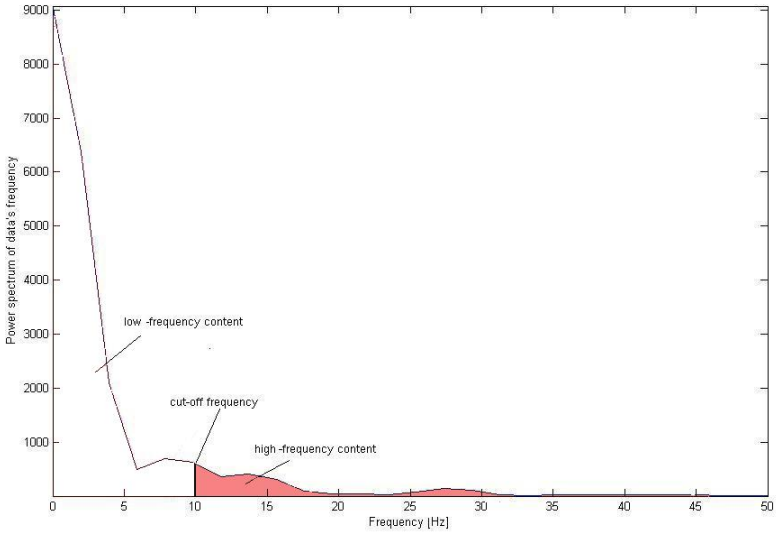


Figure 6.3. Frequency contents of Normal force and high-frequency area.

6.1 Learning to avoid an obstacle

6.1.1 Methods

The subjects were then trained to use the elastic mode. To analyze the performance, the subjects were asked to follow the straight path in elastic mode. A trial was considered as successful when the subject could avoid the obstacle on the path without hitting it and return to the desired straight guiding path after (Figure 6.2). The training was completed when no obstacle was hit and the subject could come back to the desired path in five consecutive trials.

The frequency content was analyzed by computing the Fourier transform and calculating the proportion of the high frequency content. The area below the Fourier transform corresponds to the frequency content. The 'high frequency area' is evaluated as the integral of the Fourier transform from a cutoff frequency of 10Hz to the maximum frequency of 50Hz, and normalized by the full frequency content (See Figure 6.3). A t-test was performed to evaluate the decrease between the ratio of high-frequency content to full frequency content in the first five and last five trials.

6.1.2 Results and Analysis

Using the straight line is advantageous as the subject can concentrate on handling the cobot and the force is not influenced by the changing curvature of the guiding path. Also the straight line is the easiest way to make the subject feel the guidance and elasticity supplied by the elastic path controller. On average, subjects required 17 trials (minimum 13, maximum 25, with standard deviation 4.24) in the same environment. The training session took an average of 20 minutes for each subject with a standard deviation of 5 minutes. Figures 6.4 to 6.7 present the paths and force of two representative subjects repeating trials to learn avoiding an obstacle. The first few trials generally hit the obstacle, however the subjects gradually learned to produce suitable force to avoid the obstacle and release the force after it was passed. The paths and force suggest that distinct subjects have different motion patterns (but showing this would require systematic analysis).

Table 6.1: Statistics of trials hitting the obstacle.

Subjects	Number of Trials	Trials hitting the obstacle
Jannie	18	$1, 2, 3, 4, 5, 7$
Alanna	13	$1, 2, 3, 5$
Christian	25	$1, 2, 3, 4, 6, 8, 10, 14, 16, 20$
Alison	15	$1, 2, 8$
Jeffrey	16	$1, 2, 3, 4, 5, 7$
Scotty	15	$1, 2, 4, 5$
Dave	5	ϕ

Table 6.1 gives a statistics of the trials hitting the obstacle when moving in the elastic mode.
The subjects needed, on average, 8.67 trials (with standard deviation 5.68) before succeeding in five consecutive trials. The large standard deviation indicates that the subjects require very different level of learning. One subject on Tuesday, January 25, 2005 at 11:41 pm performed immediately well in elastic mode and so needed to perform only five trials. The data of this subject will not be further analyzed as he required no learning.
Observation of the path and force (Figure 6.4 to 6.7) suggests that the first trials are jerky, and motions become smoother with learning. To investigate this trend, we compute the proportion of frequency content above 10Hz, as explained in the methods section. Similar results were obtained

using cut-off frequencies until 16Hz. However using a cut-off of 8Hz or less gave no clear trend. This is consistent with the fact that small oscillations and tremor during movements occur at about 10Hz [51]. We observe in Figure 6.8 that the high-frequency content generally decreases with learning. A t-test confirmed that the difference between mean value of first five trials and last five trials' proportion of high frequency content ($p < 0.0079$). The difference is 0.0541 in mean with a standard deviation of 0.0309 (Figure 6.9).

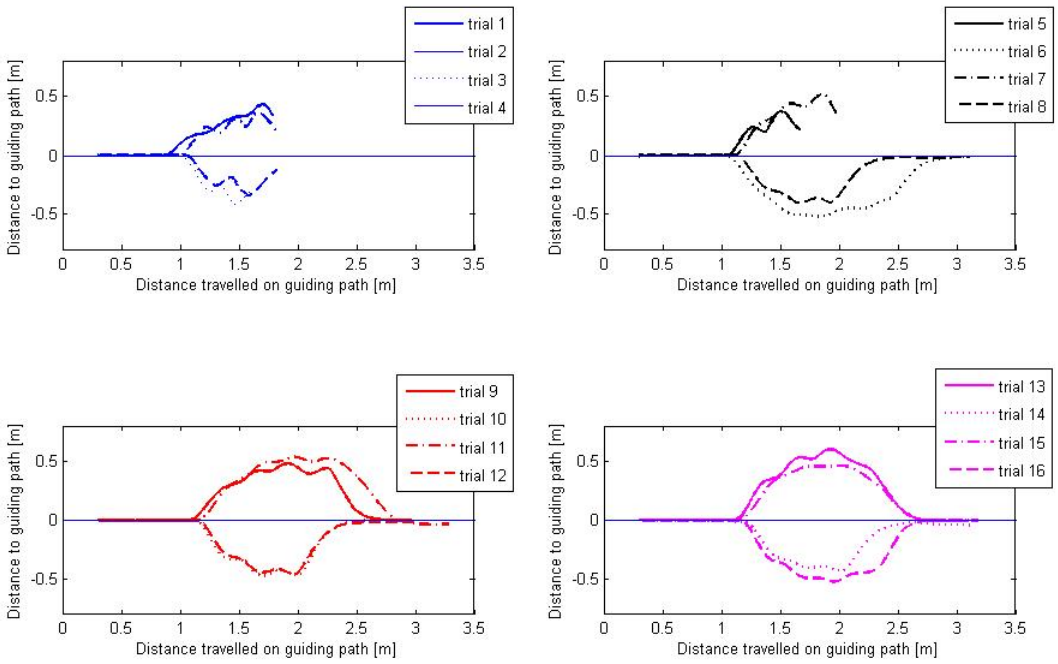


Figure 6.4. Learning to avoid obstacles using the elastic mode. This subject (Jeffrey) first hit the obstacle(trajectories not going back to 0) and gradually learned to avoid it successfully.

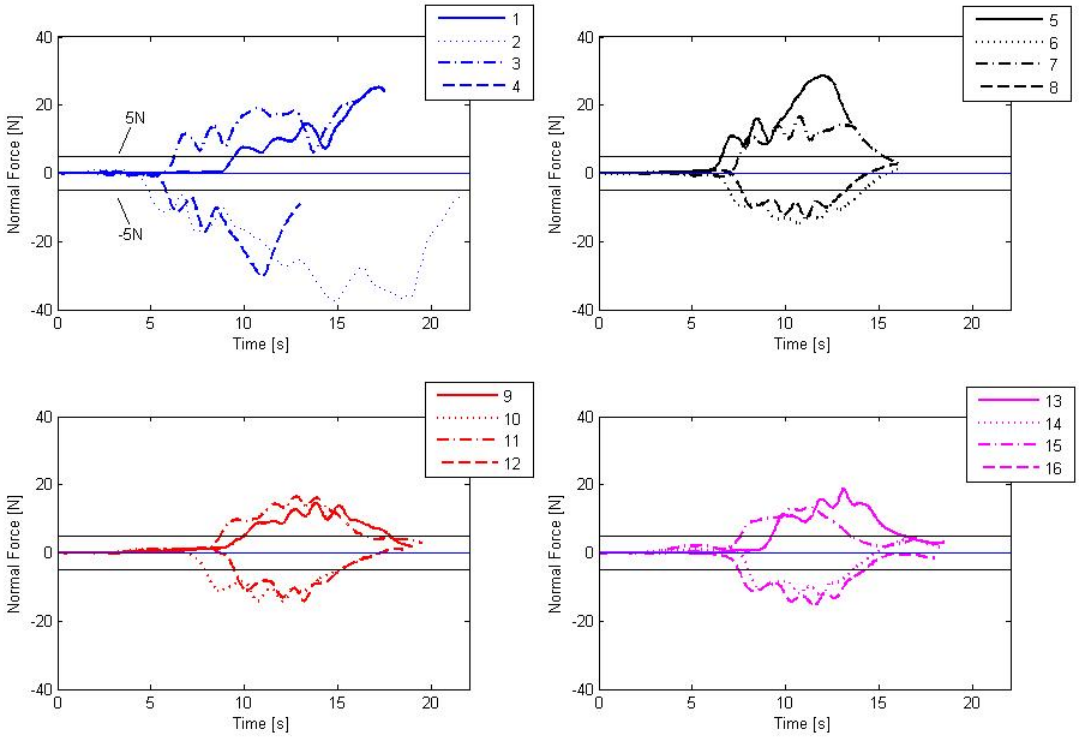


Figure 6.5. Normal force of Jeffrey's trials.

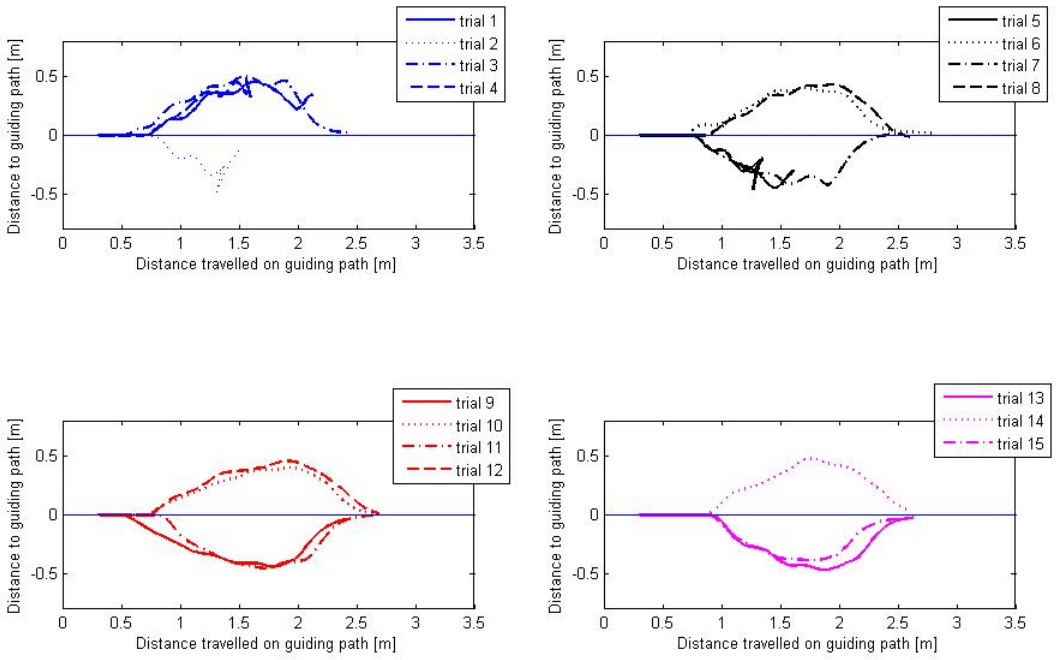


Figure 6.6. Scotty seems to learn avoiding the obstacle in less trials and more easily than Jeffrey (compare with Figure 6.4)

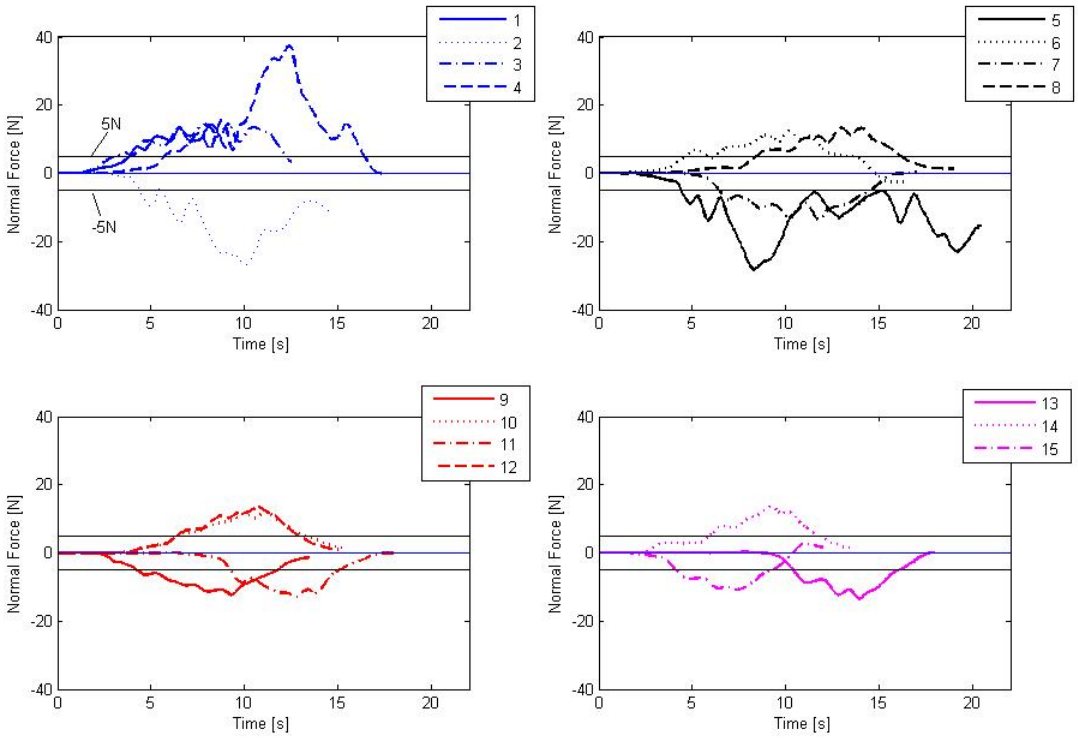
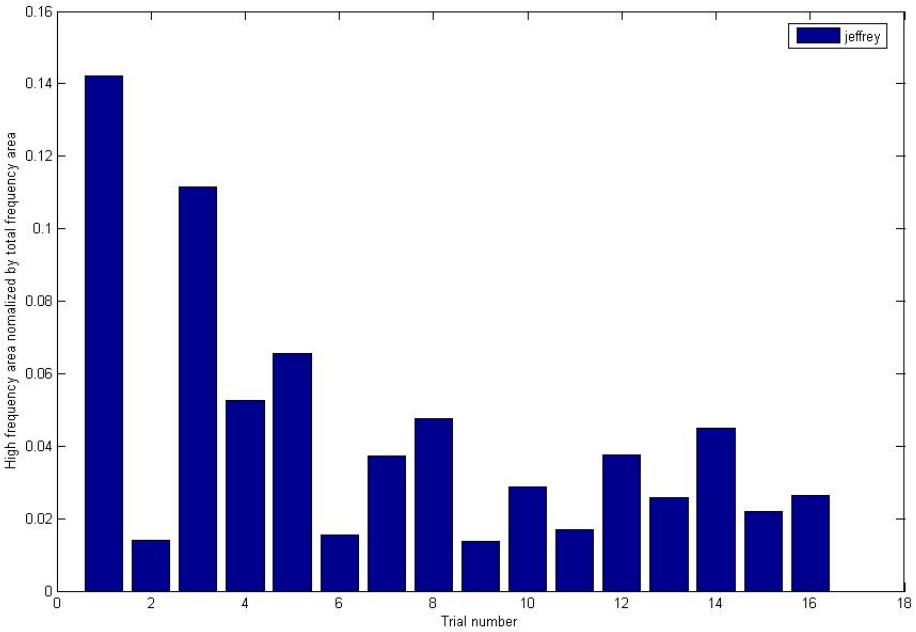
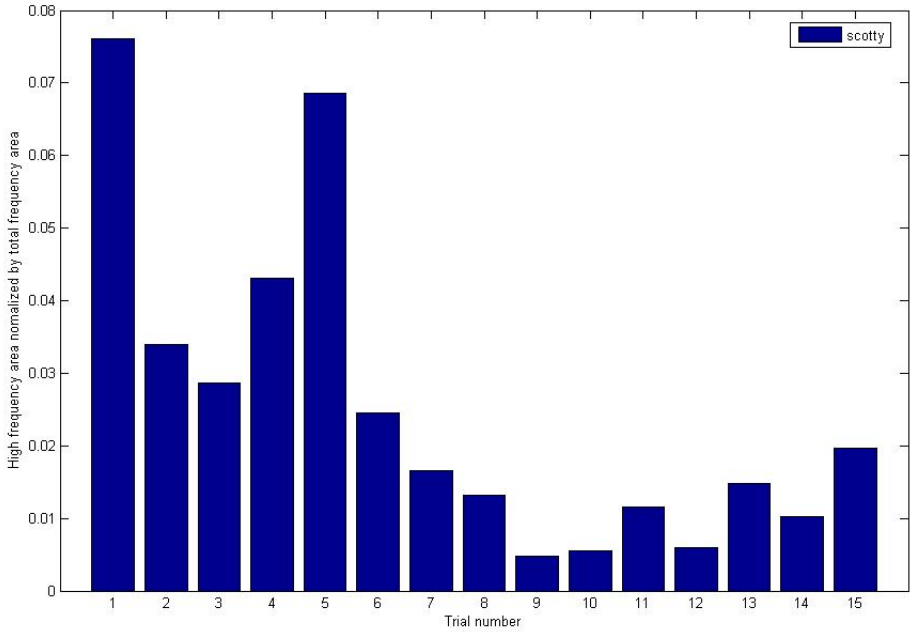


Figure 6.7. Normal force of Scotty's trials.



(a) Jeffrey



(b) Scotty

Figure 6.8. High frequency content divided by total frequency content as a function of the trial number for two typical subjects.

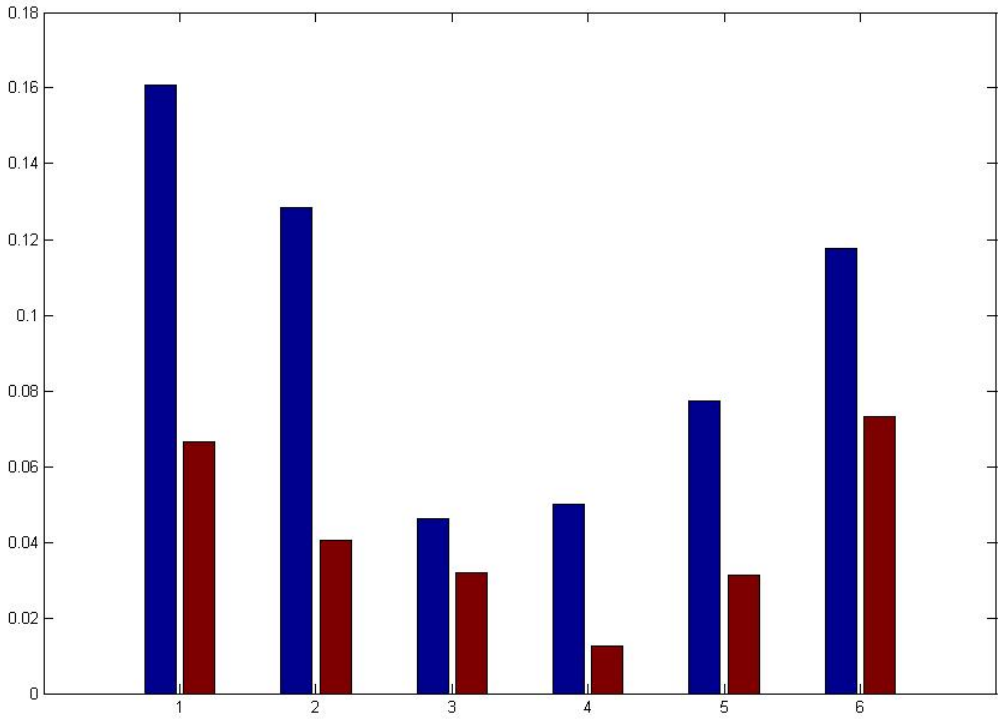


Figure 6.9. Proportion of high frequency content of first five and last five trials for all subjects.

6.2 Hidden paths experiment

The first experiment has shown that the users can learn to avoid a fixed obstacle using the elastic path controller. However, this does not prove that the users will be able to perform well in other (possibly unknown) guiding paths. To ascertain this, we designed a second experiment with unknown guiding paths.

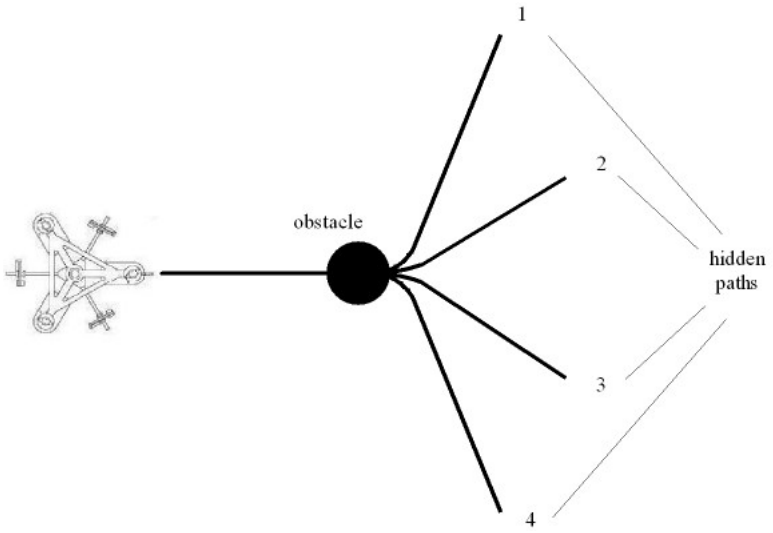
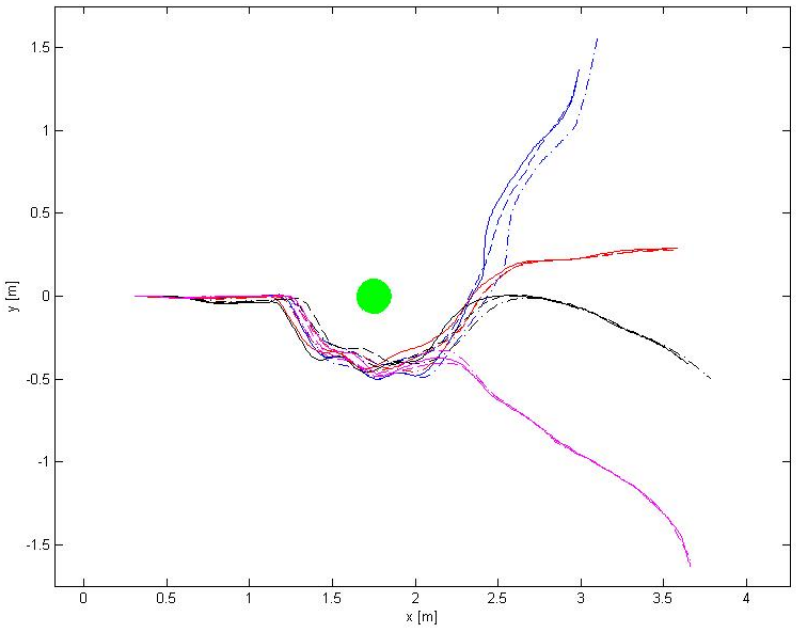


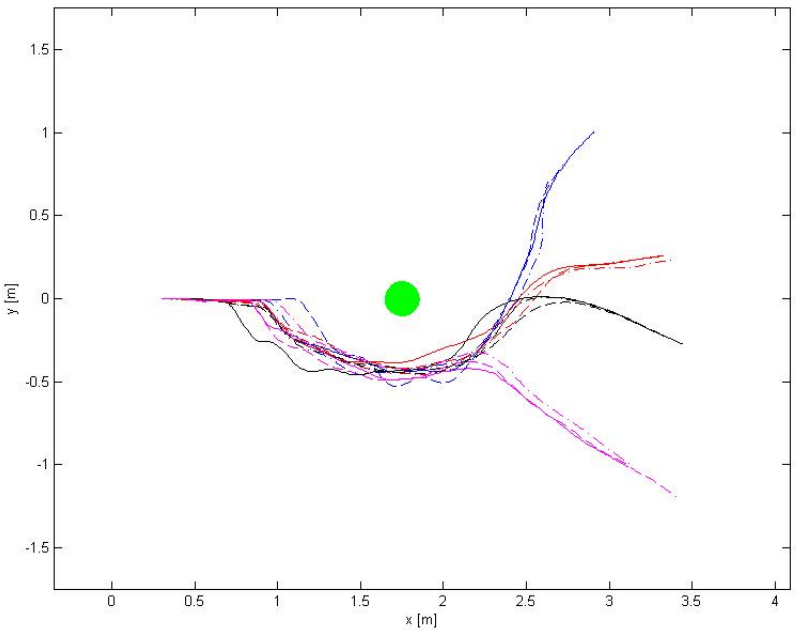
Figure 6.10. Environment for the hidden path experiment

6.2.1 Methods

The environment in which the second experiment was performed is drawn in Figure 6.10. The user is moving on a straight line and instructed to avoid an obstacle, a cylindrical object, from the left hand side. The path diverges to any of four directions after this object. The direction is selected randomly, without the help of any (visual) cues. Thus the subject cannot know which virtual path will be used on a given trial. Each subject performs twelve trials: three in each of the four directions.

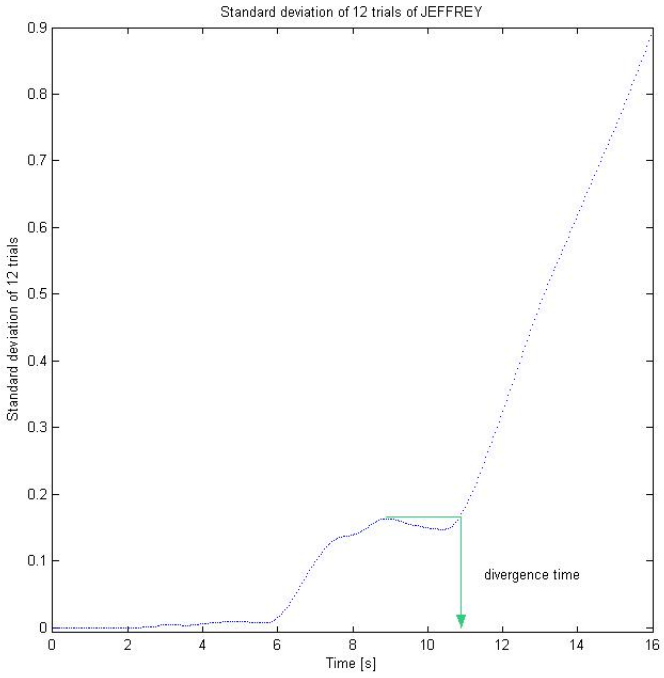


(a) Jeffrey

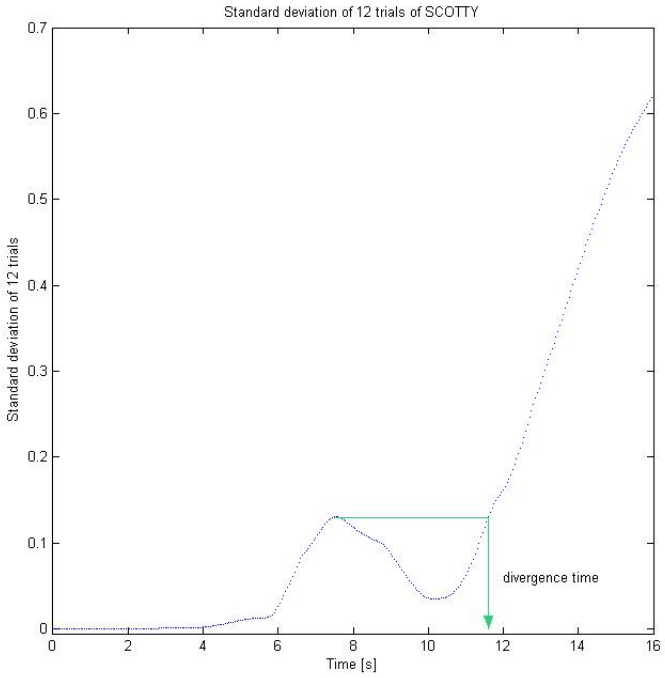


(b) Scotty

Figure 6.11. Paths used in the 12 trials by two typical subjects.

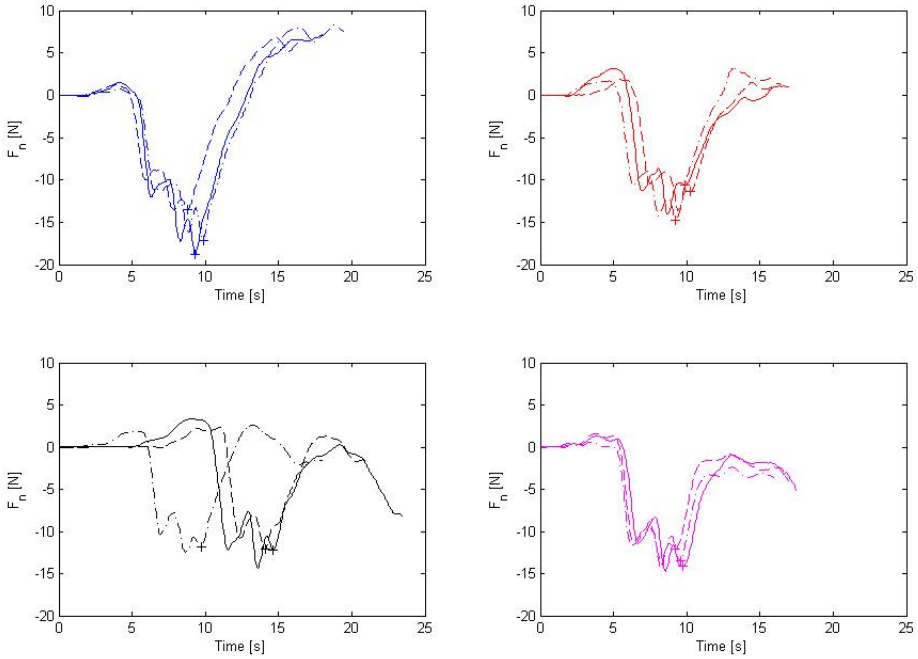


(a) Jeffrey

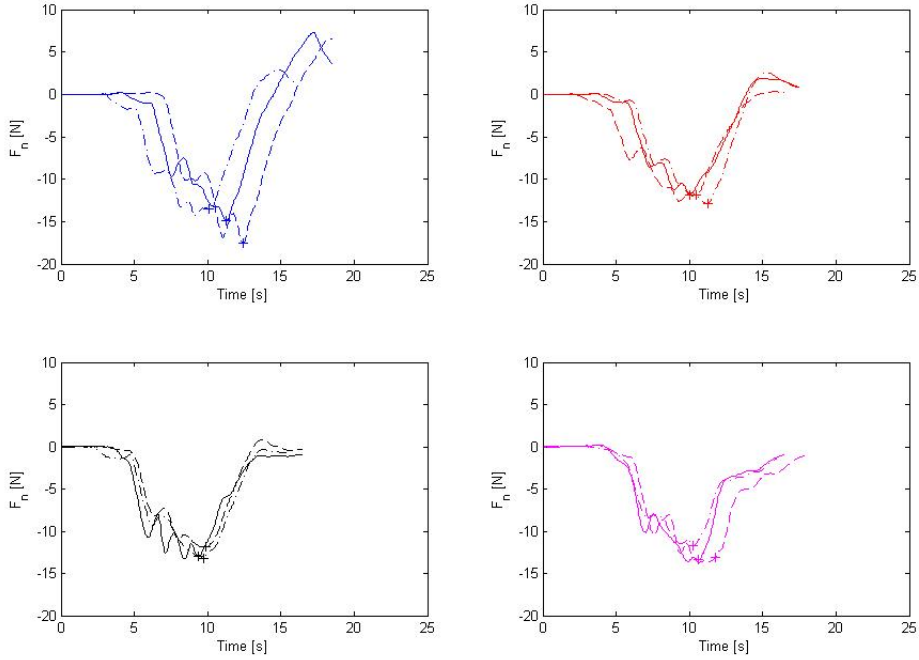


(b) Scotty

Figure 6.12. Determination of divergence time using the standard deviation of the y position (as a function of the time). (a) and (b) correspond to two typical subjects.



(a) Jeffrey



(b) Scotty

Figure 6.13. Force profiles of two typical subjects with force dropping time depicted as '+'.

6.2.2 Data Analysis

First, we consider the standard deviation of all 12 trials of every subject over time, as shown in Figure 6.12. The dotted line in the figure shows the standard deviation of the distance normal to the guiding path. The maximum value of the standard deviation gives us a radius, and the time at which the deviation becomes larger than that radius (before the final divergence) is defined as *divergence time*. The x position corresponding to this time is indicated by a vertical bar in the x-y plot of Fig. 6.14.

For each trajectory, we also consider the *force dropping time* corresponding to the last extremum ('+' in Figure 6.13). The corresponding 12 dropping points are displayed in the x-y plot ('+' in Fig. 6.14) together with their mean position ('□' in Fig. 6.14).

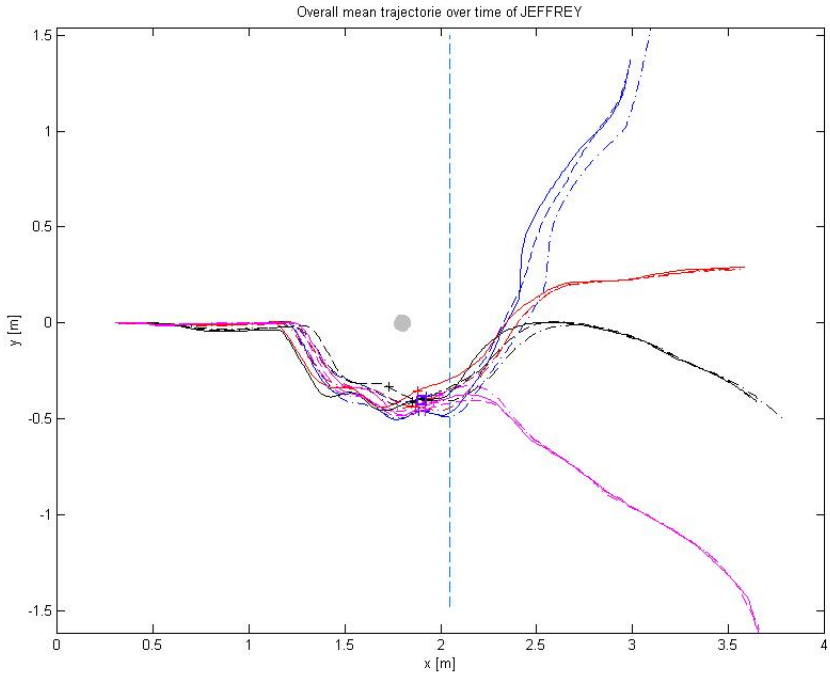
6.2.3 Results

Figure 6.11 shows the trials paths of the same two subjects as previously. We observe that all the paths seem to be roughly similar until they diverge to their respective direction.

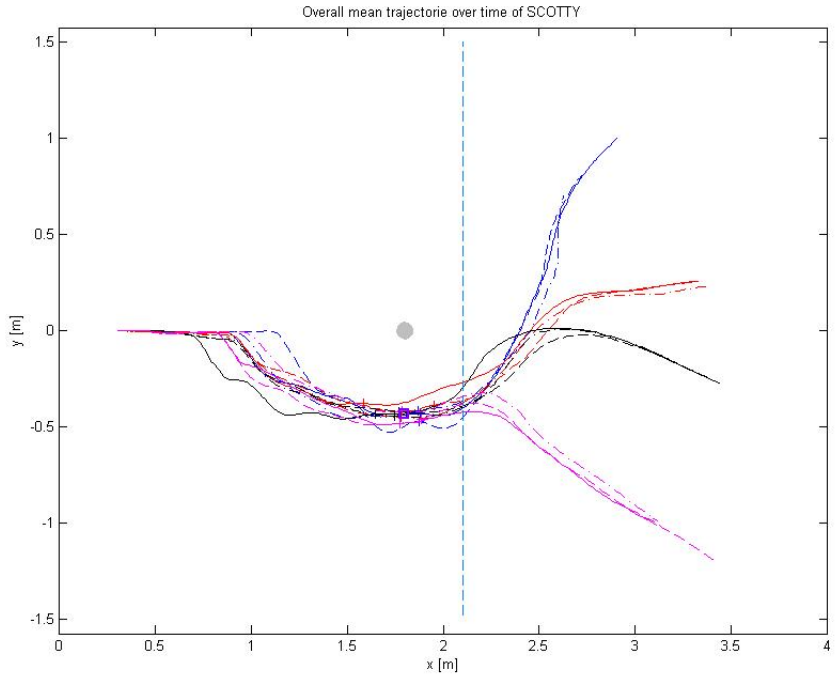
To further analyze this trend, we computed the difference between the x-position of the divergence position and the x-position of dropping point. For each subject this yields 12 differences between the common divergence point and the 12 dropping points corresponding to the trials. Figure 6.15 shows the mean and standard deviation of these 12 differences, for the 7 subjects. Over all subjects, the difference between the divergence point and the dropping point is significantly larger than 0 ($p<0.002$).

To analyze whether the subjects behave differently depending on the final direction of the path, we examined whether the difference between the (x-position of the) divergence point and the dropping point depends on the final direction. For each subject, we computed, for each of the 4 directions, the mean difference over the 3 trials in this direction. Fig. 6.16 shows, for each direction, the 7 differences corresponding to the 7 subjects for all 4 directions. The difference appears to be consistent in all directions. An ANOVA (**AN**alysis **O**f **VA**riance between groups) test confirmed that these differences yields a similar result independent on the direction, as p-value $p=0.9542$ between the 4 differences. This means that the strategy used by the subjects did not depend on the final direction of the desired path.

Finally we analyzed the difference in x-position between the mean dropping point (Fig. 6.17), and



(a) Jeffrey



(b) Scotty

Figure 6.14. Points of dropping force '+' and mean of these points (□) compared with the divergence position represented by the dashed bar. Note that the dropping points is generally slightly before the divergence point and about at the same x-position than the obstacle.

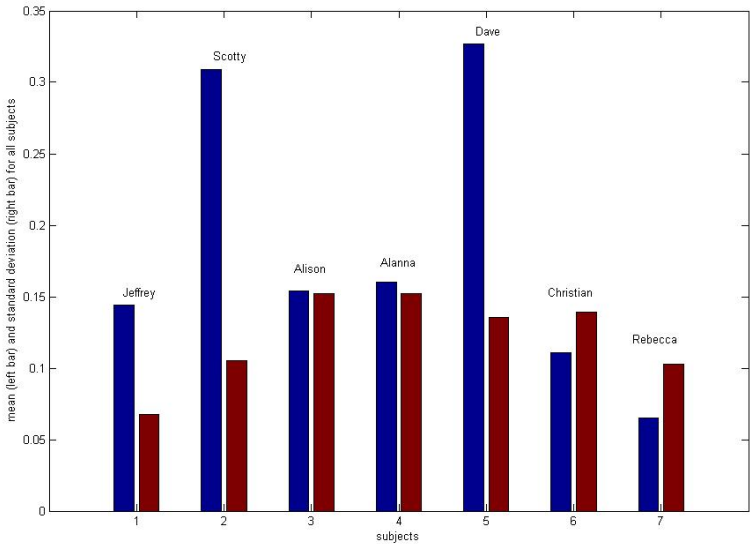


Figure 6.15. Mean and standard deviation of difference between x-position of the divergence point and dropping points of the 12 trajectories, for the 7 subjects.

found that the subjects drop the force approximately when they pass the obstacle.

Altogether, these results suggest that the subjects use a similar strategy independent of the direction. They first deviate a suitable amount to avoid the obstacle. After the obstacle is passed, they release the normal force and trust the path controller, which leads them to the guiding path. They may be able to feel the gradient of attraction from the given guiding path. In turn, these results with randomly chosen path demonstrate that the users have learned working with the elastic path controller independent on the path.

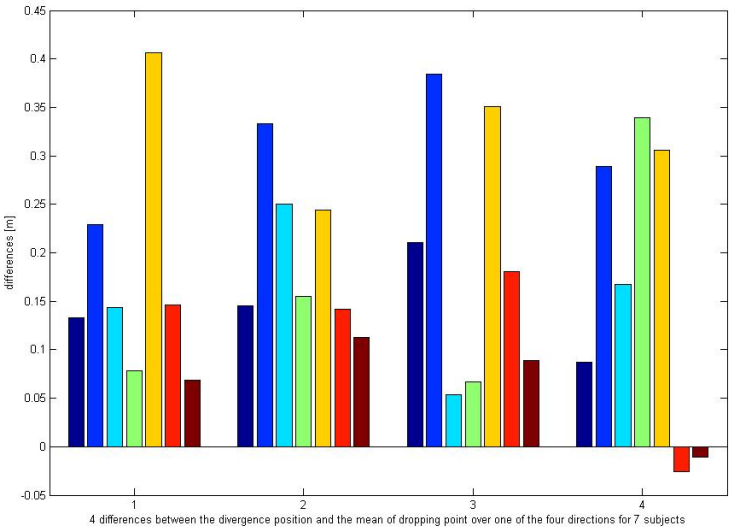


Figure 6.16. Differences between the divergence point and the mean x position corresponding to the dropping time in the four different directions. Each bar corresponds to the difference between the mean x position of over three trials in one direction and the divergence point, for a given subject.

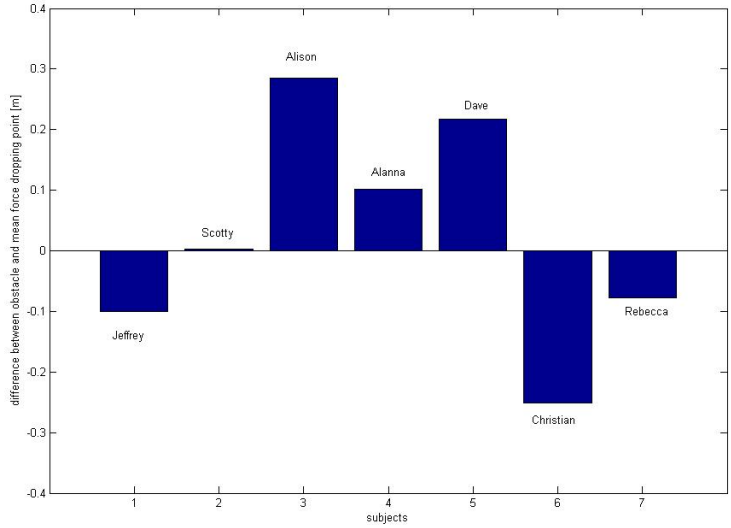


Figure 6.17. Difference in x-position between the mean dropping point and obstacle, for all subjects.

Chapter 7

Conclusion and Future Work

This thesis developed an elastic path controller for collaborative robots. This controller combines the functionalities of path tracking and modification of trajectory, which enables it to compensate for modifications of the environment such as a new obstacle or error of position sensing. Switching between these two modalities is realized in a smooth way by the operator, and requires minimal control.

Such an elastic path controller was derived and tested in simulations on the collaborative wheelchair, and on an implementation on the Scooter cobot. A psychophysical experiment demonstrated that users can learn to use this novel tool in order to modify and design guiding paths in a relatively simple way. Subjects learn to decrease the small oscillations during movement.

This Elastic Path Controller should be implemented and further investigated on the collaborative wheelchair. Also a further work may investigate the rotational elasticity on the Scooter cobot.

While the results of a second psychophysical experiment suggest that the users may feel the attraction from the guiding path, which may help them to manoeuver the cobot, we propose developing a joystick providing feedback of the current condition of the cobot, for example how far it is away from the guiding path. A commercially available force feedback device may be used for this purpose.

Bibliography

[1] http://othello.mech.nwu.edu/∼peshkin/cobot/

[2] http://mathworld.wolfram.com/MonkeySaddle.html

[3] Michael Peshkin, J. Edward Colgate Cobots Industrial Robot, 26 (5), 1999, pp 335-341

[4] Samson,C. Control of chained system: application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control*40, 64-77,1992.

[5] Wannasuphoprasit, W., Akella, P., Peshkin, M. and Colgate, J.E. Cobots: a novel material handling technology *Proceedings of ASME IMECE*, 1998.

[6] Michael A.Peshkin, J.Edward Colgate, Witaya Wannasuphoprasit, Carl A. Moore, R.Brent Gillespie, Prasad Akella Cobot Architecture. *IEEE Transactions on Robotics and Automation*,17(4):377-390.2001.

[7] Witaya Wannasuphoprasit, R.Brent Gillespie, J.Edward Colgate, Michael A.Peshkin Cobot Control. *IEEE Int Conf on Robotics and Automation*,4:3571-3576,1997.

[8] R.Brent Gillespie, J.Edward Colgate, Michael A. Peshkin A General Framework for Cobot Control. *IEEE transactions on Robotics and Automation*,17(4):391-401,2001.

[9] ES Boy, CL Teo and E Burdet Collaborative Wheelchair Assistant *Proc. International Conference on Robotics and Intelligent Systems(IROS)*,2002.

[10] ES Boy, E Burdet, CL Teo and JE Colgate The Learning Cobot *Proc. ASME International Mechanical Engineering Congress and Exposition*,2002.

[11] ES Boy. Experimental Evaluation of the Learning Cobot. Internal report, National University of Singapore, Department of Mechanical Engineering,2001.

[12] ES Boy. Experimental Evaluation of the Learning Cobot. Masters of Engineering thesis, National University of Singapore, Department of Mechanical Engineering,2001.

[13] ES Boy, E Burdet, CL Teo and JE Colgate Experimental Evaluation of the Learning Cobot. *EuroHaptics*,2005.

[14] ES Boy, E Burdet, CL Teo and JE Colgate Motion Guidance Experiment with the Scooter Cobot. *Haptic Symposium, IEEE International Conference on Virtual Reality*,2003.

[15] S.M.Sampei, T.Tamura, T.Itoh, M.Nakamichi. Path Tracking Control of Trailer-Like Mobile Robot. IROS'91, Osaka, November, 1991.

[16] C. Samson, Path-following and time-varying feedback stabilization of a wheeled mobile robot. In *Proc. of the ICARCV 92*, Singapore, 1992, pp. RO-13.1.1-RO-13.1.5.

[17] Carlos Canudas de Wit, Bruno Siciliano and Georges Bastin(Eds) Theory of Robot Control. Springer (Communications and Control Engineering Series), New York, 1996.

[18] Alain Micaelli, Claude Samson Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Tech. Rep. 2097, INRIA-Sophia Antipolis, 1993.

[19] C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sordalen. Nonlinear control design for mobile robots. In *Recent Trends in Mobile Robots*, Yuan Zheng, Ed. 1993, vol. 11, pp. 121.156, World Scientific Series in Robotics and Automated Systems.

[20] K.G.EngelHardt, Roger Wicke, Gregory L.Goodrich, and larry J. Leifer. Evaluation of a robotic aid: From theory to application using an interactive model. In *Proceeding of the 6th Annual Conference on Rehabilitation Enginnering*, pages 279-281,1983.

[21] Vibhu Mittal, Holly A. Yanco, John Aronis, and Richard C. Simpson, editors. *Lecture Notes in Artificial Intelligence: Assistive Technology and Artificial*. Springer-Verlag, 1998.

[22] Debbie Field. Powered mobility: A literature review illustrating the importance of a multi-faceted approach. *Assistive Technology*, 11:20-33, 1999.

[23] Rory A. Cooper. Engineering manual and electirc powered wheelchairs. *Critical Reviews in Biomedical Engineering*, 27(1&2):27-73, 1999.

[24] U. Borgolte, R. Joelper, H. Hoyer, H. Heck, W.Humann, J. Nedza, I. Craig, R.Valleggi, and A.M. Sabatini. Intelligent control of a semi-autonomous omnidirectional wheelchair. In *Proceedings of the 3rd International Symposium on Intelligent Robotic Systems '95*, pages 113-120, 1995.

[25] Ulrich Borgolte, Helmut Hoyer, Christian Bühler, Helmut Heck, and Ralf Hoelper. Architectural concepts of a semi-autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22:233-253, 1998.

[26] Helmut Hoyer, Ulrich Borgolte, and Ralf Hoelper. An omnidirectional wheelchair with enhanced comfort features. In *Proceedings of ICORR '97: International Conference on Rehabilitation Robotics*, 1997.

[27] Christian Bühler, Helmut Heck, and Wolfram Jumann. User-driven human-machine interface configuration for a wheelchair with complex functionality. In *Advancement of Assistive Technology*, pages 375-380. IOS Press, 1997.

[28] Karim A. Tahboub and Harry H. Asada. A semi-autonomous control architecture applied to robotic wheelchairs. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 906-911, 1999.

[29] Jonathan Connell and Paul Viola. Cooperative control of a semi-autonomous mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1118-1121, 1990.

[30] Richard L. Madarasz, Loren C. Heiny, Robert F. Cromp, and Neal M. Mazur. The design of an autonomous vehicle for the disabled. In *Autonomous Mobile Robots: Control, Planning, and Architecture*, pages 351-259. IEEE Computer Society Press, 1991. Originally appeared in *IEEE Journal of Robotics and Automation*, Volume RA-2, Number 3, September 1986, pages 117-126.

[31] Holly A. Yanco and James Gips. Preliminary investigation of a semi-autonomous robotic wheelchair directed through electrodes. In Stephen Sprigle, editor, Proceedings of the Rehabilitation Engineering Society of North America 1997 Annual Conference, pages 414C416. RESNA Press, 1997.

[32] Holly A. Yanco. Shared User-Computer Control of a Robotic Wheelchair System. PH.D Thesis. Massachusetts Institute of Technology, September 2000.

[33] David P. Miller and Marc G. Slack. Design and testing of a low-cost robotic wheelchair prototype. *Autonomous Robots*, 2:77-88, 1995.

[34] Richard Simpson, Daniel Poirot, and Mary Francis Baxter. Evaluation of the Hephaestus smart wheelchair system. In *Proceedings of ICORR '99: International Conference on Rehabilitation Robotics*, pages 99-105,1999.

[35] Richard C. Simpson, Simon P. Levine, David A. Bell, Lincoln A. Jaros, Yoram Koren, and Johann Borenstein. NavChair: an assistive wheelchair navigation system with autonomous adaptation. in Mittal et al.[Mittal et al., 1998], pages 235-255.

[36] Richard C. Simpson and Simon P. Levine. Automatic adaptation in the NavChair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*, 7(4):452-463, December, 1999.

[37] Richard D. Amori. Vocomotion-an intelligent voice-control system for powered wheelchairs. In *Proceedings of the Rehabilitation Engineering Society of North America 1992 Annual Conference*, pages 421-423. RESNA Press, 1992.

[38] N.I.Katevas, N.M.Sgouros, S.G.Tzafestas, G.Papakonstantinou, P.Beattie, J.M.Bishop, P.Tsanakas, and D.Koutsouris. The autonomous mobile robot SENARIO: a sensor-aided intelligent navigation system for powered wheelchairs. *IEEE Robotics and Automation Magazine*, pages 60-70, December 1997.

[39] P.D.Beattie and J.M.Bishop. Self-localisation in the Senario autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22:255-267,1998.

[40] Jill D.Crisman and Michael E.Cleary. Progress on the deictically controlled wheelchair. In Mittal et al.[Mittal et al., 1998], pages 137-149.

[41] Hiroo Wakaumi, Koichi Nakamura, and Takayoshi Matsumura. Development of an automated wheelchair guided by a magnetic ferrite marker lane. *Journal of Rehabilitation Research and Development*, 29(1):27-34, Winter 1992.

[42] John-David Yoder, Eric T.Baumgartner, and Steven B.Skaar. Initial results in the development of a guidance system for a powered wheelchair. *IEEE Transactions on Rehabilitation Engineering*, 4(3):143-151, September 1996.

[43] G.Bourhis and Y.Agostini. Man-machine cooperation for the control of an intelligent powered wheelchair. *Journal of Ingelligent and Robotics Systems*, 22:269-287,1998.

[44] G.Bourhis and Y.Agostini. The Vahm robotized wheelchair: system architecture and human-machine interaction. *Journal of Intelligent and Robotics Systems*, 22(1):39-50, May 1998.

[45] William S.Gribble, Robert L. Browning, Michael Hewett, Emilio Remolina, and Benjamin J. Kuipers. Integrating vision and spatial reasoning for assistive navigation. In Mittal et al.[Mittal et al., 1998], pages 179-193.

[46] R.A. Cooper, K. Prins, T.J. O'Connor, M.L. Boninger, S. Fitzgerald, C. Sunderhaus EVALUATION OF A PUSHRIM ACTIVATED ELECTRIC POWEFLED WHEELCHAIR. *Proceedings of The First Joint BMESEMBS Conference Serving Humanity, Advancing Technology* Oct, l3-l6, '99, Atlanta, GA, USA.

[47] Rory A. Cooper, Thomas A. Corfman, Shirley G. Fitzgerald, Michael L. Boninger, Donald M. Spaeth, William Ammer, and Julianna Arva Performance Assessment of a Pushrim-Activated Power-Assisted Wheelchair Control System. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, VOL. 10, NO. 1, JANUARY 2002.

[48] Eric L. Faulring and J. Edward Colgate RUN-TIME THREE-DIMENSIONAL BLEND-PATH GENERATION FOR COBOT CONSTRAINT SURFACES. *Proceedings of International Mechanical Engineering Congress and Exposition*, November 17-22, 2002.

[49] Michael L. Jones and Jon A. Sanford. People with mobility impairments in the United States today and in 2010. *Assistive Technology*, 8:43C53,1996.

[50] Melvyn Kettle, Corinne Rowley, and M. Anne Chamberlain. A national survey of wheelchair users. *Clinical Rehabilitation*, 6:67C73, 1992.

[51] E Burdet and TE Milner. Quantization of Human Motions and Learning of Accurate Movements. *Biological Cybernetics* 78(4): 307-18, 1998.

Appendix A

Coordinates Transformation

$$\left(\frac{d\overrightarrow{OM}}{dt}\right)_R = \left(\frac{d\overrightarrow{OP}}{dt}\right)_R + \left(\frac{d\overrightarrow{PM}}{dt}\right)_T + \vec{w}_c \times \overrightarrow{PM}$$

$$\left(\frac{d\overrightarrow{PM}}{dt}\right)_T = \left(\frac{d\overrightarrow{OM}}{dt}\right)_R - \left(\frac{d\overrightarrow{OP}}{dt}\right)_R - \vec{w}_c \times \overrightarrow{PM}$$

Multiplied R_T^R at both sides, we get:

$$\left(\frac{d\overrightarrow{PM}}{dt}\right)_T = R_T^R \left(\frac{d\overrightarrow{OM}}{dt}\right)_R - R_T^R \left(\frac{d\overrightarrow{OP}}{dt}\right)_R - R_T^R(\vec{w}_c \times \overrightarrow{PM})$$

Because $R_T^R \left(\frac{d\overrightarrow{OP}}{dt}\right)_R$ means the transformation of the point P from frame R to frame T, and because P is the origin of frame T, so $R_T^R \left(\frac{d\overrightarrow{OP}}{dt}\right)_R = 0$

And

$$R_T^R \left([\vec{w}]_R \times \left[\overrightarrow{PM}\right]_T\right) = \begin{bmatrix} 0 \\ y \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_c = c_c(s)\dot{s} \end{bmatrix} = \begin{bmatrix} -c_c(s)y\dot{s} \\ 0 \\ 0 \end{bmatrix}$$

So we can get the relationship between two coordinates as:

$$\begin{bmatrix} \dot{s} \\ \dot{y} \\ 0 \end{bmatrix} = R_T^R \begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix} + \begin{bmatrix} c_c(s)y\dot{s} \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{s} \\ \dot{y} \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{bmatrix} + \begin{bmatrix} c_c(s)y\dot{s} \\ 0 \\ 0 \end{bmatrix}$$

Expand the function above, we have have the function about $\dot{s}$ and $\dot{y}$:

$$\begin{cases} \dot{s} = (\cos\theta_c \quad \sin\theta_c) \cdot \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} / [1 - c_c(s)y] \\ \dot{y} = (-\sin\theta_c \quad \cos\theta_c) \cdot \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} \end{cases}$$

Appendix B

Coordinates Transformation 2

To derive the control variable, we express the equations of motion w.r.t the new variables $\eta = \int_0^t |\dot{s}|\, d\tau$ instead of the time-index t. This new variable has the physical meaning of the distance travelled by the vehicle along the path.

$$\frac{ds/dt}{dy/dt} = \frac{v\cos\theta/1 - c_c(s)y}{v\sin\theta} = \frac{\cos\theta}{\sin\theta[1 - c_c(s)y]} = \frac{ds/d\eta}{dy/d\eta}$$

Denoting $\dfrac{d}{d\eta}$ as ( )', $y' = s'\tan\theta[1 - c_c(y)]$

And $\quad \because \quad \dot{s} = \dfrac{v\cos\theta}{[1 - c_c(s)y]} \quad \therefore \quad s' = sign\left[v\dfrac{\cos\theta}{1 - c_c(s)y}\right]$

$\therefore y' = \tan\theta[1 - c_c(s)y]sign\left[v\dfrac{\cos\theta}{1 - c_c(s)y}\right]$

Similarly,we can have:

$$\frac{\theta'}{s'} = \frac{d\theta/d\eta}{ds/d\eta} = \frac{\dot{\theta}}{\dot{s}} = \frac{w[1 - c_c(s)y]}{v\cos\theta} - c_c(s)$$

$\therefore \theta' = s'\left\{\dfrac{w[1 - c_c(s)y]}{v\cos\theta} - c_c(s)\right\} = sign\left[v\dfrac{\cos\theta}{1 - c_c(s)y}\right]\left\{\dfrac{w[1 - c_c(s)y]}{v\cos theta} - c_c(s)\right\}$

After simplifying,

$$\theta' = \frac{w|1 - c_c y|}{|v\cos\theta|} - c_c sign(v\frac{\cos\theta}{1 - c_c y})$$

Finally, the kinematics model of unicycle-type vehicle can be expressed as below in term of the distance travelled by the robot:

$$\begin{cases} s' = sign(v\dfrac{\cos\theta}{1-c_c y}) \\ y' = \tan\theta(1-c_c y)sign(v\dfrac{\cos\theta}{1-c_c y}) \\ \theta' = \dfrac{w|1-c_c y|}{|v\cos\theta|} - c_c sign(v\dfrac{\cos\theta}{1-c_c y}) \end{cases}$$