A Model-Based Approach to Analaysis and Calibration of Sensor-Based Human Interaction Loops

Parisa Eslambolchilar[*,1] , Roderick Murray-Smith[2]

[1]*Computer Science Department, Swansea University, Swansea, SA2 8PP, UK*
[2] *Department of Computing Science, Glasgow University, Glasgow, G12 8QQ, Scotland*

Abstract
The dynamic systems approach to the design of continuous interaction allows designers to use analytical tools such as state-space modeling and Bode diagrams to simulate and analyse the behaviour and stability of sensor-based applications alone and when it is coupled with a manual control model of user behaviour. This approach also helps designers to calibrate and tune the parameters of the sensor-based application before the actual implementation, and in response to user action. In this paper we introduce some term definitions from manual control theory for the analysis of the continuous aspects of the interaction design and human behaviour. Then we provide a theoretical framework for specification, analysis and calibration of a sensor-based zooming and scrolling application on mobile devices including the user in the interaction loop. It is especially topical and interesting for guiding design of sensor-based applications on mobile devices. We test our framework with a tilt-controlled speed-dependent automatic zooming application on a PDA.

Key words: Dynamics, Human operator modeling, Mobile devices, Human-Computer Interaction

1 Introduction

What distinguishes interactive systems from other classes of computing systems is the user, and the general focus of research in interactive systems has been the need to accommodate the user, and specifically the "usability" of the system. One area of research within this has been concerned with the development of models of interactive systems, and sometimes of the user, in order to analyse the behaviour of the user and the system [23].

With the increasing popularity of mobile phones and in general handheld devices in recent years, more and more computers are being used in a mobile environment. Millions of people who use mobile phones carry them everywhere in their hand, pocket and bag. For many of us these devices are not perceived as computers, but rather as augmented elements of the physical environment [36]. Therefore, interaction shifts from an explicit paradigm, in which the user's attention is on computing, towards an implicit paradigm, in which interfaces themselves drive human attention when required [33].

Nowadays interaction with handheld devices is not limited to using the keyboard or touch-screen and, traditional interaction design methods based on WIMP (Windows-Icon-Menu-Pointer). These devices are now able to accept input and provide output via other means than WIMP. As topical examples, on the iPhone and Nokia N-series, the user can rotate the screen view from landscape to portrait and vice versa by rotating the device. Other means of interaction

[*] Corresponding author.
Email addresses: p.eslambolchilar@swansea.ac.uk (Parisa Eslambolchilar),
rod@dcs.gla.ac.uk (Roderick Murray-Smith).

with mobile devices are gesture input and audio/haptic output. These can facilitate one-handed control, which requires less visual attention than two-handed touch-screen control, and that multimodality in the interaction can compensate for the lack of screen space [12,15,20,27,32,38]. In such novel interaction techniques, i.e., gesture recognition, audio/haptic feedback, continuous interaction is at the heart of the interaction between the human and sensor-based application; because the human is tightly coupled to the application via interaction over a period of time and exchanges continuous input/output of dynamic information at a relatively high speed with the application, this cannot be modeled as a series of discrete events and static models [10].

The main contribution of this work is to develop a theoretical framework for specification, analysis and calibration of sensor-based applications on mobile devices without excluding the user from the interaction loop. It is especially topical and interesting for guiding design of sensor-based applications on mobile devices. The issue was motivated by analysing an interaction technique called speed-dependent automatic zooming (SDAZ) [22], which in previous research has been found to outperform manual zooming approaches on desktop computers [7, 8]. SDAZ unifies rate-based scrolling and zooming to overcome the restrictions in screen space for browsing images and texts. The user controls the scrolling speed only, and the SDAZ application automatically adjusts the zoom level so that the speed of visual flow across the screen remains constant. Using this technique, the user can smoothly locate a distant target in a large document without having to manually interweave zooming and scrolling, and without becoming disoriented by extreme visual flow.

There is already a range of existing frameworks for designing interaction such as participatory design, ethnography, and emerging inspirational methods available to provide guidelines for the calibration of interfaces, i.e., predefined constants governing interface behaviour [3,4,5]. However, none of these is in itself sufficient to address the problem. First, they have not been focused on the specific challenges raised by working with imprecise sensor technologies, i.e. current frameworks suffer from several shortcomings in terms of efficiency, effectiveness and user satisfaction. Second, they do not take into account novel input/output technologies and human behaviour model. The absence of formal guidelines means that designers are forced to adjust the properties of the interaction by trial and error. For instance, coupling the scrolling to the zoom level and controlling both via tilt input on small screen devices with present techniques requires many experiments to customise SDAZ application behaviour [7,8].

This research shows that the dynamic systems approach allows us to (a) analyse the behaviour and stability of the sensor-based applications when it is coupled with a manual control model of the user behaviour and, (b) calibrate and tune the parameters of the application based on human operator model. In the following sections we will explain the key definitions in manual control theory, which are important in modeling human behaviour in the interaction. Then we will show that this approach to the design of a continuous interaction interface allows the incorporation of analytical tools to analyse the sensor-based application behaviour when coupled with a manual control model of user behaviour. We also demonstrate calibration and tuning of the parameters and stability testing of the tilt-controlled SDAZ running on a PDA while coupled with a human user.

2 Background

In the past ten years many researchers have focused on sensor-enhanced applications in mobile-human interaction. The results of these studies have shown that gesturing with a handheld device can be used for scrolling, selection, and commanding an application without resorting to buttons, touch screens, spoken commands or other input methods [1, 15, 20, 21, 25, 32]. These sensors provide a tight coupling between the user and the application based on a continuous input/output exchange of dynamic information, which happens over a period of time. We cannot model this coupled human-application interaction as a series of discrete events and static models [10,14]. We include dynamics because feedback from the display (visual, haptic or audio) influences our actions and changes our perceptions. Thus we control what we perceive.

William Powers was one of the explorers of the idea of continuous dynamic interaction in the 60's [30,31]. His research suggested that many kinds of behaviour can be described as continuous control problems. This viewpoint provides an empirical method for the estimation of a subject's intention. Powers gave several examples which show that for identifying controlled variables in an interaction we can apply disturbances, directly or otherwise, to variables which are under the user's control. If these variables are corrected by the user after applying disturbances, then those variables are assumed to be controlled.

Currently, there are few theoretical frameworks for interaction. For example, a few theories in psychology which provide insights into human behaviour have also been applied in designing interfaces (e.g., Fitts' law). GOMS (Goals, Operators, Models and Selections) proposed by Card et al. [6] considers only user's cognitive skills in the interaction. Also, there are many physiological models of human body motion [34].

There are few recent frameworks which focus on human error in the interaction with sensor-based applications: Hinckley et al. [20,21] showed the importance of consideration of errors for foreground and background interaction, as well as transitions between grounds. They described architectures and techniques to infer attention and location via integration of sensed events (touch, tilt, proximity, and microphone), without introducing unwanted cross-talk between techniques. They contributed a number of examples of what is possible, and suggested some general design principles. However, careful experiments were required to quantify user performance with sensing techniques, as well as longitudinal studies to determine the usability of the application. Benford et al. [4] introduced a framework that encourages designers to tackle the problem of matching physical form to the capabilities of sensors and the shifting requirements of applications by analysing and comparing expected, sensed, and desired movements. This framework focused on the boundaries between these, drawing on analytic approaches, and treating mismatches as opportunities as well as problems. The authors emphasised that their examples show that the framework has the potential to support both the analysis of designs and the generation of new ideas. However, there is no guarantee that the resulting ideas are good ones. Also, the framework has not been examined on a broad set of mobile device applications.

Although dynamic systems theory has the great potential to provide a solution for qualitative but meaningful characterisation of the interactive dynamic behaviours of the application and user, it has been overlooked in HCI research.

2.1 Designing Continuous Interaction and Manual Control

A branch of control theory that is used to analyse human and system behaviour when operating in a tightly coupled loop is called *manual control theory* [23,29]. The theory is applicable to a wide range of tasks, for example, tracking of targets, and creates a framework for modeling dynamic systems. The general approach followed in manual control theory is to express the dynamics of the combined human and controlled element behaviour as a set of linear differential equations in the time domain, called *state-space* modeling. A state-space representation is the mathematical realisation of control theory. This representation provides a convenient and compact way to model and analyse applications with multiple inputs and outputs. Also, it can incorporate sensor noise, disturbance rejection, sensor fusion, changes in input/output devices, and calibration challenges.

Using the continuous control dynamic system approach and manual control theory we can *simulate* the model and observe the behaviour of the application as well as the user. This approach makes tuning and calibration a lot easier, especially when we use an input with more than one degree-of-freedom (DOF), because we can find proper settings for the interface only by observing the behaviour of the simulated coupled user and application before the actual implementation. Many successful computer interfaces have been designed and developed based on many experimental tests over a considerable amount of time but there is no solid and falsifiable theory to generalise those experimental results even to similar interfaces [2,37]. Additionally, several models include human related aspects of information processing explicitly such as delays for visual process, motor-nerve latency and neuro-motor dynamics. Control theory can be linked to Fitts' Law [16] by viewing the pointing movements towards the target as a feedback control loop based on visual input and the limb as a control element allowing most of Fitts' law results to be predicted by a simple control theory [23].

Doherty et al. [9] examine how they can apply manual control concepts in a qualitative fashion to the design and analysis of interactive systems. This involves a focus on control and feedback signals, transformations of these, and control characteristics of user, device and controlled process. Doherty et al. [9] make reference to the particularly challenging application area of performance control systems for disabled musicians. They believe that control issues of this nature will become increasingly common in audio/haptic interface design. In [10,11] authors examine the applicability of classic manual control in the area of human computer interaction in magic board interface. They show that it is possible to develop a methodological approach to the application of dynamic systems theory in HCI, incorporating both device level and human behaviour models. Williamson and Murray-Smith [39,40] present a few example interfaces built on methods from perceptual control theory and dynamic systems. For example, they provide a method for performing selection tasks based on the continuous control of multiple, competing agents who try to determine the user's intentions from their control behaviour without requiring an explicit, visible pointer. This is an example of work on use of models for predicting the usability. Eslambolchilar and Murray-Smith [13] provide a dynamic systems interpretation of the coupling of internal states involved in speed-dependent automatic zooming, and test their implementation on a text browser on a Pocket PC instrumented with a tilt sensor. A thorough exploitation of dynamic systems approach in this work presents a promising theoretical framework for designing interaction models and creating better interactive systems on portable computing devices.

In the following sections, we first focus on manual control theory as a formal modeling approach to provide appropriate concepts to deal with issues of human operator modeling. Then, we explore the dynamic systems approach in designing interaction for an example sensor-based application, tilt-controlled SDAZ on a mobile device. Finally, we analyse the performance of the tilt-controlled application with different calibrated and tuned values for coefficients once with excluding the user from the interaction loop and the next with including them. We show that (1) the closed loop of tilt-controller + device might be stable and controllable but the combined human + tilt-controller + device might be unstable; (2) the time delay and lead-lag-dynamics of typical human control behaviour can be modeled, simulated and analysed in the combined closed loop of human + controller + computing device in the Bode space before the actual implementation; and (3) tuning and calibrating coefficients and parameters can be done in the Bode space not in common trial and error experiments.

3 Human Operator Modeling

Manual control is the study of humans as operators of dynamic systems such as a gesture controlled applications. Early research focused on the human element in vehicular control [10]. Designers of these systems realised that the human was an important element in the system control loop. In order to predict the stability of the full system, they had to include mathematical descriptions of the human operators along with the descriptions of the vehicle dynamics. However, the applications of manual control theory has been modified and extended to applications in HCI [10,11]. Humans are regularly asked to position the cursor on a menu, drag the scrollbar and other tracking and positioning tasks. Thus, the human operator can be modeled using the tools of manual control theory such as Bode analysis (see Section 3.5). It has further benefits: "*First, quantitative models of the human operator may provide insights into basic properties of human performance*". "*Second, the ability to derive transfer function for human operators would greatly facilitate the ability to predict the performance of human-machine systems*"[23, p. 158].

Gaines [17] provides a complete survey of linear and nonlinear models of human behaviour. Definitive evidence has been gathered of discrete-action and discrete-time phenomena in peripheral behaviour such as hand and eye movements, and continuous-action and continuous-time phenomena in peripheral behaviour such as gesture-based interaction. These types of behaviour have been classified as nonlinear behaviour. [17,23] argue that although human behaviour is mostly nonlinear, linear analysis still provides important insights into human performance and linear models may be able to give reasonable predictions for some situations[1]. Much research has been done to develop a quasi-linear model of the human operator (Figure 1). "*The quasi-linear model is an attempt to represent the human operator as a constant coefficient linear differential equation [...] plus internal noise which is assumed to arise from perceptual or motor processes internal to the human operator*" [23, p.159].

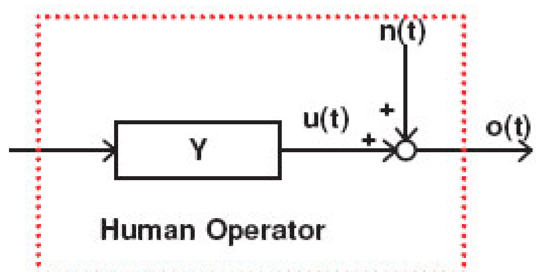


Figure 1: A quasi-linear model of the human operator. Y_h is the linear transfer function; *u(t)* is the linear response; *n(t)* is internal noise (reflected noise in the perceptual and motor systems of the operator); and *o(t)* is the quasi-linear response. The noise is generally presumed to be uncorrelated with any input signal. Adapted from [23].

[1]The term linear function here means a first-degree polynomial function of one variable. These functions are called "linear" because they are precisely the functions whose graph in the Cartesian coordinate plane is a straight line. A non-linear function is defined as a polynomial function of degree 2 or higher. These functions are called "nonlinear" because their graph in the Cartesian coordinate plane is not a straight line.

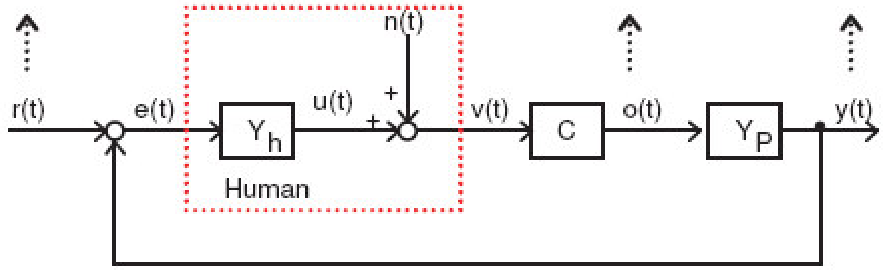


Figure 2: A typical 1D compensatory tracking task. Adapted from [23].

Figure 2 illustrates a typical 1D tracking experiment e.g. positioning a cursor (controlled by mouse/joystick) on a target displayed on a screen. The human operator, represented as Y_h and *n(t)*, is instructed to follow a quasi-random input signal, *r(t)*. The error, *e(t)*, is displayed in a compensatory tracking task[2]. Control responses, *o(t)*, are typically made with a mouse or joystick or in general a controller, *C(t)*, and these control responses are input to Y_p, a computer, for example. The output of the system, *y(t)*, is the response of the computer. The human transfer function is linked to the device transfer function, Y_p, such as a mobile device and the input device, *C(t)*, such as a tilt input via the feedback loop. Depending on how well the human transfer function is described, the outcome of tracking tasks, for example, will change dramatically (see Section 3.5).

To build a describing function for the human operator "*Bode analysis*" is a strong candidate because it is possible to create the human transfer function, Y_h, from the patterns in the Bode space. Before describing function in Bode diagram we would like to introduce some keywords which will be used regularly throughout this work.

3.1 Gain and Time-delay

A closed loop negative feedback system's behaviour is based on two parameters, firstly a delay or latency τ which is the time taken by the controlled element to react to its input, and secondly the gain K which determines the rapidity of adjustment. When K is low the system responds very sluggishly, moving only slowly towards the target signal, and when K is high the system oscillates. The delay τ can also affect the system behaviour – a high delay makes oscillatory behaviour much more likely. For example, in a concert where the music is output, τ must be low (of the order of 20ms) to produce a perceived immediate output. If τ is much higher, most musicians become unwilling to perform [10].

3.2 Order of Control

Order of control is described as "*dynamic relation between displacement of a control device*" (e.g. a mouse, tilt sensor, joystick) and the "*behaviour of the system being controlled.*" Usually the

---

[2]In closed-loop control both the target and output signal (fed back) are available, giving the user the opportunity to compensate for error. In a target acquisition task for example, the target signal is the desired position on the screen; the fed-back signal is the current position of the mouse cursor.

number of integrations between the input and output of the control system specifies the order of control [23, p. 89].

A system with no integration between input and output is a zero-order system. Such a system (also called position control system) provides a proportional relationship between the displacement of the input and the output of the system (Figure 3). For example, mouse controls typically employ a zero-order of control such as manipulation of the scroll-thumb. The mapping is from position to position and as the mouse position changes the scroll position changes proportionately (control-display gain determines the magnitude and acceleration of the mapping).

A system with one integration between control input and output is a velocity control system. Thus, there is a proportional relationship between the displacement of the input and the velocity of the output (Figure 4). Similar to position control, the gain of the velocity control system determines the proportionality between the position of input and the velocity of the output. When the input is stopped in any position (not null), the output continues in motion at a velocity proportional to the displacement from the null position. Spring-centred joystick and in general input devices that have a well-defined null or zero position employ velocity control. The most important advantage of a velocity control is that it allows a limitless range of motion on the output even when the range of the input motion is limited. However, position control requires a limitless range of the input motion for a limitless range of the output motion.

A system with two integrations between control input and system output is called an acceleration control. A second order system provides a proportional relationship between the displacement of the input and the acceleration of the output (Figure 5). Second-order systems are more difficult to use than either zero- or first-order control systems; because the "*reversal of the input must be made in anticipation of the final stopping position*." [23, p. 89] However, practice makes most people become skilled at using these systems. This dynamic is typical of vehicular control and video games which simulate vehicles.

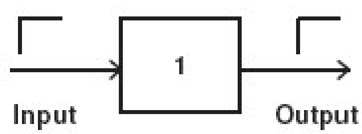


Figure 3: A zero-order system. The proportional relation between input and output is determined by the gain. Here the gain is equal to one.

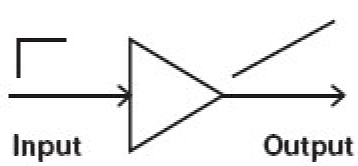


Figure 4: A first-order system

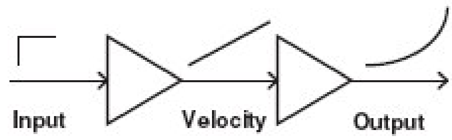


Figure 5: A second-order system

3.3 Control Order and Design Issues

"*In a target acquisition task, the goal is to move a control system output into alignment with a fixed or moving target.*" [19, p.2] In this task, the final position of the system output (e.g. cursor) is more important than the method is used to reach the target. For example, moving a cursor to a menu option (menu selection) is an example of a target acquisition.

In target acquisition tasks the target can be either stationary or non-stationary (moving target). For example, in acquiring a target with a fisheye lens [18], moving and positioning the pointer relative to the underlying data can be difficult, since the data appears to move in the opposite direction of the moving focus point. This effect has been shown to cause significant problems in targeting tasks [18]. Jagacinski et. al [24] found that velocity control had higher performance than position control in capturing small fast moving targets. However, in evaluating and comparing particular control input devices we should consider those devices which have same dynamics; because dynamic aspects of the designed controller are sometimes independent from the task they are going to be used and tested for. For example, comparing a mouse with a joystick in a target acquisition task may produce confusing results; because the dynamics are different.

Scrollbars include both a position and a velocity control. Clicking on one of the up/down arrows in the scrollbar with the mouse cursor moves the text up or down at a constant rate, and thus is a velocity control system. However, the mouse cursor can be placed directly over the scroll handle, and drag it to the desired position in the document. When the position is acquired the handle is released and the text is then updated on the screen. This mode is a position control. However, in browsing long documents, this position control can become very sensitive because many pages map into a limited range of control movement. In this case, the handle position control is commonly used for approximate positioning, and the less sensitive velocity control arrows are used for fine positioning.

The difficulty in learning and using a control system increases with increasing order of control. For most human-computer interfaces, a position or velocity control results in the best target acquisition performance.

3.4 Performance Measures

"*The performance criteria are typically expressed as a function to be minimised.*" [23, p.202] The differential and algebraic equations in the state-space model can be used to determine the path through state space (or the control law) that minimises (or maximises) the performance criterion. Typical criteria include time, distance, or resource consumption.

In Fitts' law experiments, the goal is to capture the target in minimum time, i.e. the time the user initiates the movement and completes the task should be minimised. Thus, the performance in this positioning task can be described as minimising the function [23, p.202]:

$$J=|t_f-t_0| \qquad (1)$$

t_0 and t_f are initial and final time in Fitts' Law task respectively.

One method that can be used to choose the proper coefficient settings in the state-space model is measuring the user's activity via a cost function. Thus, the function to be minimised in this targeting task would be:

$$uf = filter(u)$$

$$J_t = |t_f - t_0| \quad (2)$$

$$J_s = \sum_{t=t_0}^{t_f} |uf_{t+1} - uf_t| \quad (3)$$

$$J_a = \frac{\sum_{t=t_0}^{t_f} |uf_{t+1} - uf_t|}{t_f - t_0} \quad (4)$$

t_0 and t_f are initial and final time and uf_t is the user's input to the system, which has been filtered. Equation (2) presents the total time of the completing the task as a performance measure. Thus, when the user is comfortable with the system while interacting s/he should complete the task in the minimum time. Similarly, in equations (3) and (4) the total sum of changes or mean sum of changes in the input should be minimised if the interaction is smooth [23, p.202].

3.5 Describing Functions in Bode Diagram

As presented in Section 3 the human operator, the plant (e.g. a computing device) and the input device are tightly coupled together in the closed-loop of interaction. In modeling interaction such as tracking a target on a screen, the accuracy of the human operator transfer function plays an important role in the accuracy of the tracking task [23]. To build a describing function for the human operator "*Bode analysis*" is a strong candidate because it is possible to create the human transfer function, Y_h, from the patterns in the Bode space. The Bode plot is used widely to visualise Bode analysis. This plot provides useful information about both: how a given application running on a computing device will behave when a stream of input is provided and, how a particular human + controller + computing device combination will behave [28]. The Bode plot presents the change in amplitude (output amplitude/input amplitude) and the phase shift (output phase-input phase) that is produced by a particular linear system (human + controller + computer). The Bode plot graphs the amplitude change referred to as the gain curve, in a log-log space. The power of the amplitude change, referred to as gain curve, in log-log space. The power of the amplitude ratio (i.e., magnitude squared) is plotted in decibels ($10log_{10}$). This is plotted against log frequency (in radians/s). The phase shift is plotted in degrees against log frequency (radian/s).

Figure 6 presents a Bode diagram example that illustrates typical results of a one-dimensional compensatory tracking study. In this study the computing device was a simple gain ($Y_p = 4$). That is, the response of the simulated computing device was proportional to the control input – it was a zero-order control system. First, note the amplitude ratio presents a 10 dB/decade slope (10 dB/log_{10} Hz) at high frequencies; this is a characteristic of integration. The phase response goes down continuously with increased frequency suggesting a time delay. Thus, the human transfer function can be a gain, a lag, or an integrator at higher frequencies, and a time delay [23, p.161].

$$Y_h(j\omega)=\frac{Ke^{-j\omega\tau}}{j\omega} \qquad (5)$$

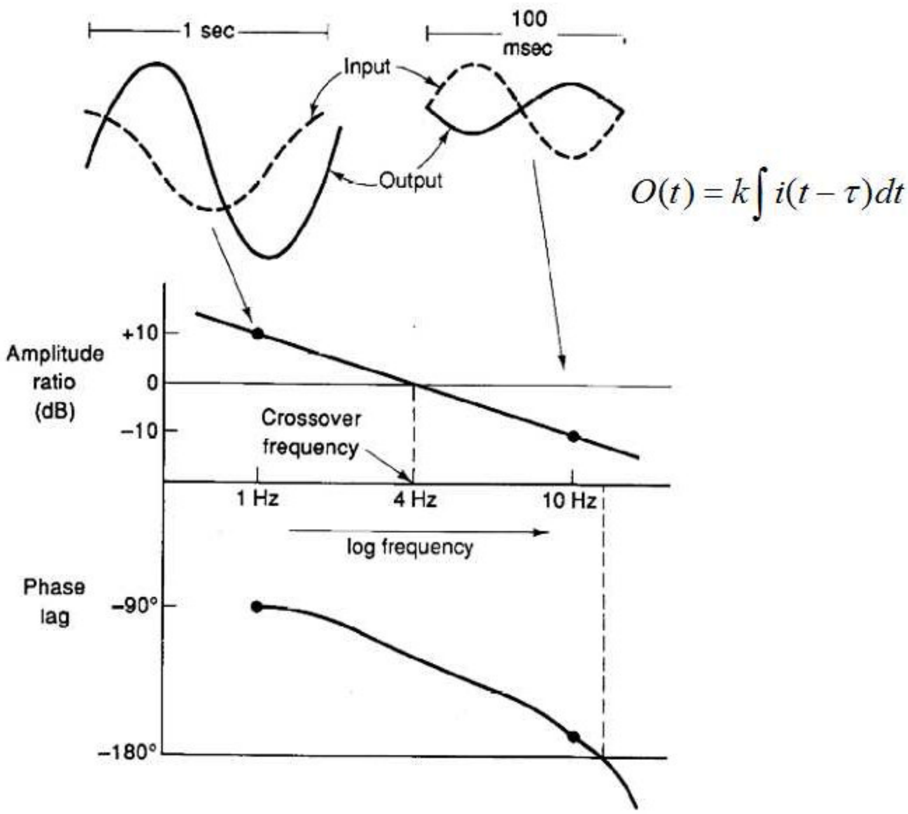


Figure 6: Top– A Bode plot representation of a first-order lag with gain *k* and time delay τ. Bottom–The frequency response for a human operator controlling a zero-order system ($Y_p = 4$)

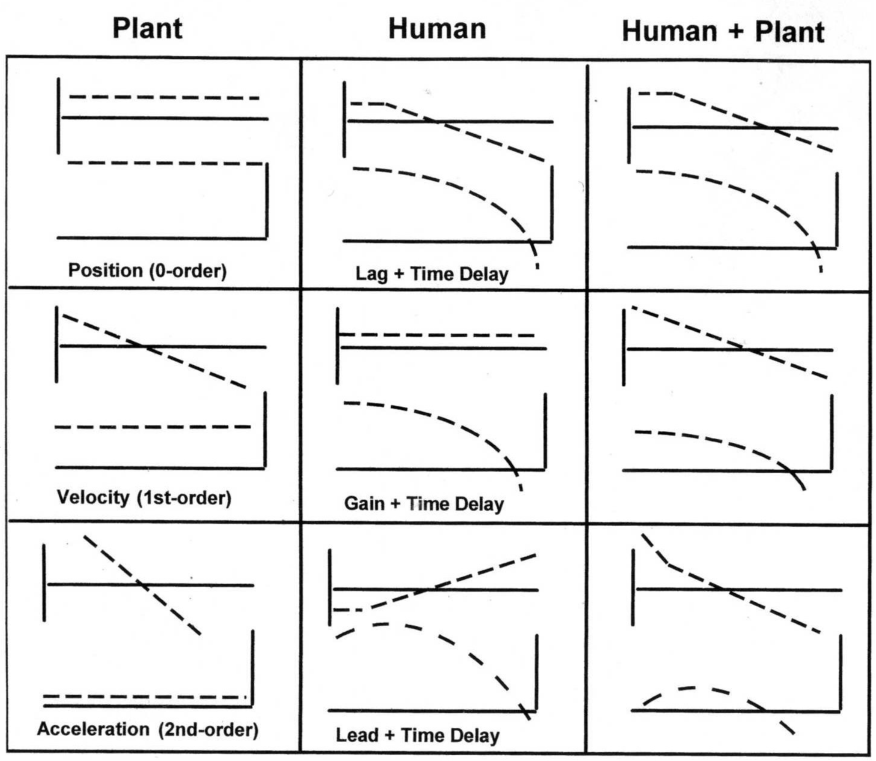


Figure 7: The schematic illustrations shows the adaptive nature of the human controller. The human transfer function (amplitude ratio and phase shift) changes depending on the system being controlled. Adapted from [23].

The gain K is a scaling factor that influences the bandwidth of the control system. The time delay τ reflects human reaction time. In simple tracking tasks the range of the time delay is between 20 ms to 150 ms, which overlaps with measures of reaction time in response to continuous stimuli (refer to Section 3.1). The lag $\frac{1}{j\omega}$ suggests that the human tracker has a low pass characteristic – that is, the human responds to low frequency components of errors and ignores (or filters out) the high frequency components of error [26].

Figure 7 illustrates Bode plots of the human controller adapting to the computing device dynamics [35]. For example, humans adjust their gain to compensate increases or decreases in the device gain, so the total open-loop gain in Figure 2 remains constant in a way that reflects the constraints on stable control [23, p.165]:

$$Y_h(j\omega)C(j\omega)Y_p(j\omega)=constant \quad (6)$$

However, the human operator model is different for each different device dynamic. Figure 7 shows that when the device is a zero-order system, the human looks like an approximate integrator or lag, in another case when the device is a first-order system the human looks more like a gain, and in the last case when the device is a second-order system the human looks more like an approximate differentiator or lead. In all cases, a time delay is evident. In all three cases, the transfer function for the forward loop $Y_h C Y_p$ (see Figure 2) looks similar, i.e., approximately like a gain, a time delay, and an integrator in the region of the crossover[3] [23, p.165]:

$$Y_h(j\omega)C(j\omega)Y_p(j\omega)=\frac{\omega_c e^{-j\omega\tau}}{j\omega} \quad (7)$$

Note that at the crossover frequency, $\omega=\omega_c$, the net gain of human + controller + computing device is 1.0, so the open-loop gain parameter in the numerator is equal to ω_c. Furthermore, in designing the controller, $C(j\omega)$ should be chosen such that:

- The closed-loop system should remain stable;
- The phase margin and amplitude ratio should be maximised;
- The cost function, J, should be minimised [23, 28, 35].

The control loop between perception and action is almost always closed through an environment [30]. So, constraints such as stability reflect global properties of this control loop. "*Models of behaviour are likely to include terms like the control strategy in the crossover model, whose parameters depend on the task context. It suggests that behaviour is "situated". Behaviour is just an adaptive response to situation constraints*" [23, p.165].

3.6 "Bang Bang" Models of Human Controller for High-Order Systems

In Fitts' law tasks, where the user is asked to move the cursor from a starting position to a target area, the hand is also moving from one position to another, or in rotating the eye from one fixation to another, the human operator has to vary the location of a mass using the force exerted by his muscles which is limited in its maximum value. Dynamically, the hand or eye is virtually a pure mass, with low dissipation of energy through friction, and low storage of potential energy through spring-like behaviour. A simple servomechanism, in controlling the location of an object, applies a force to it proportional to the deviation of the location from the desired one, in such a direction as to reduce the deviation [17]. Figure 8 illustrates an example of a discrete style of control that might be used to point a target. The switching boundaries are set at constant ratios of position and velocity. From an initial position a thrust command (i.e., thrust generated by mouse movement) causes the pointer to accelerate toward the target. The first diagonal boundary occurs where the thrust command is terminated and allows the pointer to coast toward the target at a constant velocity. For many control situations (e.g. stopping at a target on the screen) the coast region may show some deceleration due to friction drag. The second diagonal boundary happens, where a reverse thrust is initiated, causing the craft to decelerate as it approaches contact with the target [23].

[3]The point where the open-loop response goes through the zero db is referred as crossover frequency. The crossover frequency ω_c is a measure of the dynamical quality of the control loop. The higher ω_c the higher the bandwidth of the closed loop, and the faster the reaction on command inputs or disturbances [28].

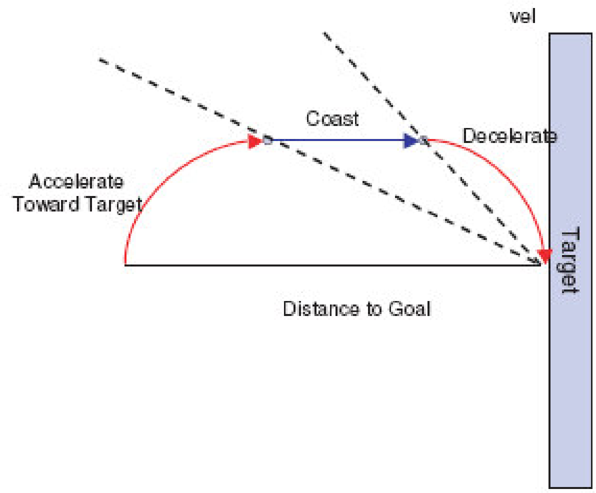


Figure 8: A finite state controller. The switching criteria are diagonal lines (constant time-to-contact). Three controllers are an acceleration toward the goal (bang), a coast (zero control input) resulting in a constant velocity, and a deceleration (bang) into the target. Adapted from [23].

In the following section we will investigate the effect of human operator modeling in calibrating and tuning parameters in the interaction with a zooming and scrolling application on a mobile platform, which employs novel input methods such as tilt.

4 Human Operator Modeling in Mobile Human Computer Interaction

4.1 Model-Based Tilt-Controlled SDAZ

Speed-dependent automatic zooming (SDAZ) couples rate-based scrolling with automatic zooming to overcome the limitations of typical scrolling interfaces and to prevent extreme visual flow. SDAZ has many parameters that should be tuned and calibrated. Calibrating SDAZ interfaces, which are built based on static models [22,7,8] requires many experiments to customise SDAZ behaviour. However, these models fail on handheld devices for three reasons: (1) they require frustrating calibration and tuning procedure by trial and error, (2) they do not take into account novel input technologies and human gain and time delay in the interaction and, (3) they are designed for desktop computers.

In [13] we built an SDAZ interface based on dynamic systems approach and control theory. In this work, the applicability of state-space modeling is demonstrated by implementing the SDAZ interface for a text browsing interface on a Pocket PC instrumented with a tilt sensor. It is shown that state-space modeling makes tuning and calibration easier even with higher DOF inputs; because proper settings can only be found by observing the behaviour of the simulated interaction before the actual implementation.

In this paper we adapt the state-space model for tilt-controlled SDAZ described in [13], accommodate the user operator model and present a document viewer to examine the coupled tilt-controlled SDAZ and user control behaviour.

4.2 State-space Model for Tilt-Controlled SDAZ - Review

Eslambolchilar and Murray-Smith [13] show that SDAZ can be simulated as a flying object like a bird or an airplane. Thus, the zoom-level is a function of position, velocity and tilting angle. They suggest a standard second-order dynamics of a mass-spring-damper system to simulate SDAZ behaviour on a handheld device using a tilt input. The first time-derivative of the state equations can be written as below:

$$\dot{x}_1(t) = v = x_2(t) \tag{8}$$

$$\dot{x}_2(t) = a = \dot{v} = \frac{-R}{m}x_2(t) + \frac{1}{m}u(t) \tag{9}$$

$$\dot{x}_3(t) = \dot{z}(t) = \frac{-b}{m}x_2(t) - \frac{R'}{m}x_3(t) + \frac{c}{m}u(t) \tag{10}$$

Where R and R' are air resistance in both horizontal and vertical directions which provide damping effects, m is mass, v is velocity of the object, a is acceleration, z is zoom, $\dot{z}$ is rate of change of zoom, b is a coefficient and c is a scaler. The standard matrix format of these equations is:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{-R}{m} & 0 \\ 0 & \frac{-b}{m} & \frac{-R'}{m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \\ \frac{c}{m} \end{pmatrix} u \tag{11}$$

$x_1(t)$, $x_2(t)$ and $x_3(t)$ are state variables; $x_1(t)$ is position of zooming-window, $x_2(t)$ is speed of scroll and $x_3(t)$ is zoom. u represents input, here input from the tilt sensor.

In [13] it is shown that we can choose suitable values for R, R', m, b and c coefficients in the state-space model such that the dynamic application running on the device is stable and controllable and the total cost function is minimised. However, the application itself might be stable, but when coupled with the time delay and lead-lag-dynamics of typical human control behaviour, the combined closed loop of human + controller + computing device might be unstable. This can vary by context as the dynamics of human responses might vary according to context.

4.3 Human Operator Modeling in Model-Based Tilt-Controlled SDAZ

From Section 3 we know the frequency response of the human element changes with changes in the device dynamics. The tilt-controlled SDAZ is a velocity controlled application, thus in this case the human looks more like a gain and time delay (Figure 7). In the velocity control mode we can introduce $a_1 = \frac{R}{m}$ and $b_1 = \frac{1}{m}$ and rewrite equation (9) for controller as below:

$$\dot{x}_2(t) = -a_1 x_2(t) + b_1 u(t) \tag{12}$$

From equation (7) we can write the open-loop transfer function for the human, controller and the device, considering the device (mobile phone/smart phone/PDA) is light and no delay is caused by that, as below:

$$Y_h(j\omega)C(j\omega)Y_p(j\omega)=\left[Ke^{-j\omega\tau}\right]\cdot\left[\frac{b_1}{j\omega+a_1}\right]\cdot 1 \qquad (13)$$

The left bracket is a transfer function for human operator, which is a simple gain and time delay, the middle bracket is the transfer function of the SDAZ application in the velocity control mode and the last one indicates a simple gain for the device.

In [23] it has been shown that time delay, τ, is between 0.1 (s) and 0.25 (s) and crossover frequency, K is between 4 s^{-1} and 6 s^{-1} for velocity control systems.

To experiment the effect of human control behaviour (changes in coefficients a_1 and b_1) in a mobile human computer interaction scenario, we do an experiment with a tilt-controlled document browser implemented on a PDA. This experiment includes simulating the behaviour of tilt controller + device in MATLAB and then comparing this behaviour with the behaviour of real tilt-controlled SDAZ running on a PDA while people interact with it. Also, we evaluate the efficiency of tuning and calibrating coefficients of tilt SDAZ controller when the human behaviour model is included in the Bode space.

4.4 A Document Browser Example

We adapted the state-space model described in [13] to implement a dynamic tilt-controlled SDAZ running on an HP 5550 PDA (Figure 9). In Sections 4.3 we learned how to model human operator in dynamic systems. In this section we show how we can tune and calibrate parameters of the tilt-controlled SDAZ in the combined closed loop of human + tilt controller + mobile device applying these models. To do this first, we simulate the behaviour of the closed-loop dynamic system (i.e. tilt SDAZ controller + mobile device) excluding the human operator model in MATLAB with different parameter settings. Then, we ask a few participants to play with the tilt-controlled

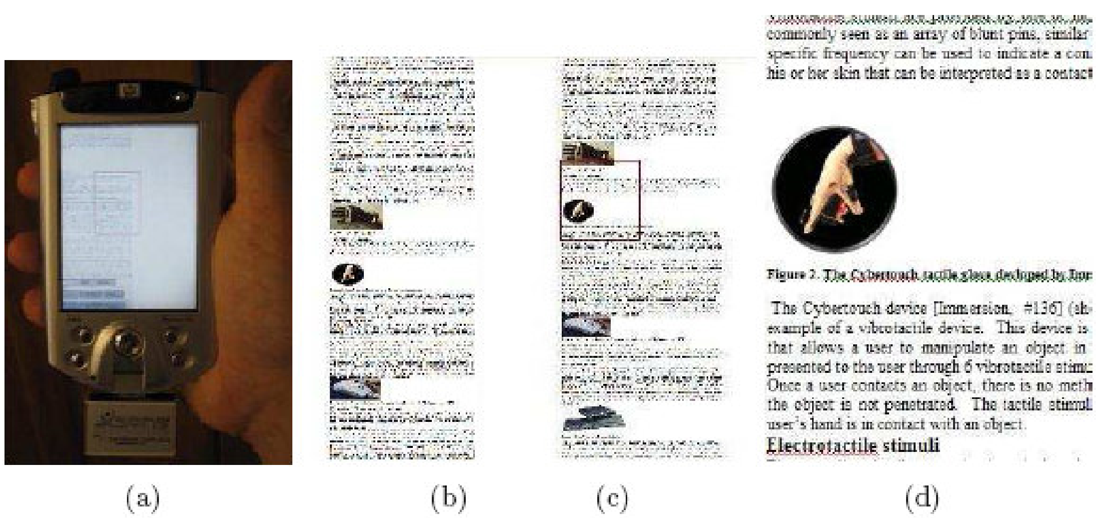


Figure 9: A Pocket PC and an accelerometer attached to the serial port. (a) Screen shots of the document browser (b,c and d). (b) shows a red box moving rapidly over the picture, (c) shows the user has found the picture and landing there, and (d) shows the zoomed-in picture. Adapted from [13].

document browser on a PDA with the same parameter settings. A well designed model of interaction should generate similar behaviour on the real device (while the user is interacting) and the simulated one. Using Bode analysis in MATLAB (presented in equation (13)) confirms that this analysis is capable of predicting the performance of human-machine systems before the actual implementation of interaction and the model which takes into account the combined closed-loop human + controller + device stability and controllability achieve the best performance in the interaction.

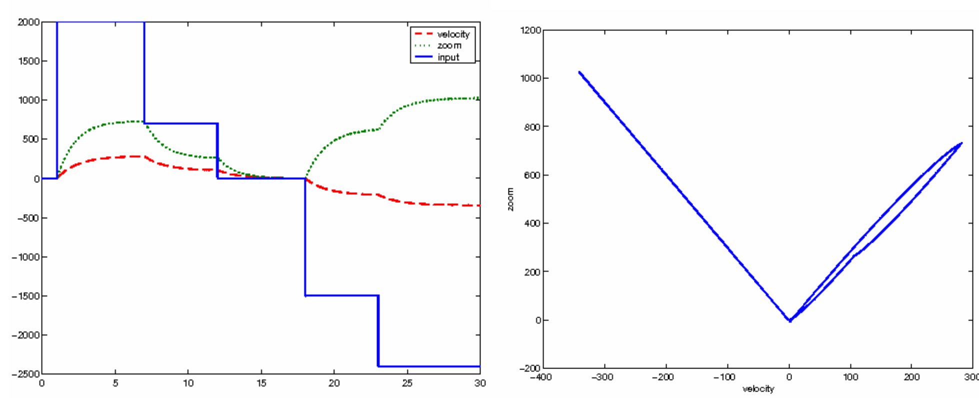
We recruited eight users for this experiment. Participants were told how the tilt-based SDAZ application works in a training session. We then asked them to try the tilt-controlled SDAZ and target two figures in the document. We examined different parameter settings and the effect of calibration on user performance.

4.4.1 Calibration and Performance Measures

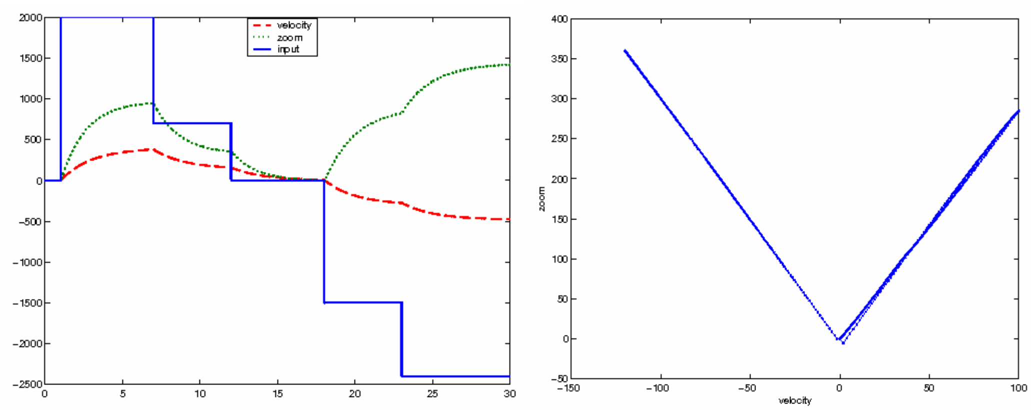
Tilt-controlled SDAZ is an example of velocity controlled application. Therefore, we chose gain, k=5 s^{-1}, and time delay, τ=0.2 (s) for human control model. These settings are in the valid range for human time delay and crossover frequency in velocity control systems (see Section 4.3).

For the first experiment we start with settings m = 10 kg, R and R' = 1 kgs^{-1}, b = 3 kgs^{-1} and c = 3, which or a_1=0.1 s^{-1} and b_1=0.1 kg^{-1}. Figure 10(a) presents the time domain response of the simulated system (tilt-controlled SDAZ + device) in MATLAB to varying step inputs with above settings. The simulated system is stable and controllable on its own, i.e. the controllable matrix is full rank [28]. The relationship between the velocity and zoom is almost linear, indicating that the system's behaviour is linear. Also, despite changes in the sign of the input data, the relationship between velocity and zoom remains linear.

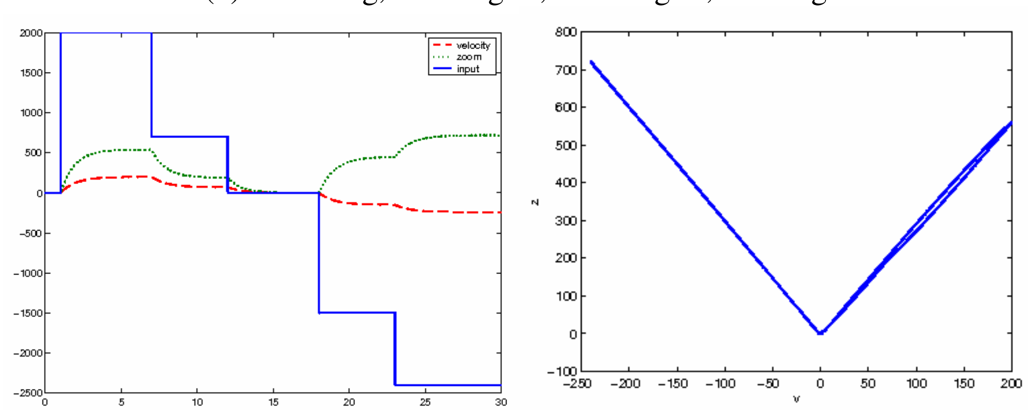
Then two users were asked to play with the document browser on a PDA with the above calibrated parameter settings and target two figures. Figure 11(a) presents the first user behaviour in this task. It is obvious that the user has not been comfortable with this task and he complained it was almost impossible to land on the figure because any slight tilt was causing sluggish behaviour in zoom (slope of zoom vs. velocity has been very variable Figure 11(b)). This also supports the comments made by Gaines (see Section 3.6) about the variability of the human operator's switching boundary in controlling an unstable combined closed-loop system. In this unstable system the user had to change his switching lines (one between acceleration to coast and the other between coast to deceleration) to control the targeting.



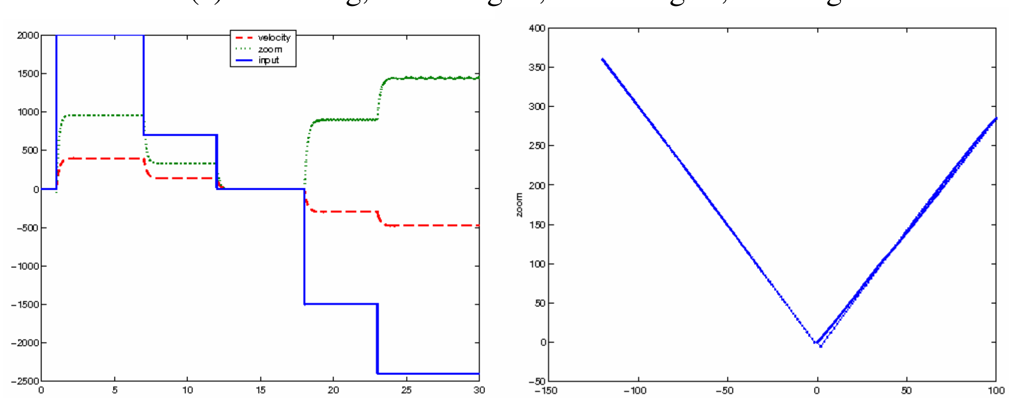
a) m = 10 kg, R = 1 kgs^{-1}, R' = 1 kgs^{-1}, b = 3 kgs^{-1} and c = 3.



(b) m = 10 kg, R = 5 kgs^{-1}, R' = 5 kgs^{-1}, b = 3 kgs^{-1} and c = 3.

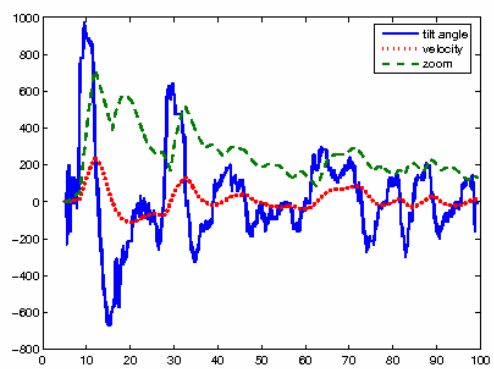


(c) m = 10 kg, R = 10 kgs^{-1}, R' = 10 kgs^{-1}, b = 3 kgs^{-1} and c = 3.

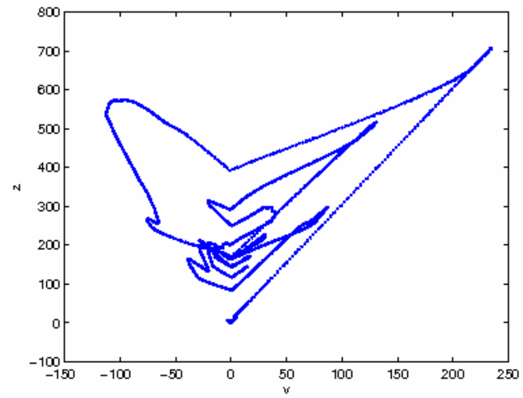


(d) m = 1 kg, R = 1 kgs^{-1}, R' = 1 kgs^{-1}, b = 3 kgs^{-1} and c = 0.5.

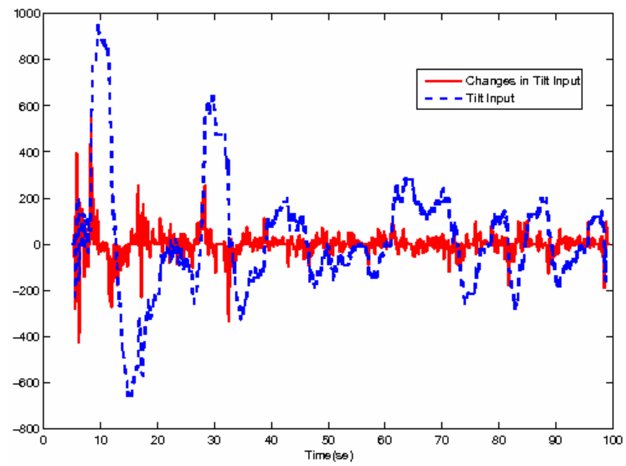
Figure 10: Left - behaviour of simulated SDAZ model in MATLAB to a step input. Right- phase plot, zoom vs. velocity



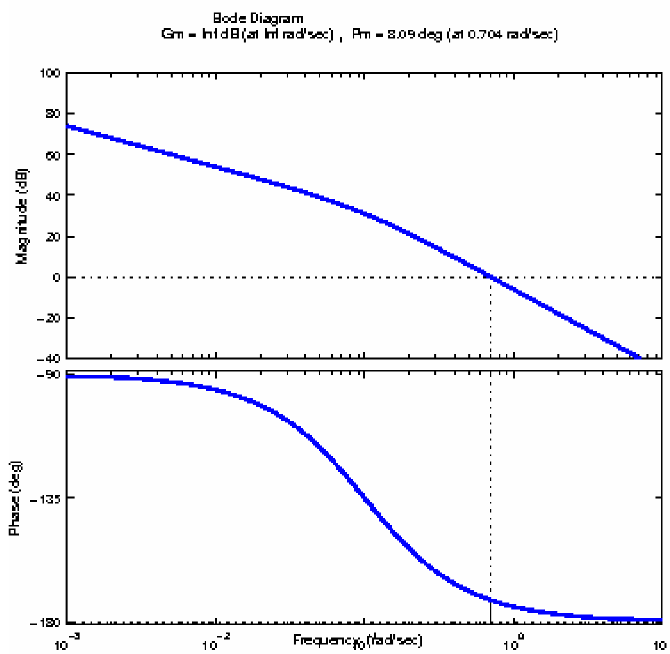
(a) User’s tilt behaviour and changes in velocity and zoom



(b) Zoom vs. velocity

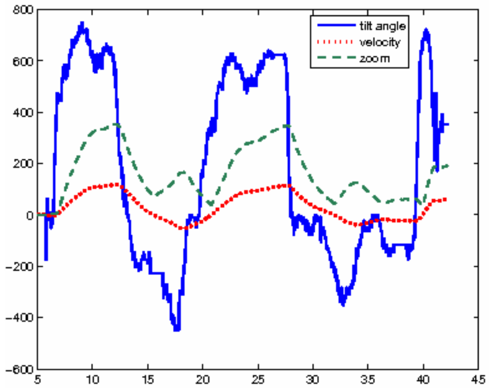


(c) Total sum of changes in input is 15304 units.

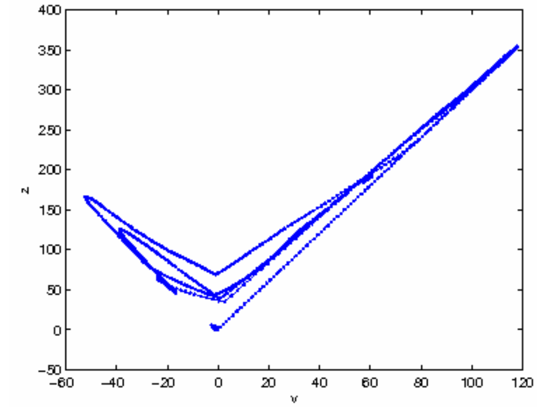


(d) Phase margin = 8.09 degrees and amplitude ratio = Infinite db

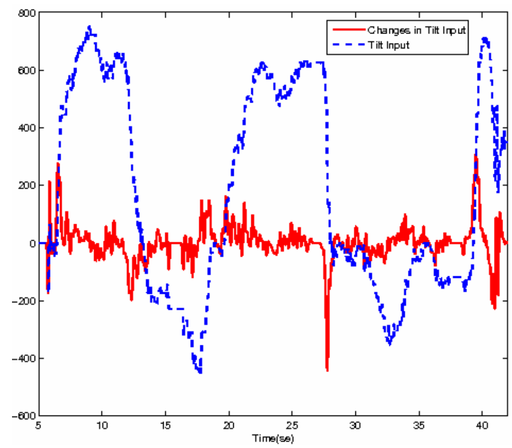
Figure 11: Controller with settings m = 10 kg, R and R' = 1 kgs^{-1}, b = 3 kgs^{-1} and c = 3 or a_1=0.1 s^{-1} and b_1=0.1 kg^{-1}.



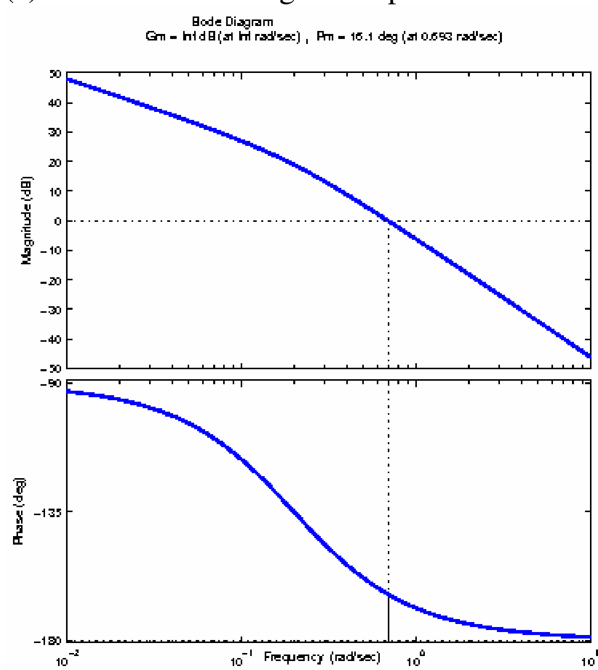
(a) User’s tilt behaviour and changes in velocity and zoom



(b) Zoom vs. velocity

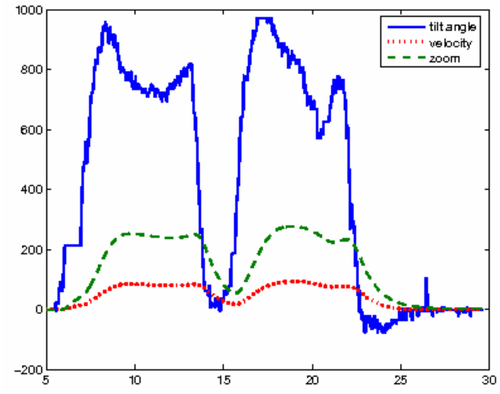


(c) Total sum of changes in input is 8118 units.



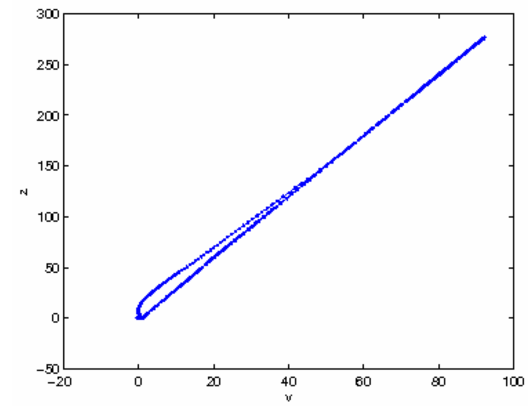
(d) Phase margin = 16.1 degrees and amplitude ratio = Infinite db

Figure 12: Controller with settings m = 10 kg, R and R' = 5 kgs^{-1}, b = 3 kgs^{-1} and c = 3 or a_1=0.2 s^{-1} and b_1=0.1 kg^{-1}.

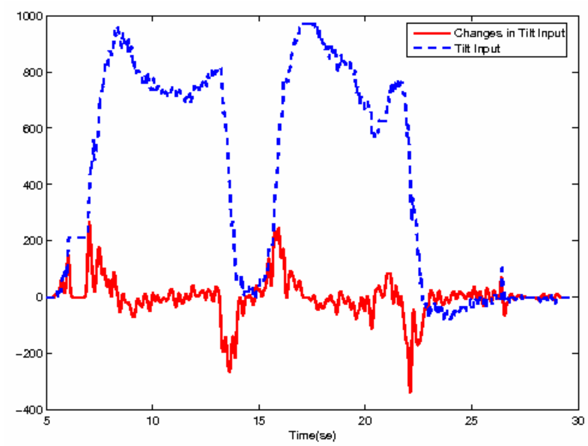


(a)

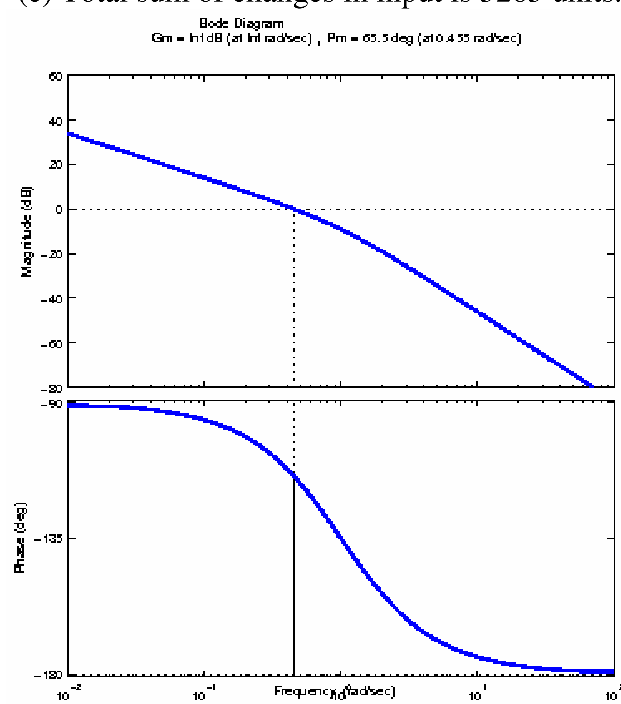
User’s tilt behaviour and changes in velocity and zoom



(b) Zoom vs. velocity

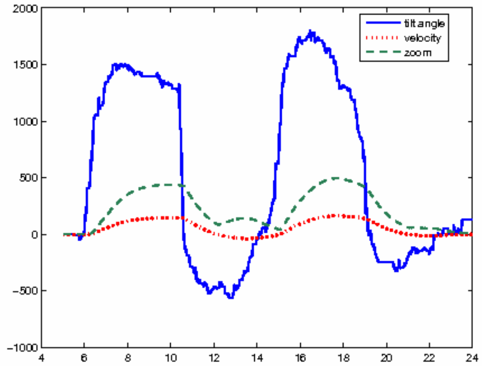


(c) Total sum of changes in input is 5265 units.

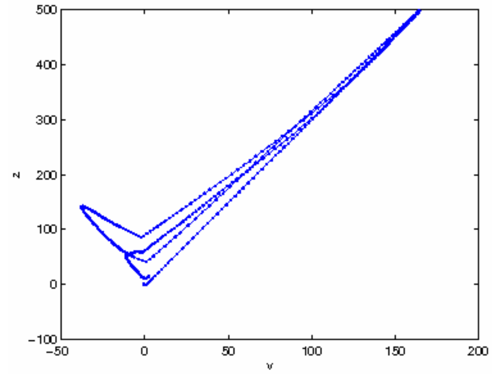


(d) Phase margin = 65.5 degrees and amplitude ratio = Infinite db

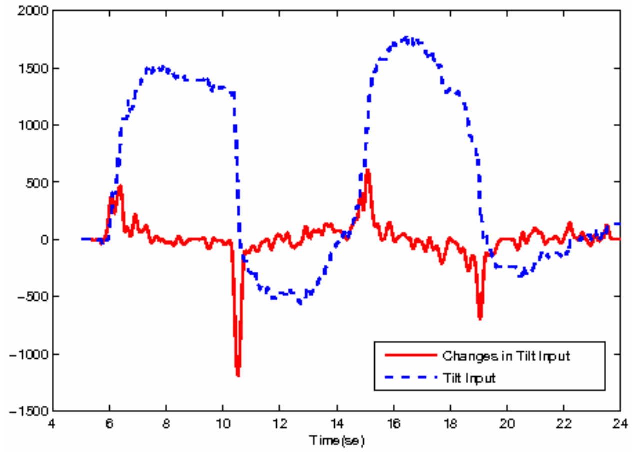
Figure 13: Controller with settings m = 10 kg, R and R' = 10 kgs^{-1}, b = 3 kgs^{-1} and c = 3 or a_1=1 s^{-1} and b_1=0.1 kg^{-1}.



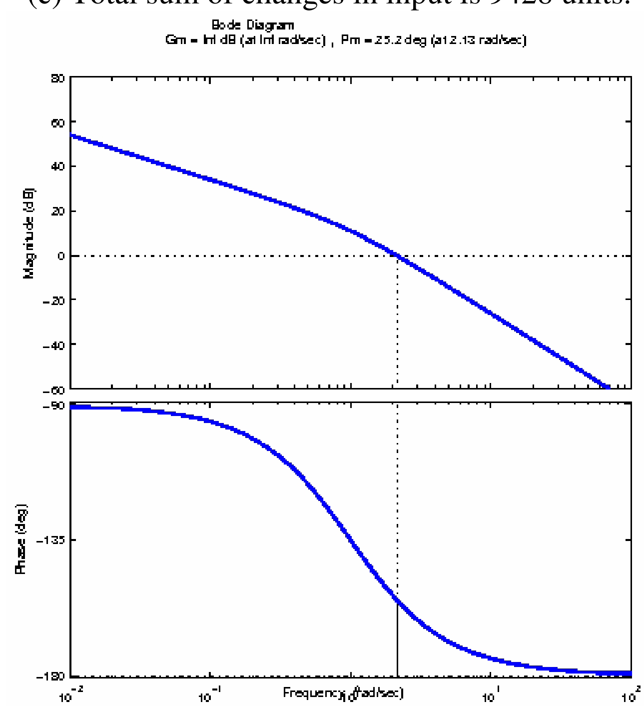
(a) User's tilt behaviour and changes in velocity and zoom



(b) Zoom vs. velocity



(c) Total sum of changes in input is 9428 units.



(d) Phase margin = 25.2 degrees and amplitude ratio = Infinite db

Figure 14: Controller with settings m = 1 kg, R and R' = 1 kgs^{-1}, b = 3 kgs^{-1} and c = 0.5 or a_1=1 s^{-1} and b_1=1 kg^{-1}.

The total sum of input changes (equation (3)) is one measure (i.e. cost function) to calculate the efficiency of the interaction. In smooth and efficient interaction this cost function is minimised (see Section 3.4). Figure 11(c) presents the first user tilt input and the total sum of changes in the input. The total sum of input changes for this user in this task was 15304 units. These settings with the second participant presented very similar behaviour. This user also complained about lack of control in landings and very sensitive tilt behaviour. His total sum of tilt input changes was 16201 units.

If we include the human operator model in the combined closed-loop of controller + device and simulate and analyse the behaviour of human + controller + device in the Bode space, then we have an earlier indication of the stability of the closed-loop interactive system before the actual implementation on the real device. For instance, Figure 11(d) presents Bode plots of the open-loop transfer function for the human and controller with the setting mentioned earlier. The phase margin was very low, 8.09 degrees, for this task.[4] This low phase margin confirms the unstable behaviour of human + controller + device while interacting with the tilt-controlled SDAZ.

We changed the coefficients to m = 10 kg, R and R'= 5 kgs^{-1}, b = 3 kgs^{-1} and c = 3 or a_1=0.2 s^{-1} and b_1=0.1 kg^{-1}. Figure 10(b) presents the time domain response of the simulated system (tilt-controlled SDAZ + device) in MATLAB to a varying step inputs with these new settings. The system is controllable and the relationship between the velocity and zoom is linear indicating the system's behaviour is linear. We asked two other users to repeat the targeting task (figure 12(a)). There is a slight improvement in controlling the task in comparison to the previous case. The total sum of input changes for the first user was 8118 unit, which is much lower than previous users' activities and J_t, the time taken to complete the task (see equation 2) was considerably shorter than previous example. The second user who tried these settings confirmed the first participant's findings. This participant had 9008 unit in the total sum of tilt input changes.

After including the human operator model in the combined closed-loop of controller (with the new coefficients) + device and simulating and analysing the behaviour of human + controller + device in the Bode space, the phase margin increased slightly, 16.1 degrees, but still low (Figure 12(d)). We changed the coefficients to m = 10 kg, R and R' = 10 kgs^{-1}, b = 3 kgs^{-1} and c = 3 or a_1=1 s^{-1} and b_1=0.1 kg^{-1} and observed the simulated system's behaviour in MATLAB. Figure 10(c) presents that the system has been controllable and stable. We experimented the system with two different users for the same targeting task and observed the behaviour presented in Figure 13(a). The first user's performance was much higher than other participants in two previous settings, with only 5265 unit in his total sum of input changes. It had also taken a shorter time than four previous subjects to complete the task (Figure 13(c)). The second user who tried these settings presented very similar results. This participant had 5123 unit in the total sum of tilt input changes.

We included the human operator model in the combined closed-loop of controller (with the new coefficients) + device and simulated and analysed the behaviour of human + controller + device in the Bode space, the phase margin increased considerably, 65.5 degrees, in compare to the previous ones (Figure 13(d)). This confirms the smooth and comfortable behaviour of human + controller + device while interacting with the tilt-controlled SDAZ.

---

[4]How to find phase margin in Bode plots?
(1) Find the frequency at which the amplitude/gain plot of the Bode plot crosses 1 (0dB). (2) Find that frequency on the phase plot and find the phase (e.g. you have -60 degrees). (3) Find how far it is from -180 degrees (e.g. in this case, -60 - (-180) = 120 degrees of phase margin).

Other settings for this system could not bring the cost down (while the users were interacting with the device) or maximise the phase margin (in the Bode space) more than settings, a_1=1 and b_1=0.1 (Figure 13). For instance, in another experiment we chose m = 1 kg, R and R' = 1 kgs^{-1}, b = 3 kgs^{-1} and c = 0.5 or a_1=1 s^{-1} and b_1=1 kg^{-1}. The simulated system (tilt-controlled SDAZ + device) in MATLAB is controllable and stable (Figure 10(d)), however, the coupled human-tilt controlled SDAZ-device presents a different behaviour (Figure 14). Two participants tried the system and their performance was worse than the previous setting: the total sum of input changes was 9428 unit for participant "x" and 10131 for participant "y"; the time taken to complete the task was longer; the phase margin of the simulated combined closed-loop of human + controller + device was also lower, 25.2 degrees.

This experiment shows that first, the controller with settings m = 10 kg, R and R' = 10 kgs^{-1}, b = 3 kgs^{-1} and c = 3 is stable; second, a quasi-linear model of human operator (first-order system, equation (13)) is a suitable model for our task; third, phase margin and amplitude ratio of the open-loop transfer function for the human + controller + device is maximised for this setting; fourth, the cost function or total sum of changes in the input, equation (3)) is minimised for this setting.

Summary

This simple example proves that: (1) we can choose suitable values for R, R', m, b and c coefficients in the state-space model such that the closed loop of tilt-controller + device is stable and controllable and the total cost function is minimised. However, the combined human + tilt controller + device might be unstable; (2) we can consider the time delay and lead-lag-dynamics of typical human control behaviour and simulate and analyse the combined closed loop of human + controller + computing device in the Bode space before the actual implementation; and (3) tuning and calibrating coefficients and parameters can be done differently; by using analytical tools such as Bode analysis correct values for coefficients are not produced by trial and error anymore. This reduces the frustration of interaction designers dramatically.

5 Conclusions and Future Work

The framework presented in [13] revealed a number of unexplored areas in the context of human-mobile device interaction and our work continued that theoretical basis for evaluating specific interaction loops in interactive systems on small, portable computing devices while coupled with human users. We presented an example of analysing an interface by taking into account the human control behaviour. The applicability of human operator modeling was demonstrated by implementing the SDAZ interface for a text browsing system on a PDA instrumented with a tilt sensor.

Using the simulated state-space model of SDAZ in MATLAB we showed that the controlled system itself might be stable and controllable, but when coupled with the time delay and lead-lag-dynamics of typical human control behaviour, the combined closed-loop system might be unstable. It proves that control theory can support design of closed-loop interaction between human and system. In the general approach (i.e., interaction with higher degrees of freedom input and many controlled variables) this model makes tuning and calibration a lot easier for the system when it is coupled with the time delay and lag-lead-dynamics of human control behaviour; because proper settings, which make the closed-loop system-human stable, can only be found by observing the behaviour of the simulated system before the actual implementation (Bode analysis).

We modeled the human operator as a gain and time delay, plotted the Bode plots of the open-loop transfer function for the human and controller and calculated the activity of the user using a performance function with different settings for the controller and human. It helped us to choose settings that suit both the system and the user and made the controlled system stable when coupled with user model. In principle we are retracing the developments in aircraft development, where measures of the handling quality of aircraft moved were associated with a standardised set of subjective measures, and where mathematical models of human control behaviour ('paper pilots') were used together with models of aircraft dynamics to predict handling qualities before an aircraft was built.

This work opens new directions in both design and usability areas for future work. The specific results we gained through the use of an accelerometer (the tilt-dynamics) in mobile human interaction particularly in zooming and scrolling environments allows us to explore new areas to inform mobile design and evaluation. The analysis in the paper is also relevant to other sensors, where the declutching issue might be less relevant (e.g., pressure sensors in touch screen based interaction). This has the potential to improve the design process, for designing interaction on modern mobile devices such as the Nokia n-series and iPhone where a variety of sensors are available and the time and cost associated with trial and error experiments for interaction loop design can become prohibitive. We also expect the approach to be applicable to challenging interaction tasks such as design for special needs, such as Brain Computer Interaction [39], where the models of user response will be more complex and variable than used in this paper.

Acknowledgments

The authors gratefully acknowledge the support of IRCSET BRG SC/2003/271 Continuous Gestural Interaction with Mobile devices, EPSRC project EP/E042740/1, HEA project Body Space, and SFI grants 00/PI.1/C067, and 00/RFP06/CMS052 the IST Programme of the European Commission, under PASCAL Network of Excellence, IST 2002-506778, and PASCAL 2 network, and EC project TOBI: Tools for Brain-Computer Interaction. Both authors had posts at the Hamilton Institute, NUI Maynooth, during the development of this work.

References

[1] Bartlett, J. F., 2000. Rock 'n' Scroll Is Here to Stay. In: *IEEE Computer Graphics and Applications*. Vol. 20, No. 3, pp. 40-45 .

[2] Beaudouin-Lafon, M., 2004. Designing Interaction, not Interfaces. In: *AVI'04: Proceedings of the working conference on Advanced visual interfaces*. ACM Press, Gallipoli, Italy, pp. 15-22.

[3] Bellotti, V., Back, M., Edwards, W. K., Grinter, R. E., Henderson, A., and Lopes, C. 2002. Making sense of sensing systems: five questions for designers and researchers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, pp. 415-422

[4] Benford, S., SchnÄadelbach, H., Koleva, B., Anastasi, R., Greenhalgh, C., Rodden, T., Green, J., Ghali, A., Pridmore, T., Gaver, B., Boucher, A., Walker, B., Pennington, S., Schmidt, A., Gellersen, H., Steed, A., 2005. Expected, sensed, and desired: A framework for designing sensing-based interaction. In *ACM Transaction of Computer-Human Interaction*, 12 (1), 3-30.

[5] Buxton, W., 1995. Integrating the periphery and context: A new taxonomy of telematics. In: *Proceedings of Graphics Interface'95*, pp. 239-246.

[6] Card, S., Mackinlay, J., Robertson, G., 1991. A Morphological Analysis of the Design Space of Input Devices. In: *ACM Transactions on Information Systems*. Vol. 9, ACM Press, pp. 99-122.

[7] Cockburn, A., Savage, J., 2003. Comparing Speed-Dependent Automatic Zooming with Traditional Scroll, Pan, and Zoom Methods. In: *People and Computers XVII: British Computer Society Conference on Human Computer Interaction*, Bath, England. pp. 87-102.

[8] Cockburn, A., Savage, J., Wallace, A., 2005. Tuning and testing scrolling interfaces that automatically zoom. In: *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI'05*. ACM Press, Portland, Oregon, USA, pp. 71-80.

[9] Doherty, G., Anderson, T., Wilson, M., Faconti, G., 2001. A Control- Centred Approach to Designing Interaction with Novel Devices. In: *Conference on Universal Access in Human Computer Interaction*, Lawrence Erlbaum Associates, Inc., New Orleans, pp. 286-290.

[10] Doherty, G., Massink, M., 1999. Continuous Interaction and Human Control. In: Alty, J. (Ed.), *Proceedings of the XVIII European Annual Conference on Human Decision Making and Manual Control*, Group-D Publications, Loughborough, pp. 80-96.

[11] Doherty, G., Watts, L., 2000, Perceptual-Motor Performance Issues in the MagicBoard Scenario. In: *Continuous Interaction Workshop*, ACM CHI 2000 Workshops.

[12] Dong, L., Watters, C., and Duffy, J. 2005. Comparing two one-handed access methods on a PDA. In *Proceedings of the 7th international Conference on Human Computer interaction with Mobile Devices and Services*, Salzburg, Austria, September. MobileHCI '05, vol. 111. ACM, New York, pp. 235-238

[13] Eslambolchilar, P., Murray-Smith, R., 2008. Control centric approach in designing scrolling and zooming user interfaces. In: Brewster, S., Oulasvirta, A. (Eds.), *International Journal of Human-Computer Studies (IJHCS)*, Special issue on Mobility. pp. 838-856, Vol.66, No.12, December 2008

[14] Faconti, G., Massink, M., 2001. Continuous interaction with computers: Issues and Requirements. In: C.Stephanidis (Ed.), *Proceedings of Universal Access in HCI, Universal Access in HCI - HCI International 2001*, Vol. 3, Lawrence Erlbaum Associates, Inc., New Orleans, pp. 301-304.

[15] Fallman, D., 2002. Wear, point and tilt. In: *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques*. ACM Press, pp. 293-302.

[16] Fitts, P. M., 1954. The Information Capacity of the Human Motor system in Controlling the Amplitude of Movement. In: *Journal of Experimental Psychology*, Vol. 47, pp. 381-391.

[17] Gaines, B. R., 1969. Linear and Nonlinear Models of the Human Controller. In: *International Journal of Man-Machine Studies*, Vol. 1, pp. 333-360.

[18] Gutwin, C., 2002. Improving Focus Targeting in Interactive Fisheye Views. In: *Proceeding of ACM Conference on Human Factors in Computing Systems*, CHI'02. ACM Press, Minneapolis, Minnesota, USA, pp. 267-274.

[19] Hancock, M. S., Booth, K. S., 2004. Improving menu placement strategies for pen input. In: *Proceedings of the 2004 conference on Graphics interface*, Vol. 62. ACM Press, Ontario, Canada, pp. 221-230.

[20] Hinckley, K., Pierce, J., Horvitz, E., Sinclair, M., 2005. Foreground and background interaction with sensor-enhanced mobile devices. In: *ACM Transactions on Computer-Human Interaction (TOCHI)*. Vol. 12, ACM Press, pp. 31-52.

[21] Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E., 2000. Sensing Techniques for Mobile Interaction. In: *UIST'00: Proceedings of the 13th annual ACM symposium on User interface software and technology*. ACM Press, San Diego, CA, USA, pp. 91-100.

[22] Igarashi, T., Hinckely, K., 2000. Automatic Speed-Dependent Zooming for Browsing Large Documents. In: *UIST'00: 13th Annual Symposium on User Interface Software and Technology*. ACM Press, San Diego, CA, USA, pp. 139-148.

[23] Jagacinski, R. J., Flach, J. M., 2003. Control Theory for Humans: Quantitative approaches to modeling performance. Lawrence Erlbaum Associates, Inc., Mahwah, New Jersey.

[24] Jagacinski, R. J., Repperger, D., Moran, M., Ward, S., Glass, B., 1980. Fitts' law and the microstructure of rapid discrete movements. In: *American Psychological Association*, Mahwah, New Jersey. Vol. 6.

[25] Karlson, A. K., Bederson, B., SanGiovanni, J., 2005. AppLens and LaunchTile: Two designs for one-handed thumb use on small devices. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '05)*. NY, ACM Press, pp. 201-210.

[26] MacKenzie, I., Ware, C., 1993. Lag as a determinant of human performance in interactive systems. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems, INTERCHI '93*, IOS Press, Amsterdam, The Netherlands, pp. 488-493.

[27] Oakley, I., Ängeslevä, J., Hughes, S., O'Modhrain, S., 2004. Tilt and feel: Scrolling with vibrotactile display. In: *Proceedings of Eurohaptics 2004*. Munich Germany.

[28] Ogata, K., 1990. Modern Control Engineering. Englewood Cliffs, NJ: Prentice Hall.

[29] Poulton, E. C., 1974. Tracking skill and manual control. Academic press.

[30] Powers, W. T., 1989. Living Control Systems: Selected papers of William T. Powers. The Control Systems Group Book.

[31] Powers, W. T., 1992. Living Control Systems II: Selected papers of William T. Powers. The Control Systems Group Book.

[32] Rekimoto, J., 1996. Tilting Operations for Small Screen Interfaces. In: *UIST'96: Proceedings of the 9th annual ACM symposium on User interface software and technology*. ACM Press, Seattle, Washington, USA, pp. 167-168.

[33] Schmidt, A., June 2000. Implicit Human Computer Interaction Through Context. In: *Personal Technologies*. Vol. 4. Springer-Verlag, pp. 191-199.

[34] Schmidt, R. A., Lee, T., 2005. Motor Control and Learning-4th Edition - A Behaviourial Emphasis. Human Kinetics.

[35] Sheridan, T. B., Ferrell, W. R., 1974. Man-Machine Systems: Information, Control, and Decision Models of Human Performance. MIT press.

[36] Streitz, N. A., 2001. Mental vs. Physical Disappearance: The Challenge of Interacting with Disappearing Computers. In: *Workshop Proceedings Distributed and Disappearing User Interfaces in Ubiquitous Computing*. ACM CHI'01, Seattle, WA.

[37] Thimbleby, H., 1990. User Interface Design. ACM Press.

[38] Wigdor, D., Balakrishnan, R., 2003. TiltText: Using Tilt for Text Input to Mobile Phones. In: *UIST'03: Proceedings of the 16th annual ACM symposium on User interface software and technology*. ACM Press, Vancouver, Canada, pp. 81-90.

[39] Williamson, J., Murray-Smith, R., Blankertz, B., Krauledat, M and Müller, K.-R., *Designing for uncertain, asymmetric control: Interaction design for brain-computer interfaces*, Int. J. Human–Computer Studies, doi:10.1016/j.ijhcs.2009.05.009

[40] Williamson, J., 2006. *Continuous uncertain interaction*. Ph.D. thesis, University of Glasgow.