Available online at www.sciencedirect.com
ScienceDirect
Int. J. Human-Computer Studies 66 (2008) 838–856
International Journal of Human-Computer Studies
www.elsevier.com/locate/ijhcs

Control centric approach in designing scrolling and zooming user interfaces

Parisa Eslambolchilar[a,*], Roderick Murray-Smith[b,c]

[a]Computer Science Department, Swansea University, Swansea SA2 8PP, UK
[b]Department of Computing Science, Glasgow University, Glasgow G12 8QQ, Scotland
[c]Hamilton Institute, NUI Maynooth, Co. Kildare, Ireland

Received 15 June 2007; received in revised form 28 April 2008; accepted 1 July 2008
Available online 5 July 2008

Abstract

The dynamic systems approach to the design of continuous interaction interfaces allows the designer to use simulations, and analytical tools to analyse the behaviour and stability of the controlled system 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 system before the actual implementation, and in response to user feedback. In this work we provide a dynamic systems interpretation of the coupling of internal states involved in speed-dependent automatic zooming, and test our implementation on a text browser on a Pocket PC instrumented with a tilt sensor. We illustrate simulated and experimental results of the use of the proposed coupled navigation and zooming interface using tilt and touch screen input.

Keywords: Human–computer interaction; Dynamics; Continuous interaction; Tilt input; Mobile devices; Zooming/scrolling user interfaces

1. Introduction

Recent developments in mobile phone software and hardware allow users to interact with the phone in more physically realistic ways than they did before. Users have touch-screen input, accelerometer sensing and interface dynamics modelled on real-world physics. We can give objects inertia, make their movements subject to friction or provide spring-like internal dynamics. As a topical example, on the iPhone, with just a flick of a finger, the screen begins to pan, and on release, momentum carries it forward and this is gradually overcome by friction, slowing it to a stop. Other physically motivated features include elastic collision at document limits, leading to a bounce back in the display, keeping a consistent physical metaphor, and using it to inform the user of a constraint in a pleasing manner. Therefore, its scrolling and zooming user interface (ZUI) feels more alive, fluid and less abrupt. However, work in dynamic zooming and panning is still in its early stages, with no comprehensive overview of concepts, specification mechanisms, calibration tools or design guidelines being available.

The main contribution of this work is to develop a theoretical framework for specification, analysis and calibration of panning and zooming dynamics. It is especially topical and interesting for guiding design of non-standard input technologies on mobile devices. The issue was motivated by analysing an interaction technique called speed-dependent automatic zooming (SDAZ) (Igarashi and Hinckely, 2000), which in previous research has been found to outperform manual zooming approaches on desktop computers (Cockburn and Savage, 2003; Cockburn et al., 2005). 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 system 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

*Corresponding author. Tel./fax: +44 1792602658.
E-mail addresses: p.eslambolchilar@swansea.ac.uk (P. Eslambolchilar), rod@dcs.gla.ac.uk (R. Murray-Smith).

1071-5819/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijhcs.2008.07.005

scrolling, and without becoming disoriented by extreme visual flow. This may improve constantly zooming in and out and pinching and dragging instead of reading/browsing web pages on the iPhone.

At present, there are few guidelines for the calibration of SDAZ, i.e., predefined constants governing their automatic zooming behaviour, on desktops (Igarashi and Hinckely, 2000; Cockburn et al., 2005). However, these guidelines fail on handheld devices and they do not take into account novel input technologies. The absence of formal guidelines means that designers are forced to adjust the properties of the automatic zooming by trial and error. Also, having tilt and gestures as input will add more complexities to the design. 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 behaviour (Cockburn and Savage, 2003; Cockburn et al., 2005). Moreover, mapping limited input events (single-degree of freedom (DOF) tilt input) to multiple actions increases user frustrations (Büring, 2007). Sensor noise in tilt and gesture based interaction is also a major concern (Hinckley et al., 2005).

We extend the research we did in Eslambolchilar and Murray-Smith (2004) to a more general framework and we demonstrate that, as suggested by Igarashi and Hinckely (2000), SDAZ is well suited to implementation on mobile devices instrumented with tilt sensors, which can then be comfortably controlled in a single-handed fashion. We present a real-world interpretation of the coupling of speed of scroll and zoom level in SDAZ, i.e., a flying object, which feels more appealing and intuitive to interact with. We demonstrate that simulation and analysis of SDAZ behaviour before the actual implementation assists designers in calibration and tuning coefficients and customising SDAZ behaviour. Moreover, continuous control systems to overcome limited input states, which is a common problem on handheld devices, offer mode switching and transition and the controller supports the user in completing the task with less effort by changing the interpretation of the inputs to being reference values, rather than control commands (Eslambolchilar, 2006). We also describe an alternative stylus-controlled implementation for the Pocket PC using the dynamic systems approach and compare it against tilt-controlled SDAZ. A further contribution is the use of a state-space formulation of SDAZ, which we believe is a promising reformulation of the technique, which opens the path to the use of analytic tools from optimal and manual control theory.

2. Speed-dependent automatic zooming

2.1. Review

Several techniques have been proposed to improve the manipulation of scroll bars. *AlphaSlider* (Ahlberg and Shneiderman, 1994) and *FineSlider* (Masui et al., 1995) are two alternative scrolling techniques for precise selection in large lists. They allow the user to control scrolling speed, enabling fine positioning in large documents. *LensBar* (Masui, 1998) combines these techniques with interactive filtering and semantic zooming, and also provides explicit control of zooming via horizontal motion of the mouse cursor. A rate-based scrolling interface is described in Zhai et al. (1997a) that maps displacement of the input device to the velocity of scrolling.

ZUIs, such as Pad and Pad++ (Perlin and Fox, 1993; Bederson et al., 1994), use continuous zooming as a central navigation tool. The objects are spatially organised in an infinite 2D information space, and the user accesses a target object using panning and zooming operations. A notable problem with the original zoomable interfaces is that they require explicit control of both panning and zooming, and it is sometimes difficult for the user to coordinate them. The user can get lost in the infinite information space. Wijk and Nuij (2003) derive an optimal trajectory for panning and zooming, for known start and end points. They use the shortest path in pan-zoom space as the most efficient path when the camera moved from point X to point Y. They consider the velocity of the moving image as a basis for measurements, i.e., they aim at a metric for the perceived average optic flow (Gibson, 1979) in the image window. Without considering the smoothness and the observer's maximum perceived velocity this path would be a straight line between these two points.[1] However, this proposed model only takes into account the perceptual level, not the cognition of the image. Wijk and Nuij suggest that their optimal path could be used to implement a system similar to SDAZ, which uses the scroll handle on scroll bars to determine the target point. A common problem with scrolling and zooming interfaces is that when users are zoomed out for orientation, there is not enough detail to do any 'real work'. When they are zoomed in sufficiently to see detail, the context is lost (Wobbrock et al., 2002). To reduce this problem, multiple windows can be provided, each with pan and zoom capability. Although this is reasonable for small information spaces, the many windows required by large spaces often lead to usability problems due to excessive screen clutter and window overlap. An alternative strategy is to have one window containing a small overview, while a second window shows a large more detailed view (Beard and Walker, 1990; Furnas and Bederson, 1995). The small overview contains a rectangle that can be moved and resized, and its contents are shown at a larger scale in the large view. This strategy, however, requires extra space for the overview and forces the viewer to mentally integrate the detail and context views. An operational overhead is also required; because the user must regularly move the mouse between the detail and context windows.

[1] By zooming out, the perceived velocity is less than the actual document velocity. Hence, the camera can be moved faster when it is moved further away from the document, with its perceived velocity remaining the same.

SDAZ, first proposed by Igarashi and Hinckely (2000), couples the user's rate of motion through an information space with the zoom level. The faster a user scrolls in the document, the 'higher' they fly above the work surface. This allows users to efficiently scroll a document without having to manually switch between zooming and scrolling or becoming disoriented by fast visual flow, and results in a smooth curve in the space-scale diagram (Furnas and Bederson, 1995). In traditional manual zooming interfaces, the user has to interleave zooming and scrolling (or panning); thus the resulting pan-zoom trajectory forms a zigzag line. Previous works on SDAZ (Igarashi and Hinckely, 2000; Cockburn and Savage, 2003) fails to give values for predefined constants governing their automatic zooming behaviour. Therefore, the properties of the automatic zooming is adjusted by trial and error. The proposed method in Cockburn et al. (2005) aids calibration by identifying the low-level components of SDAZ in a desktop implementation. Base calibration settings for these components are then established using a formal evaluation recording participants' comfortable scrolling rates at different magnification levels. The particular input device used can also influence the effectiveness of rate control. An experiment on 6 DOF input control (Zhai et al., 1997a) showed that rate control is more effective with isometric or elastic devices, because of their self-centering nature. It is also reported that an isometric rate-control joystick (Barrett et al., 1995) can surpass a traditional scroll bar and a mouse with a finger wheel (Zhai et al., 1997a). Another possibility is to change the rate of scrolling or panning in response to tilt, as demonstrated by Rekimoto (1996) as well as Harrison et al. (1998), suitable for small screen devices like mobiles phones and PDAs.

2.2. Applications of SDAZ on small screen devices

Jones et al. (2005) implemented two SDAZ interfaces for small screen displays: a document browser and a map browser, both based on pen pressure input. These interfaces were evaluated against interfaces using standard navigation techniques (scroll bars, pan and manual zoom). They found that SDAZ was, on average, 29% slower for browsing text documents on small screen displays. Participants using SDAZ were more accurate and performed significantly fewer actions when completing the tasks. Patel et al. (2004, 2006) also conducted research into the efficiency of SDAZ on small screen displays for photo browsing. They evaluated SDAZ against discrete zoom and gesture zoom interfaces. In the discrete zoom interface, thumbnails of photographs are presented ordered by creation time. Users can click/tap the desired photo to view an enlarged version, and click/tap again to return to the thumbnail view. In the gesture interface, scroll speed is proportional to vertical mouse displacement (as in rate-based scrolling interfaces) and zooming is proportional to horizontal mouse displacement. A larger horizontal displacement results in a higher zoom level. They found that the SDAZ and gesture zoom interfaces support faster navigation, higher accuracy and have lower subjective task load levels than the standard discrete zoom interface.

Büring (2007) in a comprehensive study on map-based ZUIs investigated different approaches for improving the interaction design. He developed different interfaces (SDAZ, unimanual approach based on pen pressure, and one bimanual approach in which users pan the view with the pen while manipulating the scale by tilting the device) and compared them in a usability study with 32 participants. The results show that SDAZ performed well for both simple speed tasks and more complex navigation scenarios, but that the bimanual interaction (pen and tilt) led to much user frustration; because mapping limited input events to multiple actions requires the user to switch manually between modes, and it engages their visual attention more than coupled speed-zoom SDAZ.

3. Dynamics and interaction

In the past 10 years many researchers have focused on sensory-augmented systems in mobile human–computer interaction. The results of these studies have shown that tilting and 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 (Rekimoto, 1996; Bartlett, 2000; Hinckley et al., 2000, 2005; Fallman, 2002; Karlson, 2005). These sensory augmented devices provide a tight coupling between the user and the system based on a continuous input/output exchange of dynamic information, which happens over a period of time and we cannot model this coupled human–system interaction as a series of discrete events and static models (Doherty and Massink, 1999; Faconti and Massink, 2001). We include dynamics because feedback from the display (either visual, haptic or audio) influences our actions and changes our perceptions. Thus we control what we perceive.

The idea of continuous dynamic interaction first explored by Powers in the 1960s (1989, 1992). This earlier work 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 to identify controlled variables in an interaction we can apply disturbances to variables which are under the user's control. If, despite the disturbances, the changes to the variables in question appear to be minimised by the user, 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. (1991) considers only user's cognitive skills in the interaction. Also, there are many physiological models of human body motion (Schmidt and Lee, 2005). These models are incomplete in that they only focus on the human in the interaction while the coupling between the user and the system has not been taken into account. Mathematical and dynamical models of the types used in *control theory* have been underused in HCI research.

3.1. 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* (Poulton, 1974; Jagacinski and Flach, 2003). The theory is applicable to a wide range of tasks, from target acquisition, maintaining a reference value, and tracking, and creates a framework for modelling human–computer coupled 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* modelling. A state-space representation is a mathematical representation used in control theory. This representation provides a convenient and compact way to model and analyse systems 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 system. This approach makes tuning and calibration a lot easier, especially when we use an input with more than 1 DOF; because it is now possible to tune settings appropriately for the interface by observing the behaviour of the simulated system before any 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 (Thimbleby, 1990; Beaudouin-Lafon, 2004). 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' (1954) law by viewing the pointing movements toward 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 postulating the use of a simple control law (Jagacinski and Flach, 2003) to describe user behaviour.

Using the theory, we can make consistent *conceptual models* (Liddle, 1996) using real-world effects such as haptic feedback of springs, viscous effects linked to motion in the liquid, or friction linked to speed of motion, which are easy to reproduce in a dynamic system, and we can choose to explicitly use these features to design the system to encourage interaction to fall into a comfortable, natural rhythm. This thus provides a few '*affordances*' because the presence of feedback affects the usability and understandability of the system and lets the user experience them (Gibson, 1979). For example, in a tilt-controlled visualisation application running on a PDA, the tilt sensor allows tilting but this tilting must be a meaningful, useful action, with a predictable outcome. The application presents changes in the speed of scroll and zoom according to tilt angles as an audio–visual output to the user. Presenting the current status of the controlled variables via audio, vision or haptic to the user makes their action more clear and the design model (i.e., how the designer understands how the system works) more *visible* (Norman, 1999; Preece et al., 2002). Furthermore while there is a feedback the user can determine the relationship (*mapping*) between actions and perceptions (Preece et al., 2002). Lastly, the natural constraints of human hand motion add some *constraints* to the interaction. For example, a roll tilt angle of -200° is not a convenient angle for holding the PDA and looking at the screen.

In the following sections we focus on manual control theory as a formal modelling approach to provide appropriate concepts to deal with issues of continuous dynamic interaction, exploring novel ideas in interaction with ZUIs and portable computational appliances, and human performance data analysis.

4. Tilt-controlled SDAZ on small screen devices

Implementing the SDAZ technique on a mobile device with inertial sensing allows us to investigate a number of issues: the use of single-handed tilt-controlled navigation, which does not involve obscuring the small display; human performance in tilt interaction, and the relative strength of stylus-controlled SDAZ, compared to tilt-controlled one. If successful, the user should be able to target a position quickly without becoming annoyed or disoriented by extreme visual flow, and we want the technique to provide smooth transitions between the magnified local view and the global overview, without the user having to manually change the document magnification factor. Unlike SDAZ methods discussed in Section 2.1, which are based on static models, in this paper we implement a dynamic model for tilt-controlled SDAZ running on an HP 5550 Pocket PC (Fig. 1). For more information on the implementation please see Eslambolchilar (2006).

Using an accelerometer provides a direct mapping from acceleration in the real world to the acceleration in the interface. The accelerometer (Xsens P3C, 3 DOF linear accelerometer) is attached to the serial port of the Pocket PC provides the roll and pitch angles. Similar to a mouse-controlled SDAZ (Cockburn and Savage, 2003; Savage, 2004) there are three mappings in the tilt-controlled SDAZ:

- *Tilt motion to zooming-window displacement*—The term 'zooming window' indicates a red scrolling rectangular window shown in Fig. 1. This box is a zoom-in region in the size of the PDA's screen (240×320 pixels) and the target, which the user wants to land on should be located inside this window. The zooming window acts as

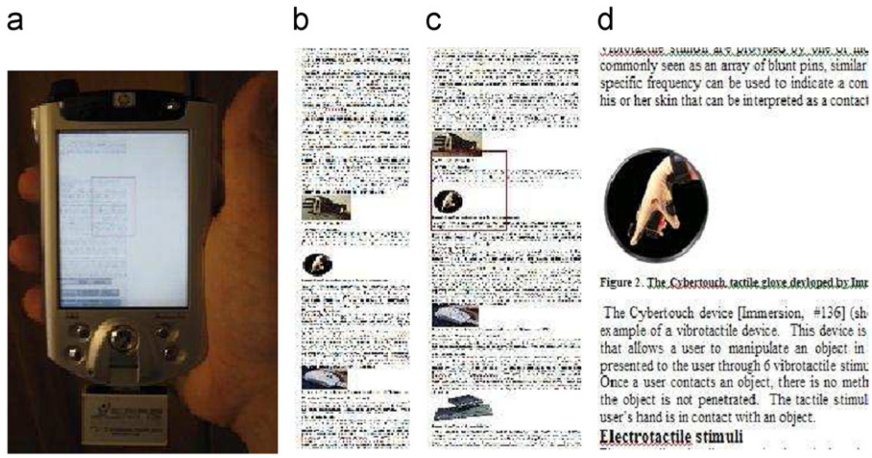

Fig. 1. A Pocket PC and an accelerometer attached to the serial port. (a) Screen shots of the document browser (b, c and d), (b) shows a red box moving rapidly over the picture, (c) shows the user has found the picture and landing there and (d) shows the zoomed-in picture.

a mouse cursor, which other researchers have used in their SDAZ implementations (Cockburn and Savage, 2003; Savage, 2004). A real-time application running on a PDA reads tilt sensor data continuously and maps the physical tilt angles of the device to the corresponding movement of the zooming window on the screen. Generally, this mapping depends on the speed of tilt sensor data reading from the serial port and it should be left to the operating system settings.

- *Zooming-window displacement to scroll speed*—This mapping describes the relationship between the tilt angle and the corresponding scroll speed in the document. As described by Zhai et al. (1997a, b), spring-centred joysticks or force sticks are generally better suited for velocity and higher order control systems and a nonlinear *dead-band* or *dead-zone* in the zero region makes the null position more distinct.
- *Scroll speed to magnification level*—This mapping describes the automatic zooming behaviour: the relationship between the scroll speed and the magnification of the document.

In this system the scrolling is performed by tilting up/down(pitch) and left/right(roll) the device and moving the zooming window in the desired direction and returning the device to the equilibrium point.[2]

An important issue in ZUIs is the location the system zooms back to, after the user returns the device to the equilibrium point. In an experiment fully demonstrated and analysed in Savage (2004) all users stated that they found the zoom-to-cursor (in our application zoom to zooming window) technique more intuitive than zoom-to-centre (see Figs. 2 and 3). For the animation of 'zoom to zooming-window position' in tilt-controlled SDAZ we can choose one of these methods:

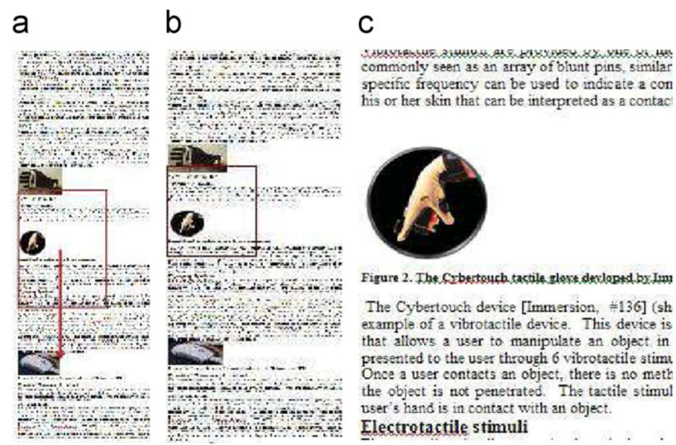

Fig. 2. Zoom-to-centre in tilt-controlled SDAZ. (a) Scrolling in the text document at maximum speed, (b) the device is returned to the neutral position. Zero scroll speed in document. Vertically falling to 100% magnification. (c) Finished zooming. Document is at 100% magnification.

- Keeping the magnification constant, scrolling to the zooming-window's position then falling to full magnification (see Fig. 4(a)).
- Optimal animation path as introduced by Wijk and Nuij (Section 2.1). In this animation, the system zooms out first, then zooms in again to achieve an optimal path. This animation, however, may cause the user to feel out of control because the system will most likely zoom out first then zoom in again to achieve an optimal path (see Fig. 4(b)).

We use zoom to zooming-window position using the method illustrated in Fig. 4(a); because (1) it provides a rapid movement to the position without being overwhelming, (2) it is the simplest path to implement and (3) Cockburn et al. (2005) reported that this method is

[2]Equilibrium point is a point from which a controlled variable will not change. Another name for equilibrium point is steady state (Ogata, 1990). An equilibrium point represents a state in which the system can be maintained using the defined action (Ogata, 1995). This point is calibrated based on the comfortable starting angle in the user's palm. This reduces irritating reflections from the Pocket PC screen.

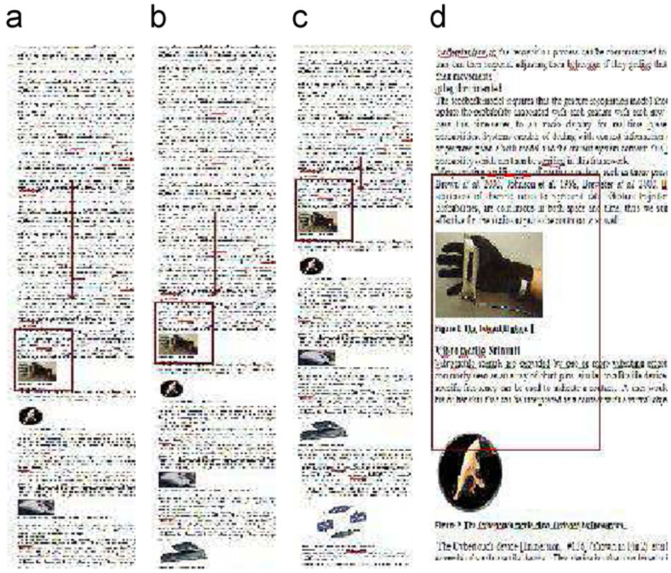

Fig. 3. Zoom to zooming-window's position in tilt-controlled SDAZ. (a) Scrolling in the text document at maximum speed, (b) the device is returned to the neutral position. Scrolling to the zooming-window's position at current magnification, (c) scrolling to the zooming-window's position with a constant magnification and (d) target point reached. Vertically falling to 100% magnification.

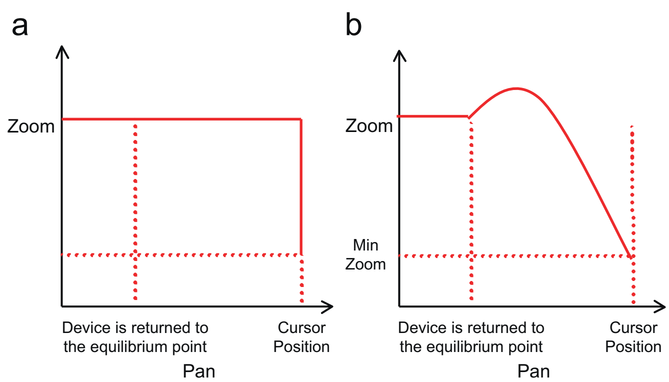

Fig. 4. Zoom–Pan space diagrams displaying zoom-in trajectories. The zooming-window's position represents the position the zooming window was over the document when the device returned to the equilibrium point in tilt-controlled SDAZ. (a) Zoom to the zooming-window's position keeping zoom level constant until zooming-window's position is reached. Adapted from Cockburn and Savage (2003). (b) Zoom to the zooming-window's position using Wijk and Nuij's optimal trajectory. Adapted from Wijk and Nuij (2003).

natural to use because the system continues scrolling at the same speed and in the same direction the user is scrolling in before, until it reaches the focus point, then falls to full magnification.

4.1. Design

State-space modelling is a well-established way of describing a dynamic system as a set of first-order differential equations. There is a wealth of knowledge and analysis techniques from systems theory, including designing controllers for multi-input/multi-output systems, optimal control, disturbance rejection, stability analysis and manual control theory (Brogan, 1991). State-space modelling allows us to model the internal dynamics of the system, as well as the overall input/output relationship as in transfer functions, so this method is an obvious candidate for the representation of the coupling between the user's speed with zoom level. The matrix formulation of state-space modelling is particularly useful for analysis of multi-variable systems.

4.1.1. State-space modelling

The general form of a state-space model can be written as two functions

$$\dot{x}(t) = f(t, x(t), u(t))$$
$$y(t) = g(t, x(t), u(t)) \tag{1}$$

The first is the state equation and the second is the output equation. The $u(t)$ is the input to the system, $x(t)$ is the state vector, t represents time and $f(\cdot)$ and $g(\cdot)$ state functions that can be linear or nonlinear. The more specific case of linear state-space representation of a system with p inputs, q outputs and n state variables can be written:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$
$$y(t) = C(t)x(t) + D(t)u(t) \tag{2}$$

$x(t)$ is called the *state vector*, $y(t)$ the *output vector*, $u(t)$ the *input* (*or control*) *vector*, $A(t)$ the *state matrix*, $B(t)$ the *input matrix*, $C(t)$ the *output matrix* and $D(t)$ the *feedthrough* (*or feedforward*) *matrix*. In this general formulation all matrices are time variant. We can analytically investigate the local dynamics for different operating points by, for example, looking at the eigenvalues of the A and B matrices to check for oscillatory (eigenvalues are complex conjugate pairs) or unstable behaviour (the real part of the eigenvalues are in the right half plane—i.e., eigenvalues are positive) (Ogata, 1990).

4.1.2. A state-space model for tilt-controlled SDAZ

The document viewer to examine the state-space model for the tilt-controlled SDAZ was implemented to browse PDF, PS and DOC files which had been converted to an image (PNG or BMP) file (see Eslambolchilar, 2006 for more information).

SDAZ can be simulated as a flying object like a bird or an airplane (see Fig. 5). If the object flies to greater heights it gets a wider view than lower heights (Fig. 6) and for that it has to fly fast. Thus two variables, velocity and field of view, are coupled together.

If we present F_1 and F_2 for horizontal and vertical vectors of the external force applied to the mass, R for air resistance which provides damping effects, m for mass, v for velocity of the object, a for acceleration, z for zoom and

$\dot{z}$ for rate of change of zoom then we can write Newton's second law of motion in the horizontal direction for the object in Fig. 5:

$$ma = F_1 - Rv \quad \text{or}$$
$$a = \frac{F_1}{m} - R\frac{v}{m} \tag{3}$$

In the vertical direction we can write

$$m\dot{z} = F_2 - Rz \quad \text{or}$$
$$\dot{z} = \frac{F_2}{m} - R\frac{z}{m} \tag{4}$$

We assume that effects of gravity are negligible. Furthermore we can assume F_2 is a function of F_1 and velocity (this assumption will couple level of zoom to speed of movement as well as tilt input, e.g., in higher speed the field of view, i.e., the area of the triangle increases and vice versa):

$$F_2 = cF_1 - bv \tag{5}$$

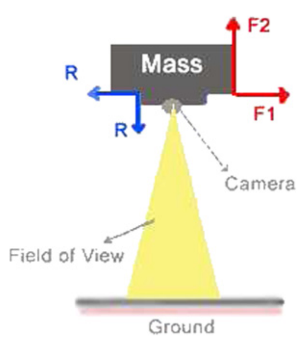

Fig. 5. Simulating SDAZ as a flying mass. F_1 and F_2 are horizontal and vertical vectors of the external force, F applied to the mass and are coupled together. R is the air resistance which provides damping effects.

where b is a coefficient and c is a scaler. Then we can rewrite Eq. (4) as below:

$$\dot{z} = c\frac{F_1}{m} - b\frac{v}{m} - R\frac{z}{m} \tag{6}$$

To prevent accidental movements in tracking tasks with spring-centred input devices, e.g., joysticks or tilt sensors, we can add friction effects in both vertical and horizontal directions. We consider two different frictions in different directions; R for horizontal direction and R' for vertical direction. Then Eq. (6) can be rewritten as below:

$$\dot{z} = c\frac{F_1}{m} - b\frac{v}{m} - R'\frac{z}{m} \tag{7}$$

The inputs to the system are the tilting angles measured using an accelerometer attached to the serial port of PDA, or the stylus position on the PDA's touch screen. The state variables chosen are $x_1(t)$ for position of zooming window, $x_2(t)$ for speed of scroll and $x_3(t)$ for zoom. u represents input, F_1 (pitch tilting angle), and the state equations are

$$x_2(t) = v = \dot{x}_1 \tag{8}$$
$$x_3(t) = z = f(u, x_2, x_3) \tag{9}$$

Thus, the zoom-level is a function of current level of zoom, velocity and tilt, u.

An initial suggestion is to reproduce the standard second-order dynamics of a mass–spring–damper system and give the scrolling movement and zoom level some inertia to provide a physically intuitive interface. The first time derivative of the state equations given in Eqs. (3)–(6) can be rewritten as below, as a linearisation of the system

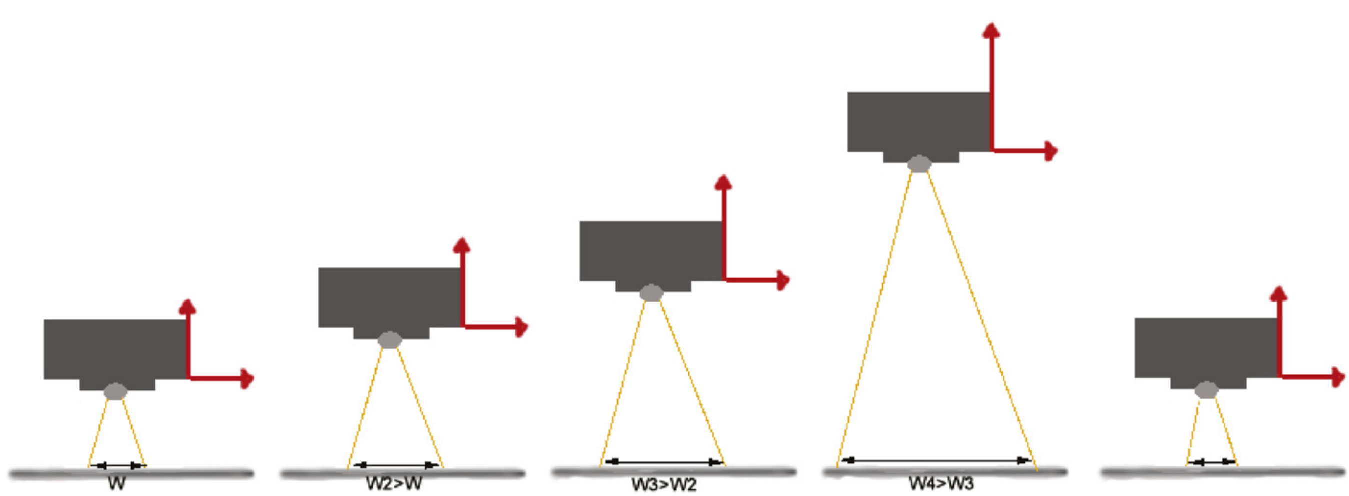

Fig. 6. View domain of a flying object. If the object flies to greater heights then it gets a global overview of the ground but in lower heights it only gets a local view of the ground.

at a given velocity and zoom:

$$\dot{x_1}(t) = v = x_2(t) \tag{10}$$

$$\dot{x_2}(t) = a = \dot{v} = \frac{-R}{m}x_2(t) + \frac{1}{m}u(t) \tag{11}$$

$$\dot{x_3}(t) = \dot{z}(t) = \frac{-b}{m}x_2(t) - \frac{R'}{m}x_3(t) + \frac{c}{m}u(t) \tag{12}$$

The standard matrix format of these equations is

$$\begin{pmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{-R}{m} & 0 \\ 0 & \frac{-b}{m} & \frac{-R'}{m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \\ \frac{c}{m} \end{pmatrix} u \tag{13}$$

This shows how a single-DOF input can control both velocity and zoom-level. The non-zero off-diagonal elements of the A matrix indicate coupling among states, and the B matrix indicates how the inputs affect each state. This example could be represented as having zoom as an output equation, rather than state, and the coupling between zoom and speed comes only primarily the B matrix, which is not particularly satisfying.

The next step is finding suitable values for R, R', m, b and c coefficients in the state-space model to make the system stable and controllable, i.e., the controllable matrix should be full rank (Eslambolchilar, 2006). One way to choose the right settings is observing the simulated system's behaviour.

4.1.3. Simulation

A well-designed model should generate similar behaviour on the actual system/device as the simulated system. For this purpose we set the coefficients $m = 10\,\mathrm{kg}$, $R = 10\,\mathrm{kg\,s^{-1}}$, $R' = 10\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 3$ to check the system behaviour to different input signals in real and simulated environment.

Fig. 7 presents the time domain response of the simulated system in MATLAB and implemented system on a PDA to a varying step inputs, which means scrolling down on a document, staying in a certain speed, landing on a figure and scrolling up by tilting the device. In the off-line simulation the input signal is without noise; however, in the real experiment, the tilt sensor data is noisy. Fig. 7(c) shows the sensitivity of the dynamic model in the real experiment to noise. A few milliseconds after tilting the device, both velocity and zoom have reached to a steady-state and this is one of main advantages of feedback dynamic systems. The relationship between the velocity and zoom in both simulated and actual system are linear indicating the system's behaviour is linear. Also, despite changes in the sign of input data the relationship between velocity and zoom remains linear.

Fig. 8 presents the time domain response of the simulated system in MATLAB and implemented system on a PDA to a varying step inputs with different settings as mentioned above, $m = 10\,\mathrm{kg}$, $R = 10\,\mathrm{kg\,s^{-1}}$, $R' = 1\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 3$ (We have only reduced friction in the vertical direction). The system is controllable, but the level of zoom is not easy to control and small changes in tilt led the system to zoom out quickly. Reducing the friction reduced the response lag, which allowed the mass to zoom out too quickly (compare with results in Fig. 7).

The linear dynamic model presented here can be extended to a general nonlinear format. In SDAZ (Igarashi and Hinckely, 2000; Wijk and Nuij, 2003; Wallace et al., 2004), the document velocity as a function of control input (mouse displacement, tilt angle, or stylus displacement) tend to be static, linear, or piecewise linear functions (Igarashi and Hinckely, 2000; Wallace et al., 2004). The general state-space equations in (9) can be written in a nonlinear form where functions mapping zoom to velocity or vice versa, and functions mapping friction to zoom and velocity can be nonlinear dynamic functions. Using these more interesting behaviour can be obtained, such as those elegantly derived by Wijk and Nuij (2003).

5. Model-based interactive behaviour in SDAZ

A well-designed control system should be able to operate successfully in a wide variety of situations by detecting the specific situation that exists at any instant and servicing it appropriately (Narendara and Balakrishnan, 1997). External disturbances, changes in subsystem dynamics, parameter variations, and so forth, are examples of different unknown contexts in which the system has to operate (Narendara et al., 1995). Therefore, switching and tuning parameters quickly and accurately play an important role in the stability of the system. In switching, the problem is to determine when to switch and what to switch to. In tuning, the problem is to determine the rule by which the parameter value is to be adjusted at each instant (Narendara and Balakrishnan, 1997).

5.1. Control modes in SDAZ

In interaction between user and model-based text browser the user provides raw tilt input data as action via accelerometer and the user's action controls what s/he perceives from the display. The state-space dynamic system representation couples the user's intention to SDAZ via tilt input. In this task the user is either looking for specific piece of information (searching) or targeting something on the display. We define a few different modes of control in this example: no action, free motion, velocity and diving control. Fig. 9 presents these modes of control and transitions among them.

Each of these control modes needs special state-space parameter settings. In the beginning the user is in the no-action state because there is no input from the user to the system. By tilting the device, therefore increasing the speed, the user goes to the free motion control mode. Here, $\dot{v}$, $\dot{z}$ are functions of current input, velocity and level of zoom or $f(u, v, z)$. Free motion control is a transient state and the user may go back to no-action or velocity control mode

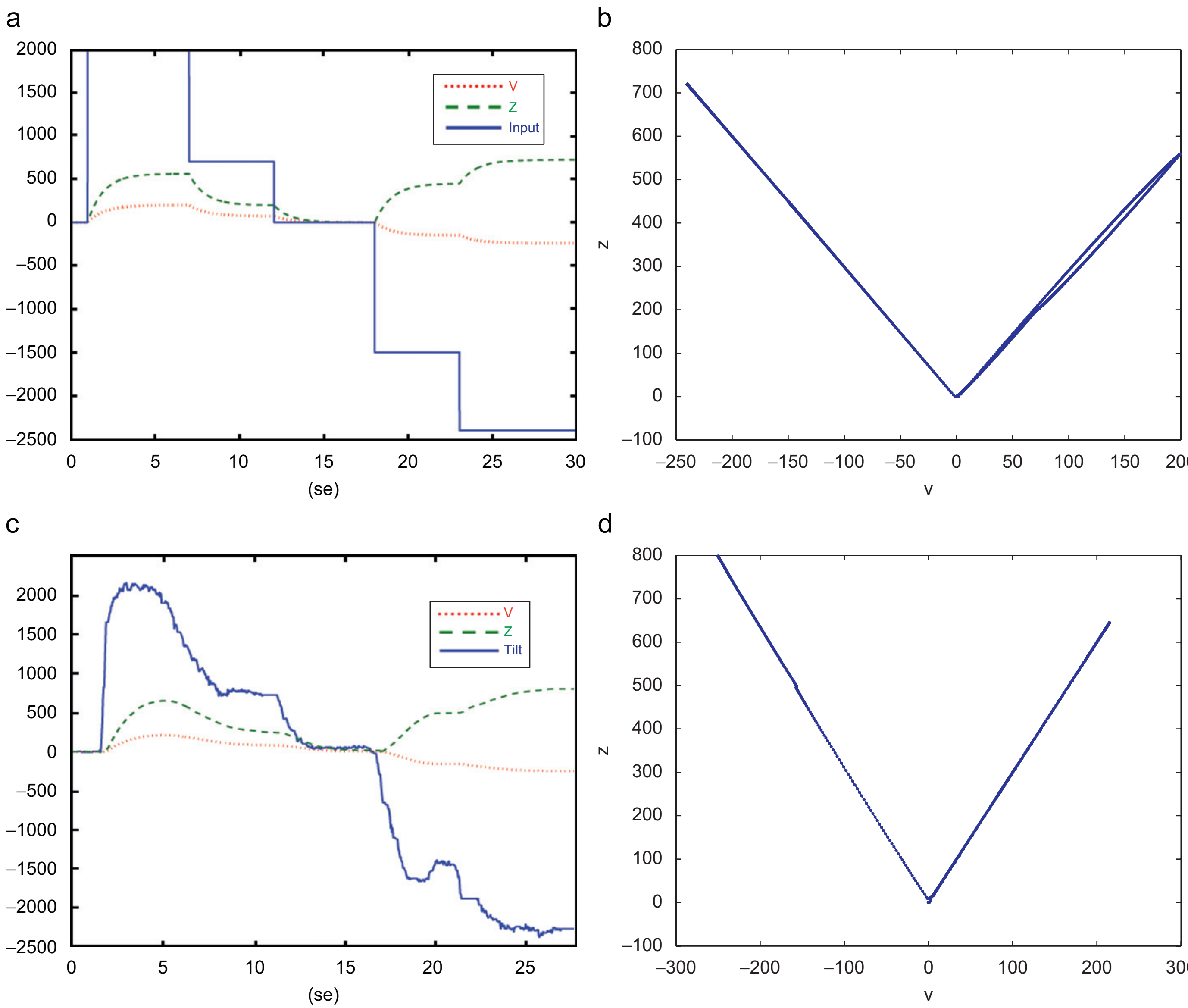

Fig. 7. Behaviour of SDAZ model to a step input (a), (b) simulated model in MATLAB, and (c), (d) implemented model on a PDA ($m = 10\,\text{kg}$, $R = 10\,\text{kg}\,\text{s}^{-1}$, $R' = 10\,\text{kg}\,\text{s}^{-1}$, $b = 3\,\text{kg}\,\text{s}^{-1}$ and $c = 3$).

afterward. In the velocity control mode the user is usually looking for some piece of information and s/he may spend a long time browsing at a steady speed in the document searching for a certain data. In this mode, the user controls v_{ref} as a desired velocity manually and the controller maintains this velocity automatically and completes the scrolling task with the desired speed for the user. This mode needs state feedback to augment control behaviour, hence the state should move toward the reference value, v_{ref}. We can create a control law such $u = L(r - x)$ and write new state equations (see Fig. 10):

$$\begin{aligned}\dot{x} &= Ax + Bu = Ax - BLx + BLr \\ &= (A - BL)x + BLr \end{aligned} \tag{14}$$

In classic control the simplest form of L is *proportional control* and $L(r - x)$ is control input. For velocity control mode we have $L = [0 \;\; l \;\; 0]$ and $r = [0 \;\; v_{\text{ref}} \;\; 0]^{\mathrm{T}}$ ('T' indicates the matrix transpose). The reference velocity is a function of input $v_{\text{ref}} = s(u)$ and changes in the speed and zoom level can be a linear or a nonlinear function of v_{ref}, current velocity and zoom level, or $\dot{v}, \dot{z} = h(v_{\text{ref}}, v, z)$. After changing the control law, the system should be both stable and controllable. In this example application with given $m = 10\,\text{kg}$, $R = 10\,\text{kg}\,\text{s}^{-1}$, $R' = 10\,\text{kg}\,\text{s}^{-1}$, $b = 3\,\text{kg}\,\text{s}^{-1}$ and $c = 3$, we choose $L = 10$, which satisfy stability and controllability conditions discussed earlier.

After finding the target (i.e., the zooming window is over the target point) the user may slow down or stop tilting thus the system scrolls to the zooming window with a constant magnification level. When the zooming window is reached, the position of the target the user is interested to land on becomes the reference signal, x_{ref}, and the controller changes the current position value to the

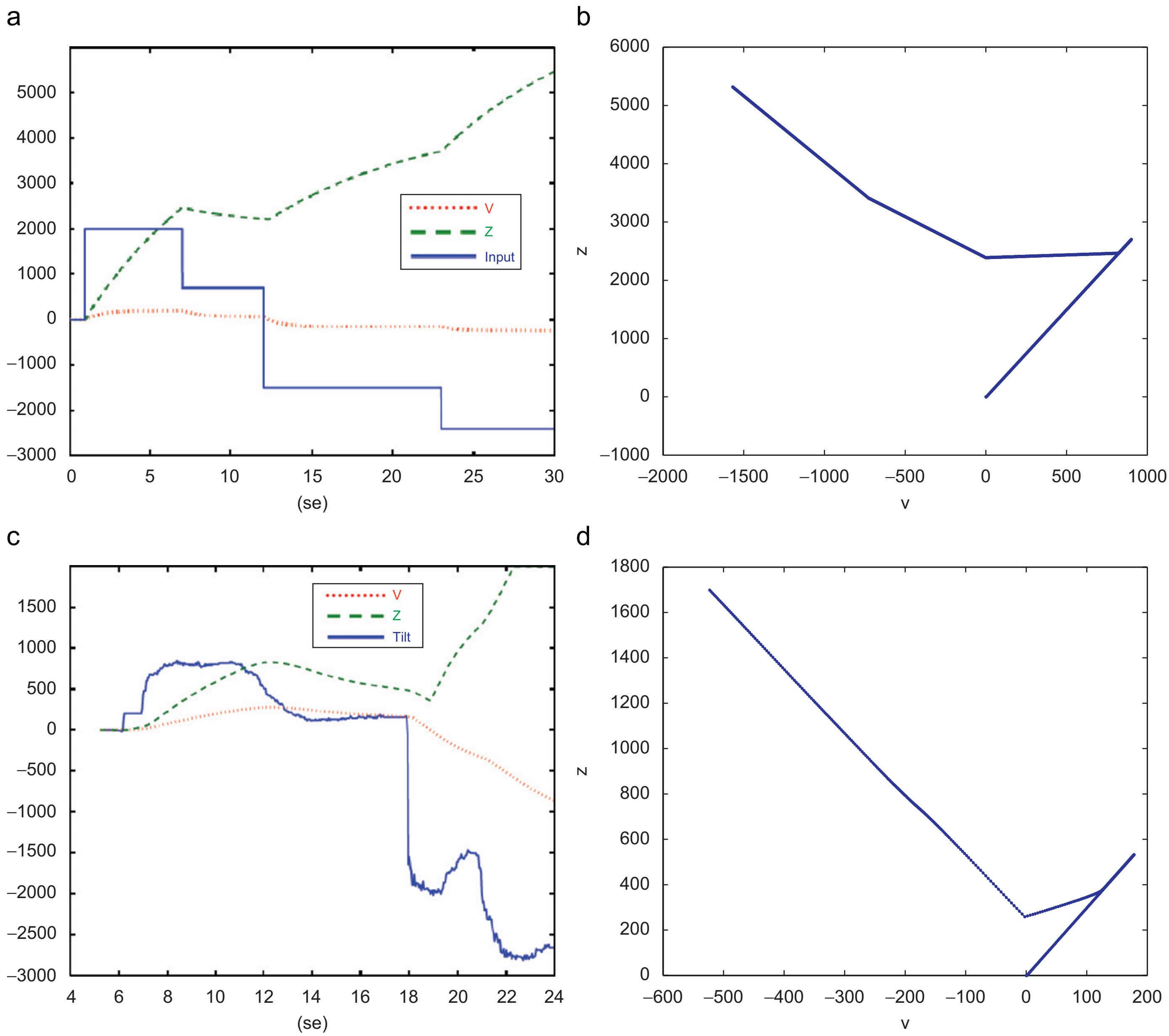

Fig. 8. An example of bad coefficients settings. (a), (b) simulated SDAZ model in MATLAB, (c), (d) implemented SDAZ model on a PDA ($m = 10\,\mathrm{kg}$, $R = 10\,\mathrm{kg\,s^{-1}}$, $R' = 1\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 3$).

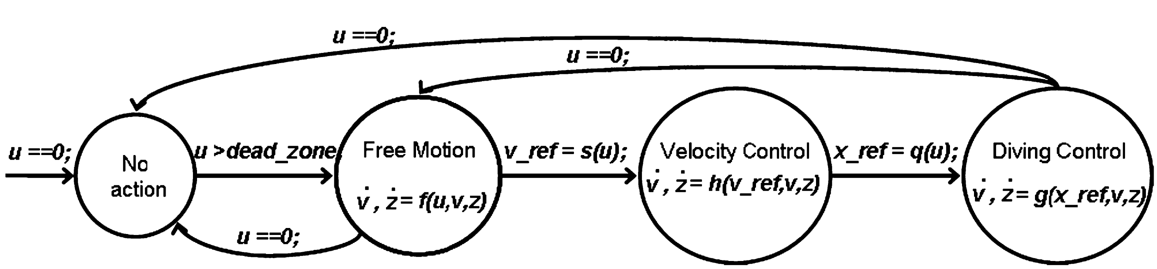

Fig. 9. Control modes in tilt-controlled SDAZ and transitions among them.

reference position, x_{ref} (i.e., moves the state position variable toward the reference value, x_{ref}), and smoothly dives toward it vertically. Similar to the velocity control mode, we create a control law $u = L(r - x)$ by introducing $L = [l\ \ 0\ \ 0]$ and $r = [x_{\mathrm{ref}}\ \ 0\ \ 0]^{\mathrm{T}}$. The reference position is a function of input $x_{\mathrm{ref}} = q(u)$ and changes in the speed and zoom level can be a linear or nonlinear function of reference position, current velocity and zoom level, or

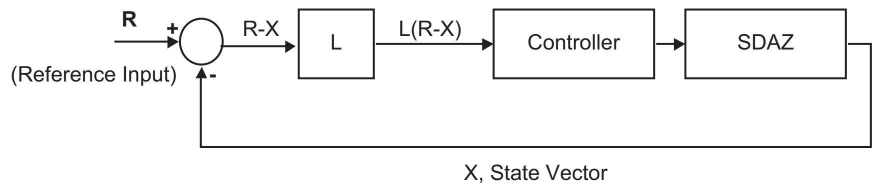

Fig. 10. An example of proportional control, where the controller uses state feedback to augment control behaviour, by making the state move toward some reference value r, and creating a new control law such $u = L(r - x)$.

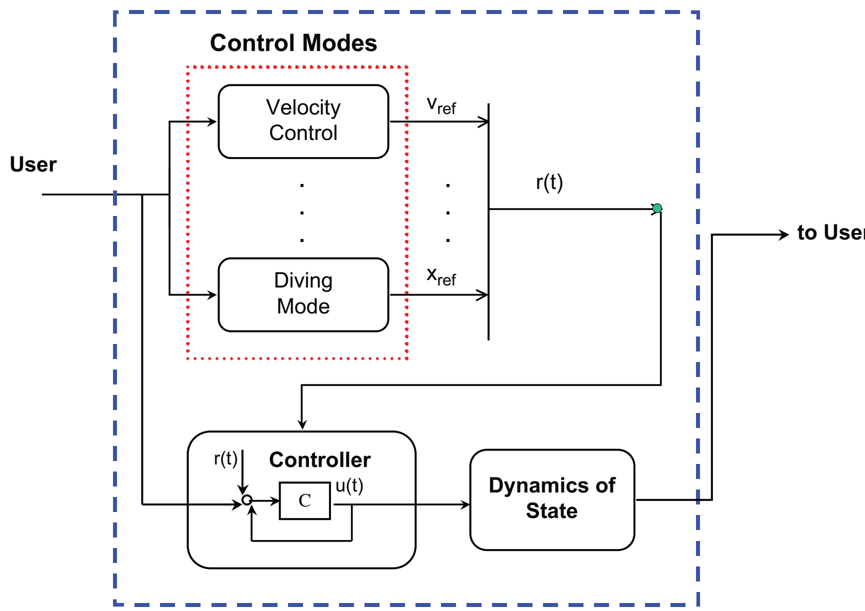

Fig. 11. Mode switching in model-based SDAZ.

$\dot{v}, \dot{z} = h(x_{\mathrm{ref}}, v, z)$. In this example application with given $m = 10\,\mathrm{kg}$, $R = 10\,\mathrm{kg\,s^{-1}}$, $R' = 10\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 3$, we choose $L = 3$, which satisfy stability and controllability conditions. After diving the user may go back to no-action mode or by any acceleration above a threshold, on route toward a goal, may switch to free motion mode and start browsing again (Fig. 9).

Fig. 11 illustrates how based on the user input and control states a control mode is selected. After switching, coefficients in the state-space model are set to new values (tuning). Also, the selected mode is coupled to SDAZ controlled variables, i.e., speed of scroll and zoom. For instance, after diving toward the target the controller adjusts the zoom level to stay in the maximum level (100% zoom), while if the user 'breaks out' into free motion, the zoom level is decreased smoothly to a lower level. Fig. 12 presents examples of changing modes and following reference signals.

5.2. Calibration, performance measures and state-space approach

SDAZ has many parameters that can be tuned, usually treated as a series of interacting, but essentially separate equations. The state-space formulation allows multiple variables, and derivative effects (e.g., position, velocity, acceleration) can be coupled with zoom level, without any further coding, by just changing the entries of the A and B matrices, simulating combinations of springs, masses and damping effects.

To enhance the smoothness of the transition between the global overview and the magnified local view after a mouse button is pressed, Cockburn and Savage (2003) use a 'falling' speed, and Igarashi and Hinckely (2000) place a limit on the maximum time derivative of zoom, with similar effect. The falling rate was calculated using trial and error—if the rate was too fast, the user felt motion sickness and lost their place in the document, whereas it being too small led to a sluggish interface. This can be represented as a straightforward switch to a particular parameterisation of the A matrix, which can be tuned to give an appropriate exponential decay in velocity or zoom. Related problems include rapid zooming in and out when making a rapid change of direction (Igarashi and Hinckely, 2000).

In the state-space representation, our basic assumption is that zoom should lead speed when speed increases, in order to avoid extreme visual flow. Zoom should, however, lag speed when $|v|$ decreases, to allow the user to slow down but still maintain the overview. This also allows, for example, the user to zoom out, without changing position

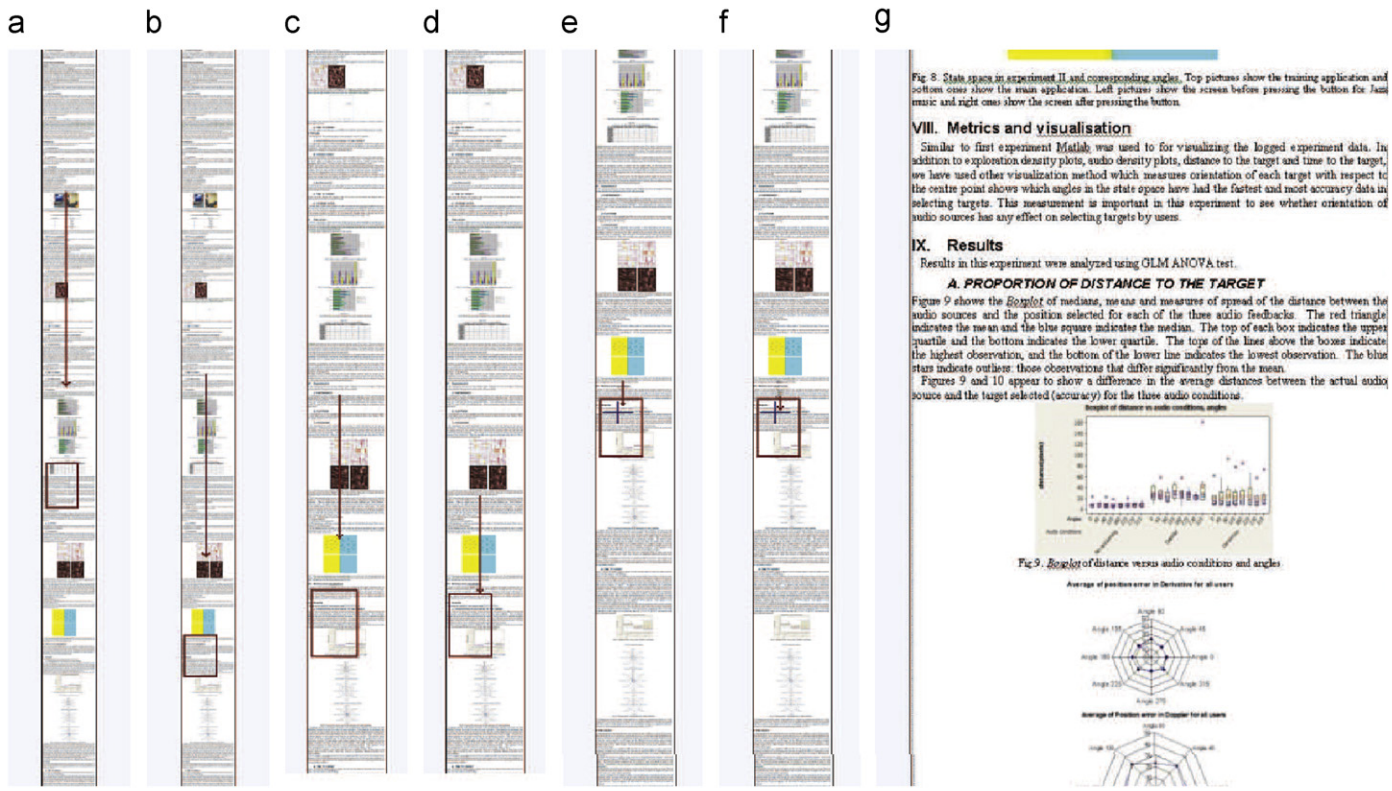

Fig. 12. An example of reference signal and mode switch. (a) The user has started velocity control mode by tilting the device. The red vertical vector represents the reference velocity (v_{ref}) and length of this vector represents speed of scroll (the higher the speed, the longer the vector appears on the screen). (b) The user tilts the device faster and increases v_{ref} and moves to a higher height (note length of the vector). Then the controller maintains the desired velocity automatically and complete the scrolling task with that velocity for the user. (c) Where the user is interested in some details in the text he tilts the device slowly and reduces v_{ref}, then the controller maintains a new reference value for the velocity automatically. (d) The user has found the target and it is located inside the zooming window and she returns the PDA back to the equilibrium point thus the controller scrolls to the zooming window with a constant magnification level. (e) When the zooming window is reached the position of the target becomes the reference position (x_{ref}, which is shown as a small blue cross) and the controller changes the current position value to the reference position. (f) The blue cross, x_{ref} is over the target and the controller dives toward the target position vertically. (g) The user has landed on the target and gets a magnified local view from the picture.

in the document, by repeated positive and negative acceleration. Having an initial dead zone helps to filter out small tilt angles or small zooming-window displacements, which cause zero or slow responses at small input level and fast responses at large input level. In order to move more rapidly through the document at high levels of zoom, here, we adapted B by making c in Eq. (13) a function of velocity. When speed is above the dead-zone threshold (here set to 0.1), $c = 3$ but below this threshold $c = 0$, where speed of scroll and level of zoom do not change. We wish to avoid rapid drop effects when the user changes direction. To achieve this, we change c to be $-0.5 \times c$, when the sign of velocity and input differ.

Igarashi and Hinckely (2000) and Wallace (2003) report the 'hunting effect' problem when users overshoot the target due to the system zooming in as the user slows, the user then rapidly adjusts behaviour to compensate, which causes the system to zoom out again. One approach to this would be to switch to a 'diving' control mode if $\mathrm{d}z/\mathrm{d}t < zthresh$, where $c = 0$, preventing zooming increases, unless a major change in velocity, occurs, which would switch the control mode back to velocity control. Fig. 13 presents effectiveness of this mode in a real implemented system on a PDA.

In Fitts' (1954) law experiments, the goal is to capture the target in minimum time. Thus, the performance in this discrete positioning could be described as minimising the function:

$$J = \|t_f - t_0\| \tag{15}$$

t_0 and t_f are initial and final time in Fitts' law task, respectively. In our continuous tracking task, we consider minimising the time of completing the task (Eq. (15)) as well as

$$J_s = \sum_{t=t_0}^{t_f} \|uf_{t+1} - uf_t\| \tag{16}$$

$$J_a = \frac{\sum_{t=t_0}^{t_f} \|uf_{t+1} - uf_t\|}{t_f - t_0} \tag{17}$$

t_0 and t_f are initial and final time and uf_t is the user's tilt input to the system, which has been filtered. (A low-pass filter with cut-off frequency 40 Hz was employed.) In the above equations the total sum of changes or mean sum of changes in the tilt input is minimised if the interaction is smooth.

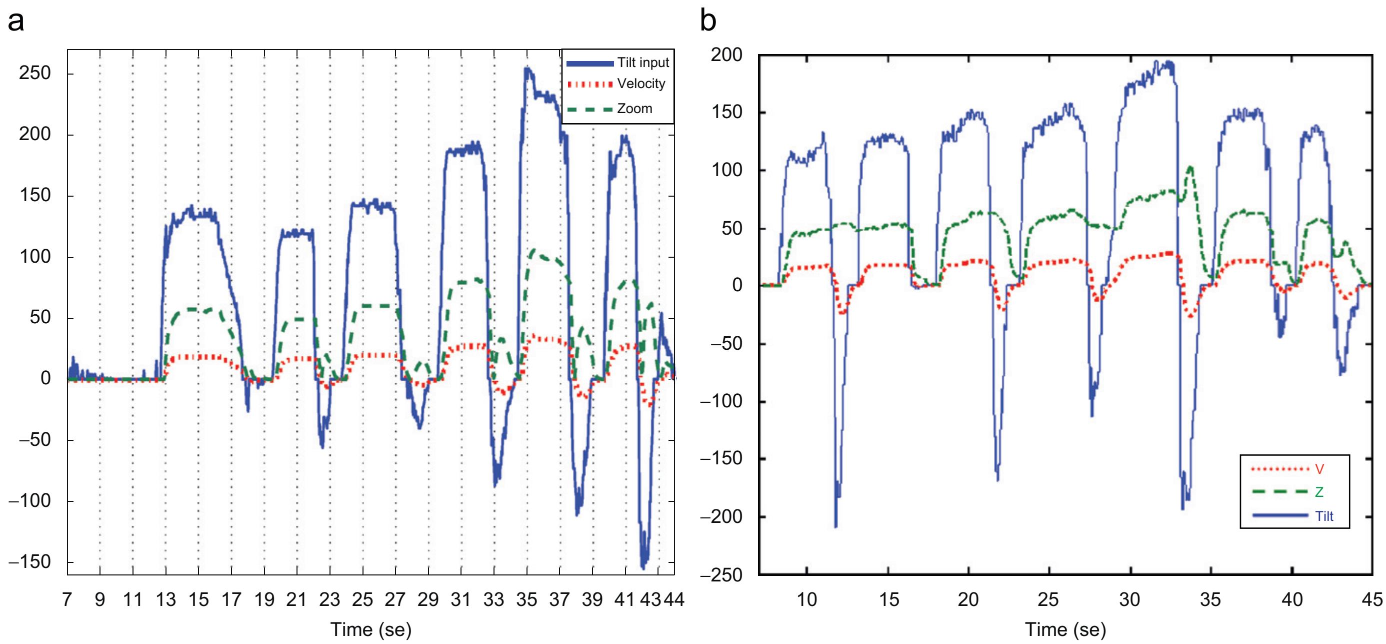

Fig. 13. Real developed system's behaviour with settings $m = 10\,\mathrm{kg}$, $R = 10\,\mathrm{kg\,s^{-1}}$, $R' = 10\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 3$ on a PDA. (a) Hunting effect happens around time 31 and 42 s when the 'diving' control is not considered. (b) Hunting effect disappears in a real system developed on the PDA after adding diving mode.

In the example shown in Fig. 13, adding the diving mode reduced the cost; without this mode the total sum of changes was 18 324 unit where after adding this mode the sum became 14 809 unit. This also explains the users' low activity in the application having this mode.

5.3. Reference signals as inputs

The reference signal is a goal or a target the user is aiming to reach. In control engineering, engineers design systems to control variables with respect to reference signals, and they plan to be able to manipulate the references (inputs) when they want to get the system to control a variable at a different level (e.g., in thermostats the user enters the temperature he wants and the thermostat automatically adjusts the room temperature) (Ogata, 1990). In these systems the controller may support the user to complete the task with less effort by changing the interpretation of the inputs to being reference values, rather than control commands. In modern aircraft controllers there are different interpretations of aircraft controls depending on flight mode (e.g., take off, altitude-hold and so forth) and blend seamlessly between modes (Tischler, 1994).

In tilt-controlled SDAZ there are two conditions that the user is aiming to achieve: a desirable speed of scroll for browsing and a target that the user is trying to land on. For these two variables there are two individual reference signals, v_{ref} and x_{ref} and these signals give an example of an intuitive mode transitions. As an example, tracking the 10th header in the document and its time series of data are shown in Fig. 14. Here, while the user tilts the device at a constant angle, the controller maintains the desired velocity automatically, e.g., v_{ref} and completes the scrolling task with the velocity the user wants to achieve, rather than the user having to do this. Any change in the tilt angle for the controller means the user wants to change the v_{ref}. Similarly, in the diving mode, the system can reinterpret tilt input to change the desired position while zooming into a point of interest after browsing. During these mode transitions, in order not to have a sharp transition at the changeover, the offset variables[3] have to be such that they are enough to cancel out the input provided by the user at the point they enter diving control mode. After transition, the offset values gradually reset to a more sustainable position, but in such a way that the user gradually returns the device back to the equilibrium position. This means that as the user performs various tasks they switch between control modes automatically, and their inputs have different meanings, but that the transitions are always smooth and natural, and the user is often not even aware that their movements are having different effects in different modes.

6. Example application—a document browser

To evaluate the efficiency of tuning and calibrating coefficients, and mode switching using state-space

[3]The offset is the equilibrium point, or bias that is needed to bring a system to rest. Trim variables are another name for offset variable more often used in the aerospace field (the settings of flaps so forth that keeps an aircraft in a stable, or trim state) (Hess and Chan, 1988; Aponso et al., 1990; Bradley, 1996; Padfield, 1996).

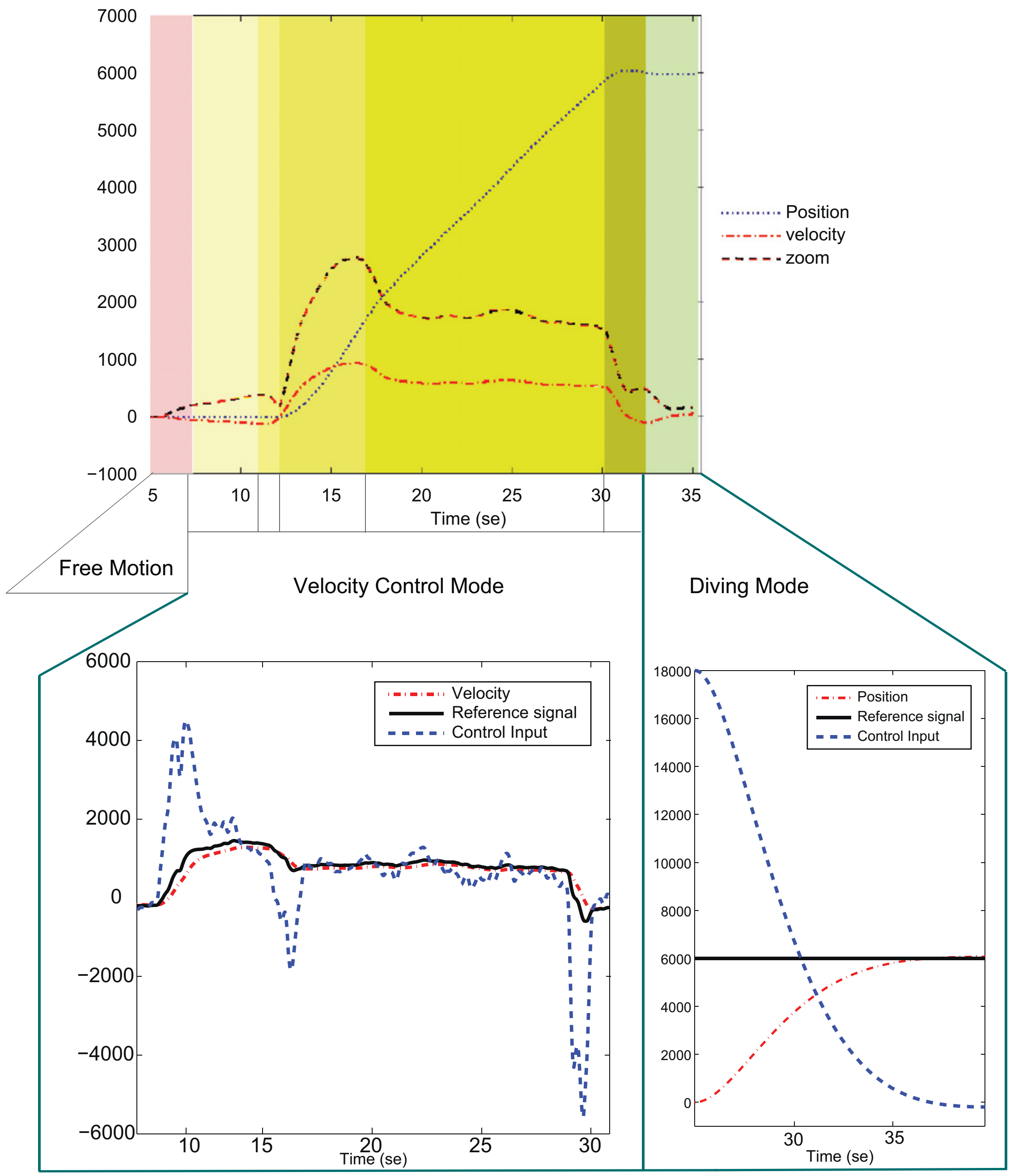

Fig. 14. A user's trajectory and position, velocity and zoom-level data when he is looking for 10th header in the document. Different colours highlight different modes of control or changes in the reference signal. Bottom figures highlight the changes in v_{ref} and x_{ref}. In the velocity control mode, the system is moving velocity state variable toward the reference input. In the position control mode, the user is over the target and dives toward it and the system moves position state variable toward the reference signal.

modelling, we not only simulated the model in MATLAB, but also developed a touch-screen controlled and tilt-controlled SDAZ for browsing a long document on the PDA. We asked eight users to work with the document browser using tilt and touch input. They were told how the tilt, touch and SDAZ application works in the training session. It was also explained the application has been developed based on a flying object model. We examined different parameter settings and the effect of calibration on user performance as well as augmented control and mode switching.

6.1. Calibration and performance measures

The users were asked to work with the document browser using tilt-controlled SDAZ with different tuned and calibrated parameter settings and target two figures. We started with settings: $m = 10\,\text{kg}$, R and $R' = 1\,\text{kg}\,\text{s}^{-1}$, $b = 3\,\text{kg}\,\text{s}^{-1}$ and $c = 3$. Fig. 15(a) presents one of users browsing behaviour in the task. The user has not been comfortable in the interaction and he complained it was almost impossible to land on figures because any slight tilt

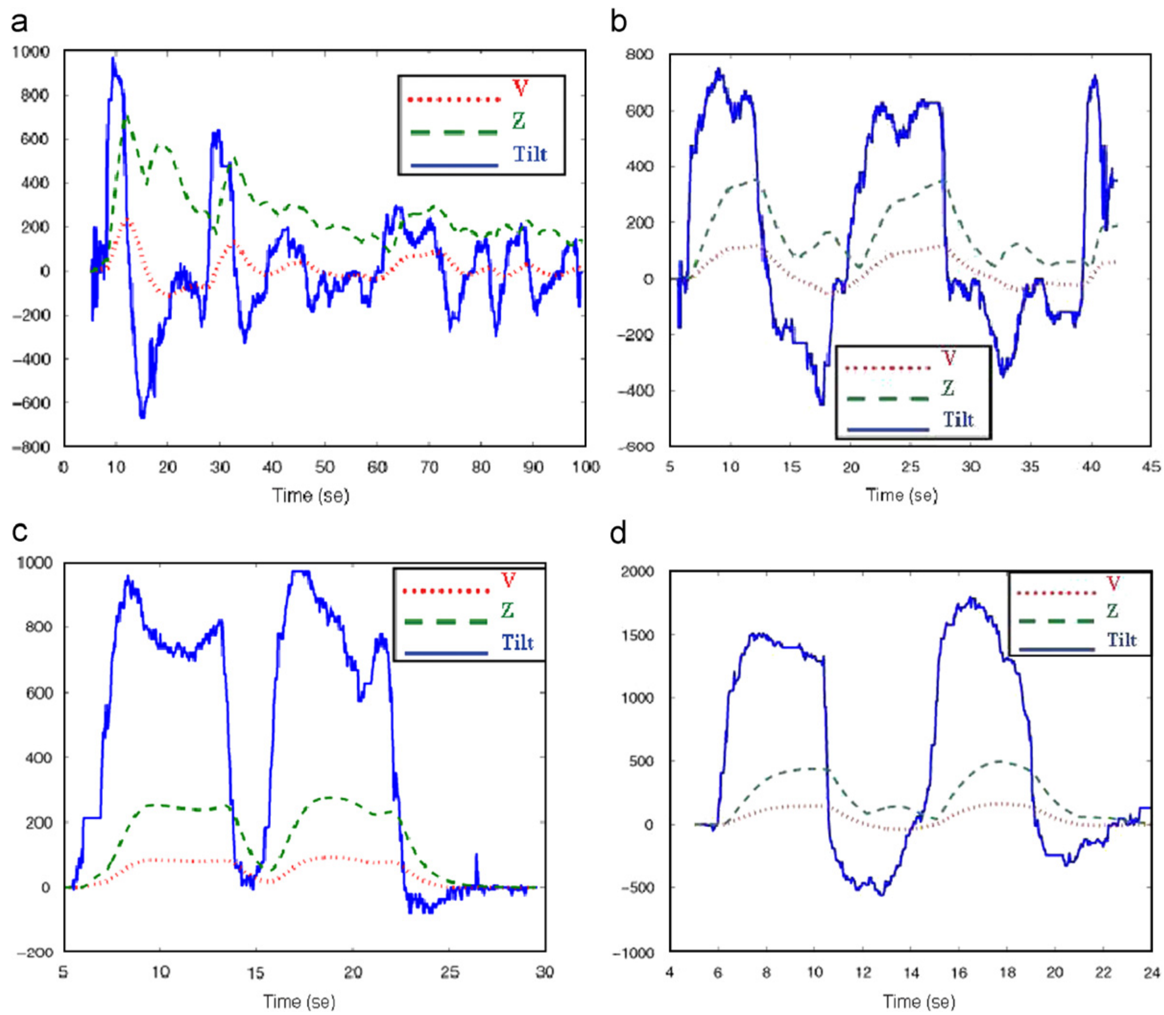

Fig. 15. User's tilt behaviour and changes in velocity and zoom. (a) Controller with settings $m = 10\,\mathrm{kg}$, R and $R' = 1\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 3$, (b) R and R' change to $5\,\mathrm{kg\,s^{-1}}$, (c) R and R' change to $10\,\mathrm{kg\,s^{-1}}$, and (d) controller with settings $m = 1\,\mathrm{kg}$, R and $R' = 1\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 0.5$.

was causing sluggish behaviour in zoom. The total sum of input changes, J_s, for this user was 15 304 unit.

We changed the coefficients R and R' to $5\,\mathrm{kg\,s^{-1}}$ and asked another user to repeat the task. Fig. 15(b) presents the user behaviour in this task and there is a slight improvement in controlling the task. J_s for this user in this task was 8118 unit, which is much lower than previous user's activity and J_t, the time taken to complete the task is considerably shorter than previous example. In a different setting for R and $R' = 10\,\mathrm{kg\,s^{-1}}$ with a different user for the same task we could get the behaviour presented in Fig. 15(c). This user's performance was much higher than the two previous ones with only 5265 unit in his total sum of input changes and they took less time than the two previous subjects to complete the task. Other settings for this system could not bring the cost down (Fig. 15(d)). This simple example shows, first, that the controller with settings $m = 1\,\mathrm{kg}$, R and $R' = 10\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 3$ is stable (as we also saw in the simulated section); second, the cost function (either the time required to complete the task, equation or total sum of changes in the input) is minimised for this setting. For the next experiment we chose the most stable setting, i.e., $m = 1\,\mathrm{kg}$, R and $R' = 10\,\mathrm{kg\,s^{-1}}$, $b = 3\,\mathrm{kg\,s^{-1}}$ and $c = 3$.

6.2. Augmented control and reference signal

In another experiment the users were asked to browse through the document and target a specific header in the document using the tilt input, where augmented control is active. Fig. 16 shows examples of tracking two headers in the document by two participants. As the user tilts the device, the controller maintains the desired velocity automatically and completes the scrolling task with the velocity the user wants to achieve, rather than the user having to do this. As soon as the user reaches over the target (presented as x_{ref}) and dives toward it the system moves position state variable toward the reference signal. Additionally, users were asked to track seven headers in the document using tilt and touch inputs and with and without augmented control. Fig. 17 presents some examples. Users who did the experiment without augmented velocity control suggested that adding a control option or a switch to control the zoom level with velocity and tilting angles

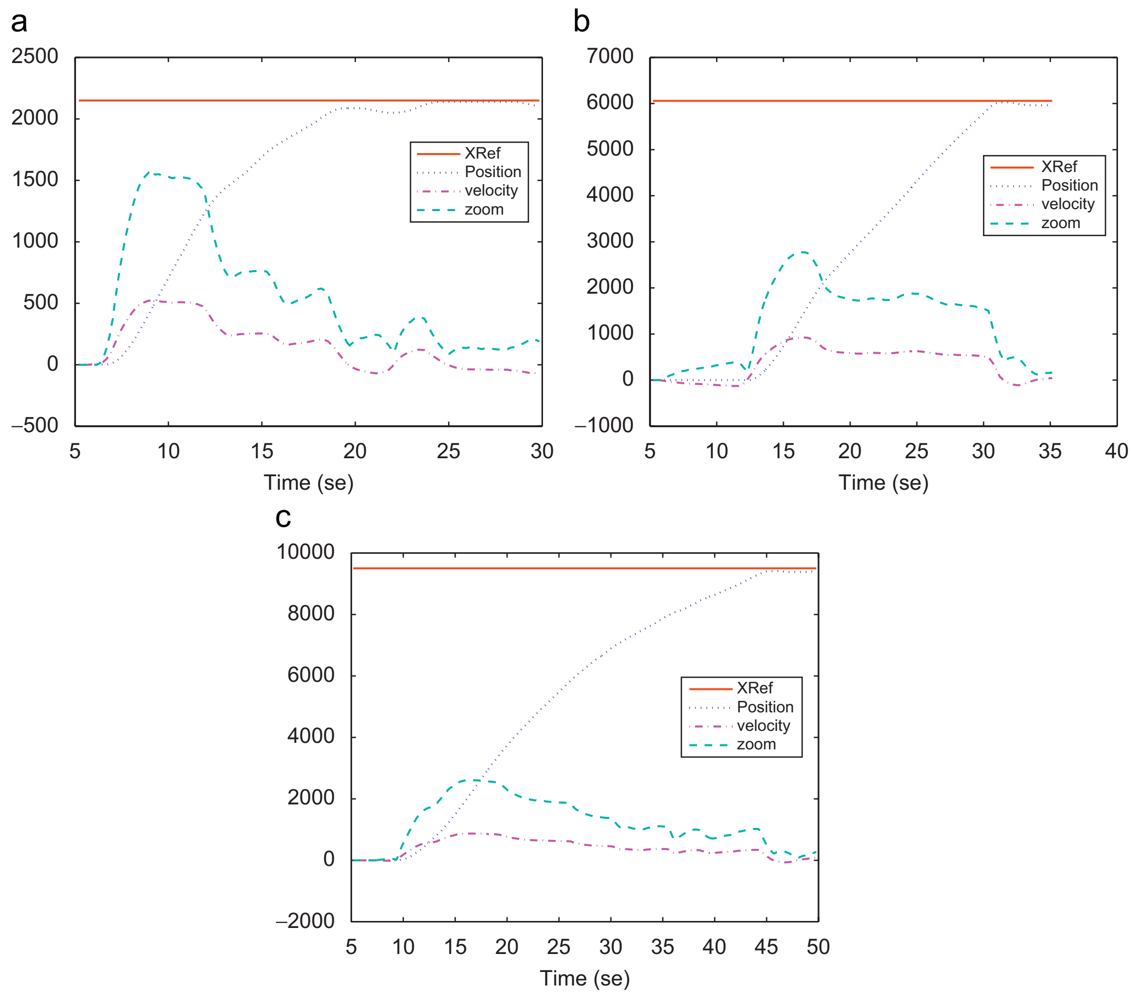

Fig. 16. Time series of zooming-window position, velocity and level of zoom while users are tracking headers in the document. (a) Targeting 5th header, (b) targeting 10th header and (c) targeting 12th header.

will make the system more comfortable to use. These users proposed that if they could control level of zoom by tapping on the screen or pressing a key on the PDA, the application would be easier to use. Fluctuations in their browsing behaviour (Fig. 17) indicate that controlling the zoom level was difficult, and hunting behaviour appeared when users tried to land on targets (e.g., $t = 20, 40, 85$ s in Fig. 17(b)). In contrast, the users who did their experiments with augmented velocity control were satisfied with the application in both tilt-controlled and touch-screen-controlled; because they commented that they easily landed on the goal without tilting back and forth to adjust their targeting. This also explains accurate landings in Fig. 16.

Users found touch-screen SDAZ developed based on state-space modelling more intuitive and easier to use than tilt-controlled version. They commented that if they were involved with other tasks, i.e., answering the phone, showing a figure on the PDA to someone, and so forth they would prefer the touch-screen-controlled SDAZ because they imagined it would be difficult to stay in the desired position in the document with a tilt-controlled SDAZ. This is one of the significant disadvantages of using motion as an input in a handheld device, while the tilt-controlled application is always on. This problem has also been reported by Hinckley et al. (2005) and Harrison et al. (1998). Furthermore, the flying object model provided a clearer understanding about the tilt-controlled SDAZ. One of users commented that '[..] The dynamical model made sense in the interaction and I felt like I am flying above the document. Scrolling and landing on targets were quite smooth and fluid.' Moreover, they liked interacting with the device using tilt input and took full advantage of tilt in the interaction with the zooming and scrolling interface on the PDA.

7. Conclusions and summary

This article presented an example of designing a ZUI by reproducing real-world dynamic effects, including elements such as the mass, spring and damper using a dynamic systems approach. The applicability of state-space modelling was demonstrated by implementing the SDAZ interface for a text browsing system on a PDA instrumented with a tilt sensor, and a stylus. The flying object model of SDAZ was appealing and fluid to users. Also, this model

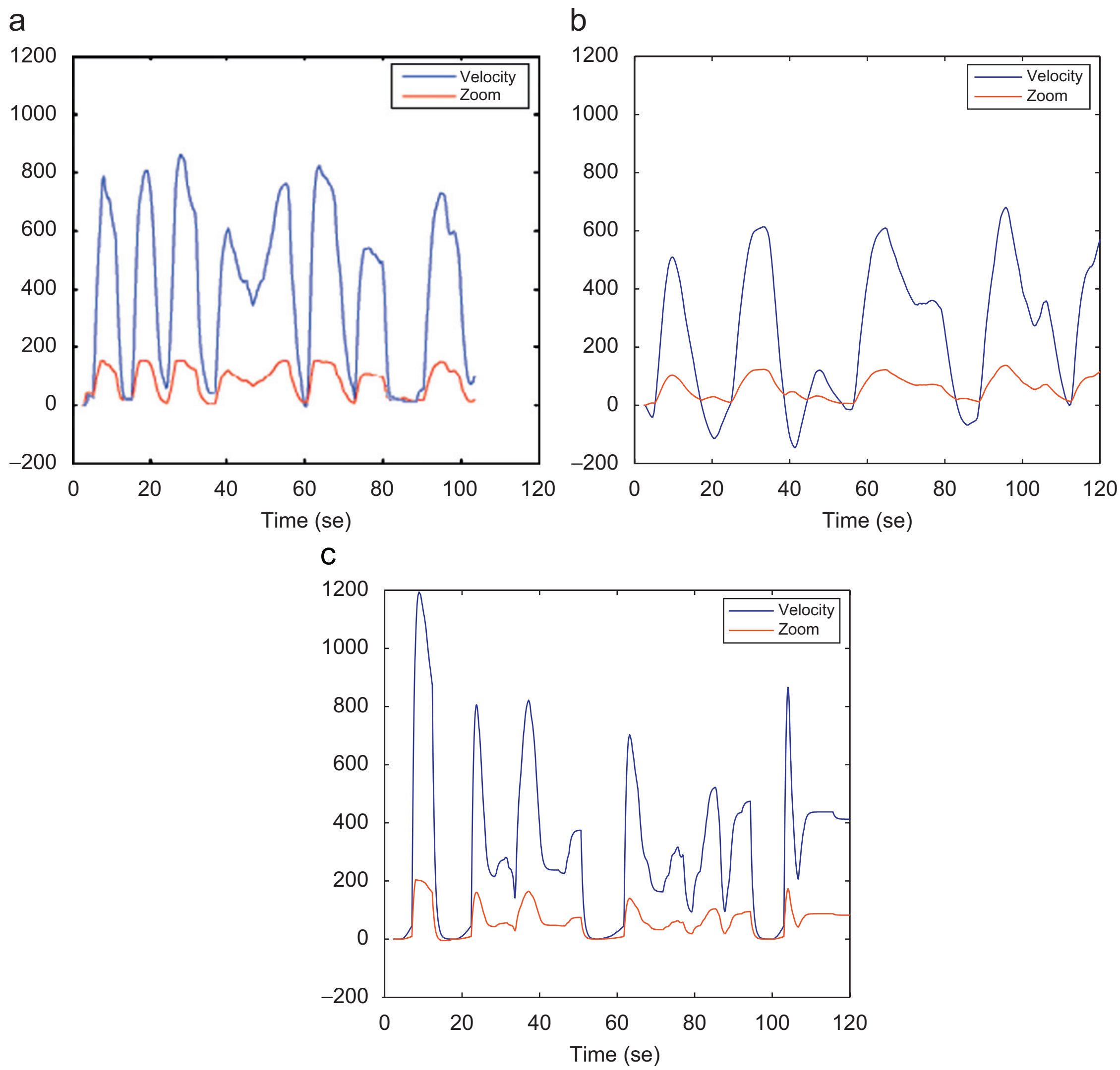

Fig. 17. (a) Tilt-controlled SDAZ with augmented velocity control. (b) Tilt-controlled SDAZ without augmented control. Hunting behaviour appears in tilt-controlled SDAZ without augmented control mode around time 20 s, 40 s, and 85 s. (c) Touch-screen-controlled SDAZ.

allowed multiple variables and derivative effects (e.g., position, velocity and acceleration) to be coupled with zoom level, without any further coding, by just changing the entries of the A and B matrices. This unified zooming and scrolling in long documents reduces the frustration caused by constantly switching between zooming and panning. Additionally, it was shown that state-space modelling makes tuning and calibration easier even with higher DOF inputs because proper settings can only be found by observing the behaviour of the simulated system before the actual implementation. In the example document browser application, it was shown that the controller can change the interpretation of inputs to being different reference values. The user controls v_{ref} or x_{ref} as a desired velocity or position manually and the controller maintains this velocity or position automatically and completes the task for the user. Also, the controller switches among different control modes based on the user input and controlled variables, however, the mode transition is smooth and natural, and the user is often not even aware that their movements are having a different effect in different modes. Moreover, augmented control made the application easier to control during landing and improved hunting behaviour.

The tilt input allowed users to control the device in a single-handed manner, without obscuring the screen. However, users preferred the stylus, as they could stop interacting by taking the stylus off the screen. Users commented that if they were involved in other tasks, e.g., answering the phone, they would prefer the touch-screen-controlled SDAZ because it would be difficult to stay in the desired position in the document, with a tilt-controlled SDAZ.

8. Future work

In mobile situations, where the user is focusing on tasks happening in the environment rather than tasks on the small screen, the touch-screen-controlled SDAZ is more popular than tilt-controlled. Because the user can take the stylus off the screen whenever she wants but in tilt input there is a continuous flow of tilt data and it is difficult to remain steady without looking at the screen. Some researches have suggested potential solutions to switch on/off the tilt input, for instance, squeezing the device (Harrison et al., 1998) or sensor data fusion and listening detection (Hinckley et al., 2005). However, these techniques suffer from high error rates in the recognition algorithm. Using motion as an input in handheld devices is challenging and without solving this, tilt input is likely to be of little widespread interest. We intend to continue this research agenda in the future, investigating controllers that respond only to *intended* tilt inputs and controllers that sense different contexts and locations the user is in and toggles on suitable inputs for the user.

This work focused more on the tilt dynamics in zooming and scrolling techniques, and that the analysis in the paper is also relevant to other sensors, where the declutching issue might be less relevant (e.g., pressure sensors as we saw in touch-screen-controlled SDAZ). Additionally, a more thorough exploitation of dynamic systems approach presents a promising theoretical framework for designing interaction models and creating better interactive systems on portable computing devices. It opens up the dynamics of the 'look and feel' on mobile applications and allows the incorporation of analytical tools and constructive techniques from manual and automatic control theory into the interface. Thus, this concrete theory provides a wide range of possibilities for future work, for instance, integrating multi-modality into the interface and modelling a human–operator in the interaction.

Acknowledgements

The author gratefully acknowledge the support of IRCSET BRG SC/2003/271 Continuous Gestural Interaction with Mobile devices, 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.

References

Ahlberg, C., Shneiderman, B., 1994. The alphaslider: a compact and rapid selector. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'94. ACM Press, Boston, MA, USA, pp. 365–371.

Aponso, B.L., Mitchell, D.G., Hoh, R.H., 1990. Simulation investigation of the effects of helicopter hovering dynamics on pilot performance. Journal of Guidance, Control and Dynamics 13 (1), 8–15.

Bartlett, J.F., 2000. Rock 'n' scroll is here to stay. In: IEEE Computer Graphics and Applications, vol. May/June.

Barrett, R.C., Sleker, E.J., Rutledge, J.D., Olyha, R.S., 1995. The negative inertia: a dynamic pointing function. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'95, Conference Companion. ACM Press, Denver, CO, USA, pp. 316–317.

Beard, D., Walker, J., 1990. Navigational techniques to improve the display of large two-dimensional spaces. In: Behaviour Information and Technology, vol. 9, pp. 451–466.

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.

Bederson, B., Hollan, J., Perlin, K., Meyer, J., Bacon, D., Furnas, G., 1994. Pad++: a zoomable graphical sketchpad for exploring alternate interface physics. In: UIST'94: Proceedings of the 7th annual ACM Symposium on User Interface Software and Technology. ACM Press, Marina del Rey, CA, USA, pp. 17–26.

Bradley, R., 1996. The flying brick exposed: non-linear control of a basic helicopter model. Technical Report, Department of Mathematics, Glasgow Caledonian University, September.

Brogan, W.L., 1991. Modern Control Theory. Prentice-Hall, Englewood Cliffs, NJ.

Büring, T., 2007. Zoomable user interfaces on small screens presentation and interaction design for pen-operated mobile devices, Ph.D. thesis, University of Konstanz, Germany.

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.

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.

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.

Doherty, G., Massink, M., 1999. Continuous interaction and human control. In: Alty, J. (Ed.), Proceedings of the 16th European Annual Conference on Human Decision Making and Manual Control. Group-D Publications, Loughborough, pp. 80–96.

Eslambolchilar, P., 2006. Making sense of interaction using a model-based approach. Ph.D. Thesis, Hamilton Institute, National University of Ireland, NUIM, Maynooth, Ireland, October.

Eslambolchilar, P., Murray-Smith, R., 2004. Tilt-based automatic zooming and scaling in mobile devices—a state-space implementation. In: Brewster, S.A., Dunlop, M.D. (Eds.), Mobile Human–Computer Interaction-MobileHCI 2004: Sixth International Symposium, Lecture Notes in Computing Science, vol. 3160. Springer, Glasgow, Scotland, pp. 120–131.

Faconti, G., Massink, M., 2001. Continuous interaction with computers: issues and requirements. In: Stephanidis, C. (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.

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.

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.

Furnas, G., Bederson, B., 1995. Space-scale diagrams: understanding multiscale interfaces. In: Proceedings of the ACM Conference on Human Factors in Computing Systems CHI'95. ACM Press, Denver, Colorado, USA, pp. 234–241.

Gibson, J.J., 1979. The Ecological Approach to Visual Perception. Houghton Mifflin, Boston.

Harrison, B., Fishkin, K.P., Gujar, A., Mochon, C., Want, R., 1998. Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'98. ACM Press/Addison-Wesley Publishing Co., Los Angeles, CA, USA, pp. 17–24.

Hess, R.A., Chan, K.K., 1988. Preview control pilot model for near-earth manoeuvring helicopter flight. Journal of Guidance Control and Dynamics 11, 146–152.

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.

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.

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.

Jagacinski, R.J., Flach, J.M., 2003. Control Theory for Humans: Quantitative Approaches to Modelling Performance. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Jones, S., Jones, M., Marsden, G., Patel, D., Cockburn, A., 2005. An evaluation of integrated zooming and scrolling on small-screens. In: International Journal of Human–Computer Studies, vol. 63. ACM Press, pp. 271–303.

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. ACM Press, NY, pp. 201–210.

Liddle, D., 1996. Design of the conceptual model. In: Winograd, T. (Ed.), Bringing Design to Software. Addison-Wesley, MA.

Masui, T., 1998. LensBar—visualization for browsing and filtering large lists of data. In: Proceedings of the International Conference in Information Visualization, InfoVis'98, pp. 113–120.

Masui, T., Kashiwagi, K., Borden, G.R., 1995. Elastic graphical interfaces for precise data manipulation. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'95. ACM Press, Denver, CO, USA, pp. 143–144.

Narendara, K., Balakrishnan, J., 1997. Adaptive control using multiple models. In: IEEE Transaction on Automatic Control, vol. 42, pp. 171–187.

Narendara, K.S., Balakrishnan, J., Ciliz, M.K., 1995. Adaptation and learning using multiple models, switching, and tuning. In: IEEE Transaction on Control Systems, pp. 37–51.

Norman, D.A., 1999. Affordances, conventions and design. In: Interactions, vol. 6. ACM Press, pp. 38–43.

Ogata, K., 1990. Modern Control Engineering. Prentice-Hall, Englewood Cliffs, NJ.

Ogata, K., 1995. Discrete-Time Control Systems, second ed. Prentice-Hall, Englewood Cliffs, NJ.

Padfield, G.D., 1996. Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling. American Institute of Aeronautics.

Patel, D., Marsden, G., Jones, S., Jones, M., 2004. An evaluation of techniques for browsing photograph collections on small displays. In: Mobile Human–Computer Interaction-Mobile HCI 2004, Lecture Notes in Computer Science. Springer, Glasgow, Scotland, pp. 271–303.

Patel, D., Marsden, G., Jones, M., Jones, S., 2006. Improving photo searching interfaces for small-screen mobile computers. In: Proceeding of the Eighth International Conference on Mobile Human–Computer Interaction, Mobile HCI'06. ACM, Espoo, Finland, pp. 149–156.

Perlin, K., Fox, D., 1993. Pad: an alternative approach to the computer interface. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Technology. ACM Press, pp. 57–64.

Poulton, E.C., 1974. Tracking Skill and Manual Control. Academic press, New York.

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

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

Preece, J., Rogers, Y., Sharp, H., 2002. Interaction Design: Beyond Human Computer Interaction. Wiley, New York.

Rekimoto, J., 1996. Tilting operations for small screen interfaces. In: UIST'96: Proceedings of the Ninth Annual ACM Symposium on User interface software and technology. ACM Press, Seattle, WA, USA, pp. 167–168.

Savage, J., 2004. The calibration and evaluation of speed-dependent automatic zooming interfaces, Master's thesis, Computer Science Department, University of Canterbury, New Zealand.

Schmidt, R.A., Lee, T., 2005. Motor Control and Learning—A Behaviourial Emphasis, Human Kinetics, fourth ed. 2005.

Thimbleby, H., 1990. User Interface Design. ACM Press.

Tischler, M.B., 1994. Advances in Aircraft flight Control. Taylor & Francis.

Wallace, A., 2003. The calibration and optimization of speed-dependent automatic zooming, Honours Report, University of Canterbury, Christchurch, New Zealand, November.

Wallace, A., Savage, J., Cockburn, A., 2004. Rapid visual flow: how fast is too fast? In: Proceedings of the Fifth Australasian User Interface Conference (AUIC2004). Australian Computer Society Inc., Darlinghurst, Australia, Dunedin, New Zealand, pp. 117–122.

Wijk, J.J.V., Nuij, W., 2003. Smooth and efficient zooming and panning. In: Munzner, T., North S., (Eds.), Proceedings of IEEE Symposium on Information Visualization (InfoVis'2003), Seattle, WA, USA, 2003, pp. 15–23.

Wobbrock, J.O., Forlizzi, J., Hudson, S.E., Myers, B.A., 2002. Webthumb: interaction techniques for small-screen browsers. In: Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'02). ACM Press, NY, pp. 205–208.

Zhai, S., Smith, B.A., Selker, T., 1997a. Improving browsing performance: a study of four input devices for scrolling and pointing tasks. In: Proceedings of INTERACT'97, pp. 286–292.

Zhai, S., Smith, B.A., Selker, T., 1997b. Dual stream input for pointing and scrolling. In: Extended Abstracts of the ACM Conference on Human Factors in Computing Systems CHI'97. ACM Press, Atlanta, Georgia, pp. 305–306.