CONTENTS

# 1

# CONTROL THEORY, DYNAMICS AND CONTINUOUS INTERACTION

**Roderick Murray-Smith**

**Abstract**

This chapter reviews the role of control theory and dynamic systems theory in understanding common interaction techniques including: targetting, trajectory generation, panning, scrolling and zooming. It explains how control can be seen to be at the foundations of Human–Computer Interaction and might be essential for making progress in novel forms of interface. It reinterprets Fitts' classical work with control theoretic tools. It also highlights the limitations of control theory for design of human–computer control loops.

## 1.1 Introduction

What do we really mean when we talk about *Human–Computer Interaction*? It is a subject with few firm, agreed foundations. Introductory textbooks tend to use phrases like "designing spaces for human communication and interaction", or "designing interactive products to support the way people communicate and interact in their everyday lives". (Rogers, Sharp and Preece, 2011). (Hornbæk and Oulasvirta, 2017) provide a recent review of the way different HCI communities have approached this question, but only touches briefly on control approaches. Traditionally HCI research has viewed the challenge as *communication of information* between the user and computer, and has used information theory to represent the bandwidth of communication channels into and out of the computer via an interface. *"By interaction we mean any communication between a user and a computer, be it direct or indirect"* (Dix, Finlay, Abowd and Beale, 2004), but this does not provide an obvious way to measure the communication, or whether the communication makes a difference.

The reason that information theory is not sufficient to describe HCI, is that in order to communicate the simplest symbol of intent, we typically require to move our bodies in some way that can be sensed by the computer, often based on feedback while we are doing it. Our bodies move in a continuous fashion through space and time, so any communication system is going to be based on a foundation of continuous control. However, inferring the user's intent is inherently complicated by the properties of the control loops used to generate the information – intention

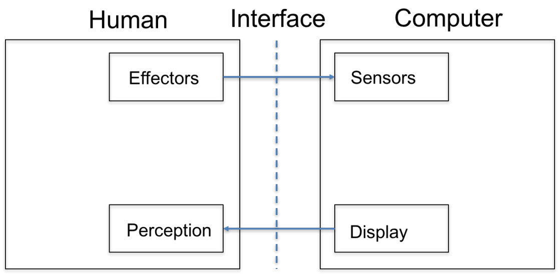


Fig. 1.1. Human–Computer Interaction as a closed-loop system

in the brain becomes intertwined with the physiology of the human body and the physical dynamics and transducing properties of the computer's input device. In a computational interaction context, the software adds a further complication to the closed-loop behaviour. (Hollnagel, 1999; Hollnagel and Woods, 2005) make a compelling argument that we need to focus on how the *joint* human–computer system performs, not on the communication between the parts.

Another reason that control theory can be seen as a more general framework is that often the purpose of communicating information to a computer is to control some aspect of the world, whether this be the temperature in a room or the volume of a music player,[1] the destination of an autonomous vehicle or some remote computational system. This can be seen in Fig 1.2, which illustrates the evolution of human–machine symbiosis from direct action with our limbs, via tools and powered control. Over time this has led to an increasing indirectness of the relationship between the human and the controlled variable, with a decrease in required muscular strength and an increasing role for sensing and thought (Kelley, 1968). The coming era of *Computational Interaction* will further augment or replace elements of the perceptual, cognitive and actuation processes in the human with artificial computation, so we can now adapt the original figure from (Kelley, 1968) to include control of *computationally enhanced systems*, where significant elements of:

1. the information fed back to the human,
2. the coordination of control actions and
3. the proposals for new sub-goals

are augmented computationally. This computational augmentation is intended to achieve a similar decrease in the complexity of human cognition, perception and actuation that earlier generations achieved over muscle strength. In some cases this will decrease the human interaction with some tasks, in order to be able to apply more attention and cognitive resources to other, currently more important, aspects of their environment. For example, computationally enhancing the music player in a car

[1] One might think the reference here is the loudness of the music, but in many social contexts it is probably the inferred happiness of the people in the room that is actually being controlled, and any feedback from volume indicators are just intermediate variables to help the user.

allows the driver to have an acceptable degree of control over the style of music played while, more importantly, being able to concentrate on driving safely. A further step, completely automating driving itself would allow the human to shift their resources to focus on engaging with family members or preparing for an upcoming business meeting.

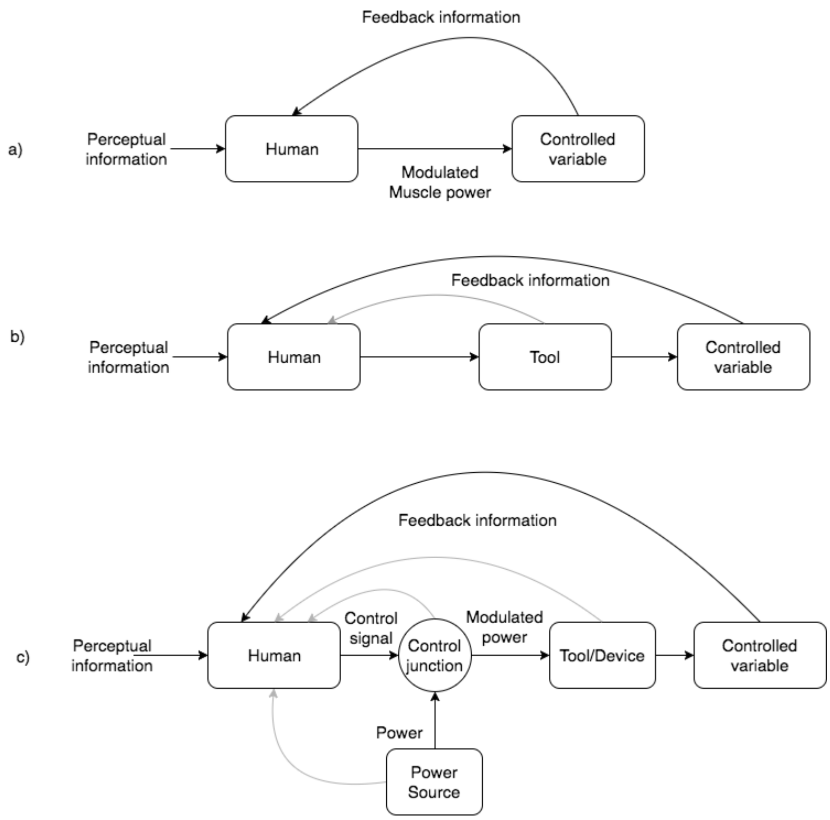


FIG. 1.2. The evolution of human control over the environment from (a) direct muscle power via (b) use of specialised tools for specific tasks and (c) externally powered devices, potentially regulated by automatic controllers. Adapted from (Kelley, 1968). Grey lines indicate optional feedback connections.

### 1.1.1 *Related work*

Few modern researchers or practitioners in HCI have received training in *control theory*, which has been an interdisciplinary branch of engineering and mathematics for 70 years. It deals with the behaviour of dynamic systems with inputs, and how their behaviour is modified by feedback.

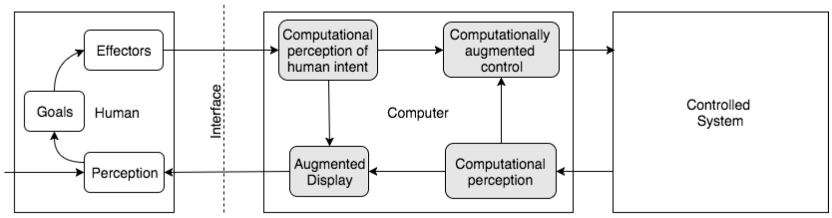


FIG. 1.3. The next step in evolution – control with power- and computationally-enhanced devices. Grey blocks indicate computational intelligence enhancement.

The specific area of control systems relating to human users became a major area of activity from the 1950's. This chapter aims to introduce human–computer interaction researchers with a computing science background to the basics concepts of control theory and describe a control perspective of some common interaction models.

*Manual control theory* (McRuer and Jex, 1967; Costello, 1968) seeks to model the interaction of humans with machines, for example aircraft pilots, or car drivers. This grew from Craik's early, war-related work (Craik, 1947; Craik, 1948), and became more well-known in the broader framing of Wiener's *Cybernetics* (Wiener, 1948). As observed by (Wickens and Hollands, 1999), the approach to modelling human control behaviour came from two major schools, the skills researchers and the dynamic systems researchers. The 'skills' group often focused on undisturbed environments, while the 'dynamics', or 'manual control theory' approach (e.g. (Kelley, 1968; Sheridan and Ferrell, 1974)) tended to seek to model the interaction of humans with machines, for example aircraft pilots, or car drivers, usually driven by engineering motivations and the need to eliminate error, making it a closed-loop system. The 'skills' group tended to focus on learning and acquisition while the 'dynamics' group focused on the behaviour of a well-trained operator controlling dynamic systems to make them conform with certain space-time trajectories in the face of environmental uncertainty. This covers most forms of vehicle control, or the control of complex industrial processes. (Poulton, 1974) reviews the early tracking literature, and an accessible textbook review of the basic approaches to manual control can be found in (Jagacinski and Flach, 2003). Many of the earlier models described above were based on frequency domain approaches, where the human and controlled system were represented by Laplace transforms representing their input/output *transfer function*. Optimal control theoretic approaches used in the time domain are described in (Kleinman, Baron and Levison, 1971; Kleinman, Baron and Levison, 1970). The well-established field of *human motor control theory*, e.g. (Schmidt and Lee, 2005), which seeks to understand how the human central nervous system controls the body, is an important component of using control theory in HCI, but this chapter will focus on the basic role of control concepts in HCI.

## 1.2 The control loop

### 1.2.1 *The classical control loop*

Figure 1.4 shows a representation of the classical control loop. It is a general representation of many possible control or tracking tasks, covering car driving, mouse movement, or the control of a chemical process. It represents a dynamic system, so the output of each block is time-varying.

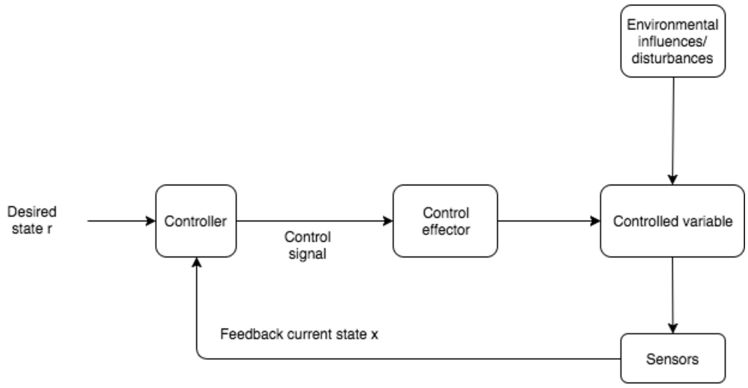


FIG. 1.4. Classical control loop

The *Goal* is also called the *reference*. It describes the desired *state* of the controlled system. The state is a real vector which describes the condition of the controlled system, a position in the *state space*. The concepts of state variable, state and state space are important tools which can help designers understand the problem. The *choice of state variables* for your model is a statement about the important elements of the system. The state dimensions combine to form the *state space*. Behaviour is visualised as movement through this space, and the values of state reflect the position compared to important landmarks such as the goal.[2] Problem constraints can be represented as boundaries in the state space and qualitative properties of the system can be described as regions in the state space (Bennett and Flach, 2011).

*Inputs and Outputs:* The controller generates control inputs to the controlled system. Transformations of the system state are observed, the *outputs*. The concepts of *controllability* and *observability* are important, as they describe the degree to which a control system can observe and control the states of a particular system.

*Open/Closed loop:* If an input is transformed in various ways but the control variable does not depend on feedback from the system state, the system is described as *open loop*. *Closed loop* systems have feedback from the state to the controller which affect the input to the controlled system.

[2]Note that for dynamic systems *position* in a state space can describe a rapidly changing situation.

*Disturbances* are external or unpredictable effects which either directly affect the system state or the input to the controlled system. In an open-loop system, the controller is unable to compensate for these, while a closed-loop controller can observe the error and change its control variable to compensate. In HCI, disturbances or noise on the input channel from user to computer are a key reason for feedback. These might be due to unintended motion, sensor noise or incompleteness or discretisation thresholds in the interface software.

*Stability:* In technical control systems, stability is a key aspect of design. This has been less of an issue in modern HCI, but is important in automotive and aircraft control, where interactions between the human operator and technical control systems can lead to instability, and 'pilot-induced oscillations'. Stability is relevant not only for equilibria, but also for period motions. For example, a pendulum is typically stable to minor perturbations around its normal limit cycle. Time delays are a prime cause of temporary instability in conventional user interfaces.

*Feedback:* The display is to provide the user with information needed to exercise control; i.e. predict consequences of control alternatives, evaluate status and plan control actions, or better understand consequences of recent actions. We can augment displays or controls. If we augment the display, we improve the input to the human to simplify their control task. If we augment the control, we change the effective dynamics between control input and system output. E.g. most mouse drivers apply nonlinear filters to the raw data from the mouse sensors.

*'Dynamics':* how a system responds over time. Investigation of the behaviour of a controlled system requires us to observe its change of state, and in physical systems this requires transfers of energy or mass. An instantaneous change of state in such a system would require an infinitely large flow of energy or mass, but in real systems we have a transition which takes place over time, and we call such systems *dynamic systems.*

In human–computer interaction, because the human effectors (e.g. an arm in pointing) have mass, and systems with mass cannot instantaneously achieve high velocity, the rate at which velocity or position builds up depends on the force applied to the limb, and its mass, resulting in an acceleration $a = \frac{F}{m}$ which then has to integrate up over time to have an impact on measured velocities or positions.

Information states in a computer can change instantaneously, from a human perspective. However, it is not only the physical limitations of our effectors that matter. Powers builds on Gibson's work in (Powers, 1973; Powers, 1989; Powers, 1992) to highlight that humans (and other animals) evolved to control their perceptions, to generate their behaviour. The human and the computer create a coupled dynamic system. As the human has perceptual bandwidth limitations, he or she requires time to process their sensory feedback and to act on it (again subject to bandwidth limitations), a well-designed system should not make changes in state at a rate faster than the human can follow – otherwise they cannot control the system.

In the case of a human controlled system the control block might be split between a 'human operator' block and a 'control' block where elements of the technical system which have a control function are conceptually separated from the process being controlled. See Fig. 1.5 In the case of computer user interfaces this may represent different layers of software.

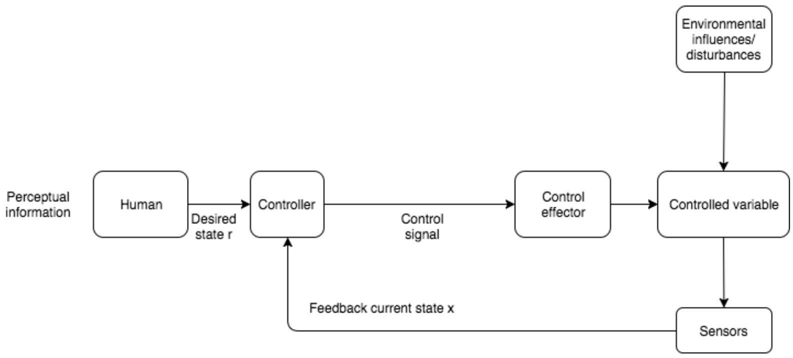


FIG. 1.5. Human–Computer control loop where the 'control' element is partitioned into human-controlled and system controlled elements. The human above is providing a reference for the automatic controller on the basis of broad perceptual information about the context.

### 1.3 Classes of continuous interaction and examples

Modern Graphical User Interfaces on personal computers controlled by mice are primarily spatial in nature, using targetting of position over stationary regions for communication of intent. The widespread use of touchscreen smartphones has increased the use of dynamic gestures where a variable is measured over *time*, such as swipes, flicks and taps, but the methods used have not tended to scale well to more complex cases with richer sensor systems, where a lot of interaction potential is disregarded. Being able to use dynamic state changes to control interaction and infer context could open up new styles of interaction, but our current analytic tools, such as Fitts' law, are not sufficient. If we can identify a continuous signal loop between the user and interface, such that the user is, at times, continuously minimising the error between his or her intention and the system state, we can say the user is interacting in a continuous fashion with the system. We say 'at times', because there will usually be a de-clutching mechanism, such as letting go of the mouse, such that the interaction is *intermittently continuous.*

The fields of physiology, cognitive science and ergonomics provide us with models of human aspects of interaction, e.g. low-level perception, motor control, short-term memory and attention. Computing science is full of models of aspects of machine behaviour, e.g. inference algorithms, FSMs, Statecharts etc., but there is less theoretical support in the area of *interaction*. Manual control theory provided the first steps, with aircraft and automobile control, and models were often implemented on analogue computers. At that time digital computers involved essentially discrete interaction, but now that improvements in sensing, simulation tools, speed and memory mean that important elements of the computer side have become essentially continuous again, we need to look at ways of analysing the joint system.

### 1.3.1 *Hitting a fixed spatial target with a pointer*

The graphical user interface involves spatial representations of actions in graphical user interfaces. In a simple 1-from-$N$ case, the $N$ possible actions are laid out as $N$ shapes in two dimensions, and the user has continuous control of a cursor via e.g. a mouse. The control task for the user is then to recognise the target (often via a visual prompt such as an icon) and to move the cursor towards it, clicking once within the boundary of the target shape. If the chosen target has a position $(x_r, y_r)$, and the current cursor state is $(x, y)$ then there is an 'error'

$$e = \left\| \begin{bmatrix} x_r \\ y_r \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \right\|,$$

which describes the distance between cursor and target, and the control task is to minimise this error as fast as the human effector dynamics allow, bringing the cursor $x, y$ towards $(x_r, y_r)$. In control theory, we describe this as a *step response* because the change of target position looks like a step in the time-series representation. The process of spatial targetting was examined in detail from the control perspective, for one-dimensional pointing in (Müller, Oulasvirta and Murray-Smith, 2017).

One important issue is that for most interaction tasks you are moving a cursor towards an *area*, not a *point*, so the conditions for selection are met before the error is brought to zero. As the size of the area increases relative to the distance travelled (a decreasing *index of difficulty* in Fitts' terminology), this becomes more significant. (Müller, Oulasvirta and Murray-Smith, 2017) found that users tended not to minimise the error in the final endpoint and used different control behaviours when faced with larger targets. We will discuss this again in Section 1.6.1.

#### 1.3.1.1 *Control order*

The control action $u$ for mouse input typically measures the velocity of the mouse over a surface, and feeds that through to the cursor via a nonlinear *transfer function*.[3] (Casiez and Roussel, 2011) develop a framework for comparing pointing transfer functions. However, the sensed input from the human could also be position (as in touch screens) or acceleration from accelerometers such as those on a smart watch or mobile phone. The *control order* refers to the number of integrations between control input to a plant and output of a plant. Higher order systems are harder to control. Zero order control is position

[3] Note that in control theory the term *transfer function* tends to refer to a linear time-invariant (LTI) system in Laplace or Fourier transform representation.

control, and the gain level will affect accuracy and speed in target space. First order control, velocity control, has one integration between position and velocity, and works well for systems with a well-defined null or zero position (like a spring-loaded joystick). The main advantage is that the range of motion in the space is not limited to the range of moment in input space. The limits on input constrain the velocity, not the range of space. With second order, acceleration control, a return to the null position zeros the acceleration but not the velocity – you need to counteract the velocity by decelerating, so it is more difficult than $1^{\text{st}}$ or $2^{\text{nd}}$ order control, but reflects real world activity. Higher order control systems are much more difficult to learn (pilots need to deal with $3^{\text{rd}}$ and $4^{\text{th}}$ order elements in fixed wing flight). If making comparisons between different input devices, it is important to compare the same dynamics, e.g. a comparison of a position mouse input with a velocity joystick input might lead to misleading conclusions.

An interesting trend in the evolution of interfaces has been to reduce the control order, moving from joysticks to mice to direct touch. Direct interaction can be cognitively simpler, but also has disadvantages associated with the size of workspace. $0^{\text{th}}$ order, position input makes the input space the same size as the display, whereas higher order inputs can change the gain to trade-off speed of movement with end target precision. Direct interaction usually means that an extra navigation layer which allows the user to move between smaller canvases needs to be added to the interaction design. A direct mapping also makes it harder to 'slide in intelligence' to flexibly modulate the user input with computational models of anticipated behaviour, whereas this is easier when the input evidence integrates up over time.

### 1.3.2 *Variability*

Sources of variability in human action include noise, trajectory planning and delays in the feedback loop. In some tasks motor planning can be complex, but with most interfaces being designed to have simple dynamics, and human limbs being typically well controlled over the range of motions used, most UI interaction is still relatively simple from a motor control perspective. The variability is therefore dominated by the human's predictive control interacting with delayed or incomplete feedback, and the variability in defining the timing of changes of motion. If the user moves extremely slowly then the effects of human lags and delays are negligible.

Will control models do a better job of representing user variability? Most of the historical applications of manual control models did not focus on variability, but the intermittent predictive control models have done a better job of this. (Gawthrop, Lakie and Loram, 2008) demonstrate that a simple predictive controller can be consistent with Fitts' law, while non-predictive controllers cannot. The same paper also presents the link between *intermittent control*, predictive control and human motor control, and further develops this in (Gawthrop, Loram, Lakie and Gollee, 2011; Gawthrop, Gollee and Loram, 2015).

We can see the interaction of difficulty of the targeting task with the speed and variability of human movement, even in simple one-dimensional spatial targeting tasks (Müller, Oulasvirta and Murray-Smith, 2017). These experiments make clear the anticipatory, or predictive element in human action, as the acceleration behaviour for different levels of difficulty changes well before the user reaches the target.

(Quinn and Zhai, 2016) describe the challenges associated with understanding user behaviour when performing gestural input, focusing on gesture keyboards. They highlight how users are not tightly constrained, but are attempting to reproduce a prototype shape as their goal. Earlier models assumed that the motor control process was similar to serial aiming through a series of landmarks. They used a minimum-jerk model of motor control to model human behaviour and describe the variability in user control in gesture keyboards, which includes some general issues, including: an inclination to biomechanical fluidity, speed–accuracy trade-offs, and awareness of error tolerances in the algorithms, visual online feedback from the interface, and the errors due to sensorimotor noise and mental and cognitive errors.

### 1.3.3 *Tracking a moving target with a pointer*

A generalisation of hitting a stationary spatial target is to allow the target to move, forcing the user to track the target (Crossman, 1960; Poulton, 1974). The control models which are optimised for this task will have different parameters, structures and qualitative properties from those suited to the static case in Section 1.3.1. The quality of the control will depend on the frequency range of the target movements and the delay and lag inherent in the user response. The most common application of this is in gaming. It may have further application in detecting attention and for selection in virtual or augmented reality situations.

An important element of the research in target pursuit is the amount of trajectory preview the human user has, as this changes the amount of information that can be communicated (Crossman, 1960) in part by making it easier for the user to use prediction to plan the exact timing of their actions, and avoid the impact of delays.

### 1.3.4 *Driving a pointer through a spatial continuum of constraints*

In hitting a spatial target, it did not matter how you got to the target, the system just has a simple mapping of click location to actions. In other cases, the selection is dependent on the user generating a trajectory that fits particular spatial constraints. This is sometimes described as a 'tunnel' task. The *Steering law* proposed in (Accot and Zhai, 1997; Accot and Zhai, 2002*a*) is an example of a generalisation of Fitts' results to trajectory tasks. In this case we have a reference trajectory, $C_r(s)$, which describes a series of $(x_r, y_r)$ values at arc length $s$. This can be accompanied by a varying constraint, or width $W(s)$.[4]

[4]The analysis of boundary crossings in (Accot and Zhai, 2002*b*) is closely related, especially if there is a sequence of multiple crossings needed to achieve a specific goal. This can be seen as a discretisation of the steering law task, with intermittent constraints.

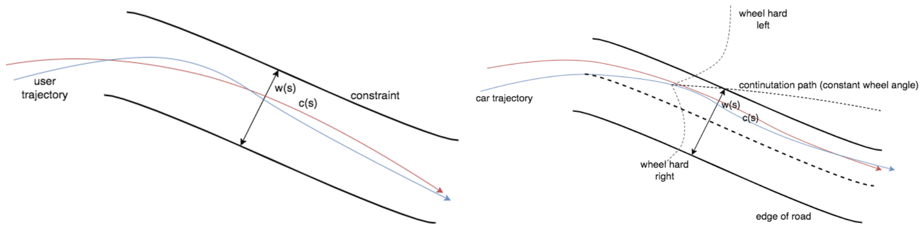


FIG. 1.6. Two-dimensional ‘tunnel’ task (left). A classical automotive steering task (right)

Most of the existing research has avoided the use of dynamic control models, preferring to generalise the task as an infinite series of spatial constraints (although the steering task was a staple of early work, e.g. Car steering (Rashevsky, 1959; Rashevsky, 1960), or pencil line-drawing tasks by (Drury, 1971)). In recent years there has been increasing awareness of the mismatch between human behaviour and the simple early steering models. (Pastel, 2006) examines performance in steering tasks with sharp turns, and (Yamanaka and Miyashita, 2016*a*; Yamanaka and Miyashita, 2016*b*) examine the impact of narrowing or widening tunnels on user performance, despite these notionally having the same index of difficulty.

This can be reformulated as a control task, where an agent has to progress along the tunnel within the constraints of the tunnel ‘walls’. If we factor in typical human delays and lags in a tracking model such as the McRuer model (Jagacinski and Flach, 2003), then with standard frequency response analysis we can analytically show that sharp turns in the tunnel will lead to larger deviations from the central path, and will require a reduction in speed, compared to a lower curvature trajectory, if the user is to stay within the tunnel constraints. This also highlights the important element of user ‘look ahead’ where a predictive or feed-forward element comes into play. Users can see the trajectory ahead, with its constraints, and adjust their velocity and plan their directional control behaviour accordingly. The impact of a given amount of timing uncertainty is greater for larger curvature in the tunnel, or tighter width constraints. The initial work (Accot and Zhai, 1997) did not incorporate curvature into the ID, but did hypothesize a likely relationship of the tangential velocity $v(s) \propto \rho(s)W(s)$, where $\rho(s)$ is the local radius of curvature, in line with the results of (Viviani and Terzuolo, 1982). The goal here is to create a model of the probability density function from the current state $(x, v)$ at $t_n$ for the user's future behaviour $t_{n+1}, t_{n+2}...$ etc. Because of motor variability, this will naturally spread out spatially over time in areas of high curvature, returning to close to the optimal path in easier areas. The nature of the trade-off between spread of trajectories and variation in speed will depend on the implicit cost function the user is performing to. Are they being cautious and never breaching the constraints, or more risk-taking and increasing speed?

Appropriately identified models of human tunnel following behaviour would allow us to create a more appropriate function for inferring intent than simply detecting that a user had excited the tunnel. Predicting the likely dynamic behaviour for a user attempting the tunnel at a given speed could allow us to infer which of $N$ targets was most likely. An interesting extension of this is instead of speed-accuracy trade-offs, we could look at effort-accuracy trade-offs, where users might choose to provide less precision in the location or timing of their actions. (The techniques can link closely to methods used for filtering noisy inputs).

Feedback during the process can change closed-loop performance significantly. For example, when examining touch trajectories associated with the 'slide to open' on an iPhone we could view it as a response to a spatially constrained trajectory following task, but because of the visual metaphors the user typically perceives it as dragging the slider (which in itself is a direct position control task), to activate the device. The physical metaphor might make sense for a new user, but as the user becomes more skilled and confident, they may be keen to have faster or more sloppy movements to achieve the same end. For example, the actual constraints on the touch input need not correspond to the limit of the drawn object in order to communicate the desired intent, depending on the context and design trade-offs. For example, if the system sensed that the user was walking while using the slide–to–open feature, it could be more forgiving on the constraints than in a stationary context, or a confident, cleanly contacting, fast swipe might be allowed to unlock the device even if it were at the wrong angle.

1.3.4.1 *Phase space tunnels* An interesting generalisation of the spatial tunnel is to define tunnels in *phase space* which include spatial coordinates and their time derivatives (e.g. $(z, \dot{z})$). This allows us to define not only a spatial trajectory but also the way that unfolds over time. An example of data collected from a finger moving above a capacitive touch screen sensitive to 5cm above the device is shown in Figure 1.7.

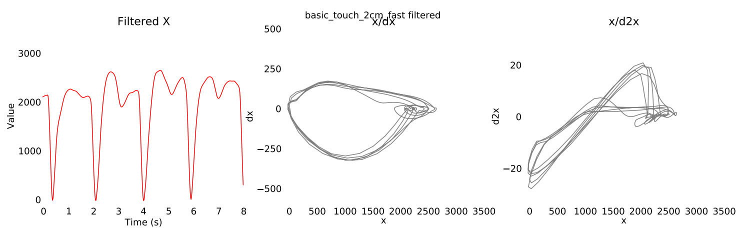


Fig. 1.7. Examples of phase space plots in velocity and acceleration against position for a finger moving above a touch screen. Note how the particular style of movement corresponds to a constrained region of the phase space.

### 1.3.5 *Gestures*

Gestures can be viewed as generating a spatial trajectory through time, and can therefore be represented by differential equations, and tolerances around a prototypical gesture. For a review of gestures in interaction, see (Zhai, Kristensson, Appert, Anderson and Cao, 2012). The underlying control task then, is almost identical to that of Section 1.3.4, although typically in gestures there is no visual guide for the user to follow – they are expected to have memorised the gestures, and implicit constraints, such that they can be generated in an open–loop fashion. In some cases, a rough reference framework is provided for the gesture. For example, the Android gesture lock screen provides a grid of points as a framework for users' gestures. This reduces the possible gesture prototypes to a discrete set, as the gesture passes through these subgoals, and helps normalise the users' spatial performance and transparently removes the variability factor of timing.

A differential equation representation of gestures is used in (Visell and Cooperstock, 2007). A specific example of differential equations for gesture is that of generating controlled cyclic behaviour as discussed in (Lantz and Murray-Smith, 2004). The advantage of rhythmic gestures is that they can be repeated until the system recognises them, whereas ballistic gestures are more frustrating to repeat from scratch if the system fails to recognise them.

Handwriting can be viewed as a specific case of gesture system, but one which leaves a visible trace, and which most humans have spent years learning. Recent developments in handwriting recognition based on the use of recurrent neural networks to learn the dynamic systems required to both generate and classify handwriting (Graves, 2013) could be generalised to other areas of gestural input. This work made progress by not requiring the training data for the handwriting task to be broken down into individual letters, but to work at a word level, letting the machine learning cope with the variability, and co-articulation effects from neighbouring letters. This might be of interest in analysis of interactions 'in the wild' where it can be difficult for a human to label when exactly a user changed their goal to a particular target or task.

### 1.3.6 *Panning, Scrolling, Zooming and Fisheye-style distortions*

When the information space a user is interacting with is too large to fit on the display, the user needs to be able to control their $(x, y)$ location in the space via panning and scrolling, and their zoom level $z$. These can be independently controlled, or can be automatically coupled to cursor movements. In many systems the continuous dynamics of transitions are not defined as differential equations, but are programmed as a series of transitory 'effects'.

In (Eslambolchilar and Murray-Smith, 2008) we created a simple 'flying brick' model which gave the panning and zooming inertia, and used state-space equations which coupled the zoom level with the velocity,

$$\dot{x}_1(t) = v(t) = x_2(t) \tag{1.1}$$

$$\dot{x}_2(t) = a(t) = \dot{v} = -\frac{R}{m}x_2(t) + \frac{1}{m}u(t) \tag{1.2}$$

$$\dot{x}_3(t) = z(t) = -\frac{b}{m}x_2(t) - \frac{R'}{m}x_3(t) + \frac{c}{m}u(t), \tag{1.3}$$

which can then be more conveniently represented in the state space form $\dot{x} = Ax + Bu$,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{R}{m} & 0 \\ 0 & -\frac{b}{m} & -\frac{R'}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ \frac{c}{m} \end{bmatrix} u. \tag{1.4}$$

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 $u$ inputs affect each state. Note that you can change from velocity to acceleration input just by changing the values of the $B$ matrix. 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.

We can create a control law such that $u = L(x_r - x)$ and write new state equations, $\dot{x} = Ax + Bu = Ax - BLx + BLr = (A - BL)x + BLr$ which shows how the control augmentation has changed the closed loop dynamics. The user input can then be linked to the $x_r$ value so that this could then be linked to a desired velocity, or a desired position in the space. It is also possible to have a switched dynamic system which changes the dynamics depending on the mode the user is in, supporting their inferred activity, and (Eslambolchilar and Murray-Smith, 2008) describes examples with different regimes for exploration, cruising and diving modes. The stability and control properties of such systems are examined in (Eslambolchilar and Murray-Smith, 2010).

This approach was further developed in (Kratz, Brodien and Rohs, 2010), where they extended the model to two-dimensions and presented a novel interface for mobile map navigation based on Semi-Automatic Zooming (SAZ). SAZ gives the user the ability to manually control the zoom level of an SDAZ interface, while retaining the automatic zooming characteristics of that interface at times when the user is not explicitly controlling the zoom level.

1.3.6.1 *Dynamic systems as canonical representations of interface dynamics* Taking a dynamic systems approach has the potential to give a cleaner underlying structure, and makes the comparison of performance between different design decisions easier to document and analyse. Research in HCI can also be slowed by the lack of transparency of many of the systems we interact with. Commercial software is often only available as a 'black box' where we can interact with it, but cannot see the underlying code. This is an area where systematic approaches to identify the dynamics of system transitions can allow us to create a canonical representation of the dynamics as a differential equation which is independent of how it was implemented. An example of this is the work on exposing scrolling transfer functions by (Quinn, Cockburn, Casiez, Roussel and Gutwin, 2012). (Quinn, Malacria and Cockburn, 2013) used robots to manipulate various touch devices to infer the scrolling dynamics. A differential equation approach provides a universal representation of the different implementations, even if they did not originally use that representation internally. This could have a role in intellectual property disputes, where more objective similarity measures could be proposed, which are independent of trivial implementation details. The differential equation approach can also be applied to other mechanisms for presenting large data spaces, e.g. fisheye lenses (Eslambolchilar and Murray-Smith, 2006).

### 1.3.7 *Homeostasis- and tracking-based interaction*

Interfaces can infer the user's intent based on detection of controlling behaviour, as developed in (Williamson and Murray-Smith, 2004; **?**) and built on by (Fekete, Elmqvist and Guiard, 2009). These models can either be set up as pursuit/tracking tasks or as homeostatic tasks, where the goal is to stabilise the system. These can be used in security-sensitive interactions to make the visibility of a user's actions irrelevant without knowing the state of the display. It can also be used with unconventional sensing configurations, and has recently been further developed as a promising approach for eye tracking and gestural interaction based on body tracking, e.g. (Clarke, Bellino, Esteves, Velloso and Gellersen, 2016) and (Velloso, Carter, Newn, Esteves, Clarke and Gellersen, 2017).

## 1.4 Fitts' Law results from a control perspective

The speed/accuracy trade-off has been a staple topic for the HCI community, with much of the attention focussed on Fitts' law (Fitts, 1954; Fitts and Peterson, 1964). Chapter 7 of (Schmidt and Lee, 2005) and (MacKenzie, 1992) provide good reviews. Fitts proposed that the time ($MT$) to move to a target area is a function of the distance to the target ($A$) and the size of the target ($W$),

$$MT = a + bID, \tag{1.5}$$

where $ID$ is the Index of Difficulty

$$ID = \log_2 \left( \frac{2A}{W} \right). \tag{1.6}$$

Movement times and error rates are important aspects of human interaction, but they do not provide a complete picture.

A feedback control based explanation was provided in 1963 by Crossman and Goodeve, reprinted in (Crossman and Goodeve, 1983), where they suggested that Fitts' Law could be derived from feedback control, rather than information theory. They proposed that there would be a ballistic, open-loop phase followed by a closed-loop homing-in phase. This is sometimes called the iterative-correction model. kinematic records of subject movements, however, tended to only have one or at most two corrections, so an alternative was proposed in Meyer's optimised-submovment model(Meyer, Smith, Kornblum, Abrams and Wright, 1990). Meyer et al.

proposed that the time ($MT$) to move to a target area is a function of the distance to the target ($A$) and the size of the target ($W$), $MT = a + bID$, where the index of difficulty, $ID = (\frac{A}{W})^{\frac{1}{n}}$, where $n$ relates to the upper limit on submovements. $n = 2.6$ minimised the RMS error. A number of authors have already related Fitts' Law to basic control models, including (Connelly, 1984; Cannon, 1994). Here, we follow the presentation in (Jagacinski and Flach, 2003) to demonstrate that Fitts' law results can be derived from first-order control behaviour. They propose that the change in position from the home position to a target be viewed as a step change in reference variable $r$. They use a simple first order controller composed of a gain $k$ and integrator. $\dot{x} = Bu$, where the control signal $u = r - x$, and $B = k$. If we imagine a step change, $r$ from initial state $x = 0$, then the response of the first order lag will be an exponential response

$$x(t) = r(1 - e^{-kt}).$$

For a target sized $w$ centered on $r$, then the time taken to get within $\frac{1}{2}w$ of $r$ is

$$\begin{aligned}
x(t) &= r - \frac{1}{2}w \\
r(1 - e^{-kt}) &= r - \frac{1}{2}w \\
e^{-kt} &= \frac{w}{2r} \\
-kt &= \ln \frac{w}{2r} \\
t &= -\frac{1}{k} \ln \frac{w}{2r}
\end{aligned}$$

which, after converting to a base 2 logarithm, via $\log_a x = \frac{\ln x}{\ln a}$, is

$$t = \frac{\ln 2}{k} \log_2 \frac{2r}{w}, \tag{1.7}$$

which is similar in form to Fitts' ID, in equation (1.6). The gain $k$ affects the speed of acquisition – the time constant for such a first order lag is $\frac{1}{k}$, the time it takes to reach 63% of the steady state response.

### 1.5 Models

Models can be used to create the controllers, or can be directly incorporated into the controller. In some cases the controller can be seen as an implicit model of the system and environment (Conant and Ross Ashby, 1970; Eykhoff, 1994). In many areas of HCI research we need to compare user behaviour to some reference behaviour. How similar are two trajectories? Use of simple Euclidean distance measures between two time series can lack robustness, especially as the dimension increases, or faced with timing variability. However, if we can identify model parameters for a specific user, we can calculate the likelihood of model parameters given the observed data, which can be more robust in some cases.

### 1.5.1 *Models of the human – human limitations*

Models of the capabilities and limitations of humans in interaction loops are well established in the research literature. From a control perspective, key elements relate to the *bandwidth* a user can control, *delays* due to cognitive processing time and neuro-physiological response times. A more challenging area for HCI researchers is that of *prediction* or *anticipation*, as it can be more difficult to measure and control for in experiments.

1.5.1.1 *Prediction/anticipation by the human* Humans can look ahead from their current state, and predict how their current state and constraints between them and their target will affect their ongoing behaviour. There will, however, be uncertainty in these predictions. The uncertainty will depend on user skill, sensing uncertainty, external disturbances and context. The prediction horizon is typically associated with the *time* needed to process the information and act such that problems at the prediction horizon are avoided, rather than being a fixed *distance* ahead, akin to the stopping distance for a car increasing with increasing speed.

Can predictive control models explain, e.g. the change in steering task performance when the constraints are widening or narrowing? A model-predictive control with state uncertainty and a prediction uncertainty increasing with the prediction horizon will typically have a distribution of future trajectories, and if a certain probability threshold of breaching the constraints is crossed, then the user needs to change behaviour, by reducing speed, or changing direction. For example, in Fig. 1.8 you can see that, in the case of a widening tunnel, the predictions are all within the tunnel, whereas a narrowing one has some breaching the constraints, forcing the user to slow down. Similarly, if the curvature of the reference trajectory increases, for a fixed tunnel width, then we would expect an impact on performance because of the impact of uncertainty in timing on control actions being greater in high curvature, narrow regions.

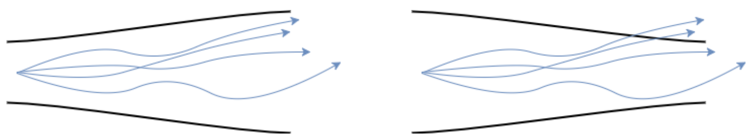

Fig. 1.8. Impact of prediction horizon and narrowing straight tunnel. The same variability that can be contained in a widening tunnel will breach the constraints of a narrowing one.

The ability to adapt can depend on whether the task is one of *forced* or *unforced* reference following. In unpaced tasks, users can increase speed and accuracy, as their preview increases. In forced pace tasks, their speed cannot change, but their accuracy improves if their preview increases (Poulton, 1974).

1.5.1.2 *Speed/accuracy/effort trade-offs* A further key area is how humans typically make trade-offs when they are asked to maximise different aspects of a cost function. *Speed*, *accuracy* and *effort* are typically mutually conflicting objectives. Azzopardi's chapter in this volume, on the use of economics models in information retrieval describes examples of users making trade-offs between effort and performance. (Guiard and Rioul, 2015) explore the tradeoffs between speed and accuracy in pointing. (Shadmehr, Huang and Ahmed, 2016; Apps, Grima, Manohar and Husain, 2015; Rigoux and Guigon, 2012) explore the role of effort in human performance. (Lank and Saund, 2005) consider sloppiness in interaction.

### 1.5.2 *Models of the computer*

The block diagram representation of control loops in engineering is intended to indicate their modularity and independence. It shows how groups of components in a feedback loop can be exchanged with others. An important assumption here is that when we change one block, other blocks remain the same. Such independence would be very valuable for system designers, as if an input device were changed from e.g. a mouse to a joystick, we could then predict the overall change to the system behaviour. With human controllers, however, this is often not the case because of the human ability to predict and adapt.

Part of the rationale for taking a control perspective is that we want to get away from the notion that behaviour is a simple response to a stimulus. In reality, the actions shape the environment, and agents often seek out stimulation. A problem with the reductionist approach is that it separates perception and action (in experiments and in theories). This often happens in HCI, where people will treat inputs and outputs separately.

Treating the system as a 'Black box' in a behaviourist manner means you just look at inputs and outputs, or you can break the process into stages – the information-processing perspective. If feedback is considered, it is often treated in a peripheral manner, and does not affect the experiment design. The key issue is that the $3^{\text{rd}}$ and $4^{\text{th}}$ diagrams are essentially the same – the circular perspective shows how the boundaries between elements become blurred and the emergent dynamics become the focus of interest. This becomes even more tricky to disentangle once we bring human predictive ability into the analysis.

The *Joint Cognitive Systems* approach examines the behaviour of the whole closed–loop system, and (Hollnagel and Woods, 2005) criticise the information theoretic approach. They point out that decomposition of block-diagrams, as used with engineering systems can be problematic when humans are involved, because humans are not fixed technical subsystems – they will adapt their behaviour to take into account the change in the computer system around them. This is well documented in McRuer et al.'s *crossover model*, described in (Sheridan and Ferrell, 1974; Jagacinski and Flach, 2003; McRuer and Jex, 1967), where pilots would adapt their behaviour $Y_H$ so that even with unstable controlled dynamics, the overall closed-loop behaviour near the 'crossover frequency' $\omega_c$ remained close to a 'good' servo reference behaviour

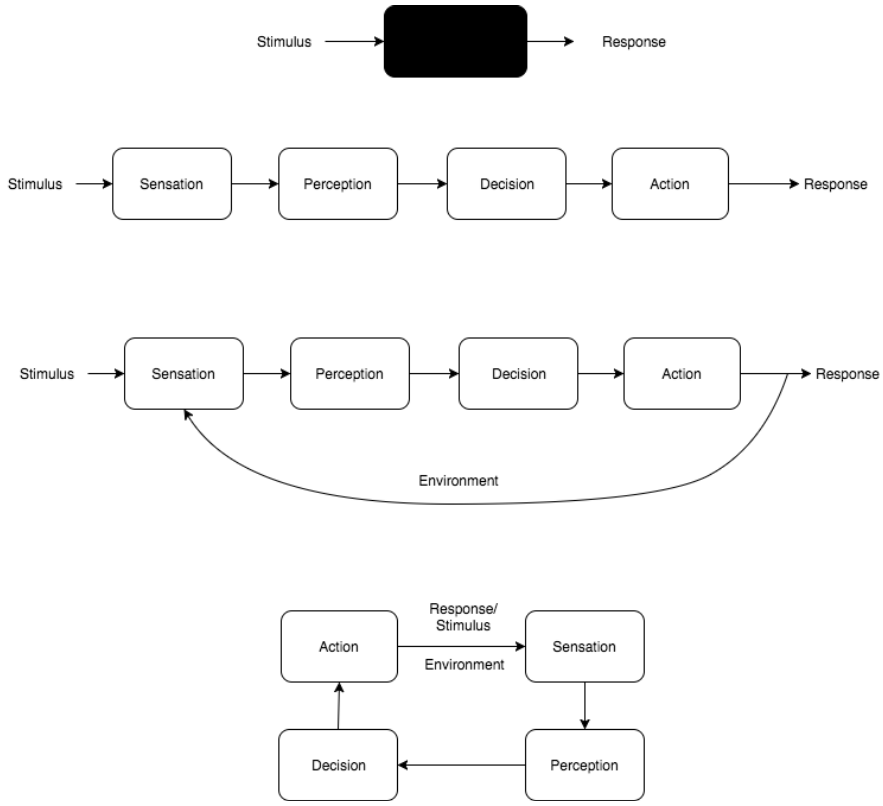


FIG. 1.9. Separation of cause and effect in the human–computer interaction loop is fundamentally problematic. (Adapted from (Jagacinski & Flach 2003)).

$$Y_H Y_C \approx \frac{\omega_c \exp(-j\omega\tau_\epsilon)}{j\omega},$$

despite changes in the aircraft dynamics ($Y_C$). $\tau_\epsilon$ represents the effective time delay, combining reaction delay and neuromuscular lag. Young provides a wide-ranging review on how the human can adapt in manual control contexts, highlighting the challenges in understanding human adaptation in complex failure settings (Young, 1969).

### 1.5.3 *Adapting the interface dynamics*

The dynamics of the computer can be adapted to try to make the task easier for the user. This includes 'sticky mouse' dynamics, magnification effects, inertia, fisheye lenses and speed-dependent zooming.

Attractors, in dynamic systems can be used to describe or implement sticky mouse dynamics, or bounce-back on images (Cockburn and Firth, 2004). (Cho, Murray-Smith and Kim, 2007) used dynamic attractors to implement a tilt-based photo browser for mobile phones. Control–display ratio adaptation (Blanch, Guiard and Beaudouin-Lafon, 2004; Casiez, Vogel, Balakrishnan and Cockburn, 2008) can be viewed as having spatially varying dynamics. Resizing the area around the target (Grossman and Balakrishnan, 2005) provides different feedback to the user and alters the effective dynamics. Enhancing pointing (Balakrishnan, 2004). Negative inertia (Barrett, Selker, Rutledge and Olyha, 1995) can be described as adding a lead term to the the system dynamics. Accelerating touchpad (Yun and Lee, 2007). (Shadmehr, 2010) looks at the temporal discounting of rewards in effort trade-offs.

1.5.3.1 *Shared control* The control representation is well suited to making clear how different control elements, whether human or automatic, can be combined to shape the closed-loop behaviour.

The contribution from different controllers can be separated out in time, via switching processes, or by frequency, via hierarchical structures, or blending mechanisms. One approach of special relevance to HCI is the *H-metaphor* (Flemisch, Adams, Conway, Goodrich, Palmer and Schutte, 2003) which proposes designing interfaces which allow users to have flexibility to switch between 'tight-reined' or 'loose-reined' control – in other words, increasing the control order and allowing variable levels of autonomy in different contexts.

## 1.6 Limitations of the control perspective for HCI

However, although almost any HCI task has a control interpretation, the natural question is whether the gain in using the concepts and tools of control theory provides a significant advantage? There are key differences from the traditional control domain, where most of the dynamic complexity was in the controlled system and disturbances applied to it, whereas with HCI most of the complexity is in the human controller. The human is a complex hierarchical controller, rapidly changing goals and working at a range of levels on a range of activities in any given period of time, subject to a wide range of external disturbances and internally generated variability.

The focus on feedback control has often overshadowed the strong feedforward/predictive effects apparent in human behaviour on typical tasks. Humans are proactive and in realistic settings they tend to anticipate issues, rather than being purely response-driven. As (Kelley, 1968) discusses, mathematical models of human control behaviour often underplayed the richness of human sensing. Will the recent developments in agents which can learn to link rich visual perception to action via deep convolutional networks (Mnih, Kavukcuoglu, Silver, Rusu, Veness, Bellemare, Graves, Riedmiller, Fidjeland, Ostrovski *et al.*, 2015) change the nature of these models?

### 1.6.1 *User heterogeneity and task uncertainty.*

A lot of the early work in manual control was focussed on well-trained pilots or drivers of vehicles which were already highly constrained in terms of viable state spaces. How much of this can we translate to the modern world of human-computer interaction, where designers need to design for a wide range of user skill levels, where the control tasks are being used primarily to transmit information, and are subject to rapid changes of reference, as the user changes their goals with exposure to new information?

A key difference between human and automatic control is that the human controller is continually going through a process of goal conception and selection (Kelley, 1968), whereas automatic control systems tend to have stable goals and cost functions. A further difference is that traditional control theory tended to have fairly simple cost functions and simple sensing. Given the complexity of human behaviour we can also question whether the complexity is in the motor control algorithm, the body dynamics, sensory perception or the nature of a possible cost function in the brain.

## 1.7 Conclusions

We argue that all fundamental building blocks in human–computer interaction have a control loop component to them – all information transfer in HCI is via control loops.

Control theory provides theoretical concepts which can provide HCI researchers and practitioners with different ways of conceiving and framing interaction problems, e.g. control elements such as state, input, order, feedback, prediction and goal, as well as practical tools for analysing, designing, measuring and documenting working interactive systems.

This gives researchers new formal analytic tools for research into the details of interaction. It also prompts us to contemplate the foundations of human–computer interaction. A key challenge, however, is the care that needs to be taken with translation of control concepts from engineering contexts, where the control is predominantly automatic, to the HCI context where the control is predominantly human. The human ability to learn, predict and adapt control behaviour means that many of the modular representations of control blocks from engineering are no longer valid. For researchers in Computational Interaction, the control loop perspective reminds us that the important thing is the closed-loop dynamic behaviour. Breaking parts of the process down and analysing these in detail in a stimulus–response manner can give a false impression of rigour, as once the overall context changes, or as the user's skill level increases, or the computer interface changes, their behaviour will also change.

### 1.7.1 *Future research challenges*

**Coping with high-dimensional input:** A recent challenge to HCI has been how to use sensed human motion as the basis for useful interaction. Recent improvements in sensor technology support the availability of high-dimensional sensor streams which could enable *natural user interfaces*, but designers have struggled to convert this to usable interaction. At the core of any successful mapping of rich, high-dimensional data to user-controllable systems will be the creation of mappings to low-dimensional spaces that users can control. This process has a lot in common with concepts which recur in the chapters of this edited volume: *distance measures*, *cost functions*, *inverse problems* and *optimisation*. Using computationally complex systems such as deep convolutional neural nets to analyse a series of images and classify these into different perceived behaviours is an example of dimensionality reduction. To become *useful* interaction, however, we need to be able to take these and turn them into control loops with appropriate feedback and associated decision logic for state transitions in the system.

**Embedding control tools in development environments:** An important practical challenge is to enhance the support for control-theoretic design tools and visualisations within the typical development environments used by HCI researchers and developers.

Control models are not just analytic, they are *generative* models which can create the behaviour in real-time – we can create control agents that can be released in the testing phase to predict performance (time and error rates) on a given interface. This fits well with recent developments in instrumented interaction and simulation environments for AI systems, such as OpenAI's *Gym*[5] or Google Deepmind's *Lab* [6]. These will potentially allow us to acquire large amounts of natural interaction behaviour, and use machine learning tools to learn and test dynamic systems which replicate human control behaviour which includes the impact of visual perception.

[5] `https://gym.openai.com/`

[6] `https://deepmind.com/research/open-source/open-source-environments/`

# REFERENCES


Accot, J. and Zhai, S. (1997). Beyond Fitts' Law: models for trajectory-based HCI tasks. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, pp. 295–302. ACM Press.

Accot, J. and Zhai, S. (2002*a*). More than dotting the i's — foundations for crossing-based interfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, pp. 73–80. ACM Press.

Accot, Johnny and Zhai, Shumin (2002*b*). More than dotting the i's - Foundations for crossing-based interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02* (4), 73–80.

Apps, Matthew A. J., Grima, Laura L., Manohar, Sanjay, and Husain, Masud (2015). The role of cognitive effort in subjective reward devaluation and risky decision-making. *Scientific reports*, **5**, 16880.

Balakrishnan, Ravin (2004). 'Beating' Fitts' law: virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, **61**(6), 857–874.

Barrett, Rob C., Selker, Edwin J., Rutledge, Joseph D., and Olyha, Robert S. (1995). Negative inertia: A dynamic pointing function. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, pp. 316–317. ACM.

Bennett, K. B. and Flach, J. M. (2011). *Display and interface design: Subtle science, exact art.* CRC Press.

Blanch, Renaud, Guiard, Yves, and Beaudouin-Lafon, Michel (2004). Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, New York, NY, USA, pp. 519–526. ACM.

Cannon, David J. (1994). Experiments with a target-threshold control theory model for deriving Fitts' law parameters for human-machine systems. *IEEE Trans Syst Man Cybern*, **24**(8), 1089–1098.

Casiez, Géry and Roussel, Nicolas (2011). No more bricolage!: Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, New York, NY, USA, pp. 603–614. ACM.

Casiez, Géry, Vogel, Daniel, Balakrishnan, Ravin, and Cockburn, Andy (2008). The impact of control-display gain on user performance in pointing tasks. *Human–computer interaction*, **23**(3), 215–250.

Cho, Sung-Jung, Murray-Smith, Roderick, and Kim, Yeun-Bae (2007). Multi-context photo browsing on mobile devices based on tilt dynamics. In *Proceedings of the 9th International Conference on Human*

*Computer Interaction with Mobile Devices and Services*, MobileHCI '07, New York, NY, USA, pp. 190–197. ACM.

Clarke, Christopher, Bellino, Alessio, Esteves, Augusto, Velloso, Eduardo, and Gellersen, Hans (2016). Tracematch: A computer vision technique for user input by tracing of animated controls. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, New York, NY, USA, pp. 298–303. ACM.

Cockburn, Andy and Firth, Andrew (2004). Improving the acquisition of small targets. In *People and Computers XVII Designing for Society*, pp. 181–196. Springer.

Conant, Roger C. and Ross Ashby, W. (1970). Every good regulator of a system must be a model of that system. *International Journal of Systems Science*, **1**(2), 89–97.

Connelly, Edward M. (1984). A control model: an alternative interpretation of Fitts' law. In *Proceedings 28th annual meeting of the Human Factors Society*.

Costello, R.G. (1968). The surge model of the well-trained human operator in simple manual control. *IEEE Transactions on Man–Machine Systems*, **9**(1), 2 – 9.

Craik, Kenneth J. W. (1947). Theory of the human operator in control systems: 1. the operator as an engineering system. *British Journal of Psychology. General Section*, **38**(2), 56–61.

Craik, Kenneth J. W. (1948). Theory of the human operator in control systems: 2. man as an element in a control system. *British Journal of Psychology. General Section*, **38**(3), 142–148.

Crossman, E. R. F. W. (1960). The information-capacity of the human motor-system in pursuit tracking. *Quarterly Journal of Experimental Psychology*, **12**(1), 01–16.

Crossman, E. R. F. W. and Goodeve, P. J. (1983). Feedback control of hand-movement and Fitts' law. *The Quarterly Journal of Experimental Psychology*, **35**(2), 251–278.

Dix, A., Finlay, J., Abowd, G., and Beale, R. (2004). Human-computer interaction. *England: Pearson Education Limited*.

Drury, Colin G (1971). Movements with lateral constraint. *Ergonomics*, **14**(2), 293–305.

Eslambolchilar, Parisa and Murray-Smith, Roderick (2006). Model-based, multimodal interaction in document browsing. In *Machine Learning for Multimodal Interaction, LNCS Volume 4299*, pp. 1–12. Springer.

Eslambolchilar, Parisa and Murray-Smith, Roderick (2008). Control centric approach in designing scrolling and zooming user interfaces. *International Journal of Human-Computer Studies*, **66**(12), 838–856.

Eslambolchilar, Parisa and Murray-Smith, Roderick (2010). A model-based approach to analysis and calibration of sensor-based human interaction loops. *International Journal of Mobile Human Computer Interaction*, **2**(1).

Eykhoff, Pieter (1994). "Every good regulator of a system must be a model of that system". *Modeling, identification and control*, **15**(3), 135.

Fekete, Jean-Daniel, Elmqvist, Niklas, and Guiard, Yves (2009). Motion-pointing: Target selection using elliptical motions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, New York, NY, USA, pp. 289–298. ACM.

Fitts, Paul M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, **47**(6), 381–391.

Fitts, Paul M. and Peterson, James R. (1964). Information capacity of discrete motor responses. *Journal of experimental psychology*, **67**(2), 103.

Flemisch, O., Adams, A., Conway, S. R., Goodrich, K. H., Palmer, M. T., and Schutte, P. C. (2003). The H-Metaphor as a guideline for vehicle automation and interaction. Technical Report NASA/TM 2003-212672.

Gawthrop, Peter, Gollee, Henrik, and Loram, Ian (2015). Intermittent control in man and machine. In *Event-Based Control and Signal Processing* (ed. M. Miskowicz), pp. 281–350. CRC Press.

Gawthrop, Peter, Lakie, Martin, and Loram, Ian (2008). Predictive feedback control and Fitts' law. *Biological Cybernetics*, **98**, 229–238.

Gawthrop, Peter, Loram, Ian, Lakie, Martin, and Gollee, Henrik (2011). Intermittent control: A computational theory of human control. *Biological Cybernetics*, **104**(1-2), 31–51.

Graves, Alex (2013). Generating sequences with recurrent neural networks. *CoRR*, **abs/1308.0850**.

Grossman, Tovi and Balakrishnan, Ravin (2005). The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, New York, NY, USA, pp. 281–290. ACM.

Guiard, Yves and Rioul, Olivier (2015). A mathematical description of the speed/accuracy trade-off of aimed movement. In *Proceedings of the 2015 British HCI Conference*, pp. 91–100. ACM.

Hollnagel, Erik (1999). Modelling the controller of a process. *Transactions of the Institute of Measurement and Control*, **21**(4-5), 163–170.

Hollnagel, E. and Woods, D. D (2005). *Joint cognitive systems: Foundations of cognitive systems engineering.* CRC Press.

Hornbæk, Kasper and Oulasvirta, Antti (2017). What is interaction? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.*

Jagacinski, Richard J. and Flach, John M. (2003). *Control Theory for Humans: Quantitative approaches to modeling performance*. Lawrence Erlbaum, Mahwah, New Jersey.

Kelley, Charles R. (1968). *Manual and Automatic Control: A Theory of Manual Control and Its Applications to Manual and to Automatic Systems.* Academic Press.

Kleinman, DL, Baron, S, and Levison, WH (1970). An optimal control model of human response part i: Theory and validation. *Automatica*, **6**(3), 357–369.

Kleinman, David L., Baron, Sheldon, and Levison, W. H. (1971). A control theoretic approach to manned-vehicle systems analysis. *IEEE Trans. Automatic Control*, **16**, 824–832.

Kratz, Sven, Brodien, Ivo, and Rohs, Michael (2010). Semi-automatic zooming for mobile map navigation. In *Proc. of the 12th Int. Conf. on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, New York, NY, USA, pp. 63–72. ACM.

Lank, E. and Saund, E. (2005). Sloppy selection: Providing an accurate interpretation of imprecise selection gestures. *Computers and Graphics*, **29**, 490?–500.

Lantz, V. and Murray-Smith, R. (2004, October 23-27). Rhythmic interaction with a mobile device. In *NordiCHI '04, Tampere, Finland*, pp. 97–100. ACM.

MacKenzie, I. Scott (1992). Fitts' law as a research and design tool in human–computer interaction. *Human-computer interaction*, **7**(1), 91–139.

McRuer, Duane T. and Jex, Henry R. (1967). A review of quasi-linear pilot models. *IEEE Trans. on Human Factors in Electronics*, **8**(3), 231–249.

Meyer, David E., Smith, J. E. Keith, Kornblum, Sylvan, Abrams, Richard A., and Wright, Charles. E. (1990). Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. In *M. Jeannerod (Ed.), Attention and Performance XIII*, pp. 173–226. Laurence Erlbaum.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K., Ostrovski, Georg et al. (2015). Human-level control through deep reinforcement learning. *Nature*, **518**(7540), 529–533.

Müller, Jörg, Oulasvirta, Antti, and Murray-Smith, Roderick (2017). Control theoretic models of pointing. *ACM Transactions on Computer Human Interaction*.

Pastel, Robert (2006). Measuring the difficulty of steering through corners. *Proceedings of ACM CHI 2006 Conference on Human Factors in Computing Systems*, **1**(2), 1087–1096.

Poulton, E. C. (1974). *Tracking skill and manual control.* Academic Press, New York.

Powers, William T. (1973). *Behavior: The control of Perception.* Aldine de Gruyter.

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.

Quinn, Philip, Cockburn, Andy, Casiez, Géry, Roussel, Nicolas, and Gutwin, Carl (2012). Exposing and understanding scrolling transfer functions. In *Proceedings of the 25th annual ACM symposium on User* *interface software and technology*, pp. 341–350. ACM.

Quinn, Philip, Malacria, Sylvain, and Cockburn, Andy (2013). Touch scrolling transfer functions. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, New York, NY, USA, pp. 61–70. ACM.

Quinn, Philip and Zhai, Shumin (2016). Modeling gesture-typing movements. *Human–Computer Interaction*. Accepted author version posted online.

Rashevsky, N. (1959). Some remarks on the mathematical aspects of automobile driving. *Bulletin of Mathematical Biology*, **21**(3), 299–308.

Rashevsky, N. (1960). Further contributions to the mathematical biophysics of automobile driving. *Bulletin of Mathematical Biology*, **22**(3), 257–262.

Rigoux, Lionel and Guigon, Emmanuel (2012). A model of reward-and effort-based optimal decision making and motor control. *PLoS Comput Biol*, **8**(10), e1002716.

Rogers, Yvonne, Sharp, Helen, and Preece, Jenny (2011). *Interaction Design: Beyond Human Computer Interaction, 4th edition*. Wiley.

Schmidt, Richard A. and Lee, Timothy D. (2005). *Motor Control and Learning*. Human Kinetics.

Shadmehr, Reza (2010). Control of movements and temporal discounting of reward. *Current Opinion in Neurobiology*, **20**(6), 726 – 730.

Shadmehr, Reza, Huang, Helen J, and Ahmed, Alaa A (2016). A representation of effort in decision-making and motor control. *Current biology*, **26**(14), 1929–1934.

Sheridan, Thomas B. and Ferrell, William R. (1974). *Man-machine Systems: Information, Control, and Decision Models of Human Performance*. M.I.T. Press, Cambridge, USA.

Velloso, E., Carter, M., Newn, J., Esteves, A., Clarke, C., and Gellersen, H. (2017). Motion correlation: Selecting objects by matching their movement. *ACM Transactions on Computer-Human Interaction (ToCHI)*, **24**(3).

Visell, Y. and Cooperstock, J. (2007, Oct). Enabling gestural interaction by means of tracking dynamical systems models and assistive feedback. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pp. 3373–3378.

Viviani, Paolo and Terzuolo, Carlo (1982). Trajectory determines movement dynamics. *Neuroscience*, **7**(2), 431–437.

Wickens, Christopher D. and Hollands, Justin G. (1999). *Engineering Psychology and Human Performance* (Third edn). Prentice-Hall.

Wiener, Norbert (1948). *Cybernetics: Control and communication in the animal and the machine*. Wiley New York.

Williamson, John and Murray-Smith, Roderick (2004). Pointing without a pointer. In *roceedings of ACM SIG CHI 2004 Conference on Human Factors in Computing Systems*, pp. 1407–1410. ACM.

Yamanaka, Shota and Miyashita, Homei (2016*a*). Modeling the steering time difference between narrowing and widening tunnels. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing*

*Systems*, CHI '16, New York, NY, USA, pp. 1846–1856. ACM.

Yamanaka, Shota and Miyashita, Homei (2016*b*). Scale effects in the steering time difference between narrowing and widening linear tunnels. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, New York, NY, USA, pp. 12:1–12:10. ACM.

Young, L. R. (1969, Dec). On adaptive manual control. *IEEE Transactions on Man-Machine Systems*, **10**(4), 292–331.

Yun, Sumi and Lee, Geehyuk (2007). Design and comparison of acceleration methods for touchpad. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*, pp. 2801–2812. ACM.

Zhai, Shumin, Kristensson, Per Ola, Appert, Caroline, Anderson, Tue Haste, and Cao, Xiang (2012). Foundational issues in touch-surface stroke gesture design - an integrative review. *Foundations and Trends in Human–Computer Interaction*, **5**(2), 95–205.