# Stratified, computational interaction via machine learning

Roderick Murray-Smith[1]

*Abstract*— **We present a control loop framework which enables humans to flexibly adapt their level of engagement in human–computer interaction loops by delegating varying elements of sensing, actuation and control to computational algorithms. We give examples of the use of deep convolutional networks in: modelling and inferring hand pose, single pixel cameras for vision in non visible wavelengths and in a music information retrieval system. In each case we explore how the user can adapt the nature of their closed-loop interaction, depending on context.**

## I. INTRODUCTION

We will discuss how we can support a human user to have more control over the level of engagement required of their interactions with technology by representing the human–computer interaction process as a control loop. We then use computational methods to augment the various blocks in the human–computer control loop. We give specific examples from touch interaction and music information retrieval and explore how we can add the ability to flexibly adapt the nature of the closed-loop interaction in each case.

## II. THE HUMAN–COMPUTER CONTROL LOOP

Traditionally, Human–Computer Interaction (HCI) is often presented 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, but this does not provide an obvious way to measure the communication, or whether the communication makes a difference.

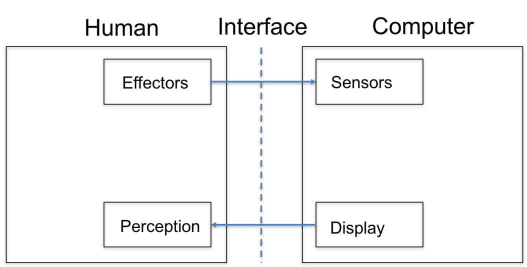


Fig. 1. Human–Computer Interaction as a closed-loop system

One 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 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. [1], [2] argue 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, the volume of a music player,[1] the destination of an autonomous vehicle or some remote computational system. This can be seen in the evolution of human–machine symbiosis from direct action with our limbs, via tools and powered control to control of computationally enhanced systems where significant elements of the information fedback to the human, the coordination of control actions and the proposals for new sub-goals are augmented computationally. Over time this has led to an increasing indirectness of the relationship between the human and the controlled variable, with a decrease in muscular strength and an increasing role for sensing and thought [3]. *Computational Interaction* will further augment or replace elements of the perceptual, cognitive and actuation processes in the human with artificial computation.

### A. History of Control in human–computer interaction

*Manual control theory* [4], [5] 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 [6], [7], and became more well-known in the broader framing of Wiener's *Cybernetics* [8]. As observed in [9], 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. [3], [10]) 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'

*We gratefully acknowledge supported by EC ICT Programme Project H2020-643955, MoreGrasp, and EPSRC project QuantIC and from Bang & Olufsen and the Danish Research Council.

[1]Prof. R. Murray-Smith is with the School of Computing Science, University of Glasgow, Scotland. r.murray-smith@glasgow.ac.uk

[1]One might think the reference here is the loudness of the music, but it is probably the happiness of the people in the room that is being controlled.

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. [11] reviews the early tracking literature, and an accessible review of the basic approaches to manual control can be found in [12].

Control theory provides an engineering framework which is well-suited for analysis of closed-loop interactive systems. This can include properties such as feedback delays, sensor noise (see e.g. [13]), or interaction effects like 'sticky mouse' dynamics, isometric joystick dynamics [14], magnification effects, inertia, fisheye lenses, speed-dependent zooming, all of which can be readily represented by dynamic models. The use of state space control methods was explored in document zooming context in [15], [16], [17], [18] and [19] reviewed the challenge of optimising scrolling transfer functions and used a robot arm to identify the dynamics of commercial products. Examples of the use of dynamic models in interactive systems are now widespread in commercial systems, and there are also examples in the academic literature, including [20], [21]. [22] uses control models to understand how input trajectories associated with words entered into gesture keyboards are likely to vary. The control perspective can also lead to unusual approaches to interaction, such as interfaces which inferred the user's intent based on detection of control behaviour, as developed in [23] and built on by [24].

## III. Casual interaction

As we introduced in [25], interface designers often assume that users focus on their device when interacting, but this is often not the case. In fact, there are many scenarios where users are not able to, or do not want to, fully engage with their devices. In general, inhibiting factors can be divided into (1) physical, (2) social, or (3) mental subgroups. *Physical reasons* users can not fully engage with a device are often closely related to questions of accessibility. *Social reasons* are mostly concerned with the question of how much engagement with a device is acceptable in a given setting. *Mental reasons* are primarily issues of distraction. Users might be engaged in a primary task, leaving little attention for other interactions.

A common feature in these scenarios is that the users still *want* to engage with their devices, just sometimes with lower effort and precision or in a more sporadic fashion. Casual interaction mechanisms should allow users to control the *level* to which they engage—they do not want to give up control completely. These are also not purely casual systems: the interaction space is continuous, spanning from very focused to very casual interactions. When interacting with devices, the level of engagement with the device will differ depending on the situation (whether the current constraints were physical in nature or related to social or mental state). The high-engagement extreme describes very focused interactions, in which a fully committed user is giving her entire attention to the interaction, and actions and responses are tightly coupled in time. Playing games or controlling a vehicle often falls in this category. On the other end of the scale are interactions that are of secondary or incidental nature. For example, muting an alarm clock by tapping it anywhere, or turning over a phone to reject a call can be done without giving the task too much attention. There can even be levels of casual interaction within otherwise highly focused settings – the popularity of kinetic scrolling in touchscreen devices is partly because after an initial flick, the user can pull back their finger, reducing their engagement level, and wait to see where they end before stopping the scrolling, or giving it a further impetus. In his Ph.D. thesis Pohl highlights the trade-offs between *control* and *engagement* [26].

*Focussed–Casual spectrum:* Figure 2 arranges common interaction types on a casual-precise spectrum. A very broad class of useful interactions fall into the discrete action and parametric adjustment models. In particular, these are the interaction types that suit disengaged, casual interaction, as opposed to attention-focused, detailed interaction.

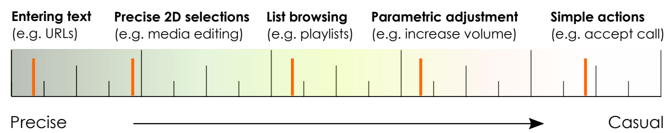


Fig. 2. User interface actions laid out on a spectrum from precise, engaged interaction to casual, disengaged interaction.

### A. Lifting the Performance–effort curve

A 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. From an interaction perspective, it can be argued that the role of AI is to lift the curve, as show in Figure 3. For a given level of human effort, the symbiosis of human and intelligent algorithms should lead to higher performance. This was originally explored in [27], but can be used to motivate the use of computational resources. In most cases algorithmic augmentation is used to raise the performance curve for low human effort – we want performance for very little effort. It becomes interesting at the top end of the effort scale. Will the augmented performance be lower or higher than the unaided human performance, when the human is putting in maximum effort?

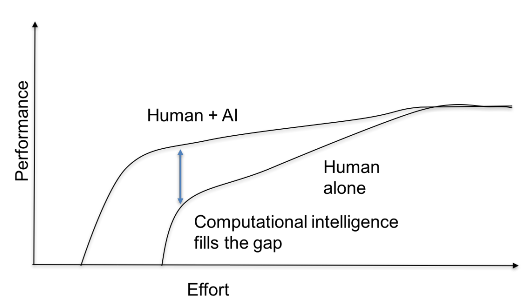


Fig. 3. The role of computational intelligence in interaction loops is to push the performance curve up – higher performance for the same amount of effort.

Are there tasks or combinations of human and computational support which lead to lower performance?

*Dangers of filling the performance gap:* Whenever computational intelligence is used to raise the performance of a human–computer combination for a given level of human effort, or equivalently, achieving more bits per second than the human was able or willing to generate. Where does this missing information come from? In some cases we use past behaviour by the current user, in others information about 'similar' users. Sometimes the control might have purely corporate or business priorities, rather than the end users'.

## IV. Examples of Computational Interaction

Three areas illustrate where machine learning can be applied to human–computer interaction to augment or replace elements of the perceptual, cognitive and actuation processes in the human with artificial computation:

1) Inferring human intent via sensed human action
2) Inferring subjective aspects of media content retrieval
3) Augmenting ability to sense the environment state

Each of these has specific challenges, once the human receives non-trivial computational support.

### A. 3D touch interaction

One of the most rapidly expanding approaches to inferring human intent is to sense changes in the pose and position of the human hand to control an interface.

*Inference of human pose – 3D capacitive touch:* The ability to sense finger position and pose accurately a distance from the device screen would allow designers to create novel interaction styles, and researchers to better track, analyse and understand human touch behaviour. Progress in design of capacitive screen technology has led to the ability to sense the user's fingers up to several centimetres above the screen. However, the inference of position and pose is inherently uncertain given only the readings from the two-dimensional capacitive sensor pads.

To infer finger pose and position away from the touch surface we need a) a sensing technology which can detect the human hand at a distance from the screen and b) an inference mechanism which can estimate the pose and position given the raw sensor readings. The sensor technologies involved will rarely provide a simple reading which will return the position $(x, y, z)$ and pose (pitch, roll and yaw, $\theta, \phi, \psi$), so inference of finger pose in 3D has two general approaches:

1) The creation of a complex **nonlinear, multivariable regression mapping**. This is an *inverse model* from a possibly high-dimensional sensor-space $X$ to the original $(x, y, z, \theta, \phi, \psi)$ vector.[2]
2) The creation of a **causal forward model** from $(x, y, z, \theta, \psi)$ to image space $X$, which can then be used to find the values of $(x, y, z, \theta, \psi)$ which minimise the difference between the observed sensor readings $X$ and the inferred readings $\hat{X}$.

[2]In this paper we do not attempt to model the roll angle $\phi$, as that is not feasible with the capacitive technology used in our system.

To get a sense of the technical challenge, we visualise the nature of the sensor readings $X$ for several poses in Figure 4. We use a touch screen with a prototype transparent capacitive sensor with an extended depth range of between $0\,\mathrm{cm}$ to $5\,\mathrm{cm}$ from the screen (although accuracy decreases with height) and a resolution of $10 \times 16$ pads and a refresh rate of 120 Hz. We sampled at $60\,\mathrm{Hz}$. It was embedded in a functional mobile phone.

*Deep Networks:* Our first approach to infer the finger pose uses Deep Convolutional networks (DCN). DCNs have a long history [28], but have made significant progress in recent years [29], [30]. In order to learn the mapping between sensor inputs and finger poses we need a large, carefully calibrated training set of fingers in different poses. The mappings can be extremely complex, so generating data with human users is unrealistically effort and time intensive. We initially used robot-generated inputs and then moved to larger sets generated by an electrostatic simulator. The network was implemented in Python, using the Keras library [31]. The architecture used was a standard one used for image processing, with four 2D $3 \times 3$ convolution layers (32 units each), with pooling, fully connected layers, and linear layers in the final densely connected layer for the regression element. Rectified linear (ReLU) activation functions were used. The input layer adds Gaussian noise of std. deviation 0.0079, as the mean observed on our physical sensor at rest.

We tested the trained model on 20% of the data removed before training (3664 points). The pitch RMSE was $8.6°$ and yaw RMSE was $24.1°$. The $x, y$ RMSE was 0.2cm, the RMSE on $z$ was $0.1\,\mathrm{cm}$. These errors increase with $z$.

*Accelerating electrostatic simulation:* One aspect of computational interaction is that if forward models can operate fast enough, we can build first-principles generative models into the sensing process. The challenge is often computational efficient. Flexible statistical models have been used to create more efficient representations of computationally complex simulators [32], [33], [34]. This process requires initial simulation to generate training data for a machine learning solution, which can then run more rapidly than the original data – it can be viewed as a 'glorified lookup table' which performs inference between the observations to avoid exponential explosions of required storage. We can represent the simulator in the form of a function $y = f(x)$. Each run of the simulator is defined to be the process of producing one set of outputs $y$ for one particular input configuration $x$. We assume that the simulator is deterministic, that is, running the simulator for the same $x$ twice will yield the same $y$. The

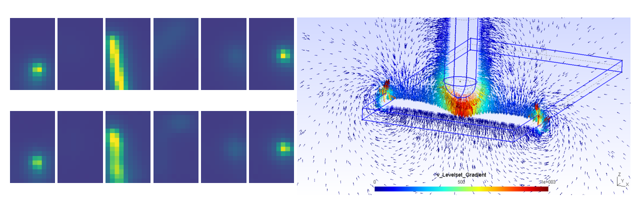

Fig. 4. Left. Comparison of simulation outputs (upper row) and the deep network model of the same pose (lower row). Right: Field plot from simulation of a finger above the capacitive screen.

DCN can therefore be used to create an accelerated implementation of the forward, or causal model implemented by the electrostatic simulation (or from physical experiments). This can then be used to infer the most likely inputs, by minimising the distance between the current sensed values and the sensor predictions from the model output.

Our tests demonstrated that a combination of the neural net implementations of regression model with the real-time forward model within a particle filter allowed us to create robust tracking of the finger pitch and yaw in 3D.

*Flow-based interaction:* As we described in the previous section, machine learning can be used to infer pose and position, but unknown hand sizes and postures mean that there is a high-dimensional latent vector required to describe the problem fully, and that even with knowledge of that, the task of solving the inverse problem to infer, e.g. $(x_i, y_i, z_i, \phi_i, \psi_i)$ from the information available in the sensor pad matrix $X$ is an ill-posed problem with many solutions compatible with the data. This means that methods which explicitly track finger states must cope with this inherent variance in estimates, and requires sophisticated filtering and processing to regularise the problem, and distinguish fingers from distractors such as the palm or arm. A further challenge specific to 3D touch interfaces is human motor variability in mid-air gesturing. People struggle to control unsupported fingers in mid-air with sufficient accuracy and consistency for conventional pose-based interaction. This is exacerbated when devices are used in typical mobile contexts where the user or the vehicle they are in might be in motion.

An alternative proposal led by Williamson [35] called *flow-based interaction* is to sidestep the cursor-based interface, and the concomitant requirement to invert the sensed electrical field to resolve the 3D structure above, and instead, implement a mediating layer between sensor and application state, which creates a relatively simple closed-loop system which the user can stimulate with around-device motions in a predictable fashion. The goal is that this dynamic system should be responsive to the typical range of human movement in interaction without being overly sensitive to the variance inherent in low-level inferences about hand state at each point in time. It should ideally act as a low-dimensional representation of the recent evolution of the high-dimensional input which can both be fed back to the user to give them formative feedback about how their physical actions were sensed, and used by automated classifiers to generate actions to be applied to the application interface. A flow-based interaction eliminates discrete "contact" points, in favour of estimating the overall motion field above a sensor. This field is used to drive the interaction *directly*, which avoids many of the artifacts common in touch-based interaction; for example it is much less sensitive to acquiring or dropping fingers during interaction, and it makes weak assumptions about the interacting objects, so is equally applicable to touch interaction with gloves, styli, glasses or other everyday objects. Because it computes relative changes between sensor "images", it also requires less precise calibration than standard tracking. The motion fields are particularly convenient to display visually, giving a rich, clean and responsive representation of the sensed activity. As a trade-off, it is also less precise than cursor-based control and requires development of new interface structures to take advantage of the flow-based models.

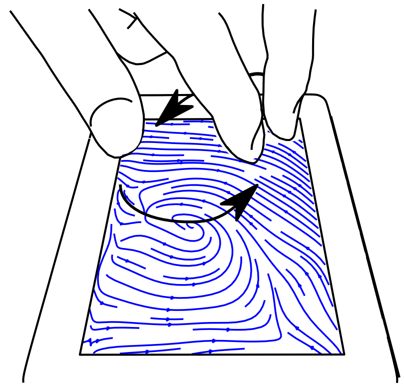

Fig. 5. In *Flow-based interaction* twirling the fingers above the screen surface induces a characteristic *motion field*.

### B. Music Information Retrieval

In [36], we built and evaluated a system to interact with 2D music maps, based on dimensionally-reduced inferred subjective aspects such as mood and genre, using a flexible pipeline of acoustic feature extraction, nonlinear dimensionality reduction and probabilistic feature mapping. interactive music exploration tool which offers interaction at a range of levels of engagement, which could foster directed exploration of music spaces, casual selection and serendipitous playback.

The features are generated by the commercial Moodagent Profiling Service[3] for each song, computed automatically from low-level acoustic features, based on a machine-learning system which learns feature ratings from a small training set of human subjective classifications. These inferred features are uncertain. Subgenres of e.g. electronic music are hard for expert humans to distinguish, and even more so for an algorithm using low-level features. This motivated representing the uncertainty of features in the interaction with the user.

When working with Bang & Olufsen on new forms of music interaction, an approach more appropriate for their style and market required the development of an even simpler interaction, which is described in [37] and expanded on in more detail in Boland's Ph.D. thesis [38]. The results were translated to a flagship product for B&O, shown in Figure 7. This interface is extremely simple – the default face does not even have a display. The music content is placed on a 1D-mood and genre manifold, and the user can explore genres by running their finger around a circular groove on the wooden surface, or using casual left/right swipes to navigate a music playlist. If, however, the user wishes to engage more, they can physically turn the device over, and access a fully featured touch interface and display on the reverse side.

[3] http://www.moodagent.com/

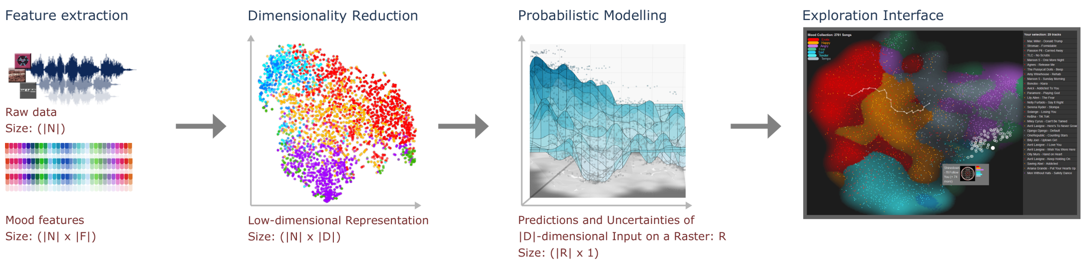


Fig. 6. (a) An audio collection, described by a large set of features automatically extracted from the content. (b) visualisation of this high-dimensional dataset in two dimensions using dimensionality reduction (c) probabilistic models showing the distribution of specific features in the low dimensional space (d) combining dimensionality reduction with these models to build an interactive exploration interface.

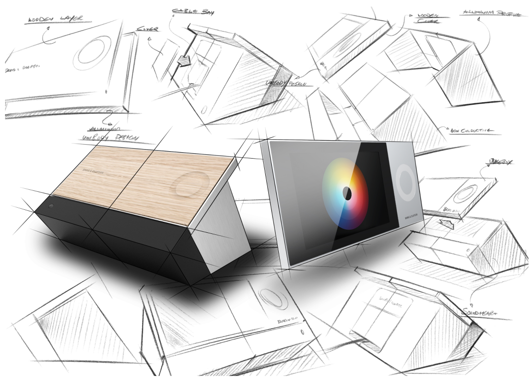

Fig. 7. Bang & Olufsen BeoSound Moment design as an example of *Casual Interaction*. The system has two sides – a simple wooden touch surface and a traditional touch screen.

### C. Single Pixel Cameras via Deep autoencoders

Computational intelligence can also be used to sense the environment around the human. Within the QuantIC project, we have been exploring the use of Single-pixel cameras for rapid prototyping of novel sensors for imaging beyond the visible spectrum. In order to achieve real-time performance in solving the associated inverse problems, we have again used Deep Convolutional networks as auto-encoders, then implemented the initial layers in optics as binary weighted digital mirror arrays, as shown in Figure 8 from [39]. Rather than performing computationally-intensive matrix inversions, the use of a neural net allows real-time video rate decoding for $128 \times 128$ images, and we can learn optimal mirror bases for target image collections. However, the image reconstructed will be very dependent on image collection used to train the system. A challenge for safe, practical use is therefore that the user must be aware of the biases such a sensing system has been optimised on. E.g. in a conflict context an imaging system which had only been trained to encode military vehicles might create misleading images. This opens a requirement for interfaces which allow users to easily perturb and manipulate prior expectations which would change the associated network parameters and act as a form of sensitivity analysis, generating alternative interpretations of the scene.

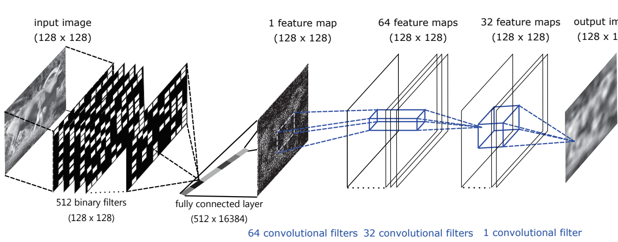


Fig. 8. Convolutional autoencoder super-resolution single-pixel camera architecture. Initial binary weights layers implemented by a digital mirror array and photodiode. 'Inverse problem' is 'solved' by the decoding layers.

## V. Shared Control

The previous section showed how we can use computational intelligence to augment different aspects of the sensing, inference and feedback blocks in the human–computer control loop. A key design task is to explore how human or automatic control loop elements can be dynamically 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 is the *H-metaphor* [40] 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.

### A. Stratification of Interaction Loops

A natural generalisation of the H-metaphor ideas is that we should design systems which can switch between a discrete number of different interaction loops by dynamically interchanging blocks of different complexity for elements that, e.g. 1. sense the human, 2. infer human goals, 3. sense the environment. As discussed earlier, these interaction strata will tend to be arranged from low engagement, casual interactions to high-engagement, focused interaction.

In traditional hierarchical control, the user's task was made easier by augmented control models, such as velocity- or bearing-hold modes, where inner loops tended to be faster, outer loops slower. These tended to be used by expert users (process operators, pilots) and the hierarchy of automation loops was associated with historically evolutions in automation. Modern applications will need to work harder on appropriate design metaphors to support new, diverse groups of users in such systems.

### B. Establishing grasp for people with spinal-cord injuries

The aim of the MoreGrasp project is to develop a non-invasive, adaptive, multimodal user interface including a brain-computer interface (BCI) for intuitive control of a semi-autonomous motor and sensory grasp neuroprosthesis supporting individuals with high spinal cord injury to re-establish hand grasp for everyday activities [41]. In this case the user uses wither a BCI or a shoulder sensor to change grasp settings and initiate actions. However, we can create simpler interaction loops by instrumenting objects in their environment, as shown in Figure 9 such that when they touch them an appropriate grasp is automatically used. Furthermore, understanding of the accuracy of sensors, dynamics of physiological systems and likely goals can be used to support users with limited actuation bandwidth (Figure 10).

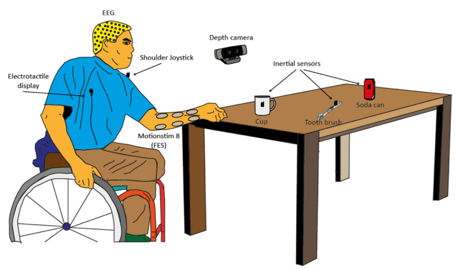


Fig. 9. A disabled user being supported by instrumented objects and external sensors. Sensors on known objects can reduce need for user to specify details of grasp style.

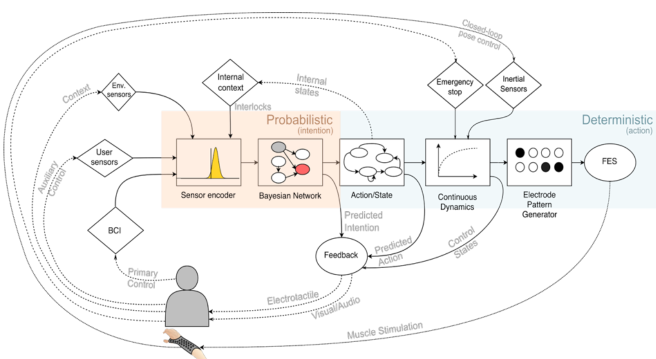


Fig. 10. A shared control structure, showing the internal processing pipeline and the user in the loop. A probabilistic intention decoder is connected to a deterministic action synthesizer.

## VI. Outlook

We presented a Computational perspective of Human Computer Interaction, within a control framework, where we explicitly design a number of ways to close the loop, with a trade-off between computational support and individual freedom of control. We gave examples of computational interaction systems which benefit from computational intelligence to provide more flexibility on sensing human actions, inferring human queries and sensing environments for humans. However, we face challenges to create systems with compelling use metaphors which deliver a clear user experience. It will require us to bring together new styles of development teams, including interaction designers, machine learning experts and sensor specialists.

## Acknowledgements

The work presented here reviews work done by the members of the Inference, Dynamics and Interaction group over the last few years, including J. Williamson, D. Boland, B. Vad, C. Higham, S. Rogers, H. Pohl, S. Stein and A. Ramsay, and in Physics from M. Edgar and M. Padgett.

## References

[1] E. Hollnagel, "Modelling the controller of a process," *Transactions of the Institute of Measurement and Control*, vol. 21, no. 4-5, pp. 163–170, 1999.

[2] E. Hollnagel and D. D. Woods, *Joint cognitive systems: Foundations of cognitive systems engineering*. CRC Press, 2005.

[3] C. R. Kelley, *Manual and Automatic Control*. John Wiley and Sons, Inc., New York, 1968.

[4] D. T. McRuer and H. R. Jex, "A review of quasi-linear pilot models," *IEEE Trans. on Human Factors in Electronics*, vol. 8, no. 3, pp. 231–249, 1967.

[5] R. Costello, "The surge model of the well-trained human operator in simple manual control," *IEEE Transactions on Man–Machine Systems*, vol. 9, no. 1, pp. 2 – 9, 1968.

[6] K. J. W. Craik, "Theory of the human operator in control systems: 1. the operator as an engineering system," *British Journal of Psychology. General Section*, vol. 38, no. 2, pp. 56–61, 1947.

[7] ——, "Theory of the human operator in control systems: 2. man as an element in a control system," *British Journal of Psychology. General Section*, vol. 38, no. 3, pp. 142–148, 1948.

[8] N. Wiener, *Cybernetics: Control and communication in the animal and the machine*. Wiley New York, 1948.

[9] C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*, 3rd ed. Prentice-Hall, 1999.

[10] T. B. Sheridan and W. R. Ferrell, *Man-machine Systems: Information, Control, and Decision Models of Human Performance*. M.I.T. Press, Cambridge, USA, 1974.

[11] E. C. Poulton, *Tracking skill and manual control*. New York: Academic Press, 1974.

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

[13] D. Trendafilov and R. Murray-Smith, "Information-theoretic characterization of uncertainty in manual control," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4913–4918.

[14] R. C. Barrett, E. J. Selker, J. D. Rutledge, and R. S. Olyha, "Negative inertia: A dynamic pointing function," in *Conference Companion on Human Factors in Computing Systems*, ser. CHI '95. ACM, 1995, pp. 316–317.

[15] P. Eslambolchilar and R. Murray-Smith, "Tilt-based automatic zooming and scaling in mobile devices – a state-space implementation," in *Mobile HCI 2004*, S. Brewster and M. Dunlop, Eds., vol. 3160. Springer LNCS, Sep. 2004, pp. 120–131.

[16] ——, "Model-based, multimodal interaction in document browsing," in *Machine Learning for Multimodal Interaction, LNCS Volume 4299*. Springer, 2006, pp. 1–12.

[17] ——, "Control centric approach in designing scrolling and zooming user interfaces," *International Journal of Human-Computer Studies*, vol. 66, no. 12, pp. 838–856, 2008.

[18] S. Kratz, I. Brodien, and M. Rohs, "Semi-automatic zooming for mobile map navigation," in *Proc. of the 12th Int. Conf. on Human Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '10. New York, NY, USA: ACM, 2010, pp. 63–72.

[19] P. Quinn and A. Malacria, S.and Cockburn, "Touch scrolling transfer functions," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '13. New York, NY, USA: ACM, 2013, pp. 61–70.

[20] J. Williamson, R. Murray-Smith, and S. Hughes, "Shoogle: Excitatory multimodal interaction on mobile devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 121–124.

[21] S.-J. Cho, R. Murray-Smith, and Y.-B. Kim, "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*, ser. MobileHCI '07. New York, NY, USA: ACM, 2007, pp. 190–197.

[22] P. Quinn and S. Zhai, "Modeling gesture-typing movements," *Human–Computer Interaction*, 2016, accepted author version posted online.

[23] J. Williamson and R. Murray-Smith, "Pointing without a pointer," in *roceedings of ACM SIG CHI 2004 Conference on Human Factors in Computing Systems*. ACM, 2004, pp. 1407–1410.

[24] J.-D. Fekete, N. Elmqvist, and Y. Guiard, "Motion-pointing: Target selection using elliptical motions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 289–298.

[25] H. Pohl and R. Murray-Smith, "Focused and casual interactions: Allowing users to vary their level of engagement," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. New York, NY, USA: ACM, 2013, pp. 2223–2232.

[26] H. Pohl, "Casual Interaction: Devices and Techniques for Low-Engagement Interaction," Ph.D. dissertation, Univ. Hannover, 2017.

[27] D. A. Norman and D. G. Bobrow, "On data-limited and resource-limited processes," *Cognitive Psychology*, vol. 7, no. 1, pp. 44–64, 1975.

[28] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[29] Y. Le Cun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[31] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[32] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "[design and analysis of computer experiments]: Rejoinder," *Statistical Science*, pp. 433–435, 1989.

[33] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pp. 425–464, 2001.

[34] S. Conti, J. P. Gosling, J. E. Oakley, and A. O'Hagan, "Gaussian process emulation of dynamic computer codes," *Biometrika*, p. asp028, 2009.

[35] J. H. Williamson and R. Murray-Smith, "FingerFlow: Cursorless interaction for hover touch," *Submitted for publication*, 2017.

[36] B. Vad, D. Boland, J. Williamson, R. Murray-smith, P. B. Steffensen, and A. S. Syntonetic, "Design and Evaluation of a Probabilistic Music Projection Interface," in *16th Int. Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 134–140.

[37] D. Boland, R. McLachlan, and R. Murray-Smith, "Engaging with Mobile Music Retrieval," in *Proc. 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '15. ACM, 2015, pp. 484–493.

[38] D. Boland, "Engaging with music retrieval," Ph.D. dissertation, University of Glasgow, 2015.

[39] C. F. Higham, R. Murray-Smith, M. J. Padgett, and M. P. Edgar, "Deep Learning for Single-Pixel Video," *Submitted for publication*, 2017.

[40] O. Flemisch, A. Adams, S. R. Conway, K. H. Goodrich, M. T. Palmer, and P. C. Schutte, "The H-Metaphor as a guideline for vehicle automation and interaction," Tech. Rep. NASA/TM?2003-212672, 2003.

[41] G. Müller-Putz, P. Ofner, A. Schwarz, J. Pereira, G. Luzhnica, C. d. Sciascio, E. Veas, S. Stein, J. Williamson, R. Murray-Smith, C. Escolano, L. Montesano, B. Hessing, M. Schneiders, and R. Rupp, "MoreGrasp: Restoration of upper limb function in individuals with high spinal cord injury by multimodal neuroprostheses in daily activities," in *7th Graz BCI Conference*, 2017.