Performance Modelling of Interface Dynamism in Motor Skills Acquisition

Olurotimi R. Akinlofa (o.r.akinlofa@rgu.ac.uk)
Patrik O'Brian Holt (p.holt@rgu.ac.uk)
Eyad Elyan (e.elyan@rgu.ac.uk)
IDEAS Research Institute, Robert Gordon University
St Andrew Street, Aberdeen, AB25 1HG, UK

Keywords: Skill Acquisition, Learning, Interface Design.

Introduction

Instructions are central to many human skill acquisition processes e.g. like those used for pilot training (Dennis & Harry, 1998). Recent advances in computing technologies have expanded the scope of computer based instructional delivery especially where safety and cost may preclude the use of traditional training systems. The effectiveness of such CBT systems has been subjected to considerable research (Höffler & Leutner, 2007). An important aspect of research into the effectiveness of such instructional methods, which is relevant to the study reported here, is the benefit of dynamic over static components of instructional interfaces used in the acquisition of procedural motor skills as typical in aviation engineering training simulators.

Akinlofa, Holt, and Elyan, (under revision), propose a model (Figure. 1) to explain the observed benefit of dynamic visualisations compared with statics for learning novel procedural motor skills by aviation engineering trainees. Following on, a representative sub-step from the study is modelled using the ACT-R 6.0 architecture to examine the post-learning task performances of the different learner groups.

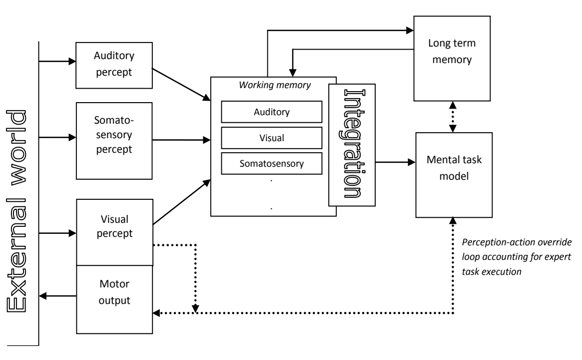

Figure 1: Information-processing model for learning a procedural motor task

Cognitive Modelling

It is assumed that different declarative knowledge structures, dependent on the instructional formats, are created for the sequence of component spatial states in a rotation movement. Static diagrammatic instructions can only afford the initial and final states of the rotated component while dynamic, video instructions will provide knowledge of the start and end states as well as all transitory states in between. Subsequent motor performance is driven by a sequential retrieval of the states, interspersed with a random strategy if retrieval fails. Figure 2 shows that the representative model for the static condition is constrained to a random strategy while the dynamic model utilises a mixed strategy.

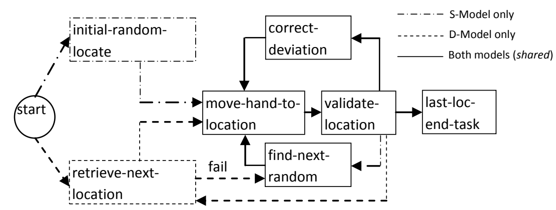

Figure 2: Schematic outline of model's productions

Motor performance for the models is implemented as a chained sequence of unit movement vectors simulating the transition of a selected reference point of the rotated component in 2-D space. The number of unit movement vectors in the movement sequence as well as their individual directions is stochastically dependent on the current position in the trajectory and the selected productions firing per cycle of cognitive processing.

The default mechanism of the motor module of ACT-R 6.0 was not suitable for our model design as it utilises Fitts's law to calculate movement execution time towards a specified target. Additionally, it calculates incremental positions along a movement path for specified start and end positions only. Our model's movement strategy however specifies only a start position, while the end position is stochastically determined by a fixed magnitude, variable direction, unit movement vector. Furthermore, as movement is implemented by a sequence of unit vectors, the transitory point from vector to vector must be modelled accurately to ensure uniform and continuous acceleration throughout the trajectory. Therefore, the default motor module is redefined through an adaptation of the dynamic cost optimisation approach for the mathematical modelling of human hand movements (Flash & Hogan, 1985). By using the minimisation of the time integral of the square of jerk on the curved component rotation trajectory, point-to-point movement is represented by the insertion of intermediate points (at times $t_1, t_2, \ldots, t_n$) between the start and end positions. The entire trajectory is then modelled through a shifting boundary condition method across the range of via points bounded by t=0 and t=t_f according to the following equations:

for all times $t \leq t_n$

$$x^-(\tau) = \frac{t_f^5}{720}(\mu_x(\tau_n^4(15\tau^4 - 30\tau^3) + \tau_n^3(80\tau^3 - 30\tau^4) \\ -60\tau^3\tau_n^2 + 30\tau^4\tau_n - 6\tau^5) + c_x(15\tau^4 - 10\tau^3 - 6\tau^5)) + x_0$$

$$y^-(\tau) = \frac{t_f^5}{720}(\mu_y(\tau_n^4(15\tau^4 - 30\tau^3) + \tau_n^3(80\tau^3 - 30\tau^4) \\ -60\tau^3\tau_n^2 + 30\tau^4\tau_n - 6\tau^5) + c_y(15\tau^4 - 10\tau^3 - 6\tau^5)) + y_0$$

and for all times $t \geq t_n$

$$x^+(\tau) = \frac{t_f^5}{720}(\mu_x(\tau_n^4(15\tau^4 - 30\tau^3 + 30\tau - 15) \\ +\tau_n^3(-30\tau^4 + 80\tau^3 - 60\tau^2 + 10)) + c_y(-6\tau^5 + 15\tau^4 - 10\tau^3 + 1)) + x_f \\ = x^-(\tau) + \frac{\mu_x t_f^5(\tau-\tau_n)^5}{120}$$

$$y^+(\tau) = \frac{t_f^5}{720}(\mu_y(\tau_n^4(15\tau^4 - 30\tau^3 + 30\tau - 15) \\ +\tau_n^3(-30\tau^4 + 80\tau^3 - 60\tau^2 + 10)) + c_y(-6\tau^5 + 15\tau^4 - 10\tau^3 + 1)) + y_f \\ = y^-(\tau) + \frac{\mu_y t_f^5(\tau-\tau_n)^5}{120}$$

where $\tau = t/t_f$; $\tau_n = t_n/t_f$; t_n is a via-point; $\mu_x, \mu_y, c_x, and\ c_y$ are constants.

This affords accurate and continuous implementation of hand acceleration through the transition points in the movement vector sequence. The partial matching mechanism of the retrieval module is further utilised to simulate the inaccuracy of recalling component intermediate positions along the trajectory of rotation. As the model movement is implemented in 2-D Cartesian space, a sim-hook function is used to define matching inaccuracies on the x-coordinates. Additionally, an extension of the activation equation is used to define matching on the y-coordinate and a summation of the matching functions outputs is computed as the overall match score of a specific location in the movement space. This design, as depicted in Figure 3, is very flexible and could be a starting point for extending to 3-D spatial movement.

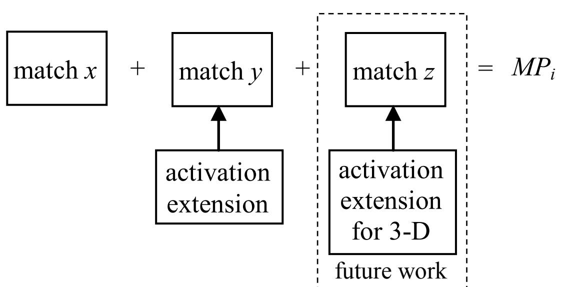

Figure 3: Spatial location partial matching

Results and Further Work

Comparison with human data shows that the model's quantitative predictions were accurate on latencies (R^2=.98, *RMSE=.52*) and trajectory tracking. This result supports an advantage of dynamism in the instructional interface for procedural skill acquisition. It further supports our hypothesis on the hybrid role of cognition in procedural motor performance and validates the assumed strategies of task execution of initially attempting to recall a mental task image and resorting to a controlled stochastic method in the event of a retrieval failure. However, there are other factors, such as the cognitive abilities of the learners, which were controlled in the larger empirical study in contrast with the implementation of the computational models. Our result is further limited because the models reflect a single step of a procedural movement and implements the movement in 2-D space only. Further collaborative work will extend the model to cover the entire movement sequence of the experimental task reported in Watson, Butterfield, Curran, and Craig (2010). Our future work will also explore the execution of the task with 3-D spatial movement using the approach of extending the activation equation for spatial matching as outlined above.

Acknowledgments

This study was supported by the Petroleum Technology Development Fund (PTDF), Nigeria Grant: PTDF/TR/PhD/AOR/206/09.

References

Akinlofa, O.R., Holt, P.O., & Elyan, E. (under revision). *Effect of interface dynamism on learning procedural motor skills*.

Dennis, K.A., & Harry, D. (1998). Computer-Based Simulation as an adjunct to ab-initio flight training. *The International Journal of Aviation Psychology, 8,* 261–276.

Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience 5,* 1688–1703.

Höffler, T., & Leutner, D. (2007). Instructional animation versus static pictures: A meta-analysis. *Learning and Instruction 17,* 722–738.

Watson, G., Butterfied, J., Curran, R., & Craig, C. (2010). Do dynamic work instructions provide an advantage over static instructions in a small scale assembly task?. *Learning and Instruction 20,* 84-93.