IFAC Man-Machine Systems Conf., Kyoto, Japan, 1998.

MODELLING HUMAN CONTROL BEHAVIOUR WITH CONTEXT-DEPENDENT MARKOV-SWITCHING MULTIPLE MODELS

Roderick Murray-Smith

Dept. of Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark. rod@imm.dtu.dk

Abstract: A probabilistic model of human control behaviour is described. It assumes that human behaviour can be represented by switching among a number of relatively simple behaviours. The model structure is closely related to the Hidden Markov Models (HMMs) commonly used for speech recognition. An HMM with context-dependent transition functions switching between linear control laws is identified from experimental data. The applicability of the approach is demonstrated in a pitch control task for a simplified helicopter model.

Keywords: Multiple model systems, Hidden Markov Models, human–machine interface, learning algorithms, Human–machine interaction, flight control, simulation.

1. INTRODUCTION

If the intentions, goals and preferences, (the human ‘state’) and the accompanying skills of the human operator were known, the human–machine interaction problem would be to coordinate, adapt, and configure the automatic control system to ensure satisfactory performance of the full human–machine control system. Unfortunately, the states of the human are usually not known – at best they can be estimated. This paper describes an approach to the simultaneous estimation of human states and behaviour models – recognising operators’ goals and ‘modes’ of behaviour from their actions.

The goal of this work is to develop approaches which could be used to estimate and predict operator skills, such that we would be able to learn individual preferences and expectations, and detect characteristic features and types of operators. In systems design, actual performance, workload and performance limitations for a given task could be better understood before construction of a prototype. Because of the complexity of human behaviour, and the richness of sensing and state, no conceivable model will be able to predict exactly what the human will do. In this paper we will use a probabilistic framework for the representation of human control behaviour, as this provides a common framework for describing the uncertainty in both the human and technical aspects of our system and allows us to develop models which for the given task behave statistically as a human would.

The need for human control models in systems design has long been known, but it was often impossible to identify and represent the complexity of human behaviour in a particular task at a reasonable cost. Improvements in computing power and learning algorithms have now made it feasible to implement complex operator models that can learn and represent high-level aspects of behaviour such as tasks and goals, as well as being able to discriminate between different human operators and various levels of operator performance and preferences.

Most classical approaches to modelling human manual control behaviour (quasi-linear, optimal control and sampled data models) are mainly applicable to low-level manual control tasks involving skilled operators. More complex tasks, higher-level information processing and inexperienced operators are typically not covered by such models. The more flexible model described in this paper assumes the operator is in one of a finite number of human ‘states’. Each of these

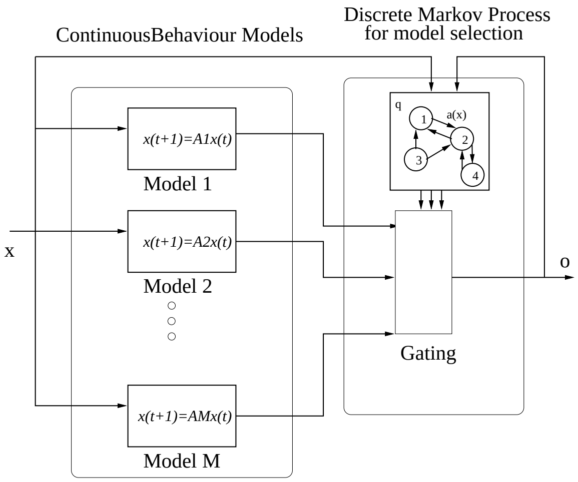

Fig. 1. The model structure – a discrete Markov process switches between the continuous state processes. The discrete process transitions are dependnet on the continuous state.

hypotheses has an associated behaviour which can be described in terms of a probability model. The learning algorithms used allow us to identify both the parameters of the individual behaviour models, and the switching functions simultaneously. We thus have a standard probabilistic framework for the interpretation of a time series of human action.

1.1 *Multiple-model representations of human control behaviour*

The multiple-model interpretation of human control action, instead of having a single complex model, prefers to describe control action by switching between a number of simple behaviours (see Johansen and Murray-Smith (1997) for a review of the multiple model approach to modelling). In experiments, it becomes clear that human control action often goes through rapid changes of behaviour, due to, for example, changes in the human's perception of the situation, goal changes, attention lapse, or change in the effective dynamics of the controlled system. [1]

This paper examines a model (as shown in Figure 1), which switches between a number of linear controllers. The transitions between models are instant, and do not involve blending of behaviours. The model switching is probabilistic but conditioned on the state/input vector, so it supports a spectrum of models from purely stochastic switching to purely deterministic switching, depending on its parameterisation. See Meila and Jordan (1997) and Bengio and Frasconi (1996) for further details. This gating or switching element can be viewed as a pattern recognition system which chooses the next model state given the 'pattern' of the current continuous state-vector. Its probabilistic nature takes into account both variations in human behaviour and measurement errors. It uses many of the tools common to speech recognition technology, and much of the framework is taken from the excellent review article by Rabiner (1989).

Related approaches to modelling human actions exist, but usually use discrete actions, not the mixture of continuous and discontinuous control used in this paper, and constant transition probabilities as opposed to the state-dependent transition functions used here. Yang et al. (1994) applied HMM's to learn human skills, and continued the work in Nechyba and Xu (1998). Pentland and Liu (1995) outline possible applications of HMM's to modelling driver control behaviour and prediction of immediate intentions.

2. MODEL STRUCTURE, INFERENCE AND LEARNING ALGORITHM

2.1 *Model structure*

We have a model $\mathcal{M} = \{A_i\,(x; w_i)\,, f_i\,(x; \theta_i)\,, \Sigma_i, \pi\}$, $i = 1, .., N_m$, with an observable continuous state x and a hidden discrete state q, which can be in one of N_m states. $A_i(x; w_i)$ is the state-transition matrix from discrete state q, dependent on continuous state x, the entries of which $a_{ij} = P(q_{t+1} = j|q_t = 1)$ are the transition probabilities at time t. This is effectively a pattern recognition system, mapping regions of the state-space to a transition probability distribution. In this paper, the transition probabilities are represented by a multinomial logit (or 'softmax') function (equation (6)), with parameters w.

The N_m submodels $b(o_t, i)$ define the emission probabilities – i.e. the probability of observing o_t given discrete state $q_t = i$, and continuous state x. In the continuous control action case examined here, one could use a mixture model density function, but in this paper we will only use a single component in each mixture, i.e. a linear model with Gaussian noise (mean $\mu_i = f_i\,(x; \theta_i) = \theta_i x$, variance Σ_i^2). The estimate of the initial discrete state distribution $\pi = \{\pi_i\}$, where $\pi_i = P(q_1 = i)$.

2.2 *Inference*

As reviewed in Rabiner (1989), there are several inference problems:

(1) *Evaluation:* Probability of model $\mathcal{M}$ given observed output time series $O = \{o_1, o_2, ..., o_T\}$?
(2) *Decoding:* Probability of hidden state i at time t given observed time series O?

[1] These ideas are not new; (Sheridan and Ferrell, 1974, Chapter 15) gives an interesting review of the reasons for using intermittent representations, but points out the disadvantage that at that time, such models were created by tedious 'cut and try' methods – now, however, available algorithms relieve the human of extensive parameter tuning.

(3) *Estimation:* What are the parameters most likely to have generated the output time series?

2.2.1. *Evaluation* To evaluate how well the model $\mathcal{M}$ matches the time series O we need to evaluate $P(O|\mathcal{M})$, which involves finding the probability of all possible paths through the hidden states, $P(O|\mathcal{M}) = \sum_{allQ} P(O|Q)P(Q|\mathcal{M})$, but this quickly becomes intractable, so to perform the calculations efficiently we use the standard Baum-Welch forward-backward procedure. Define $\alpha_t(i) = P(O_{1,t}, q_t = i|\mathcal{M})$ as the probability of a partial sequence, which allows us to recursively generate the probability of the whole sequence,

$$\alpha_1(i) = \pi_i b(u_1, i),$$
$$\alpha_{t+1}(j) = b(o_{t+1}, j) \sum_{i=1}^{N_m} \alpha_t(i) a_{ij},$$

giving us the probability of the whole sequence O, $P(O|\mathcal{M}) = \sum_{i=1}^{N_m} \alpha_T(i)$. There is an accompanying backward phase, which will be useful for the decoding step in the next section. Here we define $\beta_t(i) = P(O_{t+1,T}|q_T = i, \mathcal{M})$ and we now have a backward recursion:

$$\beta_T(i) = 1, \beta_t(i) = \sum_{j=1}^{N_m} \beta_{t+1}(i) b(o_{t+1}, j) a_{ij},$$

such that $P(O|\mathcal{M}) = \sum_{i=1}^{N_m} \pi_i b(o_1, i)\beta_1(i)$. [2]

2.2.2. *Decoding – which state are we in?* We wish to find out at each point in time the probability of the various behaviour hypotheses. [3] In other words, estimate $\gamma_t(i) = P(q_t = i|O, \mathcal{M})$. This can be expressed in terms of the forward-backward variables found in the previous section:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N_m} \alpha_t(j)\beta_t(j)}, \tag{1}$$

which is a probability measure which sums to one over all behaviours.

2.2.3. *Estimation* Given the ability to evaluate model quality and algorithms for decoding the hidden states, we can move on to the more difficult problem of parameter estimation. This problem cannot be solved analytically and there is no easy way of finding the optimal global solution, but we shall use the standard EM approach to local maximisation of $P(O|\mathcal{M})$.

[2] To avoid implementation problems, we normalise the α's and β's such that $\alpha_t(j) = \frac{1}{N_t}\alpha_t(j)$, where the normalising constant $N_t = \sum_{i=1}^{N_m} \alpha_t(i)$. The β's are also scaled using the same N_t,$\beta_t(j) = \frac{1}{N_t}\beta_t(j)$.

[3] If we knew this, our problems would be trivial – we are thus viewing these variables as 'missing data' which have to be estimated.

We define $\xi_t(i,j)$ to be the probability of switching from state i at time t to state j at time $t+1$,

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|O, \mathcal{M}), \tag{2}$$

which can again use the results from the forward-backward stage, such that

$$\xi_t(i,j) = \frac{\alpha_t(i)a(x)_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{P(O|\mathcal{M})}. \tag{3}$$

By summing $\gamma_t(i)$ for $t_{1,T}$ we have the expected number of samples where behaviour i was active, and similarly by summing $\xi_t(i,j)$ for $t_{1,T}$ we have the expected number of transitions from behaviour i to behaviour j. This leads us to reestimation formulae for the parameters to maximise likelihood:

Local Models The estimation stage for the local model parameters reduces to weighted linear optimisation, where the weighting function for each data point is provided by $\gamma_t(i)$, the probability of model i generating o_t. In the case of a single linear model,

$$\mu_i = \arg\min_{\theta_i} \sum_{t=1}^{T} \gamma_t(i)||o_t - f_i(x, \theta_i)||^2 \tag{4}$$

$$\Sigma_i^2 = \frac{\sum_{t=1}^{T} \gamma_i||o_t - f_i(x, \theta_i)||^2}{n_o \sum_{t=1}^{t=T} \gamma_t(i)}, \tag{5}$$

where n_o is the dimension of the observation vector o_t.

Transition functions As $\xi_t(i,j)$ represents the current estimate of the probability of changing from state i to j at time t, we use this as a target for the transition function (normalised by the probability of being in state i at time t). In this example we use a simple 'softmax' representation of the transition function,

$$a_{ij}(x) = \frac{\exp(w_{ij}x)}{\sum_k \exp(w_{ik}x)}, \tag{6}$$

but the same approach is valid for more complex networks or other representations, such as belief networks (see Jensen (1996) for background). [4]

To maximise the likelihood of the model the parameters of the transition functions w are adapted in N_m independent optimisation problems using a conjugate-gradient algorithm to bring $a_{ij}(x, w) \approx \frac{\xi_t(i,j)}{\gamma_t(j)}$. As we cannot guarantee that global optima are found, we are effectively performing a Generalised EM step. See Meila and Jordan (1997) for further discussion.

[4] Note that in real applications we would often use different x state-vectors for transitions from different states, and certain transitions could be excluded from the model structure in advance. The x for transition functions may also be transformed in some way.

2.3 *Role of inference in modelling human behaviour*

The algorithms for inference described above have very concrete uses in a human modelling application. We take the continuous state x to be the input/state information the human bases his or her control on. That control action u is the observable sequence O referred to in the previous section.

2.3.1. *Evaluation for Classification* There are many applications where we would wish to classify a series of human control actions. The class chosen could be to estimate which of a number of known individual users performed them (possibly for security or insurance purposes), or to compare the behaviour to a number of *types* of user (e.g. beginner, average user, expert, tired performance). This could be useful in training operators in simulated environments, when classifiers which quantify the style of behaviour could be used to guide and document the results of a training programme. A further example is to differentiate between types of behaviour of a given human operator (e.g. normal behaviour, tired or inattentive behaviour, aggressive behaviour). The approach used is to collect data O_i for each class of control action, and estimate accompanying models $\mathcal{M}_i$. We then select the model with the maximum $P(O|\mathcal{M}_i)$. This is not an explicitly discriminative approach, and if classification is the ultimate aim of modelling, it may be worth using discriminative approaches.

2.3.2. *Decoding for segmentation of the time-series* The use of standard inference with the model, and EM to iteratively optimise the parameters, automatically gives us a segmentation of the human control time-series into sub-behaviours. The γ_t's provide us with the probability that the human was in the given state at time t. This is an attempt to infer human 'intentions' or 'sub-goals'. This information can then be used to improve the interaction in a human–machine control system, and can provide context information to human–machine interfaces. Multiple-Model Adaptive Control (MMAC) is an analogue method used in control applications, e.g. Schott and Bequette (1997).

2.3.3. *Estimation for modelling* Given the segmentation of the data provided by the γ_t's, the estimation stage provides us with local models corresponding to system behaviour in each hypothesis. Again, this information, with the γ's can be used to improve co-operation in a human–machine system – we have an estimate of the human's 'hidden state', which can be viewed as current intentions or a current mode of behaviour. We also have how the human usually behaves in this state – the local model associated with that state.

Fig. 2. The screen display used in the experiment. The human operator has to track the reference horizontal line with his/her own pitch indicator (double line)

3. PITCH CONTROL EXAMPLE

The modelling task used to illustrate the approach is that of pitch control in a simplified helicopter model, the 'flying brick' (see Bradley (1996)), which is basically a point mass steered by a force acting at a distance – a crude representation of a rotor disc. No other aerodynamic forces are included. This paper used the model in pitch control mode, where roll and yaw are always zero. In this experiment no attempt was made to control velocities, or position – only pitch was important. The relevant equations of motion for the pitch angle θ, given a control input u are thus:

$$\dot{\theta} = \frac{-l_h m g(1+\theta)\sin(u)}{I_{xx}},$$

where $l_h = 1.454$m, $m = 4313$kg, $g = 9.81\text{ms}^{-2}$, $I_{xx} = 2767\text{kgm}^2$.

3.1 *Experiment design*

The task was to track a changing pitch reference value, as indicated on screen (see Figure 2). The system was implemented on a standard PC and monitor. The sampling time was 0.05s which was then resampled at 0.1s for use in modelling. For actuation we used a centre-sprung games joystick with an 8 bit accuracy (CH products flightstick pro).

As in any empirical modelling task it is important to provide sufficient excitation in the experiment to be able to identify the parameters of interest, and to avoid numerical problems. The target trajectory used included large occasional random step changes, followed by frequent small step changes,and occasional ramp-like changes, as shown in Figure 4. This allows us to study the reaction to step changes of different sizes, and tracking a moving target. Note that these have no 'preview' effect, i.e. the human does not know what the reference signal is about to do, and cannot use feedforward control.

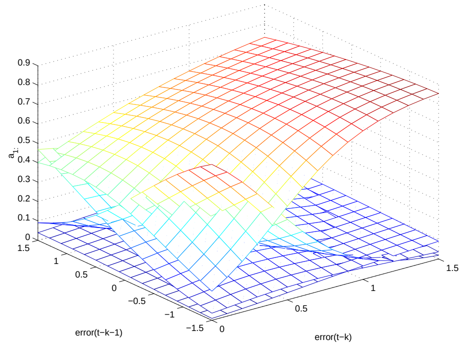

Fig. 3. A state-dependent transition function from a single state to 4 other states. Each curve indicates the probability of transition from this state for varying x. At any x the curves sum to one.

3.2 *Modelling results*

We identified simple multiple-model systems, with 3 models where each model was a second order discrete-time controller, with a pure k-delay element $u_t = \theta_i[e_{t-k}e_{t-k-1}u_{t-1}u_{t-2}]$. The transition functions were softmax functions scheduled on k-step delayed values of $\dot{e}$ and $||e||$ (e.g. see Figure 3).

3.2.1. *Segmentation of time series* An immediate question is whether we can recognise a 'sensible' segmentation of the time series from Figure 4 by plotting the γ's, as in Figure 5. Initially, $\gamma_t(i) \approx 1/N_m$, but as learning progresses we see the segmentation improve, and find a correlation between larger errors and probability of different hypotheses. Note that successful decoding does not necessarily mean that the model has learned the transition functions adequately.

3.2.2. *Parameter estimates* The parameters should look 'plausible' given available prior knowledge of the problem. This is relevant for both the local models and the transition functions. We could clearly see 'surge-like' behaviour (as discussed in Sheridan and Ferrell (1974)) with low gain models around small error regimes and high gain parameters in high error regimes. In some runs, one of the models would occasionally take on negative gains, corresponding to moments when the human went the 'wrong' way after a large change in reference value. In this example, the transition functions from each of the models were quite similar.

3.2.3. *Simulation results* Simulation of model behaviour in a closed-loop with the controlled system is probably the most interesting test of behaviour. We would hope to find typical features observed in the human reproduced in the simulation. Figure 6 shows such a test.

3.2.4. *Classification results* The classification experiment, involved 4 models trained on 4 different humans (all male researchers, new to the task), with 2 runs of 90s each, and tested on 3 new runs of 90s. The model with the largest $P(O|\mathcal{M})$ was selected for each run. Classification was 100% accurate on training runs, and for classification on new data 10 from 12 runs were classified correctly.

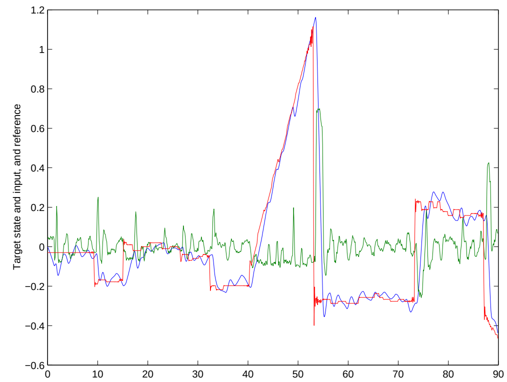

Fig. 4. A typical time-series recording from an experiment. The human control input indicates strongly that some form of intermittent model is needed.

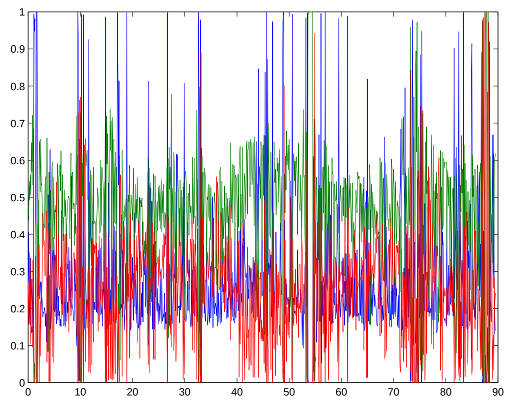

Fig. 5. The γ's segment the time-series – here we see some models are a more likely explanation of the data given large errors, while others compete in lower error regimes.

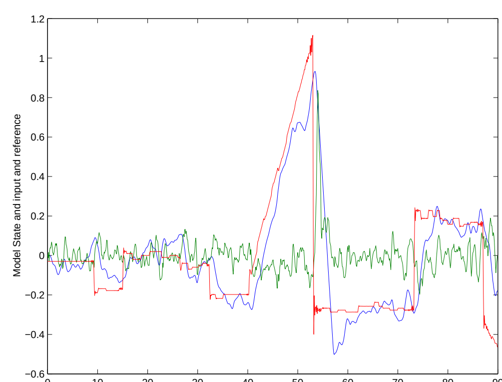

Fig. 6. The model's behaviour on the time-series shown in Figure 4. Note that the model should not be exactly the same as the human, but should be qualititatively similar. This run shows mismatch in the noise dynamics which a longer-tailed distribution would fit better.

3.3 *Relating model structure to a given application*

This example was not particularly complex, but selecting the granularity (how detailed the individual behaviour models should be) of the model is often a far from a trivial matter. In principle the engineer will examine the task, estimate the number of behaviours believed to be applicable, and the transition probabilities between them (if we can rule out certain transitions from the start, the learning task is eased significantly). [5] There are a number of sources of uncertainty in this model: Individual behaviours will vary, and the transitions between behaviours will also vary. We are trying to absorb much of the complexity of the model into the switching component. The context-dependent transition functions potentially provide us with arbitrarily powerful representations of transition uncertainty, conditioned on the measurable states.

4. CONCLUSIONS

The multiple model framework was introduced as a potentially powerful approach to modelling human control behaviour. The framework identifies a model, and estimates the current human 'state', and can be used to better coordinate human and machine control behaviour. It was illustrated in a simple pitch tracking task. The methods used were able to identify representative and meaningful models from experimental data, and were able to classify which human generated a given experimental behaviour, in a manner similar to speech recognition systems. For simplicity, these experiments examined a low-level manual control task, but we believe that the approach has more relative potential as a model of higher-level control behaviours and multivariable problems.

The models created are *generative* models. Even if we only wish to produce classifiers which recognise a given behaviour or human state, the use of generative models tends to make the approach less susceptible to minor changes in system configuration than *feature-based* classification methods would be, as well as allowing a more principled approach to the engineering task.

The modular nature of the approach means that the basic model can be extended incrementally to improve sub-models representing given behaviours, or to provide a more sophisticated behaviour switching logic. Bayesian networks could, for example, be used as a representation of the transition probabilities, or for individual behaviours. In fact, we could use a range of different approaches (e.g. fuzzy, classical control, Bayesian networks, neural networks) in a single model, given that they can be interpreted as providing a probabilistic output. We thus have the potential to integrate the uncertainty related to 'hard' engineering aspects with those of the 'soft' human aspects within a single framework.

[5] Note that the behaviours chosen need not be purely 'sensible' control actions, but can also include 'noise' behaviours which can be switched in rapidly, as well as 'human error' behaviours which are characteristic behaviours, but do not fulfill the human's stated objectives.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the support of Marie Curie TMR grant FMBICT961369. Daimler-Benz research supported earlier stages of this work, and Marina Meila and Mike Jordan at M.I.T. introduced me to many of the techniques. The simulator was developed by David Murray-Smith and Graham Dudgeon at Glasgow University.

References

Y. Bengio and P. Frasconi. Input–Output HMM's for sequence processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1248, 1996.

Roy Bradley. The flying brick exposed: Non-linear control of a basic helicopter model. Technical report, Dept. of Mathematics, Glasgow Caledonian University, 1996.

Finn V. Jensen. *An introduction to Bayesian Networks.* UCL Press, London, 1996.

T. A. Johansen and R. Murray-Smith. The operating regime approach to nonlinear modelling and control. In R. Murray-Smith and T. A. Johansen (Eds.), editors, *Multiple Model Approaches to Modelling and Control*, chapter 1, pages 3–72. Taylor and Francis, London, 1997.

M. Meila and M. I. Jordan. Markov mixtures of experts. In R. Murray-Smith and T. A. Johansen, editors, *Multiple Model Approaches to Modelling and Control*, chapter 5, pages 145–166. Taylor and Francis, London, 1997.

Michael Nechyba and Yangsheng Xu. On discontinuous human control strategies. Submitted to Proc. IEEE Int. Conf. on Robotics and Automation, 1998.

Alex Pentland and Andrew Liu. Toward augmented control systems. In *Intelligent Vehicles Symposium*, Detroit, 1995.

R. L. Rabiner. A tutorial on Hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

K. D. Schott and B. W. Bequette. Multiple model adaptive control. In R. Murray-Smith and T. A. Johansen, editors, *Multiple Model Approaches to Modelling and Control*, chapter 11, pages 269–292. Taylor and Francis, London, 1997.

T. B. Sheridan and W. R. Ferrell. *Man-Machine Systems*. MIT Press, 1974.

Jie Yang, Yangsheng Xu, and Chiou S. Chen. Hidden Markov Model approach to skill learning and its application to telerobotics. *IEEE Transactions on Robotics and Automation*, 10(5), 1994.