In Proceedings of Control'97, Sydney, 20-22 October 1997

Sensitivity Analysis of a Sensory Perception Controller

G.E. Hovland and B.J. McCarragher
Department of Engineering
The Faculty of Engineering and Information Technology
The Australian National University
Canberra, Australia
e-mail: { geir, brenan }@faceng.edu.au

Abstract

We have developed a method for the sensitivity analysis of a sensory perception controller (SPC). The SPC performs dynamic process monitor selection in a real-time discrete event control system. The SPC uses stochastic dynamic programming (SDP) to solve the underlying Markov decision process. The SDP approach has been mapped to a linear programming (LP) formulation and the sensitivity analysis is performed using well-known LP techniques. In particular, the sensitivity of the discount factor of future rewards is studied. We show that the SPC has a relatively low sensitivity to variations in this model parameter.

1 Introduction

The control of sensory perception allows for an efficient use of available sensors. In nominal operation, only a few sensing techniques (monitors) are required. Then as an anomaly develops, additional sensing techniques are utilised. The average sensing costs for a system using perception control are lower than for a system where all sensors are used all the time. Moreover, the reliability of the sensory information is higher than for single sensor systems.

Sensory perception controllers are particular useful in industrial real-time control systems, where typical requirements are fast and reliable operation. Control of sensory perception aims at collecting high quality sensory information while keeping the sensory processing time low.

Sensitivity analysis is useful for gaining insight into the performance of any solution methodology. In this paper we study the sensitivity analysis of a sensory perception controller (SPC) in a discrete event framework developed in [3, 4]. The analysis is required because of uncertainties in model parameters such as state rewards and monitor rewards. The sensory perception problem is a discounted Markov decision process. The SPC solves the decision process by using stochastic dynamic programming (SDP). In this paper we show that linear programming (LP) is an alternative solution to the sensory perception problem. We study the effects on the SPC decisions with uncertainties in the discount factor α. In particular, we are interested in answering the following type of questions. If α is doubled, will this cause the SPC to consult another monitor first? The answers to these questions help us to tune model parameters and achieve robust controllers.

In [1] a direct approach to sensitivity analysis of nonlinear programming is presented. SDP is a special area of nonlinear programming, but the theory in [1] applies to static deterministic problems only. A direct sensitivity analysis of the SDP solution of the SPC would be desirable. However, the direct approach is difficult and few results exist in the literature. Under these circumstances the mapping of SDP to LP provides a powerful alternative. The LP solution is not as efficient as the SDP solution, but LP has a mature theory for sensitivity analysis, see for example [2, 10, 12, 13].

An example of sensitivity analysis using linear programming is found in [14]. The objective of the analysis is to tune some control parameters such that the system damping is improved, which is achieved by studying the sensitivities of the closed-loop system eigenvalues. In contrast to the work presented in our paper, the sensitivity analysis problem is itself formulated as a linear program. In our work we study the sensitivity of the linear program parameters.

2 Sensory Perception Controller

The SPC considered in this paper was developed in [3] and is described in Figure 1. The aim of the SPC is to reliably recognise discrete events as they occur while keeping the processing time low. The different event monitoring techniques have different characteristics and the sensory perception is based on real-time confidence level measures and the CPU time spent by each monitor. Every sequential ordering of the monitors is evaluated by a dynamic programming algorithm.

The DP algorithm is run off-line and the optimal DP actions and value functions are stored for each sequential ordering. The stored actions and value functions are used in real-time by the SPC. The highest scoring sequential ordering is chosen and the SPC has the option of terminating or continuing the information gathering. When the SPC has completed its analysis, a recognised discrete event e^* is sent to a discrete event controller (DEC). The DEC will then use new reference commands and parameters of the continuous controller until a new discrete event occurs and the sensory perception control problem repeats.

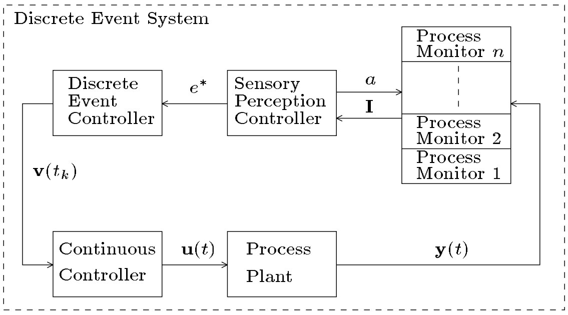

Figure 1: *Discrete event system with dynamic sensory perception capabilities.* $\mathbf{v}(k)$ *is a vector of discrete controller commands occuring at time* k, $\mathbf{u}(t)$ *and* $\mathbf{y}(t)$ *are continuous plant inputs and outputs,* $\mathbf{I}$ *is a vector of symbolic process plant information,* a *is a command (or action) from the sensory perception controller and* $e^*(k)$ *is the final output from the SPC.*

The control of sensory perception fits particularly well in the discrete event control framework. Because of the discrete nature of events, the DEC does not require feedback information continuously. Hence, processing time is available for the analysis of sensory information between the occurrence of events.

Traditionally, research dealing with discrete event systems has assumed perfect event identification. For most practical systems these assumptions are unrealistic. The significance of event identification is great; it is the main difficulty with interfacing continuous-time systems with discrete-event controllers. The sensory perception controller increases the applicability of the discrete event theory by reliably identifying discrete events as they occur while keeping the sensing costs low.

3 Mapping Stochastic Dynamic Programming to Linear Programming

The sensory perception controller developed in [3] uses stochastic dynamic programming (SDP) to solve the discounted Markovian decision process. In this section we present an equivalent solution using linear programming (LP). It has been shown, for example [5, 6, 8], that solutions to Markov decision processes can be found using linear programming. Although the LP algorithm is not as efficient as the SDP algorithm, we study LP because of its facility for sensitivity analysis.

First, in section 3.1 we present briefly the SDP formulation and then in section 3.2 we present the equivalent LP formulation. Finally, the LP is presented in standard form in section 3.3.

3.1 Stochastic Dynamic Programming Formulation

The stochastic dynamic programming equation used in [3] has the following form.

$$V_n(s_i) = \max_{a_k \in \mathcal{A}} \left[R_n(s_i, a_k) + \alpha \sum_{s_j \in \mathcal{S}} P_{ij}(a_k, n) V_{n-1}(s_j) \right] \tag{1}$$

where $R_n(s_i, a_k)$ is the reward at iteration n given state s_i and action a_k. $P_{ij}(a_k, n)$ is the probability of a state transition $s_i \to s_j$ at iteration n given action a_k and $0 \leq \alpha \leq 1$ is a discount factor. The action a_k which maximises the value function above is chosen as the optimal action for each state s_i at each iteration n.

The underlying stochastic model is chosen as in [3] and the first step is to define a state space. Continuous measurements of the process monitor confidence levels are discretised into a discrete state space. In this paper we define the state space as follows.

Definition 1 (State Space) *Let* $\mathcal{S} = \{s_1, \cdots, s_{N_S}, s_\infty\}$ *be a finite discrete state space with* N_S *elements between* 0 *and* 1, *representing the confidence level output from a process monitor, and* s_∞ *is called the terminating state.* □

The Markov decision process is formulated as an optimal stopping problem and hence a terminating state s_∞ is defined, see [9]. For stopping problems, no future reward can be earned once the program enters the terminating state. Hence, the optimal solution is found using a finite number of iterations.

The next step in defining the model, is to define the action space.

Definition 2 (Action Space) *The set of possible actions at any given state is defined as*

$$\mathcal{A} = \left\{ \begin{array}{ll} a_1 & \text{Terminate} \\ a_2 & \text{Continue} \end{array} \right\} \tag{2}$$

□

When the dynamic programming is terminated (action a_1) a discrete event is identified by the sensory perception controller and the information is sent to the discrete event controller, see Figure 1.

In dynamic programming a reward function is associated with every discrete state s_i and action a_k. The reward function is defined as follows.

Definition 3 (Reward Function) *The reward is a function of discrete state and action, ie.*

$$R_n(s_i, a_k) = r(s_i, a_k) + L_n(a_k) \quad s_i \neq s_\infty \quad (3)$$
$$R_n(s_\infty, a_k) = 0 \quad (4)$$

where $r(s_i, a_k) \geq 0$ is the state reward given action a_k. N_m is the number of available process monitors. $L_n(a_k) \geq 0$ is the monitor reward of action a_k at iteration n reflecting factors such as time spent and computational burden invoked by the process monitor. □

A large confidence level of the final decision is desired. Hence, the state rewards for a_1, $r(s_i, a_1) = Kr(s_i)$, are increasing with increasing states $s_i \in [0, 1]$, where the state reward constant $K > 0$ is a scaling factor. The continuation state rewards are zero, ie. $r(s_i, a_2) = 0$. The monitor reward for the termination action is zero, ie. $L_n(a_1) = 0$. Note that the reward in the terminating state s_∞ is zero, which is typical for optimal stopping problems.

The last step in the Markov decision process is to define discrete state transition probabilities. These probabilities will depend on the action taken. We define the transition probabilities as follows.

Definition 4 (Transition Probabilities) *The probability of being in state s_i at iteration n and state s_j at iteration $n+1$ is defined as follows.*

$$P_{ij}(a_k, n) = \begin{cases} 0 & \text{if } a_k = a_1 \text{ and } s_i, s_j \neq s_\infty \\ 1 & \text{if } a_k = a_1,\ s_i \in \mathcal{S} \text{ and } s_j = s_\infty \\ P & \text{if } a_k = a_2 \text{ and } s_i, s_j \neq s_\infty \\ 0 & \text{if } s_i = s_\infty \text{ and } s_j \neq s_\infty \\ 1 & \text{if } s_i = s_j = s_\infty \end{cases} \quad (5)$$

where $P = P_{m_n(z)}(s_j \mid s_i)$. The SDP algorithm evaluates all sequential permutations z of process monitors and $m_n(z)$ is the monitor at sequence number n in z. □ We define a discretisation function as follows.

$$\psi(C) = \text{argmin}_{s_i} |s_i - C| \quad (6)$$

ψ is a mapping from continuous monitor outputs C to the state space $\mathcal{S}$. Let $\mathcal{C} = \{C : s_j = \psi(C)\}$. $\mathcal{C}$ is simply the set of all continuous confidence levels which are mapped to SDP state s_j. We then have

$$P_{m_n(z)}(s_j \mid s_i) = \int_{x \in \mathcal{C}} p_{m_n(z)}(x) dx \quad (7)$$

where $p_{m_n}(x)$ is the probability density function of the confidence levels for monitor $m_n(z)$.

The state space $\mathcal{S}$, the action space $\mathcal{A}$, the reward function $R_n(s_i, a_k)$ and the state transition probabilities $P_{ij}(a_k, n)$ completely describe the Markov decision process.

3.2 Linear Programming Formulation

In order to perform sensitivity analysis, we formulate the decision process described in the previous section as an LP. The LP objective function is given in equation (8) under the constraints (9) and $\lambda_n(s_i, a_k) \geq 0$ for all $n = 0, \cdots, N_m$, $s_i \in \mathcal{S}$ and $a_k \in \mathcal{A}$. N_m is the number of available process monitors.

$$\max \sum_{n=0}^{N_m} \alpha^n \sum_{s_i \in \mathcal{S}} \sum_{a_k \in \mathcal{A}} R_n(s_i, a_k) \lambda_n(s_i, a_k) \quad (8)$$

subject to

$$\sum_{a_k \in \mathcal{A}} \lambda_n(s_j, a_k) = \begin{cases} d_j & \text{for } n = 0, \quad \text{otherwise} \\ \sum_{s_i \in \mathcal{S}} \sum_{a_k \in \mathcal{A}} P_{ij}(a_k, n-1) \lambda_{n-1}(s_i, a_k) \end{cases} \quad (9)$$

Let σ_0 be the initial state in the Markov decision process. Then $d_j = 1$ if $\sigma_0 = s_j$ and $d_j = 0$ if $\sigma_0 \neq s_j$. Note that for the LP problem state variables $\lambda_n(s_i, a_k)$ are needed for each SDP iteration n. Hence, with a large number of SDP iterations, the number of variables and constraints in the LP algorithm becomes large.

The variables $\lambda_n(s_i, a_k)$ are the joint probabilities of being in state $s_i \in \mathcal{S}$ and making decision $a_k \in \mathcal{A}$. There is exactly one LP basic variable $\lambda_n(s_i, a_k)$ for each state s_i and SDP iteration number n. The actions a_k for which $\lambda_n(s_i, a_k)$ are basic variables uniquely determine the sensory perception actions, see [6] for proofs. The standard Simplex algorithm determines the optimal basic variables and hence the optimal actions.

As in [3] all possible sequential orderings of process monitors are considered. Hence, as for the SDP algorithm, the LP algorithm must be solved for each sequential ordering. Given a sequential ordering, the variables $\lambda_0(s_i, a_k)$ describe the joint probabilities for the first monitor, while $\lambda_1(s_i, a_k), \cdots, \lambda_{N_m-1}(s_i, a_k)$ describe the joint probabilities for monitors $1, \cdots, N_m$. The variables $\lambda_{N_m}(s_i, a_k)$ describe the stopping state.

3.3 Linear Program in Standard Form

Linear programming has a mature theory for sensitivity analysis. To be able to use the results in the literature, we must first write the LP developed in the previous section in the following standard form.

$$\begin{array}{ll} \text{Maximise} & \mathbf{cx} \\ \text{subject to} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array} \quad (10)$$

When mapping the SDP algorithm to an LP in standard form, we get the following expressions for $\mathbf{x}$, $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$. The $\mathbf{x}$ vector $(2(N_s+1)N_m \times 1)$ is given by

$$\mathbf{x} = [\lambda_0(s_1, a_1), \cdots, \lambda_0(s_{N_s+1}, a_2),$$

$$\lambda_1(s_1, a_1), \cdots, \lambda_1(s_{N_s+1}, a_2),$$
$$\vdots$$
$$\lambda_{N_m}(s_1, a_1), \cdots, \lambda_{N_m}(s_{N_s+1}, a_2)]^T \quad (11)$$

The $\mathbf{b}$ vector ($(N_s+1)N_m \times 1$) is given by

$$\mathbf{b} = [d_0, \cdots, d_{N_s+1}, 0, \cdots, 0]^T \quad (12)$$

The $\mathbf{A}$ matrix is uniquely defined by the $\mathbf{x}$ vector, the $\mathbf{b}$ vector and equations (9) and (10). The $\mathbf{c}$ vector is found by the objective criterion in equation (8), ie.

$$\begin{aligned}\mathbf{c} = \; & [R_0(s_1, a_1), \cdots, R_0(s_{N_s+1}, a_2), \\ & \alpha R_1(s_1, a_1), \cdots, \alpha R_1(s_{N_s+1}, a_2), \\ & \vdots \\ & \alpha^n R_{N_m}(s_1, a_1), \cdots, \alpha^n R_{N_m}(s_{N_s+1}, a_2)] \end{aligned} \quad (13)$$

The $\mathbf{c}$ vector contains all the discounted rewards from the SDP, and hence $\mathbf{c}$ will be referred to as the LP reward vector. When the LP is in standard form, the standard Simplex algorithm [7] is used to find the optimal basic variables.

4 Sensitivity Analysis of the Linear Program

Once the linear program is in standard form (10), we can apply well known sensitivity analysis techniques for LP. The LP $\mathbf{c}$ coefficients allow for sensitivity analysis of the state rewards $r(s_i, a_1)$, the monitor rewards l_i, the state reward constant K and the discount factor α. In this paper we study the sensitivity analysis of the discount factor α, which is one of the difficult SPC parameters to determine.

Let $\mathbf{x}^* = (\mathbf{x}_B^*, \mathbf{x}_N^*)$ be the optimal basic feasible solution (bfs), where $\mathbf{x}_B^*$ are the optimal basic variables and $\mathbf{x}_N^*$ are the optimal non-basic variables. Let N_B and N_N be the number of basic and non-basic variables, respectively. Let $\mathbf{B}$ be an $N_B \times N_B$ submatrix of $\mathbf{A}$ corresponding to the basic variables and let $\mathbf{N}$ be an $N_B \times N_N$ submatrix of $\mathbf{A}$ corresponding to the non-basic variables. Let $\mathbf{c}_B$ and $\mathbf{c}_N$ be the reward coefficients of the basic and non-basic variables, respectively. Then the LP optimality criterion can be formulated as follows.

$$\mathbf{c}_N - \mathbf{c}_B \mathbf{B}^{-1} \mathbf{N} \leq \mathbf{0} \quad (14)$$

See for example [12] for the proof. The matrix $\mathbf{B}^{-1}$ can be found from the output matrix of the Simplex algorithm by selecting the columns corresponding to the basic variables. Equation (14) can be used to determine the range of values over which the elements of $\mathbf{c}$ can be varied without changing the optimal bfs, $\mathbf{x}^*$. The method in Table 1 calculates the α values for each optimal bfs $\mathbf{x}^*$. The corresponding vectors $\mathbf{x}_B^*(\alpha)$

1. Let $\alpha = 1$
2. Find $\mathbf{c}'(\alpha) = \mathbf{c} \cdot \mathbf{f}(\frac{\alpha}{\alpha_0})$ and solve LP, ie. find $\mathbf{x}_B^*(\alpha)$.
3. Find $0 \leq \epsilon \leq 1$ such that
$$\bar{\mathbf{c}}_N - \bar{\mathbf{c}}_B \mathbf{B}^{-1} \mathbf{N} \leq \mathbf{0}$$
where $\bar{\mathbf{c}} = (\bar{\mathbf{c}}_B, \bar{\mathbf{c}}_N) = \mathbf{c}' \cdot \mathbf{f}(\epsilon)$
4. $\alpha = \epsilon\alpha$. Save α value and corresponding $\mathbf{x}_B^*(\alpha)$, $\mathbf{c}_B'(\alpha)$.

While $\alpha > 0$ go to 2.

Table 1: *Method for finding the intervals of α where the bfs $\mathbf{x}^*$ stays constant.*

and $\mathbf{c}_B'(\alpha)$ are stored and will be used later to find the optimal value function V_{N_m}.

The nominal value of α used in equation (13) is denoted α_0, $\mathbf{f}$ is a vector valued function of same dimension as $\mathbf{c}$ and $\cdot$ denotes array multiplication. The vector valued function $\mathbf{f}$ is given by

$$\mathbf{f}_i(\epsilon) = \epsilon^n \quad (15)$$

where n is the iteration number of $\mathbf{c}_i$ in equation (13). The method above will calculate a discrete set $\mathcal{E}$ of values for α. Between any two neighbour values in $\mathcal{E}$ the optimal bfs $\mathbf{x}^*$ is constant. Hence, the optimal action, a_1 or a_2, is also constant between any two neighbour values of α in $\mathcal{E}$.

Until now we have described how to analyse the sensitivities of the parameter α for a fixed permutation of monitors. What we are really interested in, is the sensitivity of α on the decisions made by the sensory perception controller. For example, if the initial SDP state is $\sigma_0 = 0.5$ and the discount factor changes from 0.9 to 0.7, will this cause the SPC to consult another monitor first? To answer these kinds of questions, the method in Table 1 must be executed on all possible permutations. The optimal permutation z^* is given by the optimal value function. Hence, the value functions must also be calculated for all possible permutations.

Once the discrete set $\mathcal{E}$ and the corresponding $\mathbf{x}_B^*(\alpha)$, $\mathbf{c}_B'(\alpha)$ are found for all permutations, the calculation of z^* is straight-forward. First, the value functions are calculated as follows. Let $0 \leq \delta \leq 1$. For all values of δ, let $\mathbf{x}_B^*(\delta)$ be the basic components of the optimal bfs and $\bar{\mathbf{c}}_B(\delta)$ be the basic components of the modified reward vector. Then, we have

$$V_{N_m}(\sigma_0, z, \delta) = \bar{\mathbf{c}}_B(\delta)^T \mathbf{x}_B^*(\delta) \quad (16)$$

The modified reward vector $\bar{\mathbf{c}}_B(\delta)$ is calculated from the stored values of $\mathbf{c}_B'$ and α in Table 1 and the function $\mathbf{f}$, ie. $\bar{\mathbf{c}}_B(\delta) = \mathbf{c}_B'(\alpha) \cdot \mathbf{f}(\frac{\delta}{\alpha})$.

The optimal permutation z^* is finally given by

$$z^*(\delta) = \text{argmax}_z V_{N_m}(\sigma_0, z, \delta) \qquad (17)$$

5 Case Study

As in [3] we study a sensory perception controller with $N_m = 6$ available process monitoring techniques. The monitors are numbered from 1 to 6 and have characteristics as shown in Figure 2.

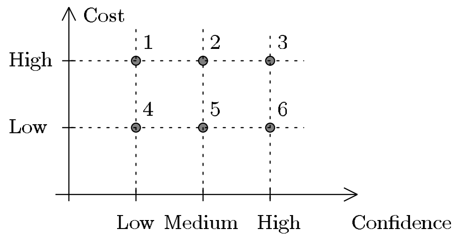

Figure 2: *6 different classes of monitoring techniques.*

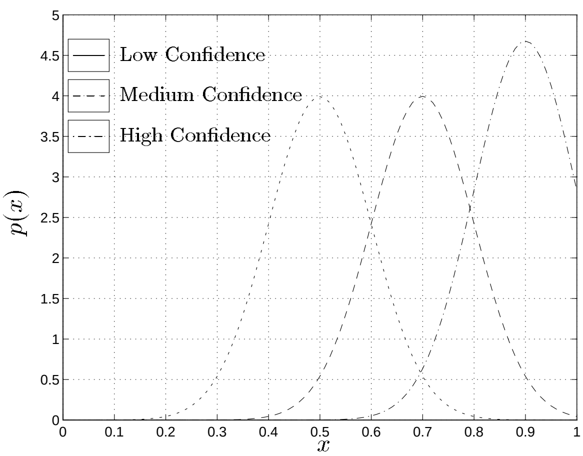

Figure 3: *Probability density functions for the monitor confidence levels.*

The probability density functions for the low, medium and high confidence monitors are shown in Figure 3. The monitor costs are given by $k_1 = k_2 = k_3 = 2$ and $k_4 = k_5 = k_6 = 1$. The monitor rewards are calculated by $l_i = k_{\max} - k_i$. The continuation monitor reward is given by

$$L_n(z, a_2) = l_{m_n(z)} \qquad (18)$$

where $l_{m_n(z)} > 0$ is the reward of using the process monitor at the n-th position in structure z.

As in [3] the discretised state space $\mathcal{S}$ for the confidence levels and the state rewards $\mathcal{R}$ were chosen as follows.

$$\begin{aligned} \mathcal{S} &= \{0.0,\ 0.5,\ 0.75,\ 0.875,\ 0.9375, \\ &\quad 0.96875,\ 0.984375,\ 0.9921875,\ s_\infty\} \qquad (19) \\ \mathcal{R} &= K * \{0,\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 0\} \qquad (20) \end{aligned}$$

The scaling factor is chosen to be $K = k_{\max}$.

Figures 4 and 5 show the sensitivity of the decisions made by the SPC as a function of the discount factor α. The method in Table 1 was used to find the optimal bfs $\mathbf{x}^*$ for all the permutations and equations (16) and (17) were used to find the optimal permutation z^*.

All the process monitors are available to the SPC. Figure 4 shows which process monitor is consulted first when the initial state is $\sigma_0 = s_2$. Figure 5 shows the same results when the initial state is $\sigma_0 = s_3$.

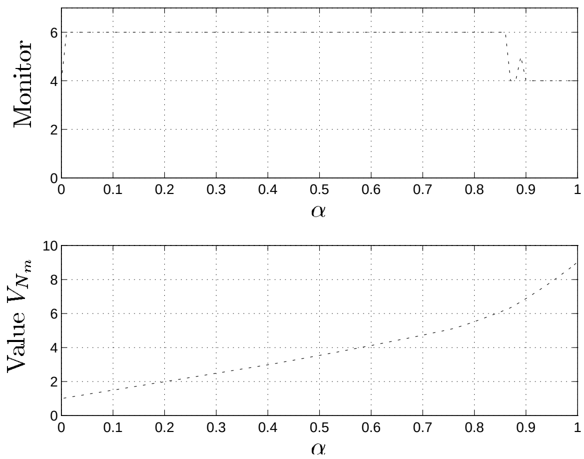

Figure 4: *Decisions made by the SPC as a function of the discount factor α with all 6 monitors available. The initial state is $\sigma_0 = s_2 = 0.50$. Note that the optimal action is always to consult a monitor (4,5 or 6), except for $\alpha = 0$.*

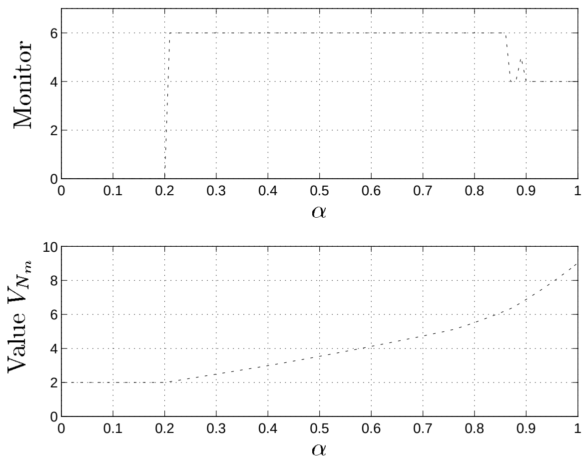

Figure 5: *Decisions made by the SPC as a function of the discount factor α with all 6 monitors available. The initial state is $\sigma_0 = s_3 = 0.75$. Where monitor=0, the optimal action is terminate.*

The main difference between Figures 4 and 5 is the α values for which terminate is the optimal action. The discount factor only alters future rewards. If the optimal action is to terminate no future rewards will be given and hence the value function V_{N_m} is independent of α as seen from Figure 5. When α decreases, the continuation reward will sooner or later become smaller than the termination reward and hence terminate will be the optimal action for small α values. The termination reward is determined from the initial state and this fact explains the difference between Figure 4 and 5. As seen from equation (20), no state reward is given for s_2 and hence the SPC never terminates in Figure 4 for $\alpha > 0$.

The discount factor value influences the average SPC cost and event recognition rate. In [3] good results were achieved for α values between 0.6 and 0.7. From Figure Figure 4 and 5 we see that the local sensitivities of the SPC decisions when $\alpha \in [0.6, 0.7]$ are low.

The results from the LP sensitivity analysis in Figures 4 and 5 and were confirmed for several α values using the original SDP algorithm developed in [3]. Note that one of the low cost monitors (4,5,6) is always consulted first. When the discount factor is large, monitor 4 is consulted first. This is caused by the fact that future rewards are not discounted by a large amount. Hence, monitor 4 is consulted first to gain the extra monitor reward. Then, later, other monitors will be used to increase the confidence levels. As α decreases, future rewards are worth less and the SPC decides to use the best individual monitor first, monitor 6.

6 Discussion and Conclusion

A new method for the sensitivity analysis of a sensory perception controller (SPC) in a discrete event framework has been developed. The original SPC developed in [3] uses stochastic dynamic programming (SDP) to solve the decision process. In a case study with 6 different process monitors, the sensitivity to the model parameter α was found to be relatively low. The results from the analysis give insight into the SDP algorithm and are invaluable when selecting the SDP model parameters.

When changing the model parameters, it is of course possible to formulate the new problem (SDP or LP) and to solve it again from scratch. However, this approach is both wasteful and unnecessary. The criterion in equation (14) tells us when a solution is no longer optimal. Hence, we only need to solve the new problem when the optimality test fails.

As seen from the figures in the previous section, the SPC has a relatively low sensitivity to changes in the model parameter α. For example, in Figure 4 the optimal solution is always monitor 6 when $\alpha \in [0.2, 0.85]$. The low sensitivity to the model parameters is desirable, and it simplifies the tuning of the discount factor α. The LP $\mathbf{c}$ coefficients allow for a similar analysis of other parameters such as K, $r(s_i, a_1)$ and l_i.

The SPC is formulated as a Markov decision process and uses stochastic dynamic programming to evaluate all orderings of the process monitors. As demonstrated in the paper, the Markov decision formulation allows for sensitivity analysis by using linear programming. This facility for sensitivity analysis is a major advantage of the sensory perception controller.

References

[1] A.V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming.* Mathematics in Science and Engineering, Vol. 165, Academic Press, New York, 1983.

[2] V. Chvátal, *Linear Programming*, W.H. Freeman and Company, New York, 1983.

[3] G.E. Hovland and B.J. McCarragher, "Control of Sensory Perception in Discrete Event Systems Using Stochastic Dynamic Programming", *ASME Journal of Dynamic Systems, Measurement and Control*, Submitted July 1997.

[4] G.E. Hovland and B.J. McCarragher, "Control of Sensory Perception Using Stochastic Dynamic Programming", *Proceedings of the 1st Australian Data Fusion Symposium*, pp. 196-201, Adelaide, 21-23 November 1996.

[5] L.C.M. Kallenberg, *Linear Programming and Finite Markovian Control Problems*, Matematisch Centrum, Tract No. 148, 1983, Amsterdam.

[6] H. Mine and S. Osaki, *Markovian Decision Processes.* American Elsevier Publishing Company, Inc., 1970.

[7] W.H. Press, et.al, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edition, Cambridge University Press, 1992.

[8] M.L. Puterman, *Markov Decision Processes.* John Wiley & Sons, Inc., 1994.

[9] S.M. Ross, *An Introduction to Stochastic Dynamic Programming.* New York, Academic Press, 1983.

[10] R.I. Rothenberg, *Linear Programming*, Elsevier North-Holland, Inc., New York, 1979.

[11] R.Y. Rubinstein and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method,* John Wiley & Sons, Ltd., 1993.

[12] D. Solow, *Linear Programming: An Introduction to Finite Improvement Algorithms.* Elsevier Science Publishing Co., Inc., 1984.

[13] P.R. Thie, *An Introduction to Linear Programming and Game Theory*, John Wiley & Sons, Inc., 1988.

[14] L. Xu and S. Ahmed-Zaid, "Tuning of Power System Controllers Using Symbolic Eigensensitivity Analysis and Linear Programming", *IEEE Transactions on Power Systems*, Vol. 10, No. 1, pp.314-321, Feb. 1995.