# Observation Process Control in Support of Stochastic Tasking Operations

William M. McEneaney Ali Oran

*Abstract*— **We consider a problem of observation control, specifically a problem where one chooses which aspects of the state to observe at each time-step. The state takes values in a finite set, and the conditional probability updates by Bayes' rule. The payoff for observation takes the form of a finite maximum of linear functions of the final observation-conditioned probability distribution, and so is a convex function of the distribution. However, the goal is maximization, not minimization. Through use of the max-plus distributive property, we are able to use a max-plus curse-of-dimensionality-free computational method for solution of the control problem. Complexity attenuation of the algorithm is addressed.**

## I. Introduction

We consider a problem of observation control. That is, one wishes to choose which physical state components to observe so as to optimize some payoff induced by the resulting conditional probability. The associated control problem has a state (distinct from the above physical state) which consists of the observation-conditioned probability distribution. In the case of a discrete set of feasible states, the dynamics of the state process include Bayesian updates of the distribution, and so forms a nonlinear Markov state process model. For the problems of interest, the payoff takes the form of a pointwise maximum of linear functions over the probability simplex. One may employ a max-plus based method to propagate the dynamic program. At each step of the dynamic program, one may contain the solution complexity by projecting the solution approximation onto the optimal max-plus subspace of a specified dimension. This can be shown to be equivalent to maximization of a monotonic, convex function over a cornice (which is defined as a union of downward cones over a convex hull of a set of points defining the cornice). Exceptionally rapid solution times are obtained.

The motivation for this problem stems from a UAV tasking application. In particular, one is concerned with using sensing UAVs to reduce human losses for teams operating in an urban setting. At the outset, one has some a priori probability distribution describing the knowledge of Red (opposing) force configurations among a finite set of locations (e.g., buildings). One may task the UAV to examine some of these locations. A resulting observation might be reduced to say, 1 or 0 depending on whether a Red force was found in the building, or not. Of course, the observation would be corrupted by noise. As one updates the probability distribution, via Bayes rule, one obtains a stochastic state process. This state process is the conditional probability, and the input noise process is due to the observation noise. This yields a nonlinear Markov state process. The payoff form alluded to in the above paragraph can be shown to be quite general for applications in this class. Note that the pointwise maximum of linear functions over the probability simplex is a convex function over the simplex, and thus has a tendency to have the minimum in the interior, while the maximum will be at an extreme point of the simplex. In the application, this corresponds to the notion that the optimal situation is for the Blue (our) forces to have certainty that the Red state is the least problematic state, while the worst situation tends to be one where one has little knowledge of the opposition location(s).

Dept. of Mechanical and Aerospace Engineering, University of California San Diego, San Diego, CA 92093-0411, USA. wmceneaney@ucsd.edu
Research partially supported by NSF grant DMS-0307229.
Dept. of Mechanical and Aerospace Engineering, University of California San Diego, San Diego, CA 92093-0411, USA. aoran@ucsd.edu
Research partially supported by NSF grant DMS-0307229.

## II. Problem Statement

Suppose the physical state belongs to a finite set indexed as $\mathcal{N} \doteq ]1,N[ \doteq \{1,2,\dots N\}$. Let the information regarding this physical state be described by probability distribution $q$ in the probability simplex $S^N$ where

$$S^N = \left\{ q \in I\!R^N \,\middle|\, q_n \in [0,1]\,\forall n \in \mathcal{N} \text{ and } \sum_{n\in\mathcal{N}} q_n = 1 \right\}.$$

We suppose that the payoff for the sensing operation takes the form

$$W(q) \doteq \max_{i\in\mathcal{I}} \{v^i \cdot q\}, \tag{1}$$

where $\mathcal{I}$ is some index set, and each $v^i \in I\!R^N$. It is worth noting that $W$ is a piecewise linear function of its argument, the probability distribution regarding the physical state; this form will be exploited in Section III.

We consider a discrete-time, finite time-horizon formulation with time denoted by $t \in \mathcal{T} \doteq ]0,T[ \doteq \{0,1,2,\dots T\}$. At the terminal time, the payoff for the observation tasking over $\mathcal{T}$ is given by $W(q_T)$ where $q_T$ is the observation-conditioned probability at terminal time, and the terminal payoff, $W$ is given by (1). As stated earlier, the estimator that defines the dynamics of the information state is based on Bayes rule. Here, for simplicity, we suppose that at each time-step, the sensor can choose among any observation task in a finite set, which we index as $\mathcal{U} \;=]1,N_u[= \{1,2,\dots N_u\}$. In particular, we simplify the problem here by allowing the observation control to instantaneously change from any task to any other in a single step. With this freedom, we do not need to include a state process component corresponding to the sensor task. Suppose the sensor task

ISBN 978-963-311-370-7 

at time, $t$, is $u_t \in \mathcal{U}$. We assume a finite set of possible observation returns, and so without loss of generality, the resulting observation, $y = y_t$, will take a value in $\mathcal{Y} \doteq ]1, N_y[$. Let $R_j^{y,u}$ be the probability of observation $y \in \mathcal{Y}$ given that the current sensor task is $u_t = u \in \mathcal{U}$ and the physical state is $j \in \mathcal{N}$. Let $\mathbf{R}^{y,u}$ be the vector of length $N$ with components $R_j^{y,u}$, and let $D(\mathbf{R}^{y,u})$ be the $N \times N$ matrix with diagonal elements $[D(\mathbf{R}^{y,u})]_{j,j} = R_j^{y,u}$, and such that $[D(\mathbf{R}^{y,u})]_{i,j} = 0$ for $i \neq j$. Then, given any sensing action $u_t \in \mathcal{U}$ and resulting (random-variable) observation $y_t$, one has:

$$q_{t+1} = \frac{1}{\mathbf{R}^{y_t,u_t} \cdot q_t} D(\mathbf{R}^{y_t,u_t}) q_t, \doteq \beta^{y_t,u_t}(q_t), \qquad (2)$$

which defines the stochastic information state dynamics. Note that this is a nonlinear Markov state process.

For the closed loop controller, the state at time $t$ consists of the current sensor task and the current information state, $q_t$. As the sensor can move from any task to any other in one time-step, we will suppress the sensor-task as state component. Now let $\mathcal{A} \doteq \{\alpha : S^N \to \mathcal{U}\}$, and let $\mathcal{A}^t$ be the set of nonanticipative feedback controls from time $s$ to terminal time, $T$. That is, we let

$$\mathcal{A}^s \doteq \big\{\alpha_{[s,T-1]} : [S^N]^{T-s} \to \mathcal{U}^{T-s} \big| \text{ if } q_r = \hat{q}_r \text{ for all } r \leq \bar{t} \text{ then } \alpha_r[q.] = \alpha_r[\hat{q}.] \text{ for all } r \leq \bar{t}\big\}$$

where $\mathcal{U}^{T-s}$ denotes the outer product of $\mathcal{U}$, $T - s$ times, and similarly with $[S^N]^{T-s}$. The payoff for information state $q_s = q$ and non-anticipative control $\alpha \in \mathcal{A}^s$ is

$$J(s, q, \alpha.) \doteq \mathbf{E}\{W(q_T)\} \qquad (3)$$

where the propagation of the state from $q_s = q$ to $q_T$ follows (2) with control $u_t = \alpha_t[q.]$ at each time, $t$. The corresponding value function is:

$$V(s, q) = \sup_{\alpha. \in \mathcal{A}^s} J(s, q, \alpha.) \qquad (4)$$

## III. Max-plus formulation of the dynamic program

The dynamic program is simply

$$V(t, q) = \max_{u \in \mathcal{U}} \mathbf{E}_{y_t}\left\{V(t+1, \beta^{y,u}(q))\right\}, \qquad (5)$$

where the expectation is over the set of possible observations.

One can notice that for larger values of $N$, the dynamic programming computations would become computationally infeasible when performed over the discretized probability simplex (grid-based methods), even for short time spans. However, the special form of $V(t, q)$ inherited from $W(q)$ can be exploited to avoid this problem. If we can show that this form is retained under the dynamic programming propagation, then we will be able to work with the $v^i$ vectors instead of a discretized form of $V(t, \cdot)$ over $S^N$. In order to demonstrate this, we first introduce the following notation. For any set, $\mathcal{I}$, and positive integer $M$, let $\mathcal{P}^M(\mathcal{I})$ be the set of all sequences of length $M$ with elements from $\mathcal{I}$. (Note that the cardinality of $\mathcal{P}^M(\mathcal{I})$ is $(\#\mathcal{I})^M$.)

*Theorem 3.1:* Suppose $V(t+1, q)$ takes the form

$$V(t+1, q) = \max_{i \in \mathcal{I}_{t+1}} v_{t+1}^i \cdot q$$

where $\mathcal{I}_{t+1} = \{1, 2, \ldots I_{t+1}\}$. Then,

$$V(t, q) = \max_{i \in \mathcal{I}_t} v_t^i \cdot q \qquad (6)$$

where $\mathcal{I}_t = \{1, 2, \ldots I_t\}$, $I_t = N_u (I_{t+1})^{N_y}$,

$$v_t^i = \sum_{y_t \in \mathcal{Y}} D(\mathbf{R}^{y_t,u_t}) v_{t+1}^{j_{y_t}} \qquad (7)$$

where $(u, \{j_{y_t}\}) = \mathcal{M}^{-1}(i)$, and $\mathcal{M}$ is a one-to-one, onto mapping from $\mathcal{U} \times \mathcal{P}^{N_y}(\mathcal{I}_{t+1}) \to \mathcal{I}_t$ (i.e., an indexing of $\mathcal{U} \times \mathcal{P}^{N_y}(\mathcal{I}_{t+1})$ ).

We now develop some helpful notation. For any $t$, let $\mathcal{V}_t \doteq \{v_t^i \,|\, i \in \mathcal{I}_t\}$. Then, by Theorem 3.1, the dynamic program can equivalently be given as

$$(\mathcal{V}_t, \mathcal{I}_t) = \mathcal{D}^{\mathcal{U}}\big[(\mathcal{V}_{t+1}, \mathcal{I}_{t+1})\big],$$

where the operator, $\mathcal{D}^{\mathcal{U}}$ is defined by the propagation (7). Also, we can denote the reconstruction of $V(t, \cdot)$ from the pair $(\mathcal{V}_t, \mathcal{I}_t)$ as $V(t, \cdot) = \mathcal{C}\big[(\mathcal{V}_t, \mathcal{I}_t)\big]$, where the reconstruction operator is given by (6). Note that, although the gridding of the state space is avoided, a very heavy complexity cost is incurred through the rapid growth of $\#\mathcal{I}_t$ as one propagates backward.

## IV. Complexity Attenuation

Between each propagation step, one can look for a simpler approximation to the solution. Note that $V(t, \cdot)$ is represented as a pointwise maximum of linear functionals. One may think of $V(t, \cdot)$ as an element of the max-plus vector space of convex functionals. Suppose $\#\mathcal{I}_t = M$. We look for an approximation of $V(t, \cdot)$ given as a pointwise maximum of $\widetilde{M}$ linear functionals where $\widetilde{M} < M$. Rather than a priori selecting a truncated basis with which to approximate the solution, the subspace is being chosen at each step.

To identify which vectors to eliminate from $\mathcal{V}_t$, we first need to define an error function at time $t$, $\epsilon_t$, over the probability simplex, $S^N$, for the pruning analysis. Two candidates chosen for this purpose were the functions defined by $L_\infty$ and $L_1$ norms. We first present our results with the $L_\infty$ norm and later with the $L_1$ norm.

Suppose that at time $t$, a number of vectors were pruned out of the index set $\mathcal{I}_t$, leaving us with the pruned set, $\mathcal{P}_t$. We define the error, occurring by omitting those vectors out of $\mathcal{I}_t$, using the $L_\infty$ norm as:

$$\epsilon_\infty(t) \doteq \| V(t, q) - \bar{V}(t, q) \|_\infty = \max_{q \in S^N} \left\{V(t, q) - \bar{V}(t, q)\right\}$$

where

$$V(t, q) = \max_{i \in \mathcal{I}_t} \left\{v_t^i \cdot q\right\} \quad \text{and} \quad \bar{V}(t, q) = \max_{i \in \mathcal{P}_t} \left\{v_t^i \cdot q\right\}$$

Since $\epsilon_\infty(t)$ is defined as the maximum over the simplex $S^N$, calculation of this error can be considered as an optimization problem over that simplex. The simple method chosen for this purpose is the well documented linear programming algorithm.

Suppose that at some time $t$, a number of vectors are already present in the set $\mathcal{V}_t$. We want to determine the maximum amount of error over the simplex $S^N$ induced by omitting $v_t^i$ out of $\mathcal{V}_t$. The following optimization scheme finds this maximum error.

$$\begin{aligned}
\max_{q\in\mathcal{S}^N} \quad &: \quad v_t^i \cdot q - z \\
\text{subject to} \quad &: \quad v_t^j \cdot q - z \le 0, \qquad \forall j \neq i, i \in \mathcal{I}_t, j \in \mathcal{I}_t \\
&\quad\ \ z \ge 0 \\
&\quad\ \ q_k \ge 0 \quad \forall k \quad \text{and} \quad \sum_k q_k = 1
\end{aligned}$$

Now, writing the inequlaties and equalities in the all-inequality form one formulates:

$$\begin{aligned}
\min_x \quad &: \quad c^i \cdot x, \quad x = [q^T\ z]^T, \quad c^i = [-v_t^{i^T}\ 1]^T \\
\text{subject to} \quad &: \quad A\,x \ge 0,
\end{aligned}$$

for proper choice of $A$, and we do not include the details. It should also be noted that when this last optimization scheme results in a negative value (positive in the first scheme) then it shows that the corresponding vector $v_t^i$ contributes to the overall value of $V(t,q)$. We call such vectors active. Otherwise, if the last optimization scheme results in a positive value (negative in the first scheme) then it shows that the corresponding vector $v_t^i$ does not contribute to the overall value of $V(t,q)$. We call such vectors inactive and they bring no error when eliminated.

Running the above scheme for a vector $v_t^i \in \mathcal{V}_t$, one can find the error induced by omitting vector $v_t^i$ from $V(t,q)$ computations. We define this error as $\epsilon_\infty^i(t)$ and it can formulated as:

$$\begin{aligned}
\epsilon_\infty^i(t) &\doteq \max_{q\in S^N} \left\{ V(t,q) - \bar{V}_i(t,q) \right\}, \quad \text{where} \\
\bar{V}_i(t,q) &\doteq \max_{j\in\mathcal{I}_t, j\neq i} (v_t^j \cdot q)
\end{aligned}$$

The maximum of these errors over the set of pruned vectors would yield $\epsilon_\infty(t)$:

$$\epsilon_\infty(t) = \max_{i\in\mathcal{I}_t-\mathcal{P}_t} \epsilon_\infty^i(t)$$

Relevantly, the pruned set $\mathcal{P}_t$ can be defined as: $\mathcal{P}_t \doteq \{i \in \mathcal{I}_t \mid \epsilon_\infty^i(t) < \bar{\epsilon}(t)\}$, where $\bar{\epsilon}(t)$ is an upper error bound for the time $t$.

At this point, one might wonder about the existence of a possible error bound during DP iterations. The following theorem highlights the boundedness property of the error, $\epsilon_\infty(t)$.

*Theorem 4.1:* Let $V(t+1,q)$ and $\bar{V}(t+1,q)$ be the functions defined above. If

$$\epsilon_\infty(t+1) = \parallel V(t+1,q) - \bar{V}(t+1,q) \parallel_\infty = \epsilon$$

then

$$\epsilon_\infty(t) = \parallel V(t,q) - \bar{V}(t,q) \parallel_\infty \le \epsilon$$

*Proof:* By the definition of V(t+1,q) and $\bar{V}(t+1,q)$,

$$V(t,q) = \max_{u_t} \left\{ \sum_{y_t} V(t+1, \beta^{y_t,u_t}(q)) P(y_t) \right\}$$

$$\text{and} \quad \bar{V}(t,q) = \max_{u_t} \left\{ \sum_{y_t} \bar{V}(t+1, \beta^{y_t,u_t}(q)) P(y_t) \right\}.$$

$$V(t,q) - \bar{V}(t,q) = \max_{u_t} \Bigg\{ \sum_{y_t} \big[ V(t+1, \beta^{y_t,u_t}(q)) - \bar{V}(t+1, \beta^{y_t,u_t}(q)) \big] P(y_t) \Bigg\}$$

Following (3) $\beta^{y_t,u_t}(q) \in S^N$ and then:

$$\parallel V(t,q) - \bar{V}(t,q) \parallel_\infty \le \max_{u_t} \left\{ \sum_{y_t} \epsilon\ P(y_t) \right\} = \epsilon$$

■

*Corollary 4.2:* If for some $t \in \{0, ..., T\}$, $\epsilon_\infty(t) = \epsilon$, then for all $0 \le \tau \le t$, one has $\epsilon_\infty(\tau) \le \epsilon$.

*Proof:* The proof follows by applying an induction argument using Theorem 4.1. ■

This corollary states that the error induced by pruning a set of vectors out of the analysis at time $t$, would not grow during subsequent steps in the DP. The easy use of linear programming and the boundedness of the pruning error make the error function defined by the $L_\infty$ norm look like a perfect candidate for measuring the error in pruning. However, $\epsilon_\infty$ also possesses two disadvantages. First, and most importantly, it was found that pruning was not optimal for approximating $V(t,q)$ with a smaller set using the $L_\infty$ norm. That is, the optimal set of, say $\widetilde{M}$, vectors for approximating $V(t,\cdot)$ (where $\widetilde{M} < \#\mathcal{I}_t = \#\mathcal{V}_t$) may not consist of a subset of the elements of $\mathcal{V}_t$. Second, the authors are concerned that using the $L_\infty$ norm might not be an accurate way to measure the pruning error. Because of these drawbacks an error function based on the $L_1$ norm, $\epsilon_1(t)$, was also considered.

We define $\epsilon_1(t)$ as below:

$$\epsilon_1(t) \doteq \int_{S^N} V(t,q) - \bar{V}(t,q) dq$$

where $V(t,q)$ and $\bar{V}(t,q)$ are as defined earlier. The main advantage of using $\epsilon_1(t)$ was highlighted in [3], where it was proven that an error function based on the $L_1$ norm would be convex, and moreover, when approximating a set of functions with another smaller set of functions the optimal reduced complexity representation would be comprised of a subset of the original set of functions. That is, with an error metric based on the $L_1$ norm, pruning does, in fact, yield the optimal solution. This is the superiority of the $L_1$ norm over the $L_\infty$ norm. Having an optimal set of pruned vectors over the refined set $\mathcal{R}_t$ we are encouraged us to use $\epsilon_1(t)$ over $\epsilon_\infty(t)$. However, contrary to these fine properties $\epsilon_1(t)$ did not posses the boundedness property of $\epsilon_\infty(t)$ during DP. A counterexample is given below.

In the two-dimensional simplex, $S^2$, consider $\mathcal{R}_{t+1} = \{1,3\}$ with $v^1_{t+1} = [0.95\ 0.25]'$, and $v^3_{t+1} = [0.7\ 0.65]'$. Suppose that out of these 2 vectors we need to prune the vector that will result to the the minimal pruning error, $\epsilon_1(t+1)$. Similar to $\epsilon^i_\infty(t)$ defined earlier, we define the pruning error induced by pruning a vector $i$ from the refined set, $\mathcal{R}_t$, but this time based on the $L_1$ norm as:

$$\epsilon^i_1(t+1) \doteq \int_{S^N} V(t+1,q) - \bar{V}_i(t+1,q)\ dq$$

where $\bar{V}_i(t+1,q)$ was defined in the previous page. Now, since we are trying to prune out one single vector that would lead to least error in our analysis, the error $\epsilon_1(t+1)$ would be $\epsilon_1(t+1) = \min_{i\in R_{t+1}} \epsilon^i_1(t+1)$ Following the definition of $\epsilon^i_1(t+1)$ we find, $\epsilon^1_1(t+1) = 0.0481$, and $\epsilon^3_1(t+1) = 0.1231$. Then, $\bar{\epsilon_1}(t+1) = 0.0481$ and vector 1 should be pruned out to give us the pruned set, $P_{t+1} = \{3\}$. Using (6), (7), and (8) the piecewise linear functions defining $V(t,q)$ and $\bar{V}(t,q)$ is found. Integrating the difference between $V(t,q)$ and $\bar{V}(t,q)$, one finds $\epsilon_1(t) = 0.1058$.

Because of this unboundedness one might be worried about error growth during the propagation process. However, even tough $\epsilon_1(t)$ might grow during subsequent steps in DP, an upper bound for the error growth is always maintained because of the relationship between the $L_\infty$ and $L_1$ norms. The following theorem highlights this fact.

*Theorem 4.3:* Suppose at some time $t \in \{0,...,T\}$, $\epsilon_\infty(t) = \epsilon$. Then during DP, for any $0 \le \tau \le t$, $\epsilon_1(\tau) \le \epsilon$.

*Proof:* Following the definition of $\epsilon_1(\tau)$ and $\epsilon_\infty(\tau)$ we have: $\epsilon_1(\tau) = \int_{S^N} V(\tau,q) - \bar{V}(\tau,q)\ dq \le \int_{S^N} \epsilon_\infty(\tau)dq = \epsilon_\infty(\tau)$. Now, from Corollary 4.2, we have $\epsilon_\infty(\tau) \le \epsilon_\infty(t) = \epsilon$ for any $0 \le \tau \le t$. Combining these, one finds that $\epsilon_1(\tau) \le \epsilon$. ■

Although this theorem eases our worries about unbounded growth of $\epsilon_1(t)$, it can be easily seen that it is actually a conservative bound. Here, one needs to trade off between computation speed and error bounding carefully. Another issue that needs attention is the numerical computation methods needed to calculate $\epsilon_1(t)$. Unlike $\epsilon_\infty(t)$ one cannot use linear programming, and numerical integration methods are required for complex piecewise functions that define $V(t,q)$. For probability simplexes with dimensions lower than four, volume calculation algorithms not subject to the curse-of-dimensionality can be developed with the help of visual aids. For higher dimensions, the construction of such volume computation algorithms appears to be a difficult task.

## References


[1] F.L. Baccelli, G. Cohen, G.J. Olsder and J.-P. Quadrat, *Synchronization and Linearity,* John Wiley, New York, 1992.

[2] V.N. Kolokoltsov and V.P. Maslov, *Idempotent Analysis and Its Applications,* Kluwer, 1997.

[3] W.M. McEneaney, *Complexity Reduction, Cornices and Pruning,* Proc. of the International Conference on Tropical and Idempotent Mathematics, G.L. Litvinov and S.N. Sergeev (Eds.), AMS, (2009).

[4] W.M. McEneaney, A. Oran, A. Cavender, *Value-Based Tasking Controllers for Sensing Assets,* Proc. 2008 AIAA Guidance, Navigation and Control Conference.