Relaxed dynamic programming in switching systems

A. Rantzer

Abstract: In order to simplify computational methods based on dynamic programming, a relaxed procedure based on upper and lower bounds of the optimal cost was recently introduced. The convergence properties of this procedure are analysed here. In particular, it is shown that the computational effort in finding an approximately optimal control law by relaxed value iteration is related to the polynomial degree that is needed to approximate the optimal cost. This gives a rigorous foundation for the claim that the search for optimal control laws requires complex computations only if the optimal cost function is complex. A computational example is given for switching control on a graph with 60 nodes, 120 edges and 30 continuous states.

1 Introduction

Optimal switching between linear systems is in many respects as challenging as optimal control of general non-linear or hybrid systems. It is rarely possible to find exact expressions for optimal control laws or the optimal cost. Instead approximative solutions need to be sought. Already in Bellman's pioneering work on dynamic programming [1], the need for approximate solutions was recognised and discussed. Since then, a variety of methods have been developed, with application to discrete optimisation as well as Markov processes, differential equations and hybrid systems. Of particular significance for this paper is the inequality version of the Hamilton–Jacobi–Bellman equation, used by Leake and Liu [2] to derive bounds on the optimal cost function. It turns out that the inequality for lower bounds on the optimal (minimal) cost is convex. This gives a natural connection to convex duality theory in optimal control, an idea introduced by Kantorovich [3] for mass transporation problems, which has recently been further explored [4–7]. An application to image databases is described in [8]. Computational methods based on convex optimisation were pursued in [9, 10] and the idea of relaxed dynamic programming was introduced in [11, 12].

There are two important iterative approaches to dynamic programming, known as value iteration and policy iteration. Value iteration is the basis for this paper. In most applications, iterations in policy space would require fewer iterations, but each iteration is more computationally demanding and harder to parallelise. A detailed analysis of policy iteration convergence was given by Puterman and Brumelle [13]. For policy iteration with approximations the analysis is still a subject of research [14].

Numerical solutions to the Hamilton–Jacobi–Bellman equation in a continuous state-space are often based on discretisation [15–18]. This gives a connection to the rich literature on optimal control in discrete state spaces [19]. In particular, error bounds for approximate dynamic programming were given in [20, 21]. An alternative method which avoids discretisation is to use Galerkin's spectral method to approximate the optimal cost function without prior discretisation [22]. Altogether, existing methods have proved effective for many small scale problems, but the complexity grows exponentially with increasing state dimension.

In contrast to general non-linear methods with exponential growth, it is well known that linear-quadratic optimal control problems grow only polynomially with state dimension and can be solved with hundreds of state variables. It is therefore challenging to search for general non-linear synthesis procedures that reduce to Riccati equations in the special case of linear-quadratic control and to linear programming in the case of network optimisation on a finite graph. One step in this direction was taken in [23]. This paper proceeds towards the goal in a more general setting.

Recent research on model predictive control and optimal control of hybrid systems is also connected to this work [24–27]. In fact, our approach resulted from an effort to treat hybrid systems by merging methods and experiences from the two fields of network optimisation and control theory. In particular, convex inequality relaxations commonly used in network optimisation are combined with computational tools from the control field, such as linear matrix inequalities and sum-of-squares optimisation.

2 Relaxed value iteration

Let $\boldsymbol{X}$, the set of states, and $\boldsymbol{U}$, the set of inputs, be arbitrary. Given f: $\boldsymbol{X} \times \boldsymbol{U} \to \boldsymbol{X}$, consider the dynamical system

$$x(k+1) = f(x(k), u(k)) \quad x(0) = x_0 \tag{1}$$

with $k = 0, 1, 2, \ldots$ Combining this with the control law μ: $\boldsymbol{X} \to \boldsymbol{U}$ gives the closed-loop dynamics

$$x(k+1) = f(x(k), \mu(x(k))) \tag{2}$$

To measure the performance of the system, we introduce a non-negative step cost l: $\boldsymbol{X} \times \boldsymbol{U} \to \boldsymbol{R}$ and define the value

IEE Proceedings online no. 20050094

doi:10.1049/ip-cta:20050094

Paper first received 17th March and in revised form 12th September 2005

The author is with the Department of Automatic Control, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden

The paper was written during a sabbatical at California Institute of Technology

E-mail: rantzer@control.lth.se

function

$$V_\mu(x_0) = \sum_{k=0}^{\infty} l(x(k), \mu(x(k)))$$

where x is given by (2).

The optimal cost function V^* is defined as

$$V^*(x_0) = \inf_\mu V_\mu(x_0)$$

and can be characterised as follows.

Proposition 1 (Dynamic programming [1]): Suppose that $\boldsymbol{V}: \boldsymbol{X} \to \boldsymbol{R}$ satisfies

$$0 \le \boldsymbol{V}(x) = \min_u [\boldsymbol{V}(f(x,u)) + l(x,u)] \quad \forall x \tag{3}$$

and $\lim_{k\to\infty} \boldsymbol{V}(x(k)) = 0$ for every $\{(x(k), u(k))\}_{j=1}^{\infty}$ with $\sum_{k=1}^{\infty} l(x(k), u(k)) < \infty$. Then $\boldsymbol{V} = V^*$ and the formula

$$\mu^*(x) = \operatorname{argmin}_u [V^*(f(x,u)) + l(x,u)] \tag{4}$$

defines an optimal control law.

Remark 1: Strictly speaking, the stated proposition makes sense only provided that the minimum with respect to u is attained. Although it is possible to give a modified statement without this assumption, we will keep it for simplicity throughout the paper.

Proof: Notice that for every solution to (1) the equality (3) implies that

$$\boldsymbol{V}(x(k)) \le \boldsymbol{V}(x(k+1)) + l(x(k), u(k))$$

As a consequence

$$\begin{aligned}\boldsymbol{V}(x(0)) - \boldsymbol{V}(x(T)) &= \sum_{k=0}^{T-1} [\boldsymbol{V}(x(k)) - \boldsymbol{V}(x(k+1))] \\ &\le \sum_{k=0}^{T-1} l(x(k), u(k))\end{aligned}$$

Taking limits as $T \to \infty$ on both sides implies that $\boldsymbol{V} \le V_\mu$ for every control law μ. Hence $\boldsymbol{V} \le V^*$. Moreover, the inequalities becomes equalities when $u(k) = \mu^*(x(k))$, so $\boldsymbol{V} = V_{\mu^*}$. This proves both that $\boldsymbol{V}$ is equal to the optimal cost and that μ^* is an optimal control law.

An iterative approach to solution of the Hamilton–Jacobi–Bellman equation (3) is known as value iteration. Next, we give a bound on the convergence rate of this scheme.

Proposition 2 (Value iteration convergence): Suppose the condition $0 \le V^*(f(x,u)) \le \gamma l(x,u)$ holds uniformly for some $\gamma < \infty$ and that $0 \le \eta V^* \le V_0^* \le \delta V^*$. Then the sequence defined iteratively by

$$V_{j+1}^* = \min_u [\boldsymbol{V}_j^*(f(x,u)) + l(x,u)] \quad j \ge 0 \tag{5}$$

approaches V^* according to the inequalities

$$\begin{aligned}\left[1 + \frac{\eta - 1}{(1+\gamma^{-1})^j}\right] V^*(x) &\le V_j^*(x) \\ &\le \left[1 + \frac{\delta - 1}{(1+\gamma^{-1})^j}\right] V^*(x)\end{aligned} \tag{6}$$

In particular, if $0 \le V_0^* \le V^*$, then

$$\left[1 - \frac{1}{(1+\gamma^{-1})^j}\right] V^*(x) \le V_j^*(x) \le V^*(x)$$

The proof is given in Section 7.

The main limiting factor in applications of value iteration is the complexity in computation and representation of the functions $V_j^*(x)$, except when $\boldsymbol{X}$ and $\boldsymbol{U}$ are finite sets of moderate size. Many schemes for approximation have therefore been developed. In this paper, we will use the following statement to quantify the effects of approximation errors in the Hamilton–Jacobi–Bellman equation.

Proposition 3 (Relaxed dynamic programming [12]): Suppose that $0 \le \alpha \le 1 \le \beta$ and let $\boldsymbol{V}: \boldsymbol{X} \to \boldsymbol{R}$ satisfy

$$\begin{aligned}\min_u \{\boldsymbol{V}(f(x,u)) + \alpha l(x,u)\} &\le \boldsymbol{V}(x) \\ &\le \min_u \{\boldsymbol{V}(f(x,u)) + \beta l(x,u)\}\end{aligned} \tag{7}$$

and $\lim_{k\to\infty} \boldsymbol{V}(x(k)) = 0$ for every $\{(x(k), u(k))\}_{k=1}^{\infty}$ with $\sum_{k=1}^{\infty} l(x(k), u(k)) < \infty$. Then

$$\alpha V^*(x) \le \boldsymbol{V}(x) \le \beta V^*(x) \quad \forall x$$

Moreover, the control law $\mu(x) = \arg\min_u [\boldsymbol{V}(f(x,u)) + \alpha l(x,u)]$ has a value function V_μ satisfying $\alpha V_\mu \le V$.

The proof is given in Section 7.

Solutions to the inequalities (7) can be found by relaxed value iteration:

Proposition 4 (Relaxed value iteration): Suppose that the sequences $\{V_j\}_{j=0}^{\infty}$ and $\{V_j^*\}_{j=0}^{\infty}$ start from the same $V_0 \equiv V_0^*$ and

$$\begin{aligned}\min_u \{V_j(f(x,u)) + \alpha l(x,u)\} &\le V_{j+1}(x) \\ &\le \min_u \{V_j(f(x,u)) + l(x,u)\}\end{aligned} \tag{8}$$

while V_j^* satisfies (5). Then $\alpha V_j^* \le V_j \le V_j^*$ for all j.

Proof: The statement follows by induction over j. □

Combining Proposition 4 with the convergence bound of Proposition 2, we get that the following bound on the distance from optimality.

Theorem 1: Given $0 \le \alpha \le 1$, suppose $0 \le V^*(f(x,u)) \le \gamma l(x,u)$ uniformly, $\gamma < \infty$ and that the sequence $V_0, V_1, V_2, \ldots$ starting with $0 \le V_0 \le V^*$ satisfies (8). Then

$$\alpha_j V^* \le V_j \le V^* \quad \alpha_j = [1 - (1+\gamma^{-1})^{-j}]\alpha \tag{9}$$

Moreover, the control policy $\mu_j(x) = \operatorname{argmin}_u \{V_j(f(x,u)) + \alpha l(x,u)\}$ gives a value function $V_{\mu_j}(x)$ satisfying

$$[\alpha + \gamma(\alpha_j - 1)] V_{\mu_j}(x) \le V^*(x) \tag{10}$$

Remark 2: The inequality (10) gives an upper bound on the cost function for the policy μ_j provided that the bracket in front of V_{μ_j} is positive. This will happen for large values of j, whenever $\alpha > \gamma/(1+\gamma)$.

Proof: The inequalities (9) follow directly from proposition 4 and proposition 2. Hence

$$V_j(f(x, \mu_j(x))) + \alpha l(x, \mu_j(x)) \le V_{j+1}(x) \le V^*(x)$$

Using that $\alpha_j V^* \le V_j$ and $V^*(f(x,u)) \le \gamma l(x,u)$, we get

$$\alpha_j V^*(f(x,\mu_j(x))) + \alpha l(x,\mu_j(x)) \le V^*(x)$$

$$V^*(f(x,\mu_j(x))) + [\alpha + \gamma(\alpha_j - 1)] l(x,\mu_j(x)) \le V^*(x)$$

For every trajectory of (1) with $u(k) = \mu_j(x(k))$, this implies

$$[\alpha + \gamma(\alpha_j - 1)] l(x,\mu_j(x)) \le [V^*(x(k)) - V^*(x(k+1))]$$

Summing over k gives (10) and the proof is complete. □

3 Iterations in a finite-dimensional subspace

When $\boldsymbol{X}$ has an infinite number of elements, the search for the optimal cost V^* is a search in an infinite-dimensional space. It is often natural to limit this search to a finite-dimensional subspace $\mathcal{L}$, for example polynomials of a fixed degree. A natural question to ask is whether existence of a solution to (7) in $\mathcal{L}$ has any implications on feasibility of the iterative inequalities (8). A striking result of this kind is given next, but for a slightly modified algorithm

Theorem 2: The conclusions of Theorem 1 remain valid if the conditions (8) are replaced by

$$\min_u \{V_j(f(x,u)) + \alpha l(x,u)\} \le V_{j+1}(x)$$

$$\le \min_u \{V_{j+1}(f(x,u)) + l(x,u)\} \qquad (11)$$

Proof: Every solution V_{j+1} to the right inequality in (11) must be bounded from above by V^* as shown in proposition 3. Moreover, the lower bound from proposition 4 remains valid with the same proof. The rest of the proof is identical to the proof of Theorem 1. □

Remark 3: Suppose that V^* has a simple approximation in the sense that $V^s \in \mathcal{L}$ satisfies

$$\min_u \{V^*(f(x,u)) + \alpha l(x,u)\} \le V^s(x)$$

$$\le \min_u \{V^s(f(x,u)) + l(x,u)\} \qquad (12)$$

Then, with $V_0 \equiv 0$, the iterative inequalities (11) define feasible convex conditions on $V_{j+1} \in \mathcal{L}$ at every step.

Remark 4: Time-varying linear quadratic optimal control problems, usually solved by Riccati equations, and shortest-path network problems solved by linear programming are two well-known special cases of our framework. One consequence of Theorem 2 is that other problems with an optimal cost function close to one of these special cases will be solvable with small computational effort.

Remark 5: Notice that the right-hand side of (12) is bounded from above by $\min_u\{V^*(f(x,u)) + l(x,u)\}$. Comparing this to the left-hand side shows that the only difference is the coefficient in front of $l(x,u)$. Hence the assumption (12) implicitly puts a constraint on the relative sizes of the cost in the next step $l(x,u)$ and the remaining cost $V^*(f(x,u))$. For optimal control problems with slow decay rate of the terms in the sum $\sum_k l(x(k),u(k))$ at optimality, this means that V^s needs to approximate V^* very accurately in order for the theorem to apply.

This observation has a natural interpretation in economic language. Let $V^*(x)$ be the value of a product with quality and location specified by x. The changes because of the business transaction u are given by $f(x,u)$. The transaction generates profit quantified by $l(x,u)$. The problem to maximise $\sum_k l(x,u)$ is then aimed to find the most profitable sequence of business transactions. In this context, the comparison of $l(x,u)$ and $V^*(f(x,u))$ says that small profit margins in each transaction increases the need for exact representation of the cost function at each step.

Remark 6: The difference between (8) and (11) is that in the second case, V_{j+1} appears also in the right-hand side, not just in the middle expression. This enables us to guarantee feasibility in every iteration. The condition (11) is slightly more complicated than (8) but is still a convex condition on V_{j+1}. A disadvantage in some applications is that the new condition leaves less room for distributed computations.

Combination of Theorem 2 with the previous bounds on value iteration convergence gives the following main result of the paper.

Theorem 3: Assume $0 \le V^*(f(x,u)) \le \gamma l(x,u)$ uniformly with $\gamma < \infty$. Let $\mathcal{L}$ be a linear space of functions $\boldsymbol{X} \to \boldsymbol{R}$. Suppose that there exists a $\boldsymbol{U} \in \mathcal{L}$ such that $(1-\epsilon)V^*(x) \le \boldsymbol{U}(x) \le V^*(x)$ where $0 \le \epsilon < (1+\gamma)^{-2}$. Then, with $V_0 \equiv 0$ and $\alpha = 1 - \epsilon(1+\gamma)^2$, the iterative convex inequalities (11) have a solution sequence V_0, V_1, $V_2 \cdots \in \mathcal{L}$ and the conclusions of Theorem 1 remain valid.

The proof is given in Section 7.

Remark 7: Combining this result with $\mathcal{L}$ as a set of polynomials and using the sum-of-squares technique [28, 29] for verification of the inequalities (11) gives a very general computational setting for optimal control. In this context, it is natural to apply the theorem with a modified interpretation of the inequalities, namely that the differences between left- and right-hand sides can be written as sums of squares.

In particular, the theorem proves an attractive feature of the algorithm defined by iteration of (11). This is that the computational effort in finding an approximately optimal control law (the polynomial degree needed in the relaxed value iteration) is related to the polynomial degree that is required to approximate the optimal cost. It also quantifies the accuracy of the outcome in terms of two fundamental parameters related to the difficulty of the problem, γ and ϵ.

4 Approximate policy iteration

Another iterative method to solve the Hamilton–Jacobi–Bellman equation (3) is known as policy iteration. Instead of keeping the value function V_j for the next iteration, the policy μ_j is kept. In many instances, this method requires fewer but more expensive iterations compared to value iteration. A detailed analysis will not be given here, just the following comparison.

Proposition 5 (Policy iteration convergence): Given a policy μ_0 with value function V_{μ_0}, consider the policy sequence defined iteratively by

$$\mu_{j+1}(x) = \operatorname{argmin}_u [V_{\mu_j}(f(x,u)) + l(x,u)] \quad j \ge 0 \qquad (13)$$

Let V_0^*, V_1^*, V_2^*, ... be defined by value iteration (5) with $V_0^* = V_{\mu_0}$. Then

$$V^*(x) \le V_{\mu_j}(x) \le V_j^*(x) \quad j \ge 1 \qquad (14)$$

Proof: Define $W_0 = V_{\mu_0}$ and

$$W_{j+1}(x) = W_j(f(x,\mu_1(x))) + l(x,\mu_1(x)) \quad j \ge 1$$

Then $W_1 \leq W_0$ by definition of μ_1 and the iteration gives $W_{(j+1)} \leq W_1$ for all j. Hence

$$V_{\mu_1}(x) = \lim_{j\to\infty} W_j(x) \leq W_1(x) = V_1^*(x)$$

This proves (14) for $j = 1$. Repeating the argument gives the general statement. □

Remark 8: Policy iteration can be viewed as an application of Newton's method for solving the Hamilton–Jacobi–Bellman equation. Hence fast convergence should be expected locally. Conditions for superlinear and quadratic convergence can be found in [21].

Remark 9: The initialisation, to find a policy μ_0 with finite cost W_0 is sometimes a non-trivial task.

Remark 10: For systems evolving on a graph, the computations of value iteration can often be parallelised, since the minimisation of (5) can be done for each node independently. In policy iteration, all nodes are usually tied together by (13) and parallelisation is more difficult.

Proposition 6 (Relaxed policy iteration): Given a policy μ_0, consider a sequence $(V_0, \mu_0), (V_1, \mu_1), (V_2, \mu_2), \ldots$ satisfying

$$V_j(x) \leq V_j(f(x, \mu_j(x))) + l(x, \mu_j(x))$$

$$\mu_{j+1}(x) = \operatorname*{argmin}_u [V_j(f(x, u)) + l(x, u)]$$

Define $V_0^*, V_1^*, V_2^*, \ldots$ by value iteration (5) with $V_0^* = V_{\mu_0}$. Then $V_j \leq V_j^*$ for all j.

Proof: By proposition 3, the inequality implies that $V_j \leq V_{\mu_j}$ for every j. Hence, the same argument as in the proof of proposition 5 gives that

$$V_1 \leq V_{\mu_1}(x) = \lim_{j\to\infty} W_j(x) \leq W_1(x) = V_1^*(x)$$

Repeating the argument gives the general statement. □

5 A model of switched linear systems

To concretise the results for switched linear systems, consider a graph defined by a set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$. A matrix $\boldsymbol{A}_{ij} \in \boldsymbol{R}^{n\times n}$ is assigned to each edge $(i, j) \in \mathcal{E}$. The state $x = (z, i)$ has two components, $z \in \boldsymbol{R}^n$ and $i \in \mathcal{N}$ and the system dynamics are

$$\begin{aligned} z(k+1) &= \boldsymbol{A}_{i(k)u(k)}z_{(k)} \quad z(0) = z_0 \\ i(k+1) &= u(k) \quad i(0) = i_0 \end{aligned} \tag{15}$$

Note that z evolves according to a linear equation defined by $\boldsymbol{A}_{ii}$ as long as the discrete state i remains constant. The role of the input u is to induce changes in the discrete state.

The step cost is defined by a set of matrices $\boldsymbol{Q}_{ij} \geq 0$ for $(i, j) \in \mathcal{E}$ such that

$$l((z, i), u) = z^T\boldsymbol{Q}_{iu}z$$

Thus, the cost is given by $\boldsymbol{Q}_{ii}$ when the discrete state i remains unchanged and by $\boldsymbol{Q}_{iu}$ when the step switches to u.

Taken together, this gives the following problem statement for switched linear systems

$$\text{Minimise} \sum_{k=0}^{\infty} z(k)^{\mathrm{T}}\boldsymbol{Q}_{i(k)u(k)}z(k) \quad \text{subject to (15)} \tag{16}$$

Example 1 (Shortest path problem): In this classical problem the objective is to find the shortest path to a given target node in a graph where each edge $(i, j) \in \mathcal{E}$ has an associated length q_{ij}. This problem is recovered in the setting above by letting z be a scalar, $\boldsymbol{A}_{ij} = 1$ for all (i, j) and $\boldsymbol{Q}_{ij} = q_{ij}$ for all $i \neq j$, whereas $\boldsymbol{Q}_{ii}$ is zero for the target node and strictly positive elsewhere.

Example 2 (Generalised shortest path problem): Again the problem is to find an optimal transportation path for goods to reach the target node. However, not only the distance matters. It is assumed that the quality of the goods changes during transportation. The continuous state vector $\boldsymbol{x}(k)$ measures the quality of the goods at time k. The matrix $\boldsymbol{A}_{ij}$ describes how the quality of the goods changes along edge (i, j). For example, the changes could be quality degradation because of transportation conditions, or quality improvements because of maintenance or upgrades. The problem (16) then describes the objective to find a path that allows for delivery at the target with optimal product quality.

Example 3 (Linear time-varying systems with quadratic cost): In the special case of a graph with only one path, that is, for every $i \in \mathcal{N}$ there is just one j with $(i, j) \in \mathcal{E}$, the cost function is a quadratic function $V^*(z, i) = z^{\mathrm{T}}\boldsymbol{P}^i z$ uniquely determined by the initial state. The Hamilton–Jacobi–Bellman equation then reduces to a time-varying Lyapunov equation

$$\boldsymbol{P}^k = \boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{P}^{k+1}\boldsymbol{A}_k + \boldsymbol{Q}_k$$

where $\boldsymbol{P}^k = \boldsymbol{P}^{i(k)}$, $\boldsymbol{A}_k = \boldsymbol{A}_{i(k)i(k+1)}$ and $\boldsymbol{Q}_k = \boldsymbol{Q}_{i(k)i(k+1)}$

Computation of the optimal control law for (16) is generally NP-hard. In fact, the classical travelling salesman problem is a special case:

Example 4 (Travelling salesman problem): A salesman is required to visit once and only once each of n different cities starting from a base city and returning to this city. What path minimises the total distance travelled by the salesman?

This problem can be modelled as a switching linear system with one node for each city. In particular $i = 1$ corresponds to the base city. A continuous state $z = (z_1, \ldots, z_n)$ is used to keep track of past visits. The matrices $\boldsymbol{A}_{ij}$ are defined by the following dynamics

$$\begin{cases} z_l(k+1) = z_l(k) - z_1(k) & \text{if } l = i(k) \in \{2, \ldots, n\} \\ z_l(k+1) = z_l(k) & \text{otherwise} \end{cases}$$

Let the initial state be $z_0 = (1, \ldots, 1)$. Define $q_{ij} = q_{ji}$ to be the distance between the cities i and j with $q_{ii} = 0$ for all i. Then, minimisation of the cost function

$$\sum_{k=1}^{\infty} z(k)^{\mathrm{T}}\boldsymbol{Q}_{i(k)i(k+1)}z(k) \quad \text{where } \boldsymbol{Q}_{ij} = \operatorname{diag}\{q_{ij}, 1, \ldots, 1\}$$

becomes equivalent to the travelling salesman problem. Every time a city is visited, the corresponding state variable z_i steps from 1 to 0. The state z_1 remains constant and equal to 1 all the time. It is easy to see that the cost becomes infinite unless the salesman first visits all cities once to get $z_l = 0$ for $l = 2, \ldots, n$, then stays in the base city. For such trajectories, the cost depends only on the total travelling distance.

Finally, the model can be modified by setting $A_{11} = (1 - \epsilon)I$ for some number ϵ. If ϵ is sufficiently small, this has no effect on the optimal trajectory for $z_0 = (1, \ldots, 1)$, but it makes it possible to get finite cost also for other initial states.

6 Computations for switched linear systems

Let us now specialise the results of Section 2 to the case of switched linear systems. Define

$$V^*(z_0, i_0) = \min_{u(0),u(1),\cdots} \sum_{l=1}^{\infty} z(l)^{\mathrm{T}} Q_{i(l)u(l)} z(l)$$

where the relationship between u, i and z is defined by the dynamics (15). Then the Hamilton–Jacobi–Bellman equation becomes

$$V^*(z, i) = \min_u \{V^*(A_{iu}z, u) + z^{\mathrm{T}} Q_{iu} z\} \tag{17}$$

For approximate solutions, a natural space $\mathcal{L}$ for a first approximation of the optimal cost is the space of quadratic forms $V(z, i) = z^{\mathrm{T}} P^i z$. For example, if $P^1, \ldots, P^m$ are symmetric matrices satisfying the matrix inequalities

$$P^i \le A_{iu}^{\mathrm{T}} P^u A_{iu} + Q_{iu} \quad \forall (i, u) \in \mathcal{E}$$

then Proposition 3 shows that $z^{\mathrm{T}} P^i z \le V^*(z, i)$ for every z, i.

With this parameterisation, the inequalities (11) can equivalently be written

$$\begin{aligned} &\min_u \{z^{\mathrm{T}} A_{iu}^{\mathrm{T}} P_j^u A_{iu} z + \alpha z^{\mathrm{T}} Q_{iu} z\} \\ &\quad \le z^{\mathrm{T}} P_{j+1}^i z \le z^{\mathrm{T}} A_{iu}^{\mathrm{T}} P_{j+1}^u A_{iu} z + z^T Q_{iv} z \end{aligned} \tag{18}$$

for all $z \in R^n$, $(i, v) \in \mathcal{E}$ and the minimisation is over all u with $(i, u) \in \mathcal{E}$. At each step of the iteration, these inequalities should be solved for the matrices $P_{j+1}^1, \ldots, P_{j+1}^m$. The second inequality reduces to standard linear matrix inequalities on the independent variables. The first inequality is also a convex constraint on P_{j+1}^i, but more cumbersome, since the minimum expression on the left-hand side does not have a simple representation.

A more conservative, but often useful, alternative to (18), is to instead require existence of scalar parameters $\theta_{j+1}^1, \ldots, \theta_{j+1}^m \ge 0$ with $\sum_{j=1}^m \theta_{j+1}^j = 1$ and such that

$$\sum_u \theta_{j+1}^u (A_{iu}^T P_j^u A_{iu} + \alpha Q_{iu}) \le P_{j+1}^i \le A_{iv}^T P_{j+1}^v A_{iv} + Q_{iv} \tag{19}$$

for all $(i, v) \in \mathcal{E}$. The parameters θ_{j+1}^i can be interpreted as the probabilities of a stochastic control law, which ignores the value of the continuous state z, hence the conservatism. The inequalities can be solved for θ_{j+1}^u and P_{j+1}^i by semi-definite programming in order to generate a sequence P_0^i, P_1^i, P_2^i, ... that converges to a solution of the inequalities

$$\sum_u \theta^u (A_{iu}^{\mathrm{T}} P^u A_{iu} + \alpha Q_{iu}) \le P^i \le A_{iv}^T P^v A_{iv} + Q_{iv} \tag{20}$$

for all $(i, v) \in \mathcal{E}$. A precise statement is given in the following corollary, stated similarly to Theorem 3.

Corollary 1: Assume $V^*(A_{iu}z, u) \le \gamma z^{\mathrm{T}} Q_{iu} z$ for all z, i, u. Suppose there exist matrices $P^1, \ldots, P^m$ such that

$$(1 - \epsilon) V^*(z, i) \le z^{\mathrm{T}} P^i z \le V^*(z, i) \quad 0 \le \epsilon \le (1 + \gamma)^{-2}$$

Let $\alpha = 1 - \epsilon(1 + \gamma)^2$. Then, with $P_0^i = 0$ for $i \in \mathcal{N}$, the iterative convex inequalities (19) have solutions P_{j+1}^i and θ_{j+1}^u for every $j \ge 0$. All such solutions generate approximations to the optimal cost according to the inequalities

$$\alpha_j V^*(z, i) \le z^{\mathrm{T}} P_j^i z \le V^*(z, i) \quad \alpha_j = [1 - (1 + \gamma^{-1})^{-j}]\alpha$$

Moreover, the control law $\mu_j(z, i) = \mathrm{argmin}_u z^{\mathrm{T}}(A_{iu}^{\mathrm{T}} P_j^u A_{iu} + \alpha_j Q_{iu})z$ defines a control law value function V_{μ_j} satisfying $[\alpha + \gamma(1 - \alpha_j)]V_{\mu_j} \le V^*$.

Remark 11: In general (19) is significantly more conservative than (18), but equivalence holds for example if the sum on the left has only two terms, that is, if there are only two options for u at every switch instance.

If instead policy iteration is used analogously for the same problem, the iterative conditions (19) are replaced by

$$\sum_u \theta_j^u (A_{iu}^{\mathrm{T}} P_{j+1}^u A_{iu} + \alpha Q_{iu}) \le P_{j+1}^i \le A_{iu}^{\mathrm{T}} P_{j+1}^v A_{iu} + Q_{iv} \tag{21}$$

However, no analogy of theorem 2 and theorem 3 is available for policy iteration.

Let us conclude the section with a major computational example to demonstrate the power of the proposed algorithms.

Example 5: First we generate a graph by randomly distributing 60 nodes in a square and defining edges by assigning two possible jumps from each node. The resulting graph is shown in Fig. 1.

We will use 30 continuous states in each node. The step costs are chosen as

$$Q_{ij} = d_{ij} I$$

where d_{ij} is the distance between two nodes. The dynamics, defined by the matrices A_{ij} will be chosen randomly, but with significant restrictions. Recall that if A_{ij} are all equal to the identity, then we recover the shortest-path-problem (provided that there is 'target node' where it is possible to stay with step cost zero). The value iteration then works without need for approximation. Similarly, if the A_{ij} are very small, then the cost function is essentially determined by the cost of the first step, and therefore close to quadratic. Relaxed value iteration will then work well with quadratic approximations.

We will consider a case somewhere in between these two extremes. Each A_{ij} matrix is randomly generated, but with a

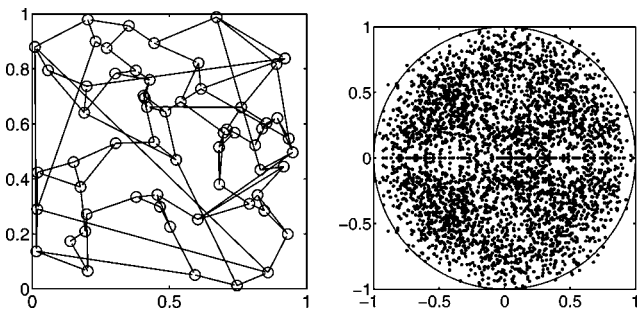

Fig. 1 *Graph with 60 nodes has been randomly generated*

From each node, there are two edges defining possible switches. For each of the 120 edges, a 30×30 matrix A_{ij} is used to define the dynamics of the continuous states along that edge. The coefficients of each matrix are randomly generated, but the matrix is scaled to have eigenvalues in a prespecified disc of diameter 0.5. To the right, all eigenvalues of the 120 A_{ij} matrices are shown in one plot

spectrum varying within a disc of diameter 0.5 arbitrarily positioned with a centre at most 0.9 from the origin. As a consequence, some of the matrices have eigenvalues outside the unit disc and are therefore expanding the continuous state in some directions. See the eigenvalue plot in Fig. 1. Once the graph and matrices $\boldsymbol{Q}_{ij}$ and $\boldsymbol{A}_{ij}$ are defined, we are ready to run the value iteration algorithm. In each iteration let α_j be the maximal value of α for which (19) holds and let α^j be the maximal number of α for which the resulting $\boldsymbol{P}_j^i$ also satisfy (20). We then get the sequence

$$\begin{aligned} \alpha_1 &= 0.58 \qquad \alpha^1 = -7.12 \\ \alpha_2 &= 0.34 \qquad \alpha^2 = -4.13 \\ \alpha_3 &= 0.28 \qquad \alpha^3 = -0.42 \\ \alpha_4 &= 0.29 \qquad \alpha^4 = 0.26 \end{aligned}$$

Hence, after only four value iterations, we have found a quadratic approximation to the optimal cost satisfying

$$0.26V^*(z, i) \leq z^{\mathrm{T}}\boldsymbol{P}^i z \leq V^*(z, i) \quad \forall x, i \tag{22}$$

and the corresponding control law yields a cost which is necessarily within a factor 4 from optimality

$$V^*(z_0, i_0) \leq \sum_k z(k)^{\mathrm{T}}\boldsymbol{Q}_{i(k)u(k)}z(k) \leq \frac{1}{0.26}V^*(z_0, i_0)$$

It is interesting to look closer at some details of the solution. It turns out, as indicated in Fig. 2, that in most of the nodes the inequalities (22) actually hold with a much higher value of α than 0.26. These are usually the nodes where one jump direction is clearly preferable to the other, regardless of the continuous state. Compare to Fig. 3.

A natural step for refinement would therefore be to increase the accuracy in the computations at the bottleneck nodes, that is, where inequalities (22) are tight. One way to improve the accuracy is to use a less conservative condition than (19) to enforce the inequalities (18). Another way is to introduce higher degree polynomials in the search for approximations to the optimal cost $V^*(z, i)$.

The source files of this example are available on the web site [30].

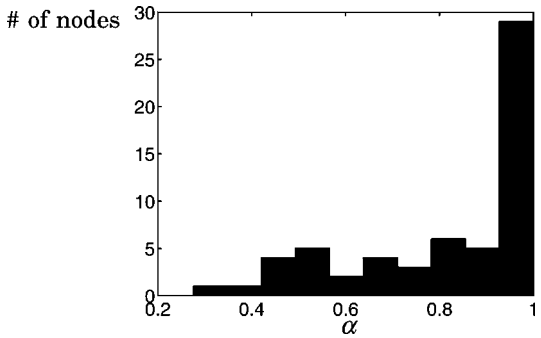

Fig. 2 *For each node, the Hamilton–Jacobi–Bellman equation needs a certain amount of relaxation to be satisfied*

This histogram reflects the fact that in most nodes of the graph, the equation can be satisfied with α around 0.9, much better than what is indicated by the worst case value $\alpha = 0.26$

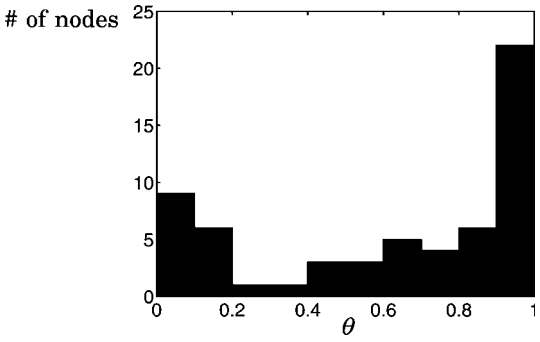

Fig. 3 *For each node, there is a number θ^j, which appears in the left hand side of (20) and indicates the optimal switch*

The histogram over the θ-values shows a preference for $\theta = 1$, which corresponds to switching to the nearest node in the graph. This is natural, since the nearest node has lowest step cost. Values between 0 and 1 can be interpreted as probabilities for jumps in different directions

7 Proofs

Proof of Proposition 2: The assumption $0 \leq V^*(f(x, u)) \leq \gamma l(x, u)$ gives

$$\begin{aligned} V_1^*(x) &= \min_u [V_0^*(f(x, u)) + l(x, u)] \\ &\geq \min_u [\eta V^*(f(x, u)) + l(x, u)] \\ &\geq \min_u \left[\left(\eta + \frac{1-\eta}{\gamma+1} \right) V^*(f(x, u)) \right. \\ &\quad \left. + \left(1 - \gamma\frac{1-\eta}{\gamma+1} \right) l(x, u) \right] \\ &= \frac{\eta\gamma+1}{\gamma+1} \min_u [V^*(f(x, u)) + l(x, u)] \\ &= \frac{\eta\gamma+1}{\gamma+1} V^*(x) \end{aligned}$$

The lower bound in (6) is obtained by repeating the argument j times.

Similarly

$$\begin{aligned} V_1^*(x) &= \min_u [V_0^*(f(x, u)) + l(x, u)] \\ &\leq \min_u [\delta V^*(f(x, u)) + l(x, u)] \\ &\leq \min_u \left[\left(\delta - \frac{\delta-1}{\gamma+1} \right) V^*(f(x, u)) \right. \\ &\quad \left. + \left(1 + \gamma\frac{\delta-1}{\gamma+1} \right) l(x, u) \right] \\ &= \frac{\delta\gamma+1}{\gamma+1} \min_u [V^*(f(x, u)) + l(x, u)] \\ &= \frac{\delta\gamma+1}{\gamma+1} V^*(x) \end{aligned}$$

and the upper bound in (6) is obtained by repeating j times. □

Proof of Proposition 3: For every solution to (1) the right inequality (7) implies

$$\boldsymbol{V}(x(k)) - \boldsymbol{V}(x(k+1)) \leq \beta l(x(k), u(k))$$

Summing over k gives

$$\boldsymbol{V}(x(0)) - \boldsymbol{V}(x(T)) \leq \beta \sum_{k=0}^{T-1} l(x(k), u(k))$$

Taking limits as $T \to \infty$ shows that $\boldsymbol{V}$ is a lower bound on βV_μ for every control law μ. Hence $\boldsymbol{V} \leq \beta V^*$.

Similarly, when $u(k) = \mu(x(k))$, the left inequality becomes

$$\alpha l(x(k), u(k)) \leq \boldsymbol{V}(x(k)) - \boldsymbol{V}(x(k+1))$$

and summing over k gives

$$\alpha \sum_{k=0}^{T-1} l(x(k), u(k)) \leq \boldsymbol{V}(x(0)) - \boldsymbol{V}(x(T)) \leq \boldsymbol{V}(x(0))$$

This shows that $\boldsymbol{V}$ is an upper bound on αV_μ. Hence

$$\alpha V^* \leq \alpha V_\mu \leq \boldsymbol{V} \leq \beta V^*$$

and the proof is complete. □

Proof of Theorem 3: Define $V^s := (1 - \epsilon\gamma)\boldsymbol{U}$. Repeating the argument of Proposition 2, we have

$$\begin{aligned}
&\min_u\{V^s(f(x,u)) + l(x,u)\} \\
&\quad = \min_u[(1-\epsilon\gamma)\boldsymbol{U}(f(x,u)) + l(x,u)] \\
&\quad \geq \min_u[(1-\epsilon\gamma)(1-\epsilon)V^*(f(x,u)) + l(x,u)] \\
&\quad \geq \min_u[((1-\epsilon\gamma)(1-\epsilon)+\epsilon)V^*(f(x,u)) \\
&\qquad + (1-\epsilon\gamma)l(x,u)] \\
&\quad \geq (1-\epsilon\gamma)\min_u[V^*(f(x,u)) + l(x,u)] \\
&\quad = (1-\epsilon\gamma)V^*(x) \geq (1-\epsilon\gamma)\boldsymbol{U}(x) = V^s(x)
\end{aligned}$$

This proves the right inequality in (12). Similarly

$$\begin{aligned}
&\min_u[V^*(f(x,u)) + \alpha l(x,u)] \\
&\quad \leq \min_u([1-\epsilon(1+\gamma)]V^*(f(x,u)) \\
&\qquad + [\alpha + \epsilon(1+\gamma)\gamma]l(x,u)) \\
&\quad = [1-\epsilon(1+\gamma)]\min_u[V^*(f(x,u)) + l(x,u)] \\
&\quad = [1-\epsilon(1+\gamma)]V^*(x) \\
&\quad \leq (1-\epsilon\gamma)(1-\epsilon)V^*(x) \\
&\quad \leq (1-\epsilon\gamma)U(x) = V^s(x)
\end{aligned}$$

which proves the left inequality in (12). Hence, the convex constraints (11) on $\boldsymbol{V}_{j+1}$ are feasible at every step and the desired conclusions follow from theorem 2. □

8 Conclusions

The main conclusion in this paper, as expressed in theorem 3, is that finding approximately optimal control laws requires complex computations only if the cost function is complex.

Algorithms for control synthesis should therefore be designed to take advantage of this fact. They should give a simple answer quickly whenever there is one, and enter into more involved computations only when simpler alternatives have been exhausted.

Let us finally remark that although example 5 was generated randomly within some restrictions, those restrictions were indeed essential. For a vast majority of problems in the class defined in Section 5, quadratic approximations of the optimal cost will most likely not be sufficient for convergence of the value iteration. Higher-order polynomials will increase the computational burden significantly, but the decentralised nature of the iteration should still leave room for a considerable number of continuous states.

9 Acknowledgments

The author is grateful to many colleagues for comments on this work, in particular the PhD students Peter Alriksson, Bo Lincoln, Ritesh Madan and Andreas Wernrud. The research was supported by the European Commission through grant IST-2001-33520 and the HYCON Network of Excellence. Much of the writing was done during a sabbatical supported by the Swedish Foundation of Strategic Research. An excellent sabbatical environment was provided by Control and Dynamical Systems at Caltech.

10 References

1 Bellman, R.E.: 'Dynamic programming' (Princeton University Press, 1957)
2 Leake, R.J., and Liu, R.-W.: 'Construction of suboptimal control sequences', *SIAM J. Contr.*, 1967, **5**, (1), pp. 54–63
3 Kantorovich, L.V.: 'On a problem of Monge', *Uspekhi Mat. Nauk.*, 1948, **3**, pp. 225–226
4 Vinter, R.: 'Convex duality and nonlinear optimal control', *SIAM J. Contr. Optim.*, 1993, **31**, (2), pp. 518–538
5 Rachev, S., and Rüschendorf, L.: 'Mass transporation problems. Volume I: Theory, probability and its applications' (Springer, 1998)
6 Rantzer, A.: 'A dual to Lyapunov's stability theorem', *Syst. Contr. Lett.*, 2001, **42**, (3), pp. 161–168
7 Rantzer, A., and Hedlund, S.: 'Duality between cost and density in optimal control'. Proc. 42nd IEEE Conference on Decision and Control, 2003
8 Rubner, Y., Tomasi, C., and Guibas, L.J.: 'A metric for distributions with applications to image databases'. Proc. 1998 IEEE Int. Conf. on Computer Vision, Bombay, India, 1998
9 Rantzer, A., and Johansson, M.: 'Piecewise linear quadratic optimal control', *IEEE Trans. Autom. Contr.*, 2000, **45**, (4), pp. 629–637
10 Hedlund, S., and Rantzer, A.: 'Convex dynamic programming for hybrid systems', *IEEE Trans. Autom. Contr.*, 2002, **47**, (9), pp. 1536–1540
11 Lincoln, B., and Rantzer, A.: 'Suboptimal dynamic programming with error bounds'. Proc. 41st Conference on Decision and Control, December 2002
12 Lincoln, B., and Rantzer, A.: 'Relaxing dynamic programming', 2003, (accepted for publication)
13 Puterman, M.L., and Brumelle, S.L.: 'On the convergence of policy iteration in stationary dynamic programming', *Math. Oper. Res.*, 1979, **4**, (1), pp. 60–69
14 Santos, M.S., and Rust, J.: 'Convergence properties of policy iteration', *SIAM J. Contr. Optim.*, 2004, **42**, (6), pp. 2094–2115
15 Falcone, M.: 'A numerical approach to the infinite horizon problem of deterministic control theory', *Appl. Math. Optim.*, 1987, **15**, pp. 1–13 Corrigenda, 1991, **23** pp. 213–214
16 Kushner, H.J., and Dupuis, P.G.: 'Numerical methods for stochastic control problems in continuous time' (Springer-Verlag, New York, 1992)
17 Gonzales, R., and Rofman, E.: 'On deterministic control problems: an approximation procedure for the optimal cost. I. The stationary problem', *SIAM J. Contr. Optim.*, 1985, **23**, pp. 242–266
18 Souganidis, P.E.: 'Approximation schemes for viscosity solutions of Hamilton–Jacobi equations', *J. Diff. Equat.*, 1985, **59**, pp. 1–43
19 Bertsekas, D.P.: 'Dynamic programming and optimal control' (Athena Scientific, 2001, 2nd edn.)
20 Tsitsiklis, J.N., and Van Roy, B.: 'Feature-based methods for large scale dynamic programming', *Mach. Learning*, 1996, **22**, pp. 59–94
21 de Farias, D.P., and Van Roy, B.: 'The linear programming approach to approximate dynamic programming', *Oper. Res.*, 2003, **51**, (6), pp. 850–865

22 Beard, R., Saridis, G., and Wen, J.: 'Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation', *J. Optim. Theory Appl.*, 1998, **96**, (3), pp. 589–626
23 Lu, W.-M., and Doyle, J.C.: 'Robustness analysis and synthesis for nonlinear uncertain systems', *IEEE Trans. Autom. Contr.*, 1997, **42**, (12), pp. 1654–1662
24 Lygeros, J., Tomlin, C., and Sastry, S.: 'Controllers for reachability specifications for hybrid systems', *Automatica*, 1999, **35**, (3), pp. 349–370
25 Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M.: 'Constrained model predictive control: stability and optimality', *Automatica*, 2000, **36**, (6), pp. 789–814
26 Borrelli, F., Baotic, M., Bemporad, A., and Morari, M.: 'An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems'. American Control Conference, Denver, Colorado, USA, 2003, pp. 4717–4722
27 Blondel, V.D., and Nesterov, Y.: 'Computationally efficient approximations of the joint spectral radius', *SIAM J. Matrix Anal.*, 2005, **27**, (1), pp. 256–272
28 Parrilo, P.A.: 'Semidefinite programming relaxations for semi-algebraic problems', *Math. Program. Ser. B*, 2003, **96**, (2), pp. 293–320
29 Prajna, S., Papachristodoulou, A., and Parrilo, P.A.: 'Introducing SOS-tools: a general purpose sum of squares programming solver'. Proc. 41st IEEE Conference on Decision and Control, Las Vegas, USA, 2002
30 http://www.control.lth.se/articles/article.pike?artkey=ran05.