Stabilizing Dynamic State Feedback Controller Synthesis: A Reinforcement Learning Approach

Miguel A. SOLIS[1], Manuel OLIVARES[2], Héctor ALLENDE[1]

[1] Departamento de Informática,
Universidad Técnica Federico Santa María, Chile.

[2] Departamento de Electrónica,
Universidad Técnica Federico Santa María, Chile.

Abstract: State feedback controllers are appealing due to their structural simplicity. Nevertheless, when stabilizing a given plant, dynamics of this type of controllers could lead the static feedback gain to take higher values than desired. On the other hand, a dynamic state feedback controller is capable of achieving the same or even better performance by introducing additional parameters into the model to be designed. In this document, the Linear Quadratic Tracking problem will be tackled using a (linear) dynamic state feedback controller, whose parameters will be chosen by means of applying reinforcement learning techniques, which have been proved to be especially useful when the model of the plant to be controlled is unknown or inaccurate.

Keywords: Adaptive control, furuta pendulum, reinforcement learning.

1. Introduction

Reinforcement learning (RL) is typically concerned about solving sequential decision problems modelled by Markov Decision Processes (MDPs). Applications of RL, as a research field inside of Machine Learning, has got extended to areas such as Robotics [10] or Control Theory [7, 12, 16], by means of choosing a suitable representation of the problem to be solved.

The first RL applications on control systems have been found on Werbos [19, 20], where the regulation problem was tackled, whose objective is to design a controller for a given process, such that the internal state of this process approaches zero as time increases unbounded. Then, an immediate extension was to apply policy iteration (PI) algorithms to solve the linear quadratic regulator (LQR) problem [4].

The LQR, i.e., the regulator problem when the system is assumed to be linear, and the performance index is given in terms of a quadratic function [1] is particularly appealing given that its solution is obtained by solving an algebraic Riccati equation (ARE). Then, PI algorithms basically start with an admissible control policy and then iterate between policy evaluation and policy improvement steps until variations on the policy or the specified value function are negligible, as seen on [4, 13, 17].

In the other hand, the linear quadratic tracking (LQT) problem also assumes a linear model for the process dynamics and a quadratic function for the performance index, but the main objective is to design a controller such that the measured output of the process to be controlled, follows an exogenous reference signal, so the LQR could be considered as a particular case of the LQT problem. Although, as mentioned before, RL algorithms have been extensively applied for solving the LQR problem, the LQT has not received much attention on the literature mainly because for most reference signals the infinite horizon cost becomes unbounded [2]. Work in [14] tackles the problem on the continuous time domain by solving an augmented ARE obtained from the original system dynamics and the reference trajectory dynamics, while [9] takes a similar approach for the discrete-time case, where a Q-learning algorithm is obtained for solving the LQT problem without any model knowledge.

Then, when considering noisy systems, the performance index and notions of stability have to be modified accordingly. This problem has been extensively treated on literature from the classical control, or model-based approach [6, 8, 21], unlike on the learning paradigm. Work on [11] uses neural networks for reducing calculus efforts on providing optimal control for the stochastic LQR, while other works focus on relaxing assumptions on the ARE under different scenarios, but still requiring knowledge of the system dynamics [5, 22]. The work on [9] could be considered as the closest to our approach, given the LQT setup and the absence of model knowledge. Nevertheless,

unlike the work therein, we consider the stochastic LQT problem, and we extend the structure of the (linear) state feedback controller to be of a more general form. Then, when analysing experimental results, RL will prove to be especially useful for the case when the model is unknown, but it is still useful when dynamics are assumed to be given, since hand-tuning of controller parameters could represent a time consuming task due to the number of degrees of freedom and the corresponding constraints.

The remainder of this document is organized as follows: Section 2 presents a brief review about the basic concepts to be used in the subsequent sections, as well as the classical approach for the LQT problem by using (static) state feedback controllers. Then, Section 3 shows the appropriate procedure for obtaining a stabilizing dynamic controller for minimizing the LQT performance criteria, and the main results. Section 4 makes an illustration on simulation results obtained for an arbitrary plant. Finally, Section 5 draw some final conclusions and give some insight into future work.

Notation:

$\xi\{\cdot\}$ denotes the expectation operator, $\acute{\lambda}(M)$ is used to describe the largest eigenvalue of M, while M^T denotes the transpose of matrix M, and M^{-1} its inverse when M is square.

R stand for the set of all the real numbers, and when used with superscripts R^n (or $R^{n\times m}$) describe a vector (or matrix) with n rows (or n rows and m columns) whose elements are real-valued.

2. Background

2.1 Reinforcement learning

In simple terms, the main objective of RL is to optimize the expected long-term reward, on an (initially) unknown environment through finding an optimal sequence of actions to take for a given problem.

Definition 1

An RL problem, depicted by a MDP is given in terms of the tuple (S,A,T,R):

- S: This is the set of all possible states on a given time step.
- A: denote the set of actions the agent could take.
- $T: S\times A\times S\rightarrow[0,1]$ correspond to the state transition function, which is assumed to be unknown and quantifies the probability of the agent being transferred from state s to state s' by executing action a.
- $R: S\times A\rightarrow R$ is the reward function, whose values are real and scalar.
- $\pi: S\rightarrow A$ is a map from states to actions, and describes the policy (decides which action to take on a given state).

As a result of the choices made by policy π, its quality is quantified by the value function $V^{\pi}(s)$, with the agent being on state s at time step k. Then, the value function corresponds to the expected (discounted) accumulated reward with initial state s:

$$V^{\pi}(s)=\xi\left\{\sum_{i=0}^{\infty}\gamma^{i}r_{k+1}\vee\pi\right\}, \tag{1}$$

where $\gamma\in[0,1]$ corresponds to the discount factor.

Then, a policy π^* is said to be optimal, if the following expression holds,

$$V^{\pi^*}(s)\geq V^{\pi}(s)\forall\, s,\pi \tag{2}$$

i.e., the optimal policy [illegible] will yield the biggest value function with respect to every possible policy π, independently of initial state s. Although there could exist more than one optimal policy, its optimal value is unique [3] and may be obtained by means of solving the Bellman optimality equation

$$V^{\pi^*}(s)=\max_{a\in A}\sum_{s\in S}Pr\left[(s,a,s')\right]\cdot \cdot\left(R(s,a)+\gamma V^{\pi^*}(s')\right) \tag{3}$$

where $Pr\left[(s,a,s')\right]$ is the probability of being led to state s' when action a is executed on state s. Then, one of the most popular and convergent in probability, dynamic programming algorithms, is known as policy iteration (PI), whose procedural form is shown on Algorithm 1, which is concerned with policy improvement. PI starts with some initial and randomly chosen policy. Then, at each successive iteration, the algorithm evaluates the current policy with its value function, and performs an improvement step where a new policy is obtained by means of performing greedy actions at each state over the current policy.

PI corresponds to a Dynamic Programming (DP) algorithm, and besides value iteration (VI) is the basis of many successful Reinforcement Learning. However, in the latter setting (RL) the learning process is usually assumed to be generated from the current interaction between the agent and its environment, with real or simulated data.

State-action values (also known as *Q*-values) are introduced in order to solve the problem of lack of knowledge on the transition model, which is involved on the policy improvement step for any value estimation algorithm.

For a given policy π, the tuple (*s*,*a*) yields the state-action value *Q*(*s*,*a*), which is the (expected) discounted return over all trajectories assuming *a* is executed on state *s*, according policy π. There is a relation between the optimal state value V^{π^*} and its corresponding optimal state-action value Q^{π^*}, and is given by

$$V^{\pi^*}(s)=\max_{a\in A} Q^{\pi^*}(s,a) \tag{4}$$

and a representative algorithm for state-action value estimation is Q-learning [18], which can be viewed as an asynchronous, stochastic version of VI.

2.2 State Feedback Control

When tackling the problem of making a system follow a given trajectory, the main objective is to find the appropriate control signals (or actions that the controller should generate) for making a variable of the system to be controlled to keep track of the desired reference value (Algorithm 1).

Then, according to notation introduced on Figure 1, for the state-feedback scheme, a control signal *u*[*k*] must be generated, and used as input of the controlled system, which has an output *y*[*k*] that should keep track of the reference *r*[*k*] at time *k*. Therefore, a model is needed.

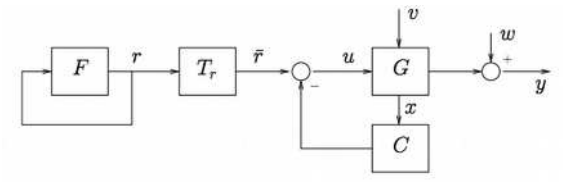

Figure 1. State feedback control scheme

One of the typical options is to assume a given structure for the model, and then tune its parameters until the model and real system dynamics matches. Another option, could be to use physical laws for building a model and set relations between all the variables of the system.

Algorithm 1 Policy Iteration

$\hat{\pi}^{(0)}(s)$: some random action $\quad \forall s \in \mathcal{S}$
$\hat{V}^{(0)}(s) = 0 \quad \forall s \in \mathcal{S}$
$i = 0$
(Policy Evaluation)
while $\Delta > \epsilon$ **do**
  for each $s \in \mathcal{S}$ **do**
    $\hat{V}^{(i+1)}(s) = \max_{a\in\mathcal{A}} Pr\{(s,a,s')\} \cdot (R(s,a) + \gamma V(s'))$
  end for
  $\Delta = \|\hat{V}^{(i+1)} - \hat{V}^{(i)}\|$
  $i = i + 1$
end while
$policy\_stable = true$
$i = 0$
(Policy Improvement)
for each $s \in \mathcal{S}$ **do**
  $\hat{\pi}^{(i+1)}(s) : argmax_{a\in\mathcal{A}} Pr\{(s,a,s')\} \cdot (R(s,a) + \gamma V(s'))$
  if $\hat{\pi}^{(i)}(s) \neq \hat{\pi}^{(i+1)}(s)$ **then**
    $policy\_stable = false$
  end if
end for
if *policy_stable* **then**
  return policy $\hat{\pi}^{(i)}$
else
  go to line 3
end if

Definition 2

For a (stochastic) linear, discrete-time and strictly causal systems, the state space representation of the process to be controlled, *G* on Figure 1, is given by

$$x[k+1]=Ax[k]+Bu[k]+v[k], \tag{5a}$$

$$y[k]=Cx[k]+w[k] \tag{5b}$$

where $x[k]\in R^{n_x}$, $y[k]\in R^{n_y}$ and $u[k]\in R^{n_u}$ corresponds to the internal state, measured output and control signal respectively at time *k*, and *A, B, C* and *D* are (usually) known matrices of appropriate dimensions, while *v*[*k*] and *w*[*k*] are uncorrelated (gaussian) zero-mean white noises, namely process and measurement noise with constant variance P_v and P_w respectively.

When considering a (static) linear state feedback controller, *C* on Figure 1, according to notation therein introduced, the control law would be given by

$$u[k]=\acute{r}[k]-Lx[k] \tag{6}$$

where $L \in R^{n_u \times n_x}$ stands for the feedback gain, and T_r is a pre-filter such that $y[k]$ gets as close as possible to $r[k]$.

Then, the performance index for the stochastic infinite-horizon LQT problem at time k will be given by

$$J[k] = \\ = \xi\left\{\sum_{i=k}^{\infty}(r[i]-y[i])^T Q(r[i]-y[i]) + u^T[i]Ru[i]\right\} \quad (7)$$

with $Q>0$ and $R\geq 0$ weighting matrices of appropriate dimensions. Since we are dealing with stochastic systems, an appropiate notion of stability is given by mean-square stability [15], on the following Lemma.

Lemma 1

The process given by (5), whose state is given by $x[k]$ at time k, will be mean-square stable (MSS) if and only if

$$\lim_{k\to\infty}\left|\xi\{x[k]x^T[k]\}\right| < \infty \quad (8)$$

independently of the initial state $x[0]=x_0$.

Moreover, the controller (6) stabilizes the system on (5), if the reference $r[k]$ decays asymptotically to zero, and L is such that the closed-loop eigenvalues are inside the unit circle, i.e.,

$$\lim_{k\to\infty} r[k] = 0 , \quad (9a)$$

$$\left|\acute{\lambda}(A-BL)\right| < 1 . \quad (9b)$$

Proof:

For the definition of mean square stability, the reader is encouraged to see [15]. Although the conditions set on Lemma 1 can be found on standard stochastic control theory literature, we show how these conditions are obtained for sake of clarity.

By replacing (6) on (5), the second order moments matrix of $M_x[k]=\xi\{x[k]\cdot x^T[k]\}$ is given by

$$M_x[k] = (A-BL)M_x[k-1](A-BL)^T + \\ + BM_{\acute{r}}[k-1]B^T + P_v \quad (10)$$

with P_v the variance of process noise as on Definition 2, and $M_{\acute{r}}[k]=\xi\{\acute{r}[k]\acute{r}^T[k]\}$.

Then, in terms of the initial state $x[0]=x_0$,

$$M_x[k] = (A-BL)^k M_x[0](A-BL)^{k,T} + \\ + \sum_{i=1}^{k}(A-BL)^{i-1}(BM_{\acute{r}}[k-i]B^T + P_v)(A-BL)^{i-1,T}$$

where it can be seen that for the system being MSS, is necessary to get $M_{\acute{r}}[k]$ bounded as k grows to infinity, so $\acute{r}[k]$, and therefore $r[k]$ has to decay asymptotically. Since the factor $(A-BL)$ is part of a matrix power series, its spectral radium has to be less than unit, which directly involves

$$\left|\acute{\lambda}(A-BL)\right| < 1 . \quad (11)$$

□ □ □

The asymptotically decaying assumption on reference r limits the class of trajectories to be used, and more important, sense of minimality in (7) is lost. Therefore, as on [9], the following section will introduce a discounted performance index for the LQT setup, and assuming the reference is being generated by a system F, as depicted on Figure 1.

3. Linear Quadratic Tracking

3.1 Problem Formulation

In this section, we will expand the class of state feedback controllers on the LQT problem, allowing it to have its own dynamics component on the resulting control signal, but still on the realm of linear controllers. Then, considering Figure 1, the controller for the system on (5) would be given by

$$x_c[k+1] = A_c x_c[k] + B_c x[k], \quad (12a)$$

$$u[k] = \bar{r}[k] - (C_c x_c[k] + D_c x[k]), \quad (12b)$$

where the c subscript is set to stress the difference between matrices (A, B, C) from the plant model and (A_c, B_c, C_c, D_c) from the controller, as well as the state of the plant, $x[k]$, and the internal state of the controller itself, $x_c[k]$.

Note that once the controller has been designed, the prefilter T_r for transforming reference $r[k]$ into $\acute{r}[k]$ should be chosen such that the transfer function from $r[k]$ to $y[k]$ is unitary, in order to ensure stationary tracking.

Remark 1. It can be seen from (12), that when $C_c=0$ we have exactly the same state-feedback law as in (6), with D_c and L being equivalent. When this is the case, we could still have a dynamic controller, but the dynamics

(and hence stability) of the controller itself doesn't play any role on the stability of the control loop, allowing the controller to be unstable, as long it stabilizes the plant.

Then, Lemma 2 set conditions for both the system on (5) and the controller on (12) to be MSS.

Lemma 2

Consider process (5) whose input is given by the state feedback law

$$u[k] = \bar{r}[k] - (C_c x_c[k] + D_c x[k]), \qquad (13)$$

with $x_c[k]$ being as on (12). Then, the closed loop and the controller itself will be mean square stable, if

$$\lim_{k\to\infty} r[k] = 0, \qquad (14a)$$

$$|\bar{\lambda}(A_a)| < 1, \qquad (14b)$$

with A_a being a block matrix given by

$$A_a = \begin{bmatrix} A - BD_c & -BC_c \\ B_c & A_c \end{bmatrix}. \qquad (15)$$

Proof:

Let $x_a[k]$ be the augmented state vector, at time k, such that

$$x_a[k] = \begin{bmatrix} x^\intercal[k] & x_c^\intercal[k] \end{bmatrix}^\intercal.$$

From (5) and (12), we have

$$x_a[k+1] = A_a x_a[k] + B_a r[k] + v_a[k], \qquad (16)$$

with

$$A_a = \begin{bmatrix} A - BD_c & -BC_c \\ B_c & A_c \end{bmatrix}, \qquad (17a)$$

$$B_a = \begin{bmatrix} BT_r \\ 0 \end{bmatrix}, \qquad (17b)$$

$$v_a[k] = \begin{bmatrix} v[k] \\ 0 \end{bmatrix}. \qquad (17c)$$

Then, by making an analogous analysis from Lemma 1, it is straightforward that for $x_a[k]$ being MSS, conditions on (14) must hold.

□ □ □

The asymptotically decaying reference r[k] requirement is also necessary for convergence of the sum on the performance index for the stochastic LQT problem, as can be seen on (7). This requirement can be relaxed when introducing a discount factor, $\gamma \in (0, 1)$, such that

$$J[k] = \xi\{\sum_{i=k}^{\infty} \gamma^{i-k} (z^\intercal[i] Q_a z[i] + u^\intercal[i] R u[i])\}, \qquad (18)$$

with

$$z[k] = \begin{bmatrix} (r[k] - y[k]) \\ x_c[k] \end{bmatrix}, \quad Q_a = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}, \qquad (19)$$

where both Q_1 and Q_2 are positive definite matrices, penalizing the control error and avoiding to get the dynamics of the controller itself boundless respectively.

3.2 PI for solving the stochastic LQT

In order to make the LQT problem look more like a RL problem, let the value function $V(X[k])$ be

$$V(X[k]) = J[k], \qquad (20)$$

where $X[k]$ stands for an augmented state vector containing the internal state of the process to be controlled, the state of the controller itself and the exogenous reference, i.e.,

$$X[k] = \begin{bmatrix} x^\intercal[k] & x_c^\intercal[k] & r^\intercal[k] \end{bmatrix}^\intercal. \qquad (21)$$

Then, the value function can be written as

$$V(X[k]) = \xi\{\sum_{i=k}^{\infty} \gamma^{i-k} (tr(P_w Q_1) + X^\intercal[i] \bar{Q} X[i] + u^\intercal[i] R u[i])\}, \qquad (22)$$

with Q_1 as on (19) and $\acute{Q}$ given by

$$\bar{Q} = \begin{bmatrix} C^\intercal Q_1 C & 0 & -C^\intercal Q_1 \\ 0 & Q_2 & 0 \\ -Q_1 C & 0 & Q_1 \end{bmatrix}. \qquad (23)$$

Theorem 1

Consider the control law

$$u[k] = T_r r[k] - (C_c x_c[k] + D_c x[k]), \qquad (24)$$

with $x_c[k]$ as described on (12), and $r[k]$ produced by the model

$$r[k+1] = F r[k], \qquad (25)$$

as depicted on Figure 1. Then, assuming the optimal value function is quadratic in the augmented state vector, i.e.,

$$V^*(X[k]) = X^\intercal[k]PX[k] + g[k], \quad (26)$$

for some stationary and symmetric matrix $P>0$, and $g[k]$ such that

$$g[k+1] = \left(\frac{1}{\gamma}\right) g[k] + tr\left(\frac{1}{\gamma}P_w Q_1 + P_v P_{11}\right), \quad (27)$$

where parameters (T_r, C_c, D_c) are given by

$$T_r = \gamma Z^{-1} B^\intercal P_{13} F, \quad (28a)$$
$$C_c = -\gamma Z^{-1} B^\intercal P_{12} A_c, \quad (28b)$$
$$D_c = -\gamma Z^{-1} M, \quad (28c)$$

with

$$Z = (R + \gamma B^\intercal P_{11} B), \quad (29a)$$
$$M = (A^\intercal P_{11} B + B_c^\intercal P_{21} B), \quad (29b)$$

and each matrix P_{ij} from P on (26) is such that

$$P_{11} = C^\intercal Q_1 C + \gamma (A^\intercal P_{11} A + B_c^\intercal P_{21} A + A^\intercal P_{12} B_c + B_c^\intercal P_{12} B_c) - \gamma^2 M Z^{-1} M^\intercal, \quad (30a)$$
$$P_{12} = \gamma (A^\intercal P_{12} A_c + B_c^\intercal P_{22} A_c) - \gamma^2 M Z^{-1} B^\intercal P_{12} A_c, \quad (30b)$$
$$P_{22} = Q_2 + \gamma A_c^\intercal P_{22} A_c - \gamma^2 A_c^\intercal P_{21} B Z^{-1} B^\intercal P_{12} A_c, \quad (30c)$$
$$P_{13} = -C^\intercal Q_1 + \gamma (A^\intercal P_{13} F + B_c^\intercal P_{23} F) - \gamma^2 M Z^{-1} B^\intercal P_{12} F, \quad (30d)$$
$$P_{23} = \gamma A_c^\intercal P_{23} F - \gamma^2 A_c^\intercal P_{21} B Z^{-1} B^\intercal P_{13} F, \quad (30e)$$

where $P_{ij} = P_{ji}^T$ as a consequence of the symmetry of P.

Proof:

Value function on (22) can be rewritten as

$$V(X[k]) = tr(P_w Q_1) + \xi\{X^\intercal[k]\bar{Q}X[k] + u^\intercal[k]Ru[k] + \gamma V(X[k+1])\}. \quad (31)$$

By the other hand, since the value function is assumed to be quadratic in terms of the augmented state vector,

$$V(X[k+1]) = X^\intercal[k+1]PX[k+1] + g[k+1], \quad (32)$$

but from (21) we have

$$X[k+1] = \begin{bmatrix} x^\intercal[k+1] & x_c^\intercal[k+1] & r^\intercal[k+1] \end{bmatrix}^\intercal. \quad (33)$$

Then, by replacing (5), (12) and (25) into (33), we have

$$V(X[k]) = tr(P_w Q_1 + \gamma P_v P_{11}) + \xi\{\gamma g[k+1] + (H^\intercal + u^\intercal) Z (H + u) + X^\intercal[k]\bar{P}X[k]\}, \quad (34)$$

where Z is defined as on (29a), H is given by

$$H = -\gamma Z^{-1}(M^\intercal x[k] + B^\intercal P_{12} A_c x_c[k] + B^\intercal P_{13} F r[k]), \quad (35)$$

with M as on (29b), and $\acute{P}$ is given by

$$\bar{P} = \begin{bmatrix} \bar{P}_{11} & \bar{P}_{21}^\intercal & \bar{P}_{31}^\intercal \\ \bar{P}_{21} & \bar{P}_{22} & \bar{P}_{32}^\intercal \\ \bar{P}_{31} & \bar{P}_{32} & \bar{P}_{33} \end{bmatrix}, \quad (36)$$

where each matrix $\acute{P}_{ij}$ is such that

$$\bar{P}_{11} = C^\intercal Q_1 C + \gamma (A^\intercal P_{11} A + B_c^\intercal P_{21} A + A^\intercal P_{12} B_c + B_c^\intercal P_{22} B_c) - \gamma^2 M Z^{-1} M^\intercal, \quad (37a)$$
$$\bar{P}_{12} = (A_c^\intercal P_{21} A + A_c^\intercal P_{22} B_c) - \gamma^2 A_c^\intercal P_{21} B Z^{-1} M^\intercal, \quad (37b)$$
$$\bar{P}_{22} = Q_2 + \gamma A_c^\intercal P_{22} A_c - \gamma^2 A_c^\intercal P_{21} B Z^{-1} B^\intercal P_{12} A_c, \quad (37c)$$
$$\bar{P}_{31} = -Q_1 C + \gamma (F^\intercal P_{31} A + F^\intercal P_{32} B_c) - \gamma^2 F^\intercal P_{31} B Z^{-1} (B^\intercal P_{11} A + B^\intercal P_{12} B_c), \quad (37d)$$
$$\bar{P}_{32} = \gamma F^\intercal P_{32} A_c - \gamma^2 F^\intercal P_{31} B Z^{-1} B^\intercal P_{12} A_c, \quad (37e)$$
$$\bar{P}_{33} = Q_1 + \gamma F^\intercal P_{33} F - \gamma^2 F^\intercal P_{31} B Z^{-1} B^\intercal P_{13} F. \quad (37f)$$

Finally, for minimizing the expression in (34), it can be seen that the optimal control law $u[k]$ has to be chosen such that

$$u[k] = \gamma Z^{-1}(M^\intercal x[k] + B^\intercal P_{12} A_c x_c[k] + B^\intercal P_{13} F r[k]) \quad (38)$$

so comparing terms on (24) and (38) yields (28).

In a similar form, by replacing (26) into (34), and comparing both sides from the expression, it can be found

$$g[k] = \gamma g[k+1] + tr(P_w Q_1 + \gamma P_v P_{11}), \quad (39)$$

so a little algebra yields (27). □ □ □

Remark 2. The assumption that $r[k]$ is generated by model described on (25), is valid for a large class of useful trajectories, such as a unit step signal, sinusoidal waveforms and more.

Then, Algorithm 2 shows the policy evaluation and improvement steps for the $l^{\text{-th}}$ iteration of PI.

It can be seen that Algorithm 2 can be implemented online, but all dynamics, including the generator model for the exogenous reference signal, have to be

known. Despite this fact, expressions can be easily computed, and we can achieve a state feedback controller with its own (stable) dynamics by just designing two parameters (A_c and B_c, since the rest depend on these) in place of the classical standard case with just one parameter, which could lead to a high gain for some processes.

Algorithm 2 Dynamic State Feedback Controller Design

1: **(Policy Evaluation)** Solve for each left-hand variable on (30), given the data from previous iteration, e.g.,

$$P_{11}^{(l)} = C^\intercal Q_1 C - \left(R+\gamma B^\intercal P_{11}^{(l)} B\right) D_c^{(l-1)} D_c^{\intercal,(l-1)} + \gamma\left(A^\intercal P_{11}^{(l)} A + A^\intercal P_{12}^{(l-1)} B_c^{(l-1)} + B_c^{\intercal,(l-1)} P_{21}^{(l-1)} A + B_c^{\intercal,(l-1)} P_{12}^{(l-1)} B_c^{(l-1)}\right),$$

$$\vdots$$

$$P_{23}^{(l)} = \gamma A_c^{\intercal,(l-1)}\left(P_{23}^{(l)} F - P_{21}^{(l-1)} B T_r^{(l-1)}\right)$$

2: **(Policy Improvement)** Update parameters A_c and B_c with this new data, such that

$$\bar{\lambda}\left(\begin{bmatrix} \bar{A}_{11}^{(l)} & \bar{A}_{12}^{(l)} \\ B_c^{(l)} & A_c^{(l)} \end{bmatrix}\right) < 1,$$

where

$$\bar{A}_{11}^{(l)} = A - \gamma B Z^{-1,(l)}\left(A^\intercal P_{11}^{(l)} B + B_c^{(l)} P_{21}^{(l)} B\right),$$
$$\bar{A}_{12}^{(l)} = -\gamma B Z^{-1,(l)} B^\intercal P_{12}^{(l)} A_c^{(l)},$$

with

$$Z^{(l)} = \left(R+\gamma B^\intercal P_{11}^{(l)} B\right). \quad (41)$$

Then, the rest of the parameters are updated as

$$T_r^{(l)} = \gamma Z^{-1,(l)} B^\intercal P_{13}^{(l)} F,$$
$$C_c^{(l)} = -\gamma Z^{-1,(l)} B^\intercal P_{12}^{(l)} A_c^{(l)},$$
$$D_c^{(l)} = -\gamma Z^{-1,(l)}\left(A^\intercal P_{11}^{(l)} B + B_c^{\intercal,(l)} P_{21}^{(l)} B\right).$$

3.3 LQT with unknown dynamics

Consider the LQT Q-function given by

$$Q(X[k],u[k]) = X^\intercal[k]\bar{Q}X[k] + u^\intercal[k]Ru[k] + tr(P_w Q_1) + \gamma\left(X^\intercal[k+1]PX[k+1] + g[k+1]\right), \quad (42)$$

with $\acute{Q}$ defined on (23). Then, this expression is equivalent to

$$Q(X[k],u[k]) = \begin{bmatrix} X[k] \\ u[k] \end{bmatrix}^\intercal H \begin{bmatrix} X[k] \\ u[k] \end{bmatrix} + tr(P_w Q_1 + \gamma P_v P_{11}) + \gamma g[k+1], \quad (43)$$

with symmetric H given by (44).

$$H = \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix}, \quad (44)$$

where

$$H_{uu} = R + \gamma B^\intercal P_{11} B, \quad (45a)$$

$$H_{xu} = \gamma \begin{bmatrix} B^\intercal P_{11} A + B^\intercal P_{12} B_c \\ B^\intercal P_{12} A_c \\ B^\intercal P_{13} F \end{bmatrix}, \quad (45b)$$

and H_{xx} given on (46). Based on (43), by finding the roots of its first derivative, it can be seen that the control law in terms of is H given by

$$u[k] = -H_{uu}^{-1} H_{ux} X[k]. \quad (47)$$

Note that the Q -function on (42) satisfies the Bellman equation

$$Q(X[k],u[k]) = X^\intercal[k]\bar{Q}X[k] + u^\intercal[k]Ru[k] + tr(P_w Q_1) + \gamma Q(X[k+1],u[k+1]). \quad (48)$$

Then, let $S[k]$ be

$$S[k] = \begin{bmatrix} 1 \\ X[k] \\ u[k] \end{bmatrix}, \quad (49)$$

so (43) is equivalent to

$$Q(X[k],u[k]) = S^\intercal[k]\bar{H}S[k], \quad (50)$$

with $\acute{H}$ given by

$$\bar{H} = \begin{bmatrix} tr(P_w Q_1 + \gamma P_v P_{11}) + \gamma g[k+1] & 0 & 0 \\ 0 & H_{xx} & H_{xu} \\ 0 & H_{ux} & H_{uu} \end{bmatrix}, \quad (51)$$

so now (48) becomes

$$S^\intercal[k]\bar{H}S[k] = X^\intercal[k]\bar{Q}X[k] + u^\intercal[k]Ru[k] + \gamma S^\intercal[k+1]\bar{H}S[k+1]. \quad (52)$$

Finally, (52) leads to Algorithm 3,which would match the same terms in Algorithm shown on [9], if we change our stochastic setup for a deterministic one, and not consider the internal state of the controller.

$$H_{xx} = \begin{bmatrix} C^\intercal Q_1 C + \gamma\left(A^\intercal P_{11} A + B_c^\intercal P_{21} A + A^\intercal P_{12} B_c + B_c^\intercal P_{22} B_c\right) & \gamma\left(A^\intercal P_{12} + B_c^\intercal P_{22}\right) A_c & -C^\intercal Q_1 + \gamma\left(A^\intercal P_{13} + B_c^\intercal P_{23}\right) F \\ \gamma\left(A_c^\intercal P_{21} A + A_c^\intercal P_{22} B_c\right) & Q_2 + \gamma A_c^\intercal P_{22} A_c & \gamma A_c^\intercal P_{23} F \\ -Q_1 C + \gamma\left(B^\intercal P_{11} A + B^\intercal P_{12} B_c\right) & \gamma B^\intercal P_{12} A_c & Q_1 + \gamma B^\intercal P_{13} F \end{bmatrix} \quad (46)$$

Algorithm 3 Controller synthesis with unknown dynamics

1: **(Policy Evaluation)** Based on observations of $S[k]$, $S[k+1]$ and $\left(X^{\intercal}[k]\bar{Q}X[k]+u^{\intercal,(l)}[k]Ru^{(l)}[k]\right)$ for the l-th iteration, use Least Squares to find $\bar{H}$,

$$S^{\intercal}[k]\bar{H}^{(l+1)}S[k] = X^{\intercal}[k]\bar{Q}X[k]+u^{\intercal,(l)}[k]Ru^{(l)}[k] + \gamma S^{\intercal}[k+1]\bar{H}^{(l+1)}S[k+1]. \quad (53)$$

2: **(Policy Improvement)** Update the control law

$$u^{(l+1)}[k] = -\left(H_{uu}^{-1}\right)^{(l+1)} H_{ux}^{(l+1)}X[k]. \quad (54)$$

4. Simulation Results

Consider the linear system given by

$$x[k+1] = \begin{bmatrix} 0.5 & 0 \\ 0.7 & 1.2 \end{bmatrix} x[k] + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u[k] + v[k], \quad (55a)$$

$$y[k] = \begin{bmatrix} 1 & 1 \end{bmatrix} x[k] + w[k], \quad (55b)$$

where $v[k]$ and $w[k]$ are the process and measurement noise respectively, with zero-mean and unitary variance.

Let the generator model F for the reference signal vary with time, set arbitrarily to

$$F[k] = \begin{cases} 0 & k < 60 \\ 10 & k \geq 60 \end{cases} \quad (56)$$

and $r[0]=1$. Also, penalizing weights for the performance index were set to $Q_1=5, Q_2=5$ and $R=1$, and the discount factor $\gamma = 0.8$.

Figure 2 shows the evolution of the generated control signal during the learning process, with u_1 obtained from Algorithm 2 and u_2 obtained from Algorithm 3, where the prior leads to parameters

$$T_r = 1.3667 \quad A_c = 0.4 \quad B_c = \begin{bmatrix} 1 & -1.52 \end{bmatrix} \quad (57a)$$

$$C_c = -0.5 \qquad D_c = \begin{bmatrix} 0.3 & 2.1 \end{bmatrix}. \quad (57b)$$

For Algorithm 3, note that 25 data samples were collected to perform least squares in each iteration, since

$$\left(n_x+n_{x_c}+n_u+n_y\right)\cdot\left(n_x+n_{x_c}+n_u+n_y+1\right)/2,$$

data tuples or more are needed. Then, H_{uu} and H_{ux} learned, which construct the control law on (47) are,

$$H_{uu} = 341.9354, \quad (58a)$$

$$H_{ux} = \begin{bmatrix} 101.134 & 704.594 & -173.186 & -450.291 \end{bmatrix}, \quad (58b)$$

leading to

$$u[k] = \begin{bmatrix} -0.2957 & -2.06 \end{bmatrix} x[k] + 0.5064x_c[k] + 1.3169r[k], \quad (59)$$

which is similar to the final parameters found by Algorithm 2.

It can be seen on Figure 2, that Algorithm 2 and 3 achieves the same performance when the learning process is finished, but the latter have a slower convergence rate as expected, since adding knowledge represents an advantage, but it comes with a very high cost if the process is too complex for obtaining an accurate model.

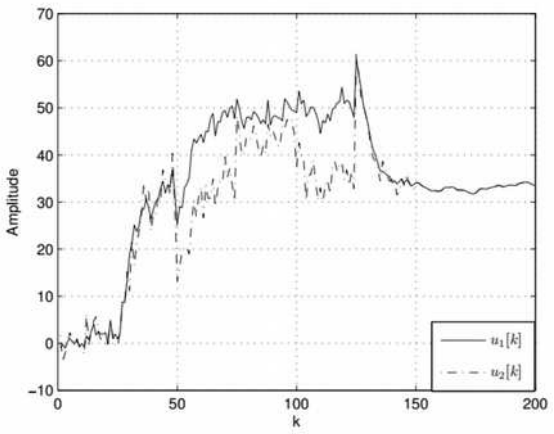

Figure 2. Control signal during learning process

In order to compare the performance of the classical state feedback controller with the proposed in this document, consider the control law

$$u_L[k] = 2r[k] - \begin{bmatrix} 0.3 & 4 \end{bmatrix} x[k], \quad (60)$$

and

$$u[k] = 1.3667r[k] + 0.5x_c[k] - \begin{bmatrix} 0.3 & 2.1 \end{bmatrix} x[k], \quad (61)$$

such that the eigenvalues of the control loop when using the classical state feedback control on (60) are {0.5, 0.8}, and when using the proposed structure on (61) its eigenvalues are {0.5, 0.59, 0.8}. Despite adding as many eigenvalues as elements have the controller state vector, the largest eigenvalue limits the speed of the control loop, so both control laws have the same speed of convergence, but it is shown in Figure 3, with parameters already learned, that an appropriate choice on the parameters of the proposed controller can lead to a better performance in terms of minimizing the energy spent on the control signal.

5. Conclusions

Two PI algorithms were presented, one for the case when full knowledge on the model of the process to be controlled is assumed to be available, and the other one when there is no knowledge at all. The former is especially useful for not having to tune by hand four parameters at a time, when the controller is allowed to have its own internal dynamics.

The case when the state is not directly measurable, and has to be estimated instead, remains as future work. This case would be of special interest for a model-free control over plants like the flywheel inverted pendulum, which would represent an initial step towards generating a safe walk learning for a biped, or legged robot.

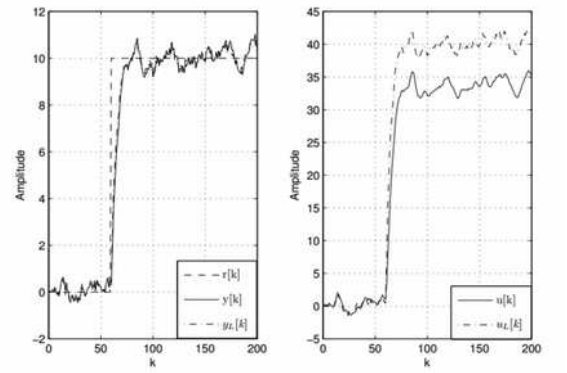

Figure 3. Static and Dynamic Controller Structures Comparison

Acknowledgements

The authors gratefully acknowledge support received from UTFSM through a DGIP 23.15.26 grant, a PIIC grant, and funding from project Mecesup FSM-0707.

REFERENCES

1. ANDERSON, B. D. O., J. B. MOORE, **Optimal Control: Linear Quadratic Methods**, Courier Dover Publ., 2007.
2. BARBIERI, E., R. ALBA-FLORES, **On the Infinite Horizon LQ Tracker.** Systems & Control Letters, vol. 40(2), 2000, pp. 77-82.
3. BERTSEKAS, D. P., **Dynamic Programming and Optimal Control,** vol. 1. Athena Scientific Belmont, MA, 1995.
4. BRADTKE, S. J., B. E. YDSTIE, A. G. BARTO, **Adaptive Linear Quadratic Control using Policy Iteration.** In American Control Conference, 1994, volume 3, pp. 3475-3479.
5. CHEN, S., X. LI, X. Y. ZHOU, **Stochastic Linear Quadratic Regulators with Indefinite Control Weight Costs.** SIAM Journal on Control and Optimization control weight costs. SIAM Journal on Control and Optimization, vol. 36(5), 1998, pp. 1685-1702.
6. DE SOUZA, C. E., M. D. FRAGOSO, **On the Existence of Maximal Solution for Generalized Algebraic Riccati Equations Arising in Stochastic Control.** Syst. & Ctrl. Letters, vol. 14(3), 1990, pp. 233-239.
7. HE, P., S. JAGANNATHAN, **Reinforcement Learning-based Output Feedback Control of Nonlinear Systems with Input Constraints.** Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans., vol. 35(1), 2005, pp. 150-154.
8. HUANG, Y., W. ZHANG, H. ZHANG, **Infinite Horizon Linear Quadratic Optimal Control for Discrete Time Stochastic Systems.** Asian Journal of Control, vol. 10(5), 2008, pp. 608-615.
9. KIUMARSI, B., F. L. LEWIS, M. B. NAGHIBI-SISTANI, A. KARIMPOUR, **Optimal Tracking Control of Unknown Discrete-time Linear Systems using Input-output Measured Data.** IEEE Trans. on Cybernetics, vol. 45(12), 2015, pp. 2770-2779.
10. KOBER, J., J. A. BAGNELL, J. PETERS, **Reinforcement Learning in Robotics: A Survey.** Intl. Journal of Robotics Research, vol. 32(11), 2013, pp. 1238-1274.
11. KUMARESAN, N., P. BALASUBRAMANIAM, P., **Optimal Control for Stochastic Linear Quadratic Singular System using Neural Networks.** Journal of Process Control, vol. 19(3), 2009, pp. 482-488.
12. LEWIS, F. L., D. LIU, **Reinforcement Learning and Approximate Dynamic Programming for Feedback Control**, volume 17. John Wiley & Sons, 2013.
13. LEWIS, F. L., K. G. VAMVOUDAKIS, **Reinforcement Learning for Partially**

Observable Dynamic Processes: Adaptive Dynamic Programming using Measured Output Data. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans., vol. 41(1), 2011, pp. 14-25.

14. QIN, C., H. ZHANG, Y. LUO, **Online optimal tracking control of continuous-time linear systems with unknown dynamics by using adaptive dynamic programming.** International Journal of Control, 87(5):1000–1009, 2014.
15. SODERSTRÖM, T., **Discrete-time stochastic systems: estimation and control.** Springer, 2002.
16. SUTTON, R. S., BARTO, A. G., WILLIAMS, R. J., **Reinforcement learning is direct adaptive optimal control.** Control Systems, IEEE, 12(2): 19–22, 1992.
17. TEN HAGEN, S., KROSE, B., **Linear quadratic regulation using reinforcement learning.** In 8th Belgian Dutch Conference on Machine learning, pp. 39–46, 1998.
18. WATKINS, C.J.C.H., **Learning from delayed rewards.** PhD thesis, University of Cambridge, 1989.
19. WERBOS, P. J., **Neural networks for control and system identification.** In Decision and Control, 1989, Proceedings of the 28th IEEE Conference on, pp. 260–265. IEEE, 1989.
20. WERBOS, P.J., **Approximate dynamic programming for real-time control and neural modeling.** Handbook of intelligent control: Neural, fuzzy, and adaptive approaches, 15: 493–525, 1992.
21. WONHAM, W. M. **On a matrix riccati equation of stochastic control.** SIAM Journal on Control, 6(4): 681– 697, 1968.
22. ZHANG, W., LI, G., **Discrete-time indefinite stochastic linear quadratic optimal control with second moment constraints.** Mathematical Problems in Engineering, 2014, 2014.