Policy Iteration for Linear Quadratic Games with Stochastic Parameters

Benjamin Gravell, *Student Member, IEEE,* Karthik Ganapathy, *Student Member, IEEE,* and Tyler Summers, *Member, IEEE*

Abstract—**Robustness is a key challenge in the integration of learning and control. In machine learning and robotics, two common approaches to promote robustness are adversarial training and domain randomization. Both of these approaches have analogs in control theory: adversarial training relates to $\mathscr{H}_\infty$ control and dynamic game theory, while domain randomization relates to theory for systems with stochastic model parameters. We propose a stochastic dynamic game framework that integrates both of these complementary approaches to modeling uncertainty and promoting robustness. We describe policy iteration algorithms in both model-based and model-free settings to compute equilibrium strategies and value functions. We present numerical experiments that illustrate their effectiveness and the value of combining uncertainty representations in our integrated framework. We also provide an open-source implementation of the algorithms to facilitate their wider use.**

Index Terms—**Stochastic optimal control, robust control, game theory, uncertain systems, numerical algorithms.**

I. Introduction

ROBUSTNESS has been a perennial concern in control system analysis and design, and remains a major challenge for emerging control systems and networks that integrate machine learning components and algorithms. Robustness under adversarial additive disturbances and systems with stochastic parameters have each seen extensive study, but have largely been treated separately. Here, we consider solving a linear quadratic game with stochastic parameters (SLQ game) via the dynamic programming technique of policy iteration.

The canonical linear quadratic regulator (LQR) problem was shown to be efficiently solvable via an iterative technique [1], later recognized as equivalent to policy iteration applied to the LQR problem. Connections between dynamic LQ games and $\mathscr{H}_\infty$ control were recognized and made popular in [2]. In [3] it was shown that the solution of a generalized algebraic Riccati equation (GARE) gave equilibrium strategies for LQ games. Nearly all work on solving this equation has focused on either value iteration [2] or semidefinite programming (SDP) methods [4]. One exception was [5] where the model-free Q-learning algorithm, a type of approximate policy iteration, was proposed. In [6], model-free first-order policy optimization techniques were shown to converge to solutions of the GARE using nested updates, which are known to be sample inefficient for LQ systems [7].

Systems with stochastic parameters, equivalent to state- and control-multiplicative noise, arise naturally in a variety of settings such as networked systems with noisy communication channels, modern power networks with large penetration of intermittent renewables, turbulent fluid flow, and neuronal brain networks. Optimal control [8] and (mean-square) stabilizability [9] of such systems have been analyzed nearly as long as that for deterministic systems. In [10] the solution to the LQR problem for discrete-time systems with multiplicative noise was given via a GARE, and in [11] it was shown that policy iteration solves this problem.

For systems with truly stochastic parameters, as for those with deterministic ones, it is advantageous to consider an adversarial disturbance in order to synthesize $\mathscr{H}_\infty$ robust controllers to ensure mean-square stability of the system. In [12] a generalized stochastic bounded real lemma (SBRL) and nonlinear matrix inequalities were developed for discrete-time systems in the more general output feedback setting, but only state- and disturbance input-multiplicative noises were considered, with control inputs perfectly executed, and the practical control design task was not investigated thoroughly. Perhaps most directly related to our work is [13] which considered the mixed $\mathscr{H}_2/\mathscr{H}_\infty$ control design problem for systems with stochastic parameters, and showed that the SBRL of [12] was equivalent to a GARE, and offered a convex SDP approach towards suboptimal solutions.

As an alternative view, stochastic parameters can be used as a device during control design to induce robustness to structured parametric uncertainty [14], [15], which can be combined in a complementary way with additive adversarial disturbances to handle unstructured model uncertainty. This is analogous to developments in machine learning where unstructured uncertainty is addressed by adversarial training examples [16] while structured uncertainty is addressed by domain randomization [17]; this perspective is elaborated on in Section II. To our knowledge, such a perspective and formulation is essentially absent in the control literature.

In contrast to the existing literature, our current work frames the problem as a zero-sum dynamic game, considers control-dependent noise, and offers an efficient policy iteration scheme that works both when the model is known and unknown. Our main contributions are:

- We present a zero-sum dynamic linear quadratic game with stochastic parameters that allows a distinct and complementary treatment of structured (parametric) and unstructured (nonparametric) model uncertainty.
- We describe policy iteration algorithms in both model-based and model-free settings to compute saddle point equilibrium strategies and value functions. In the model-

The authors are with the Control, Optimization, and Networks Lab, University of Texas at Dallas, 800 W. Campbell Rd, Richardson, TX 75080 (email: {benjamin.gravell}, {karthik.ganapathy}, {tyler.summers}@utdallas.edu).

based setting, the equilibrium strategies are computed from model and uncertainty representation parameters. In the model-free setting, they are computed by approximate policy iteration with least-squares estimation of the state-action value function from sample trajectory data.

- We present numerical experiments that illustrate and demonstrate the effectiveness of the algorithms and the value of combining uncertainty representations.
- We provide an open-source implementation of the algorithms to facilitate their wider use.

The rest of the paper is organized as follows. In Section II we formulate the stochastic dynamic game. In Section III we describe a model-based policy iteration algorithm. In Section IV we present a model-free variation that utilizes least-squares temporal difference Q-function learning from trajectory data. Section V provides numerical experiments to illustrate the results, and Section VI concludes.

II. Stochastic Dynamic Games with Multiplicative Noise

We consider the zero-sum stochastic dynamic game

$$\begin{aligned} &\underset{\pi_u \in \Pi_u}{\text{minimize}} \ \underset{\pi_v \in \Pi_v}{\text{maximize}} && \mathop{\mathbb{E}}_{\alpha,\beta,\gamma} \sum_{t=0}^{\infty} c_t, \\ &\text{subject to} && x_{t+1} = \widetilde{A} x_t + \widetilde{B} u_t + \widetilde{C} v_t, \end{aligned} \tag{1}$$

where $x_t \in \mathbb{R}^n$ is the system state, $u_t \in \mathbb{R}^m$ is the control input, $v_t \in \mathbb{R}^p$ is the (adversarial) disturbance input, the initial state x_0 is distributed according to distribution $\mathscr{D}_0$, and c_t is the quadratic stage cost

$$c_t := c(x_t, u_t, v_t) := x_t^\intercal Q x_t + u_t^\intercal R u_t - v_t^\intercal S v_t,$$

where $Q \succ 0$, $R \succ 0$, $S \succ 0$. The dynamics are described by the random system matrices

$$\widetilde{A} = A + \sum_{i=1}^{q} \alpha_{ti} A_i, \ \widetilde{B} = B + \sum_{j=1}^{r} \beta_{tj} B_j, \ \widetilde{C} = C + \sum_{k=1}^{s} \gamma_{tk} C_k,$$

which incorporate a (deterministic) mean dynamics matrix $A \in \mathbb{R}^{n\times n}$, control input matrix $B \in \mathbb{R}^{n\times m}$, disturbance input matrix $C \in \mathbb{R}^{n\times p}$ and multiplicative noise terms modeled by the i.i.d. (across time), zero-mean, mutually independent scalar random variables α_{ti}, β_{tj}, and γ_{tk}, which have variances $\sigma^2_{\alpha,i}$, $\sigma^2_{\beta,j}$, and $\sigma^2_{\gamma,k}$ respectively. The matrices $A_i \in \mathbb{R}^{n\times n}$, $B_j \in \mathbb{R}^{n\times m}$, $C_k \in \mathbb{R}^{n\times p}$ specify how each scalar noise term affects the system dynamics, control input, and disturbance input matrices. Noises are assumed zero-mean without loss of generality; non-zero-mean noises can be accommodated by simply subtracting the mean from the noises and shifting the A, B, and C matrices by corresponding amounts. The expectation is with respect to the noises $\{\alpha_{ti}\}, \{\beta_{tj}\}, \{\gamma_{tk}\}$, hereafter abbreviated to simply $\mathbb{E}_{\alpha,\beta,\gamma}$.

The goal is to determine saddle point equilibrium state feedback policies π_u and π_v with $u_t = \pi_u(x_t)$ and $v_t = \pi_v(x_t)$ from sets Π_u and Π_v of admissible policies. We assume that the problem data A, B, C, Q, R, S, A_i, B_j, C_k, $\sigma^2_{\alpha,i}$, $\sigma^2_{\beta,j}$, and $\sigma^2_{\gamma,k}$ permit existence of a saddle point equilibrium in pure policies and finiteness of the corresponding equilibrium value of the problem. This can be interpreted as a problem of designing a feedback controller for the nominal system (A, B) that is robust to the worst-case inputs of the adversary and the stochastic variations of model parameters. These two uncertainty representations are complementary: the adversarial input is well-suited to promoting robustness to *non-parametric* uncertainty, such as unmodeled dynamics, bounded nonlinearities, and truncated infinite-dimensionalities, while the stochastic model parameters promote robustness to *parametric* uncertainties in the model parameters, which may result from their estimation from noisy data or from inherent multiplicative noise phenomena (e.g., packet dropouts in networks). As a special case, taking $S = \gamma^2 I$ gives a connection to $\mathscr{H}_\infty$ control since γ is an upper bound on the L_2 gain disturbance attenuation i.e. the $\mathscr{H}_\infty$ norm of the system [2]. In machine learning, domain randomization is a heuristic for inducing robustness of a classifier by randomly changing the parameters of the environment (domain) during training [17]. Likewise, in (1) stochastic parameters randomly change the $\widetilde{A}$, $\widetilde{B}$, $\widetilde{C}$ matrices (domain) at each time step, which gives guaranteed robustness margins [14], [15].

Notation. Let $\rho(M)$ denote the spectral radius (largest magnitude of an eigenvalue) of matrix M. Let $\|M\|$ denote the spectral norm (largest singular value) of matrix M. Let $\otimes$ denote the Kronecker product. Let $\text{vec}(M)$ denote the vectorization of matrix M by stacking its columns. Let $\text{svec}(M)$ denote the symmetric (or half) vectorization of matrix M by vectorization of the upper triangular (including the main diagonal) part of matrix M with off-diagonal entries multiplied by $\sqrt{2}$ such that $\|M\|_F^2 = \text{svec}(M)^\intercal \text{svec}(M)$ where $\|\cdot\|_F$ is the Frobenius norm. Let $\text{smat}(v)$ denote the symmetric matricization of vector v i.e. the inverse operation of $\text{svec}(\cdot)$ such that $\text{smat}(\text{svec}(M)) = M$.

III. Model-Based Computation of Equilibrium Policies and Value Functions via Policy Iteration

In contrast to deterministic linear systems and linear systems with additive noise, proper functioning (i.e., finite infinite-horizon quadratic cost) of linear systems with stochastic parameters requires satisfaction of the stronger *mean-square stability* condition:

Definition 1: The closed-loop system in (1) is said to be *mean-square stable* if

$$\lim_{t\to\infty} \mathop{\mathbb{E}}_{\alpha,\beta,\gamma} [x_t x_t^\intercal] = 0 \quad \forall x_0.$$

Throughout the paper we abbreviate "mean-square stability" to "ms-stability" for brevity.

We use policy iteration to solve (1); this requires first characterizing the cost under non-equilibrium ms-stable policies (those which admit finite cost in (1), but are not the minimax solution), and under equilibrium policies (those which solve (1)). It is well known (e.g. [13]) that equilibrium policies are linear state-feedback of the form $u_t = K x_t$, $v_t = L x_t$ where K and L are control and disturbance gain matrices, so we consider policies only of this type. We begin by defining several quantities which admit compact cost characterizations, which arise naturally in the study of linear optimal control.

A. Non-equilibrium strategies

Denote the deterministic nominal closed-loop system matrix as $A_{KL} := A + BK + CL$, the stochastic closed-loop system matrix as $\widetilde{A}_{KL} := \widetilde{A} + \widetilde{B}K + \widetilde{C}L$, and the closed-loop state-cost matrix as $Q_{KL} := Q + K^\intercal RK - L^\intercal SL$. Define the stage cost matrices $P_{KL,t} := \mathbb{E}_{\alpha,\beta,\gamma}\left[(\widetilde{A}_{KL}^\intercal)^t Q_{KL} \widetilde{A}_{KL}^t\right]$. Following [18], [19], we define the linear operator $\mathscr{F}_{KL}(\cdot)$ which operates on a constant symmetric matrix X:

$$\begin{aligned}\mathscr{F}_{KL}(X) &:= \mathop{\mathbb{E}}_{\alpha,\beta,\gamma} \widetilde{A}_{KL}^\intercal X \widetilde{A}_{KL}\\ &= A_{KL}^\intercal X A_{KL} + \sum_{i=1}^{q} \sigma_{\alpha,i}^2 A_i^\intercal X A_i\\ &\quad + \sum_{j=1}^{r} \sigma_{\beta,j}^2 (B_jK)^\intercal X (B_jK) + \sum_{k=1}^{s} \sigma_{\gamma,k}^2 (C_kL)^\intercal X (C_kL),\end{aligned}$$

where expectation in the first line is with respect to the noises in $\widetilde{A}_{KL}$, and the second line follows by mutual independence of the noises. Likewise, we define $\mathscr{F}_{KL}$ (without an argument) as a matrix which acts on the vectorization of X:

$$\begin{aligned}\mathscr{F}_{KL} &:= A_{KL}^\intercal \otimes A_{KL}^\intercal + \sum_{i=1}^{q} \sigma_{\alpha,i}^2 A_i^\intercal \otimes A_i^\intercal\\ &\quad + \sum_{j=1}^{r} \sigma_{\beta,j}^2 (B_jK)^\intercal \otimes (B_jK)^\intercal + \sum_{k=1}^{s} \sigma_{\gamma,k}^2 (C_kL)^\intercal \otimes (C_kL)^\intercal,\end{aligned}$$

so that $\text{vec}(\mathscr{F}_{KL}(X)) = \mathscr{F}_{KL}\,\text{vec}(X)$. These operators characterize the *cost* dynamics as

$$\begin{aligned}P_{KL,t+1} &= \mathop{\mathbb{E}}_{\alpha,\beta,\gamma}\left[(\widetilde{A}_{KL}^\intercal)^{t+1} Q_{KL} \widetilde{A}_{KL}^{t+1}\right]\\ &= \mathop{\mathbb{E}}_{\alpha,\beta,\gamma}\left[\widetilde{A}_{KL}^\intercal P_{KL,t} \widetilde{A}_{KL}\right] = \mathscr{F}_{KL}(P_{KL,t}).\end{aligned}$$

This gives an easy way to check ms-stability, as stated in the following fact, which is straightforward to prove by following similar arguments as in [9], [19]:

Fact 1: A pair of gains K and L are ms-stabilizing if and only if the spectral radius $\rho(\mathscr{F}_{KL}) < 1$.

Now we characterize the costs (values) associated with a pair of gains (K,L). The state-value function determines the cost of starting in state $x = x_0$ and following policies $u_t = Kx_t$, $v_t = Lx_t$ forever, defined as

$$V_{KL}(x) := \mathop{\mathbb{E}}_{\alpha,\beta,\gamma} \sum_{t=0}^{\infty} x_t^\intercal Q_{KL} x_t = x^\intercal P_{KL} x,$$

where P_{KL} is the positive semidefinite solution to the generalized Lyapunov equation (which generalizes one given for systems with multiplicative noise in e.g. [13])

$$P_{KL} = Q_{KL} + \mathscr{F}_{KL}(P_{KL}).$$

The state-action value (Q-) function determines the cost of starting in state $x = x_0$, taking actions $u = u_0$, $v = v_0$, then following policies $u_t = Kx_t$, $v_t = Lx_t$ thereafter, defined as

$$\mathscr{Q}_{KL}(x,u,v) := c(x,u,v) + \mathop{\mathbb{E}}_{\alpha,\beta,\gamma} V_{KL}(\widetilde{A}x + \widetilde{B}u + \widetilde{C}v),$$

where the expectation is with respect to the noises in the stochastic $\widetilde{A}, \widetilde{B}, \widetilde{C}$ in the argument to V_{KL}. Expanding, we can also write the state-action value function as a quadratic form:

$$\begin{aligned}\mathscr{Q}_{KL}(x,u,v) &= x^\intercal Qx + u^\intercal Ru - v^\intercal Sv\\ &\quad + \mathop{\mathbb{E}}_{\alpha,\beta,\gamma}\left[(\widetilde{A}x + \widetilde{B}u + \widetilde{C}v)^\intercal P_{KL} (\widetilde{A}x + \widetilde{B}u + \widetilde{C}v)\right]\\ &= \begin{bmatrix} x \\ u \\ v \end{bmatrix}^\intercal \begin{bmatrix} H_{xx} & H_{xu} & H_{xv} \\ H_{ux} & H_{uu} & H_{uv} \\ H_{vx} & H_{vu} & H_{vv} \end{bmatrix}_{KL} \begin{bmatrix} x \\ u \\ v \end{bmatrix}, \end{aligned} \tag{2}$$

where H_{KL} is a symmetric matrix defined in terms of P_{KL} with blocks

$$\begin{aligned} H_{xx} &= Q + A^\intercal P_{KL} A + \sum_{i=1}^{q} \sigma_{\alpha,i}^2 A_i^\intercal P_{KL} A_i,\\ H_{uu} &= R + B^\intercal P_{KL} B + \sum_{j=1}^{r} \sigma_{\beta,j}^2 B_j^\intercal P_{KL} B_j,\\ H_{vv} &= -S + C^\intercal P_{KL} C + \sum_{k=1}^{s} \sigma_{\gamma,k}^2 C_k^\intercal P_{KL} C_k,\\ H_{ux} &= B^\intercal P_{KL} A, \quad H_{vx} = C^\intercal P_{KL} A, \quad H_{vu} = C^\intercal P_{KL} B, \end{aligned}$$

which follows by definition and mutual independence of the noises. If the pair (K,L) is ms-stabilizing, then $V_{KL}(x)$ and $\mathscr{Q}_{KL}(x,u,v)$ return finite values for finite x,u,v.

B. Equilibrium strategies

Equilibrium strategies are given by solving the generalized discrete-time algebraic Riccati equation (GARE)

$$P^* = H_{xx}^* - \begin{bmatrix} H_{ux}^* \\ H_{vx}^* \end{bmatrix}^\intercal \begin{bmatrix} H_{uu}^* & H_{uv}^* \\ H_{vu}^* & H_{vv}^* \end{bmatrix}^{-1} \begin{bmatrix} H_{ux}^* \\ H_{vx}^* \end{bmatrix}, \tag{3}$$

where H^* is defined in terms of P^* identically as H_{KL} is defined in terms of P_{KL}. The optimal gains are then found as

$$\begin{aligned} K^* &= (H_{uu}^* - H_{uv}^* {H_{vv}^*}^{-1} H_{vu}^*)^{-1} (H_{uv}^* {H_{vv}^*}^{-1} H_{vx}^* - H_{ux}^*),\\ L^* &= (H_{vv}^* - H_{vu}^* {H_{uu}^*}^{-1} H_{uv}^*)^{-1} (H_{vu}^* {H_{uu}^*}^{-1} H_{ux}^* - H_{vx}^*). \end{aligned}$$

To ensure well-posedness of the game we assume:

1) The pair (A,B) is ms-stabilizable,
2) There exists a minimal positive definite solution P^* to the GARE such that
$$H_{vv}^* = -S + C^\intercal P^* C + \sum_{k=1}^{s} \sigma_{\gamma,k}^2 C_k^\intercal P^* C_k \prec 0,$$
3) The optimal adversary gain L^* satisfies
$$Q - (L^*)^\intercal S L^* \succ 0.$$

The first condition (implicitly) imposes restrictions on the structure of (A,B) as well as upper bounds on the noise variances [20]. The second condition is a generalization of a standard assumption that ensures concavity of the cost in the adversary inputs leading to existence of the value of the game, as in [2], [5]. The third condition ensures the saddle-point property of the optimal gains (K^*, L^*) i.e. the input sequences generated under the policies $u_t = K^* x_t$, $v_t = L^* x_t$ constitute the Nash equilibrium of the game, as in [6]. More directly, this ensures $Q_{K^*L^*} \succ 0$ so that P^* induces a valid Lyapunov function $V_{K^*L^*}(x) = x^\intercal P^* x$ which decreases along closed-loop trajectories.

Existing approaches to solve the GARE are based either on value iteration, which turns (3) into a recursive update, or on solving an SDP which requires knowledge of the model. As an effective alternative, we propose using policy iteration, which also extends readily to the model-free setting. Policy iteration, due originally to Howard [21], is a dynamic programming technique which proceeds iteratively by updating an approximation of the optimal value function under the current policy (policy evaluation), updating an approximation of the optimal policy under the current value function (policy improvement), and repeating until convergence. For SLQ games policy iteration specializes to Algorithm 1, where each step involves only linear algebraic operations. This algorithm derives from the fact that there is known to be a saddle point equilibrium in pure *linear* policies, with a corresponding *quadratic* equilibrium value function, under the appropriate assumptions. In [22] it was shown that Newton's root-finding algorithm applied to the GARE converges at a *quadratic* rate to the equilibrium strategies from any initial feasible pair of gains. It can be shown that Newton's method is equivalent to the exact policy iteration described here, and thus Algorithm 1 enjoys the same convergence results.

Algorithm 1 Exact policy iteration for SLQ games

```
Input: Ms-stabilizing linear policy pair $(K^0, L^0)$, convergence threshold $\varepsilon > 0$.
1: Initialize $P^0 = 0$, $P^1 = \infty$, $k = 0$
2: while $\|P^{k+1} - P^k\| > \varepsilon$ do
3:     Policy Evaluation: Compute the value of the current policy pair by finding the solution $P^k_{KL}$ to the Lyapunov equation $P^k_{KL} = Q^k_{KL} + \mathscr{F}^k_{KL}(P^k_{KL})$
4:     Policy Improvement: Update the policy pair according to the dynamic programming equations
           $K^{k+1} = (H_{uu} - H_{uv}H_{vv}^{-1}H_{vu})^{-1}(H_{uv}H_{vv}^{-1}H_{vx} - H_{ux})$
           $L^{k+1} = (H_{vv} - H_{vu}H_{uu}^{-1}H_{uv})^{-1}(H_{vu}H_{uu}^{-1}H_{ux} - H_{vx})$
       with $H = H^k_{KL}$ defined as before in terms of $P^k_{KL}$.
5:     $k \leftarrow k+1$
6: end while
Output: Approximately optimal policy pair $(K^{k+1}, L^{k+1})$
```

Note that Algorithm 1 requires perfect knowledge of the problem data $A, B, C, \{A_i\}, \{B_j\}, \{C_k\}, \{\sigma^2_{\alpha,i}\}, \{\sigma^2_{\beta,j}\}, \{\sigma^2_{\gamma,k}\}$ in order to execute the policy evaluation step by solving for P_{KL} exactly. In practical applications, such knowledge may be extremely difficult or impossible to acquire, motivating a model-free formulation of policy iteration.

IV. Model-Free Computation of Equilibrium Policies and Value Functions via Approximate Policy Iteration

In the model-free setting we do not have access to the problem data, and the value functions must be estimated from sample trajectory data. In this case it is not necessary to estimate the value function matrix P, but rather is sufficient to estimate the state-action value function matrix H which is used during policy improvement. For estimation of the Q-function we first develop an "adaptive dynamic programming" type of least squares (LSADP) as developed in [23], then give a more sophisticated least-squares temporal difference learning for Q-functions (LSTDQ) developed in [24].

First we define the concatenated pseudo-state vectors

$$z = \begin{bmatrix} x^\intercal & u^\intercal & v^\intercal \end{bmatrix}^\intercal, \quad w = \begin{bmatrix} x^\intercal & (Kx)^\intercal & (Lx)^\intercal \end{bmatrix}^\intercal,$$

and similarly z_t and w_t using x_t, u_t, v_t, so $z_t = w_t$ under the closed-loop policies $u_t = Kx_t$, $v_t = Lx_t$. With a slight abuse of notation, the definition of the Q-function becomes

$$\mathscr{Q}_{KL}(z) := c(z) + \mathop{\mathbb{E}}_{\alpha,\beta,\gamma} V_{KL}(\widetilde{A}x + \widetilde{B}u + \widetilde{C}v),$$

from which it follows

$$\mathscr{Q}_{KL}(w_t) = c(w_t) + \mathop{\mathbb{E}}_{\alpha,\beta,\gamma} V_{KL}(\widetilde{A}x_t + \widetilde{B}Kx_t + \widetilde{C}Lx_t) = V_{KL}(x_t).$$

Noting that

$$\mathbb{E}\mathscr{Q}_{KL}(w_{t+1}) = \mathbb{E}V_{KL}(x_{t+1}) = \mathbb{E}V_{KL}(\widetilde{A}x_t + \widetilde{B}u_t + \widetilde{C}v_t),$$

where the expectations are all with respect to the noises $\{\alpha_{ti}\}, \{\beta_{tj}\}, \{\gamma_{tk}\}$ in the dynamic update of x_{t+1} in the argument to each function i.e. $x_{t+1} = \widetilde{A}x_t + \widetilde{B}u_t + \widetilde{C}v_t$, the Q-function can be written recursively as

$$\mathscr{Q}_{KL}(z_t) = c(z_t) + \mathbb{E}\mathscr{Q}_{KL}(w_{t+1}).$$

By using symmetric vectorization we can convert $\mathscr{Q}_{KL}$ from a quadratic form to a linear architecture:

$$\mathscr{Q}_{KL}(z) = \phi(z)^\intercal \Theta,$$

where we define the parameter vector $\Theta = \text{svec}(H_{KL})$, and the feature map $\phi(z) = \text{svec}\left(zz^\intercal\right)$. Rearranging the recursive Q-function relationship,

$$\begin{aligned} c(z_t) &= \mathscr{Q}_{KL}(z_t) - \mathbb{E}\mathscr{Q}_{KL}(w_{t+1}) \\ &= \phi(z_t)^\intercal \Theta - \mathbb{E}\left[\phi(w_{t+1})^\intercal \Theta\right] \end{aligned}$$

However, in the model-free setting we cannot compute the expectation $\mathbb{E}\left[\phi(w_{t+1})^\intercal \Theta\right]$ exactly, so a natural approximation is to replace the expectation with transition samples:

$$c(z_t) = \left(\phi(z_t) - \phi(w_{t+1})\right)^\intercal \Theta + e_t, \tag{4}$$

where the error is

$$e_t = \phi(w_{t+1})^\intercal \Theta - \mathbb{E}\left[\phi(w_{t+1})^\intercal \Theta\right].$$

By exciting the system with inputs for $\ell + 1$ time steps and recording the trajectory of observed states, inputs, and costs we obtain a data set $\{z_t, c_t\}_{t=0}^{\ell+1}$, which we call a *rollout*. From such a rollout, for a given (K, L) we can compute the augmented rollout $\{z_t, w_t, c_t\}_{t=0}^{\ell+1}$. For a rollout of length $\ell + 1$, this yields ℓ equations of the form of (4). Stacking rows, we obtain the matrix relation $Y = Z\Theta + E$ where

$$Y = \begin{bmatrix} c(z_0) \\ c(z_1) \\ \vdots \\ c(z_\ell) \end{bmatrix}, \quad Z = \begin{bmatrix} (\phi(z_0) - \phi(w_1))^\intercal \\ (\phi(z_1) - \phi(w_2))^\intercal \\ \vdots \\ (\phi(z_\ell) - \phi(w_{\ell+1}))^\intercal \end{bmatrix}, \quad E = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_\ell \end{bmatrix}.$$

Ignoring dependence of error E on parameter Θ, the least-squares solution, which we call the LSADP estimator, is

$$\widehat{\Theta} = (Z^\intercal Z)^{-1} Z^\intercal Y,$$

which is well-defined so long as $Z^\intercal Z$ is invertible, which is assured if

1) $\ell > n(n+m+p)$
(more observations than estimated parameters),
2) $\varepsilon_0 I \preceq \frac{1}{\ell}\sum_{i=1}^{\ell} \psi_{t-i}\psi_{t-i}^\intercal \preceq \bar{\varepsilon}_0 I \ \forall\ t \geq N_0$ and $N \geq N_0$
(persistency of excitation),

where $\psi_t = \phi(z_t) - \phi(w_{t+1})$, $\varepsilon_0 \leq \bar{\varepsilon}_0$, and N_0 is a positive number [25]. To ensure persistency of excitation almost surely, we add stochastic Gaussian exploration noise to the control and adversary inputs.

The LSADP estimator is biased because the errors e_t are correlated with the observations $c(z_t)$, which led to the development of the LSTDQ estimator [24]:

$$\widehat{\Theta} = \left(\sum_{t=1}^{\ell} \phi(z_t)(\phi(z_t) - \phi(w_{t+1})^\intercal \right)^{\dagger} \sum_{t=1}^{\ell} \phi(z_t) c_t.$$

The LSTDQ estimator uses the same rollout data as the LSADP estimator, so it can be used as a "drop-in" replacement. Using the LSTDQ estimator, we have the model-free approximate policy iteration for SLQ games in Algorithm 2. Due to the stochastic parameter estimates, an increasing rollout length ℓ is necessary to achieve decreasing error in the gains. It is not straightforward to determine this schedule or a meaningful convergence threshold, so for simplicity we use a fixed rollout length.

Algorithm 2 Approximate policy iteration for SLQ games

```
Input: Ms-stabilizing linear policy pair $(K^0, L^0)$, number of
    iterations $N$, rollout length $\ell$, variances $\sigma_u^2, \sigma_v^2$.
 1: Initialize $k = 0$
 2: while $k < N$ do
 3:     Approximate Policy Evaluation:
 4:     Generate inputs $u_t = Kx_t + u_{e,t}$, $v_t = Lx_t + v_{e,t}$ with
        exploration noises $u_{e,t} \sim \mathcal{N}(0, \sigma_u^2 I)$, $v_{e,t} \sim \mathcal{N}(0, \sigma_v^2 I)$.
 5:     Collect rollout of length $\ell + 1$ by exciting the system
        with the inputs $u_t$, $v_t$, yielding $\{z_t, c_t\}_{t=0}^{\ell+1}$.
 6:     Compute the augmented rollout $\{z_t, w_t, c_t\}_{t=0}^{\ell+1}$ under
        policy pair $(K^k, L^k)$.
 7:     Compute $\widehat{H} = \widehat{H}_{KL}^k$ using LSTDQ estimation as
            $\widehat{\Theta}_{KL}^k = \left( \sum_{t=1}^{\ell} \phi(z_t)(\phi(z_t) - \phi(w_{t+1})^\intercal \right)^{\dagger} \sum_{t=1}^{\ell} \phi(z_t) c_t$
            $\widehat{H}_{KL}^k = \mathrm{smat}(\widehat{\Theta})$
 8:     Policy Improvement: Update the policy pair according
        to the dynamic programming equations
            $K^{k+1} = (\widehat{H}_{uu} - \widehat{H}_{uv}\widehat{H}_{vv}^{-1}\widehat{H}_{vu})^{-1}(\widehat{H}_{uv}\widehat{H}_{vv}^{-1}\widehat{H}_{vx} - \widehat{H}_{ux})$
            $L^{k+1} = (\widehat{H}_{vv} - \widehat{H}_{vu}\widehat{H}_{uu}^{-1}\widehat{H}_{uv})^{-1}(\widehat{H}_{vu}\widehat{H}_{uu}^{-1}\widehat{H}_{ux} - \widehat{H}_{vx})$
 9:     $k \leftarrow k + 1$
10: end while
Output: Approximately optimal policy pair $(K^k, L^k)$
```

V. Numerical Experiments

In the following example we consider the problem of stabilizing a true system given knowledge only of a nominal system with both parameter mis-specification as well as unmodeled dynamics.

True system: Consider a pair of masses coupled via springs in series to fixed walls and eachother. The first mass is heavier with mass m_1 and is connected to a fixed wall by a spring with *negative* constant k_1 such that the force is not restorative, but repulsive. The second mass is less heavy with mass m_2 and is connected by springs with small positive constant to another fixed wall (k_2) and the first mass (k_3). Both masses are subject to viscous friction. An actuator applies force $F = bu$ to the first mass proportional to the input by strength constant b. The continuous-time dynamics of the true system are thus

$$\begin{bmatrix} \dot{p}_1 \\ \ddot{p}_1 \\ \dot{p}_2 \\ \ddot{p}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_1+k_3}{m_1} & -\frac{c_1}{m_1} & -\frac{k_1}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_2}{m_2} & 0 & -\frac{k_2+k_3}{m_2} & -\frac{c_2}{m_2} \end{bmatrix}}_{A_{\text{true},\,c}} \begin{bmatrix} p_1 \\ \dot{p}_1 \\ p_2 \\ \dot{p}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ b \\ 0 \\ 0 \end{bmatrix}}_{B_{\text{true},\,c}} u,$$

where p_1, p_2, $\dot{p}_1$, and $\dot{p}_2$ are the positions and velocities of masses 1 and 2 respectively. We choose the parameters $k_1 = -1.00$, $k_2 = 1.00$, $k_3 = 0.10$, $m_1 = 0.6$, $m_2 = 100$, $c_1 = 0.10$, $c_2 = 10$, $b = 1.0$. Under a forward Euler discretization with step size $\Delta t = 0.5$ we obtain the discrete-time dynamics $x_{t+1} = A_{\text{true}} x_t + B_{\text{true}} u_t$ where $A_{\text{true}} = I + A_{\text{true},\,c}\Delta t$, $B_{\text{true}} = B_{\text{true},\,c}\Delta t$.

Nominal system: The nominal model neglects the dynamics of the second mass i.e. treats it as a fixed wall. Thus, the system matrices are

$$A = \begin{bmatrix} 1 & \Delta t \\ -\left(\frac{k_1+k_3}{m_1}\right)\Delta t & 1 - \left(\frac{c_1}{m_1}\right)\Delta t \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ b\Delta t \end{bmatrix}.$$

From the full dynamics of the true model, it is apparent that the effect of the dynamics of the second mass $(p_2, \dot{p}_2)$ on the first mass is only on $\ddot{p}_1$, so we model uncertainty due to the unmodeled dynamics as an adversary with input matrix $C = \begin{bmatrix} 0 & \Delta t \end{bmatrix}^\intercal$. For the cost weight matrices we chose

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1, \quad S = 7.$$

The nominal model assumes both a smaller spring constant of the first block, $k_1 = 0.7$, and a larger actuator strength, $b = 1.2$, than the true model. It is easy to see that parametric uncertainty on the spring constant k_1 and actuator strength b can be modeled naturally by multiplicative noises with directions

$$A_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

For the variances, we chose $\sigma_{\alpha,1}^2 = 0.8$ and $\sigma_{\beta,1}^2 = 0.4$.

We evaluated the following control synthesis methods on the nominal model:

1) Open-loop response (OL) i.e. with zero gain
2) Certainty-equivalent (CE) i.e. no mult. noise or adversary
3) Multiplicative noise only (MN) i.e. no adversary

4) Dynamic game adversary only (DG) i.e. no mult. noise
5) Combined mult. noise and adversarial input (MNDG)
6) LQR optimal control given knowledge of the full dynamics of the true system (OPT)

The stability ($\rho(A+BK)$) and performance ($\mathrm{Tr}(P_K)$) of each control when applied to the true system are summarized in Table I. For this example, excluding the baseline control OPT, only the control MNDG was able to stabilize the true system, while the other methods CE, MN, DG were unable to do so. This can be seen in the fact that only control MNDG achieves $\rho(A+BK) < 1$ and finite cost $\mathrm{Tr}(P_K)$. This means that our approach of combining multiplicative noise and adversarial input control design achieves superior robustness when compared with methods which only accounted for a single type of uncertainty or did not account for uncertainty at all. These conclusions persist using different Q and R if S is chosen small enough.

Due to space constraints we have omitted the results of numerical experiments for the model-free setting, but given sufficient sample data the quality of the synthesized controls is only slightly degraded compared to the exact case.

Code which implements the algorithms and experiments described in this paper are available from the web link: https://github.com/TSummersLab/policy-iteration-slq-games.

TABLE I
ROBUST CONTROL OF SPRING-COUPLED MASSES

Method	$\rho(A+BK)$	$\mathrm{Tr}(P_K)$	K
OL	1.570	∞	$\begin{bmatrix}+0.0 & +0.0 & +0.0 & 0.0\end{bmatrix}$
CE	1.031	∞	$\begin{bmatrix}-1.6 & -1.5 & +0.0 & 0.0\end{bmatrix}$
MN	1.021	∞	$\begin{bmatrix}-1.7 & -1.6 & +0.0 & 0.0\end{bmatrix}$
DG	1.018	∞	$\begin{bmatrix}-1.8 & -1.7 & +0.0 & 0.0\end{bmatrix}$
MNDG	0.994	31074	$\begin{bmatrix}-2.3 & -2.1 & +0.0 & 0.0\end{bmatrix}$
OPT	0.965	3019	$\begin{bmatrix}-2.6 & -2.0 & -2.1 & 2.9\end{bmatrix}$

VI. CONCLUSIONS

We developed a general framework and effective solution algorithm for performing optimal control synthesis for linear quadratic games with stochastic parameters. We envision this framework being applied by practitioners in the following two broad use-cases:

1) Enhancing stability robustness of a deterministic linear system with unknown parameters to both structured and unstructured model errors.
2) Enhancing ms-stability robustness of linear systems with stochastic parameters to unstructured model errors.

Future work will go towards proving finite-sample convergence results for the model-free approximate policy iteration, which was empirically observed here.

ACKNOWLEDGMENT

This material is based on work supported by the United States Air Force Office of Scientific Research under award number FA2386-19-1-4073.

REFERENCES

[1] G. Hewer, "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Transactions on Automatic Control*, vol. 16, no. 4, pp. 382–384, 1971.
[2] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: a dynamic game approach.* Springer Science & Business Media, 2008.
[3] E. Mageirou, "Values and strategies for infinite time linear quadratic games," *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 547–550, August 1976.
[4] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to H-infinity control," *International journal of robust and nonlinear control*, vol. 4, no. 4, pp. 421–448, 1994.
[5] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, 2007.
[6] K. Zhang, Z. Yang, and T. Basar, "Policy optimization provably converges to Nash equilibria in zero-sum linear quadratic games," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 598–11 610.
[7] S. Tu and B. Recht, "The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint," in *Conference on Learning Theory*, 2019, pp. 3036–3083.
[8] W. M. Wonham, "Optimal stationary control of a linear system with state-dependent noise," *SIAM Journal on Control*, vol. 5, no. 3, pp. 486–500, 1967.
[9] W. L. De Koning, "Compensatability and optimal compensation of systems with white parameters," *IEEE Transactions on Automatic Control*, vol. 37, no. 5, pp. 579–588, May 1992.
[10] Yulin Huang, Weihai Zhang, and Huanshui Zhang, "Infinite horizon LQ optimal control for discrete-time stochastic systems," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1, June 2006, pp. 252–256.
[11] T. Wang, H. Zhang, and Y. Luo, "Stochastic linear quadratic optimal control for model-free discrete-time systems based on Q-learning algorithm," *Neurocomputing*, vol. 312, pp. 1 – 8, 2018.
[12] A. El Bouhtouri, D. Hinrichsen, and A. J. Pritchard, "H-infinity-type control for discrete-time stochastic systems," *International Journal of Robust and Nonlinear Control*, vol. 9, no. 13, pp. 923–948, 1999.
[13] Weihai Zhang, Yulin Huang, and Lihua Xie, "Stochastic H2/H-infinity control for discrete-time systems with multiplicative noise in state and control input: Infinite horizon case," in *2007 46th IEEE Conference on Decision and Control*, Dec 2007, pp. 5407–5412.
[14] D. Bernstein, "Robust static and dynamic output-feedback stabilization: Deterministic and stochastic perspectives," *IEEE Transactions on Automatic Control*, vol. 32, no. 12, pp. 1076–1084, December 1987.
[15] B. Gravell, P. M. Esfahani, and T. Summers, "Robust control design for linear systems via multiplicative noise," *IFAC Proceedings Volumes*, 2020, 18th IFAC World Congress (to appear), arXiv preprint arXiv:2004.08019.
[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
[17] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS).* IEEE, 2017, pp. 23–30.
[18] T. Damm, *Rational matrix equations in stochastic control.* Springer Science & Business Media, 2004, vol. 297.
[19] B. Gravell, P. M. Esfahani, and T. Summers, "Learning robust control for LQR systems with multiplicative noise via policy gradient," *arXiv preprint arXiv:1905.13547*, 2019.
[20] M. Athans, R. Ku, and S. Gershwin, "The uncertainty threshold principle: Some fundamental limitations of optimal decision making under dynamic uncertainty," *IEEE Transactions on Automatic Control*, vol. 22, no. 3, pp. 491–495, 1977.
[21] R. A. Howard, *Dynamic programming and Markov processes.* John Wiley, 1960.
[22] T. Damm and D. Hinrichsen, "Newton's method for a rational matrix equation occurring in stochastic control," *Linear Algebra and its Applications*, vol. 332-334, pp. 81 – 109, 2001.
[23] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proceedings of 1994 American Control Conference-ACC'94*, vol. 3. IEEE, 1994, pp. 3475–3479.
[24] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *J. Mach. Learn. Res.*, vol. 4, p. 1107–1149, Dec. 2003.
[25] G. C. Goodwin and K. S. Sin, "Adaptive filtering, prediction and control," 1984.