Towards Model-Free LQR Control over Rate-Limited Channels

Aritra Mitra[†] AMITRA2@NCSU.EDU
Department of Electrical and Computer Engineering, North Carolina State University
Lintao Ye[†] YELINTAO93@HUST.EDU.CN
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
Vijay Gupta GUPTA869@PURDUE.EDU
The Elmore Family School of Electrical and Computer Engineering, Purdue University[*]

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

Given the success of model-free methods for control design in many problem settings, it is natural to ask how things will change if realistic communication channels are utilized for the transmission of gradients or policies. While the resulting problem has analogies with the formulations studied under the rubric of networked control systems, the rich literature in that area has typically assumed that the model of the system is known. As a step towards bridging the fields of model-free control design and networked control systems, we ask: *Is it possible to solve basic control problems - such as the linear quadratic regulator (LQR) problem - in a model-free manner over a rate-limited channel?* Toward answering this question, we study a setting where a worker agent transmits quantized policy gradients (of the LQR cost) to a server over a noiseless channel with a finite bit-rate. We propose a new algorithm titled Adaptively Quantized Gradient Descent (`AQGD`), and prove that above a certain finite threshold bit-rate, `AQGD` guarantees exponentially fast convergence to the globally optimal policy, with *no deterioration of the exponent relative to the unquantized setting*. More generally, our approach reveals the benefits of adaptive quantization in preserving fast linear convergence rates, and, as such, may be of independent interest to the literature on compressed optimization.

Keywords: Model-free learning, Quantized optimization, Policy gradient algorithms for LQR, Rate-limited channels

1. Introduction

In recent years, there has been significant interest in analyzing the *non-asymptotic* performance of control algorithms that do not rely on any initial model of the dynamics. The body of work in this space can be broadly grouped into two categories: (i) (model-based) approaches that use data to construct empirical models, and then apply either certainty-equivalent or robust control techniques (Tsiamis et al., 2022); and (ii) (model-free) approaches that directly try to find the optimal policy from data, without maintaining an explicit estimate of the model (Fazel et al., 2018; Zhang et al., 2021; Zhao et al., 2023; Hu et al., 2023). Due to their ease of implementation, model-free policy gradient (PG) algorithms, in particular, have gained a lot of popularity. When applied to the classical linear quadratic regulator (LQR) problem (Anderson and Moore, 2007), the authors in Fazel et al. (2018) showed that despite the non-convexity of the optimization landscape, model-free PG algorithms guarantee convergence to the globally optimal policy. However, almost nothing is known about the *robustness of such PG algorithms to communication-induced distortions* that may be introduced if transmission of the gradient or the policy occurs over realistic communication channels.

[*] The first two authors contributed equally to this work.

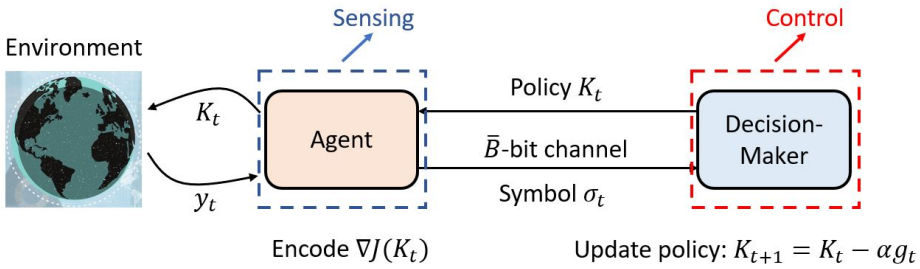

Figure 1: Communication-constrained policy optimization for LQR. At each iteration t, the decision-maker sends the current policy K_t to an agent over a noiseless channel of infinite capacity. The agent evaluates and encodes the policy gradient $\nabla J(K_t)$ using $\bar{B}$ bits, and transmits the encoded symbol σ_t to the decision-maker over a noiseless rate-limited channel. The decision-maker updates the policy based on the decoded policy gradient g_t.

This problem, where a remote sensing agent collects measurements of a dynamical process and transmits them over a communication channel to a controller, or when the controller transmits the control input to an actuator over a communication channel, has been broadly studied in the field of networked control systems in control theory. Perhaps the earliest setting studied here was when the sensor-controller channel was rate-limited while the controller-actuator communication occurred without any loss of information, and the focus was on the stabilization of the closed loop process. A celebrated result shows that even for the simplest case when the open loop plant is a linear time-invariant (LTI) system, there is a minimal bit rate (which depends on the magnitudes of the open loop unstable eigenvalues of the plant) that must be supported by the channel in order for an encoder-decoder and controller design to exist such that the plant can be stabilized (Tatikonda and Mitter, 2004; Nair and Evans, 2004). The result has been extended in various directions (e.g., Nair et al. (2007); Minero et al. (2009); Tallapragada and Cortés (2015); Martins (2006)). However, except for some limited deviations (Okano and Ishii, 2014), this line of work (or more generally, the field of networked control systems) has relied on one crucial assumption: the model of the system dynamics is *known*. As it stands, there is little to no theoretical understanding of communication-constrained control in the absence of such an assumption. In this paper, we seek to bridge this gap between the two directions of work summarized above.

Our Setup and Motivation. The goal of this work is to connect control, communication, and learning by initiating a study of model-free control under communication constraints. To that end, we introduce a new setting depicted in Figure 1.[1] As in Tatikonda and Mitter (2004), our setup involves a remote sensing agent separated from a decision-maker by a noiseless channel that can support a message of $\bar{B}$ bits per channel use. The agent interacts with an environment by executing the control policy relayed to it by the decision-maker. Assuming that the dynamics of the environment can be captured by an *unknown* LTI system, the collective objective of the agent and the decision-maker is to optimally control such dynamics by solving an LQR problem with known cost matrices.[2] To that end, the agent uses the measurements (rewards, state observations) received as feedback from the environment to construct a policy gradient (or an estimate thereof) of the LQR cost function. It then

1. The notation that appears in this figure is formally defined in Section 2.
2. Note that the objective in Tatikonda and Mitter (2004) is merely to stabilize the closed loop system.

encodes this gradient using $\bar{B}$ bits, and transmits it to the decision-maker. The decision-maker decodes the received message and then uses the resulting *inexact* policy gradient to update the policy. The motivation behind studying this abstraction is to eventually enable model-free control/reinforcement learning in multi-agent networked systems, where communication plays a key role (Lin et al., 2021; Shin et al., 2023). Given this motivation, we aim to answer the following questions.

Is it possible to design a quantized policy gradient scheme that guarantees exact convergence to the globally optimal policy? If so, is there an unavoidable loss in the rate of convergence that one incurs with a finite value of $\bar{B}$ relative to when the channel has infinite capacity?

We answer the above questions by making the following contributions.

• **Problem Formulation.** We introduce a new formulation to analyze the effects of communication constraints on the performance of the popular policy-gradient algorithm for solving the LQR problem. Our setting is inspired by two different strands of literature: the classical bit-rate-constrained control formulation (see, e.g., Tatikonda and Mitter (2004); Nair and Evans (2004)), and the more recent works on quantization in optimization (Gandikota et al., 2021; Lin et al., 2022; Mayekar and Tyagi, 2020) that, like us, also consider a single-worker single-server framework.

• **Novel Quantized Gradient Descent Scheme.** On the algorithmic front, our chief contribution is to develop `Adaptively Quantized Gradient Descent` (`AQGD`) - a novel quantized gradient descent algorithm that carefully exploits smoothness of the loss/cost function to encode the "change" (innovation) in the gradient, as opposed to the gradient itself. Our key guiding observation here is that for smooth loss functions, the gradient at the agent/worker should not change drastically from one iteration to the next. As such, it makes sense to encode the innovation in the gradient.

• **Preserving Linear Rates under Global Assumptions.** In Theorem 1, we prove that for smooth and strongly-convex loss functions, `AQGD` guarantees exponentially fast convergence to the optimal solution. Furthermore, we prove that above a finite bit-rate, the exponent of convergence for `AQGD` is *exactly* the same as that of unquantized gradient descent, i.e., *`AQGD` leads to no loss in performance relative to when the channel has infinite capacity.* Unfortunately, however, the optimization landscape for the LQR problem admits neither strong-convexity nor global smoothness. Thus, it is unclear if we can continue to use `AQGD` for our quantized policy gradient problem. Toward resolving this issue, in Theorem 2, we prove that the assertions of Theorem 1 continue to hold without any change under the weaker assumption (relative to strong convexity) of gradient-domination.

• **Preserving Linear Rates under Local Assumptions.** The above developments still leave open the following question: *Can one continue to preserve rates with quantized policy gradients when smoothness and gradient-domination only hold locally?* Moreover, to make `AQGD` amenable for the LQR problem, we need to ensure that despite the inexactness introduced by quantization, the policies generated iteratively by `AQGD` are all stabilizing. In Theorem 6, we overcome these challenges, and prove that `AQGD` continues to preserve rates under local assumptions and generates a sequence of stabilizing policies.

• **Proof Technique.** Our proofs rely on the construction of a novel Lyapunov function that simultaneously accounts for the optimization error and also the error introduced due to quantization.

Limitation. This work is only a first step in this rich area. Throughout the paper, we assume that the worker has access to *exact policy gradients.* In other words, the only source of inexactness in the policy gradients is due to quantization; however, the gradients themselves are *deterministic*. We make this assumption to focus on the unique challenges introduced by the rate-limited channel.

More Related Work. Despite the large body of work that has emerged on the topic of communication-constrained optimization (Bernstein et al., 2018; Stich et al., 2018; Gandikota et al., 2021; Mayekar and Tyagi, 2020; Richtárik et al., 2021), the only paper we are aware of that manages to preserve fast linear rates (despite quantization) is the recent work by Lin et al. (2022). In Lin et al. (2022), the authors devise an elegant new method titled "differential quantization" (DQ). The key idea behind this approach is to first compute an auxiliary sequence that mimics the trajectory of unquantized gradient descent by carefully keeping track of past quantization errors. Crucial to the DQ approach is computing the worker's gradients at the auxiliary sequence, not the true iterate sequence. For globally smooth and strongly-convex functions, it is then shown in Lin et al. (2022) that above a finite bit-rate, the DQ approach guarantees exponentially fast convergence to the optimal solution with no loss in rate relative to unquantized gradient descent. Our work differs from that of Lin et al. (2022) both algorithmically, and also in terms of results: unlike the DQ scheme, `AQGD` *does not* require maintaining any auxiliary sequence. In addition to being conceptually simpler, `AQGD` preserves rates under weaker assumptions of local smoothness and gradient-domination. As such, our proposed technique might be of independent interest to the literature on compressed optimization.

2. Problem Formulation

We begin with the standard setup in works on model-free learning in LQR problems. Consider a linear time-invariant (LTI) system given by

$$x_{t+1} = Ax_t + Bu_t + w_t,$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are system matrices, $x_t \in \mathbb{R}^n$ is the state, $u_t \in \mathbb{R}^m$ is the control input and $w_0, w_1, \ldots$ are i.i.d. disturbances with zero mean and covariance $\Sigma_w \in \mathbb{S}^n_{++}$, where $\mathbb{S}^n_{++}$ denotes the set of all positive definite $n \times n$ matrices. We also assume without loss of generality that $x_0 = 0$. The goal is to design the control policy to calculate the control inputs $u_0, u_1, \ldots$, that solve

$$\min_{u_0,u_1,\ldots} \lim_{T\to\infty} \frac{1}{T}\,\mathbb{E}\Big[\sum_{t=0}^{T-1} x_t^\top Q x_t + u_t^\top R u_t\Big], \tag{1}$$

where $Q \in \mathbb{S}^n_{++}$ and $R \in \mathbb{S}^m_{++}$ are cost matrices, and the expectation is taken with respect to the disturbance process $\{w_t\}$. It is well known that (1) can be solved by considering the static state-feedback control policy $u_t = Kx_t$ for some controller $K \in \mathbb{R}^{m\times n}$ (see, e.g., Bertsekas (2015)). In other words, (1) may be equivalently cast into the following form:

$$\min_{K\in\mathbb{R}^{m\times n}} J(K) \triangleq \lim_{T\to\infty} \frac{1}{T}\,\mathbb{E}\Big[\sum_{t=0}^{T-1} x_t^\top (Q + K^\top R K) x_t\Big]. \tag{2}$$

Moreover, we know from e.g. Bertsekas (2015) that the value of the cost function $J(K)$ is finite if and only if K is stabilizing, i.e., the matrix $(A + BK)$ is Schur-stable. One can then solve (2) by minimizing $J(K)$ over the set of stabilizing K. In particular, if K is stabilizing, $J(K)$ yields the following closed-form expression (Bertsekas, 2015):

$$J(K) = \mathrm{trace}(P_K \Sigma_w) = \mathrm{trace}\big((Q + K^\top R K)\Sigma_K\big), \tag{3}$$

where P_K and Σ_K are positive definite solutions to the following Riccati equations:

$$P_K = Q + K^\top R K + (A + BK)^\top P_K (A + BK),\ \Sigma_K = \Sigma_w + (A + BK)^\top \Sigma_K (A + BK). \tag{4}$$

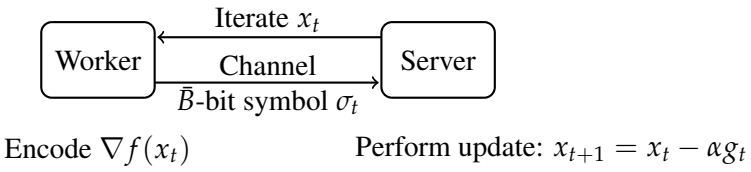

Figure 2: Communication-constrained optimization setup. This setup is analogous to that in Fig. 1, with g_t representing the decoded gradient at the server.

Even if the system matrices A and B are not known, the above properties on $J(\cdot)$ can be utilized to calculate $K^* = \operatorname{argmin}_K J(K)$ using the so-called policy gradient method (see, e.g., Mårtensson and Rantzer (2009); Malik et al. (2020); Hu et al. (2023); Bu et al. (2020); Fatkhullin and Polyak (2021)). The basic scheme is to initialize with an arbitrary stabilizing K_0 and iteratively perform updates of the form: $K_{t+1} = K_t - \alpha \nabla J(K_t)$ for $t = 0, 1, \ldots$, where $\alpha \in \mathbb{R}_{>0}$ is a step size. As shown in Fazel et al. (2018); Malik et al. (2020), if the exact gradient $\nabla J(K_t)$ of $J(K_t)$ is available, then the above algorithm converges exponentially fast to the optimal policy K^*, even though the cost $J(K)$ is not strictly convex. Further, even without the knowledge of the system model, $J(K)$ and $\nabla J(K)$ for a given K can be accurately estimated based on observed system trajectories of (2) obtained by applying the control policy $u_t = Kx_t$; hence, the name model-free learning for LQR.

Objective. We are interested in model-free learning for LQR under communication constraints on the policy gradient updates. To that end, we consider the setup in Fig. 1. In this setup, policy gradients computed by a worker agent are transmitted to the decision maker (or a server) who then updates the policy. Crucially, the transmission from the worker agent to the decision maker occurs across a channel that supports noise-free transmission of a finite number of bits $\bar{B}$ per use of the channel. Our **goal** then is to design (i) an encoding scheme at the worker, and (ii) a policy update rule at the decision-maker, such that the resulting quantized policy gradient algorithm continues to guarantee convergence to the optimal solution K^* (if possible). Furthermore, we seek to identify the rate of convergence as a function of the capacity $\bar{B}$ of the channel. The challenge here lies in the fact that the channel distorts the policy gradients; as such, *it is a priori unclear whether the sequence of policies generated using such distorted policy gradients remain stabilizing, or converge to* K^*. To focus on this challenge, we will assume throughout that the worker has access to exact deterministic policy gradients. Beyond this assumption, however, we do not require the worker or the decision-maker to possess any knowledge of the system model.

Communication-Constrained Optimization. To make concrete progress toward the above objective, we will find it convenient to first analyze the communication-constrained optimization setup shown in Fig. 2. Suppose for the moment that a function $f : \mathbb{R}^d \to \mathbb{R}$ which is L-smooth and μ-strongly convex is sought to be minimized. It is well-known that when the channel from the worker to the server has infinite capacity, i.e., when $\bar{B} = \infty$, running gradient descent $x_{t+1} = x_t - \alpha \nabla f(x_t)$ with a step-size $\alpha = 1/L$ provides a convergence guarantee of the following form $\forall t \geq 0$:

$$f(x_t) - f(x^*) \leq \left(1 - \frac{1}{\kappa}\right)^t \left(f(x_0) - f(x^*)\right). \tag{5}$$

Algorithm 1 Adaptively Quantized Gradient Descent (AQGD)

Initialization: $x_0 = 0$, $g_{-1} = 0$, contraction factor γ, and pick R_0 such that $\|\nabla f(x_0)\|_2 \leq R_0$.
For $t = 0, 1, \ldots,$ **do**
  At Worker:
  Receive iterate x_t, gradient estimate g_{t-1}, and range R_t from server.
  Compute innovation $i_t = \nabla f(x_t) - g_{t-1}$.
  If $i_t \in \mathcal{B}_d(0, R_t)$, encode the innovation: $\tilde{i}_t = \mathcal{Q}_{b,R_t}(i_t)$.
  At Decision-Maker/Server:
  Decode $\tilde{i}_t$, and estimate current gradient: $g_t = g_{t-1} + \tilde{i}_t$.
  Update the model as follows:

$$x_{t+1} = x_t - \alpha g_t. \tag{6}$$

  Update the range of the quantizer map as follows:

$$R_{t+1} = \gamma R_t + \alpha L \|g_t\|_2. \tag{7}$$

End For

Here, $\kappa = L/\mu$ is the condition number of f, and $x^* = \operatorname{argmin}_{x \in \mathbb{R}^d} f(x)$ is the unique minimizer of f. If instead, one employs the rule: $x_{t+1} = x_t - \alpha g_t$, where g_t is a *quantized* version of the gradient $\nabla f(x_t)$, the guarantee in Eq. (5) may no longer hold: the typical convergence rate achieved by almost all existing compressed gradient descent algorithms (see, e.g., Stich and Karimireddy (2019)) is $O(\exp(-\frac{t}{\kappa\delta}))$, where $\delta \geq 1$ captures the level of compression. Clearly, a higher δ introduces more distortion and causes the exponent of convergence to be a factor of δ slower than that of gradient descent. Given this premise, we will proceed in two steps. First, in Section 3, we will introduce our proposed algorithm AQGD, and prove (in Section 4) that above a finite threshold for the bit-rate, it preserves the exact same convergence rate as for unquantized gradient descent in Eq. (5). Next, in Section 5, we will show how AQGD can be applied to the LQR problem.

3. Adaptively Quantized Gradient Descent (AQGD)

In this section, we will develop our proposed approach (Algorithm 1) titled Adaptively Quantized Gradient Descent (AQGD). We start with the simplest building block of AQGD - a scalar quantizer.

Scalar Quantizer. Suppose we are given a vector $X \in \mathbb{R}^d$ such that $\|X\|_2 \leq R$, and we wish to encode each component of this vector using b bits. Clearly, $X_i \in [-R, R], \forall i \in [d]$, where X_i is the i-th component of X. For each $i \in [d]$, to encode X_i, we simply partition the interval $[-R, R]$ into 2^b bins of equal width, and set the center $\tilde{X}_i$ of the bin containing X_i to be the quantized version of X_i. This yields $\tilde{X} = [\tilde{X}_1, \ldots, \tilde{X}_d]^T$ as the quantized version of X. To succinctly describe the above operation, we will use a quantizer map $\mathcal{Q}_{b,R} : \mathbb{R}^d \to \mathbb{R}^d$ that is parameterized by the number of bits b used to encode each component of the input, and the range of each component R. Thus, given $\|X\|_2 \leq R$, we have $\tilde{X} = \mathcal{Q}_{b,R}(X)$. We are now in a position to describe AQGD.

Description of AQGD. Let g_{t-1} represent the estimate of the gradient $\nabla f(x_{t-1})$ at the decision maker or the server in iteration $t-1$. Now, since the function f is smooth, the new gradient $\nabla f(x_t)$ at the worker cannot change abruptly from what it was at the previous iteration, namely $\nabla f(x_{t-1})$. This simple observation suggests that if the decision-maker has a reasonably good estimate g_{t-1} of

the true gradient $\nabla f(x_{t-1})$ at iteration $t-1$, then such an estimate cannot be too different from $\nabla f(x_t)$. As such, it makes sense to encode the "innovation" signal $i_t = \nabla f(x_t) - g_{t-1}$, as opposed to the gradient $\nabla f(x_t)$ itself. In words, the innovation is the *new information* in the worker's gradient at iteration t, relative to the most recent estimate of the gradient held by the decision-maker from iteration $t-1$. Now if our algorithm operates correctly, then it should be that $\nabla f(x_t) \to 0$. This, in turn, would imply that the sequence $\{\nabla f(x_t)\}$ is Cauchy, i.e., the gap between consecutive gradients should eventually shrink to 0. Intuitively, one should thus expect the innovation signal i_t to be contained in balls of progressively smaller radii. Our key idea is to maintain estimates of the radii of such balls, and use this information to refine the range of the quantizer used to encode the innovation.

With the above intuition in place, we now explain the steps of Algorithm 1. The decision-maker or the server starts out with an initial iterate $x_0 = 0$, an initial gradient estimate $g_{-1} = 0$, and an initial upper bound R_0 on $\|\nabla f(x_0)\|_2$. At each iteration t, the server transmits the current iterate x_t, the gradient estimate g_{t-1}, and the dynamic quantizer range R_t to the worker without any loss of information. The worker then computes the gradient $\nabla f(x_t)$, the innovation $i_t = \nabla f(x_t) - g_{t-1}$, and checks if $i_t \in \mathcal{B}_d(0, R_t)$; here, we use $\mathcal{B}_d(0, R_t)$ to represent the d-dimensional Euclidean ball of radius R_t centered at the origin. If so, the quantized innovation $\tilde{i}_t = \mathcal{Q}_{b,R_t}(i_t)$ is transmitted to the server (line 6). The server decodes $\tilde{i}_t$ and forms an estimate g_t of the gradient $\nabla f(x_t)$ as per line 8 of AQGD. This estimate is used to perform a gradient-descent-type update as per Eq. (6). Finally, the server updates the dynamic quantizer range R_t as per Eq. (7). In this equation, $\gamma \in (0,1)$ is a contraction factor that will be specified later. In Lemma 3, we show that our range update ensures $i_t \in \mathcal{B}_d(0, R_t), \forall t$. Before we analyze AQGD in the next section, a few comments are in order.

• **Correct Decoding**. We assume that the server is aware of the encoding strategy at the worker. Thus, since the server knows b and R_t, given the $\bar{B}$-bit symbolic encoding of $\tilde{i}_t$, where $\bar{B} = bd$, the server can decode $\tilde{i}_t$ exactly.

• **Adaptive Ranges**. Notice that the number of bits b we use to encode each component of the innovation vector remains the same across iterations. However, as time progresses, we invest our bits more carefully by dynamically updating R_t as per Eq. (7). Our analysis will reveal that $R_t \to 0$, suggesting that the region we encode becomes progressively finer.

• **Exploiting Smoothness**. As explained earlier, the idea of encoding innovations hinges on the assumption that successive gradients at the worker do not change abruptly - a condition met by smooth functions. Notice also that the rule for updating the range R_t in Eq. (7) uses the smoothness parameter L. In short, AQGD carefully exploits smoothness for quantization.

4. Convergence Results and Analysis for AQGD

Our main result pertaining to the convergence performance of AQGD is as follows.

Theorem 1 *(**Convergence of AQGD**) Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and μ-strongly convex. Suppose AQGD (Algorithm 1) is run with step-size $\alpha = 1/(6L)$ and contraction factor $\gamma = \sqrt{d}/2^b$. There exists a universal constant $C \geq 1$ such that if the bit-precision b per component satisfies*

$$b \geq C \log\left(\frac{d\kappa}{\kappa - 1}\right), \tag{8}$$

then the following is true $\forall t \geq 0$:

$$f(x_t) - f(x^*) \leq \left(1 - \frac{1}{12\kappa}\right)^t \left(f(x_0) - f(x^*) + \alpha R_0^2\right), \quad \text{where} \quad \kappa = L/\mu. \tag{9}$$

Before providing a proof sketch for the above result, we first discuss its key implications.

Discussion. Comparing Eq. (9) to Eq. (5), we immediately note that AQGD *preserves the exact same linear rate of convergence (up to universal constants) as vanilla unquantized gradient descent*, provided the channel capacity $\bar{B} = bd$ satisfies the requirement on b in Eq. (8). This result is significant since it is the *only* one we are aware of - other than that in Lin et al. (2022) - which establishes linear convergence rates can be *exactly* preserved despite quantization. As mentioned earlier, commonly used compression schemes, including sophisticated ones like error-feedback (Stich et al., 2018), cause the exponent of convergence to get scaled down by a factor $\delta \geq 1$ that captures the level of compression. Our chief contribution is to show that such a scale-down of the rate can be avoided completely, *without the need for maintaining an auxiliary sequence as in Lin et al. (2022)*.

On Minimal Bit-Rates. The authors in Lin et al. (2022) prove a converse result showing that to match the rate of unquantized gradient descent, a necessary condition on the bit-rate is

$$b \geq \log\left(\frac{\kappa+1}{\kappa-1}\right).$$

Comparing the above *minimal* rate with that for AQGD in Eq. (8), we see that there is an additional logarithmic dependence on d in Eq. (8). This dependence can be directly attributed to our choice of the uniform scalar quantizer to encode the innovation in line 6 of Algorithm 1 – a choice dictated by ease of implementation. At the expense of using a more involved vector quantizer, one can easily shave off the additional $\log(d)$ factor in Eq. (8); see Mitra et al. (2024) for more details on this.

Theorem 1 assumes strong convexity. Can we hope to achieve a similar result under the weaker gradient-domination condition? Our next result establishes that this is, indeed, possible.

Theorem 2 *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and satisfies the following gradient-domination property:*

$$\|\nabla f(x)\|_2^2 \geq 2\mu(f(x) - f(x^*)), \forall x \in \mathbb{R}^d, \tag{10}$$

where $x^* \in \operatorname{argmin}_{x\in\mathbb{R}^d} f(x)$. Let α, γ, and the bit-precision b be chosen as in Theorem 1. Then, AQGD provides exactly the same guarantee as in Eq. (9).

While the above result generalizes Theorem 1, it still requires smoothness and gradient-domination to hold globally. However, for the LQR problem of interest to us, these properties only hold *locally*. Nonetheless, in the next section, we will show how our developments thus far can still be extended to the LQR setting. Before we do so, we provide a proof sketch for Theorem 2 that naturally also applies to Theorem 1. In the interest of space, a complete proof is deferred to Mitra et al. (2024).

Proof Sketch for Theorem 2. The key technical innovation in our analysis lies in carefully defining a new Lyapunov (potential) function candidate, and showing that this function contracts over time. Our choice of a Lyapunov function candidate is the following:

$$V_t \triangleq z_t + \alpha R_t^2, \quad \text{where} \quad z_t = f(x_t) - f(x^*). \tag{11}$$

For analyzing vanilla unquantized gradient descent, it suffices to use z_t (defined in Eq. (11)) as the Lyapunov function (Bubeck et al., 2015). Unfortunately, such a choice is insufficient for our situation since due to the nature of the AQGD algorithm, the dynamics of the iterate x_t are intimately coupled with the errors induced by quantization. A measure of such quantization-induced errors turns out to be the dynamic range R_t. As such, to study the *joint evolution* of x_t and R_t, we introduce

the Lyapunov function candidate in Eq. (11). However, just introducing V_t is not enough: we need to argue that V_t decays to 0 exponentially fast at the same rate as unquantized gradient descent. Toward that end, we will crucially rely on the following two lemmas.

Lemma 3 *(**No Overflow and Quantization Error**) Suppose f is L-smooth. The following are then true for all $t \geq 0$: (i) $\|i_t\|_2 \leq R_t$; and (ii) $\|e_t\|_2 \leq \gamma R_t$, where $e_t = \nabla f(x_t) - g_t$.*

The first part of the above result tells us that the innovation i_t always belongs to $\mathcal{B}_d(0, R_t)$, i.e., there is never any need for transmitting an overflow symbol. This justifies the update for the quantizer range in Eq. (7). The second part of Lemma 3 reveals that the quantization error e_t can be conveniently bounded by the dynamic range R_t. Thus, if $R_t \to 0$, then $e_t \to 0$. To argue that the dynamic range R_t does, in fact, converge to 0, we will require the following result.

Lemma 4 *(**Recursion for Dynamic Range**) Suppose f is L-smooth. If α is such that $\alpha L \leq 1$, then for all $t \geq 0$, we have:*

$$R_{t+1}^2 \leq 8\gamma^2 R_t^2 + 2\alpha^2 L^2 \|\nabla f(x_t)\|_2^2. \tag{12}$$

The above lemma establishes a recursion for R_t which depends on the magnitude of the gradient $\nabla f(x_t)$. We can immediately see that to reason about the long-run behavior of R_t, we need to understand how such behavior relates to that of x_t. This is precisely what motivates the choice of the potential function V_t in Eq. (11). The remainder of the proof constitutes two steps. Using the two lemmas above along with smoothness and gradient-domination, we establish

$$V_{t+1} \leq \underbrace{\left(1 - \frac{\alpha\mu}{2}\right) z_t}_{T_1} + \underbrace{9\alpha\gamma^2 R_t^2}_{T_2}.$$

In the above display, T_1 represents the optimization error while T_2 represents the quantization error. To achieve the final rate in Eq. (9), we need the quantization error to decay faster than the optimization error. The requirement on the bit-rate b (as stated in Theorem 1) ensures that this condition holds. For details of these steps, we refer the reader to Mitra et al. (2024).

5. Quantized Policy Gradient for the Linear Quadratic Regulator Problem

We now extend the algorithm design and analysis in Section 3 to $f : \mathbb{R}^d \to \mathbb{R}$ with only local properties, which will be applicable to the LQR problem described in Section 2. We need to resolve two major challenges. First, the objective function $J(\cdot)$ in (2) does not possess the global L-smoothness and gradient-domination properties as required by Theorem 2 (Fazel et al., 2018). More importantly, we need to ensure that all the iterates $K_0, K_1, \ldots$ stay in the set of stabilizing controllers. To proceed, we first show that the results in Section 4 can be extended to general functions $f : \mathbb{R}^d \to \mathbb{R}$ with an imposed feasible set $\mathcal{X} \subseteq \mathbb{R}^d$, and then specialize the extension to the LQR problem setting. We introduce the following definitions.

Definition 5 *(**Locally Smooth**) A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be locally (L, D)-smooth over $\mathcal{X} \subseteq \mathbb{R}^d$ if $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$ for all $x \in \mathcal{X}$ and all $y \in \mathbb{R}^d$ with $\|y - x\|_2 \leq D$.*

We then have the following result; the proof can be found in Mitra et al. (2024).

Theorem 6 *(**Convergence of AQGD under Local Assumptions**) Consider $f : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ and $\mathcal{X} = \{x \in \mathbb{R}^d : f(x) \leq v\}$, where $v \in \mathbb{R}_{\geq 0}$. Suppose $f(\cdot)$ is (L, D)-smooth, $\|\nabla f(x)\|_2 \leq G$ for all $x \in \mathcal{X}$, and $f(\cdot)$ satisfies the following local gradient-domination property:*

$$\|\nabla f(x)\|_2^2 \geq 2\mu(f(x) - f(x^*)), \forall x \in \mathcal{X}, \tag{13}$$

where $x^* \in \operatorname{argmin}_{x \in \mathcal{X}} f(x)$. Suppose AQGD (Algorithm 1) is initialized with $x_0 \in \mathbb{R}^d$ such that $f(x_0) \leq v/2$, and run with step-size $\alpha \leq \min\{D/(2G), v/(2G^2), 1/(6L)\}$. Let the contraction factor γ be the same as in Theorem 2. Then, for all $t \geq 0$, $x_t \in \mathcal{X}$ and the following is true:

$$f(x_t) - f(x^*) \leq \left(1 - \frac{\alpha\mu}{2}\right)^t \left(f(x_0) - f(x^*) + \alpha R_0^2\right). \tag{14}$$

Discussion. It was shown in, e.g., Fazel et al. (2018); Cassel and Koren (2021), that the objective function $J(\cdot)$ in the LQR problem given by (2) possesses the properties required by Theorem 6. Specifically, Fazel et al. (2018); Cassel and Koren (2021) characterize the objects G, L, μ, and D for $J(\cdot)$ in terms of the problem parameters of (2). As such, one can apply our proposed algorithm AQGD - exactly as in Section 3 - to solve the LQR problem (2). In this context, *Theorem 6 reveals that despite the inexactness due to quantization, AQGD guarantees exponentially fast convergence to the globally optimal solution of the LQR optimization problem.* To identify the rate of convergence, we note that if $v/(2G^2) \leq D/(2G)$ or $v/(2G^2) \leq 1/(6L)$ holds, the choice of the step size becomes $\alpha \leq \min\{\frac{D}{2G}, \frac{1}{6L}\}$, and the convergence rate of AQGD in Theorem 6 matches with that of the unquantized gradient descent algorithm given in Cassel and Koren (2021) for objective functions $f(\cdot)$ with the local properties described in Theorem 6. In fact, one may argue that the local properties of $J(\cdot)$ characterized in Fazel et al. (2018); Cassel and Koren (2021) naturally lead to the choice of the above step size; detailed arguments can be found in Mitra et al. (2024). For the LQR problem, the feasible set $\mathcal{X}$ comprises the set of stabilizing controllers. Theorem 6 tells us that if $x_0 \in \mathcal{X}$, then $x_t \in \mathcal{X}, \forall t \geq 0$, i.e., the sequence of iterates/policies generated by AQGD remain stabilizing. The proof of this result - provided in Mitra et al. (2024) - is a variation on that of Theorem 2, and relies on a careful inductive argument. We conclude this section with a remark on implementation.

Remark 7 *Note that both K and $\nabla J(K)$ are matrices in $\mathbb{R}^{m \times n}$, while our analysis here is conducted with vectors $x, \nabla f(x) \in \mathbb{R}^d$. Nonetheless, one can simply vectorize K and $\nabla J(K)$ to be vectors in $\mathbb{R}^{m \times n}$, and then apply Algorithm 1 to achieve the convergence result provided in Theorem 6.*

6. Conclusions and Future Directions

With the aim of merging model-free control with the area of networked control systems, we studied how policy gradient algorithms for the LQR problem are affected by communication constraints. Specifically, we considered a rate-limited channel and introduced a novel adaptively quantized gradient-descent algorithm titled AQGD. We showed that under both global and local assumptions of smoothness and gradient-domination, AQGD guarantees exponentially fast convergence to the globally optimal solution. Most importantly, above a finite bit-rate, the exponent of convergence of AQGD remains unaffected by quantization. We finally argued how our results have immediate implications for the LQR problem. Our work opens up various interesting directions for future work: one may consider (i) noisy estimated gradients, (ii) more complex channel models, (iii) model-free control problems beyond the LQR setting, and (iv) multi-agent environments. Can one continue to preserve rates in these more involved settings? This remains to be seen.

References

Brian DO Anderson and John B Moore. *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *Proc. International Conference on Machine Learning*, pages 560–569, 2018.

Dimitri P Bertsekas. Dynamic programming and optimal control 4th edition, volume ii. *Athena Scientific*, 2015.

Jingjing Bu, Afshin Mesbahi, and Mehran Mesbahi. On topological properties of the set of stabilizing feedback gains. *IEEE Transactions on Automatic Control*, 66(2):730–744, 2020.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Asaf Cassel, Alon Cohen, and Tomer Koren. Logarithmic regret for learning linear quadratic regulators efficiently. In *Proc. International Conference on Machine Learning*, pages 1328–1337, 2020.

Asaf B Cassel and Tomer Koren. Online policy gradient for model free learning of linear quadratic regulators with $\sqrt{T}$ regret. In *Proc. International Conference on Machine Learning*, pages 1304–1313, 2021.

Xinyi Chen and Elad Hazan. Black-box control for linear dynamical systems. In *Proc. Conference on Learning Theory*, pages 1114–1143, 2021.

Ilyas Fatkhullin and Boris Polyak. Optimizing static linear feedback: Gradient method. *SIAM Journal on Control and Optimization*, 59(5):3887–3911, 2021.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *Proc. International conference on machine learning*, pages 1467–1476, 2018.

Venkata Gandikota, Daniel Kane, Raj Kumar Maity, and Arya Mazumdar. vqsgd: Vector quantized stochastic gradient descent. In *Proc. International Conference on Artificial Intelligence and Statistics*, pages 2197–2205, 2021.

Bin Hu, Kaiqing Zhang, Na Li, Mehran Mesbahi, Maryam Fazel, and Tamer Başar. Toward a theoretical foundation of policy optimization for learning control policies. *Annual Review of Control, Robotics, and Autonomous Systems*, 6:123–158, 2023.

Chung-Yi Lin, Victoria Kostina, and Babak Hassibi. Differentially quantized gradient methods. *IEEE Transactions on Information Theory*, 2022.

Yiheng Lin, Guannan Qu, Longbo Huang, and Adam Wierman. Multi-agent reinforcement learning in stochastic networked systems. *Advances in neural information processing systems*, 34:7825–7837, 2021.

Dhruv Malik, Ashwin Pananjady, Kush Bhatia, Koulik Khamaru, Peter L Bartlett, and Martin J Wainwright. Derivative-free methods for policy optimization: Guarantees for linear quadratic systems. *Journal of Machine Learning Research*, 21(21):1–51, 2020.

Karl Mårtensson and Anders Rantzer. Gradient methods for iterative distributed control synthesis. In *Proc. IEEE Conference on Decision and Control*, pages 549–554, 2009.

Nuno C Martins. Finite gain lp stabilization requires analog control. *Systems & control letters*, 55 (11):949–954, 2006.

Prathamesh Mayekar and Himanshu Tyagi. Ratq: A universal fixed-length quantizer for stochastic optimization. In *Proc. International Conference on Artificial Intelligence and Statistics*, pages 1399–1409, 2020.

Paolo Minero, Massimo Franceschetti, Subhrakanti Dey, and Girish N Nair. Data rate theorem for stabilization over time-varying feedback channels. *IEEE Transactions on Automatic Control*, 54 (2):243–255, 2009.

Aritra Mitra, Lintao Ye, and Vijay Gupta. Towards model-free lqr control over rate-limited channels. *arXiv preprint arXiv:2401.01258*, 2024.

Girish N Nair and Robin J Evans. Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM Journal on Control and Optimization*, 43(2):413–436, 2004.

Girish N Nair, Fabio Fagnani, Sandro Zampieri, and Robin J Evans. Feedback control under data rate constraints: An overview. *Proceedings of the IEEE*, 95(1):108–137, 2007.

Kunihisa Okano and Hideaki Ishii. Minimum data rate for stabilization of linear systems with parametric uncertainties. *arXiv preprint arXiv:1405.5932*, 2014.

Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34: 4384–4396, 2021.

Sungho Shin, Yiheng Lin, Guannan Qu, Adam Wierman, and Mihai Anitescu. Near-optimal distributed linear-quadratic regulator for networked systems. *SIAM Journal on Control and Optimization*, 61(3):1113–1135, 2023.

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.

Pavankumar Tallapragada and Jorge Cortés. Event-triggered stabilization of linear systems under bounded bit rates. *IEEE Transactions on Automatic Control*, 61(6):1575–1589, 2015.

Sekhar Tatikonda and Sanjoy Mitter. Control under communication constraints. *IEEE Transactions on Automatic Control*, 49(7):1056–1068, 2004.

Anastasios Tsiamis, Ingvar Ziemann, Nikolai Matni, and George J Pappas. Statistical learning theory for control: A finite sample perspective. *arXiv preprint arXiv:2209.05423*, 2022.

Kaiqing Zhang, Bin Hu, and Tamer Basar. Policy optimization for h_2 linear control with h_∞ robustness guarantee: Implicit regularization and global convergence. *SIAM Journal on Control and Optimization*, 59(6):4081–4109, 2021.

Feiran Zhao, Keyou You, and Tamer Başar. Global convergence of policy gradient primal-dual methods for risk-constrained lqrs. *IEEE Transactions on Automatic Control*, 2023.