Dual-loop Iterative Optimal Control for the Continuous Time Finite Horizon Linear Quadratic Regulator Problem with Unknown Dynamics

Justin Fong*, Ying Tan, Vincent Crocher, Denny Oetomo, Iven Mareels

Melbourne School of Engineering, The University of Melbourne, Parkville, Victoria, Australia 3031

Abstract

Achieving optimal performance over a finite-time horizon has gained a lot of attention in many engineering applications. Among them the Finite Horizon Linear Quadratic Regulator (FHLQR) for continuous-time linear-time-varying systems has been well studied with an optimal solution characterized by the Differential Riccatti Equation (DRE). The solution of the DRE requires that the exact system dynamics are known. However, this assumption may not always hold, as the plant model might not completely known or may change over time due to wear and tear. This paper proposes a dual-loop iterative algorithm to find the optimal solutions of the FHLQR for continuous time LTV systems. The inner loop utilises input trajectories based on an estimate of the optimal control gain with the addition of some excitation noise, and produces measured state trajectories. The outer loop improves the estimate of the optimal control gain utilising these measured state trajectories. It has been shown in this work that with appropriate selection of the discretisation parameter T and the set of excitation signals, the proposed dual-loop iterative algorithm can converge to an arbitrarily small neighbourhood of the optimal solution. A simulation example demonstrates the effectiveness of the proposed method.

Keywords: Finite Horizon Linear Quadratic Regulator, Continuous Time, Dual-loop Iterative Algorithm

1. Introduction

The optimal control problem, in which the objective is to develop a control strategy which minimises a given cost function, is commonly-considered in the

*Corresponding author

Email addresses: justinf@student.unimelb.edu.au (Justin Fong), yingt@unimelb.edu.au (Ying Tan), v.crocher@unimelb.edu.au (Vincent Crocher), doetomo@unimelb.edu.au (Denny Oetomo), i.mareels@unimelb.edu.au (Iven Mareels)

control field. The foundation of many solutions to this problem is Bellman's Optimality Principle [1], and the solution of the Hamilton-Jacobi-Bellman (HJB) Equation. Solving the HJB Equation, however, generally requires precise knowledge of the dynamics, and often does not have a closed form solution. Although these algorithms have their place, in real world systems exact knowledge of the dynamics is often not available. Furthermore, the dynamics can also undergo changes over time, and these changes are more pronounced in some systems than in others.

Within this paper, the well-studied Continuous Time Finite Horizon (FH) Linear Quadratic Regulator (LQR) problem is considered. Many engineered systems are often posed as this problem, where the objective is to minimise a cost function quadratic in both error in state and control effort, over a given finite time period. The finite duration of time given in many of the practical specified tasks lends itself to the finite horizon of the controller, which provides an explicit mechanism to trade off the accuracy of task completion with the effort we are willing to spend to achieve it. If the dynamics are known, the optimal control scheme for this problem can be calculated using the Differential Riccatti Equation (DRE). This cannot be calculated if the dynamics are unknown. Furthermore, if an inaccurate model is used, or if the dynamics changes between iterations (either slowly, for example due to wear and tear, or suddenly due to a part failing), the control scheme becomes suboptimal. Such inaccurate or unknown dynamics can also be found in the examples of engineered systems; and is exceptionally pronounced among complex biological systems [2] which motivated this work.

As such, this paper proposes an algorithm which solves for the optimal control gain without requiring the knowledge of the dynamics. The proposed method utilises an iterative process to compute the optimal gain matrix using measured state trajectories, using the results of an iterative solution to the DRE proposed in [3]. By not assuming the knowledge of the system dynamics, the proposed algorithm can therefore re-identify an optimal control strategy should the plant dynamics change. The proposed algorithm is posed as the solution to the Finite Horizon Linear Quadratic Regulator (FHLQR) problem for Linear Time Varying (LTV) systems.

Some solutions to similar problems have been proposed in the literature. In [4], a linear time invariant (LTI) Infinite Horizon Linear Quadratic Regulator (IHLQR) problem was investigated. The present paper takes a similar approach, with the major difference being that [4] considers infinite horizon, time invariant dynamics and no terminal cost. The algorithm is proposed as an Adaptive Dynamic Programming (ADP) technique, which utilise successive estimates of the value function to estimate the optimal control law. [5] and [6] provide good reviews of existing ADP techniques. Although the algorithm proposed in the current paper follows a similar structure, the authors have chosen not to describe it as ADP, as value function is not explicitly used to estimate the optimal control gain.

Other approaches also exist for the FHLQR problem, but with discrete time dynamics — [7] proposes an adaptive algorithm for the Discrete Time FHLQR

problem with constant dynamics while [8] uses an extremum-seeking iterative approach to find an open-loop control sequence for the discrete time FHLQR problem with time varying dynamics.

Similar problems exist in Iterative Learning Control, such as the Linear Quadratic Optimal Learning Control [9] and the norm-optimal iterative learning control [10, 11], where optimal performance is sought over a finite horizon of each iteration. However, these algorithms seek an optimal control trajectory for the given task, as opposed to an optimal control law for a family of tasks characterised by the given cost function. Fundamentally, they require an identical initial conditions for each iteration, and only work when the optimal trajectory is identical for all iterations.

The present paper therefore proposes an algorithm to handle the continuous time case, where no knowledge of the system dynamics is required, other than that they are linear and time varying.

1.1. Notation

For any $\mathbf{x} \in \mathbb{R}^n$, $|\mathbf{x}| = \sqrt{\mathbf{x}^T\mathbf{x}}$. For any $A \in \mathbb{R}^{n\times m}$, $|A|$ is its induced matrix norm. The set consisting of all continuous functions defined over $\mathbb{R}^{n\times m}$ over $[t_0,\ t_f]$ for any $n, m \in \mathbb{N}$ is denoted $\mathcal{C}^{n\times m}[t_0,\ t_f]$. For any $A(\cdot) \in \mathcal{C}^{n\times m}[t_0,\ t_f]$, $\|A\|_s^{n\times m} = \max\limits_{t_0\le t\le t_f]} |A(t)|$.

For a given $\mathbf{x} \in \mathbb{R}^n$, and a given $V = V^T \in \mathbb{R}^{n\times n}$, $\mathbf{x}^T V\mathbf{x}$ can be written as $\bar{\mathbf{x}}^T\bar{\mathbf{v}}$ with $\bar{\mathbf{x}} \in \mathbb{R}^{\frac{n(n+1)}{2}}$ and $\bar{\mathbf{v}} \in \mathbb{R}^{\frac{n(n+1)}{2}}$, where:

$$\begin{aligned}\bar{\mathbf{v}} = [&V_{11}, 2V_{12}, ..., .2V_{1n}, V_{22}, 2V_{23}, ...,\\ &2V_{2n}, ..., V_{n-1,n-1}, 2V_{n-1,n}, V_{nn}]^T \qquad (1)\\ \bar{\mathbf{x}} = [&x_1^2, x_1x_2, ..., x_1x_n, x_2^2, x_2x_3, ...,\\ &x_2x_n, ..., x_{n-1}^2, x_{n-1}x_n, x_n^2]^T \qquad (2)\end{aligned}$$

where V_{ij} represents the element of $V \in \mathbb{R}^{n\times n}$ in the i^{th} row and j^{th} column and x_i represents the i^{th} element in vector $\mathbf{x}$.

Furthermore, $\mathbf{y}^T K\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^n$, and $K \in \mathbb{R}^{n\times m}$ can be written as $(\mathbf{x}\otimes\mathbf{y})^{\mathbf{k}}$ with $\mathbf{x}\otimes\mathbf{y} \in \mathbb{R}^{nm}$ where:

$$\begin{aligned}\mathbf{x}\otimes\mathbf{y} = [&x_1y_1, x_1y_2, ..., x_1y_m, x_2y_1, ...,\\ &x_2y_m, ..., x_ny_1, ..., x_ny_{m-1}, x_ny_m]^T \qquad (3)\\ \mathbf{k} = [&K_{11}, K_{21}, ..., K_{n1}, K_{12}, K_{22}, ...,\\ &K_{n2}, ..., K_{1m}, ..., K_{nm}]^T \qquad (4)\end{aligned}$$

2. Problem Formulation

The systems of interest take the following form:

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \qquad (5)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$ and dynamics matrices $A(\cdot) \in \mathcal{C}^{n\times n}[t_0,\ t_f]$ and $B(\cdot) \in \mathcal{C}^{n\times m}[t_0, t_f]$.

The objective of the Finite Horizon (FH) Linear Quadratic Regulation (LQR) problem is to minimise the following cost function, subject to the dynamic system (5):

$$J(\mathbf{u}(\cdot)) = \mathbf{x}^T(t_f)\Phi_f\mathbf{x}(t_f) + \int_{t_0}^{t_f} \left(\mathbf{x}^T(t)Q(t)\mathbf{x}(t) + \mathbf{u}^T(t)R(t)\mathbf{u}(t)\right) dt \tag{6}$$

where $\mathbf{x}$ is the resulting trajectory from the dynamics (5), $\Phi_f \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite ($\Phi_f = \Phi_f^T \geq 0$), $Q(\cdot) \in \mathcal{C}^{n\times n}[t_0, t_f]$ is also symmetric positive semidefinite, and $R(\cdot) \in \mathcal{C}^{m\times m}[t_0, t_f]$ is symmetric positive definite ($R(t) = R(t)^T > 0$).

For the cost function (6) subject to (5) the optimal control law is a time-varying feedback control scheme of the following form:

$$\mathbf{u}(t) = -K^*(t)\mathbf{x}(t), \qquad \forall t \in [t_0, t_f], \tag{7}$$

where $K^*(t) \in \mathcal{C}^{m\times n}[t_0, t_f]$ satisfies

$$K^*(t) = R^{-1}B^T(t)P(t), \tag{8}$$

here $P(t) \in \mathcal{C}^{n\times n}[t_0, t_f]$ is the solution of the following Differential Riccati Equation (DRE) [12]:

$$\dot{P}(t) = -A(t)^T P(t) - P(t)A(t) - Q(t) + P(t)^T B(t)R^{-1}(t)B(t)^T P(t) \tag{9}$$

subject to $P(t_f) = \Phi_f$.

It is noted that this is a standard optimal controller of full state feedback form, and thus for the purposes of this work it is assumed that all states are known and measurable. This is the case for many engineered systems.

3. Preliminaries

Kleinman [3, Theorem 8, page 53] proposed a method of iteratively solving the DRE offline, using a dynamic programming approach to iteratively solve the DRE. This section introduces the algorithm and two properties of this algorithm are identified to be later used in the analysis.

3.1. Kleinman's Iterative Solution to the FH LQR Problem

Analysis and description of the algorithm requires the definition of the Cost-to-go Matrix, $V_k(t)$, for a given control gain $K_k(t)$ as the solution to:

$$\dot{V}_k(t) = -\left[A(t) - B(t)K_k(t)\right]^T V_k(t) - V_k(t)[A(t) - B(t)K_k(t)] - Q(t) - K_k^T(t)R(t)K_k(t) \tag{10}$$

with final condition $V_k(t_f) = \Phi_f$. Utilising this definition, the following was proposed in [3].

Proposition 1 (Kleinman's Algorithm [3]). *For the FH LQR problem, with system dynamics (5) and cost function (6), if the following algorithm is followed:*

1. *Define arbitrary $K_0(t)$, set $k = 0$*
2. *Solve for $V_k(t)$ using Equation (10)*
3. *Update $K_{k+1}(t)$ through:*

$$K_{k+1}(t) = R^{-1}(t)B^T(t)V_k(t) \tag{11}$$

4. *Set $k = k + 1$ and go to 2.*

Then:

1. *$P(t) \leq V_{k+1}(t) \leq V_k(t)$, for all $k \in \mathbb{N}_{\geq 0}$*
2. *$V_k(t)$ converges to $P(t)$ uniformly*[1]
3. *$K_k(t)$ converges to $K^*(t)$ uniformly.*

where $K^*(t)$ is as defined in (8). The algorithm will monotonically converge to the value function associated with the optimal control gain. ∘

The proof of Proposition 1 is given in [3]. This is an iterative process, in which successive estimates of the optimal control gain are made. The algorithm provides a less computationally expensive method of solving the DRE, as the algorithm involves iteratively solving a linear matrix equation (10), rather than the quadratic DRE (9). However, the complete knowledge of the dynamics of the system ($A(t)$ and $B(t)$) is required to solve (10). In the sequel, we explore some properties of Kleinman's algorithm.

3.2. A Converse Theorem for Kleinman's Algorithm

The first property generalises the Converse Theorem proposed in [13] to characterize the convergence properties of $V_k(t)$ in the algorithm.

It is noted that $V_k(t)$ is directly related to $K_k(t)$ through (10). Therefore, for any $K_k(t)$, we can write $V_k(t)$ as $V(t, K_k(t), A(t), B(t), Q(t), R(t))$. A new variable $\Sigma_k(t) = [V(t, K_k(t)) - P(t)] \in \mathcal{C}^{n\times n}[t_0,\ t_f]$ is now introduced. Using the results listed in Proposition 1, the update laws (10) and (11) can be rewritten as:

$$\Sigma_{k+1}(t) = f(\Sigma_k(t), A(t), B(t), Q(t), R(t)) \tag{12}$$

with $\Sigma_k(t_f) = 0_{n\times n}$ for any $k \in \mathbb{N}_{\geq 0}$. Here the mapping f satisfies $\mathcal{C}^{n\times n}[t_0,\ t_f]\times \mathcal{C}^{n\times n}[t_0,\ t_f] \times \mathcal{C}^{n\times m}[t_0,\ t_f] \times \mathcal{C}^{n\times n}[t_0,\ t_f] \times \mathcal{C}^{m\times m}[t_0,\ t_f] \to \mathcal{C}^{n\times n}[t_0,\ t_f]$.

Proposition 1 indicates that the system (12) is uniformly globally asymptomatically stable (UGAS) with respect to the set $\mathcal{C}^{n\times n}[t_0,\ t_f]$ (see [13, Definition 2.1] for the definition of UGAS for discrete-time nonlinear systems — the same definition can be extended to system (12)). By using Converse Theorem [13, Theorem 1 & Lemma 2.8], and noting importantly that the convergence has no dependence on iteration number k, the following property holds:

[1] The sequence $V_k(t)$ converges to $P(t)$ uniformly indicates that $\lim_{k\to\infty} \|V_k - P\|_s^{n\times n} = 0$.

Property 1. *Let V_k and K_k be defined as per Kleinman's algorithm, leading to the dynamic system (12) and α_l, $l = 1, 2, 3$ be class-$\mathcal{K}_\infty$ functions* [2]*. Then there exists a continuous function $W : \mathbb{R}^{n\times n} \to \mathbb{R}_{\geq 0}$ such that for any $\Sigma \in \mathcal{C}^{n\times n}[t_0,\ t_f]$:*

1 $W(\Sigma)$ is bounded by:

$$\alpha_1\left(|\Sigma|\right) \leq W(\Sigma) \leq \alpha_2\left(|\Sigma|\right), \tag{13}$$

2 The update law (12) satisfies:

$$W\left(f\left(\Sigma, A, B, Q, R\right)\right) - W\left(\Sigma\right) \leq -\alpha_3\left(\left(|\Sigma|\right)\right) \tag{14}$$

Remark 1. *[13] defines a function $W : \mathbb{N}_{\geq 0} \times \mathcal{C}^{n\times n}[t_0,\ t_f] \to \mathbb{R}_{\geq 0}$, however, in Kleinman's algorithm, the convergence has no dependence on iteration number k. Therefore, the converse function defined for this problem is independent of k.*

3.3. Property of the Cost-to-go Matrix

A property of the Cost-to-go Matrix $V_k(t)$ is constructed utilising a system input which is perturbed by an excitation signal $\mathbf{w}(t)$, i.e. when considering the dynamics using the k^{th} estimate of $K^*(t)$:

$$\mathbf{u}(t) = -K_k(t)\mathbf{x}(t) + \mathbf{w}(t), \tag{15}$$

where $\mathbf{w} \in \mathbb{R}^m$.

Property 2. *Let $t_0 \leq t_a < t_b \leq t_f$ and $\mathbf{x}(t)$ be a trajectory obtained from applying (15) to (5). Then the following equality holds*

$$\begin{aligned}
&\mathbf{x}^T(t_b)V_k(t_b)\mathbf{x}(t_b) - \mathbf{x}^T(t_a)V_k(t_a)\mathbf{x}(t_a) \\
&= \int_{t_a}^{t_b} \big[2\mathbf{w}^T(t)R(t)K_{k+1}(t)\mathbf{x}(t) \\
&\qquad -\mathbf{x}^T(t)(Q(t) + K_k^T(t)R(t)K_k(t))\mathbf{x}(t)\big]\, dt
\end{aligned} \tag{16}$$

Proof: Using the dynamics of the system with control law (15) and the derivative of the Cost-to-go matrix (10), an expression for $\frac{d}{dt}\left(\mathbf{x}^T(t)V_k(t)\mathbf{x}(t)\right)$ can be found. Integrating over $t = t_a$ to $t = t_b$, and substituting $B^T(t)V_k(t) = R(t)K_{k+1}(t)$ produces the above property. ■

4. Proposed Algorithm

This section presents an algorithm which iteratively solves the Continuous Time FHLQR problem without requiring the explicit knowledge of the dynamics of the system. A high level overview of the proposed algorithm is first presented, followed by more a detailed analysis of each iteration of the outer loop, and finally the convergence of the algorithm to the optimal control gain is shown.

[2] A function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a $\mathcal{K}_\infty$ function if it is continuous, strictly increasing, $\gamma(0) = 0$ and $\gamma(s) \to \infty$ as $s \to \infty$.

4.1. High Level Overview

The overall structure of the proposed algorithm is shown in Figure 1. The nested structure contains two loops (both in the iteration domain), and allows two objectives to be fulfilled. The objective of the *Outer Loop* (index k) is to iterate over estimates of $K^*(t)$, to eventually converge to a region around $K^*(t)$, along the solutions of Klienman's Algorithm. Within the *Inner Loop* (index j), the computed feedback gain ($K_k(t)$) is applied along with sufficient excitation (in the j domain) to generate an appropriate number of online measurements of state trajectories, which are then used to estimate $K_{k+1}(t)$ and $V_k(t)$ at given sampling points.

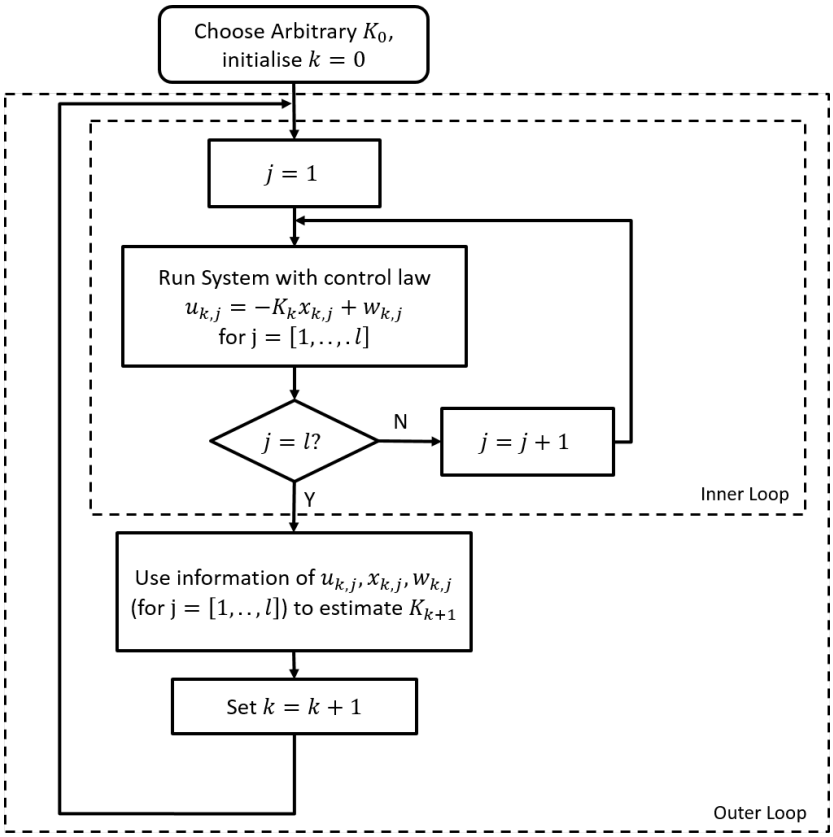

Figure 1: Proposed Algorithm Structure

4.1.1. Notation for the Algorithm

A new set of notation is defined in Table 1 for the remainder of the paper, based on the notation used in Section 3 with modifications to cater for the features in the proposed algorithm.

Table 1: Notation and Symbols for Different Control Gains

$K(t)$	Description	$V(t)$	$\Sigma(t)$
$K^*(t)$	The optimal control gain	$P(t)$	0
$K_k(t)$	The control gain applied at the k^{th} iteration	$V_k(t)$	$\Sigma_k(t)$
$K^*_{k+1}(t)$	The next estimate of the Optimal Control Gain as defined by Klienman's Algorithm, where the previous point is $K_k(t)$	$V^*_{k+1}(t)$	$\Sigma^*_{k+1}(t)$
$\tilde{K}^*_{k+1}(t)$	A piecewise approximation of $K^*_{k+1}(t)$, defined: $\tilde{K}^*_{k+1}(t) = K^*_{k+1}(t_i)$ $\forall t \in [t_i, t_{i+1})$, $i = 0, ..., N-1$	unused	$\tilde{\Sigma}^*_{k+1}(t)$
$\hat{K}^*_{k+1}(t)$	A least squares estimate of $\tilde{K}^*_{k+1}(t)$	unused	$\hat{\Sigma}^*_{k+1}(t)$

It is important to note that $K_k(t)$ is the gain applied at the k^{th} iteration of the outer loop. This has the associated cost-to-go matrix $V_k(t)$. This is *not* the same as the gain as calculated by Klienman's algorithm. For this, $K^*_{k+1}(t)$ is used. That is, given $K_k(t)$ with associated cost-to-go matrix, $V_k(t)$, $K^*_{k+1}(t) = R^{-1}(t)B^T(t)V_{k-1}(t)$. The reader is also reminded that each of these matrices can be represented in a vector form as discussed in Section 1.1.

4.2. Outer Loop

Given any arbitrary $K_k(t) \in \mathcal{C}^{m\times n}[t_0,\ t_f]$, $K^*_{k+1}(t)$ can be computed based on the knowledge of $A(t), B(t)$ using Kleinman's Algorithm. Without the knowledge of $A(t), B(t)$, Property 2 and Least Square Estimation can be used to estimate $K^*_{k+1}(t)$ and $V_k(t)$. As continuous functions are difficult to identify explicitly, these two functions are discretised over $[t_0, t_f]$ so that parameter identification methods can be applied. A sampling period T is therefore selected such that there exist $N = \frac{t_f - t_0}{T} + 1$ sampling instants: $t_i = t_0 + iT, i = 0, \ldots, N$. The proposed algorithm attempts to find a piecewise constant approximation of $K^*_{k+1}(t)$ at time instants $t_j, j \in [0, ..., N-1]$. This discretisation, of course, introduces an error, however, it will be shown that this error can be bounded.

4.2.1. Least Square Estimation of $K_{k+1}(t)$ and $V_k(t)$ at Each Sampling Instant

With the given $K_k(t)$, the system has the following controller and the corresponding closed-loop dynamics:

$$\begin{aligned}\mathbf{u}_k(t) &= -K_k(t)\mathbf{x}_k(t) + \mathbf{w}_k(t)\\ \mathbf{x}_k(t) &= (A(t) - B(t)K_k(t))\mathbf{x}_k(t) + B(t)\mathbf{w}_k(t), \mathbf{x}_k(0) \in \mathbb{R}^n.\end{aligned} \tag{17}$$

By applying Property 2 in any interval $[t_i, t_{i+1}]$, $i = 0, \ldots, N-1$, it follows that

$$\begin{aligned}&\mathbf{x}_k^T(t_{i+1})V_k(t_{i+1})\mathbf{x}_k(t_{i+1}) - \mathbf{x}_k^T(t_i)V_k(t_i)\mathbf{x}_k(t_i)\\ &\quad= \int_{t_i}^{t_{i+1}} \left[2\mathbf{w}_k^T(t)R(t)K_{k+1}^*(t)\mathbf{x}_k(t)\right] dt\\ &\qquad- \int_{t_i}^{t_{i+1}} \left[\mathbf{x}_k^T(t)\left(Q(t) + K_k^T(t)R(t)K_k\right)\mathbf{x}_k(t)\right] dt.\end{aligned} \tag{18}$$

With the introduction of $\tilde{K}_{k+1}^*(t) = K_{k+1}^*(t_i), \forall t \in [t_i, t_{i+1}),\ i = 0, ..., N-1$, using the notation from Section 1.1, and introducing:

$$\boldsymbol{\delta}_i^T(\mathbf{x}_k, \mathbf{w}_k) := 2\int_{t_i}^{t_{i+1}} (\mathbf{x}_k(t) \otimes R^T(t)\mathbf{w}_k(t))^T dt \tag{19}$$

$$\begin{aligned}&\gamma_i(\mathbf{x}_k, K_k)\\ &\quad:= -\int_{t_i}^{t_{i+1}} \left[\mathbf{x}_k^T(t)(Q(t) + K_k^T(t)R(t)K_k(t))\mathbf{x}_k(t)\right] dt\end{aligned} \tag{20}$$

$$\begin{aligned}&\rho_i(\mathbf{x}_k, \mathbf{w}_k)\\ &\quad:= 2\int_{t_i}^{t_{i+1}} (\mathbf{x}_0(t) \otimes R^T(t)\mathbf{w}_k(t))^T(\mathbf{k}_{k+1}^*(t) - \tilde{\mathbf{k}}_{k+1}^*(t_i))dt\end{aligned} \tag{21}$$

(18) can be written as:

$$\begin{aligned}&\left(\bar{\mathbf{x}}_k\right)^T(t_{i+1})\overline{\mathbf{v}}_k(t_{i+1}) - \left(\bar{\mathbf{x}}_k\right)^T(t_i)\overline{\mathbf{v}}_k(t_i)\\ &\quad= \boldsymbol{\delta}_i^T(\mathbf{x}_k, \mathbf{w}_k) \cdot \tilde{\mathbf{k}}_{k+1}^*(t_i) + \gamma_i(\mathbf{x}_k, K_k) + \rho_i(\mathbf{x}_k, \mathbf{w}_k)\end{aligned} \tag{22}$$

with $V_k(t_N) = \Phi_f$, and where $\tilde{\mathbf{k}}_{k+1}^*(t_i)$ is the vector form of $\tilde{K}_{k+1}^*(t)$ at $t \in [t_i, t_{i+1}]$, and $\overline{\mathbf{v}}_k(t_{i+1})$ is the vector form of $V_k(t_i)$.

Assuming knowledge of $\bar{\mathbf{v}}_k(t_{i+1})$, this equation can be used as the basis to estimate $\bar{\hat{\mathbf{v}}}_k(t_i)$ and $\bar{\hat{\mathbf{k}}}_1^*(t_i)$. However, (22) is a scalar equation, whereas $\bar{\mathbf{v}}_k(t_i) \in R^{\frac{n(n+1)}{2}}$ and $\tilde{\mathbf{k}}_{k+1}^*(t_i) \in R^{nm}$ — they are not scalar. Therefore, to estimate $\bar{\hat{\mathbf{v}}}_k(t_i)$ and $\tilde{\mathbf{k}}_{k+1}^*(t_i)$, more information is required. This is addressed using a Least Squares Estimation (LSE). In order to use a LSE, a sequence of dither signals $\mathbf{w}_{k,j}(t), j = 1, 2, \ldots, \ell$ are required to generate sufficient excitation along the inner loop (j) iteration domain. For a sufficiently large ℓ, and with sufficient variation in the excitation signals, it is possible to estimate $\bar{\mathbf{v}}_k(t_i)$ and $\tilde{\mathbf{k}}_{k+1}^*(t_i)$.

More precisely, at each inner loop iteration $j = 1, \ldots, \ell$, dynamics are of the form:

$$\begin{aligned}\mathbf{u}_{k,j}(t) &= -K_k(t)\mathbf{x}_{k,j}(t) + \mathbf{w}_{k,j}(t)\\ \mathbf{x}_{k,j}(t) &= (A(t) - B(t)K_k(t))\mathbf{x}_{k,j}(t) + B(t)\mathbf{w}_{k,j}(t)\end{aligned} \tag{23}$$

where $\mathbf{x}_k(0) \in \mathbb{R}^n$. Utilising this information, at time t_i, ℓ equations in the form of (22) can be obtained. By using the following notation

$$\begin{aligned} X_{k,i} &= \begin{bmatrix} \bar{\mathbf{x}}_{k,1}^T(t_i) \\ \bar{\mathbf{x}}_{k,2}^T(t_i) \\ \vdots \\ \bar{\mathbf{x}}_{k,\ell}^T(t_i) \end{bmatrix} \quad \Delta_{k,i} = \begin{bmatrix} \boldsymbol{\delta}_i^T(\mathbf{x}_{k,1}, \mathbf{w}_{k,1}) \\ \boldsymbol{\delta}_i^T(\mathbf{x}_{k,2}, \mathbf{w}_{k,2}) \\ \vdots \\ \boldsymbol{\delta}_{k,i}^T(\mathbf{x}_{k,\ell}, \mathbf{w}_{k,\ell}) \end{bmatrix} \\ \mathbf{c}_{k,i} &= \begin{bmatrix} \gamma_i(\mathbf{x}_{k,1}, K_k) \\ \gamma_i(\mathbf{x}_{k,2}, K_k) \\ \vdots \\ \gamma_i(\mathbf{x}_{k,\ell}, K_k) \end{bmatrix} \quad \boldsymbol{\epsilon}_{0,i} = \begin{bmatrix} \rho_i(\mathbf{x}_{k,1}, \mathbf{w}_{k,1}) \\ \rho_i(\mathbf{x}_{k,2}, \mathbf{w}_{k,2}) \\ \vdots \\ \rho_i(\mathbf{x}_{k,\ell}, \mathbf{w}_{k,\ell}) \end{bmatrix} \end{aligned} \tag{24}$$

this leads to the following matrix equation:

$$\begin{bmatrix} X_{k,i} & \Delta_{k,i} & -X_{k,i+1} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{v}}_k(t_i) \\ \tilde{\mathbf{k}}_{k+1}^*(t_i) \\ \bar{\mathbf{v}}_k(t_{i+1}) \end{bmatrix} = -\mathbf{c}_{k,i} - \boldsymbol{\epsilon}_{k,i} \tag{25}$$

At $i = N-1$, $\bar{\mathbf{v}}_k(t_{i+1}) = \bar{\mathbf{v}}_k(t_N)$. Furthermore, at all iterations, the value $\boldsymbol{\epsilon}_{k,j}$ is unknown, however, can be made small with small sampling period T.

Consider now the least squares problem, in which all time steps are augmented into a single estimation. With the following definitions:

$$\begin{aligned} \Phi_k &= \begin{bmatrix} X_{k,0} & \Delta_{k,0} & -X_{k,1} & 0 & \ldots & 0 & 0 \\ 0 & 0 & X_{k,1} & \Delta_{k,1} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & X_{k,N-1} & \Delta_{k,N-1} \end{bmatrix} \\ \mathbf{c}_k &= \left[-\mathbf{c}_{k,0}, -\mathbf{c}_{k,1}, \ldots, -\mathbf{c}_{k,N} + X_{k,N}\bar{\tilde{\mathbf{v}}}_k(t_N)\right]^T, \\ \boldsymbol{\epsilon}_k &= [\boldsymbol{\epsilon}_{k,0}, \boldsymbol{\epsilon}_{k,1}, \ldots, \boldsymbol{\epsilon}_{k,N}]^T \\ \boldsymbol{\xi}_k &= [\bar{\hat{\mathbf{v}}}_k(t_0), \hat{\mathbf{k}}_{k+1}^*(t_0), \bar{\hat{\mathbf{v}}}_k(t_1), \hat{\mathbf{k}}_{k+1}^*(t_1), \\ &\qquad \ldots, \bar{\hat{\mathbf{v}}}_k(t_{N-1}), \hat{\mathbf{k}}_{k+1}^*(t_{N-1})]^T \end{aligned} \tag{26}$$

where $\Phi_k \in \mathbb{R}^{N\ell \times N\left(\frac{n(n+1)}{2}+nm\right)}$, $\mathbf{c}_k, \boldsymbol{\epsilon}_k \in \mathbb{R}^{N\ell}$ and $\boldsymbol{\xi}_k \in \mathbb{R}^{N\left(\frac{n(n+1)}{2}+nm\right)}$, and $\hat{\mathbf{k}}_{k+1}^*(t_i)$ is an estimate of $\tilde{\mathbf{k}}_{k+1}^*(t_i)$, the following least squares problem can be posed:

$$\Phi_k \boldsymbol{\xi}_k = \mathbf{c}_k \tag{27}$$

It is noted that $\boldsymbol{\xi}_k$ contains estimates for $\bar{\mathbf{v}}_k(t)$ and $\tilde{\mathbf{k}}^*_{k+1}(t)$ (rather than $\mathbf{k}^*_{k+1}(t)$) at each sampling instant t_i, and thus $\boldsymbol{\epsilon}_k$ can be considered a disturbance to the solution, and the error can be bounded by:

$$\delta\boldsymbol{\xi}_k = \kappa(\Phi_k)\,\|\boldsymbol{\epsilon}_k\| \tag{28}$$

where $\kappa(\Phi_k)$ is the condition number of the matrix Φ_k (see [14] for complete definition).

Utilising the result of this least squares problem, $\hat{\mathbf{k}}^*_{k+1}(t_i)$ for $i = 0, ..., N-1$ can be used as a feedback control gain $K_{k+1}(t)$ in the piecewise constant form:

$$K_{k+1}(t) \equiv \hat{K}^*_{k+1}(t_i), \forall t \in [t_i, t_{i+1}), i = [0, .., N-1] \tag{29}$$

The rest of this section will show that $K_{k+1}(t) \equiv \hat{K}^*_{k+1}(t)$ can be made arbitrarily close to $K^*_{k+1}(t)$. This can be shown by demonstrating two facts:

1. The discretisation error $\left|\mathbf{k}^*_{k+1}(t_i) - \tilde{\mathbf{k}}^*_{k+1}(t_i)\right|$ can be made arbitrarily small.
2. The estimation error $\left|\hat{\mathbf{k}}^*_{k+1}(t_i) - \tilde{\mathbf{k}}^*_{k+1}(t_i)\right|$ can be made arbitrarily small

The following fact comes from the uniform continuity of the $\mathbf{k}^*_{k+1}(t)$ over a compact time interval $[0, T]$.

Fact 1. *Let (ν_1, Δ_1) be an arbitrary positive pair. There exists a sufficiently small T_1^*, such that for any continuous function $\mathbf{k}^*_{k+1}(t)$ satisfying $\left\|\mathbf{k}^*_{k+1}\right\|_s \leq \Delta_1$ and any $T \leq T_1^*$ and $N_1 = \frac{t_f - t_0}{T}$ is an integer such that at each sampling instant $t_j, j = 1, \ldots, N_1$, the following inequality holds*

$$\begin{aligned} \max\left\{\left|\mathbf{k}^*_{k+1}(t) - \tilde{\mathbf{k}}^*_{k+1}(t_i)\right|\right\} &\leq \nu_1 \\ \forall t \in [t_i, t_{i+1}], \forall i = 0, ..., N-1 & \end{aligned} \tag{30}$$

Before the introduction of the second fact conditioning techniques [14] for the matrix are now introduced. For the purposes of this algorithm, it is obvious that the estimate of $\tilde{\mathbf{k}}^*_{k+1}(t_i)$ is of greater interest, whereas $\bar{\mathbf{v}}_k(t_i)$ has less importance. As such, to minimise the impact of the error in the estimate of $\tilde{\mathbf{k}}^*_{k+1}(t_i)$ the least squares problem (27) is conditioned as:

$$\Phi_k D_k^{-1} D_k \boldsymbol{\xi}_k = \mathbf{c}_k, \tag{31}$$

where $D_k \in \mathbb{R}^{N\left(\frac{n(n+1)}{2}+nm\right)\times N\left(\frac{n(n+1)}{2}+nm\right)}$ is a diagonal conditioning matrix. This matrix serves as a scaling factor in estimating $\hat{\mathbf{v}}_k(t_i)$ and $\hat{\mathbf{k}}_{k+1}(t_i)$ terms. In particular, D_k is chosen such that the elements corresponding to $\hat{\mathbf{k}}_{k+1}(t_i)$ are 1, i.e. it is of the form:

$$D_k = diag\left(\mathbf{s}_0, 1_{nm\times 1}, \mathbf{s}_1, 1_{nm\times 1}, ..., \mathbf{s}_{N-1}, 1_{nm\times 1}\right), \tag{32}$$

where $\mathbf{s}_i \in \mathbb{R}^{\frac{n(n+1)}{2}}$ for $i = [0, .., .N-1]$, and $1_{nm\times 1}$ is a R^{nm} vector of ones.

The least squares can then be solved in a two-step process:

$$D_k \boldsymbol{\xi}_k = \left(\Phi_k D_k^{-1}\right)^\dagger \mathbf{c}_k \tag{33}$$

$$\boldsymbol{\xi}_k = D_k^{-1} \left(\Phi_k D_k^{-1}\right)^\dagger \mathbf{c}_k \tag{34}$$

where $\left(\Phi_k D_k^{-1}\right)^\dagger$ is the pseudoinverse of $\left(\Phi_k D_k^{-1}\right)$.

Based on this, the error in the estimate of $(D_k \boldsymbol{\xi}_k)$ is bounded by:

$$\delta(D_k \boldsymbol{\xi}_k) \leq \kappa(\Phi_k D_k^{-1}) \, |\boldsymbol{\epsilon}_k| \tag{35}$$

As D_k is a diagonal matrix, D_k^{-1} is simply:

$$D_k^{-1} = diag\left(\mathbf{s}_0^{-1}, 1_{nm\times 1}, \mathbf{s}_1^{-1}, 1_{nm\times 1}, ..., \mathbf{s}_{N-1}^{-1}, 1_{nm\times 1}\right) \tag{36}$$

where $\mathbf{s}_i^{-1} \in \mathbb{R}^{\frac{n(n+1)}{2}}$ is a vector in which each element is the reciprocal of the corresponding element in $\mathbf{s}_i$. Therefore, the error in the estimate of the elements in $\hat{\mathbf{k}}_{k+1}^*(t_i)$ is bounded by:

$$\left|\hat{\mathbf{k}}_{k+1}^*(t_i) - \tilde{\mathbf{k}}_{k+1}^*(t_i)\right| \leq \kappa(\Phi_k D_k^{-1}) \, |\boldsymbol{\epsilon}_k| \tag{37}$$

The excitation signals need to be well-selected in order to ensure that the estimation error is still bounded. The next assumption assumes that by using enough excitation signals $\mathbf{w}_{k,j}$, $j = 1, \ldots, \ell$, for any iteration k, the solution of the LSE is always bounded. It is worthwhile to highlight that the tuning parameters in the design are $\{\ell, T\}$.

The following assumption is thus needed:

Assumption 1. *For any given discretisation step size $T = t_{i+1} - t_i, i = [0, .., N-1]$, $\lambda > 1$, $k \in N_{\geq 0}$, and bound on magnitude of excitation signal $b_w > 0$, there exists a positive integer ℓ^* such that for any $\ell \geq \ell^*$, there are some appropriate set of discretisation signals $\mathbf{w}_{k,j}, j = 1, ..., \ell$, with $\|\mathbf{w}_{k,j}\| \leq b_w$ and some scaling matrix in the form of (32) such that $\kappa(\Phi_k D_k^{-1}) \leq \lambda$.*

Remark 2. *This assumption is similar to a persistent excitation condition needed for LSE. Improving the condition number can be done both by increasing variance in the $\mathbf{w}_{k,j}$ signals and using the D_k matrix. The largest differences in relative magnitudes in Φ_k are due to the difference between $\boldsymbol{\zeta}_{k,i}$ components (proportional to the magnitude of the state $\mathbf{x}_{k,i}$ only), and the components of $\boldsymbol{\mu}_{k,i}$ (proportional to the state, input weighting matrix $R(t)$, excitation signals $\mathbf{w}_{k,j}(t)$ and, importantly, discretisation step size T). The D_k matrix can be used to bring these elements to a similar orders of magnitude once Φ_k has been computed. The variance in the set of $\mathbf{w}_{k,j}(t)$ signals can then be used to ensure suitable inter-row independence. It is also noted that ℓ^* is also highly dependent on the dimension of the system. With a larger number of states and control inputs, the number of parameters to be identified increases, and thus more iterations are required to achieve sufficient excitation.*

Remark 3. *It is worthwhile to highlight that the tuning parameters are selected sequentially. The sampling interval T and bound on the magnitude of the excitation signal b_w can be first selected, creating a bound on the discretisation error. Then a family of the dither signals satisfying the bound (including the number of inner loop iterations ℓ) can be selected.*

Fact 2. *Given Assumption 1 and $(\nu_2, b_w, \lambda, \ell^*)$, there exists a sufficiently small T_2^*, such that for any $T < T_2^*$ and $N_2 = \frac{t_f - t_0}{T}$ is an integer, such that at each sampling instant $t_j, j = 1, \ldots, N_2$, the following inequality holds*

$$\max\left\{\left|\hat{\mathbf{k}}_{k+1}^*(t_i) - \tilde{\mathbf{k}}_{k+1}^*(t_i)\right|\right\} \leq \nu_2$$
$$\forall t \in [t_i, t_{i+1}], \quad \forall i = 0, ..., N-1 \tag{38}$$

Proof: Noting that $\boldsymbol{\epsilon}_k$ is expressed as

$$\boldsymbol{\epsilon}_{k,i} = \begin{bmatrix} 2\int_{t_i}^{t_{i+1}} q_{k,1}(\mathbf{k}_{k+1}^*(t) - \tilde{\mathbf{k}}_1^*(t_i))dt \\ 2\int_{t_i}^{t_{i+1}} q_{k,2}(\mathbf{k}_{k+1}^*(t) - \tilde{\mathbf{k}}_1^*(t_i))dt \\ \vdots \\ 2\int_{t_i}^{t_{i+1}} q_{k,\ell}(\mathbf{k}_{k+1}^*(t) - \tilde{\mathbf{k}}_{k+1}^*(t_i))dt \end{bmatrix} \tag{39}$$

where $q_{k,i} = (\mathbf{x}_{k,i}(t) \otimes R^T(t)\mathbf{w}_{k,i}(t))^T$. Therefore, the bound of $|\boldsymbol{\epsilon}_0|$ is proportional to the size of sampling T. The proof follows by using (37) and Assumption 1. ■

4.2.2. Convergence Towards $K^*(t)$

To prove convergence towards $K^*(t)$, the converse theorem in Property 1 is used. $W(\Sigma)$ is bounded by two K_∞ functions of $|\Sigma|$. Therefore, $W(\Sigma)$ can be used as a measure of the optimality of the solution. That is, a larger value indicates a less optimal solution, and a value of 0 indicates the optimal solution. As such, if it can be shown that the change in $W(\Sigma)$ is negative, $\hat{K}_{k+1}(t)$ is more optimal than $K_k(t)$.

This is formalised in the following theorem, which also utilises the fact that $\hat{K}_{k+1}(t)$ can be made arbitrarily close to $K_{k+1}^*(t)$ with sufficiently small T.

Theorem 1. *Let (Δ, ν) be a positive pair. For any $K_k(t)$ satisfying $|\Sigma_k(t)| \leq \Delta$ for any $t \in [t_0, t_f]$, there exists some T^* such that for any $T < T^*$, there exists a b_w, some positive integer ℓ^* (by Assumption 1), and $\rho < 1$ such that given $\ell \geq \ell^*$, appropriately constructed excitation signals $\mathbf{w}_{k,j}, j \in [1, .., \ell]$ satisfying $\|\mathbf{w}_{k,j}\|_s \leq b_w$ can be constructed such that $K_{k+1}(t) \equiv \hat{K}_{k+1}^*(t)$ computed utilising the least squares estimation (31), satisfies*

$$W(\hat{\Sigma}_{k+1}) \equiv W(\hat{\Sigma}_{k+1}^*) \leq \rho W(\Sigma_k) + \nu, \tag{40}$$

Proof: It is noted that $\Sigma = [V(t, K(t)) - P(t)]$, $W(\Sigma)$ is continuous in Σ, and $V(t)$ is also continuous in $K(t)$. Therefore, $W(\Sigma)$ varies continuously with

$K(t)$. As such, $W(\Sigma(\hat{K}_{k+1}(t))$ can be arbitrarily close to $W(\Sigma(K_{k+1}(t))$ by selecting sufficiently small sampling utilising Fact 1 and Fact 2. From Property 1, there exists some K_∞ function $\alpha_3(\circ)$ such that:

$$W(\Sigma_{k+1}^*) - W(\Sigma_k) \leq -\alpha_3(|\Sigma_k|) \tag{41}$$

Therefore,

$$\begin{aligned} &W(\hat{\Sigma}_{k+1}^*) - W(\Sigma_k) \\ &\quad = W(\Sigma_{k+1}^*) - W(\Sigma_k) - W(\Sigma_{k+1}^*) + W(\hat{\Sigma}_{k+1}^*) \\ &\quad \leq -\alpha_3(|\Sigma_k|) - \left|W(\Sigma_{k+1}^*) - W(\hat{\Sigma}_{k+1}^*)\right| \end{aligned} \tag{42}$$

Two cases are considered:
Case 1: $|\Sigma_k| \leq \alpha_2^{-1}\left(\frac{\nu}{2}\right)$. Here α_2 comes from Property 1. it is possible to select a sufficiently small T^* such that for any $T \in (0, T^*)$

$$\left|W(\hat{\Sigma}_{k+1}^*) - W(\Sigma_{k+1}^*)\right| \leq \frac{\nu}{2}. \tag{43}$$

Consequently, it follows that

$$W(\hat{\Sigma}_{k+1}^*) \leq W(\Sigma_k) + \frac{\nu}{2} \leq \alpha_2\left(\alpha_2^{-1}\left(\frac{\nu}{2}\right)\right) + \frac{\nu}{2} \leq \nu. \tag{44}$$

Case 2: $\alpha_2^{-1}\left(\frac{\nu}{2}\right) \leq |\Sigma_k| \leq \Delta$. Under such a situation, by choosing a sufficiently small T^* such that for any $T \in (0, T^*)$, we have

$$W(\Sigma_{k+1}^*) - W(\hat{\Sigma}_{k+1}^*) \leq \frac{\alpha_3(|\Sigma_k|)}{2} \tag{45}$$

$$\begin{aligned} W(\hat{\Sigma}_{k+1}^*) - W(\Sigma_k) &\leq -\frac{\alpha_3(|\Sigma_k|)}{2} \\ &\leq -\frac{\alpha_3 \circ \alpha_1^{-1}(W(\Sigma_k))}{2} \\ \Rightarrow W(\hat{\Sigma}_{k+1}^*) &\leq W(\Sigma_k) - \frac{\alpha_3 \circ \alpha_1^{-1}(W(\Sigma_k))}{2} \end{aligned} \tag{46}$$

This indicates that there exists $\rho \in (0,1)$ such that $W(\Sigma_k) - \frac{\alpha_3 \circ \alpha_1^{-1}(W(\Sigma_k))}{2} \leq \rho W(\Sigma_k)$ for any $\alpha_2^{-1}\left(\frac{\nu}{2}\right) \leq |\Sigma_k| \leq \Delta$. Consequently, it has

$$W(\hat{\Sigma}_{k+1}^*) \leq \rho W(\Sigma_k). \tag{47}$$

Combining the two cases, we have

$$W(\hat{\Sigma}_{k+1}^*) \leq \max\{\rho W(\Sigma_k), \nu\} \leq \rho W(\Sigma_k) + \nu, \forall\, |\Sigma_k| \leq \Delta \tag{48}$$

This completes the proof. ■

Remark 4. *It is also noted that presence of measurement error can also introduce an error into $\hat{K}_{k+1}$, due to the introduction of errors into the matrices used in the least squares operation. However, errors associated with this error may be mitigated with appropriate filtering of the measurements. Zero-mean noise in the dynamics of the system would not have as significant an effect, as the effects of this noise on the trajectory average out over sampling periods and iterations.*

4.3. Main Result

The main result of this paper is stated in Theorem 2. It shows that for any desired accuracy there exists some discretisation step size T^* such that when the algorithm is applied with $T < T^*$, it will converge on a set of gains close enough to optimal.

Theorem 2. *Let (Δ, ν) be a positive pair. For any $K_0(t)$ satisfying $|\Sigma_0(t)| \leq \Delta$, there exists some $T^* > 0$, such that if $T < T^*$ there exists some b_w such that Assumption 1 holds, and some associated positive integer ℓ^* , $\ell \geq \ell^*$, such that, with appropriately constructed excitation signals $\mathbf{w}_{k,j}, j \in [1, .., \ell]$ satisfying $\|\mathbf{w}_{k,j}\|_s \leq b_w$, there exists a class-$K_\infty$ function α_4 such that the proposed algorithm in Figure 1 converges to a cost close to optimal, i.e:*

$$\limsup_{k\to\infty} |\Sigma_k| \leq \alpha_4(\nu) \tag{49}$$

Proof: The proof comes directly from repeating Theorem 1 as

$$\begin{aligned} W(\hat{\Sigma}_k^*) = W(\Sigma_k) &\leq \rho W(\Sigma_{k-1}) + \nu \\ &= \rho W(\hat{\Sigma}_{k-1}) + \nu \\ &\leq \rho^2 W(\Sigma_{k-2}) + \rho\nu + \nu \\ &\vdots \\ &\leq \rho^k W(\Sigma_0) + \sum_{s=0}^{k} \rho^{k-s}\nu \end{aligned} \tag{50}$$

This completes the proof. ■

The convergence of the algorithm can be illustrated by the diagram in Figure 2. It can be observed at each iteration, Klienman's algorithm proposes some new gain $K_{k+1}^*(t)$ which is more optimal than $K_k(t)$. The errors in approximating $K_{k+1}^*(t)$ can be bounded with the appropriate choice of tuning parameters, such that $\hat{K}_{k+1}(t)$ is close to $K_{k+1}^*(t)$ in optimality. Therefore, the algorithm converges to a region close to optimal as $k \to \infty$.

5. Simulations

In this section, the performance of the algorithm is shown through a number of simulations. In particular, the simulations will illustrate (1) the convergence of the algorithm to near optimal; (2) potential difficulties in achieving excitation when smaller T is chosen; and (3) the convergence of the algorithm to closer to optimal when a smaller T is chosen, however, at increased computational cost.

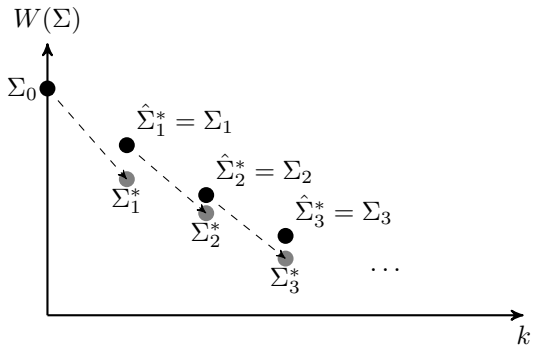

Figure 2: The convergence properties of the algorithm. Σ_k represents the gain $K_k(t)$ applied at each iteration. The dotted lines represent the trajectory of Klienman's Algorithm, the solid '$\mathbf{x}$' represents the trajectory of the algorithm due to the estimation error at each location. The algorithm continues until convergence to the vicinity of $W(\Sigma_k) = 0$. Note $\hat{\Sigma}_k^* = \left[V\left(t, \hat{K}_k^*(t)\right) - P(t)\right]$, and $\Sigma_k^* = \left[V\left(t, K_k^*(t)\right) - P(t)\right]$.

5.1. System Under Investigation

A simple system is presented in order to demonstrate the practicalities involved in implementing this algorithm. However, the algorithm can be applied in the same manner to more complex systems, such as those with time-varying dynamics, or higher order systems. The system considered here is the simple double integrator:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{u}(t) \tag{51}$$

For the purpose of this simulation, the following finite horizon task is used. The cost matrices (which were chosen arbitrarily):

$$\begin{aligned} Q(t) &= \begin{bmatrix} 0.4t + 2 & 0 \\ 0 & 0.4t + 2 \end{bmatrix}; \quad \Phi_f = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \\ R(t) &= \begin{bmatrix} 2.5 - 0.3t & 0 \\ 0 & 2.5 - 0.3t \end{bmatrix} \end{aligned} \tag{52}$$

with $t_0 = 0$ and $t_f = 5$.

5.2. Parameter Selection

In order to simplify the simulations, this paper presents changes in T, and ℓ for the purposes of achieving sufficient conditioning of the Φ_k matrices. The choice of other parameters is outlined in this section. Other choices can be made for any of these parameters.

The set of excitation signals for the simulations, given the bound $w_b = 0.1$, were set as arbitrary values in the range $[-w_b, w_b]$, constant for each $t \in [t_i, t_{i+1})$.

Each element in the scaling matrix was calculated as:

$$s_{k,i} = \begin{bmatrix} \frac{|X_{k,i,0}|}{|\Delta_{k,i}|} & \frac{|X_{k,i,1}|}{|\Delta_{k,i}|} & \dots & \frac{\left|X_{k,i,\frac{n(n+1)}{2}}\right|}{|\Delta_{k,i}|} \end{bmatrix} \tag{53}$$

where $X_{k,i,h}$ is the h^{th} column of $X_{k,i}$. This scales elements in the $X_{k,i}$ matrix to be a comparable order of magnitude to the $\Delta_{k,i}$ matrix.

Finally, the initial value for the gain matrix was selected as $K_0(t) = \begin{bmatrix} 0 & 0 \end{bmatrix}$. This again was an arbitrary choice, but selected for ease of repeatability.

5.3. Results

5.3.1. Convergence of the Algorithm

The first simulation is taken with $T = 0.5s$ and $\ell = 6$. The elements of the cost-to-go matrix $V_k(t)$ are shown in Figure 3. It can be seen that the estimates of $K^*(t)$ in the algorithm iteratively converge to optimal cost $P(t)$ — that is, the cost of the control scheme as k increases approaches optimal.

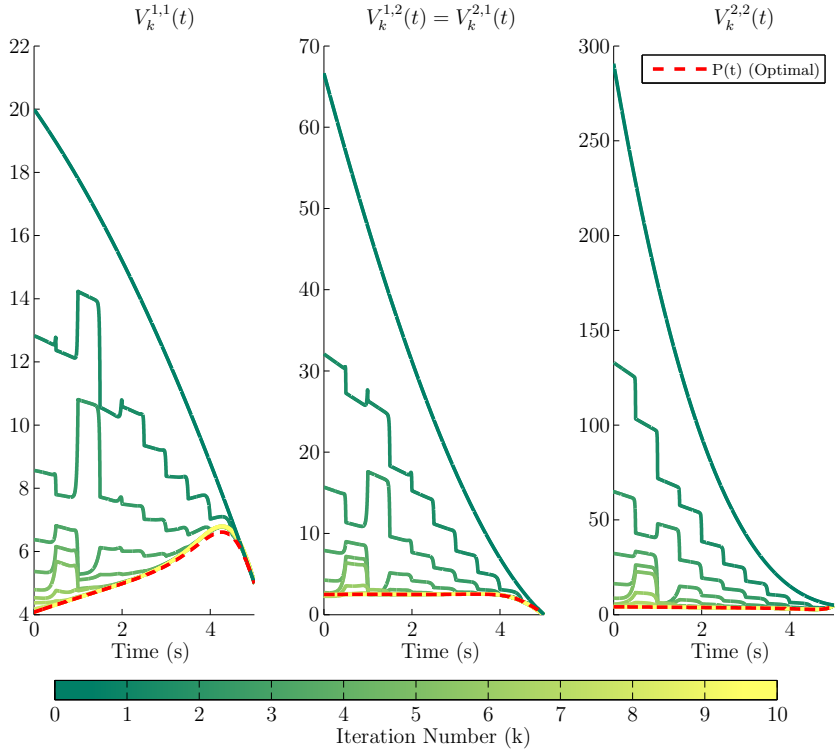

Figure 3: Elements of $V_k(t)$ for Double Integrator System with time step of 0.5 seconds.

5.3.2. Compromise Between Computational Cost and Optimality

In the next set of simulations, different values of T are compared, to demonstrate the trade-off between computational cost and optimality. In this case, the $T = 2.5, T = 0.5$ and $T = 0.05$ cases are compared, setting $\ell = 10$.

These cases are compared on two accounts. First, the optimality of the solution is measured through computation of $\|V_{10} - P\|_s$. Secondly, the average computational time for each outer loop iteration k is computed. The simulations were performed in Matlab 2015a on computer with a Intel i5-4670K (3.4GHz) Processor and 8.00 GB RAM. The results can be seen in Table 2.

Table 2: Performance of Algorithm with Varying T

T	$\|V_{10} - P\|_s$	Computational Time (s)
2.5	0.7	0.00032
0.5	0.17	0.00051
0.05	0.013	0.0029

*Computational time is average time taken for each iteration

It is apparent that reducing T produces a closer estimate of the optimal gain — due to a smaller discretisation error, however at a greater computational cost. Therefore such a tradeoff should be considered in any implementation of this algorithm. It is also noted that this matrix is a sparse matrix, and there are a number of more efficient solvers available (such as [15]), which can improve the computation times of the algorithm.

5.3.3. Achieving Sufficient Excitation

Excitation in this algorithm is required to ensure the condition number of the inverted matrix in the least squares estimate is sufficiently small, and therefore that the algorithm converges. Given any T (set to bound the discretisation error) and b_w (set to bound the disturbance in the least squares estimate), the condition number can be affected by tuning the number of inner loop iterations, ℓ, selection of the scaling matrices D_k, and selection of excitation signals $\mathbf{w}_{k,j}(t)$. As demonstrated in Section 5.3.2, a smaller T results in a more optimal solution as $k \to \infty$. However, due to the smaller T, changes in $\mathbf{x}$ between samples are small, and thus this makes attaining sufficient excitation more difficult. Tuning of the additional parameters can resolve this. As an example, Figure 4 illustrates the convergence of the algorithm with $T = 0.05s$, $\ell = 20$, whereas with $\ell = 6$, the algorithm quickly diverged.

6. Conclusion

This paper presents an algorithm which addresses the Finite Horizon Linear Quadratic Regulator problem, without the need for explicit knowledge of the dynamics of the system. This algorithm utilises a two-loop structure, in which the inner loop (index j) is used to gather information about the system, and the outer loop (index k) is used to make successive approximations of the optimal

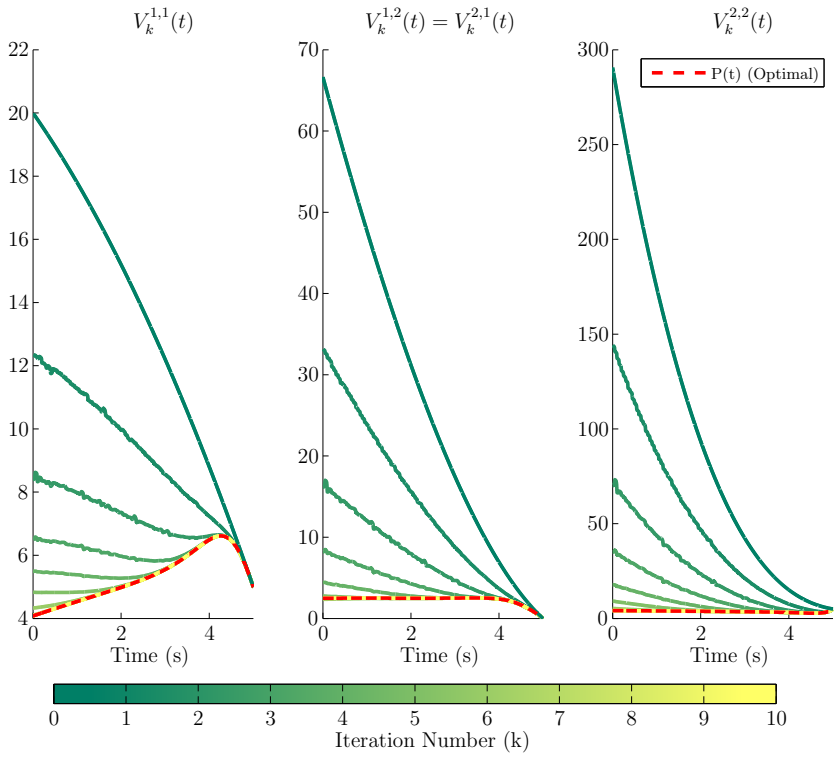

Figure 4: Elements of $V_k(t)$ for Double Integrator System with time step of 0.05 seconds, $\ell = 20$

control gain. This structure may also potentially be used in applications in which the dynamics are slowly varying over iteration, or which are mostly iteration-invariant but may change suddenly at a particular iteration, to ensure that a control scheme which is close to optimal is maintained.

Acknowledgement

This work is supported by Australian Research Council grants DP160104018 and DP130100849.

References

[1] RE Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[2] SH Zhou, J Fong, V Crocher, Y Tan, D Oetomo, and I Mareels. Learning control in robot-assisted rehabilitation of motor skills–a review. *Journal of Control and Decision*, 3(1):19–43, 2016.

[3] DL Kleinman. *Suboptimal design of linear regulator systems subject to computer storage limitations*. PhD thesis, Massachusetts Institute of Technology, 1967.

[4] Y Jiang and ZP Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10):2699–2704, 2012.

[5] FL Lewis and D Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *Circuits and Systems Magazine, IEEE*, 9(3):32–50, 2009.

[6] FY Wang, H Zhang, and D Liu. Adaptive dynamic programming: an introduction. *Computational Intelligence Magazine, IEEE*, 4(2):39–47, 2009.

[7] QM Zhao, H Xu, and J Sarangapani. Finite-horizon near optimal adaptive control of uncertain linear discrete-time systems. *Optimal Control Applications and Methods*, 2014.

[8] P Frihauf, M Krstic, and T Başar. Finite-horizon LQ control for unknown discrete-time linear systems via extremum seeking. *European Journal of Control*, 19(5):399–407, 2013.

[9] JA Frueh and MQ Phan. Linear quadratic optimal learning control (LQL). *International Journal of Control*, 73(10):832–839, 2000.

[10] S Gunnarsson and M Norrlöf. On the design of ILC algorithms using optimization. *Automatica*, 37(12):2011–2016, 2001.

[11] JH Lee, KS Lee, and WC Kim. Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, 36(5):641–657, 2000.

[12] WT Reid. *Riccati Differential Equations*. Academic Press, 1972.

[13] ZP Jiang and Y Wang. A converse lyapunov theorem for discrete-time systems with disturbances. *Systems & control letters*, 45(1):49–58, 2002.

[14] A Björck. *Numerical methods for least squares problems*. Siam, 1996.

[15] DCL Fong and M Saunders. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011.