Online Convex Optimization of Forgetting Factor for Recursive Least Squares Estimation of a Time Varying Parameter

Alexandre Amice Raphael Van Hoffelen

12/19/2019

Abstract

This paper considers the selection of the forgetting factor in the recursive least squares (RLS) with forgetting factor algorithm applied to single input single output (SISO) systems. In particular, we derive a bound on the expected deviation of the norm of the parameter estimate parameterized by the error and uncertainty of the intial estimate, the noise in the system, and the rate of change of the parameters. Minimization of this error bound provides a principled method for selecting the forgetting factor in RLS avoiding the need for handtuning. We demonstrate the evolution of the bound and the performance of the choice of forgetting factor on a simple torsional spring system.

1 Introduction

The theory of control relies heavily on the assumption that the state of a system can be estimated accurately. This had led to extensive development of the state observer model starting with Luenberger in 1965 [24], and continuing today [13, 21, 35, 41]. Such observers rely on a high fidelity model of the underlying system dynamics to enable accurate predictions of the future state to improve state estimations in the presence of noise [17]. Synthesizing these models through system identification can be involved, time consuming, and costly with entire books written on how to perform this analysis for large systems such as aircrafts [37], and submarines [6].

However, offline system identification is not always sufficient to provide the necessary high confidence estimates of the system's parameters. This is particularly true in systems where certain parameters may vary in time such as variations in mass due to fuel consumption [18], voltage drop due to battery depletion [30], and changes in spring stiffness due to wear or corrosion [40]. Such time varying parameters can significantly degrade the model's accuracy which in turn can lead to poor controller performance.

Recursive least squares (RLS) with forgetting factor is a perhaps the simplest online parameter estimation algorithm. Dating back to Gauss in 1821 [34], the algorithm provides an efficient, recursive method for estimating parameters in a linear system. A forgetting factor is introduced to enable the algorithm to track time varying system parameters by more heavily weighing recent data [33]. As with many estimation algorithms, the forgetting factor is left as a design parameter, tuned in order to improve the performance. Such tuning often requires expert knowledge of the system and significant experimentation to determine the best choice of this parameter. This has led to extensive literature on heuristics for selecting this forgetting factor and varying it online [2, 31, 32]

This experimental selection of hyperparameters is quite common for learning problems [10, 12, 25]. As the number of such parameters scales in the system, such tuning can become prohibitively difficult. This paper seeks to provide a principled method for selecting the forgetting factor of RLS in order to demonstrate that hand tuning can be avoided by exploiting prior system knowledge and structural information about the RLS estimation. This is achieved by minimizing an upper bound on the expected error of the parameter

estimate. In section 1.1, we discuss related work in the area followed by our contributions in 1.2. We then proceed by formally introducing the identification problem and RLS in section 2 before proceeding to the statement and proof of our bound in section 3. Finally, in section 4 we provide a handful of numerical simulations demonstrating how our bound from section 3 evolves as the forgetting factor is varied and how the algorithm performs on a simulation of a coupled motor system.

1.1 Related work

The RLS algorithm has been studied extensively since its reintroduction in the 1950s by Plackett [28]. Extensive work was done to understand the algorithm's rate of convergence when estimating stationary parameters [26], when using variable forgetting factors [8], and when using its estimate during online control [20].

The selection of forgetting factor in RLS has also received some attention in the past. Ljung dedicates an entire chapter of [22, Ch. 11] discussing the RLS algorithm and includes a section on the choice of forgetting factor. He provides the common heuristic of selecting the forgetting factor such that the measurements that are relevant remain strongly weighted. He notes that measurements older than $\frac{1}{1-\lambda}$ where λ is a forgetting factor are included in the estimate with a weight of approximately $e^{-1} \approx 36\%$ in subsequent updates. In practice, the time parameters of the system are expected to change slowly resulting in the folk theorem of always choosing $\lambda \in [0.95, 1)$ with 0.98 being a common choice [16].

In [38], the authors use the RLS with forgetting factor algorithm in order to estimate time varying parameters of a vehicle. They note that though RLS has been used widely in practice, little rigorous mathematical analysis of the algorithm has been done demonstrating its tracking capabilities. Campi in [4], considers the case of a parameter which drifts stochastically in time and performs a formal analysis on sufficient conditions for the algorithm to track. In [14], the authors provide minimum bounds on the evolution of the estimate covariance matrix in terms of the choice of forgetting factor to ensure convergence of the error. Direct methods for selecting the forgetting factor come from [39] where the authors propose choosing the forgetting factor by minimizing the probability of a type II error. The authors in [9] provide a method for performing this selection by deriving a bound on the expected error of the parameter estimate for parameters which evolve according to a zero mean, stationary process.

1.2 Contributions

In this work, we provide a method of selecting the optimal choice of forgetting factor by minimizing the expected error of the parameter estimate as in [9]. Our framework generalizes to an arbitrary class of functions with universally bounded rates of change. Such functions include Lipschitz continuous functions [36] and several stochastic processes such as random walks [29]. Our bound is parameterized by the initial error and uncertainty of the estimate, the power of the noise in the system, the rate of change of the parameters, and the forgetting factor. As the bound is a rational function of the scalar forgetting factor, its minimum can be efficiently computed [1] on the bounded range $(0, 1)$. We provide numerical simulations demonstrating the utility in selecting the forgetting factor in this manner for a simple system while also demonstrating how the bound can predict regimes of instability for poor choices of forgetting factor.

2 Problem Formulation

We consider the general, single input, single output (SISO) system:

$$y(t) = (a_1(t)y(t-1) + a_2(t)y(t-2) + \cdots + a_n(t)y(t-k)) + \\ (b_0(t)u(t) + b_1(t-1)u(t-1) + \dots b_m(t)u(t-m)) + v(t) \quad (1)$$

where $\{u(t)\}_{t-m}^{t}$ is a set of input sequences, $\{y(t)\}_{t-n}^{t}$ is a set of observations, $v(t)$ is random noise.

The purpose of the system identification problem is to obtain a model such that we can predict the output $y(t)$ of the system given a set of input actions [22]

We gather the input and output sequences into the size $n = k + m + 1$ information vector:

$$\phi(t) \coloneqq \begin{bmatrix} y(t-1) & y(t-2) & \dots y(t-k) & u(t) & u(t-1) & \dots u(t-m) \end{bmatrix}^T \quad (2)$$

and the parameters into the n dimensional parameter vector:

$$\theta(t-1) = \begin{bmatrix} a_1(t) & a_2(t) & \dots & a_n(t) & b_0(t) & b_1(t) & \dots & b_m(t) \end{bmatrix}^T \quad (3)$$

The dynamic system from (1) can be written as:

$$y(t) = \phi^T(t)\theta(t-1) + v(t) \quad (4)$$

An online estimate of the parameter vector $\hat{\theta}(t)$ and the covariance of this estimate $P(t)$ can be computed via the recursive least squares (RLS) algorithm [23]

$$P^{-1}(t) = \lambda P^{-1}(t-1) + \phi(t)\phi^T(t) \quad (5)$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\phi(t)\left[y(t) - \phi^T(t)\hat{\theta}(t-1)\right] \quad (6)$$

where $\lambda \in (0, 1]$ is a forgetting factor meant to bias the estimate towards more recent information to enable time-varying parameter tracking capabilities [33].

By leveraging the matrix inversion lemma [15] and the fact that $P(t)$ is nonsingular [7], this update can be written in recursive form, bypassing the computationally expensive matrix inversion.

$$P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{1}{\phi^T(t)P(t-1)\phi(t)}P(t-1)\phi(t)\phi^T(t)P(t-1)\right) \quad (7)$$

Extensive literature exists on the convergence of this algorithm to an accurate estimate of the parameter vector in the case of $\lambda = 1$ and stationary parameter vector [23], time-varying choice of λ and time-varying parameter vector [5], and for non-linear estimation using kernel methods [11].

The purpose of this paper is to provide a method of selecting the forgetting factor λ by minimizing the expected error on the deviation of $\left\|\hat{\theta}(t) - \theta(t)\right\|$. In the following section we provide a bound parametrized by the forgetting factor, initial deviation and uncertainty of the estimate, the noise in the system, and the rate of variation of the parameters.

3 Main Results

In this section we consider a probabilistic bound on the error $\left\|\hat{\theta}(t) - \theta(t)\right\|$ characterized by the following theorem.

Theorem 1. *Let* $\Delta\theta(t) = \hat{\theta}(t) - \theta(t)$ *such that*

$$\mathbb{E}[\Delta\theta(t)^T \Delta\theta(t)] \leq D^2 \tag{8}$$

$v(t) \sim \mathcal{N}(0, \sigma_v^2)$ *such that:*

$$\mathbb{E}[v(t)v(i)] = \mathbf{0} \textit{ for } i \neq t \qquad \mathbb{E}[\phi(t-i)v(t)] = \mathbf{0} \qquad \textit{for } i \geq 0 \tag{9}$$

. Suppose there exists an $N \gg n$ *such that we have strong persistence of excitation:*

$$\alpha I \preceq \sum_{i=1}^{N} \phi(t+i)\phi^T(t+i) \preceq \beta I \tag{10}$$

and let

$$\frac{1-\lambda}{N\beta} \leq p_0 \leq \frac{1-\lambda}{\alpha}$$

Then we have that

$$\mathbb{E}\left[\left\|\hat{\theta}(t) - \theta(t)\right\|^2\right] \leq \left(\frac{3\,\mathbb{E}\left[\left\|\hat{\theta}(0) - \theta(0)\right\|^2\right]}{(\alpha p_0)^2}\right)^2 \lambda^{2(t-N+1)}(1-\lambda)^2 + \left(\frac{n\sigma_v^2}{\alpha}\right)\left(\frac{(1-\lambda)}{\lambda^{N-1}}\right) + \left(\frac{3ND\beta}{\alpha\sqrt{2}}\right)^2 \left(\frac{1-\lambda^t}{\lambda^{N-1}(1-\lambda)}\right)^2 \tag{11}$$

$$= \underbrace{K_0 \lambda^{2(t-N+1)}(1-\lambda)^2}_{T_0} + \underbrace{K_v \frac{(1-\lambda)}{\lambda^{N-1}}}_{T_v} + \underbrace{K_\Delta \left(\frac{1-\lambda^t}{\lambda^{N-1}(1-\lambda)}\right)^2}_{T_\Delta} \tag{12}$$

This thereom decomposes the expectation of the error into three parts. The first term gives an error proportional to the initial error of the estimation $\hat{\phi}(0)$ and its uncertainty which are constant. Notice that if $\lambda < 1$ then as $t \to \infty$ we have that $T_0 \to 0$. If $\lambda = 1$ then $T_0 = 0$. T_v gives a bound on the error incurred by the strength of the noise. The easier it is to excite our system and the less complex the system dimension, the lower the penalty incurred by this term. This term is constant with respect to t as this noise provides a finite bound on the certainty achievable within our system. Notice again that if $\lambda = 1$, then $T_1 = 0$. Finally, the third term provides an upper bound on the uncertainty caused by the change in the system's parameters. Notice that as $t \to \infty$, then $T_\Delta \to \frac{K_\Delta}{\lambda^{N-1}(1-\lambda)}$ meaning the optimal choice must balance the trade off between forgetting information to ensure tracking ability and remembering information to suppress noise. This trade off is characterized by the fact that $T_\Delta \to \infty$ as $\lambda \to 0$ and $\lambda \to 1$. We remark that the bound is a rational function of λ for which many numerical methods exist for finding its minimum [3, 19]. Furthermore, if $D = 0$, i.e. the parameter is not changing then it is clear that this bound is minimized by $\lambda = 1$ recovering the asymptotic convergence result of recursive least squares [27].

3.1 Proof of Theorem 1

The proof of the theorem relies on the following lemma from [9]

Lemma 2. *Suppose that equation* (10) *holds. Then the covariance matrix* $P(t)$ *obeys the following ordering*

in the semi-definite cone of matrices

$$\frac{1-\lambda}{N\beta} I \preceq P(t) \preceq \frac{1-\lambda}{\lambda^{N-1}\alpha} I \tag{13}$$

for $0 < \lambda < 1$

Proof. See Appendix A or [9] □

We now prove theorem 1 drawing heavily from both [22] and [9]

Proof. Begin by noting the following decomposition of the error term:

$$\begin{aligned}
\hat{\theta}(t) - \theta(t) &= \left(\hat{\theta}(t-1) + P(t)\phi(t)\left[y(t) - \phi^T(t)\hat{\theta}(t-1)\right]\right) - (\Delta\theta(t) + \theta(t-1)) \\
&= \left[I - P(t)\phi(t)\phi^T(t)\right]\hat{\theta}(t-1) + P(t)\phi(t)y(t) - \theta(t-1) - \Delta\theta(t) \\
&= \left[I - P(t)\phi(t)\phi^T(t)\right]\hat{\theta}(t-1) + P(t)\phi(t)\phi^T(t)\theta(t-1) - P(t)\phi(t)v(t) - \theta(t-1) - \Delta\theta(t) \\
&= \left[I - P(t)\phi(t)\phi^T(t)\right]\left[\hat{\theta}(t-1) - \theta(t-1)\right] - P(t)\phi(t)v(t) - \Delta\theta(t) \\
&= P(t)\left[P^{-1}(t) - \phi(t)\phi^T(t)\right]\left[\hat{\theta}(t-1) - \theta(t-1)\right] - P(t)\phi(t)v(t) - \Delta\theta(t)
\end{aligned}$$

Note that from the recursive update of $P^{-1}(t)$ we have

$$P^{-1}(t) = \lambda P^{-1}(t-1) + \phi(t)\phi^T(t) \tag{14}$$

$$\Rightarrow P^{-1}(t) - \phi(t)\phi^T(t) = \lambda P^{-1}(t-1) \tag{15}$$

$$\Rightarrow \hat{\theta}(t) - \theta(t) = \lambda P(t)P^{-1}(t)\left[\hat{\theta}(t-1) - \theta(t-1)\right] - P(t)\phi(t)v(t) - \Delta\theta(t) \tag{16}$$

Finally, by noting the recursive structure of the above, we have:

$$\begin{aligned}
\hat{\theta}(t) - \theta(t) = \lambda P(t)P^{-1}(t-1)\Bigg[&P(t-1)\lambda P^{-1}(t-2)\left[\hat{\theta}(t-2) - \theta(t-2)\right] - \\
&P(t-1)\phi(t-1)v(t-1) - \Delta\theta(t-1)\Bigg] - P(t)\phi(t)v(t) - \Delta\theta(t) \\
= \lambda^2 P(t)P^{-1}(t-2)&\left[\hat{\theta}(t-2) - \theta(t-2)\right] - P(t)\left(\lambda^{t-t+1}\phi(t-1)v(t-1) - \lambda^{t-t}\phi(t)v(t)\right) + \\
&P(t)\left(\lambda^{t-t}P^{-1}(t)\Delta\theta(t) - \lambda^{t-t+1}P^{-1}(t-1)\Delta\theta(t)\right)
\end{aligned}$$

Carrying out this recursion until $t = 0$ results in:

$$\hat{\theta}(t) - \theta(t) = \underbrace{\lambda^t P(t)P^{-1}(0)\left(\hat{\theta}(0) - \theta(0)\right)}_{S_1} + \underbrace{P(t)\sum_{i=1}^{t}\lambda^{t-i}\phi(i)v(i)}_{S_2} - \underbrace{P(t)\sum_{i=1}^{t}\lambda^{t-i}P^{-1}(i)\Delta\theta(i)}_{S_3} \tag{17}$$

We defer to Appendix B the proof that

$$\mathbb{E}\left[\left\|\hat{\theta}(t) - \theta(t)\right\|^2\right] \leq 3\,\mathbb{E}\left[\|S_1\|^2\right] + \mathbb{E}\left[\|S_2\|^2\right] + 3\,\mathbb{E}\left[\|S_3\|^2\right]$$

To complete the proof, we must now bound these three expectations. From [22, Ch. 8,10] and lemma 2

$$\mathbb{E}[\|S_1\|^2] = \lambda^{2t}\,\mathbb{E}\left[\left\|P(t)P^{-1}(0)\left(\hat{\theta}(0)-\theta(0)\right)\right\|^2\right] \tag{18}$$

$$\leq \lambda^{2t}\lambda^2_{\max}[P(t)]\lambda^2_{\max}[P^{-1}(0)]\,\mathbb{E}\left[\left\|\hat{\theta}(0)-\theta(0)\right\|^2\right] \tag{19}$$

$$\leq \lambda^{2t}\left(\frac{1-\lambda}{\alpha\lambda^{N-1}}\right)^2 p_0^2\,\mathbb{E}\left[\left\|\hat{\theta}(0)-\theta(0)\right\|^2\right] \tag{20}$$

where the last equality follows from lemma 2. Using the circular symmetry of the trace[15] we have

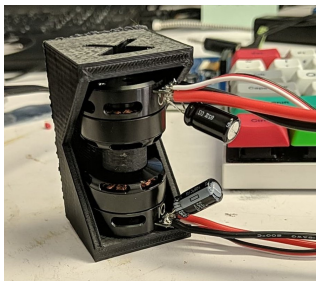
$$\begin{aligned}
\mathbb{E}[\|S_2\|^2] &= \mathbb{E}\left[\mathbf{tr}\left(\left(\sum_{i=1}^{t}\lambda^{t-i}v^T(i)\phi^T(t)\right)P^T(t)P(t)\left(\sum_{i=1}^{t}\lambda^{t-i}\phi(t)v(i)\right)\right)\right] \\
&= \mathbb{E}\left[\mathbf{tr}\left(\left(\sum_{i=1}^{t}\lambda^{t-i}\phi(t)v(i)\right)\left(\sum_{i=1}^{t}\lambda^{t-i}v^T(i)\phi^T(t)\right)P^T(t)P(t)\right)\right] \\
&= \sigma_v^2\left(\frac{1-\lambda}{\alpha\lambda^{N-1}}\right)\mathbb{E}\left[\mathbf{tr}\left(P(t)\sum_{i=1}^{t}\lambda^{t-i}\phi(t)\sum_{i=1}^{t}\lambda^{t-i}\phi^T(t)\right)\right]
\end{aligned}$$

From here we note that from (15) we can express

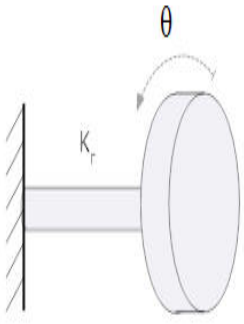
$$\mathbf{tr}\left(P(t)\sum_{i=1}^{t}\lambda^{t-i}\phi(t)\sum_{i=1}^{t}\lambda^{t-i}\phi^T(t)\right) = \mathbf{tr}\left(I-\lambda^t P(t)P^{-1}(0)\right) \leq n$$

since $\lambda^t P(t)P^{-1}(0) \succeq 0$. This yields

$$\mathbb{E}[\|S_2\|^2] \leq n\sigma_v^2\left(\frac{1-\lambda}{\alpha\lambda^{N-1}}\right) \tag{21}$$



(a) Experimental System



(b) Idealized System

Figure 1: Experimental Setup

Finally, again by [22] and the circular symmetry of trace [15]

$$\mathbb{E}[\|S_3\|^2] \leq \mathbb{E}\left[\mathbf{tr}\left(\sum_{i=1}^{t}\lambda^{t-i}\left(\Delta\theta(i)\right)^T\left(P^{-1}(i)\right)^T P(t)^T\right)\left(P(t)\sum_{i=1}^{t}\lambda^{t-i}P^{-1}(i)\Delta\theta(i)\right)\right] \tag{22}$$

$$\leq \left(\frac{1-\lambda}{\alpha\lambda^{N-1}}\right)^2 \mathbb{E}\left[\sum_{i=1}^{t}\sum_{j=1}^{t}\mathbf{tr}\left[\lambda^{2t-i-j}\left(\Delta\theta(i)\right)^T\left(P^{-1}(i)\right)^T P^{-1}(j)\Delta\theta(j)\right]\right] \tag{23}$$

$$\leq \left(\frac{1-\lambda}{\alpha\lambda^{N-1}}\right)^2\left(\frac{1}{2}\right)\sum_{i=1}^{t}\sum_{j=1}^{t}\lambda^{2t-i-j}\,\mathbb{E}\left[\mathbf{tr}\left[\left(\Delta\theta(i)\right)^T\left(P^{-2}(i)\right)\left(\Delta\theta(i)\right)\right]+\mathbf{tr}\left[\left(\Delta\theta(j)\right)^T\left(P^{-2}(j)\right)\left(\Delta\theta(j)\right)\right]\right] \tag{24}$$

$$\leq \left(\frac{1-\lambda}{2\alpha\lambda^{N-1}}\right)^2\left(\frac{N\beta}{1-\lambda}\right)^2\sum_{i=1}^{t}\sum_{j=1}^{t}\lambda^{2t-i-j}\,\mathbb{E}\left[\mathbf{tr}\left[\left(\Delta\theta(i)\right)^T\left(\Delta\theta(i)\right)\right]+\mathbf{tr}\left[\left(\Delta\theta(j)\right)^T\left(\Delta\theta(j)\right)\right]\right] \tag{25}$$

$$\leq \left(\frac{2DN\beta}{2\alpha\lambda^{N-1}}\right)\sum_{i=1}^{t}\sum_{j=1}^{t}\lambda^{2t-i-j} = \lambda^{2t}\left(\frac{DN\beta}{\alpha\lambda^{N-1}}\right)\sum_{i=1}^{t}\left(\frac{1}{\lambda}\right)^i\sum_{j=1}^{t}\left(\frac{1}{\lambda}\right)^j \tag{26}$$

$$= \left(\frac{DN\beta}{\alpha\lambda^{N-1}}\right)\left(\frac{1-\lambda^t}{1-\lambda}\right)^2 \tag{27}$$

□

4 Numerical Experiments

We simulated the operation of RLS with the selection of various forgetting factors in MATLAB. We sought to implement this on hardware but were met with difficulty as described in section 4.3

4.1 Experimental System

We consider a coupled motor system where the top motor is attempting to hold the angular position 0 with a simple feedback controller with time varying gain $k(t)$. The bottom motor seeks to the undergo sinusoidal motion path $A\cos(\omega t)$. The bottom motor can measure its angular position $x(t)$, velocity $\dot{x}(t)$, and the

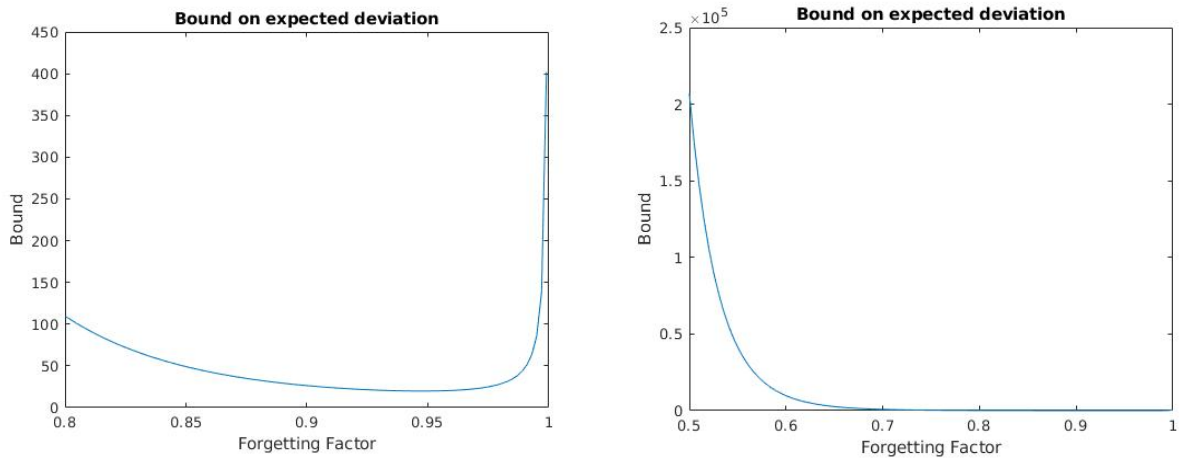


Figure 2: Inspection of the evolution of the bound as a function of forgetting factor

current input torque τ_i, but cannot directly measure the gain k from the top motor. Each motor has moment of inertia I. The setup is shown in figure 1a. We assume 0 friction as the friction is negligible when the system is driven.

This system is equivalent to the torsional spring shown in figure 1b. By integrating the total torque on the system we obtain:

$$I\ddot{x}(t) = -k(t)x(t) + \tau_i \tag{28}$$

$$\Rightarrow I\dot{x}(t+1) = -k(t)\Delta t x(t) + \tau_i \Delta t + \dot{x}(t) \tag{29}$$

$$\Rightarrow Ix(t+1) = -k(t)\Delta t^2 x(t) + \tau_i \Delta t^2 + \dot{x}(t)\Delta t + x(t) \tag{30}$$

$$\Rightarrow \frac{1}{\Delta t^2}\left(-Ix(t+1) + \tau_i \Delta t^2 + \dot{x}(t)\Delta t + x(t)\right) = k(t)x(t) \tag{31}$$

where the left hand side is our system prediction and the right hand side is a single input system with an unknown, time varying parameter. Noise is easily injected into the right hand side. For this experiment, we allow $k(t) = B\cos(\mu t) + (B + 0.5)$. We note that this function has Lipschitz constant $B\mu$, allowing it to fulfill the condition from (8) with $D = B\mu\Delta t$ for theorem 1.

4.2 Simulation Results

We simulate in MATLAB the system in (31) using a 100 second trajectory with a sampling time of 0.01. The bottom motor is tasked with maintaining the trajectory of $x(t) = \cos(\frac{3\pi}{10}t)$. The spring constant of the top motor varies according to $k(t) = 10\cos(\frac{2\pi}{10}t) + 10.5$. The size of our state is 1 and our system is persistently excited due to its oscillating nature with a full period which corresponds to approximately 10 samples. For each experiment we measure the mean square error of the position from the desired trajectory, the mean square error of the estimation of the spring constant $k(t)$, and the mean square error of the necessary input torque into the system τ_i. We inject noise $v(t) \sim \mathcal{N}(0, 0.1)$ into the position measurement.

We begin by noting the complexity of the evolution of the bound and its sensitivity as shown in figure 2. The bound's minimum is clearly in the range of $[0.85, 1)$ which aligns with the folk-logic of choosing the bound in the range $[0.95, 1)$ [16].

We experiment with various choices of forgetting factor. In the first experiment, we run the above system with a a forgetting factor of $\lambda = 0.2$. Predictably, we drive the system to instability and the experiment does not finishing running due to exponentially compounding errors resulting in an overflow of the system

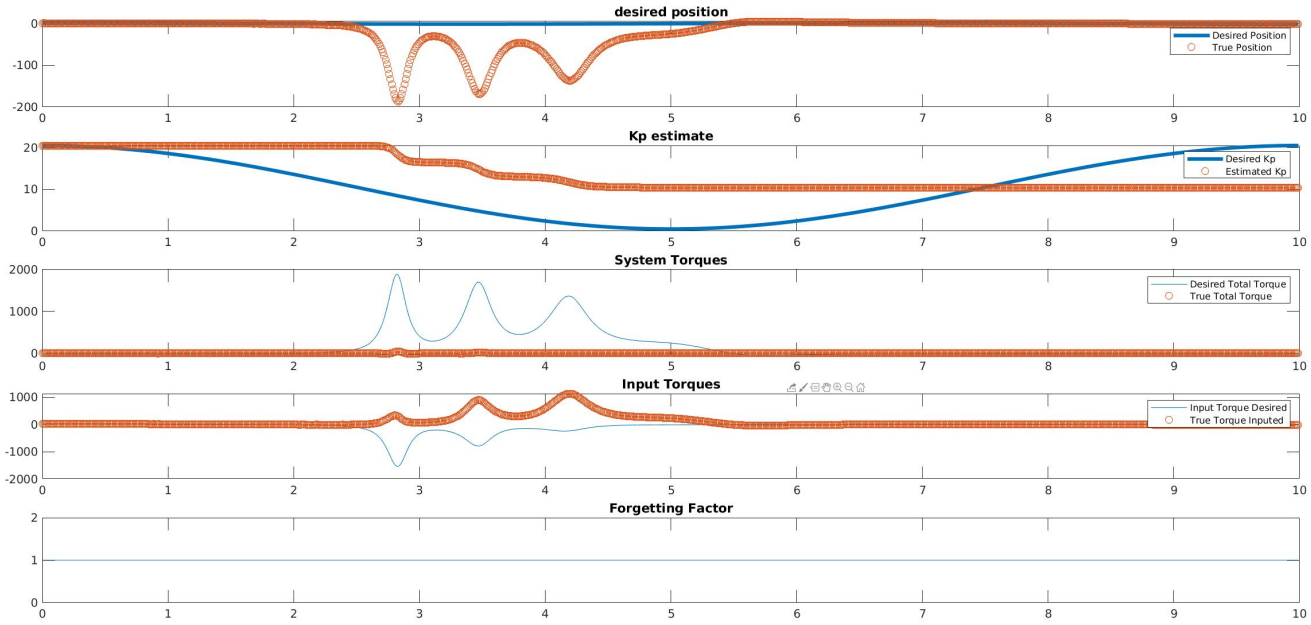


Figure 3: Errors for Forgetting Factor of 1

We next run the system with $\lambda = 1$ as shown in figure 3. This system also significantly overshoots realistic position limits and achieves an MSE on the position of 1760, an MSE on the estimation of the constant of 55.2735 and an MSE on the torque error of 1.7×10^5.

We run the system by dynamically selecting the λ which minimizes the bound in theorem 1 and plot the evolution of the system in figure 4. This simulation achieved an MSE of 0.0075 on the position, 1.1706 on the spring constant, and 0.7511 on the torque. A λ of 0.9474 was chosen for all times. We believe this is due to the low amount of excitation needed to obtain persistence of excitation. We note that the algorithm performs best in regions where the change in the parameter is closest to the computed Lipschitz constant which is where the derivative is maximal. Furthermore, this is the portion of the function with the smallest second derivative and therefore the regime where the changing spring constant is most linear. This shows that a static bound is by no means the tightest bound and that better bounds may be achievable. We note that the majority of the spring constant error occurs in a very limited regime near where the input torque must change directions to continue driving the system.

To demonstrate that it may be possible to choose a better bound on the system if we have stronger information we run an experiment with forgetting factor of 0.5. This leads to samples older than 2 seconds being weighted by a factor of less than 36% [22]. For our current choice of $k(t)$, 2 seconds corresponds to approximately a quarter of one period, leading to a linearization error of no more than 2.6 of any samples within this period. For this experiment, we attained an MSE of position of 0.0048, spring constant of 0.0022 and torque of 0.4853. This very close tracking can be seen in figure 5

Finally, the plot from figure 2 demonstrates that the bound from theorem 1 grows very rapidly if the forgetting factor is too large. We confirm that the error does in fact grow by simulating the same system with $\lambda = 0.97$ and $\lambda = 0.99$. The former leads to significantly larger errors when compared to the MSE of the "optimal" selection made by minimizing the bound. The position MSE remains low at 0.0136, yet the estimate of the spring constant suffers significantly as seen in figure 6 with an MSE of 4.0798 and torque

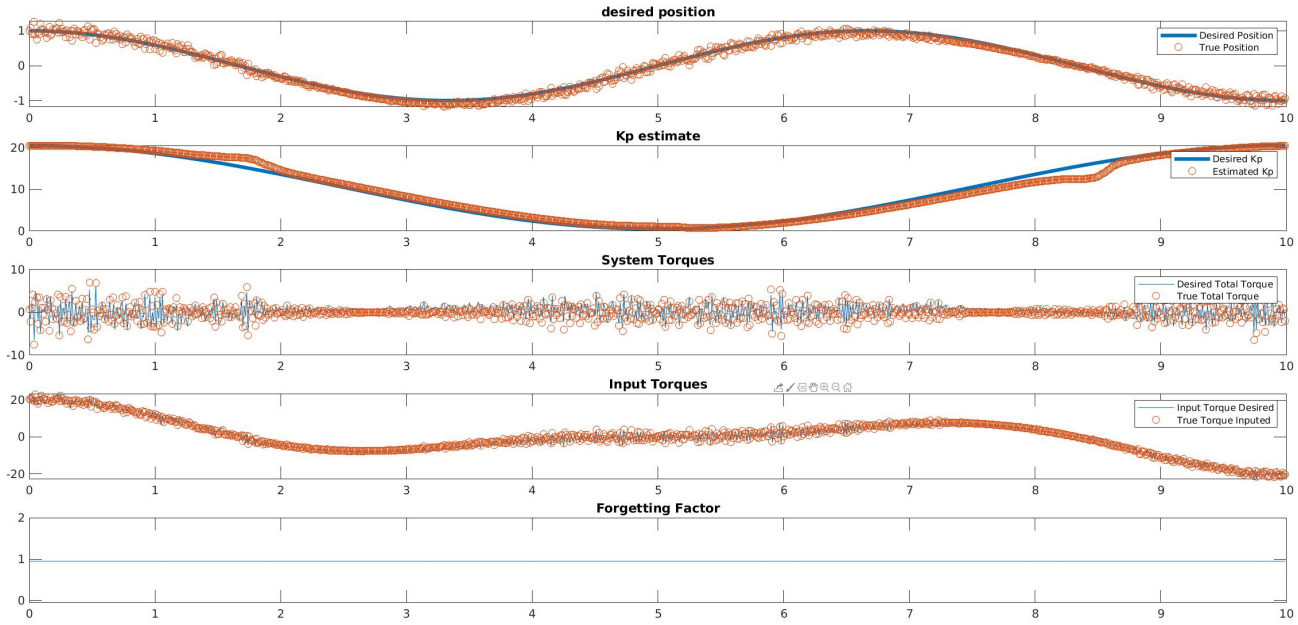


Figure 4: Errors for Optimal Forgetting Factor

MSE of 1.3467.

For $\lambda = 0.99$ we experience even larger deviations of the estimate of the spring constant which in facts leads to a highly unstable input torque towards the end of the experiment as seen in figure 7. It would likely be unsafe for a system to exhibit such erratic behavior, thus motivating the proper selection of forgetting factor.

4.3 Motor Configuration

In addition to the simulation, we sought to test our result on real hardware. We manufactured the setup shown in figure 1a using IQ motion control position modules and a 3D printed frame. A MATLAB API was constructed to enable high level control of the motors without the need of the original low level interface. The k constant of the top motor could be set while the bottom motor could be queried for its position, velocity, and input torque. We sought to run the same experimental set up from the previous section. Unfortunately, we were unable to test as the torque experienced by the 3D printed parts joining the motor shafts as well as the frame holding the two motors together fractured under the combined force of the motors. We believe this was due to the print lines having been lain out parallel to the experienced torques, providing a maximum amount of forced delivered to the weakest portion of the rig.

5 Conclusions and Future Work

We discuss a method of selecting the forgetting factor in recursive least squares based on an upper bound on the expected deviation of the estimate. The bound was derived by decomposing the expected error into three parts relying on the initial error of the estimate, the noise of the system, and the rate of change of the parameter. Experiments were run on several choices of forgetting factors, one of which performed better

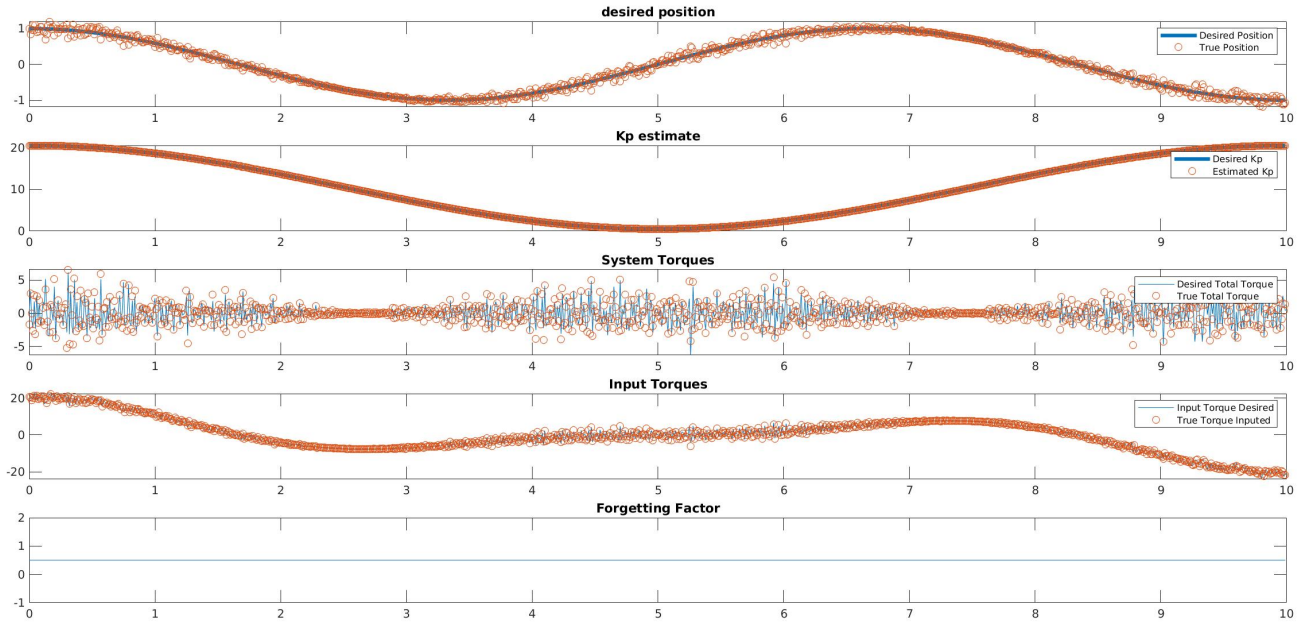


Figure 5: Errors for Forgetting Factor of 0.5

than predicted by the bound from theorem 1.

We demonstrated that though this bound is not tight, it does give a principled method of selecting the choice of forgetting factor on a simple system which performs very well. Better choices can be made using simple intuition if more information is known about the change in the function. Furthermore, we demonstrated that such a choice can be very important as certain choices of forgetting factor can cause the system to become unstable during operation if the forgetting factor is chosen to be too large or too small.

Our work relied heavily on work that dates from before 2000. The choice of hyperparameters has important applications and thus such principled selection warrants being revisited and this work stands to be improved by extension. In this vein, an attempt was made to extend this proof to SIMO systems, yet this attempt failed due to the complexity of estimating the random matrix. We believe such a bound is within reach and Alexandre may attempt to continue with this next semester. Other areas of future work may investigate the performance of different classes of functions which can have universally bounded inner product of evolution. This work considered a simple Lipschitz estimate of the rate of change of the parameters, yet the bound in theorem 1 does not depend on this choice.

References

[1] G. Aronsson, M. Crandall, and P. Juutinen. A tour of the theory of absolutely minimizing functions. *Bulletin of the American mathematical society*, 41(4):439–505, 2004.

[2] S. Bittanti and M. Campi. Tuning the forgetting factor in rls identification algorithms. In *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, pages 1688–1689. IEEE, 1991.

[3] F. Bugarin, D. Henrion, and J. B. Lasserre. Minimizing the sum of many rational functions. *Mathematical Programming Computation*, 8(1):83–111, 2016.

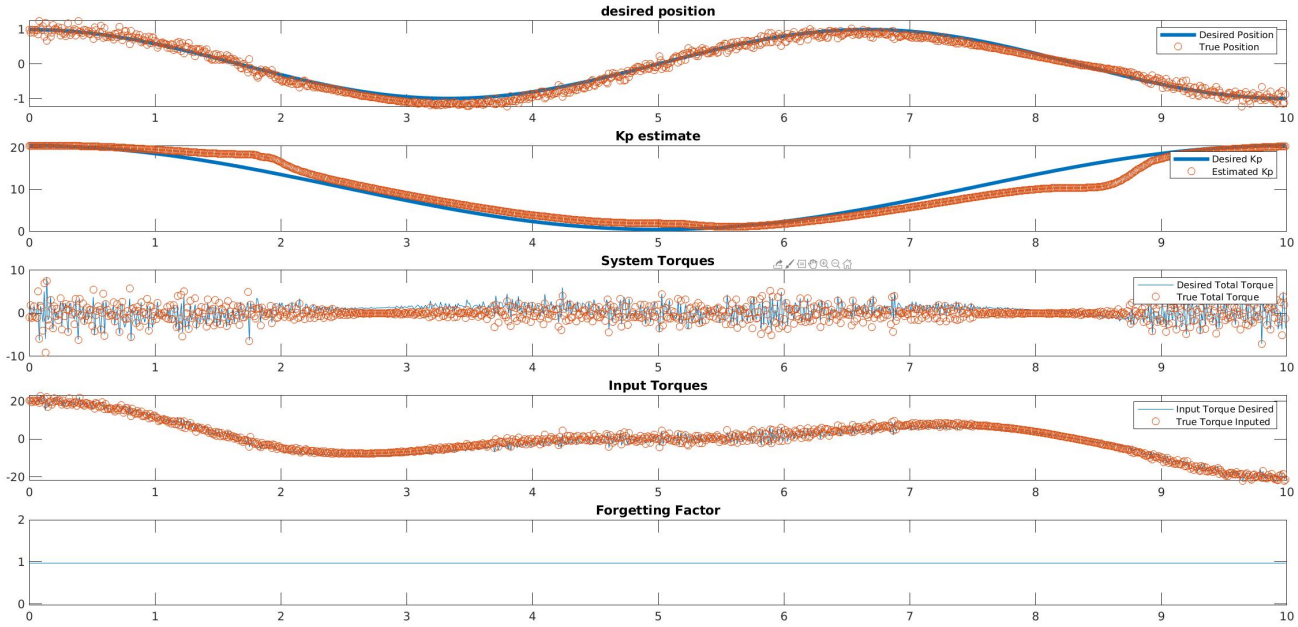


Figure 6: Errors for Forgetting Factor of 0.97

[4] M. Campi. Performance of rls identification algorithms with forgetting factor: A -mixing approach. *Journal of Mathematical Systems Estimation and Control*, 7(1):29–54, 1997.

[5] R. M. Canetti and M. D. España. Convergence analysis of the least-squares identification algorithm with a variable forgetting factor for time-varying linear systems. *Automatica*, 25(4):609–612, 1989.

[6] P. J. Coxon. *System identification of submarine hydrodynamic coefficients from simple full scale trials*. PhD thesis, Massachusetts Institute of Technology, 1989.

[7] M. Dahleh, M. A. Dahleh, and G. Verghese. Lectures on dynamic systems and control. *A+ A*, 4(100): 1–100, 2004.

[8] S. Dasgupta and Y.-F. Huang. Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise. *IEEE Transactions on information theory*, 33(3):383–392, 1987.

[9] F. Ding and T. Chen. Performance bounds of forgetting factor least-squares algorithms for time-varying systems with finite measurement data. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(3):555–566, 2005.

[10] K. Duan, S. S. Keerthi, and A. N. Poo. Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51:41–59, 2003.

[11] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm. *IEEE Transactions on Signal Processing*, 52(8):2275–2285, Aug 2004. ISSN 1941-0476. doi: 10.1109/TSP.2004.830985.

[12] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

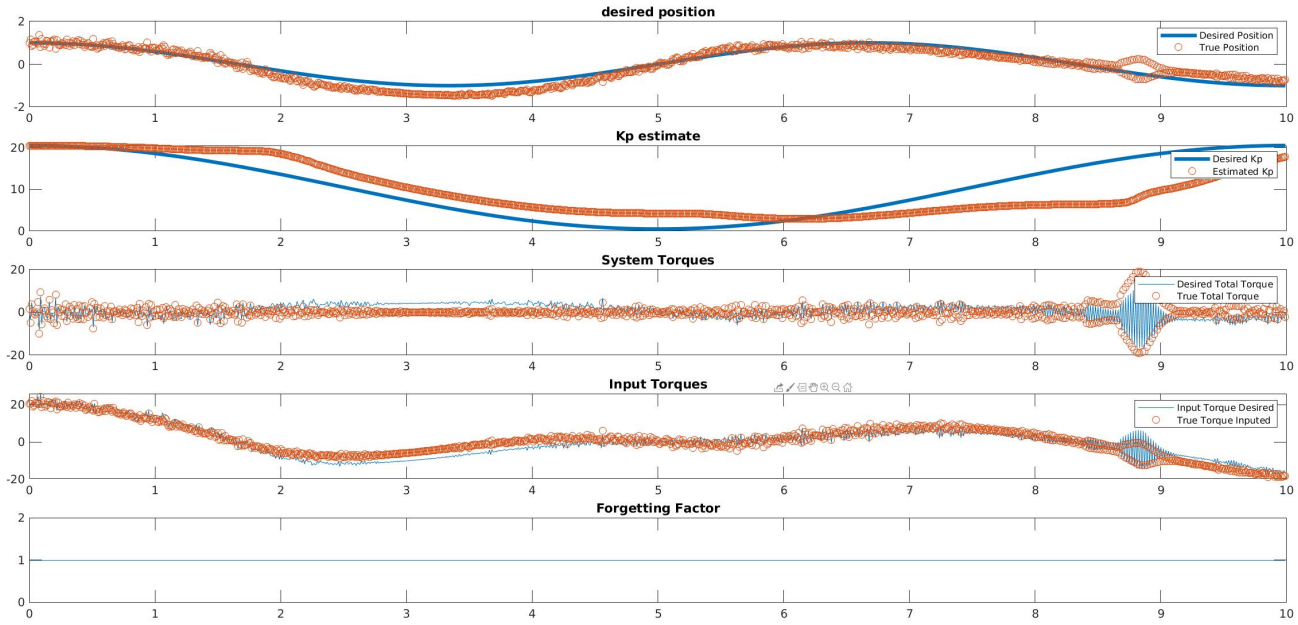


Figure 7: Errors for Forgetting Factor of 0.99

[13] B.-Z. Guo and Z.-l. Zhao. On the convergence of an extended state observer for nonlinear systems with uncertainty. *Systems & Control Letters*, 60(6):420–430, 2011.

[14] L. Guo, L. Ljung, and P. Priouret. Performance analysis of the forgetting factor rls algorithm. *International journal of adaptive control and signal processing*, 7(6):525–537, 1993.

[15] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[16] E. C. Ifeachor and B. W. Jervis. *Digital signal processing: a practical approach*. Pearson Education, 2002.

[17] R. E. Kalman et al. Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2): 102–119, 1960.

[18] H. Khadilkar and H. Balakrishnan. Estimation of aircraft taxi fuel burn using flight data recorder archives. *Transportation Research Part D: Transport and Environment*, 17(7):532–537, 2012.

[19] J. Kostrowicki and H. A. Scheraga. Simple global minimization algorithm for one-variable rational functions. *Journal of Global Optimization*, 6(3):293–311, 1995.

[20] P. Kumar. Convergence of adaptive control schemes using least-squares parameter estimates. *IEEE Transactions on Automatic Control*, 35(4):416–424, 1990.

[21] S. Li, J. Yang, W.-H. Chen, and X. Chen. Generalized extended state observer based control for systems with mismatched uncertainties. *IEEE Transactions on Industrial Electronics*, 59(12):4792–4802, 2011.

[22] L. Ljung. System identification. *Wiley Encyclopedia of Electrical and Electronics Engineering*, pages 1–19, 1999.

[23] L. Lozano. Convergence analysis of recursive identification algorithms with forgetting factor. *Automatica*, 19(1):95–97, 1983.

[24] D. Luenberger. Observers for multivariable systems. *IEEE Transactions on Automatic Control*, 11(2): 190–197, 1966.

[25] J. Luketina, M. Berglund, K. Greff, and T. Raiko. Scalable gradient-based tuning of continuous regularization hyperparameters. In *International conference on machine learning*, pages 2952–2960, 2016.

[26] A. Marcet and T. J. Sargent. Convergence of least squares learning mechanisms in self-referential linear stochastic models. *Journal of Economic theory*, 48(2):337–368, 1989.

[27] V. Panuska. An adaptive recursive-least-squares identification algorithm. In *1969 IEEE Symposium on Adaptive Processes (8th) Decision and Control*, pages 65–65. IEEE, 1969.

[28] R. L. Plackett. Some theorems in least squares. *Biometrika*, 37(1/2):149–157, 1950.

[29] P. Révész. *Random walk in random and non-random environments*. World Scientific, 2005.

[30] B. Saha and K. Goebel. Modeling li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society*, pages 2909–2924, 2009.

[31] S. L. Shah and W. R. Cluett. Recursive least squares based estimation schemes for self-tuning control. *The Canadian Journal of Chemical Engineering*, 69(1):89–96, 1991.

[32] C. F. So, S. C. Ng, and S. H. Leung. Gradient based variable forgetting factor rls algorithm. *Signal Processing*, 83(6):1163–1175, 2003.

[33] S. Song, J.-S. Lim, S. Baek, and K.-M. Sung. Gauss newton variable forgetting factor recursive least squares for time varying parameter tracking. *Electronics letters*, 36(11):988–990, 2000.

[34] H. W. Sorenson. Least-squares estimation: from gauss to kalman. *IEEE spectrum*, 7(7):63–68, 1970.

[35] D. J. Stilwell and W. J. Rugh. Interpolation of observer state feedback controllers for gain scheduling. *IEEE transactions on automatic control*, 44(6):1225–1229, 1999.

[36] R. S. Strichartz. *The way of analysis*. Jones & Bartlett Learning, 2000.

[37] M. B. Tischler and R. K. Remple. Aircraft and rotorcraft system identification. *AIAA education series*, page 72, 2006.

[38] A. Vahidi, A. Stefanopoulou, and H. Peng. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics*, 43(1):31–55, 2005.

[39] S. Van Vaerenbergh, I. Santamaría, and M. Lázaro-Gredilla. Estimation of the forgetting factor in kernel recursive least squares. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2012.

[40] Y. Wang, C. Soutis, M. Yar, and X. Zhou. Modelling corrosion effect on stiffness of automotive suspension springs. *Material design & processing communications.*, 1(1), 2019-02. ISSN 2577-6576.

[41] J. Yao, Z. Jiao, and D. Ma. Adaptive robust control of dc motors with extended state observer. *IEEE Transactions on Industrial Electronics*, 61(7):3630–3637, 2013.

A Proof of Lemma 2

We proceed from the update of $P^{-1}(t)$

$$\begin{aligned}
P^{-1}(t) &= \lambda P^{-1}(t-1) + \phi(t)\phi^T(t) \\
&= \sum_{i=1}^{t} \lambda^{t-i}\phi(i)\phi^T(i) + \lambda^T P^{-1}(0) && \text{achieved by recursing backward to } i = 0. \\
&\preceq \sum_{i=1}^{t} \lambda^{t-i} \left(\sum_{k=0}^{N-1} \phi(i+k)\phi^T(i+k) \right) + \lambda^T P^{-1}(0) && \text{adding PSD terms to the summand}
\end{aligned}$$

From the persistence of excitation condition, we have that

$$\begin{aligned}
\sum_{k=0}^{N-1} \phi(i+k)\phi^T(i+k) &\preceq N\beta I \\
\Rightarrow \sum_{i=1}^{t} \lambda^{t-i} \left(\sum_{k=0}^{N-1} \phi(i+k)\phi^T(i+k) \right) &\preceq \sum_{i=1}^{t} \lambda^{t-i}(N\beta I)
\end{aligned}$$

Therefore we have

$$\begin{aligned}
P^{-1}(t) &\preceq N\beta I \sum_{i=1}^{t} \lambda^{t-i} + \lambda^t P^{-1}(0) \\
&= N\beta I \frac{1-\lambda^t}{1-\lambda} + \lambda^t P^{-1}(0) && \text{geometric series with } \lambda \leq 1 \\
&= \frac{N\beta}{1-\lambda} I + \lambda^t \left[P^{-1}(0) - \frac{N\beta}{1-\lambda} I \right] \\
&\preceq \frac{N\beta}{1-\lambda} I && \text{by the condition on } p_0 \text{ from (11)}
\end{aligned}$$

Similarly,

$$
\begin{aligned}
NP^{-1}(t) &= N\lambda^t P^{-1}(0) + N\sum_{i=1}^{t}\lambda^{t-i}\phi(i)\phi^T(i) \\
&\succeq N\lambda^t P^{-1}(0) + \sum_{i=1}^{N}\sum_{j=i}^{t-N+j}\lambda^{t-j}\lambda(j)\lambda^T(j) && \text{telescopic sum} \\
&= N\lambda^t P^{-1}(0) + \sum_{i=1}^{N}\sum_{j=1}^{t-N+1}\lambda^{t-j-i}\lambda(j+i)\lambda^T(j+i) && \text{reindexing} \\
&\succeq N\lambda^t P^{-1}(0) + \sum_{i=1}^{N}\lambda^{t-i}\sum_{j=1}^{t-N+1} N\alpha I && \lambda^{-j}\geq 1 \text{ and (10)} \\
&= \frac{N\alpha I\lambda^{N-1}}{1-\lambda} + N\lambda^t\left[P^{-1}(0) - \frac{\alpha}{1-\lambda}I\right] \\
&\succeq \frac{N\alpha I\lambda^{N-1}}{1-\lambda} && \text{by assumption on } p_0
\end{aligned}
$$

B Proof of Decomposition of Expectation of Equation (17)

We have that:

$$
\begin{aligned}
\mathbb{E}\left[\left\|\hat{\theta}(t)-\theta(t)\right\|^2\right] &= \mathbb{E}\left[\left(S_1+S_2+S_3\right)^T\left(S_1+S_2+S_3\right)\right] \\
&= \mathbb{E}\left[\|S_1\|^2+\|S_2\|^2+\|S_3\|^2+2S_1^TS_2+2S_2^TS_3+2S_1^TS_3\right] \\
&= \mathbb{E}\left[\|S_1\|^2\right]+\mathbb{E}\left[\|S_2\|^2\right]+\mathbb{E}\left[\|S_3\|^2\right]+\mathbb{E}\left[2S_1^TS_2\right]+\mathbb{E}\left[2S_2^TS_3\right]+\mathbb{E}\left[2S_1^TS_3\right]
\end{aligned}
$$

We now analyze the cross terms:

$$
\begin{aligned}
\mathbb{E}\left[2S_1^TS_2\right] &= 2\,\mathbb{E}\left[\left(\lambda^t\left(\hat{\theta}(0)-\theta(0)\right)^T\left(P^{-1}(0)\right)^TP(t)^T\right)\left(P(t)\sum_{i=1}^{t}\lambda^{t-i}\phi(i)v(i)\right)\right] \\
&= 2\,\mathbb{E}\left[\left(\lambda^t\left(\hat{\theta}(0)-\theta(0)\right)^T\left(P^{-1}(0)\right)^TP(t)^TP(t)\right)\right]\sum_{i=1}^{t}\lambda^{t-i}\,\mathbb{E}\left[\phi(i)v(i)\right] \\
&= 0
\end{aligned}
$$

Where the first equality follows from the independence of the product $\phi(i)v(i)$ from the terms in the sequence and the second equality follows from the assumption from (9). The same reasoning applies to

$\mathbb{E}[2S_2^T S_3]$:

$$\begin{aligned}
2\,\mathbb{E}\left[S_2^T S_3\right] &= 2\,\mathbb{E}\left[\left(\sum_{i=1}^{t}\lambda^{t-i}v^T\phi^T(i)P(t)^T\right)\left(P(t)\sum_{i=1}^{t}\lambda^{t-i}P^{-1}(i)\Delta\theta(i)\right)\right] \\
&= 2\sum_{i=1}^{t}\lambda^{t-i}\,\mathbb{E}\left[v^T\phi^T(i)\right]\mathbb{E}\left[P(t)^T\left(P(t)\sum_{i=1}^{t}\lambda^{t-i}P^{-1}(i)\Delta\theta(i)\right)\right] \\
&= 0
\end{aligned}$$

Finally

$$\begin{aligned}
2\,\mathbb{E}\left[S_1^T S_3\right] &= 2\,\mathbb{E}\left[\left(\lambda^t\left(\hat{\theta}(0)-\theta(0)\right)^T\left(P^{-1}(0)\right)^T P(t)^T\right)\left(P(t)\sum_{i=1}^{t}\lambda^{t-i}P^{-1}(i)\Delta\theta(i)\right)\right] \\
&\leq 2\,\mathbb{E}\left[S_1^T S_1\right] + 2\,\mathbb{E}\left[S_3^T S_3\right]
\end{aligned}$$

Where the last equality follows from the fact that $2x^T y \leq x^T x + y^T y$