Some Background on Adaptive Estimation

Kostas Tsakalis

January 1, 1998

1 Problem Statement

Given input-output (I/O) data (w, y), we would like to estimate the parameters θ of a linear model

$$y = w^\top \theta + n$$

that produce the "best fit" to the data.

Here, n denotes the noise or fitting error. By the term "best fit," we mean the minimization of the fitting error in some sense. For example, typical norm or norm-like functionals used as arguments in the above minimization problem are:

- $\|n\|_\infty = \text{ess.sup}_t\, |n(t)|$; the L_∞ norm.
- $\|n\|_2 = \left(\int_0^\infty |n(t)|^2\, dt\right)^{1/2}$; the L_2 norm (energy).
- $\|n\|_{2,\delta} = \left(\int_0^\infty e^{2\delta t}|n(t)|^2\, dt\right)^{1/2}$; the exponentially weighted L_2 norm.
- $RMS(n) = \mu_n = \inf\{\mu | \exists c : \forall T, t > 0, \int_t^{t+T} |n(\tau)|^2\, d\tau \le c + \mu^2 T\}$; the root mean square (RMS) power of the signal.

Alternatively, in a recursive formulation, we would like to develop an adaptive law $\dot\theta = f(\ldots)$ making the estimation error

$$\epsilon = \hat y - y = w^\top \theta - y$$

small in some sense.

In both cases, it is often convenient to study the properties of the parameter error $\tilde\theta = \theta - \theta_*$, where θ_* is defined as the "ideal" or "target" parameter vector. The notion of a target parameter vector is useful in distinguishing the best parameter estimate from the minimizer of the above problem. The reason for this distinction becomes important in a system identification context, where obtaining a good fit is of lesser value than the extrapolation properties of the model. In such a case, the estimated parameters are associated (through some kind of map) with a system, referred to as the identified system. The minimizer is biased towards parameters that minimize the estimation error for the observed input signals. This bias can be significant in cases where the inputs do not provide sufficient information in some directions. As a consequence, the identifed system corresponding to the estimation error minimizer, may not be a good predictor of the system output for arbitrary inputs.

The notion of a target parameter vector is also suitable for cases where some a priori information on the system is available, e.g., through first principles modeling or preliminary identification. Depending on the problem at hand, θ_* may be constant ($\dot\theta_* = 0$) or time-varying ($\dot\theta_* = v$). In the latter case, any algorithm that attempts to estimate θ_* from I/O measurements relies on the assumption that θ_* is slowly varying (v is small).

1.1 Remark: The choice of the cost objective (the error functional to be minimized) is essentially dictated by the properties of the noise corrupting the I/O measurements. Yet, least-squares or least-square-like estimators are often selected because of their algorithmic simplicity and good overall properties. $\triangledown\triangledown$

The algorithms derived for the above parameter estimation problem are typically based on optimization principles. There are two basic assumptions used in the analysis of these algorithms.

1.2 Assumption: *All signals (y, w, n) are bounded.* ∎

1.3 Assumption: *The "regressor" vector w is persistently exciting (PE), that is, there exist constants $a_1, a_2, T > 0$ such that for all t it holds that*

$$a_1 I \geq \int_t^{t+T} w(\tau) w^\top(\tau)\, d\tau \geq a_1 I$$

∎

The boundedness assumption is needed in the convergence analysis and, in particular, when the signals are part of a closed-loop system. In certain cases, it can be cirumvented by using arguments that establish boundedness together with the convergence properties. The boundedness condition can be satisfied through normalization of the linear model equations with a suitable normalization signal (say $\sqrt{m}$):

$$\frac{y}{\sqrt{m}} = \frac{w^\top}{\sqrt{m}}\theta + \frac{n}{\sqrt{m}}$$

where the normalized y, w, n are bounded. Typical choices include:

- Static Normalization: $m = w^\top w + c$, $c > 0$.
- Dynamic Normalization: $\dot{m} = -2\delta_0 m + |QU|^2 + q_e$, $m(0) > 0$, where $\delta_0 > 0$, $Q = Q^\top > 0$, $q_e > 0$ and U denotes the I/O pair (see details below).

In both cases, some additional arguments are required to establish the boundedness of the normalized noise term. The static normalization is simple and has many similarities with the standard steepest descent and Newton optimization algorithms. The dynamic normalization, on the other hand, enjoys some system-gain properties that make it attractive for the identification of linear dynamical systems.

1.4 Property: The dynamic normalization signal ($\sqrt{m}$) bounds the output of any stable, strictly proper, linear system with input U and poles in the half plane $Re\ s < -\delta_0$. ▽▽

To illustrate the role of this property, let us consider the linear time-invariant (LTI) plant

$$y = P[u]$$

Under an observability assumption on the nominal part of P, it follows that the map $u \mapsto y$ can be written as

$$\begin{aligned} \dot{x} &= Fx + \theta_1 u + \theta_2 y \\ y &= qx + n \\ n &= \Delta_1[u] + \Delta_2[y] \end{aligned}$$

where, (F, q) is an observable pair, F is Hurwitz, and $[F + \theta_2 q, \theta_1, q, 0]$ is a state-space realization of the nominal part of the plant. The modeling mismatch is described by the term n which is defined as the output of two proper stable systems (stable factor perturbations). This is a very general description of modeling errors. In fact, the topology arising from such a description is the weakest one for which "meaningful results" can be obtained [Vidyasagar].

For the above example, it follows that the dynamic normalization bounds the state x, provided that the eigenvalues of F have real parts less than $-\delta_0$. In addition, $\sqrt{m}$ bounds the mismatch n under the additional assumption that Δ_i are strictly proper with stability margin δ_0.

Finally, the PE assumption is a natural one to require whenever a parameter convergence result is expected ($\theta \to \theta_*$). Loosely speaking, it states that in any sufficiently large interval, enough independent equations are available to solve for the unknown parameters. From a systems viewpoint, it is essentially an observability assumption. That is, θ_* is observable from y, something needed to conclude that –in the noise-free case– whenever $w^\top \theta \to y$ then $\theta \to \theta_*$. To see this, consider the linear system

$$\begin{aligned} \dot{\theta}_* &= 0 \\ y &= w^\top \theta_* \end{aligned}$$

For this system to be uniformly completely observable, the observability gramian must be uniformly bounded and positive definite. Since the state transition matrix of this system is the identity, the observability gramian is simply $\int_{<T>} ww^\top$ where the integral is taken over arbitrary intervals of length T. Notice that since w is a dependent variable, the PE condition must be ultimately translated to a condition on the external inputs of the system. It can be shown [Sastry] that, for a minimal plant, PE is satisfied if the input energy has sufficient spectral lines. (Roughly, each sinusoid corresponds to two equations which are independent if the plant is minimal.)

Although persistent excitation is crucial for the well-posedness of the estimation problem, there are several practical cases where the input is not at the disposal of the designer (e.g., adaptive control, echo cancellation). In these cases the input may not excite the plant in certain directions or, even worse, the excitation may be below the level of the noise (poor signal-to-noise ratio (SNR) in some directions). Still, an adaptive algorithm should be designed so as to minimize the estimation error and maintain bounded parameter estimates. The latter, in particular, is a serious problem since in the absence of sufficient excitation, the noise (or other perturbations) can introduce a significant bias in the minimizers (they can even occur at infinity). This, in turn, causes a slow drift in the parameter estimates and estimation error bursts. Practically useful adaptive algorithms should, therefore, contain mechanisms to cope with this problem as much as possible. Since complete elimination of parameter drifts is impossible at present, adaptive systems should be viewed as a method to optimize among sensible designs, instead of searching over all possible ones.

2 Algorithms for Adaptive Estimation

Throughout this section, we consider the standard linear model estimation problem where the I/O data satisfy

$$y = w^\top \theta_* + n$$

The I/O signals w, y are available for measurement while the noise n is not. The signals satisfy Assumption 1.2, possibly via model normalization. The parameter vector θ_* is unknown but constant or slowly time-varying. In addition, certain algorithms make use of partial information about the noise and/or the unknown parameters. This information is typically in the form of a noise bound and a set containing the unknown parameters (parametric uncertainty set). The basic algorithms for adaptive estimation are presented next, starting with the notationally more compact continuous-time case.

2.1 Gradient

Given an estimate θ of the unknown parameters, the predicted output is defined as $\hat{y} = w^\top \theta$ and the estimation error as

$$\epsilon = \hat{y} - y = w^\top \tilde{\theta} - n$$

Using gradient/steepest descent concepts, an adaptive law for the updating of θ is

$$\dot{\theta} = -\Gamma w \epsilon$$

representing the steepest descent direction for the cost objective ϵ^2. Here, Γ denotes a positive scaling matrix, referred to as the adaptation gain. This is related to step-size and Hessian inversion ideas from classical

optimization. Notice that, unlike numerical optimization, line-searches cannot be used to achieve a uniform decrease of the cost objective. Consequently, the step-size must be determined a priori in a conservative manner to guarantee the stability of the adaptation. Typical choices for the adaptation gain include:

- $\Gamma = \gamma I$, $\gamma > 0$, a scalar gain determining the speed of adaptation. For sufficiently small γ, an increase of the adaptation gain results in faster convergence but incereased susceptibility to noise.
- $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_n)$, $\gamma_i > 0$ a diagonal matrix, or $\Gamma = \Gamma^\top > 0$ in general. In both cases, the selection represents an effort to precondition the Hessian matrix $\int ww^\top$.
- $\Gamma = \frac{\gamma}{1+\gamma w^\top w}$, $\gamma > 0$; in the limit as $\gamma \to \infty$ this choice becomes the well-known Newton algorithm for solving the equation $\epsilon(\theta) = 0$. However, the pure Newton algorithm is not practical since the case $w = 0$ cannot be excluded.

2.1 Analysis: The analysis of the convergence properties of adaptive algorithms is customarily performed using Lyapunov-like arguments based on a positive definite function of the parameter error.

Let $V = \tilde{\theta}^\top \Gamma^{-1} \tilde{\theta}$; V is positive definite, radially unbounded, decrescent function of the parameter error $\tilde{\theta}$. Taking its derivative along the trajectories of θ,

$$\begin{aligned} \dot{V} &= -2w^\top \tilde{\theta}\epsilon = -2\epsilon^2 - 2n\epsilon \\ &\leq -\epsilon^2 + \eta^2 \quad \text{(via completion of squares)} \end{aligned}$$

In the ideal case ($n = 0$), $\dot{V} \leq 0$, therefore V is bounded and θ is bounded. This implies that ϵ and $\dot{\epsilon}$ are bounded (assuming that $\dot{w}$ is bounded). Hence, ϵ^2 is uniformly continuous and

$$V(t) - V(0) = \int_0^t \dot{V} \leq -\int_0^t \epsilon^2$$

implying that ϵ is square-integrable. This, together with its uniform continuity, imply that $\epsilon \to 0$ as $t \to \infty$.

Regarding the parameter error, without any PE assumptions, one can only conclude that V has a limit (as non-increasing and lower bounded) and $\dot{\theta} \to 0$. This, however, does not imply that θ has itself a limit. (E.g., consider the function $\sin(\sqrt{t})$.) Such a result can only be established if the excitation is periodic (La Salle's theorem).

If the PE condition holds, then the linear time-varying system

$$\begin{aligned} \dot{\tilde{\theta}} &= -\Gamma w w^\top \tilde{\theta} \\ \epsilon &= w^\top \tilde{\theta} \end{aligned}$$

is uniformly completely controllable. To show this, use the fact that observability is invariant under output injection. Then the observability properties of the above system are equivalent to the properties of

$$\begin{aligned} \dot{\tilde{\theta}} &= -\Gamma w w^\top \tilde{\theta} + \Gamma w \epsilon \\ \epsilon &= w^\top \tilde{\theta} \end{aligned}$$

whose observability gramian is $\int ww^\top$. Hence, under PE, $\epsilon \to 0$ implies $\tilde{\theta} \to 0$ (see [Sastry] for details).

In fact, PE implies that the zero equilibrium of the parameter error system is uniformly asymptotically stable (and, since linear, exponentially stable). Consequently, in the presence of bounded perturbations, the system trajectories are uniformly ultimately bounded. Furthermore, integrating $\dot{V}$, it follows that

$$\int \epsilon^2 \leq V_0 + \int n^2$$

where V_0 is a constant depending on initial conditions; alternatively, the above inequality implies that $RMS(\epsilon) \leq RMS(n)$.

This performance bound is an important one in adaptive algorithms. It shows that the RMS value of the error is at most as large as the RMS value of the noise, where the latter is the fitting error for the "target" parameter vector. Since the target parameters were arbitrary in the analysis, it follows that with respect to an RMS criterion on the estimation error, gradient adaptation perfroms at least as good as the best fixed-parameter linear model.

This seemingly outstanding property, however, does not have the practical implications one would expect. Consider, for example the case where the algorithm is used in the presence of small bounded noise with arbitrary inputs. Then, let w be small but PE and define $n = w^\top(\theta_0 - \theta_*)$. With w being sufficiently small, the noise can be made smaller than any given bound for any given θ_0. According to the previous analysis, this would imply that $\theta \to \theta_0$, where $|\theta_* - \theta_0|$ can be "arbitrarily" large. Even though such a limit minimizes the estimation error, an adaptation "burst" will appear as soon as the noise changes. In fact, this RMS convergence of adaptive algorithms gives rise to a fundamental performance limitation. That is the worst-case error amplitude (limsup) is roughly as big as the worst-case error within the parametric uncertainty set.

A quick fix of this problem is to disable the adaptation when the level of excitation drops below the noise level (insufficient excitation, poor SNR). Such an approach forfeits some of the intersting properties and applications of adaptive algorithms. Instead, considerable research effort has been devoted to modifications of the basic algorithm, so as to improve its viability in practical situations. However, with the likelihood of finding a panacea being small, the previous comment must be re-stated: *Adaptation should be viewed as an optimization among sensible solutions, rather than a search among all possible solutions.* Implicit in this statement is the requirement for a thorough study of non-adaptive solutions and the (nonlinear) effects of adaptive laws. In other words, good adaptive designs require more work than non-adaptive ones (there is no free lunch). $\triangledown\triangledown$

2.2 Remark: When used for linear system identification, the analysis of the adaptive algorithms must also account for the effect of nonzero initial conditions. Such an effect can be expressed as an exponentially decaying term which leaves all the asymptotic results unaffected. $\triangledown\triangledown$

2.2 Modifications of the basic algorithm

There are two, now fairly standard, modifications of the basic gradient algorithm aiming to remedy its drawbacks under insufficient excitation.

2.3 Dead-zone: The first modification is to disable adaptation when the error drops below an a priori selected threshold. This idea is motivated by SNR concepts, namely if the error is too small then its main component is due to noise and, therefore, adaptation is not reliable. With the dead-zone modification, the adaptive law becomes

$$\dot{\theta} = -\Gamma w \mathcal{DZ}_{n_0}(\epsilon)$$

where $\mathcal{DZ}_{n_0}(.)$ is a dead-zone nonlinearity with threshold n_0. Different versions of the dead-zone can be implemented, the easiest one being the so-called continuous dead-zone (identity-minus-saturation). In all cases, the threshold should be selected such that

$$n_0 > \limsup |n|$$

This condition is also the source of the main drawback of the dead-zone modification. The threshold choice relies on an effective perturbation signal that cannot always be estimated in a tight fashion. As a result, dead-zones must be conservative, thus allowing for a significant performance deterioration. $\triangledown\triangledown$

2.4 Analysis: Letting $V = \tilde{\theta}^\top \Gamma^{-1} \tilde{\theta}$ we have

$$\dot{V} = -2w^\top \tilde{\theta} \mathcal{DZ}_{n_0}(\epsilon) = -2(\epsilon + n)\mathcal{DZ}_{n_0}(\epsilon) \leq 0$$

since if $|\epsilon| > n_0$, then $\text{sign}(\epsilon + n) = \text{sign}(\mathcal{DZ}_{n_0}(\epsilon))$ and if $|\epsilon| < n_0$, then $\mathcal{DZ}_{n_0}(\epsilon) = 0$.

Hence, V is non-increasing and lower-bounded, implying that it is bounded and reaches a limit. Furthermore, using Lipschitz continuous dead-zones, it follows that

$$\limsup |\epsilon| \leq n_0$$

from which one can also obtain that $RMS(\epsilon) \leq n_0$. In other words, the dead-zone adaptation drives the parameter estimates to a residual set where the error is no larger than the threshold.

An attractive feature of the dead-zone modification is that the parameter estimates also converge (to a constant). [1] $\triangledown\triangledown$

2.5 Remark: Summarizing, dead-zone modifications offer the advantage of lim-sup performance according to a prescribed threshold and parameter convergence (implying no bursting). For this, they only require knowledge of an asymptotic bound of the noise. Their drawbacks are that some performance is lost due to the conservative selection of the threshold. The threshold itself is also hard to estimate since it often depends on the system to be identified. More subtle, but perhaps more significant, limitations of the dead-zone modification are that it cannot handle time-varying parameters; moreover, in the event that the dead-zone condition is violated, parameter drifts can occur. In other words, their performance does not degrade "gracefully" with "small" violations of the assumptions. $\triangledown\triangledown$

2.6 Parameter projection: The second type of modifications has its motivation in constrained optimization ideas. It restricts any updates to lie in a prescribed set, representing a rough estimate of the possible target parameters. While the technical part of the analysis is somewhat involved, the key idea is fairly simple. When the parameter estimates are on the boundary of the set, then the update direction is projected onto the set. The update law with the projection modification has the form

$$\dot{\theta} = \mathcal{P}_{\mathcal{M}}(-\Gamma w \epsilon)$$

where $\mathcal{M}$ is a convex, bounded set and $\mathcal{P}_{\mathcal{M}}$ indicates a vector-field projection onto $\mathcal{M}$. For technical reasons, this projection is required to be Lipschitz continuous, something that is achieved by switching-on the projection in a continuous way inside a boundary layer around $\mathcal{M}$. The equations for the implementation of the projection operator are given next.

$$\mathcal{P}_{\mathcal{M}}(z) = \begin{cases} z & \text{if } \psi(\theta) z^\top \theta_\perp \leq 0 \\ \left(I - \psi(\theta) \frac{\Gamma \theta_\perp \theta_\perp^\top}{\theta_\perp^\top \Gamma \theta_\perp}\right) z & \text{otherwise} \end{cases}$$

where $\psi(\theta)$ is a set indicator function (distance) with boundary layer $\epsilon_* > 0$ and $\theta_\perp$ is the unit normal at the boundary of the scaled $\mathcal{M}$ evaluated at θ. For example, when $\mathcal{M}$ is an ellipsoid $\mathcal{M} = \{\theta | (\theta - \theta_c)^\top R^{-1} (\theta - \theta_c) \leq 1\}$, then

$$\begin{aligned} \psi(\theta) &= \min(1, \text{dist}(\theta, \mathcal{M})/\epsilon_*) \\ \text{dist}(\theta, \mathcal{M}) &= \max(0, |\theta - \theta_c|_{R^{-1}} - 1) \\ \theta_\perp &= R^{-1}(\theta - \theta_c)/|R^{-1}(\theta - \theta_c)| \end{aligned}$$

For intersections of multiple ellipsoids and/or half-spaces $\mathcal{M} = \cap \mathcal{M}_i$, the definition

$$\text{dist}(\theta, \mathcal{M}) = \left[\sum_i \text{dist}^2(\theta, \mathcal{M}_i)\right]^{1/2}$$

with a suitable modification of the definition of the normal can be used. $\triangledown\triangledown$

[1] There is a clever argument to show this [Sastry,Goodwin]: since the dead-zone threshold is strictly larger than the noise, we can add and subtract $w^\top \theta_0$ in the error and group one of these terms with the signal ($w^\top \tilde{\theta}$) and the other with the noise. Now for a sufficiently small θ_0 the dead-zone condition holds, and $|\theta - \theta_* + \theta_0|_{\Gamma^{-1}}$ converges. That is the distance of θ from any point in an open neighborhood of θ_* converges, hence θ itself must converge.

2.7 Remark: Similar results can be obtained with "soft projections" as follows:

$$\dot{\theta} = -\Gamma w\epsilon - \Gamma\sigma(|\theta - \theta_c|)(\theta - \theta_c)$$

where $\sigma(.)$ is a function that is zero inside the parametric uncertainty set (a ball around θ_c) and switches to a constant σ_0 inside a boundary layer around the set in a Lipschitz continuous manner. This modification allows the estimates to be unbiased as long as they remain inside the set. But it penalizes parameter drifts outside the set so that the maximum drift is limited (order of $|n|/\sigma_0$). While simple and appealing, this modification has the drawback that it does not enforce strict parameter constraints and, therefore, it cannot be used to avoid singularity or instability regions that appear in some applications.

Another simple version of the soft projection is to use $\sigma(.) = \sigma_0$. This is the so-called "sigma-modification" that penalizes any deviation of the parameters from their initial estimate θ_c. It is very simple but introduces a bias in the estimates of the order of $\sigma_0|\theta_* - \theta_c|$. The sigma-modification is effective when a good estimate of the parameters is available a priori. However, if the initial estimate is poor it may cause the appearence of limit cycles and burst phenomena (especially in adaptive control). $\triangledown\triangledown$

2.8 Analysis: Again, let $V = \tilde{\theta}^\top\Gamma^{-1}\tilde{\theta}$; V is now bounded by virtue of the projection. Moreover, $\dot{V}^{Projected} \leq \dot{V}^{Not-Projected}$. To see this, notice that when the projection is active,

$$\dot{V} = -2w^\top\tilde{\theta}\epsilon - 2\psi(\theta)\frac{\tilde{\theta}^\top\theta_\perp\theta_\perp^\top}{\theta_\perp^\top\Gamma\theta_\perp}z$$

where $z = -\Gamma w\epsilon$, and $\psi z^\top\theta_\perp > 0$. The first term in the above expression is simply $\dot{V}$ in the not-projected case. The second term is non-positive since θ_* is inside the convex set $\mathcal{M}$, implying that $\tilde{\theta}^\top\theta_\perp \geq 0$. Hence,

$$\begin{aligned}\dot{V} &\leq -2w^\top\tilde{\theta}\epsilon = -2\epsilon^2 - 2n\epsilon \\ &\leq -\epsilon^2 + \eta^2 \quad \text{(via completion of squares)}\end{aligned}$$

implying that

$$\int \epsilon^2 \leq V_0 + \int n^2 \quad \text{and} \quad RMS(\epsilon) \leq RMS(n)$$

where V_0 is a constant depending on initial conditions. Notice that this inequality does not mean that the performance of the gradient-with-projection is the same as the performance of the pure gradient adaptation; it is just the worst-case bounds that are the same.

Gradient+projection algorithms feature some very desirable characteristics. Their RMS performance is at least as good as any fixed model within the parametric uncertainty set. Moreover, their lim-sup performance is comparable to the worst-case fixed model within the set. [2] These algorithms also exhibit a "graceful" performance deterioration for small violations of the underlying assumptions. That is, if θ_* is outside the set $\mathcal{M}$, the performance deterioration is proportional to the distance between θ_* and $\mathcal{M}$.

Gradient+projection adaptation is suitable for time-varying parameters, as well. Using similar analysis steps with $\dot{\theta}_* = v$, it can be shown that a performance bound is

$$RMS(\epsilon) \leq RMS(n) + \left[2\frac{\text{diam}(\mathcal{M})}{\lambda_{min}(\Gamma)}MA(v)\right]^{1/2}$$

where λ_{min} denotes the minimum eigenvalue and $MA(v)$ is the mean-absolute value of the signal v (similar to RMS). This expression quantifies the intuitively expected result, that the RMS-error performance improves with faster adaptation and deteriorates with the size of the parametric uncertainty set. To prevent any misinterpretations, however, it should be emphasized that in the identification of dynamical systems faster adaptation does not always lead to better performance. The reason is that the "identification error" (error

[2] The upper bound is $\text{diam}\mathcal{M}\,\limsup|w|$ which is tight in the case of arbitrary w, see [Tsakalis].

between the identified and the actual system) contains a term which is proportional to the adaptation gain. Loosely speaking, this term arises because the derivative of the parameter estimates is not a good estimate of the derivative of the parameters. Here, filtering the parameter estimates may be an interesting option, though not well understood at present.

Among the undesirable properties of these algorithms is that large parametric uncertainty sets still allow for the appearence of large bursts (order of diam$\mathcal{M}$). Partial remedies of this problem include:

- The use of a combination of parameter projection and dead-zone modifications with a small threshold. In this case, the dead-zone serves the purpose to prevent parameter drifts due to small levels of noise. Since the dead-zone threshold is small, its use will not cause any significant performance deterioration. On the other hand, the parameter projection ensures a reasonable operation when the noise becomes unusually large or the parameters vary with time.

- Injection of a low-level excitation signal when the "confidence" in the parameter estimates drops. The difficult problem here is to determine the least level of excitation possible to overcome the noise. (Since such an excitation signal is artificial, it will produce a deterioration of the overall system performance.)

- On line estimation and reduction of the parametric uncertainty set using some auxiliary information, e.g., noise bounds. Some hard results on this can be found in [Tsakalis] but the ideas cannot be extended to the time-varying case.

- Use of multiple estimators (e.g., [Middleton]; still under investigation).

$\triangledown\triangledown$

2.3 Least-Squares

While parameter drift remains as a largely unresolved issue for the robustness of adaptive systems, speed of convergence is an important concern for their performance. Basic optimization theory suggests that the convergence of gradient schemes can become arbitrarily slow for badly conditioned Hessians.[3] In the parameter estimation framework, the Hessian is a summation of rank-one matrices ($ww^\top$). Therefore, its condition number depends on the "rotation" of the regressor vector through a sufficient number of independent directions. Unfortunately, and especially in the identification of dynamical systems, the regressor vector often oscillates between nearly co-linear directions. This results in a poorly conditioned Hessian (covariance matrix) and extremely slow convergence.

For such cases, least-squares (LS) algorithms (or quasi-Newton algorithms from numerical optimization) offer an attractive alternative by updating the adaptation gain Γ on line. This update aims to precondition the Hessian matrix and improve the speed of convergence. Transient improvements of the order of 100:1 over gradient algorithms are not uncommon! A different interpretation of the LS updates is that the algorithm performs a Gram-Schmidt-like orthogonalization of each observed regressor direction. Then, in directions that have appeared before it uses a small step size, emphasizing noise-filtering properties. In new directions, the algorithm applies a high gain to remove the error as quickly as possible.

A drawback of the LS algorithm is that the initial-phase convergence is not monotonic or near-monotonic (similar to quasi-Newton or conjugate-directions optimization). That is, the updating of the parameters is initially very "violent". This indicates that during that phase the estimator may be able to keep the error small but the model is not reliable as a predictor. The net result is that the design of least-squares algorithms involves a difficult selection of a number of parameters trading-off speed of adaptation versus noise susceptibility. (Again, there is no free lunch.)

[3]Recall that the condition number of a positive definite matrix is the ratio of the smallest to the largest eigenvalue. The convergence of gradient/steepest descent algorithms is exponential, following a function $(1-\kappa)^k$ where κ is the condition number and k is the number of iterations. For a Hessian with condition number 0.001 (not uncommon), it takes roughly 4500 iterations to reduce the error to 1% of its initial value.

The basic LS algorithm can be motivated as a special case of a Kalman filter (LQ-optimal observer). Consider the linear system

$$\begin{aligned} \dot{\theta}_* &= A\theta_* + v \\ y &= w^\top \theta_* + n \end{aligned}$$

Its Kalman observer is

$$\begin{aligned} \dot{\theta} &= A\theta + L(w^\top\theta - y) \\ L &= -\Gamma w Q_n^{-1} \\ \dot{\Gamma} &= A\Gamma + \Gamma A^\top - \Gamma w Q_n^{-1} w^\top \Gamma + Q_v \end{aligned}$$

where the Riccati is solved forward in time, starting with $\Gamma(0) \gg 1$. Γ here corresponds to an estimate of the inverse Hessian. Q_n and Q_v denote the "strength" of the output and state noise (i.e., $\int n^\top Q_n^{-1} n \leq 1$). For well-posedness, the Kalman observer requires the uniform observability of the pair $(A, w^\top)$. In the standard estimation problem $A = 0$, and the observability condition translates into the usual notion of PE.

A variation of this solution is the exponential weighting of past data. Letting $A = 0$ for simplicity, the exponentially weighted observer uses the following modified Riccati

$$\dot{\Gamma} = 2\delta\Gamma - \Gamma w Q_n^{-1} w^\top \Gamma + Q_v$$

2.9 Analysis: Under PE, Γ, Γ^{-1} are positive definite and bounded. Then $V = \tilde{\theta}^\top \Gamma^{-1} \tilde{\theta}$ is positive definite, radially unbounded and decrescent. Further, for scalar outputs (to simplify the expressions)

$$\begin{aligned} \dot{V} &= -2w^\top\tilde{\theta}\epsilon/Q_n - 2\tilde{\theta}^\top\Gamma^{-1}v - 2\delta\tilde{\theta}^\top\Gamma^{-1}\tilde{\theta} \\ &\quad +\tilde{\theta}^\top w w^\top \tilde{\theta}/Q_n - \tilde{\theta}^\top\Gamma^{-1}Q_v\Gamma^{-1}\tilde{\theta} \\ &\leq -\epsilon^2/Q_n + n^2/Q_n + v^\top Q_v^{-1} v - 2\delta V \\ \Rightarrow & \\ \|\epsilon\|_{2,\delta}^2 &\leq Q_n(V(0)e^{2\delta t_0} - V(t)e^{2\delta t}) + \|n\|_{2,\delta}^2 + Q_n\|v\|_{2,\delta;Q_v^{-1}}^2 \\ \Rightarrow & \\ V(t) &\leq V(0)e^{-2\delta(t-t_0)} + e^{-2\delta t}\left(\|n\|_{2,\delta}^2/Q_n + \|v\|_{2,\delta;Q_v^{-1}}^2\right) \end{aligned}$$

where $\|.\|_{2,\delta}$ denotes the exponentially weighted 2-norm. Now, the first term of the last inequality converges to zero for $t \to \infty$ while the second is bounded according to the noise and parameter variation assumptions. Hence V is uniformly ultimately bounded and so are the parameter estimates.

Next, regarding the performance of the algorithm, notice that for constant V the parameter error decreases as Γ decreases. Since Γ^{-1} is roughly $\int ww^\top$, the transient convergence depends on the speed of rotation of the regressor vector. Unfortunately, this statement cannot be simplified into a simple rate expression for arbitrary w. While several interesting expressions about the asymptotic properties of the algorithm are presented below, it should be kept in mind that most of the transient benefits of LS algorithms are due to the decrease in Γ.

The intermediate derivation of the estimation error bound offers an estimate for some additional performance properties of the LS algorithm.

- For $v = 0$ (LTI system)
$$\|\epsilon\|_{2,\delta}^2 \leq Q_n(V(0)e^{2\delta t_0} - V(t)e^{2\delta t}) + \|n\|_{2,\delta}^2$$
implying that the $L_{2,\delta}$ gain of the operator $n \mapsto \epsilon$ is less than or equal to unity.

- Estimates of the parametric uncertainty can be derived on-line. For example, suppose that the noise is bounded by n_0 and $v = 0$. Define
$$\dot{E} = -2\delta E + \epsilon^2/Q_n \; ; \quad E(0) = 0$$

. Then it follows from the previous analysis that

$$\begin{aligned} V(t) &\leq V(0)e^{-2\delta(t-t_0)} - E(t) + \int_{t_0}^{t} e^{-2\delta(t-\tau)} n^2/Q_n \, d\tau \\ &\leq V(0)e^{-2\delta(t-t_0)} - E(t) + n_0^2/(2\delta Q_n) \\ &\Rightarrow \\ \limsup V &\leq -\liminf E + n_0^2/(2\delta Q_n) \\ \limsup |\tilde{\theta}| &\leq \lambda_{max}(\Gamma) \limsup V \end{aligned}$$

Thus, an asymptotic bound on the parameter error (or reliability of the parameter estimates) can be obtained on-line, given a bound on the noise. If, in addition, an estimate of the initial parameter error is available, this bound can be modified to hold for all times.

Finally, under some additional assumptions (e.g., stochastic case, n, w uncorrelated, zero-mean) it turns out that in the limit V approaches zero.

$\triangledown\triangledown$

2.10 Modifications of the basic LS algorithm: One potential problem of the otherwise exceptional LS algorithms is that they may lead to singular matrices if the PE condition is violated. Periodic restarts of the algorithm provide a partial answer, though not completely satisfactory. Alternatively, the Riccati updates can be modified with some "small" perturbation terms to guard against singular matrices (effectively implementing a floor and a ceiling on Γ). An example of that is to add a term $-(\rho/Q_n)\Gamma^2$ representing a fictitious PE regressor with magnitude controlled by ρ. [4] Thus,

$$\begin{aligned} \dot{\Gamma} &= 2\delta\Gamma - \Gamma w Q_n^{-1} w^\top \Gamma - (\rho Q_n^{-1})\Gamma^2 + Q_v \\ \dot{\Gamma}^{-1} &= -2\delta\Gamma^{-1} + w Q_n^{-1} w^\top + (\rho Q_n^{-1}) I - \Gamma^{-1} Q_v \Gamma^{-1} \end{aligned}$$

Here, for simplicity, the speed of parameter variation Q_v may be set to zero since the exponential weighting will prevent Γ from approaching singular matrices. Although not equivalent, both of these terms convey the same meaning of "forgetting" past data. The exponential weighting simply states that old data become somehow invalid with a speed determined by δ. The parameter variation term provides a more quantified notion of how old data should be progressively discarded. Indeed, as the target parameters move, old I/O data offer less information about their value. The overall effect is still the same: in directions where no recent data have been obtained, the parameter uncertainty increases. This is equivalent to the interpretation of $\sqrt{\Gamma}$ as the generalized radius of the parametric uncertainty ellipsoid. With $Q_v = 0$, the following bounds on the eigenvalues of Γ^{-1} (or Γ) can be obtained:

$$\lambda_{max}\Gamma^{-1} \leq (\rho + \|w^\top w\|_\infty)/(2\delta Q_n) \;\; ; \;\; \lambda_{min}\Gamma^{-1} \geq \rho/(2\delta Q_n)$$

Using the same analysis tools as before, it follows that when $v = 0$ (LTI case), $RMS(\epsilon) \leq RMS(n)$. In the case of time-varying target parameters, the result is also similar, i.e.,

$$\int_{<T>} \epsilon^2 \leq Q_n V(0) + \int_{<T>} n^2 + \frac{\rho + \|w^\top w\|_\infty}{\delta} \mathrm{diam}\mathcal{M} \int_{<T>} |v|$$

expressed in its integral form. This bound seems a bit awkward in the sense that the mean-absolute variation of parameters appears instead of the RMS value. The reason for this is the choice $Q_v = 0$ which eliminated a quadratic term in $\Gamma^{-1}\tilde{\theta}$ which could be used to complete the squares with $2\tilde{\theta}^\top\Gamma^{-1}v$. This, however, would only be possible if w were PE. In such a case sharper and more "symmetric" bounds can be derived (try this!).

Nevertheless, even in the non-PE case the above bound has an intuitive interpretation: if the target parameter variation is slow then the RMS of the estimation error will be small. The bound also serves as a guide for the choice of the various adaptation parameters.

[4] If it is desired to bias the covariance Γ to certain directions, a term $-(\rho/Qn)\Gamma\Lambda\Gamma$ can be used.

- Small values of Q_n imply fast convergence of the estimation error by reducing the contribution of the initial parameter error.
- Large values of δ reduce the contribution of the perturbation caused by the target parameter variation. This corresponds to the fast forgetting of past data. Also, large values of δ increase the effective adaptation gain ($\lambda_{min}\Gamma/Q_n \geq 2\delta/(\rho + \|w^\top w\|_\infty)$; cmp. with the gradient case).
- Both of the above guidelines aim to reduce the estimation error. The precise selection of these parameters involves more complicated trade-offs between keeping ϵ small, $\dot{\theta}$ small and making the best use of past data so that $\tilde{\theta}$ is small. Recall that much of the model ability to predict future outputs depends on the size of $\dot{\theta}$. The interplay of all these factors is not well-understood, especially in the identification of linear dynamical systems. Nevertheless, regardless of the optimality of the selections, LS and modified-LS algorithms have the important ability to precondition the data (Hessian) and can improve the speed of convergence over gradient algorithms.

Finally, suitable projections and dead-zones can be developed for LS algorithms along the same lines as in the gradient case. An important consideration here is the effect that the projection has on the updates of the covariance or the Hessian matrix. Guided by the Lyapunov analysis, the simplest approach that ensures the non-increase of the parameter error is to disable the update of Γ when the projection is active. The drawback of this approach is that some directionality information is lost leading to possible performance deterioration if the parameter projection is active for extended time periods. The equations for the LS+projection algorithm are:

$$\begin{aligned} \dot{\theta} &= \mathcal{P}_{\mathcal{M}}(-\Gamma w Q_n^{-1}\epsilon) \\ \dot{\Gamma} &= \psi_\Gamma(\theta)\left[2\delta\Gamma - \Gamma(wQ_n^{-1}w^\top + \rho Q_n^{-1}I)\Gamma + Q_v\right] \\ \psi_\Gamma(\theta) &= \begin{cases} 1 & \text{if } \psi(\theta)\theta_\perp^\top z \leq 0 \\ 1-\psi(\theta) & \text{otherwise} \end{cases} \end{aligned}$$

with the rest of the definitions ($\mathcal{P}_{\mathcal{M}}, \psi, z, \theta_\perp$) being as in the gradient+projection case. $\triangledown\triangledown$

3 Discrete-time Algorithms

In a similar fashion, adaptive algorithms can be developed for adaptation in discrete time. At this point, it is fairly well established that most adaptation properties are common between discrete and continuous time. There are some minor differences in primarily technical issues, for example, dealing with existence of solutions of ODEs. In most part, the continuous-time analysis and derivations leads to simpler expressions. Still, discrete-time implementation is (and should be) more favored. Regarding the application of discrete-time algorithms in continuous time cases there are two approaches. One attempts to approximate the continuous time solution through the use of integrated quantities over the sampling interval (thus requiring multi-rate sampling). The other is simply to perform discrete updates for each sampled-data equation. Both approaches would lead to similar results provided that the sampling interval is small enough. More significant differences will appear in cases where the computational load is too severe and the parameter updates can only be performed at a fraction of the sampling rate. This would have little effect on the ability to track time-varying parameters (they must be slow relative to the time-scale of the states). The only problem is that sampling the regressor at a slow rate may miss some information-carrying directions. Having said that, in the rest of this section it is assumed that the discrete-time adaptation occurs at a fast enough rate so that these issues can be ignored.

3.1 Discrete fading-memory LS: The discrete-time equations for the evolution of the target parameters and the estimation error are given by

$$\begin{aligned} \theta_{*k+1} &= \theta_{*k} + v_k \\ \epsilon_k &= w_k^\top \tilde{\theta}_k - n_k \end{aligned}$$

In this setup, a fading memory LS algorithm takes the form

$$\begin{aligned}\theta_{k+1} &= \theta_k - \frac{\gamma P_k w_k \epsilon_k}{1+\gamma w_k^\top P_k w_k} \\ P_{k+1}^{-1} &= aP_k^{-1} + (1-a)\Lambda + a\gamma w_k w_k^\top\end{aligned}$$

where $a \in [0,1]$ is the forgetting factor, $\gamma \in (0,\infty)$ is the adaptation gain and $\Lambda = \Lambda^\top > 0$ is the "floor" for P^{-1}. In this form, the algorithm requires the inversion of the matrix P at every step. In the ordinary LS this is avoided through the use of the *matrix inversion lemma* which allows the translation of the covariance updates into updates of its inverse. Here, however, the use of the matrix E to prevent singularities in the non-PE case, does not allow a similar simple derivation. [5]

It should be mentioned that other variants of the LS algorithm can be found in the literature, providing notions of optimality under a PE condition. The presented algorithm should be viewed as a simple one that can offer significant performance improvement over gradient while maintaining some desirable robustness properties in the absence of PE. $\triangledown\triangledown$

3.2 Analysis: Let $V_k = \tilde\theta_k^\top P_{k-1}^{-1}\tilde\theta_k$ and $g_k = 1+\gamma w_k^\top P_k w_k$. Then

$$\begin{aligned}V_{k+1} &= \tilde\theta_k^\top P_k^{-1}\tilde\theta_k + \frac{\gamma^2\epsilon_k^2 w_k^\top P_k w_k}{g_k^2} + v_k^\top P_k^{-1} v_k \\ &\quad -\frac{2\gamma w_k^\top \tilde\theta_k \epsilon_k}{g_k} + \frac{2\gamma\epsilon_k w_k^\top v_k}{g_k} - 2\tilde\theta_k^\top P_k^{-1} v_k \\ &= V_k - (1-a)\tilde\theta_k^\top (P_{k-1}^{-1}-\Lambda)\tilde\theta_k + a\gamma(\tilde\theta_k^\top w_{k-1})^2 \\ &\quad +\frac{\gamma^2\epsilon_k^2 w_k^\top P_k w_k}{g_k^2} - \frac{2\gamma(\epsilon_k+n_k)\epsilon_k}{g_k} \\ &\quad +v_k^\top P_k^{-1} v_k + \frac{2\gamma\epsilon_k w_k^\top v_k}{g_k} - 2\tilde\theta_k^\top P_k^{-1} v_k\end{aligned}$$

Bring in $\quad \tilde\theta_k^\top w_{k-1} = \dfrac{\epsilon_{k-1}}{g_{k-1}} + n_{k-1} - v_{k-1}^\top w_{k-1}$

and $\quad \dfrac{\gamma^2\epsilon_k^2 w_k^\top P_k w_k}{g_k^2} - \dfrac{\gamma\epsilon_k^2}{g_k} = -\dfrac{\gamma\epsilon_k^2}{g_k^2}$

$$\begin{aligned}V_{k+1} &= V_k - (1-a)\tilde\theta_k^\top (P_{k-1}^{-1}-\Lambda)\tilde\theta_k \\ &\quad +a\gamma\frac{\epsilon_{k-1}^2}{g_{k-1}^2} + a\gamma n_{k-1}^2 + 2a\gamma\frac{\epsilon_{k-1}n_{k-1}}{g_{k-1}} \\ &\quad -\frac{\gamma\epsilon_k^2}{g_k^2} - \frac{\gamma\epsilon_k^2}{g_k} - \frac{2\gamma\epsilon_k n_k}{g_k} \\ &\quad +v_k^\top P_k^{-1} v_k + \frac{2\gamma\epsilon_k w_k^\top v_k}{g_k} - 2\tilde\theta_k^\top P_k^{-1} v_k \\ &\quad -\frac{2a\gamma\epsilon_{k-1} w_{k-1}^\top v_{k-1}}{g_{k-1}} - 2a\gamma n_{k-1} w_{k-1}^\top v_{k-1} + a\gamma(w_{k-1}^\top v_{k-1})^2\end{aligned}$$

In the ideal case ($n=0, v=0$), the addition of N consecutive differences yields

$$V[N+1] - V[1] \quad \le \quad -(1-a)\sum_k \tilde\theta_k^\top (P_{k-1}^{-1}-\Lambda)\tilde\theta_k - \gamma\sum_k \frac{\epsilon_k^2}{g_k} + IC$$

[5] Observe that $P_k^{-1} - \Lambda \ge 0$ and P_k^{-1} bounded $\Rightarrow$ P_k bounded. Also, starting the iteration with Λ as initial condition, $\|P_k^{-1}\| \le \|\Lambda\| + a\gamma\|w^\top w\|_\infty/(1-a)$; for different initial conditions this is a limsup bound.

where IC denotes a term due to initial conditions. It now follows that $V[N]$ is bounded and $\sum_k \epsilon_k^2/g_k \leq C/\gamma$ where C is a constant; hence $\epsilon_k^2/g_k \to 0$ as $k \to \infty$.

When perturbations are present, ($n \neq 0$ and/or $v \neq 0$) the derivations become more tedious. In the easier of the two, (TI systems, $v = 0$)

$$\begin{aligned} V_{k+1} &= V_k - (1-a)\tilde{\theta}_k^\top (P_{k-1}^{-1} - \Lambda)\tilde{\theta}_k \\ &\quad + a\gamma \frac{\epsilon_{k-1}^2}{g_{k-1}^2} + a\gamma n_{k-1}^2 + 2a\gamma \frac{\epsilon_{k-1} n_{k-1}}{g_{k-1}} \\ &\quad - \frac{\gamma \epsilon_k^2}{g_k^2} - \frac{\gamma \epsilon_k^2}{g_k} - \frac{2\gamma \epsilon_k n_k}{g_k} \end{aligned}$$

Adding up N consecutive differences and completing squares it follows that

$$\begin{aligned} V[N+1] - V[1] &= -(1-a)\sum_k \tilde{\theta}_k^\top (P_{k-1}^{-1} - \Lambda)\tilde{\theta}_k + IFC \\ &\quad -\gamma \sum_k \left[\frac{\epsilon_{k-1}}{g_{k-1}} - (1-a) n_{k-1} \right]^2 - \gamma \sum_k \frac{\epsilon_k^2}{g_k} + [a + (1-a)^2]\gamma \sum_k n_{k-1}^2 \end{aligned}$$

where IFC denotes initial and final terms left out of the summations; these terms are bounded provided that the parameter estimates are bounded, something that can be ensured by using a parameter projection. The last inequality establishes the familiar RMS performance of LS algorithms in a slightly different form:

$$RMS(\epsilon/\sqrt{1+\gamma w^\top P w}) \leq [a + (1-a)^2] RMS(n)$$

One difference here due to discrete LS adaptation is that the effect of the various parameters (γ, a) is not immediately apparent. While the normalized error RMS bound has a minimum at $a = 0.5$, the minimum RMS error bound occurs at $a \to 1$ when $P \to 0$ (pure LS). It should also be mentioned that this result depends on the grouping of terms as well as the special form of the update equations (several variants of fading memory LS exist). Simple bounding procedures may still convey the same qualitative message but the bounds are too loose. Bounds that are closer to the continuous-time case are obtained for gradient algorithms ($P_{k+1} = P_k$). Here, it can easily be shown that

$$RMS(\epsilon/\sqrt{g}) \leq 2RMS(n/\sqrt{g}) \quad \text{and} \quad RMS(\tilde{\theta}^\top w/\sqrt{g}) \leq RMS(n/\sqrt{g})$$

In the case where the target parameters are time-varying ($v \neq 0$) obtaining perfromance bounds becomes considerably more involved. Without performing all the tedious computations, it can be seen that after completions of squares the perturbations will consist of summations of terms of the following forms

$$vP^{-1}v, \quad \tilde{\theta}v, \quad vw, \quad n^2$$

Since v is assumed to be small at least in the mean (slowly varying parameters), the usual continuity result can be established. That is, for small perturbations, the RMS of the estimation error will be small. On the other hand, the effect of the selection of the various design parameters (adaptation gain, forgetting factor) is not as transparent as in the continuous-time case. Still, there are two immediate observations: For the $vP^{-1}v$ terms to be individually bounded, P^{-1} should be bounded and therefore, the forgetting factor a should be strictly less than one. Moreover, bounding the terms $\tilde{\theta}v$ would require an a priori bound on $\tilde{\theta}$. Without any PE assumptions, this can be achieved by using a parameter projection modification (hard or soft). Notice that the need for parameter projections exists even in the TI case where a bound on the RMS error performance was derived but without establishing the boundedness of the parameter estimates. $\triangledown\triangledown$

3.3 Parameter projections in discrete-time algorithms: The basic principle in designing a parameter projection algorithm for discrete-time adaptation is the same as in the continuous-time case.

That is, enforce the constraints without increasing the parameter error, measured by the Lyapunov-like function $\tilde{\theta}P^{-1}\tilde{\theta}$. While in general terms continuity of the updates remains as a concern to avoid "ringing" phenomena, there are no strict continuity requirements (i.e., no boundary layers etc.) On the other hand, discrete parameter projections are computationally more demanding than continuous vector-field projections. Both solve a minimum distance problem from a convex set; in the former the set can be arbitrary while in the latter the set is a simple half-plane defined by the tangent hyperplane at the point of projection.

A discrete LS+projection algorithm can be defined as follows.

$$\begin{aligned}\theta_{k+1} &= \mathcal{P}^d_{\mathcal{M}}\left[\theta_k - \frac{\gamma P_k w_k \epsilon_k}{1+\gamma w_k^\top P_k w_k}\right] \\ P^{-1}_{k+1} &= \begin{cases} P_k^{-1} & \text{if } \mathcal{P}^d_{\mathcal{M}} \text{ is active} \\ aP_k^{-1} + (1-a)\Lambda + a\gamma w_k w_k^\top & \text{otherwise} \end{cases}\end{aligned}$$

where $\mathcal{P}^d_{\mathcal{M}}$ is a (discrete) parameter projection such that $V_{k+1}^{Projected} \leq V_{k+1}^{Not-Projected}$. In other words, $\mathcal{P}^d_{\mathcal{M}}$ needs to solve a minimum distance problem with the distance being defined by the weighted norm $\sqrt{x^\top P_k^{-1} x}$ (oblique projection).

Notice that the need to modify P^{-1}_{k+1} arises from the appearence of the term $\tilde{\theta}^\top_{k+1}w_k$ in $V[k+2]-V_{k+1}$. If the projection is active at $_{k+1}$ then the crucial simplification of this term found before, is no longer valid. Setting $P^{-1}_{k+1} = P_k^{-1}$ at this point, is a simple choice that eliminates this term altogether. With this choice, during projections the algorithm acts roughly as a gradient algorithm. Of course, the use of projections is not without consequences. Depending on the case, it may slow down the speed of convergence considerably. A partial remedy is to project on the intersection of the sets $\mathcal{M}$ and $\{\theta||\epsilon(\theta)| \leq \epsilon^{NP} + tol\}$, where ϵ^{NP} is the a posteriori error computed with θ_{k+1} before projection and tol is a tolerance depending on the noise bound. While this strategy can be very effective when a bound on the noise is available, it requires elaborate practical implementation to cope with the practical possibility of the intersection being empty.

The computation of the projection $\mathcal{P}^d_{\mathcal{M}}$ can be simply defined as the solution of the following optimization problem

$$\begin{aligned}\mathcal{P}^d_{\mathcal{M}}[z] &= \arg\min |\theta - z|_{P^{-1}} \\ \text{s.t.} &\quad \theta \in \mathcal{M}\end{aligned}$$

where z is the $_{k+1}$ parameter estimate (before projection) and $P = P_k$. Notice that the convexity of $\mathcal{M}$ guarantees that the projected point satisfies the desired inequality $V_{k+1}^{Projected} \leq V_{k+1}^{Not-Projected}$. The general solution of this problem can be computed via convex optimization in a fairly efficient manner. Since convex optimization codes require a significant overhead, some simpler, special-case solutions are presented next. $\triangledown\triangledown$

3.4 Projections on half-spaces: A half-space is defined by an inequality $c\theta \leq b$. Using the transformation $\sqrt{P}\bar{\theta} = \theta$, the projection becomes

$$\begin{aligned}\mathcal{P}^d_{\mathcal{M}}[z] &= \sqrt{P}\arg\min |\bar{\theta} - \bar{z}| \\ \text{s.t.} &\quad \bar{c}\bar{\theta} \leq b\end{aligned}$$

where $\bar{c} = c\sqrt{P}$ and $\sqrt{P}\bar{z} = z$. That is, the problem has been translated into a projection onto a modified half-space but with the usual distance.

Next, let $\bar{z}_p$ be the projection of a $z \notin \mathcal{M}$, in the modified coordinates. By the projection theorem, $\bar{z} - \bar{z}_p \perp \partial\bar{\mathcal{M}}$ (the difference is orthogonal to the tangent hyperplane at the boundary) or, for some $\mu > 0$, $\bar{z} - \bar{z}_p = \mu\bar{c}$. And since $\bar{z}_p \in \partial\bar{\mathcal{M}}$, $\mu = (\bar{c}\bar{z} - b)/(\bar{c}\bar{c}^\top)$. Hence,

$$\bar{z}_p = \bar{z} + \frac{\bar{c}\bar{z} - b}{\bar{c}\bar{c}^\top}\bar{c}^\top$$

This result can now be translated back to the original coordinates so that the computation of $\sqrt{P}$ or its inverse is not required.

$$\mathcal{P}^d_{\mathcal{M}}[z] \;=\; \begin{cases} z & \text{if } cz \le b \\ \left(I - P\frac{c^\top c}{cPc^\top}\right) z + b\frac{Pc^\top}{cPc^\top} & \text{otherwise} \end{cases}$$

$\nabla\nabla$

3.5 Projections on Ellipsoids: An ellipsoid is defined by an inequality $(\theta-\theta_c)^\top R^{-1}(\theta-\theta_c) \le 1$. Again, with the transformation $\sqrt{P}\bar{\theta} = \theta$, the projection becomes

$$\begin{aligned} \mathcal{P}^d_{\mathcal{M}}[z] &= \sqrt{P}\arg\min|\bar{\theta}-\bar{z}| \\ \text{s.t.} \qquad & (\theta-\bar{\theta}_c)^\top \bar{R}^{-1}(\theta-\bar{\theta}_c) \le 1 \end{aligned}$$

where $\sqrt{P}\bar{\theta}_c = \theta_c$, $\sqrt{P}R^{-1}\sqrt{P} = \bar{R}^{-1}$. (Translation into an orthogonal projection onto a modified ellipsoid.)

Letting $\bar{z}_p$ denote the projection in the modified coordinates and using the projection theorem, $\bar{z}-\bar{z}_p = \mu\bar{R}^{-1}(\bar{z}_p-\bar{\theta}_c)$, for some $\mu>0$, where $\bar{R}^{-1}(\bar{z}_p-\bar{\theta}_c)$ is the normal to $\partial\bar{\mathcal{M}}$ evaluated at $\bar{z}_p$. Hence, the required projection is computed by solving the following set of nonlinear equations

$$\begin{aligned} \bar{z}_p &= (I+\mu\bar{R}^{-1})^{-1}(\bar{z}+\mu\bar{R}^{-1}\bar{\theta}_c) \\ 1 &= (\bar{z}_p-\bar{\theta}_c)^\top\bar{R}^{-1}(\bar{z}_p-\bar{\theta}_c) \end{aligned}$$

It should be mentioned that the above equation does have multiple solutions, making the reliability of standard solvers questionable at best. Since the computation of projections can occur very often inside the adaptation algorithm, a more reliable, fast solver needs to be developed for this special case. After some analysis of the properties of the solution, it can be shown that the following constrained Newton iteration has the desired properties.

Set $\quad \xi = \bar{z}-\bar{\theta}_c\,, \quad B = \sqrt{\xi^\top\bar{R}^{-1}\xi}$

Repeat until $|f| < tol$

$Z = (I+B\bar{R}^{-1})^{-1}$

$f = \xi^\top Z\bar{R}^{-1}Z\xi - 1$

$g = -2\xi^\top Z\bar{R}^{-1}Z\bar{R}^{-1}Z\xi$

$\delta = -f/g$

WHILE $B+\delta \le 0$

$\delta = \delta/2$

END

$B = B+\delta$

Compute projection

$\bar{z}_p = Z\xi + \bar{\theta}_c$

Notice that the above algorithm requires the computation of an inverse $(I+B\bar{R}^{-1})^{-1}$ at every step. A simpler, typically faster, alternative is to use a non-orthogonal projection that still preserves the property $V_{k+1}^{Projected} \le V_{k+1}^{Not-Projected}$. The idea of such a projection is as follows:

- Pick a point in $\bar{\mathcal{M}}$ (usually, $\bar{\theta}_c$).
- Connect the point with $\bar{z}$ and find the intersection of the segment with $\partial\bar{\mathcal{M}}$.
- Project $\bar{z}$ on the hyperplane, tangent to $\partial\bar{\mathcal{M}}$ at the intersection (it is a separating hyperplane and, consequently, satisfies $V_{k+1}^{Projected} \le V_{k+1}^{Not-Projected}$).

- Repeat until convergence (guaranteed!).

The computational procedure implementing this algorithm is given below.

$$
\begin{aligned}
\text{Set} \quad & \xi = \bar{z} \\
\text{Repeat until } |K-1| < tol & \\
& K = \frac{1}{\sqrt{(\xi-\bar{\theta}_c)^\top \bar{R}^{-1}(\xi-\bar{\theta}_c)}} \\
& \xi_\perp = \frac{\bar{R}^{-1}(\xi-\bar{\theta}_c)}{\sqrt{(\xi-\bar{\theta}_c)^\top \bar{R}^{-1}\bar{R}^{-1}(\xi-\bar{\theta}_c)}} \\
& \xi = \xi - (1-K)\xi_\perp \xi_\perp^\top (\xi-\bar{\theta}_c) \\
\text{Compute projection} & \\
& \bar{z}_p = \xi
\end{aligned}
$$

Finally, with both algorithms it is straightforward to re-write the equations in the original coordinates so that the computation of $\sqrt{P}$ or its inverse is avoided. $\triangledown\triangledown$

3.6 Projections on Intersections of Half-spaces and Ellipsoids: Orthogonal or oblique projections on arbitrary intersections of half-spaces and ellipsoids requires the solution of a convex programming problem. However, simpler projections that preserve the property $V_{k+1}^{Projected} \leq V_{k+1}^{Not-Projected}$ can be computed by projecting sequentially on each set until convergence within a tolerance. Notice that by repeating the same sequence of projections, e.g., $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n, \mathcal{M}_1, \mathcal{M}_2, \ldots$, the resulting projection is continuous in the projected point. The convergence of this procedure is guaranteed provided that the intersection of all sets has a non-empty interior. When this assumption cannot be validated a priori, the use of a general convex programming algorithm is required. $\triangledown\triangledown$

3.7 Selection of the adaptation parameters: In the previous sections, several adaptive algorithms were presented and analyzed. Their performance was invariably characterized in terms of the RMS value of the estimation error. While such a performance measure provides a good insight on the capabilities and limitations of the algorithm it is rarely the case that it is the only performance measure of importance. Instead, most of the applications of adaptive algorithms call for the adaptation of a model and attempt to use its extrapolation properties to predict the system behavior. Arguably, the quality of such extrapolations depends on both the estimation error and the speed of variation of the parameter estimates. As a consequence, maintaining a good overall performance of the adaptive system (e.g., controller, noise canceller) would require the successful trade-off between estimation error and parameter variation properties. Unfortunately, simple intuitive arguments do not always lead to valid conclusions. For example, it can be argued that if the algorithm converges fast in the ideal case, then it will also provide good performance if the target parameters vary slowly with time. While this argument is valid to some extend, it breaks down when the adaptation gain is very high. In such a case it turns out that the parameter adjustments become extremely "violent" to the point that the overall performance deteriorates.

On the other hand, one could attempt to develop more precise performance bounds for the overall objective of the adaptive system. This valiant effort would certainly expose the effective contribution of both estimation error and parameter variation. However, any quantitative assessment other than the limiting cases is plagued by increased complexity (due to the nonlinear nature of adaptation) and increased conservatism (several bounding procedures are usually necessary). Even though partial optimization may provide a good starting point without becoming intractable, this subject is still far from closed.

A different, less ambitious approach is to use simulations to gain insight on the effect of the various parameters. Having said that, it should be emphasized that simulations are no substitute for good analysis. Instead, detailed analytical results should be used as a guide to construct informative and representative simulations. A few guidelines on the selection of simulation examples are given next.

- Simulations should be representative of the actual application. In case of doubt, best- average- and worst- case scenaria should be used. The algorithm parameters should be chosen to optimize the average case, have good performance in the best case and be adequate for the worst case.
- Best case scenaria or speed of convergence in the ideal case are often bad indicators for the actual performance of the system.
- In emulating actual perturbations or time-variations, white-noise is often the least informative for identifying potential problems.
- Unmodeled dynamics and nonlinear effects can be emulated in a simulation. The results should be interpreted as envelopes rather than actual behavior.
- Design optimization makes sense only up to the point where the simulation errors become comparable with the expected noise level. In the same vein, extreme values of the adaptation parameters should be avoided (e.g., high gains). Notice, however, that gradient adaptation (fast forgetting) may be acceptable depending on the problem conditioning.
- Parameter constraints should be used to the maximum possible extend. A small dead-zone is also recommended but not larger than the typical noise level.
- Arguably, a good indicator for the choice of the adaptation gain is the speed of variation of the estimates (same for the covariance floor matrix in LS). Forgetting factors, on the other hand should reflect the speed of variation of the target parameters.
- Above all, the algorithm should make analytical sense, at least qualitatively. It is rare that an algorithm will be reliable when the analysis indicates potential robustness problems.

▽▽