Chapter 13
Acceleration and Successive Projection

The previous chapters have described and analysed many Iterative Control algorithms based on optimization principles. In the main, convergence was assured by monotonic reductions in the error norm combined with semi-quantitative rules for convergence rates based on experience of optimal control theory, the influence and choice of weight parameters and matrices (in norm definitions) and insights obtained using eigenvalue and/or frequency domain analysis. These rules introduce their own problems. Using, for example, NOILC Algorithm 9.1 as a model, rapid convergence is normally associated with low control weighting in the objective function. For state space systems, this leads to high gain state variable feedback and high gain feedback may not be advisable for the application considered. Therefore, despite the persuasive content of the rules, they provide trends rather than detail of likely outcomes and, following these trends to their natural conclusion can produce unacceptable control system characteristics.

The idea of creating rapidly convergent algorithms without undue need for undesirable control system characteristics (such as high gain controls), therefore, merits further research. Three approaches to the problem are presented, namely,

1. Saving on-plant-time and cost by off-line iterations using a plant model (Sect. 13.1).
2. Section 13.2, extends successive projection to improve convergence rates by adding in extrapolation (sometimes called over-relaxation) factors.
3. In Sect. 13.3, acceleration is achieved using successive projection and iteration dependent choice of parameterized sets $S_2(\sigma^2)$.

Both sections rely heavily on the material in Chaps. 9 and 12 as the underlying computations retain the NOILC computational framework at their core.

D.H. Owens, *Iterative Learning Control*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-6772-3_13

13.1 Replacing Plant Iterations by Off-Line Iterations

One simple acceleration mechanism is obtained by noting that *there is no need for every mathematical iteration to include experimental data collection*. In reality, only iterations that are implemented on the plant are normally counted by the user when costing the process in terms of time undertaking plant experimental work. Model-based, off-line iterations simply add to the computational burden but do not add to the cost. This idea applies to any iterative algorithm but, for NOILC Algorithm 9.1, details could be as follows,

1. implementation can take the form of partition of the iteration indices into two disjoint sets I_1 and I_2 and implementing the algorithm on the plant when $k \in I_1$ and using model-based computation when $k \in I_2$. The choices $I_1 = \{0, 2, 4, 6, 8, \ldots\}$ and $I_2 = \{1, 3, 5, 7, \ldots\}$ describes a situation where, following initialization with $k = 0$, the iteration is applied to the plant when k is even but is model-based (and off-line) when k is odd. Different partitions can be used if regarded as useful but, intuitively, the ratio of on-line to off-line iterations should not be too small if the plant is to have data input to the process.
2. Using the same notation, the off-line iterations could be based on different algorithms or simply use different parameters. For NOILC and state space systems, the control weighting ε^2 used when $k \in I_1$ could represent the need to have monotonic behaviours but avoid the use of on-line, high gain state feedback controls. For off-line iterations, the weight can be reduced to achieve faster error norm reduction. The reader will note the connection between this idea and the material in Sect. 11.4.
3. In both cases, the number of plant iterations required is reduced and acceleration is achieved. This is obvious for the first suggestion as, in the absence of modelling errors, the error update relationship $e_{k+1} = Le_k$ is unchanged. For the second, the change in on-line value ε^2 to a smaller, off-line value ε^2_{off} implies that errors are updated, alternately, as $e_{k+1} = L_1 e_k$ (off-line) and $e_{k+1} = L_0 e_k$ (on-line). The changing parameter is represented by the inequality $L_1 \leq L_0 \leq I$ and it follows that acceleration is achieved as $e_{k+2} = L_1 L_0 e_k$ for all $k \geq 0$ and $L_1 L_0 \leq L_0^2$.

There are many variations on the ideas expressed above, not all of which link easily to successive projection. The interested reader is invited to explore the possibilities.

13.2 Accelerating Algorithms Using Extrapolation

In this section, attention is focussed on successive projection algorithms with just $N_S = 2$ closed, convex sets $S_1 \subset H$ and $S_2 \subset H$ in a real Hilbert space H. In particular, it is assumed that S_1 is a linear variety. This assumption is natural as, typically, S_1 describes the dynamics of the plant in the absence of constraints.

13.2.1 Successive Projection and Extrapolation Algorithms

The following Algorithm defines the proposed extrapolation algorithm using the notation of Chap. 12. It covers the case of two closed, convex sets S_1 and S_2 where the set S_1 is a linear variety in a real Hilbert space H. The starting point is a point $s_0 \in S_2$ and successive projections are onto the sequence of sets $\{\tilde{S}_j\}_{j\geq 1}$ defined by the alternating sequence $\{S_1, S_2, S_1, S_2, S_1, \ldots\}$.

Algorithm 13.1 (*Successive Projection with Extrapolation Factors*) Suppose that $S_1 \cap S_2$ is non-empty. Then an accelerated successive projection algorithm for the construction of iterates that approach $S_1 \cap S_2$ arbitrarily closely is defined by choosing a starting point $s_0 \in S_2$ and constructing the sequence $\{s_j\}_{j\geq 0}$ with $s_j \in \tilde{S}_j, j \geq 1$, by solving, recursively, the optimization problems

$$\tilde{s}_{j+1} = \arg \min_{s\in\tilde{S}_{j+1}} \|s - s_j\|_H, \quad \textit{for } j \geq 0 \tag{13.1}$$

and defining new iterates s_{j+1} by the relations $s_1 = \tilde{s}_1$ and, more generally, for $j \geq 1$,

$$s_{j+1} = \begin{pmatrix} \tilde{s}_{j+1}, & \textit{if } \tilde{S}_{j+1} = S_2 \quad (j \textit{ odd}) \\ s_{j-1} + \lambda_{j+1}(\tilde{s}_{j+1} - s_{j-1}), & \textit{if } \tilde{S}_{j+1} = S_1 \quad (j \textit{ even}) \end{pmatrix} \tag{13.2}$$

where λ_{j+1} can be chosen to be any value in the range

$$1 \leq \lambda_{j+1} \leq \frac{\|s_j - s_{j-1}\|_H^2}{\|\tilde{s}_{j+1} - s_{j-1}\|_H^2}. \tag{13.3}$$

For all such sequences and for all choices of point $x \in S_1 \cap S_2$,

$$\begin{gathered} \|x - s_{2j-1}\|^2 \geq \|x - s_{2j+1}\|^2 + \underbrace{\lambda_{2j+1}}\, \|s_{2j} - s_{2j-1}\|_H^2, \quad \textit{for } j \geq 1, \\ (\textit{The Effect of the Extrapolation Factor } \lambda_{2j+1}) \end{gathered} \tag{13.4}$$

and

$$\|x - s_1\|_H^2 \geq \sum_{j=1}^{\infty} \lambda_{2j+1} \|s_{2j} - s_{2j-1}\|_H^2 \geq \sum_{j=1}^{\infty} \|s_{2j} - s_{2j-1}\|_H^2. \tag{13.5}$$

As a consequence,

$$\begin{gathered} \lim_{j\to\infty} \|s_{2j} - s_{2j-1}\|_H = 0, \quad \textit{so that} \quad \lim_{j\to\infty} \|s_{2j} - \tilde{s}_{2j+1}\|_H = 0 \\ \textit{and} \qquad \lim_{j\to\infty} \inf_{s\in S_2} \|s - s_{2j-1}\|_H = 0. \end{gathered} \tag{13.6}$$

That is,

1. the algorithm generates a sequence of iterates that, from Eq. (13.4), get closer to $S_1 \cap S_2$ and ultimately, Eq. (13.6), lie arbitrarily close to both S_1 and S_2. It is therefore a proximity algorithm.
2. If the user chooses the value $\lambda_{2j+1} = 1$ for all indices j, the algorithm is precisely NOILC Algorithm 9.1 as expressed in the form of Theorems 12.1 and 12.4.
3. Using a value $\lambda_{2j+1} > 1$ appears, from Eq. (13.4), to reduce the range within which $\|x - s_{2j+1}\|_H^2$ can sit as compared with that achieved by NOILC. This is the fact that supports the interpretation that using larger values of the extrapolation factor will tend to accelerate the iteration process. However,
4. in practice, small errors in $\tilde{s}_{2j+1}$ and s_{2j-1} could lead to larger errors in s_{2j+1}. That is, the algorithm is likely to be less robust than NOILC, particularly if λ_{2j+1} is large. In practice, therefore, it is advisable to limit it to a chosen maximum value $\lambda_{max} \geq 1$ even if larger values are allowed by the (error free) theory. That is, the range defined by Eq. (13.17) is replaced by the range

$$1 \leq \lambda_{j+1} \leq \min\{\frac{\|s_j - s_{j-1}\|_H^2}{\|\tilde{s}_{j+1} - s_{j-1}\|_H^2}, \lambda_{max}\}. \tag{13.7}$$

Note: The algorithm has very precise properties that are revealed very well by the general approach used. In the form described, the extrapolation is placed in the set S_1. The reader will note that, if both S_1 and S_2 are closed linear varieties, there will be two choices of set that can be used.

Proof of Algorithm Properties A demonstration that the algorithm is well-defined and has the stated properties now follows. The first step is to show that the value of λ_{j+1} is well defined. First assume that j is even so that $s_{j+1} \in \tilde{S}_{j+1} = S_1$. Next observe that, if the algorithm has not converged, $\|s_j - s_{j-1}\|_H \neq 0$ and it is then only necessary to show that $0 < \|\tilde{s}_{j+1} - s_{j-1}\|_H^2 \leq \|s_j - s_{j-1}\|_H^2$. If $\|\tilde{s}_{j+1} - s_{j-1}\|_H^2 = 0$ then $\tilde{s}_{j+1} = s_{j-1}$. Using the linear variety assumption for S_1, it follows that $(\tilde{s}_{j+1} - s_j) \perp (x - \tilde{s}_{j+1})$ for all $x \in S_1$. Using the fact that $\langle x - s_j, s_j - s_{j-1}\rangle_H \geq 0$ for all $x \in S_2$ then gives two expressions, satisfied for all $x \in S_1 \cap S_2$,

$$\langle x - s_j, s_j - s_{j-1}\rangle_H \geq 0 \quad \textit{and} \quad \langle x - s_{j-1}, s_{j-1} - s_j\rangle_H = 0 \tag{13.8}$$

so that $\|s_j - s_{j-1}\|_H^2 \leq 0$ which is impossible as $s_j \neq s_{j-1}$ by assumption. Therefore $\tilde{s}_{j+1} \neq s_{j-1}$ and the fact that $\lambda_{j+1} > 1$ then follows from the identity $\|s_j - s_{j-1}\|_H^2 = \|s_j - \tilde{s}_{j+1}\|_H^2 + \|\tilde{s}_{j+1} - s_{j-1}\|_H^2 > \|\tilde{s}_{j+1} - s_{j-1}\|_H^2$.
Next, let $x \in S_1 \cap S_2$ and consider, for $j \geq 1$,

$$\begin{aligned}\langle s_{2j+1} - s_{2j-1}, s_{2j-1} - x\rangle_H &= \lambda_{2j+1}\langle \tilde{s}_{2j+1} - s_{2j-1}, s_{2j-1} - x\rangle_H \\ &= \lambda_{2j+1}\langle \left(\tilde{s}_{2j+1} - s_{2j}\right) + s_{2j} - s_{2j-1}, s_{2j-1} - x\rangle_H \\ &= \lambda_{2j+1}\langle s_{2j} - s_{2j-1}, s_{2j-1} - x\rangle_H\end{aligned} \tag{13.9}$$

where the orthogonality of $\tilde{s}_{2j+1} - s_{2j}$ and $s_{2j-1} - x$ has been used. It follows that

$$\begin{aligned}\langle s_{2j+1} - s_{2j-1}, s_{2j-1} - x\rangle_H &= \lambda_{2j+1}\langle s_{2j} - s_{2j-1}, s_{2j-1} - s_{2j} + s_{2j} - x\rangle_H \\ &= -\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2 \\ &\quad + \lambda_{2j+1}\langle s_{2j} - s_{2j-1}, s_{2j} - x\rangle_H \\ &\leq -\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2.\end{aligned} \tag{13.10}$$

Writing

$$\begin{aligned}\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2 &= \lambda_{2j+1}\left(\frac{\|s_{2j}-s_{2j-1}\|_H^2}{\|\tilde{s}_{2j+1}-s_{2j-1}\|_H^2}\right)\|\tilde{s}_{2j+1} - s_{2j-1}\|_H^2 \\ &\geq \lambda_{2j+1}^2\|\tilde{s}_{2j+1} - s_{2j-1}\|_H^2 = \|s_{2j+1} - s_{2j-1}\|_H^2\end{aligned} \tag{13.11}$$

then gives

$$\begin{aligned}\|s_{2j+1} - x\|_H^2 &= \|s_{2j-1} - x\|_H^2 + \|s_{2j+1} - s_{2j-1}\|_H^2 + 2\langle s_{2j+1} - s_{2j-1}, s_{2j-1} - x\rangle_H \\ &\leq \|s_{2j-1} - x\|_H^2 + \|s_{2j+1} - s_{2j-1}\|_H^2 - 2\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2\end{aligned} \tag{13.12}$$

Re-arranging yields the required inequality (13.4) as

$$\begin{aligned}\|s_{2j-1} - x\|_H^2 &\geq \|s_{2j+1} - x\|_H^2 + \left(\lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2 - \|s_{2j+1} - s_{2j-1}\|_H^2\right) \\ &\quad + \lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2 \\ &\geq \|s_{2j+1} - x\|_H^2 + \lambda_{2j+1}\|s_{2j} - s_{2j-1}\|_H^2\end{aligned} \tag{13.13}$$

The remainder of the proof is an application of an induction argument and the condition $\lambda_{2j+1} \geq 1$. This yields Eq. (13.5) and hence Eq. (13.6). □

13.2.2 NOILC: Acceleration Using Extrapolation

Algorithm 13.1 combines the computations of NOILC Algorithm 9.1 (and all its subsequent variations) with a simple linear combination in S_1 parameterized by λ_{2j+1}. In structure it has some similarity to the use of relaxation methods but the fact that $\lambda_{2j+1} \geq 1$ links it more to extrapolation methodologies. The precise link with NOILC Algorithm 9.1 is obtained by generalizing Algorithm 12.3 to include extrapolation. The sets used are

$$S_1 = \{(e, u) : e = r - Gu - d\} \quad \textit{and} \quad S_2 = \{(e, u) : e = 0, u \in \Omega_u\}. \tag{13.14}$$

Projection of (e, u) onto S_2 is typically an off-line computation which, in the unconstrained case, yields $(0, u)$ so no computation is needed. The algorithm statement therefore can concentrate on the projection onto S_1 and takes the form described

below. Note that there is some change in the notation used to make Algorithm 13.1 match that used in NOILC studies in previous chapters.

Algorithm 13.2 (*NOILC with Extrapolation and Input Constraints*) Suppose that there exists an input $u \in \Omega_u$ that generates a zero tracking error. Then, using the notation and terminology of Algorithms 9.1 and 12.3, the Norm Optimal Iterative Learning Control algorithm (with extrapolation) generates a sequence of inputs $\{u_k\}_{k\geq 0}$ (and associated errors $\{e_k\}_{k\geq 0}$) by using the process,

STEP ONE (**Initialization**): Choose $u_0 \in \mathscr{U}$ and find the error response e_0 to generate the iterate $(e_0, u_0) \in S_1$. Then, for $k \geq 0$, undertake steps 2, 3, 4, 5 iteratively until the desired accuracy has been achieved.
STEP TWO (**Projection onto** S_2): Given the data $(e_k, u_k) \in S_1$, find the constrained input $u_k^{(1)}$ and hence the point $(0, u_k^{(1)}) \in S_2$ solving

$$u_k^{(1)} = \arg\min_{u\in\Omega_u} \|u - u_k\|^2_{\mathscr{U}}. \tag{13.15}$$

Note: For application to physical systems, it is important to ensure that plant response data is included in the next step. In the above, this requirement is included the construction of the plant error response $e_k^{(1)}$ to the input $u_k^{(1)}$.
STEP THREE: (**Projection onto** S_1) Project the data $(0, u_k^{(1)}) \in S_2$ onto S_1, using off-line or on-line calculations to find the minimum distance to S_1 expressed as the problem of finding the solution of the NOILC optimization problem

$$u_{k+1}^{(2)} = \arg\min_{u\in\mathscr{U}}\{J(u, u_k^{(1)}) : e = r - y, y = Gu + d\} \tag{13.16}$$

and the associated tracking error $e_{k+1}^{(2)}$. To ensure a link to plant data, the process should be driven by $e_k^{(1)}$ rather than $u_k^{(1)}$.
STEP FOUR (**Evaluation of an Extrapolation Factor**): Choose a value of extrapolation factor $\lambda(k+1) \geq 1$ in the range

$$1 \leq \lambda(k+1) \leq \frac{\|e_k\|^2_{\mathscr{Y}} + \varepsilon^2\|u_k^{(1)} - u_k\|^2_{\mathscr{U}}}{\|e_{k+1}^{(2)} - e_k\|^2_{\mathscr{Y}} + \varepsilon^2\|u_{k+1}^{(2)} - u_k\|^2_{\mathscr{U}}} \tag{13.17}$$

to produce the new control input

$$u_{k+1} = u_k + \lambda(k+1)\left(u_{k+1}^{(2)} - u_k\right). \tag{13.18}$$

STEP FIVE (**Error Measurement**): Compute the response e_{k+1} to u_{k+1} by, either using the off-line formula

$$e_{k+1} = e_k + \lambda(k+1)\left(e_{k+1}^{(2)} - e_k\right), \tag{13.19}$$

or using u_{k+1} on-line to find the measured tracking error. This step has then generated the data (e_{k+1}, u_{k+1}) and the procedure returns to STEP TWO.
The statement of and analysis following Algorithm 13.1 then proves that the tracking error converges to zero and the input signals becomes arbitrarily close to the constraint set Ω_u.

Algorithm 13.1 in the form of Algorithm 13.2 has potential value in accelerating NOILC algorithms without the need to use low input signal weighting in the objective function. This is true for linear, discrete or continuous, state space systems $S(A, B, C)$ where low control weights leads to high gain state feedback through the Riccati matrix solution of the optimization problem. For example, the inclusion of extrapolation in Algorithm reference 9.4 is a simple modification with the added calculation of $\lambda(k+1)$ and the extrapolation formulae for u_{k+1} and e_{k+1}. The signal $u_{k+1}^{(2)}$ is just, with a change in notation, the iterate u_{k+1} computed in Algorithm 9.4.

13.3 A Notch Algorithm Using Parameterized Sets

With the exception of the ideas of Iteration Management in Sect. 12.4, the discussion has, so far, assumed that the sets $S_1, S_2, S_3, \ldots, S_{N_S}$ are iteration independent. There is no mathematical reason why this should be the case but the choice of any variation in the definition of the sets will need detailed consideration and analysis. One set almost always describes the system dynamics. This section considers the choice of a companion set that is based on the idea of accelerating algorithm convergence by "annihilation" of part of the spectrum of GG^*. The notation of the NOILC Algorithm 9.1 is used for a system with dynamics $y = Gu + d$ and underlying spaces $\mathscr{Y}$ and $\mathscr{U}$, although the ideas apply more generally by suitable choice of G.

The system is required to track a reference $r \in \mathscr{Y}$ with tracking error $e = r - y$ equal to zero. Plant behaviours are associated with data points (e, u) in the product space $H = \mathscr{Y} \times \mathscr{U}$. H is a real Hilbert space with inner product

$$\langle (e, u), (w, v) \rangle_H = \langle e, w \rangle_{\mathscr{Y}} + \varepsilon^2 \langle u, v \rangle_{\mathscr{U}}, \quad \textit{where } \varepsilon^2 > 0. \tag{13.20}$$

13.3.1 Creating a Spectral Notch: Computation and Properties

The plant is identified with the set

$$S_1 = \{(e, u) : e = r - Gu - d\} \subset H, \quad (\textit{Plant Dynamics}). \tag{13.21}$$

The process considered here is that of creating a set $S_2 \subset H$, projecting a point $(e_0, u_0) \in S_1$ onto S_2 to create a point $(e_1^{(1)}, u_1^{(1)}) \in S_2$. This is then followed by

the projection of this point back onto S_1 to give a point $(e_1, u_1) \in S_1$. To be useful in practice, it is essential that e_1 has useful properties of monotonic norm reduction $\|e_1\|_{\mathscr{Y}} \leq \|e_0\|_{\mathscr{Y}}$ for all e_0. A stronger version of this requirement is that the norm reduction has the potential to be considerably greater than that achieved by NOILC Algorithm 9.1 for the given norm(s) and weight. The presentation takes the form of a constructive argument.

The set S_2 is parameterized by a single parameter $\sigma^2 > 0$ and denoted by

$$S_2(\sigma^2) = \{(\tilde{e}, u) : -\sigma^2\tilde{e} = r - Gu - d\} \subset H, \quad (\textit{Modified Dynamics}). \quad (13.22)$$

Writing $\tilde{e} = -\sigma^{-2}(r - Gu - d) = -\sigma^{-2}e$ identifies the signal $\tilde{e}$ as the error in tracking a signal $-\sigma^{-2}r$ with output defined by $-\sigma^{-2}Gu - \sigma^{-2}d$. That is, G is replaced by $-\sigma^{-2}G$ and d is replaced by $-\sigma^{-2}d$. If G is a state space model $S(A, B, C, D)$ with initial state $x(0) = x_0$, this change has two interpretations, namely that, either

1. the model $S(A, B, C, D)$ can be replaced by $S(A, B, -\sigma^{-2}C, -\sigma^{-2}D)$ with no change in x_0
2. or replaced by $S(A, -\sigma^{-2}B, C, D))$ if x_0 is replaced by $-\sigma^{-2}x_0$.

The first step is to suppose that $(e_0, u_0) \in S_1$ is given. The projection onto $S_2(\sigma^2)$ is then

$$\begin{array}{c}(e_1^{(1)}, u_1^{(1)}) = \arg\min_{(\tilde{e},\tilde{u})\in S_2(\sigma^2)} \left(\|\tilde{e} - e_0\|_{\mathscr{Y}}^2 + \varepsilon^2\|\tilde{u} - u_0\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad \tilde{e} = -\sigma^{-2}(r - G\tilde{u} - d).\end{array} \quad (13.23)$$

This is simply a NOILC problem but it has two important interpretations, the first of which is

1. application of NOILC Algorithm 9.1 for the modified dynamics with the reference signal replaced by $-\sigma^{-2}r - e_0$.
2. The second interpretation is, again, that of NOILC Algorithm 9.1. Using a scaling factor of $(-\sigma^2)^2$ on the objective function, the optimization can be written as

$$\begin{array}{c}((-\sigma^2)e_1^{(1)}, u_1^{(1)}) = \arg\min_{(e,u)\in S_2(\sigma^2)} \left(\|e + \sigma^2 e_0\|_{\mathscr{Y}}^2 + \sigma^4\varepsilon^2\|u - u_0\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad e = r - y \textit{ and } \quad y = Gu + d.\end{array} \quad (13.24)$$

This problem is precisely that of the application of NOILC to the original dynamics with reference replaced by $r + \sigma^2 e_0$ and weight parameter ε^2 replaced by $\sigma^4\varepsilon^2$. That is, the computation can be undertaken using the substitutions

$$\begin{array}{ll}\varepsilon^2 \mapsto \sigma^4\varepsilon^2 & (\textit{weight change}) \quad \textit{and} \\ r \mapsto r + \sigma^2 e_0 & (\textit{error adjusted reference})\end{array} \quad (13.25)$$

followed by scaling of the error resulting from $u_1^{(1)}$ by $-\sigma^{-2}$.

The two interpretations offer two alternative approaches to evaluation. The first is, in the form presented, off-line whilst the second could also be off-line but, in principle, could be on-line. This advantage in this case would be that the simple parameter change and modification to the reference is easily incorporated into the NOILC implementation. The first interpretation is used in the following analysis.

Projection onto S_2: The analysis of Chap. 9 indicates that

$$u_1^{(1)} = u_0 + \varepsilon^{-2}(-\sigma^{-2}G)^* \left(e_1^{(1)} - e_0\right) = u_0 - \sigma^{-2}\varepsilon^{-2}G^* \left(e_1^{(1)} - e_0\right). \quad (13.26)$$

The consequent "error" for the modified dynamics is then given by

$$\begin{aligned} e_1^{(1)} &= -\sigma^{-2}\left(r - Gu_1^{(1)} - d\right) \\ &= -\sigma^{-2}\left(e_0 + \sigma^{-2}\varepsilon^{-2}GG^*e_1^{(1)} - \sigma^{-2}\varepsilon^{-2}GG^*e_0\right) \\ \textit{so that} \quad e_1^{(1)} &= (\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)e_0. \end{aligned} \quad (13.27)$$

Projection onto S_1: Next compute the projection of $(e_1^{(1)}, u_1^{(1)}) \in S_2(\sigma^2)$ onto S_1 to give

$$\begin{aligned} (e_1, u_1) = \arg\min_{(e,u)\in S_1} &\left(\|e - e_1^{(1)}\|_{\mathscr{Y}}^2 + \varepsilon^2\|u - u_1^{(1)}\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad & e = r - Gu - d. \end{aligned} \quad (13.28)$$

This is a NOILC problem with r replaced by $r - e_1^{(1)}$ and hence $u_1 = u_1^{(1)} + \varepsilon^{-2}G^*\left(e_1 - e_1^{(1)}\right)$. Substituting as required then gives

$$\begin{aligned} e_1 = r - Gu_1 - d &= \left(r - Gu_1^{(1)} - d\right) - \varepsilon^{-2}GG^*\left(e_1 - e_1^{(1)}\right) \\ &= -\sigma^2 e_1^{(1)} - \varepsilon^{-2}GG^*\left(e_1 - e_1^{(1)}\right) \\ \textit{so that} \quad e_1 &= \left(I + \varepsilon^{-2}GG^*\right)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)e_1^{(1)}. \end{aligned} \quad (13.29)$$

Combining the two projections hence states that the resultant errors e_1 and e_0 in S_1 are related by the linear mapping $e_1 = L(\sigma^2)e_0$ where the operator $L(\sigma^2) : \mathscr{Y} \to \mathscr{Y}$ is bounded and has the form

$$L(\sigma^2) = \left(I + \varepsilon^{-2}GG^*\right)^{-1}(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2 \quad (13.30)$$

where the fact that the terms commute has been used to simplify the expression.

Interesting observations about the form of $L(\sigma^2)$ include the following,

1. the operator has the form of the familiar operator $(I + \varepsilon^{-2}GG^*)^{-1}$ that describes the error evolution for NOILC Algorithm 9.1 modified by the multiplicative factor $\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}$.

2. The factor $(\varepsilon^{-2}GG^* - \sigma^2 I)$ indicates that $e_1 = 0$ if $\varepsilon^{-2}GG^* e_0 = \sigma^2 e_0$. That is, if $\varepsilon^2\sigma^2$ is an eigenvalue of GG^* and e_0 is an associated eigenvector.

The second observation indicates one possible effect of the combined projection, namely that for some initial errors e_0, the algorithm "annihilates" the signal to produce a consequent error $e_1 = 0$. A more general statement of this property is as follows,

Theorem 13.1 (Approximate Annihilation Properties of $L(\sigma^2)$) *Using the notation and definitions of the preceding discussion, the operator $L(\sigma^2) : \mathscr{Y} \to \mathscr{Y}$ is self-adjoint and has the property that*

$$\begin{gathered} \gamma_1^2 \left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2 \le L(\sigma^2) \le \gamma_2^2 \left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2 \\ \text{with} \\ 0 < \gamma_1^2 = \frac{1}{(1+\varepsilon^{-2}\|G^*\|^2)(\sigma^4+\varepsilon^{-2}\|G^*\|^2)} \le \gamma_2^2 = \sigma^{-4}. \end{gathered} \tag{13.31}$$

As a consequence, $e_1 = 0$ if e_0 is an eigenvector of GG^* with eigenvalue $\varepsilon^2\sigma^2$ and, more generally, e_1 is arbitrarily small (relative to e_0) if e_0 has the property that $\|(\varepsilon^{-2}GG^* - \sigma^2 I)e_0\|_{\mathscr{Y}} \ll \|e_0\|_{\mathscr{Y}}$.

Proof $L(\sigma^2)$ is self-adjoint as it is a function of the self-adjoint operator GG^*. The inequality for $L(\sigma^2)$ follows from the properties seen in Theorem 9.1 applied to both $(I + \varepsilon^{-2}GG^*)^{-1}$ and $(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1} = \sigma^{-4}(I + \sigma^{-4}\varepsilon^{-2}GG^*)^{-1}$. The eigenvector property then follows easily by writing the inequality in the form

$$\gamma_1^4 \left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^4 \le L^2(\sigma^2) \le \gamma_2^4 \left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^4 \tag{13.32}$$

so that $\|e_1\|^2 = \|L(\sigma^2)e_0\|^2 \le \gamma_2^4 \|(\varepsilon^{-2}GG^* - \sigma^2 I)^2 e_0\|^2$. The result now follows easily. □

The operator also has more detailed properties expressed as follows,

Theorem 13.2 (Bounds on $L(\sigma^2)$ and Monotonicity)

$$\begin{gathered} 0 \le L(\sigma^2) \le I, \quad ker[I - L(\sigma^2)] = ker[G^*] \\ \text{and} \\ \overline{\mathscr{R}[I - L(\sigma^2)]} = \overline{\mathscr{R}[G]} = ker[G^*]^{\perp} \end{gathered} \tag{13.33}$$

so that, if $e_1 = L(\sigma^2)e_0$, then $\|e_1\|_{\mathscr{Y}} \le \|e_0\|_{\mathscr{Y}}$ for all $e_0 \in \mathscr{Y}$. In particular, $\|e_1\|_{\mathscr{Y}} < \|e_0\|_{\mathscr{Y}}$ for all e_0 that do not lie in $ker[G^*]$.

Proof The proof that $L(\sigma^2) \ge 0$ follows from Theorem 13.1 as $(\sigma^2 I - \varepsilon^{-2}GG^*)^2 \ge 0$. Next, for simplicity, write $X = \varepsilon^{-2}GG^* \ge 0$ so that, as required,

$$L(\sigma^2) = I - (I + X)^{-1}(\sigma^4 I + X)^{-1}\left(1 + \sigma^2\right)^2 X \le I. \tag{13.34}$$

The same formula shows that $ker[I - L(\sigma^2)] = ker[X] = ker[G^*]$, a fact that, together with $ker[G^*]^{\perp} = \overline{\mathscr{R}[G]}$, also proves the correspondence of the closure of the ranges. It follows that $\|e_1\| \leq \|e_0\|$ in all cases. Finally, noting that $I - L(\sigma^2)^2 = X_0(I + L(\sigma^2))X_0$ where X_0 is a positive definite, self-adjoint, square root of $I - L(\sigma^2) \geq 0$, any situation where $\|e_1\|^2 = \|e_0\|^2$ corresponds to the case where $0 = \langle e_0, (I - L^2(\sigma^2))e_0\rangle_{\mathscr{Y}} \geq \|X_0 e_0\|^2_{\mathscr{Y}}$. This implies that $e_0 \in ker[X_0]$. The proof is now complete as $ker[X_0] = ker[I - L(\sigma^2)] = ker[G^*]$. □

In general therefore, *the two step projection process has properties of both error norm reduction and annihilation*. The detailed form of the annihilation property in a particular situation depends on the choice of $\sigma^2 > 0$. At first sight, this appears to improve on the properties of NOILC Algorithm 9.1 but there is no guarantee, for an arbitrary choice of e_0 that the error norm reduction achieved is greater than that achieved using NOILC. A guarantee can be provided as follows,

Theorem 13.3 (Guaranteeing Improvements on NOILC) *With the assumptions of Theorem 13.2,*

$$L(\sigma^2) \leq (I + \varepsilon^{-2}GG^*)^{-1} \quad \textit{if} \quad \varepsilon^{-2}\|G^*\|^2 \leq 1 + 2\sigma^2. \tag{13.35}$$

This condition is satisfied for all $\sigma^2 > 0$ if $\varepsilon^{-2}\|G^*\|^2 \leq 1$, a condition that requires the weight ε^2 to be sufficiently large.

Note: If applied, the stated limitation imposed on the value of σ^2 ensures that the error norm resulting from the use of one iteration of NOILC from any initial error e_0 is greater than that achieved using the two step projection process considered in this section. If it is violated, the reader should be able to use an eigenvalue/eigenvector methodology to show that reductions will be achieved in all eigen-subspaces corresponding to eigenvalues of GG^* strictly less than $\varepsilon^2\left(1 + 2\sigma^2\right)$.

Proof Examination of $L(\sigma^2)$ indicates that it is only necessary to ensure that, with $X = \varepsilon^{-2}GG^*$, the operator $(\sigma^4 I + X)^{-1}(\sigma^2 I - X)^2 \leq I$. This is simply the condition $(\sigma^4 I + X) - (\sigma^2 I - X)^2 = X((1 + 2\sigma^2)I - X) \geq 0$ which is satisfied if $((1 + 2\sigma^2)I - X) \geq 0$. The result follows from the fact that $X \leq \|X\|I$ and the relation $\|X\| = \varepsilon^{-2}\|G^*\|^2$. □

An easily obtained insight into the underlying effects on the error is to note, using Theorem 13.2, that $L(\sigma^2)$ leaves errors in $ker[G^*]$ unchanged. In addition, it maps $\overline{\mathscr{R}[G]}$ into itself. Suppose, therefore that GG^* has strictly positive eigenvalues $\{\sigma_j^2\}_{j\geq 1}$ satisfying the order property $\|G^*\|^2 = \sigma_1^2 \geq \sigma_2^2 \geq \sigma_1^3 \geq \cdots$ and generating a complete set of orthonormal eigenvectors $\{v_j\}_{j\geq 1}$ spanning $\overline{\mathscr{R}[G]}$. Write $e_0 = \sum_{j\geq 1} \gamma_j v_j + e_0^{(2)}$ with $e_0^{(2)} \in ker[G^*]$ and suitable scalars $\{\gamma_j\}_{j\geq 1}$ satisfying $\sum_{j\geq 1} \gamma_j^2 < \infty$. Using the Spectral Mapping Theorem, the eigenvalues of $L(\sigma^2)$ take the values $\{f(\sigma_j^2, \sigma^2)\}_{j\geq 0}$ where

$$f(\mu, \sigma^2) = \frac{(\sigma^2 - \varepsilon^{-2}\mu)^2}{(1 + \varepsilon^{-2}\mu)(\sigma^4 + \varepsilon^{-2}\mu)} \tag{13.36}$$

and $0 \leq f(\mu, \sigma^2) \leq 1$ for all $\mu > 0$. In addition, it follows that

$$L(\sigma^2)e_0 = \sum_{j \geq 1} \underbrace{\gamma_j f(\sigma_j^2, \sigma^2)}\, v_j + e_0^{(2)}. \tag{13.37}$$

Note that,

1. each eigenvector v_j has its contribution to e_0 reduced by a factor of $f(\sigma_j^2, \sigma^2)$.
2. Choosing $\sigma^2 = \varepsilon^{-2}\mu$ and $\mu = \sigma_p^2$ then gives $f(\sigma_p^2, \sigma^2) = 0$ and the eigenvector v_p is eliminated from the resultant error e_1.
3. In a similar manner, all eigenvalues μ of GG^* that are close to $\varepsilon^2\sigma^2$ are "almost" eliminated from the resultant error e_1.

These properties are the motivation for the use of the word *"notch"* to describe the outcome of the projection process. Clearly the choice of σ^2 can have a benefit by eliminating, or almost eliminating, specific eigenvalue components of the error. This property is used in the next section as the basis of algorithm development and also used in Sect. 13.3.3 where the choice of σ^2 is related to approximate elimination of frequency components of the error.

As a final point in this section, the function $f(\mu, \sigma^2)$, on the interval $0 < \mu \leq \|G^*\|^2$ can be interpreted as a function shaping the spectrum of $L(\sigma^2)$ as a function of the spectrum of GG^*. It is useful to note that, even if the eigenvalues of GG^* are not known, $f(\mu, \sigma^2)$ offers the opportunity of assessing the effect on any eigenvalues at, or in the vicinity of, a chosen point μ. It is the factor by which the contributions of eigenvectors to the representation of e_0 are reduced by the two step "notch" process. The reader will be able to prove the following result using elementary algebra and calculus,

Theorem 13.4 (Properties of $f(\mu, \sigma^2)$ for $\mu \in [0, \infty)$) *Suppose that $\sigma^2 > 0$ is fixed. Then the continuous, differentiable function $f(\mu, \sigma^2)$ is positive for $\mu \in [0, \infty)$ and has the properties of being monotonically decreasing on the interval $[0, \varepsilon^2\sigma^2]$, monotonically increasing on $[\varepsilon^2\sigma^2, \infty)$ with a unique minimum at the point $\mu_0 = \varepsilon^2\sigma^2$. In addition,*

$$f(0, \sigma^2) = 1, \quad f(\varepsilon^2\sigma^2, \sigma^2) = 0, \quad \textit{and} \quad \lim_{\mu \to +\infty} f(\mu, \sigma^2) = 1. \tag{13.38}$$

The shaping of the function $f(\mu, \sigma^2)$ and its effect on error norm reduction is a design option. A possible approach to influencing the shape is

1. to choose the point μ_0 where f takes its minimum value of zero and hence to approximately annihilate the contribution of the spectrum of GG^* in the vicinity of this value.
2. As $\mu = \varepsilon^2\sigma^2$, choose ε^2 and σ^2 to satisfy the conditions of Theorem 13.3. This condition has the alternative form

$$\|G^*\|^2 \leq \varepsilon^2 + 2\mu_0. \tag{13.39}$$

This relationship provides a range of values for ε^2. In particular,

a. if a small value of $\varepsilon^2 \ll \|G^*\|^2$ is desired to ensure that the underlying NOILC algorithm produces a large error reduction, then μ_0 is bounded from below by a value close to $\frac{1}{2}\|G^*\|^2$. That is, annihilation can only be attempted for those parts of the spectrum of GG^* in the range $(\frac{1}{2}\|G^*\|^2, \|G^*\|^2]$.
b. Alternatively, if complete freedom to choose $\mu_0 > 0$ arbitrarily is preferred, ε^2 must be larger than $\|G^*\|^2$. That is, the price of this flexibility is that the underlying NOILC problem cannot reduce the error too much.

To illustrate the magnitude of the effects described, suppose that the value of ε^2 is guided by Theorem 13.3. For example, if the process is to improve on the norm reduction achieved by NOILC and eliminate any limitations on the choice of σ^2, choose $\varepsilon^2 = \|G^*\|^2$. The eigenvalues of $(I+\varepsilon^{-2}GG^*)^{-1}$ are then $\{(I+\varepsilon^{-2}\sigma_j^2)^{-1}\}_{j\geq 1}$ which lie on the curve $(1 + (\mu/\|G^*\|^2))^{-1}$, $0 \leq \mu \leq \|G^*\|^2$, a curve that reduces monotonically from the value of unity to the value 0.5 when $\mu = \|G^*\|^2$. The effect of the choice of σ^2 is illustrated in the following examples,

1. A notch at $\sigma^2 = 0.5$ then ensures that approximate annihilation of the eigenvalues σ_j^2 is achieved in the vicinity of $\mu = 0.5\|G^*\|^2$. Improvements on NOILC error reductions are achieved elsewhere. For example, examining the cases of $\mu = \|G^*\|^2$ and $\mu = 0.25\|G^*\|^2$ gives $f(\|G^*\|^2, 0.5) = f(0.25\|G^*\|^2, 0.5) = 0.1$. Using the monotonicity properties of f gives $0 \leq f(\mu, 0.5) \leq 0.1$ on the interval $0.25\|G^*\|^2 \leq \mu \leq \|G^*\|^2$. The corresponding range of reductions for NOILC is [0.5, 0.8] in that same eigenvalue range. The conclusion is that, when compared with NOILC, a substantial norm reduction is achieved over a wide range of eigenvalue values.
2. A notch at $\sigma^2 = 1$ gives values $f(\|G^*\|^2, 1) = 0$, $f(0.5\|G^*\|^2, 1) = 1/9$ and $f(0.25\|G^*\|^2, 1) = 9/25$. This illustrates, again, the improvement on NOILC. It also shows that the range of eigenvalues with substantial reductions in magnitude depends on the choice of σ^2.
3. A notch at $\sigma^2 = 2$ is too large to provide annihilation properties on any eigenvalue but error reduction benefits are still seen as illustrated by the computed values $f(\|G^*\|^2, 2) = 0.1$, $f(0.5\|G^*\|^2, 2) = 1/3$ and $f(0.25\|G^*\|^2, 2) = 49/85$.

13.3.2 The Notch Algorithm and Iterative Control Using Successive Projection

The two step process and described and the properties presented in the previous Sect. 13.3.1 are central to what follows. First note that the computations can be continued using the pair $(e_1, u_1) \in S_1$ to create an algorithm. That is, the computation can be repeated from this data point to create $(e_2^{(1)}, u_2^{(1)}) \in S_2$ and, from this, the data $(e_2, u_2) \in S_1$. However, the value of σ^2 used in this second application need not be

that used in the first! With this in mind, an Iterative Algorithm based on the concept of introducing an iteration dependent notch can be described as given below.

Algorithm 13.3 (*A Notch Algorithm with Iteration Dependent Notch*) Using the notation of Sect. 13.3.1, suppose that $r \in \mathscr{Y}$, $\varepsilon^2 > 0$ and that the iterative process is initiated by the choice of an input $u_0 \in \mathscr{U}$ that produces the data $(e_0, u_0) \in H = \mathscr{Y} \times \mathscr{U}$ from the plant. Then, an Iterative *Notch Algorithm* is defined by the process, for $k \geq 0$, of sequentially/iteratively using data (e_k, u_k) in the three step process defined by,

STEP ONE: Choose a value $\sigma^2 = \tilde{\sigma}_{k+1}^2 > 0$ to create a notch in the desired part of the spectrum of $L(\tilde{\sigma}_{k+1}^2)$.

STEP TWO: Use off-line computations to find the input $u_{k+1}^{(1)}$ and associated tracking error $e_{k+1}^{(1)}$ that solves the optimization problem

$$\begin{aligned}(e_{k+1}^{(1)}, u_{k+1}^{(1)}) = \arg\min_{(\tilde{e},\tilde{u})\in S_2(\tilde{\sigma}_{k+1}^2)} & \left(\|\tilde{e} - e_k\|_{\mathscr{Y}}^2 + \varepsilon^2 \|\tilde{u} - u_k\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad & \tilde{e} = -\tilde{\sigma}_{k+1}^{-2}(r - G\tilde{u} - d).\end{aligned} \tag{13.40}$$

Note: This is the projection of (e_k, u_k) onto $S(\tilde{\sigma}_{k+1}^2)$ and is, simply, either

1. one step of the NOILC Algorithm 9.1 where the data (r, G, d) for the original plant is replaced by $(-\tilde{\sigma}_{k+1}^{-2} r - e_k, -\tilde{\sigma}_{k+1}^{-2} G, -\tilde{\sigma}_{k+1}^{-2} d)$ for the modified dynamics.
2. Alternatively, it is one step of NOILC for the original plant but with reference replaced by $r + \tilde{\sigma}_{k+1}^2 e_k$ and ε^2 replaced by $\tilde{\sigma}_{k+1}^4 \varepsilon^2$. In this case, the computations might be possible on-line.

STEP THREE: Use on-line or off-line computations to construct the data (e_{k+1}, u_{k+1}) (for use in the next iteration) as the solution of the optimization problem

$$\begin{aligned}(e_{k+1}, u_{k+1}) = \arg\min_{(e,u)\in S_1} & \left(\|e - e_{k+1}^{(1)}\|_{\mathscr{Y}}^2 + \varepsilon^2 \|u - u_{k+1}^{(1)}\|_{\mathscr{U}}^2\right) \\ \textit{subject to the constraints} \quad & e = r - Gu - d.\end{aligned} \tag{13.41}$$

Note: This is the projection of $(e_{k+1}^{(1)}, u_{k+1}^{(1)})$ onto S_1. It is one step of the NOILC Algorithm 9.1 for the original plant using the minimization of $J(u, u_{k+1}^{(1)})$ but where the reference signal r is replaced by $r - e_{k+1}^{(1)}$.

The algorithm produces an error evolution expressed as

$$e_{k+1} = L(\tilde{\sigma}_{k+1}^2) e_k, \quad \textit{so that} \quad e_k = \prod_{j=1}^{k} L(\tilde{\sigma}_j^2) e_0 \quad \textit{for } k \geq 1. \tag{13.42}$$

In particular, using Theorem 13.2, the algorithm generates a monotonically decreasing error sequence satisfying

$$\|e_{k+1}\|_{\mathscr{Y}} \leq \|e_k\|_{\mathscr{Y}} \quad \textit{for } k \geq 0 \quad \textit{with} \tag{13.43}$$

strict inequality holding if $e_k \neq 0$ does not lie in $ker[G^*]$. If $e_0 \in ker[G^*]$, then $e_k = e_0$ for all $k \geq 0$.

The detailed properties of the algorithm can be deduced in several ways. The simplest convergence result is constructed using the annihilation properties of the basic notch process.

Theorem 13.5 (Sequential Eigenvector Annihilation/Finite Convergence) *Suppose that GG^* has strictly positive eigenvalues $\{\sigma_j^2\}_{j\geq 1}$ satisfying the order property $\|G^*\|^2 = \sigma_1^2 \geq \sigma_2^2 \geq \sigma_1^3 \geq \cdots$ and generating a complete set of orthonormal eigenvectors $\{v_j\}_{j\geq 1}$ spanning $\overline{\mathscr{R}[G]}$. Then, every initial error e_0 can be written in the form $e_0 = \sum_{j\geq 1} \gamma_j v_j + e_0^{(2)}$ with $\sum_{j\geq 0} \gamma_j^2 < \infty$ and the term $e_0^{(2)} \in ker[G^*]$ identified as the orthogonal projection $P_{ker[G^*]}e_0$ of e_0 onto $ker[G^*]$. The Notch Algorithm 13.3 then produces the tracking errors, for $k \geq 1$,*

$$e_k = \prod_{p=1}^{k} L(\tilde{\sigma}_p^2)e_0 = \sum_{j\geq 1} \gamma_j \left(\prod_{p=1}^{k} f(\sigma_j^2, \tilde{\sigma}_p^2) \right) v_j + P_{ker[G^*]}e_0. \tag{13.44}$$

Let the indices $\{j_k\}_{k\geq 1}$ be a re-ordering of the non-zero eigenvalues of GG^* and set $\tilde{\sigma}_k^2 = \varepsilon^{-2}\sigma_{j_k}^2$. Then, Notch Algorithm 13.3

1. *generates a monotonically reducing sequence of error norms and convergent errors satisfying*

$$\lim_{k\to\infty} e_k = P_{ker[G^*]}e_0. \tag{13.45}$$

2. *If, in addition, $\mathscr{Y}$ is finite dimensional, then, this limit is achieved in a finite number of iterations.*

Note: The finite convergence theoretically possible in this case is a parallel to Algorithm 7.4 presented in Sect. 7.2.2. Discrete state space systems satisfy the finite dimensionality assumption as does the Intermediate Point control problem (Sect. 10.5).

Note: The formula for e_k provides useful insight into the choices of $\tilde{\sigma}_j^2$ and underlines the rapid convergence that is possible if they are selected carefully.

Proof Every initial error e_0 can be written in the form $e_0 = \sum_{j\geq 1} \gamma_j v_j + e_0^{(2)}$ with the term $e_0^{(2)} \in ker[G^*]$ identified as the orthogonal projection of e_0 onto $ker[G^*]$. The formula for e_k then follows from the eigen-properties $L(\sigma_j^2, \sigma^2)v_j = f(\sigma_j^2, \sigma^2)v_j, j \geq 1$. Let $\delta > 0$ be arbitrary. As $\sum_{j\geq 1} \gamma_j^2 < \infty$, it is possible to choose an integer $N_1(\delta)$ such that $\sum_{j>N_1(\delta)} \gamma_j^2 < \delta$. In addition, for all sufficiently large integers $k \geq N_2(\delta)$ (say), the values $\varepsilon^{-2}\sigma_j^2$, $1 \leq j \leq N_1(\delta)$ have been used and the contribution of the associated eigenvectors to the error has been annihilated. The remaining error has norm $\|e_k\|_{\mathscr{Y}} \leq \delta$ for all $k \geq N_2(\delta)$. That is, $\limsup_{k\to\infty} \|e_k - e_0^{(2)}\|_{\mathscr{Y}} \leq \delta$ and convergence to zero follows as δ can be arbitrarily small. This process is finite if $\mathscr{Y}$ is finite dimensional. □

In practice, of course, the eigenvalues of GG^* will not be known and hence the exact annihilation property cannot be used. This does not prevent the successful application of Algorithm 13.3 as indicated by the following result that uses the information provided by Theorem 13.3.

Theorem 13.6 (Convergence Using a Limited Bandwidth) *Using the notation of Algorithm 13.3, suppose that, following N_0 iterations of unconstrained choice, the remaining values of $\tilde{\sigma}_{k+1}^2$ satisfy the condition*

$$1 + 2\tilde{\sigma}_{k+1}^2 \geq \varepsilon^{-2}\|G^*\|^2 \quad \textit{for all} \quad k \geq N_0 \ (\textit{say}). \tag{13.46}$$

Then, Notch Algorithm 13.3 generates a monotonically reducing sequence of error norms satisfying $\lim_{k\to\infty} e_k = P_{ker[G^*]}e_0$. In particular,

1. *if $N_0 = 0$, the error norm sequence reduces at a rate faster than that of NOILC Algorithm 9.1.*
2. *More generally, the algorithm converges faster than NOILC from the initial error e_{N_0}.*

Proof First note that the projection $P_{ker[G^*]}e_k = P_{ker[G^*]}e_0$ and, using Theorem 13.2, the error norm sequence is monotonically reducing for all k. It is possible to assume, without loss of generality, that $N_0 = 0$. Using Theorem 13.3 then gives $L(\tilde{\sigma}_{k+1}^2) \leq (I + \varepsilon^{-2}GG^*)^{-1}$ so that

$$\prod_{j=1}^{k} L(\tilde{\sigma}_j^2) \leq (I + \varepsilon^{-2}GG^*)^{-k} \tag{13.47}$$

Convergence properties of the Notch Algorithm then follow from those of Algorithm 9.1. □

The result allows situations where choices for the first iterations are flexible but, asymptotically, it requires emphasis on the interval $1 + 2\sigma^2 \geq \varepsilon^{-2}\|G^*\|^2$ which, if $\varepsilon^{-2}\|G^*\|^2 > 1$ does not permit "small" values of σ^2 to be used. This will limit the options available to influence convergence rates as part of the spectrum cannot be annihilated. In applications, two strategies will be required namely one to choose the first N_0 values, the second stage being the systematic selection of $\tilde{\sigma}_{k+1}^2$ satisfying $1 + 2\tilde{\sigma}_{k+1}^2 \geq \varepsilon^{-2}\|G^*\|^2$. For example

1. If $N_0 = 0$, the first stage is not needed. If $N_0 > 0$, a selection of values that covers a wide range of the spectrum is covered by the equally (linearly or logarithmically) spaced points. For example,

$$\sigma^2 = \varepsilon^{-2}\|G^*\|^2 \left(\frac{j}{N_0}\right), 1 \leq j \leq N_0. \tag{13.48}$$

The reader should note that, if N_0 is sufficiently large, the notch properties suggest that the contribution to e_{N_0} of eigenvalues in, for example, the range $N_0^{-1}\|G^*\|^2 \leq \mu \leq \|G^*\|^2$ will have been greatly reduced.

2. For $k \geq N_0$, there may be many more iterations to do. Although it is permitted to use values of $\sigma^2 > \varepsilon^{-2}\|G^*\|^2$, a focus on the interval of choice $[0.5(\varepsilon^{-2}\|G^*\|^2 - 1), \varepsilon^{-2}\|G^*\|^2]$ still leaves an infinity of choices including, for example,
   a. choosing an iteration independent value $\tilde{\sigma}_{k+1}^2 = \sigma^2$ in that range,
   b. choices randomly generated by a uniform pseudo-random number generator,
   c. choices selected from N_1 equally spaced points in the interval, or
   d. choices guided by "expert" knowledge of the plant and its observed behaviour (see Sect. 13.3.3 and the proposed use of frequency domain criteria).

Finally, with an iteration independent notch and a reference that can be tracked by some input signal, it is left as an exercise for the reader to prove the existence of a limit for the input sequence in the norm topology and, in addition, that the increased convergence rate does not influence the nature of this limit which is exactly the minimum energy/minimum norm solution obtained using NOILC without the notch modification.

13.3.3 A Notch Algorithm for Discrete State Space Systems

Notch Algorithm 13.3 applies quite generally and, being based on NOILC Algorithm 9.1, computations can be applied in any situation where NOILC, and its derivative algorithms, can be applied. It is expected that the most commonly used model in many applications is that of a state space model $S(A, B, C)$ with a specified initial condition $x(0) = x_0$. In this section, m-output, ℓ-input, linear, discrete state space models, operating on an interval $0 \leq t \leq N$, are considered. Algorithm 13.3 then has a very specific realization.

13.3.3.1 Algorithm Statement

More precisely, if $Q(t), 0 \leq t \leq N$ and $R(t), 0 \leq t \leq N$ are symmetric, positive definite weighting matrices,

Algorithm 13.4 (*A Notch Algorithm for Discrete State Space Systems*) Using the notation of Algorithm 13.3, suppose that the reference r has been specified and the iterative process initiated by the choice of u_0 that produces the data (e_0, u_0) from the plant. Then, an Iterative Notch Algorithm for the linear, discrete, state space system $S(A, B, C)$ is defined by the process, for $k \geq 0$, of sequentially using data (e_k, u_k) in the three step process,

STEP ONE: Choose a value $\sigma^2 = \tilde{\sigma}_{k+1}^2 > 0$ to create the desired notch for iteration $k + 1$.

STEP TWO: Use off-line computations to find the input time series $u_{k+1}^{(1)}$ and associated tracking error $e_{k+1}^{(1)}$ that minimizes the objective function

$$\begin{array}{c}\sum_{j=0}^{N}\ \left((\tilde{e}(t)-e_k(t))^T Q(t)(\tilde{e}(t)-e_k(t))+\varepsilon^2(\tilde{u}(t)-u_k(t))^T R(t)(\tilde{u}(t)-u_k(t))\right)\\ \textit{subject to the dynamic constraints}\quad -\tilde{\sigma}_{k+1}^2\tilde{e}=r-G\tilde{u}-d.\end{array} \tag{13.49}$$

For state space computations, the constraints can be written in the form $\tilde{e}(t) = -\tilde{\sigma}_{k+1}^{-2} r(t) - \tilde{y}(t)$, $0 \le t \le N$, where $\tilde{y}(t)$, $0 \le t \le N$, is the response of the system $S(A, B, -\tilde{\sigma}_{k+1}^{-2} C)$ to $\tilde{u}(t)$ from the initial condition $-\tilde{\sigma}_{k+1}^{-2} x_0$.

Note: As noted in Algorithm 13.3 there are two possible ways to approach this problem.

1. The approach described above is one step of the NOILC Algorithm 9.1 (perhaps in the form of Algorithm 9.4) from the input data u_k but where the data (r, C, x_0) for the original plant are replaced by $(-\tilde{\sigma}_{k+1}^{-2} r - e_k, -\tilde{\sigma}_{k+1}^{-2} C, x_0)$. This data change will, for example, change the form of both the Riccati equation and the equation for the predictive term ζ_{k+1} used in the computations.
2. The second approach uses the NOILC Algorithm 9.1 for the original plant model and initial condition, using the reference $r(t) + \tilde{\sigma}_{k+1}^2 e_k(t)$ and replacing ε^2 by $\tilde{\sigma}_{k+1}^4 \varepsilon^2$.

STEP THREE: Use on-line or off-line computations to construct the data (e_{k+1}, u_{k+1}) (for use in the next iteration) as the solution that minimizes the objective function

$$\begin{array}{l}\sum_{j=0}^{N}\ ((e(t)-e_{k+1}^{(1)}(t))^T Q(t)(e(t)-e_{k+1}^{(1)}(t))\\ \qquad\qquad +\varepsilon^2(u(t)-u_{k+1}^{(1)}(t))^T R(t)(u(t)-u_{k+1}^{(1)}(t)))\end{array} \tag{13.50}$$

subject to the constraints that $e(t) = r(t) - y(t)$, $0 \le t \le N$, where $y(t)$, $0 \le t \le N$, is the response of the system $S(A, B, C)$ to $u(t)$ from the initial condition x_0.

Note: This is one step of the NOILC Algorithm 9.1 from the data $u_{k+1}^{(1)}$ for the original plant but where the reference signal r is replaced by $r - e_{k+1}^{(1)}$. This data change requires only one Riccati equation but the equation for the predictive term ζ_{k+1} is driven by the input $u^{(1)}(t)$ and the "error corrected reference" $r - e_{k+1}^{(1)}$.

The algorithm has the monotonicity and convergence properties described by Algorithm 13.3 and illustrated by Theorems 13.5 and 13.6 and other material in Sect. 13.3.1 and 13.3.

The implementation of the algorithm is very similar to that of Norm Optimal Control and an understanding of that algorithm is essential. Together with Q and R, the choice of $\tilde{\sigma}_{k+1}^2$ is central to the prediction and achievement of significantly improved convergence rates.

13.3.3.2 Notch Values and the Frequency Domain

In what follows, it is seen that the choice of $\tilde{\sigma}^2_{k+1}$ can be supported by the frequency domain properties of the signal e_k. More precisely, suppose that the system $S(A, B, C)$ is asymptotically stable and that the matrices $\{Q(t)\}_{0\le t\le N}$ and $\{R(t)\}_{0\le t\le N}$ are independent of "t". In Sect. 7.2.3, an approximate eigenvector property was derived in terms of (complex) vectors $W_j(z_p)$ associated with the positive, real eigenvalues $\sigma_j^2(z_p)$ of the matrix $G(z)R^{-1}G^T(z^{-1})Q$ at frequencies $z = z_p = e^{(2\pi ip)/(N+1)}$, $0 \le p \le N$. These relationships are precisely those required by Theorem 13.1, as, for large values of N, it follows that $\|(\sigma_j^2(z)I - GG^*)W_j(z)\| \ll \|W_j(z)\|$ for all z on the unit circle in the complex plane. As a consequence, it is concluded that the choice of a "frequency" $z_{c,k}$ on the unit circle followed by setting

$$\tilde{\sigma}^2_{k+1} = \varepsilon^{-2}\sigma_j^2(z_{c,k}) \quad \textit{on each iteration} \quad (\textit{for } k \ge 0) \tag{13.51}$$

will have the effect of annihilating, approximately, the frequency content of the error at the frequency $z = z_{c,k}$ in the subspace generated by the eigenvector $w_j(z_{c,k})$ of $G(z)R^{-1}G^T(z^{-1})Q$ corresponding to the eigenvalue $\sigma_j^2(z_{c,k})$. Of course, for multi-output systems, other parts of the frequency content (corresponding to eigenvalues of $G(z)R^{-1}G^T(z^{-1})Q$ at other frequencies that are equal to $\tilde{\sigma}^2_{k+1}$) will also be suppressed. The notch is hence a notch based on "gain" values rather than individual frequencies. This interpretation is simplified for SISO systems as $w_j(z_{c,k}) = 1$ and the associated eigenvalue is exactly $G(z_{c,k})R^{-1}G^T(z_{c,k}^{-1})Q = QR^{-1}\left|G(z_{c,k})\right|^2$. This links the choice to the frequency domain and the values of Q and R in a familiar way but does not specify a suitable value of $z_{c,k}$. Intuitively, a good choice might be either

1. a frequency, generated from an analysis of e_k, representing a frequency range where error magnitudes are large or
2. a frequency representing physical phenomena that are ideally suppressed rapidly.
3. Theorem 13.6 may also play a role in limiting the choice. That is, if better convergence rates than those achieved by the NOILC Algorithm are required, the choice of $z_{c,k}$ is limited to a frequency range defined, approximately, by values satisfying

$$\sup_{|z|=1} |G(z)|^2 \le \varepsilon^2 RQ^{-1} + 2|G(z)|^2. \tag{13.52}$$

Finally, in theoretical terms, the frequency analysis of e_k has used the supervector description and is based on representations of the error as a finite summation of the basis vectors $\{W_j(z_k)\}$. In practice, it is more likely to be assessed using the familiar Fast Fourier Transform (FFT).

13.3.4 Robustness of the Notch Algorithm in Feedforward Form

Robustness of Notch Algorithm 13.3 is an important issue. Intuitively, it is expected to be less robust than NOILC Algorithm 9.1 as modelling errors will inevitably reduce the precision and effectiveness of the annihilation that can be achieved. Given a measured error signal e_k on iteration k, let $y = Gu + d$ be a model of the plant but suppose that the plant has a modelling error represented by a linear, bounded, multiplicative modelling error U. Off-line feedforward computations using this model ignore this error and produce the data

$$\begin{aligned}
u^{(1)}_{k+1} &= u_k - \sigma^{-2}\varepsilon^{-2}G^*\left(e^{(1)}_{k+1} - e_k\right),\\
e^{(1)}_{k+1} &= (\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)e_k,\\
u_{k+1} &= u^{(1)}_{k+1} + \varepsilon^{-2}G^*\left(e_{k+1} - e^{(1)}_{k+1}\right) \quad \textit{and predicted error}\\
e_{k+1} &= L(\sigma^2)e_k = \left(I + \varepsilon^{-2}GG^*\right)^{-1}(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\left(\varepsilon^{-2}GG^* - \sigma^2 I\right)^2 e_k
\end{aligned} \tag{13.53}$$

Simple algebra then indicates that the implemented input signal takes the form

$$\begin{aligned}
u_{k+1} &= u_k + \varepsilon^{-2}G^*(I + \varepsilon^{-2}GG^*)^{-1}\Gamma(GG^*, \sigma^2)e_k\\
&\textit{where}\\
\Gamma(GG^*, \sigma^2) &= (1+\sigma^2)^2(\sigma^4 I + \varepsilon^{-2}GG^*)^{-1}\\
\textit{and}\quad L(\sigma^2) &= I - (I - L)\Gamma(GG^*, \sigma^2) \quad \textit{where} \quad L = (I + \varepsilon^{-2}GG^*)^{-1}
\end{aligned} \tag{13.54}$$

is the familiar operator seen in NOILC (Chap. 10). Note that both $\Gamma(GG^*, \sigma^2) : \mathscr{Y} \to \mathscr{Y}$ and $\Gamma(G^*G, \sigma^2) : \mathscr{U} \to \mathscr{U}$ are self-adjoint and have bounded inverses.

Left Multiplicative Modelling Errors: Suppose that σ^2 is iteration independent and that U is a *left* multiplicative modelling error. It then follows that the error seen on the plant following application of u_{k+1} will be

$$e_{k+1} = (I - \varepsilon^{-2}UGG^*(I + \varepsilon^{-2}GG^*)^{-1}\Gamma(GG^*, \sigma^2))e_k = (I - U(I - L(\sigma^2)))e_k. \tag{13.55}$$

With a simple change in notation, this is just the expression seen in Sect. 9.2.4 with L replaced by $L(\sigma^2)$. Using the same approach as that section, an inner product on $\overline{\mathscr{R}[G]}$ can be defined to be

$$\langle e, w\rangle_{\sigma^2} = (1+\sigma^2)^{-2}\langle e, (I - L(\sigma^2))w\rangle_{\mathscr{Y}} \tag{13.56}$$

where the term $(1+\sigma^2)^{-2}$ is introduced to scale the inner product and associated norm. It has the benefit of simplifying the expression and plays only a minor role in the following examination of inequalities. Note that

Theorem 13.7 (Topological Equivalence of Norms) *With the definitions given above, the norm* $\|e\|_{\sigma^2} = \sqrt{(1+\sigma^2)^{-2}\langle e, (I - L(\sigma^2))w\rangle_{\mathscr{Y}}}$ *is topologically equivalent to* $\|e\|_0 = \sqrt{\langle e, (I-L)e\rangle_{\mathscr{Y}}}$ *where* $L = (I + \varepsilon^{-2}GG^*)^{-1}$ *as*

$$\frac{1}{\sigma^4 + \varepsilon^{-2}\|G^*\|^2}(I - L) \leq (1+\sigma^2)^{-2}(I - L(\sigma^2)) \leq \sigma^{-4}(I - L) \tag{13.57}$$

and hence, for any $e \in \mathscr{Y}$,

$$\left(\frac{1}{\sigma^4 + \varepsilon^{-2}\|G^*\|^2}\right)\|e\|_0^2 \leq \|e\|_{\sigma^2}^2 \leq \sigma^{-4}\|e\|_0^2 \tag{13.58}$$

Note: As consequence, convergence or boundedness in $\overline{\mathscr{R}[G]}$ *with respect to one norm implies that using the other norm.*

It is now easily shown that,

Theorem 13.8 (A Robustness Result for the Notch Algorithm with Constant σ^2) *Suppose that* $\tilde{\sigma}_{k+1}^2 = \sigma^2$ *is iteration independent. Then, Theorem 9.15, with L replaced by* $L(\sigma^2)$, *remains valid for the Notch Algorithm 13.3 if the norm* $\|\cdot\|_0$ *used is replaced by*

$$\|e\|_{\sigma^2}^2 = (1+\sigma^2)^{-2}\langle e, (I - L(\sigma^2))e\rangle_{\mathscr{Y}}, \quad e \in \mathscr{Y}. \tag{13.59}$$

More precisely, if there exists a real number $\varepsilon_0^2 > 0$ *such that, expressed in terms of the original topology in* $\mathscr{Y}$,

$$U + U^* \geq U^*(I - L(\sigma^2)U + \varepsilon_0^2 I \quad on \quad \mathscr{R}[G], \tag{13.60}$$

then ,

1. *the monotonicity condition* $\|e_{k+1} - e_0^{(2)}\|_{\sigma^2} \leq \|e_k - e_0^{(2)}\|_{\sigma^2}$ *is satisfied for all* $k \geq 0$.
2. *In addition, Theorem 13.7 then indicates that the sequence* $\{\|e_k\|_0\}_{k\geq 0}$ *is bounded. It converges to zero if, and only if, the sequence* $\{\|e_k\|_{\sigma^2}\}_{k\geq 0}$ *converges to zero.*

Simplifications are possible by noting that,

$$I - L(\sigma^2) = \Gamma(GG^*, \sigma^2)(I - L) \leq \frac{(1+\sigma^2)^2}{\sigma^4}(I - L) \tag{13.61}$$

so that a sufficient condition for robustness is that

$$U + U^* \geq \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right)U^*(I - L)U + \varepsilon_0^2 I \quad on \quad \mathscr{R}[G]. \tag{13.62}$$

which is the condition seen in Theorem 9.15 with an additional scaling factor of $\frac{(1+\sigma^2)^2}{\sigma^4} > 1$. This expression immediately suggests that the Notch Algorithm 13.3 is likely to be less robust than NOILC and that this robustness probably reduces as σ^2 gets smaller. Use of the notch algorithm will therefore be a balance between the benefits of the improved convergence rates and annihilation properties as measured against any loss in robustness if modelling errors are thought to be large.

Further simplifications of the robustness criterion follow directly by defining

$$\beta_I = \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right)\left(\frac{\varepsilon^{-2}\|G^*\|^2}{(1+\varepsilon^{-2}\|G^*\|^2)}\right) \quad \textit{and} \quad \beta_G = \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right)\varepsilon^{-2}, \tag{13.63}$$

from which, using the same norm definitions, it follows that

Theorem 13.9 (A Simplified Robustness Result for the Notch Algorithm) *Suppose that $\tilde{\sigma}^2_{k+1} = \sigma^2$ is iteration independent. Then, condition Four of Theorem 9.16, with β_I and β_G defined as above and with $\|\cdot\|_0$ replaced by $\|\cdot\|_{\sigma^2}$, remains valid as a predictor of robustness for the Notch Algorithm 13.3.*

In particular,

1. The parameters β_I and β_G depend on the choice of Q, R, ε^2 and σ^2 and reduce to values used in Theorem 9.16 as $\sigma^2 \to \infty$.
2. For application to discrete state space models, the results are easily translated into frequency domain criteria simply by using the new definitions of parameters. The details are left as an exercise for the reader and should be related to Theorems 9.17 and 9.18.

The assumption that σ^2 is iteration independent can be relaxed by noting that

$$(1+\sigma^2)^{-2}(I - L(\sigma^2)) - (1+\mu^2)^{-2}(I - L(\mu^2)) \;\geq\; 0 \quad \textit{if} \quad \mu^2 \geq \sigma^2 \tag{13.64}$$

so that $\|e\|_{\sigma^2}$ reduces, monotonically, as σ^2 increases.

Theorem 13.10 (Robustness using Monotonic Values of $\tilde{\sigma}^2_{k+1}$) *With the notation of the discussion above, assume that the sequence $\{\tilde{\sigma}^2_{k+1}\}_{k\geq 0}$ is monotonically increasing from an initial value $\tilde{\sigma}^2_1 > 0$ and bounded by the value $\tilde{\sigma}^2_\infty < \infty$. Then, if*

$$U + U^* \geq \left(\frac{(1+\tilde{\sigma}^2_1)^2}{\tilde{\sigma}^4_1}\right) U^*(I-L)U + \varepsilon_0^2 I \quad \textit{on} \quad \mathscr{R}[G], \tag{13.65}$$

the error sequence generated by the Notch Algorithm 13.3 satisfies the monotonicity condition

$$\|e_{k+1}\|^2_{\tilde{\sigma}^2_{k+2}} \leq \|e_{k+1}\|^2_{\tilde{\sigma}^2_{k+1}} \leq \|e_k\|^2_{\tilde{\sigma}^2_{k+1}}, \quad \textit{for} \quad k \geq 0. \tag{13.66}$$

In particular, there exists a real number $E_\infty \geq 0$ *such that*

$$\lim_{k\to\infty} \|e_k\|_{\tilde{\sigma}_{k+1}^2} = E_\infty \quad \textit{and} \quad \limsup_{k\to\infty} \|e_k\|_0 \leq E_\infty \sqrt{\sigma_\infty^4 + \varepsilon^{-2}\|G^*\|^2} \tag{13.67}$$

and hence $\lim_{k\to\infty} \|e_k\|_0 = 0$ *if* $E_\infty = 0$.

Proof The proof follows from the discussion that precedes the theorem and the observation that the condition (13.65) applied to iteration $k+1$ gives, using Theorem 13.8 with $\sigma^2 = \tilde{\sigma}_{k+1}^2 \geq \tilde{\sigma}_1^2$, $\|e_{k+1}\|^2_{\tilde{\sigma}_{k+1}^2} \leq \|e_k\|^2_{\tilde{\sigma}_{k+1}^2}$. The result is then easily proved using the monotonicity assumption to give $\|e_{k+1}\|^2_{\tilde{\sigma}_{k+2}^2} \leq \|e_{k+1}\|^2_{\tilde{\sigma}_{k+1}^2}$. The existence of the limits then follows with the bounds on $\|e_k\|_0$ being deduced using Theorem 13.7. □

The monotonicity assumption for $\{\tilde{\sigma}_j^2\}_{j\geq 1}$ is a useful assumption for analysis as it adds a structure that produces a simple argument. For state space systems, Sect. 13.3.3, it can be interpreted as choosing values of $\varepsilon^2\tilde{\sigma}_j^2$ that sweep a range of eigenvalue magnitudes from a defined minimum $\tilde{\sigma}_1^2$ to some defined maximum $\tilde{\sigma}_\infty^2$. As this maximum increases, the closer the asymptotic behaviour and robustness will be to that observed in NOILC. If $\tilde{\sigma}_1^2$ is too small, then robustness in the initial iterations may be compromised, an observation that can be interpreted as a warning that the notch procedure should not be used to "annihilate" high frequency components of the error.

Right Multiplicative Modelling Errors: Again assume that $\tilde{\sigma}_{k+1}^2 = \sigma^2$ is independent of iteration. Suppose also that U is now a right multiplicative modelling error so that, in a feedforward implementation, the observed error evolution is simply $e_{k+1} = (I - \varepsilon^{-2}GUG^*L\Gamma(GG^*, \sigma^2))e_k$. The techniques used in Sect. 9.2.2 for analysis of right multiplicative perturbations can now be used. The key to the analysis is to note that $L\Gamma(GG^*, \sigma^2) = \Gamma(GG^*, \sigma^2)L$ is self-adjoint and strictly positive and to use the inner product in $\mathscr{Y}$ defined by

$$\langle e, w\rangle_{\sigma^2} = (1+\sigma^2)^2\langle e, L\Gamma(GG^*, \sigma^2)w\rangle_{\mathscr{Y}}. \tag{13.68}$$

The norm induced by this inner product is equivalent to the norm $\|\cdot\|_0$ used in Sect. 9.2.2 as

$$\begin{array}{c}\left(\frac{1}{\sigma^4+\varepsilon^{-2}\|G^*\|^2}\right) L \leq (1+\sigma^2)^{-2}L\Gamma(GG^*, \sigma^2) \leq \sigma^{-4}L \quad \textit{implies that} \\ \left(\frac{1}{\sigma^4+\varepsilon^{-2}\|G^*\|^2}\right) \|e\|_0^2 \leq \|e\|_{\sigma^2}^2 \leq \sigma^{-4}\|e\|_0^2.\end{array} \tag{13.69}$$

With this definition, the analysis of Sect. 9.2.2 can be used and leads easily to the following result

Theorem 13.11 (A Robustness Result for the Notch Algorithm with Constant σ^2) *Suppose that $\tilde{\sigma}_{k+1}^2 = \sigma^2$ is iteration independent. Then, Theorem 9.15, with L replaced by $L\Gamma(GG^*, \sigma^2)$ and $\|\cdot\|_0$ replaced by $\|\cdot\|_{\sigma^2}$, remains valid for the Notch Algorithm 13.3. More precisely, Condition One is replaced by the condition*

$$U + U^* > \varepsilon^{-2} U^* G^* L \Gamma(GG^*, \sigma^2) GU, \quad on \quad \mathscr{R}[G^*]. \tag{13.70}$$

It ensures that $\|e_{k+1}\|_{\sigma^2} \leq \|e_k\|_{\sigma^2}$ for all $k \geq 0$. Boundedness of both $\{\|e_k\|_{\sigma^2}\}_{k\geq 0}$ and $\{\|e_k\|_0\}_{k\geq 0}$ then follow from topological equivalence of the norms as does the equivalence of their convergence to zero (if it occurs).

As with the case of left multiplicative perturbations, the inequality is satisfied if

$$U + U^* > \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right) \varepsilon^{-2} U^* G^* L G U, \quad on \quad \mathscr{R}[G^*] \tag{13.71}$$

which is precisely Theorem 9.6 with the added factor of $\left(\frac{(1+\sigma^2)^2}{\sigma^4}\right) > 1$ suggesting a reduction in robustness as σ^2 gets smaller. It is now an easy task to prove that monotonicity of the sequence $\{\|e_k\|_{\sigma^2}\}_{k\geq 0}$ for the notch algorithm with iteration independent values $\tilde{\sigma}_{k+1}^2 = \sigma^2 > 0$ follows in a similar way to Theorem 9.8 from the condition

$$\hat{U} + \hat{U}^* > \theta \beta_I U^* U + (1-\theta)\beta_G G^* G \quad on \ \mathscr{U} \quad for \ some \ \theta \in [0,1], \tag{13.72}$$

provided that β_I and β_G are replaced by

$$\beta_I = \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right) \frac{\varepsilon^{-2}\|G\|^2}{1+\varepsilon^{-2}\|G\|^2} \quad and \quad \beta_G = \left(\frac{(1+\sigma^2)^2}{\sigma^4}\right) \varepsilon^{-2}. \tag{13.73}$$

The reminder of the analysis follows the pattern seen for the case of left multiplicative modelling errors and is left as an exercise for the reader who will note that $\|e\|_{\sigma^2}$ again reduces as σ^2 increases. As a consequence Theorem 13.10 remains valid in the form,

Theorem 13.12 (Robustness using Monotonic Values of $\tilde{\sigma}_{k+1}^2$) *With the notation of the discussion above, assume that the sequence $\{\tilde{\sigma}_{k+1}^2\}_{k\geq 0}$ is monotonically increasing from an initial value $\tilde{\sigma}_1^2 > 0$ and bounded by the value $\tilde{\sigma}_\infty^2 < \infty$. Then, if*

$$\hat{U} + \hat{U}^* > \left(\frac{1+\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2}\right)^2 U^* G^* L G U \quad on \quad \mathscr{R}[G^*], \tag{13.74}$$

the error sequence generated by the Notch Algorithm 13.3 satisfies the monotonicity condition

$$\|e_{k+1}\|_{\tilde{\sigma}_{k+2}^2}^2 \leq \|e_{k+1}\|_{\tilde{\sigma}_{k+1}^2}^2 \leq \|e_k\|_{\tilde{\sigma}_{k+1}^2}^2, \quad for \quad k \geq 0. \tag{13.75}$$

In particular, there exists a real number $E_\infty \geq 0$ *such that*

$$\lim_{k\to\infty} \|e_k\|_{\tilde{\sigma}^2_{k+1}} = E_\infty \quad \textit{and} \quad \limsup_{k\to\infty} \|e_k\|_0 \leq E_\infty \sqrt{\sigma_\infty^4 + \varepsilon^{-2}\|G^*\|^2} \tag{13.76}$$

and hence $\lim_{k\to\infty} \|e_k\|_0 = 0$ *if* $E_\infty = 0$.

13.4 Discussion and Further Reading

Successive projection was defined in Chap. 12 based on the work in [95]. That reference provided a convincing indication that the geometry of the Hilbert space H can lead to slow convergence. In the case of NOILC Algorithm 9.1, this is easily seen in computational studies and is primarily associated with the use of large values of weight ε^2 (as well as other dynamic factors such as non-minimum-phase properties). Unfortunately, for state space models, small values of weight can lead to high gain feedback solutions. The case for using an acceleration mechanism in Iterative Control that does not require small weight values is therefore appealing, although, being model-based, issues of robustness need to be considered.

The work in [95] proposed an extrapolation approach based on Successive Projection. This was adopted in [83] for Iterative Control and its accelerating effects have been demonstrated in computational studies [27, 83]. Reductions in robustness, as compared with NOILC Algorithms such as Algorithm 9.1, can be expected in practice as the methodology is an extrapolation process using data from models and the plant. High values of the extrapolation factor λ_{k+1} could amplify the errors leading to erratic algorithm performance and, potentially, divergence. The most dangerous situation will occur if the sets S_1 and S_2 do not intersect. That is, there is no solution to the tracking problem. Reference [95] then suggests that wild oscillations and divergence could occur unless the design uses the flexibility in the choice of λ_{k+1} and puts practical limits $\lambda \in [1, \lambda_{max}]$ on the range used in the implementation!

The concept of successive projection seems to have considerable power in linking algorithms to practical experimental processes. There is no real reason why the sets involved need not vary from iteration to iteration. The question is "how should the sets vary and what effect will the choice and variation have on convergence rates and robustness?". This text has introduced the *Notch Algorithm* 13.3 which is new to the literature and uses this flexibility in the form of a parameterized family of sets in the product space $\mathscr{Y} \times \mathscr{U}$. Essentially, it applies a scaling factor $-\sigma^2$ to one term in the error definition, follows this by constructing a set $S_2(\sigma^2)$ to complement the dynamics S_1, and then applies successive projection. In the form presented, it is an algebraic construct but has properties of acceleration demonstrated by its improvements on NOILC as stated in Theorem 13.3 and, more fundamentally, the property of exact or approximate annihilation of spectral components of the error signal. Simple calculations based on the form of $f(\mu, \sigma^2)$, $0 < \mu \leq \|G^*\|^2$ provide

considerable insight into the benefits of this algorithm, particularly when the factor σ^2 is varied from iteration to iteration in an intelligent way.

The key to analysing the benefits on iteration $k+1$ lie in the consideration of functions defined by the products, with $k \geq 0$,

$$\prod_{j=0}^{k} f(\mu, \tilde{\sigma}_{j+1}^2) = \frac{1}{(1+\varepsilon^{-2}\mu)^{k+1}} \prod_{j=0}^{k} \frac{(\tilde{\sigma}_{j+1}^2 - \varepsilon^{-2}\mu)^2}{(\tilde{\sigma}_{j+1}^4 + \varepsilon^{-2}\mu)}, 0 < \mu \leq \|G^*\|^2. \quad (13.77)$$

The first factor describes the changes that would arise from $k+1$ iterations of basic NOILC Algorithm 9.1. The second factor

$$\prod_{j=0}^{k} \frac{(\tilde{\sigma}_{j+1}^2 - \varepsilon^{-2}\mu)^2}{(\tilde{\sigma}_{j+1}^4 + \varepsilon^{-2}\mu)}, 0 < \mu \leq \|G^*\|^2, \quad (13.78)$$

describes the way in which the various notches used on the iterations in Algorithm 13.3 affect the performance of the NOILC Algorithm. This term has well-defined "zeros" at $\varepsilon^2 \tilde{\sigma}_{j+1}^2, 0 \leq j \leq k$, providing the desired annihilation properties. In addition, the placing of the values opens up the opportunity, for a given value of "k", of ensuring that f has very small values over a chosen spectral range. The interested reader will be able to assess the effects and benefits by looking at a simple case with $\varepsilon^2 = \|G^*\|^2 = 1$, choosing $k = 2$ with $\tilde{\sigma}_1^2 = 1, \tilde{\sigma}_2^2 = 0.5$ and $\tilde{\sigma}_3^2 = 0.25$ and plotting the form of the expression above in the range $0 < \mu \leq 1$. A remarkable reduction in values will be observed and demonstrates the ability to create fast convergence without "high gain" control loops. In this case, for initial errors e_0 dominated by the contribution of the spectral values $\sigma^2 > 0.25$, rapid reductions in error magnitude can be achieved.

Finally, the robustness analysis has used the mathematical construct of introducing new norms to the problems and deriving operator inequalities that define tolerable multiplicative modelling errors. These inequalities are based on the artificial requirement of requiring monotonicity properties of the error sequence as measured by the new norms. The results provide some reassurance that bounded responses will be seen in practice. However, the precise behaviour of the errors as measured by the original norm $\|\cdot\|_{\mathscr{Y}}$ is difficult to ascertain from the properties of L and $L(\sigma^2)$, particularly when the "notch" σ^2 is varied from iteration to iteration.