Chapter 1

Summary and Introduction

There is a long-standing and growing interest in computer algorithms which provide means for flexibility, for adaptation and for learning from mistakes. Within different application areas, such devices are known as learning systems [129], adaptive systems, adaptive filters [51], [52],[90],[110],[135] and adaptive controllers [41],[137].

Models of the desired response of the system and models of the external world are central building blocks of such schemes. These models are often represented as dynamic systems with adjustable parameters. When the dynamics is time-varying, the parameters need to be varied by an adjustment scheme called an adaptation law, or a parameter tracking algorithm.

The primary aim of the present thesis is to provide a framework for the systematic design of adaptation laws for linear regression models of dynamic systems. The work will result in a class of algorithms, which represent different tradeoffs between performance and algorithm complexity. The framework will also prove useful in the analysis of the properties of existing and new parameter tracking algorithms.

1.1 Parameter tracking: Outlining the problem

The task of adaptation laws can often be formulated as that of modeling the dynamic response of a known signal. A known *input* signal u_t is to be related to the response, or output, y_t by adjusting parameters of an input-output model. (Time series modeling problems without inputs will not be considered here.) Let us outline how adaptation problems are solved at present, and introduce the types of questions that have motivated the present work.

1.1.1 Tracking the parameters of linear regression models

Dynamic models could be structured in various ways, but the linear regression is, at present, the by far most popular and well understood structure [16],[107],[135]. Linear regression models are linear in the adjustable parameters, and this simplifies both the design and the analysis of adaptation laws. Our interest will in this thesis be focused on multivariate linear regression models, expressed as

$$\begin{aligned} y_t &= \hat{y}_t + \varepsilon_t \\ \hat{y}_t &= \varphi_t^* \hat{h}_t \ . \end{aligned} \tag{1.1}$$

Here, $\hat{y}_t$ is a column vector of dimension n_y. It represents an estimate of the measured signal y_t, at the discrete time instant denoted by the integer t. All signals may be complex-valued.

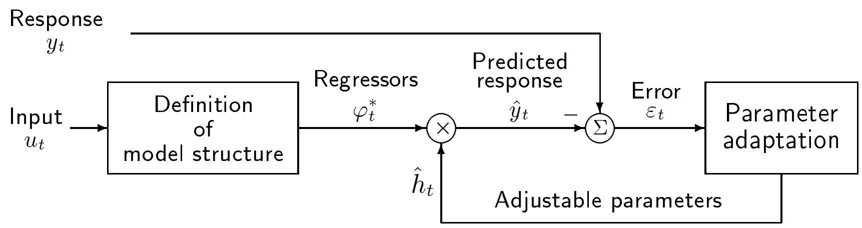

Figure 1.1: The linear regression model structure (1.1) and an error-driven adaptation law.

The column vector $\hat{h}_t$ contains parameters, which may vary with time, and the *regressor matrix* φ_t^* contains signals which are known or computable at time t. The signal y_t represents either a desired response or the response of an external process, which is to be modeled. See Figure 1.2. In the first case, the task is called direct adaptation, whereas in the second case, we have a problem of system identification. The use of adaptively identified models in the design of a time-varying filter or controller is termed indirect adaptation.

The elements of the regressor matrix φ_t^* may consist of delayed inputs and outputs as well as filtered [131] and also nonlinear functions [37] of inputs and outputs. In this thesis, the discussion will be specialized to models in which delayed or filtered versions of the input y_t are not utilized as regressor variables.

The *estimation error* ε_t of the linear regression model structure is also, in various contexts, referred to as the residual, the prediction error, the equation error and the output error. The parameters of the model (1.1) are

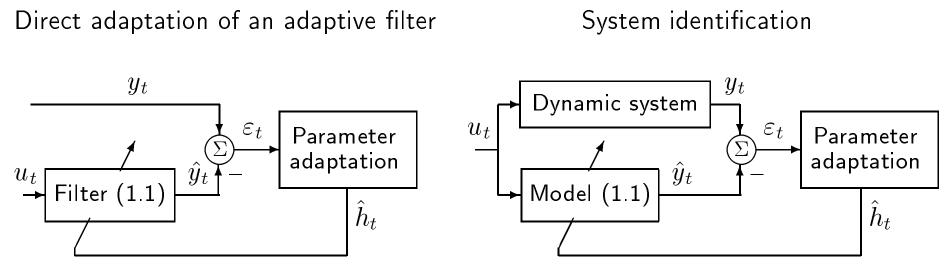

Figure 1.2: Two ways of utilizing a linear regression model: Direct adaptation of a linear filter and system identification.

to be adjusted so that the estimation error ε_t is minimized, in some sense. Model adjustment could be viewed exclusively as a task of criterion minimization; the error ε_t should be minimized, for a particular set of signals $\{y_\ell, u_\ell \ ; \ \ell = 0 \dots t\}$.

Here, the problem will be formulated somewhat differently. We will mainly focus on the misadjustment of the parameter estimates $\hat{h}_t$ with respect to an ideal adjustment. The problem of adaptive modeling can be formulated in this way only under certain conditions. If the regressors are seen as stochastic variables, then the optimal adjustment of the parameters will, in general, depend on the statistical properties, such as the spectral content, of the regressors [85]. A unique underlying set of "true parameters" will therefore, in general, not exist. An exception to this rule is a situation described below.

Assumption 1.1 The signal y_t can be described by a dynamic system with the same structure as that of the model (1.1),

$$y_t \ = \ \varphi_t^* h_t \ + \ v_t \ , \tag{1.2}$$

where the noise v_t has zero mean and is statistically independent of both the vector h_t and of the regression matrix φ_t^* □

Assumption 1.1 can be approximately fulfilled in a large number of practical adaptation problems, by selecting the structure of the model (1.1) in an appropriate way. The choice $\hat{h}_t = h_t$ would then represent an ideal adjustment of the model parameters, on average over all realizations of the regressors and the noise. We may then refer to the vector h_t as the *true parameters.* Under Assumption 1.1, it becomes meaningful to define the task of a parameter tracking algorithm as that of making the estimates follow the true parameters, by minimizing some function of the parameter estimation error

$$\tilde{h}_t \;=\; h_t \;-\; \hat{h}_t \;\;.$$

Since the true parameters are unmeasurable, the minimization of the parameter adjustment error must be performed indirectly, by minimizing a function of the estimation error

$$\varepsilon_t \;=\; y_t - \hat{y}_t \;=\; y_t - \varphi_t^* \hat{h}_t \;=\; \varphi_t^* \tilde{h}_t + v_t \;\;. \tag{1.3}$$

A commonly used criterion for minimizing the estimation error is the windowed *mean square error (MSE)*

$$\sum_{\ell=0}^{t} w_\ell \varepsilon_\ell^* \varepsilon_\ell \;=\; \sum_{\ell=0}^{t} w_\ell \sum_{j=1}^{n_y} |y_\ell^j - \hat{y}_\ell^j|^2 \;\;, \tag{1.4}$$

where the scalar function w_ℓ represents a weighting profile and $*$ denotes complex conjugation and transposition. Differing weights on the components $\varepsilon_\ell^j = y_\ell^j - \hat{y}_\ell^j$ of the error vector could also be introduced.

An example of a special case of (1.1) is a scalar second order finite impulse response (FIR) model

$$y_t \;=\; \hat{h}_t^0 u_t + \hat{h}_t^1 u_{t-1} + \varepsilon_t \;\;, \tag{1.5}$$

obtained by the choice

$$\varphi_t^* \;=\; (\; u_t \; u_{t-1} \;) \quad ; \quad \hat{h}_t \;=\; \begin{pmatrix} \hat{h}_t^0 \\ \hat{h}_t^1 \end{pmatrix} .$$

FIR filters and models are utilized in numerous applications [107]. An area of growing importance is the modeling of *fading transmission channels* in systems for personal, indoor and mobile radio communications. Such channels can be represented by FIR filters with time-varying parameters. Their estimation is important to achieve error-free retrieval of transmitted information. This thesis will, in particular, consider Time Division Multiple Access [87] systems, in which sets of symbols are transmitted in data frames of fixed length.[1] Within each frame (or burst), a small fraction of the transmitted data, the training sequence, is known by the receiver. The training sequence can be correlated with the received signal, to obtain an estimate of the transmission channel. A detector, or equalizer, is then designed by using the estimated channel model, with the aim of retrieving

[1] FIR models with time-varying parameters are also suitable as channel models for mobile radio systems based on DS-CDMA (Direct Sequence Code Division Multiple Access) [40],[109],[127] That type of application is, however, not investigated here.

the remaining data. This procedure is repeated for each data frame [87]. The time dispersion of the communication channel is caused by *multipath propagation*; different signal paths have differing transmission delays. The time variability, the *fading*, is due to the mobile moving through standing wave patterns, caused by multiple scattering of the electromagnetic waves [56],[75],[76],[130]. The estimation of channel models for TDMA systems, with parameters that vary considerably within a few samples, will recur as an example throughout the thesis. The problem is discussed in depth in the Chapters 6 and 8.

1.1.2 Commonly used tracking algorithms

The art of adjusting model parameters time-invariant systems (h_t in (1.2) independent of time), has been studied for decades within the field of system identification [15]. This rather mature field is now covered by many books [60],[85],[125]. For time-invariant systems, the well-known least squares (LS) method can be utilized to adjust the parameters of the linear regression model (1.1).

The use of a batch least squares method would be a straightforward way to estimate a time-varying parameter vector h_t. The time series are then partitioned into subsets (batches) of data. For each batch, a model with constant parameters is adjusted by minimizing the sum of squared errors ε_t. The result is a piecewise constant parameter estimate.

A more commonly used alternative is to use *recursive algorithms*, which compute new parameter estimates at time t by processing previous estimates. A frequently utilized class of recursive algorithms is specified by

$$\begin{aligned} \varepsilon_t &= y_t - \varphi_t^* \hat{h}_{t-1} \\ \hat{h}_t &= \hat{h}_{t-1} + \mathbf{\Gamma}_t \varphi_t \varepsilon_t \ . \end{aligned} \tag{1.6}$$

Here, $\varphi_t \varepsilon_t$ is the instantaneous gradient of $\frac{1}{2}|\varepsilon_t|^2$ with respect to the previous estimate $\hat{h}_{t-1}$. The gain matrix $\mathbf{\Gamma}_t$ may modify this gradient updating direction. A unifying framework for describing algorithms for recursive identification, as well as stochastic averaging methods for analysis, were introduced by Ljung in the 1970's. See the book [84] by Ljung and Söderström.

The LMS algorithm, introduced by Widrow and Hoff around 1960 [132], is obtained by substituting the gain matrix $\mathbf{\Gamma}_t$ in (1.6) by a scalar gain μ. The LMS algorithm is the earliest, the simplest, but still the most widely utilized recursive scheme.

The recursive least squares (RLS) algorithm utilizes a gain matrix obtained via a Riccati equation. That equation is derived by minimizing the criterion (1.4) with respect to a time-invariant parameter $\hat{h}$. The tracking capability will be determined by the time window w_ℓ in (1.4). With a non-windowed algorithm ($w_\ell =$ constant), the gain matrix $\mathbf{\Gamma}_t$ would go to zero as $t \to \infty$, so the capacity for tracking would be lost. That effect can be avoided by discounting old data. An exponential window

$$w_\ell = \lambda^{t-\ell}$$

is often used for this purpose, but other suggestions exist as well, see e.g. [78],[100] and [101]

The application of algorithms of the class (1.6), which were originally derived for the recursive identification of time-invariant parameters, is one way to solve parameter tracking problems. It is, however, not the only way. As will be discussed in Section 2.4.3, it is rarely the optimal way.

1.1.3 Towards a systematic methodology for the design of tracking algorithms

A fundamental tradeoff is encountered in the design of tracking algorithms based on windowed criteria, such as (1.4): with a long time window, the estimate will tend to lag behind the trajectory of the true parameter vector. This effect is called a *lag error*. The lag can be reduced by shortening the effective length of the window, but that would instead increase the noise sensitivity of the estimate.

The tradeoff between lag error and noise sensitivity can be affected by the batch length in batch-LS schemes, by the scalar adaptation gain in LMS and by the forgetting factor in exponentially windowed RLS estimators. The effect of adjusting the gain of an LMS algorithm is illustrated by Figure 1.3. Numerous papers have discussed the optimal adjustment of different methods. See e.g. [39],[89],[133] for LMS, [25],[48],[83] for RLS and [27],[78],[86],[95] for studies which compare the performance of different algorithms. The LMS and the RLS algorithms have been claimed to be just about equally good for tracking of slowly drifting parameters, cf. e.g. [134]. The same conclusion will be drawn in this thesis.

For slow time variations of h_t, various reasonably tuned algorithms may all turn out to be adequate. The situation is very different when the parameters vary on time scales on the order of tens of samples. For fast time

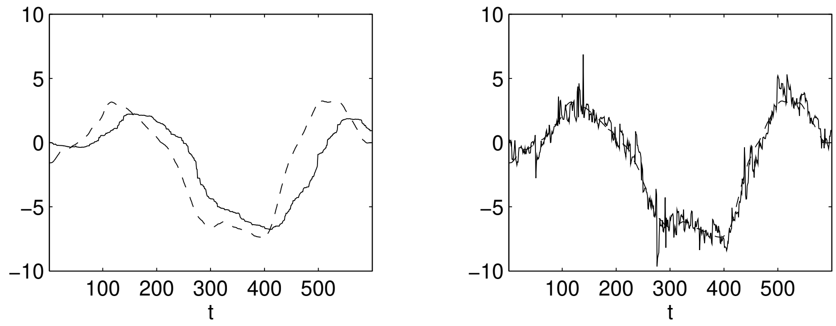

Figure 1.3: Illustration of the tracking performance of the LMS algorithm (solid line), for two different values of the step length parameter $\mathbf{\Gamma}_t = \mu \mathbf{I}, \mu = 0.02$ (left) and $\mu = 0.4$ (right). The dashed lines represent the true parameter variations.

variations, there may not exist any choice of time window which results in an acceptable tracking performance. That situation was encountered in an investigation of a digital communication problem [79], which motivated and inspired the work to be presented here.

Example 1.1 *Tracking of parameters of fading mobile radio channels in TDMA systems*

In the North American D-AMPS or IS-54 standard, the discrete-time baseband transmission channel can be described by the FIR model (1.5). Its coefficients will vary appreciably during a few symbol times, as is evident from Figure 1.4. A model based on the training sequence, which consists of 14 symbols, can therefore not be relied upon over the whole duration of a frame of 170 symbols, so adaptation will be required.

Figure 1.5 illustrates a scheme for the simultaneous estimation of the transmitted symbols and of the parameters of a channel model or an equalizer. An inherent difficulty with such adaptive equalizers is that the updating of the parameters has to utilize estimates $\bar{u}_t$ of the transmitted symbols as regressors. A small increase of the average tracking error will result in a large increase of the probability of symbol estimation errors. If the symbol estimates used as regressor variables are frequently incorrect, then there is even a potential risk of catastrophic failure: The tracking ability may be lost completely.

Consequently, the use of an adaptation algorithm that attains small tracking errors is of crucial importance in this type of receiver. When investigating the performance of different tracking algorithms, it was found that the LMS

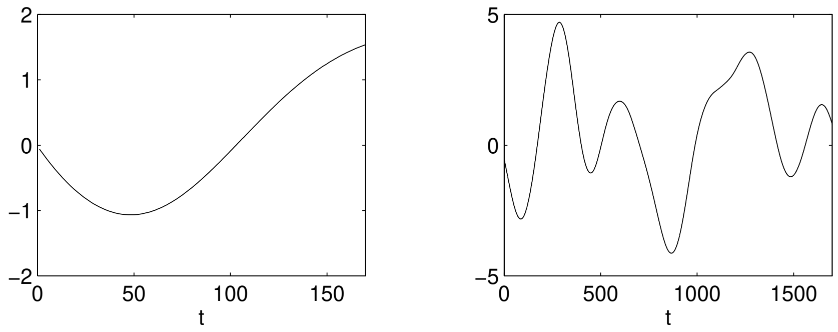

Figure 1.4: Characteristic variation of a fading mobile radio channel coefficient in an IS-54 system, for a mobile traveling at 100 km/h, during one burst (left), and during ten bursts (right).

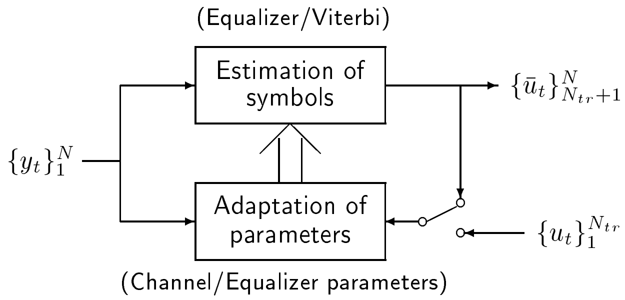

Figure 1.5: Adaptive channel equalization. In learning-directed mode, the adaptation is based on training data. At time instant $t = N_{tr} + 1$, the adaptation is switched into decision-directed mode and decisioned symbols $\bar{u}_t$ are used as regressor variables for the parameter adaptation.

method was inadequate. The use of windowed RLS resulted in a more complex algorithm, but not in an improved performance. No amount of tuning of these algorithms resulted in an acceptable performance, as measured by the bit error rate at moderate signal to noise ratios □

In situations such as the one described by Example 1.1, the designer is forced to consider more fundamental *structural adjustments*, as a complement to parameter tuning, of the adaptation scheme. This brings us to a key issue of the present thesis:

How can structural adjustments of adaptation algorithms be performed systematically?

Before this question can be answered, promising types of structural modifications of adaptation laws must be investigated. Expressed differently, we may ask: what types of additional side information or *a priori* information could be of use, and how could it guide our design efforts?

When trying to address these questions, it can be noted that there are at least two types of information, that cannot be directly utilized in recursive algorithms with the structure (1.6).

1. **The use of future data values.** In off-line modeling as well as in some on-line applications, the estimation of h_{t+k}, where $k < 0$, can be based on outputs and regressor data collected up to time t. The use of such *fixed lag smoothing estimates* $\hat{h}_{t+k|t}$ would reduce the lag error drastically. Smoothing should be implemented, wherever possible. The methodology to be presented in the coming chapters has therefore been formulated to include the design of smoothing estimators, as well as of filtering ($k = 0$) and prediction ($k > 0$) estimators.

2. **Knowledge of the dynamics of the parameters.** Consider, for example, a situation where the parameters are known to vary approximately as sinusoids with known frequencies. One could then strive to redesign (1.6) into an algorithm which performs bandpass filtering around the characteristic frequency. This could improve the tracking performance significantly, as compared to the use of (1.6) directly, which essentially performs lowpass filtering. Modifications along these lines would introduce dynamic systems, filters, into the recursions (1.6). A main aim of the present work is the development of analytical design techniques for the construction of such filters.

The performance of tracking algorithms is important, but it is far from the only issue affecting the choice of algorithm. Other important aspects include the robustness with respect to bad excitation properties of the regressors [137], and robustness with respect to large but infrequent noise samples, or outliers [5],[54],[85].

The computational complexity is also a major issue. A low complexity is crucial in many high-speed signal processing applications, such as the adaptive equalization problem introduced in Example 1.1. The present work has therefore been aimed at answering also the following question:

How can tracking algorithms be designed to provide (nearly) optimal performance at a given, specified, level of complexity?

It has turned out that the use of models for the time-varying parameters h_t is a key element in the answer to both of the above questions.

1.1.4 Hypermodels and Kalman estimators

It is convenient to formalize *a priori* knowledge of the properties of h_t as a mathematical model. The time-varying true parameters will therefore be *regarded as signals*, and dynamic models are introduced to describe their behavior. It is natural to use linear models, unless there are strong reasons to suspect that the dynamics of h_t is inherently highly nonlinear. Linear time-invariant models can be represented in state space form, as a Markovian model

$$\begin{aligned} x_{t+1} &= \mathbf{F}x_t + \mathbf{M}m_t + \mathbf{G}e_{t+1} \\ h_t &= \mathbf{H}x_t \quad . \end{aligned} \tag{1.7}$$

Here, x_t is the state vector, m_t represents any measurable vector which affects the parameters in a known way and $\mathbf{F}, \mathbf{M}, \mathbf{G}, \mathbf{H}$ are matrices of appropriate dimensions. The initial state x_0 is assumed to be unknown and the state is affected by the white random vector e_t. The random vector e_t, which has a specified covariance matrix, will be called the *driving noise* of the model.

Explicit models of the dynamics of time-varying (true) parameters have become known as *hypermodels*. Their use in the design of adaptive algorithms for regression models have been discussed by Benveniste and co-workers [10],[11]. The use of hypermodels in the adaptation of time-varying models of time series has been exploited by e.g. Grenier [44] and by Kitagawa and Gersch [63]. The model (1.7), with $m_t = 0$, can be used in the description of a variety of situations:

- Constant parameters. This case is described by the use of $\mathbf{F} = \mathbf{H} = \mathbf{I}$ and $e_{t+1} = 0$.
- Parameters may evolve as linear combinations of deterministic functions, such as ramps, parabolas or sinusoids with known frequency. Models which generate such functions will be called *deterministic hypermodels*. The states can then be regarded as coefficients in a functional series model of the time series h_t. For ramp functions, double integration of an initial value is required, while parabolic functions are generated by triple integration. The dimension of the state vector x_t will, in general, be higher than the dimension of the vector h_t, since several state variables are required for the description of each parameter. The driving noise can be zero, resulting in a purely deterministic model, with unknown initial condition. The noise could also be assumed to be a random spike sequence, such as a Bernoulli-Gaussian sequence [96], resulting in a piecewise deterministic model.

Deterministic or functional series descriptions have been utilized in e.g. [19],[23],[77],[79][102], and [55].

- In a *stochastic hypermodel*, the driving noise e_{t+1} is assumed to be a white random vector, with known covariance matrix. In the present thesis, we will mainly utilize stochastic hypermodels.

The equation (1.2) for y_t

$$y_t = \varphi_t^* \mathbf{H} x_t + v_t \quad ,$$

will constitute an output equation for the state space model (1.7). Thus, the parameter estimation problem can be formulated as a state estimation problem. Model-based tracking of state variables has been studied and utilized for decades, in e.g. radar applications [31]. If the matrices in (1.7) and the covariance matrices of e_{t+1} and v_t are known, then the *Kalman filter* [62]

$$\begin{aligned} \hat{x}_{t|t} &= \mathbf{F}\hat{x}_{t-1|t-1} + \mathbf{M}m_{t-1} + \mathbf{K}_t^f(y_t - \varphi_t^* \hat{h}_{t|t-1}) \\ \hat{h}_{t+1|t} &= \mathbf{HF}\hat{x}_{t|t} \end{aligned} \tag{1.8}$$

can be utilized to minimize the mean square tracking error. The time-varying gain matrix $\mathbf{K}_t^f$ of the Kalman filter is obtained from a Riccati difference equation, which is iterated forward in time. (See Section 2.3 for details.) Based on the estimator (1.8), predictors, fixed lag smoothers and fixed interval smoothers can be obtained [7],[58],[93]. Kalman estimators can be designed also for time-varying hypermodels.

The Kalman algorithm will constitute the linear estimator which provides the minimal mean square tracking error, in the case of exactly known hypermodels. Its performance will thus constitute the ultimate attainable bound on the performance of estimators with linear structure[2]. Note that the structure of (1.8) will, in general, differ from that of (1.6). The structure of algorithms of the type (1.6) will therefore rarely be optimal.

For further reference, let us mention that the hypermodel (1.7), with $m_t = 0$, may be expressed in transfer function form

[2]When the driving noise e_t consists of a random spike sequence, its covariance matrix will be time-varying in an unknown way. Nonlinear schemes can then attain a better performance than a Kalman estimator based on some average covariance matrix. A simple example of such a scheme would be the use of change detection based on the magnitude of the error $\varepsilon_t = y_t - \varphi_t^* \hat{h}_{t|t-1}$. The detection of a large error could be used to increase the gain of the observer (1.8). This would speed up the transient response when there is a sudden change in e.g. the slope of a linear trend. Schemes for change detection are discussed in e.g. [8]. Such devices will, however, not be utilized in the present thesis.

$$h_t = \boldsymbol{\mathcal{H}}(q^{-1})e_t \quad , \tag{1.9}$$

where

$$\boldsymbol{\mathcal{H}}(q^{-1}) = \mathbf{H}(\mathbf{I} - q^{-1}\mathbf{F})^{-1}\mathbf{G}$$

is a rational matrix with transfer operators in the backward shift operator q^{-1} as elements. The poles of the transfer functions will be assumed to be inside or on the stability limit. We will also express hypermodels as linear difference equations, either in the form

$$\boldsymbol{D}(q^{-1})h_t = \boldsymbol{C}(q^{-1})e_t \tag{1.10}$$

or as

$$h_t = \boldsymbol{F}(q^{-1})h_{t-1} + \boldsymbol{C}(q^{-1})e_t \tag{1.11}$$

where $\boldsymbol{D}(q^{-1})$, $\boldsymbol{C}(q^{-1})$ and $\boldsymbol{F}(q^{-1})$ are polynomials matrices in q^{-1}. By comparing the model structures (1.10) and (1.11), it is evident that

$$\boldsymbol{D}(q^{-1}) = \mathbf{I} - \boldsymbol{F}(q^{-1})q^{-1} \quad .$$

An assumption that the parameters are either constant or can be described as integrated white noises leads to a special case of particular interest, namely the *random walk model*. In the formulation (1.11), the random walk model corresponds to the use of $\boldsymbol{F}(q^{-1}) = \boldsymbol{C}(q^{-1}) = \mathbf{I}$, or

$$h_t = h_{t-1} + e_t \quad . \tag{1.12}$$

Thus, the use of hypermodels of the parameters, and of Kalman estimation based on these models, constitutes a powerful methodology for the design of tracking algorithms. However, the basic premises of such an approach can, and should, be evaluated critically. The points raised by such a discussion motivate the directions of research, pursued in the present thesis.

1. Is an introduction of hypermodels sensible at all?

If the parameters to be tracked are themselves unknown, can we then really postulate the existence of known dynamic models for their behavior? Will this not make the performance of the resulting algorithm sensitive to errors in the assumed dynamics?

We will argue that the selection of an algorithm structure, such as (1.6) will, in itself, entail an implicit assumption about the underlying parameter dynamics for which that particular algorithm is optimal. It can certainly be claimed that the use of a method that motivates the designer to formulate such assumptions in an explicit way will be a more rational approach. The question of sensitivity to underlying assumptions is not coupled to the explicitness of the assumptions. The fact that assumptions are not formulated explicitly does not necessarily imply that the performance of a resulting algorithm would be insensitive to changes in the dynamics of h_t.

If parameters evolve essentially as random walks (1.12), then there will be little reason to go beyond algorithms with the structure (1.6). Situations where the dynamics differs from random walk are, however, more of a rule than an exception. Knowledge of at least the relative magnitudes and the dominating frequency contents or time constants of the parameters will often be available in particular applications.

Sensitivity will in general not be a critical issue. The crucial aspect of any algorithm based on hypermodels is that an appropriate amount of inertia should be built into the adaptation law. The exact correctness of the hypermodel utilized for the design will mostly not be critical. If very little is known about the dynamics of the time variations, then the present work suggests the use of algorithms based not on random walk models, but on *integrated* random walk models, as the reasonable first guess.

A systematic design methodology should, however, take into account the fact that hypermodels are rarely exactly known. In Chapter 3, and also in the coming thesis [140] by Öhrn, methods will be presented for the robust design of tracking algorithms, based on *sets* of possible models (1.7). The aim is to obtain a minimal MSE tracking error, on average over these sets.

2. Does not the use of hypermodeling increase the difficulty and the complexity of the design of adaptive algorithms?

This criticism can certainly be true, if the search for good hypermodels is driven to an extreme. In the present thesis, the use of simple models, which describe the main "first order" dynamic properties of the time-varying parameters, will be emphasized.

Hypermodels can sometimes be parameterized by a few physically relevant parameters, which can themselves be estimated from data. The case study of Chapter 8 will discuss such an example. When investigating ways to reduce the design complexity, the use of self-tuning hypermodels is therefore an interesting topic for research.

3. Will Kalman-based tracking imply a high computational load?

The use of Kalman schemes requires a covariance matrix of dimension x_t to be updated by a Riccati equation at each time step. Smoothing estimators require further calculations. The computational complexity grows with the complexity of the hypermodels. If the dimension of the state vector x_t is higher than that of the parameter vector h_t, then the complexity will be higher than that of RLS. Complexity will be a valid, and indeed crucial, argument against the use of Kalman-based adaptation schemes in many high-speed applications. The work on algorithm design in the present thesis will therefore be focused on hypermodel-based algorithms which avoid the need for updating a Riccati equation at each time instant.

Before outlining the contents of the thesis, let us first briefly mention some alternative approaches to the design of tracking algorithms.

The methodology which is conceptually most closely related to the one to be presented below has been developed by Benveniste and co-workers [10],[11],[12]. Their approach to design and analysis is based on ODE averaging, and it is therefore restricted to situations with slow parameter variations. The method is based on exactly known continuous-time hypermodels. A continuous-time algebraic Riccati equation is utilized to optimize the design of one-step prediction trackers. The resulting adaptation laws are in state-space form and have constant gains. Smoothing and robust design is not considered.

If the parameters evolve as random walks (1.12), then the optimal one-step prediction estimate equals the filter estimate, $\hat{h}_{t+1|t} = \hat{h}_{t|t}$. When the parameter dynamics deviates from random walks, the use of stable rational *coefficient prediction filters*

$$\hat{h}_{t+1|t} = \boldsymbol{\mathcal{P}}(q^{-1})\hat{h}_{t|t}$$

in (1.6) can improve the tracking performance. The use of such filters was introduced and discussed by Kubin [64],[65], but no systematic methodology for the filter design was proposed. The application of the results of this thesis to the design of coefficient prediction filters will be discussed in Chapter 3.

In [19], Clark has suggested the use of "fading-memory prediction", which corresponds to the use of hypermodels with linear trend or integrated random walk dynamics (two integrators for each parameter). The approach by Clark will be discussed in Example 3.8. It can be seen as a special case of the methodology developed in Chapter 3.

In [78], Lin and co-workers present a systematic methodology for optimizing the time window w_τ of the criterion (1.4). It is, at present, unclear if this optimization could be implemented in recursive form.

1.2 Outline of the Thesis

The thesis will consider the design of tracking algorithms in Chapter 2 and 3, the analysis of algorithms in Chapter 4 and 5 and the application to mobile radio communications in the Chapters 6, 7 and 8.

In **Chapter 2**, the linear regression model, the tracking problem and the structure of hypermodels expressed in observer state space form, will be discussed in some detail. Kalman estimators are then reviewed in Section 2.3. The main contribution of Chapter 2 is a reformulation in input-output form of Kalman filter and prediction tracking estimators. Such estimators can, for scalar y_t, be expressed as

$$\begin{aligned}\varepsilon_t &= y_t - \varphi_t^* \hat{h}_{t|t-1} \\ \hat{h}_{t+k|t} &= \boldsymbol{F}(q^{-1})\hat{h}_{t+k-1|t-1} + \boldsymbol{G}_t^k(q^{-1})\varphi_t\varepsilon_t \quad ; \quad k \geq 0 \ .\end{aligned} \tag{1.13}$$

The fixed polynomial matrix *feedback filter* $\boldsymbol{F}(q^{-1})$ is related directly to the hypermodel (1.11). The time-varying polynomial matrix $\boldsymbol{G}_t^k(q^{-1})$ is denoted the *gain filter*. It is determined indirectly by the hypermodel, via the Riccati equation. The expression (1.13) represents a general hypermodel-based tracking algorithm with time-varying gain $\boldsymbol{G}_t^k(q^{-1})$. The RLS and the normalized LMS algorithms will in Section 2.4 be considered as special cases of this algorithm structure.

In **Chapter 3**, a class of algorithms, with structure inspired by (1.13), but with *time-invariant* and possibly rational gain filter

$$\begin{aligned}\varepsilon_t &= y_t - \varphi_t^* \hat{h}_{t|t-1} \\ \hat{h}_{t+k|t} &= \boldsymbol{F}(q^{-1})\hat{h}_{t+k-1|t-1} + \boldsymbol{\mathcal{G}}_k(q^{-1})\varphi_t\varepsilon_t \ ,\end{aligned} \tag{1.14}$$

is introduced. The feedback filter $\boldsymbol{F}(q^{-1})$ is obtained from the hypermodel (1.11), while the selection of the gain filter $\boldsymbol{\mathcal{G}}_k(q^{-1})$ will differ in different special cases. The fixed integer k determines if filtering ($k = 0$), prediction ($k > 0$) or fixed lag smoothing ($k < 0$) estimates are to be processed. The motivation for introducing such adaptive algorithms with time-invariant gain, is that the need for elaborate calculations at each time step, such as the iteration of Riccati equations, is thus eliminated. The price is some performance loss, as compared to the use of an optimally tuned algorithm with structure (1.13). In many situations, the loss will be negligible. The LMS algorithm, momentum LMS, LMS with leakage and different previous suggestions for introducing filtering into LMS update laws can all be seen as special cases of the algorithm structure (1.14).

The question now naturally arises if the gain filter can be designed to minimize the mean square parameter tracking error, for a given hypermodel, or for a set of possible hypermodels. In Chapter 3, such design problems will be expressed in terms of approximately equivalent Wiener filtering problems. These (robust) Wiener design problems are then solved by utilizing the polynomial equations approach, pioneered by Kučera [66], [67], and developed for filter design problems by Ahlén and Sternad [1],[2],[4],[119],[120],[139] and by Grimble [45]. See also the books [3] and [45].

The reformulation of tracking algorithm optimization as linear filter design can be outlined as follows. Assume the regressors to have zero means, and to have a given time-invariant autocorrelation matrix

$$\mathbf{R} \triangleq \mathrm{E}\,\varphi_t\varphi_t^* \quad .$$

If unknown, this matrix will have to be estimated recursively. Introduce the autocorrelation matrix noise [39] as

$$Z_t \triangleq \varphi_t\varphi_t^* - \mathbf{R} \quad , \tag{1.15}$$

and introduce a fictitious measurement signal

$$f_t \triangleq \mathbf{R}h_t + \eta_t \quad , \tag{1.16}$$

where η_t is called the *gradient noise*, and is given by

$$\eta_t \triangleq Z_t\tilde{h}_{t|t-1} + \varphi_t v_t \quad . \tag{1.17}$$

The algorithm (1.14), can now be expressed as linear filtering of the signal f_t, see Figure 1.6. The signal f_t can be generated from measurements of y_t. Note that η_t, which plays the role of a disturbance, consists of two terms: a vector $\varphi_t v_t$ related to the noise v_t and the term

$$Z_t\tilde{h}_{t|t-1} \quad ,$$

which henceforth will be referred to as the *feedback noise*.

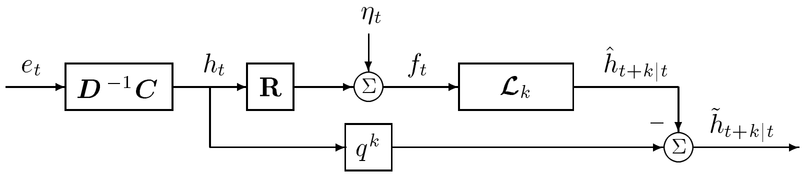

Figure 1.6: A multivariable estimation problem. The parameter vector h_t is rotated by the input correlation matrix $\mathbf{R}$ and distored by gradient noise η_t. The estimator may constitute a filter ($k = 0$), a predictor ($k > 0$) or a fixed lag smoother($k < 0$). The parameter vector is to be estimated from the fictitious measurement signal f_t, so that the covariance matrix of the estimation error $\tilde{h}_{t+k|t}$ is minimized.

The effect of $\tilde{h}_{t|t-1}$ on η_t constitutes a time-varying feedback loop within the algorithm. The approximation made, to reduce the design problem to an open loop filtering problem, is to assume that the correlation between η_t and $\tilde{h}_{t|t-1}$ is negligible. The assumption of near uncorrelatedness is evaluated and investigated in more detail in Chapter 4. Under that assumption, we directly obtain a systematic design methodology, which in the general case consists of solving a polynomial matrix *spectral factorization* and a *bilateral Diophantine equation* iteratively. The result is a stable and rational linear time-invariant estimator of the form

$$\hat{h}_{t+k|t} = \boldsymbol{\mathcal{L}}_k(q^{-1})f_t \quad , \tag{1.18}$$

from which a corresponding gain filter $\boldsymbol{\mathcal{G}}_k(q^{-1})$ in (1.14) can be obtained. The solution minimizes the parameter tracking error covariance matrix. If the regression matrix and the tracking error $\tilde{h}_{t|t-1}$ are assumed independent, then the excess MSE $E|\varepsilon_t|^2 - E|v_t|^2$ can also be minimized.

Different assumptions on the structure of the hypermodel $\boldsymbol{\mathcal{H}}(q^{-1})$ in (1.9) and on the properties of η_t in (1.17) will result in different levels of complexity in the design equations and in the resulting algorithm. The following cases deserve to be mentioned.

1. If no restrictions are placed on $\boldsymbol{\mathcal{H}}(q^{-1})$ and if the gradient noise η_t may be colored, then the gain filter will be a rational matrix. A polynomial matrix spectral factorization and bilateral Diophantine equation will then, in general, have to be solved to optimize the tracking scheme. This general solution is presented in Section 3.4.1.

2. If $\boldsymbol{\mathcal{H}}(q^{-1})$ is general, while η_t is assumed to be *white*, then $\boldsymbol{\mathcal{G}}_k(q^{-1})$ will be a polynomial matrix. In this case, *no Diophantine equation*

needs to be solved. An algebraic Riccati equation will determine the optimal estimator, see Section 3.4.5.

3. To further reduce the complexity of the algorithm, the optimization can be performed under the constraint that
$$\hat{h}_{t+k|t} = \boldsymbol{\mathcal{S}}(q^{-1})\mathbf{R}^{-1} f_t$$
where $\boldsymbol{\mathcal{S}}(q^{-1})$ is a *diagonal* rational matrix. The design equations will then consist of uncoupled sets of scalar spectral factorizations and polynomial Diophantine equations, one for each parameter. If the elements of the parameter vector h_t are uncorrelated[3] and if the elements of η_t are uncorrelated but possibly colored, then this constrained optimal algorithm will attain the unconstrained minimum of the mean square tracking error for constant gain algorithms. In other situations, some performance will be lost. This type of algorithm, presented in Section 3.6.1, will be denoted *Generalized Wiener LMS* (GWLMS).

4. We may assume that the elements of the parameter vector h_t are uncorrelated but have the same dynamics,
$$\boldsymbol{\mathcal{H}}(q^{-1}) \;=\; \frac{C(q^{-1})}{D(q^{-1})}\,\mathbf{I} \;\;, \tag{1.19}$$
and also assume that η_t is white. The optimal gain filter will then reduce to $\boldsymbol{\mathcal{G}}_k(q^{-1}) = Q_k(q^{-1})\mathbf{R}^{-1}$, where $Q_k(q^{-1})$ is a polynomial. The design equations will consist of a single polynomial spectral factorization. No Diophantine equations are required. The resulting algorithm, presented in Section 3.6.2, will be called *Wiener LMS* (WLMS).

5. If it is assumed that the elements of the true parameter vector are uncorrelated but have the same second order autoregressive dynamics,
$$\boldsymbol{\mathcal{H}}(q^{-1}) \;=\; \frac{1}{D(q^{-1})}\,\mathbf{I} \;=\; \frac{1}{1+d_1q^{-1}+d_2q^{-2}}\,\mathbf{I} \tag{1.20}$$
and if η_t is white, then no design equations at all will be required. The optimal algorithm, presented in Section 3.6.3, is obtained immediately and it will be called *Simplified Wiener LMS* (SWLMS).

In all of the above cases, $\boldsymbol{F}(q^{-1})$ from the hypermodel (1.11) is used in (1.14). The algorithms based on case (3),(4), and (5) above can all be seen as generalizations of LMS, with various complexity and structure of the design equations. It should be noted that the designs (3),(4) and (5) neglect

[3] If the dynamics is stable, then such models of communications channels are also known as wide sense stationary uncorrelated scattering models [78].

possible correlations among the parameters. On the other hand, the resulting algorithms require a small number of computations at each time step. If the regressor covariance matrix $\mathbf{R}$ is known, then the number of required multiplications will grow linearly with the dimension of the parameter vector.

If the parameters are known to be significantly mutually correlated, and if η_t is colored, then the general design (1) provides the highest tracking performance. Any of the other alternatives can, of course, be utilized, regardless of what is known about the nature of the parameter variations. If a mismatching structure is deliberately chosen, the user will be aware of what kind of approximation it corresponds to.

Due to the presence of the tracking error in (1.17), the properties of the disturbance η_t will actually depend on the tracking algorithm itself. The optimization problem will therefore have to be solved iteratively. An iterative algorithm for the general case (1) above is presented in Section 3.4.3. A simpler algorithm for case (2) (white η_t) is presented in Section 4.4.2. The simplified designs (4),(5) do instead include a scalar gain parameter, which has to be tuned.

Readers familiar with the polynomial approach to Wiener filtering will have noted the absence of Diophantine design equations in the cases above based on white gradient noise η_t. This simplification has been made possible by the derivation of explicit closed-form solutions to the required equations.

All of the above algorithms can be made robust with respect to uncertainties in the hypermodel, by averaging over a set of possible models. The methodology for robust design is outlined in Section 3.5, and it is exemplified in the case study of Chapter 8.

In **Chapter 4**, the properties of the family of algorithms represented by (1.14) is analyzed. Both stability and performance is considered. The analysis considers both slow and rapid time-variations.

Expressions for the tracking error are first derived. The importance of hypermodels which contain integration is pointed out. The use of integrating models (1.11) will introduce integration into the adaptation law (1.14). This presence of integration is a necessary condition for the bias-free estimation of time-invariant parameters.

A correspondence is then established between the concept of *slow time vari-*

ations and that of a *negligible impact of the feedback noise* $Z_t\tilde{h}_{t|t-1}$ on the tracking error covariance matrix

$$\mathbf{P}_k \triangleq \lim_{t\to\infty} \mathrm{E}\, \tilde{h}_{t+k|t}\tilde{h}^*_{t+k|t} \quad . \tag{1.21}$$

The analysis of the performance of algorithms with the structure (1.13) and (1.14) becomes straightforward if the feedback noise can be neglected. This opens up a novel, simple and direct route to the analysis of adaptive algorithms in the case of slowly time-varying parameters.

The performance of the RLS algorithm and the Kalman algorithm can, for example, be evaluated in a straightforward way. The performance of the RLS and of the Kalman algorithm, when applied to slowly varying parameters with random walk behavior is discussed in Sections 4.3.2 and 4.3.3, respectively. The results are expressions for the mean square tracking error, that are obtained by direct and simple calculations. Analysis of Kalman-based algorithms has, until now, required the use of powerful tools like averaging or weak convergence theory. See e.g. [49].

It must be emphasized here that the design methodology developed in Section 3 is *not* based on an assumption of negligible feedback noise, i.e of slow time variations. Instead, it is based on the assumption that the disturbance η_t in (1.17) is only weakly correlated to the tracking error $\tilde{h}_{t|t-1}$. This is a much weaker condition. It will, of course, be fulfilled in the case of slow time-variations, since the whole feedback noise vector $Z_t\tilde{h}_{t|t-1}$ in (1.17) can then be neglected.

When the feedback noise cannot be disregarded, which corresponds to fast time-variations, the analysis becomes more difficult. Two particular types of problems with fast time-varying parameters are considered in Section 4.4:

- Linear regression models (1.1) with independent consecutive regression vectors containing circular Gaussian data
- FIR systems with zero mean stationary white inputs.

The second case is the most important one from a practical point of view. To conduct the analysis in this case, it will be assumed that $Z_tZ_s^*$, cf (1.15), is almost independent of $\tilde{h}_{t|t-1}\tilde{h}^*_{s|s-1}$. Under this assumption, approximate expressions for the steady state mean square error are obtained. The fourth order moments of the input will affect the performance. It turns out that the kurtosis of the input u_t affects the tracking performance significantly. It is concluded that input data with large kurtosis result in larger tracking errors than input data with a smaller value of the kurtosis. Thus, the use of binary input signals to a FIR model will result in a better tracking performance than the use of Gaussian inputs. The impact of the kurtosis has also been noted in the works [38] and [39] by Gardner. The tracking performance does furthermore seem to be more sensitive to the choice of step-size for input signals with a large kurtosis.

Finally, in Section 4.4 we will address the question if there is any hope of attaining an exact performance analysis, without approximations, for fast time-varying parameters, i.e. for a non-negligible feedback noise. In most circumstances, the problem will then become so difficult that an exact analysis seems to be out of the question. However, it is conjectured that an exact analysis *is* possible for the WLMS algorithm applied to FIR systems with scalar output and white circular complex-valued inputs with zero mean and constant modulus.

It is proven in Result 4.3 that an exact analysis is indeed possible for second order FIR models (1.5) with circular, white and zero mean input data with constant modulus. By a fortunate coincidence, these rather special conditions happen to be fulfilled in the channel estimation problem introduced in Example 1.1 and studied in detail in Chapter 8. Thus, the MSE performance of WLMS-type channel estimators can in Chapter 8 be described exactly by analytical expressions.

In **Chapter 5,** the analysis is specialized to the LMS algorithm, which is the most widely utilized and analyzed of all tracking algorithms. The input-output polynomial approach introduced in Chapter 3 provides a new approach to the analysis of LMS adaptation laws. As a result, many well-known properties of the LMS algorithms can be derived in a very simple way. Several new results, such as improved conditions for convergence in MSE (Result 5.3 and 5.4) and an improved expression for the optimal step-size (Result 5.5) are also derived and discussed.

The aim of Chapters 6 to 8 is primarily to investigate the utility of the proposed hypermodel-based framework for tracking algorithm design. Although the proposed framework can potentially be utilized in many applications, Digital Communications is particularly suitable for purpose of illustration.

In **Chapter 6**, passband and baseband descriptions of mobile radio channels are introduced. A key element in the successful use of hypermodel-based estimators is the existence of reasonably accurate *a priori* information on the nature of the time-variations. Such information exists for mobile radio channels, in the form of models for the fading baseband channel parameters. A frequently utilized model for fading channels, namely Jakes model, is discussed.

In **Chapter 7**, the symbol estimation part of the adaptive equalization problem, that was introduced in Example 1.1, will be discussed. Methods for direct and indirect adaptation of detectors are reviewed. Motivated by the highly time-varying nature of the channel coefficients in the case study of Chapter 8, a time-variable transversal linear equalizer and a novel time-variable decision feedback equalizer are derived in Section 7.3.1. A brief discussion on the implementation aspects of these algorithms is also included in Section 7.3.2. Finally, the adaptive implementation of the Viterbi detector is discussed. It is noted that prediction estimates of the channel coefficients are required in this algorithm.

The thesis is concluded by an extensive case study in **Chapter 8**. The case study involves channel estimation and equalization of a fading mobile radio channel, originating from the D-AMPS, or IS-54 standard (Digital Advanced Mobile Phone System). This standard is presently the dominating standard for digital mobile communications in North America, and a similar standard is used in Japan. In Chapter 8, the physical insights obtained from Chapter 6 and the time-varying DFE derived in Chapter 7, are combined with the proposed hypermodel-based tracking algorithm design from Chapter 3. A theoretically founded comparison between hypermodel-based designs and LMS is performed. Finally, the bit error rate and tracking error performance of different adaptive equalization schemes are compared in a simulation study.

We will conclude this section by illustrating how the simplest one of the suggested fixed-gain algorithms, namely SWLMS, compares to LMS in the channel estimation problem introduced in Example 1.1.

Example 1.2 *Tracking of parameters of fading mobile radio channels in TDMA systems, continued*

In this example, we shall compare the attainable tracking performances of the SWLMS algorithm and the LMS algorithm, in a situation where the time variations of the parameters are known to differ substantially from random walks. In the D-AMPS (IS 54) system, the wireless channel is subject to fading, which can be modeled as Rayleigh fading. The channel coefficients are rapidly time-varying. An adequate description of the wireless channel is given by

$$y_t = h_t^0 u_t + h_t^1 u_{t-1} + v_t \quad , \tag{1.22}$$

where the coefficients h_t^0 and h_t^1 are subject to Rayleigh fading, u_t are QPSK modulated symbols of variance 2 and v_t represents white noise and co-channel interference. All signals in (1.22) are complex valued and stationary with zero mean. The noise/interference will be independent of the symbols and the channel coefficients, so Assumption 1.1 will be fulfilled.

The LMS algorithm is given by

$$\hat{h}_{t+1|t} = \hat{h}_{t|t-1} + \mu\varphi_t\varepsilon_t \tag{1.23}$$

where μ is the step-size. The SWLMS algorithm is, for one-step predictors, given by

$$\hat{h}_{t+1|t} = -(d_1 + d_2q^{-1})\hat{h}_{t|t-1} + \mu\left(\frac{-d_1}{1+d_2(1-\mu)} - d_2q^{-1}\right)\frac{1}{\sigma_u^2}\varphi_t\varepsilon_t \tag{1.24}$$

where d_1 and d_2 are the coefficients of the polynomial

$$D(q^{-1}) = 1 + d_1q^{-1} + d_2q^{-2}$$

in the hypermodel (1.20) and where σ_u^2 is the variance of the regressor u_t. In order to compare these algorithms, we shall regard (1.22) as the “true system”, from which we (by simulation) obtain the received signal. To generate the Rayleigh fading coefficients, Jakes' model is used [56]. (For details, see Chapter 6.) The covariance function of the channel coefficients is then described by

$$\mathrm{E}\, h_{t+\ell}^i h_t^{i*} = J_0(\Omega_{\mathrm{D}}\ell) \qquad i = 0, 1 \tag{1.25}$$

if the coefficients both have unit variance. Here, $J_0(\cdot)$ is the Bessel function of first kind and zero order, and Ω_{D} is the maximum normalized Doppler frequency. We shall here study a case with a mobile traveling at a speed of 100 km/h. In the D-AMPS system, with a sampling period of 40μs and 900 MHz carrier frequency, this corresponds approximately to $\Omega_{\mathrm{D}} = 0.02$. For the SWLMS algorithm we shall use a model of the same structure as (1.22). We describe h_t by a second order autoregressive (AR) hypermodel with poles located close to the unit circle, at an angle related to the Doppler frequency as $\Omega_{\mathrm{D}}/\sqrt{2}$. The use of pole locations corresponding to resonant modes is motivated by the quasi-periodic behavior of channel coefficients subject to Rayleigh fading, according to Jakes' model. (See also Figure 1.4 in Example 1.1.) The pole angle is selected so that a good fit between the Bessel function (1.25) and the covariance function resulting from the hypermodel is obtained.

Thus, we select a hypermodel in transfer function form, which for $\Omega_{\mathrm{D}} = 0.02$ is given by

$$\begin{aligned}\boldsymbol{\mathcal{H}}(q^{-1}) &= \frac{1}{D(q^{-1})}\,\mathbf{I}_2 \\ D(q^{-1}) &= 1 - 0.9978\,q^{-1} + 0.9980\,q^{-2} \ .\end{aligned} \tag{1.26}$$

For a signal-to-noise ratio of 15 dB and a normalized Doppler frequency of $\Omega_{\mathrm{D}} = 0.02$, the optimized SWLMS algorithm will then be given by

$$\hat{h}_{t+1|t} = (1.9978 - 0.9980\,q^{-1})\hat{h}_{t|t-1} + (0.0942 - 0.0898\,q^{-1})\frac{1}{2}\varphi_t\varepsilon_t \ . \tag{1.27}$$

The LMS algorithm with step-size optimized for the above case is given by

$$\hat{h}_{t+1|t} = \hat{h}_{t|t-1} + 0.1\varphi_t\varepsilon_t \ . \tag{1.28}$$

The properties of these tracking algorithms and their influence on the symbol error rate when used in conjunction with a Viterbi detector in an adaptive equalizer, is depicted in Figure 1.7 below. In the left-hand figures, the tracking performance is illustrated by simulations of 600 symbol times, using known transmitted symbols as regressors. It can be mentioned that the theoretical expressions developed in Chapter 3 and Chapter 4 can exactly predict the mean square tracking error as a function of μ for known regressors. These functions are depicted in the upper right-hand part of Figure 1.7.

As can be seen from the upper right-hand figure, the attainable MSE tracking performance is improved almost three times by the SWLMS algorithm. The improvement of the bit error rate obtained by using the SWLMS algorithm is approximately 2 times at 15 dB and 7 times at 25 dB. This corresponds to a gain of between 3 and 5 dB. We conclude that a considerable improvement can be obtained by simple means in cases when the parameter variations differ substantially from random walk behavior.

The reason for the improved performance can be understood by inspecting the frequency responses of the tracking filters (1.18), shown in Figure 1.8. The SWLMS tracking filter possesses bandpass character around the dominating frequencies of the time-variations illustrated in Figure 1.4, and it has lower gain at high frequencies.

□

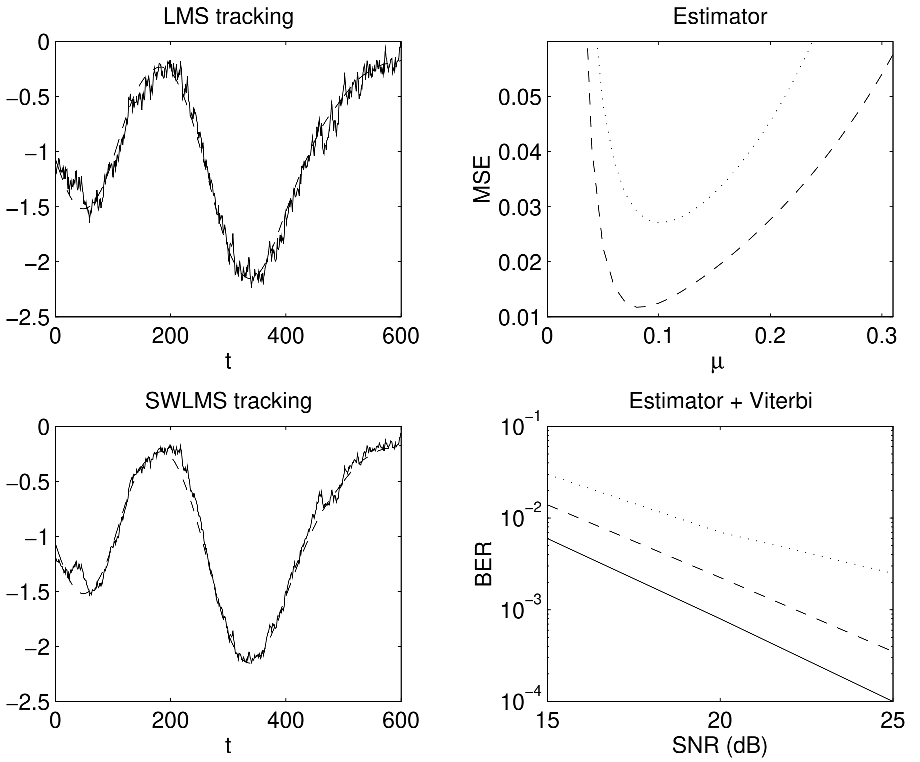

Figure 1.7: The result of LMS and SWLMS parameter estimation in Example 1.2. Top and bottom left figures, estimates (solid) and the true parameter variation (dashed). Top right, the MSE tracking performance for the LMS algorithm (dotted) and for the SWLMS algorithm (dashed), evaluated for 15 dB SNR and $\Omega_{\mathrm{D}} = 0.02$ (100 km/h). Bottom right, bit error rate for estimator LMS (dotted) and SWLMS (dashed), concatenated with a Viterbi algorithm, and the bit error rate obtained by using true channel parameters (solid).

1.3 Summary of contributions

The thesis offers a novel alternative approach to the design and the analysis of tracking algorithms for linear regression models. The major contributions of the work can be summarized as follows.

Design of adaptive algorithms

A design methodology has been developed for adaptation laws (1.14) with time-invariant gains. The approach is based on solving an approximating Wiener filter design problem by means of the polynomial equations approach. It has the following main characteristics.

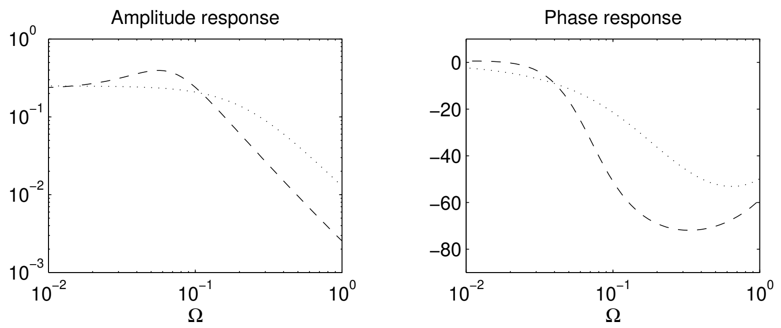

Figure 1.8: Amplitude and phase response of the learning filters in Example 1.2, corresponding to SWLMS tracking (dashed) and to LMS tracking (dotted), as a function of the normalized frequency Ω

- The designs are based on the minimization of the mean square parameter tracking error. *A priori* information on the dynamics of the time-variations, in the form of linear time-invariant stochastic models of the parameter variations, can be taken into account.
- Methods for robust design based on the minimization of the mean square tracking error, averaged over possible models of the parameter variations, have been developed.
- Filtering, prediction and fixed lag smoothing estimators can all be designed by utilizing the same basic equations.
- The methodology offers a selection of algorithms of different complexity. There is a direct connection between the complexity of the assumed hypermodel and the complexity of the resulting algorithm. Simpler models result in simpler design equations and in tracking algorithms with a lower computational complexity.

If the present work is compared to the work by Kubin [64],[65] on the use of coefficient prediction filters, it can be said to offer a methodology for the systematic design of coefficient prediction filters.

The work by Benveniste and co-workers [10],[11],[12] was also aimed at the use of hypermodels in the design of adaptation algorithms with time-invariant gains. Compared to that approach, the methodology presented here can be claimed to be simpler to use. It also provides algorithms with lower complexity, provides for robust design and it is not restricted to situations with slowly time-varying parameters.

New theory for polynomial systems design

The design methodology is based on the polynomial systems framework to multivariable Wiener filtering. It provides a new theory for approaching the design of adaptive algorithms within this framework. The thesis also includes two types of results of general interest, related to the design of Wiener filters.

- The existence of closed-form solutions to bilateral Diophantine equations for filtering problems with white noise (Lemma 3.2)
- A recursion in the prediction length k for obtaining estimators for different prediction horizons and smoothing lags (Corollary 3.1). Estimators for differing values of k are then obtained directly, without any need to re-solve Diophantine design equations.

Furthermore, a way to obtain the polynomial matrix spectral factor from an algebraic Riccati equation in the white noise case is presented as Result 3.3.

Analysis of adaptive algorithms

The algorithm structures (1.13) and (1.14) may provide a unifying framework for the understanding and the analysis of a large number of existing and novel algorithms.

A novel approach to the analysis of adaptation laws for linear regression models with slowly time-varying parameters is presented. It is based on the concept that slow time variations correspond to a negligible feedback noise $Z_t\tilde{h}_{t|t-1}$ in (1.17). This principle has been applied to the analysis of LMS, RLS, Kalman, algorithms and to algorithms with the general structure (1.14). Analysis based on assumptions of slowly time-varying parameters has had a long history, see e.g. [12] and [69]. The approach presented here simplifies this type of analysis significantly.

It is also explained why the quite restrictive assumption of independent consecutive regression vectors can lead to results which provide reasonable predictions on the behavior of adaptive algorithms. This assumption has been a standard tool in the analysis of LMS algorithms [32],[39],[53], [133],[136]. In the present framework, an assumption of independent consecutive regression vectors corresponds to an assumption of white feedback noise, and to independence between Z_t and the parameter estimation error $\tilde{h}_{t|t-1}$.

An approximate analysis is also performed in Chapter 4 and 5 for situations with fast time-variations, i.e for situations where the feedback noise cannot be neglected. The analysis is mainly performed for scalar FIR systems with white inputs of zero means.

Digital Communications

In Chapter 7, a MSE-optimal decision feedback equalizer has been derived for an exactly known time-varying FIR channels, under the assumption of correct past decisions.

In Chapter 8, a rather detailed case study is performed on channel tracking and adaptive equalization for the North American digital mobile radio standard IS-54. It is concluded that the methodology developed in the present thesis can in that application provide adaptive equalizers with both high performance and a low computational complexity.