JUST IN TIME MODELS FOR DYNAMICAL SYSTEMS

Anders Stenman, Fredrik Gustafsson, and Lennart Ljung
stenman@isy.liu.se, fredrik@isy.liu.se, ljung@isy.liu.se

Department of Electrical Engineering
Linköping University
S-581 83 Linköping
Sweden

Abstract

The concept of *just in time* models is introduced for models that are not estimated until they are really needed. The idea is to store all observations of the process in a database, and then estimate a local model at the current working point. The variance/bias tradeoff is optimized locally by adapting the number of data and their relative weighting. This is in contrast to general non-linear black-box models, like neural networks, where the performance is optimized globally.

1 Introduction

Consider a non-linear dynamical system described by

$$\begin{cases} z(t) = f(\varphi(t)) \\ y(t) = z(t) + e(t) \end{cases} \tag{1}$$

Here $e(t)$ is measurement noise and the regression vector $\varphi(t)$ typically consists of lagged inputs and outputs for dynamical systems, but could also be or include a working point vector. The problem considered here is to find a local model to be used for control or prediction. A standard approach is to divide the $\varphi(t)$-space into a number of regions and once for all compute a model or controller in each of them. This is usually referred to as *gain scheduling* in the literature [2]. Inspired and encouraged by recent progress in database technology, we will take a conceptually different point of view. We assume that all past data are stored in a database, and for each new $\varphi(t)$ we observe, a new estimate $\hat{z}(t)$ is computed. For this concept, we have adopted the name *just in time* models, which was perhaps first suggested by G. Cybenko [4]. Compared to global models, like neural nets or other non-linear black-box model structures [10], the advantage is that the prediction is optimized locally and not globally, which might increase the performance. A possible drawback is computational complexity, but a large neural network takes time to evaluate, not to mention the training time.

It is clear that only regressors $\varphi(k)$ in a neighborhood of $\varphi(t)$ are relevant for predicting $z(t) = f(\varphi(t))$. The question is how to define neighborhoods and how to form the prediction from observed $(y(k), \varphi(k))$ pairs. We will assume that there are a huge number of data available, for a typical industrial process it may occupy in the order of tens of Gigabytes. For this reason, it is important to develop efficient data structures enabling quick search for a neighborhood of $\varphi(t)$. In this contribution, we will investigate only the modeling issue.

We assume that the predictor is formed as a weighted mean

$$\hat{z}(t) = \sum_{k=-\infty}^{t-1} w(k)y(k) \tag{2}$$

This model choice is based on the assumption that $f(\varphi(t))$ is linear locally around $\varphi(t)$. Intuitively, it is clear that the weights $w(k)$ should depend on the distance $\varphi(k) - \varphi(t)$ only, so we write $w(k) = w(\|\varphi(k) - \varphi(t)\|)$. The remaining question is which norm to use and how to choose the weights $w(d)$, where d is the distance. There is an inherent tradeoff in that a small neighborhood gives small bias but large variance, while a large neighborhood makes the assumption of a linear $f(\varphi(t))$ inappropriate and introduces a bias.

We propose a two-step solution:

1. Make a Taylor expansion of $f(\varphi(t))$ in (1). Estimate the Hessian with the least squares method using the N closest regression vectors, where N is chosen by Akaike's FPE method.
2. The optimal coefficients $w(k)$, that minimize $\mathrm{E}(z(t) - \hat{z}(t))^2$ are functions only of the Hessian matrix and $\varphi(t) - \varphi(k)$. Use (2) and the estimate of the Hessian term from step one.

An estimate of $z(t)$ may be taken from step 1 directly, but we believe that the second step increases the accuracy. To our knowledge, the concept of just in time models is new in the control community. However, local models in the same fashion are well-known in the statistical literature, although there always seem to be a global optimization in some step.

The outline is as follows: A review of related statistical methods is given in section 2. Section 3 defines the problem. Sections 4, 5, and 6 discuss the problems of weight computation, Hessian estimation, and optimal data selection respectively. Section 7 provides a summary while section 8 shows some illustrating examples.

2 Local models

Local models and local non-parametric regression models have been discussed and analyzed in the statistical literature the last two decades, starting with [11] and [3]. A common theme is the use of Kernel functions, which is a standard approach in time series analysis and probability density function estimation. The coefficients are chosen *a priori* to follow a Kernel function $w(k) = \frac{1}{h}K\left(\frac{\varphi(t)-\varphi(k)}{h}\right)$. In this formulation, there is only one degree of freedom, to choose the *bandwidth* h. The bandwidth is usually optimized in a global fashion, for instance by minimizing $E\int(f(\varphi)-\hat{f}(\varphi))^2 g(\varphi)\mathrm{d}\varphi$ for some function $g(\varphi)$. That is, the concept of neighborhood for the local function is derived globally, in contrast to our approach. A multivariable Kernel is proposed in [9], where the Kernel is given by $|H|^{-1/2}K(H^{-1/2}\varphi)$.

3 Preliminaries

In the sequel of the paper, we drop the time index and use a more general index i. We assume that the stored data $\varphi_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$ obeys the relation

$$\left\{\begin{array}{l} z_i = f(\varphi_i) \\ y_i = z_i + e_i, \end{array}\right. \qquad (3)$$

where $f(\cdot)$ is a nonlinear mapping from $\mathbb{R}^n$ to $\mathbb{R}$, and $\{e_i\}$ is white noise with zero mean and variance λ. Given a new value of φ, say φ_0, we now want to determine $z_0 = f(\varphi_0)$. Under the assumption that $f(\varphi_i)$ is approximately linear in a neighborhood of φ_0, it is natural to consider a linear model structure, and to estimate it with a weighted least squares criterion. That is, z_0 is estimated as

$$\begin{aligned} \hat{z}_0 &= \tilde{\varphi}_0^T\hat{\theta} \\ \hat{\theta} &= \arg\min_{\theta} \sum_{\varphi_i\in\Omega_N} w_i(y_i - \tilde{\varphi}_i^T\theta)^2. \end{aligned} \qquad (4)$$

Here $\tilde{\varphi}_i = (1\ \varphi_i^T)^T$, arg min means "the minimizing argument of the function", Ω_N denotes a neighborhood of φ_0 containing N data, and w_i is a weighting function that assigns different measurements different weights in the criterion.

The solution to (4) is

$$\hat{z}_0 = \tilde{\varphi}_0^T\left[\sum_{\varphi_i\in\Omega_N} w_i\tilde{\varphi}_i\tilde{\varphi}_i^T\right]^{-1}\sum_{\varphi_i\in\Omega_N} w_i\tilde{\varphi}_i y_i. \qquad (5)$$

It is convenient to assume that the weights satisfies the constraints

$$\begin{aligned} \sum_{\varphi_i\in\Omega} w_i &= 1, & (6) \\ \sum_{\varphi_i\in\Omega} w_i(\varphi_i - \varphi_0) &= 0, & (7) \end{aligned}$$

which are standard for kernel estimation and smoothing windows in spectral analysis [5]. As shown in Appendix A, (5) simplifies to

$$\hat{z}_0 = \sum_{\varphi_i\in\Omega_N} w_i y_i. \qquad (8)$$

Two major questions arise:

- Which are the optimal weights w_i?
- Which data should be used in the estimation, i.e. what is the optimal number of data N in Ω_N, and what is the optimal shape of Ω_N?

4 Optimal weights

In this section we derive an expression for the optimal weights w_i in (8). We thus want to minimize the mean square prediction error (MSE),

$$W(w) = E(\hat{z}_0 - z_0)^2, \qquad (9)$$

subject to the given constraints on the weights.

Inserting (3) and (8) in (9) gives

$$\begin{aligned} W(w) &= E\left[\sum_{\varphi_i\in\Omega_N} w_i y_i - f(\varphi_0)\right]^2 \\ &= E\left[\sum_{\varphi_i\in\Omega_N} w_i(f(\varphi_i) - f(\varphi_0) + e_i)\right]^2 \\ &= E\left[\sum_{\varphi_i\in\Omega_N} w_i(\beta_i + e_i)\right]^2 \\ &= (\beta^T\omega)^2 + \lambda\cdot\omega^T\omega, & (10) \end{aligned}$$

where

$$\omega = (w_1, w_2, \ldots, w_N)^T, \tag{11}$$

and

$$\beta = (\beta_1, \beta_2, \ldots, \beta_N)^T. \tag{12}$$

Now minimize (10) subject to the constraints

$$g(w) = \omega^T \mathbf{1} - 1, \quad \mathbf{h}(w) = \Phi^T \omega - \varphi_0, \tag{13}$$

where

$$\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_N)^T. \tag{14}$$

By introducing the Lagrange multipliers μ and γ_j we thus obtain

$$\frac{\partial W(w)}{\partial w_i} + \mu \frac{\partial g(w)}{\partial w_i} + \sum_{j=1}^{n} \gamma_j \frac{\partial h_j(w)}{\partial w_i} = 0, \quad \forall w_i \tag{15}$$

where $h_j(w)$ denotes the jth component of the vector valued function $\mathbf{h}(w)$. The constrained minimization of $W(w)$ can then be stated in matrix form as

$$\begin{pmatrix} 2(\beta\beta^T + \lambda I) & \mathbf{1} & \Phi \\ \mathbf{1}^T & 0 & \mathbf{0} \\ \Phi^T & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \omega \\ \mu \\ \gamma \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \\ \varphi_0 \end{pmatrix}, \tag{16}$$

where $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)^T$. This is a linear equation system in $N + n + 1$ equations and $N + n + 1$ variables, which yields that the w_i's can be uniquely solved. The β_i's in (12) are related to the Hessian (2nd order derivative matrix) in the following way. Make a Taylor expansion of $f(\cdot)$ at φ_0

$$f(\varphi_i) \approx f(\varphi_0) + J(\varphi_i - \varphi_0) + \frac{1}{2}(\varphi_i - \varphi_0)^T H(\varphi_i - \varphi_0) \tag{17}$$

Here J and H denote the gradient and Hessian of $f(\cdot)$ at φ_0 respectively. From the constraint (7) and (10) it follows that

$$\beta_i \approx \frac{1}{2}(\varphi_i - \varphi_0)^T H(\varphi_i - \varphi_0). \tag{18}$$

In order to compute the optimal weights w_i we thus need to know the Hessian H. It is normally unknown, but can be estimated from data as shown in section 5.

5 Estimation of Hessian

Since J and H enter (17) linearly, H can be estimated by least squares theory as follows.

Introduce the vector

$$\alpha_i = \varphi_i - \varphi_0, \tag{19}$$

and define

$$\vartheta = (f, J, \mathrm{vech}^T H)^T, \tag{20}$$

$$\phi_i = (1, \alpha_i^T, \mathrm{vech}^T(\alpha_i \alpha_i^T - \frac{1}{2}\mathrm{diag}(\alpha_i \alpha_i^T)))^T, \tag{21}$$

where vech H denotes the *vector-half* of the matrix H, i.e. the $\frac{1}{2}n(n+1) \times 1$ vector obtained by stacking the columns of H underneath each other and eliminating the above-diagonal entries, and diag H is the same as H, but with all off-diagonal entries equal to zero.

We can now state the estimation of H (and f and J) as a least squares problem,

$$V_N(\vartheta, \Omega_N) = \sum_{\varphi_i \in \Omega_N} (y_i - \phi_i^T \vartheta)^2 \tag{22}$$

$$\hat{\vartheta} = \arg\min_{\vartheta} V_N(\vartheta, \Omega_N). \tag{23}$$

6 Data selection

One question remains to be answered; what is the optimal size and shape of the neighborhood Ω_N used in (22) and (23)?

6.1 Region size

When computing $\hat{\vartheta}$ in (22) and (23) it is clear that a small neighborhood would give the measurement noise a large influence on the resulting estimate. On the other hand, a large neighborhood would make the Taylor expansion (17) inappropriate and would introduce a bias. The optimal data size is thus the standard trade-off between bias error and variance error. A commonly used approach in system identification is to evaluate the loss function (22) on completely new datasets Ω_N', and choose N_{opt} as the N that minimizes $V(\hat{\vartheta}, \Omega_N')$. That is,

$$N_{opt} = \arg\min_N V(\hat{\vartheta}, \Omega_N') = \arg\min_N \sum_{\varphi_i \in \Omega_N'} (y_i - \phi_i^T \hat{\vartheta})^2. \tag{24}$$

However, in this case we are forced to evaluate V_N on the same data Ω_N as used in estimation. A number of methods has been developed that instead use the value $V_N(\hat{\vartheta}, \Omega_N)$ as a basis of an estimate of what we would have obtained if we had applied the evaluation on fresh data. One such method is Akaike's FPE (Final Prediction Error) [1],

$$W_N^{FPE} = V_N(\hat{\vartheta}, \Omega_N) \frac{1 + d/N}{1 - d/N}, \tag{25}$$

where $d = \dim \vartheta = 1 + n + \frac{1}{2}n(n+1)$.

We thus have a method of determining the region size N_{opt} as

$$N_{opt} = \arg\min_N V_N(\hat{\vartheta}, \Omega_N) \frac{1 + d/N}{1 - d/N}. \tag{26}$$

Note that this function is minimized w.r.t. N, and not w.r.t. the number of parameters d as usual in the area

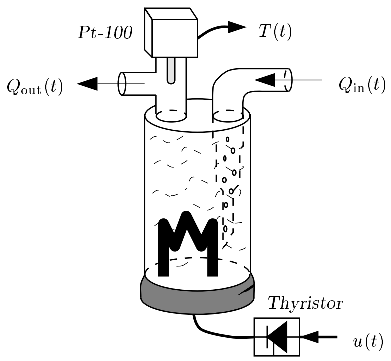

Figure 1: The water heating process.

of model structure selection as is its originally intended application.

In [3] and [8], the related Mallow's C_p criterion is used to get a good bias/variance trade-off.

6.2 Region shape

We have so far just considered Euclidian norms when retrieving the N closest regression vectors φ_i that defines Ω_N. This leads to a spherical shape of Ω_N. Another possibility though, could be to use a norm that adapts to the properties of data. Then Ω_N would have the shape of an ellipsoid. Such scaling is very important when the signal components in φ_i have different magnitudes.

7 The algorithm

We propose a two-step algorithm which can be summarized as follows:

① Make a Taylor expansion of $f(\varphi_i)$ at φ_0. Estimate the Hessian with the least squares method (23), using the N in Euclidian distance closest regression vectors. N is chosen by Akaike's FPE method (26).

② Use the result from step ① and (16) to compute weights w_i, and form the resulting estimate as a weighted mean of the outputs, i.e. $\hat{z}_0 = \sum w_i y_i$.

The estimate from step ① can be used by itself, but step ② may increase the accuracy when the function $f(\cdot)$ is not pure quadratic.

8 Examples

8.1 A water heating process

In this example we consider identification of a water heating process as depicted in Figure 1. This process has earlier been investigated in [6] and [7]. The water is heated by a resistor element which is controlled by the voltage $u(t)$. At the outlet, the water temperature $T(t)$ is measured. The inlet flow $Q_{in}(t)$ as well as the inlet water temperature is assumed to be constant. The modeling problem is to describe the temperature $T(t)$ given the voltage $u(t)$.

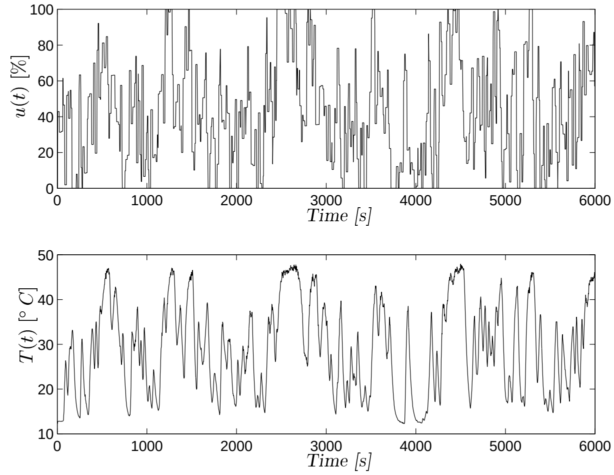

Figure 2: Heater estimation data

The dataset consists of 3000 samples, recorded every 3rd second. This set was divided into an estimation set of 2000 samples, see Figure 2, and a validation set of 1000 samples. The time delay from input to output is between 12 to 15 seconds, [6]. This yields that useful regressors stemming from the input are $u(t-4)$, $u(t-5)$ and so on.

We apply our algorithm to the heater data using a second order model, i.e. the regression vector is defined as $\varphi(t) = (T(t-1), T(t-2), u(t-4), u(t-5))^T$. Since the orders are $n_a = 2$, $n_b = 2$, and the delay is $n_k = 4$, we call this a JIT224 model. We then let the estimation dataset define our observation database, and use the voltage signal $u(t)$ in the validation dataset to obtain a simulation of the corresponding temperature $T(t)$. For comparison we also tried a linear ARX224 model. The result of the simulations is shown in Figure 3. Lindskog [7] has achieved a RMS of 1.02 using the fuzzy modeling approach.

The larger prediction error in the ARX model is due to nonlinearities in the thyristor.

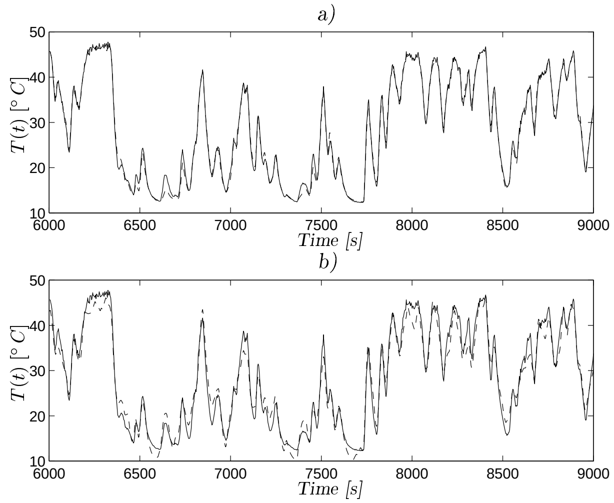

Figure 3: Simulation results (*Solid*: Measured output, *Dashed*: Simulated output). **a)** Just in time JIT224 model, RMS: 0.8860. **b)** Linear ARX224 model, RMS: 2.0821.

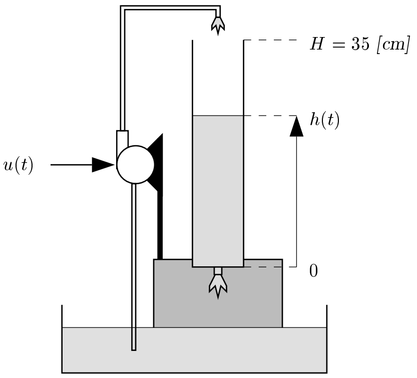

Figure 4: A laboratory-scale tank system.

8.2 Tank level modeling

In this example we will use our algorithm to model the liquid level $h(t)$ of a laboratory-scale tank system as depicted in Figure 4. The system has earlier been investigated in [7]. Two data records of 1000 samples each, one for estimation and one for validation, were available. A plot of the estimation dataset is shown in Figure 5. The regression vector was defined as $\varphi(t) = (h(t-1), u(t-1)^T$, i.e. a JIT111 structure.

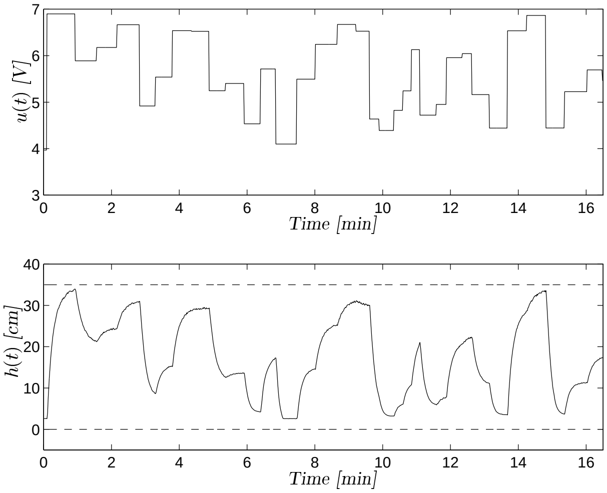

Figure 5: Tank estimation data.

A simulation using JIT111 and ARX111 model structures, and the voltage signal $u(t)$ from the validation dataset, gives the result as shown in Figure 6. Again the larger error in the ARX model is due to saturation in the $u(t)$ signal.

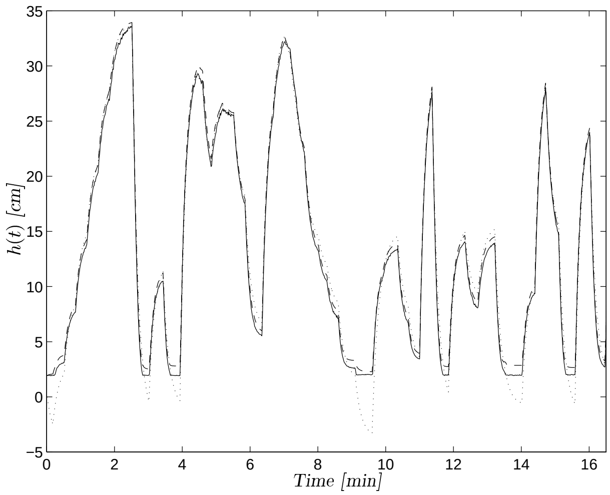

Figure 6: Result of simulation. *Solid*: Measured output, *Dashed*: Simulated output JIT111, *Dotted*: Simulated output ARX111.

9 Conclusions

The problem of system modeling utilizing data stored in a database has been studied. A *just in time* model is estimated only when needed and locally around the current working point by minimizing the expected error which leads to a good bias/variance tradeoff. A two step algorithm was proposed. In a first step the remainder term in a Taylor expansion is estimated from the N closest regression vectors, where N is given by the *Final Prediction Error* criterion. In the second step, an optimal data weighting is computed to minimize the mean square error. This weighting is a function of the remainder term, which is replaced by its estimate from step one. Two examples illustrated the algorithm.

References

[1] H. Akaike. Fitting autoregressive models for prediction. *Ann. Inst. Statist. Math.*, 21:243–247, 1969.

[2] K.J. Åström and B. Wittenmark. *Adaptive Control.* Addison Wesley, 1989.

[3] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 83:596–610, 1988.

[4] G. Cybenko. Just-in-time learning and estimation. In Sergio Bittani and Giorgio Picci, editors, *Identification, Adaption, Learning*, NATO ASI series, pages 423–434. Springer, 1996.

[5] S.M. Kay. *Modern Spectral Estimation.* Prentice Hall, 1988.

[6] H. Koivisto. *A Practical Approach to Model Based Neural Network Control.* PhD thesis, Tampere University of Technology, Tampere, Finland, December 1995.

[7] P. Lindskog. *Methods, Algorithms and Tools for System Identification Based on Prior Knowledge.* PhD thesis, Linköping University, Linköping, Sweden, May 1996.

[8] D. Ruppert, S.J. Sheather, and M.P. Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90, 1995.

[9] D. Ruppert and M.P. Wand. Multivariate locally weighted least squares regression. *Annals of Statistics*, 22, 1994.

[10] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31:1691–1724, 1995.

[11] C.J. Stone. Consistent nonparametric regression. *The annals of statistics*, 5:595–620, 1977.

A Derivation of (8)

From (5) we have

$$\begin{aligned}
\hat{z}_0 &= \tilde{\varphi}_0^T \left[\sum_i w_i \tilde{\varphi}_i \tilde{\varphi}_i^T\right]^{-1} \sum_i w_i \tilde{\varphi}_i y_i \\
&= \begin{pmatrix} 1 & \varphi_0^T \end{pmatrix} \left[\sum_i w_i \begin{pmatrix} 1 & \varphi_i^T \\ \varphi_i & \varphi_i \varphi_i^T \end{pmatrix}\right]^{-1} \sum_i w_i \begin{pmatrix} 1 \\ \varphi_i \end{pmatrix} y_i \\
&= \begin{pmatrix} 1 & \varphi_0^T \end{pmatrix} \begin{pmatrix} 1 & \varphi_0^T \\ \varphi_0 & \sum_i w_i \varphi_i \varphi_i^T \end{pmatrix}^{-1} \sum_i \begin{pmatrix} w_i y_i \\ w_i \varphi_i y_i \end{pmatrix}
\end{aligned} \tag{27}$$

where the third equality follows from the constraints (6) and (7). The matrix inverse in (27) can be rewritten using the block matrix inversion formula. This gives

$$\begin{pmatrix} 1 & \varphi_0^T \\ \varphi_0 & \sum_i w_i \varphi_i \varphi_i^T \end{pmatrix}^{-1} = \begin{pmatrix} 1 + \varphi_0^T \Delta^{-1} \varphi_0 & -\varphi_0^T \Delta^{-1} \\ -\Delta^{-1} \varphi_0 & \Delta^{-1} \end{pmatrix} \tag{28}$$

where

$$\Delta = \sum_i w_i \varphi_i \varphi_i^T - \varphi_0 \varphi_0^T \tag{29}$$

(28) inserted in (27) finally gives

$$\begin{aligned}
\hat{z} &= \begin{pmatrix} 1 & \varphi_0^T \end{pmatrix} \begin{pmatrix} 1 + \varphi_0^T \Delta^{-1} \varphi_0 & -\varphi_0^T \Delta^{-1} \\ -\Delta^{-1} \varphi_0 & \Delta^{-1} \end{pmatrix} \begin{pmatrix} \sum_i w_i y_i \\ \sum_i w_i \varphi_i y_i \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \sum_i w_i y_i \\ \sum_i w_i \varphi_i y_i \end{pmatrix} = \sum_i w_i y_i
\end{aligned} \tag{30}$$