

Blüschke-Nikolaeva, Viktoria; Blüschke, Dmitri; Neck, Reinhard

Conference Paper

Optimal Control of Nonlinear Dynamic Econometric Models: An Algorithm and an Application

Beiträge zur Jahrestagung des Vereins für Socialpolitik 2010: Ökonomie der Familie - Session: Computational Econometrics, No. A3-V3

Provided in Cooperation with:

Verein für Socialpolitik / German Economic Association

Suggested Citation: Blüschke-Nikolaeva, Viktoria; Blüschke, Dmitri; Neck, Reinhard (2010) : Optimal Control of Nonlinear Dynamic Econometric Models: An Algorithm and an Application, Beiträge zur Jahrestagung des Vereins für Socialpolitik 2010: Ökonomie der Familie - Session: Computational Econometrics, No. A3-V3, Verein für Socialpolitik, Frankfurt a. M.

This Version is available at:

<https://hdl.handle.net/10419/37494>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Optimal Control of Nonlinear Dynamic Econometric Models: An Algorithm and an Application

V. Blüschke-Nikolaeva^a, D. Blüschke^a, R. Neck^{a*}

^aDepartment of Economics, Klagenfurt University
Universitätsstrasse 65-67, A-9020 Klagenfurt, Austria

February 22, 2010

Abstract In this paper, we present a new version of the OPTCON algorithm for the optimal control of nonlinear stochastic systems with special reference to econometric models. It delivers approximate numerical solutions of optimum control problems with a quadratic objective function for nonlinear econometric models with additive and multiplicative (parameter) uncertainties. The algorithm was programmed in *C#* and allows for deterministic and stochastic control, the latter with open-loop and passive learning (open-loop feedback) information patterns. We demonstrate the applicabil-

*Corresponding author: Prof. Reinhard Neck, Department of Economics, Klagenfurt University, Universitätsstrasse 65-67, A-9020 Klagenfurt, Austria/Europe, Tel.: +43 463 2700 4121, Fax: +43 463 2700 994121, Email: Reinhard.neck@uni-klu.ac.at

ity of the algorithm by experiments with a small quarterly macroeconomic model for Slovenia. This shows the convergence and the practical usefulness of the algorithm and (in most cases) the superiority of open-loop feedback over open-loop controls.

Keywords Optimal control; Stochastic control; Algorithms; Econometric modeling; Policy applications

1 Introduction

Optimum control theory has found a great number of applications in many areas of science from engineering to economics. In particular, there exist many studies determining optimal economic policies for econometric models. Most of these optimum control applications use either algorithms for linear dynamic systems or those that do not take into account the entire stochastic nature of the econometric model. Examples of the former case are Kendrick (1981), Coomes (1987), and the references in Amman (1996), for the latter Chow (1975, 1981). An algorithm that is explicitly aimed at providing (approximate) solutions to optimum control problems for nonlinear econometric models and other dynamic systems with different kinds of stochastics is OPTCON as introduced by Matulka and Neck (1992). However, so far OPTCON was severely limited by resting on very restrictive assumptions about the information available to the decision-maker. In particular, learning about the econometric model during the process of controlling the economy was ruled out by assumption. This means that the original algorithm used an open-loop control scheme for dealing with the stochastic parameters of the econometric model during the computation of optimum control variables. In reality, however, at each period new information arrives, and econometric models are regularly re-specified using this information. Therefore, extensions of the OPTCON algorithm to various possibilities of obtaining and using new information about the system to be controlled are highly desirable.

In the present paper, we extend the OPTCON algorithm from open-loop control only (which we label OPTCON1) to the inclusion of passive

learning or open-loop feedback control, where the stochastic parameters are updated in each period. The resulting algorithm is called OPTCON2; it can deliver approximately optimal solutions to dynamic optimization (optimum control) problems for a rather large class of nonlinear dynamic systems under a quadratic objective function with stochastic uncertainty in the parameters and in the system equations under both kinds of control schemes. In the open-loop feedback part, it is assumed that in each period new realizations of both random processes occur, which can be used to update the parameters of the dynamic system, i.e., of the econometric model. Following Kendrick's (1981) approach, the updating of the stochastic parameters uses the Kalman filter. By this extension, we expect to arrive at more reliable approximations to the solution of stochastic optimum control problems. Whether this hope will materialize depends upon the comparative performance of open-loop feedback versus open-loop control schemes in actual applications. Here we provide some clue to this issue by performing comparisons of the two control schemes within a control problem for a small econometric model. This serves also to show that the OPTCON2 algorithm and its implementation in *C#* actually deliver plausible numerical solutions at least for a small problem with real economic data.

The paper has the following structure: In Section 2, we define the class of problems to be tackled by the algorithm. Section 3 briefly reviews the OPTCON1 algorithm. In Section 4, we sketch the theoretical background and explain the OPTCON2 algorithm. In Section 5, we introduce the small econometric model for Slovenia SLOVNL, show the applicability and convergence of OPTCON2 as implemented in *C#*, and compare the quality of

open-loop and open-loop feedback (passive-learning) controls in Monte-Carlo simulations for this model. Section 6 concludes. In the Appendix, we give more details about the updating procedure used for determining the open-loop feedback control solution.

2 The problem

The algorithm OPTCON is designed to provide approximate solutions to optimum control problems with a quadratic objective function (a loss function to be minimized) and a nonlinear multivariate discrete-time dynamic system under additive and parameter uncertainties. The intertemporal objective function is formulated in quadratic tracking form, which is quite often used in applications of optimum control theory to econometric models. It can be written as

$$J = \sum_{t=S}^T L_t(x_t, u_t) \quad (1)$$

with

$$L_t(x_t, u_t) = \frac{1}{2} \begin{pmatrix} x_t - \tilde{x}_t \\ u_t - \tilde{u}_t \end{pmatrix}' W_t \begin{pmatrix} x_t - \tilde{x}_t \\ u_t - \tilde{u}_t \end{pmatrix}. \quad (2)$$

x_t is an n -dimensional vector of state variables that describes the state of the economic system at any point in time t . u_t is an m -dimensional vector of control variables, $\tilde{x}_t \in R^n$ and $\tilde{u}_t \in R^m$ are given 'ideal' (desired, target) levels of the state and control variables, respectively. S denotes the initial and T the terminal time period of the finite planning horizon. W_t is the

weight matrix, which can be written as

$$W_t = \begin{pmatrix} W_t^{xx} & W_t^{xu} \\ W_t^{ux} & W_t^{uu} \end{pmatrix}, \text{ for } t = S, \dots, T,$$

where W_t^{xx} , W_t^{xu} , W_t^{ux} and W_t^{uu} are $(n \times n)$, $(n \times m)$, $(m \times n)$ and $(m \times m)$ matrices, respectively. In a frequent special case, W_t is a matrix including a discount factor α with $W_t = \alpha^{t-1}W$. W_t (or W) is symmetric with $W^{xu} = (W^{ux})'$.

The dynamic system of nonlinear difference equations has the form

$$x_t = f(x_{t-1}, x_t, u_t, \theta, z_t) + \varepsilon_t, \quad t = S, \dots, T, \quad (3)$$

where θ is a p -dimensional vector of unknown parameters (parameter uncertainty), z_t denotes an l -dimensional vector of non-controlled exogenous variables, and ε_t is an n -dimensional vector of additive disturbances (system error). θ and ε_t are assumed to be independent random vectors with known expectations ($\hat{\theta}$ and O_n , respectively) and covariance matrices ($\Sigma^{\theta\theta}$ and $\Sigma^{\varepsilon\varepsilon}$, respectively). f is a vector-valued function, $f^i(\dots)$, is the i -th component of $f(\dots)$, $i = 1, \dots, n$.

3 OPTCON1

The basic algorithm OPTCON determines approximate solutions of optimum control problems with a quadratic objective function and a nonlinear multivariate dynamic system under additive and parameter uncertainties. It com-

bines elements of previous algorithms developed by Chow (1975, 1981) incorporating nonlinear systems but no multiplicative uncertainty and Kendrick (1981) dealing with linear systems and all kinds of uncertainty. The version OPTCON1 is described in detail in Matulka and Neck (1992); here its basic idea is presented only.

It is well known in stochastic control theory that a general analytical solution to dynamic stochastic optimization problems cannot be achieved even for very simple control problems. The main reason is the so-called dual effect of control under uncertainty, meaning that controls not only contribute directly to achieving the stated objective but also affecting future uncertainty and hence the possibility of indirectly improving upon the system performance by better information about the system (see, for instance, Aoki (1989); Neck (1984)). Therefore only approximations to the true optimum for such problems are feasible, with various schemes existing for dealing with the problem of information acquisition.

A useful distinction was adapted from the engineering literature by Kendrick (1981); see also Kendrick and Amman (2006): Open-loop policies neglect the possibility of receiving information (measurements) during the operation of the system; open-loop feedback (or passive learning) policies use actual information to determine the control but do not anticipate future measurements; and closed-loop (or active learning) policies make some use also of information about future measurements. Given the intricacies of the interplay between control and information, even for very simple stochastic control problems (for example, linear scalar system, time horizon of only two periods) an exact analytical or even numerical solution is impossible, hence numerical

approximations are required that make use of simplifying assumptions.

The algorithm OPTCON1 determines policies belonging to the class of open-loop controls. It either neglects the stochastics of the system altogether (deterministic solution, identical to the Chow algorithm) or assumes the stochastics (expectation and covariance matrices of additive and multiplicative disturbances) given once for all at the beginning of the planning horizon (stochastic solution). The problem with the nonlinear system is tackled iteratively, starting with a tentative path of state and control variables. The tentative path of the control variables for the first iteration is given. In order to find the corresponding tentative path of the state variables, the nonlinear system is solved numerically using the Newton-Raphson method (or alternatively the Gauss-Seidel method).

After the tentative path is found, the iterative approximation of the optimal solution starts. The solution is searched from one time path to another until the algorithm converges or the maximal number of iterations is achieved. During this search the system is linearized around the previous iteration's result as tentative path and the problem is solved for the resulting time-varying linearized system. The criterion for convergence demands that the difference between the values of actual and previous iterations be smaller than a pre-specified number. The approximately optimal solution of the problem for the linearized system is found under the above-mentioned simplifying assumptions about the information pattern; then this solution is used as the tentative path for next iteration, starting the procedure again.

Every iteration, i.e. every solution of the problem for the linearized system, has the following structure: the objective function is minimized using

Bellman's principle of optimality to obtain the parameters of the feedback control rule. This uses known results for the stochastic control of LQG problems (optimization of linear systems with Gaussian noise under a quadratic objective function). A backward recursion over time starts in order to calculate the controls for the first period S . Next, the optimal values of the state and the control variables are calculated applying forward recursion, i.e. beginning with u_S and x_S at period S and finishing with u_T and x_T at the terminal period T . If the convergence criterion is fulfilled, the solution of the last iteration is taken as the approximately optimal solution of the problem and the algorithm stops. Finally, the value of the objective function is calculated for this solution.

4 OPTCON2

4.1 Characteristics of the algorithm OPTCON2

The new version of algorithm, OPTCON2, incorporates both open-loop and open-loop feedback (passive-learning) controls. The idea of passive learning corresponds to actual practice in applied econometrics: at the end of each time period, the model builder (and hence the control agent) observes what happens, that is, the actual values of state variables, and uses this information to re-estimate the model and hence improve his/her knowledge about the system. It should be mentioned that two kinds of errors, namely additive (random system errors) and multiplicative ('structural' errors in parameters) errors, are taken into account but not possible specification errors, hence we

assume that no re-specifications of the model are performed. Taking actual observations into account should contribute to eliminate errors in parameters or at least to reduce them. This is not trivial because the correction of structural errors is disturbed by the system error. Using the updated model shall enable the policy-maker to obtain better predictions for the next period and to improve the final results. Whether this taking into account of passive learning really improves the performance of the system is, however, an open question because the ‘true’ optimum is not known.

In this paper, our main research aim is to obtain evidence on whether applying passive learning can indeed improve the final solution. The prediction and optimization procedures for open-loop control assume that the model is known without random disturbances occurring during the optimization process. But in reality some random errors will disturb the optimization process. We consider two kinds of uncertainties, parameter and system errors. Passive-learning strategy implies observing actual information and using it in order to adjust the optimization procedure. For the purpose of comparing open-loop and open-loop feedback results, we cannot observe actual and true values, so we have to resort to Monte-Carlo simulations. We generate large numbers of random time paths for the additive and multiplicative errors representing how new information in reality could look like. In this way we create ‘quasi-real’ observations and compare both types of controls, open-loop and passive-learning (open-loop feedback). The procedure applied looks like as follows.

We generate MC_runs (a number, usually between 100 and 1000) different sets of realizations of random noises $(\varepsilon_t^m)_{t=S}^T$ and μ^m , $m = 1, \dots, MC_runs$.

We assume that there is an unknown ‘real’ model with the ‘true’ parameter vector $\hat{\theta}$. But the policy-maker (the user of the algorithm) does not know the real parameters and works with the ‘wrong’ parameters resulting from the estimates using the realization of the random processes. To simulate this, we use the system error generated in each Monte-Carlo run m $(\varepsilon_t^m)_{t=S}^T$ and create the ‘updated’ parameter $\theta^m = \hat{\theta} + \mu^m$ which is used by the policy maker. After this, the following procedure is run for every random scenario: We start a forward loop from S to T . In each time period t we determine an (approximately optimal) open-loop solution for the subproblem, i.e. the problem for the time periods from t to T . Then we fix the predicted x_t^* and u_t^* for the time period t . We assume to know the realized or actual values of the state variables x_t^{a*} at the end of period t , which is, however, disturbed by the additive errors: the difference between $x_t^* = f(x_{t-1}^{a*}, x_t^*, u_t^*, \theta^m, z_t)$ and $x_t^{a*} = f(x_{t-1}^{a*}, x_t^{a*}, u_t^*, \hat{\theta}, z_t) + \varepsilon_t^m$ comes from the random numbers ε_t^m generated and used for calculating the ‘quasi-real’ values. Next, we can use the new information to update and adjust the parameter estimate θ^m . After that we apply the same procedure for the remaining subproblem from $t + 1$ to T , and so on.

4.2 Detailed description of the algorithm

STEP I: Compute a tentative state path $(\overset{\circ}{x}_t)_{t=S}^T$ by solving the system of equations $f(\dots)$ with the Newton-Raphson algorithm (or Newton-Raphson with line-search expansion), given the tentative policy path $(\overset{\circ}{u}_t)_{t=S}^T$.

STEP II: Generate MC_runs paths of random normal distributed system noises $(\varepsilon_t^m)_{t=S}^T$ and parameter noises μ^m ($\theta^m = \hat{\theta} + \mu^m$) using the given means and covariance matrices. The given covariance matrix is Cholesky decomposed in order to get the lower-triangular matrix. Applying this to uncorrelated random numbers produces a vector with the covariance properties of the system being modeled.

STEP III: For each independent random scenario with $(\varepsilon_t^m)_{t=S}^T$ and μ^m , i.e. for each Monte Carlo run m ($m = 1, \dots, MC_runs$), do the following steps:

Step III-1: For each t from S to T find the open-loop solution for the subproblem (t, \dots, T) , i.e. do the steps **III-1a-d**:

Step III-1a: *Nonlinearity loop*: repeat the following steps until the stop criterion is fulfilled, i.e., until the difference between the values of the actual and the previous iteration is smaller than a pre-specified number or the maximal number of iterations is achieved.

Step III-1a-i: Initialization for the backward recursion:

$$H_{T+1} = O_{n \times n}, h_{T+1}^x = O_n,$$

$$h_{T+1}^c = 0, h_{T+1}^s = 0, h_{T+1}^p = 0.$$

Step III-1a-ii: *Backward recursion*: do steps [1]-[7] for each time period from T until t . This is analogous to the procedure in OPTCON1.

[1] Compute the expected values of the parameters of the linearized system of equations:

$$A_t = (I_n - F_{x_t})^{-1} F_{x_{t-1}}, \quad (4)$$

$$B_t = (I_n - F_{x_t})^{-1} F_{u_t}, \quad (5)$$

$$c_t = \overset{\circ}{x}_t - A_t \overset{\circ}{x}_{t-1} - B_t \overset{\circ}{u}_t, \quad (6)$$

$$\xi_t = (I_n - F_{x_t})^{-1} \varepsilon_t, \quad (7)$$

$$E_{t-1}(\xi_t) = (I_n - F_{x_t})^{-1} 0_n = 0_n, \quad (8)$$

$$\Sigma_t^{\xi\xi} = Cov_{t-1}(\xi_t, \xi_t) = (I_n - F_{x_t})^{-1} \Sigma^{\varepsilon\varepsilon} [(I_n - F_{x_t})^{-1}]', \quad (9)$$

where all derivatives are evaluated at the reference values $\overset{\circ}{x}_{t-1}$, $\overset{\circ}{x}_t$, $\overset{\circ}{u}_t$, z_t , and $\overset{\circ}{\varepsilon}_t$. Here the time-invariant nonlinear system $f(\dots)$ is approximated by a time-varying linear system of equations.

[2] Compute the derivatives of the parameters of the linearized system with respect to θ^m :

$$D^{A_t} = [(I_n - F_{x_t})^{-1} \otimes I_p][F_{x_t, \theta} A_t + F_{x_{t-1}, \theta}], \quad (10)$$

$$D^{B_t} = [(I_n - F_{x_t})^{-1} \otimes I_p][F_{x_t, \theta} B_t + F_{u_t, \theta}], \quad (11)$$

$$d^{c_t} = vec[((I_n - F_{x_t})^{-1} F_{\theta})'] - D^{A_t} \overset{\circ}{x}_{t-1} - D^{B_t} \overset{\circ}{u}_t, \quad (12)$$

where all derivatives are evaluated at the same reference values as above.

[3] Calculate the influence of the stochastic parameters: compute all the matrices the cells of which are defined by:

$$[\Upsilon_t^{AKA}]_{i,j} = tr[K_t D^{a_t,j} \Sigma^{\theta\theta} (D^{a_t,i})'], \quad i = i, \dots, n; \quad j = 1, \dots, n, \quad (13)$$

$$[\Upsilon_t^{BKA}]_{i,j} = tr[K_t D^{a_t,j} \Sigma^{\theta\theta} (D^{b_t,i})'], \quad i = i, \dots, m; \quad j = 1, \dots, n, \quad (14)$$

$$[\Upsilon_t^{BKB}]_{i,j} = tr[K_t D^{b_t,j} \Sigma^{\theta\theta} (D^{b_t,i})'], \quad i = i, \dots, m; \quad j = 1, \dots, m, \quad (15)$$

$$[v_t^{AKc}]_i = tr[K_t D^{c_t} \Sigma^{\theta\theta} (D^{a_t,i})'], \quad i = i, \dots, n, \quad (16)$$

$$[v_t^{BKc}]_i = tr[K_t D^{c_t} \Sigma^{\theta\theta} (D^{b_t,i})'], \quad i = i, \dots, m, \quad (17)$$

$$[v_t^{cKc}] = tr[K_t D^{c_t} \Sigma^{\theta\theta} (D^{c_t})'], \quad i = i, \dots, m. \quad (18)$$

[4] Convert the objective function from ‘quadratic-tracking’ to ‘general quadratic’ format:

$$W_t^{xx} = \alpha^{t-1} W^{xx}, \quad (19)$$

$$W_t^{ux} = \alpha^{t-1} W^{ux}, \quad (20)$$

$$W_t^{uu} = \alpha^{t-1} W^{uu}, \quad (21)$$

$$w_t^x = -W_t^{xx} \tilde{x}_t - W_t^{xu} \tilde{u}_t, \quad (22)$$

$$w_t^u = -W_t^{ux} \tilde{x}_t - W_t^{uu} \tilde{u}_t, \quad (23)$$

$$w_t^c = \frac{1}{2} \tilde{x}_t W_t^{xx} \tilde{x}_t + \tilde{u}_t W_t^{ux} \tilde{x}_t + \frac{1}{2} \tilde{u}_t' W_t^{uu} \tilde{u}_t. \quad (24)$$

[5] Compute the parameters of the function of expected accumulated loss:

$$K_t = W_t^{xx} + H_{t+1}, \quad (25)$$

$$k_t^x = w_t^x + h_{t+1}^x.$$

$$\begin{aligned} \Lambda_t^{xx} &= \Upsilon_t^{AKA} + A_t' K_t A_t, \\ \Lambda_t^{ux} &= \Upsilon_t^{BKA} + B_t' K_t A_t + W_t^{ux} A_t, \\ \Lambda_t^{xu} &= (\Lambda_t^{ux})', \end{aligned} \quad (26)$$

$$\Lambda_t^{uu} = \Upsilon_t^{BKB} + B_t' K_t B_t + 2B_t' W_t^{xu} + W_t^{uu},$$

$$\lambda_t^x = v_t^{AKc} + A_t' K_t c_t + A_t' k_t^x,$$

$$\lambda_t^u = v_t^{BKc} + B_t' K_t c_t + B_t' k_t^x + W_t^{ux} c_t + w_t^u,$$

$$\lambda_t^s = \frac{1}{2} \text{tr}[K_t \Sigma_t^{\xi\xi}] + h_{t+1}^s, \quad (27)$$

$$\lambda_t^p = \frac{1}{2} v_t^{cKc} + h_{t+1}^p,$$

$$\lambda_t^c = \frac{1}{2} c_t' K_t c_t + c_t' k_t^x + w_t^c + h_{t+1}^c.$$

[6] Compute the parameters of the feedback rule for T, \dots, t :

$$G_t = -(\Lambda_t^{uu})^{-1} \Lambda_t^{ux}, \quad (28)$$

$$g_t = -(\Lambda_t^{uu})^{-1} \lambda_t^u.$$

[7] Compute the parameters of the function of minimal expected accumulated loss:

$$\begin{aligned}
H_t &= \Lambda_t^{xx} - \Lambda_t^{xu}(\Lambda_t^{uu})^{-1}\Lambda_t^{ux}, \\
h_t^x &= \lambda_t^x - \Lambda_t^{xu}(\Lambda_t^{uu})^{-1}\lambda_t^u, \\
h_t^c &= \lambda_t^c - \frac{1}{2}(\lambda_t^u)'(\Lambda_t^{uu})^{-1}\lambda_t^u, \\
h_t^s &= \lambda_t^s, \\
h_t^p &= \lambda_t^p.
\end{aligned} \tag{29}$$

Step III-1a-iii: Repeat the following steps [1]-[2] for the time periods t, \dots, T :

[1] The expected approximately optimal policies are computed using the feedback rule:

- for the time period t : $u_t^* = G_t x_{t-1}^{a*} + g_t$;
- for the time periods $t' = t + 1, \dots, T$: $u_{t'}^* = G_{t'} x_{t'-1}^* + g_{t'}$.

[2] The approximately optimal values of the state variables are calculated:

- for the time period S : $x_{S-1}^{a*} = \overset{\circ}{x}_{S-1}$;
- for the time period t : $x_t^* = f(x_{t-1}^{a*}, x_t^*, u_t^*, \theta^m, z_t)$;
- for the time periods $t' = t + 1, \dots, T$: $x_{t'}^* = f(x_{t'-1}^*, x_{t'}^*, u_{t'}^*, \theta^m, z_{t'})$.

Step III-1a-iv: If the stop criterion for the non-linearity loop is not ful-

filled, set $(\overset{\circ}{x}_t, \overset{\circ}{u}_t)_t^T = (x_t^*, u_t^*)_{t=S}^T$ and start the search for the optimal values again (go to step **III-1a-ii**).

If the stop criterion is achieved, the approximately optimal solution $(x_t^*, u_t^*)_t^T$ is found. Then go to the next step **III-1b**. It should be noted that after several runs of nonlinearity loop only the solution (x_t^*, u_t^*) for the time period t will be taken as the optimal solution. The calculations of the pairs $(x_{t'}^*, u_{t'}^*)$ for other periods have to be done again, taking into account the re-estimated stochastic parameters for all periods.

Step III-1b: Calculate the following for only one time period t :

$$x_t^{a*} = f(x_{t-1}^{a*}, x_t^{a*}, u_t^*, \hat{\theta}, z_t) + \varepsilon_t^m.$$

Step III-1c: Update the stochastic parameters θ^m using the Kalman filter and the actual (realized) values of the variables x_t^{a*} :

[1] Prediction:

$$\begin{aligned} \hat{x}_{t/t-1} &= f(x_{t-1}^{a*}, \hat{x}_{t/t-1}, u_t^*, \theta_{t-1/t-1}^m, z_t) = x_t^*, & \theta_{t/t-1}^m &= \theta_{t-1/t-1}^m, \\ \Sigma_{t/t-1}^{xx} &= F_{\theta t-1}^x \Sigma_{t-1/t-1}^{\theta\theta} (F_{\theta t-1}^x)' + \Sigma_t^{\varepsilon\varepsilon}, \\ \Sigma_{t/t-1}^{x\theta} &= (\Sigma_{t/t-1}^{\theta x})' = F_{\theta t-1}^x \Sigma_{t-1/t-1}^{\theta\theta}, & \Sigma_{t/t-1}^{\theta\theta} &= \Sigma_{t-1/t-1}^{\theta\theta}. \end{aligned} \tag{30}$$

[2] Correction:

$$\begin{aligned}\Sigma_{t/t}^{\theta\theta} &= \Sigma_{t/t-1}^{\theta\theta} - \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} \Sigma_{t/t-1}^{x\theta}, \\ \theta_{t/t}^m &= \theta_{t/t-1}^m + \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} [x_t^{a*} - \tilde{x}_t] \quad \text{and} \quad \hat{x}_{t/t} = x_t^{a*}.\end{aligned}\tag{31}$$

Thus the update results in the new values $\theta_{t/t}^m$ and $\Sigma_{t/t}^{\theta\theta}$. The proof of (30) and (31) is given in the Appendix.

Step III-1d: Set $\theta^m = \theta_{t/t}^m$ and $\Sigma^{\theta\theta} = \Sigma_{t/t}^{\theta\theta}$

and run the procedure for the next starting period. This loop is finished when $t = T$ and the approximately optimal open-loop feedback control and state variables for all periods are found.

Step III-2: Compute the expected welfare loss:

$$J^* = \sum_{t=S}^T L_t(x_t^{a*}, u_t^*)\tag{32}$$

with

$$L_t(x_t^{a*}, u_t^*) = \frac{1}{2} \begin{pmatrix} x_t^{a*} - \tilde{x}_t \\ u_t^* - \tilde{u}_t \end{pmatrix}' W_t \begin{pmatrix} x_t^{a*} - \tilde{x}_t \\ u_t^* - \tilde{u}_t \end{pmatrix}.\tag{33}$$

5 An application

The algorithm OPTCON2 was implemented in C#. In order to test its convergence, we use a very simple small macroeconomic model for Slovenia. Section 5.1 gives the details of this model. Optimization results of this model for two different open-loop policies, a deterministic and a stochastic case, are

presented in Section 5.2. In Section 5.3, we describe the comparison between results of open-loop controls and passive-learning (open-loop feedback) control solutions.

5.1 The model SLOVNL

We consider a small nonlinear macroeconomic model of the Slovenian economy, to be called SLOVNL (**SLOV**enian model, **Non-Linear** version). SLOVNL consists of 8 equations, 4 of them behavioral and 4 identities. The model includes 8 state variables, 4 exogenous non-controlled variables, 3 control variables, and 16 unknown (estimated) parameters. The quarterly data for the time periods 1995:1 until 2006:4 yield 48 observations and admit a full-information maximum likelihood (FIML) estimation of the parameters and the covariance matrices for the parameter and system errors. The start period for the optimization is $S = 1$ (2004:1) and the end period is $T = 12$ (2006:4).

Model variables used in SLOVNL

Endogenous (state) variables:

$x[1]$	<i>CR</i>	real private consumption
$x[2]$	<i>INVR</i>	real investment
$x[3]$	<i>IMPR</i>	real imports of goods and services
$x[4]$	<i>STIRLN</i>	short term interest rate
$x[5]$	<i>GDP</i>	real gross domestic product
$x[6]$	<i>VR</i>	real total aggregate demand
$x[7]$	<i>PV</i>	general price level
$x[8]$	<i>Pi4</i>	rate of inflation

Control variables:

$u[1]$	<i>TaxRate</i>	net tax rate
$u[2]$	<i>GR</i>	real public consumption
$u[3]$	<i>M3N</i>	money stock, nominal

Exogenous non-controlled variables:

$z[1]$	<i>EXR</i>	real exports of goods and services
$z[2]$	<i>IMPDEF</i>	import price level
$z[3]$	<i>GDPDEF</i>	domestic price level
$z[4]$	<i>SITEUR</i>	nominal exchange rate SIT/EUR

Model equations:

The first four equations are estimated by FIML, the remaining equations are identities.

Standard deviations are given in brackets.

$$\begin{aligned}
CR_t = & 240.9398 + 0.740333 CR_{t-1} + 0.111727 GDPR_t \left(1 - \frac{TaxRate_t}{100}\right) \\
& (189.7449) \quad (0.1115) \quad (0.0330) \\
& - 1.007353 (STIRLN_t - Pi4_t) - 4.773533 Pi4_t \\
& (2.5848) \quad (2.4966)
\end{aligned}$$

$$\begin{aligned}
INVR_t = & 75.41731 + 0.932211 INVR_{t-1} + 0.264523 (VR_t - VR_{t-1}) \\
& (176.8549) \quad (0.1423) \quad (0.0924) \\
& - 0.455511 (STIRLN_t - Pi4_t) - 2.981241 Pi4_t \\
& (6.9044) \quad (3.1277)
\end{aligned}$$

$$\begin{aligned}
IMPR_t = & IMPR_{t-1} + 0.826449 (VR_t - VR_{t-1}) - 38.14954 SITEUR_t \\
& (0.0724) \quad (18.9336)
\end{aligned}$$

$$\begin{aligned}
STIRLN_t = & 0.811606 STIRLN_{t-1} - 0.000877 \frac{(M3N)_t}{PV_t} \cdot 100 \\
& (0.1375) \quad (0.0008) \\
& + 0.002746 GDPR_t \\
& (0.0026)
\end{aligned}$$

$$GDPR_t = CR_t + INVR_t + GR_t + EXR_t - IMPR_t$$

$$VR_t = GDPR_t + IMPR_t$$

$$PV_t = \frac{GDPR_t}{VR_t} \cdot GDPDEF_t + \frac{IMPR_t}{VR_t} \cdot IMPDEF_t$$

$$Pi4_t = \frac{PV_t - PV_{t-4}}{PV_{t-4}} \cdot 100$$

The objective function penalizes deviations of objective variables from their ‘ideal’ (desired, target) values. The ‘ideal’ values of the state and control

variables (\tilde{x}_t and \tilde{u}_t , respectively) are chosen as follows:

Table 1: ‘Ideal’ values

<i>CR</i>	<i>INVR</i>	<i>IMPR</i>	<i>STIRLN</i>	<i>GDPR</i>	<i>VR</i>	<i>PV</i>	<i>Pi4</i>	<i>TaxRate</i>	<i>GR</i>	<i>M3N</i>
1%	1%	2%	-0.25	1%	1.5%	0.75%	3	25.2	1%	1.75%

We use growth rates (denoted by % in Table 1) for calculating the ‘ideal’ values for most variables, starting from the last given observation ($t = S - 1$). For *Pi4* and *TaxRate*, we use constant values; for *STIRLN*, we use a linear decrease of 0.25 per quarter.

The weights for the variables, i.e. the matrix W in the objective function, are first chosen as shown in Table 2a (‘raw’ weights) to reflect the relative importance of the respective variable in the (hypothetical) policy-maker’s objective function. These ‘raw’ weights have to be scaled or normalized according to the levels of the respective variables to make the weights comparable. To do so, the ‘raw’ weights are multiplied by normalization coefficients $NC^i = (ML/MA^i)^2$, where ML is the mean of the series with the largest values and MA^i is the mean of the respective series i . The ‘correct’ weights obtained in this way are shown in Table 2b. The weight matrix is assumed to be constant over time (no discounting).

Next, we apply the algorithm to this optimization problem for determining approximately optimal fiscal and monetary policies for Slovenia under the assumed objective function and the econometric model SLOVNL. We run two different experiments: In experiment 1, we compare two open-loop solutions, a deterministic case where the variances and covariances of the parameters are ignored, and a stochastic case where the covariance matrix is taken into

Table 2: Weights

2a: ‘raw’ weights		2b: ‘correct’ weights	
variable	weight	variable	weight
<i>CR</i>	1	<i>CR</i>	3.457677
<i>INVR</i>	1	<i>INVR</i>	12.16323
<i>IMPR</i>	1	<i>IMPR</i>	1.869532
<i>STIRLN</i>	1	<i>STIRLN</i>	216403.9
<i>GDPR</i>	2	<i>GDPR</i>	2
<i>VR</i>	1	<i>VR</i>	0.333598
<i>PV</i>	1	<i>PV</i>	423.9907
<i>Pi4</i>	0	<i>Pi4</i>	0
<i>TaxRate</i>	2	<i>TaxRate</i>	37770.76
<i>GR</i>	2	<i>GR</i>	63.77052
<i>M3N</i>	2	<i>M3N</i>	0.090549

account. In experiment 2, we compare the quality of the open-loop and the open-loop-feedback solutions. Our exercise is not meant to determine actually optimal policies for Slovenia during the period considered (for this, the quality of the econometric model is not sufficient); instead, we want to obtain information about the convergence and the applicability of the OPTCON2 algorithm as implemented in *C#*.

5.2 Experiment 1: open-loop control

For experiment 1, we discuss two different open-loop solutions: deterministic and stochastic. The deterministic scenario means that all parameters of the model are known with certainty. In the stochastic case, the covariance matrix of the parameters as estimated by FIML is used but no updating of information occurs during the optimization process.

We show the graphical results for two variables, *GDPR* and *GR*. Each

figure shows the optimization results for the deterministic and the stochastic case and the ‘ideal’ and historical values of the respective variable.

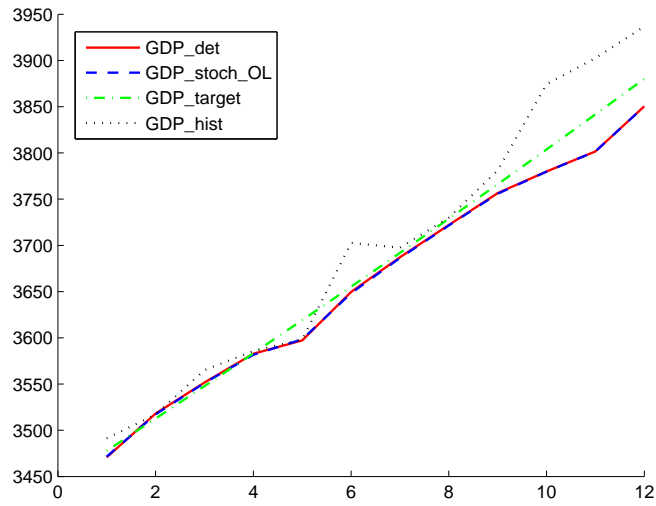


Figure 1: Deterministic vs. stochastic open-loop case (GDP)

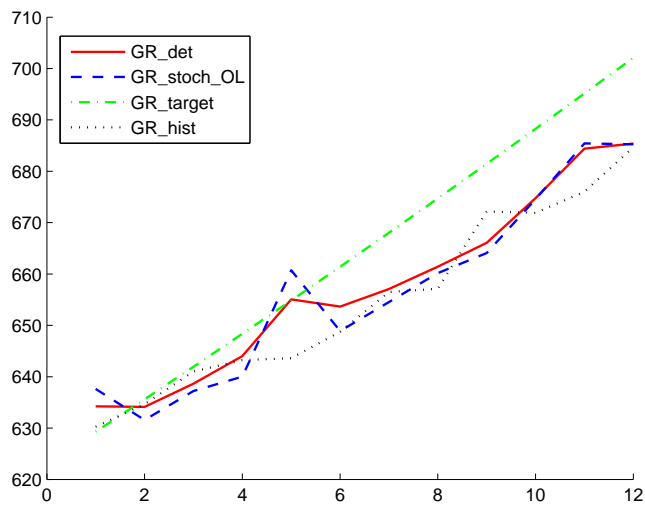


Figure 2: Deterministic vs. stochastic open-loop case (GR)

Figures 1 and 2 show that both the deterministic and the stochastic solution approximate the ‘ideal’ values rather well. This is also supported by the values of the objective function, which is 2,618,460.238 in the uncontrolled solution, 904,385.766 in the deterministic solution and 918,296.046 in the stochastic solution, showing a considerable improvement of the system performance. An interesting detail is that the deterministic and the stochastic open-loop solutions are very similar, which goes in line with previous findings in related studies Neck and Karbuz (1995).

In both cases (deterministic and stochastic) the algorithm needs 3 non-linearity runs to converge. The entire procedure requires 2 seconds for the deterministic case and 4 seconds for the stochastic case on a personal computer with 2 GHz Intel Core 2 Duo CPU and 4GB RAM. The results show that the algorithm OPTCON2 (OL-strategy) is applicable for determining optimal open-loop controls at least for small nonlinear econometric models.

5.3 Experiment 2: open-loop feedback control

The aim of the experiment 2 is to compare open-loop and open-loop feedback controls. There is a problem how to compare both strategies, because open-loop controls do not take into account random disturbances occurring during the optimization process. In order to make both strategies comparable, we need some adjustments for the open-loop controls: First we calculate the open-loop controls $(u_t^*)_{t=S}^T$ for all periods, using the generated θ^m ($\theta^m = \hat{\theta} + \mu^m$). Then we assume to know the ‘true’ model with the parameters $\hat{\theta}$ and system error ε_t^m and calculate the actual values of open-loop states

($x_t^{a*} = f(x_{t-1}^{a*}, x_t^{a*}, u_t^*, \hat{\theta}, z_t) + \varepsilon_t^m, t = S, \dots, T$). So the policies stay unchanged but the state variables are calculated taking into account the disturbances μ^m and ε^m . The open-loop feedback solution is determined according to the algorithm given in Section 4. In this way, a comparison of both strategies under simulated ‘real’ uncertainty (disturbances) becomes possible.

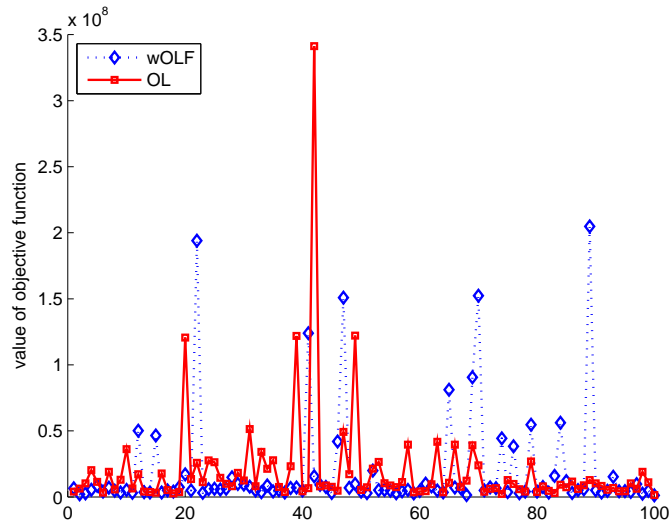


Figure 3: Open-loop vs. open-loop feedback control, value of objective function (100 Monte Carlo runs)

Figure 3 shows results from a representative Monte-Carlo simulation, displaying the value of the objective function (loss) arising from applying OPT-CON2 under 100 independent random Monte Carlo runs. Blue diamonds represent open-loop feedback results, red squares open-loop results. One can see from this figure that in most runs the blue diamonds are below the red squares (here: in 66 out of 100). This means that open-loop feedback controls give better results (lower values of the cost function) in the majority of the cases investigated. But one can also recognize that there are many outliers.

After having run many simulations (with different numbers of Monte Carlo runs), we can summarize our findings as follows:

- In 60-75 percent of the cases, open-loop feedback controls give better results than open-loop controls.
- Many outliers occur both in the open-loop and the open-loop feedback case.
- For open-loop controls, outliers seem to be more frequent.

The last result is somewhat unexpected. Outliers mean that (passive) learning may not improve the quality of the final results; it even may worsen them. One possible way out of this problem is to introduce weights in the update structure. Recalling equation (31), which describes the correction procedure, we extend the correction term for θ by a weighting parameter V_t :

$$\theta_{t/t} = \theta_{t/t-1} + V_t \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} [x_t^{a*} - x_t^*], \quad 0 < V_t \leq 1. \quad (34)$$

Adjusting the updating procedure in this way, we expect to get better results under open-loop feedback control.

Using constant values for V_t does not improve the open-loop feedback results much. The main reason for the outliers can be understood from recognizing that we have two kinds of uncertainty in our problem, additive (random system error) and multiplicative error ('structural' error in the parameters). The updating procedure aims at reducing the 'structural' error but can be disturbed by the random system error. Usually, this influence of the random system error can be expected to be especially strong at the

beginning of the planning period. Thus, we propose to give less weight to revisions called for by learning during the earlier periods of the planning horizon than during the later periods. Taking this consideration into account, we introduced a time-dependent weighting parameter V_t . We tried two different schemes:

- $V_t = \frac{t}{T-1}$,
- $V_t = \begin{cases} 0.1, & 1 \leq t \leq \frac{1}{3}T \\ 0.5, & \frac{1}{3}T < t \leq \frac{2}{3}T. \\ 1, & \frac{2}{3}T < t \leq T \end{cases}$

According to our simulations, the weighted open-loop feedback scheme with the parameter $V_t = \frac{t}{T-1}$ gives better results, so in the simulations presented next we use this variant.

From a Monte-Carlo simulation to compare the system performance under open-loop (OL) and weighted open-loop-feedback (wOLF) control, the results look like as in Figure 4, again showing the values of the objective function against the number of the Monte Carlo run. Blue diamonds represent wOLF and red squares OL results.

We can see that in this simulation wOLF control gives better results: there are more blue diamonds below red squares (44 out of 50) and nearly no wOLF outliers. However, in some simulations we obtain outliers also for wOLF like those shown in Figure 5. Due to a particular constellation of parameter and system errors, wOLF control fails in the 70th Monte Carlo run. But also here wOLF gives less outliers and in most cases blue diamonds are below red squares (here: 77 out of 100).

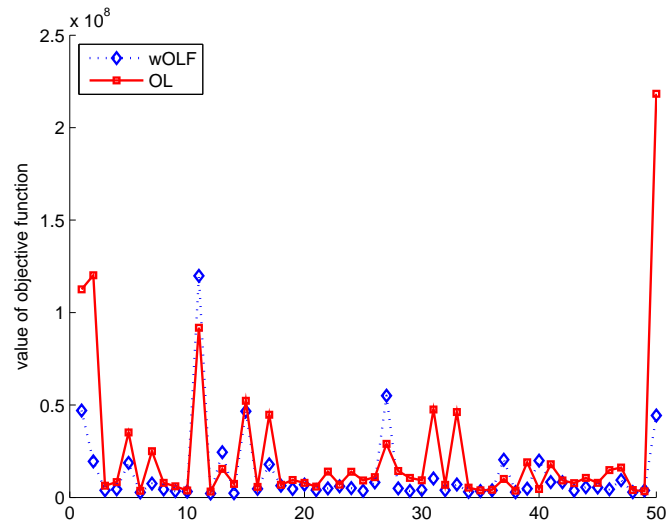


Figure 4: Open-loop vs. weighted open-loop feedback, value of objective function (50 Monte Carlo runs)

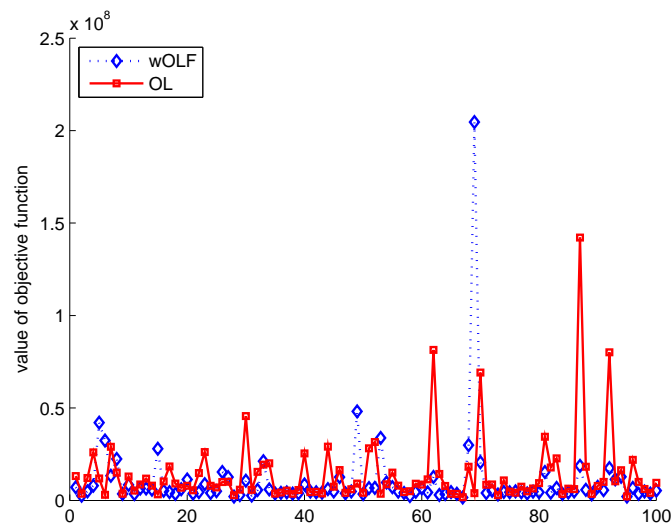


Figure 5: Open-loop vs. weighted open-loop feedback, value of objective function (100 Monte Carlo runs)

After running many simulations we can summarize our finding as follows:

- In 70-80 percent of the cases considered, weighted open-loop feedback controls give better results than open-loop controls.
- Weighted open-loop feedback controls result in less outliers than open-loop controls.

Using a weighting scheme for parameter updating increases the number of runs where passive learning controls result in better values of the objective function than the control scheme without learning. Additionally, we could achieve a decrease of the number of outliers by introducing wOLF controls.

6 Concluding remarks

The objective of our research is an extension of the OPTCON algorithm to make more reliable. We developed the new algorithm, OPTCON2, embodying open-loop feedback in addition to open-loop control policies. It was programmed in the computer language *C#* and shown to converge for a small econometric model. The main improvement described in this paper consists in learning about stochastic parameters during the control process. A comparison of open-loop control (without learning) and open-loop feedback control (with passive learning) shows that weighted open-loop feedback control in a majority of the cases investigated outperforms open-loop control for the small econometric model of Slovenia.

The next task is to apply OPTCON2 to larger and better (in terms of the theoretical and statistical quality) macroeconomic models. Because of the

expected huge amount of computational, this will probably have to be done by parallel computing. Additional comparisons of the policy performance with respect to the postulated objective function are desirable; for example, we will calculate controls determined by straightforward heuristic optimization procedures (see Gilli and Winker (2009)) and assess their performance as compared to the more sophisticated ones calculated by OPTCON2. Some other improvements will have to be done in the algorithm and the present version of the computer program, as for instance concerning the calculation of the covariances in the linearized model. Moreover, major extensions will have to include various schemes of active learning to deal with the dual nature of the control under uncertainty along the lines of Kendrick (1981) for the linear case. Another challenge consists in incorporating rational (forward-looking) expectations and hence a non-causal structure in the dynamic system; see Amman and Kendrick (2000) for the linear case.

7 Appendix: Update of stochastic parameters

The update of the parameters in OPTCON2 is performed using the Kalman filter. In our case, the predicted parameter values are corrected using the actual observations of the state (endogenous) variables x^{a*} .

At the start of step **III-1c** in Section 4.2, the predicted values of $\hat{x}_{t/t-1} = x_t^*$ are already calculated and $\theta_{t/t-1}^m = \theta_{t-1}^m$ is known.

Step 1: Get the predicted values of the covariance matrices $\Sigma_{t/t-1}^{\theta\theta}$, $\Sigma_{t/t-1}^{xx}$

and $\Sigma_{t/t-1}^{\theta x}$.

Following Kendrick (1981), we start with the general case

$$\Sigma_{t/t-1} = E\{(x_t - \hat{x}_{t/t-1})(x_t - \hat{x}_{t/t-1})' | Y\},$$

where Y is the set of actual observations.

First we show that

$$\begin{aligned} \Sigma_{t/t-1} &= F_x \Sigma_{t-1/t-1} F_x' + \Sigma^{\varepsilon\varepsilon} \\ &+ \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}(F_{xx}^i \Sigma_{t-1/t-1} F_{xx}^j \Sigma_{t-1/t-1}). \end{aligned} \tag{35}$$

Denote by $\Sigma_{t-1/t-1} = \Sigma$ the known covariance. We consider the system of equations

$$x_t = f_t(x_{t-1}, x_t, u_t, \theta, z_t) + \varepsilon_t$$

and expand it to second order (Taylor expansion) around the path $(x_{t-1}^*, u_t^*) = (\hat{x}_{t-1/t-1}, u_t^*)$:

$$\begin{aligned} x_t &\approx f_t(\hat{x}_{t-1/t-1}, x_t, u_t^*, \theta, z_t) + [F_x][x_{t-1} - \hat{x}_{t-1/t-1}] + [F_u][u_t - u_t^*] \\ &+ \frac{1}{2} \sum_i e^i [x_{t-1} - \hat{x}_{t-1/t-1}]' F_{xx}^i [x_{t-1} - \hat{x}_{t-1/t-1}] \\ &+ \frac{1}{2} \sum_i e^i [u_t - u_t^*]' F_{uu}^i [u_t - u_t^*] \\ &+ \sum_i e^i [u_t - u_t^*]' F_{ux}^i [x_{t-1} - \hat{x}_{t-1/t-1}] + \varepsilon_t, \end{aligned} \tag{36}$$

where F_x and F_u are the first derivatives of f with respect to x and u , respectively; $F_{uu}^i = \frac{\partial^2 f^i}{\partial u^2}$, $F_{ux}^i = \frac{\partial^2 f^i}{\partial u \partial x}$ and $F_{xx}^i = \frac{\partial^2 f^i}{\partial x^2}$ are the second derivatives of f .

Set $u_t = u_t^*$ in (36) and get

$$\begin{aligned} x_t &= f_t(\hat{x}_{t-1/t-1}, x_t, u_t^*, \theta, z_t) + F_x[x_{t-1} - \hat{x}_{t-1/t-1}] \\ &+ \frac{1}{2} \sum_i e^i [x_{t-1} - \hat{x}_{t-1/t-1}] F_{xx}^i [x_{t-1} - \hat{x}_{t-1/t-1}] + \varepsilon_t. \end{aligned} \quad (37)$$

Taking the expected value of (36) with data through period $t-1$ and setting $u_t = u_t^*$ yields:

$$\begin{aligned} \hat{x}_{t/t-1} &\approx f_t(\hat{x}_{t-1/t-1}, \hat{x}_{t/t-1}, u_t^*, \theta, z_t) \\ &+ \frac{1}{2} E(\sum_i e^i [x_{t-1} - \hat{x}_{t-1/t-1}]' F_{xx}^i [x_{t-1} - \hat{x}_{t-1/t-1}]). \end{aligned} \quad (38)$$

Because of the rule $E(x'Ax) = \hat{x}A\hat{x} + \text{tr}(A\Sigma)$ we get

$$\hat{x}_{t/t-1} \approx f_t(\hat{x}_{t-1/t-1}, \hat{x}_{t/t-1}, u_t^*, \theta, z_t) + \frac{1}{2} \sum_i e^i \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}]. \quad (39)$$

Using (39) and (37), we write:

$$\begin{aligned}
x_t - \hat{x}_{t/t-1} &= F_x[x_{t-1} - \hat{x}_{t-1/t-1}] \\
&+ \frac{1}{2} \sum_i e^i [x_{t-1} - \hat{x}_{t-1/t-1}] F_{xx}^i [x_{t-1} - \hat{x}_{t-1/t-1}] + \varepsilon_t \\
&- \frac{1}{2} \sum_i e^i \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}].
\end{aligned} \tag{40}$$

Now we can write $\Sigma_{t/t-1}$ in a more detailed way as

$$\begin{aligned}
\Sigma_{t/t-1} &= E\{(x_t - \hat{x}_{t/t-1})(x_t - \hat{x}_{t/t-1})'\} \\
&= E\{F_x'[x_{t-1} - \hat{x}_{t-1/t-1}][x_{t-1} - \hat{x}_{t-1/t-1}]' F_x\} \\
&+ \frac{1}{4} E\{[\sum_i e^i [x_{t-1} - \hat{x}_{t-1/t-1}] F_{xx}^i [x_{t-1} - \hat{x}_{t-1/t-1}]] \\
&\times [\sum_j e^j [x_{t-1} - \hat{x}_{t-1/t-1}] F_{xx}^j [x_{t-1} - \hat{x}_{t-1/t-1}]]'\} \\
&+ E\{\varepsilon_t \varepsilon_t'\} + \frac{1}{4} E\{[\sum_i e^i \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}]] [\sum_j e^j \text{tr}[F_{xx}^j \Sigma_{t-1/t-1}]]'\} \\
&- \frac{1}{2} E\{[\sum_i e^i [x_{t-1} - \hat{x}_{t-1/t-1}] F_{xx}^i [x_{t-1} - \hat{x}_{t-1/t-1}]] \\
&\times [\sum_i e^i \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}]]'\}.
\end{aligned} \tag{41}$$

With $E\{[x_{t-1} - \hat{x}_{t-1/t-1}][x_{t-1} - \hat{x}_{t-1/t-1}]'\} = \Sigma_{t-1/t-1}$ and $E\{\varepsilon_t \varepsilon_t'\} = \Sigma^{\varepsilon\varepsilon}$, we obtain

$$\begin{aligned}
\Sigma_{t/t-1} &= F'_x \Sigma_{t-1/t-1} F_x \\
&+ \frac{1}{4} \sum_i \sum_j E\{[[x_{t-1} - \hat{x}_{t-1/t-1}] F_{xx}^i [x_{t-1} - \hat{x}_{t-1/t-1}]] \\
&\times [[x_{t-1} - \hat{x}_{t-1/t-1}] F_{xx}^j [x_{t-1} - \hat{x}_{t-1/t-1}]]'\} \\
&+ \Sigma^{\varepsilon\varepsilon} + \frac{1}{4} [\sum_i e^{i \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}]}] [\sum_j e^{j \text{tr}[F_{xx}^j \Sigma_{t-1/t-1}]}] \\
&- \frac{1}{2} [\sum_j e^{j \text{tr}[F_{xx}^j \Sigma_{t-1/t-1}]}] [\sum_i e^{i \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}]}]'.
\end{aligned} \tag{42}$$

Adapting the rule (see Kendrick (1981, Appendix F))

$$E[(x'Ax)(x'Bx)] = 2\text{tr}[A\Sigma B\Sigma] + \text{tr}[A\Sigma]\text{tr}[B\Sigma]$$

to the second term of (42), one obtains

$$\begin{aligned}
\Sigma_{t/t-1} &= F'_x \Sigma_{t-1/t-1} F_x + \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}[F_{xx}^i \Sigma_{t-1/t-1} f_{xx}^j \Sigma_{t-1/t-1}] \\
&+ \frac{1}{4} \sum_i \sum_j e^i e^{j'} \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}] \text{tr}[F_{xx}^j \Sigma_{t-1/t-1}] \\
&+ \Sigma^{\varepsilon\varepsilon} + \frac{1}{4} [\sum_i e^i \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}]] [\sum_j e^j \text{tr}[F_{xx}^j \Sigma_{t-1/t-1}]]' \\
&- \frac{1}{2} [\sum_j e^j \text{tr}[F_{xx}^j \Sigma_{t-1/t-1}]] [\sum_i e^i \text{tr}[F_{xx}^i \Sigma_{t-1/t-1}]]'.
\end{aligned} \tag{43}$$

This shows that (35) is true.

(35) can also be rewritten for the augmented system $\hat{z}_{t/t-1} = \begin{bmatrix} \hat{x}_{t/t-1} \\ \dots\dots\dots \\ \theta_{t/t-1}^m \end{bmatrix} :$

$$\Sigma_{t/t-1} = F'_z \Sigma_{t-1/t-1} F_z + \Sigma^{\varepsilon\varepsilon} + \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}[F_{zz}^i \Sigma_{t-1/t-1} F_{zz}^j \Sigma_{t-1/t-1}] \tag{44}$$

and in matrix form:

$$\begin{aligned}
\Sigma_{t/t-1} &= \begin{bmatrix} F_x^x & F_\theta^x \\ F_x^\theta & F_\theta^\theta \end{bmatrix}_{t-1} \begin{bmatrix} \Sigma^{xx} & \Sigma^{x\theta} \\ \Sigma^{\theta x} & \Sigma^{\theta\theta} \end{bmatrix}_{t-1/t-1} \begin{bmatrix} F_x^x & F_\theta^x \\ F_x^\theta & F_\theta^\theta \end{bmatrix}'_{t-1} \\
&+ \begin{bmatrix} \Sigma^{\varepsilon\varepsilon} & 0 \\ 0 & 0 \end{bmatrix} \\
&+ \frac{1}{2} \sum_{i \in I} \sum_{j \in I} e^i e^{j'} \text{tr} \left(\begin{bmatrix} F_{xx}^i & F_{x\theta}^i \\ F_{\theta x}^i & F_{\theta\theta}^i \end{bmatrix} \begin{bmatrix} \Sigma^{xx} & \Sigma^{x\theta} \\ \Sigma^{\theta x} & \Sigma^{\theta\theta} \end{bmatrix}_{t-1/t-1} \right. \\
&\times \left. \begin{bmatrix} F_{xx}^j & F_{x\theta}^j \\ F_{\theta x}^j & F_{\theta\theta}^j \end{bmatrix} \begin{bmatrix} \Sigma^{xx} & \Sigma^{x\theta} \\ \Sigma^{\theta x} & \Sigma^{\theta\theta} \end{bmatrix}_{t-1/t-1} \right).
\end{aligned} \tag{45}$$

Notice that

$$f^x = f(\hat{x}_{t-1/t-1}, \hat{x}_{t/t-1}, u_t^*, \theta, z_t), \quad f^\theta = \theta_{t-1}$$

and

$$F_x^x = \frac{\partial f^x}{\partial x}, \quad F_\theta^x = \frac{\partial f^x}{\partial \theta}, \quad F_x^\theta = \frac{\partial f^\theta}{\partial x} = 0, \quad F_\theta^\theta = \frac{\partial f^\theta}{\partial \theta} = I.$$

Then we obtain

$$\begin{aligned}
\Sigma_{t/t-1} &= \begin{bmatrix} \Sigma^{xx} & \vdots & \Sigma^{x\theta} \\ \dots & & \dots \\ \Sigma^{\theta x} & \vdots & \Sigma^{\theta\theta} \end{bmatrix}_{t/t-1} \\
&= \begin{bmatrix} F_x^x \Sigma^{xx} (F_x^x)' + F_\theta^x \Sigma^{\theta x} (F_\theta^x)' + F_x^x \Sigma^{x\theta} (F_\theta^x)' + F_\theta^x \Sigma^{\theta\theta} (F_\theta^x)' & \vdots & F_x^x \Sigma^{x\theta} + F_\theta^x \Sigma^{\theta\theta} \\ \dots & & \dots \\ \Sigma^{\theta x} (F_x^x)' + \Sigma^{\theta\theta} (F_\theta^x)' & \vdots & \Sigma^{\theta\theta} \end{bmatrix}_{t-1/t-1} \\
&+ \begin{bmatrix} \Sigma^{\varepsilon\varepsilon} & \vdots & 0 \\ \dots & & \dots \\ 0 & \vdots & 0 \end{bmatrix}_{t-1} + \begin{bmatrix} \Pi & \vdots & 0 \\ \dots & & \dots \\ 0 & \vdots & 0 \end{bmatrix},
\end{aligned} \tag{46}$$

where

$$\begin{aligned}
\Pi &= \frac{1}{2} \sum_{i \in I} \sum_{j \in I} e^i e^{j'} \text{tr} \left(\begin{bmatrix} F_{xx}^i & F_{x\theta}^i \\ F_{\theta x}^i & F_{\theta\theta}^i \end{bmatrix} \begin{bmatrix} \Sigma^{xx} & \Sigma^{x\theta} \\ \Sigma^{\theta x} & \Sigma^{\theta\theta} \end{bmatrix}_{t-1/t-1} \right. \\
&\quad \times \left. \begin{bmatrix} F_{xx}^j & F_{x\theta}^j \\ F_{\theta x}^j & F_{\theta\theta}^j \end{bmatrix} \begin{bmatrix} \Sigma^{xx} & \Sigma^{x\theta} \\ \Sigma^{\theta x} & \Sigma^{\theta\theta} \end{bmatrix}_{t-1/t-1} \right) \\
&= \frac{1}{2} \sum_{i \in X} \sum_{j \in X} e^i e^{j'} \text{tr} \left[\begin{bmatrix} F_{xx}^i \Sigma^{xx} + F_{x\theta}^i \Sigma^{\theta x} & F_{xx}^i \Sigma^{x\theta} + F_{x\theta}^i \Sigma^{\theta\theta} \\ F_{\theta x}^i \Sigma^{xx} + F_{\theta\theta}^i \Sigma^{\theta x} & F_{\theta x}^i \Sigma^{x\theta} + F_{\theta\theta}^i \Sigma^{\theta\theta} \end{bmatrix} \right. \\
&\quad \times \left. \begin{bmatrix} F_{xx}^j \Sigma^{xx} + F_{x\theta}^j \Sigma^{\theta x} & F_{xx}^j \Sigma^{x\theta} + F_{x\theta}^j \Sigma^{\theta\theta} \\ F_{\theta x}^j \Sigma^{xx} + F_{\theta\theta}^j \Sigma^{\theta x} & F_{\theta x}^j \Sigma^{x\theta} + F_{\theta\theta}^j \Sigma^{\theta\theta} \end{bmatrix}_{t-1/t-1} \right].
\end{aligned} \tag{47}$$

Using this result for the system considered in this paper, $F_{\theta\theta} = 0$ and $\Sigma_{t-1/t-1}^{xx} = \Sigma_{t-1/t-1}^{x\theta} = 0$, we get

$$\Pi = \frac{1}{2} \sum_{i \in I} \sum_{j \in I} e^i e^{j'} \text{tr}(0) = 0.$$

Then the component matrices can be rewritten as

$$\Sigma_{t/t-1}^{xx} = F_{\theta t-1}^x \Sigma_{t-1/t-1}^{\theta\theta} (F_{\theta t-1}^x)' + \Sigma^{\varepsilon\varepsilon}, \quad (48)$$

$$\Sigma_{t/t-1}^{x\theta} = (\Sigma_{t/t-1}^{\theta x})' = F_{\theta t-1}^x \Sigma_{t-1/t-1}^{\theta\theta}, \quad (49)$$

$$\Sigma_{t/t-1}^{\theta\theta} = \Sigma_{t-1/t-1}^{\theta\theta}. \quad (50)$$

Step 2: Get the corrected values $\Sigma_{t/t}^{xx}$, $\Sigma_{t/t}^{x\theta}$ and $\Sigma_{t/t}^{\theta\theta}$.

In order to calculate these three matrices, one can use the Bayesian method (in case of a normal distribution, it is equivalent to the Kalman Filter), adopting either weighted-least-square or maximum-likelihood estimate (see Bryson and Ho (1975, Chapter 12)).

First, we consider the general case with measurement error. We consider the system of 'real' observations

$$y_t = h_t(x_t) + \zeta_t \quad (51)$$

with $E(\zeta) = 0$ and $cov(\zeta) = \Sigma^{\zeta\zeta}$.

We will later take into account that in our case y_t is equal to the actual observation $x_t^{a*} = f(\hat{x}_{t-1/t-1}, x_t^{a*}, u_t^*, \hat{\theta}, z_t) + \varepsilon_t$ and thus applying to (51), we set $h_t(x_t) = x_t$ and $\zeta_t = 0$. We use here ε_t as short-hand notation for the generated ε_t^m .

Since the conditional distribution $p(x/y)$ is Gaussian, the unconditional maximum-likelihood and weighted-least-square estimates coincide and are given by the conditional mean $E(x|y) = \hat{x}_{t/t}$. Thus according to Kendrick (1981, Appendix D), $p(x/y)$ has the following form:

$$p(x/y) = \frac{1}{(2\pi)^{\frac{n}{2}} |\zeta|^{\frac{1}{2}}} \quad (52)$$

$$\times \exp\left\{-\frac{1}{2}([x - \bar{x} - P^{xy}(P^{yy})^{-1}[y - \bar{y}]]\zeta^{-1}[x - \bar{x} - P^{xy}(P^{yy})^{-1}[y - \bar{y}]])\right\},$$

where $P = \begin{bmatrix} P^{yy} & P^{yx} \\ P^{xy} & P^{xx} \end{bmatrix}$ is the covariance of (y, x) , $\bar{x} = E(x)$ and $\bar{y} = E(y)$.

(52) is a normal distribution with mean

$$E(x/y) = \bar{x} + P^{xy}(P^{yy})^{-1}[y - \bar{y}]$$

and covariance

$$\Sigma = P^{xx} - P^{xy}(P^{yy})^{-1}(P^{xy})'.$$

After some calculations the following can be obtained (see Kendrick (1981, Appendix D, equations (D-39)-(D-41))):

$$E(x|y) = \bar{x} + [P^{xx}h_x][h'_x P^{xx}h_x + \Sigma^{\zeta\zeta} + \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}[h_{xx}^i P^{xx} h_{xx}^j P^{xx}]]^{-1} [y - \bar{y}] \quad (53)$$

and

$$\Sigma = P^{xx} - [P^{xx}h_x][h_x P^{xx}h'_x + \Sigma^{\zeta\zeta} + \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}[h_{xx}^i P^{xx} h_{xx}^j P^{xx}]]^{-1} [h_x P^{xx}], \quad (54)$$

where $P^{xx} = E\{[x - \bar{x}][x - \bar{x}]'\}$, $P^{xy} = P^{xx}h_x$ and

$$P^{yy} = h_x P^{xx} h'_x + \Sigma^{\zeta\zeta} + \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}[h_{xx}^i P^{xx} h_{xx}^j P^{xx}].$$

h_x and h_{xx} are the first and the second derivatives, respectively.

(53) and (54) are the corrections/updates of the mean and the covariance in the general case. We have to adapt the equation (54) to our system.

With the following notations

$$P^{xx} = \Sigma_{t/t-1}, \quad \Sigma = \Sigma_{t/t}, \quad h_x = h_{x,t},$$

we obtain the estimate of Σ from (54):

$$\Sigma_{t/t} = [I - V_t h_{x,t}] \Sigma_{t/t-1}, \quad (55)$$

where

$$\begin{aligned}
V_t &= \Sigma_{t/t-1} h'_{x,t} [h_{x,t} \Sigma_{t/t-1} h'_{x,t} + \Sigma^{\zeta\zeta} \\
&+ \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}(h_{xx}^i \Sigma_{t/t-1} h_{xx}^j \Sigma_{t/t-1})]^{-1}.
\end{aligned} \tag{56}$$

Moreover, we know that in our case

$$h_t(x_t) = I x_t$$

and ζ_t does not exist. Therefore the observation relationship for the aug-

mented system $\hat{z}_{t/t-1} = \begin{bmatrix} \hat{x}_{t/t-1} \\ \dots\dots\dots \\ \theta_{t/t-1}^m \end{bmatrix}$ is (see Kendrick (1981, Appendix L))

$$y_t = \begin{bmatrix} I & \vdots & 0 \end{bmatrix} \begin{bmatrix} x_t \\ \theta_t \end{bmatrix}$$

and the derivative is

$$h_z = \begin{bmatrix} h_x & \vdots & h_\theta \end{bmatrix} = \begin{bmatrix} I & \vdots & 0 \end{bmatrix}$$

with $h_x = \frac{\partial h}{\partial x}$ and $h_\theta = \frac{\partial h}{\partial \theta}$.

The second derivatives are defined as

$$h_{zz}^i = \begin{bmatrix} h_{xx}^i & \vdots & h_{x\theta}^i \\ \dots & & \dots \\ h_{\theta x}^i & \vdots & h_{\theta\theta}^i \end{bmatrix} = \begin{bmatrix} 0 & \vdots & 0 \\ \dots & & \dots \\ 0 & \vdots & 0 \end{bmatrix}.$$

Therefore $h_{zz}^i \Sigma_{t/t-1} = 0$ and thus $\text{tr}(h_{zz}^i \Sigma_{t/t-1} h_{zz}^j \Sigma_{t/t-1}) = 0$. Moreover, (55) and (56) can be rewritten for the system in z_t :

$$\Sigma_{t/t} = [I - V_t h_{z,t}] \Sigma_{t/t-1}, \quad (57)$$

where

$$V_t = \Sigma_{t/t-1} h'_{z,t} \times [h_{z,t} \Sigma_{t/t-1} h'_{z,t} + \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}(h_{zz}^i \Sigma_{t/t-1} h_{zz}^j \Sigma_{t/t-1})]^{-1}. \quad (58)$$

If one uses the information about h specified above, it yields

$$V_t = \begin{bmatrix} \Sigma^{xx} & \vdots & \Sigma^{x\theta} \\ \dots & & \dots \\ \Sigma^{\theta x} & \vdots & \Sigma^{\theta\theta} \end{bmatrix}_{t/t-1} \begin{bmatrix} I \\ \dots \\ 0 \end{bmatrix} \left[\begin{bmatrix} I & \vdots & 0 \end{bmatrix} \times \begin{bmatrix} \Sigma^{xx} & \vdots & \Sigma^{x\theta} \\ \dots & & \dots \\ \Sigma^{\theta x} & \vdots & \Sigma^{\theta\theta} \end{bmatrix} \begin{bmatrix} I \\ \dots \\ 0 \end{bmatrix} \right]^{-1}. \quad (59)$$

Thus

$$\begin{aligned}
 V_t &= \begin{bmatrix} \Sigma_{t/t-1}^{xx} \\ \dots\dots\dots \\ \Sigma_{t/t-1}^{\theta x} \end{bmatrix} [\Sigma_{t/t-1}^{xx}]^{-1} \\
 &= \begin{bmatrix} I \\ \dots\dots\dots \\ \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} \end{bmatrix}.
 \end{aligned} \tag{60}$$

Substitution of (60) into (57) yields

$$\begin{aligned}
 \Sigma_{t/t} &= \begin{bmatrix} \Sigma_{t/t}^{xx} & \vdots & \Sigma_{t/t}^{x\theta} \\ \dots & & \dots \\ \Sigma_{t/t}^{\theta x} & \vdots & \Sigma_{t/t}^{\theta\theta} \end{bmatrix} = [I - V_t h_{z,t}] \Sigma_{t/t-1} = \begin{bmatrix} I & \vdots & 0 \\ \dots & & \dots \\ 0 & \vdots & I \end{bmatrix} \\
 &- \begin{bmatrix} I \\ \dots\dots\dots \\ \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} \end{bmatrix} \begin{bmatrix} I & \vdots & 0 \end{bmatrix} \begin{bmatrix} \Sigma_{t/t-1}^{xx} & \vdots & \Sigma_{t/t-1}^{x\theta} \\ \dots & & \dots \\ \Sigma_{t/t-1}^{\theta x} & \vdots & \Sigma_{t/t-1}^{\theta\theta} \end{bmatrix}_{t/t-1} \\
 &= \begin{bmatrix} 0 & \vdots & 0 \\ \dots\dots\dots & & \dots \\ -\Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} & \vdots & I \end{bmatrix} \begin{bmatrix} \Sigma_{t/t-1}^{xx} & \vdots & \Sigma_{t/t-1}^{x\theta} \\ \dots & & \dots \\ \Sigma_{t/t-1}^{\theta x} & \vdots & \Sigma_{t/t-1}^{\theta\theta} \end{bmatrix}_{t/t-1}.
 \end{aligned} \tag{61}$$

Then the following holds:

$$\Sigma_{t/t}^{xx} = 0, \quad (62)$$

$$\Sigma_{t/t}^{\theta x} = 0, \quad (63)$$

and

$$\Sigma_{t/t}^{\theta\theta} = \Sigma_{t/t-1}^{\theta\theta} - \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} \Sigma_{t/t-1}^{x\theta}. \quad (64)$$

Step 3: Get the corrected $\hat{\theta}_{t/t}$.

Use (53), i.e. the values estimated with the Kalman Filter from the previous step:

$$E(x|y) = \bar{x} + [P^{xx} h_x] [h_x' P^{xx} h_x + \Sigma^{\zeta\zeta} + \frac{1}{2} \sum_i \sum_j e^i e^{j'} \text{tr}[h_{xx}^i P^{xx} h_{xx}^j P^{xx}]]^{-1} [y - \bar{y}].$$

With the following notation

$$E(x|y) = \hat{x}_{t/t}, \quad \bar{x} = \hat{x}_{t/t-1}, \quad P^{xx} = \Sigma_{t/t-1},$$

we obtain $\hat{x}_{t/t}$:

$$\hat{x}_{t/t} = \hat{x}_{t/t-1} + V_t [y_t - h_{x,t} \hat{x}_{t/t-1}], \quad (65)$$

where V_t is defined in Eq. (56).

For the augmented system in $\hat{z} = \begin{bmatrix} \hat{x} \\ \theta^m \end{bmatrix}$, we obtain

$$\hat{z}_{t/t} = \hat{z}_{t/t-1} + V_t[y_t - h_{z,t}\hat{z}_{t/t-1}],$$

where V_t is defined in Eq. (60).

Again with $h_{z,t} = [h_{x,t} \ : \ h_{\theta,t}] = [I \ : \ 0]$ we get

$$\begin{aligned} \begin{bmatrix} \hat{x} \\ \theta^m \end{bmatrix}_{t/t} &= \begin{bmatrix} \hat{x} \\ \theta^m \end{bmatrix}_{t/t-1} \\ &+ \begin{bmatrix} I \\ \dots\dots\dots \\ \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} \end{bmatrix} [y_t - [I \ : \ 0] \begin{bmatrix} \hat{x}_{t/t-1} \\ \theta_{t/t-1}^m \end{bmatrix}] \\ &= \begin{bmatrix} y_t \\ \dots\dots\dots \\ \theta_{t/t-1}^m + \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} [y_t - \hat{x}_{t/t-1}] \end{bmatrix}. \end{aligned} \quad (66)$$

Thus, with $y_t = x_t^{a*}$,

$$\hat{x}_{t/t} = x_t^{a*}, \quad (67)$$

$$\theta_{t/t}^m = \theta_{t/t-1}^m + \Sigma_{t/t-1}^{\theta x} (\Sigma_{t/t-1}^{xx})^{-1} [x_t^{a*} - \hat{x}_{t/t-1}].$$

8 Acknowledgements

An earlier version of this paper was presented at the 3rd International Conference on Computational and Financial Econometrics (CFE 09), Limassol, Cyprus. Financial support by the EU-HRM Program, Marie Curie Research Training Network COMISEF is gratefully acknowledged.

References

- Amman, H., 1996. Numerical methods for linear-quadratic models. In: Amman, H. M., Kendrick, D. A., Rust, J. (Eds.), *Handbook of Computational Economics*. Vol. I. Elsevier, Amsterdam, Ch. 13, pp. 587–618.
- Amman, H. M., Kendrick, D. A., 2000. Stochastic policy design in a learning environment with rational expectations. *Journal of Optimization Theory and Applications* 105 (3), 509–520.
- Aoki, M., 1989. *Optimization of Stochastic Systems*. Topics in Discrete-Time Dynamics, 2nd Edition. Academic Press, New York.
- Bryson, A. E., Ho, Y.-C., 1975. *Applied Optimal Control*. Optimization, Estimation, and Control. Hemisphere, Washington, DC.

- Chow, G. C., 1975. *Analysis and Control of Dynamic Economic Systems*. John Wiley & Sons, New York.
- Chow, G. C., 1981. *Econometric Analysis by Control Methods*. John Wiley & Sons, New York.
- Coomes, P. A., 1987. PLEM: A computer program for passive learning, stochastic control experiments. *Journal of Economic Dynamics and Control* 11, 223–227.
- Gilli, M., Winker, P., 2009. Heuristic optimization methods in econometrics. In: Belsley, D. A., Kontoghiorghes, E. J. (Eds.), *Handbook of Computational Econometrics*. John Wiley, Chichester, Ch. 3, pp. 81–119.
- Kendrick, D. A., 1981. *Stochastic Control for Economic Models*. McGraw-Hill, New York.
- Kendrick, D. A., Amman, H. M., 2006. A classification system for economic stochastic control models. *Computational Economics* 27, 453–481.
- Matulka, J., Neck, R., 1992. OPTCON: An algorithm for the optimal control of nonlinear stochastic models. *Annals of Operations Research* 37, 375–401.
- Neck, R., 1984. Stochastic control theory and operational research. *European Journal of Operations Research* 17, 283–301.
- Neck, R., Karbuz, S., 1995. Optimal budgetary and monetary policies under uncertainty: a stochastic control approach. *Annals of Operations Research* 58, 379–402.