Adaptive Koopman-Based Models for Holistic Controller and Observer Design

Annika Junker, Keno Pape, Julia Timmermann, Ansgar Trächtler

Heinz Nixdorf Institute, Paderborn University, Germany (e-mail: \{annika.junker, keno.pape, julia.timmermann, ansgar.traechtler\}@hni.upb.de)

Abstract: We present a method to obtain a data-driven Koopman operator-based model that adapts itself during operation and can be straightforwardly used for the controller and observer design. The adaptive model is able to accurately describe different state-space regions and additionally consider unpredictable system changes that occur during operation. Furthermore, we show that this adaptive model is applicable to state-space control, which requires complete knowledge of the state vector. For changing system dynamics, the state observer therefore also needs to have the ability to adapt. To the best of our knowledge, there have been no approaches presently available that holistically use an adaptive Koopman-based plant model for the state-space design of both the controller and observer. We demonstrate our method on a test rig: controller and observer adequately adapt during operation, so that outstanding control performance is achieved even in the case of strong occurring systems changes.

Keywords: data-based control, adaptive control, time-varying systems, predictive control

1. INTRODUCTION

Data-driven methods are increasingly used for modeling technical systems due to their high performance. However, in the field of control engineering, the established design methods usually require analytical physics-derived models, so it is essential to merge these two areas. Challenging for mechatronic systems is that unpredictable changes can occur over the product life cycle, e.g., the installation of a new component, wear or temperature fluctuations during operation. This requires the information processing system to measure these changes and constantly update the internal model used for the controller and observer.

Research in the field of the Koopman operator provides promising strategies for model building that is appropriate from a control engineering point of view, see Junker et al. (2022b), while at the same time utilizing the strengths of data-driven approaches. The underlying idea is that nonlinear dynamics is approximated by a linear but higher-dimensional operator (Williams et al. (2015)), enabling the control design for a nonlinear system using linear design strategies, e.g., model predictive control (MPC) and linear-quadratic regulator (LQR) design, see Korda and Mezić (2018) and Mamakoukas et al. (2019), respectively. The associated method is called Extended Dynamic Mode Decomposition (EDMD).

In the field of adaptive control, there are well-established methods to deal with changing system parameters, e.g., unpredictable fluctuations in operating or environmental conditions, or if an insufficiently accurate linear model is assumed to describe nonlinear dynamics. Direct adaptive control means adjusting the controller parameters in real-time, e.g., depending on the error between the system output and the output of the desired reference model, so is called MRAC (Model Reference Adaptive Control, see Åström (2008)). In contrast, indirect adaptive control is a two-step process: first, the parameters of the plant model are estimated depending on the error between the plant output and the plant model output, and then the controller parameters are calculated accordingly. Since we value interpretable models with deep insights into system dynamics, e.g., the state, and want to use our adaptive internal plant model for both controller and observer design, we aim for indirect adaptive control approaches.

Several approaches to adaptive Koopman-based models already exist to design the controller. Peitz and Klus (2018), Zhang et al. (2019) and Cisneros et al. (2020) utilize a recursive method to adapt a Koopman-based model online. To overcome the need for a completely measurable state, the authors in Cisneros et al. (2020) propose to use an input-output Koopman-based model. Calderon et al. (2021) use an adaptive Koopman-based model with a velocity-based linearization to obtain a quasi-linear parameter-varying model, which is simulatively tested within a Koopman-MPC. To account for uncertainties in modeling and data, Zheng et al. (2022) develop a recursive set-membership identification, applying this method to approximate a Koopman-based model within a Lyapunov-based MPC design and simulatively...
showing that model adaptation improves prediction accuracy and control performance. Gong et al. (2022) use a real-time strategy for combinatorial Koopman-based modeling through iterative learning. The authors obtain a data-driven Koopman-based model for a linear and nonlinear subspace, which is subsequently used for adaptive LQR synthesis. Simulation studies show improvement in model accuracy and control performance. In contrast to the previous parameter adaptation methods, Gueho et al. (2021) use an improved subspace method to obtain a time-variant approximation of the Koopman operator for the highest possible prediction quality.

In a state-space control scheme, all states of the system are fed back. As usually not all states can be directly measured, a state observer is required. There are only a few results published to design an observer using a Koopman-based model. The existing methods are based on static models identified offline and are furthermore solely used for open-loop prediction. Surana and Banaszuk (2016) and Surana (2016) describe a design strategy for Luenberger or Kalman-like linear observers applied to autonomous and actuated nonlinear systems. Using DMD and EDMD, respectively, Iungo et al. (2015) and Netto and Mili (2018) identify a static Koopman-based model for use within a Kalman filter to improve the prediction accuracy of the model with recent measurement data.

In the following, continuous-time control-affine systems

$$\dot{x} = f(x) + Bu$$

(1)

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^p$ and a constant input matrix $B \in \mathbb{R}^{n \times p}$ are considered. For the linear Koopman-based approximation of the dynamics, $N$ observable functions $\Psi(x) = [\psi_1(x), \psi_2(x), \ldots, \psi_N(x)]^\top$ are defined, which lift the states into the higher-dimensional space. It may be useful to exploit the time derivatives of the original nonlinear dynamics for this purpose, if they are known. The algorithm approximates the dynamics of the lifted states $\Psi(x)$ as a discrete-time system

$$\Psi(x_{k+1}) \approx K_1 \Psi(x_k) + B_1 u_k = [K_1, B_1] \begin{bmatrix} \Psi(x_k) \\ u_k \end{bmatrix},$$

(2)

where the index $t$ denotes the discrete-time property.
With $M$ sequential measurement points

$$X = \{x_1, x_2, \cdots, x_{M-1}\} \in \mathbb{R}^{n \times (M-1)},$$

(3a)

$$X' = \{x_2, x_3, \cdots, x_M\} \in \mathbb{R}^{n \times (M-1)},$$

(3b)

$$U = \{u_1, u_2, \cdots, u_{M-1}\} \in \mathbb{R}^{p \times (M-1)},$$

(3c)

for state and input results

$$\Psi(X') \approx K_t \Psi(X) + B_t U = [K_t, B_t] \left[ \begin{array}{c} \Psi(X) \\ U \end{array} \right],$$

(4)

$$\Rightarrow [K_t, B_t] \approx \Psi(X') \left[ \begin{array}{c} \Psi(X) \\ U \end{array} \right]^\dagger,$$

(5)

where $K_t \in \mathbb{R}^{N \times N}$ is the approximated Koopman operator, $B_t \in \mathbb{R}^{N \times p}$ the lifted input matrix and $^\dagger$ denotes the Moore-Penrose inverse matrix. The resulting discrete-time system description for EDMD prediction is given by

$$\hat{\Psi}(x_{k+1}) = K_t \hat{\Psi}(x_k) + B_t u_k.$$  

(6)

The hat on the symbols emphasizes that the quantities are estimated, cf. (2), as the observable functions usually do not span a Koopman invariant subspace.

3. RECURSIVE KOOPMAN MODEL FOR VARYING SYSTEM DYNAMICS

Next, we derive our algorithm for a recursive Koopman-based plant model, which we refer to as rEDMD. The main procedure, sec. 3.1, has been derived by Calderon et al. (2021) analogously to the classical recursive least squares method by Isernmann and Münchhof (2011). Due to this analogy, different modifications of the classical recursive least squares method can be integrated into the adaptation algorithm developed in this paper to ensure a stable and reasonable adaptation process, cf. Sec. 3.2.

3.1 Model Update Rules

The goal is to update the model using continuously recorded measurement data. For this purpose, analogous to (3), the snapshot matrices are extended in each time step

$$X_k = \{x_1, x_2, \cdots, x_{k-1}\} \in \mathbb{R}^{n \times (k-1)},$$

(7a)

$$X'_k = \{x_2, x_3, \cdots, x_k\} \in \mathbb{R}^{n \times (k-1)},$$

(7b)

$$U_k = \{u_1, u_2, \cdots, u_{k-1}\} \in \mathbb{R}^{p \times (k-1)},$$

(7c)

$$X'_{k+1} = \{x'_1, x'_2, \cdots, x'_k\} \in \mathbb{R}^{n \times k},$$

(7d)

$$X_{k+1} = [X_k, x_k] \in \mathbb{R}^{n \times k},$$

(7e)

$$U_{k+1} = [U_k, u_k] \in \mathbb{R}^{p \times k},$$

(7f)

enabling the Koopman-based model to be calculated

$$[K_{t,k}, B_{t,k}] = \Psi(X'_k) \left[ \begin{array}{c} \Psi(X_k) \\ U_k \end{array} \right]^\dagger,$$

(8)

$$[K_{t,k+1}, B_{t,k+1}] = \Psi(X'_{k+1}) \left[ \begin{array}{c} \Psi(X_{k+1}) \\ U_{k+1} \end{array} \right]^\dagger,$$

(9)

where $k$ is the current time step. With (A.1) follows

$$[K_{t,k}, B_{t,k}] = \Psi(X'_k) \left[ \begin{array}{c} \Psi^\top(X_k), U_k \end{array} \right] \Gamma_k,$$

(10)

$$[K_{t,k+1}, B_{t,k+1}] = \Psi(X'_{k+1}) \left[ \begin{array}{c} \Psi^\top(X_{k+1}), U_{k+1} \end{array} \right] \Gamma_{k+1},$$

(11)

with

$$\Gamma_k = \left( \begin{array}{c} \Psi(X_k) \\ U_k \end{array} \right) \left[ \begin{array}{c} \Psi^\top(X_k), U_k \end{array} \right]^{-1},$$

(12)

$$\Gamma_{k+1} = \left( \begin{array}{c} \Psi(X_k) \\ U_k \end{array} \right) \left[ \begin{array}{c} \Psi^\top(X_k), U_k \end{array} \right]^{-1},$$

(13)

$$\Gamma_k = \left( \begin{array}{c} \Psi(X_k) \\ U_k \end{array} \right) \left[ \begin{array}{c} \Psi^\top(X_k), U_k \end{array} \right]^{-1}.$$  

(14)

and with (10)

$$[K_{t,k}, B_{t,k}] \Gamma_k = \Psi(X'_k) \left[ \begin{array}{c} \Psi^\top(X_k), U_k \end{array} \right].$$

(15)

Reshaping (11) yields

$$[K_{t,k+1}, B_{t,k+1}] = \Psi(X'_k) \left[ \begin{array}{c} \Psi^\top(X_k), U_k \end{array} \right] \Gamma_{k+1} + \Psi(x_{k+1}) \left[ \begin{array}{c} \Psi^\top(x_k), u_k \end{array} \right] \Gamma_{k+1},$$

(16)

and with (14)-(15) follows

$$[K_{t,k+1}, B_{t,k+1}] = [K_{t,k}, B_{t,k}]$$

$$+ \left( \Psi(x_{k+1}) \right) \left[ \begin{array}{c} \Psi(x_k), u_k \end{array} \right] \gamma_k,$$

(17)

where

$$\gamma_k = \left[ \begin{array}{c} \Psi^\top(x_k), u_k \end{array} \right] \Gamma_{k+1}$$

(18)

is the correction vector and $\Gamma_{k+1}$ is calculated from (14) with (A.3), yielding

$$\Gamma_{k+1} = \Gamma_k - \frac{\Gamma_k \left[ \begin{array}{c} \Psi(x_k), u_k \end{array} \right] \left[ \begin{array}{c} \Psi^\top(x_k), u_k \end{array} \right] \Gamma_k}{\left[ \begin{array}{c} \Psi^\top(x_k), u_k \end{array} \right] \Gamma_k \left[ \begin{array}{c} \Psi(x_k), u_k \end{array} \right] + 1}$$

(19)

and with (19) results

$$\gamma_k = \left[ \begin{array}{c} \Psi^\top(x_k), u_k \end{array} \right] \Gamma_k$$

(20)

and

$$\Gamma_{k+1} = \Gamma_k \left( I - \left[ \begin{array}{c} \Psi^\top(x_k), u_k \end{array} \right] \gamma_k \right).$$

(21)

We now introduce the forgetting factor $\lambda$, which enables past measurements to be forgotten in the event of system changes, where the smaller $\lambda$ is, the less influence past measurements have on the current model estimate. Thus, the adaptation speed can be systematically controlled. $\lambda$ affects the calculation of $\gamma_k$ and $\Gamma_{k+1}$, resulting in the equations for the model update

$$\gamma_k = \left[ \begin{array}{c} \Psi^\top(x_k), u_k \end{array} \right] \Gamma_k$$

(22)
Thus, (22), (23) and (24) form the core of our algorithm.

### 3.2 Extended Algorithm

Our algorithm includes certain extensions to ensure that the adaption process is stable and reasonable. Below, we present the different extensions resulting in our final algorithm.

#### Checking the Need to Update the Model

To decide whether the model needs to be updated, the prediction accuracy of the current model serves as a criterion, as proposed by Cisneros et al. (2020). During operation, $M_{op}$ measurements of the states and the inputs are stored

\[
\begin{align*}
X_k &= [x_{k-M_{op}}, x_{k-M_{op}+1}, \ldots, x_k]^\top, \\
\dot{U}_k &= [u_{k-M_{op}}, u_{k-M_{op}+1}, \ldots, u_k]^\top.
\end{align*}
\]  
\tag{25a}
\tag{25b}

The current EDMD model is simulated according to (6) and the states $\hat{x}$ are extracted from the lifted states with a projection matrix $P_x$.

\[
\begin{align*}
\hat{x}_{k-M_{op}+1} &= P_x (K_{t,k} \Psi (x_{k-M_{op}}) + B_{t,k} u_{k-M_{op}}), \\
\hat{x}_{k-M_{op}+2} &= P_x (K_{t,k} \Psi (x_{k-M_{op}+1}) + B_{t,k} u_{k-M_{op}+1}), \\
& \vdots \\
\hat{x}_k &= P_x (K_{t,k} \Psi (x_{k-1}) + B_{t,k} u_{k-1}),
\end{align*}
\]  
\tag{26}
\tag{27}

resulting in

\[
\begin{align*}
\hat{X}_k &= \left[ \begin{array}{c} x_{k-M_{op}} \; \; x_{k-M_{op}+1} \; \; \ldots \; \; x_k \end{array} \right]^\top.
\end{align*}
\]  
\tag{28}

If the defined accuracy limit $\varepsilon_{\text{low}}$ is exceeded, i.e.,

\[
\left\| \hat{X}_k - \hat{X}_k \right\|_\infty \geq \varepsilon_{\text{low}},
\]  
\tag{29}

the new measurement [$\Psi^T(x_{k+1}), u_{k+1}^T$] is used for a model update.

#### Varying Forgetting Factor

Due to the better performance, we use a variable forgetting factor $\lambda_k$, see Fortescue et al. (1981),

\[
\lambda_{k+1} = 1 - \frac{1}{\sum_{j=0} \lambda_j} \left( 1 - \left[ \Psi^T(x_k), u_k^T \right] \gamma_k \right) \| P_{\text{post,k}}^2 \),
\]  
\tag{30}

where $P_{\text{post,k}}$ is the error between the measured and the predicted system output. $P_y$ is the projection matrix to extract the output $y$ from the lifted states and $\Sigma_{\text{low}} = \sigma_{\text{low}} N_0$ is an assumed measurement noise with the variance $\sigma_{\text{low}}$, which is a posteriori determined by the error behavior (32), and $N_0$ is an adjustable sensitivity factor. If the accuracy, cf. (30), even exceeds $\varepsilon_{\text{high}}$, the adaptation speed is increased by multiplying $\Sigma_{\text{low}}$ by a gain factor $\mu_{\Sigma} > 1$. Note here that the described strategy is only applicable for systems with a one-dimensional output, otherwise the use of multiple varying forgetting factors would be necessary.

#### Setting the Trace

Using a forgetting factor can lead to a covariance windup in operating situations with insufficient excitation because the elements of the covariance matrix $\Gamma_k$ may strongly increase, causing the estimation algorithm to react sensitively to changes in the measured data. Therefore, following Lozano-Leal and Goodwin (1985), we exploit a constant-trace algorithm for the covariance matrix to limit the sensitivity and thus the adaptation intensity of our algorithm. In case the current trace of the covariance matrix exceeds a maximum value, the elements of the covariance matrix are multiplied by a reduction factor $\mu_T < 1$.

\[
\Gamma_k \leftarrow \Gamma_k \mu_T \text{ with } \mu_T = \frac{\text{tr}(\Gamma_{\text{max}})}{\text{tr}(\Gamma_k)}.
\]  
\tag{33}

Algorithm 1 summarizes rEDMD with extensions.

**Algorithm 1 rEDMD algorithm with extensions**

```
while $k \leq T_{\text{sim}}$ do
  if $M_{op}$ samples of data were collected then
    if $\left\| \hat{X}_k - \hat{X}_k \right\|_\infty \geq \varepsilon_{\text{low}}$ then
      Set $\Gamma_k$ with (33)
    end if
    end if
    if $\left\| \hat{X}_k - \hat{X}_k \right\| \geq \varepsilon_{\text{high}}$ then
      Set $\Sigma_0,k \leftarrow \frac{1}{\mu_{\Sigma}} \Sigma_0,k$.
    end if
    Update rEDMD using (22), (23) and (24)
    Calculate $\lambda_{k+1}$ with (31)
  else
    No Update of rEDMD and $\lambda$
  end if
end while
```

4. RESULTS FOR A HOLISTIC ADAPTIVE CONTROLLER AND OBSERVER DESIGN

For changing systems where not all states are known and state feedback control is used, it is essential to design both the controller and the required observer as adaptive systems. This requires a holistic strategy, i.e., both designs must be equally regarded. The linearity of the rEDMD model makes it possible to systematically employ existing linear design procedures. Thus, we propose to use a model predictive control in conjunction with a Kalman filter. Note here that the linear design process is realized in a higher-dimensional space, i.e., with more states than the original system, so is $N > n$.

We experimentally demonstrate the success of our holistic method on the golf robot, whose goal is to autonomously learn to hit a golf ball into the hole from an arbitrary starting point on the green using combined data-driven and physics-based methods. A detailed description of the golf robot is given in Junker et al. (2022a), where the overall problem is decomposed into separate sub-tasks. The stroke speed control of the golf club is a challenging
task in terms of control engineering because of nonlinear effects, changes in dynamics due to the replacement of the golf club and additionally because not all states can be directly measured. Based on these considerations, we selected this subtask to demonstrate the success of our method.

Fig. 3 illustrates an example stroke with a desired speed of ... with a gain-scheduling approach, cf. Junker et al. (2022a), is shown. The dimension of the original system is \( n = 2 \) and is increased to \( N = 4 \) by the Koopman approach. We initialized the rEDMD model offline using historical open-loop training data and implemented a model predictive control with a Kalman filter using our adaptive Koopman-based model, see Fig. 2. Then we changed the robot by mounting an extra mass and thus doubling the golf club mass and additionally because not all states can be directly measured. Based on these considerations, we selected this subtask to demonstrate the success of our method. We established a method to design an adaptive Koopman-based model and demonstrated the success when using it as an internal model for the controller and observer in a state-space control scheme achieving outstanding control performance on a test rig. The complete observability of the specific rEDMD model for the golf robot has been verified offline. However, the generalized influence of the higher-dimensional model on the observability remains an open research question. Furthermore, it is a matter of interest how to extend our approach to structural system changes such as nonlinearities that have not occurred before. For system descriptions which are valuable from a control engineering point of view, the proposed recursive parameter estimation method may be transferred to data-driven PCHD models (Junker et al. (2022c)).

5. CONCLUSION & OUTLOOK

We established a method to design an adaptive Koopman-based model and demonstrated the success when using it as an internal model for the controller and observer in a state-space control scheme achieving outstanding control performance on a test rig. The complete observability of the specific rEDMD model for the golf robot has been verified offline. However, the generalized influence of the higher-dimensional model on the observability remains an open research question. Furthermore, it is a matter of interest how to extend our approach to structural system changes such as nonlinearities that have not occurred before. For system descriptions which are valuable from a control engineering point of view, the proposed recursive parameter estimation method may be transferred to data-driven PCHD models (Junker et al. (2022c)).

Table 1. Normalized control errors (34) for different stroke speeds at the golf robot without and with occurring system changes. The errors are multiplied by \( 10^2 \) for better readability.

| Stroke speed \( v \) | w/o system changes | w/ system changes |
|----------------------|-------------------|-----------------|
| 1 m/s \( m \)        | 1.38              | 2.18            |
| 3 m/s \( m \)        | 0.34              | 0.67            |
| gain-scheduling strategy | 5.68             | 6.33            |
| 46.00               | 55.08             |
| adaptive controller  | 0.31              | 0.96            |
| non-adaptive observer| 6.36              | 5.08            |
| both adaptive       | 0.34              | 0.67            |
| controller and observer| 1.38          | 2.18            |

REFERENCES

Aström, K.J. (2008). *Adaptive control*. Dover Publ, Mineola, N.Y, 2 edition.
Calderon, H.M., Schulz, E., Oehlschlagel, T., and Werner, H. (2021). Koopman Operator-based Model Predictive Control with Recursive Online Update. In 2021 European Control Conference (ECC), 1543–1549. IEEE.
Cisneros, P.S., Datar, A., Göttsch, P., and Werner, H. (2020). Data-Driven quasi-LPV Model Predictive Control Using Koopman Operator Techniques. *IFAC-PapersOnLine*, 53(2), 6062–6068.
Fortescue, T.R., Kershenbaum, L.S., and Ydstie, B.E. (1981). Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17(6), 831–835.
Gong, X., Wang, X., and Joos, G. (2022). An Online Data-Driven Method for Microgrid Secondary Voltage and Frequency Control with Ensemble Koopman Modeling. *Automatica*, 132, 109315.
Guehö, D., Singla, P., and Majji, M. (2021). Time-Varying Koopman Operator Theory for Nonlinear Systems Prediction. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 2294–2299. IEEE.
Isernmann, R. and Münchhof, M. (2011). *Identification of Dynamic Systems*. Springer, Berlin, Heidelberg.
Iungo, G.V., Santoni-Ortiz, C., Akbar, M., Porté-Agel, F., Rotea, M.A., and Leonard, S. (2015). Data-driven Reduced Order Model for prediction of wind turbine wakes. *Journal of Physics: Conference Series*, 625.
Junker, A., Fittkau, N., Timmermann, J., and Trächtler, A. (2022a). Autonomous Golf Putting with Data-Driven and Physics-Based Methods. In *2022 Sixth IEEE
A Robust Data-Driven Approach for Control Systems Dynamic State Estimation in Power Systems.

Korda, M. and Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93, 149–160.

Lozano-Leal, R. and Goodwin, G. (1985). A globally convergent adaptive pole placement algorithm without a persistency of excitation requirement. *IEEE Transactions on Automatic Control*, 30(8), 795–798.

Mamakoukas, G., Castano, M.L., Tan, X., and Murphey, T.D. (2019). Local Koopman Operators for Data-Driven Control of Robotic Systems. In *Robotics: Science and Systems XV*. Robotics: Science and Systems Foundation.

Netto, M. and Mili, L. (2018). A Robust Data-Driven Koopman Kalman Filter for Power Systems Dynamic State Estimation. *IEEE Transactions on Power Systems*, 1–10.

Peitz, S. and Klus, S. (2018). Feedback control of nonlinear PDEs using data-efficient reduced order models based on the Koopman operator.

Surana, A. (2016). Koopman operator based observer synthesis for control-affine nonlinear systems. In *IEEE 55th Conference on Decision and Control (CDC)*, 6492–6499. IEEE.

Surana, A. and Banaszuk, A. (2016). Linear observer synthesis for nonlinear systems using Koopman Operator framework. *IFAC-PapersOnLine*, 49(18), 716–723.

Williams, M.O., Kevrekidis, I.G., and Rowley, C.W. (2015). A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*, 25(6), 1307–1346.

Zhang, H., Rowley, C.W., Deem, E.A., and Cattafesta, L.N. (2019). Online Dynamic Mode Decomposition for Time-Varying Systems. *SIAM Journal on Applied Dynamical Systems*, 18(3), 1586–1609.

Zheng, Y., Zhang, Y., Li, Q., Li, S., and Luo, M. (2022). Koopman Approximator based Adaptive Model Predictive Control of Continuous Nonlinear Systems.

Appendix A. MATRIX CALCULATION

The Moore-Penrose inverse matrix of a matrix $A$ with full row rank can be calculated by

$$A^+ = A^\top \left( AA^\top \right)^{-1}.$$  

If $A$, $C$ and $(A^{-1} + BC^{-1}D)$ are non-singular quadratic matrices and

$$E = (A^{-1} + BC^{-1}D)^{-1}$$

then

$$E = A - AB(AB + C)^{-1}DA,$$

see Isermann and Münchhof (2011).