



Parameter and State Estimation of Large-Scale Complex Systems Using Python Tools

M. Anushka S. Perera¹ Tor Anders Hauge² Carlos F. Pfeiffer¹

¹Telemark University College, Kjølnes ring 56, P.O. Box 203, N-3901 Porsgrunn, Norway. E-mail: carlos.pfeiffer@hit.no

²Glencore Nikkelverk, Kristiansand, Norway.

Abstract

This paper discusses the topics related to automating parameter, disturbance and state estimation analysis of large-scale complex nonlinear dynamic systems using free programming tools. For large-scale complex systems, before implementing any state estimator, the system should be analyzed for structural observability and the structural observability analysis can be automated using Modelica and Python. As a result of structural observability analysis, the system may be decomposed into subsystems where some of them may be observable — with respect to parameter, disturbances, and states — while some may not. The state estimation process is carried out for those observable subsystems and the optimum number of additional measurements are prescribed for unobservable subsystems to make them observable. In this paper, an industrial case study is considered: the copper production process at Glencore Nikkelverk, Kristiansand, Norway. The copper production process is a large-scale complex system. It is shown how to implement various state estimators, in Python, to estimate parameters and disturbances, in addition to states, based on available measurements.

Keywords: Kalman filter, Modelica, Observability, Python, state and parameter estimation

1 Introduction

Consider a class of nonlinear deterministic systems given by eq.(1), where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $w \in \mathbb{R}^{n_w}$, $p \in \mathbb{R}^{n_p}$, and $y \in \mathbb{R}^{n_y}$ are respectively state, input, process noise, parameter, and output vectors and $C \in \mathbb{R}^{n_y \times n_x}$ is a constant matrix and $f(\cdot)$ and $g(\cdot)$ are known vector functions.¹

$$\begin{aligned} \dot{x} &= f(x, u, w, p) \\ \dot{p} &= 0 \\ y &= h(x) = C \cdot x \end{aligned} \quad (1)$$

u and y are known as well as their time derivatives. x , w , and p are unknowns and the objective is to esti-

¹ $x \equiv x(t)$, $u \equiv u(t)$, etc.

mate them based on u - y information. The state space is augmented with p via $\dot{p} = 0$. However, it is not obvious how to augment w , since w is completely unknown. One possibility is to assume w is a random variable with a given probability characteristic. An example Gelb (2001) is given in eq.(2), where $w_i \in [w_1, w_2, \dots, w_{n_w}]^T$ is a stationary random process with the autocorrelation function $\gamma_i(\tau) = (\sigma_i^2 \delta_i / 2) \cdot e^{-\delta_i |\tau|}$, ϵ_i is a given white Gaussian process and $\delta_i, \sigma_i > 0$.²

$$\dot{w}_i = -\delta_i w_i + \sigma_i \delta_i \cdot \epsilon_i \quad (2)$$

There are other alternatives for disturbance augmentation Bona and Smay (1966) and the choice may depend on whether the augmented system is observable

² $\sigma_i \delta_i \cdot \epsilon_i$, in eq.(2), may be replaced by ϵ_i .

or not — there is no point of augmenting parameters and disturbances if the augmented system becomes unobservable. The complete augmented system is given in eq.(3) and corresponding state space representation is in eq.(4), where $g(\cdot)$ is chosen appropriately by augmenting disturbances.³

$$\begin{aligned} \dot{x} &= f(x, u, w, p) \\ \dot{w} &= g(w) + \epsilon \\ \dot{p} &= 0 \\ y &= C \cdot x + v \end{aligned} \quad (3)$$

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{w} \\ \dot{p} \end{bmatrix}}_{\dot{\hat{x}}} &= \underbrace{\begin{bmatrix} f(x, u, w, p) \\ g(w) \\ 0 \end{bmatrix}}_{\hat{f}(\hat{x}, u)} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{\Gamma} \epsilon \\ \hat{y} &= \underbrace{[C \ 0 \ 0]}_{\hat{h}(\hat{x})} \cdot \underbrace{\begin{bmatrix} x \\ w \\ p \end{bmatrix}}_{\hat{x}} + v \end{aligned} \quad (4)$$

The noise model related to eq.(1) is given by the stochastic nonlinear system described by eq.(5), where ϵ and v (measurement noise) are vectors of white Gaussian processes such that $E\{w\} = 0 \in \mathbb{R}^{n_w}$, $E\{v\} = 0 \in \mathbb{R}^{n_y}$, $E\{ww^T\} = Q \in \mathbb{R}^{n_w \times n_w}$, $E\{vv^T\} = R \in \mathbb{R}^{n_y \times n_y}$, $E\{wv^T\} = S \in \mathbb{R}^{n_w \times n_y}$ (S is a zero matrix if w and v are independent), $\Gamma \in \mathbb{R}^{n_x \times n_w}$ is a constant matrix, and x_0 is independent from w and v . Associated discrete time version of eq.(5) is presented in eq.(6).

$$\begin{aligned} \dot{\hat{x}} &= \hat{f}(\hat{x}, u) + \Gamma \cdot \epsilon \\ \hat{y} &= \hat{h}(\hat{x}) + v, \end{aligned} \quad (5)$$

$$\begin{aligned} \hat{x}_{k+1} &= \hat{f}_k(\hat{x}_k, u_k) + \Gamma \cdot w_{k+1}; \quad k = 0, 1, \dots \\ \hat{y}_k &= \hat{h}_k(\hat{x}_k) + v_k; \quad k = 1, 2, \dots \end{aligned} \quad (6)$$

Where,

$$\hat{f}_k(\hat{x}_k, u_k) = \hat{x}_k + \int_{t_k}^{t_{k+1}} \left[\hat{f}(\hat{x}(\theta), u(\theta)) \right] d\theta$$

and

$$w_{k+1} = \Gamma \int_{\beta(t_k)}^{\beta(t_{k+1})} d\beta.$$

³Note that the number of states of the augmented system could be larger than $n_x + n_p + n_w$. For example, if w_i is augmented by $\dot{w}_i = 0$. Also 0's and 1's, in eq.(3) and eq.(4), represent zero and unit matrices with appropriate dimensions.

β is a Brownian motion process such that $\epsilon \cdot dt = d\beta$.⁴ Now, the system in eq.(1) is simulated for given p , $\{w_k\}$ (w_k with some added noise), $\{u_k\}$, and x_0 . $\{x_k\}$ and $\{y_k\}$ (a fictitious noise is added to y_k) are stored for $k = 1, 2, 3, \dots, n$.⁵ Then, based on the simulated data $\{u_k\}$ and $\{y_k\}$, $\{\hat{x}_k\}$ is estimated, for $k = 1, 2, \dots, n$, to verify the stability properties of state estimators.

The structure of the paper is as follows. Section 2: a brief discussion on nonlinear observability, in particular giving more attention on structural observability. Section 3: the process model. Section 4: Observability analysis. Section 5: a brief introduction to filtering theory. Section 6: structure of the Python code and results. Section 7: conclusions and future work.

2 Nonlinear observability

In order to achieve the state observability of the stochastic system given in eq.(5), it is a must that the corresponding noise-free system is observable [Margaria et al. \(2004\)](#). Hence, consider the noise-free version of eq.(5).⁶

$$\begin{aligned} \dot{\hat{x}} &= \hat{f}(\hat{x}, u) \\ \hat{y} &= \hat{h}(\hat{x}) \end{aligned} \quad (7)$$

For nonlinear systems, local observability should be concerned which is often tedious to handle for large-scale complex systems. Let an initial state \hat{x}_0^i , then for a given bounded input trajectory $u(t)$, assume there exists a solution trajectory $\hat{y}^i(t) = \hat{h}(\hat{x}^i(t))$ for $t \in [0, T]$ and $T < \infty$, satisfying eq.(7). If the points lie within the neighborhood of $\hat{x}^i(0)$, satisfying $\hat{h}(\hat{x}^i(t)) \neq \hat{h}(\hat{x}^j(t))$ such that $\hat{x}_0^i \neq \hat{x}_0^j$, then \hat{x}_0^i and \hat{x}_0^j are said to be (locally) distinguishable. Moreover, the local distinguishability property has a one-to-one correspondence with the local observability: local observability \Leftrightarrow local distinguishable. Loosely speaking, different state trajectories starting from distinct initial states \hat{x}_0^i and \hat{x}_0^j , will always generate distinct output trajectories. If not local distinguishable, then it is not possible to uniquely (locally) deduce $x^i(0)$ from $y^i(t)-u^i(t)$ information. A criterion for local observability/distinguishability or algebraic observability is given using Lie derivatives and brackets [Isidori \(1995\)](#),

⁴There are many possibilities of approximating stochastic integral $\Gamma \int_{\beta(t_k)}^{\beta(t_{k+1})} d\beta$. See in [Kloeden et al. \(2012\)](#).

⁵ $\{a_k\} \equiv \{a_1, a_2, \dots, a_n\}$ for $k = 1, 2, 3, \dots, n$. $a_k \equiv a(t_k)$. $a(\cdot)$ is a function of time.

⁶[Margaria et al. \(2004\)](#) consider affine systems. For more general treatment, refer [Hermann and Krener \(1977\)](#), [Isidori \(1995\)](#), [Slotine et al. \(1991\)](#).

however algebraic observability analysis is not easy to use for large-scale complex systems because the test related to searching for the rank of a matrix with higher dimensions and algebraic variables. The best solution is to use structural observability instead, preferably its graph-theoretic associate, where structural dependencies among output and state variables are used to define a necessary condition for observability. Observability is a structural property and by exploiting the system structure, it is possible to extract much more information than the rank test for the algebraic observability check. The structural dependencies of the system given in eq.(7) are mapped into a directed graph, so called the system digraph. The system digraph G is created as follows [Reinschke \(1988\)](#):

1. define nodes (or vertices) $x_1, x_2, \dots, x_{n_x}, y_1, y_2, \dots, y_{n_y}$,
2. there is a directed edge from x_i to x_j ($x_i \rightarrow x_j$) if $\frac{\partial f_i}{\partial x_j} \neq 0$, and
3. there is a directed edge from y_j to x_i ($y_j \rightarrow x_i$) if $\frac{\partial h_i}{\partial x_j} \neq 0$.

The system digraph may be decomposed into strongly connected components (SCCs): a SCC is a largest subgraph where there exists a directed path from each node to every other node in it. A root SCC is a subgraph such that there are no incoming edges to its nodes. In order to achieve structural observability, at least one node of each root SCC should be measured. Hence, number of sensor nodes must not be less than number root SCCs [Liu et al. \(2013\)](#). On the other hand, suppose that sensor nodes are given. Then, the system is structurally observable if the system digraph is spanned by cacti covering all nodes [Lin \(1974\)](#). Figure 1 shows a cactus. A cactus consists of a stem and possibly one or more buds attach to the stem. Stem is a directed path where starting node is the root and end is the top. A root node is always a measurement. A bud is an elementary closed path, which connects to the stem via the distinguished edge. A bud should not be connected either to the root or to the top node and no two distinguished edges share the same node.

3 Process model

See Figure 2 for the process flow sheet. The process consists of four sections: (i) the slurrification section where powdered raw material containing mostly copper oxide (CuO) is slurrified using recycled anolyte flow, which containing sulfuric acid (H_2SO_4), taken from the electrowinning section, (ii) the leaching section where

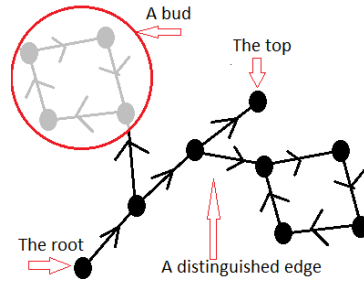


Figure 1: A cactus.

sulfuric acid is added to the slurry in order to leach more copper (Cu) into the solution, (iii) the purification section where the slurry is first filtered to extract the solution, which contains copper sulphate ($CuSO_4$), followed by the cementation and fine filtering processes, and (iv) the electrowinning section where the solution containing (Cu^{2+}) is electrolyzed to release solid copper at the cathode.

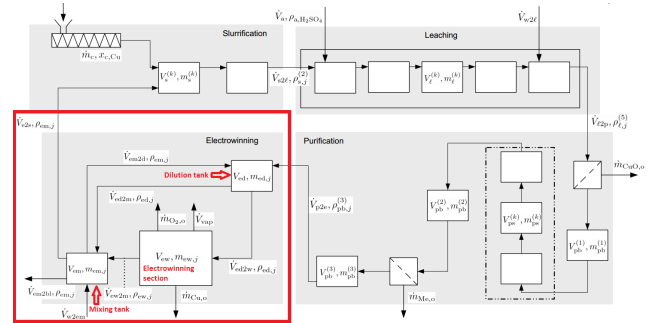


Figure 2: Process flow sheet for the Copper electrowinning process (electrowinning section is highlighted in red).

In the initial model [Lie and Hauge \(2008\)](#),⁷ tank-level dynamics are neglected (i.e. static mass balances). However, level dynamics of several tanks are included in the modified model. It is assumed that the tanks in the slurrification and leaching sections as well as the electrowinning tank have no level variations. Level dynamics in the following tanks are included: buffer tank 1, buffer tank 2, buffer tank 3, dilution tank and mixing tank. System digraph is given in Figure 3 and it is seen that it is not possible to isolate a spanning cacti covering all nodes, then the system is not structurally observable and thereby, not observable [Perera et al. \(2015\)](#). Since, it is already proved in that the system is not structurally observable, only a (structurally) observable subsystem — which is the

⁷In order to save space, the complete model is not given in this paper. Refer [Lie and Hauge \(2008\)](#) for the detailed model.

electrowinning section — is considered in the following discussion.

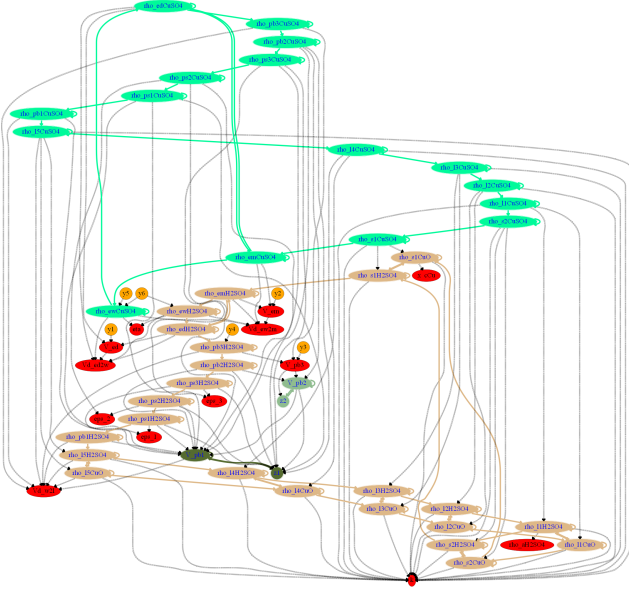


Figure 3: System digraph.

Electrowinning section consists of three subsystems: dilution tank, electrowinning tank, and mixing tank. Model equations are given below, where $\dot{m}_{Cu,o} = \frac{M_{Cu}}{z_{Cu} \cdot C} \cdot \bar{\eta} \cdot \bar{I} \cdot 3600$.

$$\frac{dV_{ed}}{dt} = \dot{V}_{p2e} + \dot{V}_{em2d} - \dot{V}_{ed2m} - \dot{V}_{ed2w} \quad (8)$$

$$\frac{d\rho_{ed,CuSO_4}}{dt} = \frac{\dot{V}_{p2e} \cdot (\rho_{pb,CuSO_4}^{(3)} - \rho_{ed,CuSO_4})}{V_{ed}} + \frac{\dot{V}_{em2d} \cdot (\rho_{em,CuSO_4} - \rho_{ed,CuSO_4})}{V_{ed}} \quad (9)$$

$$\frac{d\rho_{ed,H_2SO_4}}{dt} = \frac{\dot{V}_{p2e} \cdot (\rho_{pb,H_2SO_4}^{(3)} - \rho_{ed,H_2SO_4})}{V_{ed}} + \frac{\dot{V}_{em2d} \cdot (\rho_{em,H_2SO_4} - \rho_{ed,H_2SO_4})}{V_{ed}} \quad (10)$$

$$\frac{d\rho_{ew,CuSO_4}}{dt} = \frac{\dot{V}_{ed2w} \cdot (\rho_{ed,CuSO_4} - \rho_{ew,CuSO_4})}{V_{ew}} + \frac{\dot{V}_{vap} \cdot \rho_{ew,CuSO_4} - \frac{M_{CuSO_4}}{M_{Cu}} \cdot \dot{m}_{Cu,o}}{V_{ew}} \quad (11)$$

$$\frac{d\rho_{ew,H_2SO_4}}{dt} = \frac{\dot{V}_{ed2w} \cdot (\rho_{ed,H_2SO_4} - \rho_{ew,H_2SO_4})}{V_{ew}} + \frac{\dot{V}_{vap} \cdot \rho_{ew,H_2SO_4} + \frac{M_{H_2SO_4}}{M_{Cu}} \cdot \dot{m}_{Cu,o}}{V_{ew}} \quad (12)$$

$$\dot{V}_{ed2w} = \dot{V}_{ew2m} + \dot{V}_{vap} \quad (13)$$

$$\frac{dV_{em}}{dt} = \dot{V}_{ed2m} + \dot{V}_{ew2m} + \dot{V}_{w2em} - (\dot{V}_{e2s} + \dot{V}_{em2d} + \dot{V}_{em2bl}) \quad (14)$$

$$\frac{d\rho_{em,CuSO_4}}{dt} = \frac{\dot{V}_{ed2m} \cdot (\rho_{ed,CuSO_4} - \rho_{em,CuSO_4})}{V_{em}} + \frac{\dot{V}_{ew2m} \cdot (\rho_{ew,CuSO_4} - \rho_{em,CuSO_4})}{V_{em}} - \frac{\dot{V}_{w2em} \cdot \rho_{em,CuSO_4}}{V_{em}} \quad (15)$$

$$\frac{d\rho_{em,H_2SO_4}}{dt} = \frac{\dot{V}_{ed2m} \cdot (\rho_{ed,H_2SO_4} - \rho_{em,H_2SO_4})}{V_{em}} + \frac{\dot{V}_{ew2m} \cdot (\rho_{ew,H_2SO_4} - \rho_{em,H_2SO_4})}{V_{em}} - \frac{\dot{V}_{w2em} \cdot \rho_{em,H_2SO_4}}{V_{em}} \quad (16)$$

$$\frac{d}{dt} \bar{\eta}(t) = 0 \quad (17)$$

$$\frac{d}{dt} \dot{V}_{ed2w} = \epsilon \dot{V}_{ed2w} \quad (18)$$

$$\frac{d}{dt} \rho_{pb,CuSO_4}^{(3)} = -\beta \cdot \rho_{pb,CuSO_4}^{(3)} + \epsilon \rho_{pb,CuSO_4}^{(3)} \quad (19)$$

$$\frac{d}{dt} \dot{V}_{ew2m} = \epsilon \dot{V}_{ew2m} \quad (20)$$

Where β is a positive constant which should be specified. The last 4 equations are due to parameter-disturbance augmentation.⁸ There are 4 measurements:

- $y_1 = V_{ed}$,
- $y_2 = V_{em}$,
- $y_3 = \frac{M_{Cu}}{M_{CuSO_4}} \cdot \rho_{ew,CuSO_4}$, and
- $y_4 = \rho_{ew,H_2SO_4} + \frac{M_{H_2SO_4}}{M_{CuSO_4}} \cdot \rho_{ew,CuSO_4}$.

⁸All the variables in the system model are defined according to Lie and Hauge (2008).

4 Observability analysis

Including augmented parameter $\bar{\eta}$ and disturbances \dot{V}_{ed2w} , \dot{V}_{ew2m} , and $\rho_{pb,H_2SO_4}^{(3)}$, altogether there are 12 ($n_x = 12$) states. This makes it hard to analyze for algebraic observability. See Figure 4 for the system digraph. According to the digraph, it is always possible to estimate \dot{V}_{ed2w} and \dot{V}_{ew2m} using y_1 and y_2 . The reason is that $y_1 \rightarrow V_{ed} \rightarrow \dot{V}_{ed2w}$ and $y_2 \rightarrow V_{em} \rightarrow \dot{V}_{ew2m}$ are two cacti (just two stems without buds). There exists a spanning cacti covering all nodes, hence the system is structurally observable. The spanning cacti is as follows:

- $y_1 \rightarrow V_{ed} \rightarrow \dot{V}_{ed2w}$,
- $y_2 \rightarrow V_{em} \rightarrow \dot{V}_{ew2m}$,
- $y_3 \rightarrow \rho_{ew,CuSO_4} \rightarrow \rho_{ed,CuSO_4} \rightarrow \rho_{em,CuSO_4}$ with the bud $\rho_{pb,CuSO_4}^{(3)} \rightarrow \rho_{pb,CuSO_4}^{(3)}$, and
- $y_4 \rightarrow \rho_{ew,H_2SO_4} \rightarrow \eta$ with the bud $\rho_{em,H_2SO_4} \leftrightarrow \rho_{ed,H_2SO_4}$.

Consider eq.(19). If $\beta = 0$, then the self-loop $\rho_{pb,CuSO_4}^{(3)} \rightarrow \rho_{pb,CuSO_4}^{(3)}$ in the digraph, in Figure 4, will disappear. Consequently, it is failed to have a spanning cacti. Therefore, $d\rho_{pb,CuSO_4}^{(3)}/dt = 0$ is not a useful augmentation. Similarly, it can also be shown that $d^2\rho_{pb,CuSO_4}^{(3)}/dt^2 = 0$ fails to keep structural observability. See the formulation below:

$$\frac{d}{dt}\rho_{pb,CuSO_4}^{(3)} = \overline{\rho_{pb,CuSO_4}^{(3)}} \quad (21)$$

$$\frac{d}{dt}\overline{\rho_{pb,CuSO_4}^{(3)}} = \overline{\epsilon_{\rho_{pb,CuSO_4}^{(3)}}} \quad (22)$$

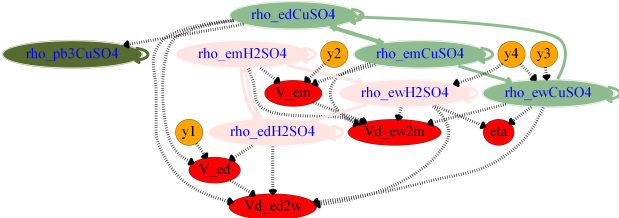


Figure 4: System digraph for the electrowinning section.

5 Filtering theory for discrete systems

The objective of filtering/estimation is to estimate system state from available noisy data [Jazwinski \(2007\)](#). Consider the discrete stochastic dynamical system given in eq.(6). Assume that system is observable.⁹ $\{w_k, k = 1, 2, 3, \dots\}$ is a random sequence with given probability distributions. Probability density function of \hat{x}_0 is also given. The famous assumptions are $\{w_k\}$ is white Gaussian sequence, $w_k \sim N(0, Q_k)$, and independent of \hat{x}_0 . Also, $\{v_k, k = 1, 2, 3, \dots\}$ is a random sequence with given probability distributions such as $v_k \sim N(0, R_k)$. A solution to the eq.(6) is the probability density function of x_k . $\{wk\}$ and $\{vk\}$ may be dependent. We may write:

$$E\left\{\begin{bmatrix} w_k \\ v_k \end{bmatrix} \begin{bmatrix} w_l^T & v_l^T \end{bmatrix}\right\} = \begin{bmatrix} Q_k & S_k \\ S_k^T & R_k \end{bmatrix} \delta_{kl} \quad (23)$$

Where $\delta_{kl} = 1$ when $k = l$, else $\delta_{kl} = 0$. $S_k = 0$ if $\{w_k\}$ and $\{v_k\}$ are independent. The two sequences $Y_N = \{y_1, y_2, \dots, y_N\}$ and $U_N = \{u_0, u_1, \dots, u_{N-1}\}$ contain input-output data and $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{N-1}\}$ to be estimated for given Y_N and U_N . There are several ways to handle nonlinear filtering problems: extended Kalman filtering (EKF) [Strang and Borre \(1997\)](#), unscented Kalman filtering (UKF) [Simon \(2006\)](#), particle filtering (PF) [Doucet et al. \(2000\)](#), etc. Figure 5 gives a comparison about different state estimation techniques with respect to accuracy and computational effort.

Often, EKF could be the starting point for a nonlinear estimation problem, where linear Kalman filtering theory is adapted based on first-order linearization. The extended Kalman filter¹⁰ is as follows for $k = 1, \dots, n$:

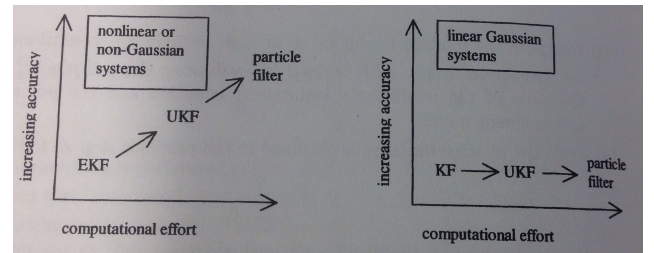


Figure 5: State estimation trade-offs (a scanned page from [Simon \(2006\)](#)).

⁹If the continuous system is observable, the discrete system is also observable [Moraal and Grizzle \(1995\)](#).

¹⁰For convergence characteristics of EKF, refer [Ljung \(1979\)](#) [Cox \(1964\)](#) [Fitzgerald \(1971\)](#).

- $Q_{k-1}, R_{k-1}, S_{k-1} = 0, P_{k-1}^+$ and \hat{x}_{k-1}^+ are given;
- $A_{k-1} = \left. \frac{\partial \hat{f}(\hat{x}, u)}{\partial \hat{x}} \right|_{\hat{x}_{k-1}^+}$, A_{k-1} is invertible and pair (A_{k-1}, C_{k-1}) is observable [Song and Grizzle \(1992\)](#);
- $P_k^- = A_{k-1} P_{k-1}^+ A_{k-1}^T + \Gamma Q_{k-1} \Gamma^T$;
- $\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1})$;
- $C_k = \left. \frac{\partial h(\hat{x})}{\partial \hat{x}} \right|_{\hat{x}_k^-}$;
- $K_k = P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1}$;
- $\hat{x}_k^+ = \hat{x}_k^- + K_k [y_k - C_k \hat{x}_k^-]$, and
- $P_k^+ = (I - K_k C_k) P_k^-$.

Where \hat{x}_k^+ is the best estimate for x_k . The main reason for the divergence of EKF is the model fidelity. This can be demonstrated easily using even a simple scalar system [Fitzgerald \(1971\)](#) [Simon \(2006\)](#). In eq.(5), $\Gamma \in \mathbb{R}^{n_x \times n_w}$ and $n_w \leq n_x$. If $n_w < n_x$, then there is at least one differential equation where a process noise term does not appear. However, by including a fictitious process noise to such equations, it may be possible to compensate model inaccuracies to some extent. For a constant parameter, ideally, we have $p_{k+1} = p_k$ and however, a small fictitious noise is added to get $p_{k+1} = p_k + \epsilon_p$. Now, we have a stochastic system given in eq.(24), where $\Gamma_x, \Gamma_w, \Gamma_p \neq 0$ and $[\Gamma_x, \Gamma_w, \Gamma_p]^T \in \mathbb{R}^{n_x + n_w + n_p \times n_x + n_w + n_p}$ and $w_k \in \mathbb{R}^{n_x + n_w + n_p}$. Also, in order to increase the stability-convergence characteristics, fading-memory filters may be used [Simon \(2006\)](#). Actually, it is not necessary to include fictitious noises to all differential equations, but it is enough to include for some of them. Fictitious noises are inserted such that the stochastic system becomes state stabilizable with respect to process noise vector [Potter \(1965\)](#).

$$\begin{bmatrix} x_{k+1} \\ w_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} f(x_k, p_k, u_k) \\ g_k(w_k) \\ p_k \end{bmatrix} + \begin{bmatrix} \Gamma_x \\ \Gamma_w \\ \Gamma_p \end{bmatrix} w_{k+1}; \quad k = 0, 1, \dots, n \quad (24)$$

EKF is susceptible to linearization errors and if the model is highly nonlinear then the EKF may not give acceptable results. The unscented Kalman filter [Julier et al. \(1995\)](#) [Doucet et al. \(2000\)](#) is a possible candidate to try with, if the EKF fails. Importantly, UKF does not require to calculate Jacobian matrices. The

following steps are followed in UKF algorithm:

- $Q_{k-1}, R_{k-1}, P_{k-1}^+$ and x_{k-1}^+ are given;
- Find sigma points $\hat{x}_{k-1}^{(i)}$ such that $\hat{x}_{k-1}^{(i)} = x_{k-1}^+ + x^{(i)}$ and $x^{(i)} = (-1)^i \cdot \left(\sqrt{n_x \cdot P_{k-1}^+} \right)_i^T$ for $i = 1, 2, \dots, 2 \cdot n_x$. $\left(\sqrt{n_x \cdot P_{k-1}^+} \right)_i$ is the $\langle i \rangle^{\text{th}}$ row of $\sqrt{n_x \cdot P_{k-1}^+}$;
- $x_k^- = \frac{1}{2 \cdot n_x} \sum_{i=1}^{2 \cdot n_x} \hat{x}_k^{(i)}$, where $\hat{x}_k^{(i)} = f(\hat{x}_{k-1}^{(i)}, u_{k-1})$;
- $P_k^- = \frac{1}{2 \cdot n_x} \sum_{i=1}^{2 \cdot n_x} \left[\left(\hat{x}_k^{(i)} - x_k^- \right) \left(\hat{x}_k^{(i)} - x_k^- \right)^T \right] + \Gamma Q_{k-1} \Gamma^T$;
- Find new set of sigma points $\hat{x}_k^{(i)}$ such that $\hat{x}_k^{(i)} = x_k^- + x^{(i)}$ and $x^{(i)} = (-1)^i \cdot \left(\sqrt{n_x \cdot P_k^-} \right)_i^T$ for $i = 1, 2, \dots, 2 \cdot n_x$. $\left(\sqrt{n_x \cdot P_k^-} \right)_i$ is the $\langle i \rangle^{\text{th}}$ row of $\sqrt{n_x \cdot P_k^-}$;
- $y_k^- = \frac{1}{2 \cdot n_x} \sum_{i=1}^{2 \cdot n_x} \hat{y}_k^{(i)}$, where $\hat{y}_k^{(i)} = C \cdot \hat{x}_k^{(i)}$;
- $P_y = \frac{1}{2 \cdot n_x} \sum_{i=1}^{2 \cdot n_x} \left[\left(\hat{y}_k^{(i)} - y_k^- \right) \left(\hat{y}_k^{(i)} - y_k^- \right)^T \right] + R_k$;
- $P_{xy} = \frac{1}{2 \cdot n_x} \sum_{i=1}^{2 \cdot n_x} \left[\left(\hat{x}_k^{(i)} - x_k^- \right) \left(\hat{y}_k^{(i)} - y_k^- \right)^T \right] + R_k$;
- $K_k = P_{xy} P_y^{-1}$;
- $x_k^+ = x_k^- + K_k [y_k - y_k^-]$, and
- $P_k^+ = P_k^- - K_k P_y K_k^T$.

6 Structure of the Python code and results

[Perera et al. \(2015\)](#) discuss a procedure for automating structural observability analysis in Python using JModelica.org-Casadi interface. Figure 3 and 4 are generated based on above mentioned article. There are several Python packages for state estimation: pyda, filterpy, pykalman, KF, etc.¹¹ First simulated noisy data is created. See the Python script given below. `model(x,t,u,w,p)` represents from eq.(8) to eq.(16) and `f(x,dt,u,w,p)` is the discrete system. Simulated data is obtained with known variations — see the for

¹¹Pyda, filterpy, pykalman, and KF packages are available at the Python package index. See in <https://pypi.python.org/pypi>.

loop below — in disturbances and inputs.

```
# Import packages
import numpy as np
from numpy import random
from numpy.random import randn
from scipy.integrate import odeint
import matplotlib.pyplot as plt
import scipy.io as sio
< define necessary parameters >
def model(x,t,u,w,p):
    """
    Continuous dynamic model:
    dx/dt = model(x,t,u,w,p)
    """
    < enter code here >
    return np.array(dxdt)
def f(x,dt,u,w,p):
    """
    State transition function:
    x(t+dt) = f(x(t),dt,u(t))
    """
    res = odeint(lambda X,T:\
        model(X,T,u,w,p),x,np.linspace(0.0,dt,2))
    return res[-1,: ]
def h(x,v):
    """
    Observation function:
    y(t) = h(x(t))
    """
    z0 = x[0] + v[0]
    z1 = x[1] + v[1]
    z2 = (M_Cu/M_CuSO4)*x[2] + v[2]
    z3 = x[5] + (M_H2SO4/M_CuSO4)*x[2] + v[3]
    z = [z0,z1,z2,z3]
    return np.array(z)
# Define initial x,u,w and p
w = w0
x = x0
p = p0
u = u0
# Start simulation
for k in np.arange(1,N+1):
    # calculate x and z using discrete model
    x = f(x,dt,u,w,p)
    z = h(x,v)
    # log x and z
    < enter code here >
    # change disturbances
    # change inputs
    < enter code here >
# Save data
sio.savemat('data.mat',{'t':t,< enter code here >)
```

Now state estimators are considered. First, $\text{model}(x,t,u,w,p)$ is updated as $\text{model}(x,t,u,w)$ with augmented states from eq.(17) to eq.(20). In order to implement EKF algorithm in Section 5, it is necessary to calculate Jacobian matrices of f and h : see methods $\text{ABL}(x,u,w,dx = 1.e-5,dw = 1.e-5)$ and $\text{CM}(x,v,dx = 1.e-5,dv = 1.e-5)$ below. Finally, two more methods are created: $\text{predict}(x,u,P,Q)$ and $\text{update}(x,z,P,R)$ and then EKF is simulated in a for loop. Fading-memory EKF can be easily implemented in a similar way.

```
def model(x,t,u,w):
    """
    Continuous augmented dynamic model:
    dx/dt = model(x,t,u,w,p)
    """
    < enter code here >
    deta_ew_dt = 0. + w_eta_ew
```

```
dVd_ed2w_dt = 0. + w_Vd_ed2w
drho_pb3CuS04_dt = -beta*rho_pb3CuS04 + w_rho_pb3CuS04
dVd_ew2m_dt = 0. + w_Vd_ew2m
return np.array(dxdt)
def ABL(x,u,w,dx = 1.e-5,du = 1.e-5,dw = 1.e-5):
    """
    Calculate A = df/dx, B = df/du, and L = df/dw,
    using finite (central) difference method.
    """
    < enter code here >
    return A,B,L
def CM(x,v,dx = 1.e-5,dv = 1.e-5):
    """
    Calculate C = dh/dx and M = dh/dv,
    using finite (central) difference method.
    """
    < enter code here >
    return C, M
def predict(x,u,P,Q):
    < enter code here>
    return x,P
def update(x,z,P,R):
    < enter code here>
    return x,P
# Simulate
for k in np.arange(1,N+1):
    # predict state
    x,P = predict(x,u,P,Q)
    # update state
    z = np.array([z0[k],z1[k],z2[k],z3[k]])
    x,P = update(x,z,P,R)
# log data
```

For UKF estimation, `filterpy` Python is used. Use following script to create UKP object:

```
from filterpy.kalman import UnscentedKalmanFilter as UKF
ukf = UKF(<enter code here>)
# Initialize UKF
ukf.x = x0
ukf.R = np.diag([0.01, 0.01,0.01, 0.01])
ukf.Q = 5*np.eye(dim_x)
ukf.P = 1.*np.eye(dim_x)
# Simulate
for k in np.arange(1,N+1):
    # predict state
    ukf.predict()
    # measurement update
    z = np.array([z0[k],z1[k],z2[k],z3[k]])
    ukf.update(z)
    x = ukf.x
    # data logging
```

7 Conclusions and future work

We have demonstrated a way of automating parameter-disturbance-state estimation process for large-scale complex dynamic systems completely using free software tools such as Modelica, Python, Casadi, etc. A real world case study is considered and managed to estimate 1 parameter, 3 disturbances and 8 states with 4 measurements using extended Kalman filter and unscented Kalman filter algorithms. Extended Kalman filter, fading-memory Kalman filter and unscented Kalman filter have shown more or less similar results. All three estimators converge. Note that in this paper, only Gaussian process noises are concerned,

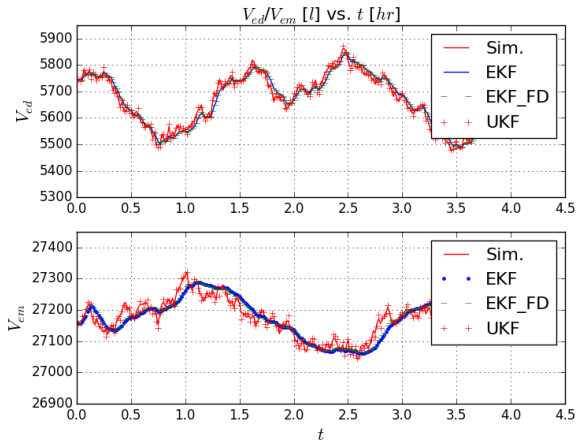


Figure 6: Estimation of V_{ed} and V_{em} .

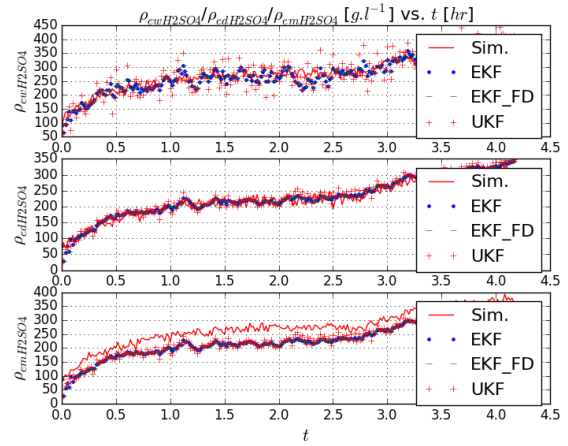


Figure 9: Estimation of ρ_{ew,H_2SO_4} , ρ_{ed,H_2SO_4} and ρ_{em,H_2SO_4} .

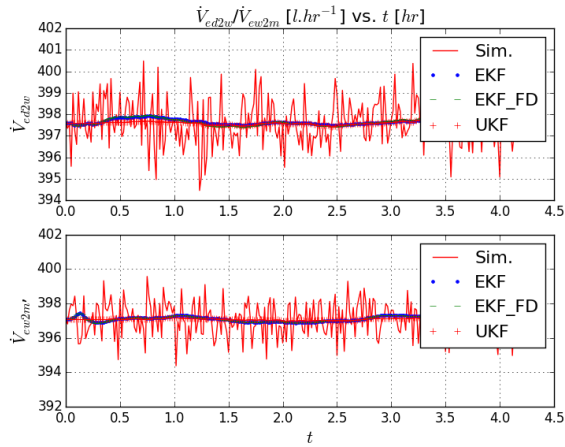


Figure 7: Estimation of \dot{V}_{ed2w} and \dot{V}_{ew2m} .

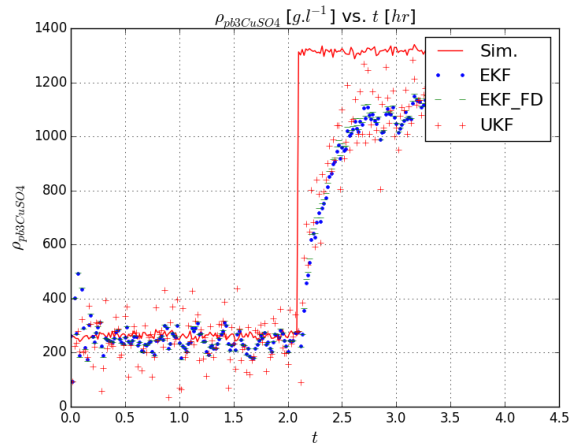


Figure 10: Estimation of $\rho_{pb3,CuSO_4}$.

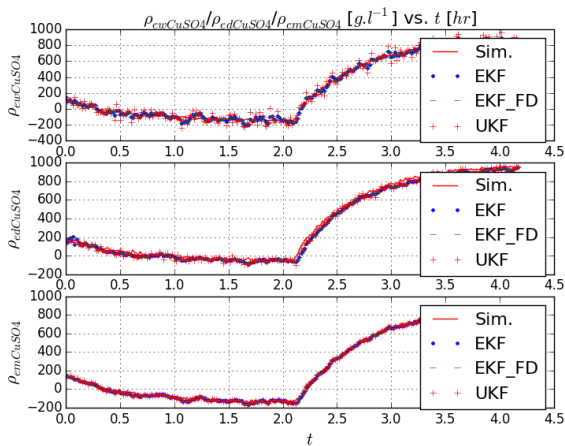


Figure 8: Estimation of $\rho_{ew,CuSO_4}$, $\rho_{ed,CuSO_4}$ and $\rho_{em,CuSO_4}$.

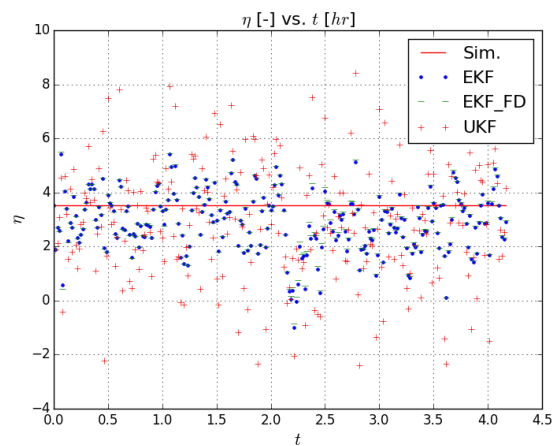
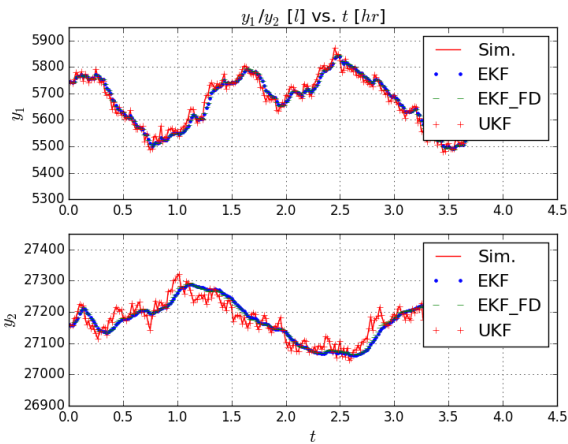
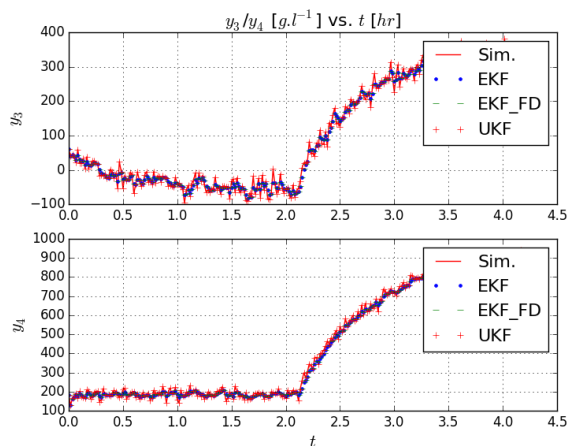


Figure 11: Estimation of η_{ew} .

Figure 12: Estimation of z_0 and z_1 .Figure 13: Estimation of z_2 and z_3 .

often which is not the case in reality. Therefore, in order to test the capabilities of various estimators, those estimators should be applied with real process data where process noises may not be Gaussian and therefore, parameter-disturbance-state estimation with real process data is set as a future work. In particular, it is expected to implement particle filter algorithms with real process data, since particle filters can handle non-Gaussian process noise as well as it works better compared to EKF/UKF algorithms when the system model is highly nonlinear.

Filtering parameters, such as Q , R , P , etc., are tuning parameters, but filter tuning is not considered in this paper. Those parameters directly link with filter-response-time and filter sensitivity (propagation of the error covariance matrix P), therefore sensitivity analysis and filter tuning should be done as a future work. As the full state of the electrowinning system can be reconstructed from measurement data, it opens up many possibilities, among others, implementing an optimal control strategy. Also, estimating disturbances alongside the states will help to improve control performance — disturbance compensation. Finally, comparing the performances among PID and Optimal control strategies would be an interesting future work.

References

- Bona, B. and Smay, R. J. Optimum reset of ship's inertial navigation system. *Aerospace and Electronic Systems, IEEE Transactions on*, 1966. (4):409–414. doi:[10.1109/TAES.1966.4501790](https://doi.org/10.1109/TAES.1966.4501790).
- Cox, H. On the estimation of state variables and parameters for noisy dynamic systems. *Automatic Control, IEEE Transactions on*, 1964. 9(1):5–12. doi:[10.1109/TAC.1964.1105635](https://doi.org/10.1109/TAC.1964.1105635).
- Doucet, A., Godsill, S., and Andrieu, C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 2000. 10(3):197–208. doi:[10.1023/A:1008935410038](https://doi.org/10.1023/A:1008935410038).
- Fitzgerald, R. J. Divergence of the Kalman filter. *Automatic Control, IEEE Transactions on*, 1971. 16(6):736–747. doi:[10.1109/TAC.1971.1099836](https://doi.org/10.1109/TAC.1971.1099836).
- Gelb, A. *Applied optimal estimation*. The M.I.T. press, 2001.
- Hermann, R. and Krener, A. J. Nonlinear controllability and observability. *IEEE Transactions on automatic control*, 1977. 22(5):728–740. doi:[10.1109/TAC.1977.1101601](https://doi.org/10.1109/TAC.1977.1101601).

- Isidori, A. *Nonlinear control systems*. Springer Science & Business Media, 1995.
- Jazwinski, A. H. *Stochastic processes and filtering theory*. Courier Corporation, 2007.
- Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. A new approach for filtering nonlinear systems. In *American Control Conference, Proceedings of the 1995*, volume 3. IEEE, pages 1628–1632, 1995. doi:[10.1109/ACC.1995.529783](https://doi.org/10.1109/ACC.1995.529783).
- Kloeden, P. E., Platen, E., and Schurz, H. *Numerical solution of SDE through computer experiments*. Springer Science & Business Media, 2012.
- Lie, B. and Hauge, T. A. Modeling of an industrial copper leaching and electrowinning process, with validation against experimental data. In *Proceedings SIMS*. pages 7–8, 2008.
- Lin, C. T. Structural controllability. *Automatic Control, IEEE Transactions on*, 1974. 19(3):201–208. doi:[10.1109/TAC.1974.1100557](https://doi.org/10.1109/TAC.1974.1100557).
- Liu, Y.-Y., Slotine, J.-J., and Barabási, A.-L. Observability of complex systems. *Proceedings of the National Academy of Sciences*, 2013. 110(7):2460–2465. doi:[10.1073/pnas.1215508110](https://doi.org/10.1073/pnas.1215508110).
- Ljung, L. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *Automatic Control, IEEE Transactions on*, 1979. 24(1):36–50. doi:[10.1109/TAC.1979.1101943](https://doi.org/10.1109/TAC.1979.1101943).
- Margaria, G., Riccomagno, E., and White, L. J. Structural identifiability analysis of some highly structured families of statespace models using differential algebra. *Journal of mathematical biology*, 2004. 49(5):433–454. doi:[10.1007/s00285-003-0261-3](https://doi.org/10.1007/s00285-003-0261-3).
- Moraal, P. and Grizzle, J. Observer design for nonlinear systems with discrete-time measurements. *Automatic Control, IEEE Transactions on*, 1995. 40(3):395–404. doi:[10.1109/9.376051](https://doi.org/10.1109/9.376051).
- Perera, M. A. S., Lie, B., and Pfeiffer, C. F. Structural Observability Analysis of Large Scale Systems Using Modelica and Python. *Modeling, Identification and Control*, 2015. 36(1):53–65. doi:[10.4173/mic.2015.1.4](https://doi.org/10.4173/mic.2015.1.4).
- Potter, J. E. *A matrix equation arising in statistical filter theory*, volume 270. National Aeronautics and Space Administration, 1965.
- Reinschke, K. J. *Multivariable control: a graph theoretic approach*. 1988.
- Simon, D. *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.
- Slotine, J.-J. E., Li, W., et al. *Applied nonlinear control*, volume 199. Prentice-hall Englewood Cliffs, NJ, 1991.
- Song, Y. and Grizzle, J. W. The extended Kalman filter as a local asymptotic observer for nonlinear discrete-time systems. In *American Control Conference, 1992*. IEEE, pages 3365–3369, 1992.
- Strang, G. and Borre, K. *Linear algebra, geodesy, and GPS*. Siam, 1997.