MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Constrained Smoothers for State Estimation of Vapor Compression Cycles

Deshpande, Vedang; Laughman, Christopher R.; Ma, Yingbo; Rackauckas, Chris

TR2022-063 June 11, 2022

Abstract

State estimators can be a powerful tool in the development of advanced controls and performance monitoring capabilities for vapor compression cycles, but the nonlinear and numerically stiff aspects of these systems pose challenges for the practical implementation of estimators on large physics-based models. We develop smoothing methods in the extended and ensemble Kalman estimation frameworks that satisfy physical constraints and address practical limitations with standard implementations of these estimators. These methods are tested on a model built in the Julia language, and are demonstrated to successfully estimate unmeasured variables with high accuracy.

American Control Conference (ACC) 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Constrained Smoothers for State Estimation of Vapor Compression Cycles

Vedang M. Deshpande[1], Christopher R. Laughman[2,‡], Yingbo Ma[3], Chris Rackauckas[3]

***Abstract*— State estimators can be a powerful tool in the development of advanced controls and performance monitoring capabilities for vapor compression cycles, but the nonlinear and numerically stiff aspects of these systems pose challenges for the practical implementation of estimators on large physics-based models. We develop smoothing methods in the extended and ensemble Kalman estimation frameworks that satisfy physical constraints and address practical limitations with standard implementations of these estimators. These methods are tested on a model built in the Julia language, and are demonstrated to successfully estimate unmeasured variables with high accuracy.**

***Index Terms*— Vapor compression cycle, state estimation, constrained smoothing, ensemble Kalman, extended Kalman**

I. INTRODUCTION

Vapor compression cycles are positioned to play an increasingly important role in our society, as they represent an effective means to decarbonize energy networks by supplanting fossil fuel-based heating systems and provide a varying demand-side load for renewable energy resources, and are a key technology for ameliorating the consequences of climate change on human health. Unfortunately, many common working fluids used in this equipment also raise significant environmental concerns, as their global warming potential is generally hundreds to thousands of times worse than CO_2 [1]. Because of their prevalence and connection to a broad range of applications, technologies to improve the operation and performance of vapor compression cycles have the potential to make a significant impact in the future.

State estimation methods can be an important tool in the development of these thermofluid systems because their spatially distributed construction, limited sensor availability, and long equipment lifetimes can make it difficult to observe a variety of salient information relevant to either control or fault diagnostics for the system under study. For example, the high cost of pressure or mass flow rate sensors often precludes their use in commercially available equipment, and control architectures that depend upon observations of these or other unmeasurable quantities to attain performance improvements are not realizable without such information. Such measurements could be used to assess the amount of subcooled refrigerant in a condensing heat exchanger for a controller which optimizes the efficiency of the heat transfer process. Moreover, the retrospective analysis of operational data can enable fault diagnostics for the early identification of performance degradation before the accumulated effects of efficiency loss result in significant energy waste.

Physics-based behavioral models of these systems pose distinct challenges when used in common approaches for state estimation. Heat and fluid flow in the heat exchangers (HEXs) used in these systems is governed by nonlinear partial differential equations (PDEs) that relates the conjugate heat transfer to the refrigerant dynamics, which is nonlinear and has discontinuous derivatives at the liquid and vapor saturation curves. These PDEs can be discretized into large sets of index-1 nonlinear differential algebraic equations (DAEs) which are numerically stiff, with time constants that range from milliseconds to hours, and require solvers that are able to correctly manage the implicit discontinuities related to the phase-change behavior of common refrigerants. System representations that use finite volume discretizations of the HEX behavior typically consist of hundreds of states.

A particular focus of previous work into estimation methods for these systems has been on the development of advanced control methods, such as model predictive control (MPC). As an example, Bortoff et al [2] used a Kalman filter for a reduced order model of a multi-terminal vapor-compression cycle to design an MPC algorithm with good robustness properties that meets a specified set of control objectives and satisfies a range of operational constraints. Along similar lines, Krupa et al [3] also employed an Kalman filter in the design of a linear MPC algorithm for a multi-terminal vapor-compression cycle, and demonstrated its efficacy on both a linear and nonlinear model of the plant.

As the dynamics of the overall cycle are dominated by the heat exchangers, these components have also been a topic of particular emphasis for state estimation research. Cheng, He, and Asada [4] developed a nonlinear observer for a reduced-order model of a HEX that accurately predicts both the evaporating temperature and two-phase length during and after a startup transient for an experimental system. More recently, Ghousein and Witrant [5] developed and proved the convergence of a boundary observer for a concentric CO_2 HEX using a Lyapunov analysis, and show initial results using a 1-D multiphase flow model.

While this prior work has demonstrated the efficacy of these approaches on relatively low-order component and system models, there are a wide variety of potential applications of this state estimation techniques for larger and more complex cycles that are in use today. Methods that demonstrate good performance on larger variable refrigerant flow (VRF)

[1]Texas A&M University; This work was completed during a research internship at Mitsubishi Electric Research Laboratories. ✉ vedang.deshpande@tamu.edu

[2,‡]Corresponding author. Mitsubishi Electric Research Laboratories ✉ laughman@merl.com

[3]Julia Computing ✉ {yingbo.ma, chris.rackauckas}@juliacomputing.com

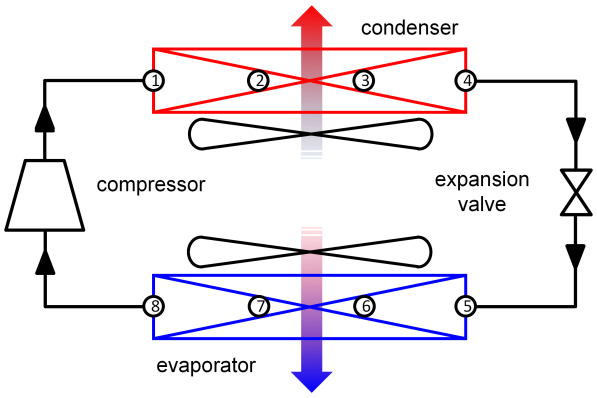

Fig. 1. Structure of a vapor compression cycle.

systems with 3 or more multiphase heat exchangers and provide more granular information about the spatial variation in the heat transfer and fluid flow throughout the system could be used to improve performance monitoring tools and provide important feedback in the system design process.

The present work is focused on the development and application of nonlinear state estimation techniques in a smoothing context to obtain information about vapor compression cycle performance that is either theoretically or practically unmeasurable from an existing set of observations. We adapt two common state estimation methods, the extended Kalman smoother (EKS) and ensemble Kalman smoother (EnKS), to this application by enforcing physics-based constraints which might be otherwise violated during the smoothing process, and apply these methods to large-scale, numerically stiff, and nonsmooth models of the cycle.

In Section 2, we provide an overview of the vapor-compression cycle model and briefly discuss its implementation in Julia along with key enabling capabilities of the modeling environment. We then describe the theoretical development of the constrained EKS and the constrained EnKS for this application in Section 3. In Section 4, we provide some results demonstrating the performance of these methods, and then briefly summarize this work and indicate directions for future investigations in Section 5.

II. Model & Implementation

A vapor compression cycle that can vary the heating or cooling capacity delivered to an occupied space, illustrated in Figure 1, consists of a variable-speed compressor, variable-position expansion device, and two refrigerant-to-air HEXs with variable speed fans. This system transports thermal energy from the air passing through the evaporating HEX to the air passing through the condensing HEX via the refrigerant by using the latent heat of condensation and evaporation, where the compressor can operate efficiently at pressures for which the refrigerant evaporating temperature can be set lower than common occupied space temperatures, while the refrigerant condensing temperature can be set higher than common ambient temperatures.

Because the dynamics of the cycle are dominated by those of the heat exchangers, the HEX models were constructed from systems of DAEs, while algebraic models were constructed for the compressor and expansion valve. A series of simplifying assumptions were used to facilitate the construction of the HEX models, including one-dimensional pipe flow, thermodynamic equilibrium in each discrete volume of the refrigerant pipe at each instant in time, negligible gravitational forces, and equality of the liquid and vapor phase velocities in the two-phase region. The HEX model under these assumptions consists of an interconnected set of models that describe the refrigerant flow, the thermal behavior of the tube wall, and the flow of air across the heat exchanger, forming an index-1 system of DAEs. The refrigerant pipe model uses a spline-based representation of the thermodynamic refrigerant properties and enforces the conservation of mass, momentum, and energy for the refrigerant pipe, e.g,

$$\frac{\partial(\rho A)}{\partial t} + \frac{\partial(\rho A v)}{\partial x} = 0 \tag{1}$$

$$\frac{\partial(\rho v A)}{\partial t} + \frac{\partial(\rho v^2 A)}{\partial x} = -A\frac{\partial P}{\partial x} - F_f \tag{2}$$

$$\frac{\partial(\rho u A)}{\partial t} + \frac{\partial(\rho v h A)}{\partial x} = vA\frac{\partial P}{\partial x} + vF_f + \frac{\partial Q}{\partial x}, \tag{3}$$

where ρ is the density, A is the cross-sectional area of the flow, v is the velocity, P is the pressure, F_f is the frictional pressure drop, u is the specific internal energy, h is the specific enthalpy, and Q is the heat flow rate into or out of the fluid. These equations can be adapted to a finite control volume discretization of the refrigerant pipe via the Reynolds transport theorem by using a staggered-grid approach. An ordinary differential equation (ODE) model was constructed for the refrigerant pipe wall with one heat storage element, and an algebraic model of the airflow was used due to the negligible energy storage of the air. Further information on the structure of the HEX models, as well as the compressor and expansion valve models, is available in [6].

This model was implemented in Julia using ModelingToolkit.jl [7], a package built on a symbolic computational algebra framework that enables the construction of large acausal system models from smaller component models via composition and inheritance. ModelingToolkit.jl provides a representation that allows the models to be defined in a declarative context and then transformed to generate imperative Julia code that can be used with other packages, such as differential equation solvers and automatic differentation tools.

A relatively small model of the vapor compression cycle was built for this study to facilitate rapid simulation and development. The heat exchanger models were discretized into 4 volumes each, with steady-state momentum balances and dynamic mass and energy balances in each volume. The resulting set of DAEs consists of 278 equations, which was reduced to 24 nonlinear ODEs after index reduction via the Pantelides algorithm [8] to produce the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{4}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the state vector of the reduced model and

$\boldsymbol{u} \in \mathbb{R}^{n_u}$ is the vector of known control inputs (compressor speed, valve position) and/or known system parameters (ambient temperature, refrigerant properties, etc.) The state vector $\boldsymbol{x}$ after index reduction is an augmented vector of pressures (P_i), specific enthalpies (h_i), and temperatures (θ_i) of all finite volumes indexed by the subscript i. The finite volumes are numbered 1 through 8 in the direction of refrigerant flow as shown in Fig. 1, with the condensing HEX consisting of volumes 1 through 4, where the inlet volume is the first, and the evaporating HEX consisting of volumes 5 through 8, where the outlet volume is the last. Therefore, the state vector $\boldsymbol{x} \in \mathbb{R}^{24}$ is defined as

$$\boldsymbol{x} := \begin{bmatrix} P_1 & h_1 & \theta_1 & P_2 & h_2 & \theta_2 & \cdots & P_8 & h_8 & \theta_8 \end{bmatrix}^T.$$

A temporally discretized model is obtained from (4) by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))dt =: \boldsymbol{f}_k(\boldsymbol{x}_k), \qquad (5)$$

where $\boldsymbol{x}_k := \boldsymbol{x}(t_k)$, and $\boldsymbol{u}(t)$ has been conveniently absorbed in the definition of $\boldsymbol{f}_k(\cdot)$. The integral in (5) is calculated numerically because of the high-dimensional complex nonlinear nature of the model. This system is numerically quite stiff, as the condition number of the Jacobian of the continuous time model when the system is at steady-state is approximately 7.423×10^4. Numerical solvers that can handle stiff problems were thus required, and the solver `QNDF` [9] was employed to propagate this model forward in time.

To account for the modeling errors and disturbances, the model (5) is updated to add a disturbance term, i.e.

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k) + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k), \qquad (6)$$

where $\boldsymbol{w}_k$ denotes a zero-mean white Gaussian process with covariance $\boldsymbol{Q}_k$.

In practice, cost and reliability considerations limit the available sensors in common vapor compression equipment to measurements of pressure and temperature at a small number of locations in the cycle. For the purpose of this work, we assume that the available sensors measure temperature at four locations ($\theta_2, \theta_4, \theta_6, \theta_8$), and pressure at two locations (P_1, P_8). The measurement model is thus given by

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{\eta}_k, \quad \boldsymbol{\eta}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k), \qquad (7)$$

where $\boldsymbol{y}_k := \boldsymbol{y}(t_k)$,

$$\boldsymbol{y} := \begin{bmatrix} P_1 & \theta_2 & \theta_4 & \theta_6 & P_8 & \theta_8 \end{bmatrix}^T, \qquad (8)$$

$\boldsymbol{H}_k \in \mathbb{R}^{6 \times 24}$ is an appropriately defined binary measurement matrix, and $\boldsymbol{\eta}_k$ is a zero-mean white Gaussian process with covariance $\boldsymbol{R}_k$. Note that the measurement model (7) is linear, and the measurement matrix $\boldsymbol{H}_k$ does not change with time step k. However, the subscript k is maintained for possible generalization to a time-varying measurement model.

III. Constrained Smoothing

As many performance monitoring and analysis applications are designed to operate on existing datasets, this fixed-window aspect of these applications enables the use of smoothers from Kalman state estimation frameworks, which assimilate sensor measurements obtained within a time interval to estimate states everywhere within that interval and avoid the requirements and limitations of filters. Extended Kalman estimation methods, e.g. EKF, are widely used to estimate the states of nonlinear models through the use of linearization to obtain the Jacobian for covariance propagation, and are well-suited for systems of small to moderate sizes. Because the computation of the Jacobian and storage of the covariance matrices can become prohibitive for large-scale systems, ensemble estimation methods (e.g. EnKF) were developed to perform Monte-Carlo simulations of the nonlinear model directly, therefore bypassing the need for linearization or storage of the explicit covariance matrix.

We present smoothing methods formulated in both extended and ensemble Kalman frameworks that incorporate the linear inequality state constraints dictated by the physics of the model, which are fundamental to the physical system and often essential to the correct execution of the nonlinear model. We refer to these smoothers as the Constrained Extended Kalman Smoother (C-EKS) and the Constrained Ensemble Kalman Smoother (C-EnKS); they resemble their filtering counterparts, and we refer to the well-known formulations for the EKF and EnKF in the literature [10]. The smoothing methods that satisfy linear constraints are presented as Algorithms 1 and 2 in the following text.

A. State constraints

Physics-based models of multiphysical systems are designed to enforce constraints between states to satisfy fundamental physical behavior. For example, a given direction of fluid flow imposes a pressure gradient which must be enforced by the model, and any estimated states for the model must satisfy this constraint to be valid. Another example of a physics-based constraint is represented by the convective transport of chemical species in a bulk flow, wherein the mass flow rate of the species must be nonnegative.

In general, simulation models (6) of the cycle will satisfy such physical constraints, since the model is based on fundamental physical principles. However, the state estimates obtained from a standard filtering method may not necessarily satisfy these constraints. These constraints must therefore be explicitly incorporated into the estimation process to ensure that the estimated states or parameters do not violate physical laws, and because the computational models are often designed under the assumption that these state constraints are satisfied; constraint violations often represent regions in which the model is invalid, and can result in failures of the model integration process.

The present study is focused on enforcing pressure constraints, though other constraints may also be imposed. We thus constrain the pressure to decrease in the direction of

refrigerant flow. In terms of the state variables P_i, we require

$$P_1 \geq P_2 \geq \cdots \geq P_7 \geq P_8, \tag{9}$$

which can be rewritten to represent 7 linear constraints, i.e.,

$$P_2 - P_1 \leq 0,\ P_3 - P_2 \leq 0,\ \cdots,\ P_8 - P_7 \leq 0. \tag{10}$$

The pressure constraints (10) at a time instant t_k can therefore be written in the compact form

$$\boldsymbol{A}\boldsymbol{x}_k \leq 0, \tag{11}$$

where $\boldsymbol{A} \in \mathbb{R}^{7\times 24}$ is an appropriately defined sparse matrix with non-zero elements ± 1, and the inequality is interpreted element-wise. The constraint (11) must be satisfied at all time steps $k = 1, 2, \cdots, N$.

B. Constrained extended Kalman smoother (C-EKS)

The C-EKS method which enforces state constraints in the form given by (11) is provided in Algorithm 1, in which Equations (12) and (13) constitute the forward filtering pass and the backward smoothing pass of the C-EKS, respectively. An initial estimate of the state at $k = 1$ is assumed to be Gaussian with mean $\hat{\boldsymbol{x}}_1^-$ and covariance $\boldsymbol{P}_1^-$. Note that the covariance matrix is denoted by boldface $\boldsymbol{P}$, whereas the scalar pressure variable is denoted by P in the preceding sections.

Algorithm 1 C-EKS

Data: $\boldsymbol{y}_k$, $k = 1, 2, \cdots, N$
Initialize: $\hat{\boldsymbol{x}}_1^-$, $\boldsymbol{P}_1^-$
for $k = 1, 2, \cdots, N$ **do**

$$\boldsymbol{K}_k = (\boldsymbol{P}_k^- \boldsymbol{H}_k^T)(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{12a}$$
$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{12b}$$
$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-) \tag{12c}$$
$$\hat{\boldsymbol{x}}_k^+, \boldsymbol{P}_k^+ \leftarrow \texttt{constrEK}(\hat{\boldsymbol{x}}_k^+, \boldsymbol{P}_k^+) \tag{12d}$$
$$\boldsymbol{P}_{k+1}^- = \boldsymbol{F}_k \boldsymbol{P}_k^+ \boldsymbol{F}_k^T + \boldsymbol{Q}_k \tag{12e}$$
$$\hat{\boldsymbol{x}}_{k+1}^- = \boldsymbol{f}_k(\hat{\boldsymbol{x}}_k^+) \tag{12f}$$

end for
$\boldsymbol{P}_N \leftarrow \boldsymbol{P}_N^+$, $\hat{\boldsymbol{x}}_k \leftarrow \hat{\boldsymbol{x}}_k^+$
for $k = N, N-1, \cdots, 1$ **do**

$$\boldsymbol{S}_k = \boldsymbol{P}_k^+ \boldsymbol{F}_k^T \left(\boldsymbol{P}_{k+1}^-\right)^{-1} \tag{13a}$$
$$\boldsymbol{P}_k = \boldsymbol{P}_k^+ - \boldsymbol{S}_k \left(\boldsymbol{P}_{k+1}^- - \boldsymbol{P}_{k+1}\right) \boldsymbol{S}_k^T \tag{13b}$$
$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^+ + \boldsymbol{S}_k \left(\hat{\boldsymbol{x}}_{k+1} - \hat{\boldsymbol{x}}_{k+1}^-\right) \tag{13c}$$
$$\hat{\boldsymbol{x}}_k, \boldsymbol{P}_k \leftarrow \texttt{constrEK}(\hat{\boldsymbol{x}}_k \boldsymbol{P}_k) \tag{13d}$$

end for

The first part of the C-EKS algorithm, i.e., subequations in (12) except (12d), is the standard EKF. This filter has been modified to enforce state constraints such as (9). The function `constrEK` in (12d), which will be discussed in turn, returns the updated posterior mean and covariance that satisfy the state constraints if the the posterior estimates obtained from (12b) and (12c) are not consistent with the physical laws.

In equation (12), $\hat{\boldsymbol{x}}_k^+$ denotes the estimated mean of the state vector at time t_k obtained after assimilating all sensor measurements up to time step k, whereas $\hat{\boldsymbol{x}}_k^-$ denotes the predicted state estimate at time t_k obtained after assimilating all sensor measurements up to time step $k-1$. $\boldsymbol{F}_k$ denotes the Jacobian of $\boldsymbol{f}_k$ at $\hat{\boldsymbol{x}}_k^+$, i.e.,

$$\boldsymbol{F}_k := \left.\frac{\partial \boldsymbol{f}_k(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^+}. \tag{14}$$

Computation of the Jacobian is computationally challenging because $\boldsymbol{f}_k$ involves a numerical integral that uses stiff solvers. The solver `QNDF` is compatible with automatic differentiation (AD) functionality in Julia which facilitates the computation of Jacobian matrices.

The second part of the C-EKS algorithm, i.e., subequations in (13) except (13d), are the well-known Rauch-Tung-Striebel (RTS) smoother [11] equations. This smoother has also been modified to enforce state constraints in (13d). The state moments estimated from the RTS smoother are denoted by $\hat{\boldsymbol{x}}_k$, $\boldsymbol{P}_k$ (note that the superscript $+/-$ is omitted for the smoother). The smoothing is initialized at $\hat{\boldsymbol{x}}_N = \hat{\boldsymbol{x}}_N^+$ and $\boldsymbol{P}_N = \boldsymbol{P}_N^+$, and performed backward in time using (13).

As discussed in Section III-A, the estimated states must satisfy the pressure constraint (11). The standard Kalman update step (12c) essentially minimizes the cost function

$$\begin{aligned} \hat{\boldsymbol{x}}_k^+ &= \arg\min_{\boldsymbol{x}} J_k(\boldsymbol{x}), \\ J_k(\boldsymbol{x}) &:= \left\|\boldsymbol{x} - \hat{\boldsymbol{x}}_k^-\right\|^2_{(\boldsymbol{P}_k^-)^{-1}} + \|\boldsymbol{y}_k - \boldsymbol{H}_k\boldsymbol{x}\|^2_{\boldsymbol{R}_k^{-1}}. \end{aligned} \tag{15}$$

The Kalman update step with the state constraints can therefore be implemented by solving the problem

$$\hat{\boldsymbol{x}}_k^+ = \arg\min_{\boldsymbol{x}} J_k(\boldsymbol{x}) \text{ subject to } \boldsymbol{A}\boldsymbol{x} \leq 0.$$

While the solution of the above optimization problem yields a constrained update for the mean, the constraint information is not enforced on the error covariance matrix. Nevertheless, it is expected that the uncertainty in the state estimate would decrease since the system state must satisfy known constraints.

Probability density function (PDF) truncation methods [12] are well-suited for this problem, and are incorporated into C-EKS by first performing the standard extended Kalman update (12c) or (13c) to obtain standard mean and covariance. If the mean violates the constraint, then PDF truncation is performed, after which both the mean and covariance are corrected to satisfy the constraints. As a result, the magnitude of the corrected covariance is reduced, which indicates increased confidence in the state estimate.

The PDF truncation approach discussed in [12] was adopted in the C-EKS. The PDF truncation method sequentially enforces one scalar constraint at a time after an appropriate coordinate transformation. The scalar Gaussian PDF is truncated at the boundaries of the linear constraint by setting the density outside the feasible region to zero. This truncated PDF is then normalized to have the total probability of one within the feasible bounds, and the constrained mean

and covariance are then calculated from the normalized truncated PDF. While the equations for this method can be found in [12], the function `constrEK` essentially returns $\boldsymbol{x}$ and $\boldsymbol{P}$ if the constraint is satisfied, and truncates the PDFs and returns the constrained version of $\boldsymbol{x}$ and $\boldsymbol{P}$ if not.

One implementation aspect which merits consideration is that extended Kalman methods require the computation of Jacobians e.g. $\boldsymbol{F}_k$ at each time step. Although the AD tools in Julia facilitate this computation, the required time and memory can be prohibitive for systems of higher dimension. We therefore explore constrained ensemble-based smoothing methods, which avoid these computations.

C. Constrained ensemble Kalman smoother (C-EnKS)

The notation for the ensemble Kalman framework is as follows. Let M denote the number of samples in the ensemble, i.e., ensemble size. Let $\boldsymbol{x}_k^{(i)+}$ denote the i-th member of the posterior ensemble (indicated by the superscript i in parentheses) at time instant t_k obtained after assimilating sensor measurements up to time step k. Similarly, $\boldsymbol{x}_k^{(i)-}$ denote the i-th member of the prior ensemble at time instant t_k obtained after assimilating sensor measurements up to time step $k-1$. A matrix of all ensemble members is denoted by

$$\boldsymbol{X}_k^+ := \left[\boldsymbol{x}_k^{(1)+}\ \boldsymbol{x}_k^{(2)+}\cdots\ \boldsymbol{x}_k^{(M)+}\right] \in \mathbb{R}^{n_x \times M}$$

whose i-th column is the i-th ensemble member. The matrix of prior ensemble $\boldsymbol{X}_k^-$ is also defined similarly. The covariance matrix $\boldsymbol{P}_k^-$ calculated from the prior ensemble matrix is given by

$$\boldsymbol{P}_k^- = \frac{1}{M-1}\widetilde{\boldsymbol{X}}_k^- \left(\widetilde{\boldsymbol{X}}_k^-\right)^T \tag{16}$$

where $\widetilde{\boldsymbol{X}}_k^-$ denotes the matrix of anomalies, i.e. the matrix of ensemble members' deviations from the ensemble mean $\hat{\boldsymbol{x}}_k^-$, given by

$$\begin{aligned}\widetilde{\boldsymbol{X}}_k^- &= \left[\boldsymbol{x}_k^{(1)-} - \hat{\boldsymbol{x}}_k^- \quad \cdots \quad \boldsymbol{x}_k^{(M)-} - \hat{\boldsymbol{x}}_k^-\right] \\ &= \boldsymbol{X}_k^- \left(\boldsymbol{I} - \frac{\mathbf{1}_M \mathbf{1}_M^T}{M}\right),\end{aligned} \tag{17}$$

and $\mathbf{1}_M \in \mathbb{R}^M$ denotes a column vector of ones.

Let us further denote by $\boldsymbol{x}_{m:k}^{(i)+}$ the augmented vector of posterior states from time steps m to k ($1 \le m \le k$), of the i-th sample trajectory, and by $\boldsymbol{X}_{m:k}^+$ the augmented posterior ensemble matrix, i.e.,

$$\boldsymbol{x}_{m:k}^{(i)+} = \begin{bmatrix}\boldsymbol{x}_m^{(i)+} \\ \boldsymbol{x}_{m+1}^{(i)+} \\ \vdots \\ \boldsymbol{x}_k^{(i)+}\end{bmatrix}, \quad \boldsymbol{X}_{m:k}^+ := \begin{bmatrix}\boldsymbol{X}_m^+ \\ \boldsymbol{X}_{m+1}^+ \\ \vdots \\ \boldsymbol{X}_k^+\end{bmatrix}. \tag{18}$$

The prior augmented state vector of the i-th sample trajectory and the prior augmented ensemble matrix, i.e. $\boldsymbol{x}_{m:k}^{(i)-}$ and $\boldsymbol{X}_{m:k}^-$ are defined analogously. The ensemble covariance matrix $\boldsymbol{P}_{m:k}^-$ of the augmented prior ensemble $\boldsymbol{X}_{m:k}^-$ is calculated similar to (16).

Algorithm 2 C-EnKS

Data: $\boldsymbol{y}_k$, $k = 1, 2, \cdots, N$
Initialize: $\hat{\boldsymbol{x}}_1^-, \boldsymbol{P}_1^-, \boldsymbol{x}_1^{(i)-} \sim \mathcal{N}(\hat{\boldsymbol{x}}_1^-, \boldsymbol{P}_1^-)$, $i = 1, 2\cdots M$
for $k = 1, 2, \cdots, N$ **do**

$$\widetilde{\boldsymbol{X}}_{m:k}^- = \boldsymbol{X}_{m:k}^- \left(\boldsymbol{I} - \frac{\mathbf{1}_M \mathbf{1}_M^T}{M}\right), \tag{19a}$$

$$\boldsymbol{P}_{m:k}^- = \widetilde{\boldsymbol{X}}_{m:k}^- \left(\widetilde{\boldsymbol{X}}_{m:k}^-\right)^T /(M-1) \tag{19b}$$

$\forall i \in \{1, 2\cdots M\}$:

$$\begin{aligned}\boldsymbol{x}_{m:k}^{(i)+} = \boldsymbol{x}_{m:k}^{(i)-} &- \boldsymbol{P}_{m:k}^- \boldsymbol{H}_{m:k}^T (\boldsymbol{H}_{m:k} \boldsymbol{P}_{m:k}^- \boldsymbol{H}_{m:k}^T + \boldsymbol{R}_k)^{-1} \\ &\times (\boldsymbol{H}_{m:k} \boldsymbol{x}_{m:k}^{(i)-} - \boldsymbol{y}_k - \boldsymbol{\eta}_k^{(i)})\end{aligned} \tag{19c}$$

$$\boldsymbol{x}_{m:k}^{(i)+} \leftarrow \texttt{constrEnK}\left(\boldsymbol{x}_{m:k}^{(i)+},\ \boldsymbol{X}_{m:k}^-,\ \boldsymbol{\eta}_k^{(i)}\right) \tag{19d}$$

$$\boldsymbol{x}_{k+1}^{(i)-} = \boldsymbol{f}_k(\boldsymbol{x}_k^{(i)+}) + \boldsymbol{w}_k^{(i)} \tag{19e}$$

where $\boldsymbol{\eta}_k^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_k)$ and $\boldsymbol{w}_k^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k)$.

end for

This notation enables us to present the constrained ensemble Kalman smoother (C-EnKS) in Algorithm 2, where the matrix $\boldsymbol{H}_{m:k} := [\mathbf{0}\ \ \mathbf{0}\ \cdots\ \boldsymbol{H}_k]$ denotes the augmented measurement matrix of appropriate size. When (19d) is omitted, this reduces to the well-known ensemble Kalman smoother (EnKS) [13]. If m is equal to 1, the complete history of the ensemble from the very first time step is updated at each time step k. Because the increase in the size of augmented variables in (18) at each time step makes the problem computationally prohibitive, we implement a fixed-lag smoother in which m is selected at each time step k such that $k-m$ is constant (after sufficient time has passed). This updates the state estimates up to only a certain number of time steps in the past at the current time step k. An alternative selection of $m = k$ recovers the standard EnKF, which updates the state estimates only at the current time step k at which a measurement is obtained.

The EnKF update step can be viewed in much the same way as the EKF update step in (15). The EnKF update step (19c) corresponding to $m = k$ thus minimizes the cost

$$\begin{aligned}\boldsymbol{x}_k^{(i)+} &= \arg\min_{\boldsymbol{x}} J_k^{(i)}(\boldsymbol{x}), \\ J_k^{(i)}(\boldsymbol{x}) &:= \left\|\boldsymbol{x} - \boldsymbol{x}_k^{(i)-}\right\|^2_{(\boldsymbol{P}_k^-)^{-1}} \\ &\quad + \left\|\boldsymbol{y}_k + \boldsymbol{\eta}_k^{(i)} - \boldsymbol{H}_k(\boldsymbol{x})\right\|^2_{(\boldsymbol{R}_k)^{-1}},\end{aligned} \tag{20}$$

and subscript $k:k$ of the augmented variables is replaced with k for the EnKF. The constrained posterior sample is therefore determined by solving the optimization problem in the EnKF [14]

$$\boldsymbol{x}_k^{(i)+} = \arg\min_{\boldsymbol{x}} J_k^{(i)}(\boldsymbol{x}) \text{ subject to } \boldsymbol{A}\boldsymbol{x} \le 0. \tag{21}$$

We now generalize (20) and (21) for the EnKS, i.e. $m < k$. The EnKS update step (19c) corresponding to $m < k$

minimizes the cost function

$$\begin{aligned}\boldsymbol{x}_{m:k}^{(i)+} &= \arg\min_{\boldsymbol{x}_{m:k}} J_{m:k}^{(i)}(\boldsymbol{x}_{m:k}) \\ J_{m:k}^{(i)}(\boldsymbol{x}_{m:k}) &:= \left\|\boldsymbol{x}_{m:k} - \boldsymbol{x}_{m:k}^{(i)-}\right\|^2_{(\boldsymbol{P}_{m:k}^{-})^{-1}} \\ &\quad + \left\|\boldsymbol{y}_k + \boldsymbol{\eta}_k^{(i)} - \boldsymbol{H}_{m:k}\boldsymbol{x}_{m:k}\right\|^2_{\boldsymbol{R}_k^{-1}}.\end{aligned} \tag{22}$$

The constrained EnKS update can be performed by solving (22) subject to state inequality constraints. As the smoother updates the history of states (augmented state vector) from time step m to k, the updated state vector must satisfy the state constraints at each time step over this smoothing window. Consequently, (22) becomes

$$\begin{aligned}\boldsymbol{x}_{m:k}^{(i)+} &= \arg\min_{\boldsymbol{x}_{m:k}} J_{m:k}^{(i)}(\boldsymbol{x}_{m:k}) \\ \text{subject to } \boldsymbol{A}\boldsymbol{x}_q &\leq 0,\ q = m, m+1, \cdots, k,\end{aligned}$$

where $\boldsymbol{x}_{m:k} = [\boldsymbol{x}_m^T, \boldsymbol{x}_{m+1}^T, \cdots, \boldsymbol{x}_k^T]^T$. For a compact representation, let us define $\boldsymbol{A}_{m:k} = \boldsymbol{I} \otimes \boldsymbol{A}$, where $\otimes$ denotes the Kronecker product. The optimization problem for constrained smoother update can thus be written as

$$\begin{aligned}\boldsymbol{x}_{m:k}^{(i)+} &= \arg\min_{\boldsymbol{x}_{m:k}} J_{m:k}^{(i)}(\boldsymbol{x}_{m:k}) \\ \text{subject to } \boldsymbol{A}_{m:k}\boldsymbol{x}_{m:k} &\leq 0.\end{aligned} \tag{23}$$

The prohibitive size and computational cost of the optimization problem (23) for systems with hundreds of states and moderately long smoothing windows ($k-m$) motivates the reformulation of this constrained smoother problem in the covariance range. Previous work [14] developed this formulation strictly for the case of the EnKF, which we extend in an analogous formulation for the EnKS.

Let us denote the optimal correction to an augmented sample by $\boldsymbol{v}_{m:k}^{(i)*}$, which is defined as $\boldsymbol{v}_{m:k}^{(i)*} := \boldsymbol{x}_{m:k}^{(i)+} - \boldsymbol{x}_{m:k}^{(i)-}$. The correction $\boldsymbol{v}_{m:k}^{(i)*}$ lies in the range of covariance matrix $\boldsymbol{P}_{m:k}^{-}$, i.e., $\boldsymbol{v}_{m:k}^{(i)*} \in \mathcal{R}\left(\boldsymbol{P}_{m:k}^{-}\right)$, as seen from (19c). We thus rewrite the optimization problem (22) using the variable substitution $\boldsymbol{v}_{m:k} = \boldsymbol{x}_{m:k} - \boldsymbol{x}_{m:k}^{(i)-}$,

$$\begin{aligned}\boldsymbol{v}_{m:k}^{(i)*} &= \arg\min_{\boldsymbol{v}_{m:k} \in \mathcal{R}\left(\boldsymbol{P}_{m:k}^{-}\right)} \overline{J}_{m:k}^{(i)}(\boldsymbol{v}_{m:k}) \\ \overline{J}_{m:k}^{(i)}(\boldsymbol{v}_{m:k}) &:= \|\boldsymbol{v}_{m:k}\|^2_{(\boldsymbol{P}_{m:k}^{-})^{-1}} + \left\|\widetilde{\boldsymbol{y}}_k^{(i)} - \boldsymbol{H}_{m:k}\boldsymbol{v}_{m:k}\right\|^2_{\boldsymbol{R}_k^{-1}}\end{aligned}$$

where $\widetilde{\boldsymbol{y}}_k^{(i)} := \boldsymbol{y}_k + \boldsymbol{\eta}_k^{(i)} - \boldsymbol{H}_{m:k}\boldsymbol{x}_{m:k}^{(i)-}$ has been defined for notational simplicity. The constrained problem (23) therefore becomes

$$\begin{aligned}\boldsymbol{v}_{m:k}^{(i)*} &= \arg\min_{\boldsymbol{v}_{m:k} \in \mathcal{R}\left(\boldsymbol{P}_{m:k}^{-}\right)} \overline{J}_{m:k}^{(i)}(\boldsymbol{v}_{m:k}) \\ \text{subject to } \boldsymbol{A}_{m:k}\boldsymbol{v}_{m:k} &+ \boldsymbol{A}_{m:k}\boldsymbol{x}_{m:k}^{(i)-} \leq 0.\end{aligned} \tag{24}$$

Since $\boldsymbol{v}_{m:k} \in \mathcal{R}\left(\boldsymbol{P}_{m:k}^{-}\right)$, let us substitute $\boldsymbol{v}_{m:k} = \boldsymbol{P}_{m:k}^{-}\boldsymbol{z}_{m:k}$ where $\boldsymbol{z}_{m:k}$ is a new variable of appropriate dimension. Using the definition of $\boldsymbol{P}_{m:k}^{-}$ (see (16)), the first term in the cost function of (24) becomes

$$\begin{aligned}\|\boldsymbol{v}_{m:k}\|^2_{(\boldsymbol{P}_{m:k}^{-})^{-1}} &= \boldsymbol{v}_{m:k}^T(\boldsymbol{P}_{m:k}^{-})^{-1}\boldsymbol{v}_{m:k} \\ &= \boldsymbol{z}_{m:k}^T\boldsymbol{P}_{m:k}^{-}(\boldsymbol{P}_{m:k}^{-})^{-1}\boldsymbol{P}_{m:k}^{-}\boldsymbol{z}_{m:k} \\ &= \boldsymbol{z}_{m:k}^T\boldsymbol{P}_{m:k}^{-}\boldsymbol{z}_{m:k} \\ &= \frac{1}{M-1}\boldsymbol{z}_{m:k}^T\widetilde{\boldsymbol{X}}_{m:k}^{-}\left(\widetilde{\boldsymbol{X}}_{m:k}^{-}\right)^T\boldsymbol{z}_{m:k} \\ &= \frac{1}{M-1}\boldsymbol{r}^T\boldsymbol{r} = \frac{1}{M-1}\|\boldsymbol{r}\|^2\end{aligned}$$

where $\left(\widetilde{\boldsymbol{X}}_{1:k}^{-}\right)^T\boldsymbol{z}_{1:k} =: \boldsymbol{r} \in \mathbb{R}^M$. Using the variable substitution $\boldsymbol{v}_{1:k} = \boldsymbol{P}_{1:k}^{-}\boldsymbol{z}_{1:k} = \frac{1}{M-1}\widetilde{\boldsymbol{X}}_{1:k}^{-}\boldsymbol{r}$, the optimization problem (24) is now written in terms of the new optimization variable $\boldsymbol{r} \in \mathbb{R}^M$

$$\begin{aligned}\boldsymbol{r}^{(i)*} = \arg\min_{\boldsymbol{r}\in\mathbb{R}^M} &\left(\frac{\|\boldsymbol{r}\|^2}{M-1}\right. \\ &\left. + \left\|\widetilde{\boldsymbol{y}}_k^{(i)} - \frac{1}{M-1}\boldsymbol{H}_{m:k}\widetilde{\boldsymbol{X}}_{m:k}^{-}\boldsymbol{r}\right\|^2_{\boldsymbol{R}_k^{-1}}\right) \\ \text{subject to } \frac{1}{M-1}&\boldsymbol{A}_{m:k}\widetilde{\boldsymbol{X}}_{m:k}^{-}\boldsymbol{r} + \boldsymbol{A}_{m:k}\boldsymbol{x}_{m:k}^{(i)-} \leq 0.\end{aligned}$$

This optimization problem can be further simplified to

$$\begin{aligned}\boldsymbol{r}^{(i)*} &= \arg\min_{\boldsymbol{r}\in\mathbb{R}^M} \boldsymbol{r}^T\boldsymbol{B}_2\boldsymbol{r} - 2\boldsymbol{b}^T\boldsymbol{r} \\ &\text{subject to } \boldsymbol{A}_{m:k}\left(\boldsymbol{B}_1\boldsymbol{r} + \boldsymbol{x}_{m:k}^{(i)-}\right) \leq 0,\end{aligned} \tag{25}$$

where

$$\begin{aligned}\boldsymbol{B}_2 &= \frac{1}{M-1}\boldsymbol{I} + \frac{1}{(M-1)^2}\left(\boldsymbol{H}_{m:k}\widetilde{\boldsymbol{X}}_{m:k}^{-}\right)^T\boldsymbol{R}^{-1}\boldsymbol{H}_{m:k}\widetilde{\boldsymbol{X}}_{m:k}^{-}, \\ \boldsymbol{b}^T &= \frac{2}{M-1}\left[\left(\widetilde{\boldsymbol{y}}_k^{(i)}\right)^T\boldsymbol{R}^{-1}\boldsymbol{H}_{m:k}\widetilde{\boldsymbol{X}}_{m:k}^{-}\right], \\ \boldsymbol{B}_1 &= \frac{1}{M-1}\widetilde{\boldsymbol{X}}_{m:k}^{-}\end{aligned}$$

Problem (25) is a quadratic program with linear inequality constraints which can be solved using standard software packages. The advantage of this formulation is that $\boldsymbol{r} \in \mathbb{R}^M$ is the optimization variable in (25), so that the size of the quadratic program to be solved is constrained by the ensemble size M. In comparison to the limited size of $\boldsymbol{r}$, the size of the quadratic program to be solved in (23) can be arbitrarily large, depending on the length of smoothing window for the optimization variable $\boldsymbol{x}_{m:k}$.

Once the optimal $\boldsymbol{r}^{(i)*}$ is determined, the updated i-th sample is given by the inverse variable transformation, i.e.

$$\boldsymbol{x}_{1:k}^{(i)+} = \boldsymbol{x}_{1:k}^{(i)-} + \boldsymbol{B}_1\,\boldsymbol{r}^{(i)*}.$$

This can be incorporated into pseudo code for the function `constrEnK` in (19d) as given below.

function `constrEnK`$\left(\boldsymbol{x}_{m:k}^{(i)+},\ \boldsymbol{X}_{m:k}^{-},\ \boldsymbol{\eta}_k^{(i)}\right)$
if $\boldsymbol{A}_{m:k}\boldsymbol{x}_{m:k}^{(i)+} \leq 0$ **then**
  return $\boldsymbol{x}_{m:k}^{(i)+}$
else
  determine $\boldsymbol{r}^{(i)*}$ by solving (25)

return $\left(\boldsymbol{x}_{1:k}^{(i)-} + \boldsymbol{B}_1 \boldsymbol{r}^{(i)*}\right)$
end if
end function

IV. Numerical Results

For the purpose of numerical simulations, the model is discretized with the time step of 0.1 s, i.e., $\Delta t := t_{k+1} - t_k = 0.1$ s. The process noise covariance matrix $\boldsymbol{Q}_k \in \mathbb{R}^{24\times 24}$ is assumed to be a diagonal matrix such that its principal diagonal is the vector $\mathbf{1}_8 \otimes [7\times 10^5, 6\times 10^4, 2\times 10^{-3}]^T$. The numerical values of the process covariance are primarily dictated by the stiffness of the cycle model. Higher variance process noise was found to make the system unstable leading to failure of the numerical integration in (5). However, these covariance values are sufficiently large to generally characterize the potential uncertainty arising due to external noises as well as the uncertain cycle parameters. Noise variance of the temperature sensors (σ_θ^2) and pressure sensors (σ_P^2) are assumed to be $\sigma_\theta^2 = 1$ K^2 and $\sigma_P^2 = 5\times 10^4$ Pa^2, based on the characteristics of available hardware. All sensor channels are assumed to be independent of each other such that the sensor noise covariance matrix $\boldsymbol{R}_k$ is diagonal with non-zero elements σ_θ^2 and σ_P^2.

The model is simulated for 500 time steps (i.e., 50 s). Compressor speed is assumed to be a piecewise constant function with a single jump at $t = 10$ s from 50 Hz to 55 Hz, while the expansion valve position remains constant except for a jump at $t = 30$ s from 56.4% to 62.4%. A similar change in the outdoor ambient temperature was also incorporated by changing from 305.15 K to 308.15 K at $t = 15$ s. An ensemble of 50 trajectories was used for ensemble Kalman based methods with covariance inflation of 1%.

Fig. 2 illustrates some of the estimated state trajectories from the C-EKS and C-EnKS methods that were not measured directly by the sensors, where the reference trajectory for each given variable is denoted by a black solid line. The estimated state trajectories (both C-EKS and C-EnKS) match the actual trajectories well, and significant deviation of the estimated trajectories from the reference trajectories occurs only during the initial time interval, as the uncertainty associated with the initial condition of the cycle is large. The standard deviations of the initial state variables are assumed to be 10% of their values for pressure and specific enthalpy, and 1% for temperature. In general, deviations of the estimated trajectories from the reference trajectory quickly reduce as additional sensor measurements are assimilated. Similar trends were observed for other state variables not shown here.

For all the states, the percentage estimation error is calculated pointwise, i.e. percentage estimation error is calculated for every state at each time step. The mean pointwise percentage error was 0.16% for C-EKS and 0.22% for C-EnKS, while the standard deviation of this error was 0.42% for C-EKS and 0.54% for C-EnKS. The maximum pointwise percentage error was found to be 17.62% for C-EKS and 12.33% for C-EnKS. The relatively large error of the C-EnKS can mainly be attributed to a small ensemble size.

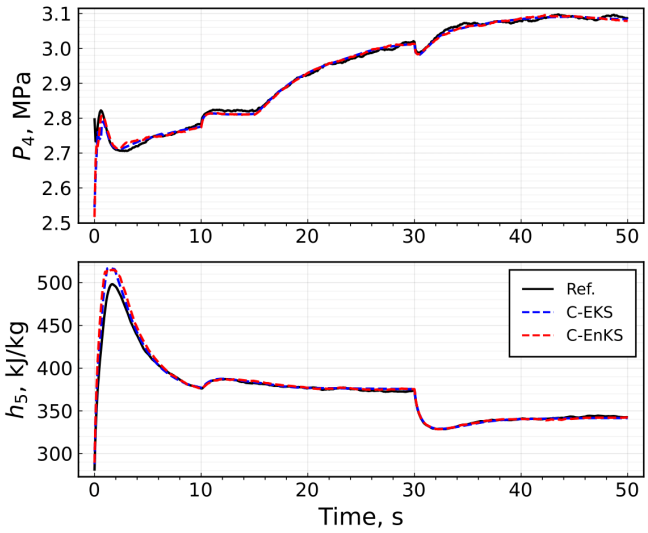

Fig. 2. Estimated states pressure P_4 and specific enthalpy h_5 which are not directly measured.

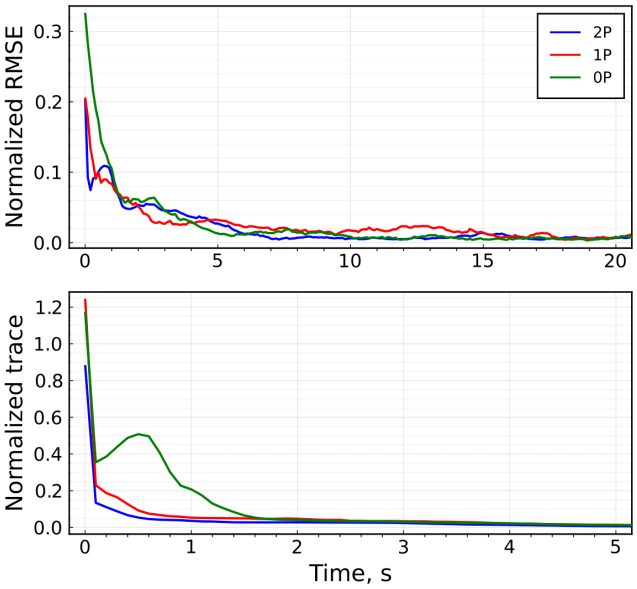

Fig. 3. Normalized RMSE and trace of error covariance matrix obtained using C-EnKS with different sensor sets. Normalization constants for pressure, specific enthalpy, and temperature are 10^6 Pa, 10^5 J/kg, and 100 K respectively.

One particular topic of study was the effect of the number of pressure sensors on state estimates, as these sensors are typically one to two orders of magnitude more expensive than temperature sensors. C-EnKS was thus employed as a representative for state estimation with three different sensor sets. The initial sensor set described in (8) is denoted by '2P', indicating that there are two pressure sensors. Sensor set '1P' was obtained after eliminating P_1 measurement from (8), while the sensor set '0P' did not include any sensors.

After performing this comparison, the mean pointwise percentage error was 0.22% for the 2P set, 0.30% for the 1P set, and 0.32% for the 0P set, while the standard deviation of this error was 0.54% for the 2P set, 0.62% for the 1P set, and 0.96% for the 0P set. Similarly, the maximum pointwise percentage error was found to be 12.33% for the

2P set, 15.39% for the 1P set, and 28.62% for the 0P set. As expected, the estimation error increases as the number of pressure sensors in the system decreases, though the estimation accuracy of 2P and 1P are largely comparable. It is interesting to note that the complete elimination of pressure measurements significantly increases the estimation errors, as seen from error statistics corresponding to 0P. We therefore conclude that at least one pressure sensor measurement is recommended for good estimation accuracy. Fig. 3 shows the trace of error covariance matrix and RMSE (both normalized) corresponding to these different sensor sets. Significantly larger estimation errors corresponding to 0P sensor set are mainly attributed to initial time interval of the trajectory. With time, the estimation errors for all sensor sets become comparable. The estimation accuracy also appears to be affected by the specific location of the pressure sensor; when there is only one pressure sensor that measures P_1 instead of P_8, the estimation error is slightly larger (mean 0.26%, standard deviation 0.81%, maximum 23.55%) than the 1P set shown in Fig. 3. Therefore, it is favorable to place a pressure sensor in the evaporator, rather than the condenser.

We also evaluated the effect and importance of incorporating constraints like (11) in the estimation process in comparison to methods without such a mechanism. Algorithms 1 and 2 were therefore implemented without enforcing the state constraints (i.e. by omitting (12d), (13d), (19d)), and are referred to as EKS and EnKS respectively. We observed that the EKS may produce physically inconsistent state estimates at initial few time steps, but that the state estimates obtained using EKS at later time instants satisfy the state constraints. The percentage estimation error of EKS (with mean 0.16%, standard deviation 0.37%, and maximum 17.54%) and trace of covariance matrix (on an average 1.04 times C-EKS) was found to be quite similar to the C-EKS.

A similar comparison of C-EnKS and EnKS showed that their estimation accuracy is comparable (with mean 0.21%, standard deviation 0.51%, and maximum 12.04%), though C-EnKS has a slightly smaller error covariance during the initial part of the trajectory. This is evident in Fig. 4, which illustrates the smaller uncertainty in the state estimate. During the initial time period, almost all sample trajectories as produced by EnKS were found to violate the state constraints. The number of sample trajectories that violate the state constraints kept decreasing with time, with only few violating the constraints at later time instants. Nevertheless, these constraint enforcement mechanisms were important for the successful forward solution of the problem, as constraint violations could break the model assumptions and prevent the estimator from functioning for a meaningful duration.

V. Conclusions & Discussion

In this work, we developed and demonstrated the effectiveness of a framework for state estimation in vapor compression cycles, with a particular focus on ensemble and extended Kalman smoothing algorithms (C-EKS and C-EnKS) that incorporate physical constraints on the system states. Constraint satisfaction is important for these models to ensure that their assumptions are satisfied and that the model functions correctly. Numerical results show that classical Kalman-based approaches for state estimation work well on cycle models if paired with appropriate modeling and computational tools. Next steps for this work include the extension of these methods to models with higher fidelity, such as the use of moist air heat exchanger models, and the further development of iterative or particle filtering approaches for these problems.

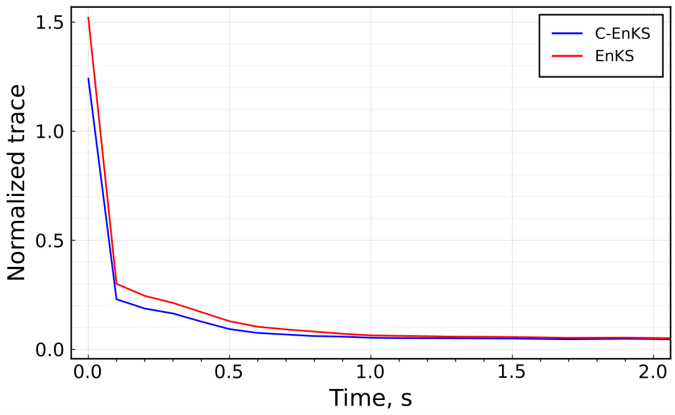

Fig. 4. Normalized trace of error covariance matrix obtained using C-EnKS and EnKS.

References

[1] IEA. The future of cooling. Technical report, International Energy Agency, Paris, 2018.

[2] S.A. Bortoff, P. Schwerdtner, C. Danielson, and S. Di Cairano. H-infinity loop-shaped model predictive control with heat pump application. In *18th European Control Conference*, pages 2386–2393, 2019.

[3] P. Krupa, C. Danielson, C. Laughman, S.A. Bortoff, D.J. Burns, S. Di Cairano, and D. Limon. Modelica implementation of centralized MPC controller for a multi-zone heat pump. In *18th European Control Conference*, pages 1784–1789, 2019.

[4] T. Cheng, X.-D. He, and H.H. Asada. Nonlinear observer design and experimental verification for heat exchangers during the start-up process. In *2005 American Control Conference*, 2005.

[5] M. Ghousein and E. Witrant. A boundary observer for two phase heat exchangers. In *18th European Control Conference*, pages 2332–2337, Napoli, Italy, 2019.

[6] H. Qiao, V. Aute, and R. Radermacher. Transient modeling of a flash tank vapor injection heat pump system–Part I: Model development. *International Journal of Refrigeration*, 49:169–182, 2015.

[7] Yingbo Ma, Shashi Gowda, Ranjan Anantharaman, Chris Laughman, Viral Shah, and Chris Rackauckas. ModelingToolkit: A composable graph transformation system for equation-based modeling, 2021.

[8] Constantinos C Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistical Computing*, 9(2):213–231, 1988.

[9] Lawrence F Shampine and Mark W Reichelt. The MATLAB ODE suite. *SIAM Journal on Scientific Computing*, 18(1):1–22, 1997.

[10] Geir Evensen. The ensemble Kalman filter for combined state and parameter estimation. *IEEE Control Systems Magazine*, 29(3):83–104, 2009.

[11] John L. Crassidis and John L. Junkins. *Optimal Estimation of Dynamic Systems*. Chapman and Hall/CRC, 2011.

[12] Dan Simon and Donald L. Simon. Constrained Kalman filtering via density function truncation for turbofan engine health estimation. *International Journal of Systems Science*, 41(2):159–171, 2010.

[13] J. Mandel, E. Bergou, S. Gürol, S. Gratton, and I. Kasanický. Hybrid Levenberg–Marquardt and weak-constraint ensemble Kalman smoother method. *Nonlinear Processes in Geophysics*, 23(2):59–73, 2016.

[14] David J Albers, Paul-Adrien Blancquart, Matthew E Levine, Elnaz Esmaeilzadeh Seylabi, and Andrew Stuart. Ensemble Kalman methods with constraints. *Inverse Problems*, 35(9):095007, aug 2019.