Estimation: A Key Technology for Digital Twins

Bortoff, Scott A.; Laughman, Christopher R.

TR2022-127 October 12, 2022

Abstract

A digital twin is a set of computer models that serve as a real-time digital counterpart of a physical object or process. The term was coined in the early 2000s in the context of Product Lifecycle Management (PLM)(1) to describe a set of computer representations of a product as it evolves through its lifecycle, from design to manufacture, then to operation, and finally to disposal. The digital twin was envisioned as an electronic repository of all aspects of design, such as 3-D CAD drawings and engineering models, in addition to operational descriptions such as bills of process. It is maintained throughout the product lifecycle via a real-time data stream of measurements obtained from the physical object. It is used to monitor and predict the behavior of the product in operation in its physical environment for diagnostic purposes, or in interrogative use cases in which past or future scenarios are analyzed to improve product design or operation.

Mitsubishi Sanden Gihou 2022

デジタルツインの鍵となる推定技術

Estimation: A Key Technology for Digital Twins

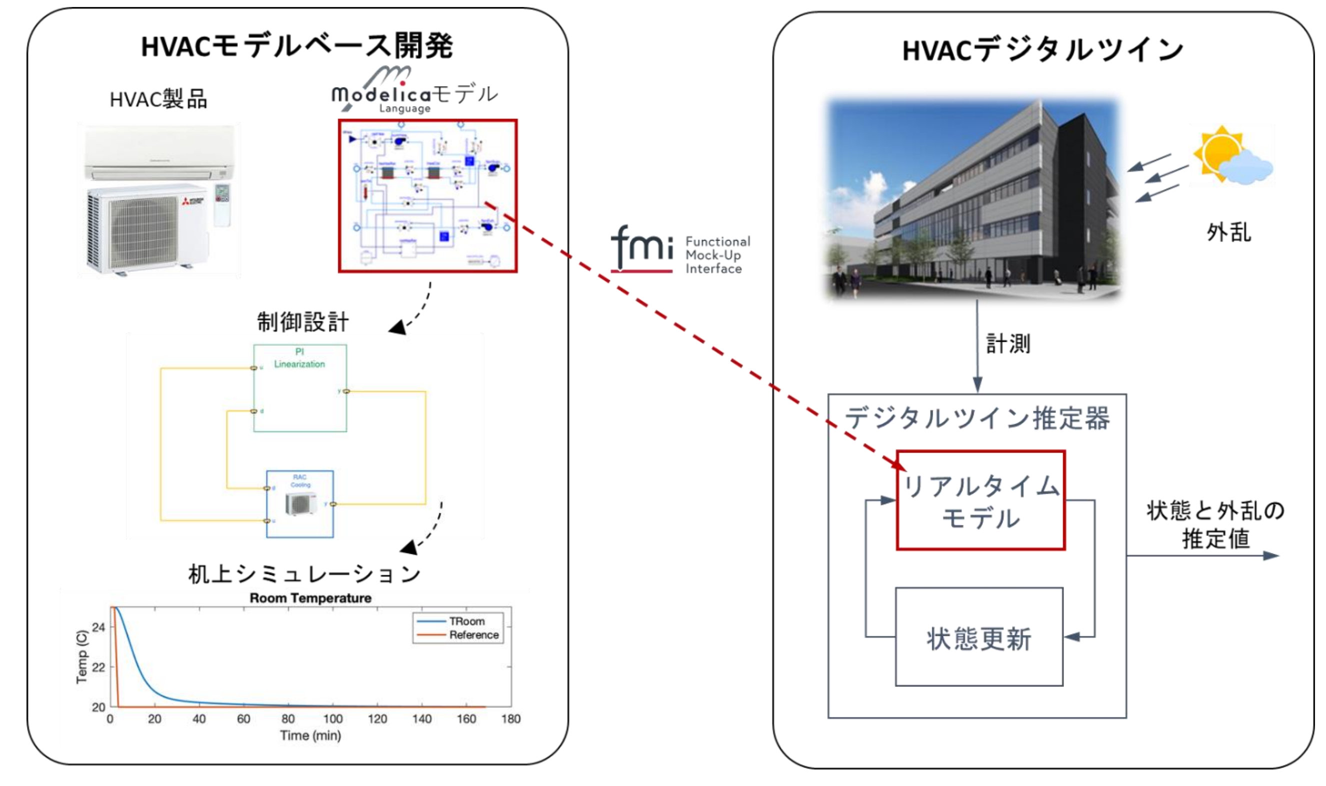
要　旨

26 文字　26 文字

12 行

デジタルツインは、物理的なモノやプロセスにリアルタイムで対応する、デジタル側のコンピュータモデルのセットである。このモデルセットは、モノやプロセスの製品ライフサイクルを通じて、継続的に進化していく。製品を開発する際に使用したシミュレーションモデルは、製品運用時にはデジタルツインとして、リアルタイムで行う計測と組み合わされて再利用される。このようにモデルと実際の計測を組み合わせ、運用時にリアルタイムで利用することで、直接計測するのが困難な場合や高コストの場合に、仮想的なセンシング、診断、モデル予測制御といったことが可能になる。このような利点を得るには、製品開発時には既知と仮定した境界条件や外乱を、運用時の状況に合わせて扱えるように、モデルを修正しなければならない。さらに、推定アルゴリズムを使用して、製品の物理的条件や外気温のような環境条件を拘束条件として考慮した上で、運用時のリアルタイムデータを推定できるようにモデルを修正していく必要がある。また、ツールソフトウェアが、モデルに基づく推定をサポートしている必要もある。

本論文では、ビルＨＶＡＣシステムを対象に、デジタルツインのための推定技術について解説する。また、製品開発時に作成したモデルを運用時に再利用可能とするための有望な技術プラットフォームである、Ｍｏｄｅｌｉｃａとそれと関連するＦＭＩ標準、Ｊｕｌｉａ言語といった次世代モデリングとシミュレーション技術についても紹介する。



ModelicaおよびfmiはModelica Associationの登録商標または商標です。

HVAC 向けのデジタルツイン

3 行

ＨＶＡＣ製品のモデルベース設計では、シミュレーションモデルを使用して制御システムの設計を行い、決定した設計内容を検証する。この時に使用したモデルは、モデルベース状態推定アルゴリズムを実装しているＭｏｄｅｌｉｃａやＦＭＩ（Ｆｕｎｃｔｉｏｎａｌ　Ｍｏｃｋ－Ｕｐ　Ｉｎｔｅｒｆａｃｅ）などを利用することで、デジタルツインの一部として運用時に再利用できる。

66 文字

Julia は Julia Computing, Inc. の登録商標です。

55 – 60 lines

1. Introduction

A *digital twin* is a set of computer models that serve as a real-time digital counterpart of a physical object or process. The term was coined in the early 2000s in the context of Product Lifecycle Management (PLM)(1) to describe a set of computer representations of a product as it evolves through its lifecycle, from design to manufacture, then to operation, and finally to disposal. The digital twin was envisioned as an electronic repository of all aspects of design, such as 3-D CAD drawings and engineering models, in addition to operational descriptions such as bills of process. It is maintained throughout the product lifecycle via a real-time data stream of measurements obtained from the physical object. It is used to monitor and predict the behavior of the product in operation in its physical environment for diagnostic purposes, or in interrogative use cases in which past or future scenarios are analyzed to improve product design or operation.

Here we consider digital twins of building Heating, Ventilation, and Air Conditioning (HVAC) systems, and define the digital twin narrowly to be a physics-based simulation model that is combined with measurements and used in real-time operation. It may provide a range of benefits, such as the following:

- **Virtual Sensing:** Heat flow through a heat exchanger, which is expensive to measure directly, may be estimated using a model and a limited set of sensor measurements. If sufficiently accurate, it may serve as a utility-grade meter for billing purposes.
- **Diagnostics:** The amount and location of the refrigerant charge inside HVAC equipment, which is difficult to measure directly, may be estimated and used to identify costly refrigerant leaks.
- **Model Predictive Control:** The digital twin model may be integrated into a product-level or building-level model predictive control (MPC), which can command actuator values that optimize a cost function, such as energy use.

For each of these use cases, a simulation model of the HVAC equipment, and possibly the building, is combined with real-time measurements to estimate a quantity of interest that is unavailable or difficult to measure directly. Such simulation models are used in product development, and may be reused for this purpose. However, their use during the equipment operation phase differs from their use in product development, and the simulation model must be modified accordingly. The purpose of this article is to describe these modifications and introduce next-generation modeling and simulation technologies that facilitate them.

2. HVAC Models in Product Development

HVAC equipment often uses vapor compression cycles to move heat among a set of heat exchangers located throughout a building to provide thermal comfort and ventilation. The relevant physical processes include heat transfer, thermodynamics, and fluid mechanics. Mathematically, these processes are modeled with a set of differential and algebraic equations (DAEs),

$$\dot{x}(t) = f_c(x(t), d(t), u(t), \theta) \tag{1a}$$

$$y(t) = h(x(t)) \tag{1b}$$

Here, $x(t)$ is an n-dimensional vector of states, $\dot{x}(t)$ denotes its time derivative, and $y(t)$ is a set of time-varying measurements. The states include quantities such as refrigerant pressures and heat exchanger (HEX) wall metal temperatures at discrete spatial locations, as well as building construction temperatures, air temperatures, and humidities. The term f_c is a nonlinear function of the states $x(t)$, time-varying boundary conditions or disturbances $d(t)$ that are typically not measured, control actuators $u(t)$ such as compressor speeds, and a vector of parameters θ that are typically calibrated for the specific problem of interest. The term h is a nonlinear function that defines the measurements $y(t)$.

In practice, models are constructed using CAE tools such as the computer language *Modelica.* This tool automates model assembly using compiler technology to produce numerically efficient simulation code. During product development, such models are used to validate design decisions and in methods of model-based design of the control system.

In these use cases, values for $x(t_0)$, θ, and $d(t)$ are given. For example, the weather acts as a disturbance on a building HVAC simulation. It can be provided in the form of typical meteorological year data, which is available open-source for many worldwide locations. Similarly, the initial condition is assumed, values of θ are obtained from laboratory or catalog data, and values of $u(t)$ are computed by the controller.

3. HVAC Models in Real-Time Operation

The following issues must be addressed to reuse a simulation model in a digital twin that was intended for product development:

55 – 60 lines

- The state $x(t)$ and the disturbance $d(t)$ must be estimated.
- Values for elements of the parameter vector θ must be calibrated.
- Constraints associated with the model (1) must be enforced.
- Real-time measurements $y(t)$ and system inputs $u(t)$ need to be incorporated.

Furthermore, the tools used to represent and simulate the model must support these needs.

3.1 Conventional State Estimation

Disturbances or boundary conditions are conventionally estimated by modeling them as a Laplace-transformable signal, augmenting the physics-based model with the signal model, and estimating the full state. Because $d(t)$ typically varies slowly, it may be represented as an unknown constant $d = x_2$ and augmented to (1) as

$$\dot{x}_1 = 0 \tag{2a}$$

$$\dot{x}_2 = f_c(x_1, x_2, u, \theta) \tag{2b}$$

$$y = h(x_2) \tag{2c}$$

where $x(t) = [x_1(t)\ \ x_2(t)]^T$ is redefined.

The extended Kalman filter (EKF) is the most conventional method for estimation of $x(t)$. All variations of the EKF share the same general structure. Numerically integrating (2) gives the discrete-time augmented model

$$x_{k+1} = f(x_{1k}, x_{2k}, u_k, \theta) \tag{3a}$$

$$y_k = h(x_{1k}), \tag{3b}$$

from which the EKF is realized as the set of prediction equations

$$\hat{x}_{k|k-1} = f\big(\hat{x}_{1k-1|k-1}, u_k, \hat{x}_{2k-1|k-1}, \theta\big) \tag{4a}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k, \tag{4b}$$

together with the set of correction or update equations,

$$\tilde{y}_k = y_k - h(\hat{x}_{k|k-1}) \tag{5a}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{5b}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{5c}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{5d}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{5e}$$

which are solved iteratively, where

$$F_k = \frac{\partial f}{\partial x}\bigg|_{\hat{x}_{k-1|k-1}, u_k} \text{ and } H_k = \frac{\partial h}{\partial x}\bigg|_{\hat{x}_{k|k-1}}. \tag{6}$$

The EKF is the heart of the digital twin, as it combines model predictions with measurement feedback to estimate unmeasured states and boundary conditions, and predicts future behavior. However, it can perform poorly for two reasons. First, the state update (5d) can produce non-physical values for states, such as relative humidity values above 100%. When these are substituted into the prediction model (4a), the model may fail because f was not defined for these non-physical conditions. Second, the Jacobians (6) need to be evaluated accurately. An inaccurate numerical approximation may cause the EKF to diverge. This occurs because (1) is numerically stiff, and the Jacobian can be ill conditioned. Therefore, constraints on the states must be explicitly enforced when the state is updated, and symbolic expressions for the Jacobians are needed. Even then, the EKF may diverge and alternative algorithms such as the constrained ensemble Kalman filter (EnKF)(4) and optimization-based constrained estimators(3) should be considered.

3.2 Constrained EKF

The standard EKF state update (5d) is the solution to

$$\hat{x}_{k|k} = \arg\min_x J_k(x), \tag{7}$$

where

$$J_k(x) = \left\| x - \hat{x}_{k|k-1} \right\|_{P^{-1}_{k|k-1}} + \left\| y_k - H_k x \right\|_{R_k^{-1}} \tag{8}$$

If the constraints on the state can be represented as $Ax \in \mathcal{A}$ for some matrix A and convex set $\mathcal{A}$, then (5d) may be computed instead by numerically solving

$$\hat{x}_{k|k} = \arg\min_x J_k(x) \text{ subject to } Ax \in \mathcal{A}. \tag{9}$$

This enforces the constraint on the mean $\hat{x}_k$ and ensures that the forward simulation model will compute $\hat{x}_{k+1|k}$ in (4). As this method does not enforce a constraint on $P_{k|k-1}$, probability density function (PDF) truncation methods may be used for this purpose. These methods first compute the standard state update (5d), and if $\hat{x}_{k|k} \notin \mathcal{A}$, then PDF truncation is performed, and $\hat{x}_{k|k}$ and $P_{k|k}$ are corrected to satisfy the constraints. For systems with hundreds of states, this is an effective way to realize a building HVAC digital twin(2).

For larger systems, the constrained EKF is numerically expensive. The constrained ensemble Kalman filter is an alternative for such cases. A conventional ensemble Kalman filter simulates an ensemble of initial conditions and computes the covariance numerically. State constraints can be enforced by solving a numerical optimization similar to (9). This does not require any Jacobian calculation, and the covariance propagation does not require a matrix inversion, making it suitable to large scale digital twin applications.

4. Tools for HVAC Digital Twins

Modelica[1] is a computer language for modeling multiphysical, heterogeneous systems such as HVAC systems in buildings. It is equation-based so that mathematical

[1] https://modelica.org

55 – 60 lines

equations may be transcribed naturally into the language. It is also object-oriented for organization, so that large models of complex, hierarchical systems may be composed from libraries of components. This composability allows the structure of the model to mimic the structure of the physical system. Together, these characteristics enable reuse of the Modelica model in its operational phase as a digital twin.

A key technology is the Functional Mockup Interface (FMI)[2], which is a standard for sharing and simulating Modelica models. A Modelica model is compiled into a Functional Mockup Unit (FMU), which is a software package that allows the simulation to be executed on a variety of platforms, such as in Python[3] or MATLAB[4]. An FMU allows for two operations that enable realization of constrained estimators. First, the state may be reset, enabling the state update (4a) and (5d) or (9). Second, Jacobians (6) may be computed, although the standard allows for these to be computed numerically. This means a constrained estimator can be realized in Python or MATLAB in a few lines of code.

However, Modelica and FMI do have limitations. There is no gradient operator in the Modelica specification, so it is not possible to compute a symbolic Jacobian. Moreover, as an FMU is an executable, the symbolic aspects of the original model are compiled away.

The more recently developed project ModelingToolkit.jl, written in the Julia language, is a modeling package built on a symbolic computational algebra framework. It has a number of features that make it particularly appropriate for implementing HVAC digital twins. Like Modelica, it is equation-based and object-oriented, allowing users to construct large, acausal system models from component models. Julia and ModelingToolkit also allow for more symbolic analyses, such as automatic differentiation. This means that a Jacobian may be computed symbolically, ensuring that it is highly accurate when it is evaluated.

We recently developed constrained estimators using ModelingToolkit.jl for a small model of an R32-based vapor-compression cycle with 278 equations. Measurements included the compressor suction and discharge pressures and wall temperatures at the inlet and outlet of each HEX. Smoothing versions of the constrained EKF and EnKF operated on these measurements to generate estimates of two unobserved variables: the pressure at the condenser outlet and the specific enthalpy at the LEV inlet, as shown in Figure 1. The estimates track the unobserved variables closely, with a mean percentage error of less than 1%(4).

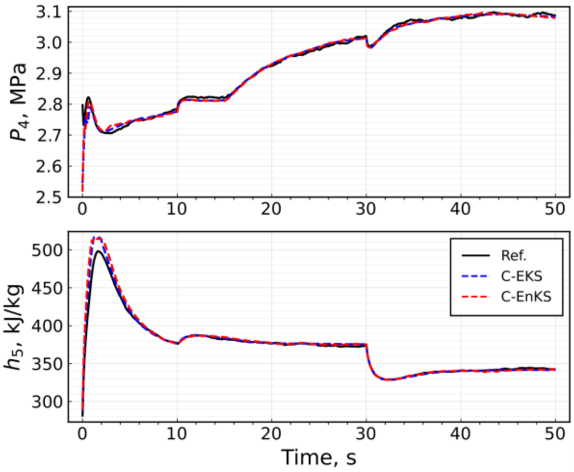


Figure 1: Unobserved and estimated condenser outlet pressure and LEV inlet specific enthalpy for prototype vapor-compression cycle written in ModelingToolkit.jl.

5. Conclusion

Simulation models used for product development can be reused as a ``digital twin" in product operation, but often require modification for this purpose. For an HVAC digital twin, an estimator must be designed and realized in real-time software to combine model predictions and measurements. Some estimators require analytic representations of modeling artifacts, such as Jacobians. Advanced CAE modeling tools, such as Modelica and Julia, support these operations, and should be adopted broadly since their use not only supports model-based design in product development, but also can be extended to development of digital twins.

References

(1) Grieves, M., “Origins of the Digital Twin Concept,” ResearchGate, August 2016.

(2) Bortoff, S.A., et al., “An extended Luenberger observer for HVAC application using FMI”, 13th Int. Modelica Conf., 2019.

(3) Goodwin, G.C., Seron, M.M., De Dona, J.A., “Constrained Control and Estimation: An Optimization Approach,” Springer, 2005.

(4) Deshpande, V., Laughman, C.R., Ma, Y., Rackauckas, C., “Constrained smoothers for state estimation of vapor compression cycles,” 2022 American Control Conference, June 2022.

[2] https://fmi-standard.org.

[3] Python is a registered trademark of the Python Software Foundation.

[4] MATLAB is a registered trademarks of The MathWorks, Inc.