Learning Optimization-based Control Policies Directly from Digital Twin Simulations

Menner, Marcel; Chakrabarty, Ankush; Berntorp, Karl; Di Cairano, Stefano

TR2022-108 September 29, 2022

Abstract

This paper proposes to use a digital twin of a dynamical system directly for optimization-based control. It proposes an algorithm based on an Unscented Kalman Filter (UKF) to solve optimization-based control problems, where the system dynamics is encoded in the digital twin. The UKF- based algorithm uses simulations of a digital twin directly to optimize the control policy and does not require gradients to be computed—making it suitable for differential-algebraic constraints, where gradients may be inaccessible. The proposed UKF-based algorithm does not require explicit knowledge of the internal model of the digital twin, nor the control map; that is, it is a purely simulation data-driven approach. The main advantage is that a high-precision simulation-oriented digital twin can approximate the physical dynamical system more accurately than an analytical control-oriented model and thus, can improve the performance of the controller. The digital twin-based optimal control approach is evaluated on two case studies. First, a pendulum on a cart is optimized to swing up and stabilize. Second, a crane controller is optimized to avoid oscillations of the load.

IEEE Conference on Control Technology and Applications (CCTA) 2022

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

Learning Optimization-based Control Policies Directly from Digital Twin Simulations

Marcel Menner, Ankush Chakrabarty, Karl Berntorp, and Stefano Di Cairano

Abstract— **This paper proposes to use a digital twin of a dynamical system directly for optimization-based control. It proposes an algorithm based on an Unscented Kalman Filter (UKF) to solve optimization-based control problems, where the system dynamics is encoded in the digital twin. The UKF-based algorithm uses simulations of a digital twin directly to optimize the control policy and does not require gradients to be computed—making it suitable for differential-algebraic constraints, where gradients may be inaccessible. The proposed UKF-based algorithm does not require explicit knowledge of the internal model of the digital twin, nor the control map; that is, it is a purely simulation data-driven approach. The main advantage is that a high-precision simulation-oriented digital twin can approximate the physical dynamical system more accurately than an analytical control-oriented model and thus, can improve the performance of the controller. The digital twin-based optimal control approach is evaluated on two case studies. First, a pendulum on a cart is optimized to swing up and stabilize. Second, a crane controller is optimized to avoid oscillations of the load.**

I. Introduction

Model-based control approaches, such as linear quadratic regulator (LQR) or model predictive control (MPC), use a mathematical model of a dynamical system to determine actuator commands. Mathematical models of dynamical systems are often simplified to facilitate numerical optimization methods; we refer to such simplified models as *control-oriented* models. Such control-oriented models are often designed using smooth analytical functions that are suited to gradient-based optimization [1]. While these control-oriented models have been demonstrably useful, modern control engineering applications are becoming increasingly sophisticated, and therefore, require significantly higher modeling complexity to generate accurate predictions. Consequently, controller design methodologies that rely on simplified control-oriented models (e.g., linear state-space or transfer-function) are becoming limited in applicability. With the advancement of computational resources and simulation software, there is a need to design controllers that can make use of high-fidelity simulators. Such simulators can contain rule-based/heuristic, neural network-based, and other non-smooth underlying dynamical models to emulate the complexities of contact dynamics, friction, inertia of complex shapes, flexible bodies as in soft robotics, and human behaviors.

A. Contributions

Conceptual Idea: In this paper, we propose a simulation-model-based optimal control approach that leverages predictions made by a high-fidelity simulator, or digital twin, rather than a simplified control-oriented model. The simulation model implicitly represents the differential-algebraic constraints, i.e., the system dynamics. It can have a very general structure and can include discontinuities, friction models, contact dynamics, and maps generated from data. The fundamental idea is that a simulation model such as a digital twin has the potential to approximate the physical dynamical system more accurately than an analytical model, and thus can improve the performance of a controller designed using such a twin.

Algorithmic Realization: This paper proposes an algorithm for digital twin-based optimal control using a gradient-free implementation. The algorithm is based on an Unscented Kalman Filter (UKF), which "estimates" the optimal input sequence to the dynamical system. The UKF uses evaluations of so-called sigma points [2], which are realizations of input sequences to the dynamical system. The evaluations of the sigma points are used to iteratively update the optimizer. Hence, the algorithm does not rely on gradients, but on evaluations of input sequences—the sigma points—making it suitable for optimal control using a digital twin.

Results: We present two case studies. First, we use a digital twin of a pendulum on a cart in order to control the system to swing up the pendulum. Results show that the proposed approach can control an unstable dynamical system with high precision. Second, we use a digital twin of a crane in order to control its motion to avoid oscillations of the load/beam. Results show that the algorithm can effectively improve the operation of a complex dynamical system.

B. Related Work

Digital twins are often utilized to improve processes through simulation-based design [3]–[5]. A gradient-based sequential quadratic programming method for set-point optimization is presented in [6], where a simulation model is used to improve accuracy. In [7], data from a simulation model are used to iteratively update the solution of the Hamilton–Jacobi–Bellman equation. In [8], a digital twin is utilized for design and evaluation. In [9], a digital twin for thermal processes is developed and suggested for control design. In [10], a digital twin is utilized for an automated conveyer system. In [11], a data-driven approach is developed to automate smart manufacturing systems. In [12], a digital twin is used to optimize processes or controllers, which is closely related to reinforcement learning. Compared to [3]–[12], we propose a concept to directly use a digital twin for optimal control as an encoding of the differential-algebraic

All authors are with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, 02139, USA (e-mail: menner@ieee.org, achakrabarty@ieee.org, karl.o.berntorp@ieee.org, dicairano@ieee.org).

constraints, rather than using a digital twin indirectly by calibrating a controller or optimizing a process.

In reinforcement learning (RL) [13] and Bayesian optimization (BO) [14], [15], digital twins can be exploited to calibrate controllers, i.e., a controller calibrated for a digital twin may perform well for the dynamical system after which the digital twin is modeled. In [16], RL is used to enhance digital twins. Other works on RL are summarized in [17], [18]. In [19], a framework for calibrating parameters using BO is proposed. In [20], BO is used to learns failure regions from simulation data. Methodologically, the proposed control approach is related to RL and BO as it uses evaluations of system performance. However, we do not rely on a trial-and-error implementation or a surrogate function. Instead, we propose to use an UKF to compute an update direction for the optimizers. While [21], [22] use an UKF for controller calibration, this paper uses an UKF to solve an optimization problem with a simulation model as differential-algebraic constraints, i.e., model for the system dynamics.

C. Notation

We define $\|x\|_\Sigma := x^T\Sigma x$, I as an identity matrix of appropriate dimension, and 0 as an all-zero vector or matrix of appropriate dimension. Further, $\mathcal{N}(\mu,\Sigma)$ is the Gaussian distribution with mean vector μ and covariance matrix Σ, and $x \sim \mathcal{N}(\mu,\Sigma)$ implies that x is sampled from $\mathcal{N}(\mu,\Sigma)$. The conditional probability density function of a vector x, conditioned on y is $p(x|y)$.

II. Digital Twin-based Optimal Control

A. Problem Statement

This section presents a framework for designing optimal control policies/laws that uses a digital twin as a predictive system model. We reiterate that such a digital twin may comprise software components that do not necessarily have closed-form analytical representations. The objective of the control policy/law is to manipulate a dynamical system,

$$0 = f_{\text{true}}(z(t),\dot{z}(t),u(t),t), \quad z(0)=z_0 \tag{1}$$

to satisfy certain specifications expressed by a cost function, $\|y_{\text{ref}} - h_{\text{true}}(z(t),u(t))\|_Q$ with $Q \succ 0$ and $Q \in \mathbf{R}^{n_h \times n_h}$, where $z(t) \in \mathbf{R}^{n_z}$ denotes the state of the dynamical system, $u(t) \in \mathbf{R}^m$ denotes the controllable input, f_{true} represents the dynamics, and h_{true} is a specification function with desired values y_{ref}. To achieve this objective, this paper uses a digital twin/simulation model of the dynamical system in (1) with

$$0 = f_{\text{twin}}(x(t),\dot{x}(t),u(t),t), \quad x(0)=x_0 \tag{2}$$

to obtain an optimal input sequence, $u(t)$, where $x(t) \in \mathbf{R}^{n_x}$ is the state of the simulation model and f_{twin} denotes a function defining a simulation environment of the dynamical system. Here, $x(t)$ may be a subset of the states $z(t)$, which is common practice in control design to reduce complexity. The simulation environment in (2) may be a simple linear system, $0 = f_{\text{twin}}(x(t),\dot{x}(t),u(t),t) = \dot{x}(t) - (Ax(t)+Bu(t))$, or a complex multibody simulation model/digital twin.

Our proposed control policy/law actuates the dynamical system in (1) using the simulation model (2) in order to optimize the surrogate cost function $\|y_{\text{ref}} - h(x(t),u(t))\|_Q$ over a prediction horizon T_{hor}, i.e.,

$$\min_{u(t)} \; \|y_{\text{ref}} - h(x(t),u(t))\|_Q \tag{3a}$$

$$\text{s.t. } 0 = f_{\text{twin}}(x(t),\dot{x}(t),u(t)) \text{ for } t \in [0,T_{\text{hor}}]. \tag{3b}$$

B. Algorithm for Digital Twin-based Optimal Control

In the following, we present an iterative algorithm to solve (3) in order to find an optimal input sequence. First, we parametrize the input sequence using parameters $\theta \in \mathbf{R}^{n_\theta}$,

$$u(t) = \phi(\theta,t) \quad \text{for } t \in [0,T_{\text{hor}}]. \tag{4}$$

Note that we do not require a functional form for ϕ, instead we treat ϕ as an oracle, where providing some θ yields a control policy.

Remark 1 (Examples for parametrization): In (4), $\phi(\theta,t)$ can be a function that uses discretized values and a zero-order hold implementation with $u(t) = \phi(\theta,t) = u_k$ for $k \leq t \leq k+T_s$ and $\theta = \begin{bmatrix} u_0 & u_{T_s} & \dots & u_{T_{\text{hor}}-T_s}\end{bmatrix}^T$; $\phi(\theta,t)$ can use discretized values and a first-order hold implementation $u(t) = \phi(\theta,t) = u_{k+T_s}\frac{t}{T_s} + u_k(1-\frac{t}{T_s})$ for $k \leq t \leq k+T_s$; $\phi(\theta,t)$ can be a linear combination of basis functions, χ, with weights θ and $u(t) = \phi(\theta,t) = \theta^T\chi(t)$; etc.

1) Digital Twin-based Optimal Control as Stochastic Estimation Problem: Let θ_i be the parameters of the algorithm at iteration i. The iterative algorithm uses

$$\theta_{i+1} = \theta_i + \Delta\theta_i, \tag{5}$$

where $\Delta\theta_i$ is the parameter update. The basic idea is to cast the optimization problem of solving for θ as an estimation problem where θ is obtained by unscented Kalman filtering. In particular, we model the parameter update and the cost function as having prior distributions with

$$\Delta\theta_i^{\text{prior}} \sim \mathcal{N}(0,R), \tag{6a}$$

$$h(x(t),u(t)) \sim \mathcal{N}(y_{\text{ref}},Q^{-1}). \tag{6b}$$

The prior on $h(x(t),u(t))$ is naturally given by the cost function in (3). The prior covariance R is a design choice for the algorithm and can be interpreted as being related to a gradient-step size. Hence, the parameters of the input sequence are obtained from the posterior distribution $p(\theta_{i+1}|\theta_0,\theta_1,\dots\theta_i,h(x(t),u(t)))$. Eq. (6b) relates to (3a) as $\mathcal{N}(y_{\text{ref}},Q^{-1})$, which is proportional to $\exp\left(-\frac{1}{2}\|y_{\text{ref}} - h(x(t),u(t))\|_Q\right)$. Thus, maximizing its logarithm, $-\frac{1}{2}\|y_{\text{ref}} - h(x(t),u(t))\|_Q$, is equivalent to minimizing the cost function in (3a).

2) Unscented Kalman Filter (UKF): The UKF uses deterministic samples—called sigma points—around the mean, which are propagated and used to update the mean and covariance estimates [2]. In the following, we use superscripts, $\text{sp}j$, to index sigma points, as opposed to the subscripts indicating iterations of the optimal control algorithm, i. The sigma points, $\theta_i^{\text{sp}j}$, are generated using the posterior distribution defined by the posterior mean, θ_i, and the posterior covariance, P_i. The sigma points, $\theta_i^{\mathrm{sp}j}$, correspond to an input sequence through (4), $u_i^{\mathrm{sp}j}(t)=\phi(\theta_i^{\mathrm{sp}j},t)$, and to a state sequence through (2), $x_i^{\mathrm{sp}j}(t)$. Hence, we can evaluate the sigma points, $\theta_i^{\mathrm{sp}j}$, through simulations and obtain

$$h_i^{\mathrm{sp}j} := h(x_i^{\mathrm{sp}j}(t), u_i^{\mathrm{sp}j}(t)), \quad u_i^{\mathrm{sp}j}(t) = \phi(\theta_i^{\mathrm{sp}j}, t). \tag{7}$$

The UKF recursion uses the evaluations of the sigma points in (7) to update its mean and covariance estimate. Then, the parameter update in (5) is given by

$$\Delta\theta_i = K_i(y_{\mathrm{ref}} - \hat{h}_i) \tag{8}$$

with the Kalman gain, K_i, and

$$\hat{\theta}_i = \textstyle\sum_{j=0}^{2n_\theta} w^{a,j}\theta_i^{\mathrm{sp}j}, \tag{9a}$$

$$\hat{h}_i = \textstyle\sum_{j=0}^{2n_\theta} w^{a,j}h_i^{\mathrm{sp}j}, \tag{9b}$$

$$S_i = Q^{-1} + \textstyle\sum_{j=0}^{2n_\theta} w^{c,j}(h_i^{\mathrm{sp}j} - \hat{h}_i)(h_i^{\mathrm{sp}j} - \hat{h}_i)^T, \tag{9c}$$

$$C_i = \textstyle\sum_{j=0}^{2n_\theta} w^{c,j}(\theta_i^{\mathrm{sp}j} - \hat{\theta})(h_i^{\mathrm{sp}j} - \hat{h}_i)^T, \tag{9d}$$

$$K_i = C_i S_i^{-1}, \tag{9e}$$

where $w^{c,i}$ and $w^{a,i}$ are weights of the sigma points, C_i is the cross-covariance matrix, and S_i is the innovation covariance. The covariance matrix for computing the sigma points is updated as

$$\tilde{P}_{i+1} = R + \textstyle\sum_{j=0}^{2n_\theta} w^{c,j}(\theta_i^{\mathrm{sp}j} - \hat{\theta})(\theta_i^{\mathrm{sp}j} - \hat{\theta})^T, \tag{9f}$$

$$P_{i+1} = \tilde{P}_{i+1} - K_i S_i K_i^T. \tag{9g}$$

In this paper, we choose

$$\begin{aligned}
\theta_i^{\mathrm{sp}0} &= \theta_i,\\
\theta_i^{\mathrm{sp}j} &= \theta_i + \sqrt{n_\theta/(1-w^0)}[A_i]^j \quad j = 1, ..., n_\theta,\\
\theta_i^{\mathrm{sp}j} &= \theta_i - \sqrt{n_\theta/(1-w^0)}[A_i]^j \quad j = n_\theta{+}1, ..., 2n_\theta,
\end{aligned}$$

with weights $w^{a,i}=w^{c,i}=(1–w^0)/(2n_\theta)$, $w^{a,0}=w^{c,0}=w^0=0$, and $[A_i]^j$ being the jth column of the square matrix A_i with $P_i = A_i A_i^T$, i.e., A_i is calculated using the Cholesky decomposition. This choice is motivated by its simplicity.

Remark 2: The UKF is agnostic to the posterior induced by $h(x(t), u(t))$, and therefore suitable to digital twins where $h(x(t), u(t))$ may be non-differentiable resulting in non-Gaussian posteriors.

3) Overall Algorithm: Algorithm 1 summarizes the proposed UKF implementation for digital twin-based optimal control at time . In each iteration of the algorithm, $2n_\theta+1$ sigma points, $\theta_i^{\mathrm{sp}j}$, are generated (Line 4). Then, for each sigma point, the corresponding input sequence is computed (Line 8), the simulation is executed to get the state sequence (Line 10), and the evaluation function is computed (Line 12). The $2n_\theta + 1$ evaluations are used in the UKF recursion (Line 15). Finally, the mean estimate is updated (Line 15). A stopping criterion based on cost convergence may also be added, similar to standard gradient descent algorithms (Line 19). Algorithm 1 relies neither on gradients nor on a strict cost decrease between iterations. This favorable property can lead to a temporary cost increase between iterations if that is necessary to reach a region with lower value of the cost function faster. Section III-A shows an example of this property. Further, the algorithm is less prone to "get stuck" in local minima compared to gradient-based optimization, by using sigma points.

Remark 3: The algorithm also optimizes the covariance matrix P_i, which defines an estimated "spread" of θ_i and provides information on the correlations of the elements in θ_i. The covariance matrix P_i speeds up the convergence of the algorithm as the sigma points are becoming more and more refined for increasing iterations.

Algorithm 1: UKF for Optimal Control

```
1  Initialize $\theta_0$, $P_0$;
2  for $i = 0, ..., i_{\max}$ do
3      %% generate sigma points;
4      $\theta_i^{\mathrm{sp}j}$ = genSigmaPoints($\theta_i, P_i$);
5      %% iterations over sigma points;
6      for $j = 0, ..., 2n_\theta$ do
7          %% get input from sigma point;
8          $u_i^{\mathrm{sp}j}(t) = \phi(\theta_i^{\mathrm{sp}j}, t)$;
9          %% simulate system using input;
10         solve $0 = f_{\mathrm{twin}}(x(t), \dot{x}(t), u_i^{\mathrm{sp}j}(t), t)$ with
             $x(0) = x_0$ for $t \in [0, T_{\mathrm{hor}}]$ to get $x_i^{\mathrm{sp}j}(t)$;
11         %% evaluate system performance;
12         compute $h_i^{\mathrm{sp}j} = h(x_i^{\mathrm{sp}j}(t), u_i^{\mathrm{sp}j}(t))$;
13     end
14     %% apply UKF recursion;
15     compute (9) to get $K_i$, $\hat{h}_i$ and $P_i$;
16     %% update optimizer;
17     $\theta_{i+1} = \theta_i + K_i(y_{\mathrm{ref}} - \hat{h}_i)$;
18     %% check stopping criterion;
19     evaluate stopping criterion;
20 end
```

III. Results

A. Swing Up of Inverted Pendulum on Cart

First, we study the control of an inverted pendulum on a cart using a digital twin, which is constructed using a multibody simulation model. Fig. 1 shows the simulation model implemented in Simscape Multibody and Table I summarizes its parameters. The input is a horizontal force applied to the cart. The objective is to swing up the pendulum to the upright position $\psi(t) = \pi$ at the cart position $p(t) = 0$ from the initial values $\psi(0) = 0$ and $p(0) = 0$. The simulation time is $T_{\mathrm{hor}} = 0.6\mathrm{s}$. The cost function $\|y_{\mathrm{ref}} - h(x(t), u(t))\|_Q$ as in (3) is chosen with $y_{\mathrm{ref}} = 0$,

$$h(x(t), u(t)) = \begin{bmatrix} h_{0\mathrm{s}} \\ h_{0.01\mathrm{s}} \\ \vdots \\ h_{0.6\mathrm{s}} \end{bmatrix}, \quad h_t = \begin{bmatrix} 10 \cdot (\psi(t) - \pi) \\ \dot{\psi}(t) \\ 100 \cdot p(t) \\ \dot{p}(t) \\ u(t) \end{bmatrix},$$

and $Q = I$. The input sequence of the horizontal force is given by a first-order hold with $u(t) = u_{k+T_s}\frac{t}{T_s} + u_k(1-\frac{t}{T_s})$ for $k \le t < k + T_s$ for all $k = 0\mathrm{s}, T_s, 2T_s, ..., 0.6\mathrm{s}$ with $T_s = 0.025\mathrm{s}$, i.e., $u(t)$ is parametrized with $\theta = [u_0 \ u_{0.025\mathrm{s}} \ \dots \ u_{0.6\mathrm{s}}]^T \in \mathbf{R}^{25}$. Further, $R = 0.01 \cdot I$ and $\theta_i = 0$.

TABLE I

PENDULUM ON A CART PARAMETERS

Group	Parameter	Value
Cart	Solid brick length Solid brick width Solid brick height	6cm 3cm 2cm
Cart friction	Breakaway friction force Breakaway friction velocity Coulomb friction force Viscous friction coefficient	0.02N 0.002m/s 0.018N 0.04N/(m/s)
Pendulum	Solid rod radius Solid rod length Solid ball radius	0.5cm 20cm 2cm
Joint friction	Breakaway friction torque Breakaway friction velocity Coulomb friction torque Viscous friction coefficient	0.01Nm 0.001rad/s 0.008Nm 0.0001Nm/(rad/s)
Density of all materials		1000kg/m^3

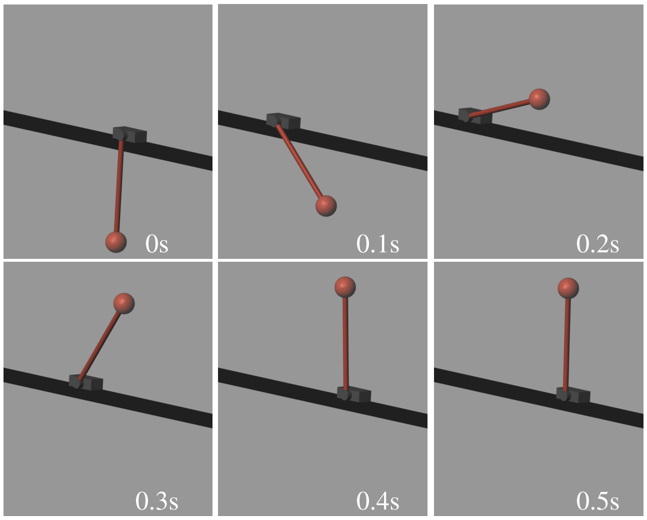

Fig. 1. Swing up of inverted pendulum on cart. The figure shows screenshots of the pendulum's motion using an input sequence computed using the optimizers at iteration 20 of the algorithm. The input sequence starts by moving the cart quickly to the left in order to gain momentum of the pendulum. Then, the input sequence waits to move back to the center position to lose momentum and move the cart underneath the pendulum.

Fig. 1 illustrates the motion of the pendulum on the cart example at iteration $i = 20$ of the optimal control algorithm. It shows that the optimal control requires only a few iterations to successfully swing up the pendulum. Fig. 2 shows the cost decrease of the optimal control algorithm. At $i = 8$, a sharp increase in cost can be observed. This increase is caused by the nature of the UKF algorithm that does not rely on gradients and does not provide/need a strict cost decrease between iterations. Instead, the gradient-free nature allows the algorithm to "move through" regions with higher cost to reach regions with smaller cost. In this particular case, the algorithm overshoots the upright position, which causes the pendulum to continue the rotation. However, at iteration $i = 9$, the algorithm found an input sequence that slows down the rotation after having reached the upright position. Fig. 2 also shows the angle of the pendulum and the cart position over time at the final iteration $i = 100$. Around $i = 80$, the optimal controller has found an input sequence that keeps the pendulum relatively steady around the unstable equilibrium, see only minor corrections of the cart position for $t > 0.3$s.

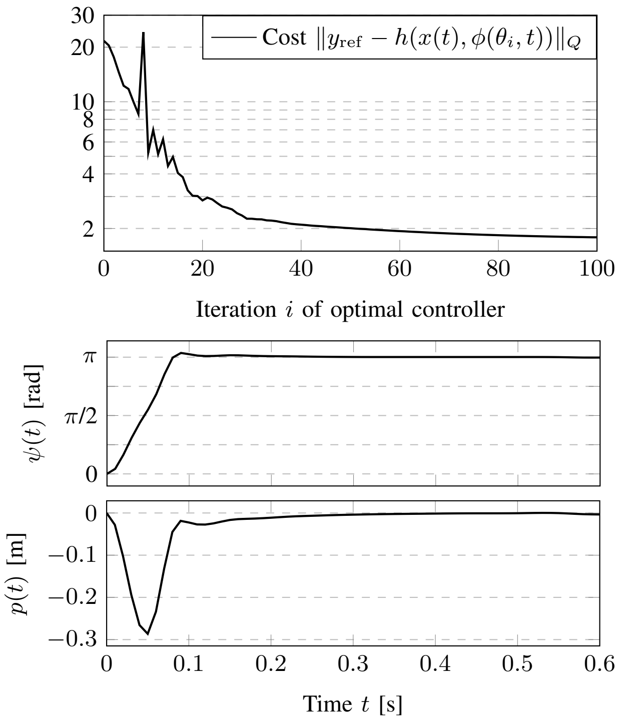

Fig. 2. Results of pendulum on a cart study. Top: Cost decrease of optimal controller as function of the algorithm's iterations. Middle: Angle of pendulum for iteration $i = 100$. At time $t = 0$s, the pendulum is in its stable equilibrium position (hanging down). At around time $t = 0.3$s, the pendulum reached its target state (upright position). Bottom: Cart position for iteration $i = 100$. While the cart at time $t = 0$s is initially at its target position, it quickly moves to a position of around $p(t) = -0.3$m in order to gain momentum and swing up the pendulum.

B. Oscillations of Crane Load

Second, we study the control of a crane as illustrated in Fig 3. The Simscape simulation model used as digital twin illustrated in Fig 4 is open access and provided by Mathworks [23], i.e., all simulation parameters and dimensions can be accessed. The crane is controlled through a hoist drum, a trolley drum, and a jib. The hoist drum is modeled to roll up/unroll a wire rope in order to raise and lower a load. The trolley drum moves the load towards and away from the crane tower. The jib moves the load horizontally by rotating around the crane tower. The inputs to the crane are the three position commands of the revolute joints for the hoist drum, the trolley drum, and the jib. The position commands are tracked by an internal controller, i.e., the crane provided in [23] is a closed-loop model with a pre-stabilizing position controller. Hence, here we show that the proposed digital twin-based optimal control can use a closed-loop plant to compute position commands. This example illustrates that the UKF is agnostic to the posterior induced by the simulations and that the control map, ϕ, can be unknown.

The goal of the optimal controller is to minimize the oscillations of the crane's load in reference to the jib. Thus,

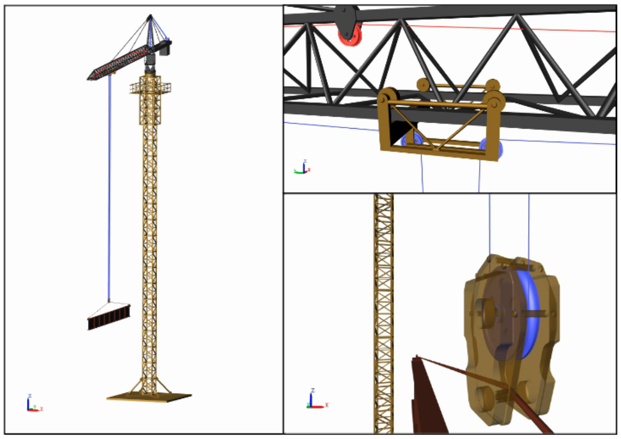

Fig. 3. Digital twin visualization of a crane (open access model, provided by Mathworks [23]).

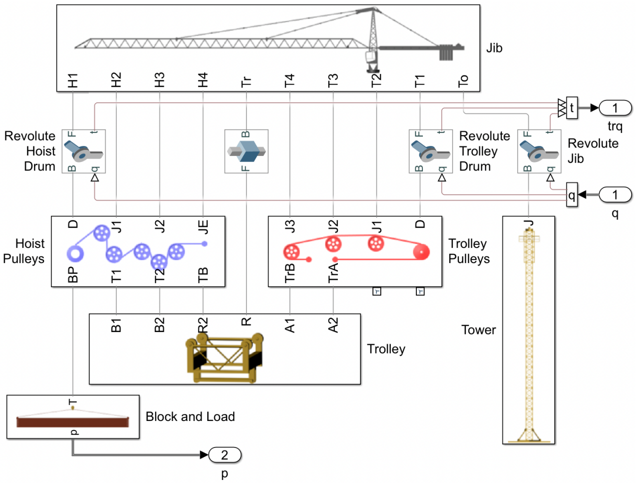

Fig. 4. Simscape model. The simulation model is used in order to determine the three optimal input sequences to the system, given by the three position commands of the hoist drum, the trolley drum, and the jib. The goal of the optimal controller is to minimize oscillations of the load.

the cost function $\|y_{\text{ref}} - h(x(t),u(t))\|_Q$ in (3) is defined as

$$h(x(t),u(t)) = \begin{bmatrix} h_{0\text{s}} \\ h_{0.1\text{s}} \\ \vdots \\ h_{54.9\text{s}} \\ h_{55\text{s}} \end{bmatrix} \text{ with } h_t = \begin{bmatrix} \Delta p_x(t) \\ \Delta p_y(t) \\ 100 \cdot \Delta\omega_x(t) \\ 100 \cdot \Delta\omega_y(t) \\ 100 \cdot \Delta\omega_z(t) \end{bmatrix},$$

and $y_{\text{ref}} = 0$, $Q = I$, and $T_{\text{hor}} = 55\text{s}$. Here, $\Delta p_x(t)$ and $\Delta p_y(t)$ are the relative positions of the load measured in the coordinate frame that moves with the trolley attached to the jib, i.e., $\Delta p_x(t) = \Delta p_y(t) = 0$ indicates that the center of gravity of the load is straight below the trolley. Further, $\Delta\omega_x(t)$, $\Delta\omega_y(t)$, and $\Delta\omega_z(t)$ are the relative rotations of the load measured in the coordinate frame that moves with the trolley attached to the jib, i.e., $\Delta\omega_x(t) = \Delta\omega_y(t) = \Delta\omega_z = 0$ indicates that the load rotates only due to the jib rotation.

The input commands of the three actuators are computed using a first-order hold with sampling period $T_s = 5\text{s}$, where

$$V^{\text{jib}} = \begin{bmatrix} v_{5\text{s}}^{\text{jib}} \\ v_{10\text{s}}^{\text{jib}} \\ \vdots \\ v_{50\text{s}}^{\text{jib}} \end{bmatrix} V^{\text{hoist}} = \begin{bmatrix} v_{5\text{s}}^{\text{hoist}} \\ v_{10\text{s}}^{\text{hoist}} \\ \vdots \\ v_{50\text{s}}^{\text{hoist}} \end{bmatrix} V^{\text{trolley}} = \begin{bmatrix} v_{5\text{s}}^{\text{trolley}} \\ v_{10\text{s}}^{\text{trolley}} \\ \vdots \\ v_{50\text{s}}^{\text{trolley}} \end{bmatrix},$$

and hence,

$$\theta = \begin{bmatrix} V^{\text{jib}} \\ V^{\text{hoist}} \\ V^{\text{trolley}} \end{bmatrix} \in \mathbf{R}^{30}.$$

The initial input command values $v_{0\text{s}}^{\text{jib}} = 90$, $v_{0\text{s}}^{\text{hoist}} = 0$, and $v_{0\text{s}}^{\text{trolley}} = -25$, and the terminal values $v_{55\text{s}}^{\text{jib}} = 150$, $v_{55\text{s}}^{\text{hoist}} = -60$, and $v_{55\text{s}}^{\text{trolley}} = -50$ are fixed. In other words, the crane is moved from a position defined by $v_{0\text{s}}$ to a position defined by $v_{55\text{s}}$, while minimizing oscillations. All velocity and angular velocity states are initialized to be zero at time $t = 0\text{s}$. The parameters, θ_0, are initialized to linearly interpolate between $v_{0\text{s}}$ and $v_{55\text{s}}$, and we choose $R = 0.01 \cdot I$.

Fig. 5 shows the cost decrease of the optimal control algorithm from iterations $i = 0$ to $i = 150$. It shows that the algorithm can quickly reduce the cost due to the oscillations of the beam. E.g., at iteration $i = 32$, the cost decreased to below 20 from an initial cost of 109. At iteration $i = 150$, the cost decreased to 14.8. Fig. 5 also illustrates a comparison between the oscillations before and after optimizing using the digital twin. The relative rotational velocities of the beam are reduced to remain within 0.5rad/s for all rotation axes, whereas the relative rotational velocities for the initial inputs using the linear position commands have higher frequency and exceed amplitudes of 1rad/s. The position errors for the optimized input sequences remain within 0.2m from the reference point of the jib, whereas the linear position commands yield higher oscillations and larger deviations from the reference point, see amplitude of -0.38m at 7s.

IV. Extended Discussion and Conclusion

We proposed to use a digital twin as implicit model of the system dynamics. The approach is based on propagating and evaluating sigma points of an UKF rather than using gradients to iteratively update the control policy. Thus, this approach is generally useful when propagating/simulating a dynamical system model or solving differential-algebraic constraints is easier than computing gradients. Further, as the approach does not require the explicit computation of gradients, it is applicable to cost functions that are nondifferentiable or with a motion constraint defined by nondifferentiable functions such as for switched systems. Advantages of the proposed control approach include that neither the control map nor the digital twin need to be known explicitly and that the digital twin approximates the physical dynamical system "more closely" than a control-oriented model, and thus improves the control performance.

Open problems associated with this methodology can be the computational burden that requires simulations to update the optimizer, which may be costly depending on

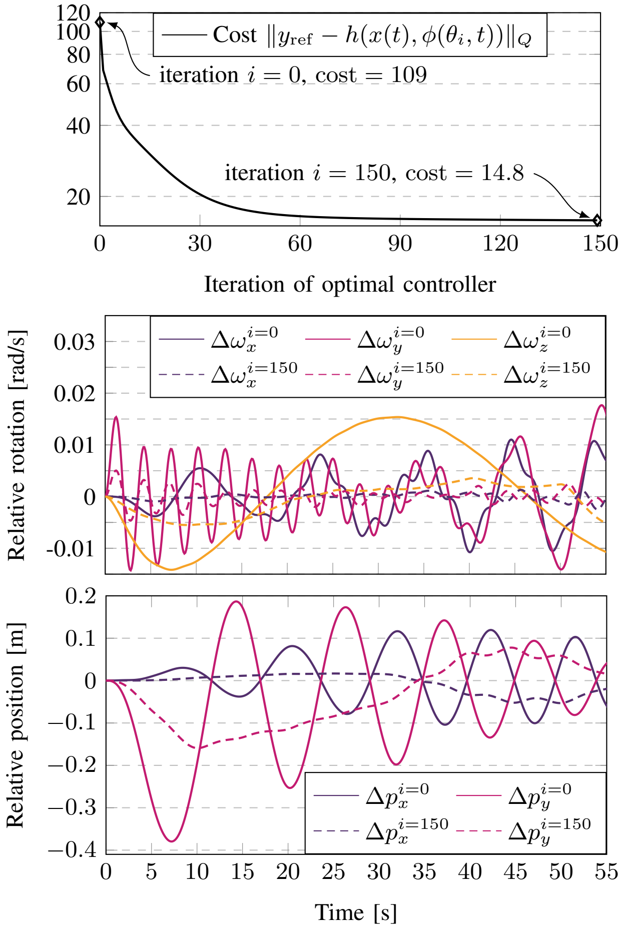

Fig. 5. Results of crane controller using a digital twin. Top: Cost decrease over iterations of optimal controller. Middle: Load rotations in reference to jib. At iteration $i = 150$, the rotations (dashed purple, magenta, and yellow) are minimized with values less than 0.005rad/s, whereas at iteration $i = 0$, rotations (solid purple, magenta, and yellow) are of higher frequency and higher amplitude. Bottom: Position error in reference to jib.

the fidelity of the simulation environment. However, future work may focus, e.g., on parallelizing the simulations within an iteration of the UKF-based optimal control algorithm. Alternatively or additionally, a smart parametrization of the input sequence, e.g., using a number of basis functions rather than zero/first-order hold with discretized values, can reduce the computation time of the algorithm. Finally, warm-starting the algorithm, i.e., initalizing the optimizer θ_0 and its covariance P_0, can significantly reduce the iterations needed, e.g., as in [24] using Bayesian principles.

This paper presented initial findings and two preliminary case studies, where digital twins were constructed and utilized to compute optimal input sequences. The case studies showed that the algorithm can control an unstable dynamical system with high precision and can effectively improve the operation of a complex dynamical system. Future work could aim at alleviating the computational burden of the proposed control algorithm, investigate theoretical properties and systematic design choices of the algorithm, and on alternative algorithms to UKF.

References

[1] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*, vol. 26. Birkhäuser, 2012.

[2] G. A. Terejanu, "Unscented Kalman filter tutorial," *University at Buffalo, Buffalo*, 2011.

[3] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21980–22012, 2020.

[4] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[5] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE access*, vol. 8, pp. 108952–108971, 2020.

[6] J. Sun and A. Reddy, "Optimal control of building HVAC&R systems using complete simulation-based sequential quadratic programming (CSB-SQP)," *Building and Environ.*, vol. 40, no. 5, pp. 657–669, 2005.

[7] S. Baldi, I. Michailidis, V. Ntampasi, E. Kosmatopoulos, I. Papamichail, and M. Papageorgiou, "A simulation-based traffic signal control for congested urban traffic networks," *Transportation Science*, vol. 53, no. 1, pp. 6–20, 2019.

[8] E. O'Dwyer, I. Pan, R. Charlesworth, S. Butler, and N. Shah, "Integration of an energy management tool and digital twin for coordination and control of multi-vector smart energy systems," *Sustainable Cities and Society*, vol. 62, p. 102412, 2020.

[9] A. Papacharalampopoulos and P. Stavropoulos, "Towards a digital twin for thermal processes: Control-centric approach," *Procedia CIRP*, vol. 86, pp. 110–115, 2019.

[10] T. Wang, J. Cheng, Y. Yang, C. Esposito, H. Snoussi, and F. Tao, "Adaptive optimization method in digital twin conveyor systems via range-inspection control," *IEEE Trans. on Automation Science and Engineering*, 2020.

[11] K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, and R. Harik, "A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence," *J. of Manufacturing Systems*, vol. 58, pp. 210–230, 2021.

[12] S. Y. Barykin, A. A. Bochkarev, E. Dobronravin, and S. M. Sergeev, "The place and role of digital twin in supply chain management," *Academy of Strategic Management Journal*, vol. 20, pp. 1–19, 2021.

[13] D. Bertsekas, *Reinforcement learning and optimal control.* Athena Scientific, 2019.

[14] J. Mockus, "Application of Bayesian approach to numerical methods of global and stochastic optimization," *J. of Global Optimization*, vol. 4, no. 4, pp. 347–365, 1994.

[15] P. I. Frazier, "A tutorial on Bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[16] C. Cronrath, A. R. Aderiani, and B. Lennartson, "Enhancing digital twins through reinforcement learning," in *2019 IEEE 15th Int. Conf. on Automation Science and Engineering (CASE)*, pp. 293–298, 2019.

[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[18] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.

[19] A. Chakrabarty, E. Maddalena, H. Qiao, and C. Laughman, "Scalable Bayesian optimization for model calibration: Case study on coupled building and HVAC dynamics," *Energy and Buildings*, vol. 253, p. 111460, 2021.

[20] A. Chakrabarty, S. A. Bortoff, and C. R. Laughman, "Simulation failure robust Bayesian optimization for estimating black-box model parameters," in *2021 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, pp. 1533–1538, 2021.

[21] M. Menner, K. Berntorp, and S. Di Cairano, "A Kalman filter for online calibration of optimal controllers," in *2021 IEEE Conf. on Control Technology and Applications (CCTA)*, pp. 441–446, 2021.

[22] M. Menner, K. Berntorp, and S. Di Cairano, "Automated controller calibration by Kalman filtering," *arXiv preprint arXiv:2111.10832*, 2021.

[23] Mathworks, "Tower crane with trolley and hoist." mathworks.com/help/physmod/sm/ug/tower-crane-with-trolley-and-hoist.html, accessed: 2022-01-15.

[24] K. Berntorp, A. Chakrabarty, and S. Di Cairano, "Vehicle center-of-gravity height and dynamics estimation with uncertainty quantification by marginalized particle filter," in *2021 American Control Conf. (ACC)*, pp. 160–165, 2021.