A Control Approach to Fault Disambiguation

Ion Matei[1], Maksym Zhenirovsky[2], Johan de Kleer[3], and Kai Goebel[4]

[1,2,3,4] *Palo Alto Research Center, Palo Alto, CA, 94304, USA*
imatei@parc,com
mazhenir@parc.com
dekleer@parc.com
kgoebel@parc.com

Abstract

To ensure resilience, systems must be endowed with capabilities for rapid detection, response, and recovery to disruptive events. In this paper we focus on faults as disruptive events and use a diagnosis engine for their detection and isolation. In particular we use model-based diagnosis, where the diagnosis engine is provided with a model of the system, nominal values of the parameters of the model and values of some of its inputs and outputs. However, there is no guarantee that the information measured by sensors is sufficient to distinguish between multiple root-causes. We address this challenge using an optimal control approach: we design control inputs such that the similarity between outputs in ambiguous fault modes is reduced. We show that under certain assumptions on the system model, minimizing a similarity metric in terms of outputs is equivalent to increasing the diagnosis certainty. We use an optimization-based approach to input design, where the system model acts as a constraint. We show that by using a surrogate model expressed with constructs endowed with differential operators, we improve the time efficiency of the optimal control problem. We demonstrate our approach on a fuel line system, where feedback control is used to ensure the mass flow rate at the engine follows a prescribed reference. We consider leak faults that affect the fuel lines. We show that under the control inputs generated by the nominal controller, mass flow rate measurements are not enough to accurately isolate leaks. We demonstrate that by using custom inputs that minimize the similarity between the outputs in the ambiguous fault modes, the diagnosis uncertainty is eliminated.

1. Introduction

We address the problem of minimizing the uncertainty of fault diagnosis of closed-loop systems. This uncertainty can be cause by: (i) not enough sensors, (ii) the system operating in steady state, or (iii) fast acting controllers that obscure the fault effects. Broadly speaking there are two approaches to the diagnosis problem: model-based diagnosis MBD and Machine Learning (ML). MBD methods use models and parameters while ML approaches require training data and extensive feature engineering. In this paper we use MBD, where a diagnosis engine is provided with a model of the system, nominal values of the parameters of the model and values of some of its inputs and outputs. Using only this information, the diagnosis must detect the presence of a fault and isolate it. MBD has a long history, with results developed independently by the artificial intelligence (de Kleer, Mackworth, & Reiter, 1992) and control (Gertler, 1998),(Isermann, 2005),(Patton, Frank, & Clark, 2000) fields. Traditional model-based diagnosis approaches in the control communities include filters (e.g., Kalman filter (Kalman, 1960), particle filter (Arulampalam, Maskell, & Gordon, 2002)), or optimization based-techniques that estimate parameters whose deviation from their nominal values indicate the presence of a fault. These methods rely on model simulations either for one sample period (Kalman and particle filters) or for some time horizon (optimization based). In this paper we consider an optimization-based approach to fault diagnosis, where sensor measurements and control inputs over some time horizon, together with the system model are used to estimate the values of a set of parameters associated with system faults. The optimization algorithm searches for parameter values so that the system model produces outputs that match the measurements. Deviations of the fault parameters from their nominal values indicate the presence of faults. There is no guarantee that the diagnosis solution is certain, i.e., there can be several explanations for the root cause of a detected fault. This paper proposes an approach for minimizing the ambiguity in the diagnosis solution. The main contributions are as follows: (i) we employ a diagnosis uncertainty metric that when evaluated indicates the magnitude of diagnosis ambiguity (or uncertainty), (ii) given a list of possible diagnoses, we use an optimal control approach to design control inputs aimed at reducing the diagnosis uncertainty; (iii) we address the scalability of the control design to real-time implementation by using a surrogate model of the physics-based model expressed in constructs endowed with automatic differentiation (AD) support.

Notations: We use upper-case to denote random variables (X) and lower-case to denote a realization of a random variable (x). We use bold letters to denote vectors ($\boldsymbol{x}$). We mark the continuous time dependency by using the notation $X(t)$ and $x(t)$ for random processes and time-varying variables, respectively. To represent discrete time dependency, we use the notation $x(t_k) = x_k$, for time instants t_k. A sequence of variables over time $\{x_k\}_{k=0}^{K}$ is denoted by $x_{0:K}$. We denote the probability distribution function (p.d.f.) of a random variable X by $f_X(x)$. We represent the conditional p.d.f. of $X|Y$ by $f_{X|Y}(x|y)$. When there is no loss of clarity, to simplify the notation, we will omit the subscript notation of $f_{X|Y}(x|y)$, that is, we will use $f(x|y)$. We denote the expectation of the random variable X by $\mathbb{E}[X]$. Let $\mathcal{S} = \{s_i\}_{i=1}^{n}$ denote a set of elements. We denote by s_{-i} the set $\mathcal{S} - \{s_i\}$. We denote by $\mathcal{N}(\mu, \Sigma)$ the multivariate Gaussian distribution with mean μ and covariance matrix Σ.

Paper structure: In Section 2 we discuss describe the system model and introduce the diagnosis problem. In Section 3 we present our approach to fault disambiguation and describe a method for improving the scalability of the diagnosis algorithm to real time implementation. We showcase our approach to reducing the uncertainty in diagnosis solution through diagnosis results before and after applying the disambiguation inputs, in Section 4.

2. PROBLEM STATEMENT

We consider parametric faults, i.e., each fault mode has an associated scalar parameter. We define $\mathcal{F} = \{p_1, p_2, \ldots, p_N\}$ as the set of fault parameters, for some integer number N, and we assume each fault parameter p_i has a nominal value $\bar{p}_i$. In nominal conditions, the parameter vectors remain close to the nominal values $\bar{p}_i$.

The diagnosis engine uses a physics-based model to reason about the health of the system. The type of systems can include nonlinearities, discrete and algebraic constraints. The mathematical model describing the behavior of the physical system is given by a (hybrid) differential algebraic equation (DAE) of the form

$$\begin{aligned} 0 &= F(\dot{\boldsymbol{X}}, \boldsymbol{X}, \boldsymbol{U}; \boldsymbol{P}),\ \boldsymbol{X}(0) = \boldsymbol{X}_0 \qquad (1)\\ \boldsymbol{Y} &= h(\boldsymbol{X}, \boldsymbol{U}; \boldsymbol{P}) + \boldsymbol{V}, \qquad (2)\end{aligned}$$

where $\boldsymbol{X}$ is the (stochastic) state of the system, $\boldsymbol{U}$ is the vector of inputs, $\boldsymbol{P}$ is the vector of model parameters, and $\boldsymbol{Y}$ is the vector of output measurements. The outputs are affected by the independent and identically distributed (i.i.d.), additive noise $\boldsymbol{V}$, assumed Gaussian with zero mean and covariance matrix Σ_v. The initial state $\boldsymbol{X}_0$ and the vector of parameters are vector-valued random variables with known prior distribution $f_{\boldsymbol{X}_0}$ and $f_{\boldsymbol{P}}$, respectively. For example, the p.d.f. of the vector of parameters can be Gaussian, with mean $\bar{\boldsymbol{p}}$ and covariance matrix Σ_p. The vector $\bar{\boldsymbol{p}}$ can be interpreted as the nominal value of the vector of system parameters, and matrix Σ_p reflects the uncertainty in the nominal value.

We use the single fault scenario, i.e., no two faults became active at the same time. The fault event is defined by $\{|p_i - \bar{p}_i| > \varepsilon_i, p_{-i} = \bar{p}_{-i}\}$, where ε_i is a positive scalar. The scalar ε_i depends on the measurement noise and the sensitivity of the behavior of the system to changes in parameter p_i. The fault magnitude is determined by estimating the value of the system parameter p_i. Given a sequence of input and output measurements over the time horizon τ, the diagnosis problem consists of computing the conditional probability $\mathbb{P}(|p_i - \bar{p}_i| > \varepsilon_i | \boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau})$, for all i together with the estimation of the parameter p_i. An ambiguous diagnosis appears when there exist a least two faults i and j so that $\mathbb{P}(|p_i - \bar{p}_i| > \varepsilon_i | \boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau}) \approx \mathbb{P}(|p_j - \bar{p}_j| > \varepsilon_j | \boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau})$, meaning that their probability is roughly the same, impeding a clear decision on what fault is the root cause of the observed anomalous behavior.

To illustrate our approach to fault disambiguation, we constructed a fuel system model shown in Figure 1, expressed in the Modelica (Fritzson, 2015) language. The model describes the fuel supply from a fuel tank to an engine through a series of pipes. The model uses components from the `Modelica.Thermal.FluidHeatFlow` library. We model the fuel tank using the `OpenTank` component from the `Modelica.Thermal.FluidHeatFlow` library, while the engine is modeled using a `sink` component from the same library. We model the interconnections between the tank and the engine using a series of pipes. The pipes can be affected by leaks, by augmenting the nominal model with specialized `leak` components. We successfully demonstrated this approach to fault diagnosis in our previous work (Saha & et al., 2014; Minhas et al., 2014). The leaks are modeled through specialized components that use (fault) parameters to set the severity of the faults. A 0 value means there is no leak, while a 1 value means the highest leak severity. We have included 4 leak points on the fuel line. The leak components are implemented using a valve connected to a sink component that sets the ambient pressure. The fault parameter sets the opening of the valve, hence the fault diagnosis algorithm tracks the leak parameters to detect and isolate leaks.

3. APPROACH

In this section we describe the MBD diagnosis algorithm, the control design algorithm for fault disambiguation and present

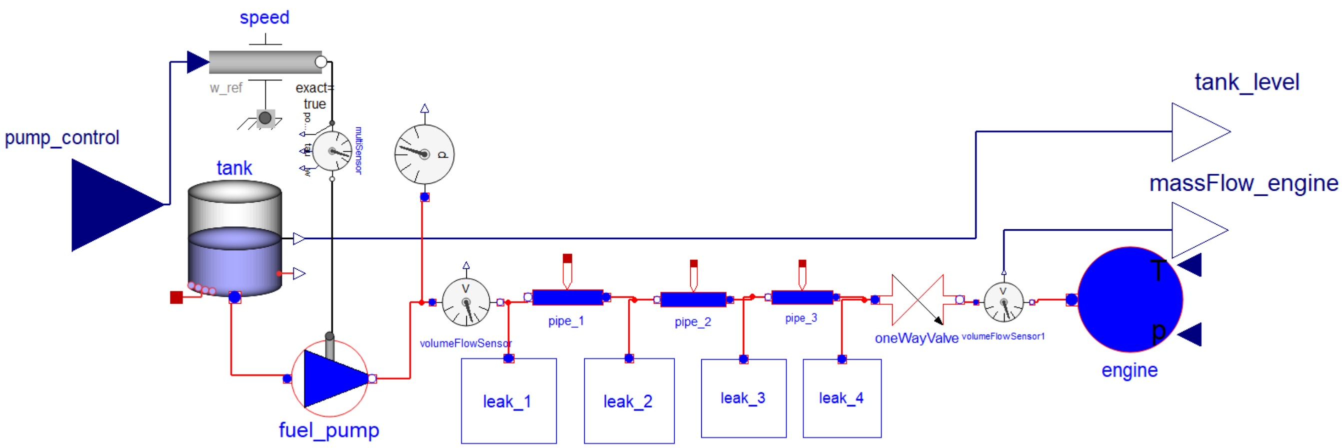

Figure 1. Modelica model of the fuel system.

an approach for improving the time efficiency of the control design.

3.1. Optimization-based diagnosis

We use an optimization-based approach to estimate the fault parameters and evaluate the fault probabilities. The optimization algorithm uses model simulations to update the fault parameters so that the simulated outputs match the measured outputs. We segment the output time series into non-overlapping windows, and for each window we update the fault parameters by solving least square problems. The choice of the size of the window depends on the time constants of the system dynamics. In the case of the fuel system, the transients are negligible, i.e., the measurements correspond to the steady state regime. Hence the window size can be relatively small. However, the size of the window must be correlated with the time needed by the optimization algorithm to generate a solution. If such a time is larger than the window size, we incur delays in generating diagnosis solutions. The least square optimization algorithm is expressed as a nonlinear program that optimizes for continuous fault parameters, i.e., the leak magnitudes. As an alternative to the optimization-based approach, we could use filtering-based techniques, by considering the fault parameters as states. Since the fuel system model is nonlinear, the linear Kalman filter cannot be applied directly. The extended Kalman filter (McElhoe, 1966) requires artifacts not readily available such as the Jacobians of the state and measurements maps. Sampling-based filtering techniques, such as the particle filter (Arulampalam et al., 2002), are computationally intensive since they require many sample points to propagate an accurate (possibly non-Gaussian) distribution of the state. The unscented Kalman filter (Julier & Uhlmann, 1997) is a compromise between accuracy and computational effort and uses a set of sigma-points to approximate the distribution of the state. Given that in the case of the fuel system the transients are negligible and the outputs do not depend on the initial states, we chose to use the optimization-based approach. Qualitative diagnosis algorithms, such as analytical redundant relations (ARRs) (Staroswiecki, 2000; Staroswiecki & Comtet-Varga, 2001) can also be applied. They have the advantage that they do not require fault models, but they typically need more sensors to generate unambiguous diagnosis solution. Since in the case of the fuel system the physics behind leaks is well understood, we opted for a diagnosis solution based on fault models. For each fault parameter, the diagnosis engine solves an optimization problem using parallel processes.

For the cases when the transients are very fast and when the steady state does not depend on the initial conditions, the system model (1)-(2) in its discrete form becomes a memoryless mathematical model:

$$\boldsymbol{Y}_k = h(\boldsymbol{U}_k; \boldsymbol{P}) + \boldsymbol{V}_k, \tag{3}$$

where we use upper cases to emphasize the stochastic nature of the system model.

Given a sequence of output measurements $\{\boldsymbol{y}\}_{k=0}^{\tau}$ and inputs $\{\boldsymbol{u}\}_{k=0}^{\tau}$ the diagnosis engine computes the p.d.f. $f(p_i|\boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau})$, under the assumption that no other fault becomes active at the same time with fault i. The optimal estimate of the fault parameter is given by $\hat{p}_i = \mathbb{E}[\boldsymbol{P}|\boldsymbol{Y}_{0:\tau} = \boldsymbol{y}_{0:\tau}, \boldsymbol{U}_{0:\tau} = \boldsymbol{u}_{0:\tau}]$. The maximum likelihood estimator is the solution of the optimization problem $\max_{p_i} f(p_i|\boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau})$, whose solution is $\hat{p}_{i_\tau}$.

Using a Bayesian approach, $f(p_i|\boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau})$ can be expressed as

$$f(p_i|\boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau}) = \frac{\prod_{k=0}^{\tau} f(\boldsymbol{y}_k|\boldsymbol{u}_k, p_i) f(p_i)}{\int \prod_{k=0}^{\tau} f(\boldsymbol{y}_k|\boldsymbol{u}_k, p_i) f(p_i) dp_i},$$

where $f(p_i)$ is the prior distribution of the fault parameter p_i.

Under the input-output dynamics (3), the conditional p.d.f.

$f(\boldsymbol{y}_k|\boldsymbol{u}_k, p_i)$ is a Gaussian p.d.f. with mean $\hat{\boldsymbol{y}}_k^i = h(\boldsymbol{u}_k; p_i)$ and covariance matrix Σ_v. The mean vector $\hat{\boldsymbol{y}}_k^i$ is generated by simulating the system model, given the input $\boldsymbol{u}_k$ and the vector of parameters p_i, while all other fault parameters are kept at their nominal values. Under the additive, Gaussian measurement noise assumption, we compute the solution of the maximum likelihood estimator by solving the following optimization problem:

$$\min_{p_i} \sum_{k=0}^{\tau} \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^i\right)^T \Sigma_v^{-1} \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^i\right) \tag{4}$$

The optimization problem above computes the most like fault magnitude $\hat{p}_i$, under the assumption that fault i is active.

We solve N such optimization problems in parallel and obtain a set of parameters $\{\hat{p}_i\}_{i=1}^N$. To decide which fault is active, we compute the empirical probabilities:

$$q_i \propto \mathcal{N}(\hat{\boldsymbol{y}}_k^i, \Sigma_v),$$

where $\hat{\boldsymbol{y}}_k^i$ are generated by simulating the model with the fault parameter $\hat{p}_i$, while setting the remaining fault parameters to their nominal values, i.e., $\bar{p}_j$ for $j \neq i$.

Ideally, we would like to have the probabilities q_i so that it is clear what fault is active, i.e., if fault i is active q_i should be close to 1. However, this may not always be possible and multiple probabilities q_i can be similar. In the following section, we describe our approach to fault disambiguation.

3.2. Fault disambiguation

Figure 2 depicts the steps for addressing ambiguous diagnoses. We assume that a physical system is controlled by a vector of exogenous inputs u_c and generates measured outputs denoted by y. A diagnosis engine receives the control inputs and the output measurements and generates an initial diagnosis. Upon the evaluation of its uncertainty, a decision is made if the diagnosis is final or is rejected if it is ambiguous. If the latter, control inputs u_d are designed to reduce the diagnosis uncertainty and applied to the system to generate new output measurements. These new measurements together with the disambiguation inputs are used the produce a new diagnosis.

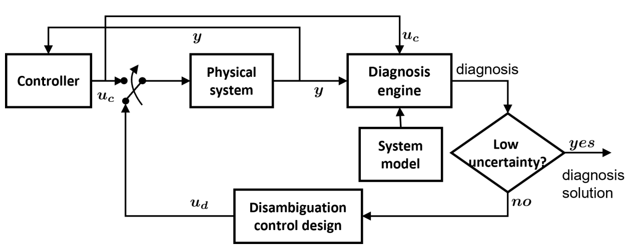

Figure 2. Block diagram of the approach for dealing with ambiguous diagnosis.

Let q_i and associated $\hat{p}_i$ be the fault probabilities and the fault parameter estimates, respectively. We can measure the uncertainty of the diagnosis solution using the Shannon entropy: $H = -\sum_{i=1}^N q_i \log(q_i)$. The worse case, i.e., the highest uncertainty happens when $q = 1/N$ for all i. The best case, i.e., the lowest uncertainty is when there exists an i such that $q_i = 1$, in which case $H = 0$. We can set a degree of acceptable uncertainty by choosing a scalar $\delta \in [0, \log(N)]$ and decide that a diagnosis solution is ambiguous if $H \leq \delta$.

Our objective is to design a sequence of inputs $\{\boldsymbol{u}_k\}_{k=0}^{\tau}$ that maximizes the Euclidean distance between the system outputs that correspond to the ambiguous fault modes. We achieve this objective by solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{u}_0,\ldots,\boldsymbol{u}_\tau} \quad & -\sum_{i>j}\sum_{k=0}^{\tau} \|\hat{\boldsymbol{y}}_k^i - \hat{\boldsymbol{y}}_k^j\|^2 \\ \text{subject to:} \quad & \hat{\boldsymbol{y}}_k^i = h(\boldsymbol{u}_k; \hat{p}_i), \forall i \\ & \boldsymbol{u}_k \in \mathcal{U}, \forall k \in \{0, \ldots, \tau\}, \end{aligned} \tag{5}$$

where $\mathcal{U}$ is a set constraining the inputs that can be applied to the system. The definition of the loss function is not arbitrary. We recall that the key term for computing $f(p_i|\boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau})$ is the product of conditional p.d.f.s $\prod_{k=0}^{\tau} f(\boldsymbol{y}_k|\boldsymbol{u}_k, p_i)$. To evaluate each conditional p.d.f. $f(\boldsymbol{y}_k|\boldsymbol{u}_k, p_i)$, we need to evaluate the quadratic term $\left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^i\right)^T \Sigma_v^{-1} \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^i\right)$. Let i be the true fault mode, with $\boldsymbol{y}_k^i = \hat{\boldsymbol{y}}_k^i + \boldsymbol{v}_k$ the output measurements expressed in term of the simulated output $\hat{\boldsymbol{y}}_k^i$ and a realization of the measurement noise $\boldsymbol{v}_k$. For each ambiguous fault mode j, the quadratic expression $\left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^j\right)^T \Sigma_v^{-1} \left(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^j\right)$ becomes $\left(\hat{\boldsymbol{y}}_k^i + \boldsymbol{v}_k - \hat{\boldsymbol{y}}_k^j\right)^T \Sigma_v^{-1} \left(\hat{\boldsymbol{y}}_k^i + \boldsymbol{v}_k - \hat{\boldsymbol{y}}_k^j\right)$. For the ground truth case $i = j$, the previous expression becomes $\boldsymbol{v}_k^T \Sigma_v^{-1} \boldsymbol{v}_k$, which is the smallest quantity that we can get under the additive noise assumption. Hence by solving (5), we in fact maximize the conditional p.d.f. $f(p_i|\boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau})$ for the ground truth fault mode i, while in all the other fault modes j, $f(p_j|\boldsymbol{y}_{0:\tau}, \boldsymbol{u}_{0:\tau})$ becomes smaller.

The disambiguation problem minimizes the diagnosis uncertainty defined by the Shannon entropy. Indeed, solving (5) brings the probabilities q_i closer to zero or one, hence minimizing the entropy.

Remark 1: In Figure 2, we separate the inputs generated by controller from the disambiguation inputs. Alternatively, the disambiguation inputs u_d can be superimposed on the controller inputs, i.e., $\tilde{u}_c = u_c + u_d$, where u_d can be designed to minimize both the effects on the system stability and the similarity among the outputs in ambiguous fault modes. Such an idea was pursued in (Kuhn, Price, de Kleer, Do, & Zhou, 2008), where a pervasive diagnosis approach based on AI planning was designed to simultaneously achieve production goals while uncovering additional information about component health. For system with a continuous action space, using planning is computationally expensive due to combinatorial explosion. Thus, a model predictive control approach is more appropriate.

Remark 2: In the optimization problem formulation, we considered all possible faults that in turn requires model simulations for all fault scenarios. We can reduce the complexity of the optimization problem by selecting the top K most likely fault modes and solving the disambiguation optimization problem in terms of them, only.

3.3. Improving the scalability to real time implementation

Solving (5) requires simulating the model for each fault scenario to generate the predicted outputs $\hat{\boldsymbol{y}}_k^i$, as the optimization algorithm searches for the best control inputs. Such a problem can be viewed as a black-box optimization since we can not analytically compute the gradients of the loss function; gradients that depend on the system variables. The black-box optimization limits the access to gradient-free optimization algorithms (e.g., Bayesian optimization, Powell, Simplex). Typically, black-box optimization algorithms scale poorly with the number of optimization variables.

We can improve the time efficiency of the optimization algorithms by using gradient-based methods and providing the algorithm access to the gradients of the loss function. Unlike gradient free algorithms, gradient-based methods scale linearly with the number of optimization variables. We achieve this goal by employing *surrogate models* expressed using constructs endowed with AD. Such models are emulators that mimic the behavior of the original model while enabling the evaluation of gradients of loss functions. We learn surrogate models using a data-driven approach, namely we use the deep learning platform Pytorch (Paszke & et al., 2017). We designed and trained a neural network (NN) based surrogate model for the physics-based model. The NN has as a total of 5 inputs: the control signal to the pump and the leak fault parameters. The outputs are the tank level and the mass flow rate to the engine. The NN has four layer: one `GRU` layer, followed by a dense layer, a `ReLU` layer and a dense layer as output. All hidden layers have the same size, namely 128. The training data is generated by simulating the physics-based model. For a large number of inputs and outputs, we have a combinatorial explosion in the size of the training data. However, the training data generation is done off-line, hence time constraints are less relevant. We used the physics-based model to generate data for training the surrogate model. We generated approximately 300k training data samples. The model simulations were done using the FMU (Blochwitz et al., 2011) representation of the physics-based model, integrated into Python code. The input consists of pump control and leak parameters, and the outputs are represented by mass flow rates and the tank level. The inputs were randomly drawn from their domain of definition, using the uniform distribution. Since the surrogate model accepts parameters of the physics-based model as inputs, we can simulate faults. We trained the model using Adam (Kingma & Ba, 2015) algorithm, with a step size 0.001. We used the typical regression models loss function, i.e., the mean square error (MSE) loss function. All other hyper-parameters were left at their default values. An exemplar of the training results are shown in Figure 3, where we compared the prediction of the mass flow rates against the ground-truth values. The overall loss function value was MSE $<$ 1e-5. Armed with the NN-based model, we can solve the disambiguation optimization problem using forward propagations of this model. In the next section we give information about the gain in time efficiency as compared to an FMU-based, black-box optimization.

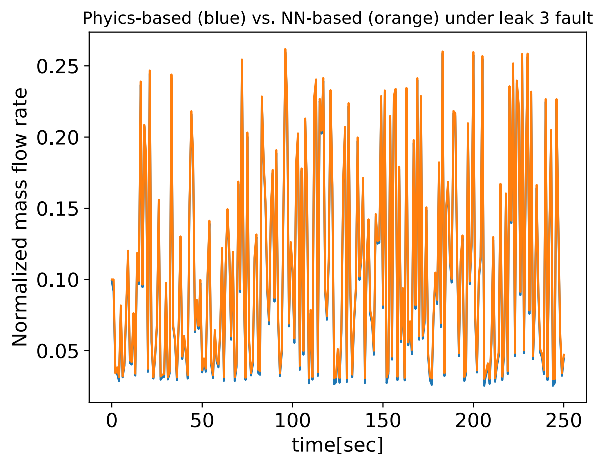

Figure 3. Physics-based vs. NN-based model comparison

4. RESULTS

The time horizon for the data gathering used for diagnosis is 500 sec. We consider two scenarios: in the first scenario we measure both the tank level and the engine mass flow rate, while in the second scenario we measure the engine mass flow rate, only. In each scenario we consider two levels of signal-to-noise rations (SNRs) that affect the measurements: 40 dB and 25 dB. Typically, measurements with SNR greater than 40 dB are measurements where the useful information dominates the measurement noise. Measurements with SNR below 20 dB are considered unreliable. To test the diagnosis algorithms for the four fault modes, we generated synthetic data to which noise was added to reach the two SNR levels. Each FMU-based simulation takes about 0.07 sec and a diagnosis solution is generated in about 4 seconds, well within the 500 seconds measurement time window.

Tables 1 and 2 present the fault probabilities for the first scenario. The rows of the table correspond to the ground truth and for each fault case, the ground truth fault parameter is set to 0.5. All correct fault modes are predicted with high probability, for both SNR levels. As expected, for SNR=25 dB, the confidence in the diagnosis solution is smaller, but the likelihoods of the true diagnoses are dominant. In addition, all fault parameters are correctly estimated, as demonstrated in Table 3 that show the parameter estimates for SNR=25 dB. The diagnosis results for the second scenario are shown in Table 4, where we present the results for SNR=40 dB, only. While the fault parameters are correctly estimated (Table 5), due to the high uncertainty in the diagnosis solution, no decision can be made about the true fault mode.

Table 1. SNR 40 dB: Fault probabilities when measuring both the tank fuel level and the engine mass flow rate.

	leak 1	**leak 2**	**leak 3**	**leak 4**
leak 1	**0.99**	0.005	0.003	0.002
leak 2	0.003	**0.976**	0.015	0.006
leak 3	0.001	0.011	0.954	0.034
leak 4	0.001	0.005	0.038	**0.956**

Table 2. SNR 25 dB: Fault probabilities when measuring both the tank fuel level and the engine mass flow rate.

	leak 1	**leak 2**	**leak 3**	**leak 4**
leak 1	**0.803**	0.095	0.056	0.045
leak 2	0.055	**0.612**	0.221	0.111
leak 3	0.021	0.143	**0.536**	0.300
leak 4	0.016	0.075	0.326	**0.582**

Table 3. SNR 25 dB: Fault parameter estimates when measuring both the tank fuel level and the engine mass flow rate.

	leak 1	**leak 2**	**leak 3**	**leak 4**
leak 1	**0.500**	0.466	0.436	0.414
leak 2	0.507	**0.500**	0.480	0.466
leak 3	0.510	0.517	**0.500**	0.487
leak 4	0.511	0.529	0.512	**0.500**

Table 4. SNR 40 dB: Fault probabilities when the engine mass flow rate, only.

	leak 1	**leak 2**	**leak 3**	**leak 4**
leak 1	**0.25**	0.25	0.25	0.25
leak 2	0.25	**0.25**	0.25	0.25
leak 3	0.25	0.25	**0.25**	0.25
leak 4	0.25	0.25	0.25	**0.25**

Next, we design disambiguation control inputs that reduce the uncertainty of the diagnosis solution. One approach for designing the control is to use black-box optimization, where the model predictions are generated by the model FMU.

Table 5. SNR 40 dB: Fault parameter estimates when measuring the engine mass flow rate, only.

	leak 1	**leak 2**	**leak 3**	**leak 4**
leak 1	**0.500**	0.503	0.506	0.510
leak 2	0.496	**0.499**	0.503	0.507
leak 3	0.494	0.498	**0.501**	0.505
leak 4	0.490	0.493	0.497	**0.500**

Gradient-free algorithms, such as Powell, or Nelder-Mead can be used to search for disambiguation inputs. Unfortunately, they do not scale with the number of optimization variables. In our case the number of optimization variables depends on the length of the time horizon and the sampling period. The gradient-based algorithms, however do scale with the number of optimization variables. Figure 4 shows a comparison between the two types of algorithms, for various number of change points of the input signal. The change points are time instants τ_i such that $u(t) = \sum_i u_i \mathbb{1}(t - \tau_i)$, where $\mathbb{1}(t)$ is the step function. The Powell-based solution

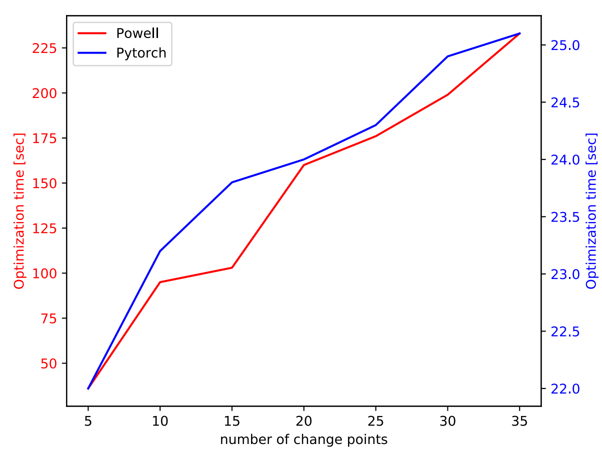

Figure 4. Optimization time comparison between the Powell gradient-free algorithm, and Adam gradient-based algorithm, as a function of number of change points of the input. The Powell algorithm uses the model FMU for simulations, while the Adam algorithms uses the surrogate model.

was solved using the Scipy scientific computing Python library, while the Adam-based solution was implemented and executed using Pytorch. Both approaches were executed on the same machine. Unsurprisingly, the graph clearly shows the superior efficiency of the gradient-based algorithm. For both algorithms we used the same tolerance value to have a common basis for comparison, i.e., `ftol`=10^{-5}.

We solved the disambiguation control design problem, where we allowed the input to change at 20 points over the 500 seconds time horizon. The problem is defined as

$$\begin{aligned}
\min_{u_l} \quad & -\sum_{i,j=1,i<j}^{4} \sum_{k=0}^{\tau} \|\hat{\boldsymbol{y}}_k^i - \hat{\boldsymbol{y}}_k^j\|^2 \\
\text{s.t.:} \quad & \hat{\boldsymbol{y}}_k^i = \hat{h}(\boldsymbol{u}_k; \hat{p}_i), i \in \{1,2,3,4\}, p_i = 0.5,\ \forall i \\
& \boldsymbol{u}(t_k) = \sum_{l=1}^{20} u_l \mathbb{1}(t_k - \tau_l)\, \boldsymbol{u}_l \in \mathcal{U}, \forall k \in \{0,\ldots,\tau\},
\end{aligned}$$

Here $\hat{h}$ denotes that NN-based surrogate model, and $\mathcal{U} = [0, 40]$ rad/s. Even for this number of change points, the algorithm took slightly over 25 seconds to generate the result. The generated control input is shown in Figure 5. For comparison, we include the inputs generated by the nominal controller, when the system is affected by the leak 1 fault mode.

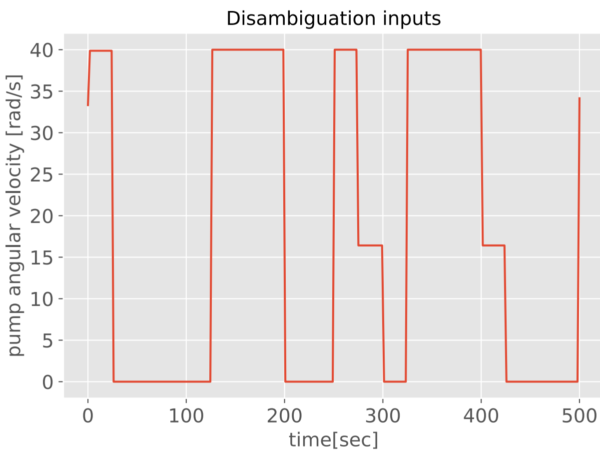

Figure 5. Optimal control for fault disambiguation.

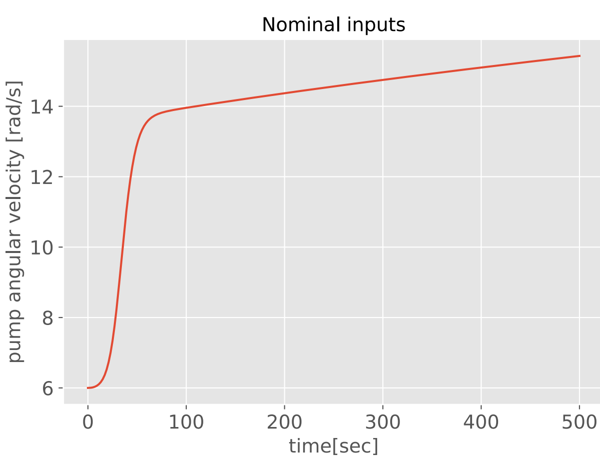

Figure 6. Inputs generated by the nominal controller under leak 1 fault mode.

We applied the control to the system, and considered the four leaks with the fault parameters equal to 0.5. The mass flow rate corresponding to the fault mode 1 as a result of applying the disambiguation input and for SNR=25 dB is shown in Figure 7.

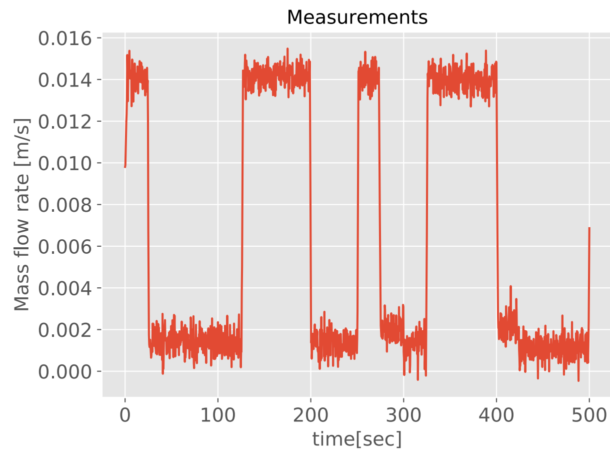

Figure 7. Mass flow rate corresponding leak 1 fault when applying the disambiguation input, and with SNR=25 dB.

The results for SNR 40 dB and 25 dB are shown in Tables 6 and 7, respectively.

We note that the probabilities of the ground truth in the four cases for SNR=40 dB, while smaller then the case where we measure both the mass flow rate and the tank level, still dominate. For SNR=25 dB, while the probabilities of the ground truth decrease even more, we do not have the uniform distribution when using the nominal inputs, as shown in Table 5. The fault parameters were correctly estimated and were approximately 0.5, as expected.

Table 6. SNR 40 dB: Fault probabilities when measuring the engine mass flow rate, using the disambiguation inputs.

	leak 1	**leak 2**	**leak 3**	**leak 4**
leak 1	**0.475**	0.332	0.164	0.029
leak 2	0.306	**0.436**	0.231	0.028
leak 3	0.176	0.264	**0.506**	0.054
leak 4	0.120	0.177	0.217	**0.486**

Table 7. SNR 25 dB: Fault probabilities when measuring the engine mass flow rate, using the disambiguation inputs.

	leak 1	**leak 2**	**leak 3**	**leak 4**
leak 1	**0.370**	0.332	0.234	0.064
leak 2	0.312	**0.351**	0.274	0.063
leak 3	0.231	0.281	**0.372**	0.116
leak 4	0.178	0.224	0.249	**0.350**

5. CONCLUSIONS

This paper has shown how to address the ambiguous fault diagnosis problem. Such a problem appears when sensor measurements do not include sufficient information to compute the fault probabilities with high certainty. We approached the fault disambiguation problem from an optimal control perspective: we designed control inputs such that the similarity between measurements in ambiguous fault modes is reduced. We showed that we can improve the scalability of the optimal control problem with the number of optimization variables by: (i) using a surrogate model expressed with constructs endowed with AD operators, and (ii) using gradient-based optimization algorithms that use automatic differentiation to compute gradients of the cost function. We demonstrated our approach on a fuel system example, and showed that using disambiguation control inputs we can isolate faults with high probability even when measuring the engine mass flow rate, only. An extension of the current results will add to the similarity-based objective control requirements such as stability. The resulting control inputs will represent a trade-off between maximizing the likelihood of fault detection and isolation, and constraining the system behavior within safe operating modes.

REFERENCES

Arulampalam, M. S., Maskell, S., & Gordon, N. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, *50*, 174–188.

Blochwitz, T., Otter, M., Arnold, M., Bausch, C., Elmqvist, H., Junghanns, A., ... Clauß, C. (2011). The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of the 8th international modelica conference* (p. 105-114).

de Kleer, J., Mackworth, A., & Reiter, R. (1992). Characterizing diagnoses and systems. *"Journal of Artificial Inteligence"*, *56*(2–3), 197–222.

Fritzson, P. (2015). *Principles of object-oriented modeling and simulation with Modelica 3.3: A cyber-physical approach* (2nd ed.). Hoboken, NJ: Wiley.

Gertler, J. (1998). *Fault-detection and diagnosis in engineering systems*. New York: Marcel Dekker.

Isermann, R. (2005). Model-based fault-detection and diagnosis - status and applications. *Annual Reviews in Control*, *29*(1), 71 - 85.

Julier, S. J., & Uhlmann, J. K. (1997, July). New extension of the Kalman filter to nonlinear systems. In I. Kadar (Ed.), *Signal processing, sensor fusion, and target recognition vi* (Vol. 3068, p. 182-193).

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, *82*(Series D), 35–45.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

Kuhn, L. D., Price, B., de Kleer, J., Do, M. B., & Zhou, R. (2008). Pervasive diagnosis: The integration of diagnostic goals into production plans. In D. Fox & C. P. Gomes (Eds.), *AAAI* (p. 1306-1312). AAAI Press.

McElhoe, B. A. (1966, July). An Assessment of the Navigation and Course Corrections for a Manned Flyby of Mars or Venus. *IEEE Transactions on Aerospace Electronic Systems*, *2*(4), 613-623.

Minhas, R., de Kleer, J., Matei, I., Saha, B., Janssen, B., Bobrow, D., & Kurtoglu, T. (2014). Using fault augmented modelica models for diagnostics. In *Proceedings of the 10th international modelicaconference* (pp. 437–445).

Paszke, A., & et al. (2017). Automatic differentiation in PyTorch.

Patton, R. J., Frank, P. M., & Clark, R. N. (2000). *Issues of fault diagnosis for dynamic systems*. Springer-Verlag London.

Saha, B., & et al. (2014, August,). Model-based approach for optimal maintenance strategy. In *Proceedings of second european conference of the prognostics and health management society.*

Staroswiecki, M. (2000). Quantitative and qualitative models for fault detection and isolation. *Mechanical Systems and Signal Processing*, *14*(3), 301 - 325.

Staroswiecki, M., & Comtet-Varga, G. (2001). Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, *37*(5), 687 - 699.