The Case for a Hybrid Approach to Diagnosis: A Railway Switch

Ion Matei and **Anurag Ganguli** and **Tomonori Honda** and **Johan de Kleer**
Palo Alto Research Center, Palo Alto, California, USA
e-mail: {imatei,aganguli,thonda,dekleer}@parc.com

Abstract

Behavioral models are at the core of Fault-Detection and Isolation (FDI) and Model-Based Diagnosis (MBD) methods. In some practical applications, however, building and validating such models may not always be possible, or only partially validated models can be obtained. In this paper we present a diagnosis solution when only a partially validated model is available. The solution uses a fault-augmented physics-based model to extract meaningful behavioral features corresponding to the normal and abnormal behavior. These features together with experimental training data are used to build a data-driven statistical model used for classifying the behavior of the system based on observations. We apply this approach for a railway switch diagnosis problem.

1 Introduction

Consider the case of developing diagnostic software for a complex system (for this paper our example is a railway switch). The task is to determine from operational data whether the switch is operating correctly or in one of a fixed number of fault modes. We are given the following very limiting (but all too common) conditions: (a) very limited resources to complete the project (a few man months); (b) limited number of sensors; (c) unavailability of the model of the system; (d) unavailability of the system itself (would require an instrumented private rail system); (e) unavailability of the parameters of the system components; (f) limited nominal data; (g) extremely limited fault data (supplied as time series); (h) highly non-linear multi-physics system having multiple operating modes. Broadly speaking there are three approaches to this type of problem: Model-Based Diagnosis (MBD), Fault Detection and Isolation (FDI) and Machine Learning (ML). None of these approaches is adequate of this task. MBD and FDI require models and parameters which are unavailable. ML approaches will require a large amount of training data, and most approaches would require extensive feature engineering. In this paper we will demonstraint a hybrid approach to this task which was ultimately fully satisfactory for the train company. Many real world diagnostic tasks have similar limitations and we believe our approach is one that yields good diagnostic algorithms for many cases.

At a high level our approach is as follows. First we build by hand an approximate model in Modelica (our switch model ultimately has 56 continuous time state and more than 2000 time-varying variables). We require this model to contain the key mechanisms which comprise a switch mechanism. Under the limiting conditions, building an accurate model of the system proved to be impractical and therefore we used simplified models for the system's components. For example, we model the controller as a PID controller while the actually mechanism surely has a more complex one. The Modelica model is fault augmented [Minhas *et al.*, 2014] including parameters which represent the fault amounts for wear, etc. Second, we develop ML classifiers to detect and diagnose faults by running the Modelica model repeatedly with various fault amounts. We mix noise in the simulation to avoid over-fitting. For the ML classifier to work requires developing a set of features for the signal. Each time series is segmented at defined conditions and a set of features is designed (e.g., mean in segment, max in segment). Multiple ML techniques can develop a classifier, the best we found are based on random-forest. Third, we throw away the model — it was only important to develop the features and the classifier. We now use the classifiers developed for the synthetic data on the real data. We were able to *detect* faults with a high level of accuracy, but were only partially successful in identifying the correct fault mode (or nominal) for the operating system. Independently, we showed that given enough data for the various fault modes, *using the same set of features*, a ML classifier can be designed that also achieves a high diagnostic accuracy. The latter effort is not the subject of the paper. Overall, the customer was very satisfied with the results of the project. Throughout the rest of the paper we describe in detail the procedure described above.

1.1 FDI and MBD

In model-based approaches (FDI and MBD), the diagnosis engine is provided with a model of the system, values of the parameters of the model and values of some of its inputs and outputs. Its main goal is to determine from only this information whether the system is malfunctioning, which components might be faulty and what additional information need to be gathered (if any) to identify the faulty components with relative certainty. The distinguishing features of the MBD [de Kleer *et al.*, 1992] approach are an emphasis on general diagnostic reasoning engines that perform a variety of diagnostic tasks via on-line reasoning, and inference of a system's global behavior from the automatic combination of physical components. Hence, MBD models are compositional - the model of a combination of two systems

is directly constructed from the models of the constituent systems. FDI methods can work with both physics-based and empirical models. The physics-based models are usually flattened, that is, the components and sub-components structure is lost into an overall behavioral model. Often, the faults are seen as separate inputs that need to be computed by the diagnosis engine. The disadvantage of this approach is that the physical semantics of the faults is ignored. In addition, treating the faults as exogenous inputs ignores the fact that the abnormal behavior may in fact depend on the variables of the systems. However, many FDI techniques were shown to be effective in diagnosing dynamical systems [Gertler, 1998; Isermann, 1997; 2005; Patton *et al.*, 2000].

The above discussion emphasizes the need for a model when using either an FDI or MBD approach. As we will see later in the paper, there are cases when such a model is very difficult to obtain and (more importantly) validate, or only a partial model is available. Naturally, both FDI and MBD approaches would not fare well in such a scenario. When no model is available, data-driven methods can be used to learn the behavior of the system and use this knowledge to predict the system behavior. Such methods require experimental data corresponding to the normal and abnormal behavior for classification purposes; data that is used to extract features representative for the system's behavior. The set of features together with observations of the system (output measurements) are used to learn a data-driven statistical model that is further used to classify the current observed behavior. Namely, when new data is available it is fed into the data-driven model, which in turn will provide a "best guess" to which class of behavior (normal or abnormal) the data corresponds to. It is well recognized that in data-driven approaches, the effectiveness of the classification is highly dependent on the quality of the features used for learning.

In this paper, we begin to bridge the gap between pure model-based and data-driven methods with a more hybrid approach. We propose the use of a partially validated model to help us determine a set of features that are representative for the normal and abnormal behavior. In this approach we build a physics based model of the system, emphasizing its components and sub-components. Due to the lack of sufficient technical specifications and measurement data, only partial validation is achieved. By this we mean that only a sub-set of the variables of interest match their counterpart in the experimental data. The rest of the variables, although not completely matching the real data, they do exhibit similar characteristics compared to the real data, e.g., same number of maxima, minima, or common regions of increasing/decreasing values, etc. In other words they are qualitatively equivalent. The physics-based model is further extended to include behaviors under different fault operating modes. In particular, physics-based models for the faults are included in the nominal model. The fault-augmented model is then used to generate synthetic simulated normal and abnormal (including multiple faults) behavior and extract representative features that are used in a data-driven approach. Note that although ideally we would like to execute the feature extraction step automatically, in this paper it is performed manually as the automatic feature extraction is a challenging problem in its own. The diagnosis procedure described above is pictorially presented in Figure 1.

The rest of the paper is organized as follows: in Section 2 we motivate and describe the railway switch diagnosis problem. Sections 3 and 4 present the physics-based model, its fault-augmented version and the partial validation of the system. Section 5 describes the diagnosis solution under a partially validated physics-based model while Section 6 puts our solution in the context of exiting work on railway switch diagnostics.

2 Problem Description

Railway signaling equipment (including switches) generates approximately 60% of the failure statistics related to traffic disruptions due to signalling problems. As a consequence more and more attention is paid to railway safety and optimal railway maintenance. As a result of the rapid technological advances in microelectronics and communication technologies in the past decades, it has become possible to add sensing and communication capabilities to railway equipment such as switches, to detect equipment failure and therefore to enhance the quality of the railway service. Although these sensing capabilities allow for easy detection of faults in the electrical components of the equipment, a significant number of faults related to the mechanical components affect parameters whose monitoring would be difficult either due to cost or impracticality of sensor placement.

The rail switch assembly considered in this paper is shown Figure 2. The component responsible for moving the switch blades is the point machine. The point machine has two sub-components: a servo-motor (generates rotational motion) and a gear-cam mechanism (amplifies the torque generated by the motor and transforms the rotational motion into a translational motion).

The adjuster transfers the motion from the point machine to the load (switch blades) through a drive rod. In particular, by adjusting two bolts, the adjuster controls the time when the switch blades start moving having as reference the time when the drive rod commence moving. The switch blades are supported by a set of rolling bearings to minimize motion friction. The manufacturer of the point machine endowed the equipment with a series of sensors that can measure the motor's angular velocity and torque, and the cam's angle and stroke (linear position). These sensors log data in real time which is ten sent to a central station for analysis. These sensors were installed by design on the point machine to monitor its safety. Although the operator of the railway switch is also interested in the diagnosis of the point machine, other possible faults are of interest as well. The faults considered in this paper are as follows: *loose lock-pin* fault (at the connection between the drive rod and the point machine), adjuster *bolts misalignment* (the bolts move away from their nominal position), *missing bearings* and the presence of an *obstacle* preventing the completion of the switch blades motion. Adding new sensors measuring forces applied to the switch blades or the position of the switch blades may facilitate immediate detection of such faults. However, due to the sheer number and possible configurations of switches in the railway transportation network, this is not a scalable solution. Therefore, the challenge is to diagnose the aforementioned faults using *only the available measurements.*

3 System Modeling

This section presents the fault augmented physics-based model of railway switch assembly, together with some

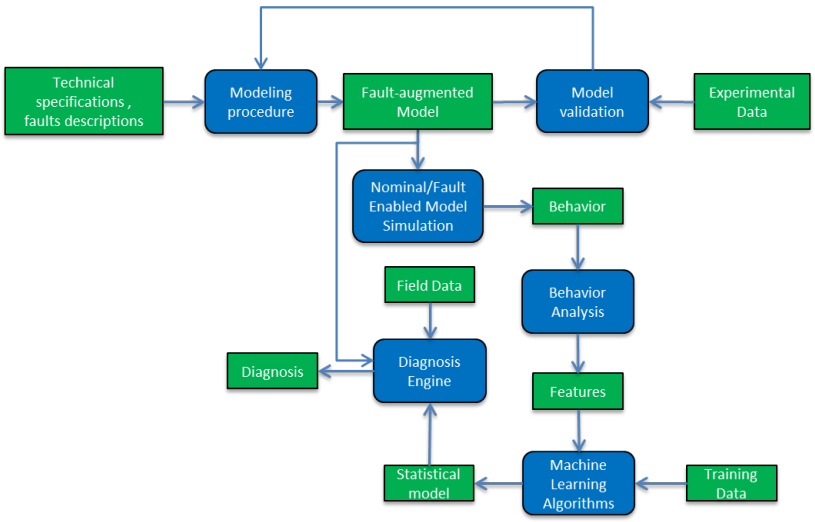

Figure 1: Diagnosis procedure with partially validated model

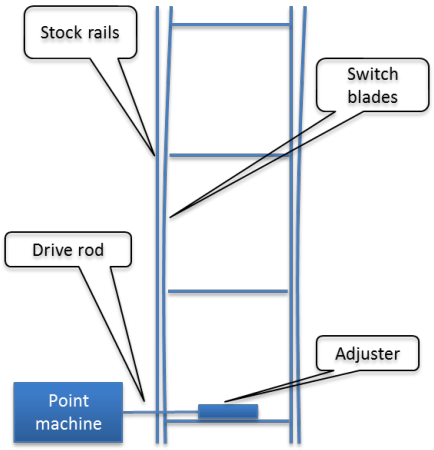

Figure 2: Diagnosis procedure with partially validated model

model validation results. Such models provide deeper insight on the behavior of the physical system. Simulated behavior helps with learning of normal and abnormal behavior patterns. The abnormal patterns are especially useful when not enough experimental data describing the abnormal behavior is available. The modeling process consists of decomposing the system into its main components, build physical models and combining them into an overall model of the system. We used the Modelica language to construct the model, which is a non-proprietary, object-oriented, equation based language to model complex physical systems [Tiller, 2001]. Models for the three main components of the railway switch, the point machine, the adjuster and the switch blades, are presented in what follows.

3.1 Point machine

The point machine is the component of the railway switch system that is responsible for moving the switch blades and locking them in the final position until a new motion action is initiated. It is composed of two sub-components: servomotor and gear-cam mechanism. The electrical motor transforms electrical energy into mechanical energy and generates a rotational motion. The gear-cam mechanism scales down the angular velocity of the motor and amplifies the torque generated by the motor. In addition, it transforms the rotational motion into a translational motion.

Servomotor

No technical details were provided on this component, such as type of motor or type of controller. Values for technical parameters (e.g., armature resistance, motor shaft inertia) were not available either. This information was not available to the switch operator either. Therefore, as a result of a literature review on the type of motors used in railway switches, a DC-permanent motor was chosen to be the most likely candidate. The dynamical model for this component is given by

$$\begin{aligned} L_a \frac{di(t)}{dt} &= -R_a i(t) - K_e \omega(t) + v(t), \\ J \frac{d\omega(t)}{dt} &= K_t i(t) - B\omega(t) - \tau(t), \end{aligned}$$

where $v(t)$ acts as input signal, $\omega(t)$ is the angular velocity at the motor flange that acts as output, $\tau(t)$ is the torque load of the motor and $i(t)$ is the current through the armature. Generic motor parameters from the literature were also chosen [Zattoni, 2006]. One question that may arise is if an empirical model can be estimated. Unfortunately since only the output $\omega(t)$ is available, an empirical model based on system identification cannot be estimated, since no voltage measurements are available. No information on the type of controller was available to us either. As a consequence, we used a PID controller for the feedback loop. Based on the observed profile of the motor output we determined that the controlled variable is the angular velocity $\omega(t)$. Indeed, Figure 3 shows the motor's angular velocity[1] that is maintained at a constant value by the controller. To compute the parameters of the PID controller we estimated metrics corresponding to the transient component of the output (angular velocity), such as rise time and overshoot; metrics that are formulated in .

[1]The angular velocity profile shown in the graph is similar but not exactly the observed one, due to proprietary information restrictions.

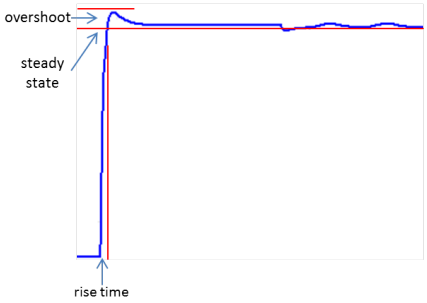

Figure 3: Motor angular velocity

The Gear-Cam mechanism

As mentioned earlier, the gear-cam mechanism amplifies the torque generated by the motor and transforms the rotational motion into a translational motion. The technical details provided to us confirmed only the presence of the cam, but not of the gear. We inferred the presence of the latter, by comparing the angular velocity of the motor with the cam's angular velocity, estimated from the measured cam's angle. This allowed us to estimate the ratio between the two velocities, and therefore estimate the gear ratio. The cam diagram is shown in Figure 4, where a wheel rotates as a result of the torque transmitted through the gear and acts on a lever that pushes the drive rod. Using the geometry of the cam,

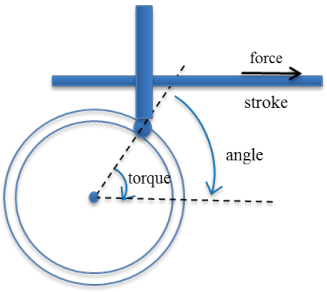

Figure 4: Cam schematics

the relation between the rotation motion and the linear motion (that is, the relation between the angle and the stroke) is given by

$$\text{stroke} = R \times \sin(\text{angle}),$$

where R denotes the radius of the cam. In addition, the map between the applied torque and the generated force is

$$\text{force} = \frac{1}{R} \times \text{torque} \times \cos(\text{angle}).$$

As both the cam angle and the stroke were included in the available measurements, we used a least square method to estimate the radius of the cam.

3.2 Adjuster

The adjuster links the drive rod connected to the point machine to the switch blades, and hence it is responsible for transferring the translational motion. There is a delay between the time instants the drive rod and the switch blades start moving. This delay is controlled by setting the positions of two bolts on the drive rod. Tighter bolt setting means a smaller delay, while looser bolt setting produce a larger delay. The high level diagram of the adjuster is depicted in Figure 5. The most challenging part in constructing the adjuster was modeling the non-sticking contact between the drive rod and the adjuster extremes. Stiff contact two bodies is usually modeled using a spring-damper component with very large values for the elasticity and damping constants. However, under this approach once contact takes place, it is permanent. To solve this challenge, we built a custom component that models the non-sticking contact.

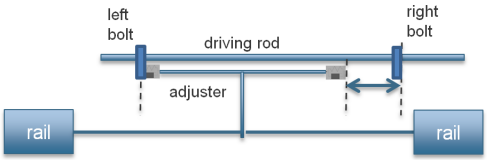

Figure 5: Adjuster diagram

3.3 Switch blades

The adjuster is connected to two switch blades that are moved from left to right or right to left, depending on the traffic needs. We look at a switch blade as a flexible body and used an approximation method to modeling beams, namely the lumped parameter approximation. This method assumes that beam deflection is small and in the linear regime. The lumped parameter approach approximates a flexible body as a set of rigid bodies coupled with springs and dampers. It can be implemented by a chain of alternating bodies and joints. The springs and dampers act on the bodies or the joints. The spring stiffness and damping coefficients are functions of the material properties and the geometry of the flexible elements. Parameters such a rail length, mass and mass moment of inertia were provided to us through technical documentation. To model the effect of the rail moving on rolling bearings, we included a friction component that accounts for energy loss due to friction. Although the component can model different friction models, the default models is Coulomb friction.

3.4 Fault augmentation

In this section we describe the modeling artifacts that were used to include in the behavior of the system the four fault operating modes: loose lock-pin, misaligned adjuster bolts, obstacle and missing bearings.

Loose lock-pin

The lock-pin referred in this fault mode connects the point machine with the drive rod that transfers the motion to the switch blades. More precisely, it locks the drive rod to the point machine. When this lock-pin becomes loose due to wear, it introduces a slackness in the way the motion is transferred to the switch blades. The lock-pin fault affects stability the connection point between the drive rod and the point machine. In time, if not fixed, this can lead to a complete failure of the pin, and therefore the point-machine cannot longer act upon the blades. A custom-built component whose main characteristic is that it implements a non-sticking pushing and pulling between two rods was built to model the effects of this fault. The impact between the two rods is assumed to be elastic, that is, we use a spring-damper assembly with large values for their parameters to model the contact. There are two types of contact: contact of the rods with the boundaries of the locking mechanism and contact between the rods. Both these types of contact must exhibit non-sticking pushing and pulling properties.

Misaligned adjuster bolts
In this fault mode the bolts of the adjuster deviate from their nominal position. As a result, the instant at which the drive rod meets the adjuster (and therefore the instant at which the the switch rail starts moving) happens either earlier or later. For example in a left-to-right motion, if the left bolt moves to the right, the contact happens earlier. The reason is that since the distance between the two bolts decreases, the left bolt reaches the adjuster faster. As a result, when the drive rod reaches its final position, there may be a gap between the right switch blade and the right stock rail. In contrast, if the left bolt moves to the left the contact happens later. The model of the adjuster includes parameters that can set the positions of the bolts, and therefore the effects of this fault mode can be modeled without difficulty.

Obstacle
In this fault mode, an obstacle prevents the switch blades reach their final nominal position, and therefore a gap between the switch blades and the stock rail appears. The effect on the motor torque is a sudden increase in value, as the motor tries to overcome the obstacle. To model this fault we included a component that implements a *hard stop* for the position of the switch blades. This component has two parameters for setting the left and right limits within motion of the switch blades is allowed. By changing the values of these parameters, the presence of an obstacle can be simulated.

Missing bearings
To minimize friction, the rails are supported by a set of rolling bearings. When they become stuck or lost, the energy losses due to friction increase. As mentioned in the section describing the switch blades modeling, a component was included to account for friction. This component has a parameter that sets the value for the friction coefficient. By increasing the value of this parameter, the effect of the missing bearings fault can be simulated.

4 Model Validation

Motor angular velocity, cam angle and stroke, together with the motor torque were used in the validation process. To these measurements, we added the rail position that was estimated from a set of movies depicting the rail motion, to which image processing techniques were applied. We achieved partial validation of the model. The simulated motor angular velocity, cam angle and stroke closely match the measured data. The simulated motor torque however matches in a qualitative sense its measured counterpart. The main reason is the fact that we had to make assumptions on the type controller motor and controller, without no way to validate these assumptions. In addition, the available measurements did not allowe for the estimating the parameters in the assumed models, as this problem is ill posed. Figure 6 depicts the simulated torque, emphasizing the five operating zone. In Zone 1, the motor rotates the cam and the drive rod moves freely. No contact with the switch blades takes place in this zone, and the (small) energy loss is due to friction in the mechanical components. Zone 2 corresponds to the case where the drive rod pushes the two switch blades. The elasticity in the switch blades can be noticed in the toque profile in this zone. In Zone 3, the switch blades accelerate (as they drop off the rolling bearings) and again the drive rod moves freely (note the drop in torque). Zone 4 depicts the case

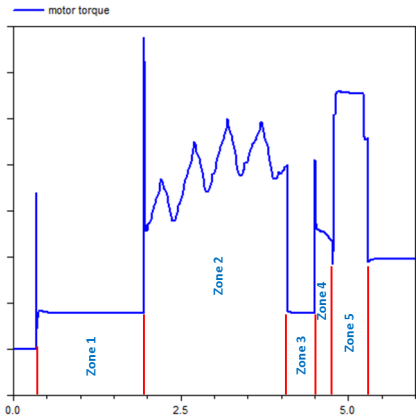

Figure 6: Motor torque with its five operating zones

where the drive rod catches up again with switch blades an pushes them to their final position. Finally, in Zone 5 the switch blades are pushed against the stock rails for a short period of time, hence the increase in torque. In support of the validation of these five operating zone, a set of movies depicting the motion of the switch blades were used. With respect to the fault operating modes, we managed to generate similar effects in the simulated data, as the ones observed in the measured data. Figure 7 shows the effect of the misaligned bolts fault, and in particular the case where the left bolt moves to the left. The effect is a delay applied on the time instant the drive rod reaches the switch blades. In addition, Zone 5 is also affected since due to the decreased distance, the switch blades are no longer pushed against the stock rails. In the case of an obstacle, the switch blades (and

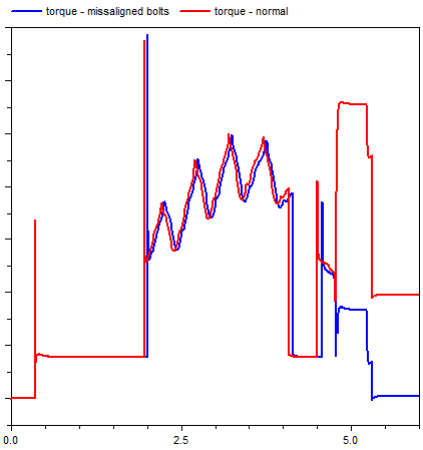

Figure 7: Motor torque in the normal and misaligned bolts fault modes

hence the drive rod) push against an obstacle that does not allow the completion of the motion. Therefore, the electric motor develops the maximum allowable torque as seen in Figure 8. In the case of the missing bearing fault mode, the motion friction of the switch blades increases, and hence the torque generated by the motor must accommodate this increase. We obtained this effect in simulation as shown in Figure 9. Finally, Figure 10 shows the effects of the lock-pin fault. The slackness introduced by the looseness of the pin induces a delay in the rail motion which also affects the behavior in Zone 5. In terms of the changes in the five op-

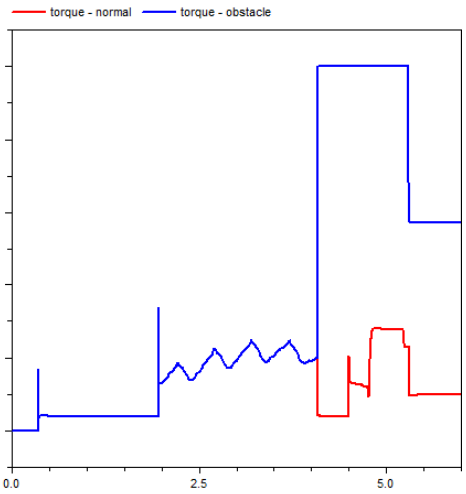

Figure 8: Motor torque in the normal and obstacle fault modes

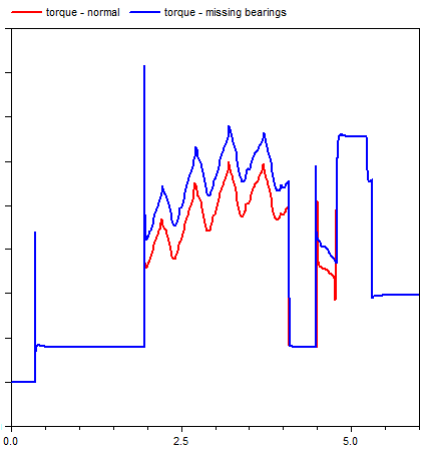

Figure 9: Motor torque in the normal and missing bearings fault modes

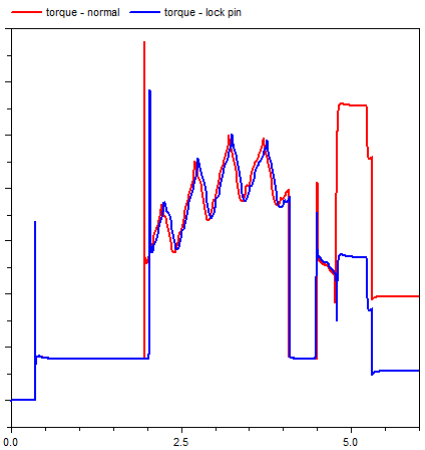

Figure 10: Motor torque in the normal and lock-pin fault modes

erating zones, the simulated behavior showed similar characteristics as in the case of the real data. The understanding of these behaviors come as a result of building the model, augmenting the model with fault modes, and analyzing their effects in simulation. The choice of features described in the next section was supported by this understanding.

5 Fault Detection and Diagnosis

In the case of a railway switch, our measurements include the motor torque and motor angular velocity. As the switch moves from one extreme position to the other, these quantities are measured at a fixed sampling rate. Thus, we obtain a time series for each of the measurements. Let $\{\tau(t_1), \ldots, \tau(t_N)\}$ denote torque measured at time instants $\{t_1, \ldots, t_N\}$. Likewise, let $\{\omega(t_1), \ldots, \omega(t_N)\}$ denote the angular velocity. For simplicity's sake, we denote the two time series of measurements by X. The diagnosis objective is to determine the underlying condition of the system from these time series. In other words, the objective is to determine a classifier $f : X \rightarrow \{N, F_1, F_2, F_3, F_4, F_5\}$, where N refers to the class label corresponding to the normal condition and F_1, F_2, F_3 and F_4 denote the class labels loose bolt, tight bolt, loose lock-pin, missing bearings, and obstacle respectively.

We adopt a machine learning approach to constructing the above mentioned classifier. The two main steps in building a machine learning classifier are feature selection and classifier type selection. These two steps are discussed next.

5.1 Feature selection

As seen in Figure 6, the motor torque profile shows five distinct operating zones. Moreover, we notice from Figures 7, 8, 9 and 10 that a given fault's impact on the torque profile seems limited to only some of the five zones. With this observation, our feature selection strategy is as follows.

1. Identify the approximate time instants that define the boundaries of the five zones. For example, Zone 1 is defined to be between times 0.8 seconds and 2 seconds, zone 2 is defined to be between times 2 seconds and 4.1 seconds, and so on.
2. Within each zone, compute a set of measures. An example of a measure is the total energy dissipated within the zone. This is computed as instantaneous power integrated over the duration of the zone. The instantaneous power is the product of instantaneous torque and angular velocity. Other examples of features include maximum and minimum torque values within the zone. The disclosure of the full set of measures used is not possible at this time for proprietary reasons. The features are normalized to have zero mean and unit standard deviation.

Note that it might be possible to combine one or more zones into one for feature selection.

5.2 Classifier selection

To map the features to the classes, $\{N, F_1, F_2, F_3, F_4, F_5\}$, we use machine learning. Examples of types of classifiers commonly used include $k-$ nearest neighbors, support vector machines, neural networks and decision trees. We chose Random Forest, an ensemble classifier, because of its robustness to overfitting. For a more detailed discussion on the advantages of Random Forest, we refer the reader to [Breiman, 2001]. In addition, we also developed a binary classifier for fault detection based on Alternating Decision Tree (AD Tree). The advantage of AD Tree is that the results are human interpretable.

5.3 Results

For each fault type, we introduce varying magnitudes of fault and simulate the switch model described earlier. The fault magnitude is parameterized by a factor k which is varied over a pre specified range. A value of k equal to zero corresponds to normal case. Higher values of k correspond to the faulty cases. In addition, we also add representative noise to the measurements. Figure 11 shows some example torque profiles generated by the simulation.

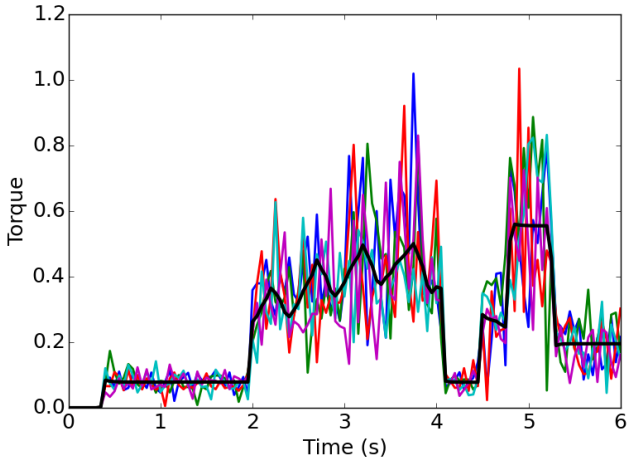

Figure 11: Simulated torque measurements with added noise.

The data generated is recorded and used to train and test the machine learning classifier. We use leave-one-out cross-validation for training and testing the classifiers. In this approach, one data sample is used for testing whereas all the rest of the data is used for training. This is repeated until each data sample has been tested once. Table 1 shows the confusion matrix for the simulated data described earlier. The (i, j)th entry of the confusion matrix refers to the percentage of cases where the true class was i but was classified as j by the classifier. A matrix with 100 along all the diagonal entries would correspond to a perfect classifier. In the results shown in Table 1, we observe some misclassification between classes N and F_4. Recall that N is the normal class and F_4 is the missing bearing class. On further investigation, we determined that the misclassification occurs between the normal data and data corresponding to *low magnitudes* of the missing bearing fault.

Table 1: Fault diagnosis confusion matrix on simulated data

	N	F_1	F_2	F_3	F_4	F_5
N	97.2	0	0	2	0.8	0
F_1	0	100	0	0	0	0
F_2	0	0	99	1	0	0
F_3	9	0	4	87	0	0
F_4	11	0	0	0	89	0
F_5	0	0	0	0	0	100

The binary classification or fault detection result using AD Tree is shown in Table 2. As in the multi-class classification case, the false positives (normal classified as abnormal), and false negatives (abnormal classified as normal) are primarily due to confusion between missing bearings and normal. Figure 12 shows part of the fault detection AD Tree. A pink oval represents a feature node. Depending on the value of the feature, one of two branches is followed until a leaf node is reached. Each edge that is traversed results in a score shown within the blue rectangles. For every root to leaf traversal, the total score is the sum of the scores accumulated on each edge. For a given data sample, multiple root to leaf paths may be traversed. In that case, the final score is the sum of the scores accumulated over all the paths. If the final score is negative, the decision is normal; otherwise the decision is abnormal.

Table 2: Fault detection confusion matrix on simulated data

	Normal	Abnormal
Normal	94.6	5.4
Abnormal	9.6	90.4

Next, we test the classifiers on real data. A key preprocessing step is to compute a linear transformation that transforms the mean and standard deviation of the features of the nominal (normal) real data to make them equal to the mean and standard deviation of the features of the nominal simulated data. The same transformation is then applied on the real faulty data before testing with the ML classifier. We emphasize here that to compute the transformation we only require examples of real data showing normal behavior. We do not use any real fault data for training the ML classifier. Table 3 shows the fault detection results on real data. As can be seen, we achieve a high accuracy of greater than 80 percent. We also tested the multi-class random forest classifier to diagnose the various faults. We were able to diagnose correctly all missing bearing faults but were unable to correctly diagnose the other faults.

Table 3: Fault detection confusion matrix on real data

	Normal	Abnormal
Normal	85.5	14.5
Abnormal	20	80

6 Related Work

A malfunctioning railway switch assembly can have a high impact on the railway transportation safety, and therefore the problem of diagnosing such systems has been addressed in other works. [Zattoni, 2006] proposes a detection system based on off-line processing of the armature current and voltage. The system implements an algorithm that realizes a finite impulse response system designed on the basis of an H_2-norm criterion, and allows for detection of incremental faults (e.g., loss of lubrication, increasing obstructions, etc.). The approach hinges on the availability of a validated model of the point machine, which was not the case in our setup. [Zhou *et al.*, 2001; 2002] propose a remote monitoring system for railway point machines. The system includes a variety of sensors for acquiring trackside data related to parameters such as, distance, driving force, voltage, electrical noise, or temperature. The monitoring system logs data for offline analysis that offers detailed information on the condition of the system in the form of event

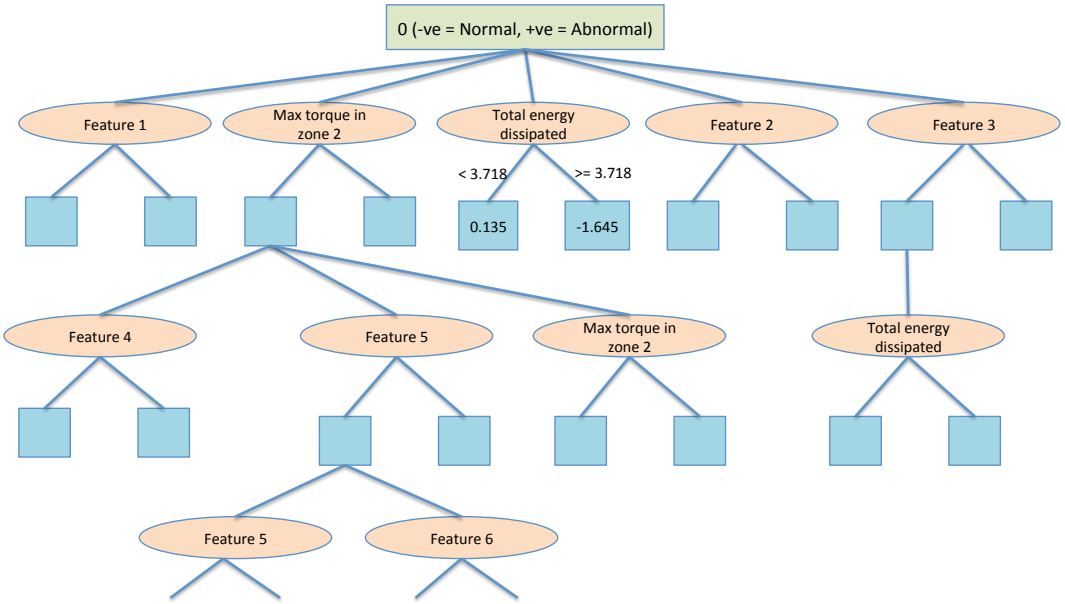

Figure 12: Part of the fault detection AD Tree

analysis and data trends. Hence unlike in our setup, the focus is on detection rather than isolation. In addition, due to scalability constraints, our solution is based on the embedded sensors, no other sensor being added. In [Asada *et al.*, 2013] classification based fault detection and diagnosis algorithm is developed using measurements such as drive force, electrical current and voltage. In particular, a classifier based on support vector machines is used. Our work also uses classification for diagnosis, but considers a wider verity of classifiers such as Multiclass Random Forest or Logitboosted Random Forest that were proved to be more robust [Opitz and Maclin, 1999]. The classification step in [Asada *et al.*, 2013] depends on a set of features extracted by applying the discrete wavelet transform on the active power. This step is oblivious on the operating modes of the point machine, which we showed to relevant in our case. Hence, the diagnosis approach in [Asada *et al.*, 2013] is purely data driven. Since we had no access to current and voltage measurements this avenue for feature construction was not available to us. Depending of the type of electrical motors, the current and the voltage could be computed from the angular velocity and torque, respectively. However, knowledge of motor parameters is needed. [Asada *et al.*, 2013] consider two type of faults: underdriving and overdriving of the drive rod. Overdriving refers to the case where the switch blades are pushed against the stock rails due to misalignment, and a higher force then normal appears between the stock rails and the switch blades. Overdriving map to misaligned bolts, missing bearings and obstacles in our setup. All these fault modes exhibit higher forces than normal. Underdriving maps to a particular instance of the misaligned bolts fault (left bolt moves to the left for example). Therefore, our solution differentiate between more possible causes of higher forces since we take advantage of the particular signature these forces have in each fault corresponding to overdriving. Another pure data-driven approach for railway point machine monitoring was proposed in [Oyebande and Renfrew, 2002], where a net energy analysis technique was used to discriminate between normal and abnormal behavior. This approach relies on a set of sensors measurements such as motors, voltage, current or switch blade positions, not all of them being available in our case. In addition, the computation of the net energy requires parameters of the electrical motor (armature resistance and motor shaft inertia) that again are not available in our setup. In addition, unlike our diagnosis objective, the focus in on detecting abnormalities within the point machine.

7 Conclusions

The three main general approaches to developing diagnostic software (FDI, MBR, and ML) all have severe limitations in many real-world applications. We believe we will see many more hybrid approaches to diagnosis that include the best of these three approaches to build accurate diagnosers.The railway switch is a critical and complex piece of equipment requiring extremely high diagnostic accuracy (the main reason this project was initiated), and the approach outlined in this paper was ultimately successful. Ultimately deployment of this approach will depend on expanding the set of faults detecting and on installation of more sensor rich switches in railroad infrastructures.

References

[Asada *et al.*, 2013] T. Asada, C. Roberts, and T. Koseki. An algorithm for improved performance of railway condition monitoring equipment: Alternating-current point machine case study. *Transportation Research Part C: Emerging Technologies*, 30(0):81 – 92, 2013.

[Breiman, 2001] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[de Kleer *et al.*, 1992] J. de Kleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. 56(2-3):197–222, 1992.

[Gertler, 1998] J. Gertler. *Fault-Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, 1998.

[Isermann, 1997] R. Isermann. Supervision, fault-detection and fault-diagnosis methods - An introduction. *Control Engineering Practice*, 5(5):639 – 652, 1997.

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

[Minhas *et al.*, 2014] R. Minhas, J. de Kleer, I. Matei, B. Saha, B. Janssen, D.G. Bobrow, and T Kortuglu. Using fault augmented Modelica model for diagnostics. In *Proceedings of the 10th International Modelica Conference*, Dec 2014.

[Opitz and Maclin, 1999] David Opitz and Richard Maclin. Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[Oyebande and Renfrew, 2002] B.O. Oyebande and A.C. Renfrew. Condition monitoring of railway electric point machines. *Electric Power Applications, IEE Proceedings* -, 149(6):465–473, Nov 2002.

[Patton *et al.*, 2000] Ron J. Patton, Paul M. Frank, and Robert N. Clark. *Issues of Fault Diagnosis for Dynamic Systems*. Springer-Verlag London, 2000.

[Tiller, 2001] Michael Tiller. *Introduction to Physical Modeling with Modelica*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[Zattoni, 2006] Elena Zattoni. Detection of incipient failures by using an -norm criterion: Application to railway switching points. *Control Engineering Practice*, 14(8):885 – 895, 2006.

[Zhou *et al.*, 2001] F. Zhou, M. Duta, M. Henry, S. Baker, and C. Burton. Condition monitoring and validation of railway point machines. In *Intelligent and Self-Validating Instruments – Sensors and Actuators (Ref. No. 2001/179), IEE Seminar on*, pages 6/1–6/7, Dec 2001.

[Zhou *et al.*, 2002] F.B. Zhou, M.D. Duta, M.P. Henry, S. Baker, and C. Burton. Remote condition monitoring for railway point machine. In *Railroad Conference, 2002 ASME/IEEE Joint*, pages 103–108, April 2002.