Development and embedded deployment of a virtual load sensor for wind turbine gearboxes

Jelle Bosmans¹,², Yon Vanommeslaeghe³,⁴, Luk Geens⁵, Jakob Fiszer¹,², Jan Croes¹,², Matteo Kirchner¹,², Joachim Denil³,⁴, Paul De Meulenaere³,⁴, Wim Desmet¹,²

¹ Flanders Make ‐ DMMS Core Lab
² Department of Mechanical Engineering, KU Leuven, Belgium
³ Flanders Make ‐ AnSyMo/CoSys Core Lab
⁴ Faculty of Applied Engineering, University of Antwerp, Belgium
⁵ ZF Wind Power, Gerard Mercatorstraat 40, 3920 Lommel, Belgium

Abstract. The loads to which a wind turbine gearbox is subjected during its lifetime can be a valuable source of information to decrease maintenance cost and downtime through enhanced monitoring, control and design. However, this load information is difficult to acquire since suitable direct load sensors are intrusive and expensive. Therefore, this paper focuses on indirect load measurement through a virtual sensing algorithm. The resulting virtual load sensor estimates the incoming load on the low speed planetary stage of the gearbox by combining strain measurements on the external surface of the ring gear with a physics-based model. The algorithm is deployed for real-time execution on low-cost embedded hardware to make a cost-effective load sensor. The effect of the configuration parameters of the virtual load sensor on the execution time and memory usage is examined in order to verify which configurations can be deployed. Since these configuration parameters also affect the estimation accuracy, the design of the virtual load sensor is tackled as a co-design problem. The resulting virtual load sensor, which is deployable for real-time execution, achieves an RMS estimation error of 0.6% in a numerical validation, using 4 strain gauges on the ring gear.

1. Introduction

In order to keep wind energy competitive with other power sources, continual efforts are spent to reduce the Operational Expenses (OPEX) and decrease the downtime of (future) wind turbines. A survey and categorization of failure events driving the OPEX of offshore wind turbines is given in [1]. According to this categorization, the subsystems with the highest failure rates are: (1) the pitch system, (2) miscellaneous components (e.g. hatches, lift), (3) the generator, (4) the gearbox, and (5) the blades. Although the gearbox has a lower failure rate, it is still a major contributor to both maintenance OPEX and downtime since gearbox repairs tend to have a higher material- and time cost. Therefore, a clear need exists for reliable gearbox designs as well as robust monitoring and control techniques to avoid overloading of critical gearbox components during operation.

The loads to which the gearbox is exposed during its lifetime constitutes a valuable source of information for such design principles, and monitoring and control techniques [2]. However, the value of this information is contrasted by the difficulty in acquiring it, since suitably heavy-duty direct load
sensors (e.g. the HBM T12HT [3]) are intrusive and expensive. Direct load measurement is therefore typically limited to lab use and prototype testing [4].

To circumvent the difficulty associated with direct load measurement, this paper focuses on cost-effective indirect load measurement through virtual sensing. A virtual sensor estimates the incoming load by combining a physics-based model with measurements from sensors that are more convenient to place than the direct load sensor [5]. Such an approach was already explored in the work of Perišić [4], who numerically validated a load torque sensor based on a torsional drivetrain model combined with encoder and generator torque measurements. Though promising results were demonstrated using this approach, the generator side torque sensor is still an intrusive sensor that is not suitable for long-term deployment. Another alternative approach is based on acceleration sensors in the blades [6], which could provide early detection of transient wind loads. This system is less intrusive since the relatively lightweight accelerometers are connected via a wireless system. However, both approaches for detecting the incoming loads on the gearbox require a coordinated effort among the OEMs of the turbine subsystems, which may not be straightforward.

From the point of view of a gearbox OEM, it is therefore relevant to develop a virtual sensing method based solely on measurements on the gearbox. Specifically, the approach presented in this paper focuses on the torque-driven loads on the low speed planetary stage of the gearbox. This focus on the low speed stage also paves the way towards the detection of important non-torque input loads. However, this goes beyond the scope of the current paper.

The virtual load sensor consists of three basic ingredients: an estimation algorithm, a sensor set, and a model of the gearbox. Each of these ingredients is discussed in sections 2, 3, and 4; respectively. The performance of the virtual load sensor is demonstrated in a numerical validation case in section 5. The effect of design parameters such as the number of sensors and the configuration of the estimation algorithm are clarified.

In order to successfully integrate the virtual load sensor into existing control and monitoring systems, it must be deployed on low-cost embedded hardware for real-time execution. To that end, a method to assess the computational performance of the algorithm and the model on the embedded hardware is introduced in section 6. The execution time is measured for different numbers of sensors and algorithm configurations. Furthermore, potential pitfalls are identified, such as the memory limits of the embedded platform.

Finally, section 7. discusses a technique to reduce the computational cost of the virtual load sensor based on a sensor switching scheme. In effect, this scheme increases the spatial resolution of the measurements while compromising on the sample rate. The effect of this trade-off on the performance and computation cost is analyzed.

2. Estimation algorithm

Estimation algorithms are designed to infer the state vector $\mathbf{x} \in \mathbb{R}^{N_x}$ of a dynamic system, based on a set of (noisy) measurements $\mathbf{y} \in \mathbb{R}^{N_y}$, where $N_x$ and $N_y$ are the amount of states and sensors, respectively. An overview and classification of several state estimation algorithms is given in [7].

The incoming loads experienced by the gearbox are an unknown input to the system. Therefore the algorithm has to solve a combined state-input estimation problem where an unknown input vector $\mathbf{u} \in \mathbb{R}^{N_{\text{inv}}}$ has to be estimated along with the states, where $N_{\text{inv}}$ is the amount of unknown inputs. One way to tackle this problem in a general state-estimation framework is to add the unknown inputs to the state vector as augmented states: $\mathbf{x}_a = \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \in \mathbb{R}^{N}$ where $N = N_x + N_{\text{inv}}$. Additionally, a model equation for the unknown inputs can also be defined. Often a random walk model is used: $\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{w}_u$, where $\mathbf{w}_u$ is a zero mean white-noise term [8].

The estimation can be performed by a Kalman Filter, which is a well-known single-step recursive stochastic estimator [9]. Its stochastic nature stems from the fact that it takes into account the relative uncertainty in the model predictions and measurement signal caused by modelling error and
measurement noise, respectively. Information with a lower uncertainty is given a higher weight in the tradeoff between model predictions and measurements.

Extending on this principle, the Moving Horizon Estimator (MHE) [10] considers the information over a finite length window of $N_w$ samples, instead of a single sample as for the Kalman Filter. When a new measurement is available, the window is shifted to include the latest sample. By considering information over a longer window, the MHE is able to deliver better estimation performance, in particular for highly nonlinear systems. Furthermore, the MHE is attractive for input estimation problems because it enables the use of more advanced input models, such as compressive sensing [11] or polynomial approximations [12]. These advantages motivate the decision to use the MHE in this paper. While the models in the present paper are not strongly non-linear, using the MHE paves the way for introducing a multibody model of the complete gearbox (with associated non-linearities), which is foreseen in future research. Additionally, investigating the real-time capabilities of the MHE in case of embedded deployment forms an important step towards the practical application of such algorithms, since they have a higher computational complexity than other estimation and control algorithms.

The MHE is expressed as an optimization problem that is solved at every time step. The estimation window contains $N_w$ discrete time steps $k = T - N_w + 1 \cdots T$ with $T$ indicating the current time step. The cost function (1) contains three terms to be minimized which are weighted through covariance matrices. These terms refer to: the arrival cost $P_a \in \mathbb{R}^{N \times N}$, which weighs the information coming from past estimates ($\bar{x}_{aT-N+1}$), the model error $Q \in \mathbb{R}^{N \times N}$, and the measurement error $R \in \mathbb{R}^{N_z \times N_z}$.

Because the optimization problem (1) is unconstrained and the covariance matrices are inherently positive definite, the optimization problem has a closed form solution that can be solved efficiently [12]. However, without loss of generality constraints on the optimization variables can be added, at the cost of a higher computation cost.

The computational complexity of the MHE is determined by the size of the optimization problem which is in turn determined by $N, N_z$ and $N_w$. The parameters $N$ and $N_w$ are design parameters of the MHE algorithm, which can be adapted to reach a certain estimation performance. The amount of augmented states ($N$) is considered fixed since it is determined by the system itself and how it is modelled.

$$\begin{align*}
\text{minimize} \quad & (x_{aT-N+1} - \bar{x}_{aT-N+1})^T P_a^{-1} (x_{aT-N+1} - \bar{x}_{aT-N+1}) \\
& + \sum_{k=T-N+1}^{T-1} (x_{a_k+1} - A_k x_{a_k})^T Q^{-1} (x_{a_k+1} - A_k x_{a_k}) \\
& + \sum_{k=T-N+1}^{T-1} (y_{k+1} - C_k y_k)^T R^{-1} (y_{k+1} - C_k y_k)
\end{align*}$$

(1)

3. Sensors

The main sensors used in the virtual load sensor are strain gauges placed on the external surface of the low speed stage ring gear (see Figure 1.a). Since the path between this strain gauge location and the forces applied on the ring gear’s internal teeth is short, these sensors have a good sensitivity to the input loads. Furthermore, the ring wheel is a component that is machined with high accuracy. Because of the low levels of geometrical uncertainty, highly accurate modelling is possible (see section 4). In a later stage, the strain gauges could be replaced by sensors that are more robust and that have a less labor-intensive mounting procedure, such as bolt-on load cells. However, because of the larger physical footprint said of load cells, additional processing is necessary to translate the output to a strain signal that can be compared to the model prediction.

An encoder on the planet carrier with an absolute reference that indicates the position of the planet gears is also included to supplement the strain gauges. This sensor is important since the relationship
between the strain and the incoming loads is strongly influenced by the angular position of the planet carrier and the planet gears.

4. Mechanical gearbox model

The gearbox model used in the virtual load sensor should be able to predict the strain measured by the strain gauges, based on the applied load. The estimation algorithm will then invert this relationship in order to determine the loads indirectly.

Three steps are taken to construct a model with these capabilities. These steps are shown graphically in Figure 1. First, the location and direction of the gear contact force is determined for all planets, given the measured position of the planet carrier. This calculation is based on the geometry of the planetary gear set and the kinematics of an ideal transmission [13]. This assumption implies that internal dynamics and flexibilities are neglected in this step.

Secondly, the contact forces are applied as point forces on the ring gear in a static Finite Element Model (FEM) of the housing. The strain at the external surface of the ring gear is considered independent from the exact load distribution along the tooth flank. This assumption is motivated by the Saint-Venant principle from elasticity theory. A similar assumption is made with respect to the boundary conditions of the gearbox at the torque arm connection.

Finally, the strain as predicted by the FEM is projected along the direction of the strain gauges. Since the FEM is linear, the resulting strain $\varepsilon$ can also be expressed as a linear relationship:

$$\varepsilon = S(\theta_c)F$$

In this equation, $F$ is a vector containing the magnitude tooth contact forces in the three planets. Only the magnitude is required since the direction and location of these forces are already known from the first modelling step. $S(\theta_c)$ is a transformation matrix, which depends on the planet carrier angle $\theta_c$, that relates the contact forces to the strain on the housing.

Figure 2 shows the simulated strain in the circumferential direction (also called hoop strain) at point S on the ring gear (Figure 1.a). In this simulation, a constant torque equal to 80% of the nominal value is applied at constant speed. Vertical lines indicate the instant when a planet passes the strain gauge, the peak strain of $80\mu m$ is found at the same time. Qualitatively, the shape of this simulated strain output agrees well with experimental and simulated results found in literature [14], [15].

Before the simulated strain is used to validate the virtual load sensor, artificial white noise is added to the signal. This white noise has a representative standard deviation of 5% with respect to the RMS value of the strain signal [16].
Figure 1: Construction of the mechanical gearbox model. The sensor position referred to in Figure 2 and Figure 3.a ($N_s = 1$) is indicated in Figure 1.a.

Figure 2: Simulated strain output at the location indicated in Figure 1.a over one rotation of the planet carrier at 80% load level. The vertical lines indicate when a planet passes the strain gauge.
5. Virtual load sensing results
In this section, the virtual load sensor is validated numerically using measurements generated by a reference simulation. The loads detected by the virtual load sensor are compared to the loads applied in the reference simulation to verify consistency and quality. The reference simulation consists of a torque ramp-up applied at constant speed, assuming an equal load sharing among the planet gears.

Figure 3 shows the estimation results and reference for the planet tooth forces, plotted in function of the planet carrier angle $\theta_c$. These results were generated with a window size of 15 time steps; sampling frequency of 1024 Hz; and 1, 4, or 8 equally spaced strain gauges.

Figure 3.a, shows the estimation results with 1 strain gauge, positioned as shown in Figure 1.a. Vertical lines indicate when a planet passes the strain gauge. When this happens, the virtual load sensor updates the estimate of the tooth force. The estimate is constant in between two updates. Because of this lag in the updating, the estimation performance with one sensor is poor, with a Root Mean Square Error (RMSE) on the resulting torque of 3.3%.

When 4 sensors are used, the updates occur more frequently, as shown in Figure 3.b (the vertical lines have been omitted from this graph to avoid over-cluttering). As a result, the RMSE drops to 0.6%. Moving to 8 sensors (Figure 3.c) does not significantly improve the performance (RMSE 0.6%), though qualitatively the estimation appears less noisy. These results confirm the feasibility of the virtual load sensing concept. The conclusions drawn here will be used to define an experimental validation in future research.

6. Embedded deployment of the virtual load sensor
Embedded deployment of the virtual load sensor poses a challenge since the MHE algorithm is computationally intensive. This is in contrast with the limited amount of resources available on embedded platforms, both in terms of available computational power and available memory. As such, the execution time of the virtual sensing algorithm as well as the memory usage need to be considered to evaluate the feasibility of embedded deployment.

While the MHE performance is affected by a number of configuration parameters, such as the window size, the number of sensors, and the sampling frequency, these parameters also affect the computational and memory requirements. For example, the window size and number of sensors determine the memory usage, but also the number of required operations at each iteration and thus the execution time on a given (embedded) platform. Similarly, the required sampling frequency determines the maximum allowed execution time (deadline) for each iteration. As such, it is important to take these relationships into account during the design and deployment process. To facilitate this,
the relationships between the different parameters and properties considered in this paper are captured in an ontology [17], shown in Figure 4. Here, lines indicate the existence of a relationship, while the plus and minus signs indicate the nature of this relationship. For example, increasing (+) the window size means that the dimensions of certain matrices increase (+) as well. Conversely, increasing (+) the sample frequency means a decrease (-) in task period (deadline).

![Ontology for the presented estimator, showing the relationships between different parameters and properties.](image)

This ontology indicates how information about execution time and memory usage might be used to make trade-offs between the control and embedded domain. For example, if the execution time is too long, or memory usage is too high, changes can be made to the virtual sensor configuration to make embedded deployment feasible. However, these changes also affect the performance of the virtual sensor itself. To support making such trade-offs during early stages of the development process, a process was set up to automatically characterize different configurations of the virtual sensing algorithm regarding Worst Case Execution Time (WCET) and memory usage.

The WCET is measured on the actual target hardware in a Processor-in-the-Loop (PiL) setup [18]. Here, the algorithm is deployed on a target embedded platform and connected to host PC which provides inputs and collects outputs, including measured execution times. The execution time is measured on the target hardware itself using internal, high-resolution timers. Memory usage can refer to both random-access memory (RAM), which is used to store data and intermediate results while the software is running, and read-only memory (ROM), which contains the software itself, but also initial values, precomputed matrices for the model, etc. In this paper memory usage refers to ROM (also known as “flash”) usage. The required amount of (flash) memory is determined from the memory map output by the compiler. As such, this information can be retrieved when the application is built for the embedded target, but does not require access to the actual hardware. Extending the characterization process to also analyze RAM usage is considered for future work.

Figure 5.a shows the WCET for different algorithm configurations on an ARM Cortex-A9 processor running at 667MHz. The horizontal plane corresponds to a sampling frequency of 1024 Hz. As such, configurations below this plane are feasible for real-time deployment while configurations above the plane can only be made feasible by lowering the sampling frequency. This the case for the configuration used in Figure 3.c.

Similarly, Figure 5.b shows the memory usage for different configurations. Note the larger range of window sizes ($N_w$) in this figure. Here, the horizontal plane represents a limit of 1MB (1024KB), which is considered representative for low-cost embedded platforms.

These results show that the WCET is the main limiting factor for real-time embedded deployment of this algorithm as the execution time limit is reached well before the memory limit. However, memory usage can become a problem in certain situations if real-time execution is not required (e.g. for online condition monitoring). In particular, larger models and/or larger window sizes significantly increase the memory requirements.
Figure 5: Measured Worst Case Execution Time (WCET) and memory usage for different configurations. Note the larger scale for $N_w$ in figure b.

7. Sensor switching scheme

The size of the matrix product in the last term of Equation (1) increases with the number of sensors. As demonstrated in Figure 5, this increase in size also results in an increasing computational cost of the MHE optimization problem. In order to improve the estimation performance without increasing the computational cost, one could consider adopting a sensor switching scheme in which two sets of sensors are used on alternating time steps. In terms of hardware, this switching could be achieved relatively easy by using analog multiplexing to switch between the sensor sets at every time step. Alternatively, the switching could also be achieved in software, assuming sufficient channels are available on the analog-to-digital converter.

This scheme increases the spatial resolution of the measurements by using more sensors, while sacrificing some of the temporal resolution since the data from each individual sensor is sampled at half frequency. The effect of this trade-off depends on the dynamic properties of the system.

Figure 6 shows the estimation results achieved using the sensor switching scheme. In Figure 6.a, two sensor sets of 2 strain gauges are alternated. The resulting RMSE is 0.6%, which is similar to the result achieved when all 4 gauges are sampled simultaneously (Figure 3.b), but the WCET is 15% lower. Similarly, the result achieved with two sets of 4 strain gauges (Figure 6.b) match those with a single set of 8 gauges (Figure 3.c) while the WCET is reduced by 23%. This reduction makes the 8-sensor layout feasible for real-time implementation, which was not the case without the sensor switching.

![Figure 6: Tooth force estimation results for two different sensor sets using the sensor switching scheme. All results were obtained with a window size of 15 time steps and a sample frequency of 1024 Hz.](image-url)
8. Conclusions

This paper has introduced a virtual load sensing concept to detect the incoming loads on the first planetary stage of a wind turbine gearbox. The virtual load sensor combines strain measurements on the external surface of the ring gear with a model to indirectly measure the load, avoiding the need for expensive and intrusive direct load measurement. The virtual load sensing algorithm is deployed on low-cost embedded hardware to make a cost-effective load sensing solution. The measured loads can be an input for more effective monitoring, control, and prognostics systems for wind turbine gearboxes. Furthermore, the information can be used to improve the design of future gearboxes.

The performance of the resulting virtual load sensor was tested in a numerical validation for a torque ramp-up scenario. It was found that with 4 strain gauges the virtual load sensor can estimate the loads with a root mean square estimation error of 0.6\%. Adding additional strain gauges did not improve performance. In future research, this analysis will be repeated for additional input load cases corresponding to transient load conditions which are typical in wind turbine drivetrains. Furthermore, the lessons learned from the numerical validation will be used to design an experimental validation campaign. Through this experimental validation, the robustness of the approach will be tested in operating conditions where a mismatch between the model and reality is sure to exist.

An extension of the method towards estimating multiple input loads (e.g., thrust load or bending moment) is also foreseen. To reach this goal, the approach will be extended with additional sensors, more advanced input models, and multibody models (which can also capture effects such as misalignment and out-of-plane loading).

Through the measurement of the worst case execution time and memory usage, it was verified which configurations of this virtual load sensor can be deployed on low-cost embedded hardware for real-time execution. The execution time was found to be the most critical factor for this specific platform. Memory usage could also become critical if larger models and/or window sizes are used and/or if the real-time constraint is dropped (e.g., for online condition monitoring). It was found that configurations with up to 4 strain gauges can run in real-time. Furthermore, configurations with more sensors can be made feasible by adapting the other configuration parameters or by using a sensor switching scheme. This sensor switching scheme reduces the computational requirements of the virtual sensing algorithm without significantly affecting the accuracy.

The results in this paper show the interdependencies between the control and embedded domain, which underlines the importance of approaching the design of the virtual load sensor as a co-design problem. As such, engineers from both domains need to work together to find a good solution, i.e., a performant virtual sensor that can be deployed efficiently. To that end, the characterization of the algorithm regarding critical resource usage (e.g., execution time and memory) on the embedded platform serves as useful feedback to both the control and embedded engineer to determine which configurations are feasible and if not, where trade-offs can be made.

The proposed algorithm can be deployed on embedded platforms within the nacelle, in order to process the sensor signals. This approach offers a local data reduction by extracting the information needed for design and monitoring, thus reducing the amount of data that needs to be stored (e.g., in a cloud-based solution).

Acknowledgements

The Research Fund KU Leuven is gratefully acknowledged for its support. This research was partially supported by Flanders Make, the strategic research center for the manufacturing industry. The authors also gratefully acknowledge ZF Wind Power nv for its support of this research.
References

[1] J Carroll, A McDonald, and D McMillan, 2016, “Failure rate, repair time and unscheduled O&M cost analysis of offshore wind turbines,” Wind Energy, 19, 1107–19.

[2] B Blockmans, J Helsen, F Vanhollebeke, and W Desmet, 2013, “Dynamic Response of a Multi-Megawatt Wind Turbine Drivetrain Under Voltage Dips Using a Coupled Flexible Multibody Approach,” Proc. ASME Power transmission and Gearing Conf. (Portland, Oregon, USA: ASME)

[3] HBM, 2020, “T12HT - Very High-Capacity Torque.” [Online]. Available: https://www.hbm.com/en/6337/t12ht-high-capacity-torque-sensor/. [Accessed: 29-May-2020]

[4] N Perišić, P H Kirkegaard, and B J Pedersen, 2015, “Cost-effective shaft torque observer for condition monitoring of wind turbines,” Wind Energy, 18, 1–19.

[5] B Forrier, F Naets, and W Desmet, 2017 “Broadband load torque estimation in mechatronic powertrains using nonlinear kalman filtering,” IEEE Trans. Ind. Electron., 65(3), 2378–87.

[6] B. Orlik, 2014, “Reduzierung der mechanischen Belastung von Windkraftanlagen durch Messung, Modellierung und Regelung der dynamischen Kräfte (ReDynForce).” [Online]. Available: http://www.fwbi-bremen.de/index.php/windenergie/articles/redynforce.html. [Accessed: 02-Jun-2020]

[7] A Radke and Z Gao, 2006, “A survey of state and disturbance observers for practitioners,”, Proc. American Control Conf., (Minneapolis, Minnesota, USA: IEEE), 5813–88.

[8] F Naets, J Croes, and W Desmet, 2015, “An online coupled state/input/parameter estimation approach for structural dynamics,” Comput. Methods Appl. Mech. Eng., 283, 1167–88.

[9] D Simon, 2006, Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. (Hoboken, New Jersey, USA: Wiley).

[10] C V Rao, J B Rawlings, and D Q Mayne, 2003, “Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations,” IEEE Trans. Automat. Contr., 48(2), 246–58.

[11] M Kirchner, J Croes, F Cosco, and W Desmet, 2018, “Exploiting input sparsity for joint state/input moving horizon estimation,” Mech. Syst. Signal Process., 101, 237–53.

[12] J Croes, 2017, Virtual sensing in mechatronic systems State estimation using system level models (Leuven, Belgium: KU Leuven Faculty of Engineering Science).

[13] International Organization for Standardization, 2007, ISO 21771:2007: Cylindrical involute gears and gear pairs - Concepts and geometry (International Organization for Standardization).

[14] M C Noll, J W Godfrey, R Schelenz, and G Jacobs, 2016, “Analysis of time-domain signals of piezoelectric strain sensors on slow spinning planetary gearboxes,” Mech. Syst. Signal Process., 72–73, 727–44.

[15] J Keller and R Wallen, 2017, Gearbox reliability collaborative phase 3 gearbox 3 test plan (Golden, Colorado, USA: National Renewable Energy Laboratory).

[16] HBM, 2020, “Measurement Uncertainty: Less Errors, Better Results.” [Online]. Available: https://www.hbm.com/en/6021/measurement-uncertainty-experimental-stress-analysis/. [Accessed: 20-Feb-2020].

[17] K Vanherpen et al., 2016, “Ontological reasoning for consistency in the design of cyber-physical systems,” Proc. Int. Workshop on Cyber-Physical Production Systems (CPPS) (Vienna, Austria: IEEE).

[18] H Shokry and M Hinchee, 2009, “Model-Based Verification of Embedded Software,” Computer, 42(4), 53–9.