Model-based virtual sensors by means of Modelica and FMI

M. González^{1,2} O. Salgado¹ J. Croes^{2,3} B. Pluymers^{2,3} W. Desmet^{2,3}

¹IK4-Ikerlan Technology Research Center, Control and Monitoring Area, Spain ²KU Leuven, Department of Mechanical Engineering, Belgium ³Member of Flanders Make, Belgium

Abstract

This paper presents an application case for the estimation of forces using Modelica and the FMI. For that purpose model-based virtual sensors are used. These techniques are presented and the development of the virtual sensor for Modelica and the FMI is discussed. The work has been done in Python where the package pyFMI is used with models exported with the FMI 2.0 for model exchange. The technique is used for the estimation of forces and the friction coefficient in a vertical transportation system. The model of this test bench is explained and the results of the estimation of forces and the friction coefficient are discussed. These estimations provide a valuable tool for the condition monitoring of guiding systems.

Keywords: FMI, virtual sensors, pyFMI, Extended Kalman Filter

1 Introduction

The condition of the guiding system influences significantly the riding quality and performance of transportation systems such as railways or elevators. The proper design and the correct maintenance of the guides is therefore of high importance. Both the design and monitoring of the guiding system require an accurate assessment of the loading condition. However the direct measurement of forces is not feasible, as a dedicated sensor is too costly and intrusive. Virtual sensors are an attractive option to overcome these difficulties.

Virtual sensors process available measurements to estimate other variables of interest that cannot be measured. Mainly two virtual sensor approaches are suggested in the literature: data-driven methods and model-based methods. Data-driven methods use a machine learning perspective to recognize patterns in the behavior of the system. These methods require previous observations of the system in order to learn the different states and conditions of the asset. A review of data driven virtual sensors can be found in (Kadlec et al., 2011). Some common approaches include developing autoregressive models of the system as in (Samara et al., 2013), using artificial neural networks ((Bizon et al., 2014),(Gonzaga et al., 2009)) or using moving window methods as in (Liu et al., 2009). The required data training may be a handicap in systems where data cannot be acquired continuously or in which faulty conditions cannot be measured.

On the other hand model-based methods combine physics-based models and measurements of the system by means of estimation algorithms. The model provides knowledge of the dynamics of the system, which in combination with off-the-shelf sensors can be used to estimate variables of interest otherwise difficult to measure. These approaches are valuable tool in several applications such as control techniques, condition monitoring or model updating (Isermann, 2005).

The performance of these techniques depends on the capability of the model to accurately represent the physics of the system (Isermann, 2005). In addition a great modeling flexibility and simplicity is required to avoid errors and speed up the process. Using Modelica has thus a great added value in the development of model-based virtual sensors. The acausal nature of Modelica allows efficiently modeling heterogeneous systems reusing already developed and tested models. However, it doesn't allow the user to manipulate the solution at each time step, as required by estimation algorithms. In order to use Modelica for state estimation the models have to be exported and manipulated at each time step (Brembeck et al., 2011).

Several modeling environments include model exchange capabilities. However, they are usually developed ad-hoc to interface with one particular tool in a certain context. Therefore they are commonly limited to certain tools and are version dependent. The Functional Mockup Interface (FMI) is a tool independent standard that can efficiently solve this. Furthermore the FMI 2.0 includes some features that aid the development of state estimation algorithms (e.g. directional derivatives).

The combination of Modelica with other programming languages by means of the FMI provides thus a suitable approach for the implementation of model-based virtual sensors. The main focus of the FMI is simulation, but it has already been applied for estimation. For instance in (Brembeck et al., 2011) and (Brembeck et al., 2014) is used to implement nonlinear state observers within Dymola. In (Bonvini et al., 2014) an Unscented Kalman Filter (UKF) is implemented in Python using the FMI 1.0 for model exchange and is used for Fault Detection and Diagnosis. In this paper the FMI 2.0 for model exchange is used to develop an Extended Kalman Filter (EKF) for state and parameter estimation in Python. The suitability of Modelica and FMI for state estimation is tested with a highly nonlinear model which includes events, rotations and friction.

The rest of the paper is organized as follows. Section 2 gives an overview of Model-based virtual sensors and explains the algorithms used in the current application. Section 3 explains how these algorithms are implemented using pyFMI with the FMI 2.0 for Model exchange. Section 4 describes the proposed application case along with the proposed model, a test bench of a vertical transportation system where contact and friction forces are estimated. The results of these estimations are shown in section 5. The final conclusions and the future work are drawn in section 6.

2 Model-based virtual sensor approaches

The core of model-based virtual sensors consists on the use of state estimation algorithms. These algorithms use the difference between the real measurements and the prediction of a physics-based model to correct the output of the model. The most common state estimation algorithms are the Luenberger observer (LO), the sliding mode observer (SMO) and the Bayesian estimators. LO and SMO are simpler to implement than Bayesian estimators but under noisy measurement conditions the Bayesian algorithms are proved to perform better (Zhang et al., 2009),(Esteban et al., 2016). Thus the presented work is focused only on Bayesian estimators.

2.1 Kalman Filter

The Kalman Filter (KF) is the optimal linear estimation filter (Simon, 2006). In the case of Gaussian noise, it provides the maximum a posteriori estimate with the smallest achievable covariance. With non Gaussian noise, it is optimal in giving the minimal mean square error. It is the most widely used Bayesian estimator and has been successfully used in a number of applications (Simon, 2006). The KF uses a linear model defined in state space form as the one shown in equation 1. In the stochastic Bayesian derivation of the KF, both the process and measurement equations are assumed to be disturbed by zero mean white Gaussian noise (w and v in equation 1) of covariance Q and R respectively. The states are assumed to be Gaussian variables with a covariance P and mean the state estimation ($\hat{x} \sim N(\hat{x}, P)$).

$$\dot{x} = f(x, u, t) + w$$

$$y = h(x, u, t) + v$$
(1)

The most common formulation of the KF requires the dynamic system of equation 1 to be described in a discrete form of equation 2, where for a linear model the matrices \mathbf{F} , \mathbf{G} and \mathbf{H} are constant. Generally the discretization of a state space model assumes a zero-order hold for the input u and continuous integration for the noise v. As explained in (Simon, 2006) the discretization involves the computation of the integral of a matrix exponential or any equivalent discretization such as Euler or Runge-Kutta.

$$x_{k} = \mathbf{F}_{k-1} \cdot x_{k-1} + \mathbf{G}_{k-1} \cdot u_{k-1} + w_{k-1}$$

$$y_{k} = \mathbf{H}_{k} \cdot x_{k} + v_{k}$$
(2)

The KF algorithm is shown in figure 1. In each k-time step the system model is evaluated and compared against measured data. This is done in two steps: prediction and update. In the prediction step an a-priori estimation of the states mean and covariance is obtained from the system's model. In the update step this a priori estimation is corrected using the system's output. This estimation process is done recursively: all the prior information is summarized in the initial mean and covariance of each step $(\hat{x}_0^+, \mathbf{P}_0^+)$. Therefore the computational effort in each time step is the same regardless the number of measurements.



Figure 1. Kalman Filter algorithm

Despite being widely used, the KF is limited to linear systems, which also makes the joint estimation of states and parameters not applicable (Simon, 2006). Several suitable extensions of the KF to non-linear systems, such as the EKF or the UKF can be found in (Simon, 2006). As the current system is highly nonlinear (events, rotations) the well known Extended Kalman Filter is used instead.

2.2 Extended Kalman Filter (EKF)

The EKF is the most widely used extension of the KF for nonlinear systems and for the joint estimation of states and parameters. If the model of equation 1 is nonlinear the F_{k-1} and H_{k-1} matrices of equation 2 are no longer constant, but change at each K-step instead. Then the EKF linearizes and discretizes the model around the KF estimate, propagating a linear approximation of the covariance (Simon, 2006). The standard KF shown in figure 1 is then applied at this linearized point. As the estimation is based on the linearization of the system a small step size is required if the system is highly nonlinear. On the other hand, the ease of implementation and the reduced computational cost of the EKF make it an attractive option for the estimation of states in nonlinear systems.

2.3 Parameter identification and Virtual Sensors

State estimation algorithms can be augmented to estimate not only the states of the system but unknown parameters too. Based on the continuous state-space system representation, an augmented version of the system can be obtained if the unknown parameters are included in the states vector (equation 3) and their directional derivatives are included in the system matrices (equation 4). Then a random walk model is used for the unknown parameters: they are assumed to remain constant except for an additive noise (Naets et al., 2015) (equation 5). The discretization of these matrices can later be done in the same way as for the non-augmented model.

$$\mathbf{x}^{aug} = \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix}$$
(3)

$$\mathbf{A}^* = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} & \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$
(4)

$$\dot{\mathbf{p}}(t) = \mathbf{0} + \mathbf{w}_p(t) \tag{5}$$

The joint estimation of parameters makes the system nonlinear. Once defined in the proper way, this augmented vector can be estimated by means of the EKF or any other nonlinear filter.

Once all the states and parameters of the model are known, we can use the model to obtain some other variables of interest (i.e. a virtual sensor). This is a postprocessing step in which the model is evaluated in the estimated set of states, parameters and inputs and the variables of interest are treated as another model output. By means of the estimated state covariance the degree of uncertainty of the virtual sensors can be estimated as well (equation 6).

$$P_{VS} = \frac{\partial f(x,u)}{\partial VS} \cdot P_x \cdot \frac{\partial f(x,u)}{\partial VS}$$
(6)

3 State and parameter estimation with Modelica and FMI 2.0 for model exchange

This section explains the implementation of an Extended Kalman Filter that uses physics-based models developed in Modelica and exported by means of the FMI 2.0. The model used for this work is developed in OpenModelica as it provides a powerful model editor that facilitates the development of models and has the advantage of being an open-source tool. The posterior translation of the Modelica models to FMUs is done by means of JModelica.org. This tool provides full functionality to export models for model exchange with the FMI 2.0, including the possibility of requesting directional derivatives. The possibility of

requesting directional derivatives is particularly useful in the development of the EKF as they are more reliable than numerical derivatives.

There are several FMI libraries aimed at programming languages suitable for the development of state estimation algorithms. In this work pyFMI is used, which has the advantage of being open-source. Thus the presented Extended Kalman Filter is written in Python. In addition to pyFMI, which allows the simulation of FMUs, Python offers several other scientific computing packages that aid the development of custom made algorithms and applications (e.g. Numpy, Scipy, Matplotlib).

To make models compatible with the EKF, the inputs of the filter also have to be defined as inputs in the model, while the measurements of the system have to be defined as model outputs. Estimated parameters are simply defined as parameters, and the newly estimated value of the parameter is set in the model at the beginning of each step. In addition, care must be taken when modeling, so that the states of the model agree with the expected ones during the whole estimation.

As explained in section 2.1, the first step of the Kalman Filter requires the prediction of the model $x_{k+1}^- = f(x_k^+, u_k, k)$. To get this prediction the model is initialized with the states and parameters estimated in the previous step and is simulated from the current step to the next one. In addition to setting the new states and parameters, in models with events these have to be updated after setting states and parameters. To reduce the computational time the results are handled in memory.

For the second step of the filter the EKF requires the matrices of the system in state space form, i.e. the system has to be linearized before it can be used with the estimation algorithm. To achieve this the FMI functionality of obtaining the directional derivatives of the system is used. This function is directly implemented in pyFMI and thus obtaining the system matrices is straightforward:

$A, B, C, D = model.get_state_space_representation()$

The same is not true for the directional derivatives of the parameters, required for the estimation of parameters along with states (section 2.3). As the FMI does not provide directional derivatives for model parameters, these derivatives are computed numerically according to the symmetric difference quotation shown in equation 7.

$$\dot{f}(x) = \frac{f(x+h) - f(x-h)}{2h}\Big|_{h \to 0}$$
 (7)

4 Application for the estimation of forces in guiding systems

The proposed application case is the guiding system of a vertical transportation system. T-shaped guiding rails are used to minimize horizontal motion ensuring travel in a uniform vertical direction (Janovský, 1999). Inappropriate

installation of these guides and their surface roughness are the main causes of vibration in the car frame (Janovskỳ, 1999). These guides are usually composed of several rail segments aligned together. The proper alignment is, however, extremely difficult, and in general out of plane or out of angle misalignment are common. Such deviations increase the contact forces and induce abrupt forces in the car frame at the rail segment joints, reducing the ride quality and efficiency of the system.

The interaction between the car frame and the guiding system is given in four discrete contact points, by means of so-called sliding shoes. These sliding shoes are U-shaped polymeric pads that grab the guide rail's web. The contact forces of the guiding system are applied in these shoes both in x and y axis. Forces in x axis may be in the positive or negative direction, whereas forces in y axis are only directed towards the car-frame. A scaled test bench of a vertical transportation system available at IK4-Ikerlan is used to validate the methodology. This test bench is a useful tool to study the behavior of such system's using sensors not available in real installations. Additionally it allows us to study the effect of defects that we could not put in a real installation. The test bench is a scaled 'rucksack' type rigid car frame, traveling in vertical direction and constrained horizontally by two T-shaped guidingrails (see figure 2). The system has 12 states corresponding to the 6 degrees of freedom of this car frame (x, y, z, z)roll, pitch and yaw). Without loss of generality, the driving force (T) is assumed to be known and acts as a system input. In the scaled test bench under study this force is measured with a load cell attached to the driving cable (see figure 2). The numbering followed in this paper for the four contact points is shown in figure 2. The current system has a maximum travel length of 1.8 meters, a nominal velocity of $0.4\frac{m}{s}$, nominal acceleration of $0.3\frac{m}{s^2}$ and a nominal jerk of $1\frac{m}{3}$

4.1 System's model

Models available in the vertical transportation literature are mainly focused on the assessment of the vertical dynamic of elevators (Isasa, 2010). However vertical dynamics are affected by the friction forces, which are directly related to the rail forces acting on the horizontal plane. Horizontal and vertical dynamics are therefore coupled and should be assessed as a whole. As a first step this paper studies the possibility of using the horizontal dynamics to assess the condition of the guiding system, opening the way to studying the system as a whole. The Modelica Standard Library is used to model the described system. The inertial properties of the cabin given in table 1 are obtained from its CAD model. The contact stiffness can be obtained from classical structural analysis, assuming the guide as a flexible beam with flexible supports. The actual stiffness will thus be a function of the vertical position of the cabin. However, in order to simplify the estimation we use a constant stiffness for the whole guide.



Figure 2. Described system and relevant parameters

 Table 1. Model parameters, positions measured from car's floor coordinate system

PARAMETER	DESCRIPTION	UNITS	VALUE
М	carframe's mass	[Kg]	14.287
I_{11}, I_{22}, I_{33}	Carframe's inertias	$[Kg.m^2]$	0.28, 0.38, 0.20
K_x, K_y	contact stiffness	[N/m]	600000
D_x, D_y	contact damping	[N/(m/s)]	10
ClearanceY	sliding shoe clearance	[m]	0.0
r_{1}^{0}	position of shoe 1	[m]	(-0.085, -0.124, 0.297)
$r_2^{\hat{0}}$	position of shoe 2	[m]	(-0.085, -0.124, 0.067)
$r_3^{\tilde{0}}$	position of shoe 3	[m]	(-0.085, 0.124, 0.297)
r_4^{0}	position of shoe 4	[m]	(-0.085, 0.124, 0.067)
$r_{c,g}^{0}$	carframe's C.G	[m]	(-0.0923, 0.0043, 0.08824)
r_T^0	position of cable	[m]	(-0.085, 0.0, 0.435)

4.1.1 Car frame

Due to the low contact forces and the high stiffness of the car frame, the latest can be modeled as a rigid body. The movement of the body is represented in the coordinate system (C.S) attached to the floor of the car frame, as it is the location where the required sensors are installed. The rotation is constraint far away from the Gimbal lock position due to the guide rails and consequently Quaternion representation is not required. Hence rotations of the car frame are represented using Euler angles (α , β and γ).

4.1.2 Guiding rails: contact and friction model

The sliding shoes are the interface between the car frame and the guiding rails. As such, the forces imposed by the guide rail system on the car frame will be applied via the sliding shoes. The sliding shoes grab the guiding rail's web, contacting it in three flanges. Contact in these three flanges at the same time is highly unlikely and commonly only one or two of the flanges of the sliding shoe are in contact with the rail. From figure 3 it can be seen that movement of the shoe in the x direction will always result in a force opposite to the movement. Thus, for modeling purposes, contact in x axis can be assumed to behave as a spring-damper system. Movement of the sliding shoe in the y axis will depend on the direction of movement. Thus contact in y direction is modeled with a modelica standard ElastoGap model. The direction of the guides web is also taken into account in the actuated prismatic in order to make the model more general.



Figure 3. Model of the guiding rail

In this application high displacements between the sliding shoes and the guiding rails are expected. The friction behavior at small displacements is not relevant and the use of dynamic friction models such as Bouc-wen, LuGre or Dahl is not required. A coulomb friction model is used instead. In order to simplify the mathematics of the state estimation algorithm, instead of using an event driven friction element from the Modelica Standard Library the friction is added with a MSL Multibody World-Force model. The value of this force the absolute value of the contact forces times a user given friction coefficient.

Froz_sup.force =mu_eq *(abs(contactx_sup.f)+abs(contacty_sup.f))*{0, 0, 1};

The advantage of this approach is its simplicity. Avoiding events simplifies greatly the estimation, as the Jacobians of the system change more smoothly. On the other hand, this simplification requires that the direction of the force has to be specified at each simulation, and the direction of the force when the car is stopped is a-priori unknown. Physically the value of the friction coefficient is greater than zero, however we directly include the direction of the force in this parameter. Therefore, a negative value of this parameter just indicates that the direction of the friction force will be negative. With this approach we can find out the direction of the force in the estimation phase.

5 Estimation results

In this section the results of the application of model-based virtual sensors for the evaluation of forces in guiding rails is presented. The estimation approach from section 2.2 is applied to the described system. The measurements used for the EKF are the lateral and vertical accelerations and the vertical position of the car. These measurements are taken with a triaxial piezoelectric accelerometer (lateral acceleration), with a DC response accelerometer (vertical acceleration) and with a draw-wire encoder (cabin position). The accelerometers are located in the coordinate system at the center of the car as depicted in figure 2. Table 2 shows the assumed measurement noise matrix \mathbf{R} (defined in section 2.1).

The measurement of the lateral acceleration provides information of the dynamic change of the contact forces, induced by the roughness and defects of the guiding rails. In addition the model provides information regarding the dynamic behavior of the car and the order of magnitude of the forces. However, the misalignment of the guiding rail results in a DC component of the contact forces which cannot be estimated, as neither the lateral acceleration nor the model have information on that regard. This misalignment affects the vertical dynamics of the system, as friction increases with it. Therefore, we are able to account for this effect within the friction coefficient μ . Additionally, the friction coefficient of each sliding shoe has a significant variability, as it depends on several factors such as, frequency of use of the system, lubrication and temperature. Consequently this parameter is estimated jointly with the states of the system as explained in section 2.3. This parameter contains thus information both on the actual coulomb coefficient and on the misalignment of the guiding system. In contrast to what happens in the actual system, where each sliding shoe has a different friction coefficient, here only one equivalent friction coefficient is assumed for all the contacts (μ_{eq}).

The parameters of the filter's design are shown in table 2. The **P** and **Q** matrices shown in the table include the covariance of the states and unknown parameter of the system. The last term of these matrices is the covariance of the unknown friction coefficient (μ_{eq}). Table 2 also shows the initial value of the states of the system in the following order: cabin.body1.phi[1], cabin.body1.phi[2], cabin.body1.phi[3], cabin.body1.phi d[1], cabin.body1.phi d[2], cabin.body1.phi d[3], cabin.body1.r_0[1], cabin.body1.r_0[2], cabin.body1.r_0[3], cabin.body1.v 0[1], cabin.body1.v_0[2], cabin.body1.v_0[3] and the initial expected value of the friction coefficient mu_{eq} .

In this application the estimated Virtual sensors and comparison variables have a significant noise. For visualization purposes a smoothing has been performed.

Figure 4 shows the parameter estimated along the position of the cabin. The gray area around the estimated μ_{eq} is the 99.7% (3 σ) confidence interval of the value of

PARAMETER	VALUE		
States	$ \begin{array}{l} \alpha \; [rad], \beta \; [rad], \gamma \; [rad], \dot{\alpha} \; [rad/s], \dot{\beta} \; [rad/s], \dot{\gamma} \\ [rad/s], x \; [m], y \; [m], z \; [m], \dot{x} \; [m/s], \dot{y} \; [m/s], \dot{z} \; [m/s] \end{array} $		
Measurements	$z [m], \ddot{x} [m/s^2], \ddot{y} [m/s^2], \ddot{z} [m/s^2]$		
Р	diag([1e-09 , 1e-09 , 1e-09 , 2e-09 , 2e-09 , 2e-09 , 1e-09 ,		
Q	diag([0e+00 , 0e+00 , 0e+00 , 1e-03 , 1e-03 , 1e-03 , 0e+00 , 0e+00 , 0e+00 , 1e-05 , 1e-05 , 1e-05 , 5.0])		
R	diag([1e-08 , 1e-16 , 1e-16 , 1e-16])		
x_initial	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0		
$F_s [Hz]$	1652		
$\mu_{eq0} \ [-]$	0.2		

Table 2. Design parameters of the filter

this parameter. This confidence interval comes from the co-variance estimated for this parameter. As explained in section 4, this parameter includes information regarding the actual friction coefficient of coulomb and regarding the misalignment of the guiding rails. The negative sign of the estimated parameter is not related to the physical meaning of the friction coefficient, but to the direction of the friction force instead. In addition to the friction coeffi-



Figure 4. Estimated friction coefficient and the estimated 99.7% confidence interval of the estimation

cient, the contact forces also provide information regarding the condition of the guiding system. More exactly, they provide information regarding sharp differences in the position of the guide rails. Such dynamic changes will be due to sharp changes of guiding rail segment or imperfections on the rail. These defects induce abrupt forces in the sliding shoes that have a negative impact on the riding quality. Figure 5 shows the estimated contact forces in X and Y directions along with the 99.7% confidence interval (light gray for X direction and light blue for Y direction. As expected, the variance of the estimated forces is relatively high. This is mainly because the model and the measurements do not provide information regarding the DC component of this forces.



Figure 5. Estimated contact forces. In black the forces in X direction and 99.7% confidence interval, in blue forces in Y direction and 99.7% confidence interval.

Finally figure 6 shows the comparison between the estimated friction force and the measured one. The estimated friction force is computed from the sum of the friction forces in each sliding shoe as explained in section 4.1.2. The measured one is the direct subtraction of the cable's tension and cabin acceleration.



Figure 6. Comparison of estimated and measured friction force

The guiding system is not the most common source of failure but it is one of the most critical systems. Failure in guiding rails leads to large down times of the whole system and it is difficult to evaluate its condition. It is complicated thus to assess both the alignment and the smoothness of the guides, as required to ensure comfort and energy efficiency. Different condition monitoring alternatives are suggested in the literature for guiding rails. In general vibration data processing is used to assess the condition of the guides (Wan et al., 2015). However these methods require well trained data which in may be difficult to obtain. Model-based virtual sensors, on the other hand, provide a suitable approach to monitor the condition of system using off-the-shelf sensors and without training data. The trade off is that an accurate model of the system is required. Nevertheless it was shown that Modelica and the FMI simplify the development of model-based virtual sensors.

The presented estimations provide a useful indicator of the condition of the guiding system. Changes in the friction coefficient indicate misalignment and abrupt changes in the forces indicate local damages in the guides. For instance the estimation from figure 4 shows a significant deviation in the friction coefficient when the cabin is at 0.6m. This deviation indicates thus a misalignment of the guides at that position. Additionally, the abrupt change of the contact forces at 0.9m indicates a local defect in the web of the rail. However, further work is required to develop a methodology to set a threshold for the value of these variables. Once the threshold is defined we can use this to assess the condition of the system, aiding the alignment of the guides and finding early damage in the rails.

6 Conclusions

An EKF with parameter identification capabilities has been developed in Python using the package pyFMI and models exported with the FMI 2.0 for model exchange. Modelica and FMI are very useful to cope with the complexities arising from the use of Model-based virtual sensor with complex systems. The combination of these tools reduces modeling effort and simplifies the implementation of the virtual sensor. As an example of the efficiency of this combination the estimation of forces in a vertical transportation system scaled test bench is presented. The EKF is used to simultaneously estimate states and parameters in a scaled vertical transportation system test bench. Additionally the forces acting on the guiding system are estimated. This estimations provide a mean to assess the condition of the guiding system. This approach opens the way to condition based maintenance strategies for guiding systems. Such maintenance schemes can improve riding quality, safety and efficiency of vertical transportation systems, fulfilling thus the requirements of modern smart systems. Future steps in the investigation include:

- Assessment of the estimated variables and parameters: a theoretical optimal value of the friction coefficient and of the contact forces should be used to set a threshold that aids assessing the condition of the guides.
- In this work the tension in the cable has been used as input of the system. Even though off-the-shelf sen-

sors exist for that purpose it would be better to use just sensors available in the system or easier to use, such as the input of the controller and the currents in the machine. To achieve this the model of the system has to be extended to include not only the cabin of the test bench, but the controller, the electric machine, the pulley, the cable and the counterweight as well. The model will be extended to include all the parts of the system. The electric machine and control systems will be included in the estimation.

• As the system grows in complexity, the use of other state estimation algorithms such as the Unscented Kalman Filter or the Moving Horizon Estimator will be explored. Finally the estimation will be used to monitor the condition of the system.

7 Acknowledgments

The authors gratefully acknowledge the European Commission for its support of the Marie-Sklodowska Curie program through the ITN ANTARES project (GA 606817) and the support from the KU Leuven research fund.

References

- K. Bizon, G. Continillo, S. Lombardi, E. Mancaruso, and B.M. Vaglieco. Ann-based virtual sensor for on-line prediction of in-cylinder pressure in a diesel engine. In 24th European Symposium on Computer Aided Process Engineering, volume 33 of Computer Aided Chemical Engineering, pages 763 – 768. Elsevier, 2014. doi:http://dx.doi.org/10.1016/B978-0-444-63456-6.50128-9.
- M. Bonvini, M. Wetter, and M. Sohn. An fmi-based framework for state and parameter estimation. In *Proceedings of the 10 th International Modelica Conference; March 10-12; 2014; Lund; Sweden*, number 096, pages 647–656. Linköping University Electronic Press, 2014.
- J. Brembeck, M. Otter, and D. Zimmer. Nonlinear observers based on the functional mockup interface with applications to electric vehicles. In *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical University; Dresden; Germany*, number 63, pages 474–483. Linköping University Electronic Press, 2011.
- J. Brembeck, A. Pfeiffer, M. Fleps-Dezasse, M. Otter, K. Wernersson, and H. Elmqvist. Nonlinear state estimation with an extended fmi 2.0 co-simulation interface. In *Proceedings of the 10th International Modelica Conference-Lund, Sweden-Mar 10-12, 2014*, volume 96, pages 53–62. Linköping University Electronic Press, 2014.
- E. Esteban, O. Salgado, A. Iturrospe, and I. Isasa. Model-based approach for elevator performance estimation. *Mechanical Systems and Signal Processing*, 68-69:125 – 137, 2016. ISSN 0888-3270. doi:http://dx.doi.org/10.1016/j.ymssp.2015.07.005. URL http://www.sciencedirect.com/science/ article/pii/S0888327015003246.

- JCB Gonzaga, L.A.C. Meleiro, C Kiang, and R. Maciel Filho. Ann-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Computers* & *Chemical Engineering*, 33(1):43–49, 2009.
- I. Isasa. *Model validation applied to locally nonlinear lift structures.* PhD thesis, Mondragon Unibertsitatea, 2010.
- R. Isermann. Model-based fault-detection and diagnosis status and applications. Annual Reviews in Control, 29(1):71 85, 2005. ISSN 1367-5788. doi:http://dx.doi.org/10.1016/j.arcontrol.2004.12.002. URL //www.sciencedirect.com/science/article/ pii/S1367578805000052.
- L. Janovský. *Elevator mechanical design*. Elevator World Inc, 1999.
- P. Kadlec, R. Grbić, and B. Gabrys. Review of adaptation mechanisms for data-driven soft sensors. *Computers & chemical engineering*, 35(1):1–24, 2011.
- Xueqin Liu, Uwe Kruger, Tim Littler, Lei Xie, and Shuqing Wang. Moving window kernel pca for adaptive monitoring of nonlinear processes. *Chemometrics and intelligent laboratory systems*, 96(2):132–143, 2009.
- F. Naets, J. Croes, and W. Desmet. An online coupled state/input/parameter estimation approach for structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 283:1167 – 1188, 2015. ISSN 0045-7825. doi:http://dx.doi.org/10.1016/j.cma.2014.08.010. URL http://www.sciencedirect.com/science/ article/pii/S0045782514002795.
- P. Samara, J. Sakellariou, G. Fouskitakis, J. Hios, and S. Fassois. Aircraft virtual sensor design via a time-dependent functional pooling narx methodology. *Aerospace Science and Technol*ogy, 29(1):114–124, 2013.
- D. Simon. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons, 2006.
- Z. Wan, S. Yi, K. Li, R. Tao, M. Gou, X. Li, and S. Guo. Diagnosis of elevator faults with ls-svm based on optimization by k-cv. *Journal of Electrical and Computer Engineering*, 2015: 70, 2015.
- Y. Zhang, Z. Zhao, T. Lu, L. Yuan, W. Xu, and J. Zhu. A comparative study of luenberger observer, sliding mode observer and extended kalman filter for sensorless vector control of induction motor drives. In 2009 IEEE Energy Conversion Congress and Exposition, pages 2466–2473. IEEE, 2009.