

Composable Modelling for a Hybrid Gearbox

Joshua Sutherland¹ Rui Gao²

¹Department of System Innovation, The University of Tokyo, Japan, joshua@m.sys.t.u-tokyo.ac.jp

²Modelon KK, Tokyo, Japan rui.gao@modelon.com

Abstract

Gearbox modelling is an important topic to the simulation community due to its difficulty in representing mixed continuous and discrete behaviour. Such kind of models also usually couple the loss of power that is dependent on the angular velocity and the load. However, one single gearbox model with a certain fidelity cannot satisfy the broader needs of modelling during product development. In this paper, we will address hybrid modelling issues and take the gearbox model as an example to illustrate how to model gearboxes from a Systems Engineering perspective. A supervisor and supervised model structure is investigated that could also be regarded as a use case to a general cyber-physical modelling approach. The resultant model achieves the same output as the LossyGear model in Modelica Standard Library.

Keywords: *Hybrid model, Gearbox, Loss power, Cyber-physical*

1 Introduction

Modelica is well accepted as the multi-domain modelling and simulation standard and its acausal model features enable efficient representation of cyber-physical systems. One of the important applications is to use it for simulation-based design in product development to reduce the use of the real prototypes. However, this makes it difficult for engineers who have experience in experimental based product development or who have deep knowledge in high fidelity simulation technologies such as Finite Element Method (FEM) to understand to what extent Modelica can capture and reproduce the physical phenomena. Such engineers tend to compare the accuracy of Modelica model with more traditional FEM models.

Any model of physics including a Modelica model is no more than a projection of the real-world behaviour in terms of a design or analysis purpose. Referring to the physical phenomena, it is difficult (or even impossible) to represent it exhaustively by one single model. However, such an exhaustive model would be undesirable as a good model is one with the simplest form that satisfies the certain design or analysis purpose.

As such, defining the adaptable scope of the model becomes a crucial task to model based design.

The hybrid modelling features in Modelica is enabled in the Modelica language by the synchronous data flow principle (known as hybrid (Elmqvist, Mattsson, & Otter, 2001; Otter, Elmqvist, & Mattsson, 1999)). It is important because these principles help to define the boundary of the physics-based model. Enabling the physics-based model to be compartmentalised by a discrete mode model.

Further, Modelica provides mode transition, finite state machine functionality by Modelica text and libraries including: StateGraph, StateGraph2 and Synchronous. With the former two being Dymola specific extensions, and the latter aiming to supersede them both as part of Modelica Language 3.3 (Elmqvist, Gaucher, Matsson, & Dupont, 2012). These modelling technologies broaden the cyber-physical modelling capability of the Modelica.

Given any model that is developed for use, either for design or analysis purposes, is the outcome of human activity with a specific objective. To capture the development requirements and setup the specification is one of the most important starting points in modelling and simulation-based design.

Take the gearbox example, the simplest IdealGear model represents the basic functionality and captures the most important requirements of a gearbox. However, the gearbox designer may also need to consider the gearbox performance with for example power loss that is dependent on the rotational velocity and load level. Such kind of functional/performance requirements need a high-fidelity model.

1.1 Past Research

Given the importance of gearboxes to many different machines significant amounts of effort have gone into creating models of them. One review paper (Parey & Tandon, 2003) presents 63 different references to spur gear dynamic models alone. Regarding Modelica modelling of gearboxes two models stand out for further investigation: Modelica Standard Library (MSL) LossyGear presented in (Pelchen, Schweiger, & Otter,

2002) and a composed gearbox model presented in (Schlegel, Hösl, & Diel, 2009).

LossyGear is a highly popular model of a single gear interaction, which successfully models not only load- and speed-dependent energy losses (associated with mesh and bearing friction) while the gears are moving, but also the stiction affects which can lead to the gears becoming stuck. To model stiction required the usage of the hybrid modelling described previously. However, all the phenomena of the model are lumped into a single model which makes it difficult to separate phenomena or modify. A representation of LossyGear is presented in Figure 1.

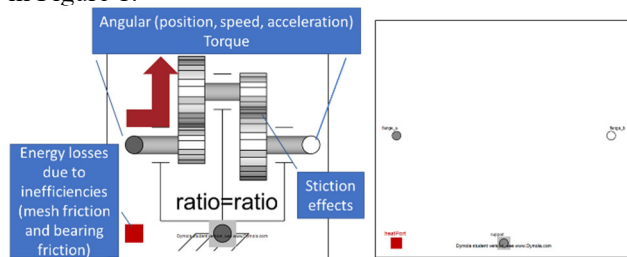


Figure 1. LossyGear. **Left side:** Icon view, annotated with the key functions of the model. **Right side:** Diagram view, showing the lack of composition of other model blocks.

In (Schlegel et al., 2009) a complex automotive gearbox model is presented composed of many different components. The focus of this model is to represent the steady state behaviour of the gearbox including energy losses. However, by focusing on the steady state work was not done on ensuring it works appropriately including the times the gearbox is stuck (what LossyGear does so well).

It seems that there is a gap between models which are highly composable but cannot capture the discontinuous behaviour of when the gear is stuck (Schlegel et al., 2009) and models which have very accurate discontinuous behaviour but lack an emphasis on composition of a large number of components (Pelchen et al., 2002).

1.2 The Focus of this Paper

This paper aims to address the gap described in the previous section and construct gearbox models which are both highly composable but also model the discontinuities associated with being stuck.

We believe a focus on composability can drive a more function-based design where the specific behaviours required of the subsystem should be identified and then created by the composition of specific components in the subsystem, which then leads to greater innovation (there is opportunity to compose highly novel new systems). Ideally such an approach would be able to do the following:

- Quickly build subsystem models with sufficient fidelity for the larger system model's purpose.

- Provide insight into the performance of the individual components which make up the subsystem such that meaningful insight can be drawn on why the subsystem is performing as it should be.

While ensuring the resulting subsystem model satisfies the needs of the system model in which the subsystem is to be embedded.

There are many phenomena worth investigation regarding gearboxes (e.g. fatigue, vibration, and defects) discontinuities associated with being stuck and fictional energy losses are to remain the focus of this paper.

Clearly the approach any individual modeler takes to building a model of a particular subsystem will result in different models. Our basic assumption is that each model is more or less successful at fulfilling its purpose and such variability is undesirable. However, by defining an approach and applying it to a Gearbox example it is our aim to share potential best practices across the community.

To accomplish our aim the paper is arranged as follows:

- Section 2: The problem formulation of gearbox modelling with the different design requirements is explored.
- Section 3: Operational mode modelling using the StateGraph2 and other methods is presented.
- Section 4: The composable modelling of gearbox is shown and comparison to the existing MSL LossyGear model is demonstrated.
- Section 5: Conclusions and further work are presented.

2 Gearboxes Modelling

2.1 Elegant Modelling

How “good” any model is, is a highly subjective matter that is dependent on the context in which the model will be constructed, verified, validated, used and modified. However, it is possible to draw out various quality measures which we believe any model maker would at least have a passing interest in. In the context of Systems Engineering the term “elegant” has been applied to systems which are considered holistically better than others (Griffin, 2010). With elegant designs being characterised as answering a yes to the questions on the left-hand side of Table 1. For this paper's purposes, we characterise those questions as shown on the right side of Table 1. We believe being able to create models consistently exhibiting the listed characteristics would be highly valuable.

Table 1. Characterization of elegant models.

<i>Elegant question:</i>	<i>Characterization for modelling</i>
Does it work?	Model provides sufficient fidelity and parametrization for its purpose. Stakeholders have sufficient confidence in the model.
Is it robust?	It can be used in contexts outside of those it was originally intended, or it fails gracefully.
Is it efficient?	Time resources needed to create the model, simulate and maintain it are not wasted.

2.2 Composing Subsystems from Components: The Preferred Solution?

A great strength of Modelica is its hierarchical object-oriented nature which enables the composition of models out of component models. Enabling the benefits recognised from the software industry of complexity management and facilitating code reuse.

Given a physical gearbox is created by the composition of individual components (e.g. bearings and interlocking gears) and then imbedding in a larger more complex system (e.g. a car) to deliver some desirable behaviour (e.g. torque change, speed change, energy losses and stiction effects) it is a good example for reviewing the described problem.

2.3 Modelling Purposefully

As described in the introduction (Section 1) we acknowledge that any model of physics including a Modelica model is no more than a projection of the real-world behaviour in terms of the design or analysis purpose. Further, it is difficult (or even impossible) to represent all potential physical phenomena exhaustively. As such we assert that one must take a highly purposeful approach when creating models.

Given that the language associated with describing the purpose is often ambiguous, we refer to the glossary of the Systems Engineering Body of Knowledge (Adcock (EIC), 2017) to obtain definitions which we subsequently use to describe the gearbox model.

2.3.1 Purpose

Defining purpose to simply be “What the system is for...” (Blockley & Godfrey, 2000) the modeller must also be clear of the potentially large difference between the purpose of system model compared to the system itself. As shown in Table 2 the purpose of a model is often to investigate negative aspects of the system such as energy loss. Clearly the purpose of a real gearbox is not to lose energy.

Table 2. Some example purposes of a gearbox and a gearbox model.

<i>System:</i>	<i>Some example purposes:</i>
Gearbox	<ul style="list-style-type: none"> • Change angular speed of a rotating shaft. • Change the torque being transferred on a rotating shaft. • Make a profit for a gearbox manufacturing company.
Gearbox model	<ul style="list-style-type: none"> • Investigate potential gear ratios which will work well with the system the gearbox is to be imbedded in. • Investigate potential energy losses from the gearbox. • Investigate potential longevity of the gearbox. • Make a profit for a gearbox modelling company.

2.3.2 Requirements

Given the definition of a requirement to be a “Statement that identifies a product ... characteristic ... necessary for product ... acceptability” (ISO/IEC, 2007), requirements are a way to state an intention of a system such that it might fulfil its purpose. Given that the purposes of the system and model are different it then follows that the requirements would also be different.

2.3.3 Behaviour

Taking the definition of systems behaviour from (Ackoff, 1971) to be “a change which leads to events in itself or other systems. Thus, action, reaction or response may constitute behaviour in some cases.” given this paper is concerned with dynamic modelling in Modelica the capturing of behaviour is incredibly important.

However, given our previous acknowledgement that a model cannot represent all phenomena and the model’s behaviour will represent a subset of the physical gearboxes total behaviour.

2.3.4 Functions and Functional Requirements

Defining a function to be “An action, a task, or an activity performed to achieve a desired outcome.” (Hitchins, 2008) it is possible to then create a definition for Functional Requirements being: An action, a task, or an activity performed to achieve a desired outcome necessary for product acceptability.

Given the purpose of a dynamic Modelica model is generally to investigate and predict the behaviour of a system which has yet to be built the most important functional requirements for the model are around modelling a subset of the behaviours of the physical system.

2.4 Functional Requirements Break-Down: An Example of a Gearbox Model

In this section we present some functional requirements for a gearbox model such that in later sections we can compose a new model.

2.4.1 Torque and motion conversion functional requirements

The basic and most important functionality of the gearbox is to convert the torque and motion by following underlying equations 1 and 2. As such any model interested in the behaviours of the rotating shafts will need to model these.

$$\varphi_a = ratio \cdot \varphi_b \quad 1$$

$$0 = ratio \cdot flange_a \tau + flange_b \tau \quad 2$$

2.4.2 Loss power functional requirements

In the real world the contact of parts (gear teeth and bearings) results in the loss of power. If the purpose of the model is to investigate such losses such behaviour must be a functional requirement.

It should be noted in the previous section that the functional requirements of a model are not the same of the system being built; i.e. stating that a model must be capable of simulating energy loss is not saying that energy loss is desirable in the physical system. Instead it is stating that the model must have a representation of this physical reality.

As per LossyGear (Pelchen et al., 2002) such losses are generally modelled as a speed dependent load torque (bearing friction) and a speed dependent efficiency (mesh friction). Both of which manifest as heat.

2.4.3 Operational mode functional requirements

Even without friction a gearbox can be in the states of moving forward, backward or be stopped. If friction is modelled closely to that of the real world it a non-linear force. It exhibits stuck and sliding behaviour (stiction) where to get two surfaces to move past each other a sufficient amount of initial force must be provided.

When applied to a gearbox it can be appropriate to think of the gearbox as having several discrete states in which it can be in at any time. The states and state transitions associated with a gearbox are presented in Figure 2. These states are important as they enable the capture of additional functional requirements of the gearbox model by which to describe the different operational modes or use cases of the model.

The number of transitions shown in Figure 2 make it somewhat complex however it is important to explicitly model all the transitions possible to be made such that an appropriate modelling approach can be taken.

2.4.4 Combined functional requirement of a gearbox model

The best practice is generally to start from a simple model and increasingly add functions to it (but also complexity).

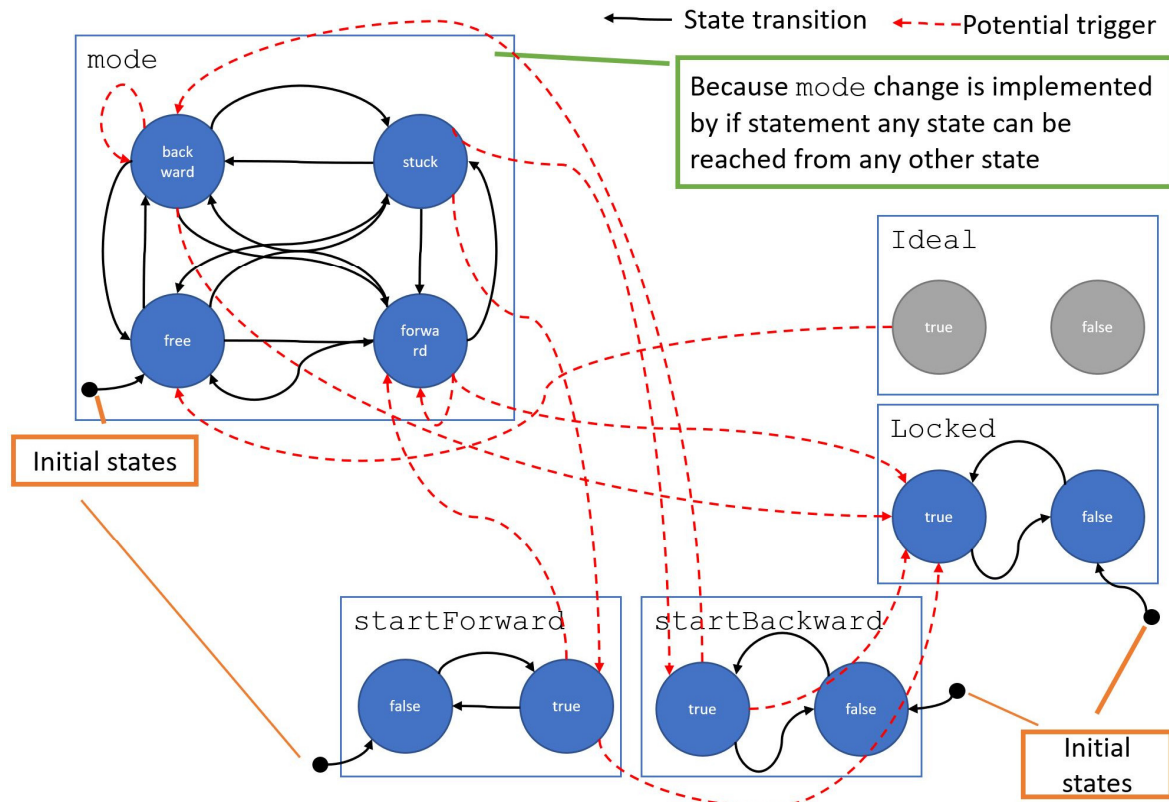


Figure 2. State and state transition representation of gearbox for depicting operational modes.

For our case of the gearbox a typical system engineer is most likely to start modelling from idealgear and then add the loss power functional requirements, then they will finally come to the real operational mode design. This is usually referred as the functional requirement that defines the fidelity of the targeted model.

Based on the investigation, the main functionality of the gearbox model is presented in Figure 3 which along with Figure 2 act as the functional requirements for the gearbox model.

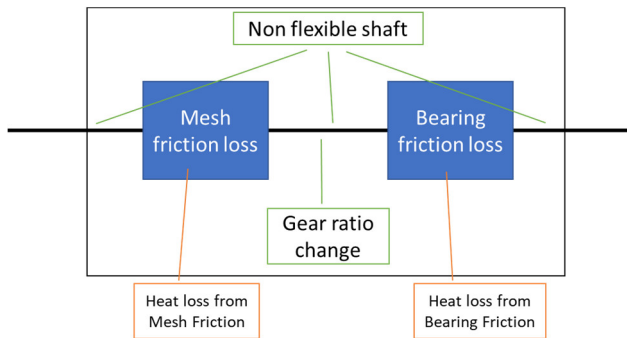


Figure 3. Describing the key aspects of a gearbox.

3 Operational Mode Modelling

Dymola has discrete event modelling capability including libraries that handle Petri Nets e.g. StateGraph library. We use the more sophisticated library StateGraph2 to implement the operational mode transition modelling presented in Figure 2. Modelica 3.3 has been shown to support a state machine approach by way of the Synchronous feature (Elmqvist et al., 2012) but currently there is no graphical library ready to be used. Given the focus of this paper is implementation of a graphical approach Synchronous was not investigated further.

We term the control logic which handles the operational mode of the model a supervisor. The gearbox model who is controlled by the supervisor is known as the supervised. This is depicted in Figure 4.

In the rest of this section we present various alternative implementations of the supervisor and evaluate how successful they are.

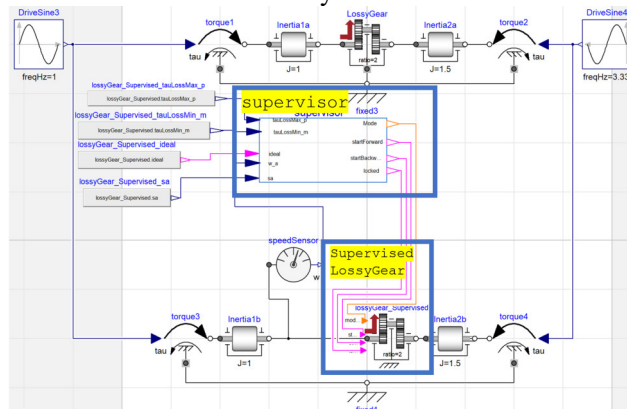


Figure 4. Supervisor for Supervised LossyGear to handle operational mode modelling.

3.1 Supervisory Control using StateGraph2

StateGraph2 is presented in (Otter, Malmheden, Elmqvist, Matsson, & Johnsson, 2009) to create “hierarchical state machines in combination with any Modelica model”. Given the state representation we present in Figure 2 it was felt that this would offer a promising option for clearly modelling the operational modes of the gearbox.

Two supervisor models are presented Figure 5 (implementing the state transitions of Figure 2 as explicitly as possible) and Figure 6 (implementing the state transitions of Figure 2 where each state is provided two state steps).

The first model, Figure 5, does succeed in a creating a control architecture which explicitly shows all the possible transitions between the states. However due to the very large density in the transitions of mode (top left of Figure 2) it does not make the model clear. Further the model fails due to chattering.

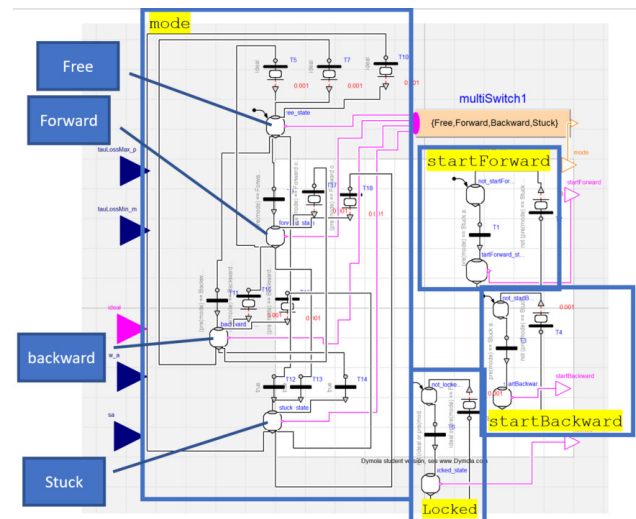


Figure 5. Implementing the state transitions of Figure 2 as explicitly as possible. Fails due to chattering.

The second model, Figure 6, takes a different approach where the mode states are modelled in a similar way to the outer edge states of Figure 2. This method is less preferred as it would make it possible for conflicting states to be held at the same time (e.g. forward and backward) if the control logic was not implemented correctly to prevent it.

Through this studies we have found that the StateGraph2 models in Figure 5 and Figure 6 capture the functional requirements depicted as states and state transitions in Figure 2 quite straightforwardly, because they can represent the structure of the state transition diagram well. Further, the simulation results of the gear behaviour using the model depicted in Figure 6 follow similar trends to those of the LossyGear model. However, this approach fails to achieve identical results to that of the reference target LossyGear model. This is

regarded as the limitation to the StateGraph2 perhaps based on a timing issue.

Based on the limited success of using StateGraph2 decided to investigated alternative ways of supervisory modelling.

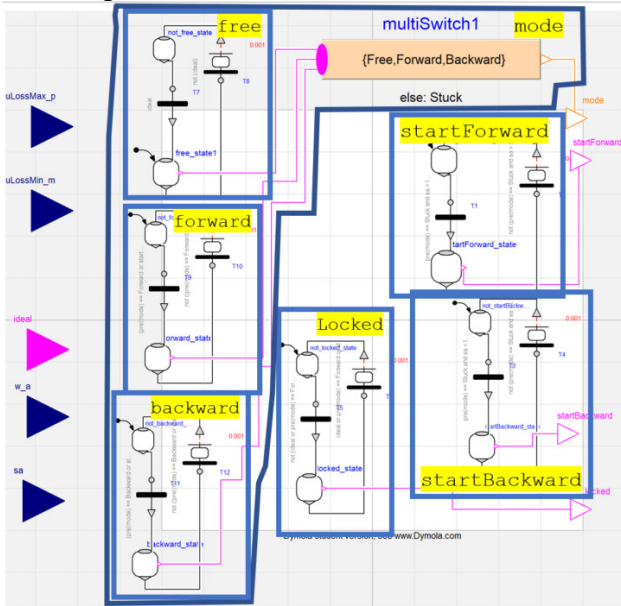


Figure 6. Implementing the state transitions of Figure 2 where each state is provided two state steps. Fails to produce identical results to the original LossyGear model.

3.2 Supervisory Control by Logical Modelling

Alternative methods for implementing the supervisor (depicted in Figure 7 and Figure 8) involve explicitly modelling the supervisory controller are described in this section. Both of which were deemed highly successful as both were able to produce identical results to the reference LossyGear model (see Figure 9).

equation

```
startForward = pre(mode) == Stuck and sa > tauLossMax_p/unitTorque
              or initial() and w_a > 0;

startBackward = pre(mode) == Stuck and sa < tauLossMin_m/unitTorque
              or initial() and w_a < 0;

locked = not (ideal or pre(mode) == Forward or startF
             or pre(mode) == Backward or startBackward);

mode =
  if ideal then Free
  else
    (if (pre(mode) == Forward or startForward) and w_a > 0
     then Forward
     else if (pre(mode) == Backward or startBackward) and w_a < 0
     then Backward
     else Stuck);
```

Figure 7. Equations implementing the supervisor.

Figure 7 shows the equations used to determine the state of the gearbox based on various physical phenomena as described in (Pelchen et al., 2002) while Figure 8 depicts the same equations implemented using Modelica.Blocks.Logical.

Given the success of these approaches they are used for the rest of this paper. However, the equation-based approach lacks the visual depiction of state available with the StateGraph2 approach.

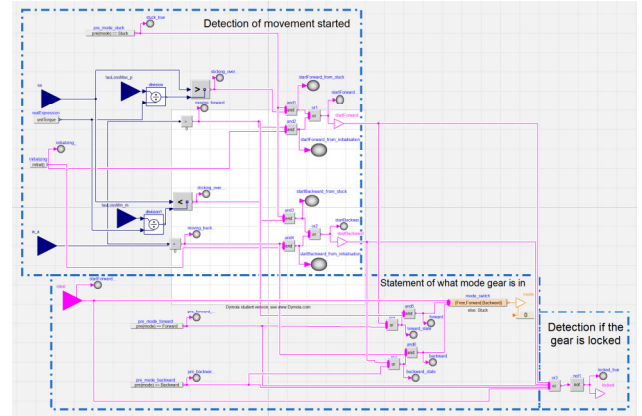


Figure 8. Implementing the supervisor using components from the library Modelica.Blocks.Logical.

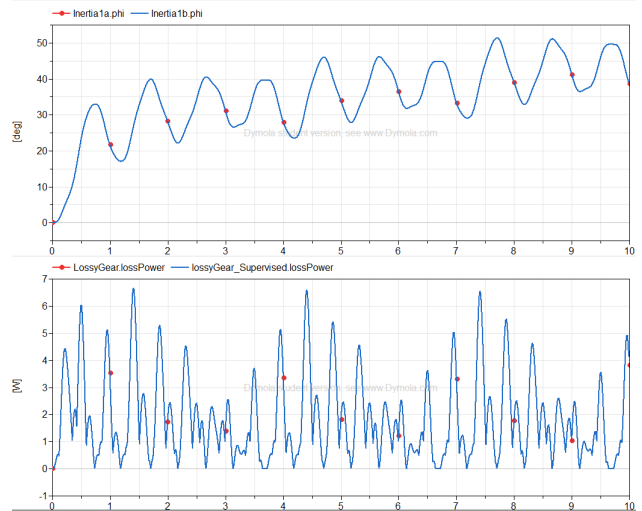


Figure 9. Comparing results of the supervisor/supervised LossyGear combination and the original model. **Top:** Position. **Bottom:** Loss power.

Now, we have achieved modelling of functionality, now it is time to combine the design requirements into the model.

4 Composable Modelling of the Gearbox

This section describes the general approach we have developed for composing a gearbox based on the functionality described in Section 2 which lead to the creation of the key aspects of the gearbox in Figure 3 and the operational mode approach presented in Section 3.

4.1 Phenomena Decomposition

As shown in Figure 3, two of the primary functional components are the two sources of energy loss, the mesh friction loss and the bearing friction loss.

The paper (Pelchen et al., 2002) provides a very detailed description of LossyGear where speed dependent energy losses are parameterised in the form of mesh efficiency and bearing friction. **Error! Reference source not found.** depicts how the key equations representing these in the reference LossyGear

model can be separated from one another to build separate models of mesh and bearing friction. The separation then offers more insight these components.

LossyGear: Separating Bearing Friction and Mesh Friction

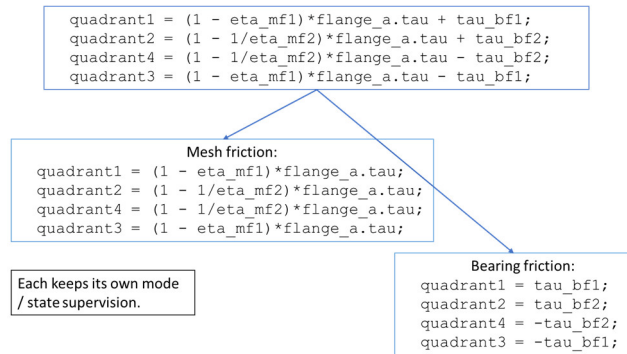


Figure 10. Modifying the equations of LossyGear to only focus on mesh friction or bearing friction.

4.2 Independent Power Losses and Independent Supervisor

In this section, we extend the study such that separate supervisors are investigated to handle the operational modes of the mesh and bearing frictions.

The benefit of such composable method enables the model user to switch on/off the supervisor to make the component run free with no friction. This is helpful to identify the location of the energy loss and improve the design e.g. if the loss occurs at bearing, then lubricating the bearing should be considered.

4.2.1 Separate supervised components

Figure 11 shows a composed gearbox model of Mesh Loss, Bearing Loss and with separate supervision for each of those components. This simulation produces results (shown in Figure 12) identical to a LossyGear model, but now with the benefit that bearing and mesh friction are separated such that their associated losses can be interrogated separately and the heat associated with them can be directed to different locations. Hence, the work is considered a success given we have created a composed gearbox model with identical behaviour to that of the reference LossyGear model.

It should be noted that the order of the components and to which the ratio is applied matters for producing identical results to the reference LossyGear model. In that Mesh Friction component should have a ratio of 1 and Bearing Friction a ratio equal to what the gear is to represent.

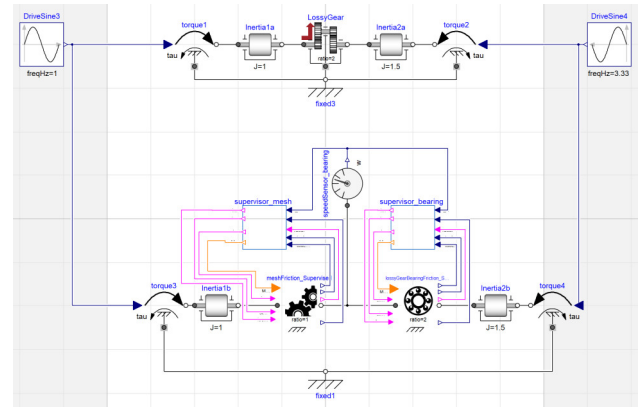


Figure 11. Composed gearbox model of mesh loss, bearing loss and with separate supervision for each of those components.

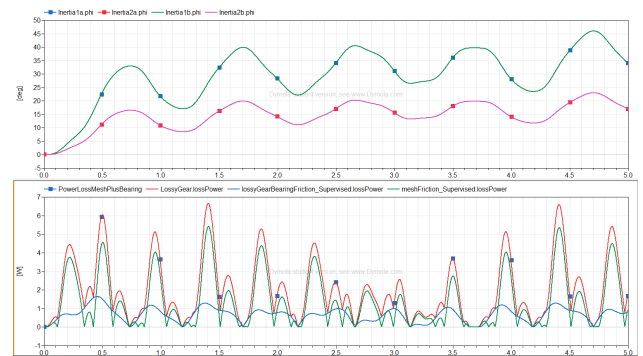


Figure 12. Verifying the composed gearbox (Figure 11) compared to a LossyGear model. **Top:** Position. **Bottom:** Loss power.

4.3 Application of the Combined Model

Given the composable nature of the new components it is possible to insert additional components into the LossyGear model. This is demonstrated in Figure 13 where a spring damper is added between the components. As shown in Figure 14 such an approach can enable the modelling of flexibility on the shaft resulting in different speeds between the components.

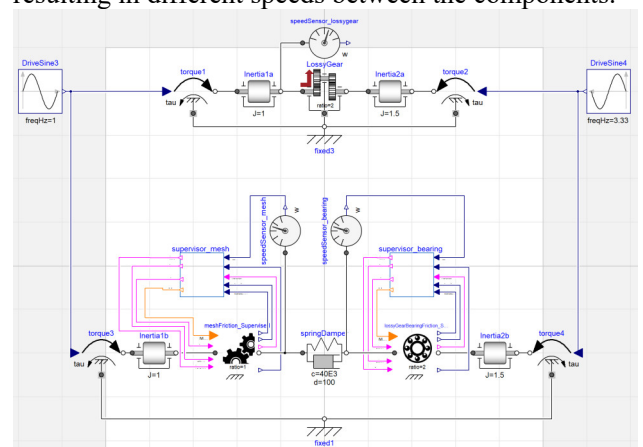


Figure 13. Adding a spring damper between the mesh friction and bearing friction.

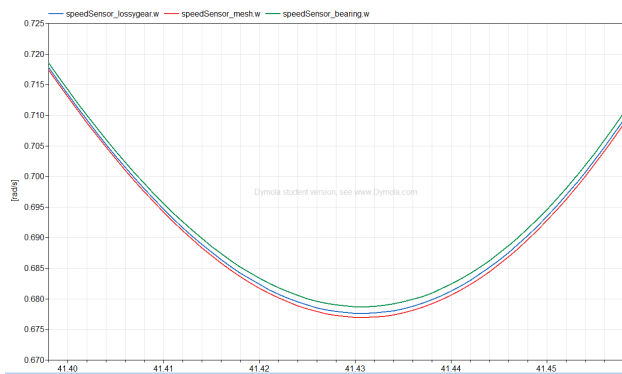


Figure 14. Comparison of shaft speeds of the model shown in Figure 13. With LossyGear compared to a composed model of mesh and bearing friction with an additional spring damper.

5 Conclusions and Further Work

This paper set out to develop and use a composable approach to create cyber-physical models driven by the requirements. We take the gearboxes design as an example to illustrate the workflow of building models with different fidelity. In particular:

- Models should be composed based on the functional requirements of the component.
- The supervisor part is designed explicitly according to the functional requirements using StateGraph2, logical blocks and Modelica text.
- While the physics part (the supervised) is kept independent but interacting with the supervisor.
- The proposed structure increases the freedom to switch on/off the mode transition functionality at the mesh and bearing parts independently, which enables the detailed analysis of the performance of individual components.

The resultant model achieves the same result as the reference LossyGear model.

5.1 Further Work

Further work to be done includes extending the approach to a general cyber-physical modelling method and applying the approach to another domain (such as electrical).

Investigation into application of Modelica 3.3 Synchronous is also worth consideration.

References

- Ackoff, R. L. (1971). Towards a system of systems concepts. *Management Science*, 17(11), 661–671.
- Adcock (EIC), R. D. (2017). *The Guide to the Systems Engineering Body of Knowledge (SEBoK) v. 1.9*. Stevens Institute of Technology Systems Engineering Research Center, the International Council on Systems Engineering, and the Institute of Electrical and Electronics Engineers Computer Society. Retrieved from [http://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_\(SEBoK\)](http://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK))
- Blockley, D. I., & Godfrey, P. (2000). *Doing it differently: Systems for rethinking construction*. Thomas Telford.
- Elmqvist, H., Gaucher, F., Matsson, S. E., & Dupont, F. (2012). State machines in modelica. In *Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany* (pp. 37–46). Linköping University Electronic Press. Retrieved from <http://www.ep.liu.se/ecp/076/003/ecp12076003.pdf>
- Elmqvist, H., Mattsson, S. E., & Otter, M. (2001). Object-oriented and hybrid modeling in modelica. *Journal Européen Des Systèmes Automatisés*, 35(4), 395–404.
- Griffin, M. D. (2010). How do we fix System Engineering? In *61st Annual International Congress, Prague, Czech Republic* (Vol. 27). International Astronautical Federation (IAF).
- Hitchins, D. K. (2008). *Systems engineering: a 21st century systems methodology*. John Wiley & Sons.
- ISO/IEC. (2007). *ISO/IEC 42010: 2007-Systems and software engineering—Recommended practice for architectural description of software-intensive systems*. Technical report, ISO.
- Otter, M., Elmqvist, H., & Mattsson, S. E. (1999). Hybrid modeling in Modelica based on the synchronous data flow principle. In *Computer Aided Control System Design, 1999. Proceedings of the 1999 IEEE International Symposium on* (pp. 151–157). IEEE. Retrieved from <http://ieeexplore.ieee.org/abstract/document/808640/>
- Otter, M., Malmheden, M., Elmqvist, H., Mattsson, S. E., & Johnsson, C. (2009). A new formalism for modeling of reactive and hybrid systems. In *Proceedings of the 7th International Modelica Conference; Como; Italy; 20-22 September 2009* (pp. 364–377). Linköping University Electronic Press.
- Parey, A., & Tandon, N. (2003). Spur gear dynamic models including defects: A review. *The Shock and Vibration Digest*, 35(6), 465–478.
- Pelchen, C., Schweiger, C., & Otter, M. (2002). Modeling and simulating the efficiency of gearboxes and of planetary gearboxes. In *Proceedings of 2nd International Modelica Conference* (pp. 257–266). Retrieved from <http://elib.dlr.de/11828>
- Schlegel, C., Hösl, A., & Diel, S. (2009). Detailed loss modelling of vehicle gearboxes. In *Proceedings of the 7th International Modelica Conference; Como; Italy; 20-22 September 2009* (pp. 434–443). Linköping University Electronic Press. Retrieved from <http://www.ep.liu.se/ecp/article.asp?issue=043&article=48>