

Modeling of a Front-engine, Rear-wheel Drive Automobile Drivetrain

1058385

Abstract

In automotive design, drivetrain of an automobile refers to the elements of the automobile that are responsible for producing power and transmitting the torque to the driving wheels. These elements include the engine, the gearbox and the driveshaft. The place and role of a drivetrain is determined by the automobile layout. With respect to the driving wheels and the location of the engine, there are a few distinct categories. Some of these categories are front-engine, front-wheel drive (FF); mid-engine, front-wheel drive (MF); front-engine, rear-wheel drive (FR); rear-engine, rear-wheel drive (RR); rear-engine, front-wheel drive (RF) and four wheel drive (4WD). In modern automobiles, the engine is an internal combustion engine. Internal combustion engine means that the gas-air mixture is ignited inside the engine to produce power. There are variants of this type of engine, however the most widely used ones have cylindrical vessels and pistons. The fuel is ignited in these vessels called cylinders and force the pistons to move up and down. This vertical (or in some designs, horizontal) motion of the pistons are then converted to a circular motion which is transmitted to the wheels. If the engine is close to the driving wheels such as in front-wheel drive (FF) layouts, this transmission is relatively simple. However, if the driving wheels are far from the engine such as in front-engine, rear-wheel drive (FR) layout, a component called “driveshaft” is needed. The driveshaft is a rotating rod that transmits the motion to the parts that are connected to the wheels. In this project, some components of a front-engine, rear-wheel drive automobile are modeled and the dependencies within the system are studied.

1 Introduction

“Empirical modeling refers to any kind of (computer) modelling based on empirical observations rather than on mathematically describable relationships of the system modelled.”^[1] The main programming language and environment for empirical modeling is Eden, whereas the main implementation of the Eden language is tkeden.

Programming for empirical modeling is somewhat different than other programming paradigms like procedural programming and object-oriented programming. Although the use of procedural statements is allowed in Eden, the power of Eden comes from the ability to establish dependencies between observables. Most real world phenomena have many inner dependencies. So by developing a model for a real world system with underlying dependencies preserved, it is possible to study these phenomena by manipulating data and observing the effects. A classical, procedural approach to model a real world system is impractical, if not impossible; because it would be the programmer's responsibility to propagate the effects of any kind of input to the system such as modifying a parameter or inserting/removing data. Empirical modeling, on the other hand, provides powerful tools to interact with the system and make observations. The implementation details of how the internal dependencies are preserved are hidden from the programmer and the user.

2 Model

The model in this project consists of an upper view of the drivetrain of a front-engine, rear-wheel drive automobile. The visible parts of the system include the engine,

gearbox, driveshaft, differential, rods and wheels. The actual model with its elements is illustrated in Figure 1.

2.1 Dependencies

Empirical modeling makes heavy use of and takes advantage of dependencies. Dependencies in this model can be classified into two groups: positional dependencies and functional dependencies.

2.1.1 Positional Dependencies

Positional dependencies determine the positions (coordinates) of the elements in the system. In a real automobile, some parts are adjacent to others and their relative positions are fixed, i.e. as the engine goes forward (relative to the ground), so does the gearbox. In the model, the coordinates of each element depends on another element, rather than having a fixed value¹.

2.1.2 Functional Dependencies

Functional dependencies determine how individual elements interact with each other. For example, the wheels are functionally dependent on the axles. They can only turn when the axles turn. Similarly, the driveshaft is functionally dependent on the gearbox, because when the clutch (not shown in the model) disengages, the torque of the engine cannot be transmitted to the driveshaft. Ultimately, every item depends on the engine functionally, because when the engine is off, no matter

1 The only exception to this is the engine, whose coordinates are hard-coded.

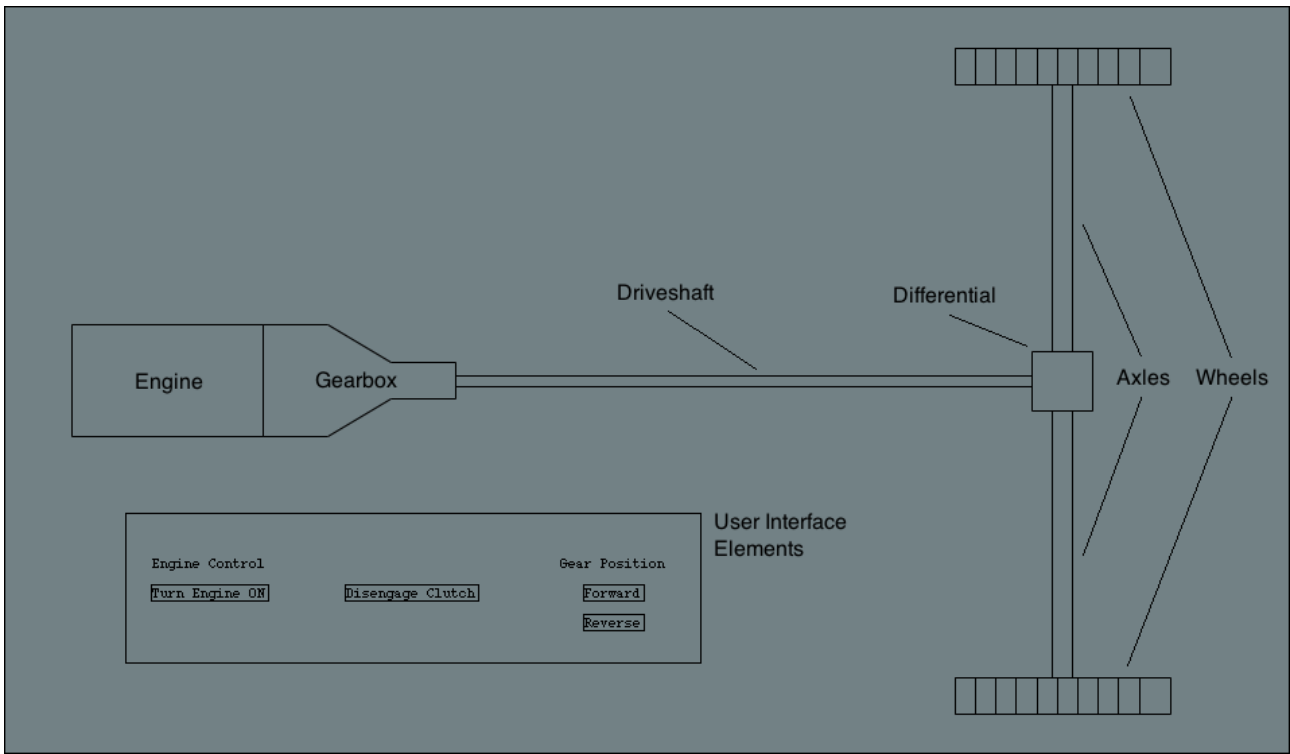


Figure 1. Illustration of the model and its elements.

what state the other elements are in, there is no motion. The states of the elements and the overall result (turning of the wheels) are tabulated in Table 1.

Engine	Clutch	Gear Pos.	Wheels
On	Engaged	Forward	Forward
On	Engaged	Reverse	Reverse
On	Disengaged	Forward	No Motion
On	Disengaged	Reverse	No Motion
Off	Engaged	Forward	No Motion
Off	Engaged	Reverse	No Motion
Off	Disengaged	Forward	No Motion
Off	Disengaged	Reverse	No Motion

Table 1. Functional dependencies among elements.

2.2 Implementation

Technologies used in the development of the project are Eden, DoNaLD and SCOUT. Eden is the main development tool. It is used for handling user input and timing. DoNaLD is used for drawing the 2D objects (automobile parts in our model) on the screen. Finally, SCOUT is used for creating multiple views, displaying messages on the screen and drawing buttons which can be used to interact with the system. Of course, empirical

modeling, *per se*, does not rely on a graphical user interface to realize interaction between the user and the model. This is true for this model as well. It is always possible (and recommended) to interact with the model by changing the definitions (either by changing the source code or using the command-line interface). However, the use of a simple user interface consisting of buttons makes interaction simpler (especially for those who are not familiar with tkeden, Eden or empirical modeling in general).

2.3 Evaluation

The model is based on a system of dependencies depicted in Table 1. This means that, when the value of an observable changes, other observables will be affected accordingly. Thus, the model is successful in modeling (a small subset of) a real world phenomenon, an automobile to be precise, preserving dependencies of the working mechanism. To make this clearer, some visual aid is used in the model. For example, in order to make it easier to see the functional dependency between the gearbox and the driveshaft visually, the driveshaft is drawn disconnected from the gearbox when the clutch is disengaged. This state is illustrated in Figure 2. In this state, the driveshaft is disconnected from the gearbox, so the torque cannot be transmitted to the wheels. When the clutch is engaged, it is drawn connected to the gearbox, linking wheels to the engine. This state is illustrated in Figure 3. The movement of the wheels are also depicted by the movement of some small lines that represent tire

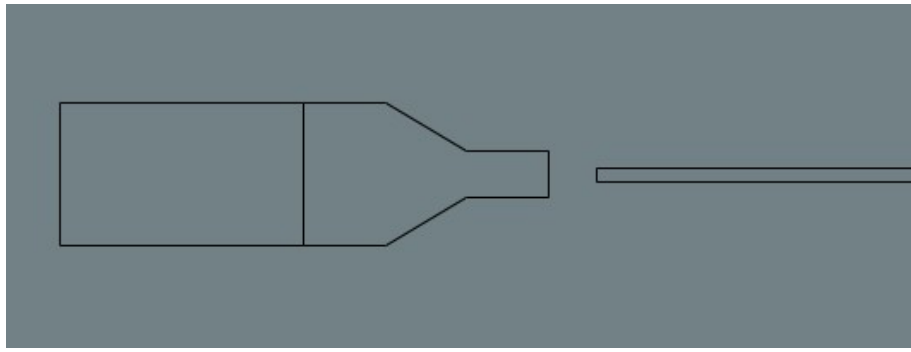


Figure 2. Driveshaft shown disconnected from the gearbox when clutch is disengaged.

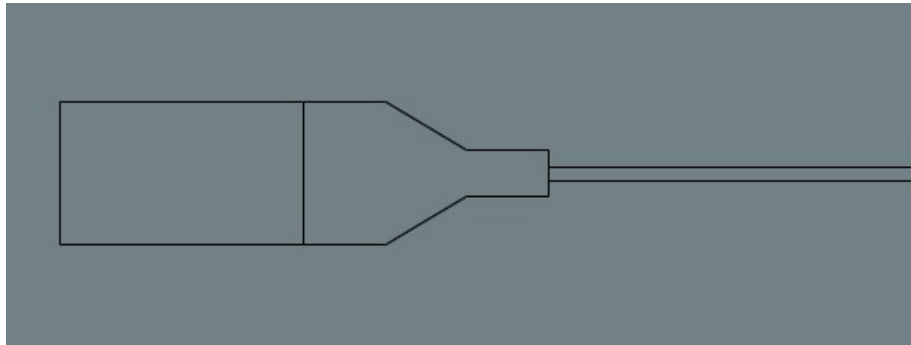


Figure 3. Driveshaft shown connected to the gearbox when clutch is engaged.

treads. These small visual aids are not very sophisticated, but they serve the purpose of making the interconnections easier to grasp and visualize. There are two reasons for them being simple. Firstly, they are intended to highlight dependencies between data, so a simple depiction is better. Secondly, drawing capacities of DoNaLD are limited.

2.4 Release Notes

This model was developed and tested on tkeden v.1-75. There might be backward and/or forward compatibility issues when executed using a different version of tkeden.

3 Conclusion

This paper is intended to be a complementary resource to the actual model described. It gives insights into the model which was built to form an association between a real world object and a computer model. Some characteristics of the real world object have been reflected onto the model and an empirical study of the model has been presented.

References

[1] http://en.wikipedia.org/wiki/Empirical_modelling