

# Tire Pressure Monitoring

DHBW Automotive Software Engineering – Stuttgart, © Kai Pinnow 2019

## 1. Introduction and Overview

The project work focuses on tire pressure monitoring. Detailed requirements below.

As depicted, the following equations govern the motion for a left turn:

$$(1) \quad R_{RR} = W + R_{RL}$$

$$(2) \quad R_{FR}^2 = B^2 + R_{RR}^2$$

$$(3) \quad R_{FL}^2 = B^2 + R_{RL}^2$$

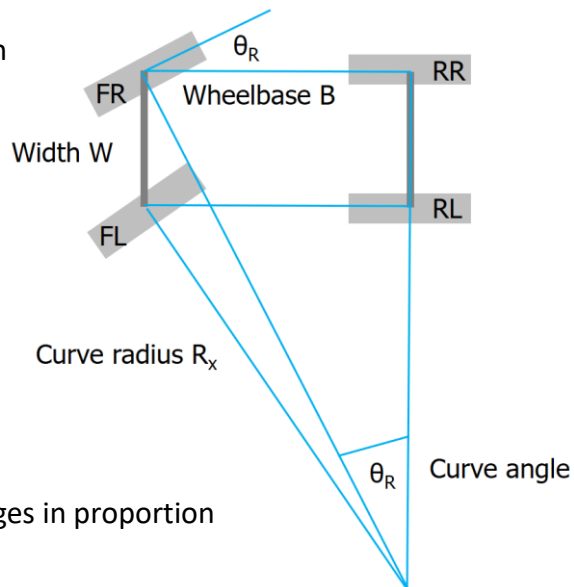
$$(4) \quad B = R_{RR} \tan \theta_R$$

$$(5) \quad B = R_{RL} \tan \theta_L$$

Please note that any pair of velocities changes in proportion to the related curve radiuses

$$(6) \quad V_x / V_y = R_x / R_y \quad \text{for any } x, y \text{ from } \{RR, RL, FR, FL\}$$

A drop in tire pressure of a single wheel results in an increase of the respective velocity of that wheel since it will be a little smaller then.



## 2. Requirements and Deliverables

Please refer to "ID" column in your documentation. A \* denotes extra tasks / requirements.

ID	Check
R1	A tire pressure monitor allows observation of the four wheel speeds to detect an unexpected imbalance for a vehicle with sizes $B=1.53$ m and $W=2.65$ m.
R2	An imbalance of a wheel speed of 0.5 % of one of the wheels concerning the expected consistent wheel speeds will be regarded as an indication of a tire pressure drop.
R3	Detecting a tire pressure drop, some warning lamp shall be switched on and some "SOS" (three times short, three times long, three times short) sound shall appear (base rate 0.8 seconds).
R4*	The system shall allow "re-calibration" after inflation.
R5	Design the solution mainly with appropriate graphical modeling elements (i.e. block diagrams and/or state machines) or with scripts or ESDL and document all your decisions, reasoning, and results clearly with screenshots and text.

D1	Plan all necessary tasks based on three point estimates and monitor progress according to below requirements.
D2	Use the provided example data "curve.mat" to calculate, display, and analyze curve radiuses for selected situations. It contains the wheel speeds (vfl, vfr, vrl, vrr) in [km/h] and the steering wheel signal sw (without direction) in [degree] with time base tv in [s], plus the corresponding lateral acceleration q in [g] (with different time base tq again in [s]).
D3	Create a Simulink model that calculates the driving distance for each wheel and analyze the provided "curve.mat" data in this regard. Remember to analyze and document settings. Are there imbalances according to requirement R2?
D4	Set-up a simple tire pressure monitor in Simulink that detects a deviation according to requirement R1 and R2 by observing driving distances of the individual wheels for straight driving i.e. driving without curves.
D5	Code, configure, and apply a simple "linear congruential" random number generator like $X(i) = (a * X(i-1) + c) \bmod m$ with suitable parameters a, c, and m to test the tire pressure monitoring without the provided "curve.mat" data.
D6	Execute some system tests in Simulink with the number generator from D6 to check the tire pressure monitoring function feasibility.
D7	Transfer the tire pressure monitoring function to ASCET.
D8	Provide unit tests for all designed tire pressure monitoring components.
D9	Design a warning function according to requirement R3.
D10	Provide the random number generator designed in D5 in ASCET with unit tests.
D11	Create a system test with the aid of the number generator and some error model i.e. simulating some pressure drop over a certain time to demonstrate the tire pressure monitoring.
D12*	Think about the way to calibrate the system by means of requirement R4. Which parts of the implementation shall change in order to support such a feature? How long does one need to drive for calibration?
D13*	Shall the analysis incorporate curve driving or just analyze segments driving straight?
D14*	Reflect: Which other observations or comments are in place concerning the model, the requirements, the prescribed functions, or your solution, the testing, and the selected graphical approach.

# 1 Aufgabe D2

## 1.1 Berechnung der Radien

Im folgenden werden die in der Aufgabenstellung gegebenen Gleichungen verwendet, um die Kurvenradien der einzelnen Räder zu berechnen.

Mit (6) folgt:  $\frac{R_{RR}}{R_{RL}} = \frac{V_{RR}}{V_{RL}}$

$$R_{RL} = R_{RR} * \frac{V_{RR}}{V_{RL}}$$

Mit (1) folgt:  $R_{RL} = \frac{V_{RR}}{V_{RL}} * (W + R_{RL})$

$$R_{RL} - \frac{V_{RR}}{V_{RL}} * R_{RL} = \frac{V_{RR}}{V_{RL}} * W$$

$$R_{RL} * (1 - \frac{V_{RR}}{V_{RL}}) = \frac{V_{RR}}{V_{RL}} * W$$

$$R_{RL} = \frac{\frac{V_{RR}}{V_{RL}} * W}{1 - \frac{V_{RR}}{V_{RL}}}$$

$$R_{RL} = \frac{W}{\frac{V_{RR}}{V_{RL}} - 1}$$

Mit (1) folgt:  $R_{RR} = W + R_{RL} = W + \frac{W}{\frac{V_{RR}}{V_{RL}} - 1}$

Mit (2) folgt:  $R_{FR}^2 = B^2 + (W + \frac{W}{\frac{V_{RR}}{V_{RL}} - 1})^2$

$$R_{FR} = \sqrt{B^2 + (W + \frac{W}{\frac{V_{RR}}{V_{RL}} - 1})^2}$$

Mit (3) folgt:  $R_{FR}^2 = B^2 + (\frac{W}{\frac{V_{RR}}{V_{RL}} - 1})^2$

$$R_{FR} = \sqrt{B^2 + (\frac{W}{\frac{V_{RR}}{V_{RL}} - 1})^2}$$

Die erstellten Gleichungen stellen die Grundlage für die folgenden Analysen dar.

## 1.2 Analyse der ausgewählten Situationen

Im Folgenden wird anhand einiger Graphiken das Verhalten bei Kurvenfahrten analysiert.

In der nachfolgenden [Abbildung 1.1](#) werden zuerst die Geschwindigkeiten der vier Räder des Fahrzeugs dargestellt. Es ist erkennbar, dass es Abweichungen zwischen den Geschwindigkeiten gibt. Diese werden nachfolgend genauer erläutern.

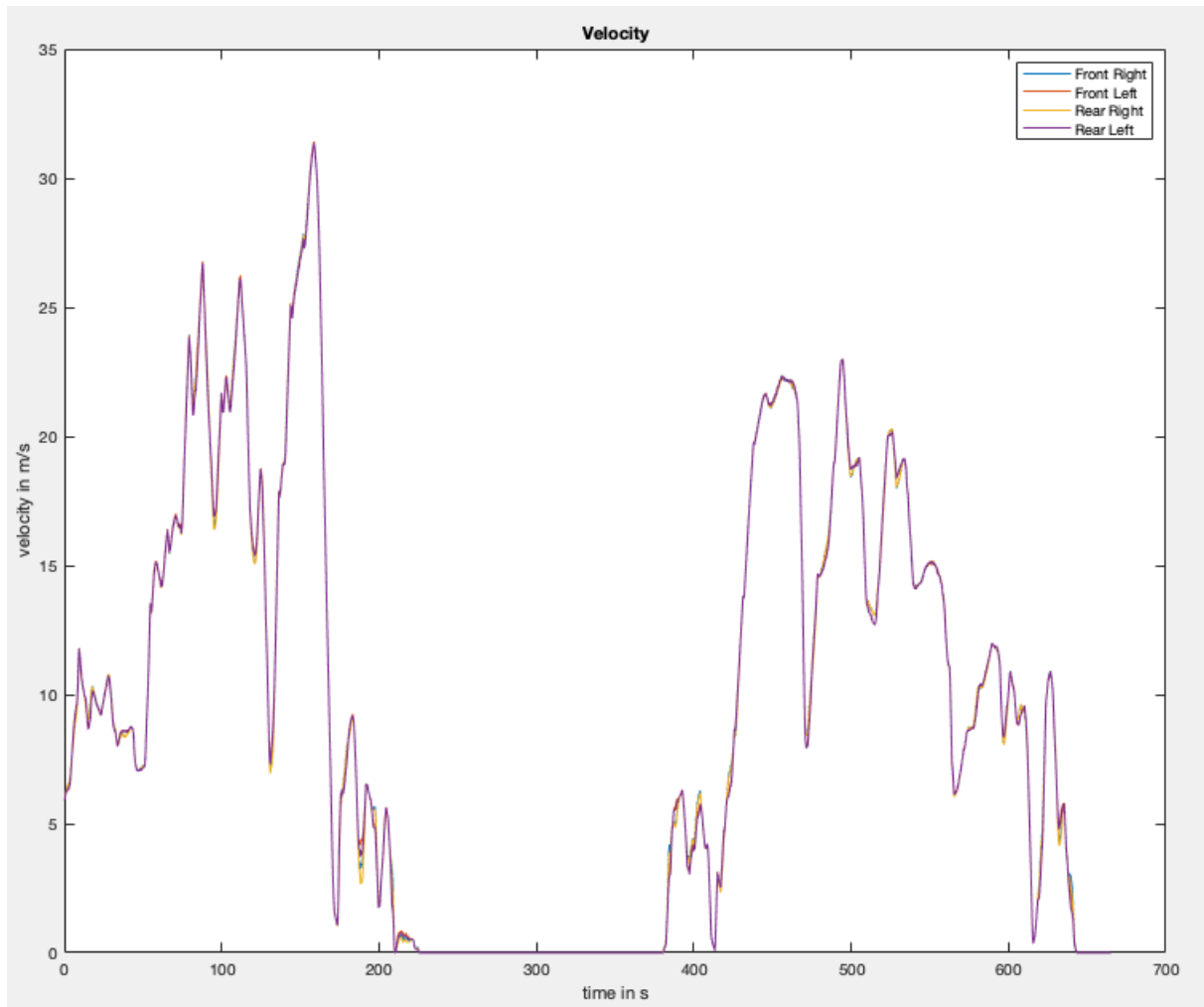


Abbildung 1.1: Übersicht der Geschwindigkeiten der vier Räder

Die [Abbildung 1.2](#) zeigt einen Ausschnitt zwischen 495s und 550s. Die Abbildung wurde in einzelne Abschnitte zur Analyse unterteilt. Im ersten Abschnitt, vor 500ms, ist der Lenkwinkel sehr klein. Man sieht dort kaum einen Unterschied zwischen den Geschwindigkeiten der linken und rechten Räder. Im zweiten Abschnitt ist der Lenkwinkel wesentlich höher. Das hat zur Folge, dass die Geschwindigkeit der rechten Räder über der Geschwindigkeit der linken Räder liegt. Beim Wechsel vom zweiten in den dritten Abschnitt ist der Winkel kurz Null und steigt dann wieder. Dort liegt die Geschwindigkeit der linken Räder über der Geschwindigkeit der rechten Räder. Daraus lässt sich schließen, dass sich die Fahrt von einer Links- in eine Rechtskurve ändert. Im vorletzten Abschnitt ist wieder eine Linkskurve erkennbar. Im letzten Abschnitt wird kaum gelenkt, aus diesem Grund weichen die Geschwindigkeiten nur sehr gering voneinander ab.

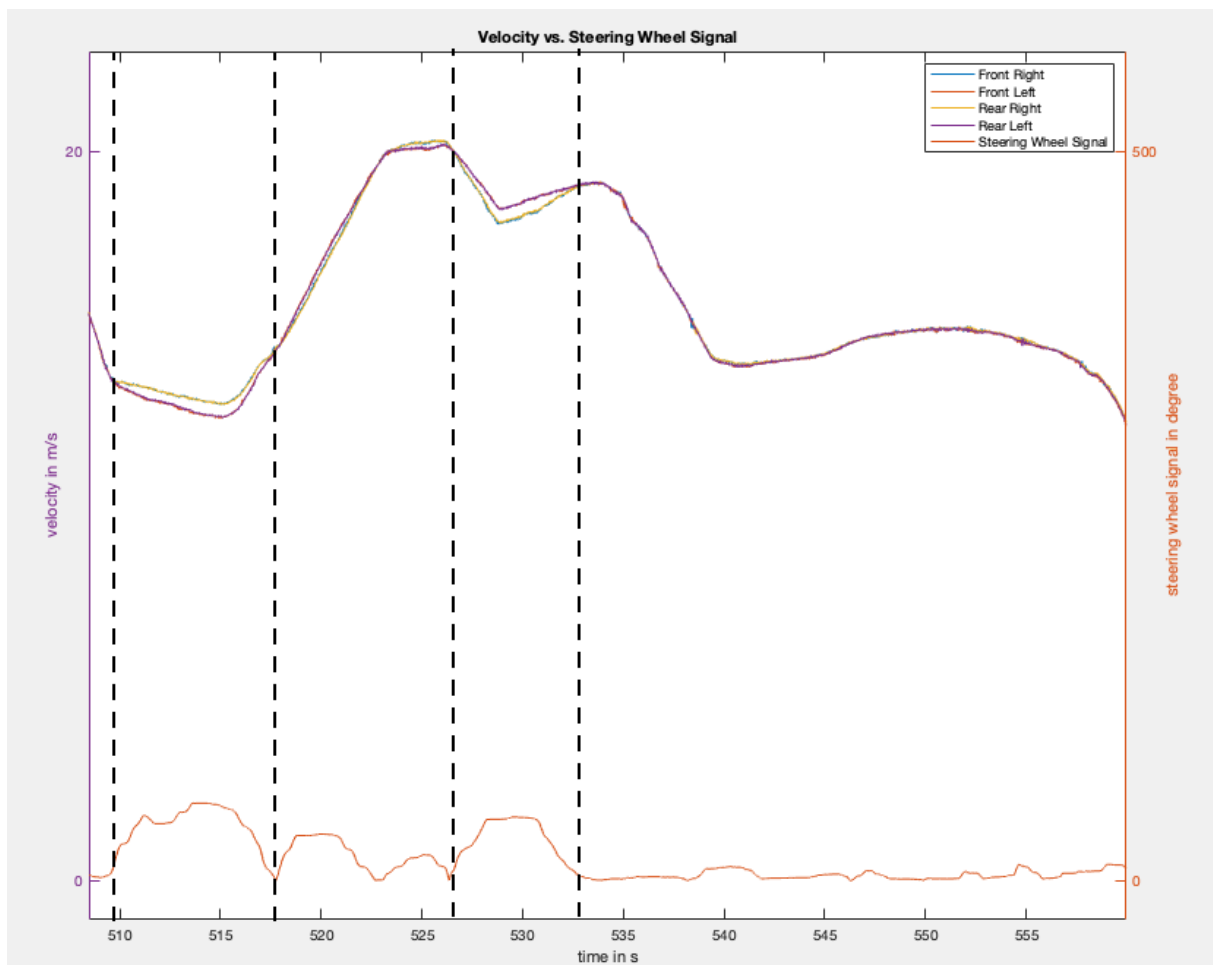


Abbildung 1.2: Übersicht der Geschwindigkeiten mit Drehwinkel des Lenkrads

Die [Abbildung 1.3](#) zeigt den Ausschnitt von ca. 510ms bis 517ms. Dieser entspricht dem dem zweiten Abschnitt der [Abbildung 1.2](#). Hier werden die Radien der beiden Vorderräder verglichen. Es ist erkennbar, dass der Radius des Rechten Vorderrads immer über dem Radius des Linken liegt. Dies zeigt nochmal, dass es sich hier um eine Linkskurve handelt.

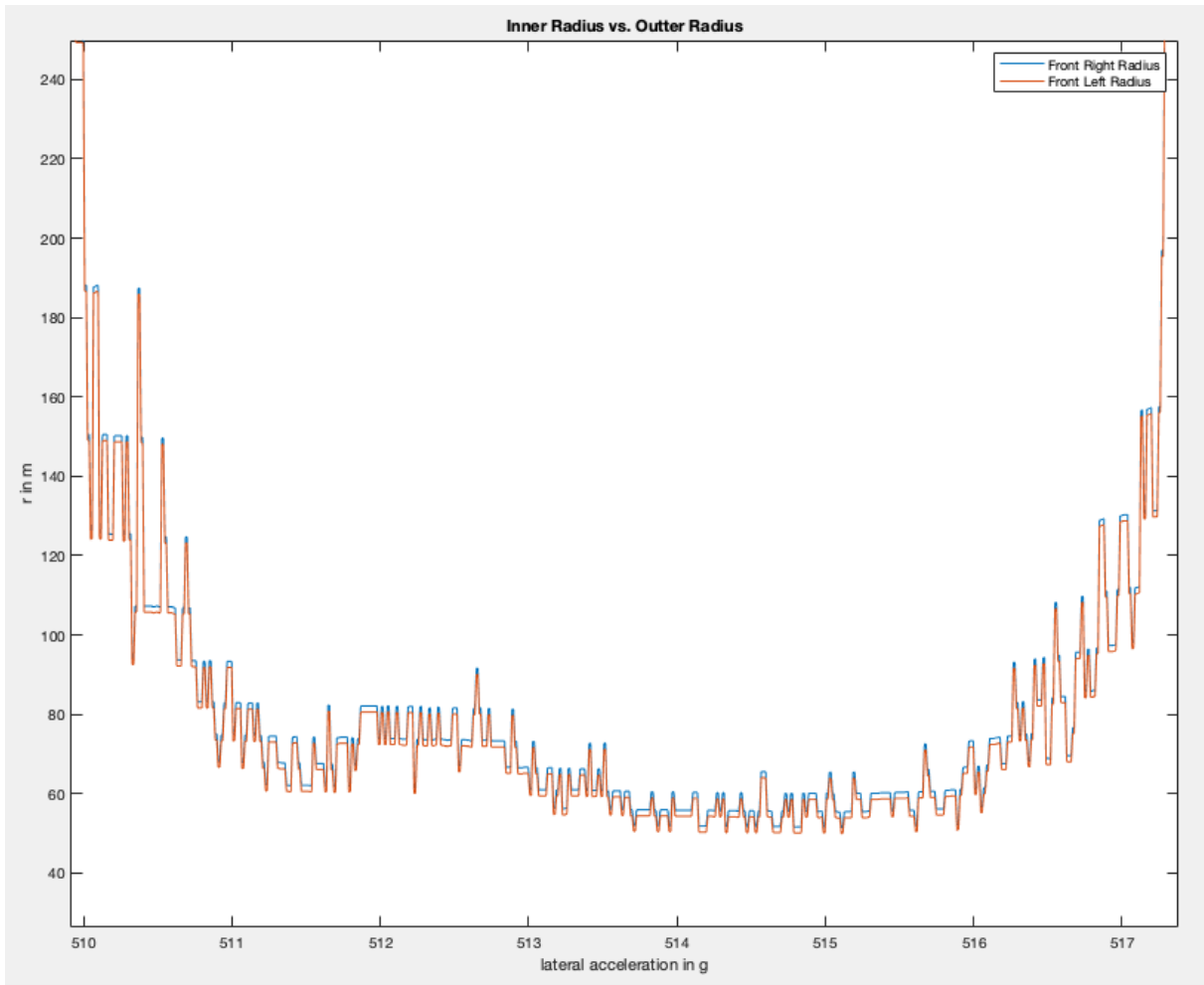


Abbildung 1.3: Vergleich des linken und rechten Radius bei Kurvenfahrt

In der nächsten [Abbildung 1.4](#) wird der Radius in Abhängigkeit des Drehwinkels abgebildet. Es zeigt, dass mit steigendem Drehwinkel der Radius immer kleiner wird. Stellen, an denen das Fahrzeug nicht fährt, können hierbei vernachlässigt werden. Allgemein gilt, dass bei Kurvenfahrten der Radradius abnimmt.

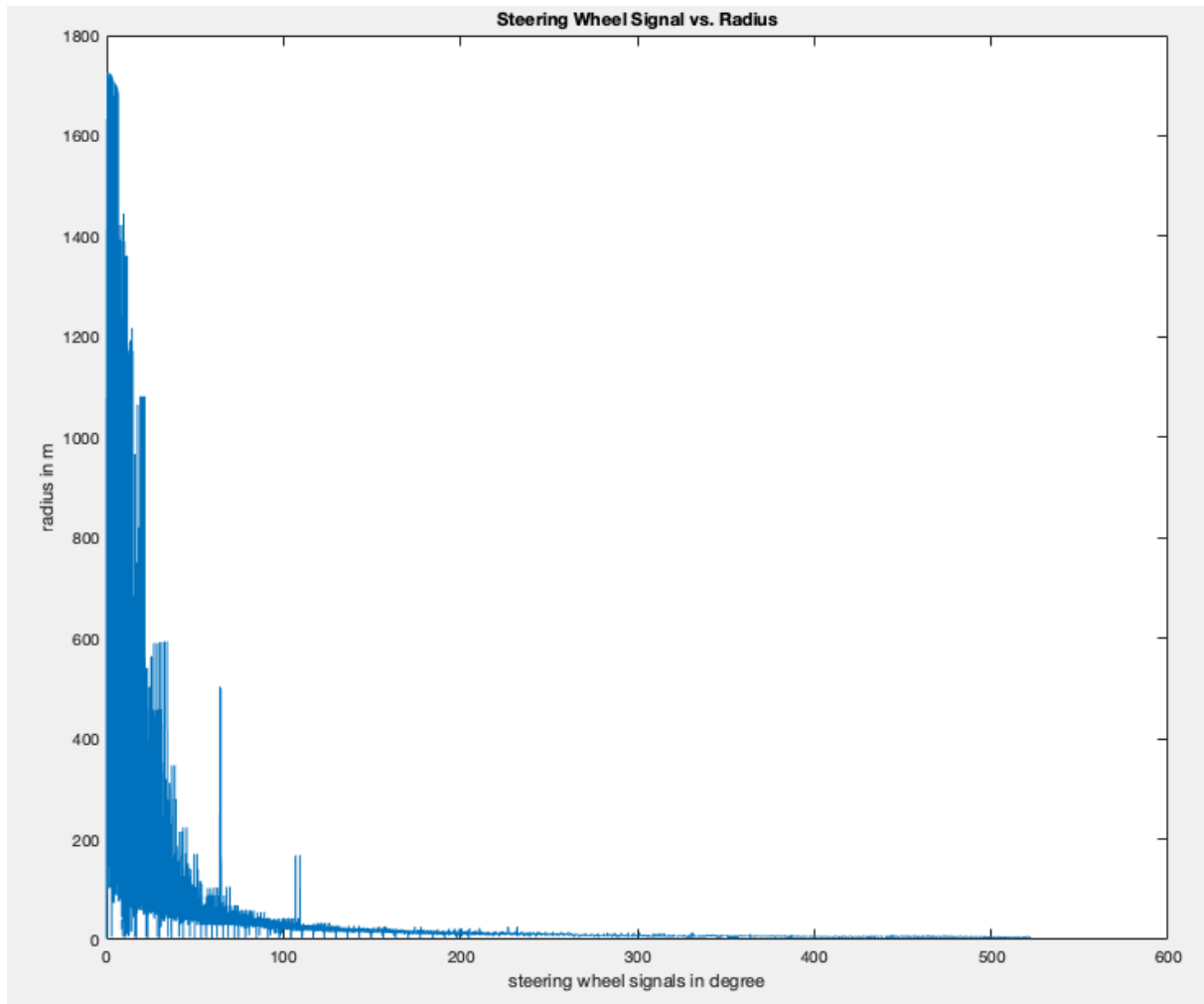


Abbildung 1.4: Lenkraddrehwinkel gegenüber Radius

In der folgenden [Abbildung 1.5](#) wird der Radius in Abhängigkeit der Geschwindigkeit dargestellt. Es zeigt, dass bei gleichem Drehwinkel des Lenkrads bei steigender Geschwindigkeit der Radius immer größer wird.

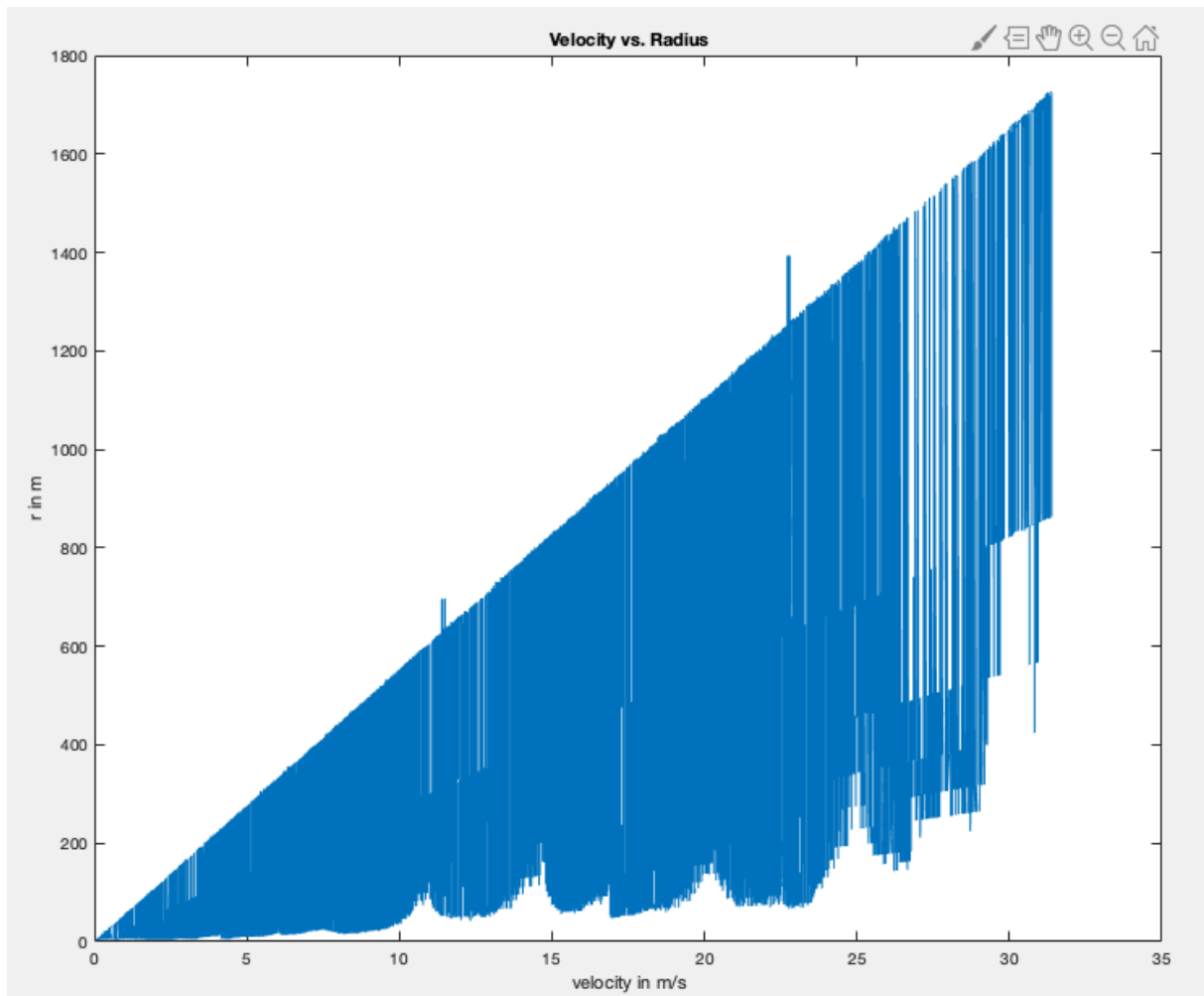


Abbildung 1.5: Geschwindigkeit gegenüber Radius



## 2 Aufgabe D3 und D4

### 2.1 Simulink-Modell

Die nachfolgenden Abbildungen zeigen das Simulink-Modell zur Berechnung der Distanz der einzelnen Räder.

Das erste Modell in [Abbildung 2.1](#) hat die vier Geschwindigkeiten (in m/s) der Räder als Eingänge. Diese werden integriert, um die Strecke zu erhalten. An den Transport Delay Gliedern wurde ein Time Delay von 10 eingestellt. Anschließend wird das verschobene Signal von der wirklichen Strecke abgezogen, damit immer nur ein Zeitraum von 10s betrachtet wird. Die Strecke in dieser Zeitspanne wird von allen vier Rädern verwendet, um den Durchschnitt zu berechnen. Anschließend wird für jedes Rad berechnet, wie weit die Strecke vom Durchschnitt abweicht. Bei einer Abweichung von mehr als 0,05% liegt die Kurve im Scope bei "1", sonst bei "0".

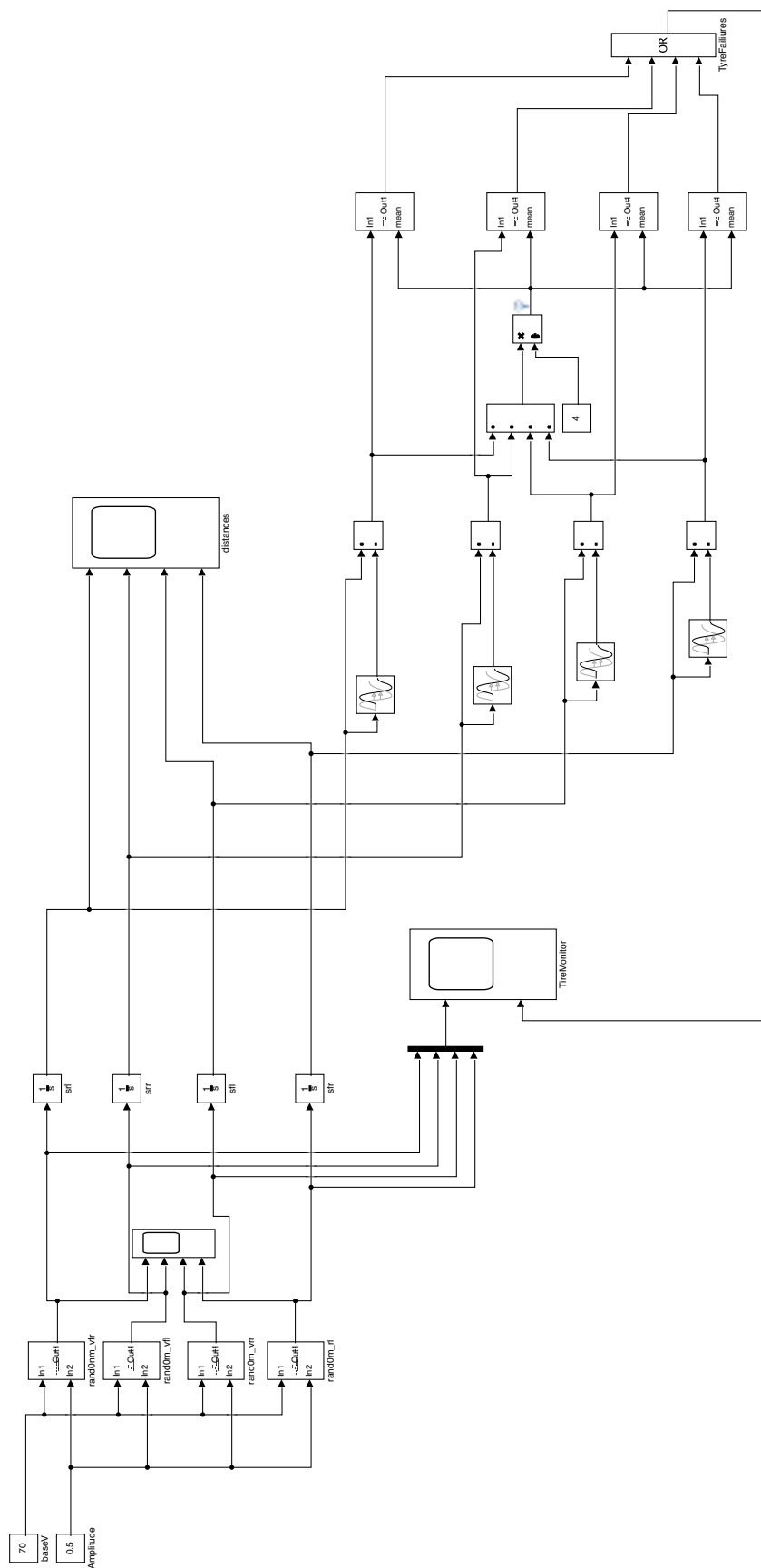


Abbildung 2.1: Simulink-Modell1

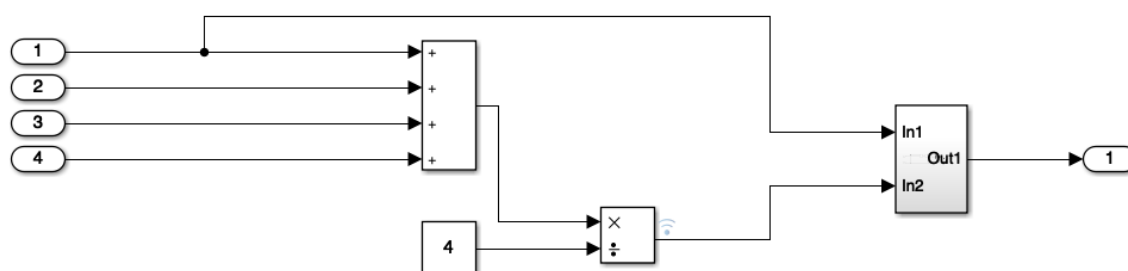


Abbildung 2.2: Subsysteme des Simulink-Modell1

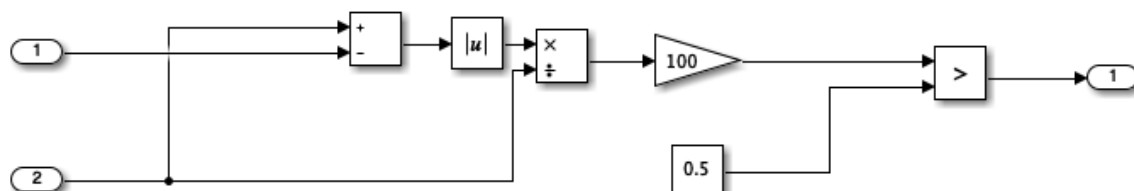


Abbildung 2.3: Subsysteme des Simulink-Modell2

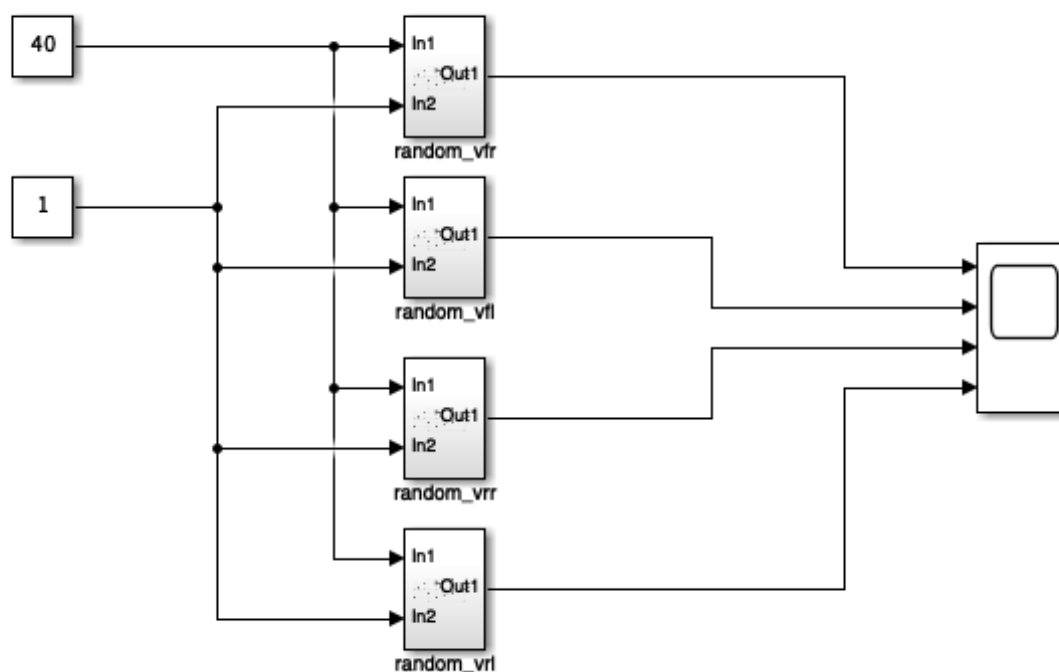


Abbildung 2.4: Subsysteme des Simulink-Modell4

## 3 Aufgabe D4

## 4 Aufgabe D5

Die untenstehende [Abbildung 4.1](#) zeigt den "Random Number Generator".

Dieser wurde durch folgende Formel vorgegeben:  $X(i) = (a \cdot X(i-1) + c) \bmod m$

Für den ersten Eingang wurde die Zahl 40 gewählt, für den Zweiten 1. Für  $c$  wurde der Wert 253 gewählt, für  $a$  der Wert 89 und für  $m$   $2^{32}$ . Die Werte von  $a$  und  $c$  müssen Primzahlen sein. Von dem durch die Formel berechneten Wert wird jeweils  $\frac{2^{32}}{2}$  abgezogen damit eine positive und negative Schwankung erzeugt wird. Der Gain-Block verdoppelt den Wert am zweiten Ausgang, damit es eine Schwankung von 1 und nicht von  $\frac{1}{2}$  erzeugt wird. Die initial Values der Gleichung sind \*\*\*\*\*

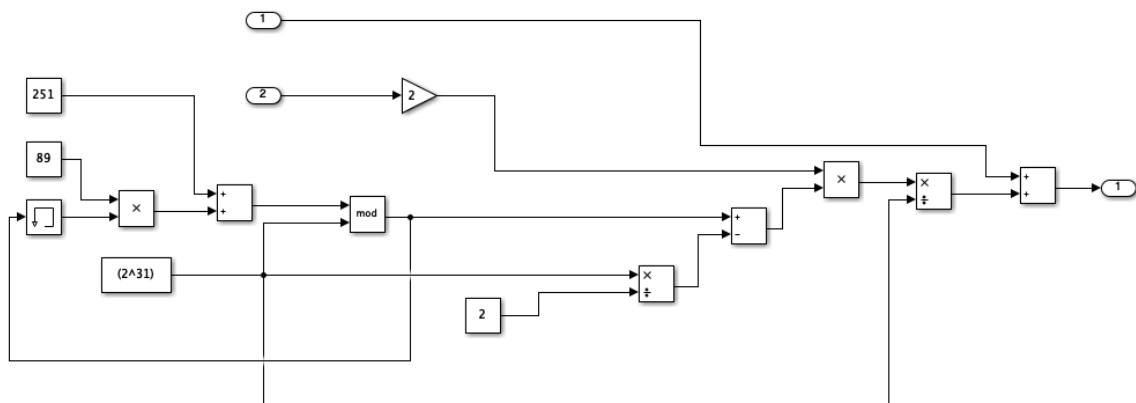


Abbildung 4.1: Random Number Generator

## 5 Aufgabe D6

## 6 Aufgabe D7

Die folgenden Graphiken zeigen die Umsetzung des Simulink Modells in ASCET. Die [Abbildung 6.1](#) berechnet die Strecken der einzelnen Räder. Dafür wird ein RingBuffer mit Werten befüllt. Dieser wurde mit folgendem Code umgesetzt:

```
1 package components;
2 class RingBuffer{
3
4     s_array buffer[1000];
5     real c;
6     real swap;
7
8     public void put(real element){
9         swap = element;
10        for(i in 0 .. 999){
11            c = buffer[i];
12            buffer[i] = swap;
13            swap = c;
14        }
15    }
16
17    public real getLast(){
18        return buffer[999];
19    }
20
21    public real getIndex(integer i){
22        return buffer[i];
23    }
24 }
```

Die [Abbildung 6.2](#) zeigt die Berechnung des Durchschnitts der vier Strecken.

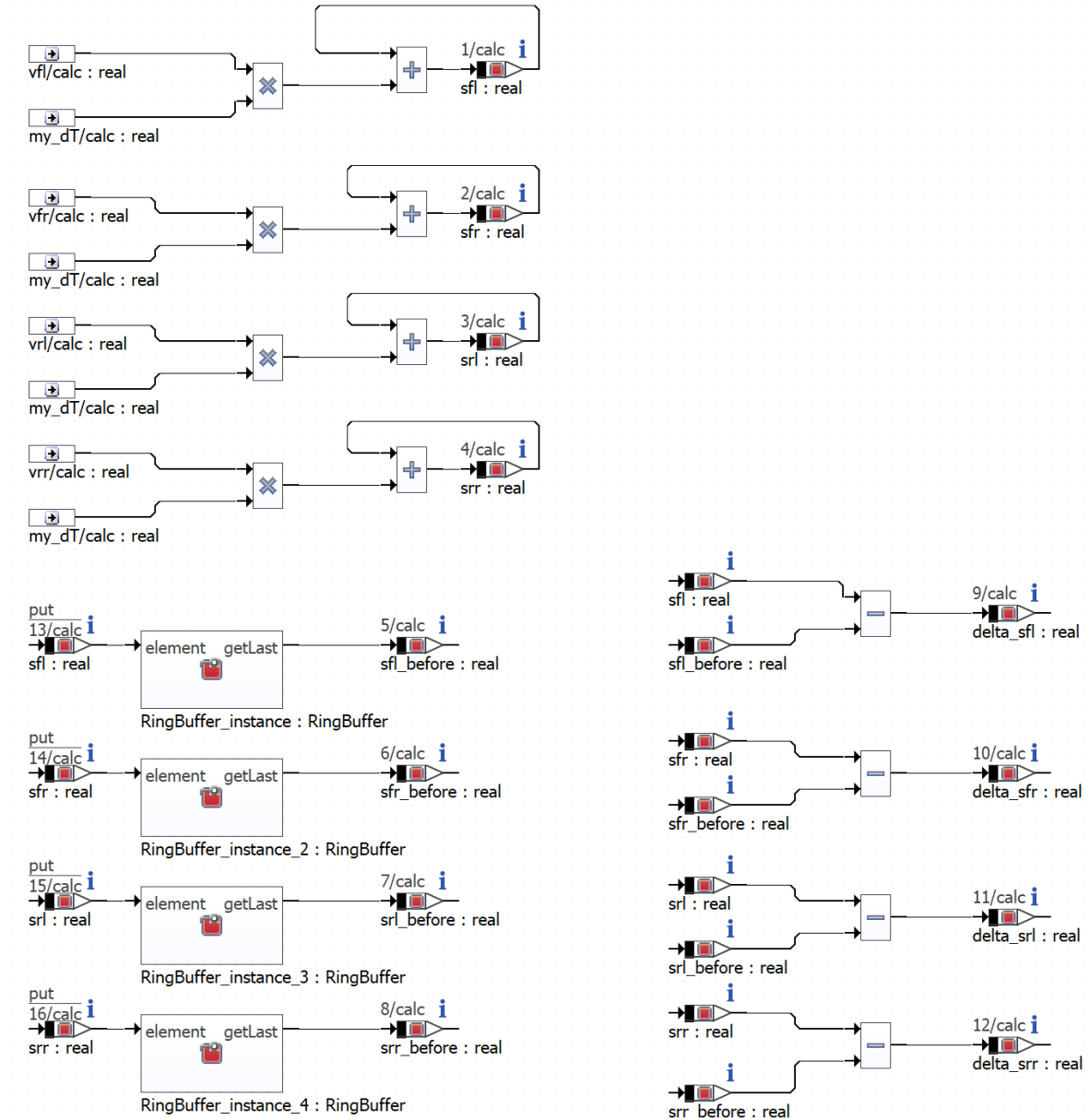


Abbildung 6.1: Model

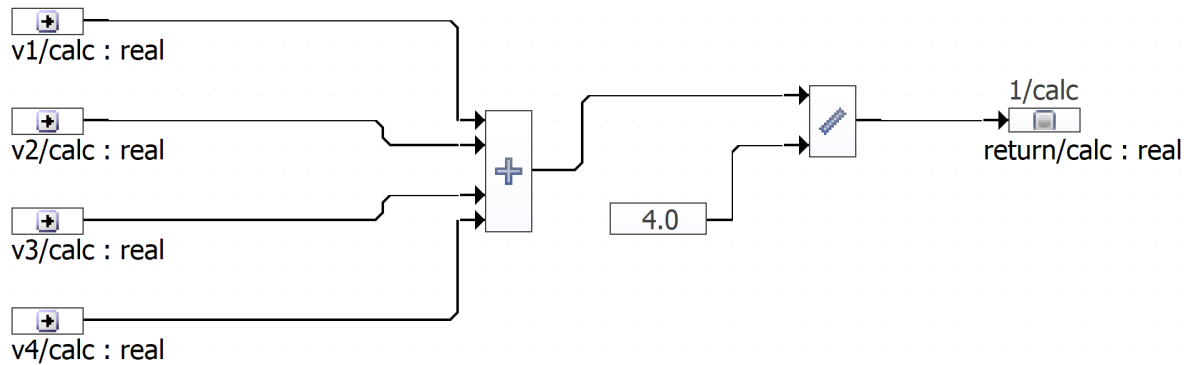
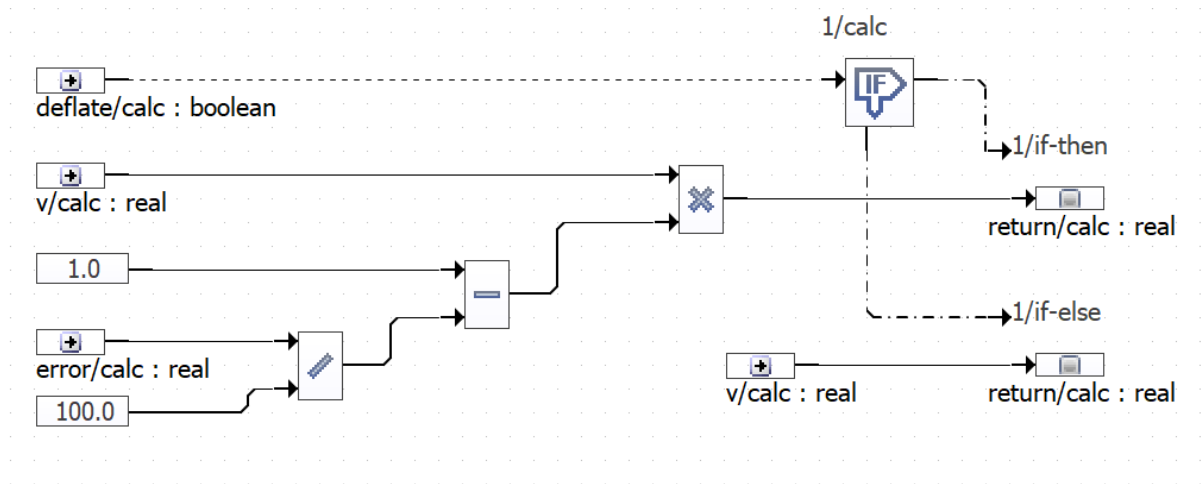
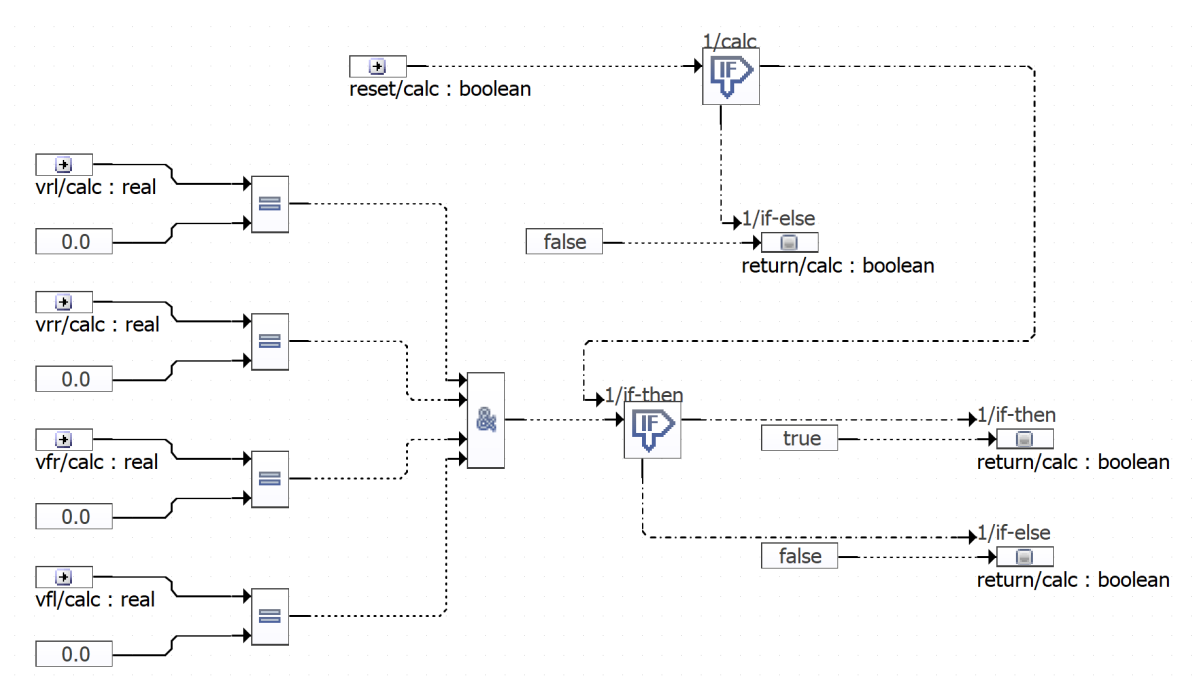


Abbildung 6.2: Tire Mean





Abbildungung 6.3: Error Module



Abbildungung 6.4: Reset

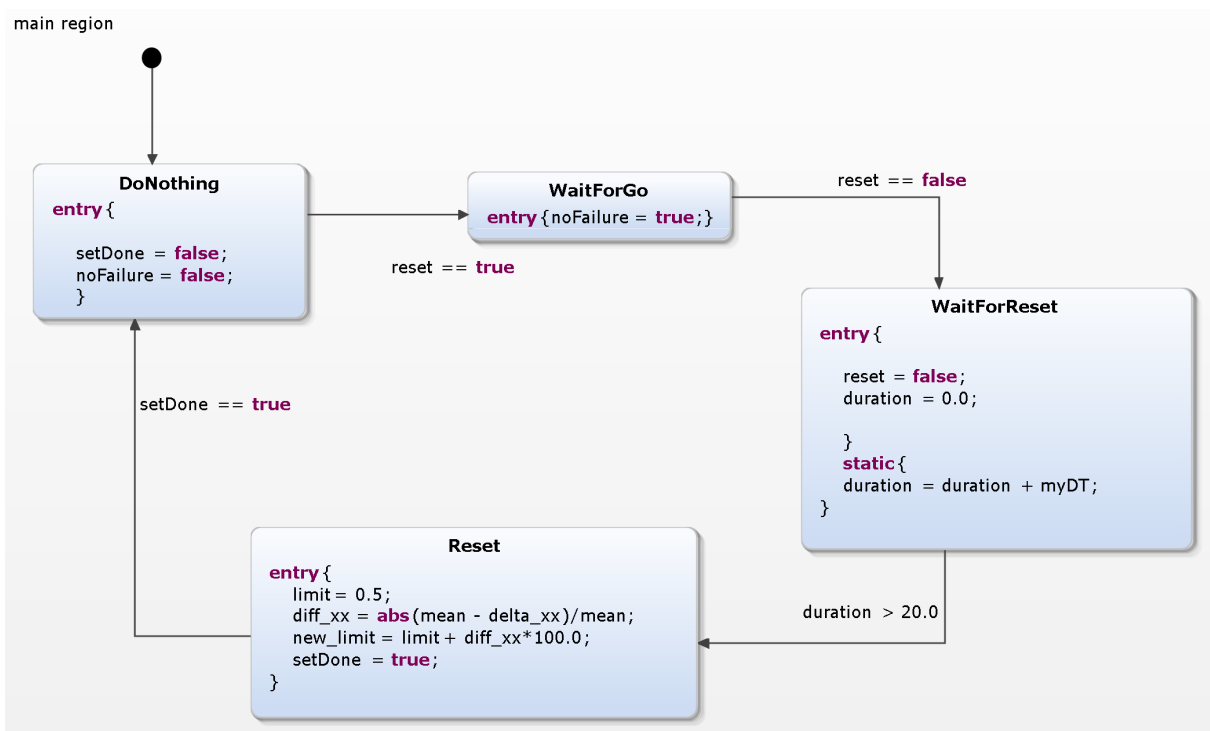


Abbildung 6.5: Reset Statemachine

## 7 Aufgabe D8

Hier werden die Codes der Unit-Tests für alle vorhandenen Funktionalitäten dargestellt.

## 8 Aufgabe D9

Für die Umsetzung der Warnfunktion durch eine LED, die erst einen Druckabfall-/anstieg signalisiert, wurde eine Statemachine verwendet. Diese besteht aus insgesamt fünf Stati, vier die jeweils 0,8 bzw. 1,6s dauern und die LED aus- bzw. einschalten und dem Off-Status.

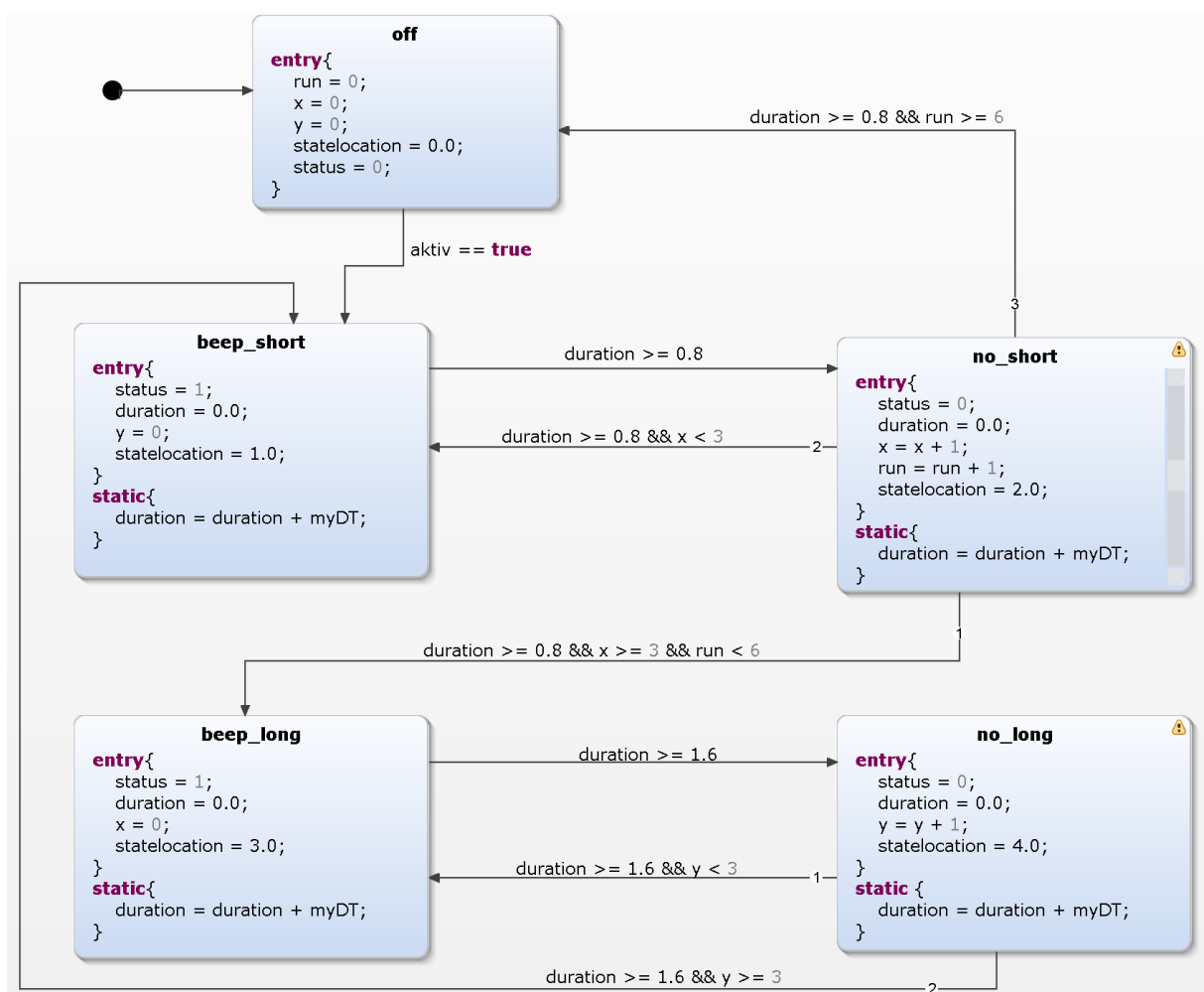


Abbildung 8.1: SOS Statemachine

## 9 Aufgabe D10

Die Abbildung 9.1 zeigt den „Random Number Generator“ umgesetzt in ASCET.

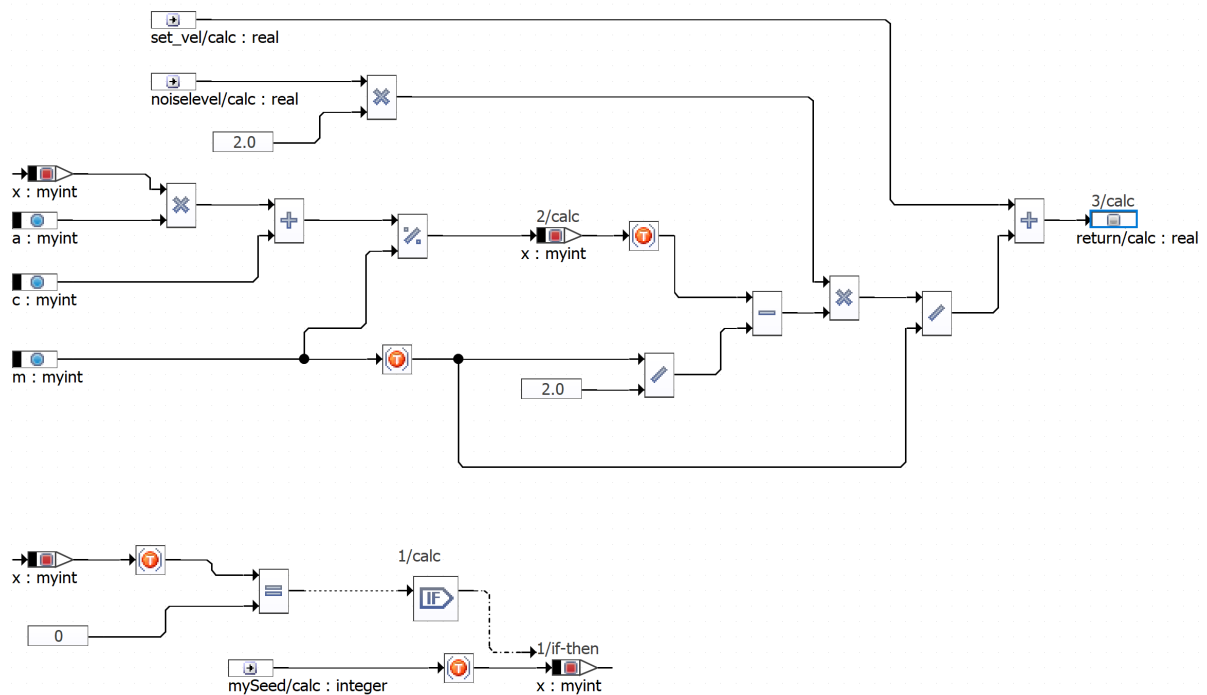


Abbildung 9.1: Random Number Generator

Der Code des Unittests für den „Random Number Generator“ wird im Folgenden dargestellt.

```

1 package components;
2 import assertLib.Assert;
3 static class RandomGeneratorTest {
4
5     RandomGenerator rg;
6     RingBuffer rb;
7
8     @Test
9     public void testcalc() {
10         real set_vel = 100.0;
11         real noiselevel = 5.0;
12         integer mySeed = 10;
    
```

```
13     real erg = rg.calc(set_vel, noiselevel, mySeed);
14     Assert.assertNear(erg, 96.1425781, 0.01);
15 }
16
17 @Test
18 public void testcalc2() {
19     real set_vel = 100.0;
20     real noiselevel = 5.0;
21     integer mySeed = 10;
22     for(i in 0 .. 99){
23         real erg = rg.calc(set_vel, noiselevel, mySeed);
24         rb.put(erg);
25         for(j in 1 .. 999){
26             real act = rb.getIndex(0);
27             Assert.assertNotEqual(act, rb.getIndex(j));
28         }
29     }
30 }
31 }
```

## **10 Aufgabe D11**

## 11 Aufgabe D12