Applied Vehicle Dynamics Control

SD2231– Applied Vehicle Dynamics Control

May 3, 2022

Gustav Jonsson, 19980412-6952
Thomas Perrein, 20010516-T137

Postal address	Visiting address	Telephone	Internet
Royal Institute of Technology	Teknikringen 8	+46 8 790 6000	www.ave.kth.se
KTH Vehicle Dynamics	Stockholm	**Telefax**	
SE-100 44 Stockholm		+46 8 790 9304	
Sweden			

Contents

1 Task 1: Washout filtering approach of side-slip estimation

1.1 Task 1.a:

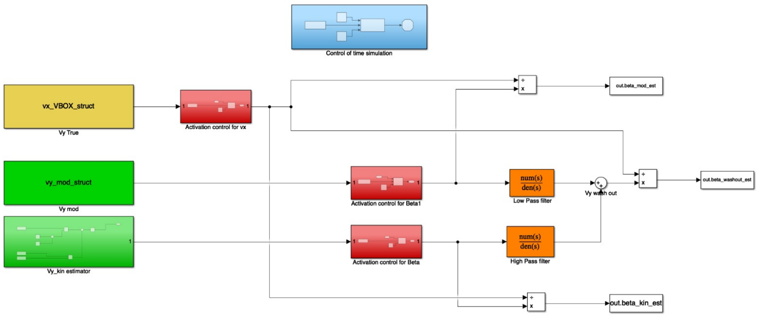


Figure 1: Overview of estimator architecture

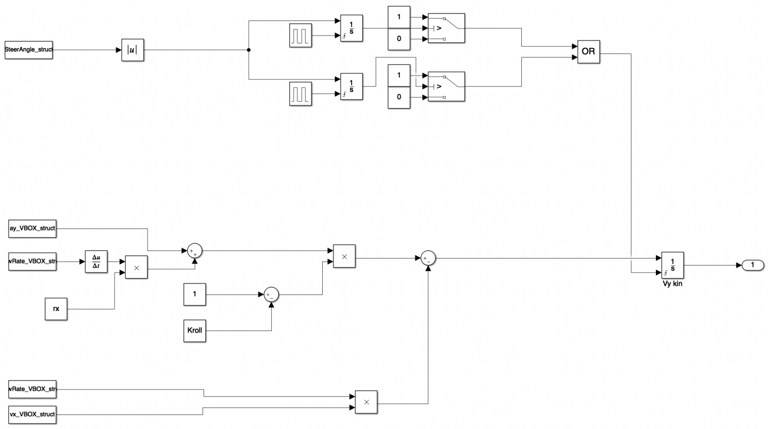


Figure 2: Illustration of v_y kinematic estimator architecture

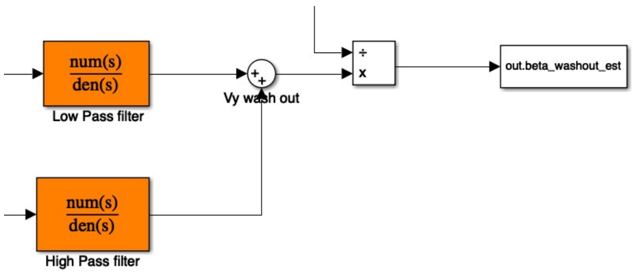


Figure 3: Illustration of v_y washout filter estimator architecture

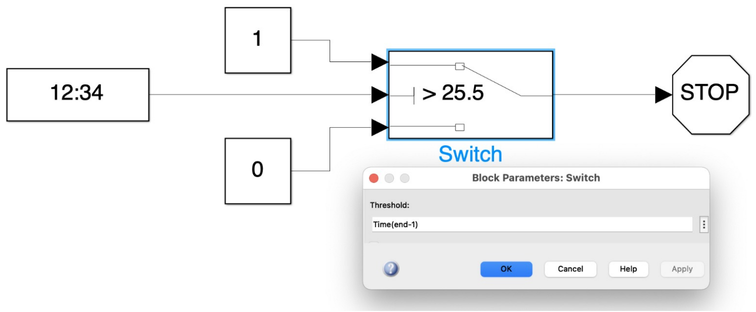


Figure 4: Architecture of the clock to end the simulation properly

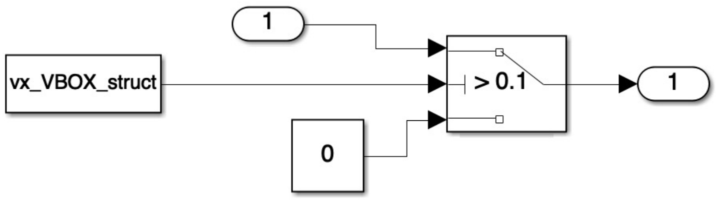


Figure 5: Activation logic to avoid infinite values or peaks in the simulation of β

Figure 1 shows the overview of the estimator architecture in Simulink. Furthermore, Figure 2 and 3 shows the illustration of the estimator for v_y kinematic and v_y washout filter respectively. For the kinematic model the lateral acceleration is transformed from the given lateral acceleration in the IMU system to the COG system using the following formula

$$a_y^{COG} \approx a_y^{IMU} + r_x \ddot{\psi}_z - r_y \ddot{\psi}_z. \tag{1}$$

Furthermore, to avoid drift when integrating, the integration time must be held short. Therefore, the above feature in Figure 2 was added. The idea is to integrate the absolute value of the steering angle and see whether it is above a certain value or not. If it is not, which means that the vehicle is more or less driving straight forward, a zero will be sent to the integrator and the output value will be the true value, without any corrections. However, if the integrated value is above this certain value, which means that the vehicle is steering, then a one will be sent to the integrator in the kinematic estimator and reset the state. Moreover, to the integrator a pulse generator is added to keep the integration time short. There are two integrator blocks above, this is due to when only having one, there will be states when the integrator gets an zero from the pulse generator. To avoid this, the second integrator block was added, but with a pulse generator with a phase delay of half the period of the first integrators' pulse generator period. Both the time period and the integrated values was tuned to get good enough results for all test. The time period was found to be 4 seconds and the integrated threshold value was found to be 0.022.

The model based v_y is calculated directly with the formula in Matlab, see equation 2 below.

$$v_{y,mod} = \frac{v_x(l_r(l_f+l_r)C_fC_r - l_fC_fmv_x^2)}{K_s((l_f+l_r)^2C_fC_r + mv_x^2(l_rC_r - l_fC_f))} \cdot SWA. \tag{2}$$

To the Simulink model, a clock and an activation logic is added according to Figure 4 and 5. The clock is used to end the simulation when all the values in the time vector is passed. Without the clock the simulation would continue even though the whole time vector have been conducted. The activation logic (the red blocks in Figure 1) is applied to avoid infinite values or peaks in the simulation of β, when the longitudinal velocity v_x gets to small, see equation 3.

$$\beta = \frac{v_{y,mod}}{v_x}. \tag{3}$$

1.2 Task 1.b:

In this task some of the vehicle parameters was tuned to improve the estimator models to better fit the true, measured value of β. To decide which of the parameters that should be tuned we used the formulas. From equation 2 we saw that the cornering stiffness C_f and C_r and the steering gear ratio K_s could be tuned to improve the model-based estimator. For the kinematic estimator it is only the vehicle roll gradient k_{roll} which can be tuned as can be seen in the architecture of the Simulink model in Figure 2. For the washout filtered based estimator, all these parameters will affect the estimation due to it is a combination of the model-based and kinematic based estimator. From equation 2 and Figure 2 it is also possible to see that the mass m of the vehicle, the distance from COG to front l_f, the distance from COG to rear l_r and the IMU position relative COG r_x also will affect the estimator. However, we concluded that the mass of the vehicle will be held constant and therefore the COG position will remain, which means that l_f and l_r also will be constant. This also means that r_x will be the same if we do not change the position of the IMU, which we decided to be fixed at the given position.

When the tunable parameters was decided, they also needed to be tuned. For tuning the parameters, they were changed one at a time to see the effect of each of them and by both increasing and decreasing them the behaviour of each estimator compared to the true value could been observed to see if it was improved or not. This was done for the four tests. Furthermore, when the behaviour was known, to say if it was improved when increasing or decreasing the parameter value from the default ones, trial and error was used to find appropriate values for the tunable parameters that was good enough for all tests. In table 1 below, the new values on the tuned parameters is stated. The parameters C_f and C_r was decided to be equal to each other due to the flexibility in the front axle and rear axle was assumed to be the same.

Table 1: Tuned parameter values

Parameter	Default value	Tuned value
C_f	100000	175000
C_r	100000	175000
K_s	16.3	25
K_{roll}	4.5	4.1

1.3 Task 1.c:

Furthermore, after the vehicle parameters have been tuned, next step was to improve the washout filter estimator by tuning the filter coefficient T. To find a T which was good enough for all four tests, we started to tune T for each test separately. See table 2 for the best T for each test.

Table 2: Best tuned T for each test

Test	T
Constant radius cornering	0.15
Slalom 30 km/h	0.01
High speed step steer	0.1
Frequency sweep	0.15

After these values of T was generated, it was observed that the first and fourth test had the same tuned filter coefficient T and the third test was close to this value. Therefore, it was concluded that T needed to be close to these values in order to obtain good results for these three tests. For the second test, the filter coefficient T was really low and differed a lot from the other T values. However, after some reasoning and discussion, it was concluded to keep $T = 0.15$ to obtain good enough results for all of the tests. In the Figures 6, 7, 8 and 9 the final results of the estimators with tuned values both for the vehicle parameters and for the filter coefficient T is shown. (The individual results of each test could be improved alot, but the task was to find good enough values for all test, which are the results presented below.)

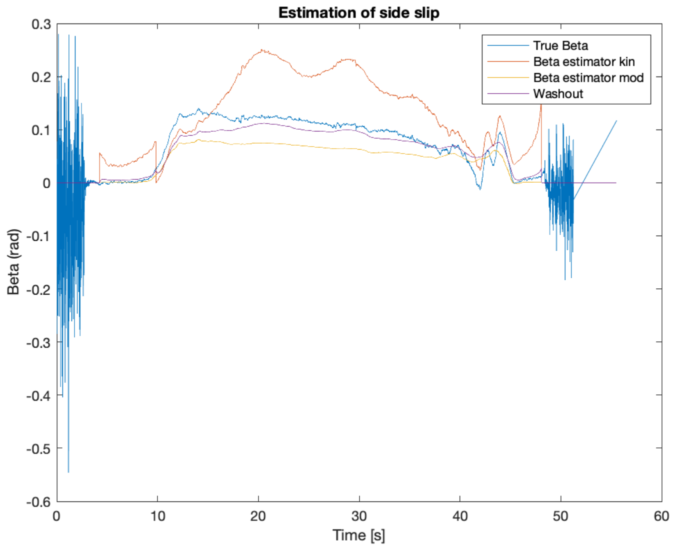


Figure 6: Estimators for constant radius cornering test

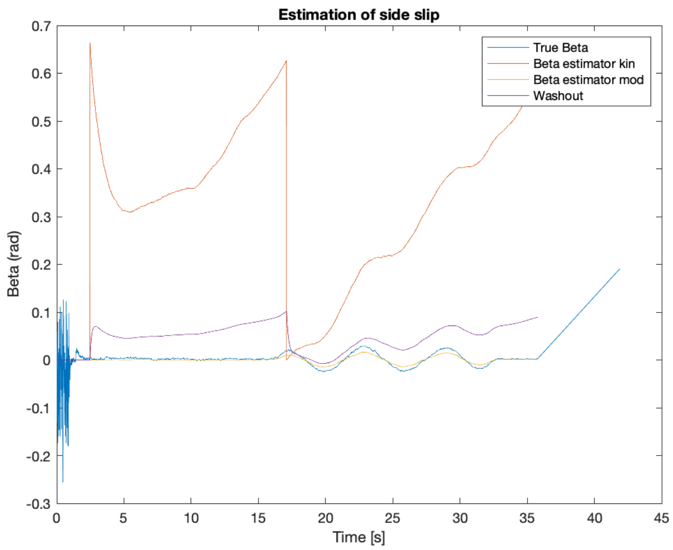


Figure 7: Estimators for slalom at 30 km/h test

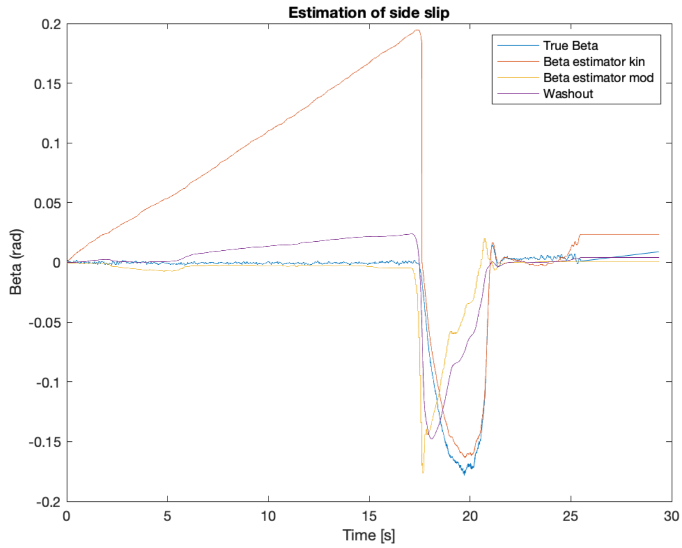


Figure 8: Estimators for high speed step steer test

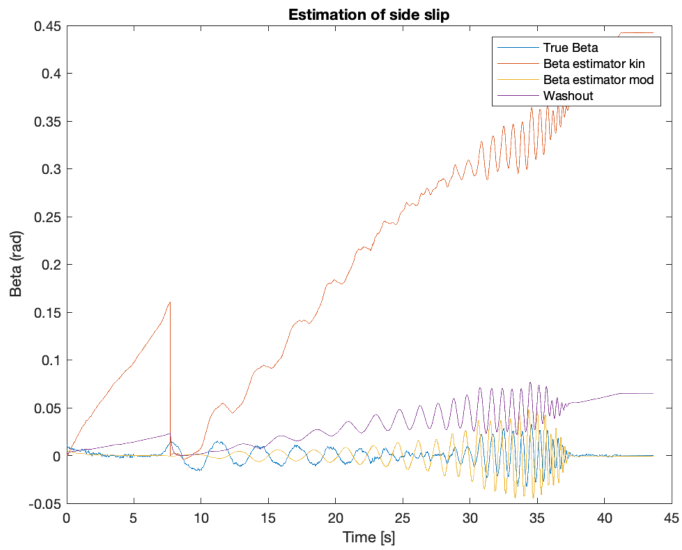


Figure 9: Estimators for frequency sweep test

1.4 Task 1.d:

In the tables below the mean squared error (MSE) and the max error is stated for all estimators and for all driving scenarios. It has been calculated from the figures below by using the Matlab function mse and the given matlab code errorCalc. To get more accurate errors, peaks in the beginning and end of the data has been removed for the first test, constant radius cornering, due to large oscillations in the true value.

Table 3: MSE and max error towards reference for constant radius cornering

Estimator	Mean squared error	Max error
Kinematic	5.105e-03	1.496e-01
Model-based	1.285e-03	5.581e-02
Washout	3.191e-04	5.338e-02

From Figure 6 the errors stated in table 3 have been calculated. For this test, the model-based estimator is slightly, overall better than the kinematic estimator and with the washout filter the mean squared error is decreased even more and is well fitted to the true beta values. From Figure 6 it is easy to see that both the model-based and kinematic model follows the oscillations of the true beta values, however there is quite large difference between the kinematic model and the true values between 15 to 40 seconds.

Table 4: MSE and max error towards reference for slalom at 30 km/h

Estimator	Mean squared error	Max error
Kinematic	1.298e-01	6.608e-01
Model-based	5.087e-05	4.538e-02
Washout	2.739e-03	8.739e-02

For the slalom test, the model-based estimator is much better than the kinematic model, see table 4 and Figure 7. It is also better than the washout filter estimator. This is because the washout filter estimator is dependent on both the kinematic and model-based estimator and due to the increasing values of the kinematic model, the washout filter will be affected. The max error is quite large for the kinematic estimator, which depends on the high peaks at the beginning. These peaks could be decreased with other tuning parameters, however then the result for the other tests would be worse, therefore it was decided to keep this results.

Table 5: MSE and max error towards reference for high speed step steer

Estimator	Mean squared error	Max error
Kinematic	8.874e-03	2.030e-01
Model-based	1.471e-03	1.538e-01
Washout	9.145e-04	1.096e-01

For the high speed step steer, see table 5 and Figure 8, the mean squared error is best for the washout filter estimator. The kinematic and model-based estimator are more or less equal in the performance. For the max error, the three estimators is more or less the same. The model-based estimator have some problem to follow the true values of β and therefore also the washout filter estimator will have some problem. The kinematic estimator is good, if one ignores the increasing part in the beginning, which could not be removed even though it was decreased with the tuned values compared to the results without tuned values.

Table 6: MSE and max error towards reference for frequency sweep

Estimator	Mean squared error	Max error
Kinematic	5.435e-02	4.425e-01
Model-based	1.934e-04	4.775e-02
Washout	1.255e-03	7.696e-02

Finally, for the frequency sweep, the best overall estimator if looking to the mean squared error, see table 6, is the model-based estimator. Also here, as for the slalom test, the kinematic estimator is increasing, see Figure 9, whereas the washout filter estimator will be affected. However, the washout filter estimator manage to follows the oscillations of the true values quite good anyways.

Overall, the max error is larger in all test than the mean squared error, which is obvious due to the max error is the largest error and the mean squared error is an type of average value from all datapoints.

1.5 Task 1.e:

The kinematic estimator is the one which tends to drift most over time. Especially for the tests, slalom and frequency sweep. As the time increasing, also the β value of the kinematic estimator is increasing, which probably depends on that this estimator is not capable to handle the fast oscillations in the true β values, because then the kinematic estimator tend to drift. The washout filter will have some small drift over time, which depends on the drift in the kinematic model. For the model-based estimator there is no drift over time.

If one wants to compare how the accuracy depends on steady-state and transient manoeuvres, we can compare the results of the error estimation of the three later tests from task 1.d with the error results from the first test, the constant cornering test from 1.d. The constant cornering radius test is a steady-state test and the other tests are so called transient tests. If then comparing the mean squared errors in table 3 with the tables 4, 5 and 6 we see that the kinematic model is more accurate for steady-state tests than for transient manoeuvres, which has the same reasoning as described above. To say, it is sensitive against oscillations and quick responses. However, the model-based estimator is better for transient manoeuvres than for steady-state tests. That is probably because for these tests, the kinematic estimator is increasing its β values over time and therefore we did not needed to tune it more accurate, because the affect of tuning the parameters was very small, instead we kept the model-based estimator accurate. But for the steady-state tests the tuning had more effect on the kinematic estimator due to it is not increasing, whereas we found tuned parameters that was good enough for both the kinematic and model-based estimator. The washout filter estimator depends on both the other two estimators and seems to be most accurate for steady-state tests. That is because the kinematic estimator is more sensitive and therefore the test where the kinematic estimator is best, will also be most accurate tests for the washout filter estimator.

About how accuracy is depending on vehicle speed, we first plot the error ($|estimator - true\_value|$) over time for each test, together with the velocity on axis x over time to see how the behaviours are linked. This leads to these four Figures 10, 11, 12 and 13:

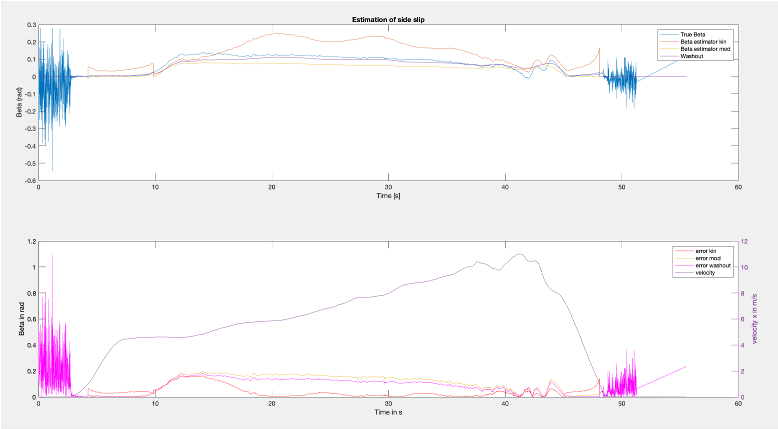


Figure 10: Error and velocity v_x for circular test

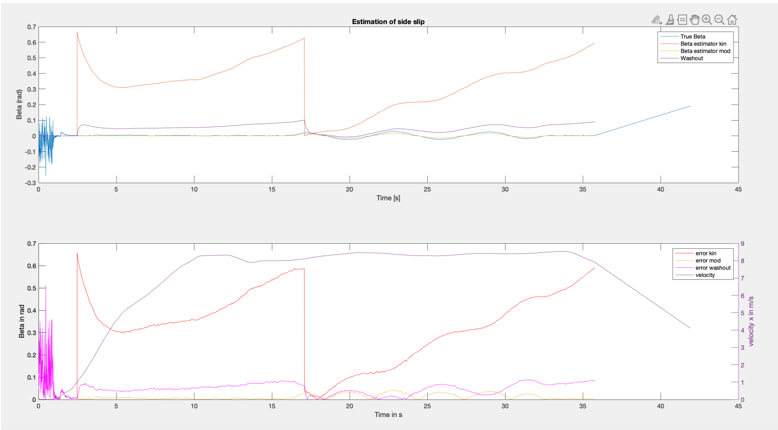


Figure 11: Error and velocity v_x for slalom test

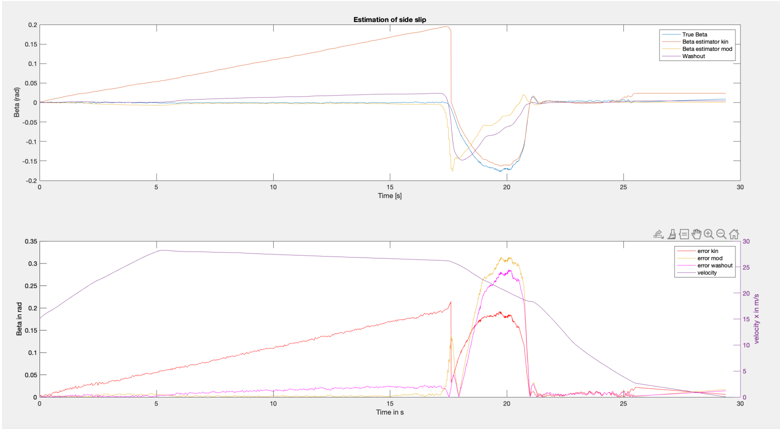


Figure 12: Error and velocity for step test

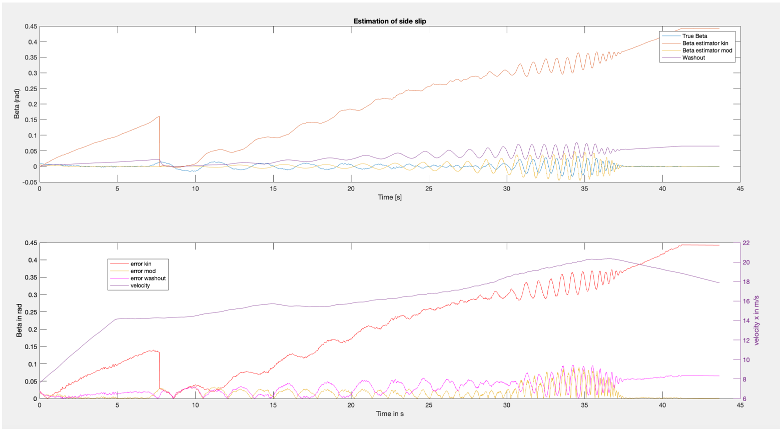


Figure 13: Error and velocity for frequency sweep test

With those figures we can understand that the accuracy of the kinematic estimator decrease when the velocity v_x increases. Nevertheless, it seems that for the circular driving test, this rule does not hold and we can not explain this.

1.6 Task 1.f:

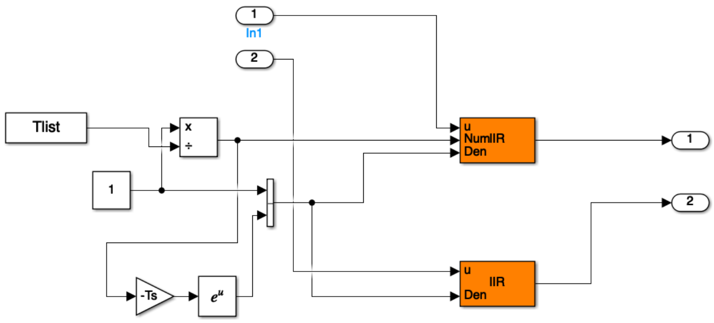
We decided to let T be a linear function over the steering angle which is the vehicle signal that describe the best rate in which the vehicle is cornering. To do that we had to change our continuous filter to discrete with the formula:

$$\frac{1}{Ts+1} \iff \frac{1/T}{1-e^{-T_s/T}z^{-1}} \tag{4}$$

In Matlab we wrote:

```
a=0.05;
b=1.5;
Tlist=[Time,a*SteerAngle + b];
```

And in Simulink we implemented it as:



In the end, with the values a=0.05 and b=0.5 which are the best compromise, we find the following errors:

Table 7: MSE and max error for T as linear function for each test

Test	Mean squared error	Max error
Circular driving to the left	2.330e-03	1.649e-01
Slalom	3.429e-02	6.612e-01
Step steer to the left	2.542e-03	2.348e-01
Frequency sweep	1.397e-02	2.231e-01

As a conclusion, we can not see a real improvement by using such a method. Either we have to change our function T using another parameter, either we have to find a better set of tuned parameters. We did not have time to fix that properly, but there is surely room for improvement.

2 Task 2: Unscented Kalman Filter estimation

2.1 Task 2.a

To implement such a filter, we used the scripts provided such as Vehicle_state_eq, Vehicle_measure_eq and UKF_start. First, we discretized our equations in Vehicle_state_eq with the help of Runge Kutta script, in order to have $x_k = f(x_{k-1}, u_k)$ where $x = [v_x, v_y, \dot{\psi}_y]^T$ and $u = \delta$. Then, for the measurement equation, we simply noticed that:

$$y = \begin{bmatrix} v_x \\ a_y \\ \dot{\psi}_y \end{bmatrix} = \begin{bmatrix} v_x \\ \dot{v_y} + \dot{\psi}_y v_x \\ \dot{\psi}_y \end{bmatrix} \Rightarrow y_k = \begin{bmatrix} v_{x|k} \\ (F_{34|k} + F_{12|k} cos(\delta_k))/m \\ \dot{\psi}_{y|_k} \end{bmatrix} = h(x_k, u_k) \quad (5)$$

Then, in UKF_start, we simply initialize first Q = 0.1*I_3 as indicated and R = 0.01*I_3, which means we first think our variables are uncorrelated.
Then we initialise the covariance matrix P=Q, inspiring ourselves from the Kalman filter example, and we take arbitrarily $x_0 = [0.1\ 0.1\ 0.1]^T$ because it cannot be $[0\ 0\ 0]^T$ since we can not divide by 0. Then at each step, we use ukf1_update and ukf1_predict properly in order to get the value of x and P, and store them for a future plot. You can find the code below:

```
for i=2:n
    predictParam.SteerAngle=SteerAngle(i);
    [x(:,i-1),P] = ukf_predict1(x(:,i-1),P(:,:,i-1),
                    state_func_UKF,Q,predictParam);
    P(:,:,i-1)=P;
    Y = [vx_VBOX(i);ay_VBOX(i);yawRate_VBOX(i)];
    [x(:,i),P,~,~,~,~] = ukf_update1(x(:,i-1),P(:,:,i-1),
                         Y,meas_func_UKF,R,predictParam);
    P(:,:,i)=P;
end
```

2.2 Task 2.b

This is the figure we obtained for standstill test:

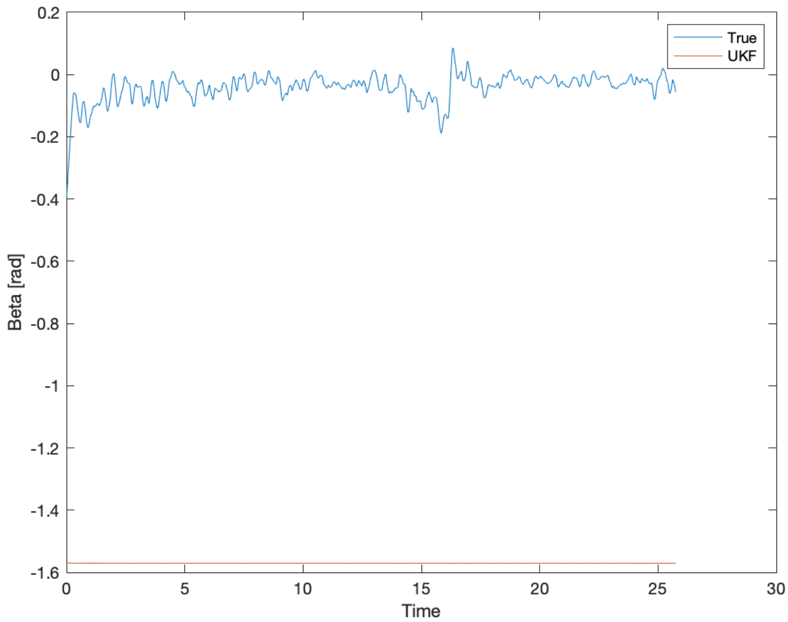


Figure 14: UKF with tuned parameter $C_r = C_f = 175000$ for standstill test

We noticed that when we are in period of standstill, the value of β is close to -1.5 or 1.5. This will be our references to delete the peak values we would have if we let the algorithm go through standstill era. We do not need to evaluate beta when standing still. In fact, for the other tests (especially the circular test) we noticed that even above 0.4 and below -0.4, you would find those peaks. So in the end of our program we added the following lines:

```
my_beta = atan(x(2,:)./x(1,:));
for k=1:n
    if my_beta(k)>0.4 || my_beta(k) < -0.4
        my_beta(k)=0;
    end
end
```

Finally, we obtained the following results:

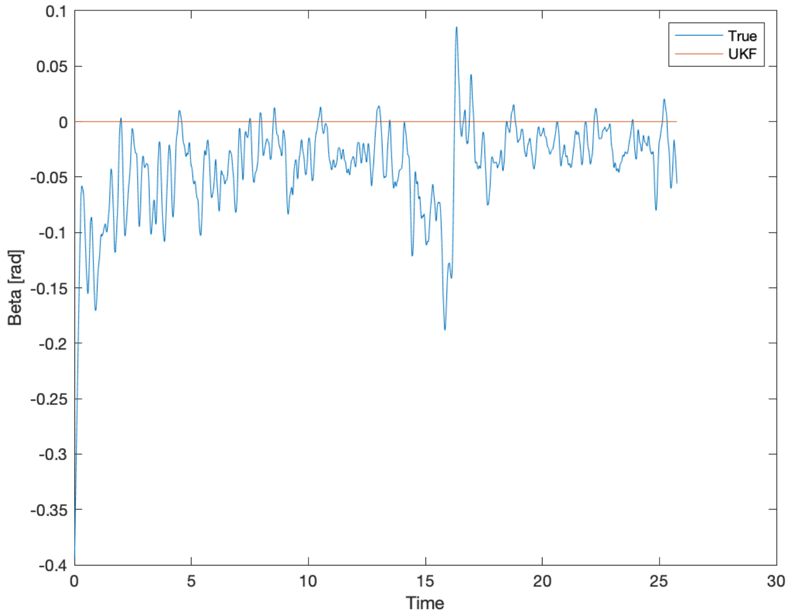


Figure 15: UKF with tuned parameter $C_r = C_f = 175000$ for standstill test w/o peak values

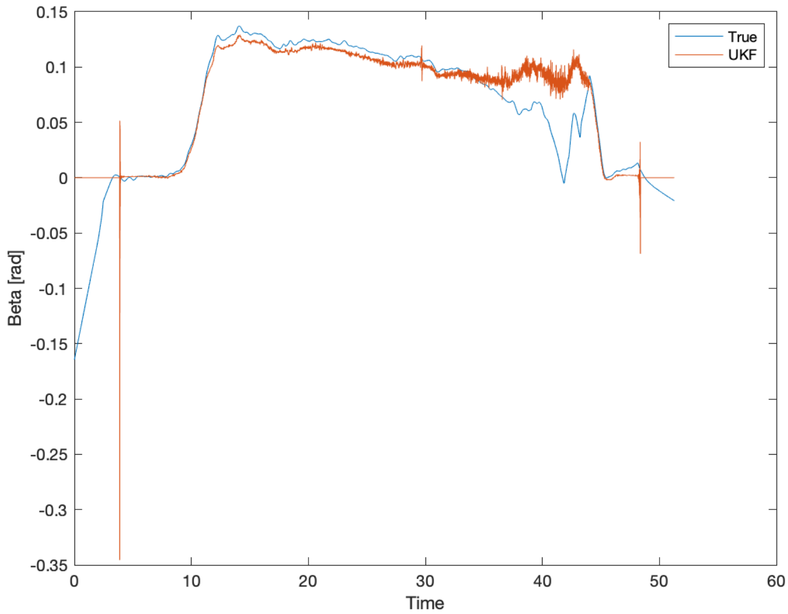


Figure 16: UKF with tuned parameter $C_r = C_f = 175000$ for circular driving test w/o peak values

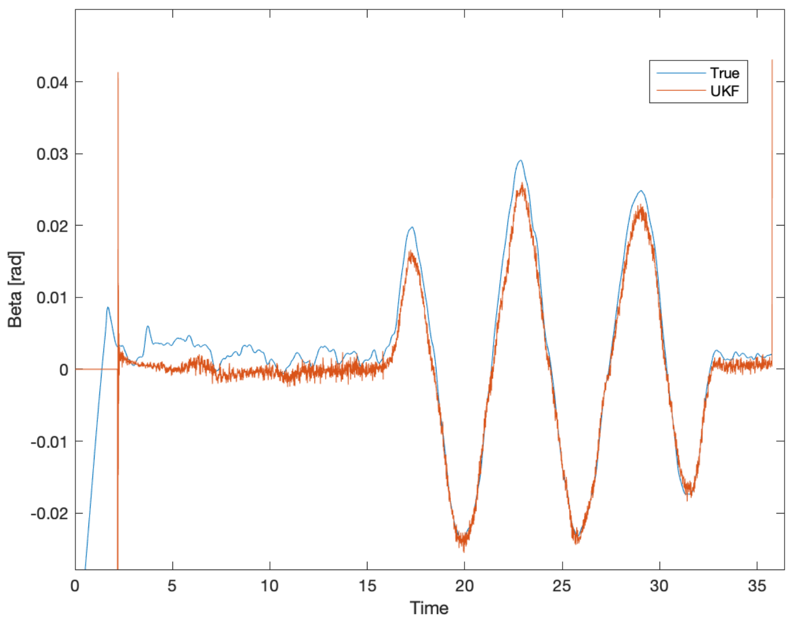


Figure 17: UKF with tuned parameter $C_r = C_f = 175000$ for slalom test w/o peak values

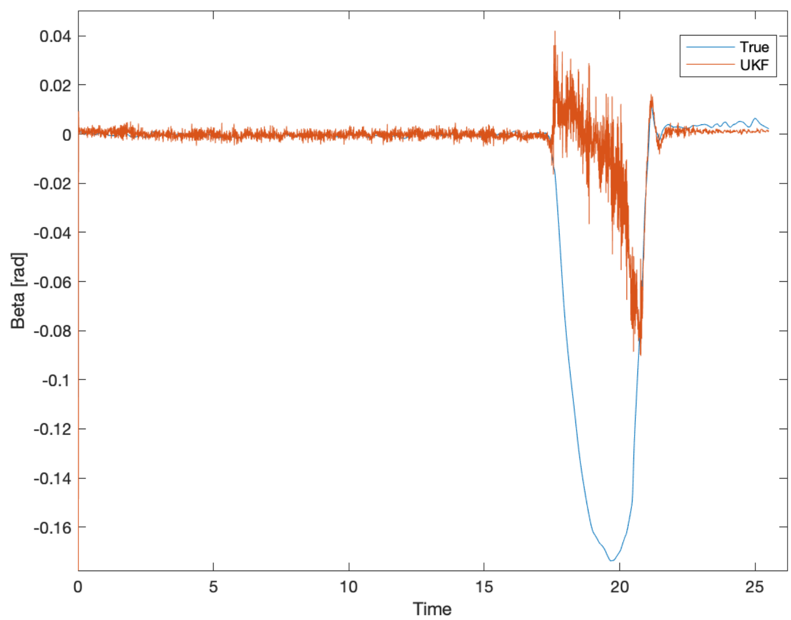


Figure 18: UKF with tuned parameter $C_r = C_f = 175000$ for step steer to the left test w/o peak values

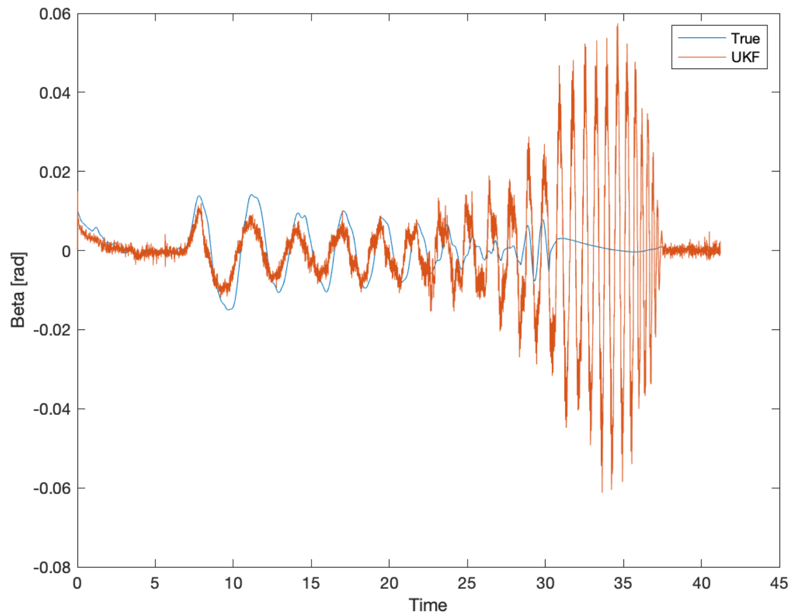


Figure 19: UKF with tuned parameter $C_r = C_f = 175000$ for frequency sweep test w/o peak values

Table 8: MSE and max error for UKF for each test

Test	Mean squared error	Max error
Standstill	3.464e-03	3.933e-01
Circular driving to the left	9.122e-04	3.461e-01
Slalom	7.743e-05	3.974e-01
Step steer to the left	2.138e-03	3.621e-01
Frequency sweep	1.687e-04	6.195e-02