Lab 2 : Lateral Dynamics and states estimation

SD2231– Applied vehicle dynamics control

June 10, 2025

Group 4
Simon Loriot
Carlo Vittorio

Postal address
Royal Institute of Technology
KTH Vehicle Dynamics
SE-100 44 Stockholm
Sweden

Visiting address
Teknikringen 8
Stockholm

Telephone
+46 8 790 6000
Telefax
+46 8 790 9304

Internet
www.ave.kth.se

Contents

Introduction

The purpose of this assignment is to provide knowledge and experience in the area of vehicle state estimation in the context of vehicle motion control. In this lab, the focus will be on the vehicle side slip angle, and multiple methods of estimation will be compared.

1 Tuning process

In this report, multiple parameters are tuned. Two different process was used. The task is to minimize a function $f : x \in E \rightarrow f(x)$. The function is usually implicit, it can be the error between the results of a simulation and the real values of the parameters. We assume the dimension of E to be finite, $dim(E) = n$

A first method is a simple brute force. We can represent E in a multidimensional matrix M and we test every value from this matrix. A plausible range is estimated for each dimension of E, and the minimum value is considered to be the minimum. $min_E(f) \approx min_M(f)$. This approach can be very efficient, since it tests a lot of different values. It doesn't fall into local minimum. The drawback is that it can be computationally long. The number of calculus required is l^n, where l is the matrix size. For high dimensional problems, this method is inefficient.

A second one is to use a gradient based method. We can estimate the gradient of f numerically $\nabla f(x) = \left(\frac{\partial f}{\partial x_i}\right)_i$ with $\frac{\partial f}{\partial x_i} \approx \frac{f(x + dx_i \cdot \delta_i) - f(x)}{dx_i}$. Once the gradient is estimated, we can use the following sequence :

$$\begin{cases} x_0 \in E \\ x_{n+1} = x_n - \mu \nabla f(x_n) \end{cases}$$

The process stops when $||\nabla f(x_n)|| < \eta$. μ, η and x_0 are chosen according to each problem. This process is way faster than the brut force method (especially for multi dimension problem), because the computational cost is proportional to n. However, this process can be stuck into local minimum since $\nabla f(x_n) = 0_n$. The convergence is also very dependent of the starting point x_0 that needs to be sufficiently close to the global minimum.

The process used in this lab is a mix of both method. A first brut force is done to have a first minimum point x_{BF} in a reasonable time. That point will then be the starting point of the gradient based process. In this case, the starting point should be close enough for the gradient based to converge to the global minumum.

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

2.1 Model-based Estimator

One can estimate the body slip angle using a vehicle model. This vehicle model takes into input the steering wheel angle δ_{sw} and the longitudinal speed v_x, and provides the vehicle body slip angle β. The bicycle model was chosen for his accuracy and his simplicity. The following equations describe the vehicle behavior.

$$\begin{cases} m(\dot{v_y^{mod}} + \dot{\psi} v_x) = -C_{34}\alpha_{34} - C_{12}\alpha_{12} \\ J_z \ddot{\psi} = -fC_{12}\alpha_{12} + bC_{34}\alpha_{34} \\ \beta_{mod} \approx \dfrac{v_y^{mod}}{v_x} \end{cases}$$

The parameters α_{ij} are the tires slip angle, and C_{ij} the cornering stiffness.
Starting from some preliminary analysis, it is observed that implementing a constant cornering stiffness value doesn't lead to satisfactory results, despite an optimization process has been undertaken. Especially during harsh scenarios, presenting high steering wheel angles combined with strong accelerations, the saturation of the tire cornering stiffness is too relevant to be neglected. Therefore, a model taking it into account is implemented.
Before describing this model, two crucial preliminary steps must be followed:

- Filtering of the acceleration data, to avoid a compromising propagation of the noise error. An example of the filtering is shown:

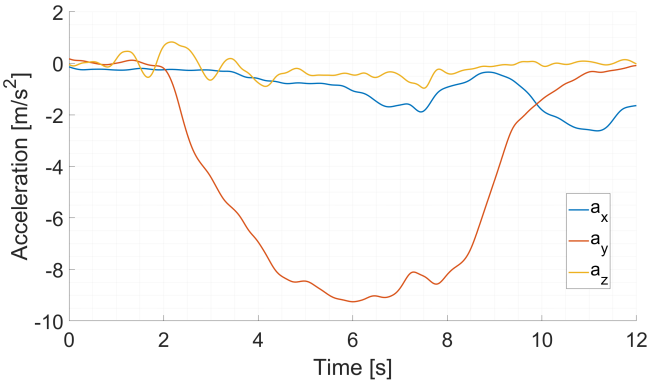


Figure 1: Filtered acceleration example

- Computation of the vertical force acting on each wheel.

The filtering is undertaken by means of a Butterworth filter of the second order.
Instead, the vertical forces are evaluated according to the following procedure:

- The accelerations on the three directions (a_x, a_y, a_z) result in the correspondent inertial forces:

$$F_x = -M \cdot a_x; \quad F_y = -M \cdot a_y; \quad F_z = -M \cdot a_z \tag{1}$$

  where M is the vehicle mass;

- Starting from a free body diagram, the results from the equilibrium are categorized into forces on the two longitudinal axles (front and rear) and forces on the two sides (left and right):

$$\Delta F^F_{z,x} = F_x \cdot \frac{h_g}{l}; \; \Delta F^R_{z,x} = -\Delta F^F_{z,x}; \; \Delta F^L_{z,y} = F_y \cdot \frac{h_g}{t}; \; \Delta F^R_{z,y} = -\Delta F^L_{z,y} \tag{2}$$

- The dynamic variation of vertical forces follows:

$$\{\Delta F_z\} = \frac{1}{2} \cdot [\Delta F^F_{z,x} + \Delta F^l_{z,y}; \; \Delta F^F_{z,x} + \Delta F^r_{z,y}; \Delta F^R_{z,x} + \Delta F^L_{z,y}; \Delta F^R_{z,x} + \Delta F^R_{z,y}] - \frac{F_z}{4} \tag{3}$$

  where F_z is the inertial force introduced in Equation 1;

- The static forces are:

$$\{F_{z,static}\} = \frac{M \cdot g}{2} \cdot [w_d; \; w_d; \; (1 - w_d); \; (1 - w_d)] \tag{4}$$

  where w_d is the weight distribution between front and rear, defined as:

$$w_d = \frac{M_F}{M};$$

- Combining Equations 2, 3 and 4, the result is:

$$\{F_z\} = \{F_{z,static}\} + \{\Delta F_z\} \tag{5}$$

Coming back to the definition of a model for the cornering stiffness, the idea is to define a quadratic function with the vertical load as independent variable: $C(F_z) = \alpha_1 \cdot F_z^2 + \alpha_2 \cdot F_z$. The ordinate at the origin is considered to be 0, since the tire is expected to not generate lateral force when it is not pushed on the tire mark. This consideration significantly simplifies the optimization process, since it reduces the number of parameters to tune.
Two different equations of cornering stiffness are optimized: one for the front tires and the other one for the rear tires, meaning that 4 coefficients ($\alpha_1^F, \; \alpha_2^F, \; \alpha_1^R, \; \alpha_2^R$) must be tuned overall.
To define a set of starting parameters for the optimization, two known constant values of approximate cornering stiffness are exploited: $C_F = 2 \cdot 10^5 N/rad$, $C_R = 3 \cdot 10^5 N/rad$.
The optimization process is followed for all the 5 different test runs available, in order to retrieve an optimized function representative of how many dynamic scenarios as possible. Thus, for each test run a set of coefficients is optimized and, at the end of the collection, a mean of all the coefficients is undertaken to retrieve the final set.
The procedure followed during the optimization, for each test run, consists of the following steps:

- Definition of vectors containing all the α coefficients to test:

```
% Test 4:
a1Fvec = -100e-4:2e-4:-40e-4;
a2Fvec = 77.5:0.2:83.5;
a1Rvec = -100e-4:2e-4:-40e-4;
a2Rvec = 83.5:0.2:89.5;
```

- Extraction of the vertical forces to evaluate the cornering stiffness values:

```
FzFL = Fz(:,1); FzFR = Fz(:,2);
FzRL = Fz(:,3); FzRR = Fz(:,4);
```

- Implementation of nested for cycles to create the different combinations of coefficients:

```
BetaErrMtx=[];
for a=1:length(a2Rvec)
    b_mtx=[];
    for b=1:length(a1Rvec)
        c_mtx = [];
        for c=1:length(a2Fvec)
            d_vec=[];
            for d=1:length(a1Fvec)
                Cfl = a1Fvec(d)*FzFL.^2+...
                a2Fvec(c)*FzFL;
                Cfr = a1Fvec(d)*FzFR.^2+...
                a2Fvec(c)*FzFR;
                Crl = a1Rvec(b)*FzRL.^2+...
                a2Rvec(a)*FzRL;
                Crr = a1Rvec(b)*FzRR.^2+...
                a2Rvec(a)*FzRR;
                Cf = (Cfl+Cfr)/2;
                Cr = (Crl+Crr)/2;
                vy_mod = vx_VBOX.*...
                (parameters.lr*(parameters.lf+parameters.lr).*...
                Cf.*Cr-parameters.lf.*...
                Cf*parameters.mass.*vx_VBOX.^2)./...
                ((parameters.lf+parameters.lr)^2.*...
                Cf.*Cr+parameters.mass.*vx_VBOX.^2.*...
                (parameters.lr.*Cr-parameters.lf.*Cf)).*...
                Steering_at_wheels;
                Beta_mod = atan2(vy_mod, vx_VBOX);
                BetaErr = rms(Beta_VBOX-Beta_mod);
                d_vec = [d_vec, BetaErr];
            end
            c_mtx=[c_mtx;d_vec];
        end
        b_mtx=[b_mtx;c_mtx];
    end
    BetaErrMtx=[BetaErrMtx;b_mtx];
end
```

In the deepest for cycle, the cornering stiffness for the particular coefficients combination is computed, and used to retrieve the error between the rms value

of the vehicle side slip angle signal, by model based estimation and by real data set extraction.
The error value is then collected in a $n^k \times n$ matrix, where n is the number of elements in the coefficient vectors and $k = 3$ is the number of for cycle used minus one;

- Extraction of the minimum value by the error matrix, to identify the optimal combination of coefficients:

```
[row,col] = find(BetaErrMtx==min(BetaErrMtx, [],
    'all'));

vecSize = length(a2Rvec);

a2Rtuned = a2Rvec(ceil(row/vecSize^2))

rowb=row;
while rowb>vecSize^2
    rowb = rowb-vecSize^2;
end
a1Rtuned = a1Rvec(ceil(rowb/vecSize^1))

rowc=row;
while rowc>vecSize^1
    rowc = rowc-vecSize^1;
end
a2Ftuned = a2Fvec(ceil(rowc/vecSize^0))

a1Ftuned = a1Fvec(col)
```

It is important to understand that the row number, corresponding to the optimal set, identifies a single value in the starting coefficient vectors.

At the end of the optimization phase, these are the results collected for the 5 test runs:

Table 1: Results of the cornering stiffness coefficients optimization

Test run	α_1^F	α_2^F	α_1^R	α_2^R	β_{ERR}
1	$-3 \cdot 10^{-3}$	81.5	$-9 \cdot 10^{-3}$	79.3	$25.9 \cdot 10^{-3}$
2	$-6.8 \cdot 10^{-3}$	75.7	$-9 \cdot 10^{-3}$	75.5	$6 \cdot 10^{-3}$
3	$-9.6 \cdot 10^{-3}$	73.5	$-7.4 \cdot 10^{-3}$	75.5	$9.5 \cdot 10^{-3}$
4	$-8.4 \cdot 10^{-3}$	77.5	$-10 \cdot 10^{-3}$	88.9	$18.3 \cdot 10^{-3}$
5	$-9.4 \cdot 10^{-3}$	76.7	$-8.8 \cdot 10^{-3}$	81.7	$8.9 \cdot 10^{-3}$

Starting from these optimized coefficients, their mean values have been computed and used to run check simulations.
The results are satisfactory for all the test runs, but the fourth, where the β estimation is pretty poor:

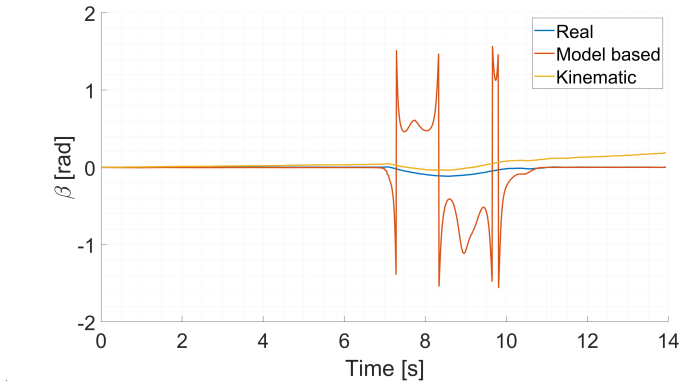
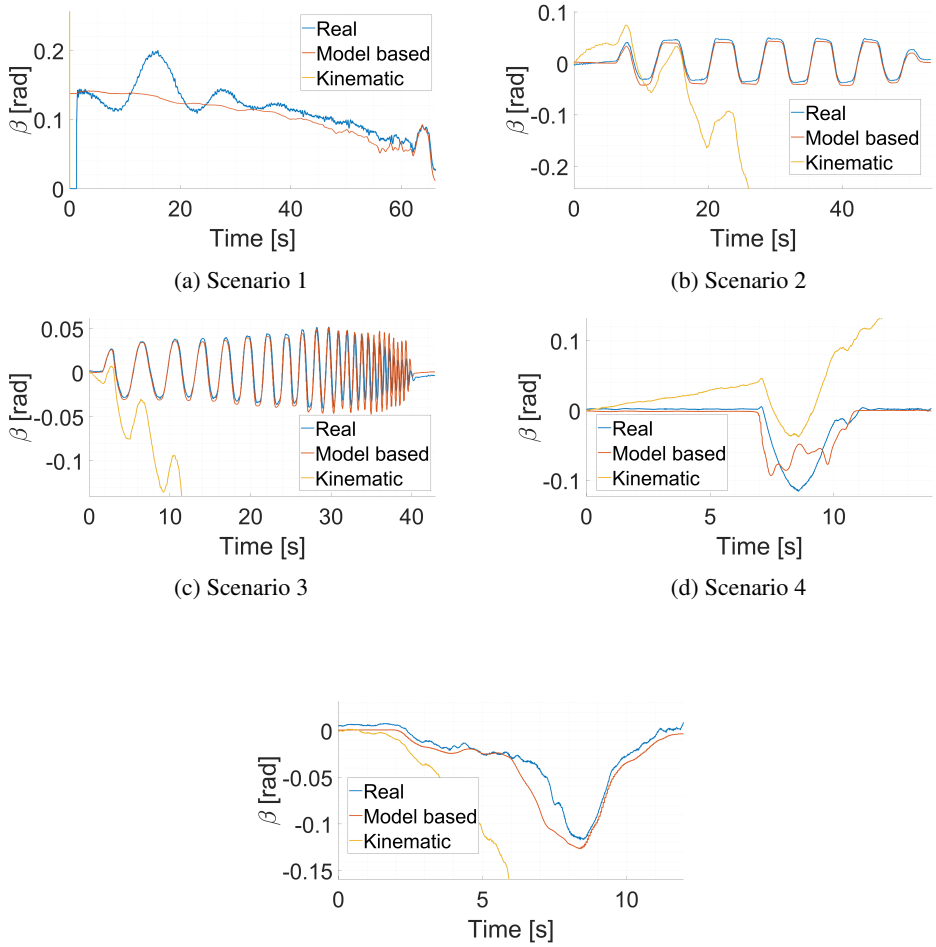


Figure 2: Test 4 results with the mean of the optimized sornering stiffness coefficients

Thus, the optimized coefficient values for the test run 4 have been used for all the other tests, to check if they still lead to good results.
The β_{ERR} values resulting from the other tests are:

$$\beta^1_{ERR} = 27.9 \cdot 10^{-3},\ \beta^2_{ERR} = 6.1 \cdot 10^{-3},\ \beta^3_{ERR} = 10.2 \cdot 10^{-3}, \beta^5_{ERR} = 14.6 \cdot 10^{-3}$$

The estimation results follow as well:



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

(e) Scenario 5

Figure 3: Estimations buy optimal quadratic coefficients

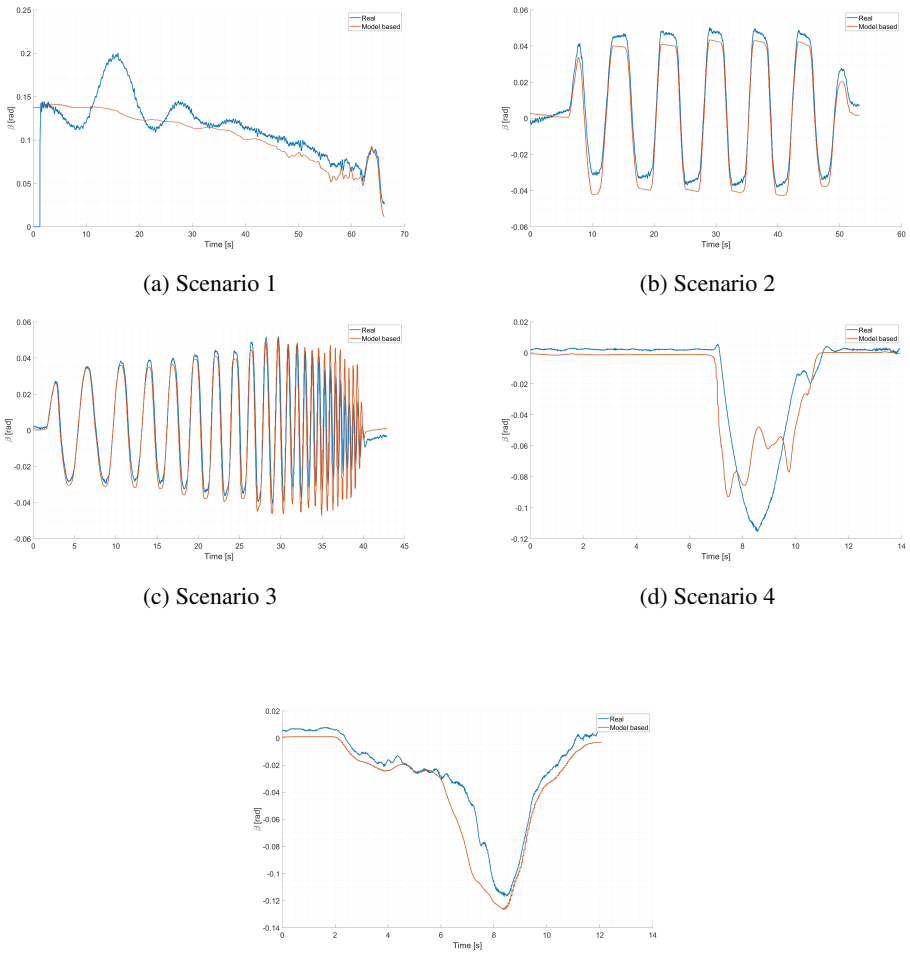
Finally, the tuned quadratic coefficients used from now on in the following Sections are listed here:

Table 2: Final optimized quadratic coefficients for the cornering stiffness

α_1^F	α_2^F	α_1^R	α_2^R
$-8.4 \cdot 10^{-3}$	77.5	$-10 \cdot 10^{-3}$	88.9

Integrating these equations provides a first estimator of the body slip angle that we will call β_{mod}.

Each estimator are tested in 5 different scenarios. These scenario represent a large range of situations with lane change, slalom at high and low slip angle. The results for the model based estimator are represented figure **??**. The model successfully predicts the vehicle body slip angle in most of the different scenario. However, in case of high variation of the parameters, the models fails to follow the real value. This effect is the most obvious in scenario 4 and 3.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

(e) Scenario 5

Figure 4: Performance of the model based estimator

2.2 Kinematic estimator

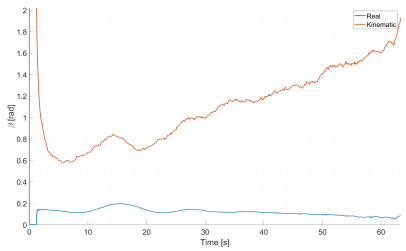
One can estimate the body slip angle by measuring the lateral acceleration, the lateral velocity and yaw rate directly in the car. This estimator rely on sensors that will provide the necessary parameters. The following equation links the different kinematic parameters with the lateral velocity v_y.

$$v_y^{kin} \approx \int_0^T a_y(1 - K_{roll} - \dot{\psi}_z v_x)dt$$

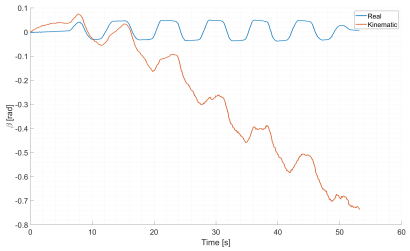
The parameter K_{roll} is the vehicle's roll gradient in [rad/g]. The vehicle slip angle output will be called β_{kin}.

The results of the kinematic estimator are represented figure 5. We notice that the kinematic estimator drifts a lot from the real value. It is due to an offset in the measurement that accumulates error into the estimation. However, the variation of the estimator follows the variation of the real value more accurately than the model

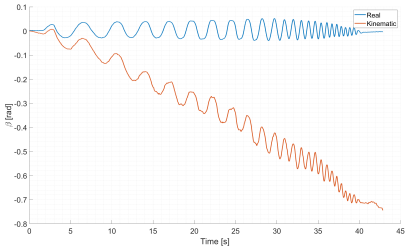
based estimator. This effect is highlighted in the scenario 4, where a the kinematic estimator variations follows the real value.



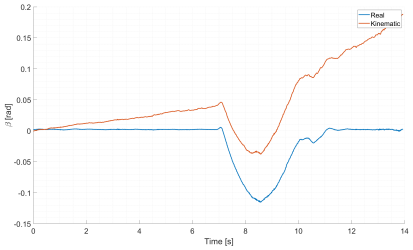
(a) Scenario 1



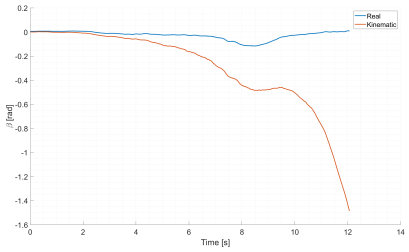
(b) Scenario 2



(c) Scenario 3



(d) Scenario 4



(e) Scenario 5

Figure 5: Performance of the kinematic estimator

2.3 Wash-out filter

An idea to improve the accuracy of the previous estimators is to mix them. A wash-out filter use both model based and kinematic estimator to estimate the lateral speed v_y. The model based estimator loses its accuracy in the transient part of the kinematic. When the slip angle changes is too high, the model cannot follow. However, the kinematic estimator does great in these situations, but drifts over time due to sensor offset errors. A low pass filter will then be applied to the model based speed, and a high pass to kinematic estimator to output a new lateral speed.

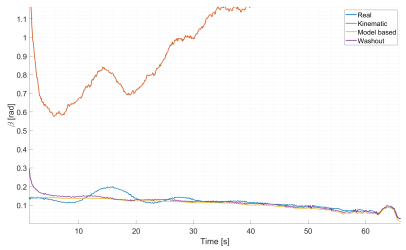
$$v_y = LP(v_y^{mod}) + HP(v_y^{kin})$$

The filter equations are the following.

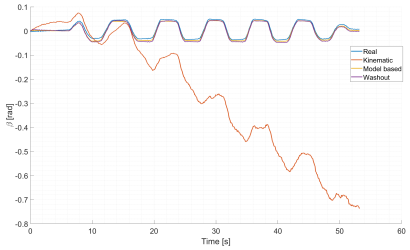
$$\begin{cases} H_{LP}(s) = \dfrac{1}{1+sT} \\ H_{HP}(s) = \dfrac{sT}{1+sT} \end{cases}$$

T is the time constant of these filters, and is a parameter to tune. The higher T, the more influent the kinematic estimator will be in the wash-out filter. The value of T is tuned using the process described in section 1 minimizing the mean rms error between the output and the real value into all the scenario.

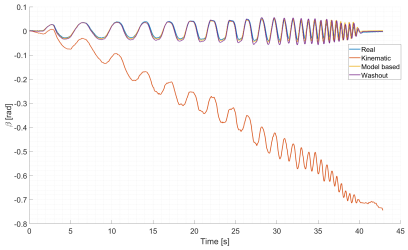
The results are represented figure **??**. The simulink structure is represented figure 7. This estimator is compared to the two previously computed model based and kinematic. The wash out estimator combines the advantages of both estimator. It doesn't drift and it follows better the complex variation. However, a fixed filter coefficient T limits the adaptability of the estimator, who doesn't know whether the model based or the kinematic is the most fitted in each situation.



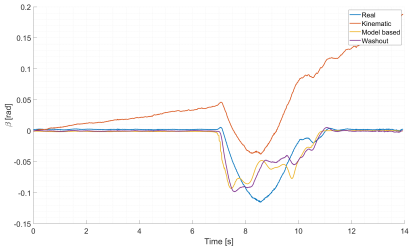
(a) Scenario 1



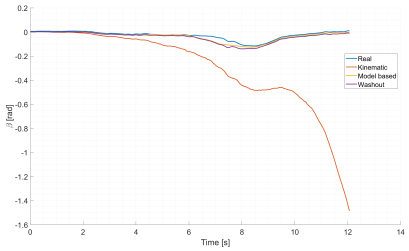
(b) Scenario 2



(c) Scenario 3



(d) Scenario 4



(e) Scenario 5

Figure 6: Performance of the washout filter estimator

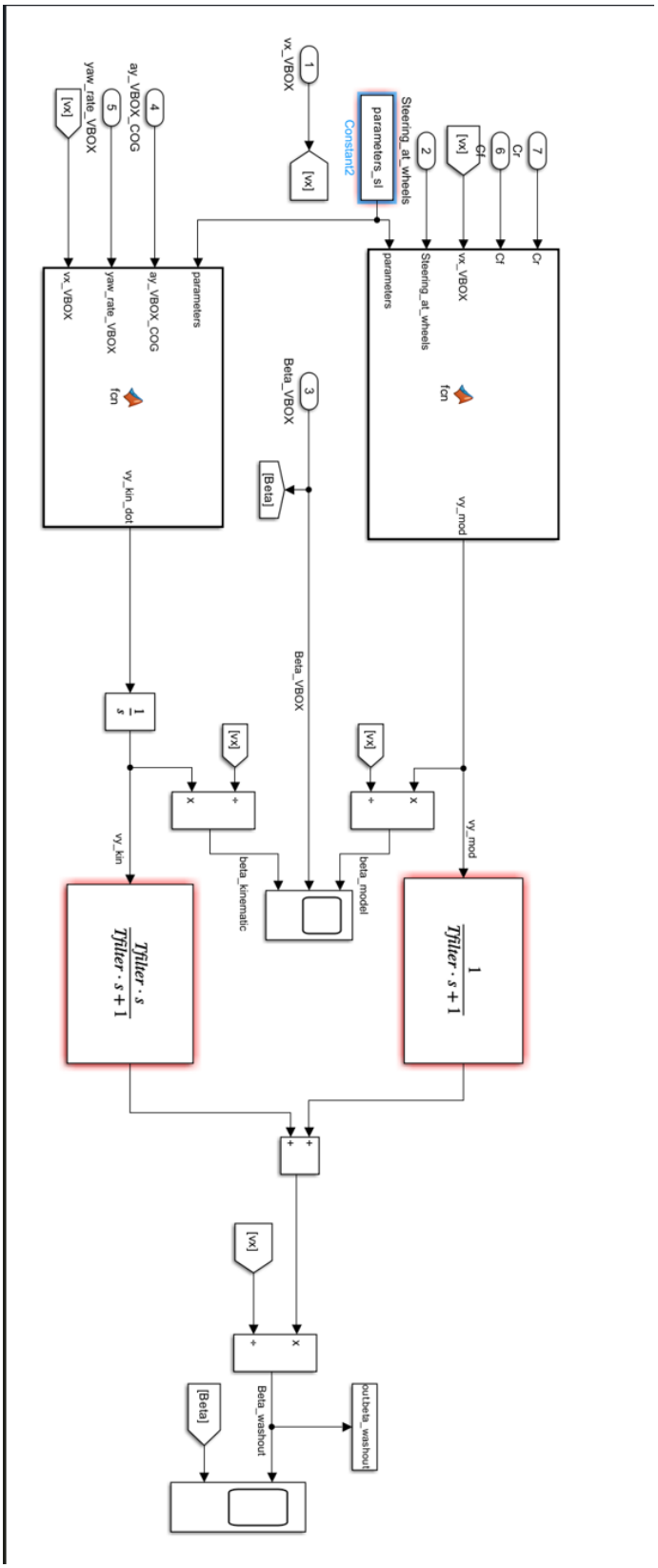


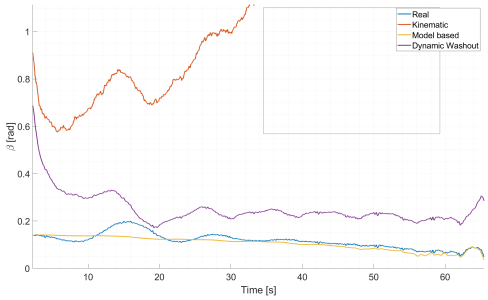
Figure 7: Simulink structure of the washout filter based estimator

Another idea to improve the accuracy of this estimator is to change the value of T dynamically. T should be higher in the transient part of the dynamics, because we want the kinematic velocity to be more important. Under the assumption that higher value of steering angle δ_{sw} means large variation of body slip angle, the filter coefficient T can be a function of δ_{sw}, the following equation describes the chosen link between T and δ_{sw}.

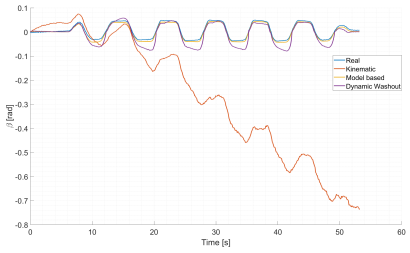
$$T(t) = a \cdot |\delta_{sw}(t)|$$

a is a coefficient to be tuned using the process described in 1. The value of T is to be saturated to a minimum value close to zero to ensure convergence. The results of this estimator are represented figure 8. The simulink structure is represented figure 9. This new estimator is more accurate in scenario 4. However in the remaining scenario, it tends to trust the kinematic estimator too much. It is probably due to the fact that for a constant steering wheel angle δ_{sw}, we don't have any transient dynamic but the value of the T will be high. Some others parameters have also been tested, like the filtered derivative of the lateral acceleration and the filtered derivative of the steering angle. However it didn't produce any satisfactory results. A suggestion to complete this study would be to use a quadratic model : $T(t) = a \cdot |\delta_{sw}(t)|^2$, or to combine multiple parameters into this equation.

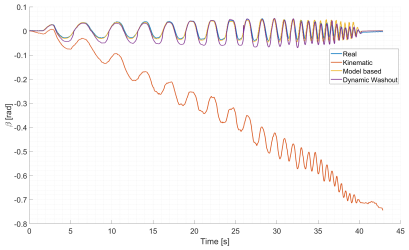
Figure 8: Performance of the dynamic washout filter estimator



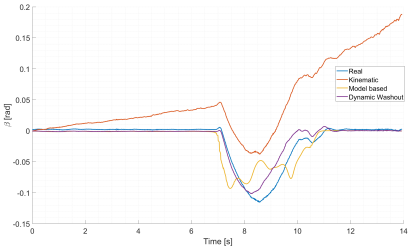
(a) Scenario 1



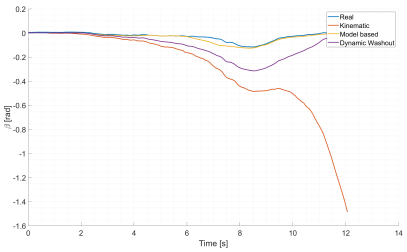
(b) Scenario 2



(c) Scenario 3

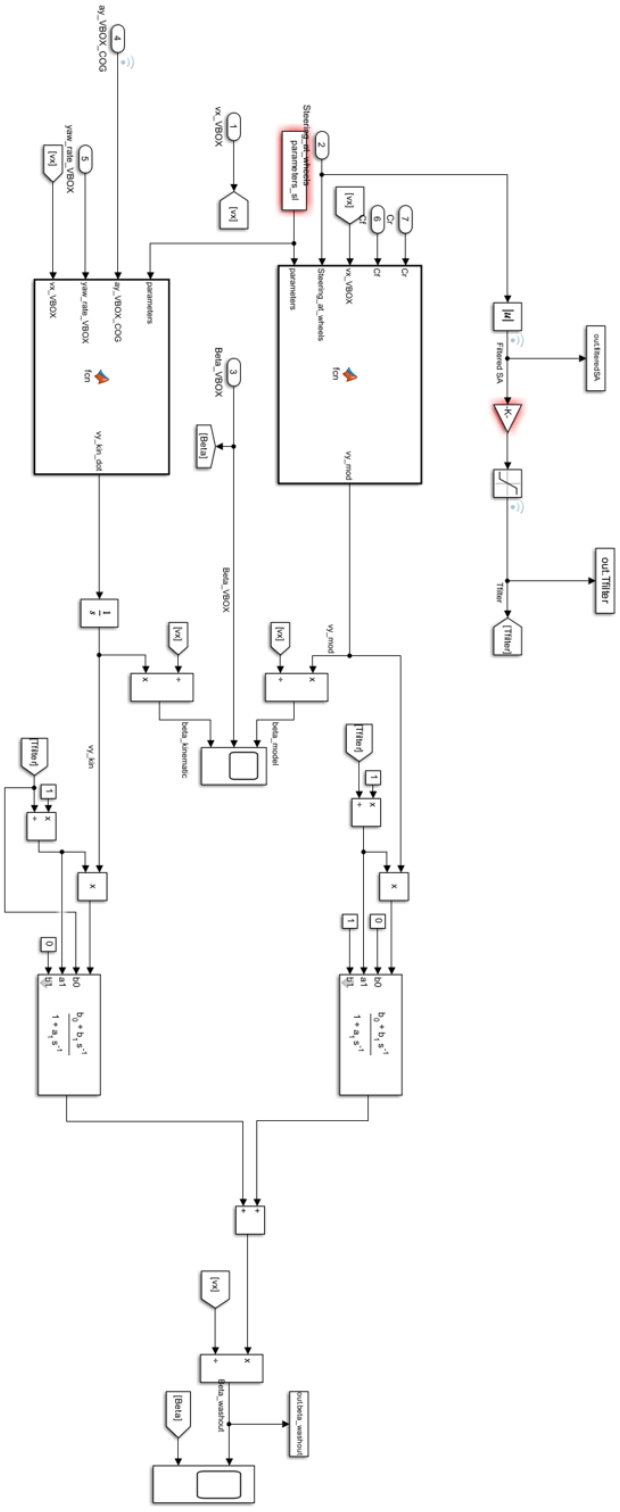


(d) Scenario 4



(e) Scenario 5

Figure 9: Simulink structure of the washout filter based estimator



3 Task 2 : Kalman Filter estimation

3.1 Squared-root Cubature Kalman Filter estimation

In this section, a squared-root Cubature Kalman Filter estimation is implemented to estimate the vehicle body slip angle. A Kalman filter is a type of estimator that combines model prediction and measurement to estimate a parameter. It requires two different equations:

- A model equations that predicts the next state using the control input under certain assumptions;
- A measurement equation that links the measurements data and the estimated state values.

In this case, the bicycle model is used again for the same reason mentioned above. The state variables are v_x, v_y and $\dot{\psi}_z$, the measurement variables are v_x, a_y and $\dot{\psi}_z$, and the control input is δ. The state equations are:

$$\begin{cases} m(\dot{v_x} - \dot{\psi}_z v_y) = -F_{12} sin(\delta) \\ m(\dot{v_y} + \dot{\psi}_z v_x) = F_{12} cos(\delta) + F_{34} \\ I_z \ddot{\psi}_z = f F_{12} cos(\delta) - b F_{34} \end{cases} \tag{6}$$

In these equations one can isolate the derivative of the states variable, and an Runge Kutta integration is used to predict the next state.

The measurement equations are the following :

$$\begin{cases} v_x^{measure} = v_x^{state} \\ a_y^{measure} = \dfrac{F_{12}^{state} cos(\delta) + F_{34}^{state}}{m} \\ \dot{\psi}_z^{measure} = \dot{\psi}_z^{state} \end{cases} \tag{7}$$

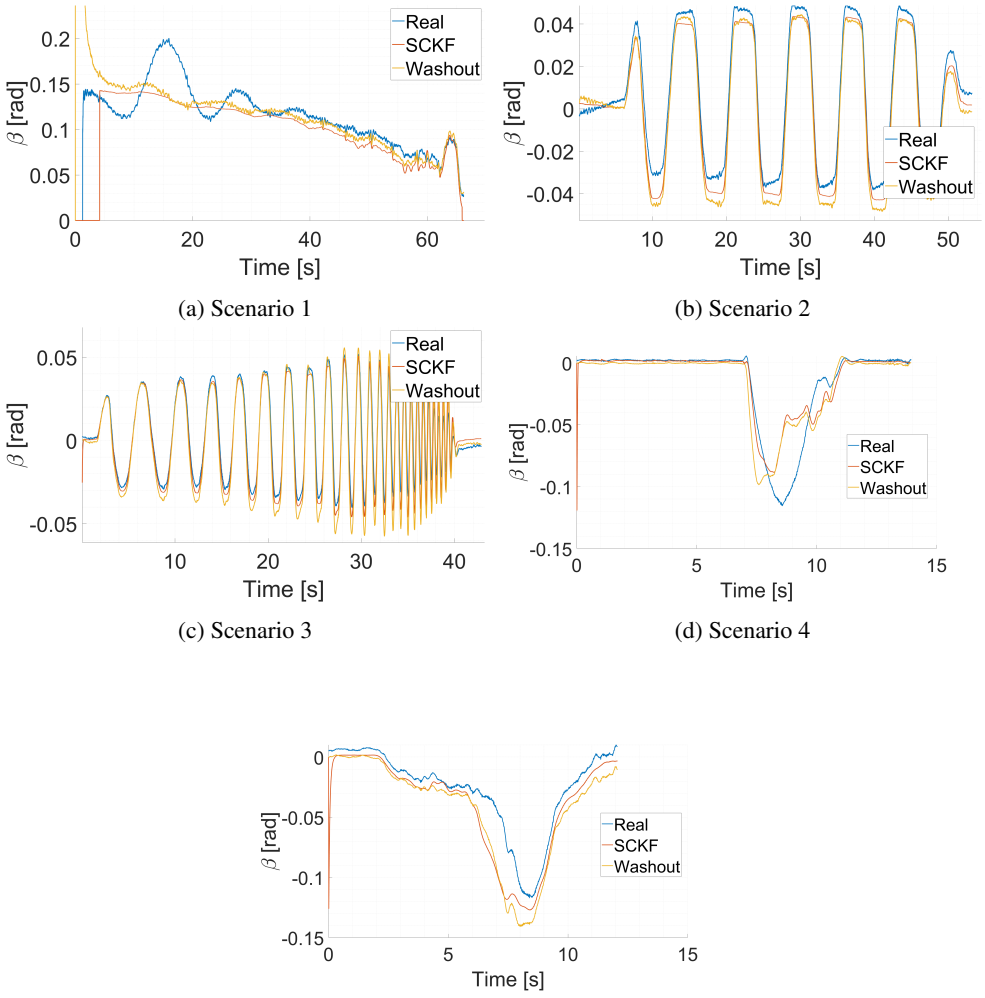
The estimator also requires some parameters to evaluate how robust the model and the measurements are. The process noise (Q) and the measurement noise (R) covariance matrices are two 3x3 matrices that need to be tuned, with only the diagonals being non-zero coefficients. Thus, a 6-dimensional optimization process, as the one described in section 2.1, has been implemented. Unfortunately, the results of this optimization are not satisfactory, as the problem itself is highly computationally demanding, and the gradient-based algorithm oscillates a lot. Therefore, it has been decided to move to a coarse hand-made optimization that still led to satisfactory results. The diagonal parameters of Q and R are reported in the following table:

Table 3: Process noise and measurement noise covariance matrices parameters

Q_1	Q_2	Q_3	R_1	R_2	R_3
10	0.001	30	0.1	0.1	0.01

From the SCKF results, it has been noticed that for vehicle longitudinal speeds below the $2m/s$ the estimator significantly drifts. Since the estimation of β for these

low speed values is not relevant, its value has been set equal to 0.
The results are represented figure 10. The SCKF estimator follows accurately the real value of β. It doesn't suffer of any drifting problem like the kinematic based estimator. Similarly to the washout filter based estimator, the SCKF struggle more in the high variation part of the dynamic, but it is still more accurate in this part than the washout filter based estimator.



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

(e) Scenario 5

Figure 10: Performance of the SCKF estimator

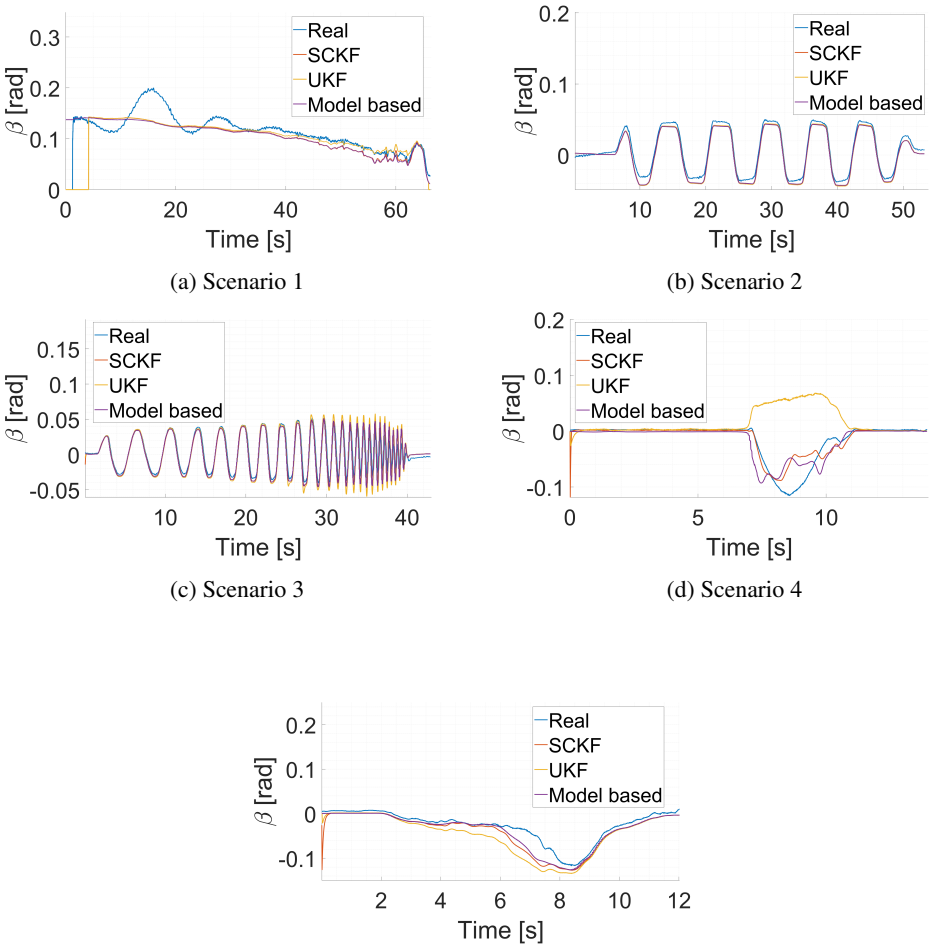
3.2 Unscented Kalman Filter estimation

The Unscented Kalman Filter is another estimation method very famous for its stability properties. It is pretty similar in the working principle to the SCKF. The main difference between the two is the generation of the sigma points that are propagated through the functions. The UKF is expected to lead to slightly better results, despite of a slightly higher instability and of a more complex tuning phase.
For this observer, the same process noise and measurement noise covariance ma-

trices parameters have been selected as in the SCKF, since it has been noticed that pretty satisfactory results were still obtained.
The other parameters to tune are α, β and κ. The following values have proven to lead to good results:

$$\alpha = 100,\ \beta = 100,\ \kappa = 100$$

The estimations obtained by the implementation of the UKF are shown here:



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

(e) Scenario 5

Figure 11: Estimations buy optimal quadratic coefficients

It is noticeable how the observer is the most reliable in the first run, attaining really similar results to the SKF in runs 2 and 3.
Concerning the run 4, instead, the results significantly worsen. This outcome is pretty strange and requires further investigation that will be carried out in a future work.
In the last scenario, the UKF works still well,but with more poor results than the SCKF.