Inversion of Vehicle Steering Dynamics with Modelica/Dymola

Tilman Bünte[1] Akin Sahin[2] Naim Bajcinca[1]
[1] German Aerospace Center (DLR), Institute of Robotics and Mechatronics,
Oberpfaffenhofen, D-82230 Wessling, Germany
[2] University of Siegen

Abstract

The task of steering a vehicle is an exercise which is usually considered hierarchically in terms of the two subtasks *path planning* and *path following*. With the driver in the loop some essential man dependent tasks such as sensing, information processing, and motor function affect the steering quality. In case of simulations, the same applies correspondingly for driver models. In this paper the aim is to investigate vehicle steering dynamics independent of any driver-related properties. The path is therefore assumed given by a *reference trajectory* together with a speed profile. The steering angle which is necessary for exact or at least approximate path following is sought after. This allows for plausible comparative assessment of different vehicle's steering dynamics in terms of the demanded steering effort for a certain maneuver. On the other hand, this approach requires dynamic inversion of vehicle steering dynamics which represents the main focus of this paper. Two vehicle models, the common single track model and a detailed model from the Modelica vehicle dynamics library are investigated. Since exact inversion of the detailed vehicle model turns out not to be feasible, approximate inversion is accomplished by means of a novel control structure called *inverse disturbance observer*. Simulations of a double lane change maneuver are conducted for illustration. Finally, wavelet power spectra of the steering angle signal are used for steering effort assessment.

1 Introduction

In the usual way of simulating vehicle models, a driver module provides inputs to the vehicle in terms of the steering wheel angle and gas/brake pedal position. As a result of this forward simulation, a trajectory of the vehicle is obtained. With *inverse simulation of vehicle steering dynamics* for a given desired trajectory and velocity profile, the aim is computation of the steering wheel angle input required from the driver.

Reference trajectories may be defined in terms of the curvature ρ as a function of the arc length λ. The reference trajectory of a *double lane change* maneuver is presented as an example. For tracking the reference path with a lateral displacement τ, instead of Cartesian coordinates a trajectory based coordinate system (λ, τ) is employed. In section 2, the representation of reference trajectories and the trajectory based coordinate system are explained in detail. For developing and investigation of the concept of vehicle steering dynamics inversion, two vehicle models are considered: the common linear *single track model* and a *detailed vehicle model* from the Modelica vehicle dynamics library. These models are introduced in section 3.

If some requirements like regularity and uniqueness of solutions hold, inverse models may be obtained in Modelica by simply providing equations for the outputs and removing an adequate number of equations for the original inputs. The perfect inverse of the detailed vehicle model from the Modelica vehicle dynamics library (using rigid linkages for the suspensions) is easily achieved. However, it turns out that the detailed vehicle model is non-minimum phase. Therefore, the inverse vehicle model is unstable and can not be simulated. To overcome this problem, as a trade-off we use approximate inversion of models, such that the resulting system is stable. For this purpose, a novel high gain control scheme, the *inverse disturbance observer* [1] is utilized. The inverse disturbance observer combines exact inversion of a simplified model as feedforward control and high-gain feedback for robust tracking performance. Simulation results for a double lane change maneuver illustrate the effectiveness of the applied approach in section 4.

Steering dynamics of different vehicles may be compared in terms of the steering inputs being necessary to perform a specific maneuver. The objective is to establish a method which can be used to assess the steering dynamics of vehicles with specific modifications like active steering control. Therefore, in section 5 the double lane change steering inputs are compared for two single track models with significantly different loading. For analyzing the steering efforts of the two vehicles, wavelet transform is applied. Conclusions on easiness or difficulty for a driver when driving these cars can be drawn from wavelet power spectra.

2 Reference trajectories and coordinate system for path tracking

For inverse vehicle simulations investigated in this paper, the vehicle's speed and a reference trajectory for the vehicle's position are given. The reference point on the vehicle representing its position is assumed to be located at the center of the front axle. With perfect inversion, this reference point exactly follows the reference trajectory, otherwise the task is to make the lateral displacement from the reference trajectory as small as possible. Therefore, this problem is closely related to the problem of path tracking for automatic car steering.

2.1 Reference trajectories

In this paper, the reference trajectory is defined in Cartesian coordinates $(x_{ref}(\lambda), y_{ref}(\lambda))$ as a function of the arc length λ. Any reasonable trajectory of a vehicle cruising at finite speed may be assumed continuous and at least twice differentiable. With $\rho(\lambda)$ and $\phi(\lambda)$ denoting the curvature and the track angle respectively, the following relations hold:

$$\begin{bmatrix} \phi' \\ x'_{ref} \\ y'_{ref} \end{bmatrix} := \frac{\mathrm{d}}{\mathrm{d}\lambda} \begin{bmatrix} \phi \\ x_{ref} \\ y_{ref} \end{bmatrix} = \begin{bmatrix} \rho \\ \cos(\phi) \\ \sin(\phi) \end{bmatrix} \qquad (1)$$

Our approach is to start from a definition of $\rho(\lambda)$ and solve (1) for ϕ, x_{ref}, and y_{ref} using appropriate initial conditions. See Fig. 1 for an exemplary definition of $\rho(\lambda)$ and the resulting reference trajectory (x_{ref}, y_{ref}) for a *double lane change* maneuver.

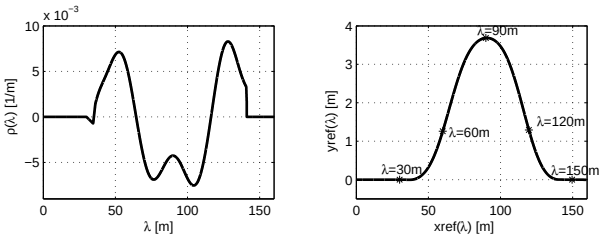

Figure 1: Curvature (left) and reference trajectory (right) for a double lane change.

2.2 Coordinate system for path tracking

For the mathematics involved with the path tracking problem, it is not expedient to describe the vehicle's position with Cartesian coordinates. Therefore, rather a trajectory based coordinate system (λ, τ) is employed, see Fig. 2. It consists of the arc length λ referring to the point $(x_{ref}(\lambda), y_{ref}(\lambda))$ on the reference trajectory which is closest to the vehicle and the lateral displacement τ, also referred to as tracking error. That is, τ is the signed closest perpendicular distance to the reference trajectory.

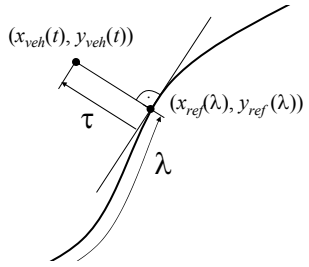

Figure 2: Vehicle position in trajectory based coordinates (λ,τ).

A coordinate transformation between Cartesian coordinates (x_{veh}, y_{veh}) and trajectory based coordinates (λ, τ) needs to be accomplished. The unit vector $[-y'_{ref}, x'_{ref}]^T$ is perpendicular to the reference trajectory and is oriented to the left hand side of the trajectory. Hence, the distance between the position of the vehicle and the reference trajectory may be written as

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} := \begin{bmatrix} x_{veh} - x_{ref} \\ y_{veh} - y_{ref} \end{bmatrix} = \tau \begin{bmatrix} -y'_{ref} \\ x'_{ref} \end{bmatrix}. \qquad (2)$$

The coordinate transformation can be done in the following way: Elimination of τ in (2) yields the nonlinear equation

$$\Delta x x'_{ref} + \Delta y y'_{ref} = 0 \qquad (3)$$

which can be solved for λ.

Using the fact $x'^2_{ref} + y'^2_{ref} = 1$ (see (1)) together with (2) yields

$$\tau = \Delta y\, x'_{ref} - \Delta x\, y'_{ref}. \tag{4}$$

Multiple solutions may exist for equation (3). Only the closest solution where $|\tau|$ has its minimum value is relevant and is to be selected. This ambiguity makes evident that the introduced trajectory based coordinate system is only suitable in a sufficiently narrow vicinity of the reference trajectory. This assumption holds, since accurate path tracking is aimed at.

Later, the coordinate transformation will be considered a part of the vehicle models. Linearization, as may be necessary, is done in the following way. A virtual object exactly following the reference trajectory as defined in section 2.1 senses a lateral acceleration given by

$$a_{yref} = \rho(\lambda)\,\dot{\lambda}^2 \tag{5}$$

with $\dot{\lambda}$ denoting the object's speed. Under the assumption of small tracking error τ and small chassis side slip angle the lateral acceleration of a vehicle closely tracking the reference trajectory with speed v (entailing $v \approx \dot{\lambda}$) can therefore be represented by

$$a_{yveh} = a_{yref} + \ddot{\tau}. \tag{6}$$

Hence,

$$\tau = \frac{1}{s^2}\left(a_{yveh} - a_{yref}\right). \tag{7}$$

2.3 Implementation in Modelica

During the simulation the actual value of λ needs to be solved from equation (3) for each integration step. This is automatically done by Dymola, provided that $x_{ref}(\lambda)$, $y_{ref}(\lambda)$, $x'_{ref}(\lambda)$ and $y'_{ref}(\lambda)$ are known. Therefore, in our Modelica model we provide look-up tables depending on λ that contain values for x_{ref}, y_{ref}, and ϕ each with the derivative w.r.t. λ. These look-up tables are pre-calculated from (1) in Matlab, saved to mat-files, and used in Modelica/Dymola for interpolation at simulation time. According to our experience, the selection of the proper solution of (3) does not cause any problems since the solution for λ is continuously and monotonically increasing along the followed reference trajectory.

A special problem occurs with the simulation of perfectly inverted vehicle models (see section 4.1). In general, for inverse simulations executed in Dymola the given output where required needs to be differentiated one or multiple times w.r.t. time. The look-up tables we use, however, only provide derivatives w.r.t. λ since the reference trajectory does not depend on time. Therefore, if needed the time derivatives are supplied by special functions[1]. They are calculated from the actual value of $\dot{\lambda}$ and the corresponding derivatives w.r.t. λ. If necessary, higher derivatives w.r.t. λ are supplied in extra columns in the look-up tables. As an example the Modelica code

```
dxdlambda = TableFunc.y(tableIDintx, 3,lambda);
```

is used to retrieve x'_{ref} from the look-up table (referred to by its identifier *tableIDintx*, 3rd column stores first derivative) for the actual value of λ. This is the used package:

```
package TableFunc
 function y // here y means a generic output
   input Integer ID, index;
   input Real u;
   output Real y;
 external "C" y=
       dymTableIpo1_my(ID,index,u);
   annotation (derivative=ydot);
 end y;
 function ydot
   input Integer ID,index;
   input Real u,dudt;
   output Real dydt;
 protected
   Real dydu;
 algorithm
   dydu :=
       dymTableIpo_my(ID,index+1,u);
   dydt := dydu*dudt;
   annotation
       (derivative(order=2)=yddot);
 end ydot;
 function yddot
 ... // analogous to ydot
 end yddot;
 function dymTableIpo_my
   input Integer ID,index;
   input Real u;
   output Real y;
 external "C" y=
       dymTableIpo1_my(ID,index,u);
 end dymTableIpo_my;
end TableFunc;
```

The C function *dymTableIpo1_my* provides the table look up. It corresponds to *dymTableIpo1* which can be found in *dymtable.c* in the Dymola source directory. Note the annotations. The standard way of differentiating inputs from look-up tables is thus replaced by use of the function *ydot* (*yddot* respectively) while applying the chain rule.

[1]The authors thank Andreas Pfeiffer (DLR) for his helpful support.

3 Vehicle Models

Vehicle steering dynamics in this paper are explored using two models with essentially different levels of detail. Firstly, for basic considerations the very simple *single track model* is implemented in Modelica. Secondly, for more advanced investigations, a *detailed vehicle model* from the Modelica vehicle dynamics library is used. In both cases, the trajectory which normally is the output of a driving maneuver is defined together with a speed profile and the necessary steering input is asked for. Therefore, both the reference trajectory and the coordinate transformation as defined in section 2 are added to the model description.

3.1 The linear single track model

The single track model [2] is a simple linear vehicle model commonly used in the analysis and control design of lateral and yaw dynamics. The wheels of the each axle are considered lumped together in the center of the vehicle. The roll, pitch, and heave motions are neglected. In Fig. 3 the single track model is illustrated. Its major variables and geometric

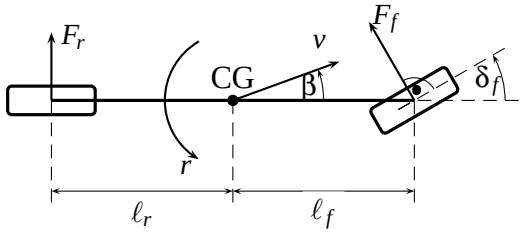

Figure 3: Single track model.

parameters are

$F_f(F_r)$	lateral wheel force at front (rear) wheel
ψ	yaw angle
$r = \dot{\psi}$	yaw rate
β	chassis side slip angle at center of gravity (CG)
v	speed, i.e. magnitude of velocity vector at CG
$\ell_f(\ell_r)$	distance from front (rear) axle to CG
i_L	steering gear ratio
δ_f	front wheel steering angle
$\delta_S = i_L\delta_f$	steering wheel angle

Linearizing the tire force characteristics lateral wheel forces at the front and rear wheels can be written as

$$F_f(\alpha_f) = \mu c_{f0}\alpha_f, \qquad F_r(\alpha_r) = \mu c_{r0}\alpha_r \tag{8}$$

with c_{f0}, c_{r0} being the tire cornering stiffnesses at the front and the rear wheels, μ the road adhesion factor and α_f and α_r the tire side slip angles at the front and the rear wheels given by

$$\alpha_f = \delta_f - \left(\beta + \frac{\ell_f}{v}r\right), \qquad \alpha_r = -\left(\beta - \frac{\ell_r}{v}r\right) \tag{9}$$

The mass of the vehicle is m and J is the moment of inertia w.r.t. a vertical axis through the CG. Under the assumptions of small side slip and steering angles and slowly varying velocity the linearized equations of motion are

$$\begin{bmatrix} mv(\dot{\beta}+r) \\ J\dot{r} \end{bmatrix} = \begin{bmatrix} F_f + F_r \\ F_f\ell_f - F_r\ell_r \end{bmatrix} \tag{10}$$

The lateral acceleration of the vehicle at the front axle is

$$a_{yveh} \quad = \quad v(\dot{\beta}+r) + \dot{r}\ell_f. \tag{11}$$

For linear considerations, (7) may be used for the lateral position w.r.t. the trajectory based coordinate system. Otherwise, the velocity of the vehicle's CG is

$$\begin{bmatrix} \dot{x}_{CG} \\ \dot{y}_{CG} \end{bmatrix} = v\begin{bmatrix} \cos(\psi+\beta) \\ \sin(\psi+\beta) \end{bmatrix} \tag{12}$$

The front axle (i.e. vehicle reference point) position is

$$\begin{bmatrix} x_{veh} \\ y_{veh} \end{bmatrix} = \begin{bmatrix} x_{CG} \\ y_{CG} \end{bmatrix} + \ell_f\begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \tag{13}$$

The *single track model* will be used later as a simple substitute for the fully detailed standard vehicle dynamics model from the Modelica vehicle dynamics library [3] (which is parametrized as a BMW 3-series car by default). The corresponding parameters for the single track model were determined in [4] and they are also used here: $i_L = 16.94$, l_f=1.0203m, l_r=1.5297m, $m = 1482.9$kg, $J = 2200\,\mathrm{kg\,m^2}$, $c_{f0} = 91776\,\mathrm{N/rad}$ and $c_{r0} = 77576\,\mathrm{N/rad}$. Only dry road conditions are considered here, therefore $\mu = 1$.

3.2 Detailed vehicle model

The vehicle dynamics library [3] of Modelica provides models for vehicle dynamics simulation. It consists of

a detailed mathematical model comprising the multibody differential equations. Since this library is freely available, documented and well known to the Modelica user community, no further details are stated here. In this paper the Modelica vehicle model described in [4] is used. The standard *chassis level 2* vehicle model is completed by the *simple power train* model and brakes. Furthermore, a PI speed controller sets an adequate gas/brake pedal position and makes the vehicle accurately follow a desired speed profile. Finally, a wheel slip controller approximates the function of an antilock braking system (ABS). In the sequel, this model will be referred to as the *detailed vehicle model*.

4 Perfect and approximate inversion of vehicle steering dynamics

The vehicle models used in this paper (see section 3) are considered as SISO (single input single output) systems with the steering wheel angle δ_S being the input and the lateral displacement τ from the reference trajectory being the output. The ideal conception of the model inversion process (referred to as *perfect inversion*) is to obtain a steering wheel angle signal such that the lateral displacement τ is always zero. Simulations executed with perfectly inverted models are denoted *inverse simulations* here. Inversion of the longitudinal dynamics (i.e. speed) may in general be considered as well. Here, however, we focus on steering (i.e. lateral) dynamics. Along the way, the vehicle speed v is set or controlled to match a given profile $v(\lambda)$ or alternatively $v(t)$. Hence, speed is rather considered a set varying parameter than an input or output. If perfect steering dynamics inversion is not possible, *approximate inversion* is aimed at. That is, the resulting lateral displacement τ and steering wheel angle error respectively should be as small as possible.

For both models, single track model and detailed model, we first try to achieve perfect inversion. As will be shown, this is possible for the single track model. In contrast, perfect inversion of the detailed model turns out not to be feasible in terms of a converging simulation. Therefore, a novel high gain control scheme is applied to approximately invert the detailed model. This approach may incidentally also be applied to the task of high fidelity path tracking for real world automatic car steering.

In the course of this section, simulations of the inverted models are conducted for the purpose of illustration. Exemplarily, the double lane change maneuver introduced with Fig. 1 is considered with a constant speed of 20m/s .

4.1 Perfect inversion of the vehicle models in Modelica/Dymola

The option of perfect inversion of Modelica models has already been exploited in a number of applications such as automatic generation of control laws for the control of aircraft [5] or industrial robots [6]. Inverse models may be obtained in Modelica by simply providing equations for the outputs and relaxing an adequate number of equations for the original inputs. As pointed out in [6], the derivation of the inverse system equations may require to differentiate certain parts of the model equations. Therefore, the model equations need to be continuous and differentiable. Moreover, since it may be necessary to the differentiate the given output signals too, their time derivatives must exist and be provided up to a certain order. Therefore, as explained in section 2.3, look-up tables for the trajectory variables and their derivatives w.r.t. λ are provided in the models together with functions to form the respective time derivatives.

With nonlinear models, for a given output not necessarily any solution in terms of input functions does exist. On the other hand, multiple solutions may exist for the same inverse simulation problem. So far, we have not worked on these questions. We have rather assumed conditions (i.e. moderate lateral acceleration) which do not cause corresponding problems.

One necessary condition for perfect inversion is that the considered input/output dynamics of the model is minimum phase. Otherwise the inverted model is not stable and therefore inverse simulation is not feasible.

4.1.1 Perfect inversion of the single track model

For investigating the perfect inversion of the single track model, the implementation of its equations and its parameters in Modelica as described in section 3.1 is employed. The model includes the reference trajectory look-up tables for the double lane change maneuver and the coordinate transformation (3),(4) introduced in section 2. The set of equations is completed by $\tau = 0$ and $v = 20$m/s and thus the number of equations matches the number of unknowns. The model

can be successfully translated and simulated. The resulting front steering angle δ_f is shown in Fig. 4. The parameters of the *light vehicle* are those given in section 3.1. For comparison, the simulation is repeated with a *heavy vehicle*. Its parameters are the same except for double values of mass m and inertia J.

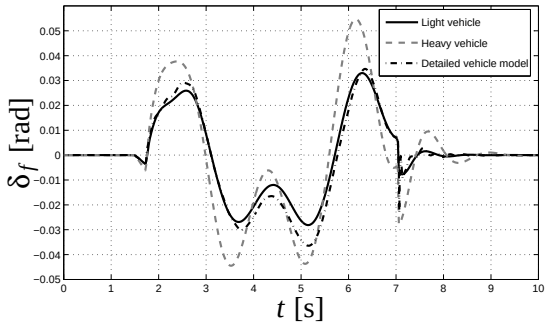

Figure 4: Front steering angle δ_f for the double lane change maneuver (v = 20m/s) obtained by inverse simulation of the single track model. Two parameter sets are used: *light vehicle* and *heavy vehicle*. Also results for the approximately inverted detailed vehicle model (see section 4.2.2) are shown.

4.1.2 Perfect inversion of the detailed vehicle model

The detailed vehicle model is inverted in the same way by adding the equation $\tau = 0$ and by setting the target value for speed control to 20m/s. The steering wheel angle is relaxed, i.e. any direct equation for driver steering input is removed.

We attempted to invert models with different suspensions. With the *SimpleSuspension* the translation of the model was successful. However, the integration in Dymola terminated 0.13s after start of the simulation due to missing convergence of the corrector. With the *MacPhersonSuspension2* Dymola was not able to differentiate some of the model equations, therefore, this inverse model could not be translated successfully. The last-mentioned problem was not investigated further since we found out, that the detailed vehicle model is non-minimum phase which causes stability problems at any rate when simulating its inverse. This is also the reason why the inverse simulation using *SimpleSuspension* did not converge.

For illustration of the non-minimum phase dynamics, the pole-zero-map of the transfer function from steering wheel angle δ_S to lateral displacement τ was investigated. The transfer function was obtained by linearisation about straight driving ($x(\lambda) = \lambda$, $y(\lambda) = 0$, $\delta_S = 0$, $r = 0$, $\psi = 0$, $\tau = 0$, $\dot{\lambda} = v = 20$m/s). The pole-zero map reveals a fast zero at $s \approx 90$ in the right half plane. The corresponding non-minimum phase behavior can be explained by the suspension construction of the steered front axle. It can briefly be depicted imagining an idle vehicle at zero speed. If the steering wheel is turned then the front end of the car moves slightly to the opposite direction due to the suspension's caster characteristic. In normal drive operation, this effect superimposes with the remaining vehicle steering dynamics and results in non-minimum phase behavior. When inverting the model, the right half plane zero becomes a fast unstable pole which makes simulation of the perfectly inverted model impossible. Therefore, in the next section a stable approximately inverted model will be generated using accurate path tracking control. For this purpose, a novel control structure denoted *inverse disturbance observer* is employed.

4.2 Approximate inversion of the detailed vehicle model

4.2.1 Inverse disturbance observer

The inverse disturbance observer (IDOB) was recently introduced in [1] as a modification of the common disturbance observer (DOB) structure. Basically, both

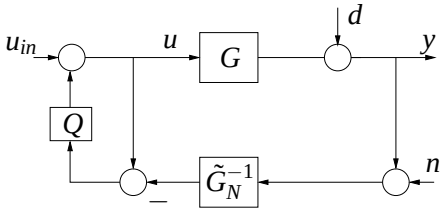

Figure 5: DOB scheme.

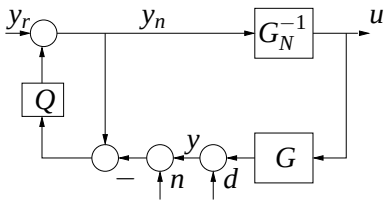

Figure 6: IDOB scheme.

DOB (see Fig. 5) and IDOB (see Fig. 6) are two degree of freedom control structures that combine high-gain and exact model inversion facilities in a simple configuration. The design parameters are an invertible nominal model G_N ($\tilde{G}_N$ respectively) approximating the

plant dynamics G (which is assumed to be stable) and a Q-filter which commonly has unity gain and low-pass properties. Compared with DOB, in the IDOB structure the block positions of the plant G and the nominal model G_N are simply interchanged (which partly gives a different meaning to the involved signals). Therefore, with IDOB the inverted nominal model G_N^{-1} is in the feedforward part instead of the feedback as it is the case with DOB.

DOB and IDOB structures are used for different purposes. The aim of the traditional DOB is matching the dynamics of the controlled system to a nominal model G_N. However, in case of IDOB the aim is matching the closed loop dynamics to G^{-1}. Therefore, the IDOB control structure is especially applicable for dynamic model inversion (in this case G represents the model to be inverted) and output tracking problems (in this case G represents a plant).

IDOB combines the facilities of feedforward control using an inverted nominal model of the plant and high gain feedback in a very simple structure while preserving the advantages of each principle. In the IDOB structure G_N^{-1} acts as a feedforward control. The term $G_N^{-1}(s) \cdot y_r(s)$ provides the main portion of the plant input $u(s)$ where y_r is the setpoint for y. The subordinate positive gain feedback loop containing the Q-filter forces this approximate inversion signal to converge to the perfect inversion signal and also provides robustness to the inversion process due to its high gain feedback feature.

The IDOB structure serves as an approximate model inversion method for a model G if the relation between the signal y_r and the plant input u is considered:

$$\frac{u}{y_r} = \frac{1}{G_N(1-Q)+GQ} \tag{14}$$

Recall that Q is a low pass filter with unity gain. The frequency interval between zero and the bandwidth of Q is denoted the *frequency operating domain* of the IDOB. In the frequency operating domain, $Q \to 1$ holds and therefore, $u \to G^{-1} y_r$. At high frequencies, the gain of Q tends to zero, therefore $u \to G_N^{-1} y_r$ which at least provides the input signal based on the model G_N. In the case that G is non-minimum phase and G_N is a minimum phase approximation for G, then by proper choice of the bandwidth of Q the stability of the IDOB system can be ensured. In practice, the bandwidth of Q will be chosen according to a compromise between (robust) stability and (robust) performance.

It can be concluded from (14) that for approximating perfect model inversion $u = G^{-1} y_r$ one of the following two criteria would be sufficient:

$$Q \to 1 \;\text{ or }\; G_N \to G \tag{15}$$

The IDOB structure combines the facilities of both high gain (subordinate loop with $Q \to 1$) and inversion with feedforward control $G_N \to G$ in the same structure. Also it is important to notice that with the IDOB structure, the approximate inverse of the model G is obtained without inverting the model explicitly.

On the other hand, considering y as the output of the system, IDOB becomes a plant controller for output tracking:

$$\frac{y}{y_r} = \frac{G}{G_N(1-Q)+GQ} \tag{16}$$

In the frequency operating domain, $Q \to 1$ holds and therefore, $y \to y_r$ i.e. good output tracking is achieved.

Due to its similar structure, the IDOB holds the known robustness properties of the disturbance observer in terms of disturbance and measurement noise rejection. Hence, the sensitivity (S) and complementary sensitivity (T) functions are the same as with DOB:

$$S = \frac{y}{d} = \frac{G_N(1-Q)}{G_N(1-Q)+GQ} \tag{17}$$

$$T = \frac{y}{-n} = 1 - S = \frac{GQ}{G_N(1-Q)+GQ} \tag{18}$$

Within the IDOB frequency operating domain ($Q \to 1$), disturbances are attenuated ($S \to 0$). For high frequencies ($Q \to 0$), noise is attenuated ($T \to 0$).

4.2.2 Application of IDOB for approximate inversion of the detailed vehicle model

As was shown in the last section, the IDOB needs a nominal model G_N. For approximate inversion of the detailed vehicle model by means of IDOB, the single track model is adopted as nominal model. It is easily invertible as already demonstrated in section 4.1.1. The actual single track model parameters (see section 3.1) were determined for good approximation of the detailed vehicle model [4].

However, the IDOB may not directly be applied to approximately invert the whole vehicle model since IDOB requires a stable plant but the vehicle dynamics with steering wheel angle δ_S as input and lateral

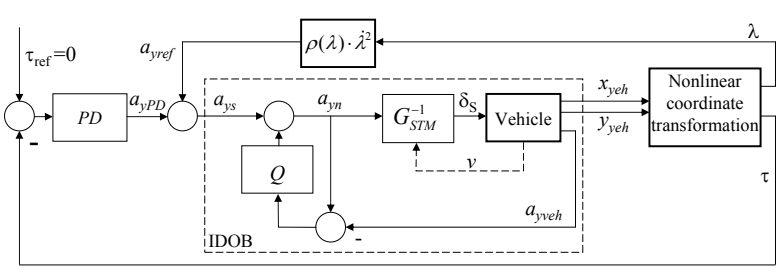

Figure 7: Path tracking control with IDOB.

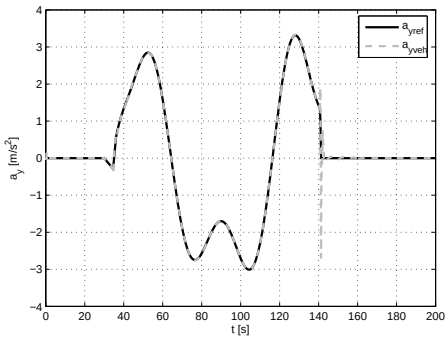

Figure 8: Lateral acceleration control with IDOB.

displacement τ as output involves two integrators, see (7). Based on (5)-(7), for approximate inversion of the whole detailed vehicle model we adopt a hierarchical control structure according to Fig. 7:

$$a_{ys} = a_{yref} - (K_d s + K_p)\tau \quad (19)$$

$$a_{yveh} = G_{IDOB} \cdot a_{ys} \quad (20)$$

A subordinate high bandwidth IDOB is used to make $a_{yveh} \rightarrow a_{ys}$. The Q-Filter is chosen a first order low-pass filter with a 0.03s time constant. An outer PD control loop with lower bandwidth compensates for the remaining tracking error τ. In the IDOB structure, henceforth only the stable part of the vehicle dynamics with output a_{yveh} is considered. $a_{y,s}$ is the set point for the inner IDOB loop and G_{STM} represents the single track model adopted as nominal model which corresponds to eqns. (8)-(11) . Note that the speed parameter of G_{STM}^{-1} is scheduled with the actual speed of the detailed vehicle model. δ_S is the steering wheel angle signal which is in the focus of interest. The reference lateral acceleration a_{yref} may be considered as a known external disturbance. Therefore disturbance feedforward compensation is applied according to Fig. 7. The resulting transfer function to τ is

$$\frac{\tau}{a_{yref}} = \frac{G_{IDOB} - 1}{s^2 + G_{IDOB}(K_d s + K_p)}. \quad (21)$$

Assuming that the bandwidth of the IDOB transfer function is sufficiently high ($G_{IDOB} \rightarrow 1$), the bandwidth and damping of the outer PD control loop may directly be affected by the PD parameters which are chosen as $K_d = 12$, $K_p = 36$.

Fig. 8 shows a simulation result of the approximately inverted detailed vehicle model performing the double lane change maneuver. The results are presented in terms of the actual vehicle lateral acceleration a_{yveh} which well tracks the reference lateral acceleration a_{yref} by virtue of the proposed IDOB based control. Remarkably, in this simulation the absolute value of the lateral displacement τ is less than 1.5mm (not depicted here). The steering wheel angle obtained is shown in Fig. 4 and can be well compared to the light vehicle single track model.

5 Comparative assessment of the steering dynamics using model inversion

In order to track a given trajectory with a given velocity profile, different vehicles potentially need different steering efforts. Therefore, using the inverse simulation results, steering dynamics of different vehicles may be compared in terms of the required steering efforts.

To illustrate our approach, the light vehicle and the heavy vehicle from section 4.1.1 are compared. The steering angles of these two models necessary to perform the double lane change maneuver with a constant speed of 20 m/s were given in Fig. 4. As it may be seen in this figure, the magnitude of the heavy vehicle steering angle is larger than that of the light vehicle during the maneuver. Moreover, especially in the time interval ca. $[3s, 6s]$ it is recognizable that the heavy vehicle needs to be steered slightly earlier than the light vehicle to follow the reference trajectory. That is, the look-ahead-time the driver needs to drive the heavy vehicle is larger compared to the light vehicle.

In the remainder of this section a method is established to quantify the conclusions mentioned above on the magnitude and look-ahead-time. Wavelets are used for appropriate time-frequency analysis of the steering angle signals.

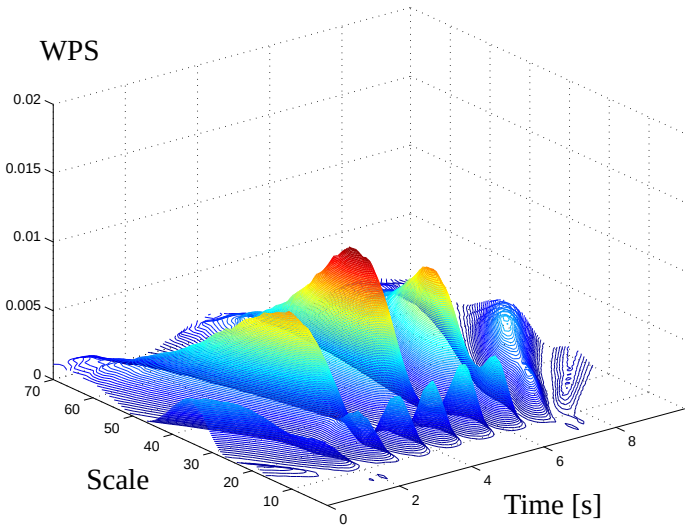

Figure 9: Wavelet power spectrum of the light vehicle steering angle using Morlet wavelet function.

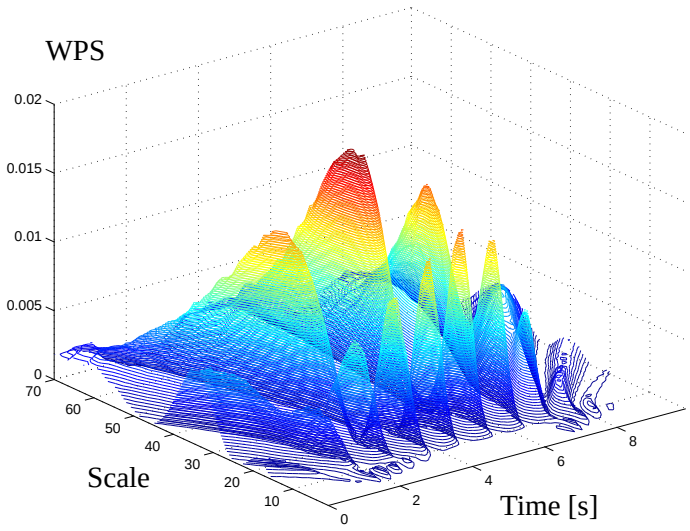

Figure 10: Wavelet power spectrum of the heavy vehicle steering angle with Morlet wavelet function.

5.1 Wavelet transform

Time-frequency analyses map the time domain signals into a two dimensional representation of energy versus time and frequency. Wavelet transform is a time-frequency analysis method that breaks a signal down into its constituent parts, wavelets, for analysis. Wavelets are oscillatory, scalable functions which are non-zero only within a limited spatial and Fourier regime. In the continuous wavelet transform, which is used in this paper, a wavelet is translated (time-shifted) through the signal. At each instant (i.e. time grid point) it is compared with the signal by means of evaluating the time integral of their product. This procedure is repeated for a grid of wavelets with different time scales. As a result, coefficients representing the similarity between sections of the signal and the scaled wavelet are produced. More detailed information on wavelets and wavelet transform may be found in [7], [8]. The wavelet transform returns a time-scale representation of the signal instead of the time-frequency representation. The scale is proportional to the reciprocal of the frequency. Large scales correspond to small frequencies and vice versa.

The single track model steering angle signals from Fig. 4 are now compared in terms of wavelet power spectra. At every instant, the time-scaled wavelet that locally best matches with the steering signal yields the maximum wavelet power spectrum value. Therefore, the local frequency content of the signal can be estimated from the scale value at which a local maximum occurs.

5.1.1 Wavelet transform of the steering angle signals

One of the basic problems in wavelet transform is choosing the appropriate wavelet function for the analysis of a given signal. In the wavelet transform of the steering signals, *Morlet* wavelet function is used, since it is recommended [9] for the analysis of time signals with smooth variations. In Figures 9 and 10 wavelet power spectra (WPS) of the steering angles of the light and heavy vehicles are given, respectively.

The two WPS are quite similar in terms of the scale and time locations of the local maxima, i.e. both signals have similar frequency contents at corresponding

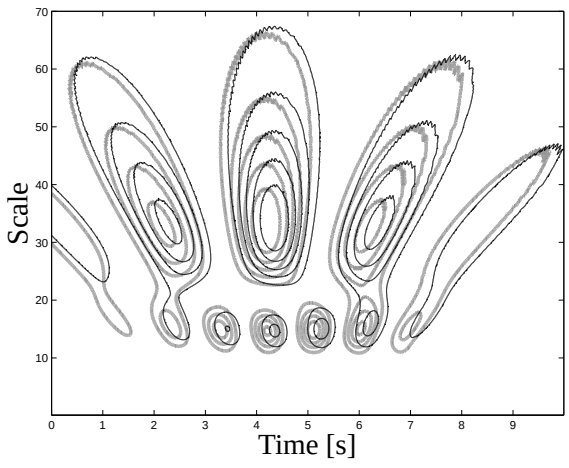

Figure 11: Wavelet power spectrum of the light vehicle steering angle (black lines) and eavy vehicle steering angle (gray lines) with Morlet wavelet function.

instants. However, almost throughout the entire time-scale domain, WPS of the heavy vehicle steering angle has higher power values compared to the light vehicle. This shows that the heavy vehicle needs more steering amplitude compared to the light vehicle all along the maneuver. This result coincides with the previous observation in the time-domain.

In Fig. 11 both WPS's are drawn in the same 2-D plot to make the differences between them better visible. The time of the local maxima can be more easily detected in Fig. 11. The power contour lines of the heavy vehicle steering angle are shifted to earlier instants by about 0.14s-0.18s compared to the light vehicle. This shows that the look-ahead-time needed to steer the heavy vehicle is correspondingly larger than for the light vehicle.

Using continuous wavelet transform with the Morlet wavelet function we are thus able to quantify the conclusions we already made from the time-domain plot of the steering angles. Another aspect in the comparison of the steering efforts is the frequency content of the signals. In Fig. 11 it may be noticed that there are mainly two accumulations of local maximum scale values which are at scale values 14 and 35. The scale values 14 and 35 corresponds 0.58 Hz and 0.23 Hz respectively which are frequencies that prevail in the signals. From Figs. (9, 10 it may be noticed that the steering angle of the heavy vehicle contains relatively higher power values at scale 14.

In other words, the heavy vehicle has to be steered with higher amplitudes, with a relatively larger portion of high frequencies and earlier (i.e. with more look-ahead-time) than the light vehicle. Hence, we conclude that the heavy vehicle is more difficult to drive.

6 Conclusions

Exact inversion of simulation models in principle is supported by Modelica/Dymola. However, it may be the case that models do not comply with the requirements to make inversion feasible. If so, approximate inversion may be an expedient way to still achieve useful results. High fidelity path tracking was demonstrated by means of the inverse disturbance observer based control. This provides a pretty accurate approximation of the steering angle signal which would result in perfect tracking. The time-scale wavelet power spectrum of the steering angle signal is an adequate basis for assessment of the steering effort.

Acknowledgment

The authors are grateful to Professor Wolfgang Wiechert from the University of Siegen Department of Simulation for academic supervision of the work [10] resumed in this paper.

References

[1] N. Bajcinca and T. Bünte, "A novel control structure for dynamic inversion and tracking," *Accepted for IFAC World Congress*, 2005.

[2] P. Riekert and T. Schunck, "Zur Fahrmechanik des gummibereiften Kraftfahrzeugs," *Ingenieur Archiv*, vol. 11, pp. 210–224, 1940.

[3] J. Andreasson, "Vehicle dynamics library," *Proceedings of the 3rd International Modelica Conference*, pp. 11–18, 2003.

[4] S. Heller and T. Bünte, "Modelica vehicle dynamics library: Implementation of driving maneuvers and a controller for active car steering," in *Proc. 3rd International Modelica Conference*, (Linköping, Sweden), 2003.

[5] G. Looye, "Design of robust autopilot control laws with nonlinear dynamics inversion," *Automatisierungtechnik*, pp. 523–531, 2001.

[6] M. Thümmel, M. Otter, and J. Bals, "Control of robots with elastic joints based on automatic generation of inverse dynamics models," *Proc. of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 925–930, 2001.

[7] R. Polikar, *The wavelet tutorial*. http://engineering.rowan.edu/∼polikar/WAVELETS/WTtutorial.html.

[8] S. Qian and D. Chen, *Joint Time-Frequency Analysis, Methods and Applications*. Upper Saddle River, NJ USA: Prentice-Hall, Inc., 1996.

[9] M. O. Domingues, "On wavelet techniques in atmospheric sciences," *First Latin American Advanced School on Space Environment*, 2004.

[10] A. Sahin, *Inversion of vehicle dynamics with Modelica*. University of Siegen, 2005. M.Sc. thesis.