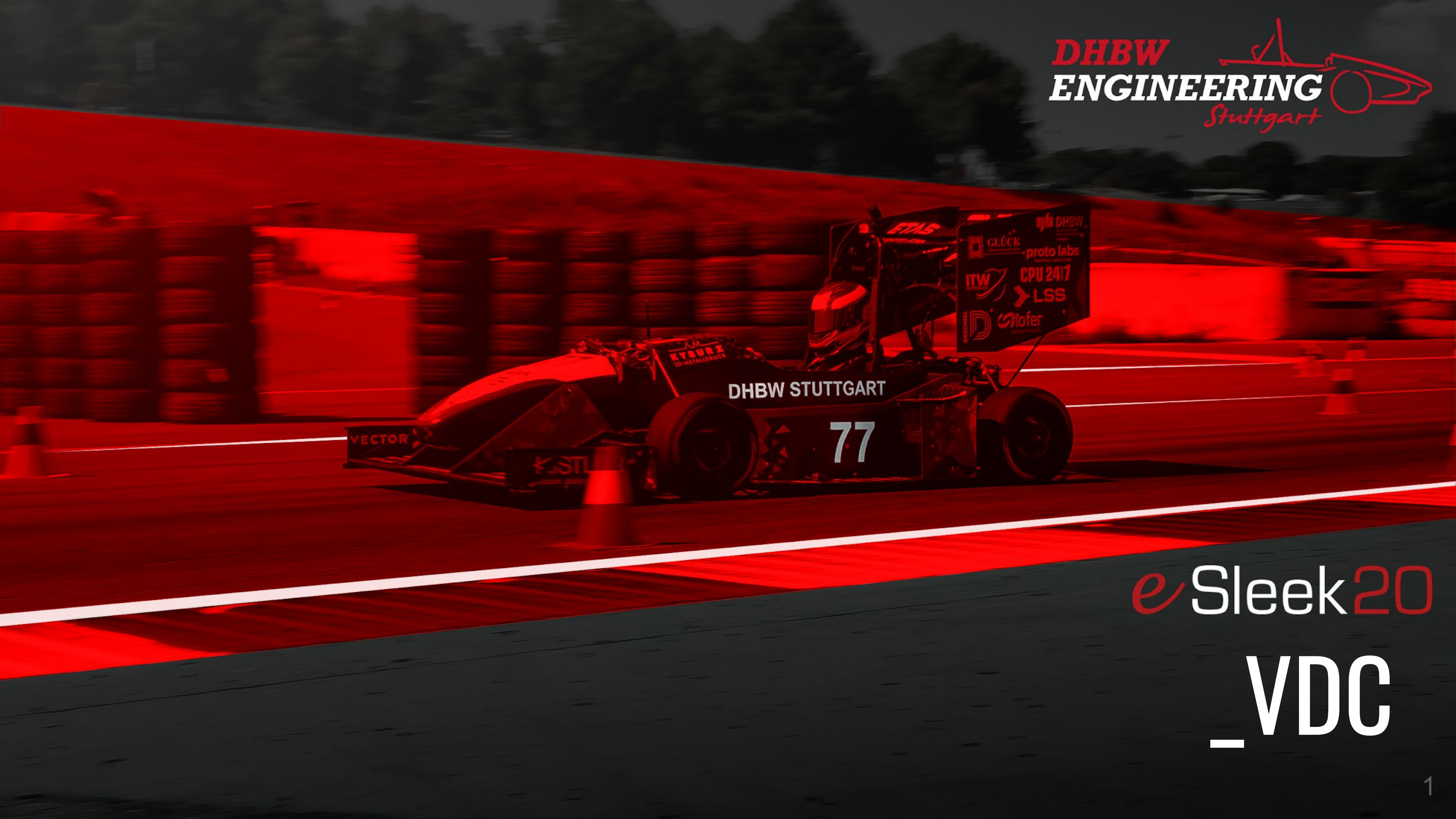


DHBW ENGINEERING Stuttgart

eSleek20

_VDC

DHBW ENGINEERING Stuttgart

eSleek20

State Estimation

CONCEPT

Goal: provide an accurate and reliable vehicle state estimate for VDC and DV software

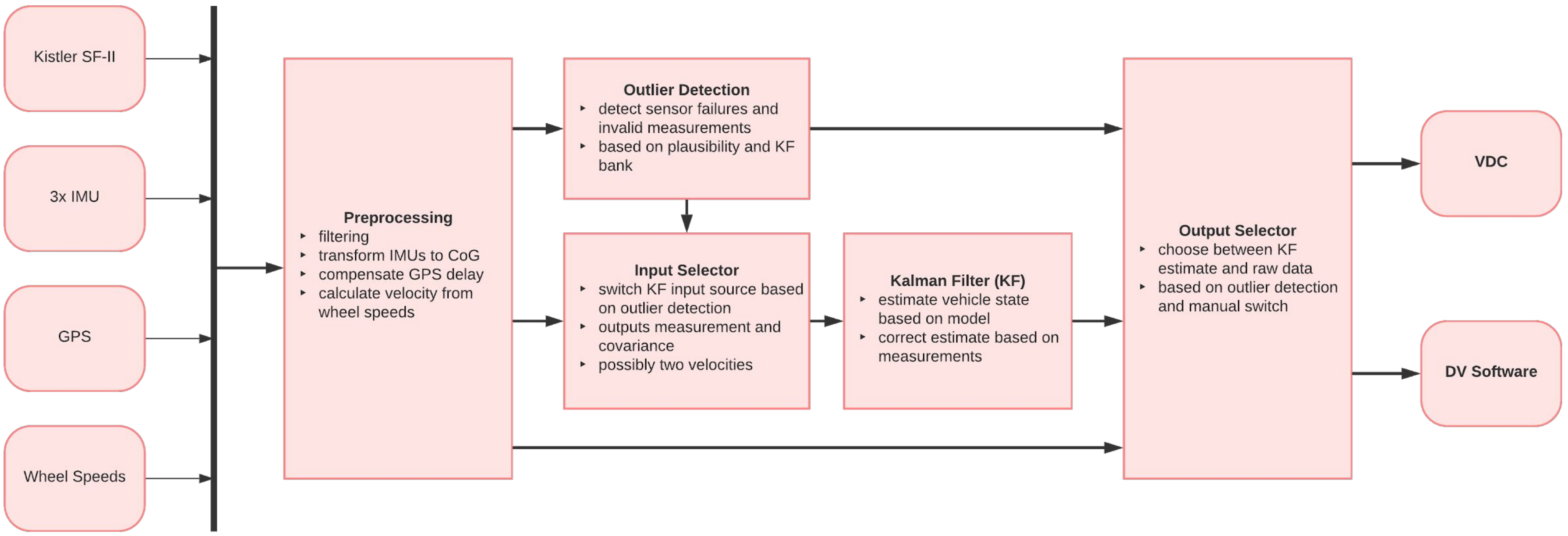
Guiding Principles:

- Flexibility to support different setups (EV/DV)
- Redundancy to handle sensor failures
- Clear architecture to facilitate maintenance and extension

New Features:

- Estimation of vehicle position and heading for DV
- Fusion of multiple velocity sources
- Robust outlier detection

DESIGN: Architecture



DESIGN: Vehicle Model

Position in x $\dot{x} = cos(\psi)v_x - sin(\psi)v_y$

Position in y $\dot{y} = sin(\psi)v_x + cos(\psi)v_y$

Velocity in x $\dot{v}_x = a_x + \dot{\psi}v_y$

Velocity in y $\dot{v}_y = a_x - \dot{\psi}v_x$

Yaw angle $\dot{\psi} = \dot{\psi}_{meas}$

Yaw rate $\ddot{\psi} = \ddot{\psi}_{meas}$

DESIGN: Signal Selection

Multiple modes for velocity measurements:

- EV
  - Normal: Kistler SF-II and GPS
  - Fallback: GPS and/or wheel speeds
- DV
  - Normal: DGPS and wheel speeds
  - Fallback: wheel speeds

Rule-based covariance matrix for wheel speed to compensate slip

DESIGN: Preprocessing

Mean/first order lowpass filtering if KF is not used

Transform IMUs to CoG, then average and calculate yaw acceleration

Outlier Detection:

- Range-based check
- Maximum plausible change rate
- Kalman Filter Bank

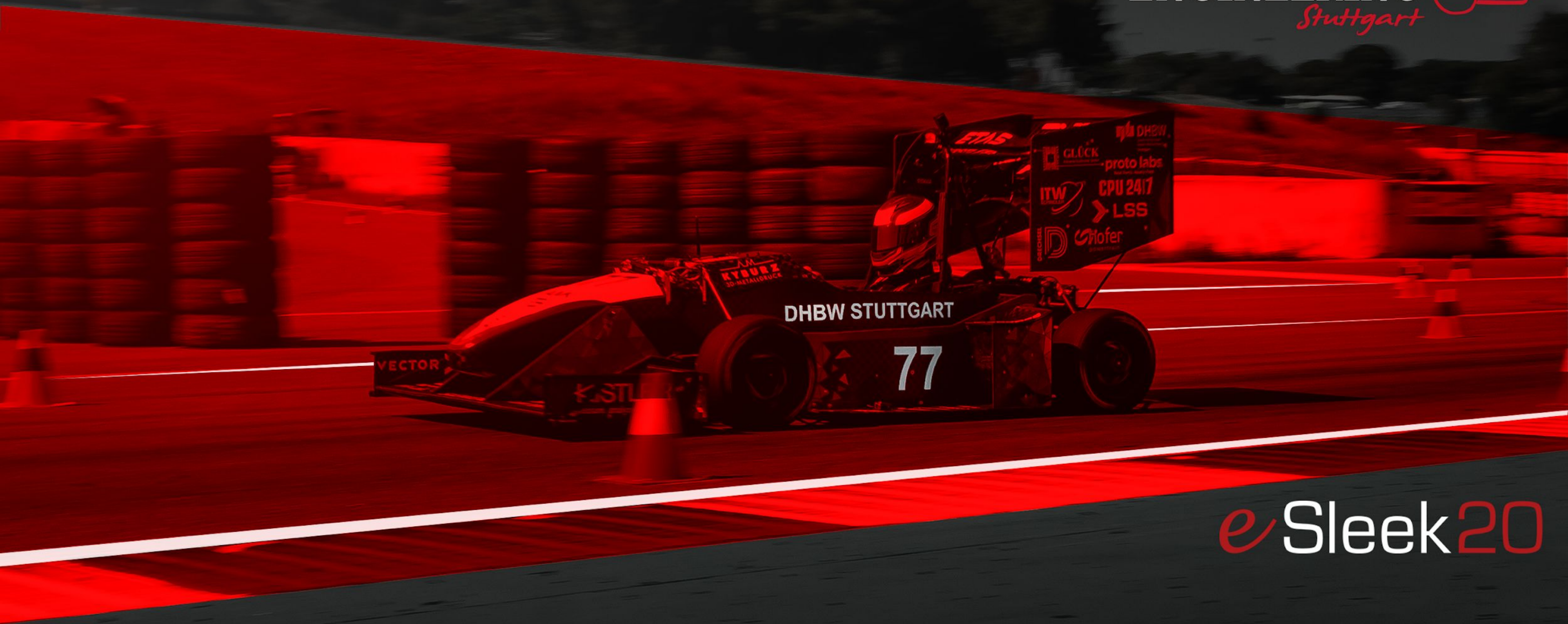
TESTING & APPLICATION

Use measurement data for initial design, refine during pre-season testing:

- Sensor noise covariances
- Filtering
- Approaches for wheel speed-based velocity calculation

Problem: no ground truth for vehicle state
→ base decisions on residuals and experience

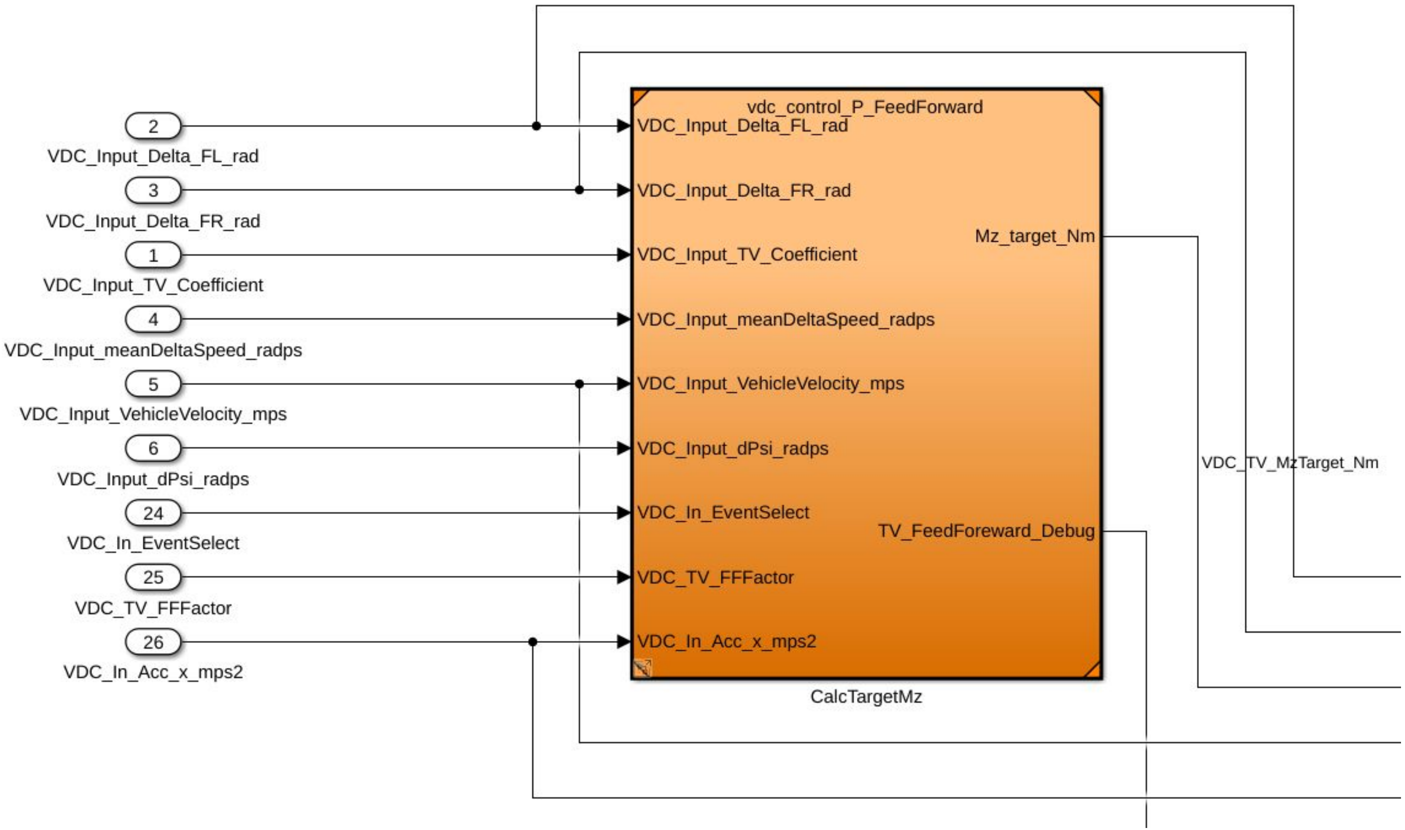
DHBW ENGINEERING Stuttgart



eSleek20

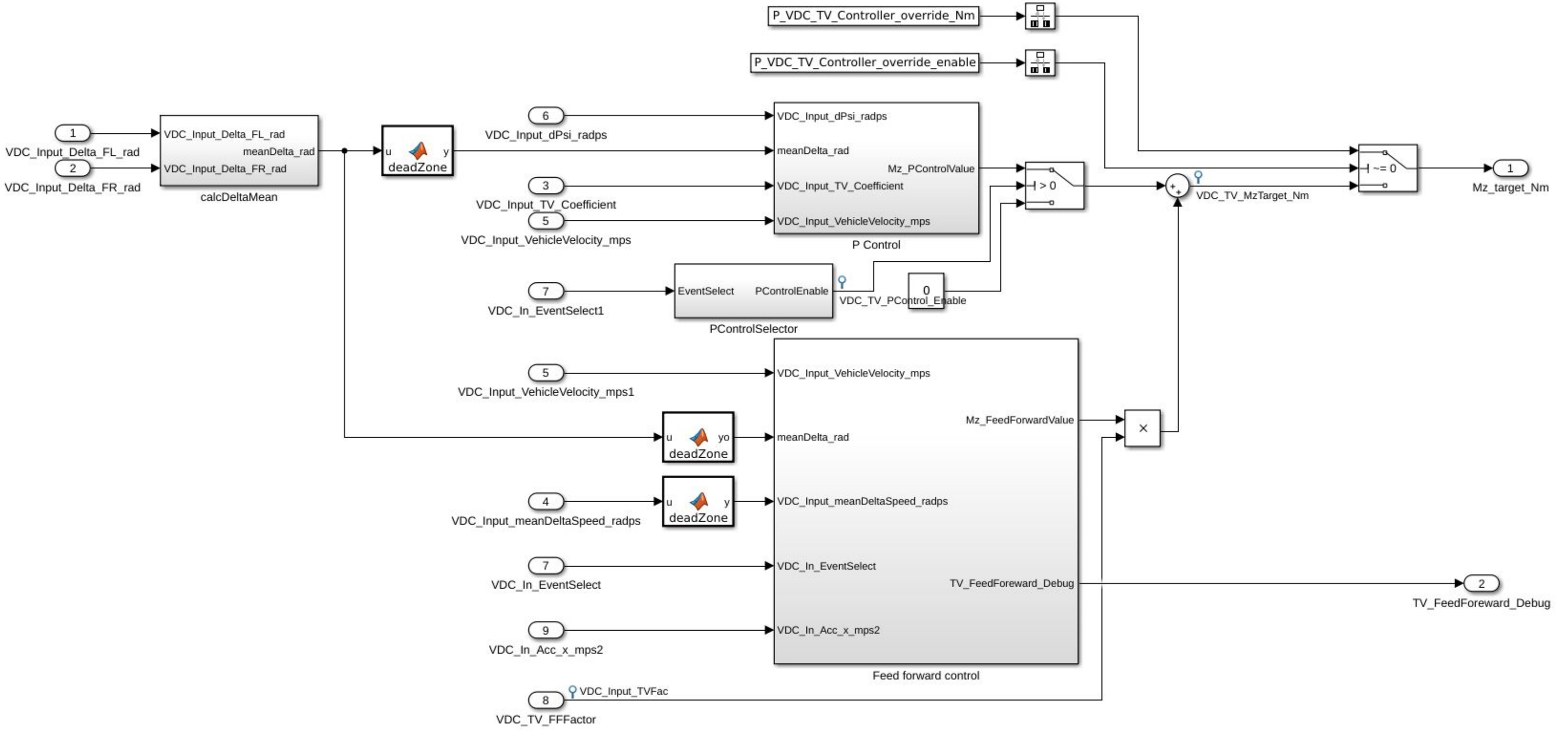
TV TARGET M_Z

CONCEPT



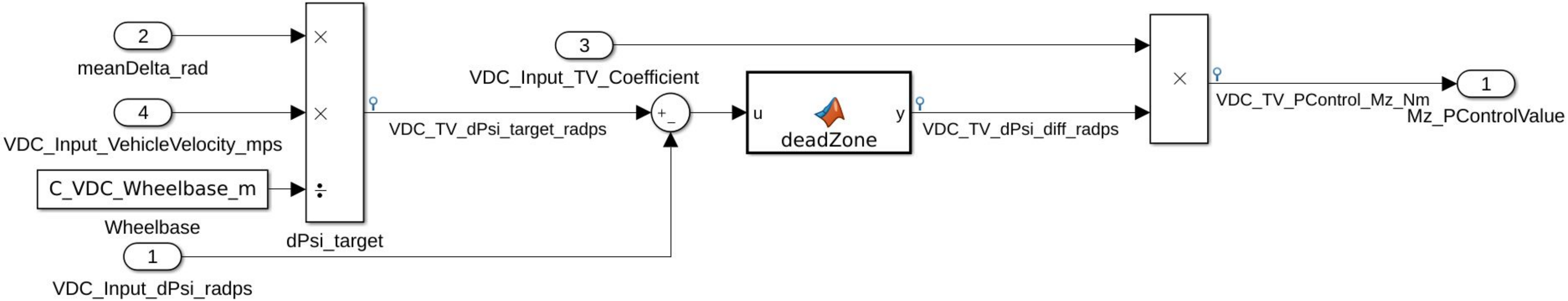
vdc_control_P_FeedForward
2
VDC_Input_Delta_FL_rad
3
VDC_Input_Delta_FR_rad
1
VDC_Input_TV_Coefficient
4
VDC_Input_meanDeltaSpeed_radps
5
VDC_Input_VehicleVelocity_mps
6
VDC_Input_dPsi_radps
24
VDC_In_EventSelect
25
VDC_TV_FFFactor
26
VDC_In_Acc_x_mps2
Mz_target_Nm
TV_FeedForeward_Debug
VDC_TV_MzTarget_Nm
CalcTargetMz

CONCEPT



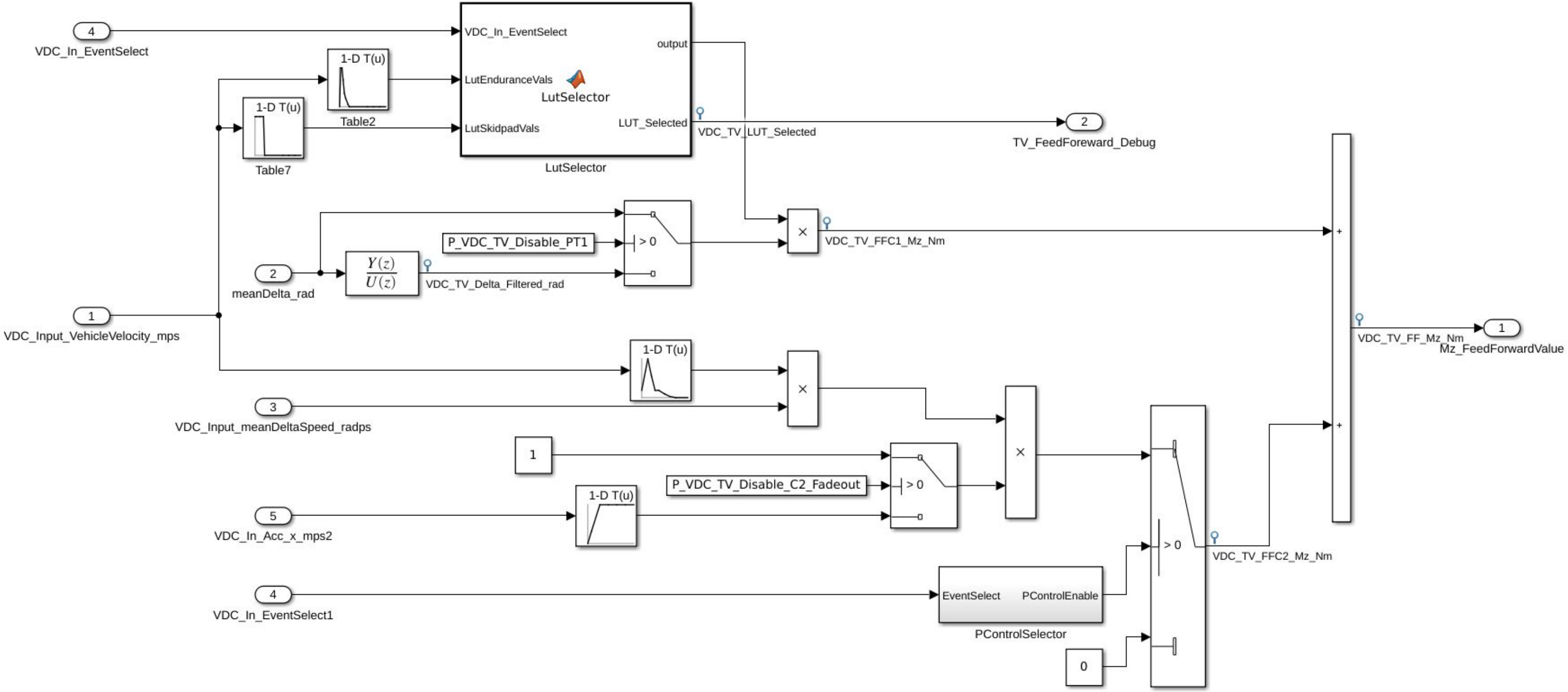
P_VDC_TV_Controller_override_Nm
P_VDC_TV_Controller_override_enable
VDC_Input_Delta_FL_rad
VDC_Input_Delta_FR_rad
calcDeltaMean
meanDelta_rad
deadZone
VDC_Input_dPsi_radps
VDC_Input_TV_Coefficient
VDC_Input_VehicleVelocity_mps
Mz_PControlValue
P Control
VDC_In_EventSelect1
EventSelect
PControlEnable
PControlSelector
VDC_TV_PControl_Enable
VDC_TV_MzTarget_Nm
Mz_target_Nm
VDC_Input_VehicleVelocity_mps1
VDC_Input_meanDeltaSpeed_radps
VDC_In_EventSelect
VDC_In_Acc_x_mps2
Mz_FeedForwardValue
TV_FeedForeward_Debug
Feed forward control
VDC_Input_TVFac
VDC_TV_FFFactor

CONCEPT



2
meanDelta_rad
4
VDC_Input_VehicleVelocity_mps
C_VDC_Wheelbase_m
Wheelbase
dPsi_target
1
VDC_Input_dPsi_radps
3
VDC_Input_TV_Coefficient
VDC_TV_dPsi_target_radps
u
deadZone
y
VDC_TV_dPsi_diff_radps
VDC_TV_PControl_Mz_Nm
1
Mz_PControlValue

CONCEPT



4
VDC_In_EventSelect
1-D T(u)
Table2
1-D T(u)
Table7
VDC_In_EventSelect
output
LutEnduranceVals
LutSelector
LutSkidpadVals
LUT_Selected
LutSelector
VDC_TV_LUT_Selected
2
TV_FeedForeward_Debug
P_VDC_TV_Disable_PT1
> 0
VDC_TV_FFC1_Mz_Nm
2
meanDelta_rad
Y(z)/U(z)
VDC_TV_Delta_Filtered_rad
1
VDC_Input_VehicleVelocity_mps
VDC_TV_FF_Mz_Nm
1
Mz_FeedForwardValue
1-D T(u)
3
VDC_Input_meanDeltaSpeed_radps
1
P_VDC_TV_Disable_C2_Fadeout
> 0
1-D T(u)
5
VDC_In_Acc_x_mps2
> 0
VDC_TV_FFC2_Mz_Nm
4
VDC_In_EventSelect1
EventSelect
PControlEnable
PControlSelector
0

CONCEPT

- Calculate target yaw rate based on a more physical model
- Use a linear vehicle model / linearize a nonlinear model for certain operating points
- Use controller to calculate correctional yaw moment

  → Hopefully better performance when used together with QP

CONCEPT

- Implement models & simulate
- Evaluate them at PreST
- Use old target M_Z generation if no better performance is achieved

VIELEN DANK FÜR EURE AUFMERKSAMKEIT!