

adjustment kpi and parameters

Failed Ehrler, Georg (TV-F13) created pipeline for commit 12185abc finished 5 days ago

For 18-scenario-based-unit-tests-implementation

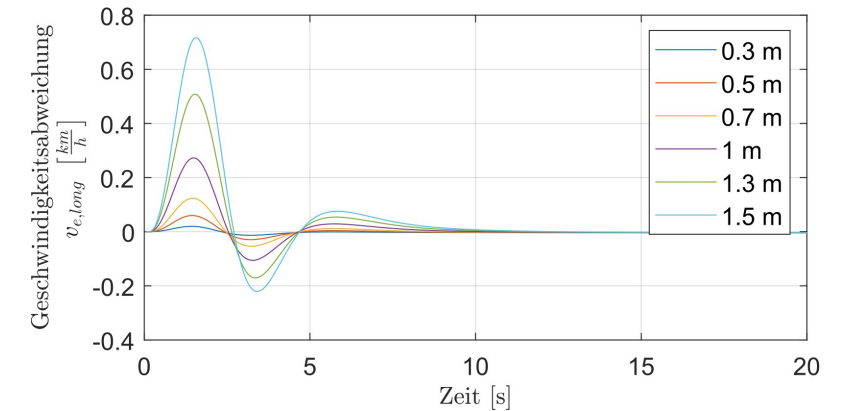
latest 13 Jobs 37 minutes 51 seconds, queued for 4 seconds

Pipeline Needs Jobs 13 Failed Jobs 2 Tests 259

test

- Failed Alice 8
- Failed IAVShuttle 5

- Success Alice: [test_ACC_Straight_VelConst]
- Success Alice: [test_Curve]
- Success Alice: [test_SCurve]
- Success Alice: [test_Straight_Brake_Soft]
- Failed Alice: [test_Straight_Offset]
- Success Alice: [test_Straight_Speed_Change]



Entwicklung und Implementierung einer automatisierten szenariobasierten Unit-Test Strategie für einen modellprädiktiven Pfadfolgeregler in einer GitLab CI Pipeline

1) Systemüberblick

1. Modellprädiktiver Pfadfolgeregler – MPFC
2. Konzeptvorstellung
3. Szenariobasiertes Testen
4. Softwareentwicklungsprozess

2) Implementierung

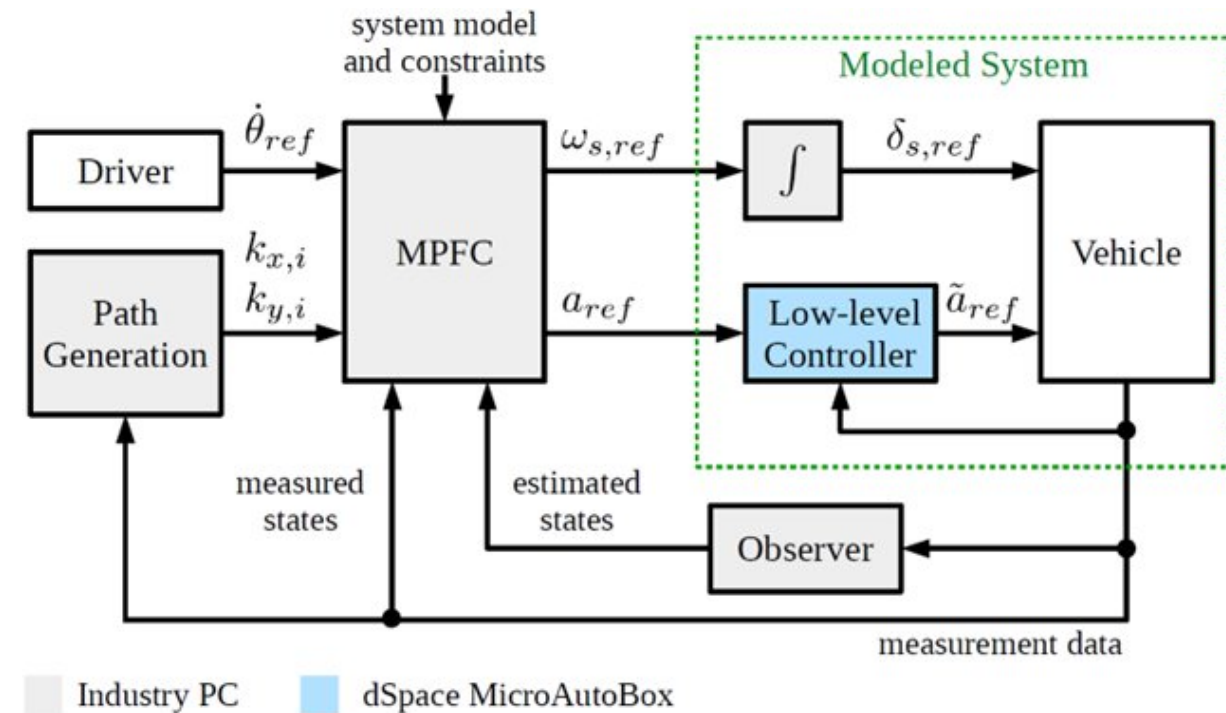
1. Ablauf eines Testskriptes
2. Parametrierung
3. Key Performance Indicators – KPIs
4. Szenariendefinition
5. GitLab CI Pipeline

3) Zusammenfassung & Ausblick

4) Referenzen

modellprädiktiver Pfadfolgeregler - MPFC

- Teil eines Systems zum autonomen Fahren
- **Eingänge:** Sollgeschwindigkeit, Pfaddaten, Fahrzeugzustände, Beschränkungen und ein Fahrzeugmodell
- **Ausgänge:** Sollbeschleunigung, Solllenkradwinkelgeschwindigkeit



[1]

modellprädiktiver Pfadfolgeregeler - MPFC

$$J(x(t_k), \theta(t_k), \bar{u}(\cdot), \bar{\vartheta}(\cdot)) = \int_{t_k}^{t_k+T_p} F(\bar{e}(\tau), \bar{x}(\tau), \bar{u}(\tau), \bar{\vartheta}(\tau)) d\tau + E(\bar{e}(t_k + T_p), \bar{x}(t_k + T_p))$$

e: Pfadabweichung

θ : Pfadparameter

ϑ : Geschwindigkeit

$$J(x(t_k), \theta(t_k), \bar{u}^*(\cdot), \bar{\vartheta}^*(\cdot)) = \min_{\bar{u}(\cdot), \bar{\vartheta}(\cdot)} J(x(t_k), \theta(t_k), \bar{u}(\cdot), \bar{\vartheta}(\cdot))$$

$$s.t \quad \dot{\bar{x}}(\tau) = f(\bar{x}(\tau), \bar{u}(\tau)), \quad \bar{x}(t_k) = x(t_k)$$

$$\dot{\bar{\theta}}(\tau) = \bar{\vartheta}(\tau), \quad \bar{\theta}(t_k) = \theta(t_k)$$

$$\bar{e}(\tau) = h(\bar{x}(\tau)) - p(\bar{\theta}(\tau))$$

$$\bar{u}(\tau) \in \mathcal{U}, \quad \bar{x}(\tau) \in \mathcal{X}$$

$$\bar{\theta}(\tau) \in [0, \theta_{max}], \quad \bar{\vartheta}(\tau) \in \mathcal{V}$$

$$h_c(\bar{x}(\tau), \bar{u}(\tau)) \leq 0$$

Dynamik des Fahrzeugs

Geschwindigkeitsvorgabe

Pfadabweichung

Beschränkungen von Zuständen
und Eingängen

Beschränkung durch Lenkaktorik

modellprädiktiver Pfadfolgeregeler - MPFC

$$J(x(t_k), \theta(t_k), \bar{u}(\cdot), \bar{\vartheta}(\cdot)) = \int_{t_k}^{t_k+T_p} F(\bar{e}(\tau), \bar{x}(\tau), \bar{u}(\tau), \bar{\vartheta}(\tau)) d\tau + E(\bar{e}(t_k + T_p), \bar{x}(t_k + T_p))$$

$$F(e, x, u, \vartheta) = \left\| \begin{pmatrix} e \\ a_{lat}(x) \end{pmatrix} \right\|_Q^2 + \left\| \begin{pmatrix} u \\ \vartheta - \vartheta_{ref} \end{pmatrix} \right\|_R^2$$

$$E(e, x) = \left\| \begin{pmatrix} e \\ a_{lat}(x) \end{pmatrix} \right\|_P^2$$

$$Q = \text{diag}(q_x, q_y, q_\psi, q_a),$$

$$P = \text{diag}(p_x, p_y, p_\psi, p_a),$$

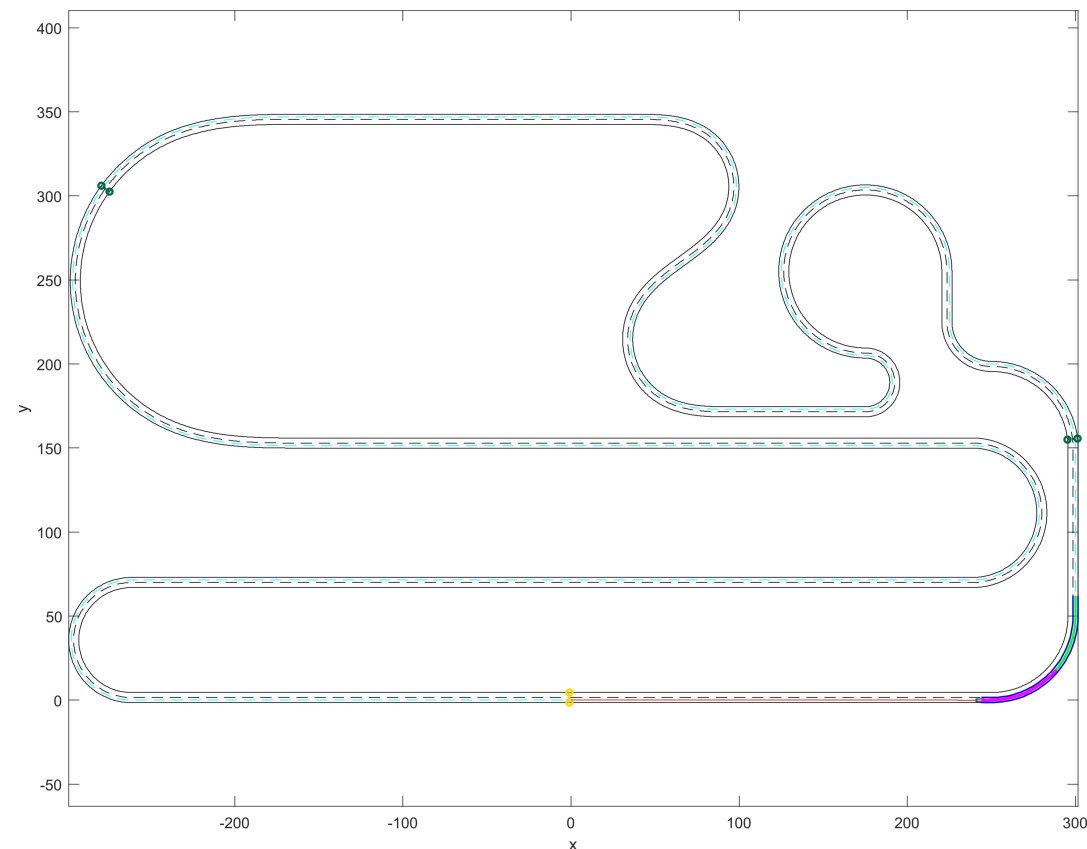
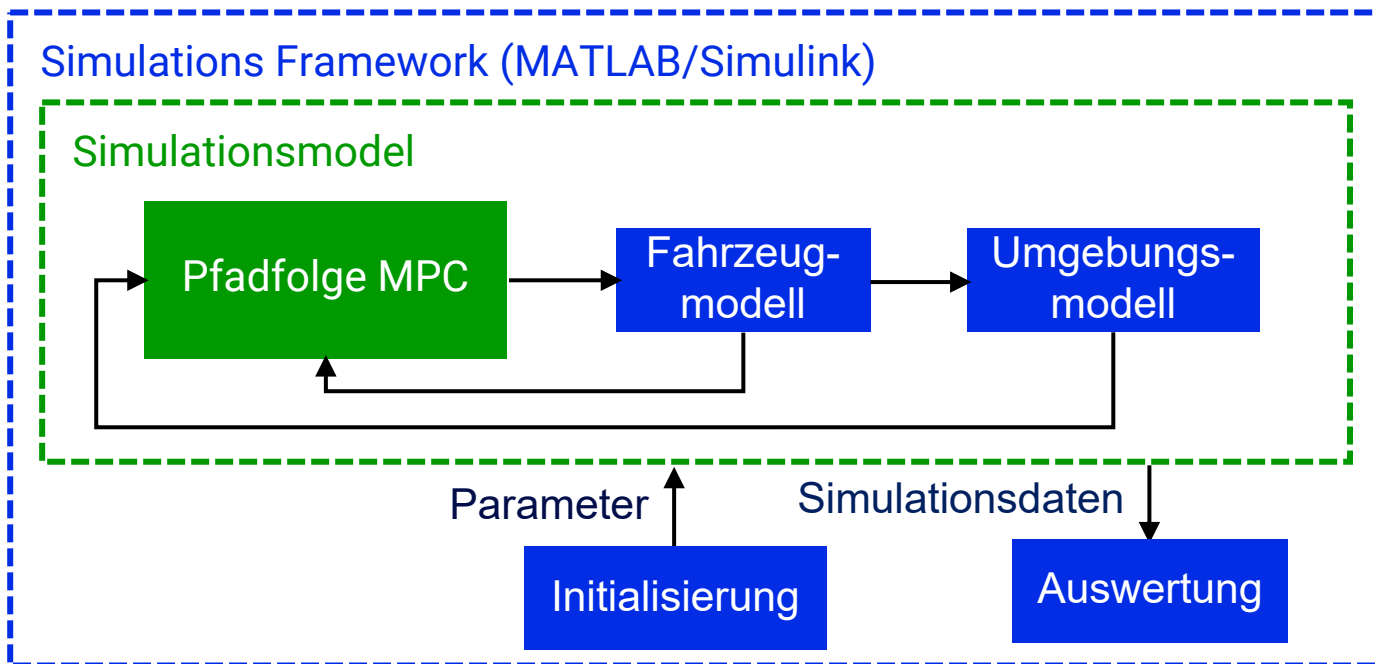
$$R = \text{diag}(r_a, r_\omega, r_v),$$

e : Pfadabweichung

a_{lat} : laterale Beschleunigung

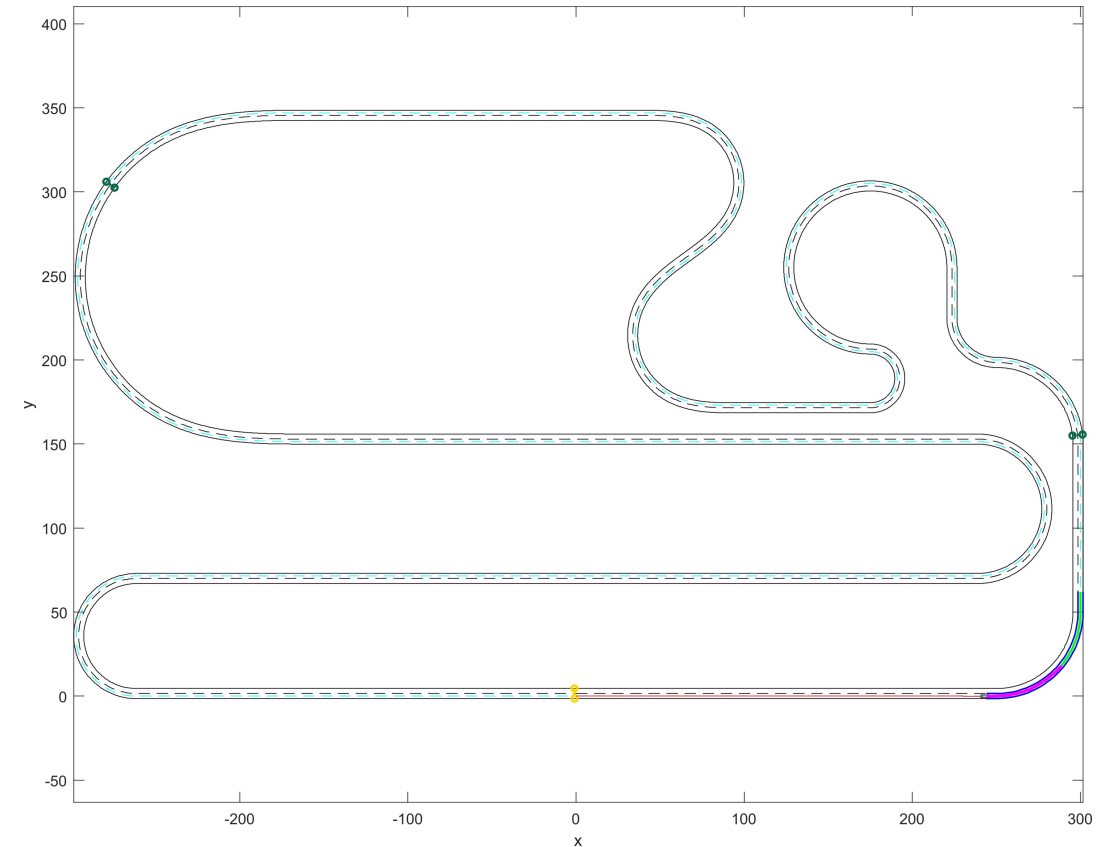
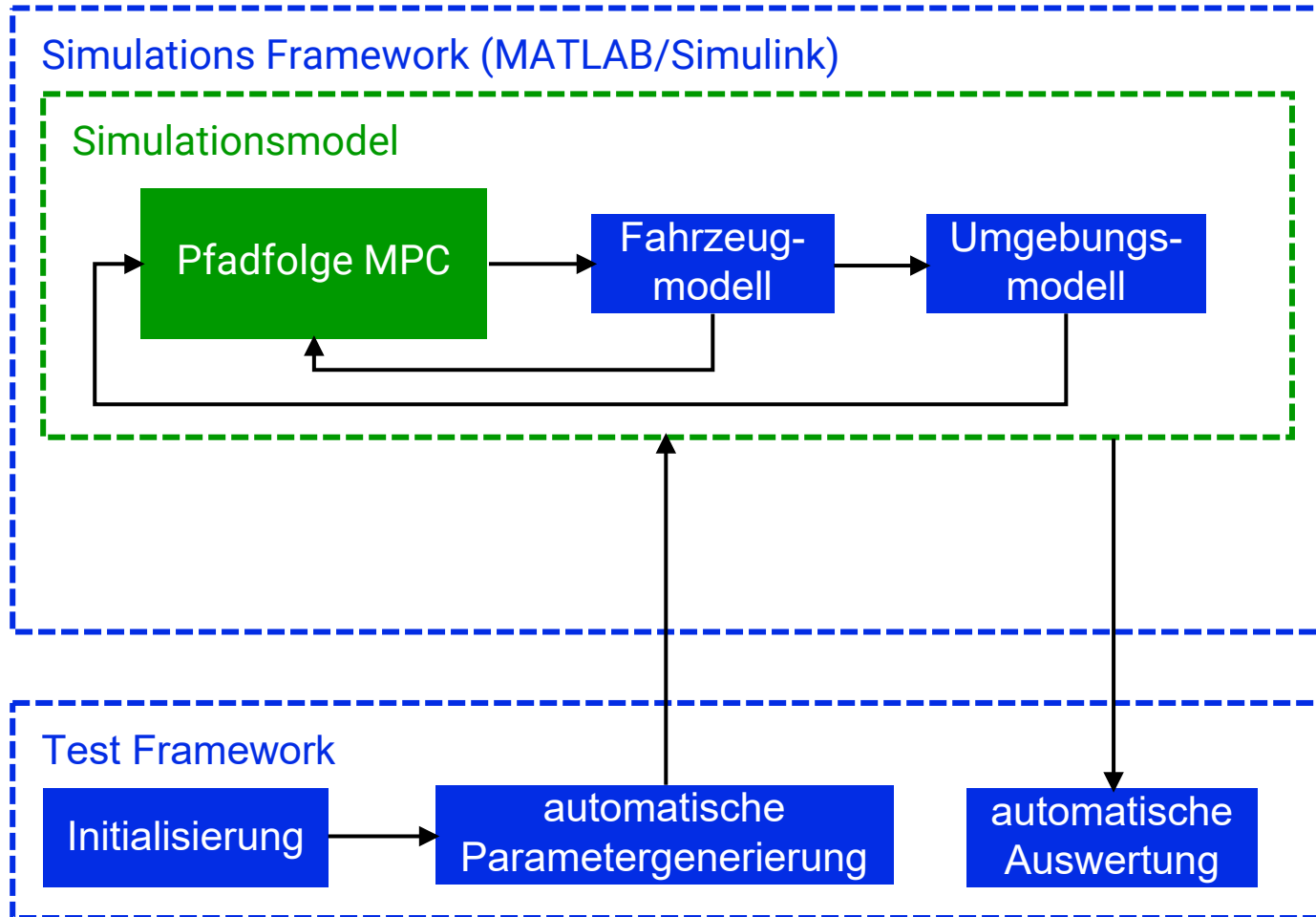
ϑ : Geschwindigkeit

Konzeptvorstellung



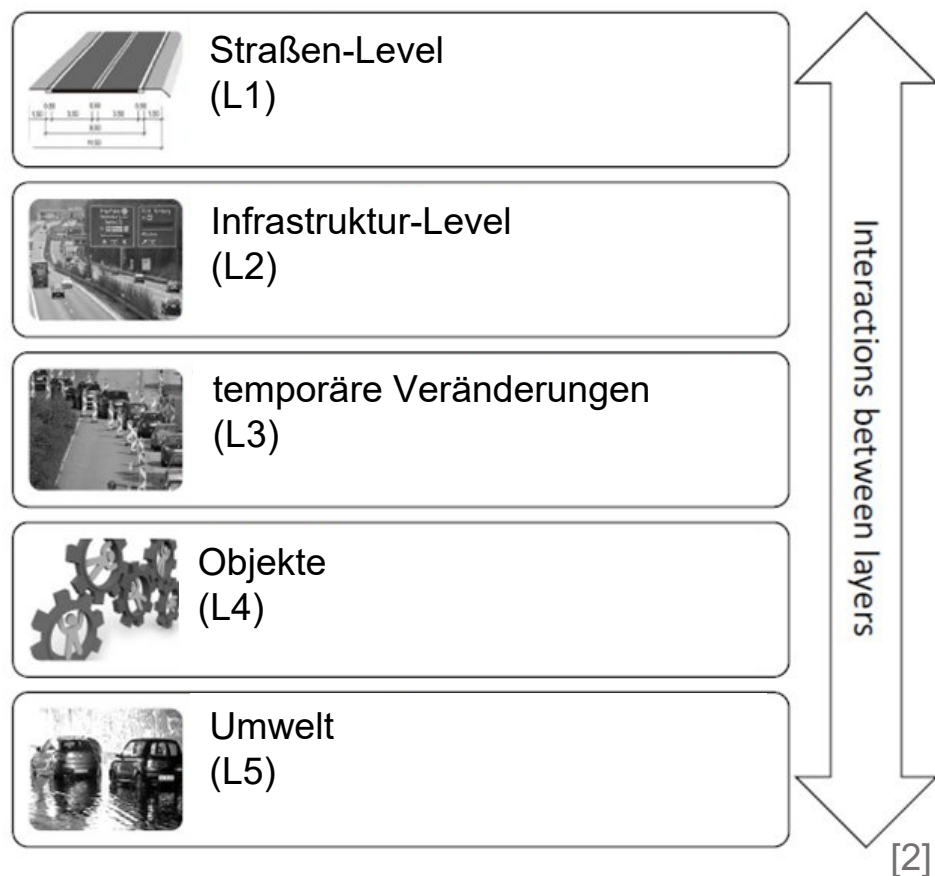
- Automatisierte Initialisierung, Ausführung und Auswertung
- Entwickler über Probleme informieren

Konzeptvorstellung

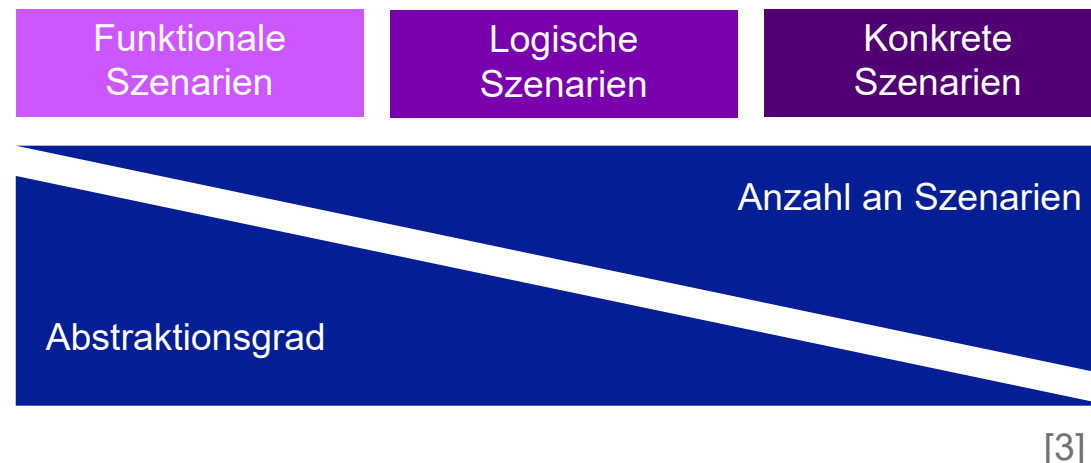


- Automatisierte Initialisierung, Ausführung und Auswertung
- Entwickler über Probleme informieren

Szenariobasiertes Testen



jedes Level erhöht Komplexität/Parameterzahl



- Funktional: verbale Beschreibung
- Logisch: Parameter und Parametergrenzen festgelegt
- Konkret: genauer Wert für jeden Parameter

Softwareentwicklungsprozess

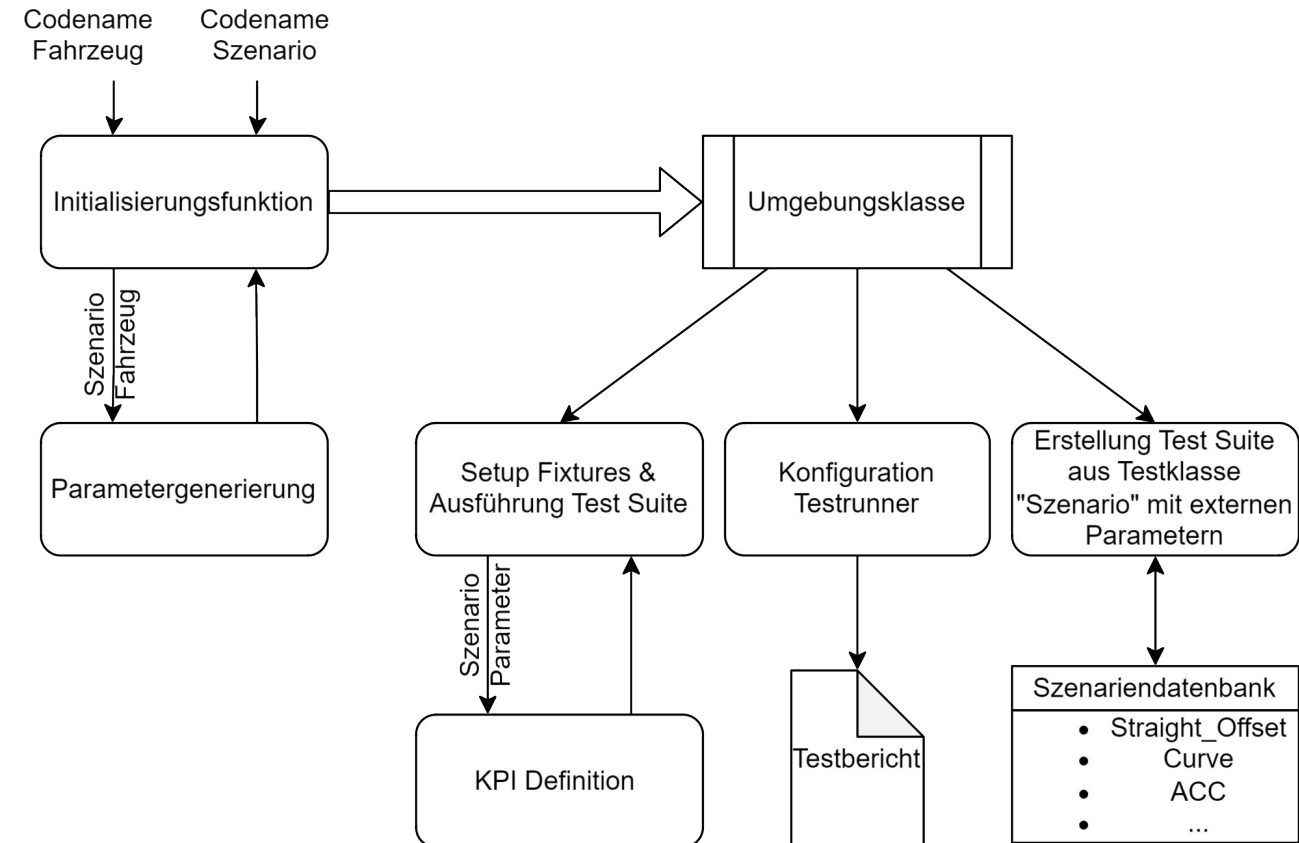


[4]

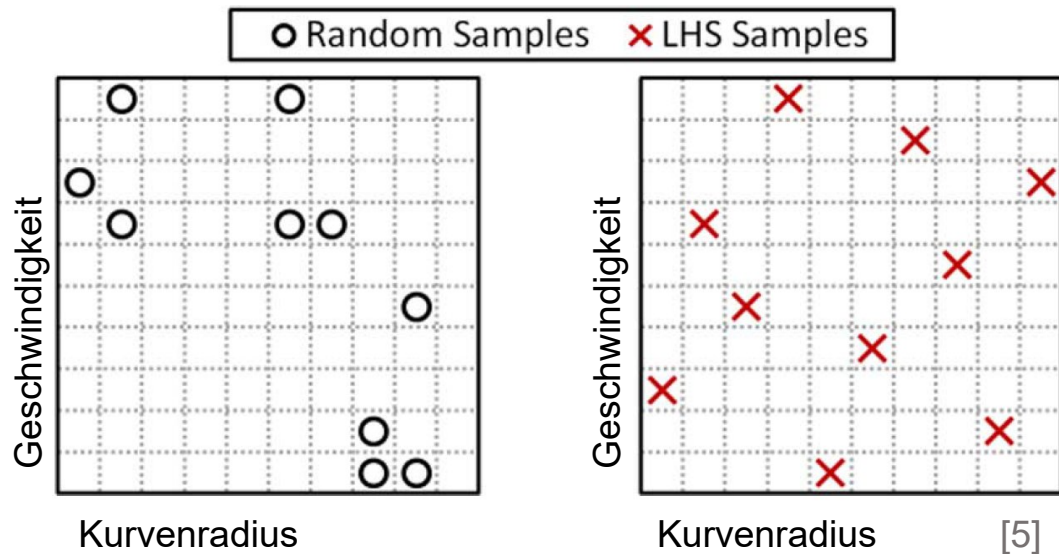
- Unit-Test: kleinster Baustein im Testprozess, deckt üblicherweise eine einzelne Funktion/Klasse ab
- Pipeline: fasst alle notwendigen Testschritte zusammen
- Continuous Integration: fortlaufendes Zusammenfügen von Teilkomponenten einer zu einer vollständigen Software

Ablauf eines Testskriptes

- Laden von Parametersätzen, abhängig von Szenario und Fahrzeug
- Umgebungsclass stellt definierte Anfangszustände durch Laden von Fahrzeugparametern und Initialisierung der Simulation her
- Jedes Szenario existiert als eigene Testklasse
- Testrunner führt Test Suite aus, sorgt für korrekten Ablauf von Setup- und Teardown-Funktionen und erstellt Testbericht
- KPIs werden dynamisch vor Ausführung einer Simulation geladen



Parametrierung



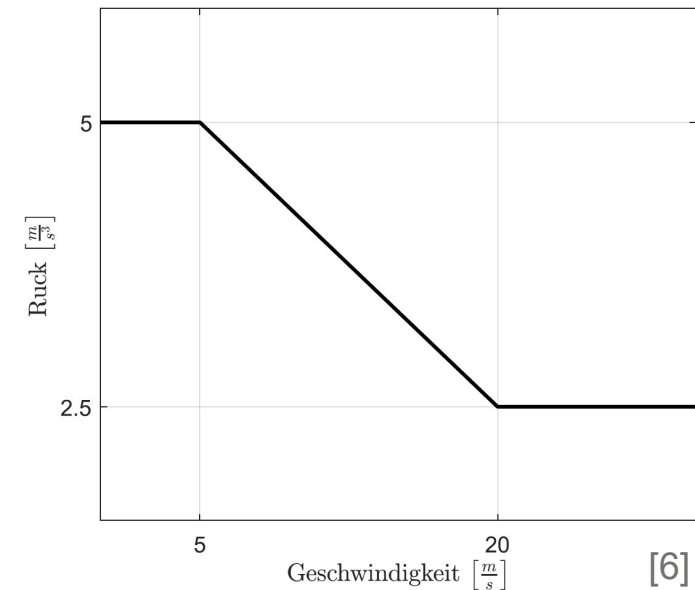
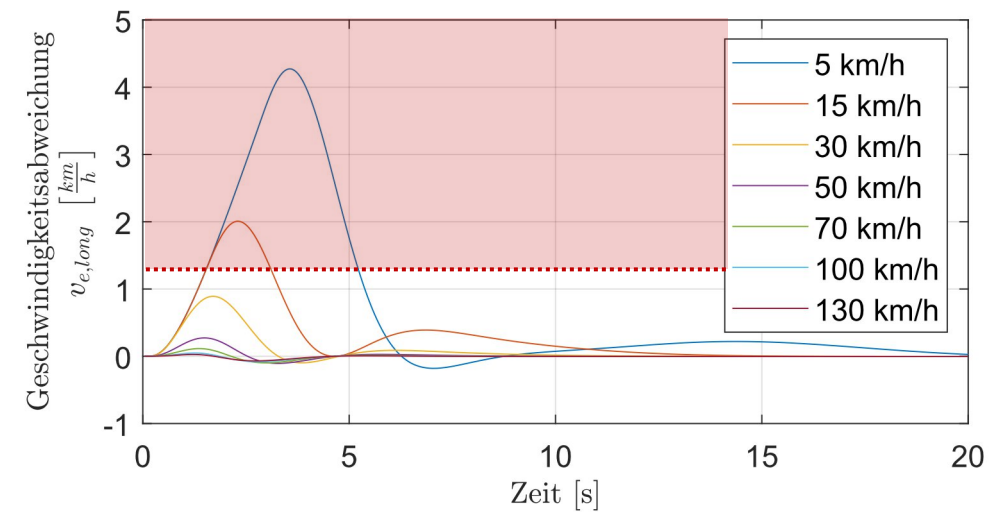
- Anzahl an Simulationen n steigt exponentiell mit Anzahl der Parameter N , bei k Werten für jeden Parameter

$$n = k^N$$

- Reduzierung durch Sampling, festlegen von Anzahl der Simulationen
- Latin Hypercube Sampling bietet bessere Abdeckung

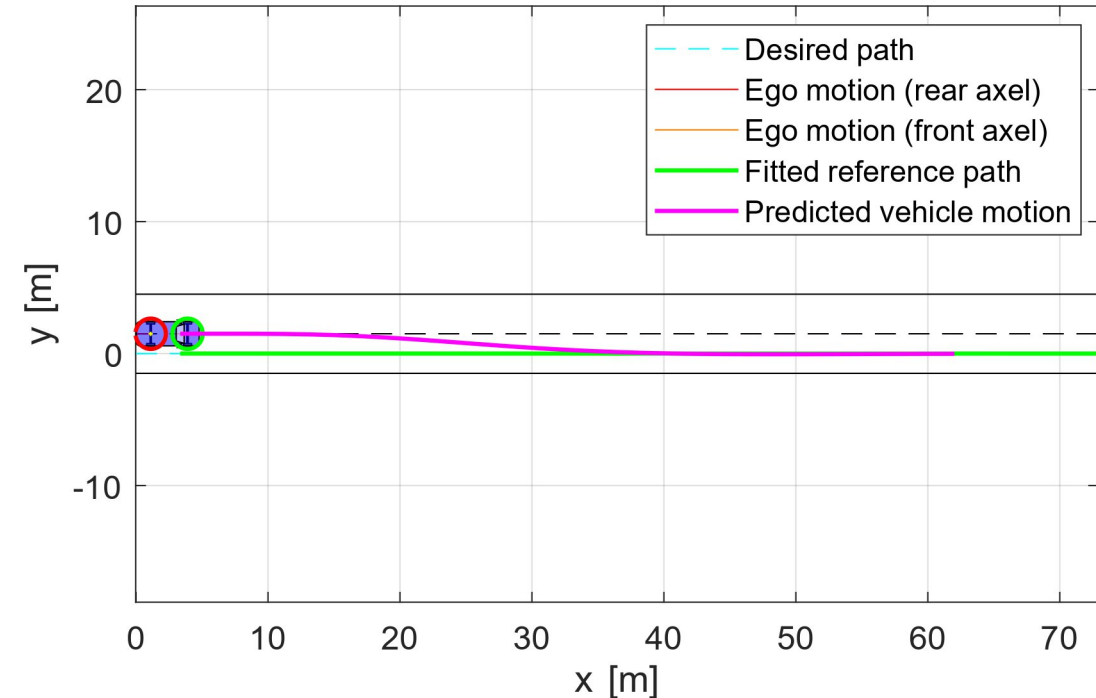
Key Performance Indicators - KPIs

- Grenzwerte für Messdaten, ab denen ein Test als fehlgeschlagen bewertet wird
- Stark szenario- und parameterabhängig
- Longitudinale/laterale Beschleunigung, Ruck, Pfadabweichung, Geschwindigkeitsabweichung, Distanzfehler (ACC)
- Teilweise existieren Normen/gesetzliche Vorgaben



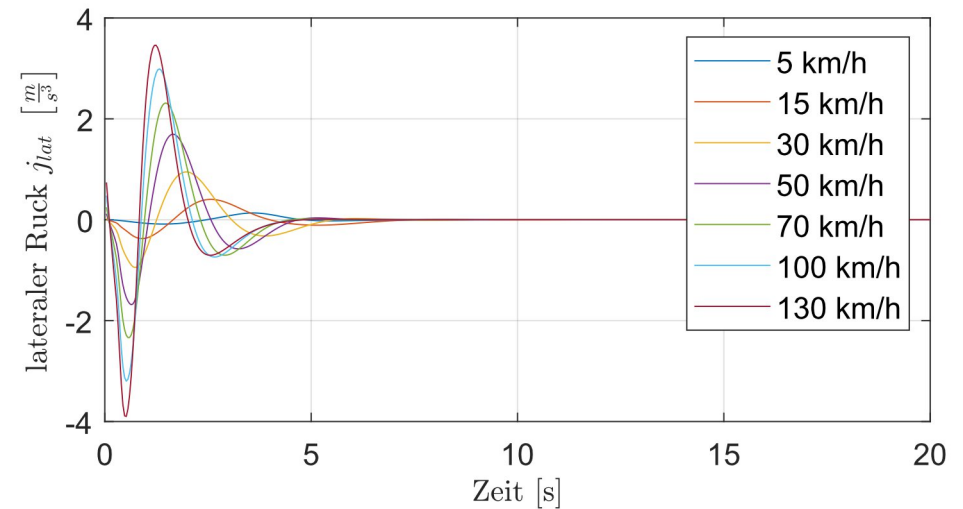
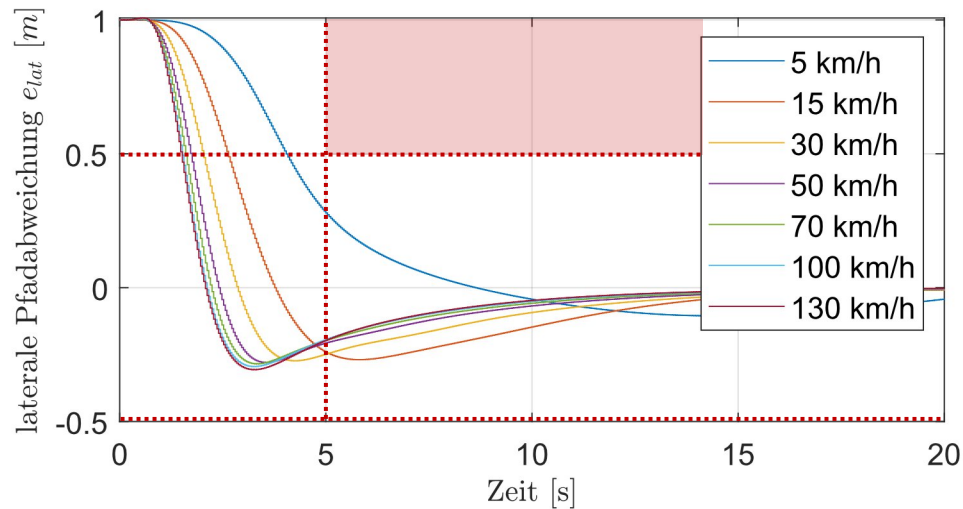
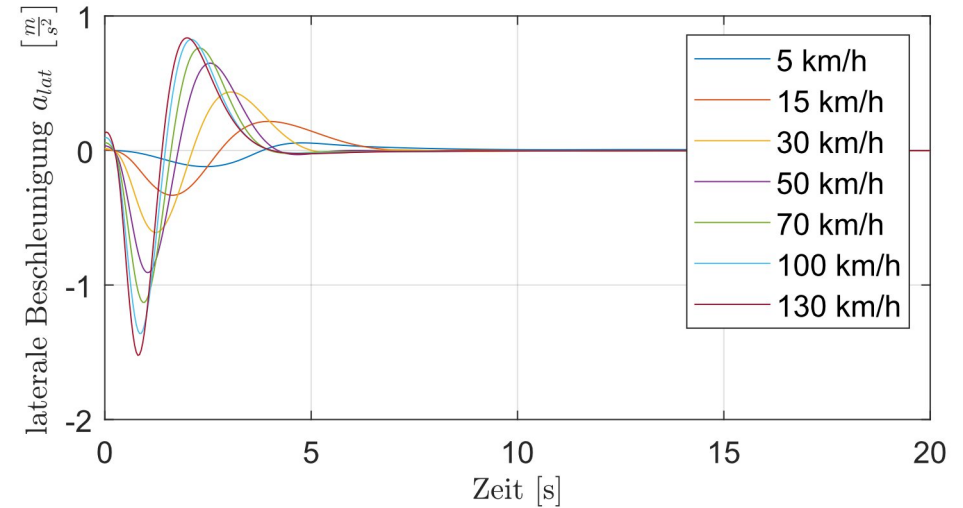
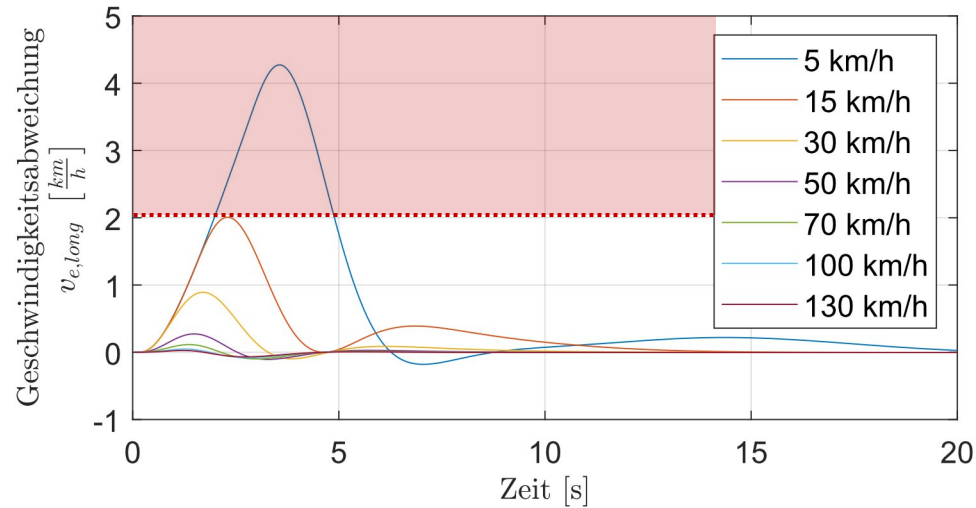
Szenariendefinition

- Berücksichtigung verschiedener Quellen:
 - Szenariendatenbank
 - Normen
- Beschreibung, Anforderungsdefinition, KPI, Parametervariation
- Beispiel: Gerade mit initialem Versatz



Szenen-ID	Beschreibung	erwartetes Verhalten	KPI	Parametrierung
Straight_Offset	<ul style="list-style-type: none"> • das Ego startet mit der Pfadgeschwindigkeit auf einer Geraden • es besteht ein lateraler Versatz zum gewünschten Pfad 	<ul style="list-style-type: none"> • der Versatz sollte komfortabel abgebaut werden und das Ego dem Pfad mit der Pfadgeschwindigkeit folgen 	<ul style="list-style-type: none"> • a_Lat • a_Long • j_Lat • j_Long • t_settle • v_diff 	<ul style="list-style-type: none"> • v_path [5, vMax] • s_offset [-1.5, 1.5] • parameterSet • v_EgoInit = v_path

Szenariendefinition: Gerade mit initialem Versatz



Gitlab CI Pipeline

- Konfiguration eines Jobs pro Fahrzeug in der Stage „test“
- Jeder Job erzeugt untergeordnete Jobs, die das jeweilige Szenario definieren
- Fahrzeug-Szenario-Kombination wird an Matlab-Skript übergeben
- Testbericht wird als Artefakt bereitgestellt
- Bei entsprechender Runner-Konfiguration auch parallele Ausführung möglich

reworked curve a_Lat kpi


 **Failed** Ehrler, Georg (TV-F13) created pipeline for commit `9c1b1628`  finished 5 days ago


For `18-scenario-based-unit-tests-implementation`


 **latest**  13 Jobs  36 minutes 15 seconds, queued for 4 seconds


Pipeline Needs Jobs **13** Failed Jobs **3** Tests **265**


test


 Alice 8


 IAVShuttle 5


 Alice: [test_ACC_Straight_AccelDecel]

 Alice: [test_ACC_Straight_Cruise]

 Alice: [test_ACC_Straight_VelConst]

 Alice: [test_Curve]

 Alice: [test_SCurve]

 Alice: [test_Straight_Brake_Soft]

Gitlab CI Pipeline: Testreport

MATLAB® Test Report

Timestamp: 08-Mar-2024 12:35:34

Host: 20IAV015842P-0

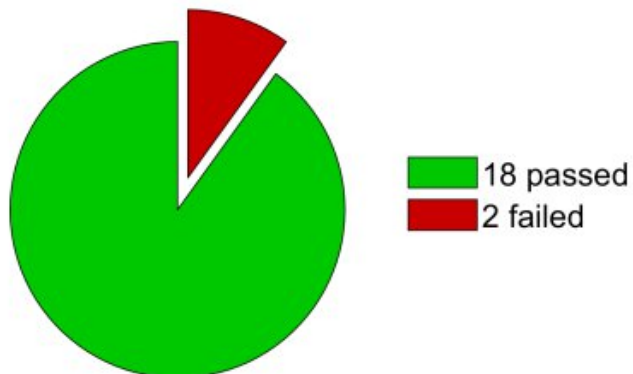
Platform: win64

MATLAB Version: 23.2.0.2428915 (R2023b) Update 4

Number of Tests: 20

Testing Time: 348.3869 seconds

Overall Result: FAILED




s_DistTargetError
 Class Setup Parameters: testParams=struct_3

The test failed.

Duration: 45.1431 seconds

Event:

Verification failed.

Test Diagnostic:

Vehicle takes too long to establish the minimum distance.

Framework Diagnostic:

verifyLessThan failed.

--> The value must be less than the maximum value.

Actual Value:

9.680000000000000

Maximum Value (Exclusive):

8

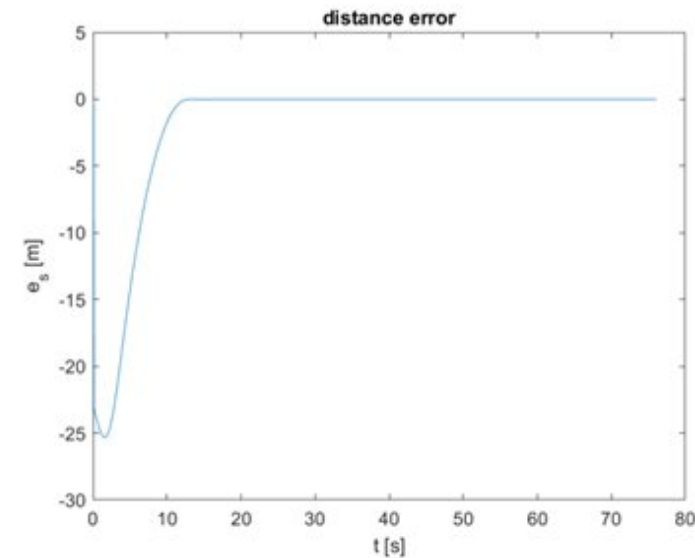
Additional Diagnostic 1:

{ "scenario": "test_ACC_Straight_VelConst", "vehicle": "Alice", "v_egoInit": 97.55, "t_gap": -0.85, "v_diff": 7.804, "parameter Set": "SET_2", "v_Target": 89.746, "v_path": 130.06666666666666 }

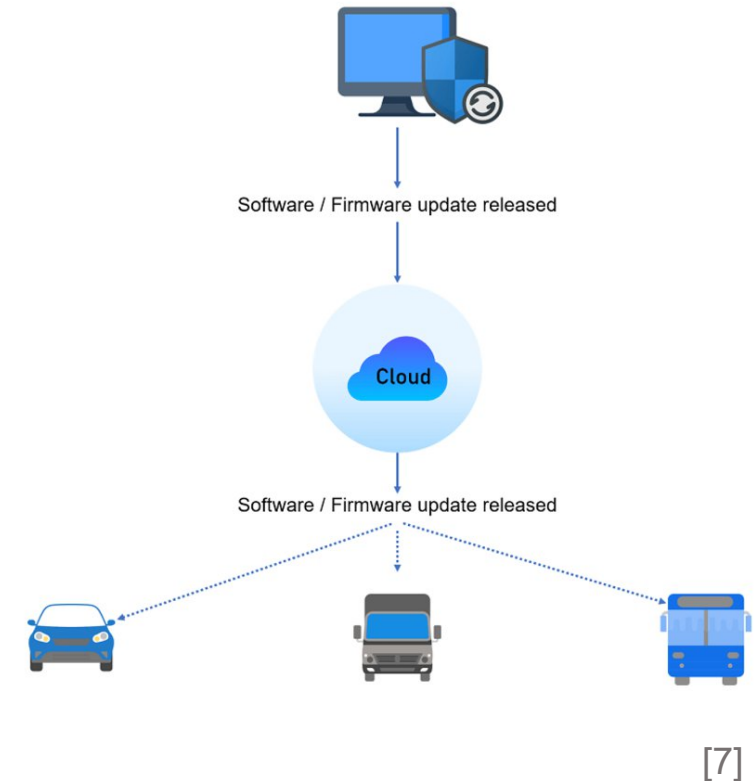
Additional Diagnostic 2:

Figure saved to:

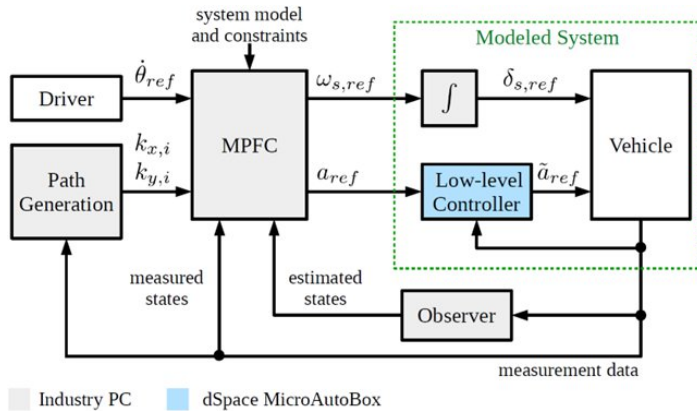
--> C:\ADPL\Runner\builds\sG-gTE6b\1\AD-platform\projects\ctrl_had\project_ctrl_pathfollowingmpc\Artifacts\857ec331-a ba2-423b-883a-1de5b6e614d4\s_DistTargetError8c57904c-1bd4-46ee-95b5-5453031b8ce3.png



- Konzept für das automatisierte Testen der Pfadfolge MPC wurde implementiert
 - Neue Szenarien können hinzugefügt werden
 - Falls benötigt Erweiterung um neue Fahrzeuge möglich
 - KPIs und Parametergrenzen sind veränderbar
- Automatisierte Ausführung in der GitLab-Pipeline und Benachrichtigung des Entwicklers
- Einbinden in vollständiges over-the-air Updatesystem
- Kritikalitätsbewertung der einzelnen Testdurchläufe, finden kritischer Parameterwerte für Test



- [1] Robert Ritschel, Frank Schrödel, Juliane Hädrich und Jens Jäkel. „Nonlinear Model Predictive Path-Following Control for Highly Automated Driving“. In: IFAC PapersOnLine 52.8 (2019). 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019, S. 350–355. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2019.08.112.
- [2] Gerrit Bagschik, Till Menzel und Markus Maurer. „Ontology based Scene Creation for the Development of Automated Vehicles“. In: 2018 IEEE Intelligent Vehicles Symposium (IV). 2018, S. 1813–1820. DOI: 10.1109/IVS.2018.8500632.
- [3] Till Menzel, Gerrit Bagschik und Markus Maurer. „Scenarios for Development, Test and Validation of Automated Vehicles“. In: 2018 IEEE Intelligent Vehicles Symposium (IV). 2018, S. 1821–1827. DOI: 10.1109/IVS.2018.8500406.
- [4] Red Hat. Was ist CI/CD? Konzepte und CI/CD Tools im Überblick. 2024. URL: <https://www.redhat.com/de/topics/devops/what-is-ci-cd> (besucht am 18. 03. 2024).
- [5] Robin Preece und Jovica Milanović. „Efficient Estimation of the Probability of Small-Disturbance Instability of Large Uncertain Power Systems“. In: IEEE Transactions on Power Systems 31 (Apr. 2015), S. 1–10. DOI: 10.1109/TPWRS.2015. 2417204.
- [6] International Organization for Standardization. ISO 15622:2018. Intelligent transport systems – Adaptive cruise control systems – Performance requirements and test procedures. Sep. 2018.
- [7] <https://quanticor-security.de/ota-updates/>



adjustment kpi and parameters

✖ Failed Ehrler, Georg (TV-F13) created pipeline for commit 12185abc finished 5 days ago

For 18-scenario-based-unit-tests-implementation

latest 13 Jobs 37 minutes 51 seconds, queued for 4 seconds

Pipeline Needs Jobs 13 Failed Jobs 2 Tests 259

test

- ✖ Alice 8
 - ✓ Alice: [test_ACC_Straight_VelConst]
 - ✓ Alice: [test_Curve]
 - ✓ Alice: [test_SCurve]
 - ✓ Alice: [test_Straight_Brake_Soft]
 - ✖ Alice: [test_Straight_Offset]
 - ✓ Alice: [test_Straight_Speed_Change]
- ✖ IAVShuttle 5
 -
 -
 -
 -
 -
 -

