HOCHSCHULE ESSLINGEN

Benutzerhandbuch

Projekt 1

Kreuzungswarner

FSB4

Florian Alles: 777715

Severin Amann: 777251

Gabriel Panic: 771601

Fabian Wenz: 777500

Betreuender Professor:

Prof. Dr. Rer. Nat. Gunther Schaaf

Projektzeitraum:
Sommersemester 2025

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Dieses Benutzerhandbuch beschreibt die **allgemeine Simulation eines Fahrerassistenzsystems**, das zur Visualisierung und Analyse verschiedener **Kreuzungsszenarien** entwickelt wurde. Die Simulation bietet eine realistische Darstellung eines **EGO-Fahrzeugs**, eines querenden Fahrzeugs auf der Kreuzung sowie eines optionalen vorausfahrenden Fahrzeugs in einer **2D-Umgebung**.

Über eine intuitive **grafische Benutzeroberfläche (GUI)** können die Szenarien ausgewählt und konfiguriert werden. Die GUI ermöglicht es, zentrale Parameter wie **Fahrzeuggeschwindigkeit, Sensorreichweite, Sensortypen und Wetterbedingungen** direkt anzupassen oder vordefinierte **Szenarien** auszuwählen. Diese Einstellungen werden unmittelbar an das Simulationsmodell übergeben, sodass eine flexible und präzise Steuerung der Simulation gewährleistet ist.

Ziel der Simulation ist es, verschiedene Fahrsituationen realitätsnah darzustellen und deren Auswirkungen auf das Fahrerassistenzsystem zu analysieren. Alle Simulationen laufen vollständig innerhalb der Simulationsumgebung, ohne Verbindung zu externer Hardware oder Echtzeit-Einfluss.

1.1 Versionskontrolle mit GitHub

In Bezug auf das **Kreuzungswarner-Projekt** dient **GitHub** als zentrale Plattform zur Versionsverwaltung und Zusammenarbeit. Es ermöglicht eine strukturierte Entwicklung und Verwaltung des gesamten Projektes.

Vorteile von GitHub für dieses Projekt sind:

- **Versionskontrolle:** Alle Änderungen am Quellcode werden auf GitHub als Commits gespeichert. Dadurch kann jederzeit eine frühere Version des Projekts wiederhergestellt werden und Änderungen sind nachvollziehbar.
- **Zusammenarbeit:** Jeder der Entwickler hat die Möglichkeit in einen unabhängigen Branch zu arbeiten. Änderungen können über Pull/Push in den Hauptcode integriert werden.
- **Automatische Sicherung:** Alle Projektdateien sind sicher in der GitHub-Cloud gespeichert. Dadurch ist das Projekt vor Datenverlust geschützt und kann jederzeit von einem anderen Rechner aus fortgesetzt werden.
- **Dateien:** Das Benutzerhandbuch, Videoaufnahmen und vieles mehr können direkt im GitHub-Repository gepflegt werden. So sind alle Informationen zentral und aktuell für jeden verfügbar.

Es empfiehlt sich für weiterführende Projekte ebenfalls mit GitHub zu arbeiten!

1.2 Hauptordner

In dem Hauptordner befinden sich alle Funktionen und Dateien, welche für den Kreuzungswarner relevant sind.

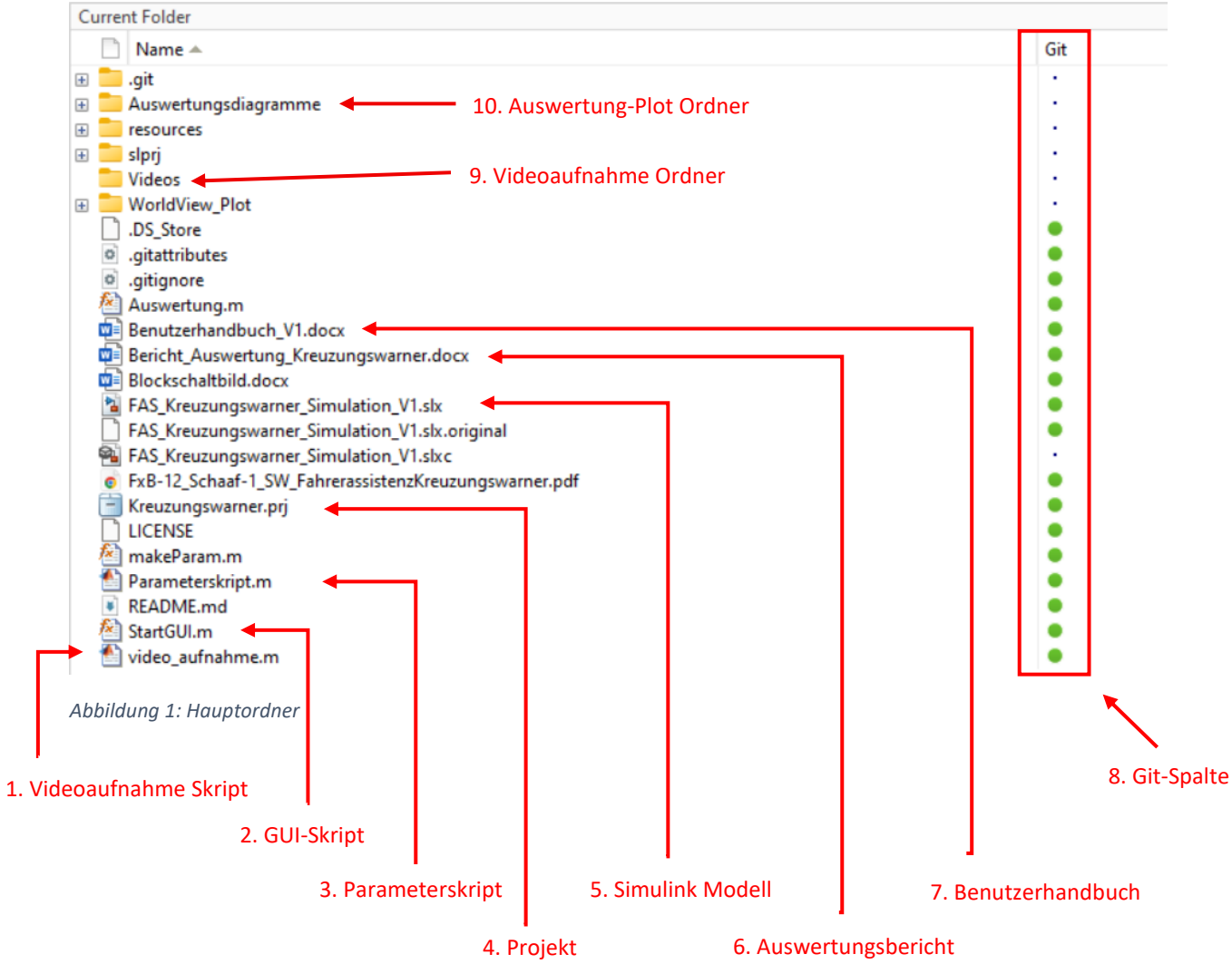


Abbildung 1: Hauptordner

1. **Videoaufnahme Skript:**
   Dient dem Aufzeichnen der Figures.
2. **GUI-Skript:**
   Automatische Aufforderung bei Simulationsstart zur Parameterübergabe.
3. **Parameterskript:**
   Zentrale Schnittstelle der Wertübergabe an Funktionen, Simulation und setzen der Werte durch die GUI bei Simulationsbeginn.
   Beinhaltet Werte wie Fahrzeugmaße, Startposition und vieles mehr.
4. **Projekt:**
   Muss geöffnet werden, um die Simulation zu starten und automatische Verknüpfung von GUI, Parameterskript und Simulation zu gewährleisten.
5. **Simulink Modell:**
   Simulationsaufbau des Kreuzungswarners.
6. **Auswertungsbericht:**
   Bericht, welcher die Auswertung verschiedener Szenarien beinhaltet.

7. **Benutzerhandbuch:**
   Eine Anleitung zum Verständnis des Projekts, welche eine grundlegende Beschreibung des Projekts, Simulation, Auswertung und der Videoaufnahmen erläutert.
8. **GitHub:**
   Dient der Übersicht, welche Dateien von Git verwaltet und welche modifiziert sind.
9. **Videoaufnahme Ordner:**
   Hier werden die aufgezeichneten Videos mit dem jeweiligen individuellen Pfad abgelegt.
10. **Auswertungs-Plot Ordner:**
    Nachdem man eine Auswertungssimulation ausgeführt hat und der Plot erstellt ist, speichert man den Plot in diesem Ordner.

2 Simulationsablauf

In den folgenden Abschnitten wird der Simulationsablauf des Fahrerassistenzsystems detailliert beschrieben. Schritt für Schritt werden die wichtigsten Einstellungen und Funktionen erläutert. Zur Veranschaulichung dienen Abbildungen, die den Ablauf und die Benutzeroberfläche der Simulation visualisieren.

2.1 Simulink Modell öffnen

Um das Modell zu öffnen, muss zunächst das Projekt ,,Kreuzungswarner.prj“ (siehe Abbildung 1: Hauptordner) geöffnet werden.

- Es öffnet sich ein neuer Bereich:

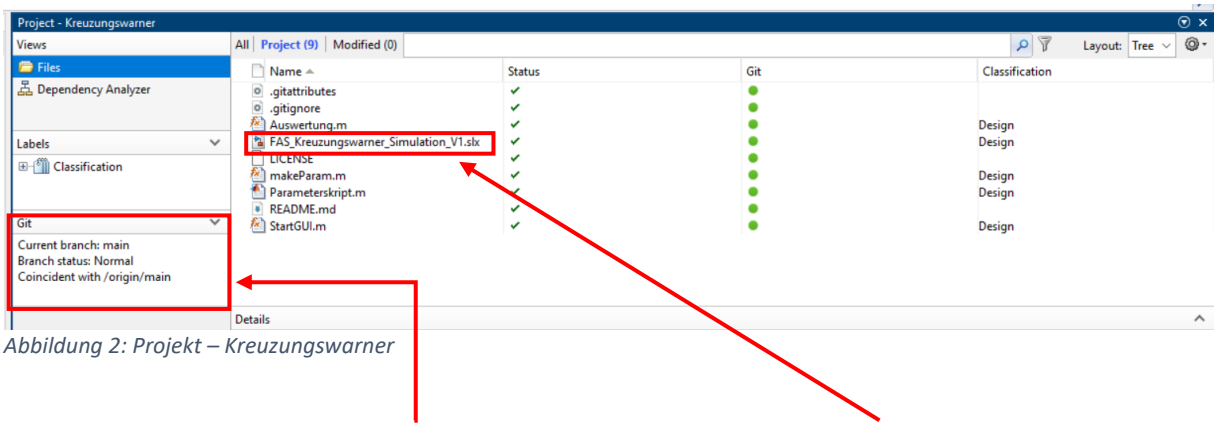


Abbildung 2: Projekt – Kreuzungswarner

In dieser Abbildung sehen wir den Branch in dem aktuell gearbeitet wird, das Simulink-Modell sowie erneut den Status der GitHub – Verwaltung.

- Öffnen des Modells ,,FAS_Kreuzungswarner_Simulation_V1“ per Doppelklick oder rechter Maustaste und Open.

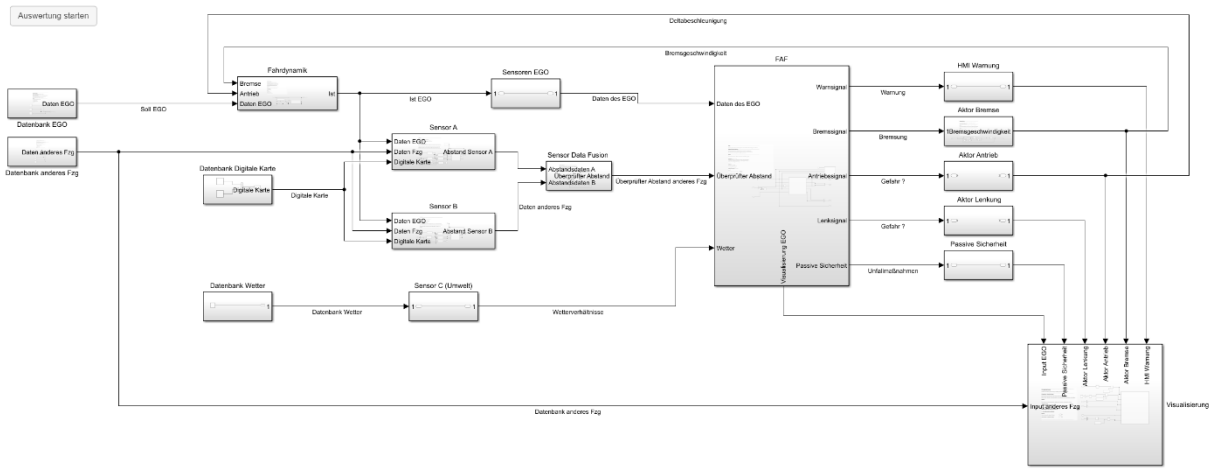


Abbildung 3 Simulink-Modell

2.2 Simulation durchführen

Sobald das Modell geöffnet ist, kann die Simulation über die Hauptleiste unter:

- „Simulation > RUN“ gestartet werden.

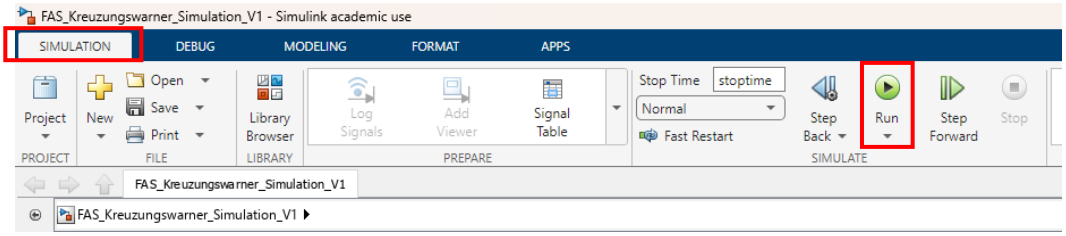


Abbildung 4: Simulink Menüleiste

- Es öffnet sich die GUI zur Parameterübergabe.

2.2.1 GUI zur Parameterübergabe:

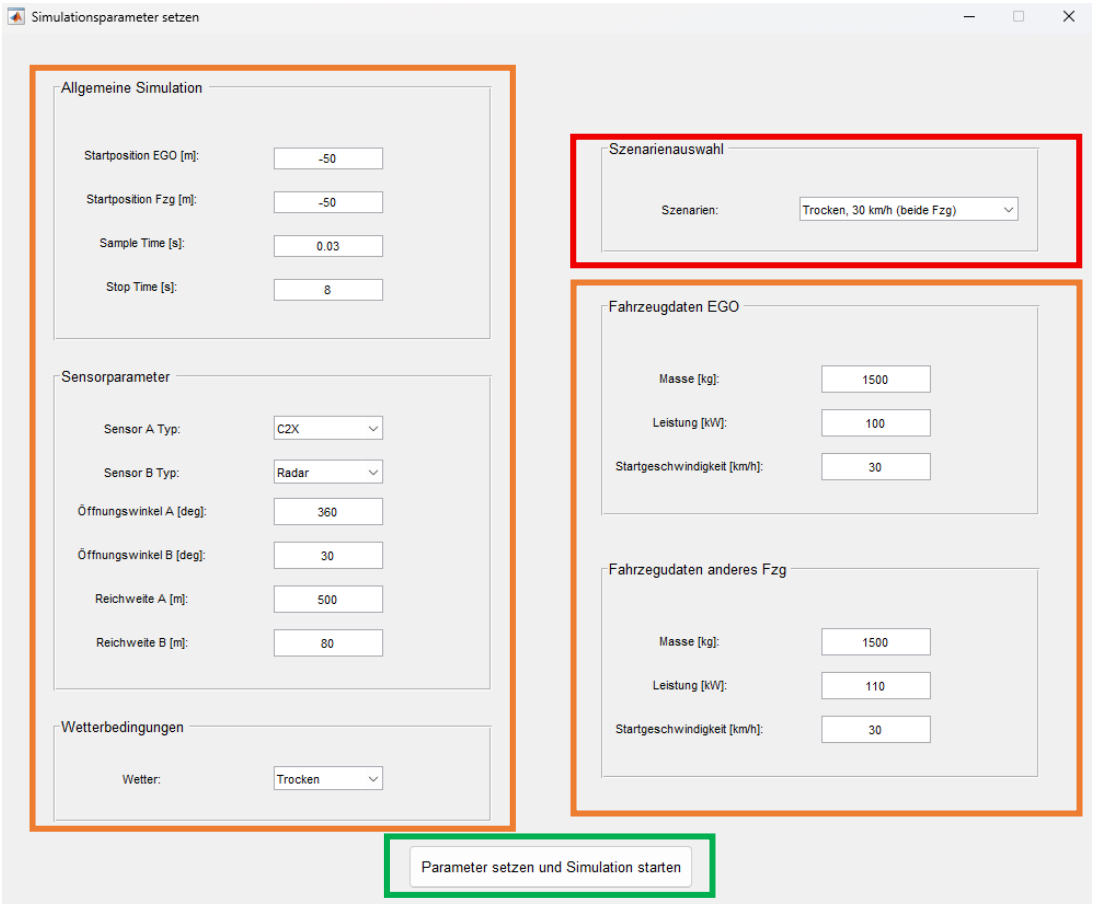
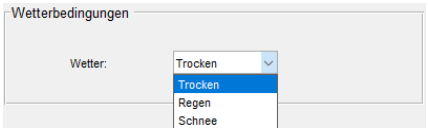


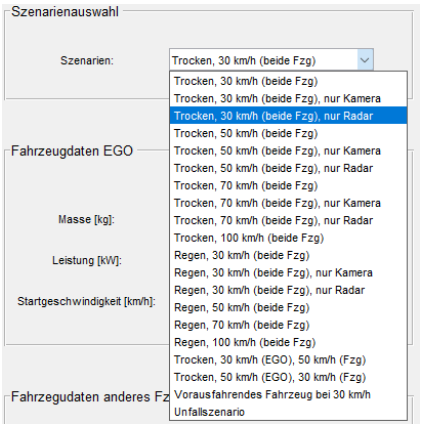
Abbildung 5: GUI-Parameterübergabe

In der GUI werden die Simulationsparameter festgelegt.



- **Individuelle Parametereingabe:**

Diese können individuell angepasst werden, wie zum Beispiel die Auswahl verschiedener Sensortypen oder Wetterbedingungen über Dropdown-Menüs.



- **Vordefinierte Szenarien:**

Alternativ können vordefinierte Szenarien ausgewählt werden, die eine automatische Voreinstellung der relevanten Parameter ermöglichen.

- **Parameterbestätigung:**

Zum Ende wird die Eingabe per ,,Parameter setzen und Simulation starten“ bestätigt und die Simulation beginnt.

3 Videoaufnahme

In den folgenden Abschnitten wird der Ablauf zur Videoaufnahme der Simulationsfiguren detailliert beschrieben. Zur Veranschaulichung dienen Abbildungen, die den Aufnahmeprozess und die Benutzeroberfläche der Videoerstellung visualisieren.

Das Ziel der Videoaufnahme ist eine schnelle und flüssige Darstellung der Szenarien, ohne eine vorherige Simulation durchführen zu müssen.

Die aufgenommenen Videos werden automatisch in einem festgelegten Ordner gespeichert, dessen Speicherort vor Beginn der Aufnahme individuell angepasst werden muss. Für die automatische Namensgebung der Videos wird der Szenarien-Name aus dem Parameterskript ermittelt und das Datum per Funktion über das Skript vergeben.

3.1 Zielordner Anpassung

- Öffnen der ,,video_aufnahme.m“ Datei (siehe 1.2 Hauptordner)

```
%% Hier den Pfad eintragen ============================================================================
videoDir = 'C:\,,Benutzerabhängig"\Projekt 1\Videos';  % Pfad wo Video gespeichert wird
if ~exist(videoDir, 'dir')
    mkdir(videoDir);                                   % Erstellt einen Ordner wenn noch nicht vorhanden
end
%% ====================================================================================================
```

Abbildung 6: Zielpfad eintragen

Der Zielpfad für die Videoaufnahme ist individuell anpassbar. Es empfiehlt sich jedoch, den Abschnitt:
,,\Projekt 1\Videos“
beizubehalten, um eine einheitliche und leicht nachvollziehbare Ablagestruktur für alle Videoaufnahmen sicherzustellen. Dies vereinfacht den Zugriff und die Verwaltung der aufgezeichneten Videos.

Falls ein **nicht** vorhandener Zielpfad Ordner eingetragen wird, erstellt Matlab diesen automatisch.

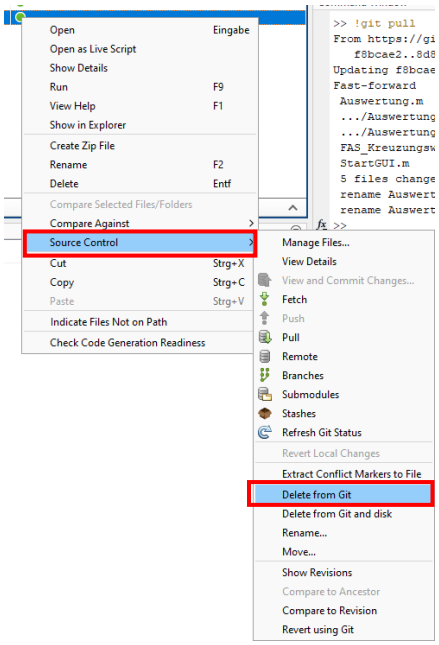
3.1.1 Implementieren von Neuerungen im Videoaufnahme-Skript

Wenn Neuerungen implementiert werden und der individuelle Pfad für die Videoaufnahme eingestellt ist, empfiehlt es sich, die Datei ,,video_aufnahme.m“ aus der Git-Verwaltung zu entfernen.

Andernfalls würde der Pfad bei jedem Push überschrieben werden.

Dies verhindert Konflikte und sorgt dafür, dass jeder Entwickler seine eigenen Einstellungen für den Speicherort der Videos beibehalten kann.

- Rechter Mausklick auf die Datei



Danach muss commitet und gepusht werden!

3.2 Simulationsmodell Namen übergeben

Falls sich für kommende Projekte die Version des Simulink Modells ändert (V1 -> V2), muss der Modellname an 2 Codestellen aktualisiert werden.

1. Automatischer Simulationsstart

```
% Simulationsstart
set_param('FAS_Kreuzungswarner_Simulation_V1', 'SimulationCommand', 'start');  % Modellname einfügen
disp('Simulation und Szenarienauswahl gestartet...');
```

Abbildung 7: 1. Simulink Modellname übergeben

2. Überprüfung bis Simulationsende

```
% Videoaufnahme
while strcmp(get_param('FAS_Kreuzungswarner_Simulation_V1', 'SimulationStatus'), 'running')  % Modellname einfügen
```

Abbildung 8: 2: Simulink Modellname übergeben

3.3 Videoaufnahme Starten und Ablauf (Im Skript zusätzliche Erklärung)

Das Projekt und Simulink Modell muss zuvor geöffnet werden.

- siehe **2.1 Simulink Modell öffnen**

Die Videoaufnahme wird über das ,,video_aufnahme.m“ Skript gestartet (siehe **1.2 Hauptordner**).

- Run in Menüleiste

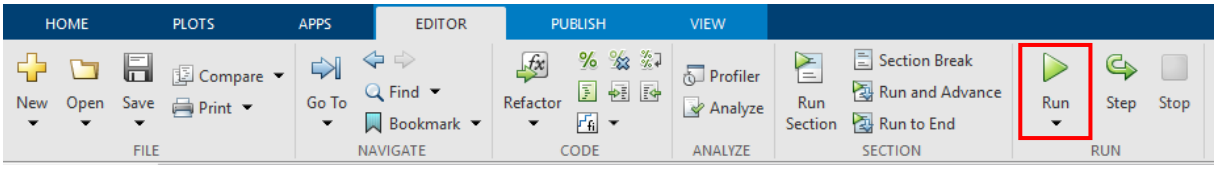


Abbildung 9: RUN-Videoaufnahme

Als nächstes beobachten wir das ,,command window“:

- >> video_aufnahme

- Simulation wird gestartet und Szenarienauswahl Abfrage...

- Simulation gestartet...
- Szenarien Auswahl per GUI erfolgreich gestartet...
- Figures erkannt, starte Videoaufnahme...

- Abfrage Szenario für Video-Name...
- Video erfolgreich gespeichert unter: ,,Zielpfad/Szenarienname_Datum“

Videoaufnahme ist jetzt erfolgreich beendet!

4 Auswertung

Im folgenden Abschnitt wird der Start der Auswertung beschrieben, welche dem Analysieren von verschiedenen Szenarien und Sensoren mit variierenden Öffnungswinkeln dient.

Die Simulation startet bei einem Öffnungswinkel von 10 Grad und endet bei maximal 90 Grad.
Je nach Schrittweitenwahl wird die Simulationsanzahl beeinflusst.
Beispiel:
Wahl der Winkelschrittweite: 10
Anzahl der Simulationsdurchläufe: 90 Grad (Maximalwinkel) / 10 Grad (Winkelschrittweite) = 9

Hinweis: Wenn die Auswertung einmal gestartet ist, kann sie nicht abgebrochen werden, bis alle Simulationsschritte durchgelaufen sind. Daher ist standardmäßig in der durch die Auswertung gestartete GUI die Winkelschrittweite 80 Grad gewählt.

Zur Auswertung gibt es einen Bericht, welcher im Hauptordner zu finden ist (siehe 1.2 Hauptordner).
Dieser Bericht dient der Präsentation an Projektunbeteiligte, denen grundlegend die Notwendigkeit von solchen Simulationen erklärt wird.
Zudem wird auf verschiedene Szenarien und Sensoren, sowie deren Einfluss auf Fahrsituationen und daraus folgende Auswirkungen eingegangen.

4.1 Auswertung starten

Das Projekt und Simulink Modell muss zuvor geöffnet werden.

- siehe 2.1 Simulink Modell öffnen

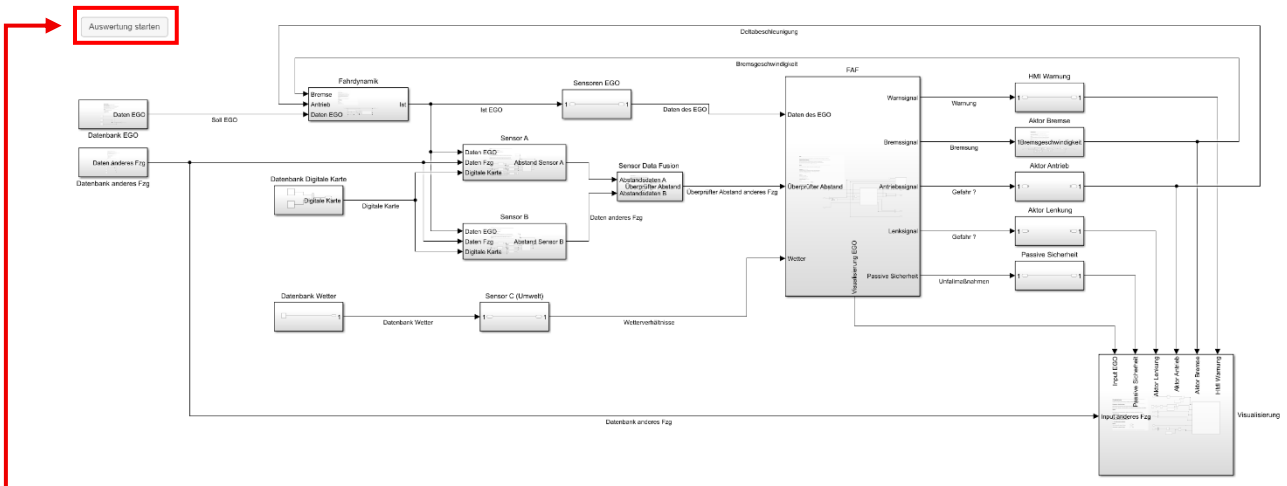
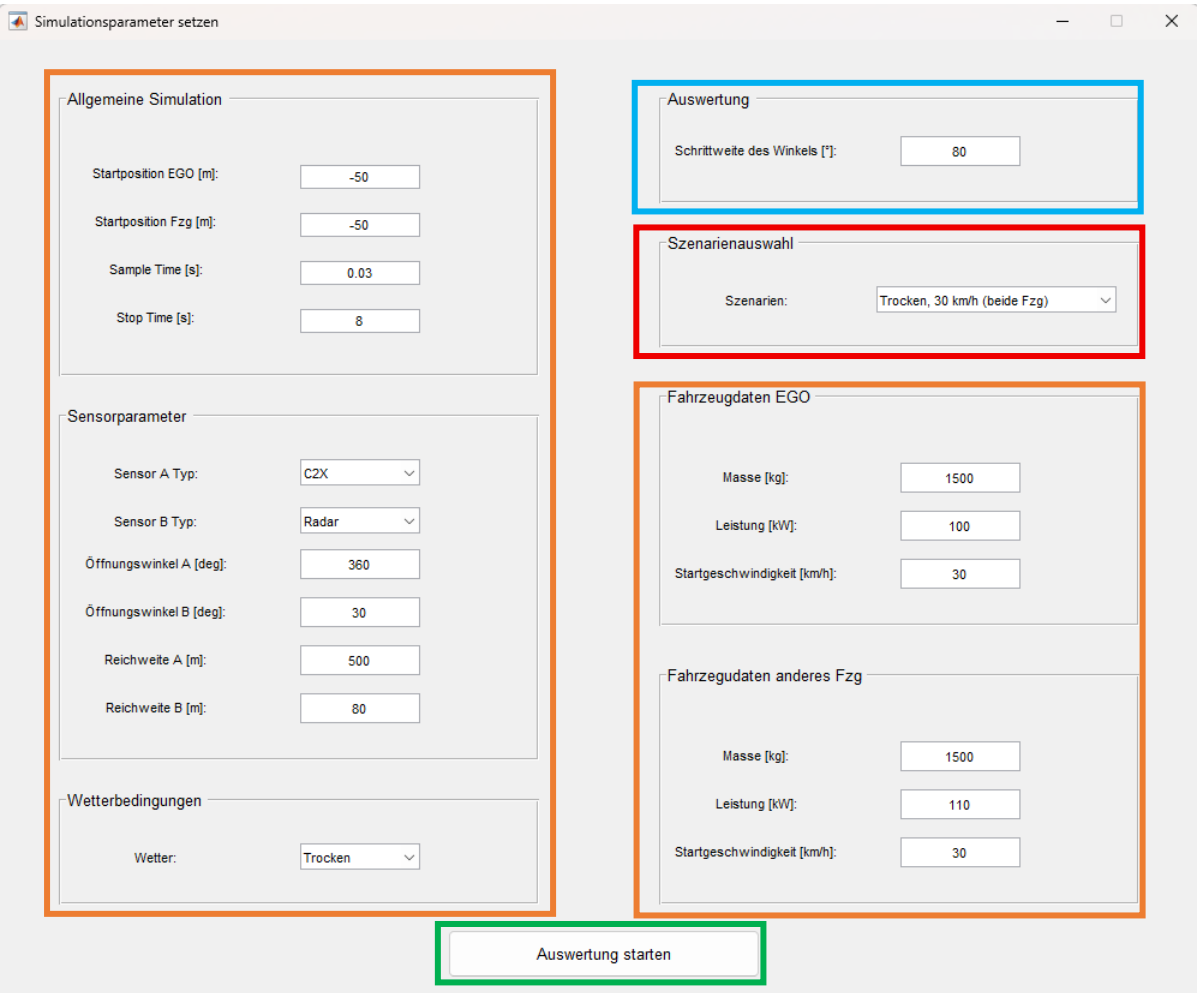


Abbildung 10: Simulink-Modell

- Starten der Auswertung mit dem Call-Back Button links oben in der Simulink-Modell-Umgebung
- Auswertung starten
- Es öffnet sich die GUI in der wir die Parameter/Szenarien und die Schrittweite des Winkels übergeben

4.2 Winkelschrittweite und Parameterübergabe/Szenarienauswahl



➢ **Individuelle Parametereingabe:**

Diese können individuell angepasst werden, wie zum Beispiel die Auswahl verschiedener Sensortypen oder Wetterbedingungen über Dropdown-Menüs.

➢ **Vordefinierte Szenarien:**

Alternativ können vordefinierte Szenarien ausgewählt werden, die eine automatische Voreinstellung der relevanten Parameter ermöglichen.

➢ **Schrittweite des Winkels**

Hier wird die Schrittweite angegeben, in der die Simulation von 10 bis 90 Grad durchläuft. Beispiel der Simulationsanzahlberechnung (siehe 4. Auswertung)

➢ **Parameterbestätigung:**

Zum Ende wird die Eingabe per ,,Auswertung starten“ bestätigt. Die Simulation beginnt und erstellt nach dem gesamten Ablauf einen Plot.

4.3 Speichern des Auswertung-Plots

Nachdem der Plot sich öffnet, kann dieser in dem Ordner ,,Auswertungsdiagramme“ (siehe **1.2 Hauptordner**) gespeichert werden.