

# Contents

<b>1</b>	<b>Die Porsche AG</b>	<b>1</b>
<b>2</b>	<b>Das Praktikum</b>	<b>3</b>
2.1	Fachpraktikum . . . . .	3
2.2	Fachbereich Porsche . . . . .	3
<b>3</b>	<b>Praktikumsbericht</b>	<b>5</b>
3.1	FP7 - Entwicklung (8 Wochen) . . . . .	5
3.1.1	Die Lattice-Boltzmann-Methode (LBM) . . . . .	6
3.1.2	Einarbeitung . . . . .	6
3.1.3	Datenauswertung in Excel . . . . .	9
3.1.3.1	Makroversion 1.0 . . . . .	9
3.1.3.2	Makroversion 1.5 . . . . .	11
3.1.3.3	Makroversion 2.0 . . . . .	11
3.1.3.4	Das Excel-Workbook . . . . .	12
3.1.3.5	Endresultat . . . . .	13
3.1.4	Aktueller Workflow der Messungsauswertung . . . . .	15
3.1.4.1	Messstellenplan . . . . .	15
3.1.4.2	Grenztemperaturen . . . . .	15
3.1.4.3	Messungen . . . . .	16
3.1.4.4	Simulationen . . . . .	16
3.1.4.5	Ergebnisausgabe . . . . .	16
3.1.5	Workflow in MATLAB . . . . .	17
3.1.5.1	Vorarbeit . . . . .	17
3.1.5.2	Verbesserungen . . . . .	17
3.1.5.3	Implementierung und Probleme in der Entwicklung . . . . .	18
3.1.5.4	Auswertung . . . . .	20
3.1.5.5	Schwierigkeiten in MATLAB . . . . .	20
3.2	FP4 - Messen, Prüfen, Qualitätskontrolle (3 Wochen) . . . . .	21
3.2.1	Messstellenpläne . . . . .	21
3.2.2	Messungen . . . . .	21
3.2.2.1	Durchführen der Messung . . . . .	22
3.2.2.2	Messdokumentation . . . . .	22
3.2.2.3	Messungsarten . . . . .	23
3.2.2.4	Datenverarbeitung in Excel . . . . .	25

	3.2.2.5	Auswertung und Weiterverarbeitung der Daten . . . . .	25
	3.2.2.6	Messtechnik . . . . .	26
3.3		FP3 - Instandhaltung, Wartung und Reparatur (x Wochen) . . . . .	27
	3.3.1	Fahrzeugverwaltung mit SAP . . . . .	27
	3.3.2	Fahrzeugumbauten für Testzwecke . . . . .	29
	3.3.3	Fahrzeuginstandhaltung und Betriebsbereitschaft . . . . .	29
<b>Literatur</b>			<b>31</b>

# 1 Die Porsche AG

Porsche wurde 1931 von Ferdinand Porsche gegründet. Seit 1945 stellt die Firma Sportwagen und Luxusautos her.

Im Finanzjahr 2018 erzielte der Konzern einen Umsatz von 25,784 Mrd. Euro.

Die aktuelle Produktpalette der AG umfasst für Serienfahrzeuge:

- Macan (2013)
- Cayenne (2002)
- 911 (1963)
- Boxster (1996)
- Panamera (2009)

## Weissach

Seit 1971 werden diese Fahrzeuge am Standort Weissach entwickelt. Das sogenannte "Entwicklungszentrum Weissach", kurz EZW, auf dem die Porsche AG ein Fahrzeug entwickeln, testen und justieren kann, bevor es in Serienproduktion in Zuffenhausen oder einem anderen Standort geht.

Alleine an diesem Standort beschäftigt Porsche um die 6500 Mitarbeiter und Mitarbeiterinnen. Zuletzt wurde hier auch der Porsche 918 entwickelt: Ein Supersportwagen mit hybridem Antrieb und einem Listenpreis von mehr als 800 000 Euro.

Das in Weissach entwickelt wird, sieht man schon an der Anzahl an Patenten die hier jährlich angemeldet werden. Knapp 400 pro Jahr sind es aktuell, mit mehr als 7000 bereits zugelassenen Patenten.

Die Signifikanz des Standortes Weissach wird immer wieder durch spezielle Produktvarianten oder Ausstattungen hervorgehoben. Ein Beispiel hierfür ist das Weissach-Paket, welches man für den 911 bestellen kann. Teil hiervon sind typischerweise die Farben Grün und Weiß, zusammen mit besonderen Aero-Elementen.



## 2 Das Praktikum

### 2.1 Fachpraktikum

Für das Fachpraktikum mussten noch 14 Wochen absolviert werden. Alle Bereiche des Grundpraktikums wurden bereits im Vorpraktikum abgearbeitet, weshalb sich das Fachpraktikum nur auf das in den Praktikumsrichtlinien der RWTH Aachen genannte Fachpraktikum Teil A und Teil B bezieht.

Die 8 Wochen aus Teil B konnten alle mit FP7 abgegolten werden, da in Weissach ausschließlich Fahrzeugentwicklung betrieben wird. Die übrigen sechs Wochen aus Teil A wurden mit FP4 - **Messen, Prüfen, Qualitätskontrolle** - mit einem Anteil von drei (3) Wochen und FP6 - **Montage** - mit weiteren drei Wochen ausgeführt.

### 2.2 Fachbereich Porsche

Bei Porsche wurde das Praktikum im Fachbereich **Aerodynamik und Thermomanagement** absolviert.

Diese Abteilung ist für die thermische Absicherung der Fahrzeuge und die aerodynamische Auslegung der Neuentwicklungen zuständig.

Neben Versuchen im Klimawind- oder Windkanal werden auch Erprobungsfahrten, Simulationen und andere Mess- und Testmethoden verwendet um die hohen Standards bei Porsche aufrecht zu erhalten.

Im engen Austausch mit den anderen Abteilungen und Bauteilverantwortlichen wird so jedes Fahrzeug und Einbauteil erprobt und entsprechend der geltenden Normen und Rechtssprechungen, sowie der Anforderungen die Porsche an sich selbst stellt, entwickelt und zur Serienreife gebracht.



## 3 Praktikumsbericht

### 3.1 FP7 - Entwicklung (8 Wochen)

Der Anfang des Praktikums wurde mit Einarbeiten verbracht.

Zu den verwendeten Tools gehören unter anderem PowerFlow, PowerTherm, ANSA, Star CCM+ und MATLAB.



Abb. 3.1: PowerFlow von 3DS

Vor allem PowerFlow spielt eine große Rolle, da es für Fahrzeugsimulationen geeignet ist. Das Programm baut auf der **Lattice-Boltzmann-Methode** auf.

### 3.1.1 Die Lattice-Boltzmann-Methode (LBM)

Die Lattice-Boltzmann-Methode beruht auf einer Vereinfachung der Euler-Gleichung und ist aufgrund geringer Speicheranforderungen pro Zelle geeignet um komplexe Geometrien zu berechnen.

Da Fahrzeuge nur selten bis in den kompressiblen Machzahlbereich von

$$Ma > 0.3 \tag{3.1}$$

ausgelegt werden, können die getroffenen Vereinfachungen angewendet und dadurch die **LBM** für die Fahrzeugauslegung verwendet werden.

Da die LB-Methode leicht für die Wärmeübertragung erweitert werden kann, spielt bei der Auslegung der Fahrzeuge auch PowerTHERM eine große Rolle für die Bauteilsicherung.

### 3.1.2 Einarbeitung

Für die Einarbeitung in PowerTherm und PowerFlow wurden Tutorien durchgearbeitet.

Um sich mit den Funktionalitäten des Programms vertraut zu machen, wurde zuerst eine einfache NACA-Flügelgeometrie geladen und diese dann mit den entsprechenden Netzen und Berechnungsvolumen ausgestattet um eine Berechnung zu ermöglichen.

Anschließend mussten die Randbedingungen an den Simulationsvolumengrenzen definiert werden und die Strömungskonditionen definiert werden. Hierzu zählen die Reynoldszahl, Temperatur, Strömungsgeschwindigkeit und Vieles mehr.

Nachdem die Berechnung abgeschlossen ist, wird dann PowerCASE, ein Programm zur Darstellung der Simulationsergebnisse, verwendet um sich verschiedene relevante Parameter oder Geschwindigkeiten, Drücke und vieles mehr anzeigen zu lassen.



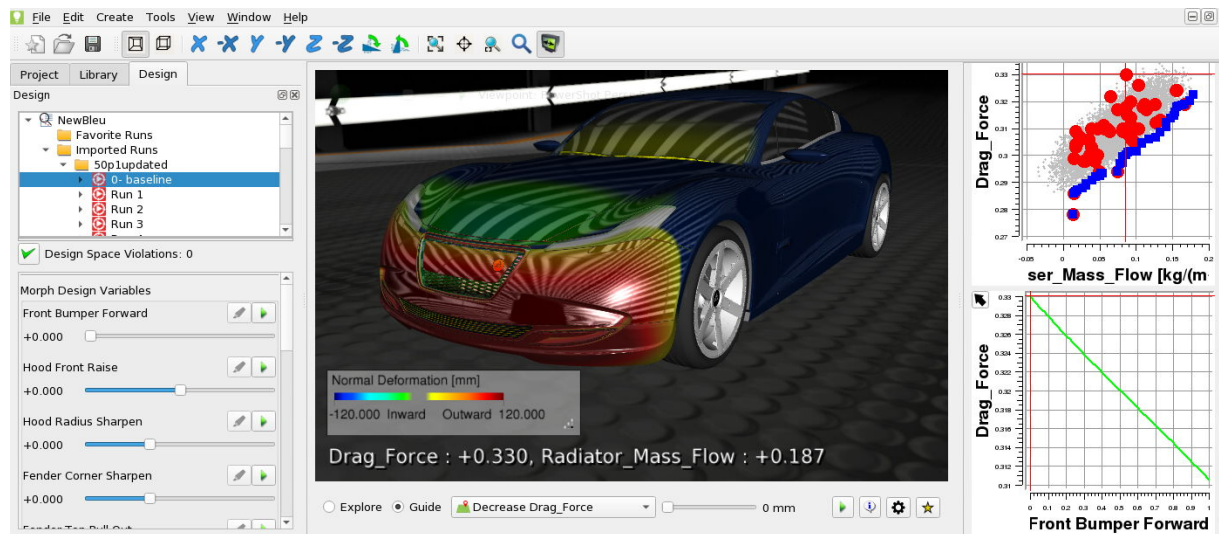


Abb. 3.2: PowerFlow User-Interface

Nach dem NACA-Profil wurde als nächstes mit ANSA, einem Pre-Processing Tool, gearbeitet. In diesem können Geometrien für die Simulationen aufgearbeitet und vernetzt werden.



Abb. 3.3: ANSA Logo

Abgesehen von der recht veralteten Benutzeroberfläche ist ANSA was die Geometrievorbereitung für PowerFlow gut geeignet, da die Geometriegüte in PowerFlow für verwertbare Ergebnisse nicht perfekt sein muss.

In ANSA wurde nun eine einfache Geländewagengeometrie aufgebaut, die dann in PowerFlow geladen und mit einem einfachen Simulationsvolumen berechnet wurde. Anschließend wurden die Ergebnisse wieder in PowerCASE ausgewertet und mit anderen Praktikanten verglichen.

Wenn man PowerFlow mit Star CCM+ vergleicht kommt heraus, dass CCM+ zwar etwas genauere Ergebnisse liefert. Dafür benötigt es jedoch eine sehr viel höhere Güte an eingeladener Geometrie, was es für die aktuelle Anwendung zum Teil ungeeignet macht.

Am Ende der Einarbeitung wurde noch eine Einweisung in die Fahrzeughandhabung auf dem Werksgelände durchgeführt. Da die Fahrzeuge selbst auf der niedrigsten Ausstattungsstufe mehr **kW** entwickeln als die meisten andere Kraftwagen, ist hier besondere Vorsicht geboten. Vor allem, wenn man Fahranfänger ist.

### 3.1.3 Datenauswertung in Excel

Als erstes Projekt wurde eine Datenauswertung in Excel zugewiesen.

Um mit den großen Datenmengen umzugehen wurde in Excel mit Hilfe des integrierten VBA-Editors ein Makro geschrieben. Dieser kann nach minimaler Vorarbeit des Anwenders automatisch die Datenmengen auswerten und das entsprechende Kollektiv erstellen.

Die hinterlegte Logik und Datenstrukturen mussten zuerst entworfen und dann implementiert und getestet werden.

Da VBA (Virtual Basic) eine free-type Programmiersprache ist, können Datentypen direkt im Code gecasted und definiert werden. Dies ist im Vergleich zu einer Sprache wie JAVA zwar schneller zum Tippen, ist aber bei der Code-Compilation und späteren Ausführung langsamer, da der Interpreter die Datentypen bei der Ausführung zuweisen muss.

Ein weiteres Problem, das speziell bei der Anwendung von VBA in der Microsoft Office-Umgebung entsteht, ist die extrem langsame Zugriffszeit von dem VBA Editor in das entsprechende Dokument in z.B. Excel.

#### 3.1.3.1 Makroversion 1.0

In der ersten Version des Makros wurde in den entsprechenden Iterationsschleifen direkt auf das jeweilige Worksheet in Excel zugegriffen um die Messdaten auszulesen und vergleichen zu können.

Die grundlegende Logik ist in Abbildung [3.4](#) zu sehen.

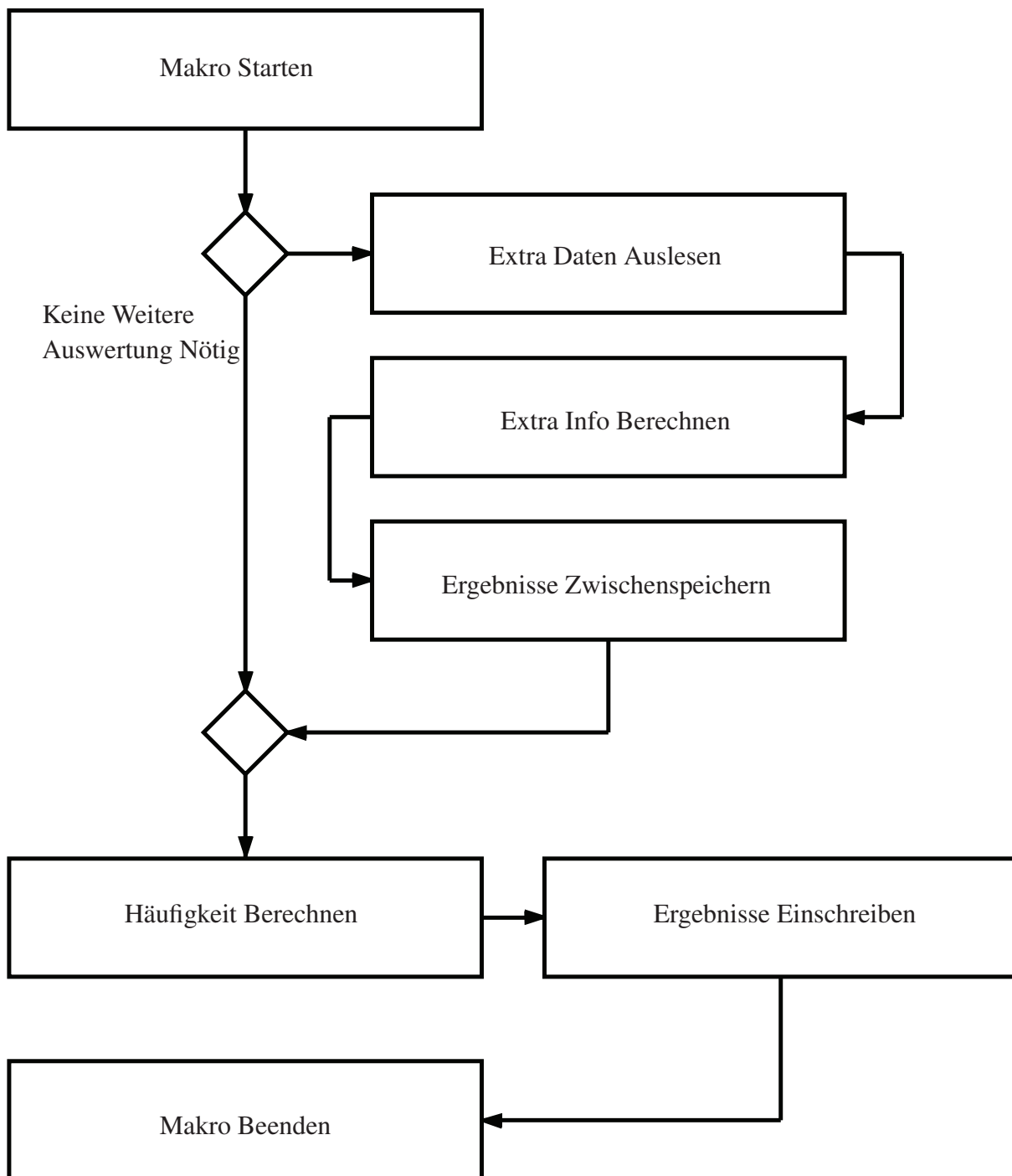


Abb. 3.4: Makro-Logik

Da die Messdatenmengen des Versuchs aus mehreren tausend Messpunkten besteht, brauchte der Makro für die Iteration über alle Datenpunkte geschätzte 120 Stunden. Diese relativ lange Bearbeitungszeit war zwar zum Teil auf die Hardware zurückzuführen, wurde jedoch vor allem

von dem oben beschriebenen Problem hervorgerufen.

Bei einer Messungsreihe mit 5 verschiedenen Messumgebungen die aus je 12 Messtemperaturen bestehen wird die Datenmenge schnell sehr groß. Die Messdaten wurden auf eine uniforme Zeitbasis von 1 Hz bestimmt. Je nach Messumgebung variierte auch die Messdauer. In der Testumgebung wurden daher nur etwa 20 000 Datenpunkte ausgewertet.

Da selbst in der Testumgebung die Berechnungsdauer bei mehreren Stunden lag, musste die Berechnungszeit verkürzt werden.

### **3.1.3.2 Makroversion 1.5**

Nach erweiterte Recherche konnte die Berechnungszeit verkürzt werden, indem die Datenauslegung am Anfang der Iteration einmal durchgeführt wurde und dann nur noch mit den eingelesenen Daten direkt in der VBA Umgebung gerechnet wurde.

Hier war nun vor allem die Indizierung der Matrizen eine Herausforderung. Je nachdem ob bestimmte Messinformationen mit ausgewertet werden mussten, mussten die Indizes und die Iterationsschrittweite angepasst werden.

Dies wurde durch eine VBA Userform gelöst, in welcher der Benutzer angeben konnte ob die Information benötigt wird. Im Code wurde basierend auf dieser Auswahl eine Iterationsvariable und die Iterationsgrenzen angepasst.

Nach weiterem Testen war diese Art der Auswertung erfolgreich.

Nun musste nur noch das Problem der Matrixindizierung gelöst werden. Um den Makro möglichst benutzerfreundlich zu gestalten, mussten die Matrixgrenzen so einprogrammiert werden, dass die Auswertung auch funktioniert wenn der Benutzer die Daten falsch in Excel einfügt oder Daten fehlen. Hierfür mussten in die Iterationsschleifen Logikabfragen implementiert werden, da die Fehlerbehandlung in VBA fragwürdig ist. Diese Maßnahme verlangsamten die Auswertung wieder etwas. Jedoch war es möglich durch Deaktivieren der Bildschirmanpassung und Hintergrundberechnung von Excel selbst diese Rechenzeit wieder auszugleichen.

### **3.1.3.3 Makroversion 2.0**

Nachdem die Matrixauswertung erfolgreich war, mussten die Auswertungsergebnisse noch aufgearbeitet werden. Die Ergebnismatrix wurde dann in ein eigenes Worksheet geschrieben. In diesem Worksheet ist Code hinterlegt, der die Matrixdaten in einer 3D-Grafik wiedergibt.

Mit dieser Matrix und dem entsprechenden Graphen können dann die Gespräche mit den Zulieferern und Lebenszeitexperten geführt werden im das Produkt möglichst optimiert zu entwickeln.

Durch die oben genannten Optimierungen konnte die Berechnungsdauer in der finalen Version des Makros auf etwa 20 Minuten reduziert werden. Diese Zeit wurde in der echten Auswertungsumgebung mit ca. 70 000 Datenpunkten erreicht.

Ein weiterer Grund warum diese Auswertung jedoch in Excel nur bedingt Effizient ist, ist der pre-processing Aufwand um verwertbare Ergebnisse zu erlangen. Die Daten müssen zuerst mit anderen Programmen aus der Messdatei ausgelesen und dann in Excel exportiert werden. Aus dieser Excel-Datei müssen die relevanten Daten dann in das Excel-Workbook mit dem hinterlegten Makro übertragen werden. Bei insgesamt 60 Messläufen ist dieser manuelle Aufwand wesentlich größer als die später benötigte Zeit um die Auswertung durchzuführen.

#### **3.1.3.4 Das Excel-Workbook**

Das Excel-Workbook in welchem der Makro hinterlegt ist wurde als Vorlage (Template) erstellt und in einem entsprechendem Ordner abgespeichert um den restlichen Abteilungsmitgliedern den Zugriff auf das Workbook zu ermöglichen.

Da in der Fahrzeugherstellung viele verschiedene Umweltzustände und dadurch Temperaturen dargestellt und getestet werden müssen, sind in dem Excel-Workbook verschiedene Worksheets hinterlegt, in denen die relevanten Umgebungsdaten hinterlegt sind.

Um diese in der Berechnung automatisch mit zu Berücksichtigen müssen diese vor Starten des Makros entsprechend über eine User-Form ausgewählt werden. Um die Bearbeitung zu erleichtern ist als Standard-Wert der am häufigsten verwendete Fall hinterlegt.

Die Lebensdauerfaktoren werden in dem ersten Worksheet festgelegt und berechnet. Der Makro liest sich diese Werte entsprechend aus und verwendet sie. Daher ist es wichtig, dass das Makro-Template immer den neuesten Werten entsprechend angepasst wird. Diese Arbeit muss leider weiterhin manuell ausgeführt werden, da die Lebensdauerfaktorenberechnung oft in Arbeitskreisen durchgeführt wird und diese Faktoren dann nicht in einer zentralen Datenbank hinterlegt werden.

**(Notiz)**

In diesem Bereich ist nach Meinung des Authors weitere Entwicklung gewünscht um Arbeitsprozesse zu Streamlinen und die gesamte Entwicklung effizienter zu gestalten.

**(Notiz-Ende)**

Für die Weiterverarbeitung der berechneten Daten ist es nun günstiger, die resultierenden Matrizen aus dem Makro-Worksheet in ein Worksheet ohne hinterlegte Makros und Berechnungsparametern zu transferieren. Dies resultiert aus der Geheimhaltungsklausel der Porsche AG. Die Makro-Datei hat unter anderem konzern-interne Daten hinterlegt, die nicht von Zulieferern oder Dritten Parteien eingesehen werden dürfen.

Hier könnte der Makro erweitert werden, damit die Ergebnisse automatisch in ein neues Excel-Workbook übertragen werden.

Ein weiterer Mangel an dem Marko-Workbook besteht darin, dass während der Makro läuft Excel unbenutzbar auf diesem Rechner wird. Der von Excel blockierte Thread wird komplett von dem Makro übernommen und lässt das Betriebssystem Excel als ein Programm ohne aktive Rückmeldung anzeigen.

Dies muss leider mit der aktuellen Version des Makros hingenommen werden.

Aufgrund des Wunsches diesen Workflow in MATLAB effizienter zu gestalten wurde an dieser Stelle der Makro nicht mehr weiterentwickelt.

**3.1.3.5 Endresultat**

Wenn der Makro erfolgreich durchlaufen ist, werden je nach Benutzereingabe eine oder zwei Matrizen ausgegeben.

Die erste Matrix wird immer erstellt und besteht aus dem Temperaturkollektiv der Messdateien. die zweite Matrix wird nur erstellt, wenn die Information in der Messung enthalten und diese Information für die weitere Entwicklung benötigt wird. Da diese Matrix den Großteil der Berechnungsdauer in Anspruch nimmt, wäre es redundant diese Berechnung durchzuführen wenn sie nicht benötigt wird.

Da dieser Gesamtprozess wie oben aufgeführt ineffizient ist, bestand einer der weiteren Aufgaben darin diesen Prozess in MATLAB zu verbessern und den Workflow damit komplett zu überarbeiten.

Nach der Erfahrung des Authors nimmt beim Entwicklungsprozess die Datenaufarbeitung einen großen Anteil der gesamt benötigten Zeit und damit einen Großteil der Kosten auf sich. Die Kosten stellen sich aus Lizenzkosten der benötigten Programme und den direkten Personalkosten zusammen.

Daher kann mit einer entsprechenden Workflow-Verbesserung auch eine Kostensenkung erwirkt werden.



### 3.1.4 Aktueller Workflow der Messungsauswertung

Der in **Datenauswertung in Excel** [3.1.3] beschriebenen Workflow beinhaltet nur einen Bruchteil der benötigten Arbeitsschritte um verwertbare Ergebnisse aus dem Gesamtprozess zu gewinnen.

Da die Messdaten nur der letzte Teil einer Arbeitskette darstellen, wird nun auf den gesamten Prozess eingegangen. Dieser wird erläutert, die Problemstellen hervorgehoben und eine Lösung in MATLAB für den Workflow vorgestellt.

#### 3.1.4.1 Messstellenplan

Der Messstellenplan oder (**MSP**) wird von den Messstellentechnikern und dem zuständigen Mitarbeiter erstellt und in Zusammenarbeit mit den relevanten Bauteilverantwortlichen und anderen Parteien kontinuierlich angepasst.

Die pro Versuchsreihe oftmals eine signifikante Anzahl von Messungen gefahren werden ist es recht aufwändig den Messstellenplan jedes Mal an die neue Messung anzupassen.

Der Messstellenplan wird über konzern-eigene Datenverwaltungstools für jeden Wagen gepflegt. Dieser kann dann in Excel exportiert werden und für jede relevante Messung referenziert werden.

Hierbei entstehen schnell eine unübersichtliche Anzahl von Excel Dateien mit zum Teil extensiven Bezeichnungen um eine genaue Zuordnung zu ermöglichen.

#### 3.1.4.2 Grenztemperaturen

Die Grenztemperaturen werden in einer separaten Datei für jede Messstelle, insofern diese Messstelle eine Grenztemperatur hat, festgelegt.

Analog zum **MSP** werden diese Grenztemperaturpläne in Excel exportiert.

Aktuell werden diese Dateien dann in einer separaten Excel-Datei mit hinterlegten Makros eingelesen und ausgewertet. Da VBA eine recht alte und langsame Programmiersprache ist, ist diese Auswertung oft zeitaufwändig oder stürzt einfach ab.

#### **3.1.4.3 Messungen**

In der Excel-Datei können die Messungen aus ihrem nativen MDF-Format nicht eingelesen werden.

Daher müssen um die Messungen zuzuordnen erst die relevanten Kanäle mit Hilfe eines Porsche-internen Tools ausgelesen und in Excel exportiert werden.

Diese exportierten Excel-Dateien sind regelmäßig weiter über 100mb groß und benötigen dementsprechend lang um eingelesen zu werden.

Bei mehreren Messungen pro Auswertung ist hier der Aufwand schnell groß, da jede Messung einzeln verarbeitet werden muss.

#### **3.1.4.4 Simulationen**

In dem aktuellen Workflow gibt es keine Möglichkeit Simulationsergebnisse der Auswertung anzufügen.

Da diese aber oft für Erstausslegungen oder Konzept-Bestätigung verwendet werden, wäre es von Vorteil diese direkt mit einsehen zu können.

#### **3.1.4.5 Ergebnisausgabe**

Die Ergebnisse werden entweder als PDF oder als Excel Workbook ausgegeben.

Diese werden dann oft in Powerpoint oder anderen Medien weiterverwendet um entsprechende Maßnahmen abzuleiten.

### 3.1.5 Workflow in MATLAB

Um diesen Workflow effizienter zu gestalten wurde er in MATLAB übertragen und dort in einem Stand-alone Programm entworfen.

Der erste Arbeitsschritt bestand darin, sich mit der MATLAB-App Designer Umgebung vertraut zu machen. Da MATLAB sehr effizient große Matrizen verarbeiten kann, bot sich das Programm für diese Art der Auswertung an.

Nach Durcharbeitung der Tutorials wurde als nächstes der Workflow entworfen.

Dieser sollte idealerweise die vorher genannte Auswertung in Excel und anderen Programmen komplett ersetzen.

Damit das Programm auch von den Kollegen eingesetzt wird, wurden die relevanten Parteien von Anfang an in den Entwicklungsprozess miteingebunden. Hierfür wurden wöchentliche Meetings angesetzt.

Im Laufe der Entwicklung des Workflows und Programms wurde schnell offensichtlich, wie wichtig diese andauernde Rückmeldung mit den anderen Teammitgliedern für eine erfolgreiche Implementierung ist.

#### 3.1.5.1 Vorarbeit

Um den Workflow zu entwerfen musste erst der gesamte Auswertungsprozess verstanden und nachvollzogen werden.

Um dies zu erreichen wurden kleine Aufgaben entlang des gesamten Prozesses übernommen und selbstständig bearbeitet. Bei Unklarheiten wurden die relevanten Kollegen hinzugezogen.

Als nächstes wurde der Ablauf ausgelegt und dann Überlegungen angestellt, wo er beschleunigt und effizienter durchgeführt werden kann.

#### 3.1.5.2 Verbesserungen

Die einfachste Verbesserung wurde im Bereich der Messstellenpläne implementiert. Diese können nun aus der exportierten Excel-Datei direkt in das Programm geladen werden.

Sollte eine neue Messung einen veränderten Messstellenplan haben, kann diese Veränderung in den Messstellenplan der bereits eingelesen ist eingefügt werden.

Um das weitere Vorgehen zu erleichtern und das Vergleichen von Messungen zu ermöglichen, werden neue Messstellen an den alten Plan angefügt und dann einsortiert. Keine Messstellen werden entfernt, damit alle vorhergehenden Messungen ihre Aussagekraft bewahren.

Des Weiteren wird der Grenztemperaturplan nur temporär eingelesen und die Werte für die relevanten Messstellen ausgelesen, wonach der Grenztemperaturplan wieder aus dem Arbeitsspeicher gelöscht wird.

Analog kann, sollte der Messstellenplan angepasst werden, die Grenztemperatur für die neuen Messstellen dann hinzugefügt werden. Sollte sich eine Grenztemperatur für eine bestehende Messstelle geändert haben, wird die alte Grenztemperatur automatisch überschrieben. Diese Maßnahme soll der Benutzerfreundlichkeit des Programms dienen.

Um die Messungen einzulesen, bedarf es nun keinem Dritt-Programm mehr. Das MATLAB-Programm gleicht den eingelesenen Messstellenplan in MDF-Format mit der gesamten Messung ab und lädt nur die Messkanäle die für den Plan von Relevanz sind.

Sollten weitere Kanäle wie zum Beispiel die GPS-Geschwindigkeit oder Ähnliches wichtig sein, können diese Messstellen manuell in den Messstellenplan in MATLAB eingefügt werden. Hierdurch werden diese Kanäle dann mitverwertet.

Genauer hierzu in Kapitel [3.1.5.3](#).

Die Simulationsergebnisse liegen normalerweise in **.csv** oder **.xml** Format vor.

Da MATLAB diese Formate nicht nativ einlesen kann, muss für diese Funktion noch eine Routine geschrieben oder eine verfügbare Funktion gefunden werden.

Aufgrund der wichtigeren Grundfunktion des Programmes wurde diese Funktion aber auf die 'Nice-To-Have'-Liste gesetzt. Daher wird an dieser Funktion nur gearbeitet, wenn der Rest des Programms zufriedenstellend läuft.

Das Ziel der Verbesserungen ist eine signifikante Zeiteinsparung und Vereinfachung des Workflows der Messungsauswertung und Ergebnisinterpretation.

### **3.1.5.3 Implementierung und Probleme in der Entwicklung**

#### **Version 0.1**

In der ersten Version des Programms wurde der Fokus darauf gesetzt die Grundfunktionen der Auswertung linear zu implementieren.

Dies sollte auch helfen den Grundprozess besser zu verstehen. Das Programm selbst sollte dann iterativ verbessert und ausgebaut werden bis hin zur finalen Version.

Die ersten Probleme bei der Messstellenplanverwertung traten recht früh auf. Einige Messstellenpläne aus dem alten und dem neuen Datenverwaltungstool von Porsche waren zwar alles Excel-Dateien, aber mit verschiedenen Datentypen abgespeichert.

Zwischen diesen Datentypen gab es bei der Spaltennummerierung und -benennung Unterschiede, was das Auslesen der korrekten Information erschwerte.

Nach Rücksprache mit dem Betreuer ergab sich, dass in kurzer Zeit das alte Datenverwaltungstool komplett durch das neue ersetzt werden würde. Daher wurde das Problem teilweise gelöst, jedoch bestand weiterhin in der Nomenklatur eine Diskrepanz.

Weitere Maßnahmen um die Spaltenbenennung zu vereinheitlichen sind daher nötig.

Meiner Meinung nach würde dies den Workflow verbessern, da man bei der Auswertung, ob manuell oder per Programm, immer mit der gleichen Informationsstruktur konfrontiert ist.

Die endgültige Lösung des Problems bestand dann darin, die Spalten mit den entsprechenden Namen einzulesen und in einer vordefinierten Reihenfolge zu speichern.

Als weitere gewünschte Funktion sollte der MSP direkt in MATLAB bearbeitet werden können. Da die grafische Umgebung von MATLAB in diesem Fall eher langsam arbeitet, wurde abgeraten dies im Programm zu machen.

Die Möglichkeit Messstellen umzubenennen, zu löschen oder hinzuzufügen wurden dennoch grafisch implementiert.

Um den Grenztemperaturplan einzulesen, der auch in Excel exportiert wird, konnten die bereits implementierten Methoden wiederverwendet werden.

Analog zum Messstellenplan ist das Hinzufügen, Löschen und Umbenennen oder Editieren der Grenztemperaturen und Messstellen hier möglich. Dafür wird die eingelesene Information in einer Tabelle dargestellt.

Die signifikanteste Zeiteinsparung wird bei dem Auswerten der Messungen erzielt.

Um die Messungen einzulesen wurden Porsche-interne Routinen verwendet. Diese lesen aus den **.mf4**-Messdateien die relevanten Informationen wie Zeitskala, Werte, Messstellennamen und -info aus.

Diese Informationen werden in einem Cell-Array gespeichert und können so abgerufen werden. Dann gleicht eine Funktion die Messstellennamen mit dem Messstellenplan und Grenztemperaturplan ab und speichert alle gefundenen Übereinstimmungen in ein weiteres Cell-Array, in dem auch der MSP mit den Grenztemperaturen ist, ab.

Dieses Array ist vordefiniert als 3-D Array, welches in Richtung der Messstellen, oder der ersten Dimension, dynamisch erweitert werden kann.

Obwohl das dynamische Erweitern von Arrays in MATLAB die Rechendauer verlängert, war dies die effizienteste Lösung.

Die zweite Dimension ist definiert und beinhaltet alle relevanten Daten der einzelnen Messstellen, sowie die jeweiligen Messstellen.

Die dritte Dimension beinhaltet die Messdaten der Messstellen. Diese Dimension wurde als  $k$  vordefiniert mit

$$k = 100 \quad (3.2)$$

Da MATLAB sonst die zum Teil aus mehreren Hundert Messstellen bestehenden Messungen dynamisch hinzufügen müsste, was zu einer signifikanten Verlängerung der Einlesezeit geführt hätte.

Vor dem Einlesen jeder Messung wird der Benutzer gefragt, ob die Messung neue Messstellen beinhaltet. Diese können dann bei Bedarf aus Excel eingelesen und zu den existierenden Messstellen hinzugefügt werden.

Die Messinformation der Messungen werden in einem zweiten Array abgespeichert, dass mit den gleichen Indizes arbeitet. So können jederzeit alle Information zusammen mit den Messungen abgerufen werden.

#### 3.1.5.4 Auswertung

Für die Auswertung müssen nach dem einlesen aller Daten noch einige Parameter vom Benutzer angepasst werden, je nachdem welche Informationen betrachtet werden sollen.

Dann wird aus den Daten eine PDF-Datei erstellt, die alle Messkurven mit den Informationen abbildet.

#### 3.1.5.5 Schwierigkeiten in MATLAB

Das größte Problem mit MATLAB ist die langsame grafische Umgebung.

Da jedoch die Konsolenumgebung schwieriger zu bedienen ist, wurde die grafische Lösung gewählt.

Eine weitere Schwierigkeit stammt aus dem Erstellen einer Speicherungsdatei. Um nicht bei jeder Teilauswertung alle Dateien neu einlesen zu müssen, ist ein weiterer Schritt die Erstellung einer solchen Speicherdatei zu ermöglichen.

Hierfür müssen alle Daten die in der GUI hinterlegt sind ausgelesen und abgespeichert werden. Dies wird bearbeitet, wenn am Ende des Projekts noch Zeit übrig ist.

## 3.2 FP4 - Messen, Prüfen, Qualitätskontrolle (3 Wochen)

### 3.2.1 Messstellenpläne

Wie in Abschnitt 3.1 ausgeführt, war die Auswertung der Messergebnisse nur ein Teil der Arbeit.

Um die Messergebnisse für die Auswertung zu gewinnen, mussten Vorarbeit und vor allem Messungen getätigt werden.

Bei dem Erstellen von Messstellenplänen war vor allem das Auswählen der Messstellennamen von großer Bedeutung.

Die Namen müssen eindeutig und leicht Verständlich gewählt werden um sicherzustellen, dass auch Dritte den Plan schnell verstehen können.

Die Bearbeitung und Erstellung dieser Pläne, die oftmals mehrere hundert Messstellen beinhalten, ist eine langsame und schlecht automatisierbare Tätigkeit.

### 3.2.2 Messungen

Ein Teil der Messungen kann auf der firmeneigenen Teststrecke in Weissach durchgeführt werden.



Abb. 3.5: Die Teststrecke in Weissach



Die Rundstreckentests werden zum Teil vor Ort durchgeführt. Wenn ein wärmeres Klima oder andere Streckenbedingungen erforderlich sind, wird auf eine andere Teststrecke im südlichen Teil von Europa ausgewichen.

Aufgrund der Corona-Beschränkungen wurden die Messungen während der Praktikumszeit fast ausschließlich in oder um Weissach durchgeführt.

Bevor die Messungen durchgeführt werden können, müssen die Fahrzeuge mit den entsprechenden Messstellen ausgestattet werden. Hierfür wird ein Auftrag gestellt und das Fahrzeug an die zuständige Abteilung übergeben.

### **3.2.2.1 Durchführen der Messung**

Aufgrund der Geheimhaltungsbestimmung der Porsche AG muss beim Messen von Neuentwicklungen oder Derivaten darauf geachtet werden, dass keine Außenstehenden die Messfahrt sehen oder aufnehmen können.

Bei einer Messung bemerkte der Werkschutz wie ein externer Bauarbeiter die Messfahrt filmte, weshalb die Messung abgebrochen werden musste. Um zu gewährleisten, dass sich dieser Vorfall nicht wiederholte wurde die Messung dann auf einen Samstag verlegt. Dies war mit erheblichem Aufwand verbunden. Hier wäre es der Meinung des Autors nach praktisch gewesen, auch die Bauarbeiter einer Geheimhaltungsklausel zu unterstellen.

Die übrigen Messungen verliefen jedoch ohne größere Zwischenfälle.

Für die Messungen mussten die Fahrzeuge oft enttarnt werden, da die Tarnung die Ergebnisse verfälschen könnte.

Dann mussten die diversen Messeinrichtungen und -geräte gestartet werden, woraufhin die Messung durchgeführt werden kann.

### **3.2.2.2 Messdokumentation**

Für jede Messung wurde ein Dokument angelegt, auf dem alle relevanten Informationen der Messung hinterlegt sind.



Information	Beschreibung
Wageninformationen	Der Wagennamen und die genaue Bezeichnung
Messdatum	Datum der Messung
Wagenkonfiguration	Aerokonfiguration, Bordcomputereinstellungen
Personal	Zuständiger Messstechniker, Fahrer
Messbedingungen	Außentemperatur, Luftfeuchtigkeit
Streckendaten	Anzahl der Runden, Rundenzeiten

Tabelle 3.1: Messinformationen

Wie in Tabelle 3.1 dargestellt, müssen am Anfang der Messung einige Daten erfasst werden. Dies passiert am Anfang der Messung.

Sobald diese Daten erfasst worden sind, beginnt der Test. Die Rundenzeiten werden über eine einfache Stop-Uhr gemessen. Die Rundenzeiten sollten während einer Messung ungefähr gleich sein um zu gewährleisten, dass sich keine signifikanten Parameter während der Messung geändert haben.

Außerdem gilt die Rundenzeit als Indikator, dass dem Fahrer nichts passiert ist. Sollte die Zeit stark überschritten werden, kann man den bereitstehenden Rettungskräften sofort Bescheid geben.

### 3.2.2.3 Messungsarten

Die Messungen lassen sich in einige übergreifende Kategorien aufteilen:

- Rundstreckentests
- Klimakanaltests
- Windkanaltests
- Hochgeschwindigkeitstests
- Bergfahrttests

Je nach Messungsart muss das Fahrzeug nach der Messung noch aufgeheizt werden. Hierfür wird eine Aufheizbox verwendet.

Das Fahrzeug bleibt hierbei nach der Messung so lange in der Box stehen, bis die Temperaturen wieder anfangen zu fallen. So wird gewährleistet, dass die Komponenten ihre größtmögliche Wärme entwickeln können da in der Box nur geringer Luftaustausch mit der Umgebung stattfinden kann.

Hierdurch wird das sogenannte "Worst-Case-Szenario" dargestellt, in dem der Kunde das Fahrzeug unter Volllast betreibt und dann an einem windstillen Ort abstellt.

Aufgrund einer ausreichend genauen Annäherung kann das Messergebnis auch linear auf Umgebungstemperaturen wie in Gleichung 3.3 verrechnet werden.

$$T_{UMG} = 50^{\circ}\text{C} \quad (3.3)$$

Nachdem die Messdaten aus den diversen Messcomputern ausgelesen wurden, müssen erst einmal einige Zeitstempel aus den Daten ausgelesen werden. Dies dient dazu, die Fahrtphasen zu differenzieren.

Hierfür hat Porsche ein internes Software-Tool. In diesem sind schon einige Grundfunktionen zur Datenauslesung implementiert. Unter anderem kann man sich die Messkurven grafisch darstellen lassen und dann aus diesen Grafiken beliebige Messpunkte auslesen. Des weiteren ist es möglich, die Messdaten auf bestimmte Zeitintervalle zu begrenzen und so nur die benötigten Daten auszulesen.

Wenn nur einzelne Punkte zwischen Messungen verglichen werden sollen, ist es effizienter die benötigten Punkte direkt in diesem Programm zu identifizieren und auszulesen, da es grafisch gemacht werden kann.

Exportiert man die Ergebnisse erst einmal in Excel oder einem anderen Programm, hat man nur noch die Rohdaten vorliegen und müsste sich so aus einer beliebig großen Matrix die Datenpunkte zusammensuchen.

Bei einer Auswertung bei der verschiedene Kühlerkonfigurationen und deren Auswirkung auf die Motoransauglufttemperatur untersucht werden sollte, war dieses Vorgehen vorteilhaft. Da nur bestimmte Betriebspunkte für den Vergleich relevant sind, konnte so einiges an Auswertungszeit gespart werden.

Für die generelle Messungsauswertung wurden die Messdaten meistens in Excel exportiert und dort weiter ausgewertet.



Abb. 3.6: Beispielbild: Der neue Taycan beim Test auf dem Nürburgring<sup>1</sup>

[1] Quelle <https://newsroom.porsche.com/de/2019/produkte/porsche-taycan-rekord-nuerburgring-nordschleife-7-42-minuten-18439.html>

#### 3.2.2.4 Datenverarbeitung in Excel

Um Aussagen aus den Messdaten zu generieren, wurde diese erst einmal aufgearbeitet. Die Abtastraten und ähnliche Parameter konnten vor dem Export in Excel angepasst und auf die gleiche Zeitbasis gebracht werden.

Dies wurde durch Interpolation bei niedrigeren Abtastraten, soweit angebracht, oder bilden des Durchschnitts bei höheren Abtastraten erreicht. Dann konnten die alle benötigten Datenreihen der relevanten Messstellen ausgewählt und exportiert werden.

Aus diesem Excel-Worksheet kann man dann die Werte und Zeitachsen in eine entweder selbst erstellte oder bereits vorhandene Auswertungstabelle in Excel einfügen und daraus die Messkurven und Maxima, Minima und Durchschnittstemperaturen bestimmen.

#### 3.2.2.5 Auswertung und Weiterverarbeitung der Daten

Die in Excel erstellten Auswertungen sind meistens für die Bauteilverantwortlichen genügend. Jedoch müssen die Maßnahmen auch von den Vorgesetzten evaluiert und abgesegnet werden.

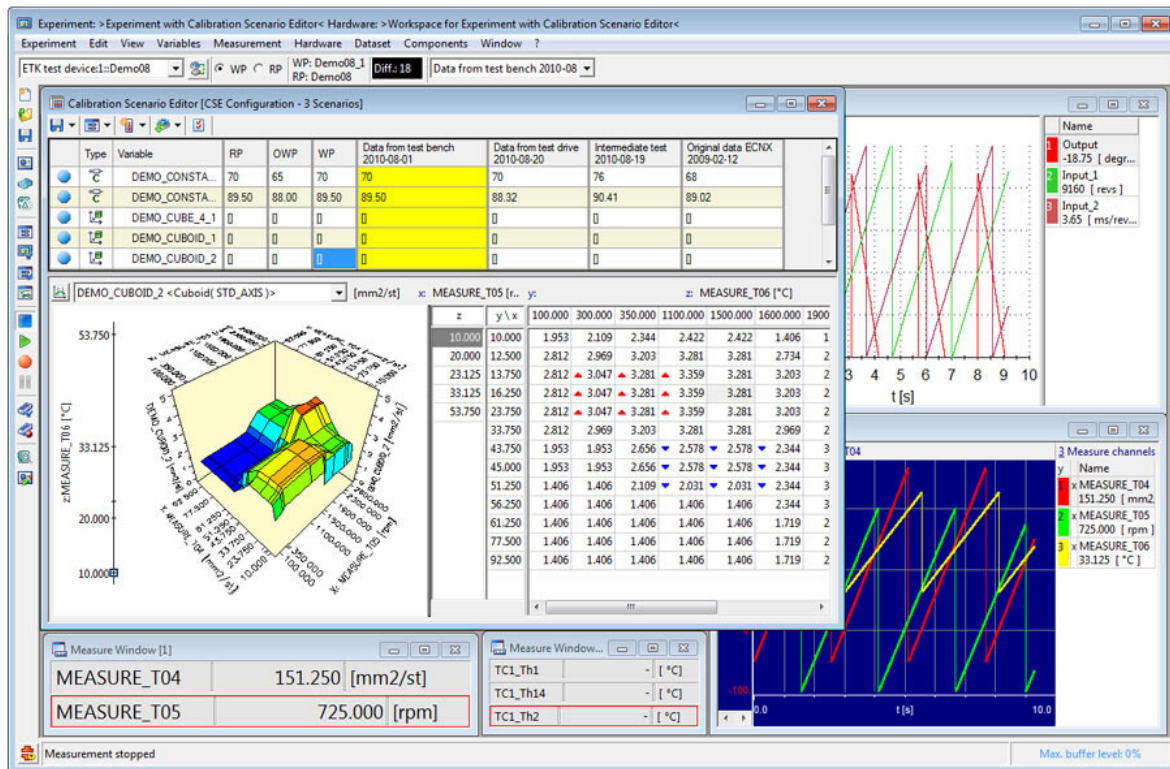


Abb. 3.7: Die INCA-Benutzeroberfläche  
 [2] Quelle: <https://www.etas.com/de/portfolio/inca.php>

Hierfür lohnt es sich, die wichtigsten Daten noch einmal schnell verständlich zusammen zu fassen und über z.B. PowerPoint zu einer Präsentation zu verarbeiten. Diese kann dann den Entscheidungsträgern vorgestellt werden.

### 3.2.2.6 Messtechnik

Die in den Messungen verwendete Technik stellt sich aus verschiedenen Herstellern zusammen. Einer der Hersteller ist **ETAS**. Die von ETAS vertriebene INCA-Softwareumgebung muss vor der Messung entsprechend eingestellt und kalibriert werden.

In **INCA** können durch eine integrierte Datenbank Voreinstellungen wiederverwendet werden. Außerdem kann durch die INCA die fahrzeug-eigene Regeltechnik angesteuert werden. Bei manchen Messungen werden die Kühler und andere Funktionen des Fahrzeug individuell eingestellt um verschiedene Szenarien abzubilden.

Die Software wird hier auf einem eigenem Messsystem betrieben, um mögliche Schwierigkeiten die aus den Schnittstellen mit herkömmlichen Betriebssysteme entstehen könnten vorzubeugen.



### 3.3 FP3 - Instandhaltung, Wartung und Reparatur (x Wochen)

Die Fahrzeuge auf dem Gelände des Entwicklungszentrums in Weissach müssen möglichst in einem fahrbereiten Zustand gehalten werden.

Hierzu gehört auch für die Fahrzeuge Transportaufträge zu anderen Standorten oder Reifenwechselaufträge vor Ort zu erstellen. Die Wagen müssen dann auch zu ihren jeweiligen Zielen gebracht und dort übergeben werden.



Abb. 3.8: Geparkte Wagen am Entwicklungszentrum Weissach

[3] Quelle: <https://www.krzbb.de/krz65111725728-13-Rueckspiegel-Neue-E-Motionen.html>

#### 3.3.1 Fahrzeugverwaltung mit SAP

Um mit der SAP-Software in der Porsche AG-Umgebung arbeiten zu können, musste zuerst ein Antrag für die benötigte Berechtigung erstellt werden.

Nachdem dieser genehmigt wurde, konnte mit dem Verwalten der Versuchsfahrzeuge begonnen werden.

Wenn ein Wagen für einen Versuch an einem anderen Standort benötigt wird, kann dieser Aufgrund der Geheimhaltungsvereinbarung nicht zu diesem Standort gefahren werden. Stattdessen wird er über ein Logistikunternehmen dorthin verfrachtet.

Die Aufträge hierfür, genannt Transportaufträge, werden über das SAP System erstellt. In dem Auftrag ist der Wagen mit seiner Kennung, dem Verantwortlichen, der Kostenstelle und weiterer notwendiger Information hinterlegt.

Bevor er jedoch abgeholt werden kann muss dafür gesorgt werden, dass der Wagen so gut wie möglich auf den Test oder die Erprobung vorbereitet ist. Hierfür muss der Wagen noch getankt werden und überprüft werden, dass die richtige Bereifung aufgezogen ist.

Im Fahrtenbuch muss außerdem die genaue Kilometerzahl hinterlegt werden.

Dann kann der Wagen zur Abholstation auf dem Entwicklungszentrum gebracht werden.

Sollten die Reifen noch gewechselt werden müssen, muss auch hierfür ein Antrag in SAP erstellt werden. Die Reifen werden jedoch vor Ort gewechselt, weshalb die Prozedur meist schneller und einfacher von statten geht als ein Transport.



Abb. 3.9: Das Logo der SAP SE

[3] Quelle: <https://de.wikipedia.org/wiki/SAP>

Im SAP System sind die Fahrzeuge alle mit Equipmentnummern versehen, über die man sofort jedes Fahrzeug finden kann.

Zusammen mit den Fahrzeugdaten, dem Nummernschild und anderen relevanten Informationen

sind außerdem das Gewicht und das Volumen hinterlegt. Dies dient dem Zweck, den zuständigen Speditionen mitzuteilen mit welchen Geräten sie anrücken müssen um die Fahrzeuge zu verladen.

Ein Problem mit der Auftragserstellung ist, dass die Aufträge sehr schnell weiterverarbeitet werden. Hierdurch wird Fehlerbehebung recht aufwändig. Um einen bereits weiterverarbeiteten Auftrag im Nachhinein abzuändern, muss man nun die relevante Stelle anrufen und den Auftrag wieder zurückschicken lassen. Je nachdem wie die Kollegen erreichbar sind, kann dieser Prozess relativ viel Zeit in Anspruch nehmen, welche dann nicht mehr für andere eventuell wichtigere Dinge verwendet werden kann.

### **3.3.2 Fahrzeugumbauten für Testzwecke**

Da die Wagen und Prototypen oftmals für die jeweiligen Testzwecke umgebaut werden müssen, gibt es im Entwicklungszentrum einige Werkstätten die sich auf verschiedene Bereiche des Fahrzeugs spezialisieren.

Schon während der Erstellung der Messstellenpläne ist der Austausch mit diesen Werkstätten wichtig. Die geforderten Messstellen müssen technisch umsetzbar und einbaubar sein.

Wenn dieser Prozess erfolgreich abgelaufen ist, wird ein Umrüstungsauftrag erstellt. Dann wird der Wagen zu der jeweiligen Werkstatt gebracht und dort von professionellen Arbeitern genau auf die Anforderungen der Messingenieure ausgerichtet.

Nach den Testläufen müssen die Fahrzeuge wieder in ihren Ursprungszustand zurückgebaut werden. Hierfür muss ein neuer Auftrag erstellt und das Fahrzeug wieder zu der gleichen Werkstatt gebracht werden.

### **3.3.3 Fahrzeuginstandhaltung und Betriebsbereitschaft**

Während ein Fahrzeug im Besitz der Abteilung ist, ist es deren und damit die Verantwortung des Autors diese Fahrzeuge in einem betriebsbereiten Zustand zu halten.

Jeder Wagen hat ein Fahrtenbuch, in welchem die Kilometeranzahl und Fahrerinformationen notiert werden müssen.

Sollte der Wagen Kraftstoff benötigen, kann dieser auf der hauseigenen Tankstelle getankt werden. Hierfür hat jeder Wagen eine spezielle Keycard. Ohne diese Karte kann der Wagen nicht getankt werden.

Bei der Tankstelle können auch die Reifen aufgepumpt werden, insofern dies notwendig ist.

Da es sich bei den Fahrzeugen oft um Prototypen oder Vor-Serienfahrzeuge handelt, müssen diese mit Vorsicht behandelt werden.

Um das Fotografieren der Fahrzeuge während sie auf dem Gelände stehen zu vermeiden, hat das Entwicklungszentrum Weissach ein speziell Blick geschütztes Parkhaus.

Um in diesem Parkhaus parken zu können, muss eine Genehmigung beantragt werden. Nachdem diese genehmigt ist, wird der RFID-Transponder des Fahrzeugs an der Parkhauspforte freigeschaltet. Sobald das Fahrzeug nicht mehr von der Abteilung benötigt wird, oder es zur Verschrottung freigegeben wird, wird der RFID-Transponder wieder gesperrt und das Fahrzeug muss aus dem Parkhaus entfernt werden.



Abb. 3.10: Das Prototypenparkhaus von Innen

[4] Quelle: <https://newsroom.porsche.com/de/2021/unternehmen/porsche-entwicklungszentrum-weissach-christophorus-397-23144.html>

Da in diesem Parkhaus und seinem Zwillings in Hemmingen nur insgesamt 375 Plätze zur Verfügung stehen und diese nicht mehr ausreichen, wurde in Weissach am Anfang 2021 ein neues Parkhaus in Betrieb genommen.

In diesem sind 1147 Plätze, sowie 400 Ladesäulen vorhanden um alle Fahrzeuge die auf dem Gelände sind unterstellen zu können [1].



## Literaturverzeichnis

- [1] Entwicklungszentrum Weissach: Die Herzkammer von Porsche. <https://newsroom.porsche.com/de/2021/entwicklungszentrum-weissach-christophorus-397-23144.html>, 18.01.2021. URL: <https://newsroom.porsche.com/de/2021/unternehmen/porsche-entwicklungszentrum-weissach-christophorus-397-23144.html>.

