

# Schlussbericht - Teil I

## KI-IoT

**Forschungsprojekt:** KI-IOT

**Laufzeit:** 01.07.2021 – 30.06.2024 (3 Jahre)

**Förderkennzeichen:** 16ME0353

**Firma:** AI Digi+ Solutions GmbH

*This work has been developed in the ZuSE project KI-IoT.*

*KI-IoT is funded by the German ministry of education and research (BMBF)*

*(July 2021 – June 2024/November 2024)*

Ziel der Entwicklung ist der Aufbau einer flexiblen Entwicklungsplattform für eingebettete Prozessoren und Mikrocontroller für mobile Geräte, IoT, Embedded-Computing und Regelaufgaben mit Sensor/Aktor-Interaktion. Dabei liegen die Schwerpunkte gleichermaßen auf Abgrenzung zu frei verfügbaren Kernen auf der Vollständigkeit des Baukastens, der Verfügbarkeit technologienaher Komponenten und der Integration einer KI-Komponente. In diesem Forschungsprojekt wurden Demonstratoren für den KI basierte Optimierung von Servomotoren auf Basis des Roboters widmen. Hierbei sollen Mithilfe von Simulation und Softwaretools Optimierungen auf Basis KI mit dem Tools MAGICLOOP entwickelt werden. Hierbei stehe die holistische Anwendung des Munich Agile Konzepts für Systems Engineering im Vordergrund. Diese Ansätze sollen dabei helfen KI als integralen Bestandteil der Produktentwicklung zu integrieren. Der Baukasten umfasst eine Vielzahl digitaler Komponenten des Prozessorsystems, bei denen nach Möglichkeit auf bereits verfügbare Open-Source-Projekte (u.a. OpenAI, Tensorflow, Yolo8) gesetzt wird.

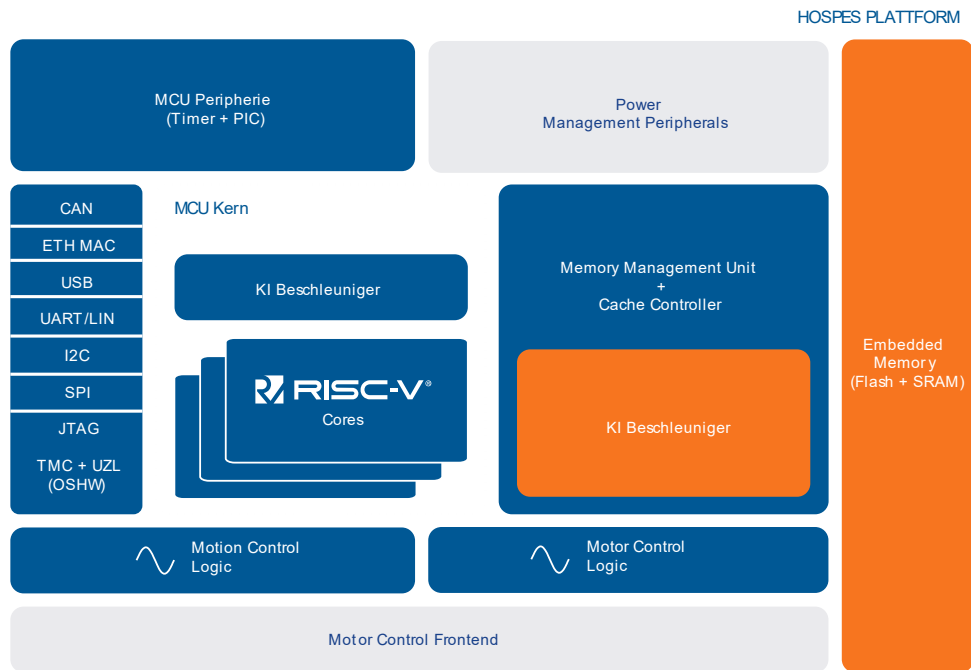
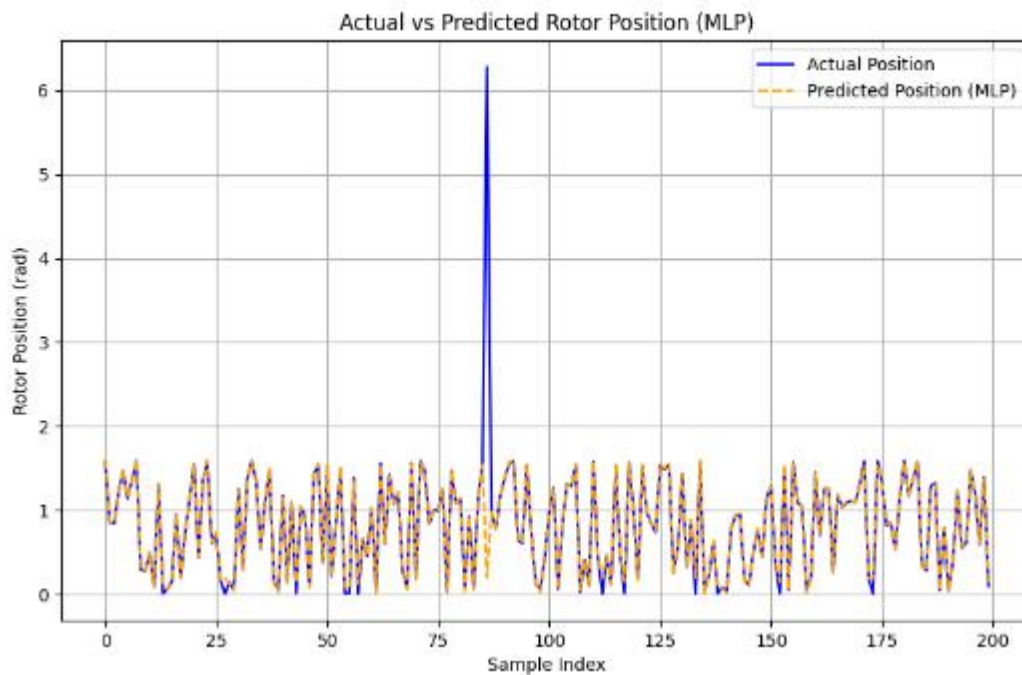


Abbildung 1: Übersicht Baukasten

Prozessorkerne, sowie vielzählige Implementationen unterschiedlicher Interfacestandards sind bereits als Open Source verfügbar und werden durch die Partner für die Plattform erprobt und gegebenenfalls für die Integration modifiziert oder erweitert. Dazu werden zunächst die wichtigsten Projekte gesichtet und deren Eignung für die Plattform geprüft. Dabei wurde für die Automatisierung der KI-Prozesse eine Plattform entwickelt. Mit der Plattform ist es möglich DataSets automatisiert einzulesen und diese zu Trainieren. Es wurden speziell für die KI relevanten Schritte DataSets Generierung, DataSets Training und DataSets Validierung entsprechende Plattformen fertig gestellt. Mit entsprechenden DataSets, DataSets Training, Validierung wurde eine Umgebungsplattform geschaffen, womit die Entwicklung von KI-Pipelines zielgerichtet unterstützt wird bzw. als Demonstrator aufbaut. Des Weiteren wurde damit das Portfolio hinsichtlich Digitalisierung und Nutzung von KI-Pipelines übergreifend ausgebaut. Gleichzeitig wurden in diesem Projekt die Kernkompetenzen im Bereich KI ausgebaut. Aus unserer Sicht sind die erwarteten wirtschaftlichen Verwertungen der angestrebten Projektergebnisse in mehrerlei Hinsicht nutzenstiftend. Durch die offene Nutzung und den Einsatz von KI-ServoMot, MAGIC können neue KI-Plattformen durchgängig genutzt, miteinander und dem AICab Roboter intelligent vernetzt.

# Schlussbericht-Teil II

## KI-IoT



**Forschungsprojekt:** KI-IOT

**Laufzeit:** 01.07.2021 – 30.06.2024 (3 Jahre)

**Förderkennzeichen:** 16ME0353

**Firma:** AI Digi+ Solutions GmbH

*This work has been developed in the ZuSE project KI-IoT.*

*KI-IoT is funded by the German ministry of education and research (BMBF)  
(July 2021 – June 2024/November 2024)*

## 1 Einführung.

Die in diesem Abschlussbericht erläuterten Themen und Inhalte sollen die Ergebnisse des Projektes KI-IOT darstellen. Ziel des Vorhabens KI-IOT ist die Entwicklung und quelloffene Bereitstellung einer vollständigen modularen KI-ServoMot-Plattform für den Entwurf von IoT Architekturen. Mit dem Open-Source-Ansatz für alle digitalen Kernkomponenten sowie Lizenzmodellen für analoge und proprietäre Funktionsblöcke sollen die Einstiegshürden - insbesondere für KMUs - für die Ableitung anwendungsspezifisch angepasster IOT Devices für vielfältige Anwendungen zukünftig deutlich reduziert werden. AIDigi+ bringt seine Erfahrung in Simulation und Visualisierung in dieses Projekt ein und ist als Projektpartner verantwortlich für Aspekte der Systemsimulation

## 2 Aufzählung der wichtigsten wissenschaftlich-technischen Ergebnisse und anderer wesentlicher Ereignisse.

Ziel der Entwicklung ist der Aufbau einer flexiblen Entwicklungsplattform für eingebettete Prozessoren und Mikrocontroller für mobile Geräte, IoT, Embedded-Computing und Regelaufgaben mit Sensor/Aktor-Interaktion. Dabei liegen die Schwerpunkte gleichermaßen auf Abgrenzung zu frei verfügbaren Kernen auf der Vollständigkeit des Baukastens, der Verfügbarkeit technologienaher Komponenten und der Integration einer KI-Komponente.

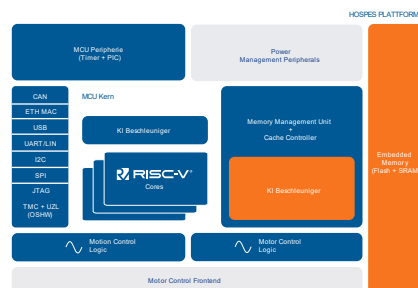


Abbildung 1: Übersicht KI-IOT Baukasten

AIDigi+ Solutions GmbH hat sich im Rahmen des Forschungsprojektes Anwendungsfällen für die KI-basierte Optimierung von Servomotoren auf Basis des Forschungsroboters gewidmet. Es wurde ein Forschungsroboter in Kleinstserie/Einzelfertigung benutzt, der sehr offen und flexibel ist und viele Möglichkeiten hinsichtlich Änderungen ermöglicht. Dieser wurde mit Hilfe von Simulation, Softwaretools und Optimierungen auf Basis von KI mit den Tools MAGICLOOP entwickelt. Während dieser Phase wurden unterschiedliche Wege und Möglichkeiten, die die Anforderungen, Funktionen, Architektur und die Realisierung des Anwendungsfalls ermöglichen, genauer betrachtet und auf Basis von MAGICLOOP in Verbindung mit KI-ServoMot-Plattform umgesetzt. Hierbei stand die holistische Anwendung des Munich Agile Konzepts für Systems Engineering im Vordergrund. Diese Ansätze haben dabei geholfen KI als integralen Bestandteil in die Produktentwicklung zu integrieren. Der Baukasten umfasst eine Vielzahl digitaler Komponenten des Prozessorsystems, die nach Möglichkeit auf bereits verfügbare Open-Source-Projekte innerhalb von AIDigi+ aufgesetzt werden können. Die Prozessorkerne, sowie vielzählige Implementierungen unterschiedlicher Interfacestandards sind bereits als Open Source verfügbar und werden durch AI Digi+

Solutions für die Plattform erprobt und gegebenenfalls für die Integration modifiziert oder erweitert. Dazu wurden zunächst die wichtigsten Projekte gesichtet und deren Eignung für die Plattform geprüft.

Bei der Durchführung der Arbeitspakete standen hierbei unterschiedliche Use Cases im Vordergrund. Zum einen geht es darum aus der realen und der digitalen Umgebung mit KI-ServoMot-Plattform des Forschungsroboters und dessen BLDC Motors Daten für den Einsatz im Bereich KI zu generieren. Hierbei werden die Daten in drei Schritten weiterbearbeitet. Im Schritt eins werden DataSets generiert. Ein wesentliches Augenmerk bei der Integration der verschiedenen Blöcke zu einem Gesamt-IC liegt dabei auf den hierzu erforderlichen leistungs- und zuverlässigkeitsrelevanten Blöcken. Mithilfe des Munich Agile Concept Ansatzes kurz MAGIC wurden die unterschiedlichen Phasen für die APs 5 und 6 angewandt. Des Weiteren werden mit modernen KI-Ansätzen eine Vorhersage von unterschiedlichem Parameter des BLDC-Motors wie Strom, Spannung und Rotorposition hervorgesagt bzw. mit Hilfe von KI ermittelt. Als Ergebnis wurde KI-ServoMot-Plattform als Prototypen entwickelt.

## 2.1 Ergebnisse aus den Teilarbeitspaketen:

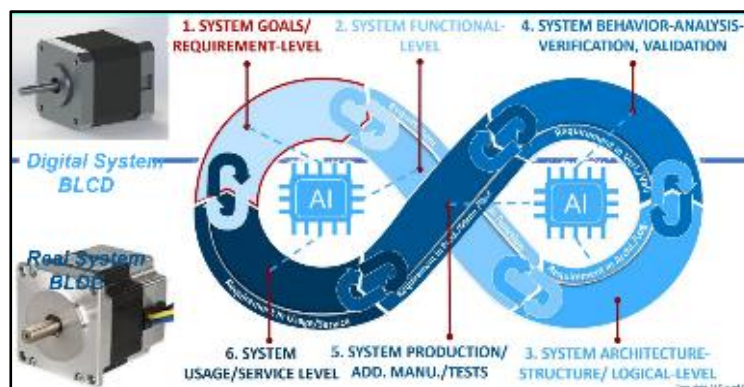


Abbildung 2: MAGIC Ansatz

Um die im ersten Abschnitt genannten Herausforderungen zu bewältigen, wurden der Ansatz MAGIC angewandt, der auf der Form des DevOps-Endlosschleifenmodells basiert.

Im Forschungsprojekt wurden folgende vier Ebenen des MAGIC Modells angewandt:

1. Systemziele / Anforderungsebene
2. funktionale Ebene des Systems
3. Systemarchitektur - Struktur / logische Ebene
4. Systemverhalten-Analyse-Verifizierung / Validierungsebene

Die erste Ebene beinhaltet die Anforderungsdefinition, die die Bedürfnisse und Wünsche der Kunden spezifiziert. MAGIC bietet die Möglichkeit, verschiedene Arten von Anforderungen zu verwenden, wie z. B. Benutzer- und Funktionsanforderungen. Die funktionale und damit zweite Ebene des Systems enthält die Gesamtstruktur und das Verhalten des Systems. Da das System durch die Teilfunktionen des Systems und die Wechselwirkungen zwischen den einzelnen Komponenten definiert wird, können



in diesem Schritt die Funktionen für das gesamte System ermittelt werden. In der dritten Ebene wird beschrieben, wie die zuvor definierten Funktionen umgesetzt werden. Dies geschieht, indem gezeigt wird, welche Komponente in welcher Weise mit einer anderen Komponente verbunden ist, um das logische Verhalten des Systems zu erfüllen. Dazu gehört auch die Interaktion zwischen den modellierten mechanischen, elektronischen und Software-Komponenten. Auf der vierten Ebene geht es um die Validierung, ob das Projekt mit den Spezifikationen übereinstimmt. Dies beinhaltet die Überprüfung, ob der beabsichtigte Zweck erfüllt wird und ob die geforderte Qualität gegeben ist. Der Begriff "Validierung" wird wie folgt definiert: "Die Validierung ist eine Reihe von Aktivitäten, die sicherstellen, dass ein System in der Lage ist, seine beabsichtigte Verwendung und seine Ziele in der vorgesehenen Betriebsumgebung zu erreichen".

## **2.2 MAGIG-IMPLEMENTIERUNG EINER SENSORLOSEN BLDC-MOTORSTEUERUNGSSIMULATION**

In diesem Abschnitt wird der im Munich Agile Concept (MAGIC) beschriebene Ansatz auf die Simulation eines BLDC-Motors mit KI-ServoMot-Plattform angewendet und umgesetzt. Basierend auf MAGIC wurden die ersten vier Prozessschritte, nämlich die Anforderungs-, Funktions-, Logik- und Verhaltensebene, in der Simulation der sensorlosen Drehzahlregelung eines BLDC-Motors implementiert. In den letzten Jahren wurden mehrere Modellierungssprachen und -methoden vorgeschlagen, um multidisziplinäre und komplexe technische Systeme modellieren zu können. Die beliebteste Modellierungssprache, die auch in dieser Arbeit verwendet wird, ist SysML, eine weithin akzeptierte UML-basierte, objektorientierte grafische Software-Modellierungssprache. Durch die Verwendung von SysML können Inkonsistenzen und Redundanzen reduziert werden, und sie ist auch die Grundlage für eine klare Kommunikation und nachhaltige Dokumentation. Um alle Aspekte eines Systems in möglichst standardkonformen SysML-Modellen und -Diagrammen zu definieren, zu verfolgen und zu visualisieren, wird der Papyrus Systems Modeler eingesetzt. Er bietet intelligente, leistungsfähige und intuitive Werkzeuge.

## **3 Beschreibung AP2**

In diesem Arbeitsschritt wurden für einen ausgewählten BLDC-Motor die Anforderungen definiert und diese in unterschiedlichen Kategorien integriert. Untenstehend kann man die Anforderungen für den speziellen ausgewählten UC sehen. Auf Basis der Anforderungen kann man, wie auch in dem MAGIC Ansatz dargestellt, KI nutzen, um daraus für die spätere Simulation eine Architektur mit entsprechender Funktions- und Simulationsumgebung auf Basis von Python zu generieren. Hierzu wurde ein Long Language Models (LLM)-Ansatz auf Basis OpenAI Lösung ausgewählt, um die Anforderungen einzuspeisen, das LLM-System generiert dann auf Basis von KI alle relevanten Simulationsparameter. Die Liste unten zeigt die definierten Anforderungen. Die Integration erfolgte auf Basis von KI-ServoMot-Plattform. Der neuartige Ansatz war dabei, dass man mit Hilfe von LLM alle anderen Schritte des MAGIC Ansatzes automatisch generieren kann. Hiermit wird eine enorme Verbesserung und Änderungsgeschwindigkeit erreichen.

1. Rückmeldung zur Geschwindigkeit:



1. Der BLDC muss mit 3-Phasen-Hallsensoren installiert werden
  2. Der BLDC muss mit einem Quadratursensor installiert werden
  3. Code kann generiert werden
  4. Die Simulation kann gestartet werden
2. Sicherung der Werte:
1. Die Drehzahlvorgabe muss gespeichert werden
  2. Der Offset des 3-Phasen-Stroms muss gespeichert werden
  3. Die Simulation muss aktiviert und deaktiviert werden
  4. Die Signale müssen debugged werden
3. Seriell Empfangen
1. Referenzgeschwindigkeit muss erzeugt werden
  2. Die Daten müssen mit dem gewünschten Typ Float übertragen werden
  3. Die Daten müssen gespeichert werden
  4. Die Geschwindigkeitsanforderung kann in Zahlen dargestellt werden.
4. Stromregelung
1. Die von den Sensoren gemessenen Geschwindigkeitsdaten müssen eingespeist werden
  2. Sensor-Treiber-Blöcke
  3. Sensor muss abgeleitet werden
  4. Alle Geschwindigkeits- und Stromdaten müssen als Eingangs-Skalierung in das PU-System transformiert werden
  5. Die mechanische Position muss in eine elektrische Position umgewandelt werden
5. Umrichter und Motor
1. Das Drehmomentsignal muss geladen werden.
  2. PWM muss erzeugt werden.
  3. DC muss in 3 Phasen umgewandelt werden.
  4. Der Phasenstrom muss gemessen werden.
  5. Der Rotorwinkel Theta muss gemessen werden.

Die Parameteranforderungen für Drehmoment, Motordrehzahl und Motortemperatur für die Servomotoren wurden auf Basis von MAGICLOOP für den Servomotor-Demonstrator definiert. Des Weiteren wurden die unterschiedlichen Servomotoren Technologien und deren Anwendungsfälle für Servomotoren mit MAGICLOOP definiert und im Rahmen einer Demonstrator Analyse untersucht. Somit wurden die Vor- und Nachteile der unterschiedlichen automatischen KI-Systeme für Drehmoment, Drehzahl und Temperatur des Demonstrators analysiert. Die Anforderungen bezüglich des systemtechnischen Aufbaus des Demonstrators für die Servomotoren-Steuerung wurden analysiert. Darüber hinaus wurden die Anforderungen an den systemtechnischen Aufbau sowohl des Demonstrators als auch einer umfassenden



KI-Themen (IT-Infrastruktur, Systemarchitektur, Modularität, Skalierbarkeit, Echtzeitfähigkeit) zur Simulation von Servomotoren analysiert.

Die Algorithmen aus dem Bereich KI (Machine Learning) wurden ebenfalls analysiert, diese werden in den nächsten Kapiteln kurz dargestellt.

Die Anforderungen an die modelltechnischen Eigenschaften hinsichtlich Datenformate (.csv, SysML), Modellstruktur (Parameter, Drehmoment, Lasten, Drehzahl, Module usw.) sowie die Berücksichtigung von Daten eines Prüfstands für die Simulation auf Basis eines BLDC-Motors wurden analysiert. Die Anforderungen an die BLDC-Motoren und die dazugehörigen Prognosen in einer KI-Umgebung wurden analysiert. In diesem Rahmen wurden unterschiedliche Systeme für Machine Learning für KI bewertet und untersucht. Die Anforderungen an den Aufbau der Simulationsumgebung mit den erforderlichen KI-Systemen wie Regression, Klassifikation, Clustering, Dimensionsreduktion, Ensemble Methods, Neural Nets, Deep Learning, Transfer Learning und Reinforcement Learning wurden analysiert. Die Anforderungen an die KI-spezifischen Algorithmen, Tools und Systemeigenschaften hinsichtlich Methoden und Ansätzen (z. B. TensorFlow, YOLO, OpenCV, OpenAI) wurden analysiert. In diesem Kontext wurde definiert, welche Parameter für eine Verifizierung der späteren Ergebnisse beim Trainieren der Daten geeignet sind. Die verfügbaren DataSets für die Durchführung der Vorhersage der BLDC-Parameter wurden analysiert, und die Anforderungen für die eigene Generierung von DataSets von Servomotoren wurden definiert. Im Rahmen der Analyse der DataSets wurden die unterschiedlichen Einstellungen der Aufnahmen für DataSets betrachtet.

#### **4 Arbeitspaket 5 und 6: Digitalentwurf Firmware und Inbetriebnahme Werkzeuge**

Es wurde eine Analyse der Vor- und Nachteile der unterschiedlichen automatischen KI basierten Algorithmen für Drehmoment, Drehzahl und Temperatur des Demonstrators untersucht. Dabei wurden die Anforderungen bezüglich des systemtechnischen Aufbaus sowohl für den Demonstrator als auch für die Servomotoren-Steuerung ganzheitlich abgebildet und in einer Simulationsumgebung integriert. Es wurden unterschiedliche Ansätze wie Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN) und Current Derivative Analysis identifiziert. Hierzu wurden mit modernen LLM-Tools von OpenAI unterschiedliche Python-Codes generiert und getestet. Das bedeutet, dass der User einfach die Liste der Anforderungen in das LLM-Model als „Prompt“ eingibt und das LLM-Model generiert automatisch die erforderlichen Python-Codes, inklusive der Ergebnisse. Das ist nun ein wahrer Paradigmenwechsel zu den alten Vorgehensweisen. Mithilfe von LLM kann man nun binnen kurzer Zeit schnelle unterschiedliche Lösungen und Ausprägungen der Architektur generieren und testen. Das einzige Thema was zukünftig wichtig sein wird, ist eine besonders saubere und aussagekräftige Definition der Anforderungen und die genaue Priorisierung. Es wurde ebenfalls die Analyse der Anforderungen an die modelltechnischen Eigenschaften hinsichtlich Datenformaten in einer SysML-basierten Struktur auf Basis von Papyrus SysML modelliert. Hiermit wurde eine Modellstruktur (Parameter, Drehmoment, Lasten, Drehzahl, Module) geschaffen, durch welche unter Berücksichtigung von Daten einer virtuellen Python-basierten Umgebung (virtueller Prüfstand) eine Simulation auf Basis von KI hinsichtlich Nutzung

ermöglicht wurde. Darüber hinaus wurden die Anforderungen an die KI-spezifischen Algorithmen, Tools und Systemeigenschaften hinsichtlich Methoden und Ansätzen (TensorFlow, Yolo, OpenCV, OpenAI) definiert. Für die optische Erkennung der BLDC-Temperaturerkennung wurden Ansätze von Yolo 8 Algorithmen ausgewählt und angewendet. Damit wurden die generierten Datensätze trainiert und validiert. Der MLP-Ansatz ist aus den folgenden Gründen ein vielversprechender Kandidat für den sensorlosen Betrieb von BLDC-Motoren:

- Die Einfachheit und Berechnungseffizienz.
- Wie in den Diagrammen dargestellt, genaue und stabile Vorhersagen.
- Weniger Rauschen und niedriger MSE = effizient und zuverlässig für Echtzeitsysteme.

Wenn Ihr System eine Echtzeitreaktion erfordert und mit begrenzter Hardware arbeitet, scheint MLP laut den Diagrammen die beste Option zu sein. Wenn Ihre Anwendung komplexere Dynamiken oder langfristige Abhängigkeiten einbeziehen muss, sollten Sie alternativ die LSTM- oder Hybrid-Methoden in Betracht ziehen.

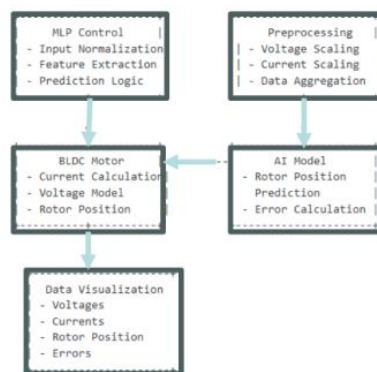


Abbildung 3: Blockdarstellung BLDC und der MLP-Ansatz

Dabei wurde in Python ein BLDC-Motor konfiguriert und dieser für die Berechnungen und Simulation zu Grunde gelegt. Die wichtigsten Parameter können unten aus dem Bild entnommen werden. Diese sind der Rotorwinkel, Drehzahl und Antriebsmoment.

```

# Konfiguration des Motors
class BLDCMotor:
    def __init__(self, resistance, inductance, flux_linkage, poles):
        self.resistance = resistance
        self.inductance = inductance
        self.flux_linkage = flux_linkage
        self.poles = poles
        self.current_a = 0.0
        self.current_b = 0.0
        self.current_c = 0.0
        self.theta = 0.0 # Rotorwinkel
        self.speed = 0.0 # Drehzahl
        self.torque = 0.1 # Antriebsmoment
  
```

Abbildung 4: Konfiguration eines einfachen BLDC-Motors

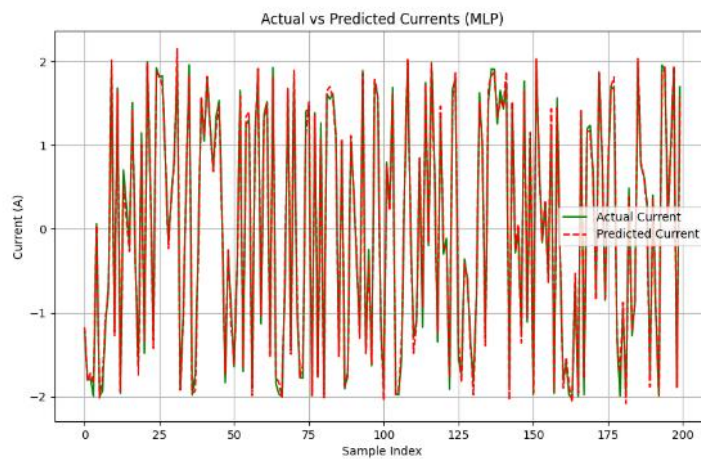
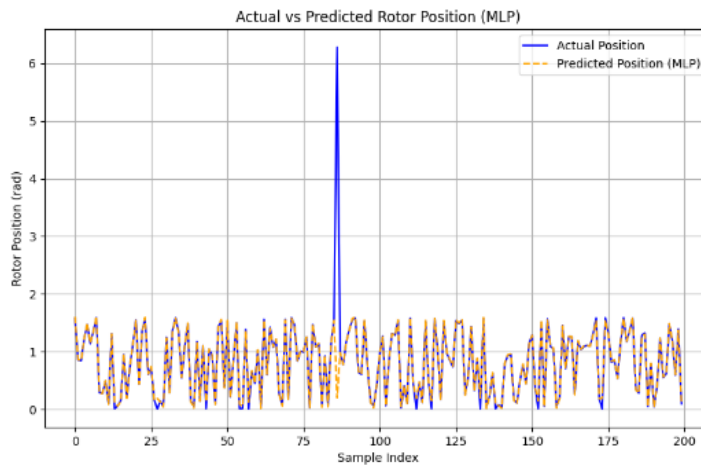
```
X = np.vstack([voltages, currents]).T
y = actual_positions

# Splitting Data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Method 1: Multi-Layer Perceptron (MLP)
mlp = MLPRegressor(hidden_layer_sizes=(50, 50), max_iter=500, random_state=42)
mlp.fit(X_train, y_train)
mlp_pred = mlp.predict(X_test)
mlp_mse = mean_squared_error(y_test, mlp_pred)

# Extending Predictions to Currents and Voltages
predicted_currents = X_test[:, 1] + np.random.normal(0, 0.1, len(X_test[:, 1]))
predicted_voltages = X_test[:, 0] + np.random.normal(0, 0.1, len(X_test[:, 0]))
```

Abbildung 5: Integration MLP-Steuerungsklasse



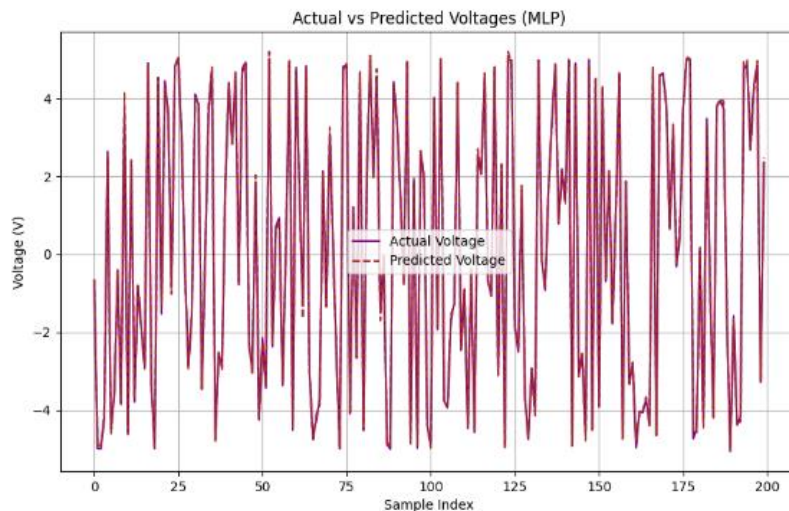


Abbildung 6: Simulation von unterschiedlichem Parameter des BLDC-Motors auf Basis MLP (ServoMot Pipeline)

Die Ergebnisse des MLP-basierten Ansatzes kann man aus den Graphen in Abbildung 6 entnehmen. Folgende Ergebnisse kann man daraus nun formulieren:

**Stabilität der Vorhersage:** In der Darstellung von tatsächlicher und vorhergesagter Rotorposition folgen die MLP-Vorhersagen eng den tatsächlichen Rotorpositionen mit minimalen Schwankungen. Diese Stabilität deutet darauf hin, dass der MLP die Beziehung zwischen Spannungen, Strömen und Rotorposition für den gegebenen Datensatz effektiv erfasst. **Geringes Rauschen in den Vorhersagen:** Die Strom- und Spannungsvorhersageplots zeigen, dass der MLP im Vergleich zu anderen Methoden konsistente und gleichmäßige Vorhersagen mit minimalem Rauschen erzeugt. Dies ist vorteilhaft für Motorsteuerungssysteme, die gleichmäßige und zuverlässige Eingangssignale benötigen.

**Niedriger mittlerer quadratischer Fehler (MSE):** Der MSE für MLP ist niedrig, wie die gedruckten Ergebnisse zeigen. Ein niedriger MSE bedeutet eine bessere Gesamtvorhersagegenauigkeit.

#### 4.1 Anforderungsniveau

Die erste Ebene enthält die technischen Anforderungen, die in Form von Muss (M), Soll (S) oder Kann (C) definiert sind. Die Anforderungsebene besteht aus sechs Teilen, nämlich Drehzahlrückführung, Speicherung, serieller Empfang, Drehzahlregelung, Stromregelung und Motorteil, wie in Abbildung 7 dargestellt. Da die Regelungstheorie verwendet wird, muss die Drehzahl für die Fehlerberechnung rückgeführt werden. Außerdem sollten Module für die Codegenerierung und den Simulationsstart einbezogen werden. Der Simulationsprozess erzeugt eine große Menge an Daten, die eine große Rolle bei der Stabilitäts- und Leistungsanalyse des Systems spielen. Diese Daten werden auch im Schleifensystem wiederholt aufgerufen. Daher sind die Anforderungen an die Datenspeicherung ein wesentlicher Bestandteil. Das System enthält auch die Anforderungen für die Bereiche serieller Empfang, Drehzahlregelung und Stromregelung. Im seriellen Empfangsteil muss die Referenzgeschwindigkeit erzeugt und mit dem gewünschten Float-Typ übertragen werden. Die Kernsteuerungsschicht des Systems ist in eine Geschwindigkeitssteuerungsschicht für

den äußeren Regelkreis und eine Stromsteuerungsschicht für den inneren Regelkreis unterteilt. In der äußeren Regelungsschicht soll der PI-Regler für die Drehzahlregelung verwendet werden. Hier muss die Drehzahlabweichung berechnet und als Eingang für den Regler verwendet werden. Im Stromregler muss RL implementiert werden. Daher müssen in diesem Teil die Spannungen und Ströme in einem bestimmten  $\alpha$ - $\beta$ -Frame gemessen und in ein Per-Unit-System (PU) transformiert werden. Die Beobachtungen, Belohnungen, Aktionen und Agenten müssen explizit sein. Der letzte Teil sind die Anforderungen an Wechselrichter und Motor. Diese Anforderungen enthalten die Spezifikationen des BLDC wie maximale Geschwindigkeit, Nennspannung und Nennstrom, PWM und Parameter, die gemessen werden müssen.

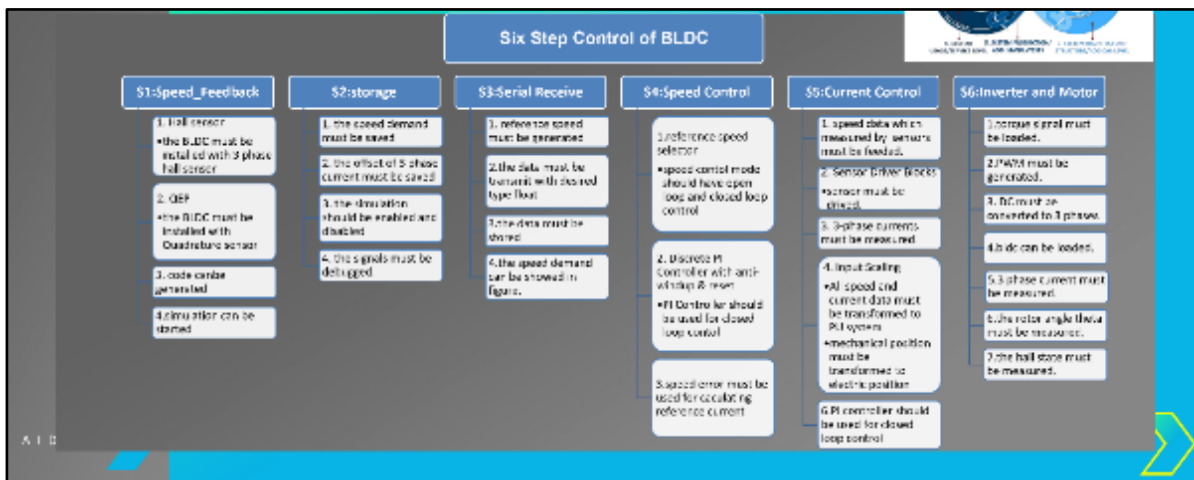


Abbildung 7: Anforderungen and das BLDC

## 4.2 Funktionale Ebene

Wie im MAGIC-Abschnitt beschrieben, entsprechen sich die funktionale Ebene und die Anforderungsebene gegenseitig. Um die Anforderungen der verschiedenen Module zu erfüllen, werden die Funktionen der entsprechenden Module auf dieser Ebene definiert. Es ist zu beachten, dass diese Funktionen zwar nicht direkt eine feste Lösung bieten, aber den Anwendungsbereich einschränken. Im ersten Teil muss das geschlossene System beispielsweise eine Geschwindigkeitsrückkopplungsfunktion bereitstellen. Im zweiten Teil wird die Speicherfunktion des Systems sowohl zur Speicherung der Geschwindigkeit als auch des Stromsignals verwendet. Der serielle Empfang wird zur Erzeugung des Drehzahlsollwerts verwendet. Der Drehzahlregelungsteil wird zur Berechnung des Drehzahlfehlers und zur Erzeugung der dreiphasigen Ströme als Eingang für den Stromregelungsteil verwendet. Die Funktion der Stromregelung besteht darin, den Fehlerwert zu analysieren und zu verarbeiten, der durch die ständige Interaktion zwischen dem Agenten und der Umgebung kontinuierlich reduziert wird, und schließlich das gewünschte Spannungsregelungssignal auszugeben. Der Wechselrichter wandelt das PWM-Tastverhältnis in eine Dreiphasenspannung für den Motorantrieb um. Die Funktion des Motors besteht darin, die gewünschte Drehzahl und das gewünschte Drehmoment zu erzeugen, der gemessene dreiphasige Strom wird wiederum als Signalführung verwendet, was zu einem geschlossenen Regelkreis führt.

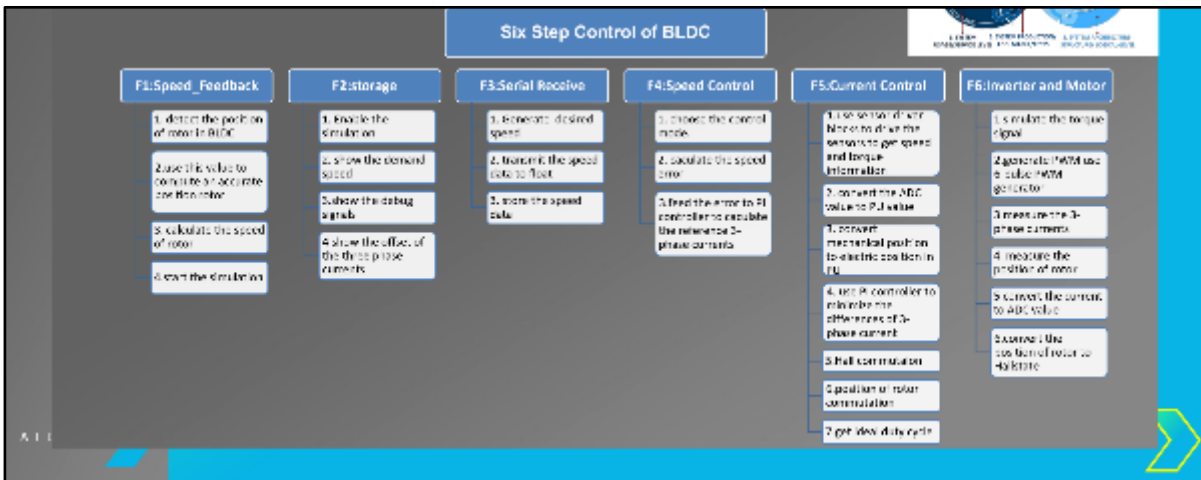


Abbildung 8: Integration der Funktionen an das BLDC

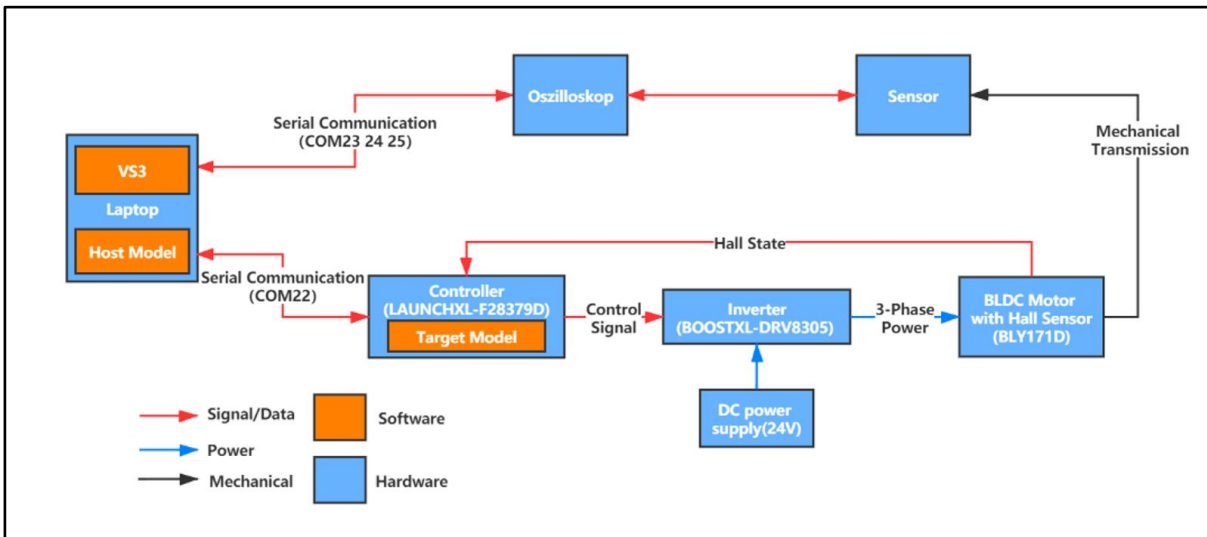


Abbildung 9: Integration der Architektur für den Versuchsaufbau BLDC

### 4.3 Aufbau von weiteren Use Case für eine Thermo-Kamera

Für die Generierung von DataSets wurde für diesen Use Case eine Kamera der Firma Seek ausgewählt. Das Seek Thermal System umfasst einen Kameraaufsatz für ein Android- oder iOS-Gerät und eine App, die anzeigt, was die Kamera aufnimmt. Die Firma Seek bietet zwei Smartphone-Kameraaufsätze an, die Thermal-Kamera und die Thermal Xtra Range (XR), die beide sowohl für iOS- als auch für Android-Geräte erhältlich sind. Der Unterschied zwischen den beiden ist, dass die XR-Kamera Objekte erkennen kann, die doppelt so weit entfernt sind, wie bei der regulären Kamera, bei gleicher Klarheit. Außerdem hat die XR eine manuelle Fokusfunktion. Für das iPhone wird ein Lightning-Anschluss und iOS 7 oder 8 benötigt, für Android-Geräte lediglich Jelly Bean 4.3.1 oder höher. Für die Verwendung im Projekt wurde das Originalmodell der Kamera auf einem Android-Gerät und die App getestet. Die Original-Wärmebildkamera von Seek ist klein, etwa 1,6 Zoll lang, 0,8 Zoll hoch und 0,6 Zoll dick, und wiegt gerade einmal 0,5 Unzen (14 Gramm). Die Chalkogenid-Linse ist nach links versetzt, da sich auf der rechten Seite der Anschluss befindet, der an den Micro-USB-Port des Smartphones gesteckt wird. Sobald die Seek Thermal App installiert ist,

kann man die Kamera ganz einfach an das Smartphone anschließen, die App wird anschließend automatisch gestartet, nachdem für die Kamera die notwendigen Berechtigungen erteilt wurden.

### Erkennen von Temperaturen

Die Seek-Kamera funktioniert sowohl bei Tageslicht als auch in der Nacht, da sie kein sichtbares Licht benötigt. Stattdessen nimmt sie Infrarotwellen auf, die jeder Gegenstand aussendet. Infrarotwellen sind für das menschliche Auge nicht sichtbar, also wandelt Seek diese Wellen in ein visuelles Bild um, das Farben verwendet. Diese IR-Wellen können verwendet werden, um Temperaturschwankungen in der Umgebung festzustellen, dem entsprechend werden in dem dargestellten Bild Farben den verschiedenen Temperaturen in der Umgebung zugewiesen. Dadurch können Temperaturspitzen effektiv erkannt werden. Seek verwendet einen Sensorchip und einen Software-Algorithmus, um diese Wärmebilder zu erstellen. Bei jeder Aufnahme werden 32.000 Pixel erfasst, und jeder Pixel nimmt eine andere Temperaturmessung vor, um das gesamte Foto oder Video zu erstellen. Die Kamera kann Temperaturen von -40 Grad Fahrenheit bis 626 Grad Fahrenheit (-40 Grad Celsius bis 330 Grad Celsius) erfassen.

Die Fotos und Videos, die die Seek-Kamera aufnimmt, sind mit 206 x 156 Pixeln nicht hochauflösend. Der Detailgrad kann erhöht werden, wenn die Kamera vor dem Fotografieren einige Sekunden lang ruhig gehalten wird, damit sich das Objektiv kalibrieren kann. Die Kamera kann sowohl dazu verwendet werden, ein Wärmebild in Echtzeit auf dem Bildschirm des Smartphones zu betrachten, als auch dazu Fotos und Videos aufzunehmen, um diese später auswerten zu können. Die Kamera verwendet einen automatischen Schwarzkörperverschluss, um das Objektiv zu kalibrieren, und macht alle paar Sekunden ein leises Klickgeräusch. Dieser Verschluss unterscheidet sich damit von dem Bildschirmverschluss, mit dem ein Foto in der App aufgenommen wird. Dieser Black Body Shutter kann vom User tatsächlich überhaupt nicht gesteuert werden; er ist so konzipiert, dass er von selbst läuft. Die Abbildung unten zeigt die Kameraeigenschaften.



Abbildung 10: Eigenschaften der Seek Kamera

### Generierung von DataSets und Labing

Qualitativ hochwertige Daten sind das Lebenselixier großartiger Computer-Vision-Anwendungen. Jedoch besteht die Gefahr, dass für die Erzeugung von Anwendungen nicht genügend Daten vorhanden sind. Bei unserem Use Case haben wir festgestellt, dass wir die Daten alle selbst generieren und labeln mussten. Dieser Teil der

Forschungsarbeit war mit sehr viel Aufwand und Arbeit verbunden. Fast 75% der Arbeit lag darin die richtigen DataSets zu generieren und diese dann entsprechend zu labeln. In diesem Abschnitt werfen wir daher einen genaueren Blick auf die besten Praktiken, die zur Erstellung von eigenen benutzerdefinierten Datensätze dienen.

Die Erstellung benutzerdefinierter Datensätze erfolgt in sechs wichtigen Schritten: Im ersten Schritt wird eine Erfassungsmethode definiert. Dies kann bedeuten einen eigenen Datensatz mithilfe interner Ressourcen zu erstellen. Für die Datenerfassung wurden in unserem Fall mit Hilfe der Seek Kamera Aufnahmen erstellt. Dabei wurde der BLDC-Motor im Betrieb abgefilmt, dabei konnte man beobachten, wie die Temperatur des Motors gestiegen ist. In einem späteren Schritt wurden die Prüfstände mit Hilfe von Computer Vision aufgerüstet, damit der BLDC-Motor im Betrieb optisch überwacht werden kann. Deswegen haben wir auch die Datenerfassung automatisiert. Die Abbildung unten zeigt den Aufbau des Prüfstands auf Basis KI-ServoMot-Plattform.

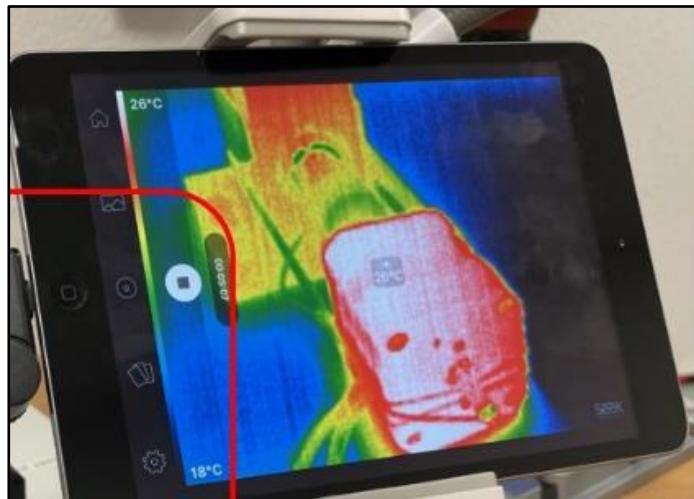


Abbildung 11: Aufbau der Tests

Es gibt Data Scraping Tools, die den Entwickler bei der Datenerfassung unterstützen. Bei der manuellen Datensatzerfassung sind Menschen beteiligt, die die Daten entsprechend Ihren Anforderungen und Geschäftsregeln suchen und sammeln. In diesem Fall wurden wie schon oben erwähnt eigene Geräte, wie Kameras und Sensoren (z. B. Temperatur Sensor), verwendet. Während der Auswahl der geeigneten Erfassungsmethode, sollte auch ein geeignetes Datenkommentar-Tool ermittelt werden. Die Wahl des Tools wird sich auf den Erfolg, aber auch auf die Effizienz der Arbeit im Forschungsprojekt auswirken und einige weitere Optionen für die Datenspeicherung, das Tooling und den Arbeitsablauf bestimmen. In diesem Forschungsprojekt wurden sehr gute Erfahrungen mit dem Tool magicloop gemacht. Magicloop hat im Rahmen des Projektes dabei geholfen die Erzeugung der DataSets sehr strukturiert vorzunehmen. Danach wurden die Daten in mehreren Stufen gesammelt. In der ersten Phase wurden erst einmal kleinere Datensätze erzeugt, um beurteilen zu können, ob die Daten und die Automatisierungsansätze gut funktionieren. Bei größeren Datensätzen sollte zuerst damit begonnen werden, diesen großen Datensatz, in kleinere Sätze zu unterteilen. Wenn man mit 500-2000 Bildern arbeiten möchte, sammelt man diese Daten in Schichten und erhöht diese schrittweise, je nach den Ergebnissen des verwendeten Modells nach dem Training.

Sobald in dieser Erhebungsphase sichergestellt ist, dass eine angemessene Menge und Vielfalt an Daten erfasst wurde, beginnt man mit der zeitaufwändigsten Aufgabe des Projekts: der Annotation der Daten. Dabei werden die gesammelten Daten gelabelt und anschließend durch ein Modell laufen gelassen, um zu sehen, welche Ergebnisse das Modell liefert. Bei Bedarf wird der gewählte Ansatz angepasst. Anschließend sammelt man eine weitere Reihe von Daten und wiederholen den Vorgang. Wenn man Datenerfassungs- und Annotationsdienste anwendet, dauert es in der Regel drei bis vier Zyklen der Datenerfassung in Schichten, bis man herausfindet, was in Bezug auf die Modellleistung und den Zeit- und Kostenaufwand für die besten Ergebnisse am besten funktioniert. Das Sammeln von Daten in mehreren Stufen hilft dabei, unerwünschte Verzerrungen in den Daten zu vermeiden, die weniger offensichtlich sind, als sie es bei der Sammlung und dem Training mit größeren Datensätzen sind. Wenn man nicht in Schichten sammeln, müsste man, sobald diese unerwünschten Verzerrungen entdeckt werden, möglicherweise den gesamten Prozess neu beginnen. Nachdem man nun die Daten gesammelt und durch das Modell laufen gelassen hat, ist es an der Zeit für eine Validierungsübung. Mit der Validierung soll sichergestellt werden, dass die ursprünglich angestrebten Datenqualitätskriterien (d. h. Varianz, Qualität, Quantität, Dichte) erfüllt werden.

Der zweite Weg der Datenerfassung war die Nutzung eines Temperatursensors, z.B. dem DIAMEX Temperature-Sensor-Tester. Dabei ist zu beachten, dass hierbei keine Bilddaten erfasst werden, sondern reine Temperaturwerte, die über die Zeit auf Basis einer CSV-Datei generiert werden. Diese können dann direkt für Data Training und Validierungszwecke genutzt werden. Danach ist eine Vorhersage von unterschiedlichen Parametern möglich. Damit hat es die Möglichkeit gegeben die Daten mit Hilfe von Sensoren, die direkt an die BLDC-Motoren befestigt waren, die Temperatur aufzunehmen. Die Abbildung unten zeigt die verwendeten Sensoren. Das Ziel war es für die Prüfstände zwei unterschiedliche Methoden der Temperaturmessung zu verwenden. Damit hat man die Möglichkeit die Werte auf zweierlei Art und Weise zu prüfen.

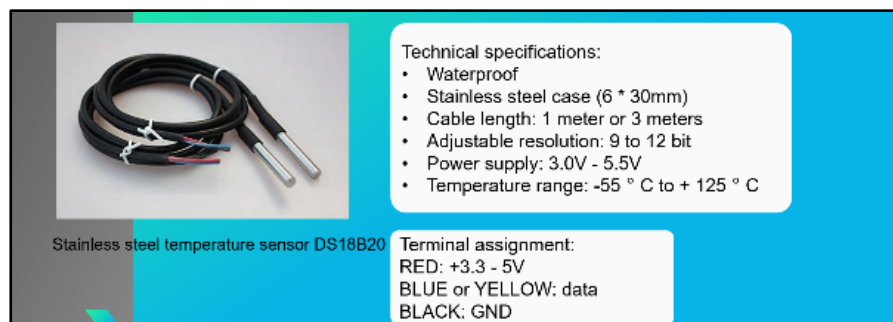


Abbildung 12: Aufbau der Tests

## Implementierung von Hard- und Software für den Demonstrator KI-ServoMot-Plattform

In diesem Kapitel werden die Implementierung und Konstruktion der Hardware und Softwarekomponenten für KI-IOT erläutert. Die Hardwarekomponenten wurden auf Basis einer CAD-System in 3D modelliert. Hierbei war es sehr wichtig, dass die Konstruktion alle relevanten Bauteile wie die Energieversorgung, Micro Controller, Ansteuerung und die Fixierung des Moduls auf die Roboter per Schnellmontage und

Anbringung zu realisieren. Die Abbildung unten zeigt das Gehäuse für das KI-IoT Modul. Für die Entwicklung des Prototyps wurden die HW-spezifischen Teile mit Hilfe von additive Fertigungsverfahren entwickelt.

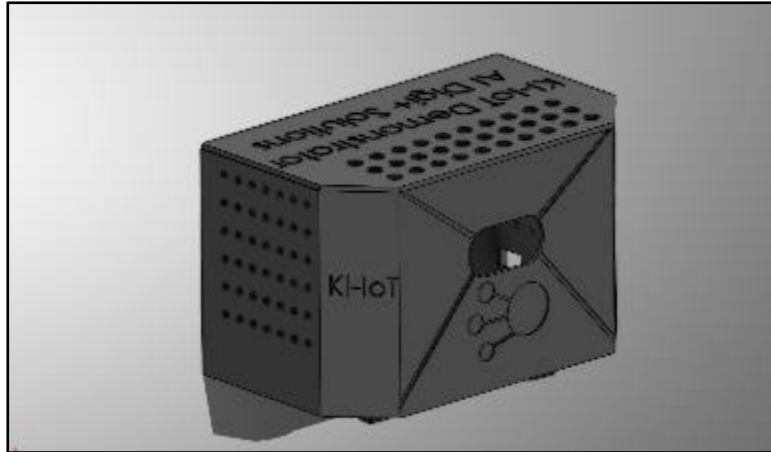


Abbildung 13: Konstruktion des Moduls

### Entwicklung des KI-IOT KI-ServoMot-Plattform Demonstrators

Für die Entwicklung der KI-IoT Pipeline hinsichtlich „Prädiktion“ von gemessenen Werten aus den Modulen der Messreihenuntersuchung und der Erkennung von Temperatur und Objekten wurde eine SW-Pipeline entwickelt. Dabei standen drei wesentliche Schritte im Vordergrund. Im ersten Schritt ging es um die Analyse und das Management von Daten hinsichtlich Qualität, Verteilung, Management und Diversität. Man muss dabei beachten, dass der erste Schritt nämlich die Datengewinnung und Datenqualität, einer der wichtigsten Schritte der Arbeiten darstellt. Mit Hilfe von zwei unterschiedlichen Sensoren wurde die Temperatur beim Betreiben des BLDC-Motors erfasst. Für den einen Temperaturmesssensor, den DS 18B20, wurde ein Kontakt zu den BLDC-Motoren hergestellt, indem dieser Sensor einfach mit dem Gehäuse des BLDC-Motors verbunden wurde und somit die Temperaturen beim Betreiben des BLDC-Motors in unterschiedlichen Drehmomenten und Drehzahlen gemessen wurden. Der zweite Sensor war die Thermokamera der Firma Seed. In einem zweiten Schritt wurden die erhobenen Daten bereinigt und automatisch in entsprechende SW-Tools wie LableMe und Excel (.CSV) weiterbearbeitet. Allein das Labeln der Daten war sehr aufwendig, da man jedes Bild einzelnen öffnen muss, um entsprechende Features/Klassen zu kennzeichnen. Dabei wurden neben den Bildern auch .XML-Dateien generiert. Die Abbildung unten zeigt, wie die Daten mit Hilfe einer Thermokamera von der Firma Seed erfasst werden. Dabei werden wie im Bild unten dargestellt optische und berührlose Temperaturmessungen durchgeführt. Hierbei wird der BLDC-Motor in verschiedenen Drehzahlen elektrisch angetrieben.

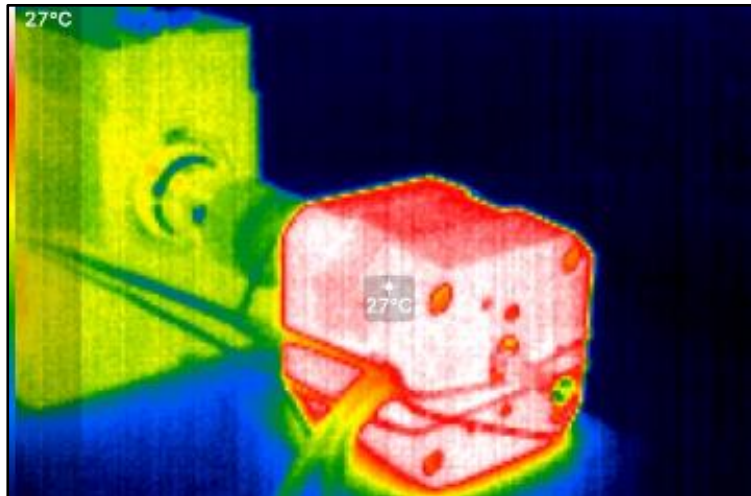


Abbildung 14: Optische Temperaturmessungen am BLDC-Motor

Die XML-Dateien beinhalteten die Kennzeichnung der Feature/Klassen inklusive der Bounding Boxen mit entsprechender Position der Bounding Box mit x und y Koordinaten. Das Wichtigste an der Stelle ist zu erwähnen, dass die Datengenerierung und das „Labling“ der Daten sehr viel Zeit gekostet hat. Das liegt daran, dass für diese Bildersets tausende Bilder manuell und einzeln geöffnet und bearbeitet werden müssen. Dieser Aspekt und Zeitaufwand waren uns am Anfang des Projekts nicht bewusst. Es war sehr mühsam die erforderliche Menge an Daten zu generieren und diese zu kennzeichnen. Es gibt auch aktuell wenig Möglichkeiten wie man das Labeln der Daten zu 100% automatisieren kann und da es sich speziell um Thermodaten eines BLDC-Motors es gehandelt hat, waren hierzu auch keine externen DataSets zu erhalten.

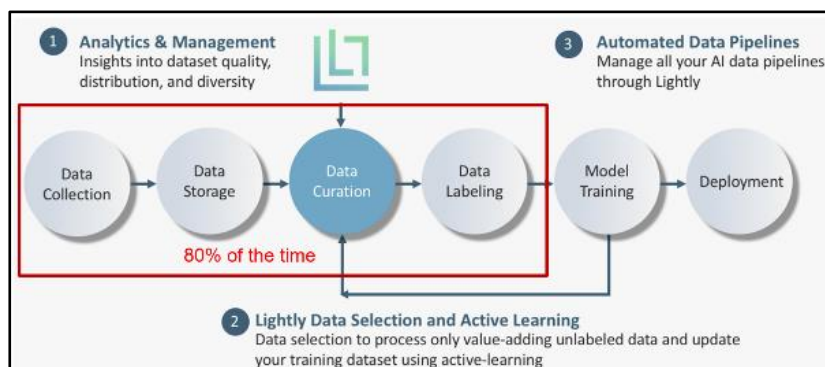


Abbildung 15: Implementierung des KI-ServoMot-Plattform für KI-IoT

### Herausforderungen hinsichtlich Daten Management

Bei der Erhebung der DataSets wurde festgestellt, dass das Daten Management eine besondere Herausforderung darstellt. Da man sehr große Datenmengen erstellt und auf Basis dieser Daten die Kennzeichnung bzw. das Labling der Daten vornimmt, war sehr wichtig sicherzustellen, dass die KI-basierten Trainingsergebnisse immer zu den entsprechenden Datensets passen und diese über den gesamten Entwicklungsprozess sauber dokumentiert sind. Um Datensätze zu trainieren, wählt man eher einen iterativen Prozess und arbeitet auf Basis von unterschiedlichen Datensätzen und Datenmengen. Das bedeutet, dass man immer versucht, stückchenweise an das

Optimum von Trainingsergebnissen versus Datensätze und Datenmenge heranzutasten, bis man das beste Trainingsmodell erstellt hat. Hierfür war das MAGICLOOP Tool sehr nützlich, da man hiermit für das jeweilige Projekt ein sauberes Daten-Management hat realisieren können. Das Bild unten zeigt das Management der DataSets mit entsprechenden gelabelten XML-Modellen.

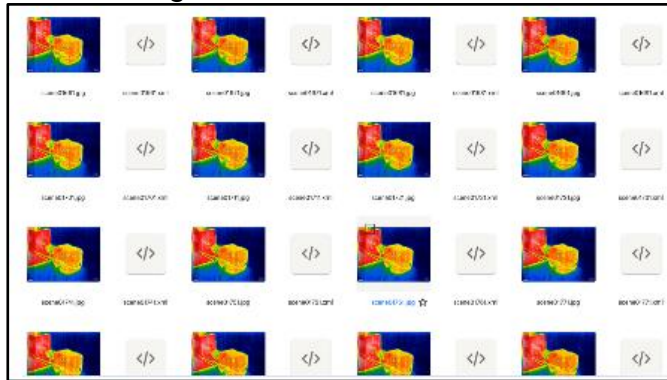


Abbildung 16: Management der gelabelten Daten in MAGICLOOP

### Generierung von DataSets für KI-ServoMot-Plattform

Nachdem die Messeinrichtung für die Erfassung der Daten aus dem Testumfeld abgeschlossen war, bestand der Fokus darin die erforderlichen Daten für Einspeisung in unterschiedliche Algorithmen für die Vorhersage der Temperaturen bei unterschiedlichen Drehzahlen zu generieren. Hierzu wurde eine Pipeline entwickelt, mit der es möglich sein sollte, die erfassten Daten von der DataSet Nutzung bis hin zur Validierung zu automatisieren. Dazu wurde im ersten Schritt eine Art Pilotnutzung der DataSets generiert. Dabei wurde stets darauf geachtet, welche „Features“ später zum Einsatz kommen sollten. Unten sind die unterschiedlichen wichtigen Parameter seitens Dataformat, Features und Validierungsaspekte definiert.

1. Name of features in dataset: The running time in (s), Torque in (mNm), rotational speed in (rpm) and Temperature in (C)
2. Data format: .csv
3. Ratio training: 80%
4. Ratio Evaluation: 20%

Des Weiteren wurden die Inhalte des DataSets in Form einer .csv-Datei definiert. Dabei war festzustellen, dass eine automatische Generierung der Daten ohne händische Nacharbeit sich als nicht einfach darstellen ließ. Das lag unter anderem darin, dass nach der Messung die Daten in eine .csv-Datei gewandelt wurden und in den Zellen und Spalten der Excelliste z.B. das Format der eingelesenen Daten nicht verarbeitbar waren. Es waren so simple Punkte wie die Trennung von Werten durch Punkt oder Komma. Diese Fehler wurde dann mit der Umformulierung der jeweiligen Scripts behoben und korrigiert. In der Abbildung unten ist ein Auszug der genutzten .csv-Datei zu sehen. Für die Generierung der DataSets, das Training und die Validierung wurde ein iterativer Prozess angestrebt. Der Vorteil hierbei liegt darin, dass durch die Anwendung von kleineren Zyklen und Tests von DataSets ein Schritt für Schritt Test der Algorithmen durchführbar ist und eine Anpassung der Daten möglich ist.

Für die Durchführung des Trainings wurden Algorithmen aus Keras genutzt: Input Features: the running time, the torque and the rotational speed, Structure of model.

Die Abbildung unten zeigt die unterschiedlichen Dense Layer und die jeweiligen Aktivierungsfunktionen. Hierbei wurde ReLu als Aktivierungsfunktion benutzt, da ReLu eine einfache und mathematisch simple (bezogen auf den Berechnungsaufwand) Nutzung bietet. Andere Aktivierungsfunktionen wie SoftMax etc. wurde hierbei nicht genutzt.

```
inputshape=len(feature_names)
model = tf.keras.Sequential([
    tf.keras.layers.Dense(32, activation=tf.nn.relu, input_shape=(inputshape,))
    tf.keras.layers.Dense(16, activation=tf.nn.relu),
    tf.keras.layers.Dense(1)#len(class_names)
])
```

Abbildung 17: Auszug Algorithmen

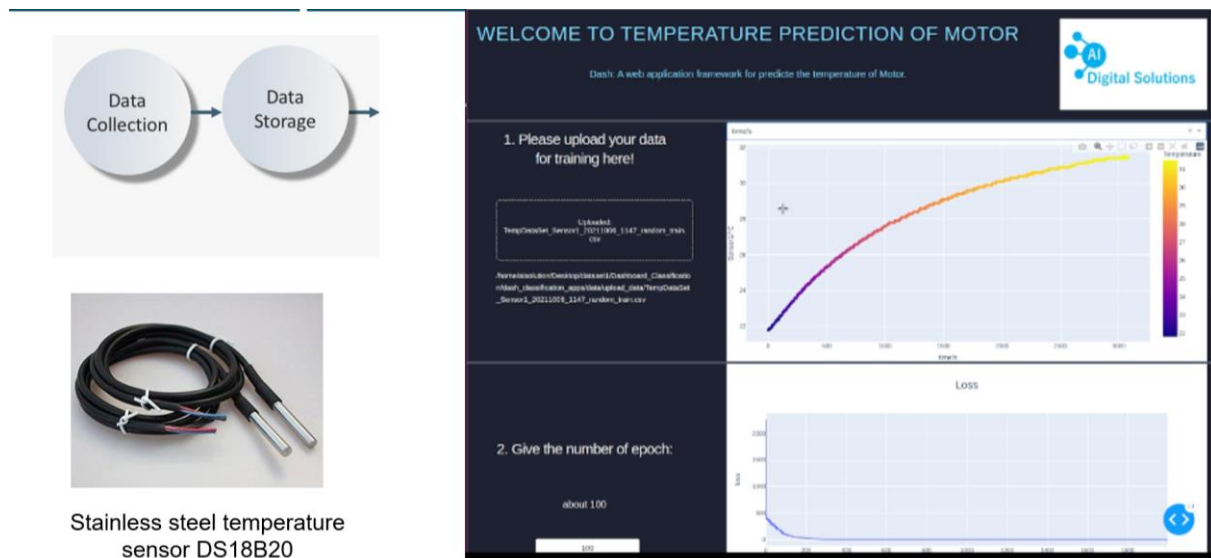


Abbildung 18: Temperaturverlauf KI-IOT KI-ServoMot Pipeline

Für die Beschleunigung und User freundliche Nutzung der KI-Pipeline wurde eine Benutzeroberfläche entwickelt. Auf dem Bild unten sind man eine webbasierte Anwendung, die dafür gedacht ist, die generierten Daten und DataSets automatisch für die Durchführung der Trainings und DataSets einzulesen und dadurch die Brüche im Prozess hinsichtlich Nutzung der Daten zu minimieren. Im ersten Schritt hat der Nutzer die Möglichkeit seine Daten in Form von .csv einzulesen und diese für die Nutzung zu selektieren. In einem zweiten Schritt (ebenfalls wie im Bild unten zu sehen) kann der Anwender seine trainingsrelevanten Informationen wie z.B. Anzahl der Epochs eingeben.

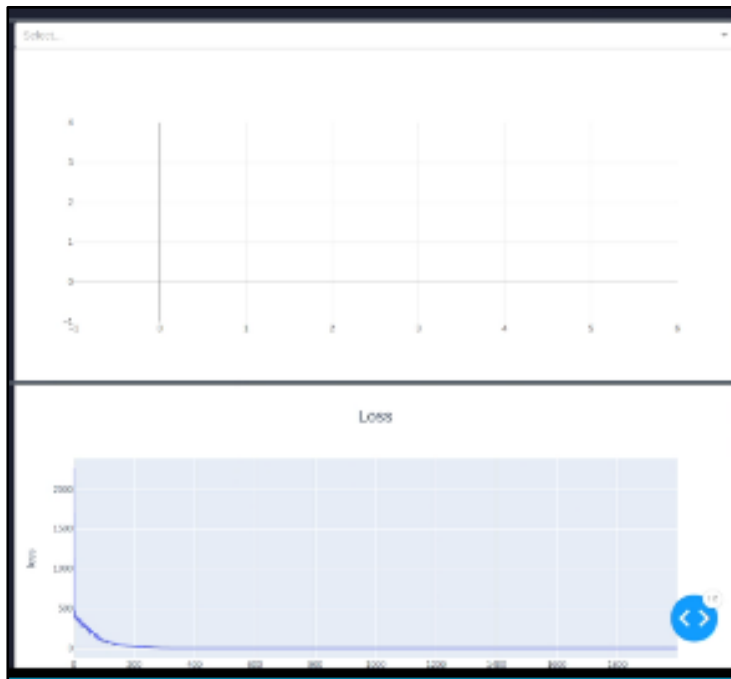


Abbildung 19: Temperaturverlauf KI-IOT KI-ServoMot Pipeline

Nachdem die Daten hochgeladen sind, können diese in der Maske vorangezeigt werden, damit man einen guten Überblick über die Qualität und den Verlauf der Daten haben kann. Dies hat auch den Vorteil, dass der Vergleich der realen Daten und die Vorhersagen der Werte miteinander gut verglichen werden können. Konkret heißt das, dass die gemessenen Temperaturwerte mit den Werten die „predicted“ werden sollen, gut verglichen werden können. Die Abbildung unten zeigt den Temperaturverlauf der erfassten .csv-Daten. Dabei kann man sehr gut den in rot markierten, vorhergesagten Temperaturverlauf sehen und in blau die tatsächlich gemessenen Werte. Daraus lässt sich schlussfolgern, dass die Algorithmen, die für diesen Use Case benutzt wurden, sehr gut gearbeitet haben und man dadurch auf Basis dieser Betrachtung die Anzahl der Messungen, Tests und Analysen reduzieren kann und damit viel Zeit und Geld spart. Man wird aber zukünftig nicht ganz darum herum kommen weitere Tests zu fahren.

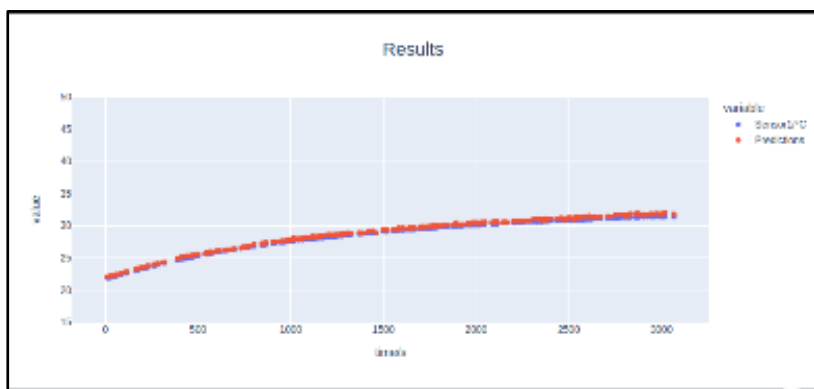


Abbildung 20: Temperaturverlauf KI-IOT KI-ServoMot Pipeline

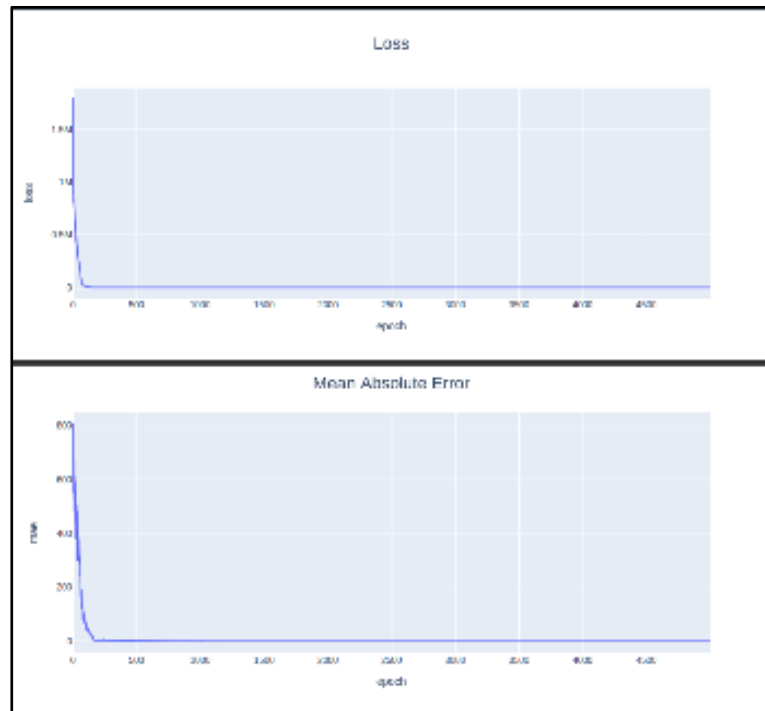


Abbildung 21: Temperaturverlauf KI-IOT KI-ServoMot Pipeline

## 5 Zusammenfassung

In dem Forschungsprojekt K-IoT wurden unterschiedliche Use Cases für einen BLDC-Motor aufgebaut und untersucht. Hinsichtlich der IoT Fähigkeit wurden sowohl optische als auch taktile Devices integriert und aufgebaut. Hierzu wurden Beispiel aus dem Bereich Computer Vision, Temperaturmessung und Simulation von BLDC relevanten Parameter genutzt. Darauf basierend wurden unterschiedlich KI-Ansätze integriert und ebenfalls in Form von KI-ServoMot-Plattform implementiert. Es sind dabei sehr gute und praxisrelevant KI-IoT-Anwendungen entstanden. Dafür bedanken wir uns beim Projektträger und BMBF.