Seminar für Sprachwissenschaft

Internship Report as Researcher and Developer

June 15, 2025

Aida Rostami

aida.rostami@student.uni-tuebingen.de

Supervised by:

Marcel Dann, BEC Robotics

Marcel.Dann@bec-robotics.com

Dr. Çağrı Çöltekin, Eberhard Karls Universität Tübingen

cagri.coeltekin@uni-tuebingen.de

1 Introduction to BEC Robotics

1.1 Company Profile: BEC Robotics

BEC Robotics[1] is a German technology company specializing in advanced robotic systems for human-robot collaboration across industrial, medical, and entertainment sectors. Founded in 2003, the company has grown into a recognized leader in the field, delivering safe, efficient, and highly customizable robotic solutions.

Headquartered in Pfullingen, Germany, BEC Robotics operates a subsidiary in Atlanta, USA (BEC Corp.) and maintains a dedicated research facility in Magdeburg. The company develops cutting-edge applications, including autonomous industrial robots, precision medical devices for radiotherapy, and interactive robotic systems for amusement parks. Notably, BEC is certified under ISO 13485, underscoring its compliance with medical device quality standards.

The company is organized into four core divisions:

- **BEC Industry**: Focused on industrial automation, mobile robotics, and smart manufacturing.
- **BEC Medical**: Develops medical robotic systems for patient alignment and rehabilitation.
- **BEC Entertainment**: Designs robotic experiences, simulators, and entertainment systems.
- **BEC Service**: Offers training tools and simulation platforms for aviation and automotive use.

What distinguishes BEC Robotics is its interdisciplinary engineering approach and its deep commitment to human safety, especially in close human-robot interaction scenarios.

1.2 My Role and Responsibilities

As a Research and Development Working Student in the Industry division, I had the opportunity to contribute to real-world robotics projects, primarily focusing on software development, systems integration, and diagnostics for industrial robots. BEC offers both six-month internships and longer-term working student positions, allowing students to gain meaningful experience in cutting-edge technology environments.

2 Tasks During the Internship

During my time at BEC Robotics, I was involved in a variety of challenging tasks. These ranged in scope and complexity, and by the time of writing this report, some of them are still in progress. In this section, I will provide a brief overview of the tasks I worked on, including a description of each project, the strategies I used to approach the problems, and the skills I developed throughout the process.

[1] https://www.bec-robotics.com/en/robotics/

The tasks often required problem-solving in unfamiliar areas and learning new technologies on the fly. Although some assignments had clearly defined goals, others were more exploratory in nature, giving me the chance to make creative decisions and propose my own solutions. This mixture of structure and freedom helped me grow both technically and professionally during the course of the internship.

2.1 Task 1: Enhancing a Generic Data Logger Application

My first major project involved extending and refining a pre-existing data logger application, which had been implemented using Python Dash and Plotly. The main purpose of this tool was to collect and visualize real-time data generated by robotic systems, making it easier for developers and engineers to monitor system states and evaluate performance over time. In the context of robotics development, such tools are essential for diagnostics, safety evaluation, and iterative improvement of system behavior.

One of the primary challenges was improving how the robot's physical position within a workspace was displayed. The application originally used a simple X-Y coordinate plot to visualize the robot's movements, but this offered limited insight into its actual position in the real-world environment. My task was to improve the spatial clarity of this visualization by overlaying a reference map onto the plot, allowing viewers to contextualize robot movements more intuitively.

Implementing this feature was not trivial. Integrating static or dynamic background images in Dash required a precise understanding of how layout containers and coordinate scaling interact. It took several iterations to find a solution that was both technically robust and visually accurate. I consulted online documentation, experimented with several plotting strategies, and reviewed similar community problems. The issue had remained open in the team for several months prior to my involvement, which made it especially satisfying to eventually resolve it successfully.

Beyond visualization improvements, I also contributed to extending the application's functionality by adding session persistence. This feature allowed the system to store historical data and retrieve previous sessions, making it easier to perform comparative analyses or debug specific behaviors. I achieved this using the `dcc.Store` component from Dash, which supports client-side storage of data. Learning to structure callbacks that linked user input, stored data, and dynamic graphs helped me gain deeper insight into state management within Dash-based applications.

Skills Gained from Task 1:

- Practical experience with Dash and plotly for creating interactive, web-based data dashboards.
- Data visualization and diagnostics, including performance tuning and interface usability.
- Using Git to manage project versions and contribute to shared repositories.

This task not only enhanced my technical skill set but also showed me the importance of good software tools in robotics workflows. If I were to continue this project, I would aim to further generalize the application for broader reuse across different robot configurations.

2.2 Task 2: Logging and Visualizing Alert Data from KMR iisy

This project was closely related to the previous task, as it also involved extending the diagnostic capabilities of robotic systems through data visualization. However, while Task 1 focused on logging continuous robot data, this assignment dealt specifically with collecting and analyzing alert and error messages generated by KUKA mobile robots (KMR), particularly the *KMR iisy* model. The original system lacked a mechanism to log and visualize these alerts, which made it difficult to trace failures or anomalies during robot operation.

The robots communicated using the MQTT protocol—a lightweight messaging system commonly used in IoT and robotics applications. My first step was to understand the message structure using tools such as MQTT Explorer. The messages were encoded in JSON format and often arrived at high frequency, so I designed a pipeline in Python to parse the messages, extract relevant fields, and store them in a structured format within an SQLite database.

The next step was to create a dashboard that would make this information more accessible. I developed a new visualization interface using Dash and Plotly, which allowed users to explore the logged alerts and gain insights into system behavior. The interface was designed to be interactive, enabling users to filter errors by severity, timestamp, or mission ID.

A key challenge in this project was the lack of explicit context in the alert messages. For example, the alerts did not directly indicate which robot mission they were associated with, making root cause analysis difficult. To address this, I implemented a timeline analysis approach. By aligning different types of messages based on their timestamps and expected mission durations, I was able to infer connections between alerts and specific robot tasks. This significantly improved the ability to trace faults to specific events, leading to more effective troubleshooting.

This task helped me further develop my technical skills in both software development and data engineering. I learned how to handle streaming diagnostic data, how to design a robust storage solution using relational databases, and how to improve the transparency of robotic systems through thoughtful visualization.

Skills Gained from Tasks 2:

- Practical experience with the MQTT communication protocol and message parsing.
- Database design and management using SQLite for storing structured robotic data.
- Deeper understanding of how KUKA Mobile Robots transmit diagnostic and alert information.

- Experience with analyzing temporal data to infer system states and relationships.

2.3 Task 3: Studying and Programming the IOSS WID Camera

This task marked my first experience working directly with hardware components in a robotics context, which presented both technical and conceptual challenges. The focus was on integrating an industrial barcode scanning system into a broader robotic workflow. Specifically, I worked with the IOSS WID camera, a high-performance device used for barcode reading and visual inspection tasks in industrial environments.

My responsibilities included gaining an understanding of the camera's hardware capabilities and software interface, learning how it could be integrated into the existing system, and implementing a communication layer in Java. At the beginning of the project, the only available Software Development Kit (SDK) was written in C/C++. Since the rest of our system was Java-based, this required me to use the Java Native Interface (JNI) to bridge the gap between the C/C++ SDK and our Java application.

JNI is a powerful but low-level method of integrating native libraries into Java applications, and using it requires careful attention to data types, memory management, and function binding. Later in the project, a basic Java library was made available by the manufacturer. While this simplified integration to some extent, the library lacked advanced features and had limited documentation. As a result, I had to evaluate both solutions and selectively combine elements from each to achieve reliable communication with the device.

Despite these challenges, I successfully developed functional Java code that allowed us to connect to the camera, configure its parameters, and receive barcode data. This code served as a foundational component for future integration into a larger robotic system. Through this task, I not only strengthened my programming skills in both Java and native code interoperability but also developed a better understanding of real-world constraints when working with third-party hardware.

Skills Gained from Task 3:

- Gained hands-on experience in integrating industrial hardware into robotic systems.
- Learned to work with vendor SDKs, including C/C++ libraries and JNI in Java.
- Developed a better understanding of barcode scanning technologies and their role in automation.
- Improved my ability to troubleshoot hardware-software communication issues.

2.4 Task 4: Integrating the IOSS Camera with the KUKA KMR iiwa Robot

Building on the outcomes of Task 3, the next step was to integrate the IOSS WID barcode scanning camera into an automated robotic workflow involving the KUKA LBR iiwa 7 R800—a 7-axis lightweight collaborative robot designed for high-precision and safe human-robot interaction. This project required connecting a third-party industrial camera to a tightly constrained robotic software environment, combining hardware control, network communication, and real-time decision-making.

The KMR iiwa (Intelligent Industrial Work Assistant) stands out due to its seven degrees of freedom, which provide a high level of dexterity comparable to that of a human arm. It is equipped with integrated torque sensors in each joint, enabling it to detect external forces and adapt its movements dynamically. This capability is essential in environments where safety, adaptability, and fine manipulation are required, such as in semiconductor handling, medical robotics, or collaborative assembly.

The KUKA Sunrise IDE is used to program the iiwa robot and is based on Java 7, which posed a significant challenge during this task. The IOSS WID camera, by contrast, required a newer Java version to use its API properly. Since upgrading the robot controller's Java runtime was not an option due to compatibility and stability constraints, I had to design an alternative integration strategy.

To overcome this, I developed a solution using telnet communication. The robot controller established a telnet connection to the camera, allowing it to trigger camera scans and receive barcode results indirectly. This approach successfully bypassed the version conflict without compromising the reliability of the robot's control system.

Once the communication pipeline was established, I programmed a complete robotic task sequence using the KUKA Sunrise environment. In this sequence, the robot:

- Moved to predefined positions to scan silicon wafers.
- Triggered the IOSS camera to perform a barcode scan at each position.
- Retrieved and interpreted the scan results through the telnet interface.
- Performed appropriate follow-up actions, such as retrying a scan or flagging a wafer if the barcode could not be read.

This task required some understanding of robot kinematics and collaborative safety mechanisms, as well as robust Java development and system integration skills. It also highlighted the importance of designing flexible communication architectures when working with legacy systems and modern hardware simultaneously.

Skills Gained from Task 4:

- Experience in robotic programming using the KUKA Sunrise Java-based development environment.
- Advanced problem-solving involving version compatibility and cross-system communication.
- Design and implementation of telnet-based inter-process communication.
- Practical understanding of robot kinematics, safety zones, and barcode scanning in automated workflows.
- Integration of external devices into closed robotic systems under strict software constraints.

2.5 Task 5 (Ongoing): Studying Navitrol, CAN Communication, and Industrial PCs (IPC)

In the latest stage of my internship, I have begun working on a new task that centers around the Navitrol system, a navigation and control platform used in robotic systems to manage motion, localization, and trajectory planning. The goal of this task is to understand the system's architecture and prepare for its integration into our existing robotic workflows.

This project has introduced me to new aspects of robotics involving lower-level communication and control. In particular, it requires developing familiarity with several key technologies and concepts:

- **CAN (Controller Area Network) communication:** a widely adopted protocol in robotics and embedded systems, used for reliable communication between microcontrollers, sensors, and actuators.
- **Industrial PCs (IPCs):** specialized computers that serve as central control units in automation environments. They are designed to handle real-time tasks, manage hardware communication, and ensure stable performance in harsh industrial settings.
- **Diagnostics and system configuration:** understanding how to configure Navitrol for specific use cases, interpret diagnostic messages, and trace issues related to sensor integration or communication failures.

Although this task is still ongoing, it is already offering valuable insights into the internal workings of advanced robotic control systems. Compared to my earlier projects, which were more application- and interface-focused, this assignment requires a more theoretical and systems-level perspective. By working closely with IPC-based setups and CAN interfaces, I am gradually building a clearer understanding of how complex robotic components synchronize and interact in real-time environments.

Skills in Progress from Task 5:

- Gaining foundational knowledge of CAN communication protocols used in robotics.
- Learning about the role and configuration of industrial PCs in control systems.

- Understanding the structure and diagnostic capabilities of the Navitrol platform.
- Expanding theoretical knowledge related to low-level robotic system design.

3 Key Takeaways from My Internship

My internship at BEC Robotics has been a deeply enriching and transformative experience, both technically and professionally. It offered me a rare and valuable opportunity to immerse myself in the real-world challenges of research and development within the robotics industry. Beyond learning specific tools and technologies, I developed a much deeper appreciation for the problem-solving mindset and interdisciplinary thinking required in this field.

At the beginning, the experience felt intimidating. I was entering a completely new domain—working hands-on with industrial robots whose manuals contained multiple pages of warnings and safety precautions. The idea that a single mistake could cause physical harm made me initially hesitant and cautious. However, with time, mentorship, and experience, I became more confident and capable in operating within this high-stakes environment.

Throughout my time in the Industry division, I encountered the practical realities of working with advanced robotic systems: dealing with hardware limitations, interfacing with legacy software, navigating undocumented protocols, and testing in live setups. These situations pushed me beyond the confines of classroom learning and helped shape a more resilient and independent working style.

Some of the most significant takeaways from this experience include:

- Learning to work autonomously in highly technical projects, often without predefined instructions or readily available support.
- Developing the ability to read and interpret SDK documentation and technical manuals, and to implement solutions through iteration and experimentation.
- Gaining hands-on experience with industrial robots, including adherence to safety standards, understanding motion control, and working within physical constraints.
- Observing and participating in the complete engineering cycle—from identifying a need, prototyping a solution, testing in situ, and eventually integrating it into production systems.

Most importantly, this internship made me a more capable, self-reliant person and has deepened my motivation to pursue future work in intelligent systems and applied research in robotics.

4 Acknowledgements

I would like to express my sincere gratitude to my internship supervisor at BEC Robotics, Marcel Dann, for his continuous guidance and support throughout the internship. His regular feedback sessions greatly contributed to my learning and progress. I am also thankful for the opportunity to work in such a friendly, innovative, and advanced environment.

Special thanks go to Lisa Willich, who was involved in the interview and selection process. Her encouragement, along with her ongoing support during the internship, was invaluable both professionally and personally.

Finally, I would like to thank the University of Tübingen for providing me with the opportunity to complete this internship as part of my Bachelor's program. The experience has been a vital step in my academic and professional development.