Machine Learning-Driven Detection of Repetitive Manufacturing Processes Using Radar Sensor

Bc. Tomáš Martiník
Department of Control and Instrumentation
Brno University of Technology
Brno, Czech Republic
xmarti95@vutbr.cz

***Abstract*—This paper presents a non-invasive system for detecting repetitive manufacturing cycles using pulse-coherent radar and machine learning. The Acconeer A111 radar sensor, combined with an Arducam USB camera, is integrated within a ROS2-based data acquisition framework. The system operates in Envelope and Sparse radar modes, optimized for tracking static and dynamic motion. A YOLO-based model analyzes radar heatmaps to detect repetitive cycles automatically.**

The approach was validated through controlled experiments and in an industrial setting. Results demonstrate the system's potential to accurately detect production cycles without modifying existing machinery, highlighting its potential for real-time process monitoring and optimization.

***Index Terms*—Radar sensing, ROS2, data collection, machine learning, production monitoring.**

I. Introduction

The rapid development of Industry 4.0 has highlighted the importance of data-driven manufacturing, where precise monitoring of production processes is essential for optimizing performance and ensuring efficiency. Traditional approaches to production cycle tracking often rely on direct machine integration, requiring physical modifications and additional sensor installations. These solutions are not only costly but can also lead to disruptions in manufacturing operations, requiring re-certification of production lines.

To address these challenges, this paper presents a non-invasive system for monitoring production cycles using a combination of a pulse-coherent radar sensor and a camera. Unlike conventional methods, the proposed approach does not require any modifications to existing machinery, making it highly scalable and adaptable to different manufacturing setups. The primary objective is to develop an automated system capable of detecting repetitive cycles in real-time using radar data, thus providing valuable insights into production efficiency without interfering with industrial workflows.

II. Pulse-Coherent Radar Overview

Pulse-coherent radar enables precise motion and distance measurements by analyzing phase and amplitude variations in reflected electromagnetic waves. This paper utilizes the Acconeer A111 radar sensor, which supports multiple acquisition modes.

A. Principle of Operation

The Acconeer A111 emits electromagnetic pulses, which reflect off objects and return to the sensor. By measuring the time delay and phase shift, the radar determines distance and motion. Coherent processing enables detection of small displacements, making it effective for cycle tracking.

B. Radar Data Acquisition Modes

The radar supports several modes, each optimized for different applications:

- **Envelope Mode** – High-precision distance measurement, ideal for detecting static and slow-moving objects.
- **Sparse Mode** – Captures motion with high temporal resolution, allowing velocity estimation.
- **IQ Mode** – Provides complex in-phase (I) and quadrature (Q) data, suitable for Doppler-based motion analysis.
- **Power bins** – Provides basic object presence information.

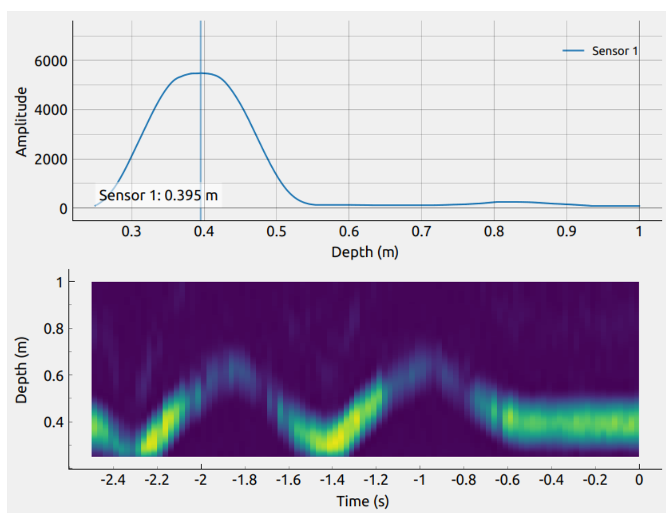

Fig. 1. Example of Envelope mode output.

After evaluation, **Envelope** and **Sparse** modes were selected as the most effective for detecting production cycles.

1) Envelope Mode: Envelope mode captures reflection amplitude across the detection range, providing high spatial resolution. Initial testing at **10 Hz** sampling was insufficient for tracking cycles, leading to an optimized **40 Hz** rate, ensuring consistent detection of moderate-speed objects.

2) Sparse Mode: Sparse mode provides reduced data density but higher temporal resolution, making it suitable for tracking rapid motion. Two configurations were tested: **16 sweeps per frame** at **90 Hz**, prioritizing higher frequency, and **32 sweeps per frame** at **50 Hz**, capturing finer motion details. Sparse mode measures at 6 cm intervals but accumulates multiple sweeps per frame, enabling high-speed motion tracking. The 32-sweep configuration provided the best trade-off between resolution and update rate [1].

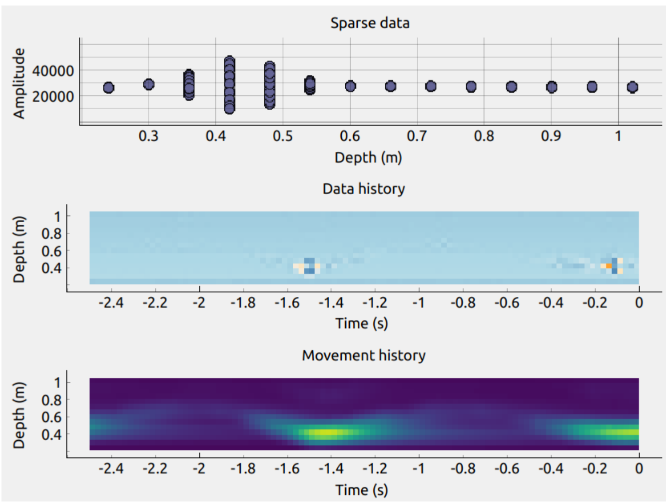

Fig. 2. Example of Sparse mode output.

C. Final Radar Configuration for Cycle Detection

The following radar configurations were used:

- **Envelope Mode** (40 Hz) – Optimized for detecting static or slow-moving components.
- **Sparse Mode** (50 Hz, 32 sweeps per frame) – Captures rapid motion changes and velocity estimation .

III. Design and Fabrication of a Sensor Holder

A custom 3D-printed holder was designed to ensure precise alignment and stable mounting of the radar sensor and camera. Developed in Fusion 360, the holder was optimized for robustness, ease of assembly, and tripod compatibility.

A. Design Requirements

The holder was designed to meet the following key requirements:

- **Parallel alignment:** Ensuring synchronized radar and camera data capture.
- **Stable mounting:** Secure attachment to a tripod with minimal vibrations.
- **Cable management:** Openings for organized routing of power and data cables.
- **Modularity:** Easy assembly, disassembly, and modification if needed.

B. CAD Modeling and 3D Printing

The radar sensor and camera were positioned as close as possible while maintaining accessibility for mounting screws. The optimal device spacing was determined to be **42.6 mm**. Key features of the 3D model include:

- Snap-fit slot for the Acconeer A111 radar sensor.
- Recessed compartment for the PCB and connectors.
- 1/4"-20 UNC threaded hole for tripod mounting.

To ensure durability, the back wall thickness was set to **4 mm**. Fig. 3 shows the final 3D model.

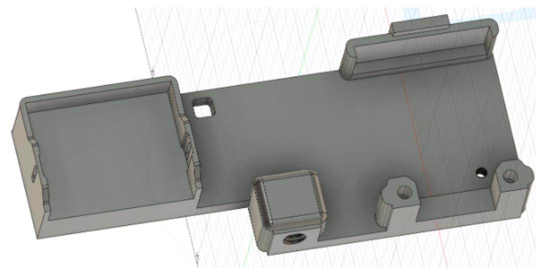

Fig. 3. 3D model of the sensor holder, designed in Fusion 360.

C. 3D Printing and Assembly

The holder was printed on an **Original Prusa i3 MK3S+** using **black PETG filament**, selected for its durability and slight flexibility. The key print parameters were:

- **Layer height:** 0.20 mm (optimized for a balance between precision and print speed).
- **Infill density:** 15% (ensuring structural integrity while minimizing material usage).
- **Supports:** Not required due to an optimized design.

The printing process was completed in approximately **3 hours**, followed by minor post-processing, including refining the tripod thread.

During assembly, the Acconeer A111 radar module was securely snapped into its designated slot. The PCB was mounted using M2 screws and 6 mm standoffs to ensure a stable fit. The Arducam USB camera was carefully positioned and fixed in place with M3 screws. To maintain a clean and organized setup, all cables were routed through dedicated openings. Finally, the entire assembly was mounted onto a standard tripod, ensuring a stable platform for data collection.

D. Evaluation and Refinements

Initial testing confirmed proper alignment and stable operation. Several minor design improvements were implemented:

- Increased back wall thickness to **4 mm** for improved stability.
- Adjusted the snap-fit mechanism to simplify sensor insertion and removal.
- Added an indentation for easier access to camera screws.

These refinements resulted in a durable and practical sensor mount, suitable for long-term industrial monitoring applications.

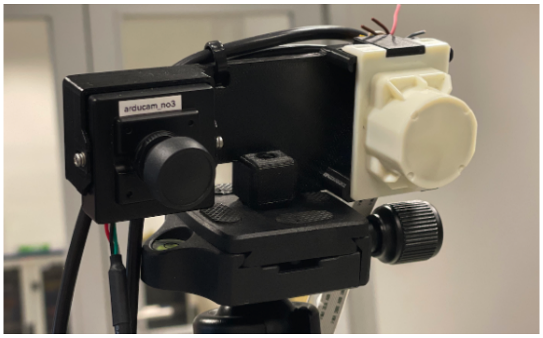

Fig. 4. Final assembled holder with radar sensor and camera.

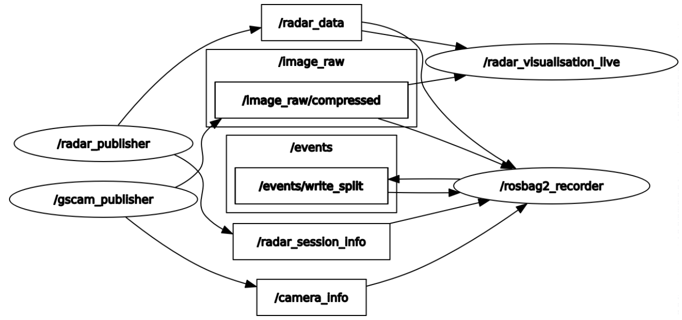

Fig. 5. System architecture of radar and camera data acquisition in ROS2.

IV. ROS2-Based Data Collection System

To ensure efficient and modular data acquisition, the system was implemented within the ROS2 framework. This architecture enables seamless synchronization between radar and camera data while facilitating real-time processing and storage. The data collection system runs in a custom Docker container and consists of two primary ROS2 nodes.

A. Docker-Based ROS2 Environment

To create a reproducible and portable development environment, a dedicated Docker container was designed for running ROS2-based data acquisition. The container includes:

- **ROS2 Jazzy** – the latest ROS2 distribution.
- **GStreamer and OpenCV** – for handling video streams.
- **Acconeer Exploration Tool** – for direct radar communication.
- **Custom Python environment** – with dependencies for radar data processing and visualization.

The Docker container ensures consistency across different machines and allows deployment on embedded hardware platforms without manual installation of dependencies.

B. Data Collection Nodes

The system consists of two primary ROS2 nodes responsible for data acquisition and publishing:

- **GSCam2 node:** This node captures and publishes image data from the Arducam USB camera. The camera is set to operate at 10 FPS and 1920×1080 resolution, and the video stream is compressed to 640×360 resolution using H.264 encoding for efficient transmission. The node is configured to output images in JPEG format, making it suitable for real-time processing and storage [2].
- **Radar Reader Node:** A custom ROS2 node developed in Python using Acconeer's SDK. It collects radar data in real time and publishes it to a ROS2 topic. The node supports both Envelope and Sparse mode.

Both nodes publish synchronized data streams, which are recorded into ROS2 bag files for offline analysis.

C. Automated System Launch

To streamline data collection, an XML-based launch file was created to automatically start both ROS2 nodes with predefined parameters. This configuration ensures proper synchronization and allows rapid switching between different radar modes.

The launch file includes:

- **Camera parameters** (device ID, resolution, encoding format).
- **Radar configuration** (serial port, range interval, update rate, service mode).
- **QoS settings** to ensure reliable data transmission in real-time applications.

The system's modularity allows easy extension with additional sensors or data processing pipelines, making it adaptable for various industrial monitoring applications [3], [4].

V. Experimental Setup and Initial Measurements

To validate the proposed system and optimize radar parameters, a series of controlled experiments were conducted to assess its ability to detect repetitive motion patterns. The experiments included:

- **Hand movement:** Initial data acquisition testing.
- **Pendulum motion:** Cyclic motion for machine learning-based cycle detection.
- **Robotic arm:** Industrial motion tracking with structured and unstructured cycles.
- **Industrial deployment:** Real-world cycle measurement in a manufacturing environment.

All experiments were conducted using both Envelope and Sparse radar modes, with optimized sampling frequencies based on preliminary findings.

A. Hand Movement and Pendulum Tests

The first tests involved simple hand movements over the radar sensor to validate live data visualization and logging. A real-time tool displayed radar responses, allowing immediate assessment of signal quality.

Subsequently, a pendulum was introduced to generate periodic motion patterns. Data from 32 recorded cycles were annotated and used to train a YOLOv8 model for cycle detection. The model successfully identified all cycles, confirming the radar's capability to detect repetitive motion [5].

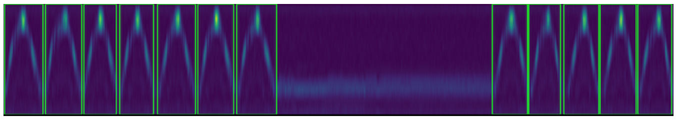

Fig. 6. Example of YOLO-based cycle detection on radar heatmap.

B. Robotic Arm Motion Analysis

To further validate the system, a robotic arm was programmed to execute both repetitive and irregular motions.

- **Cyclic movements:** The system reliably detected repetitive robotic arm actions.
- **Irregular sequences:** More complex radar responses were observed, requiring additional filtering to distinguish structured patterns from noise.

C. Industrial Deployment

The system was tested in a real manufacturing setting across three use cases:

- **Simple container movement:** Monitoring repeated box movements.
- **Drilling station:** Detecting operator-initiated cycles with variable timing.
- **6-Axis robot workstation:** Observing automated pick-and-place operations.

Radar-based cycle detection successfully identified structured movements, demonstrating its potential for non-invasive monitoring in industrial environments.

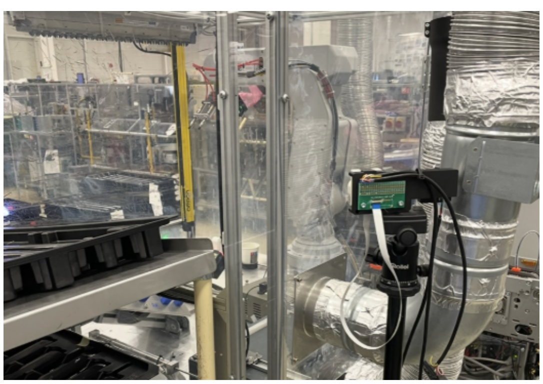

Fig. 7. Setup of industrial deployment experiment.

D. Results and Observations

The radar effectively captured cyclic motion in all tested scenarios. **Envelope mode** provided clear cyclic patterns in controlled environments, making it ideal for simpler tasks. However, in industrial settings, **Sparse mode** with variance-based post-processing was more effective. By computing sweep variance at each point, it suppressed static objects and highlighted only moving elements, reducing interference from the surroundings.

While Envelope mode excelled in detecting periodicity in structured motion, Sparse mode proved essential for real-world applications where background noise and multiple moving parts are present. This combination allows flexible adaptation to different monitoring needs.

VI. Conclusion and Future Work

This paper focused on data collection and initial validation of a radar-based system for real-time cycle detection in industrial production. While the system successfully captured cyclic motion patterns using pulse-coherent radar and a camera within a ROS2-based framework, it remains in the research phase and requires further development for practical deployment.

The next step is to implement the detection system as a dedicated ROS2 node, enabling real-time operation in an industrial environment. The system will autonomously monitor production cycles, adaptively learn motion patterns, and transition to real-time cycle detection once sufficiently trained.

Additionally, considerations for real-world deployment on embedded platforms remain a key challenge. The current YOLO-based detection model, while effective, is too large for resource-constrained microcontrollers (MCUs) or edge devices. Future work will focus on optimizing model size or exploring alternative signal-processing techniques to ensure efficient execution in embedded environments.

The long-term goal is to develop a fully autonomous system that can continuously learn, adapt to various manufacturing processes, and provide real-time cycle monitoring without requiring modifications to existing production lines.

Acknowledgment

This work was conducted as a master's thesis under the supervision of Ing. Karel Horák, Ph.D., at the Department of Control and Instrumentation, Brno University of Technology. The experimental measurements were carried out at ALPS Electric Czech, s.r.o., where they were assigned and overseen by Ing. Josef Nevrlý. The author acknowledges the use of OpenAI's ChatGPT as a support tool for translation and language refinement.

References

[1] Acconeer AB. *Acconeer Documentation*. Online. Available at: https://docs.acconeer.com/. [accessed 2025-01-27].

[2] Arducam. *Arducam 1080p Low Light WDR USB Camera Module – Datasheet*. Online. Available at: https://www.uctronics.com/download/Amazon/B020101.pdf)[accessed 2025-03-12].

[3] Open Robotics. *ROS 2 Documentation Jazzy*. Online. Available at: https://docs.ros.org/en/jazzy/index.html. [accessed 2025-03-12].

[4] C. McQueen. *GSCam2 - ROS 2 Camera Streaming Package*. Online. Available at: https://github.com/clydemcqueen/gscam2. [accessed 2025-03-12].

[5] Ultralytics. *Ultralytics Documentation*. Online. Available at: https://docs.ultralytics.com/. [accessed 2025-03-12].