TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

TOMAS SEDLACEK

A REAL-TIME POSITIONING SYSTEM OF MANUFACTURING CARRIERS DEPLOYING WIRELESS MEMS ACCELEROMETERS AND GYROSCOPES

Master of Science thesis

Examiner: Professor Jose Luis Martinez Lastra

Examiner and topic approved in the Automation, Mechanical and Materials Engineering Faculty Council Meeting on 6 June 2012.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master of Science Degree Programme in Machine Automation
SEDLACEK, TOMAS: A real-time positioning system of manufacturing carriers deploying wireless MEMS accelerometers and gyroscopes
Master of Science Thesis, 80 pages, 11 Appendix pages
September 2012
Major subject: Factory Automation
Examiner: Professor Jose L. Martinez Lastra

Keywords: real-time positioning, localization, pallet, wireless sensor, MEMS, accelerometer, gyroscope, 6LoWPAN, artificial intelligence

Modern manufacturing systems face ever-increasing pressure to maximize efficiency of production processes, minimize downtime due to unexpected deviations from normal operation, and maintain agility in dynamic market conditions. Detailed, real-time asset tracking is essential for achieving these goals.

Pallets are widely-used for transporting raw materials, intermediate products, and final products in automated assembly and manufacturing lines. A sophisticated pallet monitoring system can provide possibilities for optimizing pallet routing in real time, enable dynamic scheduling changes, and historical traceability required for error diagnosis and repair. Traditionally, pallets are monitored by networks of sensors, such as RFID readers or proximity sensors to collect location data. These sensor networks are rarely dense enough to provide precise continuous data about pallet location. Real-time pallet tracking data is thus limited to recording timestamps at static checkpoints.

This thesis presents an asset-aware management tool for continuous pallet location monitoring based on event logs obtained from intelligent wireless devices embedded in each pallet. Each wireless device, equipped with a 3-axis accelerometer and a 3-axis gyroscope, provides accurate information about pallet movement. The raw sensor data is pre-processed into an event stream, which is sent to a server over a 6LoWPAN network. The software developed in this research implements an algorithm for processing event logs to determine exact pallet location using artificial intelligence techniques. Calculated pallet position can be provided to high-level enterprise systems, and to manufacturing execution systems for use in scheduling, routing, and visualization of the production line. Designing the SCADA system was also part of this thesis.

The solution was successfully deployed in the FASTory, a 12-cell light assembly line in the Factory Automation Systems and Technologies Laboratory (FAST-lab.) at Tampere University of Technology, as part of eSONIA, a European Commission-cofunded research project on using service-enabled embedded devices for realizing an asset-aware, self-recovering plant. The proposed solution demonstrates a novel approach for continuous, real-time pallet location tracking based on wireless sensors.

PREFACE

There is end to everything and my studies are not an exception. After all those long years of studying I have finally reached the point, when I must stand up on my own legs and pay off the effort of those people, who supported me in challenging quest of high education. Right to them I would like to dedicate this page to say a word of thanks, because without their love, help and support, I would never achieve to be, where I am right now.

Firstly I would like to express my endless gratitude to my parents, who worked diligently every day to give me an opportunity to study unconcernedly, who have always encouraged me towards higher goals and who have always turned every my failure into a new motivation.

I want to thank also you Ivuško, my beloved girlfriend, for your infinite patience, love and comprehension during last long three years, while you were waiting for me. But especially I want to give you may thanks my dear, because you have never lost a faith in us whenever the things didn't go right. I love you.

I would like to thank also to Professor José L. Martínez Lastra for his guidance during developing this thesis and for giving me an opportunity to become a member of his FAST research team while working on project eSONIA that founded my work. Special thanks belong also to Corina Postelnicu and Johannes for her advices and corrections during writing this thesis.

Finally my thanks belong to all my friends who made my second life here in Finland more pleasant, happier and in fact unforgettable. It was amazing time. Thank you Navid, Hector, Luis, Payman, Aleš and many other Czech friends, who I met here.

My studies are over but educating is never ending process and I will never stop. Avšak ze všeho nejdřív chci už konečně domů!

Tampere, September, 2012
Tomáš Sedláček

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
6LoWPAN	Internet Protocol version 6 over Low-power Wireless Personal Area Network
AGV	Automatic Guided Vehicle
AJAX	Asynchronous JavaScript and XML
AOA	Angle-Of-Arrival
APIS	Advanced Pallet Information System
CAD	Computer Aided Design
CCID	Community Colleges for International Development
CVG	Coriolis Vibratory Gyroscope
DPWS	Device Profile for Web Services
eSONIA	Diagnosis and Control: Towards the Asset-aware and Self-recovery Factory
EMV	Expected Monetary Value
EU	Expected Utility
FAST	Factory Automation System Technology
FIFO	First-In-First-Out
GPS	Global Positioning System
GUI	Graphical User Interface
HMI	Human Machine Interface
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
ID	Identification number
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization
IT	Information Technology
JSON	JavaScript Ojbect Notation
M2M	Machine-to-Machine
MCL	Monte Carlo Localization
MDP	Markov Decision Process
MEMS	Micro Electro Mechanical Systems
MEU	Maximizing Expected Utility
MST	Micro System Technology
MVC	Model-View-Controller
ND	Neighbour Discovery
NFC	Near Field Communication

OEM	Original Equipment Manufacturer
PCB	Print Circuit Board
REST	Representational State Transfer
RFID	Radio Frequency Identification
RPC	Remote Procedure Call
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
RTU	Remote Terminal Unit
RW	Read-Write
S1000	Product name of used Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SCARA	Selective Compliant Assembly Robot Arm
SLAM	Simultaneous Localization and Mapping
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
TDOA	Time Difference Of Arrival
TUT	Tampere University of Technology
UDDI	Universal Description Discovery and Integration
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
USB	Universal Serial Bus
W3C	World Wide Web Consorcium
WF-net	WorkFlow net
WLAN	Wireless Local Area Network
WS	Web Service
WSDL	Web Service Definition Language
WSN	Wireless Sensor Network
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language

TABLE OF SYMBOLS AND UNITS

c_O	Orientation coefficient
c_h	History coefficient
d_x	Distance in blocks
N_p	Total number of all particles
η	Normalizing factor
$P_{NR}(NR_S\,\|R_m)$	Probability that the pallet is not on right turn block and a right turn was detected
$P_{RR}(R_S\,\|R_m)$	Probability that a pallet is on right turn block and a right turn was detected
S	Old set of old particles
S'	New set of particles
ω_{nn}	Non normalized importance weight
ω_n	Normalized importance weight
U	Control vector
V	Penalty
x'	New state
Z	Measurement vector

1 INTRODUCTION

1.1 Background

When a Scottish philosopher Adam Smith more than 200 years ago published his famous work The Wealth of Nations about division of labour and resources, he would probably never believe that his idea will completely change the face of modern industry. However it took more than one century until the idea of resource division was for a first time fully transformed into reality by Henry Ford. In 1913 he managed to develop a first assembly line for a mass production of legendary car model Ford T. Both labour and resource division significantly improved the efficiency and volume of production. Most of the machines turn to be used in their full capacity and moreover there was no need for workers to know how to build a complete car anymore. They just executed a few operations related to his or her skills and shift the product further in process chain. [1, 2]

Many things has changed since that time - new technologies were discovered, there was a revolution in electronics and informatics therefore workers are being replaced by faster and more precise robots, a single modern production machine is now capable to provide many operations at one grip and overall control has become highly automated, however the Ford’s main concept of assembly line has remained until nowadays almost the same. Whether it was breakthrough Ford’s manual line, semi-automatic mass production line in 80-ties of last century or modern automatic high-customized lines, all of them have one thing in common - Logistics.

The raw material, various components and product itself must be simply transported somehow from one machine to other machine; from one workstation to another; from factory to customer and that all happen in exact time, place and volume. The more sophisticated product, the more components is needed, more operations must be done and more complex transport system has to be developed. In fact, there are two possibilities – either the product is localized due to its size or specific features at one place and material is brought to that place (for example aircraft industry or ship manufacturing) or the machines, components and other resources are localized at certain places and the product itself is transported from one station to another. In this latter scenario, the products usually move for instance on rails, rollers, belts, they hang from ceiling transport system or they move in tubes or on air-cushion. Due to their size it is an easy and efficient way of transport and it could be found in many areas of industry (for example automotive industry, food industry, chemical industry or electronics) as well as in the testbed used in this research.

During recent years there has been increased demand for mass production of high customized products like cell phones, tablets or PCB circuits generally. Their manufacturing has become highly complicated task requiring sophisticated web of interconnected assembly lines, which are capable to guide automatically a product throughout numerous workstations to produce desired final product in the most efficient way. Very common solution for transportation of these products is a manufacturing carrier called an *industrial pallet* or simply a *pallet* that is carried by a belt conveyor system.

The pallet serves as a product carrier, which provides smooth movement of product during transportation throughout whole conveyor system as well as fixed product position if necessary. The pallet is usually made from a metal frame, equipped by a convenient interface to hold product itself. As soon as the product is completed, it is taken away from the pallet and stored. The empty pallet is subsequently sent back for new material and next production cycle starts. Under normal circumstances the pallet should never leave current conveyor system. Obviously, the more complex conveyor system is developed, the more difficult is his monitoring. Conventional systems use for monitoring usually a set of sensors (capacitive, inductive or RFID sensors) distributed around the line to detect pallet presence and thus provide basic information about current pallet position. From the process control point of view is this system relatively sufficient till the pallet control operations (changing directions, entering workstations, stopping etc.) take place exclusively at sensor places. However, every product that is often held by even more valuable pallet, presents for a company a significant asset that needs to be monitored continuously throughout the process to prevent potential losses or damages. Such a monitoring tool is called asset-aware management tool and development of its techniques is currently under intensive research. Some works related to this topic are listed in Chapter 2.6, which is dedicated to the state of the art of localization and monitoring of various assets.

1.1.1 Problem Statement

Mainly during the last years, in the time, when economic crises came, the vendors tended to optimize their processes and production lines to found all possible faults, incongruities and ineffective procedures like never before. Having numerous pallets with products on top of them represents for each vendor considerable assets and thus their continuous monitoring to prevent damages or losses become one of their main objectives. Unfortunately, traditional monitoring based on a network of distributed sensor cannot simply provide desired results. There is actually no information about pallet movement or position in between those sensors. If certain pallet stacks somewhere, falls down from conveyor or it is in the worst case even taken out by irresponsible operator, the conventional monitoring system would never be capable to detect it and react appropriately. Therefore, a problem of continuous real-time monitoring of pallets became a main motivation for this research and for writing this thesis as well.

Nowadays GPS (Global Positioning System) has become a main leader in localization systems for all outdoor activities since it is a simple and reliable technology, which provides relatively high accuracy. However, as everybody knows, it is not applicable in places out of sight of satellites and therefore there is currently an extensive research for developing enough reliable and easy applicable system for indoor premises, too. There are many proposed ways among which two most significant approaches are distinguishable. First one is an orientation towards localization using wireless sensor networks (signal strength, time of signal flight, signal angle etc.), while the latter one and of which also this thesis is part of, is based on application of inertial sensors. There is more space dedicated to both approaches in Chapter 2.6 where their advantages and disadvantages are depicted.

1.1.2 Justification of the Work

The research described in this thesis was written for a Department of Production Engineering in Tampere University of Technology (TUT) as a part of European project under the umbrella of ARTEMIS JU called Embedded Service-Oriented Monitoring, Diagnostics and Control; Towards the Asset-aware and Self-recovery Factory (eSONIA). From the project name is already evident that the main objectives of eSONIA project will head towards development of tools for asset-awareness and self-recovery in production plants using embedded devices.

Therefore also this research aims to develop expedient tools for real time positioning and monitoring of production assets in form of manufacturing carriers transporting products by implementation of low-cost and low-powered wireless embedded devices. Wide interoperability of the final solution is ensured by using 6LoWPAN as a wireless communication technology that supports direct network connection by IPv6 without a need for any gateways and versatile web services as a way of data transportation. Together with other research activities within TUT eSONIA research group, there is going to be build a prototype for Asset-aware Management Tool for complete monitoring of a production line.

1.2 Work Description

1.2.1 Objectives

The primary objectives of this thesis are as follows:

1) Design and implement to wireless embedded devices an algorithm, which will generate appropriate event logs based on MEMS data describing movement of manufacturing carrier.
2) Deploy the wireless embedded devices to carriers and establish their wireless communication
3) Propose, develop and implement algorithms that utilize artificial intelligence for real-time continuous carrier monitoring and its localization within production line by using event logs generated from wireless embedded devices.
4) Design and develop a graphical user interface that allow user to:
    - Design own arbitrary map of production line
    - Real-time monitor all pallet positions
    - Monitor pallet position based on historical event logs stored in database

1.2.2 Methodology

The approaches how to accomplish thesis objectives are:

1) Establish wireless communication using given wireless technology – deploy 6LoWPAN edge routers and set correctly the communication protocols
2) Execute a sequence of measurements to collect raw accelerometer and gyroscope data, while carrier moves in conveyor system
3) Design and simulate event log generating algorithm based on measurement data and deploy it to wireless devices
4) Make a literature review to find out the optimal solution for algorithm development while taking into account an event based character of the process
5) Design the key elements of overall software architecture and implement it (run server, create web service endpoints, develop GUI for monitoring and testing purposes and develop their mutual communication)
6) Make as soon as possible first algorithm prototype and verify its properties in order to keep agile process.
7) Based on the first results either continue or revalidate used algorithm – repeat this test-validation approach as many times as possible. If necessary apply different algorithm.
8) Verify the results
9) Polish the graphical user interface

1.3 Thesis Outline

The content of this thesis is arranged as follows. Chapter 2 is dedicated to provide reader necessary theoretical background to understand the fundamentals of methodologies and technologies that are used throughout this thesis. Chapter 3 describes chosen methodology that is going to be used for developing the final solution and outlines the process of its implementation. The implementation proper is going to be explained in Chapter 4, while results and their discussion can be found in Chapter 5. The thesis is concluded by Chapter 6 that is dedicated to final conclusions of the thesis.

2 THEORETICAL BACKGROUND

The aim of this chapter is to present a necessary theoretical background and literature review for reader to understand the content of this thesis and used technology. Because of large diversity of used technologies, the chapter is attempted to be arranged in three sections. First section of this chapter is dedicated solely to artificial intelligence as a research field that utilizes methods for autonomous solution of relatively difficult tasks without human interaction and with capabilities of self-learning. The second section attempts to get a reader familiar with theory related to used devices and communication technologies. Thus, a theory of micro system technology is presented, of which sensors are used for pallet movement detection, followed by data transfer technology called 6LoWPAN that is used for wireless communication between the sensors and outer world; and theory of Web Services as way of data transfer and presentation. In the last, the third, section a current state of the art is outlined to show the link between the thesis work and the present edge of research in this field supplemented by their comparisons.

2.1 Artificial Intelligence (AI)

In this chapter Artificial Intelligence is going to be introduced, as a multidisciplinary research field that attempts to develop autonomous tools for solving various tasks. At first a brief introduction is going to be given to get reader familiar with concept of AI. The following subchapters will be dedicated to a description of *intelligent agents* as a core element of any AI solution together with description of *environment* as a place or problem the agent is dealing with. Finally the chapter is concluded by two subchapters that present a minimum theoretical background of two AI disciplines that are going to be used in the further research. The first is related to a *decision theory,* as a theory describing a methodology for an agent how to decide “well” in particular situation and *localization* as a technique for finding agent’s position both in known and unknown environment and which is used mostly in branch of mobile robots.

2.1.1 Definition and introduction to AI

From time immemorial, the human mankind have tried to understand a process of *how humans think* i.e. how people can perceive, understand, predict and manipulate things surrounding them, in order to define the key element, which distinguishes people from animals. However, the attempt of Artificial Intelligence goes even further; it does not only try to understand the process itself but also it tries to build new intelligent entities to copy human intelligence. [3]

The history of AI is relatively new and goes back to 50's of last century along with a birth of information technology. Since that time the AI has matured a lot and nowadays encompasses many research disciplines. However, the more branches exist, the more perspectives on AI objectives arose and the more difficult is to determine unified proper AI definition now. Nevertheless, as is mentioned in [3], all AI definitions generally vary along two main dimensions:

- **Thought process and reasoning** – it is a human-centred approach that focus to measure success in terms of fidelity to human performance by implementing empirical science i.e. hypothesis and experimental confirmation.

- **Rationality** – rationalist approach which involves a combination of mathematics and engineering to fulfil some goal or to optimize a process.

Throughout the rest of the thesis, a focus will be given especially to the rational approach of AI since it corresponds to a subject of this thesis – pallet localization represents a goal, which a certain AI algorithm must fulfil in order to increase the possible output (accuracy). In following chapter a rational or intelligent agent is going to be introduced as the central of intelligence in rational approach.

2.1.2 Intelligent Agents

According to [3] an **agent** is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. The simplest example is a human agent. The human agent has eyes, ears or tongue as sensors to perceive the environment and hands as actuators to interact with it. More abstract illustration can be shown in case of software agent in virtual environment. The software agent perceives data from real environment through a set of keyboard inputs, files or web streams and acts on the environment by displaying screen, writing data or controlling some real processes. In AI terminology a term **percept** refers to perceptual input and **percept sequence** to complete history of everything what the agent perceived.

If the agent is inserted to its environment, it generates a sequence of **actions** based on percepts it received, to change its state in the environment as can be seen in the Figure 2.1. To determine whether the state was from overall perspective desired and the agent performed well, there must be implemented some objective **performance measure**. Usually the performance measure is set by an agent designer, because agents alone have tendency for deluding themselves i.e. in case of failure they attempt to divert the reason to a fact that it was not actually their goal. With all this terminology it is now possible to provide a **definition a rational agent** as follows:

For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has. [3]

In addition to above mentioned properties, the agent is also characterized by an **environment** in which the agent operates. All together it gives to every agent a unique set of properties called **PEAS**, which stands for **P**erformance, **E**nvironment, **A**ctuators and **S**ensors.

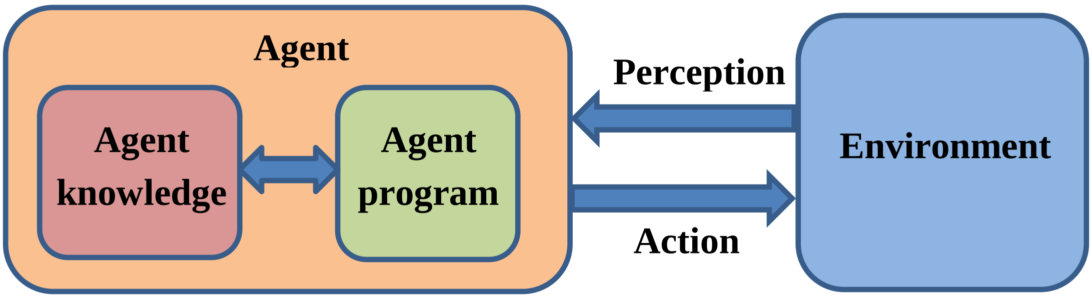


Figure 2.1 *Schema of an Intelligent Agent*

2.1.3 Agent environment

In general, the reason of building an intelligent agent is that the agent actually symbolizes a solution for certain problem and the problem is task environment. For instance in case of chess agent, the environment will be chess board and chess opponent or in case of automatic vacuum cleaner agent, the environment would be a house, where the agent cleans, a dust and house occupants. Naturally, there are many environments where agents can operate but fortunately according to [3] it is possible to categorize them into several groups. Each kind of environment then determines the appropriate agent design and techniques for its implementation.

Fully observable vs. partially observable – if agent's sensors have constant access to complete state of the environment so the agent has all aspects that are relevant to the choice of action, the environment is called fully observable (e.g. chess board, a map for navigation agent). On the other hand the environment is partially observable, and that is the most common case, when the agent does not have enough information for making an action choice for example due to missing sensors or noise in sensors (e.g. automated guided vehicle in city – traffic, passengers, construction sites; or the automatic vacuum cleaner – unknown distribution of dust, closed doors etc.)

Deterministic vs. stochastic - if agent’s next state of the environment is completely determined only by current agent state and agent action, then the environment is said to be deterministic otherwise it is stochastic. The environment can be partially observable and deterministic at the same time. For instance battleship – players don’t see opponent’s ship location, but as soon as they select a target to fire, they either hit or miss.

Episodic vs. sequential – in episodic environment the next agent’s action is not influenced by its previous action. Each agent perceiving and then performing is divided into single action (e.g. defect search – discovery of a new defect is not related to previous discoveries). In contrary, sequential environment is characterized by sequence of actions. Current agent’s state is a result of sequence of history actions (e.g. chess).

Static vs. dynamic – if the environment can change in the middle of agent’s next action decision, then the environment is dynamic (driving a car), otherwise it is static (battleship). There is also a term **semidynamic**, when the environment doesn’t change, but agent measure performance change with the time of decision like for instance in chess.

Discrete vs. continuous – to determine whether the environment is discrete or continuous a closer look must be done how the time is handled and how an agent perceives and acts. For instance battleship is a discrete environment since it has discrete amount of states and agent’s percepts and actions are discrete while in case of driving a car the perception is continuous – angle of steering wheel and speed make agent’s action also continuous thus the environment is continuous.

2.1.4 Agent program

So far agents were described in a way how they interact with their environment but this chapter introduces a way how do they behave and how do they decide based on sequence of percepts what to do. The responsible “brain” is called **agent program** and its design is the main job of AI. The goal of the agent program is to take the current percept as input from the sensors and return an action to the actuators. Based on [3] there are four basic kinds of agent program distinguishable, which encompass the principles underlying almost all intelligent systems.

Simple reflex agents – the simplest kind of agent, where decision is made exclusively on current percept according condition-action rule while ignoring the rest of percept history. Condition-action rule is set of conditions (if something happen) and appropriate actions (do this). For instance if there is a green in traffic light – continue driving, if there is a red colour - stop the car. Simple reflex agents are very easy to develop, but their intelligence is very limited.

Model-based reflex agents – this kind of agent maintains some sort of internal model of “how the world works”. The model is either already in-built or built from percept history that agent collects during his existence. The presence of model is for agent in partially observable environment very profitable since it helps agent to keep track of the part of the world it cannot see right now.

Goal-based agents - compare to the two abovementioned agents, where decisions were made either according specific situation (e.g. every right junction turn a car left) or stored knowledge respectively (e.g. on this junction turn a car right, since on the right there is no such traffic in this hour), these agents attempt to reach a specific goal for making any decision. Thus, all agents’ activities head towards the goal accomplishment while utilizing some sort of planning or search engines. (e.g. goal-based agent might be a taxi that needs to utilize map navigation to find the path to some exact location)

Utility-based agents – however goal alone might not to be enough to accomplish a task with high-quality output. Sometimes it is necessary to add to a decision process factors such as time-consumption, energy-consumption, reliability to optimize their final output. That is what exactly utility-based agents do – they try to increase the output, the utility, of their work. Their utility function maps a state onto a real number, which describes the associated degree of high-quality output.

2.1.5 Decision theory

Decision theory is a study of principles and algorithms for making correct decisions in order to allow arbitrary agent to achieve better outcome with respect to its goals. [4] Decision theory does not attempt to be a fortune teller, but rather a procedure that produces based on all available information the best possible logical decision and minimizes a possibility of unfavourable outcome. Thus the theory may be regarded as formalization of simple common sense using mathematics as unambiguous language, in which decision problem can be represented.

Decision theory is very wide spread discipline that contains many methods and algorithms but its full description is unfortunately far out of the scope of this thesis. In this chapter there are going to be presented two major decision theory branches – **Simple Decisions** and more advanced **Sequential Decisions**. Where the main focus will be given primarily to Simple decisions, since there are going to be later on used in algorithm of this thesis.

A. Simple Decisions

Sometimes there are known as a *Single shot decisions* or a *One-shot decision,* which expresses the actual meaning even more precisely. Simple Decisions are used in episodic environment, where the focus is given solely to utility of one single action and the utility is known. Therefore an agent decides always in such a manner that it does the best what is possible in that moment, regardless if it is good also in long-term perspective. According to [5] there are six major steps which should be followed in order to make a successful simple decision and of which meaning will be describe in next paragraphs. [6]

1) Clearly define the problem
2) List all possible alternatives
3) Identify the possible outcomes
4) List expected utilities for all combinations of alternatives vs. states of nature
5) Select one of the mathematical decision theory models
6) Apply the model and MAKE THE DECISION

The easiest way how to describe working principle of the decision theory, concretely of simple decisions, is to show it on a simple example – concretely on case of businessman dilemma which is as follows.

There is a businessman, who is going in the morning to the work and wearing a perfect suit. However, he has a dilemma, whether to take an umbrella with him or not. If he takes the umbrella and it will not rain, then the umbrella will represent whole-day burden for him. On the other hand, if he let the umbrella at home and it will be raining, his suit gets wet and it needs to be cleaned, what costs him money and time.

In this example there are two decision **alternatives** (having the umbrella / not having umbrella), that a businessman as a decision maker can perform and two **states of nature** (raining / not raining) that represent uncontrollable future events. Generally, alternatives can be either open, if the range of choices can possibly expand (businessman example – he can take also a car for instance) or closed, when there is limited list of available choices (e.g. yes/no questions).

A decision is naturally attempted to be done in rational way, thus in such a way that the resulting output is always maximized (e.g. to do not unnecessarily carry heavy suitcase or pay for washing). The question however arises, how to express a value pattern for each alternative, i.e. how to assign a goodness to its future output? There are two most common ways - relational and numerical representation.

Relational representation uses for value pattern description a set of relations such as: better than, not worse than, the same as and many others. Finally, according individual preferences the best alternative is chosen. Unfortunately, relation representation in not all the time unambiguous and so called cyclical preferences might appear, when relations close a loop and there is no best alternative. In case of businessman, the representation might look like as follows.

- *It is better to have an umbrella when is raining than not do have it.*
- *It is better to do not have an umbrella when is not raining than to have it.*
- *To have an umbrella when is not raining is not worse than to do not have an umbrella when is raining.*

In contrary, the **numerical representation** attempts to assign numerical values to each of alternatives. The numerical values are in decision theory more appreciated, since they are easy to calculate with. However, it is not easy to always transform everything to exact values. In case of businessman example, the wet suit can be washed for certain amount of money, which represents a monetary value. But what kind of value has the alternative: "to have unnecessarily heavy suitcase with umbrella when is not raining"? Thus, so called **utility** is presented to reduce all values into one single entity describing goodness in term of numbers. The way how the utility will be handled falls into fifth step of decision making and depends both on decision-making environment and used decision theory model.

Generally there are distinguishable three types of decision-making environments for which is defined several decision theory models. [3, 6, 7]

1) Decision making under Certainty – complete and accurate knowledge of utility for each alternative; there is only one utility value for each alternative

2) Decision making under Risk – multiple possible utilities of each alternative and each state of nature can be identified and probability of occurrence can be attach to each state of nature.

3) Decision making under Uncertainty - multiple possible utilities of each alternative and each state of nature can be identified but there is no knowledge of the probability of any state of nature.

Decision under Certainty – if all utilities are certain and known, the decision itself is reduced to simple choice of alternative with optimal utility. For instance, there are two products A and B. Product B is 15% cheaper than product A while quality of both is the same. Then the decision is clear and product B is chosen

Decision under Risk – instead of optimizing given utilities as in case of decision under certainty, the decision under risk optimizes expected utility while there is only a given probability for each utility that it might happen. The most common method is called

Expected Utility (EU) or in connection with money decision Expected Monetary Value (EMV).

Expected utility is also sometimes called more precisely as a probability-weighted utility theory that assigns to each alternative (*i*) a weighted average of its utility values under different states of nature (*j*) weighted by probabilities of those states ($p(j)$) as is shown in Eq. 2.1

$$EU_i = \sum_{j=1}^{j=n} utility_{ij} * p(j) \qquad \text{(Eq. 2.1)}$$

The final decision is done based on the principle of **maximizing expected utility (MEU)** that says that a rational agent should always choose an action that maximizes the agents expected utility and thus it is chosen alternative with the highest expected utility. The principle of EU is shown in next paragraph on the case of businessman.

If for instance it is supposed that a probability of a rain is 0.1 and not rain 0.9, then the expected utility of bringing the umbrella is $0.1 * 15 + 0.9 * 15 = 15$ and for not bringing the umbrella $0.1 * 0 + 0.9 * 15 = 16{,}2$. Therefore rational businessman would let his umbrella at home and undergo the risk to be wet.

Decision under Uncertainty

Decision under uncertainty has available utilities for each combination alternative vs. state of nature, but there are no probabilities of state occurrence and thus optimization criteria cannot be applied. Therefore, there is used one of these four decision-making models – Maximax, Maximin, Minimax or Criterion Realism. The working principle of all of them is going to be presented again on businessman example. Utility values for all alternatives and states of nature that are going to be used for method presentation are listed in Table 2.1 in simple decision matrix. [8]

Table 2.1 Decision matrix with expected utilities for case of businessman

States → Alternatives ↓	It rains	It does not rain
Umbrella	15	8
No umbrella	0	18

- **Maximax (Optimistic) –** maximizes the maximum utility for every alternative, it looks for the best what can happen for each alternative and then chooses the alternative with biggest utility – best from the best. Very similar decision as lotto player, that just see the winning price and he or she is not interested in any probabilities. Therefore an optimistic businessman would decide not to take the umbrella since: no rain + no umbrella = 18 > 15 = rain + umbrella. [8]

- **Maximin (Pessimistic) –** maximin agent always looks for the worst utility what could happen for each alternative and then from this set chooses the best one. Hence, a pessimistic businessman would take always umbrella, since the worst utility for having umbrella (8) is higher than the worst utility for missing umbrella (0). [8]

- **Minimax (Opportunistic) –** this kind of decision is based on opportunity loss. Minimax finds always alternative that minimizes the maximum opportunity loss with each alternative. It is kind of decision when if an agent looks back after the state of nature has actually occurred and says "Now that I know what happened, if I had only picked this other action instead of the one I actually did, I could have done better". So, to make their decision (before the event occurs), they create an opportunistic loss (or regret) table. In the table for each state of nature is utility by utility of all alternatives subtracted from the largest utility for that state of nature. The loss table would look like for businessman as is shown in Table 2.2. Finally, for each alternative the highest loss is taken and picked up the smallest loss out of them. Hence, an opportunistic businessman would rather take the umbrella to suitcase (loss is 10), since he does not want to risk a bigger possible loss (15). [8]

Table 2.2 Opportunistic loss table for businessman example

States → Alternatives ↓	It rains	It does not rain
Umbrella	0	10
No umbrella	15	0

- **Criterion realism (Realist) –** it is a compromise between optimistic and pessimistic decision. Criterion realism uses for determining the right alternative by choosing the highest value of weight average (WA) while using coefficient of realism *a*. The coefficient *a* is arbitrary estimated. Each utility is weighted as shown in Eq. 2.2. Maximum utility of each alternative is weighted by *a*, minimum utility of the alternative by *1-a* and any utility value for other state of nature is omitted. [5]

$$WA_i = a * (\max Utility_i) + (1 - a) * (\min(Utility_i)) \quad \text{Eq. 2.2}$$

Table 2.3 Criterion realism table for businessman example

States → Alternatives ↓	It rains	It does not rain	WA (a=0.8)
Umbrella	15	8	13.6
No umbrella	0	18	14.4

As can be seen in Table 2.3, the highest average weight in businessman example is for his second alternative and thus a realistic businessman would let the umbrella at home.

B. Sequential Decisions

Sequential decisions are decisions, where the agent's utility is not known during the moment of decision as in simple decisions, but it depends on sequence of decisions. Therefore, techniques such as planning and searching are applied to calculate whole sequence of actions that will lead to maximized utility. Very nice example of sequential decision making is chess, where the overall utility (winning) does not depend only on one particular figure movement but complete sequence of them. Three major sequential decisions are commonly used - Markov Decision Process, Partially Observable Decision Process and Game theory. Due to limited space of this thesis, only a very brief description will be given to each of them. [3]

Markov Decision Process (MDP) – MDP is applicable to fully observable environments and is based on **Markov Property**, which says that agent's action, which leads to another state, depends only on the state the agent is right now and not on the preceding states. MDP is characterized by a transition model of the environment, where the initial state is known and all possible transition from each state are known as well. Each state is determined also by a reward function describing how "good" the state is to reach the final desired state. Desired state has naturally the highest reward. Transition model or reward functions are usually invariable, but the point is to find the optimal policy, as a function that tells agent how to act on every state to yield the highest expected utility. [3]

Partially-Observable Markov Decision Process (POMDP) – while MDP was applied to fully observable environment, POMDP are used if there is an uncertainty whether the agent actually is on certain state and if the policy for supposed state will be correct. Thus POMDP works instead of agent's actual state with agent's belief state, as a state that with the highest probability represents agent state. [3]

Game theory - A multiagent decision model of optimality takes into consideration not only maximized utility, but also the interaction between agents, who are playing against each other. Game theory attempts to look at the relationships between agents in a particular model and predict their optimal decisions**.** [9]

2.1.6 Localization

Everybody knows that robots have become in recent years inseparable part of industrial production and human lives almost too. Step by step, they increase their capabilities – become faster, smarter, efficient and thus slowly take over human daily routine work. Therefore, nobody would be surprised that AI plays in their development a significant role. In fact, a robot is an excellent example of a physical agent that perceives environment with its sensors and manipulates the physical world by its effectors.

Localization is in robotics very intensively investigated problem since knowledge about where things are is at the core of any successful physical interaction of any robot. Generally, the problem of localization is in AI highly reduced to problems related to robotics. Since a localization of a robot as a physical agent in physical environment has much in common with pallet localization as a topic of this thesis, there is going to be given more space to it in the next subchapters for its further explanation.

Sebastian Thrun distinguishes in his work [10] generally three major robot localization problems. The simplest problem is a **position tracking** or sometimes called as dead reckoning, where an initial robot position is known and the problem of localization is reducing only into error compensation of robot odometry.

Bit more difficult localization problem is **global localization problem**, in which a robot does not know its initial position in advance and must find it from a scratch. As soon as it is found, one or more hypothetic positions are then followed.

Nevertheless, the most challenging problem is a **kidnapped robot problem**, in which an already well-localized robot is teleported to some different place without being informed. Hence, the robot must not blindly rely on its hypothetically found place at the time of kidnapping, but be still able to appropriately react and check other location possibilities, too. This kind of problem was intensively investigated in last decade and many new methodologies were elaborated such as Monte Carlo Localization, SLAM (Simultaneous Localization and Mapping) or Markov Localization, known as Kalman Filters. In next subchapters will be some of them presented in more details. According to [11] localization techniques fall into three basic categories.

- Behaviour-based approaches
- Dense sensor matching
- Landmark-based approaches

Behaviour-based approaches – behavioural approaches rely on the physical interaction of robot actions with the environment to navigate. Behaviour-based localization does not rely on any internal map as a central representation of world but the world is a map itself. The computational path from sensor to actuator is minimized to maintain high interactivity with the physical world. The robot's activities produce behaviours that are "layered" on top of each other in such a way, that robot localization capabilities are increasing. Behaviour-based approach is very similar to behaviour of insects and in fact is commonly used for localization of small robots with very limited computational capabilities indeed. For instance, behaviour-based robot can replay a sequence of its movements to find out the direct way to its initial position. [11, 12]

Dense sensor matching – the algorithm attempts to match dense sensor scans against a surface map of the environment without extracting any landmark features. In case of complex maps, where it is very difficult to determine proper landmarks, is this feature a significant advantage. One of the most known dense sensor matching techniques is scan matching. The scan matching is a process of translation and rotation of a range scan (ultrasonic scan, laser scan) in such a way to achieve the highest possible overlap between sensor scan map and a-priori map of environment. The result of a scan matching is a single Gaussian distribution over the environment map, that can be calculated for instance by Kalman filters. [11]

Landmark-based approaches – these approaches use landmarks, stable, recognizable features, distributed throughout the environment to make estimates of position that are combined with a model of the robot's motion. The strength of landmark-based approaches is applicability in wide range of environments – indoor or outdoor, using natural or artificial landmarks. On the other hand, it is a bit challenging to use these approaches to dynamic environments with lack of landmarks and while the sensor measurements are generally noisy. There are three major and most promising approaches – triangulation, Markov localization (Kalman filters) and Monte Carlo Localization (MLC) which are going to be briefly described in next paragraphs. In addition, a special chapter dedicated only to MLC will follow later on. [13]

> **Triangulation** depends on the range and bearing to set of landmarks and uses geometrical functions to compute a single point that is most consistent with the current location. Triangulation is probably the oldest localization techniques known from ancient times and surprisingly even nowadays many robots systems use for their navigation right this method. Triangulation may be done with low weight given to previous object positions and probably nowadays the most known application is GPS. [13]

> **Markov localization** addresses the problem of state estimation based on sensor data. It is a probabilistic algorithm that maintains a probability distribution

over the space of all hypothetical places (beliefs), where a robot might be. The probabilistic representation allows weighing each hypothesis in a mathematically sound way. Markov localization is characterized by updating all state beliefs in form of new probabilistic belief distribution (*Bel(ζ))* after every sensor measurement. This method is very suitable for kidnapped robot problem, since it always maintains also other hypothesis. However, even though Markov localization is quite efficient, it requires some extra computational time to compute all coefficients and covariance, which can be critical for some real-time localization systems. [14]

The Figure 2.2 illustrates a working principle of probabilistic localization. Initially the robot does not know where it is and thus the Bel(ζ) is uniformly distributed (Figure 2.2a). Then the robot observes a door next to it and changes its belief accordingly (Figure 2.2b). Subsequently robot moves a certain distance forward and as a result, the belief is shifted too and flattened (Figure 2.2c). Finally, the repeated observation of a door presence prompts the robot to modify its belief as is shown in Figure 2.2d. The belief now corresponds to a real location.

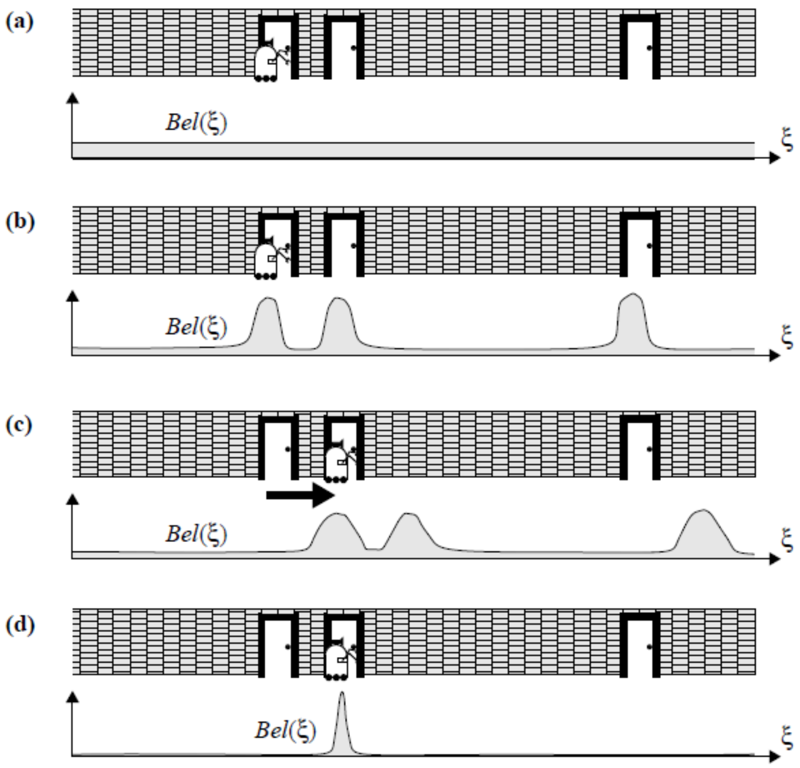


Figure 2.2 *Working principle of probabilistic localization [15]*

Monte Carlo Localization is another probabilistic localization algorithm that has been successfully used for many years in mobile robot navigation. In fact, MCL is very similar to Markov Localization. MCL in comparison to Markov localization represents the probabilistic distribution as a series of samples. The density of samples then represents the probability that a robot is at certain position. MCL, as well as Markov localization, is able to solve both global localization problem and kidnapped robot problem in very robust and efficient way.

In MCL large number of hypothetical states is initially randomly distributed around the space in form of particles. With every object move, subsequent sensor measurement is taken and the probability of hypothetical state is updated. The higher density of particles in certain state space, the higher probability the state represents the actual robot state. Because the total number of particles must be always constant and most probable states “consume” the biggest amount of particles, it logically means, that the improbable states have low or no particles at all. Although this property gives MCL very high robustness, it is also its main drawback. If there are no particles left in certain state, there is no chance at all for the algorithm to revalidate this state for potential object occurrence. This might be problem for instance in case of solution of kidnapping robot problem. One of the most common reasons is a usage of very precise sensors that do not give any uncertainty to robot movement and thus spread more the probability distribution. The simplest solutions might be to add to a sensor signal an artificial noise. [10]

The Figure 2.3 shows very nice graphical visualization of working principle of MCL for global localization problem. In Figure 2.3a, an initial phase of MCL is shown and particle distribution is equally spread around the space. As soon as the robot makes first measurement, the particle distribution is immediately updated and its result can be seen in Figure 2.3b. It is clearly visible that particles gathered mostly in two positions due to ambiguity in geometric properties of robot environment. However, as robot proceeds the environment starts to be unique and robot is successfully located as is presented in Figure 2.3c

a) b) c)

Figure 2.3 *Example of Monte Carlo Localization: a) Initialization, b) Ambiguity due to symmetry, c) successful localization [16]*

Monte Carlo Localization works based on particle filter algorithm and looks as follows:

There is a given set of particles S, control vector U and measurement vector Z. (Eq. 2.2) The set of particles consists of 1..n particles, which are associated with important weights ω.

$$Particle\ filter\ algorithm\ (S, U, Z) \qquad \text{(Eq. 2.2)}$$

Now a new set of particles S' is created according following steps (Eq. 2.3).

$$S' = \{\ \}, \eta = 0 \qquad \text{(Eq. 2.3)}$$

First of all, there is iteration from $i = 1..n$ started (Eq. 2.4), during which a particle S_j from old set of particles S is picked up randomly based directly on its important weight ω_j.

$$for\ (i = 1..n) \quad \{ \qquad \text{(Eq. 2.4)}$$

This means that particles with higher importance are chosen more frequently. Second step of the loop is a calculation of a new success state x' of chosen particle as a state transition probability using control U and that specific particle S_j (Eq. 2.5).

$$x' = p(x \mid U, S_j) \qquad \text{(Eq. 2.5)}$$

$$\omega_{nn} = p(z \mid x') \qquad \text{(Eq. 2.6)}$$

Then within the same loop a new non-normalized importance weight ω_{nn} is calculated as a measurement probability for that specific particle S_j (Eq. 2.6). A new particle with a new state x' and importance weight ω_{nn} is placed into a new set of particles S' and a loop continue in iteration until n steps is reached (Eq. 2.7).

$$S' = S' \cup \{< x', \omega' >\} \qquad \text{(Eq. 2.7)}$$

$$\eta = \eta + \omega_{nn} \quad \} \qquad \text{(Eq. 2.8)}$$

Finally there is done normalizing of each importance weight by a division of sum of all weights (Eq. 2.9 and Eq. 2.10).

$$for(i = 1..n) \quad \{ \qquad \text{(Eq. 2.9)}$$

$$\omega_{n_i} = \frac{1}{\eta}\omega_{nn_i} \quad \} \qquad \text{(Eq. 2.10)}$$

2.2 Devices and technologies

2.2.1 Micro system technology (MST)

Micro system technology (MST) or an American name - Micro electromechanical systems (MEMS) are three-dimensional (3D) structures manufactured through a set of specific micromaching techniques and reaching dimensions from micrometre- to nano-metre- scale. Such dimensions allow the MST to be embedded into various electronic devices, while keeping high reliability and low power consumption. Due to their size, the MST is usually produced in large batches, which significantly contributes to their extremely low price. Therefore, the MST enjoys high popularity nowadays and it has been extending to all areas of human activities. The application ranges from mechanical microelements needed for building microstructures and microdevices (micro gears, levers, capsules etc.), through various microsensors that can be due to their dimensions placed almost everywhere (accelerometers, gyroscopes, pressure sensors, light sensors, chemical sensors etc.), and end up with electrical circuits and numerous actuators needed for micromachining and transportation of micro particles.

Despite plentiful applications the history of MST is not that long, concerning from beginning only sensors and starts in 1960s, when accurate hydraulic pressure sensors for aircrafts were needed. However, the biggest boom of MST has started in 80s side by side with enormous growth of information technologies and automotive industry. Nowadays as it could be seen from chart in Figure 2.4, the IT peripherals (ink jet print heads, RW laser heads) cover most of all MST applications followed by automotive and consumer electronics industry, which has registered enormous growth in last 5 years. Led by smartphones, tablets, notebooks and other devices equipped by microcameras, position, light and pressure sensors, the consumer electronics has become the most developing market areas for MST todays.

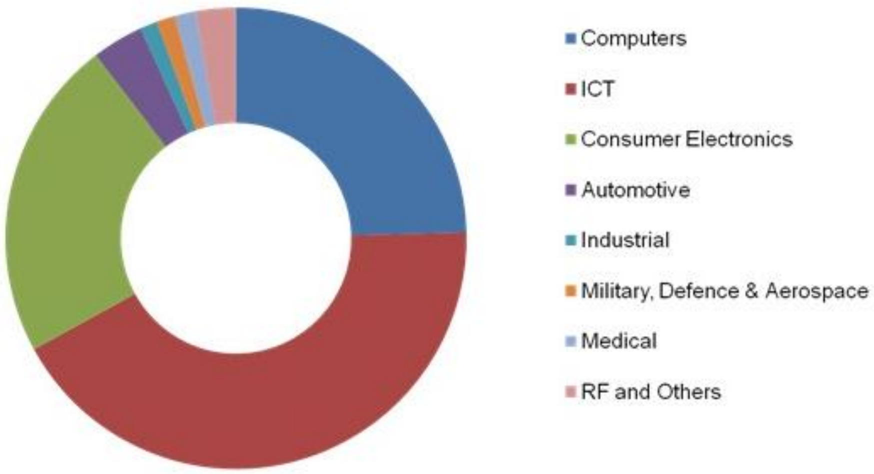


Figure 2.4 *Global MST market by application in 2011 ($10 billions) [17]*

Also the times, when a car was equipped only by a few sensors, is already passed by. Modern car nowadays consists of multitude of sensors gathering information so essential that the car would never run without them. The sensors continuously monitor hundreds of vital signs such as ignition and valve timing, exhaust quality, throttle position, intake air temperature and pressure, wheel slippage, tire inflation, steering-wheel position and many others.

As one can see there are many MST sensors available but despite this research concerns exclusively with accelerometers and gyroscopes, the following chapters will be dedicated just to them and inertia sensor generally. [18, 19, 20]

2.2.2 Accelerometers

Accelerometer is a device that measures acceleration proper of a test mass of well-defined weight. It is commonly used for motion detection as well as for space orientation. All accelerometers work in similar way. They measure a displacement (as a capacitance change between two rows of teeth, of which detail is shown in Figure 2.5) of their test mass that is mounted freely by springs and which moves according to a resultant of all forces acting on the test mass. Therefore, even if an accelerometer is let in gravitational field of Earth in steady position, the accelerometer will never measure zero acceleration, but instead the acceleration that is equal to gravitational acceleration of Earth. Hence, it is possible to determine sensor orientation in space. In contrast, there is one phenomenon if an accelerometer is in gravitational free fall and even though its speed is continuously increasing, the accelerometer will measure only zero acceleration, because it is at rest in a frame of reference in which object is weightless. Thus, accelerometers are insensitive to acceleration of gravity per se. In addition, in Table 2.4, there are listed the most significant accelerometer properties that characterize every accelerometer. [21, 22]

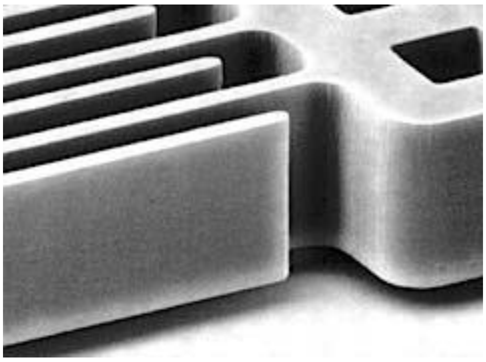


Figure 2.5 *Detail of accelerometer structure - 20µm thick single-crystal comb structure used for capacitance measurement [22]*

Table 2.4 *Accelerometer properties*

Property	Description
Range	Measurement range of each accelerometer is given in multiples of *g*, so for instance a 2g accelerometer is capable to measure acceleration in range of $-2*g \approx -20 m/s^2$ to $2*g \approx 20 m/s^2$.
Number of axes	Ability to measure acceleration in certain direction, the simplest are one-axis (1D) accelerometers having ability to detect either X, Y or Z axis, then two-axis (2D) accelerometers or the newest three-axis (3D) accelerometers that are able to detect all three axes at the same time.
Sensitivity	A minimal change in acceleration, which is detected by accelerometer

2.2.3 Gyroscopes

Gyroscope is a sensor or a device that is capable to detect angular acceleration and therefore is widely used for detection of a change in sensor orientation in space or rarely to determine its absolute orientation in space. There are many kinds of gyroscopes distinguished by their working principle. Although the chapter refers solely to MST gyroscopes, it might be appropriate to explain a working mechanism of a typical mechanical gyroscope too, due to its long evolution and easy construction which describes the working principle of gyroscope in the best way.

- **Traditional mechanical gyroscope**

  The principle of mechanical gyroscope is known for a long time although it did not serve for measurement but for amusement and religious activities. The gyroscope consists of flywheel spinning in set of gimbals, which can move around independently in respect to the flywheel as can be seen in Figure 2.6. If the flywheel is spun the angular momentum and precession movement causes the flywheel to be highly resistant against any forces that act to change its axis orientation. In simple terms if outer gimbal is twisted the flywheel still keeps the same orientation. Therefore, mechanical gyroscopes have been used for a long time in marine and aerial navigation. [19, 23]

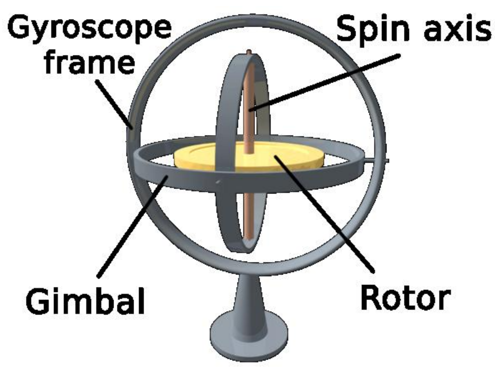


Figure 2.6 *Traditional mechanical gyroscope [23]*

- **Coriolis vibratory gyroscope (CVG) or Vibrating structure gyroscope**

  This gyroscope has become the most common type of MST gyroscope nowadays mainly due to easy construction and very low price. It does not implement any spinning flywheel as in traditional mechanical gyroscope, but on the other hand it uses a vibrating structure (a mass) flexibly mounted at least in two directions by springs or anchors. The structure oscillates in one direction as is illustrated in Figure 2.7. When the gyroscope is rotated by outer force, Coriolis force starts acting on the vibrating structure in such a way, that it shifts the structure perpendicularly to axis of vibration. This radial displacement is then measured as a change in capacitance between set of teeth on the vibrating structure and set of teeth on support. In comparison to mechanical gyroscope, the Coriolis vibratory gyroscope, due to its working principle, cannot provide information about absolute orientation, but just a change in orientation. On market there are currently available one-, two- or even three-dimensional gyroscopes. [18, 23]

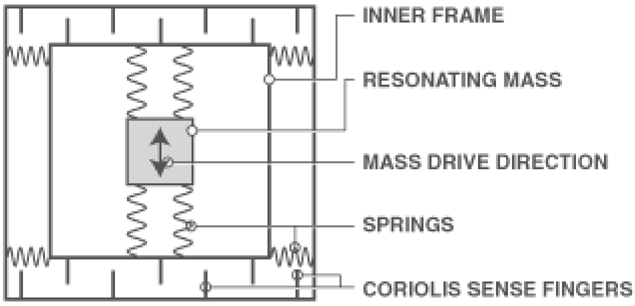


Figure 2.7 *Working principle of Coriolis vibratory gyroscope [25]*

- **Optical gyroscopes**

  Working principle of optical gyroscopes is based on fact that propagation of light beam varies in respect to a movement of conductor which leads the beam. In practice it means, that if two laser beams are sent to a long optical fibre conductor at the same time from both sides while the optical fibre is rotating, there is going to be measured a phase shift between both beams. The phase shift is equivalent to the rotation of the optical fibre. This phenomenon is known as Sagnac effect and it does not mean anything else, than a fact that the beam, propagating against rotation, have a shorter path then the second beam. Sometimes there is used a set of mirrors for beam propagation instead of optical fibre as can be seen in Figure 2.8. [25, 26, 27]

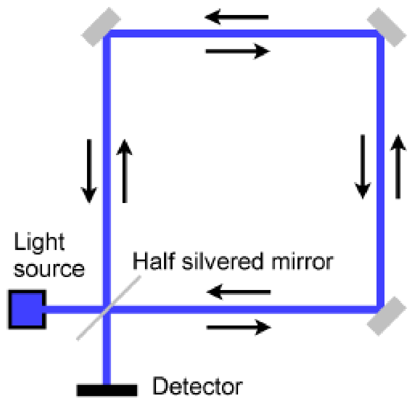


Figure 2.8 *Working principle of optical gyroscope with set of mirrors [25]*

2.2.4 Sensor components

Today a MST-based accelerometer or gyroscope is understood as a complete product that is packaged, calibrated, tested and delivered to the customer. For customer it is required to take only minimal effort for successful integration of the sensor into his or her application. Therefore, the input / output interface of sensor must be clearly specified. Nowadays there are plenty sensors available on market that can meet all kind of environmental conditions and it is only up to customer, which one to choose. Nevertheless, most of the sensors have common architecture which is shown in the Figure 2.9 and described in next paragraph.

The system consists of not only a sensor itself, but also additional components such a transducer or electronics. The role of the inner sensor is to convert input signal for instance in form of acceleration or rotation into meaningful physical objective which is transformed in transducer into an electric signal. Both the sensor and the transducer often interact with a package since they might share the same structure and thus exchange stress and heat, too. Hence, one must be aware that inconvenient external conditions (such a humidity or temperature) might significantly affect measurement capabilities of a sensor. Before the signal from transducer is processed in electronic part it is brought at first to input stage, where the signal is amplified to manageable signal. Sometimes the electronics is able to produce even a control signal that is necessary for bias setting or for operational conditioning. Actuator can serve for instance for feedback control as well as for test signals. In reality these building blocks do not have to have exact borders and their functions can overlap or be merged. [19]

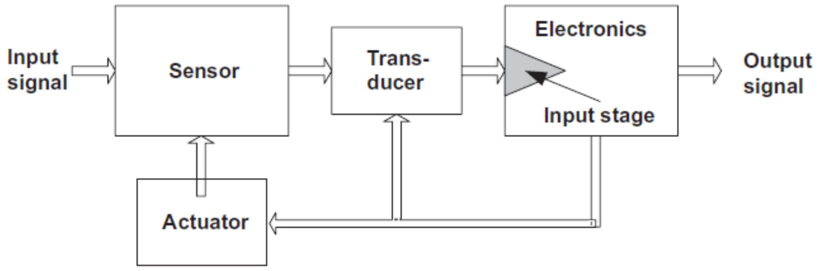


Figure 2.9 *The general architecture of a sensor system [19]*

2.3 Radio-Frequency Identification

Radio-Frequency IDentification (RFID) is a tagging technology that is gaining widespread attention by using radio waves for information propagation and having great number of advantages compare to conventional tagging methods like for instance bar-codes. The conventional tagging systems usually require direct line of sight which is their most significant drawback. In contrast, using RFID tags allows a tag to be hidden and its information is still easily accessible.

RF transmission such a radio waves can transmit information across long distances and RFID is not an exception, especially when they are powered. In fact, the way of powering divides RFID technology into 3 groups – active, passive or their combination. Active RFID tags extend their reachability by using own source of energy and they are used for instance in pet tagging, when is desired to find it in range of kilometres. The passive tags rely on energy received from interrogating devices, which they transform for transmission of their own information.

However, sometimes long distance reachability is not desirable. In case of sensitive data, stored for instance in passports, personal identification cards or credit cards a different technology called Near Field Technology (NFC) is used. NFC is a subset of RFID technology that reduces a reach of a tag up to few centimetres. The tags are usually passive, since they do not need long reach and even shielded to insecure that only appropriate device can read its data. [28]

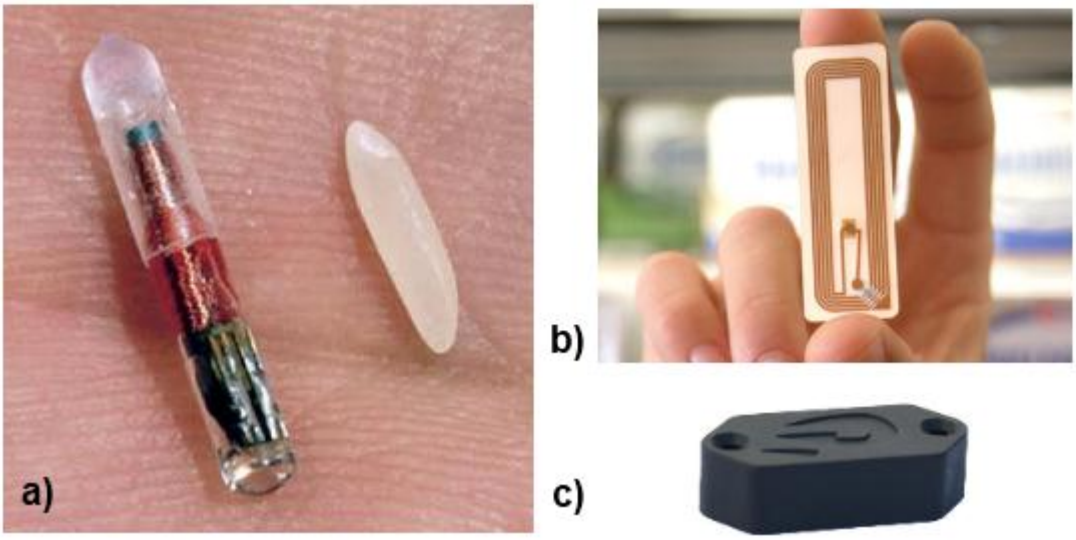


Figure 2.10 *RFID tags used for - a) animals tagging [29], b) product tagging in shops [30], c) industrial tagging (robust and mounting holes) [31]*

2.4 6LoWPAN

The abbreviation *6LoWPAN* refers to a set of standards for using IPv6 over low-power processing-limited embedded devices over low-bandwidth wireless networks. Although this definition might sound for somebody unclear and confusing, it stands for a relatively young and simple technology for wireless communication between small electronic devices. Probably the biggest impulse for development of such technology was recent increase in industrial system complexity which leads to a growing need for more process data and causing higher interest in various sensors. Flexibility needs have guided a shift from hard wiring towards wireless networks of distributed sensors. Side by side with new technologies there was also effort to standardize a communication among them so many communication protocols were developed since that time such a Bluetooth, ZigBee or 6LoWPAN. [32]

For this research 6LoWPAN was chosen, because it was the first protocol on market, which supported direct IPv6 addressing. 6LoWPAN actually brings IP (Internet Protocol) directly down to small, low-cost sensor devices. The IPv6 utilization is key factor for creation of so called “Internet of Things”, which allows every single thing in universe to have its own unique address. Moreover, the use of IP-based protocols is very convenient because they are very well known technologies that do not require any modifications in the existing network infrastructure. Devices can be directly connected to other IP networks (for example Internet) without translation gateways or proxies, which can be seen in alternative communication systems using e.g. ZigBee [33], Machine-to-Machine (M2M) [34] or Bluetooth [35]. This significant downside restricts their application without additional gateways for use only in small-scale local ad-hoc networks.

However, there is a strong trend in recent years among abovementioned protocols which converge also towards Internet-based approach. The working principle of 6LoWPAN will be more described in next paragraph. [32, 36]

2.4.1 6LoWPAN architecture

6LoWPAN architecture as shown in Figure 2.11 relies on islands of IPv6 stub networks of low-power wireless area networks (LoWPANs). Note: The terms 6LoWPAN and LoWPAN are often used interchangeably. In this paper however the 6LoWPAN will refer to general technology and LoWPAN to a specific network or node. A stub network is a network which is capable of receiving or sending IP packets but does not act as transit to other networks. Each LoWPAN network might consist of router nodes and host nodes. These nodes differ to each other only in way whether a proximal node takes a responsibility of message redirection between distant nodes and edge router, which one would be otherwise for the distal node out of reach. Edge router is a device that routes traffic in and out of the LoWPAN, while handling 6LoWPAN compression and Neighbour Discovery for the LoWPAN. If a LoWPAN is connected to IPv4 network, it is an edge router, which handles smooth interconnectivity. In case of Ad-Hoc network the role of edge router is given to arbitrary node which fulfils at least the most essential functions for smooth network running. [32, 37]

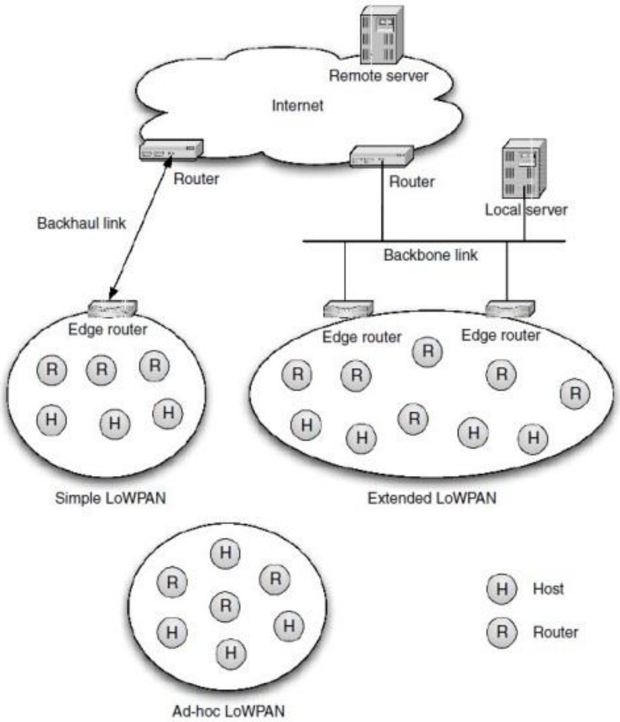


Figure 2.11 *6LoWPAN architecture overview [37]*

Three major types of LoWPANs are distinguishable, based on their architecture. [32]

- Simple LoWPANs, which are connected to other IP network through one LoWPAN Edge router.
- Extended LoWPANs use multiple edge routers within a single network connected by a backbone link (e.g. Ethernet).
- Ad hoc LoWPANs are not connected to any IP network (e.g. Internet); they work without any infrastructure, one of its routers takes on the network responsibilities and serves as a simplified edge router.

Instead of making and managing complex route tables for network communication (e.g. in ZigBee), 6LoWPAN uses so called Neighbour Discovery (ND). ND is a basic mechanism of IPv6 responsible for Router Discovery, Address Resolution, Duplicate Address Detection, Redirection and Prefix and Parameter Discovery. All these functions simplify registration of node to an edge router of any arbitrary LoWPAN and hence allow free movement of nodes throughout the LoWPAN, between edge routers or even between whole LoWPANs. [32, 36]

An ideal application of the 6LoWPAN in industrial environment would be a network of sensors distributed around the factory, collecting data not only in the raw form of digital or analogue inputs but also having capability to process data to give more meaningful information such a temperature, humidity, light intensity, power consumption etc. Wireless technology provides a flexible monitoring solution, allowing the sensors to be attached to mobile machines such as AGVs, fork lifters and obtaining information not only related to the sensor actual measured value but for instance according to a strength of wireless signal also to its position. [38, 39]

2.5 Service-Oriented Architecture and Web Services

Service-Oriented Architecture (SOA) is a set of principles and methodologies for building autonomous, agile and interoperable systems. In other words, SOA is a framework that allows business systems to expose their functionalities as a set of granular, loosely-coupled services with well-defined, standard-compliant interfaces that can be utilized across various business platforms. SOA is not new technology and in the past there were many technologies utilized to build loosely-coupled architectures such as COBRA or DCOM. Nowadays, however the easiest and most widespread approach is usage of Web services.

Web Service is defined according to World Wide Web Consortium (W3C) as a *software system designed to support interoperable machine-to-machine interaction over a network.* [40] In simple words the Web Service is a service that a machine (Web Service provider) can provide to other machines (Web Service requesters), regardless of all implementation details. The service itself can represent arbitrary activity from weather forecast, account statement to algorithm calculations. The service interoperability is completely ensured by Web Service interface.

Each Web Service interface is described in a XML-based machine-processable format called WSDL (Web Services Description Language) that explicitly specifies a set of available operations with described input and output parameters. Mutual communication between machines and Web Services is ensured primarily by exchange of SOAP messages. However, before there will be given more detailed description of the SOAP protocol and WSDL, there will be outlined the basic Web Service architecture in next subchapter as a basic structure how to reach the mutual communication. The theory of Web Services is concluded by introduction of DPWS (Devices Profile for Web Services) protocol as one of the most common standard tools for enabling Universal Plug and Play (UPnP) philosophy in world of Web Services. [41, 42]

It might be appropriate to mention that due to the scope of this thesis the following subchapters will provide only essentials of Web Services and SOA but for those, who would be interested in current state-of-the-art of Web Services especially in factory automation, there is a nice work related to Semantic Web Services written by professor Jose L. M. Lastra available here [43].

2.5.1 Web Service architecture

Existence of the Web Service alone would be pointless, if it would not be found by any requester. It would be similar to a situation when a manufacturer produces some product only to storage it in warehouse. As well as the manufacturer needs a dealer to distribute goods to customers, Web Services need own Web Service broker to broadcast their possibilities. A way how the Web Service broker is implemented is presented in typical Service Oriented Architecture in Figure 2.12.

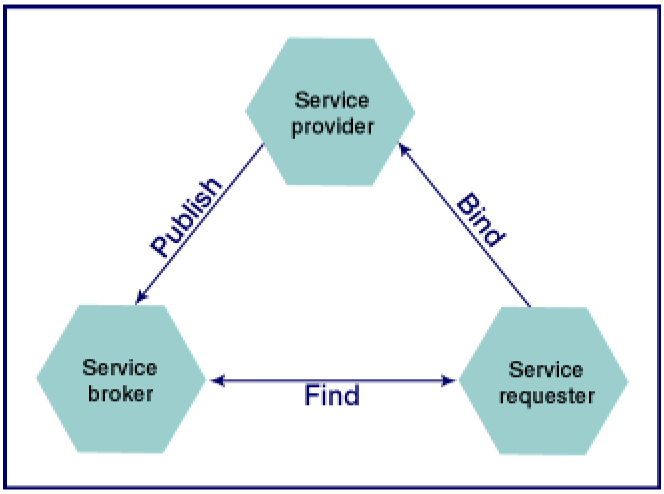


Figure 2.12 *Service Oriented Architecture [44]*

If a Web Service wants to provide certain service, it publishes the service to Web Service broker, which serves as a manager of all published services. The way how the service is published and later found is described within Web Server broker using Universal Description, Discovery, and Integration (UDDI). UDDI is a registry that provides searching algorithm for desired service in database of published Web Services according the requirements, which were sent by Web Service requester. Therefore the Web Service self-description is so valuable property. If Web Service requester searches for a service, it negotiates at first the conditions with Web Service broker, who finds for him the optimal service provider and its description sends back. Subsequently the requester binds with the found provider and uses its service. Generally a Web Service requester can be at the same time a multiple Web Service provider. [41, 44]

2.5.2 SOAP

Abbreviation SOAP, defined originally as Simple Object Access Protocol, is a protocol that specifies an exchange of structured information in the implementation of Web Services.

SOAP provides basic messaging framework upon which Web Services can be build. The reason is that SOAP relies always on top of other application layers, which ensure necessary operations such a registry, transport or security and therefore SOAP can be truly lightweight compare to other protocols. The most common application layer the SOAP uses is SOAP-over-HTTP using HTTP as an application layer, but SOAP-over-UDP is not an exception, too. [45] SOAP is formatted using XML (eXtensible Markup Language) as a wide-spread language characterized by extensibility, platform independency, self-description and machine-readability.

In the Figure 2.13 is shown a basic structure of a SOAP message consisting of three main elements: the envelope, the header and the body. The SOAP Envelope is a root element in every SOAP message, and contains two child elements – SOAP Header and SOAP Body. SOAP Header is an optional sub-element of the SOAP Envelope and it is used to pass application-related information while SOAP body is mandatory and contains intended information for recipient of the message. [42, 46, 47]

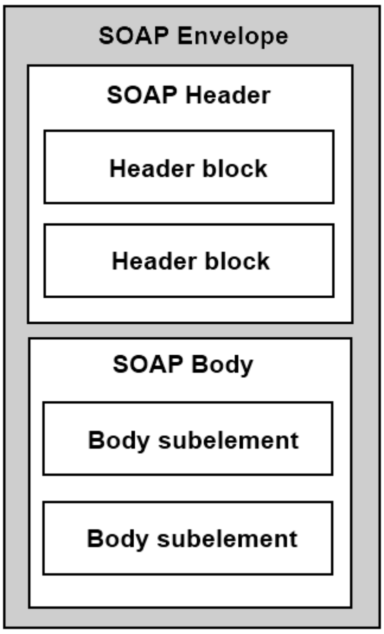


Figure 2.13 *SOAP message structure*

2.5.3 WSDL

The Web Services Description Language (WSDL) is XML-based language developed by W3C that fully describes interface and functionality of a Web Service, in order to be used by other machines without having any preliminary information about the service. Currently there are released two versions of WSDL that mutually differ mostly in different terminology and their comparison is shown in Figure 2.14. Since the second version (2.0) is the latest one, the further explanations will refer to this version primarily.

Each WSDL file (term for a WSDL document describing certain Web Service) in order to fully specify given Web Service must contain exact set of specifications. Therefore, according to W3C definition, a WSDL file defines **service** as collection of network

endpoints. In WSDL, the abstract definition of endpoints is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions such as **interface**, which is abstract collection of **operations**. The concrete protocol and data format specifications for a particular endpoint establish a reusable **binding**. Specifications summary is listed and more described in Table 2.5. [42, 48, 49]

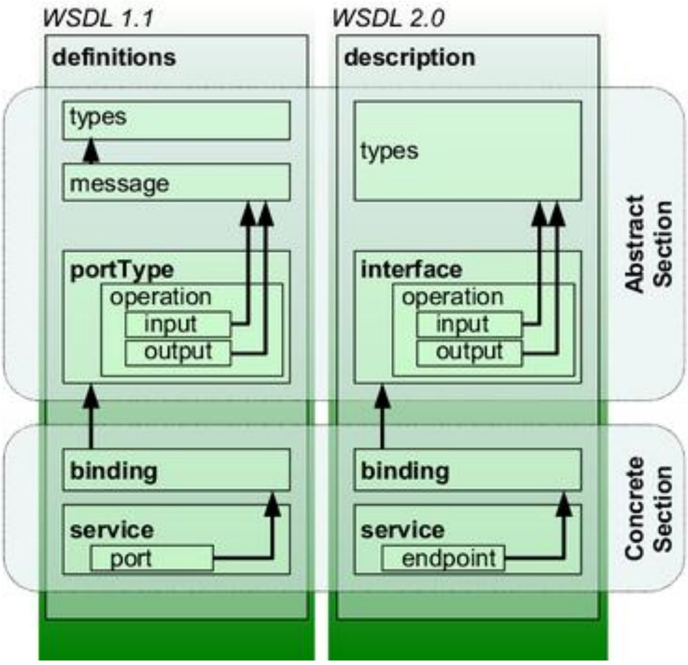


Figure 2.14 *Comparison of WSDL structures [48]*

Table 2.5 *Description of WSDL 2.0 specifications*

WSDL 2.0 Term	Description
Description	The root element of WSDL file that defines service name and namespace
Service	Collection of related endpoints
Endpoint	Defines address or connection point to Web Service. Often HTTP URL string
Binding	Specifies a SOAP binding style (RPC/Document), transport protocol for the Web Service and endpoint
Interface	Defines Web Service and all operations that its endpoints provide together with necessary messages
Operation	Defines SOAP actions and message encoding as well as the input and output parameters for the operation
Types	Describes the data in form of XML Schema

2.5.5 DPWS

Device Profile for Web Services (DPWS) is a protocol introduced by OASIS [50] that defines a minimal set of implementation constraints to enable secure WS messaging, easy and dynamic discovery, service description and eventing on resource-constrained devices. DPWS makes use of existing WS protocols and adds several extensions or constrains that are called DPWS specifications to enable Web Service based communication in embedded devices. Application of DPWS utilizes Universal Plug and Play philosophy (UPnP) where DPWS-compliant devices are after introducing to a network automatically discovered, described and immediately ready to use. [42, 51, 52]

As can be seen in Figure 2.15, from a SOA perspective, the DPWS-compliant device is an entity that hosts other services. The service is a software system that exposes its functionalities to outer world by interchange of messages. The messages utilize always SOAP protocol for transmission concretely either SOAP-over-HTTP or SOAP-over-UDP. The device plays primarily a role of resource of metadata about hosted services. The hosted services are externally visible to clients and on top of that the DPWS protocol specifies a set of in-build services that each device must implement such as

- Discovery services: self-advertising and discovering other services
- Metadata exchange: access to metadata of other hosted services
- Eventing and Subscription services

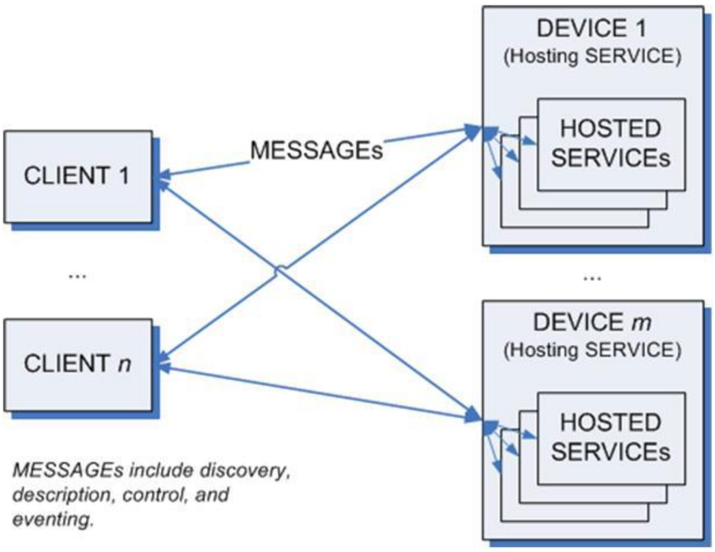


Figure 2.15 *Architecture layout of DPWS clients, devices and services [53]*

In the Figure 2.16 there is shown common DPWS stack with all transport protocols and all DPWS specifications such as WSDL, SOAP, WS-Policy, WS-Addressing, WS-Discovery, WS-Eventing or WS-Transfer. However, an explanation of these specifications is out of the scope of this thesis and can be found for instance here [50]. Nevertheless the Figure 2.16 presents also a stack of 6LoWPAN protocol to demonstrate their mutual similarity. From the figure is clear that both protocols use the same network and transport protocols i.e. IPv6 protocol and UDP/TCP transport protocol and therefore it is very easy to utilize Web Services in the 6LoWPAN communication.

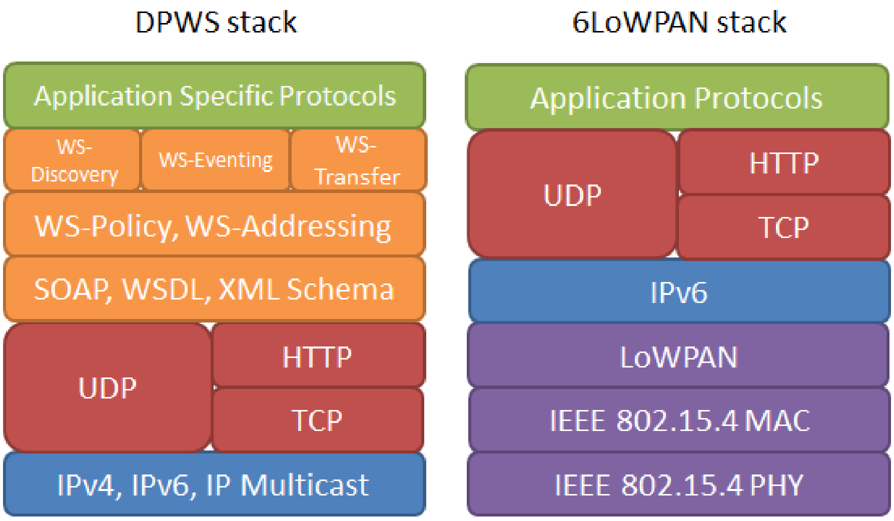


Figure 2.16 *Comparison of DPWS and 6LoWPAN protocol stack [32, 54]*

2.6 State of the art in real-time pallet localization

Although the production engineering was in industry always in the centre of interest and both companies and academic researchers from all over the world invest many resources to its development, it is astonishing that only a few of the research studies published in recent years were actually related to localization of pallets within a production line. Not that there would not be an interest in issues related to localization problems, but the vast majority of research prefers to localize more "interesting" objects such as human beings or mobile robots. Nevertheless, since a contribution of their technologies might be easily applicable for industrial environment, these works were included into a literature survey as well.

According to the survey, in way to find the present edge of industrial research in this field, it was discovered that there are currently two areas of main research stream interests – localization using wireless sensor networks and localization based on inertia sensors. In next two paragraphs both techniques are going to be described in more details.

2.6.1 Localization based on wireless sensor networks (WSN)

During the last years localization based on various wireless communicators and sensors networks has enjoyed raising popularity not only in academic circles. Small dimensions, low costs, long battery life, simple communication protocols and other characteristics of modern wireless devices directly predestine their wide application range and deployment in great amounts. However with rising sensor volume, there is also a rising need for their localization and management. Currently, there are available many proposed algorithms of sensor localization based on different technologies, but their detailed description is far out of the scope of this thesis. Therefore in next paragraphs only a brief introduction with useful references to most known techniques is going to be listed to outline possible opportunities. More extensive summarization of available WSN localization methods can be found for instance in [55, 56, 57, 58, 59].

- **Angle-of-arrival (AOA) measurements**

AOA is characterized by propagating signal angle measurement. The receiver antenna, usually of very small dimensions, is rotated electronically or mechanically, while it receives the sensor signal. The signal strength is evaluated either by phase shift or by measuring signal amplitude. The angle, from which the strongest signal is detected corresponds directly to the angle the sensor is located towards receiver. Naturally there are more receivers installed around and the final location is calculated based on their relative measurements. It is an easy method that nevertheless requires rotating element influencing negatively overall size of the sensor. A work related to this method can be found in [60].

- **Distance related measurements**

Distance related measurements are primarily based on measurement of propagation time between transmitter (sensor) and receiver (base station or sensor); and can be divided according to a way of propagation into three main groups: [61]

  - One-way propagation time measurements – it measures the difference between the sending time of a signal at the transmitter and the receiving time of the signal at the receiver. Biggest drawback of this method is a must for absolutely synchronized clocks in both devices, which makes this method less attractive.

  - Roundtrip propagation time measurements – it measures the difference between the time, when a signal is sent by a sensor and the time when the signal returned by a second sensor is received at the original sensor again. Significant advantage is that there is no need for precise synchronization since the same clocks are used for measurement, however processing time on second sensor must be taken into consideration and thus this delay must be subtracted. Processing time could be known in advance by tests or sent side by side with the localization signal.

  - Time-difference-of-arrival measurements (TDOA) – this method calculates an unknown position of transmitter as a time difference between the transmitter and set of receivers with known positions.

- **Received signal strength (RSS) measurements**

  RSS measurements estimate the distance between neighbouring sensors from the signal strength measurements. This technique is bases on standard feature found in majority of modern wireless sensors, a received signal strength indicator (RSSI). This method is very attractive since it does not require any special hardware, thus the sensor can be smaller, lighter, consumes less energy and is more cost effective. Drawback of this method is increased sensitivity to disturbances such as reflection, diffraction and multi-path effects. This method is mainly used in location estimation in WLANs but in recent years it has found a way also in localization of wireless sensor networks. Related works can be found in [61, 62]

In summary, there are many available techniques how to determine position within wireless sensor network and some of them were presented in this chapter. In reality, the localization algorithms based on AOA and propagation time measurements have better results and provide higher accuracy than algorithms based on RSS. However, this higher accuracy is at expenses of higher investment costs.

Probably one of the most extensive works related to localization in industrial environment is a thesis work of Jaacán N. M. Valdez [38], where the author tries to implement real time localization of production pallets populated by wireless sensor by signal strength. In his thesis IEEE 802.15.4 protocol was used for a wireless transmission with 6LoWPAN technology on top of it. Another work related to pallet localization are interesting works of Sarah Spieker and Christof Röhrig [63, 64], who aim to improve warehouse management workflow by continuous position monitoring of euro-pallets (wooden standardized pallet used in warehouses or for goods transportation) within the warehouse using strength of Wi-Fi signal. In another work [65] that is not directly related exactly to industrial automation, authors aim to develop an interesting solution for quasi continuous localization of wireless sensors with respect to long battery life. In fact, they propose a technique for sensor tracking while the sensor is in sleep mode to safe battery by forecasting its trajectory.

2.6.2 Localization based on data of inertial sensors

Localization using data from inertial sensors such as accelerometers or gyroscopes is nowadays almost mature discipline used primarily in navigation and localization of mobile robots. There are plenty of research works proposing different localization techniques suitable for certain environments, movement conditions, sensor types or processing capabilities of computational units. Therefore most of the works are related just to development of new mathematical models for noise reduction and accuracy improvement. Some works related to position tracking can be found for instance in [66, 67, 68], where are shown experiments related to localization of mobile robot and pedestrian using inertial sensors. Nevertheless, there is a lack of real applications especially for industrial purposes in general.

2.6.3 State of the art vs. the topic of this thesis

Based on the research survey it is evident that a topic related to localization is between researchers still very attractive. Their main interest is split into two sections. While the first one is related to localization based on wireless sensors, the latter aims to improve localization algorithms based on signal from inertial sensors. However, in case of the topic of this thesis an attempt to categorization is not unambiguous at all. Although, according to the data source it inclines to the localization based on inertia sensors, there are a few significant reasons that make this topic unique.

Firstly, vast majority of the current research works related to inertial sensors use for localization algorithms direct signal from accelerometers or gyroscopes. The continuous sensor signal is easily processed, quantities such a speed or acceleration are evaluated and appropriate localization is determined and calibrated. However, in case of this research, there is introduces brand new approach for object localization, based only on a discrete amount of event logs describing object movement. In this research the raw sensor data are preprocessed already within embedded device and only if the sensor significantly moves over some threshold value, an event log carrying movement information is generated and received by localization algorithm. Of course, this development would never be feasible, if very important assumption is not declared. The object is an industrial pallet, which moves in predefined and known conveyor system. Nevertheless, based on the survey, it seems that such a research scenario is still unexplored, which makes its development even more challenging.

3 RESEARCH METHODS AND MATERIALS

3.1 Signal preprocessing and event detection

Nowadays the wireless embedded devices equipped by modern MEMS sensors do not have to bother with analogue-to-digital conversion of the sensor data any more, since the sensors include in their packages already an embedded A/D convertor. Nevertheless, another type of conversion is desired. In recent years there has been a significant effort to shift also a low-level data preprocessing directly into embedded devices. As a result, such “smart” devices are able to convert a stream of raw digital sensor data into a set of discrete event logs containing rich and meaningful data instead. This approach brings a significant advantage to all consumers, since they are not required to have any knowledge, how the data were actually processed and generated.

The wireless devices used in this research are however not equipped by any preprocessing mechanism and so the algorithm development was a part of this research, too. Thus the prime goal of the algorithm was to *extract information of significant pallet movements from a stream of raw digital data coming from the sensor.*

The choice of appropriate data mining method, which would be able to extract desired information, was significantly influenced by low processing capabilities of their chips. Therefore as a most convenient solution, a method called **pattern matching** was chosen.

Pattern matching is a simple algorithm that checks a sequence of data for presence of constituent of some pattern. The pattern is mostly known in advance and describes such a signal behaviour that corresponds to known event (a change in amplitude, a certain shape, a phase shift). Therefore, whenever a sensor signal matches with certain pattern, the corresponding event is generated. The methodology of implementation of pattern matching goes as follows.

1) Record raw data of all sensor axes, while pallet moves throughout the conveyor system and note down all significant pallet movements
2) Study thoroughly the records and try to distinguish a special signal pattern for each pallet movement.
3) Design an easy algorithm, which would be able to recognize the patterns from raw signal and simulate it on recorded data.
4) In case of success, implement it to chips of wireless devices.

3.2 Localization algorithm using Artificial Intelligence

Artificial Intelligence disposes nowadays of many suitable localization decision-making algorithms to determine and track position of intelligent agents. This topic was more deeply described already in Chapter 2.1, from which was clearly evident that there is no universal localization algorithm, that would cover all localization scenarios and the choice of the proper algorithm is matter exclusively of agent capabilities, environment and overall task utility. Hence, to reduce the possible amount of algorithms for this research in order to choose the most convenient one, a definition of the research state in terms of AI must be done at first, i.e. to ask and answer questions:

- *how is an agent represented in this research*
- *what is his environment*
- *what is his overall task*

3.2.1 Agent

An agent in this research is going to be a virtual entity representing a real pallet in the conveyor system. One agent represents always one pallet even though there might many pallets in the system introduced and therefore their number is going to be always equal to number of pallets. Agent state is visualized in virtual map and in fact it attempts to represents the real pallet location.

Agent's **perception** is entirely discrete and ensured by receiving event logs from two sources – inertial sensor and RFID sensor. The inertial sensor is embedded in wireless device attached to the pallet and it is going to provide information about pallet movement such as left, right, start, stop and others. In contrary, RFID sensors, distributed in the conveyor system, inform the agent, when its pallet (the pallet, which the agent represents) passed certain location in the conveyor system.

Agent's **action**, as a response to perception sequence, is an adequate change of agent state visualized in the virtual map. The agent must be fully **utility-based agent**, of which goal is going to be a self-localization within the conveyor system in the most efficient way. The efficiency as a **performance measure** will in this research represent accuracy with which the agent copies actual pallet position.

3.2.2 Environment description

The research environment is represented by an arbitrary conveyor system, in which each agent needs to localize its position. The agent does not know its position in advance and once the position is known and agent is moving within the system, there is no way how the location can be proofed. While pallet moves, there are many factors influencing its movement (stoppers, other pallets, operators, defects etc.), but agent can perceive the movement only by discrete event logs that may or may not come. Even though the agent's current state is a result of previous agent's actions, the next agent action is depended only on current state and the action and thus has episodic character. Therefore in summary the research environment is **partially observable, stochastic, discrete** and **episodic.** Moreover the environment is **dynamic**, since the agent moves and new event logs are perceived, while the agent tries to distinguish the next most probable state.

3.2.3 Environment model

Since the environment is only partially observable, the agent must make noticeably bigger effort to keep track on its movement. In case of absolutely unknown environment, the agent must discover its world alone from scratch, make own map of the world and then more or less rely on it. This approach requires agent to have relatively good sensors and sufficient capability to distinguish surrounding environment. However, from agent's designer point of view, it is highly desirable to make agent's life as easier as possible. Such approach contributes among others not only to computational time reduction but mainly to more accurate results.

One of the common solutions might be to equip agent already from beginning by the map or model of world, if it is known and available. This approach will be used also in this research, since it is supposed that the layout of used conveyor system is fixed and unchangeable throughout the monitoring. Because of the discrete environment in this research, the developed map can be fully **grid-based** i.e. the world is divided in cells of fixed size. In grid map, a resolution and a precision of estimated state are fixed beforehand.

Knowing and developing of a map of the conveyor system in advance is in this research actually a key element because agent suffers by a lack of convenient sensors. With discrete event logs containing solely information of a change in agent movement in terms of left or right are absolutely insufficient for localization in unknown environment.

3.2.4 Appropriate algorithm selection

Based on abovementioned assumptions there were chosen finally two most promising AI algorithms that would be capable to localize pallet within the conveyor system and that would fit to research conditions, too. Next paragraphs are going to attempt to explain the major reasons why each of particular algorithms was used.

The first algorithm was chosen from localization algorithms. In AI, as is described in Chapter 2.1.6, there are available three localization approaches – behaviour-based, dense sensor matching and landmark-based approach. While behaviour-based approach is absolutely unsuitable for conditions of this research, since an agent can never understand the world around him based solely on the events used in this research, the scan matching could be theoretically used. In this case an operator would need to teach every agent which kind of event sequence corresponds to each section of conveyor system and the agent would then tries to match the sequence with learned scans. However since the conveyor system can consist from several identical conveyor sections, e.g. workstations, the algorithm would be “confused”. Drawback is also the need of a process of learning, what makes this approach in summary less preferable. More promising is landmark-based approach especially Markov Localization (ML) and Monte Carlo Localization (MCL). In principle both are quite similar only ML uses Gaussian distribution whereas MCL discrete distribution of particles. Because the environment is grid-based, the MCL is preferred and used as the first final algorithm for this thesis.

As the second algorithm was chosen the algorithm, that implements fundamentals from the decision theory. The main idea is that agent that is representing a pallet will decide about its next new state (a position) right after it receives an event log. Generally the decision theory offers two possibilities – simple decision and sequence decision. Sequence decision is not suitable for pallet localization since the environment is episodic and there is no final goal, to which a concrete sequence of actions would lead. On the contrary the environment requires each decision to be one-shot decision which attempts to increase momentary agent’s utility after each incoming event log. To develop proper algorithm it is necessary to determine whether the decisions are made with certainty, with risk or under uncertainty, what is represented by alternatives and state of nature and finally to build the appropriate model accordingly.

The alternative for each decision will be an agent’s new state – an exact position in grid-based environment. There are going to be as many alternatives as possible agent’s states whereas there will be only single state of nature – the agent’s observed perception of movement in form of event log. If the event log was created, then there is no uncertainty or any risk that it might have happened, it just happened and thus the decision is made under certainty. Nevertheless the problem of the decision algorithm will be to compute appropriate utility for each alternative with given state of nature and current agent’s state.

In summary, Monte Carlo Localization and Simple Decision-making algorithm will be applied for pallet localization. The way how both algorithms were implemented and realized is described in next chapter while a discussion about their results can be found in Chapter 5.

4 IMPLEMENTATION

All research experimental tests were realized in modern laboratory called FASTlab in premises of faculty of Production Engineering of Tampere University of Technology in Finland. The FASTlab is a relatively new laboratory which can boast a newish production line called FASTory. The production line provides excellent conditions for various experiments including this research. In next subchapters the production line will be introduced more in detail as well as pallets that are used for product transportation from station to station and of which localization is the objective of this research. Moreover there will be introduced also embedded wireless devices, which are equipped by accelerometers and gyroscopes, and which provide the main source of data for this research.

4.1 Production line

4.1.1 Description

The abovementioned production line which is shown in Figure 4.1 is currently located in FASTlab in Tampere University of Technology but a just few years ago it served as a final assembly line for electronic products in one Finnish company. When their production stopped, they transported the line to FASTlab and it underwent some modernization. The most significant change was unambiguously a line transformation for research activities. Therefore, electronic assembling was replaced by simpler activity of cell phone drawing and so the line is not called assembly line anymore but just production line. The reasons for transformation are obvious; there is no need to bother with buying raw material as well as with storing assembled electronic devices. Currently the production line provides excellent environment for laboratory experiments where the newest automation technologies can be applied and immediately tested.

At first glance the production line seems to consist of twelve almost identical workstations (WSs) mutually connected by a conveyor system. However, in detail look one can distinguish that the first and seventh workstation significantly differs from others. While the majority of workstations (WS2, WS3, WS4, WS5, WS6, WS8, WS9, WS10, WS11 and WS12) serve for drawing operations, the WS1 is responsible for raw material loading (a sheet of paper) and final product (drawn cell phone) unloading; and WS7 is a buffer that serves as a temporal storage for pallets. In a time this research was done and this thesis was written the pallet buffer was still under maintenance and thus for all the time out of order but for completeness it is going to be mentioned here as well since it was and soon will be again a part of the production line.

Each workstation apart from the buffer is equipped by the same 4-axis SCARA robot (SONY SRX-611), while the buffer has just a pneumatic manipulator for lifting and storing a pallet from a conveyor to prepared shelfs and vice versa. All workstation are further equipped by conveyor system, pen feeders, safety system and RFID/NFC readers. Their proper function and communication with outer world is governed exclusively by smart remote terminal unit (RTU) called S1000. S1000 is a Web Service based controller that is manufactured by Inico Technoloigies company. There are several S1000 deployed per workcell and each of them is responsible for control of different segment of workcell – robot, conveyor system or pen feeders.

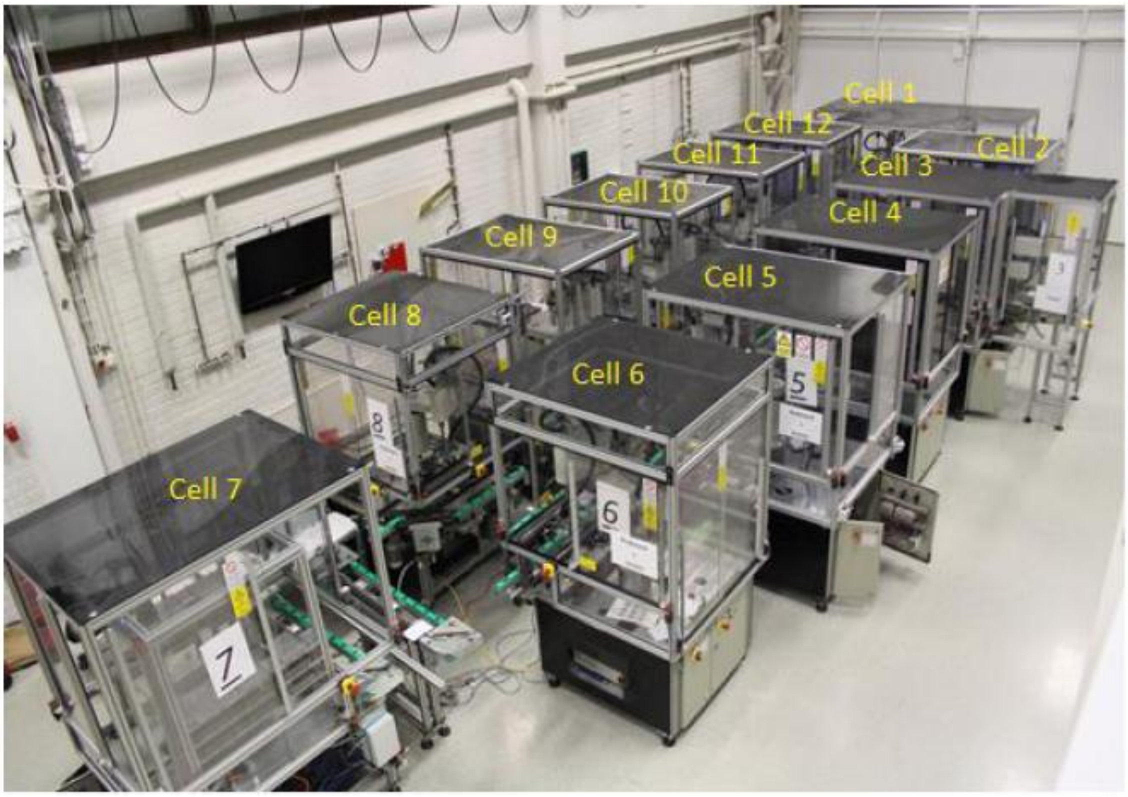


Figure 4.1 *FASTory line with labelled workstations*

4.2 Production operations

The production of drawings starts in workstation 1, where a robot equipped by a special gripper with suction cups inserts an empty sheet of paper to available pallet and covers it with a metal frame to prevent paper slide. Then a pallet is released and it moves via conveyor system to particular workstations. The conveyor system is circular and thus allows pallet to circle around the line as long as its drawing is fully complete. As soon as the pallet reaches entrance of particular cell, the system must decide whether the pallet will be in the cell processed (something will be drawn) or bypassed to other workstation. There are several scenarios listed in Table 4.1 according which the system decides, what to do. Generally, if the robot of certain workstation is idle and capable of required operation, the pallet is accepted to enter the workstation; otherwise it is by-passed to next workstation. In Figure 4.2 are shown both possible directions of the pallet movement within one workstation.

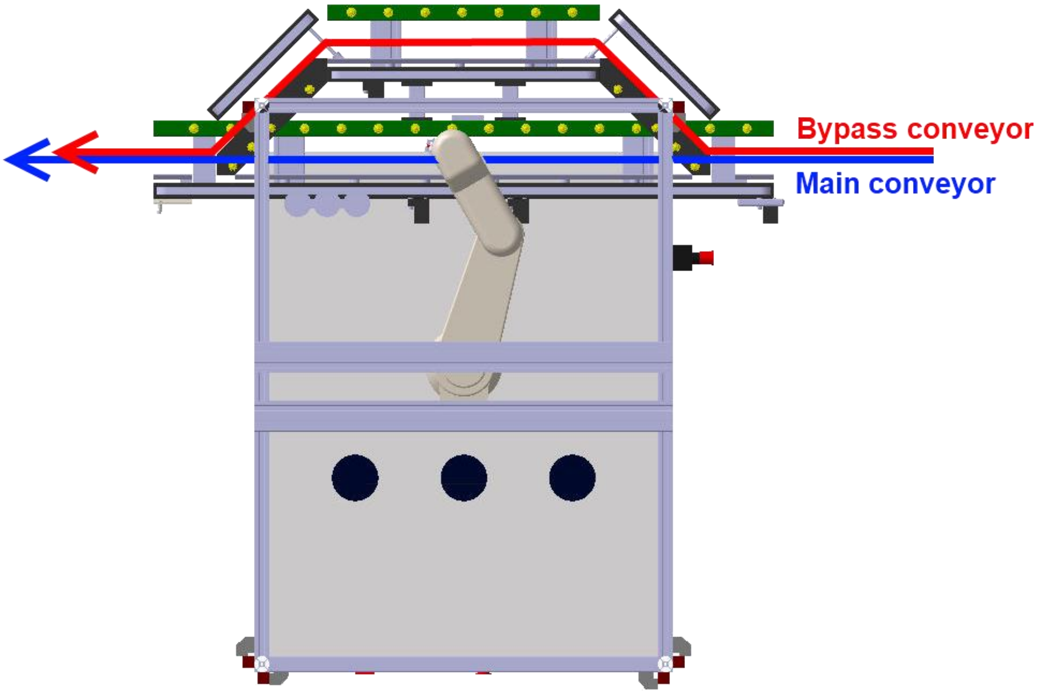


Figure 4.2 *Workstation with main conveyor and bypass conveyor marked*

As soon as the pallet undergoes all required operation steps and drawing is ready, it moves again to workstation 1, where the robot takes the finished drawing and put an empty paper sheet back in its stead. If there is no need for another drawing, the empty pallet goes to buffer, where is taken away from the conveyor system.

The final drawing which should represent a final product consists of three elements – frame, screen and keyboard. For each of those elements there are three different versions (varying in shape) in three different colours, what gives 9 combinations for each element and 9^3 = 729 combinations in total for the final drawing.

Table 4.1 *List of scenarios for the production line*

#	Scenario
1	All workstations can draw all types of elements in any colour
2	Each workstation can draw only certain element in any colour
3	Each workstation can draw all types of elements but only in one certain colour
4	Each workstation can draw all types of elements in any colour but only one element per pallet.

4.3 Pallet

A pallet is a structure of different sizes and shapes designed for product transportation. Usually the term *pallet* refers to a standardized and universal wooden structure used in industry for storage or transportation of heavy and over-sized material out of the factory. However, in production and automation industry the *pallet (*known also as *manufacturing carrier* or *industrial pallet)* refers to smaller structures that transport products within a scope of production line. Their movement is highly automated and usually provided by complex conveyor system. The conveyor system can use for pallet movement either running belts, spinning rollers or moving chains and others. Pallet design is tightly coupled with the product they are supposed to hold; in vast majority they are made to order and therefore a bit expensive.

In this research there are used brand new pallets designed by Álvaro Resúa Rey which were deployed in summer of 2012. The pallet is especially designed for transportation of a sheet of paper of size A6 along the conveyor system of production line described in Chapter 4.1. The conveyor system uses for pallet movement a complex of running belts which carry away the pallet. The pallet shown in Figure 4.3a consists of three parts.

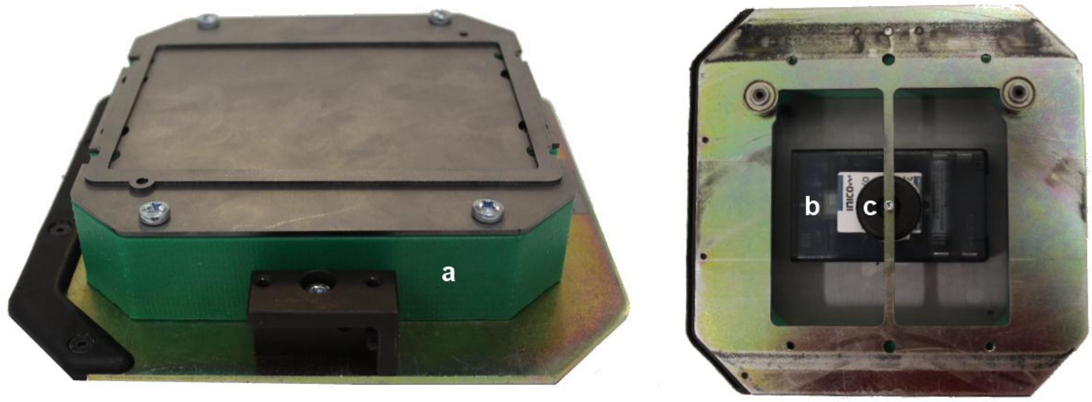


Figure 4.3 *a) Pallet, b) embedded wireless device c) RFID tag*

The lower part is made from a thick steel plate and serves as an interface between conveyor system and pallet. It must be both durable because of the wear of belts and heavy enough to provide the pallet necessary stability. The middle part is just an empty plastic box that serves as a room for various embedded wireless devices attached to the pallet from the bottom. Finally, the upper part is a detachable metal frame for keeping the sheet of paper in fixed position, while pallet moves or a robot draws. The right orientation of the frame in respect to pallet is ensured by central pins and mounted by set of magnets. The pallet has beside those three main listed parts also a rubber bumper in the front and a lock on its side. The lock serves for stopping pallet in a fixed position on the conveyor by extensible pneumatic piston, while the conveyor is still running under the pallet.

4.4 Devices

In order to detect pallet position while it moves throughout the conveyor system, certain sensors must be naturally installed to provide desired information. Since the pallet is still in motion there were only two reasonable ways of its detection. There could be either machine vision applied to track the pallet continuously or each pallet can be equipped directly by small wireless sensors. While first solution would probably provide very nice results, it is very expensive, because it requires installation of many cameras compare to the latter solution, which is significantly cheaper, since the prices of MEMS sensors are very low nowadays and does not require any special hardware installation.

4.4.1 Wireless embedded device with MEMS sensors

In this research wireless embedded device, produced by INICO Technologies, was used to determine the pallet position. The overall view of the device is shown in Figure 4.4 while in the Figure 4.5 is the device open and there are shown its particular components. The main control unit of the device is a powerful ATMEL ZibBit chip with dual antenna and wireless IEEE 802.15.4/ZigBee OEM module capable of broadcasting in 2.4GHz frequency band. [69] The chip is governed by Contiki-OS, an open-source operation system for low-power embedded devices, which is providing necessary capabilities for wireless communication and signal processing. The device is equipped by a 3-axis accelerometer and a 3-axis gyroscope which provide complete movement information of the device in a space. Additional sensor characteristics could be found in Table 4.2. The device is powered by two AA batteries and can work for long time due to its low power consumption. The chip properties allow using a 6LoWPAN protocol for a wireless communication with outer world that is described more in Chapter 2.3.

Figure 4.4 *Wireless embedded device equipped by accelerometer and gyroscope*

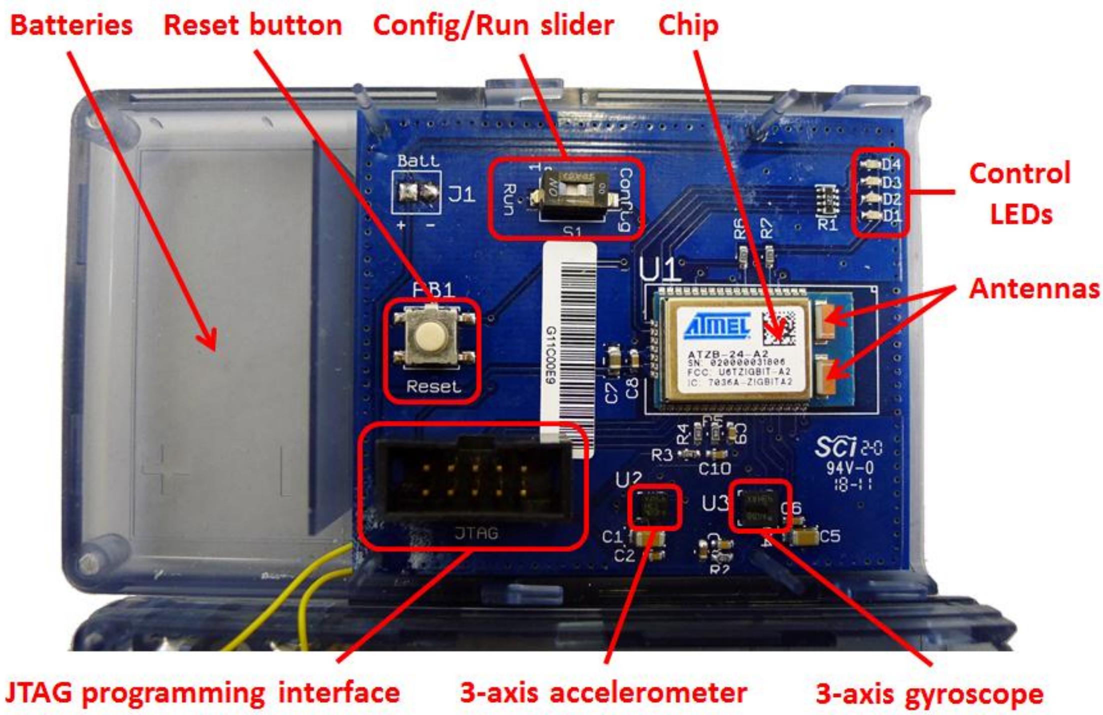


Figure 4.5 *Components of wireless embedded device*

Due to native characteristics of 6LoWPAN, where an emphasis is on low-power and low processing operations of executing device, there is used exclusively UDP (User Datagram Protocol) for a transportation layer of communication. However, in this research there was an attempt to provide sensor information as a web service using Device Profile for Web Services (DPWS). Therefore data are packed into messages according to the following standards: WSDL (Web Service Descriptive Language) for describing functionality of web service, SOAP (Simple Object Access Protocol) for specification of exchanged information and XSL schema as a set of rules, according which a message is coded. A message carrying certain information is then in a form of web service sent using HTTP and TCP transport protocols.

Nevertheless the change in transportation protocol caused significant increase in data overheads. Problem is that 6LoWPAN is designed for quick exchange of low amount of data so in case of bigger amount of data, the message must be divided into more data packets which are sent separately. In reality it means to send for the same amount of data instead of one data packet, four data packets, which represents 400% increase in communication traffic. With TCP usage another fact arises – retransmission of lost packets which again unfortunately increase a chance of network overloading.

Table 4.2 *Characteristics of sensors*

Characteristic	Accelerometer	Gyroscope
Type	LIS3DH	L3G4200D
Vendor	STMicroelectronics	STMicroelectronics
Measured axes	3-axis	3-axis
Scale	±2g/±4g/±8g/±16g	250/500/2000deg
Digital output interface	I^2C/SPI	I^2C/SPI
Size	3 x 3 x 1 mm	4 x 4 x 1.1 mm
Output	16 bit	16 bit
Supply voltage	1.71 V to 3.6 V	2.4 to 3.6 V
Other	Free-fall detection Sleep mode FIFO Temperature sensor	Integrated low- and high-pass filters Sleep mode FIFO Temperature sensor

4.4.2 RFID (NFC) tag detection

To efficiently control the drawing process, the production line needs to have a capability for pallet recognition. Therefore, each pallet is equipped by a passive NFC tag which stores unique identification number of the pallet. The tag is placed in the bottom of middle part of the pallet as might be seen in Figure 4.3c and while the pallet moves its identification numbers is read by NFC readers (sensors) distributed across the production line. The NFC reader, shown in Figure 4.6, is connected by serial link to S1000, a smart RTU that broadcasts data of the reader to all subscribers via Web Services. The reader is a commercially available device characterized by fast reaction, low price and simple interface that allows obtaining data in binary format. In following list more reader parameters can be found.

- Serial interface. Baud Rate either by default 9 600 bps or 115 200 bps
- USB interface for power supply
- CCID-like (binary) frame format
- Read/Write speed up to 424 kbps
- Build-in antenna for contactless tag access with card reading distance up to 50 mm
- Supports NFC (ISO/IEC18092) tags – reader, card emulation, peer-to-peer modes
- Extended support for various Mifare cards
- Built-in anti-collision system (only one tag is accessed at one time)

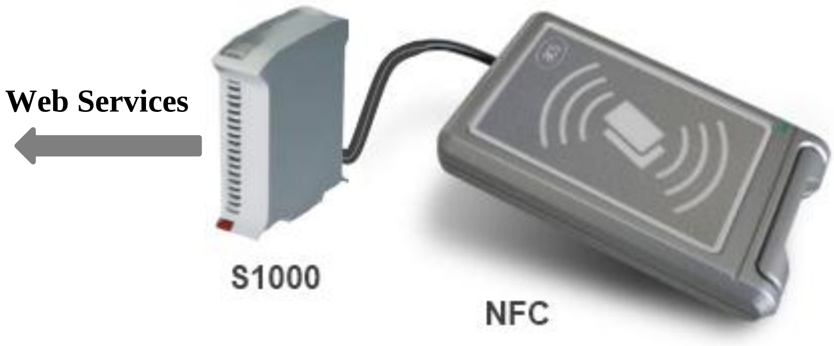


Figure 4.6 *NFC Reader ACR 120 S-B and schema of its information flow*

4.5 Data transfer architecture

The ability of sensors to measure pallet movement would be pointless if its data does not reach the appropriate interested listeners. Thus it was necessary both to build a system for mutual communication between wireless sensors and their listeners and to ensure their data will be secured against tapping, too. The overall data transfer architecture of experiment testbed is shown in the Figure 4.7.

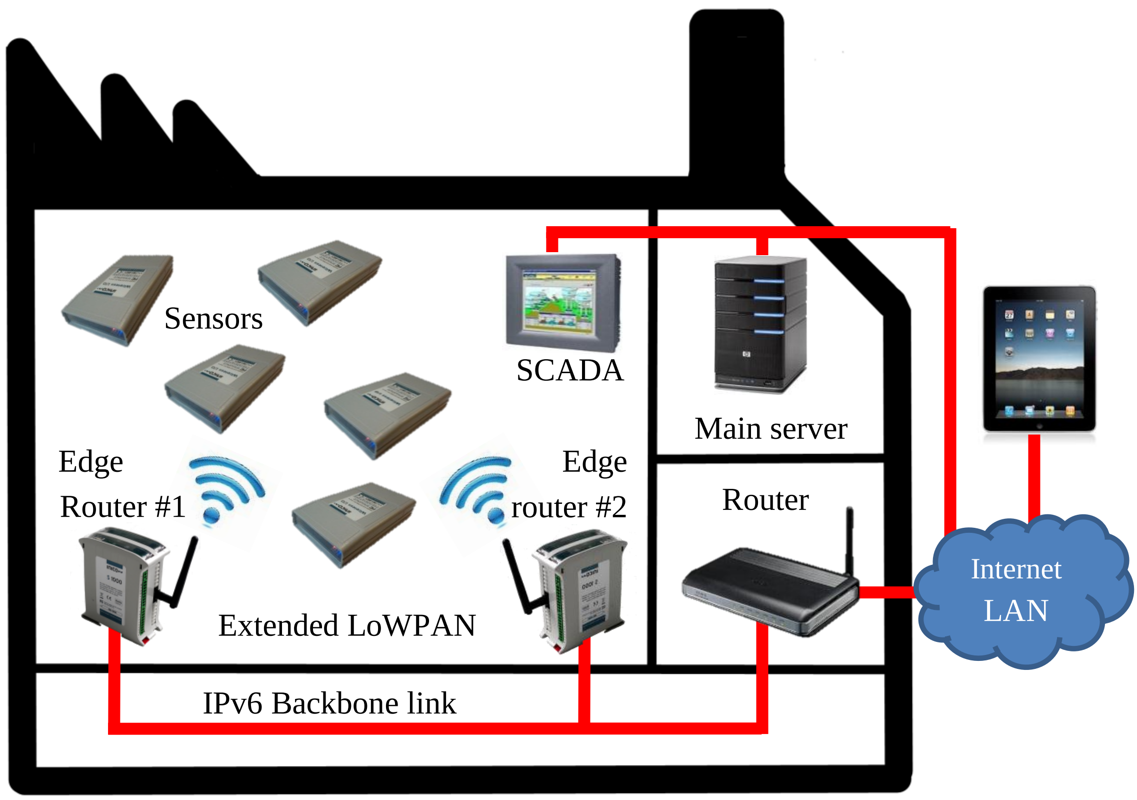


Figure 4.7 *Overall data transfer architecture*

Firstly there was an Extended LoWPAN network established in laboratory to ensure wireless communication with sensors nodes. Thus, two edge routers were installed around the production line. As an edge router a popular S1000 controller, produced by Inico Technologies, was used. However, this controller is in comparison to S1000 controllers used in control of production line, supplemented by a W1-Z add-on wireless module as can be seen in Figure 4.8. While the wireless module is responsible for all 6LoWPAN duties such network initialization, communication, maintenance or security, the S1000 itself serves only as a gateway to IPv6 network without having any processing responsibilities. Both edge routers are connected to the same backbone IPv6 local area network and to internet respectively.

Figure 4.8 *Edge router: S1000 controller with wireless module*

Sensor nodes are directly accessible by their unique IPv6, which allows easy access and quick configuration. Each sensor node consists of a list of subscribers, which will receive further the sensor data. To minimize the network traffic only one subscriber is preferably used. The prime subscriber in this research is a main server called InfoStore, which is responsible for all data collection and processing. SCADA or other visualizing device can display either raw sensor data by direct sensor accessing or processed data from the main server.

4.6 Overall software architecture

As a core platform for server software architecture an open source application framework called Spring was chosen due to its scalability, interoperability and availability of many additional modules for various services. For purposes of this research Spring MVC (Model-View-Controller) module was selected as a design pattern for internal logic and also because this framework provides hooks for extension and customization for web applications and RESTful Web Services. According to this pattern the overall software function is allocated into three main areas of responsibility – Model, View and Controller.

- **Model** – model stores all data structures, objects, algorithms and represents whole computational power. In this research the model keeps whole process model i.e. conveyor maps, pallet information and locations, history of event logs. It also stores and applies all localization and other algorithms. Finally the model is also responsible for keeping track of list of clients and amount of data that were sent to those clients.
- **View** – view is a main input/output interface through which the user can interact with the process. A view both generates the graphical output for the user to describe states of various objects and at the same time it can accept user commands, too. In this research, the view or graphical user interface (GUI) is represented by a web page in web browser on client side. For development of the web page a Google Web Toolkit was used as a perfect tool for transformation of pure Java code into JavaScript that is implemented in HTML. The developed GUI for pallet monitoring, as one of the most important outputs of this thesis, is present in Appendix A.
- **Controller** – controller is responsible for information flow between the view and the model. It actually transforms incoming REST requests from view into model operations and sends back appropriate results. Among other server responsibilities belong communication with database and endpoint data forwarding.

In addition to main Spring MVC structures there are also a two Web Service **endpoints** established in the server. The endpoint serves as a main gateway for filtering all incoming Web Service SOAP messages. Only desired messages are parsed and forwarded in meaningful data to REST controller. First endpoint is dedicated for main messages coming from wireless devices while the latter one is responsible for messages coming from line (RFID data). Whole software architecture is presented in Figure 4.9.

In summary, the overall architecture of information flow from sensor to client looks as follows. Every event log that is generated by a wireless embedded device is encoded into SOAP message, sent via LoWPAN network to edge router and then using Ethernet link to appropriate server endpoint. The WS endpoint parses the SOAP message according WSDL specification and sent decoded event log to REST controller. The controller forwards the event log to a process model for further evaluation and subsequently it sends also a copy of the event log to database for storing. The event logs, stored in database, serve later on for further analysis or for offline monitoring. As soon as the event log is processed by the process model, and pallet location is updated, it sends result data (pallet location) back to the controller together with a list of clients that are going to be informed about this update in pallet location.

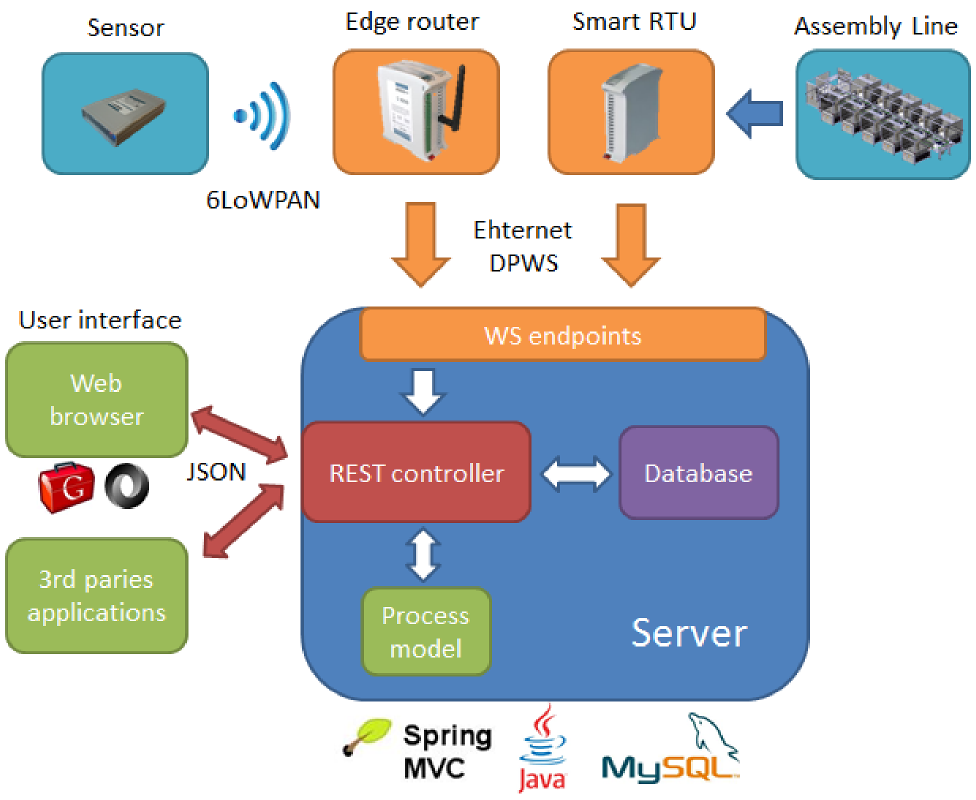


Figure 4.9 *Overall software architecture*

Finally there is also important to describe a communication between server and client or any other 3rd party interested in the process. Their mutual communication is always initiated by a client, which sends an AJAX request to the server with appropriate input data wrapped in get parameters of the request. Based on the way, how the server handles the requests there are distinguishable two methods called *short polling* and *long polling*, of which meaning is going to be explained in next paragraphs. Server response is always in a form of a JavaScript Object Notation with padding (JSONP), which is a data format that provides a possibility to be retrieved also from a server that is located in a different web domain than the requesting client is. This feature is known also as a bypass of the same origin policy. Server, as was already mentioned, can handle client's requests in two different ways and web page must be for both ways properly defined. In this research a combination of both ways were implemented, since each of them has its own advantages, which are described in following paragraphs.

- **Short polling** – it is a conventional way of gathering data by most current internet web pages. During short polling, the web page regularly (once per few minutes) sends a request to server to get a new data and the server immediately replies back with appropriate response. The response can either contain new data that were requested by the web page or it can be also empty, if no new data are currently available.

- **Long polling** – it is relatively new technology, which reduces amount of AJAX requests to necessary minimum. In fact, long polling suppresses empty responses to be sent. The mechanism is actually quite simple. The client (web page) sends an AJAX request to a server and server either replies with desired data if they are available or waits until there will be something to be send. During that time the connection is open. As soon as the server obtains a new data, a JSONP with data is created and sent as a response to the client. In case, there are more clients, the response is sent to all of them. List of all clients, or references to their connections are stored in the server. Naturally the client request does not last forever and there is always some timeout (usually it varies from 10s to 1 minute). It means that if no response comes (no new data are available on server side) till the timeout, the connection is closed and new request is send to the server again. This approach has two significant advantages. Firstly there is no delay on client side for receiving new data since long polling is fully event driven mechanism. Thus, there is no need for client to regularly “beg” for a new data anymore as in short polling, but the new data are immediately sent when they are simply generated. Secondly, all clients receive the same data at the same moment what is much appreciated featured in some kind of branches e.g. stock market information. Finally there is no server overloading anymore.

Short polling as a mature way of data retrieval was also in the beginning applied in this research with only one significant change – higher request frequency (20 requests per second). The frequency was obviously chosen very high to let the monitoring to be truly real-time. Although it worked very well in the beginning, very soon a significant drawback of short-polling was discovered. The problem was that amount of requests was directly proportional to amount of active web pages, which led on client-side to increased processor usage, if more web pages were open at the same time, and on server-side it leads to overloading by processing of high amount of requests. Therefore a long polling was deployed. Short poling remained to be used solely for one-shot requests for instance to initialize a web page, to check whether the server works or not, or for sending user control data to server and retrieving control confirmation back. On the other hand, long polling was applied for all regular updates of the web page i.e. pallet actual positions and related information, list of available maps as well as list of all initiated pallets and coefficients. By this change it was ensured that all web pages are always fresh and they do not overload a client processor.

4.7 Environment model

As was mentioned earlier, because of limited perception capabilities of agents used in this research, the model of conveyor system plays very important role in agent navigation and presents fundamental knowledge about the surrounding environment. Therefore, there was given a significant effort to match the reality as close as possible. Since the main objective of this research is pallet localization, it is not surprising that the model is related to a map. The map, presented in Figure 4.10, consists of 154 blocks, where each block represents certain section of the conveyor system of the production line. There are blocks denoting straight sections, various turnings, junctions and connections as well as places where an RFID readers or pallet stoppers are placed. Concretely there can be seen ten drawing workstations (each consists of main conveyor and bypass) and one workstation 1 that is located in very right (only one main conveyor and no bypass).

The map was created in special web interface developed especially for this purposes and which is described in Appendix A. In that interface there is a list of all available types of blocks from which an operator simply drags and drops the chosen block into a grid until like a LEGO building kit the whole map of production line is created. There is also possibility to rotate blocks to fit the scenario, if needed. The resolution of the grid is 100 mm in both directions and the size of each block in both directions is always whole multiple of 100 mm. A design of all these building blocks was done in such a manner, that if a block that occupies 2 grid cells and thus it is long 200 mm, it also represents 200mm long section of conveyor system in reality. This synchronization gives the model perfect property to determine easily the pallet position if a block, at which the pallet stands, is known. The size of 200 mm for vast majority of blocks is not a coincidence but it was chosen deliberately. The reason is that the length is exactly equal to length of pallet, means that each block represents also a position which pallet can adopted if it is stopped by s stopper or by queuing line of pallets stopped in front of the pallet.

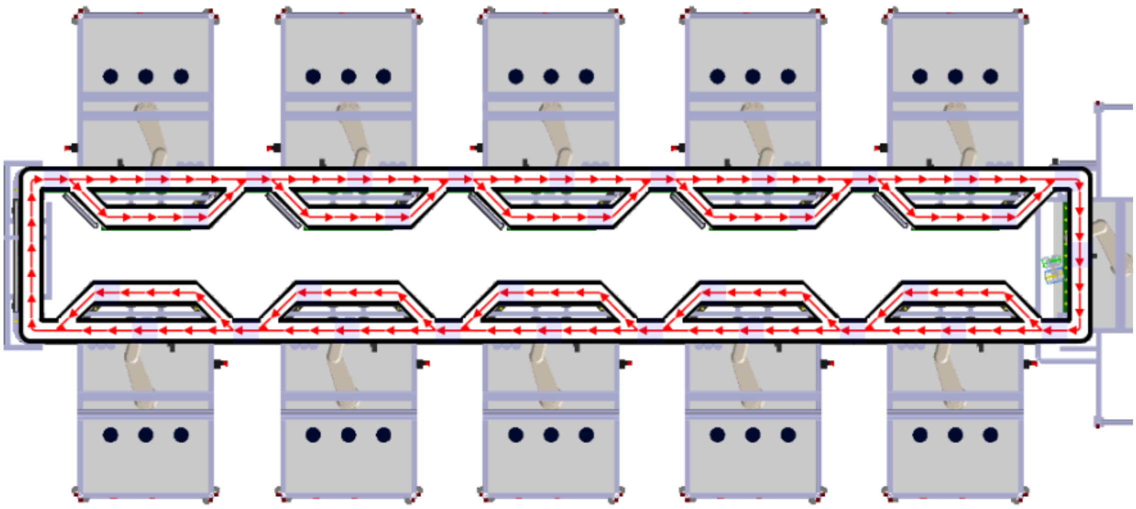


Figure 4.10 *Example of a map of FASTory production line*

Of course, the line design interfaces is versatile and allows user to create any other line in the same way as for the production line used in this research. Moreover, there are options to define each block to be part of certain workstation, to have stopper or RFID reader. Finally, the map is exported in form of text line to main server, where is stored, translated into appropriate machine language and immediately prepared for usage.

4.8 Algorithm implementation

To ensure smooth functionality of the two algorithms which are going to be developed and implemented, at first a list of assumptions needs to be created to summarize all significant characteristics of given testbed with respect to pallet localization.

Assumptions:

- **Conveyor system:**
  - Map or schema of conveyor system is known in advance
  - Conveyor system is flat – no lifters
  - Pallet movement is strongly restricted by a conveyor system – pallet never leaves the conveyor system except a case, when a pallet is taken manually away from the line
  - If the pallet leaves conveyor, the pallet is no anymore monitored and it must be initialized again.
  - Conveyor drags a pallet always only in one direction (forward)
  - Conveyor can stop
  - The conveyor system can accommodate many pallets at one time
  - Pallet can start from arbitrary place

- **Events:**
  - Motion events are generated from sensors attached to pallets
  - Besides motion events there is available information from RFID reader, too
  - Some event logs might be missing for some reason – not generated, lost etc.
  - Event log can come later contributing to a change in event order
  - No event log duplication for the same event occurs
  - Event RIGHT and LEFT are highly reliable and significant events – if one of these events is generated, it is very high chance the pallet really turned
  - Events GO and STOP are not reliable and should have only support meaning

4.8.1 Monte Carlo Localization

First attempt to process event logs and extract pallet location out of them was done using AI localization technique called Monte Carlo Localization (MCL), of which mathematical notation was presented in Chapter 2.1.6. For successful algorithm development and implementation a set of initial assumptions and clarification must be at first introduced.

- Because the pallet is pulled by the conveyor system under normal circumstances always forward – the control signal for the algorithm is given by time interval. For instance, every 700ms there is a control signal generated to move a pallet one state forward.
- The necessary time interval for the pallet to move between two states in process map was measured on real production line and lately fine-tuned
- Probability of movement is always given by a vector of probabilities that determines with what probability the pallet stays after movement again in the same state, with what probability it moves successfully to next state or even moves two states more.
- If a “STOP” event is detected, no forward move control signal is going to be generated anymore and program waits until event “GO” or any other event comes.
- Probabilities of various measurements (incoming events) are set according to a reliability of the events to be correctly generated. It means that for instance turning events copy with high reliability the actual turning of the pallet. They are rarely missed or be redundant. In contrast to events like STOP and GO, which are hard to detect or be misplaced, and their measurement does not have to necessarily mean that the pallet actually did the action.
- All probabilities were firstly manually chosen to fit into given model and prepared to be easily automatically tuned to increase overall localization efficiency.

The implementation of MCL algorithm for pallet localization is explained in simplified form in Figure 4.11. The probabilities used for this example are following.

Probability that a pallet is on right turn block and a right turn was detected is 0.8

$$P_{RR}(R_S\,|R_m) = 0.8$$

Probability that the pallet is not on right turn block and a right turn was detected is 0.2.

$$P_{NR}(NR_S\,|R_m) = 0.2$$

Probabilities related to right turn are equal to left turn as well.

a) In first step a map is given with characteristic path profile. The map consists of 8 blocks - five RIGHT turns, one LEFT turn and three STRAIGHT blocks.
b) Initially the same particle distribution is given to all blocks – in this case there are two particles. Thereupon a RIGHT event is measured.

c) Now for each block i a simple calculation is done. All particles from each box are multiplied by an appropriate probability whether the particle is on is not at certain place while some measurement is done and their probabilities are summed into non-normalized importance weight ω_{nn_i}. For example for a first left upper block $i = 1$, there are two particles multiplied by a probability of 0.8, because there are placed in right turn block and right turn event was detected. Thus the value of non-normalized importance weigh is

$$\omega_{nn_1} = 2 * 0.8 = 1.6.$$

The same operation is done for each block and their results are summed into normalizing factor η.

$$\eta = \sum_{i=1}^{n} \omega_{nn_i} = 5 * 1{,}6 + 3 * 0.4 = 9.2$$

Next step is a normalization of all non-normalized weights. Therefore the non-normalized importance weight of each block i is divided by the calculated normalizing factor η to get normalized importance weight w_{n_i}.

$$\omega_{n_i} = \frac{\omega_{nn_i}}{\eta}$$

So for all right turn blocks the normalized importance weight is 0.1739 and for other blocks only 0.043. Algorithm is now almost complete. In last step it is necessary just to recalculate number of particles for each block by following formula, where N_p is total number of particles ($N_p = 8 * 2 = 16$). In this way we get for all right turn blocks 2,7 particles and for rest of block just 0.69 particles.

$$N_i = \omega_{n_i} * N_p$$

d) In next step a forward movement is applied (blue arrows) and left turn is detected. For simplicity of this example a probability of movement is set to be equal to 1. This means that all particles will move with 100% probability to the very next block. The algorithm for recalculation of particle amount goes in the same way as in the step c.

e) Since there is only one left turn block in whole map, it is clearly seen, that algorithm in just two steps easily and unambiguously distinguished pallet location with quite reasonable probability – a block where an arrow is point at.

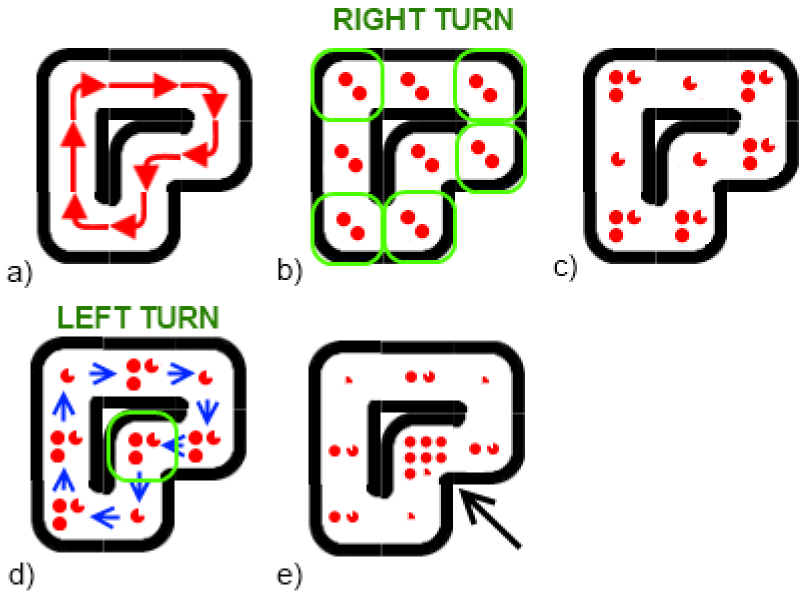


Figure 4.11 MCL algorithm for pallet localization

In addition, in real experiment the map is more complex and includes blocks such as junctions, where pallet moves either straight or right; or sinks, where two conveyors are joining together. The algorithm was thus supplemented by appropriate probabilities solving all possible scenarios and by ways how to deal with particle splitting and merging on these blocks. Also uncertainty of movement was added.

4.8.2 Simple decision-making algorithm

As was already discussed in the Chapter 3, the second algorithm that is going to be developed for pallet localization is going to utilize decision theory, concretely fundamentals of simple decision-making under certainty. The chapter also defines an agent as virtual representation of a pallet and the developed algorithm as its program, its brain that is responsible for "correct" decisions. In this chapter, the idea will be more extended and actual implementation will be presented.

Each agent is characterized by its own life cycle consisting of three different phases – **initialization**, **synchronous tracking** and **asynchronous tracking**. Since each of them represents a different life period of the agent, it was necessary to develop for each of them also a special decision-making algorithm. In following paragraphs both phase and appropriate algorithm is going to be described.

1) Agent initialization phase

Agent initialization, as name prompts, is a first phase of agent life cycle, when a new agent is introduced into the system. In reality it corresponds to a time, when a system receives an event log from new unknown pallet which was inserted to the conveyor system. This phase lasts as long as the agent is successfully localized within its environmental model. After receiving the first event log the agent position is mostly not yet known, since the single event log contains only simple information about one pallet movement. Nevertheless, the system at least registers the new agent into own maps and prepares enough resource capacities for tracking and storing agent’s incoming activities. As the pallet moves, new event logs are received and stored. After each of them the initialization algorithm is applied to search and find a predefined initial event pattern. The initial pattern is artificial set of events that must be received from a pallet to successfully accomplish initial phase of the agent.

The initial pattern can be freely chosen by a user and consists of combination of any three known events such as for instance RFID, LEFT, RIGHT or RIGHT, LEFT, RIGHT etc. The unique pattern consisting of three events received right after each other should be sufficient for vast majority of conveyor systems. If conditions allow, the user can choose also an event UNDEFINED to skip a required event and reduce amount of events in initial pattern. By default (used in this research), the first event must be a RFID event. RFID event provides information about exact pallet position within the conveyor line and thus precisely localizes agent. Therefore, there is no need for any other events more and thus by default the two rest initialization events are chosen to be UNDEFINED. In case of FASTory line, used in this research as a testbed, the RFID event must be always the first initial event because of symmetry of conveyor system.

As soon as the initialization algorithm receives the right combination of events, it launches a scan matching, a technique that tries to match the given initial pattern with a-priori model of the conveyor system. If such a pattern is found, the initialization phase is over, agent is successfully localized and it switches to synchronous tracking phase.

Process of initialization for each agent is visually displayed to the user in screen in form of percentage of completeness inside a waiting ring as is shown in Figure 4.12.

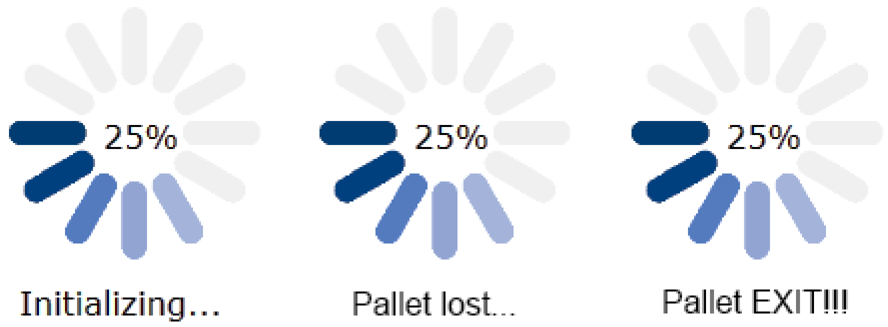


Figure 4.12 *Visualization of all three agent's abnormal states: Initialization phase, lost synchronization and pallet exited line (all in 25% of completeness)*

2) Synchronous tracking

Synchronous tracking is a phase, when an agent is continuously monitored and its estimated position corresponds to real position of the pallet – so they are in mutual synchronization. The only mechanism, how to check it, is based on RFID events and it is going to be described more in next phase. So far, in this phase it is supposed they are in the synchronization.

The synchronous tracking decision process goes in simply words as follows. As soon as the event log is registered, it is assigned to appropriate agent (the agent that represents the pallet from which the event log came) and algorithm determines utilities for all possible agent alternatives (states). The alternative with the best utility value in the end is chosen as the final one and the agent changes its state accordingly. Thus, the biggest challenge of this algorithm was to develop a mechanism for utility evaluation. However, before the mechanism can be described more in details, it is necessary for better understanding to mention a significant role of input events.

Every block in map is characterized besides other parameters by set of input events. The input event of a block is an event that must be received in order to move to the certain block. Thus, for instance a right turn block has as input event an event RIGHT. However, it is not constraint and therefore as soon as the model map is loaded to system, special engine is launched to discover for each block of the map its neighbours and its ancestors in order to determine what kind of events are input event for that particular block. Besides already known events, generated by wireless sensors or RFID sensors, there is also one artificial event TIME, generated by the system model itself. The event TIME represents pallet free movement in the conveyor system. It is generated in a case, when a pallet is located in a block of which neighbour does not expect any special event and pallet just passes by. Such a block has as an input event just TIME and it might be for instance a straight section of conveyor, when the pallet just moves from one block to other without generating any of inertial event.

When a certain event is registered (called further as examined event) to the algorithm, the algorithm at first looks certain amount of blocks forward (by default 80) and certain amount of blocks backwards (by default 20) to look up in the map for all blocks that have input event equal to the event that is examined. In backward search all blocks for which the certain event was already once applied and registered for the particular agent, are of course expelled. Each of those found blocks represents a decision alternative, to which a proper utility must be assigned. That is a work for utility estimation mechanism.

However, before the working principle will be described in next paragraph, it is necessary to explain how the mechanism understands a term of *utility*. Usually the higher utility a decision alternative has the better outcome for the agent it represents. On contrary, the developed estimation mechanism calculates utility in opposite way – the lower the better. Therefore, the term utility will be called further in the explanation by a term *penalty* that describes better the actual meaning.

A penalty V_x for each block *x* from set of alternatives is calculated according to Equation 4.1, where d_x is a distance in number of blocks between a block, at which an agent is currently located and the examined block *x;* coefficient w is an importance weight of each event according to values listed in the Table 4.3. The lower number a certain event has, the more significant and more reliable event it is and vice versa. Furthermore, the equation contains a set of adjustable coefficients too. The orientation coefficient c_0 adjusts a direction profitability, because it assigns a different value for forward movement (by default is equal to 1.2) and for backward movement (by default set to be 1.9). In this way, the alternatives in a direction of pallet movement (forward) are made more favourable. The history coefficient c_h is coefficient which takes into consideration a fact, whether an agent already visited the examined block ($c_h = 1.2$) or not ($c_h = 1$). Visited means that the agent already stood at that block, but it entered the block by different event than by the event that is currently examined.

As soon as the mechanism assigns penalties to all alternatives, the algorithm chooses the block with the lowest penalty and the block become a new state of the agent.

$$V_x = {d_x}^w * c_o * c_h \qquad \text{(Eq. 4.1)}$$

Table 4.3 *List of importance weights*

Event	Importance weight
LEFT	1
RIGHT	1
EXIT	1
GO	10
STOP	10

The backward search is a way, how the algorithm can cope with late events. In order to understand a motivation and function of this powerful mechanism, two following examples are going to be shown and described.

Scenario 1 - There are given three blocks – straight block followed by left turning and immediately by right turning. The agent stands on the straight block, when suddenly a RIGHT event is registered. According to the abovementioned formula the agent jumps across left turning directly to right turning, as is shown in Figure 4.13 – 1a. However, right after that a LEFT event comes. The algorithm searches at first for block in forward direction and finds a nearest block that is 4 block distant with penalty $V1 = 4 * 1{,}2 * 1 = 4.8$. Then it looks backwards and finds out, that just one block backwards was a block with input event equal to LEFT and at the same time, this LEFT event was not for that block and for that agent registered yet. Calculated penalty for this block is $V2 = 1 * 1.9 * 1.2 = 2.28$ which is lower than for the block in forward direction of the pallet movement and thus with high probability the examined event is just a late event. Nevertheless in result, the agent will not move back to that left turning (it is too late, pallet is actually on the right turning already), but the event will be supplemented to agent event history and the agent stays at a position where it is (Figure 4.13 1b). If a new event LEFT theoretically comes again, the algorithm would not take the block into the consideration any more, since the LEFT event has been for the block already applied.

Scenario 2 - The property of backward searching might be very useful also in scenario when agent enters junction, where it can move either straight (artificial event TIME generated) or right (RIGHT event). If the RIGHT event will be a bit delayed, the artificial time interval will elapse and TIME event will be generated causing the agent to move straight (Figure 4.13 - 2a). However, if the RIGHT event comes, the algorithm at first looks forward in order to find a block that is in this case 7 blocks distant ($V1 = 7 * 1.2 * 1 = 8.4$) and then looks backwards, where two possible blocks with input event of RIGHT are found. While the block 3 is not examined at all, since the RIGHT event was already applied a while ago to this block, the second block has penalty equals just to $V2 = 3 * 1.9 * 1 = 5.7$, which is less than for block 1 and therefore the agent will jump to the second place and agent position history is upgraded according to the new conditions as can be seen in Figure 4.13 - 2b.

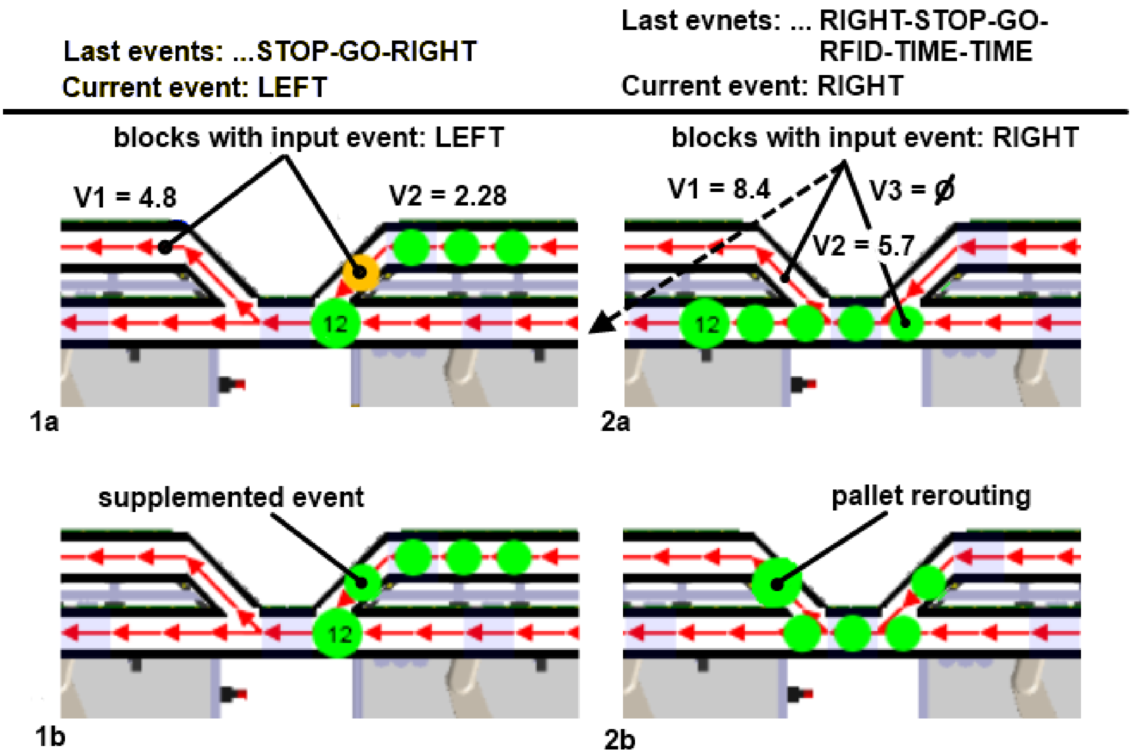


Figure 4.13 *GUI representation of algorithm behaviour with calculated penalties for two different scenarios: 1 - delayed LEFT event a) before the event is received, b) after – as a result, the event is just supplemented; 2 – delayed RIGHT event on RIGHT Y-JUNCTION a) earlier than the event is received a TIME event is generated and agent continues straight, b) after – as a result, the agent path is rerouted*

3) Asynchronous tracking phase (Calibration)

Asynchronous tracking is a phase of agent life cycle, when its position does not match with position of real pallet and thus they are not in mutual synchronization. In fact, there is only one way, how to recognize pallet dislocation – using RFID event. Hence, whenever a RFID event comes and the workstation from which the event came from does not correspond to a workstation, in which the agent is located, the agent is said to be lost – visualization is shown in Figure 4.12. The algorithm that would localize the agent again is almost identical to the one presented in the initial phase. The algorithm again waits until initial sequence of events is received and if the initial pattern is found, the agent state or location is again recovered and agent is again synchronously tracked. Besides RFID event, also EXIT event can get the agent into the asynchronous tracking phase and again the initialization sequence of events is needed in order to track the pallet again.

Based on previous paragraphs, where fundamentals of the algorithm were explained, one can notice that there are many coefficients in use. Even though they are key components for smooth algorithm operation, they were chosen based only on empirical experience and they do not have to be optimal for given process. Therefore, a revalidation mechanism was implemented in order to verify given coefficients and propose new coefficients that would lead to better efficiency i.e. better accuracy of agent positioning. Firstly however, a benchmark, according which the tracking efficiency is measured, must be introduced.

As the benchmark an overall localization efficiency of all agents will serve and which is calculated as an average value of all particular agent movement efficiencies. The movement efficiency of a single agent is calculated than as follows. The path that certain agent drove over is stored in model maps and consists of set of blocks. Each block has a certain amount of input events that the agent uses for entering the block. When the agent moves, the input events are registered and stored. During proper synchronous tracking, for each block, the agent visits, all necessary input events should be received. If it is not so and certain input event is missing, the information is going to be stored for agent movement efficiency calculation. The agent movement efficiency mechanism simply calculates for all blocks, the agent visited, a ratio between sum of all input events, which were in reality received, and sum of all events that were supposed to be received. This ration is agent movement efficiency.

By averaging one get the overall localization efficiency that serves as etalon for further calculation. In order to increase localization efficiency, the engine changes artificially coefficients and recalculate whole agent track again. The new coefficients give new overall efficiency which is compared to the etalon. In case the new overall efficiency is better, the actual coefficients are changed accordingly. This revalidation can be done either automatically (after each agent round in production line) or manually by operator command. For each revalidation there are examined all combinations of both coefficients (c_o, c_h) that can gain these values:

$$c_o' = (-0.2, -0.1, 0, +0.1, +0.2) + c_o$$
$$c_h' = (-0.2, -0.1, 0, +0.1, +0.2) + c_h$$

This in total gives 25 possible combinations from which the best one is chosen. In this way, the algorithm can dynamically update its own parameters to adapt for changing environment or simply to optimize localization process.

5 DISCUSSION OF RESULTS

The outcome of this thesis was to develop a working prototype of real-time monitoring system for localization of manufacturing carriers using data from inertial MEMS sensors, while utilizing techniques of artificial intelligence. The aim of this chapter is to present obtained results of all implemented methods supplemented by conclusions and discussions related to their implementation, suitability and reliability.

5.1 Signal processing and event detection

As was mentioned already in research methodologies, the first step of implementation of signal pattern matching was to study thoroughly the raw sensor data for determination of patterns corresponding to significant pallet movements. Before further explanation how the pattern matching was done, it is necessary to introduce a reader the coordinate system at first according which sensor axes are labelled and which is shown in Figure 5.1.

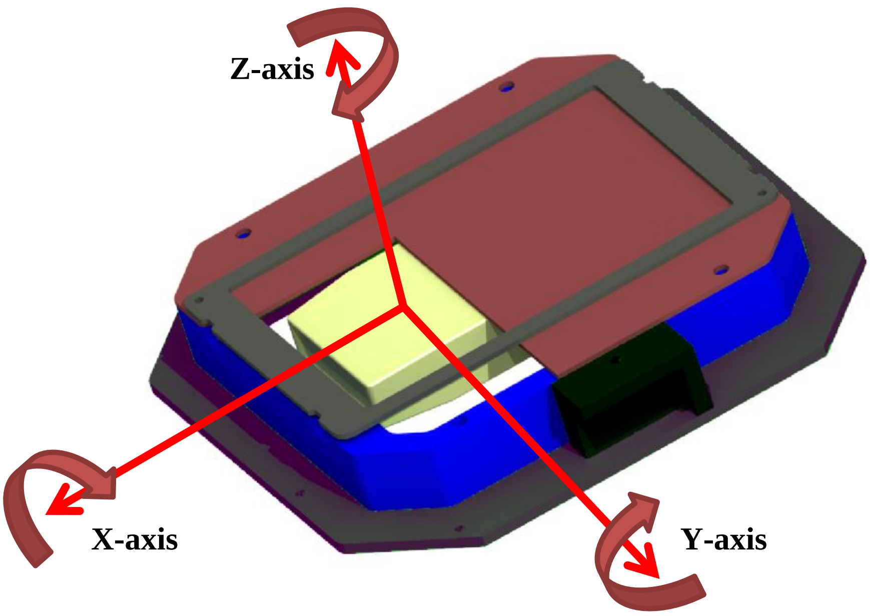


***Figure 5.1** Accelerometer and gyroscope axes*

In total, there were distinguished five significant pallet movements that are both feasible to be detected by sensors and useful for purposes of localization, too. Those five movements are **START, STOP, LEFT, RIGHT** and **EXIT** and their meaning is presented in Table 5.1. Along with the meaning, the table also contains information how the data pattern looks like and which data source is scanned in order to discover it. The process how those two last properties were discovered is based on observations which were done by studying each of sensor axes and which are listed in following paragraph.

Gyroscope:

- All gyroscope axes provide relatively reliable and noiseless data
- Gyroscope X-axis (roll) and gyroscope Y-axis (pitch) are for this research meaningless since the pallet moves and turns only in horizontal plane of the conveyor system
- Gyroscope Z-axis (yaw) is suitable for detection of pallet turnings in XY plane such as LEFT and RIGHT, where the signal creates relatively well detectable sinusoidal curve as can be seen in schema in Figure 5.2a.
- Noise level however does not allow to determine exact angle of turning unless special data mining algorithm would be implemented, which is out of scope of technical possibilities of given wireless devices

Accelerometer:

- All accelerometer axes suffer by significant noise of which source is unambiguously a way how the pallet is transported by conveyor system
  - Noise in Z-axis is caused by relatively rugged conveyor system on which pallet jumps
  - Noise in X-axis is caused primarily by the type of conveyor system, in this research belts are used for pallet transportation. Naturally, pallets are not tightly connected to belts and thus slip appears. Also when a pallet is stopped by stopper, the belt is still running underneath the pallet what in both cases causes again relatively significant noise.
- Even though START and STOP events would be very convenient to determine by X-axis, since both events actually are a result of acceleration in X-axis, but due to mentioned noise in X-axis, it was infeasible to obtain reliable results in the end at all
- Y-axis is from perspective of utilizabibility in this research absolutely useless – the pallet is guided from both sides by rails and therefore there is no meaningful acceleration in this direction.
- Z-axis does not suffer by so many disturbances as X-axis. It is a convenient data source for event EXIT, since taking a pallet out of the conveyor line means a continuous acceleration and deceleration in Z direction. Moreover, it was discovered, that Z-axis can serve for sensing START and STOP events as well with surprisingly better results than for X-axis. When a pallet stops, the noise level in Z-direction significantly decreases (in contrary to X-axis when due to vibrations on stopper the noise level stays more or less the same), which can be easily detected. When a pallet is released by stopper to move again, the noise in Z-axis rapidly increases. The schema how the patterns look like for all the three events is illustrated in Figure 5.2b.

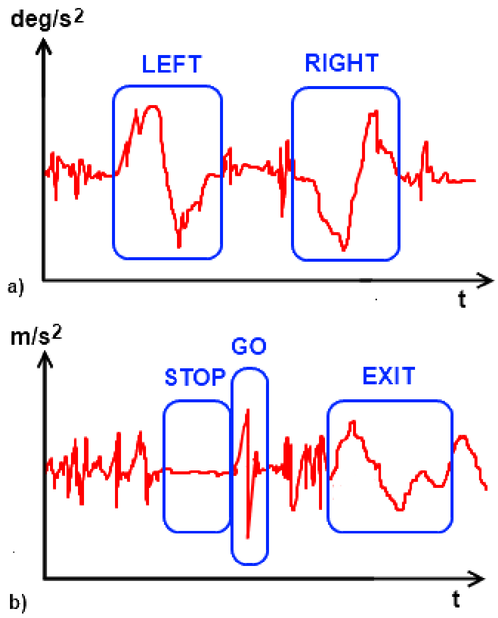


Figure 5.2 *Pattern matching - a) Z-axis of gyroscope b) Z-axis of accelerometer*

Table 5.1 *Event description*

Event	Description	Data source	Signal pattern	Reliability
LEFT	Pallet turns left	Gyroscope z-axis	Sinusoidal wave	High
RIGHT	Pallet turns right	Gyroscope z-axis	Sinusoidal wave (opposite phase)	High
STOP	Pallet stops	Accelerometer z-axis	Significant noise reduction	Medium
GO	Pallet start moving again	Accelerometer z-axis	Noise peak increased	Low
EXIT	Pallet leaves conveyor	Accelerometer z-axis	Sinusoidal wave	High

The algorithm was uploaded to chips of wireless sensors that scan continuously the sensor data output for presence of any of these patterns. As soon as the event is detected, the event log is created and sent to all subscribers.

In summary, implemented pattern matching algorithm was proved to be a sufficient way for event detection with respect to given limited resources and for purposes of pallet localization. However, its reliability strongly depends on the environment in which it is used and must be tuned for each particular application. The example of real data output from Z-axis accelerometer with described pallet behaviour and marked points, where an event was registered, is shown in Figure 5.3.

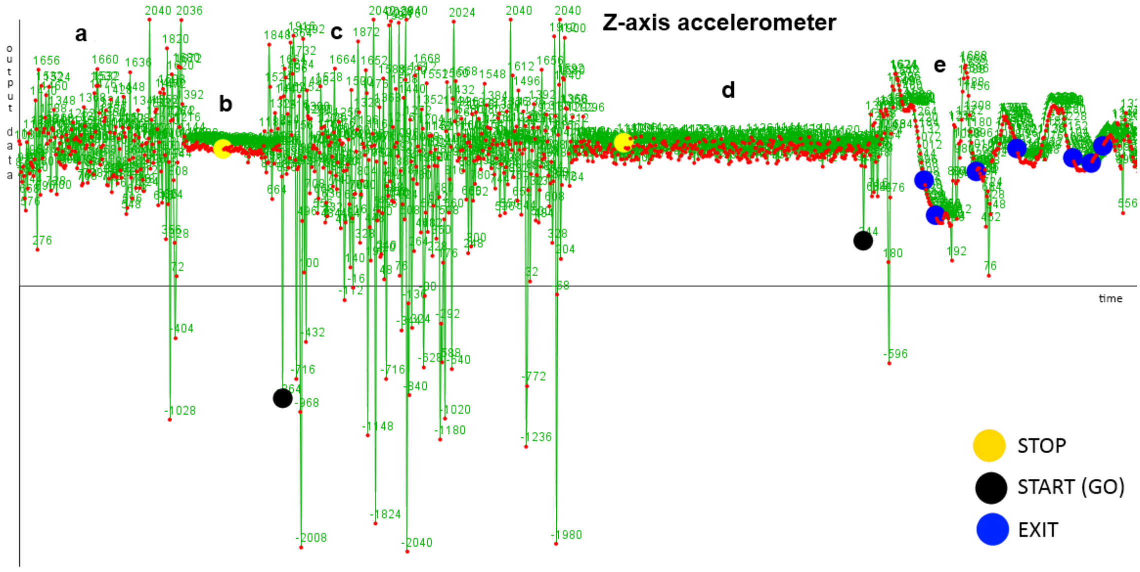


Figure 5.3 *Example of event generation from data signal of Z-axis accelerometer: 5 significant areas - a) pallet moves, b) pallets stopped by stopper, c) pallet moves again, d) pallet stops, e) pallet is taken by operator away from conveyor system*

5.2 Edge router position

The transmitted signal strength from used wireless devices is usually very low due to their simple low-powered antennas. The situation gets even worse because both the chip with antennas is packed into plastic cover and the wireless device itself is encapsulated into the pallet structure. The design of the pallet is done in such a way that the pallet body, in which the device is hidden, influences negatively the properties of signal transmission. Based on observations, it was discovered, that the top pallet metal plate shields the device signal very significantly, whereas the pallet plastic body is more signal permeable, but on the other hand its dome structure causes with high probability signal reflections. Therefore, there left only one clear propagation way for the signal and that is down towards ground through holes in pallet bottom. The signal propagation schema is shown in Figure 5.4.

To increase signal receiving efficiency as much as possible, there was done set of experiments to determine the best mutual position between wireless devices and edge router. A benchmark of efficiency was represented by amount of received messages by the edge router per minute for each wireless device. For better credibility of the experiment there were used two wireless devices simultaneously and each of them transmitted one messages per second. Thus, in ideal case the edge router should receive 60 messages per minute for each wireless device.

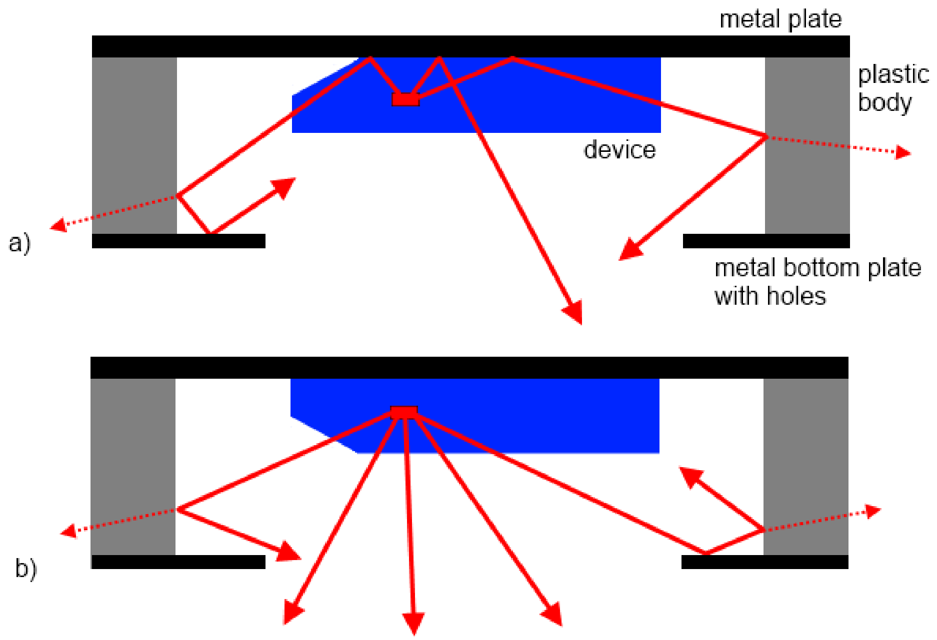


Figure 5.4 *Signal propagation schema: a) normal position, b) upside down position*

In total, there were tested seven different mutual positions between edge router and devices and each experiment lasted for exactly twenty minutes. The overall processed results are shown in Figure 5.5 where the vertical axis stands for amount of received messages while the horizontal axis displays letters that represents each experiment of which description can be found in Table 5.2. Each experiment has two values for the two pallets that were measured.

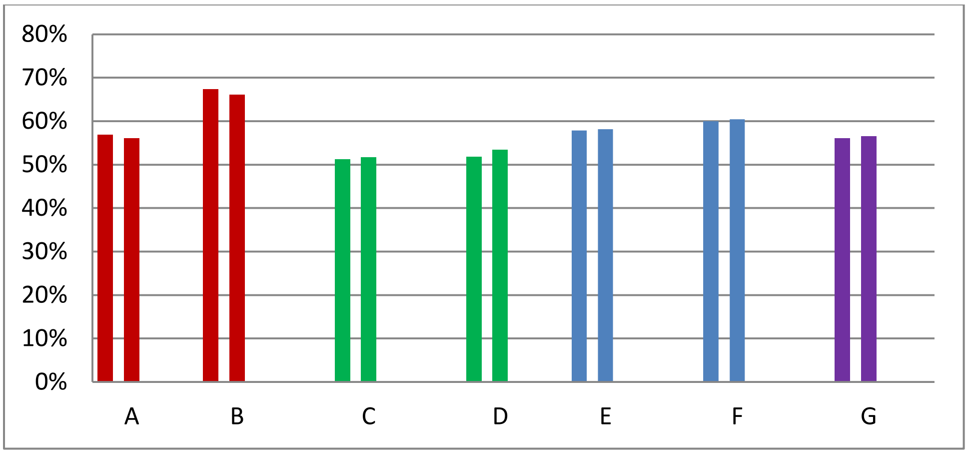
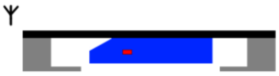
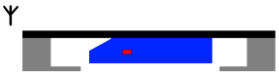
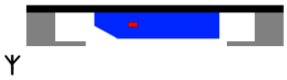
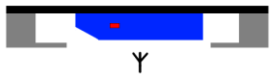


Figure 5.5 *Efficiency of received messages in dependency of mutual position between edge router and devices*

Table 5.2 *Experiment description*

#	Position of wireless device	Position of edge router	Situation	Schematic icon
A	Normal position, open pallet	On side, above conveyor system	Dynamic	
B	Normal position, open pallet	On side, above conveyor system	Static	
C	Normal position, closed pallet	On side, above conveyor system	Dynamic	
D	Normal position, closed pallet	On side, above conveyor system	Static	
E	Normal position closed pallet	In the middle, down the conveyor system	Dynamic	
F	Upside down, closed pallet	In the middle, down the conveyor system	Dynamic	
G	Upside down, closed pallet	On side, down the conveyor system	Dynamic	

Based on results, shown in the Figure 5.5, it was proved that there is a significant difference between open (red columns) and close pallet (non-red columns) which is caused by device encapsulation and subsequently by signal reflexions and absorptions. However, the choice between open and close pallet is in fact already given due to testbed properties, since open pallet cannot hold wireless device at the same time with a paper for drawing operations and these measurements serves exclusively for comparisons.

The results also proved the theory that upper plate is impermeable for the signal and thus the signal propagates mostly downwards – efficiency increased in experiments, where edge router was placed below the conveyor line (blue columns). Only minor change is then in position of edge router down the conveyor line. Position in the middle gives slightly better results than the position on side of workstation, but on the other hand middle position is very impractical place and thus the edge router was finally placed down the conveyor line and on side of one workstation.

From the results showed in Figure 5.5 is clear that the amount of messages for both devices were during all experiments almost equal, which proved that the measurement was correct, but by closer look to individual experiments (Figure 5.6 and Figure 5.7) one can notice very interesting thing. The amount of received messages for both devices varies for all the experiments over the time absolutely in the same way, regardless their positions. Next important discovery came out from static tests (Experiments B and D in Figure 5.6 and 5.7, where the pallets are not running in conveyor system, but standing in static positions – one in close distance above edge router and the second one in the most distant place of the conveyor system). It was found out that amount of received messages is for proximal device almost the same as for the distant one and therefore if the edge router is placed in the middle of the line, the position of pallets does not play any significant role.

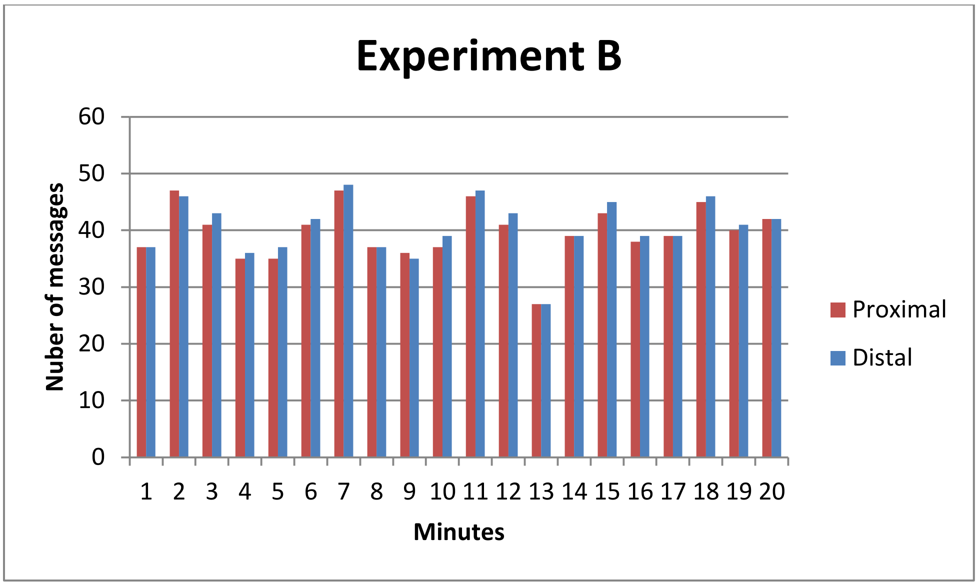


Figure 5.6 *Results of Experiment B*

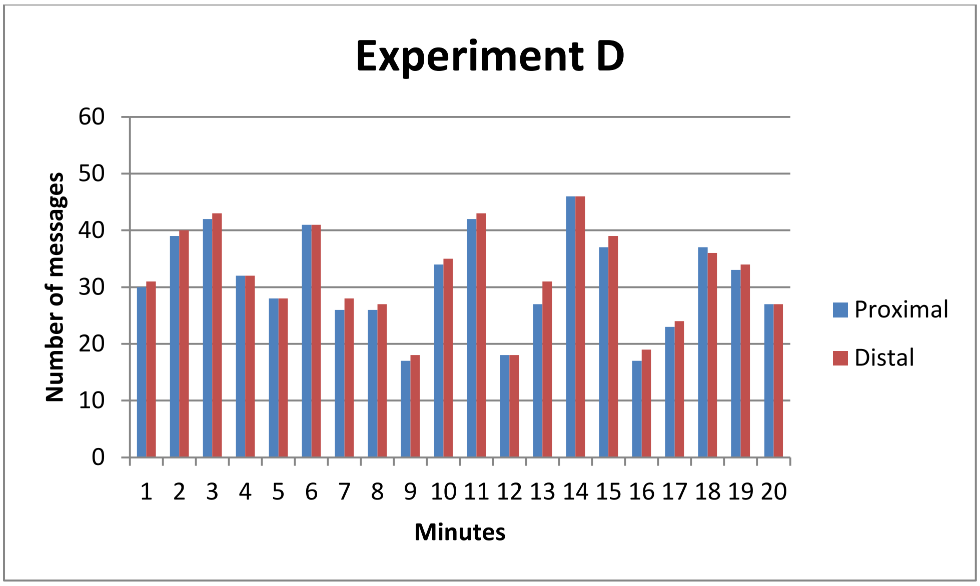


Figure 5.7 *Results of Experiment D*

5.3 Results of used artificial intelligent methods

5.3.1 Monte Carlo Localization

Monte Carlo Localization has excellent results for localization of mobile robots and thus it was expected to be a promising solution for pallet localization as well. However, this algorithm did not live up to given expectations. In fact, the major obstacle was in a way of which measurement data were gathered. In case of mobile robots, the robot at certain time moves with some probability to certain place and makes a measurement of its environment around. Then it gathers data, evaluates them and continues in movement. So the measurement data are tightly related to the particular movement, space and time interval.

In contrast to the case of mobile robots, the pallet movement is continuous, uncontrolled (just pulled by conveyor belt) and at the same time independent on measurement. Therefore, a measurement results do not correspond to any particular movement, have artificial timestamp, which is created not at time the event log is created but when it is received; and due to their transmission technology they might be even significantly delayed or lost. In reality, it was very difficult to synchronize pallet movement with agent movement. If pallet stopped and no event was accidentally received, the agent still continues moving. Because of ability of self-discovery, it was able to localize itself very soon again, but in the end most of the time the agent spent just by self-localization and thus most of the time the resulting probabilities of agent location were too small to provide sufficient determination of a place, where the pallet actually was. Generally speaking the environment in which was agent, controlled by MCL algorithm, implemented, was not suitable and any attempts to apply MCL for pallet localization on given testbed was immensely challenging or even unfeasible.

Final conclusions and observations of MCL

Method strengths are:

- Versatility – method can be easily applied to any kind of map
- Simplicity and scalability – the algorithm is very simple, the amount of particles determines directly accuracy and computing difficulty and can be scaled flexibly according to computation power of processing device.
- Fast reaction – algorithm gives in most cases very fast results while the agent is localized only in few steps.
- Self-recovery - problem of kidnapped agent is under certain conditions solved very quickly (if at the place are available particles)
- Adjustability - all model probabilities are easily tuned according localization results.

Method weaknesses are:

- Incapability to react on delayed events – algorithm is built to process real-time measurement data not delayed ones. If all events had the same time delay, the algorithm would work fine with just that time delay. But in case of floating time delays, the algorithm loses significantly an ability to track pallet at all. For instance if RIFD event, which is regarded as significant and governing event comes too late, the agent is pushed back close to RFID reader while in reality is the pallet somewhere else.
- Event-to-movement connection - algorithm simply needs to have events related to certain movement otherwise if for instance pallet stops and event is not generated, the algorithm will simply push agent further and lose its location.
- Ambiguity - in case of symmetric maps, there is always as many possible final results as number of axis of symmetry. Thus a measurement must be supplemented by another information source, such as RFID tag.

The Monte Carlo Localization algorithm might work well for mobile robots and similar application, but in case that a movement is independent of measurement and data are obtained by event logs with fluctuating time delay, this method simply fails. Therefore, the development based on MCL for pallet localization was in this research stopped and second approach using decision theory was used.

5.3.2 Simple Decision-making algorithm

Simple decision-making algorithm that was applied in this research showed up as a very promising solution. Already in the beginning of its development the algorithm demonstrated its unique robustness that was later on even more improved. The mentioned robustness is based primarily on capability of discovering and dealing with late or missing events. The utility estimation mechanism that works with several coefficients describing willingness of an agent to listen to certain events, to move forward or backwards, appeared to be very effective tool for alternative evaluation. With its help the agent chooses its next state always with respect to maximization of final utility (or to minimize final penalty).

In addition, the robustness is supported by capability of self-calibration that is applied, whenever an agent loses the track of pallet to find its new correct position again.

Generally, besides the robustness the algorithm is very flexible, applicable for any kind of map of conveyor system and extensible for other features. It automatically determines the conveyor layout, recognizes specific conveyor sections and discovers dependencies of the sections. In certain boundaries it can also react to changing environment conditions by revalidating all model coefficients and their automatic change in order to increase the localization capabilities.

The observed strengths and weaknesses of the algorithm are summarized into the next two paragraphs.

.

Method strengths:

- Robustness – dealing with late, extra or missing events
- Self-calibration if necessary
- Coefficient revalidation for environment change
- Relatively simple and extensible algorithm
- All coefficients can be changed arbitrary according to given environment

Method weaknesses:

- Dependency on coefficients from the global point of view
- Dependency of RFID events as a source of initial location in case of symmetrical maps

5.4 Graphical User Interface (GUI)

One of the goals of this thesis was also to develop graphical user interface that would allow user to monitor comfortably pallet positions, to obtain necessary important data from the conveyor line and to control and interact with the process of localization itself. Therefore, there were developed three interfaces in total – *Design view, Real-time monitoring* and *Offline monitoring* of which properties and functionalities will be briefly described in next paragraphs. Nevertheless in case of interest, more detailed information how the GUI looks like, works and should be used can be found in **Appendix A** of this thesis.

- **Design view** – the interface allows user to build own brand new conveyor map by dragging and dropping predesigned blocks. The design process is very easy, intuitive and fast. The interface allows user also to set additional properties to various sections of the designed conveyor line such as position of RFID sensors, position of stoppers or belonging to certain workstation.

- **Real-time monitoring –** as name prompts, this interface provides real-time information about positions of all pallets in given conveyor system. It consists of two sections. The prime and largest section is monitoring area, where are both map of conveyor line and pallet positions graphically displayed while the second section is dedicated to localization process control. The control panel contains all necessary parameters and coefficients used for localization algorithm and user can freely adjust them to test different scenarios. The graphical monitoring area provides user several display modes to highlight desired information.

- **Offline monitoring –** offline monitoring has similar appearance as the real-time monitoring, but in comparison it is capable to monitor pallet locations based on historical event logs stored in database. The user can freely choose a time period in which he or she is interested and then monitor pallet locations in replay in the given time period. Since it is offline monitoring, the time can be speeded up or slowed down arbitrary.

6 CONCLUSIONS

Although the topic of object localization in arbitrary form is investigated by research groups worldwide still very intensively, there is generally a lack of industrial implementations. Therefore, a goal of this thesis was to utilize current technological and methodological possibilities of real-time localization for tracking of manufacturing carriers (pallets) within real production line as a part of factory asset-aware management system.

In summary, each pallet in the production line was equipped by wireless embedded device with accelerometer and gyroscope providing fundamental data about pallet movement. Usually the sensor data are sent straight ahead for further processing. However in this research, according to the latest trends in automation, the raw sensor data were at first preprocessed right in the wireless device in order to identify only the meaningful pallet movements. If the meaningful movement occurs (e.g. pallet turns), then only a single event log, carrying information of the movement, is published via 6LoWPAN network to subscribers. Such event log can be used by any client without having knowledge how the movement was determined. The development of the algorithm that provides signal preprocessing was also a part of the thesis and it was based on technique called pattern matching, since every movement has its unique signal pattern.

The generated events from wireless devices supplemented by events from RFID readers represent the central data source for pallet localization algorithm. The algorithm is running on the server and has web interface through which a user can monitor pallet positions, adjust important algorithm settings or event to build from scratch a new conveyor map.

In this research two different approaches were utilized to develop the pallet localization algorithm and both were implementing fundamentals of artificial intelligence. The first approach was based on a technique called Monte Carlo Localization that is widely used in mobile robot localization. However as it turned out, the technique failed in case of pallet localization due to unsuitable environment conditions, where the pallet movement is uncontrolled and independent on measurement.

In contrary, the second developed algorithm used for localization was a technique called Decision theory, concretely simple decisions under certainty, in which an agent as an intelligent virtual representation of pallet decides after receiving each event log, what is the next most probable pallet position based on complex set of coefficients. This technique proved to be relatively simple but on the other hand very robust. The algorithm robustness was achieved by two ways. Firstly, the algorithm has self-calibration means that if the pallet position is lost, the algorithm recognizes it and attempts to find it again and secondly a capability to cope with late, missing or redundant event logs. During the tests the algorithm was able to track pallets for vast majority of the time very well without any unexpected errors.

6.1 Future work

The research work presents one of the first attempts to localize and monitor manufacture carriers within the production line using wireless embedded devices equipped by inertial sensors. Therefore, the main objective of this research was to focus primarily for a feasibility of this project with given technology and right this feasibility was by this research finally also successfully proved as was described in the conclusions. This thesis work somehow opened doors to new promising area of wireless sensor application and give a potential for next growth and experiments. Therefore, there is going to be proposed in next paragraph a possible direction in which a future work might go.

One of the possible ways how to continue in this research would be better utilization of resulting data from this research. Since the developed localization algorithm provides complete information about all pallet movements and locations throughout the process, it would be very convenient to use this information not only for passive monitoring as is done in this research but as valuable data source for active process control, too. Nowadays process mining seems to be one of the most convenient tools for active management and control. The process mining has potential to build brand new process model just based on those data, it can provides conformance checking, in which the real process is compared with model outputs or process mining can also actively modify and redesign process model based on data from this work to increase overall process efficiency. Therefore the utilization of process mining is proposed to be next and most convenient step in this research.

.

7 REFERENCES

[1] Dhamee, Yousuf. Adam Smith and the Division of Labor. [Online] 1996. http://www.victorianweb.org/economics/division.html

[2] Assembly Line. [Online] Wikipedia, 2012. [Citation: 12. April 2012.] http://en.wikipedia.org/?title=Assembly_line.

[3] Stuart Russel, Peter Norvig. *Artificial Intelligence, A modern approach.* 2003

[4] Decision theory. [Online] LessWrongWiki, 2012. http://wiki.lesswrong.com/wiki/Decision_theory

[5] Dung, Truong My. Decision Theory. [Online] 2005. http://www.fit.hcmus.edu.vn/~tmdung/courses/SoftApp/Ch5_TMD_2005.pdf

[6] Hansson, Sven Ove. *Decision Theory: A Brief Introduction.* 1995

[7] Decision Making. [Online] August 2010. http://decision-making-role.blogspot.fi/2010/08/decisions-under-certainty-risk-and.html

[8] Jones, James. Decision Theory. [Online] Richland University, 2003. http://people.richland.edu/james/summer03/m160/decision.html

[9] Game Theory. [Online] Investopedia. http://www.investopedia.com/terms/g/gametheory.asp#axzz24XmZTdmK

[10] Thrun, Sebastian. *Particle Filters in Robotics.* 2002. Uncertainty in AI

[11] Jens-Steffen Gutmann, Wolfram Burgard, Dieter Fox, Kurt Konolige. An Experimental Comparison of LocalizationMethods. [Online] 2002. http://www.cs.cmu.edu/~biorobotics/motion/www/papers/sbp_papers/integrated3/guttman_compare.pdf.

[12] Werger, Barry Brian. Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. [Online] 1998. http://www.cs.uml.edu/~holly/91.549/readings/werger.pdf

[13] Ashley W. Stroupe, Kevin Sikorski, Tucker Balch. Constraint-Based Landmark Localization. [Online] 2002. http://pdf.aminer.org/000/560/920/constraint_based_landmark_localization.pdf

[14] Wolfram Burgard, Sebastian Thrun. Markov Localization for Mobile Robots in Dynamic Environment. [Online] 1999. http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume11/fox99a-html/jair-localize.html

[15] Thrun, Sebastian. Bayesian Landmark Learning for Mobile Robot Localization. [Online] 1997. http://robots.stanford.edu/papers/thrun.landmark.pdf

[16] Dieter Fox, Wolfram Burgard, Frank Dellaert, Sebastian Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. [Online] 1999. https://www.aaai.org/Papers/AAAI/1999/AAAI99-050.pdf

[17] Global Digital Signal Processors (DSP) Market by Intellectual Property (IP), Design Architecture & Applications (2011 – 2016). [Online] Digital Signal

Processors Market, February 2012. http://www.marketsandmarkets.com/Market-Reports/global-digital-signal-processing-market-157.html

[18] Gottschalk, Mark A. Sensor make cars smarter. [Online] DesignNews, 10. 6 1997. http://www.designnews.com/document.asp?doc_id=225450&dfpPParams=ind_184,industry_auto,aid_225450&dfpLayout=article

[19] Kempe, Worker. *Inertial MEMS - Principles and Practice.* místo neznámé : Cambridge University Press, 2011

[20] Vigna, Benedetto. *Physical Sensors Drive MEMS Consumerization Wave.* STMicroelectronics

[21] Accelerometer. [Online] Wikipedia, 2012. http://en.wikipedia.org/wiki/Accelerometer

[22] Bernstein, Jonathan. An Overview of MEMS Inertial Sensing Technology. *ModernMedicine*

[23] Gyroscope. [Online] Wikipedia, 2012. http://en.wikipedia.org/wiki/Gyroscope

[24] Vibrating structure gyroscope. [Online] Wikipedia, 2012. http://en.wikipedia.org/wiki/Vibrating_structure_gyroscope

[25] Gyroscope. [Online] SensorWiki.org. http://sensorwiki.org/doku.php/sensors/gyroscope

[26] Bo Zhang, MTE Kahn. *Overview and Improving Fiber Optic Gyroscope Based on MEMS/NEMS Fabrication*. [editor] Institute of Physics Publishing. Journal of Physics, 2006. International MEMS Conference. Vol. 34

[27] Fibre Optic Gyroscope. [Online] Wikipedia, June 2012. http://en.wikipedia.org/wiki/Fibre_optic_gyroscope

[28] Difference Between RFID and NFC. *DifferenceBetween.net.* [Online] 2012. http://www.differencebetween.net/technology/difference-between-rfid-and-nfc/

[29] Radio-frequency identification. [Online] Wikipedia, 2012. http://en.wikipedia.org/wiki/Radio-frequency_identification

[30] A New RFID Privacy Framework Has Been Taken Up by Europe. *RFIDcanada.* [Online] November 2011. http://www.rfidworld.ca/a-new-rfid-privacy-framework-has-been-taken-up-by-europe/333

[31] RFID Tag. *Direct Industry.* [Online] 2012. http://www.directindustry.com/prod/premo-corporation/rfid-tags-50493-387062.html

[32] Zach Shelby, Carsten Bormann. *6LoWPAN.* Wiley, 2009

[33] ZigBee Alliance. [Online] http://www.zigbee.org/

[34] M2M Comunity for machine-to-machine, embedded wireless, and connected consumer devices development. [Online] www.m2m.com

[35] Bluetooth technology. [Online] www.bluetooth.com

[36] Hunn, Nick. *Essentials of Short-Range Wireless.* Cambridge University Press, 2010

[37] The Wireless Embedded Internet. [Online] http://6lowpan.net/

[38] Jaacán Martinez, José L. M. Lastra. *Application of 6LoWPAN for the Real-Time Positioning of Manufacturing Assets.* Tampere [Online] 2011. http://www.iab.org/wp-content/IAB-uploads/2011/03/Martinez.pdf

[39] Ricardo Silva, Jorge Sá Silva, Claudio Geyer, Valderi Leithardt, Fernando Boavida. *Use of GPS and 6LoWPAN in Mobile Multi-Sink Wireless Sensor Networks - Issues and Feasibility.* 2011

[40] Hugo Haas, Allen Brown. Web Services Glossary. [Online] W3C. http://www.w3.org/TR/ws-gloss/

[41] Thuraisingham, Bhavani. *Secure Semantic Service-Oriented Systems.* Auerbach Publications, 2010. ISBN: 978-1-4200-7331-7

[42] Minor, Johannes. *Bridging OPC UA and DPWS for Industrial SOA.* Tampere : Tampere University of Technology, 2012

[43] Jose L. Martinez Lastra, Ivan M. Delamer. *Semantic Web Services in Factory Automation: Fundamental Insights and Research Roadmap.* IEEE Transactions on Industrial Informatics, Vol. 2, No. 1, February 2006

[44] Web Services architecture overview. [Online] DeveloperWorks, 2000. http://www.ibm.com/developerworks/webservices/library/w-ovr/

[45] SOAP-over-UDP Version 1.1. [Online] OASIS, 2009. http://docs.oasis-open.org/ws-dd/soapoverudp/1.1/os/wsdd-soapoverudp-1.1-spec-os.html

[46] SOAP. [Online] Wikipedia, 2012. http://en.wikipedia.org/wiki/SOAP

[47] Overview of SOAP. [Online] Oracle, 2002. http://java.sun.com/developer/technicalArticles/xml/webservices/

[48] WSDL. [Online] Wikipedia, 2012. http://en.wikipedia.org/wiki/Web_Services_Description_Language

[49] Web Services Description Language. [Online] W3C, 2001. http://www.w3.org/TR/wsdl

[50] Devices Profile for Web Services. [Online] OASIS, 2009 http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.pdf

[51] G. Moritz. DPWS for 6LoWPAN. [Online] 2010. http://tools.ietf.org/html/draft-moritz-6lowapp-dpws-enhancements-01

[52] Device Profile for Web Services. [Online] Wikipedia, 2012 http://en.wikipedia.org/wiki/Devices_Profile_for_Web_Services

[53] Understanding Devices Profile for Web Services, WS-Discovery, and SOAP-over-UDP. [Online] MSDN, 2012 http://msdn.microsoft.com/en-us/library/dd179231.aspx

[54] Devices Profile for Web Services. [Online] Doxygen Trac, 2007 http://trac.e-technik.uni-rostock.de/projects/ws4d-soap/doxygen/stable/html/dpws_page.html

[55] Anurag Kumar, D. Manjunath, Joy Kuri. *Wireless Networking.* 2008

[56] Guoqiang Mao, Barıs¸ Fidan and Brian D.O. Anderson. Wireless Sensor Network Localization Techniques. [Online] 2006. http://sydney.edu.au/engineering/electrical/people/guoqiang.mao/UserFiles/File/publications/Mao06Wireless.pdf

[57] *Localization Algorithms in Wireless Sensor Networks:*. Pal, Amitangshu. 1, 2010, Network Protocols and Algorithms, Vol. 2

[58] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal. *Wireless sensor network survey.* Elsevier, 2008, Computer Networks

[59] Piedrahita, Giovanny Andrés., Guayacundo, Diana Marcela. *Evaluation of Accelerometers as Inertial Navigation System for Mobile Robots.* 2006. Robotic Symposium

[60] Rong Peng, Mihail L. Sichitiu. Angle of Arrival Localization for Wireless Sensor Networks. [Online] 2006. http://www4.ncsu.edu/~mlsichit/Research/Publications/aoaLocalizationSecon06.pdf

[61] Neal Patwari, Alfred O. Hero III, Jose A. Costa. *Learning Sensor Location from Signal Strength.* 2007, Advances in Information Security

[62] Eddie C.L.Cahn, George Baciu, S.C.Mak. *Using Wi-Fi Signal Strength to Localize in Wireless Sensor Networks.* 2009. International Conference on Communications and Mobile Computing

[63] Sarah Spieker, Christof Röhrig. *Localization of Pallets in Warehouses using Wireless Sensor Networks.* 2008. 16th Mediterranean Conference on Control and Automation

[64] Christof Röhrig, Sarah Spieker. *Tracking of Transport Vehicles for Warehouse Management.* Nice, 2008. International Conference on Intelligent Robots and Systems

[65] Hyunsook Kim, Eunhwa Kim, Kijun Han. *An Energy Efficient Tracking Method.* Springer Berlin, 2006

[66] Iozan L.I., Rusu C., Collin J., Takala J. *A study of the external factors that affect the measurement data of a MEMS gyroscope sensor - Towards an inertial navigation system.* 2010. 9th International Symposium on Electronics and Telecommunications (ISETC)

[67] Eagle S. Jones, Stefano Soatto. *Visual-Inertial Navigation, Mapping and Localization: A Scalable Real-Time Causal Approach.* 2010, Internation Journal of Robotic Research

[68] Juan C. Alvarez, Antonio M. López, Rafael C. González, Diego Álvarez. *Pedestrian Dead Reckoning With Waist-Worn Inertial Sensors.* 2012. Instrumentation and Measurement Technology Conference

[69] Mouser Electronics. [Online] http://fi.mouser.com/ProductDetail/Atmel/ATZB-24-A2/

APPENDIX A: USER MANUAL FOR GUI

Graphical User Interface (GUI) provides operator a functional interface for not only pallet monitoring but also for a control and dynamic interaction with the process of pallet localization. There are available four different GUI environments or web pages for operator disposal, each serving for specific purpose – welcome page, design page, online monitoring page and offline monitoring page. The GUI is called APIS and stands for Advanced Pallet Information System.

Welcome Page

Welcome Page is a home page of developed GUI serving at the same time as the main finger-post referring to all constituent subpages, too. Its appearance is captured in Figure A.1, where labelled circles represent buttons to the individual subpages. The user can choose by clicking on appropriate bubble from three different modes:

- Design new line – designing brand new map of conveyor system
- Real-time monitoring – real-time tracking of pallets
- Offline monitoring – offline tracking of pallets based on event stored in database

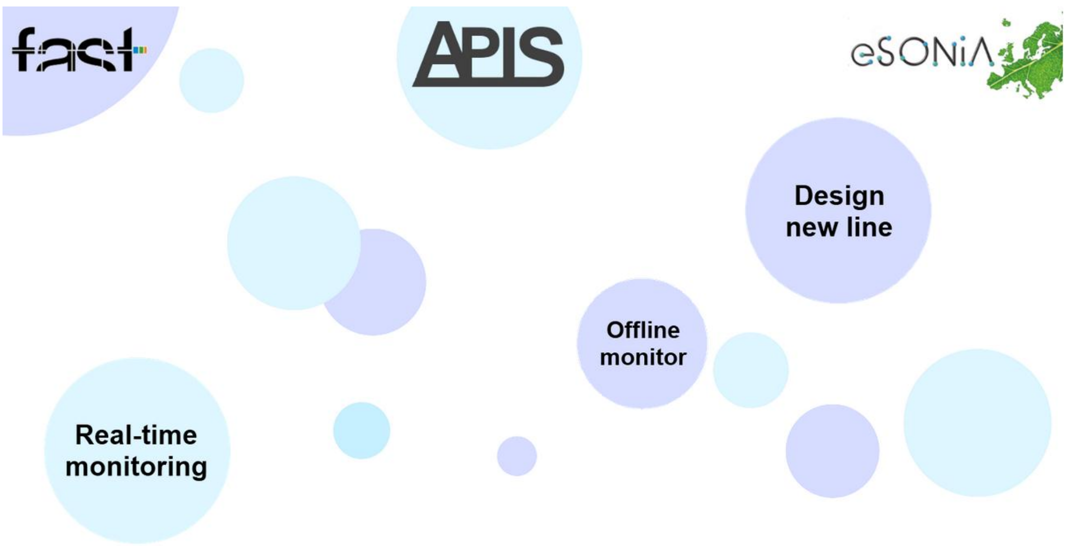


***Figure A.1** Welcome Page of APIS with three bubbles that creates hyperlinks to page of Real-time monitoring, Offline monitoring and Design of new line*

Design Page

Design page is an interface for development of own process model represented by a map of the conveyor system. The process of map design consists of two phases – a map structure design and a parameter design located in the page in two separate tab panels.

Map structure design – it is a first phase of map modelling, where user builds a raw structure of a conveyor system as a combination of well-defined blocks similarly to LEGO principle as shown in Figure A.1. The building blocks are available in block panel and each type represents certain section of conveyor system such as right turn, left turn, straight and others. Summary of all available blocks is listed Table A.1. Block by block is placed into arbitrary position in building area by simple drag and drop operation. If necessary there is also a possibility to rotate all blocks to desired orientation by clicking *Rotate* button. If a map is completed, its identification name must be inserted and the map is saved by pressing save button. As soon as the map is saved to server, the map designer can jump into second phase – Parameter Design.

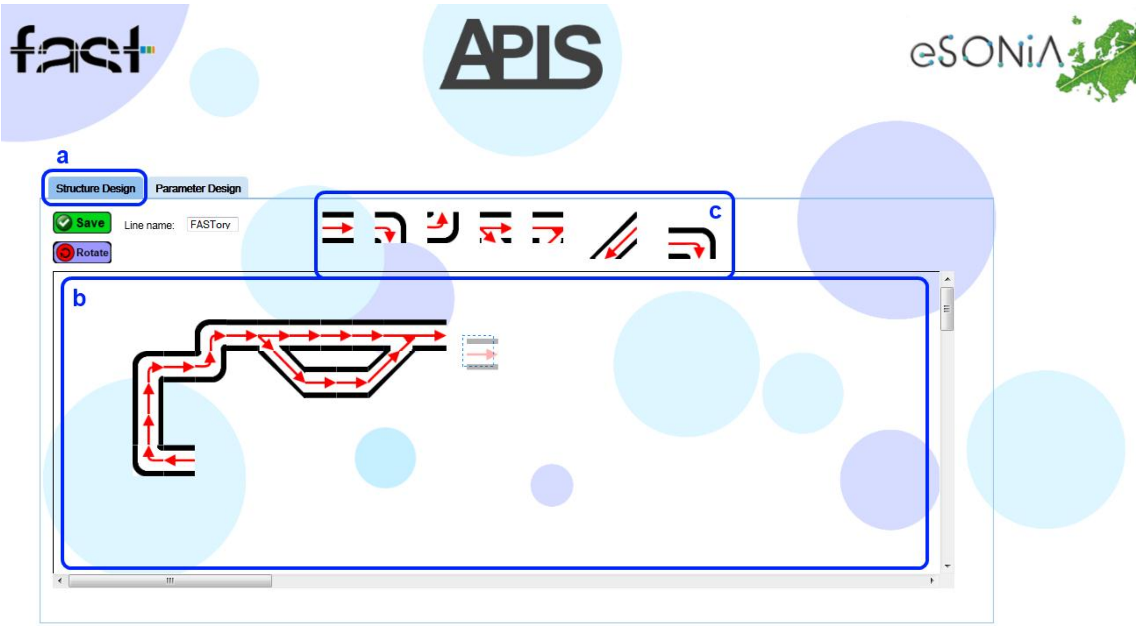
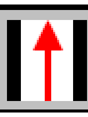
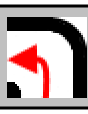
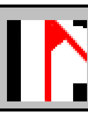
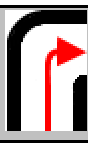


Figure A.2 *The Structure design tab of Design page – a) active tab, b) building area, c) building blocks panel*

Table A.1 ***Table A.1*** *List of building blocks*

Block	Conveyor representation description	Icon
STRAIGHT	Straight section of conveyor	
RIGHT TURN	Right turn of angle 90°	
LEFT TURN	Left turn of angle 90°	
RIGHT Y-JUNCTION	Junction where pallet can continue either straight or turn 30° to the right	
RIGHT Y-CONNECTION	Connection which a pallet can reach either by straight movement or by right turn form side	
RIGHT 30° TURN	Diagonal section of conveyor section that follows RIGHT Y-JUNCTION	
RIGHT LONG TURN	Right 90° turn - differs to RIGHT TURN in size and represents longer section of turning. The reason is better map alignment to real conveyor system	

Parameter Design – it is the second phase of map design that is not mandatory for building of sufficient working map, but it is a phase that specifies the map a bit more to allow localization algorithm to work better. During Parameter Design an operator can freely change parameters of each block by simple clicking on particular block and setting appropriate properties from option panel. For each block three possible properties can be set – S*top*, *RFID* and *WS*, of which description can be found in Table A.2. Anytime if necessary, there is a possibility to go back to Structure design and modify the map of which blocks are kept in building area. However, if the save button in map Structure design is pressed, all Parameter design properties will be lost.

Table A.2 *Additional block properties*

Name	Property description
Stop	Defines a block, where pallet is supposed to stop, mostly by a stopper. Algorithm uses this information to distinguish better between pallet stop caused by a stopper or by accident (obstacle, conveyor failure etc.)
RFID	Defines a block that is equipped by RFID reader, which provides helps pallet localization after receiving an RFID event.
WS	Defines block “membership” to certain workstation which is part of. Having this property allow algorithm to assign pallet, occupying certain block, to appropriate workstation.

In this manner, each block by block is parameterized and immediately saved by saved button. The change is confirmed by visual representation as might be seen in Figure A.3, where a group of blocks were assigned to Workstation 1 are thus those blocks are embodied into purple rectangular to show borders of the workstation.

Finally, even though the map is already saved in server, there is also possibility to export the map into single URL string for storage purposes. Thus, if server crushes the map can be restored again by just inserting the URL string into a web browser command line. (Only in case of a new designed map, FASTory line is pre-coded and loaded always automatically with server start). Now the model map is ready to be used.

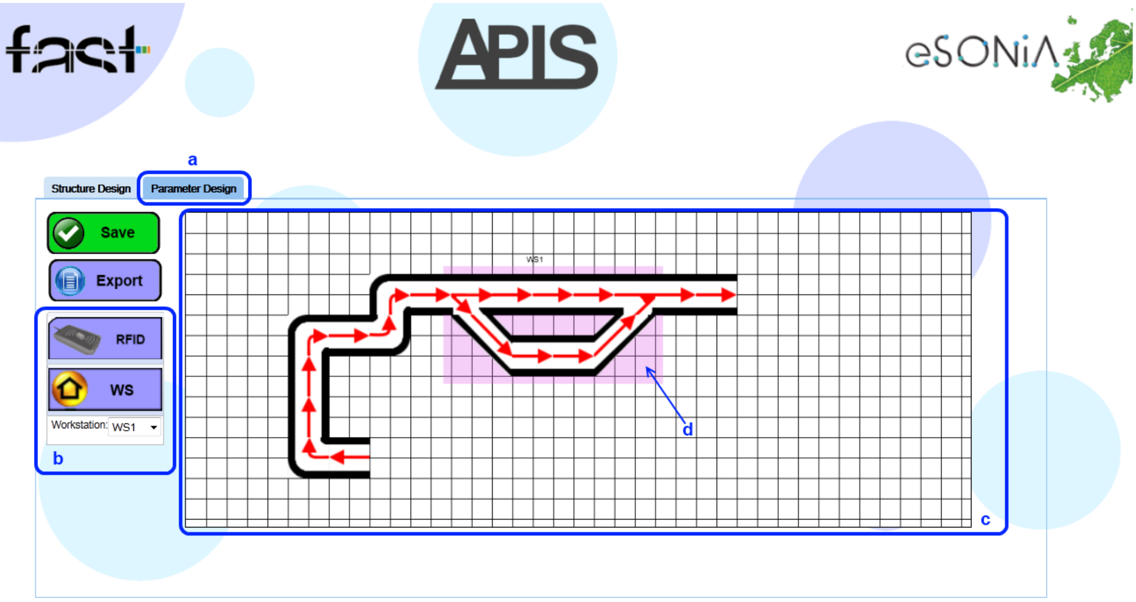


***Figure A.3** The Parameter Design tab of Design Page: a) active tab, b) property setup area, c) working area, d) example of workstation membership representation*

Real-time monitoring Page

This web page serves as a main interface for real-time monitoring and control of pallet localization system and can be seen in Figure A.4. The page consists of two main distinguishable areas a *Monitoring area* and *Control area.*

Monitoring area – monitoring area truly dominates the page. It is the main visualization element, where the map of conveyor system is displayed as well as the locations of all the pallets. To improve human readability and overall visualization credibility, there is also a background image of CAD model of the real production line displayed behind the map that matches in dimensions with the designed map. In a case that a different production line is designed and monitored than the FASTory line, of which image is available, there is a possibility to hide the image to do not confuse operator.

Pallets are represented in the monitoring area by a line of colourful circles, where a head symbolizes current pallet position and following circles are creating a tail symbolizing a historical pallet track. The diameter of circles is getting smaller from the head towards the track tail. The length of the tail can be freely chosen from 1 (only head) up to 20 by intuitive slider adjustment located in Control area.

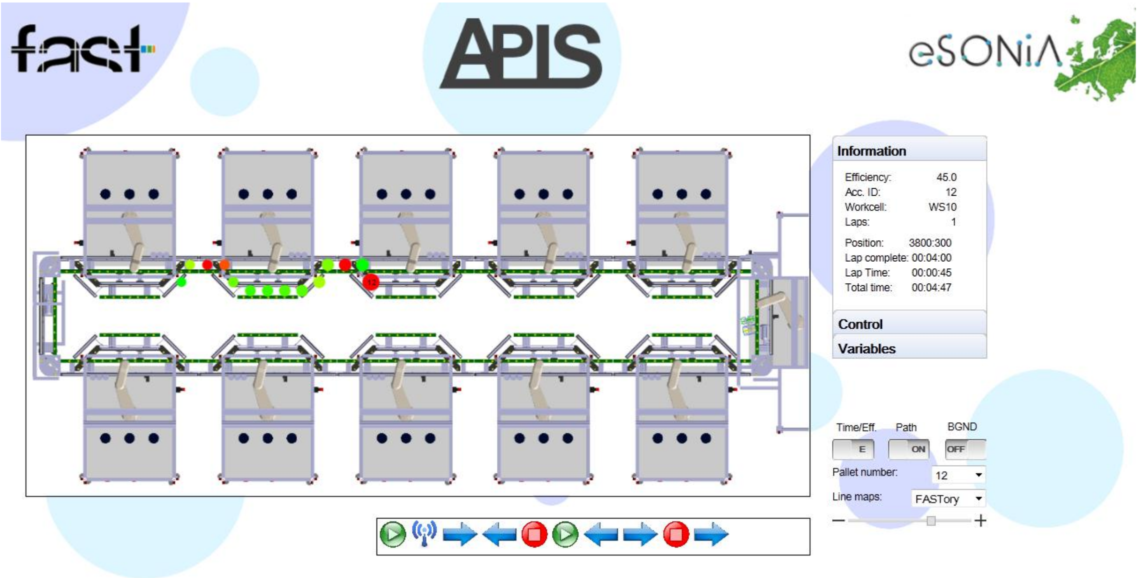


Figure A.4 *Example of real-time monitoring page*

There are available two visualization modes how a pallet can be displayed in Monitoring area. The basic mode is a **time mode**, where a time spent by a pallet in each position is represented by a smooth change of circle colour. Hence, when a pallet enters a block, the circle has green colour but while the time runs and pallet does not move, the colour is changing into orange and later to red colour. The time mode with long tracking path is a perfect tool for analysis of pallet dynamics because an operator can easily distinguish where each pallet stopped, get delayed and which section of conveyor part was smoothly ridden. Information about pallet delays can also serve to determine production line bottlenecks, but such a tool would be out of scope of this thesis. Example of time mode is illustrated in Figure A.5.

Efficiency mode is another visualization mode that graphically displays the efficiency of event generation. As was already said, every block of map is defined by certain number of input events – the events that would be received in ideal case when a pallet enters the block. The difference between ideal case and real one is represented by colour change as follows. No event missing means the circle is displayed in green colour, one missing event in orange and two and more missing events in red colour. The efficiency mode gives an overview how well the events are generated, in which section of line a sensor has difficulties of event generation and in fact also to graphically recognize whether the pallet tracking is actually effective or not.

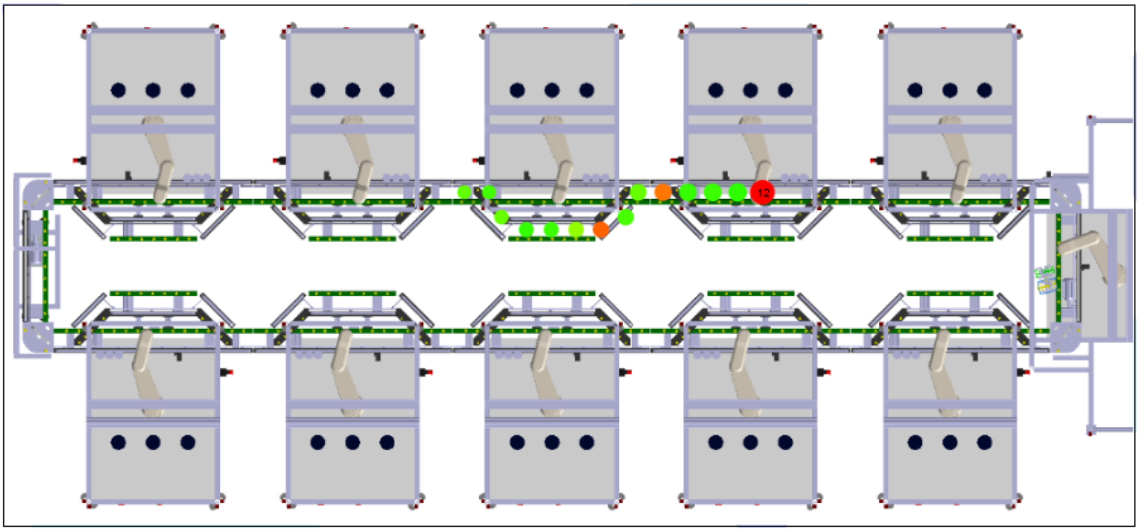


Figure A.5 *Example of main monitoring area with time mode*

Inseparable part of monitoring area is also an *Event area*. Event area shows a line of received events, represented by appropriate icons, for one chosen pallet in order they were received. Their graphical representation in Event area is shown in Figure A.6.

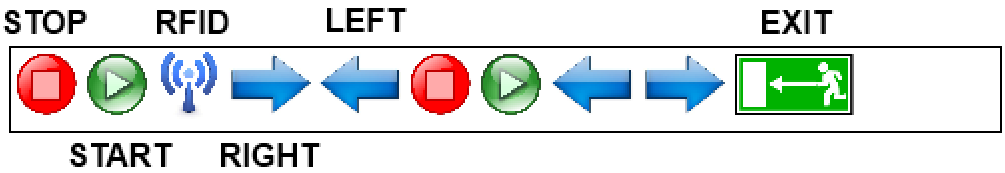


Figure A.6 *Example of Event area with labelled events*

Control area – in control area of the page there are located all adjusting elements that an operator needs to interact with localization algorithm. They are arranged in dedicated panels called *Information panel*, *Control panel*, *Variables panel* and *Display panel* and next paragraphs will bring their description.

The Information panel, which is shown in Figure A.7, displays information of all pallet properties that serve for pallet movement analysis. List of included properties is presented in Table A.3.

***Table A.3** Description of elements shown in information panel*

Property	Description
Efficiency	Overall efficiency is calculated as a percentage ratio between total amount of input events and real amount of events
ID	Pallet identification number
Workstation	Actual workstation in which the pallet is located
Laps	Number of laps, the pallet did since was introduced to the system
Position	Actual pallet position in millimetres measured from upper left corner of production line (WS8)
Lap complete	A time of last complete lap
Lap time	A time of actual lap
Total time	Total time of pallet movement

The Control panel shown in Figure A.7 has the same function as a remote control for TV. With its help an artificial pallet can be introduced to the system as well as it is possible to generate artificial events. This panel serves primarily for testing purposes and during normal monitoring phase it should not be used. The artificial pallet is created by writing its ID into text box in the centre of navigation star (a number 172 is displayed there) and then by pressing arbitrary events. If a user wants to use RFID event, a necessary name of workstation must be inserted into text box on the right of RFID button in format WSx, where x is a number of workstation. Furthermore, besides event button there are also two special buttons. The button *Reset* calls a reset function in server that deletes all stored values of model and restores its initial state. There is also a button *Calibration* that initialize a manual calibration process – recalculation of all process coefficients based on calculated overall efficiency.

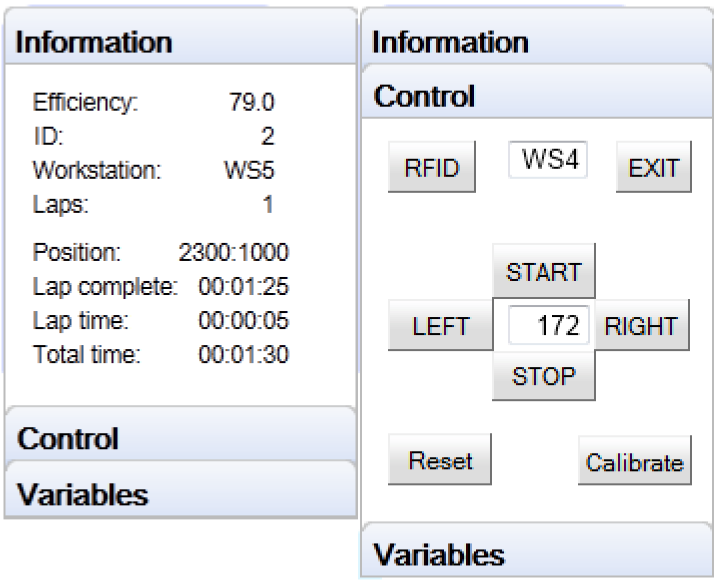


***Figure A.7** Example of Information panel and Control panel*

The purpose of Variables panel is to provide general overview of all process model coefficients. The user can freely change arbitrary coefficient in order to improve the process. For each coefficient, there is available a text box, where a user can write down a desired value. Right next to it, there are two numbers, the first black number represents current value of the coefficient stored in model while the grey number in square brackets informs a user about initial value of the coefficient before any change was done. In a Table A.4 a list of all adjustable coefficients is presented together with their description and initial values. In addition, in Figure A.8 there is shown a graphical representation of Variables panel.

Table A.4 *List of all adjustable process coefficients*

Subpanel	Coefficient	Initial value	Description
Events	Eve 1	RFID	First initializing event necessary to bring pallet to 50% of initialization process
	Eve 2	UNDEFINED	Second initializing event necessary to bring pallet to 75% of initialization process (initially skipped)
	Eve 3	UNDEFINED	Third initializing event necessary to bring pallet to 100% of initialization process (initially skipped)
Coefficients	Forward	1.2	A value of forward coefficient of C_0
	Backward	1.9	A value of backward coefficient of C_0
	Used Way	1.2	A value of coefficient C_h, representing used path or way
Searching size	Forward	80	How many block forward should algorithm search
	Backward	20	How many block backwards should algorithm search
Others	Move time	730	Time in milliseconds for generating of TIME event, in other words, how long it takes to pallet to get from one block to another block – pallet speed.
	History check	5	How many blocks backward in direction of pallet should be checked for missing events

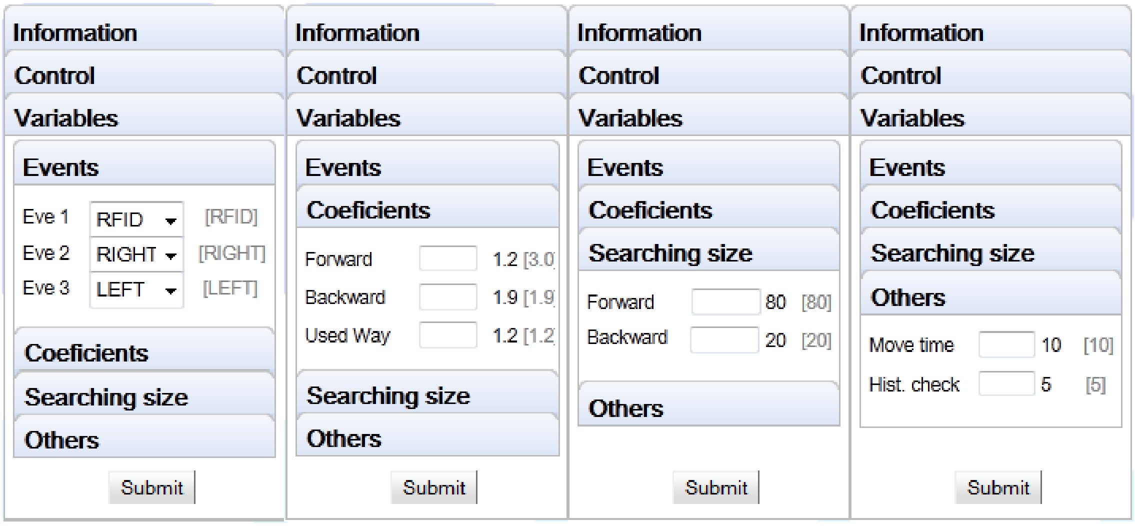


Figure A.8 *Example of Variables panel*

Display panel that is shown in Figure A.9 is used for visualization control in Monitoring area. With it help, a user can adjust arbitrary display properties. It consists of four sliders, two list boxes and one label and for better lucidity, their list and description is going to be again presented in Table A.5.

Table A.5 *Display panel*

Adjusting element	Description of function
Time/Eff	Switch between time mode and efficiency mode
Path	Enabling or disabling visualization of a map of blocks
BGND	Enabling or disabling visualization of background image of production line
Pallet number	Pallet number selection – in case of one pallet in system, only its number is listed there, if more pallets are in system, then besides their numbers also an option *all* is added to list box in order to track all pallets at the same time – however for this scenario time or efficiency mode is disabled and pallets are displayed in random colours.
Line maps	Production line selection – names of all maps stored in system are listed there, by picking a map for the list, the server map as well as the map in monitoring area is automatically refreshed, updated but all new configuration data that were saved for the particular map are lost. By default there is available a line called FASTory.
Tail slider	Intuitive slider for adjusting a length of pallet tail
Notification label	The label servers as a room for displaying all important notifications – confirmations, errors etc.

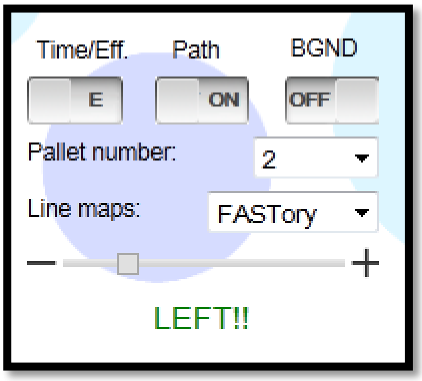


Figure A.9 *Display panel*

Offline Monitoring Page

The Offline monitoring page illustrated in Figure A.10 is an interface that serves for retrospective purposes, when a user is for some reason interested in a special event that occurred in a past. For instance, in a case of lost pallet, the user can check at exactly which place and at what time the pallet got lost what should contribute with high probability to its discovery. Another example can be a case of a defective product, when a pallet with the product can be traced back into the history in order to check whether the pallet accomplished all necessary workstations or not. The appearance of the web page is shown in Figure A.10.

The control of Offline Monitoring Page is very simple and since the monitoring area and vast majority of control area of this page is similar to Real-time Monitoring page, the emphasis is going to be put mainly on those adjusting elements that are specific for this page.

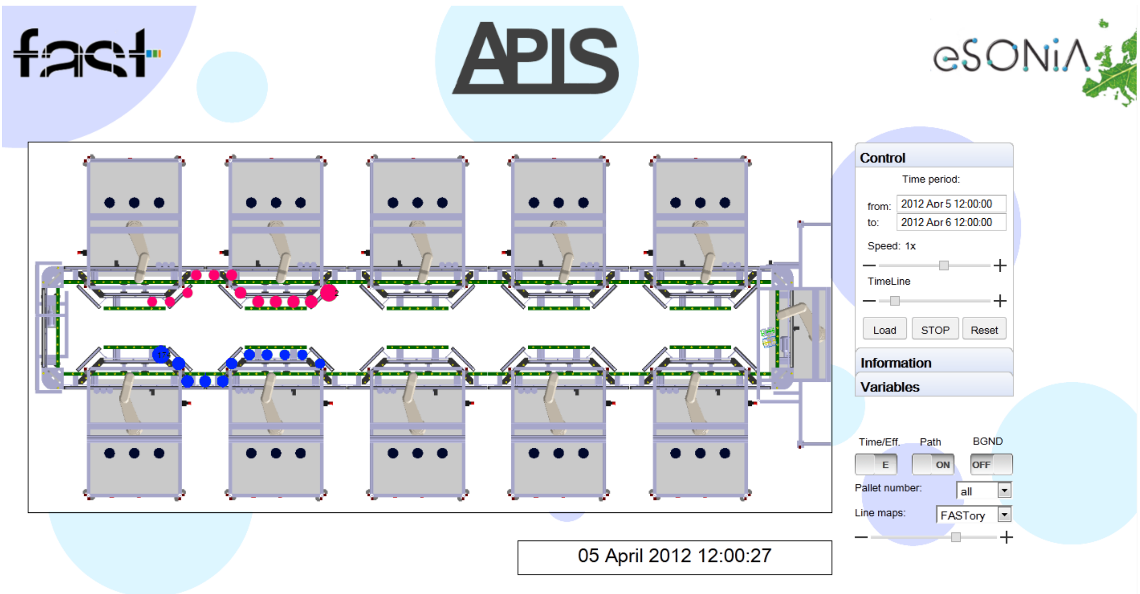


Figure A.10 *Offline Monitoring Page*

The entry point for this page is a Control panel shown in Figure A.11, where all functions for retrospective monitoring are located and which is shown in. In a top part of the panel, there are two text fields, where a user chooses freely a desired time period in a past. By pressing “Load” button, the request is sent to the server, which extracts from a database all stored events for the time period and makes retrospective calculation of pallet locations. As a result a single JSONP object with all pallet locations and their timestamps is created and sent back to the web page. Internal algorithm of the web page parses it and displays it to the user. Both adjustable speed of monitoring and position in the time period are functions that should help a user to find desired target in longer time period. As soon as the button “START” is pressed the retrospective monitoring is started and its virtual time is displayed in a black rectangular in a bottom of the page. A user can whenever stop the monitoring by pressing “STOP” button: Repeatable request to server for new data is allowed.

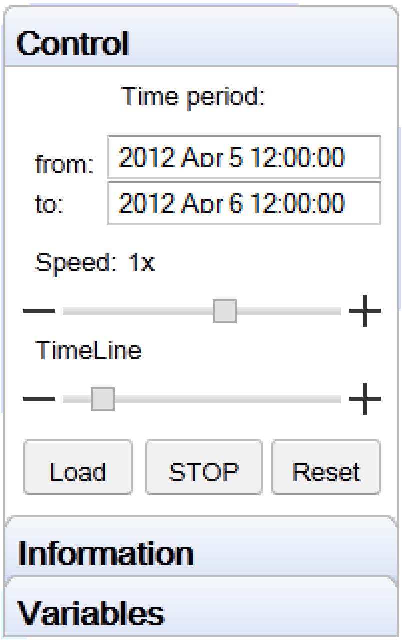


Figure A.11 *Control Panel of Offline Monitoring Page*