

Event-based and Multi Agent Control of an Innovative Wheelchair

U Diomin¹, P Witczak² and R Stetter³

¹ Brest State Technical University, Brest, Belarus

² University of Zielona Gora, Zielona Gora, Poland

³ Hochschule Ravensburg-Weingarten, Weingarten, Germany

E-mail: stetter@hs-weingarten.de

Abstract. Due to the aging population more and more people require mobility assistance in form of a wheelchair. Generally it would be desirable that such wheelchairs would be easy to use and would allow their users the possibility to move in any direction at any time. Concepts which allow such movements are existing since many years but have for several reasons not found their way to the market. Additionally for semi-autonomous (assisted) operation and fully autonomous operation (e. g. an empty wheelchair driving to its charging station) the control task is much less challenging for such drive system, because no complex manoeuvres needs to be considered and planned. In an ongoing research a drive system for a wheelchair was developed which offers such possibilities employing a relatively simple mechanical design. This drive system is based on a certain steering principle which is based on torque differences between different wheels. This allows a relatively simple mechanical design but poses challenges on the control of the vehicle. This paper describes two possible approaches to address this challenge – the use of an event based control and the application of multiple software agents. Both approaches can solve the control problem individually but can also complement each other for better system performance. The paper starts with a description of the wheelchair drive system. Then the asynchronous event based control software is described as well the multi agent based approach. The next sections report the results of the experiments and discuss the further improvements.

1. Semi-autonomous Wheelchair

Wheelchairs are currently provided mainly for handicapped persons however, the rapid growth in the elderly population suggests that the numbers of electric wheelchair users will soon increase dramatically ([1], [2]). In general, electric wheelchairs are difficult to maneuver especially for elderly people who have little experience using a joystick to operate a driven wheel system. Current wheelchairs need a complex series of movements resembling parallel automobile parking when he or she wants to move sideways.

A major achievement for any vehicle or mobile robot is the ability to move freely in any direction and performing rotation at any arbitrary position and orientation. This can be realized by obtaining a locomotion system with the ability to perform a holonomic and omnidirectional motion [3]. Both



abilities are necessary for realizing better maneuver in a crowded area or restricted space. A non-holonomic omnidirectional mechanism requires preliminary maneuvering for reorientation of steering wheel which may affect the time-travel cost and the spaces needed. The holonomic omnidirectional mechanism has better mobility and also less complexity in design compared with a non-holonomic omnidirectional mechanism.

During the last decades several approaches towards better maneuverability, usually in the form of omnidirectional mobile system were presented; Wang provides a good overview [1]. Notable is additionally the ongoing long-term research project “Omni” concerning similar systems [4]. In spite of the fact that several systems were proposed none are today available on the market. One of the causes of this can be the high complexity of the solutions. This paper presents a rather simple solution for a holonomic omnidirectional drive train for a wheelchair. The simplicity of the design is possible, because the steering is based on torque differences.

1.1. Torque steering system

Autonomous vehicles such as mobile robots and the respective steering systems have been successfully developed and built for some years (Anderson&Jones [5], Ashmore&Barnes [6], Dillard [7]). The distinctive quality of the steering system used is the dynamic behaviour. This innovative steering system that is already registered as a patent ([8], [9]) is based on the concept to use the torque of drive motors (more exactly the torque differences between wheels) to steer four independent axes of a vehicle. This system can be used for production vehicles [10], cars [11] and wheelchairs, as well. The principal steering system is shown in Figure 1.

In this example a vehicle consists of four drive motors which are fastened on arms that may freely rotate. These arms have no drive or brake, only an angle encoder is attached at the end of each axle. These angle encoders measure the angle of the motor and the wheel with regard to the vehicle platform. The distinct characteristic of the innovative drive system is the absence of dedicated steering motors. By means of angle encoders applied at the four steering axes and highly dynamic control algorithms it is possible to steer such vehicles only by means of the four drive motors (compare Figure 1).

Each of the wheels on the short axle can be directed into the desired position by means of the torque applied on the wheel. This could take place sequentially for each individual wheel but also simultaneously, if the control allows different torque on all wheels. This characteristic allows simpler and simultaneously more robust vehicle concepts. It is also a main advantage of this concept that the resulting vehicle is able to drive directly in any direction without time and space consuming turning maneuvers.

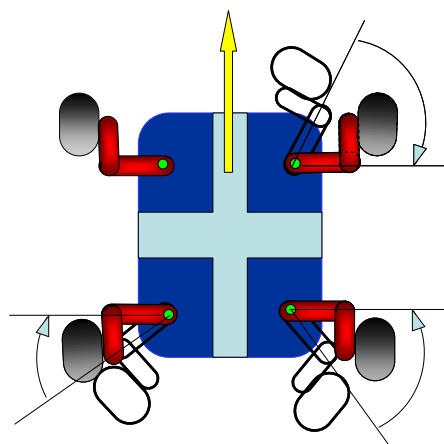


Figure 1. Principal steering system.

Furthermore, a vehicle based on the dynamic drive system is able to turn around its own centre. This characteristic is very important if cameras or other equipment are mounted on such vehicles which can only be used in a certain orientation. The innovative steering system shares these advantages with Omni drive systems (Ashmore&Barnes [6]), but has reduced friction as well as easier controllability and offers the possibility to determine an exact position and orientation from an analysis of the angles of the steering axes and the angles of the drive wheels (odometry).

1.2. Realization of Wheelchair

The wheelchair was designed, developed and realised in several steps. The drive system firstly consists of two front wheels which are each driven by an individual drive motor. The centre of these wheels is outside the vertical rotation axis, therefore torque steering is possible. On this axis a rotary encoder is fixed thus allowing a closed loop control of the steering angle of each front wheel. Additionally, a brake is present on both axes, which allows to brake the respective axis and by this to maintain a constant steering angle. This is advantageous for situations with higher driving speeds and if some obstacles as well as rough or slippery surfaces are present. These brakes are engaged (the angle is maintained) if there is no electrical current; they are powered only when steering angles are desired. The drive system secondly disposes of a third wheel in the back, which can be actively steered by means of a dedicated steering motor (the back wheel is driven by a second designated drive motor). The steering motor also disposes of an encoder functionality thus allowing a closed loop control of the back steering angle. Between the motor and the axis a self-retaining gear system is placed, which ensures a safe behavior also under dynamic operation conditions. The arrangement of the drive train is visible in Figure 2.

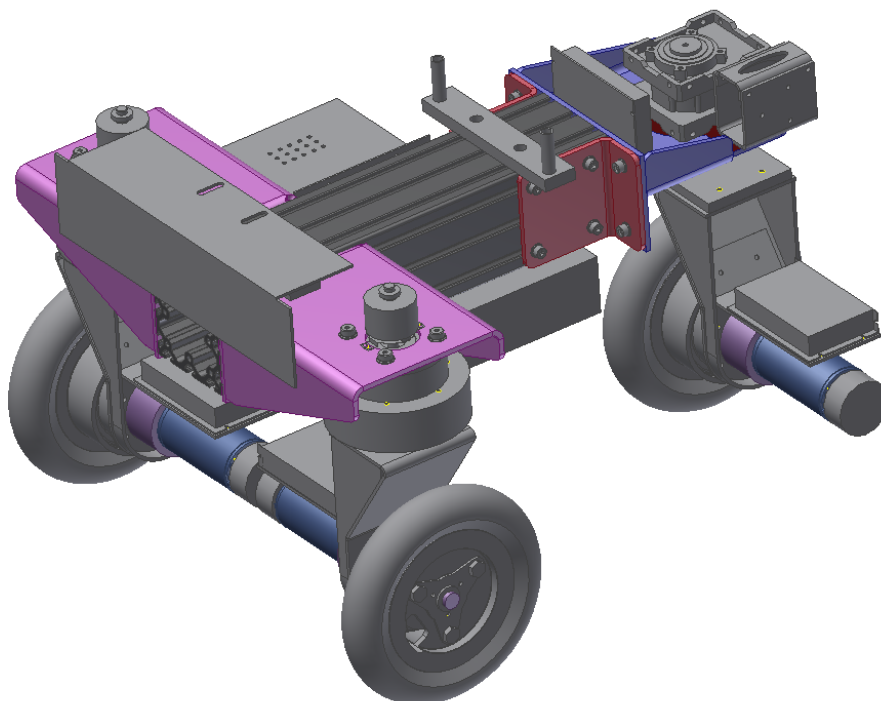


Figure 2. Arrangement of the drive train.

Generally, two driving modes can be realized. The first driving mode, which can be used for fast driving maintains a straight driving angle for the two front wheels and realizes the steering only

through steering of the back wheel and speed differences between the front wheels. This mode allows fast driving, but does not allow, for instance, sideways driving. A second driving mode also relies on the torque steering capabilities of the two front wheels and allows unlimited maneuvering. Figure 3 shows the assembled wheelchair with all equipment such as main computer and battery.



Figure 3. Assembled wheelchair.

2. Asynchronous event-based control software

In the course of the project a mathematical model of the wheelchair was derived. On the basis of this model, the control paradigms can be implemented. In order to maintain the real-time behavior of the system as well as safety of the driver and surroundings, a proper software needs to be done. The modules can be treated as separate agents with own control procedures, i.e. PID control of the wheel velocity and angular module position. Thus the main unit should handle data acquisition, control law reconfiguration and subsequently update respective agents inputs. To achieve this with relatively low computational cost, and to prevent misuse of power, the event-based control was implemented. Its idea bases on the discrete events broadcasted through the system entities (agents). Each event consists of unique identifier, creation time and the necessary data, e.g. new reference velocity, reference angle or display data. All agents are represented virtually within the main framework and as such can control the expiration time of events they handle. Each agent is spawned as thread (or threads). Thus the system emulates real-time environment, i.e. if time lapses the new event is required or buffered value is used, delays between actions are negligible as they are much smaller than the overall inertia of the

system.). Due to the fact that events are broadcasted each agent is notified when events occurs but does not need to handle it. In other words - in the given time interval each agent works in the closed-loop until the event occurs, when they accept the input and follow new control law, until the reference point is reached or changed by the precedent controller.

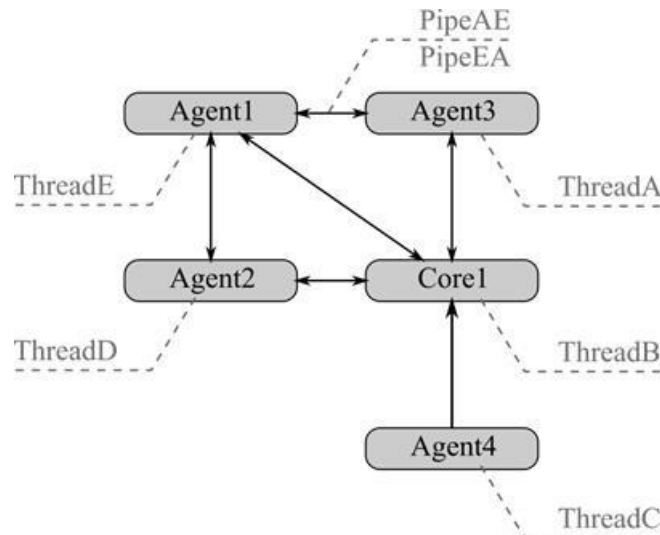


Figure 4. Thread communication scheme.

The on-board CPU and operating system handles priorities of threads, however the quality and computational power of industrial pc used is more than enough to respect the real-time schedule of the system itself. Additionally the computational unit is equipped with the software and hardware watchdogs (software faults detectors) that can restart particular threads or the entire program in order to avoid wheelchair failures due to CPU-related issues. The overall construction of software enables fast and reliable prototyping and changes both in mechanical design and control scheme.

Figure 4 presents the communication scheme between the agents represented by threads. Each Agent in depicted case is represented by a single thread. Events are send and received by unidirectional pipes(e.g. one from ThreadA to ThreadE, and second vice versa). Some agents can exchange the data, but some works only as sensor, collecting and broadcasting data (Agent4, ThreadC). Pipes collect the events that are an object of interest of the represented agent, and send them to the buffer. First-in first-out queue is then resolved internally in each thread. Obvious advantage of the solution appears in the multi-core architecture. The unavoidable disadvantage is performance of the hardware, which limits the real-time appearance of the solution. At the same time due to the inertia of the system and operator real-time can be perceived as fast-enough.

Note that in this case real-time system should not be perceived as defined in i.e. [12][13], so not like a synchronous system that requires quantitative expression of time to describe the behavior of the system. Instead it is asynchronous, event-driven with buffers and variables expiration times. Therefore a task can be performed faster than assumed, and in case of data unavailability the closest sample (i.e. from buffer) is used to preserve functioning of the system. Thus making it possibly faster, and not-less reliable than the conventional real-time systems.

3. Improving multi-agent control technique

The next sections describe the improved control technique based on the multi-agent control approach. In previous researches the multi-agent approach mixed with reinforcement learning (RL) was shown. The approach improves the control of the mobile robot based on complicated steering scheme [14, 15].

The control was done either for simulation or for real robot hardware. The main problem of such software transfer is that real hardware is not ready to be controlled by agents and needs a lot of development to use it.

To describe multi-agent systems one needs to start from the agent definition. An **agent** is an instance that can be viewed as one that perceives its **environment** through the **sensors** and acts upon that environment through the **actuators** [16]. In multi-agent systems the most important receive/perceive ability of the agent is the communication ability. The agents of the system should send messages and answer to each other. This is the main criterion to create multi-agency.

The second important criterion is the intelligence of the agent. An **intelligent** or **rational** agent for each possible percept sequence 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 [16]. Our intelligent agents contain reinforcement learning decision-making system. In reinforcement learning agents improve their behavior by exploration of the environment. To become really intelligent agent should make a lot of *epochs* (couple of iterations). Each epoch the agent starts with a random state and explores the world by making actions. Every action of the agent returns a feedback from the environment as a reward. In this case the evaluation of the reward will be: “the bigger the reward value the better”. The reward value is saved by the agent to the knowledge base using one of the reinforcement learning rules. This knowledge base is used to help in making choice of the next action in the next epochs. The goal of the agent not to only to use maximum valuable steps from the base but also to investigate not visited ones. In our approach we used the improved multi-agent reinforcement learning algorithm with *sharing state*. The description and the framework will be revealed at the end of this chapter.

As it was mentioned above the wheelchair provides two driving modes: *fast mode* as common driving one and the *unlimited maneuvering mode* used when the wheelchair needs to drive along complicated trajectories or in the limited space. So the multi-agent system was developed for working in these two modes.

The asynchronous event-based control framework provides the *control system* of high quality and moreover it can be used as the abstraction layer for control system based on multi-agent approach. The designed multi-agent architecture consists of four agents where three of them are the representation of each of driving modules and the fourth agent is the *virtual head agent*. The *virtual head agent* is the agent that achieves the desired movements of the wheelchair user and sends the cooperating commands to the agents. The decomposition of the wheelchair is shown on the Figure 5.

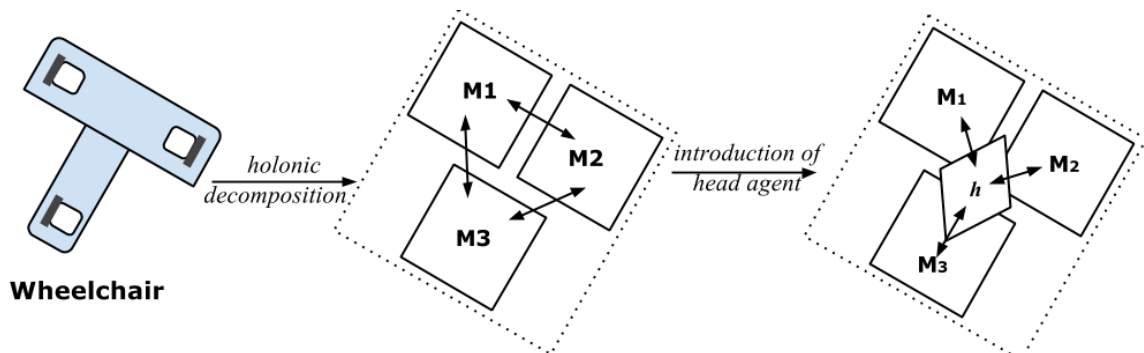


Figure 5. Wheelchair decomposition to multi-agent system with head agent.

The general architecture after decomposition step for multi-agent control system with head agent is shown on Figure 6. Every module provides control $P = \{p_i | i = 1..n\}$ of n input parameters. For every control parameter the agent is using policy π_i and after every step achieves the reward R_i from environment. The Environment/Planning subsystem provides information about the desired speed of

the platform v^d and the global state of the agents $S = \{U_{i=1}^N s_i\} \cup \{s_h\}$, where $s_i \in s$ is the state of the i -th module and s_h is the state of the head agent.

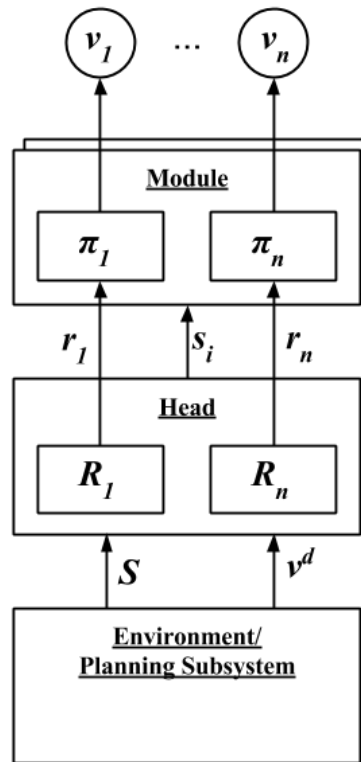


Figure 6. The architecture of the control system.

The intelligence of the agents is made by reinforcement learning approach that is extended to the multi-agent system. The standard reinforcement learning approaches are not suitable for the multi-agent systems. The curse of dimension becomes a huge problem when one needs to deal with more than one agent. In the presented approach an alternative way to learn many agents in one time was applied. The approach uses sharing of the agents' state. The sharing procedure is provided by head agent that sends synchronization command to the other agents. During the process of synchronization agents update their knowledge base due to the Q-Learning reinforcement learning rule. Q-Learning is an off-policy temporal-difference control algorithm [17]. Q-learning is a model-free reinforcement learning technique, which relies on learning an action-value function that ultimately gives the expected utility of taking a given action in a given state and following the optimal policy thereafter.

While reinforcement learning the i^{th} agent executes an action a_i at the current state s_i . Then it goes to the next state s'_i and receives a numerical reward r as the feedback for the recent action [17], where $s_i, s'_i \in s$, $a'_i \in a$, $r \in R$. Ideally agents should explore state space (interact with environment) to build an optimal policy π^* .

Let $Q(s, a)$ – represent a Q-function that reflects the quality of the specified action a in state s . The optimal policy can be expressed in the terms of optimal Q-function Q^* :

$$\pi^*(s) = \operatorname{argmax}_{a \in \mathcal{A}(s)} Q^*(s, a) \quad (1)$$

The initial values of Q-functions are unknown and equal to zero. The learning goal is to approximate the Q-function, (e.g. to find true Q-values for each action in every state using received sequences of rewards). The RL framework with *shared state* is shown on Figure 7.

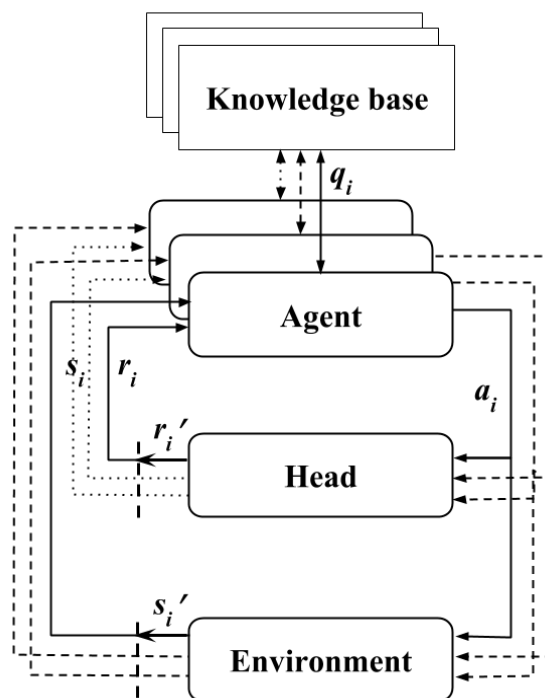


Figure 7. Reinforcement learning framework with shared state.

4. Simulation results

This section describes the results of simulations done with the wheelchair model. The experiments were divided into two parts: *the learning part* and *the driving part* that will be described sequentially.

4.1. The learning part

At this part agents learn how to work in two provided modes. Every driving agent has the driving scheme that is shown at Figure 8. Two agents at the front of the wheelchair have only one control parameter v – the velocity of the wheel. According to that we learn the agent with one output parameter. The third agent that is placed at the back of the wheelchair has two output parameters: wheel velocity v and the angle relative to the wheelchair φ . The full state of the agents and their possible actions are described in Table 1. The error of the wheelchair trajectory is an important parameter that is used as reward for the agents. The agents are learnt in team to drive in two modes.

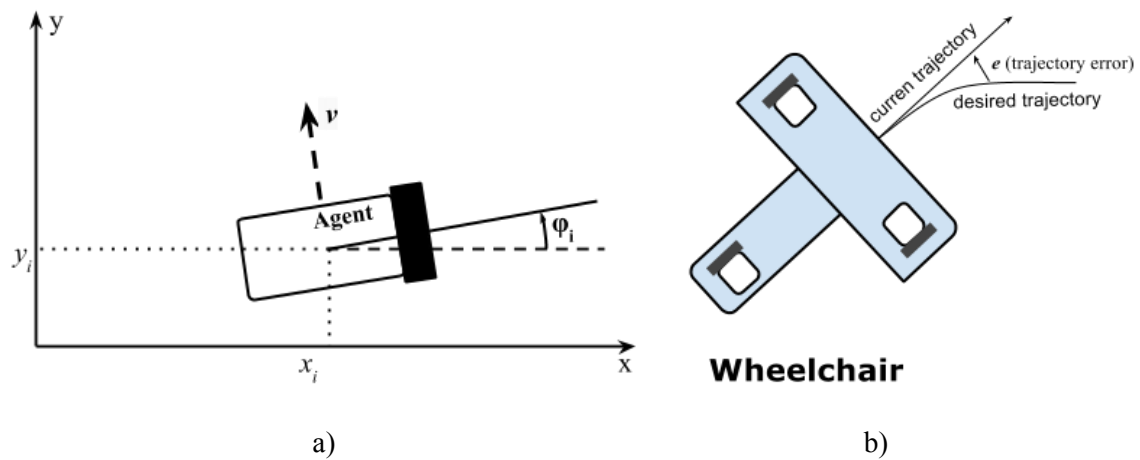


Figure 8. a) The graphical representation of the agent-module. b) The wheelchair trajectory error calculation

Table 1. Reinforcement learning parameters of the agents

Agent	Controlled parameters	Actions	The state parameters
<i>Front Agent</i>	v – wheel velocity	$+v, -v$: increase and decrease the velocity	v – current velocity d – desired trajectory of the wheelchair
<i>Back Agent</i>	v – wheel velocity φ – rotation angle	$+v, -v$: increase and decrease the velocity; $+\varphi, -\varphi$: increase and decrease the rotation angle;	v – current velocity $+\varphi, -\varphi$: increase and decrease the rotation angle; d – trajectory error of the wheelchair
<i>Head agent (virtual)</i>	k – set the knowledge base of the agents accordingly to the mode d – set the desired trajectory for the agents	k_1, k_2 : set one of the driving modes	m – the current driving mode that set by user t – trajectory from the user

4.2. The driving part

To test driving abilities of the wheelchair model we create an artificial maze environment by Robotics Operation System. The Gazebo 2.0 was used as a simulator [18]. The simulation environment is shown in Figure 9.

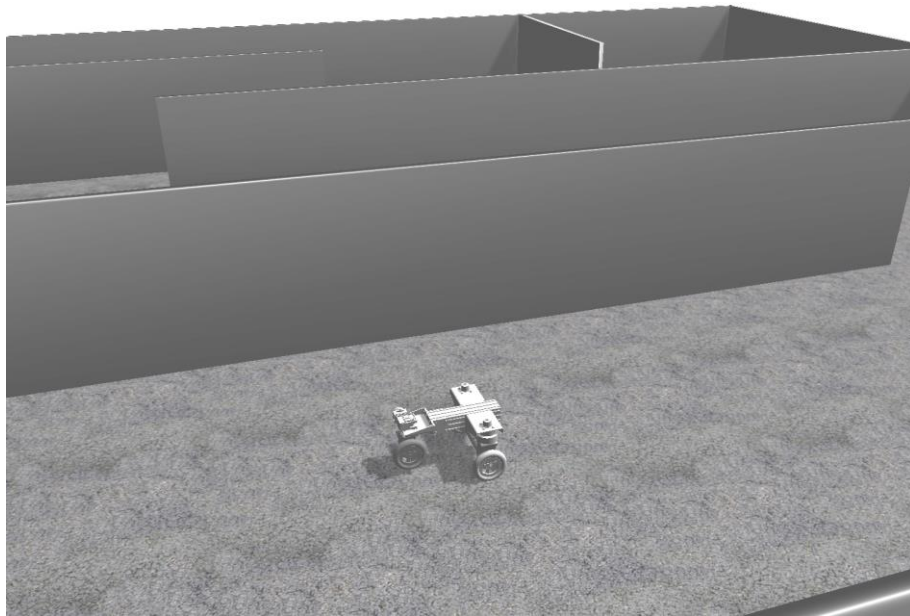


Figure 9. The simulation environment with the base of the wheelchair

For the virtual wheelchair three tasks were defined in order to allow testing. Figure 10 shows the prepared trajectories for the model. The tasks descriptions are:

- a) Driving along the straight line in mode one.
- b) Driving along the curve line in mode one.
- c) Driving along the straight line in mode two,

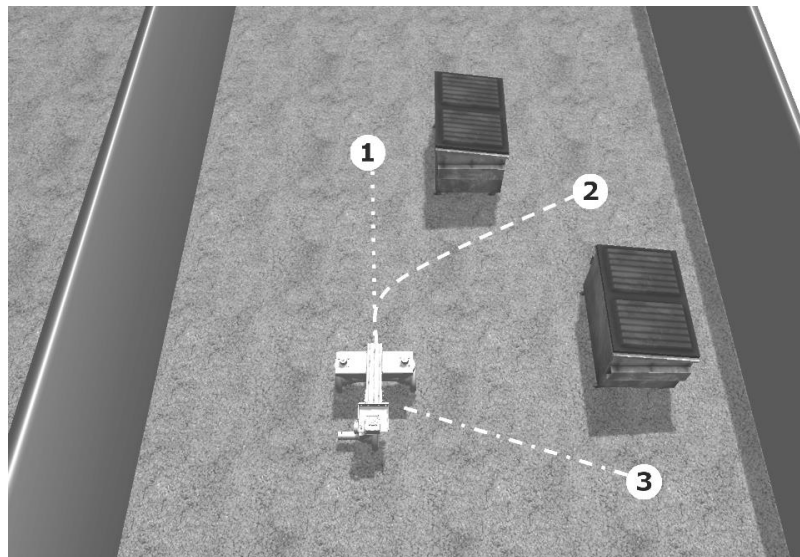


Figure 10. The trajectories of the tasks in driving part of experiments.

The *driving experiments* results are represented in Table 2. This table consists of the median results that were collected from 20 experiments for each of the tasks.

Table 2. The median of the experiments results in driving task

Task	Velocity, <i>seconds</i>	Trajectory error, %	Goal error, %
Task 1	1,43	0,8	0,3
Task 2	2,11	1,54	0,3
Task 3	1,89	1,1	0,3

This results are the proof of the quality of the control system, based on multi-agent reinforcement learning architecture. The advantages of this method are *adaptability* – using reinforcement learning the multi-agent system adapts to control the wheelchair platform through the commands of the head agent; *sharing state* of the agents *decreases the learning time and increases the provided quality* of the already learnt system.

In further experiments it is planned to use the real hardware of the robot to provide full comparison of the two control approaches:

- Asynchronous event-based control software.
- Combination of the asynchronous event-based control software and reinforcement learning multi-agent approach.

5. Summary and Outlook

The control approaches described in this paper were developed for and tested on a holonomic, omnidirectional wheelchair. Such wheelchairs are easier to use for elderly people but also beneficial for semi-autonomous and autonomous operations. The application of an event based control lead to promising results. It is important to note that not a synchronous real-time system was realized, but an asynchronous, event-driven system with buffers and variables expiration times. Therefore a task can be performed faster than assumed, and in case of data unavailability the closest sample (i.e. from buffer) is used to preserve functioning of the system. The use of a multi agent based approach consisted of a learning part with two different modes and a simulation in an artificial maze environment. The simulation was aimed at testing the driving abilities of the wheelchair model. The results of the simulation proofed of the quality of the control system, which is based on multi-agent reinforcement learning architecture. As mentioned above, both approaches can solve the control problem individually but can also complement each other for better system performance. Obviously up to know the results were only verified on one single drive train configuration. More research is needed to underline the findings of the research.

References

- [1] Wada M: Motion Control of a Four-wheel-drive Omnidirectional Wheelchair with High Step Climbing Capability. In: Behnam Miripour (Ed.): Climbing and Walking Robots, 2010.
- [2] M. Wada (2007), Holonomic and Omnidirectional Wheelchairs with synchronized 4WD Mechanism, Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2007), pp. 1196-1202.
- [3] Bin Safar, M. J. A.: Tip-over Stability Analysis and Enhancement for Omnidirectional Personal Mobility Robots with Active Dual-wheel Caster Assemblies. Ph. D. Thesis, Okayama University, 2014.
- [4] Borgolte, U.; Hoyer, H.; Bühler, C.; Heck, H.; Hoelper, R.: Architectural Concepts of a Semi-autonomous Wheelchair. Journal of Intelligent and Robotic Systems 22. Kluwer Academic Publishers, 1998; pp. 233–253.
- [5] Anderson, K. Jones, C.: “Autonomous Robotic Vehicle Design”, Proceedings of the Southwest Regional ASEE Conference, ASEE 2005.
- [6] Ashmore, M., Barnes, N.: “Omni-drive robot motion on curved paths: The fastest path between two points is not a straight line”, Proceedings of the Australian Joint Conference on Artificial Intelligence, Dec., 2002, pp. 225-236.

- [7] Dillard, W.: "Revisting the Autonomous Robot. Finding the Engineer within the Student", Proceedings of the 2004 AEE Conf. , Salt Lake City, 2004.
- [8] Patent „Fahrzeug“ Aktenzeichen DE 10 2005 057 635 A1, Deutsches Patent- und Markenamt, München.
- [9] Patentanmeldung „Fahrzeug“ Aktenzeichen EP 1 954 516 A1 und WO 2007/062858 A1, Europäisches Patentamt, München und World Intellectual Property Organization, Genf, Schweiz.
- [10] Seybold, L., Krokowicz, J., Pieczyński, A., Paczynski, A., Stetter R.: Prinzipien für das Monitoring, die Planung, Regelung und Diagnose von fahrerlosen Transportsystemen. In: Automation 2011. VDI-Berichte 2143. Düsseldorf: VDI, 2011 (S. 137 – 140).
- [11] Stetter, R., Bertsch, S.; Eckart, P.; Paczynski, A.: Torque Steering System for Electrical and Hybrid Power Trains. In: Proceedings of the Fisita EAEC Conference 2011 in Valencia, Spain, 2011.
- [12] Albert M. K. Cheng; Real-Time Systems: Scheduling, Analysis, and Verification, ohn Wiley & Sons, 2003
- [13] Rajib Mall; Real-Time Systems: Theory and Practice. Pearson Education India, 2009
- [14] Dziomin, U., Kabysh, A., Golovko, V., & Stetter, R. (2013, September). A multi-agent reinforcement learning approach for the efficient control of mobile robot. In Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on (Vol. 2, pp. 867-873). IEEE.
- [15] Dziomin, U., Kabysh, A., Golovko, V., & Stetter, R. (2014). A Multi-agent Efficient Control System for a Production Mobile Robot. In Neural Networks and Artificial Intelligence (pp. 171-181). Springer International Publishing.
- [16] Russell, S., & Norvig, P. (1995). Artificial intelligence: a modern approach. Asgd
- [17] Sutton, R.; Barto, A.G.: Reinforcement learning: An introduction. MIT press, 1998.
- [18] Koenig, Nathan, and Andrew Howard.: Design and use paradigms for gazebo, an open-source multi-robot simulator. Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. Vol. 3. IEEE, 2004.