Prediction of the RAF orientation error limits based on the interval analysis method

Y.Chouaibi[1], A.H.Chebbi[1], Z.AFFI[1] , L.Romdhane[2]

[1]Laboratoire de Genie Mécanique, University of Monastir, Tunisia
[2]Department of Mechanical Engineering, American University of Sharjah, United Arab Emirates

***Abstract*—In this paper, a sensitivity analysis, based on the interval analysis method, of the orientation error of the RAF translational parallel manipulator is performed. The orientation error is analyzed by considering, simultaneously, the joints clearances and the geometric parameters uncertainty. The orientation error is mapped within the workspace. The obtained results show that the geometric parameters uncertainty can affect the magnitude and the shape of the orientation error.**

I. INTRODUCTION

One of the main advantages of parallel manipulator are their high accuracy and large dynamic load capacity, compared to their serial counterpart. However, this accuracy can be hindered by the joints clearances and the geometric parameters uncertainty. Chouaibi et al [4] have presented the analytical model and analysis of the orientation error of the RAF translator, which is caused by the clearances in the joints. Tannous et al [2] have presented a sensitivity of the parallel robot accuracy, due to the geometric parameters uncertainty, based on the interval analysis method.

This work deals with the accuracy analysis of a parallel robot based on the interval analysis method by considering, simultaneously, the geometric parameters uncertainty and the clearances in the joints. In section 2 the mathematical tool is presented. In section 3 the structure of the RAF translator is reviewed. The orientation error is modeled and simulated in section 4. Some concluding remarks are given at the end of the paper.

II. THE INTERVAL ANALYSIS METHOD

An interval can be represented by brackets defined by the upper and lower bound of a scalar, a vector or a matrix [1]. A real scalar interval [X] is a nonempty set of real numbers denoted as follows:

$$[X]=[\underline{x},\ \overline{x}] =\{x \in R \mid \underline{x}\leq x \leq \overline{x}\} \qquad (1)$$

where x is the nominal value of the real, $\underline{x}$ is called the infimum and $\overline{x}$ is called the supremum. The set of all intervals over $\mathbb{R}$ is denoted by $\mathbb{R}$.

Intervals of vectors (also called boxes), of dimension n, are given by $[\mathbf{x}]=[\underline{\mathbf{x}},\ \overline{\mathbf{x}}]$ consisting of two real column vectors $\underline{\mathbf{x}}$ and $\overline{\mathbf{x}}$of length n with $\underline{\mathbf{x}}\leq x$. In this case the interval $[x_i]=[\underline{x_i},\ \overline{x_i}]$ is the i-th component of interval vector$[\mathbf{x}]$. The set of all boxes of dimension n is denoted by $\mathbb{IR}^n$.

$$[\mathbf{x}]=[\underline{\mathbf{x}},\ \overline{\mathbf{x}}] =\{x \in \mathbb{R}^n \mid \underline{x}\leq x \leq \overline{x}\} \qquad (2)$$

Similarly, an interval matrix is a matrix with interval components and the space of all m×n matrices is denoted by $\mathbb{IR}^{nxm}$.

$$[\mathbf{M}]=[\underline{\mathbf{M}},\ \overline{\mathbf{M}}] =\{M \in \mathbb{R}^{nxm} \mid \underline{M}\leq M \leq \overline{M}\} \qquad (3)$$

The arithmetic operations on elements with intervals is defined by $(\forall \circ \in \{+, \times, -, \div\})$.

The sum of two elements with intervals is given by:

$$[X]+[Y]=[\underline{x}+\underline{y}\ ,\ \overline{x}+\overline{y}] \qquad (4)$$

The difference of two elements with intervals is:

$$[X]-[Y]=[\underline{x}-\underline{y}\ ,\ \overline{x}-\overline{y}] \qquad (5)$$

The product of two elements with intervals is:

$$[X]\times[Y]=[\min(\underline{xy}, \underline{xy}, \overline{xy}, \overline{xy}), \max(\underline{xy}, \underline{xy}, \underline{xy}, \overline{xy})] \qquad (6)$$

The division of an element by another one, that does not contain any null element, is:

$$[X]/[Y] = [\underline{x},\ \overline{x}].[\min(1/\underline{y},1/\overline{y}), \max(1/\underline{y},1/\overline{y})] \qquad (7)$$

The operations on intervals $[X]\circ[Y]$ allow the determination of the maximum and minimum of the resulting interval [3]. For the arithmetic operations on matrices, we use the same operation on each element.

III. THE RAF MANIPULATOR

The translational parallel manipulator RAF proposed by Romdhane et al., 2002 (Fig.1), is composed of a platform connected to the base by three SPS active legs and two passive kinematic legs (PKLs), which are used to eliminate all possible rotations of the platform with respect to the base.

Let L1 and L2 be the lengths of two legs A_kF_k and F_KG_K for each PKLs (1) ant (2), respectively.

Let ak and gk be the radii of the base and the platform, respectively.

IV. THE ORIENTATION ERROR

The virtual work method is used in order to determine the relationship between the local pose error in each joint due to the clearance and the corresponding orientation error of the platform. The superposition method is used in order to quantify the pose error [4]. Let $\boldsymbol{\delta}_{ij}$be the local pose error caused by the clearance in the j-th pair of the i-th PKL and $\boldsymbol{\tau}_{ij}$ the reaction forces and moments transmitted by this pair. We

note by $\boldsymbol{\Gamma}_{ij}$ the corresponding pose error of the platform. By applying the virtual work principal:

$$\boldsymbol{\tau}_{ext}^{T}.\boldsymbol{\Gamma}_{ij} + \boldsymbol{\tau}_{ij}^{T}.\boldsymbol{\delta}_{ij} = 0 \qquad (8)$$

$$\boldsymbol{\tau}_{ij} = \mathbf{H}_{ij}.\boldsymbol{\tau}_{ext} \qquad (9)$$

Substituting $\boldsymbol{\tau}_{ij}$ by its expression, we get the pose error:

$$\boldsymbol{\Gamma}_{ij} = -\mathbf{H}_{ij}^{T}.\boldsymbol{\delta}_{ij} \qquad (10)$$

If we consider the geometric parameters uncertainty we have.

$$[\boldsymbol{\Gamma}_{ij}] = -[\mathbf{H}_{ij}^{T}].[\boldsymbol{\delta}_{ij}] \qquad (11)$$

$[\mathbf{H}_{ij}]$ is an interval matrix and $[\boldsymbol{\Gamma}_{ij}]$ and $[\boldsymbol{\delta}_{ij}]$ are interval vectors depending on the geometric parameters uncertainty.

For the revolute joints $[\boldsymbol{\delta}_{ij}] = [\mathbf{A}].\varepsilon_d + [\mathbf{B}].\varepsilon_a \qquad (12)$

For the spherical joints $[\boldsymbol{\delta}_{ij}] = [\mathbf{C}].\varepsilon \qquad (13)$

Where, [**A**], [**B**] and [**C**] are interval vectors depending on the geometric parameters uncertainty.

ε_a and ε_d correspond to the axial and radial clearance in the revolute joint, respectively, and ε corresponds to the magnitude of clearance in the spherical joint.

Thus, the platform pose error due to the clearance in all the joints and the geometric parameters uncertainty is an interval vector given by:

$$[\boldsymbol{\Gamma}] = -\sum_{i=1}^{2}\sum_{j=1}^{n_i}[\mathbf{H}_{ij}^{T}].[\boldsymbol{\delta}_{ij}] \qquad (14)$$

Where n_i is the number of joints in the i-th PKL.

The dimensions of the RAF are given in table 1.

The desired workspace, where the pose error is presented, is a cube defined by $-200 < x, y < 200$ and $200 < z < 600$.

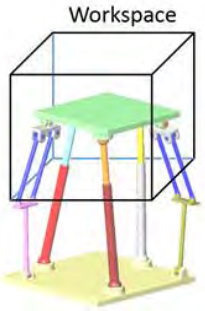


Figure 1. The translational parallel manipulator RAF

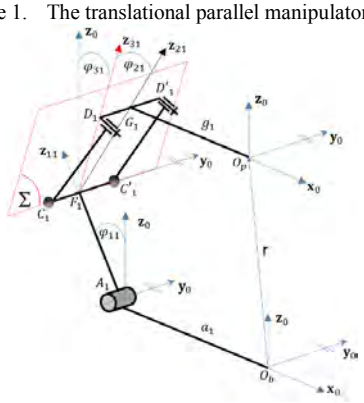


Figure 2. Geometric model of the PKL

TABLE I. PARAMETERS OF THE RAF

$L_{1,2}$ [mm]	$[a_1]_{Sb}$	$[a_2]_{Sb}$	$[g_1]_{Sb}$	$[g_2]_{Sb}$
400	$[-500\ 0\ 0]^T$	$[0\ -500\ 0]^T$	$[-300\ 0\ 0]^T$	$[0\ -300\ 0]^T$

TABLE II. CHARACTERISTICS OF EACH JOINT

Revolute joint	Axial error	ε_a [mm]	0.01
	Radial error	ε_d [mm]	0.01
	Length of the joint	λ [mm]	30
Spherical joints	Radial error	ε [mm]	0.01

Fig. 3 shows an example of the obtained results where the two surfaces represent the upper and lower values of the orientation error θ_X caused, simultaneously, by the joints clearances and the geometric parameters uncertainty. One can note that the geometric parameters uncertainty can affect the magnitude and the shape of the orientation error.

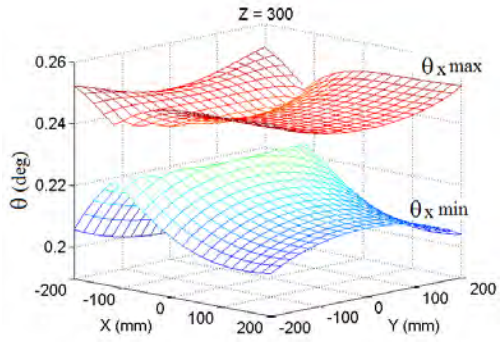


Figure 3. The envelop of the orientation error θ_X

V. CONCLUSION

In this work, we solved for the RAF translational parallel manipulator orientation error, based on the interval analysis method. The limits of the orientation errors are analyzed by considering, simultaneously, the joints clearances and the geometric parameters uncertainty. The obtained results could be useful to the designer to evaluate the accuracy limits of a parallel robot.

REFERENCES

[1] G. Alefelda, G, Mayerb, "Interval analysis: theory and applications", Journal of Computational and Applied Mathematics, vol.121, pp. 421–464, Sep.2000.

[2] M. Tannous, S. Caro, A. Goldsztejn. "Sensitivity analysis of parallel manipulators using an interval linearization method", Mechanism and Machine Theory,vol.71, pp. 93–114.

[3] S. Rump, INTLAB — INTerval LABoratory, in: T. Csendes (Ed.),Developments in Reliable Computing, Kluwer Academic Publishers, Dordrecht, (1999), pp. 77–104.

[4] Y. Chouaibi, A. H. Chebbi, Z. Affi and L. Romdhane. "Analytical modeling and analysis of the clearance induced orientation error of the RAF translational parallel manipulator". Robotica, CJO2014. doi:10.1017/S0263574714002653.

Coordination in Micro Aerial Vehicles for Search Operations

Asif Khan and Bernhard Rinner

***Abstract*— Research in micro-aerial vehicles (MAVs) has enabled the use of swarms of MAVs to search for targets in a given search region to assist search and rescue teams. Management and coordination of these resource limited MAVs is a key challenge for their use in cooperative search and surveillance operations. This research summary presents the major steps in cooperative search and various factors that can affect the performance of cooperative search algorithms.**

I. Introduction

The most important task in the automation of many security, surveillance and search and rescue applications is to search (to find the location of) the targets in a given geographical area (search region) [1], [2]. These search missions are usually time critical and span large search regions. A key research issue in searching the targets is that of coordinated sensor movement i.e., determining where sensors should be dynamically located to maximize the collection of information from the search region [3]. Coordination in sensors is achieved by sharing collected information and control decisions, to find the location of targets and is known as coordinated or cooperative search [2]. The use of camera-equipped Micro Aerial Vehicles (MAVs), as shown in Fig. 1, has recently proved to be a very cost effective and feasible solution [4] due to recent technological advances in design, sensing, embedded processing and wireless communication capabilities of these low cost MAVs.

We present a generic block diagram in Fig. 2 to highlight the major steps in cooperative search using a team of M number of MAVs. Initially, each MAV takes sensor observation (l) and updates its local information about the search region and target existence. Depending on the type of sensor, values of observation and local update can vary from simple binary quantities to complex vector quantities. Similarly, the local update can be implemented by simply over-writing the outdated values or by using sophisticated mechanisms e.g., Bayesian update rule [4]. The local information (u) of each MAV is then shared with other team members. Sharing of information depends on the available communication resources. The shared information is merged by individual MAVs to have updated information for maintaining a common view of the search region [5]. Finally, the decision making part of each MAV uses the updated information (g) to decide its next move (c) in the search region. The movement control hardware then physically moves the MAV to the next assigned position. This four step process is repeated iteratively until the locations of targets are confirmed.

This work was supported in part by the EACEA Agency of the European Commission under EMJD ICE (FPA no 2010- 0012). The work has also been supported by the ERDF, KWF, and BABEG under grant KWF-20214/24272/36084 (SINUS) and has been performed in the research cluster Lakeside Labs GmbH.

The authors are with the Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, 9020 Austria.

Fig. 1: FileFly Hexacopter (left) and Pelican Quadcopter (right) from Ascending Technologies GmbH.

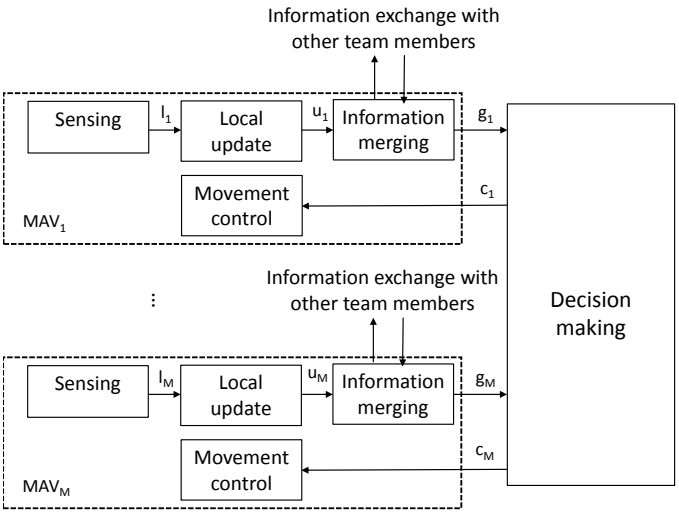


Fig. 2: Main steps in cooperative search using a team of MAVs.

II. Factors affecting cooperative search

Cooperative search using MAVs is a challenging problem. Some of the factors affecting the cooperative search solutions are: MAV resources, sensing, search region, target, communication, coordination, human interaction and heterogeneity. We briefly describe these factors in the following.

A. MAV Resources

The most challenging part of cooperative search is dealing with resource limitations of the MAVs. The MAVs have limited battery life and can fly for less than an hour. The weight that an MAV can carry is another obvious limitation. This limitation restricts the use of very high quality sensors.

Other resource limitations include the number of available MAVs, instability in positioning the MAVs and availability of on-board memory and computation unit. Recourse limitation can be handled by deploying sophisticated coordination algorithms.

B. Sensing

Sensing hardware and software are usually imperfect and are prone to sensing errors. These sensing errors greatly affect the performance of cooperative search approaches. Most approaches introduce observation redundancy to overcome the problem of sensing errors and to achieve a predefined confidence in target location [4], [5]. Type of the sensor (vision, infrared, laser scanner etc.) and shape of field of view (square, rectangle and circle) also affect the performance of cooperative search in certain applications [2]. As the MAVs can move in 3D space, their ability to change their elevation or zoom level of the sensor can also change the resolution of observation. Although, the variable resolution of observation in cooperative search has not been fully explored, it can be of great interest for certain applications [6].

C. Search region

The characteristics of the search region or environment under observation are not always uniform. Searching a mountain region with trees or a city with tall buildings is very different form searching a plane region or sea. The surface of the search region affects the distance of sensor to target, visibility of the target and sensing errors. Similarly, target occlusions due to various obstacles in the search region and boundary shape (regular, irregular, structured) would require completely different approaches of cooperative search.

D. Target

There are many variations on the nature of target. These variations include

1) Number: Prior information about the number of targets (single or multiple) [1] affects the decision of when to stop the search [4]. This information is also required for determining the redundancy in collection of information. Cooperative search approaches are also sensitive to the ratio of number of targets to number of MAVs.
2) Mobility: Targets can be stationary [4], [5] or moving [7] and the mobility model of targets is either known or unknown. The movement of targets makes the search region very dynamic as the location of targets and sensor observations vary with the time. Targets can also appear and disappear for certain duration of the search mission.
3) Cooperation: The target can be cooperative by sending some information to the MAVs (e.g., GPS coordinates) or evade by hiding itself from the MAVs. The targets can also be enemy targets that harm the MAVs in some ways.

A cooperative search procedure that works for one type of target rarely works for another type of target. Thus each type of target generates a new family of algorithms.

E. Communication

The wireless communication resources on-board the MAVs are always limited in range, bandwidth and hop count. This communication limitation is usually solved by allowing communication with the close neighboring MAVs [8], [5], [4]. However, communication only with neighbors always generate delayed, outdated and inconsistent information. This makes the information merging step very challenging, which is usually overcome by information redundancy and time stamping mechanisms [5].

F. Coordination

Efficient coordination algorithms for MAVs introduce team work and intelligent use of limited resources. Coordination among MAVs can take place by sharing only information about the search region or making joint decision about their movement. Coordination among MAVs can be both in terms of sharing a common view of the environment and making joint decisions. This coordination can be centralized performed by a ground control station or decentralized where each MAV decides for itself. Decentralized coordination among MAVs is preferred if MAVs have sufficient communication, computation, and sensing capabilities.

G. Human interaction

If human interaction with the MAVs is allowed [7], the MAV operators take the partial burden of observing, assessing and integrating all the information received from the MAVs. Human interaction can include the human error, fatigue and cost factors into the cooperative search. On other hand, it can increase accuracy of search approaches.

H. Heterogeneity

The MAV platforms, sensor on-board the MAVs, communication, and types of targets can be heterogeneous. The heterogeneity in these factors should be considered while designing or using a cooperative search algorithm.

REFERENCES

[1] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics: A survey," *Auton Robot*, vol. 31, pp. 299–316, 2011.

[2] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, "Recursive Bayesian Search-and-Tracking Using coordinated UAVs for Lost Targets," in *Proc. of IEEE ICRA*, 2006, pp. 2521–2526.

[3] P. Alberto, C. Pablo, P. Victor, and S. A. Juan, "Ground vehicle detection through aerial images taken by a UAV," in *Proc. of IEEE FUSION*, 2014, pp. 1–6.

[4] S. Waharte, N. Trigoni, and S. J. Julier, "Coordinated search with a swarm of UAVs," in *Proc. of IEEE SECON*, 2009, pp. 1–3.

[5] A. Khan, E. Yanmaz, and B. Rinner, "Information merging in multi-UAV cooperative search," in *Proc. of the IEEE ICRA*, 2014, pp. 3122 – 3129.

[6] S. Carpin, D. Burch, N. Basilico, T. H. Chung, and M. Kölsch, "Variable Resolution Search with Quadrotors: Theory and Practice," *J. Field Robotics*, vol. 30, no. 5, pp. 685–701, 2013.

[7] M. L. Cummings, J. P. How, A. Whitten, and O. Toupet, "The Impact of Human Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control," *Proceedings of the IEEE*, vol. 100, pp. 660–671, 2012.

[8] Y. Wang and I. I. Hussein, "Bayesian-based domain search using multiple autonomous vehicles with intermittent information sharing," *Asian Journal of Control*, vol. 16, no. 1, pp. 20–29, 2014.

Object Tracking by Combining Supervised and Adaptive Online Learning through Sensor Fusion of Multiple Stereo Camera Systems

David Spulak[1], Richard Otrebski[1] and Wilfried Kubinger[1]

***Abstract*— A dependable vision system is an integral part of any autonomous system that has to tackle tasks in unstructured outdoor environments. When developing an autonomous convoy of vehicles, where each vehicle has to accurately track and follow the one in front, a stable tracking system is of utmost importance. Combining different object detection methods and integrating various sensor systems can improve the overall reliability of a tracking system. Especially for long-term tracking adaptive methods with re-detection ability are advantageous. In this paper we propose a combination of supervised offline and adaptive online object learning methods through the fusion of thermal infra-red and visible-light stereo camera systems for object tracking. It is shown that by integrating multiple camera systems their individual disadvantages can be compensated for, resulting in a more reliable and stable tracking system.**

I. INTRODUCTION

As part of the project RelCon [1] an autonomous convoy of vehicles is in development. In order to identify and track the vehicle to follow computer vision is deployed. For this purpose thermal infra-red (IR) and visible-light (VL) stereo camera systems are available on the autonomous vehicle. In long-term tracking tasks changing object appearance due to dynamic lighting conditions of the outdoor environment are challenging. This issue can be addressed by deploying adaptive tracking methods. However, especially in long-term tracking some detection capability is advantageous since the tracked object will eventually be lost by the tracker and learning from incorrect examples will start [2], [3]. Another approach is to use imaging sensors that are less sensitive to varying light conditions (e.g. changing shadows, backlight, etc.). Therefore, an IR camera system was used in the RelCon project for detecting the rear-end of the truck to follow [4].

However, in order to exploit the advantages of all stereo camera systems available in the RelCon project we propose a method combining supervised and adaptive online object learning approaches for tracking. This is achieved by fusing sensor information of camera systems with intersecting fields of view (FOVs).

II. THE TRACKING ALGORITHM

By extending the already existing IR tracking algorithm [4] through integrating the available VL camera system, the advantages of both camera systems can be exploited:

- The object appearance in the IR images remains relatively stable, even during large light variations induced by movement or over time. Also during night the VL images are practically rendered useless, while object tracking via heat signature is still viable.
- The FOV of the VL system is bigger than the IR systems (compare with Fig. 2), providing images of higher resolution and wider viewing angles. This makes vehicle tracking in curves more feasible.
- The higher resolution and sharper images of the VL stereo system enable the calculation of less noisy depth information and make object classification via feature detection more practical.

The tracking system we propose in this paper was built using the Robot Operating System (ROS) [5]. This system allows individual programs – called nodes – to communicate via messages. All image data recorded by the autonomous system is published in ROS and made available to all nodes in the system. The architecture of the proposed tracking system is shown in Fig. 1. The whole detection process can be divided in a *supervised* and an *online* part. The supervised offline learning provides a pre-trained classifier to the *IR object tracking* node which detects and tracks the trained object (the rear-end of a truck). After detection the *object data* (location and orientation with respect to the autonomous vehicle) is calculated from the 3D data provided by the stereo came

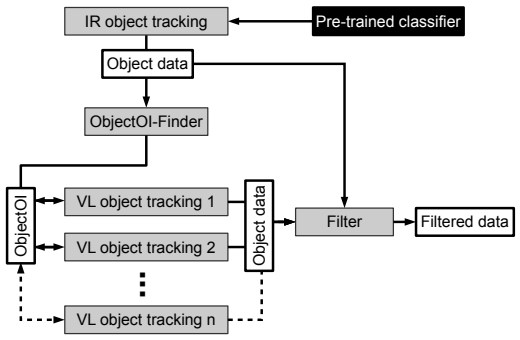


Fig. 1. Object tracking architecture

The *adaptive online* learning is done for the tracking in VL images, where changing object appearance is a bigger issue than in the IR images. The *ObjectOI-Finder* program finds corresponding image segments between the IR and the VL images, according to the *object data* calculated in the *IR object tracking* node (see Fig. 2). The corresponding image segment, which shows the object that needs to be tracked in the VL image, is then published as an object of interest (*ObjectOI*) in ROS and made available to all *VL*

[1] D. Spulak, R. Otrebski and W. Kubinger are with the Department of Advanced Engineering Technologies at the University of Applied Sciences Technikum Wien, Austria [spulak, otrebski, kubinger] @ technikum-wien.at

object tracking nodes. In all n VL camera systems these *ObjectOIs* can then be used for object detection and tracking. The features used in the RelCon project were ORB-features [7], an efficient and much faster alternative to SIFT or SURF.

Fig. 2. Intersecting fields of view of both camera systems

Additionally to the object teaching that is done by the *ObjectOI-Finder* in the intersecting FOVs, each *VL object tracking* node itself can publish new *ObjectOIs*. This enables a continuous update of the objects appearance even outside the IR systems FOV. Finally the calculated *object data* of all stereo camera systems are fused together and denoised in the *Filter* node, using a simple Kalmanfilter [6] that is provided with a motion model.

III. RESULTS

The proposed system was tested with recorded data from one IR and one VL camera system. As it is shown in Fig. 3 the approach taken in this paper makes an object detection outside the IR systems FOV possible – the truck is marked by a white square. The FOV of the object tracking system is effectively doubled, without the need for additional training efforts. By building up a stack of *ObjectOIs* to identify the tracked object, the VL object tracking is also stable against an occasional false detection in the IR images.

Fig. 3. Object tracking outside of the IR systems FOV

Fig. 4 shows the fused object data provided by the two stereo camera systems. A smooth filtering of the object data is achieved by ignoring outliers and considering different measurement inaccuracies for the imaging sensors. As seen in the figure, the vehicle position outside the FOV of the IR system is estimated only with data from the VL system (timesteps 150 – 210, one frame is shown in Fig. 3). Otherwise both VL and IR measurements are used to determine appropriate steering actions of the autonomous system.

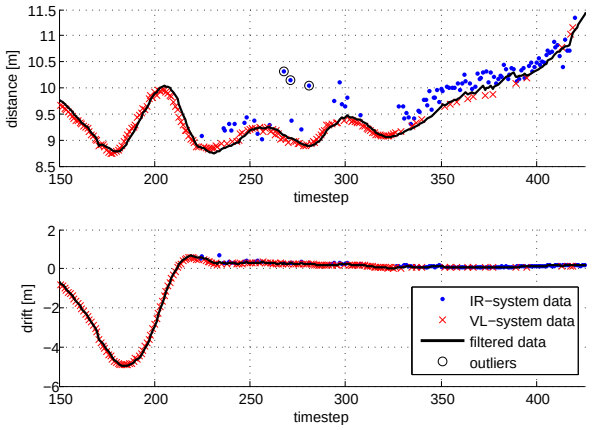


Fig. 4. Filtered data from two different vision systems

IV. CONCLUSION AND FUTURE WORK

In this paper an object tracking algorithm that combines supervised offline and adaptive online learning methods in different imaging sensor systems was presented. This was done by sensor fusion, joining an IR and a VL stereo camera system with intersecting FOVs. Hence, the FOV of the object tracking system was enlarged significantly and more precise object data was able to be obtained, while simultaneously minimizing the necessary offline object training efforts.

Since the test data available in this work was very limited, the tracking system has to be validated more thoroughly in future tests, providing different scenarios and extending the autonomous vehicle with additional VL camera systems.

ACKNOWLEDGMENT

This project has been funded by the Austrian Security Research Programme KIRAS – an initiative of the Austrian Federal Ministry for Transport, Innovation and Technology (bmvit).

REFERENCES

[1] FFG - Österreichische Forschungsfrderungsgesellschaft mbH. (2014) Reliable control of semi-autonomous platforms (relcon). [Online]. Available: http://www.kiras.at/gefoerderte-projekte/detail/projekt/reliable-control-of-semi-autonomous-platforms-relcon/, Accessed on: 5.5.2014

[2] Z. Kalal, J. Matas, and K. Mikolajczyk, "Online learning of robust object detectors during unstable tracking," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, Sept 2009, pp. 1417–1424.

[3] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, Aug 2011.

[4] W. Woeber, D. Szuegyi, W. Kubinger, and L. Mehnen, "A principal component analysis based object detection for thermal infra-red images," in *ELMAR, 2013 55th International Symposium*, Sept 2013, pp. 357–360.

[5] Open Source Robotics Foundation. (2015) Ros.org - powering the world's robots. [Online]. Available: http://www.ros.org/, Accessed on: 19.2.2015

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 2564–2571.

Intelligent Automated Process: a Multi Agent Robotic Emulator

A. Gasparetto[1], R. Vidoni[2] and P. Boscariol[1]

Abstract— **The demands of modern industry push towards new adaptive and configurable production systems where multi-agent techniques and technologies, suitable for modular, decentralized, complex and time varying environments, can be exploited. In this work a generic assembly line is evaluated and the basic features, problems and non-idealities that can occur during the production are taken into account for evaluating and developing an intelligent automated emulated process made of Multi-Agent Robotic Systems. A simplified agentification of the process is made: the main elements of the production are modeled as an agent while the operators that work to restock the local working station stores are considered as autonomous (robotic) agents.**

I. INTRODUCTION

Multi Agent Systems (MAS) have been extensively studied and applied in different fields such as electronic commerce, management and real-time monitoring of networks and air traffic management [1],[2]. The demands of the modern industry aim at creating configurable and adaptive production systems. Indeed, companies need a proper level of agility and effectiveness to satisfy the fast changes of customers' needs. Moreover, markets push manufacturing systems from a mass production to a mass customization fashion; a reduction of the product life-cycles, short lead times and high utilization of resources without increasing the costs are the main targets to satisfy. Thus, industrial needs seem to adapt well to the use of agent technology.

The MAS, in fact, are able to manage complex systems by dividing them into smaller parts and can react to dynamic environments. The main advantages of this technology and approach are: decentralized and distributed decision (i.e. each agent keeps decisions autonomously), and modularity of the structure (i.e. agents are independent). Hence, they are suitable for modular, decentralized, complex, time varying and ill-structured environments. At today, their real application in plants and manufacturing systems is still an exception. This is not because they are not suitable for a real use, but because of a lack of confidence with such systems even if MAS technologies have been already evaluated and theoretically applied in different manufacturing sectors such as production planning, scheduling and control, materials and work flow management [3], [4], [5].

The purpose of this research is to adopt the intelligent agent techniques to study and develop a distributed framework in order to emulate an industrial robotic process and a chain production activity.

[1]DIEGM, University of Udine, Via delle Scienze 206, 33100 Udine gasparetto@uniud.it, paolo.boscariol@uniud.it

[2]Faculty of Science and Technology - Free University of Bozen-Bolzano 39100 Bolzano, Italy renato.vidoni@unibz.it

II. PROCESS AND PRODUCTION LINE

In this attempt, the simulation and emulation of a a mixed-model mass production line are considered. The process is implemented with a pull logic and the supply chain management relies on the Just-In-Time (JIT) techniques. The resulting product is a household appliance, which bill of material has been simplified to facilitate the emulation. Since the product is generic, it can be easily adapted to any production field.

The objective of this research is to create a system able to manage autonomously the lowest level, making the assignment of tasks to the automatic machines. In such a phase, the autonomous agents system must be able to establish, without a centralized control, the better division of the required production, even if there are not the predicted conditions in the line (e.g. if materials in the warehouse are missing).

III. MODEL OF THE EMULATOR

The mixed-model line has been chosen since it allows to change in the mix of produced models without having to perform expensive set-up operations. In such a way, an easy adaptation to fluctuations in demand without excessive stocks of finished goods is admitted. A series of routine operations are usually carried out and, depending on the model, only some product features are modified: in the implemented model each machine has different components but the same production cycle. The appliances in production are brought automatically from one machine to another at the end of each process, and to the store. In Fig. 1 the structure of the emulator, i.e. machines, components warehouse, stores, robots and suppliers is presented.

IV. MULTI AGENT ROBOTIC SYSTEM

The production line model is managed through a system based on multi-agent technology. The following autonomous agents have been implemented:

- *Scheduler agent*. It calculates the size of the lots and schedules the production over time. It sends the lots size to the *Station agents* and instructs the *Robot agent* on what has to be ordered to finish the production.
- *Station agent*. One for each station: it receives the requests to produce the lots from the *Production agent* (only the station number 1) or previous *Station agents* and sends requests to the next stations. Station agents are in charge of controlling the number of available pieces and, if necessary, of ordering the lacking components to the *Robot agent*. For each assembly produced, the related components are decremented.

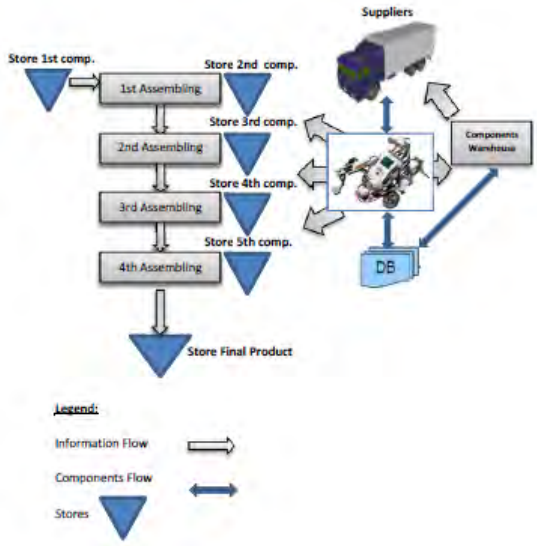


Fig. 1. Simplified model of the process

- *Production agent*. It receives requests and sends them to the line 1^{st} station to start production. It informs the *Scheduler agent* when the lot production is completed.
- *Robot agent*. It receives from the *Station agents* the requests of components delivery and is responsible of guiding a robot to and from the warehouse for the withdrawal of materials. If components are missing, it sends requests to the *Supplier agent*.
- *Supplier agent*. It receives from the *Robot agent* requests for supply components and sends the time needed to fill in the warehouse. When the delivery time is up, it takes care of adding the required parts in the warehouse. If the material is wrong restarts the time and sends a negative response. It chooses the external supplier evaluating its performances.
- *External supplier agent*. It receives requests for parts availability from *Supplier agent* and answers by sending its parameters. If this supplier is chosen, it receives the supply order and sends its time.

V. EMULATOR STRUCTURE

Agent technology has been standardized thanks to the efforts of the Foundation for Intelligent Physical Agents (FIPA). Indeed, it has developed specifications for permitting the spread of shared rules that have brought to the development of FIPA-OS (FIPA-Open Source), JADE (Java Agent Development Environment) and ZEUS agent platforms, all compliant to the FIPA rules and directives [6]. Three main flows characterize the emulator:

- the flow of information exchanged between the various autonomous agents.
- the flow of information of the product within the production line.
- the flow of information of the components between the warehouse and the machines, carried by the robots

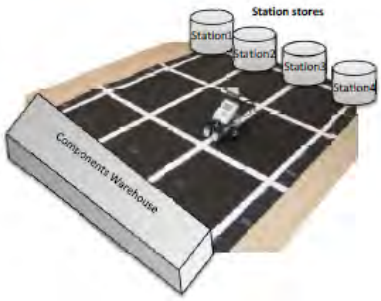


Fig. 2. NXT Robots and environment

The chosen hardware to create a realistic simplified scenario is the Lego Mindstorms NXT, shown in Fig.2. These robots are equipped with two light sensors to follow the chosen road and recognize the crosses, as in an automatic warehouse. Indeed the travel area has been defined as a grid of roads to be followed. Moreover, by means of ultrasonic sensors, they are able to evaluate and recognize obstacles in a suitable range. For each station a predefined path can be used or a shortest path searching algorithm can be exploited.

In a first simplified scenario, four *Station Agents* and one for each of the other agent types have been adopted.

VI. CONCLUSIONS

In this work a generic assembly line has been evaluated and modeled in order to take into account the basic features, problems and non-ideality that can occur during the production. An intelligent automated process made of Multi-Agent Robotic Systems has been evaluated and studied. A simplified agentification of the process has been made, each working station, external supplier and main elements of the production has been modeled as an agent. Operators that work in order to restock the local working station stores have been considered as autonomous (robotic) agents.

The overall framework has been realized and an emulator has been implemented by means of the JADE platform, that follows the FIPA standards, and NXT-robots.

REFERENCES

[1] Wooldridge, M., *An introduction to multiagent Systems*, Ed. John Wiley & Sons, UK., 2002

[2] Bussmann, S., Schild, K., *An Agent-based Approach to the Control of Flexible Production Systems*, Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation, Vol. 2, 481-488, 2001

[3] Pechoucek, M., Mark, V., *Industrial deployment of multi-agent technologies: review and selected case studies*, Autonomous Agents and Multi-Agent Systems, Vol.17/3 , Kluwer Academic Publishers Hingham, MA, USA, 2008

[4] Caridi, M., Sianesi, A., *Multi-agent systems in production planning and control: An application to the scheduling of mixed-model assembly lines*, International Journal of Production Economics, Vol.68, 29-42, 1999

[5] McMullen, P.R., Tarasewich, P., *A beam search heuristic method for mixed-model scheduling with setups*, International Journal of Production Economics, Elsevier, Vol.96/2, 273-283, 2004

[6] Bellifemine, F., Caire, G., Poggi, A., Rimassa, G. *ADE: A software framework for developing multiagent applications*, J Lessons learned. Information & Software Technology, Vol. 50/1-2, 10-21, 2008

Realtime Control of Absolute Cartesian Position of Industrial Robot for Machining of Large Parts*

Diego F. Boesel, Philipp Glocker, José Antonio Dieste, Victor Peinado

I. EXTENDED ABSTRACT

Robot manipulators are not commonly used in machining processes in the industry today. The most significant limiting factor of industrial robot manipulators in machining is their low accuracy. Industrial robots today present high repeatability, but low absolute accuracy [1]. Further, as they have low stiffness [2], they will comply with the forces resulting from machining processes and deviate from the programmed path, intensifying the low accuracy of the robot. These effects will lead to a poor geometrical tolerance and surface quality of parts machined with industrial robots [3].

A number of different techniques to overcome the problem of low absolute accuracy in machining is used today. Some of the most deployed ones include (1) iterative re-teaching of the robot path and (2) robot calibration using an external measurement system [4]. These methods in general are not flexible, among other drawbacks. Recently, visual servoing has been deployed in the academia [5]. They are however not ready for application in the industry (i.e., they work with restrictive technical or commercial conditions) or don't take full advantage of industrial robots.

A realtime 6D absolute robot position control is proposed to overcome the problem of low absolute accuracy of industrial robots. Traditionally, the position control of an industrial robot is realized by the independent position control of its joints [6]. In such an approach, the robot controller cannot take into account the structural flexibility of the kinematic chain or the intrinsic positioning error of the robot structure [4]. Furthermore, taking into account the serial kinematic chain constructive form of the robot, all sources of positioning error will have cumulative effect in the robot's end-effector. An approach to overcome this problem is to control the absolute 6D Cartesian pose of the robot.

Only commercial off-the-shelf hardware has been used to deploy this concept. A laser tracker system (Leica Absolute Tracker AT901 LR from Hexagon Metrology) measures the absolute 3D position and orientation of a tracked object (here, the robot end-effector), with an accuracy better than 30µm (in the set-up used in this project, as illustrated in Figure 1). Using the Ethercat interface of this tracker, these measurements can be obtained with 1kHz sampling rate. The concept was implemented in a Comau NS12-1.85 robot. Its controller C4G Open provides an interface to access the motor control of the robot. Through this interface, it is possible to read planned and current joint trajectory values as well as to partly or completely replace the commands of the proprietary motor control of the robot with 1kHz frequency.

The proposed controller is implemented as an outer control loop of the robot's proprietary motor controller. The robot proprietary controller plans the robot trajectory as well as it implements position, speed and current control for the motors of every joint of the robot. The controller described here controls the absolute 6D Cartesian position of the robot. It actuates by adding contributions to the joint position set-point calculated by the proprietary motor controller of the robot.

The implementation of the outer control loop is divided in three tasks (an illustration can be seen in Figure 2), which are performed each millisecond:

1. Firstly, it analyzes the information received from robot and laser tracker and filters out potential measuring errors from the laser tracker.
2. Then, a position and orientation offset is added to the ideal planned robot pose. This offset instantaneously corrects the robot pose error.
3. Finally, the joint positions for the ideal pose with offset are calculated. They are used in a control

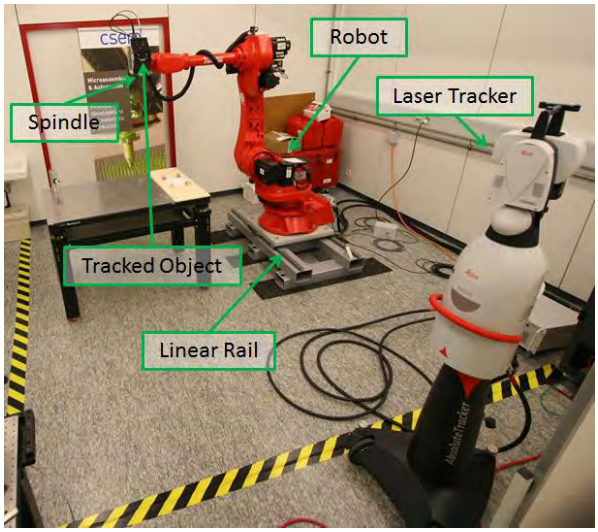


Figure 1: View of the setup of robot cell where the 6D absolute pose control was implemented

* The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement n° 314015 (Megarob). Further support received by MCCS and by the cantons of Central Switzerland.

Diego F. Boesel and Philipp Glocker are with the Robotics & Automation Section of CSEM SA, Alpnach Dorf, OW 6055 CH (corresponding author phone: +41 41 672 75 12; fax: +41 41 672 75 00; e-mail: Diego.Boesel@csem.ch).

José Antonio Dieste and Victor Peinado are with the AITIIP Technological Centre, Zaragoza, 50720 Spain (e-mail: joseantonio.dieste@aitiip.com).

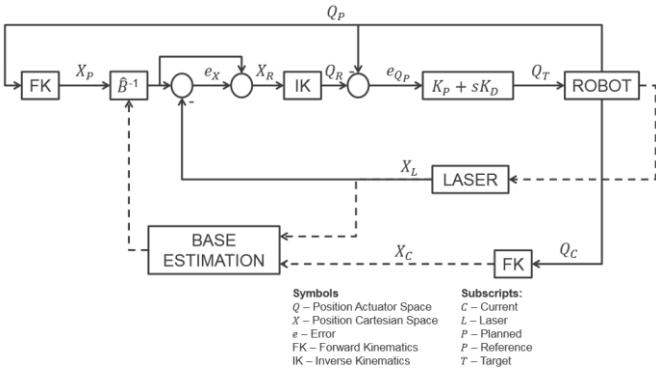


Figure 2: Block diagram of the implemented control algorithm

loop that is fed back by the planned joint positions. The output of this controller is added to the position set-point of the motor control.

The absolute 6D pose controller of the robot can be extended to cope with disturbances in the robot base. Disturbances in this case include motion of the base. In this scenario, an accurate estimation of the instantaneous robot base pose is needed for the absolute Cartesian position control of the robot. The estimation of the robot base is performed by an optimization algorithm at a frequency of 0.5 kHz using singular value decomposition framework (described in [7]).

The performance of the controller proposed in this paper was tested. Specifically, tests in three scenarios were conducted. In free space motion with no base disturbance, the proposed controller improved the static positioning accuracy by more than an order of magnitude: from around 0.5 mm to 0.03 mm (see Figure 3). In dynamic situations, specifically in circular trajectories, the accuracy of the robot with the proposed controller remained constant around 0.1 mm, with some peaks between 0.2 mm and 0.27 mm. These peaks correspond to instants when any of the three first axes of the robot (the larger ones) changed their motion direction. Without correction, the accuracy error varied smoothly between 0.4 mm and 1.0 mm. In motion with machining, the accuracy when the robot was milling showed no significant change. Finally, in motion with base disturbance, when the robot base was randomly dislocated on a linear rail (which can be seen as a disturbance in the robot base), the proposed controller significantly minimized the error of the planned robot tool trajectory.

The implemented controller shows potential for application in the industry as well as for research in the academia. It is the first time that the absolute 6D pose of a robot is controlled with the same frequency as its motor control. One of the future research works in this system is the inclusion of the robot dynamics in the external control loop in order to improve performance when axes change direction of motion.

The implemented solution provides a number of benefits for the industry. The solution is implemented using only commercial components and, therefore, is promptly available for deployment. Since this method does not depend on calibration, it provides flexibility to be used in any type of program and trajectory. Finally, it opens the use of robots for

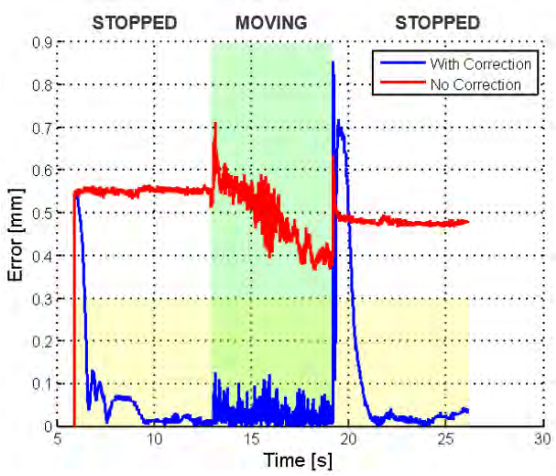


Figure 3: Comparison static accuracy error

high accuracy tasks already in batch 0, fulfilling the philosophy "first time right".

Main target applications are manufacturing of large parts. Such parts are found in the naval, aero spatial and civil construction industries. This controller enables the use of robots in cranes or mobile platforms: the robot will be able to work in parts of tens of meters of length, while ensuring absolute accuracy in the submillimeter range.

II. WORKS CITED

[1] J. Pandremenos , C. Doukas, P. Stavropoulos and G. Chryssolouris, "Machining with robots: a critical review," in *7th International Conference on Digital Enterprise Technology* , Athens, Greece, 2011.

[2] E. Abele, J. Ba, M. Pischan, O. v. Stryk, M. Friedmann and T. Hemker, "Prediction of the tool displacement for robot milling applications using coupled models of an industrial robot and removal simulation," in *CIRP 2nd International Conference Process Machine Interactions*, Vancouver, 2010.

[3] J. Wang, H. Zhang and Z. Pan, "Machining with Flexible Manipulators: Critical Issues and Solutions," in *Industrial Robotics: Programming, Simulation and Applications*, Pro Literatur Verlag, Germany / ARS, Austria , 2006, pp. 515-536.

[4] L. Schwarz, M. Neubauer and H. Gattringer, "A Contribution to Geometric Calibration of Industrial Robots with Laser Pointers," in *Austrian Robotics Workshop*, Linz, 2014.

[5] F. Chaumette and S. Hutchinson, "Visual Servoing and Visual Tracking," in *Springer Handbook of Robotics*, Springer-Verlag Berlin Heidelberg, 2008, pp. 563-583.

[6] J. J. Craig, Introduction to Robotics: Mechanics and Control, Prentice Hall, 2004.

[7] O. Sorkine, "Least-Squares Rigid Motion Using SVD," [Online]. Available: http://igl.ethz.ch/projects/ARAP/svd_rot.pdf. [Accessed 26 Februrary 2015].

Graz Griffins' Solution to the European Robotics Challenges 2014

J. Pestana[1,3], R. Prettenthaler[1], T. Holzmann[1], D. Muschick[2]*, C. Mostegel[1], F. Fraundorfer[1] and H. Bischof[1]

***Abstract*— An important focus of current research in the field of Micro Aerial Vehicles (MAVs) is to increase the safety of their operation in general unstructured environments. An example of a real-world application is visual inspection of industry infrastructure, which can be greatly facilitated by autonomous multicopters. Currently, active research is pursued to improve real-time vision-based localization and navigation algorithms. In this context, the goal of Challenge 3 of the EuRoC 2014[4] Simulation Contest was a fair comparison of algorithms in a realistic setup which also respected the computational restrictions onboard an MAV. The evaluation separated the problem of autonomous navigation into four tasks: visual-inertial localization, visual-inertial mapping, control and state estimation, and trajectory planning. This EuRoC challenge attracted the participation of 21 important European institutions. This paper describes the solution of our team, the Graz Griffins, to all tasks of the challenge and presents the achieved results.**

I. Vision-Based Localization and Mapping

The first track of the simulation contest was split into the tasks of localization and mapping. A robust solution for both tasks is essential for a safe navigation in GPS-denied environments as they form the basis for controlling and trajectory planning respectively.

A. Localization

In this task, the goal was to localize the MAV using stereo images and synchronized IMU data only. The stereo images had a resolution of 752x480 pixels each and were acquired with a baseline of 11 cm and a framerate of 20 Hz. The implemented solution had to run on a low-end CPU (similar to a CPU onboard an MAV) in real-time. The results were evaluated on three datasets with varying difficulty (see Fig. 1) in terms of speed and local accuracy.

We used a sparse, keypoint-based approach which uses a combination of blob and corner detectors for keypoint extraction. First, feature points uniformly distributed over the whole image are selected. Next, quad matching is performed, where feature points of the current and previous stereo pair are matched. Finally, egomotion estimation is done by minimizing the reprojection error using Gauss-Newton optimization. We used *libviso2* [3] for our solution, a highly optimized visual odometry library.

*The first four authors contributed equally to this work.
[1]Institute for Computer Graphics and Vision and [2]Institute of Automation and Control, Graz University of Technology; and [3]CVG, Centro de Automática y Robótica (CAR, CSIC-UPM)
{pestana,holzmann,mostegel}@icg.tugraz.at
{daniel.muschick,rudolf.prettenthaler}@tugraz.at
{fraundorfer,bischof}@icg.tugraz.at
This project has partially been supported by the Austrian Science Fund (FWF) in the project V-MAV (I-1537), and a JAEPre (CSIC) scholarship.
[4]http://www.euroc-project.eu/

Fig. 1. Input data for the localization task. *Left:* Image from the simple dataset. *Right:* Image from the difficult dataset. In comparison to the left image, the right image includes poorly textured parts, reflecting surfaces, over- and underexposed regions and more motion blur.

Fig. 2. Mapping process. *Left:* 3D points and their keyframe camera poses. *Middle:* Constructed mesh. *Right:* Evaluated occupancy grid (color coded by scene height).

B. Mapping

To successfully detect obstacles and circumnavigate them, an accurate reconstruction of the environment is needed. The goal of this task was to generate an occupancy grid of high accuracy in a limited time frame.

For our solution we only process frames from the stereo stream whose pose change to the previously selected keyframe exceeds a given threshold. From these keyframes we collect the sparse features (100 to 120) that are extracted and matched using *libviso2* [3]. For these features we triangulate 3D points and store them in a global point cloud with visibility information. After receiving the last frame, we put all stored data into a multi-view meshing algorithm based on [5]. The generated mesh is then smoothed and converted to an occupancy grid for evaluation. An example mapping process can be seen in Fig. 2.

C. Results

The final evaluation for all participants was performed on a computer with a Dual Core i7 @ 1.73 GHz using three different datasets for each task.

For the localization task, the local accuracy is evaluated by computing the translational error as defined in the KITTI vision benchmark suite [1]. Over all datasets, we reach a mean translational relative error of 2.5 % and a mean runtime of 48 ms per frame.

For the mapping task, the datasets contained a stereo stream and the full 6DoF poses captured by a Vicon system.

With increasing difficulty, the motion of the sensor changed from smooth motion to a jerky up and down movement with a lot of rotational change only. In addition, the illumination changed frequently and the captured elements consisted of fine parts that were challenging to reconstruct (e.g. a ladder). For scoring, the accuracy is calculated using the Matthews correlation coefficient (MCC). Our solution obtains an average MCC score of 0.85 on the final evaluation datasets. An MCC score of 1.0 would indicate a perfect reconstruction.

II. State Estimation, Control and Navigation

The second track aimed at the development of a control framework to enable the MAV to navigate through the environment fast and safely. For this purpose, a simulation environment was provided by the EuRoC organizers where the hexacopter MAV dynamics were simulated in ROS/Gazebo.

The tasks' difficulty increased gradually from simple hovering to collision-free point-to-point navigation in a simulated industry environment (see Fig. 3). The evaluation included the performance under influence of constant wind, wind gusts as well as switching sensors.

A. State Estimation and Control

For state estimation, the available sensor data is a 6DoF pose estimate from an onboard virtual vision system (the data is provided at $10\,\text{Hz}$ and with $100\,\text{ms}$ delay), as well as IMU data (accelerations and angular velocities) at $100\,\text{Hz}$ and with negligible delay, but slowly time-varying bias.

During flight, the position and orientation are tracked using a Kalman-filter–like procedure based on a discretized version of [7]: the IMU sensor data are integrated using Euler discretization (*prediction* step); when an (outdated) pose information arrives, it is merged with an old pose estimate (*correction* step) and all interim IMU data is reapplied to obtain a current estimate. Orientation estimates are merged by turning partly around the relative rotation axis. The corresponding weights are established *a priori* as the steady-state solution of an Extended Kalman Filter simulation.

For control, a quasi-static feedback linearization controller with feedforward control similar to [2] was implemented. First, the vertical dynamics are used to parametrize the thrust; then, the planar dynamics are linearized using the torques as input. With this controller, the dynamics around a given trajectory in space can be stabilized via pole placement using linear state feedback; an additional PI-controller is necessary to compensate for external influences like wind.

The trajectory is calculated online and consists of a point list together with timing information. A quintic spline is fitted to this list to obtain smooth derivatives up to the fourth order, guaranteeing jerk and snap free trajectories.

B. Trajectory planning

Whenever a new goal position is received, a new path is delivered to the controller. In order to allow fast and safe navigation the calculated path should stay away from obstacles, be smooth and incorporate a speed plan.

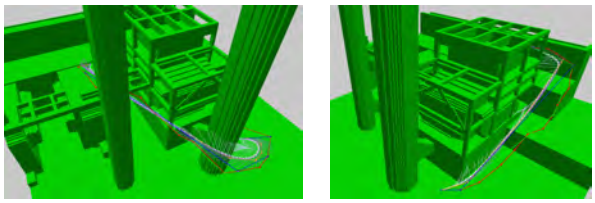


Fig. 3. Industrial environment of size $50\,\text{m}\times50\,\text{m}\times40\,\text{m}$. A typical planned trajectory is shown: The output of the *PRMStar* algorithm (red) is consecutively shortened (green-blue-orange-white).

First, the path that minimizes a cost function is planned. This function penalizes proximity to obstacles, length and unnecessary changes in altitude. Limiting the cost increase, the raw output path from the planning algorithm (shown in red in Fig. 3) is shortened (white). Finally, a speed plan is calculated based on the path curvature.

The map is static and provided as an Octomap [4]. In order to take advantage of the environment's staticity a Probabilistic Roadmap (PRM) based algorithm was selected, the *PRMStar* implementation from the *OMPL* library [8]. The roadmap and an obstacle proximity map are precalculated prior to the mission. For the latter the *dynamicEDT3D* library [6] is used.

C. Results

The developed control framework achieves a position RMS error of $0.055\,\text{m}$ and an angular velocity RMS error of $0.087\,\text{rad/s}$ in stationary hovering. The simulated sensor uncertainties are typical of a multicopter such as the Asctec Firefly. The controlled MAV is able to reject constant and variable wind disturbances in under four seconds.

Paths of $35\,\text{m}$ are planned in $0.75\,\text{s}$ and can be safely executed in $7.55\,\text{s}$ to $8.8\,\text{s}$ with average speeds of $4.2\,\text{m/s}$ and peak speeds of $7.70\,\text{m/s}$.

III. Conclusions

Our solution to EuRoC 2014 Challenge 3 Simulation Contest earned the 6^{th} position out of 21 teams. Although the developed algorithms are a combination of existing techniques, this work demonstrates their applicability to MAVs and their suitability to run on low-end on-board computers.

References

[1] The kitti vision benchmark suite, 2015. `http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php`, 26.03.2015.

[2] O. Fritsch, P. D. Monte, M. Buhl, and B. Lohmann. Quasi-static feedback linearization for the translational dynamics of a quadrotor helicopter. In *American Control Conferenc*, June 2012.

[3] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposion*, 2011.

[4] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[5] P. Labatut, J.-P. Pons, and R. Keriven. Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In *ICCV*, pages 1–8. IEEE, 2007.

[6] B. Lau, C. Sprunk, and W. Burgard. Efficient grid-based spatial representations for robot navigation in dynamic environments. *Elsevier RAS2013*.

[7] R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE TAC*, 2008.

[8] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, December 2012.

Coordination of an Autonomous Fleet

M. Bader[1], A. Richtsfeld[2], M. Suchi[3], G. Todoran[3], W. Kastner[1], M. Vinzce[3]

***Abstract*— Automated Guided Vehicles are systems normally designed to follow predefined paths in order to work reliably and predictably. Giving such vehicles the capability to leave a predetermined path enables the system to cope with obstacles and to use energy efficient trajectories. However, industrial acceptance of such autonomous systems is low, due to the fear of unpredictable behaviour. This paper presents a system design which is capable of spatially adjusting the level of autonomy for control of desired behaviour.**

I. INTRODUCTION

Automated guided vehicles (AGV) are driverless mobile platforms used for transportation processes as well as for flexible system solutions on assembly lines. An AGV is normally designed to operate precisely on predefined tracks, similar to movement along rails. This simplifies on-board self-localization and trajectory control while shifting the burden of control over to the centralised AGV Control System (ACS) server, which controls all of the vehicles in order to prevent deadlocks on the tracks under time constraints. This paper discusses an approach to the distribution of track management that enables vehicles to compute spatially limited paths and trajectories for leaving predefined tracks on-board. The vehicle is able to deal with obstacles, to drive energy-efficiently and to communicate with other vehicles if needed, e.g., at crossings. As a result, the system proposed will be less costly during installation, but also more complex to coordinate as a fleet.

AGVs have been used since the Second World War, first of all as vehicles following rails and then later magnets, coloured bands and other markers integrated into the environment [1]. Nowadays, laser scanners [2] with markers are used to follow a virtual path. On-board self-localization and trajectory planning are still avoided, by just following the handcrafted virtual path assigned by a central ACS which controls the whole vehicle fleet, as shown in Figure 1. Prevention of collisions and deadlocks is imperative, and regular tasks, such as recharging or vehicle cleaning, are managed by the ACS, which generates operation orders based on the input from the Production Planning and Control (PPC). The following section presents our approach to coordination of a fleet of AGVs with an adjustable level of autonomy.

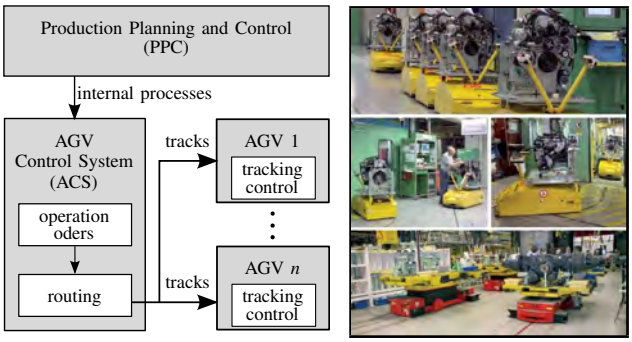


Fig. 1. This figure shows, to the left, the modules involved in a classical system. The PPC analyses general processes (e.g., costumer requests) for the ACS. The ACS identifies operation orders in internal processes and computes routing tables composed of sequences of line, arc and spline segments as tracks. A single AGV has to simply follow the tracks with a tracking control. No path planning is involved. To the right, AGVs in action on an automotive assembly line.

II. APPROACH

Currently deployed AGV systems use tracks that have been manually designed offline. These tracks are defined by a list of segments, for example, lines, arcs, and splines. An AGV's task is to follow these segments and to report on which segment it is currently driving. The segments are distributed by the ACS, as shown in Figure 1, for processing on the AGVs. Currently only two planning levels are needed:

- the overall routing on the centralised server and
- the tracking control on the AGV, which has to follow the designated segments.

We would like to present an approach which additionally enables an AGV system to:

- autonomously avoid obstacles on the track,
- solve situations without the ACS interfering, e.g., multi-robot situations, or pick and place actions,
- use optimised trajectories in order to drive time-, energy- and/or resource-optimally (e.g., in the face of floor abrasion) and
- to be easier to maintain and less expensive during system design and set-up.

Figure 2 depicts this approach. This can only be realised if AGVs are able to:

- localise themselves (even if vehicles are leaving a predefined track),
- communicate with each other and
- execute and adapt behaviour to solve local issues without centralised intervention.

[1]M. Bader and W. Kastner are with the Institute of Computer Aided Automation, Vienna University of Technology, Treitlstr. 1-3/4. Floor/E183-1, 1040 Vienna, Austria [markus.bader, wolfgang.kastner]@tuwien.ac.at

[2]A. Richtsfeld is with DS AUTOMOTION GmbH, Technology & Product Development, Lunzerstrae 60, 4030 Linz / Austria. a.richtsfeld@ds-automotion.com

[3]M. Suchi and G. Todoran are with the Automation and Control Institute, Vienna University of Technology, Gusshausstrasse 27-29 / E376, 1040 Vienna, Austria. [markus.suchi, george.todoran, markus.vincze]@tuwien.ac.at

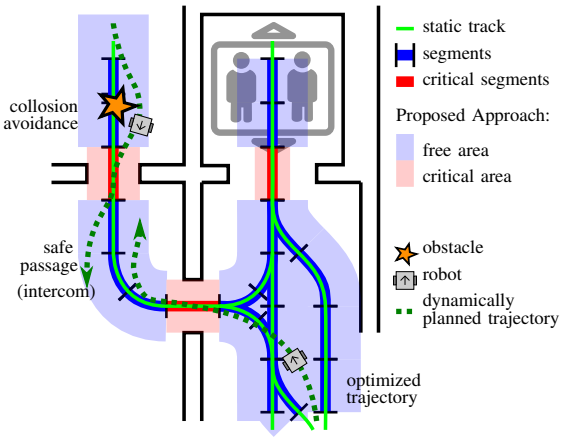


Fig. 2. The system currently used has a centralised path design based on pre-defined line and arc segments (blue). An AGV has to follow the static tracks (green) while the control system takes care of the routing. In contrast to the system currently used, the system proposed here uses pre-defined areas in which a vehicle is allowed to move freely. Obstacles can be circumnavigated and two or more vehicles are able to communicate directly with each other in order to plan trajectories for safely passing one another. Trajectories are locally planned and are time-, resource- or energy-optimised.

In the system proposed, the ACS distributes segments to the AGVs, similar to before, but encapsulates additional attributes. The additional segment attributes are used to signal the system what to expect and suggests a collection of behaviours from which one is selected by the AGV to manage the track segment. Typical behaviours are *stop if there is an obstacle* or *passing on the left is allowed*. In our approach, AGVs are also able to select one of two motion control algorithm to ensure a safe and established system behaviour in regions where no autonomy is allowed, e.g., in narrow passages, in elevators or at a fire door.

- A tracking controller based on a *flat system output* [3] which tries to follow tracks precisely.
- A *Model Predictive Control* (MPC) [4] which uses a local cost map of the environment sensed to deal with obstacles.

Both controls are able to stop in the presence of an obstacle, but the MPC is also designed to react to environmental changes by leaving the track.

The Behaviour Controller (BC) shown in Figure 3 plays a vital part in the new system. This module has to interpret information gained or received locally in order to set parameters for each module and to make binary decisions. Such binary decisions have to be made, for example, if the *scenario detection* module recognizes an obstacle on the track. The BC has to decide if it orders the vehicle to slow down and wait (perhaps the obstacle is a person who will soon leave the track), or trigger the navigation module to steer the vehicle around the obstacle. Such decisions are only possible by the system's integration of expert knowledge which is delivered to the BC from the ACS with segment attributes. This allows the system operator to spatially adjust the level of the vehicle autonomy and to simplify the decision-making process. On the ACS server, we would like to integrate a routing approach that uses Kronecker-Algebra

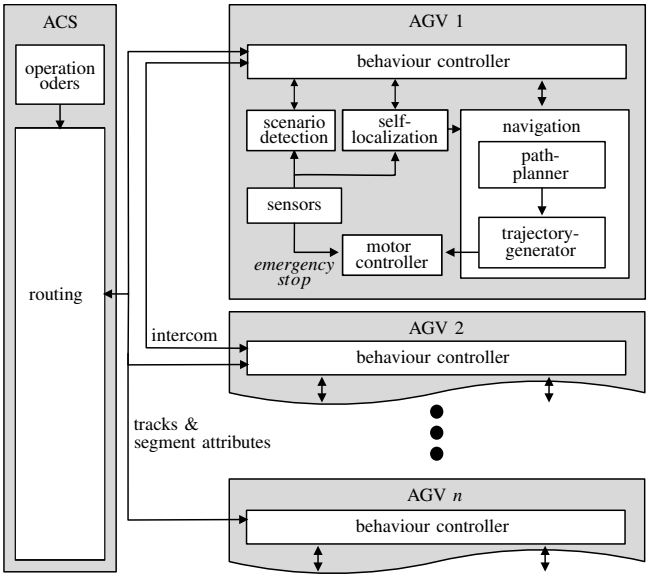


Fig. 3. AGV system overview: The AGV control system (ACS) gets orders from the production planning and control (PPC) (see Figure 1) and distributes them to the AGVs. The ACS also supervises the route planning of AGVs to optimise the execution time of all of the orders given to the system.

[5], which would not only allow computation of routing tables, but would also suggest velocities. Such an approach would reduce overall energy consumption by minimising stops.

III. Conclusion and Results

Many research questions are still up for discussion, such as life long mapping, knowledge representation, optimal multi-robot path planning, usage of smart building systems and the Internet of Things (IoT). We started to implement the proposed system using a simulated small production site in GazeboSim and to test the system using an existing ACS Server with promising results.

Acknowledgment

The research leading to these results has received funding from the Austrian Research Promotion Agency (FFG) according to grant agreement No. 846094 (Integrierte Gesamtplanung mit verteilen Kompetenzen für fahrerlose Transportfahrzeuge).

References

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses," in *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2*, ser. IAAI'07. AAAI Press, 2007, pp. 1752–1759.

[2] G. Ullrich, *Fahrerlose Transportsysteme*, 2nd ed. Wiesbaden: Springer Vieweg, 2014.

[3] M. Liess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[4] T. Howard, M. Pivtoraiko, R. Knepper, and A. Kelly, "Model-Predictive Motion Planning: Several Key Developments for Autonomous Mobile Robots," *Robotics Automation Magazine, IEEE*, vol. 21, no. 1, pp. 64–73, March 2014.

[5] M. V. Andreas Schöbel and J. Blieberger, "Analysis and optimisation of railway systems," in *In EURO-EL 2014, ilina*, 5 2014.

3D Surface Registration on Embedded Systems

Matthias Schoerghuber, Christian Zinner and Martin Humenberger
AIT Austrian Institute of Technology
Donau-City-Strasse 1, 1220-Vienna, Austria
{matthias.schoerghuber, christian.zinner, martin.humenberger}@ait.ac.at

Florian Eibensteiner
Upper Austria University of Applied Sciences
Softwarepark 11, 4232-Hagenberg, Austria
florian.eibensteiner@fh-hagenberg.at

I. Introduction

In this paper we present a fully embedded implementation of a 3D surface registration algorithm for robot navigation and mapping. The target platform is a multi-core digital signal processor, on which we achieved a significant speedup in comparison to the PC-based implementation. The main contribution of this paper is to show the potential of using multi-core DSP platforms for real-time capable implementation of computationally intensive tasks providing surface registration as application example.

Surface registration describes the process of finding a rigid transformation between two sets of 3D points describing the same object, captured from different points of view. This is used to generate a complete 3D scan of an object if the transformation between the single scans are not known. One method for solving this problem is to leverage the assumption that both point clouds perfectly overlap. Minimization of the data term (e.g. distances between matching points) would then lead to the target transformation. Therefore the result heavily depends on the similarity of the datasets, which usually cannot be guaranteed due to (i) noise and outliers in the sensor data, (ii) only small overlap, and (iii) occlusions. Thus a more robust error metric is required with which a rigid transformation can be estimated iteratively.

A well-known solution is using the iterative closest point (ICP) approach by Besl and McKay [1] as shown in Fig. 1. **A** and **B** represent point clouds of the same surface captured from different locations. At every iteration, for each point of **A** the nearest point of **B** is searched. Wrong correspondences can occur as can be seen at location $\mathbf{b}_3$. Using the resulting correspondences, a transformation is estimated and **A** is transformed accordingly. For the next iteration the point clouds are closer to each other and some correspondences, which previously were wrong, become valid. This process is repeated until a defined abortion criterion is satisfied, e.g., the sum of absolute distances (SAD) between correspondences is below a certain threshold. Ideally, the more iterations are performed, the more precise the result is. However, when working with real data containing noise and outliers it is not guaranteed that the algorithm converges to a global minimum.

The described process uses the point-to-point error metric which minimizes the sum of the Euclidean distances between point correspondences. Zhang [2] uses the point-to-point

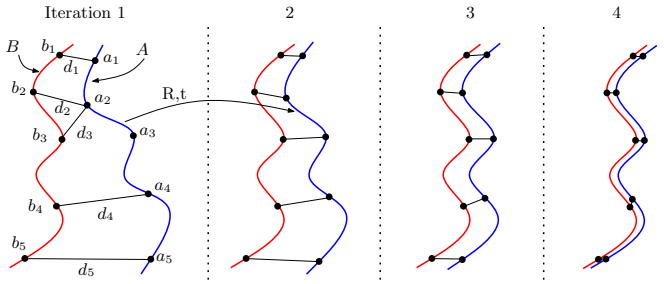


Fig. 1. ICP illustrated with 4 iterations using point-to-point Euclidean distance as error metric.

metric as well, but adds a statistical outlier rejection method. Chen and Medioni [3] introduced the point-to-plane variant where the sum of distances between normal vector to its corresponding point is minimized. Several more variants of the ICP exist in literature [4][5][6]; such as variants with different outlier rejection methods, distance metrics or transform estimation techniques. However, to the authors' best knowledge, no purely embedded real-time implementation of an ICP exists.

II. Registration on Embedded System

In this section, we describe our embedded implementation on the Texas Instruments TMS320C6678 DSP [7] used as target platform. Surface registration on embedded systems, with focus on real-time capability, is a tradeoff between precision and computational effort. Variations based on tree correspondence searches are well suited since they can be realized efficiently with a small memory footprint. For our implementation, we used the point cloud library (PCL)[1] [8] as code base. We first ported this code for execution on DSPs, then applied low-level optimization techniques, and added parallelization with OpenMP, a software framework for multicore shared-memory systems. The C6678 offers 8 VLWI cores clocked at 1.25GHz each, a 2 or 3-level memory architecture, and a Gbit Ethernet interface. For easy integration into robotic platforms, we used the robotic operating system (ROS) [9]. This is a software framework for modular distributed systems where the communication between the processing units is defined. In order to use it for our DSP implementation, we adapted a lightweight

[1] http://www.pointclouds.org

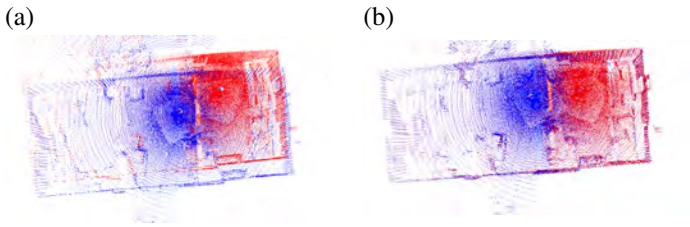


Fig. 2. Two 3D scans overlapped (with compensated initial transformation) (a) source point clouds (b) aligned point clouds; dataset from [11].

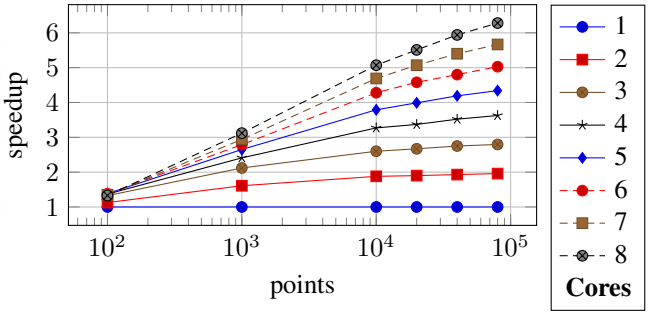


Fig. 3. Speedups of the registration parallelization.

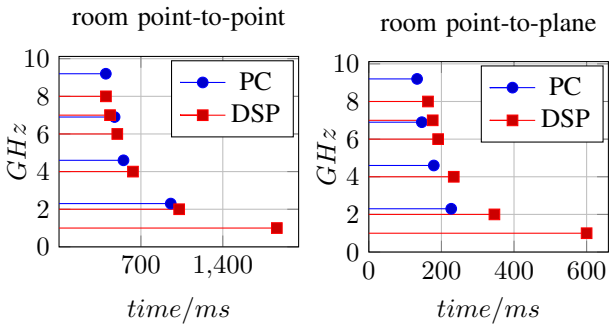


Fig. 4. Comparison of the DSP (8 cores á 1GHz) registration implementation with a PC-system (MS Win7, Intel i7-CPU (4 cores á 2.3Ghz)) on the room data set using point-to-point and point-to-plane ICP variants. The runtime is compared to the sum of used core clocks.

embedded ROS node (urosNode) [10] for fast data exchange over Ethernet.

The complexity analysis of the registration algorithm reveals that the highest complexity lies in the correspondence search ($\mathcal{O}(M \log N)$) and building of the search-tree ($\mathcal{O}(\log N)$). The complexity of the remaining steps of the algorithm is linear. Running the system with a random point cloud (80000 points) and point-to-point metric shows that correspondence estimation takes 51% and the tree build 21% of the time.

Single-Core Throughput Optimization

The correspondence estimation uses the kdtree module of the PCL which uses the FLANN library as back-end. FLANN supports parallel search with OpenMP, but the interface between PCL and FLANN does not make use of this feature. It is especially powerful in automatic parametrization on multidimensional spaces, but these features add complexity and are not needed for ICP. Thus, we use Nanoflann[2] which is optimized for low-dimensional spaces and could better be optimized, due to its reduced code complexity. Applying some further optimizations, such as moving small heap allocations to the faster stack, a speedup between the optimized (single core) and the first ported original registration for the room dataset (Figure 2) of 8.6 is achieved, random clouds with 80,000 and 10,000 points achieved 7.7 and 6.1. Hence, the speedup increases with the number of iterations and the number of points because the DSP can work more efficiently with large amounts of data, stem from the overhead which is independent of the number of points.

Parallelization of the Registration System

The PCL is primarily written for single core execution, except for some sub-libraries which uses OpenMP directives. The modules used for registration do not use any parallel constructs. The focus of adding parallelism lies on the most time consuming tasks which are the nearest neighbour search, i.e. correspondences estimation, and the building of the tree as mentioned earlier. Both tasks suit well for parallel execution as every search is independent of each other and as the branches of a tree do not depend on another. Further parallelism can be added on each loop that transforms the whole point cloud.

[2]http://code.google.com/p/nanoflann/

Figure 3 shows the speedups achieved by parallelization of the optimized single-core version for different point cloud sizes of the overall registration process.

Figure 4 shows the DSP implementation compared to a typical PC-desktop system.

References

[1] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, 1992.

[2] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, pp. 119–152, 1994.

[3] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145 – 155, 1992, range Image Understanding.

[4] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *In International Conference on Pattern Recognition*, 2002, pp. 545–548.

[5] J. Minguez, L. Montesano, and F. Lamiraux, "Metric-based iterative closest point scan matching for sensor displacement estimation," *Robotics, IEEE Transactions on*, vol. 22, pp. 1047–1054, 2006.

[6] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: Science and Systems*, vol. 2, 2009, p. 4.

[7] *TMS320C6678 Data Manual*, Sprs691b ed., Texas Instruments, 2011.

[8] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[9] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," *ICRA Workshop on Open Source Software*, 2009.

[10] (2013, 8) urosnode. [Online]. Available: https://github.com/openrobots-dev/uROSnode

[11] (2013, 8) Normal distribution transform tutorial. [Online]. Available: http://www.pointclouds.org/documentation/tutorials/normal_distributions_transform.php

A Low-cost mini Robot Manipulator for Education purpose

Mohamed Aburaia

***Abstract*— To create a better understanding for Robotic students, real robots are an essential tool. Especially for practicing the theoretical knowledge in this field of technology. On the industrial level, robot manufacturer usually use high quality and expensive components that in turn lead to an increase in the production as well as the robots sales price. This high cost strategy builds an obstacle for the use of real robots particularly for nonprofit organizations.**

This paper introduces a new concept for designing and manufacturing a robot for educational purposes by using modern production techniques and low cost components. Therefore a 4 axis robot was built. The development process from the conception through design, axis production, solving the kinematic, programming the robot and interface to assembly the robot axis were traded in this project.

I. INTRODUCTION

The aspiration of high degreed automation and the need of reproducible concepts, require the use of robots. However, the robot arm market is still limited because robot manufacturer use high quality components that in turn lead to increasing the production as well as the robots sales price [1]. In view of the high price of robots, much research has been done to develop low-cost, high performance robot arms [1] [2] [3].

In addition to the technological development which undertakes the domain robotics, new technologies such as additive manufacturing are gradually maturing leading to an enrichment of their field of application. This technology, which generates parts in a layered way, has a history of more than 25 years. These processes are not any longer used exclusively for prototyping. New opportunities and applications in appropriate manufacturing tasks opened up, even though the commercial impact is still modest [4]. By building three-dimensional parts in a layer-by layer additive manner, the Rapid Techniques allow freeform fabrication of parts of complex geometry directly and automatically from their CAD models, without having to use special fixtures as in the material removal processes. The rapid prototyping technology has helped product developers to improve their products more rapidly at lower costs in the ever changing and more competitive global market [5].

In order to fully exploit these advantages and to combine them with robotics, the idea was to build a mini 4 axis robot, for educational purposes, which would be solid enough to fulfill industrial working tasks such as pick & place using the additive manufacturing technology.

The first step in order to implement this concept, was to build a mini model with low cost components to test this theory out. Therefore an Arduino microcontroller, Servo motors (radio control servo motors) and a 6V vacuum suction as an end effector were used.

II. DESIGN AND CONSTRUCTION

Traditional engineering taught the manufacturability of the product while constructing models. This manufacturability differs according to the used production processes. Manufacturing costs and assembly are other factors that have to be taken into consideration. In addition to these factors, there are defined instructions for a product development process which has to be followed: starting off with devising the idea, up to mass production. The usage of the additive manufacturing process leads to a shortened lap time between the phases of the development process. Some phases may be even eliminated affecting a new redefinition of these factors. These changes caused a complete transformation of the product development process [6].

After drawing some sketches the next step was to start with the 3D design and modelling of links and joints of the robot.

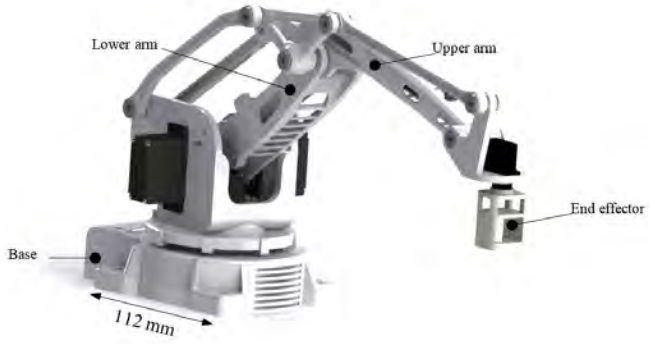


Figure 1: Robot 3D-Model rendered in SolidWorks.

During the construction of the 4 axis robot, the focus was directly placed on the function, not on the way the axis would be manufactured. Furthermore, it has been taken into account to reduce the weight of the robot axis and to adjust the axis and joints on the built engines. Using modern CAD modelling and simulation software, have made it possible to simulate the mechanical behavior of the used axis material during the motion study of the robot. The purpose was to achieve an optimum on the configured design through which we can obtain a maximum of motor torque utilization.

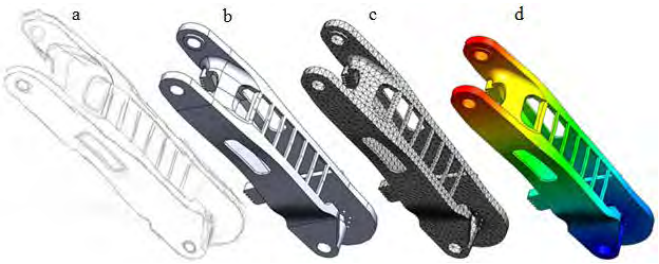


Figure 2: (a) Sketch of a robot axis; (b) 3D-Modell; (c) Meshed-Model (for static study); (d) result of a Static Analysis.

Hence it was examined if the robot axis were over-designed or if the axis would break while performing common operations. By using static and dynamic studies for the stress analysis of parts and assemblies loaded by static and/or dynamic loads, it has been determined how safe, efficient, and economical our design is. Additionally this study presents the answers to the manipulator stability during motion operations.

III. Manufacturing and Assembling of the Axis

The additive manufacturing methods are generally similar to each other in the way that they add and bond materials in layerwise-fashion to form objects [5]. The employed method called Laser Sintering uses laser to sinter powdered material, bonding it together to create a solid structure. The machines consist of a build chamber to be filled with powder with a grain size of up to 50 µm and a laser scanner unit on top that generates the x-y contour. The bottom of the build chamber is designed as a movable piston that can be adjusted at any z-level. The top of the powder bed defines the build area in which the actual layer is built. The laser beam contours each layer. Where the beam touches the surface, the powder particles are locally molten [6].

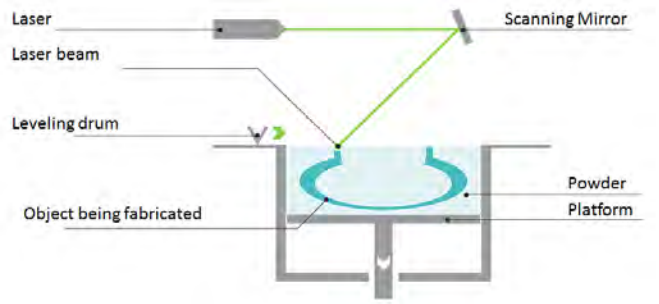


Figure 3: (a) Sketch of a robot axis; (b) 3D-Modell; (c) Meshed-Model (for static study); (d) result of a Static Analysis.

(layer thicknesses) down and an automated roller adds a new layer of material which is sintered to form the next cross section of the object. While non-sintered material is used as supporting material. The robot axis were produced by the plastic laser sintering system EOS FORMIGA P 110. With a build envelope of 200 mm x 250 mm x 330 mm the machine produces plastic products from polyamide or polystyrene within a few hours directly from CAD data without requiring tools [7]. The used powder material is PA 2200 which is a non-filled powder on basis of PA 12 with the following characteristics:

- high strength and stiffness
- good chemical resistance
- excellent long-term constant behaviour
- high selectivity and detail resolution various finishing possibilities (e.g. metallisation, stove enamelling, vibratory grinding, tub colouring, bonding, powder coating, flocking)

To start static and dynamic studies before manufacturing the robot axis, it is required to define the material properties that will be used in the additive manufacturing process.

PA2200 Properties		
Mechanical Properties	***Value***	***Unit***
Tensile Modulus	1700	N/mm²
Tensile strength	45	N/mm²
Elongation at break	20	%
Charpy - Impact strength	53	kJ/m²
Izod – Impact Strength	32,2	kJ/m²
Flexural Modulus	1240	N/mm²

Figure 4: PA2200 Properties [8]

Next step after manufacturing the robot parts, was the assembling of robot joints. The motors and robot axis were assembled with screws. The physical robot is shown in figure 5.

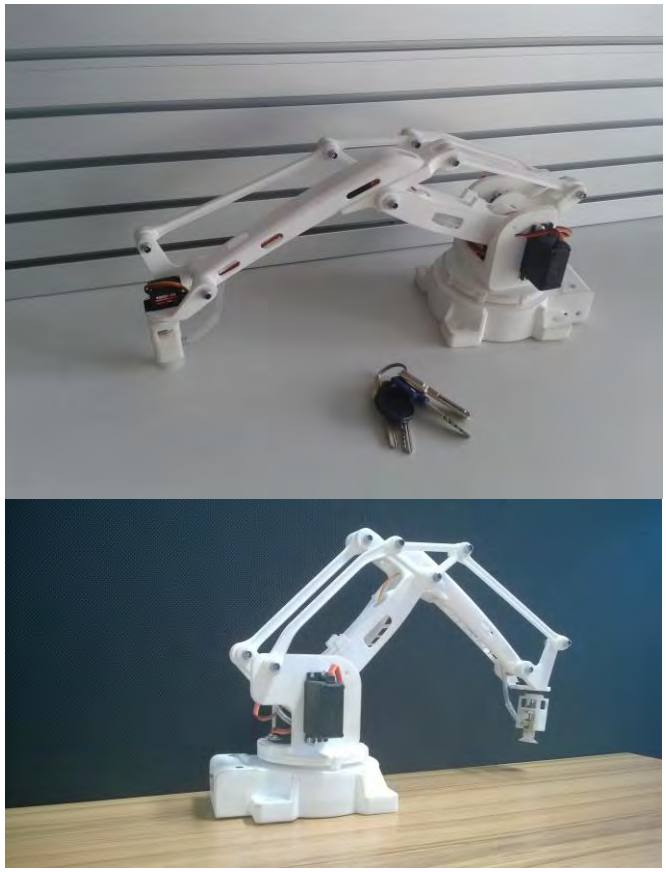
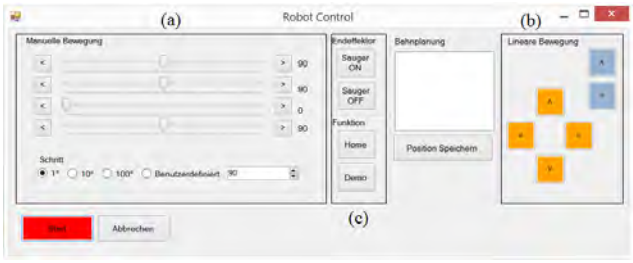


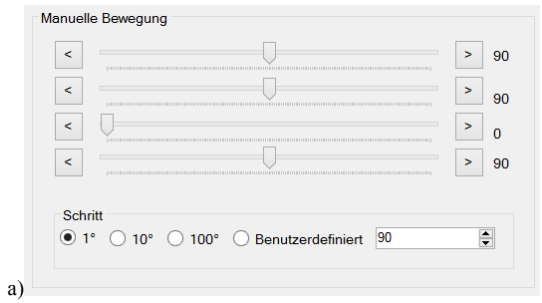
Figure 5: Physical Model.

IV. Robot Control

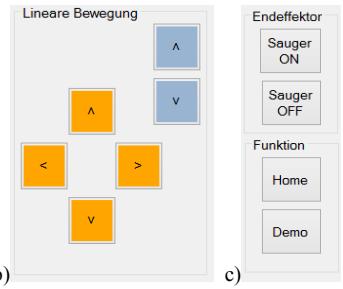
For solving the robot forward and inverse kinematics to achieve a linear motion, the Software tool SimMechanics from MathWorks was used.

After assembling the robot axis and the motors and cabling the system, the programming process began in two stages: Microcontroller programming and building a Human Machine Interface. The Interface was programmed in Visual Studio using C# as programming language. The interface allow the user a signal axis motion (Fig. 6a), a linear motion (Fig. 6b) and to control the End effector status (Fig. 6c).





a)



b)

c)

Figure 6: Human Machine Interface for the Robot.

V. Conclusion

In this research, we have shown a new method to manufacture an industrial robot. Using the additive technology and modern software tools for designing and controlling an industrial robot we were able to shorten the development periods. The first step after creating the main concept was a 3D CAD design of the links and joints of the robot, which would be stiff enough to pick and place industrial applications. Achieving the aim of this work, a Polyamide material was used. The simulation software is endowed to simulate the mechanical behavior of the material during the motion study of the robot and consequently the robot axis were optimized. The second step of this study was to design the controller for the robot-plant design. SimMechanics with the support of Simulink- Libraries from MathWorks were used to realize this task. The third step of this study was to manufacture and assemble a robot. The fourth and last step was developing a user interface for the operator to program robot motions. The developed robot is suited for pick and place applications and for educational purposes. It has been proven that additive manufacturing is an effective production process for producing high quality robot parts in short periods of times. Also this method helped to save valuable engineering time and the design engineer can focus his work more precisely on the product functionality.

References

[1] H.-S. Kim and J.-B. Song, "Low-cost Robot Arm with 3-DOF Counterbalance Mechanism," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, 2013.

[2] Q. Morgan, A. Asbeck and A. Ng, "A Low-cost Compliant 7-DOF Robotic Manipulator," in *IEEE International Conference on Robotics and Automation*, Shanghai, 2011.

[3] J. A. Naves Cocota, H. Silva Fujita and I. Jorge da Silva, "A Low-cost Robot Manipulator for Education," Brasilia, 2012.

[4] G. N. Levy, R. Schindel and J. Kruth, Rapid Manufacturing and Rapid Tooling with Layer Manufacturing (LM) Technologies, State of the Art and Future Perspectives, St. Gallen, 2003.

[5] M. A. Boboulos, CAD-CAM & Rapid prototyping Application Evaluation, Miltiadis A., 2010.

[6] A. Gebhardt, Understanding Additive Manufacturing: Rapid Prototyping - Rapid Tooling - Rapid Manufacturing, München: Carl Hanser, 2011.

[7] EOS, "Plastic laser sintering system FORMIGA P 110 for the direct manufacture of series, spare parts and functional prototypeshttp://www.eos.info/systeme_loesungen/kunststoff/system e_und_zubehoer/formiga_p_110," [Online]. Available: http://www.eos.info/systeme_loesungen/kunststoff/systeme_und_zu behoer/formiga_p_110. [Accessed 25 2 2015].

[8] EOS, "Fine Polyamide PA 2200 for EOSINT P," [Online]. Available: http://pro.sculpteo.com/media/data/faq/pa2200-datasheet.pdf. [Accessed 15 02 2015].

Multi-Hop Aerial 802.11 Network: An Experimental Performance Analysis

Samira Hayat*, Evşen Yanmaz▽
*Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Austria
▽Lakeside Labs GmbH, Klagenfurt, Austria

I. Introduction

Commercial availability of small-scale, open source and inexpensive Unmanned Aerial Vehicles (UAVs) has opened new vistas in many civil application domains, such as environmental monitoring and disaster management. However, most research efforts invested in this field incorporate usage of a single UAV. In time-critical scenarios, such as search and rescue (SAR), the use of multi-UAV systems may significantly reduce the time required for successful mission completion [1]. The victim detection information is required to be delivered by the UAV to the ground station (first responders) in a real-time manner. In large areas, UAVs may go out of communication range of the ground station. To enable real-time transfers, multi-hop communication may be required. Hence, connectivity between the UAVs may become necessary.

Establishment of a multihop aerial network has been the focus of our recent research work. Aerial networks differ from other networks due to some intrinsic characteristics, such as 3D nature and device mobility. Therefore, the communication module required for such a network may also vary from other networks such as VANETs, MANETs and WSNs [2]. To develop UAV-centric communication, implementation of pre-existing, low-cost wireless solutions may help to characterize the aerial network quantitatively.

To this end, 802.15.4-compliant radios have been tested for air-to-air and air-to-ground communication channel characterization for UAV network in [3]. The throughput, connectivity, and range are measured for 802.11b complaint mesh network in [4]. Impact of antenna orientations placed on a fixed wing UAV with 802.11a interface is illustrated in [5]. The UAVNet project [6] offers an 802.11s mesh network formation to address the quality comparison of ground-to-ground links versus air-to-ground links. The performance of 802.11n wireless modules in an aerial network is tested in [7], reporting lower-than-expected throughput.

However, none of these works describes a system that provides high throughput and reliable links. Our recent work [8] proposes a multi-antenna extension to 802.11a to overcome the height and orientation differences faced in aerial networks, providing high UDP throughput over single hop links. We extend the analysis to a multi-hop, multi-sender setup, with focus on providing high-throughput coverage to disconnected or barely connected UAVs via relaying and analyzing fairness in a multi-sender aerial network.

II. Methodology

Our experiments focus on aerial communications with downlink traffic streamed from a UAV to a ground station, either via a direct wireless link (one hop) or via a relaying UAV (two hops). All tested setups are illustrated in Fig. 1.

A. Hardware Setup

Experiments are performed using a ground station laptop and two AscTec Pelican quadrotors, all equipped with Compex WLE300NX 802.11abgn mini-PCIe modules for establishment of 802.11a and 802.11n links. 5.2 GHz channels are used to avoid interference with remote controls (RCs) operating at 2.4 GHz. To achieve omni-directionality, the triangular structure developed in [8] employing three horizontally placed Motorola ML-5299-APA1-01R dipole antennas is used. The UAVs carry an Intel Atom 1.6 GHz CPU and 1 GB RAM. A GPS and inertial measurement unit (IMU) provides tiltion, orientation, and position information. The ground station laptop is raised to a height of 2m for all experiments.

B. Software Setup

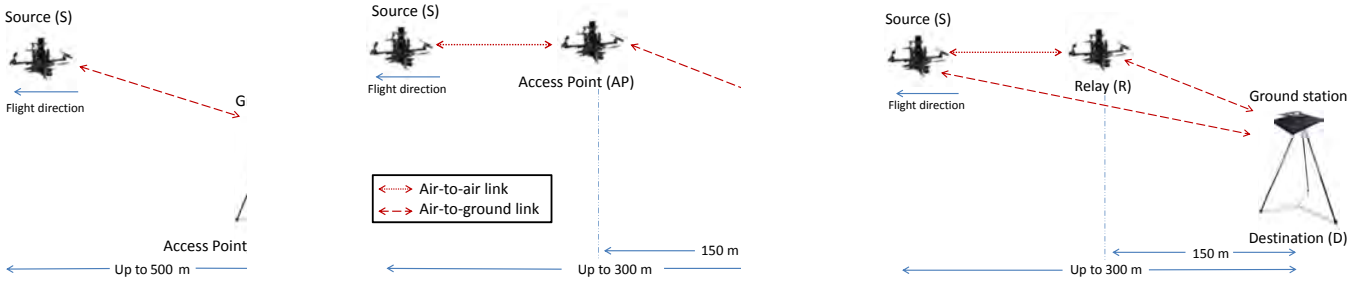
All UAVs and the ground station run Ubuntu Linux kernel 3.2. *ath9k* driver is employed to support the infrastructure, mesh and monitor modes on the 802.11 interface. *ath9k* uses *mac80211* as the medium access layer implementation for packet transmission and reception. Statistics about the transferred packets can be captured using the "iw tool" and the "monitor mode". The configuration utility "iw tool" implemented in the Linux Netlink Interface *nl80211* provides averaged values. To track individual packets, the "monitor mode" offered by Linux wireless is more useful.

We implement the wireless modes (infrastructure and mesh points) as described in Linux wireless. Specifically, *hostapd* is used to manage access point functionalities, and an implementation of 802.11s is used to form a mesh network.

C. Description of Experiments

All experiments are performed in an open field without obstacles. The corresponding pathloss for this line-of-sight scenario can be approximated by a log-distance pathloss model with a pathloss exponent $\alpha \approx 2$ (consistent with free space) [8]. We conduct one-hop and two-hop experiments and analyze performance for infrastructure-based and ad hoc mesh architectures.

For the mesh architecture, each UAV is set as a mesh point. These mesh points communicate with each other over IEEE



(a) AP mode: single (b) AP mode: two hops (c) Mesh mode: all nodes are mesh points.

Fig. 1. Experimental setups: Single and two-hop tests in access point (AP) and mesh modes.

802.11s. The default routing algorithm in the mesh network is the Hybrid Wireless Mesh Protocol (HWMP) [9], which is a variant of ad hoc on demand distance vector routing (AODV).

Fig. 1 shows the three setups analyzed. Fig. 1(a) represents the single hop scenario where we fly one UAV at an altitude of 50 m on a straight line of length 500 m, stopping every 50 m. Packets are transmitted from the UAV to the ground station acting as access point (AP). The two-hop setups are represented in Figs. 1(b) and (c), showing infrastructure-based and mesh architectures, respectively. For both setups, the altitude of the UAVs is maintained at 50 m. One UAV is hovering at a horizontal distance of 150 m from the ground station and another UAV is flying on a horizontal straight line away from the ground station up to a distance of 300 m stopping every 50 m. The hovering UAV acts as a communication bridge, either as an *access point* or a *relaying mesh point* for the infrastructure or mesh architecture, respectively.

Since our goal is to investigate multi-hop networks, we need to shrink the range of communication for our UAVs and ground station. P_{TX} has been lowered to achieve that. Single hop experiments are performed to establish benchmark performance of both 802.11a and 802.11n technologies, as well as evaluating the performance of MIMO enabling 802.11n technology using adaptive rate control. Single-sender multihop tests focus on capturing the benefits and drawbacks of MAC and network layer relaying in an air-ground network. Multi-sender multihop tests help analyze fairness in a multi-sender high-throughput aerial network.

III. Conclusions and Outlook

Multi-UAV systems can facilitate in successful mission completions in many application domains, especially time-critical scenarios like search and rescue. Such a multi-UAV system requires a communication module that can support robust and high-throughput transfers. For establishment of such high-throughput and robust networking, as a first step, there is a need to analyze the pre-existing communication standards.

In this paper, we present our experimental performance analysis of 802.11a and 802.11n technology. From the single hop experiment results, it can be seen that much higher throughput can be experienced using 802.11n employing multiple streams as compared to 802.11a. Single-sender, two-hop experiments have helped establish the advantages and disadvantages of using mesh network as compared to infrastructure network. It can be seen that the default 802.11s routing protocol uses number of hops as a routing metric. In case the single hop link is only intermittently available, it is still prioritized over a two-hop link offering better quality, thus significantly affecting the network performance. Network fairness analysis for the high-throughput 802.11n technology has shown an acceptable degree of fairness in the network, though the network is affected by very high jitter.

Future work aims to focus on development of routing metrics and routing protocols to better cater the needs of aerial network application domains.

Acknowledgements

The work was supported by the ERDF, KWF, and BABEG under grant KWF-20214/24272/36084 (SINUS). It has been performed in the research cluster Lakeside Labs. The authors would like to thank Vera Mersheeva, Juergen Scherer and Saeed Yahyanejad for acting as pilot.

References

[1] L. Reynaud, T. Rasheed, and S. Kandeepan, "An integrated aerial telecommunications network that supports emergency traffic," in *Wireless Personal Multimedia Communications (WPMC), 2011 14th International Symposium on*, Oct 2011, pp. 1–5.

[2] I. Bekmezci, O. K. Sahingoz, and c. Temel, "Survey flying ad-hoc networks (fanets): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, May 2013. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2012.12.004

[3] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni, "Sensorflock: an airborne wireless sensor network of micro-air vehicles," in *Proc. ACM Intl. Conf. Embedded Networked Sensor Systems (SenSys)*, Nov. 2007.

[4] T. X Brown, B. Argrow, C. Dixon, S. Doshi, R. G. Thekkekunnel, and D. Henkel, "Ad hoc UAV ground network (Augnet)," in *Proc. AIAA Unmanned Unlimited Tech. Conf.*, Sept. 2004.

[5] C.-M. Cheng, P.-H. Hsiao, H. Kung, and D. Vlah, "Performance measurement of 802.11a wireless links from UAV to ground nodes with various antenna orientations," in *Proc. Intl. Conf. Computer Comm. and Networks*, Oct. 2006, pp. 303–308.

[6] S. Morgenthaler, T. Braun, Z. Zhao, T. Staub, and M. Anwander, "UAVNet: A mobile wireless mesh network using unmanned aerial vehicles," in *Proc. IEEE GLOBECOM Wksp-WiUAV*, 2012, pp. 1603–1608.

[7] M. Asadpour, D. Giustiniano, K. A. Hummel, and S. Heimlicher, "Characterizing 802.11n aerial communication," in *Proc. ACM MobiHoc Wksp on Airborne Networks and Communications*, July 2013, pp. 7–12.

[8] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility," in *Proc. IEEE Intern. Conf. Comp. Commun. (INFOCOM), Mini conference*, Apr. 2013.

[9] *IEEE Std 802.11s - Amendment 10: Mesh Networking*, Sept. 2011.

Evaluation of Different Importance Functions for a 4D Probabilistic Crop Row Parameter Estimation*

Georg Halmetschlager Johann Prankl Markus Vincze

***Abstract*— The autonomous navigation of field robots requires to detect crop rows. To achieve this goal we developed a probabilistic detection approach. We evaluate five different importance functions for a 4D particle filter parameter estimation that utilizes a comprehensive geometric row model and images that are segmented based on near-infrared and depth data. The functions are evaluated with real-life datasets recorded during in-field tests with the agricultural robot FRANC. The results show that two importance functions lead to outstanding detection results without considering any a-priori information, even for sparse crop densities. Two importance functions are implemented in a particle filter crop row detection algorithm and tested with different real-life field images. First results indicate that the importance functions in combination with the 4D particle filter enable an offline crop detection with an accuracy of a few centimeters.**

I. Introduction

Most agricultural robots and guided tractors use RTK-GPS systems or laser sensors to solve the task of autonomous navigation [1]. Vision systems promise to offer outstanding advantages compared to pure GPS solutions, provide higher dimensional information, and are inexpensive compared to laser range finders [2]. Hence, we propose a pure machine vision system to solve the task of navigation in row-organized fields. Further, the crop rows have to be detected relative to the robot for the determination of the negotiable track.

Most of the developed vision based detection algorithms consist of a segmentation step and a subsequent state-of-the-art line detection algorithm such as the Hough transformation [3], [4], [5], [6], [7].

We introduced in [8] a near-infrared and depth (NIRD) data based segmentation that enables a height-bias-free detection of the rows within the ground plane. The online row detection is realized with a 3D cascaded particle filter. Each hypothesis in the 3D state space describes a parallel line pattern within a 2D plane and consists of the orientation θ, the offset r, and the distance between the lines d. The algorithm offers high detection rates for image sequences and elongated row structures.

We aim to advance this approach by adding the row width w as fourth dimension to the state space of the particle filter. The most important step towards the realization and implementation of the particle filter parameter estimator is the selection of a suitable importance function (cf. [9]). Hence, we evaluate five different functions that promise to offer maxima for correct parameter estimations.

*This work was funded by Sparkling Science a programme of the Federal Ministry of Science and Research of Austria (SPA 04/84 - FRANC).

All authors are with Faculty of Electrical Engineering, Automation and Control Institute, Vision for Robotics Laboratory, Vienna University of Technology, A-1040 Vienna, Austria. {gh, jp, mv}@acin.tuwien.ac.at

II. Concept

A single hypothesis in the 4D state space represents a pattern of parallel stripes. Starting from the NIRD segmented image (cf. Fig. 1) and the 4D estimation, we construct five importance function that exploit the white pixel count (WPC) of four $[m \times n]$ binary images:

- NIRD segmented image, $\mathbf{I}_s$ (cf. Fig. 1)
- stripe image, $\mathbf{I}_h$ (cf. Fig. 2)
- intersection image, $\mathbf{I}_i$ (cf. Fig. 3)
- union image, $\mathbf{I}_u$ (cf. Fig. 4)

with the corresponding WPCs c_s, c_h, c_i, and c_u.

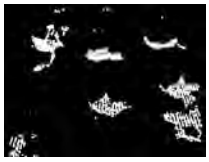


Fig. 1: NIRD, $\mathbf{I}_s$.

Fig. 2: Row image, $\mathbf{I}_h$.

Fig. 3: Intersection, $\mathbf{I}_i$.

Fig. 4: Union, $\mathbf{I}_u$.

The pixel counts c_i and c_s can be directly extracted from the images $\mathbf{I}_s$ and $\mathbf{I}_h$ with

$$c_i = \sum_{x,y} i_s(x,y) \wedge i_h(x,y) \tag{1}$$

$$c_u = \sum_{x,y} i_s(x,y) \vee i_h(x,y), \tag{2}$$

where $i_s(x,y)$ is a pixel of $\mathbf{I}_s$ and $i_h(x,y)$ a pixel of $\mathbf{I}_h$.

The four WPCs are combined to five importance functions w_n (3)-(7). The importance function

$$w_1 = \frac{c_i}{c_s} \tag{3}$$

calculates the coverage rate. $w_1 = 1$ if all pixels of the segmented image are covered by the generated row pattern. The second importance function

$$w_2 = \frac{c_i}{c_h} \tag{4}$$

rewards the hypotheses if less row pixels c_h are needed to result in an amount of intersection pixels c_i. $w_2 = 1$, if all row pixels have an corresponding pixel within the segmented image. w_3 (5) is a combination of (3) and (4).

$$w_3 = \frac{c_i}{c_s} \cdot \frac{c_i}{c_h} \tag{5}$$

w_4 (6) combines the union of $\mathbf{I}_s$ and $\mathbf{I}_h$ with their intersection.

$$w_4 = \frac{c_i}{c_u} \tag{6}$$

The last importance function

$$w_5 = \frac{c_i}{c_s} \cdot \frac{c_i}{c_u} \tag{7}$$

represents a combination of (3) and (6).

III. Experiments and Results

The different importance functions were evaluated with ten images of crop rows that offer different crop densities. They were extracted from a real-life dataset that was recorded with the stereo and NIR camera systems of the robot FRANC[1].

The 4D state space is sampled with 10000 hypotheses which are analyzed for an evaluation of the proposed importance functions. Figure 5-8 show the overlay of the segmented image and the hypothesis ($= \mathbf{I}_u$) that achieves the highest importance value for the given functions. The functions w_3 and w_5 result in identical hypotheses that fit to the crop rows in the image.

Fig. 5: $w_1 = 1$.

Fig. 6: $w_2 = 0.260$.

Fig. 7: $w_3 = 0.120$, $w_5 = 0.117$.

Fig. 8: $w_4 = 0.161$.

w_3 and w_5 are selected as the importance function for a subsequent test with a particle filter filled with 1000 particles.

The particle filter results with the importance function w_3 (w_5) in average after 4.6 (4.8) iterations in a stable estimation and low absolute errors for all four parameters. Table I shows the average absolute errors between the estimated parameters and the manually measured ground truth data for five different images.

[1] http://franc.acin.tuwien.ac.at

TABLE I: Average absolute error between ground truth and the crop row parameter estimation.

	e_θ	e_r	e_d	e_w
w_3	< 0.02rad	< 0.03m	0.020m	0.03m
w_5	0.035rad	< 0.03m	0.022m	0.02m

IV. Discussion

All five importance functions end up with plausible results. Function w_1 rewards the estimation that covers all pixels of the segmented image without taking the row area into consideration. Hypotheses that offer a big row area ($w \approx d$) automatically result in high coverage rates.

Function w_2 rewards hypotheses that have a high coverage referred to the area of the stripe pattern. Since the crops in Fig. 1 are separated from each other, w_2 results for each stripe pattern in a rating < 1. The hypothesis with the highest rating represents a single stripe that lies within the row with the highest relative crop density.

w_3 merges the first two importance functions. w_2 punishes the hypotheses if the stripe area is unnecessary big, while w_1 rewards the hypotheses if it offers a high coverage.

Function w_4 results in moderate estimation errors for the offset and orientation of the row pattern, but fails to determine the correct row distance.

w_5 is a combination of w_1 and w_4. w_1 rewards the hypotheses if more pixels can be covered, while w_4 punishes oversized row areas.

The evaluation of the different importance functions shows that w_3 and w_5 are suitable for a 4D particle-filter-based crop row detection. Both importance functions offer in combination with a 4D particle filter outstanding detection results, indicate that they are suitable for an offline ground truth estimation, and result in more accurate crop row estimations than the 3D particle filter approach.

References

[1] D. Slaughter, D. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Computers and Electronics in Agriculture*, vol. 61, no. 1, pp. 63 – 78, 2008.

[2] U. Weiss and P. Biber, "Plant detection and mapping for agricultural robots using a 3d lidar sensor," *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 265 – 273, 2011, special Issue ECMR 2009.

[3] B. Åstrand and A.-J. Baerveldt, "A vision based row-following system for agricultural field machinery," *Mechatronics*, vol. 15, no. 2, pp. 251 – 269, 2005.

[4] G.-Q. Jiang, C.-J. Zhao, and Y.-S. Si, "A machine vision based crop rows detection for agricultural robots," in *Wavelet Analysis and Pattern Recognition (ICWAPR), 2010 International Conference on*, 2010, pp. 114–118.

[5] G. R. Ruiz, L. M. N. Gracia, J. G. Gil, and M. C. S. Junior, "Sequence correlation for row tracking and weed detection using computer vision," in *Workshop on Image Analysis in Agriculture, 26-27.*, 2010.

[6] J. Romeo, G. Pajares, M. Montalvo, J. Guerrero, M. Guijarro, and A. Ribeiro, "Crop row detection in maize fields inspired on the human visual perception," *The Scientific World Journal*, vol. 2012, 2012.

[7] V. Fontaine and T. Crowe, "Development of line-detection algorithms for local positioning in densely seeded crops," *Canadian biosystems engineering*, vol. 48, p. 7, 2006.

[8] G. Halmetschlager, J. Prankl, and M. Vincze, "Probabilistic near infrared and depth based crop line identification," in *Workshop Proceedings of IAS-13 Conference on*, 2014, pp. 474–482.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

Adaptive Video Streaming for UAV Networks*

Severin Kacianka[1] and Hermann Hellwagner[1]

Abstract— **The core problem for any adaptive video streaming solution, particularly over wireless networks, is the detection (or even prediction) of congestion. IEEE 802.11 is especially vulnerable to fast movement and change of antenna orientation. When used in UAV networks (Unmanned Aerial Vehicles), the network throughput can vary widely and is almost impossible to predict. This paper evaluates an approach originally developed by Kofler for home networks in a single-hop UAV wireless network setting: the delay between the sending of an IEEE 802.11 packet and the reception of its corresponding acknowledgment is used as an early indicator of the link quality and as a trigger to adapt (reduce or increase) the video stream's bitrate. Our real-world flight-tests indicate that this approach avoids congestion and can frequently avoid the complete loss of video pictures which happens without adaptation.**

I. System Design

A. Overview

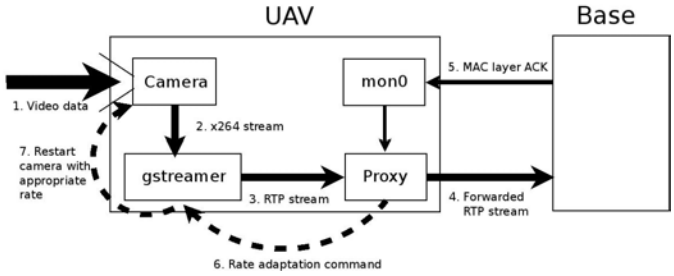


Fig. 1. Overview of the system architecture

Figure 1 shows an overview of the system architecture. The camera delivers an H.264 video stream and is controlled by `gstreamer`, an open-source multimedia framework. `gstreamer` can request (almost) any bit rate from the camera and passes the H.264 encoded video stream on to the proxy as a normal RTP stream (Real-Time Transport Protocol). `gstreamer` (or more exactly, the shell script controlling it) offers the proxy an interface to request a rate change. In one thread, the proxy forwards the RTP stream unaltered to the base station (that is running the video streaming client software), and keeps track of when each RTP packet is sent. At the same time, another thread listens on the wireless monitoring interface of the Linux kernel for the MAC layer ACK (Media Access Control layer acknowledgment) of the sent packet. Depending on the delay between the sending of an RTP packet and its acknowledgment, the proxy will adapt the stream.

*The work was supported by the ERDF, KWF, and BABEG under grant KWF-20214/24272/36084 (SINUS). It has been performed in the research cluster Lakeside Labs.

[1]Severin Kacianka and Hermann Hellwagner are with the Institute of Information Technology (ITEC), Alpen-Adria-Universität Klagenfurt, 9020 Klagenfurt, Austria {severin.kacianka,hermann.hellwagner}@aau.at

B. Adaptation Proxy

The *Adaptation Proxy* forwards the RTP stream sent from `gstreamer`, measures the delay between the sent packets and the reception of their ACKs and makes adaptation decisions.

From Kofler's [3] work the following three assumptions were drawn:

1) A frame only shows up on the monitoring interface after it was transmitted and the ACK received.
2) The lower the delay, the better the connection.
3) The delay is mainly affected by the connection quality (no random changes).

By means of extensive testing, the following observations to confirm these assumptions were made:

1) When the delay is high (above ∼1.0 sec), there will be strong artifacts in the video stream.
2) When the delay climbs above ∼0.3 sec, it usually continues to climb further.
3) When the connection is good (for example the UAV is next to the laptop), the delay does not increase and the video stream stays stable.

The proxy consists of two threads (Figure 2): The thread `proxy` just accepts RTP packets from `gstreamer` and forwards them to the base station while noting the time of sending. The `sniffer` thread listens on the Linux kernel's monitoring interface (in our case always called `mon0`) and whenever it receives an RTP packet it will log the time (which is the time of the reception of the MAC layer acknowledgment) and compare it to the send time. It matches the packets by their RTP sequence number.

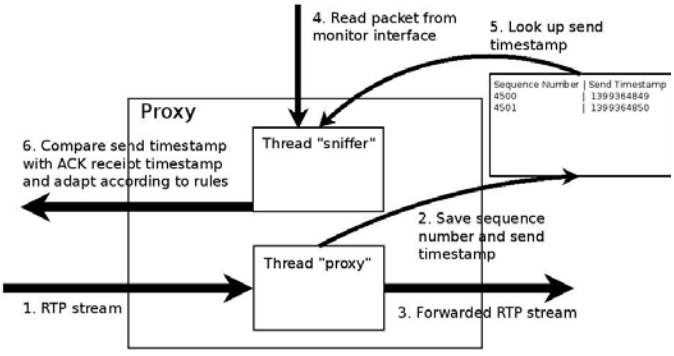


Fig. 2. Overview of the proxy architecture

The decision when to adapt is directly influenced by the three observations mentioned above. The algorithm tries to

avoid delays higher than 1.0 seconds by reducing the video stream's bit rate as soon as the delay crosses the threshold of 0.3 seconds. As it usually takes some time for the delay to react to the lower bitrate, i.e., to decrease, the algorithm will wait for a period of 25 consecutive packets ("cooldown period") before reducing the quality further. Without this cooldown period, the proxy would reduce the quality down to the lowest level most of the time. When there is an extended period (250 packets) of packets with a low delay (<0.1sec), the proxy will increase the bit rate by one level.

II. Experiments and Results

Around 50 evaluation flights were conducted, however only 12 used the same software and are thus included in the results. More details about the experiments and the results can be found in the full paper [2] and the master's thesis [1].

A. Characteristics of the Delay

Figure 3 and Figure 4 depict two examples of the delay's behavior. Figure 3 shows how quickly the delay can rise from low levels to high peaks. Without a mechanism to counter the effect, the delay will rise until the UAV stops moving and/or the antenna orientation improves.

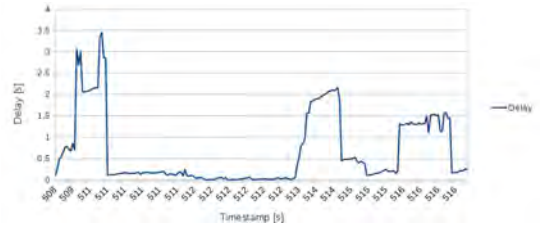


Fig. 3. Development of delay *without* adaptation

When active bit rate adaptation takes place, it can counter the rise of the delay and keep the periods of high delay shorter. Figure 4 shows how the quality level is reduced as the delay crosses the threshold of 0.3 seconds. It does not fall immediately, because the cooldown period keeps the bit rate high. As soon as quality level 3 is reached, the delay immediately falls.

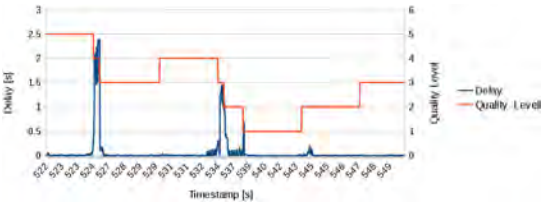


Fig. 4. Development of delay *with* adaptation

B. Delay

Figure 5 shows the mean delay. It is interesting to see that the flights with adaptation had a significantly lower mean delay than the flights without adaptation. While the data set is small, these results confirm Kofler's findings and suggest that the delay is indeed a very good indicator of the link quality.

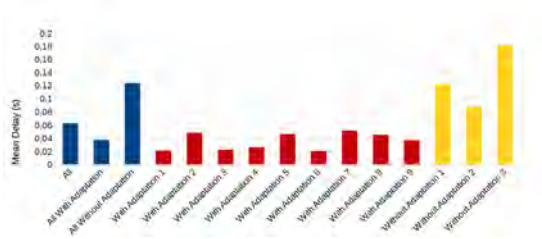


Fig. 5. Mean delay between sending a packet and receiving the ACK

C. Number of Packets to Low Delay

Figure 6 shows the number of consecutive packets that have a delay greater than 0.3s. 0.3s is the threshold used to reduce the bit rate of the video. The fewer packets have a high delay, the better the video quality is. In flights with adaptation the periods of complete loss of video picture are far shorter than in flights without adaptation.

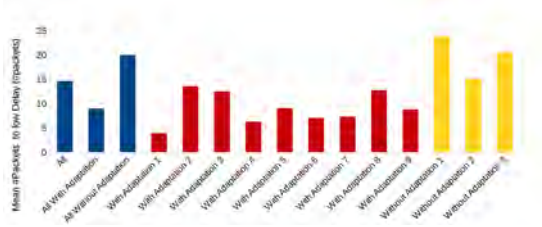


Fig. 6. Mean number of packets between the first packet with a long delay (>0.3s) until the delay is below 0.3s again

III. Reproducibility

Due to new laws, the authors could not conduct as many evaluation flights as orginally planned. The results rely on preliminary "ad-hoc" tests and do not address several influencing factors. Future work should therefore consider:

- The position of the base station and the UAV take-off spot should be kept stable.
- The UAV's flight path should not be manually controlled. Flying to a predefined set of GPS coordinates should be preferred.
- The UAV's altitude and cruise speed should be predefined.
- The UAV's yaw (and thus the antenna's orientation) should be predefined.
- The interferences of the fuselage and the mainboard on the antennas should be reduced by placing them on a base that extends below the UAV.

References

[1] S. Kacianka. Adaptive Video Streaming for UAV Networks. Master's thesis, Alpen-Adria-Universität Klagenfurt, 2014. `http://kacianka.at/ma/kacianka_thesis.pdf`.

[2] S. Kacianka and H. Hellwagner. Adaptive Video Streaming for UAV Networks. In *Proceedings of the 7th ACM International Workshop on Mobile Video*, MoVid '15, pages 25–30, New York, NY, USA, 2015. ACM.

[3] I. Kofler. *In-Network Adaptation of Scalable Video Content*. PhD thesis, Alpen-Adria-Universität Klagenfurt, 2010.

Network Connectivity in Mobile Robot Exploration

Torsten Andre[1], Giacomo Da Col[1,2], Micha Rappaport[1]

***Abstract*— In explorations it is often required for mobile robotic explorers to update a base station about the progress of the mission. Robots may form a mobile ad hoc network to establish connectivity. To connect mobile robots and base stations two strategies have been suggested: multi-hop and rendezvous. For multi-hop connections mobile robots form a daisy chain propagating data between both ends. In rendezvous, robots carry data by driving between communicating entities. While both strategies have been implemented and tested, no formal comparison of both strategies has been done. We determine various parameters having an impact on the exploration time of both strategies. Using a mathematical model we quantitatively evaluate their impact. Better understanding of the parameters allows to optimize both strategies for a given application. A general assertion whether rendezvous or multi-hop yields shorter exploration times cannot be made.**

I. Introduction

Mobile robots may be used in different applications in which the environment is unknown to robots. Robots have to explore and possibly map the unknown environment to deliver a map to users of the robot system or to allow future operation of the robots. Exploration may be performed by multiple robots. We assume that robot system users require to keep track of the progress of the exploration while the robots explore autonomously not requiring any user input. Fig. 1 depicts two strategies which have been proposed to establish connectivity. In Fig. 1a N robots form an ad hoc network in which robots are either explorers or relays. The relays form a daisy chain to connect explorers with a base station (BS). Robots may switch their role on demand. Relays propagate data from the explorers to the BS where the users of the system receive updates. Robots are connected either constantly or periodically [3]. When connected constantly, path planning [1] and/or selection of a robot's destination [2] consider connectivity between robots. The second strategy to establish connectivity is rendezvous [4], [5] illustrated in Fig. 1b. In comparison to periodic connectivity, a relay and a explorer meet at a location already known to both at a planned time to exchange data. The relay drives to the BS carrying the data. Once in communication range θ of the BS, it delivers the data and returns to the next rendezvous spot to meet with the explorers again. While both strategies have been analyzed, no formal attempt to compare both strategies has been made. Previous results were obtained by try and error [6]. We model both strategies mathematically to quantitatively compare their performance with respect to exploration time and discuss when to switch between strategies. Further, we quantitatively determine the impact of the parameters

- number of robots N,
- travel speed of robots v,
- environment size S and environment complexity ϕ,
- robot density $\rho = \frac{S}{N}$,
- update interval τ_u,
- communication range θ,
- type of relaying (parallel, sequential - see below)

on exploration time for a deeper understanding allowing improved design of rendezvous strategies.

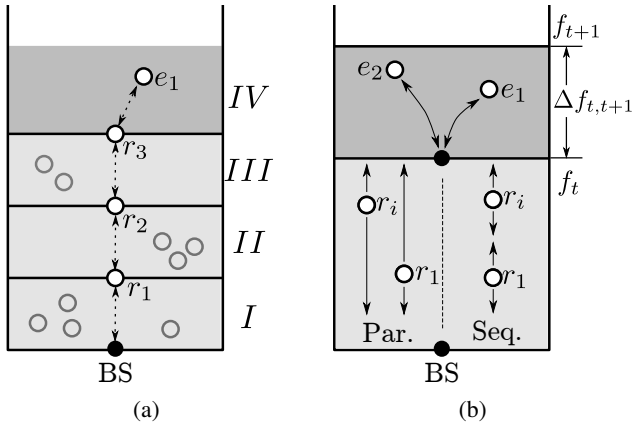


Fig. 1: Two strategies to establish connectivity between explorers and a BS: (a) multi-hop and (b) rendezvous.

II. Qualitative Strategy Comparison

Both strategies have strengths and weaknesses. While multi-hop communication allows communication between explorers and BSs with neglectable delay, rendezvous requires a relay to actually travel the distance between explorers and BSs increasing delay. Rendezvous has the advantage of allowing more robots to explore. Consider Fig. 1a in which the number of explorers decreases every time robots move out of the communication range of a relay. The environment is split into sectors of size of the communication range θ requiring a new relay. In the first sector I, out of $N = 4$, $E_t = 4$ robots may explore. All robots can communicate directly with the BS. When continuing exploration to sector II, one of the explorers is required to switch its role to relay to allow further connectivity between explorers and BSs. With each additional sector explored, the number of explorers decreases. Accordingly, the exploration radius for the multi-hop strategy is limited. Using the rendezvous strategy, the radius of operation is unlimited with respect to connectivity

[1]Networked and Embedded Systems, Alpen-Adria-Universitt Klagenfurt, Austria. torsten.andre@aau.at

[2] Università degli Studi di Udine, Italy

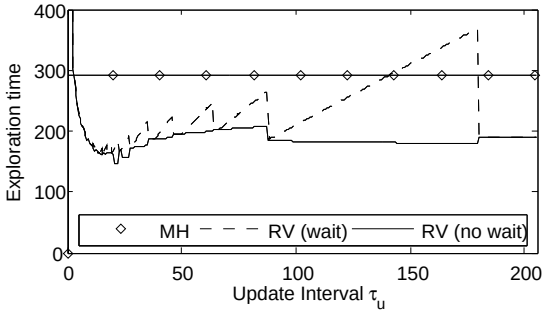


Fig. 2: Comparison of exploration time for S_{RV} and S_{MH}.

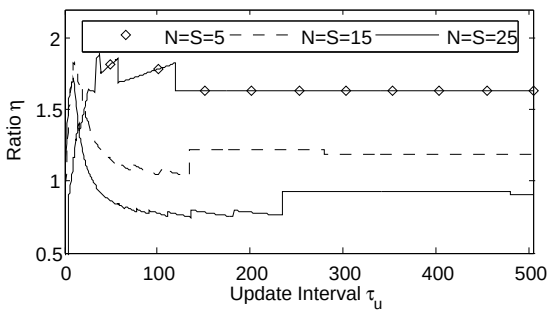


Fig. 3: Gain η for various environment sizes. For all environments the robot density $\rho = 1$.

to BSs. Though with increasing distance between explorers and BSs the update interval τ_u, i.e., the interval in which the BS receives new data, increases. If increasing τ_u is unacceptable, additional explorers have to switch roles to relays. More relays allow to decrease the update interval by pipelined data delivery. Referring to electrical circuits we distinguish *parallel* relays (see Fig. 1b left) from *sequential* relays (right).

III. Quantitative Strategy Comparison

A. Update Interval

The strategies S_{RV} and S_{MH} yield exploration times E_{RV} and E_{MH}, respectively. We only consider cases in which the number of robots N is selected large enough to allow a complete exploration for both strategies. Fig. 2 depicts exploration times for both strategies with identical parameters. We distinguish *no wait*, i.e., explorers return immediately having finished exploration, and *wait*, i.e., robots wait for the relay to arrive for the rendezvous. Not waiting for the relay has implications if explorers and the relay do not meet later on their way back. The exploration time for S_{RV} depends significantly on the update interval τ_u and the return strategy of explores. For S_{RV} (no wait) explorers return immediately having completed the exploration and do not wait for the relay to meet. If explorers finish exploration quickly and have to wait for a long time for the relay to return for a rendezvous, especially for large intervals significant time is spent waiting. In holds for S_{RV} that with increasing interval less relays are required freeing additional robots for exploring. In comparison, the exploration time for S_{MH} is constant. For the given configuration and number of robots S_{RV} reduces the exploration time by approx. 30 %. We continue to briefly discuss the impact of the number of robots N.

B. Number of Robots

We determine the ratio $\eta = \frac{E_{\text{MH}}}{E_{\text{RV}}}$, i.e., for $\eta < 1$ S_{MH} performs better, $\eta > 1$ S_{RV}. Fig. 3 depicts ratios η for various number of robots N and environment sizes S. For all graphs the robot density is $\tau = 1$.

With increasing number of robots η decreases. For large N the exploration time E_{RV} increases. Rendezvous can be planned only in space known to both explorers and relays. Dark gray space in Fig. 1b is known only to explorers. The more explorers are available, the deeper the explorers penetrate into unknown environments increasing $\Delta f_{t,t+1}$. The further robots push into unknown, the farther they have to travel back to the rendezvous which increases exploration time.

IV. Conclusions

We discuss the trade-offs of multi-hop and rendezvous strategies to establish connectivity between robots or robots and BSs. The communication delay between entities must be of inferior importance for rendezvous to be meaningful. If minimal delays are required, rendezvous is no suitable option and multi-hop communication is required. Instead, the exploration time is of higher importance under the requirement of updates in regular intervals. A variety of parameters has an impact on the performance of the multi-hop and rendezvous strategies. Understanding their impact and trade-offs allows to design optimized rendezvous strategies. A general assertion whether multi-hop or rendezvous performs better with respect to exploration time cannot be made.

Acknowledgment

This work was partly supported by the ERDF, KWF, and BABEG under grant KWF-20214/24272/36084 (SINUS). It was performed in the research cluster Lakeside Labs.

References

[1] P. Abichandani, H. Y. Benson, and M. Kam, "Multi-vehicle path coordination in support of communication," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2009.

[2] J. Yuan, Y. Huang, T. Tao, and F. Sun, "A cooperative approach for multi-robot area exploration," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, Oct. 2010.

[3] G. A. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," vol. 28, no. 4, pp. 967–973, Aug. 2012.

[4] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations," Autonomous Robots, vol. 11, no. 2, pp. 117–136, Sep. 2001.

[5] M. Meghjani and G. Dudek, "Combining multi-robot exploration and rendezvous," in *Proc. Canadian Conf. Computer and Robot Vision (CRV)*, May 2011.

[6] J. de Hoog, S. Cameron, and A. Visser, "Selection of rendezvous points for multi-robot exploration in dynamic environments," in *Proc. Int.Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS)*, May 2010.

Adaptive Feed Forward Control of an Industrial Robot

Dominik Kaserer[1], Matthias Neubauer[1], Hubert Gattringer[1] and Andreas Müller[1]

I. Introduction

Positioning and path accuracy is mandatory for state of the art robotic applications, where most of the path planning is done on computer systems. In most of the industrial robots a simple PD motor joint control is implemented, which suffers from quite high tracking errors. The performance of this controller can be improved by using a model based feed forward control. This model based feed forward control strongly depends on the dynamical parameters of the manipulator which have to be idenfified. However, parameters can vary during operation due to e.g. changes in friction conditions or changing loads. Therefore an online adaptation of the dynamical parameters is recommended. Two methods, namely the Recursive Least Squares (RLS) method and the Square Root Filtering (SRF) method, are presented and compared w.r.t tracking errors in the robots joints. The evaluation is done for a *Stäubli TX90L* industrial robot mounted on a linear axis leading to 7 degrees of freedom.

II. Dynamical Modeling - Identification

The equations of motion for mechanical systems can be written as

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q},\dot{\mathbf{q}}) = \overline{\mathbf{Q}}_M, \tag{1}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{g}$ contains nonlinear effects like Coriolis, centripedal, friction and gravitational forces, $\mathbf{q}$ are the minimal coordinates (joint angles) and $\overline{\mathbf{Q}}_M$ are the generalized motor torques. It is well known, that this equation is linear in the dynamic parameters $\mathbf{p}$ [3]

$$\overline{\mathbf{\Theta}}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\,\mathbf{p} = \overline{\mathbf{Q}}_M, \tag{2}$$

with the information matrix $\overline{\mathbf{\Theta}}$. The parametervector $\mathbf{p}$ consists of 10 parameters $\mathbf{p}_i$ for each body $i \in \{1 \dots 7\}$

$$\mathbf{p}_i^T = \left(m, m, r_{C_x}, m, r_{C_y}, m, r_{C_z}, A, B, C, C_M, r_c, r_v\right)_i. \tag{3}$$

m stands for the mass, $r_{C_j}, j \in \{x,y,z\}$ is the distance of the center of gravity in direction j, A, B and C denote the principle moments of inertia in the principal coordinate system, C_M is the moment of inertia of the motor and r_c, r_v are the Coulomb and viscous friction coefficients, respectively. Combining all parameters $\mathbf{p}_i$ in one vector $\mathbf{p}^T = \left(\mathbf{p}_1^T\ \mathbf{p}_2^T\ \dots\ \mathbf{p}_7^T\right)$, regularization (elimination of linear dependent parameters) of the identification problem (2) using a *QR-decomposition* leads to 45 independent parameters ($\overline{\mathbf{\Theta}} \in \mathbb{R}^{(7,45)}$, $\mathbf{p} \in \mathbb{R}^{45}$, $\overline{\mathbf{Q}}_M \in \mathbb{R}^7$). Performing m measurements (taken at instants of time t_k, $k = 1..m$) extends (2) to

$$\underbrace{\begin{bmatrix}\overline{\mathbf{\Theta}}|_{t_1}\\ \vdots \\ \overline{\mathbf{\Theta}}|_{t_m}\end{bmatrix}}_{\mathbf{\Theta}}\mathbf{p} = \underbrace{\begin{bmatrix}\overline{\mathbf{Q}}_M|_{t_1}\\ \vdots \\ \overline{\mathbf{Q}}_M|_{t_m}\end{bmatrix}}_{\mathbf{Q}}. \tag{4}$$

The Least Squares solution

$$\mathbf{p} = (\mathbf{\Theta}^T\mathbf{\Theta})^{-1}\mathbf{\Theta}^T\mathbf{Q} \tag{5}$$

gives an offline solution for the parameters and starting values for the online estimation of $\mathbf{p}$.

III. Online Parameter Estimation

The amount of available equations increases with proceeding time, because each time step delivers additional entries in (4). Therefore it is not suitable to solve the whole system of equations every time step again. Recursive algorithms for online parameter estimation can be used to avoid this problem.

In the following, two methods for this are evaluated, implemented and discussed in detail.

A. Recursive Least Squares

One way of solving the estimation problem is using a least squares technique. Starting from the solution (5) for the parameters $\mathbf{p}$ and the matrix $\mathbf{P}_0 = \mathbf{\Theta}^T\mathbf{\Theta}$ the adapted solution at each time step with the current measurements (index k) can be obtained by the *Constant Trace Recursive Least Squares (RLS) Algorithm* with exponential forgetting according to [1]. At each time step the following computations have to be done

$$\hat{\mathbf{p}}_k = \hat{\mathbf{p}}_{k-1} + \mathbf{K}_k\left(\overline{\mathbf{Q}}_{M,k} - \overline{\mathbf{\Theta}}_k\,\hat{\mathbf{p}}_{k-1}\right) \tag{6}$$

$$\mathbf{K}_k = \mathbf{P}_{k-1}\,\overline{\mathbf{\Theta}}_k^T\left(\lambda\,\mathbf{I} + \overline{\mathbf{\Theta}}_k\,\mathbf{P}_{k-1}\,\overline{\mathbf{\Theta}}_k^T\right)^{-1} \tag{7}$$

$$\tilde{\mathbf{P}}_k = \frac{1}{\lambda}\left(\mathbf{I} - \mathbf{K}_k\,\overline{\mathbf{\Theta}}_k\right)\,\mathbf{P}_{k-1} \tag{8}$$

$$\mathbf{P}_k = \frac{c_1}{\mathrm{trace}(\tilde{\mathbf{P}}_k)}\,\tilde{\mathbf{P}}_k + c_2\,\mathbf{I}. \tag{9}$$

to get updated dynamical parameters $\hat{\mathbf{p}}_k$. $\overline{\mathbf{Q}}_{M,k} \in \mathbb{R}^7$ denotes the actual motor torques, $\overline{\mathbf{\Theta}}_k \in \mathbb{R}^{(7,45)}$ is the information matrix of the actual measurements $(\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k)$, $0 < \lambda \leq 1$ indicate a factor for exponential forgetting

This work was supported by the Linz Center of Mechatronics (LCM) in the framework of the Austrian COMET-K2 programme

[1]Institute of Robotics, Johannes Kepler University Linz, Altenbergerstr. 69, 4040 Linz, Austria dominik.kaserer@jku.at

of old measurements, and $c_1 > 0$, $c_2 > 0$ are tuning parameters that are usually chosen such that $c_1 \approx 10^4\, c_2$ is satisfied. The scaling (9) of $\mathbf{P}_k$ is necessary to avoid an estimator windup, when the robot is not excited sufficiently.

B. Square Root Filtering

The square root filtering (SRF) is based on a Kalman filter approach and therefore additionally takes measurement noise into account. In case of conventional Kalman filtering, an error covariance matrix $\mathbf{P}_k$ has to be propagated from one time step to another. In case of SRF, a square root error covariance $\mathbf{S}_k$ instead is applied ($\mathbf{P}_k = \mathbf{S}_k^T\, \mathbf{S}_k$). Algebraically, the two approaches deliver the same result, but in case of limited computational accuracy the SRF technique has improved numerical capabilities. If $\kappa(\mathbf{P}_k)$ denotes the condition number of $\mathbf{P}_k$, then the condition number $\kappa(\mathbf{S}_k)$ is the square root of $\kappa(\mathbf{P}_k)$. Therefore, SRF is able to handle ill-conditioned identification problems much better than conventional Kalman filtering. In [2], several implementations are suggested. The most suitable version is chosen and extended by an exponential forgetting factor λ:
Measurement Update:

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k + \mathbf{K}\, \left(\overline{\mathbf{Q}}_{M,k} - \overline{\mathbf{\Theta}}_k\, \hat{\mathbf{p}}_k\right) \tag{10}$$

$$\mathbf{S}_{+,k} = \mathbf{S}_k - \gamma\, \mathbf{K}\, \mathbf{F}^T \tag{11}$$

$$\mathbf{K} = a\, \mathbf{S}_k\, \mathbf{F} \tag{12}$$

$$\mathbf{F} = \mathbf{S}_k^T\, \overline{\mathbf{\Theta}}_k^T \tag{13}$$

$$1/a = \mathbf{F}^T\, \mathbf{F} + \mathbf{R}_k \tag{14}$$

$$\gamma = 1/\left(1 + \sqrt{a\, \mathbf{R}_k}\right) \tag{15}$$

Time Update:

$$\overline{\mathbf{S}}_{k+1}^T = \mathbf{T}\, \mathbf{S}_{+,k}^T \tag{16}$$

$$\mathbf{S}_{k+1} = \lambda^{-1/2}\, \overline{\mathbf{S}}_{k+1}. \tag{17}$$

$\mathbf{R}_k$ denotes the covariance matrix of the measurement noise, see [2] for details. It is obvious that $\mathbf{P}_k$ is guaranteed to be positive semidefinite (because $\mathbf{S}_k$ is updated) at every time, which is a major improvement to conventional filtering, where (due to e.g. roundoff errors) $\mathbf{P}_k$ can become indefinite.

IV. Experimental Evaluation

The presented algorithms are verified for the industrial robot with two different trajectories. The first one is optimized for parameter identification and therefore excites all parameters sufficiently. The resulting tracking errors $e_i := q_{id} - q_i$ for axes $i = 2, 3, 6$ are depicted in Fig. 1. They are normalized to the maximum value ($\pm 2e^{-4}$ rad) of the tracking error with offline identified parameters. The dashed line indicates that the feed forward control is enabled at $t = 15$ s, and deactivated before. The second trajectory is a standardized testing trajectory (EN ISO-Norm 9283), consisting of straight and circular paths. The resulting normalized error is depicted in Fig. 2 (same illustration as in Fig. 1). The RLS procedure subsequently reduces the tracking error to a slightly lower value than the implemented SRF technique does, but the parameters during the estimation process vary much more compared to the implemented SRF approach (using other settings, the SRF approach also reduces the tracking error up to the acuracy achieved by the RLS, but on the expense of significantly higher parameter variations). For practical implementations the SRF technique might be preferable due to the better numerical characteristics, as described in Sect. III-B.

The offline identified parameters fit the physical parameters of the real robot very well. Therefore the feed forward control is able to reduce the tracking error to a very low value (the maximum joint error without feed forward control is about a factor 4 higher). Both adaptive algorithms are able to further reduce the tracking error. For the RLS the maximum tracking error is improved by 35% while for the SRF it is improved by 21%.

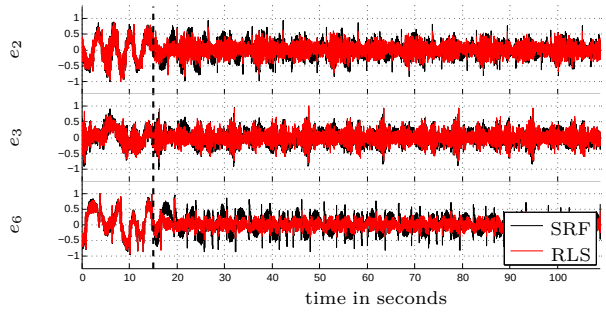


Fig. 1. Validation on optimal identification trajectory

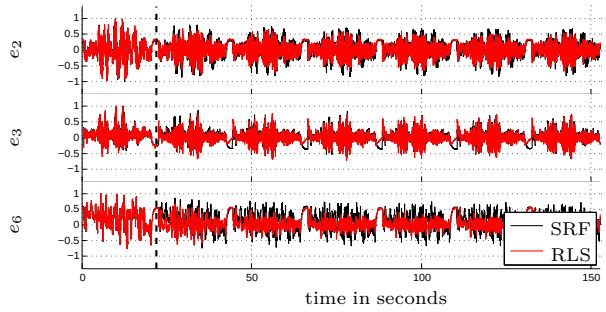


Fig. 2. Validation on testing-trajectory EN ISO-Norm 9283

References

[1] K.J. Åström, B. Wittenmark, Adaptive Control, Second Edition. Addison-Wesley Publishing, 1995.

[2] P. Kaminski; A.E. Bryson; S. Schmidt, Discrete square root filtering: A survey of current techniques, Automatic Control, IEEE Transactions on , vol.16, no.6, pp.727,736, Dec 1971

[3] M. Gautier, Identification of robots dynamics, Proc. IFAC Symp. on Theory of Robots, Vienne, Austria, December 1986, p. 351-356.

Laser Range Data Based Room Categorization Using a Compositional Hierarchical Model of Space

Peter Uršič[1] and Aleš Leonardis[2] and Danijel Skočaj[1] and Matej Kristan[1]

Abstract— **For successful operation in real-world environments, a mobile robot requires an effective spatial model. The model should be compact, should possess large expressive power and should offer good scalability. We propose a new compositional hierarchical representation of space that is based on learning statistically significant observations, in terms of the frequency of occurrence of various shapes in the environment. We have focused on a 2d ground-plan-like space, since many robots perceive their surroundings with the use of a laser range finder or a sonar. We demonstrate the performance of our representation in the context of room categorization problem.**

I. INTRODUCTION

The choice of spatial representation plays an important role in the process of designing a cognitive mobile robot, because a model influences the performance of several spatially related tasks. Moreover, designing an efficient spatial model is a challenging problem. The model should be as compact as possible, while simultaneously it needs to be able to efficiently represent huge variety of the environment. Our goal is to address the scalability issue of spatial representations. The main idea of our work is that a robot, which has already observed a lot of different environments, should have formed some understanding about general structure of spaces, and then use this knowledge when it arrives into the new environment. For example, if a robot has already seen a large number of apartments, when it arrives into the next one, it should already have some clue about what it will see. Current state-of-the-art robotic systems usually act like the newly observed space has nothing in common with previously observed ones. Therefore, the size of the generated model grows linearly with respect to the number of memorized spaces. We propose a model that can incorporate prior knowledge about the general characteristics of space, learned from previous observations, which enable the service robot to possess a scalable representation and thus reason about the newly observed place efficiently.

Hierarchical compositional models have been shown to possess many appealing properties, which are suitable for achieving our goals. A central point of these models is that their lowest layer is composed of elementary parts, which are combined to produce more complex parts on the next layer. Fidler et al. [1] have shown that a hierarchical compositional model allows incremental training and significant sharing of parts among many categories. Sharing reduces the storage requirements and at the same time makes inference efficient.

We adapt the hierarchical model from [1] to develop a description suitable for representation of space. Our proposed model is called the Spatial Hierarchy of Parts (sHoP). The representation is two dimensional, and based on data obtained with a laser range finder.

To demonstrate the suitability of our representation, we perform a series of experiments in the context of room categorization problem. Experiments performed on a demanding dataset demonstrate that our method delivers state-of-the-art results.

II. RELATED WORK

Some hierarchical representations of space have already been proposed [2]. Our representation differs from the existing approaches for spatial modeling in that we use a hierarchical compositional model on the lowest semantic level, where range sensors are used to observe the environment. Most related to our work are the approaches which perform room categorization based only on data obtained with range sensors. In [3] AdaBoost was used and categorization was based on a single scan, while in [4] Voronoi random fields (VRFs) were employed to label different places in the environment.

III. SPATIAL HIERARCHY OF PARTS

A laser-range finder mounted on a mobile robot is used to provide ground-plan-like observations, which represent partial views of the environment. Based on range data the sHoP learning algorithm [5] learns a hierarchical vocabulary of parts (Fig. 1). Parts represent spatial shape primitives with a compositional structure. Lowest layer is the only fixed layer consisting of 18 small line fragments in different orientations. In the following layers, parts are rotationally invariant and each part is a composition of two parts from the previous layer. If two compositions vary in structure to some allowed extent, they are considered to represent the same part. Therefore, small flexibility of part structure is allowed. Each layer contains only those parts, which were observed most frequently in the input data and each part is stored only in a single orientation. Learnt lower layers of the hierarchy are category-independent, therefore, each part is shared amongst all categories. This representation ensures good scalability with respect to the number of modeled categories.

[1]Authors are with Faculty of Computer and Information Science, University of Ljubljana {peter.ursic, danijel.skocaj, matej.kristan}@fri.uni-lj.si

[2]Aleš Leonardis is with CN-CR Centre, School of Computer Science, University of Birmingham a.Leonardis@cs.bham.ac.uk

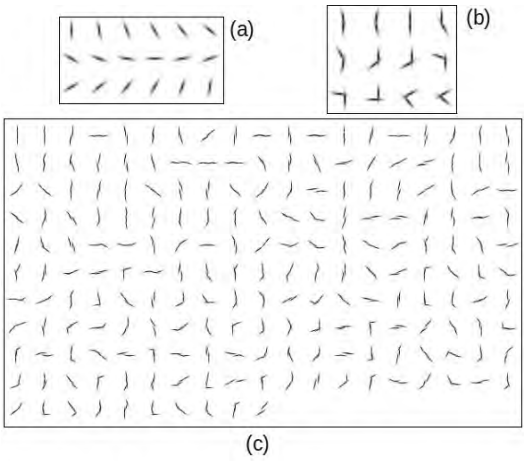


Fig. 1. Lower layers of the sHoP model. (a) Layer 1 parts. (b) Layer 2 parts. (c) Layer 3 parts.

IV. ROOM CATEGORIZATION

The low-level descriptor, called Histogram of Compositions (HoC), is used to perform room categorization. The elements of the hierarchy are used as the building blocks of the descriptor, which is then used as an input for categorization with a Support Vector Machine (SVM).

To form the descriptor, positions and orientations of parts are inferred from a range measurement obtained by the mobile robot. The positions of parts are rotated into a reference position, with the use of principal component analysis (PCA). The observed space is divided into several regions, with robot positioned in the center (Fig. 2). A sequence of histograms is created, where each histogram corresponds to a single region of the space division. The number of bins in each histogram is equal to the number of parts in the considered layer of the hierarchy. Each bin corresponds to one part type from that layer, while the value corresponding to the height of the bin equals the sum of confidences of parts in that region. All of the histograms are then concatenated into a single feature vector, forming the HoC descriptor.

V. EXPERIMENTS

Using a mobile robot, a large dataset has been obtained, called the Domestic Rooms (DR) Dataset, which is publicly available at http://go.vicos.si/drdataset. Dataset contains laser-scans and odometry readings obtained in several domestic rooms, corresponding to many different categories.

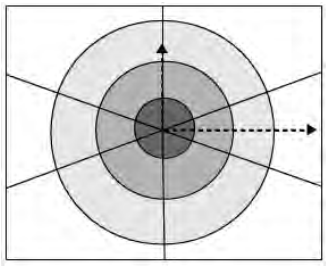


Fig. 2. The 24 regions of the image. The dashed arrows correspond to the principal axes.

Through extensive experiments on our dataset the efficiency of the proposed model is evaluated. Results of an experiment with 4 categories and Layer 3 of our hierarchy are shown in Table I. It consisted of 1000 trials, while in each trial train (80%) and test (20%) rooms were randomly chosen from the set of all rooms. Our work has also been compared to other approaches to room categorization [3]. Every laser scan obtained in a particular room has been categorized using the algorithm of [3], then, majority voting has been used to determine the room category. Our method achieves better performance with two categories, while the approach of [3] provides better results with two other categories. Experiments demonstrate that our approach delivers state-of-the-art results.

TABLE I

CONFUSION MATRIX FOR ROOM CATEGORIZATION EXPERIMENT USING LAYER 3 AND 4 CATEGORIES. CATEGORY ABBREVIATIONS: LR - LIVING ROOM, CR - CORRIDOR, BA - BATHROOM, BE - BEDROOM.

	Lr	Cr	Ba	Be
Lr	**82.74**	0	0.08	17.18
Cr	0	**91.60**	0	8.40
Ba	0.20	0.29	**92.96**	6.56
Be	13.32	0.22	16.24	**70.22**

VI. FUTURE WORK

It turns out that learning the hierarchy by following the proposed approach produces useful parts only up to a certain layer. Layer-four parts become already relatively large and therefore cover the original laser-scans quite poorly. In our future work, a few consecutive partial views, represented by lower-layer parts, will be merged together into a wider view of the environment. On top of the category-independent lower-layers of the hierarchy, which scale well with respect to the number of modeled categories, a new abstraction layer will be built, based on these wider views. The new layer will consist of category-specific parts. To ensure compactness and high expressiveness of the representation, only representative and discriminative parts will be found for each category, and only those will be stored in the model. This will provide good scalability with respect to the number of modeled entities within each category.

REFERENCES

[1] S. Fidler, M. Boben, and A. Leonardis, "Evaluating multi-class learning strategies in a generative hierarchical framework for object detection," in *NIPS*, vol. 22, 2009, pp. 531–539.

[2] B. Kuipers, R. Browning, B. Gribble, M. Hewett, and E. Remolina, "The spatial semantic hierarchy," *Artificial Intelligence*, vol. 119, pp. 191–233, 2000.

[3] O. Mozos, C. Stachniss, and W. Burgard, "Supervised learning of places from range data using adaboost," in *ICRA*, 2005, pp. 1730–1735.

[4] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields: Extracting the topological structure of indoor environments via place labeling," in *IJCAI*, 2007, pp. 2109–2114.

[5] P. Uršič, D. Tabernik, M. Boben, D. Skočaj, A. Leonardis, and M. Kristan, "Room categorization based on a hierarchical representation of space," *IJARS*, vol. 10, 2013.

An Autonomous Multi-UAV System for Search and Rescue

Jürgen Scherer[1] Saeed Yahyanejad[1] Samira Hayat[1] Evsen Yanmaz[3]
Torsten Andre[1] Asif Khan[1] Vladimir Vukadinovic[1]
Christian Bettstetter[1,3] Hermann Hellwagner[2] Bernhard Rinner[1]

[1]Institute of Networked and Embedded Systems
[2]Institute of Information Technology
University of Klagenfurt, Austria
[3]Lakeside Labs GmbH, Klagenfurt, Austria
<firstname.lastname>@aau.at

I. Introduction

Autonomous unmanned aerial vehicles (UAVs) are used with increasing interest in civil and commercial applications. UAVs can be equipped with imaging sensors and can provide aerial images of a target scene. Small-scale UAVs are constrained by limited flight time, and therefore, a team of UAVs is often used to speed up coverage of large areas. This is important for time-critical scenarios like search and rescue (SAR) missions.

In the scope of the SINUS[1] project, we aim to set up a SAR mission [1] by using an autonomous multi-UAV system. We call a multi-UAV system autonomous if it can change its behavior in response to certain events during the operation. The goal of such a mission is to locate a target such as a person or an object of interest using on-board sensors. Once the target is identified, a video stream showing the target is sent to first responders. To stream the video over a large distance, multiple relaying UAVs might be necessary. For this purpose, UAVs reposition to form a chain of relays. After achieving the required formation, a video of the detected target is transmitted either through the other UAVs acting as relays, or directly to the base station and first responders.

We summarize such missions into the following phases:

1) **Pre-planning**: The human operator defines the search region in the control base station. The optimal flight paths for all UAVs are computed to reduce the required time to search the area. Generated plans including the way-points are sent to individual UAVs.
2) **Searching**: The UAVs autonomously follow their predefined way-points while scanning the ground. The detection, collision avoidance, and frequent image transfer are active at this phase.
3) **Detection**: Upon detection of a target, the detecting UAV hovers while the other UAVs form a new formation.
4) **Repositioning**: The UAVs switch mode from searching to propagating. They change formation and set up a multi-hop link to allow viewer base stations to evaluate the situation. The location of the target is indicated at the viewer base station.
5) **Streaming**: The UAVs surveil the target by propagating videos or pictures.

In this paper we introduce an architecture for SAR scenarios, which provides flexibility to change the decision-making and autonomy level based on the application and user demand. In our system, we may have a heterogeneous set of UAVs and sensors, and failure of one UAV will not affect the whole mission. To allow easy deployment of the system, human intervention is minimized. To fulfill these requirements, we need a reliable wireless communication infrastructure. Imagine a scenario where QoS demands are such that a high resolution of captured aerial images or a video of the target is required. In order to transfer such data, we need to be able to estimate the required bandwidth, the throughput versus range relationship for the considered technology, as well as current and future UAV positions.

II. System description

The architecture design is supposed to handle the five phases explained in Section I. Fig. 1 shows the main components of our system. A *viewer base station* allows to connect to the system to receive sensor data. Multiple viewer base stations may exist providing visual feedback of the ongoing mission execution (e.g., current UAV positions overlayed on a map, received images, battery level and other information). A single *control base station* controls various aspects of the system. The UAVs and the base stations communicate over a wireless network. At the control base station, initial mission parameters such as mission area, number of UAVs to use, etc. are defined by the user via a user interface. The user can also supervise the mission execution and interact with the system at this base stations.

For the implementation we use the middleware Robot Operating System (ROS), which enables a flexible and modular design of the system. It uses TCP to exchange messages between modules and offers different message exchanging paradigms (e.g., publish/subscribe or service calls[2]).

[1]Self-organizing Intelligent Network of UAVs (uav.aau.at)

[2]wiki.ros.org

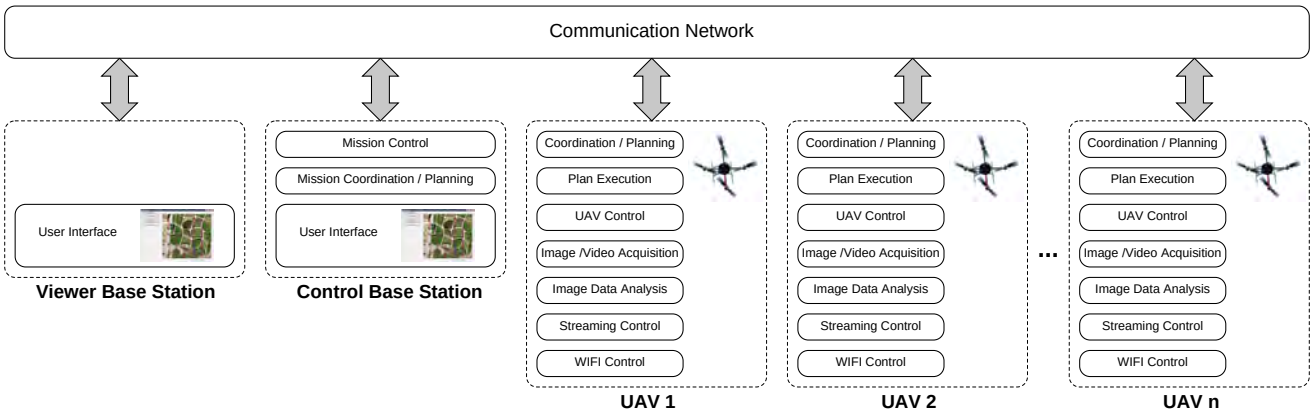


Fig. 1. System components.

The system comprises different modules that are located on the base station and the UAVs, see Fig. 1. The *Coordination / Planning* module exchanges information with the base station or with the *Coordination / Planning* module of other UAVs to provide high level coordination. High level coordination means generating feasible plans for several UAVs. A plan is a sequence of actions that are executed by a UAVs *Plan Execution* module. The *Plan Execution* module controls the behavior of other modules by sending control commands and is also responsible for low level coordination. Low level coordination means coordination between UAVs while a plan is executed, e.g. to avoid collisions. The *UAV Control* module is the interface to the UAV hardware and receives commands like *go to way-point*. The *Image / Video Acquisition* module accesses the camera and provides image and video data to the other modules. This image data is analyzed by the *Image Data Analysis* module for the sake of target detection. The *Streaming Control* module streams image and video data to the base station and adjusts the quality according to the network quality [2]. The *WIFI Control* module controls the behavior of the wireless network module (e.g., force the route of packets along a chain of relaying UAVs) and provides information about the current network quality.

For the establishment of a reliable distributed multi-UAV system, it is necessary to consider the demands posed by such a system in terms of networking of the UAVs and base stations. An aerial network in three dimensional space would benefit from antennas with nearly isotropic radiation intensity patterns. Also, to enable distributed online decision making, it is necessary to have real-time communication amongst the devices. In the SINUS project, all these requirements are addressed by using commercial off-the-shelf technology. An antenna structure in the shape of a horizontal equilateral triangle is introduced to provide isotropic coverage [3]. The requirement for peer-to-peer connectivity between the devices is addressed using an ad-hoc network. The standard IEEE 802.11s mesh technology is used for this purpose. A performance analysis was performed in [4].

III. System demonstration

For the SAR demonstration we use four UAVs with different cameras and processor boards. On all the UAVs and the base station, Ubuntu 12.04 was installed, and all were equipped with a WIFI module, which can be operated in 802.11s mesh mode. In our demo the target is identified by red color detection. In our test mission, the detection happens approximately 100 m away from the base station. We have set the minimum relay distance parameter to 30 m, which means for distances less than 30 m, a relay is not necessary and the video can be transmitted directly to the base station. However, for a distance d greater than 30 m, the number of relay positions is calculated by $\lfloor d/30 \rfloor$. The detecting UAV calculates the relay positions based on its own position and number of available UAVs. This information is sent to the other UAVs and based on their distances to the relay positions they come to a consensus on choosing their relay positions. After repositioning, a video is streamed to the base station through the relaying UAVs. Videos demonstrating the system are available on our website[3].

Acknowledgments

The work was supported by the ERDF, KWF, and BABEG under grant KWF-20214/24272/36084 (SINUS). It has been performed in the research cluster Lakeside Labs.

References

[1] T. Andre, K. Hummel, A. Schoellig, E. Yanmaz, M. Asadpour, C. Bettstetter, P. Grippa, H. Hellwagner, S. Sand, and S. Zhang, "Application-driven design of aerial communication networks," *Communications Magazine, IEEE*, vol. 52, no. 5, pp. 129–137, May 2014.

[2] S. Kacianka and H. Hellwagner, "Adaptive video streaming for UAV networks," in *Proceedings of the 7th Workshop on Mobile Video*, ser. MoVid'15, New York, NY, USA, 2015. [Online]. Available: http://doi.acm.org/10.1145/2727040.2727043

[3] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility." in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2013, pp. 120–124.

[4] E. Yanmaz, S. Hayat, J. Scherer, and C. Bettstetter, "Experimental performance analysis of two-hop aerial 802.11 networks," in *Proceedings of the Wireless Communication and Networking Conference*. IEEE, April 2014.

[3] uav.lakeside-labs.com/media/video-clips

Automatic Coverage Planning System for Quality Inspection of Complex Objects *

Michael Hofmann, Markus Ikeda, Jürgen Minichberger, Gerald Fritz, Andreas Pichler

***Abstract*—The task of coverage path planning is integral to many robotic applications. This paper presents an automatic coverage planning and data acquisition method with a minimal effort on configuration. The approach is integrated in a quality inspection system for complex and unknown geometric objects.**

I. Introduction

The task of coverage path planning is integral to many robotic applications in both the two dimensional case as well as for the three dimensional space. The problem itself is defined by determining a path that passes all points of an arbitrary area or volume of interest while avoiding obstacles within the environment [1]. The challenge is related to the covering salesman problem, a variant of the well-known traveling salesman problem [2].

The set of approaches to solve the coverage planning problem can be classified into heuristic and complete methods depending on their guaranteed coverage of the object or space under investigation. Furthermore the classic differentiation between online and off-line algorithms can be applied. Online methods do not assume full prior knowledge of the environment or surrounding as a prerequisite of their planning action. The considerable body of literature on coverage planning methods is summarized in [3] and [1].

This paper proposes an automatic quality inspection system that uses a minimal effort on teach-in action to perform a complete surface inspection of objects with complex geometries. The system is easy to configure and is capable to document automatically an unknown geometry. Furthermore it is safe w.r.t. risk of injury and does not require additional safety fences or other encapsulations. Within this paper we focus on the automatic coverage planning and data acquisition process and propose a two phase planning approach to tackle the challenging task.

II. System Overview

Figure 1summarizes the overall system for the automatic surface inspection task. Starting from the bottom left corner the system need a minimal basic teach-in phase of 4 manually selected points and the cell configuration (ground plate and obstacles in the work-cell) to define a bounding box around the unknown object. The system then automatically derives the 3d model of the object using a RGB-depth sensor in the first phase. The second phase comprises the coverage planning and the image acquisition task for the later surface inspection. In case of an unknown object both the coverage path and all reference images with its acquisition positions are stored in a database (red arcs in Figure 1). For the inspection of further objects of the same type these images and the already stored coverage plan is utilized in the inspection process (bottom left corner) to perform the quality measurements of the surface and/or additional checks on the absence/present or correct placement of identification plate or other signs and tags.

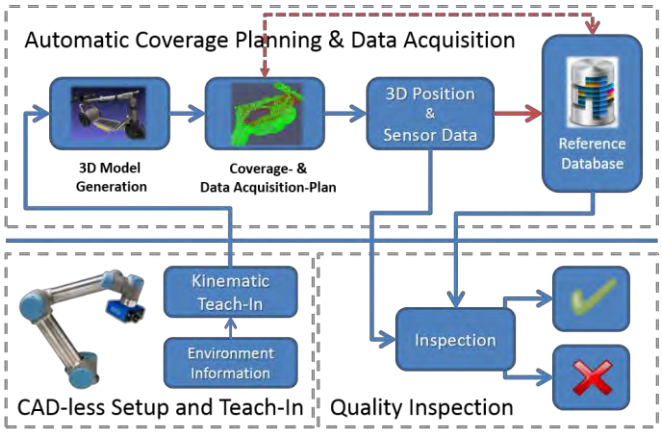


Figure 1: Two phase automatic coverage planning used for quality inspection of complex objects.

III. Automatic Documentation of Unknown Geometries

The execution and configuration of the system is separated in multiple stages. A prerequisite is the configuration of the geometrical circumstances (*CAD-less setup and teach-in*). This is done by configuring a robot motion around the intended volume of the unknown object under investigation. Typically four positions will be sufficient with the additional knowledge of the work-cell geometry. The motion is taught by manually placing the robot at known collision free positions.

Figure 2: Universal Robot UR10 with mounted Kinect and an example work piece (here the UCARVER of GW St.Pölten Integrative Betriebe GmbH).

A. 3D Model Generation

The first stage comprises the generation of a 3D model of the unknown object. During this stage the robot moves to the prior taught positions (corners of the bounding box) heading the RGB-depth sensor to the center of the volume. Simultaneously the sensor streams the captured 3D point cloud into the reconstruction software ReconstructMe [4] which finally derives the textured 3D model. An accurate

positioning of the robot is not necessary in this phase, since the reconstruction algorithm uses natural landmarks to guarantee the quality of the resulting model. Figure 3 depicts the result of the reconstruction stage for the example UCARVER.

Figure 3: The reconstructed object is used for the second phase of coverage path planning.

B. *Coverage & Data Acquisition Plan*

The second stage is utilizing the 3D model to calculate a map of inspection points of the object under investigation. All points have to be approachable by the robot. Therefore the inverse kinematic of the robot tool center point (TCP) is calculated for every inspection point. Depending on the process, additional parameters like surface normal angular offset are used to calculate appropriate inverse kinematic solutions. The calculation of the order in which the process points are approached is done by a simple but fast to calculate lexical order [3]. The lexical order guarantees a fast online calculation of the path with a fair execution time in contrast to more sophisticated methods like sample based approaches [5]. To prevent collisions during the transition of one inspection point to the next, an RRT Connect algorithm is used [6] and a Hybrid OBB-tree collision detector [7] is utilized to avoid them within the entire work cell. An example resulting path is shown in Figure 4. All points are connected with each other in a matter that transitions between inspection points are within a feasible distance. The highlighted area in the figure shows the lexical order of the points.

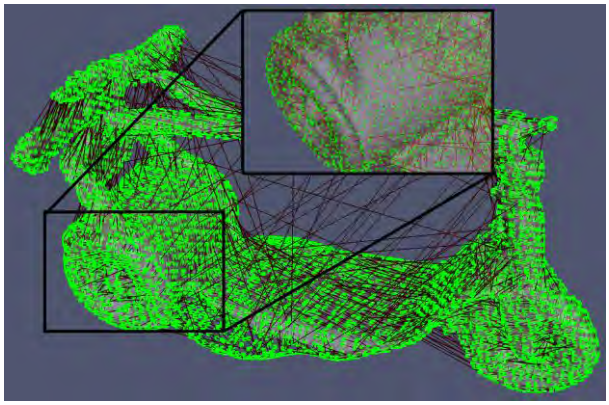


Figure 4: Derived coverage path plan using lexical ordering of inspection points (best viewed on screen).

C. *Results*

The execution time for deriving the coverage path is approx. 2 minutes using 10.000 inspection points on a standard PC. The lexical ordering takes 0.5 seconds, the inverse kinematic calculation 0.2 seconds and the remaining time is consumed by the collision checks. The high cost of the collision checking is the reason not to use more advanced algorithms to plan the path, because these algorithms need to calculate several transitions between inspection points, leading to even a higher calculation time of the coverage path. The accuracy of the image acquisition process for the latter quality inspection task is 0.1 mm and the errors during the 3D model generation (+/- 5mm using a low-cost Kinect Sensor) are compensated by using a fixture for positioning the object.

IV. CONCLUSION

An automatic coverage and data acquisition process for a quality inspection system was presented. The two phase process requires only minimal and easy configuration effort to inspect the entire surface of an object, which geometry is unknown beforehand. The first phase generates the 3D model of the object and afterwards a full surface coverage inspection plan is derived automatically for further quality inspection purposes.

ACKNOWLEDGMENT

This work was supported by the national IKT Project NexGen FFG Project 840214. We thank our project partner GW St.Pölten Integrative Betriebe GmbH for their assistance during the demonstration phase. As an integrative enterprise the company is manufacturing various kinds of products for major industrial customers. The company is an essential part of the Austrian disability policy. They are close connected to the AUVA which is the relevant public body for regulating ergonomic and safety aspects.

REFERENCES

[1] *E. Galceran and M. Carreras, "A survey on coverage path planning for robotics,"* Robotics and Autonomous Systems , *vol. 61, no. 12, pp. 1258-1276, 2013.*

[2] *E. M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem.,"* Discrete Applied Mathematics, *vol. 55, no. 3, pp. 197-218, 1994.*

[3] *H. Choset, "Coverage for robotics - A survey of recent results,"* Annals of mathematics and artificial intelligence, *vol. 31, no. 1-4, pp. 113--126, 2001.*

[4] *Profactor, "ReconstructMe - Digitize Your World," [Online]. Available: http://reconstructme.net. [Accessed 2015].*

[5] *F. S. H. Brendan Englot, "Sampling-Based Coverage Path Planning for Inspection of Complex Structures.,"* ICAPS, *2012.*

[6] *S. M. L. James J Kuffner, "RRT-connect: An efficient approach to single-query path planning,"* Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, *vol. 2, pp. 995--1001, 2000.*

[7] *S. C. D. Jia Pan, "FCL: A general purpose library for collision and proximity queries,"* Robotics and Automation (ICRA), 2012 IEEE International Conference on, *pp. 3859--3866, 2012.*

[8] *H. Choset, "Coverage for robotics - A survey of recent results,"* Annals of Mathematics and Artificial Intelligence, *vol. 31, pp. 113-126, 2001.*

Towards Safe Human-Robot Collaboration: Sonar based collision avoidance for robot's end-effector

Sriniwas Chowdhary Maddukuri, Martijn Rooker, Jürgen Minichberger, Christoph Feyrer, Gerald Fritz, and Andreas Pichler

***Abstract*— In most industrial applications, human workers and robots are purely isolated in space and/or time to avoid risky situations. Several works have been conducted for integrating human workers with robots, to achieve human-robot collaboration. The primary issue of concern involved with human-robot collaboration is the safety of the human worker. This paper presents a methodology to ensure safe human-robot collaboration through the detection of static/known and unknown objects and avoid collisions between robot's end-effector device and unknown objects using ultrasonic sensors.**

I. Introduction

Nowadays, human workers and robots have to share a common workspace which leads to high demand on enabling sufficient safety measures. The major challenge for a collaborative human-robot workspace is to come up with a real-time collision avoidance strategy which prevents collisions between the robot and dynamic object (for e.g., human worker). Therefore, the robot's workspace has to be consistently monitored over time to be aware of unexpected changes which happen in its surrounding environment. To achieve real-time collision avoidance, several approaches have been proposed where the monitoring of human-robot workspace is carried out through optimal placement of multiple depth sensors installed within the vicinity of the surrounding environment. The main disadvantage with this approach is the insufficient information about the occluded areas behind the detected obstacles. As a result, the occluded areas will lead to unexpected human-robot collisions. To overcome the occluded areas or blind spots created by static or dynamic objects and prevent collisions between the robot's end-effector and human worker, the approach adopted here is to place ultrasonic sensors around the robot's end effector.

A. Related Work

In general, collision avoidance is realized through continuous monitoring of the human-robot workspace using multiple RGB-D sensors. In [1][2][3][4][5][6], 3D depth information is obtained from RGB-D sensors to build up an internal model of the human-robot workspace. Using the internal model, collision avoidance is then performed based on the minimum distance between the robot and the detected unknown objects by employing the multiple depth images of unknown objects and extracted 3D model of the robot. A similar approach was adopted in [8] and [7], where the collision between the human worker and the robot are avoided through color and texture information and velocity and trajectory elements of the human-robot workspace respectively. In [9], collisions between the robot manipulator and obstacle are prevented by comparing individual depth and presence maps of the robot manipulator and detected obstacle. A distance processing algorithm was proposed for the adaption of robot manipulator's trajectory to realize collision avoidance through the monitoring of a confined working environment using ultrasonic sensors [10]. In [11], the Reis robot RV30-16 is safeguarded with an ultrasonic sensor system to monitor the shared human-robot workspace. A priori information about the shared human-robot workspace is created as a 3D sonar map and provided as a reference model to the collision avoidance algorithm. Here, the collision avoidance is realized by adapting the velocity of the robot with respect to the distance at which the obstacle is detected.

Sriniwas Chowdhary Maddukuri, Martijn Rooker, Jürgen Minichberger, Christoph Feyrer, Gerald Fritz, and Andreas Pichler are with the Robotics and Assistive Systems Department, PROFACTOR GmbH, Im Stadtgut A2, Steyr-Gleink, 4407 Austria (phone: +43(0)7252-885-307; fax: +43(0)7252-885-101; e-mail: sriniwas.maddukuri@profactor.at).

B. Objectives

The scientific objective is to derive an algorithm which is able to distinguish between static/known objects and unknown object and avoid collisions between the robot's end effector and human worker.

C. Paper Organization

Section II deals with the methodology proposed for collision avoidance with respect to an ultrasonic sensor setup. Section III presents the experimental results. Section IV concludes the research work and provides an outlook regarding the future work.

II. Methodology

The methodology proposed in this section focuses mainly on the collision avoidance algorithm. Furthermore, it will be defined how the distinction between static/known objects and dynamic objects is determined.

A. Collision Avoidance

Collision avoidance is realized in a sequence of two stages. In the first stage, a priori information about the robot's workspace comprised of static/known objects is obtained from the ultrasonic sensors by executing a pre-defined trajectory of the robot. Here, orthogonal placement of 5 ultrasonic sensors around the robot's end-effector was done to ensure considerable monitoring of the workspace. The distance values of the static/known objects are acquired as a data set and regarded as a reference model/known geometry of the working environment. In the second stage, the present distance values of the detected objects which represent the measured model/unknown geometry of the environment are then compared with distance values of the reference model. An error vector computed from the difference between the

distance values of the measured model at every time instant and the distance values of the reference model. The minimum of the error vector and a certain tolerance value (i.e., *tol = 20mm*) is then used for distinguishing the detected objects as static/known or dynamic over time. Based on this object determination, collision avoidance is performed with respect to the present distance at which the dynamic object is located using a pre-defined minimum obstacle distance [4][9].

III. Experimental Results

In the following, the algorithm defined for collision avoidance is validated by two sequentially performed experiments. In both the experiments, the robot executes a pre-defined trajectory in two stages. In the first stage, the robot system follows the pre-defined trajectory around a known object for multiple number of cycles (i.e., one single cycle is equivalent to the time taken by the robot to execute its trajectory) to acquire the reference model. In the second stage, the first stage is repeated in the absence or presence of a dynamic object (for e.g., human hand).

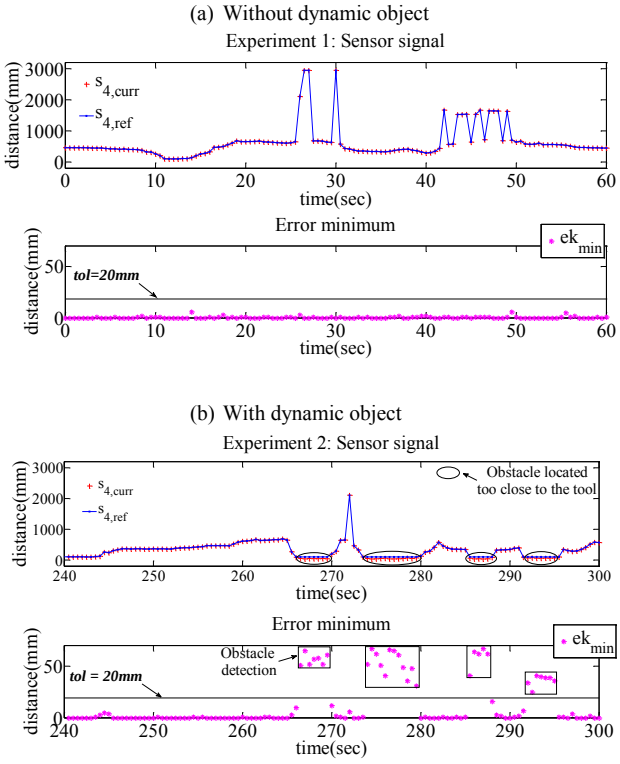


Figure 1. Single sensor signal (Sensor number 4) and its computed error minimum for two scenarios (a) without dynamic object and (b) with dynamic object

In Fig. 1, the sensor signal of reference model matches quite good with that of measured model corresponding to experiment 1. This is evident from the fact that the computed error minimum is below the assigned tolerance value. In the case of experiment 2, the motion of the robot is automatically interrupted in the presence of an obstacle and resumes only if the detected obstacle goes away from the end-effector. This can be clearly seen from the fact that sensor signal of the measured model remains at the present distance value with respect to the obstacle detected.

The results indicate that there is a very reliable detection of unknown/dynamic objects (e.g. human actively interfering in the trajectory of the robot system). Furthermore, the robot system reacts to the detected objects by either stopping and waiting until the objects has cleared the defined trajectory.

IV. Conclusion and Future Work

A collision avoidance algorithm is described for avoiding active collisions between the robot's end-effector and human worker and its performance is evaluated in real-time. The next step in this research work would be to realize the collision avoidance for the whole robot by realizing an optimal placement of ultrasonic sensors such that it covers the complete 360° view of the workspace.

Acknowledgment

This research has been funded by the Austrian Ministry for Transport, Innovation and Technology via the project KOMOPROD and the framework „Regionale Wettbewerbsfähigkeit OÖ 2007-2013“ within the project KOBEROB from funds of the European Regional Development Fund of the country of Upper Austria.

References

[1] Claus Lenz, Markus Grimm, Thorsten Röder and Alois Knoll, “Fusing multiple kinects to survey shared human-robot-workspaces”, Technical Report TUM-I1214, Technische Universität München, Munich, Germany, 2012.

[2] Henrich D., Gecks T, “Multiple-camera collision detection between known and unknown objects”, *2nd ACM/IEEE International Conference on Distributed Smart Cameras*, Stanford, 2008.

[3] Lihui Wang, Bernard Schmidt and Andrew Y.C. Nee, “Vision-guided active collision avoidance for human-robot collaborations”, Journal of Manufacturing Letters, Vol.1, Issue.4, October 2013, pp.5–8.

[4] Bernard Schmidt and Lihui Wang, “Depth camera based collision avoidance via active robot control”, Journal of Manufacturing Systems, Vol.33, Issue.4, October 2014, pp.711–718.

[5] Markus Fischer and Dominik Henrich, “3D collision detection for industrial robots and unknown obstacles using multiple depth images (Advances in Robotics Research, Torsten Kröger and Friedrich M. Wahl)”, Springer Berlin Heidelberg, 2009, pp111–122.

[6] Markus Fischer and Dominik Henrich, "Surveillance of robots using multiple colour or depth cameras with distributed processing," *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, vol., no., pp.1–8, Aug. 30 2009-Sept. 2 2009.

[7] Bosch, J.J.; Klett, F., "Safe and flexible human-robot cooperation in industrial applications," *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on*, vol., no., pp.107–110, 8–10 Oct. 2010.

[8] Kruger, J.; Nickolay, B.; Schulz, Oliver, "Image-based 3D-surveillance in man-robot-cooperation," *Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on*, vol., no., pp.411–420, 26–26 June 2004.

[9] Flacco, F.; De Luca, A., "Multiple depth/presence sensors: Integration and optimal placement for human/robot coexistence," *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, vol., no., pp.3916–3923, 3–7 May 2010.

[10] Christoph Glowa, Thomas Schlegl, „Ultrasound based object detection for human robot collaboration (Intelligent Robotics and Applications, Jangmung Lee, Min Cheol Lee, Honghai Liu and Jee-Hwan Ryu)“, Vol.8103, Springer berlin Heidelberg, 2013, pp.84–95.

[11] Östermann, B., Huelke, M., Kahl, A., „Freed from fences – Safeguarding industrial robots with ultrasound“, *Proceedings of 6th Working on Safety Conference, 2012 and Safety Science Monitor 17*, Issue.1, Article.2, pp.11, 2013.

Skill Adaptation through combined Iterative Learning Control and Statistical Learning

Miha Deniša, Andrej Gams and Aleš Ude

***Abstract*— To be successful on the market, many enterprises are customising ever more products, leading to small batch size production. Consequently, neither specialised production lines nor automated robot assembly are economically viable and manual labor is often the only possibility. In this paper we propose a new approach to quickly program robot assembly processes. To avoid manual programming of optimal robot behaviours for every single task and its variations, we combine ideas from learning by demonstration, iterative learning control, and statistical learning. The proposed method has been evaluated in a case study of a robot mounting doors in collaboration with human workers, where the size of the doors vary.**

I. INTRODUCTION

There are still no economically viable solutions next to manual labor when it comes to customized production. Having several under-utilized special production lines and special machines is too expensive for sustainable operation. Automated robot assembly could be a potential solution, but it is still very time consuming to reprogram an industrial robot for a new assembly task, integrate it in the work process, and to calibrate it in a new environment [1].

To overcome these problems we propose to use a combination of learning by demonstration, iterative learning control, and statistical learning to quickly program new assembly skills. Learning by demonstration is used to obtain a single example trajectory for the desired skill. Dynamic movement primitives (DMPs) [2] have emerged as a method for of encoding the trajectories from single user demonstrations. They provide flexibility to easy modulate and adaptat the programmed trajectories. To give robots the ability to adapt to varying conditions of the task or the environment, autonomous adaptation of trajectories through un-supervised or supervised exploration has been employed in robotics. One such method is reinforcement learning [3], but reinforcement learning often takes a large number of repetitions to converge to the desired behavior as a consequence of their unsupervised exploration. To reduce the number of needed iterations, DMPs have been employed in combination with iterative learning control (ILC) [4]. ILC usually converges much faster then reinforcement learning, but unlike reinforcement learning it needs a reference to converge to.

Even though the dimensionality of the complete robot motion space is infinite, many tasks can be described by low dimensional task parameters, which in the following are called query points. A set of example movements associated with variations of the desired task (with respect to the selected query points) can be used as input to statistical learning methods [5], [6] in order to generate optimal movements for new variations of the task. For this approach to work, the necessary condition is that example movements transition smoothly between each other as a function of the query point.

Miha Deniša , Andrej Gams and Aleš Ude are with the Humanoid and Cognitive Robotics Lab, Department for Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, 1000 Ljubljana, Slovenia {miha.denisa, andrej.gams, ales.ude} at ijs.si

II. THE METHOD

This paper combines the means of LbD, adaptation through ILC and statistical learning to achieve fast adaptation to the environment. It can be roughly divided into three parts:

1) A single demonstration trajectory $\mathbf{p}_d$ is gained by LbD and encoded as a DMP
2) Demonstrated trajectory $\mathbf{p}_d$ is modified for an external condition (query point q) through a coupling term $\mathbf{c}$ gained by ILC and human interaction. A set of coupling terms, $S = \{\mathbf{c}_m, q_m\}$, is gained by repeating this process for m query points and encoded as a set of weighted linear basis functions: $S = \{\mathbf{w}_m, q_m\}$.
3) Gaussian process regression (GPR) is employed to calculate the coupling term $\mathbf{c}$ for an arbitrary query point q within the area of the learned terms S. This new coupling term defines the needed robot trajectory for the given task variation.

The rest of this section briefly describes iterative learning control and statistical learning used in the proposed approach.

A. Iterative learning control

Initial demonstrated trajectory $\mathbf{p}_d$, encoded as a DMP, is modified through a coupling term [4],

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x), \quad (1)$$

$$\tau \dot{y} = z + c(k), \quad (2)$$

where k denotes the k-th time sample. The coupling term is gained through iterative learning control (ILC), which uses feedback error to improve the performance in the next repetition of the same behavior. We propose to apply the current-iteration ILC, which is given by the formula [7],

$$c_{j+1}(k) = \underbrace{Q(c_j(k) + Le_j(k+1))}_{\text{feedforward term}} + \underbrace{Ce_{j+1}(k)}_{\text{feedback term}}, \quad (3)$$

where c is the coupling term, k denotes the k-th time sample, j denotes the learning iteration, and Q and L are the learning parameters. C is the feedback gain. ILC is distinguished from

simple feedback control by the prediction of the error $e(k+1, j)$ in the $(j+1)$ iteration, which serves to anticipate the error caused by the action taken at the k-th time step. ILC modifies the control input in the next iteration based on the control input and feedback error in the previous iteration. In our case the error represents the robot's tracking error introduced through human intervention during the execution. For the human to be able to infer this error, the robot must exhibit some compliance.

B. Statistical Learning

ILC is used to adapt a demonstrated motion to new conditions, but several iterations are needed in order to learn an appropriate coupling term. Statistical learning can be used to gain new coupling terms adapted to new conditions. Let's assume a set of m example coupling terms, $S = \{\mathbf{w}_m, q_m\}$ which transitions smoothly between each other as a function of the query point q, and are encoded as a linear combination of weighted basis functions. For statistical learning we use Gaussian process regression (GPR), which can be used to learn a function,

$$F_s : q \mapsto \{\mathbf{w}\} \tag{4}$$

Once the function (4) is learned, it can be used to calculate appropriate coupling term trajectories, again defined by weights $\mathbf{w}$, for any given query q within the training space. Further details on statistical learning are omitted and the readers are referred to [5], [6].

III. CASE STUDY

The case study tackles the problem of a robot mounting a door of varying size on an electrical cabinet while cooperating with a human. As the robot lifts and positions the door, the human can mount the hinges and thus attach it on the cabinet. While this problem could be solved by programming a robot trajectory by hand, it would have to be done for each variation in the dimension of the door.

As an alternative the proposed approach was used. First, a single robot trajectory $\mathbf{p}_d$ was taught using LbD. This movement successfully placed a door of a certain size q_d on the cabinet's mounting position. Seven new motion trajectories $\{\mathbf{p}_m\},\ m = 1, \ldots, 7$ were gained by modifying the original trajectory for different door dimensions $\{q_m\}$ through coupling terms $\{\mathbf{c}_m\}$. These coupling terms were gained through human interaction and by using ILC as described in [4]. The new trajectories, as well as the original demonstrated trajectory, are shown in Fig. 1. For clarity the trajectories are presented in only one dimension.

In the next step, the set of coupling terms $S = \{\mathbf{w}_m, q_m\}$, used for modifications in the previous step, was used with statistical learning. GPR was used to learn a function (4) which was then used to compute coupling terms adapted for new query points, i.e., door dimensions. New position trajectories, generalized to new example door dimensions, are shown in Fig. 1. We can see that the shape of the trajectories was preserved.

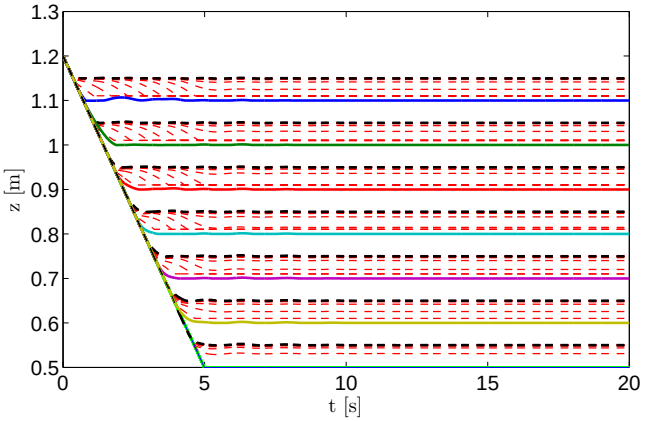


Fig. 1. Robot's position trajectories in z-axis while mounting doors of different dimensions. The bottom solid line in light blue denotes the original demonstrated trajectory $\mathbf{p}_d$. While the red dashed lines present trajectories while modifying them through human coaching and ILC, the black dashed lines denote the final learned trajectories. Solid lines of varying colors present trajectories gained through statistical generalization.

IV. CONCLUSION

In this paper we presented an approach used for gaining new robot trajectories based on learning by demonstration, iterative learning control and statistical learning. We believe the proposed approach would prove useful in the industry prone to customizing products and lowering production batches. As an alternative to hard programming robot trajectories for each small batch and variation in the product, this approach shows adaptability through a low number of human demonstration and interventions. The case study done in this paper further indicates the usefulness of the proposed approach. While the selected task is elementary, it shows the applicability of the method, as the trajectories adapt to new queries while maintaining the needed shape. In the future we plan to asses the approach by executing more complex tasks using a real robot arm.

REFERENCES

[1] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald *et al.*, "The kuka-dlr lightweight robot arm-a new reference platform for robotics research and manufacturing," in *Robotics (ISR), 2010 41st international symposium on and 2010 6th German conference on robotics (ROBOTIK)*. VDE, 2010, pp. 1–8.

[2] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[3] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, p. 0278364913495721, 2013.

[4] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," 2014.

[5] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *Robotics, IEEE Transactions on*, vol. 26, no. 5, pp. 800–815, 2010.

[6] D. Forte, A. Gams, J. Morimoto, and A. Ude, "On-line motion synthesis and adaptation using a trajectory database," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1327–1339, 2012.

[7] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

Interface for Capacitive Sensors in Robot Operating System

Stephan Mühlbacher-Karrer[1], Wolfram Napowanez[1], Hubert Zangl[1], Michael Moser[2] and Thomas Schlegl[2]

***Abstract*— In this paper we present a software interface for capacitive sensors for the Robot Operating System (ROS). We demonstrate the capability of the interface using the Kobuki platform. Our system enables detection of humans and collision avoidance in the surroundings of a robot without the need for vision or laser scanner based object detection systems. Consequently, it can be a valuable complementary sensor system whenever optical systems can not be used. The interface allows to use various sensors architectures using different number of electrodes or measurement modes. In addition, the detection and reconstruction algorithm of the sensor interface can be adapted to the needs of the application.**

I. INTRODUCTION

Capacitive sensing is a well suited and known sensing technology in the field of robotics. It is utilized for tactile, pretouch and proximity sensing in different applications such as grasping [1] or collision avoidance [2]. The capacitive sensing principle is based on the interaction of an object and an electric field in the vicinity of the sensor front end given a dielectric environment, e.g., air. In previous work we transferred the Electrical Capacitance Tomography (ECT) to robotics [3]. Consequently, we further facilitate the applicability of capacitive sensors in the field of robotics by introducing a software interface for capacitive sensors in ROS [4]. It supports single and multi-modal capacitive sensors measuring in either one or both measurement modes as described in Section II. The interface also contains a sensor node for preprocessing of the sensor data with respect to visualization and high level applications. Our interface provides the capability to exploit the advantages of the capacitive sensing technology in the field of robotic and industrial applications, e.g., tiny and mechanically robust sensor front ends and the capability to "look" inside of non-transparent dielectric objects.

II. CAPACITIVE SENSORS

Capacitive sensing is based on the measured distortion of the electric field caused by an object in the vicinity of the sensor front end. The sensor front end consists of conductive electrodes insulated from the surroundings and is sensitive to dielectric and conductive objects [5]. Generally, two different sensing modes are distinguished:

- The single ended measurement mode is utilized to determine the capacitance between the transmitter electrode and the far ground. An excitation signal is applied to a transmitter electrode and its displacement current is measured (see Fig 1).
- The differential measurement mode is utilized to determine the capacitance between the transmitter and receiver electrode. An excitation signal is applied to a transmitter electrode and the displacement current is measured at the receiver electrode (see Fig 1).

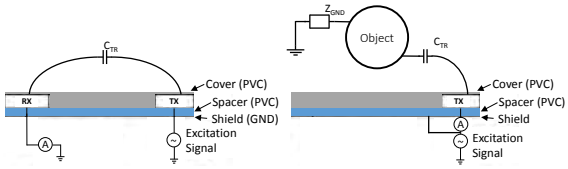


Fig. 1. Capacitive measurement modes. On the left and right side the differential and single ended measurement modes are depicted, respectively.

It should be noted that the sensing range strongly depends on the geometrical design of the electrodes.

III. SOFTWARE ARCHITECTURE

The interface consists of two ROS nodes called Read Sensor Data and Process Sensor Data (see Figure 2). The first node is responsible for collecting the raw sensor data of each connected capacitive sensor to assemble and publish the measured capacitances. Beside the measured data this message contains also a unique identifier of the sensor. This node provides the flexibility to use capacitive sensors different in architecture simultaneously and also the number of used sensors can be adapted to the needs of the application. All sensor parameters, e.g., number of electrodes, measurement mode, etc. are provided in a single parameter file. The second node is responsible for data processing, including a reconstruction/object detection algorithm and to prepare the visualization of the data. This node publishes two messages: one for the visualization of the sensor data provided as point-clouds in the ROS 3D visualization tool RVIZ and a second message containing the results of the reconstruction/detection algorithm for high level applications. Additionally, also ECT reconstruction images can be visualized in RVIZ.

IV. SENSOR PLATFORM

In our experiments we use the ROS supported robot platform Kobuki. We equipped Kobuki with two unique, wireless capacitive sensors on the front and rear side. This minimum number of sensors is sufficient as the platform

[1]Stephan Mühlbacher-Karrer, Wolfram Napowanez and Hubert Zangl are with the Institute of Smart System Technologies, Sensors and Actuators, Alpen Adria Universität, 9020 Klagenfurt, Austria stephan.muehlbacher-karrer@aau.at, wnapowan@edu.aau.at, hubert.zangl@aau.at

[2]Michael Moser and Thomas Schlegl are with the eologix sensor technology gmbh. 8010 Graz, Austria michael.moser@eologix.at, thomas.schlegl@eologix.at

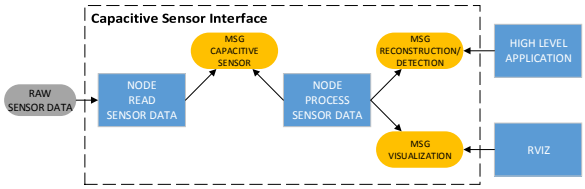


Fig. 2. Software architecture for capacitive sensors including ROS nodes and messages.

has a two wheel differential drive and can rotate around the central point of axis. Consequently, the robot is able to detect objects in its surrounding. One major advantage of capacitive sensing is flexible mounting. It can be either mounted on the surface, adding a layer of several $\sim 100\,\mu$m, or underneath the surface, to be protected against external forces thus further enhancing the mechanical robustness of the sensor system. In Figure 3 the hardware architecture used for our experiment is depicted. The sensor data of each capacitive sensor is forwarded via Radio Frequency (RF) link to a receiver module connected via USB to the robot's notebook, providing the raw sensor data to the capacitive sensor interface. It should be noted that each sensor can have a different number of electrodes. Additionally, also the sensor's measurement mode can vary. The parameters of each sensor are defined in a parameter file imported by the ROS node Read Sensor Data.

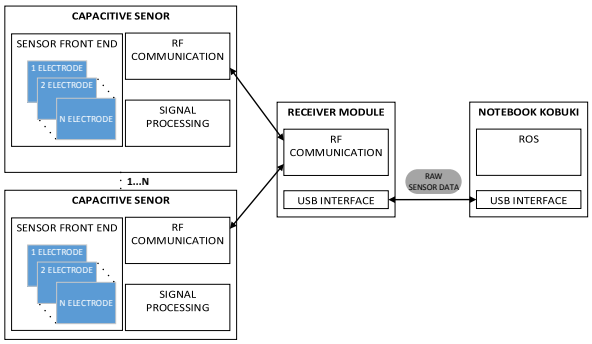


Fig. 3. Hardware architecture comprising wireless capacitive sensors, a receiver module and the robot.

V. EXPERIMENTAL SETUP AND RESULTS

In our experiment we implemented a binary threshold based collision avoidance for humans as high level application to demonstrate the usability of the interface. The robot moves forward or backward as long as no human object is detected in front or rear of the robot, respectively. The threshold values are determined in advance, where humans are positioned in front of the sensor front end. In addition an offset calibration is used for each sensor value to reduce the impact of model errors. It should be noted that this experiments are done without the support of any vision or laser based sensors. The experimental setup visualization results for a human object in front of the robot is depicted in Figure 4 and 5, respectively.

Fig. 4. Experimental setup comprising Kobuki robot and capacitive sensor.

Fig. 5. Visualization of the sensor data in RVIZ. The blue dots depict the distance between the robot an object in front of the robot.

VI. CONCLUSIONS

In this paper we presented a software interface for capacitive sensors in ROS, supporting both single and multi-modal sensor architectures. In the presented experiment, a threshold based detection algorithm is implemented to detect humans in the surroundings of the robot. In addition the ROS node Process Sensor Data can interface with a more advanced reconstruction and object detection algorithm with little effort. The presented capacitive sensor interface facilitates the applicability of capacitive sensors in the field of robotics.

REFERENCES

[1] B. Mayton, L. LeGrand, and J. Smith, "An electric field pretouch system for grasping and co-manipulation," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 831–838. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5509658

[2] T. Schlegl, T. Kroger, A. Gaschler, O. Khatib, and H. Zangl, "Virtual whiskers: Highly responsive robot collision avoidance," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 5373–5379.

[3] T. Schlegl, M. Neumayer, S. Mühlbacher-Karrer, and H. Zangl, "A pretouch sensing system for a robot grasper using magnetic and capacitive sensors," *Instrumentation and Measurement, IEEE Transactions on*, vol. 62, no. 5, pp. 1299–1307, May 2013.

[4] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[5] L. Baxter, *Capacitive Sensors, Design and Applications*. IEEE Press, 1997.

Development of the Robotic Arm for grabbing a bottle and pouring liquid in to a cup, controlled using LabVIEW*

Sudeep Sharan, Ketul Patel, Robert Michalik, Julian Umansky, Peter Nauth

Keywords — Robotics, Inverse Kinematics, Object Handling, Actuators, Robot Vision, LabVIEW.

EXTENDED ABSTRACT

This work aimed to provide assistance for a daily need to growing number of elderly people, for example pouring liquid into another container. Current statistics shows that by 2050 the number of people aged over 65 would be tripled [1]. Sending elderly people to nursing homes or hiring nursing assistant to serve them at home results into unsocial and depressive stage, such activities are expensive. In order to offset these activities, it is better to have an interactive assistive technology at home. There are many robotic technologies available for providing care to elderly people but some technical issues are still unsolved [2]. This paper focuses on the development of a robotic arm for grabbing a bottle and pouring liquid in to a cup. In any assistive robots the robotic arm is a very important aspect that needs to be taken care wisely.

In the Laboratory for Autonomous Systems and Intelligent Sensors at the "Frankfurt University of Applied Sciences", Frankfurt am Main, Germany, the first Human Assistive intelligent robot called "Roswitha" with different kind of sensors and actuators is being developed "Fig. 1" [3]. Roswitha is a robot to operate autonomously, is developing with a set of three major systems like navigation, vision and the robotic arm. The navigation system contains a laser scanner and two encoders for the movement. Basically it detects the initial position by scanning a mark in a particular area of the lab, which calculates the initial position and based on a desired point it makes its own path on a predefined room map. For a vision system, a normal digital camera is used to detect an object like bottle for a robotic arm to perform its task. The robotic arm consists of different actuators to perform the task.

Many robotic arms from different manufactures are already available, but it was our goal to develop a cost-effective robotic arm which can be easily maintained and modified within the lab for the future research.

*Research supported by Frankfurt University of Applied Sciences.

Sudeep Sharan is with Frankfurt University of Applied Sciences, 60318 Frankfurt am Main, Germany (e-mail: sss.fhffm@gmail.com).

Ketul Patel is with Frankfurt University of Applied Sciences, 60318 Frankfurt am Main, Germany (e-mail: patelketul87@gmail.com).

R. Michalik & J. Umansky are with Frankfurt University of Applied Sciences, 60318 Frankfurt am Main, Germany.

Peter Nauth is Professor at Frankfurt University of Applied Sciences, 60318 Frankfurt am Main, Germany (e-mail: pnauth@fb2.fra-uas.de).

Figure 1. Assistive Robot "Roswitha"

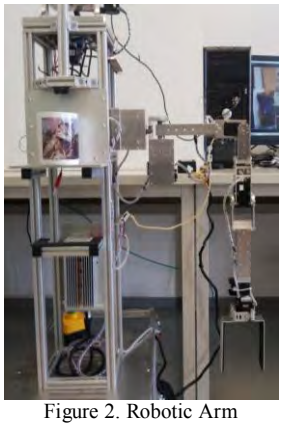


Figure 2. Robotic Arm

The 5 DOF Robotic arm with a gripper has been entirely developed under the roof of "Frankfurt University of Applied Sciences" "Fig. 2". For the communication RS485 protocol has been used between the servo motors and the computer. The work approached LabVIEW based control of the robot arm.

A. Mathematical Equation

A program in the LabVIEW gets the coordinate of the bottle in a real time environment using a vision system and this real world coordinates are being send to an inverse kinematics program. An inverse kinematics equation has been developed for joints J3 and J5 "Fig. 3" to calculate the angles of that motors. The inverse kinematics equation is derived using the Denavit-Hartenberg transformation.

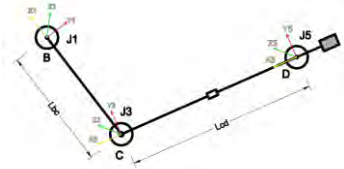


Figure 3. Simplified system geometry

The final DH Matrix is

$$\mathrm{Rot}\,(x_n,\alpha_n) = \begin{pmatrix} \mathrm{Cos}\theta_n & -\mathrm{Sin}\theta_n\mathrm{Cos}\alpha_n & \mathrm{Sin}\theta_n\mathrm{Sin}\alpha_n & a_n\mathrm{Cos}\theta_n \\ \mathrm{Sin}\theta_n & \mathrm{Cos}_{\ n}\mathrm{Cos}_{\ n} & -\mathrm{Cos}\theta_n\mathrm{Sin}\alpha_n & a_n\mathrm{Sin}\theta_n \\ 0 & \mathrm{Sin}\alpha_n & \mathrm{Cos}\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

After further solving the DH Matrix we finally get the equation as;

$$\mathrm{Cos}\theta_{30} = \frac{z^2 + x^2 - Lbc^2 - Lcd^2}{2\,Lbc.Lcd} \quad (2)$$

$$\theta_{30} = Atan2\ (\pm\ \sqrt{\left(1 - \left(\frac{z^2 + x^2 - Lbc^2 - Lcd^2}{2\ Lbc.Lcd}\right)^2\right)}, \frac{z^2 + x^2 - Lbc^2 - Lcd^2}{2\ Lbc.Lcd} \quad (3)$$

$$\theta_{50} = Atan2\ (z,x)\ \pm\ Atan2\ (\sqrt{(z^2 + x^2 - (Lbc * \mathrm{Cos}\theta_{30} + Lcd)^2)}, Lbc * \mathrm{Cos}\theta_{30} + Lcd) \quad (4)$$

θ_{50} is the angle of the joint J5 for reaching the different points
θ_{30} is the angle of the joint J3 for reaching the different points
z is the coordinate in the Z direction
x is the coordinate in the X direction
Lbc is the length between joint B & C
Lcd is the length between joint C & D

B. Software

The typical work flow of the robotic arm has been divided in several steps which help to make a modular programming for LabVIEW. "Fig. 4" shows the block diagram of the steps performed for the robotic arm to execute the task. The program has been developed using the state machine method and divided in several cases to ease understanding and quick changes during additional development process.

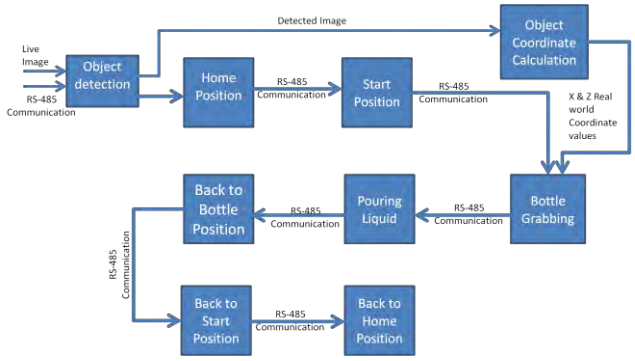


Figure 4. Block Diagram of the robotic arm's work flow

- Object Detection: Manual, automatic and histogram thresholding techniques are used for thresholding an image. Certain geometric and color features are extracted from the image and from the extracted features the object is detected.
- Home & Start Postion: Robotic arm reaches to home position and than start postion with some fixed angles provided from the program.
- Object Cocordinate Calculation: After getting the pixel values of the detected object the real world coordinates is being calculated using mathematical aprroach.
- Bottle Grabbing & Pouring liquid: Inverse kinematics has been used to calculate an angle values for joints J3 and J5 "Fig. 3" which leads an arm to reach to the bottle and pouring position and vice verca.

C. Conclusion

LabVIEW controlled robotic arm was successfully designed and developed. The developed arm was found user friendly and a good example of system integration.

REFERENCES

[1] Suzman R, Beard J (2011) Global health and aging. National Institute on Aging http://www.nia.nih.gov/research/publication/global-health-and-aging/preface. Accessed 26 Feb 2015

[2] Hayley Robinson · Bruce MacDonald · Elizabeth Broadbent The Role of Healthcare Robots for Older People at Home:A Review. Int J of Soc Robotics (2014) 6:575–591 DOI 10.1007/s12369-014-0242-2

[3] Peter Nauth (2009) Goal Understanding and Achievement for Humanoid Assistive Robots , Intelligent Interactive Assistance and Mobile Multimedia Computing Communications in Computer and Information Science Volume 53, 2009, pp 287-294

[4] Ketul Patel, Master Thesis (2014): "Development and control of the robot arm by fabricating electronic hardware and control by using LabVIEW program". Frankfurt university of Applied sciences, Frankfurt am main, Germany

Performance Evaluation of a Cognitive Robotic System

Sharath Akkaladevi, Christian Eitzinger

***Abstract*— we present an evaluation process of the DARWIN (Dextrous Assembler Robot Working with embodied INtelligence) cognitive system with an industrial assembly task. A set of demonstration scenarios was defined which included a number of variations of the assembly task to highlight specific properties of the system and also to test the capability of the systems to recover from various failures. The idea is to approach the evaluation of the systems in a 'black-box manner' to help characterize the performance comparison in an objective way.**

I. INTRODUCTION

Conventional industrial robotic applications are designed to execute clearly predefined tasks and cannot react to dynamic changes in their environment. Moreover, the applications developed are very often hardware specific and therefore cannot be applied to other robotic platforms. The DARWIN robotic system consisting of goal directed reasoning capabilities aims to answer these challenges. It provides an unique opportunity in this direction to "reenact" and "mirror" the gradual process of infant developmental learning and investigate deeper into the underlying "interplay" between multiple fundamental sensory, motor, cognitive processes from the perspective of an "integrated system" (that perceives, acts, learns, remembers, forgets and reasons) [1].

To manifest the statistical significance of the results of scientific work, three cornerstones are mentioned in the literature [2]: a) clear and complete specification of reproducible test setups, b) definition of reasonable criteria to assess test results and c) proper, well controlled execution and evaluation of the tests. This process aids in high-level of transparency and also forms the basis of objective comparison of the achieved results.

With this background, the process of evaluating the performance of the cognitive system is defined as follows: a set of demonstration scenarios is defined that includes a number of variations of the assembly task to better evaluate the functionality of the cognitive system. The assembly task involves inserting 'fuse' like objects into 'fuse boxes'. These demonstration scenarios also included dynamic changes in the environment of the robot, to test the ability of the system in recovering from various failures. As a result, the system is confronted with tasks that are both challenging and representative for a certain range of applications and an objective evaluation of the performance is obtained. The goal of this work is to present an evaluation process that aids in the performance comparison of the cognitive system in an objective way.

All the Authors are with Profactor GmbH, a research company on Production technology (Production-Planning and –Automation) located in Im Stadtgut A2, Steyr Gleink, 4407 Austria. Contact (phone: +43(0)7252 885-120; fax: +43(0)7252 885-101; Corresponding authors' e-mail: firstname.lastname@profactor.at).

II. DARWIN COGNITIVE SYSTEM

The execution of assembly tasks in the cognitive architecture can be viewed as a combination of three layers that involve:

- Localizing objects of interest
- Reasoning and plan generation
- Plan execution

A. Perception

The first step is carried out by the vision module that uses a low-cost RGB-D vision sensor. The vision module [3][4] provides a snapshot that localizes the objects of interest in the workspace. For the given objects in the demonstrations, the vision system also defines certain properties for an object which determine the operations that can be performed on the object. For example, a "grasp cylinder" makes an object graspable, and an "insert tip" makes it insert-able.

B. Cognition

The cognitive capabilities in the architecture result from three core modules (i.e. Neural PMP [7], Observer [8], and Episodic memory [9]) that provides complementary functionalities enabling the system to both reason and generate diverse goal directed plans based on context/ present environment and at the same time extend its knowledge by learning (actions, consequences of actions, new assemblies). Fig. 1 shows the overall functional organization of the cognitive architecture implemented by means of three central subsystems (namely, Observer, Episodic memory and Neural PMP) that provide different complementary functionalities (summarized inside the boxes) and their interface with other modules.

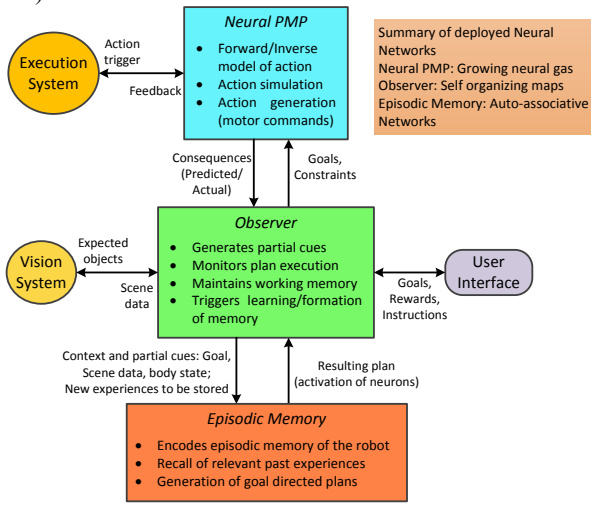


Figure 1: Functional organization of the cognitive architecture [7][8][9]

C. *Execution*

The final layer in the cognitive architecture provides interfaces for the cognitive modules to control the robot (Stäubli [5]) movement and send grasp triggers. Active feedback about the current position of the robots (TX90, RX130) and the current state of the gripper [6] are sent to the cognitive modules during execution of the task.

III. DEMONSTRATION SCENARIOS

The main task is to assemble a fuse box by inserting a number of fuses into the box. During the demonstration two different sets of objects (labeled "type A" and "type B") are considered to show the flexibility of the demonstrators. The assembly task is carried out in an industrial setting using the objects (fuse and fuse box) as shown in Fig 2.

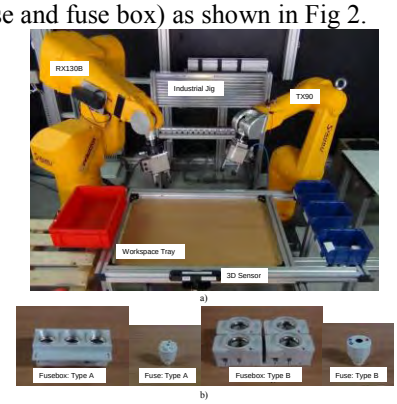


Figure 2: a) The industrial setup with its constituent workspace and b) the objects used for the assembly task

The assembly task is performed in varying object constellations, where each variation is called a demonstration scenario (DS, simply called a demonstration). The demonstration scenarios are defined such that they consist of a clear and reproducible specification of the robot's task and the full context on the one hand, and a number of figures which objectively characterize the test results on the other hand [2]. For each DS a demonstration specific success rate (DSSR) that gives the measure of success of the system in dealing with the variation of the task (of that DS) is calculated. Also, the overall task success rate (TSR) that determines a binary success rate (that defines whether the task was completed successfully or not) of that DS is calculated.

Eight variations of the assembly task (8 DS's) are used to evaluate the flexibility and robustness of the system. They are given as follows:

Basic assembly (DS 1.1): The basic assembly consists of inserting 3 fuses (upright) into a fuse box, where more than 3 fuses are present in the workspace. The DSSR chosen for this task is the time taken in completing the task in seconds.

Lying fuses (DS 1.2): 4 fuses in arbitrary positions (upright, lying) and 1 fuse box are presented. The DSSR is the measures the success of the system in inserting lying fuses.

Bi-manual assembly (DS 2): In this task 8 fuses and 2 fuse-boxes are present. The DSSR is the ability of the system to avoid collisions between robots while operating two robots in cooperation to complete the assembly task.

Recovery from grasp fail (DS 3.1): In this assembly task, artificially introduced failures occur while grasping the fuses, creating a grasp fail. The DSSR measures the success rate of the system in recovering from such grasp failures.

Recovery from insertion fail (DS 3.2): In this assembly task, artificially introduced failures occur while inserting the fuses into the fuse box, creating an insertion fail. The DSSR measures the success rate of the system in recovering from such insertion failures.

Robot Cooperation (DS 4): The fuses will be placed close only to one robot and the fuse box close to the other. The DSSR gives a measure of success that determines cooperation between two robots to solve a task that cannot be completed by a single robot.

Changeover to new task (DS 5): The assembly task will be completed with a new set of different objects (type B). The DSSR gives the time taken to change from assembling one set of objects to another one, assuming the vision system is able to provide required information about the new objects introduced.

Mixed assembly (DS 6): The robot is provided with two different sets ('type A', 'type B') of objects for the assembly task. The DSSR measures the ability of the system in completing two separate assembly tasks.

IV. RESULTS AND CONCLUSION

Each demonstration scenario was performed 20 times, so in total about 160 (20 x 8) experiments were performed on the cognitive system. The average success rate (in %) of each DS over 20 experiments is as given in Tab 1.

TABLE I. PERFORMANCE EVALUATION OF THE COGNITIVE SYSTEM

DS	DSSR	TSR
1.1	31 seconds	95%
1.2	90%	80%
2	100%	75%
3.1	100%	95%
3.2	94%	90%
4	88%	85%
5	15 minutes	85%
6	97%	85%
DS : Demonstration Scenario TSR: task success rate DSSR: Demonstration-specific success rate		

The cognitive system has substantial advantages in terms of flexibility. It showed a DSSR of above 88% in all the cases (measurable), which depicts the cognitive system's robustness and flexibility. During the DS 3.1 and 3.2 which dealt with artificially created grasp failure and insertion failure, the cognitive system reacted accordingly. The cognitive system aborted the insertion task and re-planned as soon as it received a grasp failure during DS 3.1. During DS 3.2, when the cognitive system received a reply from vision of an insertion failure at a given hole, the system re-planned and inserted a suitable fuse in the corresponding hole. The overall DSSR for DS 3.2 was 94% as the vision system mistook a failed insertion as a successful insertion in two cases. Also, the cognitive system required only 15 minutes to be taught a new plan for dealing with a new set of objects. The previous plans and the newly learned plan were merged and DS 6 was completed with a DSSR of 97%. The lower rate of TSR in some cases does not directly reflect the reasoning and planning capability of the cognitive system, as the TSR only considers if the task on whole was completed successfully or not in a binary fashion.

ACKNOWLEDGMENT

This work was co-funded by project DARWIN (Grant Agreement no: 270138 FP7-ICT-2009-6).

REFERENCES

[1] Mohan, Vishwanathan, Giulio Sandini, and Pietro Morasso. "On the interplay between learning, memory, prospection and abstraction in cumulatively learning baby humanoids." *IEEE Trans. Neural Networks* (IJCNN), pp. 1-10, 2013.

[2] I. Iossifidis, G. Lawitzky, S. Knoop, and R. Zllner, "Towards Benchmarking of Domestic Robotic Assistants," vol. 14. Springer Press, 2004, pp. 403–414.

[3] Hongping Cai, Tomas Werner, Jiri Matas. Fast Detection of Multiple Textureless 3-D Objects. Intl. Conf. on Computer Vision Systems (ICVS), 2013.

[4] M. Muja, R.B. Rusu, G. Bradski, D.G. Lowe. REIN - A fast, robust, scalable REcognition INfrastructure, ICRA 2011.

[5] STÄUBLI Robotics; [Online]: http://www.staubli.com/en/robotics/, last access: 04.03.2015

[6] SCHUNK: Servo Electric 2-Finger-Parallel Gripper Type PG 70 Assembly and Operating Manual; [Online]: http://www.schunk.com/schunk_files/attachments/OM_AU_PG__EN.pdf, last access: 04.03.2015

[7] A. Bhat, S. Akkaladevi, V. Mohan, C. Eitzinger, P. Morasso. (in preparation). "Towards a learnt neural Body schema for dexterous coordination of action in Humanoid and Industrial robots." Journal of Autonomous Robots.

[8] Mohan, Vishwanathan, et al. "From object-action to property-action: Learning causally dominant properties through cumulative explorative interactions." Biologically Inspired Cognitive Architectures 10 (2014): 42-50.

[9] Mohan, Vishwanathan, Giulio Sandini, and Pietro Morasso. "A Neural framework for organization and flexible utilization of episodic memory in cumulatively learning baby humanoids." (2014).

Action Recognition for Industrial Applications using Depth Sensors

Sharath Akkaladevi, Christoph Heindl, Alfred Angerer, Juergen Minichberger

***Abstract*— human action recognition plays a vital role in the field of human-robot interaction and is widely researched for its potential applications. In this paper we propose a human action recognition framework using 3D depth data for human-robot interaction in industrial applications. The approach learns a set of key descriptors from a collection of weak spatio-temporal skeletal joint descriptors using random forests (RFs), which reduces the dimensionality and computational effort. The key descriptors are then used with a multi-label random forest classifier for action classification that enables recognition of multiple actions in a given time instant. The resulting low latency, flexible and re-configurable method performs on par with other sophisticated approaches on challenging benchmarks like the MSR Action 3D dataset.**

I. INTRODUCTION

In recent years, the concept of robots cooperating with humans has gained a lot of interest, in both domestic and industrial areas. In industrial environments the combination of cognitive capabilities of humans with the physical strength and efficiency of the robots/machines can essentially reduce the amount of fixed production costs in relation to variable costs [2]. In this context, for the robot to assist the human operator for a given task involves understanding the actions performed by the human, interpreting the activity and eventually interacting with the human. Understanding the action performed by humans is a challenging task, as human actions are complex. The same action or activity is performed differently by different people and also the same person performs the same action differently over time [1].

Over the last few decades, human action recognition is extensively studied due to its potential application fields. Most of these approaches made use of color images or video [3], but one disadvantage of using 2D images is that they only capture the projection of the 3D world into two dimensions and are sensitive to illumination changes. With wide availability of depth sensing devices like KINECT and their low cost, many researchers and practitioners in computer science, electronic engineering, and robotics are leveraging the sensing technology to develop creative new ways to interact with machines and to perform other tasks [4]. One of the major advantages of using depth data is the alleviation of the difficulty mentioned with the 2D images where depth data has good invariance against illumination changes and provides a reliable 3D structure of the scene, especially in indoor environments.

All the Authors are with Profactor GmbH, a research company on Production technology (Production-Planning and –Automation) located in Im Stadtgut A2, Steyr Gleink, 4407 Austria. Contact (phone: +43(0)7252 885-120; fax: +43(0)7252 885-101; Corresponding authors' e-mail: firstname.lastname@profactor.at).

Related Work: In literature [1][10] different methods are described which use depth data for action classification. These methods either use depth-map data [6][7] or use the real-time tracking of 3D skeleton similar to [5] to extract skeleton joints [8][9] for classification of arbitrary actions. Most of the approaches, however, deal with recognizing human actions in a domestic scenario and very few of them consider industrial environment as use cases [11][12]. In the context of interaction between human and robots in an industrial setup there are several key criteria that need consideration: requirement of quick reaction and low latency, re-configurability and high flexibility. Also, the majority of the approaches for human action recognition limit the feature search space with a criterion specific to a scenario [8][11]. Moreover, in industrial applications, one cannot assume the occurrence of segmented actions [11][13] and it is very common for multiple actions to occur simultaneously.

Our main focus in this paper is to deal with the problem of human action recognition for human-robot interaction in an industrial setting using depth sensors. The contributions of this work are 1) a descriptor calculator that is not limited to a specific scenario 2) a framework that uses a multi-label classifier to enable the system to classify presence of more than one action at a time 3) an approach that does not depend on segmented data for action classification.

II. PROPOSED FRAMEWORK

Our proposed framework for human action classification is depicted in Fig. 1. We utilize a sliding window approach to calculate multiple weak spatio-temporal skeletal descriptors stacked into a single aggregate descriptor. To reduce the dimensionality of the initial descriptors we calculate a set of key descriptors. To complete the training we train a random forest based multi-label classifier using the key descriptors. During prediction stage we calculate only key descriptors for each sliding window and use the multi-label classifier for action classification.

A. Training Phase

Our system uses the 3D skeleton joints extracted from RGB-D data [5]. The term *skeletal descriptor* refers to any computation of joint constellations within one frame (spatio) or over multiple frames within a sliding window position (temporal). Instead of choosing skeletal descriptors specifically for each classification task, we provide a set of standard weak skeletal descriptor calculators. The intention is that classification based on a combination of weak descriptors can be used to perform action independent recognition with comparable performance to task specific feature design.

In an environment where low-level primitive actions are to be classified, one has to cope with the fact that multiple actions can be executed in parallel (e.g. waving your hand

while walking). Instead of training a multi-class classifier, we train multiple binary one-versus-all classifiers [14]. That is, for each action, we divide the training samples in two classes. The positive class contains all the training samples where the target action is executed and the negative class consists of all samples lacking the target action. Our framework utilizes random forests [15] as a binary classifier engine. RFs are based on multiple independent decision trees. One of the key properties of decision trees is the ability to compute the relative decision power for individual descriptor dimensions which aids in formulation of our dimensionality reduction method. Using the reduced descriptors (*key descriptors*) we re-train our binary classifiers. Note that each binary classifier might have a different set of reduced descriptors that are important for the individual action being trained.

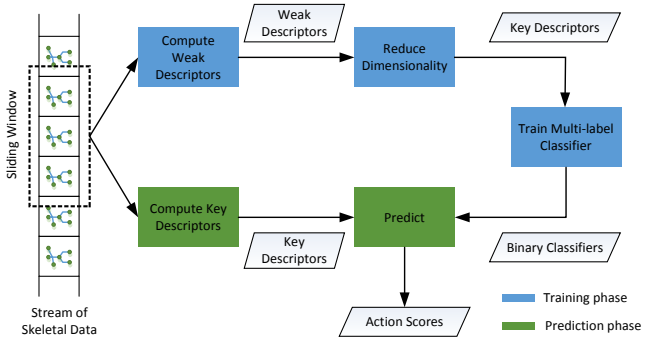


Figure 1: The proposed framework for human action recognition

B. *Prediction Phase*

Similar to training, our framework uses a sliding window approach for performing predictions. For each position of the sliding window in the stream of frames, only key descriptors are calculated for each binary one-versus-all classifier. Prediction scores for action labels are collected and normalized to one. Our system is designed to work in non-segmented situations and in order to cope with faulty predictions but still be able to report actions on a low latency basis we employ the following method:

Prediction in un-segmented streams: In a low-latency prediction scenario one deals with un-segmented streams of data. Within such streams no beginning and end for a specific action is known. To predict active labels within such a stream, we keep track of best labels / scores in a moving window like fashion. This window is separate and slightly larger than the sliding window used for calculating the descriptors. At each point in time we determine the upper 25th percentile of scores and report all labels within that range.

Prediction in segmented streams: For segmented streams we set our prediction window size to half of the number of frames in the video segment. For the most dominant action within a segmented sequence (as shown in the experiments) we store all prediction results and report the one with the maximum vote count.

III. Experimental Results

We compare our proposed approach with state of the art approaches using the MSR Action3D dataset [7] under different experimental configurations.

A. *Experimental Setup*

The MSR Action3D dataset [7] contains 20 different actions, performed by 10 different subjects with up to 3 different executions. From each of the 567 sequences the 20 joint positions per frame where used as inputs. Following the approach of [7][16] three tests are performed on the data set, TestOne, TestTwo and TestCross. In TestOne one-third of the executions are used for training and two-thirds for testing. In TestTwo two-thirds of the executions are used for training and one third for testing. In TestCross, half of the subjects i.e., (1,3,5,7,9) are used for training and subjects (2,4,6,8,10) are used for testing.

B. *Evaluation*

Tab. I shows the overall recognition accuracies of our proposed framework. While overall results for TestOne and TestTwo show high accuracies, the performance in the cross-subject testing varies. The lower accuracy rate for TestCross is explained by the fact that TestOne and TestTwo are trained on all subjects, whereas TestCross is trained on half of the subjects. Since subjects tend to carry out actions similarly every time, chances are higher that a classifier is able to capture the structure of such an action, compared to unseen actions of completely different subjects.

TABLE I. OVERALL RECOGNITION ACCURACY OF THE PROPOSED FRAMEWORK TESTED ON THE MSR ACTION3D DATASET

	TestOne	**TestTwo**	**TestCross**
Average	88.3%	93.4%	75.3%

Tab. II compares the overall accuracy of our proposed approach with state-of-the-art algorithms. Although our approach is intended to classify multiple actions simultaneously in a low-latency fashion, it compares well to state-of-the art algorithms.

TABLE II. COMAPRISION WITH STATE OF THE ART (TESTCROSS)

Method	**Accuracy**
Canonical Poses [18]	65,7%
Action Graph on Bag of 3D Points [7]	74,7%
Latent-Dynamic CRF [17]+ MP	74,9%
Multi-Label Random Forests (Ours)	**75,3%**
EigenJoints [13]	81,4%
Actionlet Ensemble [16]	88,2%

IV. Conclusion

We have presented a multi-label human action recognition framework that is capable of detecting multiple actions simultaneously in real-time. The approach reduces the dimensionality space of feature calculation which aids in low latency and computational efforts. Although our system is designed to work in non-segmented situations, we have shown that it performs on par with other state-of-the art approaches on the MSR Action 3D dataset, which assume segmented videos for action classification.

Acknowledgment

This research is co-funded in part by the Austrian Ministry for Transport, Innovation and Technology via the project

KOMOPROD, InstructMe (PdZ-843693) and by NexGen_RWP (FFG - 840214).

REFERENCES

[1] Ye, Mao, et al. "A survey on human motion analysis from depth data." Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications. Springer Berlin Heidelberg, 2013. 149-187.

[2] Krüger, J., et al. "Intelligent assist systems for flexible assembly." CIRP Annals-Manufacturing Technology 55.1 (2006): 29-32

[3] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea, „Machine recognition of human activities: A survey. Circuits and Systems for Video Technology“, IEEE Transactions on, 18(11):1473–1488, 2008.

[4] Zhang, Zhengyou. "Microsoft kinect sensor and its effect." MultiMedia, IEEE 19.2 (2012): 4-10.

[5] Shotton, Jamie, et al. "Real-time human pose recognition in parts from single depth images." Communications of the ACM 56.1 (2013): 116-124.

[6] Oreifej, Omar, and Zicheng Liu. "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences." Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013.

[7] Li, Wanqing, Zhengyou Zhang, and Zicheng Liu. "Action recognition based on a bag of 3d points." Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. IEEE, 2010.

[8] Xia, Lu, Chia-Chih Chen, and J. K. Aggarwal. "View invariant human action recognition using histograms of 3d joints." Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on. IEEE, 2012.

[9] Sung, Jaeyong, et al. "Unstructured human activity detection from rgbd images." Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012.

[10] Chen, Lulu, Hong Wei, and James Ferryman. "A survey of human motion analysis using depth imagery." Pattern Recognition Letters 34.15 (2013): 1995-2006.

[11] Alina Roitberg, Alexander Perzylo, Nikhil Somani, Manuel Giuliani, Markus Rickert, and Alois Knoll Human Activity Recognition in the Context of Industrial Human-Robot Interaction In Proceedings of AsiaPacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2014), Siem Reap, Cambodia, December 2014

[12] Lenz, Claus, et al. "Human workflow analysis using 3d occupancy grid hand tracking in a human-robot collaboration scenario." Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE, 2011.

[13] Yang, Xiaodong, and YingLi Tian. "Effective 3d action recognition using eigenjoints." Journal of Visual Communication and Image Representation 25.1 (2014): 2-11.

[14] Rifkin, Ryan, and Aldebaro Klautau. "In defense of one-vs-all classification." The Journal of Machine Learning Research 5 (2004): 101-141.

[15] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[16] Wang, Jiang, et al. "Mining actionlet ensemble for action recognition with depth cameras." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[17] Morency, L., Ariadna Quattoni, and Trevor Darrell. "Latent-dynamic discriminative models for continuous gesture recognition." Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007.

[18] Ellis, Chris, et al. "Exploring the trade-off between accuracy and observational latency in action recognition." International Journal of Computer Vision 101.3 (2013): 420-436.

Gesture control for an unmanned ground vehicle

Robert Schwaiger[1] and Mario Grotschar[2]

Abstract— **In order to arouse the public's interest for robotics, a smooth and natural way of human robot interaction (HRI) must be possible. Especially in the area of robotic education and at tradeshows, there is a demand for new human machine interfaces. This paper shows how the taurob tracker's, an unmanned ground vehicle, functionality is extended by gesture control. Different approaches for gesture recognition were explored. Modern solutions for gesture recognition showed promising results, which applies especially to statistical methods for classification of multi attribute data. Because of the many positive results of other research papers this work also makes use of a statistical method for data classification in terms of a support vector machine (SVM). Eventually the LIBSVM library was embedded into the taurob tracker's control software, which builds upon the robot operation system (ROS). There was an emphasis on the software's extensibility, so that new commands can be trained and integrated effortlessly. The recogntion results of these static gesture positions can further be used to train a probability based hidden markov model (HMM), in order to calssify and recognize dynamic gestures. Dynamic gestures are in fact sequences of static gesture positions which can be classified in a sequential manner to a single dynamic gesture movement. The implementation was thested by defining a set of commands, which were then taught by using two different training sets. These were created by two (man and woman) training subjects. Results of this work show that the recognition certainty for the trained gestures reaches up to 70-80%.**

I. INTRODUCTION

Instinctively humans use body language as a natural and efficient way of communication. Enabling a robot to interpret human poses and eventually gestures helps to enhance HRI (Human Robot Interaction) in different scenarios. The main purpose of gesture recognition is identifying a particular human gesture and interpreting its meaning. The area of interest can include the whole corpus or specific parts, such as legs, arms or hands. Application areas which especially benefit from a smooth HRI include USR (Urban search and rescue), service robotics, entertainment, rehabilitation and many more.

II. CHALLENGES AND TASKS

In a preliminary project [1] a simple variant of a gesture recognition approach was presented. This approach was based on the recognition of relative body joint positions of a human body skeleton. These relative joint positions were evaulated conditionally in relation to the body skeleton center joint. In order to extend the applicability of this application the need of a more flexible and intuitive recognition approach emerged. Furthermore, the recognition of dynamic gestures is an important step for improving the usability of the taurob tracker, an unmanned ground vehicle (UGV) used at the UAS Technikum Wien, or a gesture recognition system in general. The redesigned system should be able to recognize gesture positions more precisely and interpret dynamic gestures in terms of their kinematic quantities of the movement, such as the velocity and acceleration of the gesture movement. Including the joint velocities in the model provides a possibility to directly transfer those values to the acceleration of the differential drive. As a result a more intuitive and natural feedback can be given to the end user.

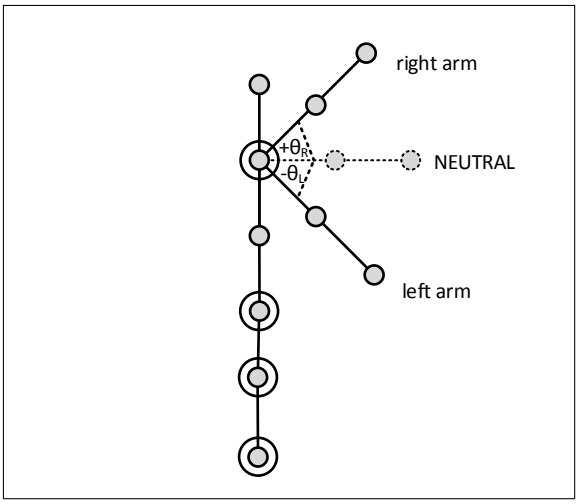


Fig. 1. Side view of the MOVE gesture for left and right arm

Figure 1 represents a side view for the implemented gesture approach. The circled joints for the shoulder, hip, knee and foot parts indicate that there is a occluded joint for each side of the skeleton as provided by the skeleton tracker [2]. Moving both arms into the neutral position results in the robot standing still. Raising an arm towards the head results in a forward motion for the according differential drive wheel. The same principle applies to a downwards rotation of the arm towards the ground - resulting in a backwards motion. Equal arm joint angles on both sides result in a straight forward - or backward motion. This model enables the user to control the robots motion for each wheel individually.

III. MATERIALS AND METHODS

A SVM classifier was used to classify the human body skeleton poses. This project used LIB-SVM [3], an open

[1]Robert Schwaiger is student of the study program Mechatronics/Robotics, UAS Technikum, A-1200 Höchstädtplatz 6, Austria robert.schwaiger@technikum-wien.at

[2]Mario Grotschar is with the Department of Advanced Engineering Technologies, UAS Technikum, A-1200 Höchstädtplatz 6, Austria mario.grotschar@technikum-wien.at

source SVM library written in C++, to solve the given problem. The LIBSVM implementation uses a kernel function $\kappa(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$ where $\gamma = \frac{1}{num\ features}$.
In this formulation, the SVM classification problem is fundamentally a two-class problem [4]. [5] describes the methology as follows. For this pose detection problem where each pose is a data point, the n joints are the feature space, and the k different poses are the classification results, there are more than two classes. A system designed to identify specific poses and not simply determine if there is a pose must go beyond this two-class formulation. To solve this problem, LIBSVM uses a one vs. many approach that generates k SVMs for each pose and then uses a generalized Bradley-Terry model that does pairwise comparisons to determine the SVM that produced the most likely result.

IV. IMPLEMENTATION

Figure 2 illustrates the block diagram for the implemented system which is similar to [6]. The tracked body joint data from the OpenNI tracker is used as an input for a trained SVM model. The model outputs classification results for the specific gesture positions. As the input data is compared with every trained SVM in the model, probabilities for each trained gesture are given. This information is further used to control the differential drive of the robot. The resulting robot velocity is based on the summation of the linear scaled velocities according to the recognized arm joint angles. Equal arm joint angles on both sides result in a straight forward or backward motion. The grayed blocks visualize the HMM part that can be integrated to the system in future work.

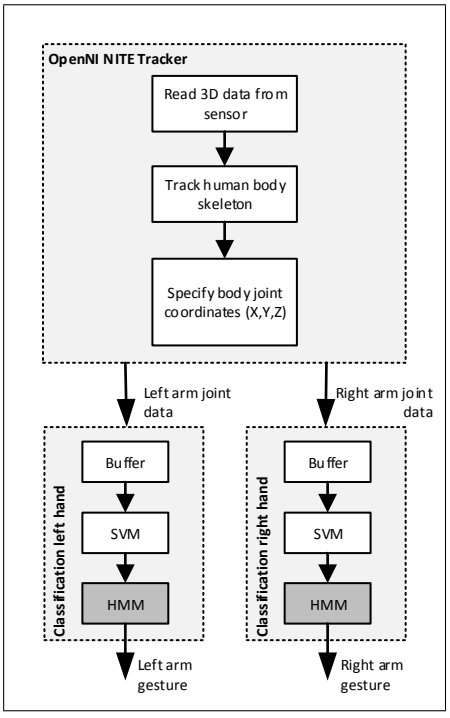


Fig. 2. Block diagram representation of the gesture recognition algorithm

V. RESULTS

The implemented model was tested for functionality. This was done by requesting two persons to engage in the previously trained gesture positions. The classification results for all gestures reached probabilities for each gesture for up to 70-80%. The probability estimate results of the model can be increased through more and especially less noisy training data. Discrete arm joint angle gesture positions in 25° interval steps were trained, resulting in 5 arm angle gesture positions for each side. For these intervals the trained classifier was good enough to produce satisfying results. Considering the 5 gesture states for each of the 2 arms, a command set space of $5^2 = 25$ unique commands can be identified and evaluated by the system.

VI. CONCLUSION AND OUTLOOK

This project focused on extending an existing gesture recognition system for the taurob tracker. Starting from a simple gesture recogntion in the form of conditional relative body joint positions of the human body skeleton, a system based on statistical classification was later on implemented. For the definition of gesture commands, a SVM model was trained with recorded gesture data. This was done in order to be able to distinguish between the different gesture states. The reocognition for those trained arm gestures was implemented specificly for each arm. This enables the operator to control each wheel of the differential drive of the Taurob robot separately. Results show that the recognition system for the pre-trained arm gestures fulfills the need of a robust system using statistical classification. The possibility to train different SVM models provides enough flexibility for future projects; namely that the existing gesture commands on the platform can therefore be extended. Better training data and exact training of the model provides an easy way to also impelment more complex gestures for recognition in the long run. In order to control the acceleration of the robot more precisely, future projects should focus on implementing a HMM model which includes the tracked joint velocities. Building upon this data, user-defined acceleration profiles for the robots driving mechanics can be specified.

REFERENCES

[1] M. Kraupp, "Anbindung einer kinect an eine mobile plattform," Wien: FH Technikum Wien, Masterstudiengang Mechatronik/Robotik, 2013.

[2] Occipital, Inc., "Openni," 2014. [Online]. Available: http://structure.io/openni

[3] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[4] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," *Department of Computer Science*, 2010, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[5] S. M. Chang, "Using gesture recognition to control powerpoint using the microsoft kinect," Massachusetts Institute of Technology, 2013.

[6] M. Sigalas, H. Baltzakis, and P. Trahanias, "Gesture recognition based on arm tracking for human-robot interaction," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct 2010, pp. 5424–5429.

Using Ultrasonic Signals for Pedestrian Detection in Autonomous Vehicles*

Andreas H. Pech, *Member, IEEE* and Peter M. Nauth, *Member, IEEE*

***Abstract*— In this paper we introduce a new method for pedestrian detection to be used in driver assistance systems. It is based on ultrasound as park distance control devices which are available in modern cars. Pedestrian detection is performed by evaluating the signal in the frequency domain and extracting selective features which serve as input of a classification system. The Smart Pedestrian Detection Sensor consists of ultrasonic transducers, a feature extraction unit and a classification system with a database containing the multivariate distribution of the features in the feature space depending on the object distance as a step towards improved pedestrian protection.**

I. Introduction

The detection of pedestrians is one of the most challenging tasks to be solved by autonomous vehicles driving within urban areas. If a car is likely to collide with a pedestrian systems to protect the person must be activated. Contrariwise, in case of a likely accident with another car or a wall systems to protect the driver and passengers need to be activated. Current sensors used in collision avoidance systems just detect obstacles rather than differentiating between pedestrians and other cars or walls.

There are already several developments in existence which serve to detect personal injuries. The majority of developments involve deriving corresponding measures from the impact design. This has the disadvantage of causing injury to the person. Furthermore corresponding measures have to be implemented very quickly (in the microsecond range), which is also suitable to lead to additional injury to the person. There are systems wherein a person is suitable to be detected via the analysis of two visual images, e g. a stereo camera [1]. The analysis of two visual images has the disadvantage that the image becomes unrecognizable via droplets in the case of rain. Furthermore the detection of persons at night or befogged is extremely difficult as well as a differentiation between moving persons and moving objects. This is why a system is described in [2] that calculates the speeds or sizes of objects via radar and camera by means of temporal recording of distances of objects. However this has the disadvantage that a precise differentiation between person and object is not readily possible. Objects are suitable to overlap, thereby resulting in unusual sizes ore movements. Furthermore, objects may take the dimensions of people, so that a differentiation is not suitable to be made. In radar-monitored scenes children cannot be detected due to their size. To overcome these disadvantages of existing systems a new system based on ultrasonic signals is proposed.

*Research supported by Frankfurt University of Applied Sciences.

Andreas H. Pech is Professor at Frankfurt University of Applied Sciences, Computational Intelligence Laboratory, Nibelungenplatz 1, D-60318 Frankfurt, Germany (e-mail: pech@fb2.fra-uas.de).

Peter M. Nauth is Professor at Frankfurt University of Applied Sciences, Laboratory for Autonomous Systems and Intelligent Sensors, Nibelungenplatz 1, D-60318 Frankfurt, Germany (e-mail: pnauth@fb2.fra-uas.de).

II. Using Ultrasonic Signals for Pedestrian Detection

Using ultrasonic transducers at a center frequency of approximately 40 kHz an ultrasonic burst is emitted to the surrounding. According to the theory of wave propagation the ultrasonic wave is reflected as it impinges on acoustic high-density materials [3]. If the surface is abrasive and curved with respect to the wavelength then the wave is not only reflected but also scattered and backscattered. The reflected and backscattered parts of the wave interfere so that the shape and the abrasiveness of the surface are modeled onto the wave. This process is resulting in a signal which not only differs significantly from the original signal that has been sent during the burst but also carries the information about the properties of the reflection surface. Scattering and interference lead to changes in the frequency spectrum which are typical for the reflection surface. As an example the frequency spectrum of a pedestrian is shown in Fig.1 whereas in Fig. 2 a frequency spectrum of a rear bumper of a car is provided.

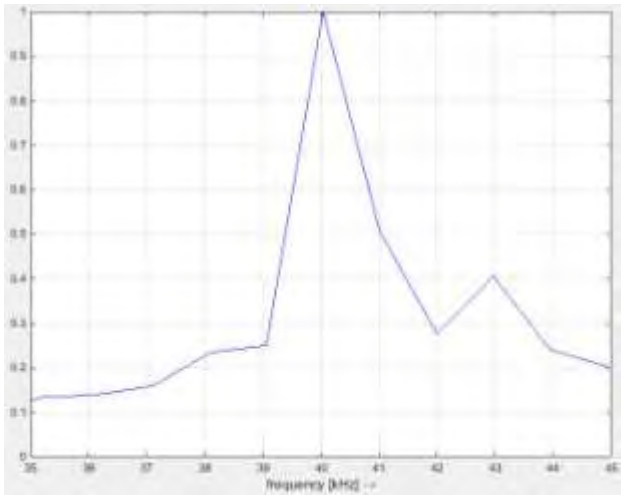


Figure 1. Normalized frequency spectrum of a pedestrian

The fact that the properties of the reflection surface are modeled upon the reflected ultrasonic wave lead to the conclusion that a differentiation of objects of different shapes or objects of different surfaces is possible with a Smart Pedestrian Detection Sensor (SPDS) utilizing the theory described above.

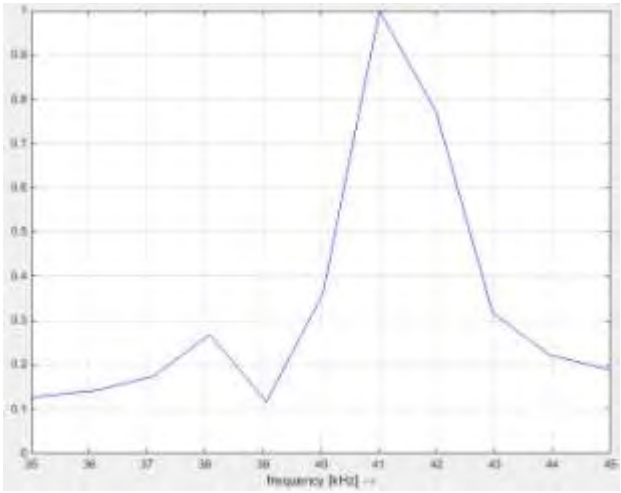


Figure 2. Normalized frequency spectrum of a rear bumper

We investigated ultrasonic waves reflected from different parts of a car and compared them to the reflection from a steel sphere and also to the reflection from a pedestrian. In the following example of the measurement the results shown in Fig. 3 are based on reflections from two different reflection positions at the rear bumper of a car (class 1, middle position, red dots and class 2, edge, black dots) and reflections from a steel sphere (class 3, pink dots) and also reflections from a pedestrian (class 4, green dots).

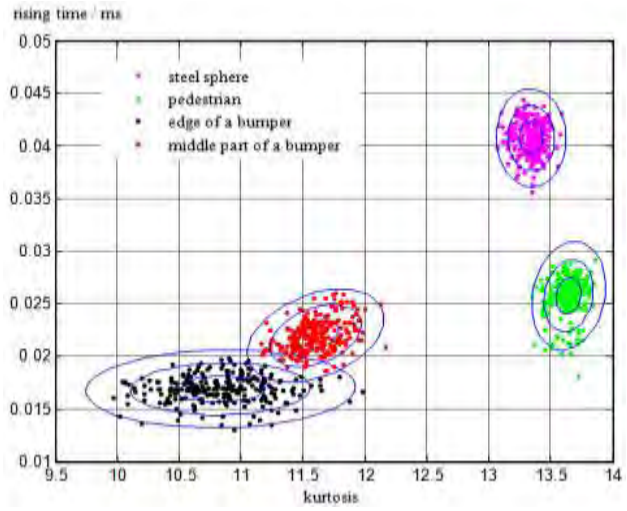


Figure 3. Objects measured at a distance of 40 cm

Each of the objects has been measured 250 times at a distance of 40cm from various angles of incidence. The angle of incidence has been chosen randomly in the range of -10 to 10 degrees. Two features were extracted from the measured data. The first feature is the rising time calculated in the time domain from the envelope of the signal. The second feature is calculated in the frequency domain by treating the distribution of frequencies around the center frequency as a random distribution and calculating the kurtosis of the distributional shape.

III. SMART PEDESTRIAN DETECTION SENSOR

Using ultrasonic transducers at a center frequency of approximately 40 kHz a SPDS demonstrator Fig. 4 has been built to proof this assumption. It is intended to run 10 complete measurement cycles per second so that a range of up to 15m in front of the sensor can be covered. It executes the functions:

- Generation of a 40 kHz ultrasonic signal pulse of 4 cycles with a bandwidth of +/- 1 kHz.
- Acquisition of the backscattered signal with a sampling frequency of 500 kHz.
- Extraction of time- and frequency-domain features from the backscattered signal.
- Classification of the feature vector by means of a Bayes classifier.

Figure 4. SPDS demonstrator

The hardware consists of components as follows:

- A Compact RIO (cRIO, National Instruments) generating a 40 kHz rectangular signal and applying it to an ultrasonic board by means of a NI 9401 module. A NI 9201 module acquires the signals received by the ultrasonic board.
- An ultrasonic board which applies the 40 kHz signal to a transmitter and amplifies the backscattered signal received.
- A PC which performs feature extraction from the acquired signal and classification.

The classification result is displayed on the graphical user interface by indicating either “Human”, “Object” or “Out of detection range” Fig. 5. All functions have been programmed in LabVIEW.

The linear Bayes classifier is trained by samples of humans (class H) and samples of objects (class O) such as car fenders. Different people varying in size, cloths and poses are used for training class H.

First, investigations with 2 time-domain features resulted in a high overlap in feature space regarding the classes. The feature vector now consists of 4 time- and 2 frequency-domain features. Applying SPDS to 10 different persons and 10 different parts of cars so far we yielded detection rates as follows:

Accuracy: 85%
Precision: 82%

Our first results suggest that a reliable object differentiation in general and the detection of pedestrians in particular is possible with the Smart Pedestrian Detection Sensor SPDS. Currently we are verifying this first assessment with a higher number of cases.

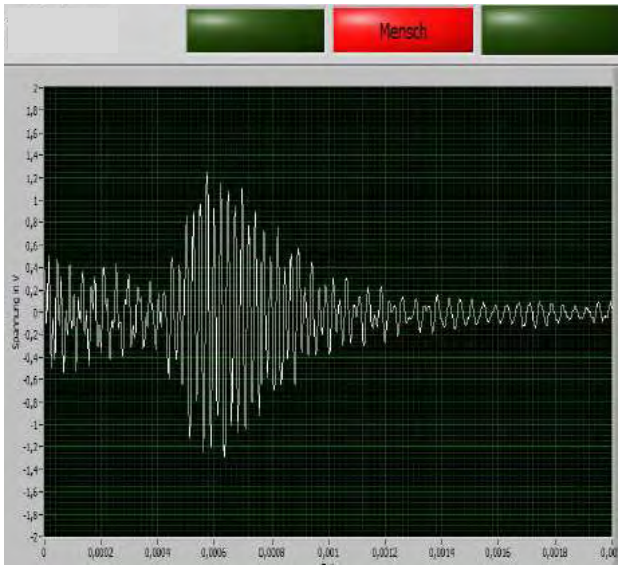


Figure 5. Indication of classification result “Human”

As for now, the full potential of SPDS has not been exploited yet. We expect that an improved transducer, more and advanced features as well as a better classification strategy will increase detection rates and will provide more robustness regarding noise and intra-class variations. Especially the linear classifier limits the performance of our prototype in case of non-linear separable features. The implementation of these measures is planned for the next project phase.

The probability distribution of the features depends on the distance of the objects. Therefore the distance range of the sensor has to be divided into sections. The distributions have to be trained separately for each section. To calculate the correct distance the ambient temperature has to be measured in order to compensate its influence on the speed of sound.

Given that many modern cars have ultrasonic sensors built-in an analysis of these signals with the SPDS approach could provide cars with an affordable pedestrian detection. This is a huge advantage for smaller cars because alternatives such as vision systems are more expensive. As for more expensive cars, a fusion with vision data is an option.

REFERENCES

[1] Tadahiro Igawa, “Objekterkennung und Fussgänger Schutzsystem”, German Patent DE602006001012T2, June 25, 2009.

[2] Ito, Yosuke, Kariya, Aichi, “Personenerkennungsvorrichtung für Fahrzeuge und Zusammenstossminderungsvorrichtung hiermit“, German Patent DE 102009038929, March 11, 2010.

[3] P. M. Morse and K. U. Ingard, *Theoretical Acoustics*, New York, St. Louis, San Francisco, Toronto, London, Sydney: McGraw-Hill, 1968, pp. 436-449.

Processing Tracking Quality Estimations in a Mixed Local-Global Risk Model for Safe Human Robot Collaboration

Frank Dittrich and Heinz Woern

***Abstract*— In this paper we describe a Bayesian Network approach, which is the basis for reasoning about the risk of a user working in a collaborative task with a partially autonomous robot in a shared workspace. For local and global inference, the probabilistic model thereby combines scene information on situation and activity classes of different body parts and the whole user. In order to process local information about the temporal development of single body parts in relation to the robot, we use Hidden Markov Models to predict relative movement classes for each body part. The core element of our approach are the estimates from the real-time human body posture tracking. In order to improve the quality and reliability of the risk estimation we also maintain a locally-resolved posture estimation quality model and inject it into early stages of the inference process.**

I. INTRODUCTION

The here proposed approach is part of a cognitive system for safe human-robot collaboration (SHRC) with applications in the industrial domain (see [1]). High and low-level scene analysis is performed based on the information of multiple RGB-D sensors. Various approaches are used to perform worklfow analysis and to reason about the the user's physical and mental stress. In our collaborative work scenarios, the user and the robot work in a shared workspace and in close proximity, without temporal or spatial separation. Also, because the robot possesses a certain degree of autonomy in the collaboration, the robot's actions are only partially controlled by the user. This setup therefore demands the examination and processing of the user's risk in order to prevent collisions and to optimize the path planning strategies.

In [2] and [3] the authors present a system which reasons about the user's risk and dynamically adapts the robot's trajectory, by processing the risk estimations, in order to prevent collisions. The risk estimation is thereby based on Fuzzy Logic, and the dynamic path planning is an adapted A^* variant. The risk estimation is thereby global and based on low-level information related to the whole body of the human worker.

Bayesian Networks have been around for a long time now, and one major scope of application is risk assessment in all kinds of areas. For the modeling of the risk we therefore use the Bayesian Network framework, which allows us to process low-level body part dependent information together with high-level user dependent information in a joint reasoning scheme. This makes it easy to inject information about certain work steps and stress levels, which contain cues about the user's attention, in the global inference process, together with information about movement types and distances of certain body parts in relation to the robot.

Frank Dittrich and Heinz Woern are with the Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics Lab (IAR - IPR), Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany {frank.dittrich, heinz.woern}@kit.edu

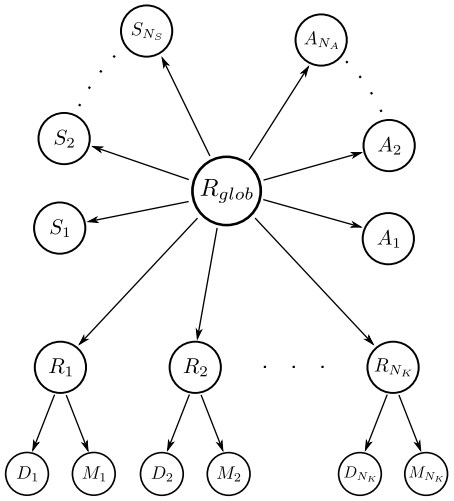


Fig. 1. The Bayesian Network used to combine local and global evidence in one model, and to perform local and global inference about the user's risk. The random variables $S_{1:N_S}$ thereby describe the global *Situation* observations, $A_{1:N_S}$ describe the global *Activity* observations, $D_{1:N_K}$ describe the local *Relative Distance* observations and $M_{1:N_K}$ describe the local *Relative Movement* observations. R_{glob} and $\hat{R}_{1:N_K}$ describe the global and local latent risk states respectively.

In order to address the high temporal and spatial variations in the estimation quality of the posture tracking, we also model the quality in a spatially resolved manner and process the updated information in every inference step.

II. LOCAL-GLOBAL RISK MODEL

To perform spatially and temporally resolved inference about the of the user's risk in real-time, we use a generative probabilistic model. Fig. 1 shows the structure and the various local and global random variables of the Bayesian Network. The tree structure of the model allows for fast inference [4], and the various scalar conditional probabilities are easy to model and determine [5]. For modeling and inference we use the GeNie & SMILE framework[1].

The observed situations and activities in regard to the user, like stress levels and certain work related activities, are conditioned on the global latent risk state, which represents the user's risk. The observed distance and the movement type relative to the robot are conditioned on the correspondent

[1]https://dslpitt.org/genie/

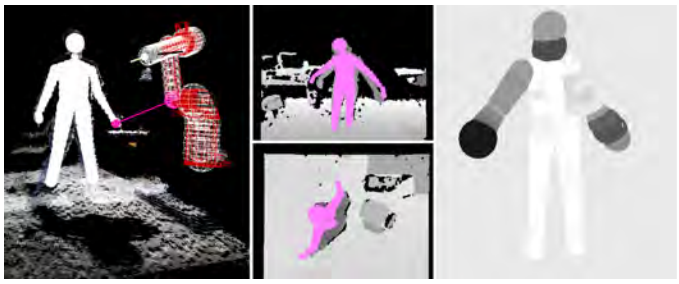


Fig. 2. *Left:* Synthetic sphere representation of the human body, which is parameterized and transformed according to the tracking results. Based on the human and robot sphere models, we perform fast distance calculation. *Center:* Overlay of the depth frames from two synthetic RGB-D sensors and from two real RGB-D sensors. Each synthetic and real sensor pair used for the overlay, has the same transformation in the virtual and real scene respectively. *Right:* Result of the human sphere model adaptation, according to the tracking quality estimations.

local latent risk state, which represents the correspondent body part risk. All local risk states are conditioned on the global risk, which allows the combination of local and global evidence when performing inference about single local risk states or the global risk state.

Because the local evidence is not directly connected to the global risk variable, it is easy to model different weighting in relation to certain body parts when performing inference on the global risk. This allows for instance to prioritize the risk related to the head over the risk related to the legs or hands, which is important when performing robot path optimization.

A. Relative Movement Model

In order to process local movements of the body, we use Hidden Markov Models (HMM) to classify the movement of each body part relative to the robot. This class predictions are then entered as local evidence (Fig.1), which allows to process temporal information when performing inference. Relative movement classes are *No Movement*, *Linear Movement Away*, *Accelerated Movement Away*, *Linear Movement Towards* and *Accelerated Movement Towards*.

III. TRACKING QUALITY PROCESSING

Basis for the local evidence used for risk inference, are the results from the body posture tracking. It is therefore important to evaluate the tracking quality and to process this information.

Both the relative distance and the relative movement prediction are based on the distance estimation between human body parts and the robot. To estimate the distance we use a sphere representation for the user and the robot, which both are parameterized and transformed in a virtual scene (Fig.2, *Left*). In case of the user, the parameterization and transformation is based on the body posture tracking. The calculation of the distance between each sphere pair, is then used to determine the shortest distance for each body part.

In order to estimate the quality we place synthetic RDB-D sensors in the virtual scene, in accordance with the relative the real-world sensors, and create synthetic depth frames of the virtual scene. The overlay and comparison of the synthetic and real-world depth frames (Fig.2, *Center*), for the corresponding synthetic and real-world sensor pairs, is then used to estimate the tracking quality for every sphere of the human body representation.

Based on the per-sphere quality measure, we adapt the human body sphere representation, by rescaling the single spheres in accordance with the quality estimations (Fig.2, *Right*). In case of low quality estimations we expand the sphere, and in case of high quality estimations we shrink the sphere. The adaptation thereby relates to adapting the size of the area, where the position of the certain body part is assumed, to the local tracking quality.

Because the local evidence is based on the distance estimations, which in turn are based on the sphere representation, we then automatically inject the quality estimations in the inference process, by using the adapted model.

IV. APPLICATION

Our intention is to use the information from the risk estimation, to adapt the cooperative system. Especially the robot's path planning has to be adapted, in order to avoid collisions with the human worker, and to produce trajectories which are convenient for the human worker. The latter thereby requires local risk information, in order to produce trajectories which, for instance, avoid the head area of the user. Optimization based approaches for dynamic path planning like [6] or [7], offer thereby a suitable way to process such information.

V. CONCLUSIONS

In this extended abstract, we outline an approach for risk modeling and inference based on local and global information. We also presented our methods to generate and process local information about relative body part movements and tracking quality estimations.

REFERENCES

[1] Frank Dittrich, Stephan Puls and Heinz Wörn, "A Modular Cognitive System for Safe Human Robot Collaboration: Design, Implementation and Evaluation," in *IROS 2013, WS Robotic Assistance Technologies in Industrial Settings*, 2013.

[2] J. Graf and P. Czapiewski and H.Wörn, "Evaluating Risk Estimation Methods and Path Planning for Safe Human-Robot Cooperation," in *Joint 41th International Symposium on Robotics and 6th German Conference on Robotics*, 2010, pp. 579–585.

[3] J. Graf and S. Puls and H.Wörn, "Incorporating Novel Path Planning Method into Cognitive Vision System for Safe Human-Robot Interaction," in *2009 Computation World: Cognitive 2009*, 2009, pp. 443–447.

[4] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[5] U. B. Kjaerulff and A. L. Madsen, *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[6] N. Ratliff, M. Zucker, J. A. D. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2009.

[7] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," *2011 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 4569–4574, 2011.

Local SLAM in Dynamic Environments using Grouped Point-Cloud Tracking. A Framework*

George Todoran[1], Markus Bader[2] and Markus Vincze[1]

***Abstract*— In the domain of mobile robotics, local maps of environments represent a knowledge base for decisions to allow reactive control, preventing collisions while following a global trajectory. Such maps are normally discrete, updated with relatively high frequency and no dynamic information. The proposed framework uses a sparse description of clustered scan points from a laser range scanner. Those features and the system odometry are used to predict the agent ego motion as well as the features motion, similar to a Simultaneous Localization and Mapping (SLAM) algorithm but with low-constraint features. The presented local Simultaneous Localization and Mapping (LSLAM) approach creates a decision base, holding a dynamic description which relaxes the requirement of high update rates. Experimental results demonstrate environment classification and tracking as well as self-pose correction in dynamic and static environments.**

I. INTRODUCTION

Local maps typically consist of close-vicinity representations of the environment. As they represent the closest layer of perception in relation with agent dynamic tasks (path-following, grasping etc.), they are required to be accurate, online and descriptor-rich. The presented framework proposes an alternative to local map-creation and environment description for mobile agents in an online, fast-computing manner. Thus, the environment is modelled and grouped as rigid dynamic objects, treating object discovery, time-out, merging, splitting and symmetries. Using the acquired information regarding the objects dynamic state, agent self-pose correction is performed, enchanting local map-building to local Simultaneous Localization and Mapping (LSLAM). In addition, the framework outputs the classified objects and their dynamic descriptors for further usage in self-localization, mapping, path-planning and other robotics tasks.

Grouping of data in higher level features–objects has been widely studied in computer vision and robotics communities and recently proposed in SLAM approaches [1]. However, this work aims to include high-level features in a more complex SLAM problem, where dynamic entities are present. Dynamic object tracking has been addressed by Montessano [3] in his PhD. thesis, analysing various filtering techniques. MacLachlan [2] presents a segmentation approach for occluded areas based on agent movement, here generalized for concave structures and used within the segmentation module.

*This work was not supported by any organization

[1]George Todoran and Markus Vincze are with the Automation and Control Institute, Vienna University of Technology, Gusshausstr. 27-29/E376, 1040 Vienna, Austria [george.todoran, markus.vincze]@tuwien.ac.at

[2]Markus Bader is with the Institute of Computer Aided Automation, Vienna University of Technology, Treitlstr. 1-3/4. Floor/E183-1, 1040 Vienna, Austria markus.bader@tuwien.ac.at

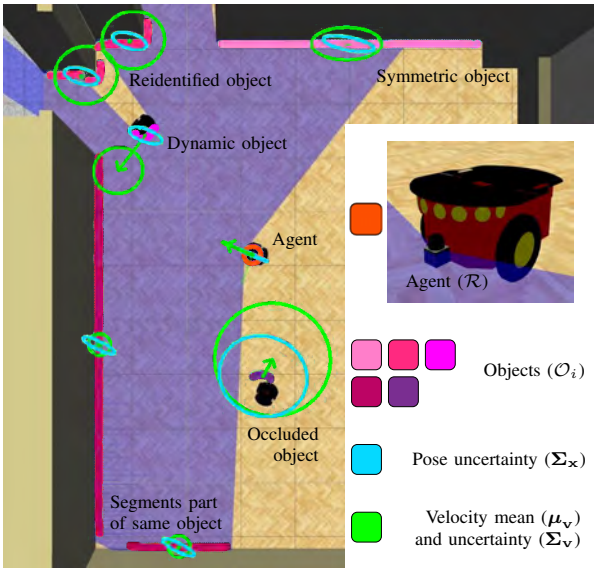


Fig. 1: Generated Local Map

II. APPROACH

As the agent $\mathcal{R}$ moves within an environment, the local environment is classified and mapped. Due to sensor characteristics and locality of the data association problem, the mapped point-clouds are represented in polar coordinates in the sensor frame, propagated in time with respect to the agent motion. However, their clusters are represented within the *LSLAM* estimator in a Cartesian space. Figure 2 presents the overview of the framework, including its modules and data-flow. In the following, the general characteristics and approaches towards each of the modules is presented.

a) Preprocessing: As the sensor input points $\hat{\mathbf{p}}_{raw}$ are modelled with noise in the range measurement, they are filtered in polar space using a continuous Gaussian-kernel. Given such sparse representation, reduced smoothing and eventually neglection of points close to a feature edge is achieved. Moreover, the module clusters the points in segments $\hat{\mathcal{S}}$, bounded by discontinuity regions.

b) Homographic Segmentation: As the entire world is assumed to be dynamic and of interest to the agent, the necessity of segmentation could be questioned. Overall, the purpose of segmentation in this framework is to reduce the number of points without correspondences in the matched point-clouds, increasing the robustness of *Covariant ICP*. Given the agent states $\boldsymbol{\mu}_{\mathcal{R}}$ and $\overline{\boldsymbol{\mu}}_{\mathcal{R}}$ when segments $\bar{\mathcal{S}}$ and $\hat{\mathcal{S}}$ are acquired, points that would not satisfy the ray assumption are segmented using occluded points removal, constant angular re-sampling and outlier removal.

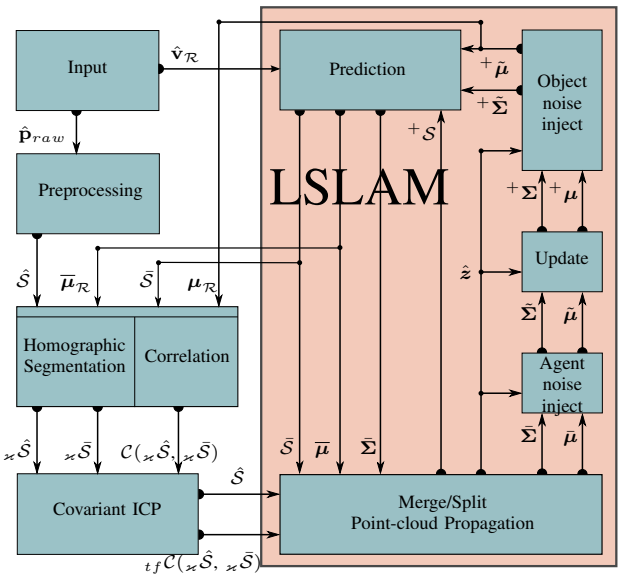


Fig. 2: Framework overview

c) Correlation: Using the same inputs, the module creates a sorted instruction list $\mathcal{C}(_{\varkappa}\hat{\mathcal{S}}, {}_{\varkappa}\bar{\mathcal{S}})$ for the *Covariant ICP* module. Segment correspondence is obtained using a two-way nearest-neighbours approach, refined using search masks according to agent and objects pose uncertainties $\mathbf{\Sigma_x}$.

d) Covariant ICP: The module evaluates the segment sorted pairs list and computes, if found, the according rigid transformations. In comparison with other approaches, the module identifies situations of symmetry (circular arc, perfect line, repetitive patterns) and includes this information by computing transform uncertainty. Thus, false data association is reduced and object merging/splitting can be evaluated.

e) LSLAM: The loop is being closed by an EKF type estimator. The objects are initially modelled as constant-acceleration dynamic with uncorrelated translation and rotation, assuming small values of process noise. However, their estimation convergence is evaluated and additional process noise is added (e.g. in situations of objects undergoing high accelerations). This way, objects that are in steady state (static, constant velocity) have reduced relative uncertainty and thus weight more in agent self-pose correction. In a nutshell, the agent learns the dynamics of the environment when its odometry is accurate. When odometry inconsistency is detected (wheel slips, time delay), noise injection in the agent state will automatically correct its pose in a probabilistic manner, using the most certain landmarks state as reference. However, if all objects are assumed to be dynamic, agent noise injection implies degradation of the entire state estimation, being feasible only for short-term inconsistencies.

III. Experimental Results

Experiments have been conducted using the Gazebo simulation environment under Robotics Operating System (ROS). The agent is a Pioneer-P3DX mobile robot equipped with a Hoyuko laser range sensor. The agent and sensor dynamics and noise are modelled and simulated accordingly: constrained agent velocities ($a_{max} \sim 0.4m/s^2$); Gaussian noise in sensor range ($\sigma_\phi \sim 10^{-2}m$) and agent velocity ($\sigma_v \sim |v| \cdot 10^{-2}m/s$). As loop skips or low frequencies of the filter will only increase the estimation uncertainty, the framework frequency is set to 8Hz (ROS publishes laser and odometry data unsynchronised at 10Hz). For all the experiments, the agent is tele-operated.

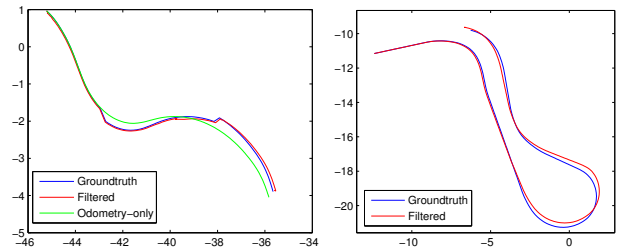


Fig. 3: Self-pose correction in dynamic (*left*) and static (*right*) environments ($m \times m$)

Object correspondences: Even though the framework provides short-term memory differential mapping, higher-level features such as object correspondences are being extracted. Figure 1 illustrates capabilities of the framework to successfully track identified objects in non-trivial scenarios.

Self-pose correction: As described in Section 2, the pose of the agent is expected to be corrected with high degrees of accuracy as long as parts of the environment are in steady state. Figure 3 (*left*) presents the filtered trajectory of the agent in dynamic environments when it undergoes short-term deviations from the motion model. Even though the framework has been designed for agents with relatively accurate odometry and a dynamic word assumption, it can as well be operated assuming a static environment. Figure 3 (*right*) presents the filtered agent trajectory, assuming a static world and feeding into the framework fixed angular and linear velocity readings set to 0 (odometry impairment).

IV. Conclusions

Agent self-pose correction in dynamic environments is still a weakly addressed problem within the robotics community. The presented framework has proven to extract sufficient information from partially steady-state dynamic environments, even though low constraint models of the environment and agent are assumed. Following such a general approach, developments towards static objects detection (e.g. using static map information) should improve the overall estimation and tracking, providing a complete solution for SLAM with dynamic descriptors and potentially broadening the capabilities of robotics tasks that make use of local maps.

References

[1] S. Choudhary, A. J. B. Trevor, H. I. Christensen, and F. Dellaert. SLAM with object discovery, modeling and mapping. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 1018–1025, 2014.

[2] Robert MacLachlan. Tracking of moving objects from a moving vehicle using a scanning laser rangefinder. Technical report, Carnegie Mellon University, 2005.

[3] Luis Montesano. *Detection and tracking of moving objects from a mobile platform. Application to navigation and multi-robot localization*. PhD thesis, Universidad de Zaragoza, 2006.

[4] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press , 2005.

Autonomous coordinated flying for groups of communications UAVs*

Alexandros Giagkos[1,a] and Elio Tuci[1] and Myra S. Wilson[1] and Philip B. Charlesworth[2]

***Abstract*— We describe a study that compares two methods of generating coordinated flying trajectories for a group of unmanned aerial vehicles. The vehicles are required to form and maintain an adaptive aerial network backbone over multiple independent mobile ground-based units. The first method is based on the use of evolutionary algorithms. The second methods looks at this coordination problem as a non-cooperative game and finds solution using a Game Theory approach. The main objective of both methods is to maximise the network coverage, i.e., the number of mobile ground-based units that have access to the network backbone, by utilising the available power for communication efficiently.**

I. Introduction

This paper discusses two different systems that address the challenges of effective coordinated repositioning of a group of unmanned aerial vehicles (UAVs), so as to provide a communication network backbone for any number of mobile ground-based units (GBUs). The common objective is to generate flying solutions (i.e., manoeuvres) that maximise the network coverage. The first system employs evolutionary algorithms (EAs) as the decision mechanism responsible for the generation of the manoeuvres and UAV to GBU allocation. The second approach exploits the strengths of game theory in order to provide coordinated UAV flying.

The results of a quantitative comparison of a 6-hour flying simulation within a dynamic large-scale environment are presented. It is found that both approaches are able to fulfil the GBU coverage objective, with a key difference being depicted in the distribution of the coverage load (i.e., the number of supported GBUs per UAV). The EA-based approach is able to achieve better coverage performance by converging to the optimal throughout the duration of the scenario. This results from the evolution of those sets of manoeuvres at different altitudes, that unevenly distribute the coverage load. The Game Theory-based approach is found to apply frequent and uniform changes to the UAVs' altitudes and, in turn, achieve even distribution of coverage load over the group of UAVs.

II. Methodology

A. Kinematics and network model

A fixed-wing UAV is treated as a point object in three-dimensional space with an associated direction vector. At each time step, the position of an UAV is defined by the latitude, longitude, altitude and heading in a geographic coordination system. The model is designed to allow flying within a pre-defined corridor such that it stays within ceiling and floor heights. No collision avoidance mechanism is currently employed.

Each UAV is equipped with two types of antennae, one omnidirectional to allow communication between UAV-to-UAV and a horn-shaped antennae to communicate directly with the ground. The transmitting power P_t that a UAV requires to feed to its horn-shaped antenna in order to cover a GBU is expressed by a Friis-based equation model found in [1]. For each GBU, the equivalent P_t is subtracted from the total P_{max}, managing the total power allowance for communication.

B. Evolutionary Algorithms and Game Theory decision units

A UAV may either perform a turn circle manoeuvre with a tight bank angle to keep its current position, or implement a manoeuvre generated by the decision making system. In the context of the EA-based system, taking inspiration from the Dubins curves and paths [2], a manoeuvre solution composes a flying trajectory that consists of three segments, which can be either a straight line, a turn left or turn right curve, depending on the given bank angle. The manoeuvres are encoded into individual chromosomes to be used for the evolution. The EAs are free to select the duration for any of the segments as long as the overall trajectory remains equal to one revolution time of a turn circle manoeuvre. Every 3 seconds the EA-based system receives messages carrying the last known positions and directions of all UAVs and GBUs. Fitness proportionate selection, 1-point crossover and mutation operators coupled elitism constitute the EAs. Once they have evolved a new set of manoeuvres, the resulting manoeuvres are broadcast across the UAV group using the network. The EA-based system is designed to perform an evaluation of the fitness of the set of group manoeuvres rather than individuals, thus ensuring that the UAVs are coordinated. The group's fitness score is the sum of the number of the uniquely supported GBUs per UAV (the allocation plan) [1].

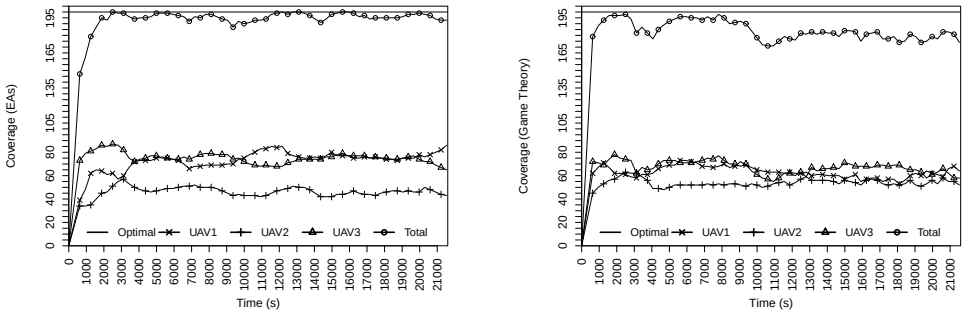
In the Game Theory-based system, the network is also used to broadcast up-to-date positions of both UAVs and GBUs. A game of multiple UAV players is solved in a non-cooperative game fashion, based on a pay-off matrix constituted by the coverage scores of each available manoeuvre strategy. The system assumes that UAV players have complete information regarding the strategies and pay-offs available to all other players. Subsequently, all players solve the same game simultaneously and are expected to reach

*This work is sponsored by EADS Foundation Wales.

[1]Intelligent Robotics Group, Dept. Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK

[2]Technology Coordination Centre 4, Airbus Group Innovations, Celtic Spring, Newport, NP10 8FZ, UK

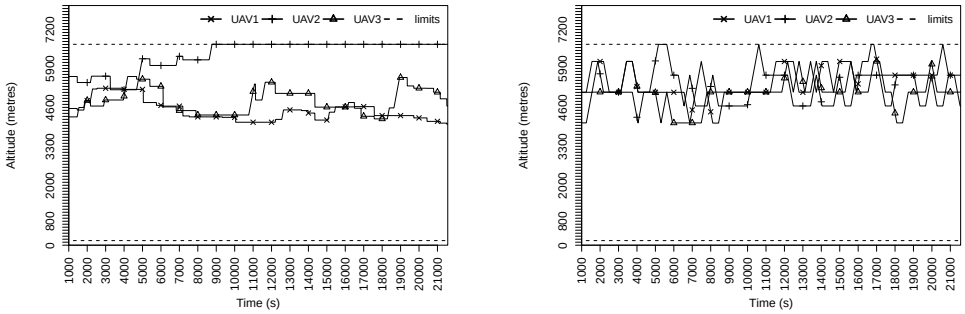
[a]Contact e-mail: alg25@aber.ac.uk



(a) EA-based 3 UAVs

(b) Game Theory-based 3 UAVs

Fig. 1: Coverage results for EA-based (a) and Game Theory-based (b) approaches, supporting 200 GBUs.



(a) EA-based 3 UAVs

(b) Game Theory-based 3 UAVs

Fig. 2: Altitude changes for EA-based (a) and Game Theory-based (b) approaches, supporting 200 GBUs.

similar results, in a deterministic way. Solving the game is achieved by applying the Chatterjee's method [4], so as the Nash equilibrium (pure or mixed strategy) outcome is decoded to appropriate flying manoeuvres for the UAVs. More information about the Game Theory approach can be found in [3]. The pay-offs of the game strategies are calculated by the effectiveness of each UAV to maximise GBU coverage, according to the allocation plan.

III. Results

The results of a quantitative comparison with three UAVs supporting 200 GBUs are presented in figure 1 (a) and (b). [1] The flexibility in implementing flying manoeuvres in the EA-based system ensures that optimal total coverage can be frequently achieved throughout the duration of the experiments. Moreover, the Game Theory-based results are found to be close to optimal. The difference is explained by observing the individual coverage results. The emergent flying behaviour resulting from the group evaluation in the EAs allows the UAVs to perform coordinated heterogeneous manoeuvres, achieving a higher total, whereas in the Game Theory-based the coverage load tends to be balanced between the UAVs. The effect of evolving coordinated heterogeneous manoeuvres is shown in figure 2 (a), where the altitude changes of the three UAVs are depicted. Compared to the Game Theory-based results in figure 2 (b), the EAs tend to increase the altitude of one single UAV and lower the rest of the group, whereas the Game Theory decision unit makes frequent, uniform altitude changes that leads to a balanced distribution of the coverage load.

[1]The results summarise the performance of 20 experimental runs.

IV. Conclusion

Two systems able to autonomously fly groups of communications UAVs are compared. The first employs EAs in order to generate coordinated heterogeneous manoeuvres, whereas the second employs concepts from Game Theory. Quantitative results show that the EA-based is efficient in terms of reaching the optimal for most of the simulation time. In addition, the Game Theory is found to allow balanced coverage load between the UAVs.

References

[1] A. Giagkos, E. Tuci, M. S. Wilson, P. B. Charlesworth, Evolutionary coordination system for fixed-wing communications unmanned aerial vehicles: supplementary online materials, Available at: http://www.aber.ac.uk/en/cs/research/ir/projects/nevocab (April 2014).

[2] L. E. Dubins, On plane curves with curvature., Pacific Journal of Mathematics 11 (2) pp 471–481, 1961.

[3] P. B. Charlesworth, A non-cooperative game to coordinate the coverage of two communications UAVs, in: 2013 - MILCOM 2013 System Perspectives (Track 4), San Diego, USA, 2013.

[4] B., Chatterjee, An optimization formulation to compute Nash equilibrium in finite games, in: International Conference on Methods and Models in Computer Science, 2009 (ICM2CS 2009), pp. 1–5, 14-15 Dec. 2009.

UAV-based Multi-spectral Environmental Monitoring

Thomas Arnold, Martin De Biasio, Andreas Fritz and Raimund Leitner

Abstract— **This paper presents the integration of a multi-spectral imaging system in an UAV (Unmanned Aerial Vehicle) to distinguish different types of vegetation and soil. A miniaturized multi-spectral imaging system was developed to fit into a compact UAV. The presented system utilized four CMOS cameras and was able to capture six spectral bands simultaneously; three visible (400nm-670nm) and three near infrared channels (670nm-1000nm). The actively stabilized camera gimbal was able to compensate the rapid roll/tilt movements during the flight of the UAV. For land cover mapping and flight path validation the acquired images were stitched together. Moreover, the normalized vegetation index (NDVI) was calculated to investigate the vegetation present in the acquired multi-spectral images.**

I. INTRODUCTION

Precision agriculture or precision farming was defined by the american national research council [1] as "a management strategy that uses information technology to bring data from multiple sources to bear on decisions associated with crop production". Moreover, it was defined that crop growth status monitoring is essential for the maximization of crop-yields. Since, farmers are able to take rapid, targeted action in case of nutrient deficiencies and pest infestation due to accurate and up-to-date maps of biophysical parameters like the water status [2]. The tradidional way to determine biophysical parameters of crops is to take ground samples at a small number of locations and estimate the parameters for unprobed locations [3]. For precision agriculture, remote sensing imaging products like biophysical parameter maps, with a high spatial resolution, have proven to be of high information content. For cost reduction and the limited availability of satellite images many photographic or video systems have been developed. Ultra light aircrafts or even UAVs equipped with a multi-spectral data acquisition system are a good compromise between the high performances that such sensors can provide and cost-effectiveness [4]. The increasing capabilities and the low price have pushed the UAV into the precision agriculture field. This paper propose a cost-effective system composed of multiple CMOS imaging sensors modified for the acquisition of multi-spectral data, fitted on board of a small UAV.

II. MEASUREMENT SETUP

The multi-spectral imaging sensor consits of 4 CMOS camera sensors with a pixel resolution of 2048 by 1536 pixels. All four cameras used fixed focus front lenses. One camera acquired VIS data by utilizing a triple-band-pass filter (400-500nm, 500-590nm and 590-670nm) to narrow the pass-band of the built-in Bayer filter of the RGB camera. Another camera acquired NIR data by utilizing a VIS-block filter with a cut-on wavelength of 850nm. The two remaining cameras utilized filters with centre wavelengths of 810nm and 910nm to acquire specific NIR data.

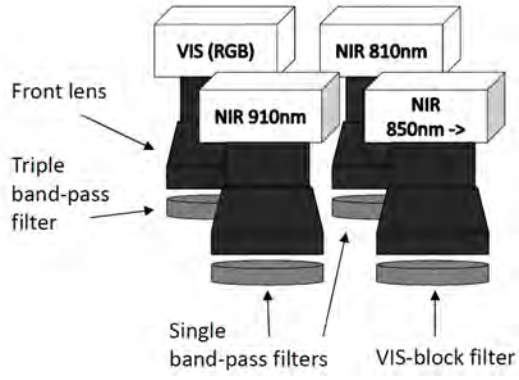


Fig. 1. Multi-spectral imaging sensor consiting of 4 CMOS camera sensors modified for the acquisition of multi-spectral image data.

The multi-spectral imaging sensor shown in Fig. 1 was mounted to a rotary wing UAV with a maximum take-off weight of 22lbs and a rotor diameter of 70in. To compensate the rapid rotational movements of the UAV the camera system was integrated in a brushless roll/tilt stabilized gimbal (see Fig. 2).

III. DATA ACQUISITION

For data acquisition a lithium polymer battery-powered PC system based on a mini-ITX mainboard and a solid state hard drive was used. Images where taken during the flight of predefined flight patterns at altitudes varying from 150 to 3

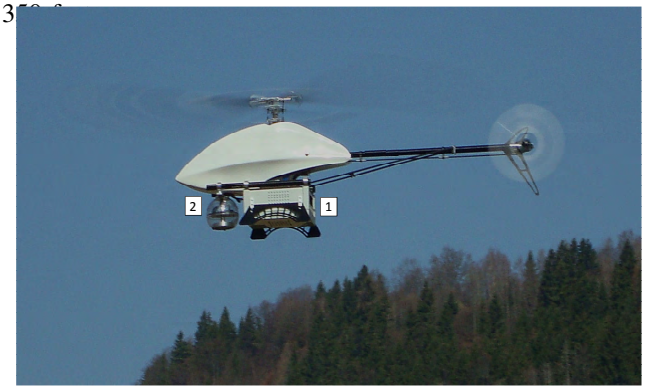


Fig. 2. Compact UAV equipped with roll-tilt compensated, multi-spectral imaging sensor (2) and data acquisition hardware (1).

All the sensors were accurately aligned and arranged in very close proximity so that the imaged field of view was the same

CTR Carinthian Tech Research AG, Europastrasse 4/1, 9524 Villach, Austria, thomas.arnold@ctr.at

for each sensor. However, due to the fact that the four sensors were unsyncronized, post-flight image registration was used to correct for the image frame offsets. A nonlinear image registration method based on optical flow was used to register consecutive images [6]. The images had to be stitched together into a larger mosaic image to get an overview image and to validate the flight path. After this post processing several vegetation indexes were calculated.

IV. Measuring Vegetation

The normalized difference vegetation index (NDVI) [5], originally developed for LANDSAT images, was used to quantify the presence of living vegetation in an image pixel. Near-infrared light from 700 to 1100nm is strongly reflected by green leaves, while visible light from 600 to 700nm is strongly absorbed. If the reflected radiation in near-infrared wavelengths is much more than in visible wavelengths, the vegetation in that pixel is likely to be green and may contain some type of forest. The NDVI is calculated by:

$$NDVI = \frac{NIR - R}{NIR + R}, \tag{1}$$

where NIR is the average signal intensity in the wavelength range from 800 to 1000nm and R refers to the average signal intensity in the wavelength range from 590nm to 670nm. The NDVI ranges from -1 to +1, where any negative values are mainly generated from water and snow, and values near zero are mainly generated from rock and bare soil. Values close to one indicate the highest possible density of green leaves. Fig. 3 (a,b) shows images of VIS and NIR data containing grassland and a field separated by an unmetalled track. These images were acquired in early spring, deciduous plants could not be imaged. However, the system was effective in classifying vegetation like grass and trees. Fig. 3 (c) shows the calculated NDVI. It can be seen that grass is well separated from the other areas in the scene.

V. Conclusion

In this paper a multi-channel CMOS camera system integrated in a small UAV is presented. The system was effective in identifying the present vegetation on flat agricultural fields. This low-cost inspection system can be used for the periodic imaging of tree canopies and field crops. With spectral imaging techniques spectral features can be identified to determine for example water stress, nutrient deficiency or pest infestation.

Acknowledgement

The Competence Centre ASSIC is funded within the R&D Program COMET - Competence Centers for Excellent Technologies by the Federal Ministries of Transport, Innovation and Technology (BMVIT), of Economics and Labour (BMWA) and it is managed on their behalf by the Austrian Research Promotion Agency (FFG). The Austrian provinces (Carinthia and Styria) provide additional funding.

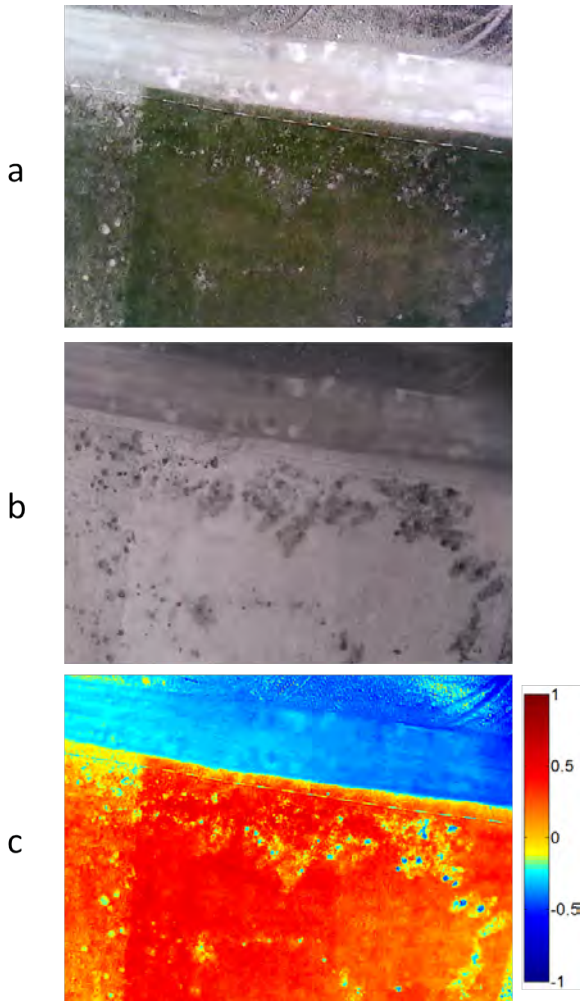


Fig. 3. This Figure shows VIS (a) and NIR (b) images acquired at an approximate altitude of 300ft. The calulated NDVI is shown in (c).

References

[1] I. S. Committee on Assessing Crop Yield: Site-Specifc Farming and N. R. C. Research Opportunities, Precision Agriculture in the 21st Century, National Academy Press, 1997.

[2] J. Berni, P. Zarco-Tejada, L. Surez, V. Gonzalez-Dugo, and E. Fereres, Remote Sensing of Vegetation From UAV Platforms using Lightweight Multispectral and Thermal Imaging Sensors, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVII, 2008.

[3] J. Bellvert, P. J. Zarco-Tejada, J. Girona, E. Fereres, Mapping crop water stress index in a Pinot-noir vineyard: comparing ground measurements with thermal remote sensing imagery from an unmanned aerial vehicle, Precision Agriculture, vol. 15(4), pp. 361-376, August 2014.

[4] C. C. D. Lelong, P. Burger, G. Jubelin, B. Roux, S. Labbe, and F. Baret, Assessment of Unmanned Aerial Vehicles Imagery for Quantitative Monitoring of Wheat Crop in Small Plots, Sensors 8, pp. 3557-3585, 2008.

[5] A. Elmore, J. Mustard, S. Manning, and D. Lobell, Quantifying Vegetation Change in Semiarid Environments: Precision and Accuracy of Spectral Mixture Analysis and the Normalized Difference Vegetation Index, Remote Sensing of Environment 73, pp. 87-102, 2000.

[6] J. Y. Xiong, Y. P. Luo, and G. R. Tang, An Improved Optical Flow Method for Image Registration with Large-scale Movements, Acta Automatica Sinica, 34:760-764, 2008.

Placing the Kinematic Behavior of a General Serial 6R Manipulator by Varying its Structural Parameters

Mathias Brandstötter[1] and Michael Hofbaur[1]

***Abstract*— Serial manipulators are most flexible when they have a general structure and a modular design. In such a case almost no limits to a robot designer are set. However, this high variability leads to a complex design problem on the basis of the purpose of the robot that should be achieved. In this paper we present a method to counter this problem by using a robot with known properties and adapt its structure according to a given task without changing its kinematic behavior. This means, a known behavior of a robot is transferred to a desired region in the workspace. Allowed adaptions within the geometric structure of modular robot are identified and are realized by suitable links of the manipulator. With this method, design parameters of the system are given, which makes it possible to apply a known kinematic property to an arbitrary end effector position and one system inherent direction. The feasibility of the method is shown by a simulated 6-DoF Cuma-type robot.**

I. INTRODUCTION

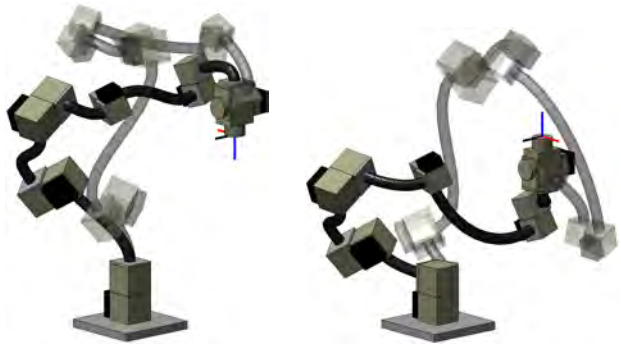
Whenever a given task for an industrial robot has to be performed an applicable manipulator (*design problem*) and its relation according to the task (*placement*) must be defined. The requirements of the according task (e.g., welding or load transportation task) can range from mechanical about kinematic to dynamic demands.

Since specific properties with monolithically structured manipulators like industrial robots are hard to achieve or even not possible, we focus in this paper on general serial manipulators in modular design. Moreover, we limit ourselves to robots with 6 rotational joints but permit any kinematic relationship between the rotational axes, which can be realized by suitable links. From this variety of design options follows that a design goal must be defined which in turn can be derived from the requested task.

To overcome this high-dimensional optimization problem we adapt a well-known manipulator with *Design Parameter Set* $\mathcal{D}$ to the specific problem. In order to perform an adaptation, it must be known which parameters of the manipulator can be changed without affecting the behavior, e.g., kinematic or dynamic properties.

A large number of papers were published regarding to manipulator design (e.g., [1], [2]) but most of them are related to performance optimization. As far as the authors know, none of the works deal with methods to vary the mechanical design of serial robots without kinematic changes.

To describe the kinematic structure of a serial robot the Denavit Hartenberg parameter (DH parameter) have proved favorable. DH parameters are partially detached from the de facto manipulator construction because they describe transformations between the intersection of consecutive joint axes with its transversals. Three parameters are required to describe the structure (d_i, a_i, and α_i) and one parameter represents the joint angle (q_i) [3]. Therefore, two degrees of freedom can be used for implementation alternatives. In Fig. 1 two different variants of the same kinematic structure of a general spacial 6R manipulator are depicted.



(a) Fist variant. (b) Second variant.

Fig. 1: Two Cuma-type arms which satisfy the identical DH parameters. The end effectors reach the same point but with various orientation. Each version is illustrated with two different postures (opaque ones correspond to the home position and the transparents to an other possible solution of inverse kinematics problem).

In this paper we show the influence of a variation of the design parameters of a manipulator on its kinematic properties. To what extent the structure of the robot can be modified without changing its properties should be clarified in the following chapter.

II. KINEMATIC INVARIANTS

As stated above, the DH parameters fix only three parameters of the robot structure and one additional parameter is needed for the joint value. As long as the transversals of the axis do not change the same DH parameters remain. For this reason, 10 design parameters are available for a given DH parameter set of a serial manipulator with 6 rotational joints. These freely selectable design parameters are denoted as $\tilde{d}_i$ and $\tilde{q}_i$ with $i = 1, \ldots, 5$, where $\tilde{d}_i$ is a translation of the ith joint along the ith axes and $\tilde{q}_i$ is a rotation of the ith joint about the ith axes.

We now want to clarify the effects of the individual freedoms on the end effector. Whenever the position of a

[1]M. Brandstötter and M. Hofbaur are with JOANNEUM RESEARCH Forschungsgesellschaft mbH., Institute for Robotics and Mechatronics, 9020 Klagenfurt am Wörthersee, Austria mathias.brandstoetter@joanneum.at and michael.hofbaur@joanneum.at

joint inside the kinematic chain ($i = 2, \ldots, 4$) is varied along its axis (translated or rotated) the modification has no kinematic influence to the end effector. But what happens, if the joints on the first or last axis are moved?

A translation of the first joint along the fist axes with $\tilde{d}_1$ leads to a displacement of all following axes with $\tilde{d}_1$ in the same direction. In Fig. 1 this influence is obvious to see when comparing the 2nd, 3rd, and 4th joint of the first and second variant. The situation behave similar with a joint rotation about the first axis arises. All subsequent axes rotate about the first axis with $\tilde{q}_1$. If the last joint is translated along or rotated about the last axis, only the end effector pose is affected. Exemplarily the orientation of the end effector of the Cuma-type arm in Fig. 1 has been changed by $\tilde{q}_5 = 180\,\text{deg}$ and moved along the first axis.

A further possibility to change the position of the end effector is to scale the whole manipulator. For this purpose, each length parameter (a_i, d_i and $\tilde{a}_i$, $\tilde{d}_i$) of the DH parameters is multiplied with the same factor C. As a result, the position of the end effector is varied but not directly scaled by C because the end effector is not scaled by itself.

A known kinematic property of a robot can now be placed in the workspace by the five design parameters $\tilde{d}_1$, $\tilde{q}_1$, $\tilde{d}_5$, $\tilde{q}_5$, and C which can be realized by adapting the link shape. As long as a desired point P is not located on the fist or last joint axis and both axes are not parallel P can be positioned arbitrarily. The freedom of orientation depends also on the orientation of the fist and last axis but it can be said that the orientation of the end effector can not be freely chosen.

III. EXAMPLE

In the following, we present a manipulator which has two specific kinematic properties. First, a pose that can be achieved by the maximum number of possible postures and second a configuration with local position isotropy.

The testbed for serial manipulators in our robotics lab consists of electromechanical joint modules from *Schunk GmbH* and customized curved link elements. By using these components, we are able to assemble general serial curved manipulators (CUMA-arms) to verify diverse theoretical results. Due to the modular design of the robot a large amount of different kinematic structures can be realized.

Table I describes the kinematic structure of the Cuma-type arm used in this example in terms of DH parameters. Two possible home positions ($q_1 = \cdots = q_6 = 0$) of the manipulator are displayed in Fig. 1 (opaque representation in each case). Both versions vary in their physical structure and end effector but the realizations are conform to the DH parameters given in Tab. I in either instance.

The inverse kinematics of the shown end effector poses in Fig. 1 provides a set $\mathbf{Q}$ with all 16 solutions of the six joint angles collected in $\mathbf{q}_i$, where $i = 1, \ldots, 16$. This manipulator is a member of the special class of 6R manipulators which have a region in configuration space where 16 real solutions can be found. The given Cuma-arm fulfills this property directly at its home position.

TABLE I: DH parameters of a Cuma-type arm with totally 16 real solutions available.

ith Link	d_i / mm	a_i / mm	α_i / deg
1	364.016	59.572	121.171
2	−176.450	331.501	−117.119
3	−104.741	123.019	−67.473
4	196.845	355.206	80.033
5	−248.121	198.526	74.333
6	57.346	0	0

An opportunity to describe the kinematic behavior of the end effector of a manipulator is the calculation of the isotropy. Therefore the Jacobian matrix $\mathbf{J}$ is needed which maps the actuator velocities $\dot{\mathbf{q}}$ to the linear and angular velocity $\dot{\mathbf{x}}$ of the end effector by $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$ for one specific configuration. $\mathbf{J}$ can be split into a position and orientation part as

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix} .$$

The isotropy of the linear and angular velocity are decoupled, hence, the *local position isotropy index* (LPII) can be computed by

$$\text{LPII} = \frac{\sigma_{P,\min}}{\sigma_{P,\max}}$$

where $\sigma_{P,\min}$ is the smallest and $\sigma_{P,\max}$ is the largest singular value of $\mathbf{J}_P$. The closer the value of this index to 1, the more homogeneous the behavior of the end effector with regard to linear movements. A configuration set with $\text{LPII} = 1$ can be found for the given Cuma-arm with

$$\mathbf{q}_{\text{iso}} = \begin{bmatrix} q_{\text{iso},1} & -120.8 & 221.5 & -149.8 & -123.9 & 41.0 \end{bmatrix}$$

where $q_{\text{iso},1} \in \mathbb{R}$. In case when only position isotropy is considered eigenvalue-based indices can be used although they are non-invariant due to changes of reference frame, scale, and physical units [4].

By the freely selectable design parameters ($\tilde{d}_1$, $\tilde{q}_1$, $\tilde{d}_5$, $\tilde{q}_5$, and C), the location of a special kinematic point (16 solutions, position isotropy) can be chosen arbitrarily.

IV. CONCLUSIONS

If a robot structure with a suitable workspace performance is known, it can be conducive to transfer the behavior of the end effector to a desired region in the workspace. In this extended abstract we have considered a method to perform this demand using the properties of a known modular serial manipulator.

REFERENCES

[1] Cem Ünsal and Pradeep K. Khosla, "Mechatronic design of a modular self-reconfiguring robotic system," in Proceedings of IEEE International Conference on Robotics and Automation, 2000, pp. 1742–1747, vol. 2, 2000.

[2] Christiaan J. Paredis and Pradeep K. Khosla, "Kinematic design of serial link manipulators from task specifications," The International Journal of Robotics Research, vol. 12, nr. 3, pp. 274-287, 1993.

[3] B. Siciliano, et. al., "Robotics: Modelling, Planning and Control," Springer Science & Business Media, 2009.

[4] E. Staffetti, H. Bruyninckx, and J. De Schutter, "On the Invariance of Manipulability Indices," Advances in Robot Kinematics, Springer Netherlands, pp. 57-66, 2002.

Autonomous charging of a quadcopter on a mobile platform

Anže Rezelj and Danijel Skočaj
University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, SI-1000 Ljubljana, Slovenia
Email: ar5781@student.uni-lj.si, danijel.skocaj@fri.uni-lj.si

***Abstract*—In this extended abstract we present the design and implementation of a mobile charging station that enables the autonomous recharging of the batteries of an unmanned micro aerial vehicle.**

I. Introduction

Unmanned micro aerial vehicles (MAVs) have recently become affordable and easily controllable. Typically, these aircrafts are equipped with different sensors. For example, the low-cost quadcopter Parrot AR.Drone [1] is equipped with two cameras, an accelerometer, a gyroscope, a magnetometer, a pressure sensor, an ultrasound sensor for ground altitude measurement and optional a GPS via on board USB. In a typical scenario, the user communicates with this quadcopter using the on-board WiFi, which is used for transfering sensor data to the operator and to remotely control the quadcopter. Since this quadcopter is affordable, easy to use, and well supported, it is widely used by hobbyist and it was well accepted by the research community as well.

One of the main requirements of a MAV is its low weight. Since lightweight materials are preferably used, MAVs are equipped with batteries with small capacity that significantly reduces the autonomy time of the robot and prevents long-term operations. On the other hand, the unmanned ground vehicles (UGVs) in forms of various mobile robots do not suffer from such severe restrictions with respect to the payload. They are therefore equipped with heavy batteries with large capacity, enabling even several hours of autonomy. A UGV can therefore serve as a very suitable mobile ground charging station for a MAV.

In this paper we present the design and implementation of a mobile charging station and describe an approach for landing of the quadcopter at this mobile platform, which enables autonomous charging of the MAV. Several solutions of this problem have already been proposed [2] [3] [4], however they do not fulfill the requirements that we have set. When designing the system we were aiming at achieving the following goals: (i) modular and lightweight modification of the MAV that does not significantly change its flying characteristics, (ii) simple and robust landing platform as a modular extension of the UGV that enables safe landing and taking off as well as fixation of the MAV for safe transportation, (iii) efficient and robust software solution for autonomous landing.

II. Hardware

The hardware that we developed to upgrade the MAV and UGV to support autonomous charging can be roughly split in two parts: (i) the landing platform and (ii) the control logic. The resulting recharging platform can be easily mounted on a mobile platform, e.g., on a Turtlebot, as depicted in Figure 1, or even placed on the ground or at any other flat surface.

A. Landing platform

The landing platform provides a safe and stable place for uninterrupted battery charging. To ensure that and to enable safe landing, we mounted four cones oriented upside-down near the corners of the landing platform. They are supposed to coincide with the MAV's landing pads and to ensure a fix position during the charging process. The wide ends of the cones help the MAV to slide in the correct position, allowing a small error (a couple of centimeters) in the landing position. The cones therefore facilitate safe and more accurate landing and at the same time allow non-occluded taking off.

A marker is placed in the center of the landing platform. It is used for more efficient and accurate detection of the platform and guiding the MAV to the correct position for precise landing, as described in Section III. To ensure better marker detection the background of the marker can be illuminated.

B. Control logic

The control logic consists of a microcontroller and a charger for a LiPo battery. Microcontroller, we use Arduino [5], implements the following functionalities. First, it detects when a drone has landed. Then it measures the MAV's battery voltage and in case of a low battery level it turns off the MAV's power supply and turns on the battery charger. The MAV's power supply must be turned off to enable safe charging. When charging is completed, the charger is turned off and the MAV is turned on. To enable turning the MAV's power supply on and off we added a power MOSFET transistor between the battery and the MAV. All the modifications made to the quadcopter increase its weight for less then 20g.

We can manually turn on the charger and adjust the marker background illumination by pressing the button and turning the potentiometer connected to the microcontroller. Arduino can be connected to a computer (tipically mounted on a UGV) via USB, which allows better integration of the landing station with the mobile platform.

III. Software

The main goal of the software part of our system is to detect the marker using the MAV's bottom camera and to control the MAV to land at the platform. An example of the MAV landing is shown in Figure 1. Landing is a critical part of our solution and can be roughly split in two parts: (i) detection of the marker and (ii) landing control. The marker detection must be robust, fast and as accurate as possible. The same requirements apply to landing.

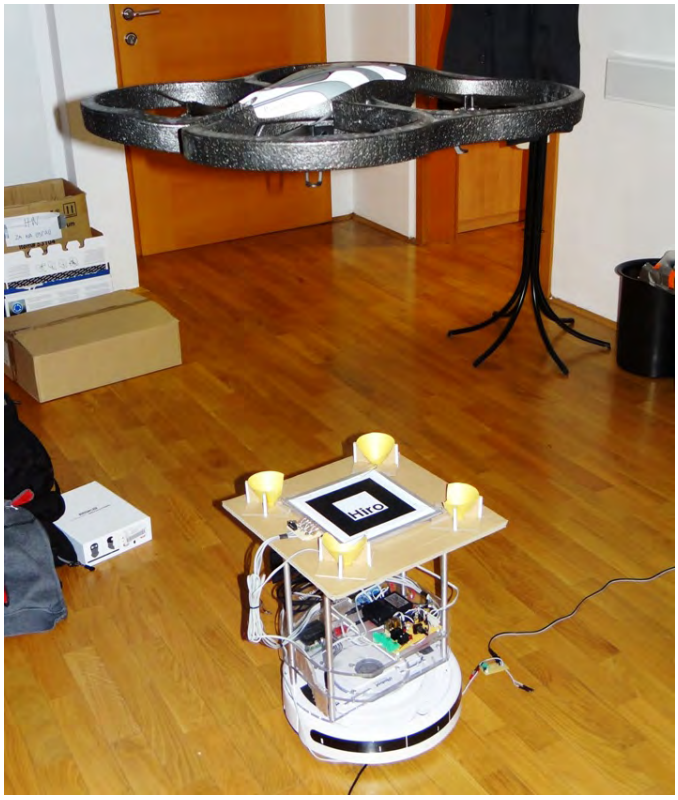


Fig. 1. Mobile platform Turtlebot with recharging platform.

A. Detection of marker

Robust, fast and accurate marker detection is also a critical requirement of augmented reality systems; an accurate position and orientation of the marker are needed to enable realistic rendering of the virtual object in the real scene in real-time. In our case the exact position in 3D space with respect to the camera, mounted at the bottom of the MAV, is needed to guide the MAV to the correct position enabling accurate landing. We therefore use the marker detection implemented in one of the commonly used AR systems, ARToolkit [6], [7]. It captures an image, which is then binarised using a dynamic threshold. In the binarised image a line fitting alghorithm is used to search for rectangles. When a rectangle is found, the algorithm searches for the pre-defined marker in it. At the end, the position and orientation of the marker are obtained, relative to the camera.

B. Landing control

The position and orientation of the marker are used to control the landing of the MAV. The system first reduces the noise in measurements by smoothing the obtained results. Then the MAV is rotated in the direction of the charging pins mounted on the landing platform, as shown in Figure 1. Then the MAV stabilizes above the marker and slowly descends and lands at the platform.

However, the developed system suffers from several limitations. The resolution of the bottom camera is very low and the viewing angle is very narrow. The low camera resolution decreases the accuracy of the estimated marker location. Due to the narrow viewing angle, the marker quickly disappears out of sight at low altitudes. When the height is decreased the MAV stability is reduced, which further complicates landing and reduces the probability of a successful landing.

IV. Conclusion

In this extended abstract we presented a system for recharging a MAV's batteries by autonomous landing on a mobile charging station. The proposed solution requires very little and simple modifications on the MAV without increasing the weight of the MAV noteworthy. Our solution can be easily adapted to other MAVs with the same or larger number of rotors. In our future work we plan to develop the electronics for power control and recharging, which is plugged between the MAV and the battery, to allow the MAV to be turned on when charging the battery. We also plan to improve the marker tracking and MAV control mechanism to ensure more reliable and accurate landing.

There are enormous practical applications of the developed system. The recharging platform can be mounted at a fixed location or on a moving platform. It can be mounted practically on any mobile vehicle or mobile robot or at any flat surface. Similar solutions can be used in natural disasters, accidents in factories, fires, monitoring of agricultural plants and much more. In a similar way, in the future the permanent locations could be used for recharging the batteries of delivery drones in cities or in less populated and harder to reach areas.

References

[1] AR.Drone. (2010) Last checked: 08.03.2015. [Online]. Available: http://ardrone2.parrot.com/

[2] D. G. K.-M. D. Michael C. Achtelik, Jan Stumpf, "Design of a Flexible High Performance Quadcopter Platform Breaking the MAV Endurance Record with Laser Power Beaming." Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE,, 2011.

[3] J. R. M. Paulo Kemper F., Koji A. O. Suzuki, "UAV Consumable Replenishment: Design Concepts for Automated Service Stations." J Intell Robot Syst, 2011.

[4] Y. Mulgaonkar, "Automated Recharging For Persistance Missions With Micro Aerial Vehicles." Master thesis, University of Pennsylvania, 2012.

[5] Arduino. (2005) Last checked: 08.03.2015. [Online]. Available: www.arduino.cc

[6] ARToolkit. (1999) Last checked: 08.03.2015. [Online]. Available: http://www.hitl.washington.edu/artoolkit/

[7] B. M. Kato H., "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System." San Francisco, USA: In Proceedings of the 2nd International Workshop on Augmented Reality, October 1999.

An Autonomous Forklift for Battery Change in Electrical Vehicles

Stefan Kaltner[1], Jürgen Gugler[2], Manfred Wonsisch[3] and Gerald Steinbauer[1]

I. Introduction

Due to increasing e-commerce on the internet and changes in peoples' shopping behavior there is a huge demand on parcel delivery agents. Most of the deliveries are made on short or medium range distances. Therefore, the use of electrical vehicles is perfectly suitable for this application. While electrical propulsion systems and charging concepts are well established for passenger cars only a few systems are available for transporters. The underlying idea of this project is to develop an automated battery change process for parcel delivery transporters. The vision is to have a service station where the transporter is parked and a robot automatically removes the empty battery and replaces it with a charged one. Due to the heavy weight of such a battery of around 600 kg a forklift was chosen as basis. The requirements for the robot were: (1) autonomy - no interaction with the human driver, (2) adaptation to the situation - no need to park the transporter on an exact spot, and (3) quick change - similar to a regular tank stop. The contribution of this paper is the development of an automated forklift based on available ROS packages that have been adopted for an uncommon kinematics and a system architecture based on CAN-bus communication. In the remainder of the paper we present the hardware and software architecture of the system and discuss the obtained results.

II. System Overview

In order to develop a control system for the autonomous forklift a fully functional model in the scale 1:2 was designed and build. Figure 1a depicts the design of the robot. The realized model is shown in Figure 1b.

(a) CAD sketch. (b) Realized System.

Fig. 1: Automated Forklift.

This work was partly supported by the Austrian Research Promotion Agency (FFG) within the Forschungsscheck program.

[1]Stefan Kaltner and Gerald Steinbauer are with the Institute for Software Technology, Graz University of Technology, Graz, Austria. steinbauer@ist.tugraz.at [2] Jürgen Gugler is with Tecpond e.U.. juergen.gugler@tecpond.at [3] Manfred Wonisch is with Ingenieursbüro Wonisch. manfred-wonisch@gmx.at

The system comprises a driving system based on Ackermann-steering and a battery holder movable up and down using a shaft. The sensor system is realized using a 2D lidar mounted above the front wheels pointing forward and a Asus 3D sensor pointing towards the movable blade. The central control unit is formed by a laptop.

Fig. 2: Ackerman-steering of the forklift. Steering motor on top not shown.

The driving system comprises two passive front wheels (yellow wheels in Figure 1a). The steering and actuation unit comprises a wheels steered by one vertically mounted motor and actuated by a second motor directly mounted on its shaft. The unit is shown in Figure 2. Each of the motors uses a gearbox. Moreover, they are connected to a shaft encoder to measure the rotation of the wheel as well as its orientation around the vertical axis to provide the robot's odometry. In order to get an absolute orientation the vertical encoder is initialized using an inductive end-switch at each side. All motors, encoders and switches are connected to two controller on a central CAN-bus.

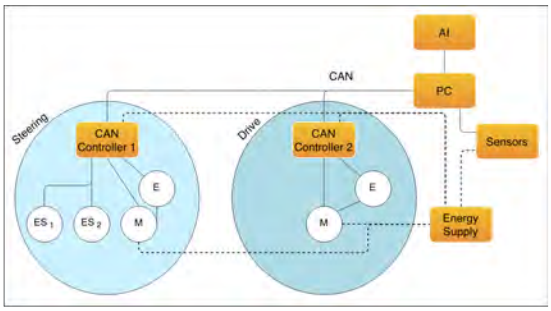


Fig. 3: Overview of the control system.

Figure 3 depicts the control architecture of the system. For a detailed description we refer the reader to [1]. The two CAN-controller are connected to the central PC using a USB/CAN adapter. The communication between the PC and

the controller uses the CANOpen protocol. The robot control software running on the central PC is implemented using the common Robot Operating System (ROS) [2]. Due to the fact that CANOpen is only supported for very specialized setups a general object hierarchy was implemented allowing to control the combination of several several differnt CAN-based modules (i.e. driving, steering) forming various kinematics.

The automated navigation of the robot is based on the well-known *move_base* package provided by ROS [3]. The package uses the adaptive monte carlo localization (AMCL) [4] to estimate the robot's position using a 2D map of the environment and data from a 2D lidar. The global path-planing is based on the same map and uses common planning techniques like NavFn. The local planing and path execution is based on the dynamic window approach. Usually after some parameter tweaking *move_base* works fine for the navigation of most robots. The drawback is that it makes some assumptions that do not hold in our application. For the path-planning it assumes that the robot has an almost circular footprint, has an differential or omni-directional drive and is able to turn on the spot around this vertical axis. Moreover, the motion model used in the localization module supports only differential and omni-directional drives.

Our robot is based on Ackermann-steering and violates some of the assumption leading to suboptimal performance of the navigation module. For instance the module is not able to plan trough a narrow door even if a path is possible for the forklift or it bumps into walls in narrow passages due to imprecise localization. Both problems are caused by the different kinematics of the forklift. Although, basically an Ackermann-steering based robot can be transfered to a differential drive kinematics it shows a different behavior because the vertical rotation axis is between the two front wheels far off the center of the robot. *move_base* uses a circular footprint around this axis for the planning. If the entire robot should be covered by the footprint the diameter has to be quite large covering a lot of unnecessary space and preventing planning trough narrow passages. Moreover, AMCL uses the velocity motion model which models the uncertainty caused by the imprecise control velocities ω_l and ω_r of the two wheels of a differential drive. In the forklift the motion is caused by the combination of a steering angle α and a single wheel velocity ω rather than by combining ω_l and ω_r.

In order to improve the performance of *move_base* used for a forklift we did two modifications. Instead of NavFn which uses the circular footprint to calculate a navigation function over the map and a gradient decent method to plan the path we integrated the search-based planner lab (SBPL) lattice planner [5]. This planner is a sample-based planner that generates a collision-free path using a sample-based search and motion primitives. The planner concatenates motion primitives. Therefore, the planner directly incorporates the true kinematics of the robot and is able to find a higher number of more suitable plans. The use of this planner allowed us to automatically navigate the robot trough doors just a little wider than the robot itself. Moreover, we implemented a updated motion model for the localization that reflects the uncertainty of the control parameter of the forklift instead the one assumed for a differential drive. This improved the precision of the localization leading to less bumps into obstacles.

sensor	global	local	# success	# collisions	goal failed
2D lidar	NavFn	TPR	1	6	3
2D lidar	NavFn	DWA	1	6	3
2D lidar	SBPL	TPR	3	7	0
2D lidar	SBPL	DWA	10	0	0
3D Asus	NavFn	DWA	1	7	2
3D Asus	SBPL	DWA	2	6	2

TABLE I: Performance of the System for different setups. TPR stands for TrajectoryPlannerROS.

III. Experimental Results

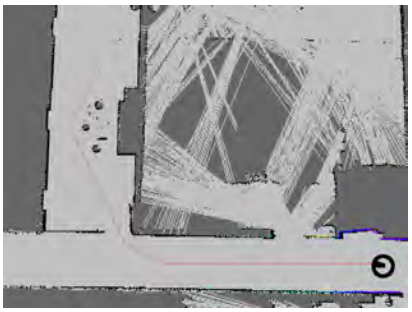


Fig. 4: Evaluation task. The figure shows the occupancy grid of the environment (gray) and a sample generated path (red).

The proposed system was systematically evaluated using different combinations of sensors and global and local planning methods. The evaluation was done in a common office environments with common doors, corridors and rooms. Figure 4 shows a typical generated path for a navigation task. The robot should navigate along a corridor, should traverse a narrow door and avoid a group of obstacles. Table I shows the performance of the system for navigation tasks using different sensors and planners. All combinations where tested 10 times. More detailed results can be found in [1].

IV. Conclusion and Future Work

In this paper we presented the hardware and software architecture for an autonomous forklift. The central navigation function is based on the ROS *move_base* package. Due to the tight coupling of the package to differential or omni-directional robots the planning module was adopted for robots based on Ackermann-steering. An empirical evaluation showed that the adopted navigation is much more reliable. In future work we will work on the replacement of the expensive 2D lidar by 3D cameras.

REFERENCES

[1] S. Kaltner, "Autonomous navigation of a mobile forklift," Master's thesis, Faculty of Computer Science, Graz University of Technlogy, 2015.

[2] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[3] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 300–307.

[4] S. Thrunn, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[5] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *International journal Robotics Research*, vol. 28, no. 8, pp. 933–945, Aug. 2009.

A smart real-time anti-overturning mechatronic device for articulated robotic systems operating on side slopes

Giovanni Carabin[1], Alessandro Gasparetto[2], Fabrizio Mazzetto[1], Renato Vidoni[1]

***Abstract*— At today, smart, and cheap mechatronic devices together with open-source software allow to design and prototype new and effective systems or at least to apply existing platforms to new fields. In this work, starting from the evaluation of the stability of autonomous robotic platforms, a smart and effective real-time anti-overturning mechatronic device for robotic systems operating on side slopes has been developed. It allows to recognize both the orientation and possible critical positions of an autonomous robotic platform, in particular articulated, and to alert the user or stop the motion in case of incipient rollover. The developed platform can be adopted as a cost-effective solution and an effective safety device.**

I. INTRODUCTION

The application of mechatronic devices and smart autonomous robots in agriculture and forestry is becoming one of the most promising research topics for roboticists and engineers [1].

In the past, machines and robots needed predefined ways, as in an assembly line and the creation of something like machine paths was necessary. Moreover, robotized or semi-automated attempts and applications, such as weeding, crop and environmental monitoring, crop harvesting and fruit picking [2], [3], [4], [5] have been developed.

Since agricultural autonomous machines should operate also on uneven terrains, the capability of traveling and eventually moving nimbly between wine rows on hills and on different surfaces becomes the challenge that has made them a very interesting research topic [6]. In this framework, automation in side-slope working activities can be considered at an early stage. This is mainly given by the problems related to the stability of the adopted vehicle and to the lack, in the past, of a sufficient technological level for creating safe and stable systems. With these requirements, the attention has been given both on wheeled and tracked conventional architectures. More recently, articulated configurations, where a central active hinge is in charge of driving the steering angle, become particularly interesting for their capabilities in terms of steering angle and versatility both in the tractor field [7] and from a theoretical point of view [8], [9]. One of the most important issues is the capability to maintain the autonomous robotic platform stable or at least to avoid unsafe configurations by means of mechatronic devices that can measure in real-time the safety margin and predict and not allow in an active manner the dangerous configurations. In this work, the target of developing a cheap and smart mechatronic device able to both measure, estimate and alert the system operator or stop the autonomous robot when approaching unstable configurations has been addressed.

* This work was supported by the Free University of Bolzano internal research funds project TN2024 - ARTI

[1]G. Carabin, F. Mazzetto and R. Vidoni are with Faculty of Science and Technology, Free University of Bolzano, Bolzano, Italy Giovanni.Carabin, fabrizio.mazzetto, renato.vidoni @unibz.it

[2]A. Gasparetto is with the DIEG, University of Udine, Udine, Italy gasparettto@uniud.it

II. INSTABILITY EVALUATION

The instability of the robotic platform, either tracked, wheeled or articulated, has been considered with a quasi-static approach. Then, the position of the projection of the vehicle center of mass (CoG*) with respect to the support polygon of the vehicle, defined by connecting the footprints of the vehicle supports with straight lines, Fig. 1, has been evaluated.

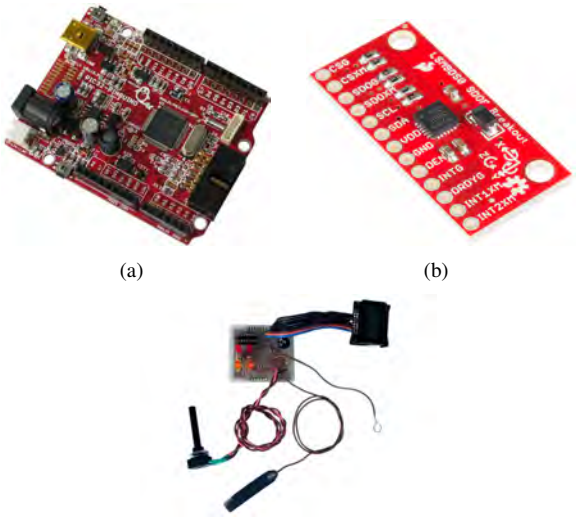
Fig. 1. Sketch of the possible different platforms and supports: on the left a classical wheeled system with a rigid frame, on the right an articulated platform with a β steering angle

The approach adopted in [9] for defining both the instability and the safety index, in particular of the articulated platform, has been exploited here.

III. ANTI-OVERTURNING MECHATRONIC DEVICE

The developed mechatronic system, which can be mounted on the robotic structure, is able to identify the orientation and stability condition of the mobile platform. The heart of the system is represented by a control board, which acquires all the signals from different sensors (e.g. inertial sensors, contact sensor, potentiometer), processes them and acts accordingly. The board is mounted directly on the robot, and an interfacing board, i.e. shield, has been designed for the connection with the sensors. Both are shown in Fig. 2.

The system control board is a generic and cheap PIC32-Pinguino board, based on the Microchip PIC32MX440F256H microcontroller and programmed in C language by exploiting the "Pinguino IDE" development suite. To determine the orientation of the robot or, in the case of an articulated chassis, of each of the two parts of the robot, inertial sensors with 9 degrees of freedom, IMU - Inertial Measurment Units - have been exploited. These are capable of measuring the angular speed, the



(a) (b)

(c)

Fig. 2. (a) PIC32 Pinguino; (b) IMU; (c) shield board

acceleration, and the Earth's magnetic field intensity, according to three axes. By processing these quantities, the space orientation of the system can be obtained. The considered IMUs are open hardware boards, based on the LSM9DS0 ST Microelectronics integrated with a 3-axis gyroscope, a 3-axis accelerometer, a 3-axis magnetometer, 16 bit resolution and a serial communication interface SPI/I^2C. The communication between the sensor and the control board has been done through the standard SPI and the parameters for the three kind of sensors of the IMU have been properly set. Since the C library to be used to manage the inertial sensor was not available in Pinguino IDE environment, a simplified version of the library has been created for the purposes of the project. As previously said, a shield interfacing board for allowing easy connection between the control board and the external devices has been prototyped with two IMU connectors, one LCD 8x2 display, 4 signaling LEDs and 4 I/O.

IV. Orientation System

As described in [10], there are essentially three strategies for obtaining the space orientation from an IMU sensor: a first type made through an Eulerian approach, with three variables, simple but expensive in computational terms and with the drawback related to singularities; a second one, very fast but without a physical interpretation of the motion, that uses a representation of the vectors by means of quaternions and, thus, four variables; a third method, the one used in this work, the so-called DCM (Direct Cosine Matrix). It represents the orientation by nine parameters, corresponding to the elements of the rotation matrix. By means of this approach, the two rotation matrices, R_{gyro}, i.e. the one derived from the gyroscope data, and R_{corr}, the one obtained from the accelerometer and magnetometer, the complementary filter has been preferred to the Kalman one to not to overload the computation. Then, after the definition of the calibration procedure, the control programme has been written.

V. Control software

The logic of the implemented controller is as follows: first of all the peripherals configuration and variables initialization are done; then the program enters in the main loop, where the stability condition is computed and the main info are displayed on the LCD and on LEDs. The main loop is interrupted every 50ms, by generating an appropriate interrupt to call functions that mainly deal with the orientation computation of the robot. This solution has been chosen to meet the time requirements for updating the orientation, with respect to the relatively long time required for the calculation of the condition of stability.

VI. Conclusions

The developed mechatronic system, basically made by a main board, a shield board and IMU sensors, together with a LCD display and warning LEDs, allows to estimate both the orientation of a robotic platform and the instability evaluation by means of a quasi-static approach. It has been developed not only by accurately choosing smart and cheap devices but also by developing and exploiting software solutions that allow a real-time estimation of the orientation and dangerousness of the condition. The system could be integrated in each autonomous robotic platform that has to work on side-slope activities. In particular, it has been developed for articulated platforms particularly suitable to move nimbly between wine rows on hills and on different surfaces.

References

[1] C.G. Sorensen, R.N. Jorgensen, J. Maagaard, K.K. Bertelsen, L. Dalgaard, M. Norremark, Conceptual and usercentric design guidelines for a plant nursing robot. Biosystems Engineering, 105(1), 119-129, 2010.

[2] N. Asimopoulos, C. Parisses, A. Smyrnaios, N. Germanidis, Autonomous vehicle for saffron harvesting. Procedia Technology, 8(Haicta), 175-182, 2013.

[3] Z. De-An, L. Jidong, J. Wei, Z. Ying, C. Yu, Design and control of an apple harvesting robot. Biosystems Engineering, 110(2), 112-122, 2011.

[4] Z. Gobor, P. Schulze Lammers, M. Martinov, Development of a mechatronic intra-row weeding system with rotational hoeing tools: theoretical approach and simulation. Computers and Electronics in Agriculture, 98, 166-174, 2013.

[5] S. Yaghoubi, N.A. Akbarzadeh, S.S. Bazargani, S.S. Bazargani, M. Bamizan, M.I. Asl, Autonomous robots for agricultural tasks and farm assignment and future trends in agro robots. International Journal of Mechanical and Mechatronics Engineering IJMME-IJENS, 13(03), 1-6, 2013.

[6] M. Tarokh, H.D. Ho, A. Bouloubasis, Systematic kinematics analysis and balance control of high mobility rovers over rough terrain. Robotics and Autonomous Systems, 61(1), 13-24, 2013.

[7] F. Mazzetto, M. Bietresato, R. Vidoni, Development of a dynamic stability simulator for articulated and conventional tractors useful for real-time safety devices. Applied Mechanics and Materials, 394, 546-553, 2013.

[8] A.L. Guzzomi, A revised kineto-static model for phase I tractor rollover. Biosystems Engineering, 113(1), 65-75, 2012.

[9] R. Vidoni, M. Bietresato, A. Gasparetto, F. Mazzetto, Evaluation and stability comparison of different vehicle configurations for robotic agricultural operations on side-slopes, Biosystems engineering, 129, 197-211, 2015

[10] N.H.Q. Phuong, H.J. Kang, Y.S. Suh, and Y.S. Ro. A DCM Based Orientation Estimation Algorithm with an Inertial Measurement Unit and a Magnetic Compass. Journal of Universal Computer Science, 15:859-876, 2009.

The future of in-house transports

incubed IT stands for innovative robotics solutions. We have developed a system to automate transport vehicles as autonomous, intelligent and cooperative shuttles, comprising navigation core, fleet management server and user interface. The result is what we call a Smart Shuttle.

Our Smart Shuttles are able to roam freely in the dedicated area without the need for special orientation landmarks. They act completely independent and flexible, dodging dynamic obstacles while driving elegantly. Routes are also changed in a dynamic way if necessary. That is, if an obstacle blocks the shuttle's path completely, it simply finds an alternative route to its goal.

Due to their safety certificate and intelligent behavior, the shuttles can share the same space with humans and other vehicles (e.g. forklifts) without obstructing or endangering them. If desired, the user can influence the shuttle movements by easily defining special traffic areas in the map they are operating on. So if a shuttle is, for example, not allowed to enter a certain region, it can be prevented to do so by the means of a forbidden area. Many more traffic areas are available, like one-way streets, right-hand traffics or roundabouts.

By breaking down the totality of transports into goods on individual shuttles (rather than moving totes on a centralized conveyor), our Smart Shuttle Solutions are especially scalable. We support large fleets of shuttles to cope with even highly-complicated mappings between arbitrary sources and goals. The shuttle system is thus able to provide a constant throughput, but at the same time the number of active shuttles can be reduced in low or increased in peak seasons/times, saving energy and resources.

This way, Smart Shuttles are a perfect match for the challenges of industry 4.0.

Smart Container Shuttle – the smaller of incubed IT's two standard shuttles