MŰEGYETEM 1782

Budapest University of Technology and Economics

Dept. of Control Engineering and Information Technology

Tamás Urbancsek

Modern control architecture for designing multiagent telerobot systems

Summary of PhD thesis

Supervisors: **Dr. Vajta László**, associate professor
Prof. Dr. Arató Péter, professor
Budapest University of Technology and Economics
Dept. of Control Engineering and Information Technology

Budapest, 2009.

Contents

List of scientific theses

1. Introduction

By the end of the millennium, due to the developments of science and engineering, the doors started to open one after the other for providing solutions with robotics to more and more material handling issues. Nowadays, we can see robots not only in environments which can be easily defined and on which simple models can be built, but also in our everyday life, the entertainment industry and in our homes. And although trends show that embedded systems are getting more and more intelligent, researchers of robotics recognized that it is impossible to develop fully autonomous robots for tasks in variable environments or where modeling the environment is difficult.

In these cases telerobotics, which incorporates the achievements of teleoperation and telepresence, comes to our assistance. It deals with performing material handling tasks in poorly specified environments with the help of personal telesupervision. Telerobotics allows of doing space research jobs, carrying out tasks in dangerous environments, or performing telesurgery operations by utilizing the residing synergy of artificial intelligence and human intuition.

The Department of Control Engineering and Information Technology of the Budapest University of Technology and Economics has been doing research on computer vision and intelligent robots since the beginning of the 1990s. Several of their scientific papers [26] [27] [28] and theses [29] [30] established and supported my research in the topic of telerobotics.

Structure of Telerobot Systems

The base of telerobot systems is the general model of human supervised control, which is represented by the figure bellow:

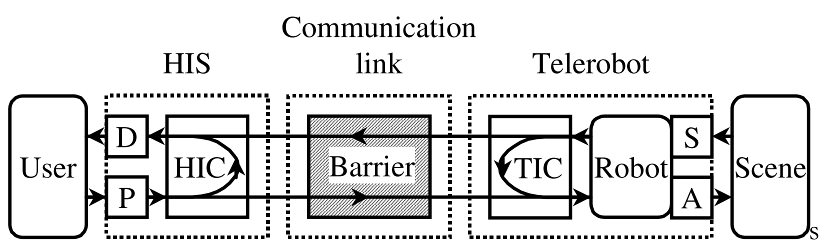

Fig. 1: General structure of human supervised telerobot systems based on [32]
D: displays, P: positioning, S: sensors, A: actuators,
HIS/HIC: Human interactive System/Computer, TIC: Task Interactive Computer:

The operator connects to the supervised telerobot system through a Human-Interactive System (HIS). His commands are sent to the telerobot via a dedicated communication channel. The signals processed by the sensors of the telerobot provide feedback about the events in the manipulation space, and so about the effect of the command.

The communication channel makes direct control difficult in most of the telerobotic applications. It comes with a fairly significant delay, provides only limited bandwidth, and is often unreliable (e.g. there may be package losses in IP networks). The sometimes dynamically changing time delay (jitter) can be a further difficulty in controlling a robot. This control engineering problem can be solved by including two feedback loops.

The operator system side feedback has an ergonomic aim: it informs the operator about the fact that the command was given and the possible effect it may cause, enabling continuous control despite the time delay in the communication channel. On the other hand, the telerobot side feedback is an intelligent control loop that has to move the telerobot based on the information processed by the robot's sensors and the instructions of the operating person.

A basic characteristic of telerobot systems is this so called intelligent **semi-autonomous operation**. This allows the telerobot to react adequately to unexpected events, such as being able to avoid a suddenly appearing object without an instruction from the operator. This is important especially in situations where a robot would collide within the double of delay time, since there is not even a theoretic chance that the operator can prevent collision. Depending on the intelligence of the telerobot and the variety of the environment, the operating person can be given its task from a wide range of possibilities from direct to strategic control.

The NASREM [28] telerobot reference model helps in designing the intelligent control loop that realizes the semi-autonomous behavior described above.

Multiagent robotics

Technical papers give various definitions for the concept of agent. According to a general definition, an agent is an autonomous entity that strives to reach its goals, can make its own decisions and can react reactively, or sometimes proactively, to the changes of environment. An agent's important feature is that it can communicate with other agents.

Multiagent robotics mostly deals with groups of mobile robots which fit the above definition. Robots that can make their own decisions and can proactively adapt to changes of the environment are used for problems in which coordinated interaction is needed in relatively far places.

Controlling each agent can be executed using various methods. The schema based behavior oriented approach of Arkin [33] was successfully applied in many cases. Robot agents have behavioral patterns, all of which try to influence their direction of movement. Each behavioral schema produces a vector in a direction that the robot considers the correct one, and with a length that is in proportion with the importance of the behavior in the given situation. Normalizing the sum of these vectors gives the resultant direction of movement of the robot, which is then executed by the motion planning algorithm. A detailed explanation of approach can also be found in [34].

Multiagent telerobotic systems

Multi-agent telerobot systems (MTR) enable one or more users to supervise a group of remote robots by combining the desirable characteristics of telerobotics and multi-agent robotics. The problems that arise during designing such a system cannot be handled with a method that is usually used in one or the other discipline. In fact, further questions come up:

- How can the operator divide his attention when controlling the robots of a group simultaneously?
- How can a group of operators who control the same robots cooperate?
- How can the commands of operators and the instructions of other agents be combined with a robot's own decision?

The planning methods used for telerobot or multi-agent systems can be made general with some modifications. For example, the schema based control can be applied to multi-agent telerobot environments in two ways.

- The supervising user can be regarded as an agent who is equal to the other robots, and who tries to influence one or more robots' direction of movement directly.
- The supervising user sets the weight of behavioral schemas directly; therefore, from the system's point of view, he is an entity above agents.

Consequently, multi-agent telerobotics can be regarded as a general discipline one of whose special case is telerobotics when controlling only one robot, and another is multi-agent robotics when there is no supervisor.

2. Research objectives

My theses deal with intelligent control of multi-agent telerobotic systems. I set up the following subtasks as my objectives:

- Hierarchical modeling of semi-autonomous control of multi-agent telerobot systems. In technical papers we often meet the fact that the NASREM hierarchical reference model used for controlling telerobots cannot be applied to the planning of supervisory control of multi-agent systems. So, the main goal of my research was to model the whole controlling issue with hierarchy levels upon one another, and creating the model of the system components that allow multi-agent behavior. The model of each controlling level contains the manipulation space's most important elements belonging to the given controlling hierarchy level. The model also explains how the parameters of these elements can be obtained from available sensor signals.
- When examining multi-agent telerobot systems we often see stepping robots, which are controlled in a motion primitive level by defining series of key positions. My second research objective was to be able to evaluate an optimal set of kinematics based elementary movements that can be used by the higher hierarchy levels to keep the telerobot in its planned trajectory.
- In most of the telerobot systems we use visual sensors. Using a lens with small depth of focus, objects' position in space (more precisely their depth) can be automatically estimated based on how defocused they seem. The task to complete is to model the geometry of a chosen telerobot's manipulation space and estimate the position of its manipulator based on passive vision.
- Applicability of the reference model and the algorithms has to be verified by explaining how two distinctly different multi-agent telerobot systems function.

3. Chosen methods and tools

The common characteristic of the position estimation and motion planning algorithms explained in the thesis is that both optimization problems are traced back to searching for the position of minimum of a non-linear cost function that can be computed numerically. The derivative of cost functions can be defined numerically, thus the optimum can be approximated with gradient based iterative methods. In the experimental phase I performed this using MATLAB Optimization Toolbox.

For the offline optimization of the experimental humanoid robot's motion I had to simulate the robot's dynamic behavior. The model was made in language Modelica [46], which is designed especially for such purpose, while the simulation was carried out in Dymola environment [45]. The position feedback algorithm, as well as the other software components of the created humanoid teleoperation system, were implemented in Microsoft. NET 2.0 using C#.

For processing the visual data for the position estimation I used MATLAB Image Processing Toolbox, while measurement and error analysis was made with Statistics Toolbox.

4. Scientific results

My scientific results can be divided in three main groups. I summarize these in the following three subsections.

4.1. Control hierarchy reference model of multiagent telerobot systems

In the beginning of my research work I studied the borderland of telerobotics and multiagent robotics.

Thesis 1 [5] [6] – Chapter 2

I have generalized the NASREM telerobot control system architecture by extending it with new components which enable the usage of the new model for designing intelligent control hierarchy of multiagent telerobot systems.

The NASREM control system architecture

A NASREM means „NASA/NBS Standard Reference Model for Telerobot Control System Architecture" [31]. It is similar to the standard ISO/OSI reference model that enables communication between open systems. Both models aim at systematizing of tasks solving a complex problem and structuring them based on their natural hierarchy. Each level hides the complexity of the subjacent layers and provides an abstract service to the superior layer.

NASREM proposes a way to design an intelligent control loop that implements semi-autonomous behavior of telerobots. It divides the control problem into the following six levels:

1. **Servo level** controls a robot joint according to a given reference signal. The control problem is how to follow the reference signal while considering outer disturbing effects and providing the optimal control signal.
2. **Primitive layer** deals with coordinated control of some adjacent servos (i.e. a leg or wheels of a robot) based on the extended dynamic model.
3. **Elemental move or E-Move level** aims at trajectory planning of the whole robot body based on the planned motion trajectory.
4. **Task level** deals with planning, coordinating and executing robot motions in order to reach a particular goal.
5. **Service bay level** deals with coordinating all tasks to be fulfilled at the same location – same service bay. Main tasks are managing and assigning shared resources – like place, tools – to robot actors.
6. **Mission level** coordinates all the tasks at all service bays.

Each level deals with an aspect of the complex telerobot control and consists of three logical modules.

- sensory processing module
- world modeling module
- task decomposition module

The levels and the internal structure of the levels are depicted in the following figure.

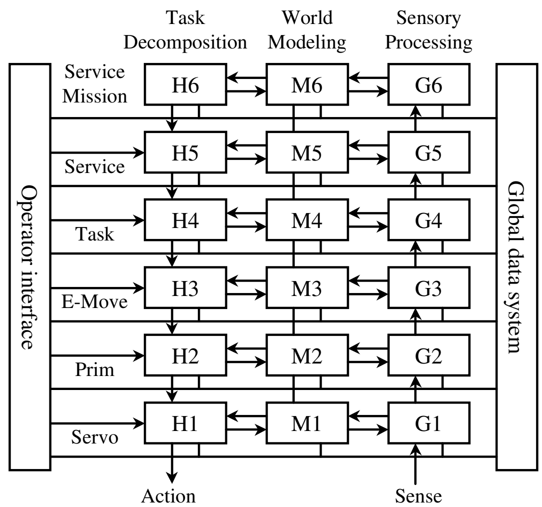

Fig. 2: Structure of the NASREM reference model [31]

4.1.1. Generalizing of NASREM to multiagent telerobot systems

My first research phase aimed at multiagent telerobot systems. I have studied how to extend fully autonomous robot systems with remote supervision control, that is, how to apply models and methods used in telerobotics for robotic systems that consist of multiple independent and intelligent robot agents.

Although there are valuable publications available in this research topic, i.e. PhD thesis of K. S. Ali [34] surveys the modern theories, I have realized that there is no unique accepted reference model for implementing the semi-autonomous behavior for multiagent telerobot systems.

First, I have elaborated the general model of a multiagent telerobot system supervised by multiple operators. This can be considered as generalized model of the human supervision control shown in Fig. 1.

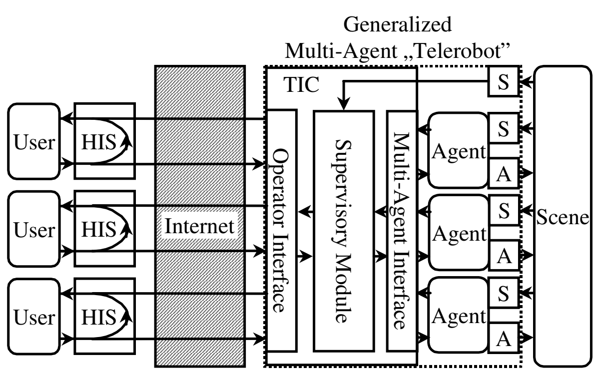

Fig 3: Generalized model of multiagent telerobot systems [5]

The robot system consists of multiple agents; each of them can possess own sensors and actuators. Even multiple operators can control these robots. I have introduced new components, which are not present in conventional multiagent systems.

The communication between robot agents occurs over the **multiagent interface**. This represents the distributed communication among the robots and of course the communication of the central system. The agents can share information about their status, environment, they can give commands and make orders. The communication can be point-to-point or can happen in a broadcasting way. The definition and the implementation of the interface depend strongly on the particular system and the particular realization; it is defined by its protocol.

Supervisory module receives and processes the commands of the operators; it also can possess a central sensor system. The generalized telerobot reaches the operators over the operator interface, which is also defined by its intercommunication protocol.

According to the literature the role of an operator may appear two different ways for the agents. The operator can be either a peer agent, in this case the role of the supervisory module is similar to the other individual agents; it communicates similar information and commands. On the other hand, it may appear as a superior coordinator; in that case its communication and internal structure is fundamentally different than the other agents.

The most important part of the model is how to control the individual agents. In this first phase of my research I have studied how to generalize NASREM for multiagent telerobot systems. I have proposed the following control architecture:

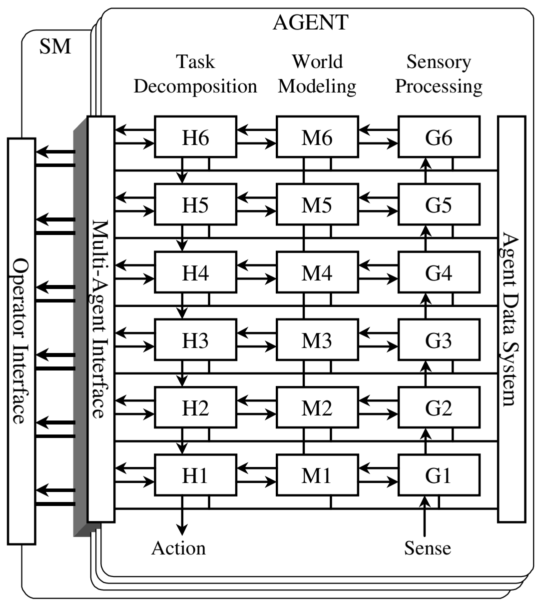

Fig. 4: Generalized NASREM for multiagent telerobot system with multiple operators

The robot system can be homogenous or heterogeneous depended on the physical and logical diversity of the robots. Each robot agent takes individual decision about their movement, thus the control architecture above is partitioned along the robot agents. Each agent shows a similar internal structure to a single telerobot; the fundamental difference is that not a single operator influences its motion but all the other agents and operators may give them commands over the supervisory module. The multiagent interface substitutes the operator interface of the conventional NASREM.

This justifies the modification of the role of the task decomposition modules. On each level they have to implement the appropriate part of the multiagent interface. The communication over this interface is bidirectional, as the agent may transmit information and commands to other agents.

Multiagent behavior schemes shall be partitioned to the appropriate levels of the control hierarchy. Each decision shall be taken based on the level's particular world model.

4.1.2. **Practical use**

I have proven the practical use of the scientific results by creating two completely different telerobot systems with a similar control architecture based on the generalized NASREM. Structure and operation of these systems are shown in chapter 5 of this summary and in chapters 3.8 and 4.6 of the thesis.

4.2. Motion planning of keyframed walking robots

If an operator controls a telerobot it is essential that the robot follows his commands with minimal position error. In a multiagent environment it is also important that the robot agents can compensate their position error, in this way the whole system follows only the commands and it can remain in a consistent state. Numerical trajectory optimization is a way to preserve consistency.

Motion planning of walking robots is a highly researched topic; the results have integrated in several commercial products like Sony AIBO robot dog [43] or Honda's ASIMO humanoid [44]. Parallel, different robotic devices appeared that have different internal structure; controlling these devices required completely different algorithms. The motion trajectory can be defined by discrete keyframes, while the internal trajectory planner processes the linear interpolation between the keyframes. Kondo KHR robots operate according to this model [42].

Chapter 3 of the PhD thesis has the research goal to investigate an efficient off-line trajectory planning and numeric optimization algorithm to this class of walking robots. Tuning of joint parameters and keyframe timestamps of the reference robot motion results in an optimized motion that compensates the inaccuracy of the dynamic effects in a simulated virtual reality environment compared to the kinematic simulation of the reference motion.

One can create a set of off-line optimized perturbated motions, which differ by displacement parameters (step length, robot rotation). They form a correction motion set. The robot can reduce its position error by selecting a correction motion instead of the base reference motion, which has different displacement properties. At the end of the motion the position of the robot will be more precisely close to the desired position.

Thesis 2 [9] [11] [12] [13] – Chapter 3

a) **I have created a new, kinematics-based off-line motion correction method for keyframed walking robots, which compensates the dynamic effects in a simulated virtual environment.**

b) **Based on the generalized NASREM I have suggested a particular control system architecture for multiagent telerobot systems consisting of keyframed walking robots, including a position compensation algorithm that manages the position error of the teleoperated system and uses the off-line motion correction algorithm introduced in section a).**

4.2.1. Keyframed walking robots

There are numerous interactive devices and methods to create motion keyframes, by which ideal kinematical motion can be created. The ideal kinematical model ignores dynamical effects of coming from Newton's physics, so it is not applicable to model a robot control directly. Although virtual forces introduced by Sardain and Bessonnet [35] or ZMP [36] are online trajectory planning methods, which continuously use dynamic model, these methods are not applicable for trajectory planning of discrete keyframed robots.

Teleoperation of keyframed walking robots requires a complex trajectory and path planning process. The robots have to select among the set of correction motions – according to the operators' commands and the sensory feedback. This step will be forwarded to robot motion execution in the next period.

4.2.2. **Trajectory planning in keyframes**

The goal is to create an ideal locomotion on a physical keyframed robot by computing the required joint parameters for a given pose. It can be fulfilled by inverting the kinematic model.

Based on the inverse kinematics technique (IK) presented in the thesis an animator is able to produce the keyframe phases of the desired motion. The ideal motion derives from the linear interpolation in joint space between the motion keyphases using the kinematic model described in the following section.

Kinematic model

Let us assume that our robot has J enumerable joints, we will refer to them $1 \leq j \leq J$. The robot has then (J+1) segments indexed by $0 \leq j \leq J$. For a reference motion we can define K keyframes, when we define the τ_k length of the k-th time interval (→τ vector) between the phases k and (k-1), as well as the $\mathbf{q}_{kj}$ joint angles (→$\mathbf{\Theta}^{[K \times J]}$ matrix).

The robot has a hierarchical structure, there is a root node where j=0. The startup pose of that segment ${}^{0}\mathbf{\Gamma}_0$ which is a 6 component vector in global Cartesian space as 3 position coordinates and 3 Euler-angles describe the pose of an object in 3D space. If we know all joint angles, we can calculate the pose of each segment relative to the root segment.

In kinematical modeling it can be assumed that the pose of a robot segment does not change during the motion phase. This so called “unyielding” object can be one of the feet: it stays always on the ground and every segment moves relatively to it. The index of the unyielding segment is also a parameter of each motion phase, so we introduce the **u** vector with length k for these indices during the whole motion.

If we know the interpolation method between the keyframes, we can formulate a continuous vector-functional, describing the pose of the j^{th} segment in global Cartesian-space:

$$ {}^{j}\mathbf{\Gamma}^{Kin}(t) \equiv {}^{j}\mathbf{\Psi}^{Kin}\left\{\mathbf{\Theta}, \mathbf{u}, \boldsymbol{\tau}, {}^{0}\mathbf{\Gamma}_0\right\}(t), \;\; 0 < j \leq J \tag{1} $$

Aggregating these pose functions to all of the segments results in the aggregate pose vector function that describes the trajectory of all the robot segments in Cartesian space:

$$ \mathbf{\Gamma}^{Kin}(t) \equiv \mathbf{\Psi}^{Kin}\left\{\mathbf{\Theta}, \mathbf{u}, \boldsymbol{\tau}, {}^{0}\mathbf{\Gamma}_0\right\}(t) \tag{2} $$

From now on, $\mathbf{\Psi}^{Kin}$ will be called the kinematic model of the robot.

4.2.3. **Iterative motion correction using dynamic model**

As a consequence of the dynamic effects disregarded in the previous chapter (inertia, impulse) executing the ideal reference motion on a physical robot results in a different robot behavior than the kinematic model foresees. If all the joints can exactly follow the reference signals and the unyielding segments could really be motionless during their active phases, the physical robot motion would exactly follow the ideal reference motion.

Let us assume we have designed a reference motion using the keyframe-technique: apparently this will run only in the ideal virtual world smoothly. Nevertheless this motion represents exactly what we would like to achieve with the real robot (e.g.: stepping forward by a given step length). If the servos could exactly follow the reference signal, $\mathbf{\Gamma}^{Kin}(t)$ would describe the robot state in Cartesian space. Considering the dynamic effects of the real world, there will be some difference between the real and the reference motions (e.g.: the given forward-step will be shorter or longer) – in extreme case the robot might also fall. We present an algorithm that finds a new motion in order that $\mathbf{\Gamma}^{Kin}(t)$ approximates better in real behavior.

Dynamic simulation

In order that the behavior of the physical robot approximates better the reference motion, off-line dynamic modeling of the physical system is required. The model has been built on the results of [37]. The dynamic behavior of a simulated robot can be calculated based on this model.

The numerical optimization method has iterative nature. Similar to (2) the input parameters of the dynamic model in the *i.* iteration define the dynamic behavior of the segments:

$$\mathbf{\Gamma}_{(i)}^{Dyn}(t) \equiv \mathbf{\Psi}^{Dyn}\left\{\mathbf{\Theta}_{(i)}, \mathbf{u}, \boldsymbol{\tau}_{(i)}, {}^{0}\mathbf{\Gamma}_{0}\right\}(t) \tag{3}$$

During each iteration steps $\mathbf{\Theta}$ joint matrix and $\boldsymbol{\tau}$ timing vector will be tuned, the other parameters remain unchanged.

Iterative conformance enhancement

Taking into account a given iteration, the difference of the actual output of the dynamic simulation and the reference motion is also a vector-scalar function:

$$\mathbf{\Gamma}_{(i)}(t) = \mathbf{\Gamma}_{(i)}^{Dyn}(t) - \mathbf{\Gamma}^{Kin}(t) \tag{4}$$

In some cases the position error has higher priority than the orientation error; furthermore it is usually desired to have better compliance on the feet as for example on the head. Each segment has a 6 dimensional $\mathbf{w}_j$ pose-weight vector, so for the whole system the following diagonal weight matrix can be introduced:

$$\mathbf{W} = diag < w_0, w_1, \ldots, w_J > \tag{5}$$

This $\mathbf{W}$ matrix has [(J+1)*6 × (J+1)*6] dimensions. In the i^{th} iteration we can define a weighted error function with help of an inner product based on (4) and (5):

$$\mathrm{E}_{(i)}^{2}(t) = < \mathbf{W} \cdot \mathbf{\Gamma}_{(i)}, \mathbf{W} \cdot \mathbf{\Gamma}_{(i)} > \tag{6}$$

Using this error function we want to define a χ norm, in order to have a non-negative scalar value that represents the correspondence between the reference motion and the simulated one in time (considering the whole simulation length is T and the i^{th} iteration):

$$\chi_{(i)}\left\{\mathbf{\Theta}_{(i)}, \mathbf{u}, \boldsymbol{\tau}_{(i)}, {}^{0}\mathbf{\Gamma}_{0}\right\} = \int_{0}^{T} \mathrm{E}_{(i)}^{2}(t) \cdot dt \tag{7}$$

One can easily see that all members $\chi_{(i)}$ of the series χ define a norm for the $\mathbf{\Gamma}_{(i)}(t)$ functions, which form a Hilbert space over this norm. As we want to lower this norm, we need to alter $\mathbf{\Theta}_{(i)}$ and $\tau_{(i)}$ input parameters each step, and leave the other input variables constant. During the steps some boundary conditions (initial and final servo configurations, motion duration, etc.) have to be fulfilled.

We would like to have a monotonous descending series of χ_i elements to reach the optimal input keyframes:

$$(\mathbf{\Theta}_{opt}, \boldsymbol{\tau}_{opt}) = \underset{(\mathbf{\Theta}, \boldsymbol{\tau})}{\arg\min}\, \chi\left\{\mathbf{\Theta}, \mathbf{u}, \boldsymbol{\tau}, {}^{0}\mathbf{\Gamma}_{0}\right\} \tag{8}$$

In the next section we present a numerical solution for this problem.

New numerical optimization method

The goal of the numerical correction algorithm is that the dynamical behavior approximates the reference movement.

Let us define $\boldsymbol{D}$ as a subset of input parameter domain $\mathbf{x}_0 := (\mathbf{\Theta}_{Kin}, \boldsymbol{\tau}_{Kin})$ with the following properties: $\boldsymbol{D}$ shall be the largest connected subset that fulfills all the boundary conditions and contains all the motion parameter combinations where the robot does not fall down and contains the reference movement.

Before applying any numerical optimization method the following presumptions must be taken

- $\chi(\mathbf{x})$ is continuous and differentiable over $\boldsymbol{D}$
- close to the boundaries of $\boldsymbol{D}$ the negative gradient $-\nabla\chi$ points inwards,
- inside of $\boldsymbol{D}$ exists an element $\mathbf{x}_{opt} = (\mathbf{\Theta}_{opt}, \boldsymbol{\tau}_{opt})$, we call it optimum. This is not necessarily global optimum it has the property in its infinitesimal range $\chi(\mathbf{x}) \geq \chi(\mathbf{x}_{opt}), \mathbf{x} \in \boldsymbol{D}$

It is clear that the final motion will not match the reference motion. It might be impossible to follow it physically due to ignored dynamics. Therefore, its norm will clearly not reach zero at the end of the optimization. In most cases a local optimum is reached where the gradient is zero.

$\chi(\mathbf{x})$ is a strongly non-linear function of its input parameters with narrow potential tunnels, therefore I decided to implement the non-linear conjugated gradient method described in [8].

For this method we needed the gradient of the potential function. We computed it numerically in a component-by-component way using partial differentials. For the i^{th} component:

$$\nabla\chi(\mathbf{x})_i = \left[\frac{\chi(\mathbf{x}+\Delta\mathbf{x}_i) - \chi(\mathbf{x})}{|\Delta\mathbf{x}_i|}\right], \qquad (9)$$
$$\Delta\mathbf{x}_i = \pm\varepsilon * \mathbf{e}_i$$

where $\mathbf{e}_i$ is the i^{th} basis vector of the input space. As $\boldsymbol{D}$ is bounded, it can happen that a $\mathbf{x}+\Delta\mathbf{x}_i$ vector points outside of $\boldsymbol{D}$. It is very uncommon as the negative gradients of $\boldsymbol{D}$ at the boundaries point inwards, so the algorithm does not approach the boundaries. If it is still the case, one has to take the inverse of $\Delta\mathbf{x}_i$. In an extreme case if it still points outside of $\boldsymbol{D}$, then $\boldsymbol{D}$ is very thick along that dimension, most likely the algorithm has reached an extremity of $\boldsymbol{D}$. The obvious opportunity is to renounce the derivative in this direction at the particular step. In this iteration the dynamical simulation has to be executed ($J*K+K+1$) times.

At initial phase, the algorithm has to perform a line search along the direction of steepest descent. It is an iterative method that should find the minimum along this line. It is a one-dimensional search method. The result is $\mathbf{x}_1$. Then, the algorithm consists of 5 steps:

1. Compute the gradient in the actual position: $\mathbf{x}_n$,
2. Compute β_n according the, Polak–Ribière formula:
$$\beta_n = \max\left[\frac{\nabla\chi(\mathbf{x}_n)^T * (\nabla\chi(\mathbf{x}_n) - \nabla\chi(\mathbf{x}_{n-1}))}{\nabla\chi(\mathbf{x}_{n-1})^T \nabla\chi(\mathbf{x}_{n-1})}, 0\right] \qquad (10)$$
3. Compute the next conjugated direction:
$$\Lambda x_0 = \mathbf{0}, \qquad \Lambda x_n = \nabla\chi(\mathbf{x}_n) + \beta_n \Lambda\mathbf{x}_{n-1} \qquad (11)$$
4. Perform a line search along the last conjugated direction:
$$\min_{\alpha_n} \chi(\mathbf{x}_n + \alpha_n * \Lambda\mathbf{x}_n) \qquad (12)$$
5. Next iteration will be then:
$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_{n*}\mathbf{\Lambda x}_n \qquad (13)$$

The algorithm needs sometimes to be restarted in terms of the conjugated directions, that is, setting the conjugated direction along the negative gradient. This should happen if $\alpha_n > 0$ or after reaching a given number of iterations. The algorithm ends if the gradient sinks below a given threshold.

Notes

The stability of the optimized motion cannot be guaranteed explicitly, it depends strongly on stability properties of the reference motion designed by kinematical methods. The algorithm described above can be used if the reference motion is stable on the real robot. The method cannot be used if the robot is exposed to dynamic effects during the execution of the reference motion, i.e. the robot slips or turns over.

In general, the method always improves on the dynamic stability of the motion. It reaches the global optimum if it exists in $\boldsymbol{D}$ and there is a continuous curve between the reference motion and the optimum where χ norm has negative derivative.

4.2.4. Teleoperation of physical robots

Thesis 2 b) describes an algorithm that helps integrating robots defined in introduction of chapter 4.2 into a multi-agent telerobot system like reported in [9]. In the followings, system presumptions will be summarized, which are required to use the kinematic simulation and the motion compensation algorithm.

Multiagent telerobot system for keyframed walking robots

Keyframing implies that the robot controller hides the **servo level** controlling functions. In order to build up the dynamic model of the mechanical system one has to know the servo dynamics.

Static robot state is represented by the joint parameters including the state of the actuators. Assuming linear interpolation in joint space a **motion primitive** can be defined by the actual state, the target state and the length of the period required state transition, that is, to every motion primitive one can assign a $\mathbf{x} := (\mathbf{\Theta}, \tau)$ vector that contains the target state and the length of the transition. Consequently, the function of the motion primitive level can be modeled with a directed state graph where the nodes are robot states and the directed edges describe the possible state transitions. This level also translates valid state transitions into reference signals to the servos.

Elemental motions consist of a set of consecutive motion primitives, which cannot be interrupted. These can be steps or even more complex running motions. An elemental motion can be trained in an off-line way. E-Move level is responsible for the coordinated execution of elemental motions.

Task level decides which elemental move should be chosen to fulfill a particular task. Tasks can be modeled with subsequent elemental motions with a set of conditions. Execution of a motion can be modeled with kinematic simulation.

Operator can intervene into robot behavior on both E-move and task levels. On E-move level he can submit a new **x** vector to the robot that results in changes in robot motion graph, the robot can perform a motion towards another new state. On task level the operator can initiate another elemental motion.

Observing the other agents in the manipulation area the robot agent can decide on both control levels how to consider and interpret operator commands, that is, which motion primitive or elemental move it chooses. The robot may disregard operator controls to avoid collisions with another robots or objects in the manipulation space.

Position error

The problem formulates as follows: executing an ideal reference motion on a physical robot it results in difference in segment poses compared to the kinematic simulation:

$$\mathbf{\Gamma}(t) = \mathbf{\Gamma}^{Phys}(t) - \mathbf{\Gamma}^{Kin}(t) \tag{14}$$

The difference has two reasons:

- the dynamic model introduced in (3) approximates the real dynamics with simplifications like neglecting friction, knocking as well as calibration errors and numeric inaccuracy. They all result in that the dynamic simulation produces deviation from the behavior of the real robot;
- the error function introduced in equation (6) quantizes the deviation between the dynamic and the kinematic simulation. In real scenario it won't be zero at *T*, due to this remainder difference in the robot pose at the end of every locomotion will be different at the two models, consequently the dynamic simulation deviates from the ideal reference motion as well.

This pose error cumulates, thus the agent has to compensate this with sensor coupling. Correction is possible always at the next locomotion (next step). The problem is assigned to the **task level** as it is responsible for choosing the appropriate reference motion. The decision shall be taken according to the measured pose error and the initiated locomotion. The task decomposition module shall choose a perturbated variant of the reference motion that brings the robot next to the desired reference position.

For this decision every locomotion requires a previously off-line generated correction motion set that differ by kinematic parameters (i.e. step length, relative rotation) by a desired granularity. These motions shall be optimized according to the thesis 2. a) in order their dynamic execution approximates well the kinematic simulation.

Sensor feedback

Calculating the position and orientation error the agent has to know the kinematic model of the whole manipulation area, so that it can simulate the expected results of the operators' commands and its own pose. On the other hand the sensor system has to be able to measure the actual robot pose in a desired accuracy.

From system point of view the system designers have two opportunities. Either the agent does this correction itself or it communicates its actual pose and the motion opportunities to the central supervision system and it is responsible for robot positioning.

Whichever task decomposition module does this pose correction it has to choose among the available correction motions. It shall choose the one that minimizes the a posteriori pose error after execution of the selected motion.

If the expected robot pose is estimated exclusively based on the kinematic model, the pose error remains static. If the system is capable of measuring the physical dislocation parameters of the execution of the various locomotions, then the stationary component of the pose error disappears.

4.2.5. **Usage of the scientific results**

The two algorithms described above are applied in a multiagent humanoid teleoperation system introduced in chapter 5.1 of this summary. The application is described in chapter 3.8 in the dissertation in more detail.

4.3. Depth estimation of a blurred object

Continuously discovering the environment is indispensable in intelligent semi-autonomous multiagent telerobot systems. In case of human supervisory control this state information shall be provided to the human operator(s) as well. Using cameras is very common due to usability reasons.

For the robot point of view the camera image is information source, from which the state of the robot and of a part of the manipulation area can be acquired, thus it is also very common to use machine vision and image processing algorithms in telerobotics. In systems based on the generalized NASREM model such algorithms are located in the sensory processing components indicated with G.

Thesis 3 [7] [8] – Chapter 4

I have elaborated a new 1D DFT-based depth-from-defocus spatial position estimation algorithm for long, narrow planar objects, which requires the sharp object model and the blurred image. The algorithm fits in the sensory processing modules of the generalized NASREM control system hierarchy introduced in the 1st thesis.

4.3.1. Introduction

Depth-from-defocus

Vision system designers often face the problem of extracting 3D position information from a 2D CCD camera image. Robot sensor coupling often leads to this positioning problem, especially in telerobotics, where visual feedback is indispensable. The biggest challenge in most cases is automatic depth coordinate quantification, that is, measuring the distance of the object from the camera plane.

There are machine vision systems realizing a sensor fusion of cameras and other distance measuring sensors. Machine vision may improve measurement accuracy even at these systems but especially at those where the only information of the manipulation area are only camera images available.

Visual sensing may happen in active or passive way. Active systems estimate object distance by changing the focal length of the optics in order to acquire sharp image. Object distance can be calculated from the optics state.

Passive vision systems calculate object distance from blur. The inherent shortcoming of this method is that it is impossible to decide whether the object is located in front of or behind the focal plane. Current realizations use two simultaneously acquired images with different focused optics. That requires special, expensive image acquisition optics, which is capable of taking two images parallel at the same time and same perspective but at different focal distances.

The basics of depth from defocus (DFD) methods were developed by Penteland (1987) [39], and used a direct relationship between camera parameters, depth and different levels of blur in several images. The advantage of this method opposed to other depth estimation techniques is its fairly reliable accuracy at a low computational cost.

The basic theory behind DFD is that blur can be modeled as a linear convolution of the sharp image and a Gaussian point spread function (PSF). The sigma parameter of the PSF is proportional to the object's distance from the camera

The estimation can occur in spatial [40] or in frequency domain [41]. As convolution in spectral domain transforms to a simple multiplication, recent publications deal with 2D Discrete Fourier analysis of blurred images.

Another serious shortcoming of 2D DFT algorithms is that they can estimate only a single distance parameter, thus, they are applicable only for a single object, which lies parallel to the focal plane, that is, they cannot utilize the continuous changing blur on skew objects for improving the estimation accuracy.

The algorithm I suggested has two desirable properties, as it is applicable for objects not completely parallel to the focal plane and it requires a single grayscale image taken from the scene and another one taken from the sharp scene; so it can extract adequate 3D coordinates using conventional apparatus; it does not need special expensive optical devices.

4.3.2. **Depth estimation with an 1D DFT algorithm**

Consider a mirror-symmetric, long convex planar object in the space. Tip is defined as the intersection of the object axis and the boundary. The problem to be solved is to estimate the coordinates of the tip and the angle between the focal plane and the object axis.

The algorithm builds upon the following assumptions:

- tip is visible in camera image;
- the mentioned object can be separated from other objects in the camera image with help of segmentation algorithms;
- the object does not overlap with other objects;
- without restriction of generality one can assume that the object axis coincides with the x axis of the image. If it would not hold, it can be assured during the preprocessing phase;
- the y object coordinate (perpendicular to the axis) is parallel to the image plane;
- sharp image of the object is already known and is invariant to a translation along the x axis; this can be taken at calibration time;
- in the camera image one can find an additive white Gaussian noise. Their parameters can be estimated at the calibration phase.

The following diagram illustrates the object and its environment from the camera point of view.

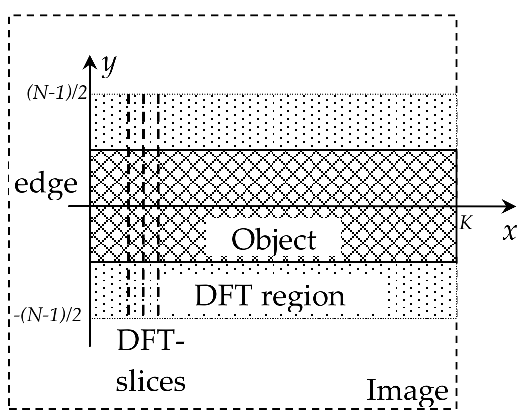

Fig. 5: Coordinate axes in the camera image and the Fourier transformation domain

z axis of the image coordinate system is perpendicular to the focal plane and is zero at the distance of sharp vision. The blur at the object boundaries provides information about object depth. Calculating the discrete Fourier Transformation (DFT) in a slice in y direction one can set the following equation:

$$\Im_z^* = \Im_0 \mathcal{H}_g + \mathcal{F}\{\eta_z\} \tag{15}$$

where $\Im_0$ denotes the Fourier transform of the sharp image without any additive noise. Because of the invariance on object translation it is same for all *x* values. $\Im_z^*$ is the discrete Fourier transform of the noisy camera image taken at *z* depth, $\mathcal{H}_g$ denotes the discrete Fourier transform of the Gaussian point spread function (PSF) $h_g(\mu_S, \sigma_S)$. In frequency domain it is also Gaussian: $\mathcal{H}_g(\mu_F, \sigma_F)$, their parameters can be calculated from the parameters of the spatial PSF:

$$\mu_F = \mu_S = 0\,,\ \sigma_F = \frac{N}{2\pi * \sigma_S} \tag{16}$$

$\mathcal{F}\{\eta_z\}$ denotes the measurement noise in frequency domain. The equation (15) describes the artificial reproduction of the camera image spectrum from a-priori information and the object distance from the plane of sharp vision.

The estimation simplifies to the following problem: the best fitting $\mathcal{H}_g(0, \sigma_F)$ function has to be founded that blurs the known sharp image so that the result matches the actual camera image. From the identified parameter of the best fitting point spread function one can calculate object depth in a straight forward way base on the following relation:

$$z = \lambda * \sigma_S = \lambda \frac{N}{2\pi * \sigma_F}, \tag{17}$$

As the object has ϑ skewness relative to the focal plane the blur is a linear function of *x*:

$$\sigma_S(x, \sigma_{S0}, \vartheta) = \sigma_{S0} + \lambda^{-1} x * \tan\vartheta \tag{18}$$

where σ_{S0} means the blur function at the tip of the object. The transformed blur function can be formulated from the relations above:

$$\sigma_F(x, \sigma_{S0}, \vartheta) = \frac{N}{2\pi(\sigma_{S0} + \lambda^{-1} x * \tan\vartheta)} \tag{19}$$

The algorithm has to be invariant to the illumination. Contrast and brightness result in a linear transformation for pixel grayscale values. In spectral domain the brightness is represented in the 0^{th} coefficient the relative contrast ratio causes a constant *k* multiplication factor in each coefficient.

Equation (15) that describes the spectrum of the artificially reproduced image will be modified in the following way to compensate the disturbing effects introduced above:

$$\Im_z^*(x, f) = k^{-1} \Im_0^*(x, f) \mathcal{H}_g(\sigma_F(x, \sigma_{S0}, \vartheta), f) \tag{20}$$

where $\Im_0^*$ is the noisy sharp model, *f* is the independent variable of the discrete Fourier transform in *y* direction, *x* is the distance from the tip in pixels, and *k* is the relative contrast ration between the sharp model and the actual light conditions. The actual acquired and the generated image will be different. The difference is described by the following least squares cost function:

$$C(\sigma_{S0}, k, \vartheta) = \sum_{x=0}^{K-1} \sum_{f=1}^{N} \left\| \mathcal{H}_g(\sigma_F(x, \sigma_{S0}, \vartheta), f) \Im_0^*(x, f) \right| - k \left| \Im_z^*(x, f) \right\|^2 \tag{21}$$

In order to be invariant against illumination changes the cost function does not contain the DC component, and considers only the absolute value of the Fourier-coefficients. The reason is that the phase may differ significantly due to numerical inaccuracies, while the absolute value is more robust against the inaccurate correspondence of matching DFT-slices.

The cost function has also three parameters, its value can be computed at every parameter combination, so its minimum can be found using numerical algorithms, which does not require the cost function to be zero at minimum place. In example Newton method or conjugate gradient methods can be successfully applied.

After finding the minimum place of the cost function, the object depth, the contrast ratio and the object skewness can be calculated directly from the identified parameters, the first one with help of equation (17).

4.3.3. Usage of scientific results

In praxis the presumptions of the algorithm detailed above cannot be fulfilled completely. In practical application there are numerous disturbing effects originated from contravening the a priori requirements. Chapter 4 of the thesis goes in detail how to apply this algorithm for position estimation of a cylindrical micro manipulator using a single microscope, it deals with the practical problems arising in real application and describes their solution.

5. Using results

5.1. EMSER – humanoid teleoperation platform

The following multi-agent, multi-operator telerobot system serves as an example for the application of the first and second thesis in real systems. In this certain case study, the results explained in the first two theses are used in a platform consisting of KHR-1 humanoid robots.

The core of the system, which has been built on the architecture of the 1st thesis, is a server side virtual reality (VR). There are multiple client side physical teleoperation platforms to be synchronized with this core. The VR represents the supervision module, which processes the operators' commands. The output of the events initiated by the robots' model in the server side world model is the reference movement that the physical platforms have to execute.

Each reference agent in the VR is controlled by an operator. The telerobots of each platform are the physical manifestation of the corresponding agents. In accordance with the 2nd thesis, the task decomposition modules of these agents regulate the client side telerobots with movements optimized based on the dynamic model of each platform. So the way of execution is the own decision of the agent's task level controlling module in line with the position improving algorithm included in the 2nd thesis.

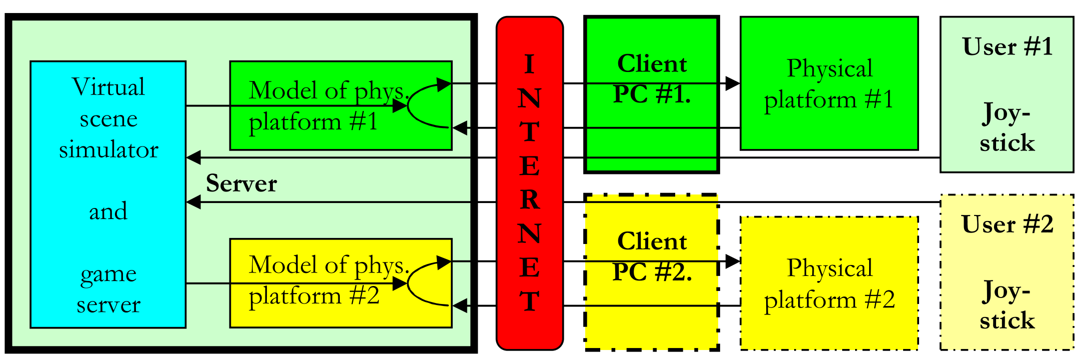

Fig 6: Application of the 2nd thesis in a multi-agent, multi-operator telerobotic system

A collision of the physical robots can be prevented using a preliminary analysis made at service level of the supervision module, and the probable time of the collision can be predicted in virtual reality. The task level of the supervision module decides what is to be done in each collision if initiating a motion resulted in such a scenario. There are two possibilities:

- If the collision happens during a locomotion, the supervising module will not allow moving;
- During a non-locomotion the task level module of the supervising module produces a new elementary movement using kinematical simulation. This

movement is the same as the initiated motion up to the time of the collision, and after that it is the same motion but reversed in time. Being a non-locomotion this does not pose a risk to the stability of the robot.

The manually generated ideal and correctional locomotion of the robots was optimized according to the 2. a) thesis. As an example, four variants of the ideal elementary movement of a step forward consists of K=9 motion primitives (keyframes). Due to the J=21 joint variable of the robot, **x** is a vector of $K(J+1)$ = 198 dimensions. The algorithm was run based on the recommendation of the conjugated gradient up to this step. The following figure shows the relative changing of χ. The values of χ are normalized by its initial value in t=0:

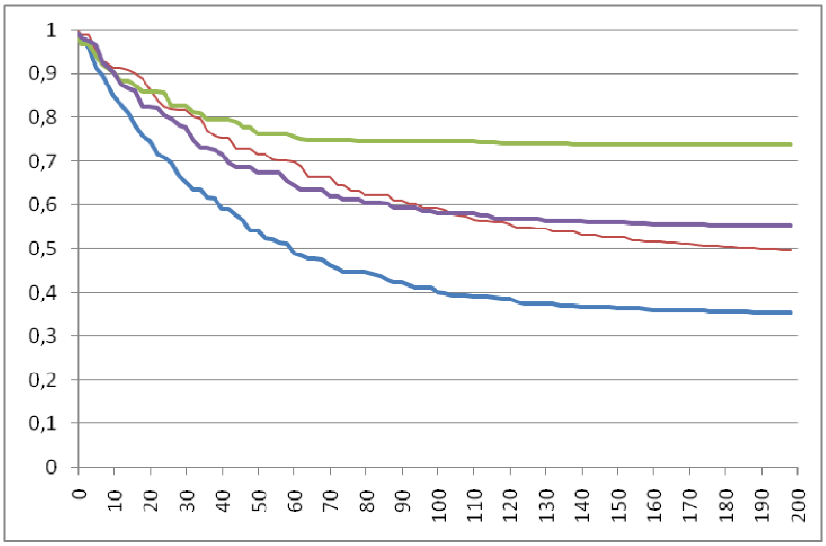

Fig. 7: A The norm function during the optimization of four locomotions

We can see that the algorithm usually finds the neighborhood of minimum in as many steps as the number of dimensions, while the error norm decreases monotonously. Due to numeric inaccuracies and the fact that the cost function is not quadratic, the algorithm had to be restarted at certain intervals, which means that by setting the conjugated directions to null the robot had to move again in the direction of the gradient. After all, the algorithm could reduce the error norm by 25-65%, meaning that the resultant motion approximated the kinematically planned reference motion almost three times better at best by compensating dynamic effects.

5.2. A MINISTER micromanipulation work cell

Within the scope of a joint research project, the Department of Control Engineering and Information Technology of BME and the Institute for Process Control and Robotics (IPR) of the Karlsruhe University set up the MINISTER micromanipulation workstation for researching the positioning of microscopic materials with high accuracy.

The work space of MINISTER is a 200x300mm glass plate, in which the 30x30x60mm robot can move in 3 degrees of freedom. The position of the robot is evaluated by a two-level visual sensor system. The task of the camera that covers the whole work space is to evaluate the position of the robot with millimeter accuracy, while the other camera watches a smaller area through a microscope providing micrometer accuracy. Both cameras are connected to a PC, which is responsible for processing and evaluating the image.

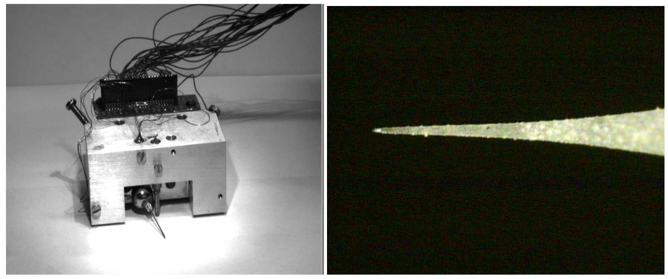

Fig. 8: The MINISTER micro robot and its manipulator

I published about the application of the MINISTER microrobot in teleoperation environments first in a scientific students' association's conference [14], then later in Híradástechnika [1]. In another scientific students' associations conference [15] I presented a suggestion for an autonomous multi-agent microrobot system without supervision, in which several robots' actions have to be harmonized; I verified my theoretic results with simulations.

The possibilities for the communication between the telerobot and the operator's system are explicated in [2] and [3]. This teleoperation work cell was also used in an experimental laboratory measurement [10]. In the INES2002 conference [4] I made a proposal to the two-level hierarchical controlling of workstations which are equipped with hierarchical sensor systems.

My first thesis about generalizing the NASREM six level hierarchy to multi-agent systems was published at INES2003 [5] and the FRAKAD2003 conferences [6]. The experiment platform was the MINISTER workstation in both cases. The intelligent control of the robots was established by a distributed algorithm implemented in a central supervisor computer, while – since lacking a physical model – results were confirmed with simulations.

The results of my first thesis proved useful also in an environment of a very different nature, namely, in the above presented EMSER platform [9].

<u>Depth estimation of the MINISTER's manipulator</u>

The MINISTER's camera, which is fixed to a microscope, has very small depth of focus due to huge magnification, thus, it meets the preconditions of the third thesis. I implemented the algorithm and as an experiment I also implemented the "traditional" depth estimation method, which can be found in technical literature, and which is based on a two-dimensional Fourier transformation. I evaluated the results of the experimental running using regression analysis methods. The accuracy of the algorithms can be described by the standard deviation of estimation residua. The table below summarizes the experimental standard deviation of the residuum distributions, which come from the estimations made on teaching (R) and verification (V) patterns and the difference of these estimations.

Table 1: The accuracy of the one-dimensional and the standard two-dimensional depth estimation algorithm

	$\sigma(z-\hat{z}_R)$	$\sigma(z-\hat{z}_V)$	$\sigma(\hat{z}_V-\hat{z}_R)/\sqrt{2}$
Algorithm based on 1D DFT of the thesis 3.	8.75	6.07	5.47
Standard 2D DFT algorithm	14.83	14.73	12.73

The table above, the 4.6 section of the thesis, and the [7] and [8] publications demonstrate that the depth estimation algorithm I suggested, which is based on one-dimensional DFT, provides more robust and accurate results in the examined case than other depth estimation algorithms (found in technical literature) based on similar but two-dimensional DFT. Using the former one the depth coordinate of the manipulator's tip can be estimated with approximately half as big error as with similar earlier methods.

Related publications

[1] Urbancsek T., Lassó A.: *Mikromanipulátorok teleoperációs alkalmazása biotechnológiai kutatásokban*, Híradástechnika, HU ISSN 0018-2028, 2000/11. pp. 36-43.

[2] Lassó A., Urbancsek T.: *Communication architectures for web-based telerobotic systems*, MED '01, Dubrovnik, Croatia, Section: Networks and Internet, 2001.06.28, Proceedings of 9th IEEE Mediterranean Conference on Control and Automation, ISBN-953-6037-35-1, Paper no. 068 (4 pages)

[3] Urbancsek T., Vajta L.: *Ferngesteuerte mikrorobotische Manipulationszelle für sichere biotechnologische Forschungen*, München, 2002.04.30, Wissenschaftliche Mitteilungen der 14. Frühlingsakademie, Budapest, 2002, ISBN 963 86234 5 4, pp. 114-120

[4] Urbancsek T., Vogel M.: *Sensing and Monitoring Methods in Telerobotical Applications Fusing Real-time Video Transmission and Calibrated Virtual Reality*, INES 2002, IEEE proceedings pp. 107-110, Opatija, Croatia, 2002.

[5] Urbancsek T., Vajda F.: *Internet Telerobotics for Multi-Agent Mobile Microrobot Systems - A New Approach*, Assiut - Luxor, Egypt, 2003.03.05, IEEE Proceedings of the INES 2003 conference, ISBN 977-246-048-3, ISSN 1562-5850, pp. 462-466.

[6] Urbancsek T., Vajta L.: *Erweiterung des NASREM Referenzmodells für Multi-Agenten Telerobotersysteme*, Balatonfüred 2003.05.17, Wissenschaftliche Mitteilungen der 15. Frühlingsakademie, pp. 71-77, Budapest 2003, ISBN 963 214 1180, pp. 71-77

[7] Urbancsek T., Garay Cs., Vajta L.: *3D Spatial Coordinate Estimation of a Micro Manipulator From a Monocular Camera Image Using DFD Algorithm*, International Conference on Intelligent Manipulation and Grasping, Genova, Italy, 2004.07.01., Proceedings of the IMG04 Conference, ISBN 88 900 426-1-3, pp. 337-342.

[8] Urbancsek T., Vajta L.: *Ein Maschinensichtsystem für Tele-Mikrorobotik – 3D Positionsschätzung eines beinah zylindrischen Objekts aus einem einzelnen Kamerabild*, München 2004.05.19, Wissenschaftliche Mitteilungen der 16. Frühlingsakademie, Budapest, 2004, ISBN 963 86697 0 5, pp. 118-129. (Német nyelven)

[9] Dálnoki Á., Helybély Á., Juhász T., Kálmán V., Urbancsek T., Vajda F., Vajta L.: *Rendszer és eljárás valós objektumok mozgatására virtuális térben végrehajtott műveletekkel*, Lajstromszám: P0600879, Közzététel éve: 2006

[10] Arató P., Vajta L., Urbancsek T., G. Chamilothoris: *Design of an Internet-Based Tele-Microrobotical Laboratory Assignment*, Tallinn, 2006, IEEE International Conference on Computational Cybernetics, ISBN: 1-4244-0071-6 IEEE Catalog Number: 06EX1270, pp. 97-100.

[11] Vajta L., Vajda F., Urbancsek T, Juhász T., Kálmán V., Vogel M., Helybély Á.: *EMSER* – Kutatási jelentés, 3DMR – IIT, BME, 2007. jan. 23.

[12] Juhász T., Urbancsek T.: *Beyond the limits of kinematics in planning keyframed biped locomotion*; Proc. of 6th Eurosim Congress on Modelling and Simulation, 9th-13th September 2007, Ljubljana, Slovenia, Vol-1 (abstracts), p. 78.

[13] Juhász T., Urbancsek T.: *Beyond the Limits of Kinematics in Planning Keyframed Biped Locomotion*, Periodica Polytechnika (elfogadva)

Other publications

[14] Lassó A., Urbancsek, T.: *Mikromanipulátorok teleoperációs alkalmazása biotechnológiai kutatásokban*, BME VIK TDK dolgozat, 2000, OTDK 2001, p. 70.

[15] Urbancsek T., Helybély Á.: *Viselkedésorientált vezérlésű miniatürizált robotrendszerek elsősorban mikrobiológiai alkalmazásra*, BME VIK TDK dolgozat, 2001, OTDK 2001, p. 68.

[16] Lassó A., Urbancsek T., Helybély Á.: *Microrobot Teleoperation through WWW*, microCAD 2001 Miskolc, 2001. március 1-2, Section J 'Gábor Dénes' Applied Information Engineering, Proceedings of MicroCad 2001 International Scientific Conference ISBN 963 661 457 1, ISBN 963 661 467 9, Miskolc, 2001 pp. 133-138.

[17] Urbancsek T. (szerk.): *Wissenschaftliche Mitteilungen der 14. Frühlingsakademie*, München, Németország, 2002.04.30-2002.05.05., Logod Verlag, 2002. 134 p.

[18] Jäkel H., Urbancsek T., Thierer C.: *A Note on the Calculation of a Discrete-Event-System's Transfer Function*, Hänsel-Hohenhausen, Frankfurt., 2002.05.30, ISBN-10: 3826712110, ISBN-13: 978-3826712111, (Ed: U. Kiencke, K.Dostert), pp 61-66.

[19] Vajda F., Urbancsek T.: *High-Level Object-Oriented Program Language for Mobile Microrobot Control*, Assiut - Luxor, Egypt, 2003.03.05, IEEE Proceedings of the INES 2003 conference, ISBN 977-246-048-3, ISSN 1562-5850, pp. 458-461.

[20] Urbancsek T. (szerk.): *Wissenschaftliche Mitteilungen der 15. Frühlingsakademie*, Balatonfüred, 2003.05.17-2003.05.23., Logod Verlag, Budapest, 2003. 82 p.

[21] Urbancsek T. (szerk.): *Wissenschaftliche Mitteilungen der 16. Frühlingsakademie*, München, Németország, 2004.05.19-2004.05.23., Logod Verlag, Budapest, 2004. 138 p.

[22] Urbancsek T. (szerk.): *Wissenschaftliche Mitteilungen der 17. Frühlingsakademie*, Balatonfüred, 2005.05.04-2005.05.08., Logod Verlag, Budapest, 2005. 156 p.

[23] Urbancsek T.: *Neue Algorithmen in der VR-Basierten Internet Telerobotik*, Balatonfüred 2005.05.04, Wissenschaftliche Mitteilungen der 17. Frühlingsakademie, Budapest, 2005, ISBN 963 86697 3X pp. 129-131.

[24] Vajta L., Urbancsek T., Vajda F., Juhász T.: *Comparison of Different 3D (Stereo) Visualization Methods - Experimental Study*; Proc. of 6th Eurosim Congress on Modelling and Simulation, 9th-13th September 2007, Ljubljana, Slovenia, Vol-1 (abstracts), p. 351

[25] Urbancsek T., Vámos G.: *Szakértői rendszer készítése Video-On-Demand tartalmak kölcsönzésének előrejelzésére felhasználói profil-képzés technikájával*, Kutatási jelentés, IIT, BME, 2007. nov. 4.

Literature

[26] Nagy I, Vajta L, Gürtler Cs.: *Vision Based Navigation for Multi-agent Mobile Robot System in the Transport Domain*, Proceeding of the INES 97,. Budapest, Magyarország, 1997

[27] Arató P, Vajta L, Felső G.: *Visual Sensoric for Microrobot Systems*, Proceedings of the IIIS World Multiconference on Systemics, Cybernetics and Informatics. Caracas, Venezuela, 1997.06.07-1997.06.11., pp. 1-5.

[28] Tél F, Kahlesz F, Lantos B: *Intelligent stereo vision system for robot control*, Proc. 9th Mediterrian Conference on Control and Automation MED01. Dubrovnik, Horvátország, 2001, pp. 1-6.

[29] Vajda F.: *Optimal Landmark Arrangement and Programming in the Field of Heterogeneous Multiagent Microrobotic Systems*, BME PhD disszertáció, 2006

[30] Felső G.: Mikromanipulációs robotrendszerek modellezése és vizuális alapú pozícióimérési eljárásai, BME PhD disszertáció, 2007

[31] Albus J. S, McCain H. G., Lumia R.: *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*, NBS Tech Note 1235.

[32] Sheridan T.B.: *Telerobotics, automation and human supervisory control.* Cambridge, MA: MIT Press., 1992

[33] Arkin, R. C.: Behavior-Based Robotics, MA. MIT Press., 1998

[34] Ali K. S.: *Multiagent Telerobotics: Matching Systems to Tasks*, PhD Thesis, GIT, 1999

[35] Sardain P., Bessonnet G.: *Forces acting on a biped robot. Center of pressure-zero moment point*; IEEE Transactions on Systems, Man and Cybernetics, Part A, Volume 34, Issue 5, Sept. 2004, pp. 630-637.

[36] Vukobratović M., Borovac B.: *Zero-moment point - Thirty five years of its life.* International Journal of Humanoid Robotics, Vol. 1, No. 1, pp. 157-173, 2004

[37] Juhász T., Schmucker U.: *Automatic Model Conversion to Modelica for Dymola-based Mechatronic Simulation*; Proc. of 6th International Modelica Conference, 3rd-4th March, 2008, Bielefeld, Germany, Vol. 2, pp. 719-726.

[38] Wikipedia*: Nonlinear conjugate gradient method,* http://en.wikipedia.org/wiki/Nonlinear_ conjugate_gradient

[39] Penteland A. P.: *A new sense for detph of field.* IEEE Transactions on PAMI, vol. 9, No.4, 1987, pp. 525-531

[40] Wei T.: *Three Dimensional Machine Vision Using Image Defocus*, PhD Dissertation at Stony Brook University, NY, 1994

[41] Ghita O., Whelan P. F.: "*Real time 3-D estimation using detph from defocus*", Proc. Irish Machine Vision and Image Processing Conference, Dublin, 1999 pp. 167-181.

[42] Kondo KHR series product page: http://www.kondo-robot.com/EN/wp/?cat=10

[43] AIBO: http://www.sony.net/SonyInfo/News/Press_Archive/199806/98-052

[44] ASIMO: http://asimo.honda.com/asimo_specifications.html

[45] Dymola, DynaSim (Dassault Systemes) Lund, Svédország: http://www.dynasim.se

[46] Modelica Association – http://www.modelica.org