In: 37th International Symposium on Robotics (ISR 2006) /
4th German Conference on Robotics (Robotik 2006)
München, Germany May 15 to 17, 2006

ENVIRONMENT GUIDED HANDLING OF DEFORMABLE LINEAR OBJECTS: FROM TASK DEMONSTRATION TO TASK EXECUTION

Jürgen Acker	Björn Kahl	Dominik Henrich
University of Bayreuth	University of Bayreuth	University of Bayreuth
Chair for Applied Computer Science III	Chair for Applied Computer Science III	Chair for Applied Computer Science III
(Robotics and Embedded Systems)	(Robotics and Embedded Systems)	(Robotics and Embedded Systems)
Germany	Germany	Germany

Speaker: Prof. Dr. Dominik Henrich, University of Bayreuth, Chair for Applied Computer Science III (Robotics and Embedded Systems), D-95440 Bayreuth, Germany, dominik.henrich@uni-bayreuth.de

Topic: Session D2: Flexible manufacturing systems

Keywords: manipulation, deformable linear objects, programming by demonstration, program generation

Abstract: The research in the field of assembly operations mainly focuses on the handling of rigid objects. Although non-rigid objects like ropes, wires, or steels springs must be handled as well. Therefore, our research investigates the sensor-based handling of deformable linear objects in the field of assembly operations. In this paper, we discuss guidelines for a task description that is able to cope with uncertainties and we analyse the restrictions of such a task description. Also we present the process of the automatically generation of the task description from a demonstration in a virtual environment, performed by a human operator.

Introduction

The development and research in the field of assembly operations mainly focuses on the handling of rigid objects. But in industrial assembly tasks non-rigid objects like ropes, wires, or steel springs must be handled as well. Such *deformable linear objects* (DLOs) are usually handled by human workers since the inherent uncertainties concerning the exact shape of an individual DLO and the high number of degrees of freedom make the handling difficult. Thus, sensor based strategies are necessary to carry out assembly tasks with DLOs involved. Therefore, our research goal is the exploration of such strategies for the handling of DLOs in the field of assembly operations.

Most research activities in the field of DLOs concentrate on single tasks e. g. [11], while in the domain of rigid objects systematic approaches exist e.g. [10]. Such a systematic approach must address two important problems namely the task description and the sensor-guided execution of such a task description. Here, different types of sensors such as force-torque, acoustic or vision sensors like colour cameras can be used. Often, a combination of different types is useful since each type of sensor has its own strengths and weaknesses [8] or [2]. But the programming of the sensor data processing is often tedious and difficult. The encapsulation of the sensor data processing in task specific routines like skills offers a conceptual solution [4] for the problem of task execution.

The task description problem has two main aspects. The first aspect is the need for a formalism able to describe the task in a manner that provides the information necessary to parametrise respectively configure the sensor-driven skills and the sensors as well. In the field of rigid object manipulation contact based task descriptions are well known [14]. The constrains imposed by the contact situations and the transitions between different contact situations provide the base to link the task description to execution skills [9].

The second aspect of the task description problem is the way how to derive a task description for a specific manipulation task. This can be solved manually by writing down the sequence of the skills which will solve the specific problem. In that case, the description formalism is used as a kind of programming language. But for the execution by a real robot the programmer must also supply detailed geometric informations like positions or distances. Therefore, it is much more comfortable if the assembly task can be automatically planned based on a CAD description of the environment e.g. [10]. But since planning is known to be NP-hard, [12] proposes an approach where the task description is automatically derived from a demonstration of the assembly task in a virtual environment.

In the field of non-rigid objects the planning becomes even more difficult because of the almost arbitrary degrees of freedom of the work piece. Here, [5] used the programming by demonstration to solve a hose insertion task. However, the task is demonstrated using the same real sensors as used later in the task execution. The recorded sensor data profile is further processed to identify the control parameters for the execution. The mapping of the continuous data profile to contact states and discrete transitions respectively still has to be done manually, according to the authors. This reveals two disadvantages of this approach. First, such a demonstration has to be carried out with the same sensors that are later used for the manipulation task. Second, human work is not only necessary for the demonstration but also for the identification of control parameters.

Therefore, in [7] the programming by demonstration paradigm is performed in virtual reality and the human worker uses a haptic input device to demonstrate the task. Fig. 1 shows the complete system concept, from the demonstration of the task to the sensor based detection of contact state transitions.

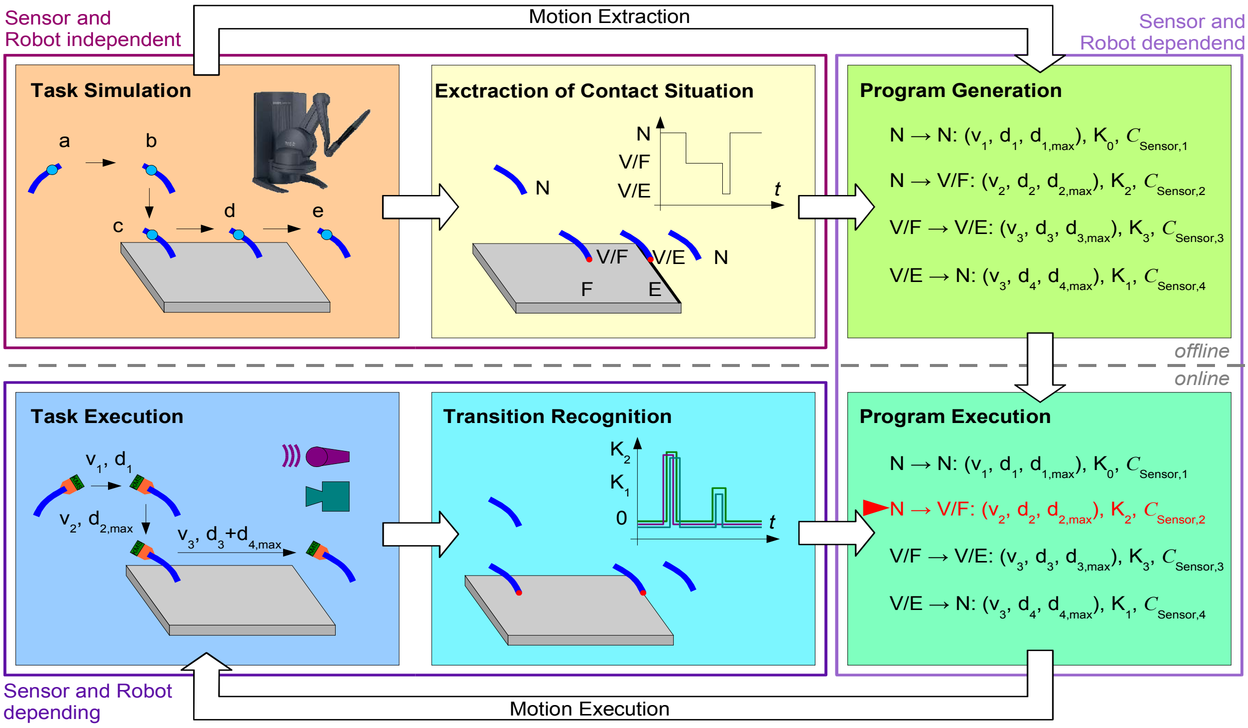

Figure 1: Depiction of the overall process of programming and executing assembly tasks with DLOs. In the off-line or programming phase (upper half). a human operator demonstrates the assembly task to a Programming by Demonstration (PbD) system. The PbD system discretise the task over space and time. The result is a abstract task description, which encodes the motions as well as the contact situations between work piece and environment, which are necessary the complete the task. In the on-line or execution phase (lower half) the robot controller moves the robot according to the motions in the task description and monitors the current contact situation to adapt the motions to uncertainties in work piece and environment geometry.

The complete system consists of an off-line phase where the task is demonstrated by a human worker and an on-line phase where the robot executes the given task description. The demonstration takes place in a virtual environment based on a CAD model of the environment and a real time simulation of the DLO [6]. The demonstrated trajectory is then segmented based on contact states [1] and the complete task description, i.e. program, is derived. The program execution module executes the given task stepwise according to this task description. For that, the motion parameters are send to the robot controller and the sensors are selected and parametrised according to the task description. After the transition is recognised the robot is stopped and the program execution proceeds with the next program step. Thus, the program execution employs an open loop control system. Further, the contact formalism provides an abstraction level for the task description which allows a complete automatic process during the on-line phase. Also, the contact states respectively the transitions between them provide a common base for all different types of sensors used for the execution like force-torque, colour cameras or acoustic sensors [3] and [13]. Indeed, the contact formalism allows to avoid task specific sensor data processing algorithms.

In this paper, we concentrate on the system's off-line phase. In particular, the task demonstration itself is discussed i.e. good practices that lead to a robust task description based on the contact state formalism. Here, a *robust* task description means a task description that achieves the assembly's goal after its execution even in the presence of uncertainties. In this regard, also the restrictions of the concept are discussed. Further, the automatic segmentation of the trajectory derived from the haptic input device and the extraction of the motion parameters are discussed. Thus, this paper covers the extraction of the contact situation and the program generation step from Fig. 1.

The paper is organised as follows: In Section 2 a brief presentation of the previous work concerning the contact based formalism for the description of assembly tasks for DLOs is given. The basic formalism is applied to an example of use in Section 3. Section 4 covers the task demonstration in the virtual environment and the extraction of the program i.e. the segmentation of the demonstrated trajectory and the associated contact situations. Additional fine-tuning aspects are discussed in Section 5. Then, Section 6 presents the resulting program. Finally, the restrictions of the presented approach are discussed in Section 7 and we give an outline of the further work which seems to be able to cope with these restrictions.

Previous Work

Before we can go into further detail we will present the underlying contact based formalism for the topological description of contact situations between DLOs and a polyhedral environment. The concept [1] includes several abstraction levels each to stepwise extract the most important information. The most concrete level is geometric level. Here, all environment objects, positions, angles and the exact shape of the DLO are known. This model corresponds to the CAD-model of the environment with the DLO-simulation from the Task simulation module. For this paper, the discrete contact states (which represent the most abstract level, Fig. 2) are the most interesting. The transitions between such discrete contact states are closely related to common skills like "Move-To-Touch" as described in [4].

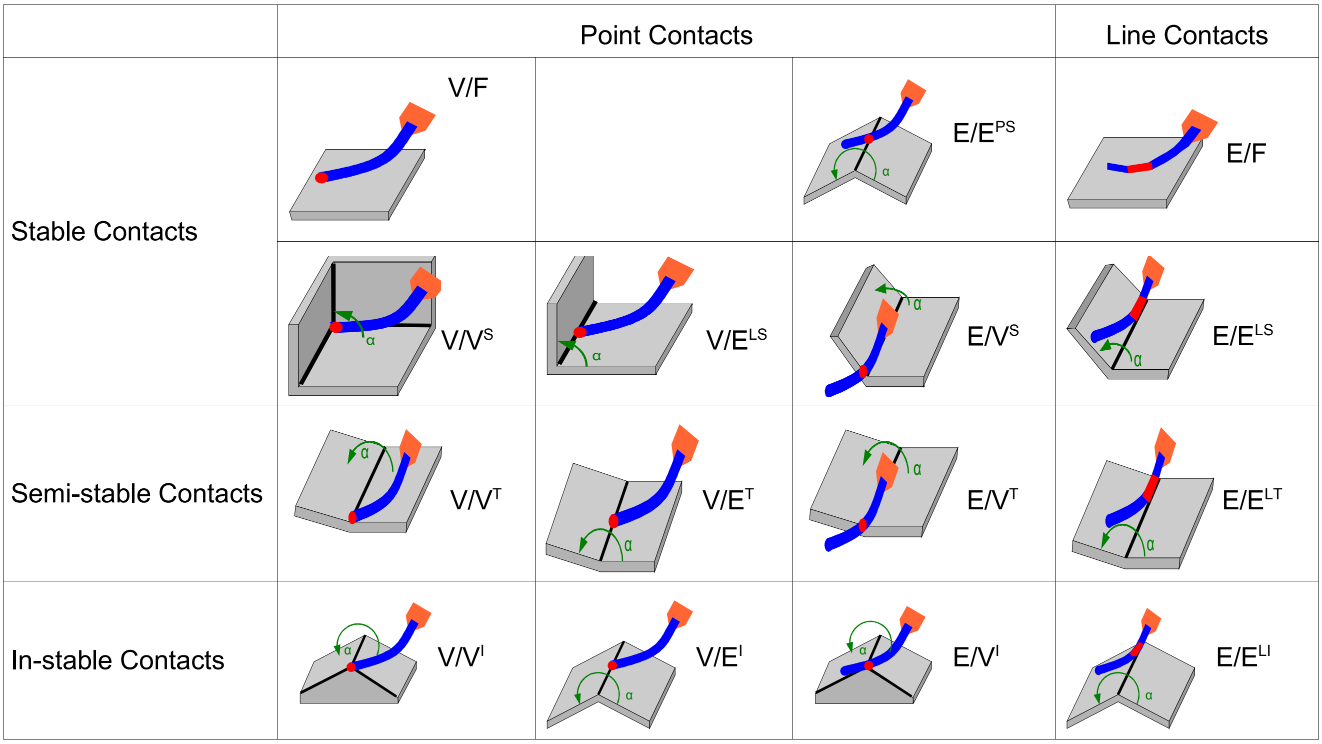

Figure 2:The discrete contact states [1] represent all meaningful combinations of all DLO- respectively environment primitives {V,E,F}, the stability class {stable (S), transient(T), in-stable(I)} and the dimension of the contact {line(L), point(P)}. Here, the enumeration of all possible discrete contact states in polyhedral environments is shown. Since any vertex contact must be a point contact E/V^{LS} is not regarded as meaningful. The amount of the angle α mainly decides the stability class. Further, the drawn sketches of topological situations are for many discrete contact states only one example out of several possible.

Te discrete contact states encode explicitly the stability class and the contact dimension of any contact state. Especially, the stability class is very important for the handling of DLOs since it reveals much information about the behaviour of the DLO. A contact state is considered to be *stable* if and only if any small motion in any direction does not result in a change of the contact state. This definition depends directly on the given flexibility of the DLO and therefore for a rigid object no such stability is possible. If such a small motion results in a contact change, the contact state is considered to be *in-stable* if and only if the resulting contact is not a priori known. Let us consider e.g. V/V^{I} here a small motion could result either in a *V/F* with any of the three surrounding faces or in an E/E^{P} with any of the three surrounding edges or even to *V/N*. In-stable contact states occur usually at convex points or edges. But if the resulting contact state is a priori known then the contact state is considered to be *semi-stable* or *transient*. Such semi-stable contact states appear due to low, non-convex angles between faces or edges. Here, a DLO is able to change the shape and to slide over it. As example, we consider V/E^T where a small motion in any direction must result in a V/F contact state. Please note that for each geometric situation resulting in a transient contact state, there is a corresponding situation with a smaller critical angle resulting in a stable contact state. Additionally, the exact amount of the critical angle from where such non-convex edge or vertex becomes stable depends also on stiffness and/or friction. If the task is demonstrated in the virtual environment with sufficient precision then the question of which type of contact state is directly answered by the simulation.

The transitions between such discrete contact states can be grouped into four important classes. Each class represents a special behaviour of the DLO. Therefore, the classes provide a link from the abstract formalism directly to the recognition algorithms. Further, the transition class can directly be derived from the differences between two consecutive discrete contact situations. These transition classes are the base for the automatic selection and parametrisation of the sensors within the execution phase. Semi-stable and in-stable contact states appear usually only for a very short period of time. Therefore, the corresponding transition classes consists typically of sequences of two contact transitions. The first is an initiated transition towards a semi-stable or in-stable contact state followed by a transient or spontaneous transition, respectively. The class of point/line transitions consists of all transitions between stable point to stable line contacts and vice versa. The class of establishing/releasing contacts consists of all contacts where translatory degrees of freedom change. To this class belong especially all transitions from and to N.

Environment guided handling for DLOs

Before we apply the concept of environment guided handling to an application example, the basic concept is introduced. The contact based description formalism offers the possibility to describe the task in a way that stepwise reduces the uncertainties. Since each contact imposes constraints on the DLO, it removes some degrees of freedom of the work piece. Thus, the uncertainties can be reduced stepwise by each new contact or change from one contact to another contact of higher degree. This concept is also used for the assembly of rigid objects [4] or [9]. There, only six degrees of freedom exist but after a contact has been established any further motion must keep the constraints imposed by the contact. Usually, such motions are controlled by force torque sensors and the control laws of the work piece must be known. Here, in the case of deformable objects with their almost arbitrary degrees of freedom, such constraint motions are rather difficult, due to the individual behaviour of each DLO. On the other hand, the definition of a stable contact shows that exact constraint motions are not necessary for DLOs, because small variations from the ideal motion direction can be compensated by an increased or decreased bending without changing the contact situation. Further details can now be discussed based on an application example for assembly tasks.

The assembly task consists of two peg-in-hole type sub-tasks. Since most research in the field of DLO handling investigates peg-in-hole variants, e.g. [11], such tasks can act as a kind of benchmark. Both sub-tasks are subsequently demonstrated. Each time the cable (made of acetylene) is gripped near one of its vertices. In the beginning the cable is assumed to be in a clamp at a fixed known location. This is necessary since the empty contact situation does not change until the robot grips the cable. Therefore, the motions necessary to describe the gripping tasks cannot be described with our basic formalism. After one of the vertices are gripped, the robot approaches to a starting position. Due to the fixed clamp the shape of the DLO including the bending, the orientation and the length is roughly known at the start position. However uncertainties still remain due to the tolerance of the clamps and some plastic deformation of the DLO. Therefore, the DLO's exact shape at the start position after the execution of the gripping sequence is always somewhat different.

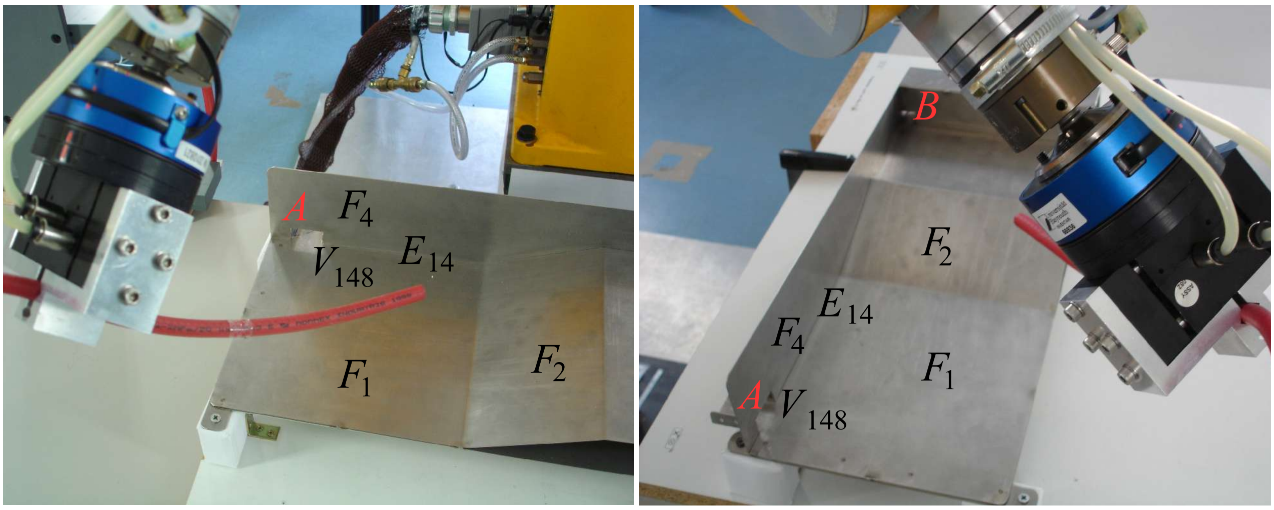

Figure 3: Start situation before the execution of the first sub-task from two different views.

The Fig. 3 shows the situation just before the execution of the first sub-task from to different views. The faces and edges and some relevant points of the environment are named. For the sake of clarity the indices of the edges and vertices are constructed as follows: E_{ij} means edge between F_i and F_j , and V_{ijk} means vertex between F_i , F_j and F_k. The first sub-task describes the peg-in-hole towards hole A and the second sub-tasks goal is hole B.

We start with the sub-task regarding hole A. Since the robot placed the first vertex roughly above of F_1, the contact V_S/F_1 is established first by a motion towards the surface of F_1. Please note that an exact goal position or length of the motion is not necessary because the motion will be stopped during the execution phase when the sensors recognise the change of the contact situation. Further, the orientation of the DLO at the start position grants that the DLO hits F_1 first with its vertex instead of any part of its edge. The motion is stopped due to the change of the contact situation even if another transition like E/N to E/F_1 occurs. The vertex contact ensures that the robot's gripper does not collide with the environment during the assembly sequence. Please note that the change of the contact situation still occurs even if the environment is moved a small distance, rotated by a small angle around an arbitrary axis or if the bending, orientation or position of the DLO has changed by a small amount. Therefore, the description is robust against any small variations.

The next contact transition is V_S/F_1 to V_S/E^S_{14} . Since this edge can guide the DLO directly towards hole A,we can drag the vertex along E_{14} towards hole A. For the success of this motion, the angle between the tangent along the DLO at V_S and the direction vector of the edge E_{14} is important. Advantageous, is an angle of 45° which is roughly the case due to the orientation of the DLO in our example. But even an angle of 0° would work as long as the robot does not collide. An angle of 90° is the upper bound since the vertex must not point to hole A. If this is the case, dragging the vertex along E_{14} would still lead to an in-stable contact V_S/V^I_{148} but the result of the following spontaneous transition is an E/V^S_{148} contact and the V_S could be inserted into hole A but depending on the exact bending V_S can also miss hole A. In this case, the goal of the assembly task can not be achieved in a robust manner. Therefore, the vertex must point towards hole B. In this case, the spontaneous transition following V_S/V^I_{148} ensures that the vertex directly snaps in hole A. Thus, the goal of the first sub-task is reached.

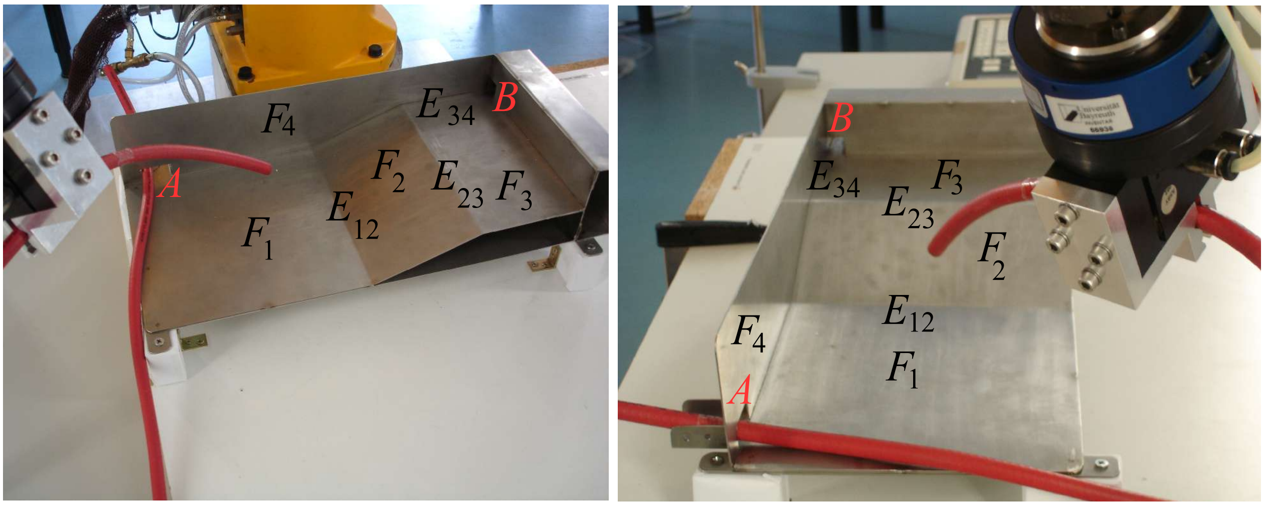

Figure 4: Start situation before the execution of the second sub-task from two different views.

The second sub-task is about inserting V_E into hole B. Here, we can use E_{34} to guide us directly to the goal. The insertion will succeed if the tangent along the DLO at the free vertex is close enough to the direction vector of E_{34}. Due to the basic compliance of the DLO a rough congruence is sufficient. Since F_1 is bigger than F_2 or F_3 we place the starting location again above F_1 and create again V_E/F_1 as the first contact. But we do not know the exact direction of the edges E_{14}, E_{24} and E_{34} since the orientation of the real environment could differ slightly from the virtual environment. Thus, first we try to reach F_3 before we start to align the DLO with E_{34} . Therefore, we must cross E_{34} first. Since the angle between F_1 and F_2 is 165° we have to expect the transient transition sequence $V_E/F_1 \rightarrow V_E/E^T_{12} \rightarrow V_E/F_2$. In order to stress the compliance of the DLO not too much and to keep the current contact the motion direction can be adapted to the current slope. After a spontaneous $V_E/F_2 \rightarrow V_E/E^I_{23} \rightarrow V_E/F_3$ the face F_3 is reached. We can now reach our sub-goal V_E/E^S_{34} by a motion towards F_4.

The remaining work is to align the DLO at the free vertex to the direction vector of E_{34}. Since the starting orientation of the DLO and the direction vector of E_{34} point roughly into the same direction a change from point to line contact is sufficient e.g. V_E/E^S_{34} to E/E^S_{34}. Now, a small transfer motion guided by E_{34} is enough to complete the second sub task. Please note, that the bending of the DLO is here again advantageous. Otherwise the change from point to line contact could result in a loss of V_E/E^S_{34} to V_E/N. Such a contact situation would prevent the insertion into hole B. With such less advantageous bending, a more complex task description is necessary. After the V_E/N^E/E^S_{34} is reached another transition to change the contact dimension from the line contact E/E^{S}_{34} back to the point contact V_E/E^{S}_{34} is needed. Due to the intrinsic compliance of the DLO all stable contact states have a rather wide range but the execution stops usually shortly after the expected transition is reached. Especially in the case of a change of the contact dimension the state of the DLO is still close to the start situation. Thus, the tangent along the DLO in the newly reached V_E/E^{S}_{34} is still close enough to direction vector of the E_{34} to allow a robust insertion of the DLO into hole B.

Generating a task-description

As pointed out earlier, it is tedious and error-prone to manually create such a task description as described in the previous section. To overcome these difficulties we propose an automatic extraction of such a task description by means of an "Programming By Demonstration" (PbD) system. The primary task (of the PbD) is to compute a sequence of contact situation and transition as well as the necessary robot motions between theses contact situations, based on the demonstration observed. This primary task can be subdivided in three steps. First, monitor a user demonstration of the manipulation task, second, find the relevant contact situations and third, compute motion instructions for a robot.

Our system consists of a "Phantom" 6-DOF haptic input device connected to a CAD system (Fig. 5). The CAD system is capable of computing the forces and contact between the work piece (including some aspects of its dynamic behaviour) and the modelled environment [6]. To generate a task description, the user performs the desired manipulation task in the

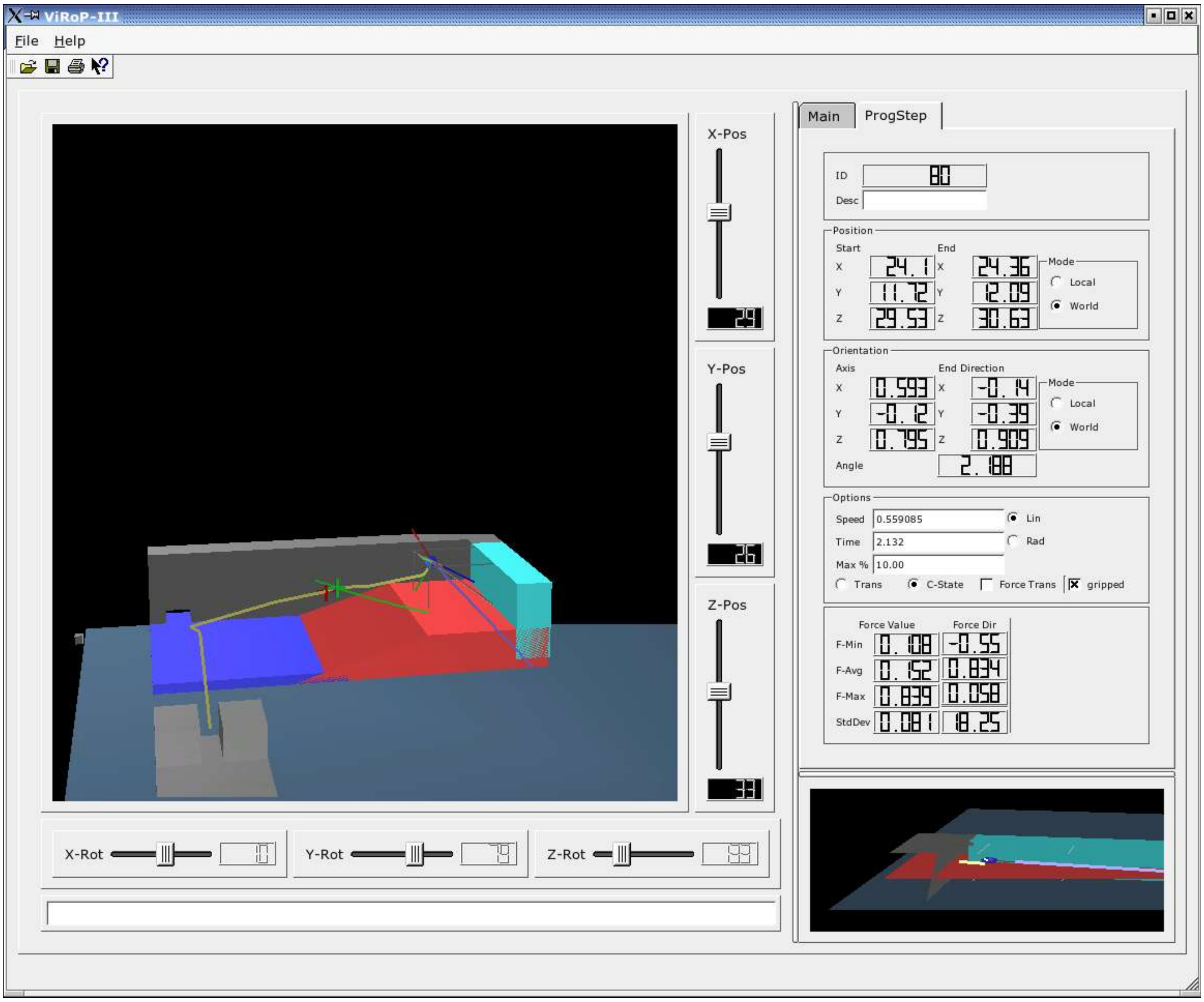

Figure 5: The Programming by Demonstration system in its trajectory editing mode. The left area shows the current manipulation scene with the previously demonstrated trajectory (yellow) and the work piece shape and position at the end of the current program step. The right area shows some characteristics of the current program step, like start and end position, expected force range and other. The area at the lower right corner shows the scene as viewed from a camera mounted at the robots gripper. The gripper itself is modeled as the blue sphere in the left scene.

virtual environment. The system assists the user by means of haptic feedback ("sense of touch") and immediate display of contact areas along the work piece. After the manipulation task has been successfully performed in the virtual reality, the system splits the demonstrated motion trajectory of the work piece into mostly linear segments. These segments realise the contact transition previously discussed. The generated trajectory is presented to the user for acceptance or further tweaking. Once accepted by the user it is translated into a robot language and written out.

While our PbD system monitors the demonstration of a manipulation task, it continuously computes the distance or intersection of the work piece (modelled as a sequence of cylindrical segments) with the CAD model of the environment object. For each discrete time step, the system maintains a separate list of all contacts between a given part of the work piece and all environment objects. Each list contains information about the type of contact as well as additional information like the identity of the object and surface involved.

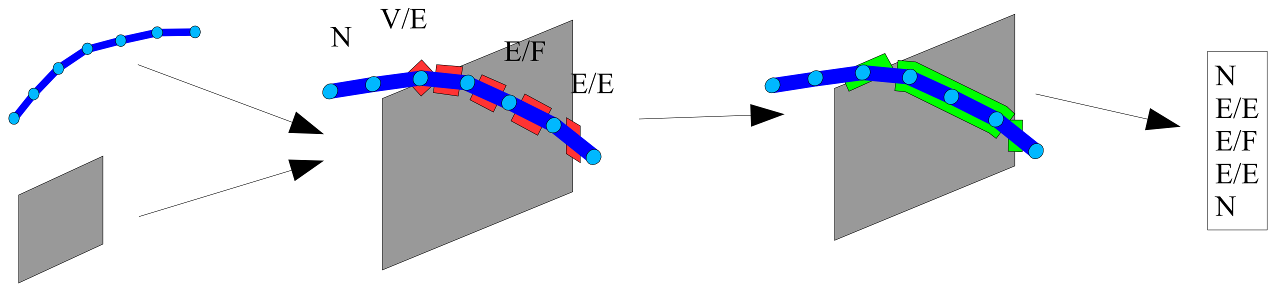

Figure 6: Computing a contact state from work piece and environment geometrics. The distance to, or intersection with, all relevant surfaces of the environment is computed for each segment of the work piece. The resulting preliminary contact states are combined along the work piece to form larger areas of similar contact, resulting in a topological description of the contact situation between a work piece and one environmental surface.

These lists of work piece areas which are in contact with one or more other objects are used to split the trajectory (originally demonstrated by the human operator) into segments. Each segment represents one distinguish change in the overall contact situation. A new segment start whenever a contact area is created, deleted or substantially changed. A substantial change is: (a) the addition of a newly contacted surface to an existing contact area, (b) the removal of a no-longer contacted surface from such an contact area, (c) a change in the type of contact. Such a change in type of contact can be a transition from e.g. E/F to E/E among others [1]. As an extension to the principle of describing a manipulation task as a sequence of contact changes, additional segments can be introduced whenever a significant change in the motion direction or work pieces orientation occurs. This extension is necessary to cover contact state preserving transfer motions as well as picking up the work piece. Each contact area has exactly one well-defined contact to each contacted surface, edge or vertex of each environmental object. All contacts between one contact area and its contacting objects are then combined into one contact state as defined in Fig. 2. The result is a two-dimensional sequence of such discrete contact states. The first dimension is along the length of the work piece and enumerates all discrete contact state from one end of the work piece to the other end at a specific time step. Such a sequence along the work piece is called "contact situation" The other dimension is time and orders the contact situations from the start situation of a manipulation task until its end situation. Each contact situation is annotated with extra information like the rough position at which the contact situation has emerged. This sequence of contact situations represents the manipulation task in a robot independent way and can be written out as a task-description, ready for execution.

Verification and fine-tuning of a task-description

The PbD system can be used to further examine and fine-tune such a task-description in preparation to execute it on a specific robot. Fine-tuning involves several optional steps like correcting positions at which a particular state transition is expected to occur, or removing unneeded motions which may result from mistakes (like missing an edge) while demonstrating the task in the PbD system in the first step, as well as adoption to a specific robot. This includes the definition of linear and angular velocity for different parts of the trajectory as well as establishing maximal movement distances for every state transition. In case of force/torque sensors as detection system in the automatic task execution, the system can compute the expected force or torque signals before and after a state transition. The system can also compute parameters used to configure the detection system for recognising the state transition in the execution phase.

For camera based approaches, the minimal number of cameras necessary for the task have to be computed based on the CAD-model. Since a good view depends not only on geometric constraints but also on the different image features. The image processing algorithms determine the quality of any observation vector. However, a detailed analysis of optimal observation vectors would be out of the scope of this paper.

Resulting program

Fig 7 shows the resulting task descriptions i.e. programs for both sub-tasks. The gripping process is not listed. Each program step consists of a robot command, the motion parameters, the corresponding contact situation and the resulting transition. Three basic commands are used for the robot. The *open* command is used to open the gripper after the execution of each sub-task. The *move* command is used to initiate a state transition from the current contact situation to a target situation. Since transition in the class of spontaneous or transient transitions involve an intermediate in-stable (or semi-stable) contact state, a *nop* command is introduce to maintain a 1:1 relation between robot commands and contact state changes. The robot motion that initiates the complete transient transitions has the intermediate in-stable contact situation as its target, while the "nop" command has the final stable contact situation as its target. All the "move" commands are special motions where the robot follows the trajectory derived from the demonstration. The motion parameters specify the direction toward the desired contact state transition, the orientation of the gripper as well as a maximal motion distance that can be travelled without setting the work piece or the robot at danger. These parameter further specify the expected minimal motion distance form the current position up to the earliest point where the transition can occur. In addition to these strictly kinematic parameters the motion parameter include information about sensor configuration and sensor selection. These latter parameters obviously depend on the available sensor equipment of the robot. The contact situations are not completely written down since for both sub-tasks only the part of the DLO between the gripper (*G*) and the manipulated vertex matters.

Robot command	Motion parameter	Contact situation	Transition/Skill
		First sub-task (hole A)	
nop		$V_S/N \wedge E/N \wedge E/G \ldots$	
move	Direction is negative Z Orientation is 45° to F_1 and F_4	$V_S/F_1 \wedge E/N \wedge E/G \ldots$	establishing contact
move	Direction is negative Z fixed-distance 1 cm[1] Orientation unchanged	$V_S/F_1 \wedge E/N \wedge E/G \ldots$	transfer
move	Direction is positive Y Orientation unchanged	$V_S/E^S_{14} \wedge E/N \wedge E/G \ldots$	establishing contact
move	Direction is positive Y fixed-distance 1 cm	$V_S/E^S_{14} \wedge E/N \wedge E/G \ldots$	transfer
move	Direction is negative X Orientation unchanged	$V_S/V^I_{148} \wedge E/N \wedge E/G \ldots$	spontaneous transition
nop	-	$V_S/F_1 \wedge E/F_1 \wedge E/N \wedge E/G \ldots$	
move	Direction is positive Y fixed-distance 2 cm	$V_S/F_1 \wedge E/F_1 \wedge E/N \wedge E/G \ldots$	transfer
open	-		
		Second sub-task (hole B)	
nop		$\ldots \wedge E/G \wedge E/N \wedge V_E/N$	
move	Direction is negative Z Orientation is 45° to F_1 and F_4	$.. \wedge E/G \wedge E/N \wedge V_E/F_1$	establishing contact
move	Direction is negative Z fixed-distance 1 cm Orientation unchanged	$.. \wedge E/G \wedge E/N \wedge V_E/F_1$	transfer
move	Direction is positive X fixed distance 2 cm	$.. \wedge E/G \wedge E/N \wedge V_E/F_1$	transfer

1 The exact amount is computed from simulated force values and depends on the exact orientation of the gripper in the task demonstration as well as and on the physical bending coefficients used in the simulation.

Robot command	Motion parameter	Contact situation	Transition/Skill
move	Direction is positive X Orientation unchanged	$_{..}\wedge E/G \wedge E/N \wedge V_E/E^T{}_{12}$	transient transition
nop	-	$_{..}\wedge E/G \wedge E/N \wedge V_E/F_2$	
move	Direction is positive X / positive Z,with 15° slope Orientation unchanged	$_{..}\wedge E/G \wedge E/N \wedge V_E/E^I{}_{12}$	spontaneous transition
nop	-	$_{..}\wedge E/G \wedge E/N \wedge V_E/F_3$	
move	Direction is positive Y Orientation unchanged	$_{..}\wedge E/G \wedge E/N \wedge E/F_4 \wedge V_E/E^S{}_{34}$	establishing contact
move	Direction is positive Y fixed-distance 1 cm	$_{..}\wedge E/G \wedge E/N \wedge E/F_4 \wedge V_E/E^S{}_{34}$	transfer
move	Direction is positive Y, negative Z with 45° slope	$_{..}\wedge E/G \wedge E/N \wedge E/E^{LS}{}_{34} \wedge V_E/E^S{}_{34}$	point to line contact
move	Direction is positive X fixed-distance 5 cm	$_{..}\wedge E/G \wedge E/N \wedge E/E^{LS}{}_{34} \wedge V_E/E^S{}_{34}$	transfer
open	-		

Figure 7: Task description / robot program for both sub tasks.

Further, the additional motion after each “establishing contact” needs to be discussed. These transfer motions are added by the program generator in order to establish a contact force that makes/keeps the corresponding contact stable. The amount of the motion depends on the stiffness of the work piece as well as the overall geometric situation. It has two complementary parameters. First it should be large enough to maintain the newly established contact even if a following motion is carried out in a slightly wrong direction. For example moving horizontally across a table which is supposed to have a horizontal surface, but in fact has a small slope in motion direction. Second, it must be small enough not to damage the work piece or to destroy the contact situation by pushing the vertex against another part of the environment. The program could now be executed, as the transition classes between the contact states can be used for the selection of the sensors. For each step, the corresponding contact situations of the start and goal situation and the derived transition is given to each sensor data processing unit. Based on this data and additional sensor-specific parameter, e.g. for cameras parameters like the camera's view, each individual sensor starts the observation or reaches a sleeping state if it cannot recognise the expected transition[1].

Restrictions and Further Research

In this section the capabilities and the restrictions of the concept are discussed and the outlook on further work is given which should allow to overcome most of the current restrictions. The presented example of use showed some of these restrictions. First, the orientation and natural bending of the DLO before the task execution must largely correspond to the situation in the virtual environment. Fig 8 shows another variant of peg-in-hole tasks. No task description exists for this variant which can be executed in a robust way by our control system.

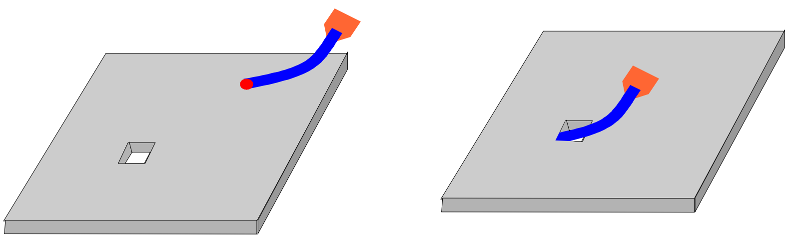

Figure 8: Start situation (left) and goal situation (right) for another variant of peg-in-hole tasks.

1 A video for the camera-based execution of both subtask is available under the project's web site http://ai3.inf.uni-bayreuth.de/projects/virop/index.php

In this example, the robot can only insert the DLO successfully into the hole if the motion is rather precise since no edge or other geometric primitive can be used to guide the DLO to the hole. It is rather likely that the hole is missed if we assume uncertainties in position or orientation of the environment. Although, the borders of the upper surface could be detected based on changes of the contact situation respectively contact state transitions. Therefore, exploration motions could be used to probe the real orientation, sizes and position of the environment object. Based on the collected information, the position of the hole could be approximated or at least a search strategy employed. However, the current open loop control system is unable to do that. Thus, we propose a control system not based on the single sequence of contact situations but based on net of contact situations. The motion parameters for each possible link between to points in the net can be computed based on the CAD model in the virtual environment. In the on-line phase, the control system is able to navigate within this net of contact situations or to localise the current state of the DLO within such a net. This also allows the implementation of a cyclic control i.e. to obtain a closed loop control.

This approach also allows an almost arbitrary start position, bending or orientation of the DLO. Thus, most of the restrictions of the current simple control mechanism can overcome. The only remaining restriction is that a robust task description for gripping a DLO from any location still does not exist. But this problem cannot be solved by any contact based approach.

References

[1] Acker J. Henrich D.: „Manipulation of Deformable Linear Objects: From Geometric Model Towards Program Generation“. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, (2005-4), pp.1553-1559.
[2] Acker J, Schlechter A., Henrich D.: „Handhabung deformierbarer linearer Objekte: Vergleich zwischen kraft- und bildbasierter Erkennung von Kontaktzustandsübergängen“. In VDI-Berichte 1841, Robotik-2004, München, 2004.
[3] Acker J. and Henrich D.: "Manipulating deformable linear objects: Characteristic Features for Vision-Based Detection of Contact State Transitions". In: 2003 IEEE Int. Symp. on Assembly and Task Planning (ISATP'03), Besancon, France, 2003
[4] Hasegawa T., Suehiro T., and Takase K.: "A model-based manipulation system with skill-based execution". In: IEEE Transactions on Robotics and Automation, vol. 8, No. 5, pp. 535-544, October 1992
[5] K. Hirana, T Suzuki, S. Okuma, K. Itabashi, F. Fujiwara: "Realization of Skill Controllers for Manipulation of Deformable Objects Based on Hybrid Automata". In: Proc. 2001 Int. Conf. on Robotics and Automation, vol. 3, pp. 2674-2679, Seoul, Korea, May 2001
[6] Kahl B; Henrich D. "Manipulation of Deformable Linear Objects: Force-based simulation approach for haptic feedback" In: Proc. of the12th Int. Conf. on Advanced Robotics pp. 115-122, Seattle USA, July 18th-20th, 2005
[7] Kahl B., D. Henrich: "Virtuelle Roboter Programmierung: Konzept und Prototypische Implementierung". In: VDI-Berichte 1841, Robotik-2004, München, 2004.
[8] Miura J. And Ikeuchi K.: "Task-Oriented Generation of Visual Sensing Strategies in Assembly Tasks". In: IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 20, No. 2, pp. 126–137, 1998.
[9] Morris G. H., Haynes L. S.: "Robotic assembly by constraints". In: Proc. 1987 IEEE Int. Conf. on Robotics and Automation (ICRA’87), pp. 1507-1515
[10] Mosemann, H. "Beiträge zur Planung, Dekomposition und Ausführung von automatisch generierten Roboteraufgaben" Fortschritte in der Robotik, Band 6, ISBN 3-8265-7710-8, Shaker-Verlag, 2000.
[11] Nakagaki H., Kitagaki K., Ogasawara T., Tsukune H.: "Study of Insertion Task of a Flexible Wire into a Hole by Using Visual Tracking Observed by Stereo Vision". In: Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 126-132, Minneapolis, USA, April 1996
[12] Onda H., Hirukawa H., Takase K.: "Assembly motion teaching system using position/force simulator – Extracting a sequence of contact state transition". In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'95), vol. 1, pp. 9-16, 1995.
[13] Schlechter A, Henrich D.: „Discontinuity Detection for Force-based Manipulation“. To be appear in: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida, USA, May 15th-19th 2006
[14] Xiao J., Liu L.: "Contact States: Representation and Reconizability in the Presence of Uncertainties". In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,Vol.2 pp.1151-1156, Victoria, B.C.,Canada 1998

Curriculum Vitae:

Prof. Dr. Dominik HENRICH obtained his Diploma (Master's Thesis) in Computer Science in 1991 from the University of Karlsruhe, Germany. From 1992 to 1994, he was supported by a stipend granted by the German Research Foundation (DFG) and finished his Doctorate (Ph.D. Thesis). From 1996 to 1999, Prof. Henrich built up the research group for "Parallel Processing and Robotics" at the Insitute for Process Control (IPR) at the University of Karlsruhe. From 1999 to

2003, he headed as Professor the research group "Embedded Systems and Robotics (RESY)" at the Informatics Faculty of the University of Kaiserslautern. Since August 2003 he has held the chair for Applied Computer Science III (Robotics and Embedded Systems) at the University of Bayreuth.

J. Acker received his Diploma (Master's Thesis) in Computer Science in 2001 from the University of Kaiserslautern, and joined the "Embedded Systems and Robotics (RESY)" research group in 2001. He worked mainly on the contact based handling of deformable linear objects using vision sensors. From 2003 on he is a member of the chair for Applied Computer Science III.

B. Kahl received his Diploma (Master's Thesis) in Physics in 2000 from the University of Kiel, Germany. He joined the "Embedded Systems and Robotics (RESY)" research group at the University of Kaiserslautern in 2000 and worked haptic simulation for robot programming. From 2003 on he is a member of the chair for Applied Computer Science III.