Building blocks for SLAM in Autonomous Compliant Motion

Herman Bruyninckx, Joris De Schutter, Tine Lefebvre, Klaas Gadeyne
Peter Soetens, Johan Rutgeerts, Peter Slaets, Wim Meeussen*
Dept. of Mechanical Engineering, K.U.Leuven, Belgium

Abstract

This paper presents our research group's latest results in fully autonomous force-controlled manipulation tasks: (i) advanced non-linear estimators for simultaneous parameter estimation and contact formation "map building" for 6D contact tasks (with active sensing integrated into the task planner), and (ii) the application of these results to programming by human demonstration, for tasks involving contacts. The paper also describes our efforts to integrate the estimators in the Free Software robot control software project Orocos.

1 Introduction

One of the major skills required in an autonomous robot system is the ability to use its sensors to make a "map" of its environment. This skill is often called *SLAM* (Simultaneous Localization And Map building) in the context of mobile robotics. The last years have seen very promising results in this area, for indoor as well as outdoor navigation.

In comparison to mobile robotics, the research community of force-controlled *Autonomous Compliant Motion* (ACM) is rather small, so much less SLAM results have been achieved. The major reasons for this lack of SLAM results are: (i) force-controlled robots are still difficult to get at, because they can not be bought off the shelf; (ii) safely "navigating" in contact with an object is a much more difficult control task than safely navigating a mobile robot; and (iii) ACM is inherently a six-dimensional problem, which makes the SLAM estimation in ACM significantly more difficult than the basically three-dimensional problem in mobile robots.

This paper presents the authors' latest experimental results in ACM. The focus is on the "SLAM"-like aspect of the problem: estimation of the current contact *formation* (i.e., which parts are in contact), and the precise contact *configuration* (i.e., what are the geometrical parameters of the contact); the generation of *active sensing* actions to get more information about nominally poorly observable variables; and the autonomous construction of contact models in tasks demonstrated by humans. We also report on the software engineering of making all the abovementioned sensing skills work together in a complex robot control system.

Of course, the experiences and algorithmic developments from the mobile robotics research have been a very important source of inspiration for our research. But our ACM research has by itself also produced some interesting new estimation approaches that can be useful for mobile robotics SLAM too.

*HB and TL are, respectively, Postdoctoral and Doctoral Fellow of the F.W.O–Vlaanderen. All authors gratefully acknowledge the financial support by K.U.Leuven's Concerted Research Action GOA/99/04, the Belgian State—Prime Minister's Office—Science Policy Programme (IUAP), and the EU projects *Orocos* and *Ocean*.

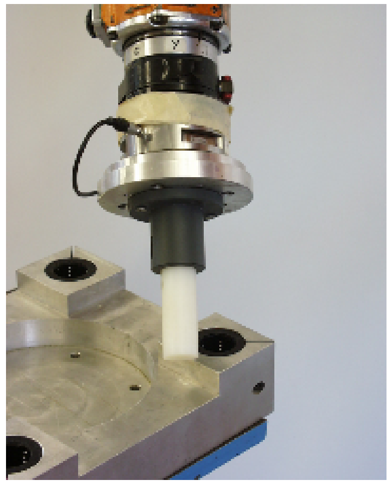


Figure 1: Typical force-controlled compliant motion task. During the task, several different contact formations occur, and the relative pose of the tool and the workpiece changes continuously. The localization and identification of these contact formations are necessary for autonomous execution.

Problem description. Figure 1 shows a typical example of a task where Autonomous Compliant Motion is needed: a robot holds a tool, and its wrist force sensor measures the contact *wrench* (= 6D force) when the tool touches the workpiece. The robot needs a force controller that can provide a stable contact, and can move compliantly and robustly within the current contact constraints. The robot also measures the *pose* of its tool, as well as its *twist* (= instantaneous 6D velocity). From this pose/twist/wrench data, the controller must deduce information about the current contact formation: Which part of the tool is in contact with which part of the workpiece? What is the relative pose of tool and workpiece? When does a change in contact formation occur? What is the type of the contact, and how does the environment look like?

Compare this problem description to a blind human being, who holds the tool in his hands and can only extract information from the contact forces and compliant motions he feels. If you try this out for yourself, you quickly notice that the general problem is certainly solvable, and a human doesn't need much time to recognize the contact formations. Approaching this "intelligence" is the main aim of our research efforts.

Overview of the paper. The paper describes our recent results in the following sub-problems of the general ACM problem. Section 2 explains how we have reduced the contact modelling complexity by symbolically representing and automatically combining contact formations, while at the same time improving the information gathering process. Section 3 gives the new powerful non-linear estimators that we have developed for the on-line estimation of contact geometry, even under very large initial uncertainties. Section 4 explains how our estimators determine which of a set of possible modelled contact formations best represents the wrench and twist measurements of the current task. Section 5 presents results on active sensing, i.e., our approaches to add extra motions to the nominally specified trajectory, for the sole purpose of gaining more information about the uncertain parameters in the contact models. (All our estimation algorithms are firmly rooted in Bayesian theory.)

Section 6 applies the results from the previous Sections to the problem of building a model of the contact situations during a task executed by a human demonstrator. Finally, Section 7 outlines our software design and implementation efforts for real-time support for ACM.

2 Modelling

Ideal rigid contacts are the simplest forms of robot-environment interactions: modelling them can be done with a rather limited number of contact features: vertex, edge and face. Any polyhedral contact formation can be modelled by a combination of these primitives, so with them, we form an "elementary contact library" of polyhedral contacts: vertex–face, edge–edge, etc. (It's easy to incorporate also non-polyhedral contact primitives, such as the circular edges and cylindrical holes in the workpiece of Fig. 1.)

For each contact formation in the library, so-called *process and measurement equations* must be derived. These equations are used in the on-line estimators, to link the state parameters of the models (i.e., the uncertain relative pose of tool and workpiece) to the pose, twist and wrench measurements, and to the propagation of the state during the executed motion. This is just the classical estimator construction, e.g., for Kalman Filters.

We have recently improved the user-friendliness of our contact estimators (see the following Sections) by making contact formation modelling easier in two ways:

- Our estimators allow *non-minimal* models for contact formations, derived by simple composition of the elementary contact primitives. Even if this results in more model primitives than strictly necessary. The goal is to facilitate the automatic derivation of estimators from a CAD programming environment.

- We developed symbolic representations of the above-mentioned measurement relationships. These symbolic representations allow the automatic generation of system and measurement equations for composite contact formations, and the derivation of the linearized versions of the filter equations (needed in, for example, the Extended Kalman Filter). The symbolic processing uses the Ginac library, `http://www.ginac.de/`.

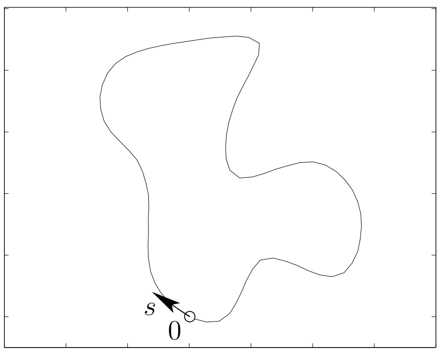
3 Non-linear estimation

The two most applied estimation algorithms in mobile robotics SLAM are Extended Kalman Filters, e.g. [1], and "particle filters", e.g. [7], for which many extensions have been developed. Kalman Filters are fast, but suffer from linearization problems; particle filters can cope more easily with non-linearities, but their computational costs can be high.

Our ACM research applies both Kalman Filters (e.g., [3]) and particle filters (e.g., [5]). But we have also developed a non-linear estimator that inherits many of the analytical advantages of the Kalman Filter, while still being able to converge on very non-linear systems, and from very erroneous instantaneous positions, [6]. Figure 2 shows the performance of this so-called *Non-minimal State Kalman Filter* (NMSKF) on the problem of estimating the contact position on a highly curved workpiece. In the experiment, a probe is tracking a *known 2D contour*, but the precise pose of the contour is not known at the start of the task. The goal is to *estimate the contact point position of the probe* on the contour, i.e., to estimate the arc length s, Figure 2 (left).

At different time steps, the absolute position of the probe with respect to some world frame $\{w\}$, is measured, Figure 2 (right). The position and orientation of the contour-object with respect to this world frame is unknown.

The localisation problem is made independent of the contour's position and orientation by identifying the contact point based on (i) the Cartesian distance between the successive probe positions and (ii) the curvature of the contour at the successive probe positions.



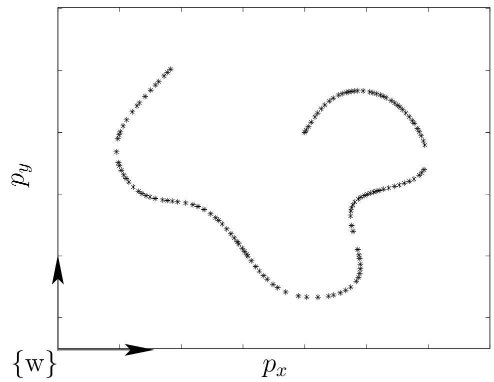


Figure 2: Contour model (left), and measured contact probe positions that are the inputs to the estimators (right).

Due to the highly nonlinear measurement model, the "classical" KFs (Extended Kalman Filter, Unscented Kalman Filter, etc.) cannot be consistent and/or informative unless started with a good initial estimate. We have solved this problem some years ago, [4], by starting multiple EKFs in parallel, each initialized with a different initial value. A *single* NMSKF, however, solves the problem, even for a very uncertain (because uniformly distributed) prior pdf $p(s_0)$. This solution is not only faster than the multiple-EKF approach, but it also scales better and is much easier to initialize.

Figure 3 shows the PDF over the contact point position s_k after $k = 1$, 50, 100 and 134 measurements. The star on the horizontal axis indicates the arc length at the real contour contact point. The top figure shows that the first measurement possibly originates from about any contact point. After 50 and 100 measurements, the probability is mainly concentrated in two regions. Note that after 100 measurements there remain still two possible matches, and the real contour contact point is not situated in the region with the highest probability. After 134 measurements, only one contact region (the correct one!) is still probable. These plots show the filter's excellent ability to cope with non-linearities, including the ambiguities generated by multiple possible matches.

4 Contact formation detection

More and more (industrial) robotics tasks are programmed off-line, using CAD models of the workpieces. These CAD models can in principle be used by the on-line controller to provide a priori information about the possible contact formations during the execution of a compliant motion.

Figure 4 shows the results of a Bayesian particle filter based contact formation estimator. That is, it shows how, during the task, the incoming sensor information adds probability mass to some contact formations hypotheses, while subtracting probability from others. This experiment uses *very poor* a priori information about the sequence of expected contact formations: the transition model is a Markov transition matrix, which gives constant but small probabilities of changing contact formations at each sample instant. Despite this very simple model, the estimator finds the correct contact formation quickly and reliably.

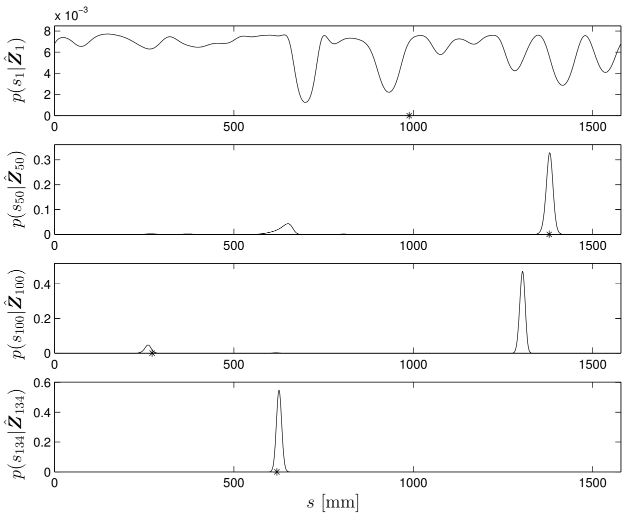


Figure 3: The PDF $p(s_k|\hat{\boldsymbol{Z}}_k)$ after 1, 50, 100 and 134 measurements. The star on the horizontal axis indicates the real contour contact point.

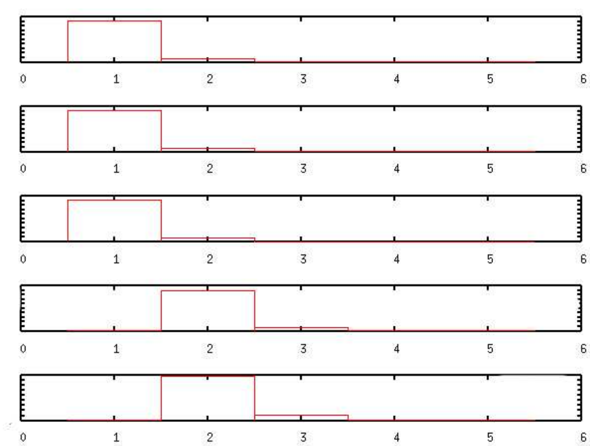


Figure 4: Identification of the current contact formation in the "cube-in-corner" experiment, with a Bayesian particle filter. The plot shows a histogram of the number of samples in a certain CF, at every change in CF (in this particular case, moving from a *vertex–plane* to an *edge–plane* contact). The horizontal axis is a discrete variable (1 = *vertex-plane*, 2 = *edge-plane*, 3 = *plane-plane*, 4 = *1 plane-plane + 1 edge-plane*, 5 = two *plane-plane*.) The vertical axis shows the number of samples in the current CF at a given instant in time. The simulation uses 500 samples. The different subplots indicate different times. The simulation shows that less than 5 measurements are necessary for all particles to "move" from one CF, which indicates that there's enough information contained in the measurements to compensate the poor system model.

5 Active sensing

Our approach to active sensing is to *optimise* the task plan (i) by *minimising* an objective function, such as the expected execution time, which is an important criterion in industrial applications; and (ii) by *constraining* the task plans to plans which observe the geometrical parameter estimates to a level of accuracy required by the task programmer.

The *task plan* consists of (i) the desired *sequence of contact formations* (CFs) and (ii) the *CF-compliant paths* to execute during each of these CFs. We are able to decouple the optimisation problem of the *sequence* of CFs, from the optimisation of the *paths* within each CF.

The active sensing is illustrated by the planning of several tasks with a cube as manipulated object and a corner object as environment object. The position and orientation of *both* objects are inaccurately known, i.e., the estimation problem is twelve-dimensional.

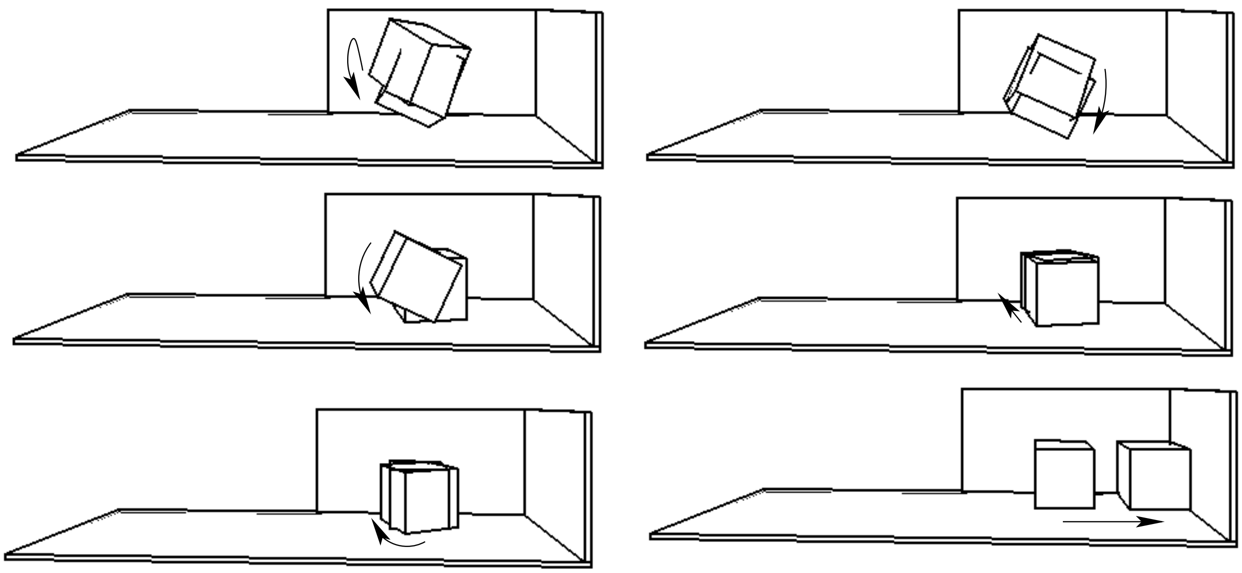


Figure 5: The automatically generated *shortest* CF sequence observing all geometrical parameters and performing a cube-in-corner assembly. First, the cube makes a *vertex-face* CF, an *edge-face* CF and a *face-face* CF with the horizontal face. Next, an *edge-face* CF with the rear face is added, followed by the rotation to a two *face-face* CF with the horizontal and rear faces. The cube is finally slid into the corner.

Figure 5 shows the automatically derived CF sequence which minimises the *number* of *CF-transitions*. The task plan needs to observe all geometrical parameters and perform a cube-in-corner assembly. Figure 6 shows the result of the optimal task plan (CF sequence plus transition motions) identifying all geometrical parameters when the *execution time* is minimised. The sequence is restricted to only free space motions and single *vertex-face CFs*. That is, the geometrical parameters are identified by “hopping” between different *vertex-face* CFs.

The active sensing motion to be executed *in* each CF in order to observe the geometrical parameters to the required accuracy is very short. Figure 7 shows the applied twist for the *face-face* plus *edge-face* CF of Figure 5. This active sensing motion identifies all observable parameters within two seconds.

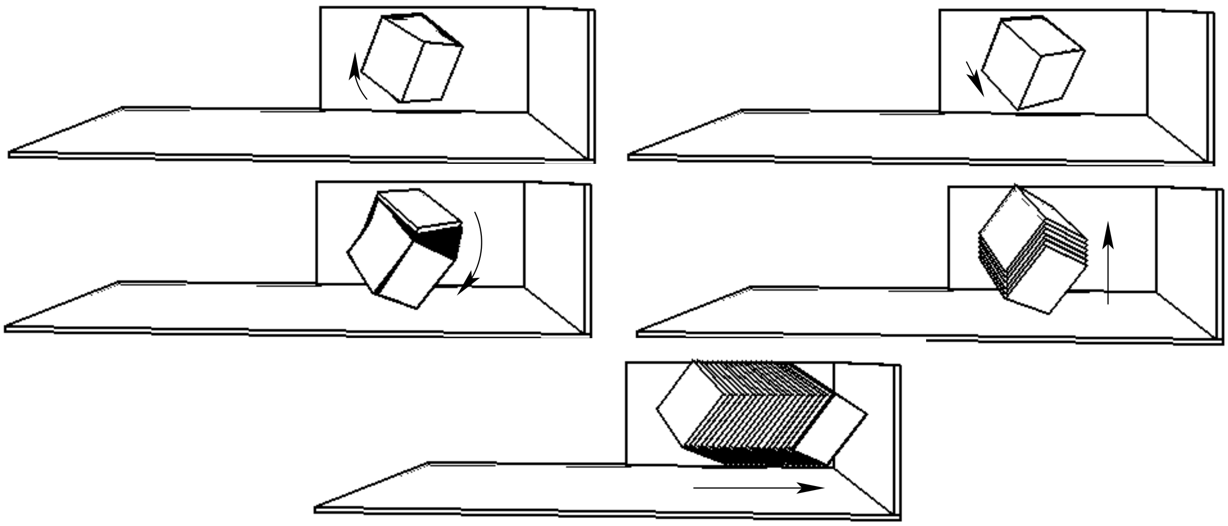


Figure 6: The automatically generated *fastest* CF sequence observing all positions and orientations of the cube and the corner object. Only free space and *vertex-face* CFs are allowed. *Vertex-face* CFs are made with the rear face, the horizontal face and the right face respectively.

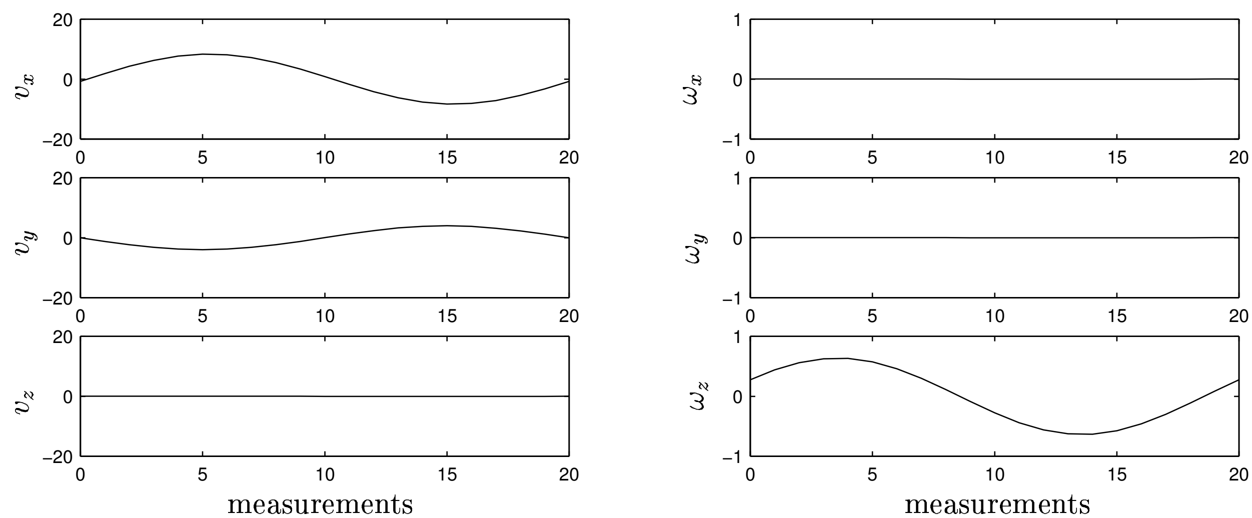


Figure 7: Active sensing motion in the *face-face* plus *edge-face* CF of Figure 5. The motion excites all observable geometrical parameters of the CF. The twist is defined in the 2D twist space of the CF and is limited to the (user-defined) maximum allowed kinetic energy of the manipulated object.

6 Model building in human demonstration

Model building in human demonstration consists of building a geometric model (of both the manipulated object and the environment) from the data that has been collected during a human demonstration. The assumption

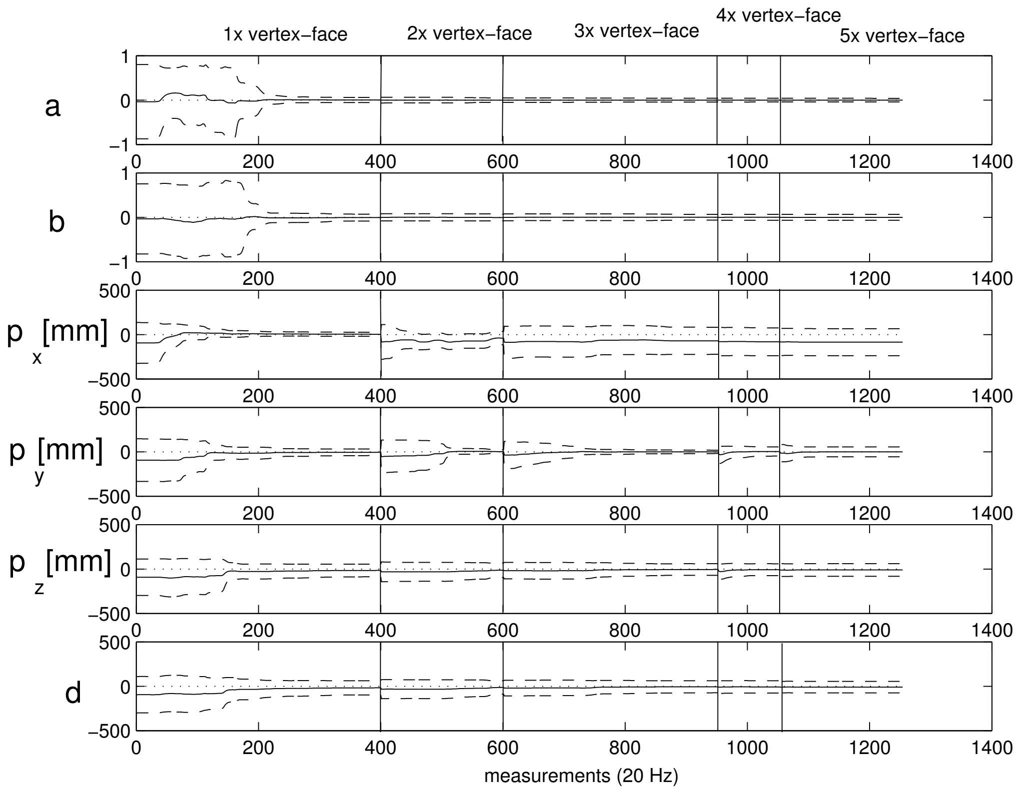


Figure 8: The output of the NMSKF on the data of a demonstrated "cube in corner" assembly, where the measurements come from the high-precision 6D optical measurment system K600 CMM (`www.krypton.be`). Estimates (full lines) for the parameters (a, b, d for the face and positions p_x, p_y, p_z for the vertex) describing the first vertex-face contact. The real values (dotted lines) and their corresponding uncertainty are visualized by two times the standard deviation (dashed lines).

being made is that all objects in the experiment are polyhedral, because this reduces the modelling work to the above-mentioned limited group of elementary contacts. The experimental data of placing "a cube in a corner" (Figures 5 and 6) was used in the derivation of the automatic model building of both the cube and the corner. (Of course, only those parts can be modelled that were involved in the contacts during the task execution.) The Non-Minimal State Kalman Filter was used in order to be able to handle large uncertainty on all the state parameters. It found five elementary vertex-face contact situations that are consecutively established as the cube moves from free space to the more constrained double face contact.

Figure 8 shows the convergence of parameters describing the estimation of the geometry of a single elementary contact: the position of a vertex on the cube and the horizontal plane of the corner. These parameters are the position (p_x, p_y, p_z) of the vertex in 3D and three parameters (a, b, d) representing the planar face $(ax + by + (1 - a - b)z + d = 0)$ of the environmental corner object. The estimations evolve during the expansion of the model from one vertex-face contact until five vertex-face contacts. Note that the contact primitives were autonomously selected by the estimator! This estimation of all the vertex-face contact parameters results in a (partial) model of the manipulated and environmental object by means of points and planes.

After filtering this redundant model could be reduced to a minimal model by examining the relationship between the different estimated points and planes. For example, by testing the hypotheses that the planes in two *vertex-face* CFs are the same. This hypothesis tests can be performed with the same Bayesian "particle filters" presented in a previous Section.

7 Software design

A really autonomous ACM system must *integrate* all the estimators discussed in the previous Sections (and some more!). *And* their implementations must be useable in real-time.

A large part of our ACM efforts are spent to provide a software framework to integrate these estimation algorithms; this framework is the Orocos software project for realtime robot control, [2]. Orocos provides a generic, component-based infrastructure for control, Fig. 9. The flow of information from one component to another is regulated through so called data objects.

Each component in this control structure must get application-specific "plug-ins." (The estimator component is the place where the ACM estimators of this paper will have to be plugged-in.) First, one defines the contents of the data objects. For a 6DOF robot with force sensor, the inputs are 6D joint position and velocity and the forces and wrenches exerted on the manipulators tool. The Estimator, Controller and Trajectory Generator components can access these data. The Estimator identifies the contact formation and the position of the object. The trajectory component adapts the generated position setpoints such that the robot follows the objects surface compliantly (*and* it can add active sensing when requested to do so). The Controller component calculates the velocity output and sends them to the 6D joint Actuator component.

As long as the exchanged data is fixed (which is generally true on a per application basis), components can be changed *at runtime* given that proper (re)initialisation sequences are identified for each change.

After the contents of the data objects have been identified, the components must be loaded with the correct algorithms (and drivers to interact with the hardware).

Besides the *data flows*, there is also a *control flow*, meaning that exceptions in components or end conditions must be propagated through the framework and delivered to the appropriate components. This "wiring" of events and the supervision is done by the Execution Engine, a configurable task executor which serialises concurrent tasks in such a way that data integrity is guaranteed and that data is processed in the correct order. An example is a force sensor overflow which triggers a controlled emergency stop (which influences most other components).

8 Conclusions

Our current research in Autonomous Compliant Motion has reached a state comparable to the *SLAM* research efforts in mobile robotics. Mobile robotics have been a large source of inspiration for our ACM developments, but we have also developed powerful nonlinear estimation techniques and distributed software implementations, which could trigger a strong cross-fertilization with mobile robotics.

The main emphasis of our current work is the application of the presented estimation and planning tools to the model building (from scratch) used to support robot programming of autonomous compliant motion tasks by human demonstration.

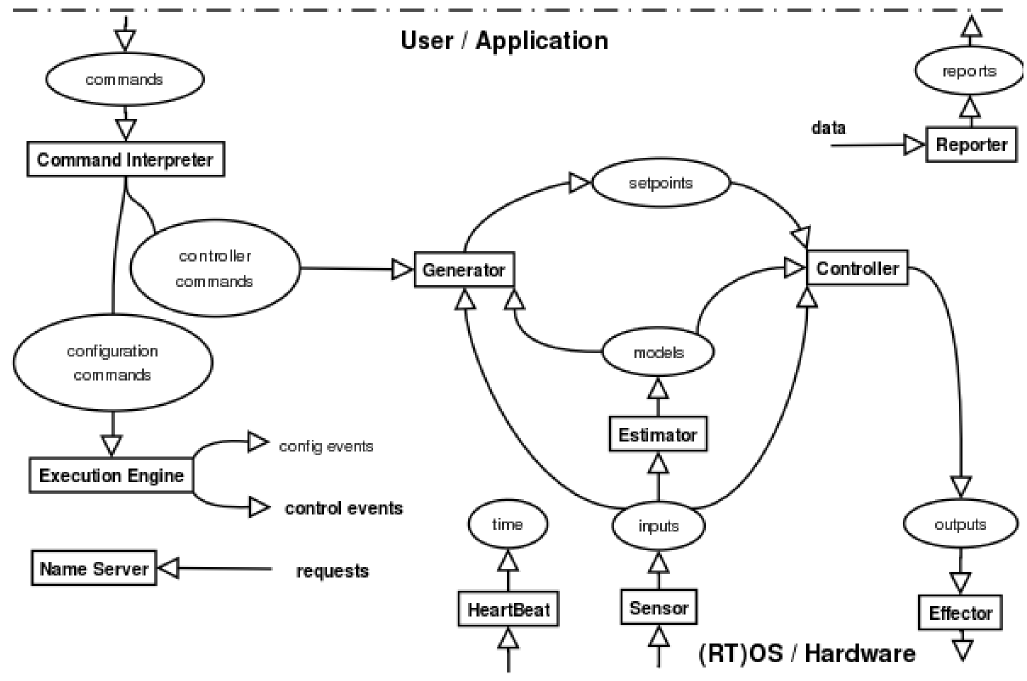


Figure 9: The *Software Pattern* for control. The rectangles are *Components*, the ovals are *Connectors*, which contain the data objects that must flow between Components.

References

[1] Y. Bar-Shalom and X.-R. Li. *Estimation and Tracking, Principles, Techniques, and Software.* Artech House, 1993.

[2] H. Bruyninckx. Open RObot COntrol Software. `http://www.orocos.org/`.

[3] J. De Schutter, H. Bruyninckx, S. Dutré, J. De Geeter, J. Katupitiya, S. Demey, and T. Lefebvre. Estimating first order geometric parameters and monitoring contact transitions during force controlled compliant motion. *Int. J. Robotics Research*, 18(12):1161–1184, 1999.

[4] S. Demey, H. Bruyninckx, and J. De Schutter. Model-based planar contour following in the presence of pose and model errors. *Int. J. Robotics Research*, 16(6):840–858, 1997.

[5] K. Gadeyne and H. Bruyninckx. Markov techniques for object localisation with force-controlled robots. In *Int. Conf. Advanced Robotics*, pages 91–96, Budapest, Hungary, 2001.

[6] T. Lefebvre, H. Bruyninckx, and J. De Schutter. A non-minimal state Kalman filter for nonlinear parameter estimation applied to autonomous compliant motion. In *Int. Conf. Advanced Robotics*, page 02PP093, Coimbra, Portugal, 2003.

[7] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5(3/4):253–271, 1998.