Towards the next generation of Autonomous Compliant Motion systems

Herman Bruyninckx, Tine Lefebvre, Lyudmila Mihaylova
Klaas Gadeyne, Ernesto Staffetti, Joris De Schutter

Dept. Mechanical Engineering, Katholieke Universiteit Leuven
`http://www.mech.kuleuven.ac.be/˜bruyninc`

Abstract

This paper gives an overview of our current research on force-controlled compliant motion. The goal is to build an Autonomous Compliant Motion (ACM) system of the next generation, i.e., with more autonomous sensing components and a less explicit, constraint-driven task specification. The ideas developed in the ACM context are useable for other areas too.

1. Introduction

We define an Autonomous Compliant Motion (ACM) system as a force-controlled robot, equipped with "high-level" sensor processing and task specification software modules. Our research of the previous decade developed "low-level" modules: the task specification is explicit, servo-oriented, and sub-optimal, [2]; the sensor processing consists of classical Kalman Filters estimating the contact formation's unknown geometric parameters on which the task specification relies (i.e., contact point positions and orientation of the contact normals), [5]. The limitations of this approach are (i) its lack of flexibility due to the explicit need for a task specification at servo level, and (ii) its validity for "small" uncertainties only.

This paper explains how we are currently working to tackle these limitations. The basic lines of this research are:

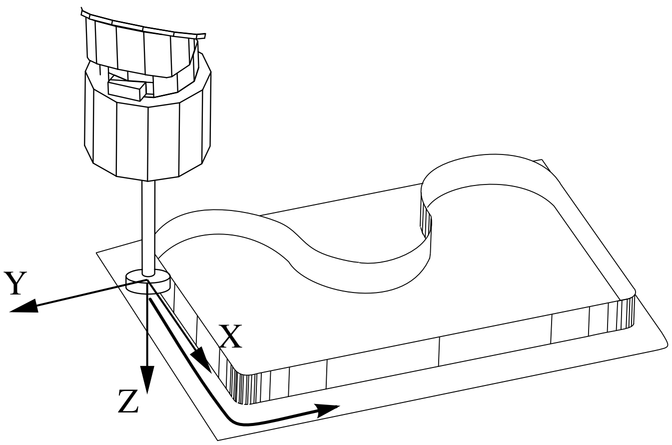
- the extension of the explicit task specification based on the "*Task Frame Formalism*" to an implicit, constraints-based specification. (Section 2.)
- the introduction of Bayesian estimators that are better able to cope with the inherent *non-linearities* of the compliant motion problem. (Section 3.)
- the development of an *Open Source* robot control software library, in order to be able to share, co-develop and compare advanced robotics research developments. (Section 5.)

2. Task specification

Up to now, the classical way to achieve on-line, continuous force feedback in a compliant motion task is to use some variant of Mason's *Compliance Frame* or *Task Frame*, [11]. A typical example is contour following, Fig. 1. In the scope of this article, we don't go into the details of the Task Frame Formalism (TFF), [2], but only focus on the limitations of the approach:

- The human task programmer must fully specify the six degrees of freedom of the task, at the servo level. We call this *explicit* programming.
- The TFF implicitly relies on the fact that the bases for the force and velocity controlled subspaces can be given by vectors along or about the axes of the Task Frame. This limits the compliant motion tasks that can be specified to the so-called *orthogonally decoupled* tasks. One simple example of a non-orthogonally decoupled task is the contact configuration with two vertex-plane contact, Fig. 2: the force and velocity bases change continuously during the motion and this time-dependence can, in general, not be specified explicitly and off-line by the human programmer.
- The TFF is not suitable for "intelligent" ACM, because it has no previsions for *active sensing* and *constraint satisfaction*: the robot controller should be allowed to deviate from the unconstrained path in order to (i) collect missing information about the geometry of the contact environment, and (ii) avoid collisions and singularities.

We experience that these drawbacks of the TFF more and more often severely limit our research on ACM: the autonomy of the ACM system *needs* more flexibility, in sensing, modelling and control. Hence, we started to design the "next generation" of compliant



move compliantly {
with task frame directions
xt: velocity v mm/sec
yt: force f N
zt: velocity 0 mm/sec
axt: velocity 0 rad/sec
ayt: velocity 0 rad/sec
azt: track (on velocities)
} until until distance $> d$ mm

Figure 1: *2D contour following*. The task frame's X axis is tangential to the contour; Y is the outward pointing normal. The right-hand side of the figure shows our textual specification for this task.

motion task specification formalism. The basic idea behind the new task specification is to change the current explicitly procedural formalism in the following directions:

- the *explicitness* of the specification: instead of programming the task by giving *setpoints* for all force, velocity or tracking directions of the Task Frame, the programmers should be able to use a geometric model or a task template in which they indicate *goal configurations* (i.e., contact formations and locations), and a set of *constraints* or *performance indices* that the on-line controller should take into account. These indices are *scalar* functions that quantify all aspects of the robot and/or the task. For example: the cost of allowed deviations from the unconstrained path, clearance around robot singularities and collisions, information gathering by on-line estimation, weighting of multiple estimation processes and sensing models, etc.

  In general, the task coordinator has to take into account indices with quite different origins simultaneously, which puts certain requirements of invariance and arbitrary choices on the mathematical representations of these indices. In our current research [16], we are carefully surveying and designing indices to be invariant and expressed in consistent physical units, in order to allow combination of indices for different subsystems in a coordinate-independent and "plug-and-play" manner. Obviously, the most appropriate framework to study this problem is differential geometry, because of its inherent emphasis on invariance and coordinate independence.

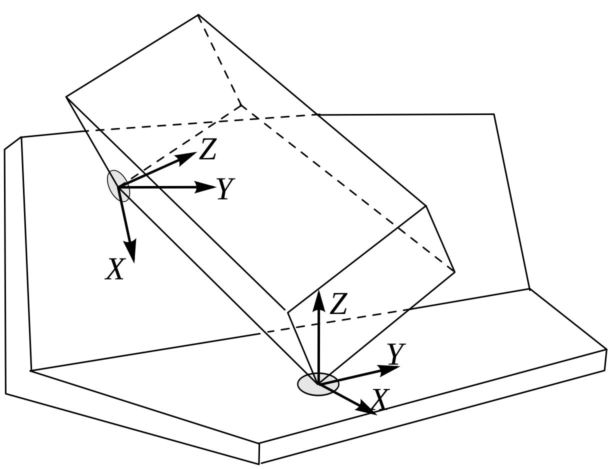


Figure 2: The double vertex-face contact formation as an example of a non-orthogonally decoupled compliant motion task. Each of the two contacts can be given a Task Frame specification, but the coupling between both is time-varying, and hence not programmable by one single Task Frame specification as in Fig. 1.

- the *behaviour* of the task control. The above-mentioned indices are needed because, in general, the users' specification determines only a subset of the available degrees-of-freedom of the robot. (Most of the indices will be defined by default; but users can of course override defaults when needed.) The instantaneous setpoints for the robot's servo are calculated on line, as the outcome of a *variational problem* that takes all explicit and implicit specifications into account in its "cost function." This approach is similar to, for example, impedance control or behaviour-based control, where the specification also determines the way in which the system has to *interact* with its environment, and not the interaction forces/velocities themselves.

  Our current research focuses on simplifying the general variational problem on a hybrid system model of compliant motion, such that the task controller can execute the algorithm on-line. No experimental validation has been realised yet.

- the *time horizon* of the task: most current implementations of control of redundant systems use pseudo-inverses of the Jacobian matrix of the robot in one way or another. This approach is *instantaneous*, in the sense that the setpoints are determined with a "horizon" limited to the next sample instant.

  Our current research wants to shift the horizon of the setpoint generation to much more than the next sample instant: in order to achieve, for example, good active sensing and singularity avoidance, the task controller should look for "global" optima instead of the local optimum of the pseudo-inverse. Of course, how global the optimization can be performed depends, among other things, on the (lack of) information the task coordinator has about the system.

- the *software engineering* of the task: the above-mentioned TFF extensions require a software framework that is adapted to the higher flexibility and modularity of the task control. Section 5. goes deeper into this topic.

3. Non-linear state estimation

Kalman Filters are currently the most popular on-line estimators, because their linear system equations allow for efficient implementations. However, many problems in compliant motion are quite nonlinear; for example, all coordinate transformations of rigid body representations are non-linear, except for pure translations. We tackle this nonlinearity problem (Sect. 3.1.) through various Bayesian methods (Sect. 3.2.), that are not limited to linear physical systems. The engineering challenge is to find efficient on-line implementations and clear interfaces to the task specification and active sensing modules of the ACM system (Sect. 4.).

3.1. Understanding non-linearities

Non-linearities come in many forms, and "disturb" the estimation process in many different places. One part of our research is to dissect the effects of non-linearities into a small set of separate causes, in order to be aware of (and to avoid when possible) most of the *ad hoc* solutions that have been given in the literature. There is first of all the well-known insight that Extended Kalman Filters introduce extra uncertainties because the they linearize the non-linear equations. However, we have identified some much less-known problems:

- Non-linearities in the *process equations*. Most on-line estimators use Gaussians to represent uncertainty, i.e., the mean is the *state estimate* and the covariance is the quantification of *uncertainty*. A dynamic system increases the uncertainty during its evolution. Estimators use the state and covariance update formulas of the classical Kalman Filter. However, Jensen's well-known inequality shows that

  $$f(E_p(X)) \neq E_p(f(X)),$$

  where f is the non-linear process update transformation on the coordinates of the stochastic variable X (e.g., the squared error), and E_p is the expected value with respect to the density $p(x)\,dx$. This means that representing information by Gaussians is *not* lossless under the process update. Note that, for example, Kalman Filters don't *require* the uncertainty to be represented by Gaussians, but their information processing uses mean and variance only, and these parameters fully determine a Gaussian distribution.

- Non-linearities in the *measurement update*. The state estimate is continuously updated by taking into account the information from new measurements. Most often the relationship between measurements and state is non-linear. Hence, a similar loss of information as in the previous paragraph is introduced. However, it is useful to make the distinction between process and measurement updates, because the former involves a mapping within one single space (configuration space), and the latter is a mapping between two different spaces.

- Non-linearities in the *stochastic and state manifolds*. Information about a certain system is, in the Bayesian framework, represented by a probability density function (PDF) $p(x)\,dx$ over the the configuration space of the system. Most estimators consider only the "$p(x)$" part of the PDF, and forget about the "dx" part (density). However, coordinate transformations over x introduce changes in the density too!

  Another effect in this category comes from the fact that the configuration space is *not* a Euclidean space (and hence not "flat"). And robot positions and orientations are such non-Euclidean space! This means, for example, that the classical concept of "estimation error" is not well-defined: calculating the error between the measured position and orientation x_i, and the estimate $\hat{x}$ as $x_i - \hat{x}$ violates the fact that the manifold SE(3) of positions and orientations is not a vector space, but a *multiplicative* Lie group. The linearization $x_i - \hat{x}$ is hence only valid for "small" errors, and is a member of the tangent space se(3) to SE(3). Hence, traditional linearization introduces more loss of information than most people are aware of, [14].

3.2. "Solving" non-linearities

One way of "solving" the problems with non-linearities has been to replace the Kalman Filter equations with alternatives that aim to be more robust, e.g., the UKF of [9]. The idea is to replace the covariance matrix in one space by a selection of sample points that give the same mean and covariance matrix, then to transform these points through the non-linearity, and find the sample covariance matrix after the transformation. This is an extreme limit of sampling based methods, and, as we found out, one with arbitrariness in the choice of sampling points.

Another approach is to start many similar filters, each starting in a different part of the configuration space and evolving to their local extremum, and/or using a weighted average of the estimates of the individual filters. We have experimented with such an approach too [4, 6, 10, 12, 13], with predictably better performance. But this method is a *brute force* approach without using any information about the non-linearities.

Our current research follows two other paths:

- transformation to a linear representation, in general in a higher-dimensional space. A similar idea has been applied in numerical linear algebra before, and consists of solving the estimation problem on a higher dimensional space; for example, in the case of rigid body location estimation, this higher-dimensional space is the 12-dimensional space of the coordinates in the rotation matrix (9) and the position vector (3) in the homogeneous transformation matrix that represents the body's position and orientation. Many estimation problems happen to have an (almost) linear form when using these coordinates; the drawbacks are that the estimation problem is of higher dimension, and that the estimation algorithm must make sure to enforce the non-linear constraints that apply between these coordinates.

- investigate more general Bayesian methods, in the sense that they don't rely on linearity and/or Gaussian assumptions: the state vector contains *all* information gathered in the past, modelled with, for example, *parametric models* (such as sums of Gaussians, exponential distributions, ...) or *non-parametric models* (such as grid-based methods, Monte Carlo methods, ...). Some of these methods have already been used in localization algorithms for mobile robotics, e.g., [18]. They are confronted much more with the "curse of dimensionality," because compliant motion is more of a six degrees-of-freedom problem than mobile robotics, which is mainly three degrees-of-freedom.

  Our experiments in this direction give results that are quite satisfactory as far as the estimation results go, [7], but they need more work to improve (i) the computational efficiency to soft real-time performance, and (ii) the general applicability of the algorithms as a stand-alone component (Sect. 5.).

4. Active sensing

One of the major flexibilities that are needed in an ACM system, and that an explicit task description cannot give, is to make *active sensing* possible: the on-line task coordinator must have the capability of doing actions with the sole purpose of gathering missing information needed for a reliable execution of the task.

Our current research tackles the active sensing problem from different sides:

- software engineering of the task coordinator. (See Sect. 5..) The goal is to allow a much more dynamic scheduling of different sensing, task monitoring and decision processes. The task coordinator is being designed as the final authority in taking decisions about the control setpoints, but it should rely heavily on the advice of all other components that have relevant information.

This software engineering problem turns out to be much less supported by proven "Software Patterns" than most of the other components in an ACM system (or any general robotics control software for that matter), Sect. 5..

- formulation of the task specification as a variational problem on a hybrid system. (See Sect. 2.) The search for missing information is driven by, in general, several performance indices, that contribute to the "Lagrangian" underlying the variational problem.

  The hybrid nature of the ACM comes from different origins: the contact configurations can change; the task coordinator can schedule several distinct "modes" requiring a different set of control algorithms; the available sensors come and go in and out of an "active" state in which they provide model parameters and advice; etc.

- *ad hoc* extensions to the TFF, to allow non-instantaneous setpoint determination, as a special case of "design of experiments," [15, 17]. This approach uses simple linear estimators, a performance index on the covariance matrix, and a predefined parameterized family of available active sensing actions, [3].

The first two approaches have had no experimental validation yet.

5. Open source robot control software

An ACM has much higher software support needs than a TFF-based compliant motion system. The latter basically only adds a force sensing input to the motion servo, but the former relies on several software "components" that deliver "services": each of the performance indices has its own component, responsible for calculating the value of the index whenever the task coordinator needs it; that task coordinator also needs support in its decision making and "global" planning; each sensor requires more processing than in the TFF case, because the data is now interpreted in the context of a "high-level" task model; the flow of information through the system is not hierarchical and constant over time, because the same sensor data is used and generated in different places; processing often occurs asynchronously (i.e., not in sync with the generation of the data by the physical sensors) because estimation and task planning require more time than is available (or needed) in the real-time of the servo control; etc.

Over the last three years or so, the progress of our research has been hindered by the software limits of the classical hierarchical control. Since no sufficiently advanced robot control software can be bought from the robot vendors, we decided to start developing the advanced software framework ourselves. However, the success and relevance of the Open Source software development model made us decide to follow the road of open co-development with other (predominantly) research groups. The results is the *Orocos* (Open source RObot COntrol Software) project, [1], that had its official kick-off in September 2001.

The Orocos project is still in its infancy (i.e., it has not yet produced much more than device driver code), but has had a very active design period, from which concrete frameworks are appearing for several basic robot control components. The frameworks aim at offering implementations of the "Software Patterns" [8] that have matured in the robotics community during the last couple of decades, and that have been identified during the design phase of the project. The project has decided to build the frameworks for the following areas: motion control, kinematics and dynamics, task specification, and intelligent sensing. The emphasis is on building components, not on implementing one single architecture.

6. Conclusions

This paper has presented our current research efforts towards an Autonomous Compliant Motion system. The focus points are on: (i) more advanced estimation (where "advanced" means that non-linearities and "global" optimization are taken into account), (ii) a next generation of task specification (based on task-directed models instead of servo-level setpoints), and (iii) the development of the software support (an ACM cannot perform its task in the quite hierarchical control frameworks of past and current robot controllers).

Acknowledgments

The authors acknowledge the support of the Fund for Scientific Research—Flanders (F.W.O.), the K.U.Leuven's Concerted Research Action GOA/99/04, and the Center of Excellence BIS21 grant ICA1-2000-70016 of the European Union.

References

[1] H. Bruyninckx. Open source robot control software. `http://www.orocos.org/`.

[2] H. Bruyninckx and J. De Schutter. Specification of force-controlled actions in the "Task Frame Formalism": A survey. *IEEE Trans. Rob. Automation*, 12(5):581–589, 1996.

[3] J. De Geeter, J. De Schutter, H. Bruyninckx, H. Van Brussel, and M. Decréton. Tolerance-weighted L-optimal experiment design: a new approach to task-directed sensing. *Advanced Robotics*, 13(4):401–416, 1999.

[4] J. De Geeter, H. Van Brussel, J. De Schutter, and M. Decréton. Local world modelling for teleoperation in a nuclear environment using a Bayesian multiple hypothesis tree. In *Int. Conf. Intel. Robots and Systems*, pages 1658–1663, Grenoble, France, 1997.

[5] 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.

[6] 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.

[7] 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.

[8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, Reading, MA, 1995.

[9] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. Amer. Control Conf.*, pages 1628–1632, Seattle, WA, 1995.

[10] T. Lefebvre, H. Bruyninckx, and J. De Schutter. Estimation and propagation of geometrical parameters during force-controlled execution of polyhedral contact formation sequences. In *Int. Conf. Advanced Robotics*, pages 85–90, Budapest, Hungary, 2001.

[11] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-11(6):418–432, 1981.

[12] L. Mihaylova, H. Bruyninckx, J. De Schutter, and E. Staffetti. Planar contour tracking in the presence of pose and model errors by Kalman filtering techniques. In *Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 329–334, Baden-Baden, Germany, 2001.

[13] L. Mihaylova, T. Lefebvre, E. Staffetti, H. Bruyninckx, and J. De Schutter. Tracking contact transitions during force-controlled compliant motion using an interacting multiple model estimator. In *Int. Conf. Advanced Robotics*, pages 665–670, Budapest, Hungary, 2001.

[14] X. Pennec and J.-P. Thirion. A framework for uncertainty and validation of 3-D registration methods based on points and frames. *International Journal of Computer Vision*, 25(3):203–229, 1997.

[15] F. Pukelsheim. *Optimal Design of Experiments*. Wiley, New York, NY, 1993.

[16] E. Staffetti, H. Bruyninckx, L. Mihaylova, and J. De Schutter. Kinematic and stochastic performance indices for autonomous manipulation. In *IEEE Int. Conf. Robotics and Automation*, Washington DC, U.S.A., 2002.

[17] J. Swevers, C. Ganseman, D. Bilgin, J. De Schutter, and H. Van Brussel. Optimal robot excitation and identification. *IEEE Trans. Rob. Automation*, 13(5):730–740, 1997.

[18] 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.