Active Compliant Motion: A survey.

Tine Lefebvre*, Jing Xiao†, Herman Bruyninckx*, Gudrun De Gersem*

keywords: compliant motion, contact modeling, fine-motion planning, contact identification, force control

Abstract

Whether they are asked to polish or assemble parts, clean the house or open doors, the future generation of robots will have to cope with contact tasks under uncertainty in a stable and safe manner. Obtaining a controlled contact motion under uncertainty is still a major challenge for the robotics community. At present most research groups focus on one of the subcomponents (i.e., modeling, planning, estimation or control) of the system, and no overall system is developed yet. This paper presents a literature survey of the state-of-the-art of the subcomponents and points to the need for effective integration of those components.

1 Introduction

Compliant motion allows a robot or an object held by a robot to comply to the interaction forces generated by its contact with the objects in an environment. Such kind of motion is necessary to reduce or overcome the uncertainties associated with the objects in contact, in terms of the objects' geometric shapes, relative locations, i.e., contact configurations, and the types of contact, i.e., contact states, in order to ensure successful and safe operations of the robot. A compliant motion is active if it is realized by force-based feedback control, and not (solely) by putting a mechanical compliance between the robot and the environment. The latter achieves passive compliance. While passive compliance can be designed to be very effective for specific tasks, the capability for active compliant motion allows a robot the flexibility to deal with a much broader range of tasks where compliance is required.

However, despite decades of academic and industrial research in robotics, only a very small number of existing robots are capable of very limited active compliant motion when interacting with an environment. A key problem seems to be that existing research effort is mostly focused on certain subproblems or components related to active compliant motion, but there is a lack of effort in integration between different components crucial to the success of a general active compliant motion system.

* Katholieke Universiteit Leuven, Department of Mechanical Engineering, Division PMA, Celestijnenlaan 300B, B-3001 Heverlee, Belgium.
† University of North Carolina - Charlotte, Department of Computer Science, College of Information Technology, Rm 317, Cameron Applied Research Center, Charlotte, NC 28223, USA.
An active compliant motion system generally requires the following active components: a planner for planning compliant motion commands, an estimator for synthesizing sensory information, identifying contact states and other state information in task operation, and monitoring state transitions on-line, and a controller for executing the compliant motion commands with both the low-level feedback provided by sensors and the high-level feedback provided by the estimator. They interact via a passive component: the models of the contacting objects, which include the geometric and topological information of the objects, their contact states, the possible forces and compliant motions that can occur in each contact state, and the possible transitions between such states that a compliant motion can generate.

This paper surveys the state-of-the-art and the ongoing developments in each of those components: modeling (Section 2), planning (Section 3), estimation (Section 4), and control (Section 5) for active compliant motion. At present no fully autonomous compliant motion system incorporating all the above components has been developed yet because of the inherent complexity involved in even developing some of those components. There do exist working compliant motion systems, but the human developers have to replace one or more of the missing components manually or to provide the missing intelligence in these components. This survey hopes that, by presenting the current landscape of the field, it facilitates more rapid advancements of not only the components needed in an active compliant motion system but also the reliable and effective integration of such components in intelligent general systems useful for many applications requiring active compliant motion.

2 Modeling

A major issue of modeling for compliant motion is how to model and characterize a contact state. In the current literature, contact states are described in a variety of ways or combination of ways based on their use and convenience, such as topological representation in terms of contacting topological surface elements (Section 2.1), force/motion degrees-of-freedom representation in terms of the possible forces and velocities in the contact (Section 2.2), or constraint representation in terms of constraints on the locations of geometric features of the contacting objects (Section 2.3). “Intelligent” compliant motion systems will make use of all these models (and possibly more) in some integrated way; it is too soon, though, to describe clearly what the most appropriate model or combination of models is for a given purpose.

2.1 Topological contact states

By referring to a “contact state”, we usually mean a state of higher level than what the configurations of the contacting objects describe, since these objects may maintain the same contact characteristics while varying their configurations to some extent and such common contact characteristics are often what one really cares about. For example, when we say “put the cup on the table”, we care about the proper contact between the cup bottom and the table top rather than some exact relative geometric
configuration between the two objects. Moreover, state information at the finest level, such as the exact object configurations, is often impossible to obtain in the first place due to insufficient sensor information, and to the inevitable uncertainties in their measurements. Indeed, almost nobody describes a contact at such lowest level, and the aforementioned topological, force, and geometrical-constraint representations are all higher-level representations of discrete contact states.

One most commonly used topological representation was the point-contact notion for polyhedra [1], where point contacts, for example, vertex-edge contacts in 2D polygons and vertex-face, edge-edge contacts in 3D polyhedra, were used as contact primitives, and a contact state was described as a set of such primitives. Many earlier contact motion strategies assumed this representation. Another topological representation was introduced by Desai et al. [2, 3] where they used the single contacts between a pair of topological surface elements (i.e., faces, edges, and vertices) of two polyhedra as primitives and described a contact state (between the two polyhedra) in terms of the set of such pairs of elements in contact, called contact formation. Here a contact primitive can characterize a contact region that is either a point, a line segment, or a planar face, unlike the aforementioned point-contact notion. From the viewpoint of contact recognition (i.e., sensing), however, both representations can result in states which are different by definition but indistinguishable in recognition due to uncertainties. Figure 1 shows a simple example.

Xiao [4, 5] introduced yet another kind of contact primitives called principal contacts (PCs). A PC denotes a contact between a pair of topological surface elements that are not boundary elements of other contacting topological surface elements. The boundary elements of a face are the edges and vertices bounding it, and the boundary elements of an edge are the vertices bounding it. A contact state between two polyhedra can then be characterized as a set of PCs formed, also called a contact formation. PCs are highest level contact primitives so that different contact states described in terms of PCs can be distinguished more effectively with force/moment sensing. Indeed, different PCs between two objects correspond to different degrees of freedom of the objects, which often also correspond to significant differences in the contact forces and moments. As shown in Figure 1, the indistinguishable states in terms of the other contact primitives are grouped as a single contact state in terms of PCs. Thus, there are also fewer number of contact states in terms of PCs, leading to a more concise characterization of contact states.

Automatic generation of contact state graphs are made easier with the representation of contact states in terms of PCs. Xiao and Ji [5] took advantage of the different degrees of freedom of neighboring
 PCs and contact formations and devised an approach based on contact constraint relaxation to generate contact state graphs in terms of contact states and their adjacency relations automatically. Recently, Pan and Schimmels [6] introduced a method to generate automatically valid contact states in terms of PCs between two polyhedra by optimizing nearby relative configurations to reach certain valid contact configurations corresponding to these states.

More recently, Luo et al. [7] proposed an approach to extend the notion of contact formations in terms of PCs to non-polyhedral objects.

2.2 Contact models based on force/motion degrees-of-freedom

In this category, a major approach towards modeling contact states is very much biased towards the capabilities of a controller. It models a contact state by means of a so-called Task Frame or Compliance Frame, in which it also specifies set-points for the controller. A Task Frame is an orthogonal reference frame, placed at an appropriate location in the contact, such that the available degrees of freedom of the objects in contact can be modeled as either pure motion freedoms or pure contact forces along one or more of the frame’s axes [8, 9]. Figure 2 gives a simple example.

The major advantage of a Task Frame model is that it is efficient for execution: the compliant motion specification consists of given set-points in the above-mentioned force- or motion-controlled directions. In addition, coming up with an appropriate Task Frame is not too difficult for most simple contacts. We refer to [9] for an extensive catalog of Task Frame models and specifications.

The main disadvantage of the Task Frame model is that it cannot model all possible contact situations, even for contacts between polyhedral objects; difficulties sometimes arise when two or more primitive contact situations (such as PCs – see Section ??) occur at the same time. The reason is that force controlled and velocity controlled directions are not necessarily orthogonal but reciprocal [10, 11]. The assumed orthogonality makes most of the above mentioned hybrid controllers non-invariant. Considerable work [12, 13, 14, 15, 16, 17] has described motion/force models for hybrid control correctly as acting on the reciprocal force controlled and velocity controlled directions.

Another approach towards modeling the contact states is by defining the set of inequalities repre-
senting admissible forces (wrenches) and displacements (twists). The main advantage of this approach is that such a contact description expresses the unidirectionality of the contact constraints. The main disadvantage of this approach is that the solution to the set of inequalities is not given in an explicit form, even though they are linear.

A big step forward was put by Hirai and Asada [18]: they noticed that the set of linear inequalities corresponds to the mathematical notion of polyhedral convex cones [19]. This gives a better insight into the problem and allows us to produce simple and systematic solutions to the equations.

### 2.3 Contact models based on geometrical constraints

The previous two category of modeling approaches are purely constructive, in that the natural way for a planner to use such a model is to derive six set-points from the model, one for each degree-of-freedom, either in force or in motion. These set-points, together with the model, fully determine the motion of the contacting objects. In addition, the degrees-of-freedom are modeled explicitly: a basis for the motion degrees of freedom is modeled, as well as a basis for the possible ideal contact forces.

There is a complementary, implicit way to model a contact state and specify constrained motion between objects in contact. This implicit approach works with geometrical features on the contacting objects, such as the ones mentioned in Section 2.1 for polyhedral objects: edges, vertices, faces. Contact modeling is then done by specifying constraints among features: coincidence, limits on relative distances or velocity, etc. Many researchers formulated geometrical contact constraints to characterize or supplement the representation of different types of contact states mainly for identification purposes [20, 21, 1, 22, 23, 24, 25, 26].

Such contact models alone often do not explicitly provide degrees of freedom of a motion under a contact state except for some simple cases. They also do not specify set-points for motion and force explicitly. All such information needed by a low-level compliant motion controller has to be obtained by solving the constraint equations or inequalities.

Samson, Le Borgne and Espiau, [27] described the constraint-based approach in a quite general and systematic way. Their method integrates modeling and planning tightly – both were specified as constraints, but they did not provide working implementations of force-controlled compliant motion.

### 3 Planning

The earliest motion strategies deliberately take advantage of compliant motions and constraints are those devised for dealing with peg-in-hole tasks of high-precision requirement [28, 29]. Later, Lozano-Pérez, Mason, and Taylor [30] proposed the concept of preimages in configuration space (C-space) [1] to compute motion strategies that would not fail in the presence of uncertainties for general fine motion planning, i.e., motion planning that takes into account the effects of uncertainties. A preimage encodes those configurations from which goal attainment is both guaranteed and recognizable. Further
development was made by Erdmann [31] in improving computability and by Donald [32] in considering modeling uncertainties of objects. In this approach, compliant motions are preferred wherever possible because they typically produce larger preimages [30, 31] than pure positioning motions. However, the computation of preimages requires that the C-space obstacles (C-obstacles) be given because preimages include configurations on the boundary of C-obstacles or contact configurations, especially if compliant motions are involved.

Hopcroft and Wilfong [33] proved that if two arbitrary objects in contact can be moved to another configuration where they are also in contact, then they can be moved from the first contact configuration to the second in such a way that the objects remain in contact throughout the motion, i.e., there is always a path of contact configurations (on the boundary of C-obstacles) connecting the initial and goal contact configurations. Therefore, not only compliant motion is desirable in many cases, but it is always possible given the initial and goal contact configurations.

However, because compliant motion occurs on the boundary of C-obstacles, planning compliant motion poses special challenges not present in collision-free motion planning: it requires the exact information of such contact configurations on the boundary of C-obstacles. Unfortunately computing C-obstacles exactly remains a formidable task to date. While there are exact descriptions of C-obstacles for 2-D polygons [34, 35], there are only approximations for 3-D polyhedra [36, 37].

Many approaches are introduced to either get around the problem of computing the C-space obstacles or reduce the dimensionality and scope of the problem when compliant motion is involved.

In fine motion planning, Xiao and Volz [38, 39, 40] introduced a systematic replanning approach to deal with the effect of uncertainties in planning motions for high-precision assembly tasks. The approach simplifies the planning problem by decomposing it into two phases: global and off-line nominal path planning assuming no uncertainty and local patch-planning based on sensing and contact analysis to provide remedy motion plans (called patch-plans) for a motion stopped by unintended collisions due to uncertainties. It is patch planning that could generate plans that involve small-scale compliant motions of translation or rotation in addition to guarded motions. This two-phase approach relies on on-line identification of local contact state information (see Section 4) and the knowledge of goal state of the task to decide on-line compliant paths.

Other researchers also proposed different variations of the two-phase approach. Dakin and Popplestone [41, 42] proposed an approach to plan compliant motions by exploring local contact space described as an adjacency graph of contact states around certain critical configurations from a given nominal motion plan. It is not clear though how the critical points were selected. Rosell et al. [43] introduced a two-phase approach for planning planar part assembly motions. They first obtained an exact cell decomposition of free configurations based on critical contact configurations in the configuration space and planned a nominal assembly motion by searching the graph. Next they evaluated the nominal path taking into account the effects of uncertainties and modified certain free-space path segments to be contact space path segments for compliant motion.

Buckley [44] developed a strategy to generate translational compliant motion plans as a search
through a state graph derived from the translational subspace of the C-space of the task environment. He partitioned such a space into states and used the arcs connecting two states to indicate compliant translation commands. More recently, Sacks [45] introduced a path planner for planar articulated robots in a planar environment that used compliant motion wherever needed to push (movable) obstacles away or to navigate narrow passages. Compliant motion segments were determined by searching paths along the contact patches on 3-D C-obstacle surfaces.

The general idea of planning compliant motions by searching certain contact state graph has been adopted by researchers in different ways. McCarragher and Asada [46] modeled an assembly task of 2-D polygonal parts as a discrete event system using petri nets, where the contact states and transitions were identified manually. They used a discrete controller to compute and issue velocity commands to guide the compliant motion for assembly towards the desired goal while avoiding unwanted transitions.

Sturges and Laowattana [47] developed a spatial remote center compliance device (SRCC), extending Whitney’s RCC device [29], to accomplish square peg-in-hole task. They identified all the possible contact states manually and derived the contact constraints together with the constraints for avoiding jamming and wedging. A compliant motion strategy was then manually identified from the contact states and constraints (arranged in a constraint network), which would be successful with the use of SRCC despite of uncertainties.

Many other researchers also used the knowledge of certain contact state graph to derive compliant motion strategies for various purposes [48, 49, 50, 51, 52] or for recognition of contact states (see Section 4). However, the information of these graphs is usually fed manually into a system as input, i.e., contact states and the relations among contact states are enumerated and presented to the system manually. This is awfully tedious for even tasks of simple geometry [47] and is practically infeasible for complex tasks due to the huge number of different contact states.

Hirukawa, Papegay and Matsui [53] were the first to generate a contact state graph between two convex polyhedra automatically by enumerating all possible contact states and their connections. They then used such a graph to plan compliant motions, which was done by first searching a sequence of state transitions connecting the start and the goal contact states in the graph and then obtaining the path of contact configurations corresponding to the sequence of state transitions by solving algebraic equations. However, their algorithm was restricted to convex polyhedra. Hirukawa and Papegay later reported a new algorithm for polygons in contact [54].

More recently, Xiao and Ji introduced a general divide-and-merge approach for automatically generating a contact state graph between two arbitrary polyhedra [5]. Each node in the graph denotes a contact state, described by a topological contact formation (CF) [4] (see Section 2) and a configuration satisfying the CF, and each edge connects the nodes of two neighboring contact states. Built upon this work, they later decomposed the problem of contact motion planning into (1) high-level: graph search for state transitions from one node to another in the automatically built contact state graph, and (2) low-level: contact motion planning within the set of contact configurations constrained by the same contact state, called CF-compliant motion planning. A general contact motion plan crossing several
contact states can be considered as consisting of segments of CF-compliant motions in different contact states. Ji and Xiao [26] subsequently introduced an approach to plan CF-compliant motions based on random sampling of CF-compliant configurations and extending the PRM motion planning technique [55].

Meeussen et al. [56] extended the approach for automatic generation of a contact state graph between arbitrary polyhedra [5] by considering that one of the polyhedra was held by a robot manipulator. Therefore, manipulator constraints were taken into account in evaluating whether a contact state or contact state transition was possible.

Based on the information of a contact state graph, Lefebvre [57] introduced methods to plan sequences of contact state transitions and compliant motion strategies that were optimized for active sensing to determine uncertain geometric parameters while achieving certain goal contact formation.

4 Estimation

The estimation component of an autonomous compliant motion system improves the knowledge of the robot during task execution by various complementary activities: (i) estimating geometrical parameters, i.e., poses and dimensions of the contacting objects (Section 4.1), (ii) estimating dynamic contact parameters such as the impedance (Section 4.2), (iii) monitoring contact state transitions, and/or (iv) recognizing the new contact states at runtime (Section 4.3).

4.1 Estimation of geometrical parameters

The geometrical parameters in a contact state are the positions and orientations of the vertices, edges, faces, etc., that are involved in the contact state. The more accurately that these parameters are known in the contact model, the smaller errors will be generated by the robot controller. In most practical situations, the uncertainty on the geometrical parameters must be within certain bounds, because otherwise the controller will command motions that will either lose contact, or generate too high contact forces.

The estimation of the contact location and orientation is often applied to the most simple contact state, i.e., the type of single vertex-surface contacts only. Only a few authors considered more complex contact situations. Unless otherwise mentioned, the estimation techniques usually were deterministic and considered only the last measured data or the batch of all data.

Ipri and Asada [58] estimated the instantaneous rotation center of a manipulated object making multiple contacts with the environment; Mimura and Funahashi [59] estimated the pose of the contacting features for vertex-face, edge-face and face-face contacts; Yoshikawa et al. [60] estimated the pose of a polyhedral environment object by touching it with a known manipulated object; McCarragher and Austin [61] estimated the contact normals at the different contacts between polyhedral objects based on a user-defined weighting between the previous estimate and the measurement; and Eberman [62]
estimated the rotation of the constrained/free space of contacts for which the configuration space model was known.

The most general contact cases between rigid objects with generally curved surfaces were considered by the research group of De Schutter and Bruyninckx [63, 64, 65, 66] and by Debus, Dupont, and Howe [67]. The former research group modeled contacts by a Virtual Contact Manipulator, describing the relative degrees of freedom between the contacting objects in function of the geometrical parameters. Identification of the parameters was based on pose, wrench and twist measurements using different Bayesian estimation techniques [65, 66]. Debus et al., on the other hand, applied a deterministic nonlinear least-squares estimation method based on a fixed-length moving data window. The estimates were calculated based on the penetration distance between the contacting objects using pose measurements.

4.2 Estimation of dynamic environment property

Several applications of robotic systems benefit not only from knowing the geometrical contact parameters, but also the dynamic properties of the interaction between a robot and the environment. Contact stability for compliant motion using impedance control is highly dependent on the environment dynamics. Without a good knowledge and model of the environment, accurate force tracking is also a challenge. Direct or indirect estimation of the stiffness of the environment then may be implemented to provide appropriate adaptation of the control gains. Another application is situated in the field of teleoperation with kinaesthetic feedback. Accurate estimation of the environment properties, and direct feedback of these, e.g., through adaptive impedance control at the master side, can yield an extremely flexible way to provide high fidelity kinaesthetic feedback.

Several adaptive schemes with inherent dynamic parameter estimation were proposed. Using a Model Referenced Adaptive System (MRAS), Yabuta et al. recursively estimated the dynamic parameters of the combined system manipulator-environment out of the error between measured interaction force and expected force, the latter based on the position/velocity input and the prior estimation [68]. This approach assumes accurate knowledge of the characteristics of the servomechanism, in order to isolate the parameters of interest from the identified parameters. Though the estimates are moderate, especially for softer environments, stability of the servomechanism is assured for a proper choice of the adaptive feedback gain, and the a reference force is tracked well.

Also based on the measurement of the interaction force and the manipulator input, in this case the input force, the adaptive estimation was proposed in [69]. Here, the manipulator, which is the slave of a teleoperation system, is equipped additionally with an active observer as proposed in [70]. This gives an estimate of the state (including force and derivative of the force), based on only force measurement data, input (force) signals, and prior estimates of the system stiffness. Based upon the difference between the measured and estimated interaction force, the stiffness estimate is updated. The adaptation algorithm is relatively complex, and requires tuning of numerous parameters. The experiments show fast convergence of the stiffness estimations, even without persistent excitation. Unfortunately, the estimation accuracy
Seraji and Colbaugh presented in [71] an indirect adaptive strategy: in order to track a desired reference force $f_r$, an appropriate reference position trajectory $x_r$ is generated online, based on estimates of the environment stiffness $\hat{k}_e$ and location $\hat{x}_e$. Also Popa and Singh investigated force-tracking impedance control, and suggested a model reference adaptive control (MRAC) strategy [72]. Both use the same Lyapunov-inspired adaptation law for the estimator, based on a purely spring-like environment model $f = k_e(x - x_e)$, measurements of the end effector position $x$, and the interaction force between the manipulator and the environment $f$. Both approaches yield stable interaction with environments with variable stiffness. Though the MRAC algorithm is much more complex and computational intensive, simulations showed no significant advantages over the indirect adaptive controller [73].

In [74], Love and Book estimated the location and dynamic characteristics of any constraints in a robot workspace using a MIMO recursive least square method. The estimates were then used in an adaptive impedance control, providing ameliorated contact stability and better performance of the position-based impedance controller. The solution of a weighted RLS can deal with time and state variations of the robot’s environment [75]. In [74] and [73], a constant weighting factor, slightly smaller than one was chosen. In [76], the weight was regarded as a 'forgetting factor', and depended on the speed of movement. This way, low speed motion yields slow forgetting, while high speed motion, inherently yielding more information about the environment dynamics, is rewarded with a higher update weight.

Several institutes implemented Kalman filters to perform independent online estimation of some environmental parameters [77, 78]. Within the dynamical equations, the parameters of interest, being mainly stiffness and also damping of the environment, appear multiplied with state variables of the system. The problem is thus ‘non-linear in its parameters’, and extended Kalman filters are used. Also for soft environments with highly non-linear characteristics, and in the presence of measurement noise, the implemented extended Kalman filter offered accurate and fast convergence to the right parameters [79].

All algorithms are capable of more or less accurate environment stiffness estimation [73, 77]. Persistent excitation is required to guarantee that the estimates converge to the real value. However, experiments show that the indirect adaptive control, the MRAC, the recursive least squares and the extended Kalman filter offer accurate stiffness estimation even without persistent excitation. On the other hand, reliable damping estimates really require it.

### 4.3 Recognition of contact states

The previous Sections deal with estimation of the geometric and dynamic parameters of one given contact state. This information can only be used effectively as soon as the type of the contact state itself is correctly identified. There are two different classes of contact state recognition schemes: those that can make their decisions based on a series of previous executions of the same compliant motion, in slightly varying conditions; and those that have no prior execution history, but only the contact models and
compliant motion specifications generated off-line.

If a large amount of experimental data are available, then it is common to circumvent the modeling problem altogether, by learning models based on such data. The data are obtained either in an unsupervised way or by human task demonstration. Contact state recognition based on learnt contact models can handle deviations of nominal parameter values for which learning data were available at training time.

Hannaford and Lee [80] modeled assembly tasks by a Hidden Markov Model of contact states ("sub-tasks"). Wrench measurement characteristics such as the mean and standard deviation for each contact state and the transition probabilities were learnt.

Takahashi et al. [81] and Wang et al. [82] used qualitative measurements for contact state recognition. The thresholds for clustering the measurements into “positive”, “zero” and “negative” values were determined experimentally. [81] used twist measurements, [82] used wrench and relative distance measurements.

Takahashi et al. [81] and Wang et al. [82] used qualitative measurements for contact state recognition. The thresholds for clustering the measurements into “positive”, “zero” and “negative” values were determined experimentally. [81] used twist measurements, [82] used wrench and relative distance measurements.

Cervera et al. [83] and Brignone and Howarth [84] recognized contact states by neural network structures based on wrench measurements. Nuttin et al. [85] additionally considered pose measurements.

Skubic et al. [86, 87, 88] compared the contact state recognition by fuzzy and neural network classifiers based on pose and wrench measurements.

Sikka, Hovland and McCarragher recognized 2D contact state transitions through wrench and change-in-wrench measurements. The recognition either was based on learnt discriminant functions [89], was rule-based using learnt measurement value thresholds [90], or was modeled by a trained Hidden Markov Model containing the time frequency model of the wrench measurements [91, 91, 92]. The contact state transition recognition was used in a Petri Net task model [93] to recognize contact states.

Analytical models eliminate the need for training by using an explicit physical model description. Uncertainty is represented by Probability Density Functions (PDFs) or uncertainty intervals.

Desai and Volz [94] recognized contact states based on pose and wrench measurements, considering uncertainty and friction. However, the contact model is based on uniform pressure distributions, an assumption rarely met (e.g., this means that a face-face contact cannot exert contact moments in the face).

Asada [95, 96] identified 2D contact states based on wrench measurements given that the contact consisted of a number of single vertex-edge contacts and that the relative pose between the contacting objects was known. The contact state classification was implemented as the first two layers of a three-layered neural network. The network was designed to learn the nonlinear mapping of wrench measurements to twist commands, i.e., to learn the compliance for force feedback.

McCarragher and Asada [97] and Schulteis et al. [98] recognized 2D contact states based on qualitative template matching of every measurement component. The templates were designed by the programmer and indicated whether the measurement values were “positive”, “zero” or “negative” for
each contact state. McCarragher and Asada used wrench and change-in-wrench measurements, and Schulteis et al. used wrench and twist measurements. Note that in contrast with the other papers, [97] modeled not only contact state recognition, but also contact state transition recognition this way. The transition recognition was used in a Petri Net task structure to recognize contact states [93].

Xiao [4] and Xiao and Zhang [99] recognized polyhedral contact states based on pose measurements by growing the objects with the pose uncertainty (defined as an interval) and by intersecting the grown regions. The method returns the set of possible valid principal contacts given the pose uncertainty. Xiao and Zhang [25] further reduced the set to the geometrically valid contact states. Xiao and Liu [100] added wrench measurement information and interpreted the contact state recognition as the calculation of “fuzzy” membership values.

Hirai and Iwata [101], Hirai and Asada [18] and Hirukawa et al. [102] modeled polyhedral contact states with polyhedral convex cones indicating the ranges of possible wrenches and displacements (twists) measurable in a contact state. The contact state can be determined by applying linear discriminant functions (representing the face vectors of the cones) to the measurements. Farahat et al. [103] and Mosemann et al. [104] extended this work to include friction. Farahat et al. additionally considered pose uncertainty by testing all of the discrete set of poses; the contact states considered for testing at each pose was given by [4].

Suárez et al. [105, 106] described 2D contact state recognition by verifying if the pose and wrench measurements lied in predefined regions. These regions were calculated offline for every contact state based on the configuration space description of the system given the uncertainties.

Spreng [107, 108] recognized polyhedral contact states based on the distance (with its uncertainty) between the objects, the history of the motion (i.e., its instantaneous approach direction), the compatibility of a hypothesized contact (one constraint lost or gained at each transition), and the distance (with its uncertainty) between the geometrical contacting elements of the objects. Discriminating between feasible contact states was based on movability tests: each test consisted of a small twist which is feasible in case of the presence of one subset of the hypotheses but infeasible in case of its complement.

If the analytical contact models are expressed as a function of unknown parameters, the contact state recognition is performed simultaneously with the parameter estimation. During the task execution, the uncertainty reduces due to the parameter estimation. This improves the force control, the contact state transition monitoring and the contact state recognition. The increased performance, however, comes with a higher computational cost.

Kitagaki et al. [109] estimated the location of the contact point based on wrench increments for a single vertex-face contact. A transition between the vertex-face and an edge-face contact is observed by monitoring the absolute difference between the expected and the observed moment. In [110] the estimation of a pseudo contact point is added. This pseudo contact point is the intersection between the force screw axis and the perpendicular to the force screw axis through the contact point. Large changes in pseudo contact point indicate a contact transition.
Mimura and Funahashi [59] recognized vertex-face, edge-face and face-face contact states based on wrench and pose measurements. The different contact state models are tested from the least to the most constrained until a model is found which is consistent with the data, i.e., a model for which the geometrical parameters can be determined.

Based on pose and wrench measurements, Eberman [62] calculated a maximum likelihood estimate of the number of constraints and the rotation of the constrained/free space for systems for which the configuration space model was known. Contact transitions are detected by a statistical consistency test (SNIS, Summed Normalized Innovation Squared). Possible next contact states are described by a contact state network. Contact recognition is based on the likelihood of the possible next contact states.

Debus et al. [67], De Geeter et al. [111], De Schutter et al. [65], Lefebvre et al. [112] and Gadeyne et al. [113] focused on general contact state recognition under uncertainty of the poses and dimensions of the contacting objects. Slaets et al. [114] also assumed topological uncertainty of the contacting objects (for polyhedral contacting parts). In [67], for every probable contact state a deterministic estimator is run on the pose measurements. The penetration distances of the objects from the different estimators are used as measurement in a Hidden Markov Model. In [111, 65, 112, 114], for each probable contact state a separate statistical estimator is run. A SNIS test indicates the estimator which is consistent with the measurement data (and hence the valid contact state and geometrical parameter estimates). The geometrical parameter estimates are those returned by this estimator. Gadeyne et al. [113] described the simultaneous estimation of the contact state and the geometrical parameters by a single Bayesian estimator.

5 Control

A controller receives set-points from the planner component, compares them to the measurements from the robot system, and calculates appropriate robot joint actuator commands (velocities, torques, pressures, etc.).

The fact that autonomous compliant motion systems still have very little “intelligence” is mainly due to the progress that still has to be made in the integration of the planning, estimation, modeling and control components. Control in itself, however, has since long reached the required maturity and “expressiveness”. This Section briefly introduces the approaches that have proven to be appropriate for compliant motion control: hybrid force/position control (Section 5.1), parallel force control (Section 5.2), and impedance control (Section 5.3). Section 6 discusses some issues about how to integrate the control approaches with planning and estimation.

5.1 Hybrid force/position control

This control approach was developed very closely together with the Task Frame or Compliance Frame modeling and specification approach – see Section 2.2 and [115, 116, 8]:

13
• The six “directions” of the Task Frame are divided into motion-controlled directions and force-controlled directions. Force-controlled directions are transformed into motion-controlled directions, by means of a feedback action that has the physical meaning of an “admittance” (inverse impedance): force errors are transformed into desired motions that will reduce the force errors. In fact, force control corresponds to the deformation control of the admittance (compliance and inverse damper).

• The result of the above-mentioned Task Frame-based force and motion control is an instantaneous twist setpoint, i.e., a six-dimensional translational and angular velocity for the Task Frame. It is straightforward to transform this twist to the robot end-effector, where it is given as input to the robot controller (assuming that the latter accepts velocity set-points, which is the case with most industrial robots).

Hybrid force/motion control has received very little attention from the research community since about a decade. The reason is that this control approach is quite mature, and no really significant new issues have arisen during the last few years. Nevertheless, industrial acceptance of hybrid force/motion control is very low, for the simple reason that most industrial robot controllers do not provide it as a standard feature. In addition, the feature is not a high priority for the industry, because the industry solves all of its current compliant motion problems by (i) highly structuring the work environment of a robot, and (ii) adding some passive compliance between the robot and the environment.

(Theses comments about the slow take-up of control techniques in industrial robots also hold for the approaches below.)

5.2 Parallel force control

This control approach does not necessarily require “decoupled” force-controlled and motion-controlled directions [117, 118]: the task can specify force and motion setpoints in the same directions of the Task Frame. Internally in the controller, the force set-points are transformed into motion set-points, as in the hybrid approach, and the two set-points in the same direction are weighted by their respective feedback control constants. “Safe” motion is obtained by giving more weight to force control. The integration of the Task Frame with the parallel force control scheme is of the same, low complexity as the integration with the hybrid control approach.

5.3 Impedance control

This control approach [119] does not regulate the contact forces directly, unlike the hybrid and the parallel approaches. Impedance control specifies two complementary things:

• Motion: the planner specifies the task completely as a desired motion in the six motion degrees-of-freedom of a robot. But the specified motion is such that it brings the robot in contact with the environment.
• Desired impedance, i.e., the desired dynamic relationship (“mass-spring-damper” behavior) between the contact force and the resulting motion of the robot.

Therefore, the actually occurring force is a result of how far the specified motion makes the manipulated object “penetrate” the environment. Impedance control is a bit more difficult to integrate with the planning approaches of Section 3, and is hence used less than the hybrid and parallel approaches.

6 Integration

While there has been considerable research in different components required of an active compliant motion system, as surveyed in the previous sections, there is a lack of effort in integrating or interfacing those components. One major problem is that the control approaches discussed above do not directly interface with the higher-level models used by planning and estimation. Whereas, being able to derive their actions based on a rich set of high-level modeling primitives is desired for more “intelligent” planners and estimators (recall the cup-on-the-table example, Section sect:modeling). The missing links are: (i) translating the output of planners into “low-level” specifications that are compatible with the control approaches so that the planned compliant motion can be automatically executed; and (ii) extending the estimator component with functionalities to transform the low-level measurement interpretations into the higher level models used by the planner. The latter link, in particular, still requires most work on the estimators, and it is also a crucial link to provide valuable feedback to the planner for planning to be more effective.

Indeed, many current planners work only in one way and off-line: they generate plans. In the two-phase approach of fine motion planning (see Section sect:planning), a replanner is supposed to work in two ways and on-line, i.e., it receives feedback from the estimator and controller about the actually reached contact state or sequence of contact states and is able to recover from “errors”: as soon as deviations occur between the executed sequence and the planned sequence, the controller and the estimator cannot proceed because the compliant motion they received from the planner is not valid anymore; the planner then re-plans the task, starting from the currently reached contact state. However, there hardly exists implemented prototype of such a two-way and high-level planner.

The most intelligent planning system should take into account the capabilities and limitations of both the controller and the estimator, and it generates active sensing actions in case the knowledge about the system’s state is insufficient to execute the task reliably. Active sensing actions are motions that do not necessarily bring the manipulated object or the robot closer to its goal configuration but generates sensing information that the estimator needs in order to reduce the uncertainty about the world.

The estimator component compares the expected sensor measurements obtained from the planner and the modeling component to the ones generated by the real execution. Based on this information, it notifies the controller (and ideally the planner, when that planner can receive on-line feedback) when
the end-conditions of each subtask are reached. Since each subtask may correspond to a motion in a different contact state, monitoring the end-conditions means monitoring contact state transitions.

Many existing estimators only detect transitions in a contact state and raise a signal such that the controller can switch to the next planned compliant motion task. Somewhat more involved estimators are capable of detecting execution errors, when the detected contact state transition results in a different contact state than the one that was expected in the compliant motion plan.

The most sophisticated estimators additionally reduce the uncertainty in the models, by estimating the geometrical, and possibly also the dynamical, contact parameters, such as the contact state (i.e., the exact number and location of the contacts between manipulated and environment objects) and the impedance of the interaction at the contact(s). Reducing the uncertainty is advantageous because it improves the quality of the task execution (speed of execution, robustness against uncertainty, reduction in contact forces, etc.). Even more important is for such an estimator to convey the estimated information, especially high-level state and model information, to the planner to enable the planner to perform on-line task adaptation. It thus can effectively enlarge the class of tasks that can be executed autonomously.

7 Conclusions

Compliant motion is a necessary component of an “intelligent” robot, as far as intelligence includes the ability to reliably, flexibly, and efficiently achieve desired interaction between the robot and its environment with uncertainty. An autonomous compliant motion system must have at least the following three active components: an interaction task planner capable of on-line replanning; an on-line estimator to reduce the uncertainties between the planned interaction and the real world and to act as a bridge between the planner and the controller; and a controller to transform a plan into real actions of the robot in its environment. These three components share a lot of common modeling information, and the information is, in addition, continuously updated by the various components.

The research on compliant motion was born with flexible assembly as the major application domain. However, industrial assembly plants invariably use only passive compliance and highly structured “environments” because of the difficult challenges of an active compliant motion system. However, the state-of-the-art in the field and the ongoing research efforts surveyed in this paper allow optimism for the future: most scientific hurdles have been taken so that at least a somewhat autonomous compliant motion system can be built. In practice, major efforts are still needed to integrate planning, estimation, control and modeling. These efforts have proven to be too much for the small research community of compliant motion researchers. But the recent progress in “humanoid” robotics will most probably increase the compliant motion community in the very near future: the humanoid robots’ state-of-the-art is such that research labs begin to use them to perform compliant motion tasks, and hopefully the large manpower available in these labs will bring the necessary mass for critical integration to the compliant motion research community.
Note that the previous paragraph describes the optimistic expectation only! As described in Section 6, the transformation from raw measurement data into the high-level contact primitives required by intelligent planners is one of the major gaps in the current state-of-the-art. The other major gap is the robustness in contact state transition detection and identification. If human behaviors can be taken as a source of inspiration about how to close these gaps, one solution would probably come from active sensing: whenever uncertainty arises about the exact type of contact state, the (re)planner should stop the ongoing task, and schedule compliant motions that “excite” missing contact parameters. This active sensing is a shared responsibility of the planner and the estimator.

Acknowledgments

T. Lefebvre is a Postdoctoral Fellow of the Fund for Scientific Research–Flanders (F.W.O.) in Belgium. Financial support by K.U.Leuven’s Concerted Research Action GOA/99/04 is gratefully acknowledged. Fundings provided by the U.S. National Science foundation under grants IIS-9700412 and IIS-0328782 are also greatly appreciated.

REFERENCES


