Robust, Compliant Assembly with Elastic Parts and Model Uncertainty

Florian Wirnshofer[1], Philipp S. Schmitt[1], Philine Meister[1], Georg v. Wichert[1], and Wolfram Burgard[2]

***Abstract*— In this paper, we present an approach to generate robot motions for robust parts assembly. The computation of motions for parts assembly usually requires an exact model of all relevant objects. Generating detailed object models, including friction and dynamics, is often complex and time-consuming, especially in the context of elastic parts. In addition, executing motions on real hardware will usually introduce further uncertainty. For this reason, we propose an approach that is inherently robust against model parameter uncertainties and unknown characteristics of elastic parts. Our planner explicitly takes into account the internal states of articulated objects, as well as uncertain model parameters, by constructing a search tree in the belief-parameter-space. It yields successful assembly motions from coarse object models and thus eliminates the need for detailed parameter tuning. We evaluated our approach with respect to four assembly tasks. Extensive simulations show that our planner significantly increases the success-rate compared to previous approaches. Numerous experiments on a real robot confirm the simulated results.**

I. INTRODUCTION

Many automation tasks in manufacturing and logistics require robots to reliably join parts of various geometries and material characteristics. Examples include packaging or assembly tasks such as the elastic peg-in-hole task in Fig. 1.

The state-of-the-art approach to automating these tasks is to hold parts in precise fixtures or part-feeders and to explicitly describe all robot motions via teach-in programming. In applications where the task or the parts change frequently, teach-in programming is not viable. We aim for a method that autonomously computes robot motions for highly reliable parts assembly. A number of aspects render the automation of motion planning for parts assembly a challenging problem.

1) **High-dimensional spaces with non-linear dynamics:** Assembly is a dynamic process which requires the inclusion of velocities into the configuration-space. For a six-axis robot this already results in a 12-dimensional configuration-space. As contact between parts is essential to assembly, the dynamics exhibit abrupt non-linearities.
2) **Uncertainty in the process:** When precise part-feeders or fixtures are not available, robotic assembly must rely on inherently noisy sensor-readings to determine the poses of objects. For elastic parts, such as a wire, the dynamic evolution of the part's deformation must be considered as well.
3) **Inaccuracy of models:** Being able to successfully deploy planned motions on real hardware requires a precise model of all parts, robots, and the execution environment. While the geometry of parts is typically available via CAD-data, material properties such as friction or elasticity are difficult to obtain. They may even change during a process step, e. g., when parts are exposed to heat or lubrication during machining. A precise calibration of models is impractical, especially in the context of small batch production.

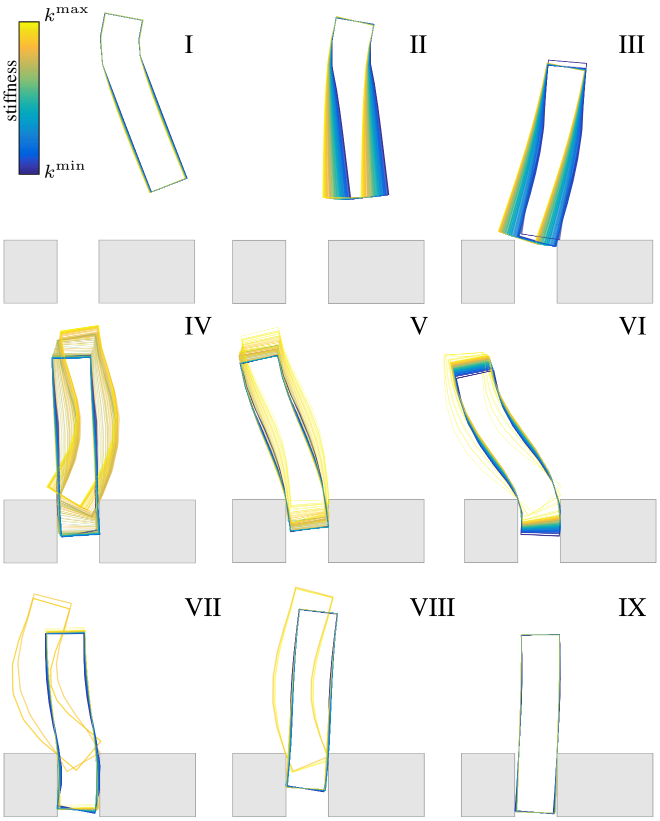

Fig. 1. Elastic peg-in-hole assembly: This problem is particularly challenging since precise estimates of material characteristics such as bending-stiffness are typically unavailable. The presented method computes assembly plans that are robust towards uncertainties in the physical model parameters.

The main contribution of this work is a kinodynamic belief-space planner for assembly, that addresses all challenges above. This planner extends our previous work [1] on robust assembly to elastic parts and noisy estimates of model parameters, such as friction. A further contribution of this work is an analysis of the influence friction has on the robustness of planned parts assembly. We conduct our analysis with regard to extensive simulated and real world experiments. These experiments show that the consideration of noisy model parameters significantly increases the robustness of robotic assembly. The proposed planner yields robust motion plans from CAD data as only input, thus circumventing the need for extensive parameter tuning.

---

[1] Siemens Corporate Technology, Otto-Hahn-Ring 6, Munich, Germany
[2] Dep. of Computer Science, University of Freiburg, Freiburg, Germany

II. RELATED WORK

The presented work relates to three major strands of research: motion generation for compliant assembly, kinodynamic motion planning, and planning under state and environmental uncertainty.

A. Compliant Assembly

There exists a broad scope of methods for computing manipulator motions for parts assembly. Among the early approaches is fine motion planning (Lozano-Pérez *et al.* [2]). It has however been shown that fine motion planning does not scale to problems of practical relevance [3]. The works of Dakin *et al.* [4] and Ji *et al.* [5] address scalability by means of hierarchization and planning over contact-manifolds, yet require a manual classification of all possible contact states. Bruyninckx *et al.* [6] employ analytical models for each phase during a peg-in-hole scenario. A formal derivation of motion constraints, though highly effective for the task at hand, is impractical considering the enormous variety of possible part geometries.

B. Sampling-Based Motion Planning

The autonomous generation of robot motion requires reasoning in high-dimensional robot configuration-spaces, for which sampling-based planners have demonstrated remarkable successes. The Probabilistic Roadmap (PRM, Kavraki *et al.* [7]) or the Rapidly-exploring Random Tree (RRT, Lavalle *et al.* [8]) are prominent examples for sampling-based motion planners. The results as obtained from kinematic planners require post-processing prior to being executed on a real robot. In order to account for more general, control-based system dynamics, Lavalle *et al.* proposed a kinodynamic extension to RRT [9]. Both, PRMs and RRTs rely on local steering-functions. Steering functions are difficult to obtain, especially in the context of highly non-linear dynamics as encountered during parts assembly. The kinodynamic Expansive Space Tree (EST) planner of Hsu *et al.* [10] grows a tree by applying random actions to already existing tree nodes. This circumvents the need for intricate steering functions. The planner presented within this work builds on the EST framework.

The above methods assume perfect state- and model-information. One contribution of this work is an elaborate analysis of the influence of imprecisely calibrated model parameters on the success rates during real-world execution. Our results emphasize the need for a systematic treatment of uncertainty when planning trajectories for parts assembly.

C. Planning under Uncertainty

In order to address uncertainties in environment and initial state, Melchior *et al.* [11] introduce Particle RRT. Similar to our planner, Particle RRT employs a particle-based uncertainty representation. However, Particle RRT requires heuristics for the assessment of path quality and, more importantly, to cluster particles into belief-nodes. Sieverling *et al.* [12] and Páll *et al.* [13] adopt a particle-based belief representation to domains, where robots enter deliberate contact with the environment. However, Contact-Exploiting RRT (CERRT) and Contingent CERRT constrain the entire particle set to be either in contact or in free space. We found that this constraint reduces the solution space especially in the context of manipulating elastic objects.

The work of van den Berg *et al.* [14] introduces LQG-Motion Planning. The LQG-MP framework risk-assesses candidate motion plans, taking into account a-priori knowledge of sensor- and motion-noise. Patil *et al.* [15] make use of the LQG-MP framework, addressing the problem of motion planning for uncertain, deformable environments. Similarly, the Chance Constrained RRT (CC-RRT) of Luders *et al.* [16] computes motion plans that take into account systems with process and environment uncertainties. Under the assumption of linear system dynamics, CC-RRT and an asymptotically optimal variant CC-RRT* [17] provide probabilistic guarantees on feasibility and asymptotic optimality, respectively. The above methods are tailored to either linear systems or to systems that are well approximated by means of local linearization. The highly non-linear and abrupt discontinuities encountered in contact-dynamics renders methods based on local models infeasible. Agha-Mohammadi *et al.* [18] generalize the PRM framework to domains with motion and sensing uncertainties. However, their Feedback-based Information RoadMap (FIRM) relies on the availability of local steering functions, which in the context of contact dynamics are difficult to obtain.

We extend our previous belief-space planner *Belief*-EST [1] (B-EST) to assembly domains containing elastic parts and model uncertainty. The proposed extensions do not rely on the availability of local controllers, guiding heuristics, or tailored reductions to lower-dimensional manifolds. We evaluate our method with regard to several assembly benchmarks, considering the full state-space of robot and objects.

III. PROBLEM SETTING AND NOTATION

In the following, we state the system class considered and our notation (Section III-A). The need for compliant robot-environment interaction is discussed in Section III-B. The system's dynamical behavior is strongly influenced by the underlying physical quantities. We present a parametric version of the system dynamics and the space of model parameters in Section III-C. The system state and the model parameters are subject to uncertainty. Section III-E therefore introduces the notion of a belief over state and model parameters.

A. Contact Dynamics System Model

We consider the problem of parts assembly, where a robot manipulator joins two objects, one rigidly attached to its end-effector, the other, statically resting on the environment geometry. Fig. 2 depicts an exemplary assembly task and clarifies upon relevant quantities. Throughout the remainder of this work, we use subscripts $(\cdot)_\mathrm{r}$ and $(\cdot)_\mathrm{o}$ in order to address quantities that refer either to the robot or to the grasped object, respectively. The robot manipulator has n_r

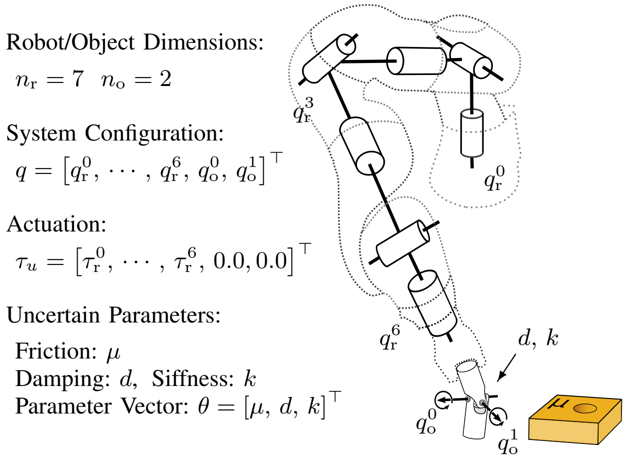
Robot/Object Dimensions:
$n_\mathrm{r}=7 \quad n_\mathrm{o}=2$
System Configuration:
$q=\left[q_\mathrm{r}^0,\,\cdots,\,q_\mathrm{r}^6,\,q_\mathrm{o}^0,\,q_\mathrm{o}^1\right]^\top$
Actuation:
$\tau_u=\left[\tau_\mathrm{r}^0,\,\cdots,\,\tau_\mathrm{r}^6,\,0.0,0.0\right]^\top$
Uncertain Parameters:
Friction: μ
Damping: d, Siffness: k
Parameter Vector: $\theta=[\mu,\,d,\,k]^\top$
q_r^3
q_r^0
q_r^6
$d,\,k$
q_o^0
q_o^1
μ

Fig. 2. Exemplary assembly scenario: a seven degrees-of-freedom (DoF) robot arm joining a two DoF hinge with an environment-mounted object (yellow). Several physical parameters affect the system dynamics. These include the damping- and stiffness-parameters of the hinge joints and the frictional coefficients of the involved materials.

fully actuated degrees-of-freedom. We assume the grasped object to be either rigid ($n_\mathrm{o}=0$), or articulated by means of n_o non-actuated degrees-of-freedom. The simple 2-DoF hinge depicted in Fig. 2 serves as an example for such an articulated object. Further examples include flexible objects such as wires, ropes, or cloth, which are commonly approximated by means of several rigid bodies interconnected with prismatic or revolute joint primitives.

A system configuration $q=\left[q_\mathrm{r},\,q_\mathrm{o}\right]^\top\in\mathbb{R}^{n_q}$, where $n_q=n_\mathrm{r}+n_\mathrm{o}$, is a composite vector of all robot configurations $q_\mathrm{r}\in\mathbb{R}^{n_\mathrm{r}}$ and object configurations $q_\mathrm{o}\in\mathbb{R}^{n_\mathrm{o}}$. With the system configuration in place, we may define the system dynamics model as

$$M\left(q\right)\ddot{q}+c\left(q,\,\dot{q}\right)+g\left(q\right)+h\left(q,\,\dot{q}\right)=\tau_u+\tau_c, \qquad (1)$$

where $M\in\mathbb{R}^{n_q\times n_q}$ is the joint-space inertia matrix. The composite vectors $g=\left[g_\mathrm{r},\,g_\mathrm{o}\right]^\top\in\mathbb{R}^{n_q}$ and $c=\left[c_\mathrm{r},\,c_\mathrm{o}\right]^\top\in\mathbb{R}^{n_q}$ denote the gravitational and Coriolis force components. We denote the input to the dynamical system $\tau_u=\left[\tau_\mathrm{r},\,\underline{0}^{n_\mathrm{o}}\right]^\top$. Since we assume a non-actuated object, the actuation torque τ_u is composed solely of the robot axis-torques $\tau_\mathrm{r}\in\mathbb{R}^{n_\mathrm{r}}$. Constraint forces $\tau_c\in\mathbb{R}^{n_q}$ occur whenever the robot or the grasped object is in contact with the environment. The generalized force $h\left(q,\,\dot{q}\right)\in\mathbb{R}^{n_q}$ represents the forces attributed to the dynamics of non-rigid objects, e.g., a retraction force of a compressed elastic object.

B. Compliant Manipulation

During assembly, we deliberately exploit motions where the grasped object enters contact with the environment geometry. When using stiff, position-controlled robot manipulators, interactions between the manipulated object and the environment geometry may cause contact forces that damage the robot and its environment. Hence, compliant robot-environment interaction is indispensable. We therefore make use of a stable, compliant robot controller

$$\tau_\mathrm{r}=C\left(q_\mathrm{r},\,\dot{q}_\mathrm{r},\,q_\mathrm{r}^\mathrm{d},\,\dot{q}_\mathrm{r}^\mathrm{d},\,\ddot{q}_\mathrm{r}^\mathrm{d}\right). \qquad (2)$$

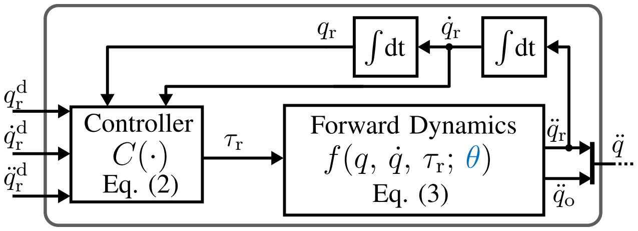

Fig. 3. Integrated model parameter-dependent system dynamics with contact and a feedback-controlled compliant robot manipulator.

The compliant controller exerts an actuation torque τ_r based on the deviation of the current robot state $\left(q_\mathrm{r},\,\dot{q}_\mathrm{r}\right)$ from an acceleration-limited reference $\left(q_\mathrm{r}^\mathrm{d},\,\dot{q}_\mathrm{r}^\mathrm{d},\,\ddot{q}_\mathrm{r}^\mathrm{d}\right)$. Concrete realizations of a compliant controller include, among others, Cartesian-space impedance control and joint-space impedance control.

C. Physical Model Parameters

Both the contact forces and the dynamical behavior of the grasped objects depend upon the parametrization of the underlying physical models. Practical examples for such parameters include surface friction coefficients, kinematic offsets, or the stiffness- and damping-rates of elastic parts.

Let therefore $\theta=\left[\theta_1,\,\ldots\theta_i\ldots,\,\theta_{n_\theta}\right]\in\Theta$ be a n_θ-dimensional vector of model parameters. Given θ, we may define the parametric version of the forward-dynamics corresponding to (1) as

$$\ddot{q}=f\left(q,\,\dot{q},\,\tau_\mathrm{r};\,\theta\right). \qquad (3)$$

As pointed out earlier, we cannot assume precise knowledge on the above stated parameters. Hence, we restrict our assumption on individual parameter values θ_i to a bounded interval $\theta_i\in\left[\theta_i^\mathrm{min},\,\theta_i^\mathrm{max}\right]$. For any choice of model parameters, the resulting parameter-space may be written as the Cartesian product over the parameter intervals, i. e.,$\Theta=\prod_{i=1}^{n_\theta}\left[\theta_i^\mathrm{min},\,\theta_i^\mathrm{max}\right]$, where $(\cdot)^\mathrm{min}$ and $(\cdot)^\mathrm{max}$ denote the respective lower and upper parameter bounds. Fig. 3 depicts the feedback-loop for our parameter-dependent system.

D. Compound State

Since the dynamics $f\left(\cdot\right)$ are a function of both state and the respective model parameters, we introduce the *compound state*

$$x=\left(q,\,\dot{q};\,\theta\right)\in\mathcal{X}, \qquad (4)$$

with $\mathcal{X}=\mathbb{R}^{2n_q}\times\Theta$ being the combined space of system configuration, velocities and model parameters.

E. Planning in Belief-Space

Uncertainties in system state and the model parameters require a probabilistic view on the evolution of the system dynamics. Let therefore $x_t=\left(q_t,\,\dot{q}_t;\,\theta\right)$ be the time discrete equivalent to the compound state introduced in (4). The compound state x_t evolves according to the state-transition probability

$$p\left(x_{t'}\mid x_t,\,u_{t\rightarrow t'}\right), \qquad (5)$$

where $t' > t$. The system input $u_{t\to t'}$ encodes the evolution of our reference trajectory $\left(q_{\mathrm{r}}^{\mathrm{d}},\, \dot{q}_{\mathrm{r}}^{\mathrm{d}},\, \ddot{q}_{\mathrm{r}}^{\mathrm{d}}\right)$ between t and t'.

We represent current knowledge about the compound state by means of a probability distribution over compound states $b_t = \mathrm{bel}\left(x_t\right) \in \mathcal{B}$. The associated space of belief distributions is given as $\mathcal{B} = \left\{\mathrm{bel}\left(x_t\right) \mid \mathrm{bel}\left(x_t\right) : \mathcal{X} \to [0, \infty)\right\}$. Given the state-transition probability in (5) and the current belief $\mathrm{bel}\left(x_t\right)$, the belief evolves as

$$\mathrm{bel}\left(x_{t'}\right) = \int_{\mathcal{X}} \mathrm{bel}\left(x_t\right)\, p\left(x_{t'} \mid x_t,\, u_{t\to t'}\right) \mathrm{d}x_t. \qquad (6)$$

Note that we do not consider sensory feedback aside from the state-feedback of the compliant controller (2).

With the notion of a belief and the corresponding belief-propagation in place, we can now formally define our objective. Let $\mathcal{F} \subset \mathcal{B}$ be the region of *valid* belief states. The belief-space region $\mathcal{F}$ is a measure of probability mass. To give an example, one could define $\mathcal{F}$ as the region, for which 99% of the belief experiences external torques less than 42 Nm. Furthermore, consider $b_0 = \mathrm{bel}\left(x_{t=0}\right)$ a given initial distribution over compound states, which encodes the prior knowledge about the system state and the model parameters.

Definition 1. A belief trajectory $\mathit{\Gamma} : \{b_0, b_1, \ldots, b_T\}$ with $T \geq 0$, generated by a sequence of inputs $\Pi : \{u_{0\to i}, \ldots, u_{j\to T}\}$ with $0 \leq i \leq j \leq T$ is *valid* iff it lies within $\mathcal{F}$ and adheres to the belief dynamics (6).

Another input to the planner is the non-empty goal region $\mathcal{G} \in \mathcal{B}$. An example for a goal $\mathcal{G}$ could be the region in belief-space, where more than 95 % of the probability mass are in immediate vicinity of a target configuration. Given the goal region $\mathcal{G}$, the initial distribution b_0, and the valid region $\mathcal{F}$, we can define a *feasible* trajectory.

Definition 2. A belief trajectory $\mathit{\Gamma}$ is *feasible* iff it starts in b_0, ends in a goal region $b_T \in \mathcal{G}$, and is valid according to Definition 1.

The overall objective is to find a feasible trajectory, solving the task of parts assembly in the presence of parametric uncertainties.

IV. ROBUST, COMPLIANT MOTION PLANNING

In the following section, we introduce *Model-Belief*-EST (MB-EST), an extension of the B-EST planner that addresses uncertain model parameters and articulated objects. A detailed outline of the algorithm for particle-based belief representations is given in Section IV-B.

A. Model-Belief-EST: Algorithm Outline

The EST [10], the B-EST [1], and the herein presented MB-EST are kinodynamic motion planners that iteratively expand a search tree. Any such expansive space tree planner has to implement the following four methods: a method for selecting a candidate-node for tree expansion, a control-input sampler, a method for integrating dynamics, and finally a method that checks, whether or not a node is valid.

```
algorithm MB-EST($b_0$, $\mathcal{G}$)
1:  $V$ = $\{b_0\}$ , $E$ = $\{\}$
2:  while withinTimeBudget() do
3:      $b_t$ = sampleWeighted($V$)
4:      $u_{t\to t'}$ = sampleControl()
5:      $b_{t'}$ = integrate($b_t$, $u_{t\to t'}$)
6:      if isValid($b_t$, $u_{t\to t'}$) then
7:          $V$.append($b_{t'}$)
8:          $E$.append($b_t$, $u_{t\to t'}$, $b_{t'}$)
9:          if $b_{t'} \in \mathcal{G}$ then
10:             return $\Pi$, the control actions from $b_0$ to $b_{t'}$
11: return failure
```

Beyond concrete realizations for these methods and the search-space domain, the high-level algorithms share a common structure. Compared to the EST, which addresses classical, deterministic motion planning, both B-EST and MB-EST account for probabilistic planning domains. B-EST constructs a search tree in the space of probability distributions over system states. MB-EST further incorporates the internal states of articulated objects as well as uncertain model parameters, by constructing a search tree in *belief-parameter-space*. In the following we briefly outline the general procedure of the MB-EST shown above.

We initialize the set of nodes V with the root node b_0. The set of edges E is initially empty (Line 1). The tree expansion is driven by the following steps: first the **sampleWeighted** method selects a random node b_t from the current tree (Line 3). Next, the **sampleControl** method returns a random control action $u_{t\to t'}$ (Line 4). Given the current state and a random control action, the algorithm according to (6), **integrate**s the dynamics (Line 5). If the trajectory segment from b_t to $b_{t'}$ is valid (Line 6), the child node $b_{t'}$ is added to the tree (Lines 7 & 8). In case a child node $b_{t'}$ is contained within the goal-region $\mathcal{G}$, the algorithm terminates and returns the sequence of control actions Π.

B. MB-EST for a Particle-Based Belief Representation

There are several approaches to representing $b_t = \mathrm{bel}\left(x_t\right)$, the probability density over the compound states. These include parametric distributions (e.g. Gaussian or mixture of Gaussian distributions) or non-parametric approximations such as histograms or particles. Contact dynamics are highly non-linear and aside from a few special cases not available as closed-form analytical models. Hence, we favor a particle-based belief representation. We approximate the belief by means of a set of N particles $\left({}^{[1]}x_t,\, \ldots,\, {}^{[N]}x_t\right)$. A particle ${}^{[i]}x_t = \left({}^{[i]}q_t,\, {}^{[i]}\dot{q}_t;\, {}^{[i]}\theta\right) \sim b_t$ denotes a sample drawn from the true belief b_t. A notable difference to the B-EST algorithm is that a state x now comprises the state of the robot manipulator, the internal states of articulated or elastic objects, and physical model parameters.

We obtain the initial particle set representing b_0 for MB-EST as follows: since we assume the initial system state to be measurable, we share $\left(q_0,\, \dot{q}_0\right)$ across all particles i. e. $\left({}^{[i]}q_t,\, {}^{[i]}\dot{q}_t\right) = \left(q_0,\, \dot{q}_0\right)\, \forall\, i{=}1 \ldots N$. Each individual parameter vector ${}^{[i]}\theta$ is drawn at uniform from the bounded parameter space Θ.

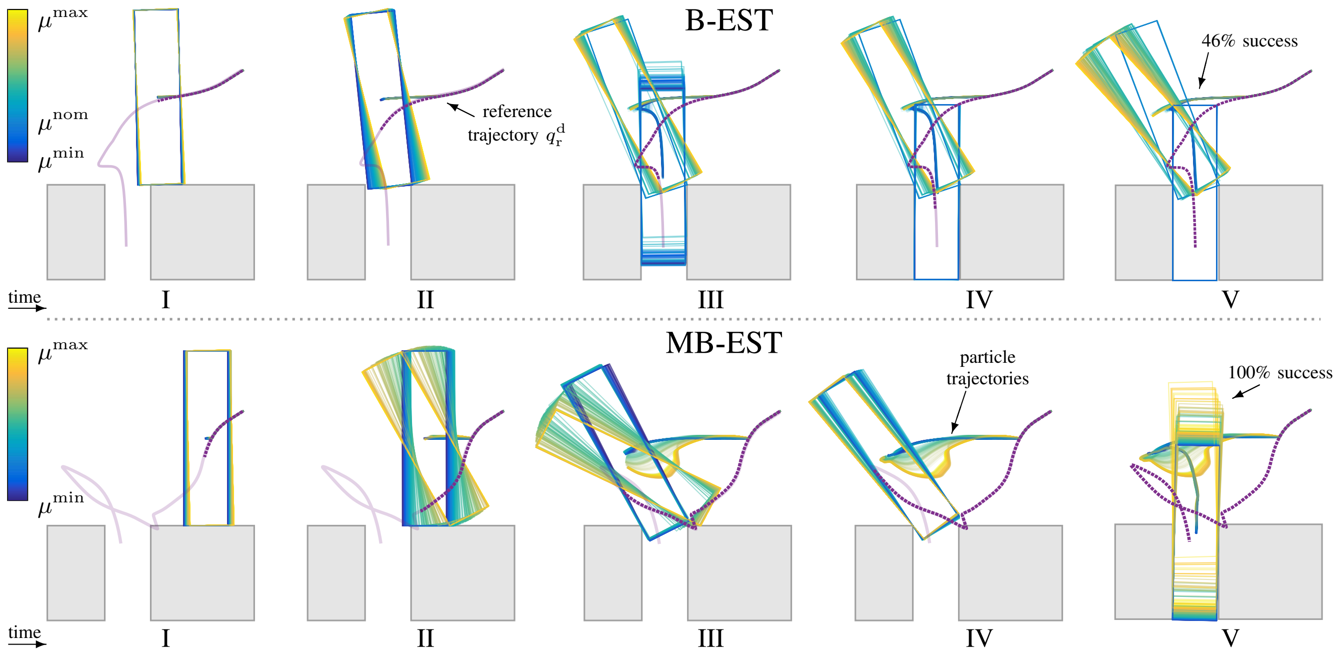

Fig. 4. Executions of trajectories for compliant peg-in-hole insertion for different material frictions. For the B-EST method, the trajectory yields a successful peg insertion only for cases where the friction parameter during execution closely matches the nominal friction coefficient μ^{nom} assumed during planning. Contrary, the MB-EST reference trajectory achieves successful peg insertion independent of the actual friction coefficient. MB-EST is able to decouple the execution success-rate from the gap between simulation and reality.

The **sampleWeighted** method biases the tree expansion towards sparsely covered regions in belief-parameter-space. We assign a weight to each node. The weight is inversely proportional to a density estimate as to which the respective region is already covered. This weighted sampling policy favors to expand nodes in sparsely covered regions. In more tangible terms, we place a coarse grid over the space of the particles' mean object pose. During tree construction, each node is assigned to one of these grid cells. The sampling policy then returns a random node from a random, occupied grid cell.

A sampled belief node evolves by means of a randomly chosen control action $u_{t \to t'}$. In the context of our parts-assembly scenarios, a random action $u_{t \to t'}$ encodes a reference trajectory segment $\left(q_{\mathrm{r}}^{\mathrm{d}},\, \dot{q}_{\mathrm{r}}^{\mathrm{d}},\, \ddot{q}_{\mathrm{r}}^{\mathrm{d}}\right)$. The length of the trajectory segment is typically sampled at uniform from a bounded interval.

The evaluation of the dynamics (6) by the **integrate** method is straightforward for particle-based belief representations. For each particle, we simulate the system response of the closed-loop model depicted in Fig. 3 to the previously sampled reference trajectory. The result of this procedure is a set of trajectory segments describing the evolution of the individual particles. The end-points of these trajectory segments is what constitutes the node $b_{t'}$. Refer to Fig. 4 for the particle and reference trajectories during a peg-in-hole scenario.

The **isValid** method checks whether a belief-space trajectory segment is vaild according to Definition 1. In the context of parts assembly, it is crucial to prevent situations in which the robot or its environment are at risk of being damaged. Hence, we enforce an upper bound on the robot torques. If we register a violation of the maximum allowed torque for any of the particles, we reject the node candidate.

We consider a particle-based belief approximation to be contained in the goal region if more than a γ fraction of the particles fulfill a certain goal condition $\operatorname{dist}(q) < d^*$. With regard to the assembly scenario in Fig. 4, we define $\operatorname{dist}(q)$ as the distance between peg and hole and d^* as a suitable distance threshold. As soon as a particle set fulfills the goal condition, the tree expansion terminates. The algorithm returns a sequence of trajectory points $\left(q_{\mathrm{r}}^{\mathrm{d}},\, \dot{q}_{\mathrm{r}}^{\mathrm{d}},\, \ddot{q}_{\mathrm{r}}^{\mathrm{d}}\right)$ corresponding to the actions Π that steer the system from b_0 to the goal node.

V. MODEL PARAMETERS FOR PARTS ASSEMBLY

The following section introduces the set of physical model parameters we consider most relevant for industrial assembly tasks.

A. Kinematic Offsets

Imprecise grasping or slippage of the grasped object induce uncertainty on the kinematic offset between the robot end-effector and the grasped part. Uncertainties on the object location render assembly tasks with small or even zero tolerance challenging. Similar to the B-EST, we systematically address these uncertainties by introducing a bounded offset transform ($\widehat{=}\ \theta_1$) between the end-effector and the grasped object.

B. Elastic Objects

The physical properties (e. g., bending stiffness) of articulated objects, ranging from simple kinematic structures such as the hinge shown in Fig. 2 to more complex flexible body approximations, are mainly determined by the stiffness

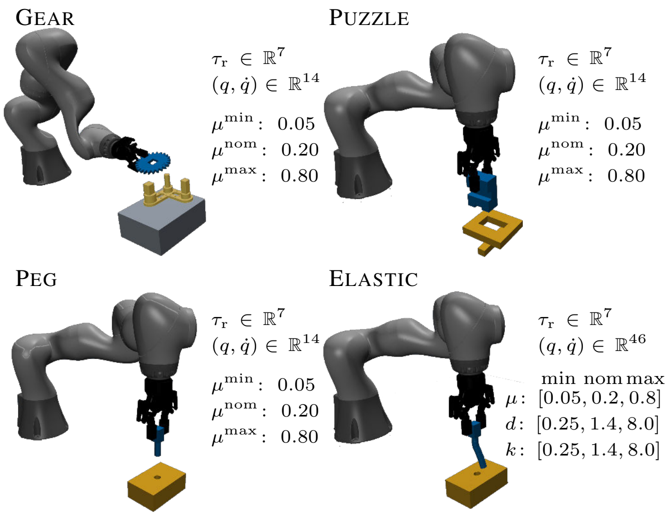
GEAR
$\tau_{\mathrm{r}} \in \mathbb{R}^7$
$(q, \dot{q}) \in \mathbb{R}^{14}$
$\mu^{\min}: 0.05$
$\mu^{\mathrm{nom}}: 0.20$
$\mu^{\max}: 0.80$
PUZZLE
$\tau_{\mathrm{r}} \in \mathbb{R}^7$
$(q, \dot{q}) \in \mathbb{R}^{14}$
$\mu^{\min}: 0.05$
$\mu^{\mathrm{nom}}: 0.20$
$\mu^{\max}: 0.80$
PEG
$\tau_{\mathrm{r}} \in \mathbb{R}^7$
$(q, \dot{q}) \in \mathbb{R}^{14}$
$\mu^{\min}: 0.05$
$\mu^{\mathrm{nom}}: 0.20$
$\mu^{\max}: 0.80$
ELASTIC
$\tau_{\mathrm{r}} \in \mathbb{R}^7$
$(q, \dot{q}) \in \mathbb{R}^{46}$
min nom max
$\mu: [0.05, 0.2, 0.8]$
$d: [0.25, 1.4, 8.0]$
$k: [0.25, 1.4, 8.0]$

Fig. 5. Benchmark scenarios targeting industrial assembly scenarios. Shown are the dimensionalities of the actuation- and state-space as well as the assumed bounds of the parameter-space.

and damping coefficients of the underlying joint primitives. Under the assumption of uniform stiffness and damping coefficients, we can write the generalized force attributed to the system dynamics as

$$h\left(q,\, \dot{q}\right) = \operatorname{diag}\left\{\underline{0}^{n_{\mathrm{r}}},\, k\,\underline{1}^{n_{\mathrm{o}}}\right\}\, q + \operatorname{diag}\left\{\underline{0}^{n_{\mathrm{r}}},\, d\,\underline{1}^{n_{\mathrm{o}}}\right\}\, \dot{q}, \quad (7)$$

where $\operatorname{diag}\{\cdot\}$ denotes the matrix operator and d ($\widehat{=}\, \theta_2$) and k ($\widehat{=}\, \theta_3$) are the damping and stiffness, respectively. For the case of a rigid object ($n_{\mathrm{o}} = 0$), the above equation yields $h\left(q,\, \dot{q}\right) = \underline{0}^{n_q}$.

C. Friction

During parts assembly, contact interactions are not only inevitable, but can systematically be exploited in order to address system uncertainties. The dynamics during contact are in large parts determined by the (possibly non-constant) friction μ ($\widehat{=}\, \theta_4$) of the involved materials, i. e., the constraint torque τ_c is a function of friction μ. At this point, we omit a further discussion on the broad details of contact dynamics.

VI. RESULTS

In the following, we introduce the benchmark tasks with regard to which we assessed our algorithms (Section VI-A) and provide implementation details in Section VI-B. Section VI-C analyzes our method with respect to several key criteria. The real-world experiments we conducted and their results are presented in Section VI-D.

A. Benchmark Scenarios

We evaluated our planner with regard to the four benchmark scenarios shown in Fig. 5. The GEAR assembly task is a benchmark problem set out by Siemens[1] that specifically addresses the demands in industrial automation. It requires placing a gear with a square opening onto a square-shaped peg. The clearance is 1 mm. The PEG task is a well studied assembly benchmark, since it covers a wide spectrum of industrial applications. The clearance between peg and hole is 0.5 mm. The PUZZLE problem was introduced in [4]. Solving it requires three consecutive joining processes with tolerances of 1-2 mm. The ELASTIC benchmark illustrates our method's applicability to the joining of non-rigid, deformable objects such as encountered in wiring tasks. A short, wire-like object is approximated by means of eight interconnected cylindrical primitives. It is important to stress that we did not rely on tailored projections to lower-dimensional manifolds. All problems were solved in the full (up to 46-dimensional) state space.

B. Implementation

The integration of the system-dynamics in contact (1) was carried out using the MuJoCo physics-engine [19]. We simulated each of our benchmarks at an integration step-size of 5 ms. The dynamics were integrated in parallel using OpenMp [20]. The analysis of all algorithms was carried out on two Intel Xeon E5-2640v4 (2.4 GHz) ten-core processors. The present work addresses offline scenarios. Yet, the time required to compute assembly trajectories consistently lay within the range of a few minutes. The real-world experiments were conducted using a KUKA iiwa R800 redundant 7-axis manipulator. As concrete realization of compliant robot-environment interaction, we used joint-level impedance control with gravity compensation. The resulting control law (2) is

$$\tau_{\mathrm{r}} = M_{\mathrm{r}}\left(q_{\mathrm{r}}\right)\, \ddot{q}_{\mathrm{r}}^{\mathrm{d}} + \underbrace{K\left(q_{\mathrm{r}}^{\mathrm{d}} - q_{\mathrm{r}}\right) + D\left(\dot{q}_{\mathrm{r}}^{\mathrm{d}} - \dot{q}_{\mathrm{r}}\right)}_{\text{imposed dynamics}} + g_{\mathrm{r}}\left(q_{\mathrm{r}}\right),$$

where $K \in \mathbb{R}^{n_{\mathrm{r}} \times n_{\mathrm{r}}}$ and $D \in \mathbb{R}^{n_{\mathrm{r}} \times n_{\mathrm{r}}}$ are the diagonal, positive-definite stiffness and damping gains and $M_{\mathrm{r}} \in \mathbb{R}^{n_{\mathrm{r}} \times n_{\mathrm{r}}}$ is the manipulator's inertia matrix. The above control law compensates for the robot dynamics and imposes a torsional spring-damper behavior on each of the joints. We chose a stiffness value of $K = \operatorname{diag}\left\{\underline{1}^7\, 200\right\}^{\mathrm{Nm}/\mathrm{rad}}$ during all experiments. The damping term D was computed using double-digitalization damping design [21]. We sampled end-effector targets in Cartesian space and computed the corresponding joint-space reference using the eTaSL/eTC framework [22].

C. Simulation Study & Analysis

In the following, we show that the assembly trajectories as obtained from kinodynamic motion planners, are likely to fail even at minor deviations between model and real-world. We demonstrate that the MB-EST algorithm performs well even in situations where there is a significant deviation between the assumed and actual model parameters.

The experiments evaluate and compare the following planning methods: firstly, the EST planner [10], a standard reference for kinodynamic motion planning, secondly the B-EST planner [1], a motion planner for robust trajectories in the face of state uncertainties and, finally, the MB-EST planner.Throughout our experiments, we assumed an uncertain translational offset between object and gripper sampled

---

[1] https://www.siemens.com/us/en/home/company/fairs-events/robot-learning.html

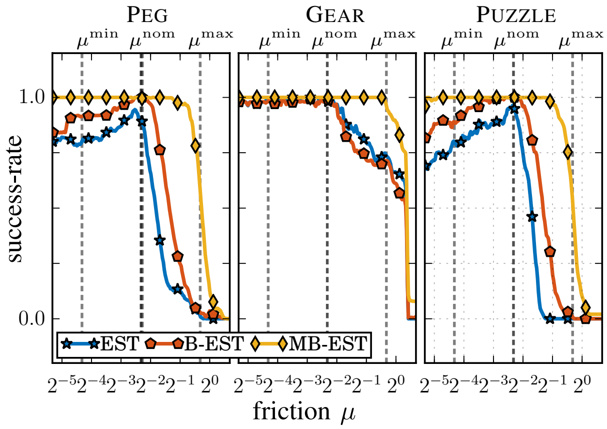

Fig. 6. Success-rate during assembly execution as a function of the friction parameter μ. During planning, the B-EST and the EST algorithm assume nominal parametrizations μ^{nom}.

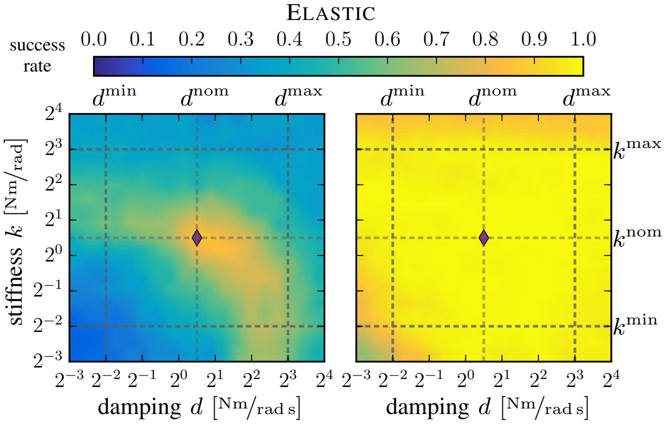

Fig. 7. Success-rate as a function of the stiffness- and damping-coefficient of elastic materials. Left: success-rates for the EST, assuming a nominal parametrization $(\cdot)^{\text{nom}}$. Right: Success-rate of the MB-EST assuming parameters intervals $\left[(\cdot)^{\min}, (\cdot)^{\max}\right]$. The EST is significantly more susceptible to a deviation from the nominal parameters assumed during planning.

within a range of ± 2 mm. The assumed parameter ranges for friction, damping and stiffness, the nominal parametrizations, as well as the dimensionality of the state-space are illustrated in Fig. 5.

In a first step, we computed 50 assembly trajectories for each of the benchmarks. We planned the EST and B-EST trajectories assuming nominal parametrizations $(\cdot)^{\text{nom}}$.

Next, we executed each of the trajectories 4000 times, whereby we drew the model parameters for each execution uniformly on a logarithmic scale (base 2). Fig. 6 and Fig. 7 depict the planners' success-rates with regard to the employed parameter values. The success-rate was obtained by binning (200 bins) the outcome of individual executions (success-failure) over the range of parameter values.

The success-rates for the EST and B-EST planners experience a drop especially for high values of material friction and unsurprisingly are at maximum when the simulation model closely matches the execution environment. Fig. 4 shows how an assembly trajectory computed for a nominally assumed friction coefficient fails, as soon as deviations between the planning- and execution-environment occurs.

The results in Fig. 6 emphasize how susceptible motion plans are to a mismatch between the assumed parametrization and the actual physical parameters. In contrast, our method yields success-rates near 100 % over a parameter range from an eighth up to eight times the nominal values. Fig. 4 demonstrates our method's ability to compute trajectories that work despite a potential mismatch between simulation and reality. As shown in Fig. 7, the effects are even more distinct when considering multiple parameters subject to uncertainty.

The fact that we use a set of N particles in order to represent a continuous parameter range raises the question as to how many particles one requires in order to produce adequately robust assembly trajectories. For this purpose, we repeated the above experiment, varying the number of particles. Fig. 8 depicts our planner's success-rate w.r.t. the number of particles N. One can observe that even few particles are sufficient to retrieve robust trajectories. We are well aware that the number of particles required grows exponentially in the dimension of the continuous parameter-space. Since a mere five particles can be sufficient to achieve robust results in a single dimension, we remain confident that our method also scales well to other problems of practical relevance.

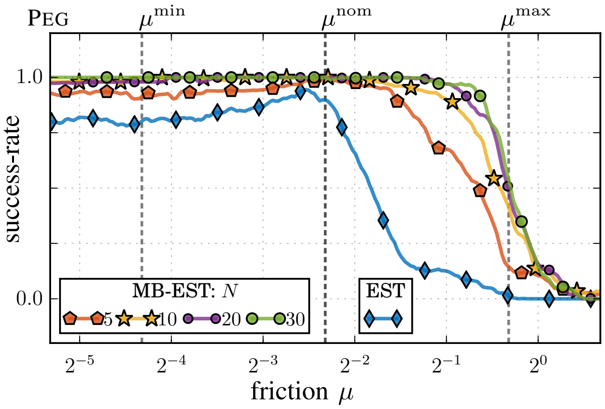

Fig. 8. Success-rate during execution of the PEG benchmark as a function of the friction parameter μ assumed during planning, for different numbers of particles N.

D. Real-World Experiments

In addition to simulated experiments, we evaluated our method with regard to two real-world assembly tasks. In total, we conducted 300 runs on hardware. The first experiment realizes the PEG benchmark (clearance 0.5 mm) depicted in Fig. 5. We executed 30 trajectories of both B-EST and MB-EST with four different surface materials for peg and hole. Sorted in ascending order according to surface friction: lubricated plastic, plasitc, paper, and sandpaper. Table I shows the success-rate for individual friction values and in total. Additionally, we tested the two planners against the null-hypothesis of identical success-rates. The results demonstrate that trajectories as obtained from MB-EST perform reliable regardless of the true friction. By assuming a wide range of possible friction parameters, MB-EST performs significantly better (p-value: 0.028 %) than a planner assuming a nominal friction value.

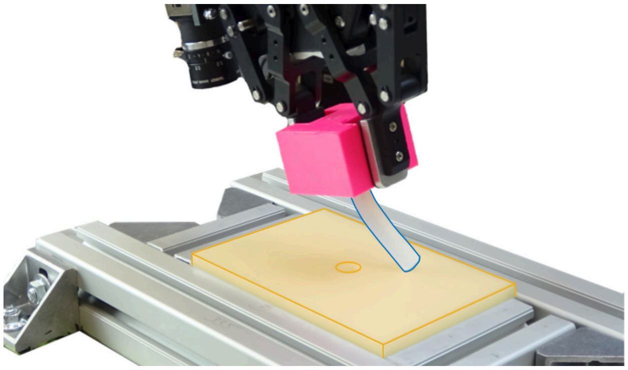

Fig. 9. Experimental setup for an elastic peg-in-hole scenario using a KUKA iiwa R800 redundant 7-axis manipulator. The clearance between peg and hole is 0.5 mm
.

TABLE I

REAL-WORLD EXPERIMENTS

SUCCESS RATES AND FISHER'S EXACT T-TEST

PEG	plastic, lubr.		plastic		paper		sandpaper		Σ	
planner	B-EST	MB-EST	B-EST	MB-EST	B-EST	MB-EST	B-EST	MB-EST	B-EST	MB-EST
success rate	**77%**	**97%**	**70%**	**93%**	**90%**	**90%**	**77%**	**97%**	**78%**	**94%**
p-value	2.6 %		2.1 %		>10 %		2.6 %		0.028 %	

The second experiment resembles the ELASTIC benchmark with a rubber-like, elastic peg material. The experimental setup is shown in Fig. 9. We computed 30 trajectories for both MB-EST and B-EST. The trajectories computed using MB-EST achieved the peg insertion in all runs (30/30) compared to a 50 % success-rate achieved by B-EST (15/30). We noticed that the elastic behavior of the peg gradually changed over the course of the experiment in that it became softer. MB-EST performed successfully despite an ongoing change in material properties. This highlights the advantage of assuming a wide range of elasticity characteristics during planning.

VII. CONCLUSION

This paper presented a novel belief-space planner for compliant assembly. It extends our previous work in two ways: We incorporate the state of deformation for articulated or elastic objects into the configuration-space of the planner. This allows to plan assembly tasks with elastic objects such as wires. Furthermore, we include model parameters, such as friction, into the belief-space. This makes the approach more robust to inaccurate estimates of these parameters.

We analyzed the impact on robustness in extensive experiments, both in simulation and on a real robot. These simulations and several hundred real-world experiments show, that the proposed approach has significantly higher success rates than two state-of-the-art planners. The experiments conducted show that our method yields robust assembly sequences, despite a potentially wide gap between model and real world. This enables autonomous assembly without beforehand parameter-tuning.

REFERENCES

[1] F. Wirnshofer, P. S. Schmitt, W. Feiten, G. v. Wichert, and W. Burgard, “Robust, compliant assembly via optimal belief space planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1–5.

[2] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, “Automatic synthesis of fine-motion strategies for robots,” *The International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.

[3] J. Canny, “On computability of fine motion plans,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 1989, pp. 177–182 vol.1.

[4] G. Dakin and R. Popplestone, “Simplified fine-motion planning in generalized contact space,” in *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*, Aug 1992, pp. 281–286.

[5] X. Ji and J. Xiao, “Planning motions compliant to complex contact states,” *The International Journal of Robotics Research*, vol. 20, no. 6, pp. 446–465, 2001.

[6] H. Bruyninckx, S. Dutre, and J. D. Schutter, “Peg-on-hole: a model based solution to peg and hole alignment,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, May 1995, pp. 1919–1924 vol.2.

[7] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[8] S. M. Lavalle, “Rapidly-exploring random trees: A new tool for path planning,” Tech. Rep., 1998.

[9] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[10] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, “Randomized kinodynamic motion planning with moving obstacles,” *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[11] N. A. Melchior and R. Simmons, “Particle rrt for path planning with uncertainty,” in *IEEE International Conference on Robotics and Automation (ICRA)*, April 2007, pp. 1617–1624.

[12] A. Sieverling, C. Eppner, and O. Brock, “Interleaving motion in contact and in free space for planning under uncertainty,” in *International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 4011–4073.

[13] E. Páll, A. Sieverling, and O. Brock, “Contingent contact-based motion planning,” in *International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 6615–6621.

[14] J. Van Den Berg, P. Abbeel, and K. Goldberg, “Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[15] S. Patil, J. Van Den Berg, P. Abbeel, and K. Goldberg, “Motion planning under uncertainty in highly deformable environments,” *Robotics: Science and Systems VII*, 2011.

[16] B. D. Luders, M. Kothari, and J. How, “Chance constrained rrt for probabilistic robustness to environmental uncertainty,” in *AIAA guidance, navigation, and control conference*, 2010, p. 8160.

[17] B. D. Luders, S. Karaman, and J. P. How, “Robust sampling-based motion planning with asymptotic optimality guarantees,” in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 5097.

[18] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, “Firm: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements,” *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2014.

[19] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 5026–5033.

[20] L. Dagum and R. Menon, “Openmp: an industry standard api for shared-memory programming,” *IEEE computational science and engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[21] A. Albu-Schäffer, C. Ott, U. Frese, and G. Hirzinger, “Cartesian impedance control of redundant robots: recent results with the dlr-light-weight-arms,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, Sept 2003, pp. 3704–3709 vol.3.

[22] E. Aertbelin and J. De Schutter, “etasl/etc: A constraint-based task specification language and robot controller using expression graphs,” in *International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2014, pp. 1540–1546.