Mechanical Part Assembly Planning with Virtual Mannequins

Jean-Paul Laumond [†] Etienne Ferré [*] Gustavo Arechavaleta[†] Claudia Esteves[†]

[†]LAAS-CNRS
7 Avenue du Colonel Roche
31077 Toulouse, France

[*]Kineo CAM
Prologue, BP 27201
31672 Labège, France

Abstract

This paper deals with mechanical part assembly planning. The goal is to automatically compute a collision-free path for both the part to be assembled and the mannequin manipulating it. Two approaches are proposed according to the difficulty of the problem. Both are based on a general probabilistic diffusion algorithm working in the configuration space of the considered system. The first approach consists in first planning a path for the part alone and then in checking the feasibility of the solution by adding the mannequin. The second one considers the part grasped and the mannequin as a single system. While the first approach performs quickly the second one is able to solve more constrained and difficult cases. Both solutions are based on the same path planning library allowing the user to easily evaluate the proposed solutions. Experimental results based on feedback experiences in automotive industry are presented.

1 Introduction

1.1 Problem Statement

This paper aims at providing an efficient and practical solution to the collision-free path planning problem applied to a mobile mechanical part to be assembled by a digital mock-up. The main contribution is to add a virtual mannequin in the process of part assembly simulation. Part assembly planning is usually considered as a 6-dimensional path planning problem for a free-flying object moving in a 3D digital mock-up. Such a context demands from algorithmic path planning challenging issues, such as fine motion planning, narrow passage following . . . Adding a human mannequin in the reasoning loop increases the difficulty. Indeed a virtual mannequin cannot necessarily execute the collision-free path computed for the part to be assembled.

This is due to the motion constraint imposed by the human body. Is a given assembly path feasible by a mannequin grasping the part? This is the first problem addressed in this paper. When feasibility checking fails, then the reasoning should consider the part and the human body as a single system. In that case the original 6-dimensional problem becomes a much higher dimensional one: all the mannequin degrees of freedom involved in the grasping task should be considered at the planning level. Moreover, the problem may be further complicated by considering assembly tasks involving two mannequin arms: in that case, the whole system becomes a closed chain system. Then, in the worst case, the assembly problem appears as a path planning problem for a high-dimensional closed chain system moving in a highly cluttered environment. This is the second issue addressed in the paper.

1.2 Related work and contribution

Motion planning for mannequins or digital actors becomes an active research area. With respect to the manipulation context we are addressing here, the contributions mainly focus on inverse kinematics.

General models for mannequin inverse kinematics have been proposed in [29, 4]. These models use the numerical algorithms described in [22] which are based on pseudo-inverse Jacobian transforms. On the basis of this machinery, a planning algorithm for digital actors is proposed in [28]. In [15] a dedicated kinematic model of mannequins is proposed allowing efficient computations in the context of grasping tasks including some capabilities of obstacle avoidance. In [3] the problem of manipulating while walking is tackled from a functional decomposition of the mannequin degrees of freedom. None of these works addresses *highly constrained spaces* which are the main feature of part assembly.

With respect to path planning in highly cluttered spaces, our work benefits from the current active research area in probabilistic motion planning. It gathers several published ideas in an integrated solution, mainly: probabilistic path planning and inverse kinematics.

After the pioneering work [6] that combines random walks and gradient descents in the configuration space, two kinds of search paradigms are investigated with success. The *sampling* approaches introduced in [16] consist in computing a so-called roadmap whose nodes are collision free configurations chosen at random and whose edges model the existence of collision free local paths between two nodes. Sampling approaches aim at capturing the topology of the collision free configuration space both in terms of covering and connectivity in a learning phase. The *diffusion* approaches introduced in [13, 19] consist in solving single queries by expanding a tree rooted at the start configuration towards the goal to be reached. How to sample or diffuse within the collision-free configuration space efficiently? Such questions give rise to numerous variants of the original algorithms (e.g., [2, 23, 26]).

As part assembly is concerned, diffusion techniques clearly behave better than random sampling ones. Indeed, the solution space of a assembly problem has the shape of a long thin tube. Random sampling within a tube requires a high density of points while tree diffusion benefits from the shape of the tube that naturally steers the diffusion process.

On the basis of this simple statement, several technical issues remain to be solved. How to control the diffusion process? For efficiency purpose, it should progress fast in empty rooms and more slowly in constrained spaces. How to guarantee the safeness of the solution path under an imposed user-defined clearance threshold? Such a constraint depends on the problem and should be automatically taken into account by the method. We have recently developed a dedicated iterative algorithm [10] addressing both issues. It applies for free-flying bodies as well as for articulated body open chains. Part manipulation with two hands requires to deal with closed kinematics chains. Here we use a complementary algorithm already published in [9]. Both algorithms are summarized in Section 2.

The contribution of our paper mainly deals with an integrated view of part assembly with articulated mannequins. Two main strategies are presented (respectively in Sections 3.2 and 3.3):

- the first one consists in first planning an assembly path; then an inverse kinematics operator computes the motions the mannequin has to perform to execute the assembly path.

- the second strategy is used when the first one fails. It consists in searching the configuration space of both the mannequin and the grasped part altogether. In such a case, the system appears as a high-dimensional system which makes the planning task more challenging. However, this strategy allows to address more complicated cases than the previous one.

2 Geometric Development Kit

This section summarizes the basic geometric tools both assembly strategies require.

2.1 Iterative path planning algorithm

The path planning algorithm we use is dedicated to highly constrained spaces where the computed motion is close to the contact space. The algorithm is iterative. A first path is computed allowing some penetration in the obstacles. Then the current paths are iteratively re-shaped by decreasing the allowed penetration threshold. The cases of failure of the iterative process are automatically detected and solved.

The approach benefits from several ideas (see [10] for details):

- As collision checking is concerned, a critical problem is to perform efficient collision checking (see overviews in [21, 14]) not only for configurations but also for local paths. Exact collision checking along computed paths has been recently addressed in [24]: path collision checking is performed with a static collision checker while the (usually costly) iterative process is speeded up thanks to distance computations. Here we show how the approach may be adapted to account for the user-defined imposed clearance constraint.

- To overcome the expansive cost of configuration and path collision checking, some approaches have been defined to put back the tests and then to avoid useless computations. This is the case of the lazy approaches [7, 23] where the algorithms put back collision checking as long as the probability of failure is high. Our approach consists in starting from a rough solution path and iteratively refining it. The iterative procedure is based on an original penetration distance control. When the refinement procedure fails (i.e. when the current path cannot be locally re-shaped into a collision-free one), then the search re-starts with a roadmap composed of the portions of the path that are collision-free. That kind of procedure has been recently introduced in [1] to improve the connectivity of roadmaps.

- Another key point is the control of the diffusion process: how to steer the diffusion process without introducing useless side effects? For instance, defining a new diffusion direction at random by fixing a new configuration goal (as in [19]) gives rise to a bias in introducing implicit bounding boxes on the translation parameters. This problem has been already noticed as critical for assembly problems in [13]. The solution in [13] depends on a local grid whose resolution appears as a parameter to be tuned. The solution we propose is parameter free (see [10] for details).

- Finally our refinement procedure for path reshaping follows the same idea as the variation approach (introduced in [5] and developed in [12]) where the search is performed by iteratively growing formerly shrunk obstacles. In our approach, the growing process is automatically controlled. Moreover the failures due to the closure of passages at some stage of the growing are automatically solved.

The algorithm is general. It works for free-flying objects as well as for articulated mechanisms. The computational performance in mechanical part assembly (without mannequins) is reported in [10].

2.2 Path planning for closed kinematic chains

When the assembly task should involve both mannequin arms, a closed kinematic loop appears between the mannequins bodies and the part (Figure 1). The second strategy we propose below requires to handle path planning in the combined configuration space of both the mannequin and the part. In order to handle the motions of closed kinematic mechanisms, some path planning methods have been proposed in the literature [20, 11, 8].

In our work we have chosen to use the Random Loop Generator (RLG) proposed in [8]. The principle consists in dividing the closed kinematic chain into active and passive parts. The main idea of the algorithm is to decrease the complexity of the closed kinematic chain at each iteration until the active part becomes reachable by all passive chain segments simultaneously. The notion of reachable workspace of a kinematic chain is introduced: it is defined as the volume which the end-effectors can reach. An approximation of such a volume is automatically computed by the RLG using a simple bounding volume (spherical shell) consisting in the intersection of concentric spheres and cones. A guided random sampling of the configurations of the active part is done inside the computed shell and within the current joint limits. The values of the passive chain parameters are computed by solving an inverse kinematics problem (Section 2.3). With respect to the other methods, the computation of the reachable space speeds up the search.

Once the roadmap is constructed, a path is found in the same way as for open kinematic chains.

2.3 Inverse kinematics and reachable space computation

In order to synthesize the coordinated manipulation paths between the virtual mannequin and the part to be manipulated in the first strategy below, we need to solve the inverse kinematics problem aiming at fixing the mannequin hands on the part.

Kinematics based techniques specify motion independently of the underlying forces that produced them. Motion can either be defined by specifying the value of each joint (forward-kinematics) or it can be derived from a given end-effector configuration (inverse-kinematics). In this work we are especially interested in generating the motions of a mannequin. In computer animation this approach has been frequently used to generate the motions of articulated human characters as in [30, 29, 4]. Several inverse kinematics (IK) algorithms for 7-DOF anthropomorphic limbs have been developed based on biomechanical data in order to best reproduce human-arm motions (e.g., [18, 27]). In our work we have chosen to use the analytic IK method presented in [27]. Kinematics-based methods are well adapted when a specified target is given, like in reaching motions.

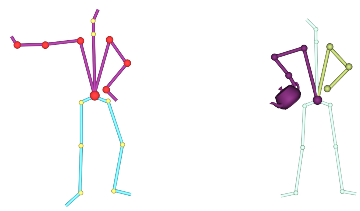
3 Planning Strategies

3.1 Model of mannequin

The mannequin model we are considering is depicted in Figure 1. The mannequin is made of 41 degree of freedom. In our approach, the number of degrees of freedom considered for planning is reduced to 15 (Figure 1(a)): each arm is modeled as a simple 7-dof kinematic chain; an additional degree of freedom in rotation is located at the pelvis. We may then apply the analytical solution of IK [27] for the arms. The pelvis dof allows the articulated mannequin to increase the reachable space of the hands. A decomposition of the closed chain involved in the grasping posture is required by the algorithm RLG (Section 2.2). It is illustrated in Figure 1(b): the active chain is constituted by an arm with the part attached to the corresponding hand; the passive chain corresponds to the other arm.

3.2 First strategy

The first strategy we propose consists in first planning a path for the part to be assembled (Figures 2(a) and 3(a)). We assume that the part lies in the reachable space of the mannequin at any time of the assembly motion. Then the mannequin is added by using the inverse kinematics operator above: the user has just to specify the position of the hands on the part (Figures 2(b) and 3(b)). In this way, a



(a) (b)

Figure 1: Mannequin model: (a) The arms and pelvis 15 dof are considered for planning. (b)The close chain is decomposed into an active chain (in dark violet) ending with the object attached to it and a passive part (light green) that closes the chain.

new path involving all the bodies of both the mannequin and the part is provided. This new path should be checked with respect to collision avoidance. If no collision occurs the feasible of the assembly path is guaranteed (Figures 2(e). The method applies for one hand manipulation as well as for two hands manipulation.

This simple strategy just requires the path planning algorithm and the inverse kinematic operator above. There is no need to call for closed loop chain operators. However refinements may be introduced at this stage to make local corrections on the arm positions: indeed the human arms are redundant systems and it is possible to tune the arm motions locally to remove small undesirable collisions [3].

In this strategy the rotational degree of freedom at the pelvis is animated by a simple linear interpolation between the initial posture and the final one.

3.3 Second strategy

The second strategy makes use of all the operators introduced in Section 2. It consists in planning a collision path for the whole 15 dof system, i.e. the mannequin grasping the part. The inverse kinematics operator is used to specify the initial and the final configurations of the whole system (indeed, an assembly problem is usually defined just by specifying the initial and the final configurations of the part alone).

In the case of a one hand manipulation, the basic iterative path planning algorithm (Section 2.1) is applied as such without any additional refinement.

Two hand manipulation requires a more technical integration of the RLG algorithm within the general iterative one: RLG is used to sample the configurations (in the iterative algorithm) which should be admissible with respect to the closure of the chain; it is also used to check the collisions along the local paths computed during the diffusion process.

4 Experimental Results and Comparison

4.1 Worked-out examples

Both approaches to assembly planning have been integrated within KineoWorks [1], the industrial software version of Move3D. The evaluation has been conducted in the framework of mechanical part assembly in automotive industry. The mannequin is made of 22,128 polygonal facets. The first example (Figure 2) is dedicated to the assembly of a radiator (total number of polygonal facets: 79,664). It requires a two-hand manipulation. The second one (Figure 3) aims at proving the maintainability of a wind-screen wiper motor by using only one hand (total number of polygonal facets: 26,504). Both use-cases are very constrained: in the radiator example there is no solution path with a clearance greater than $14cm$; this threshold is $1mm$ in the wind-screen wiper example.

Figures 2 and 3 and show the solutions computed with the first strategy. We do not display the solution paths found by the second strategy. Indeed, such paths do not visually differ from the paths computed with the first strategy.

The running time for each example is shown on Table 1.

Table 1: Computational time in seconds.

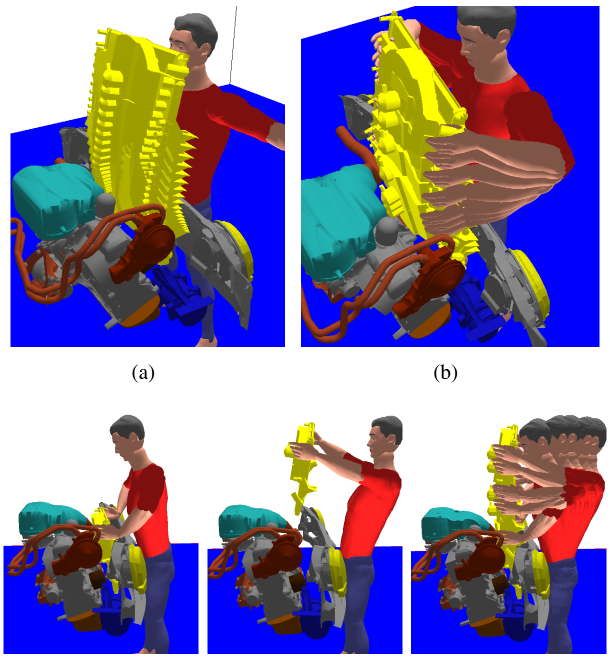
Example	Strategy	Total Time
Wiper	I	49 s
	II	–
Radiator	I	32 s
	II	41 s

5 Conclusion

This paper has presented two possible strategies to simulate part assembly planning for a digital mannequin. The first one solves quickly simple problems while the second one is more time consuming, but it solves more complicated operations. Both of them solved industrial real study-cases involving complex geometric models, highly cluttered environments and out of the scope of previous approaches.

The current extensions deal with addressing more complete models of the mannequins. For instance we want to consider the possibility for the mannequins to bend its legs. In that case the inverse kinematic problem becomes more general et requires numerical solution that will certainly affect the performance of the algorithm.

[1] KineoWorks is the path planning dedicated Software Development Kit developed by Kineo CAM.



(a) (b)

(c) (d) (e)

Figure 2: Radiator assembly: A first collision-free path is computed for the part alone (Strategy I) (a). Then, the mannequin follows the path (b) with a grasping posture automatically computed by IK (e). Initial and final configurations are shown in (c) and (d).

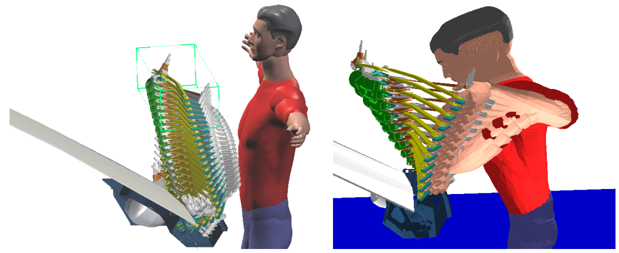
Finally, assembly tasks often require re-grasping to change the position of the hands on the object. Assembly planning in such a context has been already addressed (e.g., [17, 25]). Nevertheless it has never been applied to the real sized problems involving mechanical parts and their motions close to the contact spaces.

Videos

Videos related to this work can be found at http://www.laas.fr/RIA/RIA-research-motion-character.html

Acknowledgment

The authors would like to thank Renault for permitting the publishing of the use-cases. This work benefits from advices by J. Cortés on the integration of the RLG algorithm. It has been partially funded by the IST-2001-39250 MOVIE project from the European Community. G. Arechavaleta and C. Esteves are supported by the Mexican Conacyt program.



(a) (b)

Figure 3: Windscreen wiper motor assembly (Strategy 1): A first collision-free path is computed for the part alone (a). Then the mannequin follows the path with a grasping posture automatically computed by IK (b).

References

[1] M. Akinc, K. Bekris, C. Chen, A. Ladd, E. Plakue, and L. Kavraki, “Probabilistic roadmaps of trees for parallel computation of multiple query roadmaps,” in *International Symposium on Robotics Research ISRR*, 2003.

[2] N. Amato, O. Burchan, L. K. Dale, C. Jones, and D. Vallejo, “OBPRM: An obstacle-based PRM for 3D workspaces,” in *Proc. of International Workshop on Algorithmic Foundations of Robotics WAFR*, 1998.

[3] G. Arechavaleta, C. Esteves, and J.-P. Laumond, “Planning fine motions for a digital factotum,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[4] P. Baerlocher and R. Boulic, “An inverse kinematics architecture enforcing and arbitrary number of strict priority levels,” *Visual Computer*, vol. 20, pp. 402–417, 2004.

[5] J. Barraquand and F. Ferbach, “A penalty function method for constrained motion planning,” in *Proc. IEEE International Conference on Robotics and Automation ICRA*, 1994.

[6] J. Barraquand and J.-C. Latombe, “Robot motion planning: A distributed representation approach,” *Robotics Research*, vol. 10, no. 6, 1991.

[7] R. Bohlin and L. E. Kavraki, “Path planning using lazy PRM,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[8] J. Cortés and T. Siméon, "Probabilistic motion planning for parallel mechanisms," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[9] J. Cortés, T. Siméon, and J.-P. Laumond, "A random loop generator for planning the motions of closed kinematic chains using prm methods," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[10] E. Ferré and J.-P. Laumond, "An iterative diffusion algorithm for part disassembly," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[11] L. Han and N. Amato, "A kinematics-based probabilistic roadmap method for closed chain systems," in *Proc. of International Workshop on Algorithmic Foundations of Robotics WAFR*, 2000.

[12] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin, "On finding narrow passages with probabilistic roadmap planners," in *Proc. of International Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998.

[13] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *International Journal of Computational Geometry and Applications*, vol. 9, no. 4/5, pp. 495–, 1999.

[14] P. Jiménez, F. Thomas, and C. Torras, "3d collision detection: A survey," *Computers and Graphics*, vol. 25, no. 2, 2001.

[15] M. Kallman, A. Aubel, T. Abaci, and D. Thalmann, "Planning collision-free reaching motions for interactive object manipulation and grasping," in *Proc. ACM SIGGRAPH/Eurographics*. Eurographics Association, 2003.

[16] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. 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.

[17] Y. Koga, K. Kondo, J. Kuffner, and J.-C. Latombe, "Planning motions with intentions," in *Proc. of SIGGRAPH*. ACM Press, 1994, pp. 395–408.

[18] K. Kondo, "Inverse kinematics of a human arm," *Journal of Robotics and Systems*, vol. 8, no. 2, pp. 115–175, 1991.

[19] J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, California, 2000.

[20] S. M. LaValle, J. Yakey, and L. E. Kavraki, "A probabilistic roadmap approach for systems with closed kinematic chains," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[21] M. Lin, D. Manocha, J. Cohen, and S. Gottschalk, "Collision detection: Algorithms and applications," in *Proc. of the International Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1996.

[22] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley, 1991.

[23] G. Sánchez and J.-C. Latombe, "A single-query bidirectional probabilistic roadmap planner with lazy collision checking," in *International Symposium on Robotics Research (ISRR)*, 2001.

[24] F. Schwarzer, M. Saha, and J.-C. Latombe, "Exact collision checking of robots paths," in *Proc. of International Workshop on Algorithmic Foundations of Robotics (WAFR'02)*, 2002.

[25] T. Siméon, J.-P. Laumond, J. Cortés, and A. Sahbani, "Manipulation planning with probabilistic roadmaps," *Robotics Research*, 2003, accepted with minor revisions.

[26] T. Siméon, J.-P. Laumond, and C. Nissoux, "Visibility based probabilistic roadmaps for motion planning," *Advanced Robotics*, vol. 14, no. 6, 2000.

[27] D. Tolani, A. Goswami, and N. I. Badler, "Real-time inverse kinematics techniques for anthropomorphic limbs," *Graphical Models*, vol. 62, pp. 353–388, 2000.

[28] K. Yamane, J. Kuffner, and J. K. Hodgins, "Synthesizing animations of human manipulation tasks," in *Proc. of SIGGRAPH*, 2004.

[29] K. Yamane and Y. Nakamura, "Natural motion animation through constraining and deconstraining at will," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 352–360, 2003.

[30] J. Zhao and N. I. Badler, "Inverse kinematics positioning using nonlinear programming for highly articulated figures," *ACM Transactions on Graphics*, vol. 14, no. 4, 1994.