Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 44 (2016) 435 – 440

www.elsevier.com/locate/procedia

6th CIRP Conference on Assembly Technologies and Systems (CATS)

Controller hierarchies for efficient virtual ergonomic assessments of manual assembly sequences

P. Mårdberg[†]*, Y. Yan[†], R. Bohlin[†], N. Delfs[†], S. Gustafsson[‡], J. S. Carlson[†]

†Fraunhofer-Chalmers Centre, Chalmers Science Park, SE-412 88 Gothenburg, Sweden
[‡]Industrial Path Solutions, www.industrialpathsolutions.se, Sweden

* Peter Mårdberg. Tel.: +46-31-772-42-93; fax: +46-31-772-42-60. *E-mail address:* peter.mardberg@fcc.chalmers.se

Abstract

A novel framework for manikin motion planning has been implemented to reduce the time needed to perform virtual ergonomic assessments of manual assembly sequences. The user feeds high level instructions into a hierarchical controller system. Depending on the state of the manikin and the objects in the environment, the controllers compute a sequence of low level instructions interpreted as path planning instances for the manikin. The result is automatically generated collision-free and ergonomically sound motions that accomplish the assembly tasks. The framework is demonstrated on relevant cases from the industry and the reduction in manual simulation preparation time is proven.

Peer-review under responsibility of the organizing committee of the 6th CIRP Conference on Assembly Technologies and Systems (CATS)

Keywords: Virtual manufacturing; Manual assembly; Digital human modelling; Programming languages; Simulation; Motion planning;

1. Introduction

A Digital Human Modeling (DHM) software is valuable tool in virtual manufacturing that allows simulation of manual assembly work long before any physical product has been built [1]. The goal is to increase the sustainability, not only in terms of products and production, but also from a social point of view.

By simulating manual assembly work, it possible to find and resolve design issues, troublesome assembly sequences, awkward postures, and logistic bottlenecks early in a conceptual development stage. This increases the production quality, considerable reduce the cost of late design changes and the ramp-up time of a manufacturing process [2].

Despite the benefits, there still exist assembly tasks that are not simulated, even if all the necessary data is available. One reason for this is the time consuming and tedious work that is required to setup and to define the motions needed for a manikin to accomplish the task. To make a simulation relevant, the user must ensure that the manikin avoids collision with itself and objects in the environment, that the balance is maintained and that the motions are ergonomically sound throughout the whole assembly task. Thus, manual preparation, even of small assembly cases, may be time consuming.

A formal high level instruction language is introduced in [3] in order to make it easier to instruct the manikins and to reduce the time needed to construct simulations. The instruction language is composed in the same model as the manikins, objects in the environment and their corresponding properties. High level instructions are sent to the model, which generates a set of low level instructions that are used manipulate the automated manikins to perform assembly tasks.

In this work we introduce a framework based on a novel hierarchical controller system. High level instructions are fed into a main controller that interprets the instructions and divides them it into a set of smaller and more specific instructions. The result of the interpretation depends on the state of the manikin and the objects in the environment. In the next step, the main controller feeds the newly generated instructions to sub controllers in the hierarchical structure.

Peer-review under responsibility of the organizing committee of the 6th CIRP Conference on Assembly Technologies and Systems (CATS)
doi:10.1016/j.procir.2016.02.084

Thus, each instruction is interpreted and divided until a leaf controller is reached. A leaf controller generates a set of low level instructions, which are interpreted as a set of path planning instances for the manikin. The result is a sequence of ergonomically sound and collision-free motions that accomplish an assembly task.

A grammar structures the controllers into different levels in the hierarchical tree. Thus, a general controller such as *Assemble* defines sequences of other controllers, whereas a specific controller such as *Grasp* corresponds to a set of low level instructions that on execution generates a grasping motion.

The set of available controllers that may be executed during a simulation depends on the current state of the manikin and on the state of the objects in the assembly station. For instance, if the manikin grasps an object with both hands, it is seen as impossible for the manikin to grasp another object. Moreover, a *Grasp* controller may only be used if there is an object that is available for the manikin to grasp.

The execution of a controller depends on the state space. Several controllers may run in parallel and depending on the state, the controllers may start or pause their execution. This makes it possible to define in which order different parts of the body will be moved. For instance, different parts of the legs are used when moving from a standing to a kneeling posture.

Notice that there is no clear distinction between *planning* and *execution*. The controller concept adopted in this work is intended to handle both. In that respect, a problem description is formulated in the same language as its solution – the difference is that the solution description is more detailed.

The framework has been implemented in the Intelligently Moving Manikins (IMMA) [4] software application and it has been tested on assembly cases with relevance to industrial applications. The results show that less preparation is needed when constructing an assembly simulation.

The main contributions of this paper are (i) a hierarchical controller system that dynamically interprets high level language instructions and recursively generates a sequence of low level instructions needed for the manikin to accomplish the task, (ii) how to automatically generate planning instances for the manikin, and (iii) a fully modular way to reuse and mix controllers of different capability, generality and maturity.

This paper is organized as follows. Section 2 describes the manikin, an overview of automatic path planning methods, and ergonomics. Section 3 describes the proposed controller hierarchy, Section 4 shows two case studies followed by discussion and future work in Section 5. Concluding remarks are found in Section 6.

2. Manikin path planning and assembly models

An essential part of the controller framework is the usage of an automated manikin. If the automated manikin is instructed to grasp an object, then it should be able to automatically reposition itself without colliding with itself and the object in the environment [5,6].

Moreover, it is not sufficient for the manikin to just automatically reposition itself, it also needs to maintain the balance throughout the assembly. The balance calculations take into account the body parts and the objects being carried as well as exterior forces and torques from the environment.

Each object in the assembly station, their corresponding properties, the manikin and the controllers are composed into the same discrete model. In this way it is possible for a controller to execute events in the simulation, but also prevent the manikin from performing an assembly action unless all logical preconditions are fulfilled. The assembly model also naturally restricts which instructions are available for interaction with the DHM tool user, and thereby prevent the user from performing contradictory instructions.

2.1. Path planner in ergonomic assessment

Effective simulation of manual assembly operations considering ergonomic load and clearance demands requires detailed modeling of human body kinematics and motions, as well as a tight coupling to powerful algorithms for collision-free path planning [5]. The current path planning tools have been capable of computing and analyzing kinematically complex and dynamic motions of human manikins. However, these tools are not fully automatic and limited to static analysis or simple scenarios.

The locomotion of manikins is usually computed in the paradigm of formulating the kinematics and dynamics of manikins into an optimization problem and solving the problem with non-linear optimization techniques [7,8]. Some researchers go further in this paradigm using dexterous musculoskeletal simulation [9,10]. However, this paradigm cannot be directly used to assembly simulations involving manikins because such a formulation heavily depends on an (almost) feasible initial path. Slight collisions between the manikin and obstacles may be permitted, but such initial paths are difficult to find in the cluttered environments [11,12]. The key reason is that the continuous generalized penetration depth between the manikin and obstacles is difficult to measure and utilize efficiently in the optimization process. Some approaches guide the optimization with motion data obtained through capturing a user’s motions with Kinect or other motion tracking systems [13,14]. But they still suffer from the loss of haptic information and not being general to manikin models with different parameters.

Other researchers use numerical methods to predict whole-body postures and quasi-static motions in complex assembly tasks [15,5]. Usually, a collision-free path of the assembly part is generated first, and then the manikin follows this path according to the grasp settings. This decoupled approach has its limitations – it may not be possible or comfortable for a virtual manikin to follow the path due to the motion constraints imposed by the human body [16], and there is no guarantee that there exist a manikin motion that is collision-free and in dynamic balance. In addition, the quasi-static motion cannot reflect the real magnitude of torques induced at the joints.

To the authors’ knowledge, there is today no product that can automatically plan collision-free paths of manikins in cluttered environments and meanwhile consider the physiological constraints on the dynamics and musculoskeletal level. In this paper, we realize a general whole-body path planner to push the limits.

3. Automatic computation of manikin motions

In order for a DHM tool to gain efficiency and acceptance in industry, a key ingredient is to raise the level of abstraction by letting the user focus on *what* to accomplish rather than *how*. Our ambition is to let the user give high level instructions and then automatically compute feasible postures, motions and events, while restrictions on assembly order and other logical conditions are considered.

The high level instructions are fed into a hierarchical controller system that performs both planning and execution. On each level, instructions are interpreted, plans are created resulting in either execution or sets of instructions passed on to lower level controllers.

3.1. Controller

A controller reads an instruction, and upon subsequent requests by its parent controller it returns a state for each increment of simulation time. Controllers are defined to generate states for specific sets of joint segments on the manikin. For instance, one controller may be defined to generate the motions for the left arm and another for the left leg. The controllers interpret their instructions and depending on the state of the manikin and objects in the environment, the controllers may perform different actions such as executing a new controller, manipulate the manikin or update the state space.

3.2. General whole body path planner

Once a low level instruction is interpreted as a path planning instance, a motion of the manikin incorporating the kinematic constraints, balance, contact forces, collision avoidance and comfort is automatically computed by the whole-body path planner, according to the settings of the instance.

Previous research on motion planning of virtual manikins showed that the optimal control strategy could efficiently generate natural motions in the environments with relatively simple geometry, such as walking on the flat floor or taking stairs [17]. However, this kind of strategies could easily get stuck in a local minimum when considering collision avoidance in highly unstructured and cluttered environments, which is common in most situations of manual assembly simulations.

In this work, a general whole-body path planner for the manikin is realized, by the integration of the sampling based path planner and the flexible pose prediction technique. The previously developed pose prediction technique could efficiently optimize the pose of manikins taking into account the kinematic constraints, balance, contact forces and comfort through Jacobian pseudo-inverse projection [5]. This pose prediction technique provides a perfect property that enforces the random poses of manikins generated by the sampling based path planner to be projected onto the constraint manifolds. In spite of that we use the RRT-connect planner from the Open Motion Planning Library (OMPL) [18] in the demonstration, other sampling based planner are also possible. A similar projection strategy was also seen when planning motions of humanoid robots in the manipulation and multi-contact scenarios in [19].

Due to its generality, the whole-body path planner could seamlessly fulfil the path planning requirements of the low level instructions. In [20], it was proven that the sampling based path planner retains the probabilistic completeness under the Jacobian pseudo-inverse projection. Thereby the statically stable motion found by the whole-body path planner could be used to initialize the motion optimization to avoid local minima, which usually arise in unstructured and cluttered environments.

In our test cases, the whole-body path planner could always create a feasible action corresponding to each low level instruction. The pose prediction technique used by the path planner already demonstrated its flexibility on the assembly operation of rigid, articulated and deformable objects with one or two hands grasping or supporting in different manner [5,21]. Since adding and breaking a contact between the manikin and the environment is the same as adding and releasing a grip, the whole-body path planner could also automatically create the climbing actions when they are inevitable in some difficult assembly simulations.

3.3. Hierarchical controller structure

Based on the overall tasks given to the main controller, the state of the manikin and objects in the environment, a hierarchical system of controllers is dynamically created and destroyed as the simulation time steps forward. As language instructions are interpreted, they are subdivided into simpler and more specific instructions that are recursively passed to sub controllers.

Some controllers, referred to as leaf controllers, are able to fully solve and execute their tasks without further subdivision. Sometimes this is a straightforward execution of a linear motion, but often the execution is preceded by a considerably more challenging task, for example whole body path planning described above or a finding an assembly motion for a part like in the cases presented below.

On leaf-level, the result is a sequence of low level instructions that allows the manikin to accomplish the task through ergonomically sound and collision-free motions.

Formally, the grammar of the instructions structures the controllers into a hierarchical tree [22]. The grammar furthermore defines how the controller sequences are generated and ensures that all the arguments needed by a controller are computed [23].

The set of controllers that are available during a simulation depends on the current state of the manikin and on the objects in the assembly station. For instance, if the manikin grasps an object with right hand, then it is seen as impossible for the manikin to grasp another object with the right hand unless first releasing the current object. Moreover, each controller must have a corresponding action in the simulation. For instance, a *Grasp* controller may only be executed if there is an object that is available for the manikin to grasp. Hence, properties of an object, such as grasp and mating points, further defines the set of available controllers.

The execution of a controller depends on the state space. Several controllers may run in parallel and depending on the state, the controllers may start or pause its execution. This makes it possible to define in which order different parts of the body will be moved. For instance, different parts of the legs are used when moving from a standing to a kneeling posture.

4. Test cases

The framework is demonstrated on two industrial test cases. The two cases illustrates how it is possible for the framework to handle common assembly scenarios such as utilization of support and a two phase assembly.

4.1. Assembly of an electronic control unit

An electronic control unit, called cembox, is assembled below the driving unit of the driver position. In order to accomplish the task the manikin needs to take support with the left hand. An overview of the assembly and a close up of the balance support is shown in Fig. 1. The manikin walks to the table and picks up the cembox, turns towards the car and walks to the assembly position. Since the manikin needs to lean inside the car the left hand is used as support to maintain the balance. In the next step, the cembox is assembled inside the car, see Fig. 2. Fig. 3 illustrates how the assembly task is divided from the main controller into low level instructions.

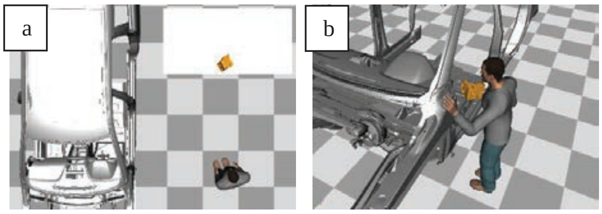

Fig. 1 (a) an overview of the assembly case. The manikin walks to the table, picks up the electronic device (shown in orange) and walks to the assembly position; (b) the manikin uses the left hand to lean onto the car chassis in order to maintain the balance throughout the assembly. Courtesy Volvo Cars.

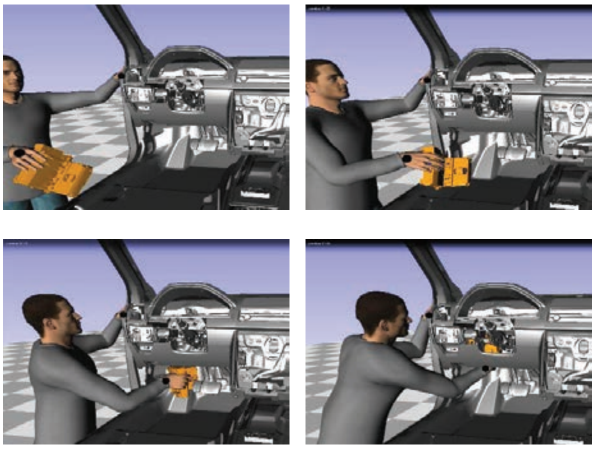

Fig. 2 Four frames of the assembly of the electronic device. Courtesy Volvo Cars.

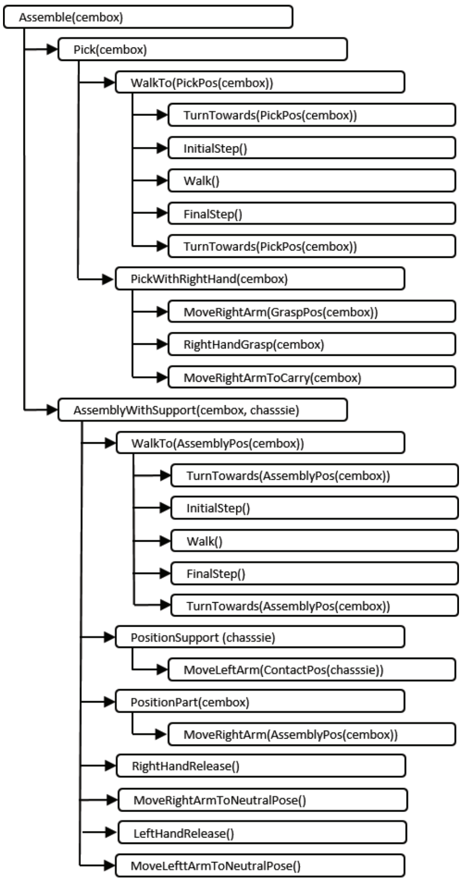

Fig. 3 The hierarchical structure of the controllers when the electronic unit is assembled. High level controllers are listed to the left hand side whereas the low level controllers are listed to the right hand side and intermediate controllers are listed in between.

4.2. Assembly of an alternator

A two phase assembly is performed where an alternator is placed into position at the engine and fixated with a screw. The alternator has to be placed before the screw is mounted and may not be released by the manikin until the screw is properly mounted. The manikin walks to the table and picks up the alternator and the screw. In the next step, the manikin turns and walks to the assembly posture in front of the engine. Finally, the alternator and the screw is mounted, see Fig. 4. In order to accomplish the assembly, the main controller divides the whole assembly task into smaller sub tasks. Fig. 5 show the hierarchical controller structure used to accomplish the task.

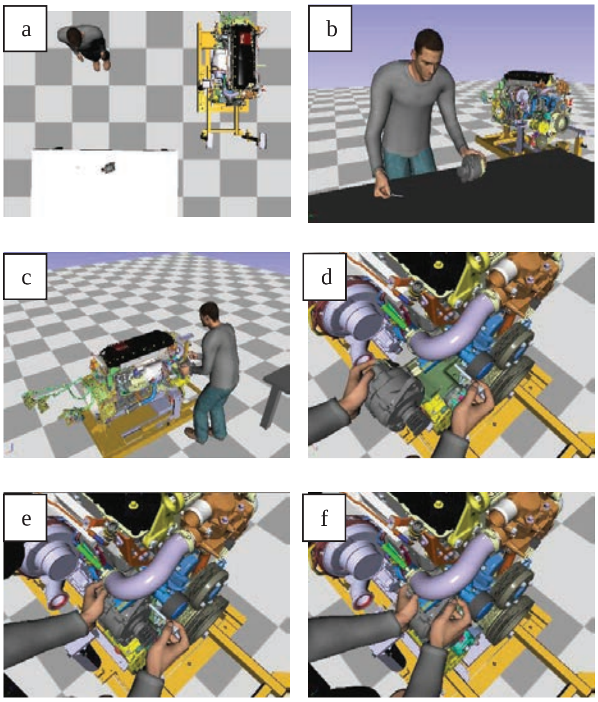

Fig. 4 (a) the starting of the assembly; (b) the manikin walks to the table and picks up the alternator and a screw; (c) the manikin walks and places the alternator into assembly position; (d) a close up of the assembly; (e) the alternator is mounted to the engine; (f) the screw is mounted to fix the alternator. Courtesy AB Volvo.

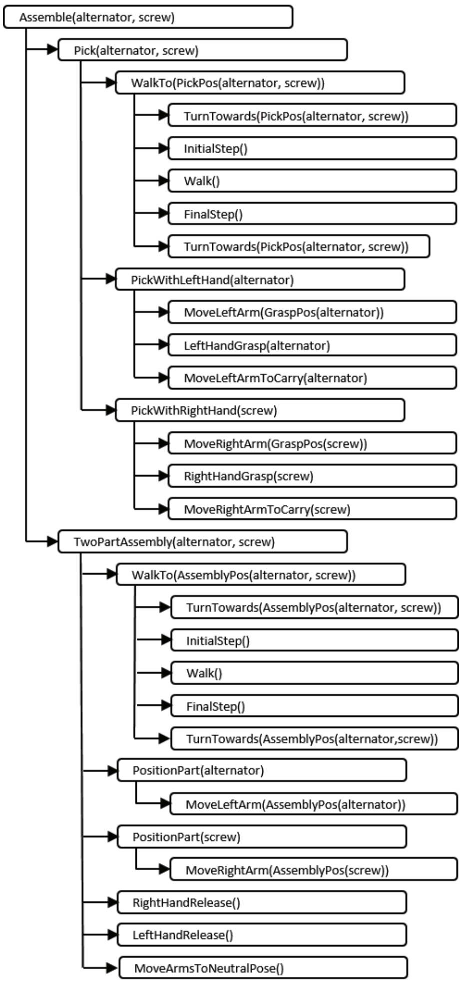

Fig. 5 The hierarchical structure of the controllers when the alternator is assembled. High level controllers are listed to the left hand side whereas the low level controllers are listed to the right hand side and intermediate controllers are listed in between.

5. Discussion

This framework offers a generic and fully modular way to reuse and mix controllers of different capability, generality and maturity. It allows controllers to interact by sending and receiving signals that may be needed when executing low-level instructions in the simulation, for example preventing controllers to execute forbidden instructions, or paths leading to deadlocks. Initially, the user needs to define start and end states, and create grip points on the objects to be used. However, in the forthcoming steps, the information is automatically passed between the subsequent controllers in the hierarchical structure. This creates a seamless transition of information between the different steps of the simulation, which reduces the need for the users to interact and to manually provide information in order to accomplish a simulation

This structure also allows controllers to be run in parallel. That is, manikins are allowed to collaborate with each other, and manikins may interact with the environment such as robots and cells. Moreover, the recursive structure of the controller system makes it possible to create dependencies between different controllers, to halt the computation or change the order in which controllers execute. This allows user input as well as interactive editing on any level in the controller hierarchy, for instance to force a controller to follow a predetermined sequence.

The proposed framework is general and not limited to only be used with manikins. It may be extended to also include robots, conveyors and other automation equipment, which makes it possible to simulate complete assembly sequences. In addition, the framework may be formally verified, which is a methodology to prove if a set of properties hold for a model [24]. It may be used to assure that no deadlock occurs, to verify that no controller violates the model and to ensure that there are no contradictions in the model specification.

6. Conclusion and future work

The hierarchical controller framework presented in this work offers a generic and fully modular way to reuse and mix controllers of different capability, generality and maturity. Furthermore, it also reduces the need of manual user interaction, which decreases the time needed to perform assembly simulations.

The system is implemented for some common industrial scenarios, and its performance is demonstrated on two specific cases.

Moreover, it is a general framework that is not limited to manipulate a single manikin, and may easily be expanded to coordinate multiple manikins, robots, conveyors, etc. in order to simulate an assembly cell. The framework may also be formally verified to avoid deadlocks and prevent violation to the model specification. Work in these directions will be explored in the near future, along with extending and improving the capability of the individual controllers.

Acknowledgements

This work was carried out within the SIO Production 2030 program, the Wingquist Laboratory VINN Excellence Centre, supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA). It is also part of the Sustainable Production Initiative and the Production Area of Advance at Chalmers University of Technology.

References

[1] Laring J. Ergonomic Workplace Design Analysis: Development of a practitioner's tool for enhanced productivity. Göteborg: Chalmers University of Technology; 2004.

[2] Falck A-C, Örtengren R, Högberg D. The Impact of Poor Assembly Ergonomics on Product Quality: A Cost–Benefit Analysis in Car Manufacturing. Human Factors and Ergonomics in Manufacturing & Service Industries, vol. 20, no. 1; 2010. p. 24–41.

[3] Mårdberg P, Carlson JS, Bohlin R, Delfs N, Gustafsson S, Keyvani A, Hanson L. Introducing a Formal High-Level Language for Instructing Automated Manikins. Proceedings of the 2nd International Digital Human Modeling Symposium; 2013.

[4] Hanson L, Högberg D, Bohlin R, Carlson JS. IMMA – Intelligently Moving Manikins – Project Status 2011. Proceedings of the 1st International Digital Human Modeling Symposium; 2011.

[5] Bohlin R, Delfs N, Hanson L, Högberg D, Carlson JS. Automatic Creation of Virtual Manikin Motions Maximizing Comfort in Manual Assembly Processes. Proceedings of the 4th CIRP Conference On Assembly Technologies and Systems; 2012.

[6] Delfs N, Bohlin R, Hanson L, Högberg D, Carlson JS. Introducing Stability of Forces to the Automatic Creation of Digital Human Postures. Proceedings of the 2nd International Digital Human Modeling Symposium; 2013.

[7] Abdel-Malek K, Arora J. Human Motion Simulation: Predictive Dynamics. Academic Press; 2013.

[8] Dai H, Valenzuela A, Tedrake R. Whole-body motion planning with centroidal dynamics and full kinematics. Proceedings of 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids); 2014.

[9] Wang JM, Hamner SR, Delp SL, Koltun V. Optimizing locomotion controllers using biologically-based actuators and objectives. ACM Transactions on Graphics, vol. 33, no. 4; 2012.

[10] Lee Y, Park MS, Kwon T, Lee J. Locomotion control for many-muscle humanoids. ACM Transactions on Graphics (TOG), vol. 33, no. 6; 2014.

[11] El Khoury A, Lamiraux F, Taix M. Optimal motion planning for humanoid robots. Proceedings of IEEE International Conference on Robotics and Automation; 2013.

[12] Schulman J, Duan Y, Ho J, Lee A, Awwal I, Bradlow H, Pan J, Patil S, Goldberg K, Abbeel P. Motion planning with sequential convex optimization and convex collision checking. The International Journal of Robotics Research, vol. 33, no. 9; 2014. p. 1251-1270.

[13] Qiu Z, Escande A, Micaelli A, Robert T. A hierarchical framework for realizing dynamically-stable motions of humanoid robot in obstacle-cluttered environments. Proceedings of 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Osaka; 2012.

[14] Liu M, Micaelli A, Evrard P, Escande A, Andriot C. Interactive virtual humans: A two-level prioritized control framework with wrench bounds. IEEE Transactions on Robotics, vol. 28, no. 6; 2012. p. 1309 - 1322.

[15] Zhou W, Armstrong TJ, Reed MP, Hoffman SG, Wegner DM. Simulating complex automotive assembly tasks using the HUMOSIM framework. SAE Technical Paper 2009-01-2279; 2009.

[16] Laumond JP, Ferré E, Arechavaleta G, Esteves C. Mechanical part assembly planning with virtual mannequins. The Proceedings of 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing, Montreal, Que.; 2005.

[17] Xiang Y, Arora JS, Abdel-Malek K. Optimization-based prediction of asymmetric human gait. Journal of Biomechanics, vol. 44, no. 4; 2011. p. 683–693.

[18] Sucan I, Moll M, Kavraki L. The Open Motion Planning Library. IEEE Robotics and Automation Magazine, vol. 19, no. 4; 2012.

[19] Dalibard S, El Khoury A, Lamiraux F, Nakhaei A, Taïx M, Laumond JP. Dynamic walking and whole-body motion planning for humanoid robots: an integrated approach. The International Journal of Robotics Research, vol. 32, no. 9-10; 2013. p. 1089-1103.

[20] Berenson D, Srinivasa S, Kuffner J. Task Space Regions: A framework for pose-constrained manipulation planning. International Journal of Robotics Research, vol. 30, no. 12; 2011. p. 1435-1460.

[21] Delfs N, Bohlin R, Gustafsson S, Mårdberg P, Carlson JS. Automatic Creation of Manikin Motions Affected by Cable Forces. Proceedings of 5th CIRP Conference on Assembly Technologies and Systems, Dresden, Germany; 2014.

[22] Hopcroft JE., Motwani R, Ullman JD. Introduction to automata theory, languages and computation. Boston : Pearson Addison-Wesley; 2007.

[23] Aho AV, Lam MS, Sethi R, Ullman JD. Compilers : principles, techniques and tools. Boston: Pearson Addison-Wesley; 2007.

[24] Voronov A, Åkesson K. Verification of process operations using model checking. CASE'09 Proceedings of the fifth annual IEEE international conference on Automation science and engineering, Bangalore, India; 2009.