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