Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 44 (2016) 395 – 400

www.elsevier.com/locate/procedia

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

Optimal robot placement for tasks execution

Domenico Spensieri[a], Johan S. Carlson[a], Robert Bohlin[a], Jonas Kressin[a], Jane Shi[b]

[a] *Fraunhofer-Chalmers Research Centre, Geometry and Motion Planning Group, 41288 Göteborg, Sweden*
[b] *General Motors Global R&D Center, Warren, Michigan, US*

* Corresponding author. Tel.: +46-31-7724252; fax: +46-31-7724260. *E-mail address:* domenico.spensieri@fcc.chalmers.se

Abstract

Automotive assembly cells are cluttered environments, including robots, workpieces, and fixtures. Due to high volumes and several product variants assembled in the same cell, robot placement is crucial to increase flexibility and throughput. In this paper, we propose a novel method to optimize the base position of an industrial robot with the objective to reach all predefined tasks and minimize cycle time: robot inverse kinematics and collision avoidance are integrated together with a derivative-free optimization algorithm. This approach is successfully used to find feasible solutions on industrial test cases, showing up to 20% cycle time improvement.

Peer-review under responsibility of the organizing committee of the 6th CIRP Conference on Assembly Technologies and Systems (CATS)
Keywords: Robotics, optimization, cell layout, path planning

1. Introduction

Flexible assembly holds the promise of removing the need of highly dedicated and structured workspace, increasing productivity for more difficult components, as well as responding more quickly to product changes. Within flexible manufacturing systems, dynamic and robust layout are crucial and strategically important, since they are often done at early stages in the process, see [1].

In many areas, such as automotive, electronics manufacturing, and inspection, robots are used to perform specific operations on a workpiece in a station. Examples range from spot, stud, laser welding on sheet metal assemblies, to camera-laser-touch measuring on different objects. A complete set of operations consists in performing a specific task/operation, e.g. measuring or welding, by a robot on a set of work-points. Often, after the robot returns to its starting configuration, a new workpiece is introduced in the station and the new operations are performed. Since these cycles are repeated several times, it is very important that they are executed as fast as possible in order to maximize the throughput and to increase resource/equipment utilization.

Generally, a rule of thumb is used to determine the work flow for each robot workstation based on the overall production throughput requirement. Once a set of specific tasks is assigned to a robot, the layout engineer has limited freedom to optimize the robot workstation:

- robot's base placement (translation and rotation);
- robot's home configuration in the station (six joints);
- visiting order of the work-points;
- robot's paths with via-points.

The last three ones may be modified by changing the robot programs, whereas the first has to be completely decided before installing the robot in the workstation. The engineers use the robot working envelop to roughly place the robot base. If some portion of the tasks is out of robot's reach, a 1dof linear track could be used to extend the reach of a 6dofs (degrees of fredom) industrial robot. This typical layout practice only considers robot's basic reachability requirement. It is unknown to the layout engineers if there is any potential optimality in the robot base placement that could yield the best cycle time with the guaranteed reachability for a given set of tasks. Therefore, the optimization of the robot base placement w.r.t. the given set of tasks is of fundamental importance and, due to the recent advances in CAD/CAM software, [2], it is now possible to face problem of industrial relevance.

In this work, we describe a new approach and related algorithms to automatically calculate an optimal robot base location. This novel method is based on a derivative-free optimization algorithm and makes use of built-in functionalities in the software Industrial Path Solutions, [3], for the computations of robot reachability analysis and distances.

This paper is organized in the following way. First, related work is presented and the problem is described in more detail

Peer-review under responsibility of the organizing committee of the 6th CIRP Conference on Assembly Technologies and Systems (CATS)
doi:10.1016/j.procir.2016.02.105

and the tools used are presented. Then, a derivative-free model for the problem is presented together with a well known optimization algorithm; results are also shown. In Section 5 the method is generalized to deal with several workpieces. Eventually, cycle times for the optima found are generated and conclusions with ideas for future work are presented.

2. Related work

The most comprehensive works regarding cycle time optimization for a given set of tasks by moving the robot base are two early works from the 90s, see [4] and [5], and a more recent one, see [6]. The problem can be also seen from the workpiece perspective, see [7]. In [4] a grid in the state space of the robot base location is built at a given resolution. Afterwards, a generalized traveling salesman problem (GTSP) is solved in order to find the minimum cycle time for a robot visiting all workpoints and performing all tasks. This is done for each base location, corresponding to the points in the grid. The method, however, does not take into account collision detection in order to avoid geometrical obstacles. In [5], simulated annealing, see [8], is applied to cycle time optimization, both when moving the robot base location and when changing tasks sequence. The first solution is accomplished also by the help of reachability analysis and collision detection exploiting analytical expressions for fast computations of the so called 'obstacle shadows' and tasks reachability regions. The method is completed by also using clustering heuristics in order to deal with large sequencing problem instances. In [6] the relative position between the robot base and a path connecting fixed locations is optimized with respect to cycle time. Since the relative position between the robot base and the path is the interesting one, the path is translated and rotated. The results obtained show that cycle time can be improved by 37% with respect to the worst cycle time. More interesting figures concern the improvement with respect to paths generated by experienced engineers: these range between ca 3,5% to ca 21%. The main idea in [6] is to try to identify how cycle time varies with respect to the change of robot base by running a series of experiments that evaluate the real cycle time (for given robot base positions). Afterwards, cycle time for positions not covered in the experiments is approximated by the response surface method. The boundaries for the values of the robot base position are found by a bisection method. The function resulting from the response surface method is optimized with respect to robot base position, allowing it to vary within the boundaries found. A simulation is performed to check whether the path is kinetically feasible and to get an exact value for the cycle time. Small adjustments exploiting sensitivity analysis are applied if the original robot base position does not satisfy kinematic constraints. Limitations for this approach include lack of collision avoidance and no reordering of task locations. This is believed to be relevant when different robot bases give paths that heavily differ topologically.

Besides these works that consider the complete process, there are several articles dealing with subproblems whose solving algorithms could be included as blocks in a more complex method solving the overall problem. In [9], the authors deal with the optimization of the base location of a manipulator in an environment cluttered with obstacles. The problem is limited to single path optimization. The strength of the approach lies in a fast path re-optimization technique that can be applied to a collision-free path when changing the robot base position. The search for the best base is done through a neighborhood search in the state space. Robot base optimization is also treated in [10], where the TCP (Tool Center Point) path is fixed and the goal is to minimize the robot energy consumption. Another work involving robot placement for minimum time motion is [11]. A core block for the optimization of the robot base position, given a set of tasks, is the identification of robot bases from which specified task can be reached. A recent work dealing with fast algorithms solving this problem is [12].

3. Definition

The input for the problem is represented by:

- the robot model, including CAD geometries and its kinematic behavior,
- the CAD models representing fixture, welding gun and environment,
- a set of N_T tasks $T = \{T^1, \ldots, T^{N_T}\}$, *e.g.* spot welding points.

In the rest of the paper tasks and welding points will be used indifferently, as common practice for this kind of application. Finding the best (minimizing cycle time) positioning for the robot base requires repeated computations of:

1. reachability analysis;
2. collision test;
3. cycle time estimation.

A brute force analysis would, in practice, look like as in Algorithm 1.

Algorithm 1 Brute force computation of optimal robot base placement b giving the minimum cycle time c.

$c \leftarrow \infty$
$b \leftarrow \mathbb{O}$
for all b_i **do**
    c_B = ComputeCycleTime(b_i)
    if $c_B < c$ **then**
        $c \leftarrow c_B$
        $b \leftarrow b_i$
    end if
end for
return b, c

The robot base dofs consist of the (x, y, z) coordinates representing translation part and (R_X, R_Y, R_Z) representing the orientation part. The 'ComputeCycleTime' procedure requires heavy computations, that, in this work, rely on the simulation software platform IPS, see [3]. For more details about the software architecture and the implementation, please refer to Appendix 9. Brute force analysis, however, does not necessarily well scale, neither is the best approach, when

- the number of tasks increases;
- the CAD geometries get more complex;