Robotics and Computer-Integrated Manufacturing 29 (2013) 71–79

Contents lists available at SciVerse ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

An online robot trajectory planning and programming support system for industrial use

C. Kohrt [a,b,*], R. Stamp [a], A.G. Pipe [a], J. Kiely [a], G. Schiedermeier [b]

[a] Department of Engineering, Design and Mathematics, UWE—University of the West of England, BS16 1QY Bristol, UK
[b] Department of Computer Sciences, UASL—University of Applied Sciences Landshut, 84036 Landshut, Germany

ARTICLE INFO

Article history:
Received 26 September 2011
Received in revised form
3 June 2012
Accepted 26 July 2012
Available online 3 September 2012

Keywords:
Robot
Path
Planning
Support
Program
Generation

ABSTRACT

The manufacturing industry today is still looking for enhancement of their production. Programming of articulated production robots is a major area for improvement. Today, offline simulation modified by manual programming is widely used to reduce production downtimes but requires financial investments in terms of additional personnel and equipment costs. The requirements have been evaluated considering modern manufacturing aspects and a new online robot trajectory planning and programming support system is presented for industrial use. The proposed methodology is executed solely online, rendering offline simulation obsolete and thereby reduces costs. To enable this system, a new cell-based Voronoi generation algorithm, together with a trajectory planner, is introduced. The robot trajectories so achieved are comparable to manually programmed robot programs. The results for a Mitsubishi RV-2AJ five axis industrial robot are presented.

1. Introduction

Robot use and automation levels in the industrial sector will grow in future, driven by the ever-present need for lower item costs and enhanced productivity. Synonymous with this projected increase will be the requirement for capable programming and control technologies. Many industries employ offline programming within a manually controlled and specified work environment. This is especially true within the high-volume automotive industry, particularly when related to high-speed assembly and component handling. Any scenario, reliant on manual data input, based on real world obstructions, necessitates the complete production system being offline, out of production, for appreciable time-periods while data is input. A consequent financial loss ensues.

The two main categories of robotic programming methods are online programming and offline programming. Usually, the teach pendant is used to manually move the end-effector to the desired position and orientation at each stage of the robot task. Relevant robot configurations are recorded by the robot controller and a robot program is successively written to command the robot to move through the recorded end-effector postures. Offline programming is based on models of the complete robot work cell and the robot is simulated.

Published research appears to be concentrated on the application of simulation tools to generate discrete portions of the total robot trajectories [17], whilst necessitating manual input to link paths associated with one particular activity with those of another. Human input is needed also to correct inaccuracies and errors resulting from unknowns and falsehoods in the environment.

Investigations have been undertaken with the aim of generating a robot control program, by considering the working production environment as a single, whole, workspace. Use is made of automated workspace analysis techniques and trajectory smoothing. Some non-productive time is necessitated, but unlike previously reported approaches, this is, for the most part, achieved automatically and consequently rapidly. As such, the actual cell-learning time is minimal.

2. Industry requirements to an online robot trajectory planning and programming support system

An up-to-date industry requires a modern production system, able to combine and support flexibility, high-speed and optimization [11]: the overall production time available must be maximized to guarantee the highest productivity possible. Considering the high level of complexity of several robot-programming tasks for human operators, the proposed solution consists in a support system that takes over all the most complicated sub-tasks. The left manageable

* Corresponding author at: University of the West of England, Department of Computer Sciences, Faculty of Computing, Engineering & Mathematical Sciences, Coldharbour Lane, Bristol BS16 1QY, UK

E-mail addresses: christian@kohrt.org (C. Kohrt), richard.stamp@uwe.ac.uk (R. Stamp), anthony.pipe@uwe.ac.uk (A.G. Pipe), janice.kiely@uwe.ac.uk (J. Kiely), gschied@fh-landshut.de (G. Schiedermeier).

0736-5845/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.rcim.2012.07.010

sub-tasks related to the given mission remain responsibility of the operator. The proposed methodology is executed solely online rendering offline simulation obsolete and thereby reduces costs for the offline preparation of robot programs. Supported online programming must be fast and flexible to reduce possible production downtimes. The generated trajectories must conform to the given requirements in quality and speed. Physical production parts and fixtures are often not available during online robot programming, thus, the support system must handle such situations to permit its use. Nevertheless, the human component still remains important and necessary: robot programs may be modified by human operators during their lifecycle because of possible changes. The so generated programs must be readable, maintainable and changeable.

Fig. 1. Devices overview.

3. Support system overview and architecture

The proposed support system is applied on a 5-axis industrial-scale, articulated Mitsubishi RV-2AJ robot with an additional Ethernet card installed. It is a nonlinear system with five rotary joints. The robot is equipped with the Mitsubishi CR1 controller and a teach pendant. The main areas of the robot are assembly, manufacture, pick & place and handling tasks. Communication between this system and a personal computer is possible [12]; the commercial viability has already been demonstrated [16]. The equipment is shown in Fig. 1. Both model-based and sensor-based data are considered in order to define the environment of the robot: perception functions, initiated by sensors (cameras or tactile sensors), provide the system with information about the environment. A general overview is given in Figs. 2 and 3.

The *Object Recognition* component converts the features of an image into a model of known objects. First, the scene is segmented into distinct objects. An analysis deriving from motion, binocular stereopsis, texture, shading and contour follows, so that orientation, shape and position of each object may be determined relative to the camera. Peter Corke's Machine Vision Toolbox for Matlab [4,5] allows the developer to use professional image processing capabilities [20] with ease. In addition, model-based data such as computer-aided design (CAD) data is used to present the world model more accurately. Computer-aided design derived data from simulation systems, such as RobCAD [24], are exported as drawing exchange format (DXF) files, including all locations attached and they are stored within the world model. The attached locations of computer-aided design objects are employed to acquire information concerning start, target and application paths locations.

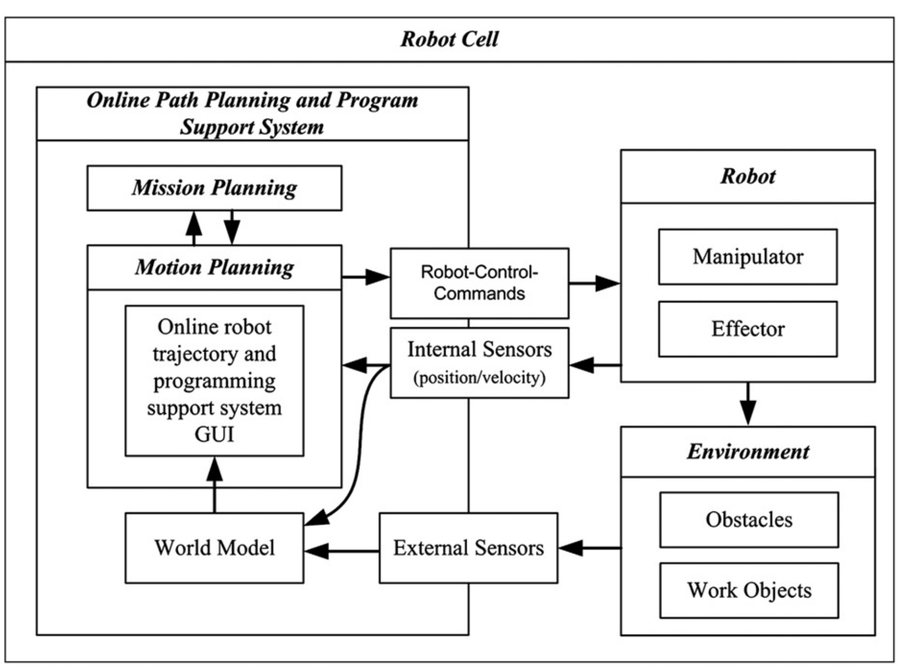

Fig. 2. Logical view of the support system.