The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

A Replanning Strategy for Preplanned Robot Trajectory in Emergency Situations

Sung Wook Yun, Moon Soo Chang, Byung Hoon Kang and PooGyeon Park*

Electrical and Computer Engineering Division, POSTECH, Pohang, Korea
(Tel: 82-54-279-5588; Fax: 82-54-279-2903)
(wook2000, gravi, anbabo, ppg@postech.ac.kr)

Abstract: The proposed replanning strategy in PC-RCA is designed for the parametric interpolated curvature-bounded path with a desired jerk-limited speed profile. Conventional methods stop abruptly in emergency situations that often cause breakdown. Besides, robots cannot get back to the jobs smoothly because these methods do not replan the speed profile for resuming. However, in the proposed algorithms, the back motion is designed to replan a jerk-limited speed profile with consideration of passed trajectory. And resuming motion executes on preplanned trajectory with newly planned speed profile.

Keywords: Back motion, emergency situation, jerk-limited speed profile, replanning strategy.

I. INTRODUCTION

In industrial environment, robots are sometimes faced with dangerous occasions. To deal with these situations, an emergency stop is employed widely in general. In order to get back in the jobs, the emergency stop requires replanning the trajectory of robots. This paper presents trajectory replanning strategy for industrial robots in emergency situations. The proposed replanning strategy is based on our own trajectory planning algorithm that is designed for PC-RCA (PC based Robot Controller Architecture.)

The PC-RCA is our new robot control architecture for multitasking process based on software. And it comes with a new robot-programming language which has ability to use rich resources of PC, user-oriented aspect and no dependency with other robot manipulators. The PC-RCA uses two major algorithms for trajectory planning: two staged parametric interpolation [1] and curvature-bounded smooth transition algorithm with a desired jerk-limited speed profile [2-3]. To inherit strong points of the architecture, proposed replanning strategy should be designed on the base of these two trajectory planning algorithms.

In general, many robot systems with conventional methods stop abruptly in emergency situations that often cause breakdown and cannot get back to the jobs smoothly. The proposed algorithm adopts back and resuming motion to restart job after emergency stop. Moreover, these two motions do not come alone. In emergency stopped state, users can choose back and resuming motions in a stopped motion. Such duplicated back and resuming motions make the replanning strategy more complicate. Therefore, proposed algorithm runs on two separated modes: emergency and normal modes.

The paper is organized as follows: Section II explains preliminaries of our robot system. Section III develops the proposed replanning strategy with its constraints. In section IV, we shall validate proposed algorithm with SCARA robot using a simulator and obtaining real data from PC-RCA. The last section summarizes and concludes the paper.

II. Robot System Description

Recently, various robot control system architectures (OSACA [4], PC-ORC [5], NEXUS [6], and etc [7-9]) have presented effective structure of PC-based controller. PC based robot controllers are good for the extensibility. And its computing power is improved dramatically. It is in this context that we developed new PC-based robot control architecture with real-time aspect and multi robot control.

The Fig.1 shows a brief structure of PC-RCA. There are the VRCM (Virtual Robot Control Machine) for high level system, RUSH (Robot User Shell) for external communication, trajectory planner, motion control part and actuator driving module for low level system.

The most characteristic feature of PC-RCA is using the VRCM. It is virtual robot control machine, implemented by software program on RT-OS. A VRCM takes single robot and this looks like a conventional controller in the PC. This VRCM is able to be operated multiply but independently in the range of surplus computing power because it is software program. This feature gives multi-robot control ability to PC-RCA and CEU (core execution unit) makes easy development using API of OS or software libraries.

RUSH supplies a unified communication interface through the abstract link layer that has no correlation with physical layer. Trajectory planner, a part of low level system, generates the reference point that is required by motion called VRCM. This reference is passed to the motion control module that generates current reference for actuator driving module.

The proposed replanning strategy is working with the trajectory planner, but it is independent of other structures of PC-RCA. And our strategy is implemented as a module of the trajectory planner.

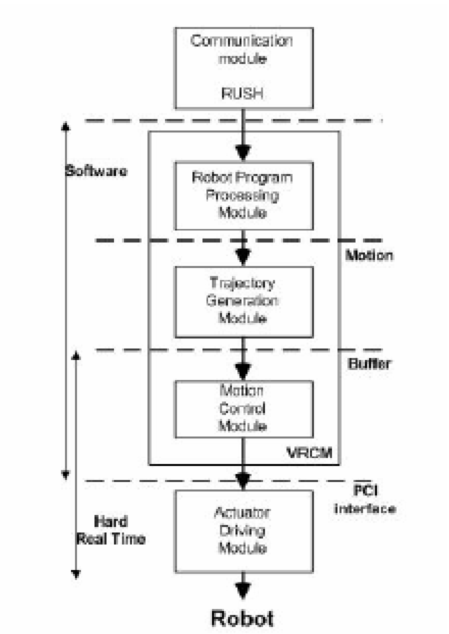

Fig.1. The overall structure of PC-RCA

III. Replanning Strategy

The proposed algorithm was implemented on the PC-RCA as a part of its module for replanning trajectory in emergency situations.

1. The design of state machine

The proposed replanning strategy is designed by the state machine. This state machine can be divided into two modes: the normal and emergency modes. In the normal mode, preplanned motions are executed with stop and resuming motions. In the emergency mode, back motion can be executed with stop and resuming motions in a stopped state. These motions do not come alone. The user can execute emergency stop again during the back and resuming motions. Such duplicated back and resuming motions make the replanning strategy more complicate. These concepts are showed in Fig.2.

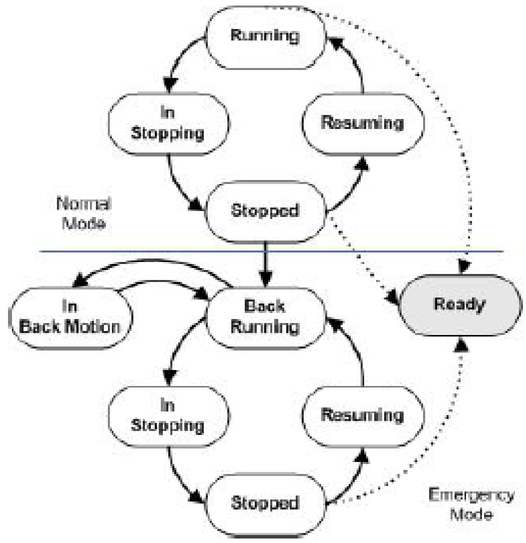

Fig.2. The state diagram

2. Constraints of PC-RCA

As mentioned above, the PC-RCA uses two major algorithms for trajectory planning: two staged parametric interpolation and curvature-bounded smooth transition. The two staged parametric interpolation method consists of two stages. In the preprocessing stage, a curve is segmented and estimated from curvature and length of small pieces, then stored in the table. In the interpolation stage, the parameter is calculated using a quintic polynomial obtained from the tabulated parameter and length data. The curvature-bounded smooth transition algorithm uses a new path-level transition method, which generates a curvature-bounded path with a desired jerk-limited speed profile.

Dealing with these constraints of PC-RCA, stop motion replaces the old speed profile with the newly generated jerk-limited stop speed profile maintaining the preplanned trajectories. And, resuming motion is executed by regenerating the jerk-limited speed profile for remained trajectories. Moreover, back motion is designed to replan a jerk-limited speed profile with consideration of passed trajectory.

The motions of industrial robots can be divided into joint motion, which is coordinated by angle of joint, and the others which is coordinated by the Cartesian coordination. Because the former is only controlled by angle, replanning algorithm considers speed profile only. This makes replanned trajectory different from preplanned. So we implement dual action with every single state.

IV. Simulation Results

In this section, we validate the proposed algorithm by generating the reference data from PC-RCA. Fig.3. shows the robot trajectories: joint motion (red), line motion (green), line motion (blue), arc motion (magenta), joint motion (cyan) and bezier motion (yellow). In first line motion (green), stop and resuming motions are executed. In the bezier motion (yellow), stop and back motions are executed. The angle of each actuator is shown in Fig.4. And, the speed profile is shown in Fig.5. As mentioned above, stop motion replaces the speed profile with the newly generated speed profile. And, resuming motion is executed by regenerating the speed profile. Moreover, back motion is designed to replan a jerk-limited speed profile with consideration of passed trajectory. The acceleration profile is shown in Fig.6.

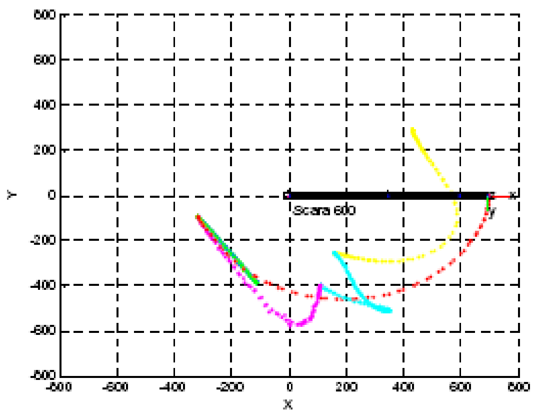

Fig.3. Robot trajectories

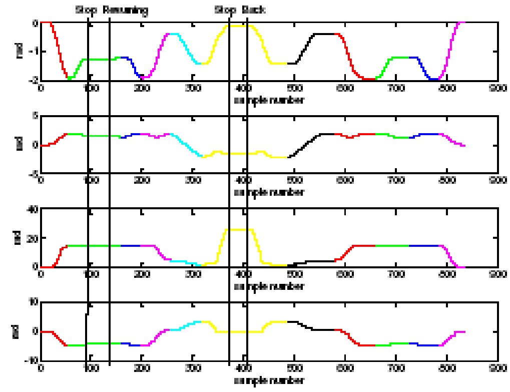

Fig.4. Angle of each actuator

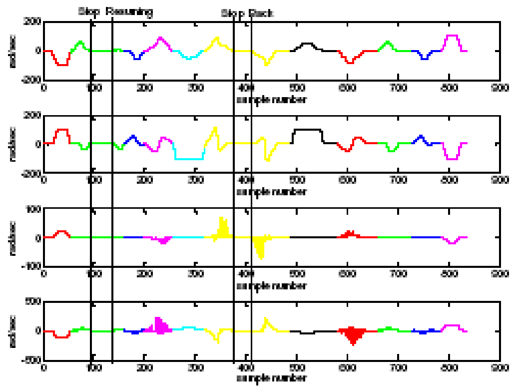

Fig.5. Angular speed profile

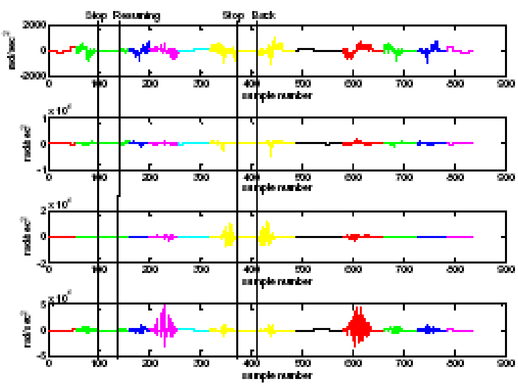

Fig.6. Acceleration of each actuator

V. Conclusion

In this paper, we present the trajectory replanning strategy for industrial robots in emergency situations. The proposed algorithm adopts back and resuming motion to restart job after emergency stop. The emergency stop motion replaces the old speed profile with the newly generated jerk-limited stop speed profile maintaining the preplanned trajectories. And, resuming motion is executed by regenerating the jerk-limited speed profile for remained trajectories. Moreover, back motion is designed to replan a jerk-limited speed profile with consideration of passed trajectory. The proposed algorithm is validated by using a simulator and obtainning real data from PC-RCA with SCARA robot.

Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement) (IITA-2007-C1090-0701-0037)

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement) (IITA-2007-C1090-0701-0004)

REFERENCES

[1] S. Y. Jeong, Y. J. Choi, P. Park (2006), Parametric interpolation using sampled data, CAD, Vol. 38, pp. 39–47.

[2] S. Macfarlane, E. A. Croft (2003), Jerk-bounded manipulator trajectory planning: design for real-time applications, Robotics and Automation, IEEE Trans. on Vol. 19, pp. 42–52.

[3] S. Nam, M. Yang (2004), A study on a generalized parametric interpolator with real-time jerk-limited acceleration, CAD, Vol. 36, pp. 27–36.

[4] OSACA, Open System Architecture for Controls within Automation Systems, ESPRIT III Project 6379/9115, 1996.

[5] K. Hong, J. Kim, C. Huh, et al (2001), A PC-based open robot control system: PC-ORC, Industrial Electronics, Proc. IEEE Int'l Symposium on, Vol.3, pp. 1901–1906

[6] J.A. Fernandez, J. Gonzalez (1998), NEXUS: a flexible, efficient and robust framework for integrating software components of a robotic system, Robotics and Automation, Proc. IEEE Int'l Conf. on Vol. 1, pp. 524–529.

[7] Hiroshi Ishiguro, Toshiyuki Kanda, Katumi Kimoto et al (1999), A Robot Architecture Based on Situated Modules, Proc. IEEE/RSJ Int. Conf. on Vol. 3, pp. 1617–1624.

[8] X. Zhou, W. Li, Q. Li (1997), A new robot control system with open architecture, Advanced Robotics, Proc. Int'l Conf. on, pp. 813–818.

[9] Y. Lu, Q. Huang, L. Zhang, et al (2006), Software Architecture for a Humanoid Robot Teleoperation Based on RT-Linux/Linux/Windows Operating System, Mechatronics and Automation, Proc. IEEE Int'l Conf. on pp. 324–328.