

< 기술논문 >

PLP 를 위한 Fast Algorithm 과 팔레타이징 작업 제어 S/W 를 위한 로봇 시뮬레이터 개발

임성진* · 유승남† · 강맹규** · 한창수*** · 송영훈**** · 김성락**** · 한정수*****
(2006년 6월 21일 접수, 2007년 4월 23일 심사완료)

Development of Robot Simulator for Palletizing Operation Management S/W and Fast Algorithm for 'PLP'

Sung-jin Lim, Seung-nam Yu, Maing-kyu Kang, Chang-soo Han, Young-hoon Song, Sung-rak Kim

Key Words : PLP(Pallet Loading Problem, 팔렛 적재 문제), Fast Algorithm(패스트 알고리즘), Robot Trajectory Optimization(로봇 경로 최적화), Auto-palletizing system(오토 팔레타이징 시스템), Robot Palletizing Simulator(로봇 팔레타이징 시뮬레이터)

Abstract

Palletizing is necessary to promote the efficiency of storage and shipping tasks. These are, however some of the most monotonous, heavy and laborious tasks in the factory. Therefore many types of robot palletizing systems have been developed, but many robot motion commands still depend on the teaching pendent. That is, an operator inputs the motion command lines one by one. It is very troublesome, and most of all, the user must know how to type the code. That is why we propose a new GUI (Graphic User Interface) Palletizing System. To cope with this issue, we proposed a 'PLP' (Pallet Loading Problem) algorithm, Fast Algorithm and realize 3D auto-patterning visualization interface. Finally, we propose the robot palletizing simulator. Internally, the schematic of this simulator is as follows. First, an user inputs the physical information of object. Second, simulator calculates the optimal pattern for the object and visualizes the result. Finally, the calculated position data of object is passed to the robot simulator. To develop the robot simulator, we use an articulated robot, and analyze the kinematics and dynamics.

Especially, All problem including thousands of boxes were completely calculated in less than 1 second and resulted in optimal solutions by the Fast Algorithm.

1. 서 론

일반적으로 팔레타이징(Palletizing)이란 공장의 생산라인에서 생산된 최종 생산품을 수송, 적재 및 보관의 용이성을 위해 보통 나무나 플라스틱

등으로 만들어진 사각형 팔렛(pallet) 위에 적재하는 과정을 의미한다.⁽¹⁾ 하지만 팔레타이징 작업은 대표적인 단순, 반복, 고 부하 작업으로서 인력을 활용하는데 어려움이 있으며, 더욱이 인력에 의한 적재는 작업품질이 일정치 않아 대량생산으로 빠르게 생산되어 나오는 생산물을 일정한 시간 내에 처리해야 하는 환경에서는 적합하지 않다.⁽²⁾ 따라서 팔레타이징 작업을 자동화하기 위한 많은 연구가 이루어져 왔으며, 현재 로봇을 이용한 다양한 형태의 오토 팔레타이징 시스템(Auto-palletizing system)이 산업현장에 도입되어 있는 상태이다. 이러한 자동화 시스템을 구현하기 위해서는 산업용 로봇과 그 제어가 기본적으로

† 책임저자, 회원, 한양대학교 대학원 기계공학과

E-mail : brass_79@hotmail.com

TEL : (031)400-4062 FAX : (031)406-6398

* 한양대학교 대학원 기계공학과

** 한양대학교 산업공학과

*** 회원, 한양대학교 기계정보경영공학부

**** 현대중공업(주) 기계전기연구소

***** 한성대학교 기계시스템공학과

필요하며, 여기에 작업을 시뮬레이션(Simulation) 하고 데이터 베이스(Database)화하기 위한 소프트웨어가 사용된다. 이는 공정의 특성상, 팔레타이징 시스템을 제어하기 위해서는 단순한 로봇만의 제어가 아닌 대상물이 이송되는 컨베이어 시스템이나 팔렛, 간지(Slip sheet)와 같은 팔레타이징 구성물들을 동시에 고려되어야 하고, 적재 시뮬레이션을 통해 작업결과를 확인할 수 있는 통합시스템이 필요하기 때문이다.

여기에는 대상물의 팔레타이징 시스템으로의 유입 속도나 로봇의 팔레타이징 작업 속도, 적재가 완료된 팔렛의 반출 시간을 고려한 버퍼(buffer) 팔렛, 대상물을 최대한 많이 적재하기 위한 적재패턴의 생성, 대상물이 적재됨에 따라 변화되는 충돌환경을 고려한 로봇의 적재경로 생성 등 여러 가지 고려사항이 존재한다.

일반적인 로봇 시스템의 제어에 관한 문제를 제외한 팔레타이징에 관련된 기존의 연구는 주로 PLP(Pallet Loading Problem)에 관련하여 연구가 진행되어 왔다.⁽³⁻⁵⁾ PLP란, 한정된 영역의 사각형 팔렛 위에 최대한 많은 사각형의 대상물을 적재할 수 있는 패턴을 빠르게 생성하는 것에 관한 문제이다. 그러나 기존의 연구는 대상물이 많아질수록 생성시간이 급격히 증가하는 한계가 있고, 상용 팔레타이징 시스템은 팔레타이징 적재 패턴을 환경에 맞게 새로 생성하기보다는 기존의 저장된 패턴을 단순히 선택하여 사용하는 사례가 많았다.

본 논문은 이와 같은 로봇을 이용한 팔레타이징 시뮬레이터를 개발하기 위해 PLP 알고리즘인 Fast Algorithm 과 적재물의 3 차원 위치를 출력하고 수정할 수 있는 인터페이스 및, 산업용 로봇의 경로계획 알고리즘을 평가하고 팔레타이징 작업을 시뮬레이션 하기 위해 개발된 3 차원 로봇 시뮬레이터 (3D Robot Simulator)의 개발에 대한 내용을 포함하고 있다.

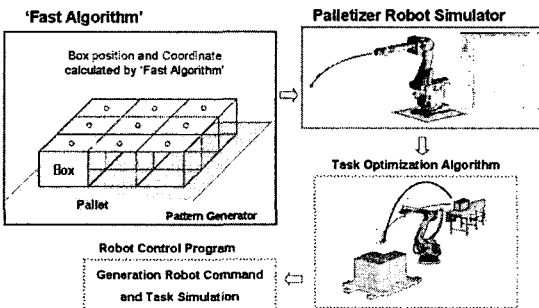


Fig. 1 Schematic of palletizing simulator

Fig. 1 은 본 논문에서 제안된 시뮬레이터의 시스템 구조를 도시하고 있다. 즉, Fast Algorithm 을 이용하여 각 박스의 위치와 방향 정보를 로봇 시뮬레이터에 넘겨 주면 로봇 시뮬레이터에서 각 박스의 위치 값에 따라 박스를 적재하는 시뮬레이션을 실행하게 된다.

향후에는, 적재 경로 최적화 알고리즘을 개발하여 작업 시간의 최적화를 이루고 실제 로봇과 연동해서 작업을 제어하게 되는 단계를 제시할 것이다.

2. 팔레타이징 패턴 생성 S/W 개발

팔레타이징 패턴은 직사각형 팔렛 위에 적재되는 대상물의 적재위치와 방향, 적재 순서로 구성된다. 팔렛의 정보와 대상물의 정보가 주어졌을 때 최대한 많은 대상물을 적재할 수 있어야 하며 대상물이 높이 적재 될수록 대상물의 물리적인 안정성도 고려되어야 한다. 또한 팔레타이저 로봇(Palletizer robot)의 위치정보로부터 팔렛 위에 먼저 적재된 대상물이 다음 대상물을 적재하는 적재경로에 충돌 체로 존재하지 않도록 적절한 적재 순서가 지정되어야 한다. 이를 구현하는데 있어 기존의 패턴 생성 알고리즘은 실시간으로 패턴을 생성하는데 다소 무리가 있다. 따라서 본 연구에서는 개선된 패턴 생성 알고리즘인 Fast Algorithm 을 이용하여 팔레타이징 패턴을 생성하는 S/W 를 개발 하였다.

2.1 Steudel's Heuristic Algorithm

본 연구에서 개발한 Fast Algorithm 은 기본적으로 Steudel 이 제안한 기존의 '4-block pattern heuristic algorithm'을 개선한 것이다.⁽⁶⁾ Steudel 의 heuristic algorithm 의 기본 패턴은 Fig. 2 와 같다.

Steudel 의 algorithm 은 각각의 block 에 box 를 적재할 때 같은 방향으로만 적재하는 방법으로서

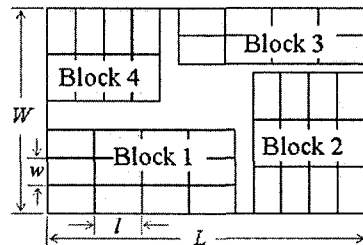


Fig. 2 4-block pattern

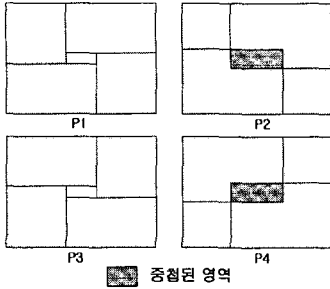


Fig. 3 Four patterns of the initial solution

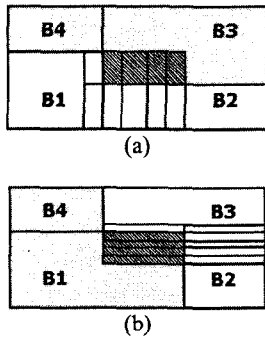


Fig. 4 Treatment of Steudel's Algorithm [(a) Treatment 1, (b) Treatment 2]

1, 3 block 은 가로방향으로 box 를 적재하고 2, 4 block 은 세로방향으로 box 를 적재한다. Steudel 의 algorithm 은 2 단계로 구성되어 있으며, 첫 번째 단계에서는 Fig. 3 와 같이 4 가지 패턴의 초기 해를 생성한다.

이 과정에서 P2와 P4와 같이 불가능한 패턴이 발생하게 되는데, 두 번째 단계에서 초기 해를 개선하는 과정을 거치게 된다. 즉, Fig. 4 와 같이, (Treatment 1) Block 3,4 를 고정시키고, Block 1,2 의 길이를 변경하여 해를 개선하는 방법과, (Treatment 2) Block 1,2 를 고정시키고, Block 3,4 의 길이를 변경하여 해를 개선하는 방법을 비교하여 더 좋은 해를 선택할 수 있다.

2.2 The Fast Algorithm

Fast Algorithm 은 Steudel 의 algorithm 와 같이 4 가지 패턴의 초기 해를 생성하되, Steudel 의 것처럼 Block 의 모든 크기를 고려하지 않아 Algorithm 수행 속도를 빠르게 하였다(Fig. 5).

\bar{a} : block 의 길이를 최대화하면서 box 를 길이 방향으로 배열했을 때 최대로 적재 가능한 box 수 = $5l$

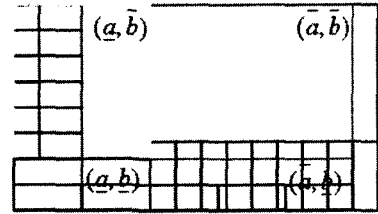


Fig. 5 Treatment 3 of the fast algorithm

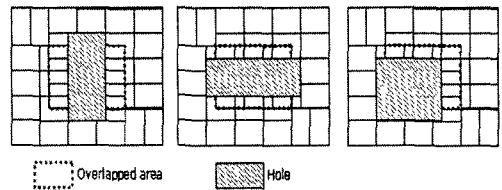


Fig. 6 Parameter of fast algorithm

a : block 의 길이를 최대화하면서 box 를 길이 방향으로 배열했을 때 최소로 적재 가능한 box 수 = $2l$

\bar{b} : block 의 너비를 최대화하면서 box 를 너비 방향으로 배열했을 때 최대로 적재 가능한 box 수 = $8w$

b : block 의 너비를 최소화하면서 box 를 너비 방향으로 배열했을 때 최소로 적재 가능한 box 수 = $2w$

Fast Algorithm 의 첫 번째 단계에서는 block1 의 크기를 (\bar{a}, \bar{b}) 와 (\bar{a}, b) , (a, \bar{b}) , (a, b) 의 4 개의 조합만을 고려하여 계산한다. 두 번째 단계에서는 초기 해를 개선하게 되는데, Steudel's algorithm 의 Treatment 1 과 2 에 새롭게 Treatment 3 추가한다. Treatment 3 는 Central hall 을 만들어 Overlapped area 를 제거하고 다시 그 Central hall 에 반복적으로 패턴을 생성하는 것이다(Fig. 6).

Central hall 을 생성하는 방법은 3 가지 방법이 있는데, 중첩영역(Overlapped Area)을 두 개의 가로 면으로 분리하는 방법과 중첩영역을 두 개의 세로 면으로 분리하는 방법, 중첩영역의 왼쪽 면과 아래 면으로 분리하는 방법 등이 그것이다. 실제 문제를 계산할 때에는 위의 3 가지 방법 중 가장 좋은 해를 선택하여 사용할 수 있다.

Fast Algorithm 의 장점은 빠른 계산 시간에 있으며 K6-350MHz 컴퓨터를 이용하여 계산시간을 검증한 결과 1000 개에서 5000 개 정도의 박스가 들어가는 패턴을 대부분 1 초안에 계산해내는 성능을 보인 바 있다.⁽⁷⁾

2.3 팔레타이징 패턴 생성 소프트웨어

팔레타이징 패턴 생성 소프트웨어는 팔레타이

징 작업을 자동화하기 위한 첫 단계로서 팔렛 정보, 박스 정보, 로봇의 배치정보를 입력 받아 Fast Algorithm 을 이용하여 3 차원 상의 팔렛 패턴을 생성하고 이 패턴을 팔렛 및 로봇의 좌표를 기준으로 하여 박스의 위치 및 적재 순서를 출력하는 소프트웨어이다. 이 소프트웨어는 Visual C++ 6.0®을 이용하여 개발 되었으며 OpenGL®을 이용하여 생성된 패턴을 3 차원으로 확인 할 수 있도록 하였다. 이 프로그램을 크게 다섯 부분으로 나누어 설명하면 다음과 같다(Fig. 7).

- ① 팔렛의 크기와 허용적재 중량, 적재할 박스의 크기, 중량, 라벨방향에 대한 정보를 입력하고 팔렛에 대한 로봇의 방향을 입력한다. 중량에 대한 정보로 박스의 적재 가능한 최대 개수만큼 적재 층이 계산되고 로봇의 방향 정보는 기존의 적재된 박스가 다음 박스의 적재 시에 방해 받지 않게 박스의 적재 순서를 로봇에서 가장 바깥쪽부터 적재 되도록 적재순서를 생성 할 수 있게 한다.
- ② 앞의 주어진 정보로부터 Fast Algorithm 으로 생성된 패턴을 적용하는 단계로서 최대의 박스 개수를 갖는 가장 최적화된 패턴부터 최대개수에서 5 개 적은 패턴까지 리스트로 출력되며 각 홀수 층, 짝수 층, 최상단층에 대하여 각기 다른 패턴을 적용 할 수 있다. 적용된 패턴은 3 차원 화면으로 확인할 수 있으며, 팔렛에 적재 될 전체 박스 개수, 총 중량, 전체 높이 등의 정보를 확인 할 수 있다.

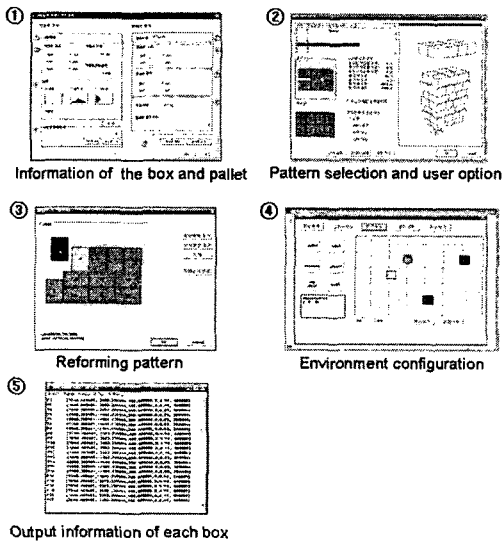


Fig. 7 Palletizing pattern generation software

③ Fast Algorithm 으로 생성된 패턴이 아닌 사용자 임의로 패턴을 수정하여 사용할 수도 있으며 박스의 개수, 위치, 방향, 적재순서 정보를 수정 가능하다. 또한 팔렛을 벗어나거나 박스끼리 겹치는 등의 잘못된 패턴에 대해서 붉은색으로 표시 하여 경고 한다.

④ 팔렛의 패턴이 결정되면 팔렛, 컨베이어 (Conveyor), 간지(SlipSheet), 로봇의 배치를 결정하게 된다.

⑤ 각 요소의 배치가 결정되면 최종적으로 팔렛 및 로봇을 기준으로 하여 전체 박스의 적재 순서, 3 차원 위치, 방향, 그리퍼(gripper) 정보 등의 데이터를 파일로 출력하게 된다.

3. 3 차원 로봇시뮬레이터 개발

3 차원 로봇 시뮬레이터는 팔레타이징 작업 중 로봇이 박스를 팔렛에 적재 할 때 중간의 충돌체를 회피하면서 최적의 경로를 생성하는 알고리즘을 개발하기 위해, 알고리즘이 생성한 이동 경로를 평가하고 전체 팔레타이징 작업을 시뮬레이션 하기 위해 개발 되었다.

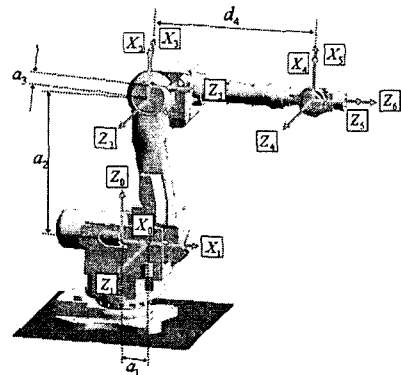


Fig. 8 Robot cartesian coordinates

Table 1 D-H parameter table

i	θ_i	d_i	a_i	α_i
1	θ_1	0	a_1	90°
2	θ_2	0	a_2	0°
3	θ_3	0	a_3	90°
4	θ_4	d_4	0	90°
5	θ_5	0	0	-90°
6	θ_6	0	0	0°

3.1 역기구학

시뮬레이터에 사용된 로봇은 산업용 6 축 수직 다관절형 로봇이며 이 로봇의 각 관절 별 좌표계는 Fig. 8 과 같으며 D-H(Denavit-Hartenberg) 파라미터는 Table 1 과 같다. 여기서 θ_i 는 관절변수이며 d_i 는 Z_i 축을 따른 거리, a_i 는 X_i 축을 따른 거리, α_i 는 X_i 축을 따라 회전된 각도이다.

역기구학 방정식을 구하기 위해서 먼저 목표로 하는 핸드 좌표계로부터 로봇의 손목 좌표계를 얻는다.

$${}^R T_W = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

손목 좌표계는 모든 관절의 변환행렬을 곱한 최종 좌표계이므로 D-H 파라미터로부터 각 관절의 변환행렬을 구한 후에 각 변환행렬을 모두 곱한다.

$${}^R T_1 {}^1 T_2 {}^2 T_3 {}^3 T_4 {}^4 T_5 {}^5 T_6 = {}^R T_W \quad (2)$$

얻어진 행렬은 로봇의 베이스로부터 손목까지의 변환을 하는 행렬이고 이 행렬과 손목 좌표계와 비교하고 첫 번째 관절의 역변환행렬부터 곱해가면서 각 관절각도를 결정할 수 있는 식을 구할 수 있다. 구해진 식은 다음과 같다.

$$\begin{aligned} \theta_1 &= \tan^{-1}\left(\frac{P_y}{P_x}\right) \text{ or } \theta_1 = \theta_1 + 180^\circ \\ \theta_2 &= \pm \cos^{-1}\left(\frac{A^2 + P_z^2 - d_4^2 - a_3^2 + a_2^2}{2a_1\sqrt{P_z^2 + A^2}}\right) + \alpha \\ \theta_3 &= \pm \cos^{-1}\left(\frac{A^2 + P_z^2 - d_4^2 - a_3^2 - a_2^2}{2a_2\sqrt{d_4^2 + a_3^2}}\right) + \alpha \\ \theta_4 &= \tan^{-1}\left(\frac{s_1 a_x - c_1 a_y}{c_{23}(c_1 a_x + s_1 a_y) + s_{23} + a_z}\right) \text{ or } \theta_4 = \theta_4 + 180^\circ \\ \theta_5 &= \tan^{-1}\left(\frac{-c_4(c_{23}(c_1 a_x + s_1 a_y) + s_{23} a_z) - s_4(s_1 a_x - c_1 a_y)}{s_{23}(c_1 a_x + s_1 a_y) - c_{23} a_z}\right) \\ \theta_6 &= \tan^{-1}\left(\frac{-s_4(c_{23}(c_1 n_x + s_1 n_y) + s_{23} n_z) + c_4(s_1 n_x - c_1 n_y)}{-s_4(c_{23}(c_1 o_x + s_1 o_y) + s_{23} o_z) + c_4(s_1 o_x - c_1 o_y)}\right) \end{aligned} \quad (3)$$

3.2 궤적 계획법

로봇의 움직임을 정의하기 위해서는 먼저 초기

위치와 끝 위치를 정하고 각 위치에 대하여 역기구학으로 각 관절각도를 계산한 다음 초기 위치에서 끝 위치로 이동하는 경로를 관절들이 어떤 각 가속도와 각속도를 가지며 이동하는 것에 대한 정의가 필요하다.

이를 위해 본 연구에서는 5 차 다항식 궤적 계획법을 이용하였다. 5 차 다항식 법은 초기와 끝에서의 각도와 각속도, 각가속도를 정해주면 시간의 변수를 갖는 식 (4)에 대입하여 연립으로 풀어 6 개의 계수 값을 결정하게 된다. 계수 값들이 결정되면 관절의 시간에 따른 각가속도와 각속도, 각도 값을 계산할 수 있다.

$$\begin{aligned} \theta(t) &= c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5 \\ \dot{\theta}(t) &= c_1 + 2c_2 t + 3c_3 t^2 + 4c_4 t^3 + 5c_5 t^4 \\ \ddot{\theta}(t) &= 2c_2 + 6c_3 t + 12c_4 t^2 + 20c_5 t^3 \end{aligned} \quad (4)$$

5 차 다항식 법은 관절이 움직일 때 경로 점에서의 각가속도와 각속도를 정의 할 수 있어 로봇이 경로점을 통과할 때 매끈하게 움직일 수 있는 장점이 있다.

연구에서 사용된 궤적계획법은 적절하게 경로 점이 주어졌을 때 경로점 사이에서 로봇의 움직임을 정의해주는 것으로 주위환경에 충돌체가 존재하는 경우에는 충돌체를 회피할 수 있는 적절한 경로점의 설정이 필요하다. 그러나 충돌체 회피를 위한 경로생성 알고리즘은 본 연구의 단계에서는 적용되어 있지 않으며 향후 연구단계에서 추가할 예정이다.

3.3 역동역학

궤적 계획법으로 관절의 각도, 각속도, 각가속도 값을 알게 되면 관절에 필요한 모터 토크 값을 계산 할 수 있다. 그러므로 로봇의 경로계획이 모터의 허용 토크 안에 있는지 평가 할 수 있으며 모터의 허용토크 안에서 가장 빠르게 작업할 수 있는 경로계획을 생성할 수 있게 된다.

모터 토크의 계산은 라그랑지안(Lagrangian)을 이용하였다. 라그랑지안 L 은 운동에너지 K 와 위치에너지 P 로부터 $L = K - P$ 로 계산된다.

운동에너지는 식 (5)와 같다.

$$\begin{aligned} K_1 &= \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i \text{Trace}(U_{ip} J_i U_{ir}^T) \dot{q}_p \dot{q}_r \\ &+ \frac{1}{2} \sum_{i=1}^n I_{i(\alpha)} \dot{q}_i^2 \end{aligned} \quad (5)$$

여기서 U 는 각 관절의 미소변위에 따른 각속도에 관련된 행렬이며 J 는 로봇 링크의 가상 관성 행렬(pseudo inertia matrix)이고 I 는 모터 구동기의 관성이다. 위치에너지는 식 (6)과 같으며 m 은 링크질량, g 는 중력벡터, \bar{r}_i 는 링크 프레임에서의 질량중심벡터이다.

$$P = \sum_{i=1}^n [-m_i g^T \cdot ({}^0T_i \bar{r}_i)] \quad (6)$$

운동방정식은 라그랑지안을 미분하여 다음과 같이 도출할 수 있다.

$$\begin{aligned} T_i &= \frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} \\ &= \sum_{j=1}^6 D_{ij} \ddot{q}_j + I_{i(act)} \ddot{q}_i + \sum_{j=1}^6 \sum_{k=1}^6 D_{ijk} \dot{q}_j \dot{q}_k + D_i \\ D_{ij} &= \sum_{p=\max(i,j)}^6 \text{Trace}(U_{pj} J_p U_p^T) \\ D_{ijk} &= \sum_{p=\max(i,j,k)}^6 \text{Trace}(U_{pk} J_p U_p^T) \\ D_i &= \sum_{p=1}^6 -m_p g^T U_{pi} \bar{r}_p \end{aligned} \quad (7)$$

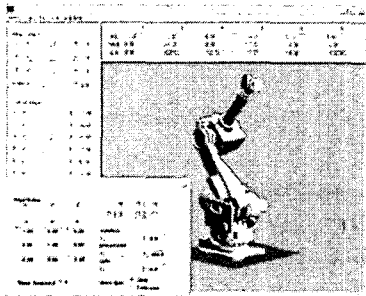


Fig. 9 3D Robot simulator

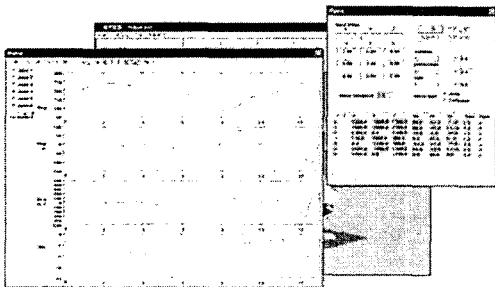


Fig. 10 Data display interface of robot simulator

식 (7) 에서, D_{ij} 의 항은 각가속도 관성의 항, $I_{i(act)}$ 의 항은 구동기 관성의 항, D_{ijk} 의 항은 Coriolis 와 원심력의 항이며, 마지막 항은 중력에 대한 항이다. 역동역학에 관한 더 자세한 설명은 참고문헌 (8)을 참조하기 바란다.

3.4 3 차원 로봇 시뮬레이터 인터페이스

앞 절의 내용을 적용하여 3 차원 로봇시뮬레이터 인터페이스를 개발하였다.

S/W 는 Visual C++ 6.0 을 이용하였으며 AutoCAD®

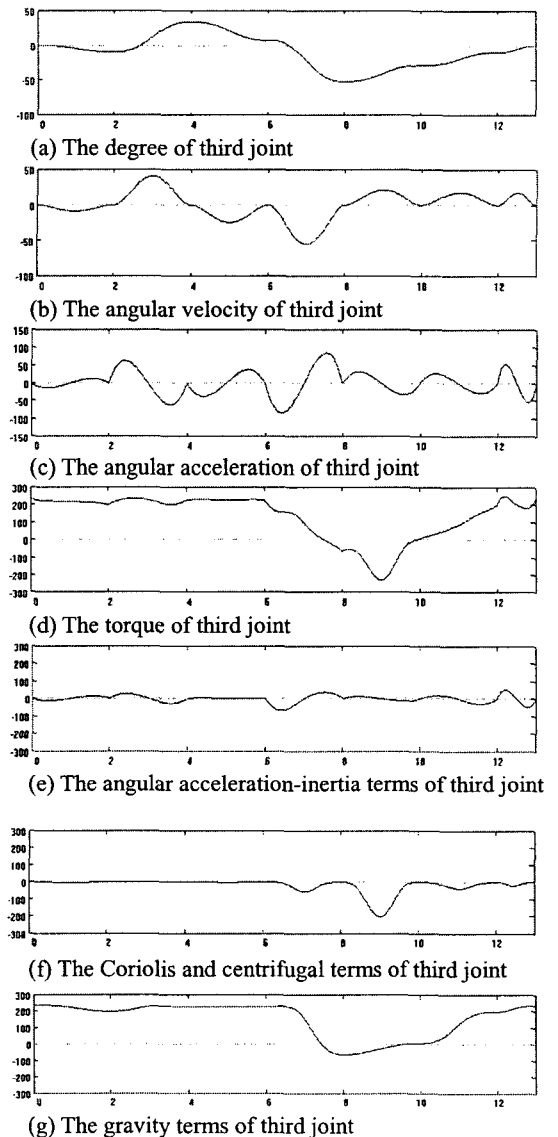


Fig. 11 The result of third joint

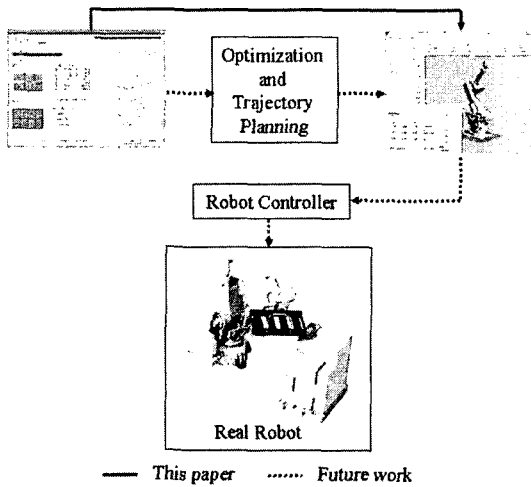


Fig. 12 System schematic of future work

를 이용하여 산업용 로봇을 Surface 모델링한 파일을 OpenGL®을 이용하여 3 차원으로 표현하였다(Fig. 9). 전체적으로 인터페이스는 크게 세 부분으로 구성되어 있다.

첫 번째는 로봇의 각 관절을 직접 슬라이더 바를 이용하여 움직일 수 있고 로봇의 모습을 3 차원으로 이동, 회전, 확대해 볼 수 있는 주 윈도우 화면, 두 번째는 로봇의 경로 계획을 로봇핸드의 위치, 회전, 이동시간의 정보로 생성할 수 있는 경로 입력화면, 세 번째는 로봇의 경로 이동 시에 각 관절 별로 시간축에 대하여 관절의 각도, 각속도, 각가속도, 필요토크 정보를 그래프로 출력하는 화면이다(Fig. 10).

시뮬레이터는 크게 2 가지의 모드로 나누어진다. 첫 번째로 로봇의 이동 경로를 입력하면 역기구학과 역동학식을 계산하여 관절의 정보와 모터의 필요토크를 그래프로 보여주는 모드, 두 번째로 모터에 토크를 주어지면 동역학식을 Runge-Kutta 방법으로 계산하여 로봇의 움직임을 시뮬레이션 하는 모드이다. 시뮬레이터의 성능평가를 위해 Fig. 11 에서는 첫 번째 모드로 총 13 초 동안 로봇이 움직일 때의 3 번째 관절의 결과를 예시하였다.

4. 결론

본 논문에서는 팔레타이징 시뮬레이터를 개발하기 위하여 Fast Algorithm 및 이를 이용한 팔레

타이징 패턴 생성 S/W 를 제안하였다. Fast Algorithm 은 기존의 PLP Algorithm 보다 향상된 계산시간을 보여 주었고, 이 알고리즘이 적용된 팔레타이징 패턴 생성 S/W 는 팔렛과 박스의 정보를 입력 받아 3 차원상의 팔렛 패턴을 생성하여 파일로 출력하였다.

3 차원 로봇 시뮬레이터는 팔레타이징 제어 S/W 의 로봇 모델과 작업경로 생성 알고리즘을 평가하기 위해 개발 하였으며 주어진 경로 점을 따라가면서 역기구학, 역동역학, 5 차 다항식계획법 등을 사용하여 팔레타이징 로봇을 평가 할 수 있게 하였다.

현재, 팔레타이징 작업에 최적화된 경로 생성 알고리즘을 개발 중이며, 향후 실제 로봇을 제어하는 부분까지 개발되면 모든 요소들이 통합될 것이다. 최종적으로는 전체 팔레타이징 작업을 시뮬레이션하고 실제 로봇과 연동되어 작업을 제어하는 S/W 를 제안할 것이다(Fig. 12).

참고문헌

- (1) Michael A. Hernan I, 2000, "Introduction to Automated Palletizing," Anderson Technical Service, Inc., 850 Morrison Road Chillicothe, Ohio 45601, pp. 1~3
- (2) Hyder Nihal Agha, William H. Decamp, Richard L.Shell and Ernest L. Hall, 2000, "Robotic Palletizing of Fixed- and Variable-Size/Content Parcels," Handbook of Industrial Automation Edited by Richard L. Shell and Ernest L. Hall, Marcel Dekker, Inc., pp. 673~686
- (3) Subir Bhattacharya, Rahul Roy and Sumita Bhattacharya, 1998, "An exact depth-first algorithm for the pallet loading problem," *European Journal of Operational Research*, Vol. 110, No. 3, pp. 610~625
- (4) Guntram Scheithauer and Uta Sommerweisz, 1998, "4-Block Heuristic for the Rectangle Packing Problem," *European Journal of Operational Research*, Vol. 108, No. 3, pp. 509~526
- (5) Seung-nam Yu, Heu-kwon Yoon, Sung-jin Lim, Soung-yel Lee, Young-hoon Song and Chang-soo Han, 2005, "The Development of Robot palletizing S/W using Fast Algorithm for PLP and 3D Robot Simulator," *Proc. KSME Spring Annual Meeting*, pp. 1663~1668
- (6) Young-Gun G. and Maing-Kyu Kang, 2000, "A Fast Algorithm for Two-Dimensional Pallet Loading Problems of Large Size," *European journal of operational research*, Vol. 134, No.1, pp. 193~202

- (7) Seungnam Yu, Sungjin Lim, Heukwon Yoon and Changsoo Han, "The Development of Robot Palletizing Simulator using Pallet Pattern Algorithm and Trajectory Optimization," AROB 2006, pp. 197~200
- (8) Sung-Jin Lim, 2004, "A Study on Analyzing Inverse Kinematics and Inverse Dynamics of 6 DOF Articulated Robot," Hankuk Aviation University
- (9) Phillip John Mckerrow, 1990, "Introduction to Robotics," Addison-Wesley, pp. 484~487
- (10) Saeed B. Niku, 2001, "Introduction to Robotics Analysis, Systems, Applications," Prentice Hall, pp. 119~172
- (11) Fu, K. S., Gonzalez, R. C., Lee, C. S. G., 1987, "Robotics, Control, Sensing, Vision, and Intelligence," McGraw Hill, pp. 462~468
- (12) John J. Craig, 1989 "Introduction to Robotics, Mechanics and Control Second Edition," Addison Wesley, pp. 230~246
- (13) Jin-seo Park, 1997, "A Comparative Analysis of Robot Languages and Design of a Robot Language Transfer, Hanyang University
- (14) Naak-hyung Lee, 1998, "A Design Experience on PC-Based Open Control for Robot Systems," Hanyang University
- (15) Yavuz A. Bozer and Ann Arbor, 1999, "Material Handling and Warehousing," Handbook of Industrial Robotics, Second Edition, Edited by Shimon Y. Nof, pp. 957~960
- (16) George, J. and Robinson, D., 1980, "A Heuristic Solution of the Rectangular Cutting Stock Problem" Computer J 23(4)
- (17) Hall, E.L., Slutzky, G.D. and Mazouz, A.K., 1987, "A Final Report, Development and Demonstration of Robotic Palletizing of Mixed Size and Weight Parcels for the Institute of Advanced Manufacturing Science"
- (18) Steudel, H.J., 1979, "Generating Pallet Loading Patterns: A Special Case of the Two-Dimensional Cutting Stock Problem," Management Science 25, pp. 997~1004
- Hong, K. D., Han, C. H. and Griffith, P., 1965, "The Mechanism of Heat Transfer in Nucleate Pool Boiling-Part I Bubble Initiation, Growth and Departure," Trans. of the KSME (B), Vol. 23, No. 5, pp. 887~904.