지능 로봇을 위한 계획기반의 혼합 주도권 제어

최재혁° 김하빈 김인철
경기대학교 전자계산학과
{01choi°, talkable, kic} @kyonggi.ac.kr

Plan-Based Mixed Initiative Control for Intelligent Robots

Jae-hyuk Choi°, Ha-bin Kim, In-Cheol Kim
Department of Computer Science, Kyonggi University

요 약

작업계획에 따라 자율 로봇을 동작시킬 때, 하드웨어상의 기계적 오차나 사람의 간섭 등 작업 계획상에 기술되지 않은 문제점들로 인해 로봇을 긴급하게 작동 중지시키거나 수동으로 조작하여 오차를 보정해야 한다. 오차를 보정하는 제어방법 중에서 특히 로봇의 초기 작업 계획상의 목표를 유지하면서 사용자가 작업제어를 할 수 있는 방법이 필요하다. 본 연구에서는 목표를 향한 일관성 있는 로봇과 사용자간의 혼합 주도권 제어방법을 설계하였으며 이것을 아이보 로봇을 이용한 작업제어에 적용하였다.

1. 서론

자율로봇을 제어하는 다양한 방법이 존재하지만, 본 논문에서는 계획기반의 계층적 제어구조를 바탕으로 로봇을 제어한다. 작업계획에 따라 자율 로봇을 동작시킬 때, 로봇 하드웨어의 오차나 여러 환경적인 영향으로 인하여 작업이 정상적으로 진행되지 않을 수 있다. 종종 로봇의 작업이 정상적으로 진행되지 않을 때 작업계획에 의한 로봇의 자율 행위를 중간에 멈추고, 정상적으로 작업이 진행될 수 있도록 사용자의 개입이 필요하다. 그러나 일단 정상적인 작업상태로 복귀한 후에는 로봇 스스로 원래 추구하던 작업목표를 달성하기 위해 자율적으로 동작하여야 한다. 가변성과 오차가 존재하는 실세계 작업환경을 고려하면 이와 같은 로봇의 자율 제어와 사용자의 직접 제어가 적절히 결합되는 혼합 주도권 제어방식이 효과적으로 적용될 수 있다[1]. 본 논문에서는 작업계획들 내에 예외 상황에 대한 사용자의 직접 제어를 처리해주는 계획을 포함시켜, 작업목표를 향한 일관성 있는 로봇과 사용자간의 혼합 주도권 제어 방법을 설계하고 구현하였다.

2. 로봇 작업 환경

본 연구에서는 Sony 사에서 개발된 강아지 로봇인 아이보(AIBO)를 이용하여 작업계획을 실행한다. 아이보 로봇을 이용하여 이루고자 하는 작업들은 [그림 1]과 같은 미로환경에서 로봇의 초기 위치에서 출발해 주어진 목적지까지 장애물을 피해 최단경로로 찾아가는 일이다. 로봇이 이러한 임무를 성공적으로 수행하기 위해서는 자신이 처음 놓여지는 위치를 파악하는 일, 이동 가능한 인접 경로들을 찾아내는 일, 일어서거나 몸을 돌려 이동을 준비하는 일, 선택된 노드까지 벽이나 다른 장애물들을 피해 걸어가는 일 등 다양한 지능행위들이 요구된다.

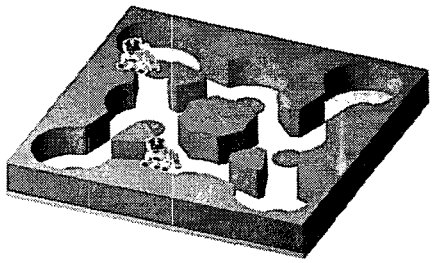

그림 1. 로봇의 작업환경

실험에서 요구되는 지능행위들은 작업실행기에서 작업계획에 따라 요청하게 되고, 아이보 Remote Framework (RFW) API를 이용하여 실행된다. 로봇 환경 특성상 실행 과정 중에 로봇이 작업실행기의 요청과 다른 행위나 요청을 처리하지 못하는 점들이 발견 되는 경우가 발생한다. 예를 들면 미로 환경에서 로봇이 벽에 닿거나 하여, 이동 할 거리보다 이동이 짧아지면 미로를 다 빠져나가지 못하게 된다. 미로를 빠져나가지 못하였음에도 작업실행기는 미로를 빠져나갔다고 인식하고 다음 경로를 찾는 작업계획을 실행하는 등의 오 동작이 발생할 수 있다.

3. 계획 기반의 작업 제어

아이보 로봇을 위한 제어구조는 크게 3 개의 계층으로 구성된다. 계획기반 제어구조의 최상위 계층은 주문형 작업계획 생성기 층으로서, 실행 중 계획 실행기의 요청이 있을 때에 필요한 새로운 작업계획을 즉시 생성해주는 역할을 수행한다. 그리고 그 아래에 행위모듈들의 실행을 제어하는 계획 실행기 층이 위치한

다. UM-PRS로 구현된 계획 실행기는 각 행위모듈의 실행과 중지에 대응하는 기본 작업계획들뿐만 아니라 이들을 결합하여 보다 복잡한 상위의 작업 목표를 달성해주는 다양한 복합 작업계획들을 미리 내장하고 있으며, 상황에 맞게 이들을 선택하여 실행하는 역할을 수행한다. 끝으로 맨 아래에 로봇의 행위모듈 층이 존재하는데, 이 계층은 Remote Framework API 를 이용하여 C++로 구현된 로봇의 다양한 행위모듈들로 구성된다. 전체의 작업계획은 작업 실행기의 작업계획으로 표현되지만, 작업 계획 실행기의 작업계획에 포함되지 못한 다른 작업 계획은 주운형 작업계획기에서 작업계획을 생성하여 실행하게 된다. 따라서 작업목표를 달성하기 위한 로봇의 자율행위는 대부분 사전에 작업실행기안에 내장하고 있는 작업계획들과 실행 중 작업 계획기에 의해 생성되는 작업계획에 의해 이루어진다.

많은 기계적 조작으로 이루어지는 로봇 동작과 잡음과 오차가 존재하는 실세계 작업환경으로 인하여, 로봇이 작업 중에 언제나 계획대로 실제 행위가 진행되고 있다고 보장하기 어렵다. 따라서 예측하지 못한 환경 변화와 기계적 오차로 인해 로봇 내부의 월드모델과 실제 상황이 달라지는 경우가 종종 발생하게 된다. 로봇과 작업 실행기의 월드모델이 동기화 되지 못하는 문제가 발생 하였을 때 일반적인 UM-PRS 상의 방법으로는 이러한 상황을 제어할 수 있는 방법이 존재하지 않는다. 이 상황에서 사용자가 해결할 수 있는 방법은, UM-PRS 를 통해 제어할 수 없기에 강제적으로 로봇을 이동시키거나, 작업계획과 무관하게 직접적으로 로봇을 직접 조작하여 해결하여야 한다. 작업 실행기의 작업이 완료되어 있으나 월드모델과 상황이 동기화 되지 못하고 다음 작업이 실행되고 있는 상황에서, 이전 작업의 결과와 로봇의 행위가 일치하지 않았다고 하여, 직접적인 제어를 하게 되면, 현재의 작업 실행기의 월드모델과 로봇의 상황이 일치하지 못해 정상적인 다음 계획의 실행을 보장받지 못하게 된다. 때때로 실행기의 월드모델과 동기화가 될 수도 있지만, 동기화 되지 못하였을 때 다음작업에 영향을 미치게 되어 모든 작업 계획이 실패하게 되는 결과가 될 수 있다. 동기화 되지 못하는 문제점들을 해결하기 위하여 자동화된 로봇의 작업계획의 과정에 있어서 로봇의 작업계획의 목표를 유지하면서 사용자의 작업제어를 하기 위한 방법이 필요하다.

4. 혼합 주도권 제어

시뮬레이션 환경과 달리 로봇 환경에서는 로봇 자체의 안전이나 주위 환경에 대한 예기치 못한 상황을 제어할 수 있어야 한다. 이를 위해 로봇의 행위를 긴급히 정지하거나, 정상적인 작업 흐름 사이에서 사용자가 제어권을 받아 로봇의 자율행동을 제어 이후에 계속할 수 있게 하는 방법 두 가지로 나눌 수 있다. 전자의 경우 응급상황에 사용할 수 있도록 즉각적으로 반응하고 더 이상 진행중인 계획을 달성하지 못하게 한다. 후자의 경우 로봇의 최소 행위단위까지 실행의 완료 후 로봇제어를 하는 계획을 통해 사용자 인터페이스로부터의 사용자 제어를 처리하게 된다.

작업 계획 실행기인 UM-PRS 에서는, 작업 계획안의 최소 작업 단위행위마다 실행되는 특별한 작업계획이 있다. [그림 2]와 같이 특별한 작업계획에 기술 된 부분들은, 작업계획내의 행위들이 실행될 때마다 지속적으로 실행 된다.

```
CYCLE
{
        RETRIEVE currentAction $action;
        EXECUTE Update_Action $action;
        UPDATE (currentAction)(currentAction $action);

        RETRIEVE currentAction $action;
        EXECUTE UpdateUserControlMode $state;
        WHEN: TEST (&& (== $state "TRUE")(!(== $action "NONE")))
        {
                UPDATE (inUserControlMode)(inUserControlMode "TRUE");
        };

        RETRIEVE inUserControlMode $currentState;
        WHEN: TEST (== $state "FALSE")
        {
                UPDATE (inUserControlMode)(inUserControlMode "FALSE");
        };

}
```

그림 2. 제어 요청 처리를 위한 작업 계획

제어요청을 통한 제어방법의 경우, 현재 실행중인 동작을 저장하고 사용자의 제어 요청이 있었는지를 확인한 뒤, 제어 요청이 있는 경우에는 사용자의 로봇 제어 모드로 동작시키도록 기술 하면 사용자의 제어를 작업계획 안에서 처리할 수 있게 한다.

```
KA {
        NAME:
                "ControlByUser"
        PURPOSE:
                ACHIEVE DirectControl;
        CONTEXT:
                FACT currentAction "NONE";
                FACT inUserControlMode "TRUE";
        BODY:
                EXECUTE GetUserCommand $command $param;
                EXECUTE AcceptUserCommand;

                OR
                {
                        TEST (==$command "Dance");
                        EXECUTE Dance $param;
                }
                {
                        TEST (==$command "WalkForward");
                        EXECUTE WalkForward $param;
                }
                                .
                                .
                                .

                };
                EXECUTE NotifyPlan "User Command : " $command;

                FAIL;
}
```

그림 3. 사용자 제어 작업계획

각각의 작업계획에는 사용자의 요청이 있었는지 확인하고 요청이 없는 경우에만 작업계획을 수행을 계속하며, 요청이 있는 경우 현재의 작업까지만 수행한다. 그 이후의 작업은 수행되지 않도록 처리 하였다가, 사용자의 작업이 종료되어 제어권이 반환된 이후에 다음 작업이 수행되도록 처리한다. 사용자의 직접 제어는 [그림 3]과 같이 로봇이 사용 가능한 행위들을 실행하는 작업계획을 호출하여 수행하게 한다. 사용자 제어모드에서도 사용자의 요청을 작업계획으로 처리한다. 각각의 작업계획들은 [그림 4]와 같이 현재의 동작 모드가 사용자 제어모드 여부를 확인하여 처리하게끔 하여 오 동작을 방지한다.

```
KA {
        NAME:
                "MoveToTarget"
        DOCUMENTATION:
                "MoveToTarget"
        PURPOSE:
                ACHIEVE MoveToTarget;
        CONTEXT:
                FACT currentAction "NONE";
                FACT inUserControlMode "FALSE";
        BODY:
                EXECUTE SetPlanText "MoveToTarget";
                EXECUTE MoveToTarget;
}
```

그림 4. 기본 작업 계획

실제 로봇을 이용하여 작업계획을 실행하는 경우 작업계획의 시작과 함께 로봇의 상태를 보여줄 수 있는 UI를 사용할 수 있게 되는데, [그림 5]에 나타나있는 UI에서는 로봇의 비전, 계획 실행 상태, 현재 이동 위치 등을 표현한다. 로봇이 작업계획을 생성할 수 있도록 목표 노드와 출발 초기 노드를 설정 하여, 주문형 작업 계획기가 작업 계획을 생성하고, 생성된 작업 계획을 바탕으로 UI에 현재 실행중인 작업을 표현한다.

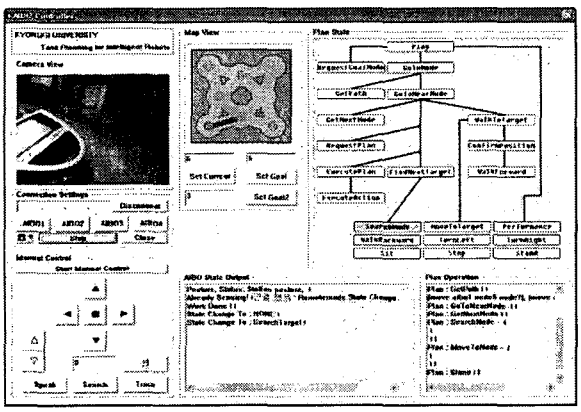

그림 4. 사용자 인터페이스 화면

계획에 따라 로봇이 동작하고, 여러 상황으로 인해 로봇의 월드 모델과 실제 위치가 동기화 되지 않는 경우 UI를 통해 사용자 제어권을 요청할 수 있다. 사용자 제어권이 요청되면, 현재 진행중인 행위까지만 실행이 되고, 그 이후에는 더 이상의 작업이 실행되지 않는다. 사용자 제어모드로 전환이 된 이후에는 사용자의 수동조작 요청에 따라 작업계획상의 제어를 받으며 작업이 수행되게 된다. 사용자의 수동 제어가 완료되면 다시 제어권 반환을 요청하여, 직전에 작업이 완료된 작업계획의 다음 작업계획을 다시 실행하게 하여 기존의 목표를 달성하도록 재 수행 한다.

UM-PRS는 실행 중에 멈추거나 하는 등의 제어를 할 수 없기 때문에, UM-PRS의 작업계획을 실행 중간에 제어하는 것은 가능하지 않다. 따라서 해당 작업계획이 모두 끝이 난 뒤에 사용자의 수동제어를 하거나, 작업계획과 무관하게 수동으로 작업계획을 해 주어야 한다. 작업이 이미 완료된 상태에서의 수동제어는 아무 의미가 없고, 작업계획의 실행 중에 작업계획과 무관하게 직접적으로 로봇을 제어하는 경우 로봇이 현재 처한 위치와 작업계획 실행기 월드 모델간의 동기화가 되지 않는 문제가 발생할 수 있다. 동기화가 되지 않는 문제로 로봇이 오작동 할 수 있어 정확한 목표 달성을 보장할 수 없다. 작업계획 중에 UI를 통해 수동제어를 요청하고, 제어권을 획득한 뒤 사용자가 수동 제어하는 이 방법은, 사용자의 제어가 지능행위 속에 포함되어 있기 때문에, 작업계획의 정상 동작을 보장하고 작업계획의 흐름을 깨지 않으면서, 기계의 동작 오차도 보정할 수 있는 장점을 가지고 있음을 알 수 있다. 아이보를 이용한 로봇 실험 환경에서 사용자의 작업제어가 작업계획에 포함되어 제어를 한 경우와 사용자가 제어하지 않은 경우를 비교 실험해 보았을 때 작업의 성공률이 크게 높아졌다.

5. 결론

자율로봇을 이용해 작업을 실행할 때 로봇이 가진 환경과 기계적 특성에 의해 오차가 발생한다. 오차를 바로 잡기 위해서 사용자의 제어가 필요하게 되고, 제어방법은 로봇의 원래 작업 목표에 영향을 주지 않으면서 로봇을 제어할 수 있어야 한다. 로봇을 제어하는 사용자의 긴급한 제어가 지능행위 속에 포함되어 사용자 제어 후 원래의 계획으로 복귀가 가능한 방법은, 사용자의 수동제어 이전에 실행 중에 있던 작업계획이 사용자의 수동제어 이후에도 복귀되어 지속적으로 실행할 수 있다는 장점을 가지고 있다. 본 논문에서 제시한 혼합 주도권 제어 방법은, 사용자의 긴급한 제어뿐만 아니라, 지능 로봇의 작업계획 실행중간에 발생하는 문제점을, 현재 실행중인 작업계획이 아닌 새로운 작업계획을 추가하여 해결하는 방법으로의 개선도 가능하다.

참고문헌

[1] Burstein, M. and McDermott, D. "Issues in the Development of Human-Computer Mixed-Initiative Planning" In B. Gorayska and J.L. Mey eds., Cognitive Technology, pp.285-303. 1996.

[2] J. Lee, M. Huber, E. Durfee, and P. Kenny, "UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications," Proc. of the Conf. Intelligent Robotics in Field, Factory, Service, and Space, pp. 842-849, 1994.

[3] Michael A. Goodrich, et al, " Experiments in Adjustable Autonomy," Proc of IJCAI-01, 2001.

[4] P. Scerri, K. Sycara, and M. Tambe, "Adjustable Autonomy in the Context of Coordination," Proc of AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit, 2004.

[5] Alberto Finzi, Andrea Orlandini, " A Mixed-Initiative Approach to Human-Robot Interaction in Rescue Scenarios," Proc of ICAPS-05 Workshop on Mixed-Initiative Planning and Scheduling, 2005.