



電氣技術者를 위한

# 産業用 로봇 技術

(10)

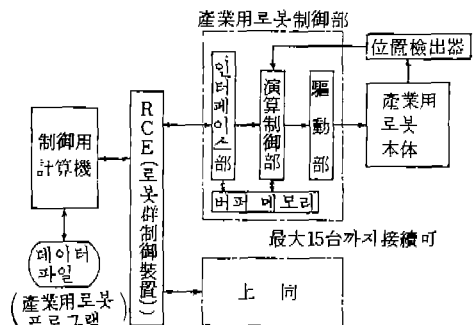
## (2) 産業用 로봇의 群制御

多數의 産業用 로봇을 1대의 계산기로 制御하는 群制御의 경우, 演算形 構成으로는 계산기의 負荷가 커지기 때문에 어렵다. 이에 대해 記憶形은 産業용 로봇에 필요한 演算을 개개의 제어장치 가 분담하기 때문에 計算기의 負荷가 대폭 軽減된다. 또한 제어장치 중의 기억장치 는 産業用 로봇의 작업내용을 보존하기 위해 不揮發性이 요구되므로 코스트 중에서 점유하는 比重이 크지만, 記憶形의 경우, 이것을 計算기 내에 集中시켜 드럼이나 디스크와 같은 大容量의 메모리를 이용할 수 있기 때문에 전체적으로 코스트 퍼포먼스가 높은 시스템을 실현시킬 수 있다.

記憶形 構成을 채용한 産業용 로봇 群制御裝置의 구성을 그림 4·46에 든다. 이 시스템은 制御用 計算기를 사용, 産業用 로봇의 제어장치와 의 사이에 産業用 로봇 群制御裝置 RCE를 부가해서 구성하고 있다. RCE는 計算기와 제어장치간의 데이터 전송을 관리한다. 그 상호간

접속에는 시스템으로부터의 요구에 따른 方式을 선정해야 한다. 즉, 計算기와 RCE 간은 計算기에 허용되는 오버 헤드 의 大小에 따라 DMA 또는 PCIO 어느 하나로 轉送을 하고 RCE와 제어장치간은 거리나 情報량에 따라 시리얼 또는 패러렐 중 어느 하나의 전송방식을 채용한다.

群制御에 있어서의 가장 중요한 처리는 複數의 産業용 로봇이나 機器의 協調動作을 가능케 하기 위한 實行監視制御이며, 이를 위해 각종 記錄 情報의 管理가 필요해진다. 그림 4·47은 産業



〈그림 4·46〉 産業用 로봇의 群制御

용 로봇과 加工機械, 搬送機械 등의 周辺機器를 群制御하는 경우의 인터록 情報의 흐름을 표시한 것이다.

계산기 시스템은 産業用 로봇 주변기기와의 사이에서 位置 데이터뿐 아니라 起動, 停止나 動作完了의 신호도 전송하고 있다. 계산기 시스템은 이러한 情報를 入力하면 미리 정해진 순서에 따라 타이밍 制御를 하고 위치 데이터나 起動, 停止信號를 出力한다. 또한 특정한 産業用 로봇과 주변기기 사이에서만 처리할 수 있고 타에 영향을 미치지 않는 정보는 계산기를 통하지 않고 直結할 수가 있다.

群制御의 또 한가지 역할은 位置 데이터 管理이다. 그 가장 기본적인 것으로 敎示作業의 簡略化가 있다. 敎示作業에서는 티치 콘솔로 産業용 로봇을 조작하면서 動作계적, 速度, 인터록 등의 정보를 버퍼 메모리를 통해 계산기의 데이터 파일에 기억한다.

이것은 상당히 勞力을 필요로 하는 作業이다. 예를 들어 공작기계에 대해 워크를 着脫시키는 경우를 생각하면 産業용 로봇 1대당 敎示點의 총수는 워크 1개당 점수에, 대상으로 하는 工作機械 대수, 워크 종류 등을 곱한 것이 되며, 1만 점을 넘는 경우도 적지 않다. 이것을 多數의 産業용 로봇에 대해서 행하는 경우, 敎示에 소요되는 시간이 방대해질 것은 쉽게 추측할 수 있을 것이다.

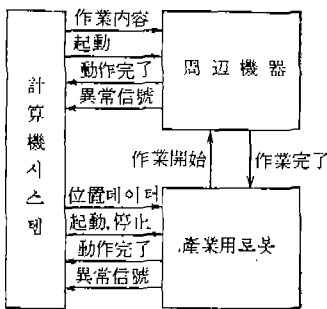
그러나 실제로 作業內容을 분석해 보면 相異

한 공작기계에 있어서도 상호 공통하는 작업은 상당히 많다. 또 워크 또는 産業用 로봇 상호에 있어서도 동일하다. 따라서 만일 共通動作을 1회의 敎示로 대표할 수 있으면 실제로 티치 콘솔을 사용해서 하는 敎示作業은 대폭 감소한다. 여기서 궤적 전체의 回轉이나 平行移動까지 자동적으로 하게 된다면 생략할 수 있는 범위는 더욱 넓어진다.

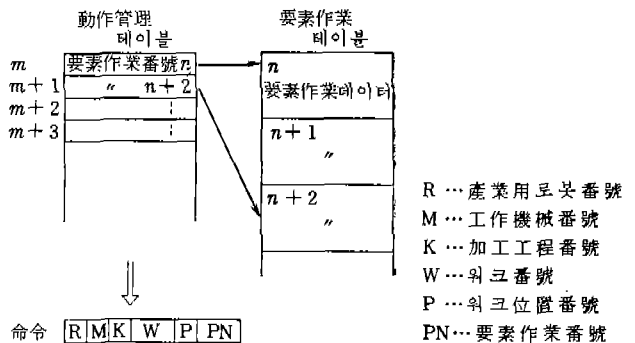
計算機를 사용한 경우, 이상과 같은 處理는 별로 어려운 것은 아니고 다음과 같은 順序로 실행된다. 최초로 모든 로봇의 動作을 要素作業으로 분해하고 여기에 번호를 붙인 후 티치 콘솔을 사용해서 敎示, 그림 4·48에 든 要素作業 테이블에 기억한다. 한편, 産業用 로봇의 작업은 그림 4·48에 표시한 命令으로 表現하고 실행순서에 따라서 作業管理 테이블을 作成한다.

動作의 실행에서는 作業管理 테이블에 따라서 要素作業을 선택하고 工作機械, 워크의 종별과 위치 등의 情報에 따라서 위치를 補正한 후에 지정된 産業용 로봇에 出力한다.

위에서 설명한 테이블 方式의 프로그램은 작업내용의 변경에 대해서는 상당히 효과적이지만 問題는 레이아웃의 변경에 대해 融通성이 적다는 것이다. 이것을 해결하고 어떠한 상황에서도 敎示作業이 손쉽게 시행될 수 있게 하려면 産業용 로봇의 動作 記述을 체계화한 로봇用 言語의 出現이 요망된다. 다만, 産業용 로봇의 프로그램 특징은 位置 데이터를 현장에서 설정할 수



〈그림 4-47〉 인터록 情報



〈그림 4-48〉 敎示 데이터의 管理

있다는 것이고 産業用 로봇의 동작을 記述하는 命命을 부여받아도 이것만으로 敎示作業과 동일한 내용을 표현할 수는 없다. 이 문제는 로봇용 言語와 현장에서의 敎示를 병용한 對話形 敎示方式이 제안되고 있다. 어느 경우이건 계산기에 의한 산업용 로봇의 群制御가 본격화하는 데 따라서 계산기의 특징을 살린 敎示作業의 간략화가 중요한 과제가 된다.

### (3) 産業用 로봇의 直接制御

그림 4·45에 든 分類中에서 演算形의 構成을 가지고 산업용 로봇을 直接制御하는 예는 연구실 단계의 것을 제외하면 극히 적다. 이것은 주로 가격면의 制約이 원인인데, 마이크로 컴퓨터의 급속한 進歩로 해결될 것이다.

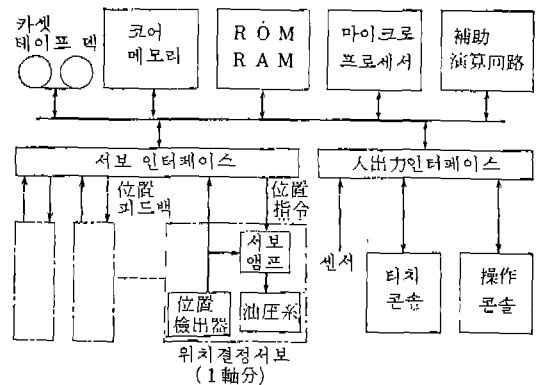
産業用 로봇에 요구되는 機能은 점점 고도화하고 있으며 특히 숙련작업자와 동일한 작업을 할 수 있는 産業用 로봇에 대한 기대가 크다. 이 경우 숙련작업자가 스스로의 感覺에 의지해서 실행하는 作業을 代行하기 위해 制御裝置에 요구되는 것은 복잡한 軌跡을 원활하게 트레이스하는 機能과 外部에 설치한 센서의 出力을 사용해서 外界에 적응하는 機能의 두가지이며 어느 경우도 계산기의 특징을 살림으로써 대처할 수 있는 것이다. 계산기를 사용하면 이것 이외에도 敎示作業의 간략화, 시방의 변경에 신속히 대처할 수 있는 柔軟性, 異常檢出이나 진단에 의한 信賴性과 保全性의 향상 등이 부가적인 效果로서 얻어진다.

마이크로 컴퓨터의 등장으로 價格面의 문제가 해결될 전망이 있으므로 나머지는 産業用 로봇이 요구하는 고속처리에 걸릴 수 있는가의 여부이다. 즉, 산업용 로봇의 경우, 가장 高速性이 요구되는 位置決定 서보에 있어서는 샘플링 周期가 1ms 이상의 고속 처리를 필요로 하기 때문에 이것에 모든 處理를 시리얼로 實行하는 계산기를 적용하는 경우의 本質的인 문제이다. 따라서 계산기를 적용해서 制御裝置를 구성하는 경우에는 프로그램의 구조 및 주변 하드웨어와

의 機能分擔에 대해 충분히 검토하여 계산기의 處理能力을 有效하게 活用할 수 있도록 배려해야 한다. 그림 4·49에 이상과 같은 방식에 입해서 構成한 制御裝置의 예를 든다.

여기서는 實用化를 전제로 마이크로 컴퓨터에 의한 構成을 들었다. 마이크로 프로세서, 半導體 메모리인 ROM과 RAM, 補助演算回路가 계산기에 상당한다. 한편, 종래의 계산기 중에서 기억장치에 상당하는 것은 코어 메모리로서 산업용 로봇의 위치 데이터나 인터록 情報를 기억한다. 카셋 테이프 팩은 코어 메모리의 데이터를 保存하는 데 사용한다. 또 터치 콘솔과 操作 콘솔은 종래의 제어장치와 같지만 센서는 感覺 機能을 실현하기 위해 부가되어야 한다.

産業用 로봇의 制御 중에서 가장 高速性이 요구되는 것은 위치결정이며, 油壓 서보의 경우 1ms 이상의 샘플링 周期로 처리하지 않으면 안정된 동작은 기대할 수 없다. 일반적으로 産業用 로봇은 數軸의 自由度를 갖고 있기 때문에 이것을 동시 처리하는 것은 마이크로 컴퓨터는 물론, 일반 계산기로도 어렵다. 한편, 위치결정 서보의 회로 자체는 애널로그 回路를 사용해서 상당히 콤팩트하게 구성되어 있으며 성능상으로도 문제가 없기 때문에 계산기로 대체하더라도 큰 효과는 기대되지 않는다. 따라서 그림에서처럼 서보 앰프는 하드웨어로서 남기고 서보 인터페이스를 통해 위치지령과 위치 피드백 信號를



(그림 4-49) 計算機를 적용한 制御裝置의 構成

轉送하는 構成이 現實的이다.

여기서 대상으로 하고 있는 것은 앞에서 설명한 것과 같은 숙련작업자에 필적하는 작업을 실행하는 産業用 로봇이므로 位置指令으로서 코어 메모리의 데이터를 그대로 出力하는 것이 아니라 補間, 座標變換, 센서의 出力에 의한 補正 등을 한 후에 出力해야 한다. 따라서 演算의 일부를 외부에 남기고 있지만 구성상은 演算形에 속한다. 연산의 주기는 위치결정 오차의 허용값에 따라서 정해지지만 일반적으로는 位置決定 서보의 周期과 비교해서 상당히 길게 잡아도 문제되지 않기 때문에 계산기로 충분히 실행 가능할 범위에 있다.

이들 演算을 실행하는 프로그램은 敎示와 같은 産業用 로봇의 기본적인 처리와 함께 ROM에 格納한다. 그러나 마이크로 컴퓨터를 사용한 경우, 현재로서는 일반의 계산기보다 수치 계산 능력이 뒤떨어지기 때문에 필요에 따라서 補助 演算回路를 부가하여 乘除算, 삼각함수의 계산 등, 마이크로 컴퓨터의 부족한 부분을 커버해야 한다.

다음에 實例에 입각해서 계산기가 실행하는 具體的인 처리를 설명한다. 實例로 든 것은 용접 로봇으로서, 直交하는 3개의 軸(X, Y, Z軸)에 굽힘(BD軸)과 흔들림(SW軸)의 2軸을 갖는 핑거를 부가시킨 다섯가지 自由度로 아크 용접 작업을 하는 경우이다. 용접용 토치는 핑거 先端에 지지하지만 아크 용접을 하기 위해서는 암의 선단이 지정된 軌跡을 等速으로 원활하게 트레이스해야 한다. 또 용접의 워크는 일반적으로 加工이나 組立의 정밀도가 좋지 않기 때문에 敎示된 點을 충실히 트레이스 하는 것만으로는 양호한 용접은 할 수 없다. 이 두가지 문제를 計算機에 의한 制御로 해결하는 것을 생각한다.

軌跡의 트레이스는 敎示作業의 간략화를 위해 PTP方式을 採用하고 데이터를 補間해서 出力한다. 그러나 이 産業用 로봇에서는 워크의 구석 등에 손목의 動作이 부가되므로 各軸의 位置 데이터를 補間할 뿐만 아니라 좌표 변환을 병용

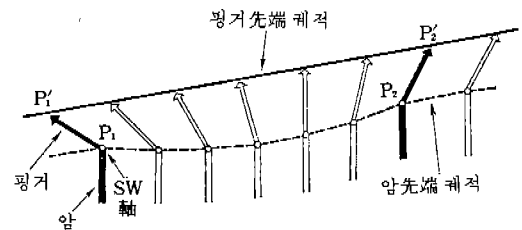
해야 한다. 예를 들면 그림 4·50과 같은 경우, 핑거의 先端이  $P_1'$ 과  $P_2'$ 의 사이를 직선적으로 이동해야 하므로 各軸의 위치 데이터를 좌표 변환해서  $P_1'$ 과  $P_2'$ 의 위치 데이터를 구한다.

다음에 이 2點間의 거리를 계산하여 이것을 等分割해서 補間의 점수를 정한 후 補間演算을 한다. 이 결과,  $P_1'$ 과  $P_2'$  사이를 직선으로 트레이스하는 補間을 얻게 되므로 이에 대해 반대의 좌표 변환을 시행, 각축의 위치 데이터를 구한다. 따라서 그림 4·50의 경우에는 암 先端  $P_1$ 과  $P_2$ 는 핑거의 움직임을 補正한 軌跡을 간다.

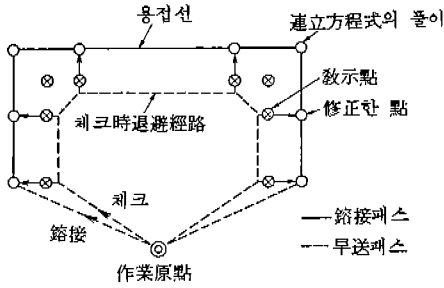
한편, 워크의 誤差는 토치와 워크의 상대 위치를 검출하는 센서를 사용하여 그림 4·51에 든 체크 動作을 행함으로써 대치한다. 즉, 용접 작업에 앞서서 신속 이송으로 敎示點을 가고, 센서로 정확한 위치를 탐색하는 動作을 반복한다. 이에 의해 하나의 軌跡 전체를 補正하지만 구석의 點에서는 센서를 사용할 수 없고 탐색을 할 수 없기 때문에 다른 點에서의 결과에서 聯立方程式을 풀고 구석 위치를 구한다.

이상이 용접용 로봇에서의 대표적인 처리인데 이것을 종합하면 실로 많은 論理判斷과 數值計算을 필요로 하며 계산기를 사용하지 않으면 실행이 불가능하다고 할 수 있다.

計算機를 이용하는 경우, 機能의 追求와 함께 중요한 것은 계산기의 특질을 살려서 機能의 유연성을 부가하는 것이다. 계산기에 의한 制御에서는 소프트웨어의 변경만으로 機能의 추가, 삭제가 가능하기 때문에 일반적으로 通用성은 증가하지만 소프트웨어의 開發 코스트를 생각하면



〈그림 4·50〉 座標變換을 수반하는 補間



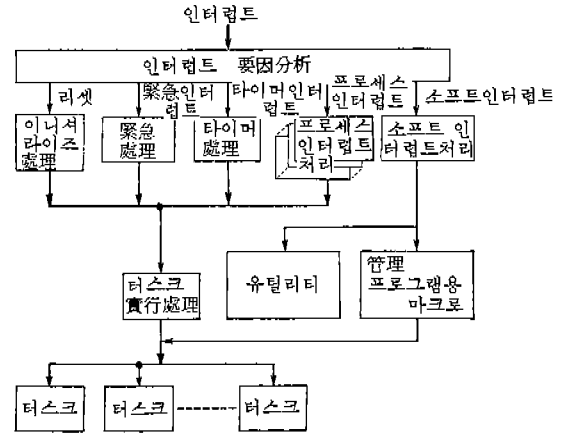
(그림 4-51) 銲接用 로봇의 체크 動作

가능한 한 制御方式의 표준화를 도모해야 한다. 또 표준화한 制御方式을 表現하는 프로그램은 일정한 순서로 管理할 수 있게 하는 것이 바람직하다. 따라서 일반의 계산기 제어에서 DS라고 하는 일종의 管理 프로그램이 필요해진다.

계산기에 의해 産業用 로봇을 직접 제어하는 경우의 일반적 프로그램의 構造, 管理 프로그램에 요구되는 機能과 役割에 대해 살펴본다.

그림 4-52에서 管理 프로그램은 계산기에 더해지는 각종 인터럽트를 分析하여 미리 정해진 優先順位에 따라서 任務이라고 하는 單位 프로그램을 기동한다. 또 任務 중에서 공통으로 사용되는 演算은 서브 루틴화하여 유틸리티로서 준비한다. 용접용 로봇의 경우, 任務에 상당하는 것은 敎示, 探索, 銲接 등이고 서브 루틴으로서서는 補間, 좌표 변환, 聯立方程式 등이 있다. 이런 構造를 가지면 任務 單位로 프로그램을 취급할 수 있기 때문에 制御方式의 표준화를 하기 쉽다. 그래서 機能의 확장이나 변경이 용이해지고 계산기 본래의 응용성이 충분히 발휘된다.

그런데 여기에 든 管理 프로그램의 기본적인 機能은 일반적인 OS와 동일하지만 産業用 로봇의 연산제어를 위해서는 管理 프로그램이 介在함으로써 생기는 오버헤드가 處理速度의 저하로 이어지지 않도록 해야 한다. 따라서 産業用 로봇의 制御를 전제로 한 管理 프로그램에서는 일반 OS에서 중요시되는 沉用性, 擴張性은 다소 희생하더라도 應答性을 첫째로 생각해야 한다.



(그림 4-52) 프로그램의 構造

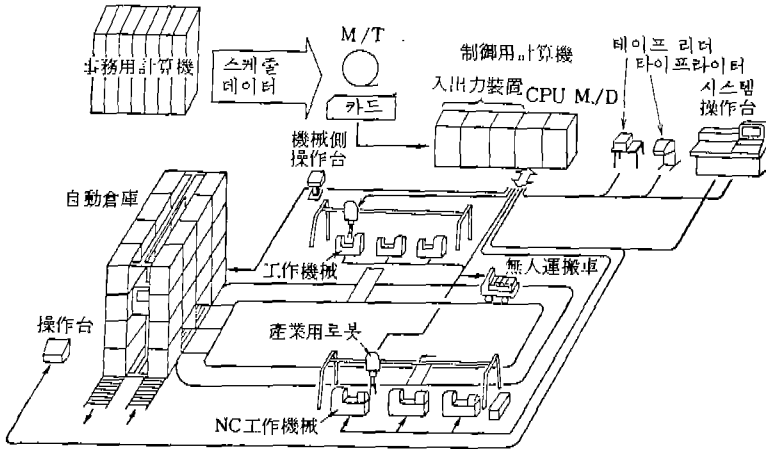
이상 마이크로 컴퓨터를 사용한 産業用 로봇의 演算制御例에 대해 설명했지만 마이크로 컴퓨터의 技術은 刻刻으로 進歩하고 있으며 개발 시점에서의 최신기술을 도입하는 努力이 있어야 한다.

#### (4) 統合生産 시스템

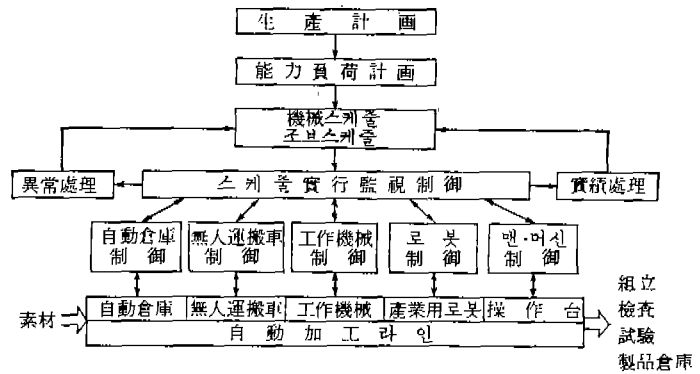
끝으로 産業用 로봇을 포함하는 生産 라인을 계산기로 制御하는 統合生産 시스템의 예에 대해 고찰한다. 그림 4-53에 든 것은 部品加工 라인으로서 産業用 로봇 이외에 工作機械, 無人搬送車, 自動倉庫 등을 포함한다.

制御는 그림 4-44에 든 階層構成에 따라서 하게 되며, 2종의 계산기를 가지고 있다. 이 예에서는 産業用 로봇은 핸들링 작업만 하면 되기 때문에 복잡한 演算制御는 필요로 하지 않는다. 따라서 앞에서 설명한 記憶形의 구성을 가지며 다른 機器와 함께 制御用 계산기에 의해 群制御하고 있다. 이 시스템에서의 情報의 흐름은 그림 4-54와 같다.

사무용 계산기는 오프 라인의 스케줄링을 분담하고 있으며 결과를 제어용 계산기에 넘긴다. 이 이후, 온 라인의 스케줄링에서 實行監視制御 및 機器 單體의 피드백 제어를 제외한 機器制御까지를 제어용 계산기가 實行한다. 이 중에서 實行監視制御는 産業용 로봇을 비롯한 각 기기



〈그림 4-53〉  
통합생산 시스템의 예



〈그림 4-54〉  
통합생산 시스템에 있어서의 處理

의 群制御를 의미하며, 그림 4-47을 일반화 한 것이다.

즉, 각 기기의 동작상태를 항상 監視, 로트內 製作員數의 파악과 다음 作業의 決定, 作業준비 指示, 計測指示, 칩 교환지시, 기기시간의 타이밍 制御, 스케줄 갱신 등의 처리를 하고 스케줄을 확실히 실행하는 역할을 다한다.

#### (5) 앞으로의 動向

계산기의 技術은 현재도 시시각각으로 進歩하고 있다. 이 進歩는 보다 높은 性能, 機能을 追求하는 방향과 마이크로 컴퓨터로 대표되는 것과 같은 小形化를 追求하는 方向으로 分極化될 것으로 예상된다. 따라서 産業用 로봇에 적용하

는 경우에도 이러한 進歩를 개발에 반영시켜야 한다.

즉 前者에 대해서는 계산기의 높은 能力을 살리기 위해 産業용 로봇의 實情에 알맞는 보다 고도의 소프트웨어를 指向할 필요가 있다. 이 方向은 종래 計算制御의 연장선 상이라고 생각해도 된다. 그러나 小形化라는 方向의 進歩에 대해서는 계산기라는 개념을 버려야 하는 경우도 있다. 마이크로 컴퓨터의 進歩에서 推定하면 가까운 장래, 10~20개의 IC로 구성되는 小形의 제어회로로 매치하여도 충분해질 것이다. 따라서 마이크로 컴퓨터를 높은 機能의 LSI라고 생각하고 사용할 수도 있고 이에 의해 産業用 로봇의 性能, 機能이 향상될 수 있을 것이다.