Journal of The Korea Society of Computer and Information
Vol. 19, No. 1, January 2014
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2014.19.1.001

순환실행체제를 이용한 무인 자율주행 실시간 스마트 크루즈 컨트롤

이 재 면*, 강 경 태*, 노 동 건**

Cyclic Executive for Autonomous Driving with Real-Time Smart Cruise Control

Jaemyoun Lee*, Kyungtae Kang*, Dong Kun Noh**

요 약

최근 사용자의 편의성과 안정성을 향상시키기 위해, 자동차 기술과 IT 기술을 접목한 지능형 자동차 개발에 많은 관심이 모아지고 있는데, 그 대표적인 기능으로 무인 자율주행과 스마트 크루즈 컨트롤이 있다. 지능형 자동차는 설계 및 구현 시 자동차가 요구하는 실시간 제약 조건을 만족하는 것이 매우 중요하다. 본 연구에서는 무인 자율주행과 스마트 크루즈 컨트롤 기능을 모형차량에 구현하고, 이에 적합한 실시간 스케줄링 알고리즘을 연산 복잡도가 낮고 구현이 간단한 순환실행체제를 이용하여 설계하였다. 또한 모의실험을 통하여 실제 시스템으로의 적용 가능성을 입증하였다.

▸ Keywords : 스케줄링, 실시간 시스템, 지능형 자동차, 모터 제어, 임베디드 시스템

Abstract

In recent years, much attention has been paid to the development of intelligent vehicles that integrate automotive technology into the information technology, with the aim of improving user friendliness and stability. The representative function is a autonomous driving and a cruise control. In designing such vehicles, it is critical to address the real-time issues (i.e., real-time vehicle control and timely response). However, previous research excluded the real-time scheduling. We develop a model car with unmanned cruise control, design the real-time scheduler using cyclic executive to easily adapt the model car, and provide some insight into potential solutions based on various experiments.

•제1저자 : 이재면 •교신저자 : 노동건
•투고일 : 2013. 10. 05, 심사일 : 2013. 10. 25, 게재확정일 : 2013. 11. 15.
* 한양대학교 컴퓨터공학과(Dept. of Computer Science and Engineering, Hanyang University)
** 숭실대학교 정보통신전자공학부(School of Electronic Engineering, Soongsil University)
본 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 과제임 (NRF-2011-0012996)

▸ Keywords : Scheduling, Real-time Systems, Intelligent Vehicle, Motor Control, Embedded Systems

I. 서 론

단순 이동수단에 불과했던 자동차는 IT 기술과 융합되면서 지능형 자동차로 거듭 진화하고 있다. 특히 인명사고를 유발할 수 있는 자동차의 안정성을 보장하는 것은 매우 중요한 문제이다. 자동차의 안정성을 보장하기 위해, 시간 제약성을 만족하는 실시간 시스템을 도입하고, 스케줄링 가능성 분석이 반드시 선행되어야 한다[1-3].

기존 지능형 자동차의 시간 제약성을 보장하는 연구는 모터 제어와 같은 지능형 자동차의 기능을 구현하지 않고, 추상적인 프로세스를 가정한다[4]. 이런 연구는 실제 자동차에 적용할 경우 물리적 환경의 다양성으로 인하여 예기치 못한 결과가 발생할 가능성이 높다. 따라서 지능형 자동차의 기능을 직접 구현하고 시현을 통하여 성능을 분석하는 것이 시스템의 신뢰성 구축에 필수적이다. 그러나 이 과정은 비용 측면에서 효율적이지 못하다. 따라서 모형 차량을 이용한 프로세스 검증이 다각도로 모색되고 있다.

본 연구에서는 지능형 자동차의 대표적인 기술 중 하나인 무인 자율주행(unmanned autonomous vehicle)과 스마트 크루즈 컨트롤 시스템(Smart Cruise Control systems)을 모형차량에 구현하고 시간 제약성을 가지는 프로세스를 관리하기 위한 실시간 스케줄링 알고리즘을 제안한다. 자율주행과 스마트 크루즈 컨트롤 기능 구현 시 운전자의 안전을 보장하는 것이 최우선 되어야 하므로 이 프로세스를 독립화 하여 다른 프로세스들로부터의 간섭(이를테면 컨텍스트 스위치(context switch)을 최소화 할 필요가 있다. 또한 예기치 못한 이벤트에 신속히 대응하기 위하여 연산 복잡도가 낮은 프로세스 관리자를 사용하는 것이 바람직하다. 이에 우리는 지능형 자동차의 시스템에 간단하게 적용 가능한 순환실행체제 기법을 제안하고 이에 대한 성능을 모형차량의 모의 무인주행을 통해 검증하였다. 그 결과, 제안한 순환실행체제가 구현하고자 하는 지능형 자동차의 시간 제약성을 충분히 만족시킬 수 있음을 확인하였다.

본 논문은 총 5장로 구성되어 있다. 2장에서는 실시간 시스템에 대하여 소개한다. 3장에서는 구현한 지능형 자동차 시스템과 제어 기법을 소개하고, 적용한 실시간 순환실행체제를 소개한다. 5장에서는 시간 제약성을 만족하는지 확인하기 위한 실험 방법과 결과를 보여주며, 6장에서 결론을 맺고 향후 연구에 대해서 검토한다.

II. 관련 연구

자동차와 같이 프로세스의 시간 제약성을 만족하는 시스템을 실시간 시스템이라고 한다. 실시간 시스템은 프로세스, 프로세스 관리자, 스케줄러로 구성된다. 프로세스는 시스템이 수행하는 작업이고, 이들은 서로간의 독립적이어야 한다. 프로세스 관리자는 프로세스의 정보를 저장하고, 현재 실행하는 프로세스를 관리한다. 스케줄러는 프로세스의 실행 순서를 결정한다. 이러한 스케줄러의 성능에 따라 실시간 시스템의 성능 및 신뢰도가 달라진다.

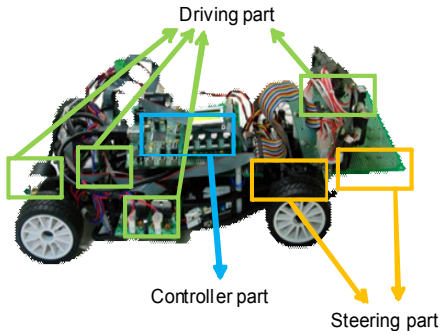


그림 1. 시스템 구성도
Fig. 1. System block diagram

실시간 시스템의 목적은 각 프로세스가 가지고 있는 시간 제약성을 위반하지 않도록 주기적인 프로세스들(periodic processes)을 스케줄링하는 것이다. 주기적인 프로세스는 정해진 시간마다 반복적으로 실행하는 작업이다. 정해진 시간은 주기(period)로써, 하나의 프로세스가 요청하는 시간 간격이다.

주기적인 프로세스는 데드라인(deadline), 즉 시간 제약성을 가진다. 실시간 시스템이 일반 시스템과 달리 특별한 실시간 스케줄러를 갖는 이유가 모든 프로세스의 데드라인을 확실하게 보장하기 위함이다. 직관적으로 데드라인이 주기보다 크면, 시간 제약성을 보장할 수 없는 불가능한 시스템임을 알 수 있다. 따라서 데드라인은 항상 주기와 같거나 작아야 한다.

또, 주기적인 프로세스는 매주기 종료될 때 까지 필요한 실행시간(execution time)을 갖는다. 다른 속성과 달리 실행시간은 상황에 따라 가변적이기 때문에 보통 최악실행시간(worst-case execution time)을 기반으로 스케줄 가능성을 분석한다[5]. 최악의 실행시간은 주기보다 작거나 같아야 하기 때문에 아래의 조건을 만족한다[6].

$$C_n \leq d_n \leq p_n \quad (1)$$

C_n는 실행시간을 나타내고, d_n은 데드라인을 나타내며, p_n은 주기이다[7].

스케줄러는 스케줄링하는 시점을 기준으로 온라인 스케줄링과 오프라인 스케줄링으로 나눈다. 온라인 스케줄링은 시스템이 동작하면서, 매 시간마다 또는 프로세스 컨텍스트 스위치가 일어날 때 마다 스케줄링을 다시 한다. 반대로, 오프라인 스케줄링은 설계 레벨에서 프로세스의 실행 순서를 정한다. 온라인 스케줄링이 보다 동적이고 적응적(adaptive)이지만, 매번 스케줄링을 재연산해야 하는 오버헤드가 있다. 그래서 프로세스의 실행 순서가 변하지 않는 경우, 또는 설계 레벨에서 실행 순서를 확실하게 예측 가능한 경우, 오프라인 스케줄링이 보다 효율적이다.

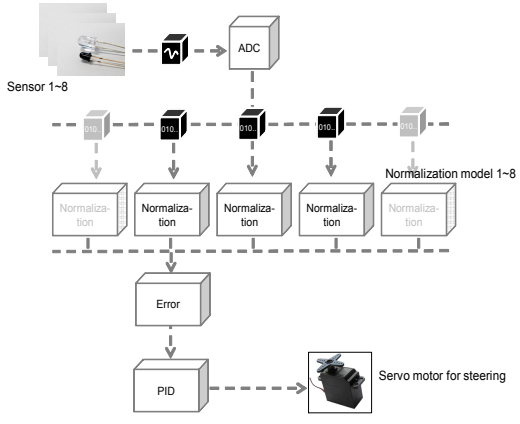


그림 2. 무인 자율주행을 위한 조향부 다이어그램
Fig. 2. Steering part diagram for unmanned autonomous vehicle

오프라인 스케줄링을 구현하는 대표적인 기법이 순환실행체제(cyclic executive)이다. 순환실행체제는 단일 CPU환경에서 다수의 주기적인 프로세스가 시간제약성을 만족하도록 실행 순서를 사전에 결정하여 배치한 프로그램 또는 그러한 컨트롤 구조를 말한다[8].

III. 제안한 시스템 구조

1. 지능형 자동차 시스템

그림 1는 제안한 지능형 자동차의 시스템 구성도이다. 크게 3 부분으로 이루어져 있다. 제어부(controller part)는 MCU와 간단한 조작을 위한 스위치 및 LCD로 이루어져 있다. 조향부(steering part)는 차로를 검출하는 센서와 방향을 조절하는 서버 모터로 구성된다. 구동부(driving part)는 선행차량과의 거리를 측정하는 센서, 현재 속도를 감지하는 엔코더, 구동 모터로 구성된다.

1.1. 무인 자율주행

본 연구에서는 지능형 자동차의 대표적인 기능 중 하나인 무인 자율주행을 모형차량에 구현하였다. 구현된 모형차량은 무인 자율주행을 위하여 정해진 특수 차로를 이용한다. 차로 중앙에 검은색 라인이 위치하고, 배경 색상은 검은색의 보색인 흰색이다. 보색은 빛의 반사량 차이를 가장 크게 만든다. 이 차이를 이용하면 차로의 위치를 추적할 수 있다. 이러한 특수차로는 현재 상용화된 무인 열차와 무선 충전 자동차에 적용 가능한 기법이다[8].

빛의 반사량을 측정하기 위한 센서부는 적외선 센서 8조로 이루어졌다. 적외선 발광 센서가 발광하고, 그와 짝을 이루는 수광 센서가 적외선의 반사량을 측정한다. 측정된 반사량을 기반으로 조향 모터를 제어한다. 그림 2은 무인 자율주행을 위한 센서부와 조향부를 설명한다.

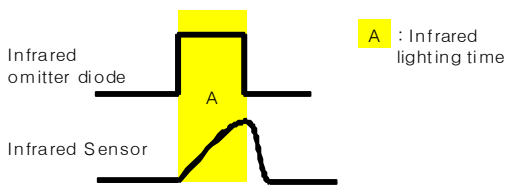


그림 3. 적외선 수•발광 센서의 전압 변화량
Fig. 3. Variation in voltage of infrared emitter diode and infrared sensor

적외선은 사람의 눈으로 확인할 수 없지만, 햇빛과 형광등,

등 대부분의 빛에 존재한다. 그래서 적외선 센서는 외란에 민감하다. 외란에 의해 뜻하지 않은 적외선이 수광될 경우 측정 결과는 신뢰할 수 없으며, 시스템에 치명적인 에러를 야기할 수 있다. 그래서 우리는 적외선 발광 센서에 주기를 더하여, 외란을 필터링하였다. 그림 3가 이를 설명한다.

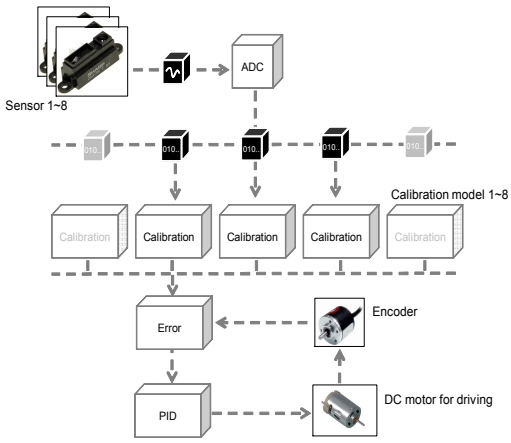


그림 4. 스마트 크루즈 컨트롤 시스템 다이어그램
Fig. 4. Smart cruise control system diagram

적외선 발광 센서가 계속 발광하지 않고, 정해진 주기를 갖고 점멸한다. 이에 맞추어 수광 센서가 정해진 주기로 필터링한다면 외란을 제거 할 수 있다. 이 방법을 발광 주기라고 명명한다.

발광 주기의 시간은 50 μs이다. 이 시간은 발광 센서가 미쳐 적외선을 발광하지 못 할 정도로 짧은 시간이다. 짧은 시간에 발광하지 못 하는 것을 해결하기 위해 과전류를 공급했고, 이는 조향부 센서가 갖는 데드라인으로 직결된다. 과전류를 오랫동안 공급할 경우 센서가 타기 때문에 엄격한 스케줄링이 필요하다.

1.2. 스마트 크루즈 컨트롤

본 연구를 수행하기 위하여 우리는 지능형 자동차의 기능으로 스마트 크루즈 컨트롤을 구현하였다. 스마트 크루즈 컨트롤이란 기본적으로 차량의 속도를 정속 주행하면서 선행 차량의 속도에 따라 안전거리를 유지할 수 있도록 자신의 속도를 자동 제어하는 시스템이다.

속도 제어를 하기 위해 적외선 거리 센서와 인코더(encoder)를 이용한다. 인코더는 차량의 속도를 측정하여 제어부에 전달한다. 제어부는 기본적으로 정속 주행을 위한 피드백 제어를 수행한다. 그리고 적외선 거리 센서가 측정한 선행차량과의 거리와 인코더가 측정한 차량의 속도로 선행 차량의 속도를 예측할 수 있다. 다시 선행 차량의 속도를 이용하면, 안전 거리를 계산할 수 있고, 정속 주행을 위한 제어부는 안전 거리와 제동 거리를 참고하여 주행 속도를 조절한다. 타이어 마찰력과 같은 외부 영향은 사전 분석 결과를 이용한다.

안전거리는 제동거리보다 크거나 같아야 하므로 선행 차량과의 거리와 제동 거리를 실시간 비교하면 스마트 크루즈 컨트롤 시스템을 구현할 수 있다. 그림 4이 스마트 크루즈 컨트롤 시스템을 보여준다.

안전거리는 선행 차량과 멀리 떨어질수록 좋지만, 상식적으로 이런 결과를 원하는 것이 아니므로, 우리는 제동 거리에서 반 바퀴 길이를 더한 거리를 안전거리로 계산했다.

올바른 동작을 위해, 피드백 제어를 했으며, 차량 속도, 선행 차량과의 거리, 선행 차량의 속도, 안전거리를 파라미터로 사용했다. 단, 이 값들의 측정 시간 오차가 메인 주기보다 작아야 한다. 만약 한 스텝(step)씩 밀리게 된다면, 과거의 상태를 반영하므로 오차가 발생한다. 그러므로 스마트 크루즈 컨트롤 시스템은 피드백 제어가 메인 주기보다 작을 것을 시간 제약성으로 요구하고 있다.

2. 실시간 프로세스 제어를 위한 제안

앞서 언급했듯이 제안한 지능형 모형차량 시스템은 두 가지의 강한 시간 제약성을 갖는다. 하나는 무인 자율주행을 위한 적외선 센서의 발광 주기 프로세스이고, 다른 하나는 스마트 크루즈 컨트롤을 위한 피드백 제어 프로세스이다. 두 가지의 시간 제약성을 보장하기 위해 우리는 실시간 스케줄러를 순환실행체제를 이용하여 설계하였다.

2.1. 동적분석

각 프로세스의 시간 속성을 계산하기 위해, 동적 분석을 사용하였다. 모터 제어의 특성상 수식 계산이 들어갈 뿐, 프로그래밍 언어의 조건문과 반복문이 복잡하게 구성되지 않는다. 그래서 동적 분석으로 얻은 프로세스 실행 시간의 신뢰도가 높다. 반면, 정적 분석은 그 복잡도 때문에 본 연구 목적인 쉽게 적용 가능한 스케줄러 설계와 상반된다.

무인 자율주행을 위한 적외선 센서의 발광 주기 프로세스는 $50\,\mu s$ 마다 발생한다. 이 프로세스의 역할은 알고리즘 1에서 설명한다. 이전에 발광했던 센서와 짝을 이룬 수광 센서의 ADC를 동작한다. ADC 샘플링 시간은 매우 짧게 설정한다. 그리고 발광 센서를 소등하고, 다음 발광 센서를 점등한다. 앞서 언급했듯이 과전류가 흐르기 때문에 시간 지연이 있을 경우 시스템 오류가 발생한다. 그러므로 데드라인을 짧게 설정하여 매 $50\,\mu s$ 마다 수행할 수 있도록 해야 한다. 제안한

시스템은 데드라인이 $5\,\mu s$ 이고, 실행시간이 $3\,\mu s$ 이다.

스마트 크루즈 컨트롤을 위한 센서와 인코더 처리 프로세스는 $400\,\mu s$ 를 주기로 갖는다. 이 값은 메인 주기(major cycle)로도 사용한다. ADC는 상시 동작하여, DMA를 통해 5개의 거리센서 값을 얻는다. 이 프로세스는 DMA로 동작하기 때문에 굳이 페치(fetch)할 필요가 없다. 그러므로 데드라인과 실행시간을 갖지 않는다. 이 과정에서 5개의 센서 값을 모두 사용하지 않고, 가중치를 조절하는 제어 알고리즘을 사용하지만, 본 논문에서는 자세히 다루지 않는다.

위 두 프로세스 외에 제어 연산 프로세스가 있다. 이 프로세스 역시 엄격한 시간 제약성을 만족하는데, 다른 두 프로세스와 다르게 시간 제약성을 불만족하였을 때, 시스템의 오류가 발생하지 않는다. 그러나 성능에 직접적인 영향을 미친다. 정도에 따라 기능을 전혀 수행하지 못 할 수 있다.

제어 연산 프로세스의 주기는 메인 주기에 따라 $400\,\mu s$ 이다. 제어 수식의 구현 방법에 따라, 많은 실수 연산과 나눗셈 등을 포함하여, 사용 효율(utilization)이 1을 초과한다. 이는 실시간 스케줄링 사전 조건을 위반한 것이므로, 스케줄링이 불가능하다. 제안한 시스템은 실수 연산을 전부 없애고, 최소한의 나눗셈을 하도록 제어 알고리즘을 수정하였다. 자세한 수정내역은 본 논문에서 다루지 않는다.

2.2. 오프라인 스케줄링

동적 분석 결과에 따라, 스케줄러가 동작하는 시점을 결정해야 한다. 온라인 스케줄링은 프로세스가 동작하면서 매번 스케줄링을 다시 한다. 메인 프로세스와 함께 스케줄러가 호출되기 때문에 복잡도, 우선순위 역전 현상, 오버헤드 등 예기치 못한 오류가 발생한다.

본 연구에서는 보다 적합한 오프라인 스케줄링을 사용한다. 제어 알고리즘과 센서 구조 상 프로세스 순서가 바뀌지 않는다. 또, 굳이 온라인 스케줄링을 사용하여 불필요한 오버헤드를 만들고 복잡도를 높일 필요 없으므로, 오프라인 스케줄링이 적합하다.

2.3. 순환실행체제

오프라인 스케줄링으로 설계된 시스템을 가장 이상적으로 구현할 수 있는 기법은 순환실행체제이다. 기존 연구는 대부분 EDF (Earlist Deadline First) 스케줄러를 탑재한 실시간 운영체제를 이용하여, 실시간 시스템을 구현하였다. EDF는 최적화(optimal) 알고리즘으로 시간 제약성을 가진 프로세스를 스케줄링하기에 효율적인 방법임에는 틀림없다. 하지만 전문 지식이 필요하며, 구현을 위한 사전 설계와 추가 인력, 추가 시간이 필요한 만큼 어려움이 있다.

Algorithm 1 Unmanned autonomous driving process

1: Enable ADC which is pair of a emitted sensor turned on
2: Turn off the emitted sensor
3: Turn on the next emitted sensor

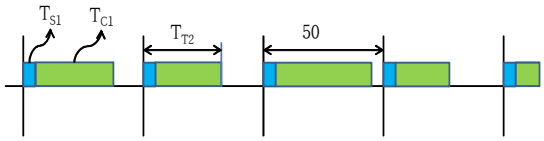


그림 5. 프로세스의 시간 다이어그램
Fig. 5. Timing diagram of processes

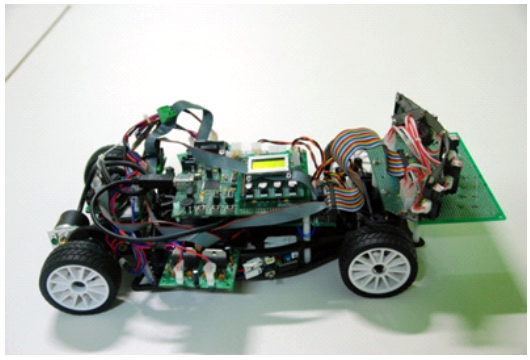


그림 6. 지능형 모형차량
Fig. 6. Intelligent model vehicle

그림 7. 실험용 차로
Fig. 7. Special road for evaluation

우리가 사용한 순환실행체제는 프로그래밍 구조가 복잡하지 않고, 기존 구현 방법에서 많은 수정을 요구하지 않기 때문에 우리의 목표에 적합한 기법이다. 또한, EDF와 다르게 오버헤드를 최소화하여, 시간 제약성을 만족하는데 기여한다.

순환실행체제의 시간적 기준은 프로세서 내부 타이머를 사용하였다. 타이머는 보조 주기(minor cycle) 마다 인터럽트를 발생시킨다.

보조 주기는 $50\,\mu s$ 이다. 이는 발광 주기 프로세스와 동일한 주기이다. 부가적으로 설명하면, 보조 주기마다 발광 주기 프로세스를 실행한다. 발광 주기 프로세스가 제어하는 센서는 총 8개이므로 전부 실행하기 위해서는 $400\,\mu s$ 가 필요하다.

앞서 말했듯이, 메인 주기는 $400\,\mu s$ 인데, 이는 발광 주기 프로세스가 모든 센서를 한번씩 제어하는 시간과 동일함을 알 수 있다. 또한 발광 주기 프로세스의 실행 시간은 $3\,\mu s$ 이므로, 남은 여유시간(laxity, slack time)이 상대적으로 많다. 우리는 이 시간에 스마트 크루즈 컨트롤 프로세스와 제어 연산 프로세스를 적절하게 분리하여 배치하였다. 그림 5는 이 과정을 도식화 하고 있다. 이 그림에서 T_{Sn} 이 발광 주기 프로세스이며, T_{Cn} 이 분리된 스마트 크루즈 컨트롤 프로세스 또는 제어 연산 프로세스의 조각이다. T_{Sn} 와 T_{Cn} 은 총 8쌍 존재하므로, n 은 1부터 8까지 존재한다.

이와 같이 순환실행체제는 메인 주기와 보조 주기만 결정한다면, 타이머만으로 구현이 가능하다. 실시간 스케줄링 하지 않은 방법도 타이머 한 개 이상 사용하는 경우가 많으므로, 이와 같이 설계한다면 약간의 수정으로 신뢰도 높은 시스템을 설계할 수 있다.

IV. 실험 및 결과

무인 자율주행과 스마트 크루즈 컨트롤은 모형차량으로 구현하였다. 실제 차량과 유사한 모형을 하고, 임베디드 시스템으로 제어하기 용이하다. 또, 소규모 지능형 자동차 연구에 많이 사용한다. 그림 6이 실험을 위해 개발한 모형 차량이며, 사용한 MCU (Micro-Controller Unit)의 스펙은 표 1과 같다.

무인 자율주행을 효율적으로 구현하기 위해 특수 차로를 설계하였다. 흰색 바탕에 검은색 라인이 주어지며, 검은 라인을 차로의 중앙으로 인식한다. 그림 7과 같으며, 회전 반경 및 곡률은 가변적이다. 차로와 수직으로 그려진 두 점선은 실험 시 사용된 출발선이다.

표 1. MCU 사양
Table 1. MCU Spec.

MCU	MC9S12XEP100
Manufacturer	Freescale
Architecture	S12X, XGATE
Clock	50 MHz
Flash	1 MB
RAM	64 kB
ADC	12 bit resolution, convertion time, and 16 channel

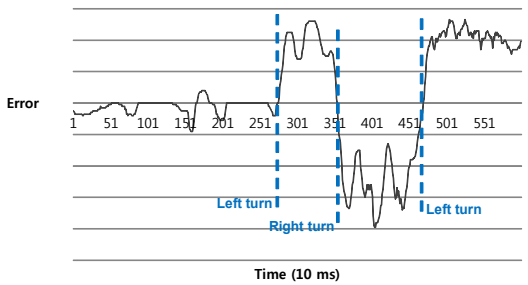


그림 8. PID 에러
Fig. 8. PID Error

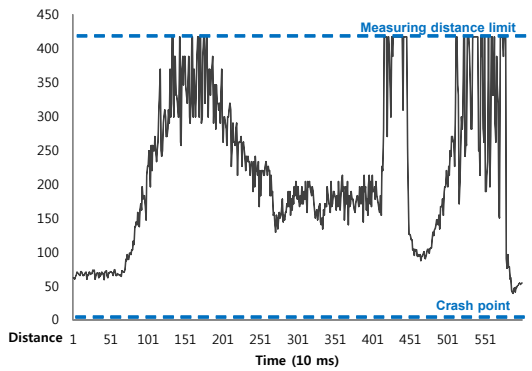


그림 9. 선행차량과의 거리 측정
Fig. 9. The distance between vehicles

무인 자율주행과 스마트 크루즈 컨트롤의 구현은 성공적이다. 그림 8이 무인 자율주행의 실험 결과이다. 무인 주행을 위해서 PID 제어(Proportional-Integral-Derivative Controller)를 하였다. PID 제어는 연산을 통해 에러 값을 출력한다.

PID 에러는 차량의 진행방향을 의미한다. 차로와 평행한

직선 주행을 할 때 에러 값은 0이다. 차로를 기준으로 왼쪽으로 회전하면 에러 값은 양수로 변한다. 값의 크기는 차로로부터 떨어진 거리와 비례한다. 반대로 오른쪽으로 회전하면 에러 값은 음수로 변한다.

출발 시 차체가 차로와 완벽하게 평행할 수 없으므로, 직선 코스에서도 약간의 에러가 있다. 그 후 회전 방향에 따라 에러가 커지고, 이를 제거하기 위해 조향하는 모습을 볼 수 있다.

그림 9은 스마트 크루즈 컨트롤의 실험 결과이다. 표현된 값은 선행 차량과의 거리이다. 최소 유지거리는 150으로 설정하였다. 이 설정 값은 시스템을 위한 값이기 때문에, 그래프의 세로 축과 최소 유지거리의 단위는 표준 단위가 아니다. 최소 유지거리 150은 약 10 cm 정도 이다.

출발 시 최소 유지거리보다 가까우므로 출발하지 않는다. 그 후 선행 차량이 출발하면서 거리가 최소 유지 거리보다 멀어지면 차량이 출발한다. 사용한 센서는 최대 2 m 까지 측정할 수 있다. 다시 말해, 측정한 값은 최대 400까지 나오는데, 1.7초, 4.2초, 5.3초 구간과 같이 갑자기 측정한 값이 한계에 도달하는 것은 선행 차량이 회전하여, 시야에서 벗어났기 때문이다. 3초에서 4초 구간을 보면 최소 유지거리를 유지하면서 따라가는 것을 볼 수 있다.

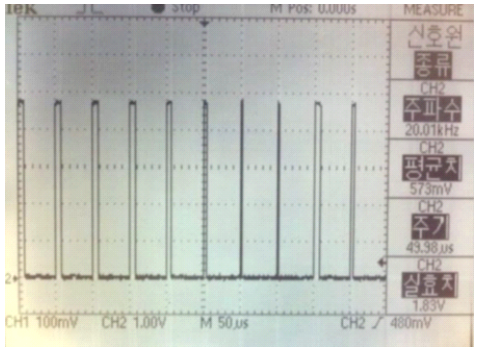


그림 10. 스케줄링 차트
Fig. 10. Schedulling chart

무인 자율주행과 스마트 크루즈 컨트롤이 성공적으로 동작함에 따라 프로세스가 정상적으로 스케줄링 되었다고 판단할 수 있으나, 객관적인 자료로 입증하기 위해 별도의 실험을 하였다. 프로세스가 시작하고 끝 날 때 전기적 신호를 출력하였다. 그림 10이 그 결과이다. 모든 신호는 $50\mu s$ 마다 상승 엣지(rising edge)를 보이고, $50\mu s$ 전에 하강 엣지(falling edge)를 보이는 것을 알 수 있다. 다시 말해, 모든 프로세스가 보조 주기마다 실행되고, 다음 주기가 오기 전에 종료하는 것을 확인할 수 있다. 즉, 스케줄링을 정상적으로 하고 있다.

무인 자율주행과 스마트 크루즈 컨트롤 기능이 실험적으로 동작함을 보이고, 프로세스를 스케줄링한 결과를 입증함으로써, 제안한 지능형 모형차량이 실시간 시스템임을 입증하였다. 또한, 사용한 스케줄링 알고리즘이 복잡하지 않고, 기존 방법에 약간의 수정과 사전 분석으로 실시간 스케줄링을 할 수 있음을 증명하였다.

제안한 시스템을 실제 지능형 자동차에 적용하기에는 아직 한계가 있다. 실제 지능형 자동차에 탑재되는 더욱 많은 기능과 같이 스케줄링 하기에는 순환실행체제만으로는 너무 복잡하다.

V. 결 론

본 연구는 무인 자율주행과 스마트 크루즈 컨트롤 기능을 모형차량에 구현하고, 이에 적합한 순환실행체제 기반 스케줄러를 동적 타이밍 분석 결과를 바탕으로 제안하였다. 또한 제안한 순환실행체제가 경미한 오버헤드로 성공적으로 무인 크루즈컨트롤 기능을 성공적으로 수행할 수 있음을 모의 무인주행을 통해 확인하였다.

우리는 자동차 연구의 안정성과 신뢰성을 향상시키기 위하여 오프라인 실시간 스케줄러의 하나인 순환 실행체제를 모형차량에 적용하였다. 그러나 실제 지능형 자동차는 더욱 많은 기능을 탑재하고 있으므로 스케줄링 환경이 보다 복잡하고 가변적이다. 따라서 연산 복잡도가 다소 높더라도 보다 정교한 프로세스 제어가 가능한 동적 실시간 스케줄링 알고리즘이 고려될 수 있으며, 이에 대한 연구가 추가 연구가 필요하다.

참고문헌

[1] H. H. Chin, A. A. Jafari, "Intelligent Hybrid Vehicle Management Systems," SSST, pp. 27-34, Mar. 2013.

[2] J. Choi, E. Jee, H. Kim, D. Bae, "A Case Study on Timing Constraints Verification for A Safety-Critical, Real-Time System," KCI, Vol. 38, No. 1, pp. 166-169, Jun. 2011.

[3] F. Chen, H. Qiu, Y. Gao, "Freescale Single-chip Microcomputer Intelligent Car Voltage Control Discussed," ICDMA, pp. 430-433, Aug. 2012.

[4] S.-J. Jang, "A Study of Real-Time Scheduling

Algorithms for Automotive System," JKIICE, pp. 1363-1370, Jul. 2009.

[5] C. L. Liu, J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," Journal of the ACM, vol. 20, no. 1, pp. 46-61, Jan. 1973.

[6] T. P. Baker, A. C. Shaw, "The Cyclic Executive Model and Ada," Real-Time Systems, pp. 7-25, 1989.

[7] A.-L. Mok, "Fundamental design problems of distributed systems for the hard-real-time environment," Ph.D. Dissertation, M.I.T., Cambridge, 1983.

[8] S. Choi, S. Kim, J. Choi, H. Kim, "The study of the linetracer-development for the Automatic Guided Vehicle using Infrared," KIEE, pp. 1967-1969, Jul. 2003.

저 자 소 개

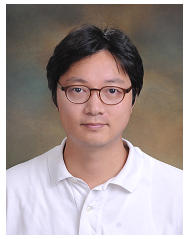
이 재 면

2012: 한양대학교 컴퓨터공학과 공학사

현 재: 한양대학교 컴퓨터정보과 석박사통합과정

관심분야: Storage System, Energy Efficiency, IoT

Email : jaemyoun@hanyang.ac.kr



강 경 태

1999: 서울대학교 전산과학전공 공학사

2001: 서울대학교 컴퓨터공학부 공학석사

2007: 서울대학교 컴퓨터공학부 공학박사

2008: Coordinated Science Lab. University of Illinois at Urbana-Champaign 박사후연구원

2010: Dept. Computer Science, University of Illinois at Urbana-Champaign 박사후연구원

현 재: 한양대학교 컴퓨터공학과 조교수

관심분야: Cyber Physical Systems, Mobile Computing

Email : ktkang@hanyang.ac.kr

노 동 건

2000: 서울대학교 컴퓨터공학과 공학사

2002: 서울대학교 전기컴퓨터공학부 공학석사

2007: 서울대학교 전기컴퓨터공학부 공학박사

현 재: 숭실대학교 정보통신전자공학부 조교수

관심분야: Embedded System, Mobile Computing, Ubiquitous Sensor Network

Email : dnoh@ssu.ac.kr