

목표물 추적 알고리즘을 적용한 DCU 기반 차량용 네트워크 플랫폼 구현

홍형근¹, 박재범^{2*}, 도영수², 전재욱²¹성균관대학교 소프트웨어융합대학²성균관대학교 정보통신대학whaihong@g.skku.edu, pjb3873@g.skku.edu, cok2529@naver.com, jwjeon@skku.edu

Implementation of a DCU-based vehicle network platform using target tracking algorism

Hyeong-Keun Hong¹, Jae-Bum Park², Young-Soo Do², Jae-Wook Jeon²¹College of Computing and Informatics, Sungkyunkwan University²College of Information & Communication Engineering, Sungkyunkwan University

요 약

최근 차량 플랫폼의 발달로 인하여 차량 내 ECU 간의 정보교환이 증가하였다. 기존의 센트럴 게이트웨이 아키텍처 네트워크에서는 늘어난 정보교환으로 인한 과부하가 발생하였다. 따라서 ECU 들끼리 Domain 을 형성하도록 하여 DCU 를 활용한 Domain 기반 차량용 통신 아키텍처가 해법으로 제시되었다. 본 연구에서는 DCU 와 ECU 간의 네트워크 플랫폼을 소형 전동차를 이용하여 제작한다. 이 네트워크를 검증하기 위하여 목표물 추적 알고리즘을 통해 네트워크가 정상 작동하는지 확인한다.

1. 배경

최근 자동차 산업에서는 탑승자의 편의 및 안전을 고려한 자율주행 자동차에 관한 연구 및 개발이 한창이다. 지속적인 연구의 성과로, 현재 출시되고 있는 자동차들은 외부 상황을 파악하기 위한 카메라, 라이다와 같은 다양한 센서를 통해 운전자에게 ADAS(Advanced Driver Assistance System)를 제공한다.

차량에서 활용하는 센서 및 기능들이 다양해짐에 따라, 차량 내 ECU(Electric Control Unit)의 정보교환이 많이 증가하였다. 이에 따라 (그림 1)의 전통적인 차량용 네트워크 시스템은 늘어난 정보교환을 처리하기에 부하가 늘어난다는 문제가 발생한다. [1] 전통적인 아키텍처에서는 센트럴 게이트웨이가 모든 ECU 의 제어를 담당하며, 내부 통신으로는 주로 CAN 을 사용하고 있다.

CAN 은 UTP 구리 선을 사용하여 노이즈에 강하고 CRC 방식을 채택해 Error Detection 에 강점이 있으나, 전송속도가 최대 1Mbps 이고, 한 프레임당 최대 8 바이트의 정보만 전송이 가능하기 때문에 센서들의 많은 데이터를 주고받기에는 느리다. 따라서 ECU 의 개수가 늘어난 최신행 차량 플랫폼에서 (그림 1)의 시스템을 활용한다면 느린 CAN 의 속도와 중앙에 집중된 아키텍처로 인하여 센트럴 게이트웨이에 부하가 발생한다.

이러한 문제점을 해결하고자 (그림 2)의 Domain 아키텍처가 제안되었다. Domain 아키텍처는 비슷한 기능을 수행하는 여러 ECU 를 하나의 Domain 으로 묶어서 DCU(Domain Control Unit)가 처리하는 방식이다. 각 DCU 가 관리하는 Domain 에서는 ECU 개수가 적으므로, 게이트웨이와 DCU 의 부하가 감소한다는 장점이 있다. DCU 간의 통신은 차량용 Ethernet 을 사용하여 속도를 개선한다.

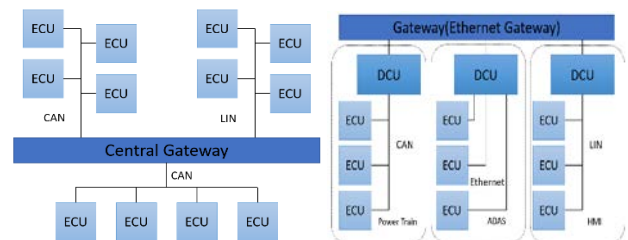


그림 1) 센트럴 게이트웨이 아키텍처

그림 2) Domain 아키텍처

이에 따라 본 연구는 DCU 기반의 최신행 차량용 네트워크 구현 및 검증을 목표로 한다.

2. 구현

2.0 플랫폼

본 연구에서는 (그림 3)의 소형전동차를 활용하여 (그림 4) 와 같이 네트워크 플랫폼을 설계하였다. 구

현한 모듈은 ADAS Domain 에서의 인지, Ethernet 과 CAN 을 활용한 통신, Arduino 를 활용한 모터제어로 크게 세 가지로 나눌 수 있다.



그림 3) 소형 전동차

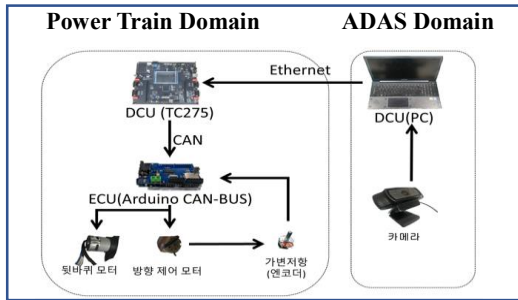


그림 4) 구현한 DCU 기반 네트워크 플랫폼

2.1 인지

컴퓨터 시스템은 Object Detection 을 활용하여 입력 센서로 카메라만을 가지고도 자동차를 제어할 수 있다. Object Detection 의 대표적인 방법 중 하나인 YOLO(You Only Look Once)는 카메라에서 이미지를 불러오고, 객체를 감지하여 화면에 출력하는 시간까지 모두 고려했을 때 뛰어난 실시간 성능을 보여준다. [2] 이에 따라 본 연구에서는 카메라를 입력 센서로 활용하여 YOLO 기반 Object Detection 을 구현하였다. 카메라는 640*480 크기의 프레임을 10(Frame/Second)의 속도로 받아들이도록 하였다. YOLO 는 카메라를 통해 프레임을 받아들일 때마다 사전에 학습된 사물들을 감지하여 해당하는 사물의 둘레에 Bounding Box 를 표시해준다.

본 연구에서는 빨간색 상의를 입은 사람을 추적할 목표물로 지정하였다. 빨간색을 감지하기 위해서 BGR(Blue, Green, Red) 픽셀값을 HSV(Hue, Saturation, Value)로 변환한 뒤, 색조(Hue) 값이 빨간색을 띠는 범위인 0 이상 10 미만, 171 이상 180 미만의 자연수 범위를 가지는 픽셀을 제외하고는 명도(Value) 값을 모두 0 으로 처리하여 빨간색을 제외하고는 모두 검게 나타냈다.

이후, YOLO 가 사람을 감지하면 사람의 둘레에 표시된 Bounding Box 내부의 위쪽 절반에 해당하는 넓이 중, 붉은색으로 표시된 비율이 10% 이상이 되면 ‘빨간색 사람’이라고 인식하도록 설정하였다. ‘빨간색 사람’을 감지하지 못하면 속도(velocity)를 0 으로, ‘빨간색 사람’을 감지하면 속도(velocity)를 0 보다 크고 16 보다 작은 hyper parameter (V) 값으로 설정하였다. 방향(direction)은 Person box 내부의 위쪽 절반에서 빨간색 비율이 가장 높은 사람의 중앙 픽셀의 x 좌표(target) 과 전체 가로 픽셀 수(width)를 이용하여 아래 수식과 같이 계산한 뒤, 반올림을 취해주도록 하였다.

본 연구에서는 640*480 크기의 프레임을 활용하므로, direction 값은 -15 이상, 15 이하의 정수가 된다.

$$direction = \frac{target - (width * 0.5)}{20}$$

이를 요약하면 (그림 5)에 나타난 순서도와 같다.

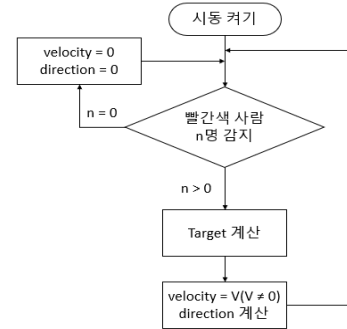


그림 5) 목표물 추적 알고리즘

2.2 통신

통신 모듈에서는 ADAS Domain 에서 계산한 결과에 부합하는 제어 명령을 Ethernet 을 통하여 TC275 로 송신해야 한다. 이때, Ethernet 프레임 내의 데이터 영역에 필요한 제어 명령 규약이 필요하다. Ethernet 프레임 내 데이터 영역은 최소 46 바이트를 차지하는데, 본 연구에서는 데이터 영역의 최초 5 바이트만을 활용한다. 최초 4 바이트는 제어정보, 5 번째 바이트는 제어정보의 끝을 알린다. 데이터영역 내의 제어 정보 4 바이트는 (표 1)과 같이 구성하였다.

표 1) Ethernet 데이터 영역 최초 4 바이트의 제어정보

Byte	기능	데이터 해석 방법
0	전진 속도	
1	후진 속도	
2	좌측으로 핸들 꺾는 각도	
3	우측으로 핸들 꺾는 각도	

아스키코드로 0x30 부터 0x39 까지인 ‘0’부터 ‘9’까지와, 아스키코드로 0x41 부터 0x46 까지인 ‘A’부터 ‘F’까지, 총 16 개 값을 활용하여 V 값과 direction 값을 16 진수로 나타냈다. direction 값이 양수일 때는 좌측 각도 값을 ‘0’으로, 우측 각도 값을 direction 으로 설정하였다. direction 값이 음수일 때는 좌측 각도 값을 direction 의 절댓값으로, 우측 각도 값을 ‘0’으로 설정하였다. 제어정보의 끝을 알리는 5 번째 바이트는 아스키코드로 0x66 인 ‘f’로 설정하였다.

TC275 에서 Arduino 로 CAN 을 송신하기 위해서도 CAN 데이터에 관한 규약이 필요하다. 본 연구에서는 CAN 데이터 규약을 Ethernet 데이터 규약과 동일하게 구성하였다. 따라서 TC275 는 수신한 Ethernet 의 제어 명령 5 바이트만을 CAN 헤더와 함께 Arduino 로 송신한다.

2.3 제어

CAN 프레임을 수신한 Arduino 는 프레임 내의 제어 명령을 이용하여 모터를 제어하는 ECU 역할을 수행해야 한다. 하지만 Arduino 본체에는 CAN 데이터를 받아 CAN 프레임으로부터 데이터를 추출할 수 있는 CAN Transceiver 가 없으므로, (그림 4)에 나타난 것과 같이 Arduino CAN-BUS 쉴드를 Arduino 에 장착한 후 사용하였다. Arduino 는 CAN 데이터 제어 규칙에 따라 방향 제어 모터와 뒷바퀴 모터에 각각 알맞은 PWM 전압을 인가하여 동작시킨다.

모터는 Arduino 로부터 PWM 전압을 인가받아 동작한다. 뒷바퀴 모터는 feedback input 이 필요하지 않지만, 방향제어 모터의 경우에는 제어 명령 각도 정보와 기존에 핸들이 돌아가 있는 각도를 비교해야 정확한 제어가 가능하다. 이를 위해 핸들에 가변저항을 달아 핸들이 돌아간 각도에 따라 변하는 저항값을 Arduino 의 입력 핀으로 전달하였다.

3. 검증 및 결과

(표 2)는 ‘빨간색 사람’을 추적하는 알고리즘을 검증한 결과이다.(표 2)의 명령 신호에 나타난 단점값은 ADAS Domain 에서 계산한 direction 을 나타낸다.

표 2) 추적 알고리즘 검증 결과

case	화면 사진	명령신호 확대
빨간색 사람 없음		정지상태 정지상태 정지상태 정지상태
우측 빨간색 사람		5단계 우회전 5단계 우회전 5단계 우회전 5단계 우회전
좌측 빨간색 사람		5단계 좌회전 5단계 좌회전 5단계 좌회전 4단계 좌회전
빨간색 사람 다수		7단계 우회전 7단계 우회전 7단계 우회전 6단계 우회전 6단계 우회전

차량 앞에 ‘빨간색 사람’과 그렇지 않은 사람이 서 있을 때, ‘빨간색 사람’을 추적하는 방향으로 제어신호를 정상적으로 출력하는 것을 확인하였다. 다수의 ‘빨

간색 사람’이 서 있을 때도 빨간색 비율이 높은 사람 쪽 방향으로 제어신호를 정상적으로 출력하는 것을 확인하였다. 또한, 차량의 전원을 켜고 실제로 추적 동작을 하는지 확인하였을 때, 정상적으로 전동차가 ‘빨간색 사람’을 추적하는 방향으로 움직였다는 것을 확인하였다. 빨간색 사람이 야외에서 도로를 활보하여 차량이 추적하는지 검증하는 실험 또한 성공적으로 이루어졌다.

4. 결론

본 연구에서는 DCU 기반 차량용 네트워크 플랫폼을 소형 전동차를 활용하여 구현하였다. Domain 은 ADAS(PC)와 Power Train(TC275)으로 구성했다. 목표물 추적 알고리즘을 통해 차량 내 통신 기술인 CAN 과 Ethernet 이 DCU 및 ECU 간 통신을 성공적으로 수행했음을 검증하였다.

다만, 본 연구에서는 실제 상용화된 크기의 전동차를 활용하지 않았다는 점에 한계가 있다. 근래의 차량 플랫폼에는 ECU 의 개수가 100 개가 넘는데, 본 연구에서 소형전동차에 적용한 ECU 및 DCU 는 고작 3 개에 불과하다. 또한, 본래 Domain 기반 아키텍처에 존재하는 게이트웨이를 구현하지 않았다는 점에서도 한계점이 존재한다. (그림 2)에서는 DCU 간을 연결하는 게이트웨이가 존재하나, (그림 4)에서는 DCU 간의 연결에 게이트웨이가 존재하지 않는다.

따라서 본 연구의 의의는 실제 차량에 적용되는 것 아니라 제한된 하드웨어 환경 내 DCU 기반 차량용 네트워크 플랫폼을 구현했다는 점에 있다. 최신형 차량용 네트워크 시스템을 간소화하여 소형 플랫폼에 적용한다면 다양한 산업 분야에 확장이 가능할 것이다. 여기에 목표물 추적 알고리즘까지 결합한다면 더욱 확장성이 높아질 것이다. 예로, 이 플랫폼을 소형 동물 로봇에 응용하여 주인을 따라다니는 반려로봇을 개발하거나, 역으로 목표물을 피하는 알고리즘으로 응용하여 산업용 기계가 사람을 피하도록 하여 안전 사고를 예방하는 등의 다양한 활용이 가능할 것이다.

5. Acknowledgement

이 연구는 정부(교육부-산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임 (P0022094, 2022 년 미래형자동차 기술융합 혁신인재양성사업)

참고문헌

[1] Sugihara, Makoto, and Akihito Iwanaga. "Minimization of FlexRay Bus Bandwidth for Hard Real-Time Applications." *Journal of Information Processing*, vol. 21, no. 1, 2013, pp. 46–52., <https://doi.org/10.2197/ipsjip.21.46>.

[2] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.