

# 장애물 극복이 가능한 구조로봇의 주행모드 변형을 위한 PI-based Feedforward 제어

## PI-based Feedforward Control for Driving Mode Transformation of Rescue Robot capable of Obstacle Overcoming

정해관\*, 강현석, 곽윤근  
(Hae Kwan Jeong, Hyun Suk Kang, and Yoon Keun Kwak)

**Abstract :** This paper offers a practical control scheme for driving mode transformation of a rescue robot already developed. The rescue robot, VSTR(Variable Single-Tracked Robot), has two driving modes, so can traverse untidy terrain and overcome obstacles such as stairs easily by use of timely driving mode transformation. Classical PI control scheme was used firstly for driving mode transformation, but stationary phenomenon, which might have a bad effect on the performance in real situation, came into existence. Therefore, we suggest a new controller, PI-based feedforward controller, which should be a good alternative for the problem, and compare it with other nonlinear control scheme.

**Keywords :** driving mode transformation, PI-based feedforward control, stationary phenomenon

### 1. 서론

구조로봇(rescue robot)은 빌딩 붕괴나 화재와 같은 재난이 발생했을 때 추가 인명 피해를 차단하면서 보다 빠르고 효과적으로 구조작업을 수행하고자 고안된 특수목적용 로봇이다. 이러한 구조로봇은 20세기 이후 전세계적으로 주목 받고 있는 USAR(Urban Search and Rescue) 분야와 맞물려 실용성에 초점을 두고 꾸준히 개발되고 있다.

구조로봇이 임무를 수행하는 환경은 일반적으로 매우 많은 장애물을 포함하고 있다. 지형 역시 불균일하고 묘사하기 힘든 모양을 하고 있기 때문에 기존의 주행로봇처럼 이동성만을 강조하는 메커니즘으로는 실용성을 논하기 힘들다. 즉, 구조로봇은 혼잡한 지형을 안정성을 유지하며 통과할 수 있고, 수 cm 높이의 둔턱이나 계단 등의 실내 장애물을 극복할 수 있는 메커니즘을 내장해야 한다. 한편, 실제로 구조작업이 로봇에 의해서 수행된다면 조종자가 외부에서 여러 형태의 구조로봇을 동시에 통제하면서 임무를 완수하는 통합 운용 시스템이 구비되어야 할 것이다. 따라서 구조로봇 관련 연구는 재난 환경에 대한 정보와 통신 장비 등이 선행된다는 가정 하에 주파력(traversability)이 뛰어난 여러 형태의 구조로봇을 동시에 운용하는 것을 원칙으로 하면서, 상호보완적 특성을 갖는 다리(leg), 바퀴(wheel), 트랙(track), 단위(module) 형태의 메커니즘 설계 및 구현에 초점을 맞춰 진행 중에 있다.

구조로봇의 메커니즘은 대형에서 중소형으로, 보행형에서 바퀴형과 트랙형을 거쳐 단위형으로 진보하고 있다. 이 중 트랙형 구조로봇은 현재까지 가장 많은 비율로 시장을 점유

하고 있는 구조로봇으로, 구현 시 제어가 용이하고 실내/외를 막론하고 탁월한 주행 성능을 보장한다는 장점을 갖고 있다. 트랙형 구조로봇의 대표적인 사례로는 Matthies et al.에 의해 USAR 및 군사 활용 목적으로 개발된 Packbot(2000)을 들 수 있다[1]. Packbot은 2001년에 미국 뉴욕에서 발생한 911 테러 당시 붕괴된 세계무역센터 잔해 속에 투입되어 구조작업을 수행한 바 있고, 2004년에는 미군의 아프가니스탄 주둔 과정에서 사막 지형으로 이루어진 아군 기지 주변을 정찰, 수색하는 작업을 수행하기도 했다. 이 외에도 Micro-VGTV (Inuktun, 1995), Talon(Foster and Miller, 2000), Solem(Foster and Miller, 2000), Urbot(SPAWAR, 2001) 등의 트랙형 구조로봇이 탐색 및 환사이송의 구조작업을 보조한 기록이 있다[2]. 그러나 이들은 조종자의 원격조종으로 하나의 로봇만 제어한다는 점에서 한계를 보이며 지능 시스템 및 통합 운용 시스템 도입의 필요성을 각인시켰고, 작은 구멍이나 절벽 등을 통과할 수 있는 소형 구조로봇의 개발을 촉진시켰다.

본 연구에서는 보다 효과적인 구조작업을 위해 가변 형상 단일 트랙 메커니즘을 기본으로 설계된 VSTR(Variable Single-Tracked Robot)을 소개한다. VSTR은 그림 1과 같이 상하좌우로 대칭인 외형을 가지고 있으며 2개의 주행모드를 가지고 있어서 유사시 구동 메커니즘의 형상을 바꿀 수 있다.

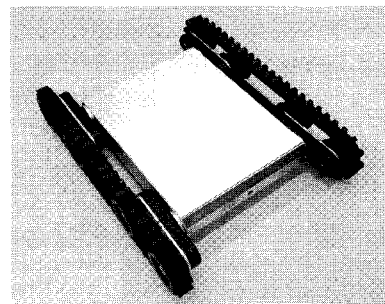


그림 1. 가변 형상 단일 트랙 메커니즘 기반의 VSTR.  
Fig. 1. Variable single-tracked mechanism based VSTR.

\* 책임저자(Corresponding Author)

논문접수 : 2007. 12. 11., 채택확정 : 2008. 2. 28.

정해관, 강현석 : 한국과학기술원 기계항공시스템학부 기계공학전공  
(hohip29@kaist.ac.kr/kangpoh@kaist.ac.kr)

곽윤근 : 한국과학기술원 기계항공시스템학부 교수  
(ykkwak@kaist.ac.kr)

※ 이 논문은 2007년도 2단계 두뇌한국(BK)21사업에 의하여 지원되었음.

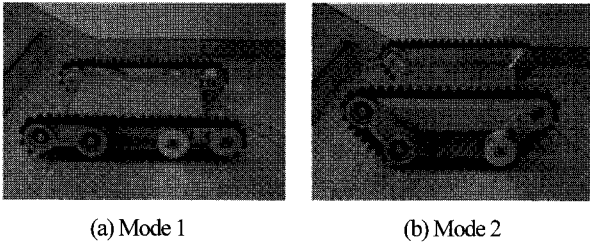


그림 2. VSTR의 2가지 주행모드.  
Fig. 2. Two driving modes of the VSTR.

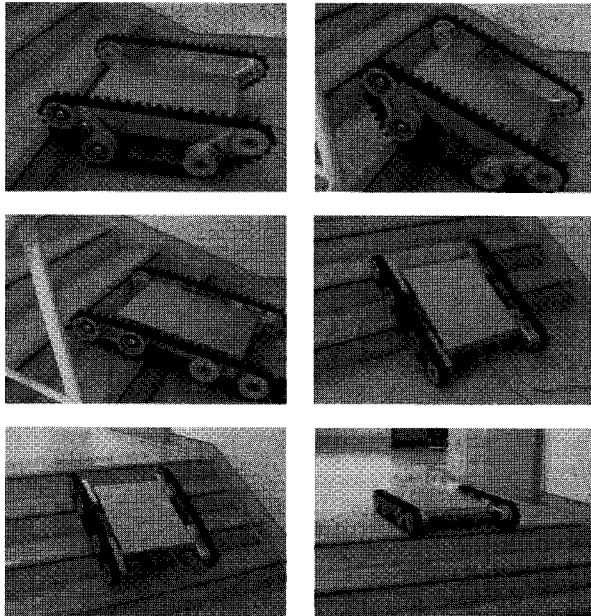


그림 3. VSTR의 계단 장애물 극복.  
Fig. 3. Overcoming stairs by the VSTR.

그림 2는 이러한 VSTR의 2가지 주행모드를 보여준다. Mode 1은 평상시 구동되는 VSTR의 주행모드로서 낮은 장애물이나 구멍 등을 비롯하여 각종 불균일 지형을 안정적으로 지나갈 수 있는 구조를 가진다. 반면 Mode 2는 로봇 좌우로 뻗어있는 암 모듈(arm module)이 회전하고 로봇의 높이가 증가하여 장애물 극복에 도움을 주는 45°의 입사각(attack angle,  $\theta$ ) 이 생성되는 주행모드이다[3]. 그림 3은 선행 연구에서 VSTR의 제작을 완료한 뒤 조종자 기반으로 2개의 주행모드를 이용하여 계단 장애물의 극복 가능 여부를 실험적으로 확인한 것이다[4].

따라서 VSTR은 재난 환경에 투입됐을 시 2개의 주행모드를 적절한 때에 번갈아 가며 선택함으로써 구조로봇의 본 목적을 달성할 수 있다. 그러나 2개의 주행모드가 서로 유연하게 교류되지 않으면 시간 지연이나 오작동 등의 난감한 상황에 봉착할 수 있다. 즉, 빠르고 안정적으로 Mode 1과 Mode 2가 변환되어야 갑작스러운 상황에 대해 로봇이 민첩하게 대응할 수가 있는 것이다.

이에 본 연구에서는 VSTR의 주행모드 변형이 보다 신속하고 안정적으로 이루어지도록 하기 위해 PI 제어를 기본으로 하는 동역학 모델 기반 제어 기법을 제안하고 기존의 비선형 제어 기법과 어떤 차별성이 있는지 논한다.

II. 정체 현상(Stationary Phenomenon)

선행 연구에서는 개발된 VSTR의 자율적인 장애물 극복을 구현하기 위해 PSD(Position Sensitive Detector) 센서를 이용한 장애물 탐지와 함께 자율 주행모드 변형을 중점적으로 다루었다[5]. 그러나 반복되는 실험으로부터 VSTR의 주행모드 변형 과정에서 그림 4와 같은 정체 현상(stationary phenomenon)이 발생함을 알 수 있었다. 정체 현상은 장애물이 탐지되어 주행모드가 Mode 1에서 Mode 2로 변형할 때 입사각의 목표점인 45°로 한 번에 수렴하지 못하고 중간 지점에서 한 번 정체했다가 이내 수렴하는 현상을 뜻한다[6].

이러한 정체 현상은 그대로 방치했을 때 로봇 성능에 악영향을 미칠 수 있다. 예컨대 입사각이 30°에서 한 번 정체했다가 45°로 수렴한다면 정체되는 구간에서 약 4cm의 극복 가능 높이를 손해 보게 된다. 따라서 주행 중인 로봇의 속도와 PSD 센서의 최대 탐지 거리를 감안한다면 극복 가능 장애물 임에도 불구하고 정체 구간으로 인해 극복하지 못하는 상황이 발생할 수 있다. 또한, 미소한 반응 시간차로 인해 극복이 가능하게 되는 경우일지라도 극복 과정에서 시간 지연 현상이 발생할 수 있다. 이는 올바른 장애물 탐지와 주행모드 결정이 선행되더라도 실제로 그 뜻을 반영하지 못하는 결과를 낳게 되는 것이기 때문에 VSTR의 실용성을 위해 반드시 해결해야 하는 문제이다.

정체 현상의 원인은 중력에 있다. VSTR의 주행모드 변형은 중력을 거스르고 수행되는 작업이다. 따라서 초기 위치부터 원하는 목표 위치까지 변위시키기 위해서는 중력에 해당하는 만큼의 힘을 보상해줘야 한다. 선행 연구에서는 주행모드 변형을 위해 PI 제어를 사용했다. VSTR의 주행모드 변형 과정은 비선형이기 때문에 가장 간단하면서도 원하는 성능에 근접하기 쉬운 PID 제어를 1차적으로 고려했고, VSTR 내부에 있는 부품 간의 마찰 등에 의해 이미 감쇠 효과(damping effect)가 충분하다는 판단 하에 D gain을 제외한 PI 제어를 최종적으로 선택했다. 그러나 PI 제어 자체에는 중력을 보상해주는 요소가 없기 때문에 이로 인한 일시적인 오차가 발생하게 되고, 이러한 오차 발생 현상이 정체 현상으로 나타나게 되는 것이다.

따라서 본 연구에서는 중력으로 인한 오차를 보상할 수 있는 PI-based feedforward 제어 기법을 대안으로 제시하고 이를 검증하였다. 검증 과정은 주행모드 변형을 운동 방정식으로 정의한 뒤, 이를 기반으로 한 모의실험으로부터 제안한 PI-

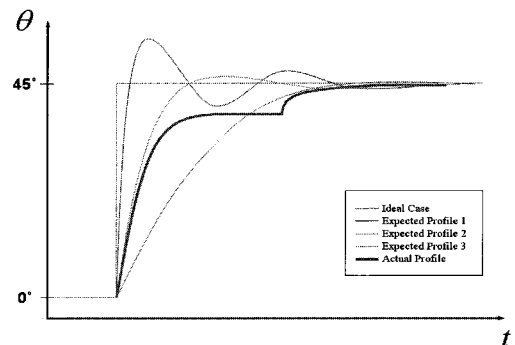


그림 4. 주행모드 변형 중 나타나는 정체 현상.  
Fig. 4. Stationary phenomenon in driving mode transformation.

based feedforward 제어와 기존의 PI 제어, 그리고 다른 비선형 시스템 제어 기법들을 비교하는 방법으로 진행된다.

III. 주행모드 변형에 대한 운동 방정식

1. 설계 변수 정의

운동 방정식을 유도하기 위해서는 우선적으로 시스템의 설계 변수를 정의해야 한다. 그림 5는 Mode 1과 Mode 2에서 정의되는 VSTR의 외형 치수를 나타낸다. 한편, 관성 모멘트 (moment of inertia)를 정의하기 위한 질량 특성(mass properties)과 입력 토크(input torque) 등은 표 1과 같이 정의된다. 이는 그림 6에 나타나있는 암 모듈의 회전을 배경으로 한다.

2. 운동 방정식

운동 방정식의 유도는 Lagrange의 운동 방정식(Lagrange's equations of motion)을 이용한다[7]. 주행모드 변형에 있어서 우리가 관심을 갖고 있는 부분은 Mode 1에서 Mode 2로 변할 때이므로 이 과정에서 발생하는 운동 에너지  $T$ 와 위치 에너지  $V$ 를 다음과 같이 정의할 수 있다.

$$T = \frac{1}{2}(M + 4m)\left(\frac{b}{2}\cos\theta\dot{\theta}\right)^2 + 4 \times \frac{1}{2}I\dot{\theta}^2 \quad (1)$$

$$V = (M + 4m)g\left(\frac{b}{2}\sin\theta\right) \quad (2)$$

위 결과를 바탕으로 Lagrangian  $L$ 을 다음과 같이 정의할 수 있다.

$$\begin{aligned} L &= T - V \\ &= \frac{1}{2}(M + 4m)\left(\frac{b}{2}\cos\theta\dot{\theta}\right)^2 + 4 \times \frac{1}{2}I\dot{\theta}^2 - (M + 4m)g\left(\frac{b}{2}\sin\theta\right) \\ &= \left\{2I + \frac{1}{8}(M + 4m)b^2\cos^2\theta\right\}\dot{\theta}^2 - \frac{1}{2}(M + 4m)gb\sin\theta \quad (3) \end{aligned}$$

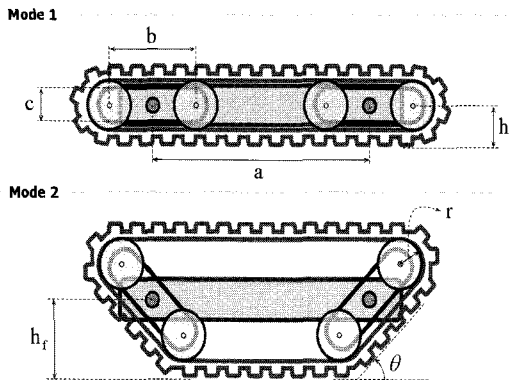


그림 5. VSTR의 설계 변수.

Fig. 5. Design parameters of the VSTR.

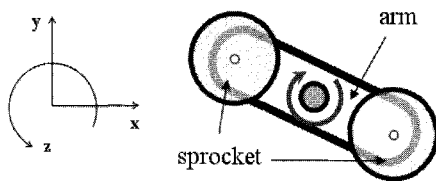


그림 6. 암 모듈의 회전.

Fig. 6. Rotation of the arm module.

표 1. 질량 특성과 입력 토크.

Table 1. Mass properties and input torque.

Variable	Definition
$M$	Mass of robot body
$m$	Mass of an arm module
$m_a$	Mass of an arm
$m_s$	Mass of a sprocket
$I$	Moment of inertia of an arm module
$I_a$	Moment of inertia of an arm
$I_s$	Moment of inertia of a sprocket
$\tau$	Applied torque

주행모드 변형은 1자유도 시스템에 해당하므로 generalized coordinate  $q$ 가 1개 존재하고, 이는 입사각  $\theta$ 로 정의할 수 있다. 또한, 주행모드 변형을 위한 외부 입력 토크가 존재하므로 non-conservative force  $Q_{nc}$ 는  $\tau$ 로 정의할 수 있다. 따라서 1자유도 시스템에 대한 Lagrangian 운동 방정식은 다음과 같이 재구성된다.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = Q_{nc} \Rightarrow \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau \quad (4)$$

그리고 위 식에 대입하기 위해 필요한  $\frac{\partial L}{\partial \dot{\theta}}$ 과  $\frac{\partial L}{\partial \theta}$ 은 각각 다음과 같이 구해진다.

$$\frac{\partial L}{\partial \dot{\theta}} = \frac{1}{4}(M + 4m)b^2\cos^2\theta + 4I\dot{\theta} \quad (5)$$

$$\begin{aligned} \frac{\partial L}{\partial \theta} &= -\frac{1}{4}(M + 4m)b^2\sin\theta\cos\theta\dot{\theta}^2 \\ &\quad - \frac{1}{2}(M + 4m)gb\cos\theta \quad (6) \end{aligned}$$

따라서 위의 (5), (6)을 (4)에 대입하여 정리하면 다음과 같은 운동 방정식이 유도된다.

$$\begin{aligned} \left\{4I + \frac{1}{4}(M + 4m)b^2\cos^2\theta\right\}\ddot{\theta} \\ - \frac{1}{4}(M + 4m)b^2\sin\theta\cos\theta\dot{\theta}^2 \\ + \frac{1}{2}(M + 4m)gb\cos\theta = \tau \quad (7) \end{aligned}$$

3. 감쇠 효과 보상

이제까지 유도된 운동 방정식은 Mode 1과 Mode 2를 분리하여 생각했을 때 발생하는 에너지를 토대로 한 이론적 결과이다. 그러나 실제로는 기어 간의 감쇠 효과, 축과 베어링의 마찰, 고무 트랙의 장력 등이 존재하여 주행모드 변형 과정에 영향을 준다. 따라서 실제 변형 과정과 동일한 모델이 되려면 이와 같은 감쇠 효과를 추가적으로 고려해야 한다.

실제로 VSTR의 물체를 지지하고 암 모듈 및 트랙을 공중에 뜨게 한 뒤 가만히 놔두면 암 모듈과 트랙의 자중이 로봇 물체를 기준으로 모멘트를 발생시켜서 평형을 이루지 못하

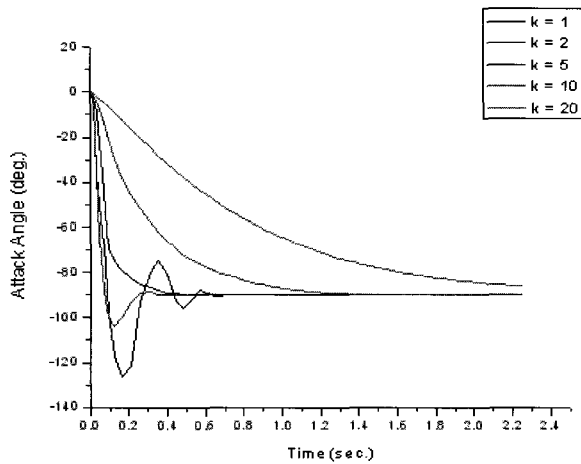


그림 7. 감쇠 계수에 따른 입사각 변화.  
Fig. 7. Attack angle according to the damping coefficients.

고 치지는 현상이 나타난다. 하지만 overshoot은 발생하지 않으며 1초 이상의 시간동안 천천히 치지게 된다.

본 연구에서는 이와 같은 현상을 동역학 모델에 추가하기 위해 공중에 떠있는 VSTR을 구속 조건으로 설정한 뒤 운동 방정식에  $k\dot{\theta}$  형태의 감쇠항(damping term)을 삽입하여 실제 상황과 동일할 때의 감쇠 계수(damping coefficient)를 추출했다. 그림 7은 VSTR의 몸체만 지지되고 트랙은 공중에 떠있는 상태에서 다양한 감쇠 계수  $k$ 에 대한 VSTR의 입사각 변화를 모의실험을 통해 비교한 것이고, 이를 토대로 실제 상황과 유사한 결과를 보이는 경우의 감쇠 계수 10을 반영하여 최종적인 운동 방정식을 완성하였다. 따라서 주행모드 변형에 대한 최종적인 운동 방정식은 다음과 같다.

$$\begin{aligned} & \left\{ 4I + \frac{1}{4}(M + 4m)b^2 \cos^2 \theta \right\} \ddot{\theta} \\ & - \frac{1}{4}(M + 4m)b^2 \sin \theta \cos \theta \dot{\theta}^2 \\ & + \frac{1}{2}(M + 4m)gb \cos \theta + 10\dot{\theta} = \tau \end{aligned} \quad (8)$$

**IV. PI-based Feedforward 제어 기법의 검증**

앞서 우리는 주행모드 변형 중 나타나는 정체 현상을 극복하기 위해 PI-based feedforward 제어 기법을 대안으로 제시한 바 있다. PI-based feedforward 제어는 고전적인 PI 제어 기법에 추가적으로 feedforward gain을 입력으로 삽입하여 정상 상태 오차를 없애주는 방법으로 VSTR의 주행모드 변형과 같이 빠르고 안정적으로 목표점에 도달해야 하는 경우 유용하게 쓰일 수 있다. 본 연구에서는 이러한 PI-based feedforward 제어 기법이 여타의 비선형 시스템 제어 기법과 비교하여 어떻게 다르고 어떤 점이 뛰어난지를 모의실험을 통해 검증하도록 한다.

**1. PI-based Feedforward 제어 vs. PI 제어**

우선 선행 연구에서 사용한 PI 제어 기법이 주행모드 변형에 어떠한 영향을 미치는지 재차 확인하고 이를 극복할 수 있는 방법을 모색한다. 그림 8은 전형적인 PI 제어의 블록선도(block diagram)를 보여준다. 앞서 정의한 운동 방정식을  $\theta$

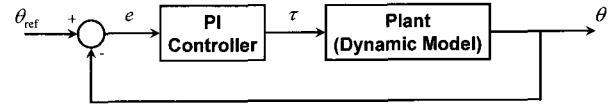


그림 8. PI 제어의 블록 선도.  
Fig. 8. Block diagram for PI control.

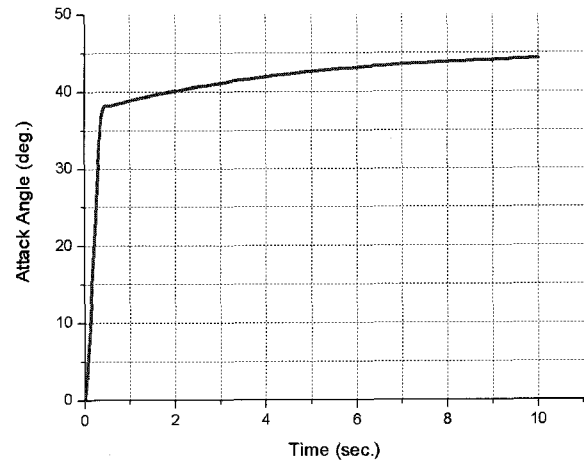


그림 9. PI 제어 시 입사각 응답 곡선.  
Fig. 9. Response for PI control.

에 대해 풀고 목표로 하는 입사각  $\theta_{ref}$ 와 비교하면 오차  $e$ 가 발생한다. 이후 P gain과 I gain을 조정하면 오차를 반영한 새로운 입력 토크를 입력해주고 이 과정을 반복하면 PI 제어를 이용한 주행모드 변형이 구현된다.

본 연구에서는 여러 경우의 P gain, I gain을 조합해서 모의 실험을 수행해보았다. 그 결과, 시스템의 안정성(stability)이 보장되는 범위 내에서 그림 9와 같이 실제 상황과 유사한 정체 현상을 확인할 수 있었다. 이미 언급한 바와 같이 이러한 현상의 주원인은 로봇 자중에 의한 정상 상태 오차이므로 응답 속도와 overshoot을 고려한 최적의 P gain을 찾은 뒤 I gain을 계속 증가시키면 제거할 수 있다. 그러나 이러한 경우는 원하는 성능을 충족시키기 위해 모터의 최대 출력 토크를 수 초 이상 유지해야 하므로 모터 수명이 단축되고 장기적으로 봤을 때 시스템에 안 좋은 영향을 주게 된다. 또한, 실제 실험에서는 모의실험과는 다르게 최적의 gain을 하나의 값으로 추출해내기 힘들기 때문에 본 경우와 같이 미소한 gain 차이로 큰 성능 차이를 보이는 사례에서는 선택된 gain을 신뢰하기 어렵다. 즉, 어느 정도의 범위를 두고 비슷한 성능을 낼 수 있는 I gain이 존재하지 않기 때문에 PI 제어만으로는 정체 현상을 제거하기에 무리가 있다.

한편, 주행모드 변형 시 로봇 자중을 이길 수 있는 만큼의 토크를 미리 입력하여 시스템의 무리를 줄여주는 PI-based feedforward 제어 기법을 사용했을 때에는 수많은 성공 사례들을 찾을 수 있었다. 그림 10과 11은 각각 PI-based feedforward 제어 기법의 블록 선도와 응답 곡선의 일례를 보여준다. 그림에서 알 수 있듯이 정상 상태 오차를 비롯한 정체 현상이 사라지고 빠르고 안정적으로 최종 입사각에 도달하는 것을 관찰할 수 있다. 따라서 PI-based feedforward 제어는 가장 간단하면서도 빠른 응답 속도, overshoot 없는 안정적인 주행모드

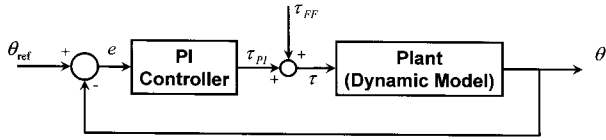


그림 10. PI-based feedforward 제어의 블록 선도.  
Fig. 10. Block diagram for PI-based feedforward control.

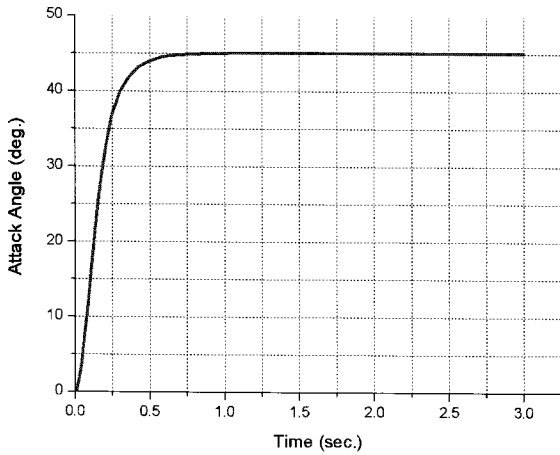


그림 11. PI-based feedforward 제어 시 입사각 응답 곡선.  
Fig. 11. Response for PI-based feedforward control.

변형을 완벽하게 추구할 수 있기 때문에 VSTR에 매우 적합한 제어기로 사려된다.

2. PI-based Feedforward 제어 vs. Feedback Linearization

PI-based feedforward 제어 기법은 기존의 PI 제어가 갖고 있는 문제점을 해결하면서 이미 그 우월성을 입증했다고 할 수 있다. 그러나 VSTR의 주행모드 변형이 비선형 시스템을 감안할 때 보다 나은 비선형 시스템 제어 기법이 존재할 수 있는 가능성을 배제하면 안 된다. 따라서 본 연구에서는 시스템의 설계 변수가 고정적이고, 설계자가 모든 설계 변수를 정확하게 알고 있을 때 가장 뛰어난 성능을 보이는 feedback linearization 기법에 대해 소개하고 제안한 PI-based feedforward 제어 기법과의 비교를 통해 우월성을 따져본다.

그림 12는 VSTR의 주행모드 변형을 feedback linearization 기법으로 제어한다고 가정했을 때의 블록 선도를 보여준다. feedback linearization 기법은 임의의 비선형 시스템이 있을 때 이에 대한 모델을 정확하게 수립한 뒤 비선형 요소들을 feedback을 통해 제거함으로써 closed-loop 시스템이 선형 시스템으로 전환되는 효과를 얻을 수 있는 방법이다[8]. 앞서 유도한 운동 방정식으로부터 A, B, C, D 를 다음과 같이 정의해보자.

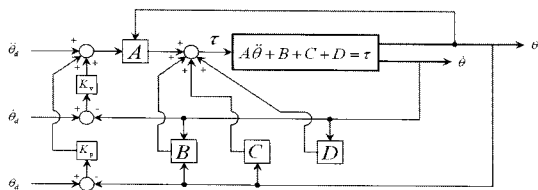


그림 12. Feedback linearization의 블록 선도.  
Fig. 12. Block diagram for feedback linearization.

$$\begin{aligned}
 A &= 4I + \frac{1}{4}(M + 4m)b^2 \cos^2 \theta \\
 B &= -\frac{1}{4}(M + 4m)b^2 \sin \theta \cos \theta^2 \\
 C &= \frac{1}{2}(M + 4m)gb \cos \theta \\
 D &= 10\dot{\theta}
 \end{aligned} \tag{9}$$

그림 주행모드 변형에 대한 운동 방정식은 다음과 같이 간소화된다.

$$A\ddot{\theta} + B + C + D = \tau \tag{10}$$

이 때, B, C, D 는 비선형 요소이므로 제어 입력  $\tau$  를 다음과 같이 설정하면 closed-loop 시스템을 선형적으로 변환할 수 있다.

$$\begin{aligned}
 \tau &= \sigma\tau' + \nu \\
 \text{where } \sigma &= A, \quad \nu = B + C + D, \\
 \tau' &= \ddot{\theta}_d + K_v\dot{e} + K_p e \quad (e = \theta_d - \theta)
 \end{aligned} \tag{11}$$

위 (11)을 (10)에 대입하고 동류항을 소거하면 다음과 같은 선형 2차 오차 방정식(linear 2<sup>nd</sup> error equation)이 성립한다.

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \tag{12}$$

이 때,  $K_v, K_p$  는 설계자가 원하는 overshoot, settling time 등의 성능 지수에 따라 달라지는 gain이고, 이는 선형 2차 시스템의 성능 지수 결정 방법에 의거하여 결정할 수 있다.

이제 본격적으로 PI-based feedforward 제어 기법과 feedback linearization 기법을 비교해보자. 그림 13은 PI-based feedforward 제어 기법 사용 시 가장 우수한 성능을 보이는 1개의 사례와 feedback linearization 기법 사용 시 가장 우수한 성능을 보이는 3개의 사례를 동일한 시간 영역에서 비교한 것이다. 그림에서 알 수 있듯이 목표점인 45° 입사각 주변에서는 차이를 가늠하기 힘들 만큼 비슷한 성능을 보이고 있다. 그러나 정상 상태까지 도달하는 rise time이 다소 차이가 나기 때문에 목표점의 90% 지점인 약 40° 입사각에서의 상황을 비교했을 때 PI-based feedforward 제어 기법이 반응 시간 면에서 미소

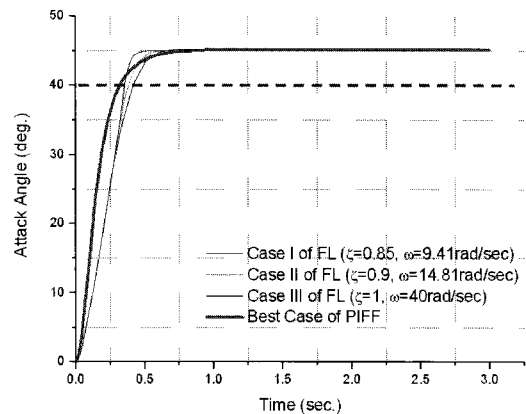


그림 13. PI-based feedforward 제어와 feedback linearization의 응답 곡선.

Fig. 13. Response for PI-based feedforward control and feedback linearization.

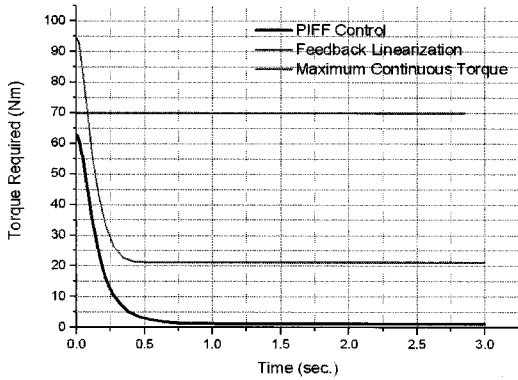


그림 14. PI-based feedforward 제어와 feedback linearization의 최대 토크 요구량.

Fig. 14. Requirement of maximum torque for PI-based feedforward control and feedback linearization.

하게나마 feedback linearization 기법보다 우월하다고 결론지을 수 있다.

한편, 두 제어 기법의 실용성을 비교하는데 있어서 가장 중요한 점은 바로 최대 토크 요구량(requirement of maximum torque)이다. 그림 14는 주행모드 변형 과정에서 두 제어 기법의 최대 토크 요구량을 보여준다. 본 연구에서 사용하는 주행모드 변형 모터는 최대 70Nm의 토크까지 출력할 수 있다. VSTR의 몸체는 약 7cm의 높이를 가지기 때문에 이로 인해 최대 허용 크기의 모터가 결정되고, 모터 출력이 모터 크기에 비례하는 것을 고려할 때 최대 토크 요구량은 무시할 수 없는 고려 대상임에 틀림없다. 그러나 그림에서 알 수 있듯이 feedback linearization 기법을 사용하게 되면 그림 13에서와 같은 성능 보이기 위해 최대 95Nm의 토크를 요구하게 되고 이후에도 추가적인 토크를 연속적으로 요구하게 된다. 따라서 이를 실제로 적용하게 된다면 사용 중인 모터의 최대 출력 토크에서 포화(saturation)가 발생한 채로 수 초가 유지될 것이고, 원하는 성능은 기대할 수 없을 것이다. 따라서 시스템의 유지와 성능 구현을 위해 feedback linearization 기법은 배제하는 것이 바람직하다.

이 외에도 제어기의 성능을 평가하는 요소로는 강건성(robustness)이 있다. 이를 검증하기 위해 본 연구에서는 임의의 외란을 입력으로 함께 넣어줬을 때 주행모드 변형이 어떤 형태로 변화하고 어느 정도의 오차가 발생하는지를 살펴보았다. 그림 15, 16은 각각 PI-based feedforward 제어와 feedback linearization에 대한 5번의 수행 결과를 보여주고 있다. 이로 부터 PI-based feedforward 제어를 수행하면 feedback linearization을 수행했을 때보다 목표점에 잘 수렴하는 것을 관찰할 수 있다. 한편, 표 2는 정상 상태에 도달한 이후, 각 시도(trial)에 대한 ‘평균 오차’와 ‘최대 오차들 간의 표준편차’를 비교한 것이다. 평균 오차는 5번의 시도를 통틀어서 목표점인 45°와 얼마나 차이가 나는지를 비교한 것이고, 최대 오차들 간의 표준편차는 각 시도에 대해 얼마나 일관성 있는 결과를 나타내는지에 대한 지수이다. 그 결과, PI-based feedforward 제어 기법이 feedback linearization 기법보다 낮은 차이를 보이며 우월성을 입증했다. 따라서 PI-based feedforward 제어 기법은 VSTR의 주행모드 변형에 적용할 시 비용(cost), 반응 시간

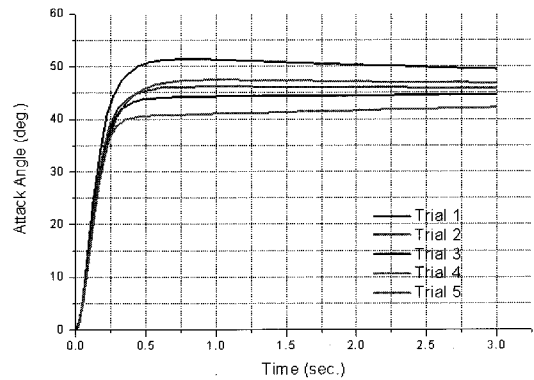


그림 15. PI-based feedforward 제어 기법의 강건성.

Fig. 15. Robustness for PI-based feedforward control.

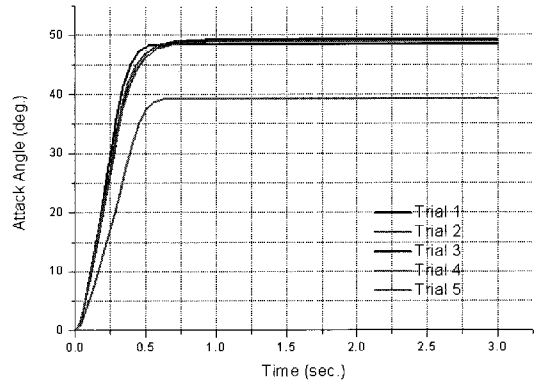


그림 16. Feedback linearization 기법의 강건성.

Fig. 16. Robustness for feedback linearization.

표 2. 강건성의 정량적 비교.

Table 2. Quantitative analysis for robustness.

Feedback Linearization		PI+Feedforward Control	
Standard deviation among max. errors	Avg. error	Standard deviation among max. errors	Avg. error
4.41°	1.97°	3.34°	1.47°

(response time), 안정성(stability), 강건성(robustness), 그리고 최대 토크 요구량에서 feedback linearization보다 모두 우위에 있다는 것을 알 수 있다.

### 3. PI-based feedforward 제어 vs. 비선형 시스템 제어

비선형 시스템 제어는 매우 실용적이면서도 광범위한 연구 분야로써 현재에도 여러 가지 기법들이 설계자의 요구에 따라 개발 및 응용되고 있다. 그러나 대부분 기본적인 틀에 입각하여 새로운 응용 제어 기법을 개발하는 것으로 정리할 수 있다. 이러한 기본적인 틀에 해당되는 제어 기법은 feedback linearization, sliding mode control, adaptive control 등이고, 여기에 neural network 혹은 유전자 알고리즘(genetic algorithm)과 같은 학습 기반 알고리즘을 삽입하여 목적을 두고 있는 성능에 근접한 제어기를 설계하는 것이 보편적인 추세라고 할 수 있다.

표 3은 본 연구에서 제안한 PI-based feedforward 제어 기법과 위에서 언급한 여타의 비선형 시스템 제어 기법의 장단점을 총괄적으로 비교한 결과이다. PI-based feedforward 제어 기법은 그 대상을 VSTR의 주행모드 변형으로 한정하였을 때 간단한

표 3. VSTR의 주행모드 변형에서의 PI-based feedforward control vs. nonlinear system control.

Table 3. PI-based feedforward control vs. nonlinear system control in case of driving mode transformation for VSTR.

	PI+Feedforward Control	Applied Nonlinear Control
Merit	<ul style="list-style-type: none"> <li>구조가 매우 단순하다.</li> <li>설계 및 구현이 용이하다.</li> <li>계산량이 적다.</li> <li>학습 알고리즘이 필요하다.</li> <li>빠르면서 동시에 안정적인 응답을 기대할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>일반적으로 PID 제어기와 같은 종래의 제어기보다 모든 면에서 우수하다. (Stability, Speed of response, Robustness)</li> <li>Gain Tuning이 필요없다.</li> </ul>
Demerit	<ul style="list-style-type: none"> <li>초기 Gain Tuning이 필요하다. (Trial and Error)</li> </ul>	<ul style="list-style-type: none"> <li>구조적으로 다소 복잡하다.</li> <li>계산량이 많다.</li> <li>구동기의 성능에 제약이 있을 경우에는 성능이 급격히 나빠진다.</li> </ul>

구조로 쉽고 보다 우수한 성능을 기대할 수 있는 최적의 방법인 반면, 원하는 성능에 근접할 수 있도록 초기에 gain tuning을 해주어야 한다는 부담이 있다. 한편, 일반적인 비선형 제어 기법들은 학습 기반의 적응 방식을 통해 원하는 성능으로 수렴하기 때문에 gain tuning이 필요 없고, 제어하고자 하는 모델의 정보가 미약하더라도 효과적으로 대처할 수 있다는 장점이 있다. 그러나 제어 구조가 다소 복잡하며 계산량이 많아 하드웨어의 성능에 좌우될 수 있다는 단점이 있고, 제어 대상인 구동기(actuator)에 제약이 있을 경우 이론적으로 기대할 수 있는 성능을 낼 수 없다는 한계점이 있다. 따라서 비선형 시스템 제어는 설계자와 시스템의 요구 사항에 따라 적절한 기법을 선택하거나 응용할 수 있는 문제라고 할 수 있다.

**V. 결론**

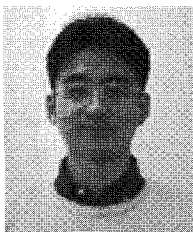
본 연구에서는 구조작업을 위해 개발된 VSTR의 주행모드 변형 과정에서 발생하는 문제점을 언급하고, 이를 해결하기 위해 제시한 PI-based feedforward 제어 기법을 검증하였다. 이를

위해 주행모드 변형을 운동 방정식으로 표현한 뒤 모의실험을 통해 PI 제어, PI-based feedforward 제어, feedback linearization 등을 비교하면서 최적의 해결책을 모색하였고, 대안으로 선정된 PI-based feedforward 제어 기법의 당위성을 성능 지수들을 비교함으로써 증명하였다.

향후에는 선정된 PI-based feedforward 제어기를 VSTR에 탑재하여 실험적인 검증 작업을 수행할 것이며 장애물 극복 과정에서 나타나는 선행 연구와의 차이점을 비교, 분석할 것이다.

**참고문헌**

- [1] Z. Wang and H. Gu, "A review of locomotion mechanisms of urban search and rescue robot," *Industrial Robot: An International Journal*, vol. 34, no. 5, pp. 400-411, 2007.
- [2] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, no. 2, pp. 138-153, 2004.
- [3] 최근하, 정해관, 최지혁, 최현도, 곽윤근, "가변형상 트랙 메커니즘의 주행로봇을 위한 장애물 탐지 및 극복," 한국군사과학기술학회 2006년도 종합학술대회 논문집, pp. 1017-1020, 2006.
- [4] 박동일, "계단 등반을 위한 가변형 단일트랙 메커니즘의 설계 및 해석," 한국과학기술원 기계공학과 기계공학전공 박사학위논문, pp. 60-68, 2006.
- [5] 최근하, 정해관, 곽윤근, "PSD 센서를 이용한 구조로봇의 장애물 탐지 및 주행 모드 결정," 대한기계학회 2006년도 추계학술대회 논문집, pp. 680-685, 2006.
- [6] 정해관, 강현석, 최현도, 곽윤근, "구조로봇의 주행 모드 변형을 위한 동역학 모델 기반 제어," 한국자동차공학회 2007년도 추계학술대회 논문집, pp. 554-559, 2007.
- [7] L. Meirovitch, *Principles and Techniques of Vibrations*, 1<sup>st</sup> Ed., Prentice-Hall, New Jersey, 1997.
- [8] E. Slotine and W. Li, *Applied Nonlinear Control*, 5<sup>th</sup> Ed., Prentice-Hall, New Jersey, 1991.
- [9] J. Craig, *Introduction to Robotics*, 2<sup>nd</sup> Ed., Addison-Wesley, New York, 1989.



**정 해 관**

1981년 10월 1일생. 2003년 한국과학기술원 기계공학과(공학사). 2005년 한국과학기술원 기계공학과(공학석사). 2005년~현재 한국과학기술원 기계공학과 박사과정 재학중. 관심분야는 구조로봇의 자율주행.



**강 현 석**

1980년 12월 9일생. 2006년 부산대학교 기계공학부(공학사). 2008년 한국과학기술원 기계공학과(공학석사). 관심분야는 주행로봇의 험지 극복을 위한 지형 특성 추정.



**곽 윤 근**

1944년 6월 26일생. 1974년 서울대학교 기계공학과(공학사). 1975년 The Univ. of Colorado at Boulder 공학설계 및 경제평가학과(공학석사). 1978년 The Univ. of Texas at Austin 기계공학과(공학박사). 1985년~현재 한국과학기술원 기계공학

과 교수. 관심분야는 주행로봇의 자율주행, 감성로봇을 위한 음성 인식.