

ID3 를 이용한 생체모방형 수중로봇-익투스의 장애물 회피 및 회피 속도 결정 알고리즘

ID3 based Obstacle Avoiding and Motion Pattern Selection Algorithm for Bio-inspired Fish Robot-Ichthus

*전명재¹, #류영선²

*M. J. Jun(21cnehemiah@gmail.com)¹ #Y. S. Ryuh(ysryuh@kitech.re.kr)²

^{1,2} 한국생산기술연구원 생체모방연구실

Key words : Underwater Robots, Machine learning, ID3 algorithm, Obstacle Avoid, Bio-inspired Fish Robot-Ichthus

1. 서론

수중로봇을 정의하면 사용자의 의도와 목적에 의해 수중에서의 작업이 가능한 로봇을 말한다. 수중로봇의 형태와 기능은 사용 목적에 따라 다양하지만 최근 물고기 형태의 생체모방형 수중로봇이 많이 연구되고 있다. 이는 기존의 프로펠러형에 비해서 효율 및 기동성이 우수한 것으로 알려져 있기 때문이며 주로 생태모니터링(오염물 추적, 어종감시, 수질검사)을 목적으로 개발되고 있다. 때문에 이러한 로봇들은 대부분 자율주행을 기본으로 하고 있는데 자율주행에 있어서 장애물회피에 관한 이슈는 모든 로봇이 풀어야 할 공통과제라 할 수 있다. 이러한 과제를 풀기 위해선 먼저 로봇이 장애물을 인지하는 것이 필수적이다. 수중에서의 장애물은 수초나 돌, 부유물이나 또다른 수중로봇 및 수중 생물이 될 수 있다. 수중이라는 공간은 3차원 공간이기 때문에 지상에서는 장애물로 인식되지 않았던 물체들이 장애물로 인식 될 수도 있는 문제를 가지고 있다. 예를 들면 로봇이 물속의 커다란 바위를 장애물로 인식해 회피할 필요는 있지만 흐르는 물속의 낙엽과 같은 부유물을 장애물로 인식해 회피할 필요는 없는 것이다. 이러한 로봇의 동작은 에너지 효율성을 감소시키기 때문이다. 이렇듯 실제 강이나 하천에서의 환경은 실험실의 수족관보다는 더 많은 변수와 더 많은 장애물을 가지고 있다. 이러한 환경에 수중로봇이 능동적으로 대처하고 불필요한 움직임을 없애며 에너지의 효율성을 높이기 위해선 수중 환경에서의 장애물 인식과 회피에 대한 좀더 지능적인 방법이 필요하다. 이에 본 논문에서는 자율 주행이 가능한 물고기 형태의 수중로봇에서 장애물 인지 및 회피방법에 대한 보다 지능적인 알고리즘을 기술한다. 논문에서 제시하는 알고리즘은 수중 로봇의 지능적 장애물 회피에 관한 것으로 기존의 단순 장애물 회피 방법과는 구별된다.

센싱정보를 모니터링하여 모니터링한 구간 동안의 값들을 분석하고 분석한 데이터들로부터 장애물인식과 회피 그리고 회피 속도를 결정하게 하는 알고리즘이다. 여기서 특정시간 동안의 센싱된 정보의 분석을 모니터링 기법이라 명하며 특정시간은 물고기 로봇의 시스템사양에 의해 가변적 일 수 있다. 제시하는 알고리즘의 구현 방법은 먼저 로봇에 사용되는 거리센서로부터 입력받는 센싱정보를 A/D변환하여 디지털화된 정보를 특정시간 동안 관찰한다. 이때 센싱된 정보들의 합은 첫번째 Energy Filter를 거쳐서 장애물의 감지 여부를 결정하게 되는데 센서값들로부터 얻어진 값들의 평균값이 중간값 보다 높을 때만을 장애물로 여기게 된다. 동시에 센싱된 데이터들은 High, Regular, Low의 3개의 Level로 분류한다. 센싱된 거리정보를 Level로 분류하는 이유는 로봇이 장애물에 가까워 질수록 에너지값은 커지게 되는데 특정 Level이상에서만 장애물로 인식하게 되면 작은 부유물등에 반응한 센싱정보를 noise로 처리 할 수 있기 때문이다. Level정보는 Decision tree(ID3) 알고리즘의 Training data로 사용되어 수중에서 로봇의 LEFT/RIGHT회피동작과 UP/DOWN회피동작을 분류하게 된다. 분류된 동작은 Position Filter를 거쳐서 최종 회피 동작과 회피 속도를 결정하게 된다.

2. Multi-sensing based Evasion Algorithm

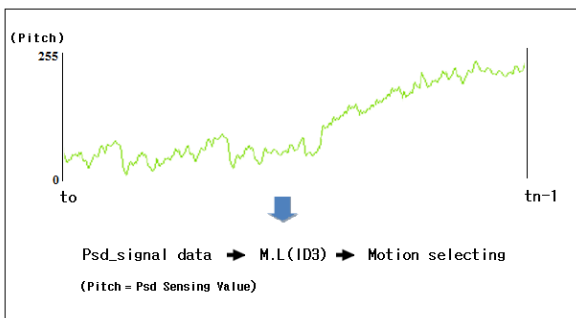


Fig. 1 Monitoring of PSD sensing data

본 논문에서 제시하는 알고리즘은 수중에서 물고기 로봇이 각각의 거리센서로부터 즉각적인 반응이 아닌 위의 Fig.1에서와 같이 특정시간동안의 입력된

2.1 System Description

시스템 구성은 다음 Fig.2 와 같다. 제시하는 System 에서는 거리감지를 위해 전방 좌, 우에 각각 한 개의 PSD 센서를 사용하고 있으며 전체적인 System 은 물고기 형태의 수중 로봇이다.

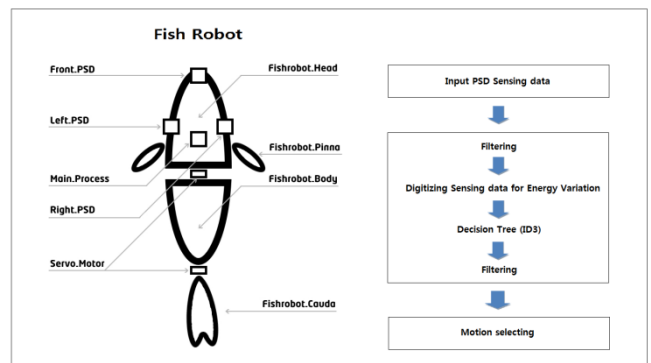


Fig. 2 System architecture

또한 수중에서의 상하 이동을 위한 옆 지느러미와 동력을 얻기 위한 꼬리지느러미도 같이 구성되어 있으며 꼬리지느러미는 몸통과 꼬리에 연결되어 있는 Servo Motor 에 의해 sin 형태의 파형으로 꼬리를 움직여 동력을 얻게 하며

센싱된 정보의 가공 및 회피동작 선택과 회피 속도는 Main process 에서 처리하도록 한다.

2.2. Digitizing Sensing-data for Energy Variation

먼저 각 센서로부터 센싱된 data 를 Level 화 시키는 방법은 아래의 식 (1), (2), (3)과 같다.

- (1) (PSD Level|High) = $\sum_{t=0}^{t_n-1} \text{Pitch} < 255(t_{n-1}) / 3$
- (2) (PSD Level|Regular) = $255(t_{n-1}) / 3 < \sum_{t=0}^{t_n-1} \text{Pitch} < 500(t_{n-1}) / 3$
- (3) (PSD Level|Low) = $\sum_{t=0}^{t_n-1} \text{Pitch} > 500(t_{n-1}) / 3$

각각의 Level 값은 위의 조건을 반영한다. 또한 Pitch 는 t_n 에서의 PSD 로부터 입력받은 센싱 data 의 A/D 변환 값이다. (min pitch: 0, max pitch 255)

2.3. Motion selection

먼저 Level 화된 센싱 data 는 기계학습방법을 거쳐 UP/DOWN 회피동작과 LEFT/RIGHT 회피동작을 결정하게 된다. 사용되는 학습방법은 Decision tree(ID3) 알고리즘이다. 밑의 Fig.3 는 Tree Model 을 구성하기 위해 Monitoring 구간 의 센싱정보를 Level 화 하여 만들어진 data table 이다.

Front_PSD	Right_PSD	Left_PSD	Aboiding Motion
Regular	High	High	UP / DOWN
High	High	Low	LEFT / RIGHT
Low	Regular	Low	UP / DOWN
Low	High	Regular	LEFT / RIGHT
Low	Low	High	LEFT / RIGHT
Regular	Low	Low	UP / DOWN

Fig. 3 Example of Training data table

2.4. Decision Tree(ID3)

Decision Tree 는 확률적으로 정보이득(information gain)이 가장 많은 것을 root 로드로 하여 Entropy(s)가 0 이 될 때 까지 밑으로 subroot 를 형성 하는 구조로 되어 있으며 목적 속성이 C 개의 서로 다른 값(예:Yes, No)을 가질 수 있을 때 예제 인스턴스 집합(s)의 Entropy 는 다음과 같이 정의 된다.

$$Entropy(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

이때 예제 인스턴스 집합(s)에서의 속성 A 에 대한 정보이득 Gain(S,A)는 다음과 같이 정의 된다.

$$Gain(S,A) = Entropy(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Fig. 3 의 Training data 를 ID3 Algorithm 에 적용시킨 결과 생성된 Tree Model 은 Fig. 4 와 같다.

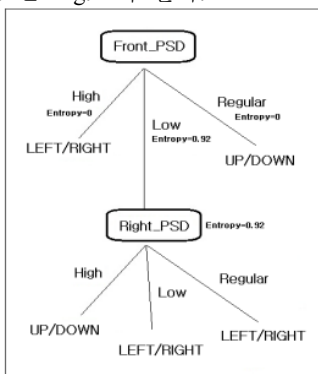


Fig. 4 Decision Tree of Training data

2.5. Motion Filtering

결정된 Tree Model 로부터 각각 2 개의 회피 동작정보(상, 하, 좌,우)를 얻을 수 있는데 Position Filter 를 거쳐 이중 하나의 동작만을 선택하게 된다. 이때 Filter 의 논리적 구성은 다음의 식(4), (5), (6), (7) 과 같이 구성된다.

- $A = \sum_{t_0}^{t_{n-1}} \text{Pitch}_{(PSD_Front)}$, $B = \sum_{t_0}^{t_{n-1}} \text{Pitch}_{(PSD_Right)}$, $C = \sum_{t_0}^{t_{n-1}} \text{Pitch}_{(PSD_Left)}$
- (4) ((Left/Right)|Left) = $B < C$
- (5) ((Left/Right)|Right) = $B > C$
- (6) ((Up/Down)|Up) = $A < (B + C)$
- (7) ((Up/Down)|Down) = $A > (B + C)$
- (8) (E Filter|Go) = $255 / 2 > (A + B + C) / 3$
- (9) (E Filter|Select avoiding Motion) = $255 / 2 < (A + B + C) / 3$

2.5. Motion + Speed Selection

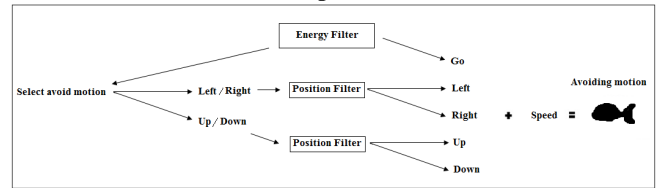


Fig. 5 Process of Avoiding motion

Energy Filter 를 거쳐 장애물을 인식하면 Select avoid motion Algorithm 이 동작한다. 이것에 의해 물고기 로봇의 최종 회피동작을 알 수 있는데 최종 회피동작은 Fig.5 에서 보이는 것 처럼 Left, Right, Up, Down 이렇게 4 개의 회피 동작중 하나를 선택하게 된다. 회피 동작은 Monitoring 하는 구간동안 계속 이루어진다. 즉 현재의 어떤 회피동작은 이전 Monitoring 구간의 결과값이 되는 것이다. 또한 본 논문에서는 실제 생태계속의 물고기가 장애물에 대해 반응하는 것을 Modeling 하여 생태모방공학적으로 접근하고자 회피동작과 함께 회피 속도의 가변성을 제시한다. 이는 물속의 물고기들을 보면 순간적으로 나타난 어떤 장애물에 대해서 최대한 빠르게 회피하는 것을 볼 수 있는데 이것은 순간적으로 나타난 장애물로부터 스스로 보호하고자 하는 동물적 보호본능이 앞서기 때문이다. 즉 빨리 피하지 않으면 위험하다는 논리가 만들어지는 것인데 이러한 동물적 보호 본능을 물고기 로봇에 Modeling 하기 위해선 회피동작과 함께 회피속도가 가변적이어야 한다. 따라서 회피 속도는 Monitoring 구간에서 각각의 센서값들을 더해 사용된 센서의 개수로 나눈 평균값으로 정의한다. (참고: 수식(10)) (min speed: 0, max speed 255)

$$(10) \text{Speed} = (A + B + C) / 3$$

3. 결론

본 논문에서는 거리 센서 데이터의 Monitoring 을 통한 장애물 인식과 회피방법 및 회피속도에 대한 결정방법을 제안하였다. 제안하는 기법은 거리센서의 단순 감지 및 단순회피방법과는 구별되어 여러 수중로봇과 여러 수중생물이 공존하는 수중환경에서 로봇의 에너지 절약 및 지능적 회피동작을 가능케 한다.

참고문헌

1. A Richards, J Bellingham, M Tillerson, and J. How "Co-ordination and Control of Multiple UAVs," submitted for publication at the 2002 AIAA Guidance, Navigation, and Control Conference.
2. Kalles, D., &Morris, T. (1996). "Efficient incremental induction of decision trees," *Machine Learning*, 24, 231-242.