

고장검출 기본 이론과 고장검출 필터의 개념

■이 은 성

(한국항공우주연구원 위성항행방법팀)

최근 다양하고 복잡한 산업 구조로 인하여 인적·물적 피해가 큰 대형 사고가 자주 발생하고 있으며 산업 전반에 신뢰성과 안전성 확보가 중요한 문제로 부각되고 있다. 산업이 전반적으로 복잡해 지고 그 시스템의 규모가 대규모로 진행됨에 따라 자동화된 장비의 사용이 빈번해 졌으며, 시스템의 고장을 유발하는 요인이 다양해 지고 고장의 발생 빈도 역시 증가하였다. 고장은 시스템의 정상운동을 저해하기도 하며, 사회적으로 큰 경제적인 손실을 가져다 주기도 한다. 고장의 영향을 최소화하기 위하여 고장검출 및 대응 기술이 필요하며, 고장검출과 관련된 이론의 고찰은 매우 중요하다. 고장검출 관련 이론은 다양한 분야에서 활용되고 있으나, 처음으로 접하는 단계에서는 그 이해에 어려움을 느낄 수 있다. 본 자료는 고장검출 관련 이론을 처음 접하는 독자에게 이를 쉽게 설명하는데 그 목적이 있으며, 고장검출 관련 기본이론과 고장검출에 많이 쓰이고 있는 고장검출 필터의 개념을 비교적 간결하게 설명하고 있다.

1. 고장검출 관련 용어

고장은 영문으로 Fault 또는 Failure로 표시할 수 있으며, 일반적으로 Fault는 물리적인 결점 자체를 말하며, Failure는 Fault의 결과로서 발생한 세부 시스템의 기능 장애나 전체 시스템 관점에서 광의적인 이상 작동상태를 말한다. 고장이 발생함에도 시스템의 신뢰도와 안전성을 향상 시키는 연구로는 Fault Avoidance, Fault Detection, Fault Isolation, Fault Masking, Fault Tolerance 등이 있다. Fault Avoidance는 과거에 일어난 고장 원인을 기반으로 경험에 의해 적절한 부품을 선택하고, 장비의 작동 환경을 조절하고, 시스템의 잡음을 제거하는 등 고장이 발생하기 전에 이를 예방하는 방법이다. Fault Detection은 고장 검출이라고 말하며, 신호의 급격한 변화, 장비 출력값의 한계 값 초과, 전체 시스템의 특성치의 변화 등을 감시하여, 이상 발생시 이를 신속하게 검출하고, 신속한 대응을 원활히 수행토록 하는데 그 목적이 있다. Fault Isolation은 고장분리하고 하며, 고장 발생시 그 원인을 분석하여 고장 부분을 확인하고 그 고

장을 분리하는데 그 목적이 있다. Fault Masking은 고장의 영향을 최소화시키는 기술이다. Fault Tolerance는 시스템에 고장이 발생하더라도 하드웨어나 소프트웨어의 중복 구조 등에 의해 시스템을 재조정하여 주어진 기능을 수행하도록 하는 적극적인 방법으로 최근에 활발히 연구되고 있다.

2. 고장검출 기법의 종류

고장검출 기법은 크게 하드웨어 중복 (Hardware Redundancy) 기법과 해석적 중복 (Analytic Redundancy) 기법으로 나눌 수 있다. 하드웨어 중복기법은 단순하게 고장의 발생 유무를 결정하는 정성적 (Qualitative) 방법이라고 할 수 있는 반면, 해석적 중복기법은 대상 시스템의 모델을 기준으로 해석적 모델을 구성하고 이를 활용하여 고장 검출을 수행한다는 점에서 정량적 (Quantitative) 방법이라고 할 수 있다. 하드웨어 중복기법은 동일한 기능을 갖는 장비를 두 개 이상 사용하여 고장에 대비하는 것을 의미한다. 하드웨어 중복기법은 시스템의 고장

에 대처하는 전통적인 방법으로 다중의 센서와 다중의 구동기를 장착하여 고장에 대처하는 방법으로 활용되었다. 하드웨어 중복기법은 쉽게 구현할 수 있다는 장점이 있으나 두 개 이상의 장비를 설치해야 하기 때문에 그 구현을 위해서 비용과 설치 장소 등의 문제가 발생할 수 있는 단점이 있다. 해석적 중복기법은 하드웨어 중복기법의 단점을 해결하기 위해 1970년대 이후에 제안되었다. 해석적 중복기법은 서로 다른 역할을 맡고 있는 장비라 하더라도 시스템의 내부의 특성에 대한 정보를 이용하면 하드웨어 중복기법을 이용한 것과 같은 효과를 얻을 수 있다는 사실에 바탕을 두고 있다. 최근 적은 수의 센서와 구동기로도 고장에 잘 대처할 수 있도록 해석적 중복기법을 이용하거나 제어시스템을 재설계하여 고장에 대처하고자 하는 연구들이 활발히 수행 되어 왔다.

3. 해석적 중복기법 종류

고장검출을 위한 해석적 중복기법은 크게 신호 검출(Signal Detection) 기법, 파라미터 검출(Parameter Identification) 기법, 고장검출 필터(Fault Detection Filter) 활용 기법, 패리티 공간(Parity Space) 활용 기법으로 구분할 수 있다. 해석적 중복기법에서는 시스템에 대한 해석적, 물리적 지식 혹은 경험적 지식을 바탕으로 구성된 해석적 모델을 이용하며 대상 시스템의 출력과 해석적 모델 출력 차이를 수학적으로 표현한 잔차(Residual)를 생성하고 이 잔차를 활용하여 고장검출을 수행하게 된다. 그림 1은 잔차를 만드는 방법을 도식적으로 설명하고 있다.

신호 검출 기법은 초기 고장검출 기법에 사용된 기법으로 스펙트럼 분석(Spectral Analysis), 통계적 결정(Statistical Decision Test) 등이 그 예가 될 수 있다. 신호 검출 기법은 통신이론을 기반으로 만들어진 기법이며, 시스템의 출력 신호만을 활용하는 것이 특징이다. 파라미터 검출 기법은 대상 시스템의 주요 구성 성분들과 파라미터의 관계를 활용하여 고장검출을 수행하는 방법으로 파라미터의 변화를 계속적으로 감시하면서 고장검출을 수행한다. 고장검출 필터 활용 기법은 고장에 대하여 잔차의 출력이 특정한 방향성을 갖도록 설정한 관측기를 이용하여 고장을 검출하는 방식이다. 일반적인 관측기와 같은 구조를 가지고 있으며, 적절한 이득행렬(Gain Matrix)을 활용하여 고장검출을 수행한다. 패리티 공간 활용 기법은 대상 시스템의 해석적 모델을 만드는 과정에서 패리티 벡터를 생성하고 시스템의 입출력의 간단한 산술적 계산

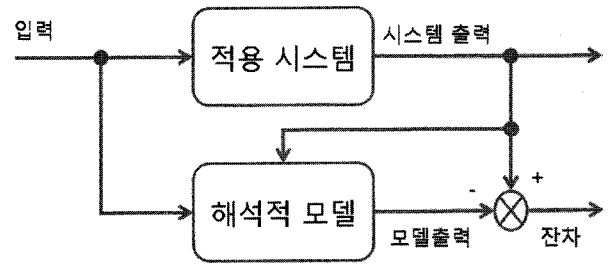


그림 1. 잔차 생성 방법

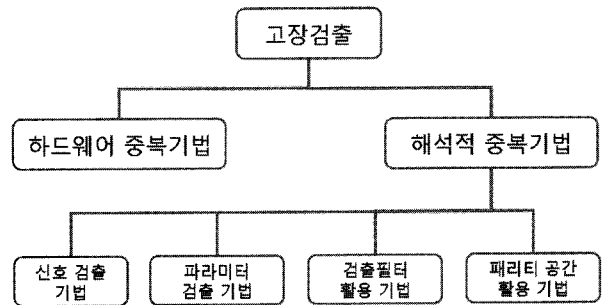


그림 2. 고장검출 기법의 종류

을 통해 잔차를 생성하여 고장검출에 활용한다. 대상 시스템에 따라 앞서 열거한 기법 중에서 효율성이 높은 기법을 선정할 수 있어야 한다. 그림 2는 앞서 설명한 고장검출 기법을 도식적으로 정리한 것이다.

4. 고장검출 필터 활용 기법

고장검출 필터를 활용한 고장검출 기법은 Beard와 Jones에 의해 처음 소개되었다. 전체 시스템에서 센서의 고장이 발생하지 않았다고 가정한다면 대상 시스템에 고장이 발생할 경우 대상 시스템의 실제 출력과 해석적 모델의 출력은 서로 다르게 될 것이다. Beard는 시스템 출력 에러를 되먹임(Feedback)시켜서 해석적 모델의 상태변수를 변화시켰다. 시스템에 임의의 고장이 발생하였을 때 출력 공간에 그 고장에 대응하는 출력을 발생시킴으로써 비교적 쉽게 시스템 내부의 고장을 구별할 수 있다. 이 방법을 활용하게 되면 시스템 내의 고장을 발견하고 고장의 위치를 파악할 수 있다. 이때, 고장이 발생하는 경우 출력 공간에서의 잔차를 구별할 수 있는 필터를 활용하게 된다.

다음은 고장검출 필터 활용 기법의 개념적 이해를 돕기 위하여 대상시스템의 수학적 모델을 설명하도록 한다. 시스템 동특성이 선형이며 시간에 따라 변화하지 않는다면 다음과

같은 상태변수 방정식으로 전체 시스템을 표현할 수 있다.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1)$$

$$y(t) = Cx(t) \quad (2)$$

여기서 A, B, C 는 각각 시스템행렬, 제어행렬, 출력행렬이고, $x(t), u(t), y(t)$ 각각 상태변수(State), 입력(Input), 출력(Output)을 나타낸다. 검출 필터의 기본적 상태변수의 해석적 모델은 출력 잔차의 되먹임이 있을 경우 다음과 같다.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + D[y(t) - \hat{y}(t)] \quad (3)$$

$$\hat{y}(t) = C\hat{x}(t) \quad (4)$$

여기서 $\hat{x}(t)$ 는 상태변수 추정치, $\hat{y}(t)$ 는 예측된 측정치 출력이다. 출력의 잔차는 실제 측정치와 예측된 측정치의 출력의 차로 정의되며 다음과 같다.

$$q(t) = y(t) - \hat{y}(t) \quad (5)$$

상태변수 오차 $e(t)$ 는 실제 상태변수 $x(t)$ 와 추정된 상태변수 $\hat{x}(t)$ 의 차이로 나타낼 수 있으며 다음과 같다.

$$e(t) = x(t) - \hat{x}(t) \quad (6)$$

상태변수 오차 방정식과 출력의 잔차 표현식을 정리하면 다음과 같이 변형할 수 있다.

$$\dot{e}(t) = [A - DC]e(t) \quad (7)$$

$$q(t) = Ce(t) \quad (8)$$

고장검출 필터 활용기법의 핵심은 되먹임 이득 D 의 알맞은 구성에 있다. 만약 실제 시스템에 고장이 발생하지 않을 경우 $[A - DC]$ 의 고유값(Eigenvalue)의 실수부가 0보다 작아 일 반적인 상태변수 추정기가 될 것이다. 되먹임 이득 D 를 구성 할 때 $[A - DC]$ 가 안정해서 $x(t)$ 가 $\hat{x}(t)$ 에 접근하도록 하여야 하며, 시스템에서 임의의 부분에 고장이 발생하는 경우 이에 대응하는 잔차가 잔차 공간에서 일관된 방향상을 갖도록 D 를 선택해야 하는 것이다.

전체 시스템에서 각각 세부시스템에 고장이 발생하는 경우 A, B, C 가 더 이상 실제 시스템을 따라가지 않으며, $y(t)$ 가

$\hat{y}(t)$ 에서 멀어지게 된다. 이러한 경우 고장검출을 수행하기 위해서는 각각 세부시스템 고장에 대처할 수 있는 적절한 수학적 모델이 필요하다. 아래에서는 전체 시스템을 구성하고 있는 구동기, 센서, 시스템의 동역학적 특성의 변화에 대한 해석적 모델을 활용하여 각각의 세부시스템 고장에 따르는 수학적 모델을 구성하는 방법에 대하여 설명하도록 한다.

가. 구동기의 고장 (Actuator Fault)

구동기의 고장은 제어 행렬 B 의 변화로 표시할 수 있다. B 의 변화에 따른 시스템의 영향을 살펴 보기 위하여 제어 행렬 B 와 제어 입력 벡터 $u(t)$ 를 재 구성할 수 있다.

$$Bu(t) = \begin{bmatrix} b_1 & b_2 & \dots & b_q \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_q(t) \end{bmatrix} = \sum_{i=1}^q b_i u_i(t) \quad (9)$$

구동기에 고장이 발생할 경우, 시스템에 명령으로 사용된 제어 벡터의 성분이 $u_{ci}(t)$ 라면, 고장에 의하여 시스템에 적용되는 명령은 $u_i(t) = u_{ci}(t) + \tilde{u}_i(t)$ 가 되며 $\tilde{u}_i(t)$ 는 구동기의 고장을 나타낸다. 이렇게 표현되는 구동기의 고장을 포함하는 상태 방정식은 (10)이 되며, 측정 방정식은 변화가 없기 때문에 (11)과 같이 (2)을 그대로 사용할 수 있다.

$$\dot{x}(t) = Ax(t) + Bu(t) + \sum_{i=1}^q b_i \tilde{u}_i(t) \quad (10)$$

$$y(t) = Cx(t) \quad (11)$$

나. 센서 고장 (Sensor Fault)

센서 고장에 대한 수학적 모델은 구동기 고장 부분의 수학적 모델과 다를 수 밖에 없다. 센서 고장의 수학적 모델은 상태변수 방정식에 포함되지 않고 다만 측정방정식에 추가되기 때문이다. 이러한 센서 고장은 비교적 간단히 표현되어, 측정 방정식에 고장을 나타내는 벡터와 단위 벡터의 곱을 더해줌으로써 표시할 수 있다. 측정값에 고장이 발생하는 경우 상태 방정식과 측정 방정식은 각각 (12)와 (13)으로 나타낼 수 있다.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (12)$$

$$y(t) = Cx(t) + I_p \mu \quad (13)$$

(13)은 새로운 행렬 F 를 활용하여 재구성 할 수 있으며, F 는 (14)를 만족하도록 선택한다.

$$CF = I_p \tag{14}$$

행렬 F 를 활용하여 재구성한 측정 방정식은 (15)와 같다.

$$y(t) = Cx(t) + CF\mu = C(x(t) + F\mu) \tag{15}$$

새로운 변수 $z(t)$ 를 $x(t) + F\mu$ 로 치환하면 (12)와 (13)은 (16)과 (17)로 표현할 수 있다.

$$\dot{z}(t) = Az(t) + Bu(t) + F\dot{\mu} - AF\mu \tag{16}$$

$$z(t) = Cz(t) \tag{17}$$

센서 고장 모형에 의해 만들어지는 잔차 벡터는 잔차 공간 안에서 일정한 방향을 갖지 못한다. 반면 그 생성된 잔차는 일정한 평면 안에 국한되게 되는데 그 평면은 해당하는 고장 센서와 유일한 관계를 갖는 평면이다. 물론 이 잔차의 거동이 일정 방향의 신호보다는 구별하기 쉽지 않지만, 이러한 고장에 의한 잔차는 센서가 아닌 다른 부분의 고장 현상과는 쉽게 구별된다.

다. 대상 시스템 고장 (System Fault)

대상 시스템 동특성의 예상하지 못한 변화 또한 고장의 범주로 분류할 수 있으며, 이를 상태 방정식으로 표현하면 (18)과 같으며, 측정 방정식인 (19)는 변화가 없으므로 (2)를 그대로 사용할 수 있다.

$$\dot{x}(t) = (A + \Delta A)x(t) + Bu(t) \tag{18}$$

$$y(t) = Cx(t) \tag{19}$$

새로운 상태 방정식을 나타내는 (18)는 특이값 분해 (Singular Value Decomposition) 등을 활용하여 $\Delta Ax(t)$ 를 $P\lambda$ 로 치환하면 새로운 상태 방정식인 (20)으로 표현할 수 있다.

$$\dot{x}(t) = Ax(t) + Bu(t) + P\lambda \tag{20}$$

라. 고장의 일반화
앞에서 구동기, 센서, 시스템의 동역학적 특성의 변화에 대

한 해석적 모델을 활용하여 각각의 세부시스템 고장에 따르는 수학적 모델을 구성하는 방법에 대하여 설명하였다. 세부시스템에 고장이 발생하는 경우 상태 방정식 혹은 측정 방정식을 표현하는 방법에 변화가 있었으나 고장의 일반화가 가능함을 알 수 있다. 시스템에 고장이 발생하는 경우 상태 방정식과 측정 방정식은 각각 다음의 (21)과 (22)로 표현할 수 있다.

$$\dot{x}(t) = Ax(t) + Bu(t) + F\mu \tag{21}$$

$$y(t) = Cx(t) \tag{22}$$

여기서 $F\mu$ 는 일반적인 고장입력이며, F 는 알고 있는 변수이며, μ 는 모르고 있는 변수이다.

5. 결론

본 자료에서는 고장검출에 관련된 기본적인 개념들과 현재 해석적 중복기법으로 많이 활용되고 있는 고장검출 필터의 간단한 수학적 모델을 살펴보았다. Fault Detection은 고장검출이라고 하며 시스템에 고장이 발생할 경우 이를 신속하게 검출하는데 그 목적이 있다. Fault Isolation은 고장분리라고 하며 발생한 고장을 분리하는데 그 목적이 있다. 고장검출 기법은 크게 하드웨어 중복기법과 해석적 중복기법으로 구분된다. 해석적 고장검출 기법은 크게 신호 검출기법, 파라미터 검출기법, 고장검출 필터 활용기법, 패리티 공간 활용기법으로 구분할 수 있다. 고장검출 필터를 활용하는 경우 구동기, 센서, 시스템의 동역학적 특성 변화에 따라 각각 재구성된 상태방정식과 측정방정식을 활용할 수 있으나, 고장에 대한 일반적인 시스템 모델을 활용할 수 있음을 수학적 고찰을 통하여 살펴 보았다.

저자 약력



이은성

- 1973년 3월 20일생
- 1996년 건국대 항공우주공학과 (공학사).
- 1998년 건국대 기계공학과 (석사).
- 2005년 건국대 항공우주공학과 (박사).
- 2006년~2007년 UCLA 박사후연수연구원.
- 2007년~현재: 한국항공우주연구원 위성항법항법팀 (선임연구원).

· 관심분야: 위성항법 시스템의 고장검출, 항공용 위성항법 보강시스템, RTK (Real Time Kinematics)