

로봇 매니플레이터의 실시간 특이점 회피를 위한 작업 재구성 법: 동적 작업 우선도에 기초한 해석

Task Reconstruction Method for Real-Time Singularity Avoidance for Robotic Manipulators : Dynamic Task Priority Based Analysis

김진현, 정완균*, 최영진
(Jinhyun Kim, Wan Kyun Chung, and Youngjin Choi)

Abstract : There are several types of singularities in controlling robotic manipulators: kinematic singularity, algorithmic singularity, semi-kinematic singularity, semi-algorithmic singularity, and representation singularity. The kinematic and algorithmic singularities have been investigated intensively because they are not predictable or difficult to avoid. The problem with these singularities is an unnecessary performance reduction in non-singular region and the difficulty in performance tuning. In this paper, we propose a method of avoiding kinematic and algorithmic singularities by applying a task reconstruction approach while maximizing the task performance by calculating singularity measures. The proposed method is implemented by removing the component approaching the singularity calculated by using singularity measure in real time. The outstanding feature of the proposed task reconstruction method (TR-method) is that it is based on a local task reconstruction as opposed to the local joint reconstruction of many other approaches. And, this method has dynamic task priority assignment feature which ensures the system stability under singular regions owing to the change of task priority. The TR-method enables us to increase the task controller gain to improve the task performance whereas this increase can destabilize the system for the conventional algorithms in real experiments. In addition, the physical meaning of tuning parameters is very straightforward. Hence, we can maximize task performance even near the singular region while simultaneously obtaining the singularity-free motion. The advantage of the proposed method is experimentally tested by using the 7-dof spatial manipulator, and the result shows that the new method improves the performance several times over the existing algorithms.

Keywords : task reconstruction, kinematic singularity, algorithmic singularity, singularity avoidance, dynamic task priority, redundant manipulator.

I. 서론

특이점은 직렬, 병렬로봇 혹은 여유 자유도 이거나 아닌 로봇 등 모든 형태의 로봇 매니플레이터 제어 문제에서 발생하는 본질적인 성질이다. 이제까지의 이전 연구에서는 다섯 종류의 특이점이 보고되었고, 그것들은 각각 기구학적 특이점(kinematic singularity, KS), 알고리즘적 특이점(algorithmic singularity, AS)[1], [2], 준-기구학적 특이점(semi-kinematic singularity, SKS)[3], 준-알고리즘적 특이점(semi-algorithmic singularity, SAS)[4], 표현적 특이점(representation singularity, RS)[5] 이다. 이들에 대한 예제를 그림 1에 나타내었다. RS는 오일러 각도(Euler angle) 표현법 등을 사용하는 경우에 많이 발생하지만 Quaternion이나 지수형(Exponential map) 각도 표현 방법 등을 사용함으로써 회피가 가능하고, SKS와 SAS는 주로 관절 제한이나 기타 하드웨어 제한 등에 의해서 발생하기 때문에 예측이 가능하며, 이를 바탕으로 설계 단계에서 고려가 가능하다. 이에 반해, KS와 AS는 발생하는 상황에 대한 예측이 힘들고, 설계 단계에서 피한다는 것도 어렵기 때문에 적절한 알고리즘을 이용하여 회피하는 것이 바람직하다. 따라서, 본 논문에서는

KS와 AS의 회피 문제에 초점을 맞추고자 한다[6-22].

KS에 대해서는 이제까지 많은 회피 알고리즘이 제안되었다. 대표적인 방법들을 나열해 보면, 특이점에 강인한 역행렬 기법(singularity-robust inverse method, SRI)[7-9], [11], [16], 경로 추적 기법(path tracking method)[12], [23], [24], 표준형 접근법(normal form approach)[18] 등이 있다. 위와 같은 방법들 중에서 SRI 방법은 쉽게 적용이 가능하다는 측면 때문에 가장 널리 쓰이고 있다. 그러나, 이득 조절의 방법이 모호하며, 특이점이 아닌 영역에서의 불필요한 큰 작업 오차를 유발한다. 경로 추적 기법은 먼저 원래 시스템보다 한 차원 더 복잡한 시스템을 다루어야 하고, 특이 영역으로 로봇이 진입하게 되면 속도를 줄여야 하며 특이점에 도달하였을 경우에는 완전히 멈추어야 하는 단점이 있다. 표준형 접근법은 특이점에 대한 정확한 정보를 미리 알고 있어야 한다는 점에서 실용성이 떨어진다.

AS 회피 기법에 대해서는 KS에 비해 비교적 적은 수의 알고리즘이 제안되었다[6], [16], [20]. Chiaverini가 제안한 최적화 기법[16]은 알고리즘적 특이점을 포함하지 않는 장점에 비해 두 번째 우선도의 작업 오차가 불필요하게 매우 커진다는 단점 때문에 사용에 제약이 있다. 두 번째 우선도 작업 오차를 향상시키시고자 Park등은[20] 가중 Bordered Gramian 방법을 제안하였다. 하지만 새로운 특이점이 발생할 뿐만 아니라, 작업 오차 측면에서도 개선의 여지가 여전히 남아 있다.

작업 경로를 로봇이 위치한 상태에 따라 바꾸는 방법도 특이점 회피를 위한 하나의 선택이 될 수 있다. 이 방법을 이용

* 책임저자(Corresponding Author)

논문접수 : 2004. 5. 31., 채택확정 : 2004. 8. 31.

김진현, 정완균 : 포항공과대학교 기계공학과

(pluto@postech.ac.kr/wkchung@postech.ac.kr)

최영진 : 한국과학기술연구원 지능로봇연구센터(cyj@kist.re.kr)

※ 본 연구는 한국 과학기술부 국제공동연구 프로그램("소형 무인 잠수정/매니플레이터 시스템 개발", M6-0302-00-0009-03-A01-00-004-00)의 지원을 받아 수행되었음.

한 것으로 주어진 조건에 따라 사전에 경로를 생성하는 경로 계획 알고리즘(motion planning algorithm)이 [17], [25-28] 통해서 제안되었다. 하지만, 자율 구동 혹은 원격 작업 시 실시간 적용이라는 측면에서 경로 계획 알고리즘은 제약이 많다.

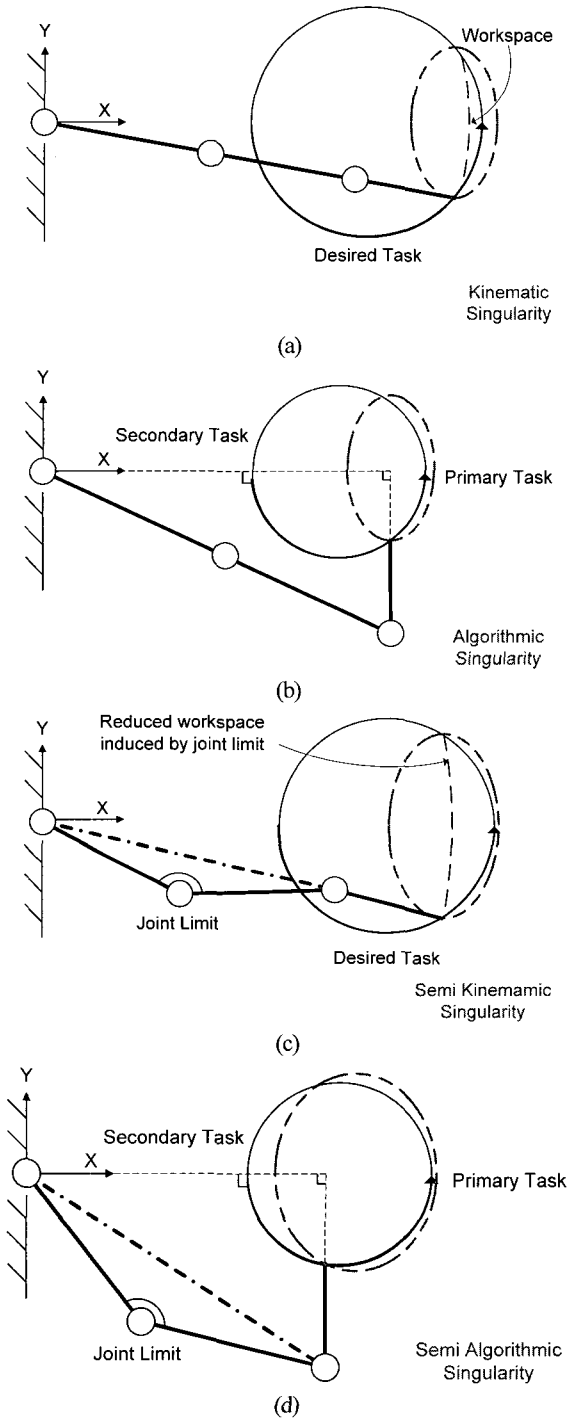


그림 1. 여러 가지 특이점의 예: (a) 기구학적 특이점(KS); (b) 알고리즘적 특이점(AS); (c) 준-기구학적 특이점(SKS); 준-알고리즘적 특이점(SAS).

Fig. 1. Examples on various singularities: (a) kinematic singularity (KS); (b) algorithmic singularity(AS); (c) semi-kinematic singularity(SKS); and, (d) semi-algorithmic singularity(SAS).

본 논문에서는 기구학적 특이점과 알고리즘적 특이점을 회피하기 위한 새로운 알고리즘을 제안한다. 이 방법은 작업 재구성 기법(Task Reconstruction method, TR-method)이라 불리며, 이 방법을 통해서 특이점 근처에서의 초기 작업 경로를 안전한 경로로 실시간으로 재구성하게 된다. TR-method를 적용하기 위해서 먼저 시스템이 특이점으로부터 얼마나 떨어져 있는지를 판단하기 위한 기준이 필요하다. 이러한 기준에 근거하여 특이점으로 향하는 방향 벡터를 구하고, 다음으로 원래의 경로 중에서 특이점으로 다가가는 성분을 제거하게 된다. 마지막으로 시스템이 정해진 기준 이하에 위치하게 될 경우 기준까지 회복하는 성분을 추가함으로써 작업을 안정하게 재구성할 수 있게 된다. 이러한 과정은 동적 작업 우선도 측면에서도 설명이 가능하다. 즉, 특이 영역에 가까이 다가감으로써 안정성이 깨지게 되면 기존의 경로 추종 작업에 우선해서 안정성을 높이기 위한 작업이 최우선 순위의 작업이 되어 안정성을 회복하도록 동작하게 된다. 안정성의 위험이 없는 특이 영역 밖에서는 원래의 작업이 최우선 순위 작업이 되어 작업 성능을 높이는 동작을 한다.

기존의 특이점 회피 알고리즘과 비교하여 제안된 작업 재구성 기법은 4가지 우수한 특징을 가지고 있다. 첫 번째로 TR-method는 기구학적 특이점뿐만 아니라 알고리즘적 특이점에도 적용이 가능하다. 두 번째로는 특이점을 회피하는 것뿐만 아니라 뛰어난 작업 추종 성능을 보여준다. 이는 대부분의 기존의 방법들이 국부적 관절 재구성(local joint reconstruction)에 기초하는데 반해 제안된 TR-method는 국부적 작업 재구성(local task reconstruction)에 기초하는 것에 기인한다. 이러한 특징 때문에 기존의 방법들에서는 시스템을 불안정하게 했던 높은 작업 이득 값(high task controller gain)을 제안된 방법에서는 작업 성능을 높이기 위해서 안정성을 해치지 않으면서 사용이 가능하다. 자세한 내용은 III.1 절에서 설명하기로 한다. 세 번째 특징은 TR-method에서 성능 향상을 위한 이득 조절의 방법이 특이점 지수(singularity measure)에 근거하기 때문에 명확한 물리적 의미를 지니고, 따라서 이득 조절이 쉽다는 점이다. 마지막 특징은 TR-method를 작업 우선도에 기초한 다수의 작업에서의 특이점 회피를 위한 알고리즘으로 확장이 가능하다는 점이다. 즉, 제안된 방법을 이용하여 다수의 작업이 우선도를 가지고 주어졌을 때 모든 기구학적, 알고리즘적 특이점을 귀납적인 방법을 통해서 회피할 수 있다.

본 논문은 다음과 같은 순서로 구성되어 있다. II 장에서는 기본적인 수학 배경에 대해서 설명이 주어지고, III 장에서는 단일 작업에서의 기구학적 특이점을 회피하기 위한 TR-method의 적용에 대한 설명과 이득 조절에 대해서 설명한다. 또한, TR-method의 국부적 작업 재구성의 장점을 국부적 관절 재구성 방법과 비교 설명하고, 동적 작업 우선 순위의 관점에서 TR-method를 설명한다. IV장에서는 TR-method를 기구학적 특이점뿐만 아니라 알고리즘적 특이점 회피 기법까지 확장시킨다. V장은 다수의 작업에서의 모든 기구학적, 알고리즘적 특이점을 회피하기 위한 귀납적인 방법에 대해서 설명한다. VI장은 다음과 같은 세 가지 경우에 대한 실험결과를 보여준다; 1) 단일 작업에서의 기구학적 특이점 회피, 2)

두 개의 작업에서의 알고리즘적 특이점 회피, 3) 두 개의 작업에서의 기구학적, 알고리즘적 특이점 회피. 실험 결과는 기존의 방법을[16] 이용한 결과와 비교 설명한다. 마지막으로 결론에서 본 논문의 내용을 요약한다.

II. 기본 수학적 배경

1. 역기구학과 기구학적 특이점

일반적인 로봇 매니퓰레이터의 기구학적 출력 값은 조작 변수($\mathbf{r} \in \mathbb{R}^m$)로 표현된다. 조작 변수는 로봇의 위치나 자세, 조작성 지수(manipulability measure) 등 관절 변수($\mathbf{q} \in \mathbb{R}^n$)로 표현되는 어떠한 함수로도 표현 할 수 있다.

$$\mathbf{r} = \mathbf{f}(\mathbf{q}) \tag{1}$$

위 식에 대해서 작은 변화율을 적용시키면 $\delta\mathbf{r}$ 과 $\delta\mathbf{q}$ 에 대해서 다음과 같은 식이 얻어진다.

$$\delta\mathbf{r} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \delta\mathbf{q} = \mathbf{J}(\mathbf{q})\delta\mathbf{q} \tag{2}$$

여기서 $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ 는 조작 변수의 자코비안 행렬(Jacobian Matrix)이다. 변화 운동에 대한 변화율 제어(Resolved motion rate control, RMRC)를 통해 주어진 $\delta\mathbf{r}$ 와 \mathbf{q} 에 대해서 (2)와 같은 선형 시스템을 이용해서 $\delta\mathbf{q}$ 을 구할 수 있다[29]. 일반적으로 이것은 다음 식과 같은 자코비안 행렬의 의사역행렬(pseudo-inverse matrix)을 이용해서 나타낸다 [30].

$$\delta\mathbf{q} = \mathbf{J}^+(\mathbf{q})\delta\mathbf{r} + (\mathbf{I}_n - \mathbf{J}^+(\mathbf{q})\mathbf{J}(\mathbf{q}))\mathbf{y}, \quad (m \leq n) \tag{3}$$

여기서 $\mathbf{J}^+(\mathbf{q}) (= \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}) \in \mathbb{R}^{n \times m}$ 은 $\mathbf{J}(\mathbf{q})$ 의 Moore-Penrose 의사역행렬이며, $\mathbf{y} \in \mathbb{R}^n$ 은 임의의 벡터이고, $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ 은 단위 행렬을 나타낸다. 만약 여유자유도가 아닌 시스템이라면($m = n$), $\mathbf{J}^+(\mathbf{q})$ 는 $\mathbf{J}^{-1}(\mathbf{q})$ 와 동일하고, $(\mathbf{I}_n - \mathbf{J}^+(\mathbf{q})\mathbf{J}(\mathbf{q}))$ 는 $\mathbf{0}$ 이 된다.

(3)에서 기구학적 특이점은 관절 형태 벡터 \mathbf{q}^* 에 의해서 정의 되는데, 이 때 $\mathbf{J}(\mathbf{q}^*)$ 는 완전 열 랭크(full row rank)를 가지지 않는다. 따라서 의사역행렬 $\mathbf{J}^+(\mathbf{q}^*)$ 가 정의될 수 없고, 그로 인해 적절한 관절 속도 벡터를 생성할 수 없게 된다. 다시 말해, 특이점은 매니퓰레이터 자코비안의 상태에 의해서가 아니라, 시스템의 순간 기구학(instantaneous kinematics)의 불확실성에 의해 정의된다. 더욱이, 특이점 근처에서는 매우 작은 $\delta\mathbf{r}$ 의 변화에 대해서도 매우 큰 $\delta\mathbf{q}$ 의 변화를 요구하기 때문에 실제 매니퓰레이터에서 실현이 불가능하고, 또한 시스템에 손상을 줄 수 있다.

Damped least-squares(DLS) 방법을 이용하면 매우 고전적이면서도 간단하게 기구학적 특이점을 회피할 수 있다. 특이점 근처에서 다음과 같은 조정 항목을 추가함으로써 특이점 회피가 가능하다.

$$\mathbf{J}_{DLS}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda\mathbf{I})^{-1} \tag{4}$$

결과적으로 (2)의 DLS 해법은 다음과 같다.

$$\delta\mathbf{q}_{DLS} = \mathbf{J}_{DLS}^+(\mathbf{q})\delta\mathbf{r} + (\mathbf{I}_n - \mathbf{J}_{DLS}^+(\mathbf{q})\mathbf{J}(\mathbf{q}))\mathbf{y} \tag{5}$$

그러나, 이 방법의 가장 큰 단점은 완충 상수(damping constant, λ)의 도입으로 특이점 이외의 영역에서 작업 추종 오차가 커진다는 점이다[16]. 따라서, 이러한 단점을 극복하기 위해서 완충 상수를 임계 값에 따라 변화시키는 알고리즘이 도입되었고[7], Chiaverini는 (6), (7)과 같이 특이 값 분해(singular value decomposition)를 통해서 제일 작은 특이 값(σ_m)이 임계 값(ε) 이하가 될 경우에만 완충 상수를 도입하는 방법을 제안하였다[16]. 그러나, 완충 상수와 임계 값 설정이 어렵고, 여전히 특이점 이외의 지역에서의 오차에 대해 개선의 여지가 남아 있다.

$$\mathbf{J}_{DLS}^+ = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \lambda^2 \mathbf{u}_m \mathbf{u}_m^T)^{-1} \tag{6}$$

$$\lambda^2 = \begin{cases} 0, & \sigma_m \geq \varepsilon \\ \left(1 - \left(\frac{\sigma_m}{\varepsilon}\right)^2\right) \lambda_{\max}^2, & \sigma_m < \varepsilon \end{cases} \tag{7}$$

(6)에서 \mathbf{u}_m 은 최소의 특이 값과 관련된 출력 특이 벡터(output singular value)를 나타낸다.

2. 작업 우선도에 근거한 방법과 알고리즘적 특이점

Nakamura는 여유자유도 매니퓰레이터에서 작업 우선도에 기초하여 여러 작업들을 수행하기 위한 역기구학 해법을 제안하였다[30]. 조작 변수 $\mathbf{r}_1 \in \mathbb{R}^{m_1}$ 은 다음과 같이 표현되며, 첫 번째 우선 순위를 가지는 작업이라고 하자.

$$\mathbf{r}_1 = \mathbf{f}_1(\mathbf{q}) \tag{8}$$

그리고, (8)의 미분 관계식은 다음과 같이 주어진다.

$$\delta\mathbf{r}_1 = \mathbf{J}_1(\mathbf{q})\delta\mathbf{q} \tag{9}$$

여기서, $\mathbf{J}_1(\mathbf{q}) \in \mathbb{R}^{m_1 \times n}$ 은 첫 번째 조작 변수 \mathbf{r}_1 의 자코비안 행렬이다. 만약 첫 번째 작업을 할당하고도 여분의 자유도가 주어진다면, 위와 비슷하게 $\mathbf{r}_2 \in \mathbb{R}^{m_2}$ 을 다음과 같은 관계를 가지는 두 번째 우선 순위 작업이라고 하자.

$$\mathbf{r}_2 = \mathbf{f}_2(\mathbf{q}) \tag{10}$$

$$\delta\mathbf{r}_2 = \mathbf{J}_2(\mathbf{q})\delta\mathbf{q} \tag{11}$$

여기서, $\mathbf{J}_2(\mathbf{q}) \in \mathbb{R}^{m_2 \times n}$ 은 두 번째 조작 변수 \mathbf{r}_2 의 자코비안 행렬이다.

(9)은 $\delta\mathbf{q}$ 에 대해서 다음과 같이 무한 해를 가지게 된다.

$$\delta\mathbf{q} = \mathbf{J}_1^+(\mathbf{q})\delta\mathbf{r}_1 + (\mathbf{I}_n - \mathbf{J}_1^+(\mathbf{q})\mathbf{J}_1(\mathbf{q}))\mathbf{y} \tag{12}$$

여기서 (12)을 (11)에 대입하면, 다음과 같은 식을 얻을 수 있다.

$$\mathbf{J}_2(\mathbf{I}_n - \mathbf{J}_1^+\mathbf{J}_1)\mathbf{y} = \delta\mathbf{r}_2 - \mathbf{J}_2\mathbf{J}_1^+\delta\mathbf{r}_1 \tag{13}$$

만약 \mathbf{y} 에 대해서 정확한 해가 존재한다면 (13)을 통해 두

번째 조작 변수를 구현할 수 있다. 그러나 일반적으로 정확한 해는 존재하지 않고, $\|\delta \mathbf{r}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \delta \mathbf{r}_1\|$ 을 최소화하는 최소 자승 법(least square method)에 근거한 의사역행렬을 이용해서 다음과 같은 식을 얻을 수 있다.

$$\mathbf{y} = \hat{\mathbf{J}}_2^+ (\delta \mathbf{r}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \delta \mathbf{r}_1) \quad (14)$$

$$\hat{\mathbf{J}}_2 = \mathbf{J}_2 (\mathbf{I}_n - \mathbf{J}_1^+ \mathbf{J}_1) \quad (15)$$

따라서 (14)를 (12)에 대입함으로써 작업 우선도에 기초한 역기구학 해법을 다음과 같이 얻을 수 있다.

$$\delta \mathbf{q} = \mathbf{J}_1^+(\mathbf{q}) \delta \mathbf{r}_1 + \hat{\mathbf{J}}_2^+ (\delta \mathbf{r}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \delta \mathbf{r}_1) \quad (16)$$

새 개 혹은 더 많은 작업들이 주어진 경우에, i 번째 작업에 대해서 다음과 같이 규격화 할 수 있다.

$$\delta \mathbf{r}_i = \mathbf{J}_i(\mathbf{q}) \delta \mathbf{q}, \quad \text{for } i = 1, 2, \dots, t \quad (17)$$

여기서 $\mathbf{J}_i(\mathbf{q})$ 은 i 번째 작업의 자코비안 행렬이고, i 번째 작업은 이전 ($i - 1$) 번째까지의 작업들에 대해서 낮은 우선순위를 갖는다.

이전 방식으로부터 다음과 같은 t 개의 작업에 대한 일반적인 작업 우선도에 근거한 역기구학 해법을 구할 수 있다 [31], [32].

$$\begin{cases} \delta \mathbf{q}_i = \delta \mathbf{q}_{i-1} + \hat{\mathbf{J}}_i^+ \delta \hat{\mathbf{r}}_i \\ \delta \hat{\mathbf{r}}_i = \delta \mathbf{r}_i - \mathbf{J}_i \delta \mathbf{q}_{i-1} \\ \mathbf{P}_i = \mathbf{P}_{i-1} - \hat{\mathbf{J}}_i^+ \hat{\mathbf{J}}_i \\ \hat{\mathbf{J}}_i = \mathbf{J}_i \mathbf{P}_{i-1} \end{cases}, \quad \begin{cases} \delta \mathbf{q}_0 = \mathbf{0} \\ \mathbf{J}_0 = \mathbf{0} \\ \mathbf{P}_0 = \mathbf{I}_n \end{cases} \quad (18)$$

여기서 $\delta \mathbf{q}_i$ 는 귀납식에 의해서 얻어지는 최종 관절 변수 값이고, \mathbf{P}_i 는 투영 행렬이다. (18)에서 만약 어떠한 $\hat{\mathbf{J}}_i (= \mathbf{J}_i \mathbf{P}_{i-1})$ 라도 랭크를 잃게 되면, 기구학적 또는 알고리즘적 특이 조건을 의미한다. 즉, \mathbf{J}_i 가 랭크를 잃게 되면 기구학적 특이 조건, \mathbf{J}_i 는 잃지 않으면서 $\hat{\mathbf{J}}_i$ 이 잃게 되면 알고리즘적 특이 조건을 의미한다.

알고리즘적 특이점을 피하기 위해 (12)에 대해서 Chiaverini 는 $\|\delta \mathbf{r}_2 - \mathbf{J}_2 \mathbf{J}_1^+ \delta \mathbf{r}_1\|$ 을 최소화 하기 위한 최소 자승 법 대신 $\|\mathbf{J}_2^+ \delta \mathbf{r}_2 - \mathbf{J}_1^+ \delta \mathbf{r}_1\|$ 을 최소화 하는 방법으로 다음과 같이 새로운 작업 우선도 식을 유도하였다[16].

$$\delta \mathbf{q} = \mathbf{J}_1^+(\mathbf{q}) \delta \mathbf{r}_1 + (\mathbf{I}_n - \mathbf{J}_1^+ \mathbf{J}_1) \mathbf{J}_2^+ \delta \mathbf{r}_2 \quad (19)$$

그러나, (19)가 알고리즘적 특이점을 포함하지 않는 대신에 두 번째 우선 순위의 작업에 대한 작업 오차가 매우 커진다는 단점이 있다[20].

3. 특이점 지수(Singularity Measures)

어떤 매니퓰레이터의 형상이 특이점이거나 혹은 특이점에 가까우냐 하는 것을 판단하기 위해서 특이점 지수가 필요하다. 기구학적 특이점에 대해서는 조작성 지수(manipulability

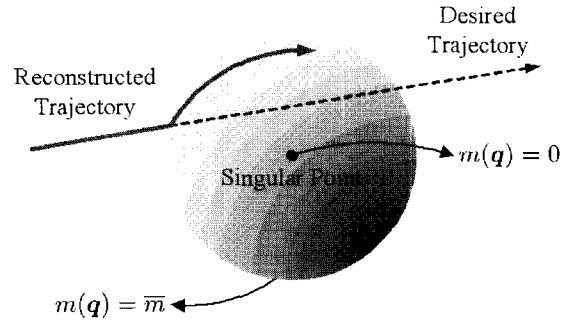


그림 2. TR-method의 개념도.
Fig. 2. Conceptual diagram of TR-method.

measure[33])가 널리 알려진 특이점 지수이다. 알고리즘적 특이점에 대해서는 O'Neil 등이 두 번째 작업의 자코비안과 첫 번째 작업의 영 기저 행렬(null basis matrix)을 이용한 지수를 사용하였다[34]. 그러나 O'Neil이 사용한 알고리즘적 특이점 지수는 두 번째 작업의 기구학적 특이점 역시 포함하고 있다. 본 논문에서 사용하게 될 지수들은 이어지는 장에서 설명하기로 한다.

이제까지 기구학적 특이점과 알고리즘적 특이점에 대해서 기본적인 설명을 하였고, 또한 그것들을 회피하기 위한 기본적인 접근법에 대해서 살펴보았다. 이어지는 장에서는 기구학적 특이점과 알고리즘적 특이점을 모두 회피할 수 있는 새로운 방법을 제안하고 설명하고자 한다.

III. 단일 작업에서의 작업 재구성 기법

이번 장에서는 TR-method에 대한 개념을 설명하고, 단일 작업에서 기구학적 특이점을 피하기 위해 TR-method를 적용한다.

그림 2는 특이점을 통과하는 경로를 온라인으로 재구성하는 TR-method에 대한 주요 개념을 보여준다. TR-method는 다음과 같은 단계로 작업을 재구성하게 된다.

- 1) 특이점 지수의 정의
- 2) 특이점 지수의 등고선을 나타내는 벡터를 관절 공간이 아닌 작업 공간상에서 계산
- 3) 주어진 경로에서 실시간으로 위험 성분 제거
- 4) 복원 성분 추가

첫 번째와 두 번째 단계는 특이 영역과 그 영역으로부터 매니퓰레이터가 얼마나 멀리 떨어져 있는냐를 정의한다. 또, 특이 영역으로 향하는 벡터도 계산한다. 다음으로 세 번째와 네 번째 단계를 통해서 작업 재구성을 수행하게 된다. 자세한 유도과정은 아래에 나타내었다.

(3)에 나타낸 역기구학 관계를 다시 생각해 보자.

$$\delta \mathbf{q} = \mathbf{J}^+(\mathbf{q}) \delta \mathbf{r} + (\mathbf{I}_n - \mathbf{J}^+(\mathbf{q}) \mathbf{J}(\mathbf{q})) \mathbf{y}$$

여기서, $\mathbf{J}(\mathbf{q})$ 가 항상 완전 열 랭크(rank)를 가진다고 가정하자. 다시 말해 시스템은 결코 특이 영역에 들어가지 않는다고 가정하자. 다소 엄격한 가정이지만, TR-method가 제대로 동작한다면 이 가정은 항상 만족한다. 그리고 앞으로의 전개에서는 일반성을 잃지 않으면서 (3)에서 일반 해만을 사용한

다. 즉, $\mathbf{y} = \mathbf{0}$ 이고 $\delta\mathbf{q} = \mathbf{J}^+(\mathbf{q})\delta\mathbf{r}$ 이다.

먼저 특이점 지수를 $m(\mathbf{q})$ 로 정의 한다. 본 논문에서는 조작성 지수 ($m(\mathbf{q}) \triangleq \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$) [33]를 단일 작업에서의 기구학적 특이점 지수로 사용한다. $m(\mathbf{q})$ 에 대해서 영의 값은 특이점을 나타내고 큰 값이 특이점으로부터 멀리 떨어져 있는 상태를 나타낸다. 주어진 경로를 특이점 지수를 이용해서 재구성하기 위해 작업 변수를 통해 특이점 지수의 등고선을 표현해야 한다. 그것은 $\delta m(\mathbf{q}) = 0$ 와 같이 지수의 작은 변화율을 취함으로써 얻을 수 있다. $m(\mathbf{q})$ 의 작은 변화율은 다음과 같이 계산된다.

$$\delta m(\mathbf{q}) = \frac{\partial m(\mathbf{q})}{\partial \mathbf{q}} \delta \mathbf{q} = \frac{\partial m(\mathbf{q})}{\partial \mathbf{q}} \mathbf{J}^+ \delta \mathbf{r} \quad (20)$$

(20)에서 $\delta m(\mathbf{q}) = 0$ 을 만족하기 위해 주어진 작업 $\delta \mathbf{r}$ 은 반드시 다음의 벡터와 직교해야 한다.

$$\mathbf{d} \triangleq \left(\frac{\partial m(\mathbf{q})}{\partial \mathbf{q}} \mathbf{J}^+ \right)^T \quad (21)$$

또는, 동일하게 $\delta \mathbf{r}$ 이 다음과 같이 정의되는 표면에 놓여 있어야 한다.

$$\left\{ \mathbf{x} \in \mathbb{R}^m : \left(\frac{\partial m(\mathbf{q})}{\partial \mathbf{q}} \mathbf{J}^+ \right)^T \cdot \mathbf{x} = 0 \right\} \quad (22)$$

주어진 작업에서 특이점으로 향하는 성분을 제거하기 위해서 표면에 수직인 벡터에 대한 관계식이 필요하고, \mathbf{n}_m 을 표면에 수직인 단위 벡터로 다음과 같이 정의한다.

$$\mathbf{n}_m = \mathbf{d} / \|\mathbf{d}\| \quad (23)$$

이제 특이점으로 향하는 성분은 주어진 작업을 (22)에서 정의된 표면으로 투영시킴으로써 다음 식과 같이 제거할 수 있다.

$$\begin{aligned} \delta \mathbf{r}_p &= \delta \mathbf{r} - (\delta \mathbf{r} \cdot \mathbf{n}_m) \mathbf{n}_m \\ &= \delta \mathbf{r} - (\mathbf{n}_m \mathbf{n}_m^T) \delta \mathbf{r} = (\mathbf{I}_m - \mathbf{n}_m \mathbf{n}_m^T) \delta \mathbf{r} \end{aligned} \quad (24)$$

그러나, 주어진 작업의 성능 향상을 위해서 투영은 미리 정의된 특이 영역으로 가까워질 경우에만 이루어져야 한다. 그러기 위해, (24)에 다음과 같은 가중치를 도입한다.

$$\delta \mathbf{r}_p = (\mathbf{I}_m - k_1(m) \mathbf{n}_m \mathbf{n}_m^T) \delta \mathbf{r} \quad (25)$$

여기서 $k_1(m)$ 은 안전 영역에서는 0의 값을 가지고, 정의된 특이 영역, \bar{m} , 이하에서는 1을 가지는 형상 함수이다. 이 두 영역의 중간에서는 연속성을 고려한 함수의 형태를 지닌다. 본 논문에서는 (26)와 같은 식으로 형상 함수 $k_1(m)$ 을 사용하였다.

$$k_1(m) = \begin{cases} 0, & \bar{m} + \sigma < m \\ c(m), & \bar{m} < m \leq \bar{m} + \sigma \\ 1, & m \leq \bar{m} \end{cases} \quad (26)$$

여기서 $c(m) = 2 \left(\frac{m - \bar{m}}{\sigma} \right)^3 - 3 \left(\frac{m - \bar{m}}{\sigma} \right)^2 + 1$ 이다.

여기에 덧붙여서 $m(\mathbf{q})$ 가 작아지는 경우에만 투영 시키는 알고리즘을 사용하여야 한다. 따라서, 다음 식과 같이 $\delta \mathbf{r} \cdot \mathbf{n}_m$ 이 음수일 경우에만 투영시키도록 한다.

$$\begin{aligned} \delta \mathbf{r}_p &= \left(\mathbf{I}_m - \frac{1 - \text{sign}(\delta \mathbf{r} \cdot \mathbf{n}_m)}{2} k_1 \mathbf{n}_m \mathbf{n}_m^T \right) \delta \mathbf{r} \\ &= (\mathbf{I}_m - k'_1 \mathbf{n}_m \mathbf{n}_m^T) \delta \mathbf{r} \end{aligned} \quad (27)$$

여기서 $k'_1 = k_1(1 - \text{sign}(\delta \mathbf{r} \cdot \mathbf{n}_m))/2$ 이다. 따라서 (27)을 이용해 원래의 작업 중에서 위험 성분을 제거할 수 있다.

그러나, 여기서 주의할 점은 (27)은 $m(\mathbf{q})$ 가 이미 특이 영역인 \bar{m} 이하에 있을 경우에는 특이 영역에서 빠져 나오는 것을 보장하지 못한다. 수치적 오차나 제어 입력 주파수의 한계 등으로 인해 \bar{m} 이하로 내려가는 경우가 발생할 수 있기 때문에, 이것을 극복하기 위해 다음과 같은 추가적인 성분이 필요하다.

$$\delta \mathbf{r}_p = (\mathbf{I}_m - k'_1 \mathbf{n}_m \mathbf{n}_m^T) \delta \mathbf{r} + k_2 \mathbf{n}_m \quad (28)$$

$$k_2(m) = \begin{cases} 0, & \bar{m} < m \\ K_r(\bar{m} - m), & m \leq \bar{m} \end{cases} \quad (29)$$

여기서 K_r 은 상수 이득 값이다. (28)의 마지막 항은 m 이 \bar{m} 이하로 작아지게 되면 \bar{m} 의 표면으로 복원시키는 역할을 한다. 따라서, 재구성된 작업은 (28)로 주어지고, 다음과 같은 심볼로 표현한다.

$$\delta \mathbf{r}_p = \text{TR-Process}(\mathbf{J}, m(\mathbf{q}), \delta \mathbf{r}) \quad (30)$$

전체적인 재구성 단계의 개략도를 그림 3에 나타내었다.

TR-method에서는 (26)와 (29)에 포함된 3개의 성능 조절 변수들이 있다. 그것들은 각각 \bar{m} , σ , K_r 이다. \bar{m} 는 안정도를 결정하고 특이 영역의 크기를 나타내는 임계 값이며, σ 는 TR-method에서 위험 성분을 제거하는 범위를 의미하며, K_r 은 위험 영역에 들어갔을 경우 벗어나기 위한 성능 조절 이득 값을 의미한다. 각각의 변수들은 시행착오적인(trial-and-error) 방법을 적용시켜 최적의 값을 찾아야 하지만, 주어진 로봇의 최대 작업 공간의 3~5% 정도의 \bar{m} 와 \bar{m} 와 동일한 수준의 σ 을 사용하면 작업 성능과 안정성을 모두 일정 수준 확보할 수 있다. 예를 들어 1m의 작업 공간을 가지는 로봇의 경우 \bar{m} 은 0.03~0.05수준에서 초기 값을 설정할 수 있다. 작은 \bar{m} 을 사용하면 작업 성능을 높일 수 있지만 0에 가까운 너무 작은 값을 사용하면 수치 에러 등에 의해서 특이점에 빠질 수 있다. 그림 4는 성능 조절 변수들의 물리적인 의미를 보여준다.

1. 기존 DLS 방법의 성능 한계

널리 사용되고 있는 기구학적 특이점 회피 알고리즘의 대부분은 국부적 관절 재구성에 기초한다. 다시 말해 주어진 작업 중 특이점 부근에 위치한 작업들은 변형된 자코비안에 의해서 관절 경로로써 재구성된다. 특히 특이 영역을 빠져

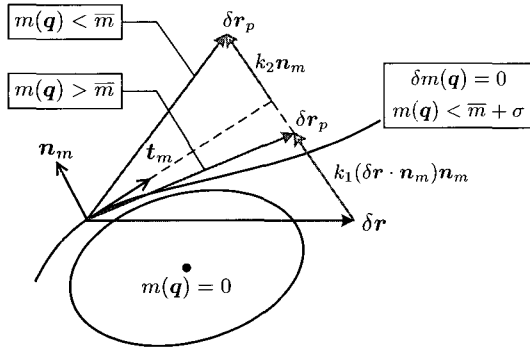


그림 3. 작업 재구성의 개략도.
Fig. 3. Schematic diagram of task reconstruction.

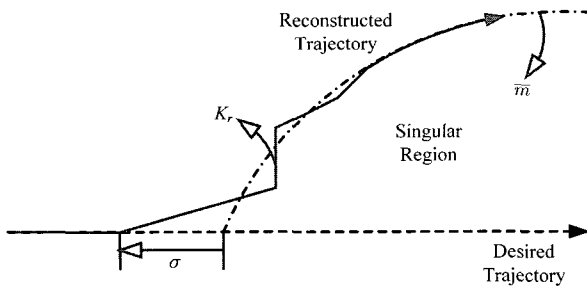


그림 4. 작업 재구성의 개략도.
Fig. 4. Schematic diagram of task reconstruction.

나온 후 제어기는 비 특이 영역 제어기로 바뀌게 된다. 이때 매니플레이터가 주어진 경로에서 많이 벗어나 있다면, 실제 시스템에서 큰 작업 오차에 의해 시스템이 발산할 수도 있다.

예를 들면 DLS 방법에서 작업 오차는 주로 (7)에 포함된 특이 영역의 크기를 결정하는 임계 값 ϵ 에 의해 발생한다. 큰 ϵ 값은 특이 영역을 통과한 후 비 특이 영역 제어기로 바뀔 때 큰 작업 오차에 의해서 overshoot이나 undershoot 현상을 유발한다. 그것은 마치 큰 계단 입력(step input)이 외란으로 작용한 상태에서 큰 작업 이득이 적용되었을 때의 현상과 유사하다. 즉, 실제 실험에서 큰 작업 이득을 사용했을 경우 시스템이 발산할 가능성이 커진다. 따라서 이러한 현상을 방지하기 위해 작은 작업 이득을 사용하게 되면 잔류 오차는 오랫동안 지속된다. 반면에 작은 ϵ 값은 수치적 오차나 제어 입력 주파수의 한계 등으로 발산하기가 쉽다. 따라서 작업 추종 성능을 높이는 것과 안정성을 높이는 것은 국부적 관점 재구성 방법에서의 딜레마로 작용한다. 그것은 기존의 방법에서는 작업이 특이 영역을 통과한 후에 어디로 향하는지 고려하지 않기 때문이다.

반면에 TR-method는 국부적 작업 재구성 방법에 기초한다. 비록 특이 영역에서 주어진 작업에 대해 실제 작업은 오차를 가지지만 내부적으로 제어기는 재구성된 작업을 추종한다. 따라서 작업 오차는 특이 영역을 통과한 후에도 매우 작게 되고, 큰 작업 이득 값은 계단 입력 반응과 같은 오차 대신 작업 오차를 줄이는 역할을 한다. 이것이 제안된 TR-method의 주요한 장점이며 실험을 통해서 성능을 검증할 것이다.

2. 동적 작업 우선도에 기초한 해석

이번 절에서는 (28)을 작업 우선도의 관점에서 설명하고자 한다. 먼저 (23)의 \mathbf{n}_m 을 이용해서 자코비안 행렬 \mathbf{J}_{r1} 을 다음과 같이 정의하자.

$$\mathbf{J}_{r1} \triangleq \mathbf{n}_m^T \tag{31}$$

그러면 \mathbf{J}_{r1} 의 Moor-Penrose 의사역행렬은 다음과 같이 유도될 수 있다.

$$\begin{aligned} \mathbf{J}_{r1}^+ &= \mathbf{J}_{r1}^T (\mathbf{J}_{r1} \mathbf{J}_{r1}^T)^{-1} = \mathbf{n}_m (\mathbf{n}_m^T \mathbf{n}_m)^{-1} \\ &= \mathbf{n}_m = \mathbf{J}_{r1}^T \end{aligned} \tag{32}$$

따라서, (28)을 다음과 같이 표현할 수 있다.

$$\delta \mathbf{r}_p = (\mathbf{I}_m - k_1' \mathbf{J}_{r1}^+ \mathbf{J}_{r1}) \delta \mathbf{r} + \mathbf{J}_{r1}^+ k_2 \tag{33}$$

여기서 m 의 상태에 따라 (33)는 다음과 같이 나누어 볼 수 있다.

$$\delta \mathbf{r}_p = \begin{cases} \delta \mathbf{r}, & m > \bar{m} + \sigma \\ (\mathbf{I}_m - \mathbf{J}_{r1}^+ \mathbf{J}_{r1}) \delta \mathbf{r} + \mathbf{J}_{r1}^+ k_2', & m \leq \bar{m} + \sigma \end{cases} \tag{34}$$

위 식에서 $k_2' = (1 - k_1')(\delta \mathbf{r} \cdot \mathbf{n}_m) + k_2$ 이다.

다음으로 k_2' 와 $\delta \mathbf{r}$ 을 각각 첫 번째와 두 번째 우선 순위를 가지는 작업이라고 가정하자. 그러면 \mathbf{J}_{r1} 은 첫 번째 작업의 자코비안 행렬이 되고, 두 번째 작업의 자코비안 \mathbf{J}_{r2} 는 \mathbf{I}_m 인 단위 행렬이라고 할 수 있다. 일반적인 작업 우선 순위에 근거한 해를 다음과 같이 기술하자.

$$\delta \mathbf{r}_p = \mathbf{J}_{r1}^+ k_2' + \hat{\mathbf{J}}_{r2} (\delta \mathbf{r} - \mathbf{J}_{r2} \mathbf{J}_{r1}^+ k_2') \tag{35}$$

여기서 $\hat{\mathbf{J}}_{r2} = \mathbf{J}_{r2} (\mathbf{I}_m - \mathbf{J}_{r1}^+ \mathbf{J}_{r1})$ 이고, 보다 간단하게 다음과 같이 나타낼 수 있다.

$$\begin{aligned} \hat{\mathbf{J}}_{r2} &= \mathbf{J}_{r2} (\mathbf{I}_m - \mathbf{J}_{r1}^+ \mathbf{J}_{r1}) \\ &= \mathbf{I}_m - \mathbf{J}_{r1}^+ \mathbf{J}_{r1} \end{aligned} \tag{36}$$

따라서, $\hat{\mathbf{J}}_{r2}$ 가 멱등원(idempotent)이며 대칭(symmetric)이라는 성질을 쉽게 보일 수 있고, 이 경우 $\hat{\mathbf{J}}_{r2}$ 는 $\hat{\mathbf{J}}_{r2}$ 와 동일하다는 것을 쉽게 유도할 수 있다. 따라서, (35)은 다음과 같이 정리된다.

$$\begin{aligned} \delta \mathbf{r}_p &= \mathbf{J}_{r1}^+ k_2' + (\mathbf{I}_m - \mathbf{J}_{r1}^+ \mathbf{J}_{r1}) (\delta \mathbf{r} - \mathbf{J}_{r1}^+ k_2') \\ &= \mathbf{J}_{r1}^+ k_2' + (\mathbf{I}_m - \mathbf{J}_{r1}^+ \mathbf{J}_{r1}) \delta \mathbf{r} \end{aligned} \tag{37}$$

여기서, (37)은 (34)의 $m \leq \bar{m} + \sigma$ 일 경우의 식과 동일하다는 것을 알 수 있다. 이 사실로부터, $m \leq \bar{m} + \sigma$ 일 경우, 다시 말해 작업이 정의된 특이 영역으로 진입했을 경우, 작업 우선 순위에 근거해서 특이 영역을 빠져 나오려고 하는 작업이 최우선 순위가 되어 시스템의 안정성을 높이는 데 주력하고, 주어진 작업은 두 번째 우선 순위의 작업으로 할당되는 것이다. 반면에 $m > \bar{m} + \sigma$ 인 경우에는 특이 영역 밖의 영역이므로, 주어진 작업이 첫 번째 우선 순위의 작업

이 되어 작업 성능을 높이기 위한 노력이 우선시 되고, 그 외의 작업은 할당되지 않는다.

정리해보면 제안된 TR-method는 특이 영역으로의 진입 여부에 따라서 동적으로 작업 우선 순위가 할당되어 시스템의 작업 성능과 안정성을 모두 만족시키도록 작용한다는 것을 알 수 있다.

IV. 두 개의 작업에서의 작업 재구성 기법

이번 장에서는 두 개의 작업이 주어졌을 경우에 발생하는 특이점을 TR-method를 통해 회피하는 방법에 대해서 살펴보기로 한다. 이 경우 기구학적 특이점에 덧붙여 알고리즘적 특이점도 시스템에 영향을 미치게 된다. 예를 들면 (16)에서는 3개의 특이점이 존재한다. 첫 번째 작업의 기구학적 특이점, 두 번째 작업의 기구학적 특이점에 더해서 두 작업 사이의 작업 우선 순위에 의해 발생하는 알고리즘적 특이점이 그것들이다. 첫 번째 작업의 기구학적 특이점은 \mathbf{J}_1^+ 이 결정될 수 없을 때 발생한다. 비슷하게 $\hat{\mathbf{J}}_2^+$ 이 결정될 수 없을 때 두 번째 작업의 기구학적 특이점과 두 작업 사이의 알고리즘적 특이점이 발생한다. 이 두 특이점 들은 모두 $\hat{\mathbf{J}}_2^+$ 와 관련이 되어 있고, 개별적으로 다룰 수도 있다. 그러나 두 특이점들의 연관성은 명확하지 않고, 한 쪽의 좋은 관절 변수가 다른 특이점을 회피하기에는 안 좋은 경우일 수도 있다. 따라서 두 특이점을 개별적으로가 아니라 동시에 다루는 것이 보다 합리적이다.

앞 장에서 제시한 TR-method를 이용해서 $\hat{\mathbf{J}}_2$ 와 관계된 특이점을 회피하기 위해서 $\hat{\mathbf{J}}_2^+$ 에 연관되어 있는 변형된 두 번째 작업($\delta\hat{\mathbf{r}}_2 = \delta\mathbf{r}_2 - \mathbf{J}_2\mathbf{J}_1^+\delta\mathbf{r}_1$)을 재구성한다. 앞 장에서와 마찬가지로 TR-method를 사용하기 위해서 먼저 두 번째 작업에 관련된 특이점 지수가 필요하다. 두 번째 작업의 가장 합리적인 특이점 지수는 다음 식과 같은 $\hat{\mathbf{J}}_2$ 의 조작성 지수 형태이다[34].

$$m_2(\mathbf{q}) = \sqrt{\det\{\hat{\mathbf{J}}_2\hat{\mathbf{J}}_2^T\}} \quad (38)$$

그리고, (38)는 다음과 같이 자세히 전개할 수 있다.

$$\begin{aligned} \sqrt{\det\{\hat{\mathbf{J}}_2\hat{\mathbf{J}}_2^T\}} &= \sqrt{\det\{\mathbf{J}_2(\mathbf{I}_n - \mathbf{J}_1^+\mathbf{J}_1)\mathbf{J}_2^T\}} \\ &= \sqrt{\det\{\mathbf{J}_2\mathbf{Z}_1^+\mathbf{Z}_1\mathbf{J}_2^T\}} \quad (39) \\ &= \frac{\sqrt{\det\{\mathbf{J}_2\mathbf{Z}_1^T\mathbf{Z}_1\mathbf{J}_2^T\}}}{\sqrt{\det\{\mathbf{Z}_1\mathbf{Z}_1^T\}}} \end{aligned}$$

$$\sqrt{\det\{\mathbf{Z}_1\mathbf{Z}_1^T\}} = |\det\{\mathbf{J}_{1m}\}|^{(r-1)} \sqrt{\det\{\mathbf{J}_1\mathbf{J}_1^T\}} \quad (40)$$

$$\mathbf{J}_1 = \begin{bmatrix} \mathbf{J}_{1m} & \mathbf{J}_{1n-m} \end{bmatrix} \quad (41)$$

여기서 n 과 m 은 각각 관절 공간과 첫 번째 작업 공간의 차수이다. 그리고, r 은 $n - m$ 을 의미한다. \mathbf{Z}_1 은 \mathbf{J}_1 의 영거 공간 행렬을 의미하고, 다음과 같이 해석적인 형태로

표현할 수 있다[35].

$$\mathbf{Z}_1(\mathbf{q}) = \left[\mathbf{J}_{1n-m}^T \text{adj}\{\mathbf{J}_{1m}^T\} \quad -\det\{\mathbf{J}_{1m}\}\mathbf{I}_{n-m} \right] \quad (42)$$

TR-method에서는 특이점 지수뿐만이 아니라 그것의 미분 형태도 필요하다. 그러나, (39)의 미분 형태를 해석적으로 구하는 것은 매우 복잡하고 이로 인해 계산량의 증가로 실시간 제어가 어려워질 수 있다. 따라서, 본 논문에서는 다음과 같이 (39)의 분자 항만을 특이점 지수로 사용한다.

$$m_2(\mathbf{q}) \triangleq \sqrt{\det\{\mathbf{J}_2\mathbf{Z}_1^T\mathbf{Z}_1\mathbf{J}_2^T\}} \quad (43)$$

TR-method가 첫 번째 작업에 대해서 적용된 경우 (39)의 분모 항은 기준 값 이상으로 유지되고, 따라서 (39)는 (43)이 0 이 되지 않는 한 0이 되지 않는다. 그러므로, (43)이 정확하게 두 번째 작업의 특이점 지수는 아니지만 본 논문에서는 제한된 형태의 특이점 지수로서 사용한다. 실제 실험 결과 (39)를 사용하였을 경우와 비교하여 작업 성능이나 안정성 등에서 큰 차이를 보이지 않는다는 것을 확인하였다.

작업 우선도 기법의 필요 조건을 만족시키기 위해 두 개의 작업에 대한 TR-method는 다음과 같이 귀납적인 방법을 통해 적용된다.

$$\begin{aligned} \delta\mathbf{q} &= \delta\mathbf{q}_2 = \mathbf{J}_1^+\delta\mathbf{r}_{1p} + \hat{\mathbf{J}}_2^+\delta\hat{\mathbf{r}}_{2p} \\ &= \delta\mathbf{q}_1 + \hat{\mathbf{J}}_2^+\delta\hat{\mathbf{r}}_{2p} \end{aligned} \quad (44)$$

먼저, 첫 번째 우선 순위의 작업을 다음과 같이 TR-method를 통해서 재구성한다.

$$\delta m_1(\mathbf{q}) = \frac{\partial m_1(\mathbf{q})}{\partial \mathbf{q}} \delta\mathbf{q} = \frac{\partial m_1(\mathbf{q})}{\partial \mathbf{q}} \mathbf{J}_1^+ \delta\mathbf{r}_1 \quad (45)$$

$$\delta\mathbf{r}_{1p} = \text{TR-Process}(\mathbf{J}_1, m_1(\mathbf{q}), \delta\mathbf{r}_1) \quad (46)$$

$$\delta\mathbf{q}_1 = \mathbf{J}_1^+ \delta\mathbf{r}_{1p} \quad (47)$$

이와 비슷하게, 재구성된 첫 번째 작업을 이용하여 변형된 두 번째 작업에 대해서도 재구성할 수 있다.

$$\delta m_2(\mathbf{q}) = \frac{\partial m_2(\mathbf{q})}{\partial \mathbf{q}} \delta\mathbf{q} = \frac{\partial m_2(\mathbf{q})}{\partial \mathbf{q}} \hat{\mathbf{J}}_2^+ \delta\hat{\mathbf{r}}_2 \quad (48)$$

$$\delta\hat{\mathbf{r}}_2 = \delta\mathbf{r}_2 - \mathbf{J}_2\delta\mathbf{q}_1 \quad (49)$$

$$\delta\hat{\mathbf{r}}_{2p} = \text{TR-Process}(\hat{\mathbf{J}}_2, m_2(\mathbf{q}), \delta\hat{\mathbf{r}}_2) \quad (50)$$

따라서, 최종적인 TR-method의 형태는 다음과 같다.

$$\delta\mathbf{q} = \mathbf{J}_1^+(\mathbf{q})\delta\mathbf{r}_{1p} + \hat{\mathbf{J}}_2^+(\mathbf{q})\delta\hat{\mathbf{r}}_{2p} \quad (51)$$

그림 5는 위의 재구성 과정을 개략적인 블록선도로 나타낸 것이다.

두 개의 작업에 대한 TR-method를 적용한 결과는 세 개나 혹은 그 이상의 작업이 주어졌을 경우 귀납적인 식에 대한 아이디어를 제공한다. 따라서 작업 우선 순위에 근거한 일반적인 귀납적 TR-method를 다음 장에서 유도한다.

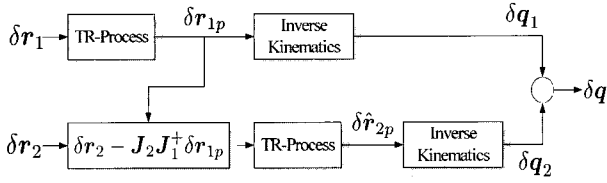


그림 5. 두 개의 작업 재구성의 블록선도.
Fig. 5. Block diagram of two subtasks reconstruction.

V. 여러 작업에 대한 귀납적 작업 재구성 기법

(18)에서 설명한 작업 우선 순위에 근거한 일반적인 귀납적 역기구학을 다시 생각해 보자. 만약 여러 작업에 대한 특이점 지수를 정의할 수 있다면, 여러 작업에서 발생할 수 있는 모든 특이점을 귀납적 TR-method의 테두리 안에서 해결할 수 있다.

(18)에서 $\hat{\mathbf{J}}_i (= \mathbf{J}_i \mathbf{P}_{i-1})$ 는 다음과 같이 영 기저 행렬을 이용해서 다시 표현할 수 있다.

$$\hat{\mathbf{J}}_i = \mathbf{J}_i \hat{\mathbf{Z}}_{i-1}^+ \hat{\mathbf{Z}}_{i-1} \quad (52)$$

여기서 $\hat{\mathbf{Z}}_i$ 는 다음의 식을 만족하는 영 기저 행렬이다.

$$\left[\mathbf{J}_1^T \mathbf{J}_2^T \cdots \mathbf{J}_i^T \right]^T \hat{\mathbf{Z}}_i^T = \mathbf{0}, \quad \hat{\mathbf{Z}}_0 = \mathbf{I}_n \quad (53)$$

그리고, $\hat{\mathbf{Z}}_i$ 은 (42)를 똑같이 적용하여 구할 수 있다.

따라서 i 번째 특이점 지수로 사용이 가능한 후보는 다음 식과 같다.

$$\begin{aligned} m_i(\mathbf{q}) &= \sqrt{\det \left\{ \mathbf{J}_i \hat{\mathbf{Z}}_{i-1}^+ \hat{\mathbf{Z}}_{i-1} \mathbf{J}_i^T \right\}} \\ &= \frac{\sqrt{\det \left\{ \mathbf{J}_i \hat{\mathbf{Z}}_{i-1}^T \hat{\mathbf{Z}}_{i-1} \mathbf{J}_i^T \right\}}}{\sqrt{\det \left\{ \hat{\mathbf{Z}}_{i-1} \hat{\mathbf{Z}}_{i-1}^T \right\}}} \end{aligned} \quad (54)$$

(54)는 총 i 개의 특이점을 포함한다. 즉, i 번째 작업의 기구학적 특이점과, i 번째 작업과 높은 우선 순위를 가지는 $i - 1$ 번째까지의 작업 사이에 발생하는 알고리즘적 특이점들이다.

그러나, 이전 장에서 설명한 바와 같이 (54)의 미분 식을 구하는 것이 복잡하고 실시간 적용이 어려울 수 있기 때문에 보다 간단하지만 합리적인 귀납적 특이점 지수를 다음과 같이 정의한다.

$$m_i(\mathbf{q}) \triangleq \sqrt{\det \left\{ \mathbf{M}_i \mathbf{M}_i^T \right\}} \quad (55)$$

여기서 \mathbf{M}_i 는 다음과 같이 정의된다.

$$\mathbf{M}_i \triangleq \mathbf{J}_i \hat{\mathbf{Z}}_{i-1}^T \quad (56)$$

그리고, 첫 번째 우선 순위 작업에 대해서 당연하게 특이점 지수($m_1(\mathbf{q}) \triangleq \sqrt{\det \left\{ \mathbf{J}_1 \mathbf{J}_1^T \right\}}$)는 조작성 지수와 동일하다.

이제 일반적인 작업 우선 순위에 근거한 일반적이고 귀납적인 TR-method 식을 다음과 같이 얻을 수 있다.

$$\begin{cases} \delta \mathbf{q}_i = \delta \mathbf{q}_{i-1} + \hat{\mathbf{J}}_i^+ \delta \hat{\mathbf{r}}_{ip} \\ \delta \hat{\mathbf{r}}_i = \delta \mathbf{r}_i - \mathbf{J}_i \delta \mathbf{q}_{i-1} \\ \mathbf{P}_i = \mathbf{P}_{i-1} - \hat{\mathbf{J}}_i^+ \hat{\mathbf{J}}_i \\ \hat{\mathbf{J}}_i = \mathbf{J}_i \mathbf{P}_{i-1} \end{cases}, \quad \begin{cases} \delta \mathbf{q}_0 = \mathbf{0} \\ \mathbf{J}_0 = \mathbf{0} \\ \mathbf{P}_0 = \mathbf{I}_n \end{cases} \quad (57)$$

그리고, 여기에 추가적으로 다음과 같은 TR-process가 필요하다.

$$\begin{cases} \delta \hat{\mathbf{r}}_{ip} = \text{TR-Process}(\hat{\mathbf{J}}_i, m_i, \delta \hat{\mathbf{r}}_i) \\ m_i(\mathbf{q}) = \sqrt{\det \left\{ \mathbf{M}_i \mathbf{M}_i^T \right\}} \\ \mathbf{M}_i = \mathbf{J}_i \hat{\mathbf{Z}}_{i-1}^T \end{cases}, \quad \left\{ \hat{\mathbf{Z}}_0 = \mathbf{I}_n \right. \quad (58)$$

이제까지 설명한 TR-method의 효율성과 우수성은 다음 장에서 실험을 통해서 검증한다.

VI. 실험

실험을 수행한 매니플레이터는 7자유도 공간 구동 여유자유도 로봇으로 하와이대학교(Univ. of Hawaii)의 MARIS7080이라는 수중 매니플레이터이다(그림 6). 이 매니플레이터는 SAUVIM(Semi-Autonomous Underwater Vehicle for Intervention Mission, Univ. of Hawaii)에 부착되어 수중 작업을 수행하게 된다. 표 1은 MARIS7080의 D-H 값들을 보여준다. 이 매니플레이터는 심해에서의 자율 구동과 원격 작업을 위해 설계되었기 때문에 특이점 회피가 매우 중요한 문제이다.

TR-method와 이전에 제안된 방법들의 성능을 비교검증하기 위해서 알고리즘적 오차(algorithmic error)라는 개념을 도입했다. 그것은 피할 수 없는 오차(inevitable error)에서 실제 오차를 뺀 것으로 나타내며, 피할 수 없는 오차는 작업 공간 제약, 관절 제한 등과 같이 매니플레이터의 물리적인 특징으로 설계 단계에서만 피할 수 있는 오차이며 알고리즘으로는 극복이 불가능하다(그림 7). 따라서 알고리즘적 오차는 알고리즘에 따른 절대적인 오차이므로, 이것을 이용해서 제안된 방법의 성능을 다른 방법들과 쉽게 비교해 볼 수 있다.

그러나, 알고리즘적 오차를 일반적인 공간상의 운동에 대해서 정의하기가 어렵기 때문에, 본 논문에서는 x-z 평면 상에서의 작업 경로를 먼저 고려했다. 이 경우 MARIS7080의 관절 중에서 3, 4, 6번의 관절만 동작하게 된다. 그런 다음, 일반적인 공간상의 운동에 대한 성능을 검증하기 위해서 모든 관절이 구동하게 되는 x-y 평면 상에서의 경로에 대해 실험을 수행하였다. 또 각각의 경우에 대해서 다음과 같은 세 가지 항목에 대한 실험을 수행하였다.

- Case I: 기구학적 특이점이 고려된 단일 작업
- Case II: 알고리즘적 특이점이 고려된 두 개의 작업
- Case III: 기구학적 특이점과 알고리즘적 특이점이 모두 고려된 두 개의 작업

1. 평면 상에서의 실험

1.1 Case I

주어진 작업은 0.3m 반지름의 원형 경로를 추종하는 것이다.

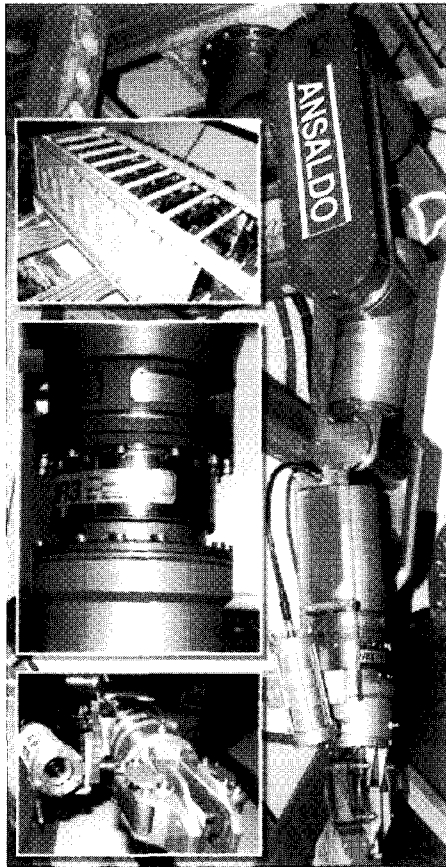


그림 6. MARIS 7080 수중 매니퓰레이터.
Fig. 6. MARIS 7080 underwater manipulator.

표 1. MARIS 7080의 D-H 값.

Table 1. D-H parameter for MARIS 7080.

Joint i	θ_i (rad)	α_i (rad)	a_i (m)	d_i (m)
1	$\pi/2$ (θ_1)	$-\pi/2$	0	-0.19
2	$-\pi/2$ (θ_2)	$-\pi/2$	0	-0.2235
3	π (θ_3)	π	0.4	0
4	$\pi/2$ (θ_4)	$-\pi/2$	0	0
5	0 (θ_5)	$\pi/2$	0	-0.4
6	0 (θ_6)	$\pi/2$	0	0
7	0 (θ_7)	0	0	0.414

주어진 경로 중에서 일부분은 작업 공간 밖에 위치하기 때문에, 이에 의해서 기구학적 특이점이 발생하게 된다. 실험 결과는 (6), (7)의 DLS(damped least-squares) 방법과 비교하였다. 제어 입력 주파수는 50Hz, $\bar{m} = 0.05$, $\sigma = 0.05$, $K_r = 0.5$ 로 각각 사용하였다. DLS 방법에 대해서는 $\epsilon = 0.2$, $\lambda_{\max} = 0.1$ 을 사용하였다. 그리고, 두 방법 모두에 대해서 작업 오차 비례 이득은 $\mathbf{K}_1 = [20 \ 20]^T$ 로 동일하게 사용하였다. 위의 값들은 안정성을 만족하면서 최고의 작업 성능을 위해서 조절되었다.

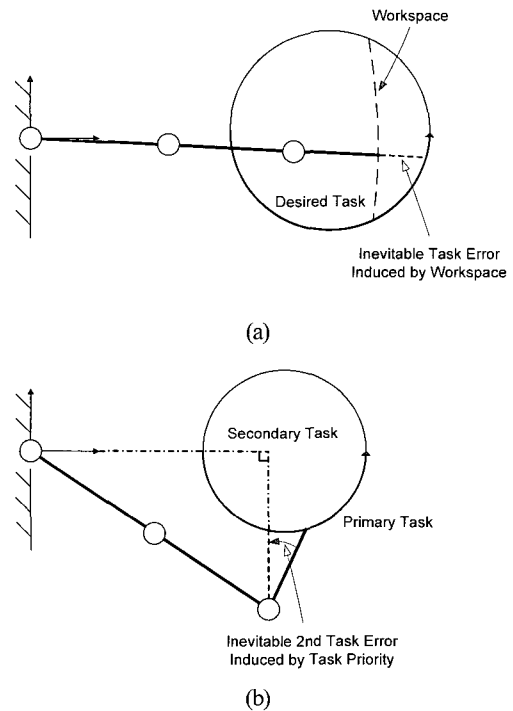


그림 7. 3자유도 매니퓰레이터에서의 피할 수 없는 오차의 예제: (a) 작업 공간 제한에 의해 발생하는 피할 수 없는 오차; (b) 작업 우선도에 의해 발생하는 두 번째 작업의 피할 수 없는 오차.

Fig. 7. Example of inevitable errors in planar 3-DOF manipulator: (a) inevitable task error induced by workspace; and, (b) inevitable 2nd task error induced by task priority.

이 실험에 대한 결과는 그림 8에 나타내었다. 8(a)와 8(c)는 TR-method를 이용한 실험의 작업 경로와 특이점 지수를 나타낸다. 8(b)와 8(d)는 DLS를 이용한 결과이다. 안정성에 대해서 TR-method의 특이점 지수는 위에서 정한 임계 값인 0.05 이상을 항상 만족한다는 것을 알 수 있다(그림 8(c)). 그러나, DLS 방법에서는 그림 8(d)에서 보는 바와 같이 특이점 지수가 0에 가까워진다. 따라서 수치 오차에 의해 높은 이득 값이나 큰 작업 오차가 시스템을 발산시킬 가능성이 많다.

그림 8(e)와 8(f)는 작업 오차의 평균 값과 알고리즘적 오차를 비교한 결과이다. DLS 방법의 경우 특이 영역 전후에 큰 overshoot이 발생하는 것을 볼 수 있다. 그림 8(f)에서 알 수 있듯이 알고리즘적 오차의 관점에서 TR-method의 방법이 DLS의 방법보다 3배 정도 좋은 결과를 보였다. 8(e)에서 35초에서 40초 사이, 그리고 60초 부근에서 발생하는 작업 오차는 이전 장에서 설명한 바와 같이 국부적 관절 재구성의 특성 때문에 발생한다. 또한, DLS 방법의 결과에서 III.1에서 설명한 계단 외란 입력 반응이 나타나는 것도 알 수 있다. 그것이 바로 높은 이득에 의한 영향이다. 실제로, DLS 방법을 이용한 실험에서 작업 성능을 보다 높이기 위해 ϵ 값을 내리거나, 혹은 작업 이득을 올렸을 경우 시스템이 발산하는 결과를 얻었다.

1.2 Case II

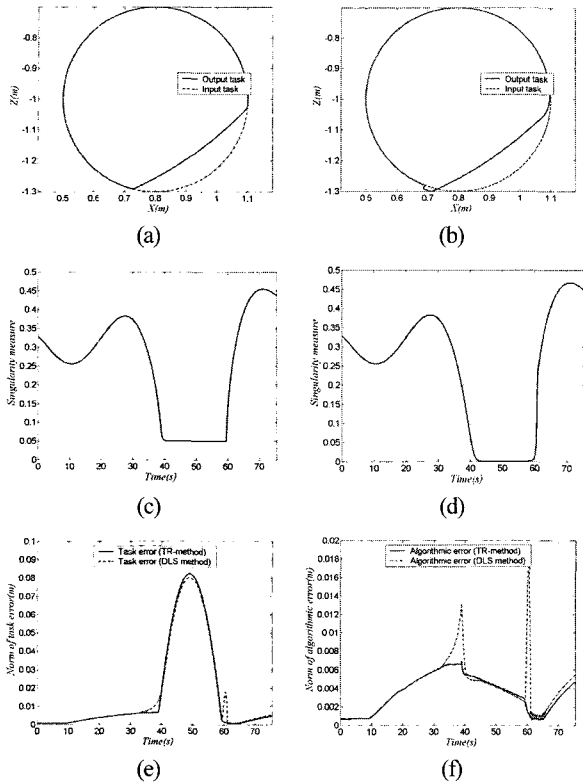


그림 8. 실험 결과(Case I): (a) 작업 경로(TR-method); (b) 작업 경로(DLS method); (c) 특이점 지수(TR-method); (d) 특이점 지수(DLS method); (e) 작업 오차의 평균값; (f) 알고리즘적 오차의 평균값.

Fig. 8. Experimental results (Case I): (a) task in the x-z plane(TR-method); (b) task in the x-z plane(DLS method); (c) singularity measure (TR-method); (d) singularity measure(DLS method); (e) norm of task errors; and, (f) norm of algorithmic errors.

주어진 작업은 두 가지로써 첫 번째 우선 순위 작업은 0.2m 반지름의 원을 추종하는 것이고, 두 번째는 초기 Euler RPY 각도 ($\pi, 0, -\pi/2$) 을 유지하는 것이다. 이 경우 기구학적 특이점은 없지만, 일부 영역에서 알고리즘적 특이점은 발생한다. 결과는 Chiaverini가 제안한 (19)와 같은 알고리즘적 특이점 회피 알고리즘(Algorithmic Singularity Avoidance Method, ASAM)과 비교하였다[16].

Case I과 동일한 제어 입력 주파수를 사용하였고, $\bar{m}_2 = 0.075$, $\sigma_2 = 0.075$, $K_{r,2} = 0.5$ 로 조절하였다. ASAM에 대해서는 조절 변수가 없다. 첫 번째 작업에 대한 변수들은 Case I과 동일하게 선택하였고, 두 번째 작업의 작업 이득은 두 경우 모두 $K_2=10$ 을 사용하였다.

그림 9는 Case II의 결과들을 나타낸다. 그림 9(a)와 9(c)는 TR-method의 작업 경로와 두 번째 작업의 특이점 지수를 나타내고, 그림 9(b)와 9(d)는 ASAM에 대한 결과들이다. 첫 번째 작업은 두 경우 모두 잘 수행되었다는 것을 알 수 있다. 그림 9(c)에는 기대와 같이 0.075 이상으로 특이점 지수가 유지되고 있다.

그림 9(e)와 9(f)는 두 번째 작업의 평균 오차와 두 번째 작

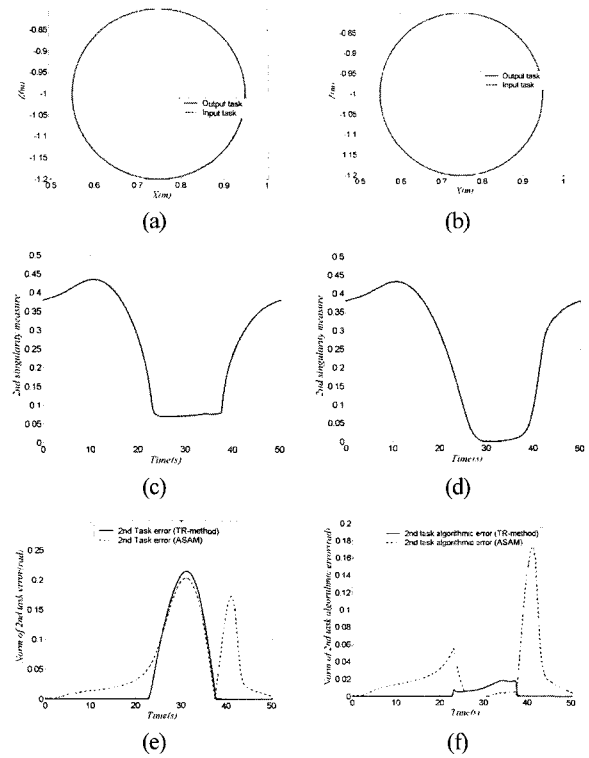


그림 9. 실험 결과(Case II): (a) 작업 경로(TR-method); (b) 작업 경로(ASAM); (c) 두 번째 작업의 특이점 지수(TR-method); (d) 두 번째 작업의 특이점 지수(ASAM); (e) 두 번째 작업 오차의 평균값; (f) 두 번째 작업의 알고리즘적 오차의 평균값.

Fig. 9. Experimental results (Case II): (a) task in the x-z plane (TR-method); (b) task in the x-z plane (ASAM); (c) 2nd task singularity measure (TR-method); (d) 2nd task singularity measure (ASAM); (e) norm of the 2nd task errors; and, (f) norm of the 2nd task algorithmic errors.

업의 알고리즘적 오차에 대한 비교 결과를 나타낸다. 두 번째 작업의 알고리즘적 오차에 근거해서 그림 9(f)로부터 TR-method의 결과가 ASAM에 비해 10배 가량 좋아졌음을 알 수 있다. 비록 ASAM은 알고리즘적 오차를 포함하고 있지만, 실험 결과에서 알 수 있듯이, 작업 오차는 전반적으로 매우 크다는 것을 알 수 있다. 반면에 TR-method에서는 알고리즘적 오차를 회피할 수 있을 뿐만 아니라 작업 성능도 대폭 향상시킬 수 있다.

1.3 Case III

주어진 작업은 Case II와 첫 번째 작업에서 0.3m 반지름을 추종한다는 차이만 있다. 따라서, 일부 경로는 작업 영역 이외에 존재하고, 일부는 동시에 수행될 수 없기 때문에 기구학적 특이점과 알고리즘적 특이점이 모두 존재한다. 앞에서와 마찬가지로 결과들은 DLS 방법과 ASAM의 결과와 비교하였다[16].

제어 입력 주파수는 Case I, II와 동일하고, $\bar{m}_1 = 0.075$, $\sigma_1 = 0.075$ 이다. 나머지 제어 변수들은 Case I, II에서 사용되었던 값들을 똑같이 사용하였다.

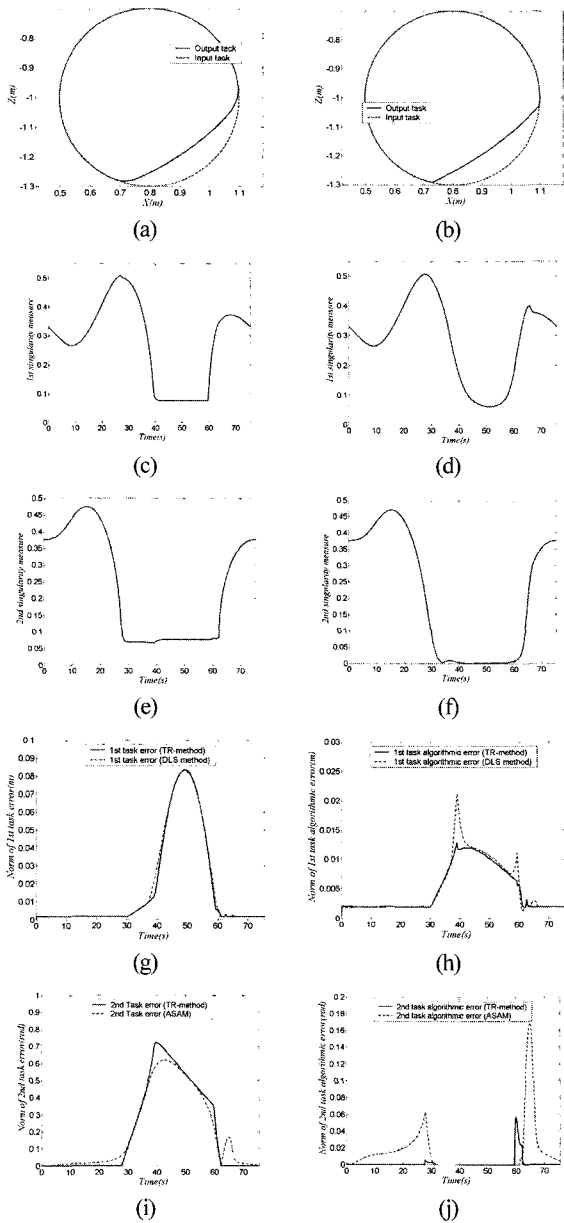


그림 10. 실험 결과(Case III): (a) 작업 경로(TR-method); (b) 작업 경로(DLS method); (c) 첫 번째 작업의 특이점 지수(TR-method); (d) 첫 번째 작업의 특이점 지수(DLS method); (e) 두 번째 작업의 특이점 지수(TR-method); (f) 두 번째 작업의 특이점 지수(ASAM); (g) 첫 번째 작업 오차의 평균값; (h) 첫 번째 작업의 알고리즘적 오차의 평균값; (i) 두 번째 작업 오차의 평균값; (j) 두 번째 작업의 알고리즘적 오차의 평균값.

Fig. 10. Experimental results (Case III): (a) task in the x-z plane(TR-method); (b) task in the x-z plane (DLS method); (c) 1st task singularity measure (TR-method); (d) 1st task singularity measure (DLS method); (e) 2nd task singularity measure (TR-method); (f) 2nd task singularity measure (ASAM); (g) norm of the 1st task errors; (h) norm of the 1st task algorithmic errors;(i) norm of the 2nd task errors; and, (j) norm of the 2nd task algorithmic errors.

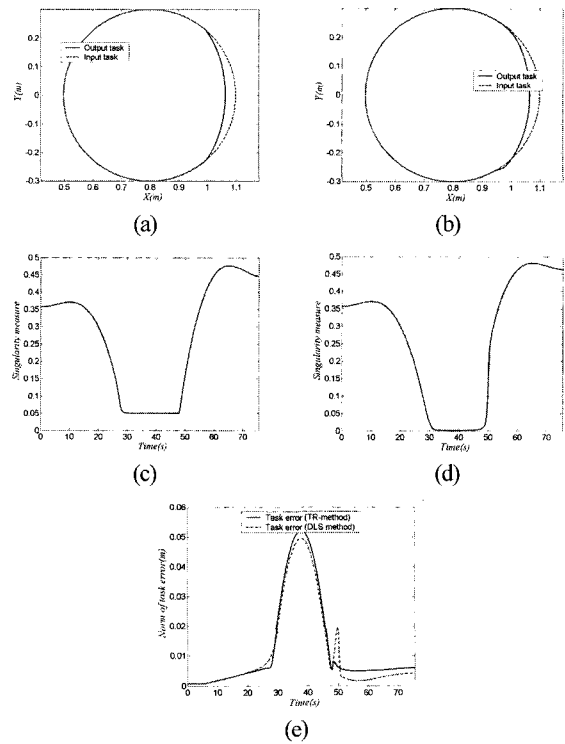


그림 11. 실험 결과(Case I): (a) 작업 경로(TR-method); (b) 작업 경로(DLS method); (c) 특이점 지수(TR-method); (d) 특이점 지수(DLS method); (e) 작업 오차의 평균값.

Fig. 11. Experimental results (Case I): (a) task in the x-y plane (TR-method); (b) task in the x-y plane (DLS method); (c) singularity measure (TR-method); (d) singularity measure (DLS method); and, (e) norm of task errors.

Case III의 결과들은 그림 10에 나타내었다. 그림 10(a), 10(c), 10(e)는 TR-method의 작업 경로, 첫 번째 특이점 지수, 두 번째 특이점 지수를 나타내고, 10(b), 10(d), 10(f)는 DLS 방법과 ASAM을 이용한 결과이다.

결과들은 Case I, II와 거의 유사하다는 것을 볼 수 있다. 이 결과를 통해서 TR-method가 여러 작업에 대해서도 잘 수행된다는 것을 검증할 수 있다. 그림 10(g), 10(h), 10(i), 10(j) 등은 비교 실험 결과이다. 그림 10(i)에서 두 번째 작업 오차에 대해서 ASAM을 이용한 실험의 결과가 TR-method에 비해 40초에서 60초 사이에서 보다 좋은 결과가 얻어졌음을 알 수 있다. 그러나, 그 영역은 이미 기구학적 특이 영역에 포함되어 있기 때문에 작업 우선 순위의 관점에서 두 번째 작업의 오차는 의미가 없다. 따라서, 기구학적 특이 영역에서는 두 번째 작업의 알고리즘적 오차를 0으로 설정하였다. 그리고, TR-method에서는 그 영역에서 두 번째 작업 오차에 우선해서 첫 번째 작업을 재구성하고자 노력한다.

위의 세 가지 경우의 실험 결과에서 주목할 점은 DLS 방법과 ASAM을 이용하였을 경우 발생하는 특이 영역 이후의 급격한 변화이다. TR-method에서는 국부적 작업 재구성이라는 특성 때문에 그런 현상이 발생하지 않고, 따라서 실제 시스템에서 작업 이득을 높임으로써 안정성을 해치지 않으면

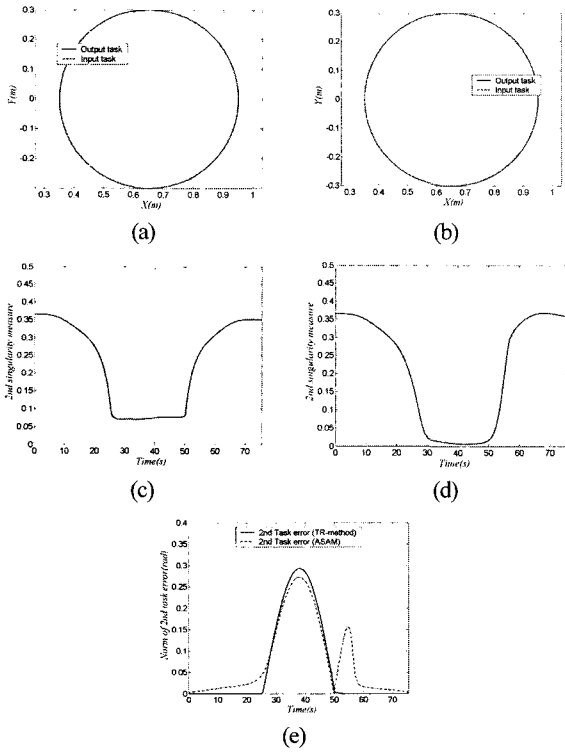


그림 12. 실험 결과(Case II): (a) 작업 경로(TR-method); (b) 작업 경로(ASAM); (c) 두 번째 작업의 특이점 지수(TR-method); (d) 두 번째 작업의 특이점 지수(ASAM); (e) 두 번째 작업 오차의 평균값.

Fig. 12. Experimental results (Case II): (a) task in the x-z plane (TR-method); (b) task in the x-z plane (ASAM); (c) 2nd task singularity measure (TR-method); (d) 2nd task singularity measure (ASAM); and, (e) norm of the 2nd task errors.

서 동시에 작업 성능 향상을 꾀할 수 있다.

2. 공간 상에서의 실험

일반적인 공간 운동에 대해서 성능을 검증하기 위해서 모든 관절이 구동하도록 x-y 원형 경로를 이용해 실험을 수행하였다. 실험은 앞 절과 마찬가지로 세 가지 경우에 대해서 수행하였다. 모든 제어 변수들은 앞 절과 동일하게 설정하였고, 결과 역시 비슷한 경향을 보이고 있다. 결과들은 각각 그림 11, 12, 13에 나타내었다.

VI. 결론

본 논문에서는 기구학적 특이점과 알고리즘적 특이점을 실시간으로 회피할 수 있는 방법을 제안하였다. 작업 재구성 기법(Task Reconstruction method, TR-method)은 실시간으로 특이점으로부터 자유로운 경로를 재구성해줄 뿐만 아니라, 작업 성능도 최대화한다. 이것은 제안된 방법이 국부적 관절 재구성이 아닌 국부적 작업 재구성이라는 내재적인 특성을 지니고 있기 때문이다. 또한, 동적 작업 우선 순위의 관점에서 TR-method를 해석함으로써 안정성과 작업 성능 효율을 모두 만족시키기 위해 작업 우선 순위가 동적으로 할당되는 것을 보였다.

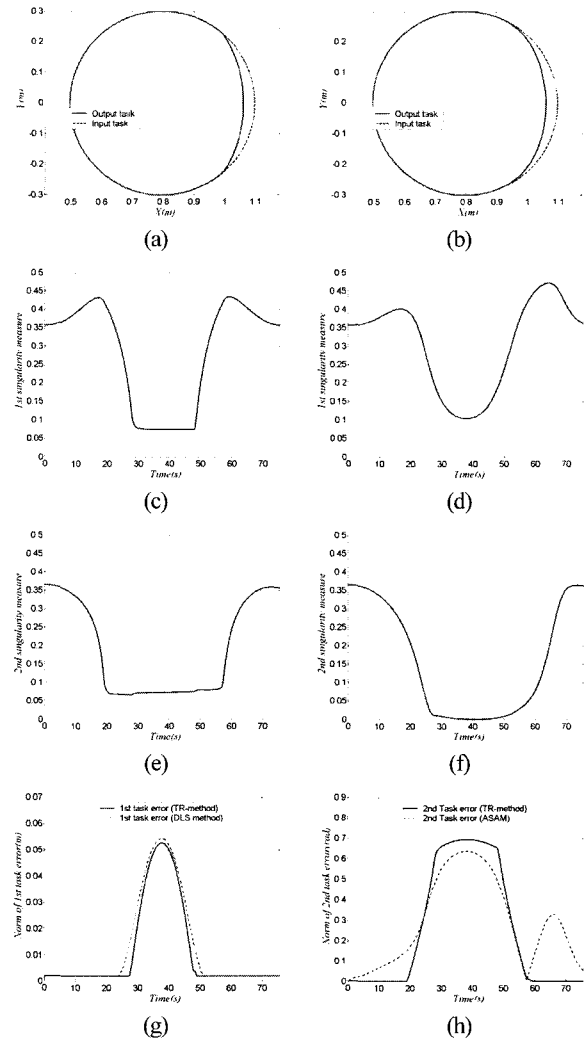


그림 13. 실험 결과(Case III): (a) 작업 경로(TR-method); (b) 작업 경로(DLS method); (c) 첫 번째 작업의 특이점 지수(TR-method); (d) 첫 번째 작업의 특이점 지수(DLS method); (e) 두 번째 작업의 특이점 지수(TR-method); (f) 두 번째 작업의 특이점 지수(ASAM); (g) 첫 번째 작업 오차의 평균값; (h) 두 번째 작업 오차의 평균값.

Fig. 13. Experimental results (Case III): (a) task in the x-y plane (TR-method); (b) task in the x-y plane (DLS method); (c) 1st task singularity measure (TR-method); (d) 1st task singularity measure (DLS method); (e) 2nd task singularity measure (TR-method); (f) 2nd task singularity measure (ASAM); (g) norm of the 1st task errors; and, (h) norm of the 2nd task errors.

기존의 DLS 방법과 Chiaverini의 알고리즘 특이점 회피 방법과 비교하여 TR-method는 다음과 같은 4가지 장점을 보여 준다. 1) 특이 영역 이후에 계단 외란 입력과 같은 반응을 보이지 않기 때문에 실제 실험에서 큰 작업 이득 값을 사용하여 작업 성능을 높일 수 있다. 2) 동일한 개념으로 특이점 지수의 교체만으로 기구학적 특이점과 알고리즘적 특이점 모두의 경우에 사용이 가능하다. 3) 성능 조절 변수가 물리적으로 명확한 의미를 지니기 때문에 성능 조율이 쉽다. 4) TR-

method는 다중 작업으로 귀납적으로 확장이 가능하며, 다중 작업 시에 발생하는 모든 기구학적, 알고리즘적 특이점을 회피할 수 있다.

TR-method는 7자유도 수중 매니퓰레이터를 이용하여 성능 검증 실험을 수행하였고, 알고리즘적 오차의 관점에서 기존의 DLS 방법과 ASAM에 비해서 기구학적 특이점의 경우에 3배의 성능 향상을, 알고리즘적 특이점에 대해서 10배의 성능 향상을 보여주었다. 또한 Case III의 결과로부터 다중 작업으로의 확장도 가능성을 입증하였다.

참고문헌

- [1] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 1985, pp. 722–728.
- [2] J. Baillieul, "Avoiding obstacles and resolving kinematic redundancy," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 1986, pp. 1698–1704.
- [3] C. L. Lück and S. Lee, "The semi-singularity and a topology-based global path planning approach for redundant manipulators," in *Proc., IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1994, pp. 510–517.
- [4] K.-C. Park, P.-H. Chang, and S. Lee, "A new kind of singularity in redundant manipulation: semi algorithmic singularity," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 1979–1984.
- [5] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, McGraw-Hill, 1996.
- [6] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. of Robotic Research*, vol. 4, no. 3, pp. 109–117, 1985.
- [7] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *J. of Dynamic Sys., Meas., and Control*, vol. 108, pp. 163–171, Sept. 1986.
- [8] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-16, no. 1, pp. 93–101, Jan. 1986.
- [9] A. A. Maciejewski and C. A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *Journal of Robotic Systems*, vol. 5, no. 6, pp. 527–552, 1988.
- [10] H. Seraji and R. Colbaugh, "Singularity-robustness and task-prioritization in configuration control of redundant robots," in *Proc., IEEE Conf. on Decision and Control*, 1990, pp. 3089–3095.
- [11] O. Egeland, J. R. Sagli, and I. Spangelo, "A damped least-squares solution to redundancy resolution," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 1991, pp. 945–950.
- [12] J. Kieffer, "Differential analysis of bifurcations and isolated singularities for robots and mechanisms," *IEEE Trans. on Robotics and Automation*, vol. 10, no. 1, pp. 1–10, Feb. 1994.
- [13] M. Kirćanski, "Symbolic singular value decomposition for simple redundant manipulators and its application to robot control," *Int. J. of Robotic Research*, vol. 14, no. 4, pp. 382–398, Aug. 1995.
- [14] D. N. Nenchev and M. Uchiyama, "Singularity consistent path tracking: a null space based approach," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 1995, pp. 2482–2489.
- [15] F. Caccavale, S. Chiaverini, and B. Sciliano, "Second-order kinematic control of robot manipulators with Jacobian damped least-squares inverse: Theory and experiments," *IEEE/ASME Trans. on Mechatronics*, vol. 2, no. 3, pp. 188–194, Sept. 1997.
- [16] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, June 1997.
- [17] J. E. Lloyd, "Desingularization of nonredundant serial manipulator trajectories using Puiseux series," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 4, pp. 590–600, Aug. 1998.
- [18] K. Tchoń and R. Muszyński, "Singular inverse kinematic problem for robotic manipulators: a normal form approach," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 1, pp. 93–104, Feb. 1998.
- [19] J. Foret, M. Xie, and J. G. Fontaine, "Bordered matrix for singularity robust inverse kinematics: A methodological aspect," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 3013–3019.
- [20] J. Park, Y. Choi, W. K. Chung, and Y. Youm, "Multiple tasks kinematics using weighted pseudo-inverse for kinematically redundant manipulators," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 4041–4047.
- [21] I. Dulceba and J. Z. Sasiadek, "Modified Jacobian method of transversal passing through the smallest deficiency singularities for robot manipulators," *Robotica*, vol. 20, pp. 405–415, 2002.
- [22] G. Marani, J. Kim, J. Yuh, and W. K. Chung, "A real-time singularity avoidance in resolved motion rate control of robotic manipulators," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 1973–1978.
- [23] D. N. Nenchev, "Tracking manipulator trajectories with ordinary singularities: A null space-based approach," *Int. J. of Robotic Research*, vol. 14, no. 4, pp. 399–404, Aug. 1995.
- [24] D. N. Nenchev, Y. Tsumaki, and M. Uchiyama, "Singularity-consistent parameterization of robot motion and control," *Int. J. of Robotic Research*, vol. 19, no. 2, pp. 159–182, Feb. 2000.
- [25] S. K. Singh, "Motion planning and control of non-redundant manipulators at singularities," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 1993, pp. 487–492.
- [26] C. Chevallereau and B. Daya, "A new method for robot control in singular configurations with motion in any Cartesian direction," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 1994, pp. 2692–2697.

- [27] C. Chevallereau, "Feasible trajectories for a non-redundant robot at a singularity," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 1996, pp. 1871–1876.
- [28] J. E. Lloyd and V. Hayward, "Singularity-robust trajectory generation," *Int. J. of Robotic Research*, vol. 20, no. 1, pp. 38–56, Jan. 2001.
- [29] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. on Man-Machine System*, vol. MMS-10, no. 2, pp. 47–53, June 1969.
- [30] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison Wesley, 1991.
- [31] B. Siciliano and J.-J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proc., Int. Conf. on Advanced Robotics*, 1991, pp. 1211–1216.
- [32] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in *Proc., IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1998, pp. 323–329.
- [33] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. of Robotic Research*, vol. 4, no. 2, pp. 3–9, 1985.
- [34] K. A. O'Neil, Y.-C. Chen, and J. Seng, "On the existence and characteristics of solution paths at algorithmic singularities," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 2, pp. 336–342, Apr. 1998.
- [35] Y.-C. Chen and I. D. Walker, "A consistent null-space based approach to inverse kinematics of redundant robots," in *Proc., IEEE Int. Conf. on Robotics and Automation*, 1993, pp. 374–381.



김진현

1975년 5월 13일생. 1998년 포항공대 기계공학(학사). 2000년 포항공대 기계공학(석사). 2000년~현재 포항공대 기계공학 박사과정 재학중. 관심분야는 기구학, 로봇 동역학 해석 및 제어, 수중 로봇, 여유자유도 로봇 등.



정완균

1959년 2월 24일생. 1981년 서울대학교 기계설계학(학사). 1983년 한국과학기술원 기계공학(석사). 1987년 한국과학기술원 생산공학(박사). 1987년~현재 포항공과대학교 기계공학과 교수. 관심분야는 여유자유도 로봇의 제어, 이동 로봇 및 수중 로봇의 설계 및 제어, 기계시스템의 강인 제어 등.



최영진

1970년 5월 12일생. 1994년 한양대 정밀기계공학(학사). 1996년 포항공대 기계공학(석사). 2002년 포항공대 기계공학(박사). 현재 한국과학기술연구원(KIST) 지능로봇연구센터 연구원. 관심분야는 Hu-manoid Robot Walking, 선형/비선형

강인 제어, 자동조정 PID제어, 디스크 구동 시스템의 제어.