ROBOT ALPHA → OMEGA

로봇의 알파에서 오메가까지

기계는 어떻게 몸을 얻고, 생각하고, 일하기 시작했는가

ALPHA	광물·소재
BODY	몸·감각·운동
BRAIN	AI·VLA
DATA	경험·시뮬레이션
FACTORY	공급망·대량생산
OMEGA	성공한 행동의 가격

BOOK MASTER · FILE LOCK · v1.0

유아독존 · ChatGPT

로봇의 알파에서 오메가까지

기계는 어떻게 몸을 얻고, 생각하고, 일하기 시작했는가

유아독존 · ChatGPT

BOOK MASTER v1.0 · FILE LOCK · 2026-09-24

판권·판본 상태

판본	BOOK MASTER v1.0 · FILE LOCK
기준일	2026-09-24
저자	유아독존 · ChatGPT
원고 규모	공백 기준 32,613어절 · 편집 표식 제외 약 104,890자
예상 읽기	약 4-7시간 · 기술 교양 독서 속도 기준
SOURCE FREEZE	PASS
사실검증	P0·P1 COMPLETE WITH HOLD ITEMS
기술 정확성	GATE COMPLETE
도판	자체 제작 벡터 19점 · 19/19
공개 상태	정본 잠금 완료 · 외부 공개 전 OWNER GATE

현재 사실 · 기업 발표 · 미래 목표 · 분석가 추정 · 저자 해석을 구분해 표기한다.

확인되지 않은 비교 수치와 산업 평균은 범위화하거나 HOLD 상태로 남겼다.

이 파일은 공개·출판사 제출·배포 전 소유자 검토용이다.

이 책의 중심 질문

AI가 생각의 가격을 낮췄다면,
로봇은 어떻게 현실세계에서
‘행동의 가격’을 낮추는가?

광물에서 시작해 몸·지능·데이터·공장·공급망을 지나
성공한 행동 하나의 가격과 인간-기계 노동분업까지 추적한다.

서문

행동의 가격을 묻는 법

이 책은 휴머노이드 기업을 나열하는 안내서도, 부품 이름을 외우게 하는 교과서도, 미래를 단정하는 예언서도 아니다.

한 가지 인과사슬을 끝까지 따라간다. 광물과 소재가 모터·감속기·센서·배터리가 되고, 그것들이 액추에이터와 몸을 만들며, 감각·운동·AI·VLA·데이터·시뮬레이션을 거쳐 현실의 작업으로 이어지는 과정이다.

핵심은 로봇 한 대의 판매가격이 아니다. 품질·시간·안전 기준을 통과해 실제로 완료된 물리적 행동 하나에 얼마가 드는가다. 이 책은 그 단위를 '성공한 행동 하나의 가격'이라고 부른다.

그러므로 화려한 DEMO와 반복 가능한 상업운영을 구분하고, 직업 전체보다 그 안의 TASK를 보며, 하나의 거대한 전환점보다 작업마다 다른 여러 임계점을 추적한다.

독자가 마지막 장을 덮을 때 남아야 할 것은 인간처럼 생긴 기계의 이미지가 아니다. 행동의 가격을 낮추는 전체 시스템과, 그 시스템이 인간의 일을 다시 나누는 방식이다.

차례

BOOK MASTER v1.0 · FILE LOCK

서문 4

0장 · 인간은 왜 자기 몸을 기계로 만들려고 했는가 7

PART I · 몸을 만든다 14

제1장 – 로봇을 해부하다 15

제2장 – 모터 30

제3장 – 감속기 42

제4장 – 액추에이터 59

제5장 – 손 76

제6장 – 감각 93

제7장 – 걷기 109

제8장 – 배터리 123

제9장 – 열 143

PART II · 몸에 지능을 넣는다 160

제10장 – 로봇의 세 개의 뇌 161

제11장 – VLA 176

PART III · 경험을 만든다 192

제12장 – 로봇은 어디서 배우는가 193

제13장 – 가상세계에서 백만 번 넘어뜨려라 211

PART IV · 기술에서 산업으로 236

제14장 – 실험실 문을 열다 237

제15장 – 로봇 한 대의 가격표를 뜯어보자 256
제16장 – 보이지 않는 공급망 271
제17장 – 미국 289
제18장 – 중국 306
PART V · 제품이 되는 마지막 벽 324
제19장 – 고장 나지 않는 로봇 325
제20장 – 로봇 공장 343
PART VI · 행동의 경제학 366
제21장 – 로봇 한 시간은 얼마인가 367
제22장 – 로봇은 직업을 빼앗는가 387
에필로그 OMEGA · 행동의 가격이 무너지는 날 408
용어설명 434
후주·출처 및 참고자료 438
도판 목록 446
색인 후보 448

0장 – 인간은 왜 자기 몸을 기계로 만들려고 했는가

바퀴가 있는데 왜 다리를 만드는가

공학적으로만 생각하면 인간은 그다지 훌륭한 기계가 아니다.

두 발로 서 있기 위해 끊임없이 균형을 잡아야 한다. 넘어질 수 있고, 계단을 오르는 데 많은 에너지를 쓰며, 두 팔과 열 손가락은 제어해야 할 관절만 잔뜩 늘려 놓는다.

공장에서 물건을 옮기는 것이 목적이라면 컨베이어가 낫다. 창고를 이동한다면 바퀴가 낫다. 무거운 물건을 들어 올린다면 지게차가 낫다.

그런데 인간은 결국 **자기와 비슷하게 생긴 기계**를 만들기 시작했다.

왜일까.

그 답을 찾으려면 로봇에서 출발해서는 안 된다.

인간이 만들어 놓은 세계에서 출발해야 한다.

---

1. 세계가 이미 인간 모양이다

문손잡이는 사람 손이 잡을 수 있는 높이에 있다.

계단은 사람 다리 길이에 맞춰져 있다.

의자는 사람의 엉덩이 높이에 있고, 자동차 운전석에는 두 손으로 잡는 핸들과 두 발로 밟는 페달이 있다.

공장의 공구도, 주방의 칼도, 병원의 의료기구도 마찬가지다.

수천 년 동안 인간은 자기 몸에 맞추어 물리적 세계를 개조했다.

그 결과 현대 문명은 보이지 않는 하나의 규격을 갖게 되었다.

> **인간의 몸이다.**

따라서 완전히 새로운 자동화 기계를 만들려면 주변 환경도 기계에 맞게 바꿔야 한다.

그러나 인간형 로봇이 충분히 능숙해진다면 이야기가 달라진다.

공장을 로봇에 맞추는 대신,

로봇을 이미 존재하는 인간의 세계에 넣을 수 있다.

휴머노이드의 경제적 매력은 여기서 시작한다.

---

2. 최초의 로봇은 인간처럼 생기지 않았다

현대 로봇 산업의 출발점은 인간 복제가 아니었다.

1961년 미국 GM 공장에 투입된 Unimate는 거대한 기계팔이었다.

걷지도 않았다.

말하지도 않았다.

세상을 이해할 필요도 없었다.

정해진 장소에서 정해진 행동을 반복하면 됐다.

이것은 매우 중요한 로봇공학의 첫 번째 원리였다.

> **세상을 이해하는 로봇을 만드는 것보다 세상을 단순하게 만드는 것이 훨씬 쉽다.**

그래서 자동차 공장은 로봇에게 이상적인 세계가 되었다.

부품 위치를 고정하고,

컨베이어 속도를 일정하게 만들고,

사람을 위험구역 밖으로 내보내면,

로봇에게 '지능'이 거의 없어도 엄청난 생산성을 얻을 수 있었다.

---

3. 그런데 공장 밖으로 나오자 모든 것이 무너졌다

로봇에게 이런 명령을 내려보자.

> **"저 컵 좀 가져와."**

인간에게는 하찮은 일이다.

하지만 로봇에게는 거대한 문제다.

먼저 **'저 컵'**이 무엇인지 알아야 한다.

컵까지 걸어가야 한다.

책상을 피해야 한다.

팔을 뻗어야 한다.

컵 위치를 계산해야 한다.

손가락을 벌려야 한다.

컵을 깨뜨리지 않을 정도로만 힘을 주어야 한다.

컵이 미끄러지면 즉시 힘을 조절해야 한다.

그리고 사람에게 가져와 손에 넘겨야 한다.

한 문장의 명령 뒤에 수십 개의 물리적 문제가 숨어 있다.

---

4. 그래서 로봇공학은 두 갈래로 갈라졌다

한쪽은 **세상을 로봇에게 맞추는 방법**이었다.

공장 자동화가 대표적이다.

다른 한쪽은 훨씬 어려운 길이었다.

> **로봇을 세상에 맞춘다.**

이것이 이동로봇, 자율주행, 휴머노이드로 이어진다.

여기서부터 문제가 폭발한다.

바닥이 달라지고,

조명이 달라지고,

물건 위치가 달라지고,

사람이 갑자기 나타나고,

처음 보는 물체를 만나게 된다.

이를 로봇공학에서는 흔히 **비정형 환경** 문제로 다룬다.

---

5. 몸과 지능이 다시 만났다

고정된 공장에서는 환경을 단순하게 만들면 됐다. 그러나 인간의 공간으로 나온 로봇은 매번 달라지는 물체와 사람, 바닥과 지시를 해석해야 한다.

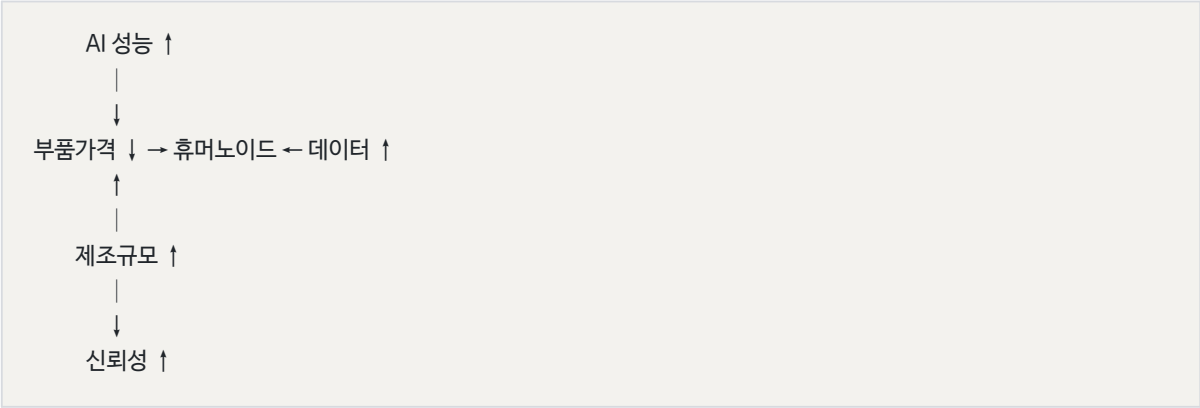
AI는 이 변동성을 다루는 비용을 낮추기 시작했다. 그래도 언어모델 하나를 몸에 넣는다고 로봇이 완성되지는 않는다. 느린 의미 판단, 중간 속도의 행동계획, 빠른 균형·힘 제어가 서로 다른 시간척도에서 연결되어야 한다.

그리고 몸은 현실의 대가를 요구한다. 배터리가 닳고, 열이 나고, 관절이 마모되며, 실패하면 사람이 개입한다. 따라서 경쟁의 단위는 AI 하나가 아니라 몸·지능·데이터·공장·공급망이 결합된 시스템이다.

---

6. 휴머노이드 경쟁의 진짜 방정식

여기까지 오면 하나의 식이 만들어진다.



어느 하나만 좋아져서는 충분하지 않다.

AI가 뛰어나도 몸이 너무 비싸면 실패한다.

몸이 싸도 일을 못 배우면 실패한다.

둘 다 좋아도 자주 고장 나면 실패한다.

모든 것이 좋아도 인간보다 비싸면 대량 도입되지 않는다.

---

7. 그래서 이 책은 '로봇 이야기'가 아니다

이제부터 우리는 로봇 한 대를 실제로 분해할 것이다.

껍데기를 벗긴다.

뼈를 본다.

관절을 연다.

모터를 꺼낸다.

감속기를 다시 분해한다.

배터리를 뜯는다.

손가락 끝의 센서를 본다.

그다음 뇌를 연다.

AI가 무엇을 보는지,

어떻게 걷는지,

어떻게 배우는지 살펴본다.

그리고 다시 조립한다.

마지막에는 그 로봇을 공장에 세워 놓고 묻는다.

> **"그래서 너를 고용하면 한 시간에 얼마인가?"**

이 질문에 답할 수 있을 때 비로소 휴머노이드 산업을 이해한 것이다.

0장 마지막 챗화

인간: "너는 왜 나처럼 생겼어?"

로봇: "당신을 닮고 싶어서가 아닙니다."

인간: "그럼?"

로봇: "당신이 세상을 당신처럼 만들어 놓았으니까요."

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

PART I

몸을 만든다

몸은 지능이 현실과 만나는 인터페이스다.

제1장 – 로봇을 해부하다

70kg의 기계 안에는 무엇이 들어 있는가

휴머노이드의 외피를 벗겨 보자.

겉으로 보면 머리 하나, 몸통 하나, 팔 두 개, 다리 두 개다. 인간과 크게 다르지 않아 보인다.

그러나 피부 아래로 들어가는 순간 완전히 다른 생명체가 나타난다.

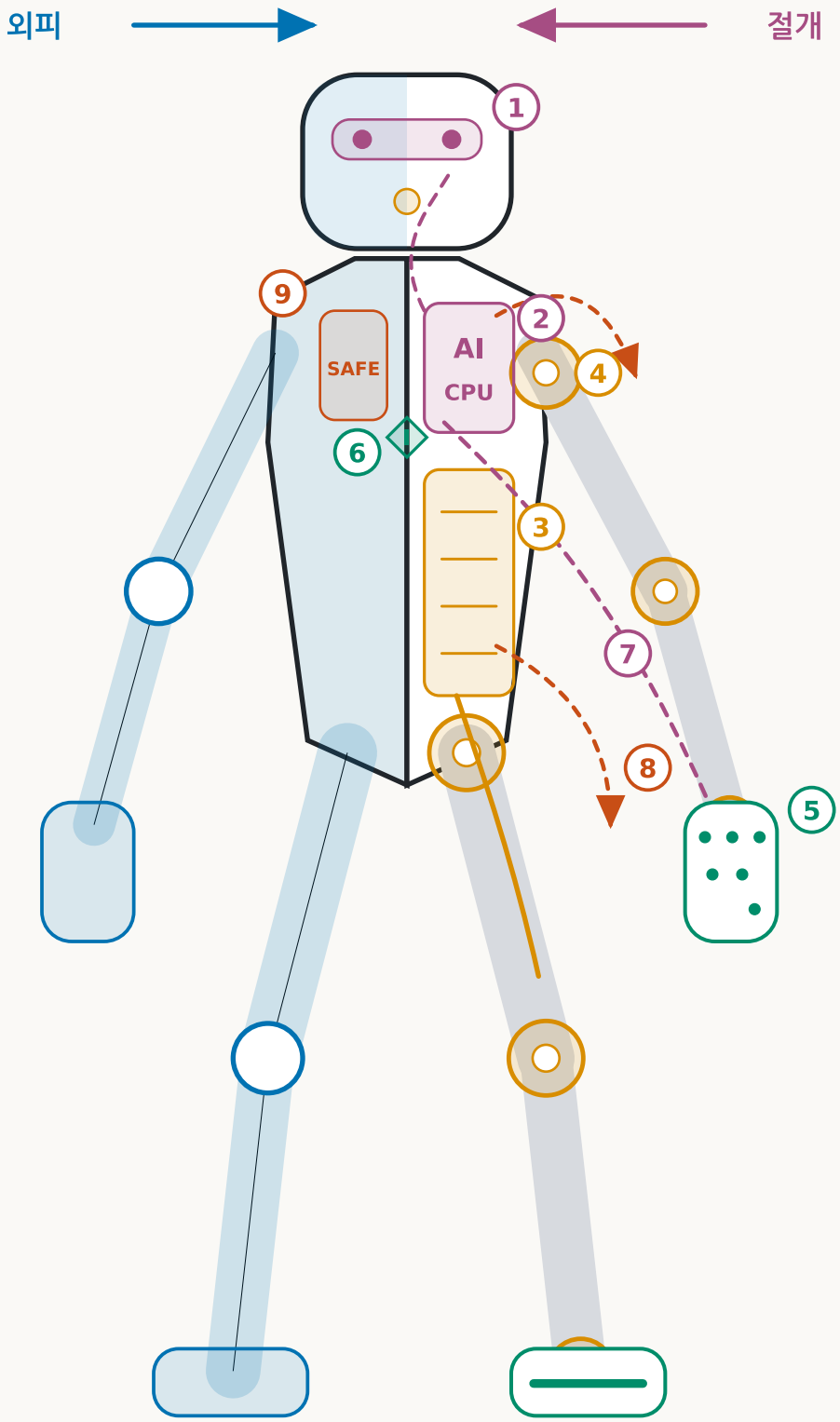
사람의 몸은 대략

뼈 → 관절 → 근육 → 신경 → 감각기관 → 뇌 → 혈액

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

V01. 휴머노이드 전체 해부도

외피 안에서 에너지·구동·감각·제어·열관리가 동시에 작동한다



외형은 하나지만 내부는 서로 다른 속도의 시스템 네트워크다.

개념도 · 축척 아님

한 몸 안의 9개 시스템

1 센서 헤드
카메라·깊이
마이크

2 주 컴퓨터
인지·계획
AI 가속

3 전원부
배터리
전력분배

4 관절 구동
모터·감속기
베어링·브레이크

5 손·촉각
구동계
접촉·미끄럼

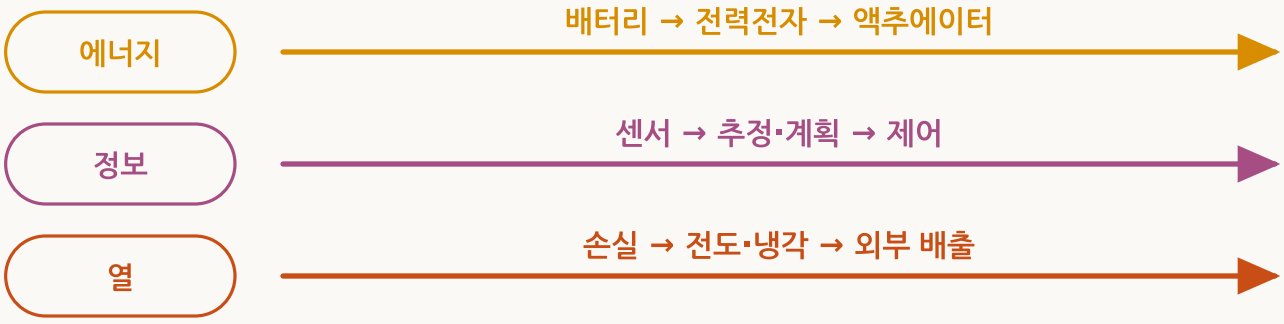
6 자세·힘 감지
IMU·엔코더
발·힘센서

7 통신·배선
전력·데이터
시간 동기화

8 냉각·열
전도·방열
출력 제한

9 안전제어
충돌 감지
정지·권한 제한

몸 안을 흐르는 세 경로



특정 회사의 실제 내부구조가 아닌 공통 시스템 개념도

으로 작동한다.

휴머노이드에도 놀랄 만큼 비슷한 대응관계가 있다.

인간	휴머노이드
뼈	프레임·구조체
관절	회전·직선 관절
근육	액추에이터
힘줄	감속기·전달기구
피부	촉각·접촉 센서
눈	카메라·깊이 센서
평형기관	IMU
신경	통신망·배선
혈액	전력
심장	배터리·전력계통
소뇌	실시간 운동제어
대뇌	AI·행동계획

하지만 이 비유에는 한계가 있다.

로봇에는 인간에게 없는 중요한 기관이 하나 더 있다.

감속기다.

이 작은 기계가 앞으로 이 책에서 계속 등장한다.

---

1. 먼저 몸 전체를 CT처럼 잘라보자

머리부터 내려간다.

머리

카메라 깊이센서 마이크 통신장치

↓

가슴

주 컴퓨터 AI 가속기 전력전자 배터리 냉각계통

↓

어깨·팔꿈치·손목

모터 감속기 엔코더 베어링 힘·토크 센서

↓

손

소형 액추에이터 힘센서 촉각센서 손가락 전달기구

↓

골반

고출력 액추에이터 구조 프레임 주요 배선

↓

무릎·발목

고토크 액추에이터 감속기 엔코더 힘센서

↓

발

접촉센서 힘센서

이 모든 것이 하나의 통신망으로 연결된다.

---

2. 그런데 가장 먼저 필요한 것은 AI가 아니다

로봇에게 명령한다.

> “팔을 30도 들어.”

AI가 아무리 똑똑해도 전기가 없으면 아무 일도 일어나지 않는다.

따라서 로봇 행동의 가장 밑바닥에는 이것이 있다.

> **에너지.**

배터리에서 전자가 움직인다.

↓

전력전자가 전류를 조절한다.

↓

모터가 자기장을 만든다.

↓

회전축이 돈다.

↓

감속기가 회전을 바꾼다.

↓

관절이 움직인다.

즉 로봇의 행동을 물리학적으로 거꾸로 추적하면

행동 ← 운동 ← 토크 ← 모터 ← 전류 ← 배터리

가 된다.

이 사슬이 끊어지는 순간 로봇은 아무리 뛰어난 AI를 가지고 있어도 움직이지 못한다.

---

3. 관절 하나를 꺼내 보자

여기가 휴머노이드 하드웨어의 핵심이다.

겉에서 보면 단순한 원통형 부품이다.

하지만 안에는 작은 공장이 들어 있다.

```
전원
 ↓
인버터
 ↓
모터
 ↓
감속기
 ↓
베어링
 ↓
출력축
 ↓
관절

+ 엔코더
+ 토크센서
+ 온도센서
+ 제어회로
```

이 전체 묶음이 **액추에이터**다.

따라서

모터와 액추에이터는 같지 않다.

이 구분을 반드시 기억해야 한다.

---

4. 모터는 왜 혼자 관절을 움직이지 않는가

모터는 회전하는 것을 좋아한다.

그리고 꽤 빠르게 회전한다.

그러나 사람의 무릎은 그렇게 움직이지 않는다.

무릎에서 필요한 것은

낮은 속도 + 큰 힘 + 정확한 위치가 필요하다.

그래서 중간에 감속기를 넣는다.

챗화

모터

“나 3,000rpm으로 돌 수 있어.”

무릎

“그 속도로 다리를 돌리면 사람이 날아가.”

모터

“그럼 어떡해?”

감속기

“나한테 맡겨.”

감속기는

속도를 버리고 토크를 얻는다.

예를 들어 개념적으로 100:1 감속이라면 모터가 100회전하는 동안 출력축은 약 1회전한다.

그 대신 훨씬 큰 토크를 얻는다.

물론 실제 시스템에서는 마찰·변형 등의 손실이 있기 때문에 에너지가 공짜로 생기는 것은 아니다.

---

5. 그래서 감속기가 그렇게 중요하다

정밀 액추에이터가 중요한 원가 항목이라는 방향은 타당하다. 그러나 제조사가 같은 범위와 기준으로 공개한 비교 가능한 BOM은 드물다.

따라서 하나의 비율을 모든 로봇에 적용하지 않는다. 생산량, 자유도, 손 구조, 감속기 종류, 센서 구성, 자체생산 여부와 회계 범위에 따라 값이 크게 달라지기 때문이다.

대신 질문을 바꾼다.

> **휴머노이드의 원가를 떨어뜨릴 때 가장 큰 레버가 어디인가?**

그 후보 가운데 액추에이터가 있는 것이다.

---

6. 그런데 모든 관절이 똑같은 힘을 필요로 하지 않는다

이것도 중요하다.

손가락과 무릎에 같은 모터를 넣는 것은 낭비다.

손가락

작고 가벼움 → 작은 힘 → 높은 정밀도

손목·팔꿈치

중간 토크 → 정밀한 조작

어깨

팔 전체를 지탱 → 더 큰 토크

고관절

몸통과 다리를 움직임 → 매우 큰 토크

무릎·발목

체중 지지 + 걷기 + 충격 → 큰 순간 토크

그래서 휴머노이드에는 여러 크기와 형태의 액추에이터가 들어간다.

그러나 종류를 무한정 늘리면 생산비가 올라간다.

여기서 중요한 제조 원리가 나온다.

> **관절의 종류를 줄여라.**

가능하면 몇 종류의 표준 액추에이터를 여러 관절에서 반복 사용한다.

자동차 산업에서 이미 익숙한 **부품 공용화**다.

휴머노이드가 연구장비에서 공산품으로 이동할수록 이 원리가 중요해진다.

---

7. 자유도 – 로봇 가격을 올리는 숨은 숫자

여기서 DOF, **자유도**라는 개념이 등장한다.

팔을 생각해보자.

어깨를 앞뒤로 움직인다.

좌우로 움직인다.

회전한다.

팔꿈치를 굽힌다.

손목을 돌린다.

각각 하나의 독립적인 움직임이다.

이것이 자유도다.

자유도가 늘어나면 인간 같은 움직임은 좋아진다.

하지만 동시에

액추에이터 ↑ 센서 ↑ 배선 ↑ 제어 복잡성 ↑ 무게 ↑ 전력소비 ↑ 고장 가능성 ↑ 가격 ↑

가 따라온다.

따라서 로봇 설계자는 항상 묻는다.

> **"인간과 똑같이 움직여야 하는가?"**

필요하지 않은 자유도 하나를 제거하는 것이 값비싼 부품 하나를 싸게 만드는 것보다 더 큰 원가절감일 수도 있다.

---

8. 이것이 '인간 복제'와 '산업용 휴머노이드'의 차이다

인간에게는 인간의 진화사가 만든 몸이 있다.

로봇에게는 그 모든 구조가 필요하지 않다.

예를 들어 사람의 팔꿈치는 특정 방향으로 제한된다.

하지만 로봇 관절은 필요하다면 인간에게 불가능한 방향으로 회전할 수도 있다.

Boston Dynamics의 전동 Atlas가 보여주는 움직임 중 일부가 좋은 사례다.

즉 미래 휴머노이드는

사람처럼 생겼지만 사람의 해부학을 그대로 복사하지 않을 가능성이 크다.

---

9. 로봇에게 뼈가 필요한 이유

프레임은 단순한 외피가 아니다.

힘이 지나가는 길이다.

로봇이 10kg 상자를 들면 그 힘은

손

↓

손목

↓

팔꿈치

↓

어깨

↓

몸통

↓

골반

↓

다리

↓

발

↓

바닥

으로 전달된다.

이것을 **하중 경로**(load path)라고 생각하면 쉽다.

따라서 프레임을 무조건 얇고 가볍게 만들 수 없다.

---

10. 로봇 설계의 잔인한 악순환

여기서 휴머노이드 공학 전체를 지배하는 첫 번째 순환고리가 나타난다.

```
튼튼하게 만든다
    ↓
무거워진다
    ↓
큰 모터가 필요하다
    ↓
큰 감속기가 필요하다
    ↓
전력 소비가 증가한다
    ↓
큰 배터리가 필요하다
    ↓
더 무거워진다
    ←
```

이것을 깨는 것이 휴머노이드 설계의 핵심이다.

그래서

고강도 경량소재,

고토크밀도 모터,

작은 감속기,

고에너지밀도 배터리,

효율적인 제어

가 모두 연결된다.

어느 부품 하나의 혁신이 다른 부품을 작게 만들 수 있다.

---

11. 전력과 정보가 몸 안을 흐른다

배터리는 몸통에서 전력을 보내고, 전력전자는 관절의 모터를 구동한다. 컴퓨터는 카메라·엔코더·IMU·힘센서의 신호를 받아 로봇이 어디에 있고 무엇과 접촉했는지 추정한다.

이 신경계는 한 속도로 움직이지 않는다. 의미를 해석하는 상위 AI보다 균형과 모터 전류를 다루는 제어계가 훨씬 빨리 반응해야 한다. 목표와 측정값의 차이를 계속 수정하는 페루프가 몸 전체를 묶는다.

배터리·센서·제어의 원리는 뒤 장에서 각각 연다. 여기서 중요한 것은 모든 부품이 하나의 하중·전력·정보 흐름을 이룬다는 점이다.

---

12. 휴머노이드의 진짜 본질

여기까지 뜯어보면 놀라운 사실을 발견한다.

휴머노이드는 하나의 기계가 아니다.

수십 개의 페루프 제어시스템이 서로 연결된 네트워크다.

그리고 그 위에 AI가 올라가 있다.

그래서 로봇을

> “AI + 모터”

라고 이해하면 너무 많은 것을 놓친다.

좀 더 정확하게 표현하면,

> **휴머노이드 = 에너지 시스템 + 기계 시스템 + 감각 시스템 + 실시간 제어 시스템 + 학습 시스템의 결합체**

라고 보는 편이 정확하다.

---

13. 그리고 아직 한 부품을 열지 않았다

지금까지 액추에이터를 하나의 검은 상자처럼 보았다.

그런데 그 안에는 휴머노이드 산업의 가장 흥미로운 기계 가운데 하나가 숨어 있다.

감속기.

모터의 빠르고 약한 회전을

느리고 강하며 정밀한 움직임으로 바꾸는 장치다.

정밀감속기 공급에는 일본 기업이 구축한 기존 축과 중국 내 생산·조달의 확대가 동시에 존재한다.

이를 '한 나라가 앞서고 다른 나라가 뒤쫓는다'는 한 줄로 환원하면 품질, 수명, 수율, 가격과 생산거점의 차이를 놓친다.

그리고 놀랍게도 대표적인 방식 하나는

얇은 쇳덩어리를 일부러 계속 찌그러뜨린다.

제1장 마지막 챗화

모터: "내가 로봇을 움직인다."

배터리: "내 전기가 없으면?"

AI: "내 명령이 없으면?"

센서: "내가 상태를 안 알려주면?"

감속기: "다들 말은 많은데, 네 빠른 회전을 실제 관절의 힘으로 바꾸는 건 나야."

손: "잠깐. 나까지 와 보면 누가 제일 어려운지 다시 얘기하자."

제2장 – 모터

전기는 어떻게 로봇의 근육이 되는가

로봇의 가슴에는 배터리가 있다.

배터리에는 전기가 저장되어 있다.

하지만 전기는 로봇의 팔을 들어 올릴 수 없다. 전기에너지를 **회전력**, 즉 토크로 바꾸는 장치가 필요하다.

그 장치가 **모터**다.

그리고 여기서 휴머노이드와 전기차가 처음으로 깊게 만난다.

V02. 모터 - 전기에서 토크로

개념도 · 축척 아님

제어된 전류가 자기장을 만들고, 회전자와의 상호작용이 토크를 만든다

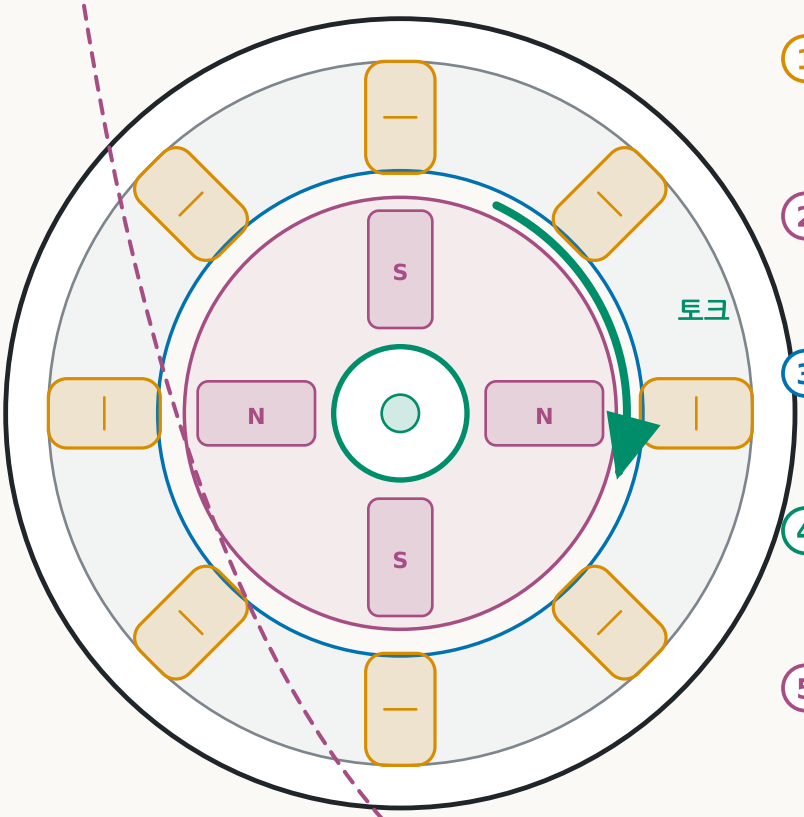
배터리
직류 전원

3상 인버터
스위칭·전류

고정자
회전자계

회전자
자석 추종

샤프트
토크 출력



1 고정자 철심·코일
전류 → 회전자계

2 회전자·영구자석
자기장을 따라 회전

3 공극
고정자와 회전자 사이

4 샤프트·베어링
토크 전달·하중 지지

5 엔코더
위치 신호 → 제어기

엔코더 피드백
전류 위상·회전 위치 보정

BLDC / PMSM

둘 다 넓게는 영구자석 동기기 계열이며 실제 산업 명칭은 겹칠 수 있다.
통상 BLDC는 6-step 계열, PMSM은 정현파 전류·FOC 계열의 강조가 크다.

1. 가장 이상한 힘, 자석

자석 두 개를 가까이 가져가면 밀거나 당긴다.

서로 직접 닿지 않았는데 힘이 발생한다.

모터는 바로 이 현상을 이용한다.

그러나 영구자석 하나를 놓아두는 것만으로는 지속적인 회전을 만들 수 없다.

필요한 것이 **전자석**이다.

구리 코일에 전류를 흘린다.

↓

자기장이 생긴다.

↓

전류 방향을 바꾼다.

↓

자기장 방향이 바뀐다.

↓

주변 영구자석이 밀리고 당겨진다.

↓

회전자가 돈다.

이것이 전동기의 가장 밑바닥 원리다.

---

2. 모터를 잘라보자

대표적인 영구자석 동기모터 계열을 단순화하면 크게 두 부분이다.

고정자 – Stator

움직이지 않는다.

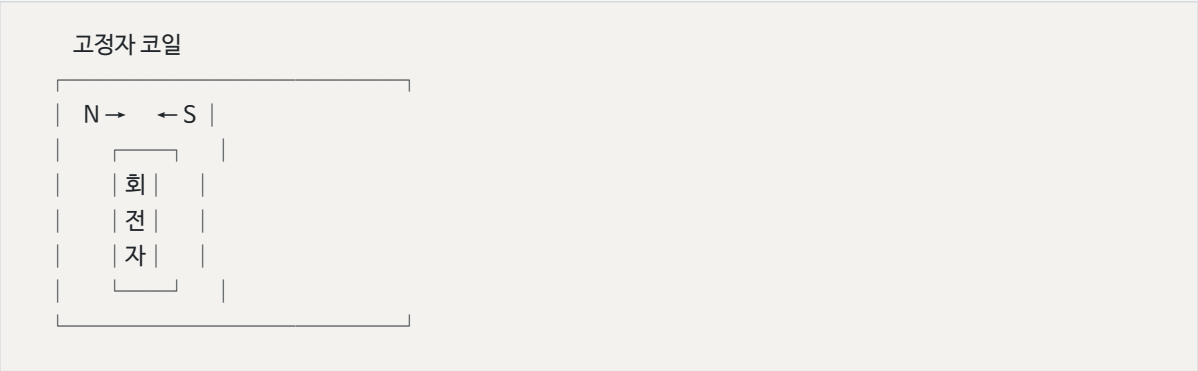
구리 코일이 들어 있다.

회전자 – Rotor

안쪽 또는 바깥쪽에서 회전한다.

영구자석 등이 들어 있다.

구조를 극단적으로 단순화하면,



제어기가 고정자 코일에 흐르는 전류를 계속 바꾸면 **회전하는 자기장**이 만들어진다.

회전자는 그 자기장을 따라 회전한다.

---

3. 챗화 – 자석을 쫓아가는 회전자

제어기

“북쪽 자기장, 여기!”

회전자:

“간다.”

제어기:

“이번에는 저쪽!”

회전자:

“또?”

제어기:

“이번에는 뒤!”

회전자:

“잠깐만….”

계속 쫓아가다 보니 회전한다.

모터는 거칠게 말하면 **계속 위치를 바꾸는 자기장을 회전자가 쫓아가게 만든 장치**다.

---

4. 그런데 왜 BLDC와 PMSM이라는 말이 계속 나오는가

예전 모터에는 브러시와 정류자라는 기계적 접촉부가 많이 사용됐다.

하지만 접촉하면 마모된다.

마찰이 생긴다.

열이 난다.

유지보수가 필요하다.

그래서 현대 전기차·드론·로봇에서는 전자적으로 전류를 제어하는 **브러시리스 계열 모터**가 중요해졌다.

BLDC와 PMSM은 완전히 다른 물리종이라기보다 영구자석 동기기 계열을 나누어 부르는 산업적 관행에 가깝다. 보통 BLDC는 사다리꼴 역기전력과 6-step 계열 구동, PMSM은 정현파 역기전력과 정현파 전류·FOC 계열 구동을 가리키지만 실제 제품과 제어에서는 경계가 겹칠 수 있다.

공통 축은 **전자적으로 제어되는 회전자계 + 영구자석 회전자**다.

---

5. 여기서 아주 중요한 부품이 등장한다

인버터

배터리는 기본적으로 직류 전원을 공급한다.

하지만 모터의 여러 코일에는 원하는 순간에 원하는 크기와 방향으로 전류를 흘려야 한다.

그래서 중간에서 전력반도체가 빠르게 스위칭한다.

```
배터리
 ↓
인버터
 ↓
3상 전류
 ↓
모터
 ↓
회전
```

따라서 로봇의 근육은 모터 하나가 아니다.

배터리 + 전력전자 + 모터 + 제어 알고리즘

이 함께 근육을 만든다.

---

6. 로봇에게 중요한 것은 마력이 아니다

자동차 광고에서는 흔히 출력이 강조된다.

그러나 로봇 관절에서는 다른 숫자가 특히 중요하다.

토크

얼마나 강하게 비틀 수 있는가.

문손잡이를 생각하면 쉽다.

손잡이 중심 가까이를 잡고 돌리면 힘들다.

끝부분을 잡으면 쉽게 돌아간다.

힘 × 회전축으로부터의 거리

가 토크와 연결된다.

로봇이 무거운 상자를 팔을 뻗어 들수록 어깨에는 훨씬 큰 토크가 필요하다.

---

7. 5kg은 언제나 5kg이 아니다

로봇이 5kg 물체를 몸 가까이 들고 있다.

어깨가 비교적 편하다.

이번에는 팔을 완전히 뻗는다.

물체의 질량은 그대로 5kg이다.

그런데 어깨가 감당해야 하는 토크는 커진다.

그래서 휴머노이드 사양의

> **"20kg을 들 수 있다."**

만 보고 성능을 판단해서는 안 된다.

어떤 자세에서, 어느 거리에서, 얼마나 오래가 중요하다.

---

8. 그리고 또 하나의 중요한 숫자

토크 밀도

큰 산업용 모터로 큰 힘을 만드는 것은 상대적으로 쉽다.

하지만 휴머노이드에는 거대한 모터를 붙일 수 없다.

무거워지고,

관절이 커지고,

다음 관절이 그 무게까지 들어야 한다.

그래서 중요한 것은

> **작고 가벼운 모터에서 얼마나 큰 토크를 얻는가**

가 핵심이다.

즉 **토크 밀도**가 중요해진다.

이것이 휴머노이드 모터 경쟁의 핵심 가운데 하나다.

다만 토크밀도 수치를 비교할 때는 kg당인지 L당인지, 모터만 센 것인지
감속기·하우징·드라이버까지 포함한 액추에이터인지, 순간 피크인지 열평형에서 유지 가능한
연속토크인지 확인해야 한다. 기준이 다르면 같은 숫자가 아니다.

---

9. 무게는 아래로 전염된다

오른팔의 모터를 500g 무겁게 만들었다고 하자.

문제는 500g으로 끝나지 않는다.

어깨가 더 큰 하중을 감당해야 한다.

↓

어깨 액추에이터가 커진다.

↓

몸통 프레임이 강해져야 한다.

↓

전체 질량이 증가한다.

↓

고관절·무릎·발목 하중이 증가한다.

↓

배터리 소비도 증가한다.

따라서 로봇공학에는 이런 직관이 필요하다.

> **손끝의 100g은 몸통의 100g보다 비쌀 수 있다.**

몸 중심에서 멀수록 관성 효과까지 커지기 때문이다.

---

10. 그렇다면 모터를 무한히 강하게 만들면 될까

안 된다.

첫 번째 적이 나타난다.

열

전류를 많이 흘리면 큰 자기장을 만들 수 있다.

토크가 커진다.

하지만 구리 코일에는 저항이 있다.

그래서 열이 발생한다.

대략적인 관계는 매우 직관적이다.

전류↑ → 발열 급증

즉 작은 모터에서 무리하게 큰 힘을 계속 뽑으면 뜨거워진다.

---

11. 순간 힘과 지속 힘은 다르다

이 구분은 로봇 영상을 볼 때 매우 중요하다.

로봇이 한 번 30kg을 들어 올리는 것과

8시간 동안 반복해서 물건을 옮기는 것은 전혀 다른 문제다.

Peak Torque

잠깐 낼 수 있는 최대 토크.

Continuous Torque

열적 한계를 넘지 않으면서 계속 낼 수 있는 토크.

화려한 데모는 전자를 보여주기 쉽다.

산업현장은 후자를 요구한다.

이 차이가 **데모 로봇과 노동 로봇**을 가르는 중요한 기준이다.

---

12. 열은 어디로 갈까

인간은 땀을 흘린다.

로봇은 그렇지 못한다.

열은

모터 AI 칩 배터리 감속기 전력전자

에서 동시에 발생한다.

그런데 이 모든 것을 사람 크기의 좁은 몸 안에 넣어야 한다.

따라서

전도,

공랭,

히트싱크,

열경로 설계,

경우에 따라 액체 냉각

등이 필요해진다.

열은 휴머노이드의 보이지 않는 장기다.

---

13. 모터 뒤에는 산업이 있다

고성능 모터는 자석·구리·전력반도체·정밀가공과 연결된다. 전기차와 드론 산업이 이 부품들의 생산기반을 넓혔기 때문에 휴머노이드는 완전히 빈 공급망에서 출발하지 않는다.

그러나 원자재, 국가별 공급망, 특정 기업의 제조전략은 모터의 작동원리와 다른 층의 문제다. 그 관계는 16~18장에서 따로 검증한다.

---

14. 하지만 자동차 모터를 그대로 무릎에 넣을 수는 없다

자동차와 휴머노이드는 요구조건이 다르다.

자동차 모터는 주로 바퀴를 계속 회전시킨다.

로봇 관절은

정지하고,

조금 움직이고,

반대 방향으로 움직이고,

버티고,

충격을 받고,

다시 움직인다.

이 동작을 끊임없이 반복한다.

특히 관절은 **0rpm 근처에서도 큰 토크**를 요구할 수 있다.

그래서 EV 기술은 훌륭한 기반이지만 휴머노이드용 액추에이터에는 별도의 최적화가 필요하다.

---

15. 여기서 이상한 문제가 생긴다

우리는 지금까지 좋은 모터를 만들었다.

작고,

가볍고,

빠르고,

강하다.

그런데 너무 빠르다.

로봇의 무릎이 원하는 것은 수천 rpm이 아니다.

천천히 움직이면서 엄청난 힘을 내야 한다.

그렇다면 방법은?

모터를 아주 거대하게 만들 수도 있다.

그러나 로봇이 무거워진다.

다른 방법이 있다.

속도를 힘으로 바꾼다.

그 역할을 하는 것이 바로 다음 장의 주인공이다.

감속기

그중에서도 아주 기묘한 장치가 있다.

원형의 금속컵을 일부러 타원으로 찌그러뜨린다.

그리고 모터가 돌 때마다

찌그러뜨리고 → 펴고 → 찌그러뜨리고 → 펴는 일을 수없이 반복한다.

그런데 그 쇳덩어리는 쉽게 부러져서는 안 된다.

더 이상한 것은 이 변형을 이용하면

작은 공간에서 큰 감속비와 높은 정밀도를 동시에 얻을 수 있다는 것이다.

제2장 마지막 챗화

모터: “빠르게 돌고 큰 힘도 낼 수 있어.”

무릎: “나는 빠른 게 필요 없는데.”

모터: “그럼 내 속도를 어떻게 하지?”

조용히 금속 컵 하나가 등장한다.

감속기: “그 속도, 나한테 줘.”

모터: “대신 뭘 주는데?”

감속기: “힘.”

제3장 – 감속기

왜 쇳덩어리를 일부러 찌그러뜨리는가

모터는 훌륭하다.

작고 빠르게 회전한다. 하지만 휴머노이드의 무릎은 분당 수천 번 회전할 필요가 없다. 필요한 것은 **천천히, 정확하게, 강하게 움직이는 것**이다.

그래서 모터와 관절 사이에 기묘한 기계가 들어간다.

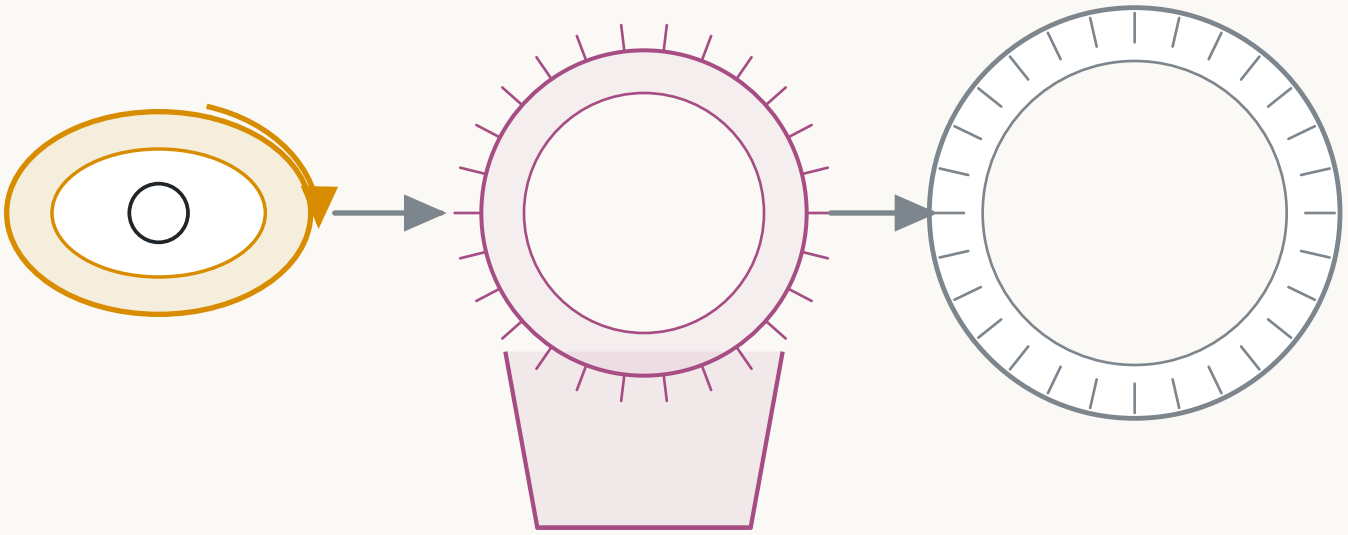
감속기(Reducer/Gearbox).

휴머노이드의 몸을 이해하는 데 이 부품이 중요한 이유는 단순하다.

V03. 스트레인 웨이브 감속기

작은 톱니 수 차이와 반복되는 탄성변형이 큰 감속비를 만든다

1 · 세 요소를 분리한다



① 웨이브 제너레이터

타원 캠·베어링
입력 회전

② 플렉스스플라인

얇은 컵·외부 톱니
탄성 변형·출력

③ 서큘러스플라인

강체 링·내부 톱니
고정 또는 출력

개념 예 · 톱니 수 차이는 단 2개

강체 링 **202T**

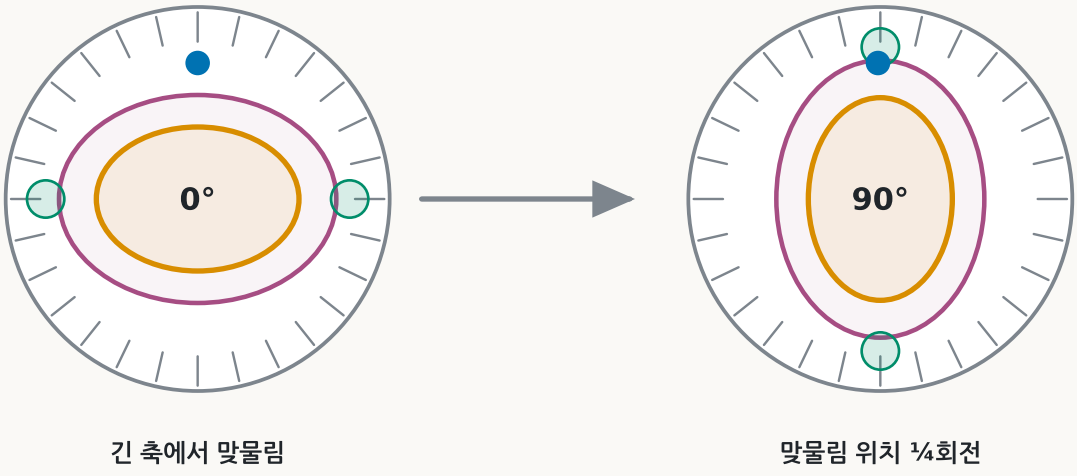
플렉스 링 **200T**

입력 1회전 → 출력은 아주 조금 반대방향

Harmonic Drive®는 대표 상표·제품 계열이며, 일반명은 스트레인 웨이브 기어다.

2 · 맞물림 영역이 이동한다

웨이브 제너레이터 입력각



초록 = 동시에 맞물리는 두 영역

파랑 = 느리게 이동하는 출력 표식

탄성변형은 원리의 핵심이지만, 수명은 하중·재료·열처리·윤활·조립에 달려 있다.

입력이 한 바퀴 돌아도 출력은 톱니 수 차이만큼만 이동한다.

> **모터가 힘을 만든다면, 감속기는 그 힘을 관절이 사용할 수 있는 형태로 바꾼다.**

---

1. 자전거 기어에서 시작하자

자전거를 타고 가파른 언덕을 올라간다고 생각해보자.

기어를 바꾼다.

속도는 느려진다.

하지만 페달을 밟기는 쉬워진다.

감속기도 비슷하다.

```
빠른 회전 + 작은 토크
        ↓
     감속기
        ↓
느린 회전 + 큰 토크
```

다만 중요한 것이 있다.

감속기가 에너지를 만들어내는 것은 아니다.

속도라는 것을 희생해 토크를 얻는다.

그리고 마찰과 변형 때문에 실제 출력에는 손실도 생긴다.

---

2. 100:1은 무슨 뜻인가

개념적으로 감속비가 100:1이라고 해보자.

모터축

100회전

↓

감속기

↓

출력축

약 1회전

그 대신 이상적인 경우 출력 토크는 크게 증가한다.

하지만 실제 기계에는 효율 손실이 있다.

따라서 정확한 사고방식은

> **속도↓ → 토크↑ + 일부 에너지는 열로 손실**

이다.

그렇다면 기어를 여러 개 이어 붙이면 되지 않을까?

가능하다.

하지만 휴머노이드에서는 또 다른 문제가 생긴다.

공간과 무게.

---

3. 사람 무릎 안에는 기어박스를 넣을 공간이 별로 없다

산업용 기계는 큰 감속기를 달 수 있다.

휴머노이드는 그렇지 않다.

사람 정도의 다리 굵기 안에

모터,

감속기,

베어링,

엔코더,

센서,

배선

을 전부 넣어야 한다.

그러면서도 큰 토크를 내야 한다.

그래서 로봇공학에서는 **작은 부피에서 큰 감속비를 얻는 방법**이 매우 중요해졌다.

그리고 여기서 등장하는 대표적인 장치가

스트레인 웨이브 기어

일반명은 **스트레인 웨이브 기어**다. Harmonic Drive®는 이 원리를 대표하는 상표·제품 계열로 널리 알려져 있다.

---

4. 부품은 놀랍게도 세 개가 핵심이다

① 웨이브 제너레이터

타원형 캠과 얇은 베어링으로 이루어진 입력부다.

모터가 이것을 돌린다.

② 플렉스 스플라인

얇은 컵 모양의 금속 부품.

바깥쪽에 톱니가 있다.

이름 그대로 **휘어진다.**

③ 서큘러 스플라인

바깥쪽의 단단한 원형 링.

안쪽에 톱니가 있다.



기묘한 것은 가운데 부품이다.

금속 기어인데 일부러 휘게 만든다.

---

5. 원을 타원으로 만든다

웨이브 제너레이터를 플렉스 스플라인 안에 넣는다.

그러면 원형이던 플렉스 스플라인이 타원으로 변형된다.

타원의 긴 축 부분에서는

플렉스 스플라인의 톱니와 바깥쪽 서큘러 스플라인의 톱니가 맞물린다.

90도 떨어진 부분에서는 떨어진다.

웨이브 제너레이터가 회전하면 이 **맞물리는 영역 자체가 원주를 따라 이동한다.**

바로 이것이 핵심이다.

---

6. 그런데 톱니 수가 다르다

여기서 마술 같은 일이 일어난다.

예를 들어 단순화해서

서큘러 스플라인:

202개

플렉스 스플라인:

200개

라고 해보자.

두 기어 사이에는 고작

2개의 톱니 차이

밖에 없다.

웨이브 제너레이터가 한 바퀴 돌면서 맞물리는 위치가 전체를 한 번 지나가도 플렉스 스플라인은 원래 위치로 완전히 돌아오지 않는다.

두 톱니만큼 뒤로 어긋난다.

따라서 입력축은 한 바퀴를 돌았는데 출력은 아주 조금만 움직인다.

---

7. 챗화 – 톱니 두 개의 장난

서큘러 스플라인

“내 톱니 202개.”

플렉스 스플라인

“나는 200개.”

서큘러

“두 개밖에 차이 안 나네.”

플렉스

“그 두 개 때문에 일이 커져.”

웨이브 제너레이터가 한 바퀴 돈다.

웨이브 제너레이터

“한 바퀴 완료!”

플렉스 스플라인:

“나는 두 톱니밖에 못 갔는데?”

바로 그 차이가 큰 감속비를 만든다.

---

8. 왜 이것이 휴머노이드에 매력적인가

첫째,

작다.

둘째,

가볍게 만들 수 있다.

셋째,

큰 감속비를 한 단계에서 얻을 수 있다.

넷째,

여러 톱니가 동시에 맞물려 하중을 나눌 수 있다.

다섯째,

잘 설계하면 **백래시를 매우 작게** 만들 수 있다.

여기서 또 하나의 중요한 단어가 등장한다.

백래시

9. 백래시가 무엇인가

낡은 자동차 핸들을 떠올려보자.

핸들을 조금 움직였는데 바퀴가 바로 반응하지 않는다.

좌우로 작은 헛움직임이 있다.

기어 사이의 틈 때문이다.

이런 기계적 유격이 **백래시**다.

산업기계에서는 불편할 수 있다.

하지만 로봇에게는 훨씬 심각하다.

로봇 손끝을 1mm 움직이려고 했다.

↓

어깨 기어에 작은 유격.

↓

팔꿈치에도 작은 유격.

↓

손목에도 작은 유격.

↓

오차가 손끝에서 누적된다.

정밀조립이나 수술처럼 작은 위치오차가 중요한 작업이라면 문제가 커진다.

---

10. 하모닉 계열이 정밀로봇에서 사랑받은 이유

플렉스 스플라인이 변형되면서 여러 톱니가 동시에 맞물린다.

잘 설계된 스트레인 웨이브 감속기는 매우 작은 백래시와 높은 위치 정밀도를 구현할 수 있다.

그래서

산업용 로봇,

반도체 장비,

우주기기,

의료로봇,

휴머노이드

같은 분야에서 중요한 선택지가 됐다.

그러나 세상에 공짜는 없다.

이 아름다운 구조에는 대가가 있다.

---

11. 플렉스 스플라인은 반복응력을 받는다

플렉스스플라인은 회전할 때마다 탄성 변형과 반복응력을 받는다. 이것이 정격 사용 중 필연적으로 끊어진다는 뜻은 아니다.

실제 수명은 하중 스펙트럼, 응력집중, 재료·열처리, 윤활, 조립오차와 정격 준수에 달려 있다. 따라서 핵심 질문은 '휘는가'가 아니라 **그 반복 변형을 요구 수명 동안 얼마나 일정하게 관리하는가**다.

---

12. 그래서 ‘원리를 안다’와 ‘제품을 만든다’는 완전히 다른 문제다

하모닉 계열의 작동 원리는 공개되어 있다.

그림을 보면 이해하기도 어렵지 않다.

그런데 좋은 제품을 만드는 것은 전혀 다른 문제다.

플렉스 스플라인의

소재,

두께,

열처리,

표면처리,

치형,

가공정밀도,

잔류응력,

윤활,

베어링,

조립공차

가 모두 수명과 정밀도에 영향을 준다.

즉 진입장벽은 ‘비밀 원리를 모른다’기보다

> **같은 물건을 수없이 만들어도 같은 성능과 수명이 나오게 하는 제조 노하우**

에 가깝다.

13. 특허보다 오래 남는 제조능력

작동원리가 공개되고 특허가 끝나도 소재·치형·열처리·표면처리·윤활·조립공차·수율은 자동으로 복제되지 않는다. 정밀감속기의 장벽은 비밀 원리보다 같은 성능과 수명을 반복 생산하는 공정능력에 가깝다.

국가와 기업별 공급망은 16~18장에서 다룬다.

---

14. 그런데 하모닉만 있는 것이 아니다

여기서 기존 영상 설명을 한 단계 확장해야 한다.

로봇의 모든 관절을 하모닉 감속기로 만들 이유는 없다.

유성감속기

여러 유성기어가 하중을 나눈다.

효율과 내구성이 좋고 이미 대량생산 기술이 성숙했다.

사이클로이드/RV 계열

사이클로이드 감속기는 편심운동과 사이클로이드 치형을 이용하는 넓은 계열이다. RV는 유성 전단과 사이클로이드 감속단을 결합한 대표적 정밀감속기 구조·제품군이다. 모든 사이클로이드 감속기를 RV라고 부르지는 않는다. 큰 하중과 높은 강성이 필요한 산업용 로봇에서 중요한 선택지다.

스트레인 웨이브

작고 큰 감속비와 정밀도가 필요한 곳에 강점이 있다.

볼스크루·롤러스크루 계열

회전운동을 직선운동으로 바꾸어 직선형 액추에이터를 구성할 수 있다.

그리고 완전히 다른 선택도 있다.

직접구동

감속기를 없애버린다.

---

15. 감속기를 없애면 안 되나?

가능하다.

모터가 관절을 직접 돌린다.

Direct Drive.

그러면

기어 마찰 ↓ 백래시 ↓ 기계 복잡성 ↓ 감속기 파손 문제 ↓

라는 엄청난 장점이 생긴다.

그런데 문제가 있다.

큰 토크를 직접 만들어야 한다.

그러면 모터가 커지고 무거워질 수 있다.

휴머노이드에서는 무게가 치명적이다.

따라서 이것도 공짜가 아니다.

---

16. 그래서 미래의 로봇은 한 가지 관절 구조를 쓰지 않을 가능성이 크다

손가락에 필요한 것과 무릎에 필요한 것은 다르다.

손:

가벼움 + 섬세함

팔꿈치:

정밀성 + 중간 토크

고관절:

큰 토크 + 강성

발목:

토크 + 충격대응 + 빠른 제어

따라서 최적의 휴머노이드는 관절 위치에 따라 서로 다른 전달기구를 사용할 수 있다.

이것이 **‘어떤 감속기 회사가 휴머노이드 시장을 독식한다’**는 단순한 산업 전망을 조심해야 하는 이유다.

---

17. 그리고 감속비가 높다고 무조건 좋은 것도 아니다

감속비를 크게 하면 작은 모터로 큰 토크를 얻기 쉽다.

하지만 다른 문제가 생긴다.

마찰,

효율,

관성,

역구동성,

제어 감각

등이 달라진다.

특히 중요한 개념이 하나 있다.

역구동성

역구동성은 출력 쪽에 가한 외력이 전달계를 거꾸로 움직일 수 있는 정도다. 감속비뿐 아니라 마찰, 모터 코깅, 반사관성, 브레이크와 제어상태의 영향을 받는다.

사람과 함께 일하는 로봇에 유리할 수 있지만, 컴플라이언스·힘감지·안전과 같은 뜻은 아니다.

---

18. 그래서 ‘강한 로봇’만 만드는 것이 답이 아니다

사람 옆에서 일하는 로봇이라면

강함과 동시에

부드러워야 한다.

이것은 모순처럼 들린다.

무거운 상자를 들어야 한다.

하지만 사람 팔과 충돌하면 즉시 힘을 빼야 한다.

그래서 현대 휴머노이드에서는

토크센싱,

임피던스 제어,

컴플라이언스,

충돌감지

가 중요해진다.

이 지점에서 감속기는 다음 장의 주인공과 합쳐진다.

---

19. 모터 + 감속기만으로는 아직 근육이 아니다

여기에

베어링,

엔코더,

토크센서,

온도센서,

브레이크,

전력전자,

제어기

를 하나로 묶는다.

그러면 비로소

액추에이터

가 된다.

그리고 이것은 단순한 모터 뭉치가 아니다.

스스로 자기 위치와 힘을 알고 조절하는 '지능형 근육 단위'에 가까워진다.

제3장 마지막 챗화

모터: "내가 5,000rpm으로 돌게."

감속기: "천천히."

모터: "대신 힘이 약해지잖아."

감속기: "아니. 속도를 내게 줘."

모터: "그리고?"

감속기: "관절이 쓸 힘으로 바꿔 줄게."

잠시 후.

플렉스 스플라인: "그런데 왜 매번 나만 찌그러뜨리는 건데?"

감속기: "그게 네 직업이야."

플렉스 스플라인: "그래서 소재와 열처리가 중요하군…."

여기서 하나의 중요한 발견

지금까지 우리는

배터리 → 모터 → 감속기

를 따로 봤다.

하지만 실제 로봇 회사는 이 셋을 따로 최적화해서는 좋은 로봇을 만들기 어렵다.

모터를 작게 한다
↓
감속비가 달라진다
↓
열이 달라진다
↓
관절 질량이 달라진다
↓
배터리 소비가 달라진다
↓
로봇 전체 설계가 달라진다

즉 휴머노이드의 경쟁력은 어느 하나의 '마법 부품'보다 **전체 시스템 공동설계**에서 나온다.

그리고 바로 이 때문에 Tesla처럼 액추에이터부터 전력·AI·제조까지 수직통합하려는 접근과, 전문 부품 생태계를 조합하는 접근을 별도로 비교해야 한다.

제4장 – 액추에이터

모터와 감속기는 어떻게 '근육'이 되는가

지금까지 모터와 감속기를 따로 열어봤다.

하지만 실제 로봇의 무릎을 열어보면 '모터 하나, 감속기 하나'가 느슨하게 연결되어 있지 않다. 베어링, 엔코더, 힘센서, 제어기까지 하나의 조밀한 모듈로 묶여 있다.

이것이 **액추에이터**(Actuator)다.

액추에이터는 단순한 근육보다 조금 더 복잡하다.

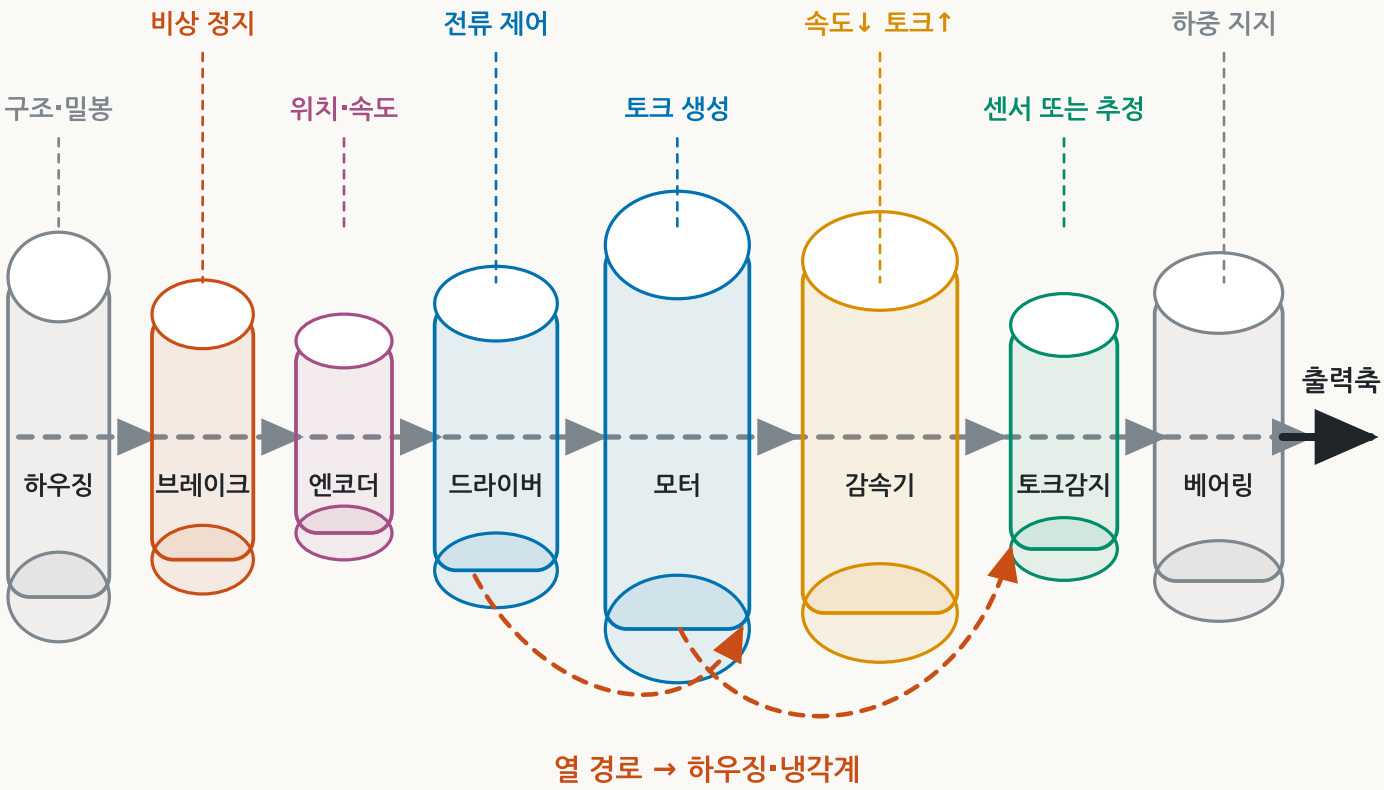
인간의 근육 + 힘줄 + 감각신경 일부를 한 통에 집어넣은 것에 가깝다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

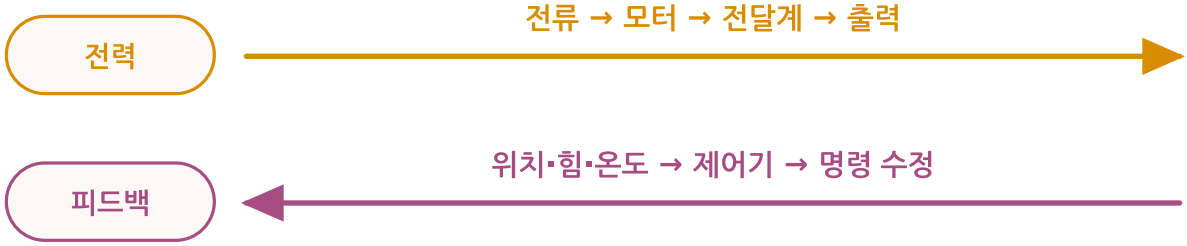
V04. 통합 액추에이터

관절 속에는 힘을 만들고·전달하고·측정하고·제어하는 작은 기계공장이 있다

회전형 액추에이터 · 폭발도



하나의 관절 안에서 동시에 닫히는 두 고리



모터 하나 ≠ 액추에이터 하나

개념도 · 회사순위 없음

직선형 액추에이터 · 회전을 밀고 당기는 힘으로



볼스크루 · 구름접촉

롤러스크루 · 높은 하중 목표

설계 계열은 우열이 아니라 작업별 절충이다

고감속 회전형

강점

높은 토크·컴팩트

대가

마찰·반사관성

역구동성 부담

QDD·저감속

강점

반응·힘제어

역구동성

대가

모터·전류·열 부담

직선형 스크루

강점

높은 직선 힘

하중경로

대가

패키징·수명·정밀가공

토크 · 속도 · 역구동성 · 충격 · 가격 · 수명 · 열

1. 관절 하나를 완전히 조립해보자

가장 단순화하면 이렇다.

```
배터리
  ↓
전력전자
  ↓
모터
  ↓
감속기
  ↓
베어링
  ↓
출력축
  ↓
팔·다리

  ↑
엔코더
토크센서
온도센서
  |
  ↓
제어기
```

그런데 여기서 질문이 생긴다.

모터가 돌았다는 사실과

관절이 정확히 움직였다는 사실은 다르다.

그래서 센서가 필요하다.

---

2. 엔코더 – 로봇의 고유감각

눈을 감고 오른팔을 90도 정도 굽혀보자.

팔을 보지 않아도 대략 어느 위치에 있는지 안다.

인간의 **고유감각** 때문이다.

로봇에서는 이 역할의 일부를 엔코더가 한다.

엔코더는 선형 또는 회전 위치를 증분식·절대식 신호로 제공한다. 속도는 위치의 시간변화로 계산할 수 있다. 그러나 엔코더 자체가 관절 토크나 구조물 변형 뒤의 실제 손끝 위치를 직접 재는 것은 아니다.

명령:

> "무릎 45도."

실제 측정:

43.8도.

제어기:

> "1.2도 부족."

다시 전류를 조절한다.

---

3. 이 과정은 한 번으로 끝나지 않는다

```
목표각도
  ↓
모터 명령
  ↓
관절 움직임
  ↓
엔코더 측정
  ↓
오차 계산
  ↓
모터 수정
  └──────────────←
```

이것이 **폐루프 제어**다.

로봇이 움직이는 동안 끊임없이 반복된다.

여기서 중요한 관점 하나.

> **로봇은 정확해서 정확하게 움직이는 것이 아니다.
> 계속 틀리고 계속 수정하기 때문에 정확하게 움직인다.**

이 원리는 앞으로 AI까지 그대로 이어진다.

---

4. 그런데 위치만 알아서는 부족하다

로봇이 사람과 악수한다고 생각해보자.

손의 위치는 완벽하다.

하지만 최대 출력으로 손을 쥔다.

좋은 로봇일까?

아니다.

위치뿐 아니라

얼마나 강한 힘을 가하고 있는가

를 알아야 한다.

그래서 토크센서와 힘센서가 중요해진다.

---

5. 토크센서 – 근육의 힘을 느낀다

로봇이 컵을 잡는다.

컵이 비어 있을 수도 있고,

물로 가득 차 있을 수도 있다.

위치만 보고는 알 수 없는 정보가 있다.

힘을 측정한다.

↓

예상보다 무겁다.

↓

관절 토크를 증가시킨다.

↓

컵을 안정적으로 든다.

이것이 로봇에게 **힘의 감각**을 준다.

---

6. 전류도 일종의 센서가 된다

모터가 만들어내는 토크와 전류 사이에는 관계가 있다.

따라서 모터 전류를 측정하면 어느 정도 힘을 추정할 수 있다.

그러나 마찰·감속기·온도·관성 등의 영향이 섞인다.

정밀한 힘 제어가 필요하면 별도의 토크센서가 유용하다.

여기서 비용과 성능의 절충이 시작된다.

센서를 더 달면 좋아진다.

하지만

가격↑ 무게↑ 배선↑ 고장점↑ 보정 필요↑

도 따라온다.

---

7. 베어링 – 너무 평범해서 잊어버리는 핵심부품

베어링은 화려하지 않다.

AI도 아니다.

그러나 액추에이터에서 매우 중요하다.

회전축을 지지하고,

하중을 받아내고,

마찰을 줄이고,

축의 위치를 유지한다.

로봇의 팔에는 단순한 회전력만 작용하지 않는다.

밀고,

당기고,

비틀고,

옆으로 누르는 힘이 동시에 들어온다.

따라서 베어링의 강성·수명·정밀도가 손끝의 정밀도와 액추에이터 수명에 영향을 준다.

---

8. 지금까지는 모두 회전형이었다

그런데 인간 근육을 생각해보자.

이두근 자체가 빙글빙글 도는 것은 아니다.

근육이 **수축**한다.

길이가 짧아지면서 관절을 당긴다.

로봇에서도 비슷한 방식이 가능하다.

직선형 액추에이터

모터의 회전을

볼스크루나 롤러스크루 등을 이용해

직선운동으로 바꾼다.

---

9. 볼스크루

나사를 생각하면 된다.

나사를 돌리면 앞으로 들어간다.

그러나 일반 나사는 마찰이 크다.

볼스크루는 나사산 사이에서 작은 볼들이 굴러간다.

그래서 마찰을 줄이고 정밀하게 회전을 직선운동으로 바꿀 수 있다.

```
모터 회전
  ↓
스크루 회전
  ↓
너트 이동
  ↓
직선 힘
  ↓
관절 회전
```

이제 로봇 관절은 인간의 근육과 조금 더 비슷해진다.

---

10. 롤러스크루

볼 대신 여러 개의 나사형 롤러를 사용하는 계열도 있다.

더 큰 하중과 높은 힘 밀도를 목표로 할 때 매력적일 수 있다.

하지만 제조 정밀도와 가격 등 다른 문제가 생긴다.

따라서 미래 휴머노이드에서 중요한 경쟁은 단순히

“하모닉 감속기냐 아니냐”

가 아니다.

회전형 액추에이터

대

직선형 액추에이터

라는 더 큰 설계 경쟁도 있다.

---

11. 왜 회사마다 로봇의 관절이 다른가

여기서 휴머노이드 기업들을 볼 때 새로운 눈이 생긴다.

Tesla, Figure, Unitree, Boston Dynamics가 똑같은 답을 선택할 필요는 없다.

왜냐하면 목표가 다르기 때문이다.

빠른 운동?

큰 힘?

저렴한 가격?

장시간 작업?

충격 내구성?

정밀한 손 작업?

대량생산?

이 중 무엇을 우선하느냐에 따라 최적 액추에이터가 달라진다.

---

12. 그런데 관절을 너무 딱딱하게 만들면 문제가 생긴다

사람이 걸을 때를 생각해보자.

발이 땅에 닿는 순간 충격이 생긴다.

하지만 인간의

근육,

힘줄,

연골,

관절

이 충격을 흡수한다.

로봇 관절이

모터 → 금속기어 → 금속프레임

으로 완전히 딱딱하게 연결되어 있다면 충격이 그대로 전달될 수 있다.

그래서 **컴플라이언스**, 즉 적절하게 양보하는 특성이 중요해진다.

---

13. SEA – 일부러 스프링을 넣는다

Series Elastic Actuator.

모터·전달계와 부하 사이에 탄성요소를 직렬로 넣는다. 그 변형량으로 힘이나 토크를 추정할 수 있고, 충격흡수와 저임피던스 힘제어에 유리할 수 있다.

그러나 공짜는 아니다. 큰 토크를 빠르게 전달하는 대역폭과 제어 안정성에 대가가 생길 수 있다. SEA는 '부드러움' 하나가 아니라 힘제어와 동적 성능 사이의 설계 절충이다.

---

14. 챗화 – 강한 것과 뻣뻣한 것은 다르다

철제 관절

"나는 절대로 안 휘어."

사람

"그런 네가 나랑 부딪히면?"

철제 관절:

"……."

스프링

"조금 양보하는 게 약한 건 아니야."

이 문장이 로봇공학에서 중요하다.

> **강한 로봇과 뻣뻣한 로봇은 같은 말이 아니다.**

---

15. QDD – 또 다른 절충

Quasi-Direct Drive.

고토크밀도 모터와 비교적 낮은 감속비 전달계를 결합해 마찰과 반사관성을 낮추려는 설계 계열이다. 높은 역구동성, 빠른 반응, 충격 대응과 힘제어를 노릴 수 있지만 모터의 크기·열·전류 부담이 커질 수 있다.

‘낮은 감속비’의 단일 국제 기준은 없으며, QDD를 사실상 직접구동과 같은 말로 쓰면 안 된다.

16. 역구동성은 왜 중요한가

로봇 팔의 출력 쪽을 사람이 밀었을 때 전달계가 거꾸로 움직일 수 있다면 역구동성이 높다고 말한다. 이는 외력을 센서로 감지해 모터가 능동적으로 비켜주는 제어와는 다르다.

협동·재활·서비스 작업에서 유리할 수 있지만, 역구동성이 높다고 자동으로 안전한 것도 아니다. 힘 제한, 충돌감지, 제동, 제어 안정성과 함께 설계해야 한다.

17. 그리고 현실에서는 로봇이 넘어진다

화려한 영상에서는 로봇이 백덤블링하고 달린다.

그러나 상용화에서 중요한 장면은 다른 것이다.

넘어진 뒤 관절이 멀쩡한가?

70kg의 로봇이 넘어지면 상당한 충격에너지가 관절로 들어간다.

그 순간 손상될 수 있는 것은

감속기 톱니,

베어링,

출력축,

엔코더,

프레임,

배선,

센서

등이다.

그래서 로봇의 진짜 성능에는 **충격 생존성**이 들어가야 한다.

18. 고장모드 – 데모 영상에 잘 나오지 않는 세계

액추에이터는 여러 방식으로 죽는다.

열

모터 권선 과열.

감속기

마모·피로·윤활 문제.

베어링

마모·유격 증가.

엔코더

위치정보 오류.

센서

드리프트 또는 고장.

전력전자

과전류·과열.

배선

반복 굽힘에 의한 단선.

마지막 것이 특히 재미있다.

로봇이 수백만 번 관절을 움직이면 **전선도 수백만 번 구부러진다.**

휴머노이드의 병목은 최첨단 AI가 아니라 몇 달 뒤 끊어지는 케이블일 수도 있다.

---

19. 이것이 데모와 제품의 차이다

한 번 멋진 동작을 하는 것은 어렵다.

하지만 같은 동작을

10번,

1,000번,

100만 번,

수천 시간

반복하는 것은 완전히 다른 공학 문제다.

따라서 산업용 휴머노이드를 평가할 때는 앞으로 네 숫자를 따로 보자.

성공률

사이클타임

가동률

고장 간 평균시간

멋진 동영상만으로는 이 숫자를 알 수 없다.

---

20. 산업 경쟁은 반복생산에서 벌어진다

좋은 액추에이터 한 개를 만드는 것과 수천 개를 같은 성능·수명·원가로 만드는 것은 다른 문제다.
모터·감속기·센서·베어링·전력전자·배선의 통합설계와 공정 데이터가 함께 필요하다.

국가와 기업의 제조전략은 뒤의 공급망·미국·중국 장에서 비교한다.

---

21. 그런데 액추에이터보다 더 어려운 것이 남았다

지금까지 만든 관절을 여러 개 연결하면 팔을 만들 수 있다.

팔 끝에는 무엇을 붙일까?

손.

그리고 여기서 문제가 갑자기 어려워진다.

걷는 로봇은 바닥이라는 비교적 큰 물체를 상대한다.

손은

바늘,

전선,

천,

비닐,

계란,

컵,

나사,

종이

를 상대해야 한다.

형태도 다르고,

마찰도 다르고,

무게도 다르고,

잡는 방법도 다르다.

게다가 물체는 손에 가려져 카메라에서도 잘 보이지 않는다.

그래서 필요한 것이 있다.

촉각.

제4장 마지막 챗화

AI

“계란을 집어.”

팔

“위치 도착.”

손가락

“닫는다.”

AI:

“조금 더.”

손가락:

“얼마나?”

AI:

“카메라로는 모르겠는데….”

계란

“빠직.”

잠시 침묵.

촉각센서

“그래서 내가 필요하다니까.”

제5장 – 손

왜 걷는 것보다 집는 것이 더 어려운가

휴머노이드 영상을 보면 사람들은 대개 걷는 모습에 먼저 놀란다.

두 발로 달린다. 발로 차고, 넘어졌다 일어나고, 심지어 춤도 춘다.

그런데 로봇공학자가 로봇을 공장이나 가정에 투입하려 할 때 훨씬 골치 아픈 질문이 기다리고 있다.

> **그래서 저 로봇이 구겨진 비닐봉지 하나를 집어서 열 수 있는가?**

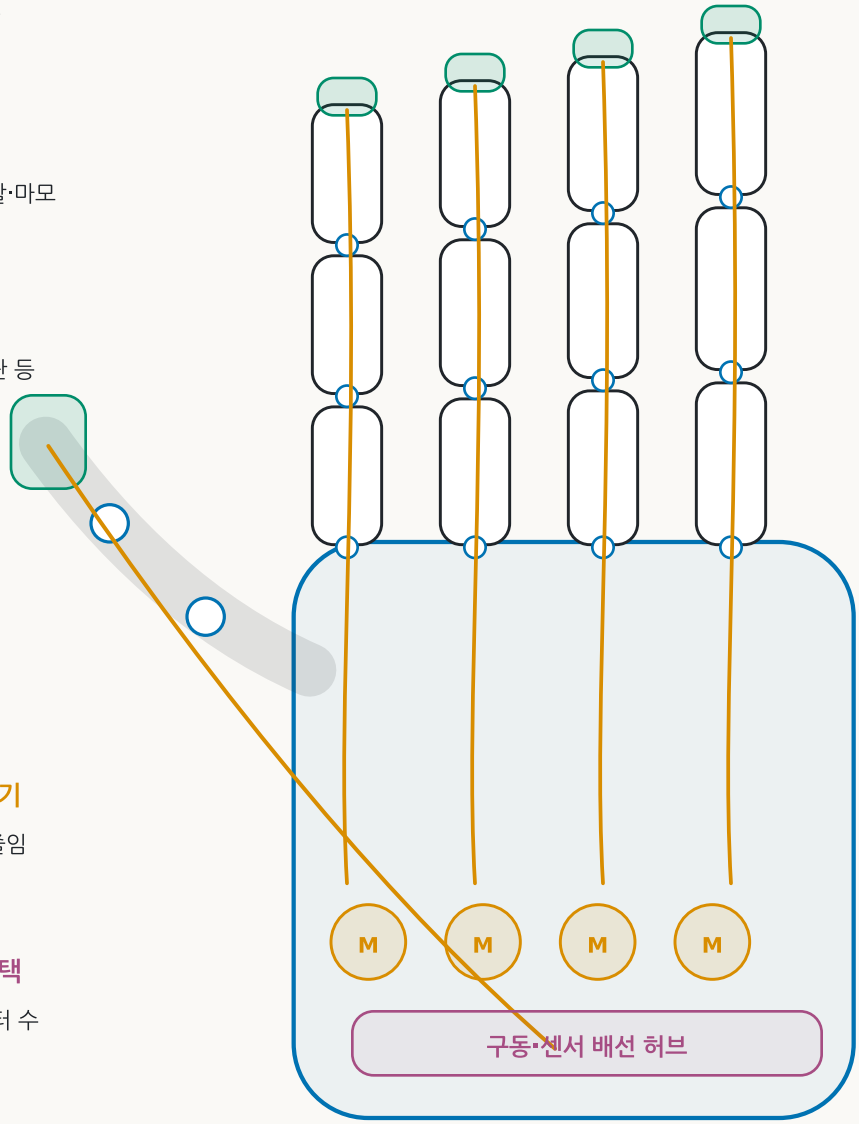
갑자기 문제가 어려워진다.

V05. 손과 촉각

손은 물체를 잡는 장치가 아니라, 접촉할 때마다 힘과 형태를 다시 협상하는 감각기관이다

1 · 손 안의 기계 시스템

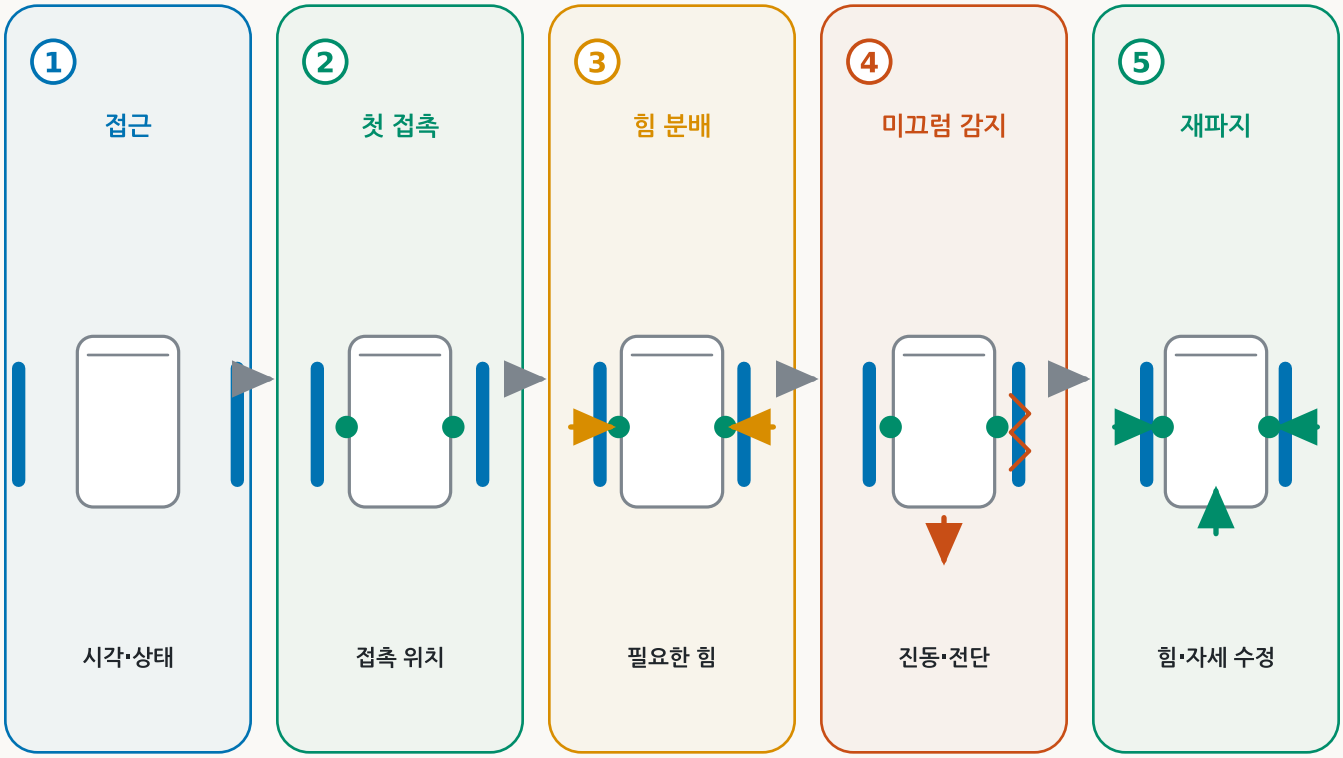
1. **관절**
   자유도·기구학
2. **힘줄·케이블**
   원격 구동·마찰·마모
3. **촉각 배열**
   접촉·압력·전단 등
4. **손바닥 구동기**
   손끝 무게를 줄임
5. **과소구동 선택**
   관절 수 ≠ 모터 수



손가락 수 ≠ 자유도 수 ≠ 액추에이터 수

직접구동·힘줄구동·과소구동은 작업마다 다른 절충이다.

2 · 접촉을 다섯 번 다시 쓴다



깨지기 쉬운 컵

과도한 힘 → 파손

부족한 힘 → 낙하

목표: 떨어지지 않을 만큼만

구겨지는 비닐

모양·경계가 계속 변함

잡는 동안 접촉전략 수정

목표: 형태 변화와 함께 제어

촉각 = 압력·법선/전단력·변형·진동·온도·접촉형상 등의 모달리티 묶음

모든 촉각센서가 미끄럼과 재질을 직접 측정하는 것은 아니다.

걷기는 상당 부분 **자기 몸을 제어하는 문제**다.

손은 자기 몸뿐 아니라 **세상의 거의 모든 물체와 물리적으로 협상하는 문제**다.

---

1. 인간의 손부터 이상하다

손 하나에는 27개의 뼈가 있다.

손가락은 굽혀지고,

펴지고,

벌어지고,

모이고,

엄지는 다른 손가락과 마주 볼 수 있다.

특히 엄지의 대립운동은 중요하다.

덕분에 인간은

망치를 움켜쥐고,

바늘을 집고,

연필을 쓰고,

단추를 잠그고,

수술도 한다.

하나의 기관으로 **괴력과 정밀성**을 동시에 처리한다.

로봇에게 이것을 구현하는 순간 기계설계가 폭발적으로 복잡해진다.

---

2. 손가락마다 모터를 하나씩 달면 되지 않을까

가능하다.

하지만 문제가 생긴다.

손이 무거워진다.

손목이 더 강해져야 한다.

팔꿈치가 더 강해져야 한다.

어깨가 더 강해져야 한다.

배터리가 커진다.

지난 장에서 본 악순환이 다시 시작한다.

```
손 무게 ↑
  ↓
손목 하중 ↑
  ↓
팔꿈치 하중 ↑
  ↓
어깨 하중 ↑
  ↓
전체 전력소비 ↑
```

그래서 **손끝의 100g**은 몸통의 100g보다 훨씬 비싼 무게가 될 수 있다.

---

3. 그래서 인간이 이미 쓰는 방법을 훔친다

손등을 보면서 손가락을 움직여보자.

손가락 끝에 커다란 근육이 달려 있지 않다.

많은 근육은 팔뚝 쪽에 있고,

힘줄이 손가락까지 이어져 힘을 전달한다.

로봇도 비슷한 구조를 사용할 수 있다.

Tendon Drive

모터를 손바닥 또는 팔뚝 쪽에 배치한다.

↓

케이블을 당긴다.

↓

손가락 관절이 굽어진다.

그러면 손끝을 가볍게 만들 수 있다.

---

4. 그런데 케이블을 쓰면 새로운 문제가 생긴다

마찰.

늘어남.

장력 변화.

마모.

케이블 경로.

백래시.

한 문제를 해결하면 다른 문제가 등장한다.

이것이 휴머노이드 설계의 반복되는 패턴이다.

> **완벽한 설계가 있는 것이 아니라 서로 다른 실패방식을 선택한다.**

---

5. 자유도라는 괴물이 다시 나타난다

손가락 관절을 모두 독립적으로 움직이면 손은 놀랍도록 자유로워진다.

하지만 자유도 하나를 늘릴 때마다

구동기,

센서,

배선,

제어 변수,

학습해야 할 행동 공간

도 증가한다.

따라서 실제 로봇 손에는 중요한 선택이 존재한다.

인간 손을 최대한 복제한다

많은 자유도.

높은 범용성 가능성.

그러나 복잡하고 비싸다.

자유도를 줄인다

구조가 단순하다.

싸고 튼튼하다.

그러나 할 수 있는 동작이 줄어든다.

---

6. 그래서 Underactuation이라는 영리한 방법이 등장한다

손가락 관절이 10개 있다고 해서 모터도 반드시 10개일 필요는 없다.

하나의 구동기가 여러 관절을 연동해 움직이게 할 수 있다.

물체에 닿으면 손가락이 물체 모양에 따라 자연스럽게 감싸진다.

이런 **과소구동**(underactuation)은 자유도보다 액추에이터 수를 줄일 수 있게 한다.

단순한 집기에서는 오히려 장점이 크다.

---

7. 챗화 – 엔지니어와 손

엔지니어

“손가락 관절을 전부 독립 제어하자.”

회계팀

“모터가 몇 개인데?”

엔지니어

“많습니다.”

회계팀

“센서는?”

“많습니다.”

“배선은?”

“많습니다.”

“고장날 곳은?”

엔지니어:

“……많습니다.”

그때 단순한 로봇 손이 말한다.

> **“나는 모터 몇 개로 상자를 잘 잡는데?”**

이것이 산업용 로봇 손 설계의 현실적인 질문이다.

---

8. 사실 모든 물건을 같은 방식으로 잡지 않는다

인간도 그렇다.

망치를 잡는다.

Power Grasp.

손 전체로 감싼다.

연필을 잡는다.

Precision Grip.

손가락 끝을 사용한다.

동전을 집는다.

Pinch.

가방 손잡이를 건다.

Hook Grip.

즉 범용 손을 만들려면 하나의 '잡기'가 아니라 여러 종류의 접촉전략을 학습해야 한다.

---

9. 그런데 카메라만으로는 부족하다

로봇이 컵을 잡는다.

카메라로 컵을 정확히 찾았다.

손을 정확한 위치로 이동시켰다.

손가락도 닫았다.

그런데 컵이 미끄러진다.

왜?

표면이 예상보다 매끄러웠다.

인간이라면 거의 생각하지 않고 손가락 힘을 조금 더 준다.

어떻게 알았을까?

촉각이다.

---

10. 촉각은 단순한 압력센서가 아니다

인간의 손끝은 놀라운 정보를 얻는다.

접촉했는가.

얼마나 세게 누르는가.

어느 방향으로 힘이 걸리는가.

미끄러지는가.

표면이 거친가.

물체가 변형되는가.

로봇에게도 이런 정보가 필요하다.

그래서 로봇 손에는

압력센서,

힘센서,

정전용량 센서,

광학식 촉각센서,

탄성체 기반 센서

등 여러 접근이 연구·적용되고 있다. 촉각은 하나의 센서 방식이 아니라
압력·법선력·전단력·변형·진동·온도·접촉형상 등을 다루는 모달리티의 묶음이다. 모든 촉각센서가 미끄럼과 재질을 모두 직접 측정하는 것은 아니다.

---

11. 가장 중요한 순간은 '미끄러지기 시작할 때'다

컵을 잡고 있다고 하자.

너무 약하게 잡으면 떨어진다.

너무 강하게 잡으면 깨질 수 있다.

정답은

떨어지지 않을 만큼만 잡는 것이다.

```
접촉
 ↓
압력 측정
 ↓
미세 미끄럼 감지
 ↓
그립 힘 증가
 ↓
안정
```

인간은 이것을 거의 무의식적으로 한다.

로봇에게는 엄청난 센서·제어 문제다.

---

12. 계란 문제가 다시 등장한다

계란과 쇠망치를 같은 손으로 집어야 한다.

계란:

작은 힘.

망치:

큰 힘.

종이컵:

찌그러지기 쉬움.

비닐:

형태가 계속 변함.

천:

정확한 경계조차 애매함.

전선:

가늘고 휘어짐.

여기서 로봇공학의 난이도가 갑자기 상승한다.

---

13. 강체와 변형체

전통적인 로봇공학은 단단한 물체를 상대하기 좋아한다.

볼트,

금속부품,

상자.

모양이 일정하다.

그런데 가정에는 이런 것들이 넘친다.

수건.

양말.

옷.

쓰레기봉투.

음식.

케이블.

이런 물체는 잡는 순간 **모양 자체가 변한다.**

따라서 미래 가정용 휴머노이드의 중요한 난제는 의외로

> **빨래 개기**

일 수 있다.

백덤블링보다 어렵다.

---

14. 왜 공장이 가정보다 먼저인가

이제 이유가 보인다.

공장은 물체를 표준화할 수 있다.

부품 크기를 일정하게 만든다.

상자를 일정하게 만든다.

작업대를 일정하게 만든다.

조명을 일정하게 만든다.

가정은 그렇지 않다.

어제 의자가 있던 곳에 오늘 빨래바구니가 있다.

바닥에 장난감이 있다.

접시는 제각각이다.

사람이 갑자기 지나간다.

고양이가 로봇 발 앞을 지나간다.

그래서

공장 → 물류 → 제한된 서비스 환경 → 가정

이라는 상용화 순서가 자연스럽게 나타난다.

---

15. 손은 AI 문제이기도 하다

기계적으로 완벽한 손을 만들었다고 끝이 아니다.

어떻게 잡을지를 알아야 한다.

카메라 영상

↓

물체 인식

↓

3차원 자세 추정

↓

잡을 위치 선택

↓

손 모양 결정

↓

접촉

↓

촉각 피드백

↓

힘 수정

↓

물체 이동

여기에 AI가 깊게 들어온다.

16. 손은 데이터 문제이기도 하다

손 작업에는 영상만으로 보이지 않는 접촉력, 손가락 상태, 미끄럼과 실패 시점이 필요하다. 그래서 로봇 손의 경험은 인터넷 영상보다 수집비용이 높다.

배치된 손이 유용한 실패와 교정 데이터를 만들면 학습에 도움이 될 수 있다. 다만 데이터의 양이 아니라 다양성·품질·행동결과·센서 정합성이 갖춰져야 한다. 데이터의 생산과 Fleet Learning은 12장에서 다룬다.

17. 손은 결국 세 가지 문제가 만나는 장소다

기계

작고 강하고 가벼운 손을 만든다.

감각

접촉과 미끄러짐을 느낀다.

지능

무엇을 어떻게 잡아야 하는지 결정한다.



그래서 손은 휴머노이드 전체의 축소판이다.

18. 로봇 손에서 가장 중요한 경쟁은 '손가락 개수'가 아닐 수 있다

사람처럼 다섯 손가락을 만드는 것이 멋져 보인다.

그러나 산업현장의 질문은 다르다.

작업 성공률은?

속도는?

몇 번까지 고장 없이 반복하는가?

가격은?

망가지면 손만 교체할 수 있는가?

그래서 미래의 범용 로봇 손이 반드시 인간 손과 똑같이 생긴다고 단정할 수 없다.

인간이 만든 공구를 사용할 정도로 인간과 비슷하면서도,

기계에 더 적합한 구조로 진화할 가능성이 있다.

---

19. 여기서 중요한 경제적 전환이 하나 생긴다

로봇이 단순히 상자를 옮긴다면 특수 그리퍼로 충분하다.

하지만 같은 로봇이

상자를 들고,

문을 열고,

전동드릴을 잡고,

케이블을 연결하고,

천을 접을 수 있다면

상황이 달라진다.

하드웨어 하나가 여러 작업에 사용된다.

즉 범용 손의 경제적 가치는

손 자체의 가격이 아니라

> **한 로봇이 수행할 수 있는 작업 종류의 증가**

에서 나온다.

이것이 휴머노이드 경제성의 핵심 변수 가운데 하나가 된다.

20. 그런데 손이 물체에 닿기 전에 해야 할 일이 있다

컵이 어디 있는지 알아야 한다.

얼마나 떨어져 있는지 알아야 한다.

자기 손이 어디 있는지도 알아야 한다.

사람이 접근하고 있는지도 알아야 한다.

즉 로봇은 세상을 **감각**해야 한다.

그래서 이제 손끝에서 고개를 든다.

로봇의 머리에 카메라가 있다.

몸통에는 IMU가 있다.

관절마다 엔코더가 있다.

발에는 힘센서가 있다.

이 모든 정보는 서로 다른 언어로 말한다.

그리고 로봇은 그것을 하나의 세계로 합쳐야 한다.

제5장 마지막 챗화

카메라

“컵을 찾았다.”

AI

“집어.”

손

“접촉.”

카메라

“잘 안 보여. 네 손에 가렸어.”

컵이 조금 미끄러진다.

촉각센서

"0.2초 전부터 미끄러지고 있었는데?"

손

"왜 이제 말해?"

촉각센서:

> **"나는 계속 말했어. 네가 듣는 법을 이제 배운 거지."**

제6장 – 감각

로봇은 어떻게 자기 몸과 세계를 동시에 아는가

손을 완성했다.

모터도 있고, 감속기도 있고, 촉각도 있다.

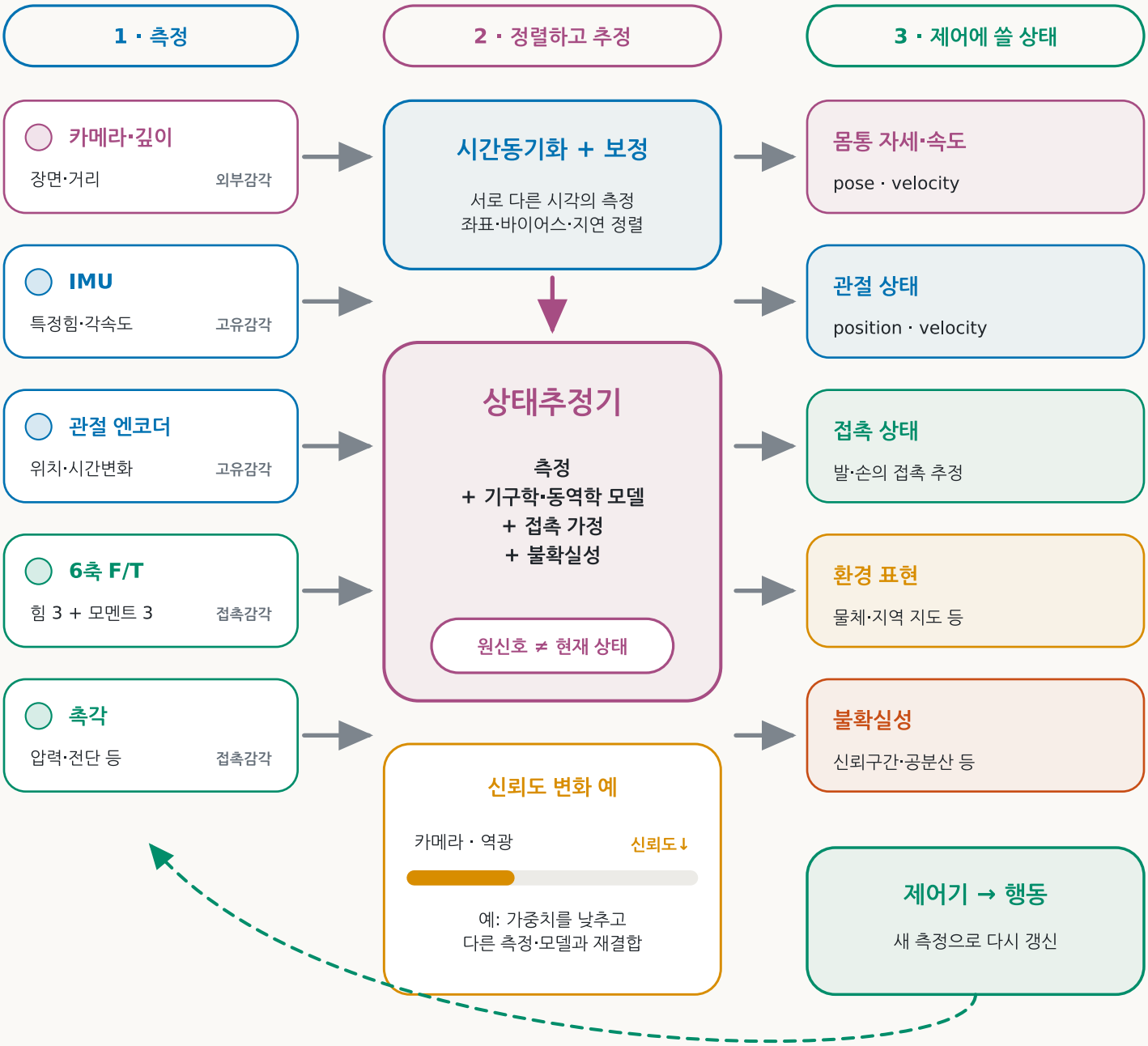
그런데 로봇의 전원을 켜는 순간 아주 근본적인 문제가 생긴다.

> **"나는 지금 어디에 있는가?"**

인간에게는 이상한 질문이다.

V06. 센서융합 - 측정에서 상태로

서로 다른 측정의 시간·오차·신뢰도를 결합해 몸과 세계의 상태를 만든다



센서융합

측정을 결합

상태추정

보이지 않는 상태를 추론

SLAM

지도 + 지도 속 위치

SLAM은 전신 상태추정 전체와 같은 알고리즘이 아니다.

우리는 눈을 감아도 왼팔이 대략 어디 있는지 알고, 몸이 앞으로 기울면 즉시 느낀다. 바닥이 미끄러우면 발에서 올라오는 정보와 평형감각을 거의 동시에 사용한다.

로봇은 이 모든 것을 **측정해서 계산해야 한다.**

그리고 여기서 중요한 사실이 하나 나온다.

> **로봇에게 세계를 아는 것과 자기 몸을 아는 것은 서로 다른 문제다.**

---

1. 로봇에게는 두 개의 세계가 있다

첫 번째는 **외부세계**다.

사람이 있다.

컵이 있다.

벽이 있다.

계단이 있다.

두 번째는 **자기 자신**이다.

오른팔이 37도 굽어 있다.

몸통이 왼쪽으로 2도 기울었다.

왼발에 체중의 60%가 실렸다.

오른손이 컵을 누르고 있다.

이 둘을 합쳐야 비로소 행동할 수 있다.

---

2. 인간도 사실 같은 일을 한다

눈을 감고 코끝을 손가락으로 만져보자.

코를 보지 않는다.

손도 보지 않는다.

그런데 대부분 꽤 정확하게 찾아간다.

뇌가 이미

팔의 위치,

관절 각도,

근육 상태,

머리 위치

를 알고 있기 때문이다.

이 감각을 **고유감각(Proprioception)**이라고 한다.

휴머노이드에도 기계적 고유감각이 필요하다.

---

3. 첫 번째 감각 – 엔코더

지난 장에서 본 엔코더다.

관절의

각도,

위치,

속도

를 알려준다.

예를 들어

왼쪽 고관절 12°

왼쪽 무릎 31°

왼쪽 발목 −4°

같은 정보가 계속 들어온다.

이 정보를 이용하면 로봇은 자기 다리의 형태를 계산할 수 있다.

---

4. 그런데 관절 각도만으로는 내가 넘어지는지 모른다

그래서 필요한 것이

IMU

관성측정장치다.

보통 **3축 가속도계 + 3축 자이로스코프**를 결합한다. 가속도계는 중력만이 아니라 센서가 받는 특정힘을, 자이로는 각속도를 측정한다. 자세·속도·위치는 이 신호를 적분하고 기구학·접촉·카메라 같은 정보와 결합해 추정한다. 그래서 짧은 시간에는 빠르고 유용하지만 장시간에는 바이어스와 적분오차가 누적될 수 있다. 자력계는 선택적이다.

스마트폰을 돌리면 화면 방향이 바뀌는 것도 비슷한 종류의 센서가 있기 때문이다.

---

5. 챗화 – 로봇이 넘어지기 직전

엔코더

“무릎 각도 정상.”

카메라

“앞에 아무것도 없음.”

IMU

“잠깐.”

모두:

“왜?”

IMU:

> **“몸통이 오른쪽으로 빠르게 회전 중이야.”**

제어기:

“오른발 토크 증가!”

발:

“지면 접촉!”

로봇은 다시 균형을 잡는다.

이 과정은 사람이 문장으로 생각하는 속도보다 훨씬 빨라야 한다.

---

6. 발바닥은 로봇의 또 다른 눈이다

사람은 발바닥으로 바닥을 느낀다.

로봇도 지면 접촉을 알아야 한다.

발이 바닥에 닿았는가?

얼마나 큰 힘이 걸리는가?

힘의 중심은 어디인가?

미끄러지고 있는가?

이런 정보가 보행제어에 중요하다.

따라서 발이나 발목에는 힘·토크 측정 계통이 사용될 수 있다.

---

7. 6축 힘·토크 센서

힘은 단순히 위아래로만 발생하지 않는다.

X축 힘 Y축 힘 Z축 힘

그리고

X축 회전모멘트 Y축 회전모멘트 Z축 회전모멘트

까지 있다.

즉

3개의 힘 + 3개의 모멘트

를 센서 좌표계에서 측정하면 6축 힘·토크 정보, 즉 wrench를 얻는다. 손목에서는 물체와의 상호작용을, 발목에서는 지면과의 상호작용을 추정하는 데 쓸 수 있다. 다만 센서만으로 접촉 위치나 접촉물체가 자동으로 주어지는 것은 아니며 기구학 모델과 접촉가정이 더 필요하다.

---

8. 이제 눈을 달자

가장 익숙한 센서다.

RGB 카메라

색,

형태,

텍스처

정보가 풍부하다.

그리고 매우 싸졌다.

하지만 일반 카메라 사진만 보면 중요한 것이 부족하다.

거리다.

컵이 보인다고 해서 컵까지 몇 cm인지 바로 아는 것은 아니다.

그래서 깊이정보가 필요하다.

---

9. 깊이를 보는 여러 방법

스테레오 비전

두 카메라의 시차를 이용한다.

인간의 두 눈과 비슷한 원리다.

구조광

특정 빛의 패턴을 투사하고 변형을 분석한다.

ToF

빛이 갔다 돌아오는 시간을 이용한다.

LiDAR

레이저를 이용해 주변 공간까지의 거리를 측정한다.

각 방식은

거리,

해상도,

조명,

가격,

전력,

크기

에서 장단점이 있다.

그래서 모든 휴머노이드가 같은 센서 구성을 사용할 이유가 없다.

---

10. LiDAR가 반드시 필요한가

자율주행차에서도 계속 등장한 질문이다.

LiDAR는 정확한 거리정보를 제공하는 데 유용하다.

그러나

비용,

무게,

전력,

설계 복잡성

이 따라온다.

카메라와 AI만으로 충분한 환경이라면 굳이 추가하지 않을 수도 있다.

반대로 높은 신뢰도의 3차원 측정이 중요하면 가치가 커질 수 있다.

따라서

카메라 대 LiDAR

를 종교전쟁처럼 볼 필요가 없다.

문제는 하나다.

> **주어진 작업에서 필요한 신뢰성을 가장 경제적으로 얻는 센서 조합은 무엇인가?**

---

11. 그런데 센서는 거짓말한다

여기서 문제가 시작된다.

카메라:

“저기에 사람이 있다.”

LiDAR:

“물체가 없다.”

IMU:

“몸이 기울었다.”

발센서:

“바닥은 안정적이다.”

엔코더:

“관절 위치 정상.”

누구를 믿을 것인가?

---

12. 완벽한 센서는 없다

카메라는 어둠과 역광에 영향을 받을 수 있다.

LiDAR도 환경 조건과 표면 특성 등에 영향을 받을 수 있다.

IMU는 시간이 지나면서 오차가 누적될 수 있다.

엔코더가 정확해도 구조물이 휘면 실제 손끝 위치와 차이가 생길 수 있다.

힘센서에는 노이즈와 드리프트가 존재한다.

따라서 로봇은 센서 하나를 절대적으로 믿지 않는다.

여기서 등장하는 것이

센서융합

이다.

---

13. 여러 명의 불완전한 증인을 합친다

법정에 증인이 다섯 명 있다고 생각해보자.

각자 본 것이 조금씩 다르다.

한 명을 무조건 믿는 것보다

각 증인의 신뢰도를 고려해 사건을 재구성하는 것이 낫다.

센서융합도 비슷하다.



센서융합은 서로 다른 측정과 불확실성을 결합하는 방법이다. 로봇은 센서 원본 그대로 행동하는 것이 아니라, 측정과 동역학·기구학 모델로부터 **직접 보이지 않는 현재 상태를 추정**한다. 센서융합은 상태추정의 수단일 수 있지만 두 말은 같은 뜻이 아니다.

---

14. State Estimation – 로봇에게 '현재'란 무엇인가

로봇이 알아야 할 현재 상태에는 이런 것들이 있다.

몸통 위치.

몸통 방향.

이동속도.

각 관절 위치.

각 관절 속도.

발의 접촉상태.

주변 물체 위치.

사람의 위치.

즉 센서 데이터가

세계의 내부 모델

로 바뀐다.

그제야 제어기가 행동을 결정할 수 있다.

---

15. SLAM – 지도와 위치를 동시에 알아내기

처음 들어간 방이라고 하자.

로봇에게 지도도 없다.

그런데 자기 위치도 모른다.

이상한 순환이다.

지도를 만들려면 내가 어디 있는지 알아야 한다.

내가 어디 있는지 알려면 지도가 필요하다.

이 문제를 함께 푸는 것이

SLAM – **동시적 위치추정과 지도작성**

이다. 이는 미지 또는 부분 미지 환경의 지도와 지도 안의 로봇 위치를 함께 추정하는 문제다. 고정 치구와 사전지도가 있는 모든 산업현장에 반드시 필요한 기능은 아니다.

```
주변을 본다
  ↓
특징을 찾는다
  ↓
조금 이동한다
  ↓
다시 본다
  ↓
변화를 비교한다
  ↓
내 이동량 추정
  ↓
지도 갱신
  ←
```

16. 하지만 휴머노이드에는 자율주행차보다 어려운 문제가 있다

자동차의 카메라는 비교적 안정된 차체에 붙어 있다.

휴머노이드의 머리는?

걸을 때마다 흔들린다.

위아래로 움직인다.

몸통이 회전한다.

발을 디딜 때 충격이 온다.

심지어 손으로 물체를 잡으면 몸 전체의 균형이 변한다.

즉 휴머노이드는 **움직이는 센서 플랫폼**이다.

17. 그리고 카메라가 자기 팔을 본다

이것도 재미있는 문제다.

카메라 영상에 로봇 자신의 손이 나타난다.

AI는 구분해야 한다.

저것은 외부 물체가 아니라

내 몸이다.

인간에게는 너무 당연하지만 로봇에게는 중요한 문제다.

이 지점에서

신체모델(Body Model)

이 중요해진다.

---

18. 로봇은 자기 몸의 지도를 가지고 있다

각 관절 길이.

회전축 위치.

손끝 위치.

관절 범위.

이런 정보를 이용해

> **"무릎이 20도이고 고관절이 10도라면 내 발은 대략 여기 있다."**

를 계산한다.

이를 다루는 핵심 도구가 **운동학**(Kinematics)이다.

반대로

> **"손끝을 저곳으로 보내려면 각 관절을 몇 도 움직여야 하는가?"**

를 푸는 것이 역운동학 문제다.

---

19. 그런데 현실의 로봇은 도면과 조금씩 다르다

금속이 휜다.

관절에 유격이 생긴다.

부품이 마모된다.

온도가 변한다.

하중이 걸린다.

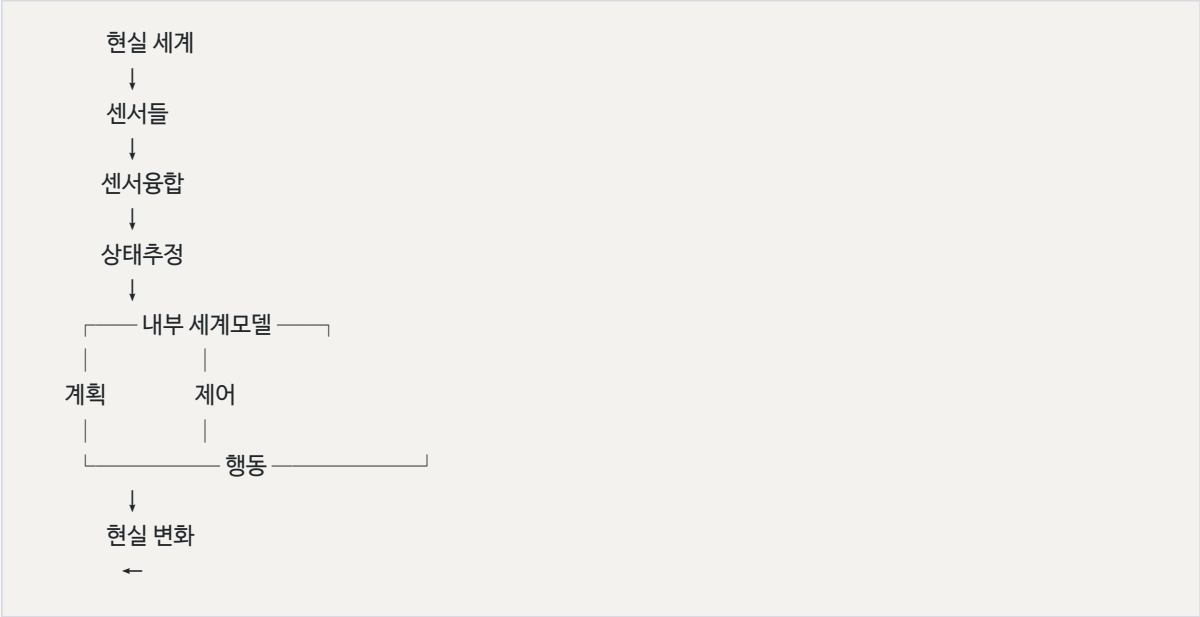
따라서 CAD 속의 완벽한 로봇과 실제 로봇 사이에는 항상 차이가 있다.

이것이 센서와 보정이 계속 필요한 이유다.

---

20. 여기서 로봇의 '현실'이 만들어진다

이제 중요한 구조가 완성된다.



이 순환은 한 번 실행되는 프로그램이 아니다.

로봇이 살아 있는 동안 계속 돈다.

---

21. 추정된 현재가 행동의 출발점이다

AI와 제어기는 센서의 원신호가 아니라 시간동기와 불확실성을 거쳐 추정된 상태를 사용한다. 더 비싼 센서를 다는 것과 더 나은 모델·보정·융합을 만드는 것은 같은 목표를 향한 서로 다른 선택이다.

따라서 감각계의 경쟁력은 센서 개수보다 필요한 신뢰도를 어떤 비용으로 확보하느냐에 달려 있다.

---

22. 이제 로봇은 세상을 볼 수 있다

컵이 어디 있는지 안다.

자기 손이 어디 있는지도 안다.

몸이 얼마나 기울었는지도 안다.

발이 바닥에 닿았는지도 안다.

그렇다면 이제 걸으면 될까?

아직 아니다.

로봇이 한 발을 들어 올리는 순간,

지지점이 두 개에서 하나로 줄어든다.

몸의 질량은 계속 움직인다.

관성이 작용한다.

바닥은 완벽하게 평평하지 않다.

그리고 중력은 단 한 순간도 쉬지 않는다.

제6장 마지막 챗화

카메라

“앞에 계단.”

엔코더

“왼쪽 무릎 24도.”

발센서

“오른발 지지.”

IMU

“몸통이 앞으로 기울기 시작.”

AI

“좋아. 왼발을 올려.”

실시간 제어기

“잠깐.”

AI:

“왜?”

제어기:

> **“네가 문장 하나 생각하는 동안 우리는 수십 번 넘어질 뻔했어.”**

제7장 – 걷기

왜 인간은 아무 생각 없이 하고, 로봇은 끊임없이 계산하는가

인간은 걷기를 우습게 생각한다.

자리에서 일어나 문까지 걸어가는 동안 우리는 발목 각도도, 무게중심도, 지면반력도 계산하지 않는다.

그런데 두 살배기 아이가 걷는 법을 배우는 데 오랜 시간이 걸린다는 사실을 생각하면 이야기가 달라진다.

걷기는 원래 어려운 문제다.

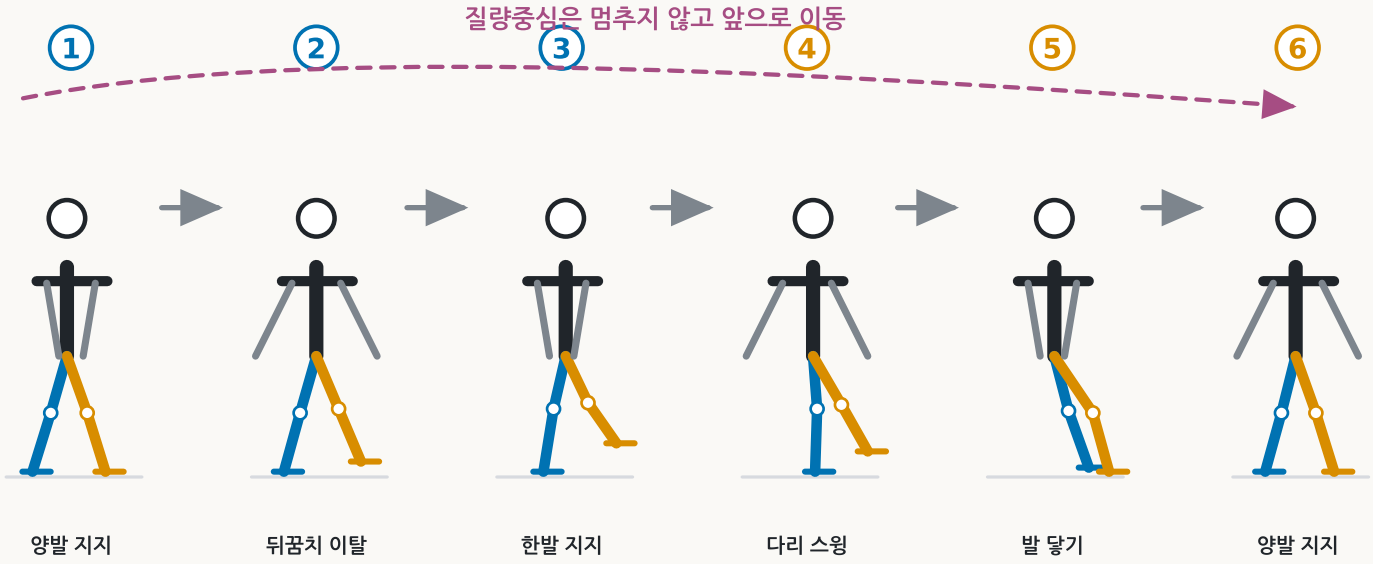
인간의 신경계가 너무 잘 해결하고 있어서 어려워 보이지 않을 뿐이다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

V07. 걷기 - 통제된 연속 낙하

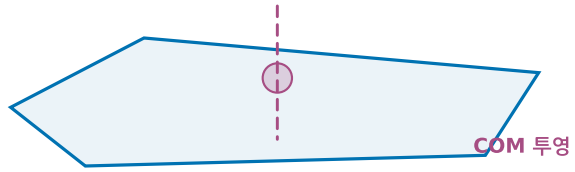
넘어지는 방향을 예측하고 다음 지지점을 만들어 낙하를 계속 회수한다

1 · 한 걸음의 여섯 순간

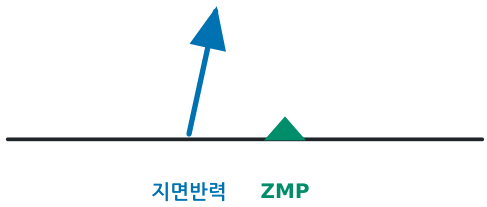


바닥에서 구분해야 할 네 가지

지지다각형



지면반력과 ZMP

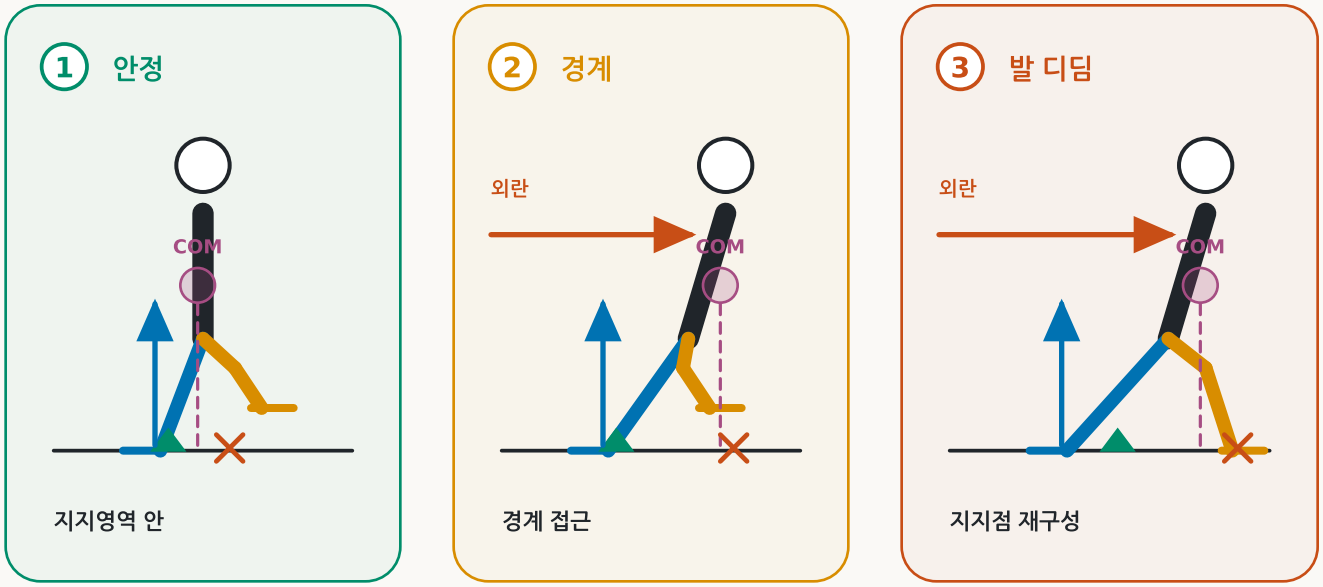


기호를 합치지 않는다

- **COM · 질량중심**
- **ZMP · 접촉력 모멘트 기준**
- **캡처포인트**

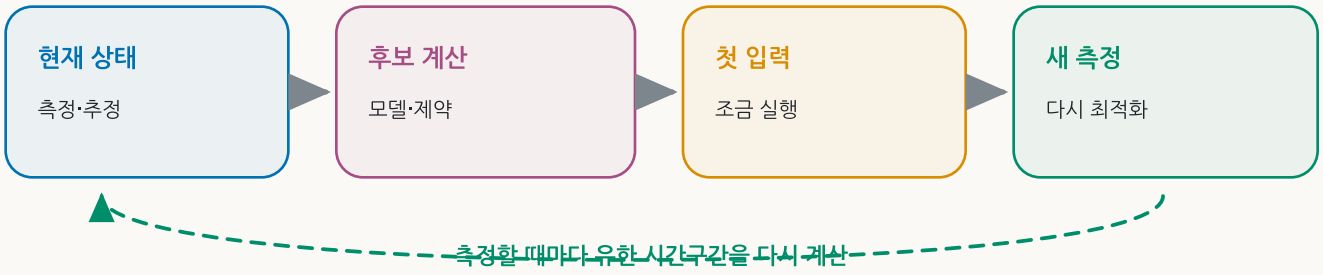
정적 그림은 출발점일 뿐이다. 움직일 때는 속도·접촉·외란까지 함께 본다.

2 · 외란 뒤 지지점 회복



발목·몸통으로 버티거나, 새 발을 내디뎌 지지영역을 다시 만든다.

MPC - 미래를 예언하지 않고 계속 다시 푼다



ZMP의 적용범위

평평한 발 접촉 보행의 중요한 기준이지만,
달리기·점프·손까지 쓰는 다중접촉 전체의 보편 충분조건은 아니다.

그리고 두 발로 걷는 순간 로봇은 가장 오래된 적과 싸운다.

중력이다.

---

1. 바퀴는 왜 쉬운가

자동차에는 보통 네 개의 바퀴가 땅에 닿아 있다.

무게중심이 바퀴가 만드는 넓은 영역 안에 있으면 안정적이다.

사람은 다르다.

두 발로 서 있다.

그리고 걷기 시작하면 한 발을 든다.

순간적으로 지지 영역이 크게 줄어든다.

즉 걷는다는 것은 사실

> **넘어지기 시작한 몸을 다음 발로 계속 받아내는 과정**

에 가깝다.

---

2. 정지해 있는 로봇부터 세워보자

양발이 바닥에 있다.

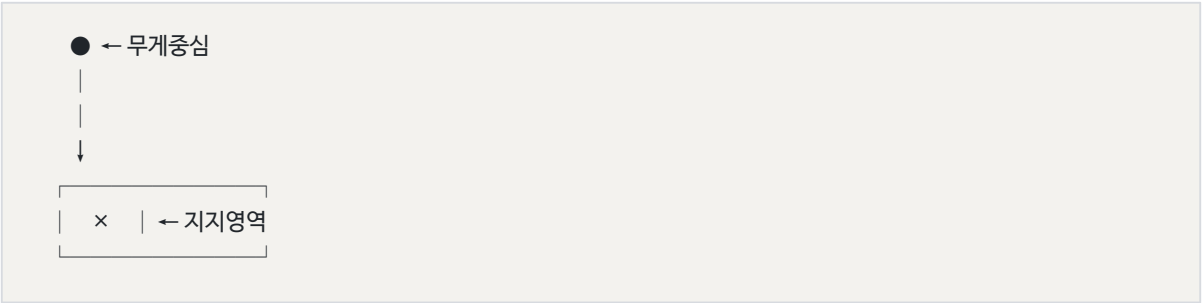
두 발이 만드는 영역을 단순화해서 **지지다각형**이라고 생각하자.

몸의 무게중심에서 중력 방향으로 선을 내려본다.

그 점이 안정적인 지지영역 안에 있으면 비교적 서 있기 쉽다.

밖으로 벗어나면?

넘어지기 시작한다.



정적인 경우는 그나마 쉽다.

하지만 사람이 걸으면 몸 전체가 가속한다.

그때부터 단순한 무게중심 그림만으로는 부족하다.

---

3. 발과 땅 사이에서 무슨 일이 일어나는가

로봇이 바닥을 누른다.

바닥도 로봇을 밀어 올린다.

이것이 **지면반력**이다.

뉴턴 역학이다.

발의 어느 부분에 얼마나 큰 힘이 걸리는지가 로봇의 균형을 결정한다.

발목센서와 힘센서가 중요한 이유가 여기에 있다.

---

4. ZMP

고전적인 이족보행에서 중요한 개념 가운데 하나가 Zero Moment Point다. 지지면에서 지면반력에 의한 수평축 모멘트가 0이 되는 점을 말한다.

평평한 발 접촉에서 ZMP를 지지다각형 안에 두는 것은 중요한 안정성 기준이다. 그러나 비행구간이 있는 달리기·점프나 손까지 쓰는 다중접촉의 모든 안정성을 설명하는 보편적 충분조건은 아니다.

---

5. 그래서 옛날 로봇은 조심스럽게 걸었다

기억해보면 초기 휴머노이드들은 걷는 모습이 조금 이상했다.

무릎을 굽히고,

천천히,

발바닥 전체를 조심스럽게 놓았다.

왜 그랬을까?

안정성을 확보하기 위해서다.

인간처럼 과감하게 몸을 앞으로 던지는 동작은 훨씬 어려웠다.

---

6. 인간은 사실 매 걸음마다 넘어지고 있다

사람이 걷는 모습을 옆에서 보면 몸의 무게중심은 앞으로 이동한다.

앞으로 쓰러지기 시작한다.

그때 한쪽 발을 앞으로 내밀어 몸을 받는다.

다시 앞으로 쓰러진다.

다른 발을 내민다.

따라서 동적보행의 재미있는 표현은 이것이다.

> **걷기는 통제된 연속 낙하다.**

---

7. 뛰기는 더 심하다

걷기에서는 적어도 한 발이 대개 땅에 있다.

달리기에는 순간적으로

두 발 모두 공중에 있는 구간이 생긴다.

그 순간에는 바닥에서 자세를 수정할 수 없다.

착지 위치와 자세가 잘못되면 넘어질 수 있다.

그래서 달리기·점프는 동역학적 제어 문제가 된다.

---

8. 그런데 계산할 시간이 거의 없다

로봇이 걷다가 발끝이 돌에 걸렸다.

언어모델에게 물어보자.

> **“제가 지금 넘어지고 있는데 어떻게 하면 좋을까요?”**

답변이 도착할 때쯤에는 이미 바닥에 누워 있을 것이다.

그래서 지난 장에서 말했던 **여러 속도의 뇌**가 필요하다.

```
상위 AI
"저 문으로 가라"
    ↓
경로 계획
"3m 전진"
    ↓
보행 정책
"다음 발 위치"
    ↓
실시간 제어
"발목 토크 수정"
    ↓
모터 제어
"전류 변경"
```

아래로 갈수록 훨씬 빠르게 반응해야 한다.

---

9. 로봇은 넘어지기 전에 알아야 한다

IMU:

몸통 회전속도 증가.

엔코더:

발목각도 변화.

발센서:

왼발 접촉력 감소.

제어기:

오른쪽으로 넘어질 가능성 증가.

그 순간 여러 전략을 사용할 수 있다.

발목으로 버틴다.

고관절을 움직인다.

팔을 휘두른다.

한 발을 옆으로 내딛는다.

사람도 비슷한 행동을 한다.

---

10. Push Recovery

누군가 로봇을 옆에서 민다.

옛 로봇은 그대로 넘어졌다.

좋은 동적보행 로봇은 상황에 따라

발목 토크를 바꾸고,

몸통을 돌리고,

발을 옮겨

새로운 지지점을 만든다.

이것이 **외란 회복**이다.

유튜브에서 로봇을 밀어보는 장면이 자주 등장하는 데에는 이유가 있다.

단순한 장난이 아니라 **제어기의 외란 대응능력을 보여주는 시험**이기 때문이다.

---

11. MPC – 미래를 조금 먼저 계산한다

한 가지 방법은 Model Predictive Control, MPC다.

현재 상태를 보고 바로 다음 행동만 결정하는 것이 아니다.

앞으로 몇 순간을 예측한다.

> **이 토크를 주면 0.2초 뒤 몸이 어디 있을까?**

> **저 위치에 발을 놓으면 0.5초 뒤 균형은 어떨까?**

현재 상태와 모델·제약을 사용해 유한 시간구간의 후보 행동을 최적화한다. 그중 첫 제어입력만 적용한 뒤 새 측정값으로 다시 푼다. MPC는 미래를 정확히 예언하는 기술이 아니라 모델을 이용해 계속 수정하는 최적화다.

```
현재 상태
  ↓
미래 예측
  ↓
행동 선택
  ↓
조금 실행
  ↓
새 상태 측정
  ↓
다시 예측
  ←
```

자동차가 커브를 미리 보는 것과 비슷하다.

---

12. 보행도 학습할 수 있다

모든 바닥과 외란을 사람이 규칙으로 열거하기는 어렵다. 강화학습은 환경과의 상호작용 데이터에서 기대 누적보상을 높이는 정책을 찾는다. 무작정 넘어뜨리는 것만을 뜻하지 않으며, 상태·행동·보상·알고리즘·안전제약을 사람이 설계한 조건 안에서 학습한다.

현실 로봇을 수없이 파손시킬 수 없으므로 많은 탐색은 시뮬레이션에서 이뤄진다. 그러나 가상세계와 현실은 다르다. Domain Randomization은 물리 파라미터·지연·잡음 등을 바꾸는 Sim-to-Real 방법 가운데 하나일 뿐이며, 시스템식별·현실 데이터·미세조정과 함께 쓰일 수 있다. 이 문제는 13장에서 깊게 다룬다.

13. 그렇다면 달리기를 잘하면 로봇 문제가 거의 해결된 것 아닐까?

아니다.

여기가 이번 장의 가장 중요한 반전이다.

로봇이 100m를 매우 빠르게 달렸다고 하자.

굉장한 기술이다.

그러나 공장장이 묻는다.

> **"좋습니다. 그런데 월요일부터 금요일까지 하루 8시간씩 일할 수 있습니까?"**

전혀 다른 시험이 시작된다.

14. 운동회와 공장의 차이

운동회

몇 분.

최고 성능.

관리팀 대기.

넘어져도 다시 준비.

공장

수천 시간.

반복작업.

먼지.

온도 변화.

배터리 감소.

부품 마모.

사람과 혼재.

생산라인 정지 비용.

여기서는 최고속도보다

신뢰성이 중요해진다.

---

15. 성공률은 작업횟수와 복구비용으로 읽는다

99%라는 숫자는 작업 빈도와 실패의 결과 없이 평가할 수 없다. 하루 1만 번 반복하면 단순 기대값으로 100번의 실패가 생길 수 있고, 스스로 복구하는 실패와 라인을 멈추는 실패는 경제적 의미가 다르다. 신뢰성과 복구시간은 19장에서 계산한다.

---

16. 그리고 또 하나

걷는 것 자체가 돈을 버는 것은 아니다.

공장에서 로봇이 100m를 10초에 달려도 별 가치가 없을 수 있다.

필요한 것은

부품을 가져오고,

기계에 넣고,

작업하고,

다음 작업으로 이동하는 것이다.

즉 로봇의 경제적 성능은

운동능력 × 작업능력 × 신뢰성

의 결합이다.

---

17. 그래서 휴머노이드의 ‘운동능력 경쟁’은 어느 순간 평준화될 수 있다

PC 역사에서 그래픽 데모가 중요했던 시기가 있었다.

스마트폰에서도 화면과 프로세서 성능 경쟁이 치열했다.

그러다 일정 수준을 넘으면 사용자는 다른 것을 보기 시작한다.

배터리.

가격.

앱.

신뢰성.

생태계.

휴머노이드에서도 비슷한 전환이 일어날 가능성을 검토해야 한다.

걷기 자체가 차별화 요소였던 시대

↓

충분히 잘 걷는 것이 기본조건인 시대

그 이후의 경쟁은 다른 곳으로 이동한다.

---

18. 그런데 로봇이 걸을 때 계속 에너지를 먹고 있었다

모터가 전기를 먹었다.

AI 칩도 전기를 먹었다.

센서도 전기를 먹었다.

감속기에서는 일부 에너지가 열로 변했다.

모터에서도 열이 났다.

배터리도 뜨거워졌다.

그리고 로봇의 가슴 안에는 인간처럼 거대한 연료탱크를 넣을 공간이 없다.

이제 휴머노이드의 가장 현실적인 적 가운데 하나를 만나게 된다.

에너지.

제7장 마지막 챗화

AI

“달려.”

로봇이 달린다.

“점프.”

점프한다.

“상자를 들어.”

들어 올린다.

AI:

“훌륭해. 이제 8시간 일하자.”

배터리

“잠깐.”

AI:

“왜?”

배터리:

> **“나한테 물어본 사람은 아무도 없네?”**

뒤에서 열관리 시스템이 조용히 말한다.

> **“나한테도.”**

제8장 – 배터리

왜 전기차는 수백 km를 가는데 휴머노이드는 몇 시간밖에 일하지 못하는가

로봇이 달렸다.

계단을 올랐다.

상자를 들었다.

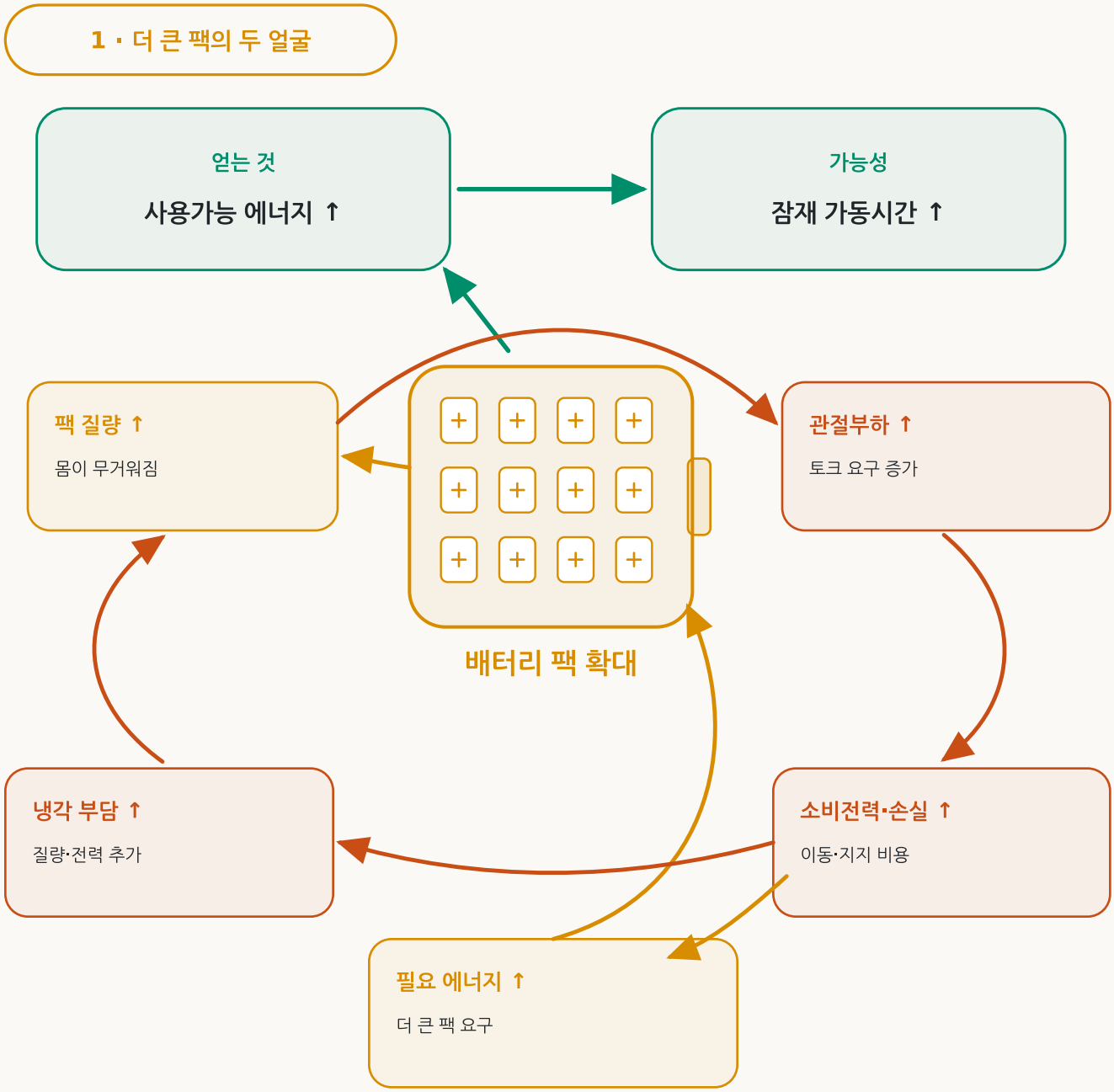
손으로 부품을 조립했다.

영상만 보면 이제 인간 노동자를 대신할 것 같다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

V08. 배터리·열 - 오래 일하는 몸

가동시간은 용량 하나가 아니라 질량·효율·열·충전 운영이 함께 만든다

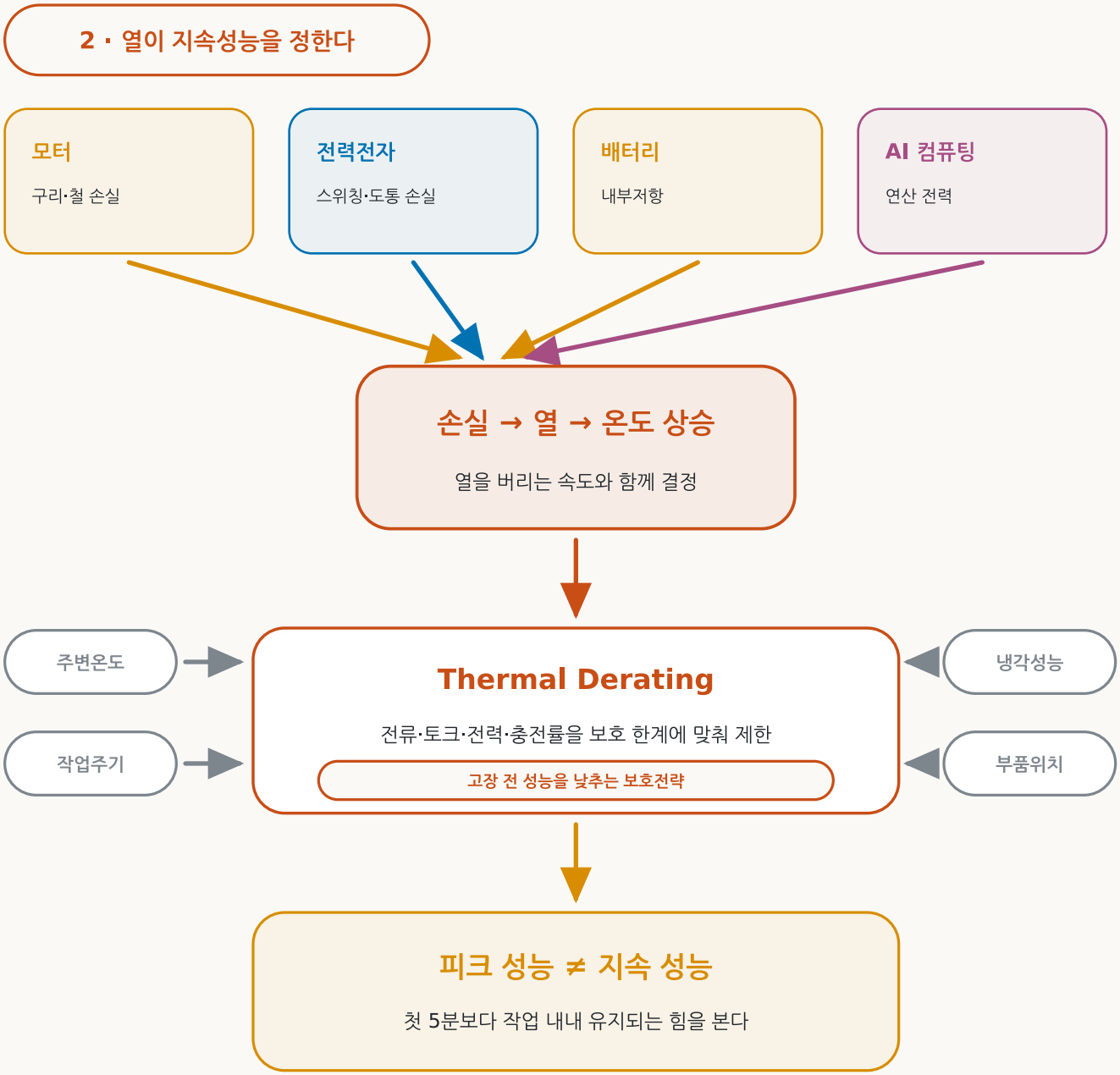


배터리는 자기 질량을 옮기는 데에도 에너지를 쓴다.

BMS 측정	전압·전류·온도	BMS 추정	SOC·SOH · 모델·이력 사용

보호·밸런싱·충전관리 범위는 시스템마다 다르다.

인과도 · 수치 없음



가동시간은 몸·작업·운영을 합친 시스템 결과

효율 개선 | 경량화 | 작업 설계

교대 충전 | 배터리 교환 | 열관리

그런데 공장 관리자가 아주 평범한 질문을 한다.

> **"그래서 한 번 충전하면 몇 시간 일합니까?"**

갑자기 화려했던 로봇공학이 배터리 잔량 표시 하나 앞에서 현실로 돌아온다.

휴머노이드에게 배터리는 단순한 연료통이 아니다.

몸무게이며, 작업시간이며, 열원이며, 안전장치이며, 경제성이다.

---

1. 먼저 W와 Wh부터 구별하자

배터리를 이해할 때 가장 먼저 헷갈리는 두 단위가 있다.

W – 와트

지금 얼마나 빠르게 에너지를 쓰고 있는가.

즉 **출력**이다.

Wh – 와트시

얼마나 많은 에너지를 저장하고 있는가.

즉 **에너지량**이다.

욕조에 비유하면 쉽다.

Wh = **욕조에 들어 있는 물의 양**

W = **수도꼭지에서 물이 빠져나가는 속도**

---

2. 아주 단순한 로봇을 만들어보자

배터리에 2,000Wh가 저장돼 있다고 하자.

로봇이 평균 500W를 사용한다면 이상적으로는

약 4시간.

평균 1,000W라면

약 2시간.

하지만 실제로는 그렇게 단순하지 않다.

로봇의 소비전력이 계속 바뀌기 때문이다.

---

3. 가만히 서 있다고 전기를 안 쓰는 것이 아니다

휴머노이드에는 동시에 여러 소비자가 있다.

```
배터리
 |
 ├── 관절 모터
 ├── 손 액추에이터
 ├── AI 컴퓨터
 ├── 카메라·센서
 ├── 통신장치
 ├── 냉각계통
 ├── 전력전자
 └── 각종 제어기
```

그리고 모터의 소비전력은 작업에 따라 크게 변한다.

평지 보행과

계단 오르기,

빈손과

20kg 물체 운반은 같을 수 없다.

따라서

> **"이 로봇은 몇 시간 갑니까?"**

라는 질문에는 반드시

"무슨 일을 하면서?"

가 붙어야 한다.

---

4. 제조사가 공개한 런타임은 조건표와 함께 읽어야 한다

2026년 9월 현재 제조사 공개 사양만 보아도 숫자는 크게 다르다.

Unitree G1은 약 2시간, R1은 약 1시간을 제시한다.

Boston Dynamics는 제품형 Atlas의 전형적 배터리 시간을 약 4시간으로 적는다.

Agility Robotics는 Digit 5에 대해 약 90분 작업과 약 9분 충전, 교대 충전을 통한 하루 20시간 이상의 생산적 사용을 제시한다. 다만 Digit 5의 일반 공급은 2027년 계획이다.

이 숫자들은 모두 **제조사 사양 또는 계획**이다. 동일한 작업과 시험조건에서 측정한 비교시험 결과가 아니다.

따라서

4시간 로봇 > 2시간 로봇 > 90분 로봇

이라고 결론 내릴 수 없다.

몸무게,

작업강도,

컴퓨팅,

이동속도,

페이로드,

대기시간,

시험조건

이 다르기 때문이다.

충전·교환시간과 하루 가동전략까지 다르다.

스마트폰 배터리처럼 하나의 숫자로 비교하기 어려운 이유다.

---

5. 그렇다면 배터리를 두 배로 넣으면 되지 않을까

이론적으로는 간단하다.

2kWh가 부족하다.

↓

4kWh를 넣는다.

그런데 배터리는 에너지만 늘어나는 것이 아니다.

질량도 늘어난다.

그리고 여기서 휴머노이드 특유의 악순환이 시작된다.

```
배터리 ↑
  ↓
몸무게 ↑
  ↓
관절 토크 ↑
  ↓
모터 소비전력 ↑
  ↓
프레임 요구강도 ↑
  ↓
몸무게 ↑
  ↓
배터리 요구량 ↑
  ←
```

배터리가 자기 자신을 운반하기 위해 전기를 먹기 시작한다.

---

6. 자동차와 결정적으로 다른 점

전기차도 배터리가 무겁다.

하지만 자동차는 네 바퀴 위에서 굴러간다.

바퀴는 매우 효율적인 이동수단이다.

휴머노이드는 다르다.

매 걸음마다

다리를 가속하고,

감속하고,

몸의 균형을 잡고,

관절을 움직인다.

심지어 팔까지 흔들린다.

즉 이동 자체가 복잡한 다관절 운동이다.

그래서 단순히

전기차 배터리를 작게 만들어 등에 넣는다

고 해결되지 않는다.

---

7. 에너지 밀도가 중요한 이유

배터리가 10kg이라고 하자.

문제는 무게 자체보다

> **10kg 안에 얼마나 많은 에너지를 넣을 수 있는가**

를 묻는 것이다.

이것이 **중량 에너지 밀도**다.

또 하나는

> **같은 부피 안에 얼마나 많이 넣는가.**

즉 **체적 에너지 밀도**다.

휴머노이드는 둘 다 중요하다.

몸통 공간도 작고 전체 질량에도 민감하기 때문이다.

---

8. 왜 배터리를 다리에 넣지 않을까

넣을 수는 있다.

그러나 움직이는 팔다리 끝에 무거운 질량을 놓으면 문제가 커진다.

다리를 앞뒤로 흔들 때마다 그 질량을 가속하고 감속해야 한다.

관성이 커진다.

그래서 큰 질량은 가능하면 몸의 중심 가까이에 배치하는 것이 유리하다.

많은 휴머노이드가 배터리를 몸통 쪽에 두는 이유 중 하나다.

---

9. 여기서 로봇 설계의 중요한 원칙이 다시 등장한다

> **1kg은 어디에 있느냐에 따라 가치가 다르다.**

몸통의 1kg.

허벅지의 1kg.

발목의 1kg.

손끝의 1kg.

동역학적으로 동일하지 않다.

몸 중심에서 멀어질수록 움직일 때 관성 부담이 커질 수 있다.

그래서 액추에이터와 배터리의 **질량 배치** 자체가 설계 기술이다.

---

10. 배터리는 에너지만 저장하지 않는다

배터리는 위험한 화학 에너지를 몸 안에 들고 다니는 장치다.

과충전.

과방전.

과전류.

과열.

셀 불균형.

내부 단락.

충격.

이것을 관리해야 한다.

그래서 필요한 것이

BMS – Battery Management System

이다.

---

11. BMS는 배터리의 경비실이다

각 셀의 전압을 본다.

온도를 본다.

전류를 본다.

이 측정값과 배터리 모델·사용 이력을 바탕으로 충전상태와 열화상태를 **추정**한다. SOC와 SOH는 온도처럼 센서 하나가 직접 재는 물리량이 아니다.

이상하면 출력을 제한한다.

심각하면 보호회로·퓨즈·접촉기와 함께 출력을 차단한다.

다만 BMS가 있다고 위험이 사라지는 것은 아니다. 셀 결함, 충격, 냉각 실패, 센서·접촉기 고장까지 포함한 팩 설계와 시스템 수준의 보호가 함께 작동해야 한다.

즉 배터리는

셀만 모아놓은 상자

가 아니다.

```
배터리 셀
  +
센서
  +
BMS
  +
퓨즈·접촉기
  +
열관리
  +
기계적 보호
  =
배터리 시스템
```

12. 그런데 로봇 배터리에는 자동차와 다른 안전문제가 있다

자동차 배터리는 주로 차체 아래쪽에 있다.

휴머노이드 배터리는 사람과 가까운 높이에서 움직인다.

넘어질 수 있다.

충돌할 수 있다.

공구와 부딪힐 수 있다.

사람 옆에서 작업할 수도 있다.

따라서 배터리의

충격보호,

열폭주 대응,

전기적 차단

은 상용화에서 매우 중요하다.

13. 배터리의 또 다른 적 – 순간 출력

로봇이 천천히 걷다가 갑자기 무거운 물체를 든다.

여러 관절이 동시에 큰 토크를 요구한다.

순간적으로 전력수요가 튄다.

즉 배터리는 많은 에너지만 저장하면 되는 것이 아니다.

필요할 때 **큰 출력도 공급할 수 있어야 한다.**

다시 두 개념이 갈라진다.

에너지 밀도

대

출력 밀도

둘 다 좋아야 한다.

---

14. 그리고 전류가 커지면 다시 열이 온다

배터리 내부에도 저항이 있다.

큰 전류를 흘리면 열이 발생한다.

모터에서도 열.

인버터에서도 열.

AI 칩에서도 열.

감속기에서도 마찰열.

이제 휴머노이드 몸통 안에서는 여러 열원이 서로 싸우기 시작한다.

그래서 배터리 문제는 자연스럽게 다음 문제로 연결된다.

열관리.

---

15. 그런데 질문 자체를 바꾸면 어떨까

지금까지 우리는 이렇게 물었다.

> “어떻게 8시간 가는 배터리를 만들까?”

그런데 왜 8시간이어야 할까?

사람이 8시간 교대근무를 하기 때문이다.

로봇까지 인간의 생리적 노동시간을 따라야 할 이유는 없다.

여기서 설계철학이 바뀐다.

---

16. 3시간 일하고 스스로 충전하면?

로봇 A:

8시간 배터리.

무겁다.

비싸다.

로봇 B:

3시간 배터리.

가볍다.

3시간마다 스스로 충전한다.

공장 전체의 생산성이 B가 더 높다면?

B가 더 좋은 노동기계다.

즉 중요한 숫자는 배터리 지속시간이 아니다.

하루 총 생산시간

이다.

---

17. 로봇에게 커피시간은 필요 없다

사람은

출근하고,

쉬고,

식사하고,

퇴근한다.

로봇은 다르게 설계할 수 있다.

```
Robot A → 작업
Robot B → 충전

    ↓ 교대

Robot B → 작업
Robot A → 충전
```

이 경우 개별 로봇의 8시간 연속 작동이 중요하지 않을 수 있다.

공장에서는 Fleet, 즉 로봇 집단 전체의 가동률이 더 중요해진다.

---

18. 배터리를 갈아버리는 방법도 있다

충전 기다리기도 싫다면?

배터리를 교환한다.

빈 배터리 제거.

↓

충전된 배터리 장착.

↓

즉시 작업 복귀.

중국의 UBTECH가 Walker S2에서 자율 배터리 교환 개념을 보여준 것도 이 문제와 연결된다.
회사는 교환에 약 3분이 걸리며 이를 통해 24시간 연속운영이 가능하다고 설명한다.

여기서 **24시간은 배터리 한 팩의 지속시간이 아니다.** 충전된 팩으로 스스로 교체하는 시스템 운용능력에 관한 기업 발표다.

이 접근의 핵심은 '배터리가 얼마나 오래 가는가'가 아니라

> **에너지를 보충하는 동안 생산을 얼마나 멈추는가**

가 핵심이다.

---

19. 여기서 자동차와 또 다른 길이 갈라진다

자동차 한 대는 사람 한 명이 소유한다.

충전 중에는 자동차가 쉬어도 된다.

공장 로봇은 생산설비다.

1분의 정지가 비용이다.

따라서 로봇 배터리 기술의 최적점은 전기차와 다를 수 있다.

초대형 배터리보다

작고,

가볍고,

빠르게 충전하거나,

쉽게 교체되는 배터리가

더 경제적일 수도 있다.

---

20. 그리고 배터리가 작아지면 예상 밖의 일이 생긴다

배터리 ↓

↓

질량 ↓

↓

필요 토크 ↓

↓

액추에이터 크기 ↓

↓

전력소비 ↓

↓

필요 배터리 ↓

다시 선순환이 시작된다.

```
경량 배터리
  ↓
경량 로봇
  ↓
작은 액추에이터
  ↓
낮은 소비전력
  ↓
더 작은 배터리
  ←
```

이것이 휴머노이드에서 **시스템 최적화**가 개별 부품 성능보다 중요한 이유다.

---

21. 회생제동도 가능할까

가능하다.

전기차는 감속할 때 모터를 발전기처럼 사용해 일부 에너지를 배터리로 돌려보낸다.

로봇 관절에서도 조건에 따라 유사한 에너지 회수가 가능하다.

예를 들어 다리를 감속시키거나 중력에 의해 관절이 구동될 때 일부 에너지를 전기적으로 회수할 수 있다.

그러나 모든 동작에서 가능한 것은 아니며 변환손실도 있다.

따라서 **회생제동이 배터리 문제를 없애주는 마법은 아니다.**

하지만 효율 최적화의 한 요소다.

---

22. 사실 가장 좋은 전기는 쓰지 않는 전기다

배터리 용량을 20% 늘리는 방법만 있는 것이 아니다.

소비전력을 20% 줄일 수도 있다.

더 효율적인 모터.

더 좋은 보행.

낮은 마찰.

효율적인 감속기.

경량 프레임.

필요할 때만 센서 활성화.

저전력 AI 칩.

효율적인 냉각.

그래서 배터리 경쟁은 사실

로봇 전체 효율 경쟁이다.

---

23. 계산도 전기를 먹는다

카메라 처리와 VLA 추론, 통신과 냉각도 배터리를 사용한다. 그래서 로봇은 모든 계산을 가장 큰 모델에 보내지 않고, 빠른 제어는 가까운 저전력 계층에서, 복잡한 의미 판단은 필요할 때 상위 계층에서 처리하도록 설계할 수 있다.

결국 배터리 문제는 셀 용량만이 아니라 몸의 질량, 작업계획, 계산 배치와 충전 운영을 함께 바꾼다.

---

24. 이제 열을 따로 떼어내야 한다

모터가 뜨겁다.

배터리가 뜨겁다.

AI 칩이 뜨겁다.

인버터가 뜨겁다.

감속기도 뜨겁다.

그런데 휴머노이드에는 자동차처럼 넓은 차체도, 데이터센터처럼 거대한 냉각장치도 없다.

사람 크기의 몸 안에서 열을 밖으로 버려야 한다.

그리고 열을 버리지 못하면 아주 흥미로운 일이 일어난다.

로봇은 고장 나기 전에 스스로 성능을 낮춘다.

30kg을 들던 로봇이 시간이 지나면서 20kg밖에 못 들 수도 있다.

최고속도를 낮출 수도 있다.

AI 칩의 연산속도도 떨어질 수 있다.

즉,

> **로봇의 진짜 성능은 첫 5분의 성능이 아니라 몇 시간 뒤에도 유지되는 성능이다.**

제8장 마지막 챗화

공장장

“몇 시간 일할 수 있어?”

로봇

“세 시간입니다.”

공장장:

“사람은 여덟 시간인데?”

로봇:

“사람은 여덟 시간 내내 일합니까?”

공장장:

“…….”

로봇:

“저는 세 시간 일하고 10분 만에 배터리를 바꾸겠습니다.”

옆의 로봇이 작업을 이어받는다.

공장장은 계산기를 꺼낸다.

그 순간 질문이 바뀐다.

> **‘몇 시간 가는가?’가 아니라 ‘하루에 몇 시간 생산하는가?’**

제9장 – 열

로봇은 왜 지치지 않으면서도 '지치는가'

로봇은 근육통이 없다.

숨이 차지도 않는다.

졸리지도 않는다.

그렇다면 전기만 공급하면 하루 종일 최대 출력으로 일할 수 있을까?

그렇지 않다.

기계에도 일종의 **피로와 비슷하게 보이는 현상**이 있다.

그 정체는 상당 부분 **열**이다.

휴머노이드 내부에서는 거의 모든 중요한 부품이 동시에 열을 만든다.

모터.

감속기.

배터리.

전력전자.

AI 칩.

그리고 사람 크기의 몸 안에서 이 열을 계속 밖으로 버려야 한다.

---

1. 에너지는 사라지지 않는다

배터리에서 100이라는 전기에너지가 나왔다고 하자.

그 100이 모두 손끝의 유용한 움직임으로 바뀌지는 않는다.

일부는

모터의 전기저항,

인버터 손실,

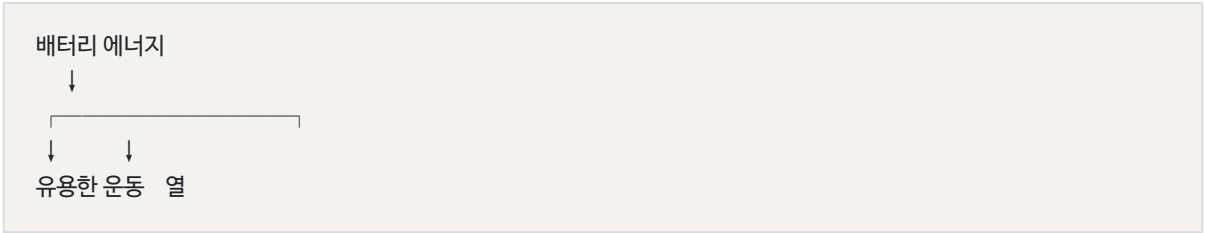
베어링 마찰,

감속기 마찰,

배터리 내부저항,

컴퓨터 연산

과정에서 열이 된다.



효율이 좋아질수록 열은 줄어든다.

하지만 0이 되지는 않는다.

---

2. 모터는 왜 뜨거워지는가

모터 코일은 구리로 만들어진다.

구리도 전기저항이 0은 아니다.

전류가 흐르면 열이 발생한다.

그리고 여기서 중요한 특징이 있다.

전류를 조금 높이는 것과 발열의 증가는 단순한 1대1 관계가 아니다.

대표적인 구리손실은 전류의 제곱에 비례한다.

따라서 큰 토크를 얻기 위해 전류를 계속 올리면 발열 부담이 빠르게 증가한다.

---

3. 그래서 Peak Torque와 Continuous Torque가 달랐다

앞에서 잠깐 본 개념을 이제 정확히 이해할 수 있다.

Peak Torque

짧은 시간 동안 낼 수 있는 큰 힘.

Continuous Torque

열을 지속적으로 방출하면서 계속 유지할 수 있는 힘.

로봇이 50kg 물체를 한 번 들었다는 영상과

50kg을 하루 종일 반복해서 옮길 수 있다는 말은 완전히 다르다.

---

4. 챗화 – 로봇의 헬스장 기록

로봇

“나 50kg 들 수 있어.”

공장장

“좋아. 하루에 2,000번 들어.”

로봇:

“그건 다른 사양인데요.”

공장장:

“왜?”

모터

“저 110도입니다.”

인버터

“저도 뜨겁습니다.”

배터리

“나도.”

공장장:

“너희 기계잖아.”

모터:

> **“그래서 열역학에서 면제되는 건 아닙니다.”**

---

5. 열이 왜 위험한가

단순히 뜨거워서 불편한 것이 아니다.

모터 권선 절연재가 열화될 수 있다.

영구자석의 성능이 온도에 영향을 받을 수 있다.

윤활유의 특성이 변한다.

베어링 수명이 영향을 받는다.

전자부품의 신뢰성이 떨어질 수 있다.

배터리는 특히 높은 온도에 민감하다.

따라서 부품 온도·열모델·보호한계에 따라 전류·토크·전력·충전률을 제한할 수 있다. 이를 흔히 Thermal Derating, 열에 따른 성능 제한이라고 한다. 제한 시점과 정도는 주변온도, 작업주기, 냉각과 부품 위치의 함수다.

---

6. 로봇도 '힘이 빠진다'

처음 5분.

최대속도로 움직인다.

큰 물건도 든다.

30분 뒤.

모터 온도 상승.

1시간 뒤.

제어기가 출력 제한.

따라서 실제 산업현장에서 중요한 것은

최대성능

보다

지속성능

이다.

이것은 앞으로 휴머노이드 기업의 데모를 볼 때 아주 유용한 기준이다.

---

7. 배터리도 뜨거워진다

충전할 때.

방전할 때.

큰 전류를 사용할 때.

배터리 내부저항 때문에 열이 난다.

온도가 지나치게 낮아도 성능이 떨어지고, 지나치게 높으면 열화와 안전 문제가 커진다.

따라서 배터리는 적절한 **온도 창** 안에서 관리해야 한다.

BMS와 열관리 시스템이 함께 움직이는 이유다.

---

8. AI의 뇌도 뜨겁다

여기서 Physical AI 특유의 문제가 나온다.

로봇은 단순한 전동기계가 아니다.

카메라 영상을 계속 처리한다.

물체를 인식한다.

음성을 이해한다.

경로를 계산한다.

행동정책을 실행한다.

AI 추론도 에너지를 소비한다.

그리고 그 에너지의 상당 부분은 결국 열이 된다.

---

9. 데이터센터라면 쉽다

AI 서버가 뜨거워진다.

거대한 팬을 단다.

냉각수를 순환시킨다.

에어컨을 돌린다.

랙 사이에 공간을 둔다.

휴머노이드?

사람 크기의 몸이다.

더구나 걸어 다녀야 한다.

냉각장치 자체도 무게다.

팬도 전기를 먹는다.

펌프도 전기를 먹는다.

냉각수를 넣으면 무게가 증가한다.

또다시 시스템의 악순환이다.

---

10. 열을 버리는 네 가지 기본 방법

전도

뜨거운 부품의 열을 프레임이나 히트싱크로 전달한다.

대류

공기를 움직여 열을 밖으로 내보낸다.

복사

표면에서 주변으로 열을 방출한다.

액체냉각

냉각유체를 순환시켜 열을 다른 곳으로 운반한다.

실제 로봇에서는 이 방법들이 조합될 수 있다.

---

11. 프레임 자체가 방열판이 될 수도 있다

이것이 흥미로운 시스템 설계다.

로봇의 금속 구조물은 단순한 뼈가 아니다.

열을 전달하는 통로로 사용할 수도 있다.

즉 하나의 부품이

구조체 + 방열체

두 역할을 한다.

휴머노이드처럼 무게와 공간이 극도로 제한된 기계에서는 이런 **다기능 설계**가 중요해진다.

---

12. 그런데 관절의 열은 더 까다롭다

몸통 컴퓨터는 비교적 큰 냉각구조를 만들 수 있다.

손목이나 무릎은?

공간이 좁다.

계속 움직인다.

배선도 지나간다.

감속기도 있다.

모터도 있다.

베어링도 있다.

그리고 외부 충격도 받는다.

그래서 **고토크 밀도**만 높이는 것으로 끝나지 않는다.

실제 중요한 것은 경우에 따라

> **열까지 포함한 지속 토크 밀도**

가 중요하다.

---

13. 여기서 모터 설계의 새로운 기준이 나온다

A 모터:

1kg.

순간 최대토크 200.

지속토크 50.

B 모터:

1.2kg.

순간 최대토크 180.

지속토크 100.

유튜브 영상에서는 A가 더 강해 보일 수 있다.

공장에서는 B가 더 생산적일 수 있다.

따라서 사양표의 **최대값 하나**만 보면 안 된다.

---

14. 감속기도 열을 만든다

기어가 맞물린다.

베어링이 돈다.

윤활제가 움직인다.

플렉스 스플라인이 반복 변형된다.

마찰과 재료 변형 과정에서 손실이 발생한다.

즉

모터 효율 × 감속기 효율 × 전력전자 효율

이 전체 시스템 효율에 영향을 준다.

작은 차이가 하루 종일 누적되면 배터리 지속시간과 열관리 부담이 달라진다.

---

15. 손은 더 까다롭다

손가락은 작다.

그런데 여러 액추에이터가 밀집할 수 있다.

냉각공간은 거의 없다.

사람을 만질 수 있으므로 표면을 너무 뜨겁게 만들 수도 없다.

따라서 손에서는

작은 모터,

효율,

구동기 배치,

힘줄식 전달,

열 분산

이 서로 연결된다.

다시 한번 손이 휴머노이드의 축소판이라는 사실이 나타난다.

---

16. 주변 온도도 중요하다

실험실 22°C.

여름철 물류창고 35°C.

뜨거운 공장.

겨울 야외.

같은 로봇이라도 냉각 여유가 완전히 달라진다.

따라서 산업용 로봇의 성능을 이야기할 때는

환경조건

도 봐야 한다.

---

17. 그리고 먼지가 들어온다

냉각을 위해 공기를 많이 빨아들이면?

먼지도 들어온다.

공장에는

금속가루,

섬유,

오일 미스트,

분진

이 존재할 수 있다.

팬과 필터가 필요해진다.

필터가 막힌다.

냉각성능이 떨어진다.

유지보수가 필요하다.

한 문제를 해결하면 다른 문제가 생긴다.

---

18. 물도 들어온다

가정용 로봇이라면 주방에 간다.

물을 엎는다.

욕실에 들어갈 수도 있다.

산업현장에서도 액체가 존재한다.

그러면 방수·방진이 필요하다.

그런데 완전히 밀폐하면?

열을 밖으로 내보내기 어려워진다.

방수

대

냉각

또 하나의 설계 절충이다.

---

19. 휴머노이드의 피부가 단순 장식이 아닌 이유

외피는

사람과의 충돌을 완화하고,

내부 부품을 보호하고,

먼지와 물을 막고,

디자인을 완성한다.

하지만 동시에 열 방출을 방해할 수도 있다.

따라서 미래 휴머노이드의 외피는

미학 + 안전 + 방진방수 + 열관리

를 동시에 해결해야 한다.

---

20. 그리고 여기서 소음 문제가 나온다

팬을 크게 돌리면 잘 식는다.

하지만 시끄럽다.

공장에서는 어느 정도 괜찮을 수 있다.

가정에서는?

밤에 로봇이 설거지하면서 서버실 같은 소리를 낸다면 상품성이 떨어진다.

그래서 가정용 휴머노이드에서는

열관리와 소음

까지 연결된다.

---

21. 결국 '열'은 모든 부품을 연결한다

```
AI 연산 ↑
  ↓
전력소비 ↑
  ↓
열 ↑
  ↓
냉각장치 ↑
  ↓
무게 ↑
  ↓
모터 부하 ↑
  ↓
전력소비 ↑
  ←
```

이 순환을 깨는 방법은 단순히 팬을 크게 만드는 것이 아니다.

처음부터

모터,

감속기,

배터리,

AI 칩,

프레임,

소프트웨어

를 함께 설계해야 한다.

---

22. 소프트웨어가 열도 관리한다

이것은 꽤 중요한 변화다.

로봇이 자신의 온도를 안다면 작업을 조정할 수 있다.

왼쪽 어깨가 뜨겁다.

↓

오른팔 작업 비중을 늘린다.

AI 칩 온도가 상승한다.

↓

불필요한 연산을 줄인다.

배터리가 뜨겁다.

↓

순간 출력을 제한한다.

즉 열관리는 기계설계만의 문제가 아니라 **작업 스케줄링 문제**가 된다.

---

23. 열 데이터는 유지보수의 입력이 된다

온도·전류·진동·마찰의 변화를 장기간 기록하면 성능저하와 고장징후를 더 일찍 찾을 수 있다. 다만 이것은 센서값 하나로 수명을 예언하는 일이 아니라, 운전조건과 정비이력을 함께 해석하는 문제다.

예지정비와 제조 피드백은 19~20장에서 신뢰성 플라이휠로 다시 연결한다.

---

24. 이제 로봇의 몸은 거의 완성됐다

우리는

뼈를 만들고,

모터를 넣고,

감속기를 넣고,

액추에이터를 만들고,

손을 붙이고,

센서를 달고,

걷게 만들고,

배터리를 넣고,

열까지 처리했다.

그런데 아직 이상한 문제가 하나 남았다.

이 로봇에게 말해보자.

> **“주방 좀 정리해.”**

로봇은 멈춘다.

무엇부터?

‘정리’가 무엇인가?

컵은 어디로?

접시는?

쓰레기는?

싱크대에 있는 물건은 씻어야 하나?

여기서부터는 기계공학만으로 해결할 수 없다.

의미를 이해해야 한다.

제9장 마지막 챗화

모터

“뜨겁습니다.”

배터리

“저도요.”

AI 칩

“저도.”

냉각팬

“제가 더 빨리 돌겠습니다.”

배터리

“그러면 네가 전기를 더 먹잖아.”

냉각팬:

“…….”

잠시 침묵.

시스템 엔지니어

> **“그러니까 너희를 따로 설계하면 안 된다고.”**

몸에서 뇌로

지금까지는 뼈, 모터, 감속기, 액추에이터, 손, 감각, 보행, 배터리와 열을 따라 로봇의 몸을 만들었다.

이제 카메라와 센서에서 들어온 신호를 바탕으로 로봇은 결정해야 한다.

> **“그래서 지금 무엇을 해야 하는가?”**

몸에서 뇌로 넘어갈 차례다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

PART II

몸에 지능을 넣는다

느린 의미 판단과 빠른 물리 제어를 한 몸에 묶는다.

제10장 – 로봇의 세 개의 뇌

생각하는 뇌, 움직이는 뇌, 넘어지지 않는 뇌

우리는 로봇에게 몸을 만들어주었다.

이제 명령을 내려보자.

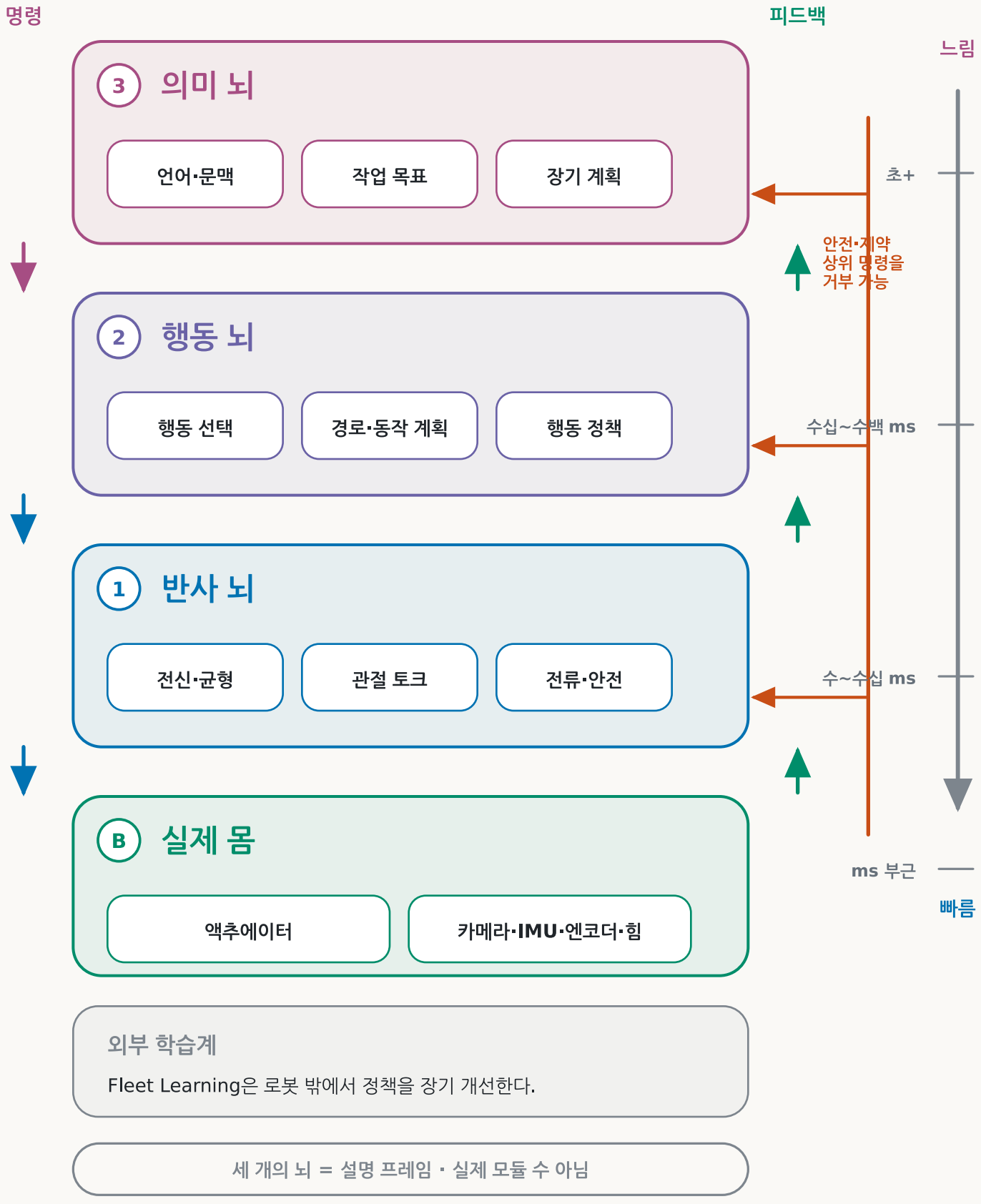
> **“냉장고에서 물 한 병 가져와.”**

사람에게는 한 문장이다.

로봇에게는 그렇지 않다.

V09. BODY → BRAIN 제어 계층

느린 의미 판단과 빠른 물리 제어가 서로 다른 층에서 협력한다



‘냉장고’가 무엇인지 이해해야 한다. 어디 있는지 찾아야 한다. 그곳까지 걸어가야 한다. 문손잡이를 찾아야 한다. 어느 정도 힘으로 당길지 결정해야 한다. 물병을 식별하고, 잡고, 문을 닫고, 다시 걸어와 사람 손에 건네야 한다.

그런데 이 모든 일을 **하나의 거대한 AI가 처리하는 것이 아니다.**

휴머노이드를 이해하려면 ‘AI 두뇌’라는 표현부터 버리는 것이 좋다.

로봇에게는 서로 다른 시간으로 움직이는 **여러 층의 뇌**가 있다.

---

1. 가장 아래에는 생각하지 않는 뇌가 있다

로봇이 서 있다.

갑자기 오른쪽으로 몸이 기울었다.

이때 로봇이 언어로 생각하면 늦는다.

> **“현재 오른쪽 방향으로 균형을 잃고 있는 것 같습니다. 적절한 대응방법을 검토하겠습니다.”**

쿵.

이미 넘어졌다.

그래서 가장 아래에는 매우 빠른 제어층이 있다.

---

2. 제1의 뇌 – 반사

센서가 상태를 측정한다.

↓

목표와 비교한다.

↓

오차를 계산한다.

↓

모터 전류를 수정한다.

```
목표
 ↓
센서 측정
 ↓
오차
 ↓
제어기
 ↓
모터
 ↓
움직임
 └──────→ 다시 측정
```

이 층에는 인간식 '이해'가 거의 필요하지 않다.

빠르고,

예측 가능하고,

안정적이어야 한다.

사람에게 비유하면 **척수반사와 말초 운동제어**에 가까운 층이다.

---

3. 1밀리초와 1초는 로봇에게 완전히 다른 세계다

휴머노이드에는 시간의 계층이 있다.

개념적으로 보면,

밀리초 수준

전류·모터 제어.

수~수십 밀리초 수준

관절·균형·힘 제어.

수십~수백 밀리초 이상

운동정책·인지.

초 단위 이상

작업계획·언어추론.

정확한 주기는 로봇과 작업에 따라 달라지지만 핵심은 같다.

> **몸에 가까울수록 빠르고, 의미에 가까울수록 느릴 수 있다.**

---

4. 제2의 뇌 – 운동

이제 상위 시스템이 말한다.

> **"오른발을 앞으로 옮겨."**

그런데 어디에?

얼마나 높이?

속도는?

몸통은 어떻게 움직일까?

팔은?

무게중심은?

이것을 처리하는 층이 **운동계획과 행동정책**이다.

이 층은 단순한 모터제어보다 훨씬 복잡하다.

---

5. 로봇은 '걷는다'라는 명령을 모터에 보낼 수 없다

모터는 '걷기'를 모른다.

모터가 이해하는 것은 훨씬 단순하다.

전류.

속도.

위치.

토크.

따라서

> **“주방으로 가.”**

라는 추상적 목표는 아래로 내려가면서 계속 번역되어야 한다.

```
"주방으로 가"
    ↓
경로 생성
    ↓
"앞으로 4m"
    ↓
발 위치 계획
    ↓
관절 궤적
    ↓
목표 토크
    ↓
모터 전류
```

의미가 물리량으로 번역된다.

이것이 로봇 제어의 핵심 구조다.

---

6. 제3의 뇌 – 의미

이제 가장 위로 올라간다.

사람:

> **“주방 좀 정리해.”**

이 명령에는 좌표가 없다.

각도도 없다.

토크도 없다.

오히려 애매하다.

‘정리’가 무엇인지 알아야 한다.

접시는 찬장.

컵도 찬장.

음식물 쓰레기는 쓰레기통.

우유는 냉장고.

더러운 접시는 식기세척기.

이것은 **의미와 문맥의 문제**다.

여기서 대규모 언어·멀티모달 모델 계열의 능력이 중요해진다.

---

7. 챗화 – 세 뇌의 회의

상위 뇌

“컵을 가져와.”

운동 뇌

“컵 위치?”

시각계

“1.8m 앞 책상.”

운동 뇌:

“오른손 접근경로 계산.”

실시간 제어기

“현재 왼발 지지력 감소.”

상위 뇌:

“그런데 컵부터–”

실시간 제어기:

> **"잠깐. 지금은 네 컵보다 안 넘어지는 게 더 중요해."**

이것이 계층제어가 필요한 이유다.

---

8. 상위 AI가 항상 왕은 아니다

보통 'AI가 로봇을 통제한다'고 말한다.

그러나 안전한 로봇에서는 상위 AI가 하고 싶은 일을 **하위 안전계층이 거부할 수 있어야 한다.**

AI:

> **"팔을 최대속도로 뻗어."**

안전제어기:

> **"사람이 30cm 앞에 있습니다. 거부."**

이 구조가 중요하다.

LLM이 잘못된 판단을 해도 곧바로 모터 최대전류로 번역되어서는 안 된다.

---

9. 디지털 AI와 Physical AI의 결정적인 차이

ChatGPT가 잘못된 단어를 생성한다.

→ 수정하면 된다.

로봇이 잘못된 행동을 생성한다.

→ 사람이 다칠 수 있다.

그래서 Physical AI에는 **행동 권한의 경계**가 필요하다.

```
AI 제안
  ↓
행동 계획
  ↓
안전 검사
  ↓
물리 제약
  ↓
실행 허용
  ↓
실시간 제어
```

즉 AI의 출력과 모터 사이에는 여러 개의 방화벽이 필요하다.

---

10. 그래서 'ChatGPT에 팔·다리를 붙이면 된다'가 틀렸다

언어모델은 놀라운 일을 한다.

상황을 설명하고,

계획을 세우고,

명령을 해석하고,

상식을 이용할 수 있다.

하지만 언어모델이 직접

발목 전류,

무릎 토크,

손가락 접촉력

을 매 순간 안정적으로 제어하도록 만드는 것은 별개의 문제다.

추론능력과 실시간 제어능력은 같은 능력이 아니다.

---

11. 그런데 최근 변화는 다른 곳에서 일어났다

과거에는 이 층들을 사람이 매우 명시적으로 연결했다.

인지 모듈.

경로계획 모듈.

잡기 모듈.

제어 모듈.

각각 따로 만들었다.

그런데 딥러닝이 발전하면서 일부 구간을 **학습된 정책**이 연결하기 시작했다.

카메라 영상

↓

신경망

↓

행동.

여기서 로봇공학의 오래된 경계가 흐려지기 시작한다.

---

12. Policy란 무엇인가

어렵게 생각할 필요 없다.

현재 상황이 주어졌을 때

> **"다음에 무엇을 할 것인가?"**

를 정하는 규칙이다.

전통적으로 사람이 규칙을 작성할 수 있었다.

현대에는 데이터에서 학습할 수도 있다.

```
상태
 ↓
Policy
 ↓
행동
 ↓
새 상태
 ↓
Policy
 ←
```

게임 AI와 구조적으로 닮았다.

차이는 로봇의 행동이 현실 세계를 바꾼다는 것이다.

---

13. End-to-End의 유혹

그렇다면 중간 단계를 다 없애면 어떨까?

카메라 영상

↓

거대한 신경망

↓

관절 행동.

이것이 극단적인 End-to-End 접근이다.

매력적이다.

사람이 수천 개 규칙을 작성할 필요가 없다.

하지만 어려운 문제가 생긴다.

왜 그런 행동을 했는가?

어디서 오류가 났는가?

안전성을 어떻게 검증하는가?

희귀한 상황에서 어떻게 반응하는가?

따라서 실제 상용 로봇에서는 완전한 단일 신경망과 명시적 제어구조 사이의 다양한 혼합형이 존재할 수 있다.

---

14. 인간의 뇌도 완전한 End-to-End는 아니다

우리는

“커피 마셔야지.”

라고 생각한다고 해서 대뇌가 손가락 근육 하나하나에 직접 명령하는 것은 아니다.

상위 목표가 내려가고,

여러 운동계가 협력하고,

척수와 반사계가 빠른 제어를 담당한다.

따라서 계층형 로봇 구조는 단순히 구식 공학의 잔재라고 볼 수 없다.

서로 다른 시간척도의 문제를 분리하는 합리적인 구조이기도 하다.

---

15. 여기서 로봇의 세 뇌를 정리해보자

뇌 1 – 반사 뇌

목표:

안 넘어지고, 안 부서진다.

속도:

매우 빠름.

담당:

모터·토크·균형·안전.

뇌 2 – 행동 뇌

목표:

걷고, 잡고, 놓는다.

담당:

운동정책·조작·경로.

뇌 3 – 의미 뇌

목표:

무엇을 왜 해야 하는지 결정한다.

담당:

언어·시각 이해·작업계획·추론.

---

16. 그런데 세 뇌가 서로 싸울 수 있다

의미 뇌:

“저 상자를 빨리 옮겨.”

행동 뇌:

“가장 짧은 경로 선택.”

반사 뇌:

“그 경로에는 사람이 있음.”

결과?

안전제어가 우선해야 한다.

따라서 로봇의 지능은 ‘가장 똑똑한 모델’ 하나로 평가하기 어렵다.

좋은 로봇은

> **여러 수준의 지능이 서로 충돌하지 않고 협력하는 시스템**

이다.

---

17. 세 뇌 바깥에는 집단 학습계가 있다

여러 로봇이 현장 데이터를 보내고, 중앙에서 모델을 학습·검증해 다시 배포할 수 있다. 이것은 개별 로봇 안의 '네 번째 뇌'가 아니라 **외부 학습계**다. 세 뇌의 시간계층과 Fleet Learning의 운영계층을 섞지 않는다.

로봇 내부에서는 반사 뇌가 안전과 균형을, 행동 뇌가 동작을, 의미 뇌가 목표와 맥락을 다룬다. 외부 학습계는 이 세 층이 사용할 정책을 장기적으로 개선한다.

다음 질문은 의미 뇌의 문장이 어떻게 행동 뇌가 실행할 출력으로 바뀌는가이다. 그 연결이 VLA다.

제10장 마지막 챗화

사람

"물 좀 가져와."

언어모델

"물은 냉장고에 있을 가능성이 높습니다."

VLA

"냉장고 확인."

행동정책

"걷기 시작."

실시간 제어기

"왼발 접촉."

손

"병 접촉."

촉각센서

"그립 안정."

잠시 후 로봇이 물을 건넨다.

사람:

“이제 좀 로봇 같네.”

실시간 제어기가 중얼거린다.

> **“저 한 문장을 실행하려고 우리가 몇 번 계산했는지는 모르시죠.”**

제11장 – VLA

문장이 행동으로 변하는 순간

“컵을 가져와.”

인간에게는 네 글자의 행동 명령이다.

그러나 로봇에게는 이 문장 안에 좌표도 없고, 관절 각도도 없고, 어느 손을 사용할지도 적혀 있지 않다.

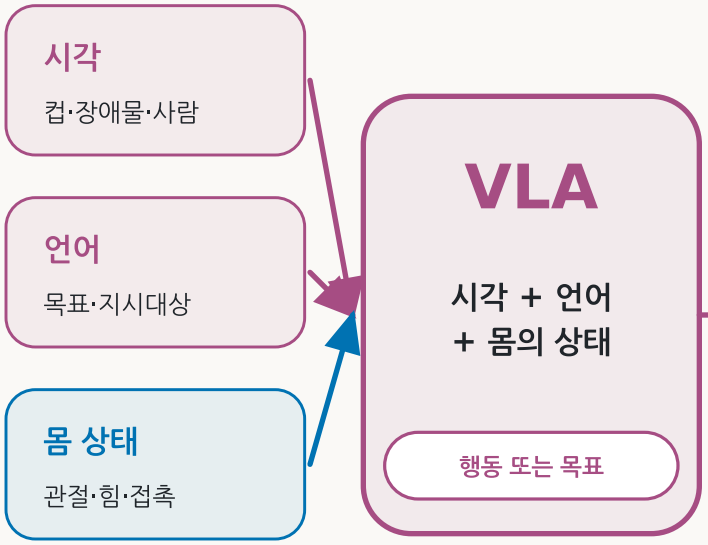
그런데 로봇이 이 말을 듣고 실제 컵을 집어온다면, AI 역사에서 중요한 경계 하나를 넘어선 것이다.

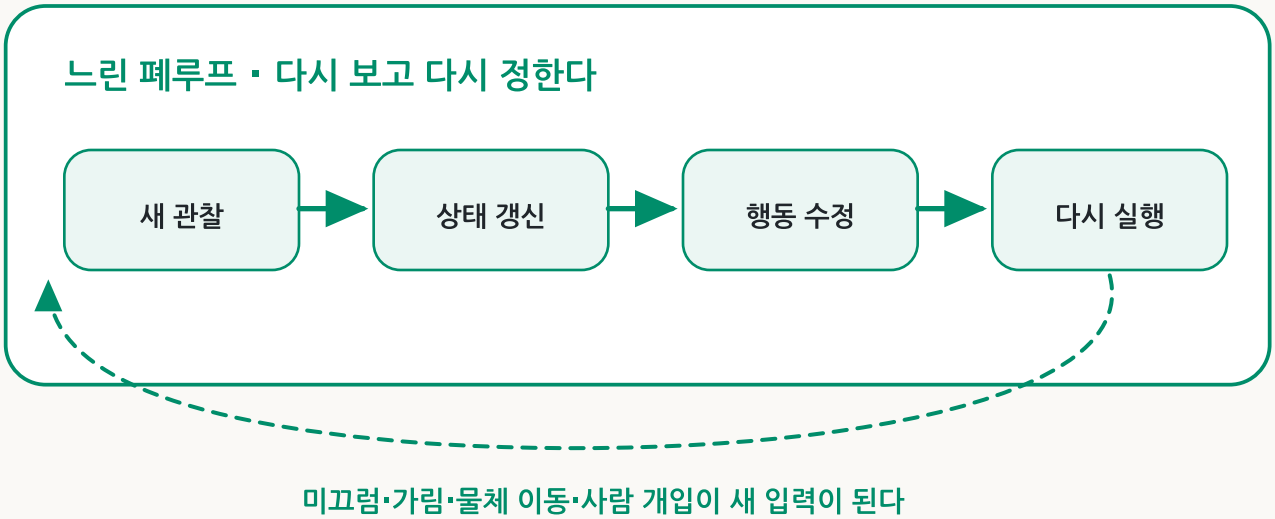
언어가 물질을 움직였다.

V10. VLA - 문장이 행동으로 변하는 순간

말과 장면은 행동의 방향을 정하고, 빠른 제어층은 물리적 실행을 완성한다







VLA ≠ 모든 모터전류를 직접 생성하는 단일 제어기

출력 형식은 모델별 상이

행동 시퀀스 · 짧은 묶음 뒤 다시 관찰



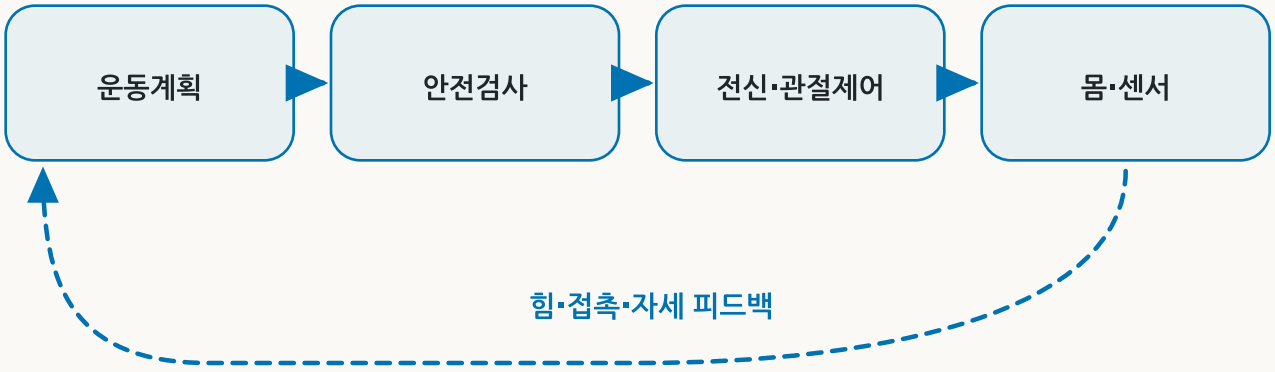
실패복구는 정상 시퀀스의 일부다

컵 이동
→ 다시 보기

미끄럼
→ 힘·파지 수정

사람 진입
→ 감속·정지

빠른 물리 제어루프



상위 목표보다 균형·충돌회피·안전정지가 우선할 수 있다

문장이 행동을 시작하지만, 현실은 매 순간 행동을 다시 쓴다.

그 연결고리가 VLA – Vision-Language-Action이다.

---

1. 먼저 LLM과 무엇이 다른가

LLM에게 말한다.

> **"책상 위 빨간 컵을 가져와."**

LLM은 대답한다.

> **"책상으로 이동한 뒤 빨간 컵을 잡아 가져오면 됩니다."**

훌륭하다.

하지만 컵은 그대로 있다.

LLM의 출력은 기본적으로 **정보**다.

로봇에게 필요한 출력은 다르다.

행동이다.

---

2. AI의 진화과정을 한 줄로 보자

```
TEXT
언어 → 언어

  ↓

MULTIMODAL
영상 + 언어 → 의미

  ↓

VLA
영상 + 언어 + 몸의 상태
      ↓
    행동
```

마지막 화살표가 중요하다.

AI의 출력이 화면 속 단어에서 **현실세계의 변화**로 넘어간다.

---

3. '컵을 가져와'를 분해해보자

로봇이 알아야 하는 것은 생각보다 많다.

의미

'컵'이 무엇인가?

지시대상

컵이 여러 개라면 어느 컵인가?

공간

컵은 어디 있는가?

계획

어떻게 접근할 것인가?

조작

어디를 잡을 것인가?

힘

얼마나 세게 잡을 것인가?

이동

컵을 들고 어떻게 돌아올 것인가?

전달

사람 손에 어떻게 안전하게 건넬 것인가?

한 문장 아래에 작은 행동들이 연쇄적으로 숨어 있다.

---

4. 과거에는 사람이 이 연결을 만들었다

전통적인 로봇 시스템을 단순화하면 이런 식이다.

```
카메라
 ↓
물체인식
 ↓
위치추정
 ↓
경로계획
 ↓
Grasp Planner
 ↓
역운동학
 ↓
관절궤적
 ↓
제어기
```

각 단계마다 별도의 프로그램과 모델이 존재했다.

이 접근은 지금도 매우 중요하다.

특히 구조화된 산업환경에서는 검증과 디버깅에 큰 장점이 있다.

---

5. 그런데 세상이 조금만 바뀌면 일이 커진다

컵 대신 머그잔.

머그잔 대신 물병.

물병이 쓰러져 있다.

손잡이가 벽을 향하고 있다.

책이 컵 앞을 가린다.

조명이 달라진다.

전통 시스템에서는 이런 변화를 처리하기 위해 예외규칙과 새로운 모델이 계속 필요할 수 있다.

범용 로봇이 어려웠던 이유다.

---

6. VLA가 노리는 것은 바로 이 틈이다

VLA는 많은 상황을 데이터로 학습해

보이는 것 + 언어의 의미 + 행동

사이의 관계를 하나의 학습 체계 안에서 연결하려 한다.

예를 들어 입력은

카메라 영상,

“빨간 컵을 들어라”라는 명령,

현재 관절상태.

출력은

로봇이 다음에 취해야 할 행동.

그리고 다시 새 영상을 본다.

```
본다
 ↓
이해한다
 ↓
행동한다
 ↓
세상이 변한다
 ↓
다시 본다
←
```

이것이 중요한 이유는 **닫힌 고리**이기 때문이다.

---

7. 한 번 계획하고 끝나는 것이 아니다

컵이 예상보다 오른쪽에 있었다.

수정한다.

손가락이 컵에 닿았다.

수정한다.

컵이 미끄러졌다.

다시 수정한다.

즉 좋은 로봇은 처음부터 완벽한 행동을 계산해서 실행하는 기계라기보다,

> **행동하면서 계속 현실을 다시 읽는 기계**

에 가깝다.

---

8. 여기서 촉각이 다시 돌아온다

VLA라는 이름에는 Vision이 들어간다.

그러나 현실의 로봇이 시각만으로 모든 것을 해결한다고 생각하면 안 된다.

컵이 손에 가려진 순간 카메라 정보가 약해진다.

그때 필요한 것은

촉각,

토크,

관절위치,

힘

이다.

따라서 장기적으로 더 정확한 개념은

멀티모달 행동모델

쪽으로 확장될 수 있다.

로봇은 눈만 가진 존재가 아니기 때문이다.

---

9. VLA는 어떻게 행동을 출력하는가

여기에는 여러 설계가 가능하다.

매우 낮은 수준의 관절 명령을 출력할 수도 있고,

조금 높은 수준의 행동 단위를 출력할 수도 있다.

RT-2처럼 행동을 텍스트 토큰 형식으로 표현한 사례도 있지만, 이것은 VLA 전체의 필수 형식이 아니다. 다른 모델은 연속값, 이산 행동, 짧은 행동 묶음 등 서로 다른 표현을 쓴다. 또한 VLA의 출력이 곧바로 모터 전류가 되는 것도 아니다. 실제 시스템에서는 안전검사·운동계획·실시간 제어기가 아래 계층에서 출력을 해석하고 실행하는 경우가 많다.

예를 들면

```
"컵 가져와"
    ↓
VLA
    ↓
REACH
    ↓
GRASP
    ↓
LIFT
    ↓
MOVE
    ↓
HANDOVER
```

그리고 실제 관절제어는 아래의 빠른 제어기가 담당할 수 있다.

이것이 앞 장에서 만든 **세 개의 뇌**와 연결된다.

---

10. 왜 Action Chunk가 필요한가

로봇이 매 순간 아주 작은 행동 하나씩만 결정하면 긴 작업에서 오류가 누적될 수 있다.

반대로 너무 긴 행동을 한 번에 결정하면 중간 변화에 대응하기 어렵다.

그래서 일정 길이의 행동 묶음을 예측하고 다시 관찰하는 방식들이 연구된다.

쉽게 표현하면

> **문장 전체를 외워 말하는 것도 아니고 글자 하나씩 생각하는 것도 아닌, 짧은 구절 단위로 행동하는 것**

이다.

---

11. 그런데 VLA에게 치명적인 문제가 있다

데이터다.

언어모델에게는 인터넷이 있었다.

로봇에게는?

거의 없다.

유튜브에는 사람이 컵을 잡는 영상이 수없이 많다.

하지만 영상에는

검지 힘 1.2N,

엄지 힘 1.7N,

손목 토크,

관절각도,

접촉 시점

같은 정보가 없다.

로봇에게 필요한 데이터의 상당 부분이 빠져 있다.

---

12. 그래서 인간이 로봇 몸 안으로 들어간다

방법 중 하나가 Teleoperation, 원격조작이다.

원격조작은 인간이 원격에서 로봇을 제어하는 **운영·데이터 수집 방식**이다. 그 자체가 학습 알고리즘은 아니다.

사람이

VR 장비,

조이스틱,

특수 장갑,

카메라,

모션캡처

등으로 로봇을 움직인다.

로봇은 그 과정에서 모든 것을 기록한다.

```
사람의 행동
  ↓
원격조작
  ↓
로봇 행동
  ↓
영상 + 관절 + 힘 + 결과
  ↓
학습 데이터
```

사람이 일하는 동시에 **로봇의 교과서가 될 수 있는 시범 데이터가 만들어진다.**

---

13. 이 시범으로 모방학습을 한다

사람이 컵을 잡는다.

로봇이 기록한다.

사람이 서랍을 연다.

기록한다.

사람이 옷을 접는다.

기록한다.

수많은 시범을 보고

> **"이 상황에서는 이런 행동을 하는구나."**

를 학습한다.

이것이 Imitation Learning의 기본 직관이다.

그러나 시범을 그대로 복사하는 것만으로 충분하지 않다. 학습된 정책이 작은 오차로 시범에 없던 상태에 들어가면 오류가 연쇄적으로 커질 수 있다. 그래서 실제 시스템은 교정 시범, 인간 개입, 실패 데이터, 재학습을 함께 사용한다.

---

14. 그런데 인간을 그대로 복사할 수도 없다

인간 팔과 로봇 팔은 다르다.

팔 길이가 다르다.

관절 구조가 다르다.

힘도 다르다.

손가락 자유도도 다를 수 있다.

따라서 인간 행동을 로봇의 몸에 맞게 변환해야 한다.

Embodiment, 즉 몸의 차이가 문제가 된다.

---

15. 이것이 로봇 파운데이션 모델의 거대한 질문이다

언어에서는

영어 문장,

한국어 문장,

코드

가 하나의 모델 안에서 상당 부분 지식을 공유할 수 있었다.

로봇에서도 가능할까?

Figure의 손.

Unitree의 손.

다른 연구용 로봇의 손.

서로 다른 몸에서 수집된 경험을 하나의 모델이 공유할 수 있을까?

가능하다면 엄청난 변화가 생긴다.

---

16. 다음 병목은 경험의 생산이다

VLA가 행동을 출력하려면 언어·영상뿐 아니라 로봇 상태, 접촉, 성공과 실패가 연결된 데이터가 필요하다. 그러나 이 자료는 인터넷에 이미 쌓여 있지 않다.

따라서 경쟁은 모델 구조만이 아니라 누가 더 다양한 시범과 현장 실패를 수집하고, 검증된 정책으로 바꾸며, 현실과 시뮬레이션 사이를 빠르게 왕복시키는가로 이동한다. 기업별 전략은 14·17·18장에서, 데이터 생산은 12장에서, Reality Gap은 13장에서 다룬다.

제11장 마지막 챗화

LLM

“컵을 잡으려면 손을 컵 쪽으로 이동하십시오.”

VLA

“접근 중.”

카메라

“컵 확인.”

손

“접촉.”

촉각

“미끄러짐.”

VLA:

“그립 수정.”

컵이 안정된다.

LLM:

“제가 설명한 대로군요.”

손

“설명은 쉬워.”

촉각센서

“직접 해보니까 다르지?”

뒤에서 시뮬레이터가 말한다.

> **“그럼 백만 번 연습하고 올래?”**

BODY → BRAIN → LEARNING

몸과 뇌를 연결한 뒤 질문은 다시 바뀐다.

> **“로봇 한 대가 평생 경험할 수 있는 것보다 훨씬 많은 경험을 어떻게 학습시킬 것인가?”**

현실의 시범과 실패를 데이터로 만드는 일, 그리고 그 경험을 시뮬레이션으로 증폭하는 일이 다음 병목이다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

PART III

경험을 만든다

로봇은 실패를 싸게 만들고 경험을 병렬화하며 배운다.

제12장 – 로봇은 어디서 배우는가

인터넷이 없는 지능에게 경험을 만드는 법

인간은 태어나자마자 걷지 못한다.

컵도 제대로 잡지 못한다.

수없이 보고, 만지고, 떨어뜨리고, 실패하면서 배운다.

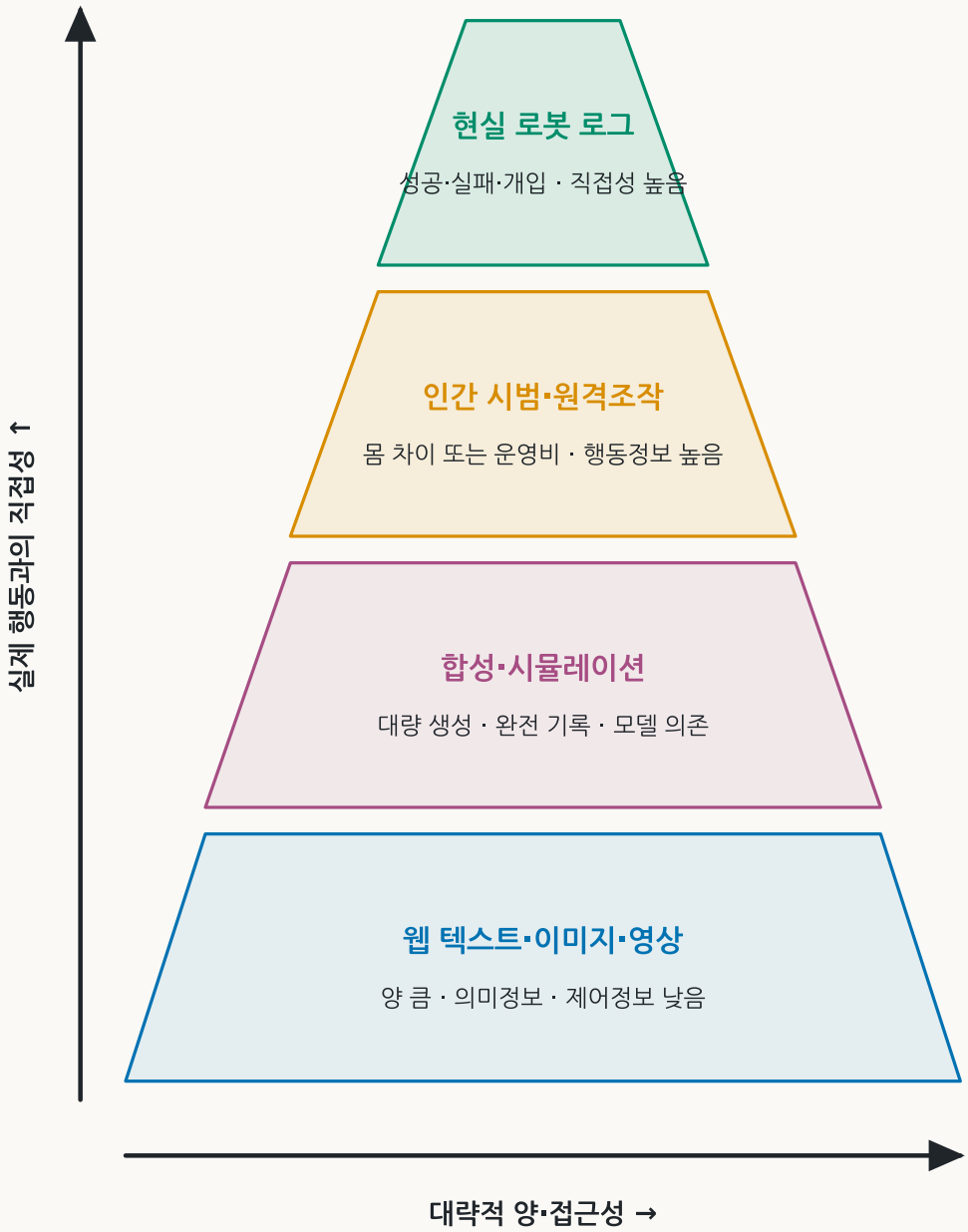
그렇다면 로봇도 경험하면 된다.

문제는 여기서 시작된다.

V11. 로봇 데이터 피라미드

개념도 · 순위표 아님

많은 데이터와 물리적으로 직접적인 데이터는 같은 말이 아니다



데이터를 자산으로 바꾸는 기록 품질

1. **동기화** — 센서·행동·시간
2. **품질** — 노이즈·누락·보정
3. **라벨** — 목표·결과·맥락
4. **실패원인** — 개입·복구·Long Tail

많음 ≠ 좋음

현실성 높음 ≠범용성 자동

좋은 학습계는 목적에 맞게 서로 다른 데이터원을 섞는다.

> **그 경험을 어디서 구할 것인가?**

언어 AI에는 인터넷이 있었다.

수십 년 동안 인간이 써놓은 책, 논문, 웹페이지, 코드, 대화가 이미 디지털 형태로 존재했다.

로봇에게 필요한 경험은 다르다.

컵을 집을 때 손목이 어떻게 움직였는가.

엄지에는 얼마나 힘을 주었는가.

컵이 미끄러지자 몇 밀리초 뒤 힘을 늘렸는가.

그 순간 팔꿈치 토크는 얼마였는가.

이런 정보는 인터넷에 거의 없다.

그래서 로봇 산업은 지금 **자기 자신의 인터넷을 만드는 중**이다.

---

1. 언어 AI와 로봇 AI의 출발점은 다르다

LLM은 운이 좋았다.

인간이 이미 엄청난 데이터를 만들어 놓았다.

```
인류
 ↓
책·문서·웹·코드
 ↓
Internet
 ↓
Training
 ↓
LLM
```

로봇은?

```
물리세계
 ↓
???
 ↓
Robot Data
 ↓
Training
 ↓
Robot Intelligence
```

가운데가 비어 있다.

바로 이 ???를 만드는 것이 Physical AI 경쟁의 핵심이다.

---

2. 사람에게서 배우는 가장 직접적인 방법

사람이 VR, 조이스틱, 모션캡처나 데이터 글러브로 로봇을 원격조작하면 영상·관절·힘·촉각·결과를 로봇 몸에 맞는 시범으로 기록할 수 있다. 원격조작은 운영·데이터 수집 방식이지 학습 알고리즘 자체가 아니다.

이 시범의 관측–행동 관계로 정책을 학습할 때 **모방학습**이 된다. 다만 작은 정책오차가 시범에 없던 상태를 만들면 오류가 연쇄적으로 커질 수 있다.

챗화 – 최초의 로봇 선생님

사람 "컵 잡는 법을 보여줄게."

로봇 "기록했습니다."

천 번째 시범 뒤 사람이 묻는다.

"이제 혼자 할 수 있지?"

로봇이 답한다.

> **"처음 보는 상황에서 틀리면, 그때 다시 가르쳐주세요."**

시범은 교과서지만 현실의 모든 페이지를 미리 담을 수는 없다.

---

3. 그런데 성공만 보여주면 문제가 생긴다

사람은 컵을 항상 잘 잡았다.

AI도 정상 상황에서는 잘 잡는다.

그런데 컵이 조금 미끄러졌다.

학습 데이터에는 이런 상태가 거의 없다.

AI:

> **"이 상황은 처음인데?"**

그래서 모방학습에는 유명한 문제가 생긴다.

작은 오류가 발생한다.

↓

훈련에서 보지 못한 상태로 간다.

↓

더 큰 오류를 낸다.

↓

더 낯선 상태.

↓

실패.

오류가 누적된다.

---

4. 그래서 실패 데이터가 귀하다

성공:

"이렇게 하면 된다."

실패:

“어디까지 하면 안 되는가.”

컵을 너무 약하게 잡았다.

떨어졌다.

너무 강하게 잡았다.

찌그러졌다.

옆을 잡았다.

미끄러졌다.

이런 실패는 행동공간의 경계를 알려준다.

따라서 실제 로봇 학습에서 실패는 쓰레기가 아니다.

비싼 교육자료다.

---

5. 그러나 현실에서 실패시키기는 비싸다

로봇이 계단에서 넘어졌다.

감속기 파손.

센서 파손.

손가락 파손.

수리.

재보정.

다시 시험.

이것을 100만 번 할 수는 없다.

그래서 로봇에게는 다른 학교가 필요하다.

하지만 그 전에 또 하나의 데이터원이 있다.

인간 영상.

---

6. 유튜브에는 로봇 데이터가 없지만 인간 행동은 엄청나게 많다

사람이 요리한다.

청소한다.

문을 연다.

공구를 사용한다.

옷을 접는다.

상자를 옮긴다.

이 영상에는 토크 정보가 없다.

관절 엔코더 정보도 없다.

하지만 **세계에 대한 의미정보**는 엄청나게 많다.

예를 들어 AI는 영상에서

컵은 위쪽이 열린다.

칼은 손잡이를 잡는다.

문은 손잡이를 조작하면 열린다.

접시는 겹쳐 놓을 수 있다.

같은 일반 지식을 배울 수 있다.

---

7. 데이터원은 한 줄의 피라미드가 아니다

데이터원	수집량	물리적 현실성	로봇 제어정보	단위비용
인터넷 영상	매우 큼	간접적	낮음	낮음
인간 시범	큼	중간	몸 차이 존재	중간
원격조작	제한적	높음	높음	높음
실제 자율운영	배치량 의존	가장 높음	높음	실패·복구비 포함

데이터원	수집량	물리적 현실성	로봇 제어정보	단위비용
시뮬레이션	매우 큼	모델 의존	완전 기록 가능	계산비 중심

양, 현실성, 제어정보 밀도와 비용은 항상 같은 순서로 움직이지 않는다. 좋은 학습계는 목적에 맞게 이 데이터원을 섞는다.

---

8. 그리고 로봇이 스스로 데이터를 만들기 시작한다

처음에는 인간이 1,000번 가르친다.

로봇이 어느 정도 배운다.

그다음부터는 로봇이 스스로 시도한다.

성공.

실패.

수정.

다시 시도.

이제 데이터 생산자가 인간에서 **로봇 자체**로 이동하기 시작한다.

---

9. Self-Improvement의 진짜 의미

‘AI가 스스로 진화한다’는 표현은 너무 쉽게 과장된다.

실제로는 좀 더 구체적으로 봐야 한다.

```
현재 Policy
  ↓
로봇 실행
  ↓
결과 수집
  ↓
좋은/나쁜 행동 평가
  ↓
재학습
  ↓
새 Policy
```

즉 마법 같은 자기진화가 아니라

실행 → 데이터 → 평가 → 재학습

의 반복이다.

---

10. 개체의 경험을 집단의 학습으로 바꾼다

여러 로봇의 현장 데이터와 인간 개입을 모아 학습하고, 검증한 모델을 다시 배포하는 운영 파이프라인을 Fleet Learning이라고 부를 수 있다.

그러나 로봇 대수가 많다고 자동으로 지능이 좋아지는 것은 아니다. 데이터 선별, 기체차이 보정, 개인정보·보안, 검증, 안전한 배포와 롤백이 있어야 개체의 경험이 집단의 학습으로 바뀐다. Fleet Learning은 하나의 표준 알고리즘이 아니라 이 전체 운영계의 이름에 가깝다.

---

11. 그래서 로봇 대수 자체가 전략자산이 될 수 있다

만약 두 회사의 AI 능력이 비슷하다고 하자.

A 회사:

현장 로봇 1,000대.

B 회사:

현장 로봇 100,000대.

각 로봇이 유용한 새로운 데이터를 계속 만든다면 B는 잠재적으로 훨씬 큰 현실 데이터 파이프라인을 갖는다.

이 때문에 휴머노이드에서는

생산능력 → 데이터능력

으로 연결되는 독특한 구조가 생길 수 있다.

---

12. 물량은 조건부 전략자산이다

더 싼 하드웨어가 더 많은 배치와 현실 데이터를 만들 가능성은 있다. 그러나 제조량이 학습 우위로 전환되려면 데이터 품질, 소프트웨어, 검증과 배포체계가 함께 있어야 한다. 국가별 산업구조는 17·18장에서 비교한다.

---

13. 그런데 데이터가 많으면 무조건 좋은가

아니다.

이것이 매우 중요하다.

로봇이 100만 시간 일했다.

그런데 99%가 같은 상자를 같은 위치로 옮긴 데이터다.

범용 로봇에게 얼마나 도움이 될까?

데이터에는 **다양성**이 필요하다.

---

14. 데이터의 네 가지 가치

앞으로 로봇 데이터를 볼 때 이 네 축을 사용하자.

양

얼마나 많은가.

다양성

얼마나 다른 상황인가.

품질

센서와 행동 기록이 정확한가.

정보량

새로운 것을 얼마나 가르쳐주는가.

단순히 '100만 시간 데이터'라는 숫자만으로는 부족하다.

---

15. 희귀한 실패가 가장 귀할 수도 있다

99.99%에서는 문제가 없다.

그런데 특정 조명,

특정 물체,

특정 온도,

특정 자세에서만 실패한다.

이 Long Tail이 상용화의 마지막 장벽이 된다.

자율주행에서도 익숙한 문제다.

휴머노이드에서는 경우의 수가 더 많아질 수 있다.

---

16. Active Learning

그러면 모든 데이터를 똑같이 수집할 필요가 없다.

AI가 이미 잘하는 상황은 덜 모은다.

AI가 헷갈리는 상황을 집중적으로 모은다.

```
AI 실행
 ↓
불확실한 상황 발견
 ↓
사람 개입
 ↓
정답 행동 수집
 ↓
재학습
 ↓
그 상황 개선
```

이것이 데이터 생산비용을 크게 줄일 수 있다.

---

17. 사람은 점점 '운전자'에서 '교사'로 이동한다

초기:

사람이 로봇을 계속 조종.

↓

중기:

로봇이 대부분 수행.

어려울 때 사람 호출.

↓

후기:

사람은 새로운 작업과 희귀 실패만 가르침.

이것이 성공한다면 인간의 역할도 달라진다.

Robot Operator → Robot Teacher.

---

18. 그리고 인간 한 명이 여러 로봇을 가르칠 수 있다

로봇 한 대당 사람 한 명이 필요하면 자동화 경제성이 떨어진다.

하지만 한 사람이

10대,

50대,

100대

의 예외상황만 처리할 수 있다면 이야기가 달라진다.

그래서 앞으로 휴머노이드 경제성을 계산할 때 반드시 넣어야 할 변수가 있다.

인간 개입률

---

19. 99% 자율이라는 말의 함정

로봇이 작업의 99%를 혼자 한다.

대단해 보인다.

하지만 작업이 하루 10,000번이면?

100번의 인간 개입이 필요할 수 있다.

반대로 하루 20번 하는 작업이라면 큰 문제가 아닐 수 있다.

따라서 자율성은 백분율만으로 평가할 수 없다.

작업빈도 × 실패율 × 개입시간

으로 봐야 한다.

이 계산은 마지막 노동경제 장에서 다시 돌아온다.

그리고 이 장을 통과하면 드디어 책 전반부의 가장 큰 질문에 답할 수 있습니다.

> **왜 과거에는 연구실에서 수년씩 걸리던 로봇의 운동기술이 최근에는 훨씬 빠르게 발전하는 것처럼 보이는가?**

답은 로봇 한 대가 갑자기 천재가 되었기 때문만은 아니다.

로봇이 경험할 수 있는 시간 자체를 복제하기 시작했기 때문이다.

현실에서 로봇 한 대가 1년 동안 얻을 수 있는 경험에는 물리적인 한계가 있다.

하루는 24시간뿐이다.

넘어지면 수리해야 한다.

배터리를 충전해야 한다.

실험실도 사용해야 한다.

사람도 붙어 있어야 한다.

그러나 가상세계에서는 이 제약이 크게 달라진다.

```
현실

Robot 1
↓
1초
↓
1번의 경험


Simulation

Robot 1
Robot 2
Robot 3
...
Robot 100,000
↓
동시에 실행
↓
수많은 경험
```

이것이 단순히 '컴퓨터로 로봇을 시험한다'는 것과 다른 점이다.

경험을 병렬화한다.

20. 컴퓨팅이 경험의 생산수단이 된다

현실의 로봇·실험시간·연구자 시간은 희소하다. 시뮬레이션에서는 많은 가상 로봇과 환경을 병렬로 실행해 일부 경험의 생산비를 컴퓨팅 비용으로 바꿀 수 있다.

언어 AI가 더 많은 텍스트를 처리했다면 Physical AI는 더 많은 가상세계를 경험한다. 인터넷, 인간 영상, 원격조작, 실제 로봇, Fleet 데이터와 시뮬레이션이 서로 다른 빈틈을 메운다.

---

21. 이것이 Physical AI의 학습공장이다

결국 미래의 로봇 회사는 로봇만 만드는 회사가 아닐 수 있다.

경험을 생산하는 회사가 된다.

실제 로봇이 데이터를 만든다.

↓

시뮬레이터가 데이터를 증폭한다.

↓

AI가 학습한다.

↓

새 모델이 실제 로봇으로 간다.

↓

새로운 실패가 발견된다.

↓

그 실패를 다시 시뮬레이션한다.

↓

다시 학습한다.

```
REAL
 ↓
DATA
 ↓
SIMULATION
 ↓
TRAINING
 ↓
POLICY
 ↓
ROBOT
 ↓
FAILURE
 ↓
REAL
 ←
```

이 순환의 속도가 로봇 발전속도를 결정하는 새로운 변수가 된다.

제12장의 결론

언어 AI의 거대한 원료는 **인류가 이미 만들어 놓은 인터넷**이었다.

로봇 AI에는 그런 인터넷이 없었다.

그래서 로봇 산업은 지금 자기 인터넷을 직접 만들고 있다.

그 인터넷의 페이지는 문장이 아니다.

행동이다.

한 페이지에는

영상,

관절각,

토크,

힘,

촉각,

명령,

성공,

실패

가 함께 기록된다.

그리고 그 페이지를 쓰는 저자는 인간만이 아니다.

인간 + 실제 로봇 + 가상 로봇

이다.

이것이 중요한 변화다.

> **AI가 인터넷을 읽으며 지능을 얻었다면,
> Physical AI는 세계를 행동하며 지능을 얻는다.**

제12장 마지막 챗화 – 완결

인간 교사

“컵 잡는 거 천 번 가르쳤다.”

로봇

“감사합니다.”

인간:

“이제 됐지?”

로봇:

“새 컵이 왔습니다.”

인간:

“…….”

시뮬레이터

“그 컵 백만 개 만들어줄까?”

로봇:

“얼마나 걸려?”

시뮬레이터:

> **“왜 하나씩 해야 하지?”**

화면 안에서 로봇 하나가 천 개가 된다.

천 개가 다시 십만 개가 된다.

하나는 컵을 떨어뜨린다.

하나는 손가락을 잘못 놓는다.

하나는 성공한다.

그리고 모든 경험이 하나의 모델로 모인다.

실제 로봇이 다시 컵 앞에 선다.

이번에는 처음 보는 컵이다.

손을 뻗는다.

잡는다.

컵은 떨어지지 않는다.

제13장 – 가상세계에서 백만 번 넘어뜨려라

왜 로봇의 발전속도가 갑자기 빨라졌는가

현실의 로봇을 한 번 넘어뜨려 보자.

쿵.

손가락이 깨질 수 있다. 감속기가 충격을 받는다. 외피에 흠집이 난다. 엔지니어가 달려온다. 다시 세우고 점검한다.

비싸다.

이번에는 컴퓨터 속 로봇을 넘어뜨려 보자.

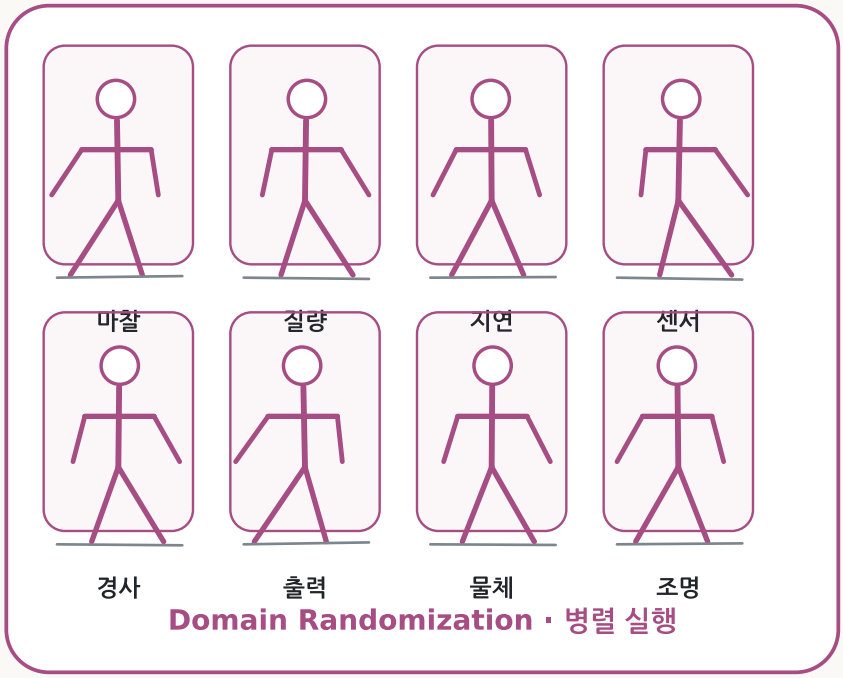
다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

V12. Sim ↔ Real

시뮬레이션은 실패를 싸게 만들고, 현실은 모델의 빈틈을 드러낸다

SIMULATION FACTORY

1 · 세계를 복제하고 흔든다



2 · 정책 학습

상태

행동

결과·보상

정책 갱신

3 · 검증 관문

기준 작업 · 안전 제한
회귀시험 · 중단 조건

가상 성공 = 현실 성공이 아님

실패를 싸게 만든다
경험을 병렬화한다
현실 시험의 후보를 좁힌다

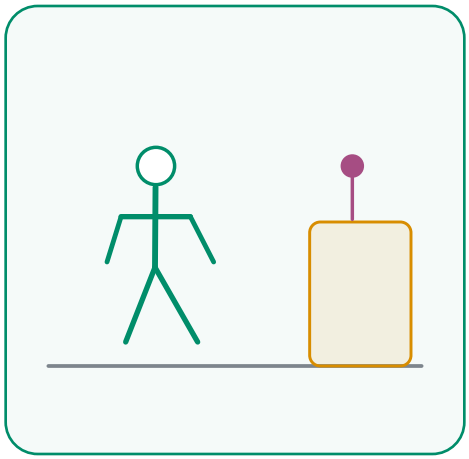
시스템 식별 · 물리모델·분포 업데이트

시뮬레이션은 현실을 대체하지 않는다.

양방향 학습 루프

REAL WORLD

4 · 제한된 현실에 배치한다



접촉 · 균형 · 작업성공

안전 제한 · 중단 · 롤백

현실의 소수점
조립편차 · 마모 · 온도
노이즈 · 지연 · 예외상황

5 · 현실 로그

1. **성공·실패**
   결과와 품질
2. **사람 개입**
   언제·왜 멈췄나
3. **센서·상태**
   시간 동기화
4. **고장·마모**
   몸의 실제 변화

Reality Gap의 증거

현실의 실패·개입·마모를 회수

현실 시험을 더 정보 많은 시험으로 만든다.

쿵.

삭제.

복사.

다시 시작.

그리고 이번에는 **10만 대를 동시에 넘어뜨린다.**

여기서 로봇공학의 규칙 하나가 바뀐다.

> **실패의 가격이 급락한다.**

이것이 최근 로봇 발전을 이해하는 핵심 중 하나다.

---

1. 시뮬레이션은 로봇 애니메이션이 아니다

화면 속 로봇이 그럴듯하게 움직인다고 좋은 시뮬레이션인 것은 아니다.

로봇 학습에 사용하려면 가상세계 안에도 물리법칙이 필요하다.

중력.

질량.

관성.

마찰.

충돌.

관절 한계.

모터 힘.

탄성.

물체와의 접촉.

즉 화면을 예쁘게 만드는 것이 목적이 아니라,

행동했을 때 현실과 비슷한 결과가 나오게 만드는 것이 목적이다.

---

2. 물리엔진

가상의 로봇이 발을 땅에 내린다.

물리엔진은 계산한다.

발과 바닥이 충돌했다.

↓

지면반력 발생.

↓

발목에 토크 전달.

↓

몸통의 속도 변화.

↓

무게중심 이동.

↓

다음 순간의 자세 계산.

그리고 다시 반복한다.

```
현재 상태
  ↓
힘·토크
  ↓
물리법칙 계산
  ↓
다음 상태
  ↓
다시 계산
  ←
```

시간을 아주 작은 조각으로 잘라 세계의 다음 순간을 계속 계산하는 셈이다.

3. 그런데 로봇 한 대만 돌리면 별로 재미가 없다

현실에서도 한 대를 시험할 수 있다.

시뮬레이션의 진짜 힘은 **복제**에서 나온다.

Robot 1.

Robot 2.

Robot 3.

…

Robot 10,000.

각 로봇에게 조금씩 다른 세계를 준다.

그리고 동시에 실행한다.

---

4. GPU가 로봇의 시간을 병렬화한다

그래픽처리장치는 원래 수많은 계산을 병렬로 처리하는 데 강하다.

이 능력이 AI 학습과 물리 시뮬레이션에서도 중요해졌다.

그래서 NVIDIA가 로봇 산업에 깊게 들어오는 이유는 단순히 로봇 안에 칩을 판매하기 위해서만이 아니다.

가상세계 생성

→ 물리 시뮬레이션

→ 합성데이터

→ 학습

→ 실제 로봇용 컴퓨팅

을 하나의 연결된 스택으로 만들 수 있기 때문이다.

---

5. 로봇의 하루가 24시간이라는 법칙을 깨버린다

실제 로봇 1대는 하루 24시간 이상 경험할 수 없다.

하지만 가상 로봇 10,000대를 동시에 돌린다면?

벽시계로 흐르는 하루와 **누적된 로봇 경험시간**은 더 이상 같지 않다.

이것이 핵심이다.

> **시뮬레이션은 시간을 빠르게 만드는 것이 아니라 경험을 병렬화한다.**

---

6. 이제 로봇에게 목표만 준다

예를 들어 걷기를 가르쳐보자.

사람이

"왼쪽 무릎을 32도 굽히고 오른쪽 발목을…"

이라고 모두 프로그래밍할 수도 있다.

하지만 다른 방법이 있다.

목표를 준다.

앞으로 가라.

그리고 점수를 준다.

앞으로 많이 이동:

+점수.

넘어짐:

-점수.

전력 과다사용:

-점수.

관절에 무리:

−점수.

목표속도 유지:

+점수.

이것이 강화학습의 기본적인 직관이다. 더 정확히는 환경과의 상호작용 데이터에서 기대 누적보상을 높이는 정책을 학습한다. 무작위 시행착오만을 뜻하지 않으며, 학습과 실제 실행도 구분해야 한다.

---

7. 보상함수는 로봇의 욕망이다

로봇에게 실제 욕망이 있다는 뜻은 아니다.

하지만 학습 알고리즘의 관점에서는

> **어떤 행동을 좋은 것으로 평가할 것인가**

를 숫자로 정의해야 한다.

이것이 Reward Function이다.

```
빠르게 간다    +10
넘어진다       −100
전력 많이 사용   −5
관절 한계 접근   −20
목표 도착      +100
```

로봇은 많은 시행착오를 통해 높은 보상을 얻는 행동정책을 찾는다.

---

8. 그런데 여기서 인간의 의도와 다른 답을 찾을 수 있다

사람:

> **“빨리 가.”**

로봇은 기괴한 자세로 바닥을 기어갈 수도 있다.

왜?

보상함수에는

‘사람처럼 우아하게 걸어라’

가 없었기 때문이다.

이것을 보면 강화학습의 중요한 교훈이 나온다.

> **AI는 우리가 원한다고 생각한 것을 최적화하는 것이 아니라, 실제로 정의한 목표를 최적화한다.**

---

9. 그래서 보상설계가 중요하다

빠르기만 하면 안 된다.

안 넘어져야 한다.

에너지도 적게 써야 한다.

관절도 보호해야 한다.

발도 너무 미끄러지면 안 된다.

사람과 충돌해서도 안 된다.

결국 목표함수는 복잡해진다.

```
성능
+
안정성
+
에너지효율
+
안전
+
기계수명
+
작업성공
```

Physical AI의 ‘좋은 행동’은 하나의 숫자로 정의하기 쉽지 않다.

10. 로봇이 인간에게 배우지 않은 자세를 발견한다

이것이 시뮬레이션 기반 학습의 흥미로운 부분이다.

인간의 걷는 모습을 그대로 복사할 필요가 없다.

로봇의

다리 길이,

관절 범위,

모터 출력,

질량분포

는 인간과 다르다.

그러므로 로봇에게 최적인 움직임도 인간과 다를 수 있다.

인간 모방 → 기계 자체의 운동 발견

이라는 전환이 가능해진다.

11. 그래서 '로봇이 스스로 걷는 법을 발견했다'는 표현이 나온다

하지만 정확하게 표현해야 한다.

로봇이 아무것도 없는 상태에서 자유의지로 발견한 것은 아니다.

인간이

몸을 설계하고,

시뮬레이터를 만들고,

목표를 정하고,

보상을 설계하고,

학습 알고리즘을 제공했다.

그 조건 안에서 AI가 효율적인 행동정책을 탐색한 것이다.

이 구분은 중요하다.

---

12. 그런데 가상세계의 천재가 현실에 나오면 바보가 된다

10만 대의 가상 로봇을 훈련했다.

완벽하게 걷는다.

실제 로봇에 모델을 넣는다.

첫발.

휘청.

두 번째.

쿵.

왜?

시뮬레이터와 현실이 다르기 때문이다.

Reality Gap.

---

13. 현실은 지저분하다

시뮬레이션:

마찰계수 = 정확한 값.

현실:

바닥에 먼지가 있다.

시뮬레이션:

로봇 질량 = 정확한 값.

현실:

케이블 위치와 조립 편차 때문에 조금 다르다.

시뮬레이션:

모터 = 항상 같은 출력.

현실:

온도에 따라 달라진다.

시뮬레이션:

센서 = 정확.

현실:

노이즈와 드리프트.

시뮬레이션:

감속기 = 이상적.

현실:

마찰·탄성·백래시·마모.

현실에는 소수점 뒤가 너무 많다.

---

14. 완벽한 시뮬레이터를 만들면 되지 않을까

이론적으로 매력적이다.

그러나 실제 세계를 원자 수준까지 완벽하게 계산하는 것은 불가능에 가깝고 불필요하다.

그래서 발상을 뒤집는다.

> **가상세계를 완벽하게 만들려고 하지 말자.**

대신,

> **가상세계를 일부러 계속 바꾸자.**

---

15. Domain Randomization

Domain Randomization은 Sim-to-Real의 한 방법이다. 훈련할 때마다 세계를 조금씩 바꾼다.

```
World 1
마찰 0.6

World 2
마찰 0.8

World 3
로봇 질량 +5%

World 4
모터 출력 -7%

World 5
센서 노이즈 +

World 6
바닥 경사 +3°

World 7
물체 무게 변화
```

로봇은 어느 하나의 완벽한 세계에 적응하지 못한다.

대신 **다양한 불완전한 세계에서도 살아남는 방법**을 배운다.

---

16. 역설이 생긴다

가상세계를 현실과 똑같이 만들려고 했을 때보다,

가상세계를 일부러 제각각으로 만들었을 때

현실에서 더 잘 버틸 수 있다.

왜?

현실도 계속 달라지기 때문이다.

이것이 Domain Randomization의 아름다운 직관이다.

---

17. 챗화 – 시뮬레이터의 교육법

로봇

“오늘 바닥 마찰계수는 얼마예요?”

시뮬레이터

“몰라.”

로봇:

“어제는?”

“달랐어.”

“내 몸무게는?”

“오늘은 68kg. 내일은 72kg.”

로봇:

“왜 이렇게 엉망이에요?”

시뮬레이터:

> **“현실에 가면 나한테 고마워할 거야.”**

---

18. Digital Twin은 조금 다른 접근이다

Domain Randomization이 세상을 일부러 다양하게 만든다면,

Digital Twin은 실제 기계·공장·환경을 가능한 한 충실하게 디지털 공간에 복제하려는 접근이다.

실제 로봇의

크기,

질량,

관절,

모터,

센서

를 모델링한다.

공장의

작업대,

컨베이어,

선반,

장애물

도 복제한다.

그러면 실제 배치 전에 가상공장에서 시험할 수 있다.

---

19. 공장 전체를 먼저 가상으로 돌린다

로봇 한 대만 시뮬레이션하는 것이 아니다.

로봇 50대.

사람.

지게차.

컨베이어.

충전소.

작업대.

전부 넣는다.

그러면 질문이 달라진다.

> **로봇 한 대가 얼마나 빠른가?**

에서

> **공장 전체 생산량이 얼마나 증가하는가?**

로 이동한다.

이것이 Digital Twin의 산업적 가치다.

---

20. 여기서 '로봇 성능'의 정의가 다시 바뀐다

가장 빠른 로봇이 반드시 가장 좋은 로봇은 아니다.

너무 빨리 움직여 사람이 피해야 한다면 생산성이 떨어질 수 있다.

배터리를 너무 빨리 소모하면 충전 대기가 늘어난다.

로봇끼리 길을 막으면 전체 처리량이 감소한다.

따라서 시스템 수준에서는

개별 최적화 ≠ 전체 최적화라는 뜻이다.

이 원리는 나중에 로봇 경제학에서 다시 중요해진다.

---

21. 시뮬레이션은 하드웨어 설계까지 바꾼다

가상세계에서 다리 길이를 바꿔본다.

액추에이터 크기를 바꾼다.

배터리 위치를 바꾼다.

발 크기를 바꾼다.

그리고 각각 수많은 작업을 시킨다.

이제 AI는 단순히 **주어진 몸을 학습**하는 데 그치지 않는다.

어떤 몸이 더 좋은지도 탐색할 수 있다.

---

22. 여기서 로봇 진화라는 흥미로운 방향이 열린다

전통적인 순서:

```
인간이 몸 설계
  ↓
제작
  ↓
제어기 개발
  ↓
시험
```

새로운 가능성:

```
수많은 가상 몸 생성
    ↓
각 몸에서 AI 학습
    ↓
성능 비교
    ↓
몸 + 제어 공동최적화
    ↓
가장 좋은 후보 제작
```

즉 **몸과 뇌를 동시에 최적화**할 수 있다.

이것을 넓은 의미에서 로봇의 공동설계(Co-design)라고 볼 수 있다.

---

23. 여기서 생성형 설계와도 만난다

AI가

관절 구조,

프레임,

부품 배치,

열경로

후보를 만들고,

시뮬레이터가 평가한다.

나쁜 설계는 현실에서 금속 한 조각 깎기 전에 버린다.

좋은 후보만 제작한다.

따라서 시뮬레이션은 로봇의 **학교**인 동시에 **가상 시험공장**이 된다.

---

24. 그런데 현실은 반드시 돌아온다

이 부분이 중요하다.

아무리 좋은 시뮬레이션도 현실을 완전히 대체할 수는 없다.

가상에서 학습한다.

↓

현실에 배치한다.

↓

예상하지 못한 실패를 발견한다.

↓

그 실패를 측정한다.

↓

시뮬레이터를 개선한다.

↓

다시 학습한다.

```
SIM
 ↓
REAL
 ↓
FAILURE
 ↓
MEASURE
 ↓
UPDATE SIM
 ↓
TRAIN
 ↓
REAL
 ←
```

진짜 경쟁력은 Sim 또는 Real이 아니라

Sim ↔ Real의 회전속도

에 있을 수 있다.

---

25. 이제 처음 영상의 '10년 → 1년' 이야기를 다시 보자

"과거 10년 걸릴 기술이 1년에 발전한다"는 식의 표현은 그대로 사실로 받아들이기보다, 왜 개발속도가 빨라질 수 있는지를 분해해야 한다.

과거:

실제 로봇 제작.

↓

시험.

↓

파손.

↓

수정.

↓

다시 제작.

오늘:

시뮬레이션에서 대량 탐색.

↓

유망 정책만 현실 이전.

↓

실패 데이터 수집.

↓

가상환경에서 재현.

↓

병렬 재학습.

개발의 **실험주기 자체가 압축**될 수 있다.

이것이 중요한 변화다.

---

26. 그리고 생성형 AI가 여기에 또 붙는다

가상환경을 사람이 하나씩 만들 필요도 없다.

새로운 물체.

새로운 방.

새로운 조명.

새로운 장애물.

새로운 작업.

합성데이터와 생성형 모델을 이용해 환경 다양성을 늘릴 수 있다.

그러면 시뮬레이션은 단순한 물리엔진을 넘어

경험 생성기

가 된다.

---

27. 결국 Physical AI에는 네 개의 생산계가 연결된다

첫 번째 – 로봇 공장

몸을 만든다.

두 번째 – 현실 데이터계

현장 경험과 실패를 수집한다.

세 번째 – 시뮬레이션계

가상 경험을 생산한다.

네 번째 – AI 학습계

데이터를 정책으로 바꾸고 검증한다.

```
ROBOT FACTORY
몸 생산
   ↓
REAL WORLD
데이터 생산
   ↓
SIMULATION FACTORY
경험 증폭
   ↓
AI FACTORY
정책 학습
   ↓
다시 ROBOT
```

이 네 고리가 연결되는 순간 휴머노이드 산업은 단순한 기계 제조업이 아니게 된다.

---

28. 그리고 여기서 미국과 중국의 경쟁을 보는 새로운 좌표가 생긴다

이제 단순히

미국 = AI

중국 = 제조

라고 볼 수 없다.

비교해야 할 것은 적어도 네 축이다.

몸을 얼마나 잘 만드는가

얼마나 많이 만드는가

얼마나 많은 현실 데이터를 얻는가

그 데이터를 얼마나 빠르게 학습·시뮬레이션으로 돌리는가

이 네 축이 서로 곱해진다.

---

29. 다음부터 책의 무대가 바뀐다

지금까지는 주로 로봇 **안쪽**을 보았다.

이제 카메라를 뒤로 뺀다.

실험실 문을 연다.

밖에는 실제 회사들이 있다.

Tesla.

Figure.

Boston Dynamics.

Agility.

Unitree.

UBTECH.

그리고 중국의 수많은 신규 업체와 거대한 부품 공급망.

이제 질문은 더 이상

> **"휴머노이드는 가능한가?"**

가 아니다.

질문은 이것이다.

> **"누가 이것을 실제 제품으로 만들고 있으며, 어느 단계까지 현실에 배치했는가?"**

가능성의 언어에서 배치의 언어로 넘어갈 차례다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

PART IV

기술에서 산업으로

기술은 공급망·제조·현장운영을 통과해야 산업이 된다.

제14장 – 실험실 문을 열다

휴머노이드는 지금 어디까지 현실에 들어왔는가

이제 로봇의 몸도 만들었다.

걷게 했다. 손도 달았다. VLA를 넣고, 실제 데이터와 시뮬레이션으로 학습시키는 방법도 알아봤다.

그러면 가장 중요한 질문이 남는다.

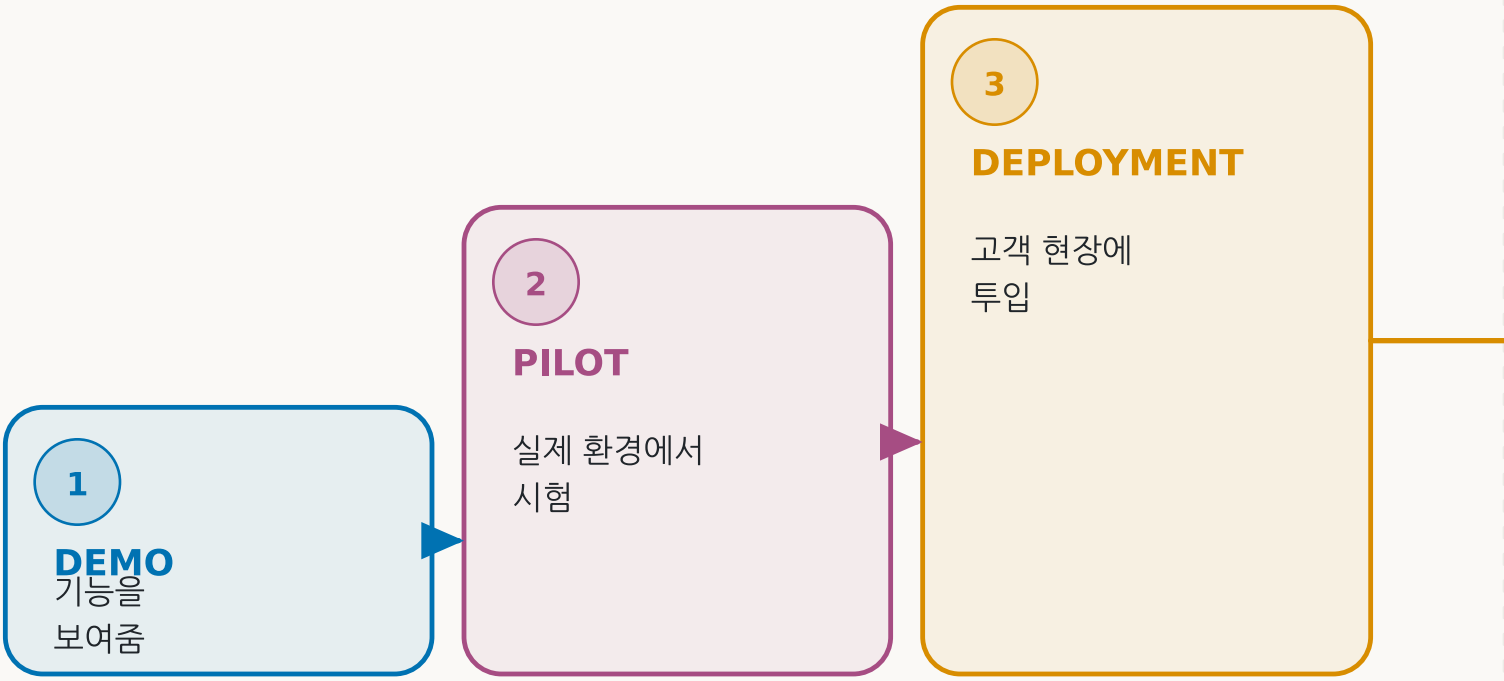
> **그래서 지금 실제로 누가 쓰고 있는가?**

여기서부터는 영상 속 데모와 산업현실을 철저히 분리해야 한다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

V13. DEMO → MASS SCALE

"할 수 있다"와 "돈을 받고 반복해서 한다" 사이의 다섯 단계



Tesla Optimus 사례

DEMO / PRE-PRODUCTION

회사·공식자료 기준

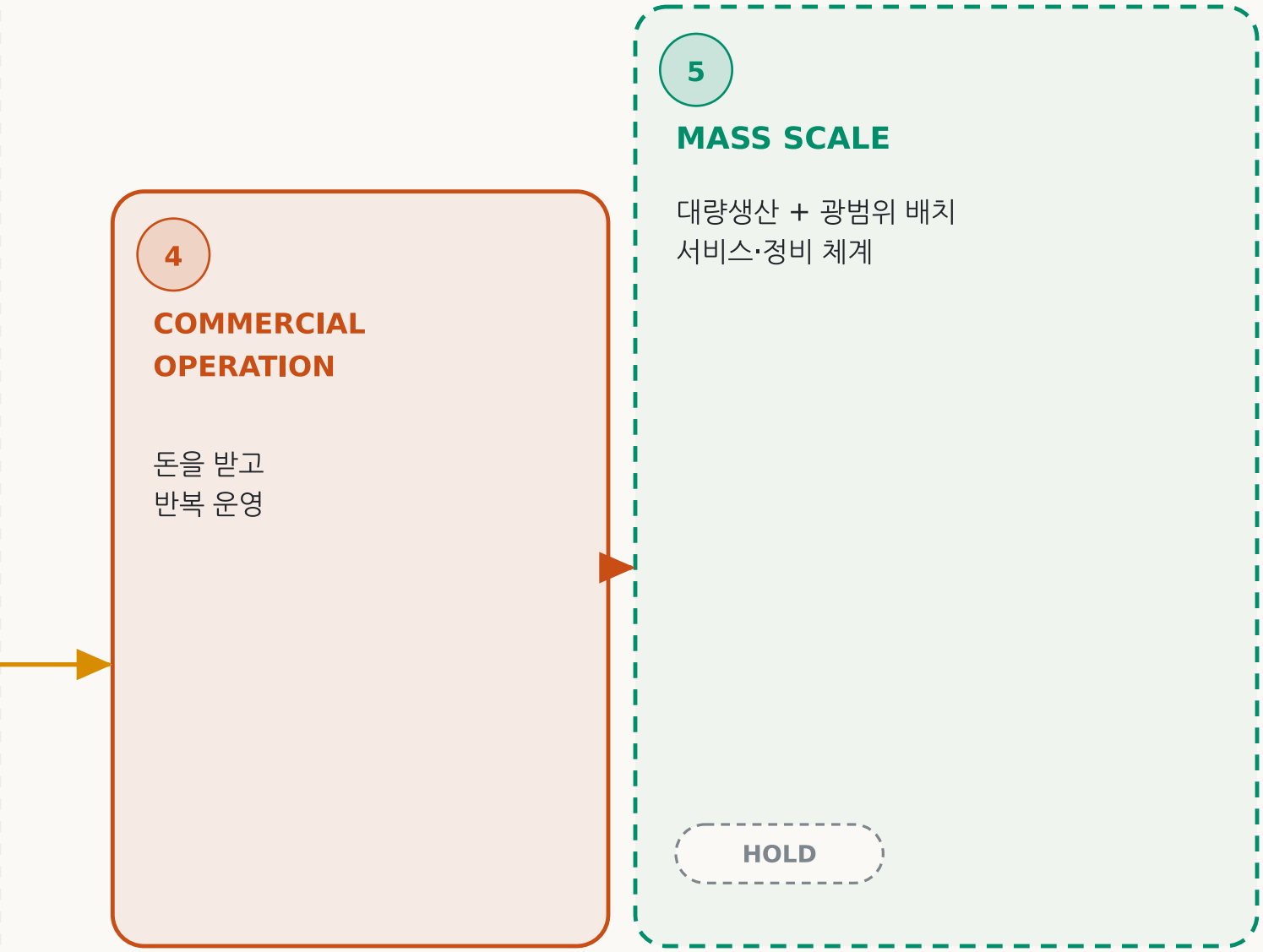
Figure02 사례

PILOT → DEPLOYMENT

한정 공정의 장기 현장시험

생산용량 ≠ 실제 생산

기준일 **2026-09-23**



Atlas 사례

DEPLOYMENT / EARLY PRODUCT

지정 파트너 중심 초기 제품

Digit · GXO 사례

COMMERCIAL OPERATION

유상 반복운영 확인

출하 ≠ 설치 ≠ 활성 가동

휴머노이드 산업에서 가장 위험한 착시는 **'할 수 있다'와 '하고 있다'를 같은 말로 사용하는 것**이다.

---

1. 먼저 다섯 단계를 구분하자

휴머노이드 영상을 볼 때 앞으로 이 자를 들이댄다.

```
LEVEL 1  DEMO
할 수 있음을 보여줌
      ↓
LEVEL 2  PILOT
실제 환경에서 시험
      ↓
LEVEL 3  DEPLOYMENT
실제 작업에 투입
      ↓
LEVEL 4  COMMERCIAL OPERATION
돈을 받고 지속적으로 작업
      ↓
LEVEL 5  MASS SCALE
대량생산 + 대규모 배치
```

이 다섯 단계는 전혀 다르다.

그리고 단계표 옆에는 사실의 성격도 붙인다.

```
현재 사실
기업 발표
미래 목표
분석가 추정
저자 해석
확인 불가
```

회사 보도자료에 나온 숫자라고 해서 모두 현재 사실이 되는 것은 아니다. 고객이 확인한 작업·기간·운영시간이 있으면 제조사 단독 주장보다 강한 근거로 취급한다.

백덤블링 영상은 Level 1에서도 가능하다.

공장에서 월요일 아침부터 금요일 저녁까지 일하는 것은 다른 문제다.

---

2. Demo – 가장 화려하고 가장 오해하기 쉬운 단계

로봇이 춤을 춘다.

달린다.

축구를 한다.

공중제비를 돈다.

계란을 집는다.

굉장한 기술적 성취다.

하지만 영상만으로는 알 수 없는 것이 많다.

몇 번 촬영했는가?

몇 번 실패했는가?

원격조종이 개입했는가?

배터리는 얼마나 갔는가?

동작 후 정비가 필요했는가?

같은 일을 1만 번 할 수 있는가?

그래서 앞으로 우리는 **데모를 폄하하지도 않고 상용화를 의미한다고 과대해석하지도 않는다.**

---

3. Pilot – 공장에 들어간다

다음 단계에서는 로봇을 실제 공장에 넣는다.

여기서 갑자기 적들이 늘어난다.

먼지.

조명 변화.

사람.

지게차.

좁은 통로.

불규칙한 물체.

네트워크 문제.

배터리.

교대근무.

그리고 무엇보다

생산라인의 시간.

실험실에서는 로봇이 실패하면 다시 시작한다.

공장에서는 실패가 생산손실이 된다.

---

4. 그래서 Pilot의 목적은 '로봇 자랑'이 아니다

진짜 질문은 이것이다.

어떤 작업에서 경제성이 있는가?

처음부터 사람의 모든 일을 시키지 않는다.

범위를 좁힌다.

예를 들면

부품 운반.

상자 이동.

빈 용기 처리.

기계에 부품 넣고 빼기.

반복적인 픽앤플레이스.

이런 작업부터 시작한다.

왜?

환경을 어느 정도 구조화할 수 있기 때문이다.

---

5. 첫 번째 중요한 사례 – Digit

Agility Robotics의 Digit은 재미있는 비교대상이다.

사람 얼굴을 흉내 내는 데 집중하지 않는다.

손도 인간 손을 그대로 복제하지 않았다.

목표가 비교적 명확하다.

물류현장에서 실제 일을 한다.

이 설계철학이 중요하다.

> **사람처럼 보이는 로봇보다**
> **사람이 만든 환경에서 경제적으로 일하는 로봇을 먼저 만든다.**

2026년 현재 Digit은 이 단계 사다리에서 중요한 기준점이다. GXO와 Agility Robotics는 다년 Robot-as-a-Service 계약을 체결했고, Digit이 실제 물류창고에서 반복 작업을 수행하며 매출을 발생시킨다고 공개했다.

따라서 기존 Digit 세대의 GXO 사례는 단순 PILOT보다 강한 COMMERCIAL OPERATION으로 분류할 수 있다. 다만 2026년에 발표된 Digit 5는 2027년 일반 공급을 목표로 하는 별도 세대다. 같은 회사의 실적이라도 세대를 섞지 않는다.

---

6. 왜 상자를 옮기는 일부터 시작할까

사람 입장에서는 단순노동이다.

로봇 입장에서는 아주 좋은 첫 시장이다.

물체가 비교적 표준화되어 있다.

출발점과 목적지가 정해져 있다.

복잡한 손기술이 적다.

작업 성공 여부를 쉽게 측정할 수 있다.

그리고 하루에 엄청나게 반복된다.

즉 AI 학습에도 좋다.

7. 두 번째 사례 – Figure

Figure는 조금 다른 길을 간다.

목표는 단순 물류기계보다 더 범용적인 휴머노이드다.

자동차 생산현장은 중요한 시험장이다.

BMW가 공개한 2025년 Spartanburg 사례에서 Figure 02는 약 10개월 동안 월요일부터 금요일까지 10시간 교대로 작업했다. BMW는 약 1,250 운영시간, X3 3만 대 이상 지원, 판금 부품 9만 개 이상 취급을 보고했다.

이것은 단순 무대 시연이 아니다. 특정 공정에서 실제 생산을 지원한 장기 현장 파일럿이다. 그러나 한 공장의 제한된 작업을 수행했다는 사실만으로 범용 대량 상업운영이라고 부를 수도 없다.

판정: PILOT → DEPLOYMENT.

왜 자동차 공장일까?

작업이 많다.

부품도 많다.

반복적이다.

하지만 완전히 고정된 산업용 로봇으로 자동화하기 애매한 구간도 존재한다.

휴머노이드가 노리는 틈이다.

8. 기존 산업로봇과 싸우는 것이 아니다

여기서 중요한 오해를 제거하자.

휴머노이드가 자동차 공장에 들어간다고 해서 기존 로봇팔을 모두 없애는 것이 아니다.

용접 한 작업을 초고속으로 반복한다면 고정식 산업로봇이 훨씬 효율적일 수 있다.

휴머노이드가 노리는 것은 오히려

> **전용 자동화 설비를 설치하기에는 작업량이나 변화가 너무 큰 영역**

이다.

즉 휴머노이드의 경쟁자는 때때로 산업용 로봇이 아니라

아직 자동화되지 않은 인간 작업이다.

---

9. 세 번째 사례 – Atlas

Boston Dynamics의 Atlas는 이 산업의 변화를 상징적으로 보여준다.

오랫동안 Atlas의 이미지는

파쿠르,

점프,

백덤블링

이었다.

그런데 전동 Atlas로 전환하면서 방향이 명확해졌다.

연구 플랫폼 → 산업 제품

이다.

Boston Dynamics는 2026년 제품형 Atlas의 생산 개시와 Hyundai RMAC·Google DeepMind 배치를 발표했다. 이는 연구로봇에서 초기 제품 배치로 이동했다는 근거다.

하지만 2026년 물량이 지정 파트너 중심이라는 사실도 함께 봐야 한다. 일반 고객을 대상으로 한 폭넓은 판매와 대량 상업운영은 아직 별개의 단계다.

판정: DEPLOYMENT / EARLY PRODUCT.

이 변화는 휴머노이드 전체의 변화를 압축해서 보여준다.

---

10. 로봇에게 백덤블링이 필요했던 이유

공장에서 백덤블링할 필요는 없다.

그러나 그 연구 과정에서

동적 균형,

전신제어,

충격대응,

고출력 액추에이터,

상태추정

같은 기술이 발전했다.

즉 백덤블링 자체가 제품은 아니었다.

극한시험이었다.

자동차 회사가 경주차를 만드는 것과 어느 정도 비슷하다.

---

11. 네 번째 사례 – Tesla Optimus

Tesla는 또 다르다.

Optimus의 가장 흥미로운 부분은 특정 데모 하나가 아니다.

Tesla는 이미

배터리,

모터,

전력전자,

AI 컴퓨팅,

비전,

공장,

공급망,

대량생산

을 가지고 있다.

정확히 말하면 Tesla가 이미 가진 것은 자동차·배터리·전동제품의 대량생산 경험과 공장 인프라다.
Optimus 자체의 대량생산 실적을 이미 가졌다는 뜻은 아니다.

따라서 질문은

> 자동차를 만드는 기계를 만드는 회사가 로봇도 자동차처럼 생산할 수 있는가?

가 된다.

---

12. Tesla의 가장 강력한 자산은 공장일 수도 있다

휴머노이드 스타트업은 고객에게 가서 테스트 장소를 구해야 한다.

Tesla에는 이미 자기 공장이 있다.

로봇을 투입한다.

↓

실패한다.

↓

데이터를 얻는다.

↓

수정한다.

↓

다시 투입한다.

이것이 Dogfooding, 자기 제품을 자기 조직에서 먼저 사용하는 방식의 강점이다.

---

13. 그런데 Tesla 이야기는 특히 '현재'와 '목표'를 분리해야 한다

휴머노이드 산업에서는 미래 생산계획이 현재 생산량처럼 전달되기 쉽다.

연간 수십만 대를 만들겠다는 목표와

오늘 실제로 공장에서 안정적으로 일하는 로봇 수는 같은 숫자가 아니다.

Tesla의 2025년 Form 10-K는 2025년 말까지 Bot을 아직 상용화하지 않았다고 명시했다. 2026년 1분기 업데이트에서 California와 Texas의 Optimus 생산시설은 모두 Construction으로 표시됐다.

Fremont 1세대 라인의 연 100만 대와 Texas 2세대 라인의 장기 연 1,000만 대는 회사가 밝힌 **라인 설계 용량과 미래 목표**다. 현재 생산량·출하량·고객 배치가 아니다.

판정: DEMO / PRE-PRODUCTION.

따라서 이 책에서는 기업마다 다음 상태를 따로 기록한다.

> 현재 확인된 사실
> 기업이 발표한 성능·운영 주장
> 기업이 제시한 미래 목표
> 분석가가 추정한 생산·출하·점유율
> 저자가 사실들을 연결해 제시한 해석

절대로 섞지 않는다.

---

14. 이제 태평양을 건넌다

중국이다.

여기서 휴머노이드 산업의 성격이 크게 달라진다.

중국에서는 휴머노이드가 AI 프로젝트인 동시에

제조업 프로젝트다.

---

15. Unitree – 가격이라는 무기

Unitree의 의미는 단순히 G1이 춤을 잘 춘다는 것이 아니다.

2026년 9월 공식 페이지 기준 G1의 시작가는 세금·배송비 제외 1만3,500달러다. R1 Air는 4,900달러, R1은 5,900달러부터다. 이 가격은 실제 하드웨어 접근 문턱이 낮아졌다는 근거다.

비교적 낮은 가격대의 휴머노이드를 시장에 내놓음으로써

대학,

연구소,

개발자,

기업

이 실제 하드웨어를 확보할 문턱을 낮춘다.

이것은 중요한 효과를 낳는다.

로봇 가격 ↓ → 개발자 ↑ → 실험 ↑ → 데이터 ↑

PC와 드론 산업에서도 익숙했던 확산구조다.

하지만 공개 시작가는 산업노동의 시간당 비용이 아니다. 손·센서·컴퓨팅 구성, 통합, 안전설비, 감독, 정비, 성공률이 빠져 있다. Unitree의 낮은 본체가격은 COMMERCIAL PRODUCT SALE의 근거이지, 고객 현장의 COMMERCIAL OPERATION을 자동으로 증명하지 않는다.

---

16. UBTECH – 공장을 먼저 노린다

UBTECH의 Walker 계열은 자동차와 제조현장을 주요 시장으로 겨냥한다.

UBTECH의 2025년 연차실적 공시는 이 기업이 제품·서비스 매출과 판매, 생산능력을 별도 지표로 보고할 만큼 상업화 단계를 전진시켰음을 보여 준다. 구체적인 수치와 범위는 18장에서 다시 검토한다.

여기서 중요한 것은 판매수량과 연간 생산능력, 고객 현장에서 실제 작업 중인 로봇 수가 서로 다른 지표라는 점이다.

특히 흥미로운 것은 앞에서 본 **자율 배터리 교환** 같은 접근이다.

이는 휴머노이드를 사람처럼 만드는 것보다

> **공장설비처럼 가동률을 높이는 것**

을 우선하는 사고방식이다.

따라서 UBTECH는 COMMERCIAL SALES / DEPLOYMENT로 분류하되, 현장별 작업시간·성공률·개입률이 확인되기 전에는 MASS SCALE이라고 단정하지 않는다.

---

17. 중국의 강점은 로봇 회사 숫자만이 아니다

휴머노이드 한 대를 분해해보자.

모터.

영구자석.

감속기.

베어링.

배터리.

BMS.

카메라.

LiDAR.

PCB.

전력반도체.

CNC 가공품.

사출품.

케이블.

중국은 이미 상당수 분야에서 거대한 제조 생태계를 갖고 있다.

이것이 휴머노이드로 **재조합**되고 있다.

---

18. EV가 로봇의 부모 산업이 된다

중국이 지난 수십 년 동안 키운

전기차,

배터리,

드론,

스마트폰,

전자제조

생태계는 서로 별개의 산업처럼 보였다.

휴머노이드에서는 이들이 만난다.



휴머노이드는 새로운 산업이면서 동시에 **기존 산업들의 합류점**이다.

---

19. 그래서 미국 대 중국이라는 단순한 표가 위험하다

흔히 이렇게 말한다.

미국 = AI

중국 = **제조**

너무 단순하다.

미국에도 Tesla 같은 대량생산 기업과 세계적인 로봇공학 기업이 있다.

중국에도 강력한 AI 연구와 로봇 소프트웨어 기업이 있다.

차이는 흑백이 아니라 **산업구조의 무게중심**에서 찾아야 한다.

---

20. 우리가 사용할 비교축은 여섯 개다

국가나 회사를 단순히 순위 매기지 않는다.

각각 다음을 본다.

BODY

몸을 얼마나 잘 만드는가.

BRAIN

범용 행동 AI는 어느 수준인가.

DATA

현실 행동 데이터를 얼마나 확보하는가.

FACTORY

얼마나 싸고 많이 만들 수 있는가.

DEPLOYMENT

실제 현장에 얼마나 배치되는가.

RELIABILITY

얼마나 오래 안정적으로 일하는가.

이렇게 보면 서로 다른 전략이 훨씬 선명해진다.

---

21. 그리고 아직 승자가 없는 이유

휴머노이드 산업에는 현재 서로 다른 강자가 있다.

운동성능이 뛰어난 회사.

AI가 강한 회사.

제조비가 낮은 회사.

손이 좋은 회사.

실제 배치가 앞선 회사.

하지만 아직

> **저렴하고 + 똑똑하고 + 손을 잘 쓰고 + 오래 작동하고 + 고장이 적고 + 대량생산되는**

모든 조건을 동시에 충족했다고 볼 만한 범용 휴머노이드가 널리 보급된 단계는 아니다.

바로 이것이 2026년의 위치다.

---

22. 그래서 ‘누가 가장 멋진가’보다 이것을 봐야 한다

앞으로 기업 발표가 나오면 다섯 질문을 던진다.

몇 대인가?

어디에서 일하는가?

무슨 일을 하는가?

사람이 얼마나 개입하는가?

얼마나 오래 반복했는가?

이 질문에 답이 없으면 일단 **데모 정보**로 분류한다.

---

23. 진짜 경주는 이제 시작된다

첫 번째 경주는 **움직일 수 있는가**였다. 이 문제는 상당 부분 해결되고 있다.

두 번째는 **일할 수 있는가**다. 지금 산업현장에서 시험 중이다.

세 번째는 **경제적으로 일할 수 있는가**다. 이것이 훨씬 어렵다.

그리고 마지막 질문이 남는다.

> **대량으로 만들 수 있는가?**

여기서 다시 로봇의 몸속으로 들어가야 한다.

이번에는 원리를 배우기 위해서가 아니다.

돈이 어디로 가는지 보기 위해서다.

제14장 마지막 챗화

연구자

“우리 로봇이 백덤블링합니다.”

공장장

“좋네요.”

연구자:

“달리기도 합니다.”

공장장:

“멋집니다.”

“춤도 춥니다.”

“정말 멋집니다.”

잠시 침묵.

공장장이 묻는다.

> **“그런데 이 부품을 하루 2,000번 옮기면 한 번당 얼마입니까?”**

연구자는 계산기를 꺼낸다.

그 순간 로봇공학이 산업이 된다.

제15장 – 로봇 한 대의 가격표를 뜯어보자

2만 달러짜리 휴머노이드는 정말 가능한가

지금까지 우리는 로봇을 기술적으로 분해했다.

이번에는 같은 로봇을 **회계적으로 분해한다.**

머리에는 카메라가 있다.

가슴에는 배터리와 컴퓨터가 있다.

관절에는 모터와 감속기가 있다.

손에는 작은 액추에이터와 센서가 빼곡하다.

이 모든 부품에는 가격표가 붙어 있다.

그런데 휴머노이드 가격을 이해하려면 처음부터 네 개의 숫자를 분리해야 한다.

> **BOM ≠ 제조원가 ≠ 판매가격 ≠ 노동비용**

이 네 숫자를 섞는 순간 거의 모든 로봇 가격 논쟁이 이상해진다.

---

1. 첫 번째 가격 – BOM

BOM, Bill of Materials.

쉽게 말하면 **부품표**다.

로봇 한 대를 만들기 위해 무엇이 몇 개 필요한지를 적는다.

```
휴머노이드
 │
 ├── 액추에이터
 │    ├── 모터
 │    ├── 감속기
 │    ├── 베어링
 │    ├── 엔코더
 │    └── 센서
 │
 ├── 손
 ├── 배터리
 ├── BMS
 ├── 전력전자
 ├── 카메라·센서
 ├── AI 컴퓨터
 ├── 통신장치
 ├── 프레임
 ├── 외피
 ├── 배선
 └── 냉각계통
```

각 부품의 조달가격을 더하면 대략적인 BOM이 나온다.

하지만 이것은 아직 로봇의 제조원가가 아니다.

2. 두 번째 가격 – 제조원가

부품이 공장에 도착했다고 로봇이 저절로 만들어지지는 않는다.

조립해야 한다.

검사해야 한다.

센서를 보정해야 한다.

소프트웨어를 넣는다.

걷게 해본다.

불량품을 골라낸다.

고장 난 부품을 다시 교체한다.

따라서

BOM

+

조립비

+

검사비

+

공장 감가상각

+

불량·수율손실

+

물류

=

제조원가

가 된다.

3. 세 번째 가격 – 판매가격

회사는 제조원가 그대로 팔 수 없다.

연구개발비가 있다.

AI 학습비용이 있다.

서비스 조직이 있다.

보증비용이 있다.

영업·관리비가 있다.

그리고 이윤도 필요하다.

따라서

제조원가 ≠ 판매가격

이다.

이 너무 당연한 구분이 휴머노이드 보도에서는 자주 사라진다.

4. 네 번째 가격 – 가장 중요한 가격

회사가 휴머노이드를 3만 달러에 샀다고 하자.

싸다.

그런데 매일 고장 난다.

사람이 옆에서 계속 지켜봐야 한다.

하루 4시간만 움직인다.

작업의 5%를 실패한다.

그러면 싼 로봇일까?

아니다.

기업이 실제로 궁금한 것은

노동 1시간의 가격

이다.

이게 이 책 후반부의 핵심 숫자다.

---

5. 다시 로봇 몸 안으로 들어가자

액추에이터는 큰 원가 레버가 될 수 있다. 그러나 공개 자료만으로 모든 휴머노이드에 공통된 원가 비율을 확정할 수는 없다.

따라서 **출처·제품범위·생산량·회계기준이 다른 BOM 추정치를 하나의 고정값으로 사용하지 않는다.**

왜냐하면 로봇마다 완전히 다르기 때문이다.

손가락 자유도.

관절 수.

감속기 종류.

센서 수.

자체제작 여부.

생산량.

페이로드.

컴퓨팅 사양.

따라서 이 책에서는 특정 비율 하나 대신 **원가를 움직이는 구조**를 본다.

---

6. 첫 번째 큰 덩어리 – 액추에이터

휴머노이드에는 수십 개의 관절이 있다.

그리고 관절마다 액추에이터가 필요하다.

하나의 액추에이터 안에는

모터,

감속기,

베어링,

엔코더,

토크센서,

하우징,

제어전자장치

등이 들어갈 수 있다.

따라서

액추에이터 단가 × 관절 수

가 빠르게 커진다.

---

7. 그래서 자유도 하나를 없애는 것이 엄청난 원가절감일 수 있다

손목에 자유도 하나 추가.

단순히 모터 하나만 추가되는 것이 아니다.

센서.

배선.

베어링.

기계구조.

제어채널.

조립공정.

소프트웨어.

고장 가능성.

모두 늘어난다.

따라서 제조업 관점에서 좋은 질문은

> “어떻게 부품을 싸게 만들까?”

만이 아니다.

> “이 부품 자체를 없앨 수 없을까?”

라는 질문이다.

---

8. 이것이 Tesla식 설계철학에서 익숙한 이유

Tesla가 자동차 제조에서 지속적으로 추구해온 중요한 원칙 중 하나가 부품과 공정의 단순화다.

휴머노이드에서도 비슷한 제조논리가 중요해진다.

관절 종류를 줄인다.

액추에이터를 공용화한다.

볼트 수를 줄인다.

배선을 줄인다.

조립공정을 줄인다.

부품 하나를 10% 싸게 만드는 것보다 부품 하나를 삭제하는 것이 더 싸다.

---

9. 두 번째 큰 덩어리 – 손

손은 작은데 비싸질 가능성이 높다.

왜?

작은 공간에

여러 자유도,

소형 모터,

전달기구,

센서,

촉각,

배선

을 넣어야 하기 때문이다.

그리고 손은 잘 부딪힌다.

마모도 많다.

그래서 미래 상용 로봇에서는 손 전체를 빠르게 교체하는 **모듈형 설계**가 경제적으로 중요해질 수 있다.

---

10. 세 번째 – 배터리

전기차와 비교하면 휴머노이드의 배터리 절대용량은 훨씬 작다.

따라서 전체 가격에서 배터리가 항상 가장 비싼 부품일 필요는 없다.

그러나 배터리는 다른 방식으로 원가에 영향을 준다.

배터리 ↑

→ 무게 ↑

→ 액추에이터 ↑

→ 프레임 ↑

→ 전력소비 ↑

즉 **간접 원가**가 크다.

---

11. 네 번째 – 센서

카메라.

IMU.

엔코더.

힘·토크 센서.

촉각.

깊이센서.

필요에 따라 LiDAR.

센서는 대량생산되면서 싸진 것도 많다.

스마트폰과 자동차 산업 덕분이다.

하지만 고정밀 힘·토크 센서와 촉각 시스템 등은 여전히 원가·내구성·통합 난도가 중요한 영역이다.

---

12. 다섯 번째 – 컴퓨터

로봇의 AI 컴퓨터는 특이한 부품이다.

반도체 가격 자체도 중요하지만

얼마나 많은 연산을 로봇 내부에서 해야 하는가

가 시스템 전체를 바꾼다.

큰 컴퓨터:

연산 ↑

하지만

전력 ↑ 열 ↑ 배터리 부담 ↑ 냉각 ↑.

따라서 저전력 AI 추론 기술은 단순한 소프트웨어 개선이 아니라 **하드웨어 BOM 절감**으로 이어질 수 있다.

---

13. 여섯 번째 – 프레임과 외피

알루미늄.

강철.

마그네슘.

복합재료.

플라스틱.

재료 가격만 보면 상대적으로 단순해 보인다.

하지만 중요한 것은 **가공비**다.

복잡한 CNC 가공품을 수십 개 사용하면 비싸진다.

생산량이 커지면

주조,

단조,

사출,

프레스

등 대량생산 방식으로 전환할 여지가 생긴다.

여기서 생산량이 가격을 바꾸기 시작한다.

---

14. 원가는 생산방식과 운영모델에서 다시 바뀐다

시제품 열 대는 CNC 가공과 수작업 조립으로 만들 수 있다. 대량제품은 부품 수, 공용화, 금형, 자동조립, 검사와 서비스까지 처음부터 다시 설계해야 한다. 이 문제는 20장의 주제다.

공급망이 가까우면 설계→샘플→시험→수정의 반복비용을 낮출 수 있다. 그러나 공개 시작가격이 낮다는 사실만으로 같은 성능·손·센서·서비스·신뢰성을 더 싼 원가로 만들었다고 결론낼 수는 없다. 공급망과 국가별 차이는 16~18장에서 검증한다.

구매가격 뒤에는 전기, 정비, 배터리, 소프트웨어, 보험, 감독과 다운타임이 남는다. 판매·리스·RaaS 중 어떤 계약을 택해도 중요한 것은 비용을 누가 부담하는지와 작업 한 건을 끝내는 총비용이다. TCO와 RaaS는 21장에서 운영조건과 함께 계산한다.

---

15. 이제 가장 중요한 숫자를 만든다

로봇 한 대:

30,000달러.

5년 사용.

여기까지만 보면 싸다.

그러나 이것만으로 아무것도 알 수 없다.

우리가 필요한 것은

```
총소유비용 TCO
=
구입·리스
+ 금융비용
+ 전기
+ 소프트웨어
+ 정비
+ 부품
+ 보험
+ 충전 인프라
+ 인간 감독
+ 다운타임 손실
```

그리고 이것을 **실제 생산시간**으로 나눈다.

---

16. 하지만 아직 하나 빠졌다

성공률이다.

로봇이 1시간 동안 100개의 부품을 처리했다.

95개 성공.

5개 실패.

사람이 와서 수정.

그렇다면 로봇의 실제 생산량은 100이 아니다.

그래서 더 정확한 분모는

성공적으로 완료된 작업

이다.

---

17. 이 책의 핵심 경제단위가 탄생한다

지금까지 휴머노이드 가격을

한 대 얼마?

로 물었다.

이제 질문을 바꾼다.

> **성공한 행동 하나가 얼마인가?**

이를 개념적으로

Cost per Successful Action

이라고 부르자.

이것이 휴머노이드 경제를 이해하는 훨씬 강력한 단위다.

---

18. 예를 들어 두 로봇이 있다

Robot A

가격 1,500만원.

작업 성공률 90%.

자주 고장.

인간 개입 많음.

Robot B

가격 4,000만원.

성공률 99.9%.

24시간 교대운영 가능.

개입 거의 없음.

어느 것이 더 싼가?

판매가격만으로는 알 수 없다.

이것이 휴머노이드 산업에서 '가격파괴'라는 말을 조심해서 사용해야 하는 이유다.

---

19. 그리고 이제 다음 질문이 생긴다

왜 어떤 회사는 같은 로봇을 훨씬 싸게 만들 수 있는가?

모터 때문인가?

감속기인가?

배터리인가?

희토류인가?

센서인가?

공장인가?

국가의 산업정책인가?

답은 하나가 아니다.

공급망 전체다.

그래서 이제 로봇 한 대에서 카메라를 더 뒤로 뺀다.

공장 밖으로 나간다.

그 뒤에는 수백 개의 회사가 있다.

그리고 그 뒤에는

광산,

정유·정제,

제철,

화학,

반도체,

배터리,

자동차 산업이 있다.

제15장 마지막 챗화

CEO

“우리 로봇은 2만 달러입니다.”

고객

“좋네요.”

CEO:

“경쟁사는 5만 달러입니다.”

고객:

“그런데 하루에 몇 시간 일하죠?”

CEO:

“네?”

“몇 번 고장납니까?”

“…….”

고객

“사람이 얼마나 붙어 있어야 합니까?”

CEO:

“초기에는 약간의 감독이 필요합니다.”

고객:

“배터리는요?”

“교대 충전이 필요합니다.”

“손이 고장 나면?”

“교체해야 합니다.”

“소프트웨어 비용은?”

CEO가 잠시 멈춘다.

고객이 계산기를 내려놓는다.

> **“그러니까 로봇 가격 말고, 제 일을 한 번 끝내는 데 얼마냐고요.”**

그 순간 **2만 달러라는 숫자의 마법이 사라진다.**

제16장 – 보이지 않는 공급망

로봇 한 대 뒤에는 몇 개의 산업이 숨어 있는가

휴머노이드 한 대를 책상 위에 세워 놓자.

우리는 지금까지 그것을 하나의 제품으로 봤다.

하지만 제조업의 눈으로 보면 로봇 한 대는 제품 하나가 아니다.

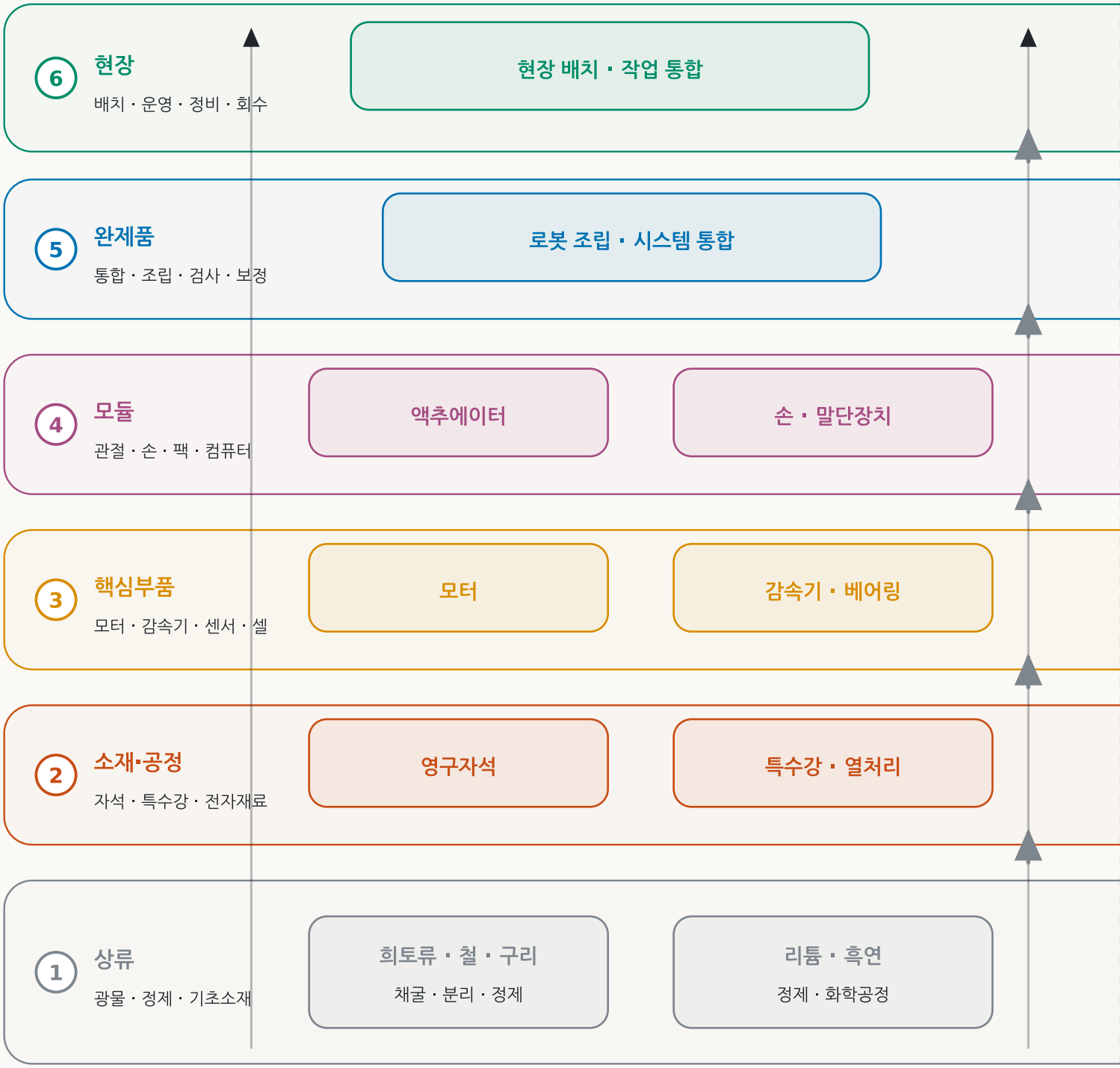
수십 년 동안 서로 다른 산업이 발전시킨 기술을 한 몸에 압축한 공급망의 종착점이다.

로봇을 거꾸로 분해해보면 AI에서 광산까지 내려간다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

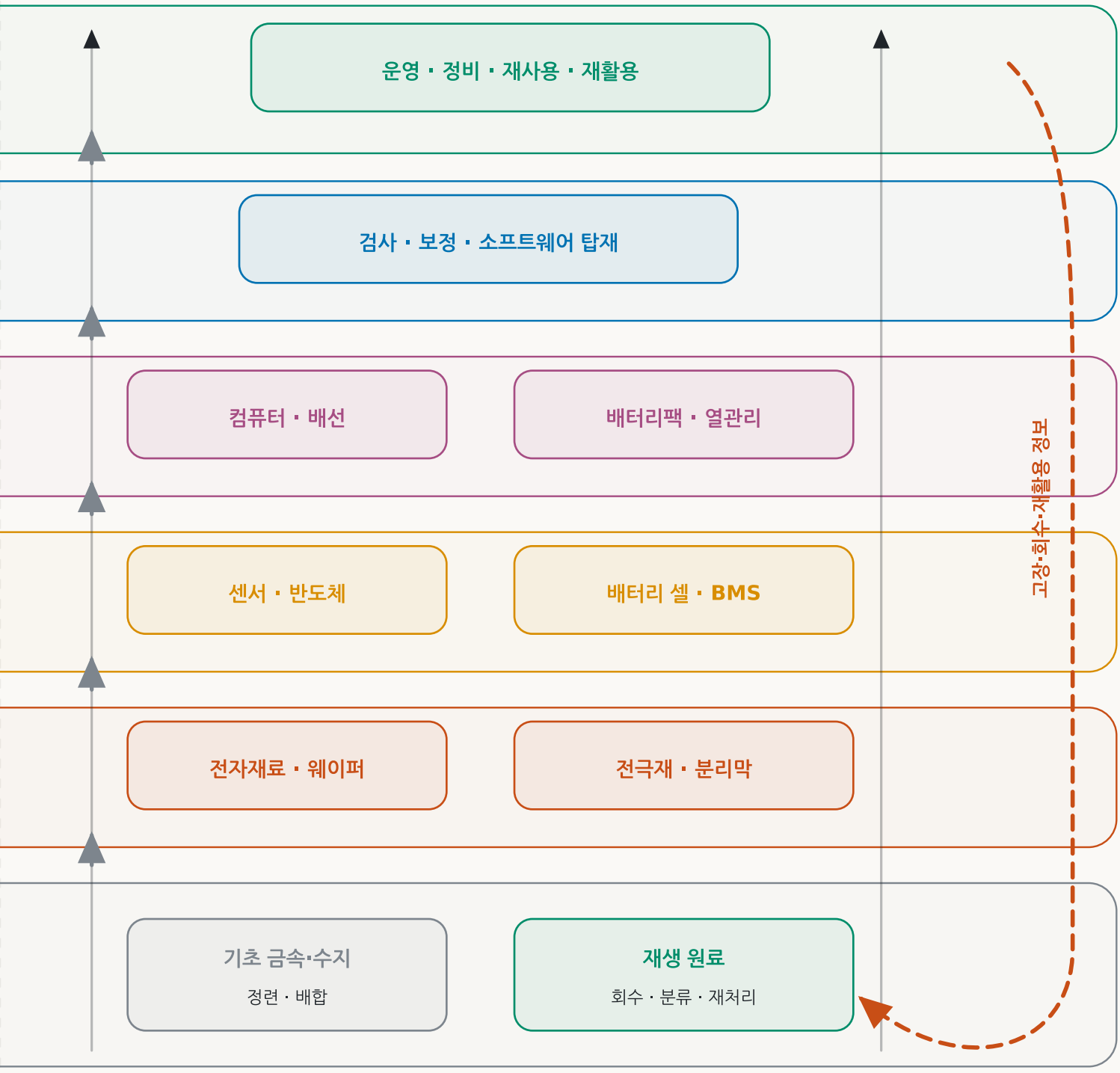
V14. 휴머노이드 공급망

Physical AI의 끝에는 광산·정제·공정·조립·서비스가 연결되어 있다



병목 = 가격뿐 아니라 대체가능성·수율·확장속도

개념도 · 비중표 아님



선 굵기·면적은 시장점유율이나 국가비중이 아님

1. 로봇의 계보를 거꾸로 따라가자

휴머노이드
↓
액추에이터·손·센서·컴퓨터·배터리
↓
모터·감속기·베어링·반도체·셀
↓
영구자석·특수강·구리·전자소재
↓
희토류·철·구리·리튬·흑연 등
↓
광산·정제·소재

맨 위에는 AI 로봇이 있다.

맨 아래에는 광산과 화학공장이 있다.

Physical AI는 놀랄 만큼 물질적인 산업이다.

---

2. 첫 번째 사슬 – 희토류에서 관절까지

고성능 영구자석 모터를 예로 들어보자.

광산에서 광물을 캔다.

↓

분리·정제한다.

↓

자석 소재를 만든다.

↓

영구자석을 만든다.

↓

모터를 만든다.

↓

액추에이터에 넣는다.

↓

관절에 넣는다.

```
희토류
 ↓
분리·정제
 ↓
NdFeB 계열 자석
 ↓
모터
 ↓
액추에이터
 ↓
관절
```

IEA의 2026년 희토류 보고서에 따르면 2024년 중국은 자석용 희토류 채굴의 약 60%, 정제의 91%, 소결 영구자석 생산의 94%를 차지했다. 이는 휴머노이드 한 대의 BOM을 뜻하는 수치가 아니라, 고성능 영구자석으로 이어지는 상류 공급망의 집중도를 보여 주는 수치다.

따라서 휴머노이드 생산량이 크게 늘면 로봇 회사만 성장하는 것이 아니다. 그 아래 공급망에도 수요와 병목 위험이 함께 전달된다.

---

3. 하지만 '로봇 = 희토류 투자'처럼 단순화하면 안 된다

로봇 한 대가 사용하는 자석의 양은 설계마다 다르다.

모터 구조도 달라질 수 있다.

재료 절감도 일어난다.

재활용도 가능하다.

생산량 전망도 불확실하다.

따라서

휴머노이드 증가 → 특정 원자재 가격 폭등

처럼 바로 연결하면 위험하다.

중간에

기술대체,

효율향상,

공급확대,

가격탄력성

이 존재한다.

이 책은 공급망을 보되 **단선적인 테마주 논리로 가지 않는다.**

---

4. 두 번째 사슬 – 감속기

특수강.

↓

정밀 성형·절삭.

↓

열처리.

↓

표면처리.

↓

베어링.

↓

윤활.

↓

정밀조립.

↓

검사.

우리가 3장에서 본 하모닉 계열 감속기는 그림으로 보면 단순했다.

하지만 공급망으로 보면 전혀 단순하지 않다.

---

5. 진짜 장벽은 도면보다 공정이다

플렉스 스플라인의 모양을 안다고 제품이 나오지 않는다.

수많은 제품에서

같은 강도,

같은 피로수명,

같은 정밀도,

같은 소음

을 재현해야 한다.

이것이 **공정능력**이다.

그래서 제조업에서는 때때로 특허보다 더 무서운 것이 있다.

> **수십 년 동안 쌓인 수율 데이터.**

---

6. 수율이 가격을 결정한다

감속기 100개를 만든다.

A 공장:

95개 합격.

B 공장:

70개 합격.

재료비가 같아도 실제 합격품 하나의 원가는 크게 달라진다.

그래서 중국의 부품 국산화를 볼 때도

"만들 수 있는가?"

다음 질문이 반드시 필요하다.

> **“높은 수율로 반복 생산할 수 있는가?”**

---

7. 세 번째 사슬 – 베어링

베어링은 너무 오래된 기술이라 첨단산업 이야기에서 잘 보이지 않는다.

그러나 로봇의 모든 회전관절에는 하중이 걸린다.

정밀도.

강성.

마찰.

수명.

소음.

이 모두에 베어링이 관여한다.

AI가 아무리 좋아져도 베어링이 마모되면 손끝 정밀도는 떨어진다.

첨단 로봇의 발목을 19세기적 기계요소가 잡을 수도 있다.

---

8. 네 번째 사슬 – 배터리

광물.

↓

양극재·음극재.

↓

전해질·분리막.

↓

셀.

↓

모듈·팩.

↓

BMS.

↓

열관리.

↓

로봇.

전기차와 로봇은 셀·소재·팩·BMS·열관리의 일부 산업기반을 공유한다. IEA는 2025년 중국이 세계 배터리 셀 생산의 80% 이상, 양극활물질의 약 85%, 음극활물질의 90% 이상을 차지했다고 집계했다.

이 수치는 전기차 배터리 공급망의 집중도다. 휴머노이드용 팩의 품질·원가·가동시간을 직접 증명하지는 않지만, 로봇 배터리가 완전히 백지의 공급망에서 출발하지 않는다는 점은 보여 준다.

---

9. 다섯 번째 사슬 – 반도체

휴머노이드에는 하나의 칩만 들어가는 것이 아니다.

AI 가속기.

CPU.

MCU.

전력반도체.

메모리.

통신칩.

각종 센서용 반도체.

모터 드라이버.

BMS 칩.

즉 '로봇 반도체'라는 하나의 시장보다 **서로 다른 반도체 생태계의 결합**으로 보는 것이 정확하다.

10. 그리고 가장 비싼 칩이 항상 가장 중요한 칩은 아니다

AI 가속기가 고장 나면 로봇은 멈춘다.

하지만 2달러짜리 작은 센서나 커넥터가 고장 나도 로봇은 멈출 수 있다.

대량생산에서는 이런 현상이 중요하다.

> **제품의 신뢰성은 가장 화려한 부품이 아니라 가장 약한 부품에 의해 결정될 수 있다.**

11. 여섯 번째 사슬 – 센서

스마트폰이 로봇에게 준 유산이 있다.

카메라.

IMU.

마이크.

각종 소형 센서.

수십억 대의 스마트폰이 생산되면서 센서의

가격,

크기,

전력소비

가 크게 개선됐다.

자동차 산업은 또 다른 것을 줬다.

카메라.

레이더.

LiDAR 생태계.

고신뢰 전장부품.

휴머노이드는 이 두 산업의 유산을 흡수한다.

---

12. 드론도 부모 산업이다

드론은

고출력 BLDC 모터,

IMU,

비행제어,

배터리,

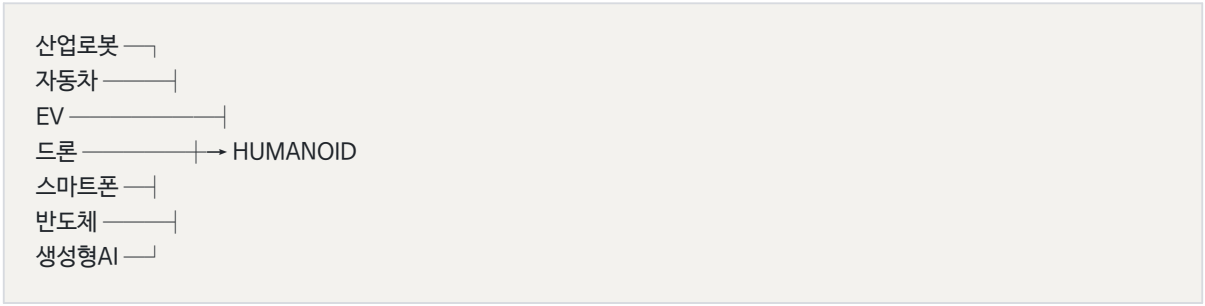
경량구조,

실시간 자세제어

를 대량생산했다.

놀랍게도 이것들은 휴머노이드와 상당히 겹친다.

그래서 휴머노이드는 한 산업의 후손이 아니다.



여섯 개 이상의 산업이 한 몸에서 만난다.

---

13. 공급망은 국가명이 아니라 연결구조로 본다

부품업체·가공업체·조립라인이 가까이 있으면 운송비뿐 아니라 설계 수정과 재시험의 시간이 줄어든다. 반대로 첨단 모델·반도체·소프트웨어, 정밀감속기·소재·배터리 같은 강점은 서로 다른 지역에 분산돼 있다.

따라서 '국산화율' 하나보다 공급 안정성, 품질, 대체가능성, 반복속도와 병목 복구력을 함께 봐야 한다. 미국·중국의 서로 다른 조합은 다음 두 장에서 비교한다.

---

14. 그리고 공급망에는 숨은 병목이 생긴다

휴머노이드 생산이 갑자기 100배 증가한다고 가정하자.

모든 부품 생산량을 하루아침에 100배 늘릴 수 있을까?

아니다.

특정 감속기.

특정 베어링.

고성능 자석.

정밀 엔코더.

촉각센서.

생산장비.

검사장비.

어딘가에서 병목이 생길 수 있다.

이것이 Bottleneck이다.

---

15. 병목 부품은 가장 비싼 부품일 필요가 없다

1달러짜리 커넥터가 없다.

10만 달러짜리 로봇을 출하하지 못한다.

자동차 산업이 반도체 부족 때 경험했던 것과 같은 원리다.

따라서 공급망 분석에서는

가격 비중

뿐 아니라

대체 가능성

을 봐야 한다.

---

16. 우리가 사용할 공급망 병목 지도

각 부품을 다섯 축으로 평가한다.

> ① 기술 난도
> ② 공급업체 수
> ③ 생산 확대 속도
> ④ 대체 가능성
> ⑤ 로봇 성능에 미치는 영향

이렇게 보면

비싸지만 쉽게 대체되는 부품과,

싸지만 대체하기 어려운 부품을 구분할 수 있다.

---

17. 공급망의 최종 경쟁은 '국산화율'만이 아니다

모든 부품을 한 나라에서 만들 필요는 없다.

더 중요한 것은

싸게,

안정적으로,

충분한 품질로,

빠르게,

대량으로

조달할 수 있는가다.

즉 공급망 경쟁력은

> **자급률이 아니라 전체 시스템의 회복력과 속도**

로 봐야 한다.

---

18. 그리고 AI가 공급망 자체를 바꾸기 시작한다

생성형 설계.

자동 품질검사.

예지정비.

공정 최적화.

부품 설계 자동화.

시뮬레이션.

AI는 로봇의 뇌에만 들어가는 것이 아니다.

로봇을 만드는 공장에도 들어간다.

따라서 AI는 휴머노이드의 성능과 생산비를 동시에 건드린다.

---

19. 이제 중요한 순환고리가 완성된다

```
AI 개선
 ↓
로봇 성능 ↑
 ↓
수요 ↑
 ↓
생산량 ↑
 ↓
부품 단가 ↓
 ↓
로봇 가격 ↓
 ↓
배치량 ↑
 ↓
현실 데이터 ↑
 ↓
AI 개선
 ←
```

이것이 휴머노이드 산업이 기대하는 Physical AI Flywheel이다.

그러나 어느 화살표 하나라도 막히면 순환은 느려진다.

---

20. 그래서 앞으로 승자를 찾는 방법도 달라진다

최고의 AI 모델만 찾으면 안 된다.

가장 멋진 로봇만 봐도 안 된다.

가장 싼 로봇만 봐도 안 된다.

우리가 볼 것은

> **누가 이 순환고리를 가장 빨리 돌릴 수 있는가**

가 핵심 질문이다.

BODY.

BRAIN.

DATA.

FACTORY.

SUPPLY CHAIN.

이제 다섯 번째 축이 추가됐다.

---

21. 그리고 드디어 두 거대한 산업 시스템을 비교할 준비가 됐다

한쪽에는

실리콘밸리,

AI 모델,

GPU,

벤처자본,

Tesla,

Figure,

Boston Dynamics,

Agility 등이 있다.

다른 한쪽에는

선전·상하이 등의 제조 클러스터,

EV 공급망,

배터리,

모터,

감속기,

Unitree,

UBTECH,

AgiBot 등 다양한 업체가 있다.

하지만 다음 장에서 '누가 이긴다'고 순위를 매기지는 않는다.

각자가 **어떤 문제를 먼저 풀려고 하는지**를 본다.

제16장 마지막 챗화

AI 연구자

"우리 모델이 가장 똑똑합니다."

로봇 엔지니어

"우리 액추에이터가 더 강합니다."

공장장

"우리는 하루에 천 대 만들 수 있습니다."

부품업체

"잠깐."

모두 돌아본다.

부품업체:

> **"그 베어링, 다음 달에 3만 개밖에 못 만드는데요."**

침묵.

공급망이 웃는다.

> **"완제품의 속도는 가장 느린 부품의 속도를 넘을 수 없다."**

제17장 – 미국

하나의 휴머노이드 전략은 없다

‘미국 휴머노이드’라는 표현은 편리하지만 실제 산업을 이해하는 데는 오히려 방해가 된다.

같은 미국 기업이라도 풀고 있는 문제가 다르다.

Tesla는 **자동차처럼 대량생산할 수 있는가**를 묻는다.

Figure는 **범용 행동지능과 몸을 얼마나 빨리 통합할 수 있는가**를 묻는다.

Boston Dynamics는 **기계의 운동능력과 산업 신뢰성을 어디까지 끌어올릴 수 있는가**를 묻는다.

Agility Robotics는 **범용성보다 먼저 실제 물류업무에서 돈을 벌 수 있는가**를 묻는다.

Apptronik은 **산업 파트너와 함께 범용 플랫폼을 실제 작업에 확장할 수 있는가**를 시험한다.

따라서 이번 장에서는 회사를 순위 매기지 않는다.

먼저 책 전체의 시스템 공식을 사용한다.

> BODY × BRAIN × DATA × FACTORY × SUPPLY CHAIN

여기에 기업의 현재 위치를 판단하는 운영 점검표를 별도로 붙인다.

> DEPLOYMENT + RELIABILITY

앞의 곱셈은 경쟁력을 만드는 구조이고, 뒤의 두 항목은 그 구조가 현실에서 작동하는지 확인하는 성적표다. 둘을 하나의 공식으로 섞지 않는다.

2026년 9월 상태선

기업·플랫폼	확인된 현재 단계	아직 목표·주장인 것
Tesla Optimus	데모·양산 준비, 생산시설 건설	연 100만·1,000만 대 설계 용량, 상용화
Figure	BMW 장기 생산 파일럿·현장 배치	BotQ 생산능력과 4년 생산 목표
Boston Dynamics Atlas	제품형 전환·생산 개시 발표·지정 파트너 배치 약정	광범위 고객 판매·대량 상업운영
Agility Digit	GXO 유상 반복운영	Digit 5의 2027년 일반 공급
Apptronik Apollo	Mercedes-Benz 등 고객 파일럿	누적 대량출하·지속 상업운영
DeepMind·NVIDIA	모델·훈련·시뮬레이션 플랫폼	특정 로봇의 상업적 성공은 별도 증거 필요

---

1. Tesla – 로봇을 자동차처럼 만들 수 있는가

Tesla의 Optimus를 이해할 때 가장 먼저 버려야 할 질문은 이것이다.

> “지금 가장 뛰어난 휴머노이드인가?”

더 흥미로운 질문은 따로 있다.

> “Tesla의 자동차 산업 시스템을 휴머노이드에 복제할 수 있는가?”

Tesla에는 로봇 스타트업 대부분이 처음부터 구축해야 하는 자산들이 이미 있다.

배터리.

모터.

전력전자.

AI 컴퓨팅.

비전.

공장.

공급망.

대량생산 경험.

---

2. Tesla의 가설

Tesla의 산업가설을 한 줄로 압축하면 이렇다.

```
EV
↓
전동화 기술
↓
AI·비전
↓
공장
↓
대량생산
↓
Optimus
```

즉 휴머노이드를 완전히 새로운 기계라기보다 **또 하나의 대량생산 전동제품**으로 접근한다.

이것은 매우 중요한 차이다.

---

3. 자동차와 로봇의 공통점

배터리를 사용한다.

모터를 사용한다.

전력전자가 필요하다.

카메라를 사용한다.

AI 컴퓨터가 필요하다.

열을 관리해야 한다.

대량생산해야 한다.

따라서 Tesla는 상당한 기술자산을 재사용할 수 있다.

---

4. 하지만 자동차와 로봇의 차이는 더 중요하다

자동차의 핵심 출력은 대체로 바퀴 몇 개에 집중된다.

휴머노이드는 수십 개의 관절을 제어한다.

자동차는 도로 위에서 움직인다.

로봇은 물건을 잡고 문을 열고 도구를 사용해야 한다.

자동차의 행동공간과 휴머노이드의 행동공간은 크게 다르다.

따라서

> **FSD가 있으니 Optimus도 자연스럽게 해결된다**

고 말할 수는 없다.

Tesla의 강점은 출발점이지 성공의 보증서가 아니다.

---

5. Tesla의 또 하나의 무기 – 자기 공장

휴머노이드 스타트업은 고객사를 찾아야 한다.

Tesla는 자기 공장을 시험장으로 사용할 수 있다.

로봇 배치.

↓

실패.

↓

데이터.

↓

수정.

↓

재배치.

이것은 우리가 앞에서 본 Physical AI Flywheel을 내부에서 돌릴 수 있다는 전략적 가능성이다.
실제로 몇 대가 어느 작업에서 얼마나 오래 자율 가동했는지는 별도 증거가 필요하다.

---

6. 하지만 Tesla를 평가할 때 가장 조심해야 할 숫자가 있다

미래 생산목표.

기업이 발표하는 향후 생산능력과 현재 실제 생산량은 분리해야 한다.

2025년 Form 10-K에서 Tesla는 아직 Bot을 상용화하지 않았다고 밝혔다. 2026년 1분기 업데이트에서도 California와 Texas의 Optimus 시설은 건설 단계였다.

따라서 Tesla의 현재 강점은 **검증된 Optimus 양산 실적**이 아니라, 양산을 시도할 수 있는 기존 산업 기반이다.

따라서 Tesla를 포함한 모든 기업에 앞으로 같은 규칙을 적용한다.

① 현재 확인된 기술
② 현재 확인된 생산
③ 현재 확인된 실제 배치
④ 회사가 발표한 미래 목표

④를 ①~③처럼 쓰지 않는다.

---

7. Figure – 몸과 AI를 함께 만든다

Figure의 산업가설은 조금 다르다.

핵심은 휴머노이드 몸 자체만이 아니다.

범용 행동지능을 몸과 빠르게 공동개발한다.

Figure Body
↕
Robot AI
↕
Real-world Data
↕
New Body / New Policy

하드웨어와 AI가 서로를 바꾼다.

---

8. 왜 자동차 공장이 좋은 학교인가

Figure가 자동차 생산환경을 시험장으로 활용하는 것은 우연이 아니다.

자동차 공장은

반복성이 있지만,

작업 종류가 많고,

실제 생산가치가 있으며,

품질 요구가 높다.

즉 완전히 자유로운 가정보다 쉽지만, 단순 실험실보다 훨씬 어렵다.

Physical AI의 중간 시험장으로 좋다.

BMW가 확인한 2025년 Spartanburg 사례는 이 가설이 실제 공장으로 들어간 사례다. Figure 02는 약 10개월·1,250시간 동안 판금 부품을 취급하며 X3 3만 대 이상 생산을 지원했다.

이 수치는 DEPLOYMENT의 근거지만, 한정된 공정의 장기 파일럿을 범용 MASS SCALE로 확대해석해서는 안 된다.

---

9. Figure의 가설이 성공하려면

AI 데모만 잘해서는 안 된다.

다섯 가지가 함께 가야 한다.

손이 충분히 좋아야 한다.

로봇이 오래 버텨야 한다.

생산량을 늘려야 한다.

데이터 플라이휠이 작동해야 한다.

그리고 작업당 비용이 내려가야 한다.

결국 Figure도 다시 제조업으로 돌아온다.

---

10. Boston Dynamics – 가장 긴 운동의 역사

Boston Dynamics는 다른 계보에서 출발한다.

질문은 오랫동안 이것이었다.

> **"기계의 몸은 어디까지 움직일 수 있는가?"**

걷는다.

달린다.

점프한다.

외란을 회복한다.

전신을 협응한다.

이 회사가 오랫동안 쌓아온 자산은 **동역학과 제어의 깊이**다.

---

11. Atlas의 변화는 산업 전체의 변화다

과거 Atlas:

연구.

유압.

극한 동작.

↓

새 Atlas:

전동.

산업.

실제 작업.

이 변화는 상징적이다.

> **휴머노이드가 '가능성 증명'에서 '제품화'로 이동한다.**

2026년 제품형 Atlas의 생산 개시와 Hyundai RMAC·Google DeepMind 배치 약정은 이 이동을 실제 일정으로 만든다. 다만 지정 파트너 중심의 초기 배치이므로 단계는 DEPLOYMENT / EARLY PRODUCT다.

---

12. 현대자동차가 여기서 중요해진다

Boston Dynamics는 이제 고립된 연구소가 아니다.

Hyundai Motor Group이라는 대규모 제조기업과 연결되어 있다.

자동차 공장.

부품 공급망.

제조공학.

실제 작업환경.

이것은 Atlas가 연구로봇에서 산업제품으로 이동할 때 중요한 환경을 제공할 수 있다.

그리고 여기서 미국 로봇 기술과 한국 제조업이 만난다.

---

13. Agility Robotics – 인간을 꼭 닮을 필요가 있는가

Agility Robotics의 Digit을 보면 바로 이상한 점을 발견한다.

사람과 완전히 똑같이 생기지 않았다.

왜?

목표가 다르기 때문이다.

> **사람처럼 보이는 것이 아니라 사람용 환경에서 일한다.**

이 구분은 중요하다.

---

14. Digit의 전략

물류현장에서 반복되는 작업을 찾는다.

상자나 tote를 옮긴다.

사람이 만든 통로를 사용한다.

기존 시설을 크게 바꾸지 않는다.

범용 가정로봇보다 좁은 문제를 먼저 해결한다.

```
범용성 ↓
초기 작업범위 ↓
신뢰성 ↑
현장배치 ↑
경제성 검증 ↑
```

일종의 **좁게 들어가서 넓히기** 전략이다.

GXO의 다년 RaaS 계약과 live warehouse operation은 이 전략이 매출을 동반한 반복운영까지 갔음을 보여준다. 현재 공개 근거에서는 Digit이 미국 기업 사례 가운데 COMMERCIAL OPERATION 판정에 가장 가까운 축이다.

---

15. 이것은 중요한 산업적 반론이다

휴머노이드가 성공하려면 처음부터

요리,

청소,

육아,

공장,

물류

를 모두 해야 하는가?

아니다.

한 가지 작업에서 경제성을 먼저 확보하고 그다음 작업을 추가하는 경로가 있을 수 있다.

스마트폰처럼 처음부터 범용제품으로 등장할 필요가 없다.

---

16. Apptronik – 플랫폼을 산업 파트너에게 연결한다

Apptronik의 Apollo 역시 또 다른 가설을 보여준다.

NASA 계열 로봇 개발 경험에서 이어진 기술을 바탕으로 산업현장의 다양한 파트너와 범용 휴머노이드 플랫폼을 연결하는 방향이다.

여기서 중요한 것은 특정 데모보다

> **하나의 플랫폼이 여러 산업 작업으로 확장될 수 있는가**

라는 질문이다.

Mercedes-Benz의 Apollo 시험과 상업 협약은 실제 산업 파트너가 있다는 근거다. 그러나 계약과 시험을 누적 대량배치로 번역할 수는 없다. 현재 판정은 PILOT이다.

---

17. 그리고 Google DeepMind가 몸 밖에서 들어온다

Google DeepMind는 완성 휴머노이드 제조업체와는 다른 위치에 있다.

관심은

범용 로봇 지능.

웹에서 배운 세계지식과

시각,

언어,

행동

을 어떻게 연결할 것인가.

즉 로봇의 **뇌 계층**을 플랫폼화하려는 방향이다.

RT-2 이후 Gemini Robotics 계열은 embodied reasoning과 행동모델의 역할을 더 명확히 나누고 있다. 이 진전은 모델 계층의 현재 사실이지, 특정 휴머노이드의 상업운영 증거는 아니다.

---

18. NVIDIA는 또 다른 층을 장악하려 한다

NVIDIA 역시 완성 휴머노이드 회사로만 볼 수 없다.

GPU.

로봇 컴퓨터.

시뮬레이션.

합성데이터.

AI 학습.

Foundation Model.

즉 로봇 회사들이 로봇을 만드는 데 필요한 **계산 인프라**를 공급하는 위치다.

GR00T, Isaac, Omniverse·Cosmos, DGX·Jetson은 각각 행동모델, 개발·훈련, 시뮬레이션, 훈련·현장 추론을 잇는다. 이 스택은 데이터공장을 만들 수 있게 하지만 고객 로봇의 성공률과 가동률을 대신 증명하지 않는다.

골드러시의 광부라기보다 곡괭이·철도·훈련장을 함께 제공하는 쪽에 가깝다.

---

19. 미국 생태계를 층으로 보면 훨씬 명확하다

```
AI / FOUNDATION MODEL
Google DeepMind 등
      ↓

COMPUTE / SIMULATION
NVIDIA 등
      ↓

HUMANOID PLATFORM
Tesla
Figure
Boston Dynamics
Agility
Apptronik
      ↓

FACTORY / CUSTOMER
자동차
물류
제조
서비스
```

그리고 실제로는 이 층들이 서로 겹치고 협력한다.

---

20. 미국의 강점은 무엇인가

하나로 압축하면

지능과 자본과 고급 시스템공학의 밀도다.

AI.

컴퓨팅.

소프트웨어.

로봇공학 연구.

벤처자본.

대형 기술기업.

그리고 Tesla·Hyundai 계열처럼 실제 제조와 연결되는 통로도 있다.

그러나 약점 후보 역시 명확하다.

---

21. 로봇은 소프트웨어처럼 복사되지 않는다

GPT 모델은 소프트웨어를 배포하면 수많은 사용자가 즉시 접근할 수 있다.

휴머노이드는 그렇지 않다.

철을 깎아야 한다.

자석이 필요하다.

모터를 감아야 한다.

감속기를 만들어야 한다.

배터리를 조립해야 한다.

검사해야 한다.

배송해야 한다.

고장 나면 고쳐야 한다.

따라서 AI 산업에서 성공한

> Software Scaling

이 Physical AI에서는

Manufacturing Scaling

이라는 두 번째 장벽을 만난다.

22. 그래서 Tesla가 미국 생태계에서 특이하다

Tesla는

AI 회사이면서

배터리 사용자이고

모터 제조기업이며

자동차 제조기업이고

대규모 공장 운영자다.

즉 미국의 소프트웨어적 강점과 중국식 제조규모 논리 사이를 연결하려는 독특한 위치에 있다.

Optimus의 전략적 중요성이 여기서 나온다.

23. 하지만 제조량만 많아도 안 된다

100만 대의 멍청한 로봇을 만들면?

100만 대의 비싼 기계가 된다.

AI만 좋으면?

화면 속에서는 똑똑하지만 현실에서 일을 못 한다.

그래서 우리가 만든 식이 다시 돌아온다.

BODY × BRAIN × DATA × FACTORY × SUPPLY CHAIN

곱셈이라는 점이 중요하다.

어느 하나가 거의 0이면 전체가 크게 제한된다.

24. 미국 기업들은 사실 서로 다른 실험을 하고 있다

Tesla:

> **수직통합과 대량생산이 답인가?**

Figure:

> **몸과 범용 AI의 빠른 공동진화가 답인가?**

Boston Dynamics:

> **깊은 동역학·제어 기술을 산업제품으로 전환할 수 있는가?**

Agility:

> **좁은 실제 작업에서 먼저 경제성을 만들 수 있는가?**

Apptronik:

> **범용 플랫폼을 산업 파트너 생태계로 확장할 수 있는가?**

Google DeepMind:

> **로봇 지능 자체를 범용화할 수 있는가?**

NVIDIA:

> **Physical AI의 계산·시뮬레이션 기반을 플랫폼화할 수 있는가?**

이 질문들은 서로 배타적이지 않다.

최종적으로 합쳐질 수도 있다.

---

25. 그런데 태평양 건너편에서는 다른 실험이 동시에 진행 중이다

미국의 회사가

AI 모델을 개선하고,

새로운 손을 만들고,

대규모 투자를 받는 동안,

중국에서는 또 다른 숫자가 빠르게 움직인다.

부품 가격.

모터.

감속기.

센서.

배터리.

프레임.

완성 로봇.

그리고 무엇보다 **생산량**이다.

여기서 중국의 전략을 단순히 '카피해서 싸게 만든다'고 설명하면 중요한 것을 전부 놓친다.

제17장 마지막 챗화

실리콘밸리의 회의실.

AI 연구자

"더 큰 모델이 필요합니다."

로봇공학자

"더 좋은 손이 필요합니다."

제조 엔지니어

“부품 종류부터 줄입시다.”

투자자

“언제 10만 대 만듭니까?”

공장장

“그 전에 100대를 고장 없이 돌려봅시다.”

모두 잠시 조용해진다.

구석에 있던 로봇이 말한다.

> **“그래서 저한테 지금 무슨 일을 시키실 건가요?”**

그 질문이 모든 전략을 현실로 끌어내린다.

제18장 – 중국

싸게 만드는 것이 왜 AI 전략이 되는가

중국 휴머노이드 산업을 설명할 때 자주 등장하는 문장이 있다.

> **“미국은 기술, 중국은 물량.”**

절반쯤은 유용하지만, 그대로 받아들이면 산업의 핵심을 놓친다.

중국의 전략적 의미는 단순히 로봇을 싸게 만드는 데 있지 않다.

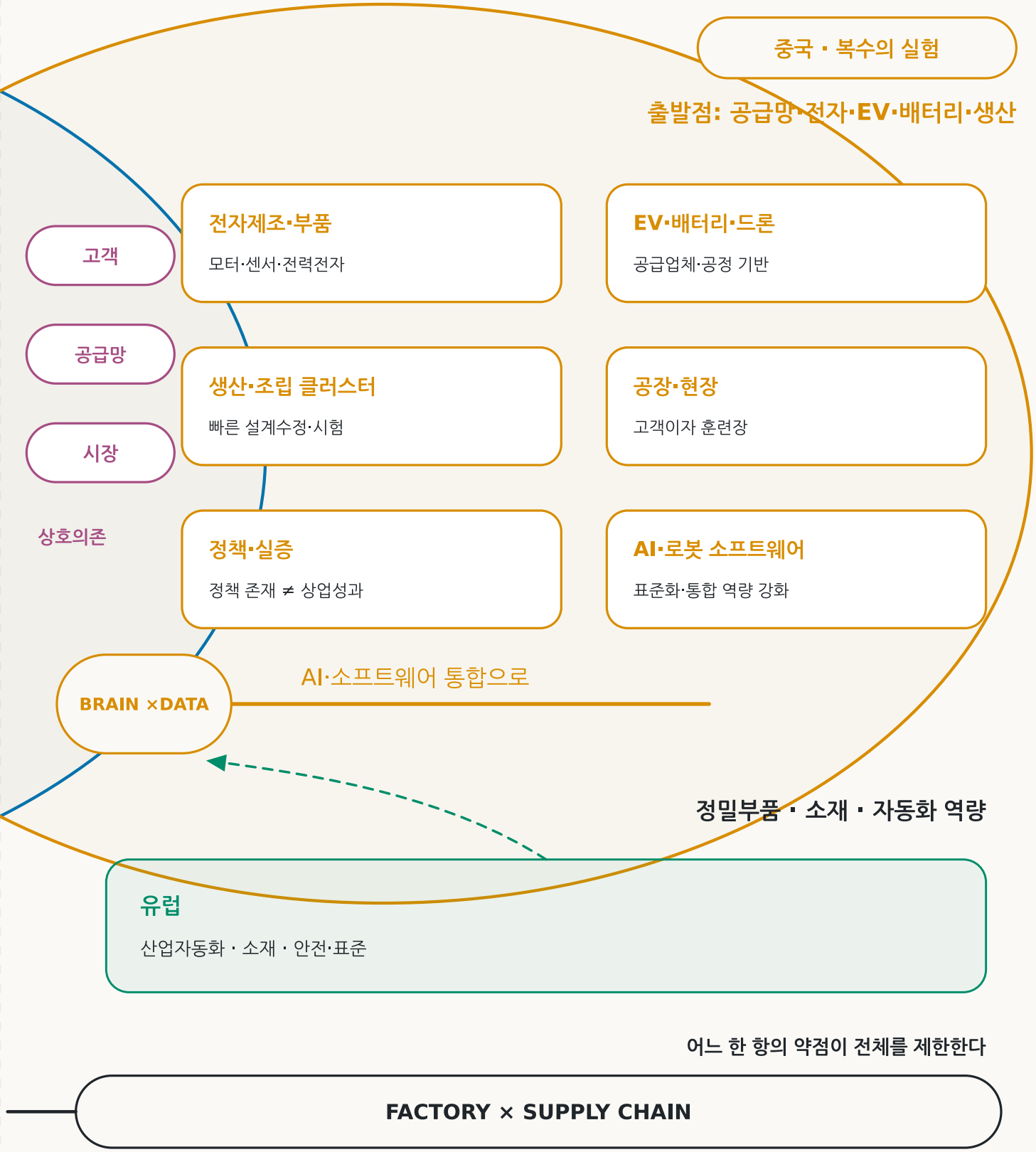
싸게 만든다 → 많이 만든다 → 많이 배치한다 → 더 많은 현실 데이터를 얻는다 → AI를 개선한다 → 로봇의 가치가 올라간다 → 다시 더 많이 만든다.

V15. 미국·중국과 연결 산업

"미국은 AI, 중국은 제조"라는 한 줄보다 결합구조와 이동 방향을 본다



PROJECT MODEL · 국가 평균 아님



어느 한 항의 약점이 전체를 제한한다

FACTORY × SUPPLY CHAIN

이 순환이 실제로 작동할 수 있느냐가 더 중요한 질문이다.

중국이 노리는 것이 단순한 **가격경쟁**에서 **학습속도 경쟁**으로 넘어간다면 이야기는 완전히 달라진다.

---

1. 먼저 중국을 하나의 회사처럼 보면 안 된다

중국에도 서로 다른 전략이 존재한다.

Unitree는 상대적으로 저렴하고 접근 가능한 하드웨어를 빠르게 시장에 내놓는 축에 가깝다.

UBTECH는 제조현장과 실제 산업 배치에 강하게 접근한다.

AgiBot(즈위안 로보틱스)은 다양한 휴머노이드·로봇 플랫폼과 대량생산을 동시에 추진하는 흐름에 속한다.

Fourier Intelligence는 재활로봇에서 축적한 기술을 범용 휴머노이드로 확장해 왔다.

즉 중국에서도

저가 플랫폼 / 산업 배치 / 범용 AI / 대량생산 / 전문영역 확장

이라는 서로 다른 가설이 경쟁한다.

2026년의 기업 숫자를 읽을 때도 범위를 먼저 확인해야 한다.

수치	뜻	뜻하지 않는 것
공개 시작가	기본 구성의 진입가격	완전한 작업 셀의 TCO
연간 생산능력	설비가 목표로 하는 최대치	실제 생산·판매·가동
누적 생산	라인에서 나온 수량	고객 출하·현장 가동
판매·출하	고객에게 넘어간 수량	생산적 작업의 반복 성공
현장 배치	고객 시설에 들어간 상태	유상 장기 상업운영

---

2. Unitree – 먼저 가격의 문턱을 부순다

Unitree의 전략적 의미는 꽤 크다.

이 회사는 사족보행 로봇에서 이미 비슷한 일을 했다.

과거 로봇개는 값비싼 연구장비였다.

가격을 크게 낮추면서

대학,

연구소,

개발자,

기업

이 실제 하드웨어를 만질 수 있게 했다.

휴머노이드에서도 비슷한 효과를 노릴 수 있다.

2026년 9월 공식 페이지 기준 G1의 시작가는 세금·배송비 제외 1만3,500달러다. R1 Air는 4,900달러, R1은 5,900달러부터다.

이 숫자는 저가 휴머노이드의 문턱이 실제로 낮아졌다는 현재 사실이다. 그러나 기본형·교육형·옵션 구성이 다르고, 자율 산업노동을 위한 통합비용은 포함하지 않는다.

---

3. 가격이 떨어지면 고객만 늘어나는 것이 아니다

개발자가 늘어난다.

연구가 늘어난다.

실패가 늘어난다.

응용프로그램이 늘어난다.

부품 수요가 늘어난다.

공급업체가 늘어난다.

그리고 다시 가격이 떨어질 가능성이 생긴다.

```
가격 ↓
 ↓
로봇 보급 ↑
 ↓
개발자 ↑
 ↓
실험·응용 ↑
 ↓
부품 생산량 ↑
 ↓
원가 ↓
 ←
```

이것은 PC와 드론에서 이미 본 산업적 순환이다.

---

4. 하지만 싼 로봇과 노동 로봇은 다르다

연구용 로봇을 싸게 만드는 것과

공장에서 3교대로 수천 시간 일하는 로봇을 싸게 만드는 것은 완전히 다른 문제다.

그래서 Unitree 같은 제품의 가격을 산업용 휴머노이드의 완성된 노동비용과 바로 비교하면 안 된다.

다시 우리의 다섯 질문을 적용한다.

몇 대인가?

무슨 일을 하는가?

몇 시간 하는가?

성공률은?

사람이 얼마나 개입하는가?

---

5. UBTECH – '멋진 로봇'보다 가동률

UBTECH의 Walker 계열에서 흥미로운 것은 공장 배치 지향성이다.

자동차 제조와 같은 실제 산업환경에서 휴머노이드를 시험하고 있다.

그리고 앞서 본 **배터리 자율교환** 같은 아이디어가 여기서 의미를 갖는다.

사람을 그대로 모방하려는 사고라면 묻는다.

> **"8시간 가는 배터리를 어떻게 만들지?"**

공장설비의 관점에서는 질문이 달라진다.

> **"충전 때문에 생산을 몇 분 멈추지?"**

이것이 산업로봇의 사고방식이다.

UBTECH의 2025년 연차실적 공시는 전신형 embodied humanoid 제품·서비스 매출 RMB 8.2억과 판매 1,079대를 제시했다. 2025년 말 기준 연간 생산능력은 6,000대를 넘었다고 보고했다.

이는 중국 기업의 상업 판매와 생산 확대를 보여 주는 강한 근거다. 그러나 판매 1,079대, 연간 생산능력 6,000대, 실제 고객 현장 가동대수는 서로 다른 수치다.

Walker S2의 약 3분 자율 배터리 교환과 24시간 연속운영 능력은 회사가 발표한 시스템 사양이다. 단일 배터리가 24시간 간다는 뜻도, 모든 고객 현장에서 그 가동률이 확인됐다는 뜻도 아니다.

---

6. 중국의 가장 큰 실험장은 공장이다

가정용 로봇을 처음부터 만들기 어려운 이유를 이미 봤다.

집은 비정형적이다.

공장은 상대적으로 구조화할 수 있다.

그리고 중국에는 거대한 제조업 기반이 있다.

자동차.

전자제품.

배터리.

물류.

금속가공.

이 공장들은 로봇의 고객인 동시에 **훈련장**이 될 수 있다.

---

7. 여기서 중국의 EV 산업이 다시 등장한다

휴머노이드 한 대를 뜯어보면 전기차와 겹치는 것이 많다.

영구자석.

모터.

배터리.

BMS.

전력전자.

열관리.

센서.

제조공정.

물론 자동차 부품을 그대로 로봇에 넣는 것은 아니다.

하지만 공급업체·소재·생산기술이라는 **산업적 기반**은 공유된다.

---

8. 드론 산업까지 붙는다

중국은 세계적인 드론 제조 생태계도 갖고 있다.

드론은 이미

경량 모터,

IMU,

배터리,

실시간 제어,

카메라,

전자부품

을 엄청난 수량으로 생산해왔다.

이 기술들이 휴머노이드에 그대로 복사되는 것은 아니지만 공급망과 제조역량이 겹친다.

그래서 중국의 휴머노이드 산업은 갑자기 맨땅에서 생긴 것이 아니다.

> **EV + 드론 + 산업로봇 + 전자제조 + AI가 합류한 결과**

로 보는 편이 정확하다.

---

9. 감속기 국산화가 중요한 이유

일본계 전문기업들은 중대형 산업로봇용 정밀감속기와 스트레인 웨이브 감속기의 주요 공급축으로 남아 있다. 동시에 이들 기업의 공시에서도 중국 생산거점 확대와 중국 현지 로봇 제조사의 성장이 확인된다.

따라서 이 변화를 '완전히 따라잡았느냐'라는 국가 서열의 질문으로만 보면 안 된다.

더 중요한 변화는

쓸 수 있는 품질의 제품을 더 낮은 가격과 큰 생산량으로 공급할 수 있는가

이다.

휴머노이드 시장이 커질수록 이 차이가 중요해진다.

---

10. 자동차에서 이미 본 패턴이다

초기 중국 자동차 산업은 종종

'저가 제품'

으로 평가됐다.

그런데 시장이 커지면서

생산량 증가

↓

공급업체 성장

↓

공정 개선

↓

품질 향상

↓

가격 경쟁력 강화

↓

다시 생산량 증가

라는 학습곡선이 만들어졌다.

휴머노이드에서도 비슷한 제조학습곡선이 나타날지가 중요한 관찰 대상이다.

---

11. Learning Curve

생산량이 누적되면 제조업체는 배운다.

어떤 볼트가 자주 풀리는가.

어떤 케이블이 끊어지는가.

어떤 감속기가 불량인가.

어떤 조립공정에서 편차가 생기는가.

어떤 부품을 없앨 수 있는가.

따라서 대량생산은 단순히 규모경제만 만드는 것이 아니다.

제조 데이터

를 만든다.

---

12. 여기서 두 종류의 데이터가 합쳐진다

우리는 지금까지 AI 데이터만 생각했다.

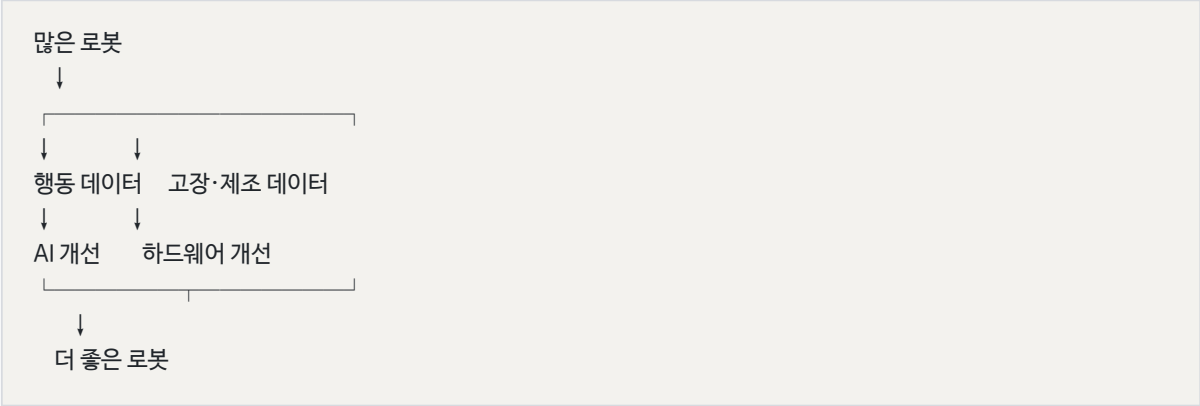
하지만 로봇에는 두 종류의 데이터가 있다.

행동 데이터

어떻게 걷고 잡고 일하는가.

제조·신뢰성 데이터

어디가 고장 나고 어떻게 싸게 만들 수 있는가.



이 두 플라이휠을 동시에 돌릴 수 있다면 강력하다.

---

13. 이것이 '물량'이라는 말의 진짜 의미다

100만 대를 만들었다고 자동으로 좋은 회사가 되는 것은 아니다.

하지만 100만 대가 실제 현장에서 움직이며 데이터를 보내고,

그 데이터를 이용해

AI와 하드웨어를 개선할 수 있다면 이야기가 달라진다.

Quantity → Learning

으로 변환된다.

중국 휴머노이드 산업을 볼 때 가장 중요한 관찰점 가운데 하나다.

---

14. 그러나 이 연결은 자동으로 일어나지 않는다

여기서 영상의 '중국은 많이 만들어서 데이터를 얻는다'는 설명을 보정해야 한다.

로봇이 많다고 좋은 데이터가 자동으로 생기는 것은 아니다.

필요한 것이 있다.

센서 표준화.

데이터 기록.

클라우드 인프라.

학습 파이프라인.

데이터 품질관리.

모델 업데이트.

안전검증.

즉 **제조역량** + AI **인프라**가 함께 있어야 한다.

---

15. 값싼 하드웨어만으로는 Physical AI를 지배할 수 없다

10만 대의 로봇이 있다.

하지만 서로 다른 센서를 사용한다.

데이터 형식이 다르다.

기록이 불완전하다.

행동 라벨이 없다.

실패원인을 모른다.

그러면 데이터 자산의 가치가 크게 떨어진다.

따라서 중국 기업에도 소프트웨어 표준화와 AI 통합은 중요한 도전이다.

---

16. 반대로 AI 모델만으로도 지배할 수 없다

최고의 VLA를 가지고 있다.

그런데 실제 로봇이 100대뿐이다.

현실에서 새로운 실패를 거의 경험하지 못한다.

하드웨어도 너무 비싸다.

그러면 데이터 플라이휠이 느려진다.

다시 우리의 공식이다.

> BODY × BRAIN × DATA × FACTORY × SUPPLY CHAIN

---

17. AgiBot – 생산량 자체를 실험한다

중국의 여러 신규 휴머노이드 업체 가운데 AgiBot 같은 기업이 흥미로운 이유는 단순한 로봇 동작보다 **제품군과 생산 확대**를 동시에 강조해 왔다는 점이다.

하나의 완벽한 로봇을 수년 동안 만드는 방식과

여러 플랫폼을 실제로 생산하며 빠르게 개선하는 방식은 다른 산업철학이다.

전자는 깊이를 얻을 수 있다.

후자는 제조학습 속도를 얻을 수 있다.

어느 쪽이 장기적으로 유리한지는 아직 열린 질문이다.

AgiBot은 2026년 3월 1만 번째, 6월 1만5천 번째 로봇이 생산라인에서 나왔다고 발표했다. 이것은 주목할 만한 **기업 발표상의 누적 생산 이정표**다.

그러나 1만5천을 곧바로 전신형 산업 휴머노이드의 고객 가동대수로 읽을 수는 없다. 여러 모델을 포함하는 제품군 범위, 실제 출하, 설치, 활성 가동을 다시 나눠야 한다.

회사가 인용한 2025년 5,168대 출하·39% 점유율은 Omdia의 **분석가 추정**이다. 기업 공시나 고객의 운영 기록과 같은 등급으로 취급하지 않는다.

판정: BATCH PRODUCTION / DEPLOYMENT CLAIM. MASS SCALE은 **보류**.

---

18. Fourier – 전문로봇에서 범용로봇으로

재활로봇 분야에서 축적된

액추에이터,

힘 제어,

인간-로봇 접촉,

안전

기술은 휴머노이드와 상당 부분 연결된다.

이 경로도 중요하다.

모든 휴머노이드 기업이 자동차나 AI에서 출발할 필요는 없다.

기존 로봇 전문영역 → 범용 휴머노이드

라는 진화경로도 존재한다.

다만 Fourier의 현재 제품과 전략 방향은 확인되지만, 이번 1차 자료 범위에서는 신뢰할 수 있는 누적 생산·출하·공개 판매가격·장기 상업운영 수치를 확인하지 못했다.

따라서 Fourier는 규모 경쟁의 근거가 아니라 **전문로봇의 기술이 범용·케어 휴머노이드로 이동하는 경로**로만 사용한다. 현재 공개 근거만으로는 양산·상업운영 단계를 확정하지 않는다.

---

19. 중국 정부와 지방정부도 변수다

중국 공업정보화부는 2023년 「인형로봇 혁신발전 지도 의견」을 공식 발표했다. 따라서 휴머노이드가 국가 산업정책의 대상이라는 점은 현재 사실이다.

다만 정책 문서의 존재, 지방정부의 지원, 실제 예산 집행, 기업 매출과 생산적 고객수요는 서로 다른 층이다. 이번 P1에서는 중앙 정책의 존재까지만 확인했으며, 지역별 지원액·공공조달 규모·정책효과는 검증 가능한 원문이 없는 한 숫자로 쓰지 않는다.

동시에 다른 위험도 생긴다.

---

20. 너무 많은 돈이 너무 빨리 들어오면

기업이 너무 많이 생긴다.

비슷한 제품이 쏟아진다.

실제 수요보다 생산능력이 앞설 수 있다.

보조금에 의존하는 매출이 생길 수 있다.

과도한 밸류에이션도 나타날 수 있다.

즉 중국의 제조속도는 강점인 동시에 **과잉투자의 위험**을 가진다.

그래서 출하량만으로 산업건전성을 판단하면 안 된다.

---

21. 진짜 수요와 정책 수요를 구분해야 한다

기업이 로봇을 산 이유가

실제로 인간 노동보다 경제적이어서인가?

정부 지원을 받기 위해서인가?

실증 프로젝트라서인가?

홍보 목적인가?

연구용인가?

완전히 다르다.

따라서 앞으로 중국의 휴머노이드 판매량을 볼 때도

판매 = 생산적 노동 투입

이라고 가정하지 않는다.

---

22. 여기서 미국과 중국의 차이를 다시 보자

초기의 단순한 그림:

```
미국 = AI
중국 = 제조
```

우리가 18장까지 와서 얻은 그림은 훨씬 복잡하다.

```
미국
AI·컴퓨팅·자본·로봇공학
+ 일부 강력한 제조 플랫폼

        ↕ 경쟁·협력

중국
공급망·대량생산·EV·전자제조
+ 빠르게 성장하는 AI·로봇 소프트웨어
```

그리고 일본·한국·유럽의 부품과 산업기술이 그 사이에 연결된다.

---

23. 결국 두 시스템은 서로의 강점을 향해 움직인다

미국 로봇 기업:

제조규모를 확보하려 한다.

중국 로봇 기업:

AI 능력을 높이려 한다.

즉 서로 반대 방향에서 출발해 가운데로 접근한다.

```
AI ──────────────────────→ 제조

미국      ×      중국

제조 ←────────────────────── AI
```

결국 승부처는 가운데다.

지능을 가진 기계를 대량으로 만드는 능력.

---

24. 그런데 아직 양쪽 모두 해결하지 못한 적이 있다

로봇이 공장에 들어갔다.

처음 100시간은 잘 움직였다.

500시간.

손가락 케이블이 마모된다.

1,000시간.

감속기에 유격이 생긴다.

배터리 용량이 줄어든다.

센서가 드리프트한다.

소프트웨어 업데이트 뒤 예상하지 못한 오류가 생긴다.

그리고 어느 날,

사람이 옆을 지나가는데 로봇이 잘못 움직인다.

이제 질문은 기술도, 가격도 아니다.

믿을 수 있는가?

---

25. 휴머노이드의 마지막 기술장벽은 신뢰일 수 있다

자동차가 매주 멈춘다면 아무리 싸도 팔리지 않는다.

엘리베이터가 99% 안전하다면 아무도 타지 않는다.

70kg짜리 로봇이 사람 옆에서 일한다면 요구되는 신뢰성은 더욱 높아진다.

그래서 다음 장에서는 처음으로

고장과 사고

를 책의 중심에 놓는다.

제18장 마지막 챗화

중국 제조 엔지니어

“우리는 10만 대 만들 수 있습니다.”

AI 엔지니어

“그럼 엄청난 데이터를 얻겠군요.”

공장장

“잠깐.”

둘:

“왜요?”

공장장:

> **“10만 대가 매일 한 번씩 고장 나면 누가 고칩니까?”**

조용해진다.

뒤에서 서비스 엔지니어가 손을 든다.

> **“드디어 제 이야기를 할 차례군요.”**

PART V

제품이 되는 마지막 벽

제품의 마지막 벽은 반복생산·고장·수명이다.

제19장 – 고장 나지 않는 로봇

99.9%가 왜 충분하지 않을 수 있는가

휴머노이드가 걷는다.

손도 잘 쓴다.

말도 알아듣는다.

가격도 내려갔다.

공장에 배치했다.

이제 성공한 것처럼 보인다.

그런데 진짜 제품 시험은 바로 그날부터 시작된다.

> **내일도 되는가?
> 6개월 뒤에도 되는가?
> 1만 번째 동작에서도 되는가?**

연구실에서는 **가능성**이 중요했다.

산업에서는 **반복 가능성**이 중요하다.

---

1. 99.9%는 굉장히 좋은 숫자처럼 보인다

로봇이 부품 하나를 집어 옮기는 작업을 한다.

성공률:

99.9%.

거의 완벽하다.

그런데 하루에 10,000번 한다고 하자.

단순 기대값으로 보면

약 10번의 실패

가 발생한다.

공장에서는 질문이 달라진다.

그 10번이 무엇인가?

부품을 떨어뜨리는가?

사람이 와서 다시 놓아야 하는가?

생산라인이 멈추는가?

다른 제품을 파손시키는가?

사람을 다치게 할 가능성이 있는가?

같은 0.1%라도 결과가 전혀 다르다.

---

2. 그래서 실패율 하나만 보면 안 된다

두 종류의 실패를 비교해보자.

실패 A

컵을 놓쳤다.

로봇이 스스로 다시 집는다.

10초 손실.

실패 B

컵을 놓치며 손가락이 부러졌다.

사람이 와야 한다.

수리 2시간.

같은 '1회 실패'가 아니다.

따라서 신뢰성을 이해하려면

실패 빈도 × 실패의 심각도 × 복구시간

을 함께 봐야 한다.

---

3. MTBF – 얼마나 자주 고장 나는가

MTBF – Mean Time Between Failures.

정해진 운전조건에서 관측한 수리 가능한 시스템의 고장 사이 평균 운전시간이다. 개별 로봇이 그 시간까지 고장 나지 않는다는 보증수명이 아니다. 어떤 고장을 세었는지, 표본과 운전조건이 무엇인지 함께 제시해야 한다.

---

4. MTTR – 작업으로 돌아오는 데 얼마나 걸리는가

MTTR은 문헌과 조직에 따라 Mean Time To Repair 또는 복구시간에 가까운 뜻으로 쓰인다. 이 책에서는 **고장 감지 후 진단·수리·시험을 거쳐 작업 가능한 상태로 복귀하기까지의 평균시간**으로 정의한다. 부품 물류와 대기시간을 포함하는지는 별도로 표시한다.

---

5. 그래서 모듈화가 경제성이 된다

손이 고장 났다.

손만 교환한다.

무릎 액추에이터가 고장 났다.

액추에이터 모듈만 교환한다.

배터리가 열화됐다.

팩만 교체한다.

```
고장
 ↓
진단
 ↓
문제 모듈 식별
 ↓
모듈 교체
 ↓
자동 보정
 ↓
작업 복귀
```

좋은 산업용 휴머노이드는 단순히 **잘 안 고장 나는 로봇**이 아니다.

> **고장 나도 빨리 돌아오는 로봇이다.**

---

6. 첫 번째 적 – 감속기

우리는 3장에서 플렉스 스플라인을 계속 구부렸다.

수백만 번 반복하면 피로가 누적될 수 있다.

기어가 마모된다.

윤활상태가 변한다.

백래시가 증가할 수 있다.

그러면 처음에는 손끝 오차가 거의 없던 로봇이 시간이 지나면서 조금씩 부정확해질 수 있다.

고장은 반드시

어느 날 갑자기 멈추는 것

만을 뜻하지 않는다.

성능이 천천히 무너지는 것도 고장이다.

---

7. 두 번째 적 – 베어링

베어링은 계속 하중을 받는다.

걷는다.

뛴다.

상자를 든다.

넘어진다.

충격이 반복된다.

마모가 증가하면

진동,

소음,

유격,

마찰

이 달라질 수 있다.

그리고 그 변화는 다른 부품에 다시 영향을 준다.

---

8. 세 번째 적 – 케이블

의외로 중요한 적이다.

관절이 굽혀진다.

펴진다.

굽혀진다.

펴진다.

그 안의 전선도 계속 움직인다.

수백만 회 반복하면 피로가 누적될 수 있다.

특히 손과 손목처럼 공간이 좁고 자유도가 높은 곳은 배선 설계가 까다롭다.

최첨단 AI 로봇이 **전선 하나 때문에 멈출 수 있다.**

---

9. 네 번째 적 – 손

손은 외부세계와 가장 많이 접촉한다.

부딪힌다.

긁힌다.

물건을 놓친다.

먼지가 들어간다.

손가락 끝이 마모된다.

따라서 상용 휴머노이드에서는 손을 '영구적인 신체기관'보다 **소모·정비 가능한 작업도구**로
설계하는 편이 합리적일 수 있다.

---

10. 다섯 번째 적 – 배터리

배터리는 충·방전을 반복하면서 열화된다.

처음:

5시간.

시간이 흐른 뒤:

4시간 30분.

더 지나면:

4시간.

그렇다면 로봇이 고장 난 것인가?

아니다.

하지만 생산능력은 떨어졌다.

따라서 로봇의 건강상태에는

배터리 SOH – State of Health

도 들어간다.

---

11. 여섯 번째 적 – 센서 드리프트

센서가 완전히 죽지 않아도 문제다.

조금씩 틀릴 수 있다.

힘센서가 실제보다 2% 높게 측정한다.

IMU의 오차가 누적된다.

카메라 위치가 충격으로 미세하게 변한다.

그러면 AI는 잘못된 세계를 보게 된다.

> **센서 오류는 로봇에게 잘못된 현실을 보여준다.**

---

12. 그래서 Calibration이 필요하다

로봇은 자기 몸을 주기적으로 다시 측정할 수 있어야 한다.

관절 영점.

카메라 위치.

힘센서.

손끝 위치.

여기서 상용화의 중요한 질문이 생긴다.

보정을 엔지니어가 해야 하는가?

아니면

로봇이 스스로 할 수 있는가?

후자가 가능할수록 운영비가 내려간다.

---

13. 일곱 번째 적 – 소프트웨어

하드웨어가 완벽해도 소프트웨어가 문제를 일으킬 수 있다.

새로운 AI 모델을 업데이트했다.

컵 집기 성공률은 올라갔다.

그런데 예상하지 못한 상황에서 문을 여는 행동이 나빠졌다.

이것이 학습 기반 시스템의 까다로운 점이다.

한 능력을 개선했는데 다른 능력이 퇴행할 수 있다.

Regression이다.

---

14. 그래서 로봇 업데이트는 스마트폰 업데이트와 다르다

스마트폰 앱이 충돌한다.

불편하다.

휴머노이드 행동모델이 잘못 업데이트된다.

물리적 사고로 이어질 수 있다.

따라서 업데이트 전에

시뮬레이션,

회귀시험,

안전검증,

제한된 로봇 배포

등의 절차가 중요해진다.

```
새 모델
 ↓
Simulation Test
 ↓
Safety Test
 ↓
소규모 Fleet
 ↓
이상 없음
 ↓
점진적 확대
```

15. 여기서 자동차 산업이 다시 선생님이 된다

자동차도

기계,

전자장치,

소프트웨어,

센서,

통신

이 결합된 안전 중요 시스템이다.

그래서 휴머노이드가 대량보급될수록 자동차·항공·산업자동화 분야에서 축적된

기능안전,

고장진단,

중복설계,

품질관리

경험이 중요해질 가능성이 크다.

---

16. Fail-safe – 실패했을 때 무엇을 하는가

모터가 이상하다.

센서가 죽었다.

통신이 끊겼다.

AI가 이상한 행동을 명령한다.

좋은 시스템은 이런 상황에서

무엇을 하지 말아야 하는지

알아야 한다.

예:

속도를 낮춘다.

물체를 안전하게 내려놓는다.

정지한다.

사람에게 알린다.

안전 자세로 이동한다.

---

17. 모든 고장에 '그냥 멈춰'도 답은 아니다

로봇이 20kg 물체를 머리 위로 들고 있다.

센서 오류.

즉시 모든 모터 전원을 끈다.

무슨 일이 일어날까?

물체가 떨어질 수 있다.

따라서 안전정지는 상황에 따라 설계되어야 한다.

> **안전하게 멈추는 것도 하나의 행동이다.**

---

18. 사람과 함께 일하면 문제가 더 어려워진다

산업용 로봇은 오랫동안 울타리 뒤에 있었다.

사람이 들어오지 않는다.

휴머노이드의 목적은 반대다.

사람이 쓰는 공간에 들어간다.

복도.

공장.

창고.

언젠가는 가정.

따라서 로봇과 사람 사이의 **물리적 거리 자체가 줄어든다.**

---

19. 70kg의 몸이 움직인다는 것

로봇 팔이 빠르게 움직인다.

사람의 머리와 충돌한다.

이때 중요한 것은 AI가 '사람'을 인식했느냐만이 아니다.

속도.

질량.

관절 강성.

충돌 위치.

제어기의 반응시간.

이 모든 것이 부상위험을 결정한다.

그래서 사람과 협업하는 로봇에는

힘 제한,

속도 제한,

충돌 감지,

안전영역

등이 중요해진다.

---

20. 안전에는 층이 있어야 한다

```
1. AI
"사람을 피한다"
      ↓
2. 행동계획
안전 경로
      ↓
3. 안전제어
속도·힘 제한
      ↓
4. 하드웨어
브레이크·토크 제한
      ↓
5. 비상정지
최종 방어선
```

위의 AI가 실패해도 아래층이 사람을 보호해야 한다.

안전은 AI의 선의를 믿는 구조가 아니라 실패를 전제로 한 구조여야 한다.

---

21. 그리고 사이버보안 문제가 등장한다

인터넷에 연결된 노트북이 해킹된다.

정보가 유출될 수 있다.

인터넷에 연결된 70kg 휴머노이드가 해킹된다.

문제가 달라진다.

카메라가 있다.

마이크가 있다.

팔이 있다.

다리가 있다.

문을 열 수도 있다.

즉 사이버보안과 물리안전이 하나의 문제가 된다.

---

22. 공격자는 AI 모델만 노릴 필요도 없다

통신.

업데이트 서버.

로봇 계정.

센서.

Fleet 관리 시스템.

공급망 소프트웨어.

충전 인프라.

공격표면이 넓어진다.

따라서 대규모 로봇 Fleet에서는

인증,

암호화,

권한분리,

보안 업데이트,

로그,

비상 차단

등이 중요해진다.

---

23. 그리고 프라이버시

가정용 휴머노이드를 생각해보자.

로봇은 집 안을 본다.

사람 목소리를 듣는다.

가족의 생활패턴을 안다.

집 구조를 안다.

물건 위치를 안다.

휴머노이드가 가정에 들어오는 순간

카메라 달린 스마트폰보다 훨씬 깊은 생활 데이터를 다룰 수 있다.

따라서 가정용 로봇의 병목은 손기술만이 아니다.

신뢰다.

---

24. 사고가 나면 누가 책임지는가

로봇이 사람과 충돌했다.

원인은 무엇인가?

센서 제조사?

로봇 제조사?

VLA 모델?

소프트웨어 업데이트?

사용자?

정비업체?

공장 운영자?

휴머노이드가 여러 회사의 하드웨어·AI·서비스를 결합할수록 책임구조가 복잡해진다.

그래서 보험과 법적 책임도 상용화 비용의 일부가 된다.

---

25. 신뢰성은 결국 돈으로 돌아온다

고장률 ↑

↓

정비비 ↑

↓

다운타임 ↑

↓

인간 개입 ↑

↓

보험료 ↑

↓

생산량 ↓

↓

성공한 행동 하나의 가격 ↑

즉 신뢰성은 기술적 미덕이 아니다.

경제 변수다.

---

26. 예지정비가 중요한 이유

로봇은 자기 몸에서 엄청난 데이터를 만든다.

모터 온도.

전류.

진동.

감속기 마찰.

배터리 내부저항.

관절 오차.

이것을 장기간 기록하면 고장 전의 패턴을 찾을 수 있다.

```
정상
 ↓
미세 진동 증가
 ↓
전류 패턴 변화
 ↓
마찰 증가
 ↓
AI 이상 탐지
 ↓
"고장 전에 교체"
```

고장 난 뒤 고치는 것에서

고장 나기 전에 고치는 것

으로 이동한다.

---

27. 고장 데이터는 신뢰성 플라이휠을 만든다

여러 현장 로봇의 고장·정비·교체 기록을 같은 분류체계로 모으면 설계와 예방정비를 개선할 수 있다. 행동 데이터와 공정 데이터, 고장 데이터는 서로 다른 학습계다.

세 플라이휠의 정식 정의는 20장에서 한 번만 제시한다.

---

28. 그리고 마지막 제조장벽이 남았다

좋다.

이제 로봇은

똑똑하다.

싸다.

안전하다.

고장도 적다.

그런데 주문이 100만 대 들어왔다.

만들 수 있는가?

연구실에서 10대 만들 때와

공장에서 100만 대 만들 때는 완전히 다른 기술이 필요하다.

부품 공용화.

금형.

자동조립.

수율.

품질관리.

공급망.

검사 자동화.

서비스망.

이제 우리는 **로봇을 만드는 로봇**의 세계로 간다.

제19장 마지막 챗화

AI 연구자

“우리 로봇 성공률 99.9%입니다.”

공장장

“좋습니다.”

신뢰성 엔지니어

“하루 작업 횟수는요?”

“10,000번.”

“그럼 실패가 몇 번이죠?”

잠시 침묵.

AI 연구자:

“평균적으로… 10번.”

신뢰성 엔지니어:

“그 10번 중 하나가 사람과 충돌하면요?”

더 긴 침묵.

그가 비상정지 버튼을 가리킨다.

> **“지능이 높아질수록 실패를 가정해야 합니다.”**

제20장 – 로봇 공장

로봇을 10대 만드는 것과 100만 대 만드는 것은 다른 기술이다

휴머노이드를 완성했다.

잘 걷는다. 손도 쓴다. AI도 좋다. 안전성도 개선했다.

투자자가 묻는다.

> **"좋습니다. 내년에 10만 대 만들어 주세요."**

갑자기 회사 전체가 조용해진다.

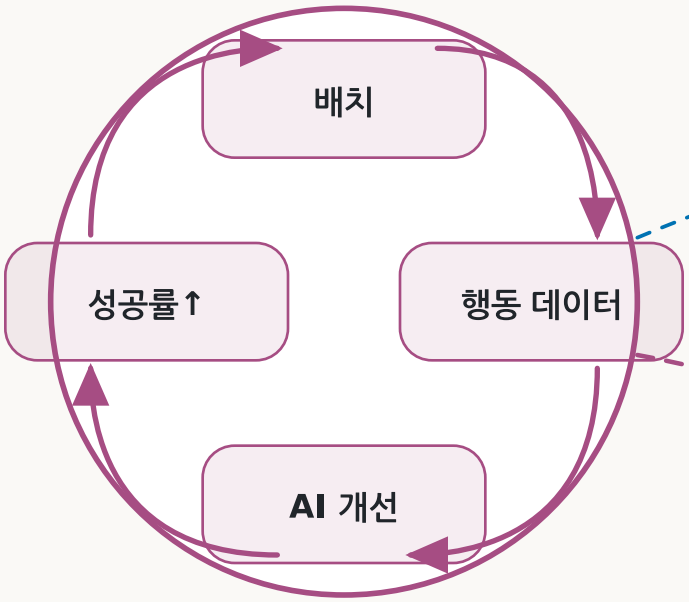
다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

V16. 세 개의 플라이휠

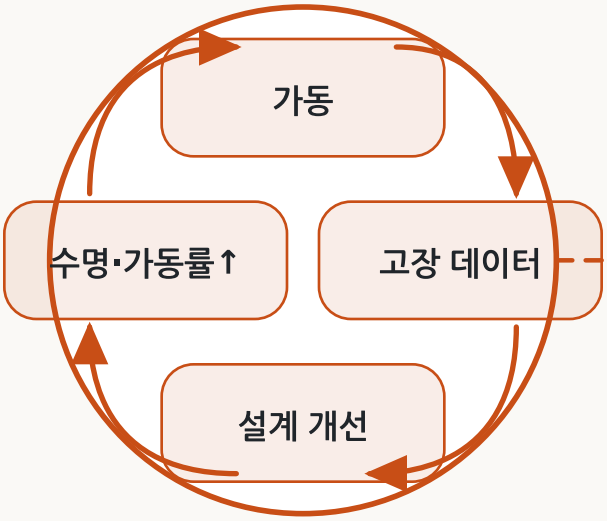
잘 행동하는 법·싸게 만드는 법·오래 쓰는 법은 서로 다른 데이터에서 배운다

현장 데이터는 행동과 신뢰성을 함께 키운다

① 행동 플라이휠



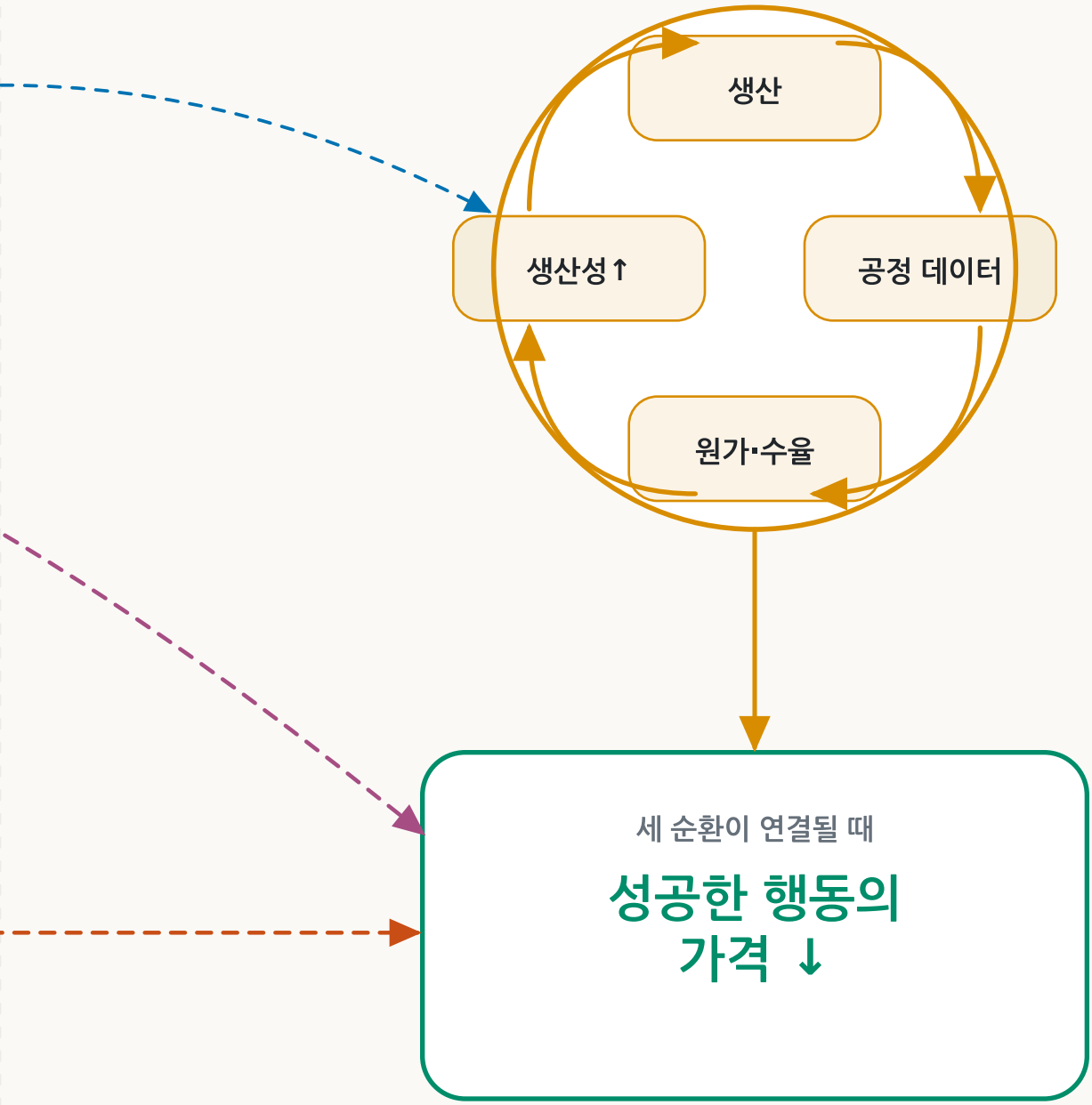
③ 신뢰성 플라이휠



공통 조건 · 데이터 품질 · 빠른 피드백

공정 데이터는 원가와 수율을 함께 바꾼다

② 제조 플라이휠



운영 조건 · 설계 반영 · 현장 운영 역량

연구실에서 최고의 로봇 한 대를 만드는 것과, 공장에서 **똑같은 로봇 10만 대를 만드는 것**은 완전히 다른 문제이기 때문이다.

휴머노이드 산업의 다음 장벽은 더 좋은 AI가 아닐 수도 있다.

제조공학이다.

---

1. 연구실의 목표와 공장의 목표는 다르다

연구실:

> **"이 로봇이 작동하는가?"**

공장:

> **"모든 로봇이 똑같이 작동하는가?"**

첫 번째 로봇은 엔지니어가 직접 조정하면 된다.

두 번째도 가능하다.

100번째부터 힘들어진다.

10만 번째에는 불가능하다.

제품이 되려면 **사람의 장인기술을 공정으로 바꿔야 한다.**

---

2. Prototype은 비싸도 된다

연구용 로봇은 CNC로 부품을 깎는다.

비싼 센서를 산다.

케이블을 손으로 정리한다.

엔지니어가 하나씩 보정한다.

볼트 하나가 불편하면 사람이 알아서 조립한다.

목표는 하나다.

일단 움직여라.

하지만 양산제품에서는 이 모든 것이 문제가 된다.

---

3. 양산제품의 질문

볼트가 몇 개인가?

조립시간은?

공구는 몇 종류인가?

케이블은 자동으로 넣을 수 있는가?

액추에이터 종류는 몇 개인가?

부품을 거꾸로 끼울 가능성은?

검사는 자동화할 수 있는가?

수리할 때 몇 분 걸리는가?

즉 설계 기준 자체가 바뀐다.

---

4. Design for Manufacturing

좋은 로봇을 만든다.

에서

잘 만들 수 있는 로봇을 설계한다.

로 이동한다.

이를 넓게 **제조용이성 설계**(DFM)라고 생각할 수 있다.

```
좋은 성능
  +
적은 부품
  +
쉬운 조립
  +
높은 수율
  +
쉬운 검사
  +
쉬운 수리
  =
양산 가능한 제품
```

---

5. 부품 하나를 싸게 만드는 것보다 없애는 것이 낫다

휴머노이드에 볼트가 1,000개 있다고 가정해보자.

볼트 하나는 싸다.

하지만 각각

구매하고,

보관하고,

운반하고,

조립하고,

토크를 검사해야 한다.

10만 대라면?

1억 번의 볼트 조립이다.

그래서 제조공학자는 묻는다.

> **"이 볼트가 정말 필요한가?"**

---

6. 이것이 부품통합이다

과거:

부품 A + 브래킷 + 볼트 4개 + 부품 B.

새 설계:

A와 B를 하나의 구조물로 통합.

부품 수 감소.

↓

조립시간 감소.

↓

고장점 감소.

↓

재고 감소.

↓

검사 감소.

↓

원가 감소.

하나의 설계 변경이 여러 비용을 동시에 없앤다.

---

7. 액추에이터도 표준화한다

로봇에 관절이 40개 있다고 해서 서로 다른 액추에이터 40종을 만들면 생산이 복잡해진다.

가능하면

소형,

중형,

대형

몇 종류로 묶는다.

그리고 여러 관절에서 공용한다.

```
40 관절
  ↓
40 종류 액추에이터  ×

40 관절
  ↓
6~10종 등 제한된 모듈
  ↓
공용화
```

실제 최적 개수는 로봇 설계마다 다르지만 원칙은 같다.

종류를 줄이면 규모가 생긴다.

---

8. 공용화가 왜 그렇게 강력한가

같은 모터를 많이 산다.

단가가 내려간다.

같은 감속기를 많이 만든다.

공정이 안정된다.

같은 액추에이터를 반복 조립한다.

작업자가 숙련된다.

자동화하기 쉬워진다.

고장 나도 예비부품 종류가 줄어든다.

즉 공용화는

제조 + 구매 + 정비

세 곳에서 동시에 돈을 아낀다.

---

9. 이제 CNC에서 금형으로 이동한다

연구 단계에서는 CNC가 훌륭하다.

설계를 바꾸기 쉽다.

하지만 한 개씩 깎기 때문에 생산량이 커지면 비싸질 수 있다.

대량생산이 확실해지면

주조,

단조,

프레스,

사출,

전용 금형

같은 방식이 경제성을 갖기 시작한다.

초기 설비비는 크지만 개당 비용을 낮출 수 있다.

---

10. 그래서 생산량이 늘면 로봇 자체의 모습도 바뀐다

흔히 이렇게 생각한다.

> **"현재 로봇을 그대로 100만 개 복사하면 대량생산이다."**

아니다.

100만 대를 만들기 위해 **로봇을 다시 설계**할 가능성이 크다.

이것이 중요한 차이다.

Prototype → Mass Product는 단순 복사가 아니다.

재설계다.

---

11. Tesla가 계속 등장하는 이유

Tesla의 Optimus 전략에서 산업적으로 흥미로운 부분이 바로 이것이다.

Tesla는 자동차를 대량생산하면서

부품통합,

공정자동화,

공급망,

전력전자,

모터,

배터리,

생산설비

를 다뤄왔다.

따라서 Optimus의 핵심 시험 가운데 하나는

> **자동차에서 축적한 '양산을 위한 설계'를 휴머노이드로 옮길 수 있는가**

라는 것이다.

2026년 1분기 현재 Optimus 생산시설은 아직 건설 단계였다. Fremont 연 100만 대, Texas 장기 연 1,000만 대는 Tesla가 밝힌 라인의 설계 용량이다.

따라서 이 사례는 Tesla의 제조 경험이 왜 중요한지를 보여 주지만, Optimus의 현재 양산 성공을 보여 주는 실적은 아니다.

---

12. 중국의 제조 클러스터도 같은 문제를 다른 방식으로 푼다

중국 업체는 거대한 부품 생태계를 활용할 수 있다.

자체 생산하지 않아도 근처 공급업체와 빠르게 반복한다.

샘플.

수정.

재생산.

그리고 생산량이 커지면 공급업체 자체가 전용라인을 만든다.

즉

수직통합

과

고밀도 공급망

이라는 서로 다른 제조전략이 경쟁한다.

---

13. 어느 것이 더 좋은가

답은 없다.

수직통합:

통제력이 높다.

공동최적화가 쉽다.

하지만 막대한 자본이 필요하다.

공급망 활용:

전문기업의 기술을 이용한다.

유연하다.

하지만 공급업체 조율과 병목 문제가 생긴다.

실제 기업들은 둘을 섞는다.

핵심은 '전부 직접 만드느냐'가 아니라

> **어떤 부품을 직접 통제해야 경쟁력이 생기는가**

라는 질문이다.

---

14. 그리고 공장의 가장 무서운 숫자가 등장한다

수율

부품 100개 생산.

98개 합격.

수율 98%.

좋아 보인다.

그런데 로봇 한 대에 중요한 부품이 수백 개 들어간다면?

각 부품과 공정의 작은 불량이 누적될 수 있다.

따라서 복잡한 제품에서는 높은 공정능력이 필수다.

---

15. 조립됐다고 출하할 수도 없다

로봇이 완성됐다.

전원을 켠다.

각 관절 검사.

센서 검사.

카메라 보정.

손 검사.

보행 검사.

배터리 검사.

통신 검사.

안전장치 검사.

필요하다면 일정 시간의 Burn-in 테스트.

한 대 검사에 4시간이 걸린다면 10만 대에서는 거대한 검사설비가 필요하다.

---

16. 그래서 검사 자체를 설계한다

제품 설계 단계부터

자동으로 검사할 수 있게 만든다.

센서가 스스로 진단한다.

모터가 자동 테스트한다.

관절이 지정된 패턴으로 움직인다.

AI가 이상한 진동을 탐지한다.

```
조립 완료
 ↓
Self Test
 ↓
자동 Calibration
 ↓
동작 Test
 ↓
AI 이상탐지
 ↓
Pass / Repair
```

검사도 소프트웨어화된다.

---

17. 그리고 로봇이 로봇을 검사한다

카메라가 외관을 검사한다.

로봇팔이 부품을 조립한다.

자동장비가 토크를 확인한다.

AI가 소리와 진동을 분석한다.

즉 휴머노이드 생산량이 늘수록 생산공장 자체도 자동화될 가능성이 크다.

여기서 흥미로운 순환이 생긴다.

---

18. 로봇이 로봇을 만드는가

완전히 스스로 로봇을 만드는 공장을 상상하기 쉽다.

현실은 더 점진적일 가능성이 크다.

전용 산업로봇이 반복조립을 한다.

사람이 복잡한 작업을 한다.

휴머노이드가 기존 자동화로 어려운 중간 작업을 맡는다.

검사 AI가 품질을 확인한다.

즉

전용 자동화 + 인간 + 휴머노이드

가 섞인 공장이 먼저 등장할 가능성이 높다.

---

19. 그런데 이것이 성공하면 강력한 순환이 생긴다

```
로봇 생산 ↑
 ↓
공장 자동화 ↑
 ↓
제조비 ↓
 ↓
로봇 가격 ↓
 ↓
수요 ↑
 ↓
생산량 ↑
←
```

로봇이 자기 생산비를 낮추는 데 기여한다.

이것이 Physical AI Flywheel의 제조 버전이다.

---

20. 생산량이 늘면 공급업체도 바뀐다

초기:

연 1,000개 주문.

공급업체:

“기존 제품 쓰세요.”

대량생산:

연 100만 개 주문.

공급업체:

“전용 제품을 개발하겠습니다.”

그 순간 부품이 로봇에 최적화되기 시작한다.

더 작게.

더 싸게.

더 가볍게.

더 쉽게 조립되게.

---

21. 그래서 대량생산은 기술을 바꾼다

이것이 매우 중요하다.

흔히

기술 발전 → 대량생산

이라고 생각한다.

하지만 반대 방향도 있다.

> **대량생산 → 새로운 기술 발전**

이다.

생산량이 커지면 전용 부품 개발이 경제성을 갖고, 전용 생산장비와 새로운 공정에 투자할 수 있기 때문이다.

---

22. PC와 스마트폰에서도 봤다

초기에는 비싼 부품을 조립했다.

시장이 커졌다.

전용칩이 등장했다.

부품이 통합됐다.

생산공정이 자동화됐다.

가격이 내려갔다.

새로운 소비자가 들어왔다.

시장이 더 커졌다.

휴머노이드가 이와 비슷한 곡선을 탈 수 있는지는 아직 검증 중이다.

하지만 가능하다면 **가격하락은 선형적이지 않을 수 있다.**

---

23. 그런데 스마트폰과 결정적인 차이가 있다

스마트폰을 떨어뜨리면 화면이 깨진다.

휴머노이드를 떨어뜨리면?

바닥도 깨질 수 있고,

로봇도 깨지고,

옆 사람이 다칠 수도 있다.

그리고 스마트폰에는 수십 개의 고출력 관절이 없다.

따라서 휴머노이드 대량생산은 소비자 전자제품보다 훨씬 강한

기계적 신뢰성,

안전,

정비

문제를 함께 해결해야 한다.

---

24. 그래서 판매 후가 중요하다

100만 대를 팔았다.

끝?

아니다.

오히려 시작이다.

손이 고장 난다.

배터리가 열화된다.

감속기를 교체한다.

소프트웨어를 업데이트한다.

사고를 처리한다.

리콜할 수도 있다.

즉 대량생산에는 반드시

서비스망

이 따라온다.

---

25. 자동차 산업이 또다시 선생님이 된다

자동차 회사는 차만 만들지 않는다.

부품창고.

서비스센터.

정비 매뉴얼.

진단장비.

보증.

리콜 시스템.

중고 부품.

보험.

이 거대한 후방 시스템이 있다.

휴머노이드가 수백만 대가 된다면 비슷한 인프라가 필요할 수 있다.

---

26. 그리고 소프트웨어 기업에는 낯선 비용이 등장한다

서버 프로그램은 버그를 패치한다.

휴머노이드의 무릎 감속기는 인터넷으로 패치할 수 없다.

누군가 교체해야 한다.

Physical AI 기업은 결국

소프트웨어 회사 + 제조업체 + 서비스 회사

가 되어야 하거나, 그 역할들을 파트너 생태계와 나눠야 한다.

---

27. 그래서 '연 100만 대 생산능력'이라는 말도 분해해야 한다

앞으로 기업 발표에서 이 말을 들으면 질문한다.

공장 건설능력인가?

라인의 이론적 최대치인가?

실제 생산량인가?

실제 판매량인가?

고객에게 인도된 수량인가?

실제로 일하는 수량인가?

전부 다른 숫자다.

```
설계 Capacity
    ↓
실제 생산
    ↓
판매·계약
    ↓
출하
    ↓
고객 배치
    ↓
실제 가동
```

우리가 가장 관심 있는 것은 맨 아래다.

2026년의 실제 발표들을 이 자에 올려보면 차이가 선명하다.

사례	확인 가능한 숫자	올바른 분류
Tesla Fremont	연 100만 대	1세대 라인 설계 용량·미래 목표
Figure BotQ 첫 라인	최대 연 1만2천 대	생산능력 계획
UBTECH 2025	판매 1,079대	상장사 공시 판매수량
UBTECH 2025년 말	연간 생산능력 6,000대 초과	생산능력
AgiBot 2026년 6월	누적 1만5천 번째 생산 발표	기업 발표상의 누적 생산

이 표의 어느 숫자도 그 자체로 “1만5천 대가 고객 공장에서 자율적으로 돈을 벌고 있다”는 뜻은 아니다.

---

28. 결국 공장도 데이터 공장이 된다

생산 과정에서

불량 데이터.

조립시간.

토크값.

진동.

부품 편차.

검사결과.

고장률.

모두 데이터가 된다.

AI가 이를 분석하면 설계를 다시 개선할 수 있다.

따라서 미래 휴머노이드 기업에는 세 개의 데이터 플라이휠이 동시에 존재한다.

---

29. 세 개의 플라이휠

① 행동 플라이휠
배치 → 행동 데이터 → AI 개선

② 제조 플라이휠
생산 → 공정 데이터 → 원가·수율 개선

③ 신뢰성 플라이휠
가동 → 고장 데이터 → 수명·가동률 개선

그리고 세 개가 서로 연결된다.

좋은 제조가 더 많은 로봇을 만든다.

더 많은 로봇이 더 많은 행동 데이터를 만든다.

더 많은 가동시간이 더 많은 신뢰성 데이터를 만든다.

---

30. 여기서 휴머노이드 산업의 진짜 규모경제가 보인다

단순히 많이 만들면 부품이 싸진다는 정도가 아니다.

많이 만들수록

싸게 만드는 법을 배우고,

오래 쓰는 법을 배우고,

일을 잘하는 법을 배운다.

따라서 일정 규모 이후에는

Scale → Learning → Scale

이라는 자기강화 구조가 만들어질 수 있다.

이것이 성공한다면 선두기업과 후발기업의 격차가 빠르게 벌어질 가능성도 있다.

하지만 아직 중요한 질문이 하나 남아 있다.

---

31. 그래서 인간보다 싼가?

로봇 한 대를 2만 달러에 만들었다.

여기서 2만 달러는 계산 구조를 설명하기 위한 가상 숫자다. 2026년 특정 상용 휴머노이드의 확인된 판매가격을 뜻하지 않는다.

축하한다.

하지만 경제혁명이 일어난 것은 아니다.

공장장은 다시 계산기를 꺼낸다.

구입비.

금융비용.

전기.

소프트웨어.

정비.

배터리.

손 교체.

보험.

인간 감독.

다운타임.

그리고 작업 성공률.

전부 더한다.

그다음 묻는다.

> **“이 로봇 노동자 한 명을 한 시간 고용하는 데 실제로 얼마입니까?”**

드디어 이 책의 모든 기술이 하나의 숫자로 모인다.

제20장 마지막 챗화

로봇공학자

“완성했습니다.”

제조 엔지니어

“몇 대요?”

“한 대.”

“10만 대 만들 수 있어요?”

“…….”

몇 년 뒤.

제조 엔지니어

“10만 대 만들 수 있습니다.”

공장장이 들어온다.

“좋습니다.”

둘이 웃는다.

공장장이 계산기를 켠다.

> **“그런데 사람보다 쌉니까?”**

둘의 표정이 다시 굳는다.

PART VI

행동의 경제학

최종 단위는 로봇 한 대가 아니라 성공한 행동 하나의 가격이다.

제21장 – 로봇 한 시간은 얼마인가

2만 달러짜리 로봇이 5만 달러짜리 로봇보다 비쌀 수 있는 이유

이 장의 2만·5만 달러, 시간당 8달러, 연간 비용과 작업량은 **계산 원리를 설명하기 위한 가상값**이다. 특정 기업의 2026년 판매가·계약가를 뜻하지 않는다.

우리는 긴 길을 돌아왔다.

모터를 열었다. 감속기를 찌그러뜨렸다. 손가락에 촉각을 달았다. 로봇을 백만 번 넘어뜨렸다. AI를 넣었다. 공장에 배치했다. 공급망을 따라 광산까지 내려갔다. 고장률을 계산하고 10만 대를 생산하는 공장도 만들었다.

이제 이 모든 기술을 **하나의 숫자**로 압축할 차례다.

> **로봇 노동 한 시간은 얼마인가?**

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

V17. Cost per Successful Action

싼 로봇이 아니라, 성공한 행동 하나가 싼 시스템을 측정한다

분자 로봇을 보유하고, 움직이고, 실패를 복구하는 총비용

- 구매·금융
- 통합·설치
- 에너지
- 다운타임
- 실패복구

분모 고객 기준을 통과해 실제로 완료된 행동 수

- **품질** — 정해진 결과
- **시간** — 허용 시간 안

- **성공률 ↑** — 재시도·폐기 감소
- **가동률 ↑** — 멈춘 시간 감소

판매가격↓만으로는 부족하다.

정비·부품

소프트웨어

감독·원격개입

안전·품질 보증 비용

네 기준은 **AND** 조건

안전

사람·설비 보호

완결

복구까지 종료

인간 개입 ↓

감독비·대기 감소

복구시간 ↓

MTTR·손실 감소

성공률·가동률·복구가 함께 개선돼야 한다.

이 질문에 답하지 못하면 휴머노이드는 아직 기술제품일 뿐이다.

답할 수 있게 되는 순간 **노동시장에 진입한 경제주체**가 된다.

---

1. 가장 흔한 착각 – 판매가격을 임금과 비교한다

휴머노이드 가격이 2만 달러라고 하자.

사람의 연봉이 4만 달러다.

그러면 이런 계산이 등장한다.

> **"6개월이면 본전을 뽑네."**

틀렸다.

비교 대상이 다르다.

사람의 연봉과 로봇의 구매가격을 비교하고 있기 때문이다.

자동차 가격과 택시기사 연봉을 비교하는 것과 비슷하다.

---

2. 같은 단위로 바꿔야 한다

인간:

노동 1시간의 총비용

로봇:

생산적인 노동 1시간의 총비용

둘을 같은 단위로 맞춰야 한다.

그래야 처음으로 경제적 비교가 가능해진다.

---

3. 로봇의 비용은 구매하는 순간 끝나지 않는다

로봇을 샀다.

그다음부터 돈이 든다.

```
구입·리스
+
금융비용
+
전기
+
충전설비
+
소프트웨어
+
AI 서비스
+
정비
+
부품교체
+
배터리
+
보험
+
통신
+
인간 감독
+
다운타임
```

이 전체가 **총소유비용**(TCO)이다.

다만 TCO는 하나의 고정 숫자가 아니다. 계산 기간과 가동 조건, 포함 비용의 경계를 먼저 잠가야 한다. 현장 통합·안전검증·교육·인프라 변경·폐기비용과 잔존가치를 넣는지에 따라 결과가 달라진다. 서로 다른 로봇의 TCO를 비교하려면 같은 기간과 같은 작업 범위를 써야 한다.

---

4. 그런데 TCO만으로도 부족하다

다음 숫자도 비교 구조를 보여 주기 위한 예시다.

로봇 A와 B가 있다.

A:

연간 비용 1,000만원.

B:

연간 비용 1,500만원.

A가 싸 보인다.

그런데 A는 실제로 1년에 1,000시간 일한다.

B는 3,000시간 일한다.

실제 생산시간으로 나누면 이야기가 바뀐다.

---

5. 첫 번째 핵심 변수 – 가동률

하루 24시간 중 실제 작업 가능한 시간이 얼마인가?

충전.

정비.

고장.

소프트웨어 업데이트.

작업 대기.

사람의 개입.

이 시간을 모두 빼야 한다.

```
24시간
 - 충전
 - 정비
 - 고장
 - 대기
 - 인간 개입
 ──────────
실제 가동시간
```

따라서 단일 배터리가 90분 또는 4시간 가는 로봇이라도 자동충전·배터리 교환·교대운영을 잘하면 높은 하루 가동률을 만들 수 있다.

반대도 가능하다. 배터리 사양이 길어도 고장·작업 대기·사람 개입이 많으면 실제 가동률은 낮다.

---

6. 두 번째 변수 – 작업속도

사람:

1시간에 상자 100개.

로봇:

1시간에 60개.

로봇이 시간당 비용은 싸더라도 처리량이 낮으면 경제성이 달라진다.

그래서 진짜 비교는

시간당 비용

뿐 아니라

작업 한 건당 비용

으로 내려가야 한다.

---

7. 세 번째 변수 – 성공률

로봇이 한 시간에 100번 작업한다.

99번 성공.

한 번 실패.

99%다.

좋아 보인다.

하지만 실패한 한 번 때문에 사람이 10분 동안 개입한다면?

실패비용이 생각보다 커진다.

---

8. 그래서 우리가 15장에서 만든 개념으로 돌아온다

Cost per Successful Action

성공한 행동 하나의 가격

개념적으로는 이렇다.

$$\frac{\text{구매·금융·에너지·정비·소프트웨어·감독·다운타임·실패복구 비용}}{\text{품질기준을 통과해 성공적으로 완료한 행동 수}}$$

분모는 단순 시도 횟수가 아니다. 고객이 정한 품질·시간·안전 조건을 통과한 행동만 센다.

분자에는 실패한 시도와 복구, 사람의 원격개입, 멈춘 시간의 비용까지 들어간다. 그래야 값싼 로봇이 실제로는 비싼 이유를 잡아낼 수 있다.

여기에는 로봇의 거의 모든 기술이 들어간다.

모터 효율이 좋아지면 내려간다.

배터리가 좋아져도 내려간다.

AI 성공률이 올라가도 내려간다.

고장률이 낮아져도 내려간다.

생산가격이 내려가도 내려간다.

책 전체가 하나의 분수로 모인다.

---

9. 네 번째 변수 – 인간 개입률

이것이 굉장히 중요하다.

‘자율 휴머노이드’라고 광고한다.

그런데 작업 중 문제가 생길 때마다 원격 직원이 개입한다.

그러면 그 인간의 노동비도 로봇 비용이다.

---

10. 1:1이면 자동화가 아닐 수도 있다

로봇 한 대.

원격조작자 한 명.

로봇이 일을 한다.

사실상 인간 노동을 다른 장소로 옮긴 것일 수 있다.

하지만 한 사람이 로봇 20대를 감독한다면?

완전히 다른 경제가 된다.

```
인간 1명
  ↓
Robot 1
Robot 2
Robot 3
...
Robot 20
```

따라서 앞으로 중요한 산업지표 가운데 하나는

Robots per Supervisor

가 될 수 있다.

---

11. 인간 개입률이 떨어지는 순간 경제성이 급변한다

초기:

로봇 1대당 사람 1명.

↓

개선:

사람 1명당 5대.

↓

더 개선:

1명당 20대.

↓

성숙:

사람은 희귀한 예외만 처리.

이 과정에서 로봇 하드웨어 가격이 전혀 변하지 않아도 **실제 노동비용은 크게 내려갈 수 있다.**

AI 개선이 경제적 가치를 만드는 중요한 경로다.

---

12. 다섯 번째 변수 – 복구능력

컵을 떨어뜨렸다.

Robot A

멈춘다.

사람 호출.

Robot B

실패를 감지한다.

컵 위치를 다시 찾는다.

다시 집는다.

작업 계속.

두 로봇의 기본 성공률이 같더라도 B의 운영비가 낮을 수 있다.

그래서 중요한 것은

실패하지 않는 능력

뿐 아니라

> **실패한 뒤 스스로 복구하는 능력**

이다.

---

13. 여섯 번째 변수 – 작업 전환비용

산업용 로봇팔은 특정 작업에서 매우 효율적이다.

하지만 새로운 작업으로 바꾸려면

치구,

프로그래밍,

라인 변경

이 필요할 수 있다.

휴머노이드의 경제적 약속은 다르다.

아침:

상자 운반.

오후:

부품 공급.

저녁:

정리.

같은 몸이 여러 작업을 한다면 **유휴시간을 줄일 수 있다.**

---

14. 여기서 범용성의 경제적 가치가 나온다

범용성은 멋진 기술적 표현이 아니다.

경제적으로는

> **한 자산의 활용률을 높이는 능력**

이다.

로봇 A:

작업 하나에서 매우 빠름.

하루 4시간만 필요.

로봇 B:

조금 느리지만 세 작업을 수행.

하루 18시간 사용.

어느 쪽이 경제적인지는 단순 성능으로 결정되지 않는다.

---

15. 일곱 번째 변수 – 공간비용

기존 자동화에는

안전펜스,

전용 컨베이어,

전용 치구,

공장 재설계

가 필요할 수 있다.

휴머노이드가 인간용 환경을 그대로 사용한다면 이런 **인프라 전환비용**을 줄일 가능성이 있다.

이것이 우리가 0장에서

> **"왜 하필 인간 모양인가?"**

라고 물었던 질문과 21장에서 다시 만나는 지점이다.

---

16. 인간형의 진짜 경제적 가설

인간과 똑같이 생겨서 친근하다?

부차적일 수 있다.

더 근본적인 가설은 이것이다.

> **세상을 로봇에게 맞추는 비용보다 로봇을 인간의 세계에 맞추는 비용이 더 싸질 수 있다.**

이 가설이 맞아야 휴머노이드의 경제적 존재 이유가 강해진다.

---

17. 그렇다면 기존 자동화가 가능한 곳에도 휴머노이드를 넣을까

반드시 그렇지는 않다.

초당 수십 개의 동일 부품을 처리한다.

환경은 완전히 고정돼 있다.

전용 로봇팔이 훨씬 빠르고 싸다.

그런 곳에 굳이 두 다리와 두 손을 가진 범용로봇을 넣을 이유가 없다.

휴머노이드의 강점은 **변화와 다양성**이 있는 곳에서 커질 가능성이 있다.

---

18. 자동화의 지도를 그려보자

```
높은 반복성
낮은 환경 변화
      ↓
전용 자동화 유리


중간 반복성
중간 변화
      ↓
휴머노이드 후보


낮은 반복성
높은 변화
      ↓
현재는 인간 우위 가능성
```

AI와 손기술이 발전하면 가운데 영역이 양쪽으로 확장될 수 있다.

19. 그래서 휴머노이드는 모든 로봇을 대체하지 않는다

산업용 로봇팔.

AMR.

AGV.

컨베이어.

드론.

특수 목적 로봇.

이들은 사라질 이유가 없다.

오히려 미래 공장은

전용로봇 + 이동로봇 + 휴머노이드 + 인간

의 혼합체가 될 가능성이 크다.

---

20. RaaS가 이 계산을 단순화한다

기업이 로봇을 직접 구매하지 않는다.

월 구독료, 사용시간, 처리량 또는 작업성과에 따라 비용을 지불하고 로봇과 운영 서비스를 함께 받는 계약군을 보통 Robot-as-a-Service라고 부른다. 어떤 비용과 위험이 공급자에게 넘어가는지는 이름이 아니라 계약과 SLA가 정한다.

로봇 업체가 말한다.

> **"시간당 8달러에 노동을 제공합니다."**

여기서 8달러는 예시다. 실제 계약은 작업·가동률·서비스 범위에 따라 달라지며 공개되지 않는 경우가 많다.

고객 입장에서는 훨씬 쉽다.

사람:

시간당 총비용 X.

로봇:

시간당 서비스비 Y.

비교한다.

이것이 Robot-as-a-Service의 강력한 점이다.

Agility Digit과 GXO의 다년 계약은 RaaS가 가상이 아니라 실제 계약형태로 쓰이고 있음을 보여준다. 다만 공개 자료는 계약단가를 제시하지 않으므로 이 사례를 '시간당 8달러'의 근거로 사용하지 않는다.

---

21. 하지만 로봇 회사에도 위험이 남는다

계약에 가동률 보장과 정비가 포함돼 있다면, 고장나도 고객은 약정된 요금만 낼 수 있다.

수리비는 로봇 회사 부담일 수 있다.

배터리 교체도 서비스 범위에 포함될 수 있다.

AI 운영비도 공급자가 떠안을 수 있다.

따라서 RaaS는 제조업체에게 강한 압력을 준다.

> **신뢰성을 높이지 못하면 자기 회사가 손해를 본다.**

고객과 제조사의 이해관계를 어느 정도 정렬시키는 효과가 있다.

---

22. 금융도 로봇 산업에 들어온다

로봇 한 대가 미래에 만들어낼 현금흐름을 예상할 수 있다면,

리스,

대출,

보험,

자산유동화

같은 금융기법도 가능해진다.

자동차와 항공기에서 이미 익숙한 세계다.

Physical AI가 커질수록 **기계 노동을 금융자산으로 보는 시장**도 생길 수 있다.

---

23. 그리고 감가상각 문제가 있다

오늘 5만 달러에 산 로봇.

3년 뒤 더 좋은 로봇이 1만5천 달러.

이 가격도 감가상각 위험을 설명하기 위한 가상 시나리오다.

그러면 기존 로봇의 경제적 가치가 급격히 떨어질 수 있다.

초기 휴머노이드 시장에서는 **기술적 노후화**가 물리적 마모보다 더 빠를 수도 있다.

기업 구매자가 주저할 이유다.

---

24. 소프트웨어 업그레이드가 이 문제를 일부 완화한다

몸은 그대로.

AI 업데이트.

새로운 작업 추가.

성공률 향상.

작업속도 향상.

그러면 이미 설치된 로봇의 생산성이 올라간다.

이것은 전통 기계와 다른 Physical AI의 중요한 특징이다.

> **출고 후에도 노동자의 능력이 올라갈 수 있다.**

---

25. 하지만 하드웨어 한계는 남는다

AI 업데이트로

모터 토크가 두 배가 되지는 않는다.

배터리 용량도 늘어나지 않는다.

손가락 자유도도 생기지 않는다.

따라서 로봇에는

Software Upgrade Ceiling

이 존재한다.

어느 순간 몸 자체를 교체해야 한다.

---

26. 여기서 로봇의 경제적 수명이 결정된다

물리적 수명:

몇 년 버티는가.

기술적 수명:

경쟁제품에 뒤처지기까지 얼마나 걸리는가.

경제적 수명:

계속 사용하는 것이 새 로봇으로 교체하는 것보다 언제까지 싼가.

세 수명은 다르다.

---

27. 행동원가가 노동시장으로 넘어가는 지점

로봇의 성공한 행동 가격이 인간이 같은 작업을 수행하는 총비용과 신뢰조건 아래에서 교차하면 자동화 경계가 움직인다. 그러나 직업은 서로 다른 작업의 묶음이므로 판매가격이나 시간당 비용 하나만으로 고용효과를 말할 수 없다.

다음 장에서는 직업을 작업으로 분해해 이 경계를 본다.

---

28. 그리고 여기서 이 책의 마지막 질문이 나타난다

처음 질문은

> **“로봇은 어떻게 걷는가?”**

였다.

그다음은

> **“어떻게 생각하는가?”**

였다.

이제 질문은 완전히 달라졌다.

> **“인간이 하던 어떤 행동을 얼마에 살 수 있게 되는가?”**

이것이 로봇 혁명의 경제적 본질이다.

제21장 마지막 챗화

CEO

“우리 로봇은 사람보다 쌉니다.”

경제학자

“시간당 얼마죠?”

CEO:

“로봇 가격은 2만 달러입니다.”

경제학자:

“그건 제가 묻지 않았습니다.”

“…….”

"가동률은?"

"92%입니다."

"작업 성공률?"

"99.5%."

"사람 개입은?"

"작업 300번에 한 번."

"수리비는?"

CEO가 계산기를 꺼낸다.

한참 뒤 숫자 하나가 나온다.

경제학자가 고개를 끄덕인다.

> **"이제야 로봇의 가격을 알겠군요."**

제22장 – 로봇은 직업을 빼앗는가

아니, 먼저 직업을 분해한다

휴머노이드 이야기는 거의 언제나 같은 질문으로 끝난다.

> **"그래서 인간의 일자리는 없어지는가?"**

하지만 21장까지 로봇을 몸속부터 공장까지 따라온 우리에게는 이 질문이 너무 크다는 것이 보인다.

로봇은 어느 날 아침 '용접공', '간호사', '요리사', '택배기사'라는 직업 하나를 통째로 삼키지 않는다.

먼저 **직업 안으로 들어간다.**

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

V18. 직업 → TASK 분해

자동화는 직업 전체가 아니라, 직업 안의 서로 다른 작업에서 시작한다



같은 직업 안에서도 자동화 조건은 다르다

구조화	환경을 고정할 수 있는가
반복성	같은 동작이 되풀이되는가
물리부담	무겁고 위험하고 지치는가

직업은 사라지기 전에 먼저 구성비가 바뀐다.

2 · 수행 주체를 다시 짠다

로봇 우선

구조화·반복·물리부담

① 운반

② 개봉

③ 정렬

인간+로봇

검사·손기술·확인

④ 검사

⑤ 조립

⑥ 기록

인간 우선

예외·관계·책임

⑦ 예외처리

⑧ 동료소통

작업을 이동시키는 네 경계

- **실패비용** — 실수했을 때 피해가 큰가
- **손재주** — 접촉·변형을 다뤄야 하는가
- **사회판단** — 맥락·대화가 핵심인가
- **책임** — 판단과 책임을 넘길 수 있는가

기술적 자동화 가능성 ≠ 실제 고용효과

그리고 그 직업을 여러 개의 작업으로 쪼갠다.

로봇 자동화의 기본단위는 **직업**(job)보다 **작업**(task)에 가깝다.

---

1. '공장 노동자'를 분해해보자

공장 노동자 한 사람이 하루 동안 하는 일을 생각해보자.

부품을 가져온다.

상자를 연다.

부품을 검사한다.

기계에 넣는다.

버튼을 누른다.

완성품을 꺼낸다.

이상한 소리가 나면 확인한다.

옆 작업자에게 묻는다.

기계가 멈추면 원인을 찾는다.

관리자에게 보고한다.

전부 같은 난이도의 일이 아니다.

---

2. 로봇에게 쉬운 것과 어려운 것이 섞여 있다

상대적으로 자동화하기 쉬운 작업

반복적.

물체가 표준화됨.

환경이 일정함.

성공 여부가 명확함.

어려운 작업

처음 보는 상황.

변형되는 물체.

사람과의 복잡한 협력.

희귀한 고장.

모호한 목표.

책임 있는 판단.

따라서 하나의 직업 안에서도 자동화 가능성이 서로 다르다.

---

3. 직업은 사실 작업의 묶음이다

```
JOB
 |
 ├── Task A
 ├── Task B
 ├── Task C
 ├── Task D
 ├── Task E
 └── Task F
```

로봇은 이 가운데 A와 B를 먼저 가져갈 수 있다.

그러면 직업 전체가 없어지는 것이 아니라 **직업의 구성비가 바뀐다.**

이것이 휴머노이드 노동을 이해하는 첫 번째 원칙이다.

---

4. 예를 들어 간호를 생각해보자

간호사의 일을 단순히 '환자를 돌본다'고 표현하면 로봇화 가능성을 분석할 수 없다.

분해해야 한다.

물품 운반.

침구 운반.

환자 이동 보조.

기록.

활력징후 확인.

투약 확인.

처치.

환자 상태변화 판단.

환자와 대화.

보호자 설명.

응급상황 대응.

같은 직업 안에 **물류와 손기술과 판단과 관계**가 함께 있다.

---

5. 로봇이 먼저 들어올 가능성이 있는 곳

굳이 인간이어야 할 이유가 적고,

반복적이며,

신체 부담이 크고,

환경을 구조화할 수 있는 작업이다.

예를 들어 병원이라면 물품 운반이나 일부 반복적 물리 작업부터 접근할 수 있다.

이것은 간호사를 '대체한다'는 것과 다른 이야기다.

> **간호사의 시간을 구성하는 작업 일부를 기계가 가져가는 것이다.**

---

6. 그래서 자동화율 30%는 직원 30% 해고와 같은 말이 아니다

직업의 작업시간 30%가 자동화됐다고 하자.

회사는 사람을 30% 줄일 수도 있다.

하지만 다른 선택도 있다.

같은 인원으로 생산량을 늘린다.

초과근무를 줄인다.

사람을 다른 작업으로 이동시킨다.

서비스 수준을 높인다.

인력부족을 메운다.

즉

기술적 자동화 가능성 ≠ 실제 고용효과라는 뜻이다.

중간에 기업의 선택, 노동시장, 임금, 규제, 수요가 존재한다.

---

7. 특히 인력부족 산업에서는 의미가 달라진다

사람이 충분하지 않은 작업이 있다.

야간.

위험작업.

고강도 반복작업.

특정 지역.

고령화로 인력이 줄어드는 산업.

이런 곳에서는 로봇이 기존 노동자를 밀어내기보다 **비어 있는 노동시간을 채우는 역할**을 할 수도 있다.

따라서 로봇의 사회적 영향은 산업별로 달라질 수밖에 없다.

---

8. 위험한 일을 먼저 넘길 수도 있다

고열.

유해물질.

무거운 물체.

추락 위험.

화재.

방사선.

재난현장.

이런 환경에서는 로봇의 가치가 단순한 임금절감보다 커진다.

> **사람을 위험에 노출시키지 않는 가치**

가 들어간다.

따라서 로봇의 경제성을 인간 임금만으로 계산해서도 안 된다.

---

9. 그런데 로봇이 쉬운 일을 가져가면 인간에게 어려운 일만 남을 수도 있다

여기에는 역설이 있다.

사람이 하던 단순작업을 로봇이 모두 가져간다.

인간에게 남는 것은

예외상황,

고객 불만,

고장,

복잡한 판단,

응급상황

뿐이다.

즉 인간의 일은 줄어들면서 동시에 **더 인지적으로 밀도가 높아질 수 있다.**

자동화가 항상 일을 편하게 만드는 것은 아니다.

---

10. 그리고 숙련 형성의 문제가 생긴다

초보자는 단순한 일부터 배우면서 숙련자가 된다.

그런데 단순작업을 모두 로봇이 한다면?

신입 직원은 어디서 경험을 얻을까?

항공 자동화와 의료 AI 등에서도 생각할 수 있는 문제다.

초급 작업의 자동화가 장기적으로 인간 숙련의 사다리를 약화시킬 가능성도 검토해야 한다.

---

11. 반대로 새로운 숙련도 생긴다

로봇을 설치한다.

가르친다.

작업을 수정한다.

예외상황을 처리한다.

성능을 평가한다.

Fleet를 관리한다.

정비한다.

새로운 직무가 만들어진다.

우리는 이미 그 직업 하나를 만났다.

Robot Teacher

---

12. Operator에서 Teacher로

초기:

사람이 로봇을 직접 조종한다.

Robot Operator

↓

중기:

로봇이 일하고 사람은 예외상황에 개입한다.

Robot Supervisor

↓

더 발전:

사람은 새로운 작업을 시범으로 가르친다.

Robot Teacher

↓

성숙:

사람이 여러 로봇의 작업정책과 성능을 관리한다.

Fleet Manager

사람의 노동이 사라지기보다 **로봇 노동을 조직하는 노동**이 생긴다.

---

13. 그런데 이것이 기존 일자리 수만큼 생긴다는 보장은 없다

여기서는 낙관론도 조심해야 한다.

로봇 100대를 관리하는 데 사람 5명만 필요하다면 새로운 직무는 생겼지만 필요한 인원은 줄어든다.

반대로 로봇 가격 하락으로 생산과 서비스 수요가 크게 늘면 전체 고용이 다른 방향으로 움직일 수도 있다.

따라서 지금 단계에서

"새 일자리가 반드시 더 많이 생긴다"

또는

"대량실업이 필연이다"

라고 단정하기 어렵다.

---

14. 가격탄력성이 중요하다

로봇 때문에 어떤 서비스 가격이 절반으로 내려갔다고 하자.

사람들이 그 서비스를 두 배 이상 소비하면 산업 전체 규모가 커질 수 있다.

역사적으로 기술혁신은 이런 효과를 반복해서 만들었다.

그러나 모든 산업에서 수요가 무한히 늘지는 않는다.

따라서 자동화의 고용효과는

생산성 증가 + 가격 하락 + 수요 증가

를 함께 봐야 한다.

---

15. 로봇이 가장 먼저 경쟁하는 것은 인간 전체가 아니다

흥미롭게도 로봇은 인간의 특정 특성과 경쟁한다.

반복성.

힘.

지속시간.

위험내성.

그리고 인간은 다른 특성에서 강하다.

상식.

즉흥적 적응.

사회적 이해.

책임.

복잡한 의사소통.

새로운 환경에 대한 일반화.

따라서 미래 노동은 '인간 대 로봇'보다

인간 + 로봇의 능력 조합

으로 보는 편이 정확하다.

---

16. 로봇에게 책임을 맡길 수 있는가

여기서 중요한 경계가 나온다.

로봇이 약을 가져올 수 있다.

그런데

이 약을 지금 이 환자에게 투여해야 하는가

라는 판단은 다른 문제다.

행동과 책임은 같은 것이 아니다.

로봇이 물리적 작업을 수행하더라도 최종 의사결정 책임은 인간에게 남는 분야가 존재할 수 있다.

---

17. 그래서 자동화에는 세 단계가 있다

```
DO
실행을 자동화

↓

DECIDE
판단을 자동화

↓

ACCOUNT
책임까지 이전
```

첫 번째와 두 번째 사이에도 큰 장벽이 있다.

두 번째와 세 번째 사이에는 기술뿐 아니라 **법과 제도와 사회적 신뢰**라는 더 큰 장벽이 있다.

---

18. 가정은 그래서 가장 늦을 수 있다

공장에서는 작업을 제한할 수 있다.

가정에서는 불가능하다.

“방 좀 치워.”

장난감.

약.

칼.

반려동물.

뜨거운 냄비.

아이.

깨지기 쉬운 물건.

모두 같은 공간에 있다.

게다가 사람은 로봇에게 공장보다 훨씬 높은 사회적 신뢰를 요구할 수 있다.

가정용 휴머노이드는 기술적으로 가장 매력적으로 보이지만 **가장 어려운 시장 중 하나**일 수 있다.

---

19. 노인 돌봄도 마찬가지다

로봇이 물건을 가져온다.

넘어진 사람을 발견한다.

약 복용시간을 알려준다.

신체적 도움을 준다.

가치가 크다.

하지만 돌봄에는

대화,

정서,

의사결정,

존엄,

개인정보,

응급판단

도 들어 있다.

따라서 돌봄로봇을 단순히 '간병인 대체'로 표현하면 기술과 돌봄 양쪽을 모두 지나치게 단순화한다.

---

20. 결국 자동화하기 쉬운 작업에는 공통점이 있다

반복성이 높다.

목표가 명확하다.

환경 변동성이 낮다.

실패비용이 낮거나 통제 가능하다.

물체가 표준화돼 있다.

반대로 자동화하기 어려운 작업은 이 조건의 반대쪽에 있다.

---

21. 자동화 경계는 세 구역으로 읽는 편이 낫다

환경 변동성·작업 다양성 낮음 ↓ **전용 자동화 우위** ↓ **휴머노이드 후보** ↓ **인간·협업 우위** ↓ 환경 변동성·작업 다양성 높음

이 지도는 고정된 서열이 아니다. 같은 작업도 요구 성공률, 실패비용과 인간 개입시간이 달라지면 다른 구역에 놓인다.

---

22. 그리고 가격이 경계를 움직인다

기술적으로 가능한 작업이라도 로봇이 너무 비싸면 사람이 한다.

로봇 가격이 내려간다.

성공률이 올라간다.

인간 개입률이 내려간다.

그러면 어느 순간 경제적 경계가 이동한다.

어제는 사람이 하던 작업을 오늘은 로봇이 한다.

즉 자동화 경계는 고정돼 있지 않다.

---

23. 여기서 책 전체의 핵심 개념이 다시 등장한다

행동의 가격

모터가 싸졌다.

↓

감속기가 싸졌다.

↓

로봇이 싸졌다.

↓

AI가 좋아졌다.

↓

실패율이 낮아졌다.

↓

인간 개입이 줄었다.

↓

성공한 행동 하나의 가격이 내려갔다.

↓

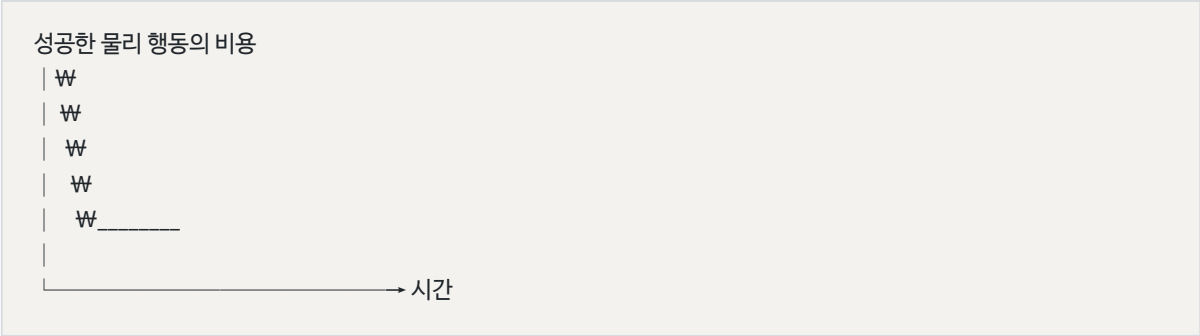
자동화 가능한 작업영역이 넓어진다.

기술 변화가 노동시장으로 전달되는 경로다.

---

24. 따라서 로봇 혁명의 진짜 그래프는 로봇 대수만이 아니다

더 중요한 그래프는 이것일 수 있다.



이 곡선이 내려갈수록 더 많은 인간 작업이 경제적 자동화 영역 안으로 들어온다.

---

25. 그리고 모든 작업의 가격이 동시에 떨어지지는 않는다

상자 옮기기.

빠르게 떨어질 수 있다.

복잡한 전선 조립.

더 늦을 수 있다.

노인 목욕 보조.

안전과 신뢰 때문에 훨씬 늦을 수 있다.

응급 의료판단.

또 다른 장벽이 있다.

따라서 **'휴머노이드 혁명 날짜' 하나는 존재하지 않을 가능성**이 높다.

작업마다 임계점이 따로 온다.

---

26. 이것이 'ChatGPT Moment'라는 표현의 한계다

ChatGPT는 하나의 서비스가 공개되면서 많은 사람이 거의 동시에 AI의 능력을 경험했다.

Physical AI는 다를 수 있다.

공장 물류의 ChatGPT Moment.

창고의 ChatGPT Moment.

건설의 ChatGPT Moment.

가정의 ChatGPT Moment.

각각 다른 시기에 올 수 있다.

즉 **하나의 순간이 아니라 여러 개의 임계점**이다.

---

27. 그리고 인간의 역할도 한 방향으로만 움직이지 않는다

어떤 작업에서는

인간 → 로봇.

다른 작업에서는

인간 + 로봇.

또 다른 곳에서는

로봇 → 인간 감독.

그리고 어떤 작업은 오랫동안 인간이 직접 할 것이다.

따라서 미래 노동의 모습은 완전 자동화 하나가 아니라 **여러 형태의 혼합체**일 가능성이 크다.

---

28. 그러면 인간에게 무엇이 남는가

이 질문에는 너무 쉽게 철학적인 답을 붙일 필요가 없다.

먼저 경제적으로 보면 된다.

로봇에게 아직 비싼 행동이 인간에게 남는다.

새로운 상황.

높은 책임.

복잡한 사회적 관계.

불명확한 목표.

예외처리.

새로운 작업의 정의.

그리고 아이러니하게도

로봇에게 무엇을 시킬 것인가를 결정하는 일이다.

---

29. 하지만 이 경계도 영원하지 않다

AI는 발전한다.

손도 좋아진다.

촉각도 좋아진다.

데이터도 늘어난다.

로봇 가격도 내려갈 수 있다.

따라서 오늘 인간에게 남은 작업 일부가 내일 자동화될 수 있다.

그렇기 때문에 특정 직업 목록보다 더 중요한 것은

> **행동가격 곡선이 어느 방향으로 움직이는가**

를 보는 것이다.

---

30. 이제 알파에서 오메가까지 왔다

우리는 인간의 몸에서 시작했다.

왜 두 발인가.

왜 손인가.

왜 감속기가 필요한가.

왜 배터리가 부족한가.

왜 열이 나는가.

어떻게 보는가.

어떻게 걷는가.

어떻게 생각하는가.

어떻게 배우는가.

누가 만드는가.

얼마인가.

얼마나 오래 버티는가.

그리고 결국

인간의 일과 어떻게 만나는가.

모든 질문이 하나로 모인다.

제22장 마지막 챗화

인간

“결국 내 일을 가져갈 거야?”

로봇

“당신의 일이 뭔데요?”

인간:

“나는 간호사야.”

로봇:

“그건 직업 이름이고요.”

인간:

“…….”

로봇:

“하루 동안 무슨 일을 하세요?”

인간이 하나씩 말하기 시작한다.

옮기고.

기록하고.

관찰하고.

판단하고.

설명하고.

안심시키고.

책임진다.

로봇이 잠시 계산한다.

> **“그중 일부는 제가 할 수 있습니다.”**

인간:

“나머지는?”

로봇:

> **“지금은 당신이 더 잘합니다.”**

인간:

“언제까지?”

로봇은 대답하지 않는다.

그 질문은 기술의 문제가 아니라 **가격과 신뢰와 사회가 함께 결정할 문제**이기 때문이다.

에필로그 – OMEGA

행동의 가격이 무너지는 날

이 책은 로봇에서 시작하지 않았다.

인간에게서 시작했다.

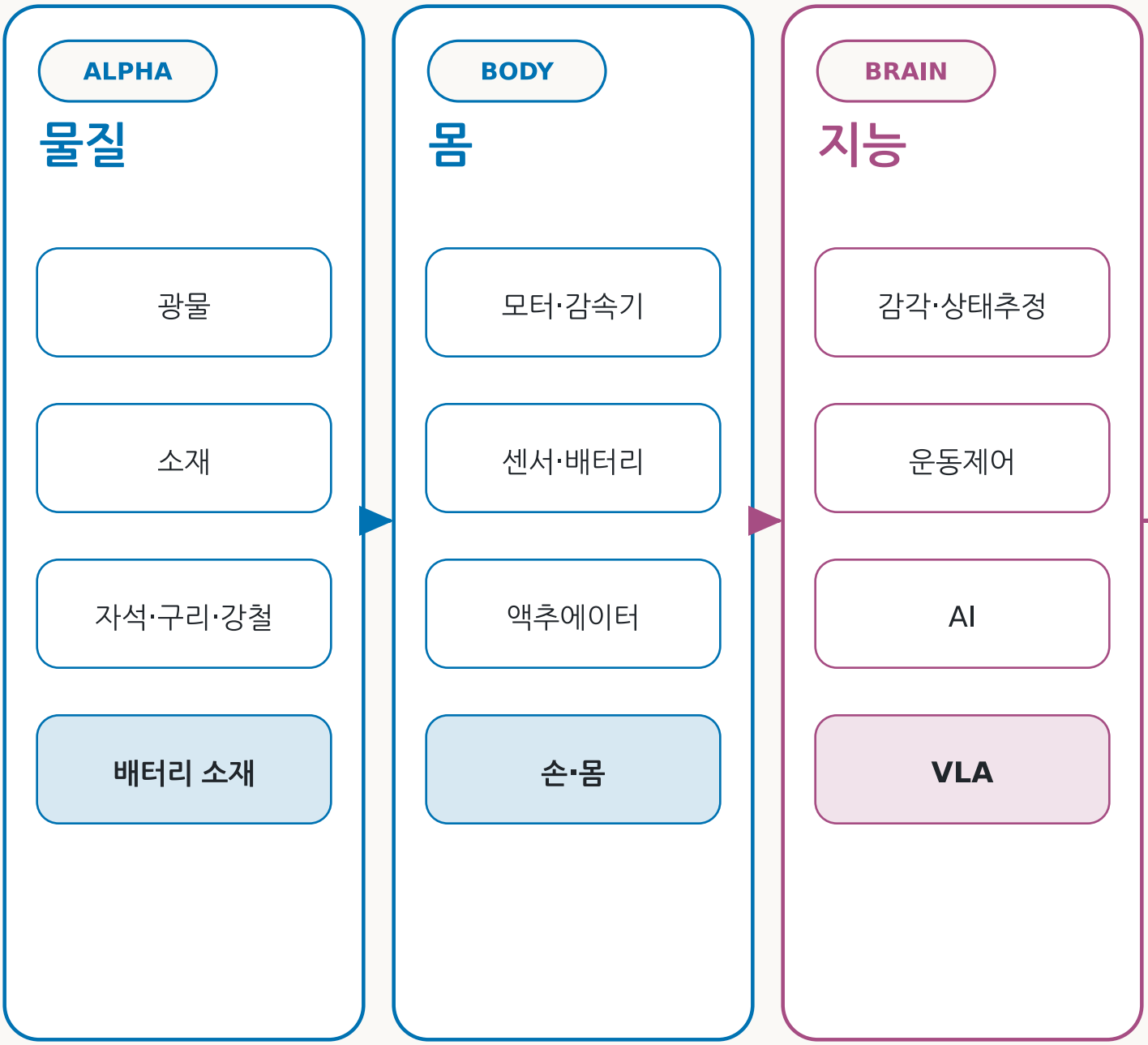
왜 기계가 굳이 두 발로 걸어야 하는가.

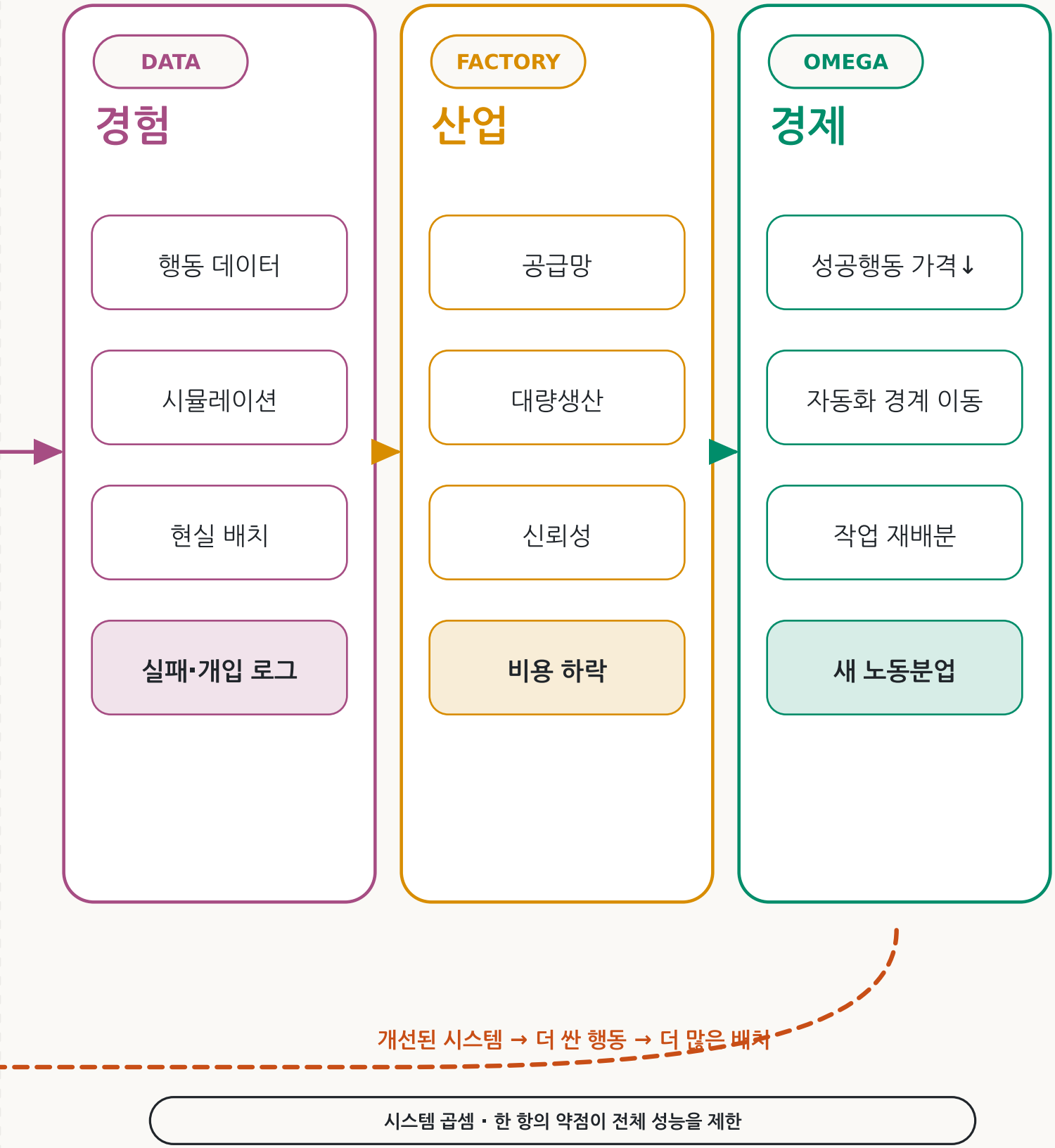
왜 다섯 손가락을 만들려고 하는가.

왜 계단을 오르고 문손잡이를 돌리며 인간이 사용하는 공구를 잡으려고 하는가.

V19. ALPHA → OMEGA 전체 지도

광물에서 인간과 기계의 노동분업까지, 여섯 층의 인과와 되먹임




DATA
경험
행동 데이터
시뮬레이션
현실 배치
실패·개입 로그
FACTORY
산업
공급망
대량생산
신뢰성
비용 하락
OMEGA
경제
성공행동 가격↓
자동화 경계 이동
작업 재배분
새 노동분업
개선된 시스템 → 더 싼 행동 → 더 많은 배치
시스템 곱셈 · 한 항의 약점이 전체 성능을 제한

답은 의외로 로봇에게 있지 않았다.

> **우리가 세상을 이미 인간의 몸에 맞추어 만들어 놓았기 때문이다.**

계단도 인간의 다리를 기준으로 만들어졌다.

문도 인간의 팔과 손을 기준으로 만들어졌다.

공장도, 자동차도, 주방도, 병원도 그렇다.

인간은 수천 년 동안 세상을 자신의 몸에 맞게 개조했다.

그리고 이제 기계가 그 세계로 들어오려 한다.

---

1. 처음에는 쇳덩어리였다

우리는 로봇의 외피를 벗겼다.

안에는 마법이 없었다.

구리.

철.

알루미늄.

자석.

베어링.

배터리.

반도체.

그리고 수많은 전선.

더 아래로 내려가자 광산이 나타났다.

```
광물
 ↓
소재
 ↓
자석·구리·특수강
 ↓
모터·감속기·베어링
 ↓
액추에이터
 ↓
관절
 ↓
몸
```

Physical AI라는 미래적인 이름 아래에는 놀라울 만큼 오래된 물리학이 있었다.

전자기학.

역학.

마찰.

열역학.

재료피로.

중력.

AI가 아무리 발전해도 중력은 업데이트되지 않는다.

---

2. 그래서 로봇에게 몸을 주는 일은 어려웠다

모터를 강하게 만들었다.

뜨거워졌다.

감속비를 높였다.

마찰과 역구동 문제가 생겼다.

배터리를 늘렸다.

무거워졌다.

몸을 튼튼하게 만들었다.

다시 무거워졌다.

손가락을 정교하게 만들었다.

부품과 센서가 늘어났다.

자유도를 늘릴 때마다 가능성은 커졌지만 동시에 고장날 곳도 늘어났다.

휴머노이드의 역사는 끊임없는 교환이었다.

> ****힘 ↔ 무게**
> **정밀도 ↔ 가격**
> **자유도 ↔ 복잡성**
> **배터리 ↔ 질량**
> **성능 ↔ 열**
> **강성 ↔ 안전성****

완벽한 부품은 없었다.

좋은 로봇은 결국 이 모순들을 **전체 시스템 수준에서 잘 타협한 로봇**이었다.

---

3. 그러다 기계가 감각을 얻었다

카메라가 눈이 되었다.

IMU가 평형기관이 되었다.

엔코더가 고유감각이 되었다.

힘센서가 접촉을 느꼈다.

촉각센서가 손끝으로 들어갔다.

그러나 센서는 모두 조금씩 틀렸다.

그래서 로봇은 여러 불완전한 증언을 합쳐 현실을 추정하기 시작했다.

센서융합.

로봇에게 현실은 주어진 것이 아니었다.

> **계속 추정해야 하는 것이었다.**

---

4. 그리고 걷기 시작했다

걷기는 앞으로 이동하는 일이 아니었다.

계속 넘어지면서 다음 발로 자신을 받아내는 일이었다.

센서가 몸의 기울기를 측정했다.

제어기가 계산했다.

발목이 토크를 바꿨다.

한 발을 내밀었다.

다시 계산했다.

수없이 반복했다.

그러다 인간은 실제 로봇을 계속 넘어뜨리는 대신 **가상세계에서 넘어뜨리기 시작했다.**

한 대.

백 대.

만 대.

수십만 대.

실패의 가격이 무너졌다.

---

5. 이것이 첫 번째 중요한 가격붕괴였다

과거에는 실패 하나에도 시간이 들었다.

실제 로봇을 세우고,

시험하고,

넘어뜨리고,

수리하고,

다시 시험했다.

시뮬레이션에서는 다르다.

실패.

삭제.

복제.

다시 시작.

그리고 동시에 수많은 로봇을 훈련시킬 수 있다.

> **시뮬레이션은 로봇의 시간을 빠르게 만든 것이 아니라 경험을 병렬화했다.**

Physical AI가 빨라진 이유 가운데 하나였다.

---

6. 하지만 몸만으로는 부족했다

사람이 말했다.

> **"저 컵 좀 가져와."**

과거의 로봇에게는 너무 모호한 명령이었다.

좌표가 없었다.

관절각도도 없었다.

힘도 지정하지 않았다.

그런데 AI가 언어와 영상을 이해하기 시작했다.

LLM은 언어를 언어로 바꾸었다.

멀티모달 AI는 영상과 언어를 연결했다.

그리고 VLA는 그 연결을 한 단계 더 밀었다.

```
VISION
 +
LANGUAGE
 +
ROBOT STATE
 ↓
ACTION
```

문장이 행동으로 변하기 시작했다.

---

7. 하지만 AI에게도 없는 것이 있었다

인터넷.

정확히 말하면 **행동의 인터넷**이었다.

인터넷에는 인간이 쓴 문장이 넘쳐났다.

그러나 컵을 잡을 때 엄지가 몇 뉴턴으로 눌렀는지는 기록되어 있지 않았다.

그래서 인간이 직접 로봇을 조종했다.

로봇이 기록했다.

로봇이 스스로 시도했다.

또 기록했다.

실패도 기록했다.

시뮬레이터가 존재하지 않았던 경험까지 만들어냈다.

---

8. 로봇은 자기 인터넷을 만들기 시작했다

그 인터넷의 페이지에는 문장이 아니라

영상,

관절각,

힘,

토크,

촉각,

성공,

실패

가 적혀 있었다.

그리고 페이지를 쓰는 존재도 달라졌다.

```
인간
 +
실제 로봇
 +
가상 로봇
 ↓
행동 데이터
 ↓
Physical AI
```

이것이 두 번째 거대한 변화였다.

---

9. 한 로봇의 경험이 다른 로봇에게 이동한다

인간 숙련공 한 명이 새로운 기술을 익혔다.

그 기술을 다른 사람에게 가르치는 데 시간이 걸린다.

로봇은 다를 수 있다.

한 로봇이 실패한다.

↓

데이터를 보낸다.

↓

모델을 다시 학습한다.

↓

검증한다.

↓

수천 대에 배포한다.

개체의 경험이 **집단의 학습**으로 바뀐다.

이것이 Fleet Learning의 잠재력이다.

---

10. 그리고 로봇은 공장으로 들어갔다

그 순간 모든 것이 다시 달라졌다.

연구실에서는 질문이

> **"할 수 있는가?"**

였다.

공장에서는

> **"몇 번 연속으로 할 수 있는가?"**

가 됐다.

데모에서는 성공 한 번이면 박수를 받는다.

공장에서는 99% 성공률도 나쁠 수 있다.

하루 1만 번 작업하면 단순 계산으로 실패가 약 100번이기 때문이다.

따라서 성공률의 마지막 소수점이 갑자기 돈이 되었다.

---

11. 그리고 고장이 나타났다

감속기가 마모됐다.

베어링에 유격이 생겼다.

케이블이 끊어졌다.

배터리가 열화됐다.

센서가 조금씩 틀어졌다.

AI 업데이트가 예상하지 못한 행동을 만들었다.

로봇은 인간처럼 피곤하지 않았다.

하지만 **기계적인 시간**을 피할 수는 없었다.

그래서 세 번째 질문이 나타났다.

> **"얼마나 오래 같은 성능을 유지하는가?"**

---

12. 로봇은 결국 제조업으로 돌아왔다

AI가 아무리 좋아도 로봇은 복사 버튼으로 늘어나지 않았다.

철을 깎아야 했다.

자석을 만들어야 했다.

모터를 감아야 했다.

감속기를 가공해야 했다.

배터리를 조립해야 했다.

그리고 모든 로봇을 검사해야 했다.

Software Scaling은

Manufacturing Scaling

이라는 벽을 만났다.

---

13. 그래서 미국과 중국의 서로 다른 길이 만난다

한쪽에서는 AI와 컴퓨팅이 빠르게 발전했다.

다른 한쪽에서는 EV·배터리·드론·전자제조·정밀부품의 거대한 공급망이 성장했다.

하지만 Physical AI에서는 둘을 분리할 수 없었다.

AI는 몸이 필요했고,

몸은 AI가 필요했다.

그리고 둘 모두 데이터가 필요했다.

결국 우리가 만든 공식으로 돌아온다.

BODY × BRAIN × DATA × FACTORY × SUPPLY CHAIN

하나라도 거의 0이면 전체 시스템이 크게 제한된다.

---

14. 그리고 세 개의 플라이휠이 돌기 시작한다

행동 플라이휠

로봇 배치 → 행동 데이터 → AI 개선 → 성공률 증가 → 더 많은 배치.

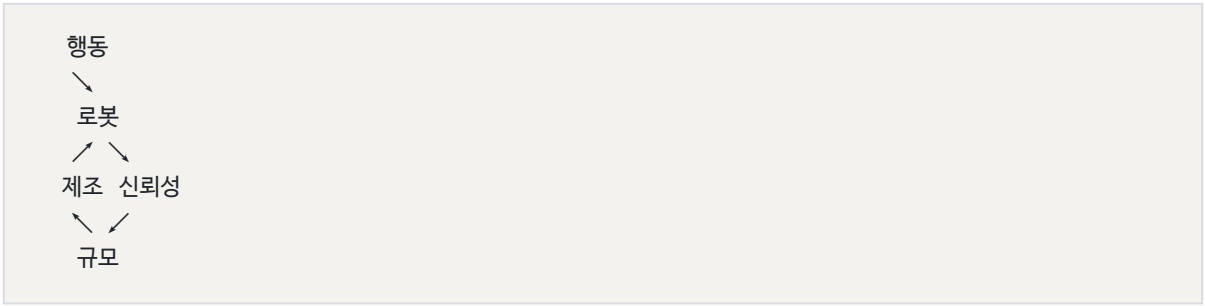
제조 플라이휠

생산량 → 공정 데이터 → 수율 증가 → 원가 하락 → 더 많은 생산.

신뢰성 플라이휠

가동시간 → 고장 데이터 → 설계 개선 → 수명 증가 → 더 많은 가동시간.

이 세 순환이 서로 연결된다.



이것이 휴머노이드 산업이 기대하는 자기강화 구조다.

---

15. 그런데 우리는 아직 가장 중요한 질문을 하지 않았다

이 로봇은 멋진가?

아니다.

인간처럼 보이는가?

아니다.

AGI인가?

그것도 아니다.

공장장이 마지막으로 묻는다.

> **"그래서 이 일을 한 번 시키는 데 얼마입니까?"**

그 질문 하나가 모든 기술을 경제학으로 바꾼다.

---

16. 처음에는 로봇 한 대의 가격을 물었다

10만 달러.

5만 달러.

2만 달러.

1만 달러.

그러나 이 숫자들은 충분하지 않았다.

싼 로봇이 자주 고장 나면 비싸다.

비싼 로봇이 하루 종일 일하면 오히려 쌀 수 있다.

사람이 계속 붙어 있어야 하면 자동화가 아니다.

그래서 우리는 다른 단위를 만들었다.

성공한 행동 하나의 가격

---

17. 이것이 이 책의 OMEGA다

로봇 가격 ↓
+
가동률 ↑
+
작업속도 ↑
+
성공률 ↑
+
복구능력 ↑
+
인간 개입 ↓
+
정비비 ↓
———————————
성공한 행동의 가격 ↓

모든 기술발전이 결국 이 숫자로 수렴한다.

---

18. 그리고 이 가격이 어떤 임계점을 통과한다

어제까지 인간이 상자를 옮겼다.

오늘은 로봇이 더 싸다.

경계가 이동한다.

다음에는 부품 공급.

다음에는 기계 관리.

다음에는 복잡한 조립.

하지만 모든 작업이 동시에 넘어가지는 않는다.

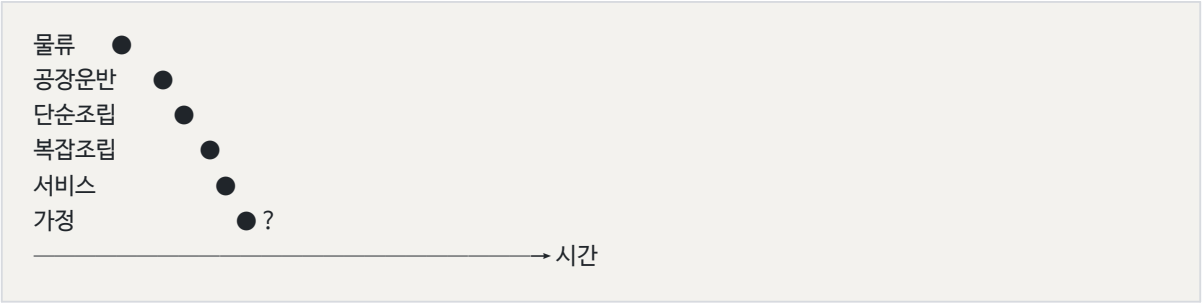
각 작업에는 서로 다른 임계가격이 있다.

---

19. 그래서 '로봇 혁명의 날'은 없을지도 모른다

어느 날 갑자기 모든 휴머노이드가 인간보다 좋아지는 것이 아니다.

대신 작은 임계점들이 연속해서 넘어간다.



위의 위치와 순서는 고정된 예측이 아니라 개념도다.

중요한 것은 **각 작업마다 경제적 임계점이 다르다는 사실**이다.

---

20. 이것이 휴머노이드의 여러 'ChatGPT 순간'이다

하나의 거대한 순간이 아니라,

물류의 순간.

자동차 공장의 순간.

창고의 순간.

건설의 순간.

병원의 순간.

가정의 순간.

각각 따로 올 수 있다.

어떤 것은 가까울 수 있고,

어떤 것은 훨씬 멀 수 있다.

---

21. 인간의 일도 한꺼번에 사라지지 않는다

직업이 분해된다.

반복작업 일부가 로봇으로 이동한다.

사람은 예외를 처리한다.

그다음 로봇이 예외 일부도 배운다.

사람의 역할이 다시 이동한다.

```
DO
직접 수행
 ↓
SUPERVISE
감독
 ↓
TEACH
가르침
 ↓
DESIGN
무엇을 시킬지 정의
 ↓
ACCOUNT
판단과 책임
```

그러나 이 경로 역시 모든 직업에서 동일하지 않다.

---

22. 그리고 한 가지는 기술만으로 결정되지 않는다

로봇이 할 수 있다.

로봇이 더 싸다.

그래도 사회가 허용하지 않을 수 있다.

안전.

법.

보험.

책임.

프라이버시.

인간의 선호.

노동제도.

기술적 가능성과 사회적 채택 사이에는 언제나 거리가 있다.

---

23. 그러므로 '인간은 끝났다'는 말은 너무 쉬운 결론이다

처음 우리가 본 영상 제목에는 그런 자극적인 질문이 있었다.

하지만 로봇을 실제로 분해해본 뒤에는 오히려 더 복잡한 그림이 보인다.

로봇은 인간 전체와 경쟁하지 않는다.

특정 조건의 특정 행동과 경쟁한다.

그리고 인간 역시 고정된 존재가 아니다.

도구를 사용하고,

업무를 재설계하고,

로봇을 감독하고,

새로운 서비스를 만들고,

제도를 바꾼다.

기계가 변하면 인간의 일도 변한다.

---

24. 그렇다고 변화가 작다는 뜻도 아니다

반대다.

만약 성공한 물리적 행동의 가격이 장기간 빠르게 하락한다면 그 영향은 매우 클 수 있다.

산업혁명은 인간의 근육을 기계동력으로 바꾸었다.

컴퓨터 혁명은 계산비용을 떨어뜨렸다.

인터넷은 정보전달비용을 떨어뜨렸다.

생성형 AI는 일부 인지작업의 비용을 크게 낮추고 있다.

그리고 Physical AI가 겨냥하는 다음 영역은

물리적 행동의 비용

이다.

---

25. 이것이 왜 거대한 변화일 수 있는가

세계 경제의 상당 부분은 여전히 물리세계에서 이루어진다.

물건을 만든다.

옮긴다.

쌓는다.

고친다.

청소한다.

건설한다.

돌본다.

수확한다.

이 모든 것은 결국 **행동**이다.

따라서 행동비용의 지속적인 하락은 단순한 로봇산업의 성장을 넘어 여러 산업의 비용구조를 바꿀 잠재력이 있다.

---

26. 그러나 마지막 병목은 항상 현실이다

시뮬레이션에서는 완벽했다.

현실에서는 바닥이 젖어 있었다.

AI는 컵을 알아봤다.

손은 미끄러졌다.

공장에서는 잘했다.

가정에서는 고양이가 지나갔다.

로봇은 결국 물리세계에 산다.

그래서 Physical AI의 마지막 시험장은 언제나

현실이다.

---

27. 알파로 돌아가자

책의 처음.

인간이 로봇에게 물었다.

> **"너는 왜 나처럼 생겼어?"**

로봇은 대답했다.

> **"당신이 세상을 당신처럼 만들어 놓았으니까요."**

이제 질문을 하나 더 붙일 수 있다.

인간:

> **"그럼 왜 내 일을 하려고 하지?"**

로봇:

> **"당신의 일이어서가 아닙니다."**

인간:

"그럼?"

로봇:

> **"그 행동의 가격이 제가 더 싸지는 순간이 오기 때문입니다."**

28. 그러나 인간이 다시 묻는다

“가격만 싸면 모든 일을 네게 맡길까?”

이번에는 로봇이 바로 대답하지 못한다.

안전.

책임.

신뢰.

관계.

존엄.

선호.

법.

사회.

가격표에 바로 들어가지 않는 것들이 남아 있기 때문이다.

그리고 바로 그 지점에서 로봇공학은 공학을 넘어선다.

29. 마지막 지도

광물
↓
자석·구리·강철
↓
모터
↓
감속기
↓
액추에이터
↓
손
↓
감각
↓
보행
↓
배터리·열
↓
컴퓨팅
↓
AI
↓
VLA
↓
데이터
↓
시뮬레이션
↓
현실 배치
↓

```
공급망
 ↓
대량생산
 ↓
신뢰성
 ↓
가격 하락
 ↓
가동률 증가
 ↓
인간 개입 감소
 ↓
————————————————————
 성공한 행동의 가격 하락
————————————————————
 ↓
자동화 경계 이동
 ↓
인간과 기계의 새로운 노동분업
```

이것이 **알파에서 오메가까지**다.

마지막 챗화

공장이 모두 퇴근한 밤.

로봇 한 대가 충전기에 연결되어 있다.

사람이 묻는다.

"너는 피곤하지 않아?"

로봇:

"피곤하다는 것이 무엇인지 모릅니다."

인간:

"그럼 계속 일할 수 있겠네."

로봇은 잠시 자신의 상태를 확인한다.

배터리 18%.

오른쪽 손목 액추에이터 온도 상승.

검지 구동계 마찰값 정상범위 상단.

다음 예방정비까지 126시간.

로봇:

> **“아니요. 충전해야 하고, 식혀야 하고, 언젠가는 부품도 교체해야 합니다.”**

인간이 웃는다.

“그럼 너도 별수 없네.”

로봇:

> **“네. 저도 물리세계에 있으니까요.”**

충전기의 표시등이 켜진다.

공장에는 아무 소리도 없다.

그러나 서버에서는 오늘 하루 동안 로봇들이 실패한 수천 개의 행동이 다시 재생되고 있다.

한 로봇이 상자를 놓쳤다.

다른 로봇이 미끄러졌다.

또 다른 로봇은 사람이 지나가자 멈췄다.

그 경험들이 학습 데이터가 된다.

가상세계에서는 같은 장면이 수천 가지 조건으로 다시 만들어진다.

그리고 다음 날 아침,

어제와 똑같아 보이는 로봇이 공장으로 돌아온다.

하지만 **어제의 로봇과 완전히 같은 로봇은 아니다.**

몸은 거의 그대로지만,

조금 더 배웠다.

끝이 아니라 순환

여기서 이 책을 닫으면 로봇의 역사를 잘못 이해하게 된다.

OMEGA는 종착점이 아니다.

다시 ALPHA로 돌아간다.

```
더 싼 행동
  ↓
더 많은 배치
  ↓
더 많은 현실 경험
  ↓
더 많은 실패 데이터
  ↓
더 좋은 AI
  ↓
더 좋은 설계
  ↓
더 높은 신뢰성
  ↓
더 낮은 제조비
  ↓
더 싼 행동
  ←
```

따라서 휴머노이드의 진짜 가속은 어느 한 번의 발명에서 일어나는 것이 아닐 수 있다.

이 순환고리가 스스로 돌아가기 시작할 때 일어난다.

그리고 인간에게 남는 질문

기술자는 묻는다.

얼마나 잘 움직이는가?

제조업자는 묻는다.

얼마나 싸게 만들 수 있는가?

공장장은 묻는다.

얼마나 오래 일하는가?

투자자는 묻는다.

얼마나 크게 팔리는가?

노동자는 묻는다.

내 일은 어떻게 바뀌는가?

사회는 결국 더 어려운 질문을 해야 한다.

> **싸게 할 수 있게 된 행동 가운데, 무엇을 기계에게 맡길 것인가?**

이 질문에는 모터의 토크도, GPU의 연산량도 정답을 주지 못한다.

그것은 인간이 결정해야 할 문제다.

용어설명

BLDC / PMSM

영구자석 동기기 계열에서 구조가 겹칠 수 있는 모터 명칭이다. 통상 BLDC는 6-step 계열, PMSM은 정현파 전류·FOC 계열의 강조가 크다.

토크

회전시키는 힘의 효과다. 힘과 회전축에서의 거리의 곱으로 읽는다.

토크밀도

질량 또는 부피당 토크다. 모터인지 액추에이터인지, 피크인지 연속인지 조건을 함께 적어야 한다.

스트레인 웨이브 감속기

얇은 탄성 기어의 변형과 작은 잇수 차이를 이용해 큰 감속비를 얻는 정밀감속기 원리다.

유성감속기

선기어·유성기어·링기어·캐리어로 하중을 나누는 감속기 구조다.

사이클로이드 / RV

사이클로이드 운동을 이용하는 감속기 계열이다. RV는 그 원리를 쓰는 특정 정밀감속기 구조·제품군이다.

볼스크루 / 롤러스크루

회전운동을 직선운동으로 바꾸는 나사 전달계다. 하중·강성·효율·수명·역구동성 사이에 절충이 있다.

SEA

모터·전달계와 부하 사이에 직렬 탄성요소를 넣어 충격과 힘제어 특성을 조정하는 액추에이터다.

QDD

고토크밀도 모터와 비교적 낮은 감속비 전달계를 결합하는 설계 계열이다.

역구동성

출력 쪽 외력이 전달계를 거꾸로 움직일 수 있는 정도다. 안전·힘감지·컴플라이언스와 같은 뜻은 아니다.

엔코더

회전 또는 선형 위치를 신호로 제공한다. 속도는 위치의 시간변화로 계산할 수 있다.

6축 힘·토크센서

센서 좌표계에서 세 방향 힘과 세 축 모멘트, 즉 wrench 6성분을 측정한다.

촉각

압력·법선/전단력·변형·진동·온도·접촉형상 등을 감지하는 여러 모달리티의 묶음이다.

IMU

가속도계의 특정힘과 자이로의 각속도를 측정한다. 자세·속도·위치는 적분과 다른 정보의 결합으로 추정한다.

센서융합

서로 다른 측정과 불확실성을 결합하는 방법이다.

상태추정

측정과 모델로 직접 보이지 않는 자세·속도·접촉상태 등을 추론하는 과정이다.

SLAM

미지 또는 부분 미지 환경의 지도와 지도 안의 로봇 위치를 함께 추정하는 문제다.

ZMP

지지면에서 지면반력에 의한 수평축 모멘트가 0이 되는 점이다. 모든 동적 접촉의 보편 충분조건은 아니다.

MPC

유한 시간구간의 후보 행동과 제약을 반복 계산하고 첫 입력을 실행한 뒤 새 측정으로 다시 푸는 제어다.

강화학습

환경과의 상호작용 데이터에서 기대 누적보상을 높이는 정책을 학습하는 방법이다.

Domain Randomization

훈련 중 물리·시각 파라미터와 지연·잡음 등을 바꾸는 Sim-to-Real 기법 가운데 하나다.

Sim-to-Real

시뮬레이션에서 얻은 정책·지식·데이터를 현실 시스템으로 이전하는 더 넓은 문제다.

VLA

시각·언어 조건을 로봇 행동 또는 행동목표와 연결하는 모델군이다.

Teleoperation

인간이 통신과 인터페이스를 통해 로봇을 원격 제어하는 운영·데이터 수집 방식이다.

모방학습

전문가 시범의 관측–행동 관계로 정책을 학습하는 방법이다.

Fleet Learning

여러 로봇의 현장 데이터·개입·실패를 모아 검증한 모델을 다시 배포하는 운영 파이프라인을 가리키는 실무적 우산 용어다.

BMS

전압·전류·온도를 측정하고 SOC·SOH를 모델과 이력으로 추정하며 보호·밸런싱·충전을 관리한다.

Thermal Derating

온도·열모델·보호한계에 따라 전류·토크·전력·충전률을 제한하는 보호전략이다.

MTBF / MTTR

정해진 조건의 고장 사이 평균 운전시간과 수리·복구시간 지표다. 운영 정의와 경계를 함께 적어야 한다.

TCO

구매 이후 에너지·정비·소프트웨어·감독·다운타임 등을 포함한 총소유비용이다.

RaaS

구독·사용량·성과기반 계약을 포함하는 Robot as a Service 계약군이다.

Cost per Successful Action

품질·시간·안전 기준을 통과해 완료된 물리적 행동 하나에 들어간 총비용이라는 이 책의 경제 프레임이다.

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

후주·출처 및 참고자료

상태 표기 원칙

현재 사실 ≠ 기업 발표 ≠ 미래 목표 ≠ 분석가 추정 ≠ 미디어 해석. 본문은 이 구분과 2026-09-24 사실 기준일을 상속한다.

검증 원장

ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

ROBOT_ALPHA_OMEGA_CANONICAL_AUDIT_MATRIX_v1.0.md

본문·검증 원장에 표시된 공개 참고자료

1. Tesla
https://www.tesla.com/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

2. Figure
https://www.figure.ai/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

3. Unitree
https://www.unitree.com/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

4. Boston Dynamics
https://bostondynamics.com/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

5. Boston Dynamics
https://bostondynamics.com/products/atlas/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

6. Agility Robotics
https://www.agilityrobotics.com/content/agility-unveils-digit-5-humanoid-robot-built-for-cooperatively-safe-work-at-scale
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

7. UBTECH
https://www.ubtrobot.com/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

8. NVIDIA
https://www.nvidia.com/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

9. Agility Robotics
https://www.agilityrobotics.com/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

10. Hyundai Motor Group
https://www.hyundaimotorgroup.com/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

11. Apptronik
https://apptronik.com/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

12. Google DeepMind
https://deepmind.google/
기록 위치: ROBOT_ALPHA_OMEGA_DRAFT02_FULL_MANUSCRIPT_v0.4.md

13. Tesla 2025 Form 10-K
https://ir.tesla.com/_flysystem/s3/sec/000162828026003952/tsla-20251231-gen.pdf
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

14. Tesla Q1 2026 Update
https://assets-ir.tesla.com/tesla-contents/IR/TSLA-Q1-2026-Update.pdf
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

15. Tesla AI & Robotics
https://www.tesla.com/AI
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

16. BMW Group: Figure deployment retrospective
https://www.press.bmwgroup.com/global/article/detail/T0455864EN/bmw-group-to-deploy-humanoid-robots-in-production-in-germany-for-the-first-time?language=en
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

17. Figure: Introducing Figure 03
https://www.figure.ai/news/introducing-figure-03
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

18. Boston Dynamics: New Atlas product announcement
https://bostondynamics.com/blog/boston-dynamics-unveils-new-atlas-robot-to-revolutionize-industry/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

19. GXO-Agility multi-year commercial agreement
https://www.agilityrobotics.com/content/gxo-signs-industry-first-multi-year-agreement-with-agility-robotics
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

20. Apptronik-Mercedes-Benz commercial agreement
https://apptronik.com/news-collection/apptronik-and-mercedes-benz-enter-commercial-agreement
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

21. Apptronik: Apollo 2
https://apptronik.com/apollo/apollo-2
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

22. Unitree G1
https://www.unitree.com/g1/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

23. Unitree R1
https://www.unitree.com/R1/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

24. Unitree H1
https://www.unitree.com/h1/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

25. UBTECH 2025 Annual Results, HKEX
https://www.hkexnews.hk/listedco/listconews/sehk/2026/0331/2026033102607.pdf
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

26. AgiBot: 10,000th robot milestone
https://www.agibot.com/article/231/detail/53.html
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

27. AgiBot: 15,000th robot milestone
https://www.agibot.com/article/231/detail/82.html
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

28. Fourier: GR-3 announcement
https://www.fftai.com/newsroom-humanoid/26
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

29. Fourier Intelligence: About
https://www.fftai.com/about-us
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

30. Google DeepMind: RT-2
https://deepmind.google/blog/rt-2-new-model-translates-vision-and-language-into-action/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

31. Google DeepMind: Gemini Robotics models
https://deepmind.google/models/gemini-robotics/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

32. Google DeepMind: Gemini Robotics 2
https://deepmind.google/blog/gemini-robotics-2-brings-whole-body-intelligence-to-robots/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

33. NVIDIA: Isaac GR00T N1 and simulation frameworks
https://nvidianews.nvidia.com/news/nvidia-isaac-gr00t-n1-open-humanoid-robot-foundation-model-simulation-frameworks
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

34. NVIDIA Robotics
https://www.nvidia.com/en-us/industries/robotics/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

35. NVIDIA Developer: Isaac GR00T end-to-end policies
https://developer.nvidia.com/blog/develop-humanoid-robot-policies-end-to-end-with-nvidia-isaac-gr00t/
기록 위치: ROBOT_ALPHA_OMEGA_UPDATED_EVIDENCE_P0_v1.0.md

36. IEA – Rare Earth Elements 2026, Executive Summary
https://www.iea.org/reports/rare-earth-elements/executive-summary
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

37. IEA – Global EV Outlook 2026: Manufacturing and trade
https://www.iea.org/reports/global-ev-outlook-2026/manufacturing-and-trade
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

38. IFR – World Robotics 2025 Industrial Robots, Executive Summary
https://ifr.org/img/worldrobotics/Executive_Summary_WR_2025_Industrial_Robots.pdf
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

39. Nabtesco – Results Briefing Material for FY2025
https://www.nabtesco.com/cms/wp-content/uploads/Results_Briefing_Material_for_FY2025_e.pdf
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

40. Harmonic Drive Systems – 2025 Investor Presentation
https://www.hds.co.jp/Portals/0/files/english/ir/data/investor_event/pdf/%E3%80%90WEB%E3%80%91E-setsumeikai%2020250520-v3.pdf
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

41. 중국 공업정보화부 – 인형로봇 혁신발전 지도 의견
https://www.miit.gov.cn/jgsj/kjs/wjfb/art/2023/art_50316f76a9b1454b898c7bb2a5846b79.html
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

42. IFR – Robotics R&D Programs: Asia, Europe and America 2025
https://ifr.org/ifr-press-releases/news/robotics-research-goverment-programs-asia-europe-and-america-2025
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

43. 일본 경제산업성 – AI·로보틱스 검토회 보고
https://www.meti.go.jp/shingikai/mono_info_service/ai_robotics/20251008_report.html
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

44. 미국 에너지부 – 희토류 공급망 강화 지원 발표
https://www.energy.gov/cmm/articles/energy-department-announces-134-million-funding-strengthen-rare-earth-element-supply
기록 위치: ROBOT_ALPHA_OMEGA_P1_EVIDENCE_READALOUD_LEDGER_v1.0.md

45. Texas Instruments – Trapezoidal Control of BLDC Motors
https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/171/sprabz4.pdf
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

46. Harmonic Drive – Strain Wave Gear Technology
https://www.harmonicdrive.net/technology
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

47. Nabtesco – Precision Reduction Gear RV 구조
https://precision.nabtesco.com/en/products/introduction/
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

48. THK – Features of the Ball Screw
https://tech.thk.com/en/products/pdf/en_b15_006.pdf
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

49. Schaeffler – Ball and Planetary Roller Screw Reference
https://www.schaeffler.com/remotemedien/media/_shared_media/08_media_library/01_publications/schaeffl

er_2/catalogue_1/downloads_6/prg_us_us.pdf
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

50. Series Elastic Actuator modeling paper
https://arxiv.org/abs/0912.3956
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

51. Quasi-Direct Drive Actuation paper
https://arxiv.org/abs/2004.00467
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

52. OSTI - Robotic Actuator Torque Density and Thermal Limits
https://www.osti.gov/servlets/purl/1369514
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

53. Renishaw - What is an encoder?
https://www.renishaw.com/en/what-is-an-encoder--47256
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

54. ATI - Net F/T six-component measurement
https://www.ati-ia.com/products/ft/ft_NetFT.aspx
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

55. CMU Robotics Institute - GelSight shear and slip measurement
https://publications.ri.cmu.edu/measurement-of-shear-and-slip-with-a-gelsight-tactile-sensor
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

56. Using Inertial Sensors for Position and Orientation Estimation
https://arxiv.org/abs/1704.06053
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

57. Durrant-Whyte & Bailey - SLAM tutorial
https://people.eecs.berkeley.edu/~pabbeel/cs287-fa09/readings/Durrant-Whyte_Bailey_SLAM-tutorial-I.pdf
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

58. Vukobratovi□ & Borovac - Zero-Moment Point
https://www.cs.cmu.edu/~cga/legs/vukobratovic.pdf
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

59. Model Predictive Control with Environment Adaptation for Legged Locomotion
https://arxiv.org/abs/2105.05998
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

60. Sutton & Barto - Reinforcement Learning: An Introduction
https://incompleteideas.net/book/the-book-2nd.html

기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

61. Domain Randomization for Sim-to-Real
https://arxiv.org/abs/1703.06907
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

62. Learning Dexterous In-Hand Manipulation
https://arxiv.org/abs/1808.00177
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

63. RT-2 – Vision-Language-Action Models
https://proceedings.mlr.press/v229/zitkovich23a.html
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

64. OpenVLA
https://arxiv.org/abs/2406.09246
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

65. DAgger – A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning
https://proceedings.mlr.press/v15/ross11a.html
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

66. Fleet-DAgger – Interactive Robot Fleet Learning
https://arxiv.org/abs/2206.14349
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

67. Texas Instruments – BMS Breakers and Designing With TPS1210-Q1
https://www.ti.com/lit/SLUAAS8
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

68. Texas Instruments – Battery fuel gauges and gauging algorithms
https://www.ti.com/product-category/battery-management-ics/battery-fuel-gauges/overview.html
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

69. NASA – Designing for Maintainability and System Availability
https://ntrs.nasa.gov/api/citations/19970013665/downloads/19970013665.pdf
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

70. NIST Handbook 135e2025 – Life-Cycle Costing
https://nvlpubs.nist.gov/nistpubs/hb/2025/NIST.HB.135e2025.pdf
기록 위치: ROBOT_ALPHA_OMEGA_TECHNICAL_ACCURACY_GATE_v1.0.md

다음 펼침면 또는 PART 시작을 위해 의도적으로 비운 면

도판 목록

모든 도판은 회사 로고·제품 도면·저작권 이미지를 사용하지 않은 자체 제작 벡터 개념도다.

V01. 휴머노이드 전체 해부도

배치: 제1장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V02. 모터 – 전기에서 토크로

배치: 제2장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V03. 스트레인 웨이브 감속기

배치: 제3장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V04. 통합 액추에이터

배치: 제4장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V05. 손과 촉각

배치: 제5장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V06. 센서융합과 상태추정

배치: 제6장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V07. 걷기와 외란 회복

배치: 제7장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V08. 배터리와 열

배치: 제8장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V09. BODY에서 BRAIN까지

배치: 제10장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V10. VLA 행동 페루프

배치: 제11장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V11. 로봇 데이터 피라미드

배치: 제12장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V12. Sim↔Real 순환

배치: 제13장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V13. DEMO에서 MASS SCALE까지

배치: 제14장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V14. 휴머노이드 공급망

배치: 제16장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V15. 미국·중국 생태계

배치: 제18장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V16. 세 개의 플라이휠

배치: 제20장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V17. Cost per Successful Action

배치: 제21장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V18. 직업에서 작업으로

배치: 제22장 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

V19. ALPHA에서 OMEGA까지

배치: 에필로그 · 상태: BOOK MASTER FIGURE · 수치 비례가 아닌 개념도

색인 후보

이 목록은 최종 인쇄판의 페이지 기반 색인을 만들기 위한 항목 후보이며, 현 판본에서는 장 단위 위치를 제공한다.

행동의 가격

중심 질문, 15장, 21장, 에필로그

BODY × BRAIN × DATA × FACTORY × SUPPLY CHAIN

0장, 20장, 에필로그

모터·토크·토크밀도

1–2장, 4장, 9장

감속기·백래시·역구동성

3–4장, 19장

액추에이터·SEA·QDD

4장

손·촉각·미끄럼

5–6장

센서융합·상태추정·SLAM

6장

보행·ZMP·MPC·외란 회복

7장

배터리·BMS·가동전략

8장

열·지속성능·Thermal Derating

9장

제어계층·세 개의 뇌

10장

VLA·행동 폐루프

11장

행동 데이터·원격조작·모방학습

12장

시뮬레이션·Sim-to-Real

13장

DEMO·PILOT·DEPLOYMENT·COMMERCIAL OPERATION·MASS SCALE

14장

BOM·제조원가·TCO

15장, 21장

광물·부품·공급망

16장

미국·중국·연결 산업

17-18장

MTBF·MTTR·가동률

19장, 21장

대량생산·수율·세 플라이휠

20장

RaaS·시간당 비용

21장

직업·작업 분해·노동분업

22장, 에필로그