

나노로봇 집합 제어 시뮬레이션: 고급 물리 기반 다중 에이전트 시스템

1. 시뮬레이션 개요

본 시뮬레이션은 순수 JavaScript로 구현된 나노로봇(nanorobot) 집합 제어 시스템으로, 나노미터 규모(0.1~100 nm)의 초소형 로봇들이 유체 환경에서 협력적으로 작업을 수행하는 과정을 모델링합니다. 이 시스템은 의료(약물 전달, 종양 타겟팅), 환경 모니터링(오염 물질 감지), 산업 응용 등 다양한 분야에서 활용될 수 있는 나노로봇 기술의 핵심 메커니즘을 시뮬레이션합니다.

시뮬레이션의 핵심은 **Model Predictive Control(MPC)**와 **파티클 필터(Particle Filter)** 기반의 다중 에이전트 제어 시스템으로, Navier-Stokes 방정식을 이용한 유체역학(FDM, Finite Difference Method), 회전 동역학(Rotational Dynamics), 확률적 상태 추정, 그리고 분산 최적화(ADMM, Alternating Direction Method of Multipliers)를 통합하여 현실적인 나노 규모 물리 현상을 재현합니다.

1.1 주요 특징

- 고급 물리 엔진:** NS 방정식 FDM, 회전 동역학, Euler-Maruyama 확률적 적분
- 지능형 제어:** MPC(N=20 horizon), 파티클 필터(최대 50,000 입자), ADMM 분산 최적화
- 통신 프로토콜:** 무선(THz), 광학(FRET), 분자/화학 통신 시뮬레이션
- 클러스터 관리:** 리더/워커 역할 기반 계층적 제어, 동적 재구성
- 실시간 모니터링:** 11개 메트릭(집합 달성률, 에너지, 통신 횟수 등)
- 다양한 시나리오:** 자기장 제어, 장애물 회피, 대형(원형/선형) 형성

2. 물리 엔진 및 수학적 모델링

2.1 Navier-Stokes 방정식 기반 유체역학

시뮬레이션은 나노로봇이 유체(혈관, 세포 환경 등) 내에서 이동할 때 받는 힘을 Navier-Stokes 방정식으로 모델링합니다:



$$\partial v / \partial t + (v \cdot \nabla) v = -\nabla p / \rho + \mu \nabla^2 v + f$$

구현 세부사항:

- 압력 구배(Pressure Gradient):** $\nabla p = [0.1, 0.05, 0.0]$ 로 설정되어 유체의 압력 차이로 인한 힘을 시뮬레이션
- 점성 항(Viscous Term):** $\mu \nabla^2 v$ 는 라플라시안 연산자를 통해 점성력 계산. 기본 점성 계수는 0.001 Pa·s(물과 유사)
- 대류 항(Convective Term):** $(v \cdot \nabla) v \approx \rho v^2$ 로 근사하여 유체의 운동량 전달 모델링

- 유체 밀도: $\rho = 1000 \text{ kg/m}^3$ (물의 밀도)

Finite Difference Method(FDM)를 사용하여 공간 미분을 이산화하고, 각 타임스텝($DT = 0.008$ 초)마다 나노로봇에 작용하는 유체 힘을 계산합니다.

2.2 회전 동역학 및 쿼터니언 기반 방향 제어

나노로봇의 회전 운동은 쿼터니언(quaternion) 표현을 사용하여 김벌 락(gimbal lock) 문제를 방지하고 수치적 안정성을 확보합니다.

각운동량 방정식:



$$\tau = J \cdot d\omega/dt$$

파라미터:

- 관성 모멘트: $J = 1 \times 10^{-20} \text{ kg}\cdot\text{m}^2$
- 각가속도: $\omega' = \tau / J$
- 토크 소스:
 - 무작위 브라운 토크: $0.01 \times \text{random}()$
 - 자기장 토크(자성 재료): $M \times B = 10^{-18} \times 0.1 = 10^{-19} \text{ N}\cdot\text{m}$

쿼터니언 업데이트:



$$\begin{aligned} \dot{q} &= 0.5 \times [0, \omega_x, \omega_y, \omega_z] \otimes q \\ q(t+\Delta t) &= q(t) + \dot{q} \cdot \Delta t \\ q &\leftarrow q / \|q\| \text{ (정규화)} \end{aligned}$$

이를 통해 나노로봇의 3차원 방향을 정확하게 추적하며, 특히 자기장을 이용한 비침습적 제어 시나리오에서 중요합니다.

2.3 Euler-Maruyama 확률적 적분

나노 규모에서는 브라운 운동(Brownian motion)이 지배적이므로, 확률 미분 방정식(SDE)을 사용합니다:



$$dx = a(x,t)dt + b(x,t)dW$$

구성 요소:

- **드리프트 항:** $a(x,t) = u(t) + F_{NS} - \gamma v$ (제어 입력 + NS 힘 - 점성 저항)
- **확산 항:** $b(x,t) = \sqrt{(2k_{BT}/\gamma)} \cdot \Delta t$
 - **Stokes 저항:** $\gamma = 6\pi\mu r$ (r: 로봇 반지름 5~50 nm)
 - **열 에너지:** k_{BT} (T = 300~310 K)
 - **확산 계수:** $D = k_{BT}/\gamma$

이산화:



$$v(t+\Delta t) = v(t) + [u + F_{NS} - \gamma v]\Delta t + \sqrt{(2k_{BT}/\gamma \cdot \Delta t)} \cdot \xi$$

$$x(t+\Delta t) = x(t) + v(t+\Delta t) \cdot \Delta t$$

여기서 $\xi \sim N(0,1)$ 은 가우시안 백색 잡음입니다. 이는 나노 규모의 불확실성과 열 요동을 현실적으로 반영합니다.

2.4 에너지 모델 및 열역학

각 나노로봇은 제한된 에너지 예산을 가지며, 활동에 따라 에너지가 소모됩니다:

에너지 소비:

- 초기 에너지: 100 nJ
- 통신 에너지: 1 nJ/전송
- 추진 에너지: $0.5 \text{ nJ} \times \sqrt{(u^2)}$
- 센싱 에너지: 0.5 nJ/샘플

온도 모델:

- 기본 온도: 300 K (체온 환경)
- 변동 범위: $\pm 10 \text{ K}$ (대사 활동, 환경 영향)
- 볼츠만 상수: $k_B = 1.38 \times 10^{-23} \text{ J/K}$

에너지가 0 이하로 떨어지면 로봇은 비활성화되며, 이는 배터리 고갈 또는 연료(포도당/산소) 부족을 시뮬레이션합니다.

3. 지능형 제어 알고리즘

3.1 Model Predictive Control (MPC)

MPC는 예측 모델을 사용하여 미래 N 스텝(horizon)에 걸쳐 최적 제어 입력을 계산합니다.

최적화 문제:



$$\min \sum_{t=0 \text{ to } N} ||x(t) - x_target||^2 + \lambda ||u(t)||^2$$

subject to: $x(t+1) = f(x(t), u(t))$

구현 세부사항:

- **예측 구간:** $N = 20$ 스텝 (조정 가능: 5~50)
- **목표 속도:** $v_target = [1.0, 0.0, 0.0]$ m/s (x 방향 이동)
- **최적화 방법:** 간단한 무작위 탐색(10 반복) - 실제 구현에서는 경사하강법/Newton 방법 사용 가능
- **비용 함수:** 추적 오차의 제곱합
- **제어 제약:** $u \in [0, 0.5]$ m/s

MPC는 장애물 회피, 대형 유지, 에너지 최적화 등 복잡한 제약 조건을 처리할 수 있습니다.

3.2 파티클 필터 (Particle Filter)

나노 규모의 높은 불확실성(센서 노이즈, 브라운 운동)을 다루기 위해 베이지안 상태 추정을 사용합니다.

알고리즘 단계:

1. **초기화:** N_p 개 입자 생성 (기본 1,000개, 최대 50,000개)



$$p_i \sim N(x_0, \Sigma_0), w_i = 1/N_p$$

2. **예측(Propagation):** Euler-Maruyama로 각 입자 전파



$$p_i(t+1) = p_i(t) + noise_i \cdot \Delta t$$

$noise \sim N(0, 0.01)$

3. **가중치 업데이트:** 측정값과 입자 상태의 우도 계산



$$w_i \leftarrow \exp(-||p_i[0] - \text{measurement}||^2/2)$$

4. 정규화:



$$w_i \leftarrow w_i / \sum w_i$$

5. 리샘플링: Systematic resampling으로 유효 입자 유지



누적 분포 함수(CDF) 기반 선택

6. 상태 추정: 가중 평균 계산



$$x_{\text{est}} = \sum (w_i \cdot p_i)$$

성능 고려사항:

- 1,000 입자: 실시간 성능 우선
- 50,000 입자: 정확도 우선(브라우저 부하 주의)

3.3 ADMM 분산 최적화

여러 나노로봇이 협력적으로 최적화 문제를 해결할 때 사용합니다.

ADMM 공식:



$$\min \sum f_i(x_i) \text{ subject to } x_i = z \text{ (합의 제약)}$$

반복 알고리즘:



for $k = 1$ to K_{\max} :

로컬 업데이트 (각 로봇 독립적)

$$x_i^{(k+1)} = \operatorname{argmin} f_i(x_i) + (\rho/2) \|x_i - z^k + \lambda_i^k/\rho\|^2$$

글로벌 업데이트 (평균 합의)

$$z^{(k+1)} = (1/N) \sum x_i^{(k+1)}$$

듀얼 변수 업데이트

$$\lambda_i^{(k+1)} = \lambda_i^k + \rho(x_i^{(k+1)} - z^{(k+1)})$$

파라미터:

- **증강 라그랑주 계수:** $\rho = 1.0$
- **반복 횟수:** $K_{\max} = 5$
- **수렴 기준:** $\|x_i - z\| < \epsilon$

ADMM은 통신 제약이 있는 분산 환경에서 효율적이며, 각 로봇이 로컬 계산만 수행하고 글로벌 정보만 공유합니다.

4. 통신 프로토콜 및 네트워크 아키텍처

나노로봇 간 통신은 Internet of Nano-Things(IoNT) 개념을 기반으로 하며, 나노 규모의 제약(에너지, 크기, 환경 간섭)을 고려합니다.

4.1 통신 계층 구조 (OSI 모델 적응)

물리 계층(Physical Layer):

- **전송 매체:** 분자(화학 신호), THz 전자기파(~~0.1 ± 0.1 THz~~), 초음파(~~10~~300 MHz)
- **통신 거리:** 단거리 최적화(수십~수백 μm)
- **변조 방식:** Pulse Position Modulation(PPM) - 에너지 효율성

네트워크 계층(Network Layer):

- **주소 지정:** 기능 중심 네트워킹(Function Centric Networking)
 - IP 주소 대신 역할 기반 주소 사용 (예: LEADER-1, WORKER-3)
- **라우팅:** 나노라우터가 클러스터 데이터 집계

전송 계층(Transport Layer):

- **신뢰성:** PPM과 ERPPM(Event Recognition PPM)으로 데이터 신뢰성 보장
- **흐름 제어:** 에너지 기반 제어 - 저에너지 로봇은 전송 빈도 감소

애플리케이션 계층(Application Layer):

- **인터페이스:** 바이오사이버 인터페이스(Biocyber Interface)
- **응용:** 약물 전달, 환경 모니터링, 진단

- 보안: PSO-ANN 기반 암호화

4.2 통신 방법 상세

4.2.1 무선 통신 (THz Electromagnetic)

특징:

- 주파수 대역: 0.1~10 THz
- 안테나: 그래핀 기반 나노 안테나(100~1000 nm)
- 데이터 속도: 최대 수 Gbps (이론적)
- 통신 거리: 수 mm (높은 감쇠로 제한)
- 에너지 소비: 1 nJ/전송

프로토콜:

- THz 무선 나노센서 네트워크: PPM으로 정보 인코딩
- 이벤트 기반 전송: ERPPM으로 이상 징후 감지 시만 전송
- 거리 제약: MAX_COMM_DIST = 100 nm 이내에서만 통신

구현:



javascript

```
if (distance < MAX_COMM_DIST && Math.random() < 0.1) {
  wirelessCommCounter++;
  robot.energy -= ENERGY_COMM;
}
```

4.2.2 광학 통신 (FRET 기반)

FRET(Förster Resonance Energy Transfer):

- 원리: 분자 간 비복사 에너지 전달
- 거리 의존성: 1~10 nm (매우 근거리)
- 효율: 거리의 6승에 반비례 (R^{-6})
- 응용: 밀집 환경에서 고신뢰 통신

프로토콜:

- 멀티홉 라우팅: FRET 체인을 통한 장거리 전달
- 신뢰성: 분자 간 직접 결합으로 높은 신뢰성
- 에너지: 매우 낮은 소비(양자 효과 활용)

4.2.3 분자/화학 통신

메커니즘:

- **정보 캐리어:** 칼슘 이온, E-cadherin, 기타 생체 분자
- **전파 방식:** 확산 기반(브라운 운동)
- **수신:** 그래핀 기반 화학 센서

프로토콜:

- **화학 기울기 기반:** 농도 기울기를 따라 방향 결정
 - 종양 세포 탐지: E-cadherin 기울기 추적
 - 클러스터 형성: 리더 로봇이 화학 신호 방출
- **분자 확산:** 메시지를 분자로 인코딩
 - 데이터 속도: 매우 낮음(수 bps)
 - 생체 적합성: 매우 높음

장단점:

- 장점: 생체 적합성, 저에너지, 환경 친화적
- 단점: 느린 전파(확산 속도), 분자 충돌로 인한 간섭

4.2.4 음향 통신

특징:

- **주파수:** 10~300 MHz (초음파)
- **통신 거리:** 생체 내 100 μ m
- **데이터 속도:** 최대 10^4 bps
- **조직 투과성:** 우수

프로토콜:

- **방향성 빔:** 주파수 기반으로 데이터 속도 향상
- **브로드캐스트:** 이동형 로봇이 고정형 로봇에 메시지 전달
- **집합 제어:** 링셋 구조로 방향성 강화
- **안전 제약:** 압력 한계 1500 Pa 이내 유지(캐비테이션 방지)

4.3 통신 카운터 및 모니터링

시뮬레이션은 다음 통신 메트릭을 추적합니다:

- **무선 통신 카운트:** THz 기반 전송 횟수
- **광학 통신 카운트:** FRET 기반 전송 횟수
- **클러스터 신호 전송:** 리더→워커 제어 신호 횟수
- **클러스터 간 통신:** 리더 간 협력 통신 횟수

5. 클러스터 제어 시스템

클러스터 제어는 나노로봇 집단을 소규모 그룹으로 조직화하여 협력적 작업을 수행하는 핵심 메커니즘입니다.

5.1 클러스터 형성 알고리즘

형성 기준:

- **그룹 크기:** 10개 로봇당 34개 클러스터(평균 23 로봇/클러스터)
- **근접성:** 공간적으로 가까운 로봇들을 그룹화
- **역할 할당:** 각 클러스터에 1개 리더(LEADER), 나머지는 워커(WORKER)

리더 선출 기준:



$$\text{leaderScore} = \text{health} \times 0.5 + \text{energy} \times 0.3 + (1 - \text{positionVariance}) \times 0.2$$

가장 높은 점수를 가진 로봇이 리더로 선출됩니다.

동적 재구성:

- **재선출 조건:**
 - 리더의 건강도가 0.3 이하로 하락
 - 리더의 에너지가 20 nJ 이하
 - 환경 변화(혈관 분기점, 장애물)
- **재구성 프로토콜:**
 1. 클러스터 내 모든 로봇의 상태 평가
 2. 리더 점수 재계산
 3. 새 리더 선출 및 역할 재할당
 4. 화학 신호 또는 THz로 재할당 명령 전송

5.2 리더 기반 제어 메커니즘

리더의 역할:

1. **경로 계획:** MPC로 클러스터의 목표 궤적 계산



$$\text{target} = \text{leader.position} + \text{direction} \times \text{speed}$$

2. **상태 모니터링:** 워커 로봇의 건강도, 위치, 에너지 주기적 수집



for each worker in cluster:

collect(health, position, energy, sensorData)

3. 제어 신호 전송:

- 분자 통신: 화학 기울기 신호 방출 → 워커가 기울기 추적
- THz 통신: PPM 펄스로 속도/방향 명령 전송
- 음향 통신: 초음파 버스트로 집합 명령

4. 피드백 루프:



sensorData ← collect from workers

if pH_change > threshold:

send drug_release_command

워커의 역할:

1. 리더 추적: 리더의 제어 신호를 수신하여 속도 조정



velocity_adjustment = leaderSignal - currentVelocity

2. 센서 데이터 수집: 압력, pH, 온도 측정



pressure = random() × 20 // 0~20 kPa

pH = 7 + gaussian(0, 0.1) // pH 7 ± 0.1

3. 리더 보고: 주기적으로 상태 전송(10% 확률)



if random() < 0.1:

send(health, position, sensorData) to leader

5.3 클러스터 제어 신호 프로토콜

신호 유형:

1. 방향 제어 신호:

- 페이로드: [target_x, target_y, target_z, speed]
- 전송: 리더 → 모든 워커
- 빈도: 매 스텝(DT = 0.008초마다)

2. 대형 유지 신호:

- 원형 대형:



for i in 0 to N_workers:

angle = $2\pi \times i / N_workers$

position[i] = leader.position + radius \times [cos(angle), sin(angle), 0]

• 선형 대형:



for i in 0 to N_workers:

position[i] = leader.position + i \times spacing \times direction

3. 이벤트 기반 신호:

- 장애물 감지: AVOID [obstacle_position]
- 목표 도달: RELEASE_DRUG [target_cell]
- 응급 상황: EMERGENCY_REGROUP

에너지 효율성:

- 클러스터 내 통신은 최소화(10% 확률)
- 멀티홉 라우팅으로 장거리 통신 에너지 절감
- 이벤트 기반 전송으로 불필요한 통신 방지

5.4 클러스터 간 협력

시나리오:

- **다중 목표:** 클러스터 A는 종양 세포 A 타겟, 클러스터 B는 종양 세포 B 타겟
- **자원 공유:** 에너지가 부족한 클러스터에 워커 재할당
- **정보 공유:** 한 클러스터가 발견한 경로 정보를 다른 클러스터와 공유

프로토콜:



```
if interClusterEvent:
    leaderA.send(information) to leaderB // 리더 간 직접 통신
    interClusterCommCount++
```

6. 센서 시스템 및 환경 인식

6.1 센서 유형

압력 센서:

- 범위: 0~20 kPa
- 정확도: ± 0.1 kPa (노이즈 표준편차)
- 응용: 혈압 모니터링, 조직 압력 감지

pH 센서:

- 범위: 0~14
- 정확도: ± 0.1 pH
- 응용: 종양 미세환경 감지(pH 6.5~7.0), 위장관 pH 측정

온도 센서:

- 범위: 290~320 K
- 정확도: ± 1 K
- 응용: 염증 부위 감지(국소 온도 상승)

6.2 센서 데이터 처리

노이즈 모델:



```
measured_value = true_value + gaussian(0, SENSOR_NOISE_STD)
SENSOR_NOISE_STD = 0.1
```

데이터 필터링:

- 이동 평균 필터로 노이즈 제거
- 파티클 필터로 센서 융합(다중 센서 데이터 통합)

이벤트 감지:



```
if pressure > 15: // 고혈압 감지
    trigger_alert()
if pH < 6.8: // 종양 환경 감지
    release_drug()
```

6.3 센서 데이터 수집 카운터

시뮬레이션은 각 로봇의 센서 샘플링 횟수를 추적합니다:



```
sensorDataCollected += 1
robot.energy -= ENERGY_SENSE // 0.5 nJ/샘플
```

7. 시뮬레이션 파라미터 및 사용자 인터페이스

7.1 제어 파라미터

로봇 설정:

- 개수: 1~50개 (기본 10개)
- 크기:
 - 10 nm ($r=5$ nm): 바이러스 크기, 높은 브라운 운동
 - 100 nm ($r=50$ nm): 박테리아 크기, 낮은 브라운 운동
- 재료:
 - 비자성(non-magnetic): 자기장 영향 없음
 - 자성(magnetic): 자기장으로 제어 가능(토크 추가)

환경 설정:

- 점성: 0.0001~0.01 Pa·s
 - 0.001: 물(혈액)
 - 0.01: 점액(위장관)
- 자기장: 체크박스로 활성화/비활성화
 - 강도: 0.1 T
 - 자기 모멘트: 10^{-18} A·m²
- 장애물: 0~20개
 - 무작위 위치 및 크기
 - 충돌 감지 및 회피
- 유동 강도: 0~1.0
 - 혈류, 림프액 흐름 시뮬레이션

알고리즘 설정:

- **파티클 필터 입자 수:** 1,000~50,000
 - 1,000: 실시간 성능
 - 50,000: 최대 정확도(브라우저 부하 주의)
- **MPC 구간(Horizon):** 5~50 스텝
 - 5: 빠른 반응, 단기 계획
 - 20: 균형(기본값)
 - 50: 장기 최적화, 높은 계산 비용

시뮬레이션 설정:

- **스텝 수:** 10~500 (기본 100)
- **시간 간격:** DT = 0.008초 (125 Hz 업데이트)
- **대형 형성:**
 - 없음: 자유 이동
 - 원형: 리더 중심 원형 배치
 - 선형: 일렬 대형

7.2 실시간 메트릭

시뮬레이션은 11개의 주요 메트릭을 실시간으로 모니터링합니다:

1. 스텝 (Step):

- 현재 시뮬레이션 진행도 표시
- 형식: `currentStep / totalSteps`

2. 집합 달성률 (Ensemble Success Rate):



$ensembleRate = (\text{목표 달성 로봇 수} / \text{전체 로봇 수}) \times 100\%$
 목표 달성 기준: $||\text{position} - \text{target}|| < \text{threshold}$

3. 개별 평균 성공률 (Individual Average Rate):



$individualRate = (87\% \times \text{활성 로봇 수} / \text{전체 로봇 수})$
 $P_INDIVIDUAL = 0.87$ (개별 성공 확률)

4. 활성 로봇 (Active Robots):



$activeCount = \sum(robot.active == true)$

비활성화 조건: $energy \leq 0$ 또는 $health \leq 0$

5. 무선 통신 카운트 (Wireless Communication Count):

- THz 기반 전송 총 횟수
- 통신 발생 조건: 거리 < 100 nm, 확률 10%

6. 광학 통신 카운트 (Optical Communication Count):

- FRET 기반 전송 총 횟수
- 매우 근거리(1~10 nm) 통신

7. 센서 데이터 수집 (Sensor Data Collected):

- 압력/pH 센싱 총 횟수
- 각 샘플링마다 0.5 nJ 소모

8. 클러스터 신호 전송 (Cluster Signals Sent):

- 리더→워커 제어 신호 총 횟수
- 대형 유지, 속도 조정 명령 포함

9. 평균 에너지 (Average Energy):



$avgEnergy = (\sum robot.energy) / numRobots$

단위: nJ (nanojoules)

10. 리더 재선정 횟수 (Leader Reelections):

- 클러스터 리더 교체 총 횟수
- 재선정 조건: 리더 건강도 < 0.3 또는 에너지 < 20 nJ

11. 클러스터 간 통신 (Inter-Cluster Communication):

- 리더 간 협력 통신 총 횟수
- 정보 공유, 자원 재분배

7.3 시각화 캔버스

600×400 픽셀 캔버스에 다음을 렌더링:

- 로봇 표현:
 - 리더: 빨간색 원 (더 큰 크기)
 - 워커: 파란색 원

- 비활성: 회색 원
- 크기: 로봇 반지름에 비례
- 장애물: 검은색 원 또는 다각형
- 경로 추적: 로봇의 이동 궤적(선)
- 클러스터 시각화: 같은 클러스터는 연결선으로 표시
- 실시간 업데이트: requestAnimationFrame으로 부드러운 애니메이션

7.4 로봇 상태 테이블

각 로봇의 상세 정보를 14개 열로 표시:

1. ID: 로봇 고유 번호 (0~N-1)
2. 역할: LEADER(빨간색) 또는 WORKER(파란색)
3. 클러스터: 소속 클러스터 ID
4. 건강도: 0.0~1.0 (1.0 = 완전 건강)
5. 속도: $|v|$ (m/s)
6. 온도: 290~320 K
7. 위치: (x, y, z) μm 단위
8. 각속도: $|\omega|$ (rad/s)
9. 무선 통신: 해당 로봇의 THz 전송 횟수
10. 광학 통신: 해당 로봇의 FRET 전송 횟수
11. 센서: 압력(kPa) / pH 값
12. 에너지: 잔여 에너지(nJ)
13. 재료: 비자성/자성
14. 활성: 활성/비활성 상태

8. 주요 알고리즘 구현 세부사항

8.1 가우시안 랜덤 생성기

Box-Muller 변환을 사용한 정규분포 난수 생성:



javascript

```
function gaussianRandom(mean = 0, std = 1) {
  let u = 1 - Math.random(); // (0,1]
  let v = Math.random();     // [0,1)
  let z = Math.sqrt(-2.0 * Math.log(u)) * Math.cos(2.0 * Math.PI * v);
  return z * std + mean;
}
```

이는 브라운 운동, 센서 노이즈, 파티클 필터의 확산 모델에 사용됩니다.

8.2 쿼터니언 정규화

수치적 오차 누적을 방지하기 위한 정규화:



javascript

```
function normalizeQuaternion(q) {  
  const norm = Math.sqrt(q[0]2 + q[1]2 + q[2]2 + q[3]2);  
  return [q[0]/norm, q[1]/norm, q[2]/norm, q[3]/norm];  
}
```

매 회전 업데이트 후 호출되어 $|q| = 1$ 유지.

8.3 장애물 생성

무작위 위치 및 크기의 구형 장애물:



javascript

```
function generateObstacles(numObstacles) {  
  let obstacles = [];  
  for (let i = 0; i < numObstacles; i++) {  
    obstacles.push({  
      position: [  
        Math.random() * 2 - 1, // x: [-1, 1] μm  
        Math.random() * 2 - 1, // y: [-1, 1] μm  
        Math.random() * 2 - 1 // z: [-1, 1] μm  
      ],  
      radius: 0.1 + Math.random() * 0.2 // 100~300 nm  
    });  
  }  
  return obstacles;  
}
```

8.4 충돌 감지 및 회피

각 스텝마다 로봇과 장애물 간 거리 계산:



javascript

```

for (let obstacle of obstacles) {
  let distance = Math.sqrt(
    (robot.position[0] - obstacle.position[0])2 +
    (robot.position[1] - obstacle.position[1])2 +
    (robot.position[2] - obstacle.position[2])2
  );

  if (distance < (robotRadius + obstacle.radius)) {
    // 충돌 감지 - 반발력 적용
    let repulsion = obstacle.position.map((p, i) =>
      (robot.position[i] - p) / distance * 0.1
    );
    robot.velocity = robot.velocity.map((v, i) => v + repulsion[i]);
  }
}

```

9. 응용 시나리오

9.1 의료 응용

약물 전달:

1. 나노로봇이 혈관을 통해 이동
2. 리더 로봇이 E-cadherin 기울기를 감지하여 종양 위치 파악
3. 클러스터가 종양 주변에 집결(원형 대형)
4. pH 센서로 종양 미세환경 확인(pH < 6.8)
5. 화학 신호로 약물 방출 명령 전송
6. 워커 로봇들이 동시에 약물 페이로드 방출

진단:

- 혈액 내 바이오마커 감지
- 압력 센서로 동맥경화 진단
- 온도 센서로 염증 부위 탐지

9.2 환경 모니터링

수질 오염 감지:

1. 나노로봇 클러스터를 수원에 배치
2. pH, 압력, 화학 센서로 오염 물질 탐지
3. 리더 로봇이 데이터 집계
4. THz 통신으로 중앙 모니터링 시스템에 전송

대기 질 모니터링:

- 나노 입자 농도 측정
- 화학 센서로 유해 가스 감지
- 실시간 데이터 전송

9.3 군사 및 보안

화학 무기 탐지:

1. 나노로봇 클러스터를 의심 지역에 배치
2. 화학 센서로 독성 물질 탐지
3. 클러스터 간 통신으로 탐지 패턴 형성
4. 고속 THz 통신으로 경보 전송

생물학적 위협 감지:

- 병원균 표면 단백질 감지
- 분자 통신으로 병원균 식별
- 조기 경보 시스템 구축

9.4 산업 응용

나노제조:

- 나노 규모 조립 작업
- 클러스터 협력으로 복잡한 구조 제작
- 정밀 위치 제어(파티클 필터 활용)

품질 검사:

- 반도체 웨이퍼 결함 검사
- 나노 규모 표면 특성 측정

10. 성능 최적화 및 한계

10.1 계산 복잡도

시간 복잡도:

- **기본 시뮬레이션:** $O(N)$ - N개 로봇의 상태 업데이트
- **파티클 필터:** $O(N \times P)$ - P개 입자 처리
 - 1,000 입자: ~1 ms/로봇
 - 50,000 입자: ~50 ms/로봇
- **MPC 최적화:** $O(N \times H \times I)$ - H는 horizon, I는 반복 횟수
 - H=20, I=10: ~0.5 ms/로봇
- **ADMM:** $O(N \times K)$ - K는 반복 횟수
 - K=5: ~0.1 ms

공간 복잡도:

- 로봇 상태: $O(N)$ - 각 로봇당 ~30개 변수
- 파티클: $O(N \times P)$ - 50,000 입자 시 ~6 MB
- 통신 버퍼: $O(N^2)$ - 최악의 경우 전체 통신

10.2 브라우저 제약

권장 설정:

- 로봇 수: ≤ 20
- 파티클 수: $\leq 5,000$
- 시뮬레이션 스텝: ≤ 200

고성능 설정 (강력한 PC):

- 로봇 수: 50
- 파티클 수: 50,000
- 시뮬레이션 스텝: 500

10.3 물리 모델의 한계

단순화된 가정:

- NS 방정식: 라플라시안을 0으로 근사(저해상도 FDM)
- 충돌: 탄성 충돌만 고려(소성 변형 무시)
- 브라운 운동: 백색 잡음 근사(색잡음 효과 무시)
- 자기장: 균일 자기장 가정(실제는 비균일)

개선 방향:

- 고차 FDM 스킴(2차 이상)
- CFD(Computational Fluid Dynamics) 결합
- 분자 동역학(MD) 시뮬레이션 통합
- GPU 가속(WebGL/WebGPU)

11. 확장 가능성

11.1 기능 확장

추가 가능한 모듈:

- 기계 학습: 강화학습으로 제어 정책 학습
- 진화 알고리즘: 최적 파라미터 자동 탐색
- 다중 목표 최적화: Pareto 최적해 탐색
- 실시간 경로 계획: A*, RRT 알고리즘
- 협력 게임 이론: Nash 균형 기반 제어

물리 모델 확장:

- 전기역학: 전하 간 쿨롱 상호작용
- 광학: 광압, 광학 트랩
- 화학 반응: 촉매 작용, 약물 방출 동역학

11.2 스케일 확장

수천 로봇 규모:

- 계층적 클러스터 구조(클러스터의 클러스터)
- 공간 해싱으로 근접 탐색 가속
- 병렬 처리(Web Workers)

장시간 시뮬레이션:

- 적응적 시간 스텝(adaptive time-stepping)
- 이벤트 기반 시뮬레이션
- 체크포인트 및 복원 기능

12. 나노로봇 통신 및 클러스터 제어 이론적 배경

12.1 Internet of Nano-Things (IoNT) 아키텍처

IoNT는 나노 규모 장치들을 인터넷에 연결하는 패러다임으로, 다음 계층으로 구성:

나노 노드(Nano-nodes):

- 가장 작은 단위(10~100 nm)
- 제한된 계산/저장/통신 능력
- 센서, 액추에이터, 기본 처리 유닛

나노 라우터(Nano-routers):

- 나노 노드보다 크고 복잡(100~1000 nm)
- 데이터 집계 및 라우팅 기능
- 클러스터 리더 역할 수행

나노 마이크로 인터페이스(Nano-micro Interface):

- 나노 네트워크와 마이크로/매크로 네트워크 연결
- 프로토콜 변환(분자→전자기)
- 바이오사이버 인터페이스 역할

게이트웨이(Gateway):

- 인터넷 연결
- 클라우드 데이터 전송
- 원격 제어 명령 수신

12.2 기능 중심 네트워킹 (Function Centric Networking)

전통적인 IP 주소 대신 기능 기반 주소 지정:

주소 체계:



Address = <Function, Region, Priority>
예: <DRUG_DELIVERY, TUMOR_SITE_A, HIGH>

라우팅:

- 목적지 IP 대신 기능 이름으로 라우팅
- 동적 서비스 발견
- 자가 조직화 네트워크

장점:

- 확장성: 수천~수만 노드 지원
- 유연성: 노드 추가/제거 용이
- 내결함성: 노드 고장 시 자동 재라우팅

12.3 펄스 위치 변조 (Pulse Position Modulation)

나노 규모 통신의 에너지 효율적 변조 방식:

원리:

- 고정 진폭 펄스를 시간 슬롯에 배치
- 펄스 위치로 정보 인코딩
- 진폭 변조 대비 에너지 절약

수식:



$$s(t) = \sum p(t - iT_s - d_i)$$
$$d_i \in \{0, \Delta, 2\Delta, \dots, (M-1)\Delta\} \text{ // } M\text{-ary PPM}$$

에너지 효율성:



$$E_PPM \approx E_pulse \times (duty_cycle)$$

duty_cycle << 1 (예: 0.01)

12.4 PSO-ANN 암호화

Particle Swarm Optimization과 Artificial Neural Network 기반 보안:

PSO 최적화:

1. 입자 군집(암호 키 후보)을 랜덤 초기화
2. 각 입자의 적합도 평가(암호 강도)
3. 개인 최적(pbest)과 전역 최적(gbest) 업데이트
4. 속도 및 위치 업데이트:



$$v_i = w \times v_i + c1 \times r1 \times (pbest - x_i) + c2 \times r2 \times (gbest - x_i)$$

$$x_i = x_i + v_i$$

5. 수렴까지 반복

ANN 암호화:

- 입력: 평문 데이터
- 은닉층: PSO로 최적화된 가중치
- 출력: 암호문
- 복호화: 역방향 신경망

장점:

- 양자 컴퓨터 저항성
- 적응적 보안(환경에 따라 키 변경)

13. 고급 제어 이론

13.1 LQR (Linear Quadratic Regulator)

상태 공간 모델에서 최적 제어 계산:

시스템 모델:



$$\dot{x} = Ax + Bu$$

비용 함수:



$$J = \int (x^T Q x + u^T R u) dt$$

최적 제어:



$$u^* = -Kx$$

$$K = R^{-1} B^T P$$

여기서 P는 Riccati 방정식의 해:



$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

시뮬레이션 적용:

- 상태: [position, velocity, orientation]
- 제어: [thrust_x, thrust_y, thrust_z]
- Q: 추적 오차 패널티
- R: 에너지 소비 패널티

13.2 칼만 필터 vs 파티클 필터

칼만 필터 (선형 가우시안 시스템):



예측: $\hat{x} = A\hat{x} + Bu$, $P = APA^T + Q$

업데이트: $K = PH^T(HPH^T + R)^{-1}$

$$\hat{x} = \hat{x} + K(z - H\hat{x}), P = (I-KH)P$$

파티클 필터 (비선형 비가우시안 시스템):

- 나노 규모의 복잡한 역학 처리
- 다중 가설 추적 가능
- 브라운 운동의 비가우시안 분포 대응

선택 기준:

- 시스템 선형성: 선형 → 칼만, 비선형 → 파티클
- 노이즈 분포: 가우시안 → 칼만, 비가우시안 → 파티클
- 계산 자원: 제한적 → 칼만, 충분 → 파티클

13.3 분산 제어 vs 중앙 제어

중앙 제어:

- 장점: 전역 최적해, 간단한 구현
- 단점: 통신 병목, 단일 실패점, 확장성 제약

분산 제어 (ADMM):

- 장점: 확장성, 내결함성, 통신 효율
- 단점: 수렴 속도, 로컬 최적해 가능

하이브리드 접근: 시뮬레이션은 계층적 하이브리드 제어 사용:

- 클러스터 내: 리더 기반 중앙 제어
- 클러스터 간: ADMM 기반 분산 제어
- 전역: 바이오사이버 인터페이스를 통한 상위 레벨 조정

14. 도전 과제 및 미래 연구 방향

14.1 현재 도전 과제

에너지 하베스팅:

- 포도당 연료 전지: ~1 pW/cell
- 압전 효과: 혈류에서 에너지 수확
- 무선 전력 전송: THz 빔으로 충전

나노 제조:

- 대량 생산 기술 부재
- 품질 관리의 어려움
- 비용 문제

생체 적합성:

- 면역 반응 최소화
- 독성 물질 배제
- 생분해성 설계

보안 및 프라이버시:

- 의료 데이터 보호
- 해킹 방지
- 오작동 방지 메커니즘

14.2 미래 연구 방향

양자 통신:

- 양자 얽힘 기반 보안 통신
- 초장거리 통신 가능성

DNA 컴퓨팅:

- DNA를 계산 매체로 사용
- 초병렬 처리
- 생체 내 자연 통합

뇌-기계 인터페이스:

- 나노로봇이 뉴런과 직접 통신
- 신경 질환 치료
- 인지 능력 향상

자가 복제:

- 체내에서 자가 복제 가능한 나노로봇
- 지속적인 치료 효과
- 윤리적 문제 고려 필요

15. 결론

본 시뮬레이션은 나노로봇 집합 제어의 복잡한 물리적, 계산적, 통신적 측면을 통합한 포괄적인 모델을 제공합니다. Navier-Stokes 유체역학, 회전 동역학, 확률적 상태 추정, 분산 최적화 등 고급 수학적 모델과 MPC, 파티클 필터, ADMM 등 최신 제어 알고리즘을 결합하여 현실적인 나노 규모 현상을 재현합니다.

클러스터 제어 시스템은 리더-워커 계층 구조를 통해 확장 가능하고 에너지 효율적인 집단 행동을 가능하게 하며, 분자, 전자기, 음향 기반 다중 모드 통신 프로토콜은 다양한 환경 조건에서의 신뢰성 있는 정보 전달을 보장합니다.

의료, 환경 모니터링, 군사, 산업 등 다양한 응용 분야에서 나노로봇 기술의 잠재력은 무궁무진하며, 본 시뮬레이션은 이러한 기술의 개발과 평가를 위한 강력한 도구로 활용될 수 있습니다. 순수 JavaScript로 구현되어 웹 브라우저에서 즉시 실행 가능하며, 파라미터 조정을 통해 다양한 시나리오를 탐색할 수 있습니다.

향후 양자 통신, DNA 컴퓨팅, 뇌-기계 인터페이스 등 혁신적인 기술과의 융합을 통해 나노로봇 시스템은 더욱 발전할 것이며, 인류의 건강과 삶의 질 향상에 크게 기여할 것으로 기대됩니다.