

6 軸アームの制御に Xinu メッシュは効くか

— 現状評価、効果の見積もり、実機構成の提案

2026 年 7 月 21 日

Abstract

6 関節のアーム型ロボットを Raspberry Pi 5 経由で制御する構成において、Xinu ベースの 12 台メッシュネットワークが効果を発揮するかを検討する。結論を先に述べる。**関節サーボループに対しては効果がなく、むしろ有害である。**一方で**動作計画・学習・分散知覚・冗長性には効く**が、効果は台数に比例しない。既存の実測値からの外挿では 12 ノードで 4-6 倍程度である。さらに、**現状の Xinu は 3 ボードとも I2C・PWM・GPIO ドライバを一切持たず、アームに物理的に接続されていない。**本書はその根拠を示し、メッシュの価値を最大化する実機構成を提案する。

1 前提と問い

対象は 6 関節のアーム型ロボットで、全ての関節指令は Pi 5 を通して送る構成を想定する。本書が答える問いは 3 つ。

1. この構成で Xinu メッシュ 12 台は効果を発揮するか。
2. 効果があるとして、どの程度か。
3. メッシュの効果を最大限に生かすなら、どのような実機構成が考えられるか。

評価対象の現物は Raspberry Pi 3 B+ / Pi 4 / Pi 5 の 3 台で、WiFi ad-hoc (IBSS) による MANET を構成している (SSID MANET、ch 6、静的 10.0.0.<n>、HELLO は UDP:5000 の 2 秒周期)。2026-07-21 時点で 3 ノード完全メッシュが成立している。

2 結論

用途	効果	理由
関節サーボループ	効果なし	制御ループが Pi 5 内で閉じるため、他ノードはクリティカルパス外。加えて WiFi の共有媒体が Pi 5 のジッタを増やす
動作計画 (RRT* 等)	効く	ほぼ完全並列。ただし線形には伸びない
強化学習のロールアウト	効く	エピソードが独立。既存 TinyML RL 資産と相性が良い
分散知覚	効く	各ノードで特徴抽出し、特徴だけを送る
冗長性・フェイルオーバー	効く	安全性の観点では最も価値が高い

3 なぜ制御ループには効かないのか

3.1 構成上の必然

制御ループが Pi 5 → サーボである限り、他の 11 台はクリティカルパスの外にしか存在できない。ネットワークを渡らない処理にノードを足しても、そのループは 1 マイクロ秒も速くならない。これは実装の問題ではなく構成の問題である。

3.2 数値で見た不一致

6 軸アームの関節ループは典型的に 100–1000 Hz、周期 1–10 ms で回す。ジッタ許容はその 1 割程度、すなわち 0.1–1 ms 程度である。

測定対象	実測値	出典
Pi 4 のローカル RT スケジューラのジッタ	max 10 μ s	/rtos-jitter 10 ms 周期・CPU hog 下、ミス 0/100
メッシュ経由の ICMP RTT (WiFi)	3–27 ms	Pi 5 $\rightarrow$ Pi 3 / Pi 4、2026-07-21
必要なジッタ許容 (1 kHz ループ)	0.1 ms	一般的な設計値

ローカルのスケジューラは**十分すぎる性能**を持っている。max jitter 10 μ s は 1 kHz ループの許容値の 1/10 であり、RTOS として本物である。問題はネットワークで、**WiFi 経由のレイテンシは要求より 1–2 桁大きい**。しかも CSMA/CA は競合ベースで再送を伴うため分布の裾が重く、平均値が満たせても最悪値が保証できない。関節ループを WiFi 越しに回す設計は成立しない。

4 効果の見積もり

台数効果は既存の実測値から外挿できる。分散 N-Queens のベンチマークでは、**3 ボード $\times$ 4 コア = 12 コア**で N=14 を 1458 ms、最速単体比 **1.87 倍**であった。3 倍のハードウェアで 1.87 倍、すなわち**効率 62%** である。しかもこの問題は通信をほとんど必要としない、並列化に最も有利な部類に属する。

12 ノードに広げた場合、WiFi の帯域と結果を集約するコーディネータがボトルネックになるため、**現実的な期待値は 4–6 倍**、通信を伴う問題ではそれ以下と見るべきである。

具体的な意味は次のとおり。単体で 500 ms かかる動作計画が **100 ms 前後**になる。再計画の応答性としては意味のある改善だが、サーボループとは別次元の話である。

5 現状の欠落

5.1 アームに繋がっていない

両ボードの device/ ディレクトリの内容は次のとおりである。

```
genet gic mbox sd timer uart usb video wifi
```

I2C も PWM も GPIO もサーボドライバも存在しない。すなわち現状このメッシュは**アームに物理的に一切接続されていない**。DOFBOT や mecharm の成果は Mac 上のシミュレータ (AIPL / Three.js) であり、実機の Xinu 側にアクチュエータ I/O は無い。

5.2 ROS 相当として欠けているもの

重要度順に挙げる。

1. **アクチュエータ・センサ I/O**。上記。これが無い限り何も動かない。最優先。
2. **時刻同期**。各ノードが自分の CNTPCT を見ているだけで、ノード間の共通時刻がない。加えて時間軸自体に既知の欠陥がある — rpi3 ではアクターランタイムが「100 Hz クロック」を仮定して clkticks * 10 しているが実際は 1000 Hz で **10 倍ずれ**、さらに clkticks が毎秒 0 にリセットされ**単調でない**。センサ融合も協調軌道も成立しない。
3. **Pub/Sub と QoS**。現在の通信は HTTP のリクエスト/レスポンスとアクターメッセージのみ。トピックによる多対多、信頼性/ベストエフォートの選択、デッドラインや liveliness が無い。

4. **有界遅延のトランスポート**。今の TCP は RTO 500 ms の go-back-N で、制御ループには使えない。RT 制御には UDP ベースで再送回数を制限した別経路が要る。
5. **座標系と運動学**。TF (フレーム木)、URDF 相当のロボット記述、FK/IK、軌道表現がボード側にはない。
6. **安全機構**。ウォッチドッグと非常停止がなく、ユーザ空間が存在しない (EL1 単一アドレス空間) ため、JIT で送り込んだコードの暴走がノード全体を落とす。アームを動かす前に必須。

5.3 あるもの

公平のため、既に本物である部分も記す。Pi 4 のプリエンプティブ優先度スケジューラ (実測 max jitter 10 μ s、100 回中ミス 0) は制御ループの土台として通用する。マルチホップ MANET、アクターラントタイム、ノードへコードを送り込む JIT も揃っている。土台はあるが、ロボットミドルウェアとしての層が丸ごと無い、というのが正確な評価である。

6 実機構成の提案

現在の「Pi 5 が 6 関節を直接叩く」構成は、**メッシュの価値を最も引き出せない形**である。メッシュを活かすなら、産業用アームが実際に採っている構造に寄せるのが筋である。

6.1 提案 1: スマートジョイント化 (最重要)

各関節にモータドライバと同居する小さな計算ノードを置き、**電流・位置ループを関節ローカルで 1–10 kHz で閉じる**。ネットワークを渡るのは目標値と軌道だけ (100 Hz–1 kHz) とする。こうして初めて「6 関節ノード + 計画ノード + 知覚ノード」という構成に意味が生まれる。

関節側は Pi である必要はない。ハードウェア PWM とエンコード入力を持つ RP2040 / Pico 2 や STM32 が適任である (RP2040 なら PIO でエンコードを読める)。これは EtherCAT スレーブを関節ごとに置く産業用アームと同じ構造である。

6.2 提案 2: 決定性のあるバックボーン

関節チェーンは CAN-FD または EtherCAT、上位はスイッチド Gigabit Ethernet、可能なら TSN (802.1Qbv) とする。WiFi ad-hoc は制御には向かない。

6.3 提案 3: 無線メッシュが正当化される構成

WiFi メッシュが本当に必要なのは、**物理的に離れたノードがある場合**だけである。

- ・移動マニピュレータ (台車 + アーム) — 台車と外部センサ間は無線が必然
- ・複数アームの協調 — 別筐体間の通信
- ・環境設置型センサ (部屋のカメラ、触覚マット) — 配線したくない場所

すなわちアーム内部は有線・決定性、アームと外界は無線メッシュという役割分担が、今の資産を最も活かす形である。

7 今の機材で効果を測る実験

新しいハードウェアを購入する前に、既存の 3 ボードで答えを出せる。

1. 6 関節は予定どおり Pi 5 に直結する (まず I/O ドライバが必要)。
2. 動作計画だけをメッシュに出す。Pi 5 が計画クエリを投げ、 N ノードが並列に探索し、最良の軌道を返す。

3. 単体ノード時とのエンドツーエンド計画レイテンシを比較する。

これは既存の JIT とアクターランタイムで新規ハードウェアなしに測定できる。N-Queens で 1.87 倍という実測値が既にあるので、同じ枠組みで計画問題の実効値が取れる。この数字が出て初めて、12 台に増やす価値があるかを根拠を持って判断できる。

8 本書の測定値について

数値の出所と限界を明示する。

- **max jitter 10 μ s / ミス 0/100** — Pi 4 実機、/rtos-jitter、10ms 周期・CPU hog 下での実測。
- **メッシュ RTT 3–27 ms** — 2026-07-21 に Pi 5 から Pi 3 / Pi 4 へ実施した ICMP の実測。サンプル数は少なく、最悪値の統計は取っていない。制御用途の判断には本来もっと長時間の分布測定が要る。
- **N-Queens 3 ボード 1.87 倍** — 既存ベンチマークの記録値。12 ノードでの 4–6 倍は実測ではなく外挿であり、WiFi の帯域とコーディネータの負荷という 2 つの仮定に依存する。
- **device/** の内容は 2026-07-21 時点のソースツリーを直接確認した結果である。

対象: *xinu-rpi3* (BCM2837 / Cortex-A53)、*xinu-rpi4* (BCM2711 / Cortex-A72)、*xinu-rpi5* (BCM2712 / Cortex-A76)。メッシュは *WiFi ad-hoc (IBSS)*、*SSID MANET*、*ch 6*、*10.0.0.2/3/5*。