

博士論文

多形態多自由度ロボットにおける
体内分散実時間制御システムの
設計実現法に関する研究

令和3年12月3日 提出
指導教員 稲葉雅幸 教授

東京大学大学院 情報理工学系研究科
知能機械情報学専攻
48177507 永松 祐弥

目次

第1章	序論	9
1.1	本研究の背景	11
1.2	本研究の目的	12
1.3	本論文の構成	14
第2章	多自由度ロボットの動作制御系に求められる実時間環境適応性能	17
2.1	はじめに	19
2.2	多自由度ロボットの環境適応行動実現	19
2.2.1	多自由度ロボット	19
2.2.2	環境適応行動	22
2.3	多自由度ロボットの体内制御システム	24
2.3.1	ロボット体内分散制御システム	24
2.3.2	分散制御ノード	28
2.3.3	中枢全身制御系	30
2.4	ロボット動作制御におけるフィードバック遅延の課題と実時間性の要求	32
2.4.1	フィードバックに基づくロボット動作制御	32
2.4.2	ロボット動作制御におけるむだ時間の影響	33
2.4.3	フィードバック遅延の解決によるロボット行動実現	36
2.4.4	実時間性を満たすロボット体内制御システムの要件	38
2.5	分散ネットワークにおける低遅延データ転送技術	40
2.5.1	産業用フィールドバス	40
2.5.2	高性能計算機のペリフェラル通信インタフェース	40
2.5.3	ロボット体内分散ネットワーク	42
2.6	組込計算機における低遅延演算技術	43
2.7	ロボット体内における多階層分散制御技術	45
2.7.1	ヘテロジニアス分散制御モデル	45
2.7.2	ヒトの神経系に見られる多階層環境適応フィードバック機構	46
2.7.3	ロボット実時間制御の分散実行プラットフォーム	49
2.8	本研究で提案する体内分散実時間制御システムの設計実現法	50
2.8.1	本研究の提案手法	50
2.8.2	子細適応法	51
2.8.3	統合節減法	53
2.8.4	実地進展法	55
2.9	多形態多自由度ロボットのための体内分散実時間制御システムの構成	57
2.10	本章のまとめ	61

第 3 章	体内分散実時間ネットワークを形成する小型高信頼な感覚駆動通信系	63
3.1	はじめに	65
3.2	多自由度ロボットの体内分散ネットワークとデータ転送性能	66
3.2.1	体内分散ネットワークトポロジ	66
3.2.2	体内分散ネットワークにおけるデータ転送遅延	69
3.2.3	体内分散ネットワークに求められるデータ転送性能	70
3.3	低遅延化に向けた従来の組込分散制御系の課題	74
3.3.1	ギア駆動型大出力ヒューマノイドの従来実装	74
3.3.2	腱駆動型筋骨格ヒューマノイドの従来実装	79
3.3.3	多リンク飛行ロボットの従来実装	82
3.4	多自由度ロボットに適した実時間体内分散ネットワークの設計	84
3.4.1	ロボットの実時間制御に適う組込分散ネットワーク	84
3.4.2	高速光通信分散ネットワークの設計	87
3.5	分散ネットワークを構成する小型高速通信モータ制御モジュールの設計と実装	91
3.5.1	高速通信ネットワークの小型高信頼化設計	91
3.5.2	腱駆動向け小型モータ制御モジュール	92
3.5.3	小型光通信モータ制御モジュール	94
3.5.4	光通信分散ネットワークの性能評価	96
3.6	中枢全身制御系の基幹通信インタフェース	97
3.6.1	PCI バスを利用した従来実装の課題	97
3.6.2	中央計算機の通信インタフェース	101
3.6.3	低遅延基幹通信インタフェースの小型高信頼化設計	102
3.7	Ethernet を利用した低遅延高信頼基幹通信ネットワークの設計と実装	103
3.7.1	EtherCAT によるプロトタイプ実装	103
3.7.2	高速大容量な Gigabit Ethernet 通信インタフェースの設計	105
3.7.3	Ethernet 中継モジュールの小型化実装	108
3.7.4	Gigabit Ethernet 通信インタフェースの性能評価	111
3.8	USB 通信ネットワークの低遅延化実装	113
3.8.1	USB High-Speed 中間制御基板	113
3.8.2	USB 低遅延化実装の性能評価	115
3.9	本章のまとめ	117
第 4 章	階層アーキテクチャ設計に基づく低遅延全身応答のための組込中間自律系	119
4.1	はじめに	121

4.2	ヘテロジニアス中枢全身制御系による低遅延フィードバック機構	121
4.2.1	組込計算機を交えたヘテロジニアス中枢全身制御系の構成	121
4.2.2	体内分散制御システムにおけるフィードバック経路	122
4.3	組込中間層計算機的设计と実装	126
4.3.1	FPGA SoC を利用した組込中間層計算機プロトタイプ実装	126
4.3.2	組込中間層計算機モジュールの概要	127
4.3.3	組込中間層計算機モジュールの小型高信頼化設計	129
4.3.4	組込中間層計算機における組込制御システム実装	140
4.4	組込中間層計算機における高速演算実装	147
4.5	中枢全身制御系におけるパケット切替中継機構	151
4.6	本章のまとめ	154
第 5 章	多形態多自由度動作制御における組込高速演算を交えた全身階層協調系	157
5.1	はじめに	159
5.2	多自由度ロボットの動作制御計算構造	159
5.3	多階層から構成されるロボット体内分散制御系	162
5.4	分散制御ノードのローカル低遅延フィードバック制御機構	164
5.4.1	モータ制御モジュールの基礎サーボシステム構成	164
5.4.2	ギア駆動型ロボットの関節トルク推定	170
5.4.3	ローカルフィードバックによる関節トルク制御	174
5.5	低遅延全身動作制御へ拡張可能なハードウェア制御系の一般化実装	178
5.5.1	ハードウェア制御系の共通構造	178
5.5.2	ハードウェア制御系の従来実装における非同期構造の課題	182
5.5.3	一般化デバイス制御プラットフォーム	185
5.6	組込中間層計算機を活用した低遅延全身動作制御	190
5.6.1	ヘテロジニアス全身動作制御系アーキテクチャ	190
5.6.2	アドミタンス制御演算の実装	193
5.7	本章のまとめ	202
第 6 章	体内分散実時間制御システムの応用戦略と行動実装評価の実現	205
6.1	はじめに	207
6.2	体内分散実時間制御システムの応用	207
6.3	ギア駆動型大出力ヒューマノイドの柔軟環境接触と衝撃緩和への応用	208
6.3.1	ギア駆動型大出力ヒューマノイドへの応用戦略	208
6.3.2	体内分散実時間制御システムの要素技術適用事例	209

6.3.3	腕部マニピュレータにおける安定接触打撃作業	210
6.3.4	脚部における跳躍着地動作	225
6.4	腱駆動型筋骨格ヒューマノイドの干渉駆動精密動作への応用	237
6.4.1	腱駆動型筋骨格ヒューマノイドへの応用戦略	237
6.4.2	低遅延 USB 通信ネットワークの適用	239
6.4.3	高信頼な小型高速通信モータ制御モジュールの適用	244
6.4.4	ハイブリッド構成ロボットへの拡張展開	252
6.5	多リンク飛行ロボットの力制御型空中マニピュレーションへの応用	256
6.5.1	多リンク飛行ロボットへの応用戦略	256
6.5.2	飛行ロボット関節モジュールにおける把持力制御	260
6.6	連続跳躍ロボットの高追従姿勢安定化への応用	263
6.6.1	連続跳躍ロボットへの応用戦略	263
6.6.2	ワイヤ張力制御と全身姿勢フィードバックに基づく連続跳躍	265
6.7	本章のまとめ	268
第 7 章	結論	273
7.1	本論文の総括	275
7.2	本研究の成果と結論	277
7.3	将来への展望	280
	謝辞	285
	発表文献	289
	参考文献	295

第 1 章

序論

1.1 本研究の背景

世界初の産業用ロボット Unimate が登場した 1961 年より半世紀を超え，ロボット工学の発展ならびに材料・半導体・ソフトウェアをはじめとした数多の技術革新に支えられて，ロボット技術は目まぐるしい進化を遂げてきた [1][2]．我が国でも国家プロジェクトの先導により先端ロボット技術の様々な実践応用が試みられてきた [3]．

電気モータに代表されるアクチュエータの小型・高効率・大出力化ならびに合金や繊維強化プラスチックなどの軽量・高強度素材の開発に伴い，ロボットは，自身の関節リンク機構を高速に駆動し，大きな操作力を発揮しうるだけの精強な機械性能を得ている．その機械性能を活用して，製造業や第一次産業，あるいはアミューズメント施設など定型な精密作業を実行する労働力が求められる現場では既に大きな役割を担ってきている．

近年では計算機やセンシングの性能が向上し，現実環境における複雑な問題の解決がより容易になりつつある．これに伴いロボットには，人間の生活する環境下で人間を補助し，屋外で設備の保守作業を行い，あるいは災害に対応するなど，応用的で複雑な行動実現が期待されている．現在の我が国でも，スマート社会の実現目標や多発する災害への対応要求を受けて，新エネルギー・産業技術総合開発機構の主導による次世代人工知能・ロボットの中核となるインテグレート技術開発事業 [4] やロボット・ドローンが活躍する省エネルギー社会の実現プロジェクト [5] をはじめ，ロボット技術を現実環境で活用することを目指した多くの研究開発が進められている．

複雑な現実環境へ適応すべく多種多様な形態のロボットが開発されている．応用的な行動の実現可能性を拡充する中で求められる関節数やアクチュエータ数は増大し，ロボットの多自由度化も進んでいる．

中でも人間の作業を代行するものとして期待されている等身大ヒューマノイド [6] は，拘束なく自由に運動する身体に人間を模した脊椎や四肢を備えており，複雑な多自由度系を有するロボットの代表例であると言える．多彩な動作を実行可能とする多くのヒューマノイド設計では四肢を備えた骨格構造におよそ 30 から 40 前後の関節自由度が採用される [7]．これにハンドや視覚センサなどを加えると更に自由度は増大する．ヒューマノイド ARMAR-4 [8] の例では 63 に及ぶ能動関節が搭載されている．

環境によく馴染む柔軟な身体構造や駆動原理を有する筋骨格ロボット [9] の研究も進んでいる．多くの筋骨格ロボットは多数の関節自由度や弾性機構を多数のアクチュエータで複雑に干涉駆動 [10] させる仕組みを採用している．等身大の身体に 100 を超えるアクチュエータを搭載した筋骨格ヒューマノイド [11] も開発されている．筋骨格ロボットの制御にはその自由度に応じて大規模なデータの処理が伴う．駆動原理に基づく単純なモデル化が困難であり学習に基づく写像計算 [12] などの数理的手法による制御が必要となる場合も多い．

更に近年ではフィールドロボットの分野でも多自由度化が進んでいる．適応的な多形態移動を実現する多脚ロボット [13] や関節リンク機構を有し空中で姿勢変形する飛行ロボット [14] などが提案されている．

特に即応的な環境適応行動や動的な高速・大出力動作の実現を目指す上では，ロボットの全身フィードバック動作制御系の応答性が重要な鍵となる．フィードバック制御系の時定数が大きく，あるいは大きな遅延が存在すると，応答性が低下し行動の失敗に繋がる．マニピュレータの適切な操作力制御における応答性低下は過大な力を引き起こしマニピュレータ自身や操作対象の損壊を引き起こしうる．脚ロボットの地面反力や姿勢変化に対する応答性低下はバランス崩壊や転倒を引き起こしうる．環境に対して高速に作用を及ぼす工具 [15] や車両 [16] などの道具操作における遅延は道具の暴走および重大な事故に繋がりがねない．

多形態化および多自由度化が進む昨今のロボットではその制御系もハードウェア・ソフトウェアの両面において複雑なものとなっている．制御系が肥大化しセンサ入力からフィードバック制御系を経てアクチュエータの出力に至るまでの経路が延伸することは，大きな遅延が生じる原因となる．これらのロボットに期待されているのは実際に人間の生活する環境や災害現場などでの活躍であり，行動の失敗やそれに起因する内外の損壊は致命的な問題となりうる．複雑な制御系を有する多形態多自由度ロボットにおける全身フィードバック動作制御系の応答性向上は，環境適応行動の実現に向けた重要な課題であると言える．

1.2 本研究の目的

本研究の目的は，多形態多自由度ロボットの動作制御におけるフィードバック遅延が行動実現目標に与える影響を明らかにし，これを低減する合理的なロボット体内制御システムの設計実現法を樹立することである．多形態多自由度ロボットの環境適応行動を実現してゆくためには，単に低遅延な制御システムを設計して実装するのみに留まらず，ロボットへの展開と実証評価を繰り返す中で制御システムを更新しロボット技術を継続的に発展させてゆく必要がある．先端技術に追従したロボット体内制御システムの設計，実装，および実証評価を通してロボット技術の加速的な発展を促すことで次世代ロボットや環境適応行動を実現してゆくための手法を設計実現法として提案する．

ロボットのセンサ入力からアクチュエータ出力までの電子制御系における遅延には，ハードウェアによるものとソフトウェアによるものがある．

ハードウェア遅延とはすなわち通信ネットワーク遅延である．通信の高速化，もしくはネットワーク経路の縮約によって低減することができる．多自由度ロボット体内での運用に適した高速通信プロトコルに基づいてデータを転送するとともに，よりハードウェアと

密接に結びつく組込計算機をフィードバック制御に利用することで，ハードウェア遅延の課題が解決される．

ソフトウェア遅延は更に，待機時間と計算時間に分類して考察できる．高い汎用性や拡張性を目指して非同期マルチプロセス化する多自由度ロボットの制御ソフトウェアシステムでは，特にプロセス間通信に起因する待機時間の影響が大きい．低遅延化が求められる全身フィードバック動作制御の実装にあたっては非同期なソフトウェア構造を見直し，データ処理経路の縮約を考える必要がある．計算時間の中に合う限り実時間制御周期を短く設定することも待機時間の削減に効果を発揮する．演算能力が限られている組込計算機を用いたフィードバック制御では計算時間の短縮も大きな課題となる．

低遅延なフィードバック性能を主軸とする合理的なロボット体内制御システムには，実時間性および展開実用性を兼ね備えることが求められる．実時間性は制限時間内の確実な応答を保証する性質であり，低遅延，低ジッタ，および信頼性の特性要件に基づいて成り立つ．展開実用性は多形態ロボットへの合理的な展開を支持する性質であり，小型軽量，低消費電力，汎用性，および運用性の特性要件に基づいて成り立つ．

本研究では実時間性および展開実用性の要件を満たすロボット体内制御システムの検討および開発を通して，次の 3 手法の体系化を目指す．

- 低遅延制御を達成する実時間システムの実装評価法
- 多形態多自由度ロボットへの合理的展開法
- ロボット体内制御システム継続発展に向けた設計実現法

通信の高速化，フィードバック経路の縮約，制御周期の短縮，および演算の高速化に基づく低遅延化を主軸として，実時間性および展開実用性のそれぞれの要件に寄与する要素技術を開発する．実時間性および展開実用性を満たすには，ロボット技術と実装技術のそれぞれの特性をともに深く理解した俯瞰的な要件定義が重要である．現在の最先端ながらも広く得られる技術を活用した要素技術実装例を与え，その組み合わせにより多様な形態の多自由度ロボットにおけるフィードバック遅延の低減を図る．ロボットに展開し運用する中でシステム性能を動的に評価し，実装仕様の更新や次の設計へと繋げてゆく．先端技術に追従した制御システムの実地設計実現が，ロボット技術の加速的な発展には欠かせない．要素技術実装を通して，多くのロボット研究開発の基盤となるシステムを提供するとともに，将来に渡る技術革新を受け統合的なロボット技術を発展させてゆくための普遍的手法を抽出し明確なものとする．

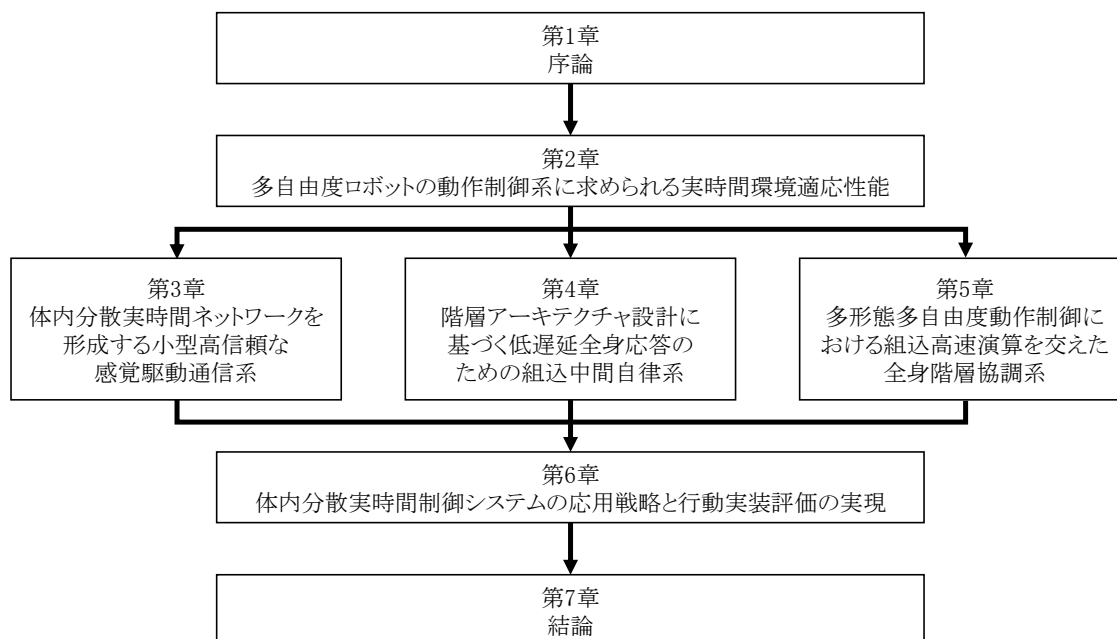


図 1.1 Thesis chapter outline

1.3 本論文の構成

本論文では多形態多自由度ロボットにおける体内分散実時間制御システムの設計実現法に関して論じる．本論文は全 7 章から構成される．本論文の構成を図 1.1 に示す．

- 第 1 章「序論」では，本研究の背景と目的を述べた．
- 第 2 章「多自由度ロボットの動作制御系に求められる実時間環境適応性能」では，多形態多自由度ロボットの環境適応行動実現に向けたフィードバック遅延の課題を考察し，これを解決するための技術検討を行った上で，本研究の体内分散実時間制御システムおよびその設計実現法を提案する．
- 第 3 章「体内分散実時間ネットワークを形成する小型高信頼な感覚駆動通信系」では，ロボット制御に必要なデータ転送性能を達成する組込分散ネットワークおよび基幹通信ネットワークの設計と，その小型・高信頼なモジュール実装に関して述べる．
- 第 4 章「階層アーキテクチャ設計に基づく低遅延全身応答のための組込中間自律系」では，低遅延フィードバック動作制御系実装の基盤となる，組込中間層計算機の設計とハードウェアアクセラレーションを交えた組込高速演算の実装手法に関し

て述べる．

- 第 5 章 「多形態多自由度動作制御における組込高速演算を交えた全身階層協調系」では，組込計算機を交えて多階層から構成されるロボット体内分散制御系において，低遅延フィードバックを達成する動作制御系の実装に関して述べる．
- 第 6 章 「体内分散実時間制御システムの応用戦略と行動実装評価の実現」では，提案する体内分散実時間制御システムを多形態多自由度ロボットに展開する戦略および手法を述べ，実証的な評価を行う．
- 第 7 章 「結論」では，本論文を総括し，本研究の成果と結論をまとめ，将来への展望を述べる．

第 2 章

多自由度ロボットの動作制御系に
求められる実時間環境適応性能

2.1 はじめに

本研究では、多形態多自由度ロボットの環境適応行動実現に適う体内分散実時間制御システムの設計実現法を樹立する。近年の多自由度ロボットでは、高度な制御計算の負荷を分散し、また省スペース性や電磁ノイズ・機械的振動のもとでの信頼性を向上するため、体内に分散制御システムを構成する手法が主流となっている。体内分散制御システムにおけるフィードバック動作制御では、分散ネットワークのデータ転送遅延およびソフトウェア制御計算遅延が応答性に影響を与える。ロボットの行動実現に向けた研究開発の中で、高信頼なフィードバック遅延低減やそれを満たすシステムの実用化に向けた課題が明らかとなってきている。進化を続ける半導体技術や電子制御技術に追従しロボット研究開発の要求に応じてゆく、体内制御システムの柔軟な実装と継続的な開発が求められる。

本章では、多自由度ロボットの環境適応行動実現に向けたフィードバック遅延の課題を分析し、実時間性および展開実用性を伴ってこれを合理的に解決する体内分散実時間制御システムの設計実現法を提案する。2.2 節で、本研究の対象とする多自由度ロボットおよび目指す環境適応行動に関して述べる。2.3 節で、多自由度ロボットの体内に構成される分散制御システムに関して述べる。2.4 節で、フィードバック遅延が多自由度ロボットの行動実現に与える影響を考察し、体内制御システムに求められる実時間性および展開実用性の要件に関して述べる。2.5 節で、第 3 章に述べる体内分散実時間ネットワークに関する技術検討を行う。2.6 節で、第 4 章に述べる組込中間層計算機に関する技術検討を行う。2.7 節で、第 5 章に述べるロボット全身動作制御に関する技術検討を行う。2.8 節で、本研究の目的とする体内分散実時間制御システムの設計実現法を提案する。2.9 節で、体内分散実時間制御システムの構成と要素技術実装に関して述べる。

2.2 多自由度ロボットの環境適応行動実現

2.2.1 多自由度ロボット

1.1 節でも述べたように、複雑な現実環境への適応に向けてロボットの多自由度化が進んでいる。本項では改めて多自由度ロボットの特徴やその自由度の規模に関してまとめる。

ロボットの自由度はそのリンク機構および駆動システムの 2 つの観点から記述できる。

リンク機構における自由度はロボットの幾何モデルに従って決定される。剛体リンク機構でモデル化されるロボットはそのモデル化された関節に自由度を有する。代表的な関節構造とその自由度の例として、

- 軸周り回転：1 自由度

- 直動機構：1 自由度
- 球関節：3 自由度
- 拘束無し of 自由運動：6 自由度

というように換算できる。ソフトロボットの研究も盛んに行われており、リンクを剛体でモデル化し難いロボットも多く登場している。その場合は単一リンク内の作用点同士の幾何拘束関係をモデル化することにより自由度として換算できる。

一方、駆動システムにおける自由度はハードウェア構成に従って決定される制御系の出力の次数、すなわちアクチュエータ数を指す。リンク機構自由度に対してアクチュエータが 1 対 1 の対応で搭載されるシステムは取り扱いが容易であり、産業用マニピュレータを中心に広く採用されている。リンク機構自由度とアクチュエータが 1 対 1 でないシステムを取り扱う問題は、

- 劣駆動：アクチュエータ数よりそれに対応する機構自由度が多いシステム
- 冗長駆動：アクチュエータ数よりそれに対応する機構自由度が少ないシステム
- 非ホロノミック系 [17][18]：アクチュエータ数より多くの機構自由度を制御可能なシステム
- 干渉駆動 [10]：複数のアクチュエータがひとつの機構自由度に寄与するシステム

としてよく知られている。

本研究における多自由度ロボットとは、自由度数の厳密な閾値で定義するものではなく、ある指標において比較的自由度数が多いと言えるロボットを指す。本研究ではこの指標として、その制御系に求められる実時間性に着目する。ロボットに搭載されるそれぞれのアクチュエータによる作用は独立しておらず、ロボット全体で複雑な行動を実現するために全身を管理する中央コントローラが必要となる。一方でアクチュエータ制御に低遅延な実時間性が求められる場合、多数のアクチュエータをひとつのコントローラのみで扱うことが困難となり、分散制御系の構成も求められる [19]。このような特徴を有するものを本研究の対象となる多自由度ロボットとしている。多自由度ロボットの分散制御系に関しては 2.3 節に、実時間性に関しては 2.4 節に、それぞれ詳細を述べる。

多自由度化が進む代表的なロボットの例として、1.1 節に述べたようにヒューマノイド・筋骨格ロボット・フィールドロボットなどが挙げられる。

我々が開発しているヒューマノイドの例を図 2.1 に示す。図 2.1 (a) はギア駆動型大出力ヒューマノイド JAXON [20] である。JAXON はエンドエフェクタや視覚センサを除く骨格構造に 33 の関節自由度を有し、これを 39 のモータで駆動している。一部の高負荷な関節では 2 個のモータを並列に利用している。図 2.1 (b) は筋骨格ロボットの特徴を併せ持つ腱駆動型筋骨格ヒューマノイド腱悟郎 [21][11] である。腱悟郎は当初の仕様で、骨

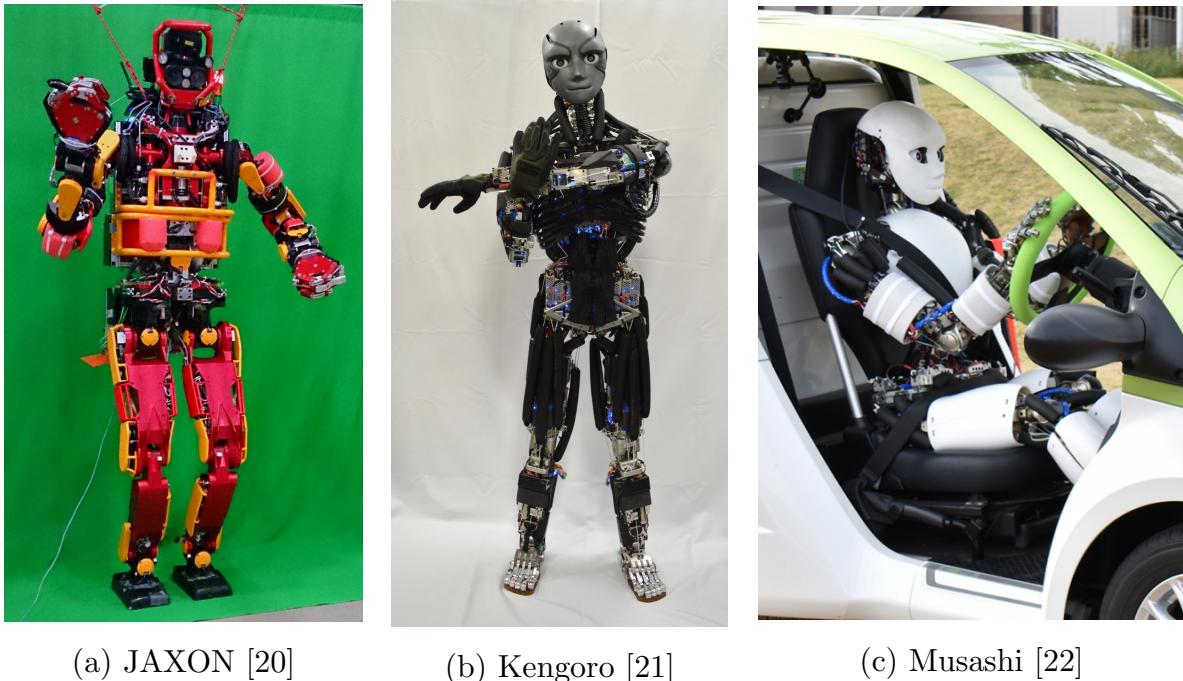


図 2.1 Example of multi-DOF humanoid robots

格構造に 174 の関節自由度を有し，これを 116 のワイヤ付きモータで駆動していた．最新仕様の臙悟郎では更に自由度が追加されモータ数が 124 に増えている．図 2.1 (c) は臙悟郎の後継機にあたる臙駆動型筋骨格ヒューマノイド Musashi [22][23] である．Musashi は骨格構造に 36 の関節自由度を有し，これに柔軟なハンドを加えた機構を 74 のワイヤ付きモータで駆動している．

多自由度化が進むフィールドロボットの例として図 2.2 に多リンク飛行ロボットを示す．図 2.2 (a) の DRAGON [14] は 4 リンクから構成された飛行ロボットであり，空中での 3 次元変形を可能としている．それぞれのリンクに 2 個のプロペラを搭載し，2 個 1 組のプロペラの姿勢が 2 自由度可変となっている．リンク間は 2 自由度の関節で接続されている．DRAGON は全体で 22 のアクチュエータ自由度を有している．図 2.2 (b) の HALO [24] は水平方向の閉ループで構成された 8 リンクの飛行ロボットである．図 2.2 (c) の HYDRUS [25] は水平方向関節リンク機構で構成される飛行ロボットである．それぞれのリンクに搭載された 1 個のプロペラは 1 自由度で傾斜方向が可変となっている．図 2.2 (c) に示した，ベースリンクに対して 6 リンクが接続された計 18 自由度の構成以外にも，モジュール化されたリンクの接続によって構造変更が可能となっている．

いくつかの代表的な多自由度ロボットの自由度数を紹介する．表 2.1 にヒューマノイドの自由度数を示す．2010 年頃までのヒューマノイドの自由度数は Akhtaruzzaman ら [26][7] による調査でも紹介されている．表 2.2 に筋骨格ロボットの自由度数を示す．



(a) DRAGON [14]

(b) HALO [24]

(c) HYDRUS [25]

図 2.2 Example of multilink aerial robots

表 2.3 にフィールドロボットの自由度数を示す。

2.2.2 環境適応行動

ロボットは接触による環境との複雑な相互作用を繰り返しながら目的の行動を実施する。ロボットの環境接触はその能動性から、

- 能動的あるいは計画的な環境接触
- 受動的あるいは外乱的な環境接触

に分類できる。また、接触力の発生形態から、

- 時系列での大きさが計画的に予見可能な接触力
- 滑らかに連続変動する接触力
- 発生およびその規模は判明している既知の衝撃的接触力
- 規模や発生そのものが予見できない未知の衝撃的接触力

に分類できる。

これらに基づいて分類すると、ロボットの行動においては表 2.4 のような環境接触力が発生する。環境接触力に適切に対応することで目的の行動が実施可能となる。しかし安定した接触力の維持や衝撃的接触力への対応には多くの課題が残されている。

打撃作業や着地動作では、衝撃的な環境接触を適切に取り扱う必要がある。このとき、

- ロボット自身や操作対象の損壊に繋がる過大な衝撃力を生じないこと
- 行動の失敗や転倒に繋がる過大な関節軌道誤差を生じないこと
- 行動を遂行するために求められる適切な大きさの操作力を発揮すること

の全てを満たすことが求められる。ただしなやかに衝撃を受け流すだけでなく、精密な

表 2.1 DOF in humanoid robots

robot	developer	year	joint DOF	actuator DOF	including endeffector
ASIMO [27]	Honda	2000	26 (34)		
ETL-Humanoid [28]	ETL	2001	46		*
KHR-2 [29]	KAIST	2005	41		*
CB ² [30]	Osaka Univ.	2007	58	56	*
LOLA [31]	TUM	2009	25		
ARMAR-4 [8]	KIT	2013	63		*
TORO [32]	DLR	2014	27		
JAXON [20]	Univ. of Tokyo	2015	33	39	
DRC-HUBO+ [33]	KAIST	2015	32		*
Valkyrie [34]	NASA	2015	44		*
ESCHER [35]	Virginia Tech	2015	38		*
Hydra [36]	Univ. of Tokyo	2016	41	47	*
BHR-6 [37]	BIT	2016	23		*
WATHLETE-1 [38]	Waseda Univ.	2016	22		
WALK-MAN [39]	IIT	2017	69	33	*
HRP-5P [15]	AIST	2018	37		*
ARMAR-6 [40]	KIT	2018	51	28	*

動作のもとで所望の操作力を発揮することが重要であり，困難で挑戦的な課題となっている．

可変剛性機構 [45] や筋骨格構造 [23] を採用した柔軟な身体によって，ハンマを用いた打撃作業を達成している例がある．柔軟性により過大な衝撃力の発生を避けられる一方で，動作や操作力の精密性には難点を残している．ハンマによる物体加工やスポーツにおける打球など高度な打撃作業を伴う行動実現に向けて，精密性と柔軟性を兼ね備えることが求められる．

ヒューマノイドや脚ロボットによる跳躍・走行の研究も多く進められている [46][47][48][49][50][51]．跳躍・走行においては，着地時の衝撃緩和，転倒を防ぐ姿勢制御，および適切な鉛直運動を生じる地面反力が求められる．精密にモデル化できない着地動作に対応することができれば，不整地における高速移動などへの応用可能性も広がる．

表 2.2 DOF in musculoskeletal robots and soft robots

robot	developer	year	joint DOF	actuator DOF	including endeffector
ECCE1 [41]	ECCEROBOT	2010		43	*
Athlete Robot [42]	Univ. of Tokyo	2010	14	20	
Kengoro [11]	Univ. of Tokyo	2016	174	116 (124)	*
Musashi [23]	Univ. of Tokyo	2019	36	74	*

表 2.3 DOF in field robots

robot	developer	year	joint DOF	actuator DOF	including endeffector
RoboSimian [43]	NASA	2015	28		
WAREC-1 [13]	Waseda Univ.	2017	28		
DRAGON [14]	Univ. of Tokyo	2018	22		
HALO [24]	Univ. of Tokyo	2018	16		
HYDRUS [25]	Univ. of Tokyo	2019	18		
HSL-A1 [44]	Ritsumeikan Univ.	2019	16	12	

2.3 多自由度ロボットの体内制御システム

2.3.1 ロボット体内分散制御システム

多自由度ロボットは全身に多数のアクチュエータおよびセンサを搭載する。センサ入力を反映して、複雑に相互干渉するアクチュエータ出力を適切に制御することで全身行動が実施される。このフィードバックを担う制御系の実装構成は重要な検討項目である。

ヒューマノイドに採用されてきた体内制御システムの例をもとに分析する。代表的なヒューマノイドの体内制御システム変遷を図 2.3 に示す。

1996 年に初の自立型等身大ヒューマノイド P2 [52][53] が開発された。小型ヒューマノイドや脚ロボットから P2 の先代にあたる P1 に至るまで、P2 以前のヒューマノイド研究開発においては体外に大型の計算機を据え置くりモートブレイン方式 [54] が主流であった。高性能計算機の小型化に伴い、ロボット体内への計算機の設置が現実のものとなった。P2 では 4 台の CPU 基板と I/O 基板が VME バスで接続された中央計算機が胴体

表 2.4 Contact force pattern in environment-adaptive robot activity

	能動的/計画的	受動的/外乱的
予見可能	定型マニピュレーション (計画可能な物体操作力) 歩行 (支持脚地面反力)	
連続変動	押し付け作業 (所望の物体操作力発揮) 地形馴染み動作 (所望の環境反力発揮) 柔軟マニピュレーション (モデル化困難な物体操作)	外乱馴染み動作 (インピーダンス制御) 姿勢安定化 (バランス制御)
既知衝撃	既知の対象への打撃作業 (大きさの予見可能な打撃反力) 跳躍・走行 (着地衝撃緩和・地面反力)	
未知衝撃	空振りの生じる打撃作業・スポーツ (過小となりうる打撃反力) 未知の対象への打撃作業 (剛性の予測できない打撃反力)	衝撃的外乱対応 (衝撃緩和・バランス制御) 不整地着地 (未知の衝撃的地面反力)

に設置され、全身のアクチュエータおよびセンサを中央計算機が集中的に管理している。その後 2000 年代頃までに、WABIAN [55][56] および WABIAN-2 [57]、H6 および H7 [58][59]、ASIMO [27]、KHR-1 [60][61]、BHR-01 [62][63]、HRP-2 [64] など、同様のバス接続による集中制御型の構成を取るヒューマノイドがいくつか開発されている。

集中制御型の構成では、中央計算機が全身のアクチュエータ・センサに対して密接にアクセス可能であり、また制御系が集約されているため管理が容易である、といった利点が得られる。しかし、高度化・多自由度化するロボットの制御に向けて実用上の課題が多く浮上してきた。集中制御型には、

- 中央計算機に負荷が集中し制御計算遅延の低減が困難となる
- 中央計算機から全身のアクチュエータ・センサに直結する配線が膨大で煩雑となり断線やノイズによるトラブルも生じやすい
- 大型の中央計算機が機械的な振動に弱い構造となりやすい

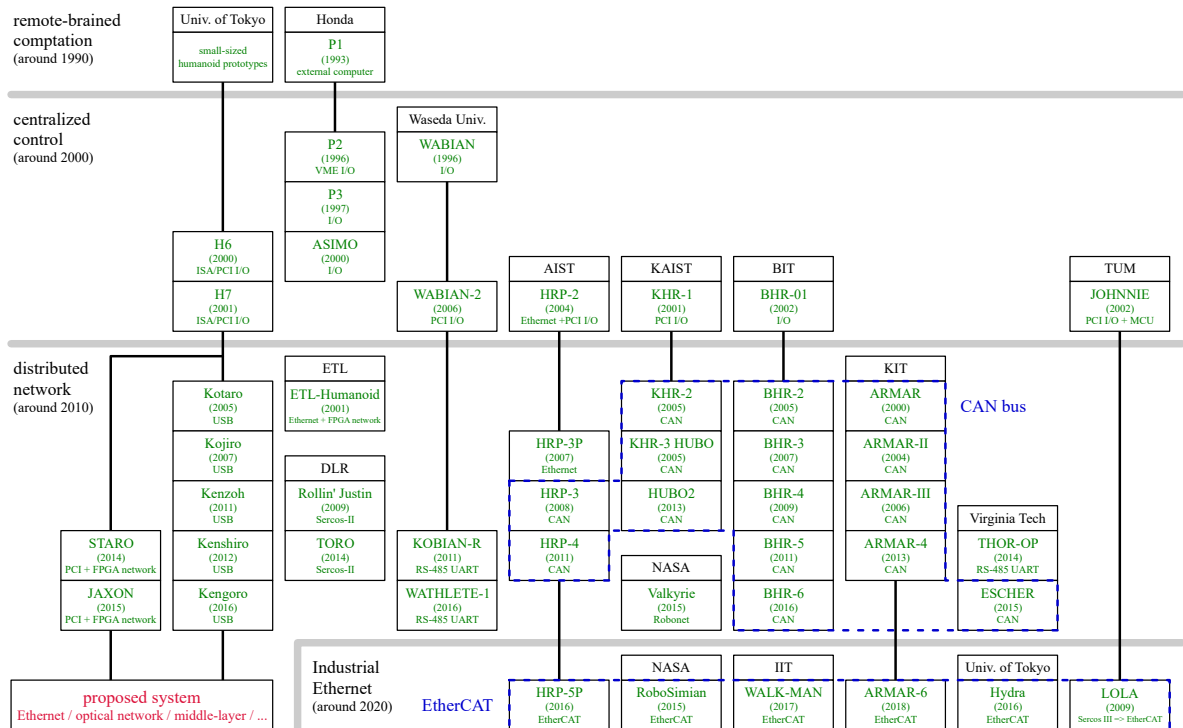


図 2.3 History of humanoid robot controller system

- 高頻度な実時間周期制御計算を中央計算機に実装すると電力効率の向上が困難となる

といった問題が存在する。

集中制御型の課題を受けて、2000年代前半頃からヒューマノイドの分散制御化に向けた議論と実装が進められてきた [65]。

JOHNNIE [66][67] では、省スペース化のためロボット体内に CAN で接続された 16 bit MCU のみによる組込分散制御系を搭載する構成が検討された結果、計算能力の不足が判明したため、中央計算機と 16 bit MCU を PCI バスで接続した段階的な組込制御系実装の試みがなされている。ETL-Humanoid [68][28] や R. Daneel Study 1 [69] では、単純なアクチュエータ出力とセンサ値取得を行うノードを分散配置し、FPGA を利用して体内通信ネットワークを実装している。部位ごとに分散ネットワークが構成され、これと接続し制御を担う中央計算機同士は Ethernet で接続されている。双腕上半身ロボット ARMAR [70] では中央計算機と複数の 16 bit MCU が CAN ネットワークで接続される分散制御系が実装されている。CAN による分散ネットワークは ARMAR の後継機にあたる ARMAR II [71], ARMAR-III [72] で開発が進められ、ヒューマノイド ARMAR-4 [8] まで引き継がれている。その後開発された ARMAR-6 [40] では EtherCAT ネット

ワークが採用されているが, EtherCAT を利用した試みについては後述する. KHR-2 [29] とその後継機 [73][61][33], および BHR-2[74] とその後継機 [75][76][77][37][63] でも CAN による分散制御系が採用されている. HRP-3P [78][19] では 10 Mbps の Ethernet バスによる分散ネットワークが試されているが信頼性や運用性に欠ける点が判明し, その後 HRP-3 [79] や HRP-4 [80] では CAN が採用されている. Rollin' Justin [81] や TORO [32] ではマニピュレータの関節モジュール開発 [82][83][84] を通して培われた Sercos II フィールドバス実装により, 物理層に光通信を採用した分散ネットワークを構成している. KOBIAN-R [85] では従来の集中制御型の構成に加えて, 一部の関節が独自のモータ制御モジュール [86][87] で制御され, RS-485 シリアル通信を物理層としたネットワークが構成されている. その後開発された WATHLETE-1 [38] では RS-485 シリアル通信による分散制御系が全身の関節に採用されている. THOR-OP [88] では ROBOTIS 社の DYNAMIXEL PRO サーボモータモジュールが採用され, RS-485 シリアル通信を物理層としたネットワークが構成されている. その後開発された ESCHER [89][35] では独自アクチュエータモジュールによる CAN ネットワークが採用されている. Valkyrie [34] では LVDS シリアル通信を物理層とした独自プロトコル Robonet [90] による分散ネットワークが構成されている.

ロボット体内に更に高度な分散制御系が求められる中で, CAN をはじめとした最大でも 10 Mbps 程度の汎用シリアル通信による分散ネットワークではデータ転送性能に不足が生じてきた. そこで近年では, より高速な 100 Mbps 産業用 Ethernet フィールドバスの採用が進んでいる. LORA [91][31][92] では, 中央計算機に対して複数の中間制御基板が Sercos III で接続され, 中間制御基板に対してモータ制御モジュールが CAN バスで接続される 2 段構成の分散ネットワークが構成されている. 中間制御基板では FPGA および DSC を利用して, 通信の中継と簡易的な異常状態対応が実装されている. その後 LORA の体内分散制御系は, 信頼性の向上を主な目的として既製品のモータ制御モジュールを用いた EtherCAT ネットワークによる実装に更新されている [93]. 他にいくつかのヒューマノイドで EtherCAT が採用されている. RoboSimian [43] では, モータ制御に関わる EtherCAT ネットワークとセンサおよびハンドに関わる Modbus プロトコルの RS-485 ネットワークが併用されている. Hydra [36] では油圧アクチュエータを制御する複数の分散ノードが中央計算機に対して EtherCAT で接続されるネットワークが採用されている. それぞれの分散ノードは複数の基板から構成され, 基板間は FPGA を利用したプロトコル実装による LVDS 通信で接続される. WALK-MAN [39], ARMAR-6 [40], HRP-5P [15][94] でも EtherCAT による分散ネットワークが構成されている.

ヒューマノイドにおける多くの事例が示している通り, 多自由度ロボットの体内制御システムとしては, 高信頼で合理的な実時間制御を実現する分散制御系の構成が適しているものと考えられる. それぞれのアクチュエータ単体を制御する分散制御ノード, および,

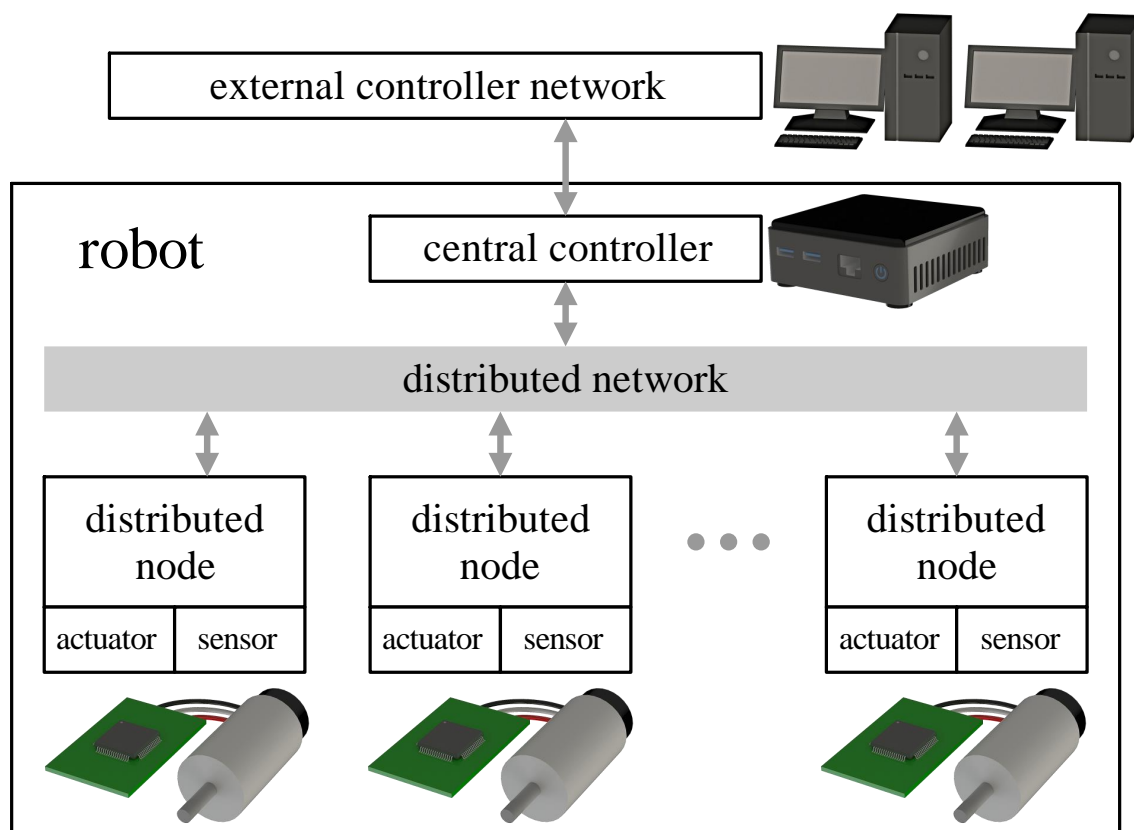


図 2.4 Local distributed control system for multi-DOF robot

身体モデルや高度な動作計画に基づいてロボット全身制御を実行する中枢全身制御系の組み合わせによるネットワークで実時間制御系が実装される．図 2.4 にロボット体内分散制御システムの模式図を示す．

2.3.2 分散制御ノード

現在開発されている多くのロボットは電力を動力源としてアクチュエータに電気モータを採用している．電力および電気モータ以外に，内燃機関・エアコンプレッサ・人工筋肉・形状記憶合金などを利用するロボットも存在する．電気モータとしては，小型で大出力の得られるブラシレスモータ・ブラシ付き DC モータ，あるいは容易で精密な位置制御を得意とするステッピングモータなどが主流で利用されており，その他にリニアモータ・ソレノイド・超音波モータ・圧電モータなどを利用する例もある．電気モータの出力を，直接もしくは減速機を通じたリンク機構によって関節に接続したり，ポンプ等により油圧・空気圧に変換したり，ウィンチによってワイヤ張力に変換したり，といった機械構成

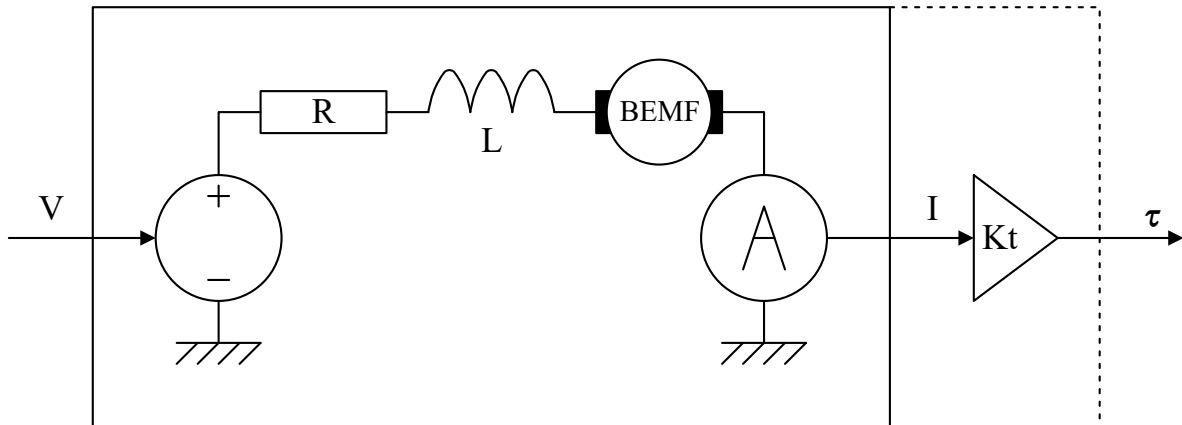


図 2.5 Simplified equivalent single-phase motor output model

によってロボットの関節が駆動される。

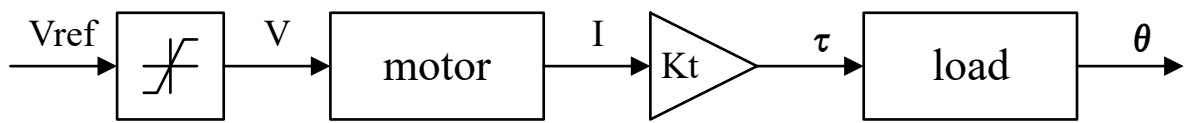
ここでは代表的なアクチュエータモデルとして、ベクトル制御型ブラシレスモータのように等価出力電流に比例した出力軸トルクが発生するモータモデルを例にとって議論を進める。単純化したモータモデルを図 2.5 に示す。モータに電圧 V を印加すると電流 I およびそれに比例したトルク τ が出力される。なお実際のブラシレスモータでは、3 相の端子印加電圧に対して巻線抵抗・巻線インダクタンス・逆起電力に基づく 3 相巻線電流が発生し、電流のうち回転子磁束ベクトルに直交する成分に比例したトルクが発揮される。

分散制御ノードでは受信した指令入力に対して、単一のアクチュエータもしくは関節のモデルに基づいて出力制御を行う。指令入力の種類とその制御構造の概念モデルとして図 2.6 のような例が挙げられる。

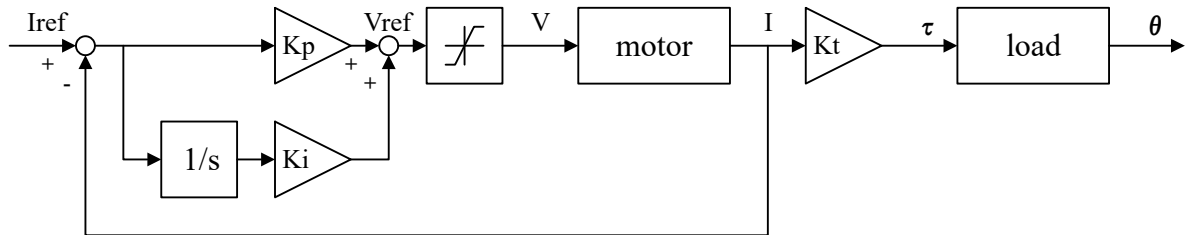
図 2.6 (a) は、電圧指令値の入力を受けてそれをフィードフォワードに印加する電圧制御モデルである。単純構造で比較的 low コストに実装可能である。一方で主に逆起電力の影響により出力電流・トルクは陽に取り扱えない。冷却ファンなどのように低負荷で高速回転させる用途には十分な場合もあるが、サーボ制御に見られる高負荷・低速回転では過大な電流が生じやすく、指令値生成における制御が別途必要となる。

図 2.6 (b) は、電流指令値の入力を受けて PI 制御により電圧を印加する電流制御モデルである。モータ出力トルクを陽に制御できるため機械特性の観点から有意義な制御系である。離散システムでの実装にあたってはおよそ 10 kHz 以上の比較的細かい時間分解能が求められる場合が多い。また離散システムにおいてはむだ時間の影響により比例ゲインを大きく取ることができないため積分項が指令値追従性を担う主要項となる。比例項は応答性の向上に寄与する。むだ時間の問題は 2.4 節で詳しく論じる。

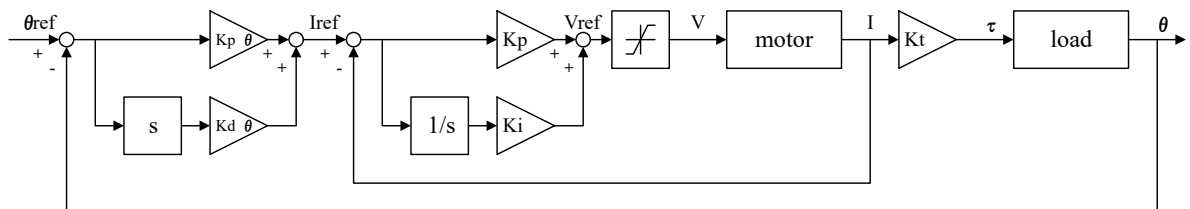
図 2.6 (c) は、位置指令値の入力を受けて PD 制御により電流指令値を決定する位置制御モデルである。指令値を動的に変化させずともローカルに安定するため取扱が容易で、



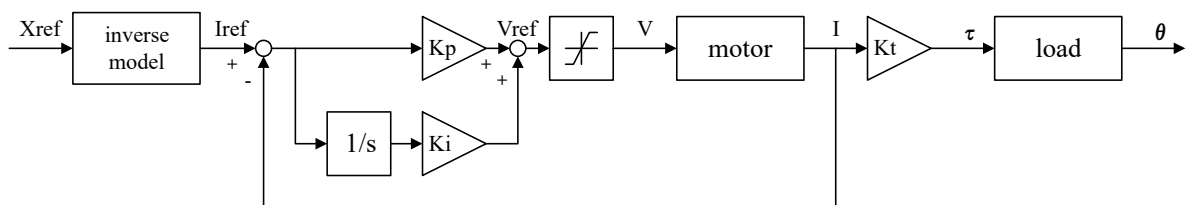
(a) Voltage feed-forward control



(b) Current PI control



(c) Position PD control



(d) Any model-based control

図 2.6 Actuator controller model

多くの位置制御型ロボットで採用されるサーボシステムである．用途によっては位置ではなく速度や加速度を制御する場合も考えられる．また実装によっては位置フィードバックの出力を電流指令値ではなく直接の電圧指令値とする場合も考えられる．

図 2.6 (d) のように，モータ単体だけではなく関節まで含めた出力特性モデルを考慮して任意の応用的な制御系を構築することもできる．例えばモータトルクと関節トルクの間係をモデル化しその逆変換を挿入することで関節トルク制御が可能となる．

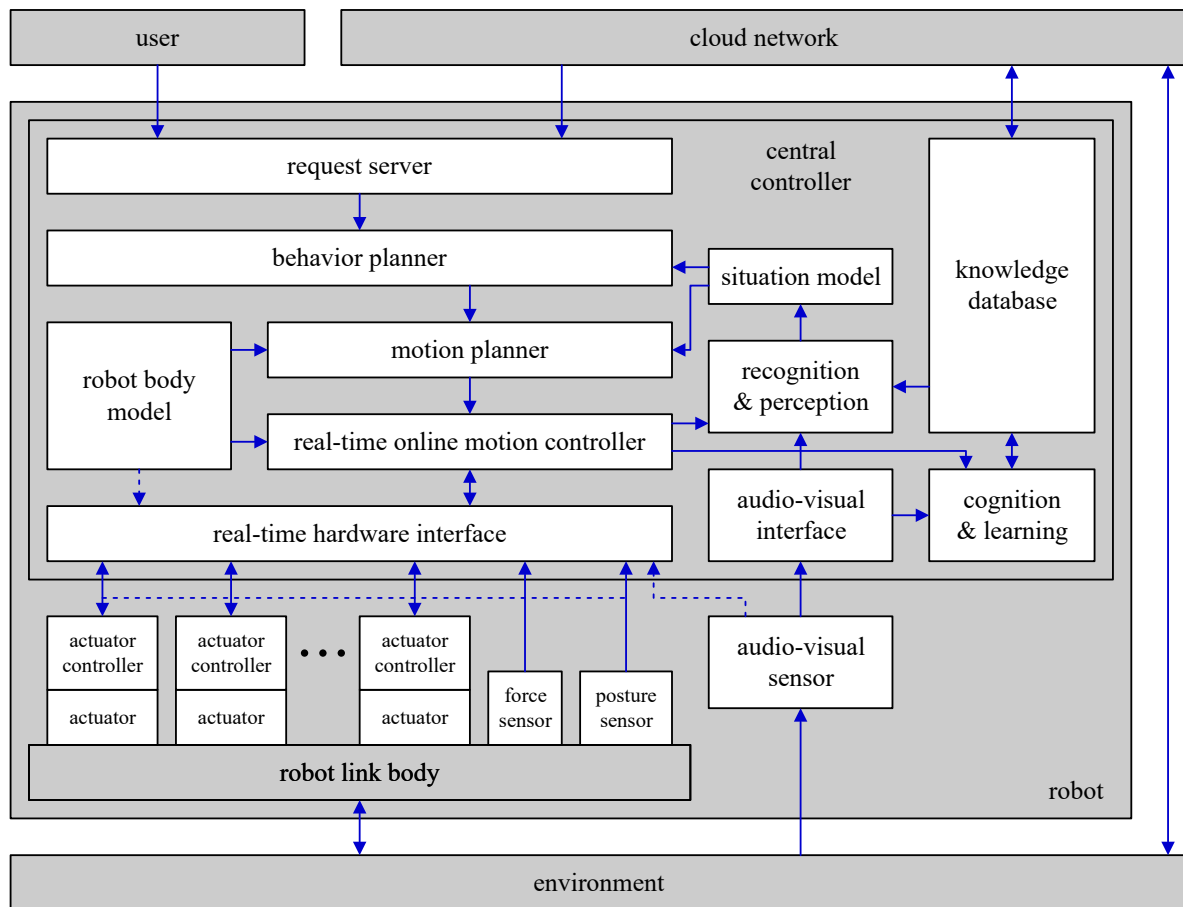


図 2.7 Intelligent robot system architecture

2.3.3 中枢全身制御系

中枢全身制御系では、分散制御ノードと交換した情報に基づいて、ロボット全身の姿勢制御やデカルト空間における操作などを交えた統合的なロボットの行動を実施する。

図 2.7 に典型的な知能ロボットシステムの構成例を示す。多自由度ロボットの制御系は中枢全身制御系と分散制御ノード、およびその他周辺の視聴覚センサなどから構成される。中枢全身制御系には認識・計画・実行に基づいた行動制御システムが実装される。

- 認識

- audio-visual interface : カメラやマイクなどの視聴覚センサからデータを取得するインタフェース
- cognition & learning : 視聴覚データやロボット身体データに基づく環境認識モデルを生成

- knowledge database : 環境認識モデル
- recognition & perception : 環境認識を実行
- situation model : 環境認識結果として得られるロボット状況モデル
- 計画
 - request server : ユーザ入力やその他信号入力による行動指示
 - behavior planner : 抽象的な行動計画
 - motion planner : ロボット身体モデルに基づく具体的な関節軌道計画
 - robot body model : ロボット身体リンク機構モデル
- 実行
 - real-time online motion controller : 実機フィードバックを含むロボット動作制御器
 - real-time hardware interface : 分散制御ノードを含む体内分散制御系に中枢全身制御系を接続してデータを交換するインタフェース

環境適応行動における動的な動作実現のために、特に実行システムには高い実時間性と時間分解能が求められる。多自由度ロボットの複雑な全身制御系の実装では、実時間性の要求を満たし、あるいは高い拡張性を達成するために、非同期マルチプロセス構造が積極的に採用される。制御の応答性に関する議論においてはこの非同期処理構造の影響も重大な要素となる。

2.4 ロボット動作制御におけるフィードバック遅延の課題と実時間性の要求

2.4.1 フィードバックに基づくロボット動作制御

旧来より利用されてきた産業用マニピュレータなどによる定型動作は、多くの場合フィードフォワードに実行されている。中枢全身制御系では計画器が生成した動作に基づいて一方的にアクチュエータに位置指令を与え、指令された通りのロボット動作が再生される。センサデータ入力は動作モードのシーケンシングや動作計画計算などに利用される。

一方で、これからのロボットに期待される環境適応行動においては、環境との相互作用をフィードバックして動作をオンライン修正することによる複雑な行動実現が求められる。

環境接触力の適切な調整を目指す力制御のうち力フィードバック型のものとして、

- エンドエフェクタに力センサを搭載

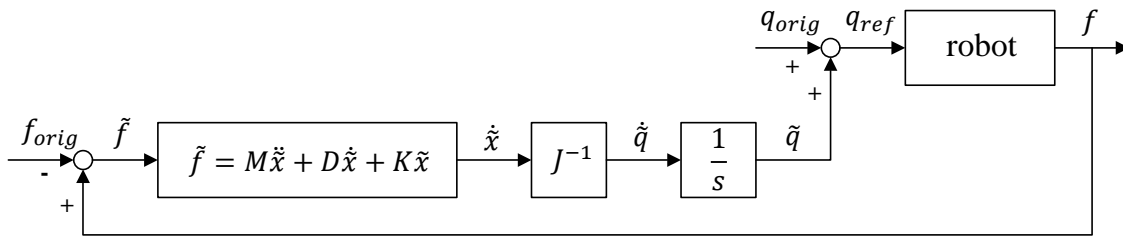


図 2.8 Simplified admittance control model

- 関節ごとに力センサを搭載
- アクチュエータの出力や関節のバックドライブからオブザーバ等で力を推定

などの手法により環境接触力を得てフィードバックする手法が提案されている．エンドエフェクタに搭載した力センサデータをフィードバックすることでデカルト空間に擬似的な機械インピーダンスを再現するアドミタンス制御 [95] は，一例として図 2.8 のように構成される．設定したインピーダンスモデルに基づいてエンドエフェクタの力 f からデカルト空間での運動 \dot{x} を求め，関節空間で積分することで目標動作 q_{ref} が得られる．この目標動作をロボットに再現することで機械インピーダンスに基づくものとおおよそ等価な挙動が得られる．

2.4.2 ロボット動作制御におけるむだ時間の影響

実際の多自由度ロボット全身制御系は 2.3 節に述べたような分散制御系および非同期マルチプロセス構造により，多階層の離散システムで構成される．そのため，機械的な入力によるセンサの反応からそれをフィードバックして行うアクチュエータの出力までには多階層の処理時間や通信時間による遅延が生じる．

フィードバック系に遅延が挿入されることにより生じる問題は，むだ時間系問題 [96][97][98] として知られている．むだ時間 T の挿入は伝達関数 e^{-Ts} で表される．現在の制御出力が時刻 T だけ過去の入力に基づく計算結果を参照するため，遅延の無い力学としては本来安定であるはずのモデルも不安定となりうる．

図 2.6 (b) のモータ電流制御モデルを簡略化した図 2.9 のモデルを例に考察する．モータを抵抗 R で近似し，これを電流 PI 制御により駆動する．フィードバックする電流値には，むだ時間 T が挿入されている．

$$I = \frac{1}{R} \left(K_p + \frac{K_i}{s} \right) (I_{ref} - e^{-Ts} I) \quad (2.1)$$

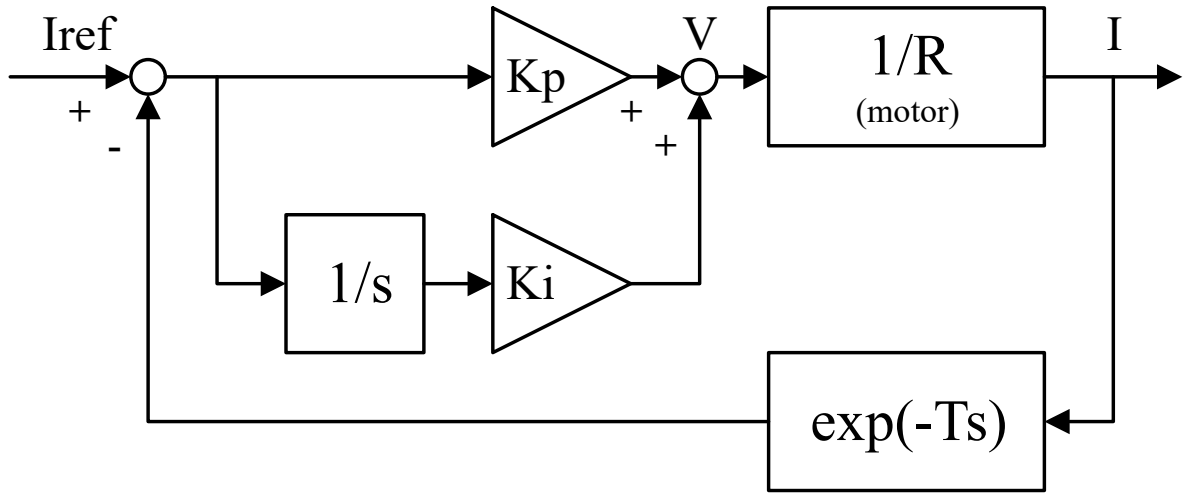


図 2.9 Simplified motor current control model with time-delay

より伝達関数は,

$$\frac{I}{I_{\text{ref}}} = \frac{\left(\frac{K_p}{R}s + \frac{K_i}{R}\right) e^{Ts}}{\left(\frac{K_p}{R}s + e^{Ts}\right) s + \frac{K_i}{R}} \quad (2.2)$$

と表される. [1/1] 次の Padé 近似,

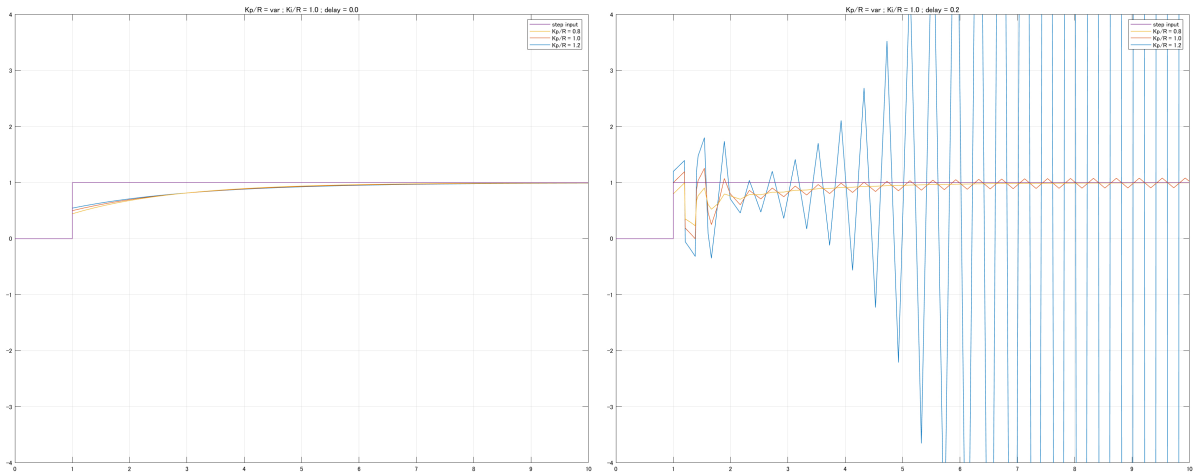
$$e^{Ts} \simeq \frac{1 + \frac{Ts}{2}}{1 - \frac{Ts}{2}} \quad (2.3)$$

により式 2.2 は,

$$\frac{I}{I_{\text{ref}}} \simeq \frac{\frac{K_p}{R}Ts^2 + \left(2\frac{K_p}{R} + \frac{K_i}{R}T\right)s + 2\frac{K_i}{R}}{\left(1 - \frac{K_p}{R}\right)Ts^2 + \left(2\frac{K_p}{R} - \frac{K_i}{R}T + 2\right)s + 2\frac{K_i}{R}} \quad (2.4)$$

と近似される.

$T = 0$ でむだ時間が存在しない場合, システムは安定であり比例ゲイン K_p が大きいほど追従性が向上する. 一方で, むだ時間が存在する場合, 比例ゲイン $\frac{K_p}{R} > 1$ において過去に生じた誤差がより大きな値の逆符号出力に繋がるため, これを繰り返して発散するおそれがある. この特性を顕著に表す例として, $\frac{K_p}{R} = (0.8, 1.0, 1.2)$, $\frac{K_i}{R} = 1.0$ の設定のもとに $T = (0.0, 0.2)$ における大きさ 1 のステップ応答シミュレーション結果を図 2.10 に比較する. むだ時間の無い図 2.10 (a) ではすべての $\frac{K_p}{R}$ で安定した電流指令値追従が見られるのに対し, むだ時間の存在する図 2.10 (b) では $\frac{K_p}{R}$ が 1 を超える $\frac{K_p}{R} = 1.2$ の場合に出力が発散する.



(a) Without delay ($\frac{K_p}{R} = (0.8, 1.0, 1.2)$, $\frac{K_i}{R} = 1.0$, $T = 0.0$), (b) With delay ($\frac{K_p}{R} = (0.8, 1.0, 1.2)$, $\frac{K_i}{R} = 1.0$, $T = 0.2$)

図 2.10 Step response simulation for simplified motor current control model with time-delay

むだ時間による安定性の問題は，むだ時間がシステムの時定数に比べて無視できないほど大きくなる場合に顕著になることも知られている [99]．図 2.9 にモータのインダクタンスを加えて，図 2.11 のようにプラントが遅れ系となる場合を考える．時定数 $\frac{L}{R}$ の 1 次遅れ系である．このとき伝達関数は式 2.2 にインダクタンス項を導入して，

$$\frac{I}{I_{\text{ref}}} = \frac{\left(\frac{K_p}{R}s + \frac{K_i}{R}\right)e^{Ts}}{\left(\frac{K_p}{R}s + \left(1 + \frac{L}{R}s\right)e^{Ts}\right)s + \frac{K_i}{R}} \quad (2.5)$$

と表される．[1/1] 次の Padé 近似結果は，

$$\frac{I}{I_{\text{ref}}} \simeq \frac{\frac{K_p}{R}Ts^2 + \left(2\frac{K_p}{R} + \frac{K_i}{R}T\right)s + 2\frac{K_i}{R}}{\frac{L}{R}Ts^2 + \left(1 + \frac{2L}{T} - \frac{K_p}{R}\right)Ts^2 + \left(2\frac{K_p}{R} - \frac{K_i}{R}T + 2\right)s + 2\frac{K_i}{R}} \quad (2.6)$$

となり，式 2.4 と比べて比例ゲイン $\frac{K_p}{R}$ の安定限界に $\frac{2L}{T}$ だけ余裕が生じる．すなわちむだ時間 T が時定数 $\frac{L}{R}$ と比べて小さいほど，安定した高追従制御系を構成することが容易となる．

図 2.8 のアドミタンス制御をはじめとした力制御においても同様にむだ時間による問題が生じる [100]．むだ時間が存在しない場合は，

$$\tilde{f} = M\ddot{\tilde{x}} + D\dot{\tilde{x}} + K\tilde{x} \quad (2.7)$$

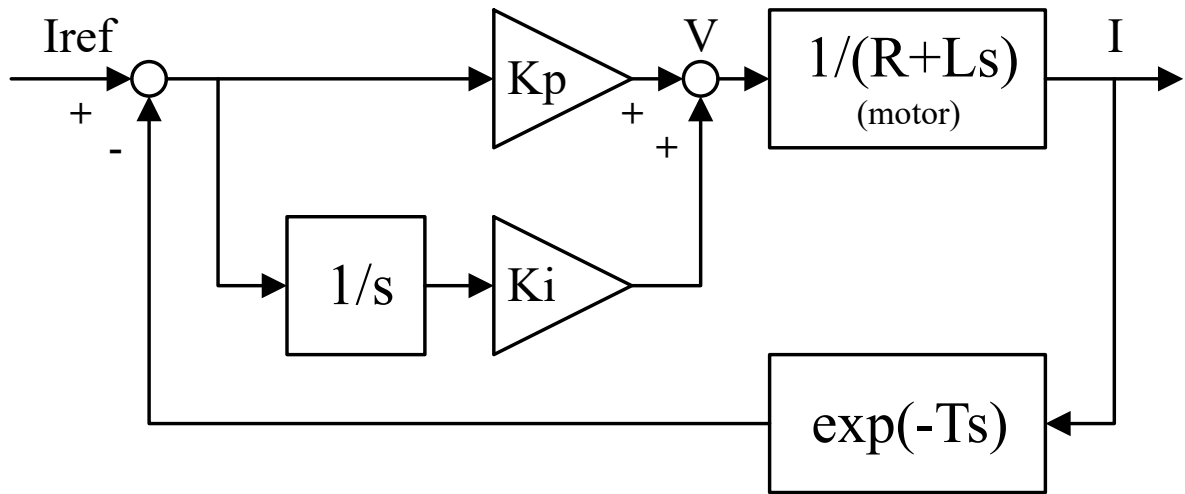


図 2.11 Simplified motor current control model with time-delay and plant lag

のようにモデル化された見かけ上の機械特性のみに従って安定性を議論することができるが、プラントの時定数と比べて大きなむだ時間が存在する場合はフィードバックシステム全体の比例ゲインに関する複雑な議論が必要となる。

フィードバック遅延による問題は、システムの定常的な安定性だけでなく過渡的な現象への対応においても生じる。衝撃的な外乱が加えられた場合に、フィードバック遅延時間より小さな時間での変動をシステムで取り扱うことはできない。ロボットに力制御を適用する主要な目的のひとつに、過大な環境接触力の発生を避ける、というものが挙げられる。過大な力が生じると内外の損壊に繋がるおそれがあるため、環境との接触に対してよく馴染む柔軟な制御が求められる。しかし機械的な伝達時間よりも大きなむだ時間がフィードバック制御系に存在すれば、衝撃的な接触力に応答しこれを所望の値に抑えることは不可能となる。

2.4.3 フィードバック遅延の解決によるロボット行動実現

2.4.2 項で考察したように、ロボットの制御系に大きなフィードバック遅延が存在した場合、

- 安定した高追従制御系の構成が困難
- 衝撃的な外乱への応答が不可能

という問題が生じる。むだ時間を含むシステムがモデル化可能な場合は、Smith 予測器などによる補償で安定性の問題は解決される [101]。一方で衝撃的な外乱への応答は、衝撃

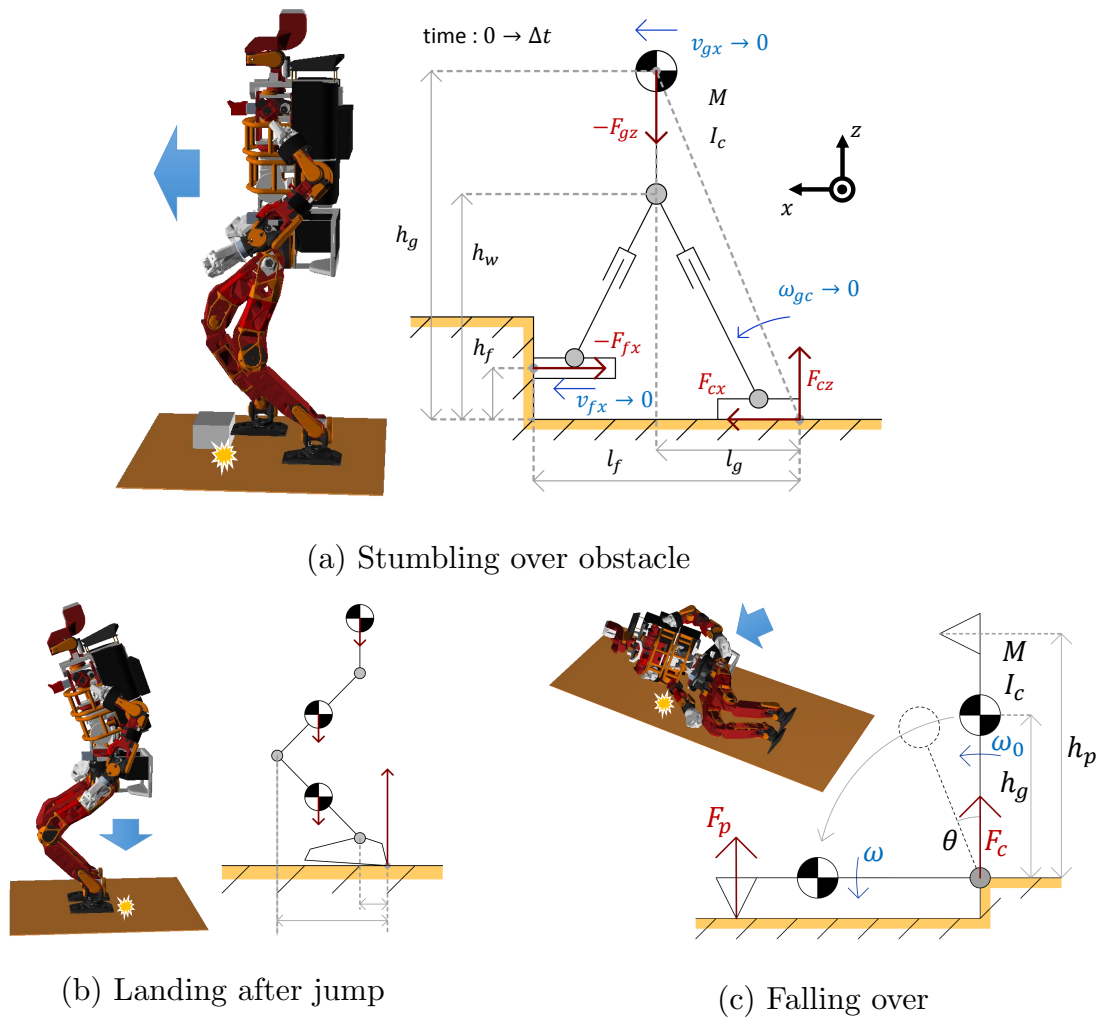


図 2.12 Humanoid robot model with impact force [102]

力の時定数を超えるような遅延を伴うと原理的に不可能となる．表 2.4 に挙げたような環境接触を伴う行動の中でも，大きな衝撃力を伴うものは従来実現が困難であった．逆にフィードバック遅延をシステムの時定数よりも小さく抑えることができれば，これらの問題により未達成であったロボット行動目標の実現可能性が開かれる．

ヒューマノイドの場合，図 2.12 に例示されるような躓き・着地などの衝突を伴う行動において衝撃緩和および応答性の高い姿勢制御が可能となる．これにより，転倒や内外の損傷を避けることが容易となる．更に，打撃作業・スポーツ・跳躍・走行などの応用的な行動実現への可能性も開かれる．

筋骨格ロボットの場合，柔軟でモデル化し難い身体および非常に多数のアクチュエータによる干渉駆動系のため，所望の動作を再現するだけの追従性が重要となる．一方で，多数のアクチュエータ制御に伴う多量の体内データ通信により，低遅延化や制御周期の短縮

が困難となっている場合も多い。腱駆動型筋骨格ヒューマノイド [11] では2脚でのバランス保持や歩行の実現に課題を残している。

2.4.4 実時間性を満たすロボット体内制御システムの要件

ロボット動作制御のフィードバック遅延を解決する上で、ロボット体内制御システムにおいて実時間性を満たすことが重要な課題となる。実時間性とは、規定された応答時間制限を高信頼に満たす性質のことを指す [103]。ロボットが環境と相互作用を繰り返す中で、機械的入力を受けて十分低遅延に機械的出力に反映させる性能を単に有するだけでなく、信頼性を伴って低遅延フィードバックを確実なものとするこも、ロボット体内制御システムに求められる実時間性の重要な要件である [65][104]。

実時間性を満たすことに加えて、実際のロボットに展開し運用してゆく上での合理的なシステムとして構成することも重要である。ヒューマノイド・筋骨格ロボット・フィールドロボットといった多様な形態の多自由度ロボットに共通して適用可能な設計および実装により、ロボット体内制御システムとしての真価が発揮される。

図 2.13 の項目を、本研究で目指す体内分散実時間制御システムの要件として掲げる。多形態多自由度ロボットの環境適応行動実現に適う体内制御システムには、実時間性および展開実用性を兼ね備えることが求められる。

実時間性は次の特性によって満たされる。

- 低遅延 (low latency) : センサ入力をアクチュエータ出力に反映するまでの平均的な遅延を十分小さく抑えるデータ転送性能および制御演算性能を備えること
- 低ジッタ (low jitter) : データ転送や周期制御のタイミングを安定に保ち最悪フィードバック遅延を小さく抑えること
- 信頼性 (dependability) : 電磁ノイズや振動を伴うロボット体内環境の中で継続動作しフィードバック制御を確実に実行すること

所望の応答時間に十分間に合わせるだけの通信・演算性能向上とフィードバック遅延低減を達成することが前提として求められる。その上で、低ジッタで確実なフィードバック制御実行を保証する実装によって、実時間性が満たされる。ロボット体内では、多数のアクチュエータを駆動するための変動的な大電流が発生するため、電源や通信信号がノイズの影響を受けやすい。またロボットがダイナミックな動作を繰り返す中で、衝撃や振動が発生し、またケーブルの摺動が生じる場合もあるため、断線やコネクタの接触不良への対策も求められる。このような劣悪な環境の中で高信頼に低遅延フィードバック制御を実行する必要がある。

展開実用性は次の特性によって満たされる。

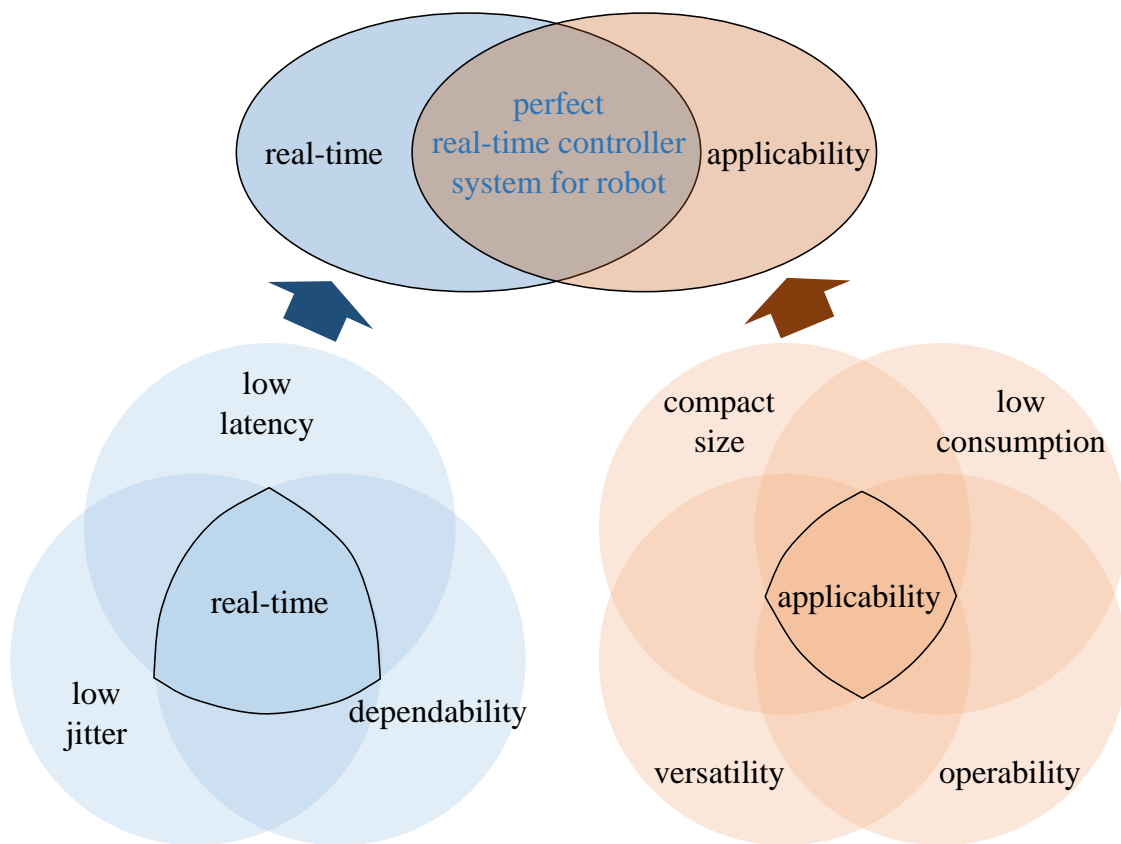


図 2.13 Requirement for desired real-time robot controller system

- 小型軽量 (compact size) : スペース制約の厳しいロボット体内に収まるサイズでの実装を可能とすること
- 低消費電力 (low consumption) : 余計な発熱を抑えた高効率な制御実行により長時間の継続動作に耐えること
- 汎用性 (versatility) : 多様なロボットに共通構造で適用可能な汎用性および継続的開発に向けた拡張性を備えること
- 運用性 (operability) : 容易な利用・複製・保守開発を可能としロボットへの展開における高い利便性を備えること

多自由度ロボットでは限られた体積の中に多数のアクチュエータを駆動する体内制御システムを実装する必要がある。同時に搭載可能な電池のサイズや容量も限られる。このような制約を持つ多様な形態のロボットで共通して実用化できる小型軽量・低消費電力な体内制御システムの実装と継続的開発への展開が課題となる。

2.5 分散ネットワークにおける低遅延データ転送技術

2.5.1 産業用フィールドバス

組込機器同士を接続するネットワーク技術はファクトリオートメーションをはじめとした産業機器分野のフィールドバスネットワークを中心に発展してきた [105][106]。計算機やインターネットの技術発展および普及を受けて、2011 年には Industry 4.0 が提唱され、サイバフィジカルシステム・クラウドコンピューティング・IoT などのコンセプトを交えた高度な分散オートメーションシステムの開発が進められている。

1980 年代から 1990 年代前後には、RS-485 汎用シリアル通信など数 Mbps 程度のシリアル通信規格を物理層に利用したフィールドバス通信プロトコルが多く提案されている。現在でもよく利用されている例として、Modbus (1979)・CAN (1986)・Sercos (1987)・PROFIBUS (1989) などが挙げられる。中でも CAN は車載ネットワークで広く普及し、2000 年頃には低コスト化を図った LIN (1999) やパフォーマンスを向上させた FlexRay (2000) および MOST (2001) といった次世代プロトコルの展開へと繋がっている。Sercos ではシリアル通信の物理媒体として光ファイバを利用することで耐ノイズ性向上や絶縁化を図っている。

1990 年代頃にはインターネット技術が急速に発達し、LAN 向けの規格として 100Mbps の 100BASE-TX や 1Gbps の 1000BASE-T などの高速な Ethernet 通信が実用化された。これに伴い 2000 年代前後には主に 100Mbps Ethernet を利用したフィールドバス通信プロトコルが多く提案され、現在でもパフォーマンスの求められるファクトリオートメーション現場での主力となっている。近年では車載ネットワークにおいても、従来の CAN・FlexRay・MOST での帯域不足を受けて Ethernet の導入が進んでいる [107]。Ethernet 物理層に基づく拡張により特に低遅延なネットワークを実現しているプロトコルの例として、Ethernet POWERLINK (2001)・PROFINET (2001)・EtherCAT (2003)・Sercos III (2006)・MECHATROLINK-III (2007)・CC-Link IE (2007) などが挙げられる。

現在広く利用されているフィールドバス通信プロトコルは国際規格 IEC 61158 (Industrial communication networks - Fieldbus specifications) および IEC 61784 (Industrial communication networks - Profiles) として標準化されている。

2.5.2 高性能計算機のペリフェラル通信インタフェース

多自由度ロボットの中核全身制御系には非常に高い計算能力が求められる。CPU や SDRAM などに利用される半導体技術の進展とともに、計算機の性能は時を重ねるごと

表 2.5 Typical communication interfaces for recent PC

interface	typical bit rate	feature
PCI Express	5 Gbps	card edge
USB 2.0	480 Mbps	
USB 3.0	5 Gbps	USB 2.0 compatible
Ethernet	1 Gbps	lock connector
Thunderbolt	20 Gbps	
eSATA	3 Gbps	for storage/drive
IEEE 1394	800 Mbps	a.k.a. FireWire

に益々向上している。ロボットの制御技術もまた、計算機の性能向上とともに革新を重ねてきた。近年では、ロボットに搭載可能なほど省スペース・低消費電力ながら、旧来メインフレームやワークステーションなどと呼ばれてきた大規模計算機に匹敵あるいはそれを凌駕するような性能を有する小型計算機製品も多く実用化されている。これらの高性能計算機を組込分散制御系に接続する通信インタフェースの選択も重要な要素となる。

計算機製品には通常、その時代の普及状況に即したペリフェラル通信インタフェースが搭載される。現在多くの小型計算機には表 2.5 のような通信インタフェースが搭載されている。なおここでは画面表示用の接続規格に関しては取り扱わない。

PCI Express は現在の計算機システムにおいて根幹的な役割を担うシリアルバス規格である。単一レーンの片方向差動シリアル信号で 2.5 Gbps の帯域を有する PCI Express Gen1 が 2005 年に実用化されて以来、高速化が進められバージョンが更新されている。ペリフェラルデバイスとの接続のほか CPU やチップセットの I/O バスとしても多用され、他の多くのインタフェースは PCI Express の拡張や変換によって実装される。USB は現在特に普及しているペリフェラルデバイス接続インタフェースである。USB 2.0 までに対応している半二重通信では、1.5 Mbps の Low-Speed、12 Mbps の Full-Speed、480 Mbps の High-Speed が定義されている。USB 3.0 以降で利用される高速全二重通信では、5 Gbps の SuperSpeed、10 Gbps または 20 Gbps の SuperSpeed+ が定義されている。更に高速な USB4 の規格化も進められている。Ethernet は計算機およびネットワーク機器同士を接続する LAN の通信規格であり、2.5.1 項でも述べたようにフィールドバスへの応用も進められている。USB および Ethernet は現在大半の計算機製品に搭載されており、入手性の観点から非常に優れている。その他に Thunderbolt・eSATA・IEEE 1394 といったインタフェースも場合により搭載される。

あるいは独自の組込機器に高性能な汎用計算機モジュールを規格化された多線バスコ

ネクタで接続可能にする試みとして、Computer on Module (CoM) および System on Module (SoM) のコンセプトおよび製品群が存在する。代表的な CoM 規格の例として COM Express が挙げられる。一般的な小型計算機と比べて頑強なコネクタで広帯域に組込制御系へ接続可能であり、ロボットへの応用も期待されるが、実用化されている既存製品としては CPU 性能が限られているのが実情である。SoM 方式は組込計算機においてもよく用いられており、近年では FPGA SoC や GPU SoC などの多機能・高性能な組込計算機を搭載した製品も多く登場している。

2.5.3 ロボット体内分散ネットワーク

多自由度ロボット体内制御システムにおいても、産業用フィールドバス技術や計算機通信インタフェースを活用したネットワークが構成されてきた。

ヒューマノイド研究開発に見られる分散通信プロトコルの採用事例は 2.3.1 項で紹介した。ヒューマノイド体内制御システムを分散制御化する流れの中で CAN による分散ネットワークが多く採用されている。CAN 以外にも RS-485 や LVDS を利用した汎用シリアル通信バスによるネットワークの採用事例がいくつか見られる。近年では高速化の需要から 100 Mbps Ethernet に基づくフィールドバスプロトコルである EtherCAT の採用事例が増えてきている。高速化に向けて Ethernet 以外の高速信号を利用する事例として、SpaceWire [81][45] や Responsive Link [65][108][109] による手法も提案されている。

中央計算機を分散ネットワークに接続するインタフェースとしては PCI バスがよく利用されている。CAN などのネットワークインタフェースを PCI バスに変換する中継基板を利用して接続する構成が取られる。EtherCAT の採用に伴い、中央計算機のインタフェースとして Ethernet を用いる例も増えている。アクチュエータ制御を含む主要な分散ネットワークの他に、センサなどその他のデバイスを別の経路で中央計算機に接続する場合もある。USB や IEEE 1394 などのペリフェラルデバイス接続インタフェースが併用される場合も多い。

従来のロボット体内制御システム研究開発においては、産業分野で十分成熟した技術やアプリケーションを実装する比較的小型で高性能なモジュール製品を組み合わせることで全体システムを構成する手法が主流となっている。CAN や EtherCAT のインタフェースを搭載したモジュール製品を導入し、その仕様に従ったネットワークの運用が行われている。実績のある実装を低コストで導入できるメリットが得られる一方で、ロボットの行動実現への挑戦が既製品の実装仕様により大きく制限されてしまう。

ロボットの研究開発の中では体内制御システム実装における課題が常々新たに明らかとなる。体内分散ネットワークにおいて、データ転送レートが律速となったり、プロトコル仕様で 1 周期に交換できるデータ量の制限が律速となったり、あるいはジッタ・オーバ

ヘッド・ノイズ耐性などが問題となる場合もある。CAN をはじめとした汎用シリアル通信のデータ転送レートに不足が生じて EtherCAT に置き換えられているのと同様に、100 Mbps の物理層の上で個別の仕様を定めて提供される EtherCAT 製品でもまた、多自由度化するロボット体内で多数のデータを転送するには十分でなくなっている。要求に応じてロボットの行動実現を支えてゆくためには、最新の技術に追従しロボット研究開発の現場で柔軟に実装仕様を更新する体内制御システムの継続的な開発が重要である。

本研究では、現在得られる半導体技術を活用して、ロボットの行動実現において生じる体内制御システム実装への要求に応える基盤システムの研究開発を試みる。第 3 章で、FPGA と汎用の通信物理層部品を搭載して小型パッケージ化した、体内分散ネットワーク構成モジュールの開発に関して述べる。従来の汎用シリアル通信の性能を大きく凌駕する高速光通信による組込分散ネットワークを構成する。中央計算機と接続する基幹通信ネットワークでは Gigabit Ethernet や USB High-Speed の物理層仕様を柔軟に適用する。FPGA を活用することで、ロボット研究開発の現場において子細な実証評価に基づき妥協なく通信ネットワークやアクチュエータ制御の仕様を更新し、次世代の技術へ繋げてゆくことを可能とする。

ロボット体内制御システムにおける通信技術適用指針を図 2.14 に示す。通信技術は先端計算機ネットワーク分野で開発され、産業分野に応用される。産業分野のファクトリオートメーションや車載コントローラにおいて通信技術が普及し、一般的なアプリケーションを実装する応用メカトロニクス製品が開発される。従来のロボット研究開発では、CAN や EtherCAT などのアプリケーションプロトコルを実装した製品を導入する方式が主流であった。ロボット研究開発の要求が産業分野のアプリケーション製品に反映されるまでは多くの時間を要する。FPGA を活用して、Ethernet・USB・高速シリアル通信などの基礎的な通信技術を柔軟に利用する実装により、ロボット研究開発で生じる要求に素早く応じ、更に次世代の通信技術の模索へと直接繋げることができる。

2.6 組込計算機における低遅延演算技術

高性能計算機製品の演算性能が時を重ねるごとに益々向上している一方で、組込計算機においても同じく技術革新が重ねられ演算性能が向上している。

携帯電話やゲーム機をはじめとした組込モバイル機器に搭載される CPU において、現在 ARM アーキテクチャ [110] が支配的なシェアを掌握している。演算性能の向上と消費電力の低減を目指した 32 bit アーキテクチャとして 1985 年に ARM1 が開発され、2012 年には 64 bit アーキテクチャである ARM Cortex-A53 が登場している。ARM に代表されるような組込計算機向けアーキテクチャをマルチコア実装したハイエンドな SoC も多数実用化され、Linux などの高度な OS とともに、x86 系アーキテクチャなどによる高性

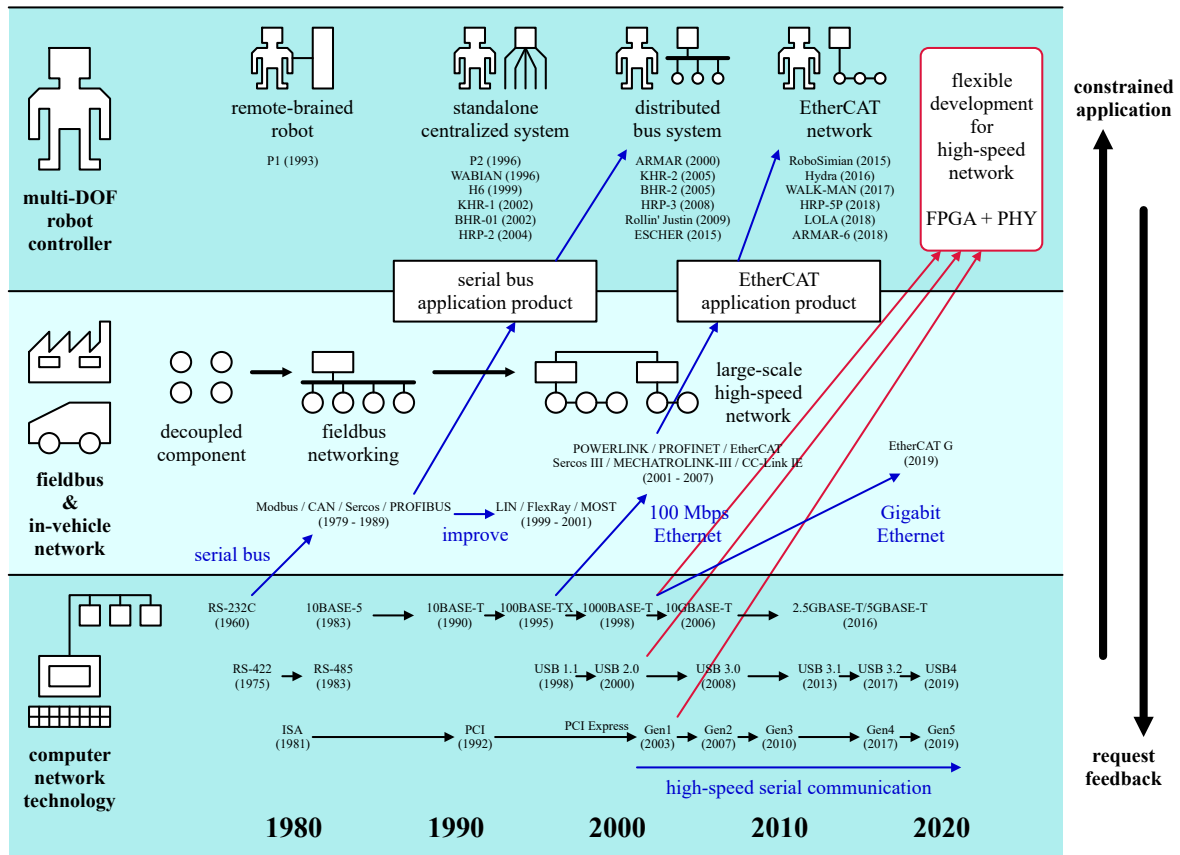


図 2.14 Development paradigm for robot network

能計算機の場合とおよそ同様の構造でソフトウェアを実行する実用的な環境が提供されている。

IoT やエッジコンピューティングの需要も高まり，組込計算機におけるハードウェアアクセラレーション技術もまた進展している．組込向けに小型・低消費電力化した GPU [111] や FPGA [112]，あるいは近年では TPU [113] や VPU [114] など機械学習向けの新しい並列計算機によるハードウェアアクセラレーションが提案されている．Liu [115] は YOLOv2 による物体検出アルゴリズムを例に取り，GPU と FPGA による組込アクセラレーションのパフォーマンスを比較している．ロボットの制御計算においてもハードウェアアクセラレーション技術を利用している例がいくつか見られる [116][117]．

ハードウェアアクセラレーションを抽象的に利用するための API 標準化もいくつか提案されている．マルチコアプロセッサ環境においては並列プログラミングを補助するために OpenMP が提供されてきた．ハードウェアアクセラレータを利用したヘテロジニアス並列プログラミングにおいても同様に OpenCL や更に抽象度の高い OpenACC といったフレームワークが提供されている．組込ハードウェアアクセラレータを利用して深層学習

における推論を実行するための OpenVINO も普及が進んでいる。

CPU からのメモリアドレス空間やバスの操作をハードウェア制御に直結させることが可能な点も，組込計算機を利用した低遅延制御に向けた重要な要素となる。組込制御系とフィールドバスで接続された高性能計算機からは，あらゆるハードウェア制御のために所定のプロトコルによる通信が必要となり，大きなオーバーヘッドが生じる傾向にある。一方，組込計算機では，同様のハードウェア制御処理をオーバーヘッドの小さな単純メモリアクセスで実行することも可能となる。特に FPGA を利用すると，プログラマブルなハードウェア構成によりこの利点を非常に効率的かつ柔軟に活用することができる。

組込計算機を活用すると消費電力の抑制もより容易で明確となる。RISC 型の ARM 系アーキテクチャや高性能計算機を代表する CISC 型の x86 系アーキテクチャなど，一般的に同量の計算に対する電力効率においてアーキテクチャごとに大差は生じない [118]。組込計算機では，求められる制御計算量の計画に対して不必要な計算機リソースやクロック周波数を容易に抑えられるメリットを得やすい。更に GPU や FPGA によるハードウェアアクセラレーションを組み合わせると，同量の計算量に対して高効率な計算も可能となる [119]。

本研究では組込分散ネットワークにおいて組込計算機や FPGA を活用した低遅延フィードバック制御を実現する。分散制御ノードでは FPGA と MCU を組み合わせてサーボ制御を行う。中枢全身制御系では中央計算機に加えて，FPGA SoC や GPU SoC で構成される組込計算機を利用して低遅延な全身フィードバック動作制御を行う。第 4 章で，FPGA SoC および GPU SoC を搭載した組込中間層計算機的设计開発とその要件に関して述べる。大量のデータを応答的に処理する組込計算機やハードウェアアクセラレーション技術を，ロボット体内に適用可能な小型パッケージとして実用化するための要件を明らかにする。

2.7 ロボット体内における多階層分散制御技術

2.7.1 ヘテロジニアス分散制御モデル

能力の均一でない計算機が分散接続されて一連の計算を実行するシステムはヘテロジニアス並列計算環境と呼ばれている。並列計算環境におけるクラスタ型並列プログラミング手法の標準化規格として MPI が広く利用されている。並列計算環境において一連の計算をどのようにタスク分割しどのように各計算機に割り当てるべきかという問題は，タスクスケジューリング問題として計算機科学に基づく研究が進められている [120][121]。

タスク T_i を計算機 M_p で処理するのに要する計算時間を t_{ip} とおく。また，タスク T_i を計算機 M_p で完了してからタスク T_j を計算機 M_q で開始するまでに必ず経過する時

間，すなわち通信時間を c_{ijpq} とおく．タスク計算時間を頂点，通信時間を辺として，頂点を辺で接続した有向非巡回グラフを考えることで一連の計算に必要な時間が議論できる．計算時間および通信時間のそれぞれにおいて，次のような代表的パターンモデルがよく用いられる．

- single
 - ひとつの計算機のみ利用する場合
 - 計算時間はタスク T_i の仕事量 w_i のみで決定され $t_{ip} = w_i$
 - 通信時間は $p = q$ において通信が発生せず $c_{ijpq} = 0$
- identical
 - 計算機によらず性能が一定，すなわちマルチコアプロセッサなどのホモジニアス並列計算環境の場合
 - 計算時間は single と同様に $t_{ip} = w_i$
 - $p \neq q$ の通信時間はタスク T_i と T_j の間の通信量 c_{ij} のみで決定され $c_{ijpq} = c_{ij}$
- uniform
 - 計算機ごとの速度性能に従う一様に線形なヘテロジニアス並列計算環境の場合
 - 計算時間は計算機 M_p の速度 s_p によって $t_{ip} = \frac{w_i}{s_p}$
 - $p \neq q$ の通信時間は計算機 M_p と M_q の間の通信帯域 b_{pq} によって $c_{ijpq} = \frac{c_{ij}}{b_{pq}}$
- unrelated
 - uniform の形式でモデル化できない一般の計算時間 t_{ip} および通信時間 c_{ijpq}

実装される制御器の計算時間を議論する場合，有向非巡回グラフに始点と終点を設け，始点を制御入力，終点を制御出力として考えれば良い．ロボット全身のフィードバック遅延を求める場合は，始点がセンサ入力，終点がそれをフィードバックするアクチュエータ出力となる．

2.7.2 ヒトの神経系に見られる多階層環境適応フィードバック機構

ロボットのフィードバック制御系を構成する上で，生物の神経科学において積み重ねられてきた知見の中にも多くのヒントが見いだせる．このような発想からニューロロボティクス分野の研究が盛んに行われている [122][123]．

ヒューマノイドをはじめとした多自由度ロボットのフィードバック動作制御系においてもいくつかの試みが見られる．Ott ら [124] は神経系のフィードバック構造を分析し，旧来のロボットモデルベースのバランス制御手法と人間のモデル化による外乱推定補償機構 [125] を比較することでヒューマノイドのバランス制御のあり方について考察している．

Capolei ら [126][127] は小脳を模したネットワークによるヒューマノイドの運動学習機構をシミュレーションにより検討している。Abadía ら [128] は小脳を模した遅延補償機構を利用してロボットのフィードバック動作制御系に存在する可変むだ時間に対応する運動学習機構を提案し双腕ロボットの遠隔操縦によって検証している。

解剖学や神経科学の分野でよく知られているように、ヒトの神経系は次のように構成される。

- 中枢神経系 (central nervous system)
 - 脳 (brain)
 - * 大脳 (cerebrum)
 - ・ 大脳皮質 (cerebral cortex)
 - ・ 白質 (white matter)
 - ・ 大脳基底核 (basal ganglia)
 - * 間脳 (diencephalon)
 - * 脳幹 (brain stem)
 - ・ 中脳 (midbrain)
 - ・ 橋 (pons)
 - ・ 延髄 (medulla oblongata)
 - * 小脳 (cerebellum)
 - 脊髄 (spinal cord)
- 末梢神経系 (peripheral nervous system)
 - 体性神経系 (somatic nervous system)
 - * 運動神経 (motor nerve)
 - * 感覚神経 (sensory nerve)
 - 自律神経系 (autonomic nervous system)
 - * 交感神経系 (sympathetic nervous system)
 - * 副交感神経系 (parasympathetic nervous system)
 - * 腸管神経系 (enteric nervous system)

生理学の観点からヒトの身体機能は、

- 動物性機能：随意的な行動に関わる機能
- 植物性機能：不随意的な生命維持に関わる機能

に分類される。ここではロボット制御システムの構成に関連してヒトの動物性機能を考察する。

ヒトの行動における基本的な出力経路では、大脳皮質の運動前野をはじめとする高次運

動野において行動が計画され、同じく大脳皮質の一次運動野で動作・運動に変換され、主に脳幹(橋)・脊髄・運動神経を通じた遠心性投射によって全身の筋に出力される [129][130]。一方、感覚器からの入力には主に感覚神経・脊髄・脳幹を通じた求心性投射により大脳皮質の感覚野に送られて、一次運動野における動作生成に利用され、あるいはその他の高次機能に接続される [129][131]。高次運動野と一次運動野はそれぞれ図 2.7 の behavior planner および motion planner の機能ブロックに相当する。

大脳皮質による行動・動作計画のフィードバックには大きな遅延が発生するため、ヒトの神経系にはこの他に脊髄および小脳によるフィードバック動作制御機構が備えられている。

最も低遅延なフィードバック経路は、感覚神経から脊髄を経由して運動神経に信号が伝達される反射弓である。筋に強く関連する反射として、筋長の変化で興奮する筋紡錘や筋張力の変化で興奮する Golgi 腱器官といった受容器による深部感覚をフィードバックして、筋を収縮・弛緩させるものが知られている [132]。単一の筋に対して働く反射には、筋紡錘の入力に従って伸びた筋を収縮させる伸張反射、および Golgi 腱器官の入力に従って過大な筋張力を緩和させる Golgi 腱反射がある。主となる反射に対して、拮抗する筋に作用する相反性神経支配や対の肢に作用する交叉性伸展反射といった仕組みにより全身で反射の統合が行われる。反射は一次運動野の判断により抑制される場合もある。反射を含む脳幹・脊髄における信号伝達機能は図 2.7 の real-time hardware interface ブロックに相当する。

小脳は滑らかで動的な全身動作制御およびそれを実現するための身体モデル学習を担うことが知られている [133][134]。学習に基づいて全身動作を生成し、平衡感覚や筋の深部感覚に従った調整を施して運動神経に伝達する。小脳は図 2.7 の real-time online motion controller ブロックに相当する。

ヒトの神経系における動物性機能構成とロボット制御システムの対比をまとめると図 2.15 のようになる。

多自由度ロボットの中核全身制御系における全身動作制御機構は、ロボット身体モデルと密接に関わるため、従来 1 台の高次な計算機で集中的に実装されてきた。一方でヒトの神経系では、高次な行動・動作計画を担う大脳皮質とは別に、脊髄反射や小脳による動作制御系がより応答的な神経回路として設けられている。複合的で協調的な中枢神経系の構成により複雑な行動を実現する優れた構造となっている。本研究では脊髄や小脳に相当する応答的なフィードバック制御系を組込計算機のレイヤに実装し、2.4 節に述べたようなフィードバック遅延の問題を改善することを目指す。

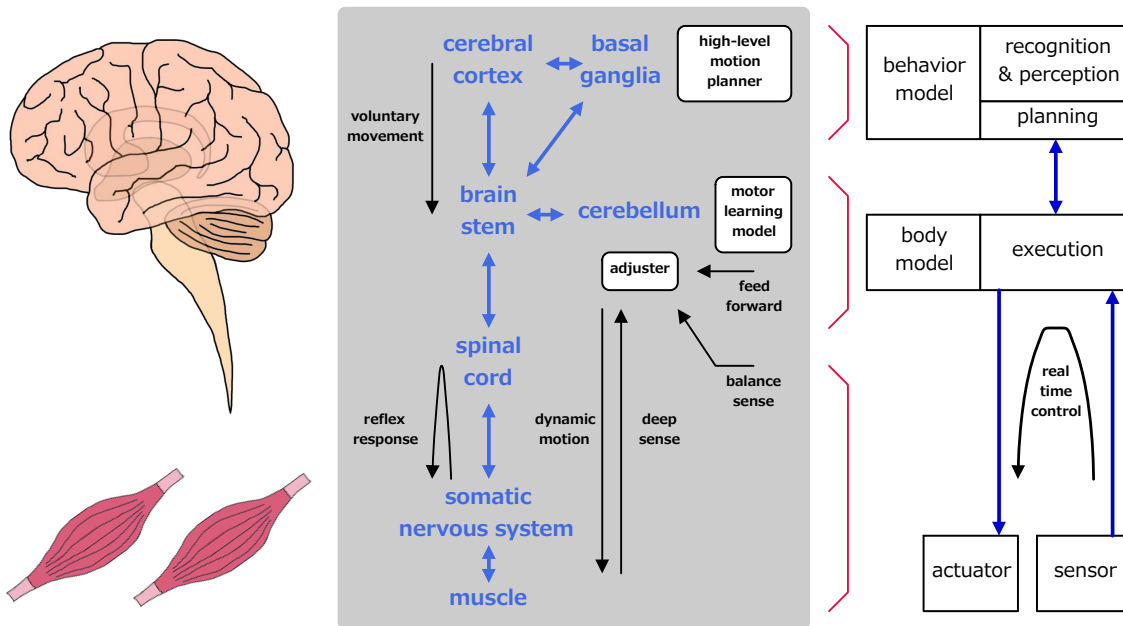


図 2.15 Functional contrast between human nervous system and robot control system

2.7.3 ロボット実時間制御の分散実行プラットフォーム

ロボットの中枢全身制御系では複数の計算機ハードウェアや複数のソフトウェアを利用して制御系が多くのタスクに分割して実装される。実時間性の求められる実行系ソフトウェアを分散実行する上で、タスクの分割構成法やタスク間のデータ交換が重要となる。

計算機ネットワークにおいてコンポーネント化されたロボット制御タスクを分散実行するためのプラットフォームがいくつか提案されている。RT-Middleware [135] の代表的実装である OpenRTM-aist は分散オブジェクト指向技術 CORBA を利用した共通化インタフェースによりロボット制御コンポーネントを接続するシステムを提供している。ROS [136][137] は出版・購読モデルに基づく独自のメッセージ通信システムにより多様なパッケージを接続するロボット制御環境を提供している。ROS を発展させた ROS 2 [138] では出版・購読モデルを標準規格化した DDS に基づいた通信が実装されている。これらの中で汎用性やスケラビリティが重視され、多様なロボット環境で多様なコンポーネントが適用できる優れたシステムが実装されている。

RT-Middleware においてタスク群を実時間実行する手法も検討されている [139][140]。Linux OS に実時間カーネルを適用した移植性の高い実装が多く提案されている。

組込技術を交えてロボット制御系を展開してゆくためのフレームワークも提案されている。ORiN [141] は産業用ロボットにおける装置制御インタフェースの共通規格化プラッ

トフォームを提供している。IXM [142] は性能の高くない組込計算機に向けて、制御タスクをコンポーネントに分割し共有メモリによるプロセス間通信を利用して高速に実行するためのミドルウェアとして開発されている。

本研究では中央計算機と組込中間層計算機を組み合わせることで中枢全身制御系を構成する。特に低遅延な実時間性が求められるフィードバック動作制御タスクを組込中間層計算機で実行することを検討する。

組込分散ネットワークからセンサデータを取得し、これを処理してアクチュエータ出力指令を与えるまでのソフトウェア遅延を小さく抑えるためには、ハードウェア制御系におけるデータ交換と動作制御系は同期したタスクとして実行される必要がある。非同期マルチプロセス実装では、計算機ソフトウェア上での大きなデータ転送遅延が生じてしまう。そこで、ハードウェア制御系の実装をプラグイン拡張して動作制御タスクを実行可能とする一般化デバイス制御プラットフォームを開発する。中央計算機と組込中間層計算機のそれぞれや、多様な形態のロボットの制御計算機において、共通構造で適用可能なソフトウェアとして実装する。ハードウェア制御系として、既存のロボット制御ソフトウェアプラットフォームのコンポーネントや別プロセスの連携ソフトウェアとして利用可能となる。

第5章で、一般化デバイス制御プラットフォームの実装も交えながら、多階層から構成されるロボット体内制御システムの中で低遅延フィードバック動作制御を実行する手法に関して述べる。

2.8 本研究で提案する体内分散実時間制御システムの設計実現法

2.8.1 本研究の提案手法

本研究では、多形態多自由度ロボットにおいて実時間性および展開実用性の要件のもとで低遅延な全身フィードバック動作制御を合理的に実現する体内分散実時間制御システムの設計実現法を提案する。本研究の提案手法を以下に列挙し、詳細を後述する。

- (A) 子細適応法 (detailed adaptation method)
 - (A1) 機能完全適応装備 (complete adaptive equipment)
 - (A2) 確定応答中継回路 (deterministic packet relay sequencer)
 - (A3) 通信診断効率設計 (transfer diagnosis optimization)
 - (A4) 拡張型中間制御層 (extensible middle-layer controller)
- (B) 統合節減法 (integrative curtailment method)

- (B1) 応需型複合物理層 (on-demand multifunction interface)
- (B2) 多態論理型伝送網 (polymorphic logical network)
- (B3) 稼働組込複製展開 (online firmware deployment)
- (C) 実地進展法 (on-site advance method)
 - (C1) 動的細部信号解析 (dynamic circuitual operation analysis)
 - (C2) 一貫設計展開過程 (on-site robotics development cycle)

2.8.2 子細適応法

低遅延制御を達成する実時間システムの実装評価法として、(A) 子細適応法を提案する。図 2.16 に子細適応法を概観する。多自由度ロボットの体内制御システムに用いられる電子制御系ハードウェアは、アクチュエータ・センサ・電源などに制御回路系を接続することで構成される。制御回路系には主に駆動回路・通信回路・計算回路が搭載される。2.3.1 項や 2.5.3 項でも述べたように、従来はこれらの回路の一部もしくは全部を実装する応用メカトロニクス製品を転用して組み合わせ、その実装仕様に応じたロボットの研究開発が行われてきた。近年では CAN や EtherCAT の通信インタフェースを搭載するサーボモータ制御モジュール製品が広く用いられている。あるいはその一部、CAN や EtherCAT の通信プロトコルを実装したメカトロニクス向け MCU や通信モジュールなども利用される。しかし、駆動回路・通信回路・計算回路の実装仕様はロボットの制御性能を決定付ける重要な要素である。駆動回路・通信回路・計算回路にブラックボックスを設ける従来手法では、ロボットの行動実現に向けて大きな機能実装制約が生じてしまう。ハードウェアが本来提供する性能を、制約無しで最大限に活用する態勢を整えることが、先端半導体技術に追従したロボット技術の加速的な発展には欠かせない。そこで、駆動回路・通信回路・計算回路の実装をブラックボックスとせず、子細な仕様更新を可能とするハードウェア構成法、およびハードウェア性能を活用してロボット体内制御システムに適応する機能実装法を提案する。子細な仕様へのアクセスが可能となることで、内部状態を監視した性能評価に基づく適応的なシステム実装仕様更新が実現される。

- (A1) 機能完全適応装備 応用メカトロニクス製品を転用した従来の制御回路系には大きな機能実装制約が存在し、ロボット研究開発で生じる仕様更新要求に対応するための機能完全性が損なわれていた。従来の制御回路系に存在したブラックボックスを取り除くことで、機能完全性を有しロボット体内制御システムへの適応を可能とした制御回路系ハードウェアを構成する。駆動回路では直接的に出力スイッチングを制御する回路構成とする。通信回路では直接的に低次の物理層を制御することで既存のプロトコルによる制約を生じない回路構成とする。FPGA のプログラマブル

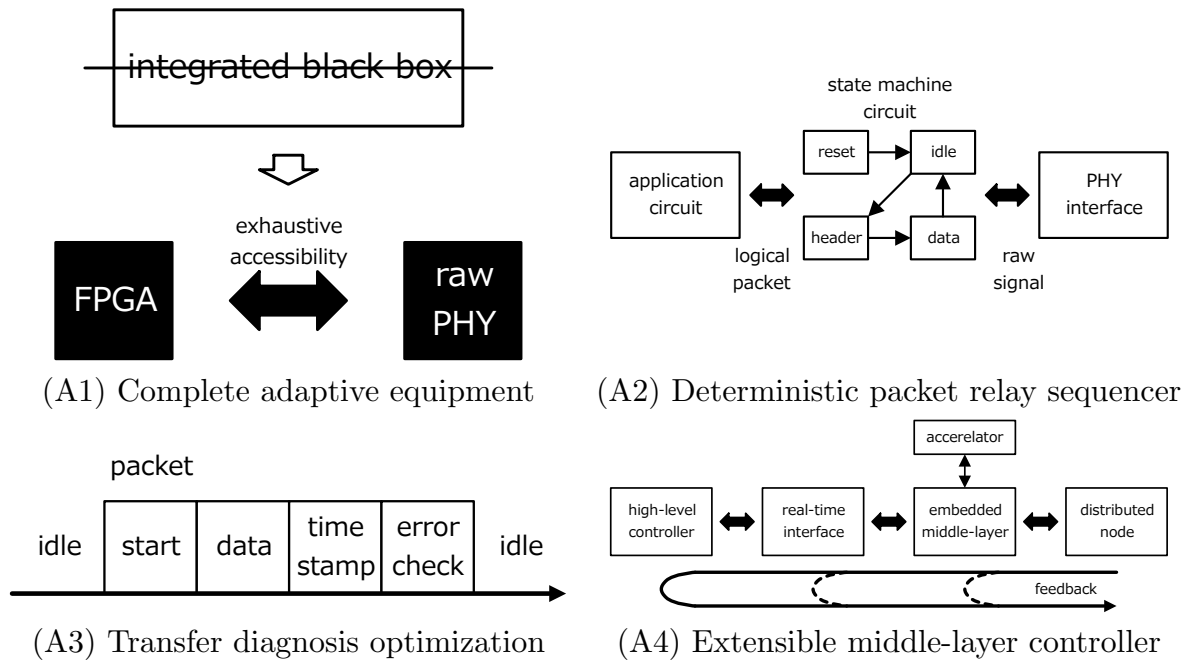


図 2.16 Proposed method: (A) Detailed adaptation method

論理回路を活用して、駆動回路や通信回路のインタフェースを高周波なクロックの分解能で任意タイミングで制御するとともに、ハードプロセッサ・ソフトプロセッサ・ハードウェアアクセラレーションを含む計算回路と任意の構成で接続する。回路を設計する際には、予め実装予定の機能を想定して利用しない信号接続を省略することが一般的であるが、このような制約を設けずに制御可能なデジタル信号を全て FPGA に接続しておくことが子細な内部状態監視や適応的なシステム実装仕様更新に向けて重要である。

- (A2) 確定応答中継回路 フィールドバス規格や製品仕様により制約される従来の通信プロトコル実装では、アプリケーションと接続する大きなオーバーヘッド遅延やジッタが生じ、またプロトコル実装の内部状態が不明であるため正確な詳細評価が行えない。実時間性を満たすためには、通信物理層の受信信号を元にロボット制御アプリケーションの論理データへ低遅延・低ジッタに変換して、計算回路や次の分散ノードへ中継する必要がある。プログラマブル論理回路を用いてステートマシン型のデータ変換回路を構成する。これにより最適に低遅延化して時系列の受信信号に回答し、クロック単位で時間確定的な中継が実現される。データ変換回路の全ての内部状態をイベント信号として監視できるため詳細な遅延性能評価が可能となる。
- (A3) 通信診断効率設計 多自由度ロボットの実時間動作制御では制御周期の中で多数の小さなデータを転送する要求が生じる。小さなデータに対してオーバーヘッドを抑

えた高効率なパケット通信が求められる。一方で、ロボット体内での高信頼通信および通信性能評価を実現するためには、データ区間検出・誤り検出・時間計測の機能が求められる。従来生じていた通信プロトコル仕様の制約を取り除いた実装により、オーバヘッドとのトレードオフから適切なデータ区間検出・誤り検出・時間計測の機能実装を計画する通信プロトコル設計が可能となる。誤り検出には 8b/10b・CRC・Reed-Solomon といった符号化手法を用いる。データにタイムスタンプを付与することで時間を計測し、通信中継ステートマシン回路の実装と合わせた診断機構により、転送時間・転送データ数・エラー発生数の統計を取得する。データ転送レートの間合う限り強力な誤り検出・誤り訂正能力を持つ符号化手法を適用すること、またそのように最適なプロトコル構成を選択可能とする設計が、高信頼な実時間制御を実現する。

- (A4) 拡張型中間制御層 従来は中枢全身制御系においてハードウェア制御と切り離された上位のソフトウェアに全身フィードバック動作制御を一任する構成が主流であった。センサ入力からアクチュエータ出力までのフィードバック経路が多段のデータ転送を含むため大きな遅延が生じてしまう。非同期マルチプロセスソフトウェア構造により数 ms オーダの遅延が生じ、これがダイナミックな環境適応行動実現における重大な支障となる。特に遅延が課題となる動作制御計算はハードウェア制御と同期した処理として実装し、更に通信遅延の小さな組込計算機で実行する。多階層から構成されるフィードバック経路の中間層に拡張展開することで低遅延化が実現される。従来上位のソフトウェアで一貫して実装されていた動作制御を多階層なヘテロジニアス分散制御系に展開するために、計算タスクモデルに基づく制御処理構造の分析とタスクスケジューリングを行う。組込計算機の演算性能を補うために FPGA や GPU によるハードウェアアクセラレーションを導入する。動作制御計算の中から、FPGA 並列反復計算回路や GPU 多並列計算機構と親和性の高い行列計算タスクを抽出することで高速な組込演算を実現する。

2.8.3 統合節減法

多形態多自由度ロボットへの合理的展開法として、(B) 統合節減法を提案する。図 2.17 に統合節減法を概観する。低遅延なロボット体内分散制御システムを多形態なロボットへ迅速に展開し環境適応行動の実現を目指す。従来のロボット制御システムでは、設計時点で特定ハードウェアのロボットを想定して最適化した構成により性能が追求されてきた。制御システムを多形態ロボットへ迅速に展開するためには、共通構造で容易に適用可能なものとする必要がある。しかし、構造共通化のために大きなオーバヘッド遅延が生じ、も

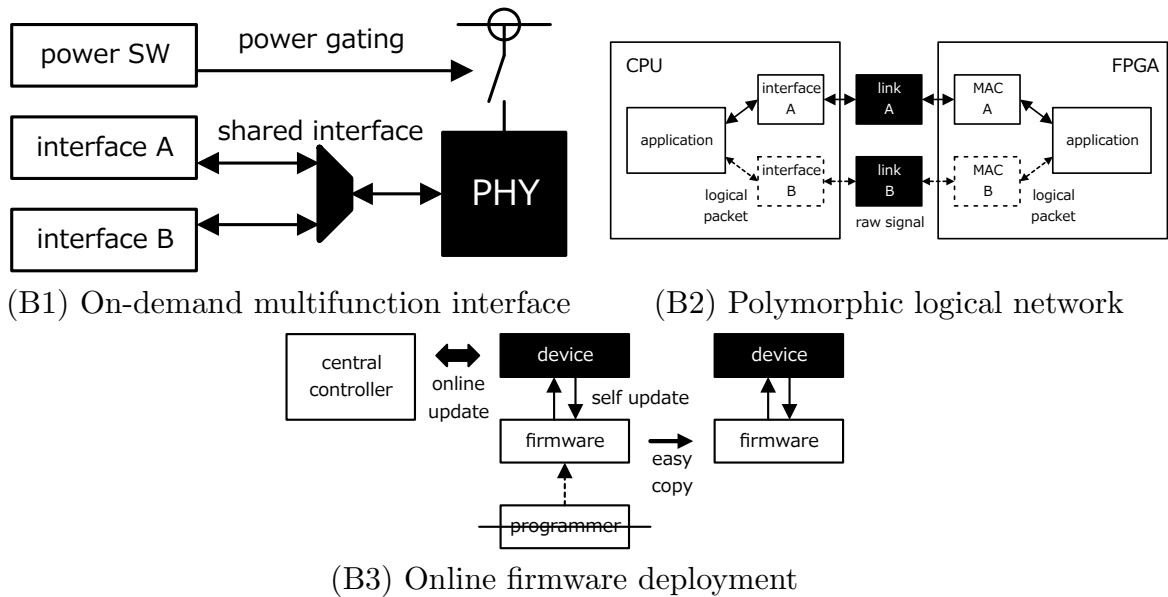


図 2.17 Proposed method: (B) Integrative curtailment method

しくは小型軽量・低消費電力を満たす実用性を損なうようでは本末転倒である。そこで、主要なロボット応用を網羅する構造定義により、小さなオーバーヘッドで性能を維持しながら多形態ロボットへの展開コストを節減する統合的な機能実装法を提案する。

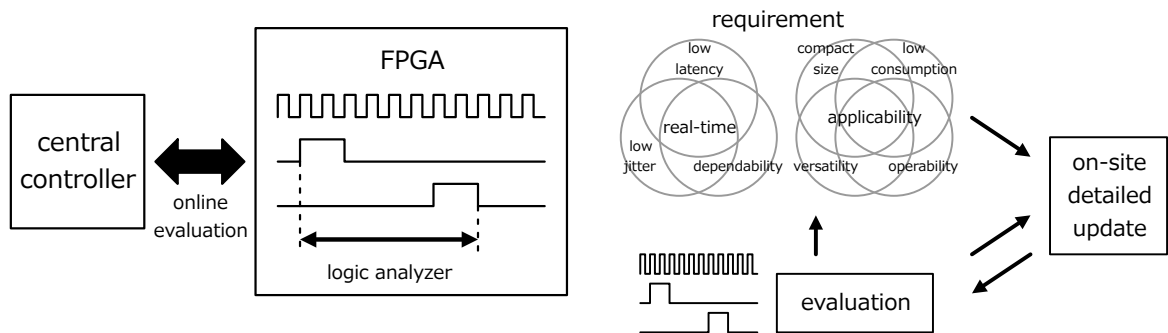
- (B1) 応需型複合物理層 従来の制御回路系モジュール設計は、具体的なアクチュエータ・センサ構成を想定して、専用のインタフェース回路と対応付けて行われてきた。ロボット研究開発における新たな挑戦を支えてゆくためには、新たなアクチュエータ・センサの導入や異なる形態のロボットへの応用に適うモジュール設計が求められる。多様なハードウェア構成に応じて切替可能な複合的実装のインタフェース回路を構成することで、小型・低消費電力ながらも汎用性の高い制御回路系モジュール設計を実現する。汎用シリアル半二重通信に対応する小型汎用物理層回路を FPGA のプログラマブル論理回路に接続することで、任意プロトコルの分散通信や多くのセンサへ対応可能なインタフェースを構成する。FPGA と MCU など複数のコントローラを搭載するモジュールではマルチプレクサでひとつのインタフェースに接続することで回路を共有してモジュールを小型化する。それぞれのインタフェース回路に供給する電源を小型半導体スイッチ素子で遮断可能として、利用しない回路への電源供給を停止するパワーゲーティングにより低消費電力化する。
- (B2) 多態論理型伝送網 中枢全身制御系の実時間ハードウェア制御層では、上位の動作制御ソフトウェアとロボットモデルで記述されたデータを交換して、分散制御ノー

ドと交換する通信データとの間で相互変換する。ロボットモデルデータから特定のハードウェアに対応したデータ構造に直接変換する従来手法では、新たなハードウェア構成のロボットを取り扱う際に実時間制御ソフトウェア全体を作り直す必要が生じる。ロボット体内分散制御システムにおける通信データは、特定アドレスのデバイスと特定種類のデータをパケットで交換する、という構造で共通している。予め多態性を考慮しておき、共通構造を網羅する論理データパケット定義に基づいた通信処理を実装することで、多形態ロボットに展開可能な実時間ハードウェア制御層を小さなオーバヘッドで実装可能となる。同様に分散制御ノードにおいても、FPGAの中継回路で論理データパケット定義に準拠することで、異なる通信インタフェースを搭載するモジュールの間でアクチュエータ制御アプリケーションの共通化が果たされる。

- (B3) 稼働組込複製展開 従来の体内分散制御システム実装では組込モジュールの仕様が頻繁に更新されることはなく、ロボットに搭載した後にファームウェアを更新することは稀であった。ロボット研究開発の要求に応じて体内分散制御システムを適応的に進化させてゆくためには、アクチュエータ制御や通信処理の実装を更新する際のファームウェア展開手順を節減する必要がある。更に組込中間層計算機で全身制御を実行すると、制御データが蓄積されファームウェアに付随する組込データが随時更新される。ファームウェアを他のロボットに複製することで、蓄積されたデータが再利用できる。ロボットの運用中にネットワーク経由で組込モジュールとファームウェアデータを交換し、組込モジュールが稼働中に自らのファームウェアを複製・更新できるシステムを設けることが重要である。

2.8.4 実地進展法

ロボット体内制御システム継続発展に向けた設計実現法として、(C) 実地進展法を提案する。図 2.18 に実地進展法を概観する。メカトロニクス技術はファクトリオートメーションや車載電子制御系などに代表される産業技術実装の中で培われてきた。従来の多自由度ロボットは成熟したメカトロニクス技術を派生させたひとつの応用実装として研究開発が進められてきた。多自由度ロボットを中核産業技術のひとつと位置づけて社会実装を進めるにあたり、既存のメカトロニクス技術の枠組みに制約されないロボット技術発展が期待される。応用メカトロニクス製品を組み合わせるばかりの実装に留まらず、実際にロボットを運用しながらの細部に至る実証評価でロボットならではの要求を抽出して仕様更新を重ねる態勢が求められる。そこで、特にロボット動作制御の性能を決定付ける体内分散ネットワークの低遅延な実時間性を子細に動的評価して組込制御系の実装に反映してゆ



(C1) Dynamic circuital operation analysis (C2) On-site robotics development cycle

図 2.18 Proposed method (C): On-site advance method

く設計実現法を提案する。ロボット研究開発の現場で実際のロボット行動実験から生じる要求に迅速に応えてゆくことで、目指す環境適応行動を実現する。

- (C1) 動的細部信号解析 アクチュエータ制御モジュールやこれに搭載される通信インターフェースは従来、単体での実装評価で仕様が確定し、ロボットに組込まれた後はブラックボックスとして仕様に則って運用されるのみであった。実際に多ノード接続しロボット体内で運用すると、遅延やジッタの累積、電磁ノイズの影響、あるいは実践的な温度特性など、動作中でなければ判明し難い性能の課題も出てくる。運用中に性能を動的解析評価可能な組込制御機能実装を行い、これを用いて実装手法を検証し課題を抽出する。通信プロトコルやアクチュエータ制御器を FPGA のプログラマブル論理回路に実装することでその細部状態がイベント信号として観測可能となる。通信プロトコル実装ではデータ入出力に付随して、タイマ回路・タイムスタンプによる時間計測や誤り検出・誤り訂正の挙動監視を行う。アクチュエータ制御器では内部変数の応答波形を取得する。これをクロック単位で監視しキャプチャするロジックアナライザ回路を併設して、測定結果情報をレジスタやバッファに保持する。測定結果をネットワーク経由で取得し、あるいは組込計算機に統計データとして蓄積することでシステム性能を動的評価する。
- (C2) 一貫設計展開過程 従来のロボット研究開発では、体内制御システムを構成する制御回路系ハードウェアおよび体内分散ネットワークの設計実装が先行し、これと切り離された形で応用的な動作制御ソフトウェア開発や行動実験が行われてきた。高度な環境適応行動を実現してゆくためには、ロボットの運用と実証評価に基づいて更新した要件を、迅速に体内制御システムの設計実装に反映する必要がある。実時間性および展開実用性に基づくロボット体内制御システムの要件定義を行い、これを満たすシステムを設計実装する。プログラマブルな半導体を活用して、ロボット

での運用中にシステムの子細な動的評価を行い実装仕様を修正する．システムを迅速に展開し仕様更新するためにコンポーネント分離と共通インタフェース化を施す．これにより一貫した設計実現で実時間性・展開実用性の目標を達成し，従来なし得なかった加速的なロボット技術発展へと繋げる．

2.9 多形態多自由度ロボットのための体内分散実時間制御システムの構成

提案する体内分散実時間制御システム応用実装の概要を図 2.19 に示す．体内分散実時間制御システムの要素を大別すると，

- 感覚駆動通信系 (sensory-motor network system): 中枢全身制御系と多数の分散制御ノードが接続される低遅延な体内分散ネットワーク
- 組込中間自律系 (middle-layer autonomy system): 中枢全身制御系の低遅延フィードバックを担う組込中間層計算機
- 全身階層協調系 (multi-layer cooperation system): 全身フィードバック動作制御における高速制御計算とデータ転送遅延の低減

から構成され，それぞれ第 3 章・第 4 章・第 5 章で手法提案と技術実装を行う．第 6 章ではこれらの要素技術を多様なロボットへ応用する．

第 3 章に述べる感覚駆動通信系では，ロボットの研究開発の中で生じる要求に柔軟に対応するため，FPGA を活用してハードウェア性能を引き出したネットワーク構成モジュールを小型・高信頼に実装する．PCI Express の普及に伴う近年の半導体技術進展により実用化されてきた高速シリアル通信を活用して，省スペースで高信頼な高速光通信による組込分散ネットワークを構成する．高速光通信や RS-422 汎用シリアル通信のインタフェースに加えてベクトル電流制御やサーボ制御を可能とするモータ制御モジュールを，モータと同程度の幅に収めて小型パッケージ化する．中央計算機と組込分散ネットワークを接続する基幹通信ネットワークとして，現在の高性能計算機製品に搭載される Ethernet や USB のインタフェースを利用して，FPGA と汎用の物理層回路の組み合わせによりインタフェース性能を十分引き出す実装を与える．

第 4 章に述べる組込中間自律系では，中枢全身制御系に FPGA SoC および GPU SoC を導入して応答性の高い全身フィードバック動作制御の基盤を提供する．FPGA や GPU のハードウェアアクセラレータ技術や組込分散ネットワークとのバス接続による低遅延データ交換性能を備えた組込中間層計算機モジュールを開発する．電子回路実装としての特性とロボット部品としての特性を考慮して，組込計算技術および高速ネットワーク技術

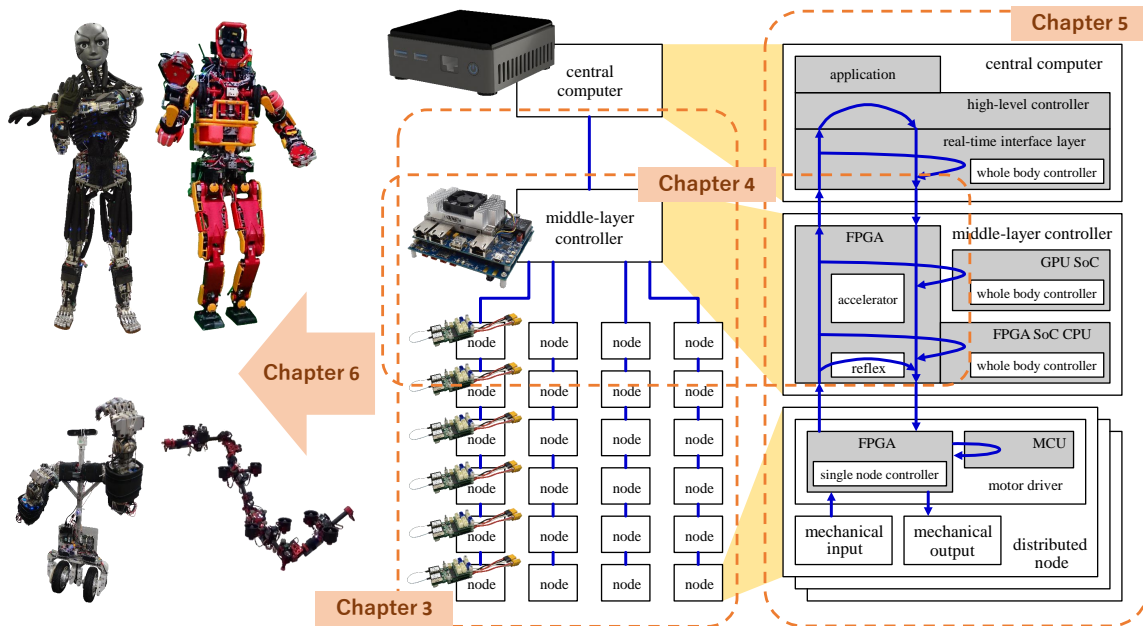


図 2.19 The distributed real-time control system for multi-DOF robot

を小型パッケージで合理的に実用化するための設計手法を明らかとする。

第 5 章に述べる全身階層協調系では、中央計算機・組込中間層計算機・分散制御ノードから構成される多階層な体内分散制御システム全体で低遅延なフィードバック制御を実装する。分散制御ノードでは FPGA および MCU を用いて高速なローカルフィードバック制御を実行する。電流制御や位置制御に加えて、関節摩擦やモータ慣性力を補償したトルク制御を実装する。中枢全身制御系を構成する中央計算機および組込中間層計算機において、多様な形態のロボットに共通構造で実時間ハードウェア制御系を提供する一般化デバイス制御プラットフォームを開発する。一般化デバイス制御プラットフォームを用いて、全身フィードバック動作制御を組込中間層計算機に展開する。

本研究における体内分散実時間制御システムの要素技術実装成果を表 2.6 にまとめる。要素技術の応用方針とその効果を図 2.20 に示す。ギア駆動型大出力ヒューマノイド・腱駆動型筋骨格ヒューマノイド・多リンク飛行ロボットをはじめとした多様な形態のロボットに、合理的な実時間制御システムを展開する。それぞれのロボットに向けた応用戦略とその実装評価を第 6 章に述べる。本研究の提案手法および提案システムを図 2.21 に総括する。

表 2.6 Design and development deliverables in this work

紹介	抽象名 型式名	概要
3.5.2 項	小型モータ制御基板 FPGA-03D	狭幅高密度モータドライバ ベクトル電流制御とサーボ制御 汎用シリアル通信
3.5.2 項	小型光通信モータ制御モジュール SU-05-OPT	狭幅高密度モータドライバ ベクトル電流制御とサーボ制御 高速光通信と汎用シリアル通信
3.7.1 項	EtherCAT 中継モジュール ECB-02	既製品 EtherCAT モジュール搭載 汎用シリアル通信 中間的速度の光通信
3.7.3 項	小型 Ethernet 中継基板 ENB-04	Gigabit Ethernet 完全準拠 PHY 汎用シリアル通信
3.7.3 項	小型 Ethernet 光通信中継基板 ENB-05	Gigabit Ethernet 完全準拠 PHY 高速光通信と汎用シリアル通信
3.8 節	USB High-Speed 中間制御基板 FPGA01C	USB 2.0 完全準拠 PHY 通信とサーボ制御計算
3.8 節	USB 2.0 ハブ基板 HSHUB	小型 USB 2.0 ハブ 大容量電源レギュレータ
4.3.1 項	組込中間層計算機プロトタイプ OPTFIBER ADDON	既製品 FPGA SoC 基板に接続 既製品 EtherCAT モジュール搭載 通信インタフェースの開発用
4.3.2 項	組込中間層計算機モジュール ENBSoC-03	FPGA SoC 既製品 GPU SoC モジュール搭載 Gigabit Ethernet 完全準拠 PHY 高速光通信と汎用シリアル通信
5.5.3 項	一般化デバイス制御プラットフォーム code name: neuro_vm	C++ オブジェクト指向 ハードウェアパケット通信 API 実時間ロボット制御 API 同期・非同期プラグイン拡張

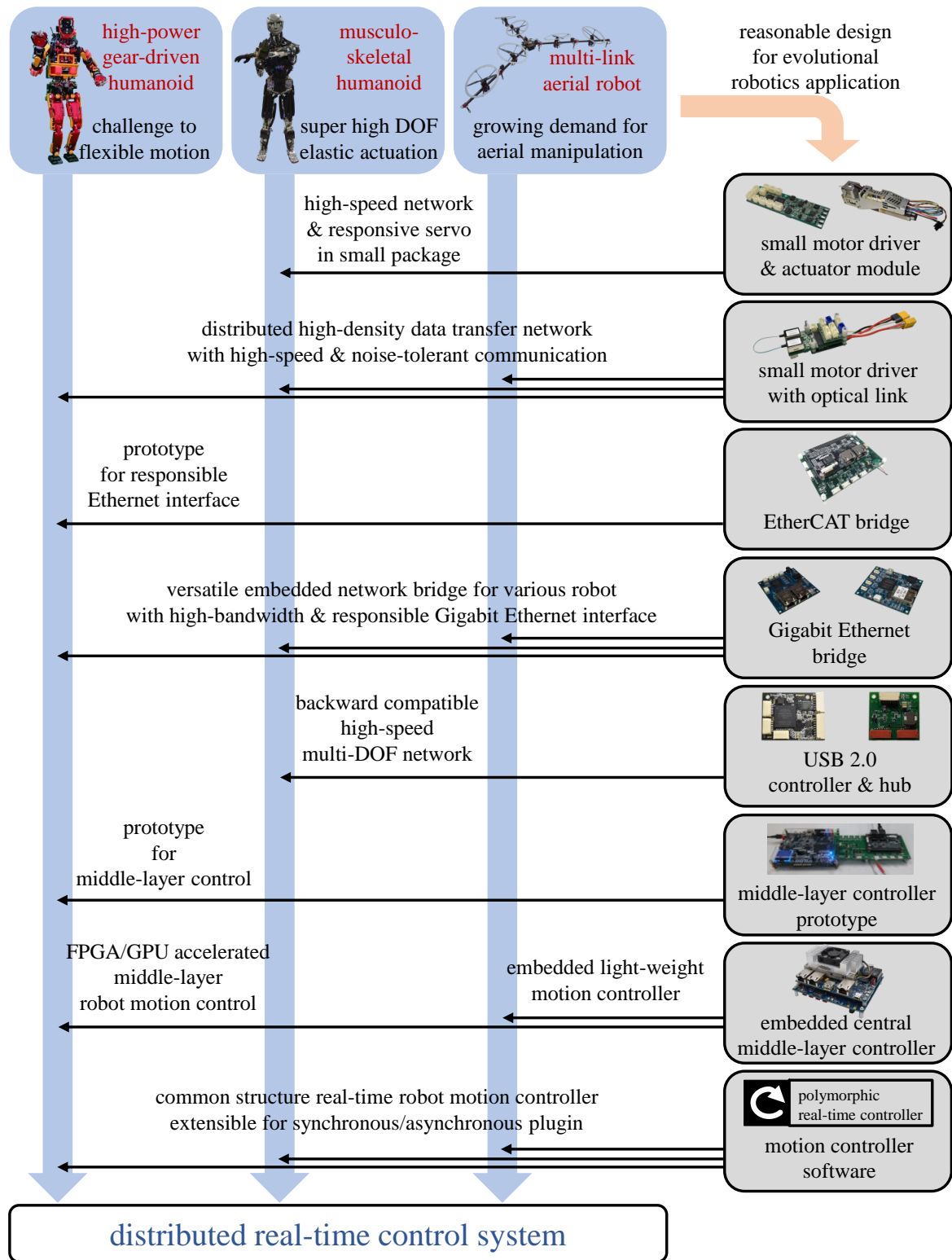


図 2.20 Application and implementation of the distributed real-time control system for multi-DOF robot

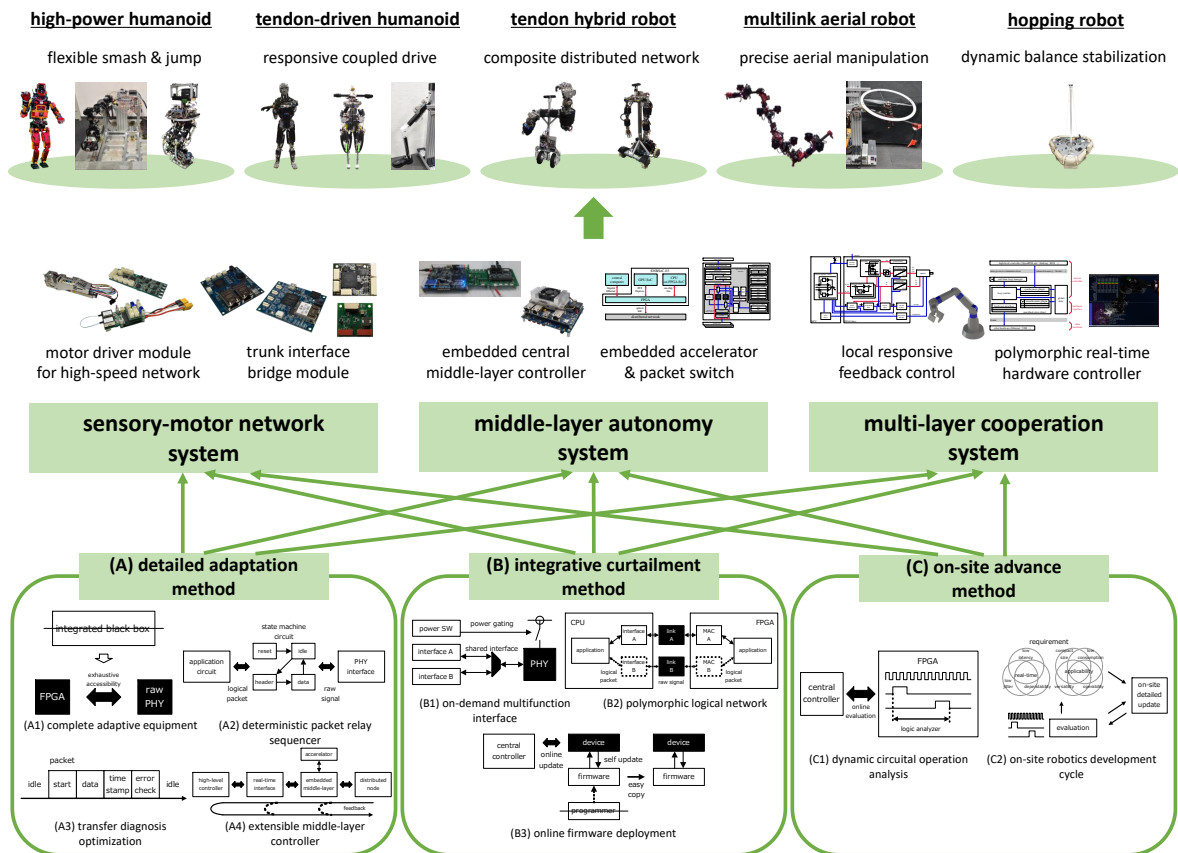


図 2.21 Summary of proposed system: The distributed real-time control system for multi-DOF robot with the design and fruition method

2.10 本章のまとめ

本章では、多自由度ロボットの体内分散実時間制御システムの要件と、これを満たす設計実現法に関して述べた。

本研究の対象とする多自由度ロボットとして、ヒューマノイド・筋骨格ロボット・フィールドロボットを挙げ、数十、あるいは多いもので 124 にも及ぶアクチュエータ自由度例を紹介した。ロボットの行動における環境接触を分析し、衝撃的な環境接触を伴う打撃作業や跳躍着地が挑戦的な課題であることを述べた。

ヒューマノイドの体内制御システムに見られる技術的実装の推移を紹介した。リモートブレイン方式や集中制御型の構成から、CAN などの汎用シリアル通信を用いた分散制御化が進められ、近年では EtherCAT などの産業用 Ethernet フィールドバスが採用され始めている。体内分散制御システムを構成する分散制御ノードおよび中枢全身制御系に関し

て述べた．分散制御ノードでは電流制御や位置制御などに基づいてアクチュエータを駆動する．中枢全身制御系では認識・計画・実行に基づいた行動制御システムが実装され，特に実行系では高い実時間性と時間分解能が求められる．

ロボットの力フィードバック制御に大きな遅延が生じると，安定性や衝撃的外乱への応答に問題が生じることを述べた．多様な形態の多自由度ロボットにおいてフィードバック遅延の課題を解決する体内制御システムの要件として，低遅延・低ジッタ・信頼性に基づく実時間性と，小型軽量・低消費電力・汎用性・運用性に基づく展開実用性を挙げた．

ロボット体内分散ネットワークの設計に向けて，産業用フィールドバスおよび計算機通信インタフェースの背景技術を分析した．ロボットにおいても，CAN や EtherCAT などの産業用フィールドバス技術，および PCI や Ethernet などの通信インタフェースが利用されてきたことを述べた．産業分野で成熟した仕様実装のモジュール製品を組み合わせる体内制御システム構成手法では，ロボットの行動実現への挑戦を支えるために大きな制約が生じることを指摘した．最新の技術に追従しロボット研究開発の現場で柔軟に実装仕様を更新する体内制御システムの継続的な開発が重要であることを述べた．

組込中間層計算機の導入に向けて，近年の組込計算機の高性能化やハードウェアアクセラレーション技術の進展を紹介した．FPGA を活用した分散ネットワークと相性の良い組込計算機で低遅延なフィードバック制御が達成されることを述べた．

多階層なロボット体内分散制御システムにおいて全身動作制御を分散実行する手法を検討した．ヘテロジニアス並列計算環境における計算時間および通信時間のモデルを紹介した．ヒトの神経系に見られる複合的で協調的な構成とロボット制御システムを対比した．ロボット制御タスクを分散実行するためのプラットフォームを紹介し，動作制御に拡張可能な実時間ハードウェア制御系の実装目標を述べた．

これらの背景や要件に基づき，多形態多自由度ロボットにおける体内分散実時間制御システムの設計実現法として，子細適応法・統合節減法・実地進展法を提案した．提案手法に基づき，感覚駆動通信系・組込中間自律系・全身階層協調系による体内分散実時間制御システムの構成を述べた．

第 3 章

体内分散実時間ネットワークを形成する 小型高信頼な感覚駆動通信系

3.1 はじめに

多自由度ロボットの体内分散ネットワークではそのアクチュエータ数におよそ比例した大量のデータ転送が行われる。全身フィードバック制御の低遅延化は、実時間制御周期の中で必要なデータを十分低遅延に転送することのできる分散ネットワーク通信技術に支えられてこそ初めて達成される。また、分散ネットワークを構成する電子制御系をロボットに実装する上ではハードウェアの小型化や機能動作の高信頼化も欠かせない。低遅延なデータ転送ネットワークを小型・高信頼な電子制御系ハードウェアで実現することにより、多様なロボットへの技術波及が見込まれる。

本章では、体内分散ネットワークの要求性能に対して従来技術が抱えていた課題を考察し、これを解決する電子制御系ハードウェア実装手法を提案する。ロボット体内分散ネットワークは、分散制御ノード同士を接続する組込分散ネットワークと、中枢全身制御系において中央計算機を接続する基幹通信ネットワークにより実装される。

従来より多くのロボットの組込分散ネットワークに利用されてきた汎用シリアル通信は、大容量化するデータ転送要求に対する律速となってきた。特に多自由度なロボットにおいて高密度に配置されるアクチュエータ制御系では、そのネットワーク規模から本来殊更に大容量なデータ転送が求められるにもかかわらず、そのスペース制約によって、アクチュエータ制御性能とともに通信性能も厳しく制限されていた。本研究では近年益々進展する半導体技術を活用し、小型なモータ制御モジュールの制御性能および通信性能の向上を図る。更に従来の汎用シリアル通信の性能を大きく凌駕する高速光通信技術により、小型なモータ制御モジュールによる非常に低遅延で高信頼な体内分散ネットワークを実現する。

時を重ねるごとに益々高性能化が進む計算機製品には、その時代の技術に応じた通信インタフェースが搭載される。基幹通信ネットワークの設計においては、全身制御に利用する中央計算機の実装仕様による先行的な制約が生じる。中央計算機を十分に活用した高度なロボット制御の実現に向けて、通信インタフェースの制約に対して柔軟に応じ、性能を余すことなく引き出す実装が求められる。本研究では現在の計算機に搭載される Ethernet や USB のインタフェースに基づいて、FPGA を用いて通信性能を引き出す実装例を交えて、基幹通信ネットワークの小型高信頼設計手法を明らかとする。

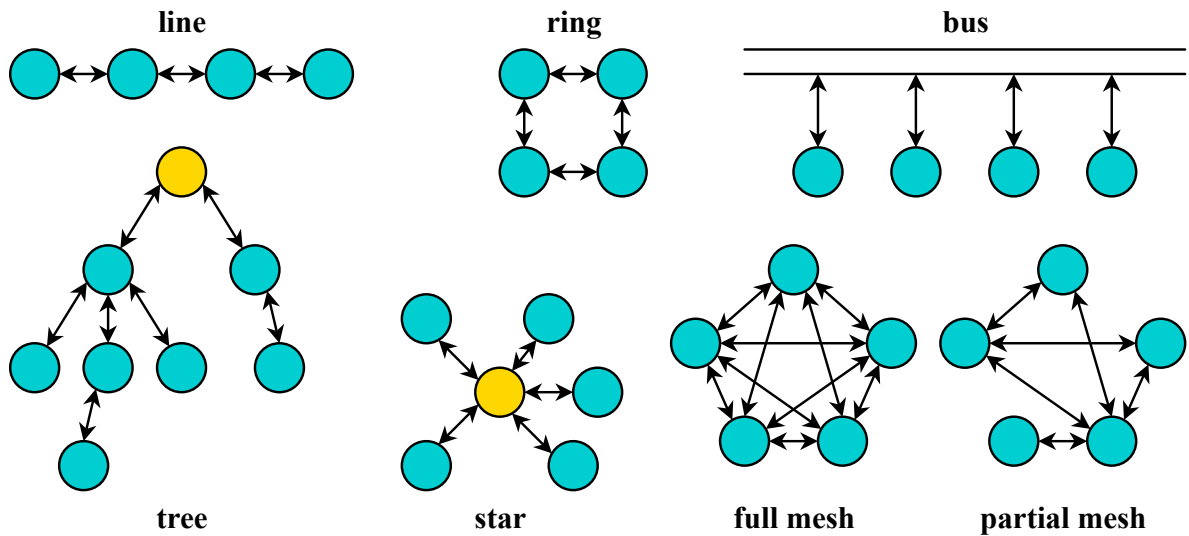


図 3.1 Typical network topology

3.2 多自由度ロボットの体内分散ネットワークとデータ転送性能

3.2.1 体内分散ネットワークトポロジ

ネットワークにおける接続構造は一般に図 3.1 のようなトポロジに分類して考察される。

- ライン型：始点ノードから終点ノードまでストレートに数珠繋ぎ
- リング型：始点ノードから数珠繋ぎで再び始点ノードまで戻る
- バス型：全てのノードでひとつの伝送経路を共有
- ツリー型：親ノードからの枝分かれを再帰的に繰り返す構造
- スター型：全ての子ノードがひとつの親ノードに接続
- 完全メッシュ型：全ての2ノード間で直接接続
- 部分メッシュ型：任意のノードの組み合わせで直接接続

実際には複数のサブネットワーク同士が接続されて複合ネットワークが構成される場合も多い。物理的な信号接続に基づく物理トポロジの他に、より抽象度の高い機能的接続に基づく論理トポロジを考えることもできる。

例えば分散制御ノードがそれぞれ2ポートの通信インタフェースを搭載し、中枢全身動

作制御系からデジチェーン接続される構成は、ライン型の物理トポロジに分類される。これに対する論理トポロジとしては、

- 物理トポロジと同じライン型
- 中枢全身動作制御系から送信されたパケットがネットワークを循環してそれぞれの分散制御ノードで 1 度だけ処理されてから中枢全身動作制御系に戻ってくるためリング型
- 中枢全身動作制御系に対して全ての分散制御ノードが子ノードとして対応するためスター型

など、機能的な概念次第でいくつかの捉え方ができる。中枢全身動作制御系から複数チャンネルでデジチェーン接続される場合は、複数のライン型ネットワークがスター型で接続された複合ネットワークとして考えることができる。

データ転送遅延に関して議論する際にはネットワークの物理トポロジが重要な要素となる。多自由度ロボットにおいてはリンク機構の関節部分にアクチュエータが分散配置され、ルートリンクから見るとアクチュエータが遠方まで直列に並ぶ場合が多い。そのため、多自由度ロボットの体内分散ネットワークは、配線ケーブルの長さや並列数を抑えることのできるライン型やツリー型の物理トポロジと相性がよい。例えば、体幹を中心に脊椎や四肢のそれぞれを成す垂直多関節マニピュレータが接続されるヒューマノイドでは、中枢全身制御系に対してマニピュレータごとのライン型ネットワークを並列接続することで効率的な配線が可能となる。

図 3.2 に体内分散ネットワークの構成例を示す。ヒューマノイドの脊椎・四肢構造を模してそれぞれのネットワークトポロジを比較している。

図 3.2 (a) はスター型で、中枢ノードに対して全ての分散ノードが直に接続される構成である。全身の分散ノードに対して低遅延に直接アクセスできる一方で配線が煩雑に多数並行するという特徴を有する。分散制御系を構成せずに中枢ノードで集中的に全身のアクチュエータやセンサを制御する構成も、スター型と同様の形式で中枢ノードから全身への直接並列接続で模式化することができる。図 3.2 (b) はバス型で、全ノードが同一の信号線を共有する構成である。スター型よりも少ない配線で全身の分散ノードに直接アクセスできるが、ノード数が増えても同一の信号線を共有した全体の最大データ転送レートが一定であるため、多ノード化に向けては不利である。バス型の通信プロトコルのひとつである CAN の利用例が多く見られる。実用的には図 3.2 (f) のように全身をいくつかのドメインに分割してそれぞれでバス型ネットワークを構成する場合も多い。図 3.2 (c) はライン型で、中枢ノードに対して分散ノードがデジチェーン接続される構成である。省配線で高速通信プロトコルを実装することができる一方で、接続ノード数とともに通信遅延も増大する。EtherCAT などの産業用 Ethernet 規格によるデジチェーン接続実装例が多

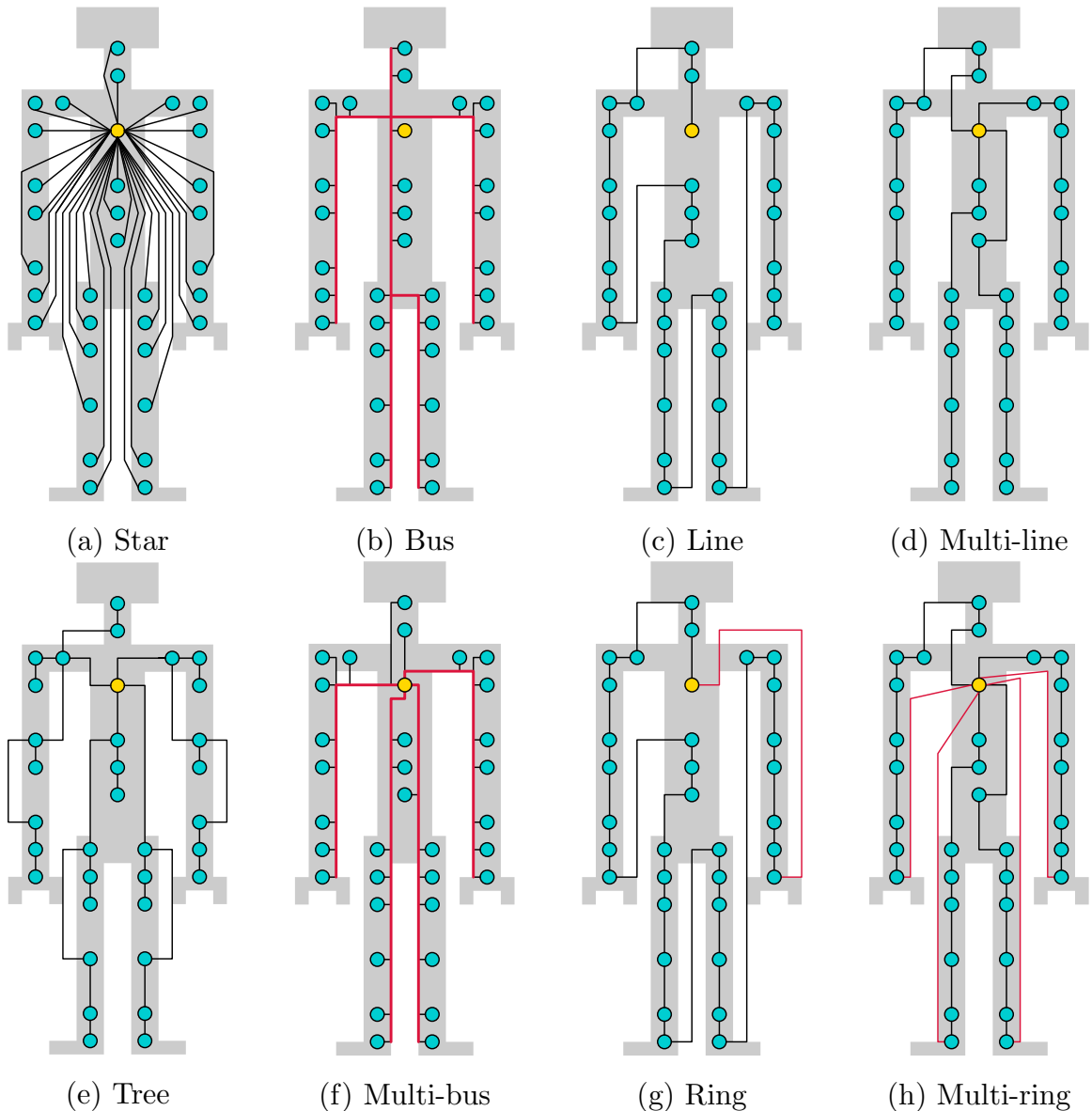


図 3.2 Network topology for humanoid robot

く見られる．中枢ノードが多チャンネルの並列通信に対応する場合は図 3.2 (d) のように複数のライン型ネットワークを並列接続することができる．これによりひとつのチャンネルに接続されるノード数が減少し通信遅延が低減されるとともに，さらなる省配線化が果たされる．ライン型の末端ノードから中枢ノードへの接続を加えることで図 3.2 (g) や図 3.2 (h) のようなリング型の構成へ拡張することもできる．ライン型のネットワークを往復させる必要がなくなり通信遅延が低減するとともに，接続を冗長化させることで部分切断対応への応用も可能となる．図 3.2 (e) はツリー型で，分散ノードにおける分岐に対応させた構成である．それぞれのノードに 2 チャンネルを超える通信インターフェースを実装する必

要があるが，機械構造に応じて更に自由な配線が可能となる．

本研究ではFPGAを用いた自在な通信プロトコル実装により，図3.2(d)のライン型多チャンネル並列接続構造を基本とした低遅延省配線ネットワークを構成する．

3.2.2 体内分散ネットワークにおけるデータ転送遅延

体内分散ネットワークに採用されるライン型やツリー型のパケット循環ネットワークにおけるデータ転送遅延をモデル化する．

ネットワーク上のノード間データ転送において，途中で中継するノードの数はホップ数と呼ばれる．始点ノードから終点ノードまで直にデータ転送する場合を0 hopとして数える．

ここでは中枢ノードから送信したパケットをツリー型ネットワーク内で循環させて再び中枢ノードまで到達させるパケット循環ネットワークを考える．ライン型はツリー型に含まれる特殊例であるため，ツリー型で一般化できる．ツリー型の物理トポロジに対して，パケット循環ネットワークの論理トポロジはリング型となり，ホップ数はリング型の中継ノード数で計上する．中枢ノードに対して分散ノードを1 nodeだけ接続した構成は，途中で分散ノードによる中継を1度だけ経由する1 hopの経路となる．以降，ツリー型によって分散ノードを1 node追加するごとに2 hopずつ増加してゆく．中枢ノードに対して分散ノードを N_n node接続した場合の通信経路は $(2N_n - 1)$ hopとなる．なお1ノードあたり1度だけパケットに対する処理を実行する場合， $(2N_n - 1)$ hopのうち N_n hopはパケットの処理を伴い，残りの $(N_n - 1)$ hopは分散ノードの重複によるものでパケットが中継されるのみである．中枢ノードに実装されるアプリケーションからみた遅延を考える際には，中枢ノードにおける入出力遅延も等価的に1 hop分として換算すると有用な近似となる場合がある．

分散ノードではストアアンドフォワード方式に基づき，データの最小単位である1パケットを完全に受信および処理してから次のノードへ送信するものとする．パケットサイズを D_p とする．通信プロトコルにおけるデータ転送レートを R_d ，1 hopあたりの平均パケット処理時間を T_h とにおいて， N_p 個のパケットを転送するのにかかる総合データ転送遅延 T_d を求める．

転送する全データサイズは $N_p D_p$ である．サイズ $N_p D_p$ のデータを速度 R_d の1経路で転送するには時間 $\frac{N_p D_p}{R_d}$ がかかる．ストアアンドフォワード方式のため1 hopあたり $\frac{D_p}{R_d}$ の転送時間が追加される．更に1 hopあたり T_h の処理時間が追加され，これには中枢ノードにおける入出力も1 hopとして加算する．すなわち総合データ転送遅延は，

$$T_d = \frac{N_p D_p + (2N_n - 1) D_p}{R_d} + 2N_n T_h \quad (3.1)$$

と求められる。

式 3.1 から逆に，サイズ D_p のパケットを N_n node のネットワークで循環させる構成において，データレート R_d およびパケット中継処理時間 T_h が同定されていれば，許容されるデータ転送遅延 $T_d \leq T_d^{\max}$ から，転送できるパケット数の上限，

$$N_p \leq \frac{R_d}{D_p} (T_d^{\max} - 2N_n T_h) - (2N_n - 1) \quad (3.2)$$

が決定される。

式 3.1 および式 3.2 を用いることで，体内分散ネットワークにおけるデータ転送性能・許容通信遅延・転送データ量の関係について，ホップ数の影響も加味して定量評価することができる。

3.2.3 体内分散ネットワークに求められるデータ転送性能

図 2.4 のように中枢全身制御系に対して分散制御ノードをアクチュエータの数だけ接続した，多自由度ロボットの体内分散ネットワークを考える。中枢全身制御系はその実時間制御周期ごとに，分散制御ノードとの間で必要なデータを交換し全身動作制御を実施する。体内分散ネットワークの実装は，分散制御ノード同士を接続する組込分散ネットワークと，中央計算機と組込分散ネットワークを接続する基幹通信インタフェースから構成される。組込分散ネットワークと基幹通信インタフェースの両者が，求められる転送データ量および許容遅延を十分満たす必要がある。

ここでは図 2.1 (a) で紹介したヒューマノイド JAXON を具体例に取り挙げ，その体内分散ネットワークにおけるデータ転送の規模に関して考察する。JAXON はおよそ 40 に近い典型的なアクチュエータ自由度を有するヒューマノイドの例であると言える。分散制御ノードには，図 2.6 (c) の位置制御器，あるいは図 2.6 (d) 形式の派生で損失トルク補償に基づく関節トルク制御器 [143] が実装される。

運用上想定される通信データ項目を集計して要求仕様の試算に利用する。中枢全身制御系が分散制御ノードから収集するデータには，全身制御にフィードバックするものと状態監視用のものがあり，必要な収集頻度はデータごとの性質による。逆に中枢全身制御系から分散制御ノードに送信するデータは，制御入力指令と制御パラメータに分類できる。前者の制御入力指令はロボットの動作に伴い毎周期更新する必要がある一方で，後者の制御パラメータは値の変更が稀であり，変更時以外には時折送信することで確実な値の同期を取る程度で十分である。

ここで具体的な通信データ項目を列挙する。モータ電流 PI 制御器に関連して表 3.1 のデータを交換する。モータ位置 PD 制御器に関連して表 3.2 のデータを交換する。関節トルク制御器に関連して表 3.3 のデータを交換する。その他，制御系の管理および状態監視

表 3.1 Communication data item list for current control

data	Read/Write	mandatory every cycle
motor d-axis current	R	-
motor q-axis current	R	-
motor q-axis current reference	R	-
motor parameter by PN	W	-
current control mode	W	-
motor current limit	W	-
current P-gain	W	-
motor d-axis current reference	W	-

表 3.2 Communication data item list for position control

data	Read/Write	mandatory every cycle
motor position	R	*
motor velocity	R	*
motor position reference readback	R	-
motor position error	R	-
motor velocity reference readback	R	-
motor position reference	W	*
motor velocity reference	W	*
position P-gain	W	-
position D-gain	W	-
motor position error limit	W	-

のために表 3.4 のデータを交換する．制御モードの切替時には必要に応じて表 3.5 のような臨時のリクエストコマンドを送信する．一部の分散アクチュエータ制御ノードはエンドエフェクタに搭載された力センサの読み出しおよびデータ送信の役割も兼ねており表 3.6 のデータ項目が追加される．JAXON の場合は力センサデータを同じ体内分散ネットワークに載せているが，これに限らず姿勢センサなど他のデバイスが接続されたノードが存在する場合も同様である．以上を集計すると表 3.7 に示すデータ項目数となる．

臨時コマンドを含めた最大値で数えると，交換したいデータとしてモータあたり 39 項目および力センサあたり 6 項目が存在している．JAXON は 39 個のモータと 4 個の力セ

表 3.3 Communication data item list for torque control

data	Read/Write	mandatory every cycle
joint torque	R	*
joint torque reference	W	*
torque output gain	W	-
torque compensation gain	W	-
motor torque constant	W	-
motor inertia	W	-
joint coulomb friction	W	-
joint viscous friction	W	-

表 3.4 Communication data item list for manager and monitor

data	Read/Write	mandatory every cycle
error status	R	*
joint position	R	*
motor power voltage	R	-
controller power voltage	R	-
motor temperature	R	-
motor hall sensor	R	-
communication error CH0	R	-
communication error CH1	R	-
control mode	W	-
motor temperature limit	W	-

ンサを搭載しているため、合計 $39 \times 39 + 6 \times 4 = 1545$ 項目となる。

1 項目のデータは、3.3.1 項で紹介する図 3.7 の 32 bit 統一フォーマットの packets で交換される。そのため実時間制御周期あたりに求められる通信データ量は $1545 \times 32 = 49440$ bit となる。8 bit を 1 Byte と換算して、6180 Byte 分に相当する。

近年の多くのヒューマノイドでは動的な動作実現のためにおよそ 1 kHz の制御出力更新頻度、すなわち 1 ms の制御周期を目標に掲げており [144][145][146][15]、これに倣って 1 ms の制御周期を想定する。1 周期あたりのデータ量が 49440 bit であるため、1 ms の制御周期での転送レートは 49.44 Mbps となる。この数値が、体内分散ネットワークに

表 3.5 Communication data item list for temporary request

data	Read/Write	mandatory every cycle
control mode change	W	temporary
error status reset	W	temporary
emergency stop	W	temporary

表 3.6 Communication data item list for force sensor

data	Read/Write	mandatory every cycle
x-axis force	R	*
y-axis force	R	*
z-axis force	R	*
x-axis moment	R	*
y-axis moment	R	*
z-axis moment	R	*

表 3.7 Communication data item count

target	Read/Write	mandatory every cycle	item number
every node	R	*	5
every node	R	-	12
every node	W	*	3
every node	W	-	16
every node	W	temporary	max. 3
node with force sensor	R	*	6

おける目標データ転送レートに関して具体例を交えたひとつの指標となる。

実装上の制約により制御周期内に所望のデータを全て交換しきれない場合、何らかの妥協を伴いデータ量を減らす工夫が必要となる。JAXON の例では毎周期交換することが強く求められるデータは一部であり、その他のデータに関しては交換頻度を低減させることもできる。表 3.7 で毎周期は求められないとされている 28 項目を N 周期に 1 度の頻度に分散させると、1 周期あたり $((11 + 28 \div N) \times 39 + 6 \times 4)$ 項目に抑えられる。例えば 3.7.1 項に述べる EtherCAT インタフェースの実装では 1 周期あたりの通信データ量をおよそ 2048 Byte = 16384 bit に制約している。この場合

$((11 + 28 \div N) \times 39 + 6 \times 4) \times 32 \leq 16384$ より, $N \geq 18.508$ の周期に分散させれば十分運用可能となる. 実際には臨時コマンドの発生パターンを考慮すると制約が緩和される場合もある. 臨時コマンドの3項目を完全に除いた下限は $N \geq 6.205$ である. EtherCAT インタフェースを JAXON に適用しデータ転送を実装するには, 十分値で 19 以上, あるいはパターン次第の下限で 7 以上の周期に低頻度データを分散させる運用が求められる.

3.3 低遅延化に向けた従来の組込分散制御系の課題

3.3.1 ギア駆動型大出力ヒューマノイドの従来実装

ギア駆動型大出力ヒューマノイドに関する先行研究 [147][148][20] では, FPGA による制御および厚銅基板や液冷による排熱技術を利用して大出力を実現したモータ制御モジュールが開発され, モータ制御モジュールによるデジチェーン接続の体内分散ネットワークが PCI 中継基板を介して中枢制御を担う中央計算機と接続されていた.

図 3.3 に大出力モータ制御モジュール H8-17 の概観を示す. 大出力モータ制御モジュールは主に制御基板とモータドライバ基板の2枚から構成される. ブラシレスモータと図 3.4 のように接続してモータ出力制御を実行する. 過去の実験 [147] では最大 159 A に上るブラシレスモータ q 軸電流の出力に成功したと報告されている.

モータドライバ基板では, 最大銅箔厚 500 μm の厚銅基板上に, MOSFET による3相ハーフブリッジドライバおよびバイポーラトランジスタによるプッシュプルゲート駆動回路が実装されている. MOSFET は液冷ヒートシンクに接続され, 高周波スイッチングで生じる熱を効率的に排出可能となっている.

制御基板上の主要なコントローラとして Intel 社の Cyclone V E FPGA 5CEBA4F17C8N が搭載されている. FPGA には,

- ブラシレスモータのベクトル電流 PI 制御器
- RS-422 分散ネットワークにおけるパケット通信およびデータ処理
- Nios II ソフトコアプロセッサによる位置 PD 制御をはじめとしたサーボ制御演算
- その他周辺回路制御

といった機能が実装される. FPGA の他にルネサスエレクトロニクス社の 16 bit MCU H8S/2215R が搭載されているが, これは旧実装 [147] との互換性を目的としたものであり, 近年では FPGA のプログラマブル論理回路上に構成された Nios II ソフトコアプロセッサに置き換えられ, 利用されていない.

FPGA は 24 MHz の基本クロックをベースに駆動される. モータ出力回路では, 78

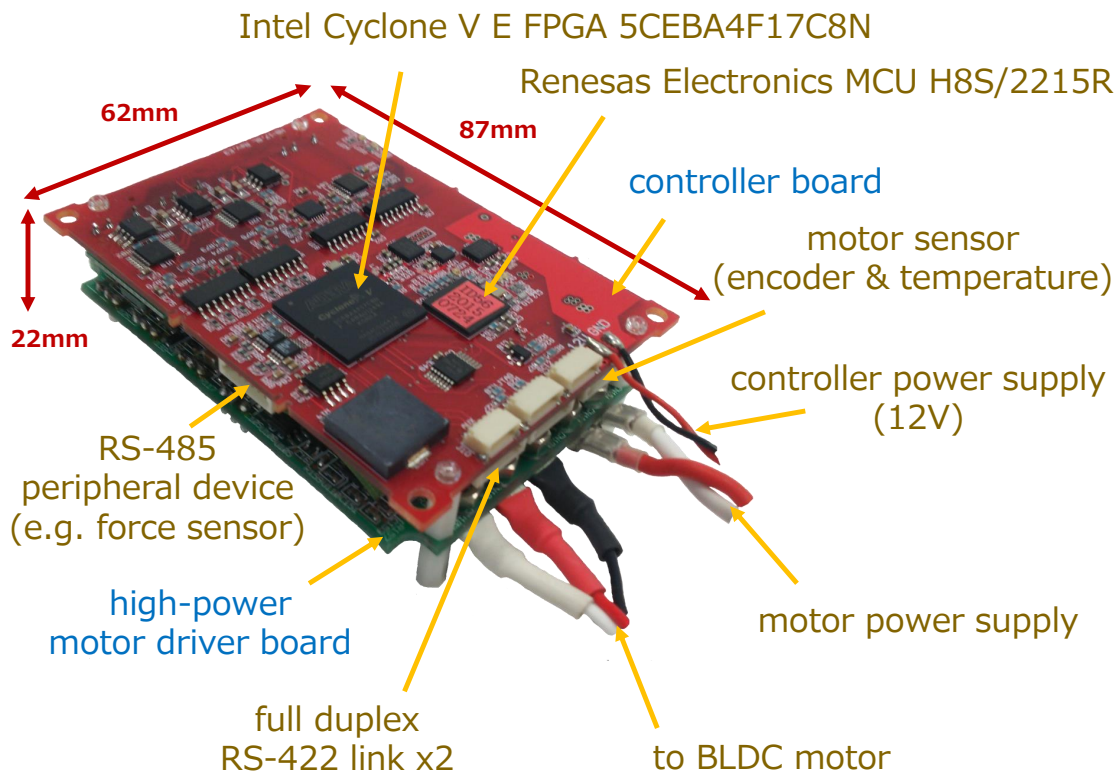


図 3.3 Overview of the high-power motor driver module H8-17 in previous work

kHz を基本設定とした PWM 出力によって MOSFET ハーフブリッジドライバのスイッチングを行う。スイッチング周波数を高く設定するとモータの出力応答性が向上しリップル電流による損失も減少するが、一方で MOSFET のスイッチングによる損失が増大するため、このトレードオフにより適切な値が決定される。電流 PI 制御器では、10 MHz の $\Delta\Sigma$ 変調による電流測定値の更新頻度を基調として構成されたデジタルフィルタ回路の応答性に従ってフィードバックが行われる。位置 PD 制御をはじめとしたサーボ制御はプロセッサにより 1 kHz で実行される。

モータドライバ基板による分散ネットワークと中央計算機は図 3.5 の PCI 中継基板を介して接続される。PCI 中継基板には旧 Altera 社の Cyclone II FPGA EP2C20F256C6 および旧アドテックシステムサイエンス社の PCI インタフェースコントローラ APIC22R10 が搭載され、PCI と RS-422 信号を利用した独自分散ネットワークを中継する。PCI インタフェースコントローラでは、FIFO バッファを介して中央計算機側に接続された 32 bit バスと FPGA 側に接続された汎用 32 bit バスが相互変換されている。PCI インタフェースコントローラによるバスの転送レートはスペック最大値で 1056 Mbps とされて

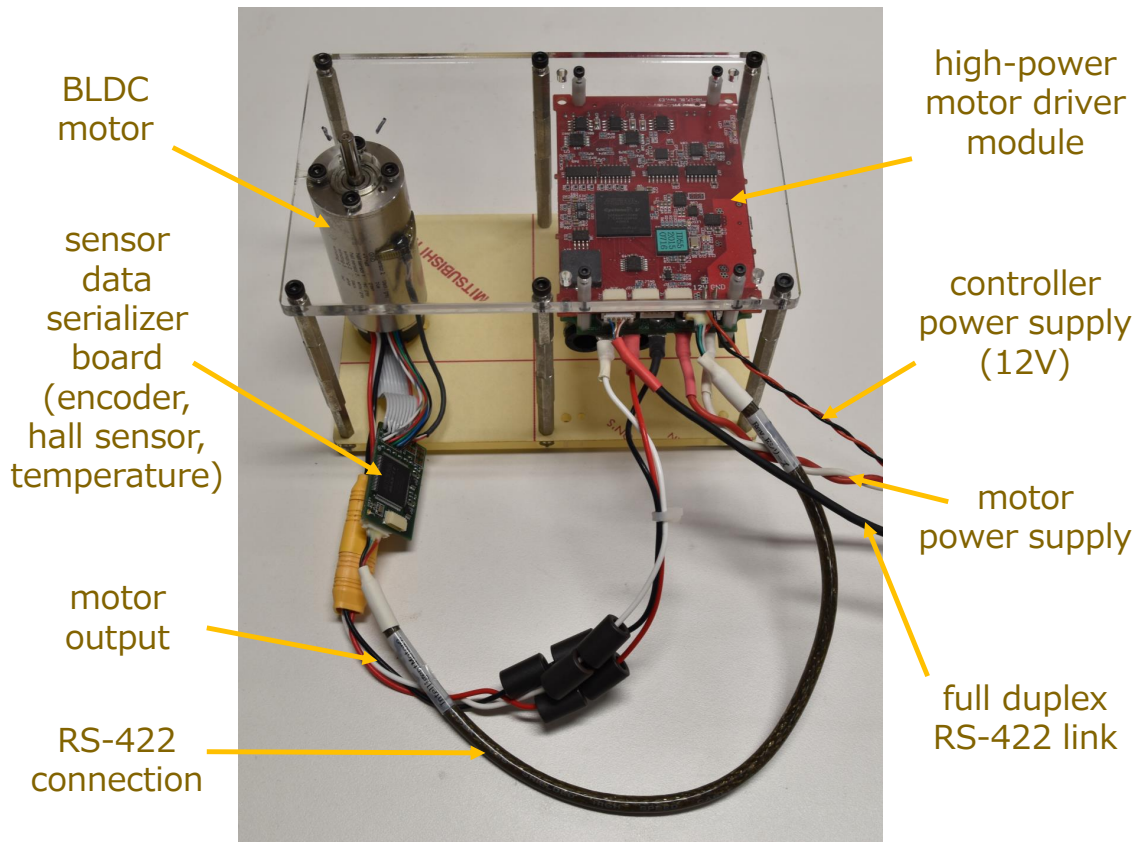


図 3.4 Brushless motor connected to the high-power motor driver module H8-17

いる。FPGA では PCI インタフェースコントローラからのバスアクセスに従い RS-422 分散ネットワークとの間でパケット交換を行う。PCI バスにおける DMA 転送の仕組みを利用した高速な読み出しデータ転送に対応している。

体内分散ネットワークは RS-422 信号を利用した独自プロトコルにより実装される。JAXON の例における体内分散ネットワークの接続構成および接続ノード数を図 3.6 に示す。

RS-422 分散ネットワークは、比較的大きな電源ノイズや空間電磁ノイズが生じるロボット体内環境において高信頼な通信を実現するため、Reed-Solomon 符号化を利用した独自プロトコルにより実装される。図 3.7 の 32 bit パケットを 1 単位として通信する。パケットには 16 bit のヘッダと 16 bit のデータが含まれる。ヘッダは、

- tag (2 bit) : NOP/Write/Read/ReadAck に対応するパケットタイプ
- device address (6 bit) : 特殊アドレス 0 に最大 63 ノードを加えたデバイスアドレス

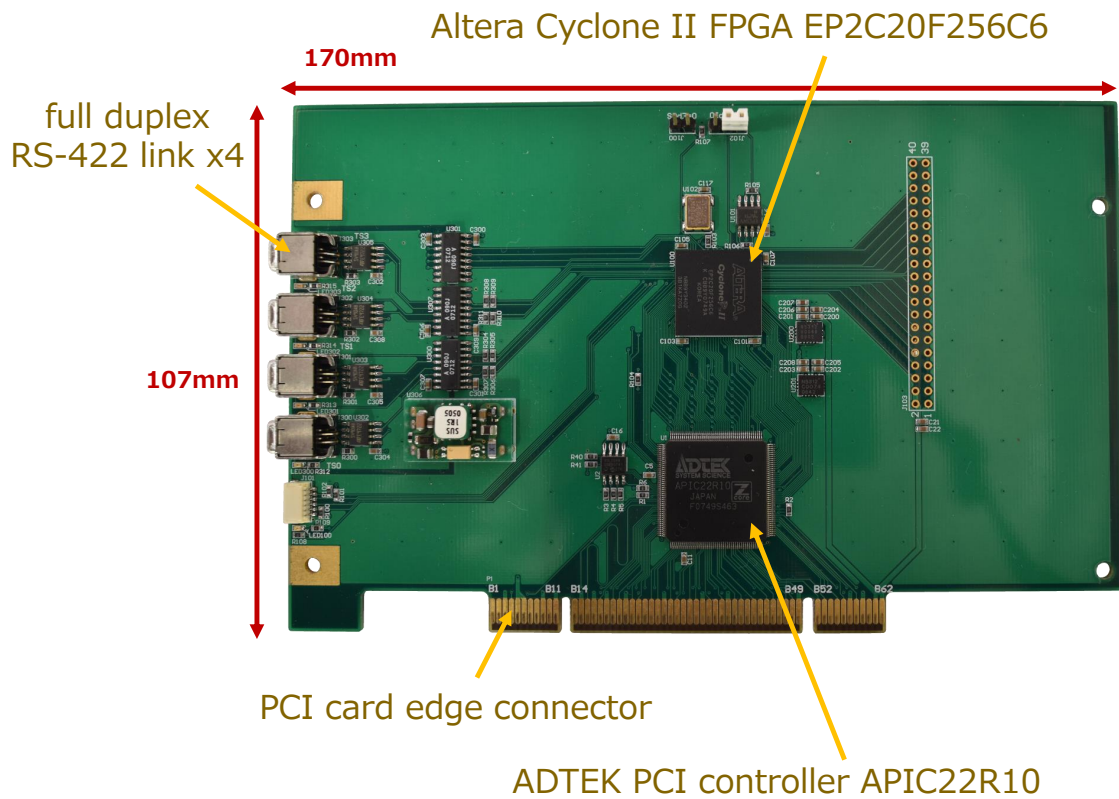


図 3.5 Overview of the PCI bridge board in previous work

- data address (8 bit) : 各ノード内のアドレス空間におけるデータアドレス

から構成される．パケットにフレームカウンタおよび追加データを加えた図 3.8 の 64 bit フレームが通信ペイロードの 1 単位として交換される．追加データは例えば，ダブルモータ駆動関節における兄弟関係を有する 2 ノード間のデータ共有などに利用されている．なお本論文におけるペイロードの計算では特別に断らない限りフレームで付加された 32 bit は含めず，本来の 32 bit のパケット部分のみを計上してデータ転送レートなどの議論を進める．フレームは Reed-Solomon 符号によって図 3.9 のように 96 bit に冗長符号化される．これにパリティビットを加えた図 3.10 の 120 bit 配列が最終的に RS-422 信号で送受信される．8 bit データ ($b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7$) に対してパリティビットを，奇数ビット分の排他的論理和の負論理，

$$p_0 = \overline{b_0 \oplus b_1 \oplus b_2 \oplus b_3 \oplus b_4} \quad (3.3)$$

$$p_1 = \overline{b_5 \oplus b_6 \oplus b_7} \quad (3.4)$$

と定義することで，誤り検出に加えて同極性ビットの連続を避ける効果も狙っている．

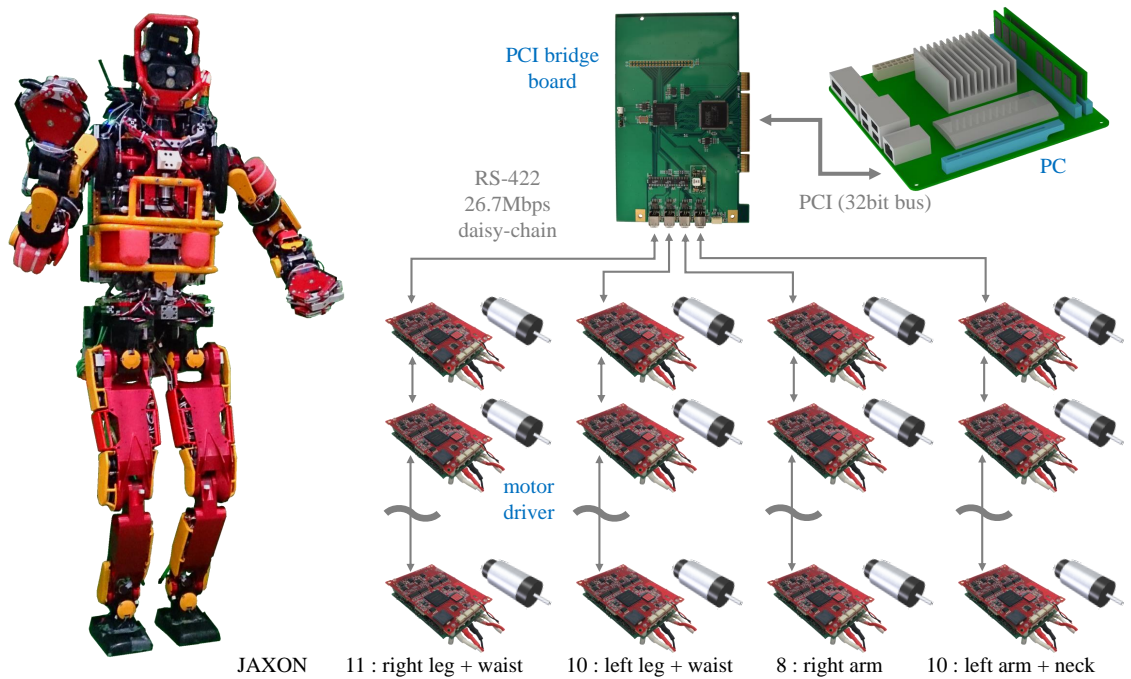


図 3.6 Distributed network for the humanoid robot JAXON in previous work

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1-2 Byte	tag		device address					data address								
3-4 Byte	data															

図 3.7 Data format of single packet with 16 bit header and 16 bit data

RS-422 物理層の転送レートは 80 MHz の内部クロックを 3 分周して、 $80 \div 3 = 26.667$ Mbps としている。32 bit のパケットが 120 bit に冗長符号化されるため、実効スループットはおよそ $80 \div 3 \times 32 \div 120 = 7.111$ Mbps となる。

すなわち、ひとつのダイジチェーンネットワークで 1 ms の制御周期のうちに転送可能なデータ量は、ホップ数などの影響を無視した上限値でも 7111 bit である。これは 3.2.3 項で示した JAXON の具体例による目標データ量 49440 bit のおよそ 7 分の 1 に過ぎない。実際には 1 周期あたりの通信データ量に 16384 bit の制約を設けた通信データ量分散スケジューリングを行うことで、およそ四肢に相当する 4 系統のダイジチェーンネットワークによるデータ転送を辛うじて間に合わせている。

ロボットの多自由度化が進み、また高度なアクチュエータ制御や大規模データ処理技

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1-2 Byte	frame counter															
3-4 Byte	auxiliary data															
5-6 Byte	tag	device address						data address								
7-8 Byte	data															

図 3.8 Data format of 64 bit frame

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1-2 Byte	frame counter															
3-4 Byte	auxiliary data															
5-6 Byte	tag	device address						data address								
7-8 Byte	data															
9-10 Byte	Reed-Solomon code[15:0]															
11-12 Byte	Reed-Solomon code[31:16]															

図 3.9 Data format of 96 bit Reed-Solomon encoded frame

術に伴う要求で通信データ項目が増加する中で，RS-422 分散ネットワークのデータ転送レートによる律速が顕現している．

3.3.2 腱駆動型筋骨格ヒューマノイドの従来実装

腱駆動型筋骨格ヒューマノイドに関する先行研究 [149][21] では，中央計算機に対して複数の中間制御基板が接続され，中間制御基板に対して更に複数のモータ制御モジュールが接続される，ツリー型論理トポロジのネットワークが構成されていた．

図 3.11 にモータ制御モジュールおよびモータ制御基板 FPGA-02D の概観を示す．ブラシレスモータでウィンチを駆動しワイヤを巻き取る機械構造とその制御回路がモジュール化されている．FPGA-02D には旧 Altera 社の Cyclone III FPGA EP3C5M164C7N が搭載され，モータのホールセンサおよび電源電流センサによる簡易的な電流 PI 制御器が実装されている．ワイヤを巻き取る用途から，モータの出力方向や外乱負荷方向が頻繁に切り替わることはないという仮定のもとで電源電流センサによるフィードバック制御が

	0	1	2	3	4	P0	5	6	7	P1	8	9	10	11	12	P2	13	14	15	P3
20 bit	frame counter				P	frame counter				P	frame counter				P	frame counter				P
40 bit	auxiliary data				P	auxiliary data				P	auxiliary data				P	auxiliary data				P
60 bit	tag	device address			P	device address			P	data address				P	data address				P	
80 bit	data				P	data				P	data				P	data				P
100 bit	RS code				P	RS code				P	RS code				P	RS code				P
120 bit	RS code				P	RS code				P	RS code				P	RS code				P

図 3.10 Data format of 120 bit Reed-Solomon encoded frame with parity bit

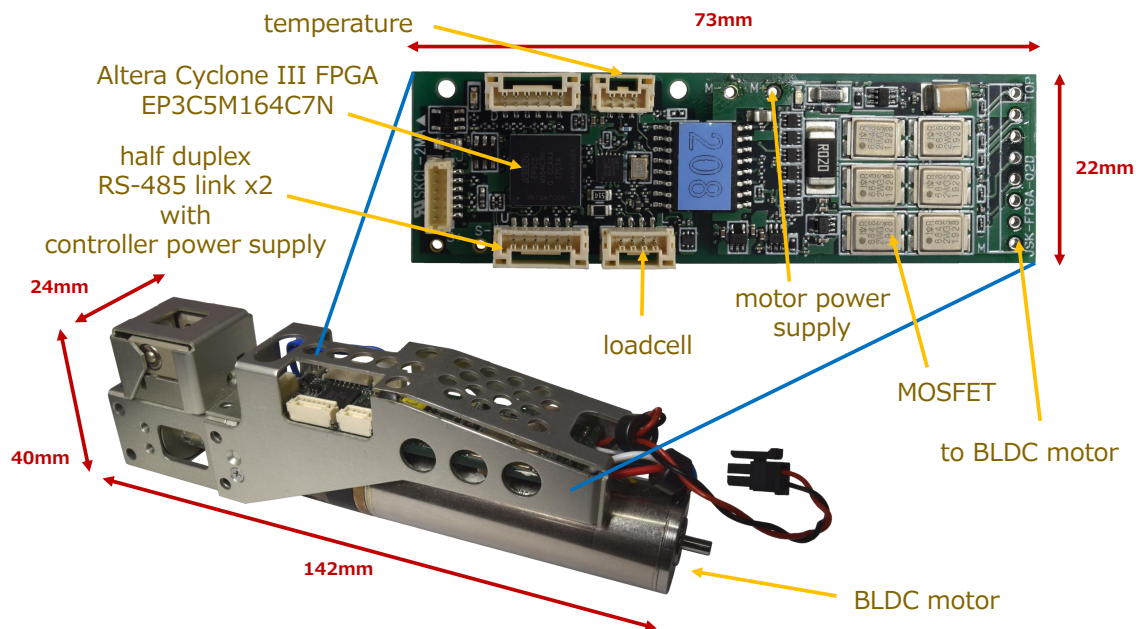


図 3.11 Overview of the wire actuator module and the motor driver board FPGA-02D in previous work

成立している．近年では細かく精密なモータ出力トルク制御やワイヤ張力制御への機運が高まる中で，この簡易仮定に基づく電源電流 PI 制御では十分な応答性を得るのが困難となる問題も生じている．

図 3.12 に中間制御基板 H8-18 の概観を示す．ルネサスエレクトロニクス社の 16 bit MCU H8S/2215R が搭載され，アクチュエータの制御計算および中央計算機との間の通信を担っている．モータ制御基板 FPGA-02D と同じく旧 Altera 社の Cyclone III FPGA EP3C5M164C7N が搭載され，FPGA-02D との間の通信を担っている．

中間制御基板 H8-18 に対して，複数台のモータ制御基板 FPGA-02D が RS-485 半二重

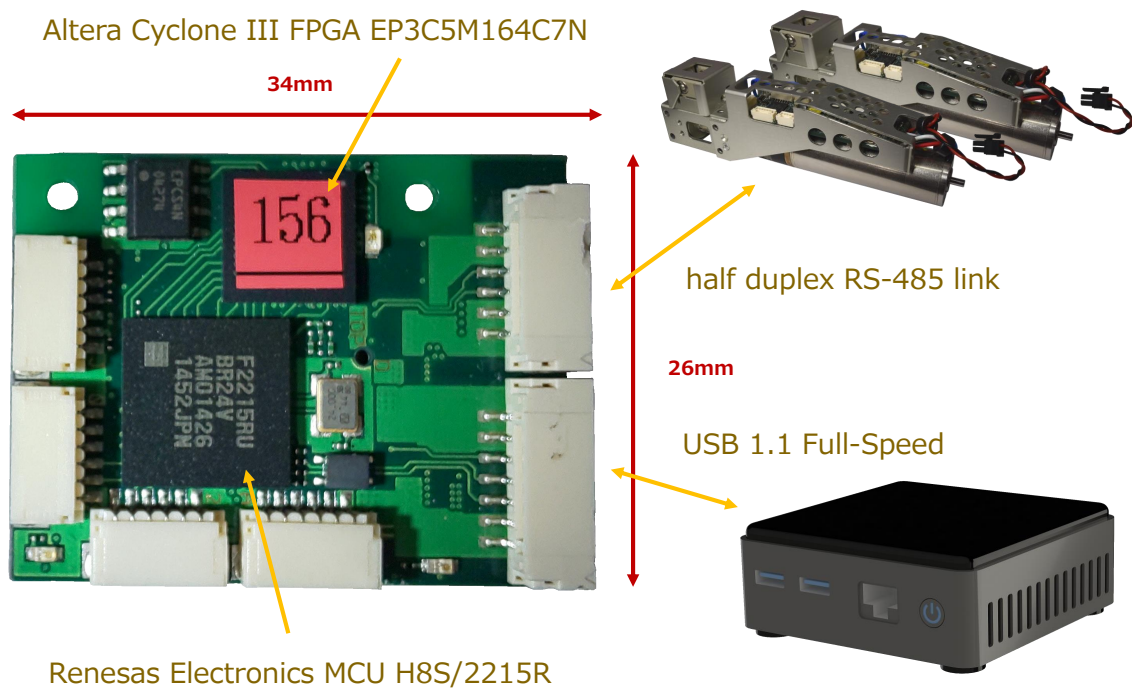


図 3.12 Overview of the USB 1.1 controller board H8-18 in previous work

通信によりデジチェーン接続される．信号線の数減らすため半二重通信が採用されている．RS-485 信号の上には 24 MHz の内部クロックを 8 分周した 3 Mbps の UART 型双方向シリアル通信が実装される．半二重通信のためプロトコルにより信号駆動タイミングが定義され，送受信を交互に繰り返す．

中央計算機と中間制御基板 H8-18 は，12 Mbps の USB Full-Speed 通信により接続される．USB デバイスの物理層は H8S/2215R MCU の機能として実装されている．USB インタラプト転送による周期通信機能を利用した 8 ms 周期のデータ交換プロトコルが実装されている．

臆悟郎の例における体内分散ネットワークの接続構成および接続ノード数を図 3.13 に示す．

USB で定義されている通信プロファイルを表 3.8 にまとめる．USB Full-Speed 規格では 12 Mbps の半二重通信が行われる．周期的なポーリング機能を提供するインタラプト転送では 1 周期あたりの転送データ量が 64 Byte に限られ，ポーリング周期は最短で 1 ms となっている．USB Full-Speed 規格の制約および MCU を用いた実装により，全身で 44 ノードの中間制御基板と通信を行う臆悟郎のシステムでは，実用的な制御周期が 8 ms にとどまっていた．

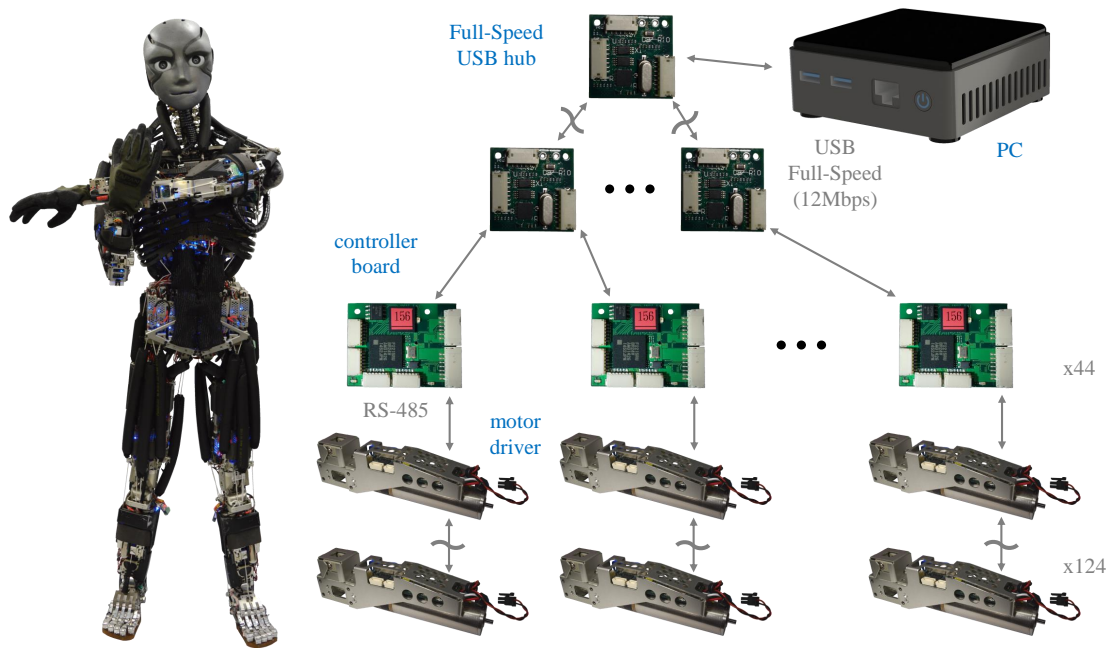


図 3.13 Distributed network for the humanoid robot Kengoro in previous work

3.3.3 多リンク飛行ロボットの従来実装

多リンク飛行ロボットに関する先行研究 [150][25] では，中枢全身制御系に対してモータ駆動出力を行う分散制御基板が CAN によるマルチドロップバスネットワークで接続されていた．図 3.14 に飛行ロボット HYDRUS の例におけるネットワーク構成を示す．

それぞれのリンクにはアクチュエータとして，プロペラ回転モータ・プロペラ傾斜用サーボモータ・関節駆動用サーボモータが搭載されている．プロペラはブラシレスモータに接続されており，PWM 信号で指令を与える既製品の ESC を利用して回転速度を制御する．プロペラ傾斜と関節駆動には ROBOTIS 社のサーボモータモジュール DYNAMIXEL MX-28AR を利用する．サーボモータモジュールは 1 Mbps の UART 互換バスプロトコル通信で制御する．分散制御基板 Neuron により，リンクに搭載された ESC およびサーボモータモジュールに対して指令を与える．分散制御基板 Neuron には STMicroelectronics 社の MCU STM32F413CHU6 を搭載している．1 Mbps のマルチドロップ CAN バスにより分散ネットワークを構成する．

中枢全身制御系は，上位の制御を担う中央計算機と姿勢制御を担う中間制御基板 Spinal が USB UART で接続された構成を取る．中間制御基板 Spinal には STMicroelectronics

表 3.8 Specification of USB communication link profile

	Low-Speed	Full-Speed	High-Speed	Super Speed	Super Speed+
compatibility	USB 1.1	USB 1.1	USB 2.0	USB 3.0	USB 3.1 (USB 3.2)
duplex	half	half	half	full	full
speed	1.5 Mbps	12 Mbps	480 Mbps	5 Gbps	10 Gbps (20 Gbps)
number of wire	4	4 - 5	4 - 5	9 - 10	9 - 10 (24)
frame period	1 ms	1 ms	125 μ s	125 μ s	125 μ s
packet size					
control	8 Byte	64 Byte	64 Byte	512 Byte	512 Byte
bulk	-	64 Byte	512 Byte	1024 Byte	1024 Byte
interrupt	8 Byte	64 Byte	1024 Byte	1024 Byte	1024 Byte
isochronous	-	1023 Byte	1024 Byte	1024 Byte	1024 Byte

社の MCU STM32H743VIT6 を搭載している．中間制御基板 Spinal が分散ネットワークに接続され，それぞれのリンクに対してモータの出力指令を与える．

多リンク飛行ロボットの従来実装では比較的少数の関節に対してフィードフォワードに姿勢軌道を与える動作が達成されていた．この用途では CAN バスで十分な性能が得られていた．しかし多リンク飛行ロボットによる空中マニピュレーションの研究が進む中で，よりしなやかな環境接触の達成が期待されている．多リンク飛行ロボットの体内制御システムには，更なる多リンク化への対応や応答性の高い関節力制御が求められてきている．そのために，力制御を可能とするモータ制御モジュール，およびこれを接続する実時間全身制御ネットワークが必要となる．

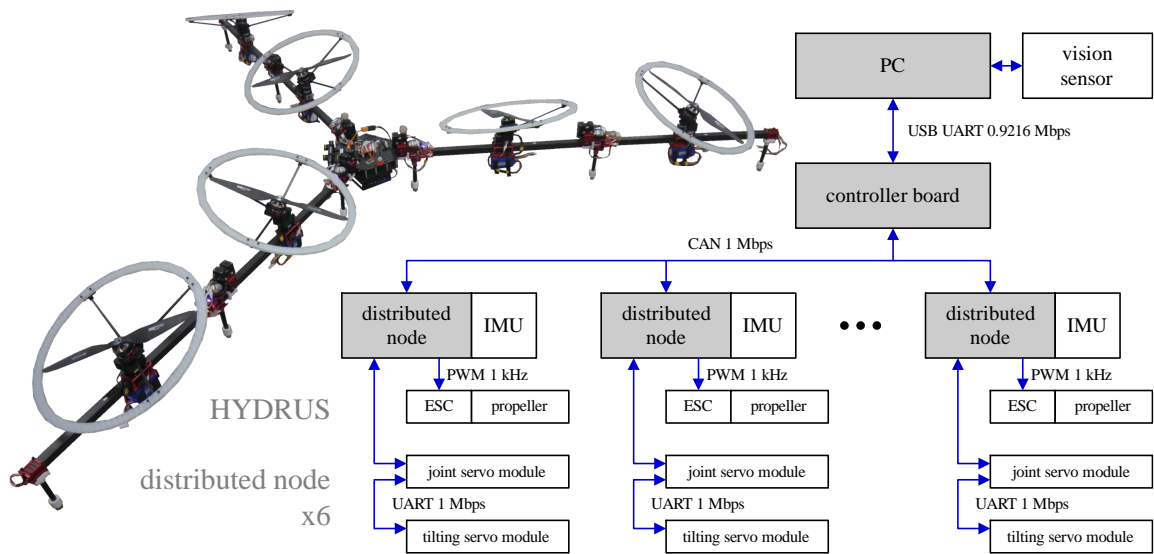


図 3.14 Distributed network for the multilink aerial robot HYDRUS in previous work

3.4 多自由度ロボットに適した実時間体内分散ネットワークの設計

3.4.1 ロボットの実時間制御に適う組込分散ネットワーク

分散制御ノードと中枢全身制御系から構成される多自由度ロボット体内制御システムにおいて、分散ネットワークは、

- 多数の分散制御ノードを接続する組込分散ネットワーク
- 中枢全身制御系において中央計算機と組込分散ネットワークを接続する基幹通信インタフェース

により実装される。

ネットワークの実装にはその時代ごとに得られる通信インタフェースから適切な構成を検討・適用し、更に次世代の通信インタフェースへと繋げてゆく、継続的な開発が必要となる。本研究でも現在の半導体技術や計算機技術の実情を踏まえつつ試行錯誤を進める。

組込分散ネットワークには、従来より CAN をはじめとした汎用シリアル通信がよく利用されてきた。普及している汎用シリアル通信のデータ転送レートは最大で数十 Mbps 程度である。トランシーバを実装する半導体製品、および伝送路を構成するコネクタやケーブルなど、安価に利用することができ、これからも多くの用途では強力な手段となり

うる．しかし3.3節で考察したように，より多自由度化や複雑化が進むロボットにおいては不十分となる場面が出てきた．

組込分散ネットワーク高速化の要求を受けて Ethernet ベースのフィールドバスを採用する事例も出てきた [93][15]．既に実用化されている産業用 Ethernet のデータ転送レートは最大で 100 Mbps 程度となっている．一般的な Ethernet ネットワークの物理層に利用されるコネクタおよび絶縁回路は大型であるため LVDS などのより小型に収まる物理層が採用される場合もある．LVDS 信号を利用した他の独自プロトコルによる実装例も存在する [109][36][90]．

本研究では 2.4.4 項で分析したロボット体内制御システムの実時間性・展開実用性要件に基づいて，従来手法の課題を解決する組込分散ネットワークの設計および実装を提案する．

実時間性を満たすには，必要なデータ量を十分低遅延に交換完了するだけの通信速度が前提として求められる．式 3.1 において，データ転送レート R_d を十分大きく，1 hop あたりの平均パケット処理時間 T_h を十分小さくして，データ転送遅延 $T_d \leq T_d^{\max}$ の制約を満たす必要がある．ネットワークのドメインを分割して通信を並列化することが可能なプロトコルであれば，ドメイン内のノード数 N_n を小さく抑えることで制約を緩和することができる．ノード数 N_n を小さくすると，それに伴いパケット数 N_p も小さくなる．

CAN や RS-485/RS-422 に代表されるような最大でも数十 Mbps の汎用シリアル通信を物理層に用いた実装では，多自由度ロボットのおよそ 1 ms 以下の目標制御周期に対して十分なデータ転送レートが確保できなかった．3.3.1 項の JAXON の例では RS-422 を利用して，物理層 26.667 Mbps，実行スループット $R_d = 7.111$ Mbps のプロトコルを実装しており，本来の理想的なデータ量 $N_p D_p = 1545 \times 32 = 49440$ bit を $T_d \leq 1$ ms で交換完了するには不足があった．そのため，RS-422 によるダイジチェーンの 4 並列接続に対応させるとともに，1 周期あたりのデータ量に妥協的な制限を設け，ノード数 N_n およびパケット数 N_p を小さくすることで $T_d \leq 1$ ms を辛うじて満たす実装で運用されていた．

EtherCAT に代表される 100 Mbps の産業用 Ethernet 実装では，汎用シリアル通信と比べて通信速度が向上し，1 ms 以下の目標制御周期で全身のデータを交換するネットワークの実現に近づいているが，十分高速であるとは言い難い．実際には EtherCAT インタフェースを搭載した既製品のモータ制御モジュールを利用して，決まった仕様で必要最小限のデータのみを交換するシステムにより 1 ms 以下の制御周期を実現している例が多い．しかし，体内制御システムを多様なロボットに展開し高度な分散制御系の構成を模索してゆく上で，分散制御ノードの拡張開発可能性が重要となる．3.2.3 項に例示したような多項目のデータや，筋骨格ロボット・フィールドロボットへの適用で更に増える追加アクチュエータ・追加エンコーダ・姿勢センサなど，多くの拡張データにも対応して十分

低遅延な通信を達成する性能が求められる。

近年では PCI Express 技術の進展もあり、1 Gbps を超えるオーダの高速シリアル通信が実用化されている。高速シリアル通信に対応したトランシーバを搭載する小型半導体製品も増えている。USB SuperSpeed 規格に代表されるように、離れたノード間のケーブル接続により高速シリアル通信を実装する技術も登場してきている。高速シリアル通信では、従来よりロボット体内分散ネットワークで利用されてきた汎用シリアル通信と比べると 2 桁から 3 桁ほど高速なデータ転送が可能である。高速シリアル通信をロボット体内分散ネットワークに適用できれば、多自由度ロボットの高度な体内分散制御システムにおいても十分低遅延なデータ転送性能が得られる。

一方で、強い電源ノイズや空間電磁ノイズの影響を受け、かつ省スペースが求められる多自由度ロボットの体内分散ネットワークにおいて、分散制御ノード間を電気信号で接続する方法は高速化の限界に達している。26.667 Mbps の RS-422 通信では大出力モータを多数駆動するロボット体内電磁ノイズ下での信頼性が低く、図 3.10 の低効率な冗長符号化が運用上必要となっていた。導体ケーブルによる高速シリアル通信をロボット体内に実装するのは、ノイズの影響により現実的ではない。電磁ノイズや機器間電位差の影響をほとんど受けない信号伝送方式として、光信号を利用する方式が挙げられ、低速なものは Sercos などの産業用フィールドバスでも利用されてきた。本研究では光信号を媒体とした、組込分散ネットワークにおける高速光通信の実装を試みる。高速光通信の物理層実装に関しては先導研究を通して検討が進められている [151][152]。低遅延で高信頼な高速光通信の導入は、本研究で目指す実時間性の達成に大きく貢献する。

光通信を利用する場合に限らず電気信号の場合にも共通して、ロボットの实時間分散制御のためには低遅延だけでなく実時間性や展開実用性を総合的に満たす分散ネットワークが求められる。中でもハードウェアの小型軽量化は多自由度ロボットにおいて非常に重要かつ困難な課題となってきた。

半導体技術の進展に伴い、小型で高性能な FPGA デバイスが実用化されてきている。FPGA を活用することで、多くの要件を満たす小型パッケージな分散制御ハードウェア実装が可能となる。プログラマブル論理回路により、周辺回路との接続やその制御方式を自在に再構成することができる。そのため、

- 低遅延：自由なプロトコル実装によるロボット制御に適した低遅延通信
- 低ジッタ：時間確定的で非常に低ジッタな論理回路レベルのプロトコル実装
- 信頼性：信号の誤り検出や異常対応による高信頼化
- 小型軽量：外部に最低限の物理層回路のみを搭載した部品削減による小型軽量化
- 低消費電力：電力高効率な回路構成とパワーゲーティングによる低消費電力化
- 汎用性：通信プロトコルのコンポーネント分離と異種ハードウェアへの再利用

- 運用性：柔軟な拡張開発可能性の提供と多ノードへの容易な複製

といった総合的な特性向上が見込まれる。

3.4.2 高速光通信分散ネットワークの設計

ロボット体内分散ネットワークに向けて，光信号を媒体とした高速シリアル通信の実装を考える．従来型の汎用シリアル通信と比べて，高速光通信には次のようなメリット・デメリットが挙げられる．

- 高速光通信のメリット
 - 1 Gbps を超える高速データ転送
 - 光伝送経路で空間電磁ノイズの影響を受けない
 - 送受信ノード間が完全に絶縁されるため電源ノイズによる電位差の影響を受けない
 - 細く曲率の大きい光ファイバケーブルが利用でき配線の制約が緩和される
- 高速光通信のデメリット
 - 電源電力は伝送できない
 - 単純な電気端子と比べてコネクタが大きくなる
 - 高速トランシーバを搭載する半導体製品に限られる
 - トランシーバおよびコネクタともに高価
 - ケーブルをカスタマイズする自由度が低下し長さの変更も困難

高速・高信頼なデータ転送性能に優れている一方で実装上の難点が多い．特に分散制御ノードを構成する制御基板のサイズは，多様な多自由度ロボットへの適用に向けて重要な課題となる．本研究では，高速・高信頼なデータ転送を実現しつつ小型サイズに収めた光通信ネットワークの実現を目指す．

光通信では電気信号と光信号を相互変換するための発光素子および受光素子が必要となるためコネクタが大型になる傾向にある．その中で，小型なコネクタと細い光ファイバで構成された光通信ケーブルが実用化されてきている．

図 3.15 のような小型光通信ケーブル製品例が存在する．パナソニック社の全二重光アクティブコネクタ製品とヒロセ電機社の片方向光アクティブコネクタ製品を一般的な導体ケーブルと比較している．光アクティブコネクタは光ファイバに発光・受光素子内蔵コネクタを接続したケーブルであり，基板上に搭載された電気コネクタと接続して電気信号と光信号の相互変換により基板間の信号伝送を担う．これらの小型光通信ケーブルでは導体ケーブルと比べて非常に細い直径で高速信号を高信頼に伝送することを可能としてい

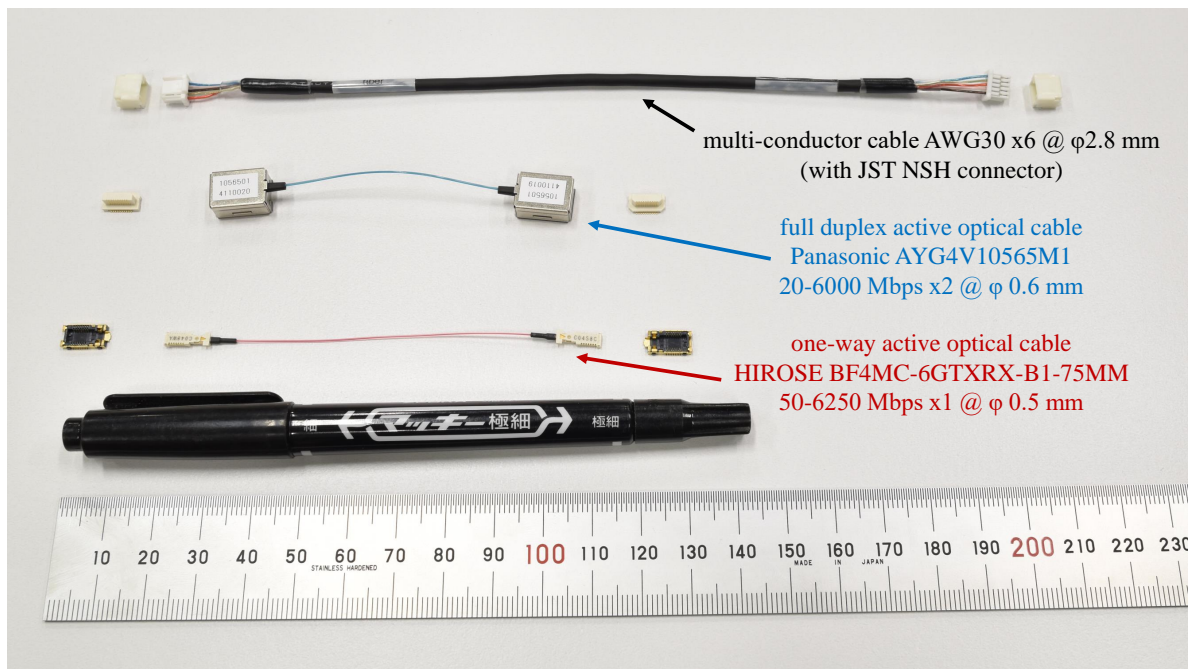


図 3.15 Overview of active optical cable

る．一般的なツイストペア導体ケーブルでは Ethernet で利用されるように最大でも数百 Mbps 程度の信号伝送が限界であるが，Panasonic 社の全二重光アクティブコネクタ製品では最大 6 Gbps の信号伝送を可能としている．細い光ファイバにより柔軟な導体ケーブルと同程度である 15 mm の最小曲げ半径を保証している．本研究の組込分散ネットワーク実装例でも Panasonic 社の全二重光アクティブコネクタ製品を利用する．

高速光通信の実装構成例を図 3.16 に示す．それぞれのネットワークノードには，高速差動電気信号と光信号を相互変換するコネクタおよび高速トランシーバユニットを備えた FPGA を搭載する．FPGA の高速トランシーバユニットによる高速シリアル通信を光信号に変換することでノード間の高信頼な高速通信を実現する．

高速シリアル通信は，2 Gbps のデータフレームを 8b/10b 符号化した 2.5 Gbps の信号により実装する．8b/10b 符号化により直流成分を除いた高品質な高速シリアル通信が実現され，同時にエラー検出も可能となる．FPGA の高速トランシーバユニットに対して PCML 信号で入出力される．高品質な高速シリアル通信のためには基板上の電気信号伝送路の設計も重要な要素となる．PCML 信号には 100 Ω の差動インピーダンスによる実装が求められる．基板の層構成や配線幅・間隙を適切に設計することで差動インピーダンスを 100 Ω に調整する．信号線にコンデンサを直列配置することで AC カップリングを施し送受信のバイアス電圧を整合する．受信端には Thevenin 終端などの方法により，所望のバイアス電圧で差動抵抗 100 Ω の終端を施し，効率的な信号伝送を行う．終端抵抗

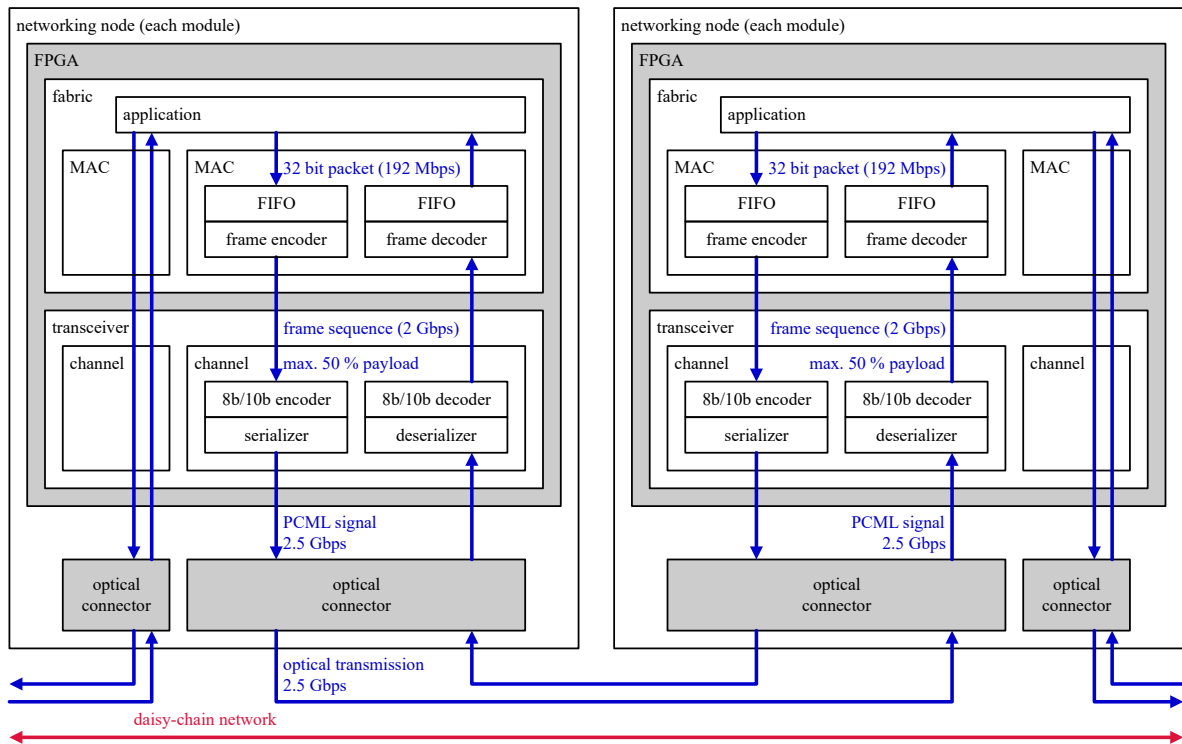


図 3.16 Optical transmission link and configured maximum data rate

は FPGA や光通信ケーブルのトランシーバに内蔵されている場合もある。

光通信ケーブルでは PCML 信号で FPGA とデータを交換して、これを光信号に変換することで、ノード間の信号伝送を行う。光信号の伝送路では電磁ノイズの影響をほとんど受けない非常に高品質な通信が行われる。

FPGA の高速トランシーバユニットとアプリケーションの間でデータを交換する MAC 層はプログラマブル論理回路により任意の構成が可能である。ここでは FIFO バッファを介して、アプリケーションで取り扱う 32 bit パケットと 2 Gbps 高速フレームデータを相互変換して高速トランシーバユニットと交換する。図 3.7 と同じフォーマットの 32 bit パケットに図 3.8 と同様の追加データを施した 64 bit フレームを利用する。8b/10b 符号化では 8 bit データを 10 bit に符号化する冗長性を利用して非データ信号を挿入することができる。フレームの間隙はシリアル信号の一意的なアラインメントにも対応した 8b/10b K28.5 信号で埋めることでアイドル状態を示す。フレームの開始直前には 8b/10b K27.7 信号を挿入することでフレーム位置を伝える。

アプリケーションからは任意のタイミングで MAC 層とパケットを交換する。MAC 層から下位は十分高速であるため数十 MHz 程度のクロックで駆動するアプリケーション回路が受ける制約は少ない。

多自由度ロボットにおいて光通信ネットワークを採用するには、その多数のアクチュ

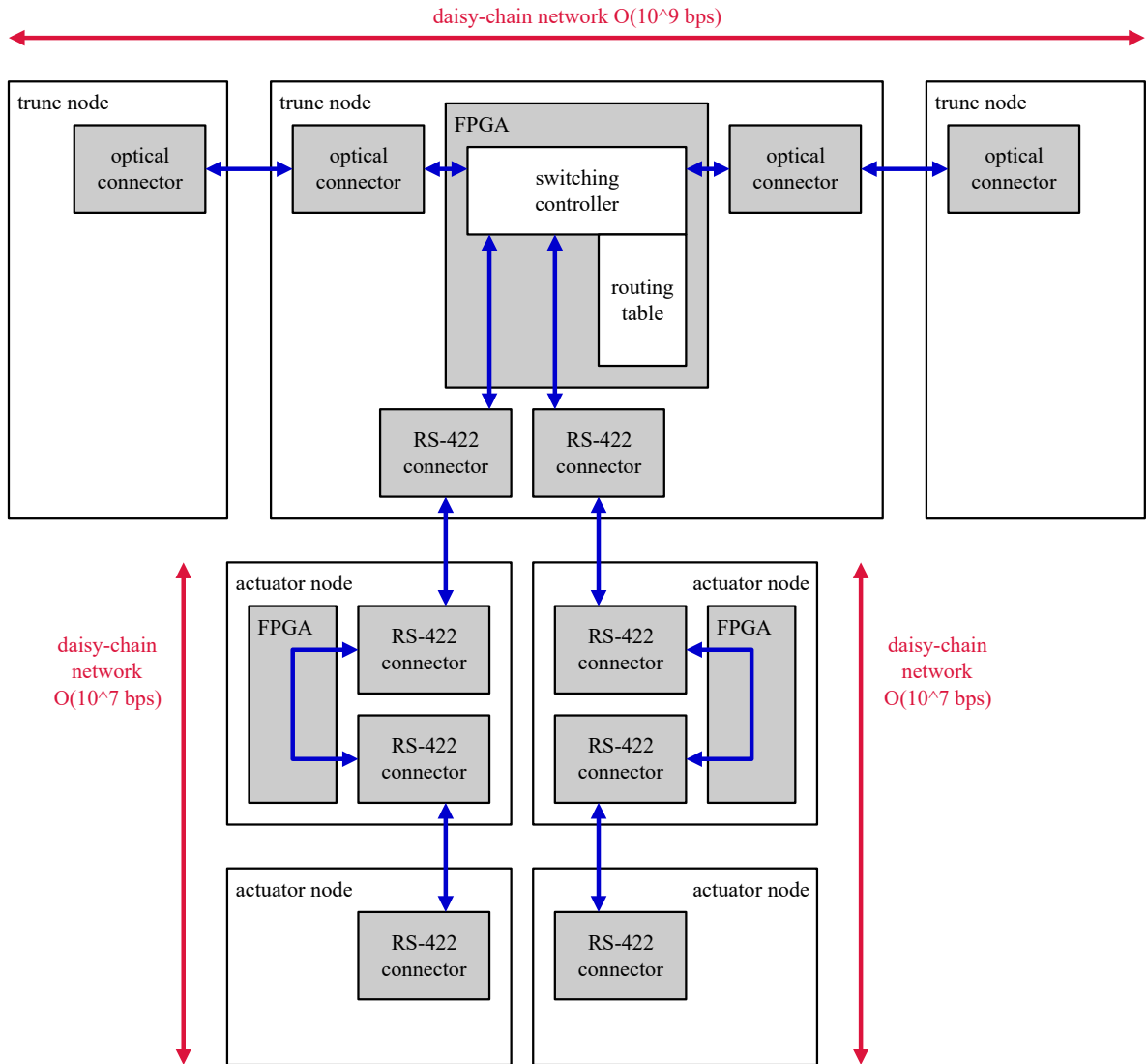


図 3.17 Trunc network with optical transmission link

エータゆえに、実装費用が切実な問題となる場合も多い。光通信による高速なデータ転送は幹線ネットワークのみに適用し、末端ネットワークには従来型の汎用シリアル通信を採用する構成も考えられる。図 3.17 のように、光通信によって幹線ネットワークを構成し、幹線から末端ネットワークを分岐させることで、少数の光通信ノードのみの利用により、低速な汎用シリアル通信ネットワークの直列ノード数を抑えつつ、ネットワーク全体の転送可能データサイズを増大させることができる。ロボットの体内ネットワーク構成が動的に変更される場合は少ないため、これを考慮したルーティング機能を幹線ネットワークに実装すれば良い。

3.5 分散ネットワークを構成する小型高速通信モータ制御モジュールの設計と実装

3.5.1 高速通信ネットワークの小型高信頼化設計

多自由度ロボットの体内分散制御系の設計において、ネットワークを構成する分散制御ノードの小型軽量実装が重要な課題となる。およそアクチュエータ数に等しい数の制御モジュールが分散配置され、その総体積はロボットの構造設計に対する無視できない制約となる。

腱駆動型筋骨格ヒューマノイドに関する先行研究では、図 3.11 で紹介した小型なモータ制御モジュールが開発されている。直径 22 mm のブラシレスモータを駆動するために同じ 22 mm 幅に収めたモータ制御基板が開発され、モータと一体化した小型モジュールとして運用されている。

腱駆動型筋骨格ヒューマノイド腱悟郎の例では、体重 56.5 kg の人間と同程度のプロポーションで設計されており [11] その専有体積はおよそ 60 L と見積もられる。モータ制御モジュールの直方体専有体積は図 3.11 より $142 \times 40 \times 24 = 136320 \text{ mm}^3$ であり、124 のモータを搭載する腱悟郎のモータ制御モジュール専有体積を見積もるとおよそ 16.9 L となる。この小型モジュール設計の場合でも、ロボットのうちおよそ 3 割に及ぶ体積をアクチュエータが占めていることになる。モジュール小型化の重要性が見て取れる。

ロボットの分散制御ノードに用いる制御基板の小型化設計においては、電子回路実装としての特性とロボット部品としての特性をともに深く理解することが重要な意義を持つ。電子部品を搭載した基板として単に小型化するだけでなく、ロボットに適用するための信頼性・汎用性・運用性を満たしながら、ロボットを構成する機械部品としての適切な設計を作り込むことが求められる。

制御基板の構成部品には小型ながら高信頼な部品を選定する。ロボットで運用するには、理想状態での回路特性に加えて、振動や電源電圧変動の影響を注意深く考慮する必要がある。

コネクタ部品の耐振動性は継続動作の信頼性を大きく左右するため、頑強に嵌合可能な、接触不良を生じにくいコネクタが適している。特に大型の基板対基板コネクタは問題を生じやすいため、機械的な固定方法とも合わせて設計を整合させる。ロボットに固定した際にコネクタや締結部に大きな応力が残らない設計が有効である。

ロボット体内では電源電圧が大きく変動したり接地電位が不安定となったりする状況が多発するため、それぞれの部品の定格電圧には大きな余裕を取る。ロボットの電源電圧は、搭載される各種デバイスの受電仕様や電池および外部給電の仕様など、ロボットごと

の制約を考慮して決定される。制御基板を広範囲の電源電圧に対応させることで、多様な電源構成を有するロボットへの展開も容易なものとなる。電源電圧変動やノイズの影響を受けやすいオンボード電源線や信号線にはフィルタの適用も効果を発揮する。

高速通信ネットワークを構成するアクチュエータ制御基板は、高速信号回路と大電流回路の混合構成となる。大電流の生じる空間経路を考慮した高速信号配線を行わなければ信頼性が大きく損なわれる。大電流の生じる経路では同時に大きな熱が発生する。基板の銅箔配線や部品表面を利用した排熱構造も重要になる。特に発熱する部品はヒートシンクに接続できるように機械設計と整合させる。コネクタから外部に接続される配線においても、高速信号の配線と変動的な大電流が生じる電源線やアクチュエータ駆動線を隔離しやすくするための配置を調整する。

これらの要素に注意しながら、ロボット体内の機械構造と相性の良い分散配置が可能な小型化設計を行うことで、信頼性・汎用性・運用性を満たす実用的な高速通信ネットワークが実装可能となる。

3.5.2 腱駆動向け小型モータ制御モジュール

従来の腱駆動向けモータ制御モジュールではモータ制御基板を小型に収めるためにその機能は最低限のものに留まっている。モータ制御機能として電源電流に基づく簡易的な電流制御が実装されている。より高位なサーボ制御は別途配置された中間制御基板に任せている。通信機能はRS-485信号による半二重で3 Mbpsの低速なものとなっている。

近年では半導体プロセスの微細化が進み、より小型で高性能な半導体製品が実用化されてきている。従来と同じ基板サイズでもサーボ制御やより高速な通信に対応したシステムの構成が可能となっている。

腱駆動型モータ制御モジュールへの適用を目指し、従来の簡易電流制御型モータ制御基板と同じサイズに最新のFPGAやMCUによる高度なコントローラを収めた小型モータ制御基板を開発した。図3.18に腱駆動向け小型モータ制御基板FPGA-03Dの概観を示す。

FPGA-03Dには低消費電力で超小型パッケージのFPGAであるIntel社のCyclone 10 LP FPGA 10CL016YM164I7GおよびSTMicroelectronics社のMCU STM32F411CEU6を搭載している。従来の腱駆動向けモータ制御基板で達成されていなかった次の機能を実現している。

- モータエンコーダを用いた分解能の細かいベクトル制御
- モータ巻線電流フィードバック制御
- 基板単体での1 ms周期サーボ制御

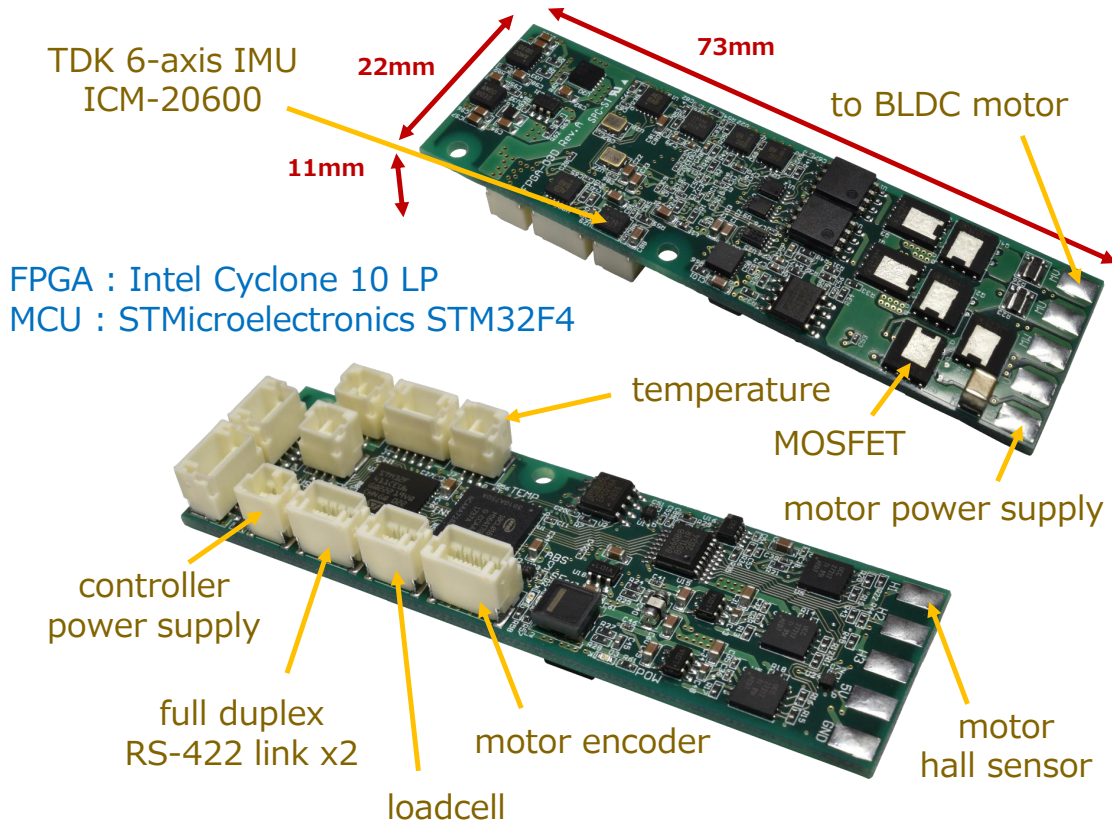


図 3.18 Overview of the motor driver board FPGA-03D

- 全二重 RS-422 による 26.667 Mbps 通信
- IMU を用いた姿勢計測

通信プロトコルや制御システムに関して多くの部分を，ギア駆動型大出力ヒューマノイドのために開発しているものと共通化している．共通化することで，本研究で提案する体内分散実時間制御システムをより多様なロボットへ一般化して適用することを目指している．

図 3.19 に，モータと合わせて従来と同様の小型パッケージに収めた腱駆動向け小型モータ制御モジュールの概観を示す．

ワイヤの張力はわずかな巻取り量に応じて大きく変化するため，精密な張力制御のためには細かい位置分解能が重要である．そのため，エンコーダを搭載したモータに対応し，モータの回転位置を細かく測定可能としている．例として図 3.19 の構成では，maxon motor 社の 4 極ブラシレスモータ EC-4pole 22 90W および 1 周分解能 4096 のアブソリュートエンコーダを用いている．従来のホールセンサを用いた手法では 1 周あたり電

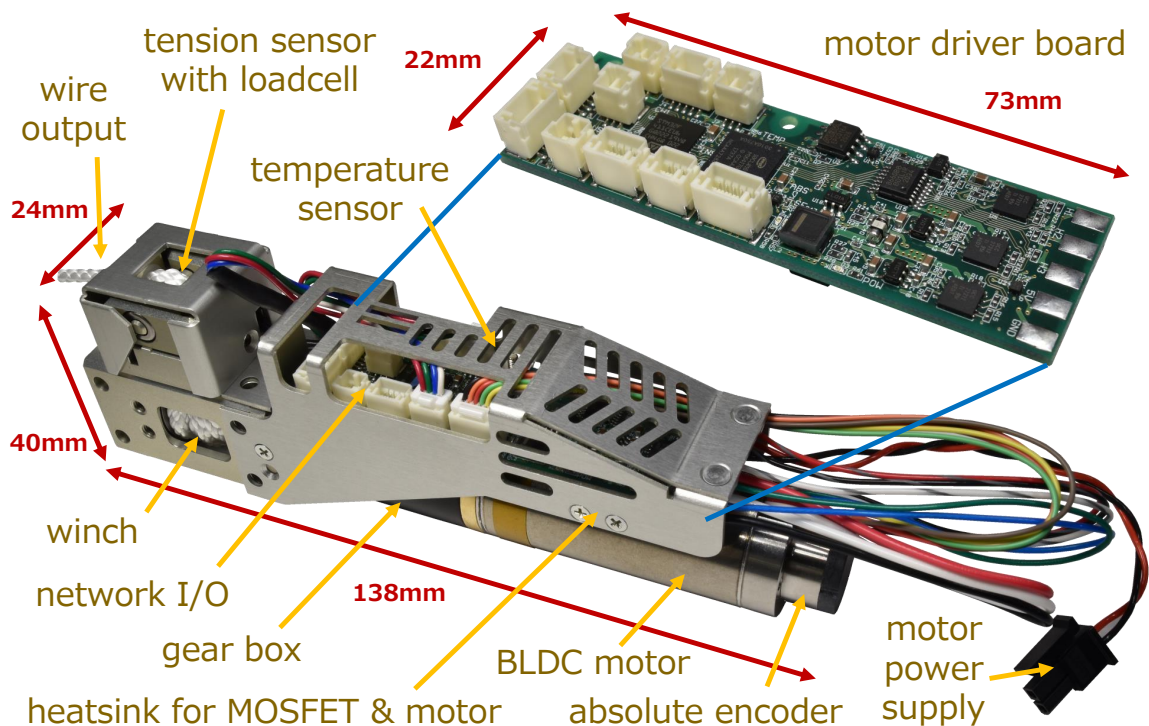


図 3.19 Overview of the wire actuator module with the motor driver board FPGA-03D

気角 2 周に相当するため分解能が 12 となっていた。アブソリュートエンコーダにより 341.3 倍の細かい位置測定が可能となっている。

モータ制御基板の MOSFET やモータ巻線で生じる熱を容易に筐体へ伝達できる構造としている。図 3.20 のようにアルミニウム合金製のフレームでヒートスプレッドを構成し、絶縁伝熱シートで発熱部へ接続する。これにより大出力なモータ駆動を長時間継続して実行することが可能となる。

3.5.3 小型光通信モータ制御モジュール

制御基板との一体化が要求されるほど高密度にアクチュエータが配置される多自由度ロボットでは、それだけ全身のアクチュエータ数も多いものとなる。アクチュエータあたりに必要な通信データ量は通常、制御基板のサイズによらず同程度であると考えられる。同程度の総体積のロボットで考えた場合、分散ネットワークを構成するアクチュエータ制御モジュールの小型化の要求とはすなわち、その程度の 2 乗に相当するモジュールサイズ対通信データレート密度の要求に他ならない。小型モータ制御モジュールにこそ高速な通信が求められる。

そこで、高密度アクチュエータ配置要求を抱える多様な形態のロボットに 3.4.2 項の高

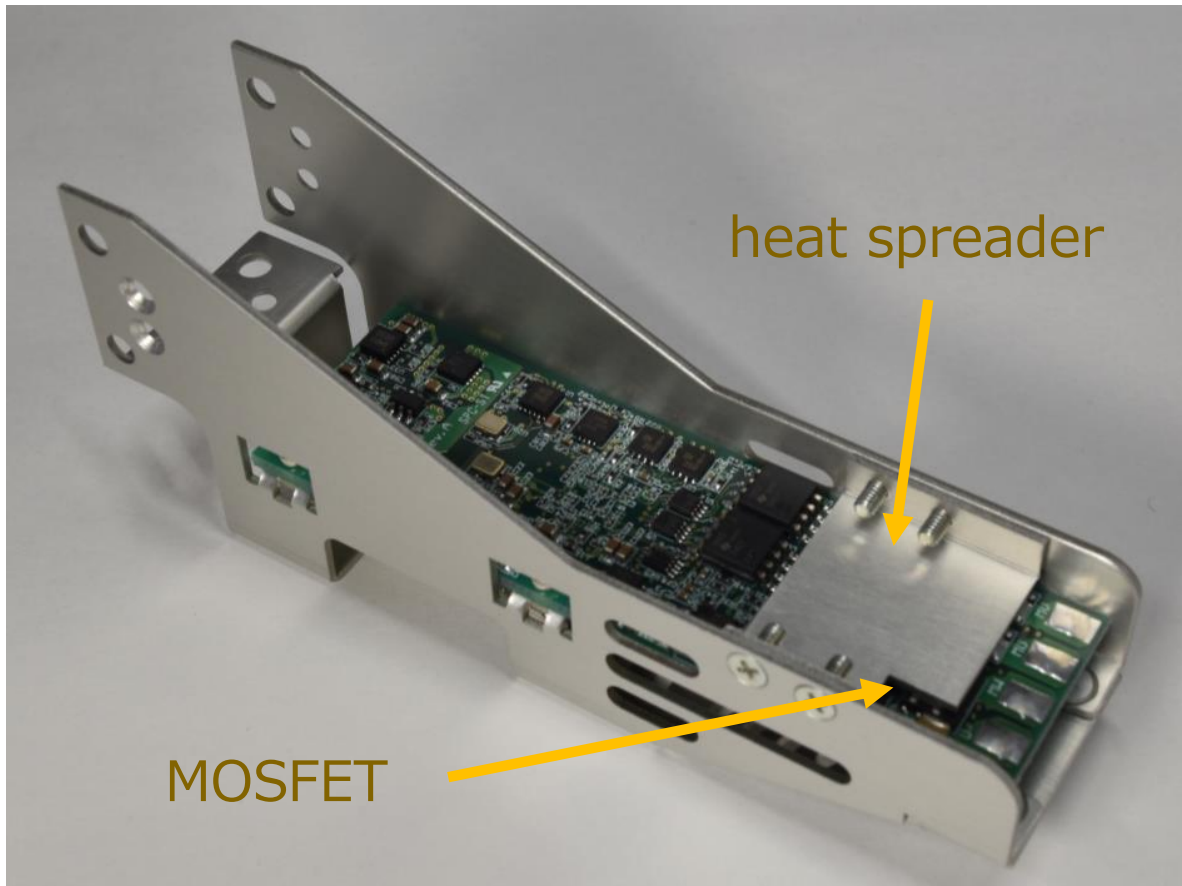


図 3.20 Heat spreader structure in the wire actuator module with the motor driver board FPGA-03D

速光通信を適用することを目指し、省スペースながら高速光通信を実現する小型光通信モータ制御モジュールを開発した。これによりモータと同程度の 22 mm 幅で、1 Gbps を超える通信が可能となる。

図 3.21 に小型光通信モータ制御モジュール SU-05-OPT の概観を示す。

SU-05-OPT には小型ながら高速トランシーバユニットを備えた Intel 社の Cyclone V GX FPGA 5CGXFC3B6U15I7N および STMicroelectronics 社の MCU STM32F411CEU6 が搭載されている。モジュールはコントローラ基板・モータドライバ基板・I/O 基板の 3 基板から構成されるスタック構造となっている。3 基板のスタック構造を採用することにより、

- 高さ方向に伸ばして水平面積を極小化することができる
- 一部の基板を別のものに換装することが可能であり構成変更を試す際のコストを抑えることができる
- 故障頻度の高いモータドライバ部分を高価なコントローラ基板と切り離して独立し

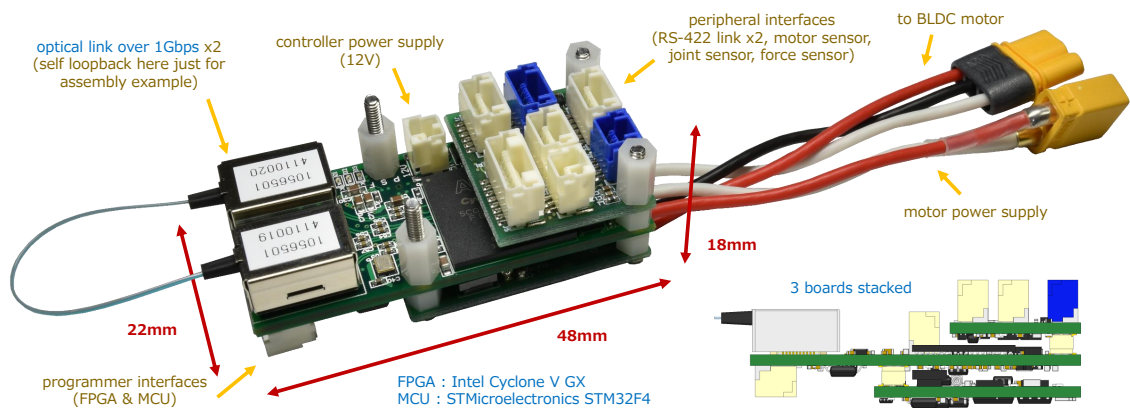


図 3.21 Overview of the motor driver module SU-05-OPT with optical link

た修理・交換ができる

といったメリットが生じる。

3.5.4 光通信分散ネットワークの性能評価

従来型の汎用シリアル通信に対する高速光通信の優位性の程度を評価する。

光通信のプロトコルとして、実用上妥当な図 3.16 のデータ転送レートを実装している。物理層は 2.5 Gbps の高速シリアル通信と 8b/10b 符号化により 2 Gbps のデータ転送レートとなっている。実際のアプリケーションではモータ制御モジュールの 24 MHz のベースクロックに対して 32 bit のパケットを最大で 4 クロックに 1 度送信する回路を構成して運用している。すなわち実効データ転送レートは 192 Mbps となっている。アプリケーション構成次第で更に数倍程度の高速化は困難ではない。

4.3.2 項で紹介する組込中間層計算機モジュールをネットワークの中核ノードとして、これに対して最大 6 ノードの小型光通信モータ制御モジュールをダイジチェーン接続する構成で、光通信による接続と RS-422 汎用シリアル通信による接続の場合でデータ転送遅延を比較する。通信遅延は組込中間層計算機モジュールの FPGA に実装されたロジックアナライザにより測定する。

ダイジチェーンネットワークに送信した最初のパケットに対するレスポンスが返ってくるまでの時間を接続ノード数ごとに図 3.22 に示す。また複数のパケットを連続して送信した際に全てのレスポンスが返ってくるまでの時間をデータサイズと接続ノード数ごとに図 3.23 に示す。3.2.2 項で求めた線形なモデルによく従う結果が実測されている。式 3.1 をもとにデータ転送レート R_d と 1 hop あたりの平均パケット処理時間 T_h は表 3.9 のよ

表 3.9 Identified transfer latency parameter for optical transmission link and electrical RS-422 link

parameter	optical link	RS-422 link
data rate R_d	191.58 Mbps	7.1200 Mbps
hop time T_h	0.34022 μ s	3.52522 μ s

うに同定される．192 Mbps の光通信と 7.111 Mbps の RS-422 汎用シリアル通信，両者ともに設計値通りのデータ転送レートが達成されている．

式 3.2 により求まるパケット数上限を図 3.24 に示す．デジチェーン接続数が 1 node の場合と 10 node の場合を例に挙げている．およそ 1 ms の制御周期を指標とする現在のロボット実時間制御系において，制御周期内に十分転送可能なパケット数を比較する．ここでは十分転送可能な指標として制御周期の 30 % に相当する 300 μ s で比較する．従来型の RS-422 汎用シリアル通信では 1 周期あたり 64 packet に留まる．10 node のデジチェーンの場合は 1 node あたり 32 packet の限られたデータしか取り扱えない．3.3.1 項でも考察した通り 3.2.3 項の目標データ量に到達していない．一方で光通信では 10 node で 1 周期あたり 1736 packet を取り扱うことができる．RS-422 汎用シリアル通信と比べると 54.25 倍のパケット数となる．RS-422 汎用シリアル通信と同じ 32 packet は 16 μ s のうちに転送完了する．実時間制御に利用する目標データ量に対しても十分であり，更に大規模ログデータ分析や画像データ転送などへの今後の応用も期待される．

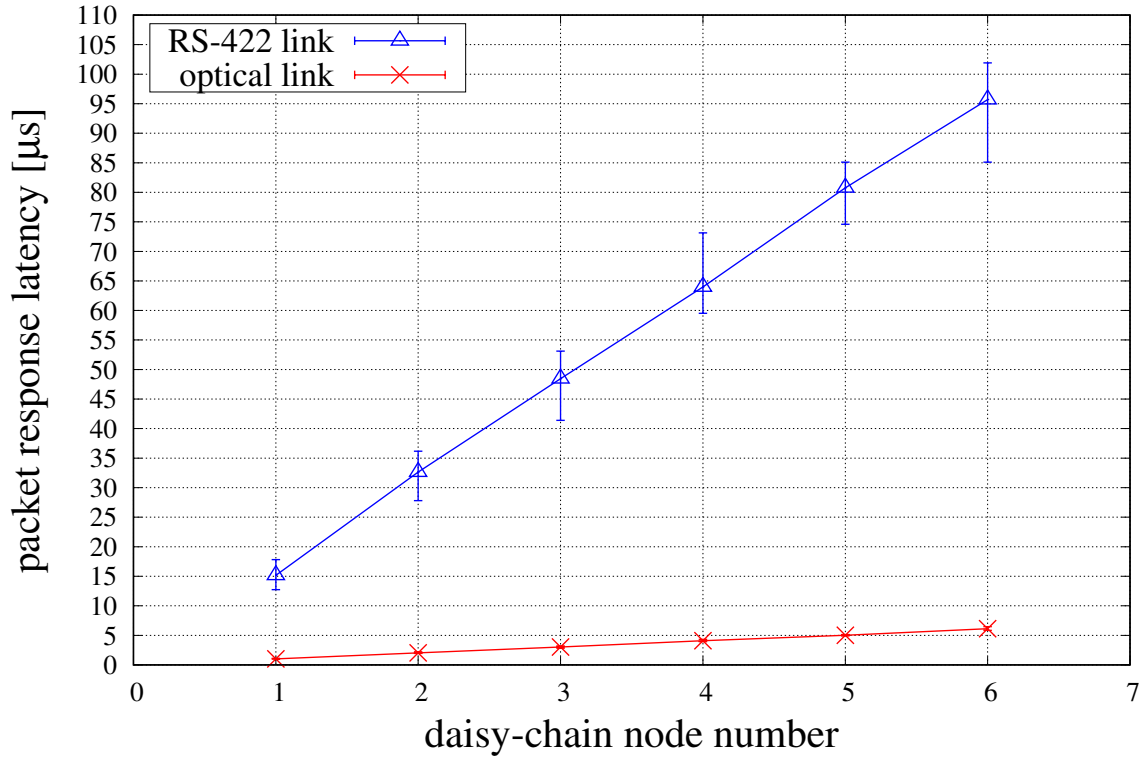
3.6 中枢全身制御系の基幹通信インタフェース

3.6.1 PCI バスを利用した従来実装の課題

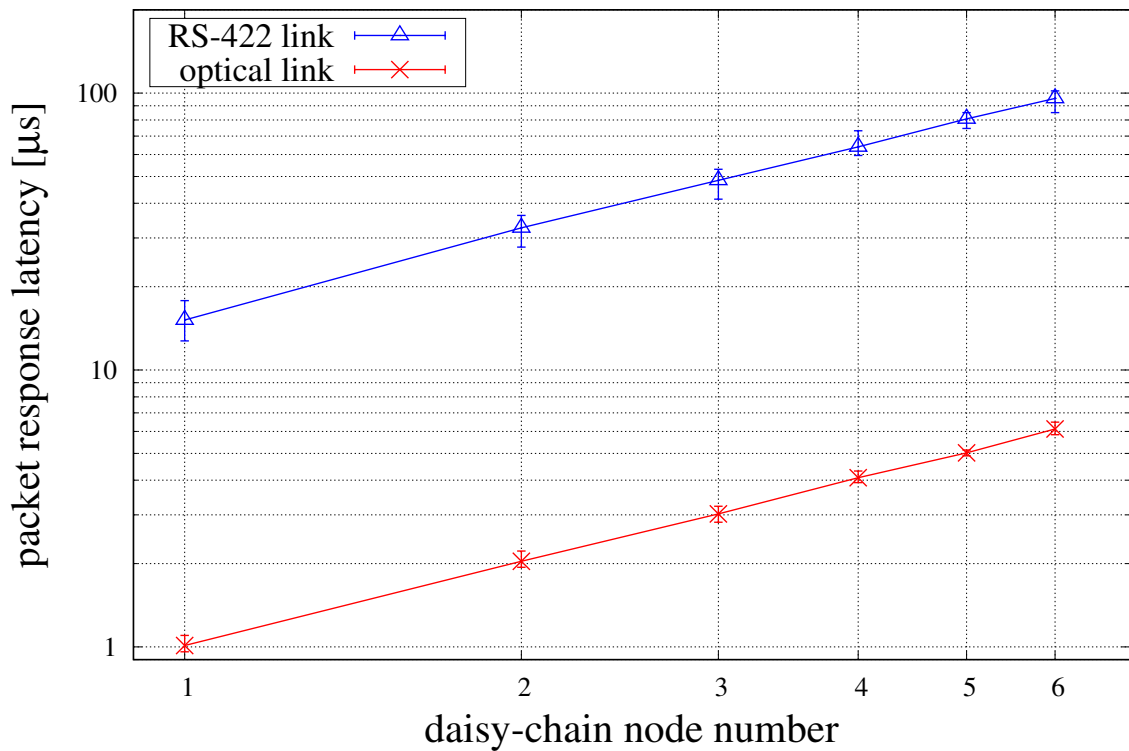
2000 年代から 2010 年代前半頃のヒューマノイドでは，その基幹通信インタフェースとして PCI バスがよく利用されていた [57][74][153][8]．3.3.1 項で紹介したギア駆動型大出力ヒューマノイドに関する先行研究でも，体内分散ネットワークと中央計算機が PCI バスを介して接続されていた．

中央計算機側には PCI バスを介して FPGA 内部のアドレス空間にアクセスするための独自のデバイスドライバが必要となる．Linux の場合はカーネルモジュールとして実装される．

PCI バスを利用した中央計算機接続インタフェースは実装上の限界を迎えている．現在では PCI Express に置き換えられた旧規格となり，PCI バスを標準搭載した計算機製品は非常に少ない．旧規格となった点以外にも，PCI バスを利用した構成には次のような

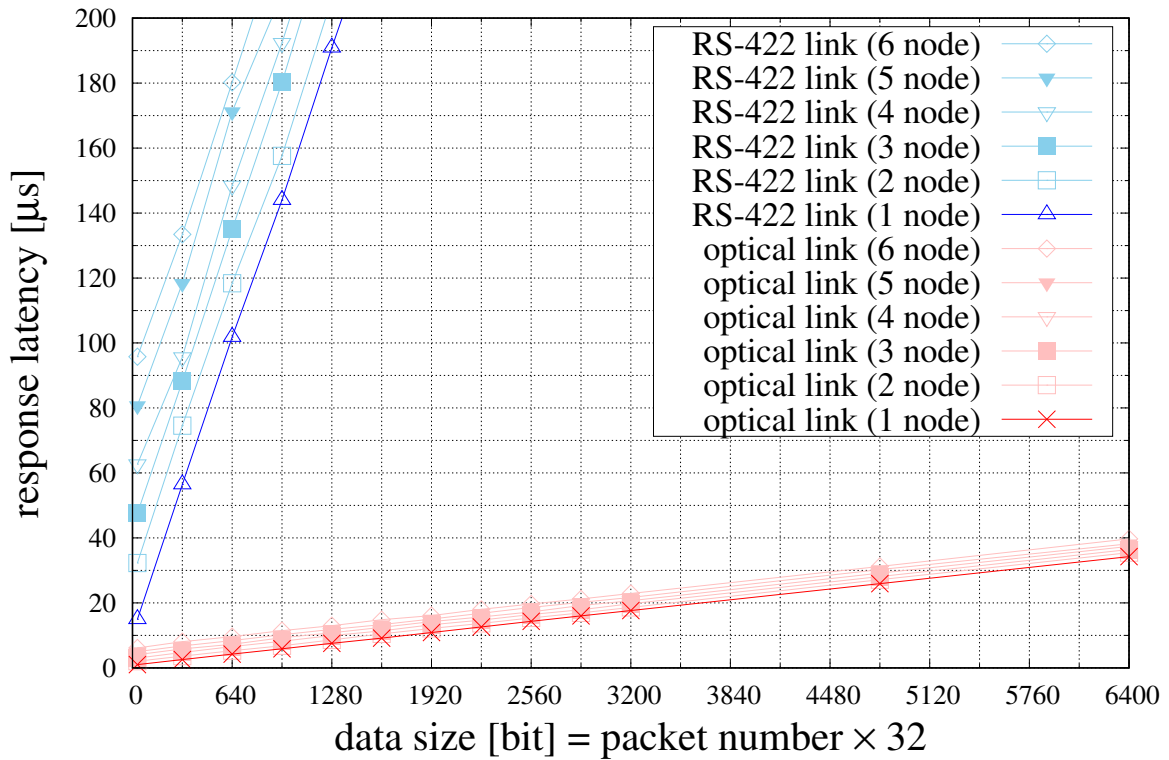


(a) Linear scale view

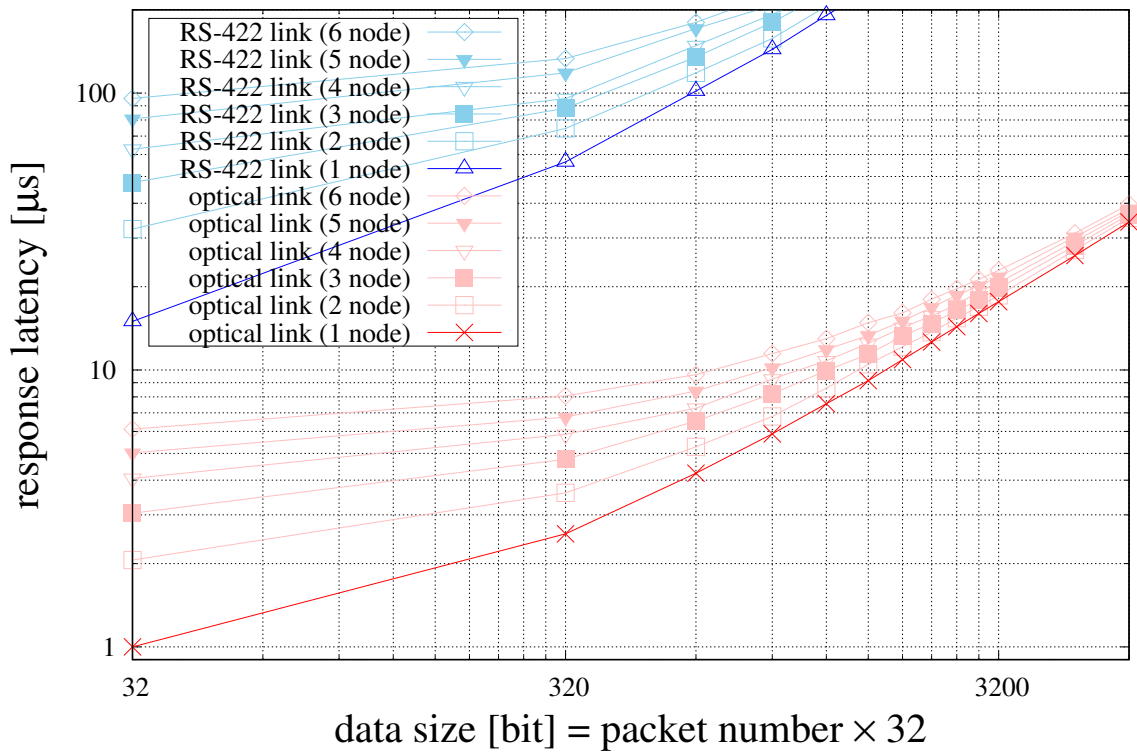


(b) Log scale view

図 3.22 Single packet response latency for optical transmission link and electrical RS-422 link (showing average and min-max error bar of 10 trials)

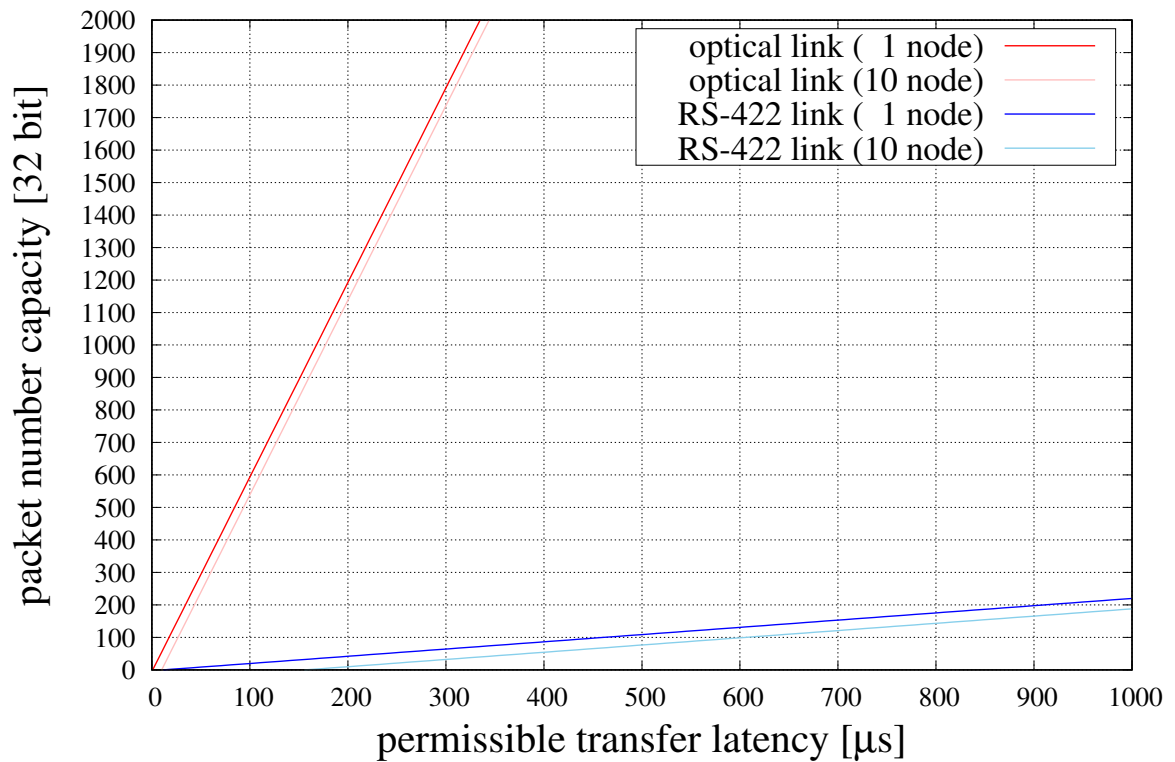


(a) Linear scale view

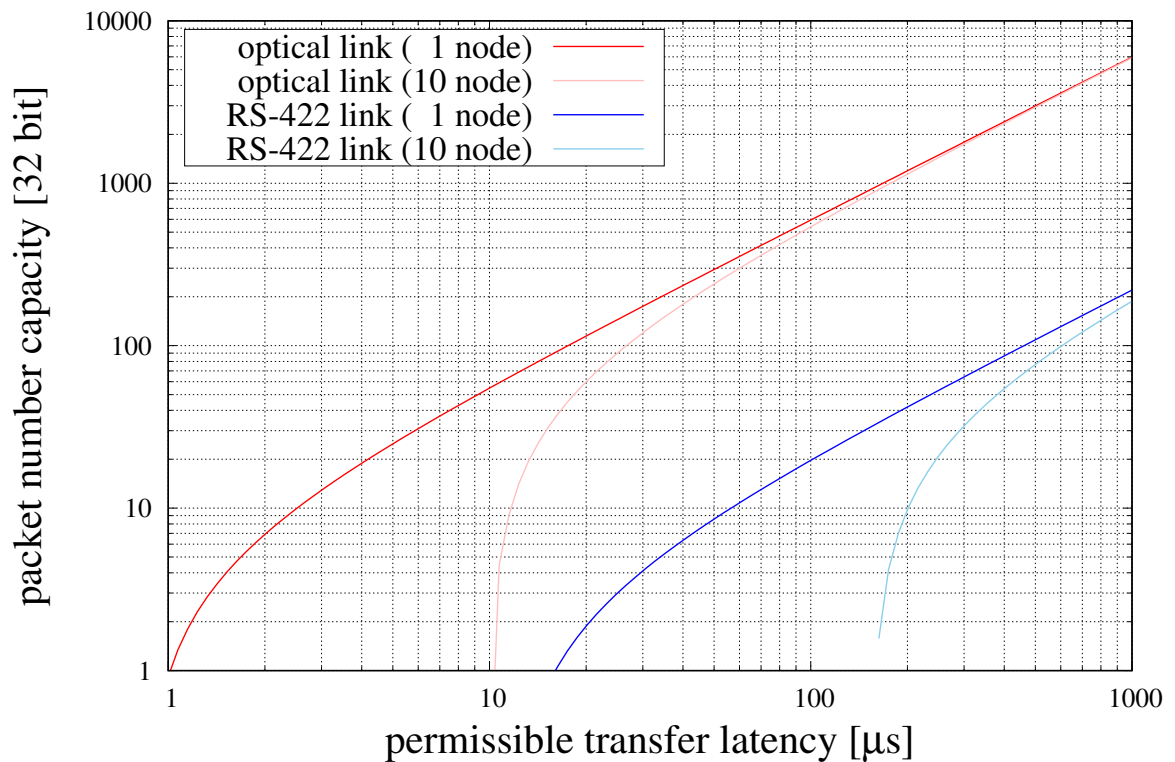


(b) Log scale view

図 3.23 Multiple packet response latency for optical transmission link and electrical RS-422 link



(a) Linear scale view



(b) Log scale view

図 3.24 Identified packet number capacity by permissible transfer latency for optical transmission link and electrical RS-422 link (log scale)

欠点が存在した。

- 中央計算機側が一方向的に双方向バスの制御権を持つため組込計算機を活用して低遅延を目指すデータ処理構造に制約が生じる
- カードエッジコネクタにより振動に弱くロボットの動的な動作実現の支障となる [154]
- PCI スロットを搭載した汎用マザーボード製品に限られるため小型化が困難となる
- ハードウェア構成に応じた独自のデバイスドライバが必要となり中央計算機や OS を更新するたびに保守開発の手間がかかる

これらを考慮しつつ次世代のインタフェースを検討する。

3.6.2 中央計算機の通信インタフェース

PCI バスを利用した中央計算機接続インタフェースの問題を解決するため、次世代の通信インタフェースを検討する。

現在の計算機製品には 2.5.2 項で表 2.5 に挙げたような通信インタフェースが搭載されている。多くの小型計算機での入手性を考慮すると、PCI Express・Ethernet・USB が候補に挙がる。それぞれのメリット・デメリットを考察する。

- PCI Express
 - メリット
 - * PCI と比べて上位互換なシステム
 - * 多くの計算機システムで基幹に近い最速のインタフェース
 - * 大量データのスループットと細かいデータアクセスの両方にパフォーマンスを発揮
 - * mini PCIe や M.2 などコネクタ次第で小型・頑丈
 - デメリット
 - * 独自デバイスドライバの保守開発が必要でソフトウェア上の扱いにくさは PCI 同様
 - * 搭載状況や形態が計算機製品によって大きく異なる
- Ethernet
 - メリット
 - * 大量データをバースト転送する際に大きなスループットを発揮
 - * LAN 用のポートとして現在の計算機製品で比較的入手性が高い
 - * LAN ポートが余っていない場合は USB 等に変換して利用することも

容易

- * コネクタが頑強
- * OS の標準 Socket API によりメンテナンスフリー実装が容易

– デメリット

- * パケットあたりのオーバーヘッドが大きく細かい単位でのデータ転送に不向き

• USB

– メリット

- * 細かいデータアクセスの応答性が高い
- * 多数のデバイスを接続したネットワーク構築が容易
- * 現在の計算機製品で非常に入手性が高い
- * 汎用デバイスドライバによりメンテナンスフリー実装が容易

– デメリット

- * 5 Gbps SuperSpeed 以上ではロボット体内での高信頼な運用の実現性が低い
- * 480 Mbps High-Speed 以下では大量データのスループットが高くない
- * 厳密なコネクションに基づいており途切れた際の復帰が容易でない

本研究ではこれらのインタフェースを適材適所に利用する。

3.6.3 低遅延基幹通信インタフェースの小型高信頼化設計

中央計算機と接続する基幹通信インタフェースの設計においても、3.4.1 項の組込分散ネットワークと同様に FPGA を活用した自由自在な通信プロトコル実装が効果を発揮する。中央計算機として既製品の高性能計算機製品を利用するにあたり、基幹通信インタフェースの設計には、高性能計算機製品の実装仕様による先行的な制約が生じる。その時代の計算機に搭載される通信インタフェースの性能を十分引き出して低遅延なプロトコルを実装する継続的な開発が求められる。これは最新の計算機技術を交えた高性能なロボット体内制御システムを利用してゆくための基盤となる。FPGA を活用することで、基幹通信インタフェースの設計に先行する制約を中央計算機側の最小限なものにとどめて、その性能を引き出すことが可能となる。

最終仕様が既に決まっている用途の場合は、その仕様を実装した専用の半導体製品を利用することでサイズや消費電力を抑えることができる。一方で、ロボットの継続的な開発においては、都度発生する要求に応じて通信インタフェースの仕様を更新できる態勢を整えておくことが重要である。高度な環境適応行動実現を目指して、通信データフォーマッ

トや通信項目の変更・追加が頻繁に発生しうる．FPGA を活用したプロトコル実装は再構成が容易であるため，ロボット研究開発に適している．

低遅延・低ジッタな通信プロトコル実装を小型パッケージに収めることを考えると，通信インタフェースの物理層を担う最小限の周辺回路を FPGA に接続したハードウェア構成を採用することが最適な実装手法となる．FPGA では任意の実装により物理層を操作できるため，ロボットの制御に向けて遅延およびジッタを小さく抑えたプロトコルを構成することができる．周辺回路が最小限のものとなるため小型化にも貢献する．FPGA に実装すべき機能が多くなり開発コストは大きくなるが，ロボットの研究開発においてはプロトコル実装の妥協を排除できる利点が開発コストの欠点を上回る場合が多い．

計算機システムで標準的に利用される通信規格では，多くの場合，物理層のバス接続インタフェースにも標準的な規格が存在する．例えば，現在の計算機に搭載される Ethernet の場合，MAC 層と物理層の接続において MII およびその派生インタフェースが標準規格化されている．MII 派生インタフェースには，Gigabit Ethernet に対応した GMII や，更に配線数を減らした RGMII などが存在する．物理メディアインタフェースと MAC 側の MII 派生インタフェースを備えた Ethernet 物理層の半導体デバイスも製品化されている．USB 2.0 の場合では，UTMI およびその派生インタフェースが標準規格化されている．UTMI 派生インタフェースには，USB OTG に対応した UTMI+ や，更に配線数を減らした ULPI などが存在する．このようなインタフェースを搭載した物理層デバイスを周辺回路として FPGA にバス接続することで，物理層の提供する通信規格の全機能に対して FPGA からのアクセスが可能となる．ロボットの体内制御システムに最適な低遅延・低ジッタな通信プロトコルが実装でき，また物理層の異常状態などに関する情報が隠蔽されないため異常対応による高信頼化も果たせる．

3.7 Ethernet を利用した低遅延高信頼基幹通信ネットワークの設計と実装

3.7.1 EtherCAT によるプロトタイプ実装

中央計算機との間の基幹通信インタフェースで PCI バスを置き換えるものとして，信頼性，汎用性，および保守性を重視して Ethernet に着目する．中央計算機と組込制御系の間で Ethernet インタフェースを利用してデータを交換する手法として，利用実績の多い EtherCAT による実装を試みる．EtherCAT は近年のいくつかのヒューマノイドでも利用されている [81][36][39][93]．

PCI 中継基板を置き換える実装として，EtherCAT インタフェースを備えた通信中継モジュールを開発した．図 3.25 に EtherCAT 中継モジュール ECB-02 の概観を示す．

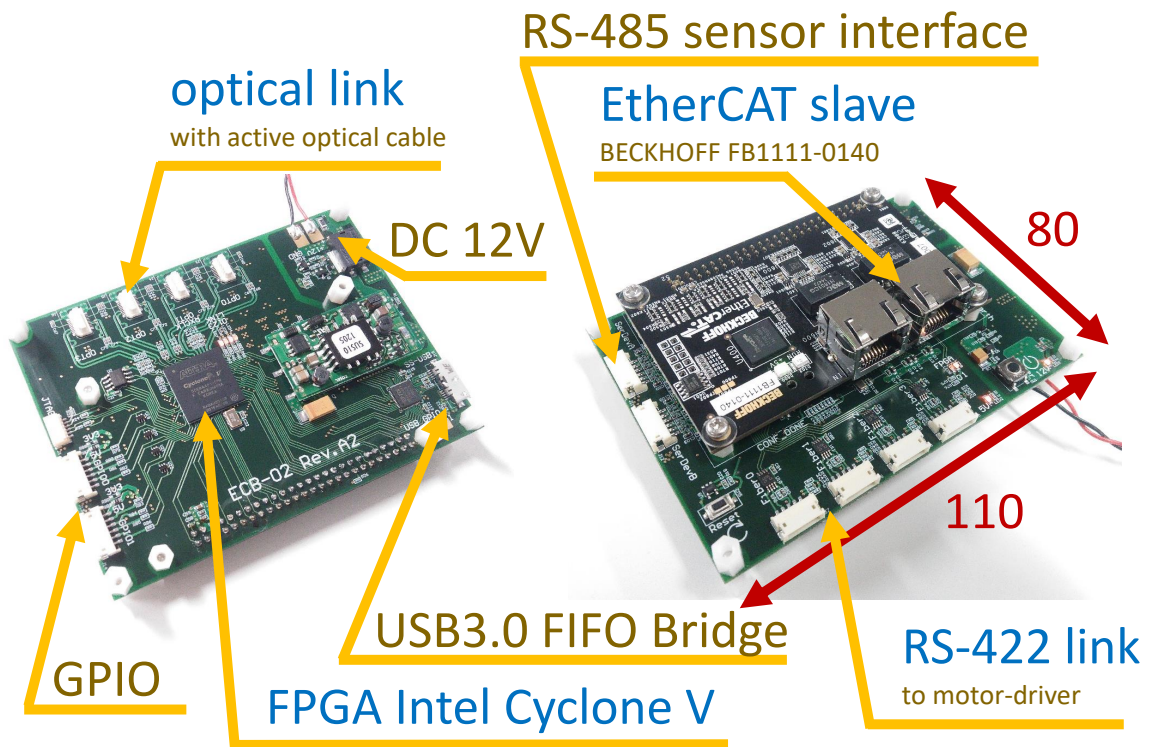


図 3.25 Overview of the EtherCAT bridge module ECB-02

Beckhoff Automation 社の EtherCAT スレーブモジュール FB1111-0140 に独自の通信変換基板を接続した構成となっている。Ethernet 化のプロトタイプ段階で、実績の多い EtherCAT を一般的な利用方法で試す目的から既製品の EtherCAT スレーブモジュールを用いた。

EtherCAT スレーブモジュール FB1111-0140 は同 Beckhoff Automation 社の EtherCAT スレーブコントローラ ET1100 を搭載したモジュールである。EtherCAT は、Ethernet パケット上に表現されるコマンドによってスレーブ内部のアドレス空間を読み書きするプロトコルである。EtherCAT スレーブモジュールは内部にデュアルポートメモリを実装している。任意の目的に利用可能なメモリは 128 Byte の User RAM と 8192 Byte の Process RAM から構成される。マスタとなる中央計算機からは EtherCAT パケットを利用してメモリにアクセスすることができる。一方で、デュアルポートメモリの片側ポートは 16 bit アドレス・16 bit データのバスとして EtherCAT スレーブモジュールの外部ピンに提供される。

ECB-02 には Intel 社の Cyclone V E FPGA 5CEBA4F17C6N が搭載されている。EtherCAT スレーブモジュールから提供されるバスを FPGA に接続し、FPGA から

EtherCAT スレーブモジュール内部のメモリにアクセスする．これにより中央計算機との間で 32 bit 単位のデータパケットを交換し，これを RS-422 分散ネットワークに中継する．

パケット交換のシステム構成を図 3.26 に示す．中央計算機は 32 bit データパケットを EtherCAT パケットに載せて，EtherCAT スレーブモジュール内部の Process RAM に読み書きする．一方で，User RAM は接続やコンフィグなどの管理に利用する．8192 Byte の Process RAM アドレス空間に対して，前半 4096 Byte をダウンリンク領域，後半 4096 Byte をアップリンク領域として割り当てる．中央計算機から分散ネットワークに宛てたコマンドパケットはダウンリンク領域に書き込み，分散ネットワークからのレスポンスパケットはアップリンク領域から読み出す．FPGA 側では，ダウンリンク領域からコマンドパケットを読み出して分散ネットワークに送信し，分散ネットワークから受信したレスポンスパケットをアップリンク領域に書き込む．ダウンリンク領域は全 4096 Byte のリングバッファの形式で利用する．中央計算機からは前回送信時の続きから書き込んで，終端を表す特殊パケットによって現在の最終位置を示す．一方でアップリンク領域は，中央計算機で事前に受信パケット量を知ることができないため，固定長のバンクに分けて，一度にバンク全域を読み出す．FPGA は中央計算機からの特殊パケットによる指示に従い，書き込み先のバンクを切り替える．

EtherCAT 中継モジュールを利用したネットワーク接続例を図 3.27 に示す．中枢制御を担う中央計算機からは Ethernet パケットに載せた EtherCAT プロトコルによって EtherCAT 中継モジュールとデータを交換する．EtherCAT 中継モジュールは中央計算機と交換したデータを RS-422 信号によるデジチェーン接続の体内分散ネットワークに中継する．

PCI から EtherCAT に置き換えたことで従来抱えていた問題がいくつか解決される．PCI のカードエッジコネクタによる接続がなくなり，代わりに頑強で高信頼な Ethernet コネクタを利用することで，ロボットの行動実現における耐衝撃性が向上する．これにより高速な歩行や跳躍着地を含むような動作を行った際にシステムが停止する可能性が低減される．大型のカードエッジコネクタが不要となったことにより，最新の小型計算機を採用することが可能となる．Ethernet は多くのシステムで利用できるため，高度なロボット制御に向けて多様な計算機環境を試すことが可能となる．PCI で必要となっていたデバイスドライバの保守開発が不要となり，標準 Socket API を利用した汎用性の高いソフトウェアが利用可能となる．この点も多様な計算機環境を試す上での一翼を担っている．

3.7.2 高速大容量な Gigabit Ethernet 通信インタフェースの設計

3.7.1 項で導入した EtherCAT による基幹通信インタフェースは多くの課題を解決した

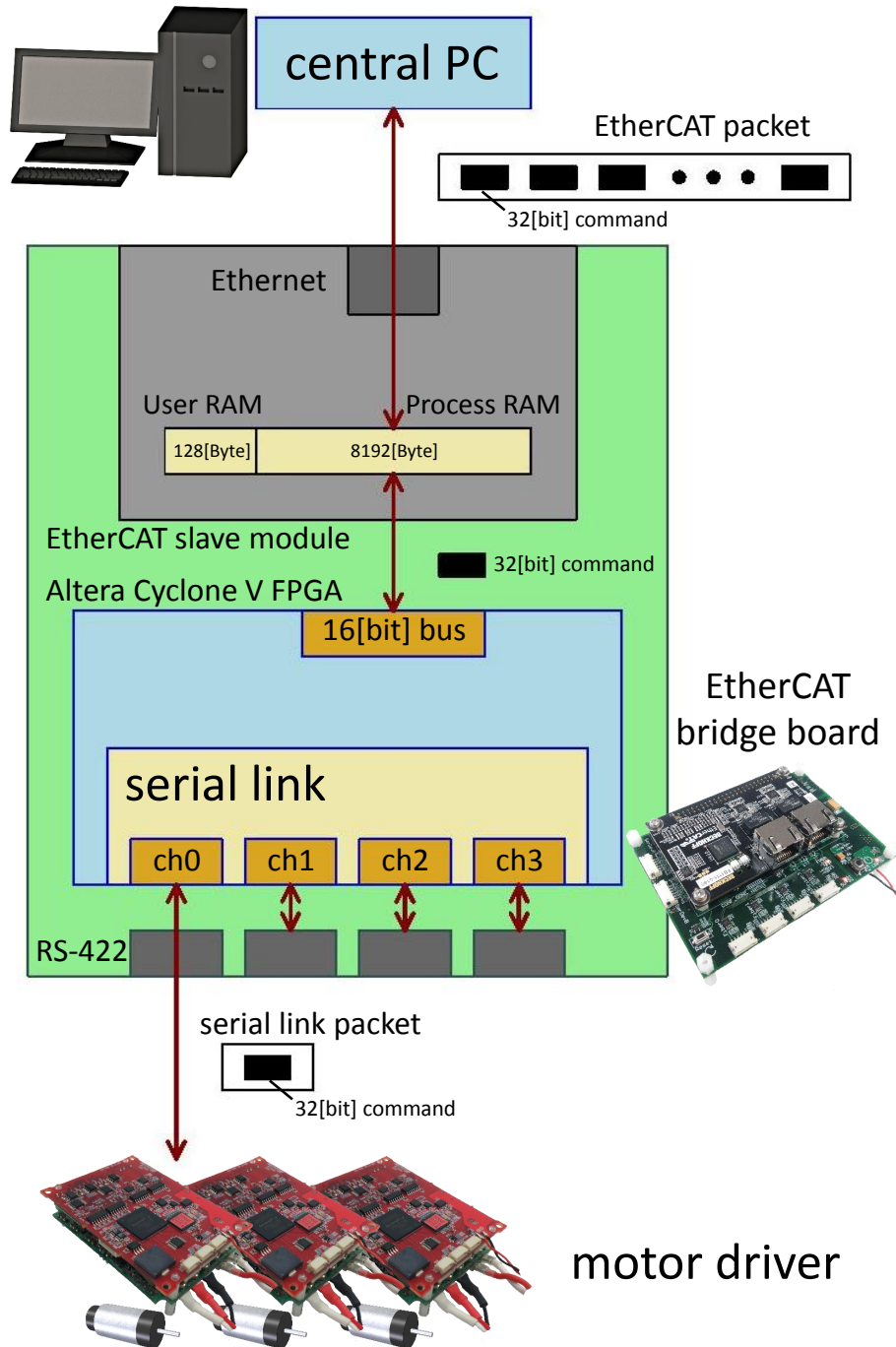


図 3.26 Data transfer system around the EtherCAT bridge module ECB-02

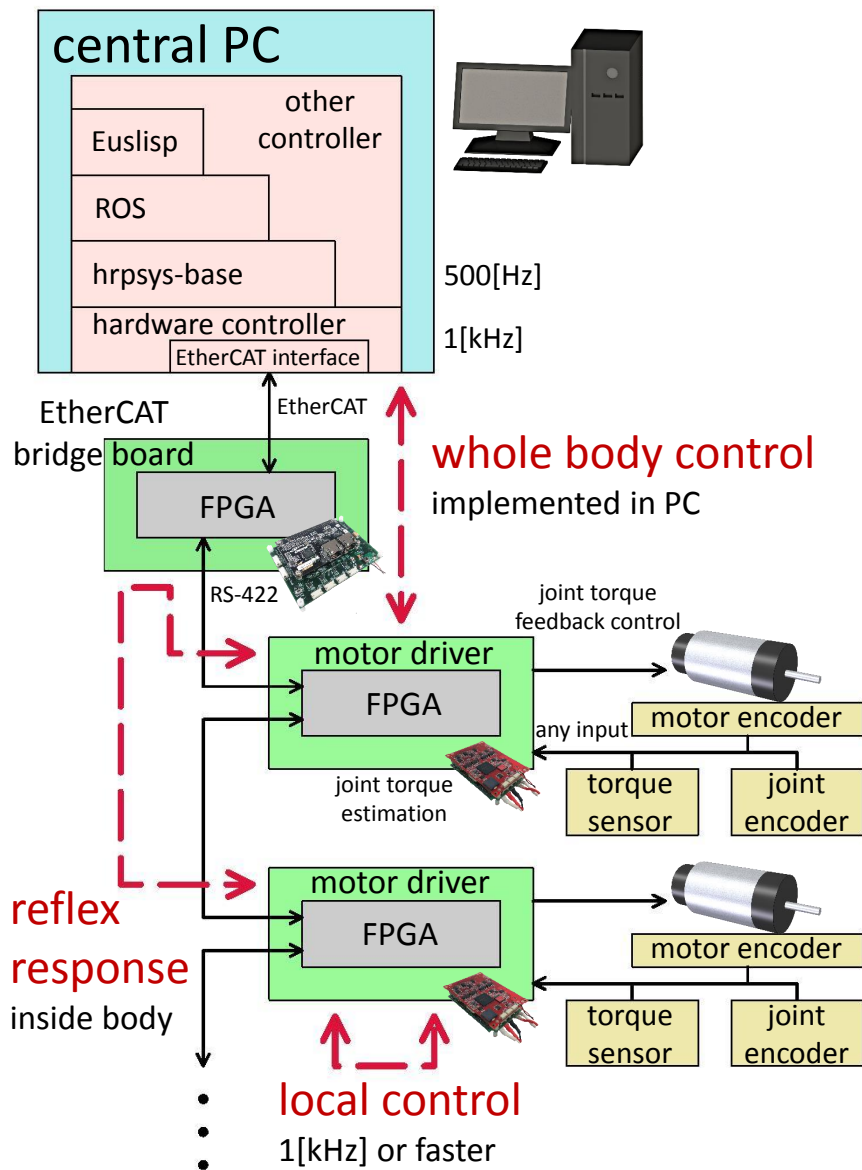


図 3.27 Network system overview with the EtherCAT bridge module

一方で、まだいくつか実用上の課題を残している。

これまで EtherCAT の実装は 100 Mbps に限られていた。3.2.3 項で試算した時間あたりの転送データサイズ 49.44 Mbps と比較すると、理想的な 100 Mbps の最大スループットを発揮した場合でも制御周期に対して無視できない時間を通信に割くこととなる。実際にはソフトウェアのオーバーヘッドなどもあり更に大きな時間が必要となる。なお、このような需要を受けてか、近年では 1 Gbps 以上の Ethernet を利用した EtherCAT G 規格が開発されている。今後の技術動向には期待が高まる。

また既製品の EtherCAT スレーブモジュールを利用した実装による制約もある。EtherCAT スレーブモジュールはデータ交換に利用する主要メモリを 8192 Byte だけ搭載している。このメモリを領域分割した双方向のデータ交換実装により 1 周期あたりのデータ量は 2048 Byte，すなわち 16384 bit に制約されている。3.2.3 項でも述べた通り，これでは所望のデータ量に到達していない。また，EtherCAT スレーブモジュールと FPGA の間は 16 bit アドレスと 16 bit データのバスで通信を行う。バスは 50 MHz のクロックで駆動され，データ転送レートは最大でも双方向の合計で $16 \times 50 = 800$ Mbps となる。実際にはデータ読み出しごとの待ち時間も発生し，これより数倍遅くなる。バスの速度も律速に近いものとなっている。

Ethernet の利用による信頼性や保守性の恩恵をそのまま受けつつ通信データサイズや転送遅延の問題を解決すべく，既製品による EtherCAT 実装から 1 Gbps の高速 Gigabit Ethernet に置き換えることを考える。そこで組込中間層計算機モジュールでは，Gigabit Ethernet の利点を最大限に活かすために FPGA に Gigabit Ethernet 完全準拠の PHY を直接接続する構成を採用した。

FPGA には図 3.28 に示す物理層コントローラを構成し中央計算機との間の通信を実装する。既製品のモジュールによる制約を受けず Gigabit Ethernet のデータ転送レートを最大限発揮するアプリケーション構成を FPGA 内部で自由自在に実現できる。

EtherCAT のプロトコルでは，データの読み書きのそれぞれでデータサイズ分のパケットを往復させる方式が採用されている。多数のノードが接続されたフィールドバスでパケットを循環させる上で妥当なプロトコルであるが，本研究のように Ethernet を中央計算機と組込制御系のインタフェースとしてのみ利用する場合には些か冗長性に過ぎる。FPGA によるカスタマイズ可能な構成では，プロトコルも自由に設定することができる。送られてきたパケットに対する応答を，組込制御系の状況を鑑みて最適な方式で返送することができる。これは，中央計算機側が一方的にバスの通信タイミング制御権を持つ PCI に対する利点でもある。

3.7.3 Ethernet 中継モジュールの小型化実装

既製品の EtherCAT スレーブモジュールを利用する必要がなくなったことにより Ethernet 中継回路の小型最適化実装が可能となる。多様な形態の多自由度ロボットへ適用可能とすることを目指し，小型かつ比較的安価なパッケージとして Ethernet 中継機能を実装する。

Ethernet と RS-422 分散ネットワークを中継する小型 Ethernet 中継基板を開発した。図 3.29 に小型 Ethernet 中継基板 ENB-04 の概観を示す。

ENB-04 には RGMII インタフェースの Gigabit Ethernet PHY である Microchip

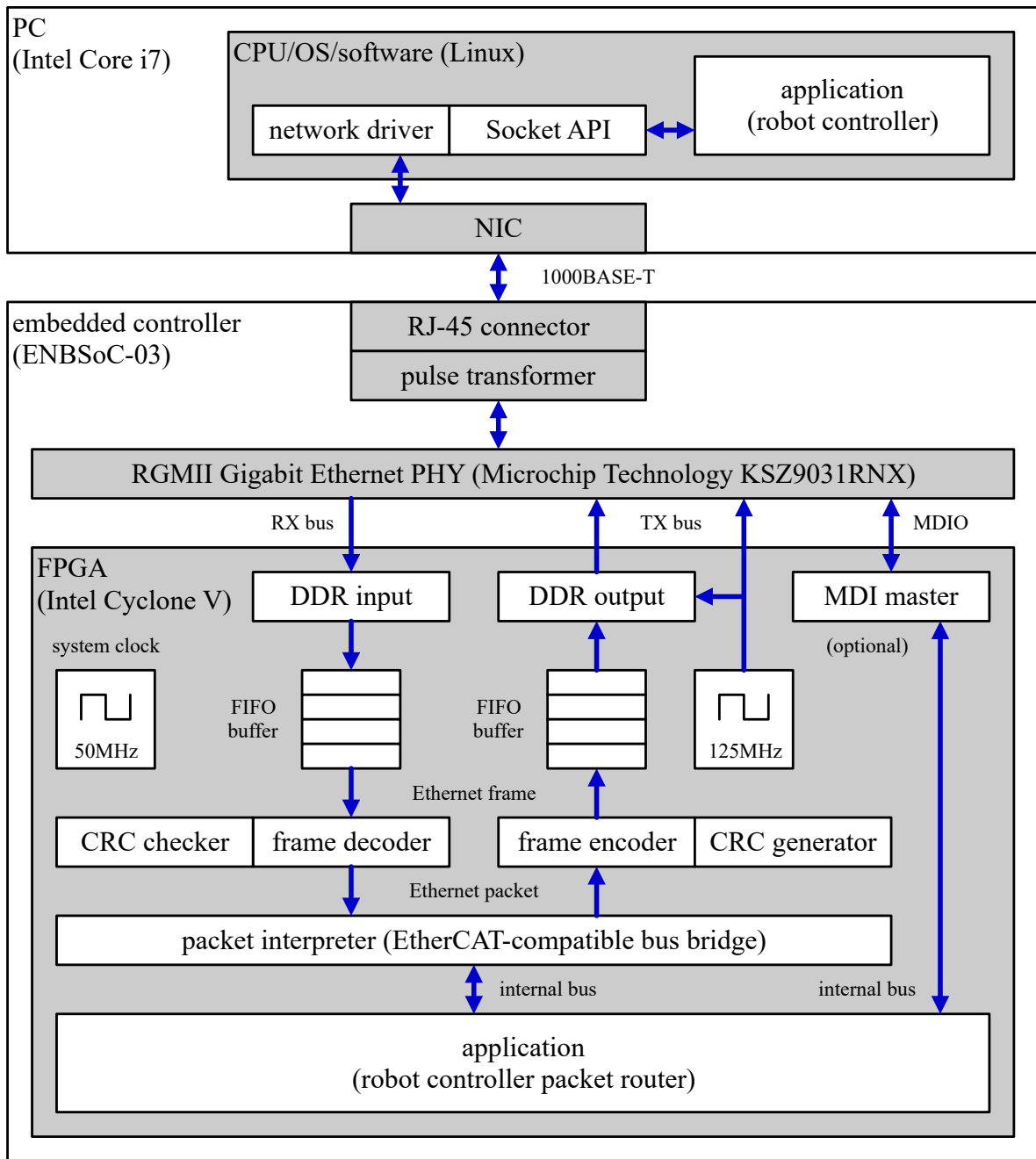


図 3.28 Physical layer controller architecture for Gigabit Ethernet

Technology 社の KSZ9031RNX を利用した Ethernet インタフェースが 2 ポート用意されている。Intel 社の Cyclone V E FPGA 5CEBA4F17C8N を用いて通信中継を実装する。

図 3.28 に示した EtherCAT 互換のプロトコルを実装する。EtherCAT スレーブモジュールを用いた実装で問題となっていた、データ転送レート・メモリ容量・メモリアク

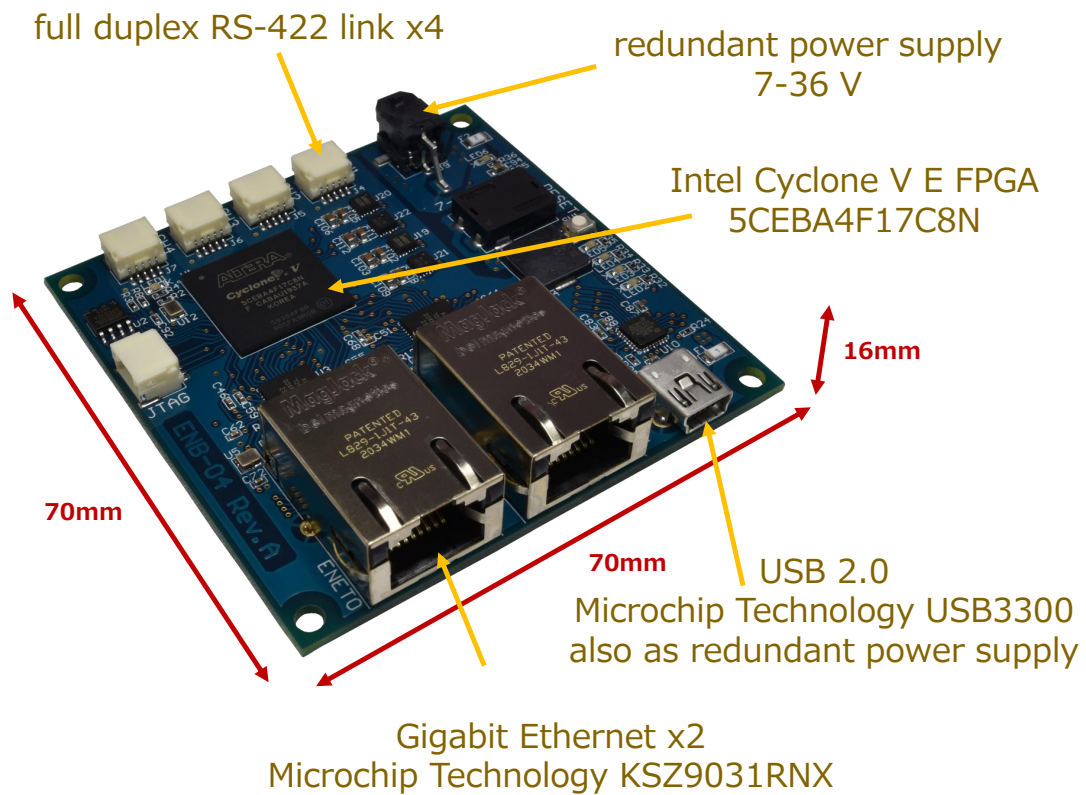


図 3.29 Overview of the Ethernet bridge board ENB-04

セス速度が改善される。Ethernet インタフェースを 2 ポート搭載したことで、

- 複数の Ethernet 中継モジュールをダイジチェーン接続
- Ethernet LAN にバイパス接続することで中央計算機の 1 ポートに制御通信と LAN 通信を共存
- 2 個の計算機から同時接続

といった応用が可能となっている。

Ethernet 中継モジュールに光通信インタフェースを搭載することで、小型光通信モータ制御モジュールを利用した高速光通信分散ネットワークとの接続に対応する。図 3.30 に小型 Ethernet 光通信中継基板 ENB-05 の概観を示す。高速トランシーバユニットを搭載した Intel 社の Cyclone V GX FPGA 5CGXFC4C6U19I7N により高速光通信を実装する。

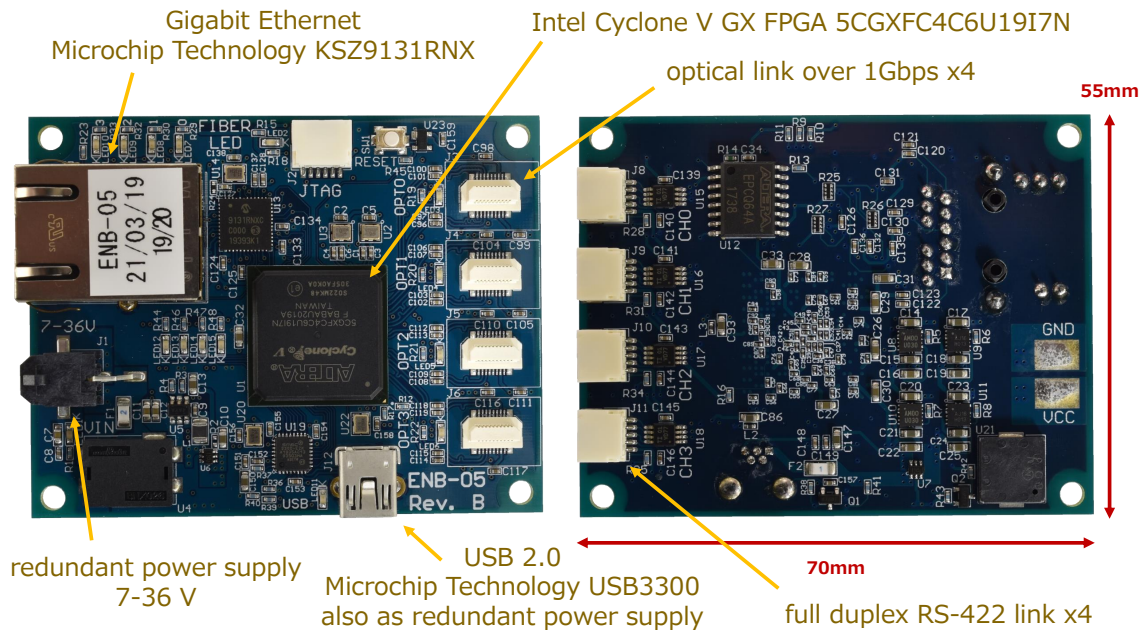


図 3.30 Overview of the Ethernet bridge board ENB-05 with optical link

3.7.4 Gigabit Ethernet 通信インタフェースの性能評価

Ethernet のデータ転送レートを 100 Mbps から 1 Gbps に向上したことによる通信遅延低減効果を実証評価する。

組込中間層計算機上モジュールに 100 Mbps の 100BASE-TX および 1 Gbps の 1000BASE-T の Ethernet 物理層コントローラをそれぞれ実装する。EtherCAT のような往復型のプロトコルを想定して、受信したパケットをストアアンドフォワード方式によりループバック返送する仕様のアプリケーションを構成する。

組込中間層計算機モジュールに現在の一般的な計算機製品を接続して、Ethernet パケットの実用的なループバック遅延を測定する。実験には、Intel 社の CPU Core i7-6600U を搭載した Lenovo 社のラップトップ PC T460s および Ubuntu 18.04.2 LTS / Linux 4.15.0-45-generic の OS を利用する。Linux の Socket API を利用して、send 命令によるパケット送信から recv 命令によるパケット受信完了までの実行時間を測定する。

いくつかのペイロードサイズ設定ごとに 10000 回ずつ試行した結果を図 3.31 に示す。ここでペイロードサイズは、14 Byte の標準 Ethernet ヘッダの後に続く、フレーム最後尾 FCS の直前までのデータサイズを指す。

ペイロードが 0 Byte の場合は 100 Mbps と 1 Gbps で実行時間が大きく変わらない。1 Gbps の場合でおよそ 240 μ s の時間を要しており、このほとんどが標準的な Linux OS

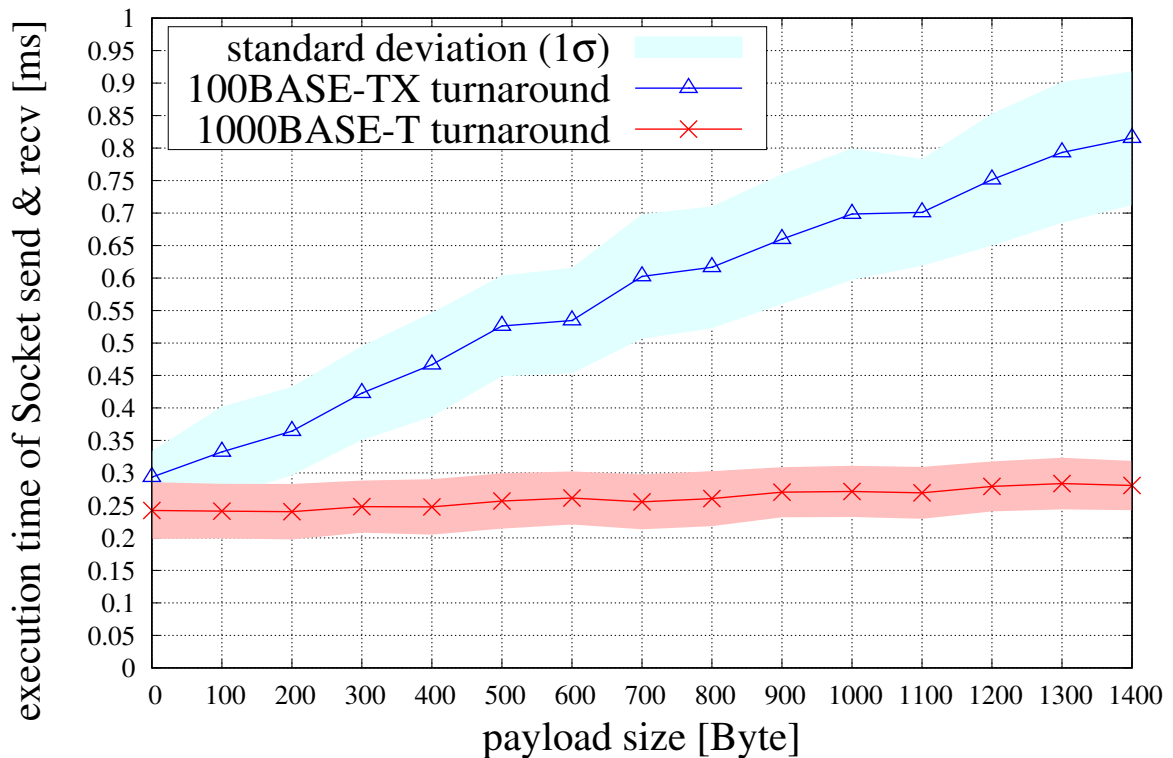


図 3.31 Execution time of Ethernet packet turnaround with Linux PC and Socket API (showing average and standard deviation of 10000 trials)

で Ethernet デバイスドライバおよび Socket API を利用する場合のソフトウェアオーバーヘッドによるものと考えられる。ペイロードサイズが増加すると、100 Mbps の場合では時間が大きく増加するのに対し、1 Gbps では 1400 Byte ペイロードによる増加をおよそ $40 \mu\text{s}$ に抑えている。

標準 Ethernet のパケットサイズはおよそ 1500 Byte の MTU に制限されており、これを超えるデータサイズは通常複数のパケットに分割されて送信される。send 命令から recv 命令までの送受信を 1 単位としてこれを繰り返すとその回数分だけソフトウェアオーバーヘッドが累積されるが、順序に工夫を加えて、複数のパケットをまとめて複数回の send 命令により送信してから同じ回数の recv 命令により受信すると、オーバーヘッドは最小限に抑えられる。

例として、計 5000 Byte のデータ送受信を想定して、1000 Byte の send 命令および recv 命令の組み合わせを 5 回繰り返すと、実行時間は $1195 \mu\text{s}$ となった。一方で 5 回の send 命令を実行した後に 5 回の recv 命令を実行する手順では、実行時間が $287 \mu\text{s}$ に抑えられた。

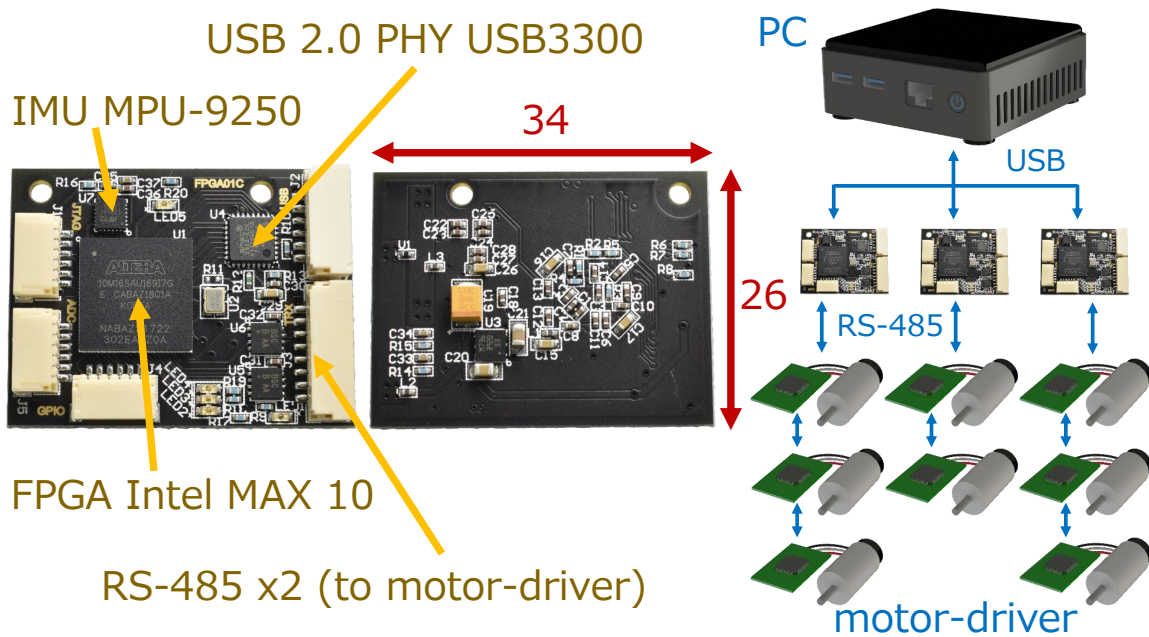


図 3.32 Overview of the USB 2.0 controller board FPGA01C

3.8 USB 通信ネットワークの低遅延化実装

3.8.1 USB High-Speed 中間制御基板

3.3.2 項に述べた腱駆動型筋骨格ヒューマノイドの，USB Full-Speed を利用した体内分散ネットワークの問題を解決するため，USB High-Speed による実装を試みる．

表 3.8 から，USB Full-Speed を USB High-Speed に置き換えることで，

- データ転送レートが 12 Mbps から 480 Mbps に向上
- 周期通信の最小間隔が 1 ms から 125 μ s に低減
- バルク転送やインタラプト転送による周期通信のデータサイズが 64 Byte から 512 Byte に拡大

といった改善が図れる．従来困難であった 1 ms 周期の目標制御周期の実現可能性が見込まれるとともに，通信データサイズに課せられていた最大 64 Byte という厳しい制約もなくなる．

そこで，腱駆動型筋骨格ヒューマノイドの USB Full-Speed 通信を高速化すべく，FPGA に USB 2.0 完全準拠の PHY を接続して USB High-Speed 通信に対応した中間制御基板を開発した．図 3.32 に USB High-Speed 中間制御基板 FPGA01C の概観を示す．

FPGA01C は小型シングルパッケージの FPGA である Intel 社の MAX 10 FPGA 10M16SAU169I7G により制御される。USB 2.0 規格完全準拠 PHY である Microchip Technology 社の USB3300-EZK を搭載している。USB 2.0 PHY は ULPI インタフェースにより FPGA に接続されている。完全なカスタマイズに対応した USB 通信制御器を FPGA 内に構成することが可能となっている。モータ制御モジュールによる分散ネットワークに接続するための RS-485 通信インタフェースには 2 系統の信号ポートが実装されており、RS-422 全二重ネットワークなどその他の構成との互換性も兼ね備えている。基板上には TDK 社の 9 軸 IMU MPU-9250 が搭載されている。全身に多数分散配置される制御基板に IMU を搭載することで柔軟な身体を有する腱駆動型筋骨格ヒューマノイドの全身運動を計測することも可能となり、新たな身体学習手法開発などへの可能性も開かれる。

FPGA01C におけるデータ接続の実装を図 3.33 に示す。FPGA01C には、中央計算機との USB 通信、モータ制御モジュールとの RS-485 通信、およびプロセッサによるモータ制御計算が実装される。

中央計算機と USB 2.0 PHY を通して接続し、中央計算機による USB パケット通信は ULPI インタフェースで FPGA に中継される。中央計算機の USB ポートでルートハブと直に接続する他、追加の USB 2.0 ハブによる中継を挟んでもよい。FPGA で USB パケットを処理し、アプリケーションとデータを交換する。USB デバイスではデスクリプタと呼ばれる設定レジスタが定義されており、中央計算機からこれを読み出すことで USB デバイスに実装された通信プロトコルなどを確認する。アプリケーションデータやデスクリプタの送受信を含めて、USB パケット通信処理は FPGA に実装した論理回路ブロックで完結している。これにより非常に低ジッタなパケット通信が可能となる。

RS-485 インタフェースを通してモータ制御モジュールによる分散ネットワークと接続する。3.3.2 項で紹介した従来の腱駆動向けモータ制御モジュールと接続する場合は、半二重 3 Mbps の UART 型双方向シリアル通信が用いられる。MAC 層ではプロセッサからの要求に応じて通信を実行する。

FPGA のプログラマブル論理回路に Nios II ソフトプロセッサを展開して、通信データ交換やモータ制御計算を含むシステム管理をソフトウェアにより実行する。FPGA で管理したクロックカウンタにより 1 ms 周期で制御タスクを実行する。中央計算機からモータ回転位置やワイヤ張力の指令値を受け取り、モータを駆動する電流もしくは電圧の指令値を計算してモータ制御モジュールに送信する。プロセッサからは IMU のデータを取得することも可能となっており、中央計算機へ IMU データを送信する。

ロボット体内で多数の USB 2.0 中間制御基板を接続するため、小型な USB 2.0 ハブ基板を開発した。図 3.34 に USB 2.0 ハブ基板 HSHUB の概観を示す。

HSHUB には Microchip Technology 社の 4 ポート USB 2.0 ハブである USB2534-

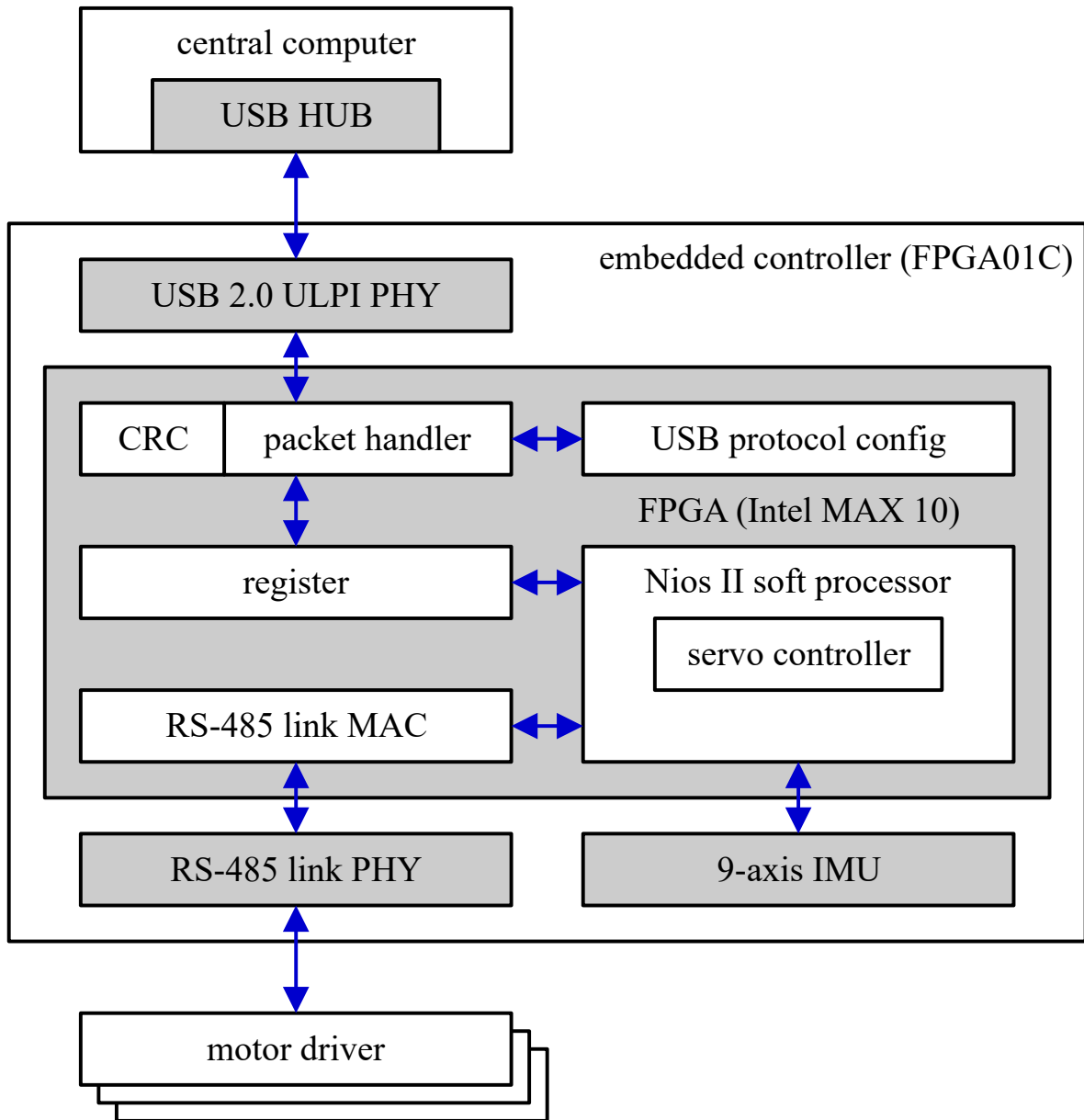


図 3.33 Data connection architecture for the USB 2.0 controller board FPGA01C

1080AEN が搭載されている。接続された USB 2.0 中間制御基板やモータ制御モジュールに電力を供給するための電源電圧生成も担っている。多数の基板に供給する大電流電源を適切なシーケンスで生成するように調整している。外部電源入力は 7-28 V の広範囲な電圧に対応しており、多様なロボット体内の電源環境で適用することができる。

3.8.2 USB 低遅延化実装の性能評価

FPGA による USB High-Speed 実装の性能を評価する。

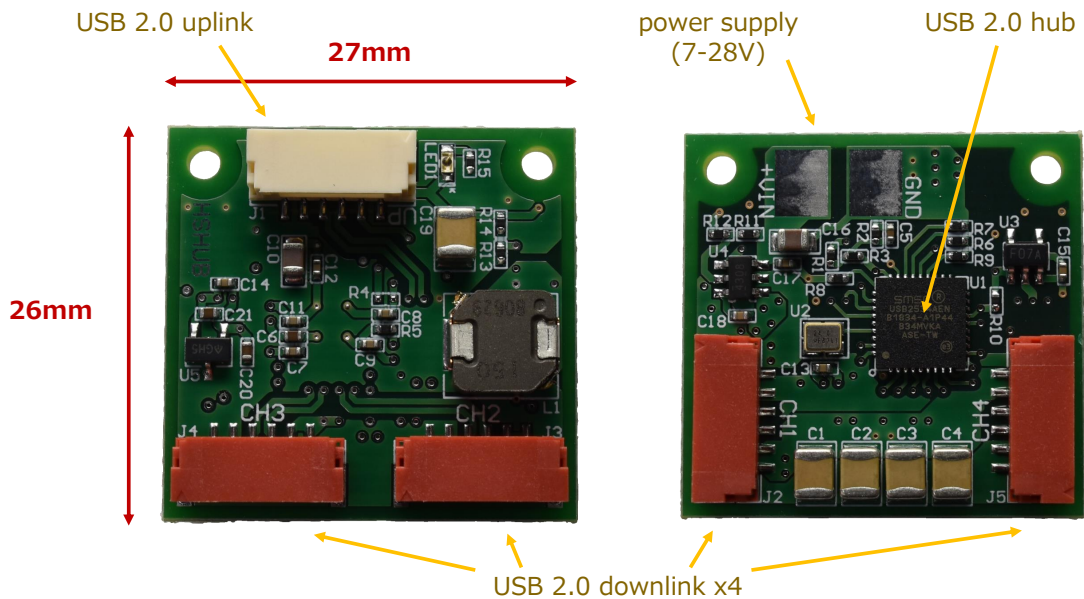


図 3.34 Overview of the USB 2.0 hub board HSHUB

3.3.2 項で紹介した従来の MCU による USB Full-Speed 実装，FPGA による USB Full-Speed 実装，および FPGA による USB High-Speed 実装の 3 条件で，1 パケットのコントロール転送に要する実行時間を比較する．実験には，Intel 社の CPU Core i7-6600U を搭載した Lenovo 社のラップトップ PC T460s および Ubuntu 18.04.2 LTS / Linux 4.15.0-45-generic の OS を利用する．汎用 USB 通信ライブラリ libusb-1.0 を利用し，`libusb_control_transfer` 命令によりコントロール転送を実装する．

いくつかのペイロードサイズ設定ごとに 10000 回ずつ試行した結果を図 3.35 に示す．

同じ USB Full-Speed の実装で比較しても，MCU による実装では 46 Byte のペイロードを送信するのに $593.3 \mu\text{s}$ を要しているのに対し，FPGA による実装では $226.3 \mu\text{s}$ に抑えられ，およそ 2.6 倍高速になっている．FPGA による USB High-Speed の実装で 46 Byte のペイロード送信時間が $102.5 \mu\text{s}$ となっており，FPGA による USB Full-Speed の実装に対しておよそ 2.2 倍，MCU による USB Full-Speed の実装から比べるとおよそ 5.8 倍高速になっている．FPGA による実装では，ペイロードの増加に対する実行時間の増加が小さく，ソフトウェアオーバーヘッドに比べてデータ転送時間が僅かとなっていることが分かる．

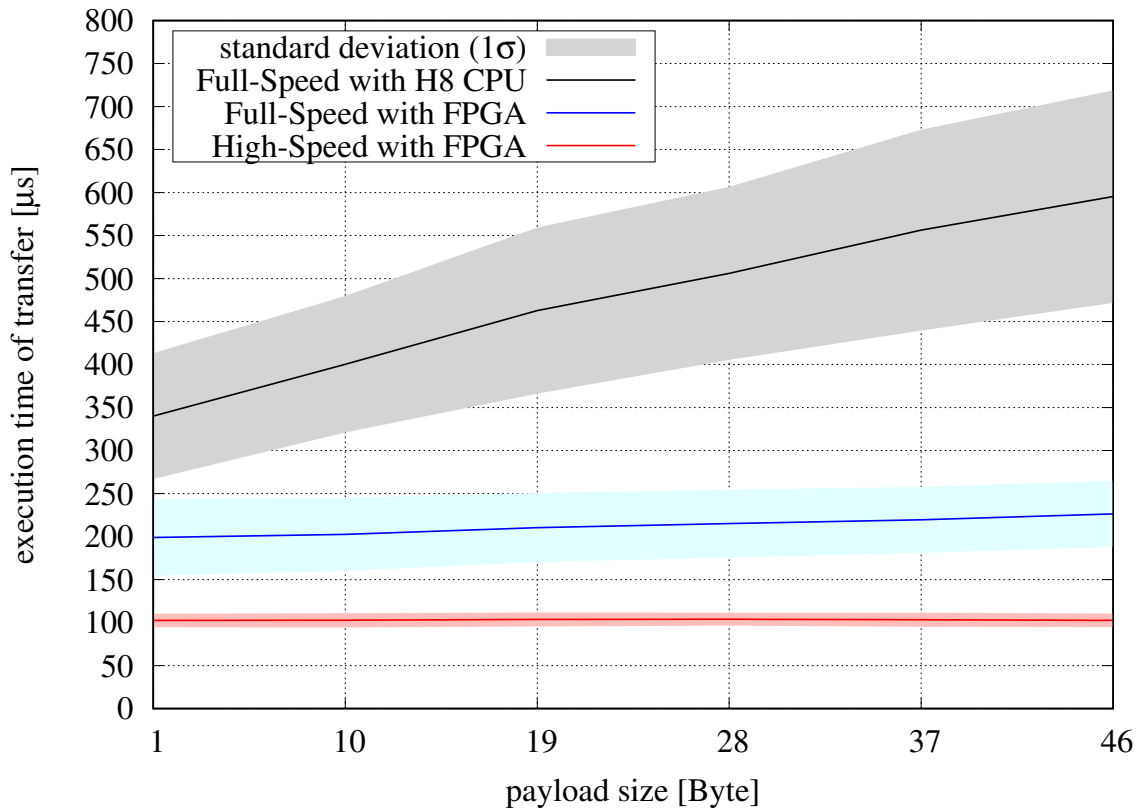


図 3.35 Execution time of single USB transfer (control out) with Linux PC and libusb-1.0, showing average and standard deviation of 10000 trials

3.9 本章のまとめ

本章では、多自由度ロボットの体内分散ネットワークに求められる高速なデータ転送性能に適う電子制御系ハードウェア実装に関して述べた。

ヒューマノイド JAXON の例を用いて体内分散ネットワークに求められるデータ転送性能を試算した。従来の RS-422 汎用シリアル通信および PCI インタフェースによる組込分散ネットワークの律速を分析した。また、とりわけ多自由度な腱駆動型筋骨格ヒューマノイドで採用されていた MCU による USB Full-Speed ネットワークの課題を紹介した。

従来技術の課題を踏まえて、小型ハードウェアによる体内分散実時間ネットワークの実現手法を提案した。

省スペース要求を伴う多様な形態のロボットに適用することを目指して、モータと同程度の幅にベクトル電流制御型モータ駆動回路および高速通信インタフェースを実装した小

型モータ制御モジュールを開発した。従来型の 26.667 Mbps 汎用シリアル通信に加えて、高速シリアル通信技術を活用した 2.5 Gbps の高速光通信を利用して、高速で高信頼な組込分散ネットワークを構成した。実用的なプロトコルでも汎用シリアル通信の数十倍以上に及ぶ高速光伝送ネットワークのデータ転送性能を検証した。

基幹通信ネットワークの設計において、妥協を排除した通信プロトコルを継続的に開発してゆく重要性を確認し、これが FPGA と最小限の周辺回路を組み合わせた柔軟な構成により達成可能となることを述べた。従来利用されていた PCI バスの課題を分析し、PCI Express・Ethernet・USB による次世代通信インタフェースを検討した。

信頼性や汎用性の高い Ethernet を利用した基幹通信インタフェースを設計した。EtherCAT を利用したプロトタイプ開発を通じて、その利点を分析するとともに低遅延通信に向けた課題を見出した。FPGA のプログラマブル論理回路を活用して更に高速大容量化した Gigabit Ethernet 通信を実装し、その効果を検証した。また、USB Full-Speed ネットワークの課題を解決するために、FPGA を利用して性能を最大限に活用した USB High-Speed ネットワークを実装し、効果を検証した。

これらの要素技術は、小型・高信頼な実装によりヒューマノイドをはじめとした多様なロボットでの運用に適うものとなっている。

第4章

階層アーキテクチャ設計に基づく 低遅延全身応答のための組込中間自律系

4.1 はじめに

多自由度ロボットの全身フィードバック動作制御系の遅延を低減するにあたり，体内分散ネットワークの低遅延化に加えて，中枢全身制御系におけるデータ交換や処理の応答性を向上することも重要な要素となる．組込分散ネットワークと密に接続される組込計算機を導入し，組込中間層を交えた全身フィードバック動作制御系を構成することが，より応答的で多彩なロボット行動実現に向けた基盤となる．

本章では，中央高性能計算機に加えて更に低遅延に応答可能な組込計算機を導入した中枢全身制御系の構成を提案する．FPGA や GPU のハードウェアアクセラレータ技術を搭載した組込計算機を，体内分散ネットワークへの低遅延バス接続により導入することで，複数の計算機を交えた相補的なヘテロジニアスアーキテクチャを構成する．これを実現する組込中間層計算機の開発事例を通して，高度な組込計算技術およびネットワーク技術をロボットに導入するために求められる電子制御系ハードウェア実装手法に関して述べる．

4.2 ヘテロジニアス中枢全身制御系による低遅延フィードバック機構

4.2.1 組込計算機を交えたヘテロジニアス中枢全身制御系の構成

ロボット体内分散制御システムに，高速光通信もしくは RS-422 汎用シリアル通信を利用した組込分散ネットワーク，および Gigabit Ethernet を利用した中央計算機の基幹通信インタフェースを適用することで，およそ 1 ms の制御周期目標を満たす性能を有したネットワークが構成される．しかし，中央計算機における実時間制御の低遅延化や信頼性向上に向けては実用上の課題が残されている．

中枢全身制御系は図 2.7 に示したように，

- 総合的な環境認識およびモデル学習
- 高次の行動計画および動作生成
- 実時間のハードウェア通信および動作制御

といった機能を担う．大規模計算と実時間処理が同時に求められる．これらが同一の CPU や OS を共有すると大規模計算のスループットと実時間性の両立に困難が生じる．特に実時間性の求められる動作制御機能は，低遅延・低ジッタに特化した計算機環境で実行することで性能が発揮される．従来より環境認識・行動計画と実時間動作制御を異なる計算機に分けて実装しているロボットも多い．

高性能な中央計算機で Ethernet パケット通信を実装すると、図 3.31 の結果にも見られるように、およそ 1 ms 以下を目標とする制御周期に対して無視できないオーバヘッドが生じる。より低遅延に組込分散ネットワークとデータを交換できる計算機の導入が、実時間性を保証する信頼性向上や、より短周期で低遅延なフィードバック制御の実現に繋がる。

本研究では、高性能な中央計算機に加え、より低遅延・低ジッタなフィードバック制御を可能とする組込計算機を併用して、ヘテロジニアスな中枢全身制御系を構成する。高い実時間性の求められるフィードバック動作制御を組込計算機に任せることで、動作制御の実時間性が高信頼に満たされるとともに、中央計算機のリソースをより高度な計算に振り分けることができる。

組込分散ネットワークと低遅延にデータ交換できる組込計算機を検討する。

組込 CPU と FPGA がパッケージ化された FPGA SoC デバイスが提供されてきている。CPU が FPGA に対して密に接続されており、物理メモリ空間へのアクセスによって直接 FPGA とデータを交換することができる。FPGA を活用した組込分散ネットワークと親和性の高いシステムが構成できる。FPGA によるハードウェアアクセラレーションを活用して制御計算を高速化する応用も可能となる。FPGA は同一構造の回路を再利用する反復計算のアクセラレーションを得意としている。

FPGA SoC として製品化されているデバイスでは得られる演算性能に限られる傾向にある。FPGA SoC では実時間動作制御計算に不足する場合もある。比較的高性能な組込 CPU と GPU がパッケージ化された GPU SoC デバイスも実用化されてきている。GPU による多並列計算のハードウェアアクセラレーションも利用可能であり、大規模なデータの学習や蓄積への応用も期待される。

本研究では、ネットワークにおけるデータ転送遅延を低減するとともに、分散制御ノードにより近い組込計算機にフィードバック制御系を構成することで、多自由度ロボットのフィードバック遅延の低減と環境適応行動の実現を目指す。ロボット体内制御システムの中枢全身制御系において低遅延なデータ転送およびフィードバック動作制御の要となる組込中間層計算機を提案する。中央計算機に加えて組込計算機として FPGA SoC および GPU SoC を導入することでヘテロジニアスな中枢全身制御系を構成する。2.7.2 項で述べた、ヒトの脊髄や小脳に相当する応答的なフィードバック制御系の役割を担う。

4.2.2 体内分散制御システムにおけるフィードバック経路

組込計算機を導入して動作制御系を多階層化するにあたり、所望の外乱応答全身フィードバック動作制御に対する実装が、データ転送および制御計算に関するいくつかの経路から選択可能となる。FPGA SoC および GPU SoC を搭載した組込中間層計算機を用いて

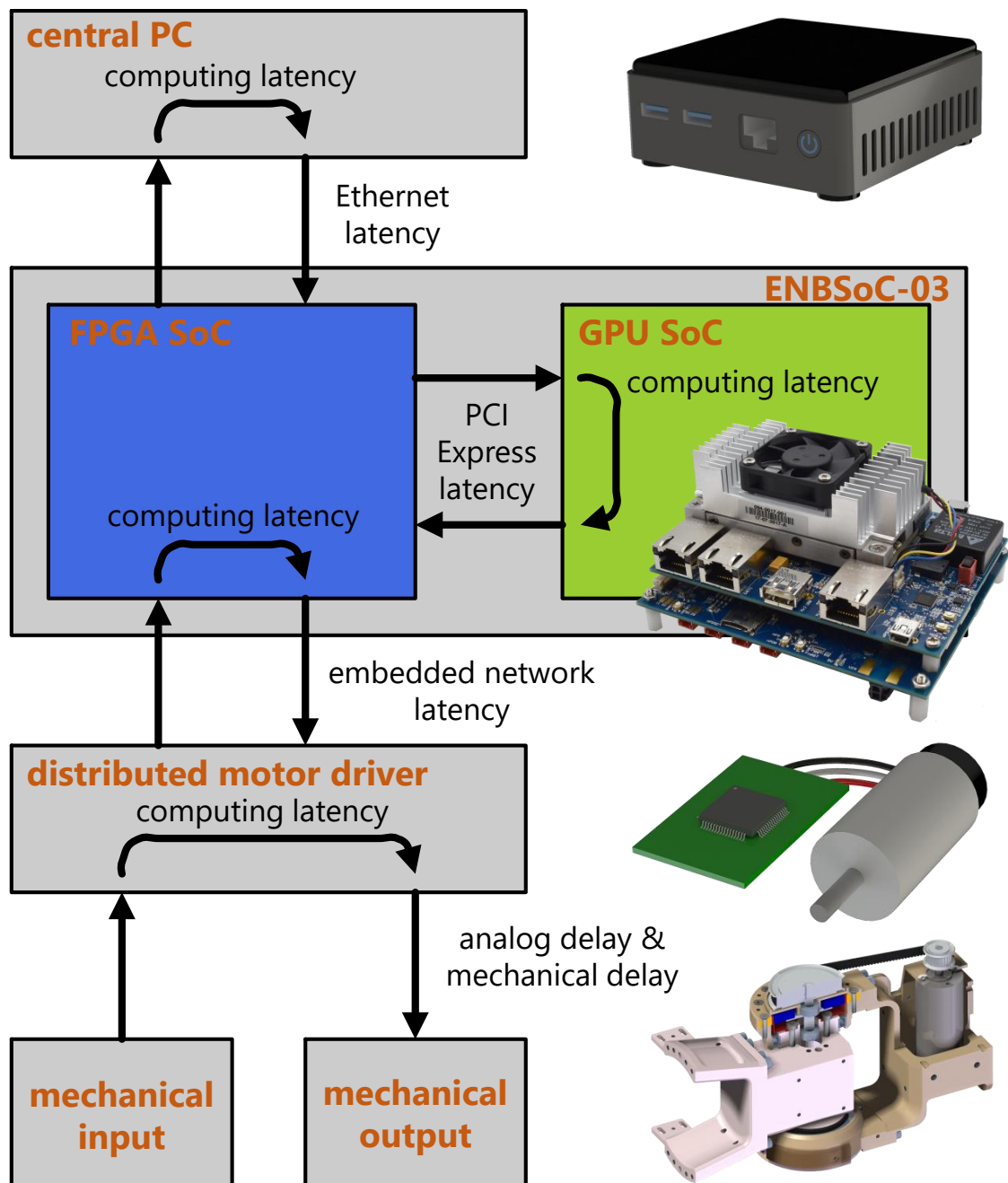


図 4.1 Feedback latency elements in the hierarchical control system

多階層化されたロボット体内制御システムでは，図 4.1 のようなフィードバック経路が利用できる．

外乱などの機械的入力を受けて，これを全身のアクチュエータ出力に反映することでロボットの動作を実施する．分散制御ノードごとに外乱に応答するか，あるいは中枢全身制

御系を経たフィードバック経路で全身動作制御を行う。中枢全身制御系は中央計算機と組込中間層計算機から構成される。FPGA を中心に据えて計算機・分散ネットワーク間でデータを交換する。センサ入力からアクチュエータ出力までのフィードバック経路でデータ転送や制御計算を経るごとにフィードバック遅延が累積する。

組込中間層計算機を導入して構成した体内分散制御システムでは、主に図 4.2 のような分類のフィードバック経路が考えられる。中央計算機・組込中間層計算機・分散制御ノードから構成される分散制御系の中で、一連のデータ転送および計算によりフィードバック制御が実装される。青線は組込分散制御系から得られる環境入力センサデータの経路、赤線は出力指令値データの経路を表す。データはそれぞれの階層でフィードバック制御計算やその他のデータ処理により変換・補正される。

図 4.2 (a) は、中央計算機に実装される動作制御系ソフトウェアでフィードバック制御が行われる一般的な実装である。多自由度ロボットの複雑な制御系を構成するマルチプロセスのソフトウェア構成において、動作制御系はハードウェア制御系から切り離され非同期に実行される。この非同期処理構造による中央計算機内部の大きなソフトウェア遅延も問題となる。

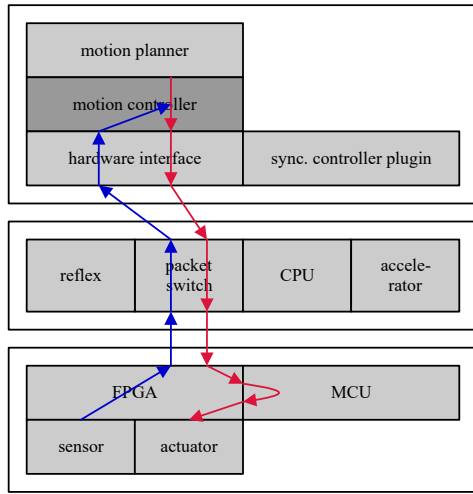
図 4.2 (b) は、中央計算機の動作制御機能をハードウェア制御系のソフトウェアと同期して実行する実装である。非同期処理構造によるソフトウェア遅延が排除できる。ハードウェア制御系で動作制御を行うための拡張プラグイン機構などにより実装できる。

図 4.2 (c) は、組込中間層計算機の CPU をフィードバック制御に用いる実装である。本研究で導入する組込中間層計算機モジュールでは FPGA SoC と GPU SoC の 2 通りの CPU が利用可能である。中央計算機を利用する図 4.2 (a) や図 4.2 (b) の実装と比べて、低遅延に組込分散ネットワークとデータを交換できる。

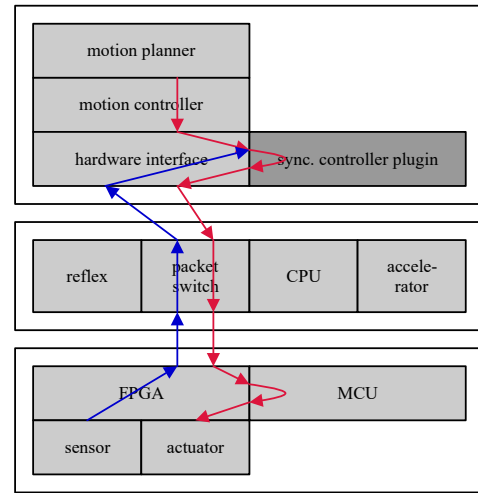
図 4.2 (d) は、組込中間層計算機で FPGA や GPU によるハードウェアアクセラレーションを利用した実装である。組込計算機の演算能力は中央計算機よりも劣るが、ハードウェアアクセラレーションを活用して演算能力を補うことでより高度な動作制御の実装が期待される。

図 4.2 (e) は、組込分散ネットワークの中枢ノードで CPU を利用せず FPGA で反射的に応答する実装である。FPGA ハードウェアアクセラレーションと同等の機能を単体でフィードバック回路として完結させるに等しい。実装可能な用途はロボット次第で限られるが中枢全身制御系の中では最も低遅延なフィードバック経路となる。単純で効果的なものとして、条件判定により全身のアクチュエータ制御モードを切り替えたり、干渉する複数のアクチュエータを連動させたり、といった用途が考えられる。特定の機能に特化させた回路構成を組み合わせてゆくことで、力フィードバック制御などの複雑な全身動作制御が実装可能となる場合もある。

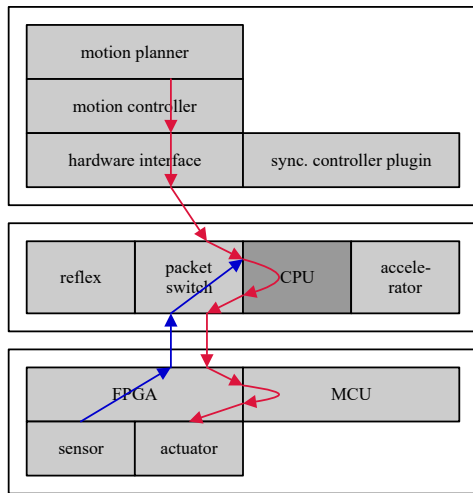
図 4.2 (f) は、ネットワークを経由せずに分散制御ノードごとに外乱に応答する実装で



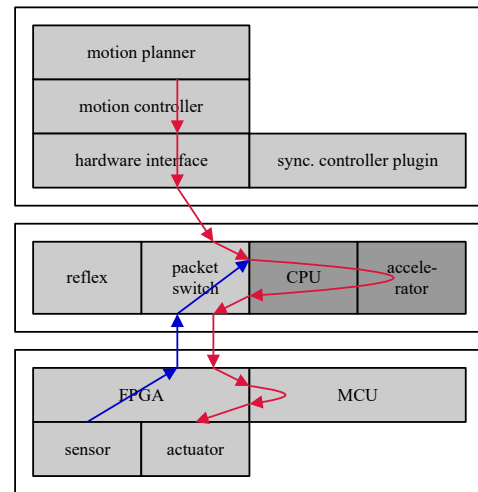
(a) High-level asynchronous motion controller



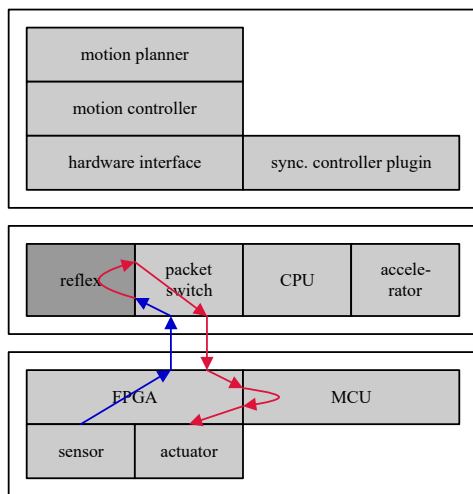
(b) Synchronous motion controller plugin



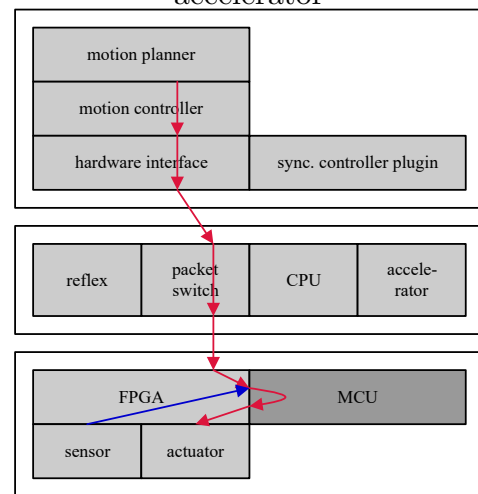
(c) Middle-layer computer



(d) Middle-layer computer with accelerator



(e) Reflex circuit



(f) Distributed local feedback

図 4.2 Feedback control path for disturbance response

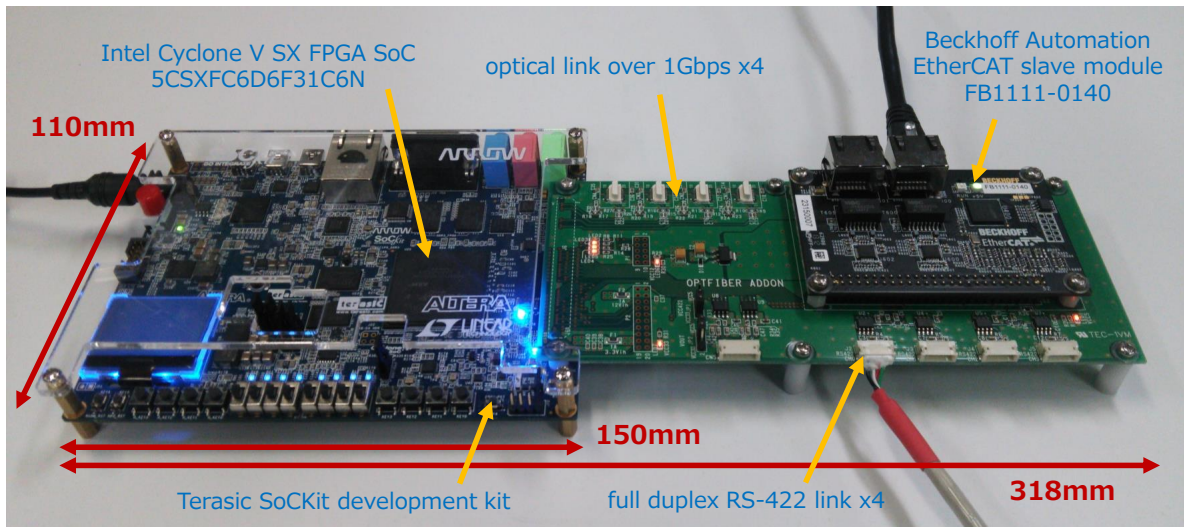


図 4.3 Overview of the embedded central middle-layer controller prototype

ある．分散制御ノードでローカルに取得できるデータに関しては，非常に低遅延なフィードバックが行われる．電流や位置をフィードバックした一般的なサーボ制御に加えて，関節ごとに外乱に馴染む力フィードバック型アクチュエータ制御の実装も考えられる．

4.3 組込中間層計算機的设计と実装

4.3.1 FPGA SoC を利用した組込中間層計算機プロトタイプ実装

組込中間層計算機の開発に先立ち，EtherCAT 実装を応用して FPGA SoC を搭載した組込中間層計算機のプロトタイプを開発した．図 4.3 に組込中間層計算機プロトタイプの概観を示す．Terasic 社の FPGA SoC 開発キットである SoCKit および Beckhoff Automation 社の EtherCAT スレーブモジュール FB1111-0140 を，独自のインタフェース基板 OPTFIBER ADDON で接続した構成となっている．

SoCKit には Intel 社の Cyclone V SX FPGA SoC 5CSXFC6D6F31C6N が搭載されている．FPGA SoC パッケージ内にはデュアルコア ARM Cortex-A9 CPU および高速トランシーバユニットが組込まれている．

組込中間層計算機プロトタイプのネットワーク接続インタフェース構成を図 4.4 に示す．インタフェース基板 OPTFIBER ADDON には，SoCKit および EtherCAT スレーブモジュールの接続インタフェースに加えて，従来型の RS-422 ネットワークおよび高速光通信ネットワークのインタフェースが搭載されている．高速光通信ネットワークのインタフェースは FPGA SoC の高速トランシーバユニットに接続されている．FPGA SoC の CPU は 100 Mbps Ethernet インタフェースを備えており，直接 LAN に接続するこ

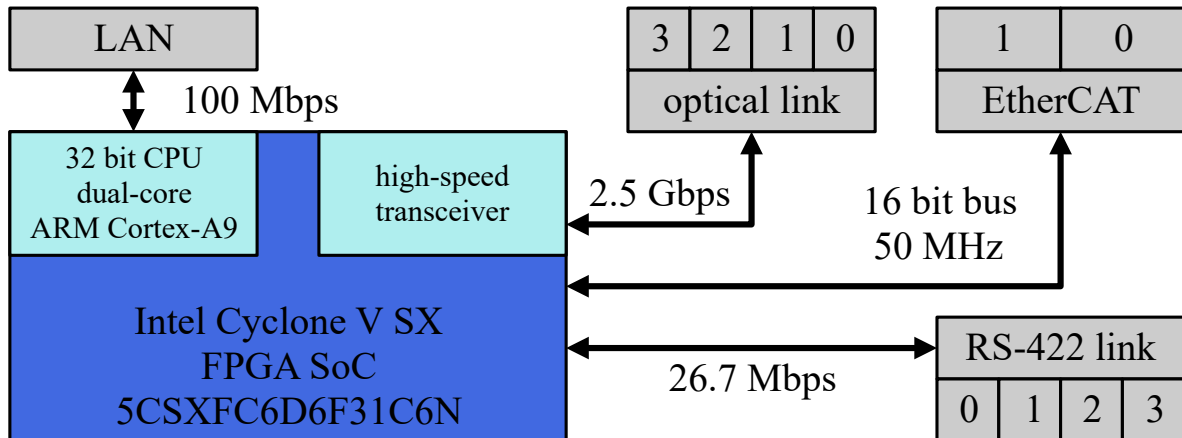


図 4.4 Network interface of the embedded central middle-layer controller prototype

とができる。

組込中間層計算機プロトタイプでは汎用的に利用される FPGA SoC 開発キットをベースにしており，豊富な周辺回路や端子を基礎的な開発・デバッグに活用することができる．本研究の研究開発工程の中では，FPGA SoC のシステム実装法検討，EtherCAT インタフェースの基礎実装開発，および高速トランシーバ・高速光通信の基礎実装開発において，組込中間層計算機プロトタイプを重宝した．一方で，そのまま多自由度ロボットの体内に搭載して運用することは想定しておらず，サイズの大きな卓上開発用装置となっている．多自由度ロボットに搭載するには，ロボット体内での運用に適したインタフェース仕様や電源仕様での小型パッケージ化設計が求められる．

4.3.2 組込中間層計算機モジュールの概要

プロトタイプ開発や通信インタフェース基礎実装の成果に新機能を加えて小型・多機能パッケージ化した組込中間層計算機モジュール ENBSoc-03 を開発した．図 4.5 に組込中間層計算機モジュールの概観を示す．複数基板によるスタック構造の小型パッケージの中に FPGA SoC および GPU SoC の 2 パッケージの組込計算機と各種インタフェースを搭載している．FPGA SoC や基本周辺回路に加えて組込分散ネットワークインタフェースを実装した基板，計算機ネットワークや GPU SoC モジュールのインタフェースを実装した基板，および既製品の GPU SoC モジュールから構成される．FPGA SoC として Intel 社の Cyclone V SX FPGA SoC 5CSXFC6D6F31C6N を採用している．GPU SoC を搭載したモジュールとして NVIDIA 社の Jetson TX2 を採用している．

図 4.6 に組込中間層計算機モジュールのハードウェア要素接続構成の概略を示す．FPGA SoC を中心に据えて，GPU SoC や各種インタフェースが接続されている．

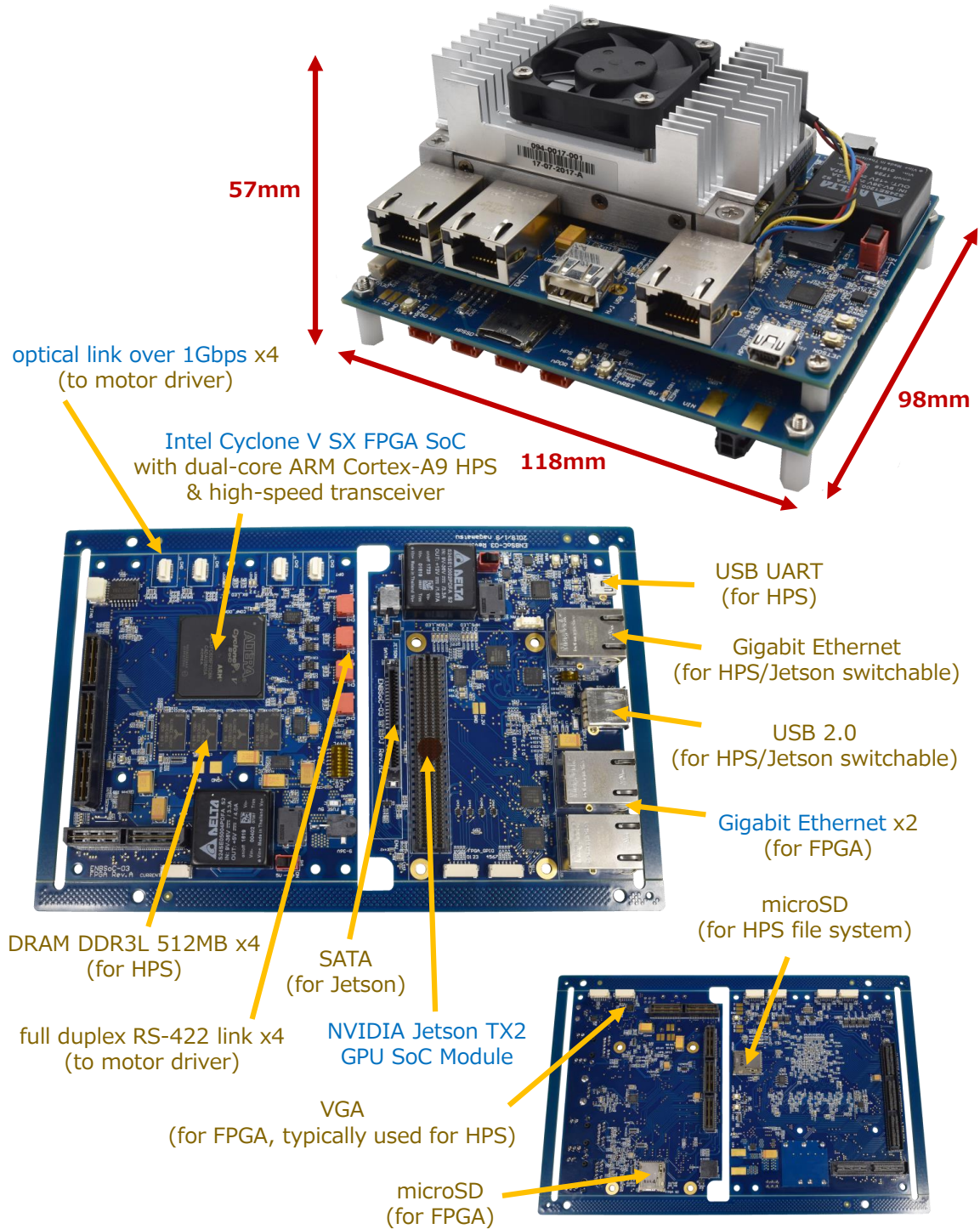


図 4.5 Overview of the embedded central middle-layer controller module ENBSoc-03 with FPGA-GPU dual SoC hybrid architecture

FPGA SoC と GPU SoC は PCI Express で接続される．高速光通信や従来型の RS-422 汎用シリアル通信による組込分散ネットワークにおいて中枢ノードの役割を果たす．上位の中央計算機とは Gigabit Ethernet で接続される．3.7.3 項の小型 Ethernet 中継基板 ENB-04 と同様に，FPGA の Ethernet インタフェースを 2 ポート設けている．その他，組込計算機の周辺接続インタフェースや，ロボット制御において利便性を有するオンボード周辺回路を搭載する．多段スタックによる FPGA SoC 基板の複数接続も可能としている．

4.3.3 組込中間層計算機モジュールの小型高信頼化設計

多機能な組込中間層計算機モジュールをロボットの体内で実用化するための小型パッケージ化は多くの設計要件および実装技術を伴い達成される．3.5.1 項でも述べたように，電子回路実装としての特性とロボット部品としての特性をともに深く理解し，高い信頼性や実用性を備えた設計を作り込む必要がある．本項では，組込中間層計算機モジュール ENBSoC-03 の設計実例を交えて，ロボット体内制御システムを構成するモジュールの設計要件や実装技術の要点を述べる．

キーデバイスと小型高信頼部品

電子回路設計においては初期段階でキーデバイス構成を決定する．キーデバイス構成により性能やサイズ規模など全体の仕様がほぼ決定されるため，非常に重要なプロセスである．制御計算を担う回路では中央コントローラである CPU や FPGA が代表的なキーデバイスとなる．通信回路ではトランシーバやコネクタ，モータ駆動回路では MOSFET などのスイッチング半導体やゲートドライバがこれにあたる．

ロボットの研究開発に用いるプログラマブル制御モジュールでは，構想段階で具体的な性能仕様が確定せず，運用の中での仕様変更に対応し多彩な制御システム実装への挑戦を支えてゆく基盤の提供が求められる．構想段階で想定する実装よりも高性能なデバイスの採用が功を奏する場合も多い．FPGA を用いると運用の中で周辺回路への入出力仕様が容易に変更できるため，ロボットの研究開発における有用性が高い．

ENBSoC-03 に搭載する FPGA SoC デバイスでも，極端に小型化のみを追求することではなく，主にコネクタ配置から決定される基板サイズに対してバランスよく規模の大きなデバイスを採用する．FPGA SoC の汎用 I/O 端子には，設計段階で利用予定のない周辺回路信号や GPIO も含めてなるべく多くの信号を接続し，運用の中での仕様変更に備えている．ENBSoC-03 では FPGA SoC に備えられている 288 端子の汎用 I/O のうち，90.6 % に及び 261 端子を引き出して信号接続している．

ENBSoC-03 に採用する GPU SoC としては設計段階で最新の高性能モジュールである

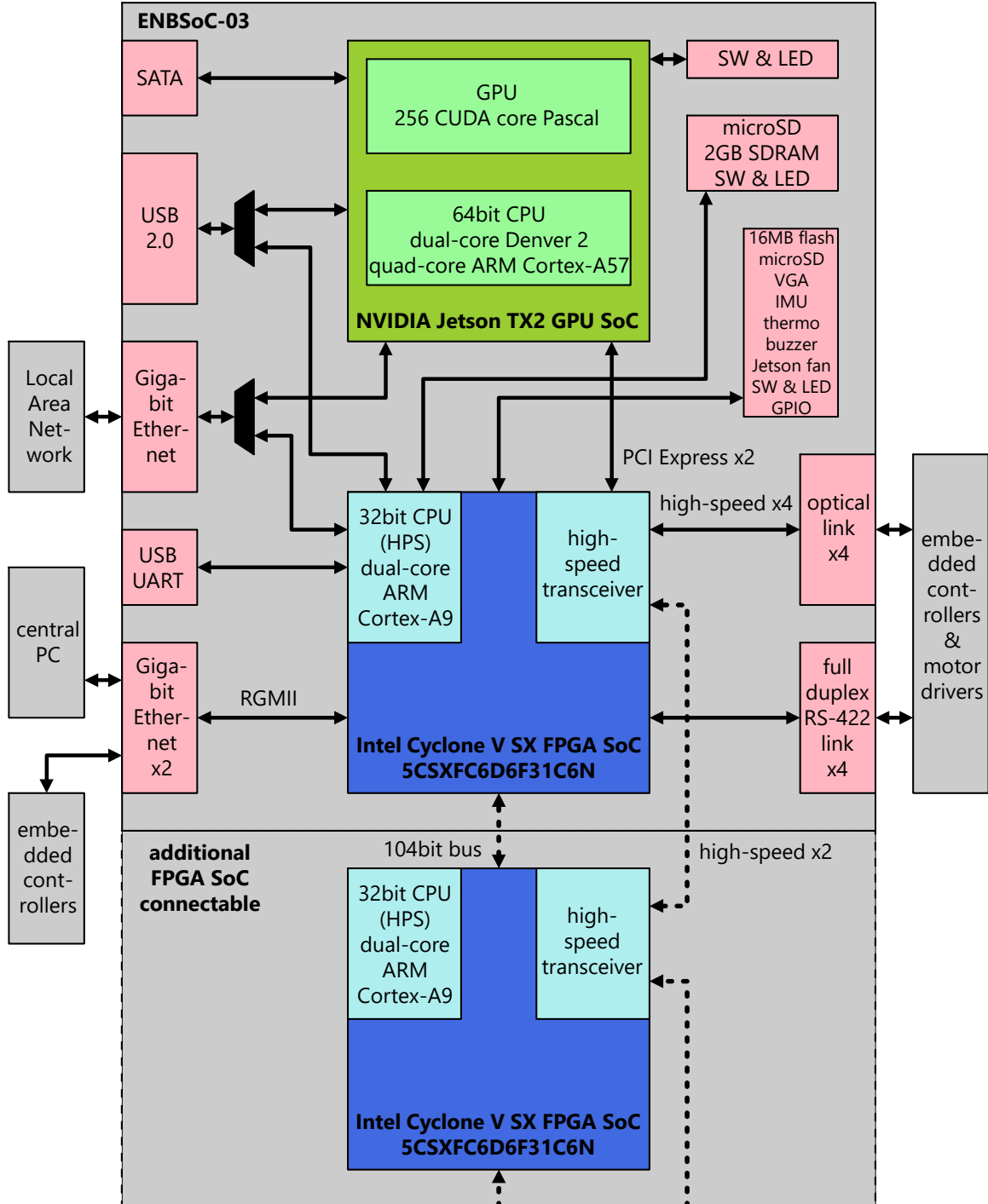


図 4.6 Hardware architecture of the embedded central middle-layer controller module ENBSoC-03

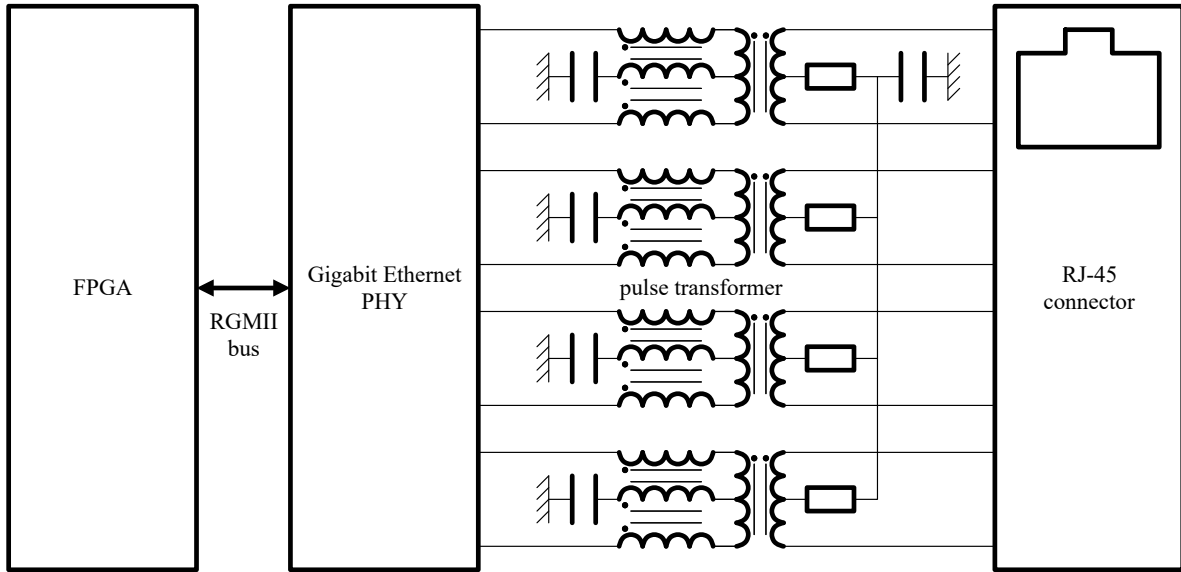


図 4.7 Physical layer of Gigabit Ethernet 1000BASE-T with pulse transformer

NVIDIA 社の Jetson TX2 を採用する . 87 mm × 50 mm の小型パッケージの中に 256 コアの GPU を含む計算機システムが実装されている .

それぞれの通信インタフェースでは, 3.6.3 項でも述べたようにシステム実装自由度の高い物理層回路を FPGA に接続することで, 柔軟なプロトコル実装に対応する . Gigabit Ethernet 1000BASE-T の物理層部品は図 4.7 のように主に, トランシーバ半導体デバイス, コネクタ, および絶縁化のためのパルストランスで構成される . コネクタだけでなくパルストランスも大型の部品であり, 小型化設計における厳しい制約を生じる . ENBSoc-03 では RS-45 コネクタとパルストランスが一体化された Bel Fuse 社の一体型コネクタ MagJack L829-1J1T-43 を用いて Ethernet インタフェースをコンパクトに実装する . FPGA SoC の CPU と GPU SoC からは汎用目的で USB および Ethernet のインタフェースを出力しているが, 図 4.6 のように 2 系統をマルチプレクサで統合することで大型コネクタの数量を削減する .

衝撃や振動を伴うロボット体内環境ではコネクタの信頼性が継続動作の可能性を左右する . ENBSoc-03 でも一般配線コネクタには小型ながらもロック付きで頑強に嵌合可能なものを利用する . 電源用には Molex 社の Micro-Fit 3.0, 信号用には日本圧着端子製造社の NSH もしくは GH を採用する .

周辺デバイス

ロボット体内の限られたスペースに分散配置する制御基板では, その主要な用途に直接関係のない場合でもロボットの身体状況や周辺環境を測定できるセンサを搭載しておくこ

とで、応用的な全身制御や環境認識に利用可能となり研究開発の幅が広がる。3.5.2 項・3.5.3 項のモータ制御モジュールや 3.8.1 項の中間制御基板では、9 軸 IMU を搭載し全身に分散配置する制御基板からの姿勢情報収集を可能としている。ENBSoC-03 にも旧 InvenSense 社の 9 軸 IMU MPU-9250 を搭載している。また Bosch Sensortec 社の温湿度・気圧センサ BME280 を搭載し、ロボットが作業を行う際の周辺環境を測定することができる。

センサ以外にもロボットの体内制御システムとして運用性を向上するための周辺デバイスを搭載する。基本的なものとして、それぞれの FPGA や CPU には LED を多数接続している。LED 表示はデバッグには有効であるが、注視しなければ目立たないため運用中の異常状態を知らせる用途には向かない。これを補うため FPGA にはブザーを接続する。音を用いた明瞭な状態通知を可能とし、運用性が向上する。小型で低消費電力な圧電スピーカを利用する。また FPGA には汎用目的の microSD カードを接続可能とする。小型で大容量な microSD カードを FPGA に接続することで、体内分散ネットワークを流れる大量のデータを常に蓄積してゆき、計算機の制御周期よりも細かい時間分解能の現象分析や異常時の状態保存などに応用することができる。

電源回路

ロボット体内の不安定な電源環境で動作する制御基板では、入力電源の電圧変動やノイズへ対応することが高信頼な継続動作の要件となる。そのためには、定格電圧を余裕のある広範囲に設定し、また絶縁やフィルタによって安定したオンボード電源を生成することが有効である。基板上に存在する、

- 不安定な入力電源
- 変動的な大電流の生じるアクチュエータ駆動電源
- 高速な通信や計算の電力を供給する高周波論理電源
- その他の一般電源

などの電源ドメインを適切に分離することで高品質な回路動作が達成される。

図 4.8 に ENBSoC-03 のオンボード電源ドメイン構成を示す。入力電源からレギュレータやフィルタを通してそれぞれの電源を生成する。

入力電源電圧は 9-36 V の広範囲に対応する。これにより、電源電圧変動への信頼性を向上するとともに、多様な電源構成のロボットに適用可能となる。ロボットを電池で駆動する場合、電源電圧には LiPo バッテリや LiFe バッテリなどの電池の化学的特性による制約が生じ、電池の充電残量によっても変動する。定格電圧の限られたデバイスを別途搭載する場合も多い。多様なロボットに展開する上では広範囲な入力電源電圧への対応が効果を発揮する。

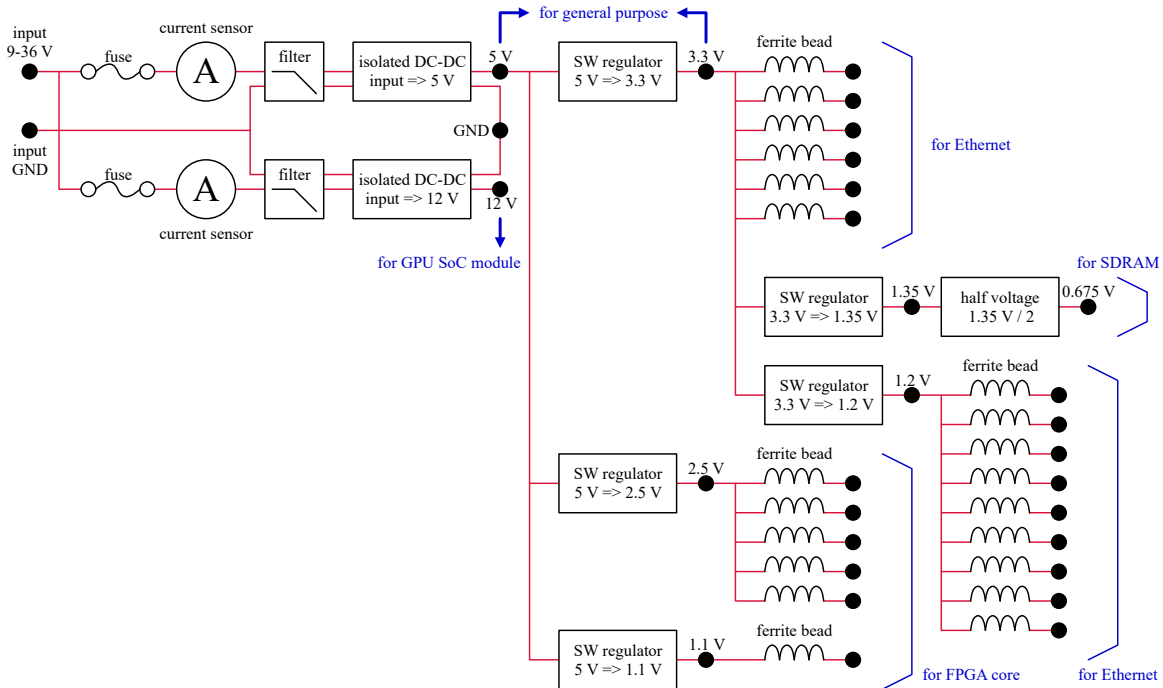


図 4.8 On-board power domain of the embedded central middle-layer controller module ENBSoC-03

入力電源からフィルタでノイズを除去して，絶縁 DC-DC コンバータにより安定なオンボード電源を生成する．フィルタには，LC フィルタとコモンモードチョークを組み合わせることでノーマルモードとコモンモードの両方を強力に除去可能な構成を採用している．いくつかのフィルタ製品を比較検討した結果，村田製作所社の BNX026H01 を利用する．ロボット体内では，アクチュエータを駆動する大電流やスイッチングの影響により大きなノーマルモードのノイズが発生し，また安定した接地電位が得られない中での空間電磁ノイズにより大きなコモンモードのノイズが発生する．これらの影響を抑えた安定したオンボード電源を生成することで信頼性が向上する．

高効率なスイッチングレギュレータを用いてそれぞれの必要な電圧を生成する．特に高速な通信や計算のために低ノイズ性が求められる電源ドメインはフェライトビーズなどによって十分なアイソレーションを施す．空間配置にも注意して，大電流によるノイズの影響を受けにくくするような配線設計が必要である．高速な通信や計算で変動的な負荷を生じる箇所には十分なバイパスコンデンサを配置することも信頼性向上のために重要となる．

高速信号

高速な通信や計算を実装するにあたり高周波な信号を取り扱う場面が生じる。高周波信号を高信頼に伝送する回路や配線の設計が重要となる。

2.5 Gbps の高速光通信や PCI Express の実装では、基板上で伝送する信号周波数が 1.25 GHz に及ぶ。従来より体内分散ネットワークに用いられてきた最大で数十 MHz 程度の汎用シリアル通信では厳しく求められなかった伝送路の品質管理が必要となる。信号伝送路では並走する配線や電源層との間に生じるリターン電流に基づいて構造に固有のインピーダンスが存在し、高速通信ではインピーダンスの設計が重要となる。伝送路のインピーダンスを受信側の終端抵抗と合致させることで効率的な信号電力伝達が行われ、また伝送路上のインピーダンスの不連続な変化を避けることで信号の反射が抑えられ、高品質な高速通信が達成される。高速光通信や PCI Express で利用する PCML 信号では送受信のバイアス電圧を整合させるためにコンデンサを直列配置して AC カップリングする。また受信側のデバイスの仕様によっては終端抵抗の挿入が求められる。伝送路上に挿入するこれらの部品配置も考慮して、等長・等幅で所望のインピーダンスを保った配線が必要となる。

高性能な計算機を実装するために、高速大容量な外部メモリの搭載が必要になる場合もある。現在利用されている SDRAM では多数の信号によるバス配線で数百 MHz から数 GHz の信号を伝送するため、高精度な信号タイミングの整合が求められる。そのために多数の信号を等長配線する必要がある。

ENBSoC-03 における高速信号の実装例を図 4.9 に示す。ENBSoC-03 の FPGA SoC が実装される面を図示している。

FPGA SoC の高速トランシーバユニットには、高速光通信のコネクタや GPU SoC モジュールの間の PCI Express インタフェースを接続する。これらの高速シリアル通信では 100 Ω の差動インピーダンスで設計した差動信号配線を行う。図 4.10 に高速シリアル通信の配線例を示す。

FPGA SoC の CPU には DDR3L 規格の SDRAM を接続する。4.3.1 項の組込中間層計算機プロトタイプでは FPGA SoC 開発キット上に 1 GB の DDR3 SDRAM が搭載されていた。組込中間層計算機プロトタイプでソフトウェア実装を試す中でロボットの動作制御計算に向けてメモリの不足が感じられる場面があったため、ENBSoC-03 では容量を 2 GB に増やしている。なお、設計時点では 1 GB の SDRAM デバイスである Alliance Memory 社の AS4C1G8MD3L-12BCN を 4 台用いて本来の容量を計 4 GB と想定していたが、FPGA SoC で利用できる最大メモリアドレス幅の仕様を見誤っており、実際には 2 GB のみ利用できる構成となっている。低消費電力化のため 1.5 V 信号の DDR3 よりも低電圧な 1.35 V 信号の DDR3L を採用している。FPGA SoC と SDRAM の間は

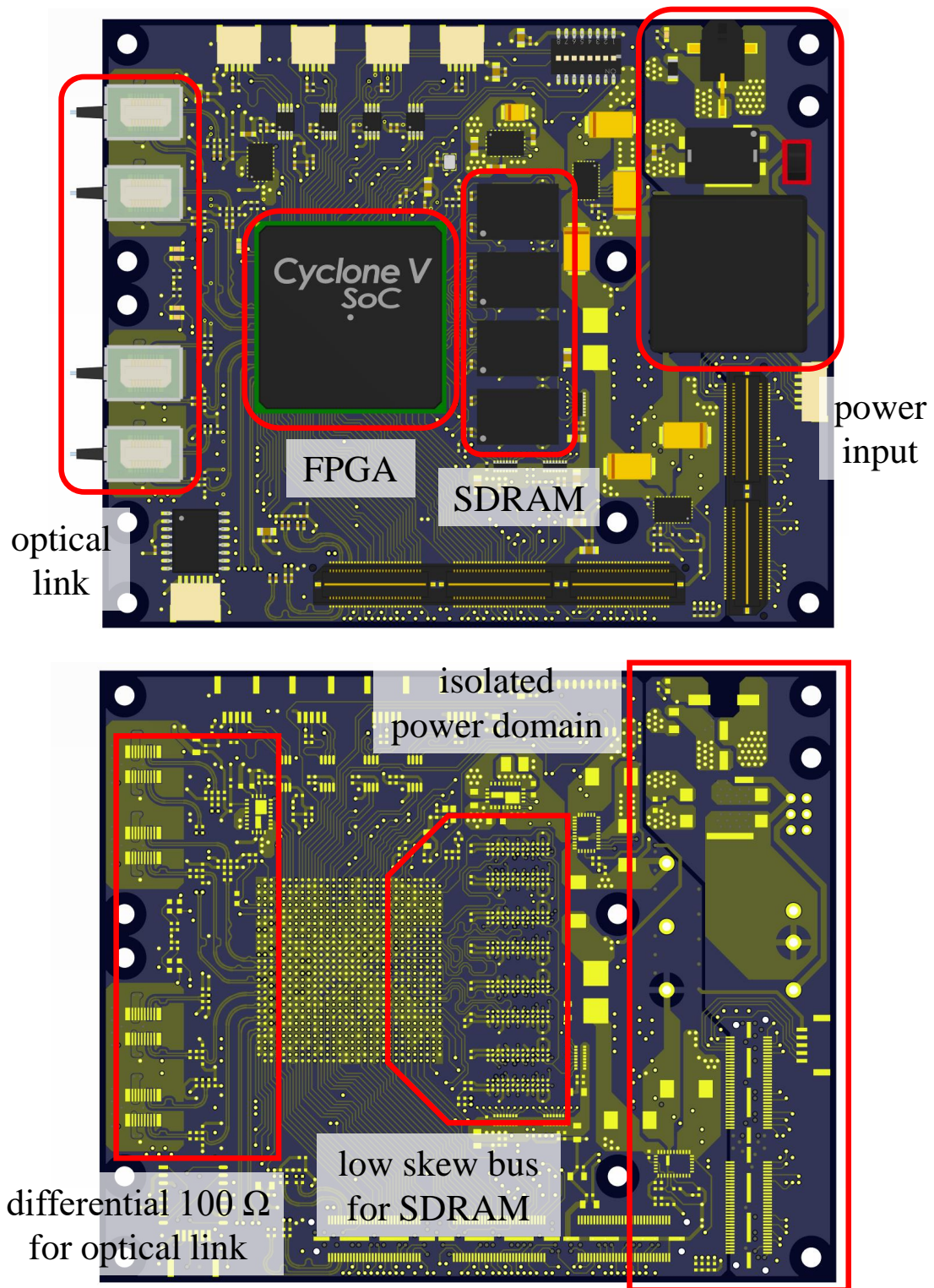


図 4.9 PCB design keypoint for the embedded central middle-layer controller module ENBSoC-03

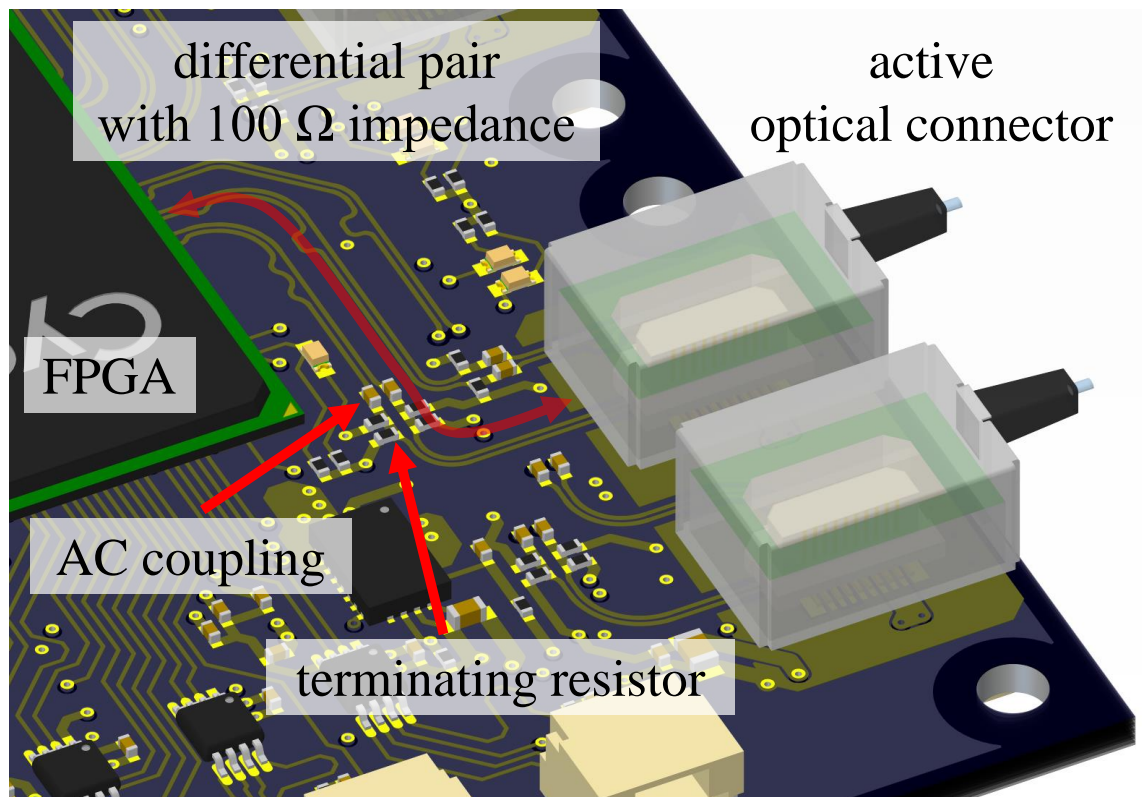


図 4.10 High-speed signal wiring for optical link

72 本幅のバス配線で接続される。FPGA SoC と SDRAM の間の短い距離の中で、徹底してスキューを低減した等長配線を行って信頼性を向上している。1 台目の SDRAM との配線長は全信号で最大配線長誤差を設計値で 0.1 mm 以下に抑えている。2 台目以降は一部の信号が 1 台目の配線から分岐して接続されるためおよそチップ間距離程度の差が生じる。

所望のインピーダンスを満たす高品質な高速信号配線を行うには、リターン電流経路として隣接する層に連続した安定な電源パターンを配置する必要がある。ENBSoC-03 では高速シリアル通信配線に十分広い GND パターンを隣接させる。高速な配線が特に多い FPGA SoC 実装面の隣接層は、高速信号のインピーダンス設計に加えて回路全体に安定した電源を供給する目的も兼ねて、GND パターンを一面に敷き詰める。不安定で大電流の生じる入力電源パターンと絶縁後のオンボード電源パターンは十分隔離して高品質な電源層を確保する。

多段スタック構造

ロボット制御モジュールの設計において、モジュールを複数の基板に分割して、基板対基板コネクタでスタックする構成が有効となる場合が多い。スタック構造には次のようなメリット・デメリットが挙げられる。

- スタック構造のメリット
 - 多段スタックで制御計算性能をスケラブルに拡張
 - 面積方向に小型化
 - 異なる仕様のロボットに同じモジュールを再利用
 - 故障時や仕様変更時の部品交換の手間や費用を削減
- スタック構造のデメリット
 - スタック構造によってモジュール高さが増大
 - スタック化のためにコネクタなどの部品点数が増大
 - 基板間配線の電流容量や高周波特性が劣化

3.5.3 項の小型光通信モータ制御モジュールでも、コントローラ基板・モータドライバ基板・I/O 基板をスタックしていた。

ENBSoC-03 では、FPGA SoC 基板・I/O 基板・GPU SoC モジュールのスタック構造を採用し、コンパクトなモジュールとして設計する。GPU SoC モジュールは既製品を利用する。FPGA SoC 基板と I/O 基板の間では高速信号と大電流電源の配線が必要となるため、これを可能とした小型コネクタである Samtec 社の Q Strip を利用する。主に Ethernet 通信インタフェースおよび GPU SoC モジュールとの接続を担う I/O 基板を FPGA SoC 基板から分離することで、高性能な一方で高価な FPGA SoC 基板を異なる構成を求めるロボットに再利用できる。FPGA SoC 基板には両面に対称な配置およびピン配列で基板対基板コネクタを搭載し、複数の FPGA SoC 基板をスタックすることができる。FPGA SoC 基板同士は図 4.6 のように 104 本の汎用デジタル信号と 2 ペアの全二重高速シリアル通信で接続される。複数の FPGA SoC を密に接続することで、1 台では達成できない高度な制御計算の実装にスケラブルに応用することが可能となる。

3次元構造設計

プリント基板設計は平面的な印刷製品を成果物とするため、2次元図面での設計作業に終始しがちである。高さ方向の情報は粗い部品高制限として取り扱われる場合も多い。しかし、ロボット体内で合理的に運用できる小型モジュールの設計開発においては、3次元構造を正確に把握することが重要である。

ロボットに搭載するための周辺の機械部品との干渉を避けて省スペースに収めるのに 3

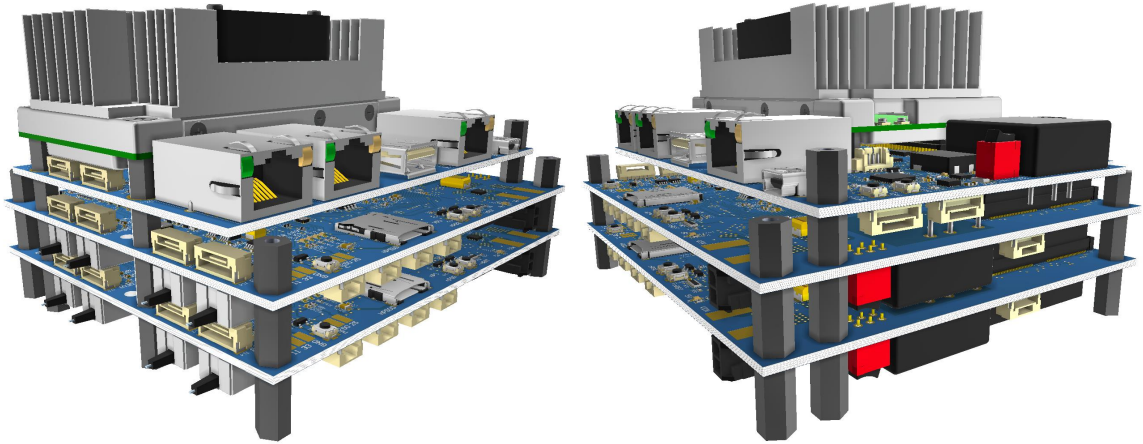


図 4.11 Design verification with detailed 3D model of the embedded central middle-layer controller module ENBSoC-03

次元形状情報を用いる。複数基板のスタック構造を採用する場合は、モジュール内部での基板同士の部品の干渉も避ける必要がある。

ロボット体内の限られたスペースで運用するにあたり、ケーブルアセンブリやスイッチ配置を正確に意識した設計も重要である。ケーブルアセンブリをコンパクト化し、またケーブルの曲率や張力を緩和するコネクタの配置計画が求められる。頻繁に操作するスイッチ類はユーザの操作性を考慮して配置する。

図 4.11 に ENBSoC-03 の 3 次元設計検証の様子を例示する。基板に実装する部品は可能な限り詳細な 3 次元モデルを用意して配置している。多段スタック構造の中で部品同士が接近している箇所も多く、干渉やショートのおそれがないか、3 次元形状をもとに一目瞭然で検証が可能となっている。コネクタ・スイッチ・microSD などユーザが操作する要素も、それぞれ合理的な配置を検証している。副次的な効果として、設計開発成果が高品質に視覚化されるため、設計のモチベーション向上に繋がり、また論文などに掲載する図の作成も容易となる。

消費電力

限られた電力で駆動するロボットの制御基板では消費電力の評価も重要である。電源の電圧および電流を計測可能とすることで、消費電力を明確に評価することができ、また運用中の異常状態を知覚することにも繋がる。

ENBSoC-03 では図 4.8 に示したように入力電源に電流センサを搭載する。絶縁電源の電流を測定するため、Allegro Microsystems 社のホール効果電流センサ ACS70331EESA-005U3 を利用する。また主要な電源電圧も AD コンバータで測定する。

表 4.1 Typical power consumption of each computer module

Module	Idle	Test Code	Max CPU
FPGA SoC & Peripheral	4.95 W	5.10 W	5.25 W
GPU SoC Module	3.90 W	4.80 W	5.55 W
Intel Core i7 PC (Typ.)	(5-10 W)	-	(20-40 W)

制御計算の状態に応じた細かい消費電力の評価が可能となっている。

ENBSoC-03 の FPGA SoC 系統および GPU SoC 系統の消費電力を一般的な Intel 社の Core i7 CPU と比較して表 4.1 に示す。4.4 節に後述する計算機性能評価実験構成の中で、

- OS を起動しただけのアイドル状態
- 評価実験テストコード実行中
- CPU に最大負荷をかけた状態

の条件で比較している。周辺回路を含む FPGA SoC 系統と GPU SoC 系統はともにおよそ 5W 前後の消費電力となっている。

排熱

ロボット体内での限られたスペースに配置する制御基板では、発熱する部品を効率的に冷却することが重要である。ロボット体内は熱が籠りがちであり、大電流部品が故障する他、スイッチングする半導体部品が熱暴走を起こす場合も多い。

ENBSoC-03 の場合は FPGA SoC・GPU SoC・入力電源レギュレータが主な発熱源である。モータ制御モジュールの場合は FPGA などのコントローラや電源回路に加えて、モータ駆動出力を高周波にスイッチングする MOSFET などが発熱源となる。

発熱が問題となる部品の周辺にヒートシンクや空冷ファンを適切に配置できるような 3次元構造設計が求められる。ENBSoC-03 では 3次元構造設計を通して FPGA SoC に薄型ファンが設置できる空間を確保している。GPU SoC モジュールは製品に備え付けられたヒートシンクとファンで冷却する。

部品上面に設置するヒートシンクや空冷ファンだけでなく、基板の導体パターンも排熱に大きく寄与する。内層の電源層を広く取ることで、電源として安定させるだけでなく効率的な熱の拡散にも有効となる。ENBSoC-03 の通常利用時にファンによる強制空冷を行わない場合でも誤作動を起こした例は確認されていない。

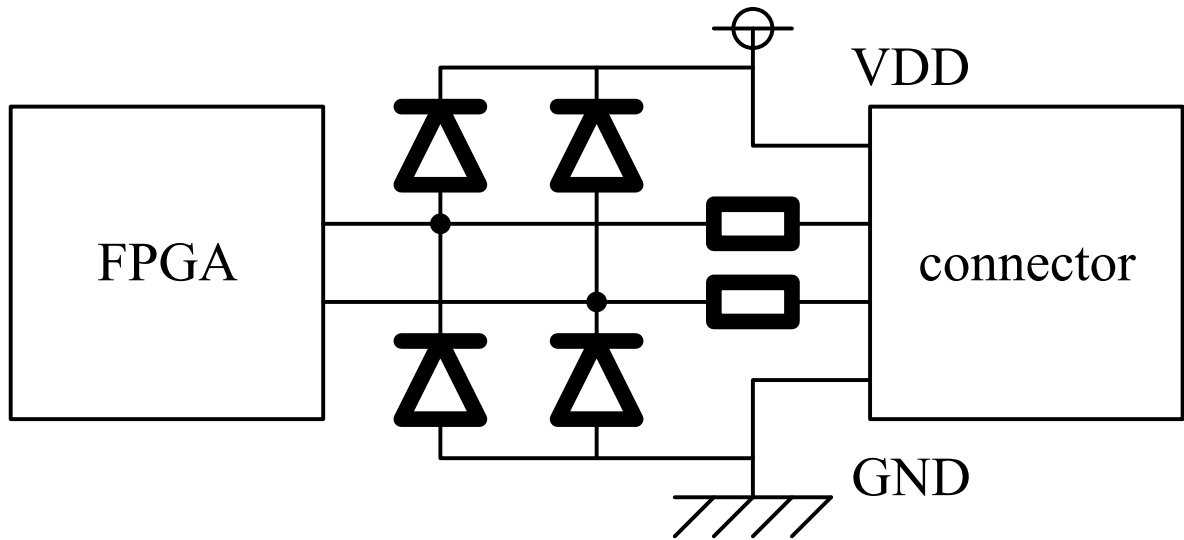


図 4.12 Antistatic circuit for FPGA I/O outlet

静電気対策

ロボットの研究開発の中では制御基板へのコネクタ挿抜が頻繁に生じる。ユーザがコネクタを挿抜する際に、静電気によるサージ電圧で半導体デバイスが故障する事例も珍しくない。ENBSoC-03 など特に高価なモジュールでは静電気への対策も重要である。

高い汎用性を備える FPGA を搭載した基板では、FPGA の汎用 I/O 端子をそのまま引き出してコネクタに接続する場合も多い。コネクタの挿抜で生じた静電気で FPGA が故障すれば、当該基板への致命的なダメージとなり修理も困難である。FPGA 端子から直接コネクタに接続する場合など、特に保護が必要な信号線には図 4.12 のような静電気対策を施す。電源に接続したダイオードによって過電圧の発生を抑える。信号の特性に不都合のない場合はダンピング抵抗の挿入も有効となる。

4.3.4 組込中間層計算機における組込制御システム実装

高性能な中央計算機と組込中間層計算機モジュール ENBSoC-03 を組み合わせてロボットの中枢全身制御系を構成する。このとき中枢全身制御系には、中央計算機・GPU SoC・FPGA SoC によるヘテロジニアス分散計算機アーキテクチャが構成される。

中枢全身制御系のヘテロジニアスアーキテクチャを図 4.13 に示す。図 4.13 (a) は計算機接続構成の概要を表す。図 4.13 (b) は ENBSoC-03 の詳細な計算機アーキテクチャを表す。水色で塗った枠内の項目が中枢全身制御系の計算機である。赤線矢印は実時間制御に通常用いるデータ転送経路、青線矢印はその他の非実時間・臨時のデータ転送に用いる

経路を表す。ENBSoC-03 では複数の FPGA SoC 基板を多段スタック構造で接続することを可能としており、図 4.13 (b) には FPGA SoC 基板を 1 台追加した場合の接続構成も含めている。

本項では、ENBSoC-03 の実装例に基づき、組込中間層計算機を用いて信頼性・汎用性・運用性を伴う低遅延制御計算を実装するための計算機アーキテクチャ活用法や組込制御システム開発技術の要点を述べる。

組込中間層計算機モジュールの詳細アーキテクチャ

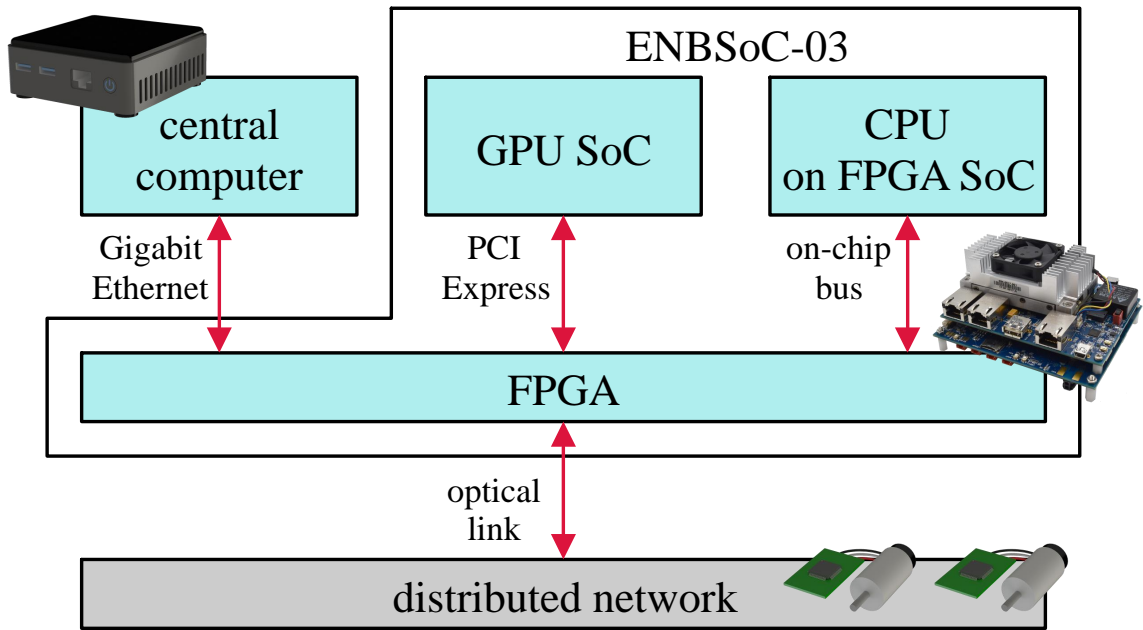
図 4.13 (b) に示した計算機アーキテクチャの詳細構成を記す。

ENBSoC-03 は FPGA SoC として、Intel 社の Cyclone V SX 5CSXFC6D6F31C6N を搭載している。FPGA SoC 上では主に、FPGA プログラマブル論理回路、高速トランシーバユニットに加えて、Hard Processor System (HPS) と呼ばれる CPU 関連システム回路が統合されている。HPS では 32 bit アーキテクチャの ARM Cortex-A9 CPU 2 コアが最大 925 MHz で駆動する。HPS と FPGA は複数のオンチップバスで接続される。HPS と FPGA の汎用的なデータ交換に用いる最大 128 bit 幅の AXI バスや、HPS から FPGA の管理に用いるバスが存在する。HPS がバスのマスタとなり FPGA にアクセスする経路の他に、FPGA がマスタとなる経路も用意されており、FPGA による高速な DMA データ転送も可能となっている。HPS には 2 GB のメインメモリと microSD カードのストレージを接続する。FPGA プログラマブル論理回路や周辺回路接続は、16 MB のフラッシュメモリ、HPS からの管理アクセス、および JTAG デバッグのいずれかによってコンフィグ可能となっている。

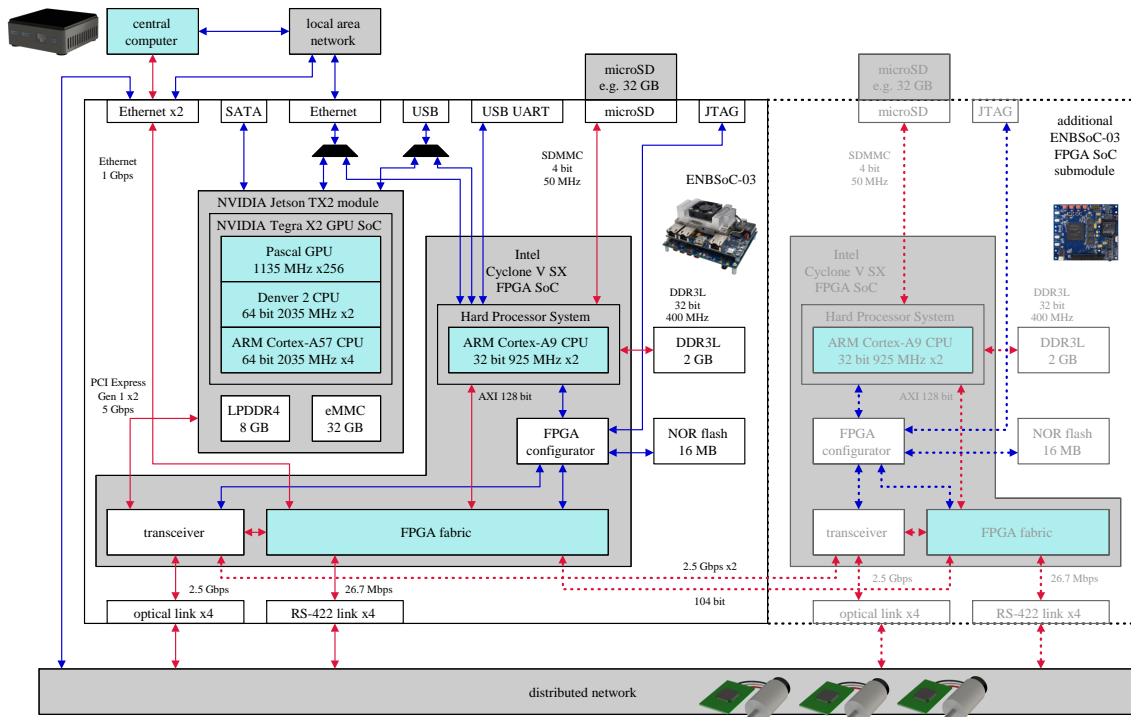
ENBSoC-03 は GPU SoC として、NVIDIA 社の Tegra X2 が周辺回路とともにモジュール化された Jetson TX2 を搭載している。GPU SoC では 64 bit アーキテクチャの ARM Cortex-A57 CPU 4 コアおよび Denver 2 CPU 2 コアが最大 2035 MHz で駆動する。256 コアの Pascal GPU がオンチップバスで CPU システムに接続され、最大 1135 MHz で駆動する。GPU SoC モジュールでは周辺回路として 8 GB のメインメモリや 32 GB のストレージが GPU SoC に接続される。

FPGA SoC と GPU SoC の間は 2 レーンの PCI Express で接続する。FPGA 側では、PCI Express も高速光通信と同様に高速トランシーバユニットを利用して実装する。

HPS と GPU SoC は FPGA を経由して体内分散ネットワークにアクセスする他、いくつかの周辺 I/O インタフェースで直接外部と接続可能となっている。HPS と GPU SoC はコネクタ削減のためにマルチプレクサを通して、Ethernet および USB の周辺接続インタフェースを共有している。HPS は UART でコンソールを出力しており、HPS で動作する OS を Ethernet もしくは UART を経由して操作できるようになっている。GPU SoC も同様の UART コンソールやシステム管理用の USB インタフェースを出力してい



(a) Overview



(b) Detailed system configuration

図 4.13 Heterogeneous computer architecture with the embedded central middle-layer controller module ENBSoC-03

るが、ENBSoC-03 モジュール上ではこれらのインタフェース接続を省略しており、デバッグや開発には別途キャリアボードを用いる。GPU SoC には SATA インタフェースを接続しており、内蔵ストレージが不足した場合に外部接続で補うことができる。

FPGA SoC ファームウェア展開

常に新たな機能実現を目指してロボットの研究開発を進めてゆく中で、制御基板に搭載した FPGA や MCU の実装を修正しそれを実機に反映する工程の簡略化・利便化がシステムの運用性向上に直結する。制御基板のファームウェアを更新するたびにロボットを分解し制御基板を取り出す必要が生じると、工程の増大と開発停滞に繋がるためこれはなるべく避けたい。

現在実用製品化されている FPGA としては、SRAM ベース FPGA もしくはフラッシュベース FPGA が主流となっている。SRAM ベース FPGA は揮発性の論理セルで回路を構成する方式を取り、別途不揮発メモリに格納したコンフィグデータがデバイス初期化時に展開される。一方フラッシュベース FPGA は不揮発性の論理セルで構成される。

SRAM ベース FPGA ではコンフィグデータが実際に動作するプログラマブル論理回路と切り離して保持されるため、FPGA の動作中にコンフィグデータのみを書き換えることができる。ネットワーク経由で FPGA に自らのコンフィグデータを更新させることも可能であり、運用性向上に有効な方式となる。本研究でもそれぞれの制御基板に SRAM ベース FPGA を採用している。一方でフラッシュベース FPGA の方が低消費電力化に有利であるため、頻繁に実装を修正しない制御基板においてはフラッシュベース FPGA が効果を発揮する場合も多い。

ENBSoC-03 の FPGA コンフィグシステムを図 4.14 に示す。FPGA のコンフィグデータは FPGA SoC に接続されたフラッシュメモリもしくは microSD カードに格納される。デバッグ時には、FPGA SoC の JTAG インタフェース経由で臨時的コンフィグを行うこともできる。フラッシュメモリに格納されたコンフィグデータは FPGA の初期化機能で読み出して展開する。フラッシュメモリの内容は JTAG インタフェースもしくは FPGA に展開したアプリケーション回路から更新できる。microSD カードは CPU からアクセス可能で、CPU で実行するブートローダにより microSD カードに格納されたコンフィグデータを読み出し展開する。microSD カードは容易に基板から脱着して、外部の計算機環境で内容を交換したり複製したりすることができる。ふたつの経路のうちいずれを用いるか、ENBSoC-03 に用意されたユーザスイッチで切り替え可能となっている。

CPU からは microSD データを通常のファイルシステムとして利用できるため、ソフトウェアから FPGA のコンフィグデータを更新することができる。FPGA の実装を修正した際は、コンフィグデータをネットワーク経由で送信し、microSD の当該ファイルを書き換えて再起動するだけで実機に反映することができる。デバッグ用の JTAG コネクタを

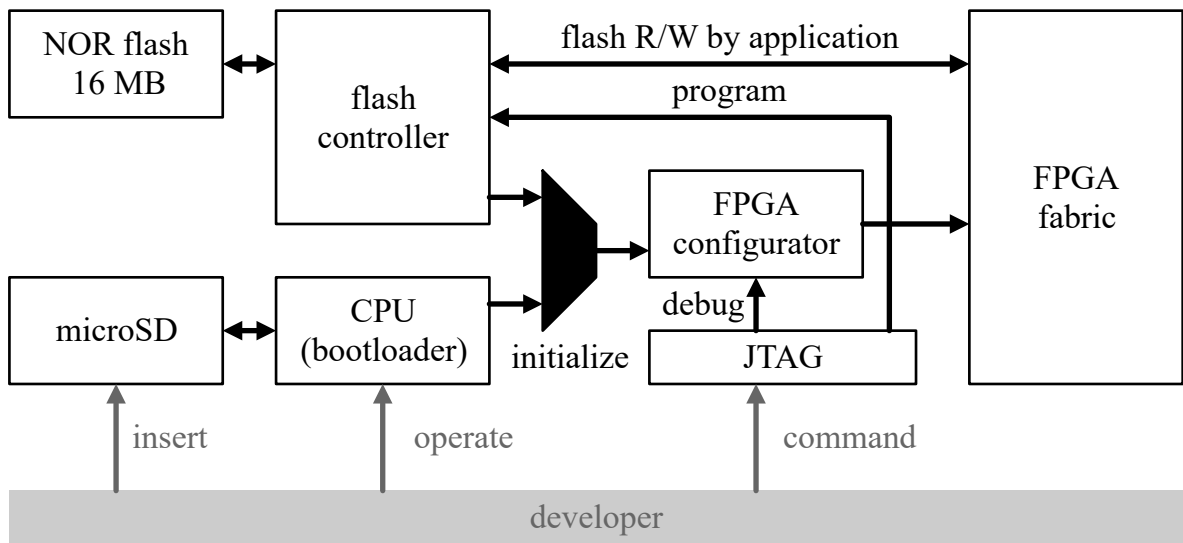


図 4.14 FPGA configuration architecture of the embedded central middle-layer controller module ENBSoC-03

都度挿抜する必要がなくなる。FPGA のコンフィグデータを、同じく microSD 内に展開される CPU ソフトウェアとともにパッケージ化することで、開発したソフトウェア環境を複製バックアップし別のロボット個体に適用することもできる。ロボットの研究開発において非常に利便性の高い運用が可能となっている。CPU や microSD カードを利用しない場合はスイッチを切り替えるのみで、フラッシュメモリによる運用へ変更できる。

組込計算機 OS

計算機上に高度なロボット制御システムを構成する上で OS の選択は重要な要素である。システム全体の制御計算遅延を低減するためには、演算スループットを十分に引き出す性能とジッタを低く抑える性能を両者欠かすことなく兼ね備える必要がある。

複数のタスク間で優先度を明確に定義し厳密な割り込み管理を行うことで、割り込みで起床する高優先度タスクが低遅延に実行されることを保証することができる。これにより低ジッタな実時間周期制御が達成される。実時間 OS (リアルタイム OS) と呼ばれる OS ではこのような仕組みを提供しており、QNX や VxWorks などの実時間 OS がよく用いられる。ロボットの制御においては Linux カーネルを元にした実時間 OS である ART-Linux [155] がよく用いられていたが現在では新規開発が行われていない。

一方で、厳密な割り込み管理を行うと計算機の利用効率が低下しシステム全体の演算スループットが抑制される場合も多い。高度なロボット制御計算を低遅延に実行するためにはバランスの良い割り込み処理が求められる。

本研究では組込中間層計算機の FPGA SoC および GPU SoC の OS として Linux を

用いる。代表的な構成において、中央計算機でも同様に Linux を用いる。いずれの計算機でも、Linux ディストリビューションとして、利用事例が多く開発が盛んに行われている Ubuntu を採用する。

Linux ではカーネルを交換して異なるオプションの構成を適用することで、高スループットなシステムから厳密な割り込み処理を実行する実時間システムまで段階的に選択して適用することが可能となっている。運用中のシステムにおいてもカーネルの交換は容易に行えるため、様々な構成を手早く試すことができる。Linux はオープンソースの OS であり、システムの複製や異なるロボットへの展開における利便性も備える。中央計算機と組込中間層計算機とともに Linux を用いることで、両者に同様のソフトウェアを適用可能となり、ソフトウェアの汎用性向上への足掛かりとなる。Linux では多様なハードウェア構成の組込計算機環境へ展開するために、デバイスツリーと呼ばれるハードウェア構成記述によるコンフィグが可能となっている。独自構成の制御基板である ENBSoC-03 でもハードウェア構成に基づいたデバイスツリーを適用することで Linux を導入する。

FPGA SoC データ転送

組込中間層計算機を導入する最大のモチベーションは、中枢全身制御系において組込分散ネットワークとの低遅延なデータ交換を達成することにある。FPGA SoC では CPU と FPGA がオンチップバスで接続され、CPU からは物理メモリアドレス空間へのアクセスによって FPGA とデータを交換できる。データ転送のオーバーヘッドが小さく、バス幅単位の細かいデータアクセスで高いスループットが発揮される。

ENBSoC-03 が搭載している Intel 社の Cyclone V SX FPGA SoC では主に、

- HPS-to-FPGA bridge : CPU から FPGA にアクセスできる最大 128 bit AXI バス
- Lightweight HPS-to-FPGA bridge : CPU から FPGA にアクセスできる補助的な最大 32 bit AXI バス
- FPGA-to-HPS bridge : FPGA から CPU 周辺のアドレス空間にアクセスできる最大 128 bit AXI バス

のデータバスが実装されている。CPU から見た 32 bit アドレス空間では、HPS-to-FPGA bridge は 0xC0000000、Lightweight HPS-to-FPGA bridge は 0xFF200000 といったベースアドレスで物理メモリアドレス空間にマッピングされている。

ソフトウェアからは物理メモリアドレス空間にマッピングされた FPGA バス領域に対してデータを読み書きする。Linux では高度なメモリ管理機能が実装されており、物理メモリアドレス空間はユーザアプリケーションに対して隠蔽されるため、アプリケーションから利用できる仮想メモリへのマッピングが必要となる。Linux のユーザ空間で動作する

ソフトウェアの場合は、物理メモリアドレス空間へのアクセスを提供する `/dev/mem` デバイスファイルおよび `mmap` システムコールによって実装する。Linux のカーネル空間で動作するカーネルモジュールの場合は、`ioremap` 関数によって実装する。カーネルモジュールではよりオーバヘッドの少ないハードウェアアクセスが可能となる一方で、ユーザ空間のソフトウェアと比べて実装や移植のための保守開発コストが増大するため、実装手法は用途に応じて選択する。ソフトウェアによる物理メモリアドレス空間へのアクセスで不足する場合は、FPGA がマスタとなる FPGA-to-HPS bridge を用いることで、FPGA から CPU メモリへの更に高速なデータ転送も可能である。

GPU SoC データ転送

組込中間層計算機に搭載された GPU SoC は、中央計算機と FPGA SoC に対して中間的な演算性能および通信遅延性能を提供し、エッジコンピューティング技術により組込制御系の機能の幅を広げるための計算機として利用される。ENBSoC-03 における GPU SoC の制御計算ネットワーク構成を図 4.15 に示す。GPU SoC では CPU と GPU の間でデータを共有しながら高速演算を実行し、PCI Express インタフェースを通して FPGA とデータを交換する。

CPU と GPU はオンチップバスで接続される。CPU が主体となり GPU にカーネルコードおよびデータを与えることで多並列計算のハードウェアアクセラレーションを利用する。GPU 並列計算は NVIDIA 社の提供する CUDA プラットフォームを利用して実装可能である。デスクトップ PC・サーバ・データセンタなどで利用されるような PCI Express 接続グラフィックボードの場合は、CPU と GPU が別デバイスであるため全てのデータをデバイス間で転送する必要がある。一方で、組込 GPU SoC の場合は CPU と GPU がオンチップでメモリコントローラを共有できる特徴を有する。そのため CPU の利用するメモリ領域に対する GPU からの直接アクセスが可能であり、GPU 並列計算のオーバヘッドを低減することもできる。

PCI Express は FPGA SoC の高速トランシーバユニットを利用して実装する。PCI Express Gen1 プロトコルの 2.5 Gbps 全二重高速シリアル通信を 2 レーン接続している。高速トランシーバユニットでは 8b/10b 符号化を用いたデータフレーム変換を行う。3.4.2 項の高速光通信で用いているものとおよそ同じ構成である。高速トランシーバユニットを拡張した PCI Express 実装により、Avalon-MM バスインタフェースで FPGA のプログラマブルロジック回路に接続する。PCI Express のホスト計算機である GPU SoC から FPGA の Base Address Register (BAR) と呼ばれるアドレス空間にアクセスする経路と、FPGA からホスト計算機のメモリにアクセスする DMA 経路が利用できる。DMA を利用することでデータ転送のスループットを向上することができる。PCI Express を利用するソフトウェアは Linux のカーネルモジュールとして実装する。

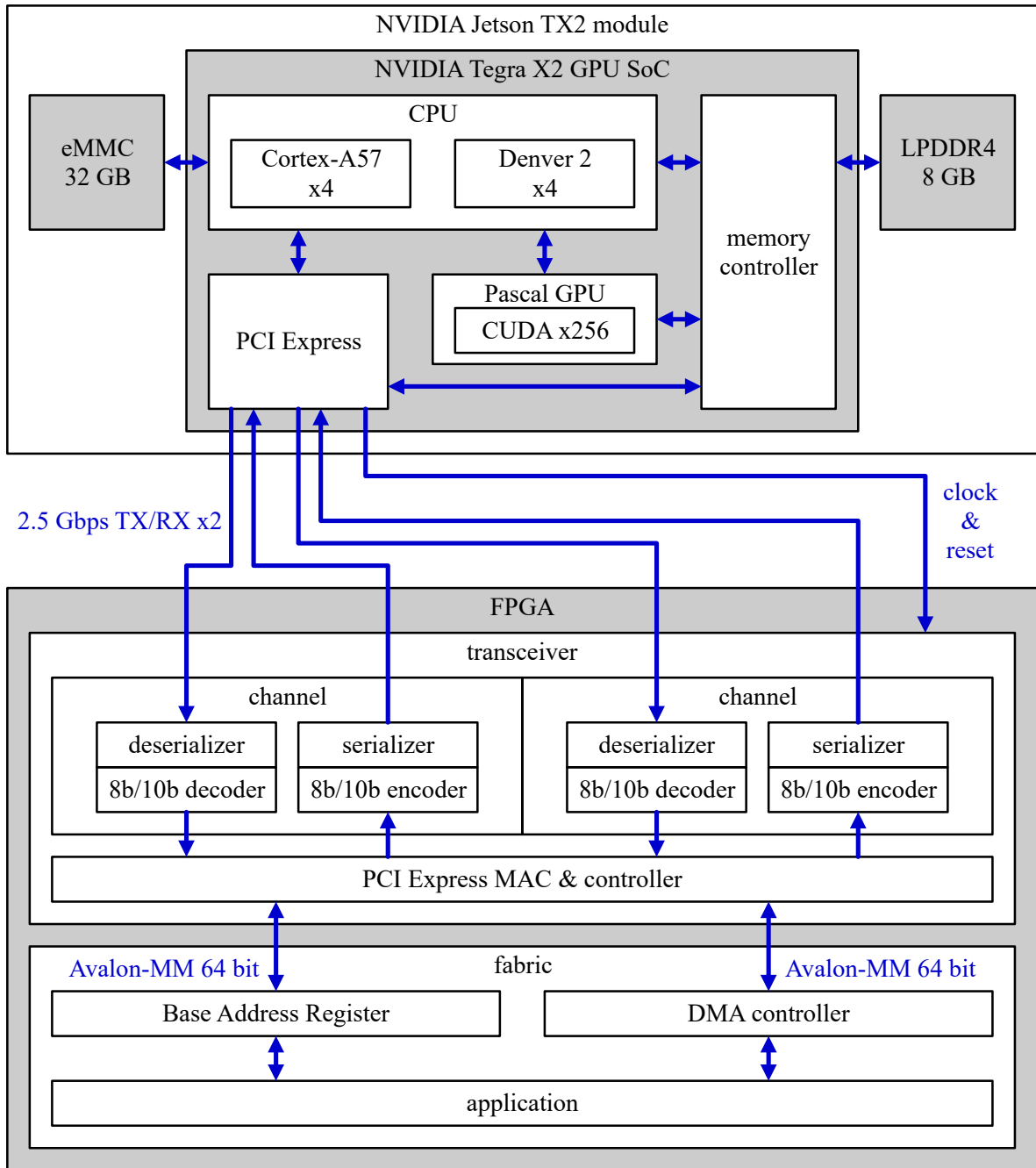


図 4.15 GPU SoC computational network architecture on the embedded central middle-layer controller module ENBSoc-03

4.4 組込中間層計算機における高速演算実装

本研究で開発した組込中間層計算機モジュールには Intel 社の Cyclone V SX FPGA SoC と NVIDIA 社の Tegra X2 GPU SoC のふたつの SoC が搭載されている。FPGA

SoCにはデュアルコア 32 bit ARM Cortex-A9 CPU が搭載され、925 MHz で動作する。GPU SoCにはクアドコア 64 bit ARM Cortex-A57 CPU およびデュアルコア 64 bit Denver 2 CPU が搭載され、2035 MHz で動作する。両者とも Linux をはじめとした高度な OS 環境が動作可能で、CPU 単体でも複雑な計算をある程度高速に実行できる。

加えて、FPGA および GPU のそれぞれ特徴的なハードウェアアクセラレーション技術を利用することができる。

FPGA ではプログラマブル論理回路を利用して任意のアクセラレータが構成できる。プログラマブル論理回路の容量が許す限りの並列計算ができる。同じ構成の回路を再利用した反復計算が適用できるアルゴリズムの場合は、クロックベースの厳密なシーケンスによって、非常に高速に計算を完了させることができる。例えば順運動学計算や Newton-Euler 法 [156] による逆動力学計算など、単純な漸化的アルゴリズムに効果的であると考えられる。FPGA SoC に搭載された CPU は FPGA のプログラマブル論理回路と密接に接続されており、通常メモリアクセスの要領でアクセラレータの性能を最大限活用することができる。一方で、その汎用性のため並列に展開できる回路規模はそれほど多くない。この欠点を補うため、組込中間層計算機モジュールは 1 台に複数の FPGA SoC 基板を接続可能としたスケーラブルな構成を採用している。隣接する FPGA SoC 基板同士は 2 系統の高速シリアル通信と 104 bit の汎用ロジックバスで接続される。

GPU SoC には 256 コアの Pascal GPU が搭載され、1135 MHz で動作する。GPU では同じ構造のカーネルコードを多数並列に実行することができる。大きなサイズの行列演算など多数並列化できるアルゴリズムの場合に、高速に計算を完了させることができる。例えば筋骨格ロボットの微分運動学による関節空間とアクチュエータ空間の写像などに効果的であると考えられる。GPU SoC はまた FPGA に対して PCI Express で接続されており、FPGA によるアクセラレーションを同時に利用する構成も実装可能である。

抽象化した行列演算の例を用いて組込中間層計算機モジュールの演算能力を評価する。ロボットの動作制御計算には、デカルト空間の自由度に相当する 6 次のベクトルやそれを並進と回転に分けた 3 次のベクトルが頻繁に登場する。ここではロボットの動作制御計算とおおよそ同規模の計算量を想定して、6 次正方行列の積算を評価に利用する。

Algorithm 4.1 に示した 32 bit 整数行列の積算をテストコードに用いる。これは整数行列 A の累乗 $B = A^{\text{ITERATION}+1}$ を求める計算に等しい。

Algorithm 4.1 と同等のテストコードを、FPGA SoC・GPU SoC・高性能計算機製品のそれぞれに実装して計算時間を比較する。FPGA SoC には Ubuntu 18.04.4 LTS / Linux 4.15.0 の OS を利用する。GPU SoC には Ubuntu 18.04.4 LTS / Linux 4.9.140 の OS を利用する。比較対象として、Intel 社の CPU Core i7-6600U を搭載した Lenovo 社のラップトップ PC T460s および Ubuntu 18.04.4 LTS / Linux 4.15.0-88-generic の OS を利用する。ラップトップ PC の CPU は 2600MHz で動作する。

Algorithm 4.1 Test Code with Matrix Multiplication

```

ORDER = 6, ITERATION = 1 to 1000
provide  $A \in \mathbb{N}^{\text{ORDER} \times \text{ORDER}}$  ( $\mathbb{N} = \text{uint32}$ )
 $B = A$ 
for  $i = 1$  to ITERATION do
     $C = AB$ 
     $B = C$ 
end for

```

評価実験結果を図 4.16 に示し，実験条件は考察とともに後述する．

テストコードを CPU で実行する場合は，イテレーションあたり 6^3 回の積算と 6^3 回の和算の繰り返しで実装される．FPGA にはそのプログラマブル論理回路の容量から，6 次正方形行列と 6 次ベクトルの積算を 1 度に行う単位積算回路が実装される．GPU では行列の要素数 $6^2 = 36$ だけ並列化して計算することができ，イテレーションあたりの計算量は 6 回の積算と 6 回の和算である．

条件 “FPGA SoC CPU” は，FPGA SoC の CPU のみを利用した結果である．他の比較対象である GPU SoC およびラップトップ PC と比べて低速であるため，計算に大きな時間がかかる．制御周期の目安とする 1 ms の時間では，50 回程度の積算が限界となっている．

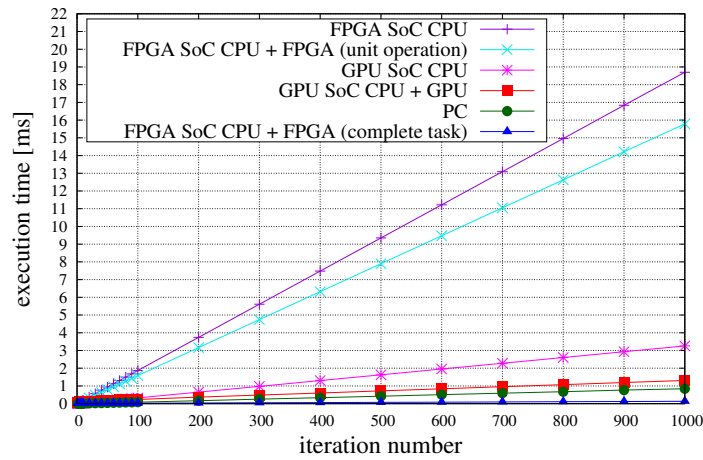
条件 “FPGA SoC CPU + FPGA (unit operation)” は，FPGA SoC の CPU で行列とベクトルの積算が必要となるたびに FPGA の単位積算回路にアクセスした結果である．CPU は FPGA に対して積算する行列とベクトルを書き込み，その後 FPGA から計算結果であるベクトルを読み出す．CPU と FPGA の間のデータ転送時間が，CPU での積算にかかる時間と比べてそれほど変わらず，全体の実行時間はごく僅かな短縮に留まっている．

条件 “GPU SoC CPU” は，GPU SoC の CPU のみを利用した結果である．速度に応じて中間的なベンチマークとなっている．

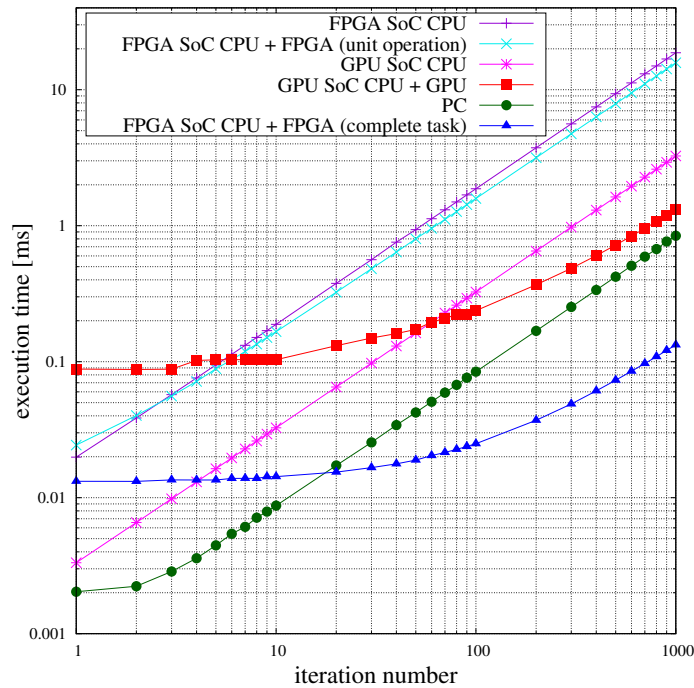
条件 “GPU SoC CPU + GPU” は，GPU SoC で GPU アクセラレーションを利用した結果である．CPU からは最初に GPU に対して積算する行列データを送信し，計算完了を待って GPU から計算結果の行列データを読み出す．積算回数が大きくなるとアクセラレーションの効果が現れる一方で，小さい積算回数では CPU と GPU の間のデータ転送時間が目立ちパフォーマンスを発揮できない．GPU はより並列数の大きなアルゴリズムに適しているものと考えられる．

条件 “PC” は，ラップトップ PC の CPU で実行した結果である．

条件 “FPGA SoC CPU + FPGA (complete task)” は，FPGA 内に Algorithm 4.1



(a) Linear scale view



(b) Log scale view

図 4.16 Execution time of iterative 6-order square matrix multiplication with FPGA SoC, GPU SoC, and PC (showing average of 10000 trials, “FPGA SoC CPU”: Cortex-A9 CPU on FPGA SoC, “FPGA SoC CPU + FPGA (unit operation)”: CPU on FPGA SoC with FPGA-implemented single-shot matrix-vector multiplication circuit, “GPU SoC CPU”: Denver 2 & Cortex-A57 CPU on GPU SoC, “GPU SoC CPU + GPU”: CUDA accelerated multiplication on GPU SoC, “PC”: Core i7-6600U CPU on T460s laptop PC, “FPGA SoC CPU + FPGA (complete task)”: CPU on FPGA SoC with FPGA-implemented iterative matrix multiplication circuit)

を実行するシーケンス回路を実装した結果である。GPU の場合と同様に、CPU からは最初に FPGA に対して積算する行列データを送信し、計算完了を待って FPGA から計算結果の行列データを読み出す。これは非常に優れた成績を収めている。FPGA のアクセラレータ回路は 50 MHz のクロックで駆動している。行列とベクトルの単位積算回路を利用して、1 度の行列の積算を 6 clock で実行する。テストコード全体はおよそ $(6 \times \text{ITERATION})$ clock、すなわち $(0.12 \times \text{ITERATION}) \mu\text{s}$ で実行できる。1000 回のイテレーションの場合、積算を行うシーケンス回路はおよそ $120 \mu\text{s}$ で実行を完了し、CPU とのデータ転送時間を含めた実行結果は平均 $131.1 \mu\text{s}$ であった。

体内分散ネットワークにおけるデータ転送遅延を考慮することで、それぞれの計算機による最短フィードバック遅延を試算する。図 4.16 の行列積算時間評価結果に、典型的なアプリケーションを仮定した最小データ転送遅延を加えて図 4.17 にプロットする。体内分散ネットワーク全体で 1 制御周期あたり 2000 Byte のデータ往復の発生を想定して試算している。全ての条件に、組込ネットワークの遅延として、2000 Mbps の光通信で 20 hop 分のストアアンドフォワード方式の転送を想定して $160 \mu\text{s}$ を加える。GPU SoC には PCI Express による通信時間 $100 \mu\text{s}$ を加える。ラップトップ PC には 1000BASE-T Ethernet の遅延 $300 \mu\text{s}$ 、および 100BASE-TX Ethernet の遅延 $1100 \mu\text{s}$ のそれぞれを加えて比較する。

データ転送遅延を考慮すると、どの計算量においても FPGA 内に専用のシーケンス回路を構成して利用した場合が最短のフィードバック遅延となっている。FPGA アクセラレータの構成はプログラマブル論理回路のリソース消費量とのトレードオフになるものの、実装できる場合は非常に高いパフォーマンスの低遅延フィードバック制御の実現可能性を提供する。FPGA 内の専用シーケンス回路による結果を除外すると、3 個の計算機それぞれが得意とする計算量の領域が浮き出る。制御計算量を 6 次正方行列の積算回数で換算して、およそ 7 回以下だと FPGA SoC、およそ 8 回から 200 回だと GPU SoC、それ以上だとラップトップ PC が、それぞれ最短のフィードバック経路となっている。これはあくまで抽象的なテストコードによる試算例だが、組込計算機を交えた相補的な構成による中枢全身制御系のコンセプトをよく表している。

4.5 中枢全身制御系におけるパケット切替中継機構

組込計算機によるフィードバック動作制御を可能とするために可変なデータ接続経路が必要となる。組込中間層計算機モジュールの FPGA 上にパケット切替中継機構を実装する。中央計算機と組込分散ネットワークの間でパケットを中継する通常の経路上に、隣接するコントローラから特定のデータを上書き可能なスイッチ回路を挿入することで、パケット切替中継機構を構成する。スイッチ回路に CPU や反射応答回路を接続し、場合

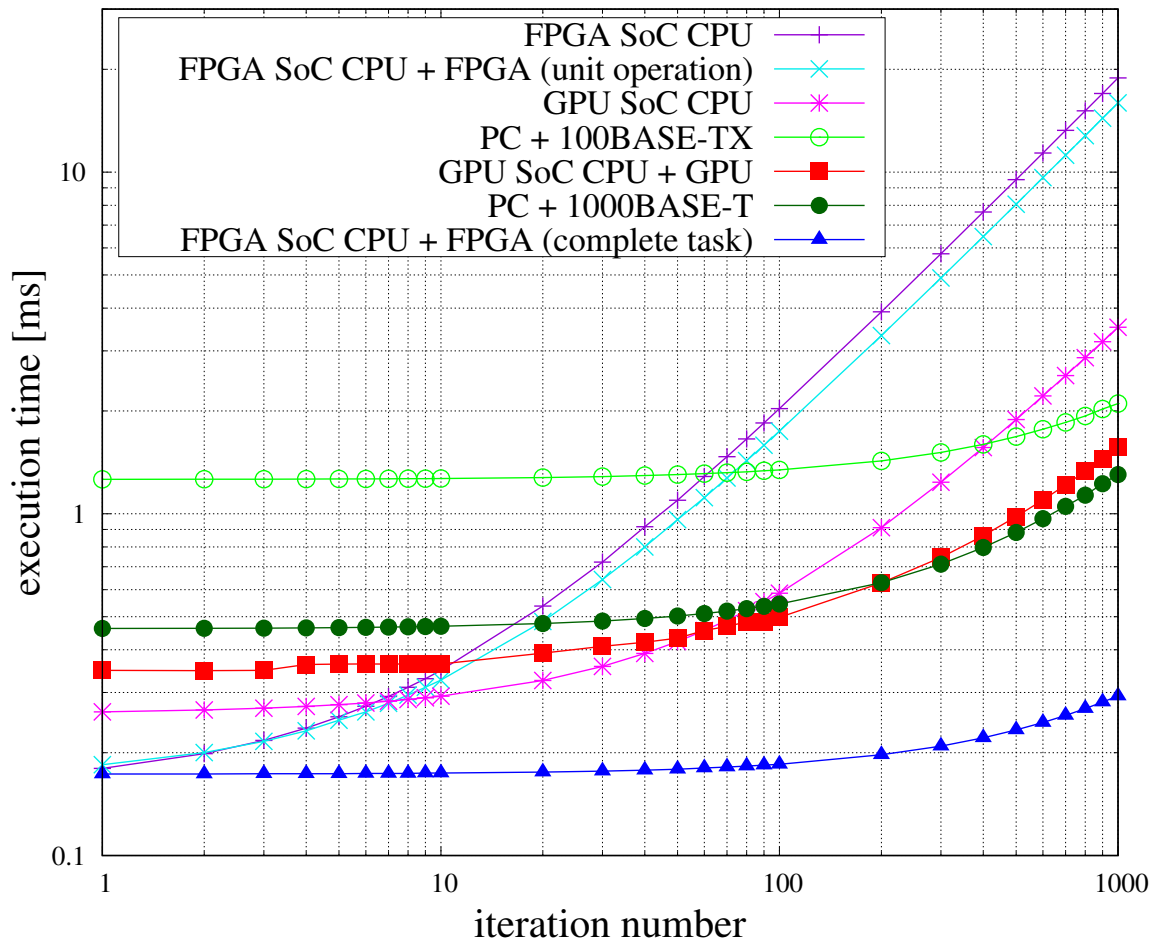


図 4.17 Estimated total electronic feedback system latency with constant data transfer offset in addition to execution time of iterative 6-order square matrix multiplication in 図 4.16 (FPGA SoC has 0.16 ms offset, GPU SoC has 0.26 ms offset, PC + 1000BASE-T has 0.46 ms offset, conventional PC + 100BASE-TX has 1.26 ms offset)

によってアクセラレータ回路とも組み合わせることで、様々なデータ接続が可能となる。図 4.18 に、パケット切替中継機構を含む組込中間層計算機モジュール上のデータ接続アーキテクチャを示す。組込中間層計算機モジュールの FPGA を中心に据えて、中央計算機・組込分散ネットワーク・GPU SoC・FPGA SoC 内蔵 CPU がそれぞれ FPGA に接続される。

中央計算機は Gigabit Ethernet インタフェースで接続される。Gigabit Ethernet 完全準拠の PHY を FPGA に接続した構成により、Ethernet パケット上には任意のプロトコルを実装することができる。ここでは EtherCAT を利用した旧実装との運用互換性を残しつつデータ転送パフォーマンスを向上させた、疑似 Gigabit EtherCAT プロトコルを

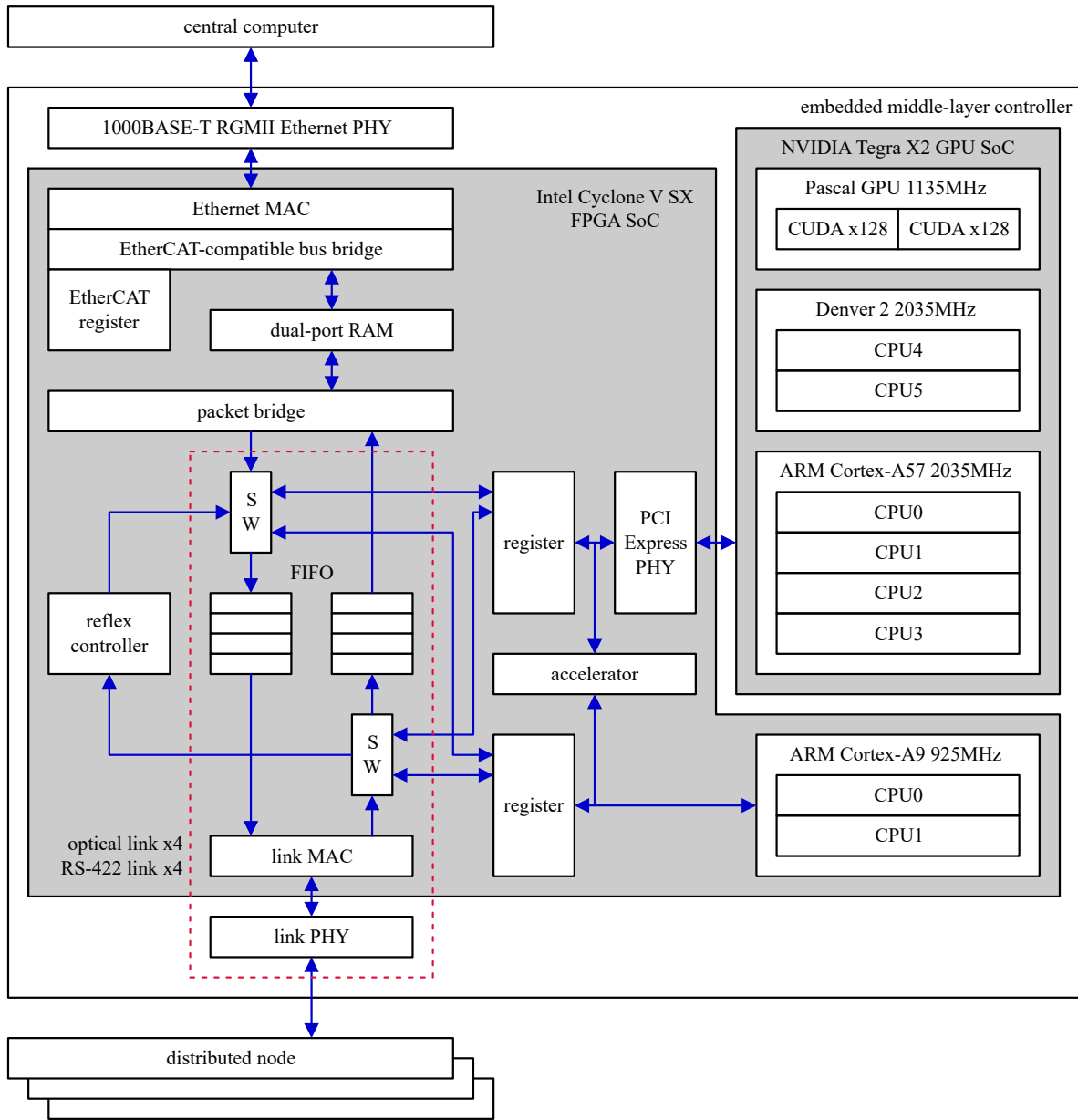


図 4.18 Hardware system with packet switch and accelerator in the embedded central middle-layer controller

実装している。中央計算機は EtherCAT プロトコルのパケット通信によって FPGA 内部のデュアルポート RAM を読み書きする。デュアルポート RAM 上に 32 bit 単位のデータパケットの配列を展開する。デュアルポート RAM のもう一方のポートにパケット中継コントローラからアクセスして、データパケットを体内分散ネットワークと交換する。

組込分散ネットワークのインタフェースとして、組込中間層計算機モジュールには 4 系統の高速光通信インタフェースと 4 系統の RS-422 汎用シリアル通信インタフェースが搭

載されている。それぞれの系統には分散制御ノードがデジチェーン接続されネットワークを構成する。これらの計8系統のインタフェースがFPGAに接続され、データパケットの送受信を可能としている。

GPU SoCはPCI ExpressインタフェースでFPGAに接続される。GPU SoC上で動作するOSからのPCI Expressメモリアドレスへのアクセスは、FPGA内部のバスに変換される。FPGA SoCに内蔵されたCPUは直接FPGA内部のバスに接続される。

組込計算機を利用しない通常構成の場合は、中央計算機と接続されたEthernet系と8系統の組込分散ネットワークの間でFIFOバッファを介してデータパケットが交換される。この経路にスイッチ回路を挿入して組込計算機を交えたデータ接続システムを構成する。スイッチ回路はFIFOバッファの入力前段に挿入される。組込計算機からスイッチ回路のルーティング設定を行い、スイッチ回路では設定に従って入力データパケットのデータ種別に基づく切替中継を行う。

4.6 本章のまとめ

本章では、組込中間層計算機を交えた相補的なヘテロジニアスアーキテクチャに基づくロボット中枢全身制御系を提案した。

中央計算機において、高次の行動計画および動作生成を担う計画系と実時間のハードウェア通信および動作制御を担う実行系が計算資源を共有し、Ethernetインタフェースで組込分散ネットワークと接続する構成の課題を確認した。ハードウェアアクセラレータを伴うFPGA SoCおよびGPU SoCを搭載した組込中間層計算機を導入することで、低遅延動作制御が実現可能となることを述べた。

汎用的なFPGA SoC開発キットを利用した組込中間層計算機プロトタイプによるシステム基礎実装開発を紹介した。プロトタイプ開発に続いて、FPGA SoCおよびGPU SoCモジュールを搭載し小型パッケージ化した組込中間層計算機モジュールの設計開発に関して述べた。

多機能な組込中間層計算機モジュールの実用化設計の実例を通して、ロボット体内制御システムを構成するモジュールの設計要件および実装技術を明確化した。また、組込計算機アーキテクチャを活用して実時間性および展開実用性を満たすための組込制御システム開発技術に関して述べた。ロボット制御モジュールの設計開発において、ロボット技術と実装技術のそれぞれの特性をともに深く理解し、高い信頼性や実用性を備えた設計として作り込むことの重要性を確認した。

組込中間層計算機において、FPGAおよびGPUによるハードウェアアクセラレーションの効果を行列演算に基づき評価した。FPGA SoC・GPU SoC・中央計算機による相補的なヘテロジニアスアーキテクチャに基づく中枢全身制御系構成の効果を示した。

中枢全身制御系において組込中間層計算機を活用した低遅延フィードバック経路を実現するためのパケット切替中継機構を開発し、柔軟なデータ転送にFPGAを利用する効果を確認した。

第 5 章

多形態多自由度動作制御における
組込高速演算を交えた全身階層協調系

5.1 はじめに

中央計算機におけるロボット全身動作制御では，高性能な CPU により高度な演算が可能一方で，分散制御ノードとの間の大きなデータ転送遅延が問題となる．通信ネットワーク遅延に加えて，マルチプロセス構成の制御ソフトウェアにおけるプロセス間通信遅延も大きな影響を与える．分散制御ノードにおけるローカルフィードバックや，通信遅延を小さく取れる組込中間層計算機での全身動作制御，あるいはソフトウェアデータ転送経路の縮約によってフィードバック遅延が低減される．組込中間層計算機で高度な動作制御計算を高速に実行する技術も重要な役割を果たす．

本章では，ロボットの力フィードバック制御における動作制御計算の構造を分析し，組込計算機を交えながら低遅延に実行する手法を提案する．環境接触力をフィードバックする最も低遅延な手法は，分散制御ノードにおけるローカル関節力フィードバック制御である．トルクセンサを搭載しない関節において，モータ制御モジュールでローカルに得られる情報のみを利用して近似的に関節トルクを推定してフィードバックする手法を検討する．一方，位置制御型のアクチュエータにおいては，力フィードバック制御を担うのは中枢全身制御系である．中枢全身制御系の計算機において低遅延なデータ交換と動作制御計算を担う基盤ソフトウェアの実装に関して述べる．また，組込中間層計算機を用いてハードウェアアクセラレーションを交えながら高速に動作制御計算を実行する手法を検討し，より低遅延な全身フィードバック動作制御を実現する．

5.2 多自由度ロボットの動作制御計算構造

多自由度ロボットの分散制御ノードには，主に位置制御型もしくは力制御型のアクチュエータ制御システムが実装される．

位置制御型では，高剛性な位置フィードバックにより中枢全身制御系から指令された関節位置を忠実に再現する．高い軌道追従性により精密な動作を容易に実現する．また中枢全身制御系から指令された通りの位置姿勢で安定するため，取り扱いも容易である．位置制御型を利用する場合，力制御や外乱への応答は中枢全身制御系が担う．

力制御型では，中枢全身制御系からの指令に基づき，目標となる力学特性をアクチュエータや関節に再現する．力指令を受け取ってこれをそのまま発揮する場合や，剛性などの力学特性を力フィードバックにより再現する場合がある．位置に関する剛性は低く，外乱に対する高追従な応答がアクチュエータごとに実現できる．一方で，制御対象が位置ではなく力になることで，動力学特性を考慮した出力計算が必要になるとともに，位置に関して不安定なシステムになる場合も多く，所望の動作を実行するためには中枢全身制御系による高度な制御が必要になるという難点もある．

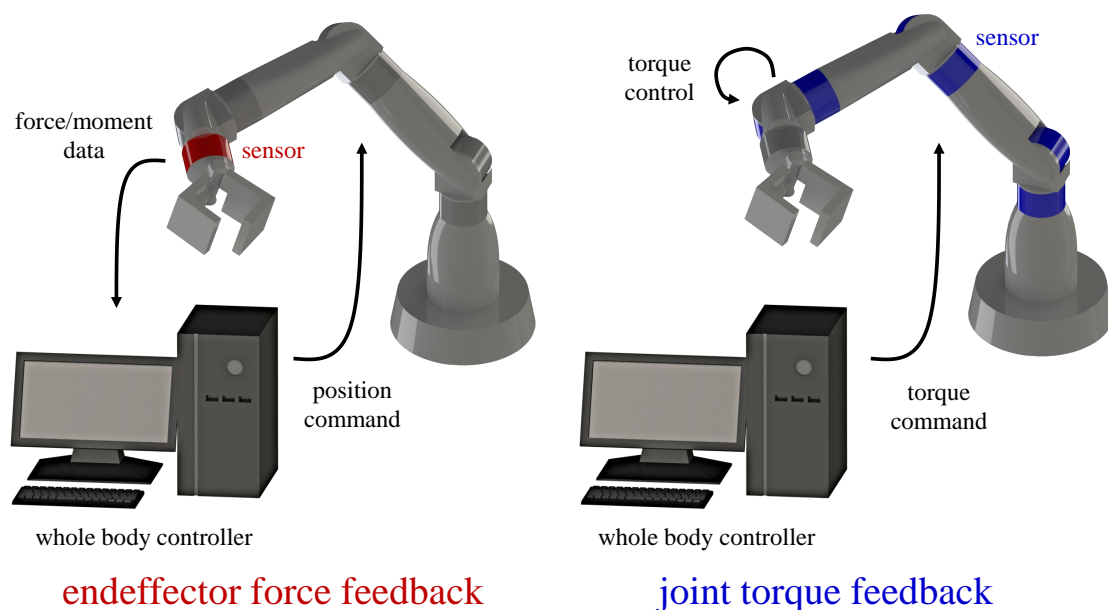


図 5.1 Force feedback control architecture for position controlled robot and torque controlled robot

多自由度ロボットの動作制御における外乱応答性を向上するには、

- 関節力制御型のアクチュエータ制御システムを採用することで関節ごとに外乱に
応答
- 位置制御型のアクチュエータ制御システムを採用する場合は中枢全身制御系の力
フィードバック遅延を低減

といった2つの方針が考えられる。図 5.1 に位置制御型と力制御型のそれぞれにおける力フィードバック制御構造の例を示す。

関節力制御は、分散制御ノードにおける応答性の高い外乱フィードバックによって実現される。中枢全身制御系のフィードバック遅延低減は、体内ネットワークの低遅延化に加えて、ロボット動作制御器を応答性の高い組込制御系に実装することで実現される。いずれの場合も、組込計算機における高速演算技術が重要な役割を果たす。

中枢全身制御系に実装されるロボット動作制御器では、動作計画器から与えられた目標動作指令に対して現実環境に適応した補正を施すことで、最終的な分散制御ノードに対する出力制御指令を発行する。図 2.7 に示した知能ロボットシステムの real-time online motion controller に相当する機能ブロックである。ロボット動作制御器における動作補正は主に、

- ロボットのモデルのみに基づくフィードフォワード補正
- ロボット実機のセンサ情報も反映したフィードバック補正
- 異常な出力を避けるための特別なフェイルセーフ補正

に分類して考えられる [157] . 具体例を交えながらより細かく分類すると ,

- フィードフォワード補正
 - 粗い目標動作指令の補間
 - * 関節位置軌道補間
 - * ステップ目標に対する歩行軌道生成などの動的動作生成
 - 実行に適した動作への修正
 - * 重心位置や ZMP の補正
 - * 冗長自由度の最適化
 - アクチュエータ出力への変換
 - * 関節位置軌道から関節発揮力への変換
 - * 関節位置軌道から冗長駆動筋骨格アクチュエータ出力への変換
- フィードバック補正
 - 力学特性の再現
 - * インピーダンス制御・アドミタンス制御・剛性制御
 - * 接触力制御
 - 外乱の中での動作継続
 - * 姿勢安定化
 - * プッシュリカバリ
- フェイルセーフ補正
 - 緊急動作停止
 - * 動作計画器からの緊急停止指令
 - * フィードバックによる異常検出
 - 異常出力緩和
 - * アクチュエータ出力指令と現在値の乖離量を制限

といったものが挙げられる . このうちフィードバック補正や一部のフェイルセーフ補正の機能が外乱応答を担う .

ロボットの動作制御計算の多くは , デカルト空間・関節空間・アクチュエータ空間の座標系変換によって成り立っている . これは主に ,

- 順運動学
- 逆運動学

表 5.1 Multi-layer response path in hierarchical robot system

大分類階層	機能ブロック階層	応答内容
外部入力	ユーザ	状況判断
	外部コントローラ	状況判断
中枢全身制御系	行動計画系	状態に基づく計画更新
	動作計画系	状態に基づく計画更新
	動作制御系	フィードバック動作制御
	ハードウェア制御系	反射応答
組込分散ネットワーク	ノード間直接通信	反射応答
分散制御ノード	MCU	サーボ制御
	FPGA	アクチュエータ出力制御
受動的応答	アクチュエータ出力回路	電氣的応答
	機械構造	機械的応答

- 微分運動学
- 順動力学
- 逆動力学

といったモデルに基づく計算に支えられている．それぞれの空間で表現されたベクトルを相互変換するため，その多くでは行列計算が基礎となっている．ロボットが多自由度化すると行列計算の次数もしくは必要回数が増加する．

5.3 多階層から構成されるロボット体内分散制御系

分散制御ノードと中枢全身制御系の組み合わせによる多自由度ロボットの体内分散制御系は，階層的な機能ブロックにより構成される．ロボットに対する物理的な環境入力は階層ごとに存在する応答経路により，ロボットの物理的な身体運動に反映される．主な機能ブロック階層ごとの外乱応答を表 5.1 にまとめる．通常，上位の階層になるにつれて応答性は低くなる．

本研究では中枢全身制御系の動作制御系にあたる一部の実時間応答機能を，中央計算機だけでなく組込計算機も利用して複合的に実装する．これにより動作制御系が更に多階層化されて，組込計算機を利用した応答性の高いフィードバック動作制御が可能となる．

FPGA SoC と GPU SoC を搭載した組込中間層計算機モジュールを導入して図 5.2 に示す制御系を構成する．図 5.2 では 2.7.2 項に述べたヒトの神経系との対比も交えてい

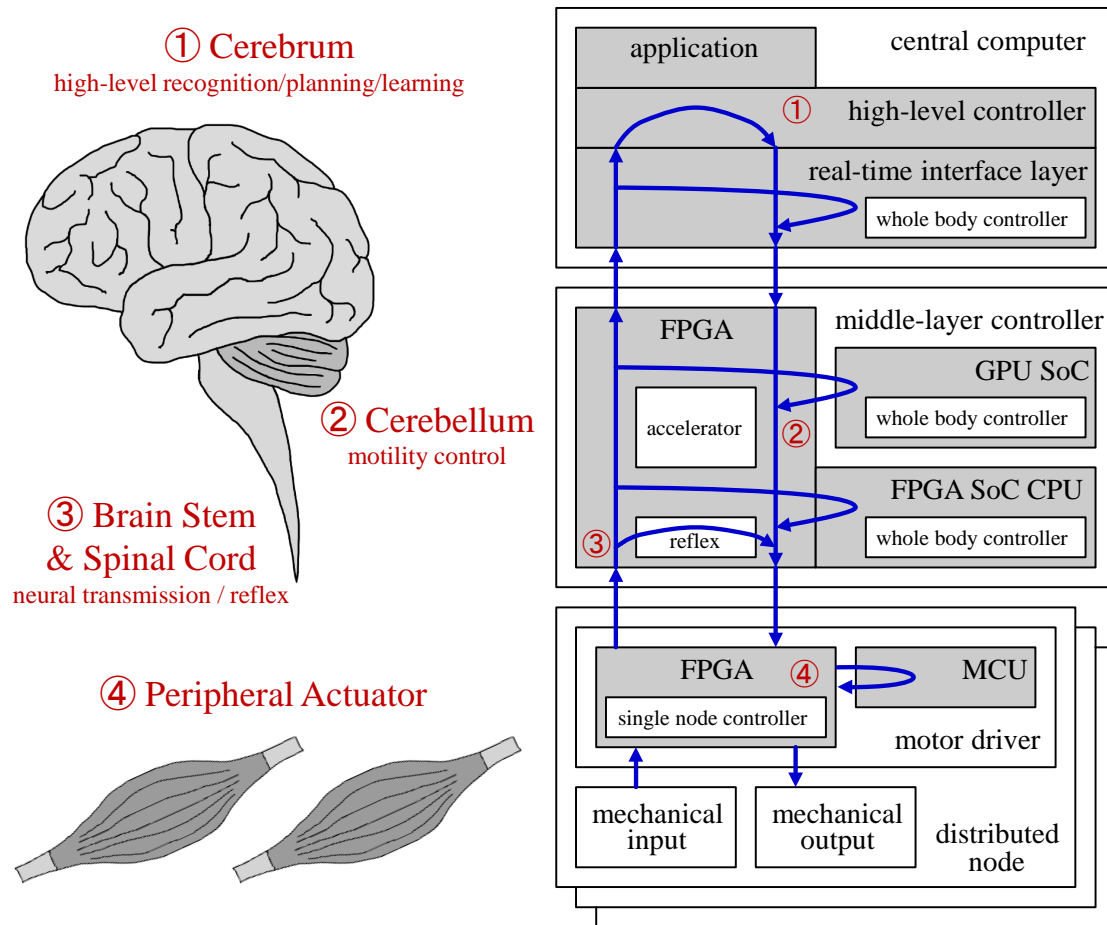


図 5.2 Comparison between the hierarchical control system and human nervous system

る．大脳皮質に相当する中央計算機では，状態把握や状況判断に基づく高次の行動・動作計画更新が行われる．中央計算機の実行系ソフトウェア層から組込計算機の GPU SoC および FPGA SoC に渡って多階層に，小脳や脊髄に相当する実時間動作制御系が構成される．分散制御ノードでは FPGA および MCU を利用して，中枢全身制御系からの遠心性投射に従ってアクチュエータに所望の特性を再現し，求心性投射によって現在の状態を中枢全身制御系に通知する．

最も低遅延なフィードバック経路は分散制御ノードでローカルに実行されるサーボ制御である．アクチュエータごとに位置や力をフィードバックして所望の力学特性を再現する．中枢全身制御系における制御構造や入出力データもアクチュエータ制御方式により大きく左右される．FPGA ではプログラブル論理回路による時間確定的な制御計算を実行し，非常に低遅延・低ジッタな応答性能を提供する．MCU を利用すると，FPGA による専用の回路実装と比べて応答性は低下するものの，von Neumann 型の計算機アーキテ

クチャで複雑な逐次計算へ拡張することができる。

中枢全身制御系ではネットワークを經由して全身フィードバック動作制御が行われる。典型的には中央計算機で実行系ソフトウェアが展開される。複雑化するロボットの動作制御系はマルチプロセスなソフトウェア群から構成され、高度な制御モデルの実装と引き換えに遅延は累積する。中央計算機における実行系ソフトウェアは計画系など更なる上位層への橋渡しも担う。

組込中間層計算機ではより応答的な全身フィードバック動作制御系を実装することができる。場合により FPGA や GPU によるハードウェアアクセラレーションも交えて、行列を基礎としたロボット動作制御計算を高速実行する。FPGA のネットワーク回路で完結する更に高速な反射応答への応用も可能である。

5.4 分散制御ノードのローカル低遅延フィードバック制御機構

5.4.1 モータ制御モジュールの基礎サーボシステム構成

本研究で開発するモータ制御モジュールには図 2.6 (c) に示した位置制御モデルを基本としたモータ制御システムを実装する。図 5.3 に基本となるモータ制御システム構成を示す。FPGA と MCU によって制御系が構成され、ブラシレスモータを駆動する。

FPGA にはブラシレスモータのベクトル電流制御器を実装する。ベクトル電流制御器は概ね、

1. UVW 相電流を測定して Clark-Park 変換により dq 軸電流に変換
2. dq 軸電流の PI 制御で dq 軸電圧のフィードバック項を決定
3. 非干渉化および逆起電力補償により dq 軸電圧を補正
4. dq 軸電圧を逆 Clark-Park 変換により UVW 相電圧としてモータに出力

という要素で構成される。モータに電圧を出力する PWM 周波数はモータのリプル損失と MOSFET のスイッチング損失のトレードオフにより 78 kHz と設定する。周波数を低くするとモータのリプル損失が増大し、高くすると MOSFET のスイッチング損失が増大する。dq 軸電流の PI 制御は PWM 周波数と同程度のオーダとなるように 42.479 kHz で実行する。

FPGA にはまた、モータのロータリエンコーダ入力から位置情報を得て、離散微分およびフィルタリングにより速度を求める位置検出器を実装する。離散微分はエンコーダカウンタ値の後退差分により実装する。後退差分の取得周期はエンコーダの分解能をもとに必要となる速度の範囲および分解能を満たす程度に設定する。速度のローパスフィル

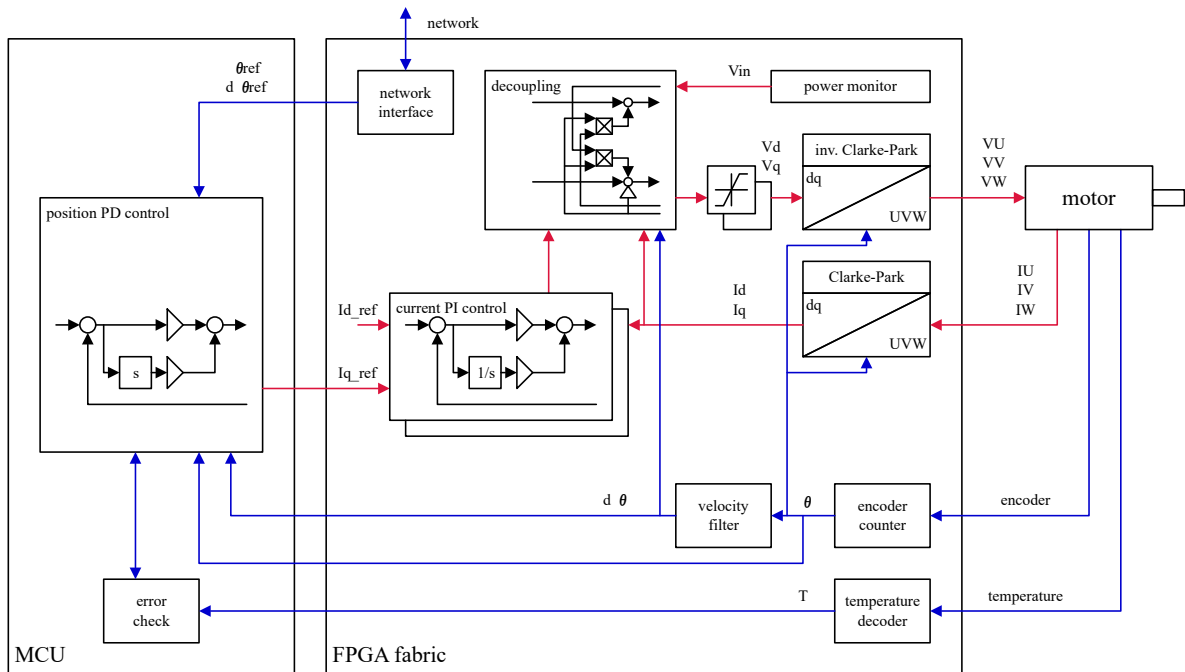


図 5.3 Position feedback control system in motor driver

タには，必要な周波数特性に加えて利用可能な FPGA 論理回路規模を考慮して等リプル FIR フィルタや低次の IIR フィルタから選択して採用する．代表的な構成例として，1 周あたり 2000 カウントのエンコーダに対して 24 MHz で 4096 クロックに一度，すなわち $170.667 \mu\text{s}$ 周期で後退差分を取得し，カットオフ周波数 176 Hz で減衰域-50 dB の等リプル 10 次 FIR フィルタで平滑化する．これにより 16 bit で最小分解能 2.93 rpm，有効範囲 ± 96000 rpm の速度値を得る．

MCU には位置制御器が実装される．FPGA から位置・速度の情報を得て，PD 制御により q 軸電流指令値を決定して FPGA に出力する．MCU の制御器はソフトウェアによって，位置制御以外に直接電流制御など他のモデルに切り替えることができる．

MCU による制御は 1 kHz で実行される．主に MCU と FPGA の間でのデータ交換時間によって制御周期に制約が生じる．1 ms 以下の制御周期での低遅延フィードバック制御に向けては分散制御ノードの MCU 制御周期が律速となる．その場合は図 5.4 のように FPGA 上に位置 PD 制御ループを実装する．位置と速度を独立に入力する PD 制御は順序回路を必要とせず組み合わせ回路により構成できるため律速とはならず，分散制御ノードの応答性はそれ以外の電流制御および速度フィルタによって決定される．

位置 PD 制御を MCU と FPGA のそれぞれに実装して比較する．図 3.4 と同様に，筐体に固定した単体無負荷のブラシレスモータに図 3.21 の小型光通信モータ制御モジュール SU-05-OPT を接続した装置を用いる．実験装置および制御実行周波数の構成を図 5.5

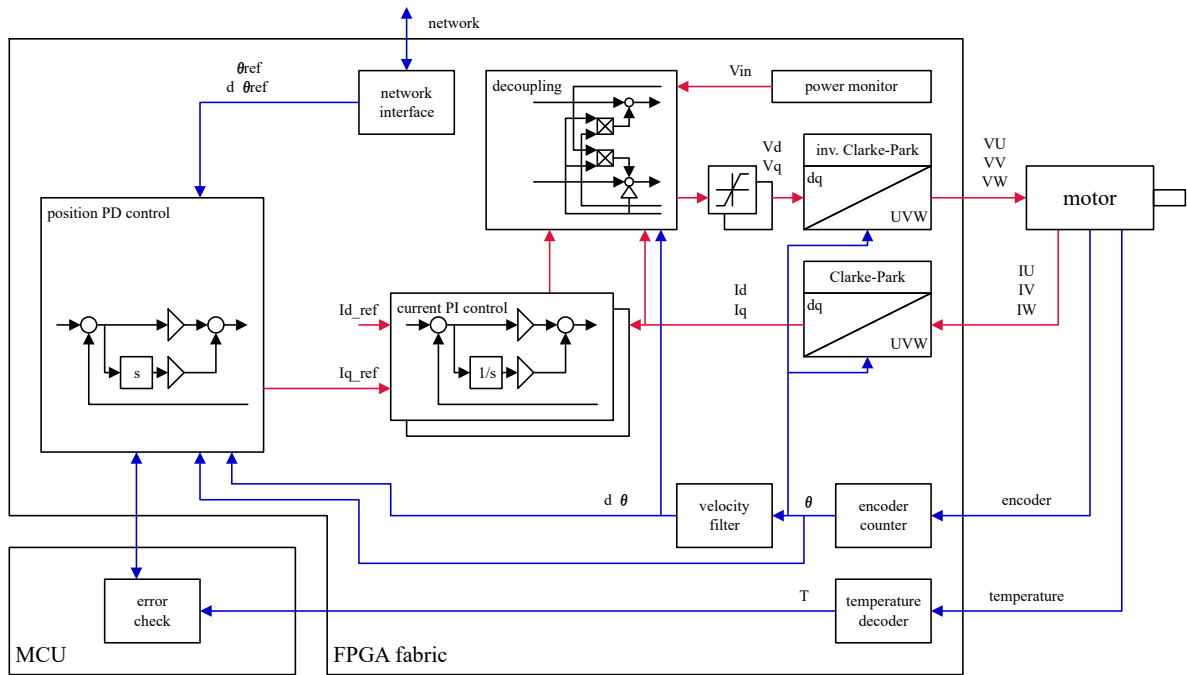


図 5.4 Position feedback control system in motor driver (position PD control implemented on FPGA)

に示す。

maxon motor 社の EC-4pole 30 200W ブラシレスモータと 1 周分解能 2000 のインクリメンタルエンコーダを用いる。シリアル通信変換基板を経由してエンコーダ情報を 228.571 kHz でモータ制御モジュールに入力する。8 MHz の RS-422 信号を用いて 35 クロックで送信している。位置 PD 制御計算は MCU で 1 kHz、もしくは FPGA で 24 MHz で実行する。その他の制御周期設定は前述の通りである。

モータ制御モジュールにおける位置 PD 制御の応答性向上は、安定性および指令追従性の改善に繋がる。位置 PD 制御のゲインに対する制御性能を評価する。

比例ゲインと微分ゲインを変更した際のサーボ制御の安定性を確認する。固定の位置指令に対して外乱で初速を与えた場合に、大きく発散的な振動が生じた条件を不安定と判定する。微分ゲインの符号は比例ゲインと同符号で粘性を生じる方向である。

比例ゲインに対して微分ゲインが過小である場合は減衰が不足して振動する。機械的なばねダンパ系を再現する理想的な位置 PD 制御の場合は正の微分ゲインを与えている限り振動は減衰する。しかし実際には、むだ時間や遅れが存在するために振動は発散する。比例ゲインや微分ゲインが過大である場合も同様に、むだ時間が原因となり振動は発散する。

実験結果を図 5.6 に示す。MCU で 1 kHz の位置 PD 制御を実行する場合と比べて、

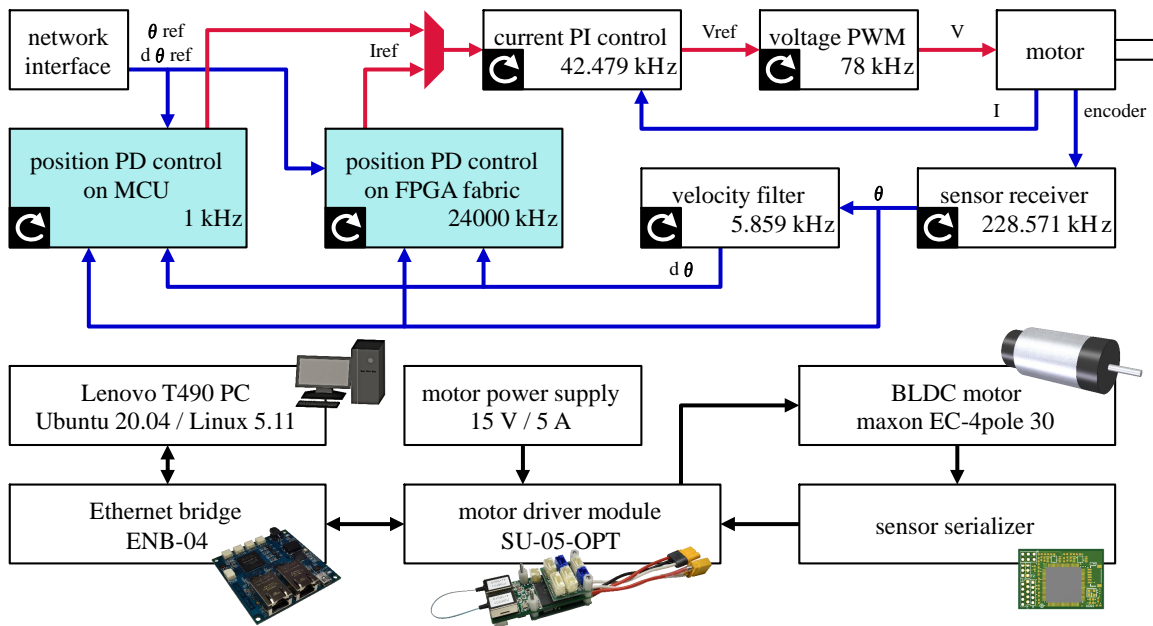


図 5.5 Experimental configuration for position PD control on MCU and FPGA

FPGA で実行すると安定限界が大きく拡張されていることが確認できる．比例ゲインは最大でおよそ 5 倍，微分ゲインは最大でおよそ 4 倍の値まで大きくすることができる．これにより，

- 比例ゲインを大きく設定した剛性および追従性の高い位置制御
- ゲイン余裕を大きく設定したロバスト性の高い安定制御

が容易に実現される．

いくつかのゲイン設定のもとで，中央計算機からステップ状の位置指令を与えて応答を確認する．ステップ入力の大きさはモータ 1 回転分の 2π rad とする．中央計算機の指令送信およびデータ計測は $500 \mu\text{s}$ 周期で実行する．実験結果を図 5.7 に示す．中央計算機でステップ状の位置指令を送信してからおよそ 2 ms 遅れて，これが反映されたデータが返ってきている．

図 5.7 (a) は MCU でゲイン余裕を大きく取った設定である．モータ回転子の慣性モーメント $3.33 \times 10^{-6} \text{ kgm}^2$ より，粘弾性ゲインの設定値から求められる理論上の減衰比は $\frac{0.769 \times 0.6}{1000} / \left(2 \times \sqrt{3.33 \times 10^{-6} \times \frac{187.5 \times 0.2}{1000}} \right) = 0.653$ である．僅かにオーバーシュートするステップ応答が期待される．FPGA で実行した場合はおよそ期待通りの滑らかな応答が得られる．一方で，MCU で実行した場合は歪な波形で減衰している．これは位置 PD 制御器から出力される電流指令値が 1 ms 単位の粗いステップ状となるため， 42.479 kHz の

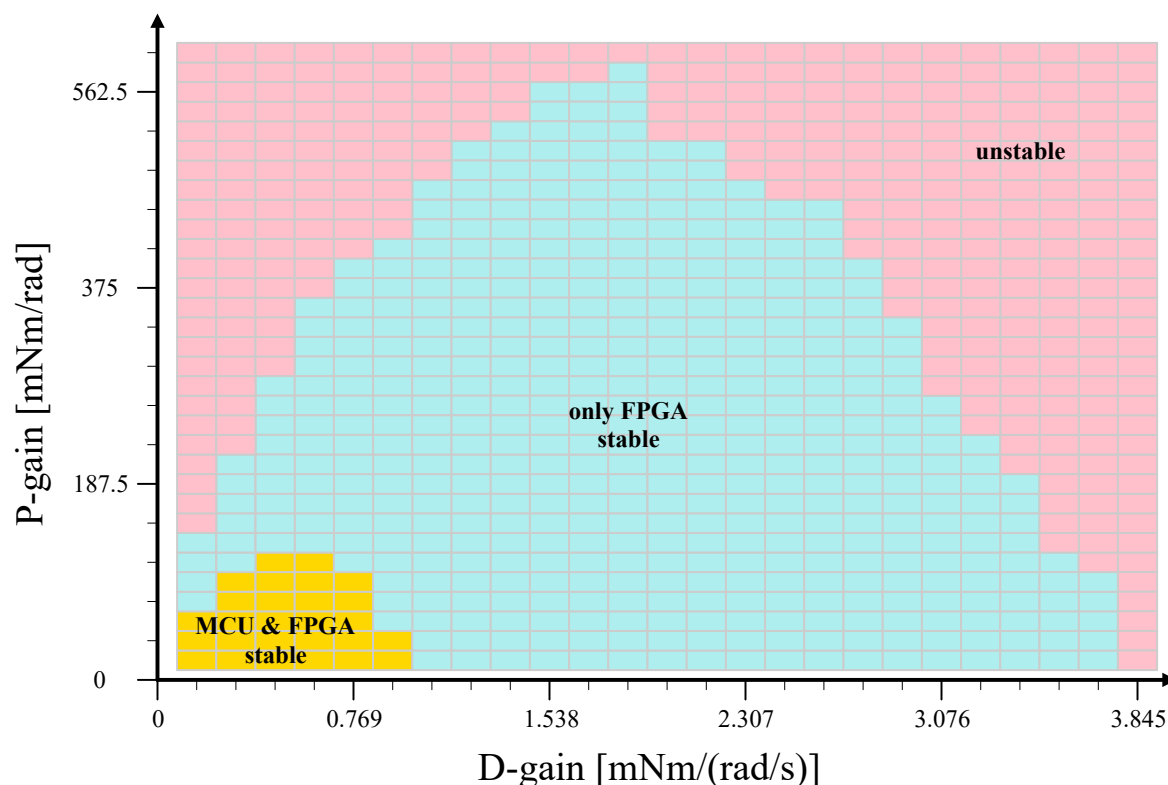


図 5.6 Shmoo plot for stability of position PD control on MCU and FPGA

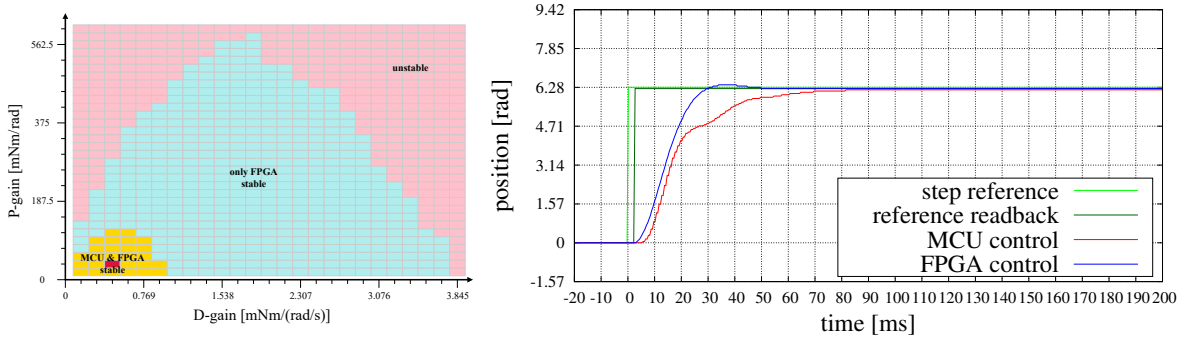
電流 PI 制御の応答により減速トルクが不規則にオーバシュートしているためである。位置 PD 制御を FPGA で実行することで、電流指令値が滑らかに生成され、高ゲインな電流 PI 制御の挙動を改善することができる。

図 5.7 (b) では MCU の限界まで比例ゲインを大きく設定している。理論上の減衰比は 0.377 である。MCU で実行した場合は長時間振動が継続している。

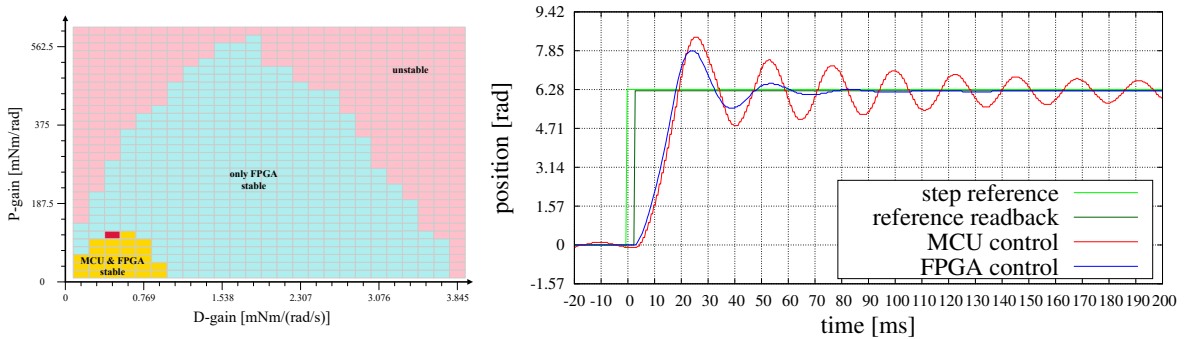
図 5.7 (c) では MCU の限界を超えた比例ゲインを設定している。理論上の減衰比は 0.292 である。FPGA では正常に減衰しているのに対し、MCU では減衰せず $\pm \frac{\pi}{2}$ rad の大きな振動が発生している。

図 5.7 (d) では MCU の限界を超えた微分ゲインを設定している。理論上の減衰比は 2.176 である。FPGA では正常に過減衰しているのに対し、MCU では振動が発生している。

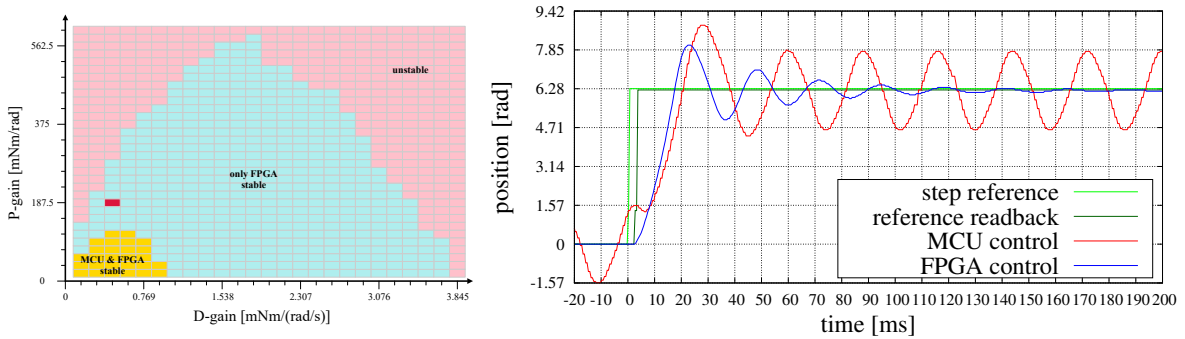
位置 PD 制御を FPGA で実行することで、MCU の場合と比べてより広範囲のゲインで安定した滑らかな挙動を示すことが確認できる。



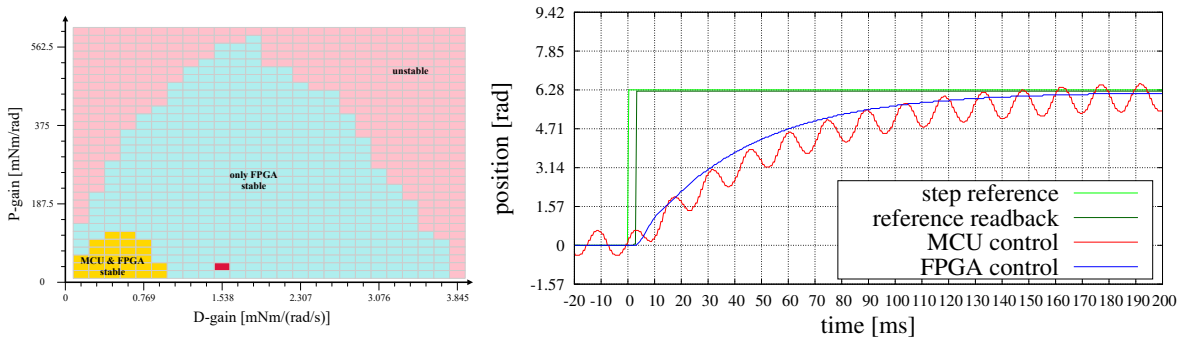
(a) Margined gain for MCU (187.5×0.2 mNm/rad, 0.769×0.6 mNm/(rad/s))



(b) Maximum P-gain for MCU (187.5×0.6 mNm/rad, 0.769×0.6 mNm/(rad/s))



(c) Too large P-gain for MCU (187.5×1.0 mNm/rad, 0.769×0.6 mNm/(rad/s))



(d) Too large D-gain for MCU (187.5×0.2 mNm/rad, 0.769×2.0 mNm/(rad/s))

図 5.7 Step response of position PD control on MCU and FPGA

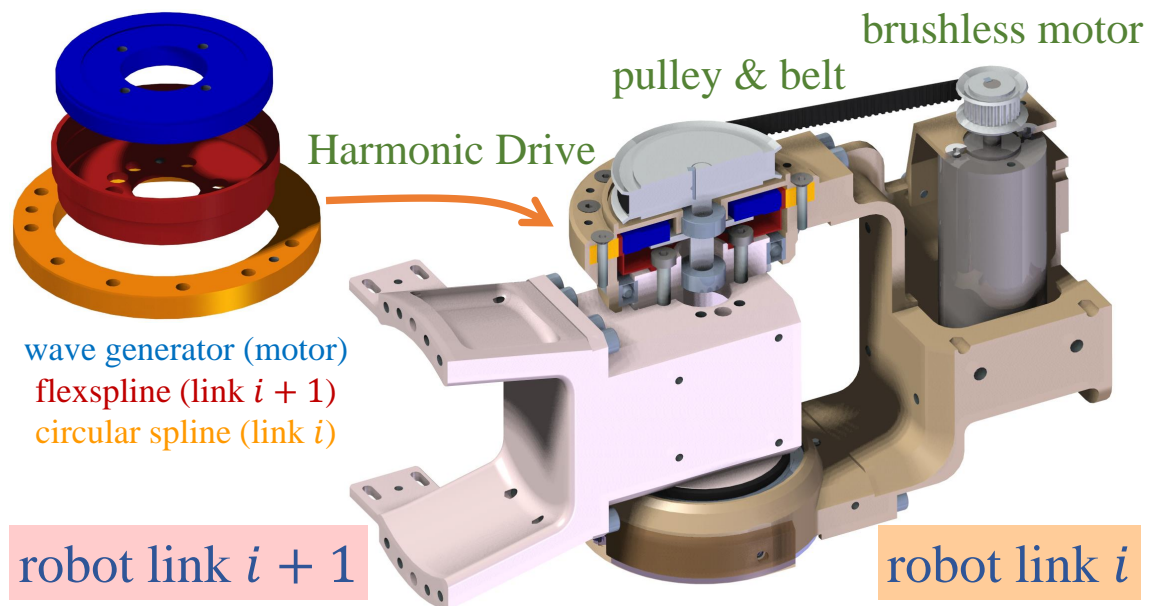


図 5.8 Robot joint with brushless motor and Harmonic Drive

5.4.2 ギア駆動型ロボットの関節トルク推定

図 5.8 に例示されるような大減速比の関節構造を有するギア駆動型ロボットでは，モータの慣性や減速機の摩擦が関節トルクに大きな影響を与え，モータ出力トルクと関節トルクには大きな差が生じる．関節トルクに基づく制御を行うには，関節トルクを測定もしくは推定する必要がある．トルクセンサを搭載せず関節トルクが直接測定できないロボットにおける関節トルクの推定には，

- モータ出力トルクおよびモータ・関節の運動から外乱オブザーバにより推定
- モータ出力トルクおよびモータ・関節の運動から関節力学モデルにより推定
- モータと関節の相対運動から関節力学モデルにより推定

といった方法が考えられる．外乱オブザーバによる手法は関節力学モデルのパラメータ同定を必要とせず汎用性は高いものの，オブザーバに用いるロボットモデルが全身の動力学に基づくためアクチュエータごとのローカル制御には利用できない．本研究ではアクチュエータごとに低遅延なフィードバックを行うため，関節の摩擦モデルに基づく簡易的な推定手法を用いる．

ベースリンクに加え n リンクからなるリンクロボットのラグランジュ法による運動方

程式は,

$$M(q)\ddot{q} + h(q, \dot{q}) + g(q) = \tau \quad (5.1)$$

$$M(q) = \sum_{i=1}^n \left(m_i J_P^{(i)T} J_P^{(i)} + J_O^{(i)T} \hat{I}_i J_O^{(i)} \right) \quad (5.2)$$

$$h(q, \dot{q}) = \dot{M}(q)\dot{q} - \frac{\partial K}{\partial q} \quad (5.3)$$

$$g(q) = \frac{\partial P}{\partial q} \quad (5.4)$$

$$\dot{s}_i = J_P^{(i)} \dot{q} \quad (5.5)$$

$$\omega_i = J_O^{(i)} \dot{q} \quad (5.6)$$

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (5.7)$$

$$P = \sum_{i=1}^n m_i \mathbf{g}^T s_i \quad (5.8)$$

と表せることがよく知られている [158]. これは関節 $1, 2, \dots, n$ の変位 $q = [q_1, q_2, \dots, q_n]^T$ とトルク $\tau = [\tau_1, \tau_2, \dots, \tau_n]^T$ の関係を記述している. その他未定義の変数は,

- m_i : リンク i の質量
- \hat{I}_i : リンク i に関してリンク重心周りで基準座標系で記述した慣性テンソル
- s_i : リンク i の基準座標系での位置
- ω_i : リンク i の基準座標系での角速度
- g : 基準座標系での重力加速度

を表す. また各式の物理的意味に関して, $M(q)$ (式 5.2) は慣性行列と呼ばれ, 式 5.1 の第 1 項は慣性力, 第 2 項 (式 5.3) は遠心力およびコリオリ力, 第 3 項 (式 5.4) は重力をそれぞれ表す. 式 5.5 で定義される $J_P^{(i)}$ はリンク i の位置に関するヤコビ行列, 式 5.6 で定義される $J_O^{(i)}$ はリンク i の姿勢に関するヤコビ行列を表し, K (式 5.7) はロボット全体の運動エネルギー, P (式 5.8) はロボット全体のポテンシャルエネルギーである.

関節に働くトルク τ は, アクチュエータの出力や関節の回転抵抗成分などロボットの内部で発生する成分と, 環境接触等で生じた重力以外の外力による成分の合計であるが, 通常, 求めたい関節トルクとは関節に生じる内力やアクチュエータで発生すべき出力を指すため, これらは分けて考えられるべきである. リンク i に外力による並進力 f_i およびモーメント n_i が働いたとき, リンク i のヤコビ行列,

$$J_\omega^{(i)} = \begin{bmatrix} J_P^{(i)} \\ J_O^{(i)} \end{bmatrix} \quad (5.9)$$

を用いて，関節に働く外力トルク $\tau_{\text{ext}}^{(i)}$ は，

$$\begin{aligned}\tau_{\text{ext}}^{(i)} &= \mathbf{J}_{\omega}^{(i)T} \begin{bmatrix} \mathbf{f}_i \\ \mathbf{n}_i \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{J}_P^{(i)T} & \mathbf{J}_O^{(i)T} \end{bmatrix} \begin{bmatrix} \mathbf{f}_i \\ \mathbf{n}_i \end{bmatrix} \\ &= \mathbf{J}_P^{(i)T} \mathbf{f}_i + \mathbf{J}_O^{(i)T} \mathbf{n}_i\end{aligned}\quad (5.10)$$

と表せる．すなわち関節に働く合計の外力トルク τ_{ext} は，

$$\begin{aligned}\tau_{\text{ext}} &= \sum_{i=1}^n \tau_{\text{ext}}^{(i)} \\ &= \sum_{i=1}^n \left(\mathbf{J}_P^{(i)T} \mathbf{f}_i + \mathbf{J}_O^{(i)T} \mathbf{n}_i \right)\end{aligned}\quad (5.11)$$

となる．関節に働くトルクの残りはロボットの内部で発生する成分 τ_{int} であり，

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{\tau}_{\text{ext}} + \boldsymbol{\tau}_{\text{int}} \\ &= \sum_{i=1}^n \left(\mathbf{J}_P^{(i)T} \mathbf{f}_i + \mathbf{J}_O^{(i)T} \mathbf{n}_i \right) + \boldsymbol{\tau}_{\text{int}}\end{aligned}\quad (5.12)$$

となる．よって式 5.1 は改めて，

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \sum_{i=1}^n \left(\mathbf{J}_P^{(i)T} \mathbf{f}_i + \mathbf{J}_O^{(i)T} \mathbf{n}_i \right) + \boldsymbol{\tau}_{\text{int}}\quad (5.13)$$

と記述される．

図 5.8 の関節 $i+1$ の運動方程式は，全身の運動方程式式 5.1 の $i+1$ 行目を取り出して，

$$\mathbf{M}_{i+1}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}_{i+1}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_{i+1}(\mathbf{q}) = \tau_{i+1}\quad (5.14)$$

と表せる．このときの左辺のうち重力項以外を，重力を含めた関節 $i+1$ の総トルク $\tau_{i+1} - g_{i+1}(\mathbf{q})$ によって駆動されるべき動力的慣性要素，

$$\mathbf{D}_j^{(i+1)}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \mathbf{M}_{i+1}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}_{i+1}(\mathbf{q}, \dot{\mathbf{q}})\quad (5.15)$$

とする．

続いてモータの運動を考える．モータをリンク i の次に接続されたひとつのリンクであると考えるとモータの運動方程式も同じ形で表すことができる．ここでは便宜上同じ記号を使って $n = i+1$ リンクからなるロボットのリンク $i+1$ としてのモータ $i+1$ の運動を記述する．モータの軸周りの運動方程式は同様に，

$$\mathbf{M}_{i+1}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}_{i+1}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_{i+1}(\mathbf{q}) = \tau_{i+1}\quad (5.16)$$

と表せる．まず $M_{i+1}(\mathbf{q})$ について，式 5.2 でリンク 1 からリンク i までのヤコビ行列の $i+1$ 列目は零ベクトルであることから，

$$M_{i+1}(\mathbf{q}) = \left[m_{i+1} \mathbf{J}_P^{(i+1)T} \mathbf{J}_P^{(i+1)} + \mathbf{J}_O^{(i+1)T} \hat{\mathbf{I}}_i \mathbf{J}_O^{(i+1)} \right]_{i+1} \quad (5.17)$$

となる．更に，モータの原点を軸上にとると $\mathbf{J}_P^{(i+1)}$ の $i+1$ 列目も零ベクトルとなるため，

$$M_{i+1}(\mathbf{q}) = \left[\mathbf{J}_O^{(i+1)T} \hat{\mathbf{I}}_i \mathbf{J}_O^{(i+1)} \right]_{i+1} \quad (5.18)$$

である．次に $h_{i+1}(\mathbf{q}, \dot{\mathbf{q}})$ について，式 5.3 で $\frac{\partial K}{\partial q_{i+1}} = 0$ より，

$$\begin{aligned} h_{i+1}(\mathbf{q}, \dot{\mathbf{q}}) &= \dot{M}_{i+1}(\mathbf{q}) \dot{\mathbf{q}} \\ &= \sum_{j=1}^{i+1} \dot{q}_j \frac{\partial M_{i+1}(\mathbf{q})}{\partial q_j} \dot{\mathbf{q}} \\ &= \sum_{j=1}^{i+1} \dot{q}_j \frac{\partial \left[\mathbf{J}_O^{(i+1)T} \hat{\mathbf{I}}_i \mathbf{J}_O^{(i+1)} \right]_{i+1}}{\partial q_j} \dot{\mathbf{q}} \end{aligned} \quad (5.19)$$

となる．最後に $g_{i+1}(\mathbf{q})$ は式 5.4 で，

$$\begin{aligned} g_{i+1}(\mathbf{q}) &= \frac{\partial P}{\partial q_{i+1}} \\ &= 0 \end{aligned} \quad (5.20)$$

となる．以上よりモータの運動方程式は，

$$\left[\mathbf{J}_O^{(i+1)T} \hat{\mathbf{I}}_i \mathbf{J}_O^{(i+1)} \right]_{i+1} \ddot{\mathbf{q}} + \sum_{j=1}^{i+1} \dot{q}_j \frac{\partial \left[\mathbf{J}_O^{(i+1)T} \hat{\mathbf{I}}_i \mathbf{J}_O^{(i+1)} \right]_{i+1}}{\partial q_j} \dot{\mathbf{q}} = \tau_{i+1} \quad (5.21)$$

となる．このときの左辺をモータの動力的慣性要素，

$$D_m^{(i+1)} = \left[\mathbf{J}_O^{(i+1)T} \hat{\mathbf{I}}_i \mathbf{J}_O^{(i+1)} \right]_{i+1} \ddot{\mathbf{q}} + \sum_{j=1}^{i+1} \dot{q}_j \frac{\partial \left[\mathbf{J}_O^{(i+1)T} \hat{\mathbf{I}}_i \mathbf{J}_O^{(i+1)} \right]_{i+1}}{\partial q_j} \dot{\mathbf{q}} \quad (5.22)$$

とおくと運動方程式は $D_m^{(i+1)} = \tau_{i+1}$ となる．

式 5.21 の厳密な運動方程式には，モータの固定子から見た回転子の回転の慣性力に加え，モータ自体が移動することによる慣性力，遠心力，コリオリ力が含まれている．厳密な理解は重要であるが，一方で関節の減速比が大きい場合にはロボットの運動がモータの軸トルクに与える影響は微小であるとして無視しても問題ない場合が多い．すなわちモー

タ $i+1$ の回転速度 $q_{i+1} = \theta_m^{(i+1)}$ が支配的であるとして、 $\dot{\mathbf{q}} \simeq [0, 0, \dots, 0, \dot{\theta}_m^{(i+1)}]^T$ と近似する。行列 $\mathbf{J}_O^{(i+1)T} \hat{\mathbf{I}}_i \mathbf{J}_O^{(i+1)}$ の $i+1$ 行 $i+1$ 列成分はモータ $i+1$ 単体の慣性モーメント $I_m^{(i+1)}$ である。すると式 5.22 は、

$$\begin{aligned}
 D_m^{(i+1)} &\simeq I_m^{(i+1)} \ddot{\theta}_m^{(i+1)} + \sum_{j=1}^{i+1} \dot{q}_j \frac{\partial I_m^{(i+1)}}{\partial q_j} \dot{\theta}_m^{(i+1)} \\
 &= I_m^{(i+1)} \ddot{\theta}_m^{(i+1)}
 \end{aligned} \tag{5.23}$$

となり、モータ単体の回転慣性力による運動方程式に近似できる。

トルクの釣り合いを考える。関節番号を示す添字は省略する。減速比を R として、関節の慣性力 $D_j(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ およびモータの慣性力 $D_m \simeq I_m \ddot{\theta}_m$ による釣り合いの運動方程式は、

$$D_j(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + R D_m = R \tau_m - g(\mathbf{q}) + \tau_{\text{ext}} - \tau_{fj} \tag{5.24}$$

と表せる。式 5.13 より関節が受ける内力トルク $\tau_j = \tau_{\text{int}}$ は、

$$\tau_j = D_j(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + g(\mathbf{q}) - \tau_{\text{ext}} \tag{5.25}$$

であり、式 5.24 から

$$\begin{aligned}
 \tau_j &= R \tau_m - R D_m - \tau_{fj} \\
 &\simeq R \tau_m - R I_m \ddot{\theta}_m - \tau_{fj}
 \end{aligned} \tag{5.26}$$

となり、モータの出力トルクに加えてモータの慣性と減速機の摩擦が関節トルクに影響するモデルが導かれる。 τ_j は内力であるため符号は都合に合わせて適宜変更して $\tau_j = -\tau_{\text{int}}$ としてもよい。

5.4.3 ローカルフィードバックによる関節トルク制御

式 5.26 に基づいてモータ制御モジュール内にトルク制御器を構成することができる。減速機の摩擦を速度に基づくクーロン粘性摩擦モデル

$$\tau_{fj} = K_{fc} \text{sgn} \dot{\theta}_j + K_{fv} \dot{\theta}_j \tag{5.27}$$

で近似すると式 5.26 は、

$$\tau_j \simeq R \tau_m - R I_m \ddot{\theta}_m - K_{fc} \text{sgn} \dot{\theta}_j - K_{fv} \dot{\theta}_j \tag{5.28}$$

となる。モータの出力トルク・速度・加速度をフィードバックすることで関節トルクの推定が可能となる。ブラシレスモータの出力トルクはおよそ q 軸電流に比例するため、電流値と相互変換して取り扱うことができる。

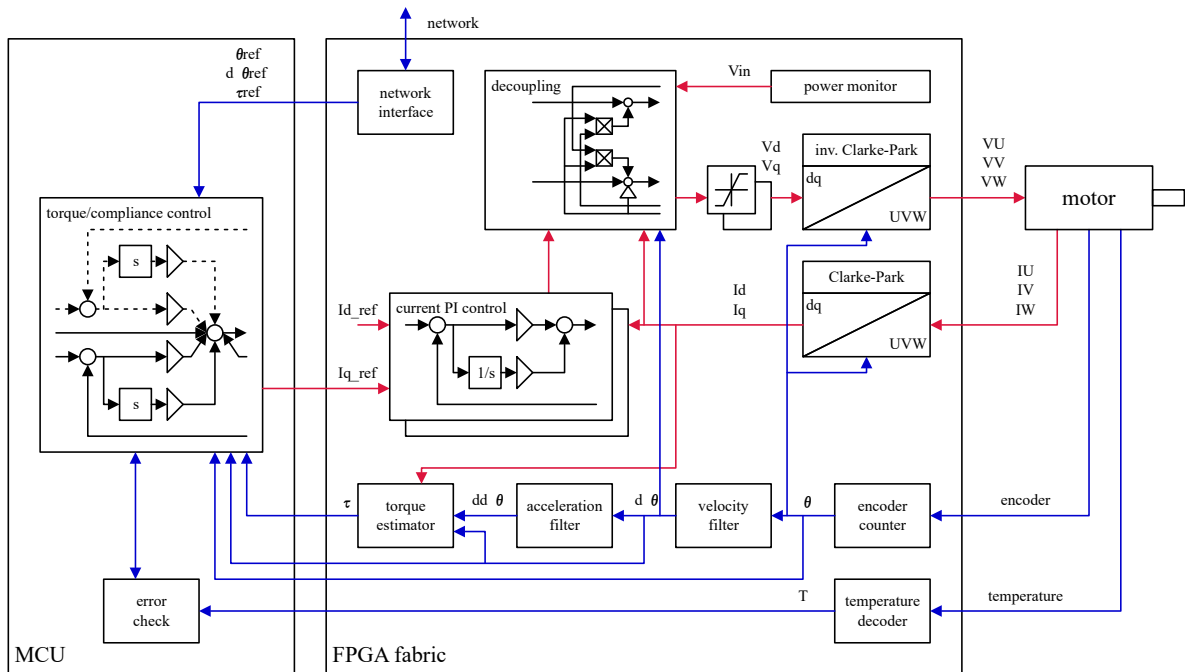


図 5.9 Torque estimation control system in motor driver

図 5.3 のモータ制御システムを図 5.9 のように更新する。

関節トルク推定は FPGA でクロック周波数単位で高速に実行される。速度・加速度のフィルタを FPGA に実装することで、MCU の制御周期よりも非常に細かい時間分解能が得られるため、応答性を自由に調整することができる。3.5.3 項の小型光通信モータ制御モジュールの例では、MCU の制御周期が 1 ms (1 kHz)、FPGA の基本クロックが 24 MHz となっている。モータのインクリメンタルエンコーダから得られる情報は粗く量子化されているため、そのままクロック周波数で離散微分すると疎らで値の大きなパルス状になってしまう。そのため速度は所望の最小分解能により求まる間隔で差分を取得して、クロック周波数でフィルタを適用する。一方加速度は、速度の周波数特性が上記のフィルタ構成により既知であるためクロック周波数で差分を取りフィルタを適用することができる。これにより応答性を最大限に高めた関節トルク推定が可能となる。推定された関節トルクは、MCU での制御に利用される以外に、体内分散ネットワークで共有される。また FPGA の電流制御器に直接フィードバックすることもできる。そのため推定の応答性向上は有意義である。

関節トルク推定に基づいて次のようなトルク制御器が実装される。

- 直接トルク制御：推定されたトルクのうち慣性や摩擦による損失分のみを補償した出力により関節に所望のトルクを発揮
- コンプライアンス制御：位置とトルクをフィードバックした複合 PD 制御により

関節に所望の剛性や粘性を発揮

- 衝撃緩和制御：関節トルク軌道から衝撃力の印加を検出してこれを緩和

直接トルク制御では式 5.28 の逆変換により，目標関節トルク τ_j^{ref} に対して慣性や摩擦を補償した，

$$\tau_m^{\text{ref}} = \frac{1}{R} \left(\tau_j^{\text{ref}} + RI_m \ddot{\theta}_m + K_{fc} \text{sgn} \dot{\theta}_j + K_{fv} \dot{\theta}_j \right) \quad (5.29)$$

をモータに出力する．なおトルクセンサを利用して関節トルク値が陽に得られる場合は摩擦トルクのオブザーバによって同様の補償制御が可能である [159][160]．実際の制御では位置追従項 $K_{p\theta} \tilde{\theta} + K_{d\theta} \dot{\tilde{\theta}}$ およびトルク補償重み w_τ を加えて，

$$\tau_m^{\text{ref}} = \frac{1}{R} \left(\tau_j^{\text{ref}} + w_\tau \left(RI_m \ddot{\theta}_m + K_{fc} \text{sgn} \dot{\theta}_j + K_{fv} \dot{\theta}_j \right) \right) + K_{p\theta} \tilde{\theta} + K_{d\theta} \dot{\tilde{\theta}} \quad (5.30)$$

として実装する．

コンプライアンス制御では位置とトルクを PD 制御型でフィードバックして重ね合わせ [161]，

$$\tau_m^{\text{ref}} = K_{p\theta} \tilde{\theta} + K_{d\theta} \dot{\tilde{\theta}} + K_{p\tau} \tilde{\tau}_j + K_{d\tau} \dot{\tilde{\tau}}_j \quad (5.31)$$

とすることで見かけ上の弾性 $\frac{RK_{p\tau} + 1}{RK_{p\theta}}$ を再現する．弾性のうちトルクと位置の釣り合いによる成分が $\frac{K_{p\tau}}{K_{p\theta}}$ ，位置フィードバックによる弾性が $\frac{1}{RK_{p\theta}}$ である．しかしむだ時間系の制御であるために関節トルクに関する比例ゲイン $RK_{p\tau}$ をおよそ 1 より大きく取ることができない．このままでは単純な位置制御のおよそ 2 倍までしか柔らかくすることができず，弾性と追従性を両立できない．そこで遅れ系の出力を追加してゲインの安定限界を拡張する．ゲイン K_L のローパスフィルタを通した出力を加えて，

$$\tau_m^{\text{ref}} = K_{p\theta} \tilde{\theta} + K_{d\theta} \dot{\tilde{\theta}} + K_{p\tau} \tilde{\tau}_j + K_{d\tau} \dot{\tilde{\tau}}_j + K_L \text{LPF} \left(\tilde{\theta} + \frac{K_{p\tau}}{K_{p\theta}} \tau_j \right) \quad (5.32)$$

とすることで，低周波な見かけ上の追従性を向上する．

トルクセンサを搭載した図 5.10 の単軸試験機を用いて，関節トルク推定および関節トルク制御の基本的な動作を確認する．実験装置の概観を図 5.11 に示す．

単軸試験機を，

- サーボオフ
- 高剛性な位置制御
- コンプライアンス制御

とした状態で手で外乱を加えた際の，トルクセンサ値と推定トルク値を図 5.12 に比較する．図 5.12 (a) のサーボオフの場合は推定トルク値のうち慣性項および摩擦項が支配的となる．図 5.12 (b) の位置制御の場合はモータ出力トルク項が支配的となる．図 5.12 (c)

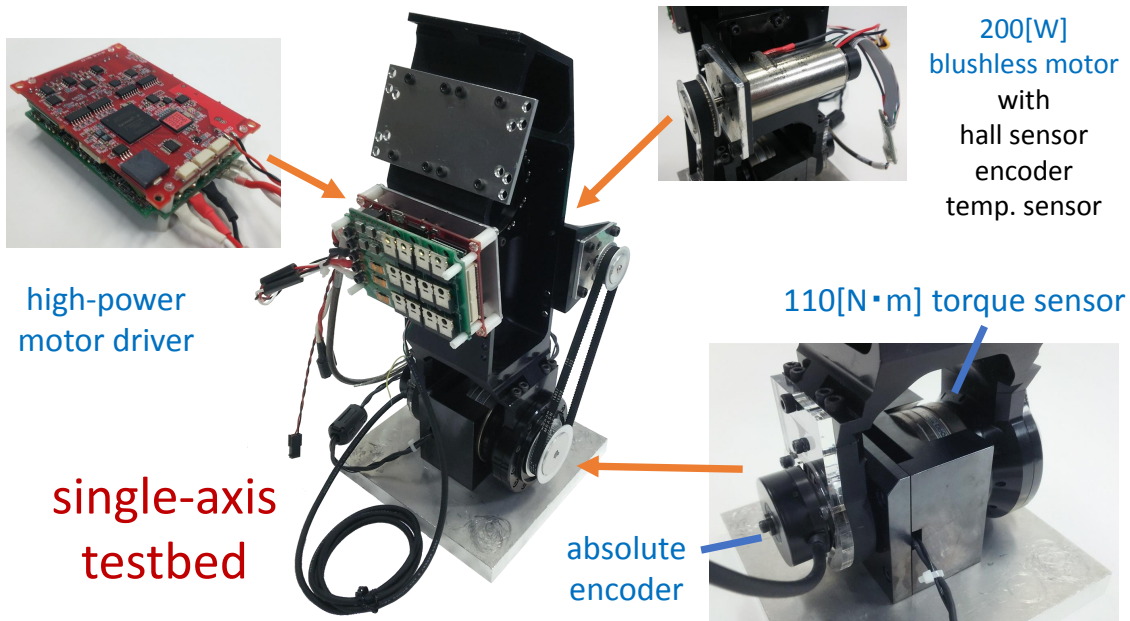


図 5.10 Overview of the single-axis testbed for joint torque control

のコンプライアンス制御の場合は図 5.12 (d) に示すとおり各項がそれぞれ大きな役割を持つ．いずれの場合も概ね正しいトルクが得られている．コンプライアンス制御の場合は一部誤差が生じている．これは主に単純なクーロン粘性摩擦の仮定に起因する．実際の減速機では回転状態や捻れに応じて粘性が変動するため，摩擦の推定においてやや精密性が損なわれる場合もある．

単軸試験機を直接トルク制御とした状態で手で外乱を加えた際の結果を図 5.13 に示す．0 Nm のトルク指令値に，位置の D 制御による粘性項を加えている．式 5.30 でパラメータを， $\tau_j^{\text{ref}} = 0 \text{ Nm}$ ， $w_\tau = 1$ ， $K_{p\theta} = 0 \text{ Nm/rad}$ ， $K_{d\theta} = 0.0001111 \text{ Nm/(rad/s)}$ と設定している．図 5.13 (a) はトルクセンサ値，推定トルク値，および速度から計算された位置の D 制御による粘性項の換算値を比較している．図 5.13 (b) は対応する位置軌道を表示している．測定されたトルクが，出力しているはずの粘性項とよく合致しており，適切に慣性項および摩擦項が補償されていることが確認できる．

コンプライアンス制御で LPF を通した遅れ系の高ゲイン出力により追従性が向上する効果を検証する．式 5.32 でパラメータを， $K_{p\theta} = 0.0002713 \text{ Nm/rad}$ ， $K_{d\theta} = 0.0001111 \text{ Nm/(rad/s)}$ ， $RK_{p\tau} = 0.9775 \text{ Nm/Nm}$ ， $RK_{d\tau} = 0 \text{ Nm/(Nm/s)}$ ， $K_L = 0.001085 \text{ Nm/rad}$ ，IIR フィルタによる LPF の時定数を 0.1984 s と設定する．トルクの比例ゲイン $RK_{p\tau}$ を 1 未満に設定している．LPF 項により追従性を表す低周波の位置ゲインは $K_{p\theta}$ から $K_{p\theta} + K_L$ でおよそ 5.0 倍となる．LPF 項を $K_L = 0$ として除く場合と有効化

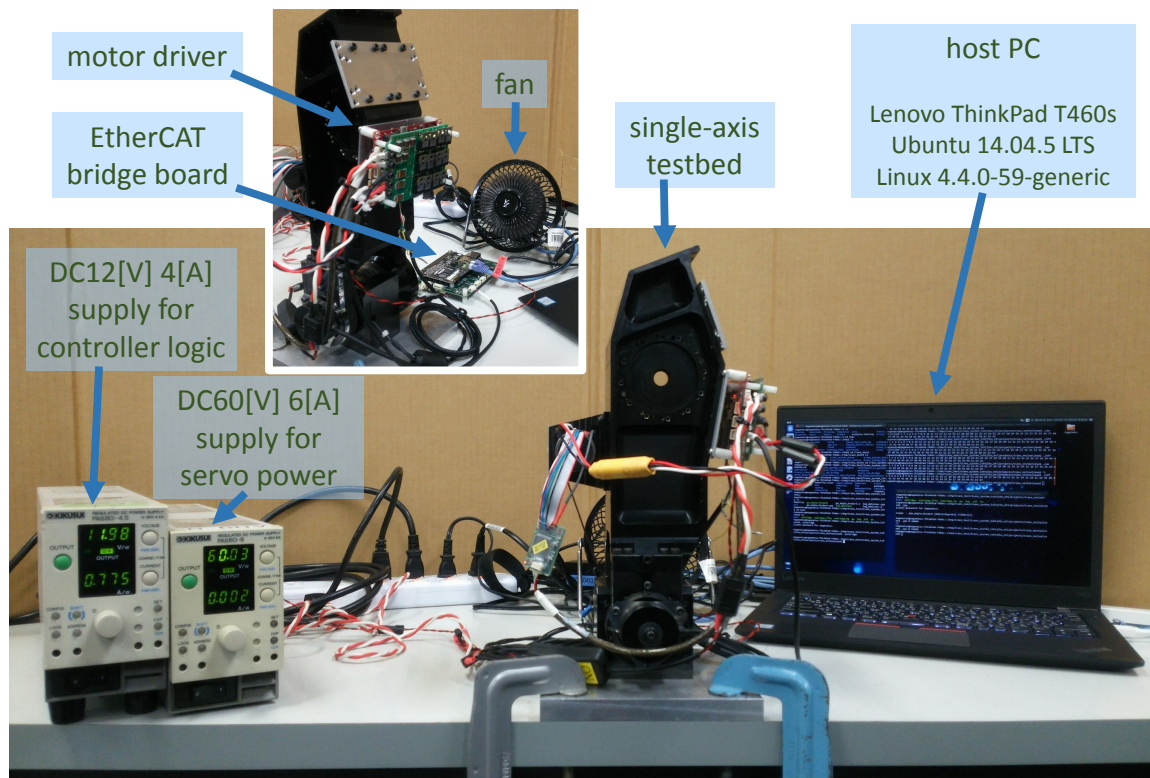


図 5.11 Experimental setup with the single-axis testbed

する場合のそれぞれの関節トルクに対する関節位置の挙動を図 5.14 に示す．LPF 項の効果により，トルク印加の程度に対して位置がよく追従していることが確認できる．

5.5 低遅延全身動作制御へ拡張可能なハードウェア制御系の一般化実装

5.5.1 ハードウェア制御系の共通構造

多自由度ロボットの全身制御を担う計算機は，様々なインターフェースおよびプロトコルによって組込分散ネットワークをはじめとしたロボットのハードウェアに接続して制御データを交換する．本研究で開発しているシステムの例では，中央計算機として用いられる高性能計算機製品が Ethernet や USB のインターフェースによって組込制御系と通信する．組込中間層計算機モジュールに搭載された CPU は PCI Express やオンチップバス接続によって FPGA とデータを交換する．力センサや姿勢センサなど，他のハードウェアが追加で計算機に接続される場合も多い．本研究で直接取り扱っていない他のロボットにも多くのハードウェア構成が存在する．また今後も技術の進展に伴い新たな構成の開発

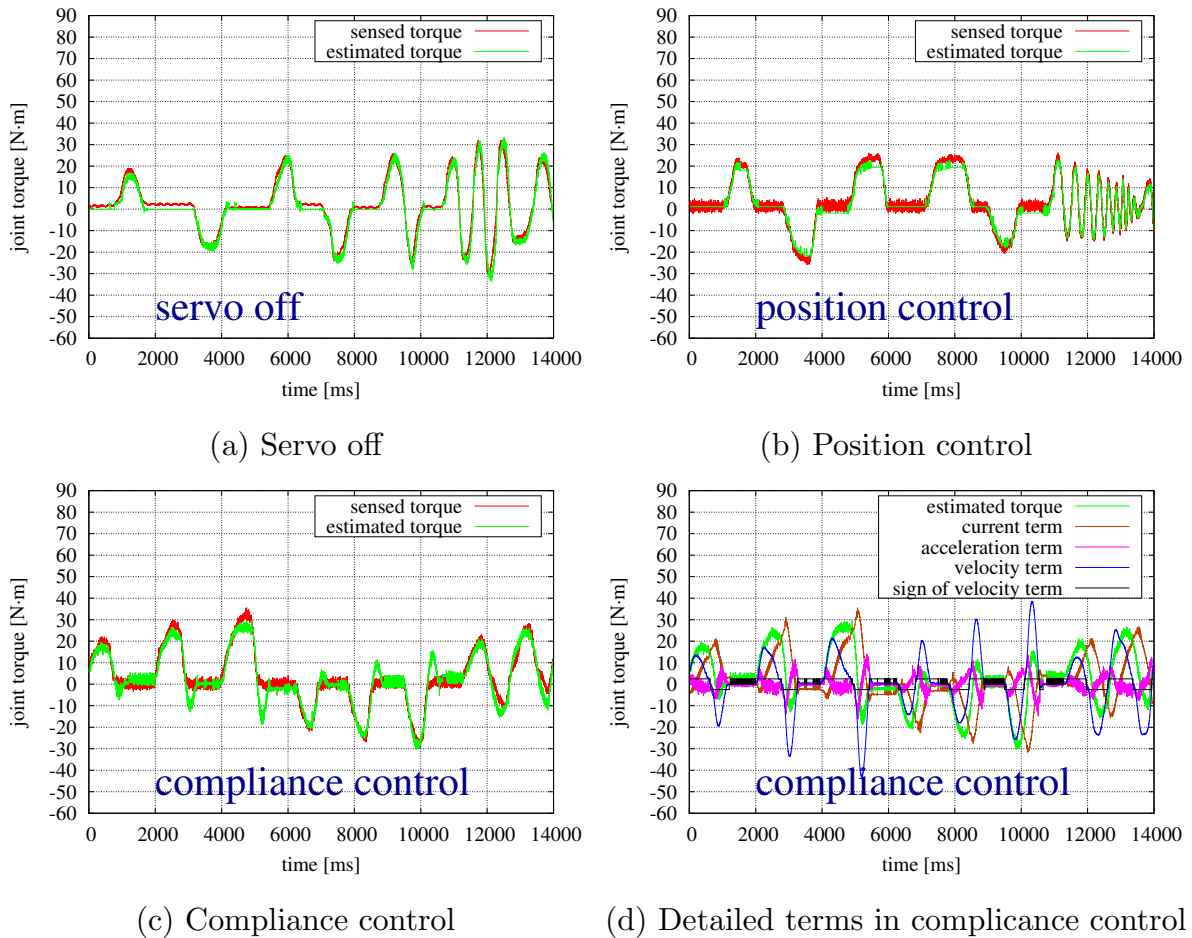


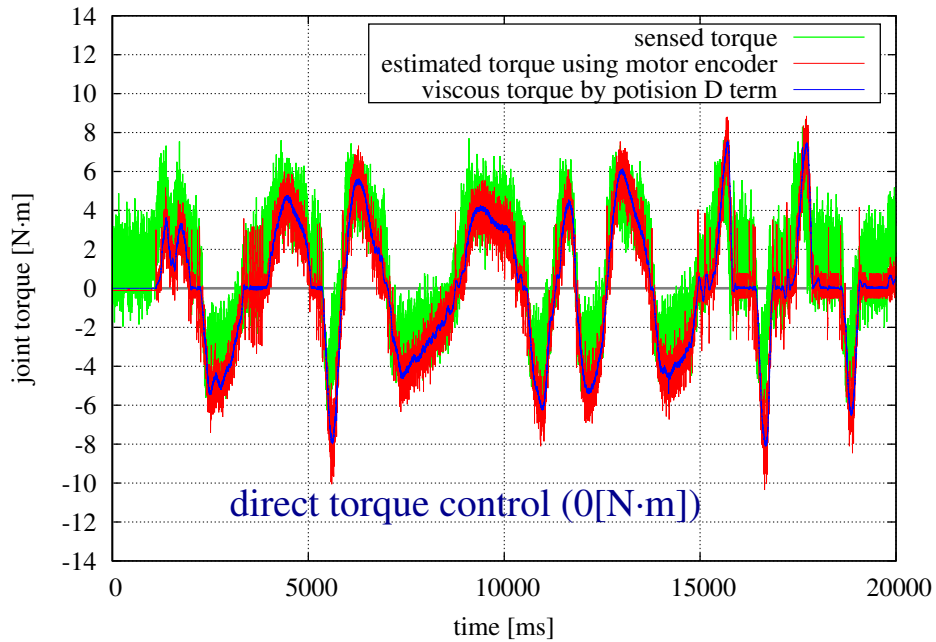
図 5.12 Sensed joint torque and estimated joint torque of the single-axis testbed

が見込まれる。

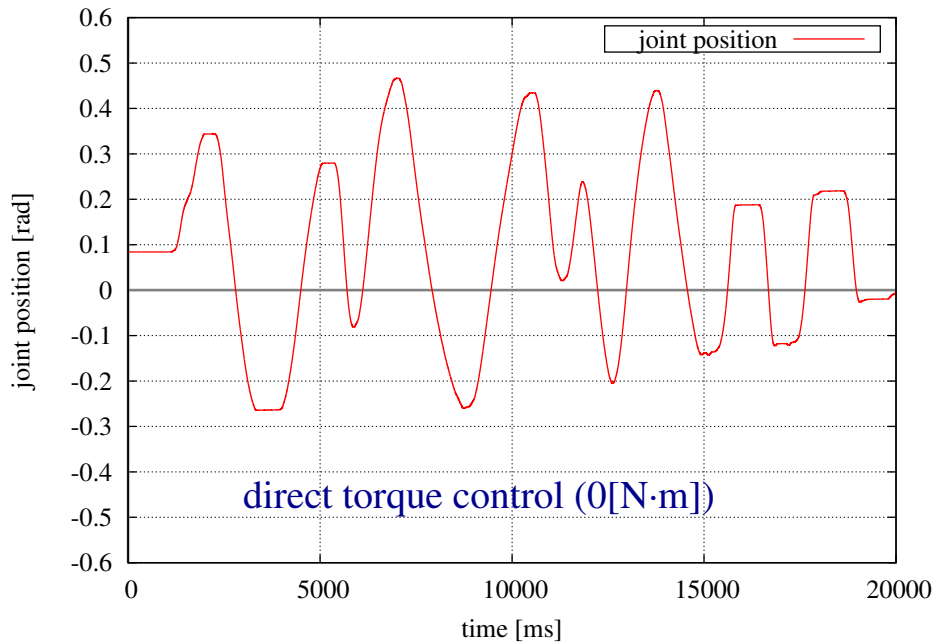
ロボットのハードウェア制御系を担うソフトウェアは、その実時間性の要求から、個別のハードウェア構成ごとに最適化して実装されるのが一般的である。しかしその実時間処理の中でフィードバック遅延を最適に低減するための、

1. ハードウェアからセンサ値や状態に関する最新のデータを収集
2. 収集したデータや他のソフトウェアからの入力进行处理して最新の指令データを生成
3. ハードウェアに指令データを送信

という周期処理構造はほとんどの場合で共通している。個別のハードウェア構成に向けては、通信プロトコルごとに異なる最適な手順に従って、ロボット構成ごとに異なる必要なデータを交換して処理する機能実装が必要となる。共通の周期処理構造に基づいて、個別のハードウェア制御系の機能を共通インターフェースで実装し提供することで、多様なロボットに共通のソフトウェアプラットフォームを適用することができる。

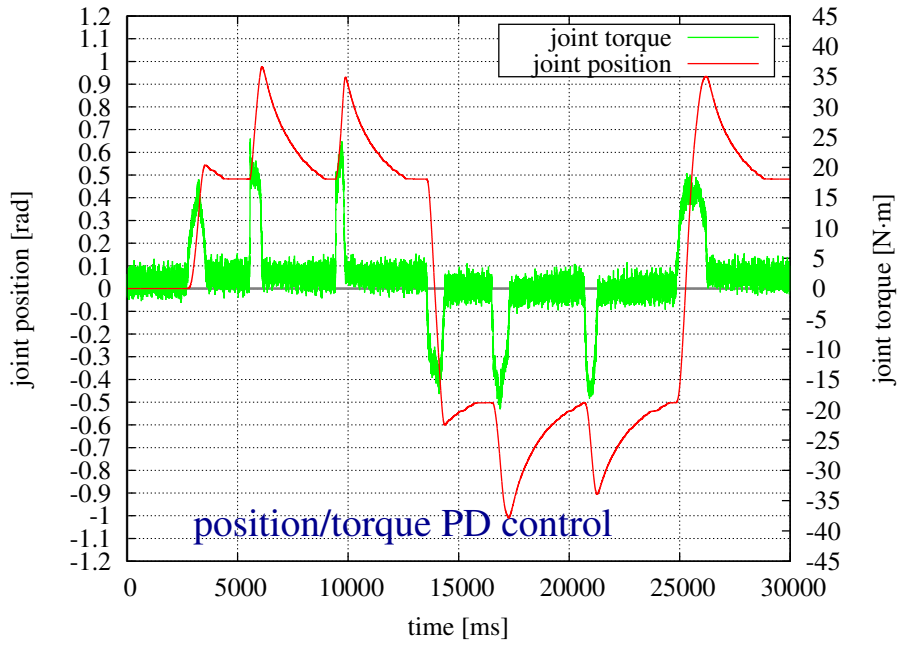


(a) Actual torque and viscous torque command calculated with joint velocity

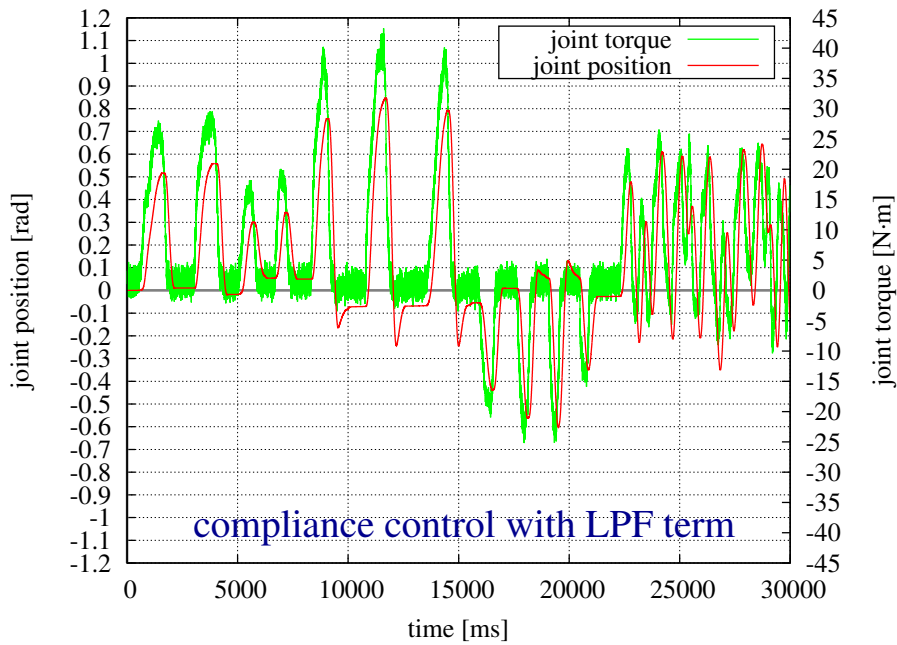


(b) Joint position

図 5.13 Joint torque of the single-axis testbed with direct torque control set as 0 Nm



(a) Only position/torque PD control



(b) Compliance control with the additional low-frequency term

図 5.14 Joint torque and joint position of the single-axis testbed with compliance control improved by the additional low-frequency term

5.5.2 ハードウェア制御系の従来実装における非同期構造の課題

ギア駆動型大出力ヒューマノイドに関する先行研究 [162] では、中枢全身制御系における実行系ソフトウェアが、OS や基礎的なフレームワークに加えて、

- PCI インタフェースで組込制御系とデータを交換する 1 ms 周期の主要ハードウェア制御系
- IMU や力センサの値を読み出す個別の補助ハードウェア制御系
- OpenRTM-aist や OpenHRP3 を利用した 2 ms 周期の動作制御系

から構成されていた。

図 5.15 に JAXON の実行系ソフトウェア構成を示す。全身の分散制御ノードを接続する組込分散ネットワークと中央計算機は PCI インタフェースで接続される。中央計算機の OS には Ubuntu Linux を利用している。PCI インタフェースでデータを交換するデバイスドライバは Linux のカーネルモジュールとして実装される。ハードウェア制御系では 1 ms 周期の制御ループを実行する。ハードウェア制御系の周期処理の中では、PCI デバイスドライバに対してシステムコールを発行することでハードウェアとデータを交換し、高速なプロセス間通信機構である共有メモリを利用して上位の動作制御系に中継する。ハードウェアの通信プロトコルに応じたデータ変換に加え、異常対応など一部の特殊なフィードバック処理がハードウェア制御系の中で実行される。組込分散ネットワークとは異なる経路で接続される IMU や力センサには、別途通信ソフトウェアを用意してセンサ値を共有メモリに書き込む。動作制御系は、OpenRTM-aist [135] のプラットフォーム上で OpenHRP3 [163][164] を利用して制御計算やデータ処理を行う HRPSYS [165] によって実装する。HRPSYS では一連の制御コンポーネント群が OpenRTM-aist の RTC Daemon により 2 ms 周期で実行される。ハードウェア制御を担うコンポーネントにおいて共有メモリとのデータ交換を実装する。

ハードウェア制御系と動作制御系の周期処理は異なるプロセスでそれぞれ非同期に実行される。このようなマルチプロセス非同期処理構造では大きなソフトウェア遅延が生じる。

実行系ソフトウェアの周期処理における遅延の模式図を図 5.16 に示す。図 5.16 (a) は一連の処理が同期して周期実行される場合を表す。制御周期を T_1 とする。それぞれの周期では、ハードウェアからセンサ値を読み出し、センサ値などの入力に基づいて出力指令値を決定し、ハードウェアに出力指令値を送信する、という 3 段階の処理が実行される。周期処理はジッタ無しで実行されるものとする。ハードウェアでセンサ値を取得して計算機から読み出される準備が整ってから、それが実際に周期処理で読み出されるまでの

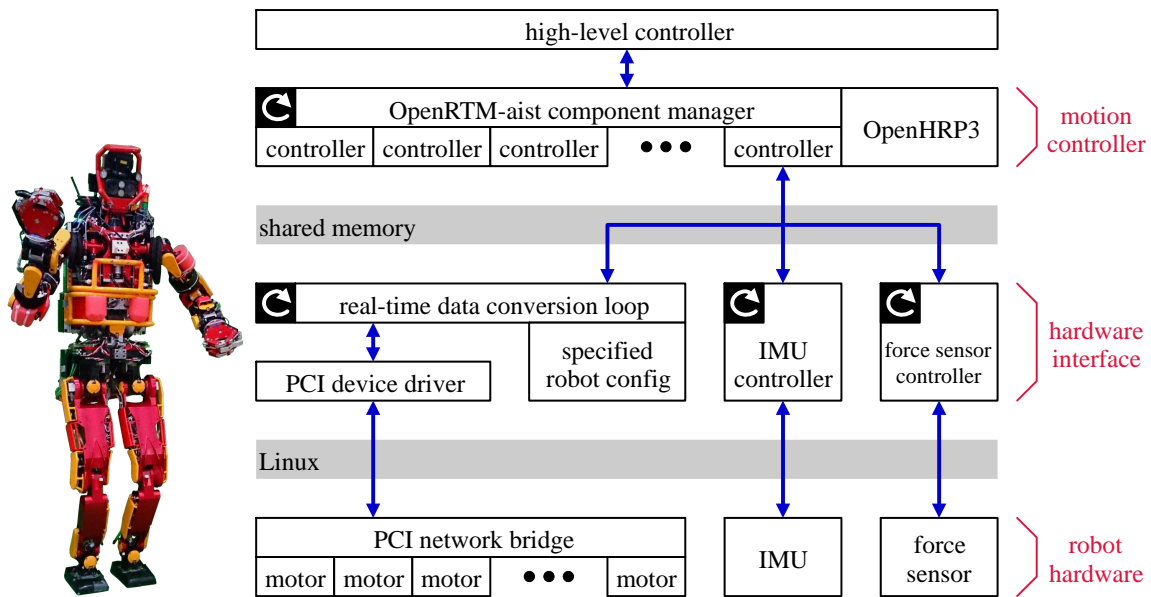


図 5.15 Motion controller software for the humanoid robot JAXON in previous work

最悪時間は制御周期 T_1 である．周期処理内で入力が出力に反映されるまでの最悪時間も同じく T_1 である．すなわちジッタ無しで同期処理が周期実行される場合の最悪応答時間は 2 周期分の $2T_1$ となる．一方で，図 5.16 (b) は複数の非同期周期処理が実行される場合を表す．センサ値を取得する入力タスク，入力から出力を決定する制御タスク，およびハードウェアに指令値を送信する出力タスクから構成される．ハードウェア制御系のソフトウェアが共通の場合入力タスクと出力タスクが同一のものとなる．それぞれのタスクにおいて，入力データが用意されてからそれを出力に反映するまでの最悪時間は，図 5.16 (a) と同様に 2 周期分となる．入力タスクの制御周期 T_1 ，制御タスクの制御周期 T_2 ，および出力タスクの制御周期 T_3 に対して，実行系ソフトウェア全体の最悪応答時間は累計 $2T_1 + 2T_2 + 2T_3$ となる．タスク間の通信に遅延が生じる場合は，更にこれも最悪応答時間に加算される．

JAXON の実行系ソフトウェアでは，ハードウェア制御系が 1 ms 周期，動作制御系が 2 ms 周期で実行される．そのため，ハードウェア制御系のみで完結する処理の最悪応答時間が $2 \times 1 = 2$ ms であるのに対し，動作制御の最悪応答時間は $2 \times 1 + 2 \times 2 + 2 \times 1 = 8$ ms に及ぶ．ここで周期処理のジッタは考慮せず，共有メモリにおけるプロセス間通信遅延も無視できるものとしている．

ハードウェア制御系は 1 ms 周期で低ジッタに実装されており実時間性を備えている一方で，PCI を利用した特定のハードウェア制御に最適化された構成となっている．本研究で PCI を置き換えるものとして EtherCAT 通信インタフェースを導入するにあたり，

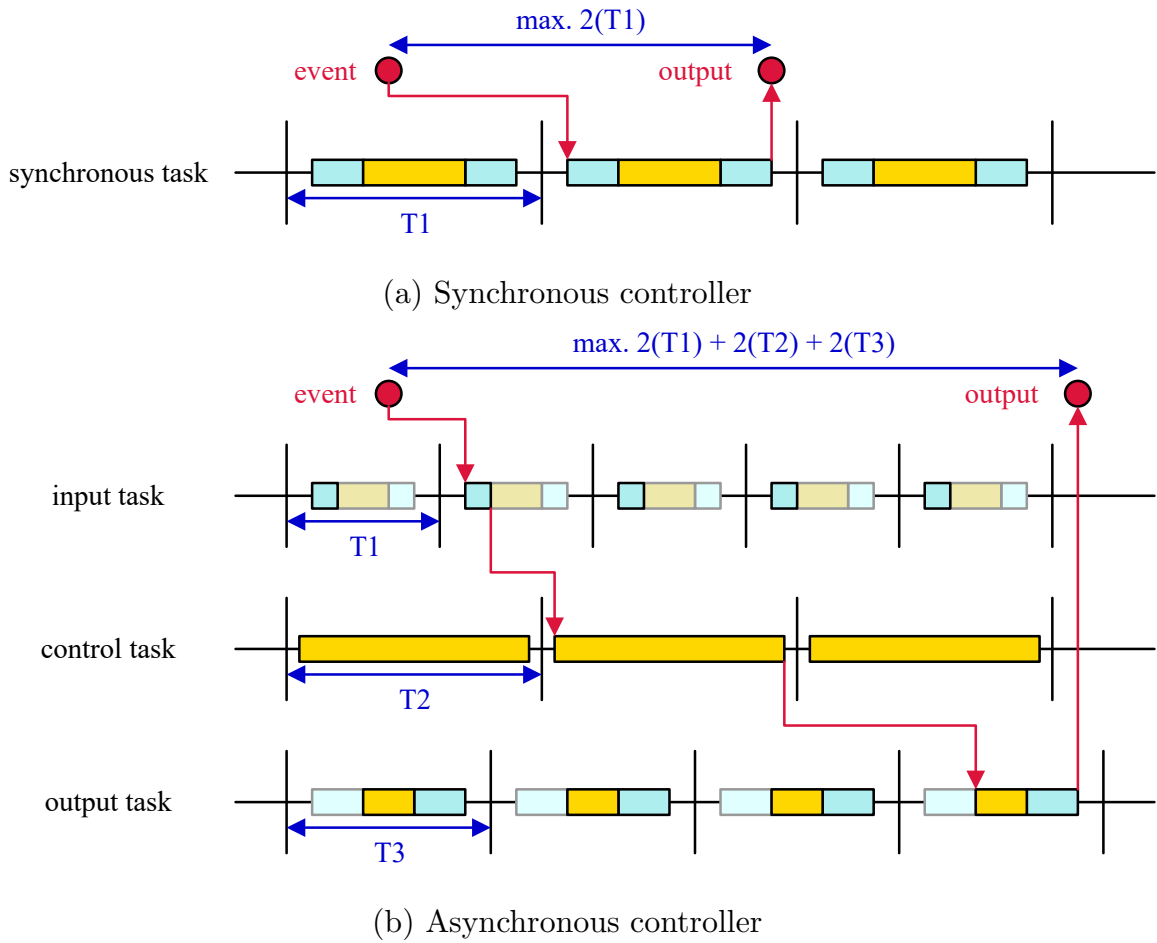


図 5.16 Software delay in periodic controller

EtherCAT を利用した同様のソフトウェアをあらためて開発する必要が生じた．その他にもハードウェア制御系における追加処理を実装するたびにソフトウェアの構造を作り直す必要が生じていた．

腱駆動型筋骨格ヒューマノイドに関する先行研究 [166] では，拡張性の高いマルチスレッド実行系ソフトウェアが実装されていた．図 5.17 に腱悟郎の実行系ソフトウェア構成を示す．

組込制御系は USB ネットワークで中央計算機に接続される．USB デバイスとのデータ交換には汎用 USB 通信ライブラリ libusb-1.0 を用いる．かつては独自の USB デバイスドライバが利用されていたが信頼性向上や保守開発の利便化のために libusb-1.0 に置き換えられた．それぞれの USB デバイスごとにセンサ値を受信するスレッドと指令値を送信するスレッドが生成される．その他に，上位のコントローラからのコマンド通信を処理するスレッドや制御データをプラグイン処理するスレッドが生成され，実行系ソフトウェアは多くのスレッドから構成される．一部のスレッド同士は排他処理実装やデバイスドラ

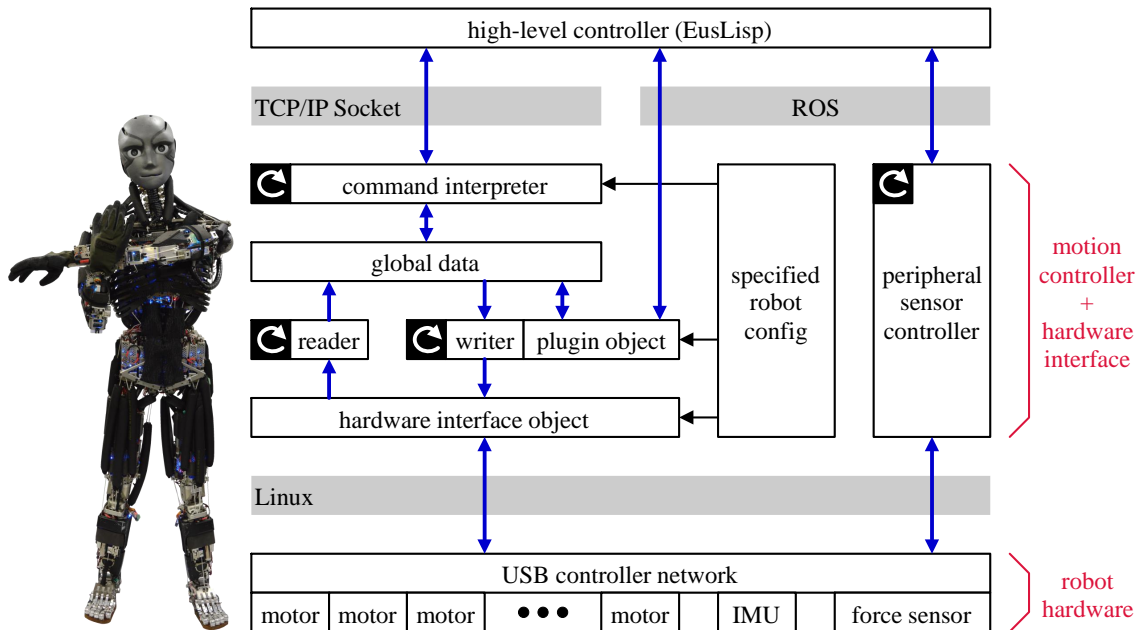


図 5.17 Motion controller software for the humanoid robot Kengoro in previous work

イバの共有・競合により同期関係にある．大別するとセンサ値受信・指令値送信・コマンド処理がそれぞれで同期している．プラグイン処理は指令値送信と同期している．上位のコントローラとは Socket API を利用した TCP/IP 通信で接続される．一部の通信に ROS を併用している．

フィードバック制御経路全体では非同期であり，実時間性も備えていない．実行系ソフトウェアでは制御周期を管理しておらず，それぞれの送受信スレッドにおける処理タイミングは USB 通信の実行に依存している．一方で，ロボットのハードウェア構成ごとに，通信インタフェース・プラグイン・設定コマンドなどを柔軟に拡張できる構造で実装されている．ハードウェアとのデータ交換が主な役割であるが，プラグイン機能によりフィードバック動作制御などの多機能実装に対応している．

5.5.3 一般化デバイス制御プラットフォーム

ハードウェアとのデータ交換において多くの場合で共通して見られる構造に基づき多様なロボットのハードウェア制御系の実装基盤を提供するソフトウェアとして，一般化デバイス制御プラットフォームを開発した．一般化デバイス制御プラットフォームを用いて，実行系ソフトウェアは図 5.18 のように構成される．従来のギア駆動型大出力ヒューマノイドにおける実時間ハードウェア制御系実装と，腱駆動型筋骨格ヒューマノイドにおける拡張性の高い実行系ソフトウェア実装の，それぞれの利点を取り入れている．

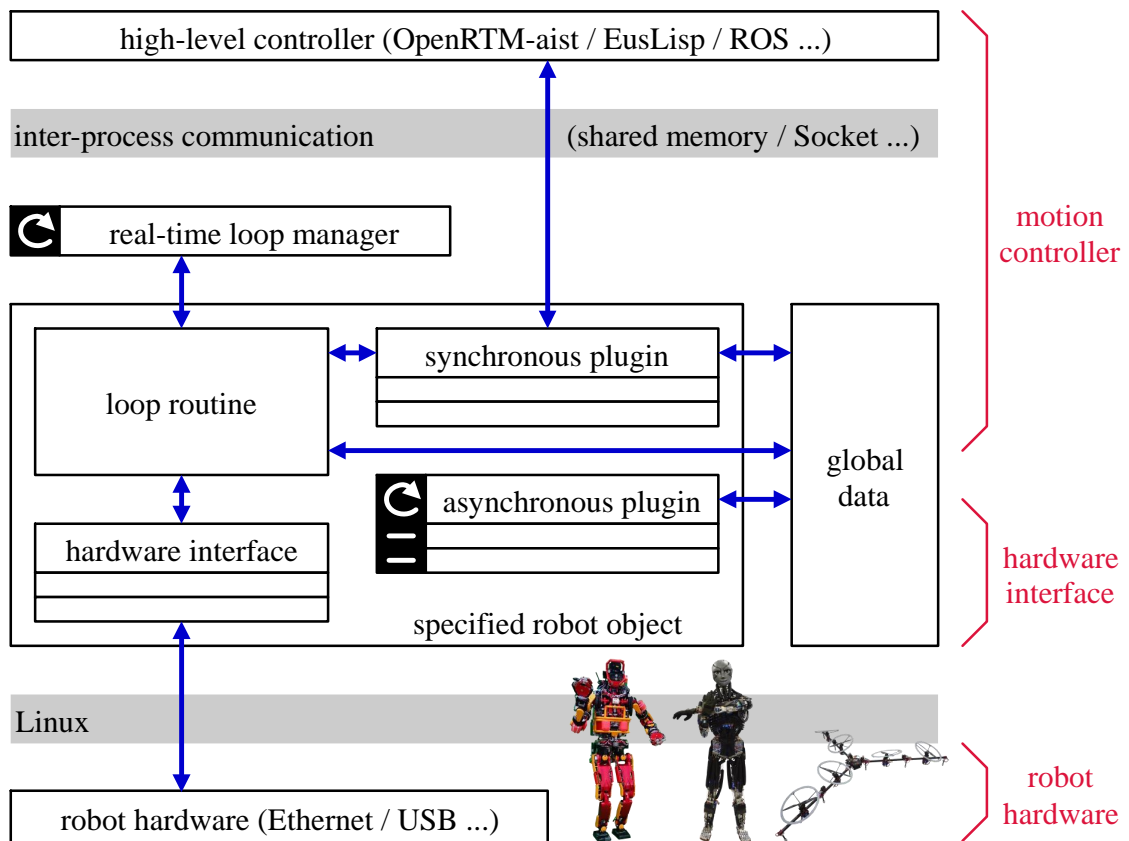


図 5.18 Motion controller software for various robots

一般化デバイス制御プラットフォームは、主に Linux を対象とした C++ によるオブジェクト指向ソフトウェアとして実装する。データパケットモデルに基づくデバイスデータ交換や、周期的なデータ処理の振る舞いを、オブジェクト指向で共通インターフェース化する。ポリモーフィックに継承されるオブジェクト構造によりロボットごとの具体的な実装に対応する。

ハードウェアからデータを収集し、これをフィードバックして反映した指令データを送信するまでのソフトウェア遅延はできる限り低減するのが望ましい。特に低遅延が求められるフィードバック制御に関しては、ハードウェア制御系の周期処理構造の中でデータ交換と同期して実行することが実時間性の達成に効果的である。一般化デバイス制御プラットフォームでは、これをプラグイン機能により実現する。

ソフトウェアの基本的なデータ処理フローを図 5.19 に示す。実時間な周期処理と非同期処理を組み合わせ、ハードウェアおよび外部ソフトウェアのそれぞれとデータを交換し、データの変換や制御処理を行う。

ロボットおよびソフトウェアの状態をグローバルデータとして保持し、プロセス内で共

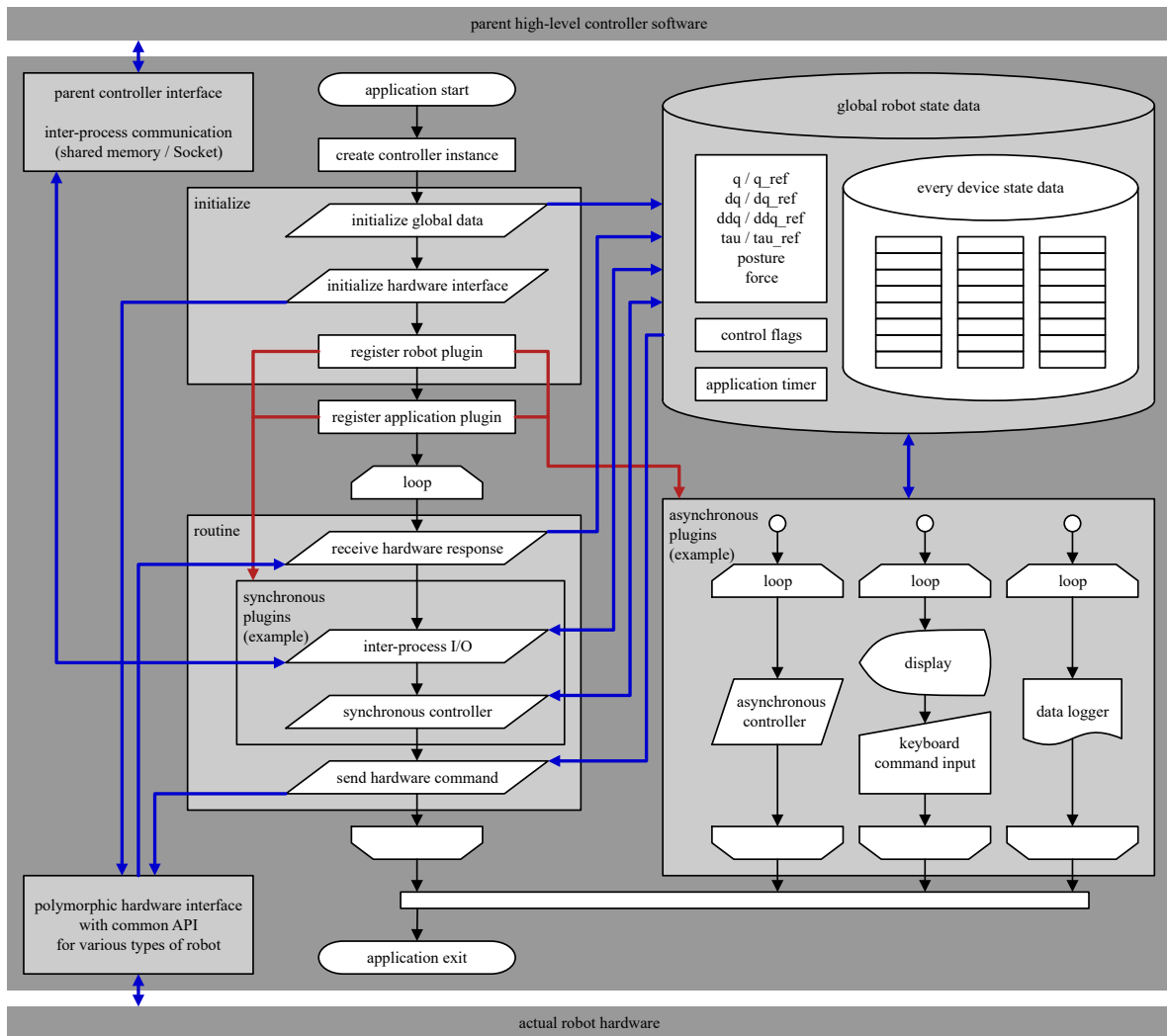


図 5.19 Application flow and data connection in the polymorphic real-time hardware controller platform

有する。ロボットの状態データとして、関節の位置や力、基準リンク姿勢、エンドエフェクタ力などの抽象化されたロボットモデルパラメータに加えて、デバイスごとに取り扱う詳細なデータ項目をオブジェクト化して保持する。周期処理の中でデバイスデータへのアクセスは頻繁に発生する。データの格納方法は周期処理の実行時間パフォーマンスに大きな影響を与える。多項目のデータを格納するデータ構造として、C++ の標準的なコンテナを用いて、

- array : 固定長配列
- vector : 可変長配列
- list : 双方向リスト

- map : 連想配列

といった実装方法が候補に挙がる．一般化デバイス制御プラットフォームの目指すところは，汎用的なロボット動作制御ソフトウェアではなく，応答性を重視したハードウェア制御系の機能を基本とする拡張実装基盤である．そのため，ハードウェアデバイスのデータ項目およびそれを格納するデータベース構造に過度な一般化は施さず，ハードウェアと交換するビット配列をそのまま当該データスロットに高速に格納できる整数型の固定長配列を用いる．

アプリケーションでは，ロボットごとに実装される初期化处理と周期処理を通して，グローバルデータの操作やハードウェアとのデータパケット交換を行う．同期・非同期・臨時のプラグインによって，グローバルデータに対して任意の操作を施し，あるいは外部ソフトウェアとのプロセス間通信インタフェースやユーザインタフェースを提供するなど，多様な処理実装への拡張を可能としている．一例として `ncurses` ライブラリを用いて CUI 上に展開したユーザインタフェースプラグインの実行画面を図 5.20 に示す．ユーザインタフェースは非同期プラグインとして実装され，周期的な画面表示更新や入力コマンド処理を行う．

ハードウェアとのデータ交換機能はデータパケットモデルに基づく共通 API で定義し，ハードウェアごとに個別の内部実装を与える．本研究で開発している Ethernet・USB・PCI Express・オンチップバスによる実装をはじめ，多様なインタフェースのロボットを統一的に扱うことが可能となる．複数のインタフェースで分散制御ノードを接続する場合や追加のセンサを搭載する場合には，複数のハードウェアインタフェースを組み合わせることで，同期処理による低遅延な全身フィードバック制御が容易に実現される．

データパケットモデルでは，種類やアドレスの情報を含む単一データの packets を基本要素としたデータ交換を定義する．packet を表すデータオブジェクトは，

- type : packet の処理方法を決定するための種類情報
- address : デバイスにおけるデータのローカルアドレス
- node : デバイスのノードアドレス
- channel : 対象デバイスを含むサブネットワークの番号
- size : データ長
- data : データ

の情報を保持する．中枢全身制御系からデバイス宛にコマンド packet 配列を送信し，デバイスからレスポンス packet 配列が返る，という構造がデータ交換の基本となる．周期処理の中で必要な全てのデータを低遅延に交換完了するための適切な手順は個別の通信プロトコルごとに存在する．複数のコマンド packet をセットして 1 度に送信する API を

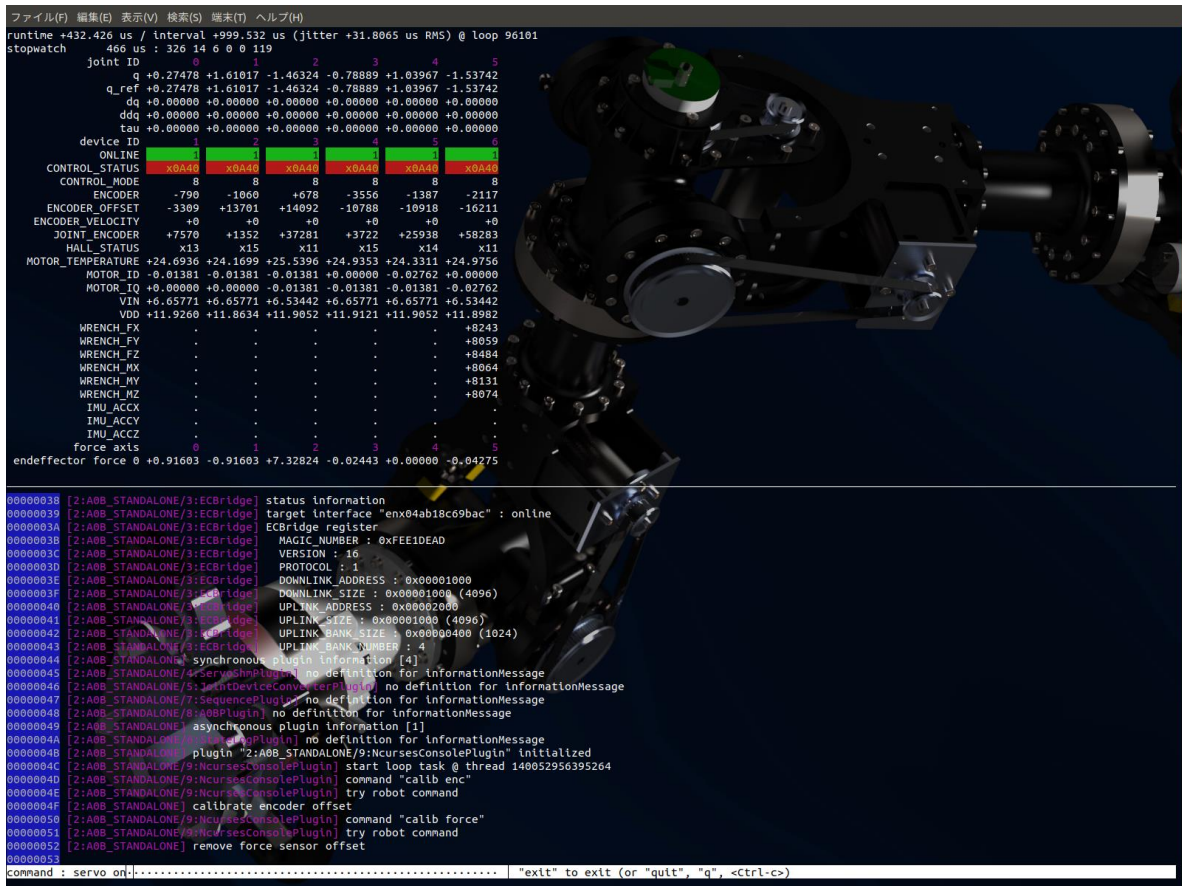


図 5.20 Example application screen of the polymorphic real-time hardware controller platform

備えることで、多くのハードウェア構成において低遅延データ交換実装に対応することができる。

これらの機能を実現する一般化デバイス制御プラットフォームは図 5.21 のオブジェクトデータ構造で実装される。プロセス内のグローバルデータを RobotState クラスのシングルトンインスタンスに格納する。AbstractRobot クラスでそれぞれの具体的な処理を実装する。AbstractRobot クラスでは、ハードウェアとのデータ交換を担う AbstractHardwareInterface クラス、および拡張プラグイン処理を担う AbstractPlugin クラスのインスタンスを必要な数だけ生成して取り扱う。その他に、マルチスレッドのメッセージ出力を管理し画面表示プラグインなどに渡す機能を担う、MessageStream クラスのインスタンスをグローバルに配置する。図 5.21 で水色に塗ったオブジェクトは、ポリモーフィズムに基づいて個別のロボットに向けた派生クラスで実装する。

典型的なアプリケーションでは AbstractRobot クラスの API を操作することで周期的な実時間制御を実行する。プラグインによりプロセス間通信インタフェースを実装し、上

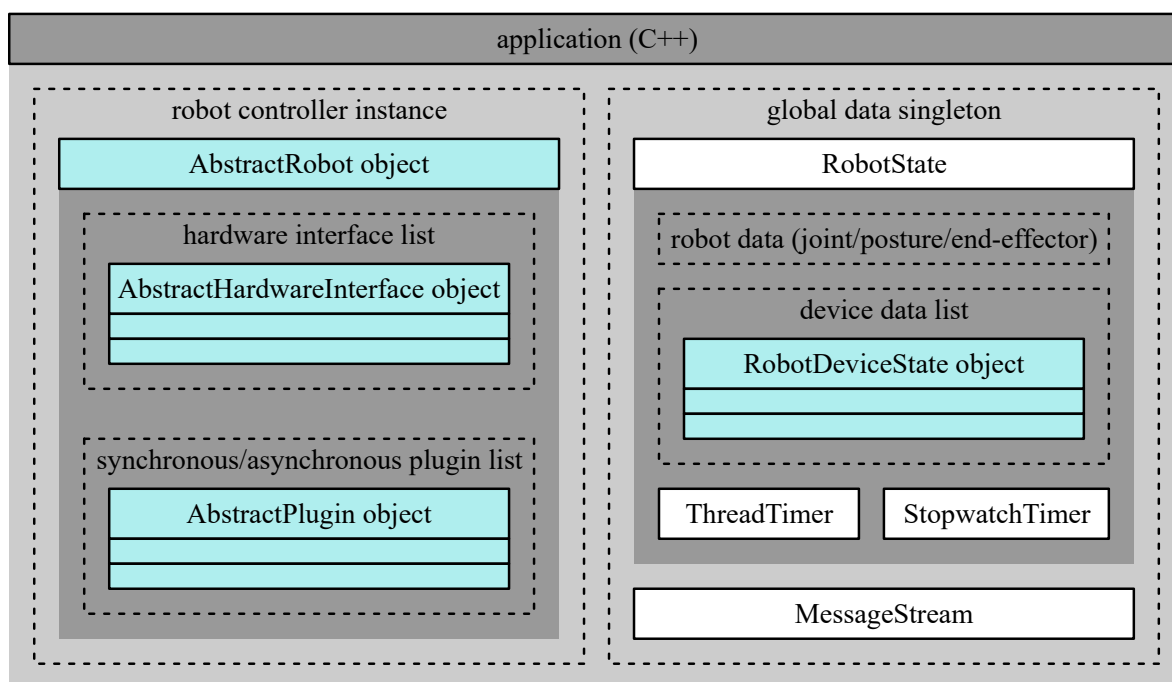


図 5.21 Data structure in the polymorphic real-time hardware controller platform

位のコントローラと接続して利用する。しかし，上位のコントローラまで含めたフィードバック制御系の遅延が問題となる場合もある。上位のコントローラにおける同期実行に対応する拡張性が効果を発揮する。一般化デバイス制御プラットフォームではオブジェクト構造のソフトウェア設計により，一部のオブジェクトのみを取り出してコンポーネントとして再利用することが可能となっている。AbstractRobot クラスを取り出すと，ロボットのデータ転送・変換機能の全体を任意のプロセス内で利用できる。共通 API で個別のハードウェアとの低遅延な通信機能を実装する AbstractHardwareInterface クラスは，他のデータ構造に依存せず単体で利用できるように設計されており，コンパクトな移植も可能である。

5.6 組込中間層計算機を活用した低遅延全身動作制御

5.6.1 ヘテロジニアス全身動作制御系アーキテクチャ

中央計算機に実行系ソフトウェアを実装する典型的な構成に加えて組込中間層計算機を活用することで，ヘテロジニアス計算機環境による多階層な全身動作制御系を構成する。中央計算機と組込中間層計算機によるヘテロジニアス計算機アーキテクチャは図 4.13 に示した。それぞれの計算機と組込分散ネットワークの間でデータを交換するパケット切替中継機構は図 4.18 に示した。多階層なフィードバック経路は図 5.2 に示した。

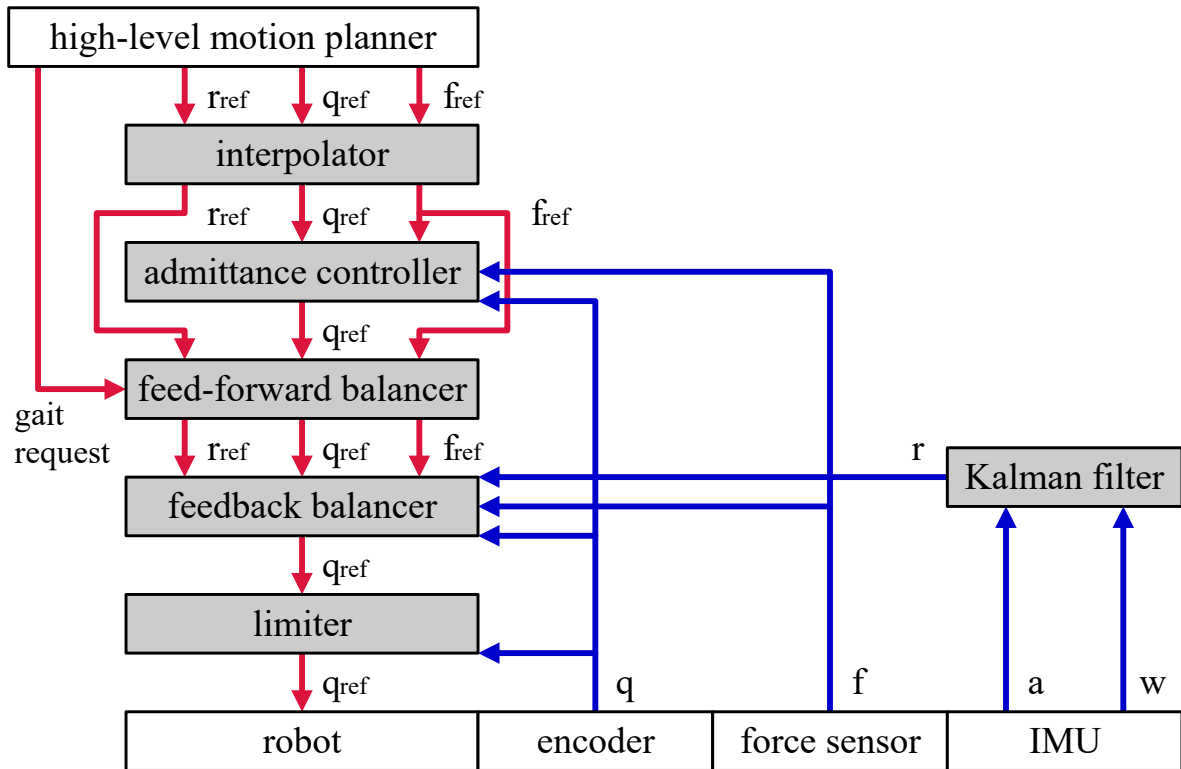
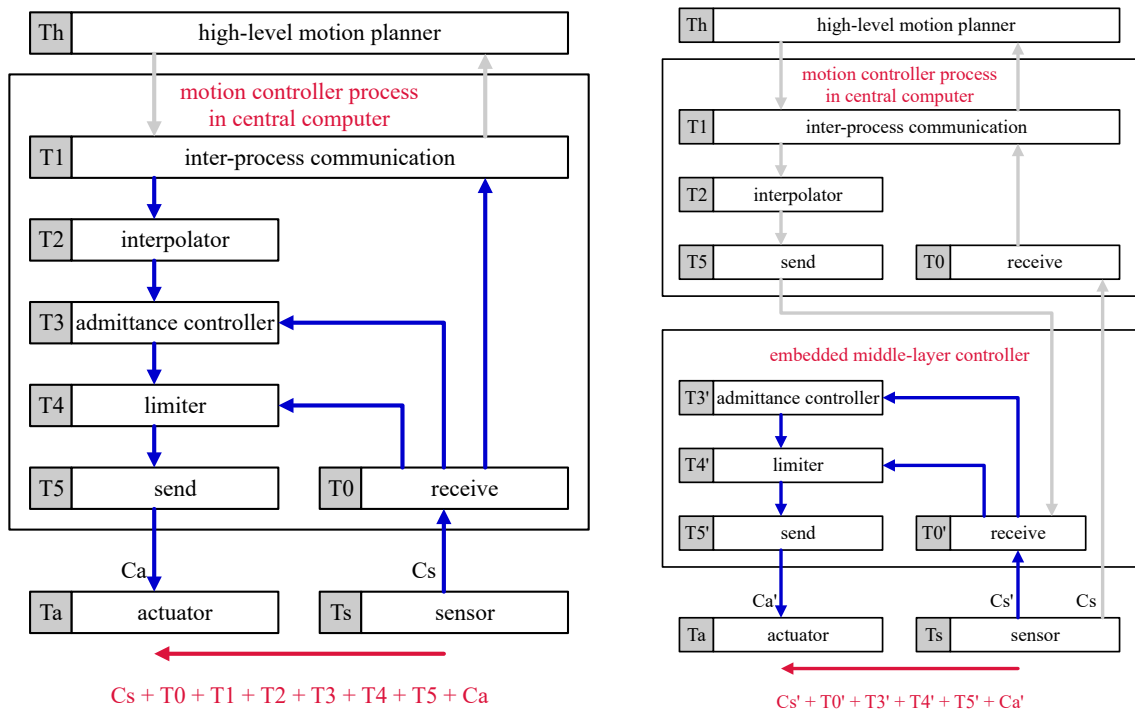


図 5.22 Motion controller system for the humanoid robot JAXON

ロボットの動作制御系では 5.2 節で述べたようにフィードフォワード・フィードバック・フェイルセーフの動作補正を実施することが求められる。図 5.15 で紹介した JAXON の動作制御系を実装する HRPSYS においては、上位の動作計画系から与えられた目標動作をおよそ図 5.22 のように補正してハードウェアに送信する [157]。 r は基準リンク姿勢， q は関節位置， f はエンドエフェクタ力， a は IMU で計測される加速度， w は IMU で計測される角速度を表している。

上位の動作計画系は実時間性や周期性を有するとは限らない。動作計画系からは時間分解能の粗い目標動作が与えられる。これを動作制御系の実行周期と整合するように補間して用いる。ロボット実機から得られる力などのセンサデータをフィードバックして、四肢のアドミタンス制御および下半身のバランス制御を実行する。生成された動作に対してリンクの干渉や過大な誤差を避けるような制限を施して、ロボットの実機に指令値として送信する。

目標動作の補間はフィードフォワード補正に分類される。アドミタンス制御はフィードバック補正に分類される。バランス制御はフィードフォワードなバランス補正とフィードバックによる姿勢安定化制御から構成される。動作指令制限は実機のセンサデータを利用するフェイルセーフ補正に分類される。



(a) Central computer

(b) Embedded middle-layer controller

図 5.23 Computational graph for heterogeneous admittance controller

動作制御系の一部を組込中間層計算機で実行することでフィードバック遅延が低減される可能性がある。通常，中央計算機は組込中間層計算機よりも高い演算性能を有するため，上位から与えられた目標動作をフィードフォワードに処理することで完結する部分は中央計算機で実行する。フィードバック制御は組込中間層計算機で実行した方が通信遅延を低減できる。全体のフィードバック遅延の優劣は計算時間と通信時間のトレードオフで決定される。

2.7.1 項で紹介したヘテロジニアス分散制御モデルに基づいて，図 5.22 のうちバランス制御を除いたアドミタンス制御の実装の例を用いて分析する。アドミタンス制御を中央計算機と組込中間層計算機のそれぞれで実行する場合の実行時間グラフモデルを図 5.23 に示す。

図 5.23 (a) はアドミタンス制御を含む動作制御系の全体を中央計算機で実行する場合である。センサ入力を始点，アクチュエータ出力を終点とする。直列実行される動作制御系タスク T_0 から T_5 における計算時間と入出力経路 C_s および C_a における通信時間の総和として実行時間が得られる。 C_s および C_a の通信時間は入出力のソフトウェア遅延 T_0 および T_5 に含めてモデル化しても良い。図 5.23 (b) はアドミタンス制御を組込中間層計算機で実行する場合である。センサ値および中央計算機からの指令値をネットワーク

経由で読み出し，アドミタンス制御や動作指令制限の計算を行い，ネットワーク経由でアクチュエータ出力を送信する．中央計算機で実行する場合と比べて，入出力の通信時間は小さく，制御計算時間は大きくなる．組込中間層計算機で FPGA や GPU によるハードウェアアクセラレーションを活用すると，制御計算時間を小さく抑えて優位性を向上できる可能性がある．

5.6.2 アドミタンス制御演算の実装

組込中間層計算機でロボットの全身力フィードバック動作制御を実行する．ここでは汎用性が高く代表的な制御構造を有する力フィードバック制御として，図 2.8 に示したアドミタンス制御の実装手法を考える．

実装するアドミタンス制御の計算構造を図 5.24 に示す．アドミタンス制御は，

1. 順運動学計算により前段の制御器から入力された基準関節位置をデカルト空間の基準エンドエフェクタ位置に変換
2. 順運動学計算により実際の関節位置と力センサ値をデカルト空間の運動と力に変換
3. 関節運動からエンドエフェクタ運動に対するヤコビ行列を計算
4. 設定した機械インピーダンスモデルを再現するエンドエフェクタ運動を生成
5. ヤコビ行列を利用した逆運動学計算によりエンドエフェクタ運動を関節運動に変換して出力

の流れで実行される．

順運動学問題は，ロボットの関節空間運動からデカルト空間運動を求める問題である．順運動学計算は多くのロボット動作制御計算の基礎となっている．それぞれの関節による座標変換の伝播を同次変換行列で表現する手法がよく知られている．

n 個の回転関節で構成される垂直多関節ロボットの場合，関節位置 $q = (q_0, q_1, \dots, q_{n-1})^T$ に対して，次の漸化式により各リンクのデカルト空間姿勢が求まる．

$$p_i = p_{i-1} + R_{i-1} b_i \quad (5.33)$$

$$R_i = R_{i-1} r_i \quad (5.34)$$

$$r_i = E + \hat{a}_i \sin q_i + \hat{a}_i^2 (1 - \cos q_i) \quad (5.35)$$

ここで，

- E : 単位行列 (3 次正方行列)
- p_i : デカルト空間でのリンク座標 (3 次ベクトル)
- R_i : デカルト空間でのリンク姿勢回転行列 (3 次正方行列)

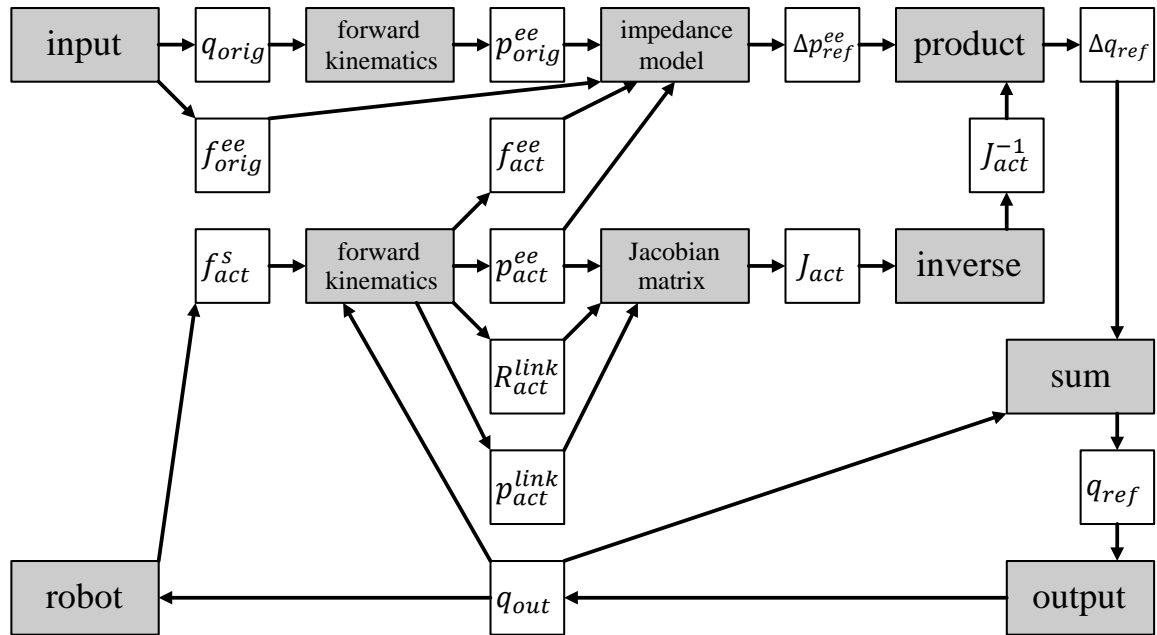


図 5.24 Computation and input/output structure of admittance control

- b_i : 親リンクからの相対リンク座標を表すパラメータ (3 次ベクトル)
- r_i : 親リンクからの相対回転行列 (3 次正方行列)
- a_i : ローカル回転軸を表すパラメータ (3 次ベクトル)

である． \hat{a}_i は a_i による外積を表現する 3 次正方行列である．

ローカル回転軸を基準座標軸に限る場合は回転行列の計算を簡略化できる．x 軸周りの回転，すなわち $a_i = (1, 0, 0)^T$ の場合は，

$$r_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos q_i & -\sin q_i \\ 0 & \sin q_i & \cos q_i \end{pmatrix} \quad (5.36)$$

となる．同様に y 軸周りの回転，すなわち $a_i = (0, 1, 0)^T$ の場合は，

$$r_i = \begin{pmatrix} \cos q_i & 0 & \sin q_i \\ 0 & 1 & 0 \\ -\sin q_i & 0 & \cos q_i \end{pmatrix} \quad (5.37)$$

z 軸周りの回転，すなわち $a_i = (0, 0, 1)^T$ の場合は，

$$r_i = \begin{pmatrix} \cos q_i & -\sin q_i & 0 \\ \sin q_i & \cos q_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.38)$$

となる．

エンドエフェクタの運動に関するヤコビ行列 J は関節速度 \dot{q} からデカルト空間におけるエンドエフェクタ速度 v_{ee} への写像を,

$$v_{ee} = J\dot{q} \quad (5.39)$$

として表す行列である．関節 i の速度 \dot{q}_i からエンドエフェクタ速度への写像に対応するヤコビ行列の i 列部分 J_i は, 順運動学計算結果に基づいて,

$$J_i = \begin{pmatrix} \omega_i \times (p_{ee} - p_i) \\ \omega_i \end{pmatrix} \quad (5.40)$$

$$\omega_i = R_i a_i \quad (5.41)$$

$$p_{ee} = p_{n-1} + R_{n-1} b_{ee} \quad (5.42)$$

と計算できる． ω_i はデカルト空間での関節 i の回転軸である．

ヤコビ行列による写像の逆変換を行うことでエンドエフェクタの運動から関節空間の運動へ変換することができる．6 関節のロボットでヤコビ行列が正則となる場合は, ヤコビ行列の逆行列をエンドエフェクタ速度にかけることで,

$$\dot{q} = J^{-1}v_{ee} \quad (5.43)$$

として関節速度が求まる．計算機上ではヤコビ行列の LUP 分解に基づく Gauss の消去法アルゴリズムを用いることで高速に計算することができる．一般の関節数の場合は Moore-Penrose 疑似逆行列を用いることで最小二乗・最小ノルム解が得られる．実際の制御に利用するにあたり, 特異点が近く可操作度が小さい場合に関節速度を過大としないために関節速度を評価項に加えた SR-inverse [167][168] を利用することもできる．

位置の次元の場合でも微小変位の仮定のもとに同様のヤコビ行列写像の逆変換によってエンドエフェクタの変位から関節空間の変位へ近似的に変換することができる．アドミタンス制御では, 機械インピーダンスモデルに基づいて生成された 1 周期分のエンドエフェクタ変位からヤコビ行列写像の逆変換によって関節変位を求め, これを現在の関節位置に加算することでエンドエフェクタに所望の運動を再現する．

機械インピーダンスモデルでは図 5.25 のようにエンドエフェクタに仮定の弾性・粘性・慣性による力学特性を設定する．力と運動の関係は,

$$f = M\ddot{x} + D\dot{x} + Kx \quad (5.44)$$

と表される． f は 6 自由度の力・モーメントであり, 実際の制御ではエンドエフェクタが本来発揮すべき基準力を設定してこれに対する差分を用いる． x は 6 自由度の位置・姿勢であり, 力 0 での釣り合い位置となる基準位置を設定してこれに対する相対的な変位を用いる． $M \cdot D \cdot K$ は自由度ごとのパラメータを表す対角行列である．6 自由度の力・

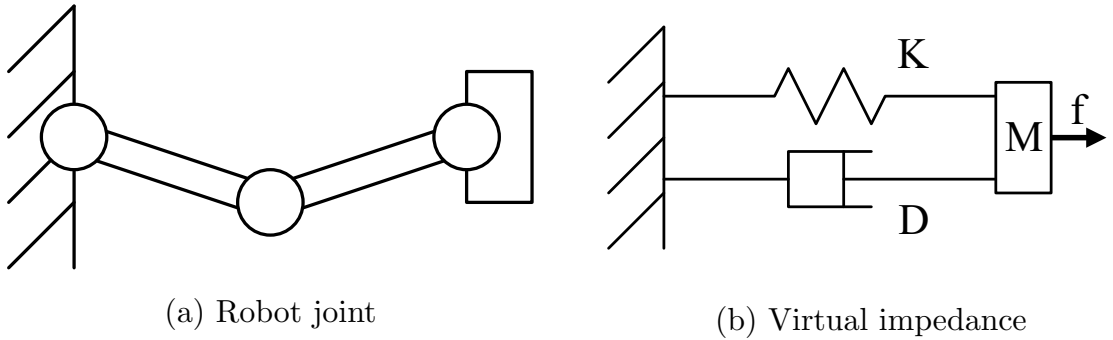


図 5.25 Virtual mechanical impedance model for robot

モーメントに対する 6 自由度の位置・姿勢の運動に対して，そのうち 1 方向のみ剛性を低くするなど，アプリケーションに応じて自由度ごとに粘弾性を設定することができる．

制御周期ごとの位置の変位は離散モデルにより導出される．1 自由度の運動方程式，

$$f = M\ddot{x} + D\dot{x} + Kx \quad (5.45)$$

を後退差分に基づいて離散化すると，

$$f = M \frac{x_i - 2x_{i-1} + x_{i-2}}{\Delta t^2} + D \frac{x_i - x_{i-1}}{\Delta t} + Kx_i \quad (5.46)$$

$$f\Delta t^2 = (M + D\Delta t + K\Delta t^2)x_i - (2M + D\Delta t)x_{i-1} + Mx_{i-2} \quad (5.47)$$

$$x_i = \frac{f\Delta t^2 + (2M + D\Delta t)x_{i-1} - Mx_{i-2}}{M + D\Delta t + K\Delta t^2} \quad (5.48)$$

と変形できる． Δt は離散化の周期時間， i は周期を表すインデックスである．これをもとに 1 周期分の変位 Δx が，

$$\begin{aligned} \Delta x &= x_i - x_{i-1} \quad (5.49) \\ &= \frac{f\Delta t^2 + (M - K\Delta t^2)x_{i-1} - Mx_{i-2}}{M + D\Delta t + K\Delta t^2} \end{aligned}$$

として得られる．

あるいは質量を無視して粘弾性のみで考えると，

$$f = D\dot{x} + Kx \quad (5.50)$$

より，

$$\begin{aligned} \Delta x &= \dot{x}\Delta t \quad (5.51) \\ &= \frac{f - Kx}{D}\Delta t \end{aligned}$$

と計算を単純化できる．式 5.51 では粘性を小さく設定することができないため，後退差分に基づく式 5.49 から $M = 0$ として変形して，

$$\Delta x = \frac{f - Kx}{D + K\Delta t} \Delta t \quad (5.52)$$

とする考え方もできる．式 5.51 と比べて，前進差分・後退差分の離散微分モデルの違いにより分母に $K\Delta t$ の項が現れて $D \geq 0$ で成立するようになるが，実装上は $K\Delta t$ の分だけ D を大きく設定するのと等価である．

以上のモデルに基づき，図 5.24 のアドミタンス制御を実装して評価する．機械インピーダンスモデルとして式 5.51 の粘弾性モデルを用いる．6 軸マニピュレータのアドミタンス制御を例に取り，これを高性能計算機製品および組込中間層計算機の FPGA SoC に実装する．高性能計算機として Intel 社の NUC 10 Performance Kit NUC10i7FNH を利用する．1.1 GHz の 6 コア 12 スレッド CPU，Core i7-10710U を搭載している．

高性能計算機および FPGA SoC のそれぞれで 6 軸マニピュレータのアドミタンス制御を含む一連の周期制御タスクを実行し，その実行時間を図 5.26 に示す．FPGA SoC の CPU では Ethernet で接続される高性能計算機と比べて FPGA とのデータ交換を短時間で完了する一方で制御計算に時間を要する．タスク実行時間全体では，高性能計算機では $425 \mu\text{s}$ を要するのに対し，FPGA SoC では $188 \mu\text{s}$ に収まった．組込中間層計算機ではおよそ 5 kHz の高速なアドミタンス制御も可能となる．

6 軸マニピュレータの一連のアドミタンス制御計算を高性能計算機上で実行するのに必要な時間は $8 \mu\text{s}$ であった．一方，FPGA SoC の CPU で実行すると $88 \mu\text{s}$ となった．高性能計算機と比べて 11 倍の時間となっている．FPGA SoC の CPU で実行した場合の計算時間の内訳を図 5.27 に示す．順運動学および逆運動学の行列計算に大きな時間を要していることが確認される．

ヤコビ行列の計算を含む順運動学計算は式 5.33 や式 5.34 を基本とした関節ごとの逐次反復計算により実装される．図 5.28 に順運動学計算のデータ処理構造を示す．3 次正方行列と 3 次ベクトルの積算，および 3 次ベクトル同士の加算を多用する構造となっている．行列やベクトルの積算・加算は要素ごとに並列に実行できる．このように並列化が可能な同一構造の計算の反復には FPGA を用いたハードウェアアクセラレーションが有効である．

CPU によるアドミタンス制御計算の一部を FPGA で高速化することを目指し，図 5.28 に枠線で示した順運動学の逐次反復計算部分を FPGA に実装する．図 5.29 に順運動学計算アクセラレータ回路の構造を示す．

並列計算回路として，3 次正方行列と 3 次ベクトルの積算回路，および 3 次ベクトル同士の加算回路を構成する．IEEE 754 単精度 32 bit 浮動小数点の積算回路と加算回路を基本要素としてこれを複数並列配置する．組込中間層計算機モジュールに搭載した Cyclone

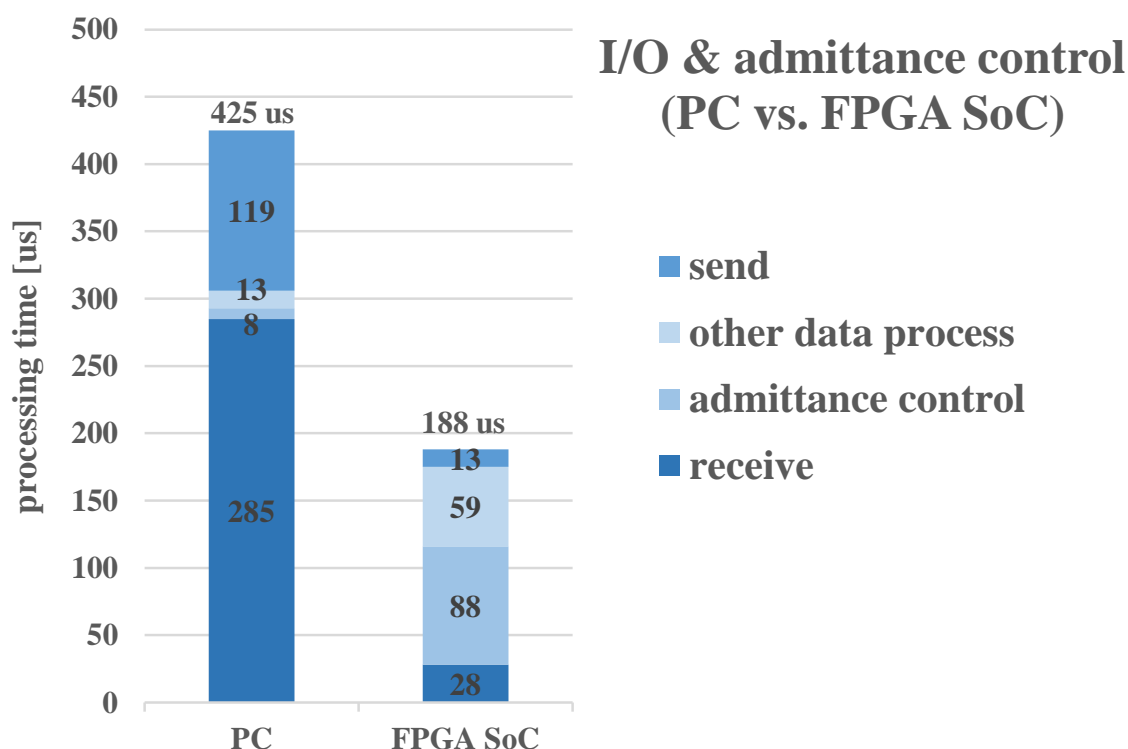


図 5.26 Periodic processing time with admittance control for 6-axis manipulator

V SX FPGA SoC では、整数積算に対応した DSP ブロックが利用できる。DSP ブロックとプログラマブルロジック回路を組み合わせることで浮動小数点演算回路が実装可能となっている。浮動小数点積算回路では仮数部の積算に DSP ブロックを 1 個利用する。浮動小数点積算回路および浮動小数点加算回路は 50MHz の動作周波数に対してそれぞれ 2 クロックの入出力遅延を有する構成となる。

図 5.30 に 3 次正方行列と 3 次ベクトルの積算回路の構成を示す。9 個の浮動小数点積算回路と 6 個の浮動小数点加算回路から構成される。3 段の接続となり、1 段あたり 2 クロックの遅延を有するため計 6 クロックの遅延となる。内部遅延を調整してパイプライン化された処理により、連続実行可能な N セットの積算は $N + 5$ クロックで完了する。

図 5.31 に 3 次ベクトル同士の加算回路の構成を示す。3 個の浮動小数点加算回路が並列配置された構造で 2 クロックの遅延となる。

行列・ベクトルの並列計算回路を利用して順運動学計算を実行するステートマシンを構成する。ステートマシンは 1 セットの計算ごとに上位のコントローラからリセットされた後、関節ごとに次の手順で状態を更新する。

1. 上位のコントローラから関節位置の余弦値 $\cos q_i$ を入力

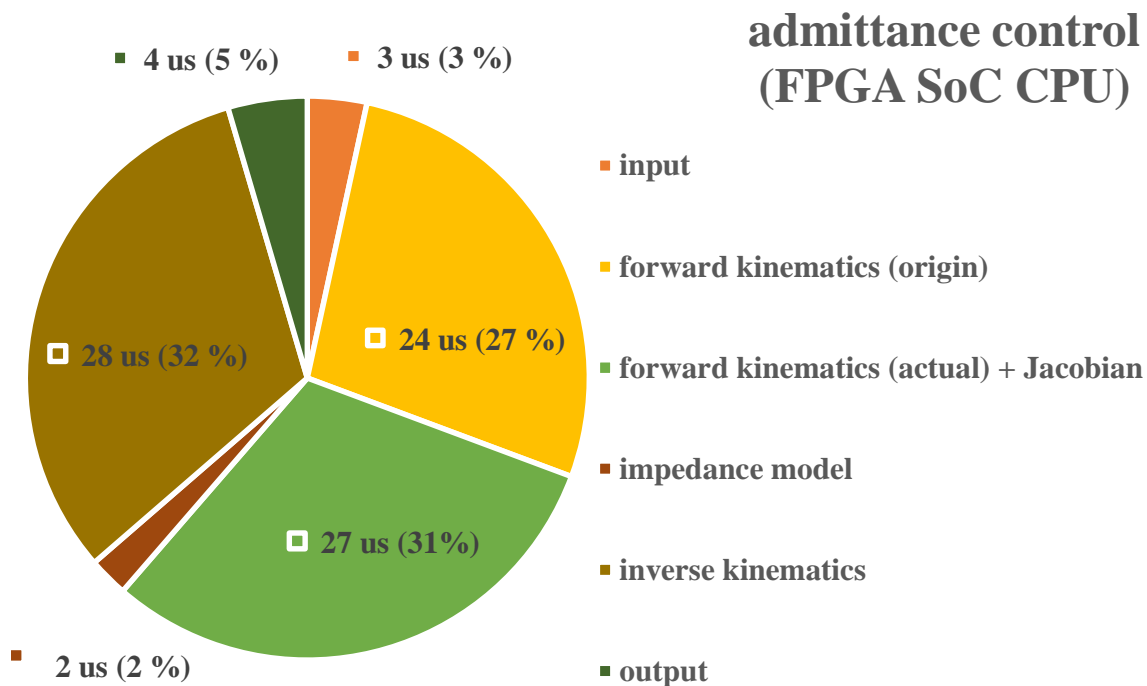


図 5.27 Breakdown of admittance control computation time for 6-axis manipulator

2. 上位のコントローラから関節位置の正弦値 $\sin q_i$ を入力
3. 関節回転軸 a_i をもとに $\cos q_i$ および $\sin q_i$ からローカル回転行列 r_i を生成
4. 行列・ベクトル積算回路に r_i の列を順次入力
5. 行列・ベクトル積算回路の出力からデカルト空間回転行列 R_i を取得して出力
6. R_i および a_i からデカルト空間回転軸 ω_i を出力
7. 全関節の計算が完了していれば停止し計算が残っている場合は関節インデックスを次に進めて処理続行
8. 行列・ベクトル積算回路にローカルリンク位置 b_{i+1} を入力
9. ベクトル・ベクトル加算回路に直前のデカルト空間リンク位置 p_i を入力
10. ベクトル・ベクトル加算回路の出力からデカルト空間リンク位置 p_{i+1} を取得して出力
11. 初めの余弦値入力に戻る

処理を単純化するため関節回転軸 a_i は x/y/z の 3 軸に限定している．これによりローカル回転行列 r_i およびデカルト空間回転軸 ω_i を求める工程での浮動小数点演算が不要となる．

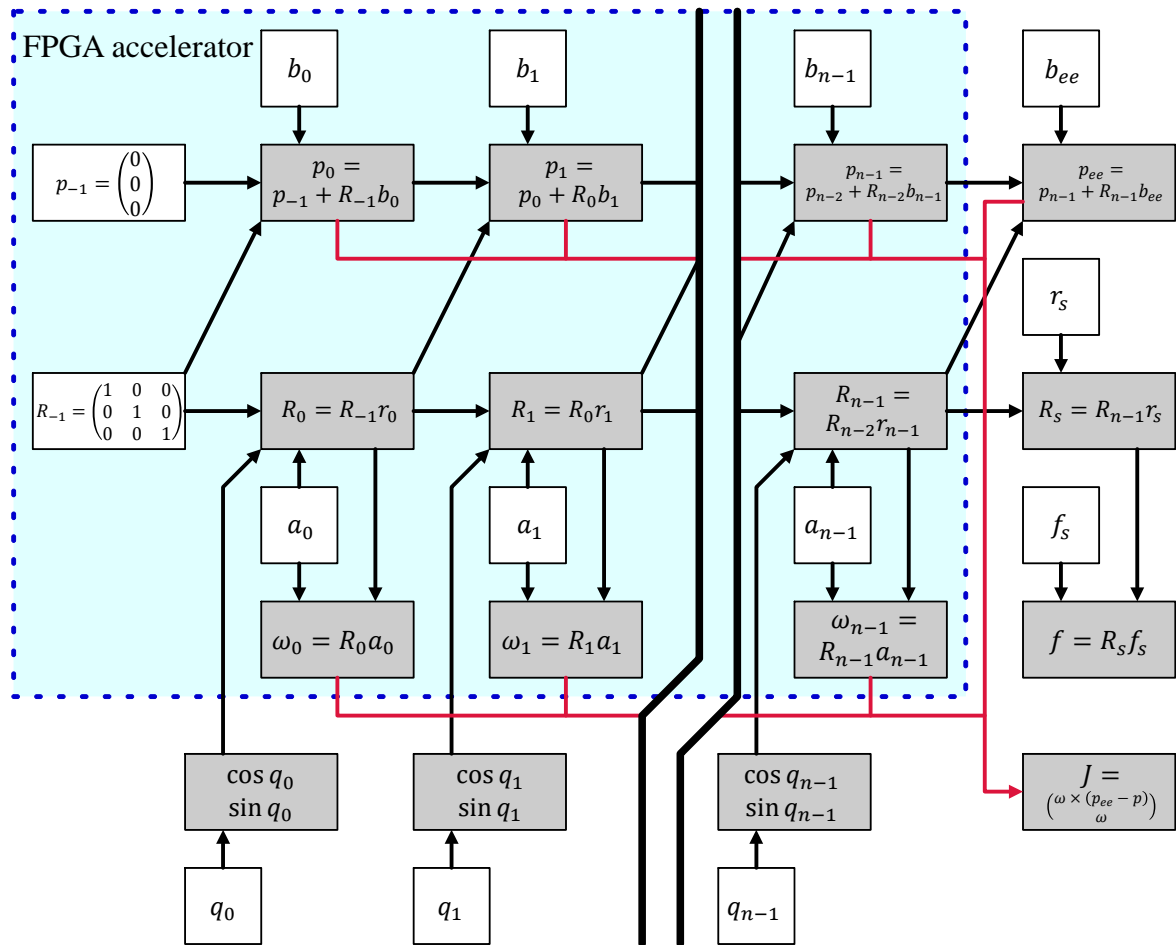


図 5.28 Computation structure of FPGA forward kinematics accelerator in admittance control

上位のコントローラから関節位置の余弦値 $\cos q_i$ と正弦値 $\sin q_i$ が最速タイミングで入力された場合，ステートマシンでは 1 関節の計算を 22 クロックで完了する．6 関節の順運動学計算は最短 132 クロック，50 MHz のため $2.64 \mu s$ で完了する．これは高性能計算機で実行した場合とおよそ同程度の計算時間である．組込制御系においても FPGA によるハードウェアアクセラレーションを利用することで高性能計算機に匹敵する演算性能が得られる．

CPU と FPGA の間のデータ交換に要する時間を含めたアドミタンス制御全体の計算時間は図 5.32 のようになる．順運動学計算の実行時間がおよそ 3 分の 2 に削減されている．

ここでは CPU で実行するアドミタンス制御のうち一部の順運動学計算を FPGA で実行しており，CPU と FPGA の間で頻繁にデータ交換も発生するため高速化の程度は限られている．他の工程も同様に FPGA で実行することで更なる高速化が見込まれる．順

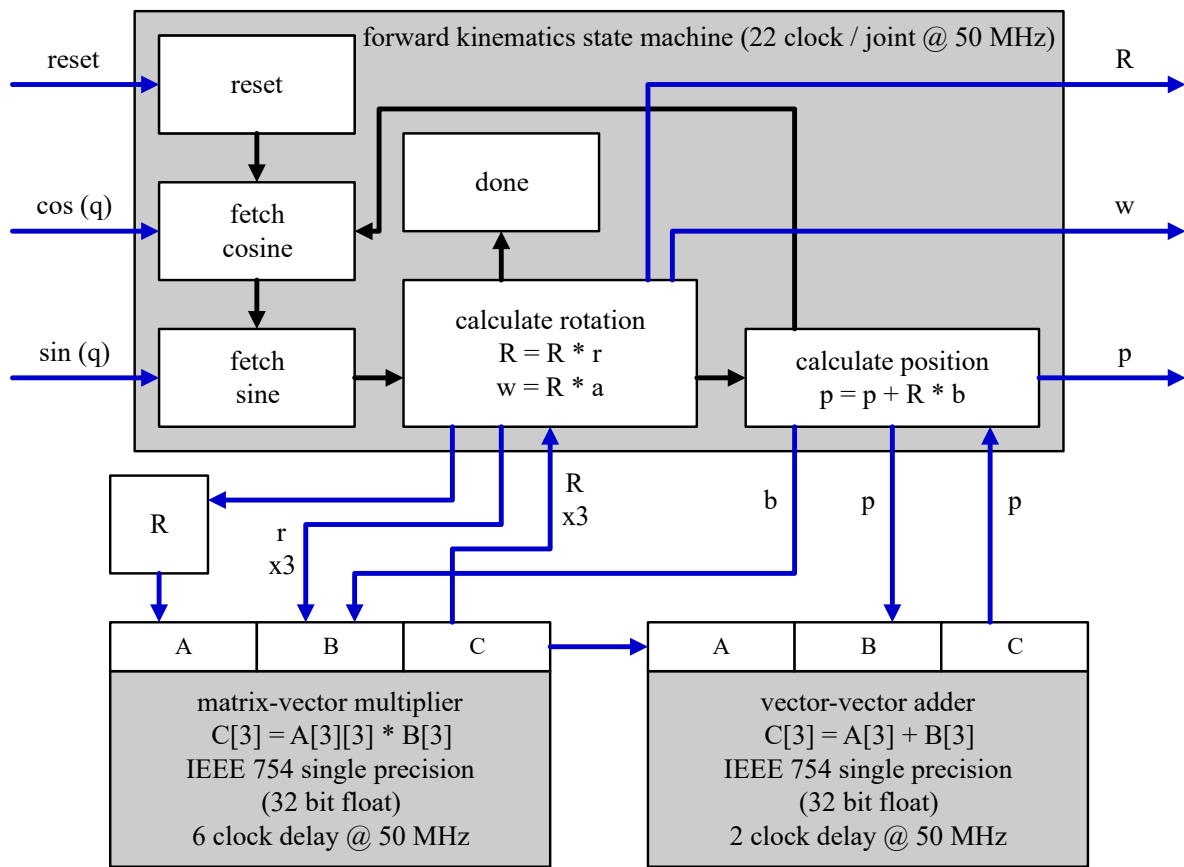


図 5.29 FPGA forward kinematics accelerator circuit

運動学計算の入力には三角関数の計算が求められる．FPGA において CORDIC アルゴリズムを利用して三角関数の計算を含む順運動学計算を高速に実行する手法もいくつか提案されている [169][170][171]．ヤコビ行列による写像の逆変換に基づく逆運動学計算を FPGA で高速に実行する手法も提案されている [116][172]．FPGA によるそれぞれの計算のアクセラレータを組み合わせると，CPU データ転送などによるオーバーヘッドの少ない高速演算が可能となる．

表 5.2 に順運動学計算アクセラレータ回路の FPGA リソース使用量を示す．順運動学計算アクセラレータ回路は FPGA で利用可能なプログラマブルロジック回路のうち 9.2% を利用している．同様のアクセラレータ回路を複数追加するだけの容量が残されている．必要なアプリケーションに対して容量が不足する場合は組込中間層計算機モジュールの FPGA 基板を多段接続することで FPGA アクセラレータのネットワークを拡張することが可能である．

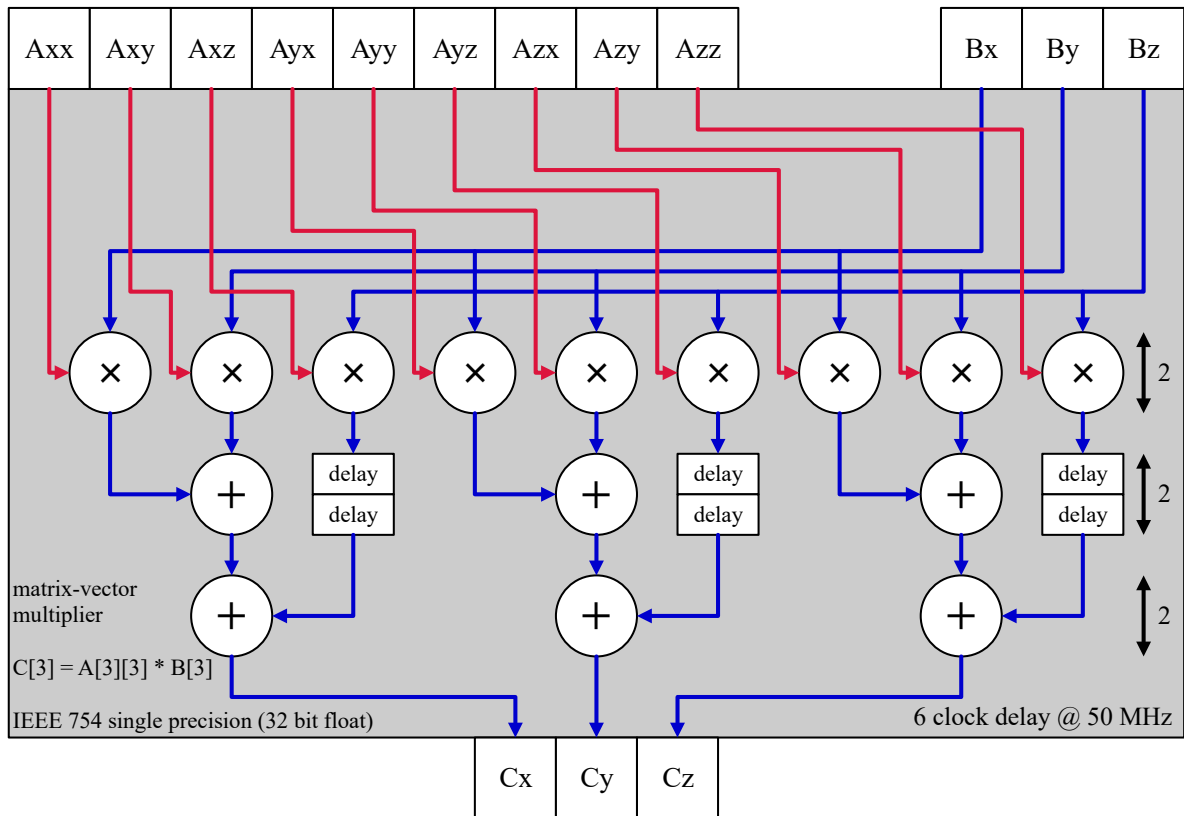


図 5.30 Matrix-vector multiplier circuit

5.7 本章のまとめ

本章では，組込計算機による高速な動作制御計算の実装手法と，体内分散制御システム全体での低遅延なフィードバック制御系の構成手法に関して述べた．

多階層から構成される多自由度ロボットの体内分散制御系におけるフィードバック動作制御の構造を分析した．主な外乱応答を担うフィードバック制御の低遅延化に向けて，アクチュエータごとにフィードバックする方法と中枢全身制御系でのフィードバック遅延を低減する方法のそれぞれを検討した．

モータ制御モジュールに実装されるサーボ制御システムを紹介し，FPGA を活用することで低遅延なローカルフィードバック制御が達成されることを述べた．ギア駆動型ロボットのモータ制御モジュールごとに実行可能な慣性・摩擦補償による推定トルクフィードバック制御を FPGA 上に実装し，その効果を確認した．

中枢全身制御系の計算機におけるハードウェア制御系にみられる共通構造を分析した．非同期マルチプロセス構造の動作制御系ソフトウェアにおける課題に関して述べた．これ

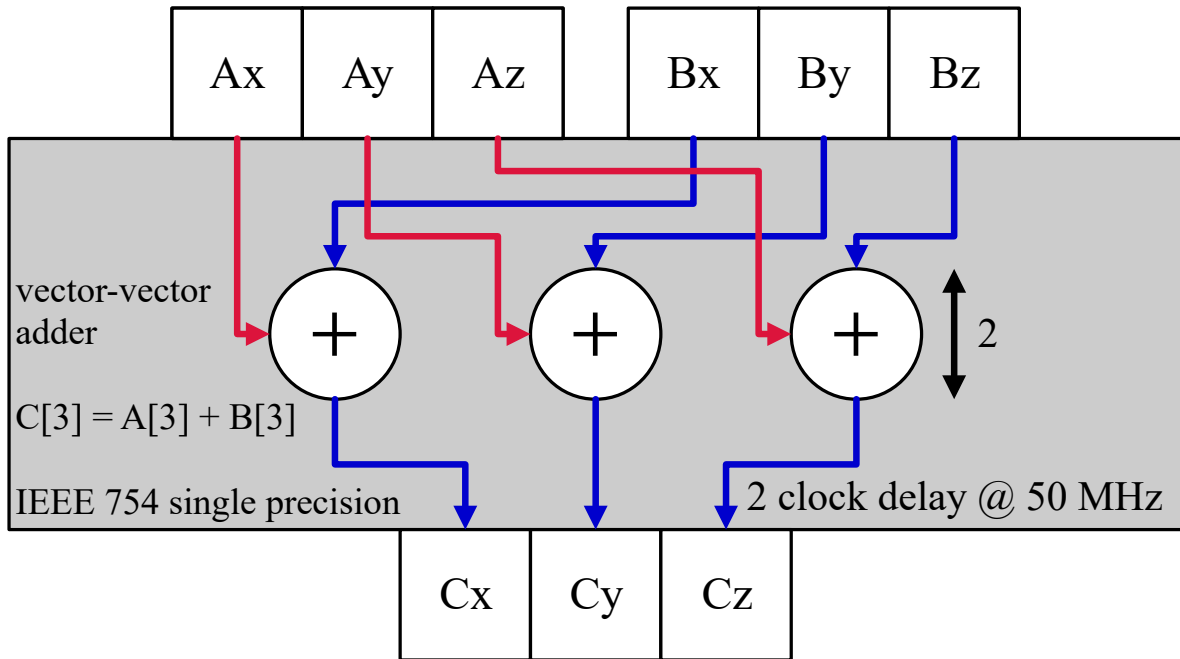


図 5.31 Vector-vector adder circuit

表 5.2 Resource usage by FPGA forward kinematics accelerator

component		ALM logic element			DSP block	
forward kinematics accelerator	matrix-vector multiplier	2589.2 (6.2 %)	3845.7 (9.2 %)	41910 available	9 (8.0 %)	112 available
	vector-vector adder	918.7 (2.2 %)			0	
	state machine	337.8 (0.8 %)			0	
	other application circuit	9659.3 (23.0 %)			0	
free		28405.0 (67.8 %)			103 (92.0 %)	

らに基づき，多様なロボットにおいて低遅延性を損ねずに処理構造を共通化する一般化デバイス制御プラットフォームを提案した．ポリモーフィズムに基づき多様なロボットに向けて拡張可能な実時間ハードウェア制御系実装を提供した．一般化デバイス制御プラットフォームのプラグイン拡張機能により，中央計算機と組込計算機における共通構造の低遅延実時間動作制御を実現した．

フィードバック動作制御計算を組込中間層計算機に展開する手法を検討した．アドミタ

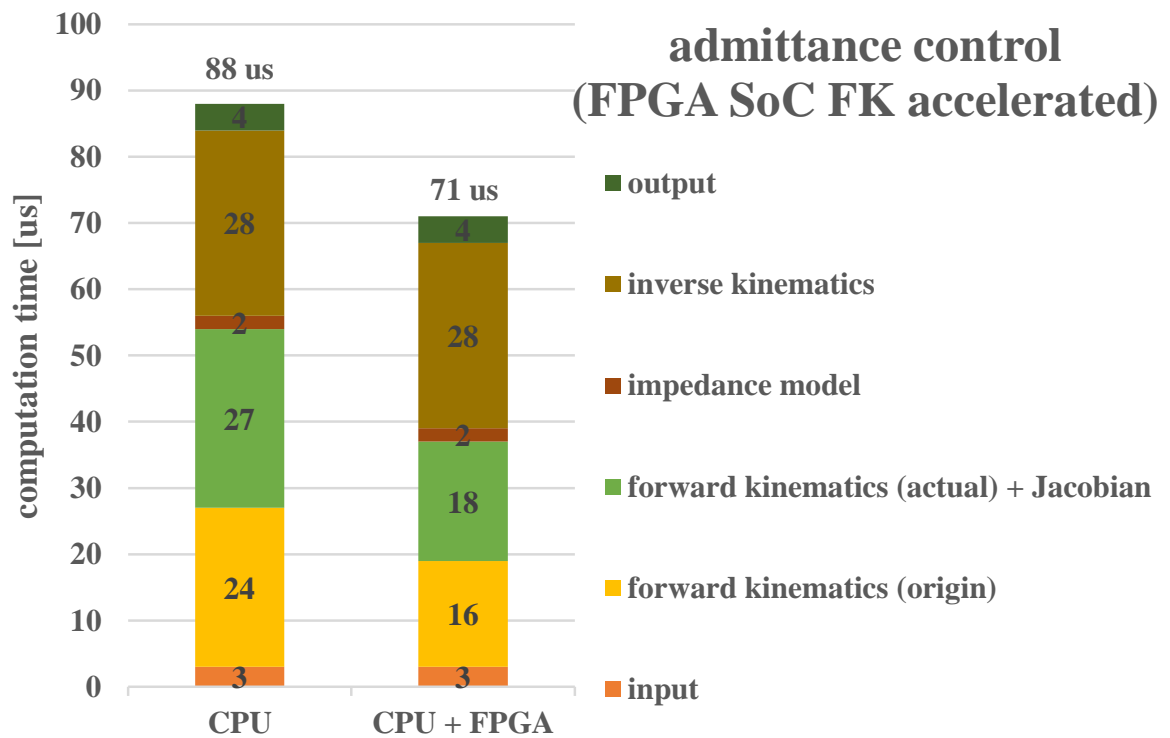


図 5.32 Breakdown of admittance control computation time for 6-axis manipulator with FPGA forward kinematics accelerator

ンス制御計算の実装例を交えて、低遅延なデータ転送に基づく中央計算機に対する優位性を確認した。6 軸マニピュレータのアドミタンス制御の例では、中央計算機では周期タスク実行に $425 \mu\text{s}$ を要する一方で組込中間層計算機の FPGA SoC では $188 \mu\text{s}$ に収まり、およそ 5 kHz の短周期な制御も可能となる。アドミタンス制御における順運動学計算の反復構造を利用して FPGA によるハードウェアアクセラレーションを適用し、更なる低遅延化の可能性を見出した。

第 6 章

体内分散実時間制御システムの応用戦略と 行動実装評価の実現

6.1 はじめに

本研究では体内分散実時間制御システムとして、多自由度ロボットにおけるフィードバック動作制御の低遅延化を主軸に据えつつ、多様なロボットへの技術波及を目指して実時間性および展開実用性を総合的に充足するシステムの設計実現法を提案している。低遅延フィードバックを高信頼に実行する実時間制御システムを、多様な形態のロボットに向けて合理的に応用可能としている。

本章では、いくつかの代表的な形態のロボットを例に挙げ、体内分散実時間制御システムの適用手法を述べるとともに、実機実証による考察と評価を行う。ギア駆動型大出力ヒューマノイドにおいて、剛性の高いハードウェアと低遅延な力フィードバック制御を組み合わせることで精密性と柔軟性を兼ね備えた動作制御を実現し、衝撃的環境接触を伴う行動へ応用する。腱駆動型筋骨格ヒューマノイドにおいて、非常に多自由度な分散制御系を取り扱う実時間制御システムの短周期化を実現する。多リンク飛行ロボットにおいて、大出力で応答性の高い関節力制御に基づく空中マニピュレーションへ応用する。連続跳躍ロボットにおいて、低遅延な姿勢フィードバック制御により弾性駆動系や力制御のしなやかさを活用した跳躍動作へ応用する。

6.2 体内分散実時間制御システムの応用

本研究における体内分散実時間制御システムの応用実装は次に挙げる要素技術の組み合わせにより構成される。

- 高速光通信分散ネットワーク
- 小型高速通信モータ制御モジュール
- 高速高信頼 Ethernet 基幹通信ネットワーク
- USB 高速多ノード通信ネットワーク
- マルチ SoC 搭載組込中間層計算機
- 組込ハードウェアアクセラレーション
- 高速 FPGA サーボ制御
- 分散ローカル推定関節トルク制御
- 一般化デバイス制御プラットフォーム

それぞれの要素技術が、

- 実時間性：低遅延・低ジッタ・信頼性
- 展開実用性：小型軽量・低消費電力・汎用性・運用性

の特性を向上しながら，低遅延なフィードバック制御や環境外乱応答に寄与する．多様なハードウェア構成を有しスペース制限や衝撃的外乱・電氣的ノイズの影響など実装上の制約を多く受ける，多岐に渡る多自由度ロボットへ，いずれかの要素技術の組み合わせを適用することができる．多自由度ロボットに限らず，同様の要求を携える広範なロボット技術ないしオートメーション技術への応用も期待される．

本研究では，特に多自由度で制御の実時間性が要求されるヒューマノイドおよびその技術派生によるロボットに体内分散実時間制御システムを応用する．実装した要素技術の適用手法を提案しその実証評価を行うとともに，同様の体内分散実時間制御システム設計実現法による今後の継続的な研究開発へと繋げる．

6.3 ギア駆動型大出力ヒューマノイドの柔軟環境接触と衝撃緩和への応用

6.3.1 ギア駆動型大出力ヒューマノイドへの応用戦略

3.3.1 項で紹介したギア駆動型大出力ヒューマノイドに関する先行研究の課題を踏まえて，これを解決した上で更に応用的な環境適応行動の実現を目指す．

中央計算機と組込制御系を接続する基幹通信ネットワークとして従来利用されていた PCI バスに代わり，Ethernet を適用する．最新の小型・高性能な計算機製品を採用し，振動などに対しても頑強な高信頼通信を行うことが可能となり，ロボットの小型化設計やダイナミック動作実現の幅が広がる．100 Mbps の EtherCAT を利用した実装と 1 Gbps の Gigabit Ethernet を利用した実装を段階的に適用してゆく．EtherCAT では通信に要する時間がハードウェア制御系の目標基準とする 1 ms の実時間制御周期に対して支配的なほど大きくなってしまう場合がある．実時間性の保証に難があるとともに，周期内に同期実行できる制御処理の量も制限される．また既製品の EtherCAT スレーブモジュールを利用した実装により 1 周期に交換可能なデータ量にも大きな制約が生じる．FPGA による Gigabit Ethernet 実装ではこれらの問題も解決し，中央計算機による全身フィードバック動作制御の低遅延化への足がかりとなる．

組込分散ネットワークに従来利用されていた RS-422 汎用シリアル通信が低遅延化に向けての律速となる場合は高速光通信の適用が有効である．光通信には高速・低遅延なデータ転送性能だけでなく，電氣的ノイズへの耐性向上やケーブルの細線化の効果もある．頻繁に大電流が発生しモータ制御モジュール間の電位差も安定しない劣悪な環境のギア駆動型大出力ヒューマノイドの体内電子制御系において高信頼な高速通信が可能となる．

課題を解決し高信頼な高速通信を達成した上で，組込制御系を利用した低遅延フィードバック動作制御へ応用する．

組込中間層計算機モジュールを適用し、5.2 節で分類したフィードバック動作制御の安定性および外乱応答性を向上する。フィードバックのむだ時間が低減することにより、インピーダンスや接触力などの所望の力学特性を、より広いパラメータ範囲で安定して高追従に再現できる。従来対応できなかった衝撃的な環境接触に対しても低遅延なフィードバックにより応答が可能となり、衝撃緩和や転倒回避が達成される。

モータ制御モジュールごとの推定関節トルク制御を適用することで更にしなやかに外乱に馴染むことができる。関節トルク制御の適用にあたっては、全身フィードバック姿勢制御も重要となる。

低遅延フィードバックにより、ギア駆動ロボットが元来備える精密性の利点を最大限に残しつつ、柔軟性も兼ね備えた動作制御が可能となる。これにより打撃作業や跳躍着地を含む環境適応行動の実現が見込まれる。

6.3.2 体内分散実時間制御システムの要素技術適用事例

本研究で開発を進めている EtherCAT および Gigabit Ethernet を利用した通信インタフェースは、関連するいくつかのロボットにおいて実証的に運用されている。図 3.6 に従来の構成を紹介していた JAXON の例では、PCI 中継基板が図 6.1 のように EtherCAT 中継モジュールに置き換えられている。PCI バスを利用できる旧世代の比較的大型な計算機製品に限られていた制約や、デバイスドライバの保守開発に必要となっていた手間がなくなり、振動にも頑強になったことで、様々なロボット行動研究の礎となっている。3.7.2 項で考察したように EtherCAT 中継モジュールによる通信データ量の不足が生じてきたため、Gigabit Ethernet への換装も進められている。

JAXON の名を継ぐ JAXON3-P [50] では、電子制御系ハードウェア小型化の貢献もあり、これを格納するロボット構造の小型化設計に成功している。JAXON3-P は図 6.2 の跳躍着地をはじめとしたダイナミック動作をいくつか達成している。これは振動に強い Ethernet 通信インタフェースの実装によりなし得たものである。

なお JAXON3-P では跳躍着地行動の実現に向けた戦略として、JAXON のような高減速比ギア駆動構造ではなく、ワイヤ駆動減速機を利用した可変剛性の直列弾性駆動系を採用している [173]。構造上の特性として摩擦の少ないワイヤ駆動減速機と平歯車を採用し関節のバックドライバビリティを高めている。これをブラシレスモータ電流制御で駆動し、簡易的にトルク制御を行っている。本研究で開発した関節トルク制御系とシステム実装を共通とし、バックドライバビリティの高さや直列弾性要素の存在から摩擦や慣性力の影響は小さいものとして省略している。これらの実装によって着地衝撃緩和を達成している。

一方、本研究では低遅延フィードバック制御によるアプローチで衝撃緩和を目指す。ロ

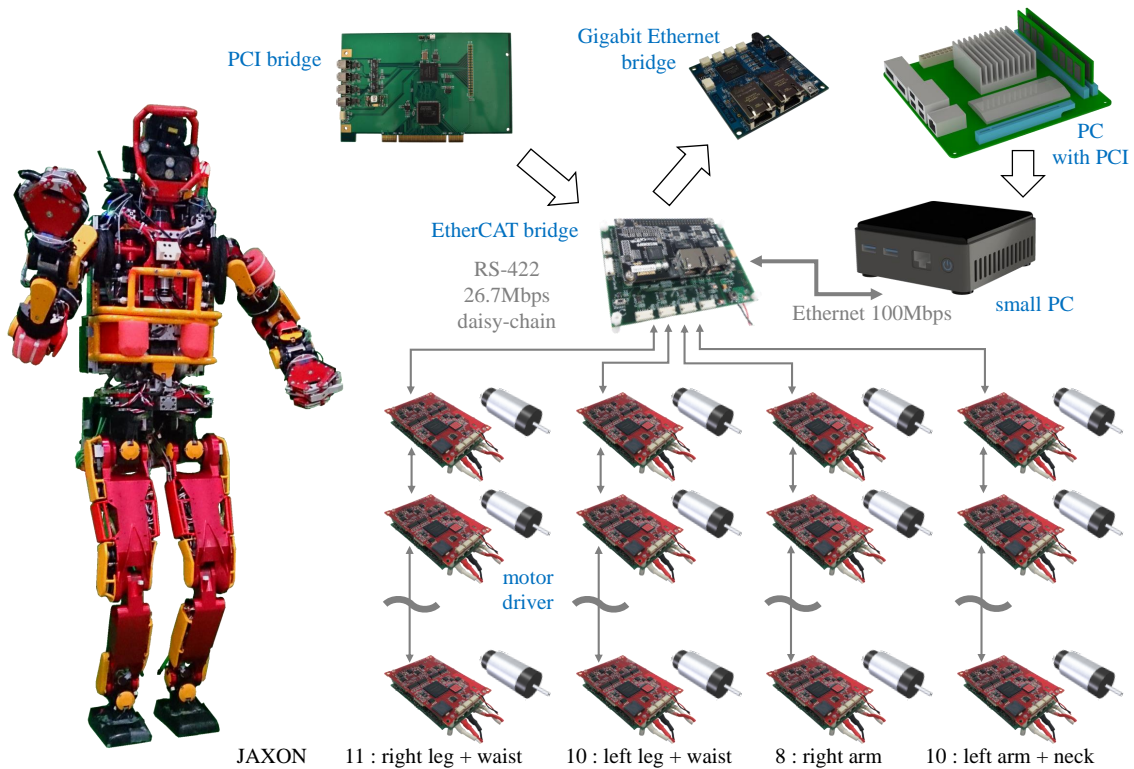


図 6.1 Distributed network for the humanoid robot JAXON with the EtherCAT bridge module

ロボットの耐衝撃戦略において、構造上のバックドライバビリティによるアプローチと制御によるアプローチはよく比較対象として議論に挙げられる [174][175]。しかしこれらは決して排他的なものではなく、両者の技術が向上することで、その適度な組み合わせにより相補的に外乱応答性と制御指令追従性を兼ね備えたシステムが実現されるものと考えられる。

JAXON は摩擦や慣性力を補償した関節トルク制御により、土砂掘削や土砂運搬などの大きな荷重の生じるマニピュレーション行動を達成している [143]。同様の関節トルク制御で 2 脚ロボットが地形変動する不整地や外乱のもとでの安定歩行を達成している [176][152]。

関節トルク制御はハプティック技術へも応用されている。図 6.3 のバイラテラル制御向け操縦コクピットロボット TABLIS では、所望の操作力の再現が求められる。一方で、大出力を実現する慣性の大きなモータと減速機によって生じる抵抗感が課題となる。そこで慣性力を補償した関節トルク制御を適用することで操作性の向上を果たしている。

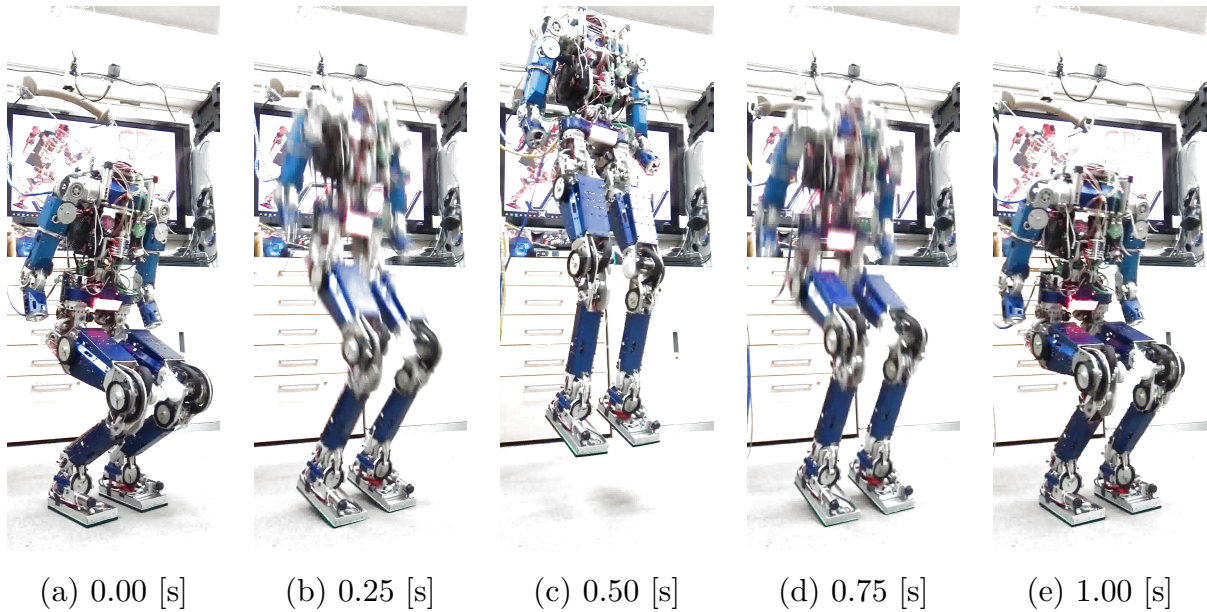


図 6.2 Jumping motion by JAXON3-P [50]

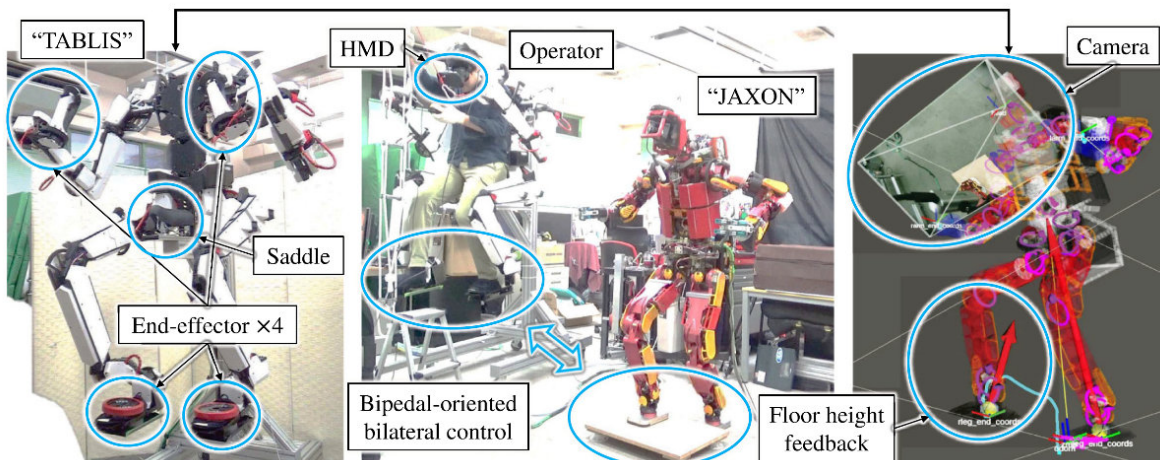


図 6.3 Images of the seat-type whole-body exoskeleton cockpit “TABLIS” and a bilateral quasi-3D locomotion performed by the robot JAXON [177]

6.3.3 腕部マニピュレータにおける安定接触打撃作業

本研究で提案する体内分散実時間制御システムをギア駆動型大出力ヒューマノイドの腕部に相当するマニピュレータに適用し，低遅延フィードバック動作制御を実装する．ロボットが目的となる行動を実現するにあたり，腕部には，操作対象物体に対して適切な接触力を発揮し，あるいは未知の環境接触によって生じる接触力を適切に抑えるための所望

の力学特性の再現が求められる。

マニピュレータではエンドエフェクタの環境接触力や関節のトルク情報に基づいたフィードバック制御を行う。エンドエフェクタの環境接触力は力センサを搭載することで得られる。関節トルク情報は主に、

- 関節に搭載されたトルクセンサの値
- 関節ごとの力学モデルに基づき推定されたトルク値
- ロボットの力学モデルに基づき推定されたトルク値

から得ることができる。関節ごとのローカルトルク制御では利用できないロボット全身の力学モデルを用いたオブザーバなどの推定手法も、全身フィードバックでは有効な手段のひとつとなる。

得られた力・トルクの情報をもとに、

- インピーダンス制御：運動に対して擬似的な機械インピーダンスを再現するだけの接触力を発揮
- アドミタンス制御：接触力を反映して機械インピーダンスモデルに基づいた運動を再生する一種のインピーダンス制御
- 接触力制御：指定された値や範囲の接触力を発揮

に代表される力制御が行われる。インピーダンス制御における機械インピーダンスモデルとしては、仮想質量を設定して慣性を考慮する場合や、より単純に粘弾性のみを考慮する場合がある。仮想質量の設定は制御の安定化に寄与することが知られている [178]。

エンドエフェクタの環境接触力をフィードバックする手法では、エンドエフェクタにおける操作力のみに関して力学特性を設定できる。一方関節トルク情報をフィードバックする手法では、エンドエフェクタ以外にも適用できる場合が多く、全身を利用した環境接触への応用が期待される。

衝撃的な環境接触を伴う物体操作行動は、腕部の力フィードバック制御における遅延が低減されることにより実現の可能性が高まる。接触力やリンク機構内力が低減され、自身や操作対象の損壊が避けられるとともに、ヒューマノイド全身の姿勢安定化も容易になる。過大な接触力の発生を避けるだけでなく所定の接触力を発生させる必要がある場合、中でも特に操作対象の剛性などが不明で接触力が予見できない場合、行動実現は更に難易度の高い課題となる。これは特性が任意に設定できる力フィードバック制御の利点を活かしたまま応答性を向上することで実現の可能性が見いだせる。

衝撃的な環境接触には、腕部が自ら動作することにより発生する能動的なものと、物体が腕部に対して衝突する受動的なものがあり、それぞれ次のような行動において発生する。

- 腕部の能動的な衝突
 - 物体への殴打による加工
 - 衝撃力を利用する道具の操作
 - 物体の打ち飛ばし
 - 飛んでくる物体の打ち返し
- 腕部の受動的な衝突
 - 飛んでくる物体の受け止め
 - 倒れてくる物体の受け止め
 - 外部からの打撃の受け止め
 - ヒューマノイドの移動に伴う環境との衝突
 - ヒューマノイド体幹の姿勢変化による環境との衝突

能動的なものを中心に，接触力の適切な大きさが重要となる場合も多い．そこで体内分散実時間制御システムが重要な役割を果たす．

図 6.4 の 6 軸マニピュレータに体内分散実時間制御システムを適用して評価する．ハーモニックドライブ減速機を搭載した 2 軸直交ユニット [179] を 3 セット利用して構成したマニピュレータである．衝撃的な環境接触を伴う評価実験のため，地面に設置したアンカおよび剛性の高い土台によって固定している．環境接触の実験装置も同様に土台に固定することが可能となっている．

関節リンク機構の構成を図 6.5 に示す．土台に対して全長 1013.75 mm，質量 10.48 kg の 6 軸マニピュレータを固定する．3 セットの 2 軸直交関節を 300 mm の間隔で配置する．末端側の 3 関節が直交するため土台側 3 関節の位置のみにより末端直交関節のデカルト空間位置が決定され，必要であれば逆運動学問題の解析解を得ることも比較的容易である．

6 軸マニピュレータに対して図 6.6 の制御系ハードウェアを構成する．maxon motor 社の EC-4pole 30 200W ブラシレスモータの出力を，ハーモニック・ドライブ・システムズ社の CSD-20-160 もしくは SHD-20-160，および減速比 2.88 のタイミングプーリで減速してそれぞれの関節を駆動する．いずれの関節も総減速比は 460.8 である．ブラシレスモータを小型光通信モータ制御モジュールにより駆動する．組込中間層計算機モジュールと 6 ノードの小型光通信モータ制御モジュールにより高速光通信ネットワークを構成する．中央計算機として Intel 社の NUC 10 Performance Kit NUC10i7FNH を利用し，組込中間層計算機モジュールと Gigabit Ethernet で接続する．エンドエフェクタにワコーテック社の垂直方向定格力 2000 N の 6 軸力センサ DynPick を搭載している．センサデータは小型光通信モータ制御モジュールで読み出して体内分散ネットワークにより中枢全身制御系と共有する．

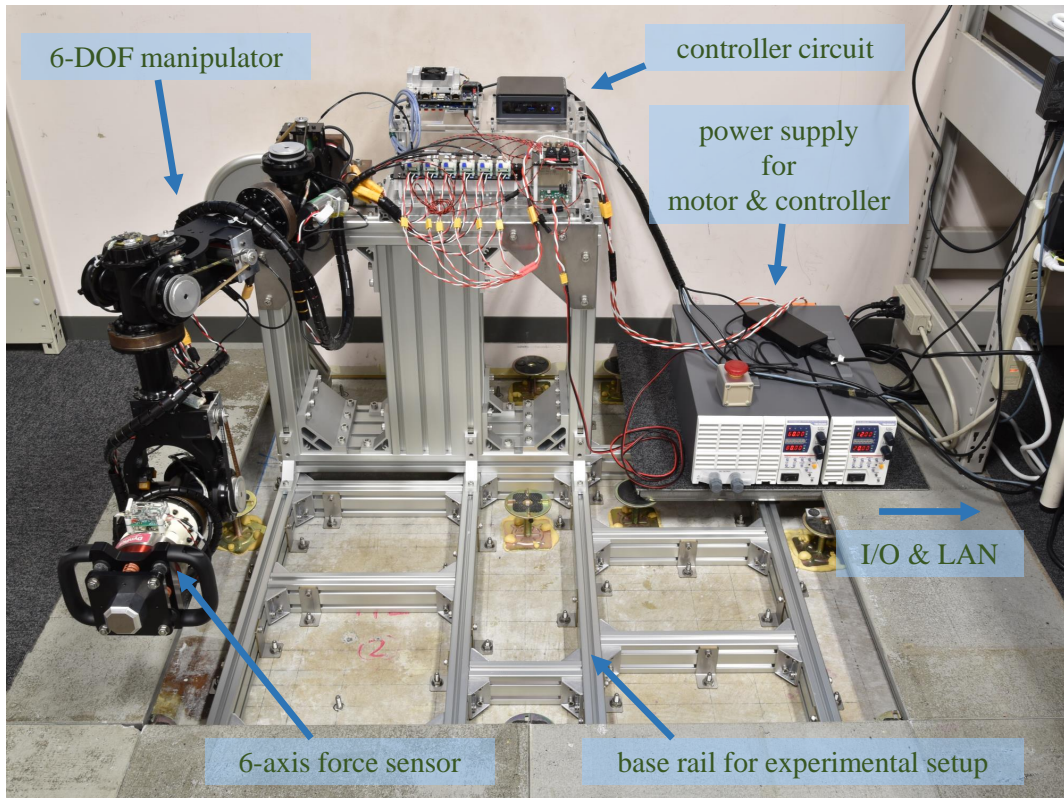


図 6.4 Experimental setup overview of the 6-DOF manipulator

5.6.2 項に述べた粘弾性モデルに基づくアドミタンス制御を中央計算機および組込中間層計算機の FPGA SoC に実装して、受動的な衝突に対する制御遅延の影響を検証する。特異点の影響を受けにくい固定の関節位置を基準姿勢としてマニピュレータに与え、並進 1 方向に粘弾性を設定する。この方向に重りを衝突させて応答を確認する。

図 6.7 の装置により実験を行う。剛性 500 N/m, 粘性 300 N/(m/s), 不感帯 ± 10 N, のパラメータを設定する。振り子の先端に 10 kg の重りを搭載する。重り以外の構造部品は十分軽いものとして質量を無視する。振り子の回転軸から重りまでは 934 mm, 衝突点までは 767 mm となっている。振り子はおよそ最下点でマニピュレータ先端と衝突する。実験では重りを 50mm の高さから落下させる。

衝突時の重りの速度は単純な力学的エネルギー保存則に従うものとして、 $\sqrt{2 \times 9.80665 \times \frac{50}{1000}} \approx 0.99029$ m/s である。衝突点で見ると等価的に、 $10 \times \frac{934}{767} \approx 12.177$ kg の重りの $0.99029 \times \frac{767}{934} \approx 0.81323$ m/s での衝突に相当する。設定した 300 N/(m/s) の粘性および 10 N の不感帯から、単純モデルでの衝突力設計値は最大 $300 \times 0.81323 + 10 = 253.969$ N として期待される。

重りの衝突点等価質量にエンドエフェクタ自体を加えたおよそ 15 kg の質量に対して、剛性 500 N/m, 粘性 300 N/(m/s) から、減衰比はおよそ $300 / (2 \times \sqrt{15 \times 500}) = 2.236$

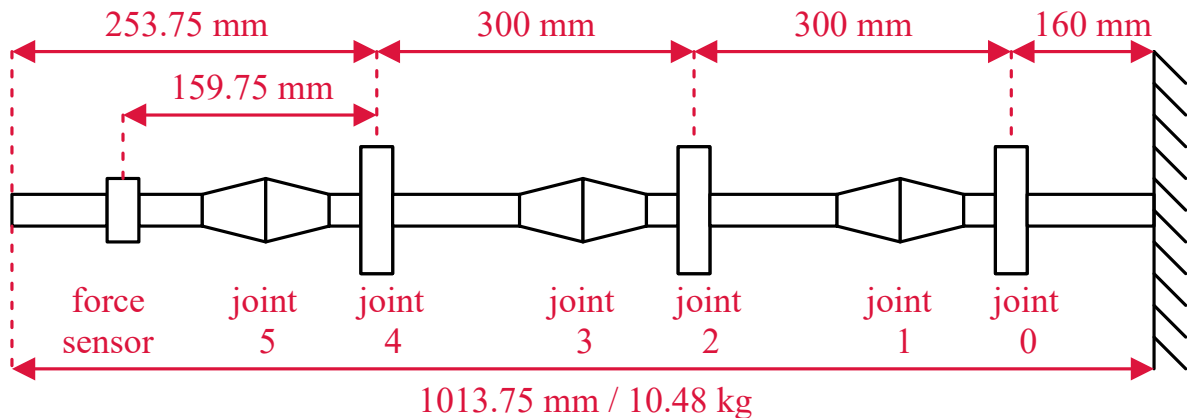


図 6.5 Kinematic property of the 6-DOF manipulator

程度と見積もれる．理想モデルに従えば，重りの運動は衝突後速やかに制動される．

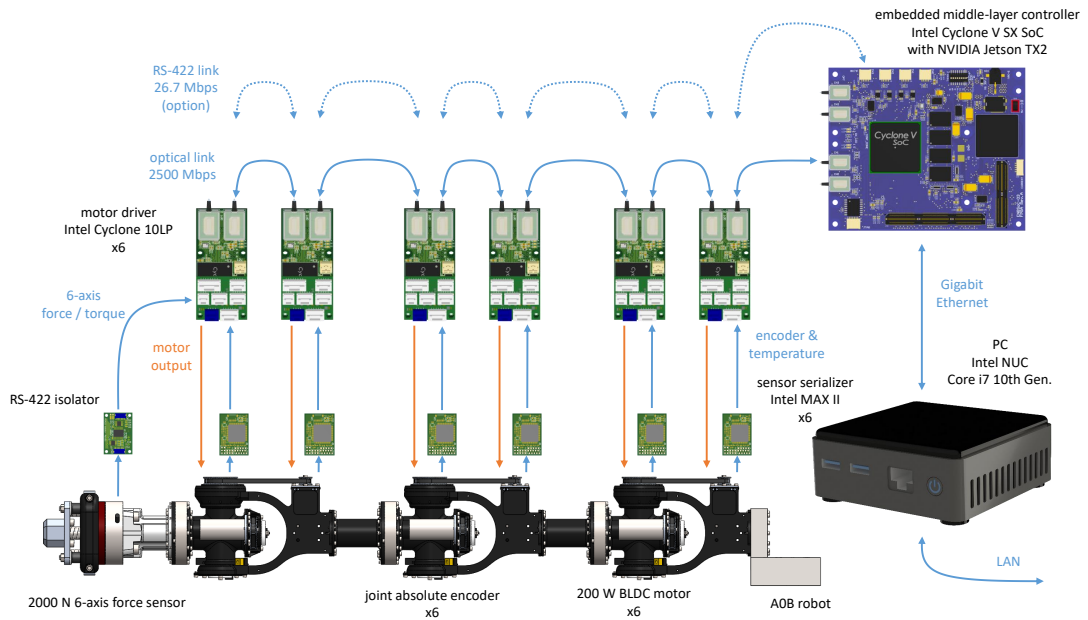
モータ制御モジュールで 1 ms 周期の位置制御を実行し，高速光通信による組込分散ネットワークで接続する．アドミタンス制御は，中央計算機で 1 ms 周期，あるいは FPGA SoC で 0.5 ms 周期で実行する．表 6.1 に示す実験条件で比較する．条件 (A) は FPGA SoC によるアドミタンス制御実装である．条件 (B0)-(B20) は中央計算機によるアドミタンス制御実装であり，センサ値のフィードバックに設定した周期分の遅延を挟む．条件 (C) ではアドミタンス制御を行わず関節位置指令を更新しない．

実験の様子を図 6.8 に示す．動画の記録では重りは落下を開始してからおよそ 500 ms でマニピュレータ先端と衝突している．なお 934 mm の振り子長と 50 mm の高さから計算される最下点到達までの振り子運動時間は 488 ms となる．実行結果が理想モデルとおよそ合致することから，振り子棒質量や振り子軸ベアリング摩擦などの影響は，評価の上で無視しても差し支えない範囲である．

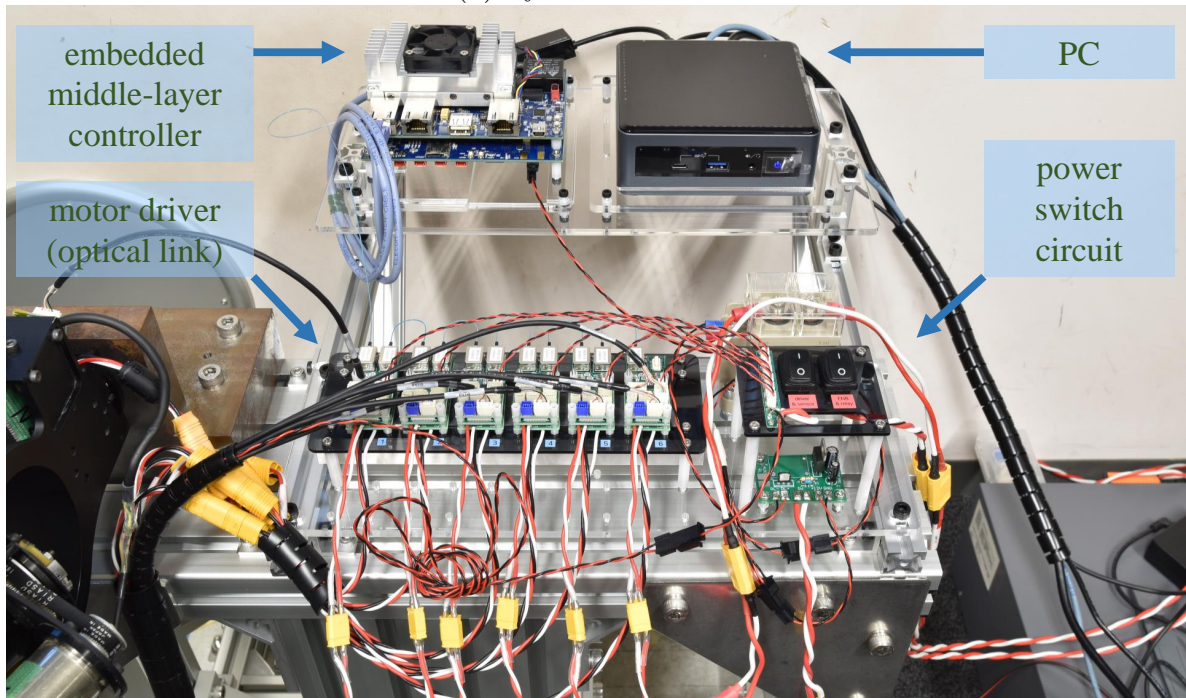
マニピュレータ先端に重りが衝突すると，力センサで測定した接触力をフィードバックして，これに対して機械インピーダンスモデルを再現するマニピュレータ運動が出力される．衝突に反応するようにエンドエフェクタ位置を移動させている様子が確認できる．低遅延な実験条件では，大きな跳ね返りもなく重りが制動される様子が確認された．一方で遅延が大きくなると衝突表面の物理的な剛性に従って重りが跳ね返った．

それぞれの実験条件において観測された衝突接触力の波形を図 6.9 に示す．接触力は立ち上がりからおよそ 30 ms のうちにピークを迎える．遅延が大きくなるにつれてピーク接触力も大きくなる．

実験条件ごとの接触力ピーク値の平均値を図 6.10 に示す．アドミタンス制御を実行しない固定の位置制御では 426.7N のピーク接触力が発生している．中央計算機で 10 ms 以



(a) System overview



(b) Photo view

☒ 6.6 Controller circuit for the 6-DOF manipulator

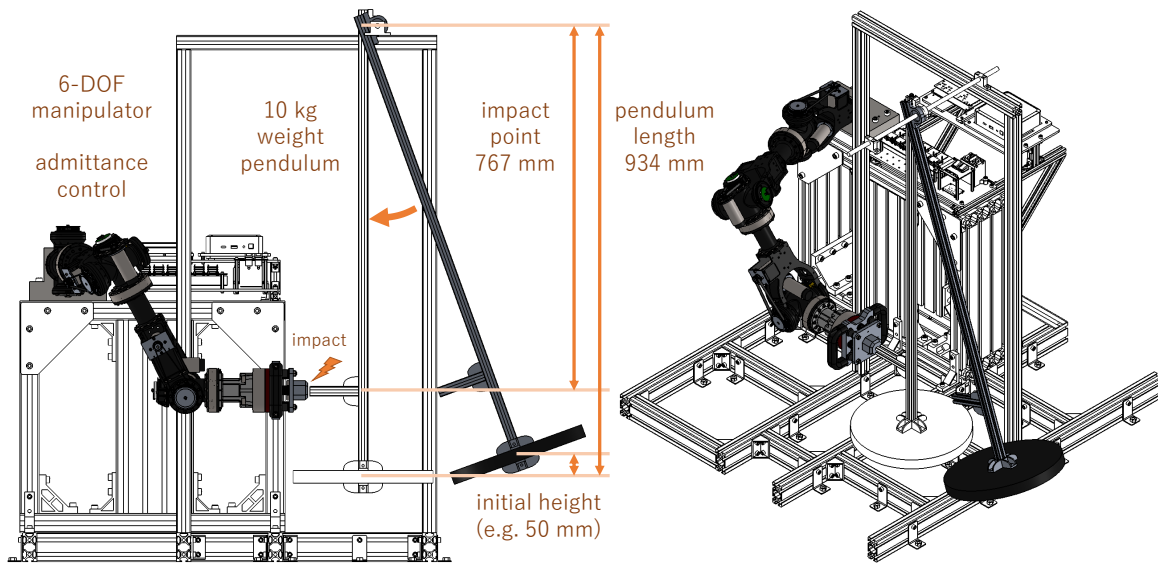


図 6.7 Experimental setup for the 6-DOF manipulator struck by weight pendulum

上の遅延を挿入した場合には、アドミタンス制御を実行しない場合とほとんど変わらない衝突力が発生している。一方小さな遅延においては設計された初期最大粘性衝突力に近い特性が得られている。組込中間層計算機の FPGA SoC で実行した場合にはピーク接触力は 246.6 N となり、設計された初期粘性力を初めて下回っている。

重り衝突実験の結果から、アドミタンス制御における低遅延化の効果が見て取れる。組込中間層計算機での低遅延フィードバック制御により、設定した力学特性モデルに近い結果が得られる。また中央計算機のみを利用する場合においてもソフトウェア遅延を低減することの重要性が確認できる。

続いて、能動的な環境接触力を発揮する作業ヘアドミタンス制御を応用する。適切な接触力を発揮するためには、定常的な接触の安定性と過渡的な衝突への応答性がともに重要となる。

力をフィードバックして位置を更新するアドミタンス制御において、粘性を大きく設定するほど制御は安定となる。位置・速度をフィードバックして力を出力する構造の場合は図 5.6 に示したモータ位置 PD 制御のように粘性力の出力値が過大となり逆に不安定になる場合もある。これとは対照的に、力をフィードバックして位置・速度を出力する構造では、式 5.49 や式 5.51 における力のゲインから分かるように粘性を大きく設定したことが原因で出力が過大となるおそれはない。一方で、粘性を大きく設定すると外乱応答のしなやかさが損なわれ、衝突への対応が困難となる。フィードバック遅延を低減し、粘性を含む、より幅広い機械インピーダンス設定範囲で安定な制御を達成することが重要である。

図 6.11 の装置により能動的な環境接触の実験を行う。頑強な土台で枠を構成し、枠の上

表 6.1 Experimental condition of controller for the 6-DOF manipulator struck by weight pendulum

condition ID	computer	additional delay
(A)	FPGA SoC	0 ms
(B0)	PC	0 ms
(B1)	PC	1 ms
(B2)	PC	2 ms
(B3)	PC	3 ms
(B4)	PC	4 ms
(B5)	PC	5 ms
(B6)	PC	6 ms
(B8)	PC	8 ms
(B10)	PC	10 ms
(B12)	PC	12 ms
(B15)	PC	15 ms
(B20)	PC	20 ms
(C)	fixed position	-

に板材などの物体を設置して鉛直方向に接触する．図 6.11 の例では枠の上にステンレス板を設置している．枠の横幅は内寸 290 mm，外寸 350 mm である．

3 個の直交関節が水平となりエンドエフェクタを下に向けた基準姿勢から，エンドエフェクタを鉛直方向に上下させる動作を基本として実験する．基準姿勢のエンドエフェクタ先端から枠の上面まではおよそ 145 mm の距離である．実験ではエンドエフェクタの先端を 200 mm 降下させる位置指令を与える．およそ 55 mm に物体の厚みを加えた距離だけ物体を貫通する位置指令である．

モータ制御モジュールでの位置制御は MCU ではなく FPGA で実行する．アドミタンス制御は中央計算機もしくは FPGA SoC でともに 1 ms 周期で実行する．エンドエフェクタを上下させる基本動作の位置・速度指令は中央計算機から 1 ms 周期で与える．

定常的な接触安定性における遅延の影響を確認する．枠上に硬いステンレス板を設置して，エンドエフェクタを降下させる．ステンレス板は SUS303 材で板厚 10 mm，幅 60 mm である．エンドエフェクタの降下軌道は，静止した始点から静止した終点までの関節ごとの躍度最小軌道で与える．降下させたエンドエフェクタ指令位置からステンレス板上面までは 65 mm の距離となる．

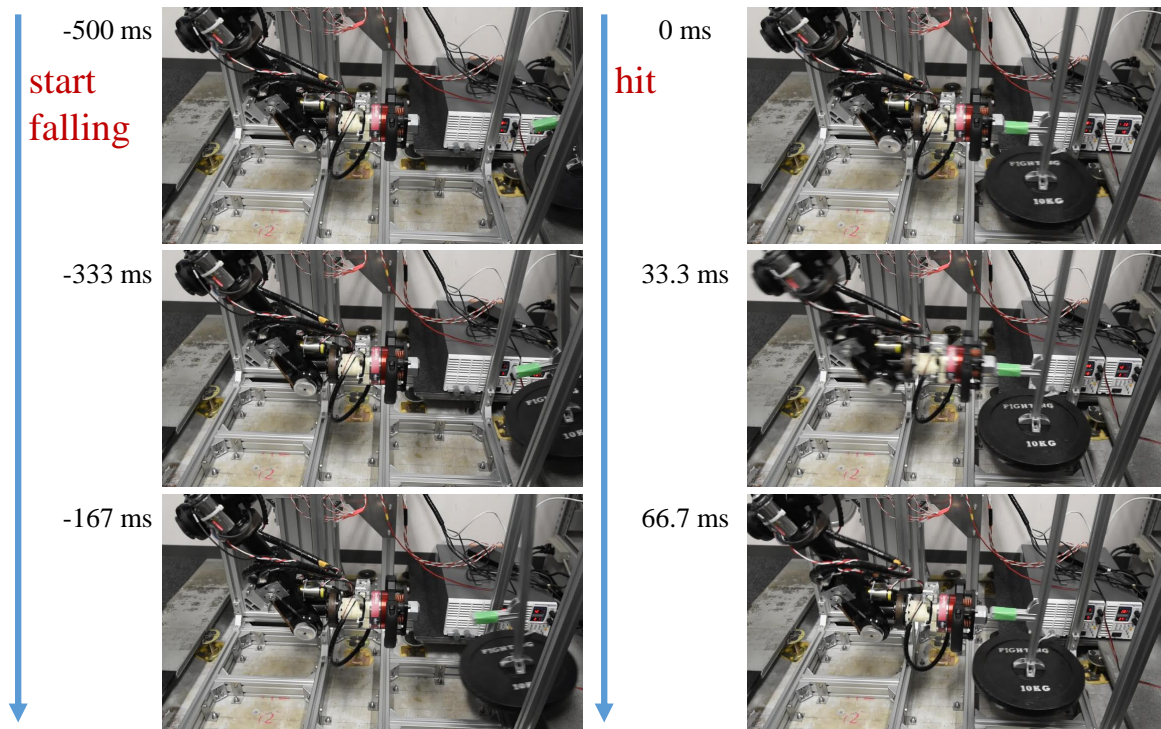


図 6.8 The 6-DOF manipulator struck by 10 kg weight pendulum from 50 mm height

剛性 2000 N/m，不感帯 ± 30 N の設定で粘性と遅延を変更して挙動を調べる．エンドエフェクタを基準姿勢から 200 mm 下まで 2000 ms で降下させる．正の粘性のもとで理想的な機械インピーダンスモデルに基づけば，およそ $2000 \times \frac{65}{1000} + 30 = 160$ N の接触力で安定に釣り合う．むだ時間の影響により制御が不安定になると接触力が発散的に振動する．0 N に近い値まで低下する振動を繰り返すと，物体との接触を保つことができない．ステンレス板の場合はアルミニウム合金製の枠との摩擦が小さいため，振動的な接触の繰り返しにより板の位置が移動して枠から落下する．

表 6.1 の条件 (A) および条件 (B0)-(B4) の遅延条件で粘性の設定値に対する安定性を確認する．振動が発散しステンレス板との接触を長時間保つことができなかった条件を不安定と判定する．なお制御が不安定となり振動した場合でも接触を保つことができる程度であれば安定とするため，厳密な安定性の評価とは異なる．

それぞれの遅延で接触を保つことのできた粘性設定値の条件を表 6.2 に示す．200 N/(m/s) から 500 N/(m/s) までの粘性で比較する．エンドエフェクタに集中する等価質量をマニピュレータ全体質量の 3 分の 1 であるおよそ 3.5 kg と近似した場合の減衰比を記載している．組込中間層計算機の FPGA SoC で実行した低遅延なアドミタンス制御では粘性を 250 N/(m/s) まで下げることができる．遅延が増大するにつれて許容できる粘性設定値が限られる．

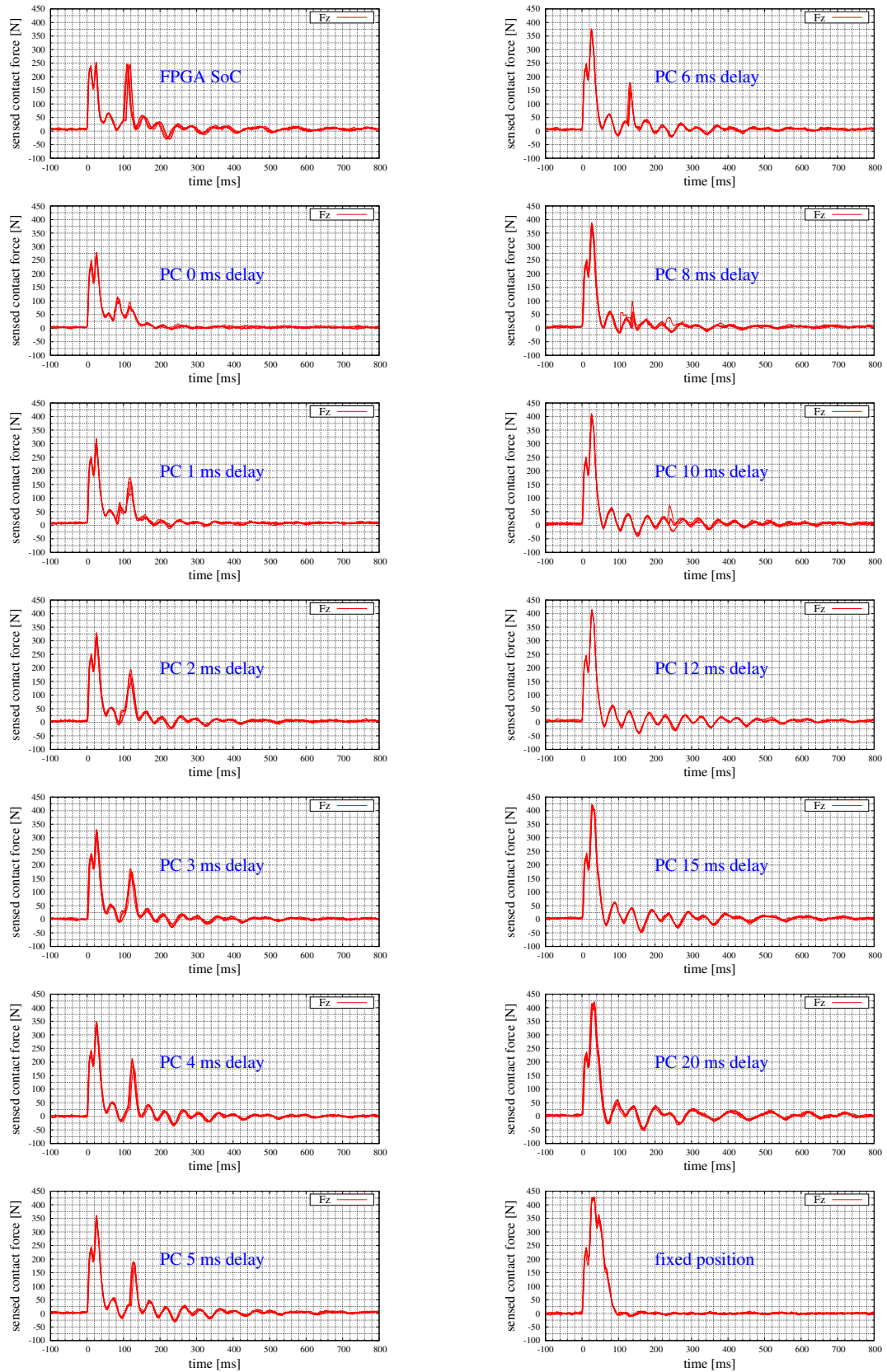


図 6.9 Sensed contact force on the 6-DOF manipulator struck by weight pendulum (comparing between (A) admittance control on FPGA SoC, (B0)-(B20) admittance control on PC with 0-20 ms variable delay, and (C) fixed position with no admittance control)

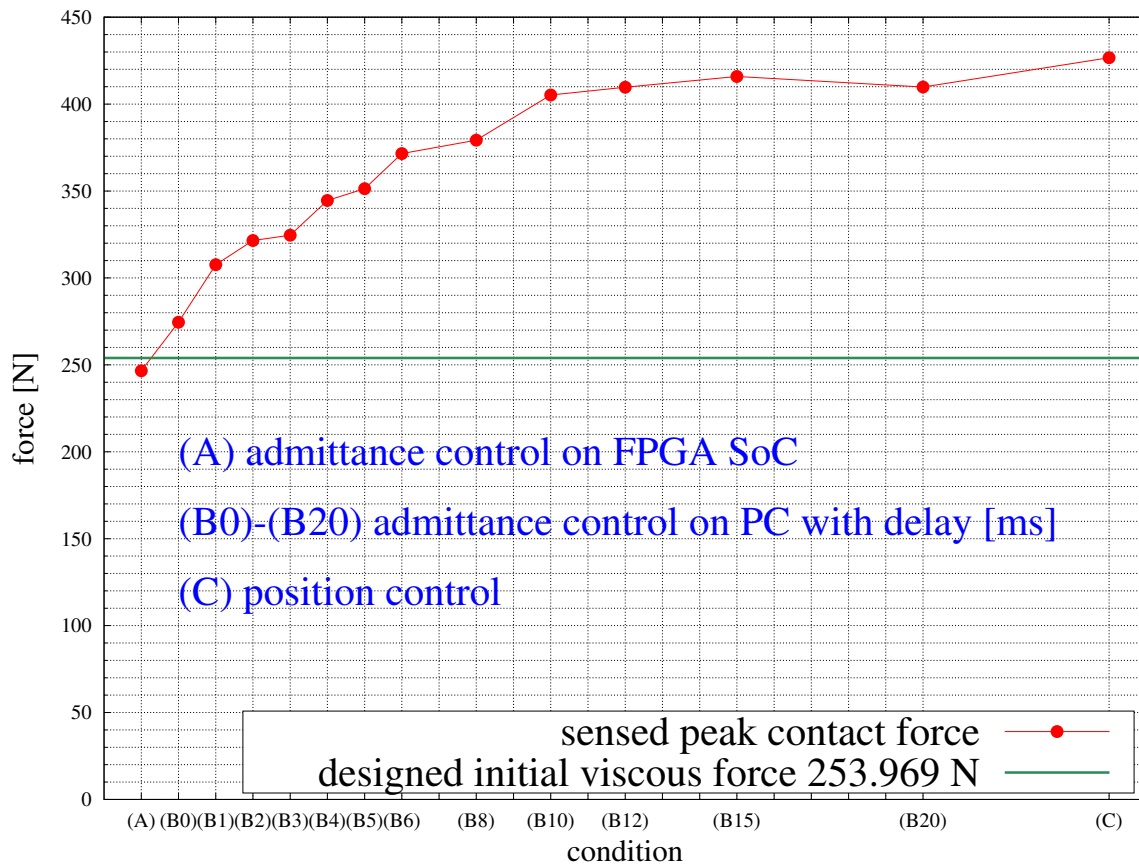


図 6.10 Sensed peak contact force on the 6-DOF manipulator struck by weight pendulum (comparing between (A) admittance control on FPGA SoC, (B0)-(B20) admittance control on PC with 0-20 ms variable delay, and (C) fixed position with no admittance control)

粘性 $350 \text{ N}/(\text{m}/\text{s})$ の場合の接触力と関節速度の波形を図 6.12 に示す．接触力には力センサ値の鉛直方向成分を用いる．関節速度には図 6.5 の中で，この実験の接触姿勢において最も速度が大きくなる joint 4 の値を用いる．

ステンレス板と接触した瞬間を時刻 0 としてプロットしている．アドミタンス制御を組込中間層計算機の FPGA SoC で実行する条件 (A) では安定した接触を達成している．小さな振動でおよそ 150 N の接触力を保つことができている．一方で，中央計算機で余分な遅延を挿入しない条件 (B0) では，接触を保てないほどでは無いまでも発散的な振動が生じている．中央計算機で 1 ms の遅延を挿入した条件 (B1) では更に大きな振動が生じている． 2 ms の遅延を挿入した条件 (B2) では大きな振動により接触を保つことができず，途中でステンレス板が枠から落下して終了している．

低遅延フィードバックに基づくアドミタンス制御で衝突への過渡的な応答と安定した柔

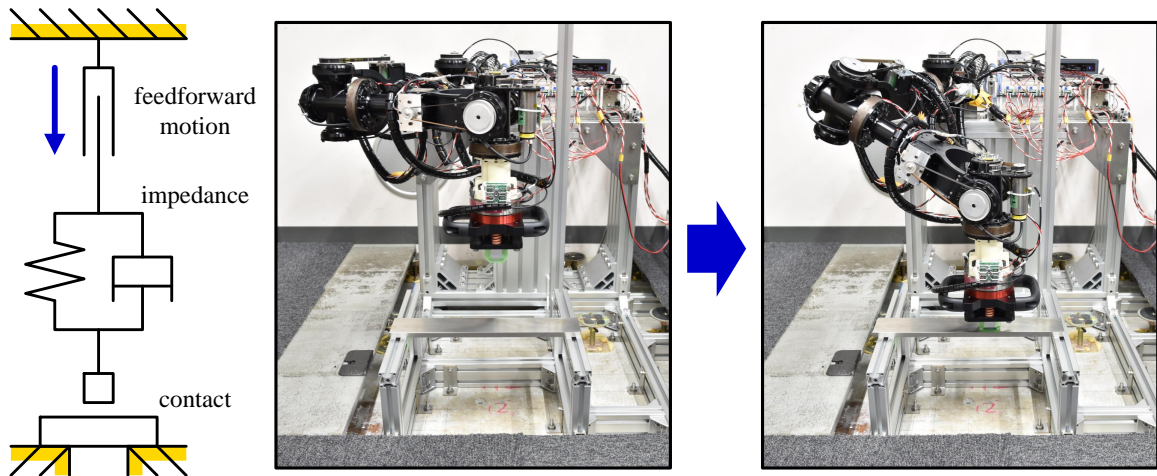


図 6.11 Experimental setup for the 6-DOF manipulator touching/smashing object

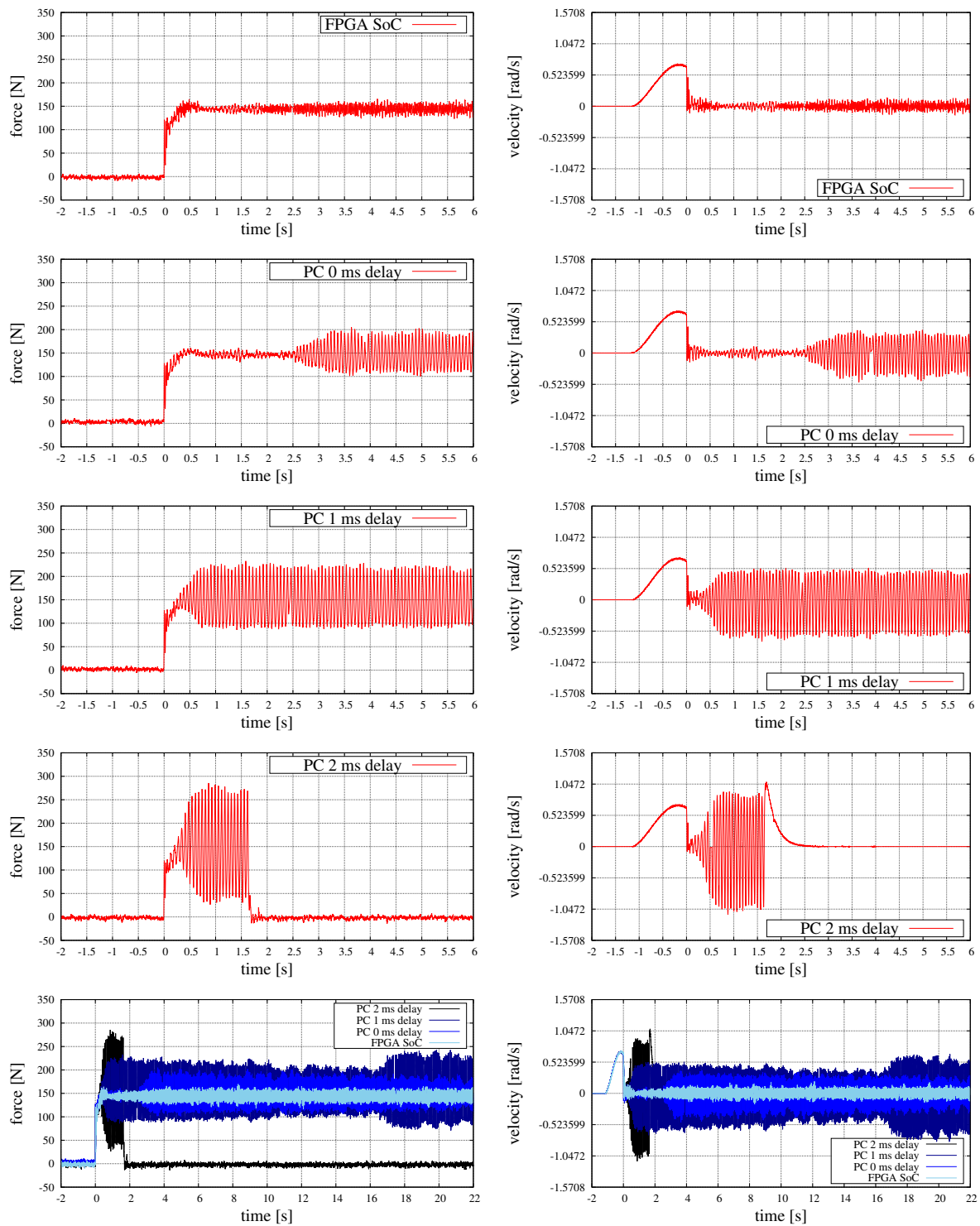
表 6.2 Shmoo plot for stability of the 6-DOF manipulator touching object (* = stable)

damping coefficient [N/(m/s)]	approximate damping ratio	FPGA SoC (A)	PC with delay [ms]				
			0 (B0)	1 (B1)	2 (B2)	3 (B3)	4 (B4)
200	1.2						
250	1.5	*					
300	1.8	*	*				
350	2.1	*	*	*			
400	2.4	*	*	*	*		
450	2.7	*	*	*	*	*	
500	3.0	*	*	*	*	*	*

軟な環境接触が実現されたことで、打撃作業への応用が可能となる。アドミタンス制御の剛性および打撃対象物体を変更した 3 通りの条件での打撃作業実験の様子を図 6.13 に示す。

エンドエフェクタを基準姿勢から 200 mm 下まで 300 ms の短時間で降下させる。アドミタンス制御は組込中間層計算機において 1 ms 周期で実行する。3 条件ともに組込中間層計算機により安定な接触保持が可能となった 350 N/(m/s) の小さな粘性を設定することで柔軟な打撃を図る。また不感帯 ± 30 N を設定している。

図 6.13 (a) では木材板を低剛性で打撃している。厚さ 3 mm、幅 50 mm のアガチス材を用いる。剛性は 500 N/m と設定する。図 6.13 (b) では同じ木材板を 10000 N/m の高



(a) Vertical contact force

(b) Major joint velocity

図 6.12 Force and joint velocity in stable/unstable contact of the 6-DOF manipulator touching stainless steel plate ($K = 2000 \text{ N/m}$, $D = 350 \text{ N/(m/s)}$, comparing between (A) admittance control on FPGA SoC and (B0)-(B2) admittance control on PC with 0-2 ms variable delay)

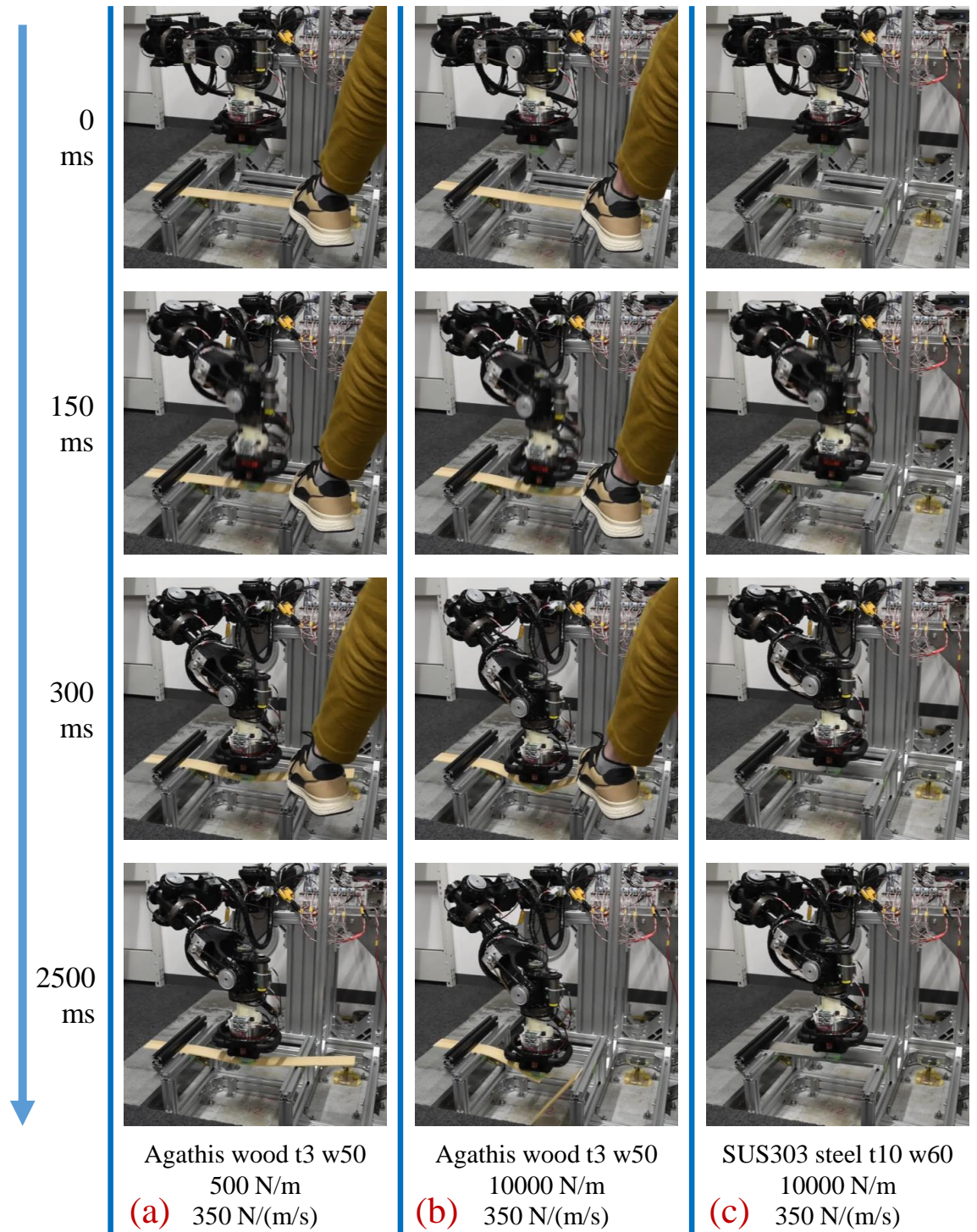


図 6.13 The 6-DOF manipulator smashing object in 300 ms motion

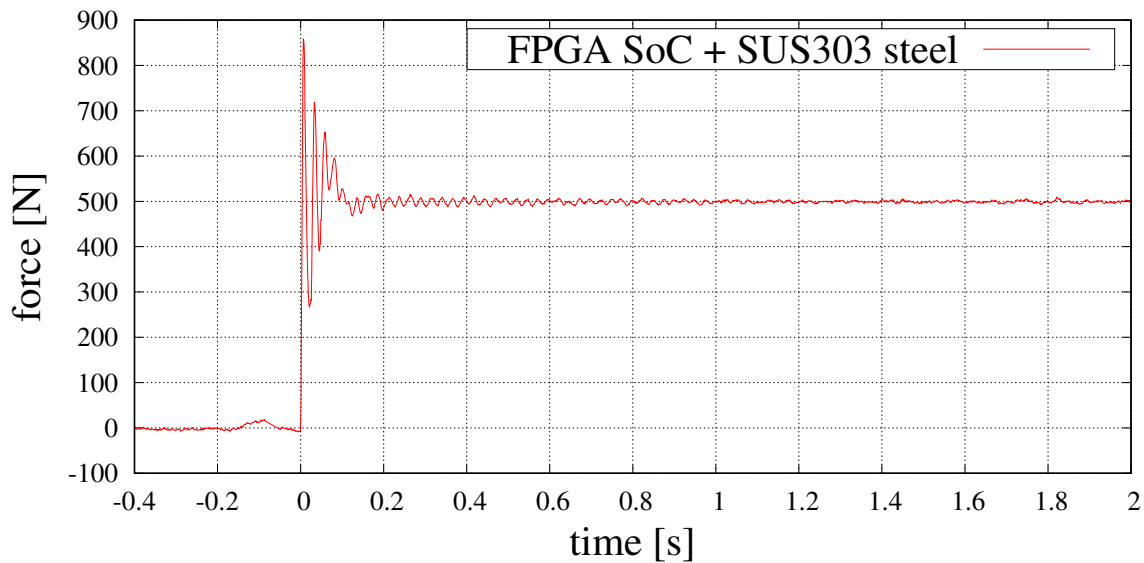


図 6.14 Contact force of the 6-DOF manipulator smashing stainless steel plate in 300 ms motion ($K = 10000 \text{ N/m}$, $D = 350 \text{ N/(m/s)}$)

剛性で打撃している．図 6.13 (c) では硬い金属板を 10000 N/m の高剛性で打撃している．厚さ 10 mm ，幅 60 mm の SUS303 ステンレス板を用いる．

剛性を変更することで，木材板を割らないしなやかな接触と，木材板を叩き割る十分な力での打撃を使い分けることができる．木材板を叩き割れるだけの十分な力を発揮する打撃を，硬くてほとんど変形しない金属板に対しても実行することができる．金属板を打撃した際の接触力を図 6.14 に示す．衝突後速やかに安定した接触力へ収束している．このように低遅延フィードバックによって剛性が未知の対象物体に対する安全な打撃作業が実現される．

6.3.4 脚部における跳躍着地動作

本研究で提案する体内分散実時間制御システムをギア駆動型大出力 2 脚ロボットに適用し，低遅延フィードバック動作制御を実装する．脚部では，転倒や過大な地面反力の発生を防ぎながら所望の移動を達成するための，衝撃緩和や姿勢安定化が求められる．

脚部では，エンドエフェクタの環境接触力，すなわち足部の地面反力や，ロボット体幹姿勢情報に基づいたフィードバック制御が行われる．関節のトルク情報が得られる場合は，足部以外での環境接触に対応することも可能となる．これらの情報をもとに，

- 地面反力制御：地面反力の大きさや脚間の分配を適切に保持
- 着地衝撃緩和制御：着地時の過大な地面反力を抑制

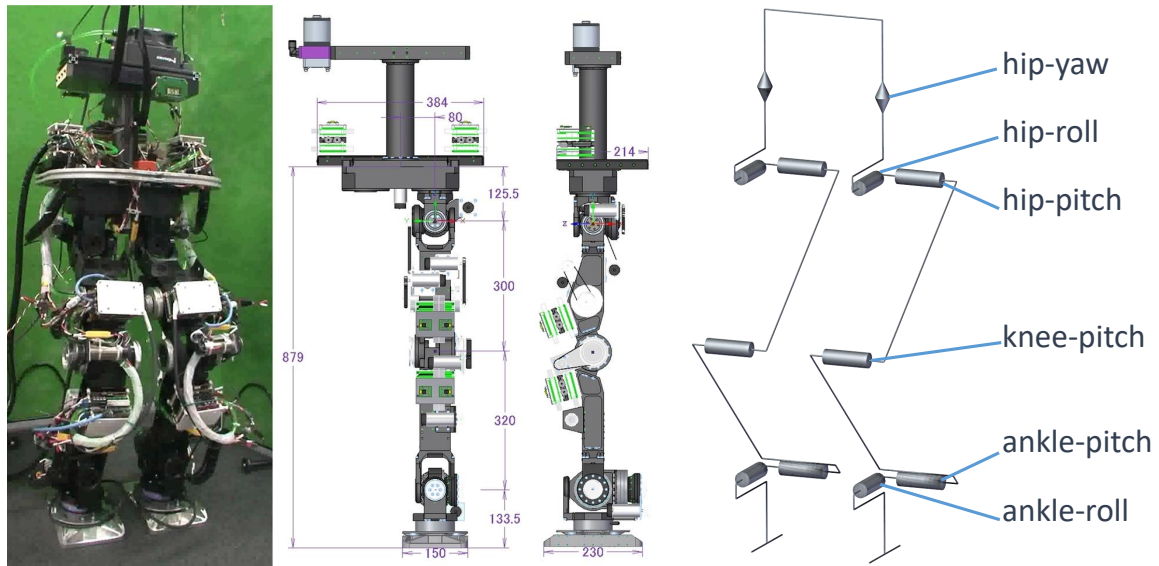


図 6.15 Overview of the high-power biped robot

- 姿勢安定化制御：体幹姿勢や ZMP を適切に保持
- プッシュリカバリ：接地状態を保てない外乱に対して歩行などを交えて転倒回避

のようなフィードバック動作制御を実行しながら，所望の移動を達成するための動作を再現する．

図 6.15 の 2 脚ロボットに体内分散実時間制御システムを適用して評価する．伊藤らにより開発された 2 脚ロボット [180] をもとに仕様変更と改造が重ねられた，各脚 6 関節，計 12 自由度の実験用 2 脚ロボットである．質量は全体でおよそ 48.5 kg である．関節の減速比および定格トルクを表 6.3 に示す．全ての関節において総減速比は等しく 240 であるが，特に大きな負荷の生じる膝関節のみ構成が異なり，定格トルクの大きな減速機が搭載されている．通常の運用においては瞬間許容最大トルクを超えないことが望ましく，減速機の歯飛びが生じるラチェティングトルクは超えてはならない．

2 脚ロボットに，推定関節トルク制御を実装した大出力モータ制御モジュール，および EtherCAT 中継モジュールを搭載し，トルク制御による着地衝撃緩和実験を行う．モータ制御モジュールごとのトルクフィードバック制御実験を通して，電子制御による着地衝撃緩和の可能性を確認する．

このときの体内制御システムの構成を図 6.16 に示す．中央計算機としてイノテック社の小型 100 mm 角 CPU ボード IX-5050 を搭載している．PC と EtherCAT 中継モジュールが接続され，大出力モータ制御モジュールにより左右の脚でそれぞれ 1 系統のディジチェーン接続ネットワークが構成されている．足首関節には評価実験に向けて，旧第

表 6.3 Joint reduction ratio and maximum torque of the high-power biped robot

joint	pulley ratio	Harmonic Drive ratio	Harmonic Drive model number	instantaneous maximum torque	ratcheting torque
hip-yaw	1.5	160	CSD-20-160	76 Nm	220 Nm
hip-roll	1.5	160	CSD-20-160	76 Nm	220 Nm
hip-pitch	1.5	160	CSD-20-160	76 Nm	220 Nm
knee-pitch	2.4	100	CSD-25-100	152 Nm	500 Nm
ankle-pitch	1.5	160	CSD-20-160	76 Nm	220 Nm
ankle-roll	1.5	160	CSD-20-160	76 Nm	220 Nm

一精工社の定格トルク 330 Nm の静電容量型トルクセンサ，およびニコン社の光学式アブソリュートエンコーダ SAR-ML50A を搭載している．足部にはワコーテック社の定格鉛直力 1000 N の 6 軸力センサ DynPick を搭載している．トルクセンサ・アブソリュートエンコーダ・力センサの値はモータ制御モジュールで読み出され，体内分散ネットワークを経由して収集される．体幹には姿勢センサとして Xsens 社の MTi-100 を搭載し，中央計算機から直接 USB 通信によりデータを取得する．

2 脚口ボットを空中から落下させることで着地衝撃緩和性能を検証する．図 6.17 に示す実験装置を用いる．クレーンリフトに装着された電磁石の吸着により 2 脚口ボットを空中に吊り下げてから，吸着を遮断することで地面に向けて落下させる．これにより初期高さを可変とした自由落下実験を行う．

2 脚口ボットの各関節に高剛性位置制御およびコンプライアンス制御を適用し，着地衝撃を比較する．それぞれの関節制御のもとで 10 mm・30 mm・50 mm の高さから落下させた際の膝関節推定トルクを図 6.18 に比較する．落下開始時刻を 0 としてプロットしている．重力加速度により計算される落下時間は表 6.4 に示す通りであり，実験結果もおよそこれに合致する．50 mm からの落下では，位置制御の場合 180 Nm に達しているピークトルクが，コンプライアンス制御の場合は 130 Nm に抑えられている．着地からおよそ 10 ms のうちに急峻に立ち上がる衝撃トルクに対してモータの制御出力が高速に応答できていることが確認できる．

関節トルクフィードバック制御と全身着地動作の組み合わせにより，より高所からの落下着地を実施する．関節ごとのコンプライアンス制御および衝撃緩和制御によって着地瞬間の衝撃力を緩和する．中枢全身制御系では関節から得られたトルク情報から着地を検出し，膝を曲げて着地動作を行う位置指令およびトルク指令を関節に与える．着地動作は，

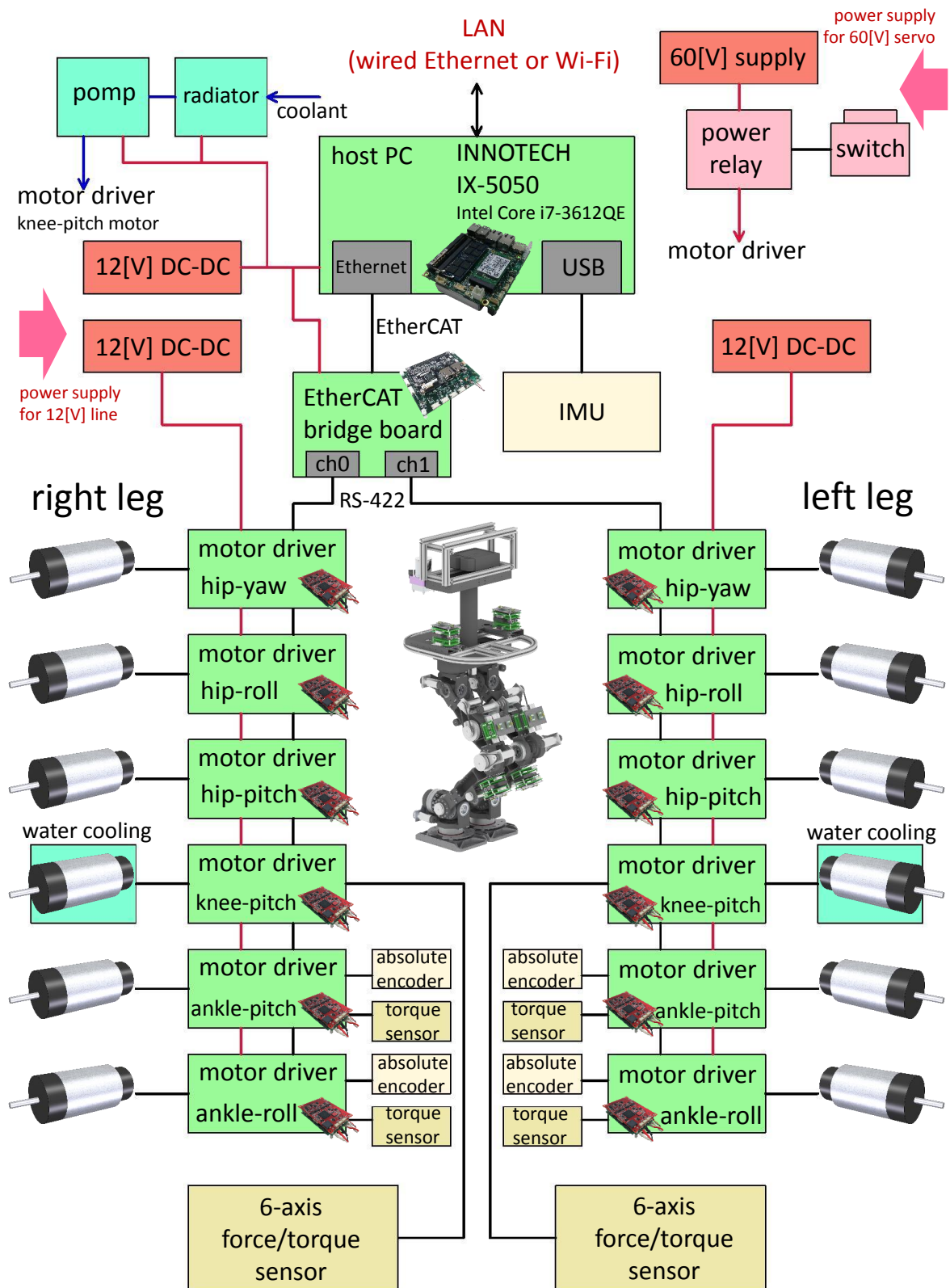


図 6.16 Electrical Controller System on the High-Power Biped Robot

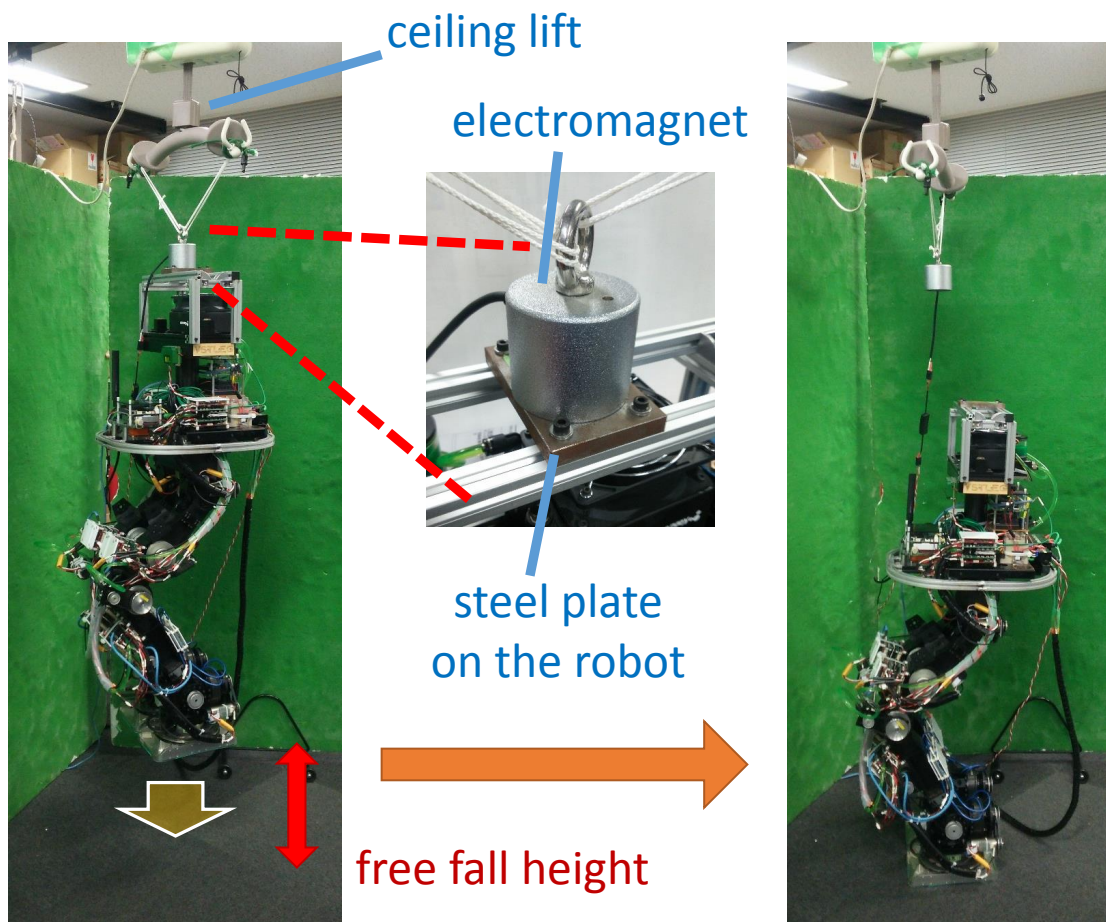


図 6.17 Free fall experiment setup with electromagnet for the high-power biped robot

表 6.4 Calculated landing time after free fall

initial height	landing time
10 mm	45.16 ms
30 mm	78.22 ms
50 mm	100.98 ms

着地瞬間の初期速度から一定の加速度で脚を曲げる放物線軌道として与える。

250 mm の高さから落下させた実験の様子を図 6.19 に示す。このときの全関節の推定トルクは図 6.20 のようになった。表 6.3 に示した関節のラチェッティングトルクを下回っていることが確認できる。また関節にエンコーダを搭載した足首部分で実際に歯飛びが生じていないことを確認する。図 6.21 に足首関節のモータエンコーダと関節エンコー

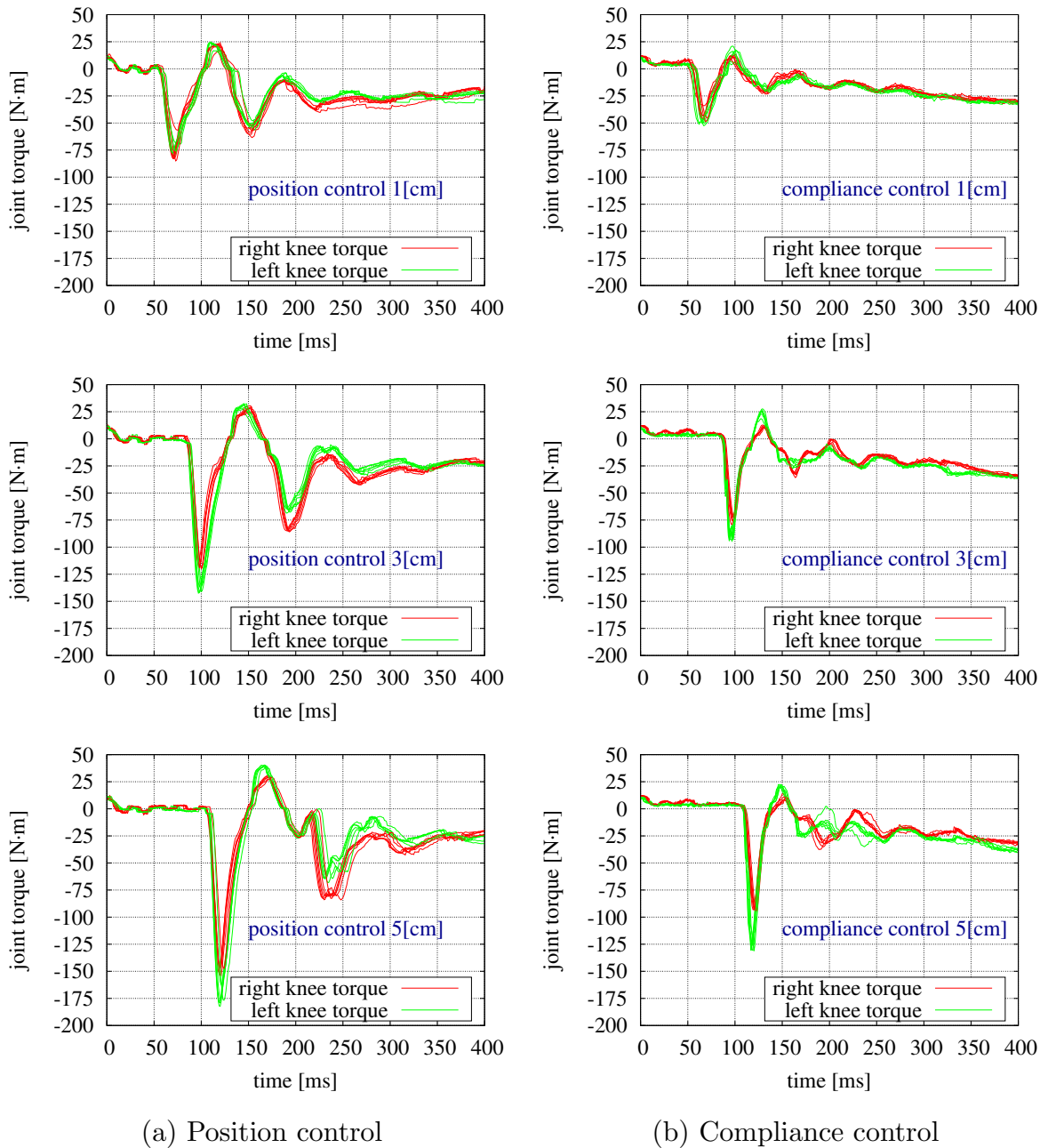


図 6.18 Estimated knee joint torque of the high-power bided robot dropped from 10 mm, 30 mm, and 50 mm height

ダの差分を示す．着地後にエンコーダの差分が減速ギアのピッチよりも十分小さな値に収束しており，問題となるギアの歯飛びが生じていないことが確認できる．

即応的なモータ制御出力により着地衝撃の緩和が可能であることを確認した．一方で跳躍を含むような一連の行動を実現するには，しなやかに外乱に馴染むだけでなく姿勢を保ちながら精密に動作を再現することも求められる．精密な跳躍動作と柔軟な着地動作を両

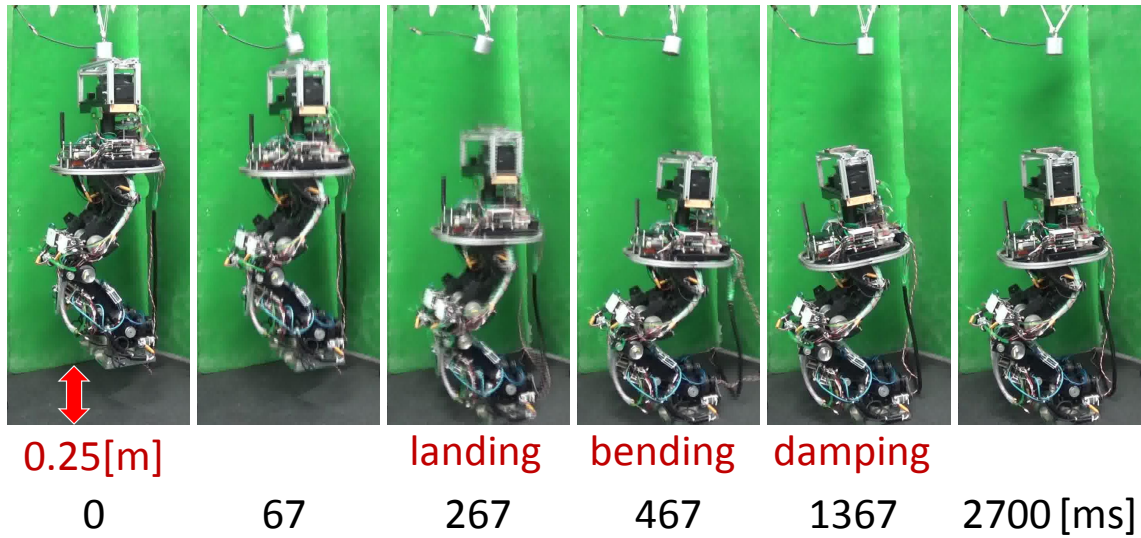


図 6.19 The high-power biped robot dropped from 250 mm height

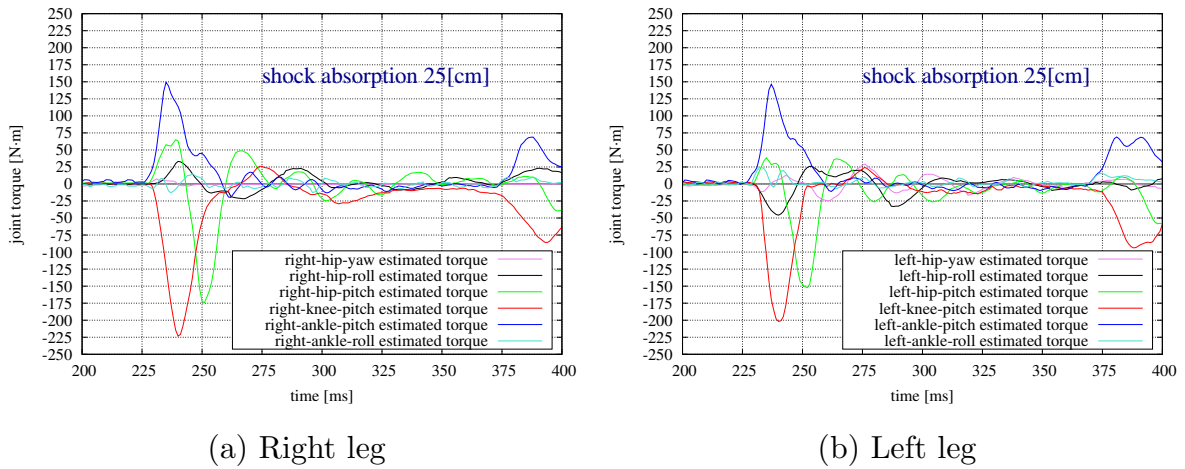


図 6.20 Estimated joint torque of the high-power biped robot dropped from 250 mm height

立するためには、全身フィードバック動作制御が重要な役割を果たす。これには、

- 関節を柔軟なトルク制御型として全身フィードバックにより動作の精密性を補償する方法
- 関節を高剛性な位置制御型として全身フィードバックにより着地衝撃を緩和する方法

が考えられる。前者のトルク制御型では関節ごとの高速応答により十分な衝撃緩和性能が

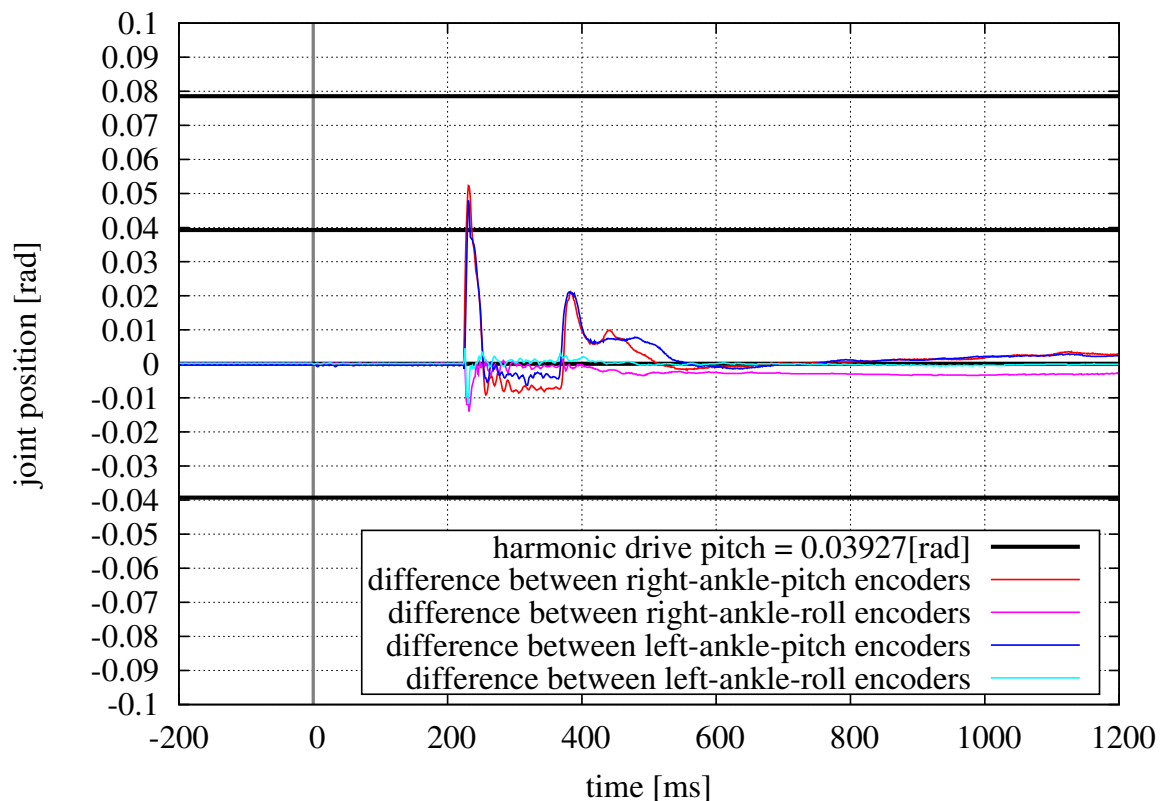


図 6.21 Difference between the motor encoder and the joint encoder on ankle joints of the high-power biped robot dropped from 250 mm height

達成されるが，安定した精密動作に向けた課題は多い．後者の位置制御型では中枢全身制御系の指令通りの精密動作が再生されるが，着地衝撃緩和性能はフィードバック応答性の影響を大きく受ける．

関節にコンプライアンス制御を適用し，跳躍中は位置追従が支配的となる高剛性なゲイン，着地中は衝撃緩和に有利となる低剛性なゲイン，と設定する動的なゲイン変更を伴う全身動作指令による跳躍着地実験を試みる．400 ms の跳躍動作による実験の様子を図 6.22 に示す．高剛性な跳躍動作により地面からおよそ 20 mm 浮いた後，低剛性な着地動作で衝撃を緩和している．着地後に低剛性な高速動作に伴い姿勢が大きく変動している．更に高速な 300 ms の跳躍動作の場合では図 6.23 のように，関節の変位が過大となり，リンク間干渉により部品が破損してしまう失敗事例も見られた．

関節を位置制御型とする場合では中枢全身制御系が関節の運動指令を掌握するため，それぞれの関節変位が過大となってしまう可能性は小さく抑えられる．関節を位置制御型として，中枢全身制御系でエンドエフェクタの力センサ値を用いて着地動作指令を与えることによる跳躍着地実験の様子を図 6.24 に示す．250 ms の高速な跳躍動作によりおよそ

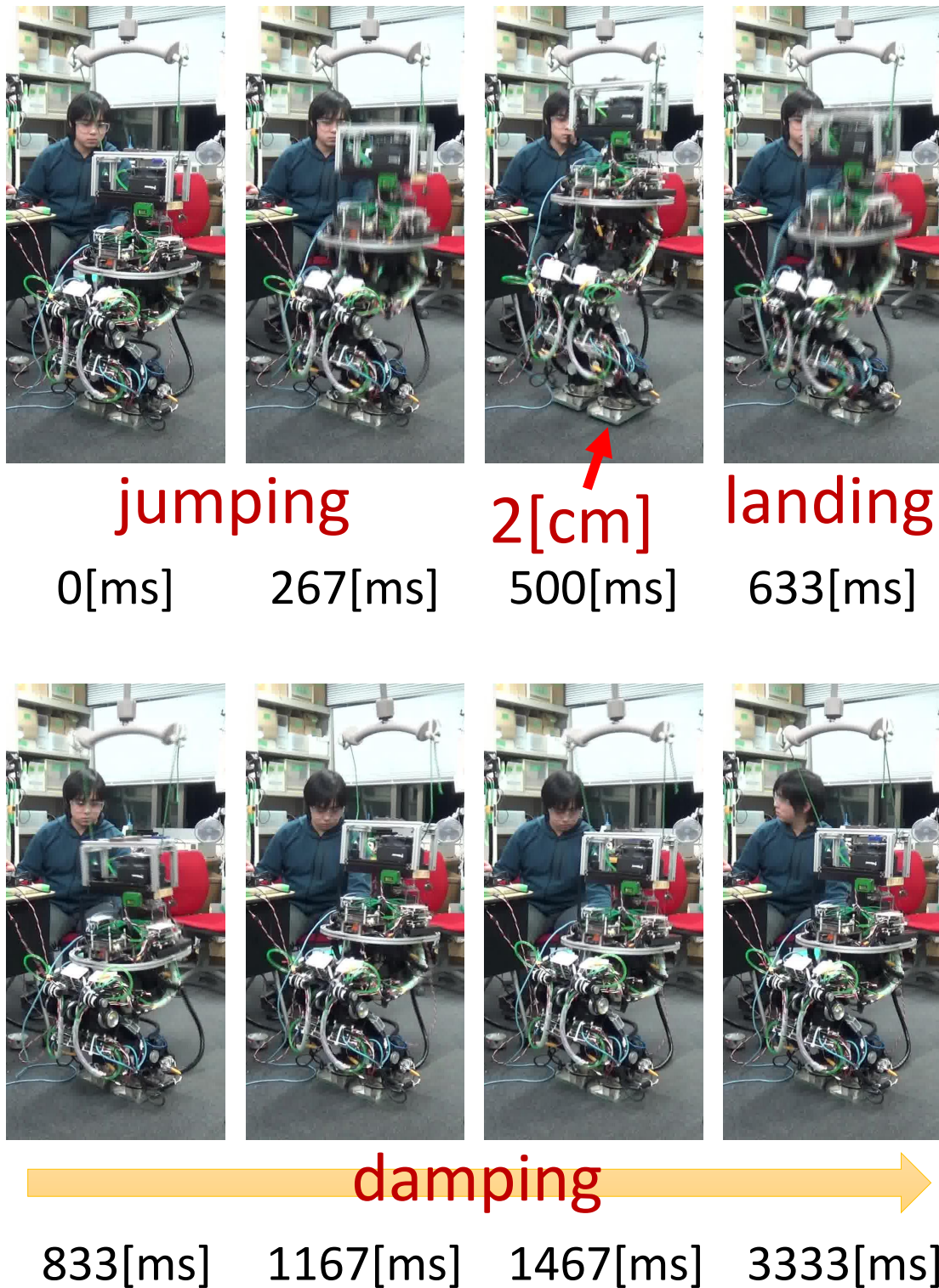


図 6.22 Preliminary experiment for jumping and landing by the high-power biped robot stretching in 400 ms

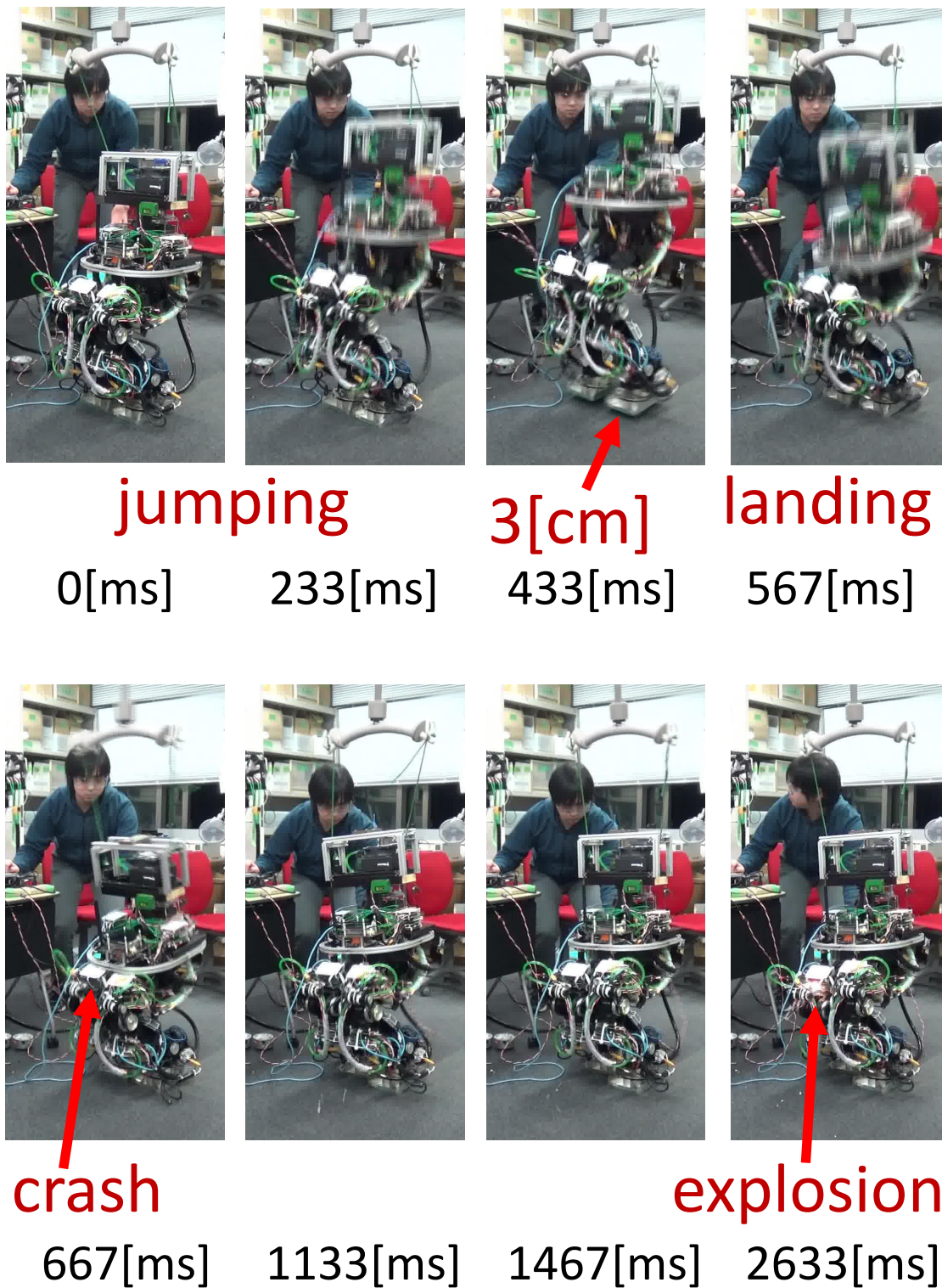


図 6.23 Preliminary experiment for jumping and landing by the high-power biped robot stretching in 300 ms, but failing into motor driver crash and explosion

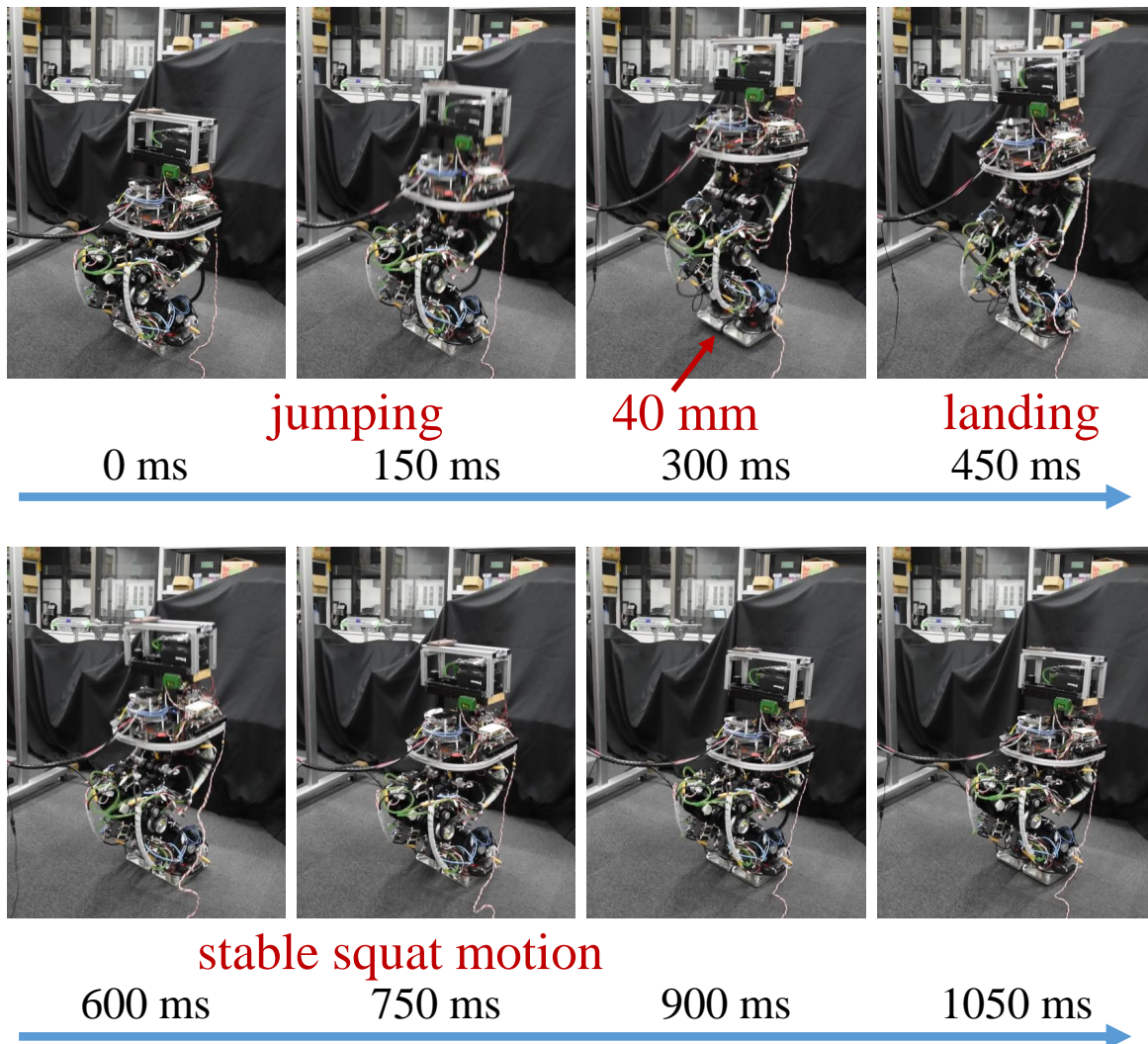


図 6.24 Jumping and landing experiment by the high-power biped robot stretching in 250 ms (position-controlled joints, squat motion control by whole-body force feedback)

40 mm 浮上して，地面と接した後に着地動作を行っている．それぞれの関節位置が大きく振動することはなく，速やかに屈曲姿勢に収束している．

位置制御型関節を用いた 250 ms の跳躍動作における関節位置・速度の軌道を図 6.25 に示す．最も大きく変動する膝関節の値をプロットしている．跳躍軌道は静止状態から開始して離地の瞬間に大きな速度を持つ躍度最小軌道として与える．着地軌道は逆に着地の瞬間に大きな速度を持つ躍度最小軌道とする．離地および着地の瞬間に速度指令値が不連続に変化するため，このときにオーバーシュートが見られるが，その後速やかに指令値に追従していることが確認できる．

関節を位置制御型として着地動作指令を中枢全身制御系による全身フィードバックに任

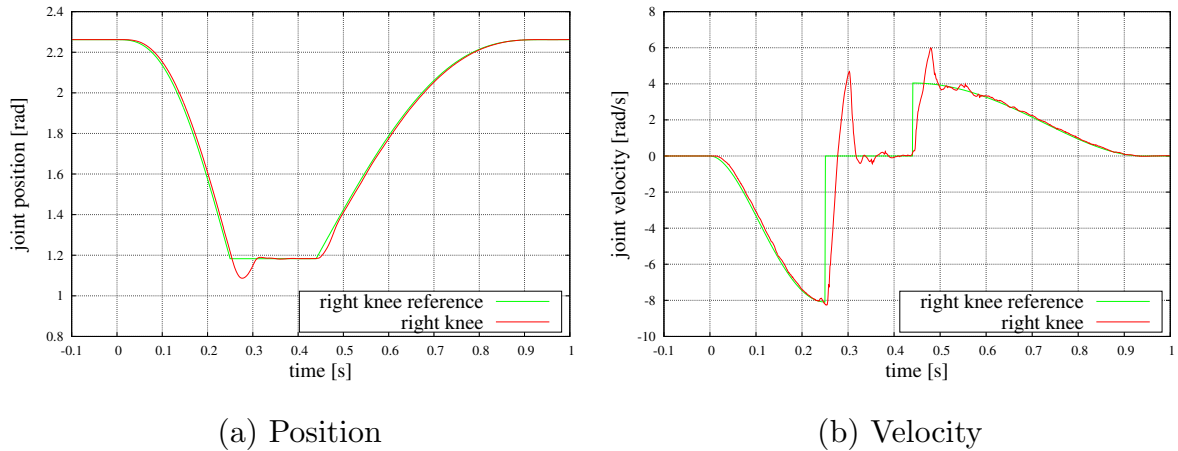


図 6.25 Joint reference trackability in jumping and landing experiment by the high-power biped robot stretching in 250 ms

せる手法では，全身で一貫した精密な着地動作が可能となる一方で，全身フィードバックの応答性が重要な鍵となる．僅かに浮上する 400 ms の跳躍動作において，力センサによる着地の検出から着地動作までに遅延を挿入して影響を調べる．中央計算機での 1 ms 周期の制御の中でセンサ値のフィードバックに設定した周期分の遅延を挟む．着地動作を行わない場合も合わせて比較する．表 6.1 の条件 (B0)-(B8) および条件 (C) に相当する条件で比較する．

それぞれの条件における着地時の接触力ピーク値を図 6.26 に示す．遅延を挟まない場合，接触力ピーク値は 560 N に抑えられている．遅延を大きくするに連れてピーク値は増大し，着地動作を行わない場合は力センサの定格 1000 N を超える力が発生している．

遅延を挟まない場合と着地動作を行わない場合の接触力波形を図 6.27 に比較する．図 6.27 (a) は遅延を挟まずに着地動作を行う場合の接触力波形である．図 6.27 (b) は跳躍動作の後に関節位置指令を固定して着地動作を行わない場合の接触力波形である．跳躍動作の開始時刻を 0 s とする．400 ms の跳躍動作の後に浮上しておよそ 50 ms の間 0 N に近い反力が観測される．着地しておよそ 30 ms で接触力がピークまで増大する．遅延を挟まない場合は小さな接触力に抑えられているのに対し，着地動作を行わない場合は大きな衝撃力が生じ，振動などの影響を受けて力センサが動作を一時停止する様子が見られた．柔軟な環境接触の実現はロボットが故障せず動作を継続する上で重要な要件となる．

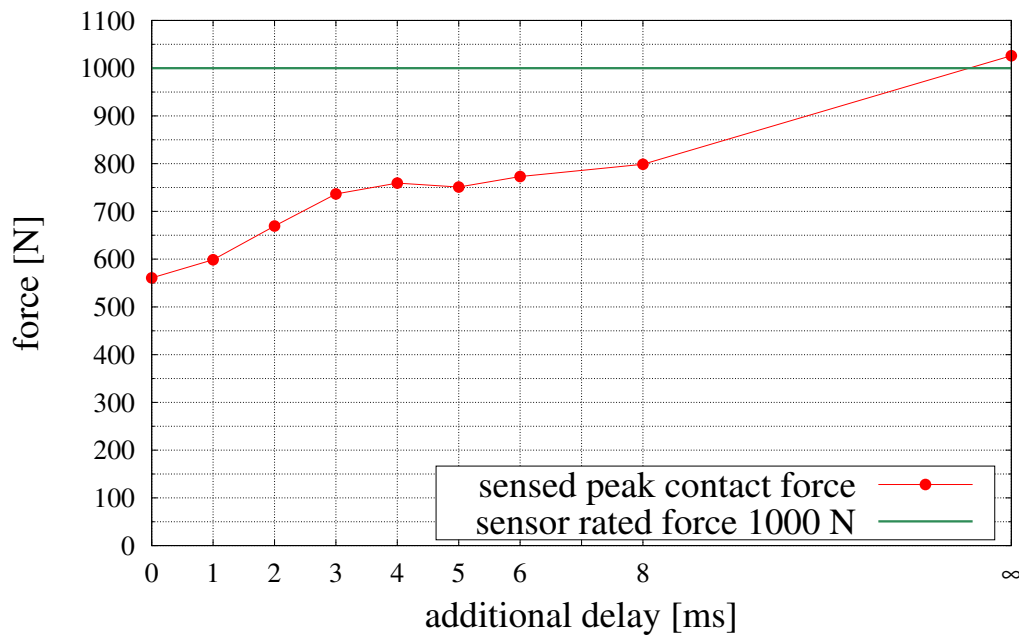


図 6.26 Sensed peak contact force of left ankle in jumping and landing experiment by the high-power biped robot stretching in 400 ms

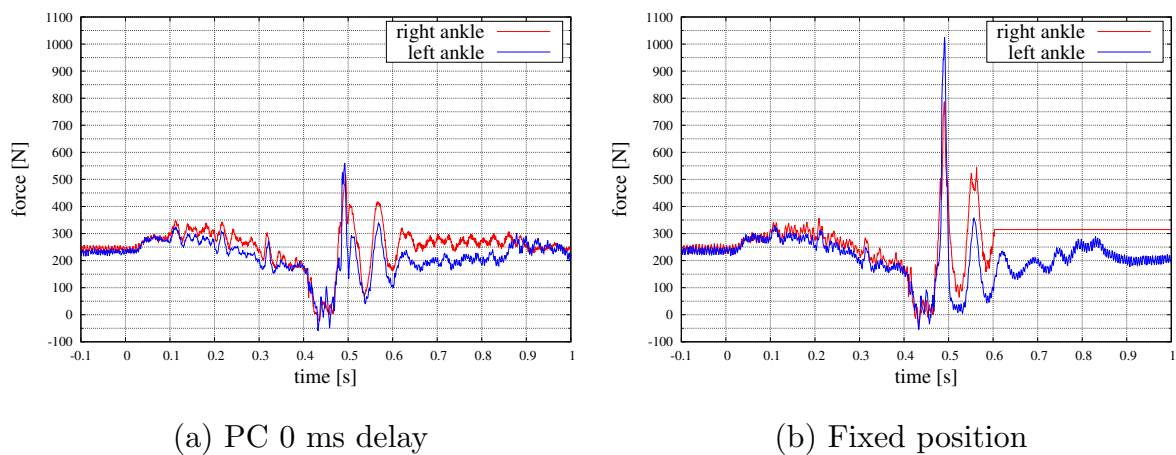


図 6.27 Sensed contact force in jumping and landing experiment by the high-power biped robot stretching in 400 ms

6.4 腱駆動型筋骨格ヒューマノイドの干渉駆動精密動作への応用

6.4.1 腱駆動型筋骨格ヒューマノイドへの応用戦略

腱駆動型筋骨格ヒューマノイドの場合は，柔軟な身体構造を冗長駆動するため特に多自由度なアクチュエータ制御が求められる．2.2.1 項で紹介したように，人間を模した等身

大の構造の中に 100 を超えるアクチュエータを搭載するロボットも存在する。

従来の腱駆動型筋骨格ヒューマノイドの体内制御システムには 3.3.2 項に述べたように、次のような課題が残っていた。

- MCU を利用した USB Full-Speed 実装により 8 ms に留まる通信周期の制約
- USB Full-Speed による 1 ノード 1 周期あたり送受信それぞれ 64 Byte のデータ転送量の制約
- 信号線を減らすための 3 Mbps RS-485 半二重汎用シリアル通信によるデータ転送速度の制約
- 小型モータ制御モジュールの計算能力制限に伴い中間制御基板によるフィードバックサーボ制御が別途必要
- 小型モータ制御モジュールの簡易電源電流制御による出力トルク精度と追従応答性の制約

これらの課題を解決するために、ネットワークの低遅延大容量化とモータ制御モジュールへの即応的サーボ制御系実装が求められる。本研究では課題を解決するとともに、ギア駆動型大出力ヒューマノイドとも共通する技術の広い適用を目指し、腱駆動型筋骨格ヒューマノイドへの体内分散実時間制御システムの段階的な適用を試みる。

USB Full-Speed による通信周期およびデータ転送量の制約を改善するために、これを FPGA 制御による USB High-Speed 通信で置換する。ヒューマノイド腱悟郎における適用イメージを図 6.28 に示す。USB High-Speed 中間制御基板および USB 2.0 ハブ基板を用いることで、データ転送レート 480 Mbps、最短通信周期 125 μ s、最大データ転送量 512 Byte の通信が実現される。従来は最短でも 8 ms となっていた全身制御周期が 1 ms 以下まで短縮され、高追従なダイナミック動作の実現可能性が開かれる。また従来は 1 ノードの中間制御基板あたりの送受信データが最大 64 Byte となっており、1 ノードで最大 4 個のモータを制御するためモータあたり 16 Byte のおよそ必要最小限の送受信データに限られていた。一方 USB High-Speed による実装では 1 ノードあたり最大 512 Byte まで拡張され、多くの補助データやセンサデータを交換することが可能となる。USB High-Speed 中間制御基板には IMU が搭載されており、モータ制御データと同時に腱駆動型筋骨格ヒューマノイドの柔軟身体に分散配置された基板の姿勢が取得可能となり、応用的なロボット身体学習研究への展開も期待される。

一般化デバイス制御プラットフォームにおいて、USB 通信および腱駆動型筋骨格ヒューマノイドの制御インタフェースに向けた拡張実装を施す。これにより、Ethernet インタフェースで制御されるギア型駆動大出力ヒューマノイドなど、他のロボットと共通構造で、中枢全身制御系における実時間制御が達成される。USB High-Speed 実装との組み合わせにより、1 ms 以下の制御周期による低遅延な全身フィードバック動作制御が可能と

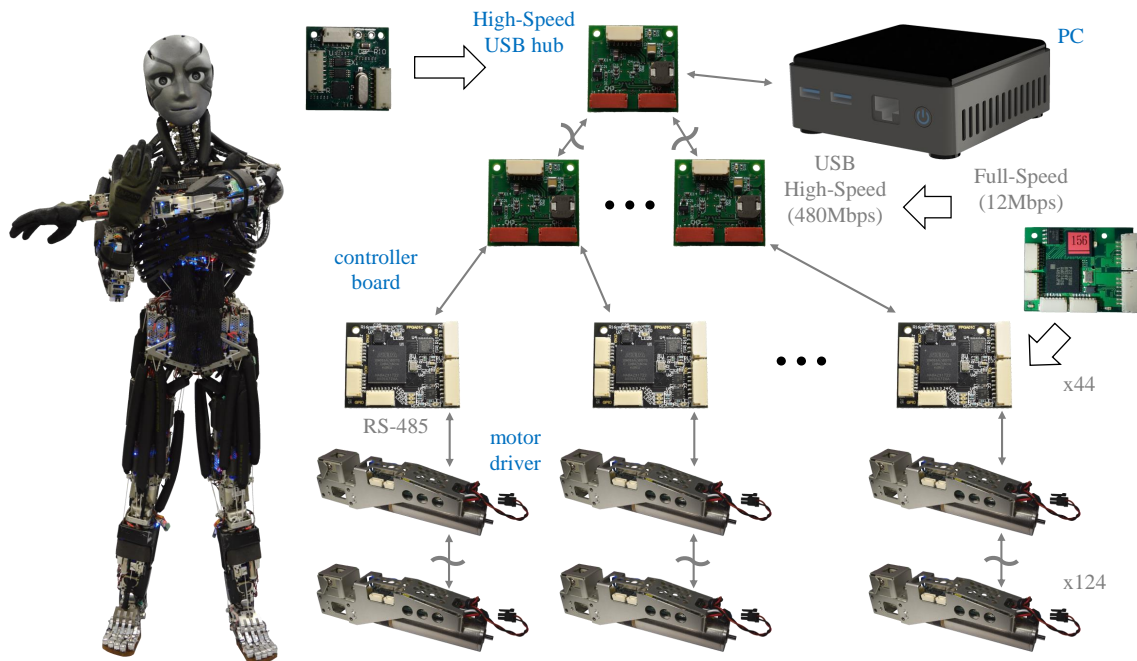


図 6.28 Example of USB High-Speed network with the humanoid robot Kengoro

なる。

更に本研究で新たに開発した小型高速通信モータ制御モジュールを利用することで省スペース性と制御性能の両立が可能となる。モータ制御モジュールごとの高精度で応答性の高い電流制御およびサーボ制御が可能となる。高速光通信や RS-422 全二重通信による分散ネットワーク、および Ethernet 基幹通信ネットワークを適用することでギア駆動型大出力ヒューマノイドと同様構造の低遅延ネットワークが構成される。

6.4.2 低遅延 USB 通信ネットワークの適用

USB High-Speed ネットワークを腱駆動型筋骨格ヒューマノイドに適用し評価する。図 6.29 の脚部ロボット MusashiOLegs [181] において、USB High-Speed 中間制御基板および USB 2.0 ハブ基板によるネットワークを実装する。両脚および腰部に 5 個の直交関節モジュールを搭載し、13 の関節自由度を有している。これを 40 のモータで駆動する構成となっている。

全身制御を担う中央計算機として Intel 社の NUC 7 Mainstream Kit NUC7i7BNH を利用する。中央計算機に対して、11 台の USB High-Speed 中間制御基板、10 台の USB 2.0 ハブ基板、および 40 台のモータ制御モジュールから構成される分散制御系を接続す

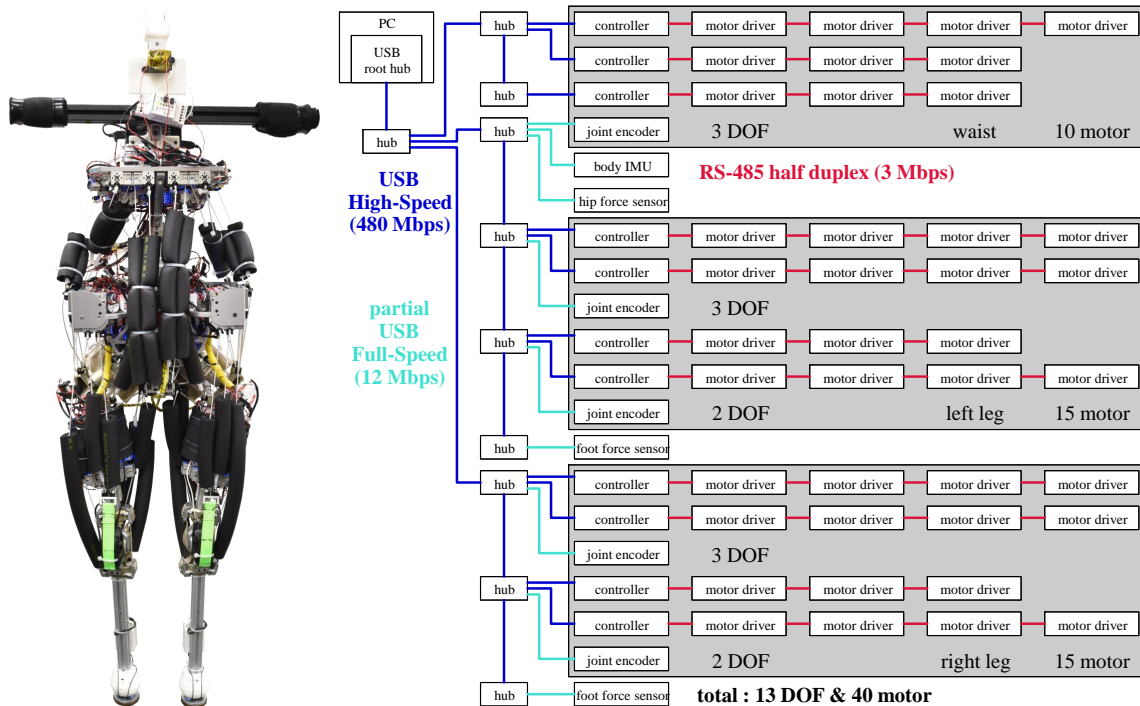


図 6.29 USB High-Speed network for the biped robot MusashiOLegs

る．USB ネットワークにはこれに加えて，関節に搭載されたポテンショメータ，体幹の IMU，腰部の接触力センサ，および両足部の力センサが接続され，中央計算機でそれぞれのセンサ値を取得する．これらのセンサノードは 12 Mbps の USB Full-Speed で USB 2.0 ハブ基板に接続される．

FPGA を利用した USB High-Speed 実装によるロボット全身制御の低遅延化効果を検証する．MusashiOLegs に搭載された 11 台の USB High-Speed 中間制御基板に，

- 従来の MCU 実装で限界となっていた 8 ms 周期の USB Full-Speed 通信
- FPGA により可能となる最短周期 1 ms の USB Full-Speed 通信
- 更に USB 2.0 完全準拠 PHY により可能となる最短周期 250 μ s の USB High-Speed 通信

のそれぞれを実装する．なお USB High-Speed 実装で利用可能なインタラプト転送の最短周期は表 3.8 の通り 125 μ s であるが，MusashiOLegs の USB ノード構成に対する中央計算機側の帯域制限により有効な設定値は最短で 250 μ s となった．

中央計算機で全身のデータを収集してそれに応じた制御指令を送信する制御ループを，アプリケーションソフトウェアによる待機時間を加えず最速で実行した際の平均実行時間を測定する．これがロボット全身実時間制御における実用上の最短制御周期となる．中央

表 6.5 Communication data item list for the biped robot MusashiOLegs

data	Read/Write	size
motor position	R	2 Byte
motor position reference readback	R	2 Byte
wire tension	R	2 Byte
motor temperature	R	2 Byte
motor power voltage	R	2 Byte
controller power voltage	R	2 Byte
motor position reference	W	2 Byte
wire tension reference	W	2 Byte
control mode	W	1 Byte

計算機では Ubuntu 16.04.6 LTS / Linux 4.15.0-142-generic の OS 上で libusb-1.0 を利用してソフトウェアを実装する．制御周期ごとに表 3.1 のデータをアクチュエータの数だけ中間制御基板と交換する．アクチュエータあたり 12 Byte のデータを読み出し，5 Byte のデータを書き込む．交換するデータの種別は初期化時に予め設定しておき，制御周期ごとの転送データにはヘッダなどアプリケーション上の付加データは求められない．読み出しにはインタラプト転送，書き込みにはバルク転送を利用する．1 ノードの中間制御基板には図 6.29 の通り 3 台もしくは 4 台のアクチュエータ駆動モジュールが接続され，4 台の場合は計 48 Byte の読み出しと計 20 Byte の書き込みが実行される．

それぞれの USB 通信プロトコル条件において，11 ノードの中間制御基板のうち指定したノード数のみ有効化した場合の平均実行時間を図 6.30 に示す．それぞれの条件でおよそ 20 s の間実行時間のサンプルを取得し統計処理を施している．図 6.29 で上部に記載したのから順に中間制御基板，すなわち図で controller と記載されたノードを有効化している．中間制御基板以外のセンサノードのデータは同期スレッドの制御ソフトウェアからは読み出しておらず，これは中間制御基板のみとの通信を含む実験結果である．

従来の 8 ms 周期転送による実装では，USB 通信に合わせて制御ループもおおよそ 8 ms 周期で実行されている．USB 通信の周期転送に任せるのみで低ジッタな周期制御が実装される反面，これが低遅延化の律速となっている．同じ USB Full-Speed で 1 ms 周期転送とした場合，および USB High-Speed で 250 μ s 周期転送とした場合は複数の周期に渡り全 11 ノードとデータを交換している．原則として 1 度の実行時間はおよそ USB 通信周期の倍数となり，ノード数が増えるに連れて必要な平均周期が増加する．全 11 ノードとの通信を含めた平均実行時間は，USB Full-Speed では 2069 μ s，USB High-Speed で

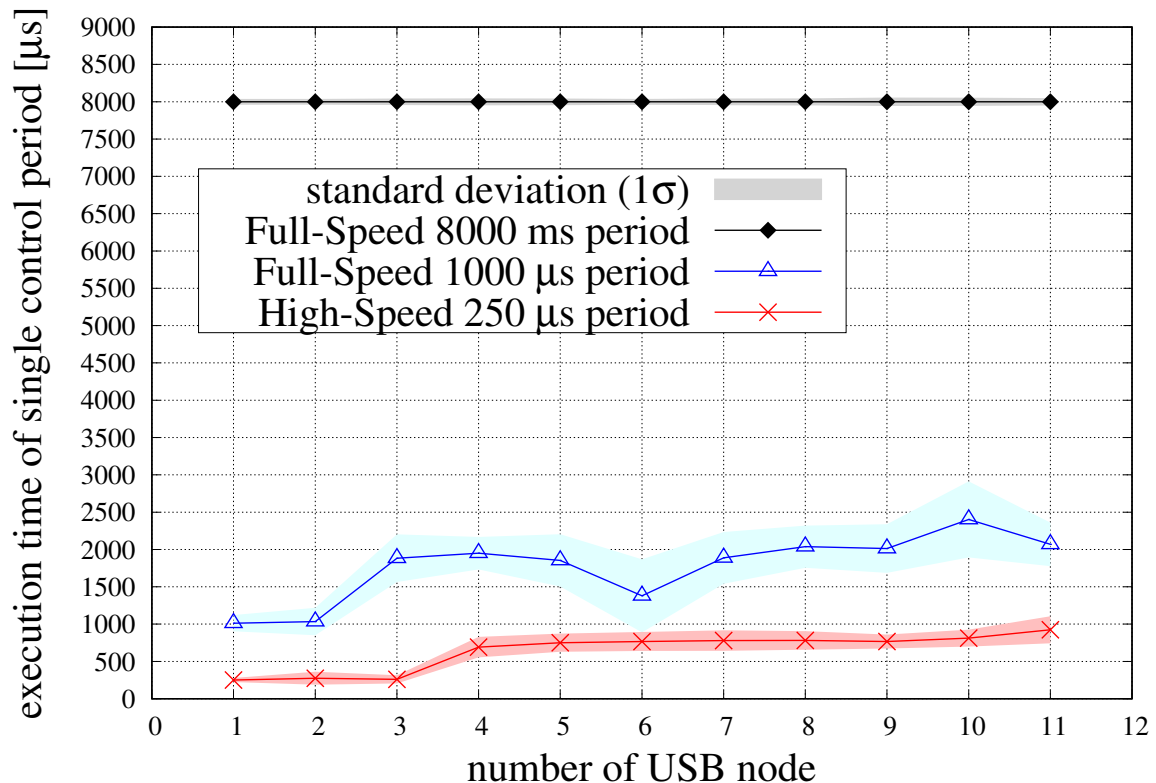


図 6.30 Minimum execution time of control routine determined by USB transfer time for the biped robot MusashiOLegs

は $924 \mu\text{s}$ となっている。

USB High-Speed を利用することで、全身に 40 のモータを搭載した MusashiOLegs において、従来の 8 分の 1 に相当する 1 ms 以下の周期での制御が実現される。

480 Mbps の USB High-Speed では同じ物理層ハードウェアを利用した 12 Mbps の USB Full-Speed と比べてロボット体内で運用する場合の耐ノイズ性が低下するため、USB Full-Speed のままでの中間的な低遅延化が求められる場合もある。FPGA を活用した実装により、USB Full-Speed の場合でもおよそ 2 ms の周期で制御することが可能となる。

MusashiOLegs に適用した USB ネットワークにおけるロボット体内ノイズの影響を確認する。ロボット体内の多数のモータをスイッチング制御により駆動すると、制御回路系には大きくかつ変動的な電流が発生する。電源の負荷変動の影響、および空間に流れる大電流による電磁ノイズの影響により、ケーブル配線された電源や通信ネットワークには電圧変動が生じる。

全身のモータを駆動しているときとサーボオフしているときの制御系電源の電圧波形を測定する。測定には Tektronix 社のオシロスコープ MDO4054B-3 および Tektronix

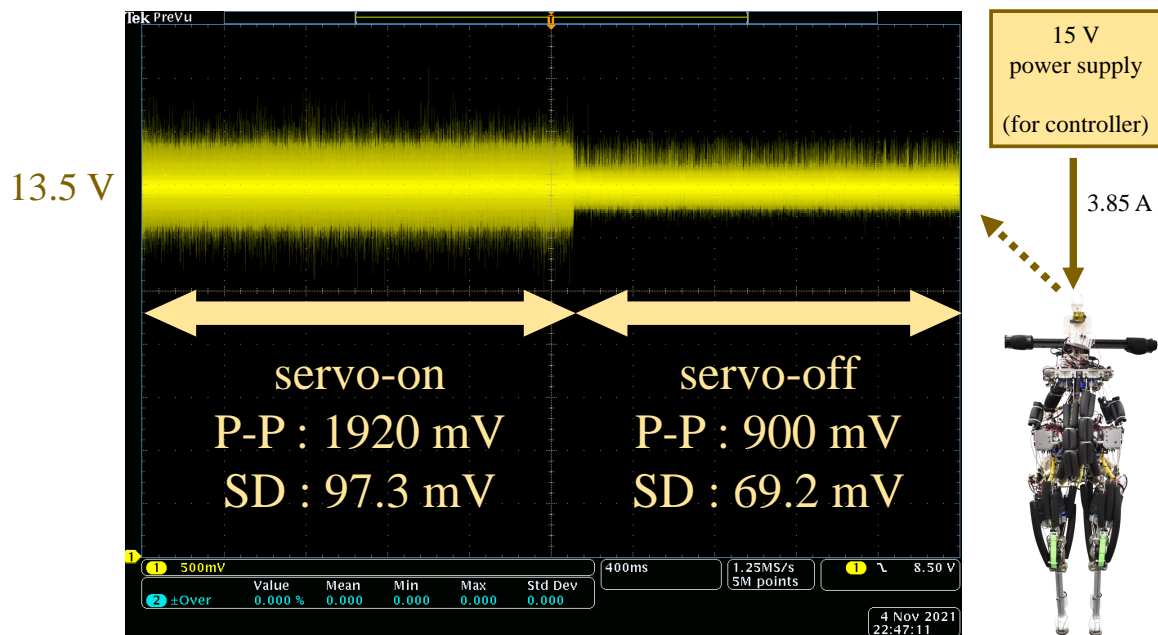


図 6.31 Measured voltage noise of controller power for the biped robot Musashi-iOLegs with motor output ON/OFF

社の受動電圧プローブ TPP0500B を用いる．図 6.31 に測定結果を示す．制御系電源は USB 2.0 ハブ基板に入力されて，それぞれの USB 接続機器へ電力を供給している．外部の電源装置より 15 V の電圧で入力され，配線上の電圧降下によりロボット体内ではおよそ 13.5 V となっている．モータを駆動しているときは 1.92 V 幅のノイズが発生しており，13.5 V の電圧に対して 14.2 % の値となっている．サーボオフ状態と比べると 2.13 倍のノイズが発生している．

同様に USB 通信の信号波形を測定する．12 Mbps の USB Full-Speed を実装している場合の信号波形の例を図 6.32 に示す．480 Mbps の USB High-Speed を実装している場合の信号波形の例を図 6.33 に示す．全身のモータを駆動しているときとサーボオフしているときで 1 パケット分の信号波形を比較している．GND に対する USB 通信の差動信号電圧を測定し，それぞれの信号線電圧に加えて差動電圧とコモンモード中間電圧を表示している．差動電圧およびコモンモード中間電圧には，USB 2.0 PHY として利用している Microchip Technology 社の USB3300-EZK の適切な電圧範囲の表示を添えている．

USB Full-Speed の場合はおよそ 3 V 振幅の信号で通信している．図 6.32 (a1) の例ではモータ駆動中に 0.8 V 幅のノイズが発生しているが通信に対して問題を生じる程度とはなっていない．一方で，USB High-Speed の場合は通信信号の振幅がおよそ 0.4 V であり，モータ駆動中のノイズが無視できないものとなる．ツイストペアケーブルや同

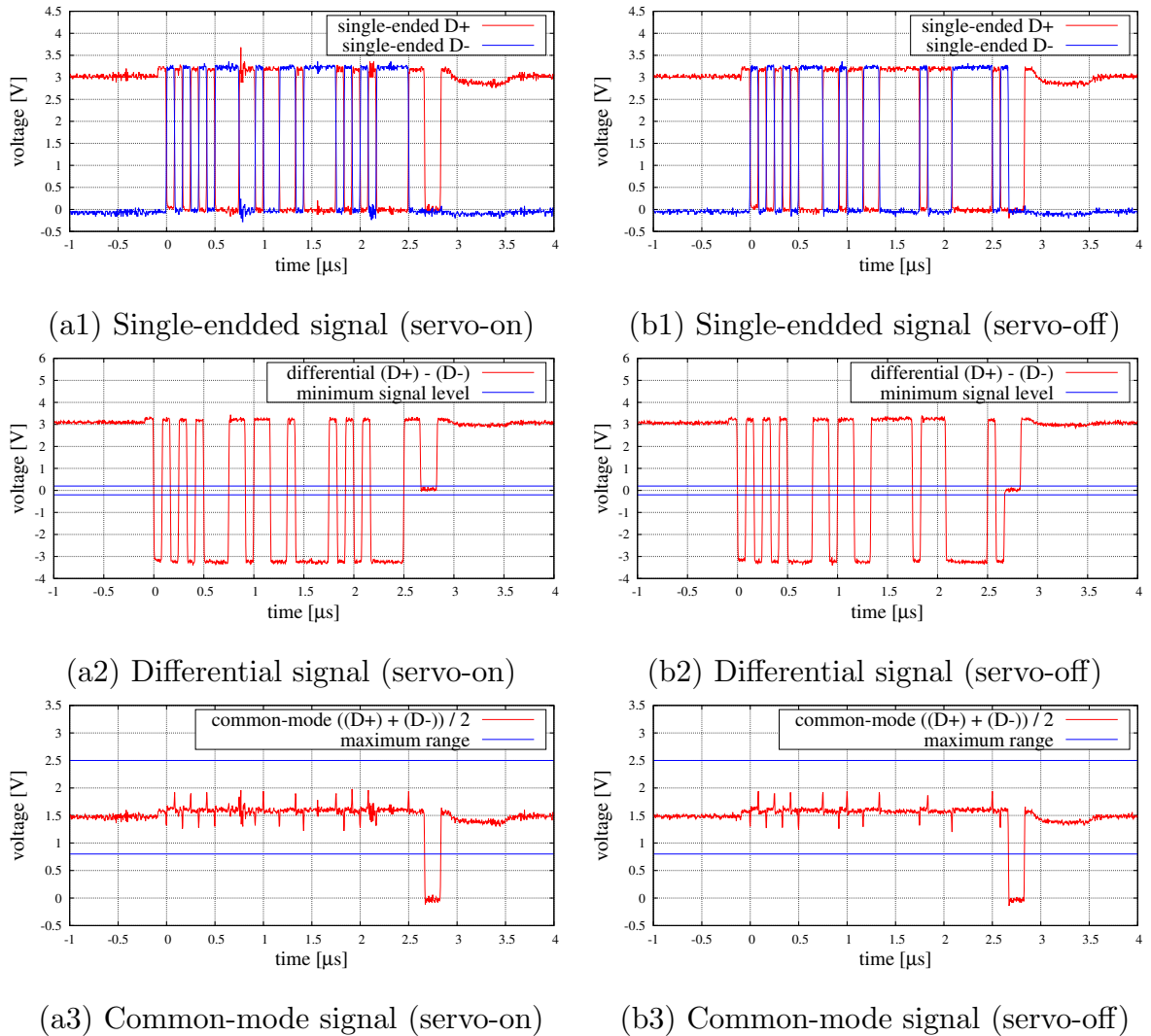


図 6.32 Measured USB Full-Speed signal waveform on the biped robot Musashi-iOLegs with motor output ON/OFF

軸ケーブルを用いて信号の差動電圧に対するノイズの影響を小さくすることが重要となる。MusashiOLegs ではツイストペアケーブルにより USB ネットワーク配線を施しており、図 6.33 (a2) の例では差動電圧への影響が小さく抑えられていることが確認できる。図 6.33 (a3) ではコモンモード中間電圧に発生したノイズが USB 2.0 PHY の適切範囲を超えている。コモンモードノイズが問題となる場合はフェライトコアなどのフィルタによりノイズの影響を緩和できるが、同時にフィルタによる信号品質への影響も発生し、問題が解決できない可能性も想定される。このような状況では、USB Full-Speed の FPGA 実装による中間的な低遅延化が効果を発揮する他、光通信を導入して電磁ノイズの影響を受けにくい高速ネットワークを構成することが有効となる。

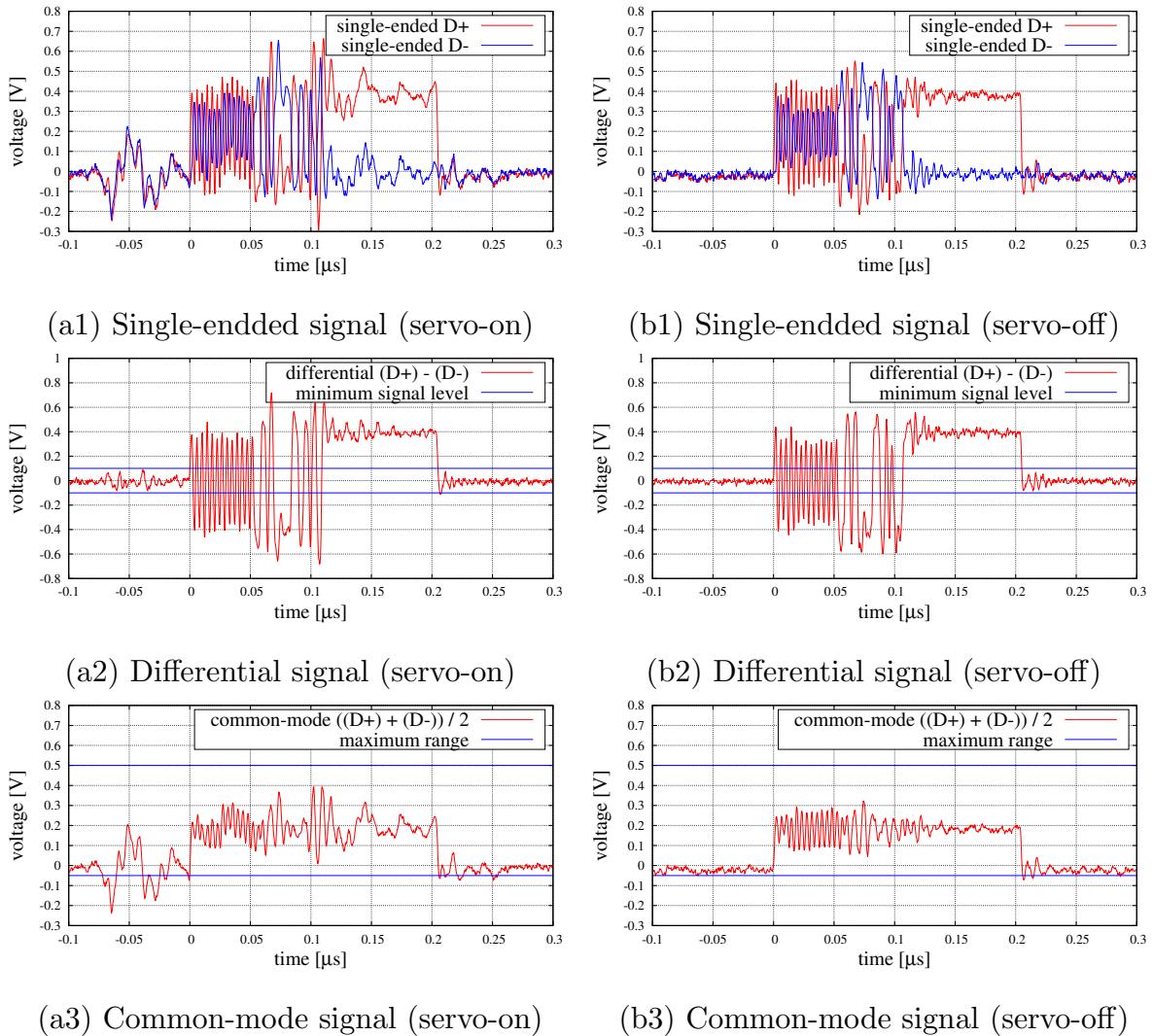


図 6.33 Measured USB High-Speed signal waveform on the biped robot Musashi-iOLegs with motor output ON/OFF

6.4.3 高信頼な小型高速通信モータ制御モジュールの適用

FPGA を利用した USB High-Speed 通信によりネットワークを低遅延化する手法で、制御周期の短縮と大容量データ転送が可能となった。非常に多自由度な腱駆動型筋骨格ヒューマノイドにおいて、1 ms 以下の制御周期で 1 ノードあたり最大 512 Byte の多くのデータを交換する全身制御システムが現実のものとなった。これにより、従来 8 ms 周期で少数の限られたデータのみを取り扱っていた全身制御システムと比べて、より複雑な現象を取り扱うことが可能となる。しかし環境接触などの外乱に対する応答性の観点からは、従来のモータ制御モジュールにおけるホールセンサの位置分解能や簡易電流制御の応

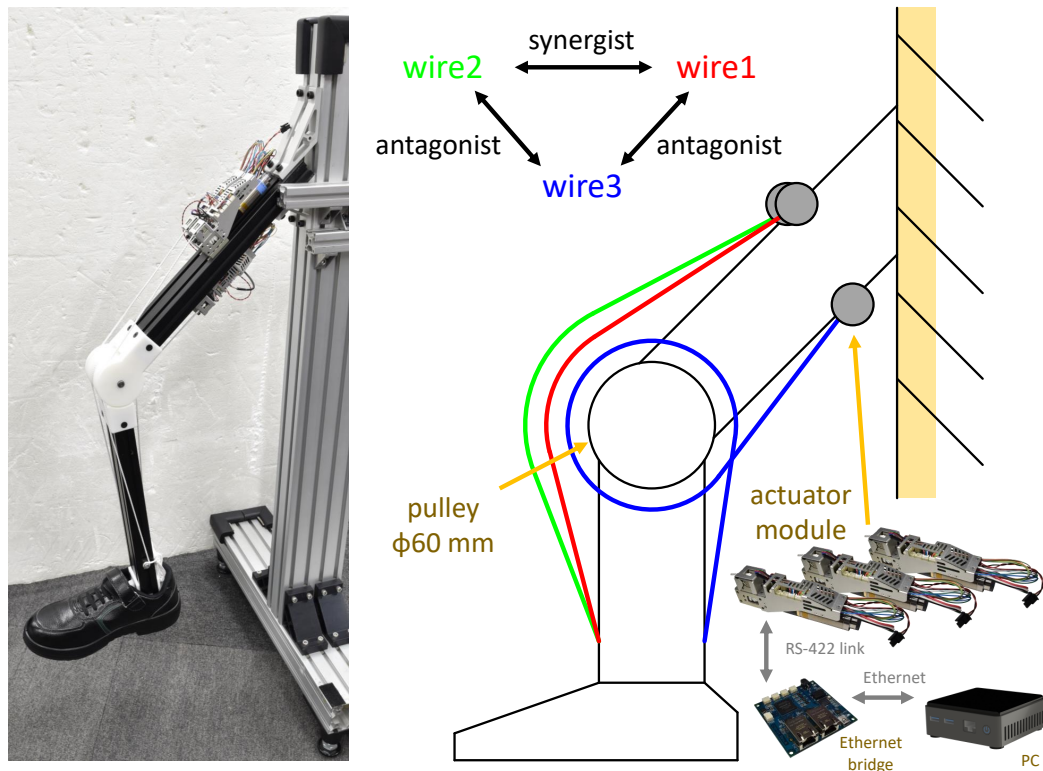


図 6.34 Experimental setup overview of the knee robot

答性が律速となりフィードバック制御性能向上には限りがある．また，USB High-Speed では電氣的な信号規格や厳密な接続に基づくプロトコルから，厳しいロボット体内ノイズ下での実用性にもやや難点がある．

3.5.2 項の腱駆動向け小型モータ制御モジュールを用いることで，サーボ制御の応答性および通信の信頼性を向上した体内分散ネットワークが構成可能となる．図 6.34 の膝部ロボットに小型モータ制御モジュールを適用して低遅延フィードバックの効果を確認する．膝部ロボットは 1 自由度の膝関節を有し，これを 3 本のワイヤで駆動する．前部の大腿四頭筋・膝蓋腱に相当する 2 本のワイヤと，後部のハムストリングスに相当する 1 本のワイヤで構成される．前部のワイヤ同士は共同筋の関係にある．前部のワイヤと後部のワイヤは拮抗筋の関係にある．それぞれのワイヤは関節部のプーリを経て 2 リンクに接続し張力を発生させる．ワイヤを駆動するモータ制御モジュールは RS-422 汎用シリアル通信ネットワークを構成し，Ethernet 中継基板によって中央計算機と接続する．

それぞれのモータ制御モジュールではモータの回転をギアボックスで減速してウィンチを駆動する．ウィンチの回転によりワイヤが巻取られて筋長が変化し，関節が駆動される．単純なモデル化に基づけば，ウィンチの巻取り半径と関節のプーリ半径の比が減速比となる．そのため，モータの回転と関節の回転を対応付けるヤコビ行列は関節位置によら

ず一定であり，目的の関節運動に対応するモータ位置軌道は容易に計画可能である．モータ位置を $\theta = (\theta_1, \theta_2, \theta_3)^T$ ，ギアボックス減速比を R ，ウィンチ半径を r_w ，関節プーリ半径を r_j ，関節位置を q とおく．関節運動に対応するモータ速度から，ヤコビ行列 J の疑似逆行列 J^+ は，

$$\dot{\theta} = J^+ \dot{q} \quad (6.1)$$

$$J^+ = \begin{pmatrix} \frac{Rr_j}{r_w} \\ \frac{r_w}{Rr_j} \\ -\frac{r_w}{Rr_j} \end{pmatrix} \quad (6.2)$$

とモデル化できる．Moore-Penrose 疑似逆行列によりヤコビ行列は，

$$\dot{q} = J\dot{\theta} \quad (6.3)$$

$$J = (J^+)^+ \quad (6.4)$$

$$\begin{aligned} &= \left((J^+)^* J^+ \right)^{-1} (J^+)^* \\ &= \begin{pmatrix} \frac{r_w}{3Rr_j} & \frac{r_w}{3Rr_j} & -\frac{r_w}{3Rr_j} \end{pmatrix} \end{aligned}$$

となる．

フィードフォワードなモータ位置軌道は容易に求められる一方で，実際には，

- ワイヤが巻き取られて変化するウィンチの巻取り半径
- ワイヤ経路のずれ
- 弾性や摩擦に基づく不規則なワイヤ長変化

などの誤差により，フィードフォワードなモータ位置指令に追従するのみではワイヤ間の均衡を保つことができない．目的の関節運動に追従しながら，ワイヤ張力をフィードバックして適切に分配することが求められる．

ワイヤ張力を $f = (f_1, f_2, f_3)^T$ とする．モータトルク $\tau = (\tau_1, \tau_2, \tau_3)^T$ との関係は，摩擦・損失や寸法誤差の影響を無視するとおよそ $f = -\frac{R}{r_w}\tau$ である．なおここでモータ位置 θ およびモータトルク τ はワイヤを送り出して緩める方向を正と定義している．フィードフォワードなモータ位置指令に対して，次の3種類のワイヤ張力フィードバック補償を施して合理的なワイヤ張力分配を実現する．

- 合計張力 $f_1 + f_2 + f_3$ を目標値に追従させることで剛性を制御
- 共同筋の張力 f_1 と f_2 を等しくすることで共同筋張力分配を最適化
- 拮抗筋の張力 $f_1 + f_2$ と f_3 に最低値を設けることでワイヤの緩みを防止

PID 制御によりそれぞれの補償量を決定する．合計張力の PID 制御出力を $\Delta\theta[f_1 + f_2 + f_3]$ とする．合計張力 $f_1 + f_2 + f_3$ を目標値 f_{ref} に追従させる．共

同筋張力の PID 制御出力を $\Delta\theta[f_1 - f_2]$ とする．共同筋張力差 $f_1 - f_2$ を 0 に追従させる．拮抗筋の最低張力には符号で場合分けした不連続な 2 入力積分制御を施し，その出力を $\Delta\theta[f_1 + f_2, f_3]$ とする．拮抗筋の張力 $f_1 + f_2$ および f_3 をそれぞれ目標最低値 f_{\min} と比較し， $f_{\min} - (f_1 + f_2) > 0$ もしくは $\Delta\theta[f_1 + f_2, f_3] > 0$ であれば $f_{\min} - (f_1 + f_2)$ を正のゲインで積分， $f_{\min} - f_3 > 0$ もしくは $\Delta\theta[f_1 + f_2, f_3] < 0$ であれば $f_{\min} - f_3$ を負のゲインで積分する．これらの制御出力を用いて，フィードフォワードなモータ回転指令に加算する補償量 $\Delta\theta = (\Delta\theta_1, \Delta\theta_2, \Delta\theta_3)^T$ を，

$$\begin{aligned} \Delta\theta = & \Delta\theta[f_1 + f_2 + f_3] \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} \\ & + \Delta\theta[f_1 - f_2] \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} + \Delta\theta[f_1 + f_2, f_3] \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} \end{aligned} \quad (6.5)$$

と決定する．

3 種類のワイヤ張力分配制御はいずれも複数のモータ制御モジュールの情報を用いるため，全身フィードバック制御として実装される．PID 制御ゲインを大きく設定すると追従性が向上する．大きなゲインで安定した制御を実現するために，全身フィードバック制御の低遅延化が重要である．

中央計算機として，Intel 社の CPU Core i7-10 を搭載した Lenovo 社のラップトップ PC T490 および Ubuntu 20.04.3 LTS / Linux 5.11.0-43-generic の OS を利用して実験する．ワイヤ張力分配制御を中央計算機で $500 \mu\text{s}$ 周期で実行する．中央計算機における $500 \mu\text{s}$ 周期の制御を安定して実行できる表 6.6 のパラメータを用いる．

ワイヤ張力分配制御を有効化した際のモータ位置補償量およびワイヤ張力の推移例を図 6.35 に示す．膝部ロボットの下腿リンクを自然に垂れ下げてワイヤをやや緩めた状態を模擬的なフィードフォワード位置指令として与える．時刻 0 でワイヤ張力分配制御を有効化している．ワイヤを緩めたおよそ張力 0 N の状態からワイヤ張力分配制御を有効化すると，初期段階では主に合計張力制御出力によってそれぞれのワイヤが巻き取られる．ワイヤに張力が生じ始めると，主に拮抗筋張力分配制御出力によってそれぞれのワイヤの巻き取り量に差が生じ，適切な張力分配に収束する．この例ではワイヤ張力分配制御を有効化してからおよそ 300 ms で安定に収束している．全ワイヤの合計張力は目標値の 300 N に追従している．下腿リンクを自然に垂れ下げた姿勢付近では拮抗筋の張力がおよそ等しくなり， $\text{wire1} \cdot \text{wire2}$ の合計張力と wire3 の張力が同程度の値となっている．

ワイヤ張力フィードバックにより大きな遅延が存在すると制御は不安定となる．制御周期を $500 \mu\text{s}$ に保った状態でワイヤ張力値に遅延を挿入して確認する．およそ 5 s 間隔でソフトウェア上のフィードバック遅延を 1 周期分ずつ増大させた際のワイヤ張力値を

表 6.6 Wire tension distribution control parameter for the knee robot

control	parameter	value for motor angle θ	converted value for joint angle q
total tension $\Delta\theta [f_1 + f_2 + f_3]$	reference f_{ref}	300 N	-
	P-gain	0.00182 rad/N	0.00001 rad/N
	I-gain	1.82 rad/Ns	0.01 rad/Ns
	D-gain	0 rad/(N/s)	0 rad/(N/s)
synergist distirbution $\Delta\theta [f_1 - f_2]$	P-gain	0.00182 rad/N	0.00001 rad/N
	I-gain	1.82 rad/Ns	0.001 rad/Ns
	D-gain	0 rad/(N/s)	0 rad/(N/s)
minimum tension $\Delta\theta [f_1 + f_2, f_3]$	reference f_{min}	80 N	-
	I-gain	2.18 rad/Ns	0.012 rad/Ns

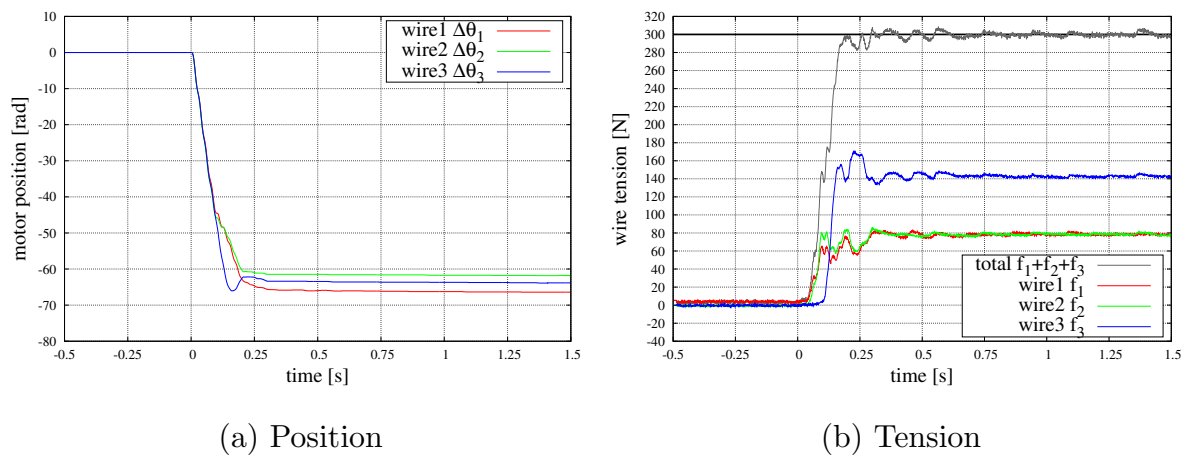


図 6.35 Transition of motor position and wire tension after enabling wire tension distribution control of the knee robot

図 6.36 に示す．遅延を挿入しない状態から開始して 16 周期すなわち 8 ms 分の遅延までをプロットしている．遅延を挿入すると制御が不安定となり振動が生じている．遅延が増大するにつれて振動も大きくなっている．8 ms の遅延が存在する場合，300 N の目標合計張力に対して 300 N を超える振幅が観測されている．

大きな遅延が生じて制御が不安定となる場合は，制御ゲインを小さくする必要性が生じる．500 μs 周期で 8 ms の遅延を挿入した場合，および制御周期を 8 ms とした場合は，ともに同程度の安定性を示し，500 μs 周期で遅延を挿入しない場合と比べてそれぞれの積分ゲインをおよそ 0.3 倍に設定すると安定となった．このとき追従性もおよそ 0.3 倍と

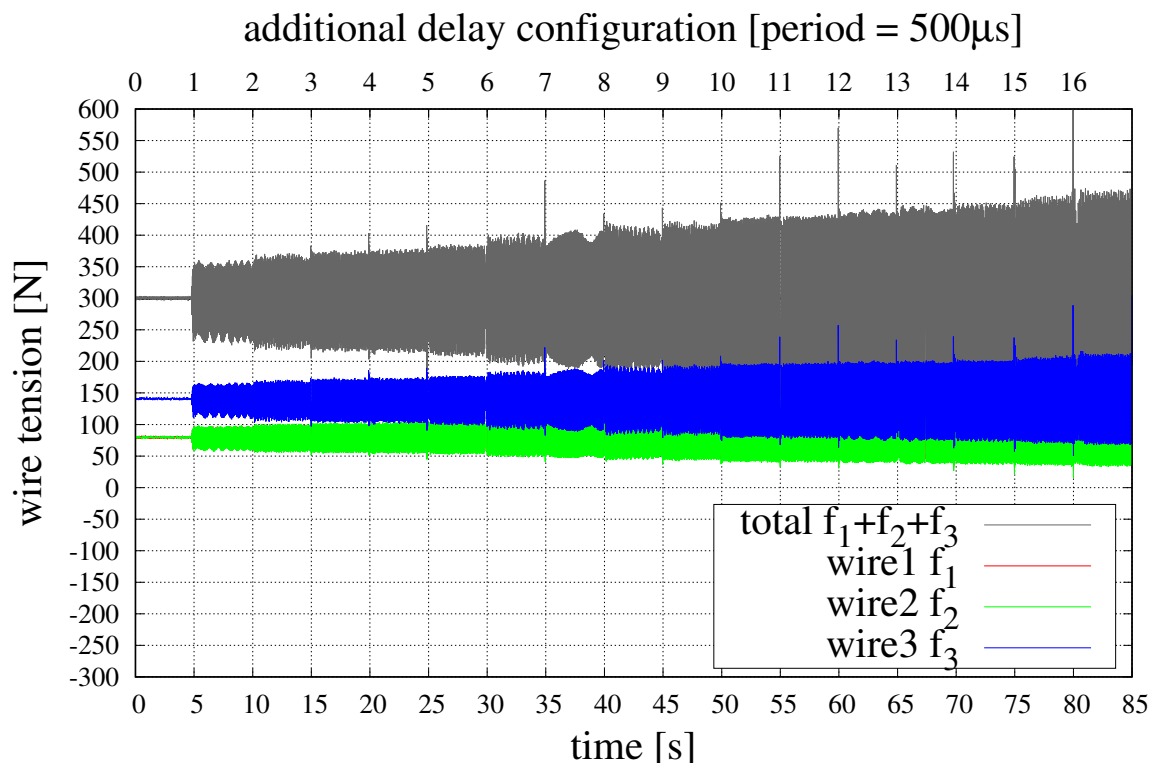


図 6.36 Unstable vibration corresponding to configured additional feedback delay in wire tension distribution control of the knee robot

なり，すなわち収束までに 3.3 倍の時間を要することになる．図 6.35 から，500 μ s 周期で遅延を挿入しない場合，指令値への追従はおよそ 100 ms 程度の時定数で近似できる．これが 3.3 倍になると俊敏な動作において大きな影響が生じてしまうことから，低遅延化の重要性が確認できる．

高ゲインなワイヤ張力フィードバック制御により，適切な張力に素早く追従しながら高速に物体を打撃するような作業が可能となる．膝部ロボットでいくつかの物体を蹴り飛ばす実験の様子を図 6.37 に示す．下腿リンクが鉛直となる姿勢から前後に $\pm \frac{\pi}{6}$ rad の範囲を 300 ms で移動する躍度最小関節位置軌道をフィードフォワードな指令として与える．鉛直よりやや手前の関節位置で物体と接触する．動画の記録では蹴り出し動作を開始してからおよそ 133 ms で物体と接触している．

図 6.37 (a) ではフットボールを用いている．高速な蹴り出し動作によって軽量のフットボールを大きく飛ばしている．図 6.37 (b) では 3.6 kg のダンベルを用いている．比較的軽量のダンベルであり，300 N の目標合計張力を設定したワイヤ張力分配制御によってしなやかに転がされている．図 6.37 (c) では合計 13.6 kg のダンベルを用いている．転がり出さない程度の重量を持つダンベルとしなやかに衝突して姿勢を保持している．

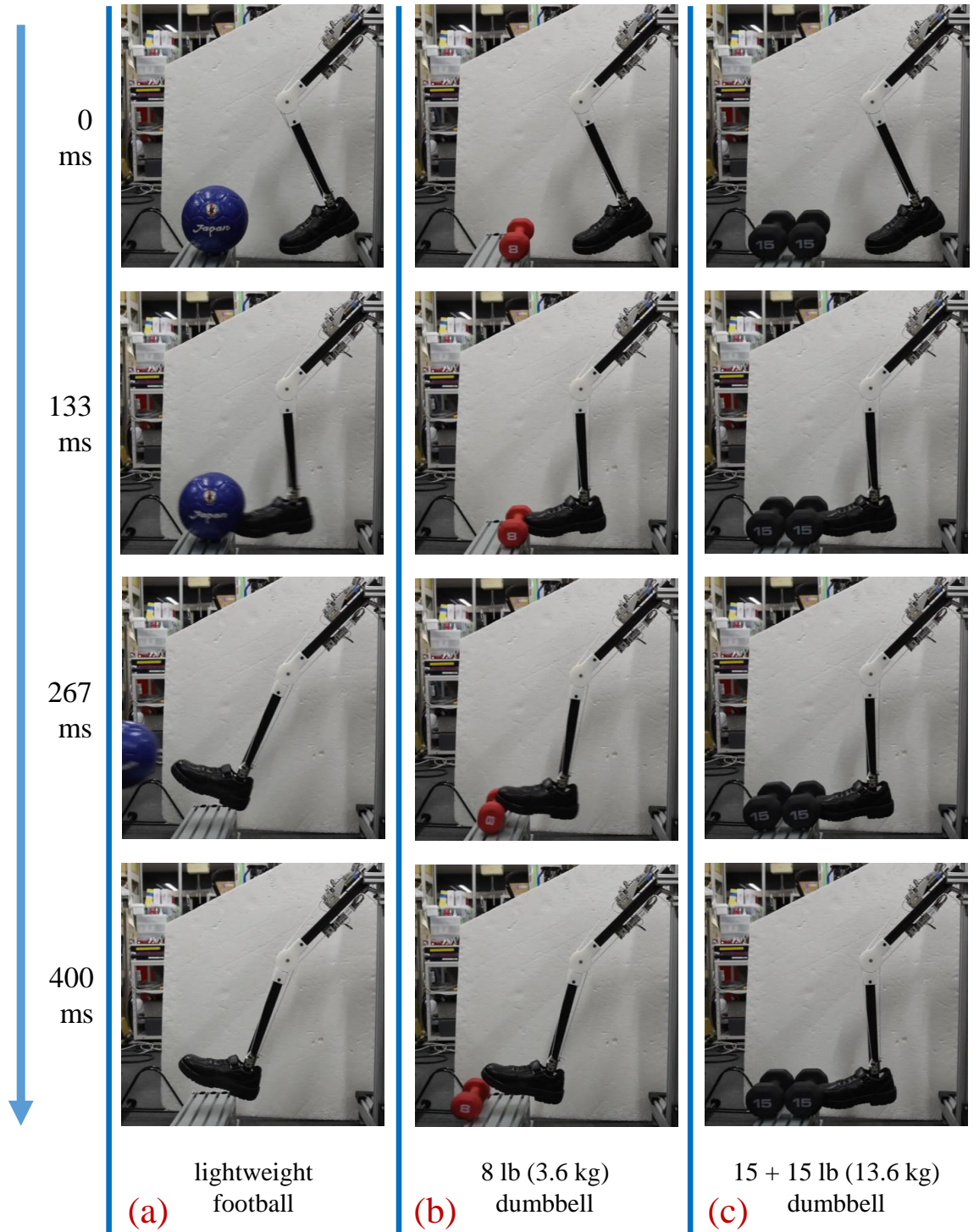


図 6.37 The knee robot kicking object in 300 ms motion

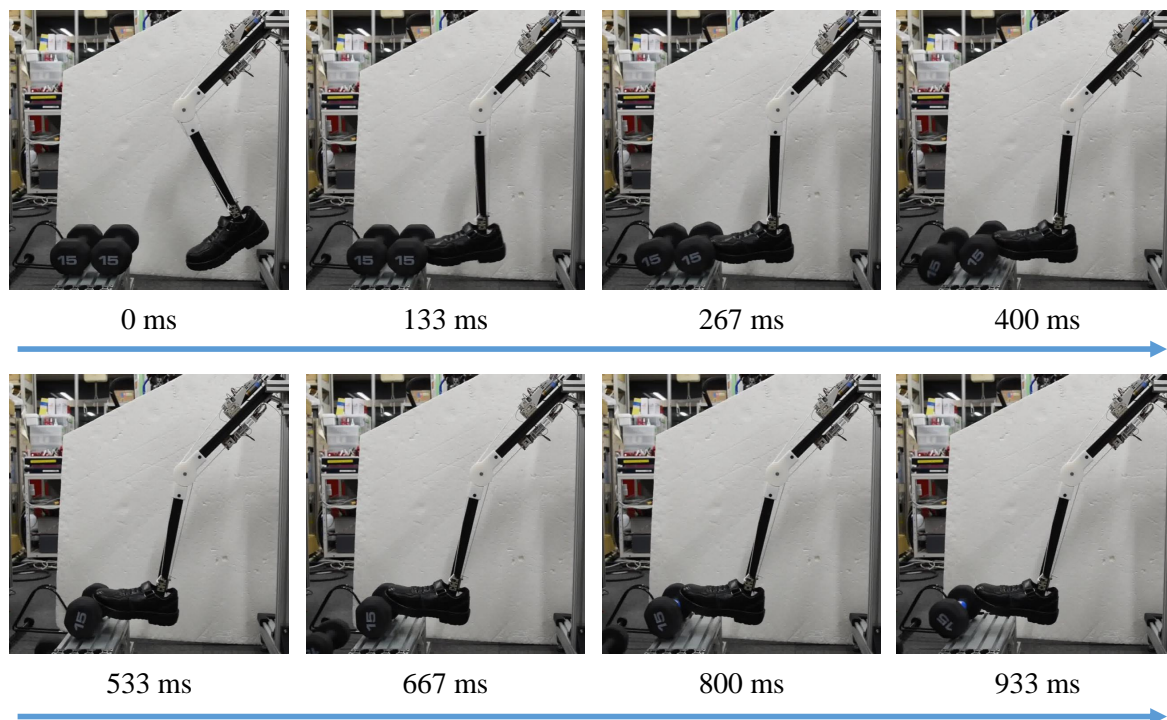


図 6.38 The knee robot kicking 13.6 kg dumbbells in 300 ms motion (8 ms control period, every I-gain is 0.545 rad/Ns)

同様の制御を 8 ms 周期で実行し，合計 13.6 kg のダンベルを蹴り飛ばして比較する．8 ms 周期で安定するように，合計張力制御・拮抗筋張力分配制御・最低張力制御のそれぞれの積分ゲインを全て 0.545 rad/Ns とする．8 ms 周期での実行結果を図 6.38 に示す．接触時に大きな力が発生し，ダンベルを転がしてしまっている．

500 μ s 周期で実行した場合と 8 ms 周期で実行した場合のワイヤ張力を図 6.39 に比較する．500 μ s 周期で実行した場合は接触時の合計張力がおよそ 700 N まで振れている．一方で，8 ms 周期の場合はこれが 1100 N まで増大している．安定な制御を達成するゲイン設定値の範囲が限られるため，追従性が低下している．目標合計張力である 300 N からの誤差はおよそ 2 倍となっている．また，制御周期が大きいために取得できるデータの時間分解能も粗くなっていることが分かる．

6.4.4 ハイブリッド構成ロボットへの拡張展開

筋骨格ロボットでは柔軟な身体構造によるしなやかな環境接触が実現できる一方で，精密動作や高速移動には難点が残る．精密なロボットに柔軟な筋骨格マニピュレータを組み合わせたハイブリッド構成を取ることでより多彩な環境適応行動への可能性が広がる．八

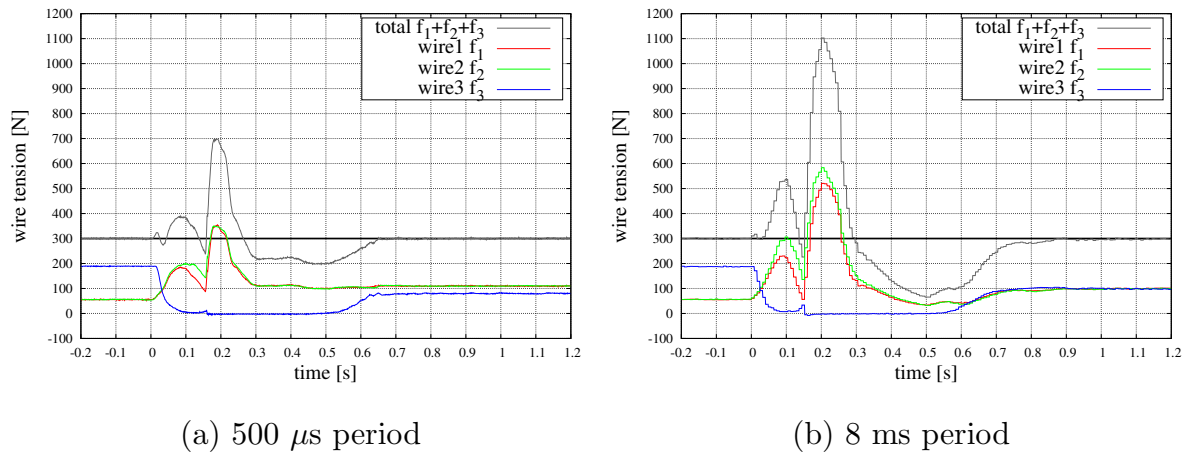


図 6.39 Wire tension of the knee robot while kicking 13.6 kg dumbbells with wire tension distribution control (comparing 500 μ s and 8 ms control period, stable I-gain for each control period)

イブリッド構成ロボットではそれぞれの形態に適切なアクチュエータ制御方式や通信方式を採用する．ロボット体内で複数のインタフェースやプロトコルによる複分散ネットワークが構成される場合も考えられる．ハイブリッド構成ロボットの実時間制御システムでは，単一ロボットの体内分散ネットワークとして複数のインタフェースを統合的に取り扱うことが求められる．また，多くの形態のロボットに適用可能な要素技術を提供することで，更に合理的な体内分散ネットワークが構成可能となる．

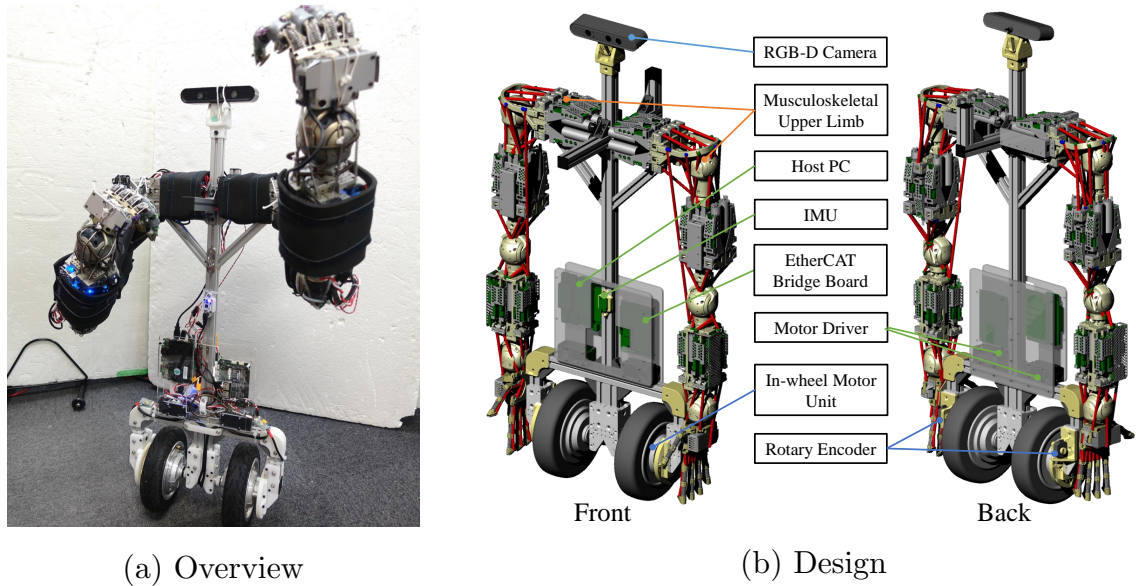
図 6.40 の筋骨格倒立振子ロボット TWIMP [182] は，腱駆動型筋骨格双腕マニピュレータと二輪倒立振子の組み合わせで構成されている．上半身では筋骨格マニピュレータを採用することで柔軟な物体操作を可能としている．インホイールモータを利用した応答性の高い二輪倒立振子により高速な移動を実現している．体幹には倒立振子の制御に用いる IMU を搭載している．

インホイールモータの制御では倒立振子のバランス保持のために高い応答性が求められる．倒立振子の制御は矢状面の線形運動方程式および線形 2 次レギュレータに基づいて設計される姿勢フィードバック制御を基本とする．

矢状面の姿勢角度を θ とする．左右の車輪回転位置を ϕ_l および ϕ_r とすると矢状面の車輪の運動はその平均値 $\phi = (\phi_l + \phi_r) / 2$ で記述できる．状態変数を $x = (\theta, \phi, \dot{\theta}, \dot{\phi})^T$ ，入力を矢状面の車輪トルク τ とした状態方程式，

$$\dot{x} = Ax + B\tau \quad (6.6)$$

は矢状面の倒立振子に関する Lagrangian 運動方程式を線形近似することで得られる．運



(a) Overview

(b) Design

図 6.40 TWIMP: Two-wheel inverted musculoskeletal pendulum [182]

動エネルギー K は,

$$K = K_b + K_w \quad (6.7)$$

$$K_b = \frac{1}{2} m_b \left(\left(R(\dot{\theta} + \dot{\phi}) + L\dot{\theta} + \cos\theta \right)^2 + \left(-L\dot{\theta}\sin\theta \right)^2 \right) + \frac{1}{2} I_b \dot{\theta}^2 \quad (6.8)$$

$$K_w = \frac{1}{2} m_w \left(R(\dot{\theta} + \dot{\phi}) \right)^2 + \frac{1}{2} I_w (\dot{\theta} + \dot{\phi})^2 \quad (6.9)$$

ポテンシャル U は,

$$U = m_b g L \cos\theta \quad (6.10)$$

と表せる．ここで， m_b は倒立振り子となる上半身の質量， I_b は上半身の重心周り慣性モーメント， m_w は車輪の質量， I_w は車輪の慣性モーメント， R は車輪の半径， L は上半身の重心と車輪軸の距離， g は重力加速度である．姿勢と車輪回転に関する Lagrangian 運動方程式，

$$L = K - U \quad (6.11)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = 0 \quad (6.12)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}} \right) - \frac{\partial L}{\partial \phi} = \tau \quad (6.13)$$

より,

$$(a + 2b\cos\theta + c)\ddot{\theta} + (a + b\cos\theta)\ddot{\phi} = b\dot{\theta}^2\sin\theta + d\theta \quad (6.14)$$

$$(a + b\cos\theta)\ddot{\theta} + a\ddot{\phi} = b\dot{\theta}\sin\theta + \tau \quad (6.15)$$

$$a = (m_b + m_w)R^2 + I_w \quad (6.16)$$

$$b = m_bRL \quad (6.17)$$

$$c = m_bL^2 + I_b \quad (6.18)$$

$$d = m_bgL \quad (6.19)$$

となる. $\theta = 0$ 周りで, $\sin\theta \simeq \theta$, $\cos\theta \simeq 1$, $\theta\dot{\theta} \simeq 0$ として線形近似すると, 状態方程式,

$$\mathbf{E}_0\dot{\mathbf{x}} = \mathbf{A}_0\mathbf{x} + \mathbf{B}_0\tau \quad (6.20)$$

$$\dot{\mathbf{x}} = \mathbf{E}_0^{-1}\mathbf{A}_0\mathbf{x} + \mathbf{E}_0^{-1}\mathbf{B}_0\tau \quad (6.21)$$

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a + 2b + c & a + b \\ 0 & 0 & a + b & a \end{pmatrix} \quad (6.22)$$

$$\mathbf{A}_0 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (6.23)$$

$$\mathbf{B}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (6.24)$$

が得られる. 適当な重み Q および R に対して, 評価関数,

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + R\tau^2) dt \quad (6.25)$$

を最小化する線形 2 次レギュレータによって, 最適なゲインによる状態変数誤差 $\tilde{\mathbf{x}}$ のフィードバック制御則,

$$\tau = \mathbf{K}\tilde{\mathbf{x}} \quad (6.26)$$

$$= K_{\theta}\tilde{\theta} + K_{\phi}\tilde{\phi} + K_{\dot{\theta}}\dot{\tilde{\theta}} + K_{\dot{\phi}}\dot{\tilde{\phi}}$$

が求まる.

矢状面のフィードバック制御で得られる車輪トルク τ を左右の車輪に $\tau = \tau_l + \tau_r$ として分配することでインホイールモータの目標出力トルクを決定する. 水平面角度 ψ の誤差 $\tilde{\psi}$ をフィードバックして,

$$\tau_l = \frac{1}{2}\tau - K_{\psi}\tilde{\psi} \quad (6.27)$$

$$\tau_r = \frac{1}{2}\tau + K_{\psi}\tilde{\psi} \quad (6.28)$$

とする。

フィードバックに用いる姿勢角度 θ および水平面角度 ψ は IMU から，車輪回転位置 ϕ_l および ϕ_r はインホイールモータのエンコーダから得られる。

TWIMP の体内分散ネットワークは図 6.41 のように構成される。1 台の中央計算機で筋骨格マニピュレータと二輪倒立振子を制御する。筋骨格マニピュレータでは小型制御基板による分散ネットワークを USB インタフェースで中央計算機に接続する。IMU も別途 USB インタフェースで接続する。二輪倒立振子のインホイールモータは，ギア駆動型大出力ヒューマノイドでも用いられているモータ制御モジュールで駆動する。インホイールモータのモータ制御モジュールは RS-422 汎用シリアル通信による分散ネットワークで接続され，EtherCAT 中継基板を利用して Ethernet インタフェースで中央計算機とデータを交換する。Ethernet インタフェースを利用することで高速・高信頼な通信を小型パッケージで実現し，省スペース性が求められる倒立振子ロボットの体内に収めている。

分散制御ノードが USB と Ethernet のインタフェースで中央計算機に接続する複合的な体内分散ネットワークが構成される。Ethernet で接続された二輪倒立振子の制御において，USB で接続された IMU の情報をフィードバックし，また USB で接続された筋骨格マニピュレータの運動が干渉するため，中枢全身制御系では複数のインタフェースを統合的に取り扱える実時間制御ソフトウェアの実装が効果を発揮する。

6.5 多リンク飛行ロボットの力制御型空中マニピュレーションへの応用

6.5.1 多リンク飛行ロボットへの応用戦略

3 次元のフィールドにおける高い機動力を有する飛行ロボットは，第一次産業や設備管理の分野で監視・検査の用途に活躍してきた。近年では，飛行ロボットの高い機動力を能動的な物体の操作・運搬に活用する期待が高まり，空中マニピュレーションの研究が盛んに行われている [183][184]。多くの研究では飛行ロボットにグリッパや多リンクマニピュレータを組み合わせることで空中マニピュレーションが実施されている。多リンクマニピュレータを利用すると，単独のグリッパによる手法と比べて物体に印加可能な操作力方向・軌道の制約が少なく，広い問題を取り扱うことができる。更にマルチロータ構造そのものを多リンク化することで全身を利用した環境接触や大型物体の把持を目指した，多リンク飛行ロボットによる空中マニピュレーション手法が開発されている [185][186][187]。

多リンク飛行ロボットに関する先行研究では 3.3.3 項で紹介したように，関節を既製品のサーボモータモジュールで駆動し，独自の制御基板による CAN バス分散ネットワークを経由してサーボモータモジュールに指令を与えていた。サーボモータモジュールでは与

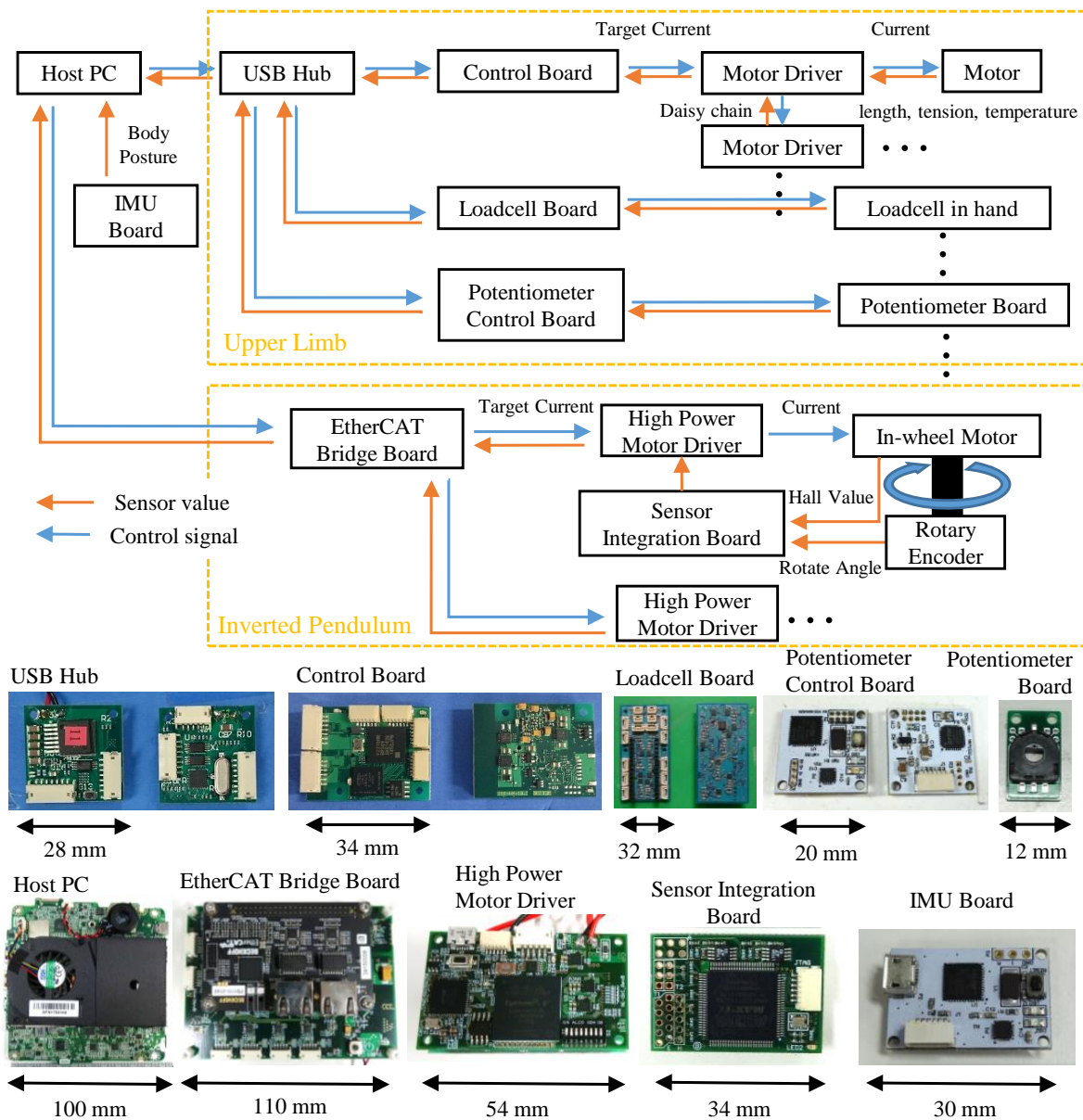


図 6.41 Circuit configuration of TWIMP [182]

えられた位置指令に基づくサーボ制御が実行される。

多リンク飛行ロボットによる物体の抱え込み把持動作では、ロボットのリンクと操作対象物体による閉リンクが生じる。そのため、適切な把持力を発揮するためにはフィードフォワードな位置制御では不十分であり、力・トルクの制御が求められる。また飛行ロボットの姿勢制御に伴い、プロペラには回転方向トルクや変動的な推進力が反力として生じ、これが関節の負荷に干渉する。多リンク飛行ロボットに搭載された既製品の小型サーボモータモジュールでは簡易的な出力負荷情報が得られるため、これを用いた負荷制限制御が実装されている [185]。閾値を超える負荷を検出した場合にこれを緩和するように関

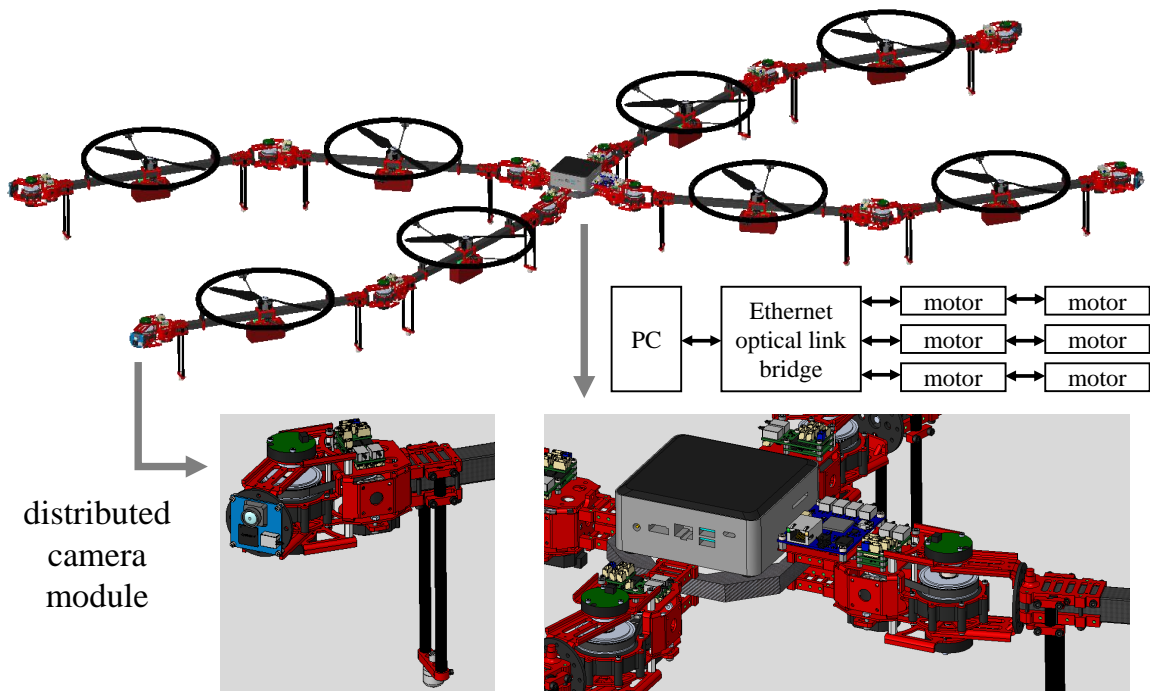


図 6.42 Example of optical network for multilink aerial robot

節位置指令を更新することで過大な負荷を避ける。

簡易的な負荷制限制御では比較的軽量で頑丈な物体のマニピュレーションに限られる。実用的な空中マニピュレーションの応用では、より精密で大きな把持力が求められる。操作対象物体を含めて閉リンクが構成され、また飛行姿勢制御が関節トルクに干渉するため、全身フィードバックの応答性が重要である。ヒューマノイドの関節制御に用いる小型高性能なモータ制御モジュールを同様に飛行ロボットに適用することで、

- 小型パッケージでの大出力モータ駆動
- 電流計測に基づく精密な出力トルクフィードバック
- 既製品の小型サーボモータモジュールの仕様に囚われない関節制御実装
- 高速光通信による低遅延広帯域ネットワーク
- 組込中間層計算機を利用した低遅延全身フィードバック

の効果により、高度な空中マニピュレーションが実現可能となる。

多リンク飛行ロボットへの小型光通信モータ制御モジュールの適用例を図 6.42 に示す。多リンク飛行ロボットを構成するリンクを図 6.43 に示す。それぞれのリンク間を関節モジュールで接続する。

関節モジュールには小型光通信モータ制御モジュールを搭載し、ブラシレスモータおよ

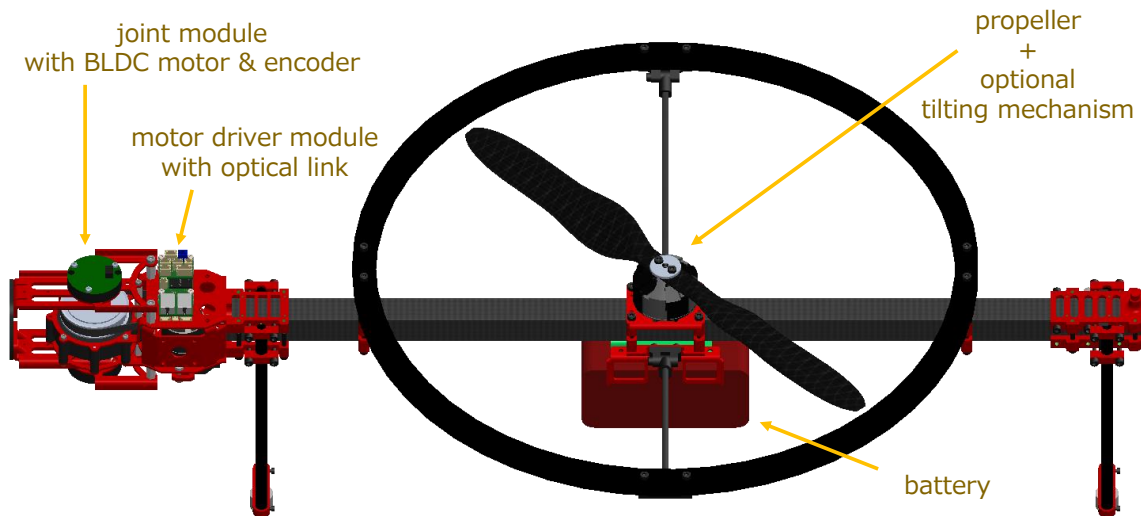


図 6.43 Link module with optical network for multilink aerial robot

び減速機で関節を駆動する．モータ制御モジュールではモータや関節のエンコーダ値を取得してブラシレスモータを制御し，リンクに搭載されたプロペラに出力指令を与える．高速光通信による分散ネットワークが中央計算機に対して Ethernet インタフェースで接続される．

モータ制御モジュールではブラシレスモータ巻線電流をフィードバックすることで精密な出力トルクを制御することができる．閉リンクが構成されるマニピュレーションにおいても適切な把持力を安定維持することが可能となる．位置制御を利用する場合においても分解能の高いエンコーダを利用した剛性の高いサーボ制御が可能となる．

高速光通信により大きな通信帯域が確保され，実時間制御周期の短縮や通信データ量の増大に対応できる．関節モジュールに搭載された複数のエンコーダを含む関節駆動情報やプロペラ回転情報に加え，IMU やバッテリー電圧など多くの情報を，短い実時間制御周期のうちに中枢全身制御系と共有することができる．行動・マニピュレーション計画において重要となるカメラや距離センサなどを末端リンクに分散配置し，高速光通信ネットワーク経由で中央計算機にデータを送信することもできる．飛行ロボットの構成ではリンク間を接続するケーブルを減らすことも重要な課題である．高速光通信を利用すると図 3.15 に示した非常に細い光ファイバでノード間が接続される．通信信号が絶縁されるため電源電位を共有する必要がなく，リンクごとにバッテリーを搭載することでリンク間の接続を光ファイバのみとすることができる．更に多自由度化が進んで合計のリンク長さが延伸しても，リンクごとに絶縁することで電気配線長が小さく抑えられ安定した通信が可能となる．

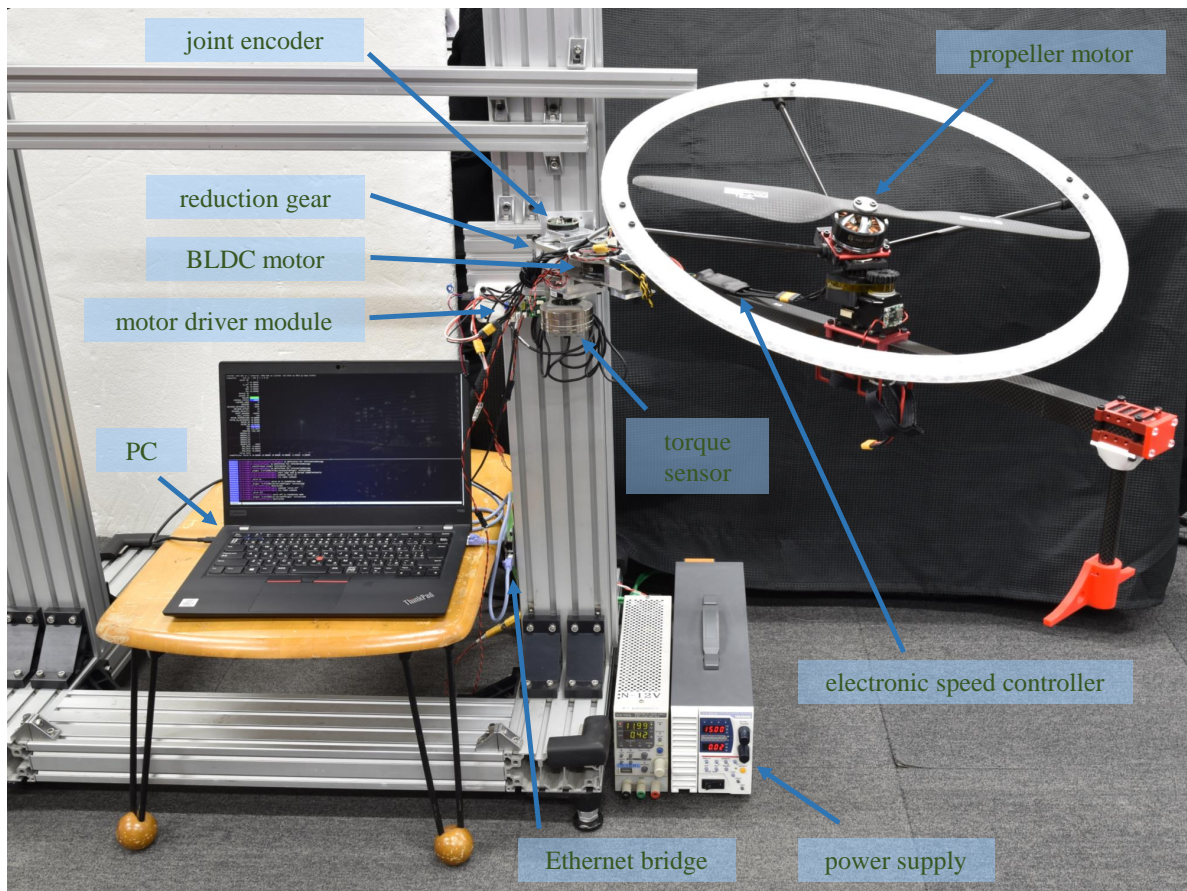


図 6.44 Experimental setup overview of the aerial robot link module

6.5.2 飛行ロボット関節モジュールにおける把持力制御

飛行ロボットに向けて小型光通信モータ制御モジュールを搭載した関節モジュールを実装し、精密で大きな把持力を実現するための関節制御を実証する。固定土台に関節モジュールと飛行ロボットリンクを組み付けて構成した実験装置の概観を図 6.44 に示す。

関節モジュールでは、ブラシレスモータの出力を減速比 3.14 のタイミングプーリおよび減速比 31 のサイクロ減速機で減速して関節を駆動する。総減速比は 97.4 である。飛行ロボットへの適用を目指して小型軽量で大出力な T-MOTOR 社のブラシレスモータ MN3508 を用いる。モータにはエンコーダが搭載されていないため、CUI Devices 社の 14 bit アブソリュートエンコーダ AMT21 を接続してモータの回転位相を検出する。小型光通信モータ制御モジュールによりモータを駆動する。関節モジュールには関節のエンコーダを搭載し、また実験装置における検証用に関節トルクセンサを接続している。これらの周辺センサを小型光通信モータ制御モジュールに接続して、データを取得する。

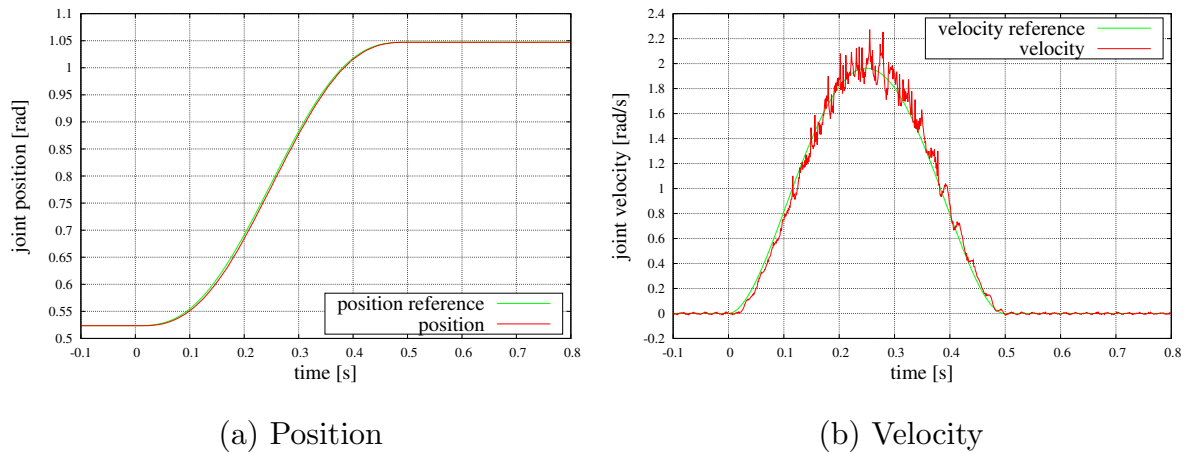


図 6.45 Joint reference trackability in position-controlled motion by the aerial robot link module

飛行ロボットリンクではプロペラを T-MOTOR 社のブラシレスモータ MN4010 で回転させる。プロペラには精密な位置サーボではなく低負荷高速回転が求められるため、センサレス速度制御を可能とする ESC 製品を用いて駆動する。ESC として T-MOTOR 社の AIR 40A を用いる。関節モジュールに搭載した小型光通信モータ制御モジュールに ESC を接続して PWM 信号により速度指令を与える。

1 組の関節モジュールおよび飛行ロボットリンクの制御を 1 台の小型光通信モータ制御モジュールに集約させている。これを 1 ノードとしてロボット体内ネットワークを構成する。実験装置では小型 Ethernet 中継基板を経由して評価実験用の中央計算機に接続する。

小型光通信モータ制御モジュールでは中央計算機からの制御指令に基づいて、電流制御もしくは位置制御により関節モータを駆動する。環境接触を伴わない姿勢制御の場合は位置制御を用いることで、中央計算機で計画された姿勢を容易に再現することができる。位置指令に対する追従性の例を図 6.45 に示す。関節角度 $\frac{\pi}{6}$ の範囲を 500 ms の躍度最小軌道で高速に移動している。指令値に対してよく追従していることが確認できる。

能動的な環境接触を伴う空中マニピュレーションにおいては、電流制御を応用することで適切な把持力を発揮することができる。サイクロ減速機を中心に構成される関節の損失は比較的小さく、モータ q 軸電流におよそ比例した関節トルクが出力される。小型光通信モータ制御モジュールを用いることで、従来利用していた既製品の小型サーボモータモジュールの仕様に囚われず、精密で大きな把持力を発揮するための柔軟な制御系を構成することが可能となる。

中央計算機では目的となる空中マニピュレーションに応じて適切な把持力を計画し、これに基づいた電流指令を小型光通信モータ制御モジュールに与える。基本的には把持力に

比例した電流指令値を与えればよいが，一定の電流指令値を与えるのみでは，物体にアプローチするまでの無負荷状態で過度に加速してしまう．最大アプローチ速度を設定してこれに基づく粘性を加えることで過度な加速を避けることができる．

把持状態における目標モータ q 軸電流 I_{grasp} は，目標把持力 f_{grasp} ，関節軸から把持点までのモーメントアーム l ，関節減速比 R ，モータの電流トルク定数 K_{τ} より，

$$I_{\text{grasp}} = \frac{lf_{\text{grasp}}}{RK_{\tau}} \quad (6.29)$$

と表せる．無負荷状態でのアプローチ中に発生する最大関節速度を \dot{q}_{max} に抑えたい場合，関節速度に関する粘性係数 $\frac{I_{\text{grasp}}}{\dot{q}_{\text{max}}}$ を電流指令値に設定すれば良い．関節速度 \dot{q} をフィードバックして，モータに与える電流指令値 I_{ref} を，

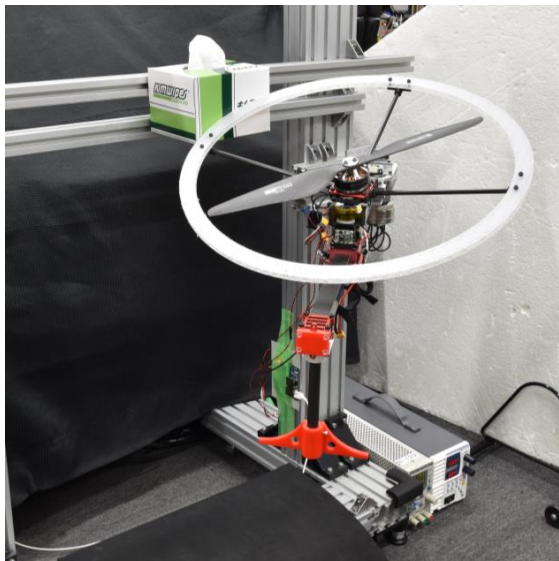
$$I_{\text{ref}} = \begin{cases} I_{\text{grasp}} & (\dot{q}\text{sgn}I_{\text{grasp}} < 0) \\ I_{\text{grasp}} \left(1 - \frac{\dot{q}\text{sgn}I_{\text{grasp}}}{\dot{q}_{\text{max}}}\right) & (0 \leq \dot{q}\text{sgn}I_{\text{grasp}} \leq \dot{q}_{\text{max}}) \\ 0 & (\dot{q}_{\text{max}} < \dot{q}\text{sgn}I_{\text{grasp}}) \end{cases} \quad (6.30)$$

と求める．把持方向関節速度 $\dot{q}\text{sgn}I_{\text{grasp}}$ が負の場合は把持状態にあるものとして目標把持力を発揮する．把持方向関節速度が設定した最大値を超えている場合は，主に目標把持力と逆方向の外乱が加わっている状態が想定されるが，これは出力電流 0 で受け流す．

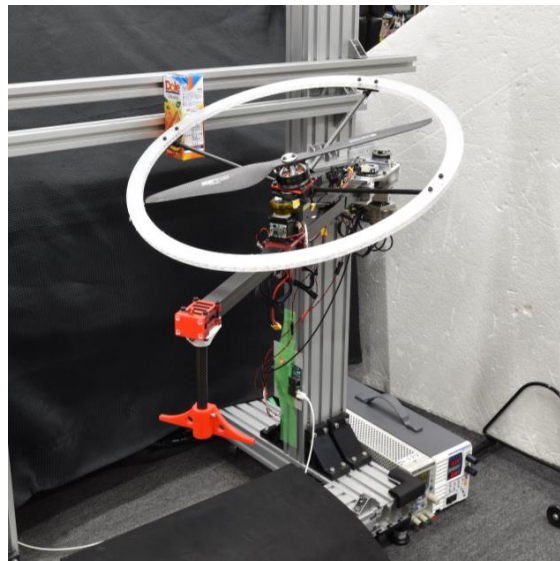
把持力制御の実行例を図 6.46 に示す．4 種類の物体をそれぞれ異なる目標把持力で把持している．図 6.46 (a) では軽量のペーパーオールの箱を関節トルク $lf_{\text{grasp}} = 1 \text{ Nm}$ で把持している． $l = 0.4 \text{ m}$ であり目標把持力 $f_{\text{grasp}} = 2.5 \text{ N}$ に相当する．図 6.46 (b) ではおよそ 200 g の紙パックを $lf_{\text{grasp}} = 2 \text{ Nm}$ ，すなわち $f_{\text{grasp}} = 5 \text{ N}$ で把持している．図 6.46 (c) ではおよそ 350 g のペットボトルを $lf_{\text{grasp}} = 4 \text{ Nm}$ ，すなわち $f_{\text{grasp}} = 10 \text{ N}$ で把持している．図 6.46 (d) ではおよそ 600 g のペットボトルを $lf_{\text{grasp}} = 6 \text{ Nm}$ ，すなわち $f_{\text{grasp}} = 15 \text{ N}$ で把持している．物体の重量や脆性を考慮した把持計画を受けて，目標把持力を使い分けることが可能となっている．

目標関節トルク $lf_{\text{grasp}} = 6 \text{ Nm}$ に相当する目標モータ q 軸電流 $I_{\text{grasp}} = 4.83 \text{ A}$ のステップ入力に応じて，物体にアプローチして把持した際の応答を図 6.47 に示す．最大関節速度を $\dot{q}_{\text{max}} = 1.27 (\simeq 0.4\pi) \text{ rad/s}$ に設定している．関節速度値 \dot{q} ， q 軸電流出力値 $I \simeq I_{\text{ref}}$ ，およびトルクセンサで計測した関節トルク値をプロットしている．

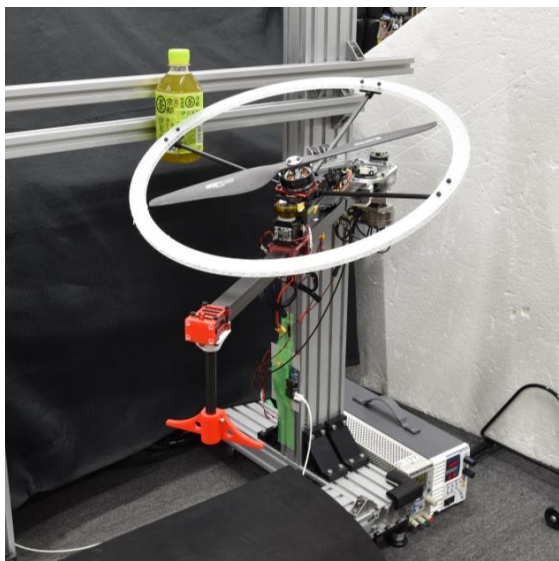
時刻 0 で目標モータ q 軸電流 4.83 A のステップ入力を与える．関節速度 0 から開始するためおよそ 4.83 A に近い電流が瞬間的に出力される．これにより関節が加速して，無負荷回転中はおよそ 1.27 rad/s に近い関節速度を維持する．摩擦などによって僅かに必要となる電流の分だけ小さな関節速度で釣り合う．物体と接触すると速度が 0 となり，電流は 4.83A に収束する．トルクセンサ計測値からもおよそ 6 Nm のトルクが発揮されていることが確認できる．



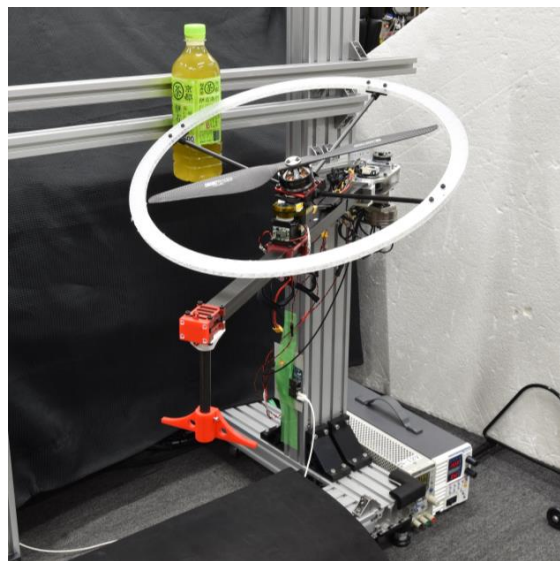
(a) Tissue box (1 Nm torque)



(b) 200 mL juice box (2 Nm torque)



(c) 350 mL tea bottle (4 Nm torque)



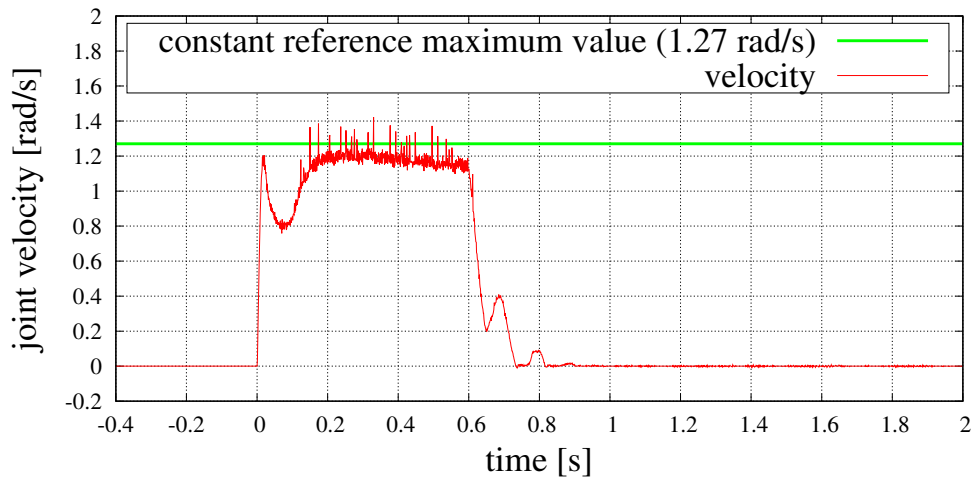
(d) 600 mL tea bottle (6 Nm torque)

図 6.46 The aerial robot link module grasping several objects by force control

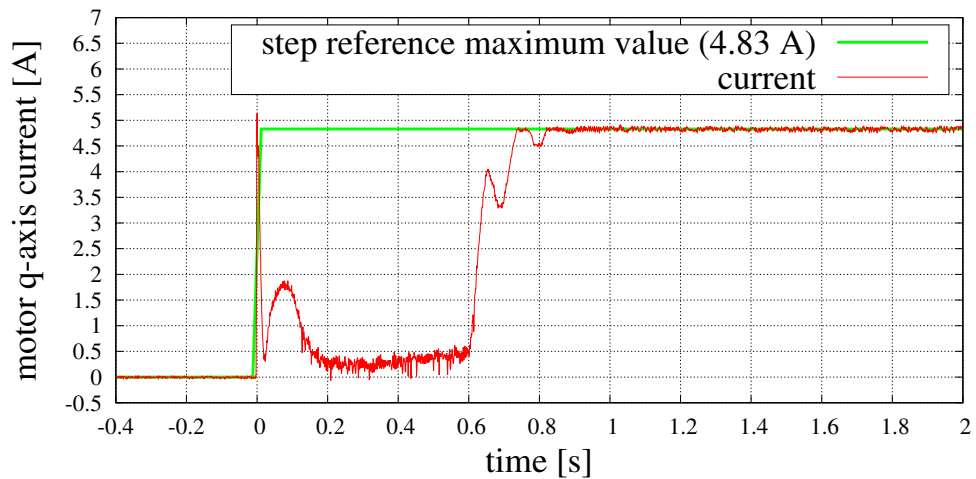
6.6 連続跳躍ロボットの高速追従姿勢安定化への応用

6.6.1 連続跳躍ロボットへの応用戦略

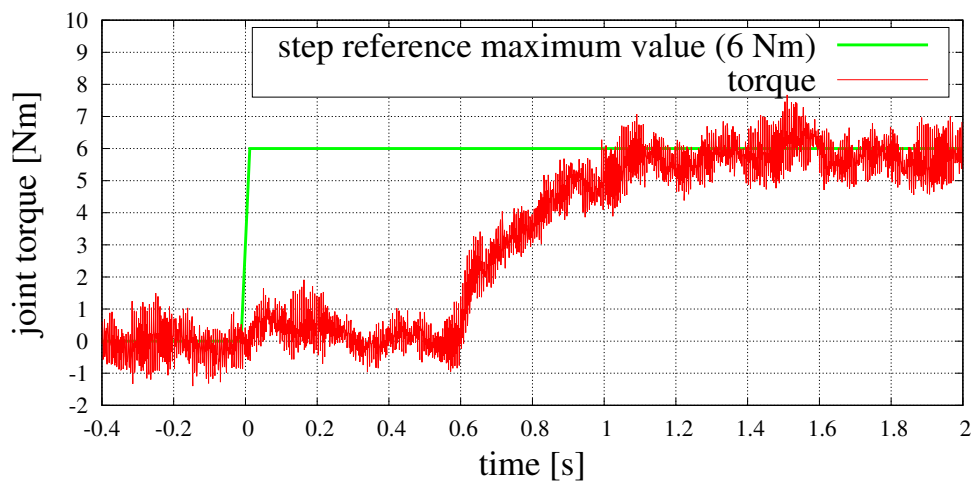
連続したホッピングを行う脚ロボットは段差や不整地を含むような多様な地形における高速移動を実現するものとして期待され、機構ハードウェアや制御に関する多くの研究が



(a) Joint velocity



(b) Motor q-axis current



(c) Joint torque

図 6.47 Response trajectory in force-controlled grasping motion by the aerial robot link module

行われてきている [188][189][190][191][192]．連続跳躍ロボットは滞空と接地を繰り返す中で，跳躍に求められる鉛直方向の大きな運動量を制御し，所望の水平方向移動を実施するとともに，姿勢制御により転倒や暴走を防止しなければならない．滞空・接地は通常およそ数百 ms の高速な周期で繰り返される．ホッピングを保つためには，姿勢を速やかに目標値に追従させる応答性の高いフィードバック制御が求められる．

連続跳躍ロボットには事前の関節位置軌道計画に対して精密に追従する性能よりも，長期的に安定した跳躍着地動作を繰り返すことへの適正が求められる．しなやかな弾性を有する駆動系ハードウェアを用いることで，衝撃を受け流すと同時にエネルギー効率を向上させることができる．また高ゲインな位置制御型ではなく力制御型のアクチュエータを用いることで外乱に対して滑らかに応答し，同じく衝撃を受け流しエネルギー効率も向上する．ハードウェア弾性や力制御型アクチュエータを活用したホッピングを実施する上で，モデル通りの挙動をフィードフォワードに再現するのは困難であり，全身姿勢フィードバック制御の性能が殊更に重要な要素となる．応答性の高いフィードバック制御の高信頼な実現がホッピングの成功に向けた鍵を握る．

跳躍動作を担う大出力な駆動系における力制御は，5.4.3 項のモータ制御モジュールローカルフィードバック型関節トルク制御を応用することで実現できる．駆動系に内在する損失を補償して所望の力を発揮させる．モータ制御モジュールおよび中央計算機を低遅延分散ネットワークで接続し，全身姿勢フィードバック制御を行う．小型制御基板および Ethernet 基幹通信インタフェースを利用して耐衝撃性を向上した高信頼な分散ネットワークを構成する．ロボットが頻繁に空中に浮上し，またハードウェアの弾性が大きいいため，全身のセンサ情報を用いた姿勢の推定が必要となる．モータ制御モジュールから得られるモータ位置情報に加えて，モータ制御モジュール上に搭載した IMU の情報を組み合わせることで，分散ネットワークと親和性の高い低遅延姿勢推定およびフィードバック制御が可能となる．

6.6.2 ワイヤ張力制御と全身姿勢フィードバックに基づく連続跳躍

大出力モータ制御モジュールと拮抗ワイヤ駆動系を組み合わせた図 6.48 のパラレルワイヤ駆動跳躍ロボットに体内分散実時間制御システムを適用する．パラレルワイヤ駆動跳躍ロボットは駆動系や電子制御系を収めた本体と跳躍力を伝達するための軸部から構成される．本体と軸部は回転 2 自由度・直動 1 自由度の関節で接続される．本体の駆動系から巻き取るワイヤ末端を軸部に固定する．上下にそれぞれ 3 本ずつのワイヤを配置することで，6 自由度アクチュエータの拮抗駆動により 3 自由度機構の制御を可能としている．

大出力モータ制御モジュールとブラシレスモータでウィンチを駆動することでワイヤを巻き取る．Ethernet 中継基板を用いて中央計算機を含む分散ネットワークを構成す

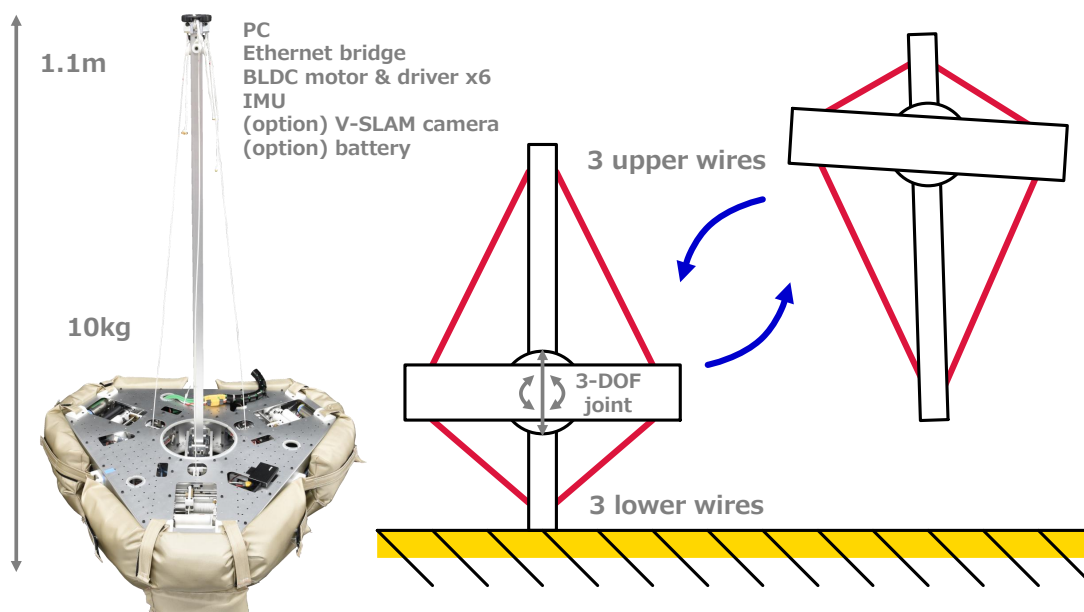


図 6.48 Overview of the wire-driven hopping robot

る．従来より用いられていた大出力モータ制御モジュールには IMU が搭載されておらず，3.5.2 項の小型モータ制御モジュールを同じネットワークに接続して IMU 情報を取得する．

ホッピング制御は，

- 姿勢推定
- 目標関節位置軌道修正
- 関節力・ワイヤ張力制御

により構成される．これらを周期的に反復する実時間制御として実装する．姿勢推定では主にワイヤ長および IMU の情報から，拮抗駆動される関節位置，空間中のロボット姿勢，接地状態を推定する．精度が求められる場合や空間中の絶対位置の情報が必要となる場合などはこれに応じて V-SLAM カメラ・モーションキャプチャなどの情報を併用する．実時間分散ネットワークで得られるワイヤ長・IMU の情報と比べると，V-SLAM カメラ・モーションキャプチャは応答性に欠けるため，適切に使い分ける必要がある．目標関節位置軌道修正では推定された姿勢情報を用いて，ホッピングを達成するための関節位置軌道を求める．接地中は水平な姿勢を整えながら，跳躍に求められる運動量を鉛直方向に発揮する．滞空中は姿勢を安定化させ水平方向の目標速度に追従するために，着地点および軸部の角度を更新する．関節力・ワイヤ張力制御では，目標関節位置軌道に追従するための

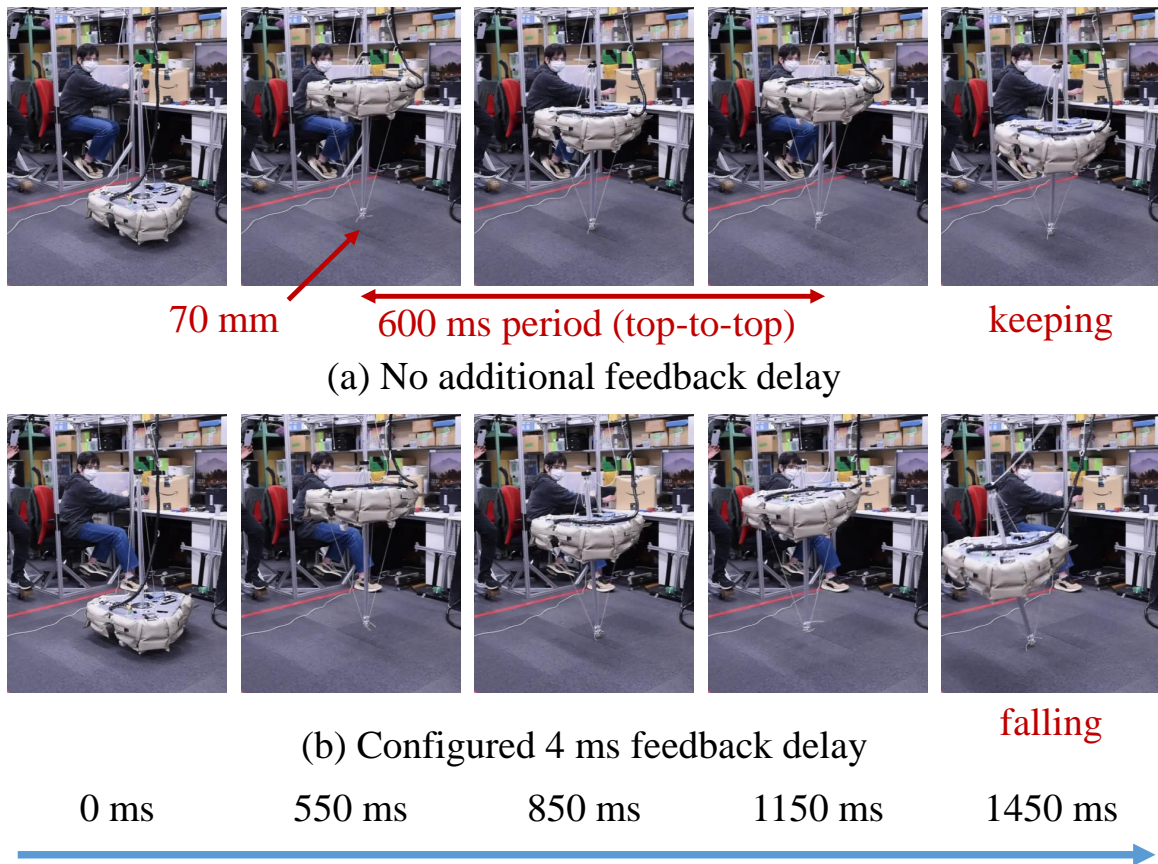


図 6.49 Hopping experiment with the wire-driven hopping robot (0 ms and 4 ms feedback delay added)

関節力を求めて、ワイヤ張力指令に変換してモータ制御モジュールへ出力する。

目標関節位置軌道修正では、外乱やモデル化誤差を速やかに補償して姿勢を安定化させるためにフィードバックの応答性が重要となる。関節力・ワイヤ張力制御では、高ゲインで目標位置軌道指令によく追従するために低遅延で安定したフィードバック制御が重要となる。ホッピング制御におけるフィードバック遅延の影響を検証する。

一般化デバイス制御プラットフォームの同期プラグインとして実装したホッピング制御において、ワイヤ長および IMU の値更新に遅延を挿入する。図 6.49 に、遅延を設けない場合と 4 ms の遅延を挿入する場合のホッピング動作の様子を比較する。図 6.49 (a) では追加の遅延を設けずにホッピング動作を実行している。およそ 600 ms 周期で高さ 70 mm の跳躍を繰り返している。一方 4 ms の遅延を挿入した図 6.49 (b) ではホッピング制御が不安定となりやすく、バランスを崩す様子が見られる。

追加遅延無しから 10 ms 遅延までの条件でサンプルを取得して図 6.50 に比較する。初回の射出時刻を 0 として、ワイヤ巻取り位置およびロボット姿勢の応答波形をプロット

する．ワイヤ巻取り位置は射出前の初期姿勢を 0 として相対的なモータの回転位置から求めている．ロボット姿勢はワイヤ長および IMU の情報から推定した値である．ワイヤ長からヤコビ行列変換モデルおよび拡張 Kalman フィルタによって関節位置を推定する．IMU に Madgwick フィルタを適用して本体姿勢を推定する．この組み合わせにより軸部の姿勢も推定される．

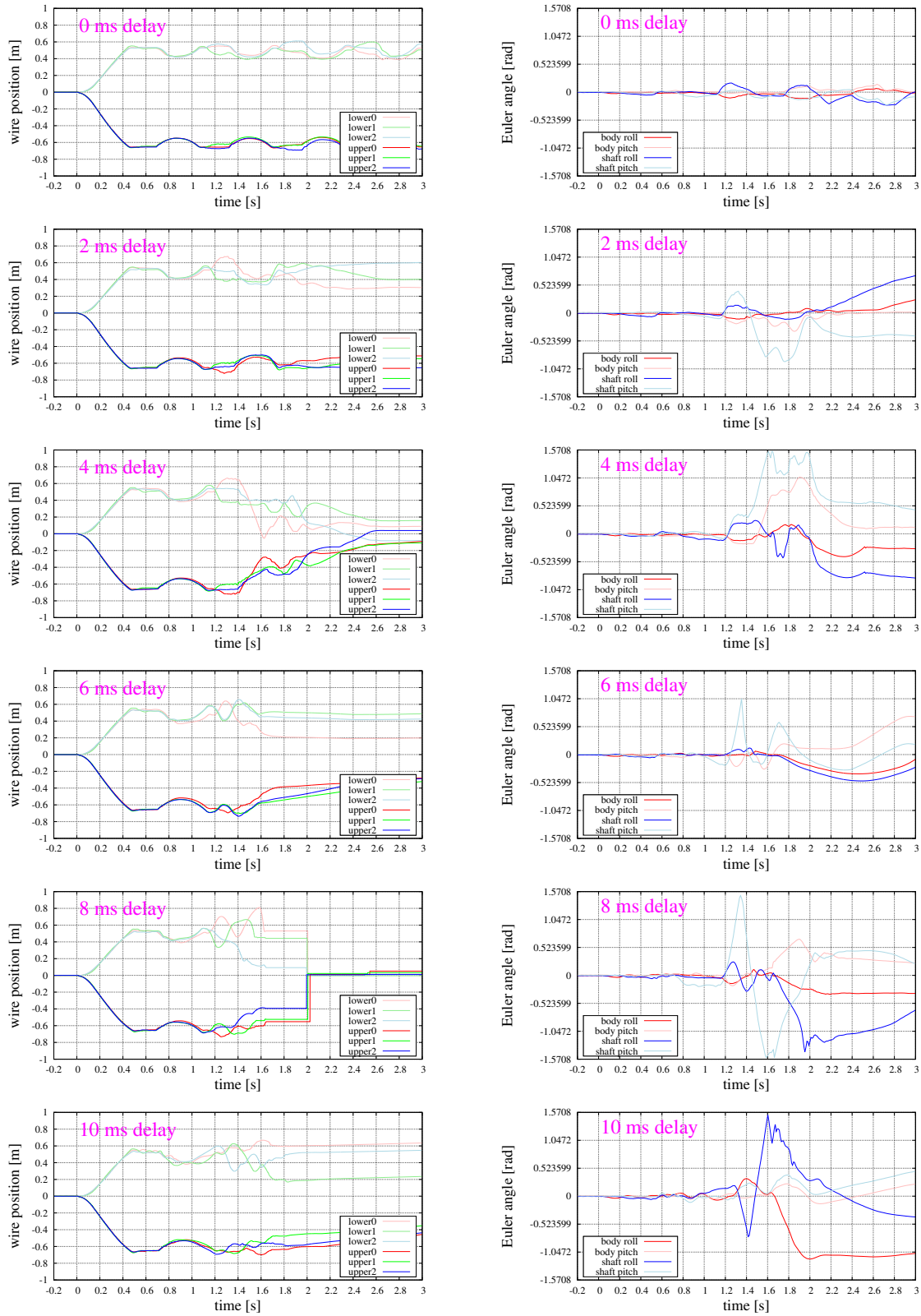
遅延無しの場合，射出からおよそ 2.8 s のうちに 4 周期のホッピングを行っている．姿勢の傾きも小さく抑えられていることが確認できる．2 ms の遅延を挿入した場合，2 回目の跳躍着地で姿勢を崩して 3 回目の跳躍に失敗した．更に遅延を大きくしてゆくと制御がより振動的になり早い段階から姿勢を崩すようになる．

低遅延なホッピング制御により連続した 8 回のホッピングに成功した．図 6.51 に，ホッピングを 8 回実行した際の応答波形を示す．図 6.51 (a) はワイヤの巻取り位置である．図 6.51 (b) はモータ制御モジュールへの出力指令となるワイヤ張力である．図 6.51 (c) は Intel 社の V-SLAM カメラ RealSense T265 を搭載して推定したロボット位置軌道である．図 6.51 (d) はワイヤ長・IMU から推定したロボット姿勢である．姿勢の傾きも小さく安定したホッピング動作が実現されている．ロボットの水平位置は速度 0 を指令値として与えているが，その積分に相当する絶対位置への追従は行っていない．8 回のホッピングでバランスを崩したのではなく水平位置のドリフトが大きくなったため，ホッピングを中止して着陸させている．

6.7 本章のまとめ

本章では，本研究で提案する体内分散実時間制御システムを多様なロボットへ適用するための応用戦略とその実証評価に関して述べた．

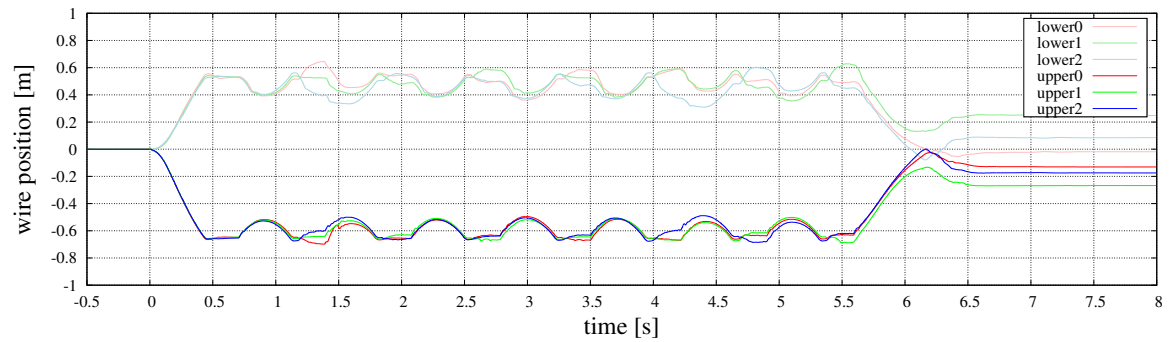
ギア駆動型大出力ヒューマノイドでは，基幹通信ネットワークに従来利用されていた PCI バスを Ethernet に置き換えて，小型化したハードウェアで振動に対して頑強な高信頼通信を実装する．100 Mbps の EtherCAT から Gigabit Ethernet へと段階的に適用を進め，高速で柔軟なプロトコル実装による低遅延化を果たす．組込分散ネットワークに従来利用されていた RS-422 汎用シリアル通信を高速光通信に置き換えて，高速・高信頼通信を適用する．関節トルク制御や組込中間層計算機を活用した低遅延フィードバックにより，精密性と柔軟性を兼ね備えた動作制御を実現する．Ethernet インタフェースを利用したダイナミック行動の達成事例や，関節トルク制御を利用した外乱応答行動の達成事例を紹介した．腕部マニピュレータに高速光通信と組込中間層計算機を適用した実験により，低遅延フィードバックで衝撃緩和が達成され，アドミタンス制御で設定した機械インピーダンスに近い力学特性を再現可能となることを示した．安定した環境接触を実現し，衝撃緩和と組み合わせて能動的な打撃作業への応用を実現した．2 脚ロボットに



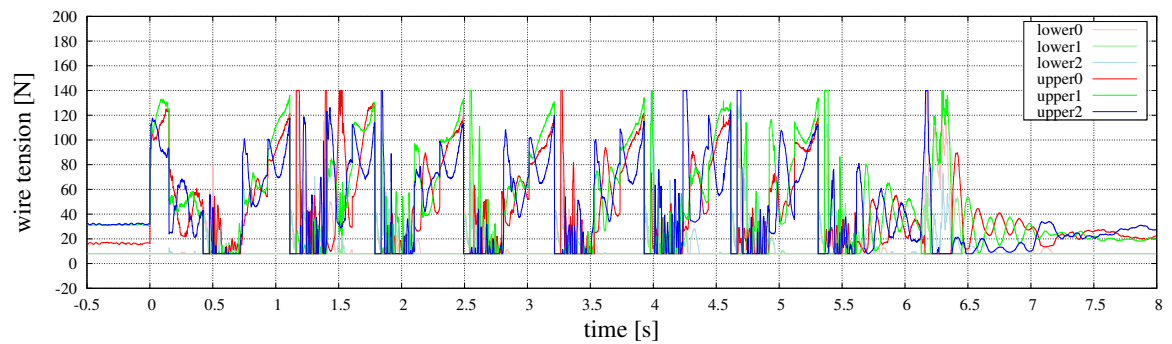
(a) Wire position

(b) Body link tilt

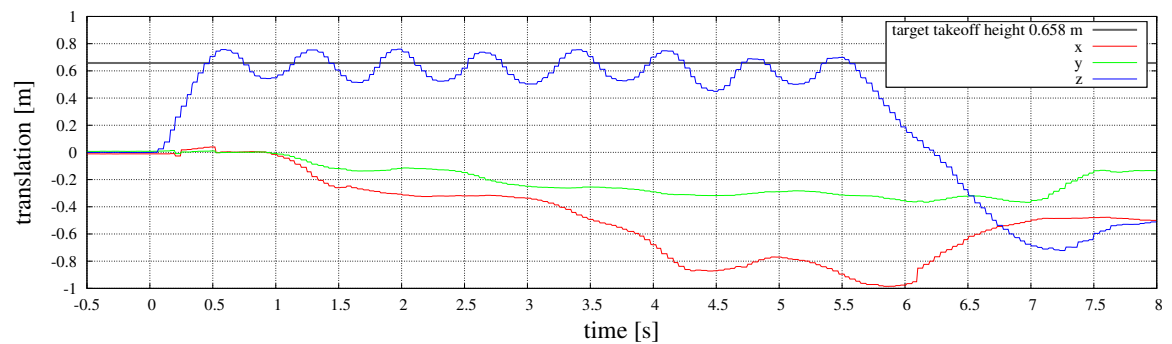
図 6.50 Wire winding position and estimated body link tilt angle of the wire-driven hopping robot with configured feedback delay (comparing between hopping control on PC with 0-10 ms variable delay of motor position and IMU data)



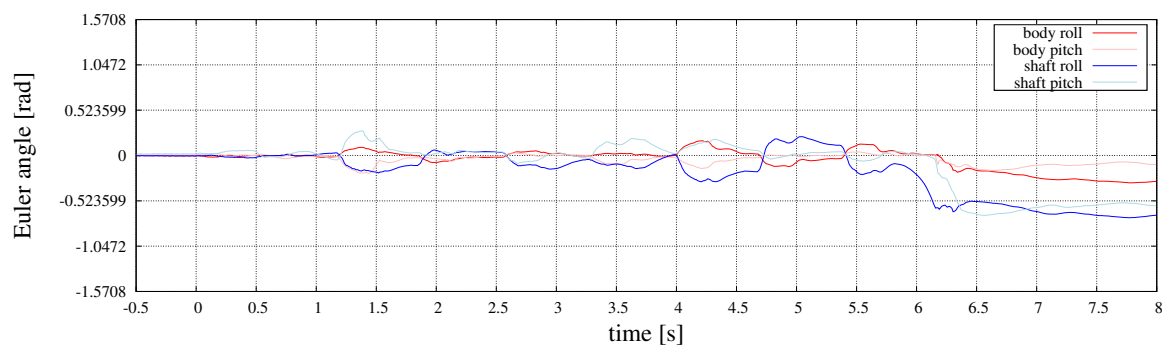
(a) Wire winding position



(b) Controlled wire tension



(c) Body translation measured by V-SLAM camera



(d) Estimated body link tilt angle

図 6.51 Motion trajectory in 8 step experiment with the wire-driven hopping robot

Ethernet インタフェースと関節トルク制御を適用した実験により，着地衝撃緩和の可能性を示し，250 mm の高さからの落下着地を達成した．全身フィードバックの応答性を向上することで，精密な関節位置制御を利用した跳躍と着地衝撃緩和を実現した．

腱駆動型筋骨格ヒューマノイドでは，従来利用されていた MCU による USB Full-Speed ネットワークを FPGA による USB High-Speed ネットワークに置き換えて，制御周期を短縮し，交換可能データ量を増大する．脚部ロボットでの実証により，従来 8 ms に制限されていた制御周期がおよそ 1 ms まで低減可能であることを示した．更に小型高速通信モータ制御モジュールを適用することで，省スペース性を保った実装で，ギア駆動型大出力ヒューマノイドとの共通技術による低遅延な高信頼ネットワークと応答性の高いサーボ制御が実現可能であることを述べた．膝部ロボットに小型高速通信モータ制御モジュールを適用し，安定で追従性の高い共同筋・拮抗筋のワイヤ張力分配制御による打撃作業を実現した．USB ネットワークと Ethernet ネットワークを併用したハイブリッド構成ロボットへの拡張事例を紹介した．

多リンク飛行ロボットでは，既存技術によるフィードフォワードな関節位置制御実装で空中マニピュレーションの力制御に課題が残されていた．より精密で大きな把持力が求められる実用的な空中マニピュレーションのために小型高速通信モータ制御モジュールの適用が有効である．小型光通信モータ制御モジュールを用いた実装により力制御性能の向上に加えて，大規模データ通信への応用や，絶縁による高信頼運用が可能となる．小型軽量のブラシレスモータを小型光通信モータ制御モジュールで駆動する構成の飛行ロボット関節を用いて，電流制御に基づく精密で大出力な把持動作を実現した．

連続跳躍ロボットでは，駆動系ハードウェアの弾性を活用するとともに力制御を適用することが，しなやかな跳躍着地に向けて有効となる．しなやかな跳躍着地を安定して継続させるために応答性の高い姿勢フィードバック制御が求められる．モータ制御モジュールによるローカルフィードバック型トルク制御および低遅延分散ネットワークにより，しなやかさを活用した連続跳躍を実現する．継続した跳躍着地のために小型制御基板や Ethernet を利用して衝撃に対する信頼性を向上する．パラレルワイヤ駆動跳躍ロボットを用いて，ワイヤ張力制御および低遅延全身姿勢フィードバックに基づく安定なホッピング動作を実現した．

第 7 章

結論

7.1 本論文の総括

本論文では多形態多自由度ロボットにおける体内分散実時間制御システムの設計実現法を論じ、その応用実装と評価に関して述べた。以下に各章の概要をまとめる。

- 第1章「序論」では、複雑な行動実現が期待されるロボットの多形態化および多自由度化が進んでおり、即応的な環境適応行動実現のために全身フィードバック動作制御の低遅延化が重要な鍵となることを述べた。これを受けて、フィードバック遅延低減を主軸として実時間性および展開実用性を満たすロボット体内制御システム設計実現法の樹立を本研究の目的として掲げた。
- 第2章「多自由度ロボットの動作制御系に求められる実時間環境適応性能」では、多形態多自由度ロボットの環境適応行動実現に適う体内制御システムの要件として実時間性および展開実用性を挙げ、これを満たす体内分散実時間制御システムの設計実現法に関して述べた。環境適応行動の中でも特に衝撃的な環境接触を伴う打撃作業や跳躍着地が挑戦的な課題であることを述べた。ヒューマノイドにおける実例に基づいて、多自由度ロボットに適した体内分散制御システムの構成を述べた。ロボットの動作制御におけるフィードバック遅延の課題を分析し、低遅延・低ジッタ・信頼性に基づく実時間性、および小型軽量・低消費電力・汎用性・運用性に基づく展開実用性の重要性を述べた。背景技術の分析に基づき、多形態多自由度ロボットにおける体内分散実時間制御システムの設計実現法として、子細適応法・統合節減法・実地進展法を提案した。体内分散実時間制御システムを構成する、感覚駆動通信系・組込中間自律系・全身階層協調系の設計手法を示した。最新の技術に追従しロボット研究開発の現場で柔軟に実装仕様を更新する体内制御システムの継続的な開発が重要であることを述べた。
- 第3章「体内分散実時間ネットワークを形成する小型高信頼な感覚駆動通信系」では、多自由度ロボットの体内分散ネットワークには実時間制御周期のうちに大量のデータを転送する性能が必要であることを、ヒューマノイドの例による試算とともに述べた。汎用シリアル通信や USB Full-Speed を利用した従来技術では要求性能に対して不十分であり転送可能なデータサイズや制御周期に律速が生じていることを分析した。省スペース要求を伴う多自由度ロボットにおいて高速・高信頼な分散ネットワークを適用するためのモジュール設計手法を述べた。ベクトル電流制御型モータ駆動回路および高速通信インタフェースを搭載する小型高速通信モータ制御モジュールを開発した。従来型の 26.667 Mbps 汎用シリアル通信に加えて、高速シリアル通信技術に基づく 2.5 Gbps の高速光通信を実装し、汎用シリアル通信と比べて数十倍以上に及ぶデータ転送性能を示した。基幹通信ネットワークにおいて、FPGA と最小限の汎用物理層回路を組み合わせた構成により、ロボットの研究

開発における要求に応える柔軟な通信プロトコル実装が可能となることを述べた。従来利用されていた PCI バスの課題を分析し、PCI Express・Gigabit Ethernet・USB High-Speed による次世代通信インタフェースを検討した。FPGA を活用して Ethernet や USB のインタフェース性能を引き出す小型モジュールの実装例を与え、データ交換低遅延化の効果を検証した。

- 第 4 章 「階層アーキテクチャ設計に基づく低遅延全身応答のための組込中間自律系」では、中央計算機および組込中間層計算機による相補的なヘテロジニアスアーキテクチャに基づくロボット中枢全身制御系を提案した。高次の計画系と実時間の実行系が中央計算機の資源を共有し Ethernet インタフェースで組込分散ネットワークと接続する構成における実時間性の課題を述べた。ハードウェアアクセラレータを伴う FPGA SoC および GPU SoC を搭載した組込中間層計算機を導入することで低遅延動作制御が実現可能となることを述べた。FPGA SoC および GPU SoC を小型パッケージに搭載した組込中間層計算機モジュールを開発した。組込中間層計算機モジュールの開発事例を通じて、高度な組込計算技術およびネットワーク技術を小型パッケージ化してロボットに導入するための電子制御系ハードウェア設計における技術要件を明らかとした。ロボット技術と実装技術の両観点から実時間性や展開実用性を満たすための要件を示した。FPGA および GPU によるハードウェアアクセラレーションの効果を行列演算時間に基づいて評価し、中枢全身制御系における相補的なヘテロジニアスアーキテクチャの効果を示した。組込中間層計算機による低遅延フィードバックで用いられるパケット切替中継機構を実装し、FPGA を利用した柔軟なデータ転送の効果を確認した。
- 第 5 章 「多形態多自由度動作制御における組込高速演算を交えた全身階層協調系」では、位置制御型および力制御型のアクチュエータ制御システムに基づくロボットの力フィードバック動作制御構造に関して考察した。これに基づき、組込技術を活用して力フィードバック制御を低遅延に実行する手法を検討した。ギア駆動型ロボットのモータ制御モジュールにおいて低遅延なローカル関節トルク制御を実現するために、アクチュエータごとに実行可能な慣性・摩擦補償型の関節トルク推定機構を FPGA 上に実装し、効果を検証した。多様なロボットの中枢全身制御系において実時間実行系に見られる共通構造を分析した。低遅延なデータ転送およびデータ処理を共通構造化した一般化デバイス制御プラットフォームを提案し、そのプラゲイン拡張機能により低遅延実時間動作制御を可能とした。全身フィードバック動作制御を組込中間層計算機に展開する手法を検討し、FPGA SoC にアドミタンス制御を実装した。中央計算機では周期タスク実行に $425 \mu\text{s}$ を要する一方で FPGA SoC では $188 \mu\text{s}$ での実行を可能とした。順運動学計算に FPGA ハードウェアアクセラレーションを適用し、さらなる低遅延化の可能性を見出した。

- 第 6 章 「体内分散実時間制御システムの応用戦略と行動実装評価の実現」では、本研究で提案する体内分散実時間制御システムが多岐に渡る形態のロボットへ応用可能であることを述べた。ヒューマノイド技術を応用したいくつかのロボットに体内分散実時間制御システムを適用し、その効果を検証した。ギア駆動型大出力ヒューマノイドに Ethernet 基幹通信ネットワークと高速光通信を適用し、小型ハードウェアによる高速・高信頼通信を実現した。関節トルク制御や組込中間層計算機を適用し、低遅延な力フィードバック制御を実現した。腕部マニピュレータや 2 脚ロボットによる実験を通して、低遅延フィードバックにより衝撃緩和が可能となることを示した。腕部マニピュレータでは柔軟な機械インピーダンスの再現や安定した接触が可能であることを示し、未知の対象物体に対する安全な打撃作業を実現した。2 脚ロボットでは関節トルク制御に基づく 250 mm の高さからの着地、および位置制御全身フィードバックに基づく跳躍着地を達成した。腱駆動型筋骨格ヒューマノイドに FPGA による USB High-Speed ネットワークを適用し、従来 8 ms に制限されていた制御周期をおよそ 1 ms まで低減した。更に小型高速通信モータ制御モジュールを適用することで、サーボ制御の応答性や通信ノイズ耐性に残されていた課題を解消し、打撃作業における応答的な筋張力分配を実現した。柔軟な筋骨格マニピュレータと高速移動向けの駆動系を組み合わせたハイブリッド構成ロボットへの応用も紹介した。多リンク飛行ロボットに小型光通信モータ制御モジュールを適用することで、空中マニピュレーションのための精密で大きな把持力制御が可能となることを示した。大規模データ通信や絶縁高信頼運用への応用も可能となることを述べた。高信頼な電子制御系による応答性の高い力制御および全身姿勢制御を連続跳躍ロボットへ応用することで、精密性と柔軟性を兼ね備えた安定なホッピング動作を実現した。

7.2 本研究の成果と結論

本研究では、多自由度ロボットの環境適応行動実現に向けた全身フィードバック動作制御系の応答性向上を目的とし、多形態多自由度ロボットに適用可能な体内分散実時間制御システムの設計実現法を提案した。

応用的な行動実現が期待される多自由度ロボットの複雑な体内制御システムにおいて、従来、全身動作制御のフィードバック遅延が課題となっていた。特に精密性と柔軟性の両立が求められる衝撃的環境接触を伴う行動実現には課題が多く、電子制御によって衝撃をしなやかに受け流すことは困難であるとされてきた。

そこで本研究では、半導体技術とともに益々性能を上げる高速通信および高速演算の技

術を背景に，これを活用した低遅延フィードバック動作制御の実現を図った．多様な形態のロボットに合理的な低遅延フィードバック動作制御系を展開するために，実時間性および展開実用性をロボット体内制御システムに求められる要件として掲げた．実時間性および展開実用性の要件を満たす合理的な体内分散実時間制御システムの設計実現法として，子細適応法・統合節減法・実地進展法を提案した．

子細適応法 低遅延制御を達成する実時間システムの実装評価法として，物理層の全機能にアクセス可能な制御回路系によりロボット応用に適応したフィードバック制御系を構成する子細適応法を提案した．産業分野で成熟した応用メカトロニクス製品のアプリケーション実装に囚われることなく先端技術を積極的に取り入れ，FPGAを活用して細部に至るまで妥協の無いロボット体内制御システムを実装する効果を示した．現在得られる高速シリアル通信や Ethernet，組込ハードウェアアクセラレーションなどの先端技術を取り入れたシステム実装例を与えて効果を検証した．1 Gbps を超える高速シリアル通信では従来の汎用シリアル通信の数十倍以上に及ぶデータ転送レートを達成し，光伝送を媒体として省スペース・高信頼に実装した．Gigabit Ethernet や USB High-Speed を用いた通信プロトコルの実装例を交えて，通信インタフェースが本来有する性能を FPGA で引き出す回路構成と時間確定的なパケット中継プロトコル実装の有効性を示した．同期・非同期・臨時のプラグイン拡張を可能とする一般化デバイス制御プラットフォームや組込中間層計算機におけるロボット制御計算アクセラレーションの実装により，中間層を利用したフィードバック遅延低減の効果を明らかにした．

統合節減法 多形態多自由度ロボットへの合理的展開法として，ロボット体内制御システムの共通構造定義を行うことで小さなオーバヘッドでシステム性能を維持しながら展開コストを抑える統合節減法を提案した．小型高速通信モータ制御モジュールや組込中間層計算機モジュールの設計開発を通して，小型軽量化および低消費電力化の厳しい要求に応えるロボット体内制御システムの設計要件を明確化した．汎用性の高い物理層回路構成を複数の機能で共有することで，省スペース性を保ったままシステムが多形態ロボットに展開可能となることを示した．主要な実時間動作制御への応用を網羅する論理パケット構造定義に基づく全身制御ソフトウェアおよびパケット中継回路実装により，異なる通信インタフェースを有する多形態ロボットの体内制御システムを共通化した．先端技術を取り入れて実時間性および展開実用性を満たすロボット体内制御システムを実装する上で，ロボット技術と実装技術のそれぞれの特性をともに深く理解して作り込む重要性を示した．

実地進展法 ロボット体内制御システム継続発展に向けた設計実現法として，実際のロボット応用と動的評価に基づく子細な仕様更新を繰り返してシステムを進展させる

実地進展法を提案した。ロボットの行動実現を目指す研究開発の中では、体内制御システムにおける実時間性および展開実用性の課題が常々新たに判明する。FPGAに実装する体内分散ネットワーク機能と共に時間計測回路やロジックアナライザを併設することで、ネットワークの性能をロボット運用中に動的評価する手法を示した。実証評価を通して体内制御システムの仕様を更新し、将来に渡る技術進展に追従してロボット研究開発の挑戦を支え続ける効果と重要性を明らかにした。

低遅延フィードバック制御を実現する提案システムの実証評価を通して、遅延がロボット動作制御の外乱応答性能に与える影響の程度を明らかにした。感覚駆動通信系・組込中間自律系・全身階層協調系の構成に基づき、多様な形態のロボットに同様に適用可能な体内分散実時間制御システムの応用実装を与え、低遅延フィードバック制御系アーキテクチャにおける共通構造を見出した。

ギア駆動ロボットに擬似的な力学特性を再現する力制御では、通信遅延やソフトウェア遅延が衝撃的外乱への応答に大きな影響を与えることを示し、組込中間層計算機を利用した低遅延フィードバックによって目標の力学特性の再現に成功した。従来困難とされていた電子制御による衝撃的環境接触への対応が、フィードバック遅延の低減により達成される可能性を見出した。

剛性の高いギア駆動ロボットとは逆に駆動系ハードウェアに柔軟性を有する筋骨格ロボットでは、精密な力や姿勢の環境接触を維持することが重要な課題となっていた。柔軟な接触を適切に制御して打撃作業やダイナミック動作を実現するためにも、フィードバック遅延の低減による安定した目標追従が効果を発揮することを示した。すなわちいずれの場合でも、精密性と柔軟性を両立するにはフィードバック制御の応答性向上が不可欠なものとなっている。拮抗筋や共同筋を含む冗長駆動系により厳しい小型化要求が生じる筋骨格ロボットにおいても、小型化するFPGAや通信回路といった先端技術を取り入れて実用化してゆく体内制御システムの設計手法を明らかにした。

小型軽量ギア駆動の多リンク飛行ロボットやワイヤ駆動の連続跳躍ロボットなど、更に異なる形態で厳しい要求を有するロボットにも共通構造で提案システムを適用し、環境適応行動に向けた応答性の課題を解決する手法となっていることを実証した。

本研究による成果は、現在の先端技術に基づく実装例を与えたとともに、将来的な技術向上の中での課題解決の指標を明らかにしたものである。多様な形態のロボットに適用可能な共通構造の実装を示したことにより、その成果は今後の多くのロボット開発に継承され、進化してゆく技術に追従しこれを統合するための基盤となる。提案手法の効果は次世代のロボット技術発展において広く波及するものと考えられる。

7.3 将来への展望

ロボットの体内制御システムにおいて先端的な通信技術や演算技術を積極的に取り入れてその設計要件を明確化することの真価は、多くのロボットに派生する中で新たに生じる要求を技術開発にフィードバックし、加速的な発展を促すことにある。ロボット技術と実装技術の両者に基づいて明確化された設計要件により、次世代ロボットの開発が促進される。既存の成熟したアプリケーション実装に囚われない応用展開により、先端半導体技術開発とロボット技術開発の距離が縮まり、新たな技術革新へと繋がる。

本研究で利用した高速シリアル通信や FPGA SoC の実装は、主に産業分野やデータセンタ用途の需要を受けて開発されてきたものである。ロボットに浸透させることでロボット体内分散制御システムならではの小型化や低コスト化、低消費電力化などの需要が高まり、これを反映した新技術が登場するのではないかと期待する。それにより更に多くのロボットに適用可能な技術となり、加速的な発展が見込まれる。

小型光通信モータ制御モジュールの実用化に向けて、高速シリアル通信を利用可能な FPGA の消費電力はひとつの大きな懸念事項として残っている。小型パッケージに収めるために排熱もより困難となっており、過熱状態となることも想定される。小型システムに適した高速シリアル通信物理層の実現が望まれる。

腱駆動型筋骨格ヒューマノイドなどの非常に多自由度で高密度にアクチュエータ搭載されるロボットで真に実用化可能な高速ネットワークとなれば、全身の膨大なデータを共有することができ、多自由度ロボットのデータサイエンス研究における新たなビッグデータ処理のアプローチが登場することも予想される。本研究では主に幾何・力学モデルに基づいた低遅延フィードバック制御に焦点を当てて開発と実証を進めてきた。しかし全身の膨大なデータを組込 GPU や FPGA で処理して蓄積してゆくことで、モデルを持たないフィードバック制御系を学習して獲得してゆく手法も考えられる。モデルでは定式化できない筋同士の高周波な協調関係や、駆動系とは一見関係のなさそうなデータ項目による干渉などに対応して、従来困難とされてきた柔軟身体によるバランス保持やダイナミック動作も実現しうる。

分散ネットワークを低コストで実装できるようになれば、主要なモータ制御モジュールのみでなく他の多様なデバイスをロボットに搭載してネットワークを構成する応用も考えられる。パラレルワイヤ駆動跳躍ロボットでは小型モータ制御モジュールを IMU デバイスとしてネットワークに接続した。多リンク飛行ロボットではカメラ画像転送への応用に関して検討した。他にも腱駆動型筋骨格ロボットに搭載される触覚センサや関節エンコーダ、あるいはフィールドロボットの測距センサなど、体内分散ネットワークに接続して環境適応行動に向けた低遅延フィードバックに利用したいデバイスは多く挙げられる。

多リンク飛行ロボットに向けて実証を進めた関節モジュールは小型で多彩な駆動に対応している。飛行ロボット以外にも多くのモジュール型ロボットへの応用が考えられる。関節モジュールに加えて、小型軽量で計算効率の高い中枢全身制御系実装として組込中間層計算機の技術を応用し、体内制御システムをコンパクトに収めた多形態ロボットの展開も期待される。

謝辭

本論文は筆者が東京大学大学院情報理工学系研究科知能機械情報学専攻博士課程在学中に、情報システム工学研究室において、稲葉雅幸教授のご指導のもとで執筆に取り組んだ論文です。

稲葉雅幸教授にはお忙しい中多くのご指導を頂きました。筆者が研究の進め方に迷うことも多々生じた中で、誰よりも熱心に情熱的に耳を傾け相談に乗っていただき、大変励みになりました。広い視野から道を照らしていただき、人生の選択において大切な考え方も多く学びました。稲葉先生のもとで多くのロボット技術に触れながら研鑽を積めたこと、これからも感謝と誇りを抱いて参ります。

岡田慧教授にはロボットのシステム実装を広く知り尽くした観点から多くの率直なご助言を頂きました。東京大学工学部に進学して以来、現在に至るまでの学生生活も多方面から支えていただきました。

お忙しい中、本論文の審査において大変貴重なご意見を下さりまいした、國吉康夫教授、深尾隆則教授、竹内昌治教授にもこの場を借りて感謝申し上げます。客観的な観点に立ち返り、研究における哲学を見つめ直すことができました。

情報システム工学研究室の教員、元教員の方々にも大変お世話になりました。垣内洋平特任准教授、野沢峻一先生にはヒューマノイド研究の進め方について親身に相談に乗っていただきました。菅井文仁先生、白井拓磨先生には本研究を遂行する上で多くの直接的な技術指導を頂きました。浅野悠紀特任講師、小島邦生助教、石黒康裕先生には実験や実装のサポートを頂きました。

本研究の開発成果およびロボットへの応用実験には、多くの方から協力を頂いています。鈴木裕登君、小松慎太郎君にはギア駆動型大出力ヒューマノイドおよび関節トルク制御の開発と実験を共に遂行していただきました。安齋智紀君、真壁佑君には高速光通信開発と飛行ロボット関節制御実験に協力いただきました。河原塚健人君には腱駆動型筋骨格ヒューマノイド実験に協力いただきました。利光泰徳君、鈴木天馬君には執筆終盤の修羅場となった連続跳躍ロボットの実験を共に遂行していただきました。中島慎介さん、小椎尾侑多君、石川達矢君、片山健君、矢野倉伊織君をはじめ、多くの同僚や卒業生の方々のおかげで楽しい研究生活を送ることができました。

最後に、家族をはじめ本論文の完成を支えてくださった全ての方々に感謝いたします。ありがとうございました。

2021年12月3日 永松 祐弥

発表文献

筆頭著者論文

国際会議

1. Yuya Nagamatsu, Takuma Shirai, Hiroto Suzuki, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Distributed torque estimation toward low-latency variable stiffness control for gear-driven torque sensorless humanoid. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4239–5244, 2017.
2. Yuya Nagamatsu, Fumihito Sugai, Kei Okada, and Masayuki Inaba. Basic implementation of FPGA-GPU dual SoC hybrid architecture for low-latency multi-DOF robot motion control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7255–7260, 2020.

国内学術講演会

3. 永松祐弥, 白井拓磨, 大久保壮一, 熊谷伊織, 菅井文仁, 垣内洋平, 岡田慧, 稲葉雅幸, 溝谷圭悟, 久村雄輔, 山崎信行. 実時間通信 Responsive Link 付き I/O Core を用いた大出力脚制御システムの構成と評価. 第 15 回 SICE システムインテグレーション部門講演会講演概要集, pp. 2447–2452, 2014.
4. 永松祐弥, 白井拓磨, 大久保壮一, 熊谷伊織, 菅井文仁, 垣内洋平, 岡田慧, 稲葉雅幸, 大沢幸平, 山崎信行. 実時間通信 Responsive Link 型体内通信系を備えた大出力ヒューマノイドにおける衝撃反射応答行動の実現. 日本機械学会ロボティクス・メカトロニクス講演会'15 講演論文集, pp. 1P2-B07, 2015.
5. 永松祐弥, 白井拓磨, 鈴木裕登, 垣内洋平, 岡田慧, 稲葉雅幸. 大出力軸駆動ヒューマノイドのための高速衝撃加速度センシングと関節トルク制御に基づく関節衝撃緩和機構. 第 21 回ロボティクスシンポジウム予稿集, pp. 7–12, 2016.
6. 永松祐弥, 菅井文仁, 浅野悠紀, 白井拓磨, 岡田慧, 稲葉雅幸. 市販コンピュータを搭載する多自由度ヒューマノイドの FPGA 低遅延分散制御系の検討と実装. 日本機械学会ロボティクス・メカトロニクス講演会'19 講演論文集, pp. 1P2-J02, 2019.

受賞

7. 浅野悠紀, 永松祐弥, 中島慎介, 岡田慧, 川崎宏治, 稲葉雅幸. 腱駆動ヒューマノイド要素技術を活用した義足プロトタイプの開発と車両ペダル操作における基礎的検討. SI2020 優秀講演賞.

共著論文

学術論文誌

8. Makoto Miyamura, Toshitsugu Sakamoto, Xu Bai, Yukihide Tsuji, Ayuka Morioka, Ryusuke Nebashi, Munehiro Tada, Naoki Banno, Koichiro Okamoto, Noriyuki Iguchi, Hiromitsu Hada, Tadahiko Sugibayashi, Yuya Nagamatsu, Soichi Ookubo, Takuma Shirai, Fumihito Sugai, and Masayuki Inaba. NanoBridge-based FPGA in high-temperature environments. *IEEE Micro*, Vol. 37, No. 5, pp. 32–42, 2017.
9. Kento Kawaharazuka, Kei Tsuzuki, Yuya Koga, Yusuke Omura, Tasuku Makabe, Koki Shinjo, Moritaka Onitsuka, Yuya Nagamatsu, Yuki Asano, Kei Okada, Koji Kawasaki, and Masayuki Inaba. Toward autonomous driving by musculoskeletal humanoids: A study of developed hardware and learning-based software. *IEEE Robotics and Automation Magazine*, Vol. 27, No. 3, pp. 84–96, 2020.
10. Yasuhiro Ishiguro, Tasuku Makabe, Yuya Nagamatsu, Yuta Kojio, Kunio Kojima, Fumihito Sugai, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit TABLIS. *IEEE Robotics and Automation Letters*, Vol. 5, No. 4, pp. 6419–6426, 2020.

国際会議

11. Hiroto Suzuki, Yuya Nagamatsu, Takuma Shirai, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Torque based stabilization control for torque sensorless humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 425–431, 2017.
12. Takuma Shirai, Yuya Nagamatsu, Hiroto Suzuki, Shunichi Nozawa, Kei Okada, and Masayuki Inaba. Design and evaluation of torque based bipedal walking control system that prevent fall over by impulsive disturbance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 739–746, 2018.
13. Kento Kawaharazuka, Tasuku Makabe, Shogo Makino, Kei Tsuzuki, Yuya Nagamatsu, Yuki Asano, Takuma Shirai, Fumihito Sugai, Kei Okada, Koji Kawasaki, and Masayuki Inaba. TWIMP: Two-wheel inverted mus-

- culoskeletal pendulum as a learning control platform in the real world with environmental physical contact. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 784–790, 2018.
14. Shintaro Komatsu, Yuya Nagamatsu, Tatsuya Ishikawa, Takuma Shirai, Kunio Kojima, Yohei Kakiuchi, Fumihito Sugai, Kei Okada, and Masayuki Inaba. Humanoid robot's force-based heavy manipulation tasks with torque-controlled arms and wrist force sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3055–3062, 2019.
 15. Kento Kawaharazuka, Shogo Maino, Kei Tsuzuki, Moritaka Onitsuka, Yuya Nagamatsu, Koki Shinjo, Tasuku Makabe, Yuki Asano, Kei Okada, Koji Kawasaki, and Masayuki Inaba. Component modularized design of musculoskeletal humanoid platform Musashi to investigate learning control system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7300–7307, 2019.
 16. Tasuku Makabe, Takuma Shirai, Yuya Nagamatsu, Kento Kawaharazuka, Fumihito Sugai, Kei Okada, and Masayuki Inaba. Development of joint module with two-speed gear transmission and joint lock mechanism during driving for task adaptable robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5123–5130, 2019.

国内学術講演会

17. 垣内洋平, 白井拓磨, 菅井文仁, 大久保壮一, 熊谷伊織, 永松祐弥, 岡田慧, 稲葉雅幸, 和田喜久男, 山崎信行. ディペンダブルなロボット用組込プロセッサ系の開発 - 組込みリアルタイムシステム用ディペンダブル SoC 及び SiP に関する基盤技術の研究成果報告 -. 第 32 回日本ロボット学会学術講演会講演論文集, pp. 1N1-05, 2014.
18. 大久保壮一, 白井拓磨, 永松祐弥, 熊谷伊織, 菅井文仁, 垣内洋平, 岡田慧, 稲葉雅幸, 溝谷圭悟, 久村雄輔, 山崎信行. 分散制御システム向け実時間通信 Responsive Link を備えた I/O Core SiP 基板による多自由度ヒューマノイド制御システムの構成と評価. 第 15 回 SICE システムインテグレーション部門講演会講演概要集, pp. 2453–2456, 2014.
19. 鈴木裕登, 永松祐弥, 白井拓磨, 野沢峻一, 垣内洋平, 岡田慧, 稲葉雅幸. 足首関節にトルクセンサを有した大出力 2 脚ロボットによる過剰関節負荷の緩和制御. 日本機械学会ロボティクス・メカトロニクス講演会'16 講演論文集, pp. 2A1-13a7, 2016.
20. 河原塚健人, 真壁佑, 牧野将吾, 都築敬, 永松祐弥, 浅野悠紀, 白井拓磨, 菅井文仁, 岡田慧, 稲葉雅幸. 環境接触を伴う学習型制御研究のための筋骨格型倒立二輪ロボッ

- トの開発. 第 36 回日本ロボット学会学術講演会講演論文集, pp. 1P2-02, 2018.
21. 河原塚健人, 牧野将吾, 都築敬, 鬼塚盛宇, 永松祐弥, 新城光樹, 真壁佑, 浅野悠紀, 岡田慧, 川崎宏治, 稲葉雅幸. 学習制御模索のためのモジュラー型筋骨格プラットフォームの設計開発. 日本機械学会ロボティクス・メカトロニクス講演会'19 講演論文集, pp. 2P1-C06, 2019.
 22. 真壁佑, 白井拓磨, 永松祐弥, 河原塚健人, 菅井文仁, 岡田慧, 稲葉雅幸. 用途適応型ロボットのための, 駆動時二段可変減速非駆動時ロック機構を持つ関節モジュールの設計開発. 日本機械学会ロボティクス・メカトロニクス講演会'19 講演論文集, pp. 2A2-F08, 2019.
 23. 浅野悠紀, 都築敬, 河原塚健人, 鬼塚盛宇, 古賀悠矢, 大村柚介, 永松祐弥, 真壁佑, 藤井綺香, 新城光樹, 中島慎介, 岡田慧, 川崎宏治, 稲葉雅幸. 腱駆動ヒューマノイドにおける認識判断操作統合に基づく自動車運転の実証実験 - ヒューマノイドによる自動車運転を通じたロボット技術の社会実装プロジェクト -. 第 37 回日本ロボット学会学術講演会講演論文集, pp. 3L2-06, 2019.
 24. 大村柚介, 河原塚健人, 永松祐弥, 都築敬, 鬼塚盛宇, 古賀悠矢, 西浦学, 浅野悠紀, 岡田慧, 川崎宏治, 稲葉雅幸. 外耳構造を有し音響処理を行う人体模倣ヒューマノイドの耳機構の設計開発. 日本機械学会ロボティクス・メカトロニクス講演会'20 講演論文集, pp. 1A1-E12, 2020.
 25. 真壁佑, 永松祐弥, 木村航平, 菅井文仁, 岡田慧, 稲葉雅幸. 腰部軸を備えた用途適応三輪車型パーソナルモビリティロボットの開発. 第 38 回日本ロボット学会学術講演会講演論文集, pp. 3C1-03, 2020.
 26. 浅野悠紀, 永松祐弥, 中島慎介, 岡田慧, 川崎宏治, 稲葉雅幸. 腱駆動ヒューマノイド要素技術を活用した義足プロトタイプの開発と車両ペダル操作における基礎的検討. 第 21 回 SICE システムインテグレーション部門講演会講演概要集, pp. 2993-2996, 2020.
 27. 大村柚介, 河原塚健人, 永松祐弥, 古賀悠矢, 西浦学, 浅野悠紀, 岡田慧, 川崎宏治, 稲葉雅幸. 人体外耳機構を模したヒューマノイドの両耳間スペクトル差学習に基づく空間音源方向推定システム. 第 21 回 SICE システムインテグレーション部門講演会講演概要集, pp. 1C3-17, 2020.

参考文献

- [1] Antoni Grau, Marina Indri, Lucia Lo Bello, and Thilo Sauter. Industrial robotics in factory automation: from the early stage to the internet of things. In *Annual Conference of the IEEE Industrial Electronics Society*, pp. 6159–6164, 2017.
- [2] Alessandro Gasparetto and Lorenzo Scalera. A brief history of industrial robotics in the 20th century. *Advances in Historical Studies*, Vol. 8, No. 1, pp. 24–35, 2019.
- [3] 浅間一. 平成のロボット工学を振り返る ~ロボット技術開発の動向と今後の展望~. *精密工学会誌*, Vol. 86, No. 1, pp. 23–27, 2020.
- [4] 渡邊恒文. 人間と相互理解できる次世代人工知能技術を目指した研究開発プロジェクトの取組み. *人工知能*, Vol. 34, No. 6, pp. 761–767, 2019.
- [5] 宮本和彦. ドローンが活躍する社会に向けた研究開発プロジェクトの取組み. *IATSS Review*, Vol. 44, No. 2, pp. 140–149, 2019.
- [6] 橋本周司, 成田誠之助, 白井克彦, 小林哲則, 高西淳夫, 菅野重樹, 笠原博徳. ヒューマノイド-人間形高度情報処理ロボット-. *情報処理*, Vol. 38, No. 11, pp. 959–969, 1997.
- [7] Md. Akhtaruzzaman and Amir A. Shafie. Advancement of android and contribution of various countries in the research and development of the humanoid platform. *International Journal of Robotics and Automation*, Vol. 1, No. 2, pp. 42–56, 2010.
- [8] Tamim Asfour, Julian Schill, Heiner Peters, Cornelius Klas, Jens Bückler, Christian Sander, Stefan Schulz, Artem Kargov, Tino Werner, and Volker Bartenbach. ARMAR-4: A 63 DOF torque controlled humanoid robot. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 390–396, 2013.
- [9] 水内郁夫. 人体構造に示唆を得た筋骨格型ヒューマノイドの構成と設計. *日本ロボット学会誌*, Vol. 28, No. 6, pp. 689–694, 2010.
- [10] 広瀬茂男, 佐藤幹夫. 多自由度ロボットの干渉駆動. *日本ロボット学会誌*, Vol. 7, No. 2, pp. 128–135, 1989.
- [11] Yuki Asano, Kei Okada, and Masayuki Inaba. Design principles of a human mimetic humanoid: Humanoid platform to study human intelligence and internal body system. *Science Robotics*, Vol. 2, No. 13, p. eaaq0899, 2017.
- [12] K. Kawaharazuka, K. Tsuzuki, S. Makino, M. Onitsuka, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba. Long-time self-body image acquisition and its application to the control of musculoskeletal structures. *IEEE Robotics and Automation Letters*, Vol. 4, No. 3, pp. 2965–2972, June 2019.
- [13] Kenji Hashimoto, Shunsuke Kimura, Nobuaki Sakai, Shinya Hamamoto,

- Ayanori Koizumi, Xiao Sun, Takashi Matsuzawa, Tomotaka Teramachi, Yuki Yoshida, Asaki Imai, Kengo Kumagai, Takanobu Matsubara, Koki Yamaguchi, Gan Ma, and Atsuo Takanishi. WAREC-1 - a four-limbed robot having high locomotion ability with versatility in locomotion styles. In *IEEE International Symposium on Safety, Security and Rescue Robotics*, pp. 172–178, 2017.
- [14] Moju Zhao, Tomoki Anzai, Fan Shi, Xiangyu Chen, Kei Okada, and Masayuki Inaba. Design, modeling, and control of an aerial robot DRAGON: A dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation. *IEEE Robotics and Automation Letters*, Vol. 3, No. 2, pp. 1176–1183, April 2018.
- [15] Kenji Kaneko, Hiroshi Kaminaga, Takeshi Sakaguchi, Shuuji Kajita, Mitsuharu Morisawa, Iori Kumagai, and Fumio Kanehiro. Humanoid robot HRP-5P: An electrically actuated humanoid robot with high-power and wide-range joints. *IEEE Robotics and Automation Letters*, Vol. 4, No. 2, pp. 1431–1438, 2019.
- [16] 浅野悠紀, 都築敬, 河原塚健人, 鬼塚盛宇, 古賀悠矢, 大村柚介, 永松祐弥, 真壁佑, 藤井綺香, 新城光樹, 中島慎介, 岡田慧, 川崎宏治, 稲葉雅幸. 腱駆動ヒューマノイドにおける認識判断操作統合に基づく自動車運転の実証実験. 第37回日本ロボット学会学術講演会講演論文集, pp. 3L2–06, sep 2019.
- [17] 中村仁彦. 非ホロノミック系制御研究の展望. 計測と制御, Vol. 36, No. 6, pp. 384–389, 1997.
- [18] 石川将人. 非ホロノミックシステムの制御 - 拘束条件の非線形性を活かす -. 日本ロボット学会誌, Vol. 27, No. 4, pp. 384–387, 2009.
- [19] 金広文男, 石綿陽一, 齋藤元, 赤地一彦, 宮森剛, 五十棲隆勝, 金子健二, 比留川博久. 実時間 Ethernet を用いたヒューマノイドの分散 I/O システム. 日本ロボット学会誌, Vol. 25, No. 3, pp. 466–477, 2007.
- [20] 小島邦生, 唐澤達史, 上月豊隆, 黒岩英則, 柚木崎創, 岩石智志, 石川達矢, 小山遼, 野田晋太郎, 植田亮平, 菅井文仁, 野沢峻一, 垣内洋平, 岡田慧, 稲葉雅幸. 高速大出力ヒューマノイドの研究用プラットフォーム JAXON の開発. 日本ロボット学会誌, Vol. 34, No. 7, pp. 458–467, 2016.
- [21] Yuki Asano, Toyotaka Kozuki, Soichi Ookubo, Masaya Kawamura, Shinsuke Nakashima, Takeshi Katayama, Iori Yanokura, Toshinori Hirose, Kento Kawaharazuka, Shogo Makino, Youhei Kakiuchi, Kei Okada, and Masayuki Inaba. Human mimetic musculoskeletal humanoid Kengoro toward real world physically interactive actions. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 876–883, November 2016.

- [22] 河原塚健人, 牧野将吾, 都築敬, 鬼塚盛宇, 永松祐弥, 新城光樹, 眞壁佑, 浅野悠紀, 岡田慧, 川崎宏治, 稲葉雅幸. 学習制御模索のためのモジュラー型筋骨格プラットフォームの設計開発. 日本機械学会ロボティクス・メカトロニクス講演会講演論文集, pp. 2P1–C06, jun 2019.
- [23] Kento Kawaharazuka, Shogo Makino, Kei Tsuzuki, Moritaka Onitsuka, Yuya Nagamatsu, Koki Shinjo, Tasuku Makabe, Yuki Asano, Kei Okada, Koji Kawasaki, and Masayuki Inaba. Component modularized design of musculoskeletal humanoid platform Musashi to investigate learning control systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7294–7301, nov 2019.
- [24] Tomoki Anzai, Moju Zhao, Shunichi Nozawa, Fan Shi, Kei Okada, and Masayuki Inaba. Aerial grasping based on shape adaptive transformation by HALO: Horizontal plane transformable aerial robot with closed-loop multilinks structure. In *IEEE International Conference on Robotics and Automation*, pp. 6990–6996, may 2018.
- [25] Tomoki Anzai, Moju Zhao, Masaki Murooka, Fan Shi, Kei Okada, and Masayuki Inaba. Design, modeling and control of fully actuated 2D transformable aerial robot with 1 DoF thrust vectorable link module. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2820–2826, nov 2019.
- [26] Md. Akhtaruzzaman and A. A. Shafie. Evolution of humanoid robot and contribution of various countries in advancing the research and development of the platform. In *International Conference on Control, Automation and Systems*, pp. 1021–1028, 2010.
- [27] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent ASIMO: System overview and integration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2478–2483, 2002.
- [28] Akihiko Nagakubo, Yasuo Kuniyoshi, and Gordon Cheng. The ETL-Humanoid system - a high-performance full-body humanoid system for versatile real-world interaction. *Advanced Robotics*, Vol. 17, No. 2, pp. 149–164, 2003.
- [29] Ill-Woo Park, Jung-Yup Kim, Seo-Wook Park, and Jun-Ho Oh. Development of humanoid robot platform KHR-2 (KAIST humanoid robot 2). *International Journal of Humanoid Robotics*, Vol. 2, No. 4, pp. 519–536, 2005.
- [30] Takashi Minato, Yuichiro Yoshikawa, Tomoyuki Noda, Shuhei Ikemoto, Hiroshi Ishiguro, and Minoru Asada. CB²: A child robot with biomimetic body for

- cognitive developmental robotics. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 557–562, 2007.
- [31] Sebastian Lohmeier, Thomas Buschmann, and Heinz Ulbrich. Humanoid robot LOLA. In *IEEE International Conference on Robotics and Automation*, pp. 775–780, 2009.
- [32] Johannes Engelsberger, Alexander Werner, Christian Ott, Bernd Henze, Maximo A. Roa, Gianluca Garofalo, Robert Burger, Alexander Beyer, Oliver Eiberger, Korbinian Schmid, and Alin Albu-Schäffer. Overview of the torque-controlled humanoid robot TORO. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 916–923, 2014.
- [33] Taejin Jung, Jeongsoo Lim, Hyoin Bae, Kang Kyu Lee, Hun-Min Joe, and Jun-Ho Oh. Development of the humanoid disaster response platform DRC-HUBO+. *IEEE Transactions on Robotics*, Vol. 34, No. 1, pp. 1–17, 2018.
- [34] Nicolaus A. Radford, Philip Strawser, Kimberly Hambuchen, Joshua S. Mehling, William K. Verdeyen, A. Stuart Donnan, James Holley, Jairo Sanchez, Vienny Nguyen, Lyndon Bridgwater, Reginald Berka, Robert Ambrose, Mason Myles Markee, N. J. Fraser-Chanpong, Christopher McQuin, John D. Yamokoski, Stephen Hart, Raymond Guo, Adam Parsons, Brian Wightman, Paul Dinh, Barrett Ames, Charles Blakely, Courtney Edmondson, Brett Sommers, Rochelle Rea, Chad Tobler, Heather Bibby, Brice Howard, Lei Niu, Andrew Lee, Michael Conover, Lily Truong, Ryan Reed, David Chesney, Robert Platt Jr., Gwendolyn Johnson, Chien-Liang Fok, Nicholas Paine, Luis Sentis, Eric Cousineau, Ryan Sinnet, Jordan Lack, Matthew Powell, Benjamin Morris, Aaron Ames, and Jide Akinyode. Valkyrie: NASA’s first bipedal humanoid robot. *Journal of Field Robotics*, Vol. 32, No. 3, pp. 397–419, 2015.
- [35] Coleman Knabe, Robert Griffin, James Burton, Graham Cantor-Cooke, Lakshitha Dantanarayana, Graham Day, Oliver Ebeling-Koning, Eric Hahn, Michael Hopkins, Jordan Neal, Jackson Newton, Chris Nogales, Viktor Orekhov, John Peterson, Michael Rouleau, John Seminatore, Yoonchang Sung, Jacob Webb, Nikolaus Wittenstein, Jason Zigar, Alexander Leonessa, Brian Lattimer, and Tomonari Furukawa. Team VALOR’s ESCHER: A novel electromechanical biped for the DARPA Robotics Challenge. *Journal of Field Robotics*, Vol. 34, No. 5, pp. 912–939, 2017.
- [36] Hiroshi Kaminaga, Tianyi Ko, Satoshi Yorita, Shunsuke Sato, Ryo Masumura, Mitsuo Komagata, Tatsuya Ishikawa, Taira Miyatake, and Yoshihiko Nakamura.

- Enhancement of mechanical strength, computational power, and heat management for fieldwork humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 786–793, 2016.
- [37] Yan Huang, Qingqing Li, Aiguo Ming, Yu Liu, Yaliang Liu, and Qiang Huang. Dynamic gait transition of a humanoid robot from hand-knee crawling to bipedal walking based on kinematic primitives. In *IEEE International Conference on Advanced Robotics and its Social Impacts*, pp. 240–245, 2019.
- [38] T. Otani, K. Hashimoto, S. Miyamae, H. Ueta, M. Sakaguchi, Y. Kawakami, H. O. Lim, and A. Takanishi. Angular momentum compensation in yaw direction using upper body based on human running. In *IEEE International Conference on Robotics and Automation*, pp. 4768–4775, 2017.
- [39] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V. G. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, L. Natale, E. Mingo Hoffman, H. Dallali, N. Kashiri, J. Malzahn, J. Lee, P. Kryczka, and D. Kanoulas. WALK-MAN: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, Vol. 34, No. 7, pp. 1225–1259, 2017.
- [40] Tamim Asfour, Mirko Wächter, Lukas Kaul, Samuel Rader, Pascal Weiner, Simon Ottenhaus, Raphael Grimm, You Zhou, Markus Grotz, and Fabian Paus. ARMAR-6: A high-performance humanoid for human-robot collaboration in real-world scenarios. *IEEE Robotics and Automation Magazine*, Vol. 26, No. 4, pp. 108–121, 2019.
- [41] Hugo Gravato Marques, Michael Jäntschi, Steffen Wittmeier, Owen Holland, Cristiano Alessandro, Alan Diamond, Max Lungarella, and Rob Knight. ECCE1: the first of a series of anthropomorphic musculoskeletal upper torsos. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 391–396, 2010.
- [42] Ryuma Niiyama and Yasuo Kuniyoshi. Design principle based on maximum output force profile for a musculoskeletal robot. *Industrial Robot*, Vol. 37, No. 3, pp. 250–255, 2010.
- [43] Paul Hebert, Max Bajracharya, Jeremy Ma, Nicolas Hudson, Alper Aydemir, Jason Reid, Charles Bergh, James Borders, Matthew Frost, Michael Hagman, John Leichty, Paul Backes, Brett Kennedy, Paul Karplus, Brian Satzinger, Katie Byl, Krishna Shankar, and Joel Burdick. Mobile manipulation and mobility as manipulation - design and algorithms of RoboSimian. *Journal of Field*

- Robotics*, Vol. 32, No. 2, pp. 255–274, 2015.
- [44] 玄相昊. 油圧による柔軟で機動性の高い多脚ロボットの実現. 日本ロボット学会誌, Vol. 37, No. 2, pp. 150–155, 2019.
- [45] Markus Grebenstein, Alin Albu-Schäffer, Thomas Bahls, Maxime Chalon, Oliver Eiberger, Werner Friedl, Robin Gruber, Sami Haddadin, Ulrich Hagn, Robert Haslinger, Hannes Höppner, Stefan Jörg, Mathias Nickl, Alexander Nothhelfer, Florian Petit, Josef Reill, Nikolaus Seitz, Thomas Wimböck, Sebastian Wolf, Tilo Wüsthoff, and Gerd Hirzinger. The DLR Hand Arm System. In *IEEE International Conference on Robotics and Automation*, pp. 3175–3182, 2011.
- [46] Shuuji Kajita, Takashi Nagasaki, Kenji Kaneko, and Hirohisa Hirukawa. ZMP-based biped running control. *IEEE Robotics and Automation Magazine*, Vol. 14, No. 2, pp. 63–72, 2007.
- [47] Toru Takenaka, Takashi Matsumoto, Takahide Yoshiike, Tadaaki Hasegawa, Shinya Shirokura, Hiroyuki Kaneko, and Atsuo Orita. Real time motion generation and control for biped robot -4th report: Integrated balance control-. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1601–1608, 2009.
- [48] Ryosuke Tajima, Daisaku Honda, and Keisuke Suga. Fast running experiments involving a humanoid robot. In *IEEE International Conference on Robotics and Automation*, pp. 1571–1576, 2009.
- [49] Baek-Kyu Cho, Jung-Hoon Kim, and Jun-Ho Oh. Online balance controllers for a hopping and running humanoid robot. *Advanced Robotics*, Vol. 25, No. 9-10, pp. 1209–1225, 2011.
- [50] 小島邦生, 小椎尾侑多, 石川達矢, 菅井文仁, 垣内洋平, 岡田慧, 稲葉雅幸. 関節駆動力とフレーム剛性に基づく大出力ヒューマノイド JAXON3-P の軽量化設計法と跳躍動作の実現. 日本ロボット学会誌, Vol. 38, No. 10, pp. 975–984, 2020.
- [51] Xuechao Chen, Wenxi Liao, Zhangguo Yu, Haoxiang Qi, Xinyang Jiang, and Qiang Huang. Motion coordination for humanoid jumping using maximized joint power. *Advances in Mechanical Engineering*, Vol. 13, No. 6, pp. 1–12, 2021.
- [52] 広瀬真人, 竹中透, 五味洋, 小澤信明. 人間型ロボット. 日本ロボット学会誌, Vol. 15, No. 7, pp. 983–985, 1997.
- [53] Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of Honda humanoid robot. In *IEEE International Conference on Robotics*

- and Automation*, pp. 1321–1326, 1998.
- [54] 稲葉雅幸. リモートブレインロボット. 計測と制御, Vol. 35, No. 4, pp. 286–290, 1996.
- [55] Jin'ichi Yamaguchi, Eiji Soga, Sadatoshi Inoue, and Atsuo Takanishi. Development of a bipedal humanoid robot - control method of whole body cooperative dynamic biped walking -. In *IEEE International Conference on Robotics and Automation*, pp. 368–374, 1999.
- [56] Hun ok Lim and Atsuo Takanishi. Biped walking robots created at Waseda University: WL and WABIAN family. *Philosophical Transactions of the Royal Society A*, Vol. 365, pp. 49–64, 2006.
- [57] Yu Ogura, Hiroyuki Aikawa, Kazushi Shimomura, Hideki Kondo, Akitoshi Morishima, Hun ok Lim, and Atsuo Takanishi. Development of a new humanoid robot WABIAN-2. In *IEEE International Conference on Robotics and Automation*, pp. 76–81, 2006.
- [58] 井上博允, 加賀美聡. ヒューマノイドロボットの知能ソフトウェア統合. 日本ロボット学会誌, Vol. 19, No. 7, pp. 830–833, 2001.
- [59] 五十棲隆勝, 川田忠裕. ヒューマノイドロボットの開発と未来. 日本ロボット学会誌, Vol. 20, No. 7, pp. 664–667, 2002.
- [60] Jung-Hoon Kim, Seo-Wook Park, Ill-Woo Park, and Jun-Ho Oh. Development of a humanoid biped walking robot platform KHR-1 - initial design and its performance evaluation. In *The Third IARP International Workshop on Humanoid and Human Friendly Robotics*, pp. 14–21, 2002.
- [61] Jung-Woo Heo, In-Ho Lee, and Jun-Ho Oh. Development of humanoid robots in HUBO laboratory, KAIST. 日本ロボット学会誌, Vol. 30, No. 4, pp. 367–371, 2012.
- [62] Qiang Huang, Kejie Li, and Tianmiao Wang. Control and mechanical design of humanoid robot BHR-01. In *The Third IARP International Workshop on Humanoid and Human Friendly Robotics*, pp. 10–13, 2002.
- [63] Qiang Huang, Zhangguo Yu, Xuechao Chen, Weimin Zhang, Tianqi Yang, Wenxi Liao, and Macro Ceccarelli. Historical developments of BHR humanoid robots. *Advances in Historical Studies*, Vol. 8, No. 1, pp. 79–90, 2019.
- [64] 五十棲隆勝, 赤池一彦, 平田勝, 金子健二, 梶田秀司, 比留川博久. ヒューマノイドロボット HRP-2 の開発. 日本ロボット学会誌, Vol. 22, No. 8, pp. 1004–1012, 2004.
- [65] 松井俊浩, 比留川博久, 石川裕, 山崎信行, 加賀美聡, 堀俊夫, 金広文男, 斎藤元, 稲邑哲也. ヒューマノイド・ロボットのための実時間分散情報処理. 情報処理学会研究報

- 告システム LSI 設計技術, Vol. 2004, No. 33, pp. 1–7, 2004.
- [66] F. Pfeiffer, K. Löffler, and M. Gienger. The concept of jogging JOHNNIE. In *IEEE International Conference on Robotics and Automation*, pp. 3129–3135, 2002.
- [67] Sebastian Lohmeier, Klaus Löffler, Michael Gienger, Heinz Ulbrich, and Friedrich Pfeiffer. Computer system and control of biped “johnnie”. In *IEEE International Conference on Robotics and Automation*, pp. 4222–4227, 2004.
- [68] 長久保晶彦, 國吉康夫. ヒューマノイド・インタラクション研究. 日本ロボット学会誌, Vol. 15, No. 7, pp. 979–982, 1997.
- [69] 國吉康夫, 大村吉幸, 寺田耕志, 長久保晶彦. 等身大ヒューマノイドロボットによるダイナミック起き上がり行動の実現. 日本ロボット学会誌, Vol. 23, No. 6, pp. 706–717, 2005.
- [70] Tamim Asfour, Karsten Berns, and Rüdiger Dillmann. The humanoid robot ARMAR: Design and control. In *IEEE-RAS International Conference on Humanoid Robots*, p. 06, 2000.
- [71] Rüdiger Dillmann, Regine Becher, and Peter Steinhaus. ARMAR II - a learning and cooperative multimodal humanoid robot system. *International Journal of Humanoid Robotics*, Vol. 1, No. 1, pp. 143–155, 2004.
- [72] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. ARMAR-III: An integrated humanoid platform for sensory-motor control. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 169–175, 2006.
- [73] Jun-Ho Oh, David Hanson, Won-Sup Kim, Il Young Han, Jung-Yup Kim, and Ill-Woo Park. Design of android type humanoid robot Albert HUBO. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1428–1433, 2006.
- [74] Zhangguo Yu, Qiang Huang, Jianxi Li, Qing Shi, Xuechao Chen, and Kejie Li. Distributed control system for a humanoid robot. In *IEEE International Conference on Mechatronics and Automation*, pp. 1166–1171, 2007.
- [75] Zhangguo Yu, Qiang Huang, Jianxi Li, Xuechao Chen, and Kejie Li. Computer control system and walking pattern control for a humanoid robot. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1018–1023, 2008.
- [76] Xuechao Chen, Qiang Huang, Zhangguo Yu, Jing Li, Wei Xu, Fei Meng, and Jiayu Liu. Design and experiment of an open control system for a humanoid

- robot. In *IEEE International Conference on Automation and Logistics*, pp. 367–372, 2010.
- [77] Zhangguo Yu, Qiang Huang, Gan Ma, Xuechao Chen, Weimin Zhang, Jing Li, and Junyao Gao. Design and development of the humanoid robot BHR-5. *Advances in Mechanical Engineering*, Vol. 6, p. 852937, 2014.
- [78] Fumio Kanehiro, Yoichi Ishiwata, Hajime Saito, Kazuhiko Akachi, Gou Miyamori, Takakatsu Isozumi, Kenji Kaneko, and Hirohisa Hirukawa. Distributed control system of humanoid robots based on real-time Ethernet. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2471–2477, 2006.
- [79] Kenji Kaneko, Kensuke Harada, Fumio Kanehiro, Go Miyamori, and Kazuhiko Akachi. Humanoid robot HRP-3. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2471–2478, 2008.
- [80] Kenji Kaneko, Fumio Kanehiro, Mitsuharu Morisawa, Kazuhiko Akachi, Go Miyamori, Atsushi Hayashi, and Noriyuki Kanehira. Humanoid robot HRP-4 - humanoid robotics platform with lightweight and slim body -. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4400–4407, 2011.
- [81] M. Fuchs, Ch. Borst, P. Robuffo Giordano, A. Baumann, E. Kraemer, J. Langwald, R. Gruber, N. Seitz, G. Plank, K. Kunze, R. Burger, F. Schmidt, T. Wimboeck, and G. Hirzinger. Rollin’ Justin – design considerations and realization of a mobile platform for a humanoid upper body. In *IEEE International Conference on Robotics and Automation*, pp. 4131–4137, 2009.
- [82] B. Gombert, G. Hirzinger, G. Plank, and M. Schedl. Modular concepts for a new generation of light weight robots. In *International Conference on Industrial Electronics, Control and Instrumentation*, pp. 1507–1514, 1994.
- [83] G. Hirzinger, A. Albu-Schäffer, M. Hähle, I. Schaefer, and N. Sporer. On a new generation of torque controlled light-weight robots. In *IEEE International Conference on Robotics and Automation*, pp. 3356–3363, 2001.
- [84] A. Albu-Schäffer, S. Haddadin, Ch. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger. The DLR lightweight robot: design and control concepts for robots in human environments. *Industrial Robot*, Vol. 34, No. 5, pp. 376–385, 2007.
- [85] Matthieu Destephe, Kenji Hashimoto, and Atsuo Takanishi. Emotional gait generation method based on emotion mental model - preliminary experiment

- with happiness and sadness -. In *International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 86–89, 2013.
- [86] 大谷拓也, Przemyslaw Kryczka, 遠藤信綱, 岸竜弘, 橋本健二, 高西淳夫. ヒューマノイドのためのモジュール化された多機能小型モータコントローラの開発. 第29回日本ロボット学会学術講演会, pp. 3K2–8, 2011.
- [87] Takuya Otani, Tatsuhiro Kishi, Przemyslaw Kryczka, Nobutsuna Endo, Kenji Hashimoto, and Atsuo Takanishi. Development of distributed control system and modularized motor controller for expressive robotic head. In *CISM-IFTToMM Symposium on Robot Design, Dynamics and Control*, pp. 183–190, 2012.
- [88] Seung-Joon Yi, Stephan McGill, Larry Vadakedathu, Qin He, Inyong Ha, Jeakwon Han, Hyunjong Song, Michael Rouleau, Dennis Hong, and Daniel D. Lee. THOR-OP humanoid robot for DARPA Robotics Challenge Trials 2013. In *International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 359–363, 2014.
- [89] Coleman Knabe, John Seminatore, Jacob Webb, Michael Hopkins, Tomonari Furukawa, Alexander Leonessa, and Brian Lattimer. Design of a series elastic humanoid for the DARPA Robotics Challenge. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 738–743, 2015.
- [90] Andrew Stuart Donnan and James J. Holley. Robonet: A data bus for distributed control systems. In *IEEE Aerospace Conference*, pp. 1–6, 2018.
- [91] Sebastian Lohmeier, Thomas Buschmann, Markus Schwienbacher, Heinz Ulbrich, and Friedrich Pfeiffer. Leg design for a humanoid walking robot. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 536–541, 2006.
- [92] Valerio Favot, Thomas Buschmann, Markus Schwienbacher, Alexander Ewald, and Heinz Ulbrich. The sensor-controller network of the humanoid robot LOLA. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 805–810, 2012.
- [93] Felix Sygulla, Robert Wittmann, Philipp Seiwald, Tobias Berninger, Arne-Christoph Hildebrandt, Daniel Wahrmann, and Daniel Rixen. An EtherCAT-based real-time control system architecture for humanoid robots. In *IEEE International Conference on Automation Science and Engineering*, pp. 483–490, 2018.
- [94] Iori Kumagai, Mitsuharu Morisawa, Takeshi Sakaguchi, Shin'ichiro Nakaoka, Kenji Kaneko, Hiroshi Kaminaga, Shuuji Kajita, Mehdi Benallegue, Rafael

- Cisneros, and Fumio Kanehiro. Toward industrialization of humanoid robots: Autonomous plasterboard installation to improve safety and efficiency. *IEEE Robotics and Automation Magazine*, Vol. 26, No. 4, pp. 20–29, 2019.
- [95] Neville Hogan. Impedance control: An approach to manipulation: Part I - theory. *Journal of Dynamic Systems, Measurement, and Control*, Vol. 107, pp. 1–7, 1985.
- [96] 示村悦二郎, 山中一雄. むだ時間を含むシステムの諸問題. 計測と制御, Vol. 19, No. 11, pp. 1051–1056, 1980.
- [97] 阿部直人, 延山英沢. むだ時間系の制御. 計測と制御, Vol. 42, No. 4, pp. 316–319, 2003.
- [98] 阿部直人, 延山英沢. むだ時間システム入門 1 - 伝達関数からのアプローチ -. 計測と制御, Vol. 44, No. 11, pp. 799–804, 2005.
- [99] 桑田龍一. 改良形限界感度法と PID; I-PD 制御の特性. 計測自動制御学会論文集, Vol. 23, No. 3, pp. 232–239, 1987.
- [100] 積際徹, 淵上康徳, 神吉厚之, 横川隆一, 吉田和信. 人間とロボットの協調作業系におけるロボットのインピーダンス制御の安定性解析に関する一考察. 日本機械学会論文集 C 編, Vol. 71, No. 707, pp. 2267–2272, 2005.
- [101] 木原康之, 林浩一郎, 藤平祥孝, 上野光, 村上弘記, 吉川恒夫, 永井清. 外部制御 PC の導入による産業用ロボットの応答特性整形. 日本ロボット学会誌, Vol. 37, No. 5, pp. 419–426, 2019.
- [102] 永松祐弥. 外力高速反応系を有する等身大ヒューマノイドの環境適応動作生成に関する研究. Master's thesis, 東京大学大学院, 2017.
- [103] 井原廣一. リアルタイムシステム: 1. リアルタイムシステムとは. 情報処理, Vol. 35, No. 1, pp. 12–17, 1994.
- [104] Frédéric Py and Félix Ingrand. Dependable execution control for autonomous robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1136–1141, 2004.
- [105] Martin Wollschlaeger, Thilo Sauter, and Jürgen Jasperneite. The future of industrial communication: Automation networks in the era of the internet of things and Industry 4.0. *IEEE Industrial Electronics Magazine*, Vol. 11, No. 1, pp. 17–27, 2017.
- [106] 元吉伸一. 産業用デジタル通信の変遷と将来展望. 計測と制御, Vol. 57, No. 1, pp. 24–28, 2018.
- [107] Jin Ho Kim, Suk-Hyun Seo, Nguyen-Tien Hai, Bo Mu Cheon, Young Seo Lee, and Jae Wook Jeon. Gateway framework for in-vehicle networks based on CAN,

- FlexRay, and Ethernet. *IEEE Transactions on Vehicular Technology*, Vol. 64, No. 10, pp. 4472–4486, 2015.
- [108] Tetsuya Taira, Nobuhide Kamata, and Nobuyuki Yamasaki. Design and implementation of reconfigurable modular humanoid robot architecture. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3566–3571, 2005.
- [109] Nobuyuki Yamasaki. Responsive Link for distributed real-time processing. In *International Workshop on Innovative Architecture for Future Generation Processors and Systems*, pp. 20–29, 2007.
- [110] Muhammad Nabeel Asghar, Muhammad Umar Chaudhry, Najeeb Ul Hussan, Muhammad Yasir, and Sumayya Bibi. A review of ARM processor architecture history, progress and applications. *Journal of Applied and Emerging Sciences*, Vol. 10, No. 2, pp. 171–179, 2020.
- [111] Nitin Singhal, Jin Woo Yoo, Ho Yeol Choi, and In Kyu Park. Implementation and optimization of image processing algorithms on embedded GPU. *IEICE Transactions on Information and Systems*, Vol. E95-D, No. 5, pp. 1475–1484, 2012.
- [112] Valery Sklyarov, Iouliia Skliarova, João Silva, and Alexander Sudnitson. Analysis and comparison of attainable hardware acceleration in all programmable systems-on-chip. In *Euromicro Symposium on Digital System Design*, pp. 345–352, 2015.
- [113] Bertalan Kovács, Anders D. Henriksen, Jonathan Dyssel Stets, and Laaros Nalpantidis. Object detection on TPU accelerated embedded devices. In *International Conference on Computer Vision Systems*, pp. 82–92, 2021.
- [114] Qizhi Wang and ZiRu Wang. Research on deploying the deeplearning models with embedded devices. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, pp. 1337–1341, 2019.
- [115] Congjun Liu. YOLOv2 acceleration using embedded GPU and FPGAs: pros, cons, and a hybrid method. *Evolutionary Intelligence*, 2021.
- [116] Fynn Schwiegelshohn, Florian Kästner, and Michael Hübner. FPGA design of numerical methods for the robotic motion control task exploiting high-level synthesis. In *IEEE International Conference on the Science of Electrical Engineering*, pp. 1–5, 2016.
- [117] Shiqi Lian, Yinhe Han, Ying Wang, Yungang Bao, Hang Xiao, Xiaowei Li, and Ninghui Sun. Dadu: Accelerating inverse kinematics for high-DOF robots. In

- ACM/EDAC/IEEE Design Automation Conference*, pp. 1–6, 2017.
- [118] Ayaz Akram and Lina Sawalha. A study of performance and power consumption differences among different ISAs. In *Euromicro Conference on Digital System Design*, pp. 628–632, 2019.
- [119] 小池汎平. 計算デバイス戦国時代 ~ 信長 CPU と秀吉 GPU と家康 FPGA ~. *電気学会誌*, Vol. 137, No. 6, pp. 369–372, 2017.
- [120] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, Vol. 5, pp. 287–326, 1979.
- [121] 須田礼仁. ヘテロ並列計算環境のためのタスクスケジューリング手法のサーベイ. *情報処理学会論文誌: コンピューティングシステム*, Vol. 47, No. SIG18(ACS16), pp. 92–114, 2006.
- [122] 有本卓. ロボティクスと脳科学. *計測と制御*, Vol. 46, No. 10, pp. 765–771, 2007.
- [123] Junjun Li, Zhijun Li, Fei Chen, Antonio Bicchi, Yu Sun, and Toshio Fukuda. Combined sensing, cognition, learning, and control for developing future neuro-robotics systems: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, Vol. 11, No. 2, pp. 148–161, 2019.
- [124] Christian Ott, Bernd Henze, Georg Hettich, Tim Niklas Seyde, Máximo A. Roa, Vittorio Lippi, and Thomas Mergner. Good posture, good balance: Comparison of bioinspired and model-based approaches for posture control of humanoid robots. *IEEE Robotics and Automation Magazine*, Vol. 23, No. 1, pp. 22–33, 2016.
- [125] Thomas Mergner and Vittorio Lippi. Posture control - human-inspired approaches for humanoid robot benchmarking: Conceptualizing tests, protocols and analyses. *Frontiers in Neurorobotics*, Vol. 12, No. 21, pp. 1–16, 2018.
- [126] Marie Claire Capolei, Emmanouil Angelidis, Egidio Falotico, Henrik Hautop Lund, and Silvia Tolu. A biomimetic control method increases the adaptability of a humanoid robot acting in a dynamic environment. *Frontiers in Neuro-robotics*, Vol. 13, No. 70, pp. 1–18, 2019.
- [127] Marie Claire Capolei, Nils Axel Andersen, Henrik Hautop Lund, Egidio Falotico, and Silvia Tolu. A cerebellar internal models control architecture for online sensorimotor adaptation of a humanoid robot acting in a dynamic environment. *IEEE Robotics and Automation Letters*, Vol. 5, No. 1, pp. 80–87, 2020.
- [128] Ignacio Abadía, Francisco Naveros, Eduardo Ros, Richard Rafael Carrillo, and Niceto Rafael Luque. A cerebellar-based solution to the nondeterministic time

- delay problem in robotic control. *Science Robotics*, Vol. 6, No. 58, p. eabf2756, 2021.
- [129] 丹治順. 高次運動野による行動と運動の認知的制御. *理学療法学*, Vol. 30, No. 3, pp. 69–73, 2003.
- [130] 渡邊裕文. 中枢神経系の機能解剖 - 運動出力系 -. *関西理学療法*, Vol. 5, pp. 23–29, 2005.
- [131] 後藤淳. 中枢神経系の機能解剖 - 感覚入力系 -. *関西理学療法*, Vol. 5, pp. 11–21, 2005.
- [132] 畠山一平. 生体におけるフィードバック調整. *計測と制御*, Vol. 2, No. 2, pp. 81–90, 1963.
- [133] Masao Ito. *The Cerebellum and Neural Control*. Raven Press, 1984.
- [134] 五味裕章. 小脳が実現する滑らかな運動. *電気学会誌*, Vol. 115, No. 12, pp. 782–785, 1995.
- [135] Noriaki Ando, Takashi Suehiro, Kosei Kitagaki, Tetsuo Kotoku, and Woo-Keun Yoon. RT-Middleware: Distributed component middleware for RT (robot technology). In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3933–3938, 2005.
- [136] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
- [137] 岡田慧. ROS (ロボット・オペレーティング・システム). *日本ロボット学会誌*, Vol. 30, No. 9, pp. 830–835, 2012.
- [138] 岡田慧. ROS, 5年経って. *日本ロボット学会誌*, Vol. 35, No. 4, pp. 270–273, 2017.
- [139] 清水正晴, 戸田健吾, 林原靖男, 大和秀彰, 古田貴之. Linux 標準機能を利用した RT ミドルウェア周期実行機能のリアルタイム化- ハプティックジョイスティックによる全方位移動電動車椅子操縦システムへの適用-. *計測自動制御学会論文誌*, Vol. 46, No. 1, pp. 16–23, 2010.
- [140] 安藤慶昭. ロボットミドルウェアの実時間処理・排他処理の比較. *日本ロボット学会誌*, Vol. 34, No. 6, pp. 366–369, 2016.
- [141] 水川真. 産業機器・ロボット標準ネットワークインタフェース: ORiN. *日本ロボット学会誌*, Vol. 21, No. 6, pp. 585–590, 2003.
- [142] 菅谷みどり, 松原豊, 住谷拓馬, 中野美由紀. IXM: ロボット制御ソフトウェア向けプロセス間通信ミドルウェア. *情報処理学会論文誌*, Vol. 58, No. 10, pp. 1578–1590, 2017.
- [143] Shintaro Komatsu, Yuya Nagamatsu, Tatsuya Ishikawa, Takuma Shirai, Ku-

- nio Kojima, Yohei Kakiuchi, Fumihito Sugai, Kei Okada, and Masayuki Inaba. Humanoid robot's force-based heavy manipulation tasks with torque-controlled arms and wrist force sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3055–3062, November 2019.
- [144] Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimmering, Stefan Schaal, and Ludvic Righetti. Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, Vol. 40, No. 3, pp. 473–491, 2016.
- [145] O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Sourères, N. Mansard, F. Lamiroux, J.-P. Laumond, L. Marchionni, H. Tome, and F. Ferro. TALOS: A new humanoid research platform targeted for industrial applications. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 689–695, 2017.
- [146] Tianyi Ko, Ko Yamamoto, Kazuya Murotani, and Yoshihiko Nakamura. Compliant biped locomotion of Hydra, an electro-hydrostatically driven humanoid. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 587–592, 2018.
- [147] 浦田順一, 中西雄飛, 岡田慧, 稲葉雅幸. 高速・高トルク動作のための大出力 2 脚ロボットの開発. *日本ロボット学会誌*, Vol. 28, No. 7, pp. 865–871, 2010.
- [148] Kunio Kojima, Tatsuhi Karasawa, Toyotaka Kozuki, Eisoku Kuroiwa, Sou Yukizaki, Satoshi Iwaishi, Tatsuya Ishikawa, Ryo Koyama, Shintaro Noda, Fumihito Sugai, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Development of life-sized high-power humanoid robot JAXON for real-world use. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 838–843, November 2015.
- [149] Yuki Asano, Toyotaka Kozuki, Soichi Ookubo, Koji Kawasaki, Takuma Shirai, Kohei Kimura, Kei Okada, and Masayuki Inaba. A sensor-driver integrated muscle module with high-tension measurability and flexibility for tendon-driven robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5960–5965, Oct 2015.
- [150] Tomoki Anzai, Moju Zhao, Xiangyu Chen, Fan Shi, Koji Kawasaki, Kei Okada, and Masayuki Inaba. Multilinked multirotor with internal communication system for multiple objects transportation based on form optimization method. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5977–5984, september 2017.

- [151] 白井拓磨. 高速応答行動可能なロボットのための体内分散通信制御系に関する研究. PhD thesis, 東京大学大学院, 2017.
- [152] Takuma Shirai, Yuya Nagamatsu, Hiroto Suzuki, Shunichi Nozawa, Kei Okada, and Masayuki Inaba. Design and evaluation of torque based bipedal walking control system that prevent fall over by impulsive disturbance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 739–746, October 2018.
- [153] Christian Ott, Christoph Baumgärtner, Johannes Mayr, Matthias Fuchs, Robert Burger, Dongheui Lee, Oliver Eiberger, Alin Albu-Schäffer, Markus Grebenstein, and Gerd Hirzinger. Development of a biped robot with torque controlled joints. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 167–173, 2010.
- [154] Shuuji Kajita, Rafael Cisneros, Mehdi Benallegue, Takeshi Sakaguchi, Shin'ichiro Nakaoka, Mitsuharu Morisawa, Kenji Kaneko, and Fumio Kanehiro. Impact acceleration of falling humanoid robot with an airbag. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 637–643, 2016.
- [155] 加賀美聡. ロボットシステム開発のための実時間 os “ART-Linux”. *日本ロボット学会誌*, Vol. 31, No. 3, pp. 222–225, 2013.
- [156] John M. Hollerbach. A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 10, No. 11, pp. 730–736, 1980.
- [157] Shunichi Nozawa, Eisoku Kuroiwa, Kunio Kojima, Ryohei Ueda, Masaki Murooka, Shintaro Noda, Iori Kumagai, Yu Ohara, Yohei Kakiuchi, Kei Okadad, and Masayuki Inaba. Multi-layered real-time controllers for humanoid's manipulation and locomotion tasks with emergency stop. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 381–388, November 2015.
- [158] 川崎晴久. *ロボット工学の基礎*. 森北出版, 第 2 版第 1 刷, 2012.
- [159] Luc Le Tien, Alin Albu-Schäffer, Alessandro De Luca, and Gerd Hirzinger. Friction observer and compensation for control of robots with joint torque measurement. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3789–3795, 2008.
- [160] Min Jun Kim, Fabian Beck, Christian Ott, and Alin Albu-Schaffer. Model-free friction observers for flexible joint robots with torque measurements. *IEEE Transactions on Robotics*, Vol. 35, No. 6, pp. 1508–1515, 2019.

- [161] Alin Albu-Schäffer and Gerd Hirzinger. State feedback controller for flexible joint robots: A globally stable approach implemented on DLR's light-weight robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1087–1093, 2000.
- [162] Yohei Kakiuchi, Kunio Kojima, Eisoku Kuroiwa, Shintaro Noda, Masaki Murooka, Iori Kumagai, Ryohei Ueda, Fumihito Sugai, Shunichi Nozawa, Kei Okada, and Masayuki Inaba. Development of humanoid robot system for disaster response through team NEDO-JSK's approach to DARPA Robotics Challenge Finals. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 805–810, 2015.
- [163] 金広文男, 藤原清司, 梶田秀司, 横井一仁, 金子健二, 比留川博久, 中村仁彦, 山根克. ヒューマノイドロボットソフトウェアプラットフォーム OpenHRP. 日本ロボット学会誌, Vol. 21, No. 7, pp. 785–793, 2003.
- [164] 中岡慎一郎, 山野辺夏樹, 比留川博久, 山根克, 川角祐一郎. 分散コンポーネント型ロボットシミュレータ OpenHRP3. 日本ロボット学会誌, Vol. 26, No. 5, pp. 399–406, 2008.
- [165] Shin'ichiro Nakaoka, Fumio Kanihiro, Kanako Miura, Mitsuharu Morisawa, Kiyoshi Fujiwara, Kenji Kaneko, Shuuji Kajita, and Hirohisa Hirukawa. Creating facial motions of cybernetic human HRP-4C. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 561–567, 2009.
- [166] Ikuo Mizuuchi, Tamaki Nishino, Yoshinao Sodeyama, Yuto Nakanishi, Yuta Namiki, Tomoaki Yoshikai, and Masayuki Inaba. An autonomous reactive system for humanoids equipped with very many actuators and sensors. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 422–427, 2007.
- [167] 中村仁彦, 花房秀郎. 関節形ロボットアームの特異点低感度運動分解. 計測自動制御学会論文集, Vol. 20, No. 5, pp. 453–459, 1984.
- [168] Yoshihiko Nakamura and Hideo Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control*, Vol. 108, No. 3, pp. 163–171, 1986.
- [169] Zheng Yili, Sun Hanxu, Jia Qingxuan, and Shi Guozhen. Kinematics control for a 6-DOF space manipulator based on ARM processor and FPGA co-processor. In *IEEE International Conference on Industrial Informatics*, pp. 129–134, 2008.
- [170] Min-Kuang Wu, Ying-Shieh Kung, Yi-Hsien Huang, and Tz-Han Jung. Fixed-point computation of robot kinematics in FPGA. In *International Conference on Advanced Robotics and Intelligent Systems*, pp. 35–40, 2014.

- [171] Ziyang Jiang, Yu Dai, Jianxun Zhang, and Su He. Kinematics calculation of minimally invasive surgical robot based on FPGA. In *IEEE International Conference on Robotics and Biomimetics*, pp. 1726–1730, 2017.
- [172] Carlo Sau, Tiziana Fanni, Claudio Rubattu, Luca Fanni, Luigi Raffo, and Francesca Palumbo. Feasibility study and porting of the damped least square algorithm on FPGA. *IEEE Access*, Vol. 8, pp. 175483–175500, 2020.
- [173] Kunio Kojima, Yuta Kojio, Tatsuya Ishikawa, Fumihito Sugai, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Drive-train design in JAXON3-P and realization of jump motions: Impact mitigation and force control performance for dynamic motions. pp. 3747–3753, October 2020.
- [174] 灰屋和勇, 駒田諭, 平井淳之. 腱駆動機構における非線形バネの特性式誤差補償による張力制御. *電気学会論文誌 D(産業応用部門誌)*, Vol. 130, No. 6, pp. 816–823, 2010.
- [175] 加古川篤, 川村貞夫, 武居直行, 徳永晋也, 深澤俊樹, 山本章, 徳田貴司, 栗本直彰. 機械的接触を基盤とするロボットの事業化に向けて - 低摩擦ギアード電動モータからのアプローチ -. *日本ロボット学会誌*, Vol. 39, No. 2, pp. 125–131, 2021.
- [176] Hiroto Suzuki, Yuya Nagamatsu, Takuma Shirai, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Torque based stabilization control for torque sensorless humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 425–431, November 2017.
- [177] Yasuhiro Ishiguro, Tasuku Makabe, Yuya Nagamatsu, Yuta Kojio, Kunio Kojima, Fumihito Sugai, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit TABLIS. *IEEE Robotics and Automation Letters*, Vol. 5, No. 4, pp. 6419–6426, August 2020.
- [178] 榊泰輔, 舘暲. インピーダンス制御の接触安定性についての一考察. *日本ロボット学会誌*, Vol. 12, No. 3, pp. 489–496, 1994.
- [179] Yoshito Ito, Takuya Nakaoka, Junichi Urata, Kazuya Kobayashi, Shunich Nozawa, Yuto Nakanishi, Kei Okada, and Masayuki Inaba. Development and verification of life-size humanoid with high-output actuation system. In *IEEE International Conference on Robotics and Automation*, pp. 3433–3438, may 2014.
- [180] Yoshito Ito, Takuya Nakaoka, Junichi Urata, Yuto Nakanishi, Kei Okada, and Masayuki Inaba. Design and development of a tendon-driven and axial-driven hybrid humanoid leg with high-power motor driving system. In *IEEE-RAS*

- International Conference on Humanoid Robots*, pp. 475–480, 11 2012.
- [181] Moritaka Onitsuka, Manabu Nishiura, Kento Kawaharazuka, Kei Tsuzuki, Yasunori Toshimitsu, Yusuke Omura, Yuki Asano, Kei Okada, Koji Kawasaki, and Masayuki Inaba. Development of musculoskeletal legs with planar interskeletal structures to realize human comparable moving function. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 17–24, July 2021.
- [182] Kento Kawaharazuka, Tasuku Makabe, Shogo Makino, Kei Tsuzuki, Yuya Nagamatsu, Yuki Asano, Takuma Shirai, Fumihito Sugai, Kei Okada, Koji Kawasaki, and Masayuki Inaba. TWIMP: Two-wheel inverted musculoskeletal pendulum as a learning control platform in the real world with environmental physical contact. In *IEEE-RAS International Conference on Humanoid Robots*, pp. 784–790, November 2018.
- [183] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, Vol. 3, No. 3, pp. 1957–1964, 2018.
- [184] Hossein Bonyan Khamseh, Farrokh Janabi-Sharifi, and Abdelkader Adbessameud. Aerial manipulation - a literature survey. *Robotics and Autonomous Systems*, Vol. 107, pp. 221–235, 2018.
- [185] Moju Zhao, Koji Kawasaki, Xiangyu Chen, Shintaro Noda, Kei Okada, and Masayuki Inaba. Whole-body aerial manipulation by transformable multirotor with two-dimensional multilinks. In *IEEE International Conference on Robotics and Automation*, pp. 5175–5182, may 2017.
- [186] Moju Zhao, Fan Shi, Tomoki Anzai, Kei Okada, and Masayuki Inaba. Online motion planning for deforming maneuvering and manipulation by multilinked aerial robot based on differential kinematics. *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, pp. 1602–1609, April 2020.
- [187] Fan Shi, Moju Zhao, Masaki Murooka, Kei Okada, and Masayuki Inaba. Aerial regrasping: Pivoting with transformable multilink aerial robot. In *IEEE International Conference on Robotics and Automation*, pp. 200–207, june 2020.
- [188] 中野栄二, 大久保宏樹. 跳躍ロボット全般について. *日本ロボット学会誌*, Vol. 11, No. 3, pp. 40–45, 1993.
- [189] Sang-Ho Hyon and Takashi Emura. Energy-preserving control of a passive one-legged running robot. *Advanced Robotics*, Vol. 18, No. 4, pp. 357–381, 2004.
- [190] Yanheng Zhang, Lufeng, Zhang, Wei Wang, Yangmin Li, and Qingwen Zhang. Design and implementation of a two-wheel and hopping robot with a linkage

- mechanism. *IEEE Access*, Vol. 6, pp. 42422–42430, 2018.
- [191] Justin K. Yim and Ronald S. Fearing. Precision jumping limits from flight-phase control in Salto-1P. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2229–2236, 2018.
- [192] 山崎貴大, 近藤良, 高柳克哉, 北山文矢. 周期離散時間系に基づく一脚ホッピングロボットの姿勢安定化制御. *日本ロボット学会誌*, Vol. 39, No. 8, pp. 735–743, 2021.

以上

1p – 316p 完

博士論文

令和3年12月3日提出

東京大学大学院 情報理工学系研究科

知能機械情報学専攻

48177507 永松 祐弥