

事例紹介

ロボットの運動計算・解析からハードウェア制御まで支える基盤ソフトウェアライブラリの自作

Scratchbuilding Software Libraries to Support from Robot Motion Computation to Hardware Control

杉原 知道* *大阪大学

Tomomichi Sugihara* *Osaka University

1. はじめに

筆者の研究グループでは、「人と人型ロボットの双方向理解」を大目標とし、ロボットの運動計算 [1]~[3], 制御 [4]~[10], 経路・軌道計画 [11]~[15], 操縦 [16] [17], 人の運動計測と解析 [18] [19], ロボットのハードウェア開発 [20], ソフトウェアシステム設計 [21] 等の研究を行っている。運動学や力学を核としてこれらの課題を結び付け、工学的方法による人の行動理解と高機能な人型ロボットの開発を相乗的に進めることが狙いである。恐らく大半のロボット研究者にとってそうであるように、研究活動における実質作業の大部分はソフトウェア開発である。したがって、各課題の成果を効果的に結合するためには、基盤となる計算資源(ライブラリ)の整備が何にも先んじて重要と考えた。

人型ロボットは形状・機構・力学の複雑さから高度な計算が要求されるものの、計算内容自体はそれほど特殊なものではない。しかし、オフラインでの運動計算・解析からオンラインでのハードウェア制御までシームレスな計算が求められる点に、固有の難しさがあると考えた。これを解決するために、筆者は独自にライブラリを製作してきた。このような方法は、既存のライブラリやミドルウェアを活用し研究開発の効率を上げるノウハウの紹介、という本特集号の趣旨には必ずしもそぐわない。また筆者自身、この方法が絶対的に優れているとは考えていない。一方で、日々の研究活動において自作ライブラリの良さを感じる場面はやはり多くある。本稿では、その長所・短所を論じながら、稚拙な手作りライブラリを用いた筆者のソフトウェア開発の事例を紹介させていただきたい。

2. 運動計算・解析からハードウェア制御まで応用可能な基盤計算資源の要件

前述のとおり、ロボットの計算基盤は、高度な幾何学・運動学・力学計算が可能であり、かつハードウェアを意識し

たものでなければならない。具体的には、次のような性質が要求される。

(1) 実時間性

物理法則に支配される実世界の現象とループを形成する情報処理において、処理の終了時刻が厳密に保証されることは何よりも重要な性質である。スケジューラの実時間性とは別に、個々の計算処理も明確な命令数の上限を持たなければならない。数値計算には反復解法を伴うことも多いため、この要求はライブラリ実装の根幹にかかわる。

(2) ロバスト性

ソフトウェアが原因で、ロボットの挙動を不安定にしてはならない。時間制約の中で、厳密性を犠牲にしても物理的に安全な結果、例えば連続性や微分可能性が保証されている結果を出さなければならない。

(3) 軽量性

ハードウェアと最も強く関係する組み込みソフトウェアは、大抵の場合はエンドユーザ用計算機よりも非力な計算機の上での動作が求められる。そのような場合、巨大なメモリ使用を前提とするライブラリはそもそも使えない。

(4) 並列計算や割り込みに対する安全性

組み込みソフトウェアにおいてもオフライン解析においても、並列計算は今や常識的に行われている。また実世界の事象に縛られるシステムでは、頻繁に割り込みが発生する。このため、少なくとも内部計算はスレッドセーフかつリエントラントでなければならない。

(5) プラットフォーム互換性

ロボットの実装形態、運用形態は極めて多様であるので、基本的な幾何学計算・運動計算においては特定の開発環境や運用環境を前提としないことが好ましい。

ハードウェアにかかわる性質とは別に、一般的なライブラリへの要求として再利用性も重要で、筆者の経験では、異なる処理間でデータをどのように受け渡すかが要点である。

3. ブラックボックス／ホワイトボックスアプローチによる計算環境構築

金広ら [22] は、肥大化が避けられないロボットシステム

原稿受付 2012年8月17日

キーワード: Robot Software, Scratchbuild, Kinodynamical Computation

*〒565-0871 吹田市山田丘 2-1

*Suita-shi, Osaka

の細部に到るまで独自に実装するのは非合理的である、という考えに基づき、他者の提供するソフトウェアを粗に結合していくことで大規模なシステムを発展的に構築する方法を提案している。おのおののソフトウェアの保守はベンダに一任し、システム開発者の関心をソフトウェア間の結合部分のみに集中できることが利点である。しかしこの方法には、次のような欠点もある。

- (i) 前章に述べた、ハードウェア制御に適した性質を有するソフトウェアはそもそも少ない。
- (ii) ベンダの提供するソフトウェアは、部品としては過度に高機能である場合が多く、軽量化の要求と相容れない。
- (iii) ベンダが異なればデータ型も相異なるため、ソフトウェアを結合するための労力は決して小さくない。その上、全体として一貫性のない継ぎ接ぎシステムとなり、拡張性・保守性が著しく低下する。
- (iv) ソフトウェアの保守をベンダに一任せざるを得ない状況は、開発工程の管理が困難な点で決して望ましいものではない。ベンダ側の事情で保守が終了するソフトウェアも多い。またバグが発生したとき、原因を突き止めるのが困難である。

以上のことから、他者の開発したソフトウェアが部品として適当か否かは十分吟味する必要がある。筆者は結局、基盤計算ライブラリのほぼすべてを自前で開発することを選択し、上記の欠点をすべて回避した。処理のどこで何が起きているのか完全に把握できている状況は、アプリケーションの開発効率を大いに高めている。その一方で、次のような問題も実感している。

- (i) 開発には極めて時間が掛かる。筆者の場合、現在の形に整備するまでに10年以上掛けている。商用製品ではない気楽さから、筆者はこれを半分以上楽しみながら行っているが、一般には通用しないだろう。
- (ii) 各計算は、おのおのの専門家が実装したソフトウェアには性能的に当然劣る。軽量性や再利用性の維持に努めた結果、多くの部分でハッシュ等を利用した高速化を諦めている。最適化もコンパイラに任せることになる。
- (iii) データ型の一貫性を維持しながらの機能拡張は大元からの再設計をしばしば伴うが、これは利用者の立場からすれば不便である。筆者の場合、これまでに60回強のメジャーバージョンを重ね、その際に後方互換性を犠牲にしたことも度々ある。このため、自分のライブラリを他者に利用していただくことは今でも逡巡している。
- (iv) そもそもこのようなライブラリ開発は、重要性に反して実績として極めて認められにくい。開発時間と労力に見合う成果が出るまで忍耐が必要である。

ロボット研究用基盤ソフトウェアの開発事例は幾つかある[23]~[28]が、同様の苦勞を抱えているのではないかと

想像する。あるいは筆者には思いもつかない方法で解決されている事例があるだろうか。読者諸兄から情報・ご意見をお寄せいただければ幸甚である。

4. 基盤ライブラリ群の階層的設計

図1に、筆者の使用しているライブラリ群の構成と、それらの上に開発されるアプリケーションのイメージを示す。灰色部が、独自製作した部分である。部品間を糊付けする基本データ型とアルゴリズムを定義したライブラリを最下層に配置し、その上に数値解析・最適化ライブラリ、計算幾何学ライブラリと自動制御ライブラリ、さらに計算幾何学の上にロボットの運動学・力学計算ライブラリをブロックのように組み上げている。また、OSに依存するプロセス間通信やプラグイン処理等を行うためのシステムプログラミングライブラリと、ウィンドウシステムやレンダリングエンジンに依存するグラフィカルツールキット、可視化ライブラリを除いて、完全にプラットフォーム非依存であり、実際にSolaris, Linux, FreeBSD, Windows XP等で動作実績がある。ライブラリを構成する部品を含め、各要素は自分より下位の階層にある要素のみ利用を許す、というルールを設け、保守性の維持に努めている。

ライブラリ名の隣にある括弧内の数字はコードの行数を示しており、「運動学・力学を核とする」と言いながら、実態としてはそれらを支える数値計算、最適化、計算幾何学の分量のほうが大きいことがお分かりいただけるかと思う。なお、開発言語にはC(ANSI-C準拠)を採用している。これは当初、RTLlinux [29]で本ライブラリを使用しており、コードレベルでLinuxカーネルモジュールにそのまま利用できることを開発方針の一つに掲げていたことに起因する。現在は標準Linuxカーネルもある程度リアルタイムシステムに対応し、カーネル空間で高度な計算を行う必要はなくなった。一方、組み込み計算機のための軽量性やプラットフォーム互換性、システムプログラミングとの相性、筆者が可視化に用いているXlibやOpenGLとの相性等の要求が依然あり、多言語への移行は検討していない。また、コンパイラにも依存するが、基本的にCの名前解決はC++のそれよりも単純でトラブルが少ない。このことによる恩恵は、C++で開発経験のある方にはご共感いただけるのではないだろうか。その負の側面として、大規模ソフトウェアの開発には慎重さを要する。引数付きマクロ、関数ポインタ、型不定ポインタを多用し、技巧的ではあるがオブジェクト指向な実装をしている。

5. アプリケーション開発事例

前章で示したライブラリ群の上に開発したアプリケーションソフトウェアの例を幾つか示そう。

図2は、単位ベクトル法[30]を用いた順動力学シミュレー

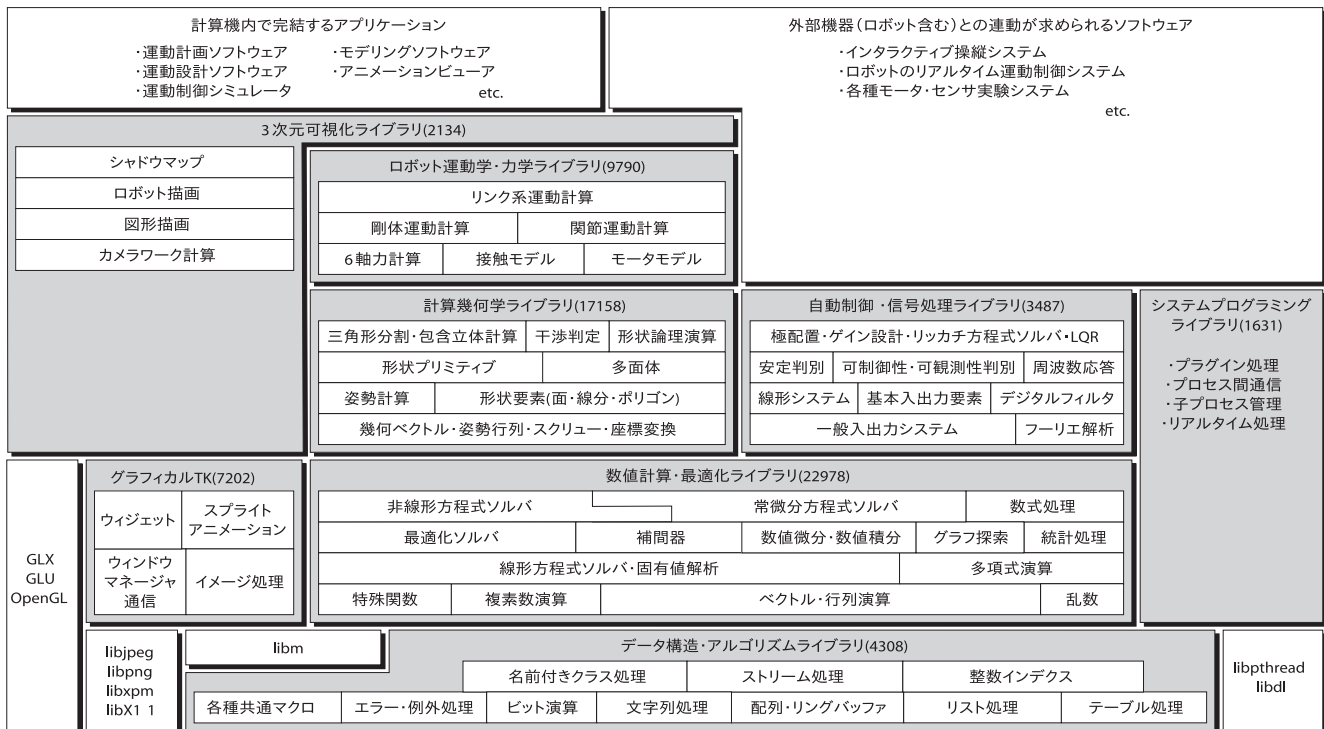


図1 ライブラリ構成とアプリケーションイメージ

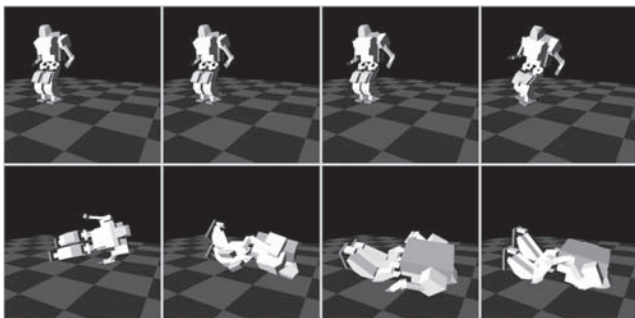


図2 ロボット順動力学シミュレータ

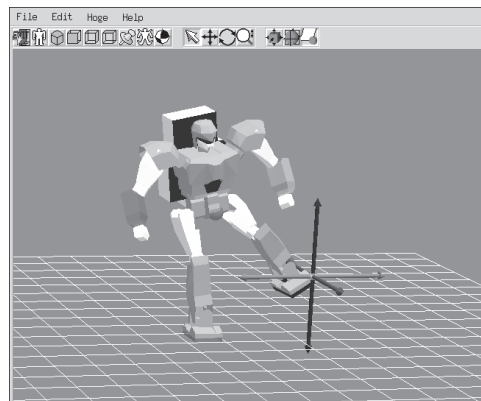


図3 ロボットのインタラクティブ動作設計ソフトウェア

タ [3] の実装例である。ライブラリは順動力学計算メソッドを直接提供しないが、逆動力学演算を用いれば、慣性行列や非線形力項の計算と常微分方程式の求解は 100 行あまりの C コードで行える。ロボットの順動力学は、衝突やアクチュエータダイナミクスとのカップリング計算等まで含めまだ技術的課題が残る難問であり、筆者の研究室では博士前期課程学生の脇坂尚樹君が取り組んでいる。

図 3 は、ピン・ドラッグインタフェース [31] を用いてインタラクティブに逆運動学 [2] を解くロボット動作設計ソフトウェア [15] の例である。プロトタイプは 1,000 行程度の C コードであり、3 日間で作成した。現在は幾何学的干渉や力学的不整合を自動解消するソフトウェアとして、博士前期課程学生の松本恭典君が開発を継続している。

図 4 は、一般的な家庭生活環境を想定した人型ロボットの

移動計画ソフトウェア [13] [14] であり、当時博士前期課程学生であった西俊哉君が Windows XP 上で開発してくれた。現在は博士前期課程学生の清水康志君が Linux (Ubuntu) 上に移植し、開発を受け継いでいる。

図 5 は、2 脚ロボットの操縦シミュレータ [17] で、計算機内の仮想ロボットをジョイスティックで手軽に操縦する計算の検証に用いている。なお、動力学計算は行っていない。博士後期課程の小林英仁君が短期間で開発してくれた。

図 6 は、2 脚ロボットの立位安定化、踏み出し、歩行等を統一的に行う制御理論 [6]~[8] を検証するために作成したソフトウェアである。画面上のツールバーを用いてインタラクティブにロボットの運動を変化させることができる。

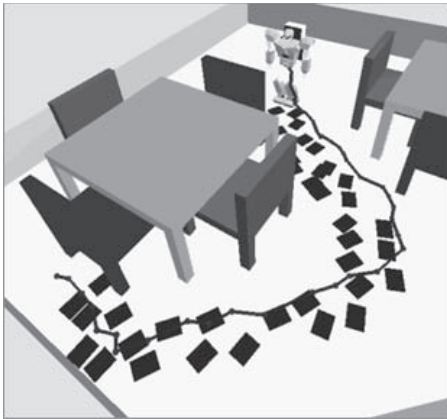


図 4 人型ロボット運動計画ソフトウェア

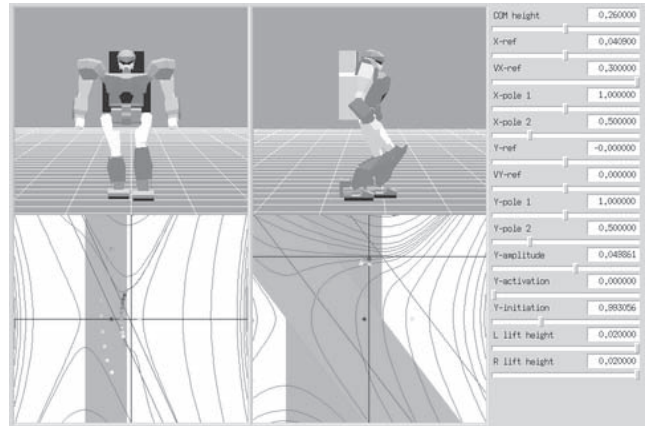


図 6 2脚ロボット制御シミュレータ

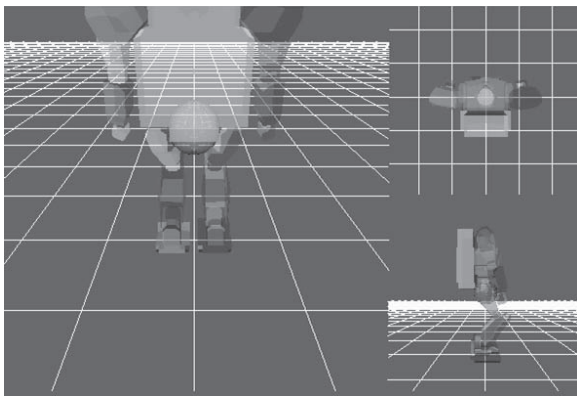


図 5 2脚ロボット操縦シミュレータ

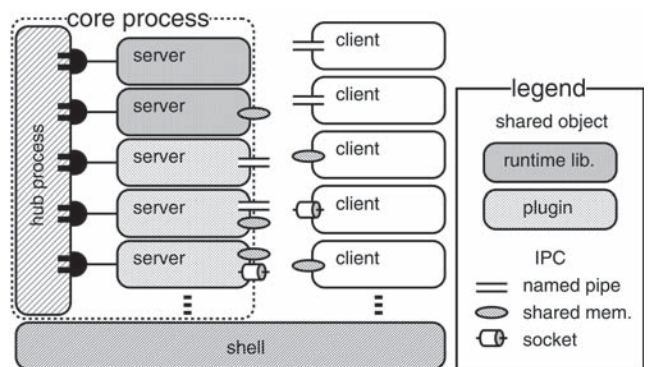


図 7 マイクロカーネルを模したロボット制御ソフトウェアフレームワーク

ただしロボットモデルは可視化と接触状態表現にのみ用いており、動力学計算は全質量が重心に集中した近似モデルに基づいている。プロトタイプは1,000行弱のCコードであり、1日間で作成した。その後、理論を拡張するたびにコードを追加しており、現在では1,500行弱である。

図 7 は、人型ロボットやレスキューロボットなど、運用場面に関する情報が開発初期に十分得られないロボットの制御システムを逐次的に拡張していくフレームワーク [21] の模式図である。ロボットモデル管理、センサ信号からの状態量推定、全身運動指令値計算、モータ制御等の機能を提供するサーバ群がプラグインアーキテクチャ上で連携することで、システム内部でのデータ可視性と保守性を両立する。テキストファイルで定義されたロボット情報の差し替えだけで様々なロボットに適用でき、実際に mighty や magnum [20] で用いられている。各サーバもクライアントも共通のライブラリ群の上に実装しているため、通信におけるオーバーヘッドは少ない。

6. おわりに

オフラインでの運動計算・解析からオンラインでのハードウェア制御まで扱えるソフトウェア資源の要件と、それ

をブラックボックス／ホワイトボックス的に構築することの長所と短所をまとめ、ホワイトボックス開発の事例として拙作ライブラリ群と幾つかのアプリケーションを紹介させていただいた。コスト対効果が上がるまでには忍耐が必要だが、それでも、いつでも「痒い処に手が届く」ソフトウェア開発のストレスの少なさは、大規模になるほど実感できる。

誤解を避けるために記すが、信頼して使える一貫した資源が揃っているという前提の下では金広らの主張は正しい。一研究室の開発にとどまらず、分野全体での開発効率を高め産業化を促進するために、ロボットのために信頼して使える標準的計算資源を、ロボット開発者自身が整備する必要がある。これはライブラリに限った話ではない。例えば、運動の可視化については優れたソフトウェアが多くあるが、それらにロボットの情報をどう渡すかが問題である。ゲームやアニメーション、バーチャルリアリティ等を目的としたフォーマットは、ロボット用途には過不足がある。ロボットに適したフォーマットの策定も、現在この分野が抱える課題の一つだろう。本稿を以て、そのような動きをもとに起こそうという呼びかけに替えさせていただきたい。

参 考 文 献

- [1] 杉原知道, 中村仁彦: “非駆動自由度の陰表現を含んだ重心ヤコビアンによる脚型ロボットの全身協調反力操作”, 日本ロボット学会誌, vol.24, no.2, pp.222–231, 2006.
- [2] T. Sugihara: “Solvability-Unconcerned Inverse Kinematics by the Levenberg-Marquardt Method,” IEEE Transaction on Robotics, vol.27, no.5, pp.984–991, 2011.
- [3] 杉原知道, 中村仁彦: “マイクロ・マクロ衝突モデルの融合によるリンク系順動力学接触力計算の数値的悪条件緩和”, 日本ロボット学会誌, vol.26, no.7, pp.767–777, 2008.
- [4] 杉原知道, 中村仁彦: “時間二重外乱吸収法に基づくヒューマノイドロボットの全身協調運動制御”, 日本ロボット学会誌, vol.24, no.1, pp.64–73, 2006.
- [5] 杉原知道, 中村仁彦: “ZMP-重心モデルと台車型倒立振りモデルのアナロジーによるヒューマノイドロボットの高機動化制御”, 日本ロボット学会誌, vol.24, no.1, pp.74–83, 2006.
- [6] T. Sugihara: “Standing Stabilizability and Stepping Maneuver in Planar Bipedalism based on the Best COM-ZMP Regulator,” Proceedings of the 2009 IEEE International Conference on Robotics & Automation, pp.1966–1971, 2009.
- [7] T. Sugihara: “Dynamics Morphing from Regulator to Oscillator on Bipedal Control,” Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2940–2945, 2009.
- [8] T. Sugihara: “Consistent Biped Step Control with COM-ZMP Oscillation Based on Successive Phase Estimation in Dynamics Morphing,” Proceedings of the 2010 IEEE International Conference on Robotics & Automation, pp.4224–4229, 2010.
- [9] F. Seto and T. Sugihara: “Motion Control with Slow and Rapid Adaptation for Smooth Reaching Movement Under External Force Disturbance,” Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1650–1655, 2010.
- [10] K. Masuya, T. Sugihara and M. Yamamoto: “Design of Complementary Filter for High-fidelity Attitude Estimation based on Sensor Dynamics Compensation with Decoupled Properties,” Proceedings of the 2012 IEEE International Conference on Robotics and Automation, pp.606–611, 2012.
- [11] T. Sugihara and Y. Nakamura: “Boundary Condition Relaxation Method for Stepwise Pedipulation Planning of Biped Robots,” IEEE Transaction on Robotics, vol.25, no.3, pp.658–669, 2009.
- [12] T. Sugihara and Y. Nakamura: “Enhancement of Boundary Condition Relaxation Method for 3D Hopping Motion Planning of Biped Robots,” Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.444–449, 2007.
- [13] T. Nishi and T. Sugihara: “Thinning and Smoothing of Randomly-sampled Support Transitions Toward Practical Motion Planning for Humanoid Robots,” Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1702–1707, 2010.
- [14] 清水康志, 杉原知道: “環境の複雑さに応じた身体表現の自動適合による人型ロボットの効率的運動計画”, 日本機械学会ロボティクス・メカトロニクス講演会'12 予稿集, 2012.
- [15] 松本恭典, 杉原知道: “人型ロボットのインタラクティブ動作設計における運動拘束条件の自動整合”, 日本機械学会ロボティクス・メカトロニクス講演会'12 予稿集, 2012.
- [16] T. Sugihara and H. Kobayashi: “A Handy Humanoid Robot Navigation by Non-interruptive Switching of Guided Point and Synergetic Points,” Proceedings of the 2008 IEEE-RAS International Conference on Humanoid Robots, pp.640–645, 2008.
- [17] H. Kobayashi and T. Sugihara: “Self-consistent Automatic Navigation of COM and Feet for Realtime Humanoid Robot Steering,” Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3525–3530, 2009.
- [18] T. Sugihara, W. Takano, K. Yamamoto, K. Yamane and Y. Nakamura: “Online Dynamical Retouch of Motion Patterns Towards Animatronic Humanoid Robots,” Proceedings of the 2005 IEEE-RAS International Conference on Humanoid Robots, pp.117–122, 2005.
- [19] 兼田大史, 杉原知道: “全身運動計測に基づく人の運動制御の巨視的な構造可視化”, 日本機械学会ロボティクス・メカトロニクス講演会'12 予稿集, 2012.
- [20] T. Sugihara, K. Yamamoto and Y. Nakamura: “Hardware Design of High Performance Miniature Anthropomorphic Robots,” Robotics and Autonomous System, vol.56, pp.82–94, 2007.
- [21] 杉原知道: “擬似マイクロカーネルに基づく成長型ロボットソフトウェアアーキテクチャ”, 日本機械学会ロボティクス・メカトロニクス'08 講演会予稿集, 2008.
- [22] 金広文男, 稲葉雅幸, 井上博允: “人間型ロボットの発展的実現のためのソフトウェアシステム構成法”, コンピュータソフトウェア, vol.17, no.6, pp.52–56, 2000.
- [23] T. Matsui and M. Inaba: “EusLisp: An Object-Based Implementation of Lisp,” Journal of Information Processing, vol.13, no.3, pp.327–338, 1990.
- [24] 高橋孝, 松永三郎, 狼嘉彰: “形態変化ロボットシステムのための汎用動力学プログラム”, ロボティクス・メカトロニクス講演会'98 講演論文集, pp.2BI1–6, 1998.
- [25] <http://www.astro.mech.tohoku.ac.jp/spacedyn/index-j.html>.
- [26] T. Hirano, T. Sueyoshi and A. Kawamura: “Development of ROCOS (Robot Control Simulator)—Jump of Human-type Biped Robot by the Adaptive Impedance Control,” Proceeding of 6th International Workshop on Advanced Motion Control, 2000.
- [27] I.A.D. Nesnas, R. Volpe, T. Estlin, H. Das, R. Petras and D. Mutz: “Toward Developing Reusable Software Components for Robotic Applications,” Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2375–2383, 2001.
- [28] A. Mallet, S. Fleury and H. Bruyninckx: “A Specification of Generic Robotics Software Components: Future Evolutions of GenoM in the Orocos Context,” Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2292–2297, 2002.
- [29] M. Barabanov: A Linux-based Real-Time Operating System, Master's thesis, New Mexico Institute of Mining and Technology, 1997.
- [30] M.W. Walker and D.E. Orin: “Efficient Dynamic Computer Simulation of Robotic Mechanisms,” Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, vol.104, pp.205–211, 1982.
- [31] K. Yamane and Y. Nakamura: “Natural Motion Animation through Constraining and Deconstraining at will,” IEEE Transactions on Visualization and Computer Graphics, vol.9, no.3, pp.352–360, 2003.



杉原知道 (Tomomichi Sugihara)

1999年東京大学工学部機械情報工学科卒業, 2004年同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修了. 同年同専攻学術研究支援員. 2005年より2007年まで同専攻助手. 九州大学高等研究院特別准教授を経て, 2010年より大阪大学工学研究科知能・機械創成工学専攻准教授. 博士(情報理工学). 人型ロボットの運動学, 力学, 運動計画と制御, 操縦システム開発, ハードウェア設計, ソフトウェアシステム開発, 人の運動計測と解析等に関する研究に従事. IEEEの会員. (日本ロボット学会正会員)