

Controlling an inverted pendulum using Linux-IOC

Atsuyoshi Akiyama, Jiang Geyang, Jun-Ichi Odagiri, Noboru Yamamoto

KEK, Accelerator Lab., 1-1 Oho, Tsukuba

Ibaraki, Japan, 305-0801

Abstract

New version of EPICS toolkit R3.14 supports Linux as an IOC(Input Output Controller). We have build a system to control an inverted pendulum using Linux-IOC. Device support program for PCI DA/AD boards were developed. We discuss the possibly to apply EPICS-IOC for real-time control system and its limitation.

1 始めに

KEKB や PF-AR などの加速器制御システムで採用されている、加速器制御ソフトウェア・ツールキット EPICS の最新版 R3.14 では Linux を機器装置側の制御計算機 (IOC, Input Output Controller) として利用することが可能になった。本研究では、この Linux-IOC を用いて倒立振り子装置を制御[1]するためのソフトウェアを開発し、Linux-IOC の制御システムへの適用上の問題点や利点などを検討した。

Linux-IOC を使うことにより、PCI 入出力ボードを EPICS ソフトウェアから制御するためのデバイスサポートソフトウェアの開発が必要となるが、Linux 対応の PCI 入出力ボードドライバソフトウェアがボードメーカーなどから供給されている場合には、比較的短時間で EPICS からの利用が可能となることが示された。また、Linux 上で利用可能な汎用ソフトウェア開発支援のプログラムを利用することで、プログラム開発が効率よく行えた。

通常の Linux は実時間制御を考慮した OS ではないため、そのままでは倒立振り子のような実時間性が要求される制御対象を制御することは困難であった。Linux の kernel パラメータの調整と EPICS ioc-sh プロセスのスケジューリングポリシーの変更を併用することで、倒立振り子を制御し倒立状態を維持できるようになった。

以下の第二章では本研究に利用した倒立振り子装置のハードウェアについて、第三章ではソフトウェアについて説明する。第四章ではそれらの開発研究によって明らかになった知見をまとめる。

2 倒立振り子装置

本研究で使用した倒立振り子装置は、サーボテクノ株式会社(URL <http://www.servotechno.co.jp/>)製の倒立振り子装置である。(図-1)装置は倒立振り子、台車駆動用 DC サーボモータ、角度、位置検出用ポテンシオメータ、ドライバボックスなどから構成される。二つのポテンシオメータの出力は、ドライバボックスの端子から+10Vのアナログ出力として取り出される。ドライバボックスには台車位置制御のための制御電圧入力端子も備えられており、これも+10Vの速度制御信号を入力する。

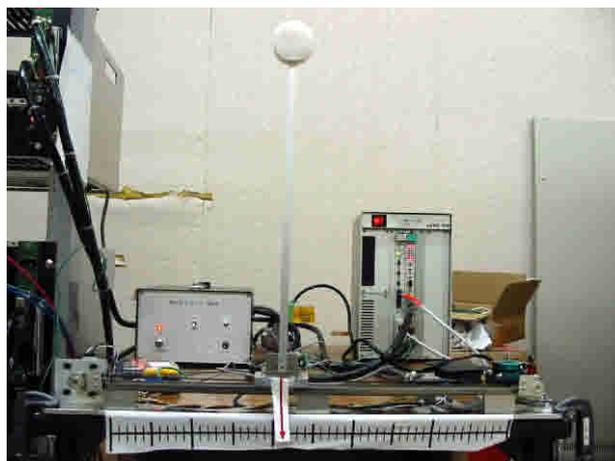


図 1 : 倒立振り子装置の外観

本研究で使用した機器にはさらにリミットスイッチを設置し、ドライバボックス内の DC モータ制御基盤に接続している。

ドライバボックスの二つの出力信号は Contec 社製 AD 変換 PCI ボード(Contec AD12-16)を経由して Linux PC に取り込まれる。また制御信号はこれも Contec 社製の DA 変換 PCI ボード(Contec DA12-4)を通じてドライバボックスに送られている。

制御計算機は、日本コンピューティングシステム(JCS)社製の PC(CPU: Intel® Pentium® 4, 1.8Ghz, Memory:512MB, HD:80GB)を使用し、EPICS R3.14.2 の制御プログラムは Red Hat Linux 9 の上で稼働している。

倒立振り子の制御アルゴリズムには状態推定器付き状態フィードバック法を用いている[“EPICS による倒立振り子の制御” <http://www-acc.kek.jp/WWW-ACC-exp/KEKB/control/member/N.Yamamoto/InvPendControl.pdf>]. 制御係数は Octave-2.1.50 を用いて計算された。

3 EPICS デバイス・サポートソフトウェア

倒立振り子の制御プログラムはこれまで VME 計算機上で利用してきた EPICS のソフトウェアを利用することができた。これまでのシステムでは CAMAC の DA/AD モジュールを使っていたが本システムは Linux/PCI の DA/AD 変換ボードを利用するため、これらのボード用の EPICS デバイスサポートソフトウェアを開発する必要があった。このシステムで使用している AD/DA 変換 PCI ボードには

Contec 社から Contec 社製ボードの汎用 Linux ドライバースフトウェアが提供されている。この汎用ドライバーを EPICS デバイスサポートから呼び出すことで、EPICS からの入出力を実現することとした。

今回のデバイスサポート作成では必要最小限の機能として、ボードのアナログ・チャンネル毎の入出力機能のみをサポートすることとし、ai および ao レコードのデバイスサポートを作成した。

汎用ドライバーのプログラム開発を支援するために、この汎用ドライバーをスクリプト言語 Python から呼び出すためのラッパーも作成した。ラッパーは SWIG(Simple Wrapper Interface Generator)を用いて開発した。これを Python と合わせて用いることで、ボード動作の確認が容易に行える。

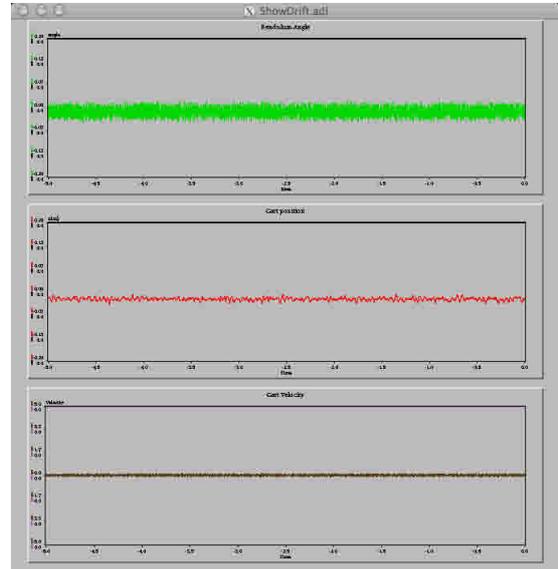
Linux-IOC の基本プロセスである ioc-sh は定期的にフィードバック制御のためのレコードの処理(process)を開始する。このレコード処理はレコード間のリンク機能を通じて、PCI カードのデバイスサポートソフトウェアを呼び出して実際の入出力を行う。離散化された状態推定器付き状態フィードバックのアルゴリズムでは、このフィードバックのステップが等間隔で行われると仮定しており、この処理間隔の揺れはフィードバックのエラーの原因として、系の安定性を損なうこととなる。

PCI カードの汎用ドライバーを用いることで、EPICS デバイスドライバーの開発は大きく簡略化される。しかしながら、この汎用ドライバー(iolib)は Unix のデバイスドライバーの作法に従い、デバイスファイルへの入出力の形で実現されている。汎用ドライバーであるがために、一つのアナログ値の入出力に対しても複数のボード上のレジスタへのアクセス、したがって複数のデバイスファイルへの入出力が発生してしまう。Linux の標準のスケジューリングアルゴリズムでは、このデバイスファイルへの入出力が生じる毎に、EPICS IOC-sh プロセスは待ち行列に入ってしまうため、EPICS レコードの処理もここで待ち状態に入ってしまう、フィードバックプロセスの処理間隔が不定となってしまう。また、この待ち行列の処理は Tick と呼ばれる Linux の kernel パラメータで決められる時間間隔で行われている。通常この Tick は 10msec と設定されている。フィードバックの一連の処理ではデバイスファイルの操作毎にこの待ち行列処理が入ってしまうため、System I/O 呼び出しの回数 \times 10msec が処理時間としてかかってしまう。このためフィードバックの繰り返しを十分あげることができない。

このスケジューリング上の問題に対処するため二つの方策をとった。

1. Linux のシステムコール sched_setscheduler()を呼び出すことで、ioc-sh プロセスの静的プライオリティをあげ、かつスケジューリングポリシーとして FIFO を割り当てる。
2. Kernel パラメータを変更し再構築することで Tick を 1msec (あるいはそれ以下)にする。

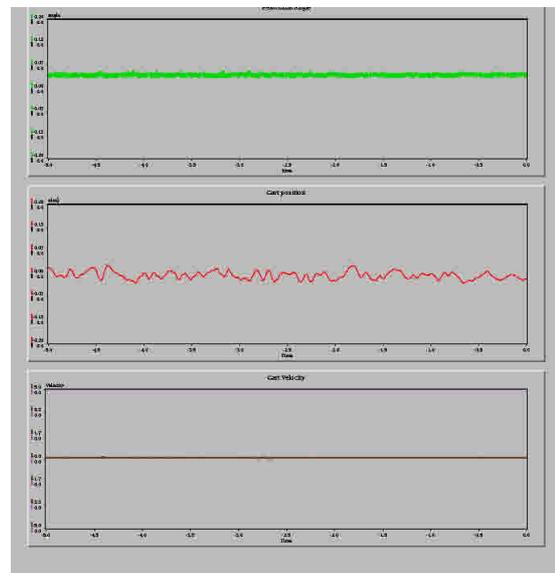
図 2 : Linux-IOC(SCAN rate=100Hz)制御時の倒立振り子の状態の時間変化、グラフは上からそれぞれ、倒立振り子の傾き、台車位置、台車速度の指令値を表す。



始めの対策により、システムコールの呼び出しによりいったん ioc-sh プロセスが待ち状態に入っても、プロセスの待ち行列の先頭に居残ることが可能になる。これにより、システムコール終了後ただちに ioc-sh の処理に制御が戻ってくる。

また後者の対処により、たとえ別プロセスにより実行権を失っても、数 Ticks(数ミリ秒)後には制御は ioc-sh に戻ってくる。

図 3 : VxWorks IOC(SCAN rate = 60Hz)制御時の倒立振り子の状態の時間変化。台車位置の遅い変化は環境の空気の流れなどによると考えられる。振り子の傾きは図 2 に比べ安定していることがわかる。



これらの対処により、ioc-sh 上でのフィードバックプロセスをほぼ 100Hz で実行できるようになった。

図-2 にこのフィードバックによる制御中の倒立振り子の傾きおよび位置の時間変化を示す。

4 結論

EPICS R3.14 ソフトウェアと汎用 PC カードドライバーを組み合わせることで倒立振り子制御システムを構築した。

汎用ドライバーを使うことで、容易に EPICS による制御システムを構築することができた。また、汎用ドライバーをテストするためにスクリプト言語とのインタフェースを開発した。これによってボードの動作確認が容易になり、開発時間の短縮に役立った。

EPICS を 通常配付の状態の Linux で実行するとプロセススケジューリングの問題で、実時間制御を行うことはできなかった。EPICS プロセスのスケジューリングポリシーを変更する必要がある。また kernel パラメータの変更によりプロセススケジューリングの時間粒度を細かくすることが必要であった。これらの変更によりはじめて、倒立振り子の倒立状態を維持するのに十分なフィードバック時間の精度が確保される。

Linux による倒立振り子の制御結果(図-2)とこれまでの実時間 OS(VxWorks)の制御結果(図-3)を比べてみると、Linux-IOC による制御ではフィードバック周期が短時間 (10m sec=100Hz) であるにもかかわらず VxWorks (1/60sec) のそれに比べ、残留誤差が大きい。フィードバックのその他のパラメータは同じであるので、これは Linux では離散的なフィードバックループが実行される間隔にばらつきがあるために生じていると考えられる。

Linux においても Real Time 性を導入する様々な提案されている。これらの Real Time 性を持った Linux の上で EPICS ソフトウェアを使うことで、より厳しい実時間性を要求する機器の制御が可能になると考えられる。

5 謝辞

参考文献

- [1] 山本 昇、“EPICS による倒立振り子の制御”、<http://www-acc.kek.jp/WWW-ACC-exp/KEKB/control/member/N.Yamamoto/InvPendControl.pdf> and references therein.