

分割主鏡制御系構成案 II — 制御ソフトウェア ver.1

岡山新技術望遠鏡グループ / Atsushi Shimono

平成 22 年 5 月 18 日

目次

1	概要	1
2	分割主鏡制御の概要	2
2.1	必要な対外インターフェース	2
2.2	必要な内部制御系	2
3	ソフトウェア構成案	2
3.1	基本プラン I — 下層モジュール構造	2
3.1.1	汎用制御インターフェース変換モジュール	3
3.1.2	ハードウェア仮想化モジュール	3
3.2	基本プラン II — 制御モジュール	3
3.2.1	位置・ギャップセンサーの HVL	3
3.2.2	CTRL1 と HVL との関係	3
3.2.3	CTRL1 と CTRL2 との関係	4
3.2.4	CTRL2 と CTRL3 との関係	4
3.2.5	CTRL2 とギャップセンサー (HVL) との関係	4
3.2.6	全体構造図	4
3.3	内部で共有すべきステータス	4

1 概要

分割主鏡の制御系ソフトウェアは、直接制御するハードウェアの数が非常に多くなっており、かつ制御やセンサーのステータス送受信などにより主鏡と関係する望遠鏡コンポーネントも非常に多岐にわたる。このため、制御系ソフトウェアでは耐障害性やメンテナンスの容易さを重視し、可能な限り独立した複数のモジュールによって構成されることが望ましいといえる。その上で接続・制御する必要のあるハードウェアへのインターフェースや、ソフトウェアの負荷が過大になったときにスケールアウトが可能になるように設計されていることが望ましい。

これらを実現するためには、各モジュールで提供されるステータス配信もしくは制御のインターフェースを IDL¹ などの言語で定義することや、ステータス供給・配信のモジュールはネットワーク透過に設計しておくことが望ましいといえる。また、分割主鏡シミュレータと直接接続する形での制御系構築・パラメータ試験を行うためには、ハードウェアや外部と関連する部分のモジュールをダミーモジュールと置換可能な構成にしておく必要がある。

本レポートでは上述のような基本的な要求を考慮しながら、必要とされる制御を行うためのソフトウェア全体についての構成案を提案する。

¹IDL; Interface Definition Language

2 分割主鏡制御の概要

2.1 必要な対外インターフェース

対外インターフェースは大きくステータス参照と制御 (アクチュエーター) の二つに分けられる。

制御の方はアクチュエーターのみであり、デバイスドライバへ制御パラメータを渡すというもの。また、逆にアクチュエーターの動作位置が正常化を検出する必要がある。

ステータス参照の方は、この制御系が直接読み出すステータスと外部から供給されるステータスがある。制御系が直接読み出すステータスには、位置センサー、温度センサー、湿度センサーの各センサーがあり、これらに直接アクセスしてステータスを参照する。外部から供給されるステータスは環境パラメータとしての気圧、望遠鏡の向きなどの望遠鏡制御パラメータがある。外部から供給されるステータスについては、ネットワーク経由で望遠鏡全体として定義される何らかのプロトコルに従って取得する。

2.2 必要な内部制御系

全体の制御は、制御対象となるアクチュエーターに近い部分からいくつかのレイヤーに分けることができる。線図にすると図 1 のようになる。

単体アクチュエーター (CTRL 1) 無間接でこの機構を通した後のヘッド位置を制御する、単体アクチュエーターのフィードバックループ。制御系への入力はヘッド位置である。

セグメント間ギャップ制御 (CTRL 2) セグメント間のギャップセンサーで測定されたギャップをなくすように制御する、アクチュエーター全体に対するフィードバックループ。制御系への入力はセンサーの読みの値である。

PC による位相ずれリファレンス (CTRL 3) セグメント間の位相ずれを光学的に測定し、セグメント間ギャップセンサーの絶対位置リファレンスとする制御機構。入力は SH/PCS からのずれ量。

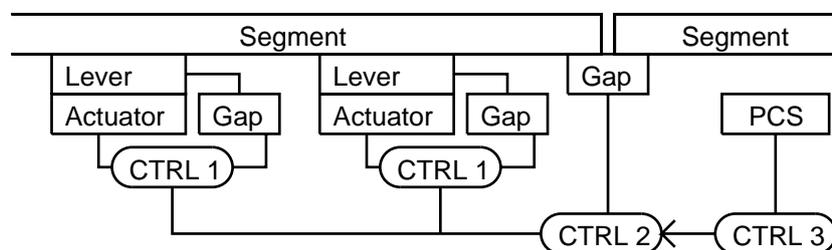


図 1: 内部制御系線図

3 ソフトウェア構成案

3.1 基本プラン I — 下層モジュール構造

ソフトウェアの内部モジュール構造・分割方法をまず考える。

ハードウェアと制御系間のインターフェースは 2 段階のモジュール化を採用する。ハードウェアに最も近い部分には、ハードウェア固有のドライバをラッピングし汎用制御インターフェースに変換するようなモジュールを作成する。その上で、それらの個々のハードウェアに対応する汎用インターフェースをまとめてひとつの制御系 (アクチュエーター系やギャップセンサー系など) ごとに仮想化するモジュールを作成する。

各制御モジュールは、これらのハードウェアを仮想化するモジュールの上のレイヤーとして実装し、制御モジュール内ではハードウェアの構成から独立した形で制御を行うようにする。

3.1.1 汎用制御インターフェース変換モジュール

デバイスドライバの固有 API を汎用制御インターフェースに変換するモジュールは、単純にデバイスの死活制御を行い、外部インターフェースとしてある程度汎用に利用可能なデバイスへのアクセス用 API を提供する。内部処理としては、デバイスへのアクセスステータスを保持し、その上で単純に API 同士の変換を行うだけのモジュールとし、特殊な制御などは行わない。

このモジュールを何らかのダミーモジュールに入れ替えることでより上のレイヤーのモジュールが正常に稼働しているかの試験や、シミュレータとしても利用することが可能になる。

3.1.2 ハードウェア仮想化モジュール

ハードウェア仮想化モジュールでは、デバイス番号という意味を含む実デバイスとシステム全体での制御で利用するデバイスの対応関係をとることを主目的とする。また、生データの平滑化処理や、IPC などを経由してのハードウェアステータス情報の他プログラムとの共有などもこのモジュール内部で行う。また、このレイヤーには、複数の CPU (PC) を利用してハードウェア全体を制御する場合や、外部ステータスを参照する場合に必要な、ネットワークアクセスの機能も含む。つまり、ハードウェアがどのような構成・配置をとっていたとしても、このモジュールが上位レイヤーに提供する API・ステータスやアクセス手段は、システム全体における ID のみを利用するものとなる。

3.2 基本プラン II — 制御モジュール

3.2.1 位置・ギャップセンサーの HVL

これらの HVL は内部で温度・湿度・気圧のステータスを参照してセンサー出力値のキャリブレーションを行う必要がある。

しかしながら、これらのキャリブレーションは静的なステータスを参照して行うだけでよく、かつ温度などのパラメータはセンサー出力値の読み出し頻度よりかなり遅い頻度で更新されるため、多少のタイミングずれは許容できる。

つまり、HVL 内部で持つステータス共有機構に依存するのだから大丈夫。

また、これらの HVL については、センサーの読み出し速度がセンサードライバ側の制限で決まっており、ステータスを一定頻度で読み出し続けて IPC/SHM を更新し続けるデーモンと、そこにアクセスするためのライブラリで構成されていても、運用上は変わらないともいえる。このため、ここの部分の具体的な実装に関しては、ソフトウェア安定性要求などの仕様面から検討する必要もある。

3.2.2 CTRL1 と HVL との関係

CTRL1 — アクチュエーターのクローズドループについては、CTRL1 が持つハードウェアを除く外部とのインターフェースは外部からアクチュエーターの移動量の入力のみである。また、制御中にアクチュエーター側もほかの制御入力を受け付けることはない。逆に、アクチュエーター側の駆動ステータスを参照しながら制御する必要があるため、これらのモジュールは密接に接続しておく方が望ましい。つまり、IPC のような外部を経由する形態をとらず同じ制御ソフト内で直接 API に接続する形にすべきである。

CTRL1 — 位置センサー側との間については、アクチュエーターと位置センサーは 1 対 1 対応であり、ほかからの制御入力を受け付けることはない。ステータスについてはこちら側はセンサーであるため、HVL 内のステータス共有機構でかまわない。(逆に、HVL 側が IPC/SHM 上に定期的に情報を更新する構造である場合、そのステータスを参照するライブラリにアクセスするのでもかまわない、はず²。)

² タイミングと頻度の関係の問題を検討する必要がある。

3.2.3 CTRL1 と CTRL2 との関係

CTRL2 は非常に巨大な演算を行う必要があるモジュールであり、いくつかの方式の試験が必要になる可能性がある。このため、CTRL1 と CTRL2 の間は祖結合にしておく必要がある。

今回の場合、この二つのモジュール間のやり取りは、基本的には CTRL2 から CTRL1 へのアクチュエーター駆動指示であり、メッセージキューのような構造での接続を行うことを検討する。

3.2.4 CTRL2 と CTRL3 との関係

CTRL3 はかなり長い間隔でユーザのトリガーにより駆動されるシステムであり、CTRL2 の駆動に影響する原点の情報をアップデートすることになる。

このため、CTRL3 と CTRL2 は直結する必要がなく、CTRL2 の原点情報の領域を CTRL3 が更新するような形でよい。

3.2.5 CTRL2 とギャップセンサー (HVL) との関係

CTRL2 とギャップセンサー HVL との関係は、CTRL1 とギャップセンサー HVL との関係と同じく、IPC のような外部を経由する形態をとらず、同じ制御ソフト内で直接 API に接続する形で問題ない。

3.2.6 全体構造図

これらをまとめた全体図は図 2 のようになる。

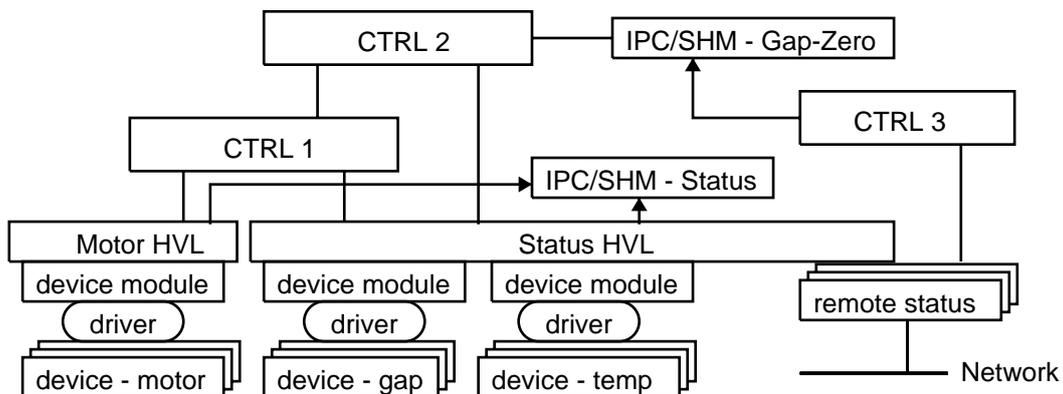


図 2: 制御系モジュール関連図

3.3 内部で共有すべきステータス

上記の内部制御系のモジュール構成において、デバッグの意味を含めた状態把握のために共有することが必要なステータスには以下のものがあげられる。

- 各アクチュエーターの制御位置と位置センサー読み
- 温度センサーの生読み出し値
- 位置センサー・ギャップセンサーの生読み出し値と補正值
- 望遠鏡環境データ (外部からネットワーク経由)

- 望遠鏡姿勢制御データ (外部からネットワーク経由)
- CTRL 1-3 の内部制御値
- ギャップセンサーの原点キャリブレーション情報