Robotising Existing Astronomical Observatories

G. Tuparev * [1], H. R. Andreasyan, † [1,2], and Thomas M. Jackson[1]

[1]Tuparev AstroTech, Sofijski geroj 3, Sofia 1612, Bulgaria
[2]V. A. Ambartsumian Byurakan Astrophysical Observatory, National Academy of Sciences of Armenia

Abstract

Astronomical observatories can be operated either manually or remotely, but both of these options currently have many disadvantages. Manual observatories require real-time staffing by on-site astronomers and technicians, and observation time is often not used optimally. Remote observatories, on the other hand, often make it cumbersome or near-impossible to modify any observing schedules already fed into the system, making reacting to important events like gamma-ray bursts unrealistic. In addition, it is generally expected that the cheapest way to upgrade an ageing observatory is to simply build a new one, but this is not always the case. For many relatively modern observatories, it is possible to convert to a fully robotic mode of operation. In these proceedings, we describe a few straightforward ways to robotise an existing observatory and how to connect it to a network of other robotic observatories. We also discuss the use of inexpensive device controllers, the importance of emergency shut-down procedures, the introduction of local and network schedulers, and the implementation of fully automated observation planning. We finally also describe hardware and software solutions, including an example of how this is currently being applied.

Keywords: *observatories, robotics, front-end and back-end software systems*

1. Introduction

With the modern age of astronomy well underway, larger and more complex projects with significantly more extensive data sets are now commonplace. One area of astronomy which has profited from this expansion, driven by advancing electronics, engineering, and software, is the area of robotic astronomy. A robotic observatory is an observatory that can take observations by itself when given instructions or a schedule, and it can react by itself to changing observational, weather, or mechanical/electronic conditions without human intervention. This differs from manual or remote observatories which require observers to manually control or input the instructions or commands by hand, regardless of where they are located with respect to the observatory.

There are now multiple examples of robotic observatories — the MONET project (Bischoff et al., 2006, 2008, Hessman, 2001), the Zwicky Transient Facility (Bellm et al., 2019, Dekany et al., 2020), and the Las Cumbres Observatory project (Brown et al., 2013), just to name a few. Indeed, robotic observatories provide many advantages over remotely or manually operated observatories, such as lower costs for maintenance of the telescope, little or no need for an operator, and more efficient observations. They also reduce the travel needed to and from the observatory itself, have the ability to operate in conditions which would be impossible for a manual operator such as deep snow or extreme heat, and are accessible world-wide, making them very suitable for education or outreach projects.

Despite these advantages, there remains a relative lack of robotic observatories to date, driven by the common perception that it is cheaper and easier to build a new observatory when it could be more efficient and cost effective to renovate and robotise certain observatories. While some observatories may be too old to robotise (such as the Royal Observatory in Greenwich), mainly due to parts no longer being available, and some of the biggest systems may be too complex to robotise (such as the Very Large Telescope at Cerro Paranal), there are still many potential candidates currently out there. Here we show some of the steps needed and the considerations for robotising an observatory.

*gtupar@tuparev.com
†h.andreasyan@tuparev.com, Corresponding author

doi:https://doi.org/10.52526/25792776-22.69.2-302

2. Implementation

As a basis for robotising an observatory or telescope, two things are needed: A dedicated, multi-disciplinary team of astronomers, mechanical engineers and software developers, and knowledge of the observatory itself.

In order to gather this knowledge and experience, the team should extensively use the observatory in remote mode, gathering notes about the operating procedures and connecting these to possible automations. Examples of this procedure are noting and archiving weather conditions such as dew-point and rain conditions, the cabling of the de-rotator, the conditions under which auto-focusing is used, etc.

Once this experience has been developed, implementations can begin with the six steps outlined in this section and the left hand side of Figure 1, and with the three points in mind, which we dub as Tuparev's Three Laws of Robotelescopics, shown on the right hand side of Figure 1.

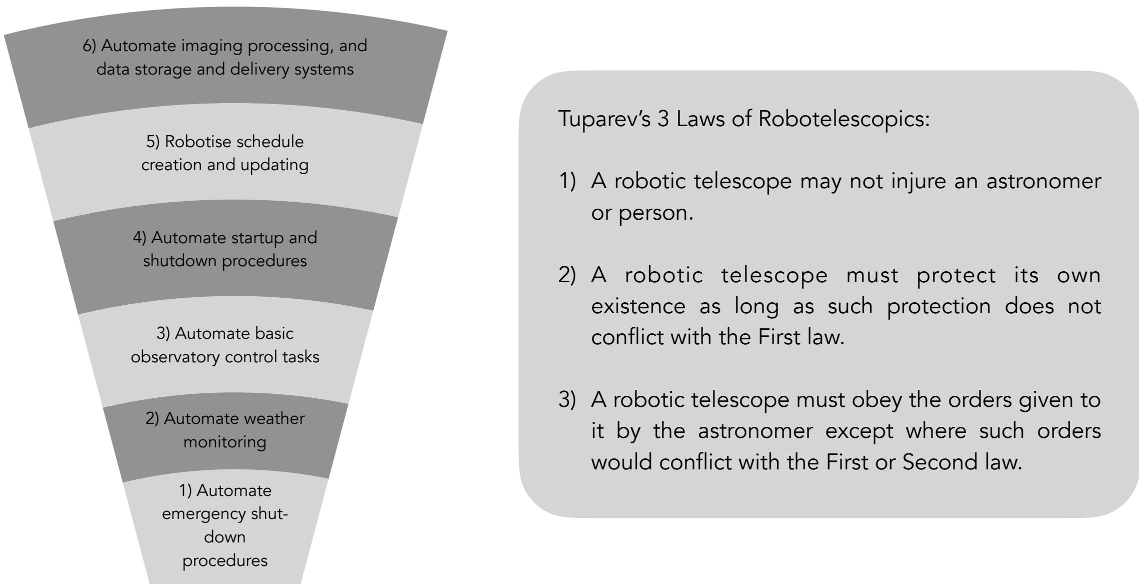

Figure 1. Left side: The 6 steps, presented here in a shell structure to reflect the software architecture, for robotising an observatory, which we outline in the following subsections. Right side: The 3 laws to keep in mind when designing the fundamentals for a robotic observatory.

2.1. Automation of the emergency shut-down procedures

The protection of the system is of relatively high priority according to the three points above, and is the first step we outline for robotising an observatory. If equipment is damaged, repair costs can be expensive or even render the observatory unsustainable. It follows that if the remote monitoring connection is lost in any way, which could potentially lead to a risk of damage to the equipment, the enclosure should close automatically. This includes loss of communication with vital systems such as weather stations or dome control.

In addition to automatic shutdown, to mitigate risks there are also other alternative fail-safes which can be implemented, such as making sure there is an emergency, uninterruptible power supply, that webcams are placed in all appropriate areas for monitoring, the automatic parking spot of the telescope is out of range of water damage if the dome does not close, or implementing passive or gravitational mechanisms for closing the dome.

doi:https://doi.org/10.52526/25792776-22.69.2-302

2.2. Automatic monitoring of the weather conditions

Second to the emergency shut-down procedures, the observatory's weather station, the monitoring it does and the communication it has with the control centre are all vital for a robotic observatory. The minimum weather parameters that should be monitored include rain, humidity, temperature, pressure (for dew-point calculations), dust, wind speed, and wind direction, as changes or the occurrence of any of these can at worst damage the telescope or at best severely reduce the observatory's performance.

These can be monitored via hardware such as cloud monitors on the dome with image analysis, online services, commercial weather sensors placed outside the dome, temperature sensors around the mirror etc. All of the information collected should be communicated to the central system, with limits set on operations dependent on the current weather conditions (e.g. a high risk of rain automatically shuts the dome, cloud to the eastern horizon means scheduled observations to the west are preferably weighted). It is also advisable to record these parameters for long-term records and observing logs.

2.3. Automating routine tasks

Following the placement of emergency procedures and weather monitoring, the automation of standard routines can be added. This focuses on the activation and control of most of the equipment via coded software routines. Basic functions include the steering of the mount, the activation of the camera, or the changing of filters in the filter wheel, which all need to be implemented for the observatory to function on the most basic level.

Once the basic activation and control functions are in place, more complex functions should be considered. Routines such as auto-focusing the camera or differentiating between taking calibration and science frames can be added to the software architecture, all of which are needed in some capacity for observations.

2.4. Startup and shutdown procedures

As a final step to automating the hardware, the startup and shutdown procedures of the observatory should be added.

Considerations for the startup mechanisms should include an initial check of weather conditions. As an example, there is no point opening the dome if it is raining or the sky is completely covered, and it should be checked again in a given time frame. This should be followed with a check of the operational status of all hardware, so the observatory is fully operational and risk-free before initiating standard startup procedures such as dome opening, cooling of the sensors and the taking of calibration images.

Frameworks should also be coded for shutdown procedures, in a somewhat reverse order, whereby final calibration images are taken, the telescope is parked and the enclosure is closed. Additional steps to consider for shutdown include getting a webcam confirmation that the shutdown mechanism has completed as an extra fail-safe before software-side tasks such as in Section 2.6 are initiated.

2.5. Robotic scheduling

Once the basic hardware procedures are automated, basic software functions can be automated too, including the scheduling which the robotic observatory will follow. Robotic schedules have a number of different parameters to account for and must contain two layers.

The initial layer of the schedule should account for a number of different target parameters, including the angular distance of the target to the zenith at any single time, the expected signal-to-noise of the target, the exposure time needed for the target, and the angular distance of each target from the previous and subsequent observations. In addition to the target constraints, there are also constraints such as sky conditions to account for, such as the moon phase and location in the sky at any one time. All of these factors should have weights and then constraint optimization algorithms applied to optimise the initial observing schedule.

Once the initial schedule is created, on-site constraints mean a secondary layer must be applied: Sky conditions such as cloud cover and distribution, or rain may hinder observations of certain targets or time slots, and rapid follow-up observations may have to be accounted for. This means constant updating of the schedule must be established to feed in unscheduled priority targets, or to re-optimise the initial schedule.

Observation scheduling must consider not just nightly observations but those over periods of weeks or months as well. To account for this, frameworks for daily feedback and logging to re-optimise schedules

doi:https://doi.org/10.52526/25792776-22.69.2-302

for subsequent nights, as well as networks of interconnected robotic telescopes are preferable for a fully optimised system.

2.6. The imaging pipeline

Once observations are taken from the optimised schedule and are readily available, automating the data analysis of the data is preferable to manual reduction for time and efficiency purposes. As robotic observatories work, on average, more efficiently than manual or remote observatories, more data is collected. This means efficient pipelines have to be created.

The basic steps should be preferably carried out on-site, as this reduces the amount of data that needs to be transferred over the network and makes archival side operations easier due to that lower volume. Such on-site processes may include deciding on the calibration images, running data reduction processes, calculating and integrating time and coordinates systems into the data, and compressing FITS files. Similarly, science analysis pipelines may also be included on-site, such as photometric measurements, line finding algorithms for spectroscopic data, or streak and transient detection.

Further steps to account for include making sure there is enough local storage space for a given amount of time (e.g. a week, month or year), so that the observatory can operate by itself for a given time frame without the need for human intervention or automatic transferring of the data to a hosting centre. These steps also need to be supported by a priority framework, whereby important event logs or important image or spectroscopic data has priority of being handled and distributed to central servers or alert services before lower priority data. Finally automatic removal of older data also needs to be implemented to avoid the discarding of new observations due to a lack of storage space.

3. Hardware requirements and configurations

We turn our attention now to the hardware needed for a robotic observatory. At it's core, a robotic observatory should contain the same equipment as a manual observatory; a dome, a mount, a telescope, an instrument such as a camera or spectrograph, a weather station, and a control computer. An external server, which can easily be located offsite, for implementing some of the automatic steps and storage is desirable, as well as a separate computing unit for analysis, especially if large amounts of data are expected to be produced so that the amount of data being transferred is reduced.

Every sensor or device should also be connected to the central control computer or server by micro-controller (a raspberry pi, for example) and be accessible on the LAN. This enables all stations to constantly send messages or "pings" confirming they are operational and, if a system malfunctions, allows for the automatic starting of emergency shut-down procedures and sends a warning message or notification to the maintenance staff astronomers. This combined system results in an ecosystem quite similar to the diagram presented in Figure 2.

4. Software architecture

In order to achieve a robust and maintainable software system, a multi-layered approach is strongly recommended. The lowest software level should connect all sensors (weather, temperature sensors associated with the enclosure and the telescope, etc.), peripheral devices (screen, webcams, etc.) with the dome, the UPS, and the internet router. Every component should periodically send an availability signal. In the case that such a signal is not received, the emergency shutdown procedure should be triggered.

Another very important task of the low-level software is logging. The observatory log should always be complete and easily searchable. The development of visual debugging tools to enable the monitoring of the messages sent between components is strongly recommended.

Higher level software should include real-time schedulers, long-term observation observatory schedulers, and optional modules that allow the automatic upload of processed images to a hosting site and communication with external network schedulers in order to negotiate the exchange of observations with other telescopes or to receive requests for high-priority observation requests (e.g. GRB observations). For more detailed information see Allan et al. (2006), Hessman et al. (2004), Tuparev et al. (2006).

The role of the real-time scheduler is to decide what observation should be scheduled next depending on the constantly changing environment and observation conditions, while the long-term scheduler should

doi:https://doi.org/10.52526/25792776-22.69.2-302

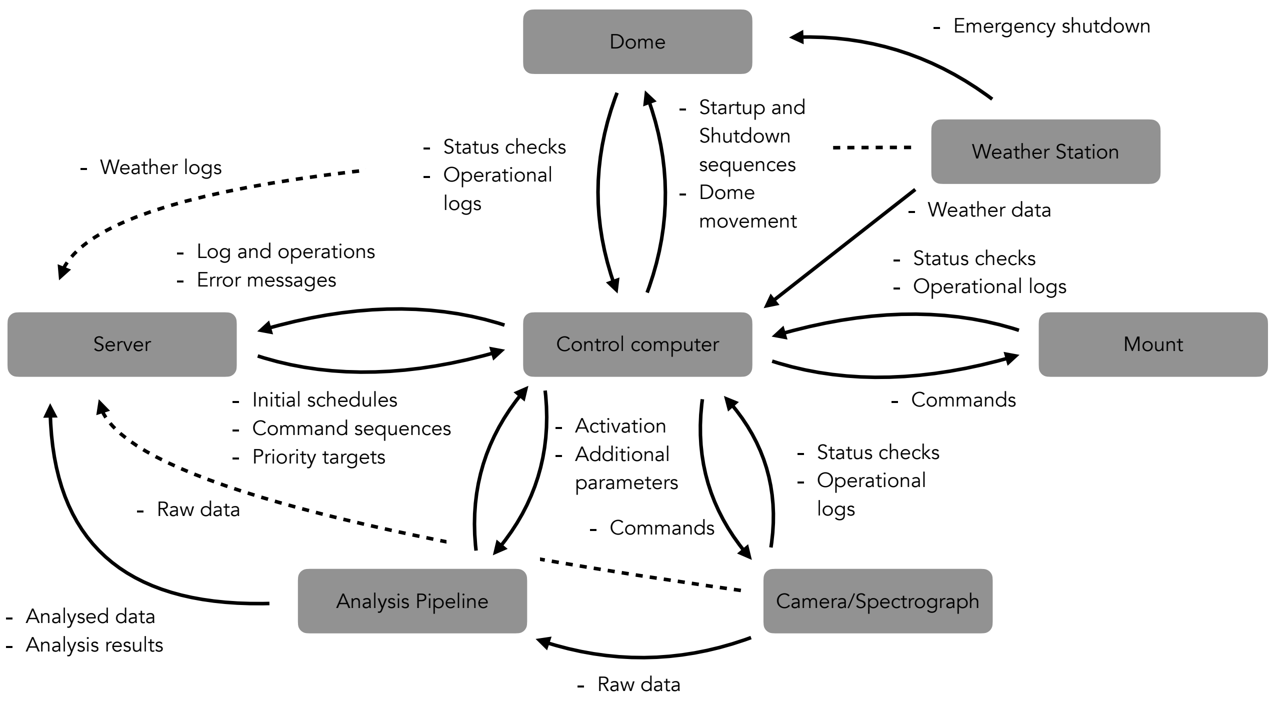

Figure 2. A diagram of the minimal hardware components and a simplified/minimal structure of the relations between each component and the operations that may be exchanged between each one.

create the observation planning for at least a few days in advance, taking into account weather condition forecasts as well as the nature of the scheduled observations and the corresponding science projects (e.g. spectroscopic observations could be avoided when the moon is visible, time-constrained observations, etc.).

5. Summary

In this work, we firstly outline the advantages of robotising existing observatories, namely that they are more efficient in observations, have significantly lower maintenance and running costs, and can generally operate in a wider range of conditions than manual or remote observatories.

We also outline the necessary steps needed to robotise an observatory given in the 6 steps outlined in Section 2, and an overview of what considerations are needed. We also present some of the relevant hardware components and considerations in Section 3 and software considerations in Section 4.

We add a final message that anyone considering robotising an observatory should a) consider the option of new equipment first, b) start slowly and take the process step-by-step, c) automate emergency procedures, d) carefully follow the six automation steps outlined here, and e) carefully architecture their software. We finally wish to state that we are happy to help and advise any organisations or individuals robotising their systems, or to provide entire robotising or archival systems services from our team.

Acknowledgements

The authors would like to thank Dr. Frederic Hessman, who discussed many of the ideas outlined here.

References

Allan A., et al., 2006, Astronomische Nachrichten, 327, 744

Bellm E. C., et al., 2019, Publ. Astron. Soc. Pac. , 131, 018002

Bischoff K., Tuparev G., Hessman F. V., Nikolova I., 2006, in Silva D. R., Doxsey R. E., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 6270, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. p. 62701Q, doi:10.1117/12.671433

Bischoff K., Hessman F. V., Tuparev G., Atanasova E., Pessev P., 2008, in Stepp L. M., Gilmozzi R., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 7012, Ground-based and Airborne Telescopes II. p. 70122X, doi:10.1117/12.788920

doi:https://doi.org/10.52526/25792776-22.69.2-302

Brown T. M., et al., 2013, Publ. Astron. Soc. Pac. , 125, 1031

Dekany R., et al., 2020, Publ. Astron. Soc. Pac. , 132, 038001

Hessman F. V., 2001, in Paczynski B., Chen W.-P., Lemme C., eds, Astronomical Society of the Pacific Conference Series Vol. 246, IAU Colloq. 183: Small Telescope Astronomy on Global Scales. p. 13

Hessman F. V., Pennypacker C. R., Romero-Colmenero E., Tuparev G., 2004, in Lewis H., Raffi G., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 5496, Advanced Software, Control, and Communication Systems for Astronomy. pp 290–301, doi:10.1117/12.547429

Tuparev G., Nicolova I., Zlatanov B., Mihova D., Popova I., Hessman F. V., 2006, Astronomische Nachrichten, 327, 771

doi:https://doi.org/10.52526/25792776-22.69.2-302