ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-202312.00140

Distributed Control Software for the Active Surface System of Tian-ma Radio Telescope Postprint

Authors: Dong Zhang, Rong-Bing Zhao, Zhen Yan, Wei-Hua Shang-Guan, Qian Ye, Zhi-Qiang Shen, Qing-Hui Liu, Jin-Qing Wang, Li Fu, Chu-Yuan Zhang et al.

Date: 2023-12-15T00:00:00+00:00

Abstract

The Tian-ma Radio Telescope (TMRT) employs an Active Surface System (ASFS) to correct for large-scale deformations of the primary reflector induced by gravitational and thermal effects. The centralized and automated management of the ASFS via software presents a significant technical challenge, for which we have developed the TMRT Active Surface System Control Software (TASCS). This paper delineates the design and implementation of TASCS, encompassing functionalities for device control, status monitoring, human-computer interaction, and data management. TASCS leverages the open-source Tango Controls framework and distributed middleware technology to achieve real-time automated adjustment of the primary reflector through remote centralized control of numerous actuators. Currently, TASCS has been successfully deployed on the TMRT and has contributed substantially to Event Horizon Telescope observations.

Full Text

Preamble

Research in Astronomy and Astrophysics, 23:115024 (7pp), 2023 November
© 2023. National Astronomical Observatories, CAS and IOP Publishing Ltd. Printed in China and the U.K.
https://doi.org/10.1088/1674-4527/acfa58

Distributed Control Software for the Active Surface System of Tian-ma Radio Telescope

Dong Zhang[12], Rong-Bing Zhao[1], Wei-Hua Shang-Guan[12], Qian Ye[1], Zhi-Qiang Shen[1], Qing-Hui Liu[1], Zhen Yan[1], Li Fu[1], Jin-Qing Wang[1], Dong-Fang Song[12], and Si-Yu Liu[13]

[1] Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, China; zhaorb@shao.ac.cn
[2] Chongqing Jiaotong University, Chongqing 400074, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China

Received 2023 June 1; revised 2023 August 9; accepted 2023 August 15; published 2023 October 17

Abstract

The Tian-ma Radio Telescope (TMRT) employs an Active Surface System (ASFS) to correct for large-scale deformations caused by gravitational and thermal effects on the primary reflector. Centralized and automated management of the ASFS through software has become a critical challenge, for which we have developed the TMRT Active Surface System Control Software (TASCS). This paper describes the design and implementation of TASCS, which provides functionalities for device control, status monitoring, human-computer interaction, and data management. TASCS adopts the open-source Tango Controls framework and distributed middleware technology to achieve real-time automated adjustment of the primary reflector through remote centralized control of numerous actuators. Currently, it has been successfully deployed on TMRT and has played an important role in Event Horizon Telescope observations.

Key words: Astronomical Instrumentation – Methods and Techniques – telescopes – instrumentation: high angular resolution – techniques: high angular resolution

1. Introduction

Due to their large size and weight, large-aperture antennas are susceptible to large-scale distortion caused by gravitational and thermal effects, which reduces antenna efficiency. Currently, major international radio telescopes such as the Green Bank Telescope (GBT), Sardinia Radio Telescope (SRT), and Tian-ma Radio Telescope (TMRT) have adopted Active Surface Systems (ASFS). ASFS can restore the ideal shape of the primary reflector by adjusting actuators, thereby improving antenna efficiency.

TMRT is currently the largest fully steerable Cassegrain-type radio telescope system in Asia. The ASFS of TMRT is equipped with 1104 actuators that provide real-time adjustments to the 1008 panels of the primary reflector. Figure 1 [Figure 1: see original paper] (Dong et al. 2018) demonstrates significant improvement in antenna efficiency when the ASFS operates in the Q-band (35–50 GHz).

The control software serves as the brain of the ASFS, responsible for coordinating all devices and monitoring their working status. Given the large number of devices and the complex structure of the ASFS, we adopted a distributed control framework for software design and development. The distributed control framework offers numerous advantages: by hiding details of underlying mechanisms (such as message queues, asynchronous communication, thread pools, and lifecycle management), it helps engineers improve development efficiency; and by standardizing device attributes and behaviors (such as data structures, interfaces, protocols, and state machines), it provides unified resource sharing and invocation for the system. Internationally, several large radio telescopes have adopted distributed control frameworks for their ASFS, such as Alma Common Software (ACS; Chiozzi et al. 2004 and Shen et al. 2011), Ygor (Ford & Clark 2002 and Ford et al. 2014), and Tango Controls (Baffa et al. 2019).

In this paper, we describe the design and implementation of the TMRT Active Surface System Control Software (TASCS) using Tango Controls. As an open-source, cross-platform, object-oriented distributed system framework, Tango Controls offers good scalability, rich application programming interfaces (APIs), and a series of practical toolkits (Verdier et al. 2011). It enables users to implement device control, data acquisition, and real-time transmission functions, making it an ideal choice for ASFS control software implementation.

The following sections first briefly introduce the ASFS hardware, then describe the software from three aspects: design, implementation, and testing.

2. Hardware

The ASFS of TMRT consists of actuators, control buses, subcabinets, and control software. The system is equipped with a total of 1104 actuators installed on the back structure of the antenna (Dong et al. 2016). The antenna panel position is adjusted by controlling four adaptive screws on each actuator. The ASFS utilizes an RS-485 bus to connect all actuators to Ethernet (see Figure 2 [Figure 2: see original paper]) and is divided into 24 fan sections. Each section is controlled by a subcabinet that includes three buses, resulting in 72 buses total. Each bus connects 15 or 16 actuators.

3. Software Design

3.1. Function Design

TASCS is designed to meet the diverse needs of its users, including astronomers and observation engineers. The system provides three core functional categories:

Monitoring and Control: The software continuously collects and analyzes device status information and raises alarms for system abnormalities. It reads the Antenna Deformation Model (ADM) from the database, calculates the optimal displacement for each actuator at the current elevation, and commands the actuators to move to their desired positions.

Human-Machine Interaction: The system provides remote access for multiple users through a graphical user interface (GUI) that displays real-time device status and offers a device debugging interface for fault analysis and troubleshooting.

Data Management: TASCS manages access to the ADM, stores device status data, and maintains log files for system monitoring and diagnostics.

3.2. Architecture Design

TASCS adopts the open-source Tango Controls framework, which utilizes CORBA (Common Object Request Broker Architecture) middleware technology (Götz et al. 2003). The core concept of Tango Controls is based on devices and device classes. A device, as the control object, represents either a physical hardware device or a logical device. It has unified features including attributes, properties, pipes, and commands, which are defined by the device class. Device objects are instantiated within a device server, which acts as a container process (Tan 2015).

As shown in Figure 3 [Figure 3: see original paper], the system architecture includes Actuator-Bus Device Servers, Coordinator Device Servers, and Subcabinet Device Servers on the server side, while the client side provides Device Control, Data Visualization, and Debugging functions. Communication between device servers and clients uses the Tango Middleware Protocol. The database layer consists of the Tango Database and Model Database.

3.3. Model Design

We use Unified Modeling Language (UML) to model the software structure, as shown in Figure 4 [Figure 4: see original paper]. The Actuator-Bus class inherits from the DeviceImpl base class and implements actuator initialization, state acquisition, and operation control through customized code. The Actuator-Bus Class defines unified interface characteristics including attributes, commands, and properties. The attributes include actuator position, speed, temperature, current, and alarm status; the properties include actuator address and serial number; and the commands include actuator startup, shutdown, stop operation, and abnormality recovery.

4. Implementation

TASCS comprises modules including Actuator-Bus Device Server, Coordinator Device Server, State Machine, GUI, Database, and Logging. The specific implementation of each module is described below.

4.1. Actuator-Bus Device Server

The Actuator-Bus Device Server is the core component of the entire software, using a bus as the control object. As shown in Figure 5 [Figure 5: see original paper], the Actuator-Bus Device Server first reads the ADM from the model database and calculates the optimal displacement for each actuator according to commands, then calls the device driver interface to control the actuators. Simultaneously, it collects actuator status data in real time, pushes it to clients, and records log files.

4.2. Coordinator Device Server

Due to the numerous devices in the radio telescope, we use the Coordinator Device Server (hereinafter referred to as coordinator) to synchronize various devices during the observation process (see Figure 6 [Figure 6: see original paper]). Generally, a complete observation process consists of three stages: preparation, observation, and termination. During the preparation stage, the antenna must move toward the target, the receiver must configure parameters, and the ASFS must adjust panel positions. To ensure all devices are ready before starting observation, the coordinator calculates a Synchronization Start Time (SST) based on the preparation time required by each device and distributes the SST to all device servers. Subsequently, each device server autonomously manages its corresponding device—for example, ensuring the antenna has pointed to the target, the ASFS has adjusted actuators to optimal positions, and the Data Acquisition System (DAS) is ready to record data. After reaching the SST, the antenna begins tracking, the ASFS starts real-time adjustment, and the DAS begins recording data. During the termination stage, all devices stop running and return to their initial state.

4.3. State Machine

To unify actuator behavior and ensure operation according to a strict timing sequence, we employ a state machine to describe the various working states of actuators and the transition methods between states. As shown in Figure 7 [Figure 7: see original paper], after power-on, actuators are initialized and configured with parameters to transition to the “Ready” state. When receiving commands from the coordinator, each actuator enters the “Activate” state, during which all actuators move toward their target positions. After reaching the synchronization start time, they enter the “Running” state and make real-time adjustments based on current elevation. During operation, if system abnormalities occur—such as device failure or transmission error—the actuators stop running and enter the “Standby” state.

4.4. Graphical User Interface (GUI)

The GUI is developed using Python, PyQt5, and the OpenGL library. The interface consists of three parts: Device Control, Data Visualization, and Debugging (see Figure 8 [Figure 8: see original paper]). Device Control provides users with device control interfaces. Data Visualization displays real-time device state information. Debugging provides users with device debugging interfaces for analyzing and troubleshooting device faults.

4.5. Database

The database consists of the Tango database and a model database. The Tango database provides device configuration information and runtime catalogs for both clients and device servers (Tan 2015). The ADM is stored in the model database, which records the displacement of each actuator at every elevation.

4.6. Logging

The logging module records the historical states of the ASFS, including the position and running mileage of all actuators and the state information of each device for system recovery and troubleshooting.

5. Measurement Results

Currently, TASCS has been successfully deployed on TMRT and plays an important role in Event Horizon Telescope (EHT) observations (Akiyama et al. 2019). It is primarily used in two cases during observations:

In the first case, the surface error of the primary reflector is measured using the Out-of-Focus holography technique (Nikolic et al. 2007 and Dong et al. 2014) prior to observation. TASCS converts the surface error into displacement values for the actuators and makes corresponding adjustments.

In the second case, the Antenna Control Software (ACS) guides the antenna to track the target during observation. It sends the current elevation to TASCS, which reads the ADM and adjusts the panel position in real time.

The control logic diagram for the ASFS is shown in Figure 9 [Figure 9: see original paper]. After enabling TASCS, we conducted surface measurements on the primary reflector and achieved a surface accuracy of 201 microns RMS (see Figure 10 [Figure 10: see original paper]), meeting our expectations.

6. Conclusion and Discussion

This paper describes the design and implementation of TASCS, which uses Tango Controls as the distributed control framework, follows a client/server design pattern, and employs C++ and Python languages. TASCS supports multi-user remote control and monitoring, enables easy upgrades, and allows flexible cross-platform deployment. During actual operation, TASCS automatically adjusts the primary reflector through actuators according to the ADM, thereby improving surface accuracy. Considering the advantages of Tango Controls in building distributed control systems, we will adopt it in more radio telescope projects in the future.

Acknowledgments

This work was supported by the National SKA Program of China, No. 2020SKA0120104 and the Natural Science Foundation of Shanghai, No. 20ZR1467600.

References

Akiyama, K., Alberdi, A., Alef, W., et al. 2019, ApJL, 875, L4

Baffa, C., Giani, E., & Vela Nunez, M. 2019, in SKA Monitor and Control: Harmonization Challenges Astronomical Data Analysis Software and Syst. XXVI ASP Conf. Ser., 521 (Trieste: ASP), 185

Chiozzi, G., Jeram, B., Sommer, H., et al. 2004, Proc. SPIE, 5496, 205

Dong, J., Jin, H., Ye, Q., et al. 2016, Proc. SPIE, 9913, 991306

Dong, J., Juan, L., Ya-jun, W., & Jin-qing, W. 2014, The Application Research of Out-of-focus Technique to Shanghai 65m Radio Telescope, in Annals of Shanghai Astronomical Observatory, CAS, 1 (Shanghai: Shanghai Astronomical Observatory), 51–56

Dong, J., Zhong, W., Wang, J., Liu, Q., & Shen, Z. 2018, ITAP, 66, 2044

Ford, H. A., Anderson, R., Belousov, K., et al. 2014, Proc. SPIE, 9145, 91450B

Ford, J. M., & Clark, M. H. 2002, Proc. SPIE, 4848, 12

Götz, A., Taurel, E., Pons, J., et al. 2003, TANGO a CORBA Based Control System, in ICALEPCS2003 (Gyeongju: ICALEPCS), 220–222

Nikolic, B., Prestage, R., Balser, D., Chandler, C., & Hills, R. 2007, A&A, 465, 685

Shen, T., Ibsen, J., Olguin, R., & Soto, R. 2011, Status of ALMA software, in ICALEPCS2011 (Grenoble: ICALEPCS), 487–490

Tan 2015, https://tango-controls.org/, accessed: July 19, 2023

Verdier, P., Pons, J., Poncet, F., & Leclercq, N. 2011, Tango control system management tool, in ICALEPCS2011 (Grenoble: ICALEPCS), 713–716

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv — Machine translation. Verify with original.