ROSI: a mobile robot for inspection of belt conveyor

Henrique D. Faria [*] **Fernando Lizarralde** [*] **Ramon R. Costa** [*]
Ricardo H. R. Andrade [*] **Thales H. Silva** [*]
Raphael F. S. Pereira [*] **Evelyn S. Barbosa** [*] **Filipe Rocha** [**]
André Franca [***] **Gustavo M. Freitas** [****] **Gustavo Pessin** [**]

[*] *Dept. of Electrical Eng., Federal University of Rio de Janeiro, Brazil*
[**] *Vale Institute of Technology, Ouro Preto, Brazil*
[***] *Vale S.A., Brazil*
[****] *Federal University of Minas Gerais, Brazil*

Abstract: ROSI is a mobile robot designed to inspect belt conveyor machinery in the mining industry. The proposed system is a wheeled and tracked mobile platform equipped with a robotic manipulator and several sensors to allow execution of the scheduled tasks.

Keywords: Mechatronics, Field Robotics, Mobile Robots, Mining Robotics

1. INTRODUCTION

Belt Conveyors (BCs) are widely used in the mining industry to transport all sorts of bulk material. In harbors, BCs are used to handle the incoming and outcoming material from trains to ships, as well as all needed maneuvers inside the harbor facilities.

For instance, VALE, one of the world leaders in the mining industry, has harbors with more than 120km of belt conveyor, such structures present a huge challenge for maintenance. The exposure of the belt conveyors, especially its moving parts like the idler rolls, to detrimental environments such as uncovered seashore atmosphere and being subject to the air-suspended residue of the transported material, shortens its life expectancy, requiring constant inspection and maintenance.

Given the dimensions of such facilities, and therefore the diverse range of exposure which the equipment is subject to, it is hard to predict maintenance routine because the inspection is carried out by several teams walking by the BCs assessing temperature, noise, and vibration during operation. The measured data evaluation is based on the operator experience and sensitiveness to decide if the BC idlers should be replaced or not, as well as its urgency.

It is also worth noting that as a consequence of the described procedure, the operator is exposed to that harsh environment and laborious work.

To use of autonomous robots for inspection and intervention tasks in unstructured industrial environments was recently boosted by initiatives like ARGOS challenge (Total-Website, 2015) and the DARPA Subterranean challenge (Ackerman, 2019).

For BC automated maintenance, in (Lodewijks, 2004) is discussed strategies for inspection and servicing using the concept of an automated maintenance trolley. In (Garcia et al., 2018) is proposed the use of a mobile platform with a robotic manipulator arm attached with a pack of sensors to inspect the BCs. Another robot with similar characteristics was presented in ARGOS Challenge by Vikings team (Merriaux et al., 2019), that uses a telescopic mast, instead of a 6DOF manipulator arm, and a different set of sensors since it was intended for Oil and Gas sites. Moreover, in (Staab et al., 2019) is discussed a new rail-guided robot that carries sensors for inspection of belt conveyors.

In (Chuanwei et al., 2017) is analyzed the virtual prototype of a wheel-rail inspection robot and presents simulation results indicating that although the robot presents good planar motion characteristics, it has bad performance climbing slopes. In (Cao et al., 2018), is proposed the use of a suspended inspection robot. It describes the control, positioning, sensing, and communication functions of the robot.

This paper presents ROSI (see Fig. 1), a tracked and wheeled mobile robot, with a commercial manipulator arm mounted atop to give the extra degrees of freedom needed to perform the desired inspection tasks (Freitas et al., 2018). Developed by COPPE/UFRJ in collaboration with ITV and VALE, ROSI mobile platform is designed to cope with the unstructured environments where the inspections are expected to take place.

2. GENERAL DESIGN

The robot is composed of a custom-designed mobile platform responsible for moving the system over the terrain. It is embedded with several sensors to allow it to navigate, either autonomously or teleoperated, as well as a commercial manipulator arm, at which other task-specific sensors may be mounted to execute the inspection.

⋆ This study was financed in part by CNPq, the Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior - Brasil (CAPES) - Finance Code 001, VALE S.A. and Instituto Tecnologico Vale.

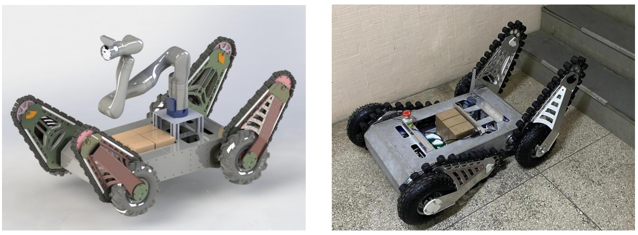

Fig. 1. ROSI - a tracked/wheeled mobile robot

ROSI description is divided along the next four sections, Mechanics, Embedded Electronics, Power Systems, and Software.

The Mechanic design section focuses on the description of the mobile platform provided that in this design the manipulator arm is considered a commercially available piece of equipment installed on the platform.

The Embedded Electronics describes the communication equipment, the computer, the electrical integration of the several sensors mounted on the platform, the robotic manipulator itself as well as the sensors attached to its end-effector.

The software architecture is based on the Robot Operating System(ROS) Melodic Morenia, a framework for robotic applications that currently runs on top of the Ubuntu 18.04 distribution of Linux.

3. MECHANIC DESIGN

The mobile platform has a hybrid traction system with wheels and tracked arms, allowing it to move around the facility with minimum structural intervention in the environment, such as the installation of platforms or slopes. It is possible to inspect long distances of BCs at a low energetic cost on wheels, while the tracked arms are used for the robot to overcome obstacles, like railways, iron pellet piles, and stairs (Lima, 2016).

The platform mechanical system is composed of:

- a chassis, designed in a combination of sheet metal and milled parts, structurally reinforced to withstand the loads and sealed to prevent ingress of water and dust;
- four traction modules that embed most of the mechanical complexity and acts as an interface between the ground and the chassis,

The mobile platform chassis is divided into four chambers, shown in the figure, where the two chambers, pointed as A, are mostly dedicated to the traction system, hosting two modules each. Chamber in yellow, identified with the letter B, gathers most of the Power System, and the fourth chamber in green stores all the internal electronic equipment, like the single-board computer and IMU, in C, and the ethernet switch and access point, in D.

On top of the chassis are mounted external sensors, such as cameras and lasers, and a KINOVA Gen3 manipulator arm. The structure is designed to handle a manipulator of up to 21kg plus 5kg of a manipulator payload. For the first indoor test, the arm is fixed over a structure made of extruded aluminum structural framing and aluminum sheet.

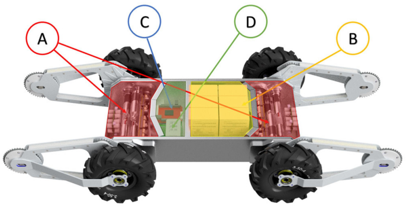

Fig. 2. Internal structure. A - Traction system, B - Communication equipment, C - Single Board Computer, D - Power system

The traction modules integrate the entire drive mechanism of the wheels and tracked arms, motors, gears, bearings, and some related electronics such as the motor controllers. Each module is driven by two motors, one acting on the drive shaft of both wheel and track systems, and the other acting on the tracked arm positioning. Through a mechanical relay, it is possible to passively engage or disengage the traction of the tracks depending on the angular position of the mobile platform arm. This mechanism avoids the use of additional electric actuators for this function, which makes the system more robust and efficient.

This mechanism allows the robot to change its driving mode passively, using the most appropriate locomotion mechanism for each terrain, without the increase of weight and energy consumption with additional equipment.

3.1 Mechanism description

The axle that drives the wheel/track passes through the axial center of the system. In addition to allowing the wheel positioning outside of the mechanism, this favors the efficiency of the drive. An eccentric shaft transmits the power to the arm via helical gears, figure 4, near the interface between the bushing and the arm. Such an arrangement provides the space at the interface between the arm and shaft for the coupler.

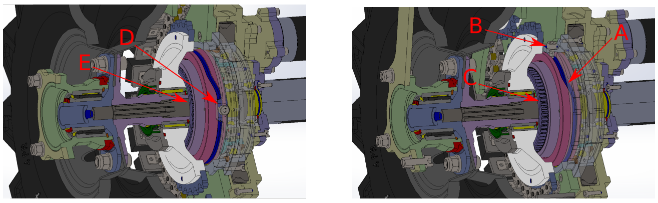

Fig. 3. Switch mechanism details

The track engagement is driven by passively shifting one part in the axial direction of the wheel axle. This displacement is the result of the interaction of two pieces, one attached to the bearing frame, and another attached to the arm frame. The first has a rolling pin (in figure 3 indicated by the letters D and B), and the second has a smooth pathway where the rolling pin slides through (in the figure 3, indicated by the letter A). The path shape is used to program the arm positions at which the drive engages and disengages the tracks, allowing the coupler (in the figure 3 indicated by the letters C and E), which is mounted on the wheel axle through a prismatic joint.

In figure 3, the pin, indicated by the letters D and B, also points the direction of the tracked arm. Note that in figure 3 (left image), the pin is in the innermost part of the track, which pushes the coupler out and engages its teeth. As the arm rotates upward, the pin travels the outermost part of the track (see figure 3 right image). As a result, the coupler is pulled inwards, disengaging the gear teeth, which are then visible in the figure 3 (letter C), thus disconnecting the traction drive from the tracks.

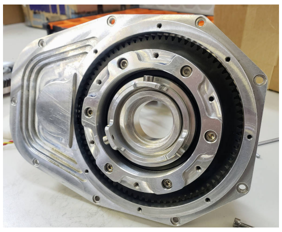

Fig. 4. Traction module view without external cover

In the coupler is mounted a gear that also provides prismatic movement relative to its counterpart on the track sprocket, which spins freely over the entire mechanism. The teeth of this pair of gears are machined to engage easier while the wheels are moving. Its interface, at the moment of coupling, has minimal collision area, and the wedge shape in each tooth induces relative positioning correction.

To prevent an unwanted engagement configuration, there is a spring in the drive mechanism that, together with the drive path geometry, prevents the drive from applying undesired force to the coupling, in the unlikely event of teeth collision by alignment. The spring only maintains a slight force until some disturbance on the track corrects the collision.

The vehicle has 4 identical traction modules that can be controlled independently, differing only in the assembly of adjacent components in the robot, so that the passive track drive mechanism engages and disengages in the correct positions. This feature has a significant impact on parts manufacturing and robot maintenance, reducing the cost and time and maintenance flexibility.

3.2 Motor Selection

To choose the motors of the traction module, it is considered that the robotic system should perform the inspection with performance parameters similar to those of human operators. The criteria considered are travel speed on flat terrain, ability to climb up- /downstairs, capacity self-lift, and rotational arm speed:

- Maximum speed on flat ground(using wheels): $1m/s$
- Maximum force at the wheel contact point to maintein in-plane movement: $74N$
- Maximum force at the tracks contact point to maintain ladder advancement 45°: $156N$
- Maximum ladder incline angle: $\phi = 45°$
- Maximum torque to self-lift over the traction arms: $T_{max} = 120Nm$. (see Fig. 5)
- Maximum angular speed of the traction arms: $45°/s$

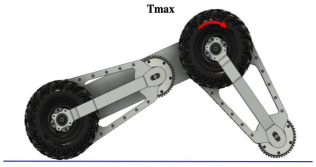

Fig. 5. Traction arm torque considering flat ground

The following physical dimensions of the robot are considered for motor specification:

(1) Robot mass: $M = 60\ Kg$
(2) Wheel radius: $R = 120\ mm$
(3) Track gear radius: $r = 60\ mm$.
(4) Rolling resistance coefficient (wheel/terrain): $\mu = 0.5$
(5) Motion efficiency on wheels: 70%
(6) Number of wheels/tracks: 4.
(7) Tracked arm length: $450mm$.

Thus, taking into consideration the specifications and physical dimensions of the robot, the following system variables were calculated/estimated:

(1) Force at the interface ground/wheel: $F_r = \mu\ Mg/4 = 74\ N$. (see Fig. 6(a))
(2) Axle torque (wheel): $T_r = F_r\ R = 9,3\ Nm$. (see Fig. 6(a))
(3) Force at the interface ground/track: $F_e = Mg/4\ (\mu \cos(\phi) + \sin(\phi)) = 156\ N$. (see Fig. 6(b))
(4) Axle torque (track): $T_e = F_e\ r = 8,5\ Nm$. (see Fig. 6(b))

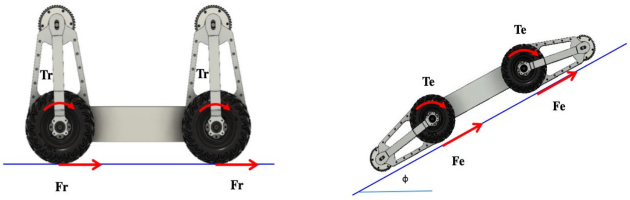

Fig. 6. Forces and torques on the interface: wheels/flat ground, on the left; and tracks/ϕ angled stairs, on the right

In figure 7 the maximum torques and speeds of each stage of the traction bearing are shown.

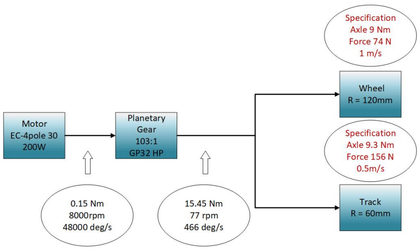

Fig. 7. Maximum torques and speeds of each stage of the traction motion

For the tracked arm part specification, it was considered a maximum self-elevating torque (estimated by simulation) of $120Nm$. (see Fig 8)

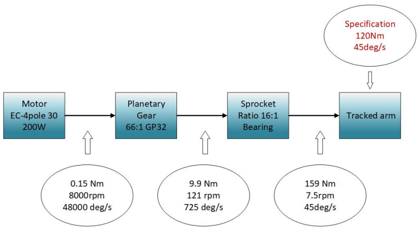

Fig. 8. maximum torques and velocities of each stage of the tracked arm motion

These values were obtained using finite element analysis and the software V-REP (Virtual Experimentation Platform) (reference), a simulation tool that allows the user to import CAD files of the robot, build a simulation scene, and handles the dynamic behavior of objects. The simulation environment was build using a simplified, but still representative, version of the mobile platform preliminary design. Its weight and other constructive characteristics were configured based on data estimated using CAD programs. From the simulation, the estimated torque curve for each joint was used to specify the torques that should be applied by the real motors.

To simplify maintenance and maximize spare part availability, the same brushless motor model was selected for the wheel traction and lever arm systems. Thus, the different torque characteristics are achieved by the correct combination of planetary gear used in each motor, and, in the track movement, the chain sprocket ratio.

4. EMBEDDED ELECTRONICS

The robot has embedded an ADLQM87PC, a PCIe/104 single-board computer (SBC), as its primary processing unit, stacked with a PCAN-PCI/104-Express card to enable the connection of up to two CAN buses to the PCI/104-Express bus on the computer.

The system has a $1Gb/s$ switch to connect all sensors, the manipulator arm module, and the Ubiquity M2 access point to the computer, allowing communication and control. The robot also has an XBEE PRO S3B radio emergency stops and other low band consuming communication, for the case of the sudden absence of WIFI connection.

Figure 9 shows the system arquitecture, where are considered the following sensors:

- IP camera to help teleoperated visual navigation
- Hokuyo UTM-30LX Scanning Rangefinder
- Velodyne VLP-16 PUCK LiDAR
- Spatial OEM GNSS/INS
- Battery Monitoring System provides individual information about each battery, temperature information of up to 4 sensors distributed inside the robot chassis and has up to 4 ADC ports available for expansion

5. POWER SYSTEMS

The robot has six lithium-ion Bren-Tronics batteries rated as 24V, 9.9Ah @ 2A, which might be enough to run the system for five hours of regular inspection tasks.

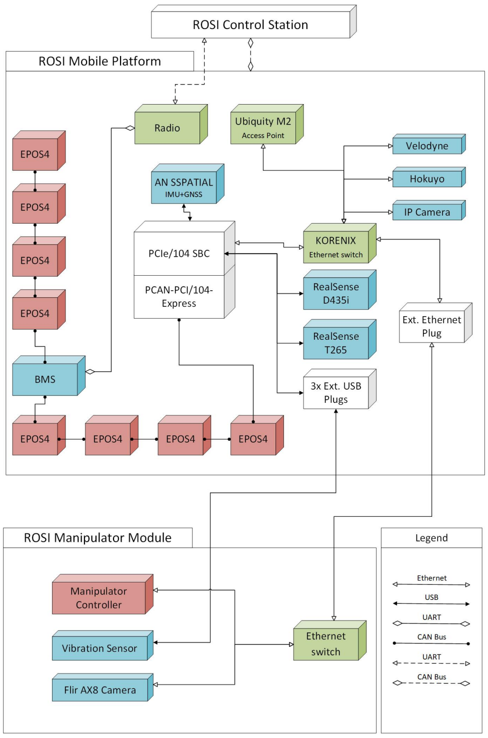

Fig. 9. ROSI Control System Architecture

On board, there are two 200W DC-DC converters, one from 24V to 5, mainly for computer power, and another from 24V to 12V. Each of these converters is mounted on the metal sheet on each side of the platform to dissipate heat. Another 200W 24V to 24V converter is mounted on the support structure for the robotic manipulator arm to power it.

Additionally, to keep modularity each of the four traction modules has a small LM2596 regulator circuit, trimmed to output 24V for the motor breaks. And to account for the inductive load of the motors, each traction section, shown in red in figure 2, has a capacitor bank of 16000uF, calculated according to (Freitas et al., 2015).

A printed circuit board (PCB) accommodate the batteries, and control the start-up process in which a solid-state relay is used to bypass the resistor placed in between the batteries and the capacitor banks. An AT90CAN64 microcontroller, programmed with a state-machine, measures the current drop before allowing the bypass. It also uses two-wire interface(TWI) to get data from each of the smart batteries, and up to 4 humidity and temperature sensors, and send them over to the computer.

6. SOFTWARE ARCHITECTURE

The nature of the robot operation leads to an increase in the complexity of the platform control software. The presence of obstacles, operators, stairs, among other challenges to navigation, requires the robot to explore all of its degrees of freedom to accomplish a secure operation.

6.1 Operation Modes

The possibility to act individually on each of the platform actuators increases the degrees of mobility, thus making it hard to operate manually. The embedded software has initially three different modes of operation, each of them restricting at a certain level the operator autonomy and automating the management of the actuator control to allow easier teleoperation. The implemented modes are:

(1) Wheeled motion
(2) Motion over caterpillar tracks
(3) Individual control of each traction arm

Robot motion: During wheeled operation, the strategy is to apply the same velocity set-point to both wheels on the same side - left, right - of the robot, thus $\omega_l = \omega_1 = \omega_3$ and $\omega_r = \omega_2 = \omega_4$ in figure 10. On the other hand, the vehicle control signal is composed by a linear velocity $v[m/s]$ and an angular velocity $\omega[rad/s]$, which are the vehicle desired velocities. From a differential driven mobile robot model, the relation between the vehicle velocities and the wheels velocities is given by:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{r}{2} \begin{bmatrix} 1 & 1 \\ -1/W & 1/W \end{bmatrix} \begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} \qquad (1)$$

where W is the distance from the center of the vehicle to the dashed line passing through the middle of the wheels (Fig. 10).

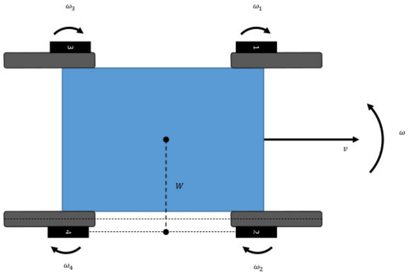

Fig. 10. Wheels and Arms representation.

In Figure 11, the coupling system between the tracks and the traction system depends on the angular position of the arms. The traction system has a mechanism that only activates the tracks when the arms are beyond a pre-specified position. To move using the tracks, as it is done on wheels, both the tracks on the same side have equal speed applied to it. In any other situation where the wheels and tracks are used simultaneously, the desired speeds of the respective motors must be set, considering the vehicle geometry, in order to achieve the same translational speed at the point of contact between the wheel/track and the ground.

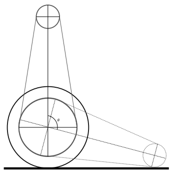

Fig. 11. Tracked arms representation.

Caterpillar tracked arms control: Arm positioning is performed by motors driven independently from the wheel/track drive system. The arms have two modes of operation. In the first, the front/rear arms work in a mirrored manner, behaving as if they were connected to the same axle. This mode is intended to facilitate the execution of tasks such as crossing obstacles and climbing up- and downstairs. In the second, the arms are entirely managed by the operator. Control of joints can be done by both position and speed, ensuring a more extensive range of arm applications such as robust orientation adjustment to terrain imperfections.

6.2 ROSI software

Developed for ROS (Quigley et al., 2009), using **C++**, the robot software includes 9 packages: *rosi_controller*, *epos*, *controller*, *robot_ps*, *keyboard*, *simple_xml_settings*, *gamepad*, *linux_devs*, and *rosi_mapper*.

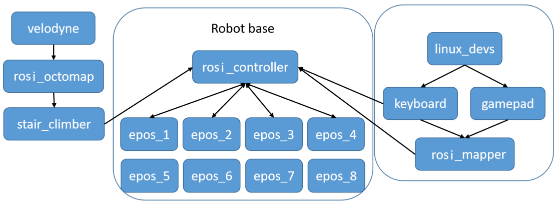

Fig. 12. Software processes schematics.

rosi_controller package: This package consists of a set of **C++** classes that perform specific functions working as an interface between the high level desired control actions and each EPOS.

- Rosi class: In this class, the physical and structural characteristics of the robot, such as width, wheel radius, and number of controlled joints, are configured
- RosiController class: This class gathers all variables and functions related to joint movement. Its primary function is to receive desired control actions, such as the translational and angular velocity of the robot, and then convert them into an operating mode and set-point for the respective motor
- ArmLogger class: To deal with the joint homing problem, when the nodelet is loaded, a pointer to an object of this class is allocated. The RosiController constructor calls a function that reads files with the last joint positions. Thus the current joint position is the sum of the starting position and the encoder position variation read by the epos. In the RosiController destructor, the last joint position is saved again
- RosiControllerNodelet class: This class starts all publishers, subscribers, service servers, and clients. Soon after, an internal loop begins that continuously calculates set-point values and calls the respective EPOS control service, thus making the robot move

controller package: The controller package manages new control messages received, checking its validity and, according to the analysis, takes the appropriate measures.

epos package: The epos package is responsible for communication between the computer and EPOS4 controllers via CAN bus. Its function is to read, write, and interpret the frames in the CAN network.

The ROSI robot has 8 independently controlled motors, so 8 *epos_nodelets* are loaded into *nodelet_manager*, one for each device. The package bundles the following classes:

- Epos class: this class encapsulates EPOS configuration and operation functions and variables. This class loads the Object Dictionary, configures the contents of transmit PDOs, which are EPOS-transmitted PDOs triggered by a sync event, and receive PDOs, which are EPOS-sent PDOs triggered by a sync event
- CANopen class: this class encapsulates functions and variables related to CAN network operation using the CANopen protocol. It manages device connection, sends and receives PDOs, SDOs, NMT states, and heartbeat
- EposNodelet class: This class is loaded into the *nodelet_manager*. It starts publishers with EPOS operational information and provides device connection and control services

7. TESTS AND RESULTS

Prior to ROSI development, some field test was held in Tubarão harbor to proof the concept of using a mobile robot with a mounted manipulator arm, to carry out the inspection task. The robot has a pack of sensors composed of a FLIR AX8 RGB and thermal camera, a Hokuyo UTM-30LX planar laser scanner, a microphone with frequency response from 20Hz to 20kHz, an Xsens MTI-G-710 IMU/GNSS, and a touching tip to transmit forces to a vibration sensor.

This preliminary test covered 3km of structure in different environments and under adverse condition, such as rain, wind, dust, and high insolation (Garcia et al., 2018).

During operation, a joystick was used to generate input for the mobile platform and the software MoveIt! (Chitta et al., 2012) was used to generate the commands to the manipulator (Garcia et al., 2018).

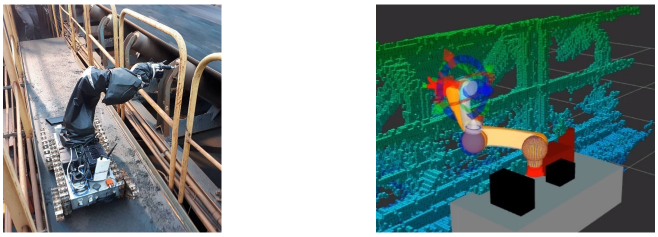

Fig. 13. Concept tests held at Tubarão Harbor

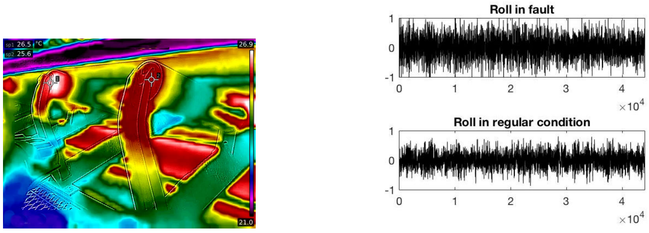

Fig. 14. Thermal camera and microphone Measurements

8. CONCLUSIONS AND FUTURE WORK

The good results obtained in this first field test motivated the development of a robot designed to handle the task of inspecting long distances of BC in a harsh environments.

This paper presented the robot ROSI, whose prototype is now in its late-stage of test. ROSI could be compared to VIKINGS (Merriaux et al., 2019) from ARGOS Challenge. ROSI locomotion system is more efficiente and is expected to perform better when moving along the BCs. VIKINGS (Merriaux et al., 2019) telescopic mast may have a simpler model and control system, but may also impose limits to the inspection task that could be easily overcome by an anthropomorphic manipulator. The robot presented by (Staab et al., 2019) was also conceived for dealing with the Belt Conveyor inspection problem. Being trail-based makes it safer for industrial operation, but adds the cost and load to the structure, and more structure to be inspected regularly. ROSI is designed to be able to reach most places a human operator would be able to safely reach by walking.

Future works include extensive field tests at VALE's facilities, and the algorithm development for intelligent detection, diagnose and repair of BC. It is also intended for ROSI to be a fully autonomous inspection robot.

REFERENCES

Ackerman, E. (2019). Darpa subterranean challenge: Meet the first 9 teams. https://spectrum.ieee.org/automaton/robotics/robotics-hardware/darpa-subt-meet-the-first-nine-teams.

Cao, X., Zhang, X., Zhou, Z., Fei, J., Zhang, G., and Jiang, W. (2018). Research on the monitoring system of belt conveyor based on suspension inspection robot. In *2018 IEEE Int. Conf. on Real-time Computing and Robotics (RCAR)*, 657–661.

Chitta, S., Sucan, I., and Cousins, S. (2012). Moveit![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1), 18–19.

Chuanwei, W., Hongwei, M., Wanfeng, S., and Gang, D. (2017). Modeling and simulation of the inspection robot for belt conveyor. In *IEEE 3rd Information Tech. and Mechatronics Eng. Conf.*, 296–299.

Freitas, G.M., Torre, M.P., Garcia, G., Rocha, F.A.S., Franca, A., Fonseca, F.R., Lizarralde, F., Costa, R., Neves, A.F., and Monteiro, J.C.E. (2018). ”robotic device and method for inspection of belt conveyor components”. Patent: BR1020180102133.

Freitas, R.S., Xaud, M.F., Marcovistz, I., Neves, A.F., Faria, R.O., Carvalho, G.P., Hsu, L., Nunes, E.V., Peixoto, A.J., Lizarralde, F., Freitas, G., Costa, R.R., From, P., Galassi, M., Derks, P.W., and Røyrøy, A. (2015). The embedded electronics and software of doris offshore robot. In *2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, 208–215.

Garcia, G., Torre, M., Monteiro, J., Franca, A., Ribeiro, F., and Freitas, G. (2018). A novel robotic inspection system for belt conveyor idlers. In *IEEE Int. Conf. on Robotics and Autom.*

Lima, L.C. (2016). *Semi-autonomous Stair Climbing for a Tracked Mobile Robot*. Master's thesis, Dept. of Electrical Eng., Federal University of Rio de Janeiro, Brazil.

Lodewijks, G. (2004). Strategies for automated maintenance of belt conveyor systems. *bulk solids handling*, 24, 16–22.

Merriaux, P., Rossi, R., Boutteau, R., Vauchey, V., Qin, L., Chanuc, P., Rigaud, F., Roger, F., Decoux, B., and Savatier, X. (2019). The vikings autonomous inspection robot: Competing in the argos challenge. *IEEE Robotics & Automation Magazine*, 26, 21–34.

Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Staab, H., Botelho, E., Lasko, D., Shah, H., Eakins, W., and Richter, U. (2019). A robotic vehicle system for conveyor inspection in mining. In *2019 IEEE Int. Conf. on Mechatronics*, 352–357.

TotalWebsite (2015). Enhanced safety thanks to the argos challenge. https://www.total.com/en/media/news/news/enhanced-safety-thanks-argos-challenge.