

Walking Machine @Home

2018 Team Description Paper

Jeffrey Cousineau et al.

École de Technologie Supérieure
1100 rue Notre-Dame Ouest, Montreal, QC, Canada H3C 1K3
<http://walkingmachine.ca>, walking@ens.etsmtl.ca,
<https://github.com/WalkingMachine>

Abstract. This paper gives details about the RoboCup@Home league team Walking Machine, from ETS University in Montreal, Canada for the next competition in Montreal, Canada in June 2018. The robot from Walking Machine, named SARA for "Système d'Assistance Robotique Autonome" (in English, Automated Robotic Assistance System), is a robot entirely built by the scientific club from ETS, mainly composed of undergraduates students. The robot is used for social interaction with humans, navigation and object manipulation. This document shows the electrical, mechanical and software properties and functionalities of SARA. It specifically emphasizes on human following, object and people recognition as well as navigation, manipulation and human-robot interaction.

1 Introduction

Walking Machine's team is a young team from Montreal, Quebec, in Canada, composed of engineering student in the field of mechanical, electrical and software engineering. We have been working on our robot for the last year in prevision of the Robocup at Home competition. As this is our second participation, we learned a lot on our first attempt last year and have made various modifications to get better results. In the past, the team went in many competitions like the Eurobot, but made the leap for the RoboCup@Home competition to get a bigger challenge.

SARA, our creation, was designed for polyvalent human-robot interaction as well as efficient navigation and object manipulation. Our robot is mounted on four mecanum wheels powered by Roboteq drives, has one arm mimicking a normal human arm, and sensors for communication and navigation. Our team has developed knowledge in object and people detection/recognition, as well as navigation using a laser scanner, odometry on the wheels and a Asus Xtion camera. All of these parts are interfaced through ROS (Robot Operating System).

2 Electrical and mechanical design of SARA

2.1 Electrical

The electrical design was mostly done using pre-built electrical assets. For example, we use Roboteq drives to power Maxon motors. The robot can run at more than 1 m/s in automatic navigation mode. The feedback of the motor speed is done with precise optical encoders and the drive control the motor speed with a PID algorithm.

SARA uses lithium-ion power cells with built-in voltage balancing. Each cell has a nominal voltage of 4.2 volts and, once linked together, offers a supply of 25.2 volts to the robot systems. This year, we conceived a new battery which is smaller than last year but gives us the possibility to be swap, so we get two times the capacity. The health of the battery system is monitored with a custom BMS (Battery Monitoring System) developed by our team. An emergency stop is linked directly to the battery to cut any power if something should go bad. Moreover, a motion stop is being developed to stop the robot during any dangerous action. A custom power distribution board was made by our team to provide enough power to the lidar, the gripper, the arm and the various embedded computer.

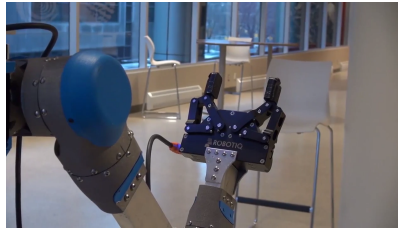


Fig. 1. Robotiq gripper

The robot uses sensors like a laser sensor for 2D mapping and leg tracking as well as Xtion camera for body tracking. The camera also provides depth perception to recognize objects and faces. We are also currently implementing the kinect 2 for future uses. The face of the robot was developed by the team and use Neopixel RGB LEDs to provide a range of possible emotions and is also linked to voice, so the LEDs will modulate the opening of the mouth according to the voice of SARA when she's talking.

2.2 Mechanical

The general design of the robot is made of a ground base, a trunk, a robotic arm and the head which include a motion sensor, a microphone and a DEL face

matrix. The basic frame is assembled with aluminum sheet metal. The shells are made from prototyping plastics, and also with acrylic sheets. We made a significant weight loss this year since we change the main desktop pc for a laptop and the battery reduction also takes away some of this weight.

One of the principal interests of the robot is its omnidirectional wheels. The 45 angle of each roller on the wheel allows frontal displacements as well as lateral movements. The only things we control are the angular speed and the rotational direction of all the DC motors connected to each wheel.

Our main base has been rethink and redesign to be smaller. This decision leads to a better navigation because we experienced some difficulties to go through doors with our old base due to his dimensions being too large. We took advantage of this redesign to build a frame with easy changing compartments for the electric system and the computer.

An innovation is also made on this robot with the titanium printed arm. Considering the high power of the arm actuators, a decision has been taken to use a high-performance factor metal to lift the biggest load as possible without any break. Moreover, this metal should be printed to make the fabrication easier. This arm was entirely designed with finite elements method to optimize the resistance and ensure required stiffness, as it is possible to see in figure 2.

This specific arm has currently five DOF (Degrees Of Freedom) but will soon have 7 with the add of two other links. This will give us more liberty and help us get to harder to reach objects. The motors used are the same as the JACO arm model from Kinova and use harmonic drive, which gives us an extreme precision. The arm, once fixed to SARA's frame, can lift up to 2kgs at full extension. Each motor is linked in series and receives its power from the one before. The arm is controlled by a JACO arm base, which distributes commands to the motors to move them.

3 Environment perception

3.1 Navigation

Most of the software development is toward the movements of the robot in its environment. For this, SARA can build a 2D map using a laser scanner. Using the simultaneous localization and mapping (SLAM) method she can build a 2D map while exploring it and simultaneously know where she is in using the odometry on her wheels, the inertial measurement unit (IMU) was also implement this year for a better localization. We also implemented, base on the robot_pose_ekf ROS package [4], a kalman filter to get a better pose estimation of our robot.

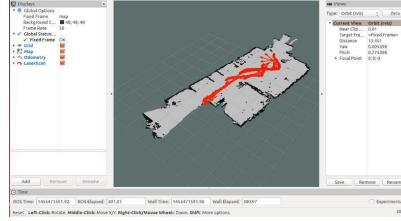


Fig. 2. Mapping

Using the sensor, the robot builds a 2D map, which can be saved and then sent to the part of the software responsible for moving the base of the robot. The 2D map updates itself often to be able to react to fast-changing environment, such as an object thrown in front of the robot.

3.2 Knowledge

This year we develop our own knowledge representation system. We called this system Wonderland¹. This system allowed us to add way point, objects and person to a representation of the world. The hold system is based on Django with a REST API so it is middle-ware agnostic.

3.3 People and objects perception

An RGB/IR Asus Xtion camera is used in parallel with the Hokuyo to build a 3D map of the environment, but at this point, we integrated only the 2D mapping and are using the Xtion camera for object and people recognition. Knowing the distances from the base of the robot to the camera, it is possible to use body tracking, which means the robot detects a human body and can analyze it to know where its head, shoulders, hands, legs and feet are. The robot will be able to interpret specific movements using this.

Face recognition is done through the Xtion. It is possible to add new people via specific services of the software. The face recognition differentiates people through banks of images of the faces and depths of the multiple points. Using statistical analysis, it goes through the bank of images when somebody is presented to the robot and assigns the most reliable face if a match is possible, else the new person is unknown.

For objects recognition, we decided to use object recognition kitchen (ORK) [2] package combine to the Kinect V2, which give us a good quality pointcloud. As undergraduates students, we tough that using ORK would be a good alternative as we don't have the required experience to developed our own object

¹ <https://github.com/WalkingMachine/wonderland/>

recognition algorithm. This package gives us very powerful tools to detect 3D objects in an environment.

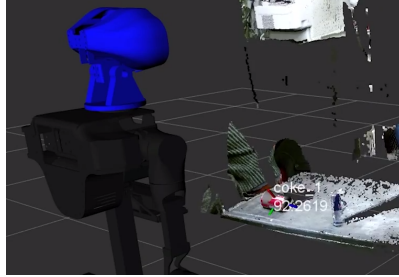


Fig. 3. Object recognition

4 Decision making

Our high-level decision making is made by a hierarchical finite state machine (HFSM). At the highest level, the HFSM wait for an order. Once an order is received, it switches to the corresponding state. It then enters a lower FSM that describes in details all the steps to execute the task.

4.1 Navigation

Autonomous navigation uses the Navigation package from ROS. This package is composed of many algorithms that allow a robot to localize itself in a map, plan a trajectory to reach a given target position and compute velocity command. A planar laser scanner and velocity readings are used to build the map. A particle filter tracks the position of the robot as it moves. Velocity commands are computed by the main algorithm and are then sent to the wheels.

4.2 Object manipulation

First of all, a model of the manipulator was created by using the Denavit-Hartenberg [7] parameters. These parameters allow to compute the transform matrix between different reference frames. Using sensor stream from the stereoscopic camera, the robot is able to locate objects and register their position in its coordinate frame. The object pose becomes the manipulator's end effector's target pose. Inverse kinematics is computed using the Cyclic Coordinate Descent (CCD) method [15]. Starting at the most distal joint and for each joint, this iterative method computes the movement required to minimize the alignment error between the joint pose to effector pose vector and the joint pose to

target pose vector. Joint position adjustments are made until the end effector is within tolerance or the maximum number of iterations is reached. Considering only one joint at the time reduces the complexity of the problem while making the algorithm robust against singularities [11].

4.3 Human-robot interaction

SARA has voice recognition package which branches directly in the decision making part. She has multiple pre-made voice command that she can understand and each of this command will need to be enabled beforehand by saying her name. Each time SARA hears her name, she will switch in command waiting mode. If the said command is available in her vocal commands tree, she will demand confirmation of it and then proceed on. At any time during an action, she can give details on what she is doing and can be stopped by the safe word STOP. This command overrides every other and acts as a software stop. She uses a speech synthesis package that we embedded in ROS known as SVOX pico2wave [5], to express herself according to her state.

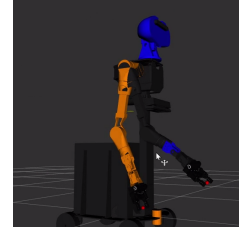


Fig. 4. Arm simulation

Giving that our robot needs to achieved a higher level of autonomy, we had to integrate some language syntactic analysis, which we call semantic analysing. Since we can't have only pre-made voice command, we decided to use LU4R [1] made by University of Rome. This gives us the possibility to extract main topics in a sentence and then, our robot can act according to these topics. For this to work, we also have to give as a parameter a description of our semantic map. This means, using KnowRob, we analyse the whole map and extract the localisation of every object in a specific room. Furthermore, we use this information to have a better understanding of our environment, which is complementary to the LU4R sentence analysis.

5 Conclusions and future work

As you can see, since last year, we made large progress, starting with the improvement of our object detection module. Also, the implementation of a semantic representation of the environment is a big step for us since it helps our robot to have a better understanding of his surrounding. Finally, the various mechanicals modifications will help for a better navigation and a better management of our robot. So SARA is now a completely autonomous system able of interaction with the human as well as navigation in her environment. Using his multiple sensors, like laser sensor, depth camera, IMU and odometry our robot can analyse his environment, recognize people and objects and navigate through obstacles. It can then interact with objects and people using his voice recognition

and voice synthesis abilities to keep track of his state. All of the requirements for the Robocup@home are met by SARA and it's a fast-evolving robot considering the team is really young and composed of undergraduates in various engineering programs. As future works, we will add a second arm to give us the possibility for two-hands manipulation. Also, we'll add a vertical translation to our body to give our robot a larger range for objects manipulation.

Robot SARA Hardware Description

Specifications for robot SARA are as follows:

SARA	
Base	Custom base with fully holonomic platform
Right arm	5 DoF custom arm made of Kinova motors
Neck	Tilt and pan unit using two Dynamixel MX-64R servo actuator
Head	Custom head made of RGB neopixels leds and Asus Xtion Pro
Gripper	Robotiq 2 fingers 85mm
Dimensions	Base : 0,61m. X 0,77m. Height : 1,68m.
Weight	~100kg
Additional sensors	Hokuyo UTM-30LX on base
Microphone	Rode microphone and Matrix Creator MEMS microphone array
Batteries	1x 24V 6 cells Lithium-ion battery
Computer	1x Lenovo p50 with 32GB RAM and nVidia Quadro M2000 4GB, 2x Raspberry Pi 3

Table 1. Robot's hardware description

Robot's Software Description

For our robot we are using the following software:

- Platform: Robotic Operating System (ROS) Kinetic on Ubuntu 16.04
- Navigation, localization and mapping: RTAB-Map, Gmapping, AMCL, DWA
- Face recognition: Cob people detection
- Speech recognition: Pocketsphinx
- Speech comprehension : LU4R
- Speech generation: Svox pico
- Object recognition: Object recognition kitchen
- Arms control: Moveit and Kinova API
- Task executors: SMACH



Fig. 5. Robot SARA

Team members

Jeffrey Cousineau, Maxime St-Pierre, Jimmy Poirier, Philippe La Madelaine, Samuel Otis, Redouane Laref, Louis-Charle Labarre, Simon Landry, Lucas Maurice, Léonore Jean-François, Nicolas Nadeau

References

1. LU4R Project - adaptive spoken Language Understanding For Robots.
2. Object Recognition Kitchen — object_recognition_core. http://wg-perception.github.io/object_recognition_core/.
3. pocketsphinx - ROS Wiki. <http://wiki.ros.org/pocketsphinx>.
4. robot_pose_ekf - ROS Wiki. http://wiki.ros.org/robot_pose_ekf.
5. Svox package : Ubuntu. <https://launchpad.net/ubuntu/precise/+source/svox/>.
6. Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.
7. J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics*, 22:215–221, 1955.
8. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
9. F.F. Sales. *SLAM and Localization of People with a Mobile Robot using a RGB-D Sensor*. Master of Science Dissertation. University of Coimbra, 2014.
10. Philipp Schillinger, Stefan Kohlbrecher, and Oskar von Stryk. Human-Robot Collaborative High-Level Control with an Application to Rescue Robotics. In *IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 2016.
11. Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, 2016.
12. Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA, 2004.
13. Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.
14. Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.
15. Stephen J. Wright. Coordinate descent algorithms. *Math. Program.*, 151(1):3–34, June 2015.