

Reconfigurable Holonomic Drive Robot in Robotics Education

Isaac Van Horn and Sherine Antoun

INTRODUCTION

Increasingly in the 21st century employers are requiring specialized knowledge in delicate practices; this is prevalent in many disciplines and industries, including medicine and mining. In the medical and health services delivery field, such as training medical and health services undergraduate students' great advantages are found in robotic application use and development. Recovery times are often cited as key reasons for robotics in laparoscopic (keyhole) surgical procedures. Furthermore, humans are usually not available for learners to practice complex medical procedures for a variety of reasons (such as injury risk or ethics). Thus, robotic humanoids are at times the most suitable stand-ins. As technology evolves, humanoid robots will be designed and constructed to mimic indications of human life breathing, sensory reactions, and heartbeat. In time, these biological facsimiles will be used as training subjects in medical procedures such as injections, surgeries, and midwifery.

Another notable application of robotics is in mining. Robots equipped with a variety of sensors to detect hazardous substances and situations can and are being used as autonomous safety inspectors, keeping miners safe without putting humans at risk... Robots are essential when scanning abandoned mine shafts and other mining operation terrains that may potentially not be safe for people. Some mine safety operation robots are tasked to explore abandoned mine shafts in search for potash leftover. The robot carries camera and an array of sensors to capture environmental data from underground. The robots use rubberized tracks for translation and are waterproof to cope with instances of mine flooding (Fig. 1).



Fig 1 Tracked survey, and inspection teleoperate robot

In industry it is not uncommon for machines to outlive the purpose for which they were commissioned; at times these perfectly serviceable devices are simply decommissioned and left neglected in storage to slowly decay. Motors, actuators,

and structural mechanics are still operational, while control system and a design purpose may have passed into obsolescence. In applied educational fields students may be reluctant to consider working with seemingly obsolete systems.

Reconfigurable systems may offer a solution to this challenge for education and training in the field of autonomous robotics. Often custom-built devices are beyond the expenditure budget constraints of a college department for one course. Mecanum wheeled robots with tractive forces at each wheel offer the opportunity to explore translation controllable in a degree of freedom that is equal to total degrees of freedom of the robot. They are also excellent platform to demonstrate and study holonomic drive theories.

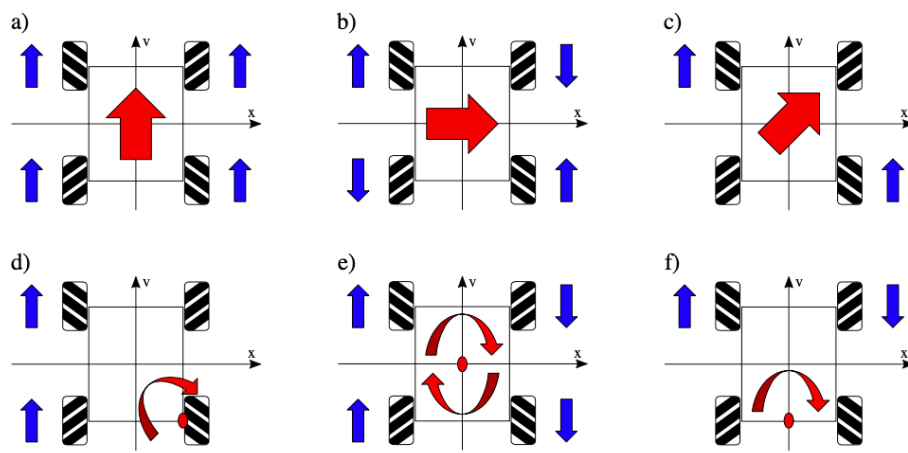


Figure 1 Directional control with tractive forces at each mecanum-wheel [3].

Holonomic drive systems apply the force of the wheel at a 45° angle to the mobile platform rather than orthogonal to its axes. Tractive force, when applied at an angle to the mobile platform, varies the magnitude of the force vectors and results in gaining translational control of the robot. This allows the mobile platform to move in any direction, while keeping its orientation at a constant azimuth. Figure 1 describes the translational attitudes achievable by various combination of tractive forces [3] [4].

Differential drive theory would require an alternate tracked platform, or standard traction wheels resulting in skid steering mobile platform. The challenge is to meet the training needs within reasonable budgetary and time (semester length) constraints.

Reconfigurable platforms offer the opportunity for students to have applied, experimental based training of various mobile robotic translation systems and controller architectures within this context. Furthermore, it offers the opportunity to explore various perception, sensor data acquisition, signal analysis, and control systems.

Purpose built robots are often characterized by monolithic control and sensory software, where the design evolved as the robot design and construction evolved over time [1]. As a teaching aid monolithic software presents many challenges in the dissemination of the knowledge on the part of teachers and in understanding the system functionality on the part of the learners. (McKerrow [2] cites PhD candidates giving up their candidature due to such challenges). In this proposed study, we explore the viability of using, as a practical teaching tool, a reconfigurable platform with concise motion control software that we developed to be reusable (not monolithic) and provide as open-source code libraries to robotic students. We seek to explore this approach as a viable teaching tool for introducing Computer Science and Engineering students to the field of autonomous mobile robotics. In the subsequent section we will examine various robotics sensors and control architectures to identify those suited for teaching at entry level. We will outline our software design with examples of the open-source code we provide to students, and we will outline the course requirements and expected outcomes as well as a success rubric to evaluate the study outcomes.

I. Robot

We reengineered the Pitsco TETRIX Max Ranger platform [6] into a tool for teaching entry level robotics. The idea was to create a teaching tool that allows students to begin applying perception, planning, and localization theory to autonomous navigation robot controllers by programming a real-life robotics platform with little to no experience in mechanical and electronic builds. The kits, while comprehensive, are still too complex for inexperienced students to start with.

II. Hardware Reconfiguration

The hardware reconfiguration required changes to the original build, as the Ranger MAX platform had been already assembled in its default configuration. The modification required an additional motor controller, as the main board only allowed for two motors and we needed to replace the original 2 motors tractive system with four new motors with pulse encoders. The rolling hardware was also replaced with four mecanum wheels in diagonally matched pairs to form a Holonomic drive system [4].

The software system for the Tetrix Prizm controller board is an Arduino UNO equivalent board and has its own code library for the Arduino IDE [5]. Upon examination of this library, it was determined to be too low level for users coding Arduino C for the first time. Writing code for the holonomic drive system and motor encoders also added more complexity, as any movement required instructions to be sent to all four motors and the encoders had to be cleared afterwards.

Basic movement functions were designed, implemented, verified and are provided as an open-source code file ("master"). With the movement functions, students can create movement paths for the rover using simple function calls that only require the desired degrees of rotation and degrees per second. This decouples coding challenges from sensor navigation and robot path control tasks.

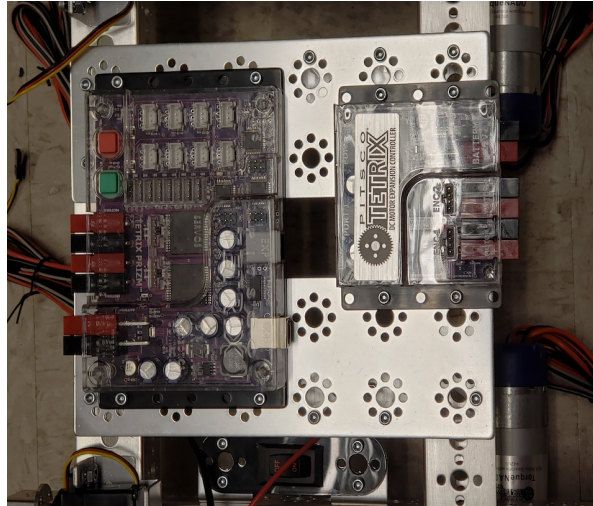


Fig 4 Reconfigured Ranger robot with expansion controller to control 4 motors

III. Design

The hardware portion of this project was relatively straight forward. Extensive documentation can be found in the “Assembly Modifications” document. The Tetrix Max platform uses very straightforward modular parts, so the platform can be readily adapted to a given use case. The code portion of the project was less straight forward, as it is going to be the main point of interaction for the students and needs to be accessible for beginners.

Because the mecanum wheel system and encoder enabled motors increased the complexity of the motor instructions for any given movement, it quickly became clear that encapsulating these commands into simple directional functions would increase the usability / versatility of the outcome.

By creating separate functions for each direction of movement with assignable distance and speed variables, it is relatively easy for students to create infinitely customizable lists of instructions for the Ranger without having to dive into individual motor commands, thus eliminating the complexities inherent in interacting with monolithic systems [1],[2]. The master code template with comments and a demo function included can be found in the open-source code file we will make available to students. This source code serves as exemplar for future custom code design. Appendix A describes an exemplar subset of the open-source code we provide to students.

IV. Hardware reconfiguration

i) DC Motor Expansion Controller

Adding the motor controller expansion required some modifications to Ranger's original build to locate a viable mount point that left all of its ports accessible. The main Arduino control board was relocated from its original position (Fig. 3) and arranged both boards (Fig. 4) leaving all ports accessible.

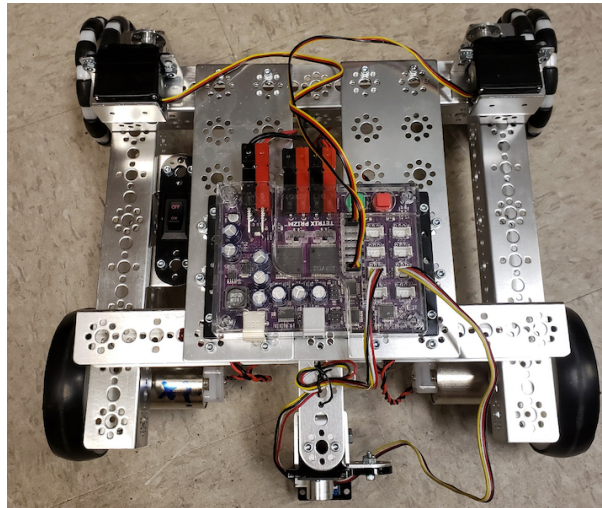


Fig 3 Ranger robot original configuration 2 traction wheels and 2 omnidirectional passive wheels.

The DC Motor expansion controller is an expansion peripheral that allows the addition of multiple DC motors to the Arduino controller. The device provides 2 DC motor output channels and two quadrature encoder inputs increasing motor control capacity to 4. It is worth noting that additional expansion controllers can be daisy-chained for a total of four motor expansion controllers connected to the Arduino main controller at one time.

ii) **Tractive System**

Four new TorqueNADO geared (60:1 ratio) motors were mounted, two in the original mount point of old motors. The TorqueNADO motors gearbox puts the drive shafts off center to its body, so their mounting was such that the drive shaft was located at the lowest point possible increasing the robot's undercarriage ground clearance.

iii) **Mecanum Wheels**

We employed a kit of TETRIX mecanum wheels from Pitsco, the sellers of the robot platform. Mounting the mecanum wheel tractive system for Holonomic drive requires pairwise orientation, so special care MUST be given to ensure these are mounted correctly. The diagonal rollers orientation must match on diagonal located wheel pairs. To avoid errors the manufacturer labels wheels in pairs 'A' or 'B'. Wheel pair 'A' and the 'B' wheel pair have different roller orientations, and they must be mounted in the correct pattern to work properly and avoid potentially damaging the motors' gearboxes due to traction resistance from the mismatched pairs.

v. **The study**

The aim of this proposed study is to quantify the positive innovative pedagogical impact of decoupling complex hardware related coding from core controller architecture design for robotic teaching practices. For this study we adopt the same model as described by Kozma and Anderson [7].

The selection of the course is based on topic and principles, in this case autonomous controller implementation for mobile robotic platforms as part of Robotic Perception and Planning applied computer science course.

The study will follow a set of common international criteria, which will be modified to accommodate the circumstances and educational needs for this course. The objective is to evaluate whether this practice plays a substantial role in achieving the goals of the course curriculum:

as educational material or infrastructure, the reconfigurable robot and the open-source code is agile for enough to suit the needs of practicing:

- the design of various control architectures
- exploring multiple modes of robotic perception and planning
- demonstrate with evidence of measurable positive student outcomes;
- test its sustainability and transferability.

VI. Data collection

We propose to adapt the model described by Kozma and Anderson [7] to suit this study. Subject to IRB ethics application approval the data collection shall be as follows:

- Course outcomes (de-identified) conducted by author, using sets of data collected from previous semesters, statistical analysis.
- Course evaluation survey with anonymous student participation.
- Historical data (the authors records of teaching the subjects sine 2010) comprising end of semester reports.
- Program materials, lesson plans, instructional materials, curriculum outline, assessment instruments, etc.
- Students' products, such as assignment submission projects, and student tree show demos, as well as teacher analyses of the submissions.

VII. Analysis method

While the goal of the analysis of the cases is to understand the impact of this innovative pedagogical practice using reconfigurable robotic technology, the analysis plan will be developed as part of a subsequent effort once the study period has concluded. The study is anticipated to run for 18 months where the course is delivered once per year yielding data from 2 cycles for analysis.

Appendix A: Open Source Robot Control Code Exemplar

```
/*
Written by Isaac Van Horn
Released as open-source code for personal and educational use.
February 9th, 2021
*/
#include <PRIZM.h>
PRIZM pzm;
EXPANSION exc;
using namespace std;

void setup()
{
  pzm.PrizmBegin();
}

//Function for straight forward motion. 90 degrees if sideways right is 0/360
degrees
void forward(int dps,int degTarget)
{
  //invert left side motors. make sure right side motors are not inverted
  pzm.setMotorInvert(2,1);
  exc.setMotorInvert(1,1,1);
  pzm.setMotorInvert(1,0);
  exc.setMotorInvert(1,2,0);
  //send instructions to the motors ("do" block) then continue looping until the
  relevant motors report not busy ("while" condition)
  do
  {
    pzm.setMotorDegrees(dps,degTarget,dps,degTarget);
    exc.setMotorDegrees(1,dps,degTarget,dps,degTarget);
  }while (pzm.readMotorBusy(1)!= 0 && pzm.readMotorBusy(2)!= 0 &&
  exc.readMotorBusy(1,1)!= 0 && exc.readMotorBusy(1,2)!= 0);
  //reset encoders so that the next instruction set can start form 0 position
  resetEnc();
}

//Function for straight backward motion. 270 degrees if sideways right is 0/360
degrees
void back(int dps,int degTarget)
{
  //invert right side motors. make sure left side motors are not inverted
  pzm.setMotorInvert(2,0);
  exc.setMotorInvert(1,1,0);
  pzm.setMotorInvert(1,1);
  exc.setMotorInvert(1,2,1);
  //send instructions to the motors ("do" block) then continue looping until the
  relevant motors report not busy ("while" condition)
  do
```

```
{
    pzm.setMotorDegrees(dps,degTarget,dps,degTarget);
    exc.setMotorDegrees(1,dps,degTarget,dps,degTarget);
}while (pzm.readMotorBusy(1)!= 0 && pzm.readMotorBusy(2)!= 0 &&
exc.readMotorBusy(1,1)!= 0 && exc.readMotorBusy(1,2)!= 0);
//reset encoders so that the next instruction set can start form 0 position
resetEnc();
}
```

Bibliography:

- [1] C. Kapoor, A Reusable Operational Software Architecture for Advanced Robotics. 1996.
- [2] P. J. McKerrow and S. M. Antoun, "A software architecture for mobile robot navigation," p. 10.
- [3] "Driving Mecanum Wheels Omnidirectional Robots." <https://www.roboteq.com/applications/all-blogs/5-driving-mecanum-wheels-omnidirectional-robots> (accessed Feb. 07, 2021).
- [4] K. P. Laby, "Holonomic platform for a robot," US6888333B2, May 03, 2005.
- [5] "TETRIX® PRIZM® Robotics Controller (W43000)." <https://www.pitsco.com/TETRIX-PRIZM-Robotics-Controller> (accessed Feb. 07, 2021).
- [6] "Pitsco Video Gallery | Watch. Learn. Do." <https://video.pitsco.com/default.aspx?vid=631> (accessed Feb. 07, 2021)
- [7] R. B. Kozma and R. E. Anderson, "Qualitative case studies of innovative pedagogical practices using ICT," *Journal of Computer Assisted Learning*, vol. 18, no. 4, pp. 387–394, 2002, doi: <https://doi.org/10.1046/j.0266-4909.2002.00250.doc.x>.