USING EULER ANGLES AND GRAVITY AS INPUTS FOR DIGITAL CONTROL OF VELOCITY

Eng. **Daniel SZÖCS** PhD candidate[1], Eng. **Andrei FENEŞAN** PhD candidate[1],
Prof. Eng. **Teodor PANĂ** PhD[1]

[1]Technical University of Cluj-Napoca, Electrical Engineering Faculty,
Department of Electrical Machines and Drives

REZUMAT. Prezenţa senzorilor de acceleraţie şi de orientare în interiorul dispozitivelor mobile creează noi oportunităţi de sesizare a mişcării, cât şi de aplicaţii utilizând valorile obţinute ca şi mărimi de comandă. Orice rotaţie a sistemului de coordinate legat de dispozitivului mobil este detectată. Folosindu-ne de această proprietate, se poate creea o aplicaţie ce comandă viteza de deplasare şi de rotaţie a unui prototip de vehicul electric diferenţial.

Cuvinte cheie: unghiuri Euler, orientare, servomotor, control viteză

ABSTRACT. The presence of acceleration and orientation sensors inside mobile devices creates new opportunities to sense motion, also to create applications that use the received values as command parameters. Any rotation of the mobile device's coordinate system is detected. Using this property, an application can be created to command forward and angular velocity of a differentially steered electric vehicle prototype.

Keywords: Euler angles, orientation, servomotor, velocity control

1. INTRODUCTION

Using orientation sensor or accelerometer values as velocity command, an application has been created.

Two continuously rotating servomotors are driving a differentially steared electric vehicle prototype in the form of a National Instruments Robotics Starter Kit™. Pulse width command for the motors is calculated from Euler angle values or from gravitational acceleration components. Pulse width modulated speed commands are sent through DIO ports to a dual motor controller.

A person rotating the mobile device would also command the vehicle to move forward and steer left or right. The transmission of data is done via a Bluetooth connection.

2. ELECTRIC VEHICLE PROTOTYPE COMPONENTS AND FUNCTIONALITY

The electric vehicle prototype NI Robotics Starter Kit™ (Fig. 1) is comprised of a sbRIO9631 embedded device[1] with AI, AO, DIO, 1M Gate FPGA. The programming is realized using the LabVIEW™ graphical development environment, programs are compiled using Xilinx™ tools and written to field programmable-gate arrays (FPGAs) on NI reconfigurable I/O hardware through the Ethernet port.

Digital I/O Port P2 is connected to a mezzanine board, which further connects to the motors controller, encoder sensors[2], ultrasonic sensors and its servomotor controller. A 12(V), 3000(mAh) Ni-MH battery serves as a power supply and a 12(V)-24(V) DC converter adjusts the power requirements for the sbRIO9631. Two motors are powered and controlled by a dual RC servo motor controller. The type of the motors is continuous-rotation servo. Rotation is sent to the wheel using a 2:1 gear ratio.

Motors are two RC servo controlled continuously rotating motors. A PWM signal is generated to control the motors: 1000(μs) pulse width corresponds to full backward speed, 1500(μs) to motor stop and 2000(μs) to full forward speed. Only the pulse width affects the speed, not the amplitude or the frequency. The duty cycle and current amperage affect torque. Amplitude of current dictates the amplitude of torque, and the duty cycle affects the jerkiness of the torque. For example, a 0.5 duty cycle means half of the time torque will be one hundred percent and half of the time it will be zero. Frequency of the signal affects the command rate, or how often one can change the command in a period of time.

The signal command is generated through ports DIO4 and DIO5 for the left and right motor respectively. Note that the motors have opposite motor movement, due to the opposite physical position of the motors shaft, this affects the PWM command.

Rotary optical encoders, one for each of the two motors, give absolute position feedback with 400 position increments. Data is sent through DIO0, DIO1 for the left motor and DIO2, DIO3 for the right.

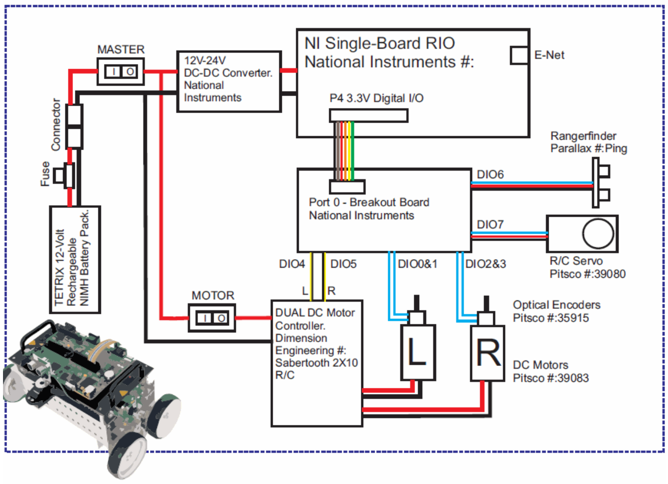

Fig. 1. LabVIEW Robotics Starter Kit™ block diagram.

In LabVIEW™, forward and angular velocity are transformed into left and right wheel velocities. A differential steer and fixed wheel frame is defined, wheel radius, wheel base and ratio are set.

THEORETICAL BASIS

Servomotor speed command

Typically, an RC servo's intended use is rotation about an axis, with a limited range of about 180 degrees. The servo motors used in the Robotics Starter Kit™ vehicle prototype are an exception to the general rule. They permit full 360 degree rotation. This special type is called continuous-rotation servo[3]. Its gear box has teeth all the way around and does not have an over-rotation prevention pin on any of the toothed wheels. In contrast, this is generally the case in typical servos to limit the movement to a set of rotation angles.

Actually, this system cannot even be categorized as a genuine servo, because it has no feedback potentiometer[4]. A feedback potentiometer gives an analog position value for the servo system to compare with the commanded position. We would call a servo a system that uses feedback to reach or maintain a parameter value. In the particular case of the Robotics Starter Kit™, the position command has no closed loop to regulate it. The closed loop is achieved in software using a PID controller and the feedback loop contains position data from digital encoders encased in the motor package. Rotary optical encoders, one for each of the two motors, give absolute position feedback with 400 position increments. Determining the precision from the number of pulses per revolution, we arive at the value of 0.9 degrees per pulse.

$$\delta_{enc} = \frac{360}{P_{rev}} = \frac{360}{400} = 0.9\ (deg/pulse) \qquad (1)$$

A typical RC servo motor is controlled by the pulse width of a PWM signal. 1500 (μs) correspond to the zero or initial position, 1000 (μs) correspond to 90 degrees counterclockwise and 2000 (μs) correspond to the same amount of rotation but in clockwise direction. The precision of the control depends on the race of the servomotor, or the allowed rotation.

$$\delta_{serv} = \frac{2000-1000}{360} = \frac{1000}{360} = 2.(7)\ \left(\frac{\mu s}{deg}\right) \qquad (2)$$

The precision is half of that of a typical servomotor with 180 degree race.

Position control is done by regulating the pulse width of the PWM signal. A new pulse position command will be given when the difference in width of the pulse has passed the threshold of 2.(7) (μs/deg).

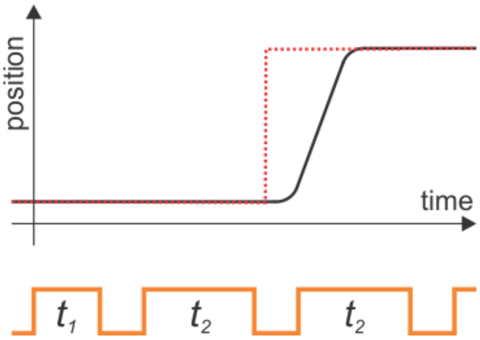

Fig. 2. Pulse width position command

To achieve speed control for a servomotor[5], the controller has to continuously change the position command in a linear time dependency for a constant speed. For this to happen, the width of the PWM signal has to increase in time until the motor does a full rotation.

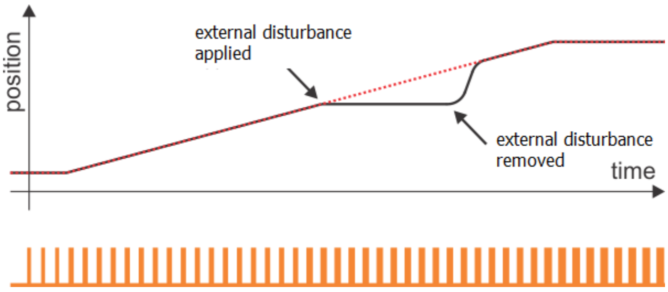

Fig. 3. Pulse width continuously varied position command

Giving the controller a command for position without a position feedback loop will create a continuous rotation of the motor, with the rotational speed directly proportional to the pulse width. A 1000 (μs) pulse width will correspond to full backward velocity, while 2000 (μs) correspond to full forward velocity. If an external force resists motor rotation, the servomotor position lags behind until the disturbance disappears. The effect of external disturbances is less obvious if the torque of the servomotor is high enough. Torque can be increased by increasing pulse amplitude or preferably pulse frequency if possible.

Forward and angular velocity to differential drive

In LabVIEW™, forward and angular velocity are transformed into forward wheel velocity (Figure 4). A differential steer and fixed wheel frame is defined, wheel radius is set to 2 (inches) or 0.0508 (m), wheel base is 14 (inches) or 0.3556 (m) and the gear ratio is 2:1.

In a differential steering vehicle[6], each wheel can move at different speeds. If both wheels move at the same speed, the vehicle moves forward. If the right wheel moves faster than the left one, the vehicle will turn left. The radius of turn is dependent of the difference in speeds. Let's assume the vehicle turns, with the left wheel having a turning radius of *r*, while the right wheel has a turning radius of *r*+*b*, where *b* represents the wheel base or distance between wheels. The distance travelled by the left wheel is

$$s_L = r\vartheta \qquad (3)$$

while the right wheel travels the distance:

$$s_R = (r + b)\vartheta \qquad (4)$$

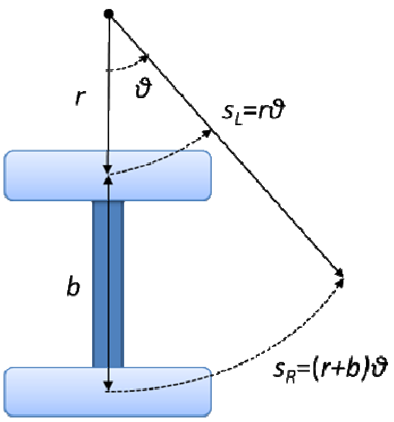

Fig. 4. Rotation radius and wheel positions

Knowing the right and left speed, we can determine the angular velocity:

$$v_a = \frac{d\vartheta}{dt} = \frac{v_R - v_L}{b} \qquad (5)$$

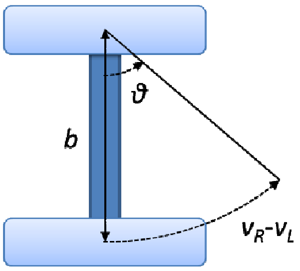

Fig. 5. Angular velocity as a function of angular position and differential speed

Heading of the vehicle is obtained by integrating the previous equation:

$$\vartheta(t) = \frac{(v_R - v_L)t}{b} + \vartheta_i \qquad (6)$$

Let's assume a fixed coordinate system with the origin in the median point of the wheel base and the *y* axis along the direction of the axle. Linear velocity vector projections in the *x* and *y* directions are given by the following equations:

$$\frac{dx}{dt} = \frac{v_R + v_L}{2}\cos\vartheta(t) \qquad (7)$$

$$\frac{dy}{dt} = \frac{v_R + v_L}{2}\sin\vartheta(t) \qquad (8)$$

Instantaneous linear velocity is then determined as:

$$v_M = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} = \frac{v_R + v_L}{2} \qquad (9)$$

In the orientation control application, the initial values are $d\theta/dt$ and v_M and the controled values are v_L and v_R.

$$v_L = v_M - \frac{b \cdot v_a}{2} \qquad (10)$$

$$v_R = v_M + \frac{b \cdot v}{2} \qquad (11)$$

All velocities in equations (5) to (11) are expressed in radians per second. If the desired unit of measurement is meters per second, the values must be multiplied by the radius of the wheel expressed in meters.

Euler angles and gravitational acceleration

Sensors are provided from an LG P970 smartphone and data is transmitted via Bluetooth. Using an Android application called SendSensor, acceleration and Euler orientation angles are transmited via Bluetooth.

The Euler angles[7] are a set of three angles that describe a number of three consecutive rotations around predefined axes. The mobile device that contains an accelerometer, also gives Euler angle values for an *XYZ* type rotation. It has a particular condition, which is that the final orientation is generated using only the first two rotations, meaning the rotation around *z* is considered null.

Rotation matrices describe rotation around a particular axis. By multiplying them, we get a generalized rotation matrix that is comprised of a rotation around x by φ degrees, then around y by θ degrees and lastly around z by ψ degrees.

$$R_{XYZ}=R_x(\psi)R_y(\theta)R_z(\varphi) \tag{12}$$

The generalized form is:

$$R_{XYZ}=\begin{bmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{bmatrix} \tag{13}$$

where, for an *XYZ* rotation type, the coefficients are:

$$\begin{cases} R_{00}=\cos\theta\cos\varphi \\ R_{01}=-\cos\theta\sin\varphi \\ R_{02}=\sin\theta \\ R_{10}=\cos\psi\sin\varphi+\cos\varphi\sin\psi\sin\theta \\ R_{11}=\cos\psi\cos\varphi-\sin\psi\sin\theta\sin\varphi \\ R_{12}=-\cos\theta\sin\psi \\ R_{20}=\sin\psi\sin\varphi-\cos\psi\cos\varphi\sin\theta \\ R_{21}=\cos\varphi\sin\psi+\cos\psi\sin\theta\sin\varphi \\ R_{22}=\cos\psi\cos\theta \end{cases} \tag{14}$$

By multiplying the rotation matrix with the gravitational vector, we get the projections of the gravitational acceleration on the three axes.

$$R_{XYZ}\times\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}=\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \tag{15}$$

In figure 6, the mobile device was oriented with the x axis oriented perpendicular to the ground and pointing downward. The variations in acceleration are caused by moving the phone up and down along its x axis. *Sensor acceleration* is the raw data from the sensor; *gravitational acceleration* has a constant value as long as the sensor moves parallel to the ground plane;

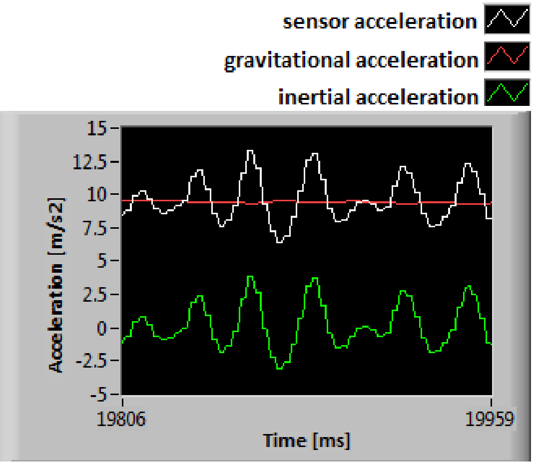

Fig. 6. Sensor acceleration in the x direction, contains raw sensor data (combined acceleration); gravitational component and inertial component.

SPEED CONTROLLER APPLICATION

Using Euler angles as inputs, an velocity controller is proposed. Rotation around the x axis by angle ψ commands the forward velocity v_m. By rotating the mobile device around the y axis by angle θ, angular speed v_a is imposed. Through a differential steer block, left and right wheel velocities are determined. Because wheel velocity command can differ from actual wheel velocity, a PID controler compares the two values and corrects the pulse width period.

Wheel velocity is calculated by deriving the wheel position curve.

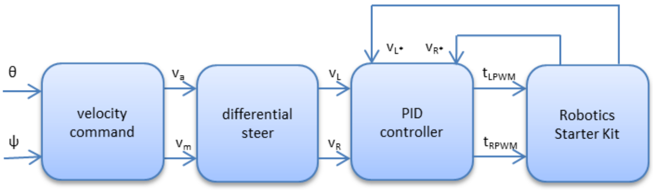

Fig. 7. Speed controller block diagram

When transforming Euler angle values to velocity, one must consider the range of allowed orientation angles. The considered range is ±90 degrees. Also, the order of rotation is important in Euler angles, different orientation results if the order of rotation is inversed. For example, if you want to command the vehicle to turn left at full backward speed, one would have to first rotate the device around the x axis 90 degrees counterclockwise so that ψ is negative and a_{yg} is –9.8

(m/s^2). Then, one must rotate the device around the *y* axis clockwise 90 degrees so that θ is positive. The device is now oriented with the *y* axis pointing down and the *x* axis pointing toward the person holding the device. If one would inverse the order of rotation, the device would be oriented such that the *x* axis would point down and the *y* axis would point right.

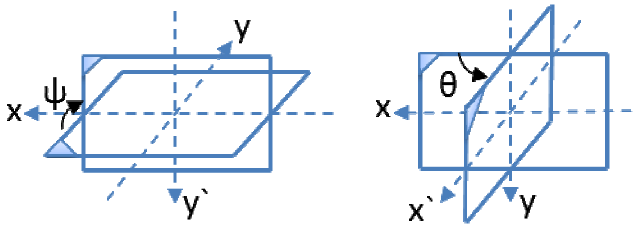

Fig. 8. Position of the device after two consecutive rotations around different axes

The vehicle moves as you rotate the device. It seems that the right and left velocities are inversed, so the control is not intuitive. It is recomended that they are switched, so that tilting the device counterclockwise should rotate the vehicle toward left.

Values for the forward and angular velocities are normalized and the forward value is multiplied by 0.8, so that the resulting values for left and right velocities are in the ±10 (rad/s) range. This is done to simplify the next step, that is transforming velocity values into time values for pulse width. The minimum value will correspond to 1000 (μs) pulse width, while the maximum value will give a 2000 (μs) pulse width command.

Another variant of the orientation controled velocity is to replace the Euler angles with gravitational acceleration components for the *x* and *y* axes.

If raw acceleration values are used, the signal also contains an inertial component, as seen in figure 6. It is recomended to determine the gravity components from the rotation matrix, using Euler angles.

inertial acceleration or linear acceleration along one axis is the acceleration generated by movement.

Another observation is that using gravitational components instead of Euler angles gives different results. For example, in figure 8, using the Euler angles the vehicle will move backwards at full speed, because both angle values are high in value. Using acceleration, the *x* component is zero and the *y* component has a maximum value, so the vehicle would move forward at full speed in a straight line.

CONCLUSIONS

✓ To obtain speed control of a servomotor it is necessary that it is of the continuous-rotation type, to permit 360 degree rotation.

✓ Only the pulse width determines the motor speed

✓ Forward and angular velocity can be transformed to left and right velocity for a differential steering, fixed wheel frame.

✓ Euler orientation angles can be used, after computational processing, as forward and angular velocity commands.

✓ Gravitational acceleration components are an alternative to Euler angles as velocity commands, with different results.

BIBLIOGRAPHY

1. *NI LabVIEW Robotics Starter Kit™ Datasheet* http://sine.ni.com/ds/app/doc/p/id/ds-217/lang/en
2. **Bräunl T.,** "Embedded Robotics", Springer, 2003.
3. *Continuous Rotation Servos and Multi-Turn Servos.* http://www.pololu.com/blog/24/continuous-rotation-servos-and-multi-turn-servos
4. *Servos and Servomotors.* http://www.pololu.com/blog/15/servo-servo-motor-servomotor-definitely-not-server
5. *RC Servo Speed Control* http://www.pololu.com/blog/22/rc-servo-speed-control
6. **Lucas G.W.,** "A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators" http://rossum.sourceforge.net/papers/DiffSteer/DiffSteer.html
7. **Slabaugh G., G.,** "Computing Euler Angles from a Rotation Matrix" http://gregslabaugh.name/publications/euler.pdf

---

About the authors

Eng. **Daniel SZÖCS,** PhD candidate
Technical University of Cluj-Napoca
email:daniel.szocs@edr.utcluj.ro

Electrical Engineer of the Technical University of Cluj-Napoca, Department of Electrical Machines and Drives. Currently is a third year PhD candidate. From 2011 has done research for the doctoral thesis at National Instruments Romania, the hardware basis for the research has been National Instrument Robotics Starter Kit™. Research topics: direct-drive electrical vehicles, anti-slip control.

Eng. **Andrei FENEŞAN,** PhD candidate
Technical University of Cluj-Napoca
email:daniel.szocs@edr.utcluj.ro

Electrical Engineer of the Technical University of Cluj-Napoca, Department of Electrical Machines and Drives. A second year PhD candidate. Research topics: direct-drive electrical vehicles, obstacle avoidance, trajectory control with LADAR and GPS technology.

Prof. Eng. **Teodor PANĂ,** PhD
Technical University of Cluj-Napoca
email:teodor.pana@edr.utcluj.ro

Professor of Microprocessor and Computer Systems and two other digital control based courses at the Technical University of Cluj-Napoca, Faculty of Electrical Engineering. Head of the Department of Electrical Machines and Drives. He has four books published in Romania, fourteen published articles in engineering publications and over 70 papers published at international conferences. Research topics: vector control systems, electric vehicle drive control, industrial robots.