Low-Level Position Control for Omni-Directional Bases in a Manufacturing Environment

Neil Abcouwer, Electrical and Computer Engineering, 2013

Carnegie Institute of Technology
Senior Honors Research, 2012-13

Introduction

Manufacturing requires fixed-position elements, known as "monuments", which grant the precision required for an engineering process. But this fixed-element approach lacks flexibility for reconfiguration of the assembly line.

Mobile robotic bases can be used to replace these monuments. Position tracking for these bases is critical to achieve precision. This presentation outlines efforts to create precise movement tracking to track omni-directional bases in intervals between global localization routines.

Background

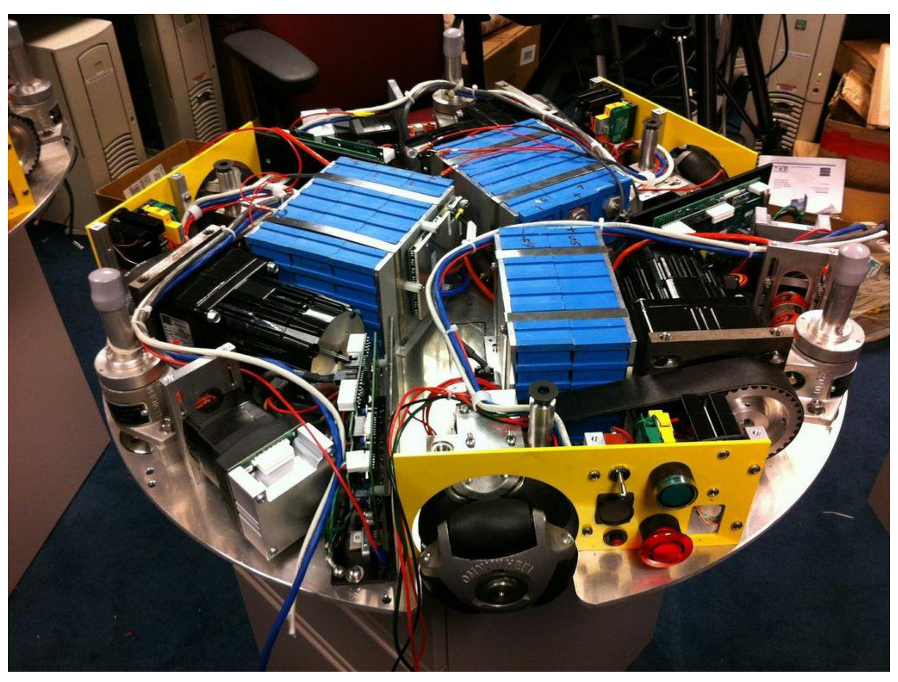
Dead reckoning is a traditional approach for position tracking in mobile robotics. The term originates from naval navigation, where sailors would use the velocity of the ship to estimate position during the day until they could use the stars at night. The same approach is used to track the position of robots in the absence of fixed-position beacons.

Omni wheels are wheels with small discs around their circumference. These wheels can exert force perpendicular to the wheel axis but are free to slide laterally. The use of three of these wheels at 120 degree angles between the axes allows a robotic base to move unconstrained in a two dimensional plane.

Research Question

The purpose of this research is to determine the feasibility of a dead reckoning position estimation approach for omnidirectional bases. Dead reckoning is well established for mobile robots with two traditional wheels, which constrain motion parallel to the wheel axes. This research attempts to adapt this approach to omnidirectional mobile robots.

Research Platform



PF3-30 Robot during Initial Construction.
Photo Credit: Gregg Podnar

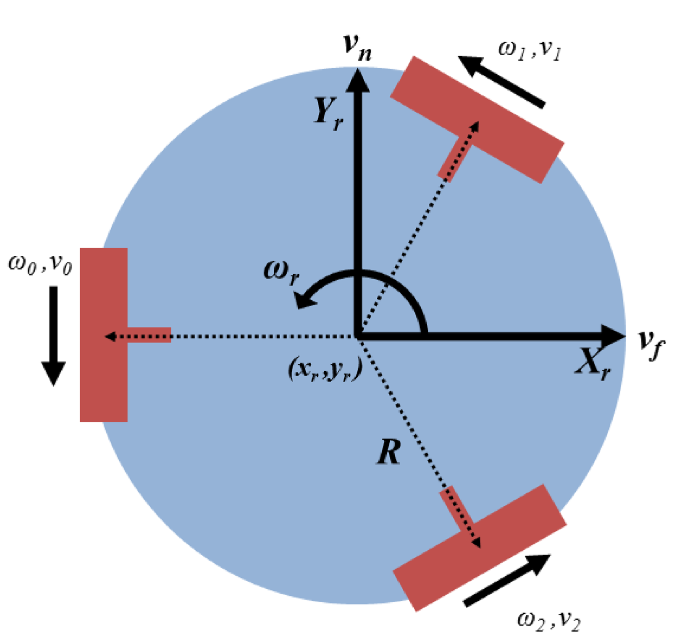
The platform used for this research is the PF3-30 mobile robot base developed by the Biorobotics Laboratory at the Robotics Institute. PF3-30 stands for Precision Flexible Factory Floor, 30 inch diameter version. These robots are research platforms for cooperative flexible manufacturing algorithms.

The PF3-30 features 3 120mm diameter omni wheels mounted on axes 120 degrees apart. Each omni wheel features 6 rollers along the circumference of the wheel which can spin freely, allowing lateral sliding.

Each wheel is driven by one motor, each of which is controlled by a motor controller board. Each wheel axis also has an attached encoder that allows tracking of the wheel position at a resolution of 5000 ticks per revolution.

Kornylak Heavy Duty 2530 Series 120mm diameter omni wheel with 6 rollers

Algorithm



Motion Model for a Three Wheel Omni-directional Mobile Base

The robot axes and variables are defined as follows. The origin of robot frame is located at the intersection of the wheel axes. The x-axis of the robot frame is defined as parallel to axis 0, with positive orientation being away from wheel 0. The y-axis is 90 degrees counterclockwise from the x-axis. Motion of the robot in the direction of its own x-axis is defined as forward velocity v_f, and movement in the direction of the y-axis is defined as normal velocity v_n. The robot is also free to rotate in the plane. This movement is defined as the robot's angular velocity, ω_r.

The geometry of the robot can be used to represent each of the wheel velocity vectors in terms of these body velocities. Define transform $T_{w,r}$ as the transformation that maps robot body velocities to wheel velocities.

$$V_w = T_{w,r} V_r$$

$$\begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \cos(270°) & \sin(270°) & R \\ \cos(150°) & \sin(150°) & R \\ \cos(30°) & \sin(30°) & R \end{bmatrix} \begin{bmatrix} v_f \\ v_n \\ w_r \end{bmatrix} = \begin{bmatrix} 0 & -1 & R \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & R \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & R \end{bmatrix} \begin{bmatrix} v_f \\ v_n \\ w_r \end{bmatrix}$$

To find the body velocities for known wheel velocities, this transformation matrix can be inverted.

$$V_r = (T_{w,r})^{-1} V_w$$

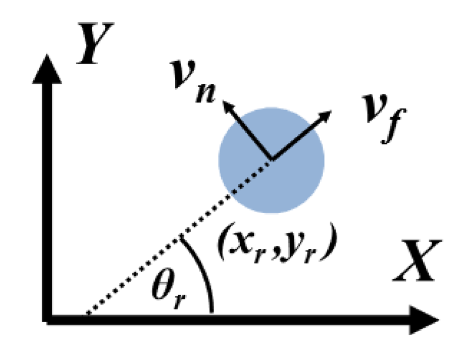
$$\begin{bmatrix} v_f \\ v_n \\ w_r \end{bmatrix} = \begin{bmatrix} 0 & -1 & R \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & R \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & R \end{bmatrix}^{-1} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ -\frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3R} & \frac{1}{3R} & \frac{1}{3R} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}$$

To find the velocity of the robot with respect to the world frame, define the rotation from the robot frame to the world frame as

$$H_r = H_z(\theta_r) = \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The world frame velocities are

$$\dot{Y}_r = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = H_r V_r = \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_f \\ v_n \\ \omega_r \end{bmatrix}$$



Robot State

The next step is to use the 4th Order Runge-Kutta Method to numerically integrate the robot's velocity to find its position.

For a given differential equation $\dot{Y} = f(t, Y)$, initial condition $Y(t_0) = Y_0$, and time step Δt, the next state Y_{n+1} can be approximated as follows:

$$Y_{n+1} = Y_n + \Delta t \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$t_{n+1} = t_n + \Delta t$$

$$k_1 = f(t_n, Y_n) \qquad k_2 = f\left(t_n + \frac{1}{2}\Delta t, Y_n + \frac{1}{2}\Delta t k_1\right)$$

$$k_3 = f\left(t_n + \frac{1}{2}\Delta t, Y_n + \frac{1}{2}\Delta t k_2\right) \qquad k_3 = f(t_n + \Delta t, Y_n + \Delta t k_3)$$

Using differential equation $\dot{Y}_r = H_r V_r$, robot position Y_r can be estimated.

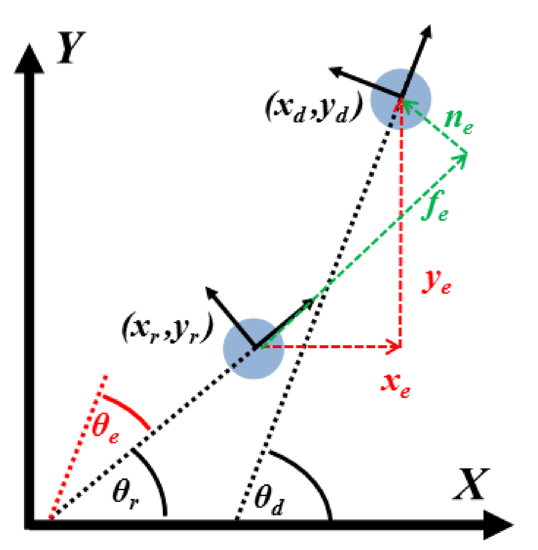
The required input to this position estimation algorithm is the angular velocity of the three omni wheels. This can be estimated from the encoders on each wheel axis. By taking the difference in encoder ticks after one time step and dividing by the time step, velocity in ticks per second can be estimated. Divide by the number of ticks per revolution (k=5000) and multiply by 2π radians to get angular velocity. Angular velocity can be converted to linear velocity at the edge of the wheel by multiplying by the wheel diameter.

Once an estimate of position is possible, a position controller can be used to drive the base to a desired position. For a desired position Y_d, a position error can be estimated:

$$E_Y = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = Y_d - Y_r = \begin{bmatrix} x_d - x_r \\ y_d - y_r \\ \theta_d - \theta_r \end{bmatrix}$$

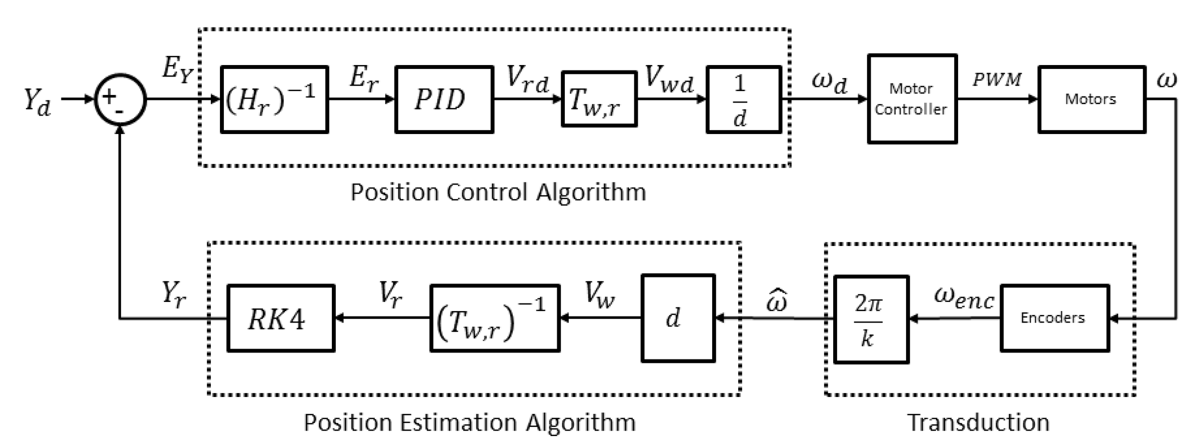
This can be transformed from the world frame to the robot frame with the inverse of the robot frame rotation in order to get the error in terms of the robot axes.

$$E_r = \begin{bmatrix} e_f \\ e_n \\ e_\theta \end{bmatrix} = (H_r)^{-1} E_Y$$



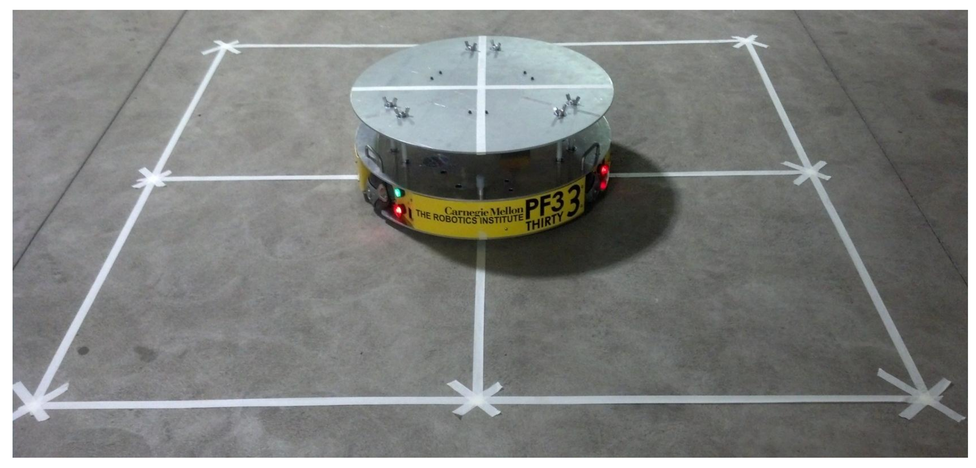
Position Error in Robot and World Frames

This error can be used as the input to a control algorithm such as Proportional-Integral-Derivative Control in order to generate V_{rd}, the desired robot velocities. This can be transformed to desired wheel velocities V_{wd} and sent to the motor controller boards. These wheel velocities will drive the position error to 0.



Overall Control Loop

Experiments



Experimental Setup, 2m x 2m Grid on Rough Concrete

To test the accuracy of this control, drift experiments were conducted. A 2m x 2m test area on rough concrete was measured and labeled as seen in the figure below. The robot was given 8 sets of waypoints to traverse, and each set was tested over three trials. After the robot finished its traversal, the final position was measured and recorded.

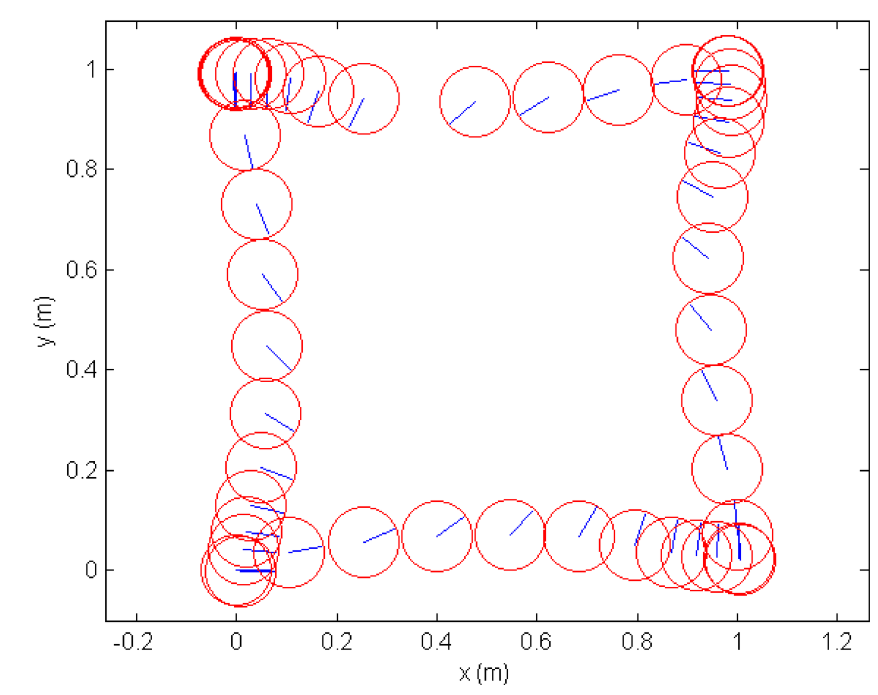
Test	Description	Start	Waypoints	Stop
1	Forward 1 meter	(0m, 0m, 0°)		(1m, 0m, 0°)
2	Left 1 meter	(0m, 0m, 0°)		(0m, 1m, 0°)
3	Counterclockwise 90°	(0m, 0m, 0°)		(0m, 0m, 90°)
4	Diagonal -1, -1	(0m, 0m, 0°)		(-1m, -1m, 0°)
5	Rotate -90 and Left 1	(0m, 0m, 0°)	(0m, 0m, -90°)	(0m, 1m, -90°)
6	FW 1, Rotate, FW 1	(0m, 0m, 0°)	(1m, 0m, 0°), (1m, 0m, 90°)	(1m, 1m, 90°)
7	Linear Square	(0m, 0m, 0°)	(1m, 0m, 0°), (1m, 1m, 0°), (0m, 1m, 0°)	(0m, 0m, 0°)
8	Rotating Square	(0m, 0m, 0°)	(1m, 0m, 90°), (1m, 1m, 180°), (0m, 1m, 270°)	(0m, 0m, 360°)

Waypoint Tests

Data

Test	Final Position 1	Final Position 2	Final Position 3	Root Mean Squared Deviation
1	(.99m, -.02m, -1°)	(1m, -.03m, 0°)	(1.01m, 0m, 0°)	(.01m, .02m, 0.6°)
2	(0m, 1.03m, -4°)	(.06m, 1.03m, -4m)	(.01m, 1m, 0°)	(.04m, .02m, 3°)
3	(0m, 0m, 90°)	(0m, 0m, 90°)	(0m, 0m, 90°)	(0m, 0m, 0°)
4	(-1.01m, -.97m, 0°)	(-1.01m, -.96m, -2°)	(-1.00m, -.97m, -3°)	(.01m, .03m, 2°)
5	(0m, 1.02m, -90°)	(0.01m, 1.01m, -89°)	(0.02m, 1.02m, -91°)	(.01m, .02m, 0.8°)
6	(1.04m, .97m, 88°)	(1.04m, .97m, 90°)	(1.02m, .97m, 90°)	(.03m, .03m, 1°)
7	(0.03m, 0.02m, 0°)	(0m, 0.01m, 0°)	(0.03m, 0.01m, 1°)	(.02m, .01m, 0.6°)
8	(0m, 0m, 360°)	(-0.02m, 0m, 359°)	(0m, 0m, 359°)	(.01m, 0m, 0.8°)

Waypoint Test Results



Recorded Robot Position, Test 8

Conclusion

The reported position for the omnidirectional base experiences about 2 to 3 cm of position drift and as much as 3 degrees of angular drift for each meter traveled. This is an acceptable amount of drift over short periods of time. Inherent problems of wheel slip, varying geometry, and the approximate nature of numerical integration make some amount of drift inevitable. Drift is reduced when the robot engages in slow, smooth motions.

In addition, the position control algorithm effectively generates wheel velocities that drive the omnidirectional base to a desired position.

Dead-reckoning alone is not sufficient for longer durations or further distances. This reality necessitates additional localization tools. Additional techniques such as LIDAR and computer vision must be used in the manufacturing environment. Dead reckoning will effectively estimate position in the intervals between these global localization techniques.

Future Research

Future work includes improvement of the firmware on the motor controller boards. PID control can be implemented on each board for fine wheel position control, which should give more consistent wheel velocities and smoother motion, which will improve position tracking results.

The accuracy of the position control can be better measured if motion capture technology is used to record the true robot position during trajectory following. Hopefully the robot can be tested in the MOCAP (Motion Capture) Laboratory this summer and the drift can be measured in real-time. This will be a large improvement over human measurement.

Acknowledgements

Thank you to my advisor, Dr. Howie Choset, and thank you to Cornell Wright, Gregg Podnar, Jim Picard, Chao Li, and Trevre Cusma.