(PAGE INTENTIONALLY LEFT BLANK)

AN ABSTRACT OF THE THESIS OF

Patrick Schmidt for the degree of Master of Science in Electrical and Computer Engineering presented on May 18, 2010.

Title: Inertial Control of a Beamforming Antenna Array for Use in Cellular Phones

Abstract approved:

____

Mario E. Magaña

This research explores the viability and effectiveness of using an inertial navigation system (INS) to control a beamforming array of microstrip patch antennas with the aim of reducing users' exposure to electromagnetic radiation. The system reduces radiated power directed toward a cellular phone user to below 10% of the total in the worst case.

Inertial Control of a Beamforming Antenna Array for Use in Cellular Phones

by
Patrick Schmidt

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented May 18, 2010
Commencement June 2010

Master of Science thesis of Patrick Schmidt presented on May 18, 2010.

APPROVED:

____

Major Professor, representing Electrical and Computer Engineering

____

Director of the School of Electrical Engineering and Computer Science

____

Dean of the Graduate School

I understand the my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

____

Patrick Schmidt, Author

ACKNOWLEDGEMENTS

Special thanks are due to my advisor, Professor Mario E. Magaña, who has provided me with guidance every step of the way and without whom this thesis could never have been completed. I would also like to thank Professor Magaña's doctoral candidate, Panupat (Pat) Poocharoen. Despite his own heavy work load, Pat always made time to answer my questions and provide wise advice.

Thanks also to Oregon State University and the Bonneville Power Administration, whose financial support over the last two years allowed my studies to continue without accumulating a great deal of debt.

Most importantly, I would like to thank my family for their material and moral support, and my fiancée Rachel for her patience and cheerful, encouraging spirit throughout this process.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF APPENDIX FIGURES

For my mom and dad. Thanks for sending me back to school.

Inertial Control of a Beamforming Antenna Array for Use in Cellular Phones

by
Patrick Schmidt

1 INTRODUCTION

1.1 MOTIVATION

Recently in the news more questions have been raised about the safety of long term cellular phone use and a possible connection with brain cancer due to the absorption of electromagnetic radiation energy [1]. Unfortunately, a confident conclusion about the effects of mobile phone use may still be years away. This is due to the fact that these effects are difficult to measure. Studies cited in the news up to now have relied on methods whose legitimacy are often questioned—such as asking a patient that has already been diagnosed with brain cancer to estimate their past phone use. A conclusive study would require large test and control populations, an objective means of measuring phone use that does not rely on how the subjects *feel* about their past phone usage and a long period of time over which to take measurements. Moreover, a conclusive study would need a means of separating the effects of electromagnetic radiation from other phenomena such as the low intensity thermal radiation, which is also given off by all cellular phones. This report assumes that, once the verdict is in, there will in fact be a need to reduce electromagnetic radiation exposure; and a method for reducing this exposure is presented.

1.2 TOPIC

Currently, to improve reception, the design of cellular phone antennas seeks to make radiation characteristics as omnidirectional as possible. In this report, a beamforming array of microstrip patch antennas, designed in [2], is used to direct radiation away from the user [2]. Beamforming is the practice of transmitting/receiving the same information over multiple antennas (the array), while varying the phase, amplitude, and/or spatial location of the individual antennas (elements) with the intention that radiation from the elements will sum constructively in a desired direction, and sum destructively (or cancel) in other directions. In the present application, beamforming will only be effective if the phone knows which way to direct (scan) the beam. If the phone cannot estimate the proper beam direction effectively, the system may expose the user to *more* radiation than a typical omnidirectional antenna.

Beamforming alone, therefore, does not tell the whole story. The phone must also track its own position and orientation with respect to the user so that it can determine the best direction to form the beam. The main purpose of this report is to suggest and test an algorithm to control the antenna array. The inputs to the array control algorithm are integrated accelerometers, which already come in most new Smartphones. The final algorithm is a simple inertial navigation system (INS) with zero-motion detection. For readers familiar with navigation, INS is an advanced electronic form of dead reckoning that accounts for rotation in addition to translation. For readers new to INS, the complete derivation is the topic of chapter 2. In short, inertial navigation works by detecting acceleration and rotation, and then calculates a position by a series of integrals and rotational transforms.

1.3 THE BASIC MEMS ACCELEROMETER

As mentioned in the section 1.2, most new Smartphones come with integrated accelerometers. More specifically, these are Micro Electrical Machine System (MEMS) accelerometers. These tiny devices have been around for decades [3] but were launched out of specialized applications and into the mainstream by their use in Nintendo Wii controllers in 2006.

Before the development of MEMS technology, accelerometers were only available in bulky, sensitive and high power packages. The proliferation of MEMS in the 1990's and 2000's has brought many exotic technologies to the domain of inexpensive mass production and miniaturization. Unlike microprocessors and other semiconductors, which work exclusively by the transport of electrons, MEMS devices have moving parts. MEMS feature sizes range from a few to hundreds of microns [5]. Fig. 1.3.1, which shows a mite next to a MEMS device, provides a good size comparison [6].

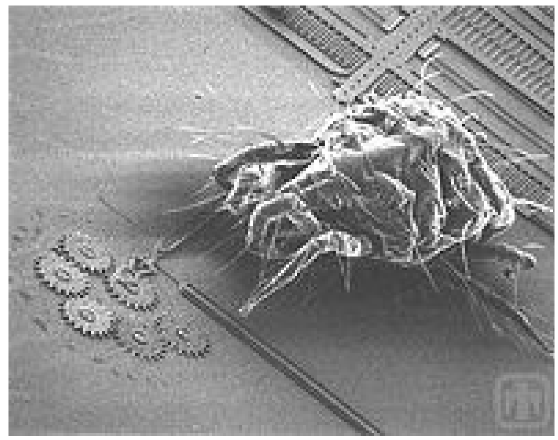


Figure 1.3.1 Mite next to a MEMS device.

MEMS accelerometers, in particular, consist of a structure suspended by springs which provide resistance to motion. Deflection of the suspended structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the spring suspended structure. Measurements of the differential capacitor voltage provides a value that is proportional to the acceleration of the device [7]. The accelerometers are designed such that acceleration is measured only along a single axis. Most MEMS accelerometer devices include two or three orthogonally oriented single-axis accelerometers in the same chip. Single-axis accelerometers have limited use, and are much harder to find than two- or three-axis accelerometers. In this paper, "accelerometer," may refer to a single axis of a three-axis accelerometer, or an entire three-orthogonally-oriented accelerometer package, depending on context.

Since the three axes are orthogonal, the accelerometer's measurements give a complete description of the force experienced by the device at any instant. Fig. 1.3.2 shows an example of a typical accelerometer package and axis orientations [7].

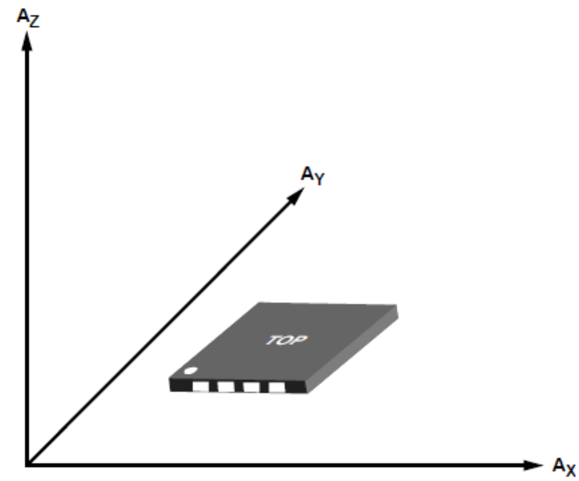


Figure 1.3.2 MEMS 3-axis accelerometer package with axis orientations shown.

1.4 WHY IS AN INS ALGORITHM NECESSARY?

MEMS accelerometers also detect the force of gravity (g), and in fact, their output values are usually given in multiples of g. For example, if the accelerometer of Fig. 1.3.2 were lying stationary on a table, the output of the device would be: $A_X = 0,\ A_Y = 0,\ A_Z = -1$. While stationary, the magnitude of the three outputs is always 1g, even though the vector $\vec{A} = \begin{bmatrix} A_X & A_Y & A_Z \end{bmatrix}^T$ may point in any direction. The acceleration reported by each axis is the dot product of that axis' basis vector and the true acceleration. Equation (1.4.1) and Fig. 1.4.1 show this relationship.

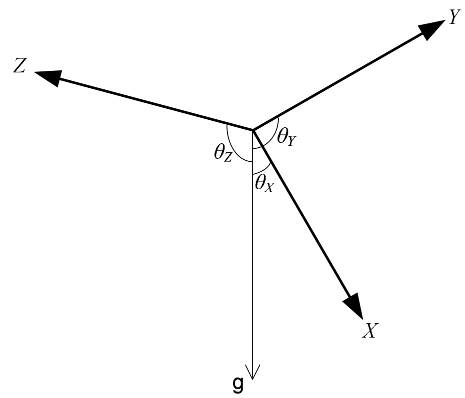


Figure 1.4.1 Gravity (g) shown with the angles between each measurement axis and itself.

In (1.4.1), $\mathbf{e}_i$ are the basis vectors representing the coordinate axes, and θ_i is the smaller of the two angles between $\mathbf{e}_i$ and $\mathbf{g}$.

$$A_i = \mathbf{e}_i \bullet \mathbf{g} = \|\mathbf{e}_i\| \|\mathbf{g}\| \cos(\theta_i) \text{ , } i \in x, y, z \tag{1.4.1}$$

Noting that the magnitudes of both $\mathbf{e}_i$ and $\mathbf{g}$ are one, the angles θ_x, θ_y, θ_z can be calculated.

$$\theta_i = \mathrm{acos}(A_i) \text{ , } i \in x, y, z \tag{1.4.2}$$

This simple calculation looks promising for finding the orientation of the accelerometer, but unfortunately the inputs and outputs are not unique to the orientation shown. Consider the case when the orientation of Fig. 1.4.1 is rotated any angle about the g axis. Clearly, the accelerometer outputs will not change. This fact is particularly damaging to the hope for a very simple solution to finding the phone's orientation. Rotation about the axis of gravity is key, because errors in this calculation quickly result in forming the radio beam directly at the user, rather than away from them. It is for this reason that a true inertial navigation algorithm is necessary.

1.5 PRIOR WORK

INS has been used for decades in aircraft, ships, spacecraft, and missiles [3]. The most notable use of INS, perhaps more recognizable by the acronym IG (inertial guidance), is used in the control of intercontinental ballistic missiles. In some traditional implementations, an inertial computer could be as large as a refrigerator and take weeks to calibrate. Modern inertial systems typically use gyroscopes to measure rotational velocity and accelerometers to measure translational acceleration. The rotational velocities are used to determine the device's orientation, which is used to convert the measured accelerations to a particular fixed frame, from which the position can be calculated through a double integration.

INS that use only accelerometers must determine rotational velocity indirectly via a set of nonlinear differential equations. This has not been done historically because this method increases algorithm complexity and results in a deterioration in performance. For miniature applications like the present one, however, gyroscopes are unfeasible because of their high power consumption. MEMS accelerometers use about a tenth as much power as their gyroscope counterparts [7].

1.6 NOTATION

Scalar, vector, and matrix equations are used throughout this report. A summary of the notations used is given in the following bulleted list.

- Vectors are denoted by **boldface** type.
- The first subscripted letter of a vector denotes the point to which the vector corresponds and the second letter gives the frame from which vector is measured.
- When a vector is split into its component parts, the first subscript letter denotes the point to which the vector corresponds, the second gives frame of reference, and the third subscript letter denotes the component part.

2 THE MATHEMATICS OF GENERAL MOTION

2.1 INTRODUCTION

The purpose of an INS algorithm is to calculate an object's motion solely from knowledge of the acceleration forces experienced by that body. As a very simple example, suppose one wants to find the final position of a point mass that begins at the origin and experiences $2m/s^2$ acceleration for 5 seconds in the x direction. The displacement is easily calculated by a double integration of the acceleration, namely:

$$d = \int_0^5 \int_0^t 2 d\tau dt = 25m . \tag{2.1.1}$$

Unfortunately, this problem is anything but simple when the object in motion is not a point mass, translates in three dimensions, and rotates. From classical mechanics, three dimensional translation and rotation is called general motion. To develop a useful INS algorithm, a thorough understanding of the mechanics of a rigid body undergoing general motion is required. The following derivation is explored to provide that background. For readers not well acquainted with the mechanics of general motion, the following derivation is an important part of this report. On the other hand, readers who are already 3D mechanics savvy may skip to the next chapter without loss of continuity.

A SPECIAL NOTE: The derivations in chapters 2 and 3 were motivated by [3] and [7], but present an implementation level of detail, which those works do not provide.

2.2 SETUP AND QUANTITIES

PROBLEM: Suppose one has a moving, rotating, rigid body. Describe the acceleration of a point (B) on the body.

O – inertial frame with basis vectors **I**, **J**, **K**

S – rotating frame with basis vectors **i**, **j**, **k**

A – origin of S

B –a point fixed in S

Ω – angular velocity of S

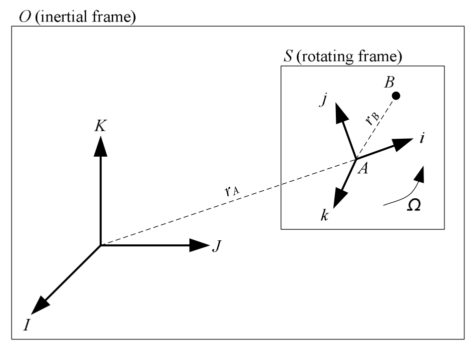


Figure 2.2.1 Fixed inertial frame O and a moving rotating frame S.

2.3 VELOCITY OF A POINT ON A RIGID BODY UNDER GENERAL MOTION

One can begin by writing an expression for the position of point B.

$$\mathbf{r}_{B,O} = \mathbf{r}_{A,O} + \mathbf{r}_{B,S} \tag{2.3.1}$$

Differentiation by time gives the velocity of B.

$$\mathbf{v}_{B,O} = \mathbf{v}_{A,O} + \mathbf{v}_{B,S} \tag{2.3.2}$$

Intuition might suggest that the right-most term in (2.3.2) is zero because S represents a *rigid* body, so $\mathbf{v}_{B,S}$ is constant. This is only partly true, and makes more sense when expanded.

$$\frac{d}{dt}\mathbf{r}_{B,S} = \frac{d}{dt}\left(r_{B,S,x}\mathbf{i} + r_{B,S,y}\mathbf{j} + r_{B,S,z}\mathbf{k}\right) \tag{2.3.3}$$

The scalar quantities $r_{B,S,x}$, $r_{B,S,y}$, and $r_{B,S,z}$ are constant, however the unit vectors **i**, **j**, and **k** vary with time as the body rotates. The differentiation, then, is applied to **i**, **j**, and **k**. If

$\mathbf{r}_{B,S}$ was not fixed—i.e. under general motion of a non-rigid body—the product rule would be needed to calculate (2.3.4). As it is, time differentiation yields

$$\frac{d}{dt}\mathbf{r}_{B,S} = r_{B,S,X}\frac{d\mathbf{i}}{dt} + r_{B,S,Y}\frac{d\mathbf{j}}{dt} + r_{B,S,Z}\frac{d\mathbf{k}}{dt}. \tag{2.3.4}$$

How does one find an expression for these basis vector derivatives? The angular velocity ($\mathbf{\Omega}$) of the rotating frame (S) provides the answer. Fig 2.3.1(a) shows the effect and orientations (right-handed) of the components of $\mathbf{\Omega}$. Since $\mathbf{\Omega}$ represents the instantaneous angular velocity of S, one can approximate the new orientation of S after a small time increment Δt. Consider, for example, what happens to **i** in time Δt (see Fig. 2.3.1(b)).

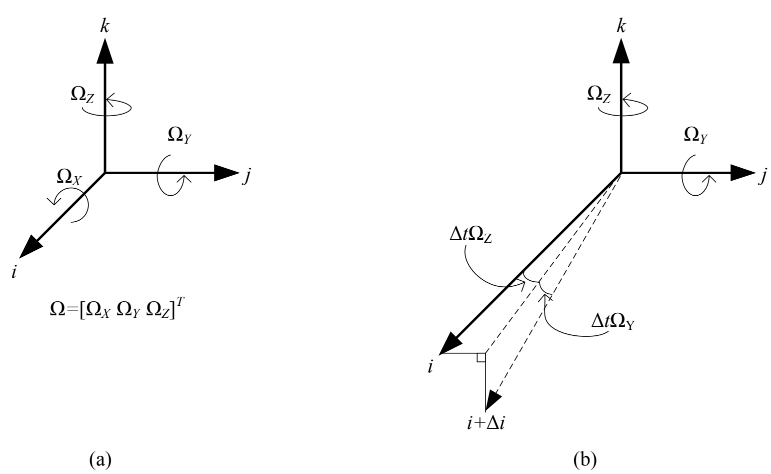


Figure 2.3.1 (a) Orientation of rotations of S. (b) The change the x-axis undergoes during small time Δt.

During time Δt, the frame rotates about the **j** axis by angle $\Omega_Y\Delta t$ and the **k** axis by $\Omega_Z\Delta t$. If the sine-small-angle approximation is used, **i**(t+ Δt) can be written as

$$\mathbf{i}(t+\Delta t) = \mathbf{i} + \Delta t\Omega_Z\mathbf{j} - \Delta t\Omega_Y\mathbf{k}\,, \tag{2.3.5}$$

and using (2.3.5) to write the derivative yields

$$\frac{d\mathbf{i}}{dt} = \dot{\mathbf{i}} = \lim_{\Delta t \to 0} \frac{\mathbf{i}(t+\Delta t) - \mathbf{i}(t)}{t + \Delta t - t} = \Omega_Z \mathbf{j} - \Omega_Y \mathbf{k}\ . \tag{2.3.6a}$$

Proceeding in precisely the same way, derivatives for **j** and **k** can be determined.

$$\dot{\mathbf{j}} = -\Omega_Z \mathbf{i} + \Omega_X \mathbf{k} \tag{2.3.6b}$$

$$\dot{\mathbf{k}} = \Omega_Y \mathbf{i} - \Omega_X \mathbf{j} \tag{2.3.6c}$$

Substituting $\dot{\mathbf{i}}, \dot{\mathbf{j}}, \text{and}\, \dot{\mathbf{k}}$ back into (2.3.4) and collecting basis vectors gives

$$\frac{d}{dt}\mathbf{r}_{B,S} = \left(\Omega_Y r_{B,S,Z} - \Omega_Z r_{B,S,Y}\right)\mathbf{i} + \left(\Omega_Z r_{B,S,X} - \Omega_X r_{B,S,Z}\right)\mathbf{j} + \left(\Omega_X r_{B,S,Y} - \Omega_Z r_{B,S,X}\right)\mathbf{k}\ . \tag{2.3.7}$$

One may recognize (2.3.7) as the cross product $\mathbf{\Omega} \times \mathbf{r}_{B,S}$, a fact which will be exploited later. For now, consider the nature of **i**, **j**, and **k**—they are basis vectors for frame S written in terms of frame O. To put it more plainly, **i** is a vector in O. If (2.3.7) is written in this way,

$$\frac{d}{dt}\mathbf{r}_{B,S} = \left(\Omega_Y r_{B,S,Z} - \Omega_Z r_{B,S,Y}\right)\begin{bmatrix} i_I \\ i_J \\ i_K \end{bmatrix} + \left(\Omega_Z r_{B,S,X} - \Omega_X r_{B,S,Z}\right)\begin{bmatrix} j_I \\ j_J \\ j_K \end{bmatrix} + \left(\Omega_X r_{B,S,Y} - \Omega_Z r_{B,S,X}\right)\begin{bmatrix} k_I \\ k_J \\ k_K \end{bmatrix}, \tag{2.3.8}$$

It becomes apparent that the expression can be factored into a matrix multiplication.

$$\frac{d}{dt}\mathbf{r}_{B,S} = \begin{bmatrix} i_I & j_I & k_I \\ i_J & j_J & k_J \\ i_K & j_K & k_K \end{bmatrix}\begin{bmatrix} \Omega_Y r_{B,S,Z} - \Omega_Z r_{B,S,Y} \\ \Omega_Z r_{B,S,X} - \Omega_X r_{B,S,Z} \\ \Omega_X r_{B,S,Y} - \Omega_Z r_{B,S,X} \end{bmatrix} \tag{2.3.9}$$

The 3x3 matrix in (2.3.9) is a rotation matrix that maps coordinates in S onto O. Let this matrix be designated by $\underline{\underline{F}}$.

$$\frac{d}{dt}\mathbf{r}_{B,S} = \underline{\underline{F}}\left(\mathbf{\Omega} \times \mathbf{r}_{B,S}\right) \tag{2.3.10}$$

Substituting (2.3.10) into (2.3.2) gives the final equation for the velocity of point B.

$$\mathbf{v}_B = \mathbf{v}_A + \underline{\underline{F}}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right) \tag{2.3.11}$$

2.4 ACCELERATION OF A POINT ON A RIGID BODY UNDER GENERAL MOTION

Equation (2.3.11) must be time differentiated once more to get the acceleration.

$$\mathbf{a}_B = \mathbf{a}_A + \frac{d}{dt}\underline{\underline{F}}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right) \tag{2.4.1}$$

Expanding the term on the right in (2.4.1) with the product rule yields,

$$\mathbf{a}_B = \mathbf{a}_A + \dot{\underline{\underline{F}}}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right) + \underline{\underline{F}}\left(\frac{d}{dt}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\right). \tag{2.4.2}$$

The second and third terms on the right in (2.4.2) require some simplification, and the quantity $\dot{\underline{\underline{F}}}$ will turn out to be of particular importance in chapter 3. For this reason, the terms $\dot{\underline{\underline{F}}}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)$ and $\underline{\underline{F}}\left(\frac{d}{dt}\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)$ will be taken up separately in the following two sections.

2.4.1 $\dot{\underline{\underline{F}}}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)$

First consider $\dot{\underline{\underline{F}}}$ alone, its expansion gives

$$\dot{\underline{\underline{F}}} = \begin{bmatrix} \dot{i}_I & \dot{j}_I & \dot{k}_I \\ \dot{i}_J & \dot{j}_J & \dot{k}_J \\ \dot{i}_K & \dot{j}_K & \dot{k}_K \end{bmatrix}. \tag{2.4.1.1}$$

Equation (2.4.1.1) shows that the columns of $\dot{\underline{\underline{F}}}$ are the time derivatives of the axes of S, so $\dot{\underline{\underline{F}}}$ can also be written as

$$\dot{\underline{\underline{F}}} = \begin{bmatrix} \dot{\mathbf{i}} & \dot{\mathbf{j}} & \dot{\mathbf{k}} \end{bmatrix}. \tag{2.4.1.2}$$

Substitution of (2.3.4) into (2.4.4.2) yields

$$\dot{\underline{\underline{F}}} = \begin{bmatrix} \Omega_Z\mathbf{j} - \Omega_Y\mathbf{k} & -\Omega_Z\mathbf{i} + \Omega_X\mathbf{k} & \Omega_Y\mathbf{i} - \Omega_X\mathbf{j} \end{bmatrix}. \tag{2.4.1.3}$$

Though it may not be obvious, $\begin{bmatrix} \Omega_Z\mathbf{j} - \Omega_Y\mathbf{k} & -\Omega_Z\mathbf{i} + \Omega_X\mathbf{k} & \Omega_Y\mathbf{i} - \Omega_X\mathbf{j} \end{bmatrix}$, can be factored into a matrix product as

$$\dot{\underline{\underline{F}}} = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \end{bmatrix} \begin{bmatrix} 0 & \Omega_Z & -\Omega_Y \\ -\Omega_Z & 0 & \Omega_X \\ \Omega_Y & -\Omega_X & 0 \end{bmatrix}. \tag{2.4.1.4}$$

Notice that $\begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \end{bmatrix}$ is actually the matrix $\underline{\underline{F}}$ in disguise. Let the 3x3 matrix of Ωs be denoted by $\underline{\underline{\Omega}}$; then (2.4.1.4) can be rewritten as

$$\dot{\underline{\underline{F}}} = \underline{\underline{F}}\,\underline{\underline{\Omega}}. \tag{2.4.1.5}$$

Substituting (2.4.1.4) back into the total expression gives

$$\dot{\underline{\underline{F}}}\left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S}\right) = \underline{\underline{F}}\,\underline{\underline{\Omega}}\left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S}\right). \tag{2.4.1.6}$$

$\underline{\underline{\Omega}}$ has a useful property called skew-symmetry [4], that allows this matrix to be written as a cross product with the factor to the right of it. Moreover, the vector corresponding to $\underline{\underline{\Omega}}$ is precisely the vector **Ω**. This property can be confirmed by direct evaluation. Using skew-symmetry to write $\underline{\underline{\Omega}}$ as a vector, the expression becomes

$$\dot{\underline{\underline{F}}}\left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S}\right) = \underline{\underline{F}}\left(\boldsymbol{\Omega} \times \left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S}\right)\right). \tag{2.4.1.7}$$

2.4.2 $\underline{\underline{F}}\left(\frac{d}{dt}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\right)$

This simplification begins by expanding the cross product and distributing the differentiation of the expression.

$$\underline{\underline{F}}\left(\frac{d}{dt}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\right)=+\underline{\underline{F}}\begin{bmatrix}\frac{d}{dt}\left(\Omega_Y r_{B,S,Z}-\Omega_Z r_{B,S,Y}\right)\\ \frac{d}{dt}\left(\Omega_Z r_{B,S,X}-\Omega_X r_{B,S,Z}\right)\\ \frac{d}{dt}\left(\Omega_X r_{B,S,Y}-\Omega_Z r_{B,S,X}\right)\end{bmatrix} \tag{2.4.2.1}$$

Next carry out the differentiation.

$$\underline{\underline{F}}\left(\frac{d}{dt}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\right)=\underline{\underline{F}}\begin{bmatrix}\dot{\Omega}_Y r_{B,S,Z}-\dot{\Omega}_Z r_{B,S,Y}+\Omega_Y\dot{r}_{B,S,Z}-\Omega_Z\dot{r}_{B,S,Y}\\ \dot{\Omega}_Z r_{B,S,X}-\dot{\Omega}_X r_{B,S,Z}+\Omega_Z\dot{r}_{B,S,X}-\Omega_X\dot{r}_{B,S,Z}\\ \dot{\Omega}_X r_{B,S,Y}-\dot{\Omega}_Z r_{B,S,X}+\Omega_X\dot{r}_{B,S,Y}-\Omega_Z\dot{r}_{B,S,X}\end{bmatrix} \tag{2.4.2.2}$$

Recall that $\mathbf{r}_{B,S}$ is constant, so any quantity involving a derivative of $\mathbf{r}_{B,S}$ is zero. Equation (2.4.2.2) becomes

$$\underline{\underline{F}}\left(\frac{d}{dt}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\right)=\underline{\underline{F}}\begin{bmatrix}\dot{\Omega}_Y r_{B,S,Z}-\dot{\Omega}_Z r_{B,S,Y}\\ \dot{\Omega}_Z r_{B,S,X}-\dot{\Omega}_X r_{B,S,Z}\\ \dot{\Omega}_X r_{B,S,Y}-\dot{\Omega}_Z r_{B,S,X}\end{bmatrix}. \tag{2.4.2.3}$$

One can see that the right-most term is a cross product, so the expression can be written as

$$\underline{\underline{F}}\left(\frac{d}{dt}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\right)=\underline{\underline{F}}\left[\dot{\boldsymbol{\Omega}}\times\mathbf{r}_{B,S}\right], \tag{2.4.2.3}$$

which concludes the simplification of $\underline{\underline{F}}\left(\frac{d}{dt}\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\right)$.

2.5 THE FINAL EXPRESSION FOR a_B

Using the simplified forms of (2.4.1.7) and (2.4.2.3) in (2.4.2), and factoring $\underline{\underline{F}}$ to the left gives

$$\mathbf{a}_{B} = \mathbf{a}_{A} + \underline{\underline{F}}\left(\boldsymbol{\Omega}\times\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right) + \dot{\boldsymbol{\Omega}}\times\mathbf{r}_{B,S}\right). \tag{2.5.1}$$

Equation (2.5.1) is the acceleration of point B in the units of O, alternatively, one may need to know acceleration in the units of S. This can be accomplished by simply applying the reverse transform of rotation $\underline{\underline{F}}$ to (2.5.1).

$$\mathbf{a}_{B,S} = \underline{\underline{F}}^{T}\mathbf{a}_{A} + \boldsymbol{\Omega}\times\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right) + \dot{\boldsymbol{\Omega}}\times\mathbf{r}_{B,S} \tag{2.5.2}$$

Equation (2.5.1) provides a basis for the understanding of the acceleration that can be measured at a fixed point on a rigid body undergoing general motion. The goal, however, is to find the position and orientation of the rigid body that moves with translational acceleration $\mathbf{a}_A$ and angular velocity $\boldsymbol{\Omega}$.

There are six variables in general motion—three relating to translation in any of the coordinate directions, and three relating to rotation about the three coordinate axes. Clearly, for six variables of motion, at least six accelerometer measurements will be required to solve the general motion. In sections 2.6 to 2.9, a method will be considered whereby a system of N≥6 accelerometers placed strategically throughout the rigid body will provide N instances of (2.5.2). By solving simultaneously for $\mathbf{a}_{A,S}$ and $\dot{\boldsymbol{\Omega}}$, a system of nonlinear differential equations can be constructed and one can attempt to find solutions for the position and orientation of the cellular phone in three dimensional space.

2.6 INCLUDING AN ACCELEROMETER

In chapter 1, it was decided to consider only MEMS devices that include three orthogonally oriented single-axis accelerometers in one package. Fig. 1.3.2 showed this arrangement. Even though accelerometer measurements will therefore come in triplets, it will be useful to consider each single-axis accelerometer independently. Equation (1.4.1) reminds us that a single-axis accelerometer measures a value equal to the dot product of the true acceleration and the orientation of the accelerometer.

In the context of the present discussion, if point B, fixed in frame S, is undergoing general motion, its acceleration is given by (2.5.1). If there is a single-axis accelerometer located at B, it will only measure the projection of $\mathbf{a}_{B,S}$ onto that measuring axis. Denoting the measurement axis of the single-axis accelerometer at point B by $\boldsymbol{\theta}_{B,S}$, the value measured $a_{M,S}$ is precisely

$$a_{M,S} = \mathbf{a}_{B,S} \cdot \boldsymbol{\theta}_{B,S} . \tag{2.6.1}$$

Substituting (2.5.2) into (2.6.1) gives

$$a_{M,S} = \left(\underline{\underline{F}}^T \mathbf{a}_A + \boldsymbol{\Omega} \times \left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S} \right) + \left(\dot{\boldsymbol{\Omega}} \times \mathbf{r}_{B,S} \right) \right) \cdot \boldsymbol{\theta}_{B,S} . \tag{2.6.2}$$

$\underline{\underline{F}}^T \mathbf{a}_A$ is the acceleration of the origin of S, in the units of S, so one can also write $\mathbf{a}_{A,S} = \underline{\underline{F}}^T \mathbf{a}_A$. By distributing the dot product, and making use of the dot-cross-product property $\mathbf{a} \times \mathbf{b} \cdot \mathbf{c} = \mathbf{a} \cdot \mathbf{b} \times \mathbf{c}$ in the right most term, (2.6.2) can be written as

$$a_{M,S} = \boldsymbol{\theta}_{B,S} \mathbf{a}_{A,S} + \left(\mathbf{r}_{B,S} \times \boldsymbol{\theta}_{B,S} \right) \cdot \dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S} \right) \cdot \boldsymbol{\theta}_{B,S} . \tag{2.6.3}$$

$\boldsymbol{\theta}_{B,S}$ and $\mathbf{r}_{B,S}$ are known and constant, so the first two terms in (2.6.3) are linear and can be written as a vector-vector product, changing the form of (2.6.3) to

$$a_{M,S} = \left[\left(\mathbf{r}_{B,S} \times \boldsymbol{\theta}_{B,S} \right) \quad \boldsymbol{\theta}_{B,S} \right] \begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \mathbf{a}_{A,S} \end{bmatrix} + \boldsymbol{\Omega} \times \left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S} \right) \cdot \boldsymbol{\theta}_{B,S} . \tag{2.6.4}$$

Recall that $a_{M,S}$ is a scalar and is the actual value measured by a single accelerometer. The steps from (2.6.2) to (2.6.4) were an effort to put $a_{M,S}$ into a form whereby multiple accelerometers providing many different versions of $a_{M,S}$ could be used to solve for the motion. Now consider the case where there are N accelerometers in the rigid body, (2.6.4) provides the acceleration of the i^{th} accelerometer. So that one could write

$$a_{M,S,i} = \left[\left(\mathbf{r}_{B,S,i} \times \boldsymbol{\theta}_{B,S,i} \right) \quad \boldsymbol{\theta}_{B,S,i} \right] \begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \mathbf{a}_{A,S} \end{bmatrix} + \boldsymbol{\Omega} \times \left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S,i} \right) \cdot \boldsymbol{\theta}_{B,S,i} \text{ , } i = 1, \cdots, N . \tag{2.6.5}$$

Writing (2.6.5) as a system

$$\underline{a_{M,S}} = \left[\underline{\left(\mathbf{r}_{B,S} \times \boldsymbol{\theta}_{B,S} \right)} \quad \underline{\boldsymbol{\theta}_{B,S}} \right] \begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \mathbf{a}_{A,S} \end{bmatrix} + \underline{\boldsymbol{\Omega} \times \left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S} \right) \cdot \boldsymbol{\theta}_{B,S}} . \tag{2.6.6}$$

The underbars in (2.6.6) represent a non-spatial vectors of the quantity, so for example, $\underline{\left(\mathbf{r}_{B,S} \times \boldsymbol{\theta}_{B,S} \right)}$ is a vector of cross products with dimensions 3xN. Equation (2.6.6) does not add any physical understanding to the situation, but a system of equations is required to explicitly solve the system since there are six unknowns. Additionally, though the purpose of the research is to solve the general motion from measurements, it is only possible to quantify the system's performance if the real motion is known. Equation (2.6.6) is the driving function that takes known test motions and converts them into accelerometer inputs.

2.7 SOLVING THE ACCELERATIONS

Equation (2.6.6) represents another milestone in the representation of general motion, but the goal is to solve for the general motion, not the accelerations. Therefore, (2.6.6) needs to be solved for $\begin{bmatrix} \dot{\mathbf{\Omega}} & \mathbf{a}_{A,S} \end{bmatrix}^T$. To accomplish this, first let

$$\underset{=}{J} = \begin{bmatrix} \underline{\left(\mathbf{r}_{B,S} \times \mathbf{\theta}_{B,S}\right)} & \underline{\mathbf{\theta}_{B,S}} \end{bmatrix}. \qquad (2.7.1)$$

If the 6xN matrix $\underset{=}{J}$ has rank six (note that N≥6 for this to be possible), it must have an inverse or pseudoinverse, which can be used to solve for $\begin{bmatrix} \dot{\mathbf{\Omega}} & \mathbf{a}_{A,S} \end{bmatrix}^T$. Assume $\underset{=}{J}$ has an inverse and define matrix $\underline{\underline{Q}}$ as

$$\underline{\underline{Q}} \triangleq \underset{=}{J}^{-1}. \qquad (2.7.2)$$

Apply $\underline{\underline{Q}}$ to (2.6.6) and moving the non-linear term to the other side of the equation.

$$\begin{bmatrix} \dot{\mathbf{\Omega}} \\ \mathbf{a}_{A,S} \end{bmatrix} = \underline{\underline{Q}}\underline{a_{M,S}} - \underline{\underline{Q}}\underline{\mathbf{\Omega} \times \left(\mathbf{\Omega} \times \mathbf{r}_{B,S}\right) \cdot \mathbf{\theta}_{B,S}} \qquad (2.7.3)$$

If the second term on the right in (2.7.3) is nonzero, the system is nonlinear, and therefore will likely not have a simple closed form solution. Since this system is a discrete one working in real time, however, it will not be necessary to consider closed form solutions, as the INS algorithm will simply integrate $\begin{bmatrix} \dot{\mathbf{\Omega}} & \mathbf{a}_{A,S} \end{bmatrix}^T$, using a one sample delayed version of $\mathbf{\Omega}$ to provide the feedback, to find the motion.

2.8 SOLVING THE ROTATION

With $\dot{\mathbf{\Omega}}$ now known, it is possible to find the orientation—i.e. $\underline{\underline{F}}$—of the cellular phone. $\dot{\mathbf{\Omega}}$ is the angular acceleration of frame S about its own axes. Integration of $\dot{\mathbf{\Omega}}$ yields the angular velocity of S, from which $\underline{\underline{F}}$ can be formed.

$$\mathbf{\Omega} = \int \dot{\mathbf{\Omega}} dt + \mathbf{\Omega}_0 \tag{2.8.1}$$

An intuitive way to find the orientation of the phone from $\mathbf{\Omega}$ might be to integrate once more, and then construct a rotation matrix from the rotation angles **Or**. This will not work, however, because rotations do not commute. That is, if one integrated $\mathbf{\Omega}$ and found

$$\mathbf{Or} = \int \mathbf{\Omega} dt + \mathbf{Or}_0 = \begin{bmatrix} Or_X & Or_Y & Or_Z \end{bmatrix}, \tag{2.8.2}$$

and then tried to form $\underline{\underline{F}}$ from the components of **Or**, the result would be incorrect. This is because the information about the *order* of rotation is lost when integrating $\mathbf{\Omega}$. The key to finding $\underline{\underline{F}}$ from $\mathbf{\Omega}$ lies in (2.4.1.5), repeated below for convenience.

$$\dot{\underline{\underline{F}}} = \underline{\underline{F}}\underline{\underline{\Omega}}. \tag{2.4.1.5}$$

In general, (2.4.1.5) represents a non-linear differential equation. To find an approximate solution for tracking the cellular phone's orientation in real time, however, the assumption will be that $\underline{\underline{\Omega}}$ is constant over any single discrete sample period, that is,

$$\underline{\underline{\Omega}}(t) \cong \underline{\underline{\Omega}}(nT), \; nT \le t \le (n+1)T. \tag{2.8.3}$$

Over the interval described by (2.8.3), (2.4.1.5) is a linear matrix differential equation with straightforward solution

$$\underline{\underline{F}}(t) = \underline{\underline{F}}(t_0) e^{\int_{t_0}^{t} \underline{\underline{\Omega}} d\tau}, \; nT \le t \le (n+1)T, \; t_0 = nT. \tag{2.8.4}$$

Evaluating (2.8.4) at the endpoint, t=(n+1)T yields

$$\underline{\underline{F}}((n+1)T) = \underline{\underline{F}}(nT)e^{\underline{\underline{\Omega}}T} \;. \tag{2.8.5}$$

Using (2.8.5), $\underline{\underline{F}}$ can be determined iteratively, and in real time.

2.9 SOLVING THE TRANSLATION

The translational motion of the cellular phone is simpler than the rotation in that it does not involve a matrix exponential, but it does have one additional step; because the acceleration solved by (2.5.2) is in the units of frame S, a rotational transformation must first be applied to $\mathbf{a}_{A,S}$ so that the acceleration is given in terms of frame O. Fortunately, $\underline{\underline{F}}$ de-rotates $\mathbf{a}_{A,S}$ and was just found in (2.8.3). Now, one can write the translational acceleration in frame O as

$$\mathbf{a}_{A,O} = \underline{\underline{F}}\mathbf{a}_{A,S} \;. \tag{2.9.1}$$

The position of the origin of frame S, $\mathbf{r}_A$ (the center of the cellular phone), is given by double integration of (2.9.1).

$$\mathbf{v}_A = \int \underline{\underline{F}}\mathbf{a}_{A,S}dt + \mathbf{v}_A(t_0) \tag{2.9.2}$$

$$\mathbf{r}_A = \int \mathbf{v}_A dt + \mathbf{r}_A(t_0) \tag{2.9.3}$$

2.10 CONCLUDING REMARK / WHAT'S AHEAD

The position and orientation information contained in $\mathbf{r}_A$ and $\underline{\underline{F}}$ will be used to form parameters that direct the main beam of the antenna array. Before applying $\mathbf{r}_A$ and $\underline{\underline{F}}$ to an antenna array, however, a real-time discrete model for $\mathbf{r}_A$ and $\underline{\underline{F}}$ will be constructed, and then evaluated by itself to see what kind of performance a discrete model can provide.

3 MODEL SIMULATION AND PERFORMANCE

3.1 INTRODUCTION

In this chapter, the analysis of chapter 2 will be used to create a simulation model that tracks the user's position and orientation, with respect to the phone, in real time. Accelerometer measurements are fed to the model, which performs the necessary mathematical computations to reconstruct the phone's motion. In this chapter and in chapter 7, these accelerometer measurements will correspond to known motions, so that the effectiveness and accuracy of the system can be evaluated.

3.2 IMPLEMENTING EQUATION (2.7.3)

Equation (2.7.3) is the starting point of the model, and is repeated below for convenience.

$$\begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \mathbf{a}_{A,S} \end{bmatrix} = \underline{\underline{Q}}\,\underline{a_{M,S}} - \underline{\underline{Q}}\,\underline{\boldsymbol{\Omega}\times\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\cdot\boldsymbol{\theta}_{B,S}} \tag{2.7.3}$$

Recall that $\underline{\underline{Q}}$ is the inverse of matrix $\underline{\underline{J}} = \left[\underline{\left(\mathbf{r}_{B,S}\times\boldsymbol{\theta}_{B,S}\right)} \quad \underline{\boldsymbol{\theta}_{B,S}}\right]$, which is constant and only dependent on the configuration of the accelerometers. $\underline{\underline{Q}}$, therefore, is also a constant matrix and is calculated during pre-processing. The quantity $\underline{\underline{Q}}\,\underline{\boldsymbol{\Omega}\times\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\cdot\boldsymbol{\theta}_{B,S}}$ looks complex, but careful investigation reveals that this is not quite the case. Note that $\underline{\underline{Q}}$, $\mathbf{r}_{B,S}$, and $\boldsymbol{\theta}_{B,S}$ are known and constant, and the dimensions of $\underline{\underline{Q}}\,\underline{\boldsymbol{\Omega}\times\left(\boldsymbol{\Omega}\times\mathbf{r}_{B,S}\right)\cdot\boldsymbol{\theta}_{B,S}}$ are 6x1. Though the cross products make (2.7.3) non-linear, carrying out the computations symbolically and then placing them in a functional block provides a simple means to compute the non-linear portion of the equation. The cost of this approach are errors in

$\mathbf{\Omega}$ due to the fact that the feedback is always one sample behind $\dot{\mathbf{\Omega}}$. A flow diagram for the implementation of (2.7.3) is shown in Fig. 3.2.1. Note that the actual values of $\underline{\underline{Q}}$ and the feedback block implementing $\underline{\underline{Q}\mathbf{\Omega}\times\left(\mathbf{\Omega}\times\mathbf{r}_{B,S}\right)\cdot\mathbf{\theta}_{B,S}}$ are not given. This is because these quantities will change depending on the number and layout of accelerometers within the phone.

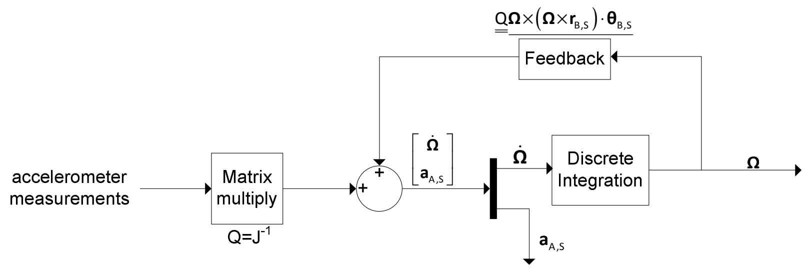


Figure 3.2.1 Flow diagram implementing (2.7.3).

3.3 IMPLEMENTING MATRIX $\underline{\underline{\mathbf{F}}}$

After $\dot{\mathbf{\Omega}}$ is computed, one integration and then a matrix exponential with feedback—actually a second integration—are required to find $\underline{\underline{F}}$ for the cellular phone. The first integration has already been computed as part of the nonlinear feedback. $\mathbf{\Omega}$ is passed through a series of blocks that implements (2.8.5), which is shown again below for convenience. Fig. 3.3.1 shows the additions required to find $\underline{\underline{F}}$.

$$\underline{\underline{F}}((n+1)T) = \underline{\underline{F}}(nT)e^{\underline{\underline{\Omega}}T} . \qquad (2.8.5)$$

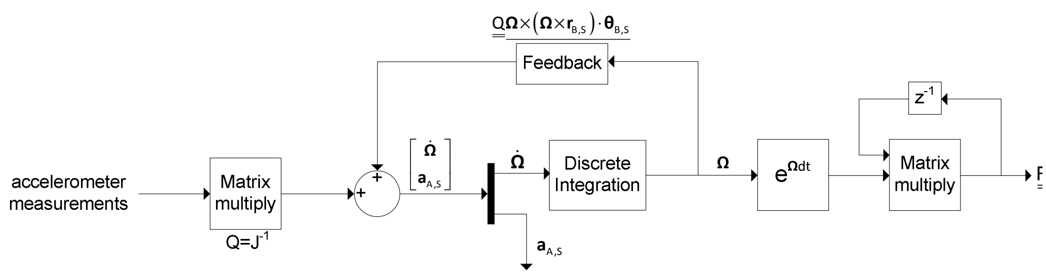


Figure 3.3.1 Flow diagram implementing the INS up to the calculation of $\underline{\underline{F}}$.

3.4 FINDING THE TRANSLATIONAL ACCELERATION, VELOCITY, AND POSITION

To calculate the translational acceleration, velocity, and position, $\mathbf{a}_{A,S}$ must first be transformed to $\mathbf{a}_A$ via matrix $\underline{\underline{F}}$. Once $\mathbf{a}_{A,S}$ is de-rotated by

$$\mathbf{a}_{A,O} = \underline{\underline{F}}\mathbf{a}_{A,S}, \qquad (2.9.1)$$

$\mathbf{a}_{A,O}$ can be integrated twice according to equations (2.9.2) and (2.9.3) to find the position of the device. Additionally, though not included in the mathematical derivation, recall that MEMS accelerometers measure gravity continuously. To prevent the force of gravity from being integrated by the system, an acceleration opposite gravity is inserted at $\mathbf{a}_{A,O}$. With these steps complete, the discrete time INS that calculates both orientation and position is shown in Fig. 3.4.1.

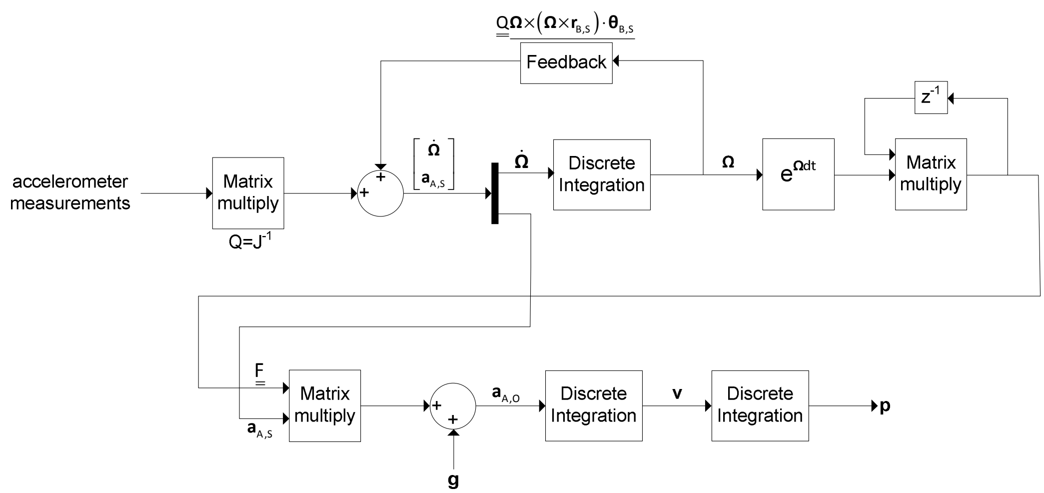


Figure 3.4.1 Complete inertial tracking system.

3.5 MODEL VERIFICATION / PERFORMANCE

With a model in hand, the next logical step is to test the performance of this system. Specifically, one cannot expect such a model to function indefinitely because discrete integrators introduce accumulating error to the system. The nonlinear feedback also introduces error because this block always operates with a one sample delay. In the remainder of this chapter, the INS will be used to simulate a few test motions to confirm functionality.

3.6 FRAME ORIENTATION AND ACCELEROMETER ARRANGEMENT

Before actually running simulations to see if the INS model works, it will be useful to define some quantities and references for the phone, user and INS. One issue of particular importance is the arrangement of the accelerometers in the phone. For the system to be functional, matrix $\underline{\underline{J}}$ must have rank six, so that $\underline{\underline{Q}}$ exists. The merits of different numbers and configurations of accelerometers will be assessed later. For now, an intuitive and symmetric arrangement of four three-axis accelerometers at each of the four corners of the phone will be used to check that the tracking algorithm works. Fig. 3.6.1(a) shows this configuration, where the numbers 1, 2, 3, and 4 represent three-axis accelerometers one through four.

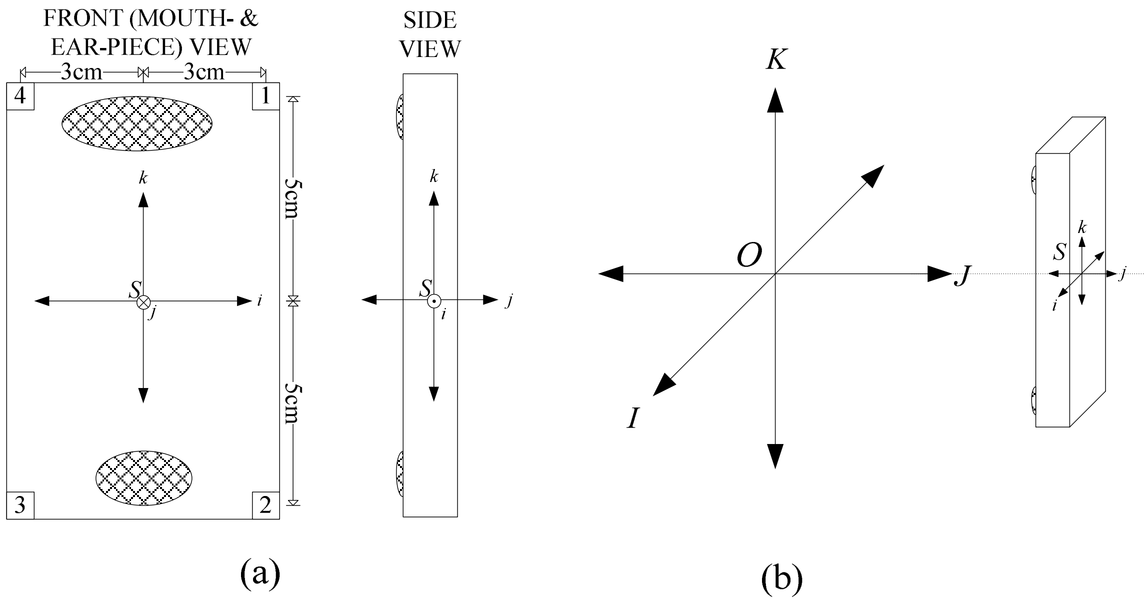


Figure 3.6.1 (a) Orientation of S within the phone. (b) Orientation of the phone and S with respect to the inertial frame O.

In addition to the accelerometer arrangement, Fig. 3.6.1(a) and (b) show how frame O and S are oriented with respect to each other. Fig. 3.6.1(a) shows the orientation of S in the phone, which is considered to be in an un-rotated state when the basis vectors **i**, **j**, **k**, are parallel to the inertial frame's basis vectors **I**, **J**, **K**, respectively. Fig. 3.6.1(b) shows one example of the phone in an un-rotated state. Furthermore, the origin of O will always be assumed to be located at the center of the user's head. With the user's head located at the center of O, Fig. 3.6.1(b) could correspond to a left-handed user looking in the +**I** direction holding the phone upright to their left ear. On the other hand—no pun intended—a user is looking in the –**I** direction would be holding the phone with their right hand.

Besides considering the orientations of O and S with respect to each other, each of the four accelerometers has three axes that must be specified. The orientation of the accelerometers within S is specified by the vector $\boldsymbol{\theta}_{B,A}$ from chapter 3. Since a three-axis accelerometer forms an orthonormal basis for measuring acceleration, any orientation of accelerometers provides a complete set of information. Strange orientations, however, confer no advantage and complicate computation of the dot product in (2.6.1). For this reason, accelerometers will be oriented with axes parallel to the axes of S, so that the x-axis of all accelerometers is parallel to **i**, the y-axis of all accelerometers is parallel to **j**, and the z-axis of all accelerometers is parallel to **k**.

3.7 TESTING THE TRACKING ALGORITHM

3.7.1 TEST MOTION 1—A SIMPLE ORBIT

With the orientation and placement of the accelerometers and frames specified, a test of the tracking algorithm can be performed. In the first test, the phone orbits the origin of O in the xy-plane at a distance of 0.1 meters. During the same time, the phone rotates about the **k** axis such that the same face of the phone is always presented to the origin of O. This is the same motion that the moon has as it rotates about the earth. The phone completes one orbit every four seconds and the test is run for 10 seconds. Fig. 3.7.1.1 shows a diagram of this test's motion.

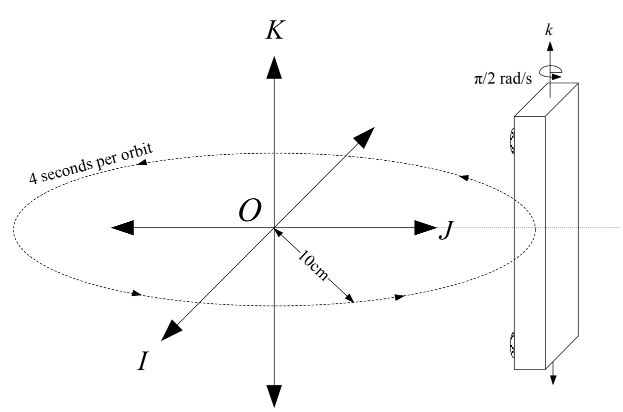


Figure 3.7.1.1 Simple orbit motion

The results of this simulation are shown in Fig. 3.7.1.2. As time increases, the envelope of position errors is increasing. Errors are accumulating in position because of the four discrete approximations of the integral and the matrix exponential. The INS model uses the trapezoidal method for its discrete integration. As time increases, the position error will continue to grow. In this simulation, however, the total error is less than one fifth of a millimeter after ten seconds; such small errors are well within the operating limits of the INS, which will work reliably up to 5cm of error. In the following section, the tracking algorithm will be tested under a more complicated condition, in which nonlinear feedback is present.

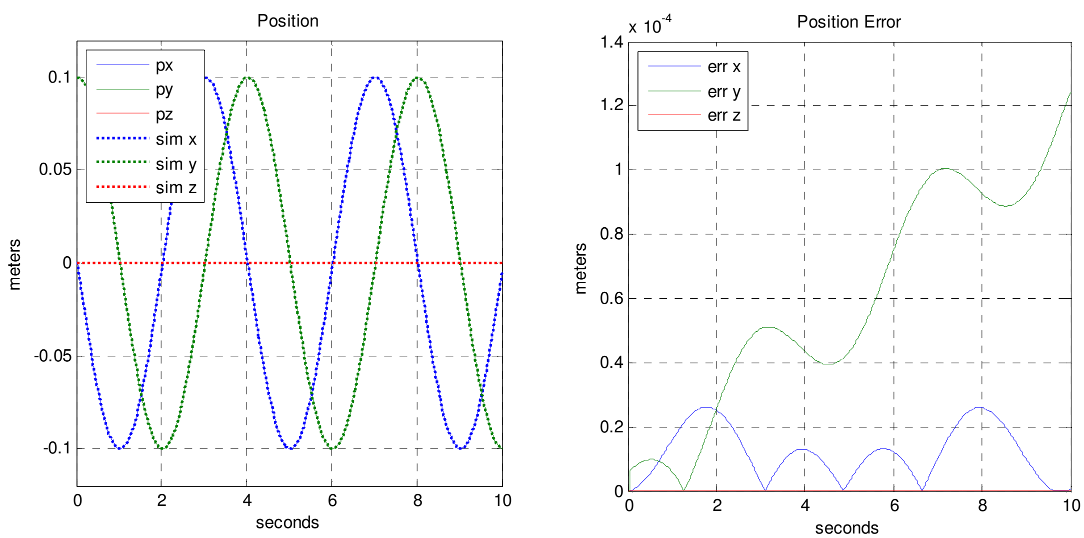


Figure 3.7.1.2 Result of simple orbit motion test motion.

3.7.2 TEST MOTION 2—A DOUBLY ROTATING ORBIT

A more complicated motion under which to test can be described as a doubly rotating orbit. This motion is similar the simple orbit of the last test, but now includes rotation about the **i** axis in addition to the **k** axis. Fig. 3.7.2.1 provides an illustration of this motion. While a human could never replicate this motion, inputs of this type will demonstrate whether or not the INS works.

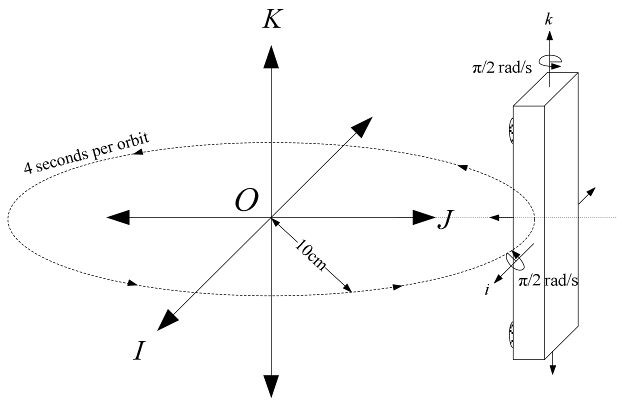


Figure 3.7.2.1 Orbiting cellular phone that rotates about two of its axes.

Fig. 3.7.2.2, below, shows the error plots for and orbiting cellular phone that rotates about two of its axes. Errors grow so quickly for this motion that the system is effectively useless. The cause of the system failure for this more complicated motion lies in the nonlinear feedback portion of the model (section 3.2). For the standard configuration described in section 3.6, the nonlinear differential equation (2.7.3) is

$$\begin{bmatrix} \dot{\Omega}_x \\ \dot{\Omega}_y \\ \dot{\Omega}_z \\ a_{A,S,x} \\ a_{A,S,y} \\ a_{A,S,z} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} + \begin{bmatrix} -\Omega_y\Omega_z \\ \frac{8}{17}\Omega_x\Omega_z \\ \Omega_x\Omega_y \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.7.2.1}$$

where $\underline{\underline{Q}}\underline{a_{M,S}} = \mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix}^T$.

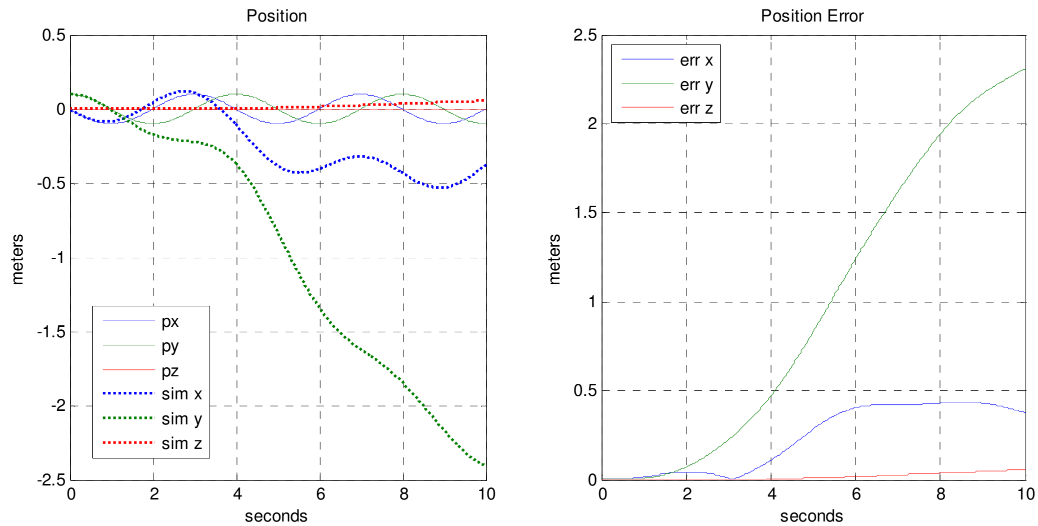


Figure 3.7.2.2 Result of doubly rotating orbit test motion.

The system in (3.7.2.1) will be nonlinear only if there are rotations about multiple axes, which is precisely the case for the doubly rotating orbit. This difference, coupled with the excellent performance of the simple orbit of section 3.7.1, which has no nonlinear elements, makes the nonlinear feedback an obvious candidate for the source of the system failure. To demonstrate that this is the case, the errors present in $\mathbf{\Omega}$ are given below in Fig. 3.7.2.3.

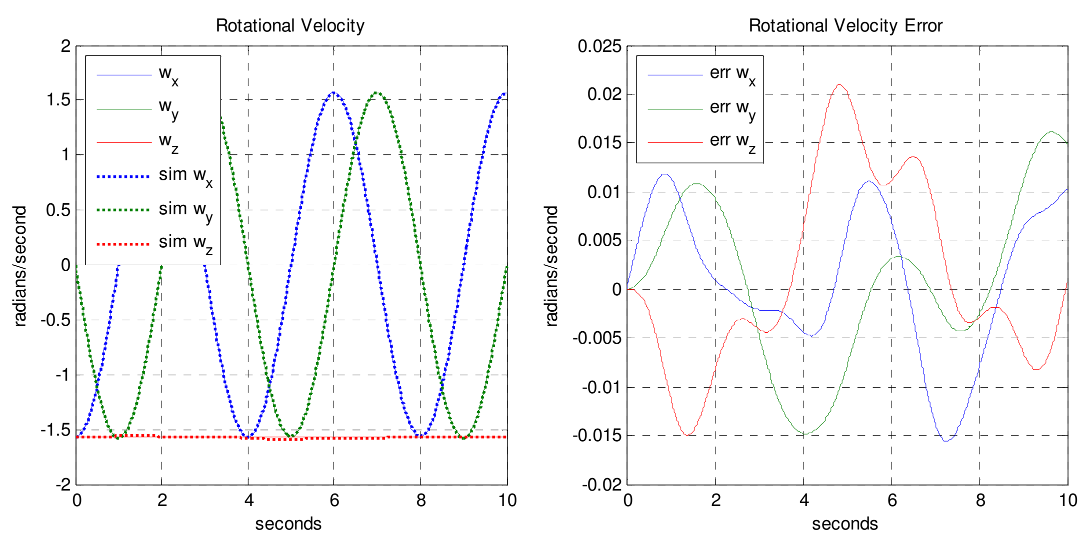


Figure 3.7.2.3 Measured and calculated rotation velocities. Rotational velocity error is significant.

3.8 CORRECTING THE NONLINEAR HALF OF THE MODEL

The errors seen in Fig. 3.7.2.3 may not seem very large, and if the system's terminus were at the rotational velocity, they may not be a problem. In the INS, however, these errors accumulate via integration, and result in a misaligned $\underline{\underline{F}}$. The erred rotation matrix is then applied to translational velocities, including gravity, and the system perceives itself as undergoing constant acceleration (i.e. that it is falling). The constant force of gravity requires that errors in the rotation matrix, and therefore in the rotational velocity, remain very small. Additionally, the matrix exponential is nonlinear under the same conditions as (2.7.3). For this reason, generation of $\underline{\underline{F}}$ also needs to be corrected.

Human motion is actually smooth and continuous, and the INS is a discrete model that uses samples of this continuous motion. To correct the catastrophic error caused by the rough approximation of the INS's nonlinear nature, one can upsample the inputs using a polynomial fit, and then use the same algorithm on the new signal. Incorporating upsampling results in the model shown in Fig. 3.8.2. The main difference is the presence of the *up* and *down* blocks. The *up* block uses three samples of from its input to find the coefficients of a second order polynomial, and then outputs a finely sampled version of this polynomial. At the *down* block, system speed is reduced to its original sample rate.

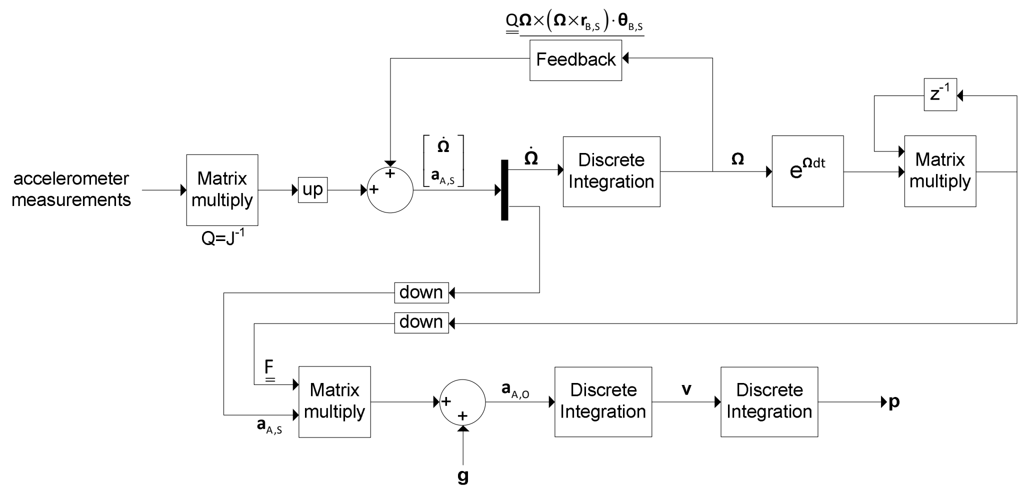


Figure 3.8.2 INS model that uses an upsampled polynomial fit to calculate the nonlinear portion of the system.

Using this new model, the doubly rotating orbit motion of section 3.7.2 is re-simulated. As seen in Fig. 3.8.3, the upsampling-polynomial fit method described in this section is quite effective and gives the INS similar performance to the linear accelerations of the simple orbit of section 3.7.1.

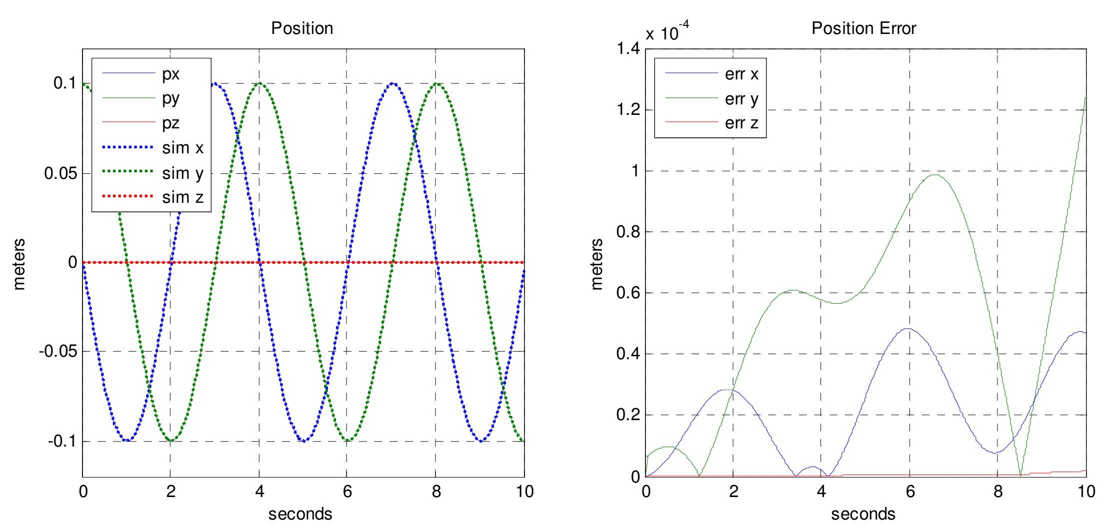


Figure 3.8.3 Position and error outputs for a doubly rotating orbit using the improved INS model of Fig. 3.8.2.

3.8 CONCLUDING REMARK / WHAT'S AHEAD

The motion tests of 3.7.1 and 3.8 demonstrate that the INS algorithm works. Up to this point, however, the presence of noise in the system has been ignored. In chapter 4, a quantitative assessment of the effect of noise on the performance of the INS will be attempted.

4 NOISE

4.1 INTRODUCTION

Though noise is not included in the simulations performed for this report, Gaussian noise is present in all MEMS devices. Noise is both damaging and difficult to quantify in inertial systems. It is damaging because noise can accumulate quickly through the repeated integrations required to calculate position. The effect of noise is also difficult to quantify because inertial systems are nonlinear, so that noise characteristics change significantly depending on the current motion. Never-the-less, the noise performance for a zero input signal will be presented to demonstrate the cumulative effect of noise on the INS.

4.2 NOISE IN Ω

To analyze noise, one must return all the way to (2.6.6), which governs the accelerometer measurements. Noise in MEMS accelerometers is additive white Gaussian (AWGN), and affects each accelerometer and each axis independently, therefore, a noise vector $\underline{\mathrm{n}}$ can simply be added to (2.6.6).

$$\underline{\mathrm{a}_{M,S}} = \begin{bmatrix} \underline{\left(\mathbf{r}_{B,S} \times \boldsymbol{\theta}_{B,S}\right)} & \underline{\boldsymbol{\theta}_{B,S}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \mathbf{a}_{A,S} \end{bmatrix} + \underline{\boldsymbol{\Omega} \times \left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S}\right) \cdot \boldsymbol{\theta}_{B,S}} + \underline{\mathrm{n}} \tag{4.2.1}$$

Applying $\underline{\underline{\mathrm{Q}}} = \begin{bmatrix} \underline{\left(\mathbf{r}_{B,S} \times \boldsymbol{\theta}_{B,S}\right)} & \underline{\boldsymbol{\theta}_{B,S}} \end{bmatrix}^{-1}$ to (4.2.1) and solving for $\begin{bmatrix} \dot{\boldsymbol{\Omega}} & \mathbf{a}_{A,S} \end{bmatrix}^{T}$ yields

$$\begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \mathbf{a}_{A,S} \end{bmatrix} = \underline{\underline{\mathrm{Q}}} \left(\underline{\mathrm{a}_{M,S}} - \underline{\boldsymbol{\Omega} \times \left(\boldsymbol{\Omega} \times \mathbf{r}_{B,S}\right) \cdot \boldsymbol{\theta}_{B,S}} + \underline{\mathrm{n}} \right). \tag{4.2.2}$$

To simplify writing of the equation, let $f(\mathbf{\Omega}) = \underline{\mathbf{\Omega}\times\left(\mathbf{\Omega}\times\mathbf{r}_{B,S}\right)\cdot\boldsymbol{\theta}_{B,S}}$.

$$\begin{bmatrix} \dot{\mathbf{\Omega}} \\ \mathbf{a}_{A,S} \end{bmatrix} = \underline{\underline{Q}}\left(\underline{a_{M,S}} - f(\mathbf{\Omega}) + \underline{n}\right) \tag{4.2.3}$$

The noise in $\dot{\mathbf{\Omega}}$ will lead to noise in $f(\mathbf{\Omega})$, but the noise in $f(\mathbf{\Omega})$ can be neglected because the feedbacks always involve products of the component axes, and these products will be much smaller than the direct noise. The noise in $\begin{bmatrix} \dot{\mathbf{\Omega}} & \mathbf{a}_{A,S} \end{bmatrix}^T$ can therefore be approximated by

$$\begin{bmatrix} \underline{n}_{\dot{\mathbf{\Omega}}} & \underline{n}_{\mathbf{a}_{A,S}} \end{bmatrix}^T = \underline{\underline{Q}}\underline{n}\,. \tag{4.2.4}$$

Since every element in the noise vector has the same distribution, (4.2.4) can also be written as a product with a vector of standard normals.

$$\begin{bmatrix} \underline{n}_{\dot{\mathbf{\Omega}}} & \underline{n}_{\mathbf{a}_{A,S}} \end{bmatrix}^T = \sigma\underline{\underline{Q}}\underline{z} \tag{4.2.5}$$

Equation (4.2.5) represents the noise present in $\begin{bmatrix} \dot{\mathbf{\Omega}} & \mathbf{a}_{A,S} \end{bmatrix}^T$. The noise in $\mathbf{\Omega}$, then, is simply

$$\underline{n}_{\mathbf{\Omega}} = T\sum\sigma\underline{\underline{Q}}\underline{z}\,. \tag{4.2.6}$$

4.3 NOISE IN THE ROTATION MATRIX

From chapter 2, the skew-symmetric version of $\mathbf{\Omega}$—which is $\underline{n}_{\mathbf{\Omega}}$ here—is applied to a matrix exponential to calculate matrix $\underline{\underline{F}}$. Normally, the rotation matrix is calculated by the recursive relation of (2.8.5), so $\underline{\underline{F}}$ would be

$$\underline{\underline{F}}((n+1)T) = \underline{\underline{F}}(nT)e^{\underline{\underline{n}}_\Omega T} , \tag{4.3.1}$$

where $\underline{\underline{n}}_\Omega$ is the skew symmetric version of $\underline{n}_\Omega$. If $\underline{n}_\Omega$ remains small, (2.8.5) can be approximated as

$$\underline{\underline{F}}(nT) = \underline{\underline{F}}_0 e^{T\sum \underline{\underline{n}}_\Omega} , \tag{4.3.2}$$

This approximation's accuracy deteriorates as the magnitude of $T\sum \underline{\underline{n}}_\Omega$ increases, but it is nevertheless appropriate for the present noise analysis in which there is no motion in this INS. Even though $T\sum \underline{\underline{n}}_\Omega$ will become large eventually, the system will have become useless long before the approximation becomes an issue because gravity will have erroneously tracked the system far beyond its operating limits by that time. Using the sine small angle approximation to write $\underline{\underline{F}}$ yields

$$\underline{\underline{F}} = \begin{bmatrix} 1 & -n_{F_z} & n_{F_y} \\ n_{F_z} & 1 & -n_{F_x} \\ -n_{F_y} & n_{F_x} & 1 \end{bmatrix}, \tag{4.3.3}$$

where

$$\begin{bmatrix} n_{F_x} & n_{F_y} & n_{F_z} \end{bmatrix}^T = T\sum \underline{n}_\Omega . \tag{4.3.4}$$

4.4 NOISE IN THE TRANSLATIONAL ACCELERATION

For the zero input motion being considered presently, the measured translational acceleration is the constant force of gravity, so $\mathbf{a}_{A,S}$ is

$$\mathbf{a}_{A,S} = \begin{bmatrix} n_{a_x} & n_{a_y} & g + n_{a_z} \end{bmatrix}^T . \tag{4.4.1}$$

To find $\mathbf{a}_{A,O}$, apply (4.3.1) to (4.4.1) and subtract gravity.

$$\mathbf{a}_{A,O} = \underline{\underline{F}}\mathbf{a}_{A,S} = \begin{bmatrix} n_{a_x} - n_{F_z} n_{a_y} + n_{F_y}(g + n_{a_z}) \\ n_{F_z} n_{a_x} + n_{a_y} - n_{F_x}(g + n_{a_z}) \\ n_{F_x} n_{a_y} - n_{F_y} n_{a_x} + n_{a_z} \end{bmatrix} \tag{4.4.2}$$

Equation (4.4.2) may look daunting, but in fact, most of the noise terms may be ignored. As will be discussed in chapter 5, the limited size of cellular phones creates a situation where the translational noises will always be much smaller than the rotational acceleration noises; moreover, the force of gravity is much greater than the noise present on the z-axis, so the noise in the z-axis may be ignored when it is involved in a product, as in the x- and y- axes. Ignoring the insignificant elements of (4.4.2) allows one to write a much simpler expression that is a good approximation of $\mathbf{a}_{A,O}$.

$$\mathbf{a}_{A,O} \cong \begin{bmatrix} n_{F_y} g \\ n_{F_x} g \\ n_{F_x} n_{a_y} + n_{F_y} n_{a_x} + n_{a_z} \end{bmatrix} \tag{4.4.3}$$

Equation (4.4.3) shows that noise in the x- and y-axes will approximately normal with variances increased from the previous step by g^2.

The z-axis, on the other hand, presents more of a problem. Recall from (4.2.5) that $\underline{n}_{\dot{\Omega}}$ and $\underline{n}_{\mathbf{a}_{A,S}}$ were formed from linear combinations of $\sigma\underline{\underline{Q}}\underline{z}$, which means that $\underline{n}_{\dot{\Omega}}$ and $\underline{n}_{\mathbf{a}_{A,S}}$ are not independent. The quantity on the z-axis, therefore, will be a sum of three types of random variables: normal, chi^2, and product of normal. Fortunately, because g is so much larger than the variances of n_{a_x} and n_{a_y}, noise in the x- and y- axes will be the performance limiter in this case.

4.5 NOISE IN THE POSITION

To find the noise in the position, $\mathbf{a}_{A,O}$ must pass through two more integrators. The effect of the two integrators results in a double sum of the noise multiplied by the square of the sample time. The noise in the position is

$$p = T^2 \sum \sum \mathbf{a}_{A,O} \, . \tag{4.5.1}$$

Substitution of (4.2.6) and (4.3.4) into (4.5.1) yields

$$p = T^2 \sum \sum \begin{bmatrix} \left(T^2 \sum \sum \sigma \underline{\underline{Q}} \underline{z} \right)_{row2} g \\ \left(T^2 \sum \sum \sigma \underline{\underline{Q}} \underline{z} \right)_{row1} g \\ - \end{bmatrix}, \tag{4.5.2}$$

where the row subscripts refer to the corresponding row of the matrix or vector quantity in parentheses and the dash in the third row is to remind the reader that an estimate of noise on the z-axis is not being made because it is much smaller than the noises in the x- and y-axes and is non-normal. Simplifying (4.5.2) gives

$$p = \begin{bmatrix} T^4 \sigma g \sum \sum \sum \sum \underline{\underline{Q}}_{row2} \underline{z} \\ T^4 \sigma g \sum \sum \sum \sum \underline{\underline{Q}}_{row1} \underline{z} \\ - \end{bmatrix}. \tag{4.5.3}$$

The rows of $\underline{\underline{Q}}$ create linear combinations of the standard normals in $\underline{z}$. Because the elements of $\underline{z}$ are independent, the resulting linear combinations are also zero mean normal random variables, with variances given by the sum of the squares of the corresponding row of $\underline{\underline{Q}}$. Using this fact, (4.5.3) can be further simplified to

$$p = \begin{bmatrix} T^4 \sigma g \sqrt{q_2} \sum\sum\sum\sum \underline{z} \\ T^4 \sigma g \sqrt{q_1} \sum\sum\sum\sum \underline{z} \\ - \end{bmatrix}, \tag{4.5.4}$$

where $q_i = \underline{\underline{Q}}_{rowi} \underline{\underline{Q}}_{rowi}^T$. Keeping only the highest ordered term of the quadruple sums, whose computations are detailed in the appendix, results in the approximation for p seen below in (4.5.5).

$$p = \begin{bmatrix} \frac{\sigma g}{6} \sqrt{\frac{q_2 T t^7}{7}} \underline{z} \\ \frac{\sigma g}{6} \sqrt{\frac{q_1 T t^7}{7}} \underline{z} \\ - \end{bmatrix} \tag{4.5.5}$$

To state (4.5.5) another way: for the INS operating with no accelerations, the outputs will be approximately normal, with variances given by:

$$var_x = \frac{\sigma^2 g^2 q_2 T t^7}{252} \quad \text{and} \quad var_y = \frac{\sigma^2 g^2 q_1 T t^7}{252}. \tag{4.5.6}$$

4.6 ANALYSIS CONFIRMATION

At last, (4.5.5) provides an expression for the error variance present in the position calculation. To check this approximation of noise variance growth, a noisy simulation with no motion is simulated many times with confirm that the calculated has the variances given by (4.5.5). The parameters for this test are: T=0.01 seconds, t=0 to 5 seconds, σ=0.019 m/s^2, $\underline{\underline{Q}}$ is the matrix for the standard configuration of Fig. 3.6.1,

$g=9.8\ m/s^2$, and number of simulations is 500. Both estimates are close to the calculated values and, as expected, noise in the z-axis is much smaller than both the x- and y- axes.

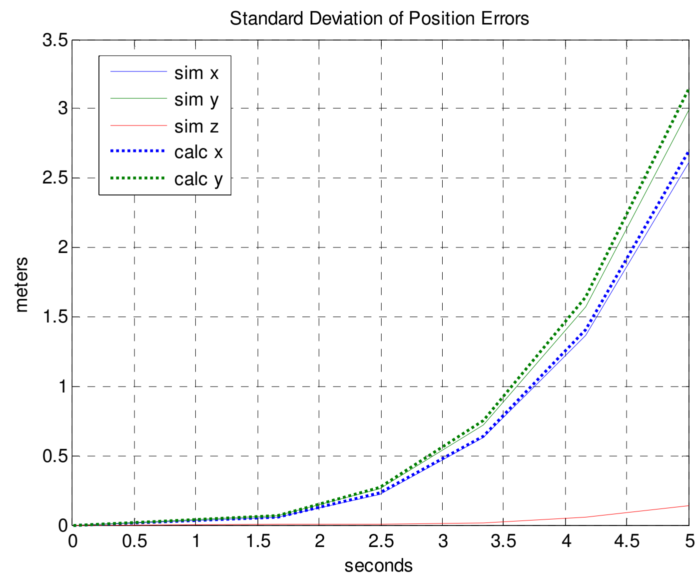


Figure 4.6.1 Comparisons of calculated and simulated standard deviations for the INS undergoing no motion.

4.7 CONCLUDING REMARK / WHAT'S AHEAD

Though the results of the error growth analysis are very encouraging in terms of agreement between the calculated and observed results, it is clear that error may grow very rapidly. Equation (4.5.6) shows which parameters affect noise accumulation. The MEMS noise variance, the sample rate, the INS's accelerometer configuration, and the total system run time all play a major role in noise accumulation. Additionally, for a system in motion, noise variance will be affected both by g and the other translational accelerations experienced by the INS. The variance estimates will change for other types of motion, sample rates, and accelerometer configurations, but noise variance growth will always be on the order of at least t^7, so the INS will have a limited run time no matter what the inputs or precision of the input devices. In chapter 5 a few performance improving optimizations will be considered.

5 PERFORMANCE OPTIMIZATIONS

5.1 INTRODUCTION

Recall that the 6xN matrix $\underline{\underline{Q}}$ is the inverse or pseudoinverse of matrix $\underline{\underline{J}} = \left[\underline{\left(\mathbf{r}_{B,S} \times \boldsymbol{\theta}_{B,S} \right)} \quad \underline{\boldsymbol{\theta}_{B,S}} \right]$. $\underline{\underline{J}}$ is a function of the locations and orientations of the accelerometers. As $\underline{\underline{J}}$'s inverse, $\underline{\underline{Q}}$ is also a function of the accelerometer configuration only. In the previous chapter it was seen that the noise variance of the orientation is directly related to $\underline{\underline{Q}}$. The natural question arises—what configurations of $\underline{\underline{Q}}$ minimize the noise in the orientation? The variance imparted to the system by $\underline{\underline{Q}}$ for a particular axis is the sum of the squares of the elements in the corresponding row of $\underline{\underline{Q}}$. Therefore, to minimize variance added by $\underline{\underline{Q}}$, one should seek to minimize the elements of $\underline{\underline{Q}}$.

5.2 MAXIMIZE THE DISTANCE BETWEEN ACCELEROMETERS—THE MOST IMPORTANT THING

Though for any given phone shape there are an infinite number of possible accelerometer arrangements; a very limited set—often just one—of these arrangements is optimal. The optimal arrangement for the phone shapes considered in this report are those arrangements where the average accelerometer to accelerometer distance is greatest. Though this simple rule is given without proof, a two dimensional example provides some intuition as to why maximizing the distance between accelerometers is effective.

Consider a particle traveling on a circular path about the origin as in Fig. 5.2.1.

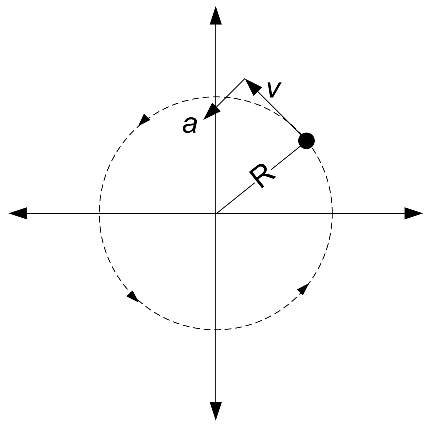


Figure 5.2.1 Simple circular motion.

The acceleration of the particle is well known [10] and is

$$a = \omega^2 R\,. \tag{5.2.1}$$

Suppose, as in the present case, that this acceleration also has a component of noise so that

$$a = a_T + n\,, \tag{5.2.2}$$

where a_T is the true value of the acceleration, and n is the noise. Substituting (5.2.1) into (5.2.2) for a_T yields

$$a = \omega^2 R + n\,. \tag{5.2.3}$$

Equation (5.2.3) demonstrates that increases in R will result in proportional improvements in the signal to noise ratio of the acceleration. For the same reason that larger separation in this simple example reduces the effect of noise, one should seek large separations between accelerometers in the INS to similarly reduce noise. This design rule's effectiveness is easily demonstrated. Consider the standard example configuration of Fig. 3.6.1. By generating a range of $\underline{\underline{Q}}$ matrices from scaled versions of Fig. 3.6.1, and then calculating the variance of $\underline{\underline{Q}}\,\underline{z}$ for each of these configurations, the dependence of the noise variance on the accelerometer to accelerometer distance can be calculated. The result is shown below in Fig. 5.2.1. Clearly, the variance of $\underline{\underline{Q}}\,\underline{z}$ decreases as the distance between accelerometers gets larger.

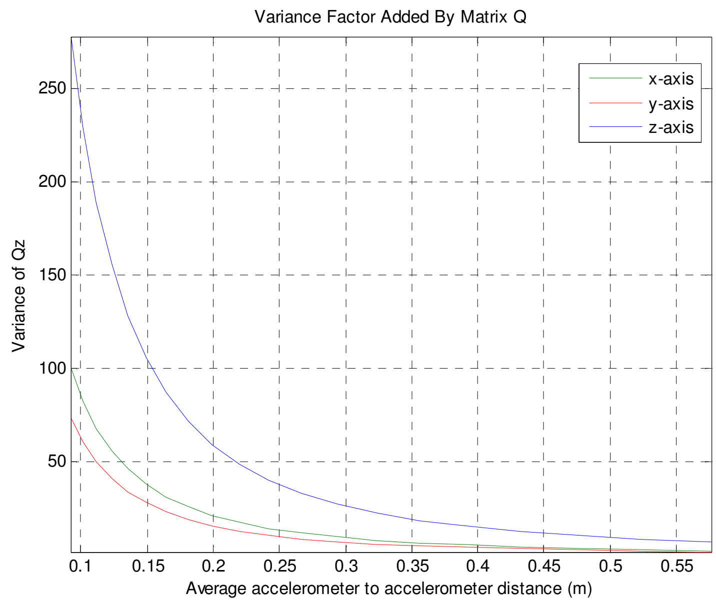


Figure 5.2.2 Variance of $\underline{\underline{Q}}\,\underline{z}$ as a function of the average accelerometer to accelerometer distance.

5.3 USE A MORE COMPLEX PHONE GEOMETRY?

While the shape of the phone may not really be a design *choice* to an engineer developing an INS, it may be interesting to consider $\underline{\underline{Q}}$ for at least one other type of cellular phone—the flip phone. Typical dimensions and frame arrangement for a flip phone are shown in Fig. 5.3.1.

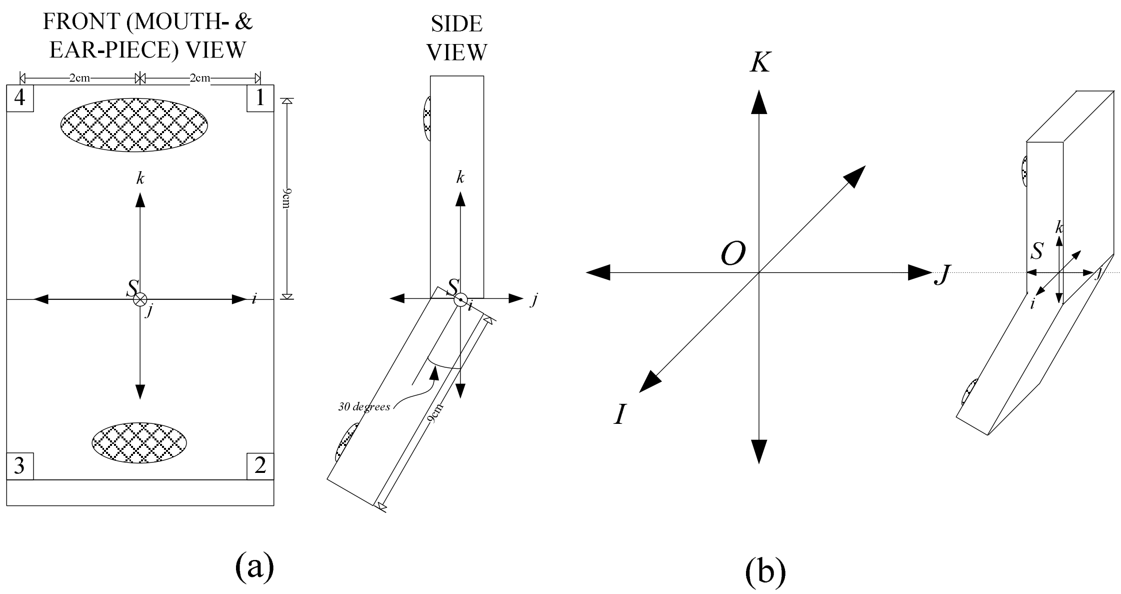


Figure 5.3.1 Accelerometer configuration and frame definitions for a typical flip phone.

Unlike the smart phone configuration of Fig. 3.6.1, accelerometers two and three can no longer have measurement axes that are parallel to all three of the bases of S and O. The basis vectors of the bottom two accelerometers are a rotated version of the original basis vectors. The plane of the bottom half of the flip phone is rotated by approximately -30 degrees about the **I** (or **i**) axis. A rotator matrix can be used to transform the axes of accelerometers 2 and 3 in Fig. 3.6.1, to the axes of accelerometers 2 and 3 in Fig. 5.3.1.

The rotator matrix is

$$A_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}. \tag{5.3.1}$$

Multiplying A_x with the original measurement axes $\theta_x = [1 \quad 0 \quad 0]$, $\theta_y = [0 \quad 1 \quad 0]$, and $\theta_z = [0 \quad 0 \quad 1]$, provides the new axes of measurement for accelerometers two and three.

$$\theta_{x,2-3} = [1 \quad 0 \quad 0] \tag{5.3.2a}$$

$$\theta_{y,2-3} = \begin{bmatrix} 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \tag{5.3.2b}$$

$$\theta_{z,2-3} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \tag{5.3.2c}$$

The locations of accelerometers 2 and 3 are also a little different than before.

$$\mathbf{r}_{A,2} = [0.02 \quad -0.09\sin(30°) \quad -0.09\cos(30°)] \tag{5.3.3a}$$

$$\mathbf{r}_{A,3} = [-0.02 \quad -0.09\sin(30°) \quad -0.09\cos(30°)] \tag{5.3.3b}$$

The variance of the elements in $\underline{\underline{Q}}\underline{z}$ for this geometry are

$$\text{var}\left(\underline{\underline{Q}}\underline{z}\right) = [33.08 \quad 46.19 \quad 261.15 \quad 0.40 \quad 0.25 \quad 0.27]^T. \tag{5.3.4}$$

For comparison, the variance of the elements in $\underline{\underline{Q}}\underline{z}$ for the standard geometry of Fig. 3.6.1 are

$$\mathrm{var}\left(\underline{\underline{Q}}\,\underline{z}\right)=\begin{bmatrix}100 & 73.53 & 277.78 & 0.25 & 0.25 & 0.25\end{bmatrix}^{T}. \tag{5.3.5}$$

The improved variances for the first and second term are due to the fact that a typical flip phone is longer in the z-dimension than the smart phone, this provides improved noise performance in the x and y dimensions. The flip phone shape provides both a cost and a benefit to noise on the z-axis. The phone width has been reduced from 6cm to 4cm, which reduces resolution (per the discussion of section 5.2) for rotations about z and y, but the angle of the bottom half of the phone (which adds an offset in the y direction to accelerometers 2 and 3) adds resolution to the z and x direction. The result is a slight net improvement in detection of z rotations. On the other hand, there are slight decreases in the performance of all three translational axes. It is disappointing that the improvements in noise performance come mostly to the x and y rotational axes, as these axes are less important to the overall function of the beamforming array than the z axis.

5.4 USE MORE ACCELEROMETERS?

Noise performance can also be improved by using more accelerometers. Initially, multiple accelerometers were required to ensure that $\underline{\underline{J}}$ had rank six. Any additional accelerometers work to reduce the noise by averaging the independent noise from different accelerometers. As an example, again consider the standard configuration of Fig. 3.6.1, but now suppose that in each of the four corners there are two 3-axis accelerometers at the same point. Compare the $\underline{\underline{Q}}$ matrix of Fig. 3.6.1 with the doubled configuration:

$\underline{\underline{Q}}$ from four 3-axis accelerometers:

$$\begin{bmatrix} 0 & -5 & 0 & 0 & -5 & 0 & 0 & 5 & 0 & 0 & 5 & 0 \\ \frac{125}{34} & 0 & -\frac{75}{34} & \frac{125}{34} & 0 & \frac{75}{34} & -\frac{125}{34} & 0 & \frac{75}{34} & -\frac{125}{34} & 0 & -\frac{75}{34} \\ 0 & \frac{25}{3} & 0 & 0 & -\frac{25}{3} & 0 & 0 & -\frac{25}{3} & 0 & 0 & \frac{25}{3} & 0 \\ \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} \end{bmatrix} \tag{5.4.1}$$

$\underline{\underline{Q}}$ from eight 3-axis accelerometers:

$$\begin{bmatrix} 0 & -\frac{5}{2} & 0 & 0 & -\frac{5}{2} & 0 & 0 & \frac{5}{2} & 0 & 0 & \frac{5}{2} & 0 & 0 & -\frac{5}{2} & 0 & 0 & -\frac{5}{2} & 0 & 0 & \frac{5}{2} & 0 & 0 & \frac{5}{2} & 0 \\ \frac{125}{68} & 0 & -\frac{75}{68} & \frac{125}{68} & 0 & \frac{75}{68} & -\frac{125}{68} & 0 & \frac{75}{68} & -\frac{125}{68} & 0 & -\frac{75}{68} & \frac{125}{68} & 0 & -\frac{75}{68} & \frac{125}{68} & 0 & \frac{75}{68} & -\frac{125}{68} & 0 & \frac{75}{68} & -\frac{125}{68} & 0 & -\frac{75}{68} \\ -\frac{25}{6} & 0 & 0 & \frac{25}{6} & 0 & 0 & \frac{25}{6} & 0 & 0 & -\frac{25}{6} & 0 & 0 & -\frac{25}{6} & 0 & 0 & \frac{25}{6} & 0 & 0 & \frac{25}{6} & 0 & 0 & -\frac{25}{6} & 0 & 0 \\ \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 \\ 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 \\ 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} & 0 & 0 & \frac{1}{8} \end{bmatrix} \tag{5.4.2}$$

With a complete second set of measurements, one would expect the variance of $\underline{\underline{Q}}\underline{z}$ to be halved, and this is precisely the case. The sum of the squares of the rows in (5.4.2) is half the sum of the squares of the rows in (5.4.1).

The use of additional accelerometers is effective for mitigating noise. The cost, however, in dollars, power consumption, board space, and computational loading on the algorithm, prevents the use of additional accelerometers from being a viable method of noise mitigation. Moreover, examples like this one, in which each accelerometer is duplicated, add no insight or intuition to the understanding of noise in the system and is simply a costly brute force method of improving performance. For these reasons, using more than four 3-axis accelerometers will not be considered. One may, however, use fewer than four.

5.5 USE FEWER ACCELEROMETERS?

Another question to consider, which will invariably increase the noise, is whether fewer than four accelerometers should be used. Originally, four were selected because it is intuitive to place one accelerometer at each corner of the smart phone, as this is an efficient way to maximize the number of accelerometers without sacrificing accelerometer to accelerometer distances. However, the use of three three-axis accelerometers (two three-axis accelerometers cannot provide a sufficiently ranked $\underline{\underline{J}}$) also provides a rank 6 $\underline{\underline{J}}$ matrix, provided that the three accelerometers are not collinear. Holding to the original design rule that accelerometers ought to be as far apart on the phone as possible, one will find that the optimal three three-axis accelerometer configuration is the same as the four three-axis accelerometer configuration with one accelerometer deleted. Arbitrarily, choose accelerometer 4 of Fig. 3.6.1 to be the deleted accelerometer. The variance of $\underline{\underline{Q}}\underline{z}$ with one accelerometer deleted is

$$\mathrm{var}\left(\underline{\underline{Q}}\underline{z}\right)=\begin{bmatrix}200.00 & 110.29 & 555.56 & 0.37 & 0.50 & 0.34\end{bmatrix}^T . \qquad (5.5.1)$$

As expected, the variances are all higher than in the four accelerometer case (compare to (5.3.5)). In the remainder of this report, only the standard four accelerometer configuration will be considered, but a three accelerometer configuration would certainly be something to consider in a design being optimized for mass production.

5.6 CONCLUDING REMARK / WHAT'S AHEAD

The most important consideration when arranging accelerometers on the phone is distance. Accelerometers ought to be placed as far apart as possible. Noise performance of a flip phone is comparable to that of a smart phone, and would therefore also represent a viable platform for an INS. Increasing the number of accelerometers is not really a practical optimization due to its cost. This type of brute force optimization may be required in some (non-miniature) applications, but for the present research, duplicating accelerometers adds no understanding or ingenuity to the problem. In chapter 6, beamforming arrays will be discussed along with the connection of a beamforming array to the INS.

6 BEAMFORMING

6.1 INTRODUCTION

With the INS functional, it is now possible to use the estimated information about the phone's position, together with a beamforming antenna array, to direct the bulk of electromagnetic radiation away from the user. The type of array used will be a uniform rectangular array, fixed to the back face of the phone (i.e. the side of the phone where the battery is usually stored). This chapter will include an introduction to the mathematics of beamforming, followed by a description of the transformations necessary to convert the position and orientation output of the INS to progressive phase angles—which are the inputs to the array.

6.2 M-ELEMENT LINEAR ARRAY

To begin understanding the array used in this project, first consider a set (the array) of antennas (the elements), that are arranged on a straight line, as shown in Fig. 6.2.1 [9]. Suppose also that each element in the array is a copy of the same antenna structure, but may have a different amplitude or phase than the other elements. At any point in the far field, the total field magnitude, created by the superposition of all M elements, is given simply by

$$E_{LINEAR} = E_1 + E_2 + \cdots + E_M . \tag{6.2.1}$$

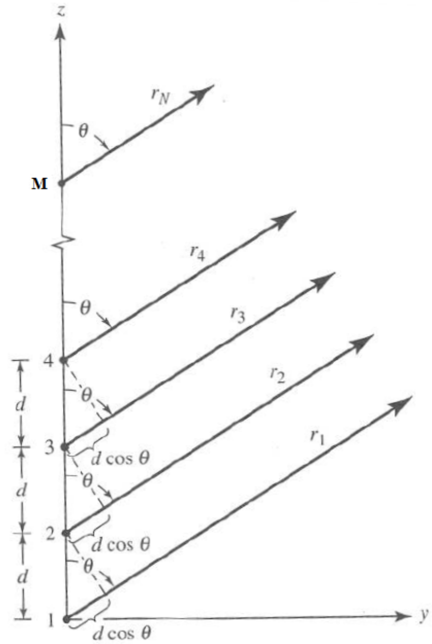


Figure 6.2.1 M element linear array.

For the moment, assume that each of the M elements represents an isotropic source with phase offset θ_i. The electric field for each element is

$$E_i = A_i \frac{e^{j(kr_i+\theta_i)}}{4\pi r_i}, \tag{6.2.2}$$

so the total field is

$$E_{LINEAR} = A_1 \frac{e^{j(kr_1+\theta_1)}}{4\pi r_1} + A_2 \frac{e^{j(kr_2+\theta_2)}}{4\pi r_2} + \cdots + A_N \frac{e^{j(kr_N+\theta_M)}}{4\pi r_M}. \tag{6.2.3}$$

The effect of the r_i are of special importance. In the far field, r_i are near enough in value so that the effect of the denominator in each quotient is approximately equal. On the other hand, the complex exponentials are periodic functions and the differences between r_i's matters greatly. In fact, referring to Fig. 6.2.1, because the r_i's are so nearly parallel, the element to element difference in r_i is approximately $d \cdot \cos(\theta)$. If one lets the bulk radius be R, and the radius difference be given by these $d \cdot \cos(\theta)$'s, (6.2.3) can be written as

$$E_{LINEAR} = A_0 \frac{e^{j(kR+\theta_0)}}{4\pi R} + A_1 \frac{e^{j(k(R+d\cos(\theta))+\theta_1)}}{4\pi R} + \cdots + A_M \frac{e^{j(k(R+Md\cos(\theta))+\theta_M)}}{4\pi R}, \quad (6.2.4)$$

which can be further simplified to

$$E_{LINEAR} = \frac{e^{jkR}}{4\pi R}\left(A_0 e^{j\theta_0} + A_1 e^{j(kd\cos(\theta)+\theta_1)} + \cdots + A_M e^{j(kMd\cos(\theta)+\theta_M)}\right). \quad (6.2.5)$$

Notice the first factor in the right hand side of (6.2.5) is precisely the electric field of an isotropic source! This derivation demonstrates a rule that is also generally true [9]—the field of any antenna array can be given as the product of the field of a single element in the array and a function, called the array factor (AF), which contains information about the spatial and temporal arrangement of the array elements with respect to each other. Equation (6.2.5), therefore, can be written more generally as

$$E_{LINEAR} = E_{ELEMENT}\left(A_0 e^{j\theta_0} + A_1 e^{j(kd\cos(\theta)+\theta_1)} + \cdots + A_M e^{j(kMd\cos(\theta)+\theta_M)}\right), \quad (7.2.6)$$

or

$$E_{LINEAR} = E_{ELEMENT} AF_{LINEAR}. \quad (7.2.7)$$

To continue, two simplifying design choices are used. First, assume all element amplitudes are equal and that the angles θ_i increase by a set amount β from one element to the next. An array with these characteristics is called a uniform array. Equation (6.2.3), as a uniform array, simplifies to

$$E_{LINEAR} = A\frac{e^{jkR}}{4\pi R}\left(\sum_{i=0}^{M} e^{ji(kd\cos(\theta)+\beta)}\right). \quad (6.2.8)$$

Additionally, with a bit of algebra, the sum in (6.2.7) can be simplified further to

$$AF_{LINEAR} = \sum_{i=0}^{M} e^{ji(kd\cos(\theta)+\beta)} = \left[\frac{\sin\left(\frac{M}{2}\Psi\right)}{\sin\left(\frac{1}{2}\Psi\right)} \right], \tag{6.2.9}$$

where

$$\Psi = kd\cos(\theta) + \beta\,. \tag{6.2.10}$$

6.3 UNIFORM PLANAR ARRAY

In this section, the results of 6.2 are abstracted and applied to an array that extends in a second coordinate direction, to achieve a rectangular array of coplanar elements. Consider Fig. 6.3.1(a) [9], its field was calculated in (6.2.9). Suppose now that this array is replicated in the y direction to create the array seen in Fig. 6.3.1(b) [9].

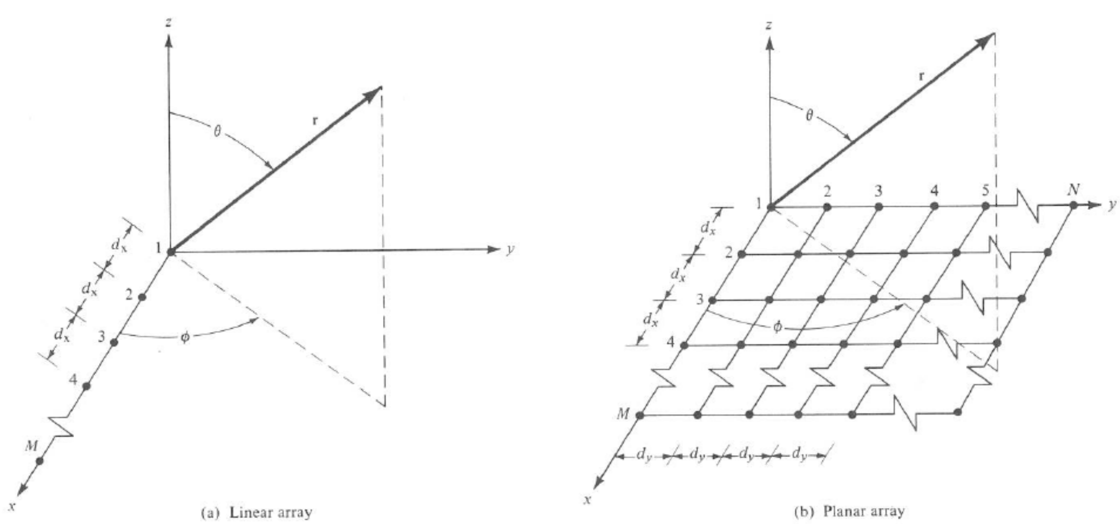


Figure 6.3.1 (a) Linear array of M elements along the x axis. (b) Planar array in the xy-plane, created by replicating the linear array of (a) in the y direction.

Think of (6.2.9) as a single element, and then create an N element linear array of these elements, extending in the y direction. The resulting field is

$$E_{PLANAR} = E_{LINEAR} AF_{y\,LINEAR}. \tag{6.3.1}$$

Expanding E_{LINEAR} yields

$$E_{PLANAR} = E_{ELEMENT} AF_{xLINEAR} AF_{yLINEAR}, \tag{6.3.2}$$

which is

$$E_{PLANAR} = E_{ELEMENT}\left[\frac{\sin\left(\frac{M}{2}\Psi_x\right)}{\sin\left(\frac{1}{2}\Psi_x\right)}\right]\left[\frac{\sin\left(\frac{N}{2}\Psi_y\right)}{\sin\left(\frac{1}{2}\Psi_y\right)}\right], \tag{6.3.3}$$

where,

$$\Psi_x = kd_x\sin(\theta)\cos(\phi) + \beta_x \tag{6.3.4}$$

$$\Psi_y = kd_y\sin(\theta)\sin(\phi) + \beta_y \tag{6.3.5}$$

Before moving on to scanning the array, some explanation is required for (6.3.4) and (6.3.5), which were given without explanation and are not the same as their counterpart (6.2.10). Specifically in (6.2.10), the quantity $k\,d\cos(\theta)$ is the element to element distance in degrees ($d\cos(\theta)$ is the distance in meters, and k is the wave number, which converts meters to radians). Equations (6.3.4) and (6.3.5) include an additional trigonometric factor, which ought to be explained: To arrive at (6.3.4), consider the abstraction where $AF_{xLINEAR}$ is the array, and $E_{ELEMENT}AF_{yLINEAR}$ are the elements of the array. Equation (6.3.2) would be rewritten as

$$E_{PLANAR} = E_{ELEMENT'} AF_{xLINEAR}, \tag{6.3.6}$$

where $E_{ELEMENT'} = E_{ELEMENT} AF_{yLINEAR}$.

Fig. 6.3.2 shows the geometry of (6.3.6) for the first two elements (A and B) in the array. To find the phase difference caused by the different lengths of $\mathbf{r}_A$ and $\mathbf{r}_B$, one must calculate the length of dp.

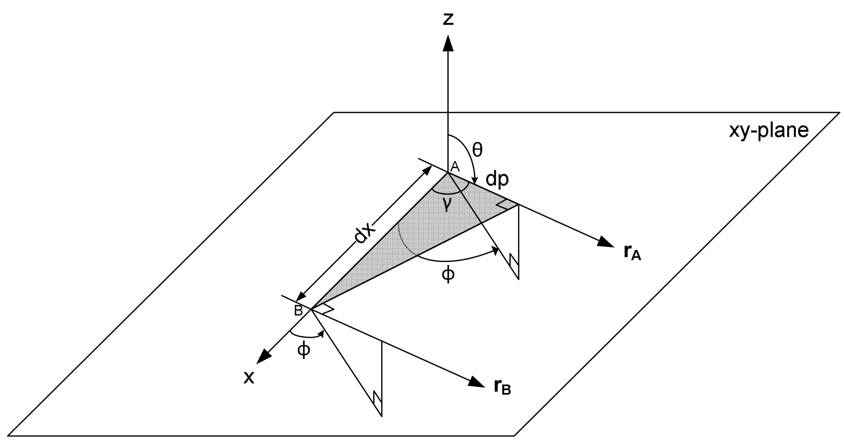


Figure 6.3.2 3D view of the array representation of (6.3.5) with two $E_{ELEMENT}$'s at vertices A and B. The shaded triangle (which is not in the xy-plane) is the shape of interest. The side of this triangle labeled dp provides the element to element distance difference. For clarity, ϕ is in the xy-plane, and γ is in the plane of the shaded region.

The hypotenuse of the grey shaded triangle is dx. If angle γ can be determined, then d_p is simply

$$dp = dx\cos(\gamma)\,. \tag{6.3.7}$$

Angle γ can be found using the dot product of a unit vector pointing in the $\mathbf{r}_A$ direction and the x-direction basis vector.

$$\cos(\gamma) = \begin{pmatrix}1 & 0 & 0\end{pmatrix} \cdot \begin{pmatrix}\sin(\theta)\cos(\phi) & \sin(\theta)\sin(\phi) & \cos(\theta)\end{pmatrix} \tag{6.3.8}$$

$$\gamma = \mathrm{acos}(\sin(\theta)\cos(\phi)) \tag{6.3.9}$$

Lastly, substitution of (7.3.9) into (7.3.7) gives

$$dp = dx\sin(\theta)\cos(\phi)\,. \tag{6.3.10}$$

Multiplication by the wave number will then give dp in radians, and matches (6.3.4). The derivation for (6.3.5) would proceed in a similar fashion.

6.4 SCANNING

Scanning is the process of varying the amplitude, phase, or geometry of an array in order to point the main beam in a certain direction. From (6.3.2), the array will have its maximum when both Ψ_x and Ψ_y are zero. β_x and β_y, the progressive phase shifts of the element in the x and y directions, are adjustable; therefore, a user may point the radio beam in a particular direction θ_0, ϕ_0, by solving for β_x and β_y with Ψ_x and Ψ_y equal to zero. Solving (6.3.4) and (6.3.5) for β_x and β_y gives

$$\beta_x = -kd_x\sin(\theta_0)\cos(\phi_0)\,, \tag{6.4.1}$$

and

$$\beta_y = -kd_y\sin(\theta_0)\cos(\phi_0)\,, \tag{6.4.2}$$

which are the phase shifts required to direct the radio beam in direction θ_0, ϕ_0.

6.5 PLANAR ARRAY SCANNING EXAMPLE

To demonstrate the effectiveness of the uniform planar scanning array, three sample radiation patterns are shown below for an isotropic source and an array with four elements in the y direction and three elements in the x direction (a four by three array is the one designed in [2]). In Fig. 6.5.1, the rectangle at the origin plane of the array; no phase shift is applied, so the array's main beam is directed completely in the z directions. The field is symmetric about the plane of the array because no information was given about what type of object the array is resting on, if any. Without such specification, the uniform array looks identical both from above and underneath. In the same way, if one were to draw uniformly spaced dots on a transparency paper, it would be impossible for viewers to determine which side of the transparency was the "top", and which was the "bottom". Fig. 6.5.2 shows the system with a θ=45°, ϕ=30° scan.

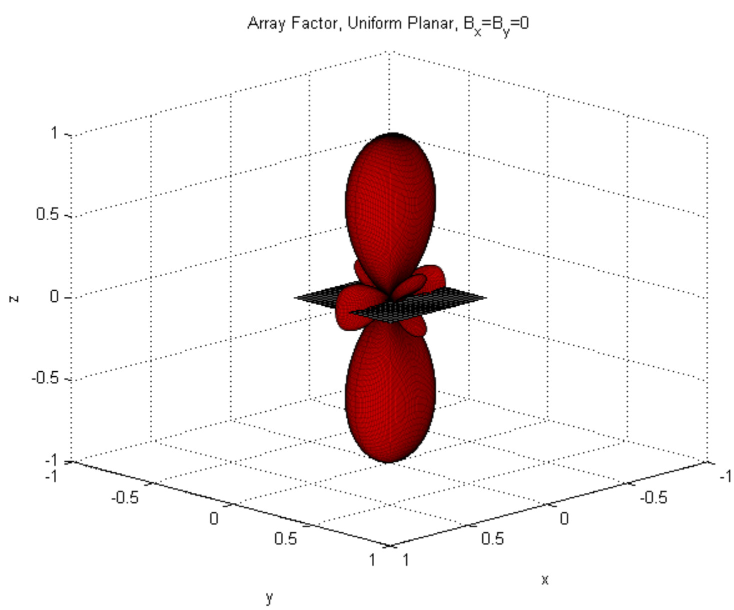


Figure 6.5.1 Vertical array pattern with no progressive phase shift.

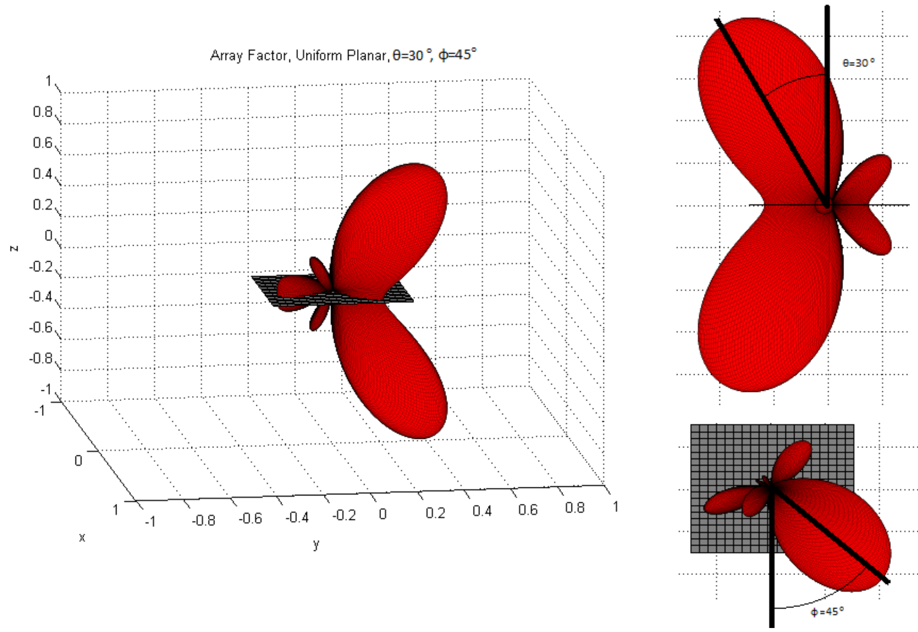


Figure 6.5.2 Same array as in Fig. 6.5.1, but with nonzero progressive phase shifts that combine to scan the main beam to θ=30^o and ϕ=45^o.

6.6 USING ASYMETRIC ELELEMENTS

The symmetry about the xy-plane seen in Fig. 6.5.1 and Fig. 6.5.2 is totally undesired. The rectangle shown at the origin in these examples will ultimately represent the back of a cellular phone, therefore, it is required that the bulk of the radiation is present only on one side of this rectangle. To achieve this, the array designed specifically for this application by [2], which uses microstrip patch antennas, is used. Unlike the isotropic sources used in sections 6.2 to 6.5, microstrip patch antennas have the useful property that they transmit negligible power behind the plane of the array. The radiation pattern for a patch antenna source shown below in Fig. 6.6.1. Notice that the field radiation does not extend behind the plane. One objection to this pattern might be the obvious discontinuities at the plane surface in the figure. The pattern shown is a first order approximation. Real microstrip patch antennas have a negligibly small amount of far field radiation behind the array plane. The bigger inaccuracy in this approximation arises in showing that the field touches the plane and then abruptly stops. In fact, the field diminishes asymptotically as it approaches the plane so that the field doesn't actually touch the plane. For testing the effectiveness of the INS to mitigate radiation exposure, the first order approximation is quite effective.

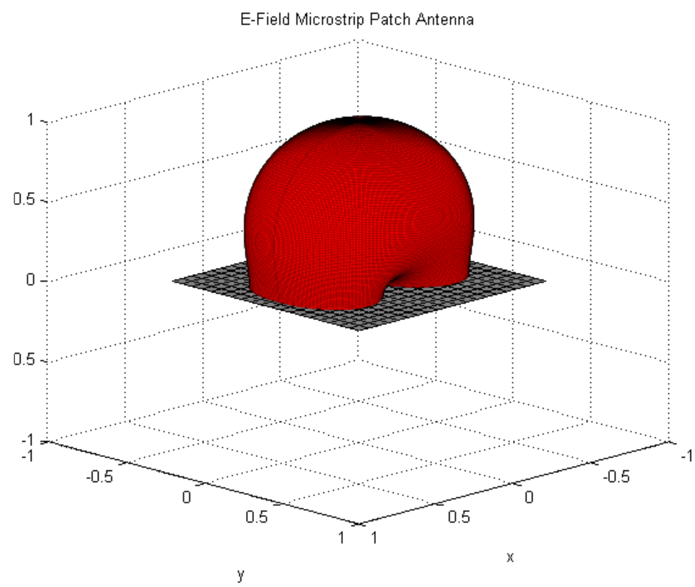
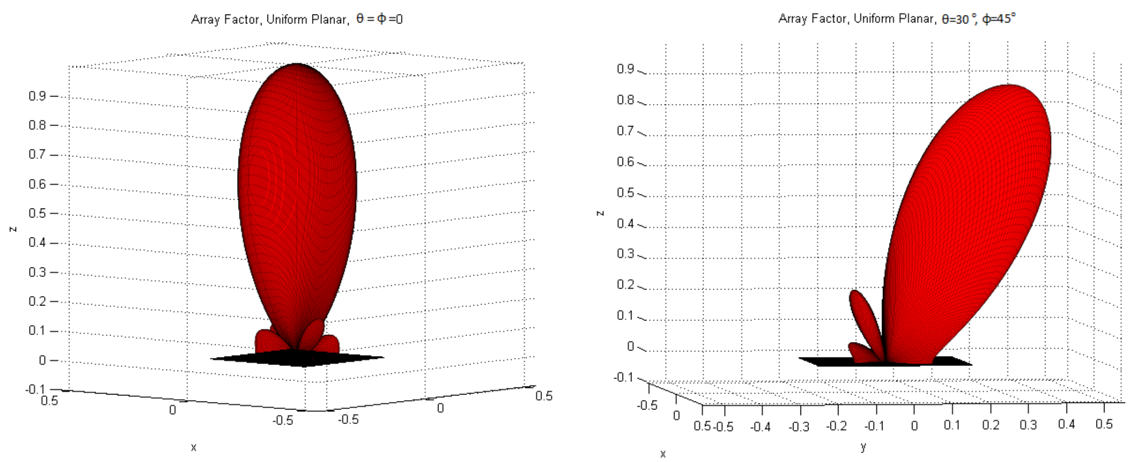


Figure 6.6.1 First order approximation of a microstrip patch antenna.

Returning to (6.3.2), the first factor in the product is replaced by the electric field of a microstrip patch antenna E_{MSP}, so that (6.3.2) becomes

$$E_{PLANAR} = E_{MSP}\left[\frac{\sin\left(\frac{M}{2}\Psi_x\right)}{\sin\left(\frac{1}{2}\Psi_x\right)}\right]\left[\frac{\sin\left(\frac{N}{2}\Psi_y\right)}{\sin\left(\frac{1}{2}\Psi_y\right)}\right]. \qquad (6.6.1)$$

The new radiation patterns are shown in Fig. 6.6.2



(a) (b)

Figure 6.6.2 Examples of beamforming with microstrip patch antennas. (a) No progressive phase. (b) Scanned to $\theta=30^{o}$ and $\phi=45^{o}$.

6.7 BINOMIAL ARRAY EXCITATION

Before moving on to the task of connecting the array to the INS, one other type of array ought to be considered in testing: binomial excitation. Binomial arrays have the distinct property that, for low scan angles, they do not have sidelobes, which may prove to be very desirable in protecting users from radiation. In binomial excitation, the relative amplitudes of successive elements are distributed according to a binomial tree, the first six rows of which are shown below in Fig. 6.7.1

m	Relative Element Amplitudes										
1						1					
2					1		1				
3				1		2		1			
4			1		3		3		1		
5		1		4		6		4		1	
6	1		5		10		10		5		1

Figure 6.7.1 Relative element to element amplitudes for a binomially excited linear array.

To find an AF and total field for a binomial planar array, one must first return to (6.2.6), which is repeated below.

$$E_{LINEAR} = E_{ELEMENT}\left(A_0 e^{j\theta_0} + A_1 e^{j(kd\cos(\theta)+\theta_1)} + \cdots + A_M e^{j(kMd\cos(\theta)+\theta_M)}\right) \tag{6.2.6}$$

From [2], there are three elements in the x direction, and four elements in the y direction, these correspond to the linear array factors seen in (6.7.1) and (6.7.2) below.

$$AF_{xLINEAR} = e^{j\theta_0} + 2e^{j(kd\cos(\theta)+\theta_1)} + e^{j(k2d\cos(\theta)+\theta_2)} \tag{6.7.1}$$

$$AF_{yLINEAR} = e^{j\theta_0} + 3e^{j(kd\cos(\theta)+\theta_1)} + 3e^{j(k2d\cos(\theta)+\theta_2)} + e^{j(k3d\cos(\theta)+\theta_2)} \tag{6.7.2}$$

Using the progressive phase requirement (i.e. $\theta_i = i\cdot\beta$), one can rewrite (6.7.1) and (6.7.2) as

$$AF_{xLINEAR} = 1 + 2e^{j(kd\cos(\theta)+\beta_x)} + e^{j2(kd\cos(\theta)+\beta_x)}, \tag{6.7.1}$$

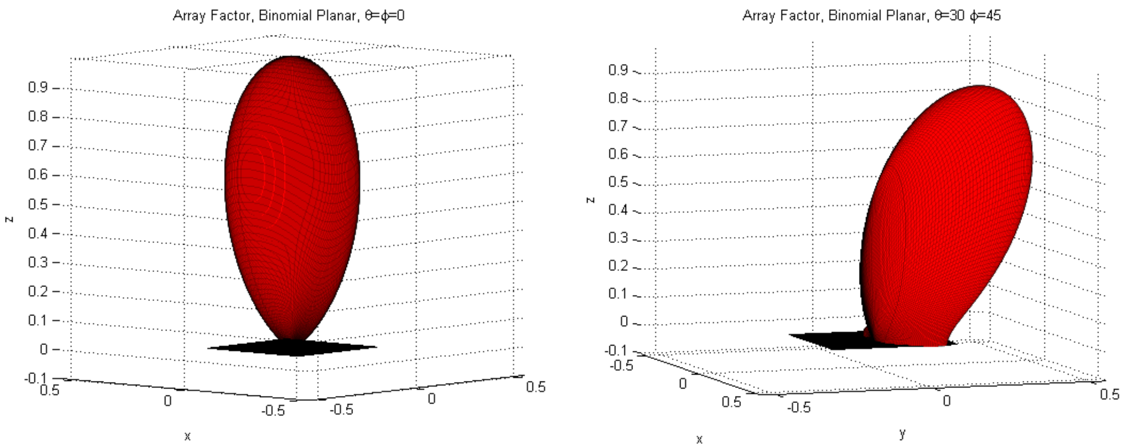
and,

$$AF_{yLINEAR} = 1 + 3e^{j(kd\cos(\theta)+\beta_y)} + 3e^{j2(kd\cos(\theta)+\beta_y)} + e^{j3(kd\cos(\theta)+\beta_y)} \tag{6.7.2}$$

The planar field, according to (6.3.2), is now

$$E_{PLANAR} = E_{MSP} AF_{xLINEAR} AF_{yLINEAR}. \tag{6.7.3}$$

Because the progressive phase shift is defined in the same way as in section 6.3, the binomial array can be scanned using (6.4.1) and (6.4.2). For comparison, the radiation patterns with the scan angles of Fig. 6.6.2 are repeated using a binomially excited array and shown in Fig. 6.7.2. The two main differences between the uniform and binomial excitation are the lack of prominent sidelobes with binomial excitation and noticeably wider beamwidths.



(a) (b)

Figure 6.7.2 Examples of beamforming with microstrip patch antennas and binomially excited elements. (a) No progressive phase. (b) Scanned to $\theta=30^o$ and $\phi=45^o$. A small sidelobe can be seen protruding from behind the mainlobe in (b).

6.8 ATTACHING THE ARRAY TO THE INS

With a functional array and microstrip patch design, the INS can be married to a controller that calculates the necessary progressive phase shifts, β_x and β_y, which will direct radiation away from unsuspecting users. In the following sections, the INS will be expanded to its final form—a system that takes accelerometer sensor inputs and returns the progressive phase shifts required to direct the radio beam in an optimal direction.

In Fig. 6.3.1 the antenna array was laid out on the xy-plane, but in implementation the array will be in the xz-plane, affixed to the back of the phone, as shown in Fig 6.8.1. The patch array designed in [2] has four elements in the vertical (z) direction and three elements in the horizontal (x) direction.

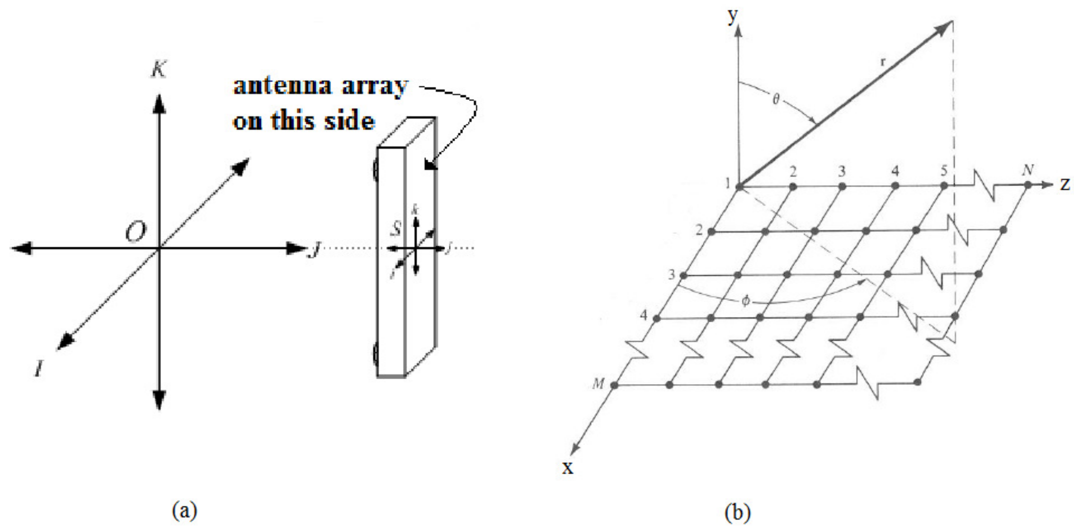


Figure 6.8.1 (a) Antenna array location. (b) Antenna array orientation.

6.9 POINTING THE BEAM

The vector **r** shown in Fig. 6.8.1(b) represents the desired beam direction provided by the INS. This vector is simply vector $\mathbf{r}_{A,S}$ from chapter 2. Fig. 6.9.1 shows this fact: by simply attaching $\mathbf{r}_A$ to the origin of S, one has a vector that points away from the user. Unfortunately, however, $\mathbf{r}_A$ is in the coordinates of O, while the array, because it is fixed to the back of the phone, is defined in frame S. Using $\underline{\underline{F}}$ from the INS, one can rotate $\mathbf{r}_A$ back into the coordinates of S. After this rotation is made, the geometry of the system is now identical to the geometry shown in Fig. 6.8.1(b), so that angles θ and ϕ are easily calculated as,

$$\theta = \mathrm{acos}\left(\frac{r_{y,A,S}}{\left\|\mathbf{r}_{A,S}\right\|}\right) \tag{6.9.1}$$

and,

$$\phi = \mathrm{atan}\left(\frac{r_{z,A,S}}{r_{x,A,S}}\right) \tag{6.9.2}$$

Lastly, one can calculate the progressive phase angles using (6.4.1) and (6.4.2). Equation (6.9.1) may return any angle up to 90°, but the maximum scan angle for θ is limited to about 30°, per the recommendation of [2].

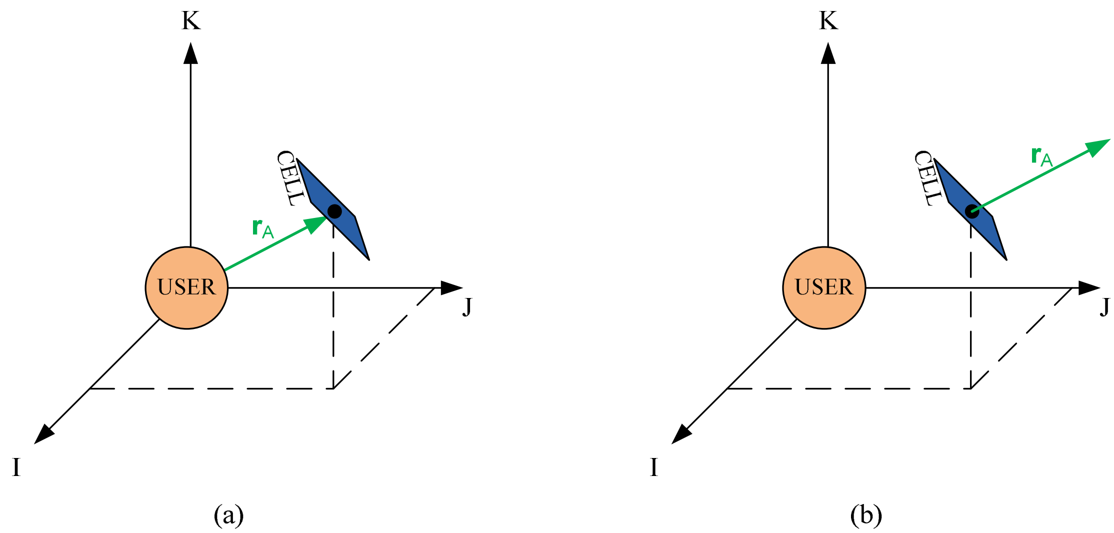


Figure 6.9.1 (a) Vector r_A, which points to the origin of frame S. (b) Affixing r_A to the origin of S provides the optimal direction for the main beam of the array.

6.10 ADDITION TO THE MODEL

The flow diagram that implements these calculations is shown in Fig. 6.10.1. The labels $\mathbf{p}_O$ and $\mathbf{p}_S$ represent vectors $\mathbf{r}_{A,O}$ and $\mathbf{r}_{A,S}$, respectively; and $\underline{\underline{F}}$ is the currently simulated rotator matrix.

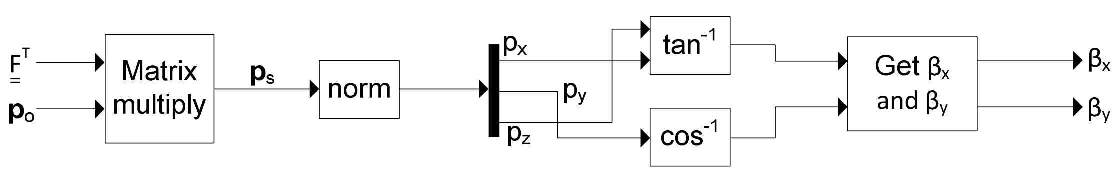


Figure 6.10.1 Flow diagram of the progressive phase shift calculations.

6.11 CONCLUSION / LOOKING AHEAD

With the beamforming array now controlled by the INS, a final performance analysis can be conducted. In chapter 7 the complete system will be simulated and radiation measured to see what kind of reduction in exposure users can expect when using a system of this type.

7 PERFORMANCE ASSESSMENT

7.1 INTRODUCTION

With the model complete, it is now possible to measure the effectiveness of the combined INS antenna array system. For this analysis, three typical motions a phone might experience are simulated and the percentage of radiation directed toward the user is measured. The three motions are: (1) a user turning his head, (2) a user rotating the phone in hand, and (3) the user picking up the phone. For comparison, assume a typical cellular phone's radiation pattern is omnidirectional, so that a high percentage of transmitted power is always directed at the user. The following simulations will be performed with both the uniform and binomial array excitations, but snapshots are shown only for the uniform field. Profiles of the radiation directed toward the user will be given for both the uniform and binomial array cases.

7.2 MOTION 1—TURNING THE HEAD

This motion simulates a user that turns the head, pauses, and turns back, all the time holding the phone to his/her ear. A few samples of this motion are shown in Fig. 8.6.1. In the figure, the small green sphere at the center is the user's head. The blue semi-transparent hemisphere shows the region of integration of bad radiation. The red lobes extending to the right and down in the snapshots are the radiation pattern of the uniform array.

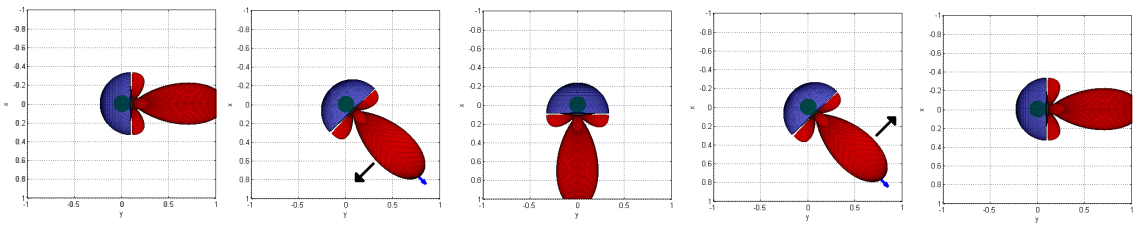


Figure 7.2.1 Snapshots of uniform array field as the user rotates his/her head.

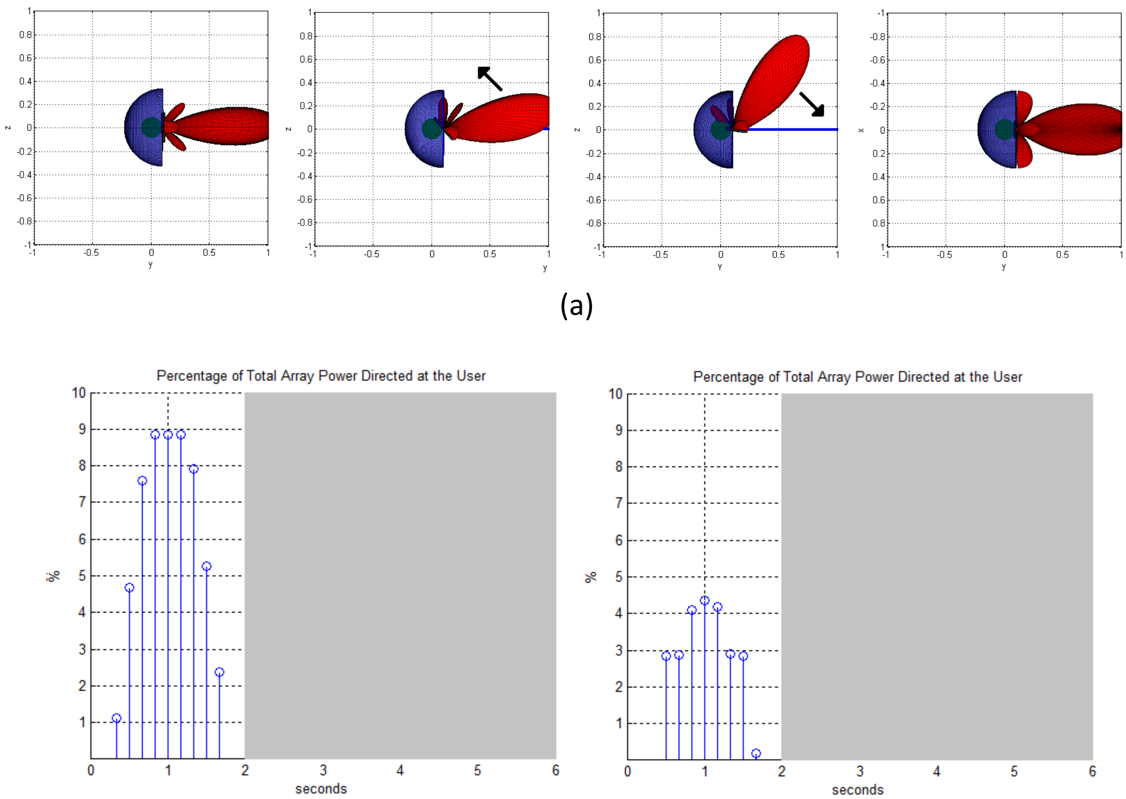
As the user turns his head, the beam direction, though changing with respect to the user, does not change with respect to the phone. The direction normal to the surface of the phone always presents the optimal direction for the beam, which is why the beam shape is unchanging across all the frames. For this reason, the motion of the user turning his head is rather trivial. The INS plays no role, because the radio beam never needs to be moved from its nominal direction. This motion does demonstrate the effectiveness of the antenna array (no radiation is directed at the user), but not the INS.

7.3 MOTION 2—ROTATING THE PHONE

In this motion, the user rotates the phone in his/her hand; first about the x-axis, then about the y-axis, then about the z-axis. This motion is more complex from the beam directing perspective, and is a little easier to see if broken up into three sub-motions.

7.3.1 X-AXIS ROTATION

In the first part of the motion, the phone is rotated about the x-axis, as shown in Fig. 7.3.1.1. As the phone is rotated, a side lobe of the radiation pattern is seen to impinge on the user. In (b) and (c) of the figure, the back radiation from both the uniform and binomial arrays are shown. The binomial array exposes the user to about half the radiation as the uniform array. Both configurations are much better than a typical cellular antenna.



(a)

(b) (c)

Figure 7.3.1.1 (a) In this yz-plane view, the phone is rotated about the z-axis. (b) Back radiation for the uniform array. (c) Back radiation for the binomial array.

7.3.2 Y-AXIS ROTATION

In the second part of the motion, the phone is rotated about the y-axis. This motion represents a trivial situation similar to section 7.2, in which the beam's natural direction is also the optimal one. In Fig. 7.3.2.1 no radiation reaches the user for rotations for either type of array.

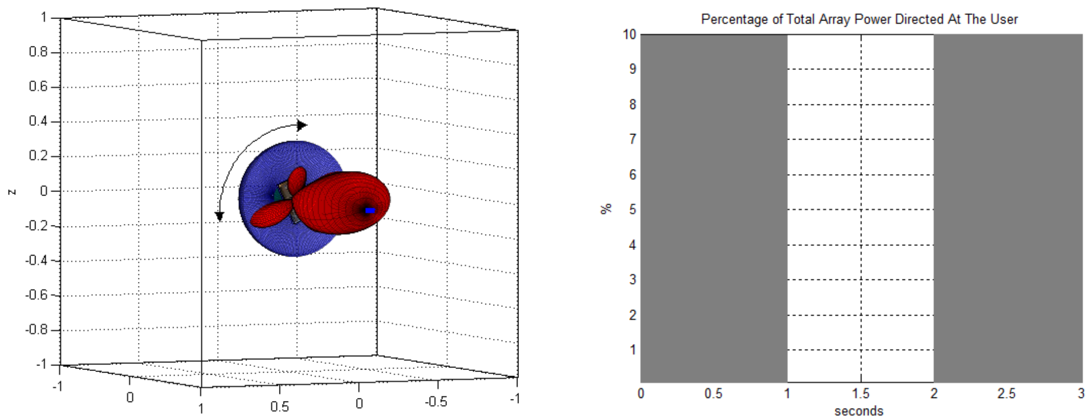
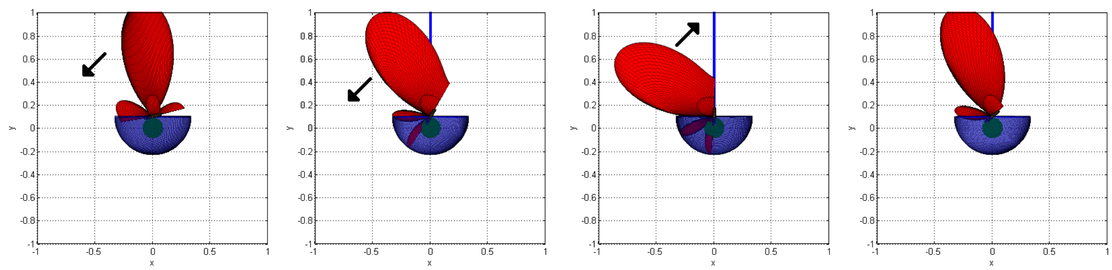


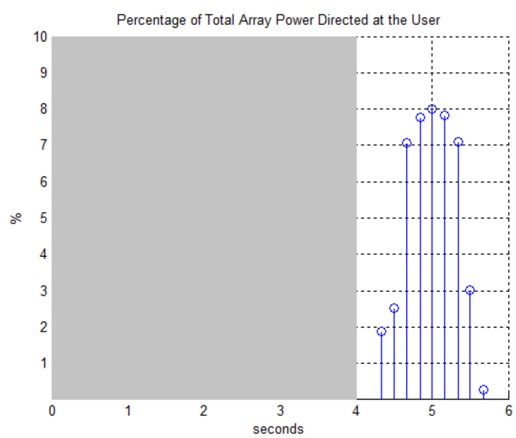
Figure 7.3.2.1 In this 3D view, the phone is rotated about the y-axis. No radiation reaches the user.

7.3.3 Z-AXIS ROTATION

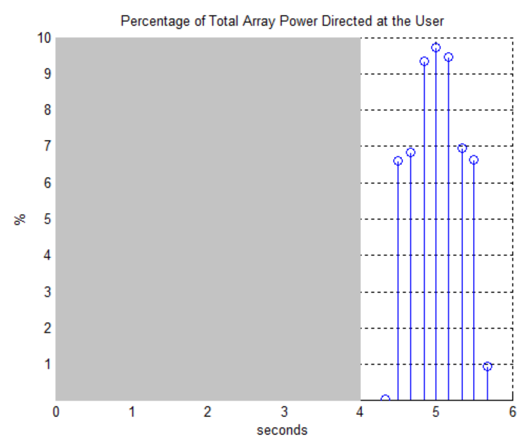
In the third part of the motion, the phone is rotated about the z-axis, as shown in Fig. 7.3.3.1. As with the x-axis rotation of section 7.3.1, a side lobe of the radiation pattern is seen to impinge on the user. In (b) and (c) of the figure, the back radiation from both the uniform and binomial arrays are shown. Unlike in section 7.3.1, binomial excitation exposes the user to slightly more radiation than the uniform array.



(a)



(b)

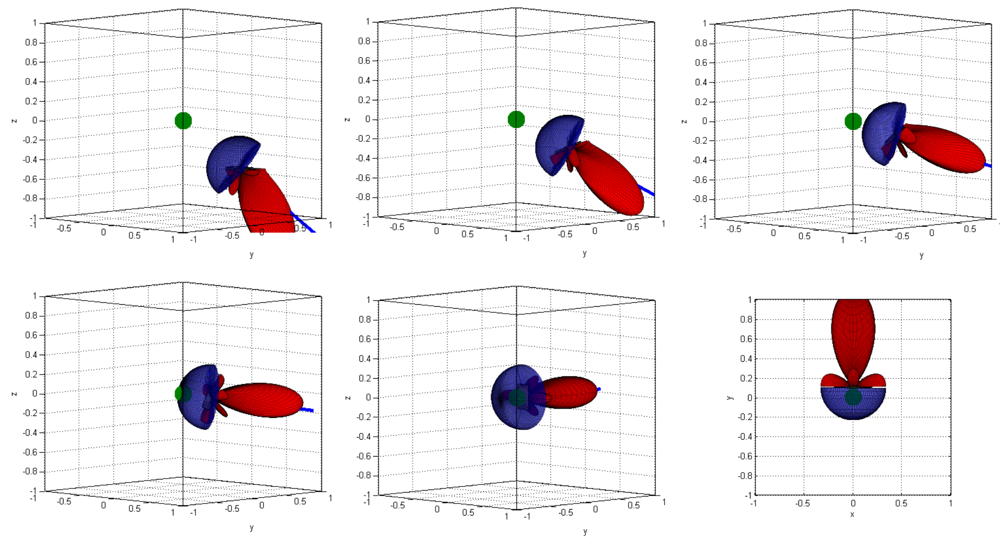


(c)

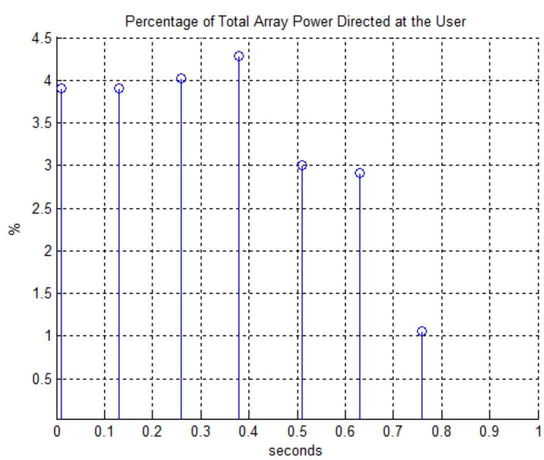
Figure 7.3.3.1 (a) In this xy-plane view, the phone is rotated about the x-axis. (b) Back radiation for the uniform array. (c) Back radiation for the binomial array.

7.4 MOTION 3—USER PICKS THE PHONE UP FROM TABLE

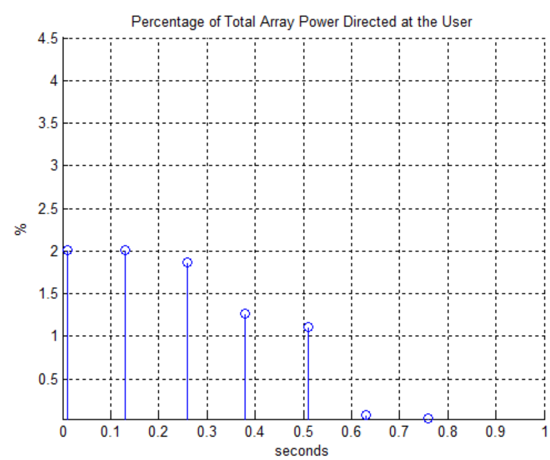
In this motion, the user picks up the phone from a table or desk and brings it to his ear. The phone is initially lying with antenna facing down and the top of the phone away from the user. Fig. 7.4.1 shows the snapshots and bad radiation percentage for this motion. Comparing the uniform and binomial arrays, the binomial array outperforms the uniform array, exposing the user to about half the radiation as the uniform array.



(a)



(b)



(c)

Figure 7.4.1 (a) User picks up the phone from a desk or table and brings it to his/her ear. (b) Back radiation for the uniform array. (c) Back radiation for the binomial array.

7.5 CONCLUDING REMARK

The total system performs well and the binomial array typically outperforms the uniform array. The only time the uniform array outperforms the binomial array is near the scan angles $\theta=90^o$, $\phi=\{0, \pi\}$. In these cases, the binomial array's beamwidth is greater than twice maximum scan angle, so a portion of the main beam impinges on the user. A zoomed in view for the binomial array is shown for this situation in Fig. 7.5.1.

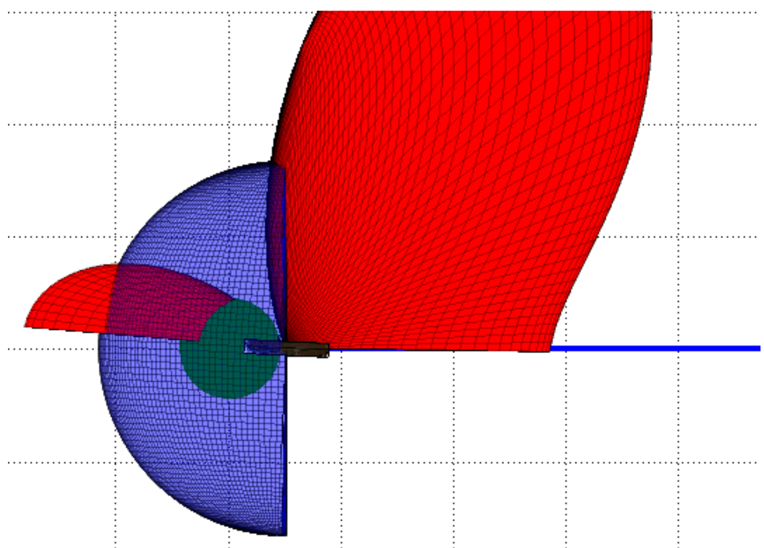


Figure 7.5.1 The main beam of the binomial array sometimes enters the plane of the user for extreme rotations of the phone. This does not happen for a uniform array (compare to snapshot 3, Fig. 7.3.3.1(a)).

8 CONCLUDING REMARKS / LOOKING AHEAD

8.1 CONCLUDING REMARKS

The results of chapter 7 demonstrate the effectiveness of beamforming, coupled with an INS, to reduce users' exposure to radiation from their cellular phones. The results are encouraging. Even in the worst case (section 7.3), total radiation exposure never even reaches 9% of total power. Compared to a typical omnidirectional source, which bombards the user with a high percentage of total radiation continuously, this method provides a huge reduction in radiation exposure.

In a real implementation, both the antenna array and INS controller would be straightforward to implement with existing technology. The major hurdle to any real implementation is the presence of noise. There is reason to be optimistic that this will not prove to be a design killer, however, because the noise parameters and sample rates used in these simulations were quite pessimistic—they are typical of low end accelerometers currently available. Future generations of MEMS accelerometers will no doubt have reduced noise variances and higher sample rates. In fact, because the bandwidth of human motion is limited, if MEMS accelerometer sample rates improve in the near future by even a few times, one could consider the input to be heavily oversampled, and even more signal processing and estimation methods could be brought to bear, improving signal-to-noise ratio dramatically.

8.2 LOOKING AHEAD

For future designs, a few other additions might be appropriate for improving performance.

The first design addition would be to include different types of sensors to complement and correct the INS. Traditional inertial systems have used additional sensors (i.e. magnetic compasses, speed gauges, and altimeters) for decades to correct their calculations. More recently, some GPS guidance systems use INS to estimate their position with fine resolution between GPS samples. In the present application, sensors might be thermal, optical, or acoustic.

Binomial planar arrays were found to be a bit more effective at protecting users from unwanted radiation, except when scan angles were extreme. In future designs, one might consider an adaptive, in which the controller uses a combination of binomial and uniform element excitations to give the user the best protection in any situation.

From chapter 4 it was seen that noise variance, however reduced by better sensors, sample rates, and accelerometer geometries, still grows at a rate proportional to t^7 or worse. If an INS could turn itself off while stationary, it would be possible to reduce t. Traditional INS do not incorporate any such feature because their applications prohibit it: an airplane traveling at constant velocity, for example, experiences no acceleration, but that does not mean it is not moving! In the present application, however, accelerations must be followed by almost immediate decelerations because human motion is, quite literally, limited to the arm’s reach. For this reason, a phone that is not accelerating or decelerating is almost certainly not moving with respect to the user. An INS that recognizes this fact may provide improved performance in terms of longer running time without the need for correct / resetting.

BIBLIOGRAPHY

[1] E. Landau, "Evidence Still Fuzzy On Cell Phones, Cancer," CNN Health, [Online document], Nov. 10, 2009, [cited 2009 Dec 7]. Available HTTP: http://www.cnn.com/2009/HEALTH/11/10/cell.phones.cancer.tumors/index.html

[2] W. Gruenwald, Design of a Smart Antenna For Cellular Phones For Minimal Radiation Absorption By The Human Brain/Body. Offenburg: University of Applied Sciences, 2009

[3] A. D. King, "Inertial Navigation—Forty Years of Evolution," GEC Review, vol. 13, no. 3, 1998

[4] C. Tan and S. Park, "Design of Accelerometer-Based Inertial Navigation Systems," IEEE Transactions On Instrumentation and Measurement, vol. 54, no. 6, Dec. 2005

[5] B. Stark, Ed., MEMS Reliability Assurance Guidelines for Space Applications, Pasadena: Jet Propulsion Laboratory, 1999

[6] [Online document], [cited 2010 Mar 9]. Available HTTP: http://mems.sandia.gov/gallery/images/Bug_1b.jpg

[7] "Small, Low Power, 3-Axis ±5 g Accelerometer," Analog Devices, [Online document], [cited 2010 Mar 9]. Available HTTP: http://www.analog.com/static/imported-files/data_sheets/ADXL325.pdf

[8] R. C. Hibbeler, Engineering Mechanics Dynamics, 8th ed., Upper Saddle River: Prentice-Hall, 1998

[9] C. A. Balanis, Antenna Theory – Analysis and Design, 2th ed., New York: John Wiley & Sons Inc, 1997.

[10] R. A. Serway, R. J. Beichner and J. W. Jewett Jr., Physics for Scientists and Engineers, 5th ed., Forth Worth: Saunders College Publishing, 2000

[11] S. Ross, A First Course In Probability, 7th ed., Upper Saddle River: Prentice Hall, 2006

[11] W. E. Boyce and R. C. DiPrima, Elementary Differential Equations and Boundary Value Problems, 7th ed., New York: John Wiley and Sons, 2001.

[12] V. M. Zatsiorsky, Kinematics of Human Motion. Human Kinetics, 1998.

APPENDIX A — SIMULINK MODEL

The Simulink models shown throughout the report were used to illustrate the implementation of the equations and algorithm developed in chapter 2. Some supporting blocks, such as unit delays, and output sinks, were removed for clarity. Additionally, some of the blocks in the report were subsystems. These subsystems are also provided in this appendix.

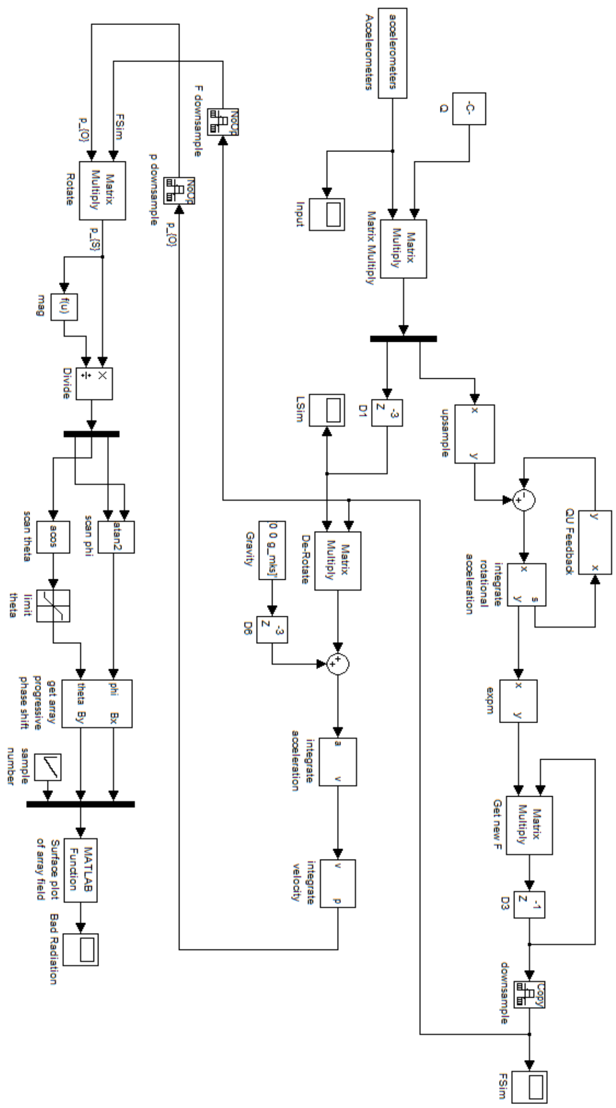


Figure A1 Complete top level view of the INS and beam controlling system.

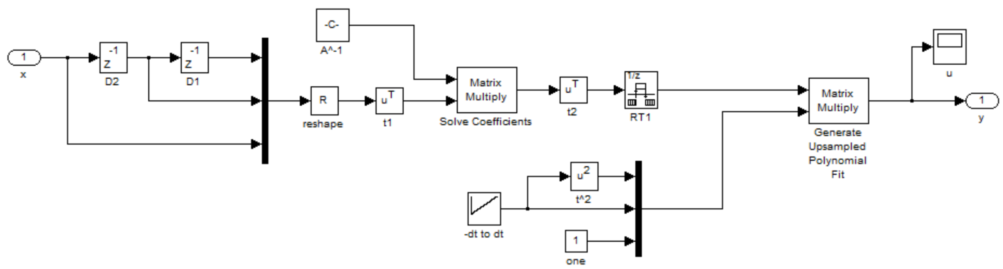


Figure A2 Upsample subsystem.

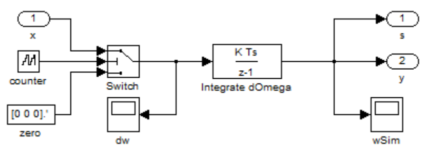


Figure A3 Integrate rotational acceleration subsystem.

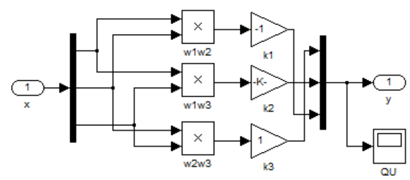


Figure A4 QU feedback subsystem.

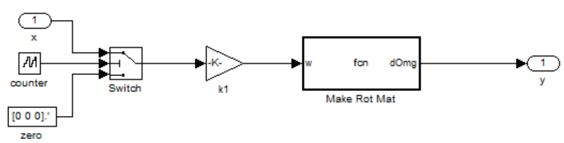


Figure A5 Expm subsystem.

```
function dOmg = fcn(w)

dOmg=expm([     0  -w(3)    w(2); ...
             w(3)       0  -w(1); ...
            -w(2)    w(1)       0]);
```

Figure A6 Code of the embedded m-function Make Rot Mat.

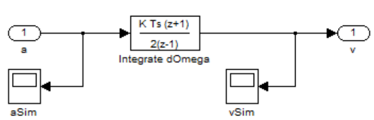


Figure A7 Integrate acceleration subsystem.

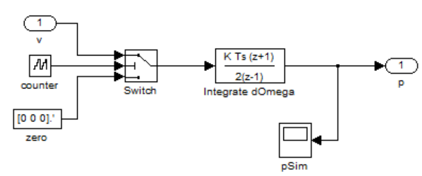


Figure A8 Integrate velocity subsystem.

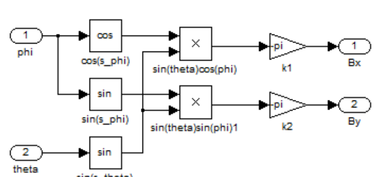


Figure A9 Get array progressive phase shift subsystem.

APPENDIX B — NOISE VARIANCE DETAILS

The noise term at the position output is.

$$p = \begin{bmatrix} T^4 \sigma g \sqrt{q_2} \sum\sum\sum\sum \underline{z} \\ T^4 \sigma g \sqrt{q_1} \sum\sum\sum\sum \underline{z} \\ - \end{bmatrix}. \tag{4.5.4}$$

To compute this sum it will be useful to exploit the property that a sum of independent normal random variables has variance equal to the sum of the individual variances. It would be tempting to compute a closed form for this sum one at a time, four times; however, after the first sum, successive noise terms are no longer independent and can no longer be summed in a simple way. The quadruple sum, therefore, was computed in one step. A table of the sum for the first six samples was used to discover the pattern of growth for the sum. Fig B1 shows these samples.

	err:	a	b	C	d	e	F
SUM	1	a	a + b	a + b + c	a + b + c + d	a + b + c + d + e	a + b + c + d + e + f
	2	a	2a + b	3a + 2b + c	4a + 3b + 2c + d	5a + 4b + 3c + 2d + e	6a + 5b + 4c + 3d + 2e + f
	3	a	3a + b	6a + 3b + c	10a + 6b + 3c + d	15a + 10b + 6c + 3d + e	21a + 15b + 10c + 6d + 3e + f
	4	a	4a + b	10a + 4b + c	20a + 10b + 4c + d	35a + 20b + 10c + 4d + e	56a + 35b + 20c + 10d + 4e + f

Figure B1 Letters a through f are standard normal random variables for the first through sixth sample, respectively. Successive rows show the result of the first through fourth sums.

From the sums in Fig. B1, expressions can be written for the output of each integrator. These expressions are given in Fig. B2, below.

SUM	1	$\sum_{i=0}^{n} \underline{z}_i = n$
	2	$\sum_{j=0}^{n}\sum_{i=0}^{j} \underline{z}_i = n\underline{z}_1 + (n-1)\underline{z}_2 + \cdots + 2\underline{z}_{n-1} + \underline{z}_n$
	3	$\sum_{k=0}^{n}\sum_{j=0}^{k}\sum_{i=0}^{j} \underline{z}_i = \underline{z}_1\sum_{i=1}^{n} i + \underline{z}_2\sum_{i=1}^{n-1} i + \cdots + \underline{z}_{n-1}\sum_{i=1}^{2} i + \underline{z}_n$
	4	$\sum_{m=0}^{n}\sum_{k=0}^{m}\sum_{j=0}^{k}\sum_{i=0}^{j} \underline{z}_i = \frac{\underline{z}_1}{2}\sum_{i=1}^{n} i(i+1) + \frac{\underline{z}_2}{2}\sum_{i=1}^{n-1} i(i+1) + \cdots + \frac{\underline{z}_{n-1}}{2}\sum_{i=1}^{2} i(i+1) + \underline{z}_n$

Figure B2. Expressions for the noise at each stage in the quadruple sum.

Of particular interest is the output of the fourth sum, which corresponds to the noise in the position vector. Therefore, a closed form expression for the noise will be sought only for the fourth sum. To calculate noise in the position, first note that the coefficient associated with each $\underline{z}_i$ is of the form

$$c_j = \frac{1}{2}\sum_{i=1}^{j} i(i+1), \tag{B.1}$$

where j varies from one up to n. Computation of this sum reveals that the c_j^{th} coefficient is

$$c_j = \frac{j^3}{3} + j^2 + \frac{2j}{3}. \tag{B.2}$$

Substituting (B.2) into the fourth sum of Fig. B.2 yields

$$\sum_{m=0}^{n}\sum_{k=0}^{m}\sum_{j=0}^{k}\sum_{i=0}^{j} \underline{z}_i = \frac{c_n}{2}\underline{z}_1 + \frac{c_{n-1}}{2}\underline{z}_2 + \cdots + \frac{c_2}{2}\underline{z}_1 + \frac{c_1}{2}\underline{z}_1. \tag{B.3}$$

Equation (B.3) now shows the quadruple sum in a form where the sum of the variances [11] can be taken, i.e.

$$\sum_{m=0}^{n}\sum_{k=0}^{m}\sum_{j=0}^{k}\sum_{i=0}^{j}\underline{z}_i = N\left(0, \sum_{i=1}^{n}\frac{c_i^2}{4}\right). \tag{B.4}$$

Computing the sum in the variance term yields

$$\sum_{i=1}^{n} c_i^2 = \frac{n^7}{252} + \frac{n^6}{24} + \frac{61}{364}n^5 + \frac{4}{12}n^4 + \frac{23}{72}n^3 + \frac{n}{140}. \tag{B.5}$$

Beyond the first few samples in (B.5), the lowered ordered terms are negligible, therefore, only the highest ordered term is kept in the final expression for noise in the position. Substituting (B.4) and (B.5) into (4.5.4), and keeping only the highest ordered term yields the final expression for the noise, in terms of the sample number.

$$p = \begin{bmatrix} \frac{T^4 \sigma g}{6}\sqrt{\frac{q_2 n^7}{7}}\underline{z} \\ \frac{T^4 \sigma g}{6}\sqrt{\frac{q_1 n^7}{7}}\underline{z} \\ - \end{bmatrix} \tag{B.6}$$

One can also move T^4 inside the square root to arrive at an approximate expression in terms of t rather than n,

$$p = \begin{bmatrix} \frac{\sigma g}{6}\sqrt{\frac{q_2 T t^7}{7}}\underline{z} \\ \frac{\sigma g}{6}\sqrt{\frac{q_1 T t^7}{7}}\underline{z} \\ - \end{bmatrix} \tag{B.7}$$