Extreme Joystick:

A Cobot with Stored Energy

Proposal Submitted by:
Julio J. Santos-Munné

In Partial Fulfillment of the Requirements for the
Degree Doctor of Philosophy
in Mechanical Engineering

Northwestern University

September 28, 1998

Table of Contents

Figures and Tables

Abstract

This proposal studies the design of a cobot which has the ability to store energy. Energy can be stored by allowing the cobot to have a 'second' degree of freedom (dof). The 'second' degree of freedom is de-coupled from the user and free to move. Therefore, the user is still only able to move with zero or one dof at any time. Using the *free* dof to store energy allows the cobot to simulate haptic effects in addition to virtual walls. These include simulations of inertial, assistance/resistance, and gyroscopic effects. I refer to the haptic effects enabled by the stored energy as virtual behaviors. How to efficiently store energy and transmit it back to the user are the core contribution of my research.

The design and development of a Joystick cobot with the capacity to store energy is discussed. The Joystick has two degrees of freedom with a three dimensional configuration space, (ϕ, γ, ψ). The degrees of freedom are precession and spin. The first degree of freedom describes the motion of the handle, and consequently the user. This motion is analogous to precession. The second, *free*, dof is described by the ball's spin about the handle.

1. Review of Previous Work in Passive Programmable Constraint Machines

1.1 Introduction

This chapter surveys all previous work done in cobotics and some of the most relevant work in passive devices with dynamic constraints. This includes work by Colgate et alii [1,2], Peshkin et alii [3], Wannasuphoprasit et alii [4], Moore et alii [5], Mills et alii [6], Russo et alii [7], and Troccaz et alii [8].

The conventional approach to robotics is usually an active one, involving actuators to move the robot and generate forces when involved in a task (e.g. assembly line, haptic applications, etc.). The power of the actuators necessary for these industrial applications, and/or haptic tasks prevent robots from working in close proximity to humans. This is because conventional approaches to haptic display employ servo control to reduce the degrees of freedom. In order to implement a rigid constraint, it has been found that servo controller require high gains. While significant progress has been made in the design of haptic interfaces which admit high gain controllers (Colgate et alii [9]), stability is usually not compatible with a high gain system. In addition to the inherent problem of high gain servo controllers, hardware and software failures have a real potential to cause harm. For these reasons, robots with haptic capabilities, which are intrinsically passive have great appeal. Cobots meet this requirement. Cobots do not use servos to implement constraints, but instead employ non-holonomic transmissions (which are intrinsically passive). As a consequence of their passivity, cobots are well-suited to safety-critical tasks.

The following sections briefly describe some examples of other approaches to passive dynamic constraint devices and previous work done in cobotics.

1.2 Previous Approaches to Passive Devices with Virtual Constraints

Surveying passive constraint machines immediately brings me to the works of Russo and Tadros [7], and Delnondedieu and Troccaz [8]. Their respective approaches to implementing constraints involve magnetic particle brakes and clutches.

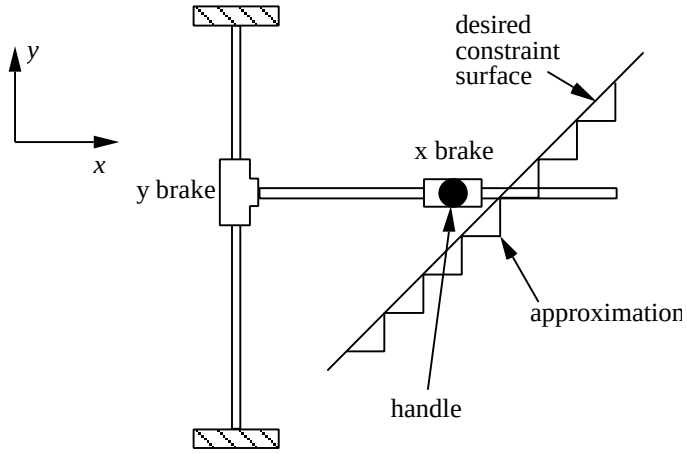


Figure 1-1. A passive haptic device using brakes [2].

Brakes are completely passive and can simulate hard constraints very well. Nevertheless, they are deficient in that they can only implement constraints which are perpendicular to their direction of motion. For example, implementing a virtual wall in the x or y direction where the user wants to be able to glide freely but not penetrate it, a brake mechanism would work wonderfully. For instance, in the case of a wall in the *y* direction the controller actuates the *x*-brake. However, if the user wanted to implement a diagonal wall the brake mechanism will in fact stick to the wall and not allow him/her to glide on such a surface. It follows then that in order to prevent penetration of the wall both brakes must be applied when on the wall. The alternatives are to either attach a brake at all desired wall directions, not very practical, or to simulate the wall with a series of steps, see Figure 1-1. This is done by repeatedly alternating which brake is activated, creating a bothersome and sticky feel.

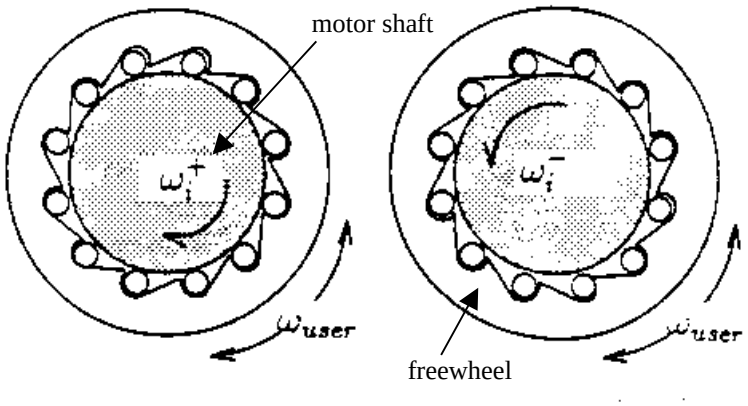


Figure 1-2. PADyC's clutching mechanism. (Drawing taken from [8]).

A different approach with very similar problems is PADyC, proposed by Troccaz [8]. Here a system of clutches is used to control the velocity of possible motions that are available. It works by replacing the classical revolute or prismatic actuated joint with a system that allows for four possible motion states:

1. motion in both positive and negative directions,
2. only in the positive direction,
3. only in the negative direction, or;
4. in any direction.

These movements are accomplished by instrumenting each joint with two freewheels which allow motion in opposite directions and with a mechanical system that allows clutching of each freewheel independently. The motions are then obtained by coordinating the two clutch systems associated with the two freewheels. By linking a motor to each freewheel in a way that it rotates in the direction that does not propel the joint (and in that direction only) they play the role of brakes since the speed in one direction is limited by the motor velocity. Depending on the relative velocity between the user's desired velocity and the motor the resulting motion of the mechanism is blocked or not. In effect the user is bound by the motor's velocity. For instance, if the user tries to move with a faster velocity than the motor the clutch is engaged disallowing any motion in that direction. Also if the motor is not moving, the freewheel is clutched and no motion is allowed. Thus by alternating between states each joint portrays one or zero degrees of freedom.

The main difficulty with this design arises when implementing what they call 'trajectory' mode, essentially when trying to follow a trajectory or path. With this clutch system the control of the joints consists of calculating the speed of the motors (which bounds the users velocity) at each joint in real time. This calculation is in turn directly dependent on the distance the user is away from the constraint. Therefore when trying to *glide* 'on' a path the computed velocity for the motors results in zero, unless the user moves off the path and come back to it much like the braking mechanism described above. This motion again results in a sticky, non-continuous feel. For this reason trajectory mode has been replaced with a 'corridor' mode. In this mode the trajectory is enclosed by a thin corridor which allows for smoother motion but still far from a gliding feel. Another effect of the motor velocity dependance on distance to constraint is that this mechanism starts slowing you down as you approach the constraint.

1.3 Conceptual Designs of Cobots

Colgate and Peshkin [1] have proposed a novel approach to passive constraint machines in order to better implement virtual surfaces. A *virtual surface* is a computer simulated physical surface. In practice, and in both of the above mentioned devices, surfaces are implemented by reducing the degrees of freedom of a mechanical system. In cobotics the opposite approach is taken. Here feedback control is used to increase the degrees of freedom of a one or zero degree of freedom (dof) device. In general a cobot has one dof less than the dimensions of its configuration space. This allows for two modes of operation: 1) free or otherwise called *caster mode*, and 2) *path-tracking*. In caster mode the device behaves as if it has all the dof available, while in path-tracking mode it loses some of those dof's and is forced to follow a specified path. Virtual walls are implemented by alternating between both modes of operation in an application.

The key to implementing such a device is the use of non-holonomic transmissions. By using non-holonomic transmissions Colgate and Peshkin effectively achieve smooth, hard, passive constraints. The following sections briefly describe the previous work done in cobotics.

1.3.1 Non-holonomic Transmissions

Before continuing with a description of the two existing cobots it is appropriate to define and give some examples of non-holonomic transmissions. Nonholonomic transmissions are the tools that make cobots possible. If they are not properly understood then the concept of cobots will not be properly grasped.

It is appropriate to describe a holonomic transmission before describing a non-holonomic one. A holonomic transmission is one which couples both velocity and displacements. For instance, a gear transmission couples the velocity of the input shaft to the velocity of the output shaft with a constant gear ratio. The displacements of input and output shaft are equally coupled. At any moment in time the controller can integrate the velocity of the output shaft and arrive at its position.

In contrast, a non-holonomic transmission is one which couples velocity without coupling position. Position is not coupled since the velocity ratio is variable over time. A consequence of this de-coupling of position is that the displacement components of the transmission can reach a higher-dimensional configuration space than its number of dof's. Nonholonomic transmissions therefore have fewer degrees of freedom than generalized coordinates – or coordinates necessary to describe its position. For instance, while having one dof a non-holonomic transmission moves in a second order configuration space. A good way to think about dofs and configuration space is that dofs relate to velocities, while C-space relates to displacements. A continuously variable transmission is

an example of a non-holonomic transmission, in fact all cvts are non-holonomic (they couple two velocities with an adjustable ratio).

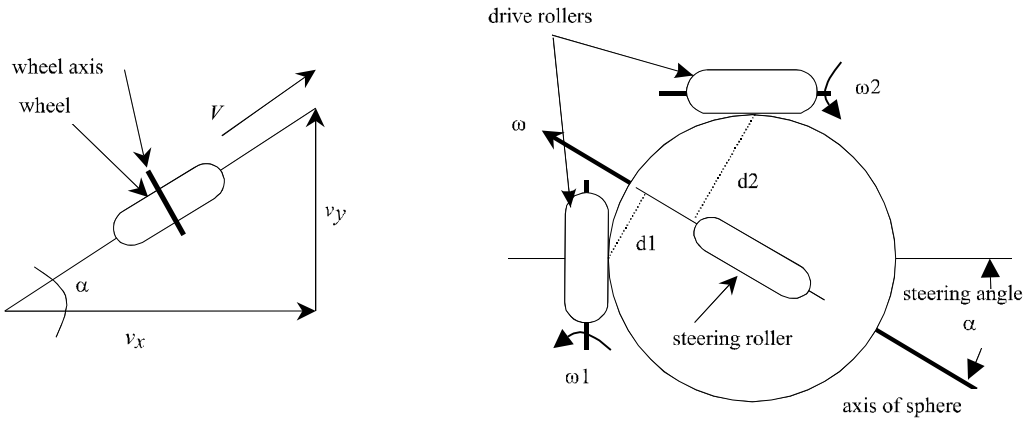


Figure 1-3. (left) Illustration of how a wheel (translational cvt) couples two velocities with an adjustable transmission ratio. (right) A rotational cvt, analogous to the translational cvt. The angular velocities of the two drive rollers are held in the proportion $\frac{\omega_2}{\omega_1} = \tan(\alpha)$ by the steering roller.

Two examples of continuously variable transmissions are: 1) a translational cvt, and 2) a rotational cvt. The simplest translational cvt is a rolling wheel. The wheel is a device which couples two translational velocities, v_x and v_y, into a resulting velocity, V, with an adjustable transmission ratio given $\frac{v_y}{v_x} = \tan(\alpha)$; α is in effect the steered angle of the wheel. Therefore, kinematically the rolling wheel has only one dof, V. A second cvt is seen in Figure 1-3. This cvt can be thought of as the rotational analog to the wheel. I refer to this device as the RCVT.

The RCVT couples two drive rollers, with angular velocity w_1 and w_2, such that $\frac{w_2}{w_1} = f(\alpha)$ where $f(\alpha)$ is the adjustable transmission ratio. These rollers are the input and output shafts responsible for the transmission of power. The remaining roller is the steering roller, this roller is analogous to the wheel (of the translational cvt) in that its orientation controls the transmission ratio. The axis of the steering roller, unlike the drive rollers is adjustable in the direction of α. The remainder of the hardware necessary for keeping the RCVT from falling apart is not shown, since only a theoretical description is necessary in defining the RCVT as a non-holonomic transmission.

The kinematics of the RCVT is described as follows. The sphere must be in rolling contact with all three rollers if it is to move. Since the sphere is not allowed to translate, the sphere's axis of rotation must pass through its center. In order to have a roller and a sphere in rolling contact it is required that both axes of rotation lie on the same plane. This plane must also pass through the center of the sphere. All three rollers must form such a plane with the sphere. The axis of rotation of the sphere is expressed as the line that results from the intersection of the three planes formed. If the planes do not intersect in a single line the RCVT cannot rotate.

1.3.2 Unicycle

The Unicycle is a one dof cobot that moves in a two dimensional (x, y) configuration space. It uses a wheel as a non-holonomic transmission to couple v_x and v_y velocity components. The wheel rolls on a horizontal surface supported by a Cartesian rail system and is moved by holding a handle

mounted above the wheel, Figure 1-4 (right). The unicycle constrains motion in *x* and *y*, but it cannot constrain orientation.

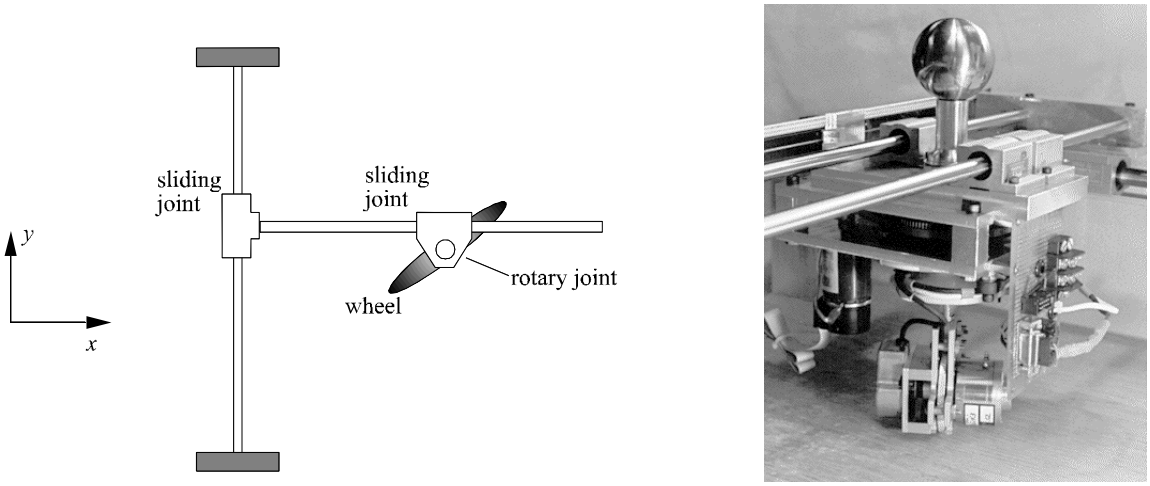


Figure 1-4. (left) Top view of the unicycle cobot [1], (right) picture of the unicycle prototype.

The Unicycle's design consists of a motor, a wheel, and a force sensor. The force sensor is the input to the system and the motor imposes the output, a direction of motion. Basically the force sensed is used to compute a desired direction of motion. The direction of motion is imposed (using the motor) by steering the wheel accordingly.

1.3.2.1 Unicycle Control [4]

The unicycle's controller is designed to allow it to exhibit two behaviors, virtual caster and path-tracking. In order for these behaviors to succeed the steering mechanism must be designed to, in the case of caster mode, null all lateral forces on the wheel and in the case of path-tracking to steer the wheel so that its rolling direction is tangential to the constraint. Path-tracking mode also requires that the absolute location and orientation of the wheel is known at all times since the steering angle is dependent on its position on the path.

The desired behavior of the unicycle is one of a point mass. For a point mass, the acceleration and force vectors are collinear and in fixed proportion. This in effect is the basis of the caster controller for the unicycle since it implies that not only must forces in the wheel direction $F_{\parallel}$, produce accelerations of $a_{\parallel} = \frac{F_{\parallel}}{M}$, but also that forces which are normal to the wheel must produce accelerations of $a_{\perp} = \frac{F_{\perp}}{M}$. However, the instantaneous normal acceleration of a wheel traveling at a speed, u (u is established by the user and cannot be controlled), with a steering velocity ω_s is $a_{\perp} = u\,\omega_s$. A virtual caster controller that mimics a point mass is given by

$$\omega_s = \frac{F_{\perp}}{uM_c} \tag{1}$$

Where M_c is the mass of the point-mass to mimic. Notice that the unicycle has a singularity when the measured speed, *u*, is zero. Details of how the unicycle caster controller was implemented are in [2].

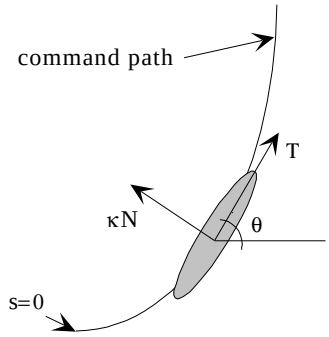


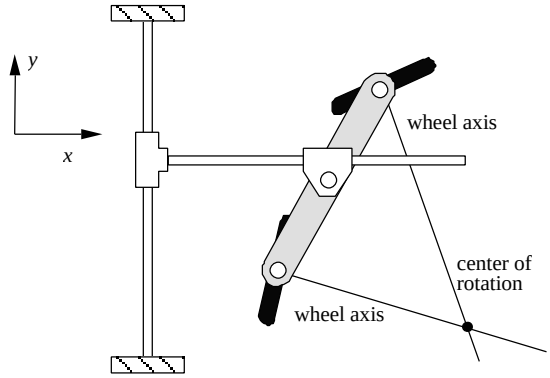
Figure 1-5. Top view of the unicycle in path tracking mode [4].

A feed-forward controller was derived for the unicycle by assuming perfect path tracking. This assumption suggests that the unicycle is always tangent to the intended path, Figure 1-5. However, since cobots are passive devices which cannot control the speed of motion, $\left|\frac{d\theta}{dt}\right|$, it is beneficial to derive a controller which is in terms of path length. Such a controller would be

$$\omega_s = \frac{d\theta}{ds}\frac{ds}{dt} = \kappa\, u.$$

The curvature, κ, and the measured speed, *u*, along the path determine the necessary steering velocity for the unicycle. As expected, however, assuming *perfect* path tracking is not practical. For instance, wheel slip ensures that such a controller cannot keep the unicycle on the desired path. Therefore, a feedback controller was implemented. The details of the feedback controller are not discussed since they are not pertinent to this proposal, such details are described in Wannasuphoprasit [4].

1.3.3 Scooter



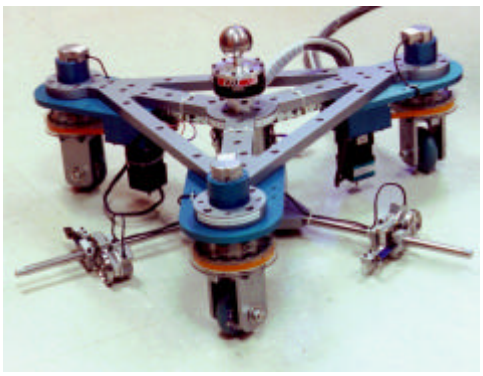


Figure 1-6. (left) Two wheel cobot concept drawing. Requires a support frame like the Unicycle and does not allow for the implementation of zero dof's. (right) Scooter three wheel cobot.

Scooter is a three wheel cobot with a three dimensional configuration space. It translates in *x* and *y* and rotate about *z* an angle *q*. This three wheel design allows the device to support itself without

the need of a frame. Another advantage of a three wheel cobot is that it resolves a singularity inherent to a two wheel cobot. A two wheel device has a singular configuration when both wheels are positioned parallel to each and their axes of rotation align. At this singular configuration the device has two dof's, it can translate in the direction of the wheels and rotate about any point on the line that contains the axes of both wheel. A third advantage of scooter's design is that it has the ability to exhibit zero dof's without the use of brakes. This benefit is accomplished by aligning all three wheels in such a way that their axes do not intersect in a single center of rotation.

The one dof kinematics of scooter can be described as a motion on a planar surface about an instantaneous center of rotation. By steering the wheels independently the center of rotation is placed anywhere in its workspace or they can be steered so that there is no instantaneous center of rotation. This enables scooter to exhibit one or zero dof's at any time in a three dimensional configuration space.

1.3.3.1 Scooter Control

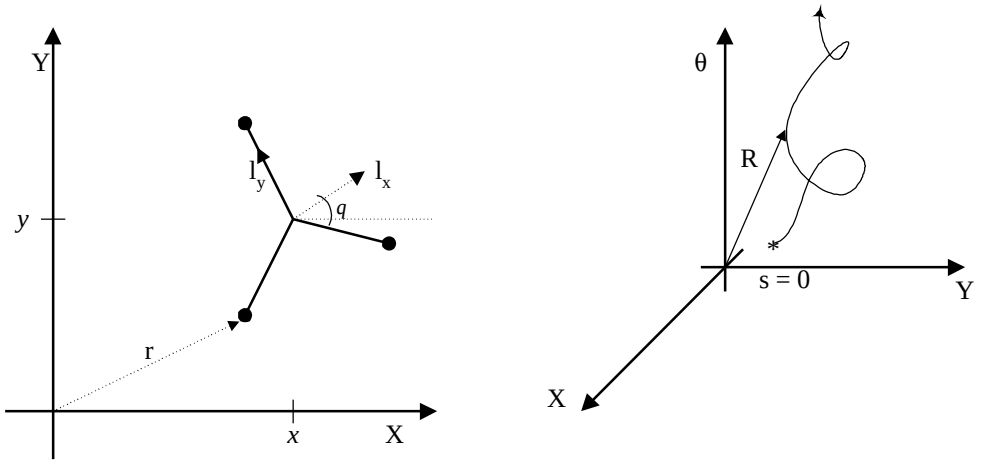


Figure 1-7. (left) Illustration of scooter's kinematics, (right) Scooter's C-space.

The controller for scooter was designed with the assumption that when controlling cobots, it is desirable to plan paths and constraint surfaces in configuration space (C-space), rather than in joint space. There are essentially four steps in deriving such a controller.

First, a kinematic relationship must be expressed between the path in configuration space and the path followed by a wheel. Scooter's C-space may be described as the position and orientation of the handle measured with respect to some global frame, or the vector $R = [x, y, \theta]^T$. The path followed by a wheel is thus given by

$$r_i = L_i(R) = \begin{bmatrix} \cos(q) & -\sin(q) & 0 & x \\ \sin(q) & \cos(q) & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} l_i \qquad i = 1, 2, 3 \qquad (2)$$

L(R) is a function which involves a single translation and rotation, where $l_i = [l_{xi}, l_{yi}, 0, 1]^T$ specifies the coordinates of wheel *i* in the local frame of scooter, and $r_i = [r_{xi}, r_{yi}, 0, 1]^T$ specify the path of wheel *i* in the global frame, Figure 1-7.

Secondly, since the rolling direction of the wheel must be tangent to the desired path a *tangent transformation* from the unit vector tangent to the C-space path, to the unit vector tangent to the wheel path must be derived. Therefore, if

$$\mathbf{T} = \frac{dR}{ds} \tag{3}$$

and

$$\mathbf{t}_i = \frac{dr_i}{ds_i} \qquad i = 1, 2, 3 \tag{4}$$

and using equation 2, **t** is expressed as

$$\mathbf{t}_i = \frac{J_i\mathbf{T}}{|J_i\mathbf{T}|} \qquad i = 1, 2, 3 \tag{5}$$

where J is the Jacobian from the cobot's C-space to the two dimensional path space of wheel *i*. See [4] for derivations.

The third step is then to relate the C-space curvature, κ, and unit normal, **N**, to those of the joint space path (κ_i, $\mathbf{n}_i$). Thus, by definition

$$\kappa\mathbf{N} = \frac{d\mathbf{T}}{ds}$$

$$\kappa_i\mathbf{n}_i = \frac{d\mathbf{t}_i}{ds_i} \qquad i = 1, 2, 3 \tag{6}$$

or

$$\kappa_i\mathbf{n}_i = \frac{I - \mathbf{t}_i\mathbf{t}_i^T}{|J_i\mathbf{T}|^2}\left[\begin{bmatrix}\mathbf{T}^T H_{ix}\mathbf{T}\\ \mathbf{T}^T H_{iy}\mathbf{T}\end{bmatrix} + J_i\kappa\mathbf{N}\right] \qquad i = 1, 2, 3 \tag{7}$$

where $H_{ix} = \frac{\partial^2 L_{ix}}{\partial R^2}$ and $H_{iy} = \frac{\partial^2 L_{iy}}{\partial R^2}$ are the Hessian matrices which represent the spacial rate of change of the Jacobian.

In order to control scooter with a caster like behavior all that is required is a desired acceleration. This desired acceleration **A** or direction of travel is calculated from the operator applied forces measured at the handle. The idea is to use **A** in order to predict a C-space path, which using equation 7 produces the joint space curvatures necessary for calculating the steering velocities. The equation that relates the desired acceleration to the C-space curvature is given by

$$\kappa N = \frac{1}{u^2}[I - \mathbf{T}\mathbf{T}^T]A \tag{8}$$

see [4] for derivation. This equation implies that the instantaneous velocity of scooter must also be measured, both magnitude *u* and direction **T**. Therefore the steering velocity command for each wheel is a product of a curvature and the speed at the joint, u_i. Then with

$$\mathbf{t}_i u_i = J_i\mathbf{T}u \qquad i = 1, 2, 3 \tag{9}$$

and since

$$\omega_i = | \mathbf{t}_i u_i \times \kappa_i \mathbf{n}_i |$$

the steering velocity for an individual wheel, given a desired acceleration is determined by

$$\omega_i = \left| J_i \mathbf{T} u \times \frac{I - t_i t_i^T}{|J_i T|^2} \left[\begin{bmatrix} T^T H_{ix} T \\ T^T H_{iy} T \end{bmatrix} + J_i \frac{1}{u^2} [I - TT^T] A \right] \right| \qquad i = 1, 2, 3 \tag{10}$$

A feedforward controller is then developed on the basis of, again, ideal path tracking. For this case then $\mathbf{T}$, and $\kappa\mathbf{N}$ are simply replaced with the command path's $\mathbf{T}_0$, and $\kappa_0\mathbf{N}_0$.

2. Design and Development of the Joystick

2.1 Contribution

This chapter describes the design of a cobot with zero, one or two degree of freedom at any time. The second dof is de-coupled from the user and *free* to move. Similar to previous cobots the user is still only able to move with zero or one dof at any time. The *free* dof is used to store energy and consequently to re-direct it back to the user. This energy in effect allows the cobot to simulate haptic effects in addition to virtual walls. These include simulations of inertial, assistance/resistance, and gyroscopic effects. How to efficiently store energy and simulate virtual behaviors is the core contribution of my research.

Up to now cobots have been limited to guiding tasks. Implementing an energy mode, therefore, is an important contribution to cobotics because it will permit cobots to participate in a larger range of applications and consequently become a more 'complete' haptic device. These applications include video games, exercise machines, etc.

2.2 Introduction

The cobots described in the previous chapter control planar motion, in a two or three dimensional C-space, by coupling planar degrees of freedom using wheels. Instead of planar motion this design considers spherical motion about a fixed center, (f, g, y); rotations about x, y, and z respectively.

The cobot shown in Figure 2-1 is a three axis haptic interface. I refer to this device as the *Joystick*. The three dof's of the sphere are again reduced to zero, one or two through the use of steering wheels. In contrast to Scooter, the Joystick has the ability to spin about its handle without imparting any energy to the user. Thus, aside from a bearing between the handle and the rest of the device the Joystick possesses kinematics analogous to those of scooter, they are described below.

2.3 Conceptual and Mechanical Design

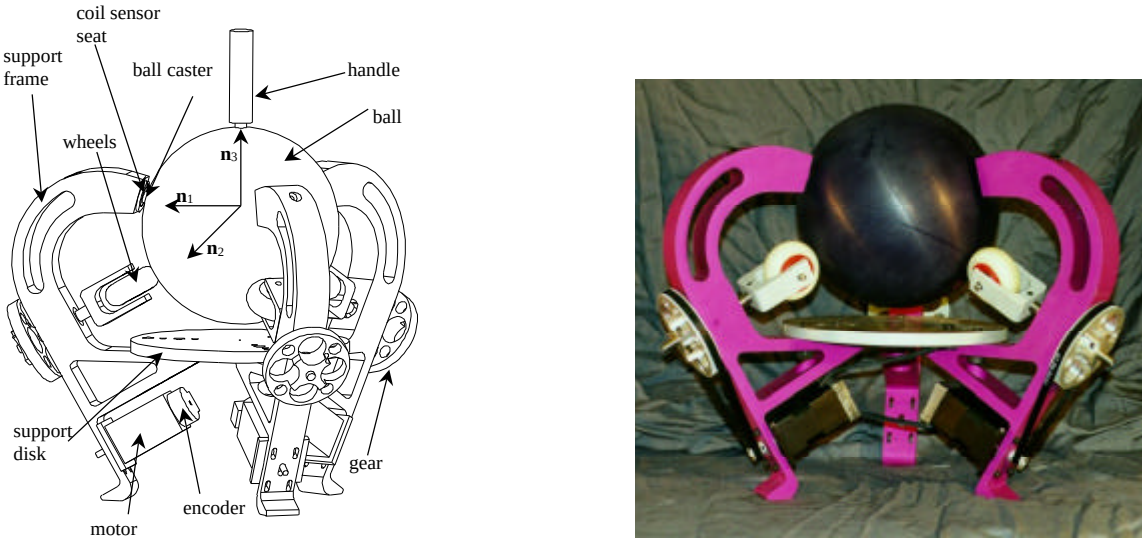


Figure 2-1. (left) Concept drawing. (right) Manufactured prototype.

The Joystick is a three wheel cobot, similar to scooter. The handle has one or zero degrees of freedom at any moment in time and moves in a three dimensional configuration space; (f, g y) rotation about a set of mutually perpendicular vectors, $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$. $\mathbf{N}$ defines a Newtonian reference. Its design consists of three steerable wheels attached to motors via geared transmissions; encoders are mounted piggyback on the motors. The wheels are supported by an aluminum frame and are positioned at 30° from horizontal. The frames are held together by a support disk and are set at 120° from each other. The wheels 'carry' a ball with a handle for maneuvering, Figure 2-1. Also, a prerequisite for motion of the ball is that the wheel's steering axis passes through the center of the ball.

Three ball casters are necessary in order to keep the ball from losing contact with the wheels in the event of sudden changes in motion. These are located 15° above the equator of the ball, Figure 2-1. Located around each ball caster is an inductor-coil. These coils together with a fourth coil which is located inside the ball are used to measure the position and velocity of the axis of the handle. The coil sensor is described further in the next section. Finally, a force sensor is mounted on the handle in order to determine the user's desired direction of motion.

The design of the Joystick is rather analogous to that of scooter. It has a three dimensional C-space and it uses three wheels in order to control motion. There is, however, one subtle but interesting difference. In the case of the Joystick the third dof, rotation about the handle, is de-coupled from the user. In other words even though the Joystick moves in a three dimensional C-space the handle only rotates about two of them. For instance, with scooter the handle translates in x and y and rotate about z but with the Joystick the handle can only rotate about x and y. The ball, however, is allowed to spin about its handle. This is done by mounting the handle on bearings which reside inside the ball.

2.4 Coil Sensor Design and Implementation

In order to determine the absolute location and velocity of the handle a special purpose sensor was designed. It is based on the electrical principle of induction.

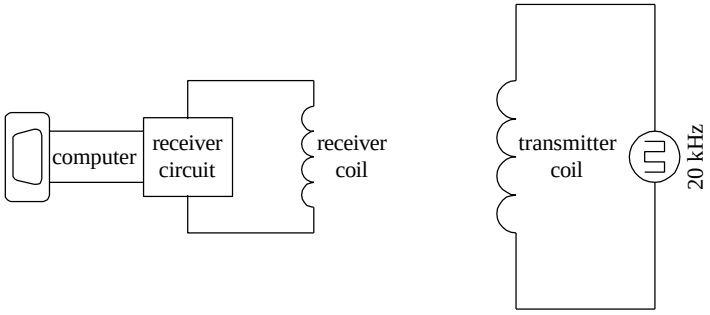


Figure 2-2. Induction sensor used to determine the orientation of the axis of rotation of the Joystick.

The sensor consists of a transmitter and a receiver circuits. The transmitter is positioned in the center of the ball and transmits a known signal, see Figure 2-2. Three receivers are then placed at the coil-sensor seat on each frame, Figure 2-1. The coil-sensor seats are equidistant to the center of the ball. The transmitted signal then induces a current in the receiver coil. This signal is dependent on the position and orientation of the receiver with respect to the transmitter coil. By placing the transmitter coil in the center of the ball I assume a constant distance (*d*) to the receiver coils, and as a result lump the $\frac{1}{d^3}$ factor in a constant. This avoids having to measure *d*; the $\frac{1}{d^3}$ factor is due to **B** $= \frac{\mu_0 I r^2}{2(r^2 + d^2)^{3/2}}$, the magnetic field measured at a distance *d* on the axis of the transmitter coil, where *r* is the radius of the coil.

The orientation of the handle is immediately calculated from the induced current. Only two receiver coils are required for this, however, three were used for symmetry and redundancy. A redundant configuration is necessary to account for the singular case when the ball is spinning about an axis perpendicular to one of the receiver coils; or when the transmitter coil and a receiver coil are perpendicular to each other. No current can be induced when the coils are perpendicular to each other.

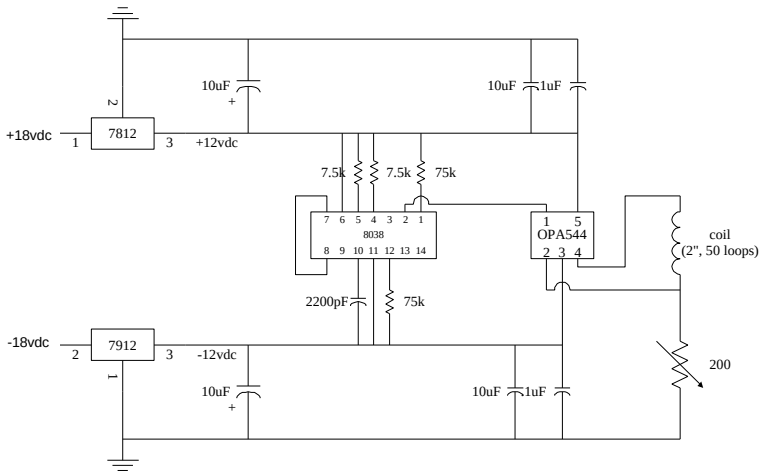


Figure 2-3. Transmitter circuit which transmits a 20kHz signal.

The key components of the transmitter circuit are an ICL8038 Precision wave form generator, a OPA544 high-voltage/high-current operational amplifier, a positive and a negative regulator, and a coil. These are used to generate a strong 20kHz signal at the transmitter coil. A variable resistor is connected in series with the coil in order to control the signal strength; the amount of current through the coil. Figure 2-3, shows a schematic of the transmitter circuit used.

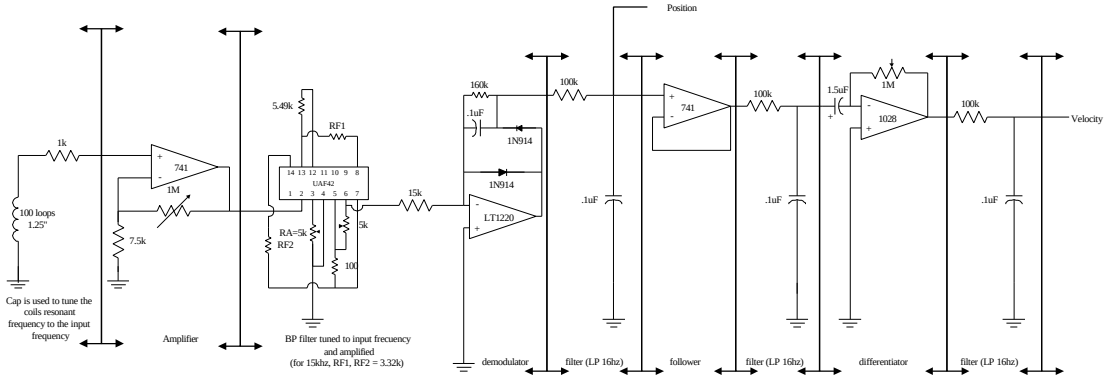


Figure 2-4. Receiver circuit.

The receiver circuit on the other hand consists of three sub-circuits, see Figure 2-4. First there is a receiver coil tuned to the resonant frequency of the transmitter coil (20kHz) and an amplifier. The second sub-circuit filters the signal using a UAF42 (universal active filter), demodulates it, and subsequently filters it again. The signal measured at this point corresponds to the relative position of that particular pick-up coil and the transmitter coil. Specifically, the voltage measured, once normalized, relates the two coils by the cosine of the angle between them; $\frac{V(oltage)}{V_{\|}(max)} = \cos(\theta)$.

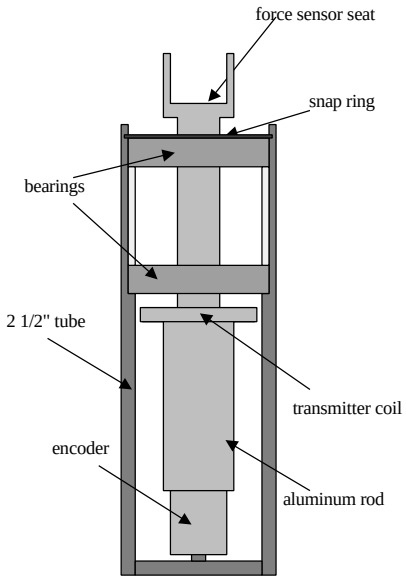


Figure 2-5. Drawing of the instrumentation located inside the ball.

Following the position sensor a differentiator circuit is wired in order to implement a velocity sensor, dV/dt (V as in $V = \cos(\theta)$). The signal measured at this level determines the rate of change of the position of a pick-up coil with respect to the transmitter coil. This information is used to determined the velocity of the handle. The signals are read by the computer via an A/D card.

Note that the spin of the ball cannot be measured at this point since the coils are not sensitive to motion in that direction. To measure the velocity of the spin of the ball an encoder is placed inside the ball, see Figure 2-5. The math involved in calculating the position and velocity of the handle axis follows:

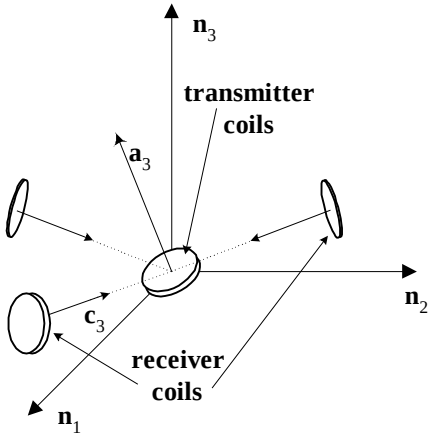


Figure 2-6. Receiver and transmitter coil configuration.

Once the signals from the receiver coils are read by the computer they are used to calculate the position of the axis of the handle of the ball using the following equations.

Representing the orientation of the transmitter with unit vector $\mathbf{a}_3$ and the receivers' with $\mathbf{c}_{3i}$ then it follows that

$$\mathbf{a}_3 = a_1\,\mathbf{n}_1 + a_2\,\mathbf{n}_2 + a_3\,\mathbf{n}_3 \tag{11}$$

$$\mathbf{c}_{3i} = c_{1i}\,\mathbf{n}_1 + c_{2i}\,\mathbf{n}_2 + c_{3i}\,\mathbf{n}_3 \qquad i = 1, 2, 3 \tag{12}$$

and V_{ri} and $V_{\|i}$ are the voltage received and the maximum possible voltage received by that sensor (this occurs when the coils are parallel to each other), respectively, thus

$$\frac{V_{ri}}{V_{\|i}} = \cos(q) = \left(\mathbf{c}_{3i} \bullet \mathbf{a}_3\right) \qquad i = 1, 2, 3 \tag{13}$$

written in matrix form

$$\begin{bmatrix} \frac{V_{r1}}{V_{\|3}} \\ \frac{V_{r2}}{V_{\|3}} \\ \frac{V_{r3}}{V_{\|3}} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

or

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}^{-1} \begin{bmatrix} \frac{V_{r1}}{V_{\|3}} \\ \frac{V_{r2}}{V_{\|3}} \\ \frac{V_{r3}}{V_{\|3}} \end{bmatrix} \tag{14}$$

to solve for the **N** measure numbers of $\mathbf{a}_3$. Keep in mind that these equations assume the distance from the transmitter sensor to the receiver sensors is constant.

The second piece of information available from these sensors is the velocity of the $\mathbf{a}_3$. Differentiating equation 13

$$\frac{1}{V_{\|}}\frac{dV_r}{dt} = \frac{{}^N d\mathbf{c}_{3i}}{dt} \bullet \mathbf{a}_3 + \mathbf{c}_{3i} \bullet \frac{{}^N d\mathbf{a}_3}{dt} \qquad i = 1, 2, 3 \tag{15}$$

and given that $\mathbf{c}_3$ is fixed in **N** reduces to

$$\frac{1}{V_{\|}}\frac{dV_r}{dt} = \mathbf{c}_{3i} \bullet \frac{{}^N d\mathbf{a}_3}{dt} \qquad i = 1, 2, 3 \tag{16}$$

$$\mathbf{v} = \frac{{}^N d\mathbf{a}_3}{dt} = \mathbf{a}' = a'_1\,\mathbf{n}_1 + a'_2\,\mathbf{n}_2 + a'_3\,\mathbf{n}_3 \tag{17}$$

and solving for the **N** measure numbers of **v** yields

$$\begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{V_{\parallel}} \frac{dV_{r1}}{dt} \\ \frac{1}{V_{\parallel}} \frac{dV_{r2}}{dt} \\ \frac{1}{V_{\parallel}} \frac{dV_{r3}}{dt} \end{bmatrix} \tag{18}$$

Recall that V_r and $\frac{dV_r}{dt}$ are the coil sensors' output.

2.5 Geometry

Before continuing with the kinematics of the Joystick a description of the geometry on which it is formulated is appropriate. This section essentially describes the frames of reference and vectors that are used throughout the proposal.

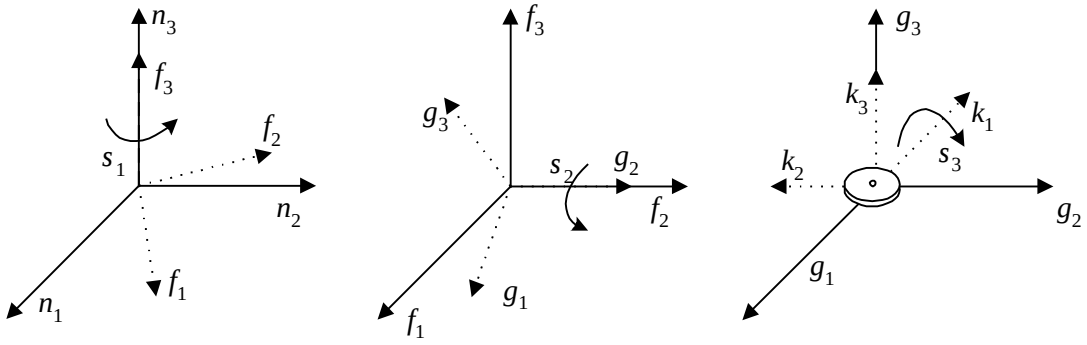


Figure 2-7. Coordinate frames. k_3 defines the axis of rotation of the wheel, k_1 defines the axis of the wheel shaft and points towards the center of the ball and k_2 defines the axis normal to both k_1 and k_2.

The geometry of the Joystick is best described using reference frames. Starting from the Newtonian frame previously established, **N**, four rotations describe the orientation of a wheel. Figure 2-7 shows the four such frames, also refer to Figure 2-1 to see the placement of the **N** frame on the Joystick ($\mathbf{n}_1$ points towards the first support frame).

The three reference frames are **f,** which is rotated about $\mathbf{n}_3$ by some angle σ_1, **g** which is rotated about $\mathbf{f}_2$ some angle σ_2 and **k**, the frame of the wheel, rotates an angle (steering angle) θ_i about $\mathbf{k}_1$; the wheel rolls about $\mathbf{k}_3$. The following equations completely express the orientation of a wheel in **N**:

Wheel (i)	Angle σ_1	Angle σ_2	Angle θ_i (steering angle)
wheel 1	0°	30°	θ_1
wheel 2	120°	30°	θ_2
wheel 3	-120°	30°	θ_3

Table 1. Specific values of s for each wheel.

$$\mathbf{k}_{1i} \equiv -\cos(\sigma_{1i})\cos(\sigma_{2i})\,\mathbf{n}_1 - \sin(\sigma_{1i})\cos(\sigma_{2i})\,\mathbf{n}_2 + \sin(\sigma_{2i})\,\mathbf{n}_3 \qquad i = 1, 2, 3 \tag{19}$$

$$\mathbf{k}_{2i} \equiv [\cos(\sigma_{1i})\sin(\sigma_{2i})\sin(q_i) + \cos(q_i)\sin(\sigma_{1i})]\,\mathbf{n}_1 + [\sin(\sigma_{1i})\sin(\sigma_{2i})\sin(q_i) - \cos(\sigma_{1i})\cos(q_i)]\,\mathbf{n}_2 + \cos(\sigma_{2i})\sin(q_i)\,\mathbf{n}_3 \qquad i = 1, 2, 3 \tag{20}$$

$$\mathbf{k}_{3i} \equiv [\cos(\sigma_{1i})\cos(q_i)\sin(\sigma_{2i}) - \sin(\sigma_{1i})\sin(q_i)]\,\mathbf{n}_1 + [\cos(q_i)\sin(\sigma_{1i})\sin(\sigma_{2i}) + \cos(\sigma_{1i})\sin(q_i)]\,\mathbf{n}_2 + \cos(\sigma_{2i})\cos(q_i)\,\mathbf{n}_3 \qquad i = 1, 2, 3 \tag{21}$$

2.6 Kinematics

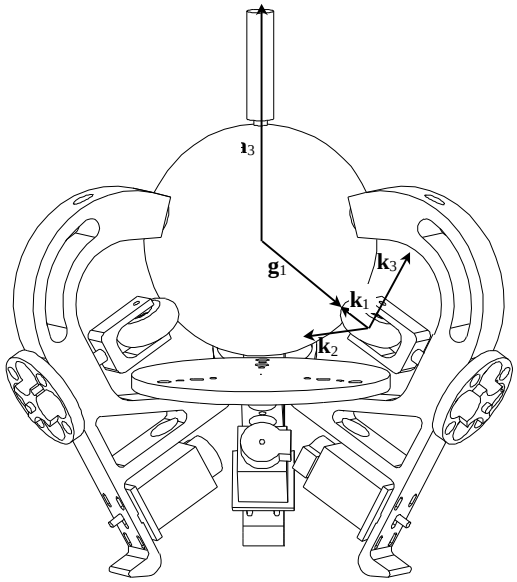


Figure 2-8. Illustration of the most pertinent vectors and frames which are used through out this proposal.

The motion of the Joystick can be described in terms of two simple rotations each performed with an angular velocity. The first component is the motion of the ball in the frame of the handle, and the second is the motion of the handle in a Newtonian frame, **N**. I refer to the angular velocity of the ball about the handle as *spin velocity* and to the angular velocity of the handle in **N** as *precession velocity*. The addition of these two angular velocities result in the angular velocity of the ball in **N** ($^N\mathbf{W}^S$, or simply $\mathbf{W}$) which I refer to as the angular velocity of the ball (or the Joystick) throughout this proposal. For purposes of notation, **N** is the Newtonian frame, **A** is the frame of the handle, and **S** is the frame of the ball.

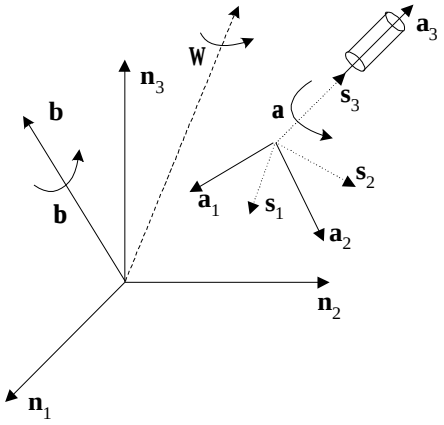


Figure 2-9. Representation of the angular velocity of the ball, $\mathbf{W}$. The spin velocity of the ball, a, rotation of frame **S** about $\mathbf{a}_3$, and the precession velocity b of frame **A** about vector **b** results in the angular velocity $\mathbf{W}$ of the ball in **N**.

The two components of the angular velocity of the ball, Figure 2-9: 1) *spin*, and 2) *precession* are expressed mathematically as

$$\textit{spin velocity} \equiv {}^{A}\mathbf{W}^{S} = a\ \mathbf{a}_3 \tag{22}$$

and (23)

$$\textit{precession velocity} \equiv {}^{N}\mathbf{W}^{A} = b\ \mathbf{b} \tag{24}$$

using the addition theorem for angular velocities

$${}^{N}\mathbf{W}^{S} = a\ \mathbf{a}_3 + b\ \mathbf{b} \tag{25}$$

Some interesting observations that stem from these equations are that when there is no precession velocity, ${}^{N}\mathbf{W}^{S}$ aligns with $\mathbf{a}_3$. On the other hand, when there is no spin velocity, ${}^{N}\mathbf{W}^{S}$ aligns with **b**. Therefore, as ${}^{N}\mathbf{W}^{S}$ approaches **b** the spin speed of the ball decreases and inversely as ${}^{N}\mathbf{W}^{S}$ approaches $\mathbf{a}_3$ precession speed decreases. Also the axis of ${}^{N}\mathbf{W}^{S}$ must lie on the plane that includes $\mathbf{a}_3$ and **b**. These equations also serve to show how the ball can be wound up (have it start spinning). For instance, by initially aligning ${}^{N}\mathbf{W}^{S}$ with **b** and *precessing* the ball, or moving the handle in **N,** while steering the wheels so that ${}^{N}\mathbf{W}^{S}$ moves away from **b** the user will immediately start the ball spinning about the handle.

The angular velocity of the ball is calculated using an encoder which is placed inside the ball and the coil sensors described above. Recall that the encoder measures a and the coil sensors measure ($\mathbf{a}_3$ and $\frac{{}^{N}\mathrm{d}\mathbf{a}_3}{\mathrm{dt}}$). The following equations are used to solve for the angular velocity of the ball in **N,** ${}^{N}\mathbf{W}^{S}$.

$$\frac{{}^{N}\mathrm{d}\mathbf{a}_3}{\mathrm{dt}} = {}^{N}\mathbf{W}^{A} \times \mathbf{a}_3. \tag{26}$$

$${}^{N}\mathbf{W}^{S} = a\ \mathbf{a}_3 + {}^{N}\mathbf{W}^{A} \tag{27}$$

Since $\mathbf{a}_3$ and ${}^{N}\mathbf{W}^{A}$ are perpendicular to each other

$$\left\| \frac{{}^{N}d\mathbf{a}_3}{dt} \right\| = \|\mathbf{a}_3\| \, \|{}^{N}\mathbf{W}^{A}\| \tag{28}$$

Any motion of the ball can be expressed as a rotation about an axis of rotation, $\frac{{}^{N}\mathbf{W}^{S}}{\|{}^{N}\mathbf{W}^{S}\|}$. It is useful, then, to find an expression that relates, ${}^{N}\mathbf{W}^{S}$, to the steering wheel angles. This expression follows from the rolling constraint equation

$${}^{N}\mathbf{W}^{S} \times \mathrm{R}\, \mathbf{g}_{1i} = {}^{N}\mathbf{W}^{W} \times \mathrm{r}\, \mathbf{k}_{1i} \qquad i = 1, 2, 3 \tag{29}$$

Where, ${}^{N}\mathbf{W}^{W}$, is the angular velocity of a wheel expressed in **N**, R, is the radius of the ball and, r, is the radius of the wheel. Further, substituting

$${}^{N}\mathbf{W}^{W} = \dot{q}_{wi}\, \mathbf{k}_{3i} + \dot{q}\, \mathbf{k}_{1i}, \qquad i = 1, 2, 3 \tag{30}$$

where $\dot{q}_{wi}$ is the wheel's angular velocity about $\mathbf{k}_3$ and $\dot{q}$ is its steering velocity (about $\mathbf{k}_1$), in equation 29 I get

$${}^{N}\mathbf{W}^{S} \times \mathrm{R}\, \mathbf{g}_{1i} = \left(\dot{q}_{wi}\, \mathbf{k}_{3i} + \dot{q}\, \mathbf{k}_{1i} \right) \times \mathrm{r}\, \mathbf{k}_{1i} \qquad i = 1, 2, 3 \tag{31}$$

or

$${}^{N}\mathbf{W}^{S} \times \mathrm{R}\, \mathbf{g}_{1i} = \dot{q}_{wi}\, \mathrm{r}\, \mathbf{k}_{2i} \qquad i = 1, 2, 3 \tag{32}$$

In order to have an equation independent of $\dot{q}_{wi}$, which is unknown, I solve for the $\mathbf{k}_3$'s component of the angular velocity of the ball, hence,

$$\mathbf{k}_3 \cdot \left({}^{N}\mathbf{W}^{S} \times \mathrm{R}\, \mathbf{g}_{1i} \right) = 0 \qquad i = 1, 2, 3 \tag{33}$$

From this I get an expression for the steering angles of each wheel given the desired angular velocity of the ball,

$$w_i \equiv {}^{N}\mathbf{W}^{S} \cdot \mathbf{n}_i \qquad i = 1, 2, 3$$

$$q_1 = \operatorname{atan2}\left(\frac{2\, w_2}{w_1 + \sqrt{3} w_3} \right) \tag{34}$$

$$q_2 = \operatorname{atan2}\left(-\frac{2\left(\sqrt{3} w_1 + w_2\right)}{-w_1 + \sqrt{3}\left(w_2 + 2 w_3\right)} \right) \tag{35}$$

$$q_3 = \operatorname{atan2}\left(-\frac{2\left(\sqrt{3}\, w_1 - w_2\right)}{w_1 + \sqrt{3}\left(w_2 - 2 w_3\right)} \right) \tag{36}$$

Differentiating 34, 35, and 36 an expression for the steering velocity, $\dot{q}$, of the wheels in terms of the angular acceleration vector $\dot{\mathbf{W}}$ is derived,

$$\dot{q}_1 = \frac{-2\cos^2(q_1)\left(w_2\dot{w}_1 - (w_1 + \sqrt{3}\,w_3)\dot{w}_2 + \sqrt{3}\,w_2\dot{w}_3\right)}{(w_1 + \sqrt{3}w_3)^2} \tag{37}$$

$$\dot{q}_2 = \frac{\cos^2(q_2)\left[-4(2w_2 - 3w_3)\,\dot{w}_1 + 4(2w_1 - \sqrt{3}w_3)\,\dot{w}_2 + 4\,(3w_1 + \sqrt{3}w_2)\dot{w}_3\right]}{\left(-w_1 + \sqrt{3}(w_2 + 2w_3)\right)^2} \tag{38}$$

$$\dot{q}_3 = \frac{\cos^2(q_3)\left[-4(2w_2 - 3w_3)\,\dot{w}_1 + 4(2w_1 - \sqrt{3}w_3)\,\dot{w}_2 - 4\,(3w_1 - \sqrt{3}w_2)\dot{w}_3\right]}{\left(w_1 + \sqrt{3}(w_2 - 2w_3)\right)^2} \tag{39}$$

I use these equations together with the Joystick's equations of motion, to steer the wheels so as to follow any force applied at the handle. The equations of motion relate the forces applied at the handle to the angular acceleration, $\dot{w}$.

N	Newtonian frame.
$\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$	dextral set of mutually perpendicular unit vectors fixed in **N**.
A	Frame of the handle. (**A** precesses in **N**)
$\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$	dextral set of mutually perpendicular unit vectors fixed in **A**.
S	Frame of the ball. (**S** spins in **A**)
$\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$	dextral set of mutually perpendicular unit vectors fixed in **S**.
$\mathbf{a}_3$	axis of the handle
$\mathbf{k}_1$	axis of the wheel steering shaft
$\mathbf{k}_3$	wheel's rolling axis
${}^N\mathbf{W}^S$	angular velocity of the ball observed in Newtonian frame **N**.
${}^N\mathbf{W}^A$	angular velocity of the handle observed from the Newtonian frame **N**. (precession velocity)
${}^A\mathbf{W}^S$	angular velocity of the ball observed from the handle frame. (spin velocity)
θ_i	steering angle of wheel *i*.
$\dot{\theta}_i$	steering velocity of wheel *i*.
b_i	${}^N\mathbf{W}^A \cdot \mathbf{a}_i$ precession velocity expressed in **A**
a	${}^A\mathbf{W}^S \cdot \mathbf{a}_3$ spin velocity expressed in **A**
ω_i	${}^N\mathbf{W}^S \cdot \mathbf{n}_i$ angular velocity of the ball expressed in **N**.
$\dot{\omega}_i$	${}^N\dot{\Omega}^S \cdot \mathbf{n}_i$ angular acceleration of the ball expressed in **N**.
ϕ	Angle specifying a rotation about $\mathbf{n}_1$.
γ	Angle specifying a rotation about $\mathbf{n}_2$.
ψ	Angle specifying a rotation about $\mathbf{n}_3$.
AOR	Axis of Rotation.
dof	degree of freedom

Table 2. Table of definitions.

3. Proposed Research

3.1 Introduction

I propose to implement three control modes for the Joystick: 1) caster, 2) path-tracking, 3) and energy mode. C*aster mode* refers to the control mode which responds to the user's applied force by steering the wheels in a way so as to minimize those forces. In effect, caster mode permits any motion that the user attempts, both magnitude and direction. This mode of operation is analogous to moving a chair on casters. *Path-tracking mode*, on the other hand, is a bilateral constraint mode which restricts the direction of motion desired by the user to that of a predetermined path. I focus this discussion on the case of bilateral constraint since, as with scooter, a bilateral constraint is easily made unilateral with a software switch [1, 2]. Path-tracking is enforced by steering the wheels in a direction that is tangential to the path, or constraint surface. If a force tends to push the wheel into the constraint surface it is ignored. Both caster and path-tracking are modes that deal with the control of the Joystick when the ball is not spinning (i.e. the angular velocity of the ball has only one component ${}^{N}\mathbf{W}^{A}$). These control modes are analogous to scooter's and are implemented in a similar fashion (i.e. using curvature control).

The Joystick is in *energy* mode when the ball is spinning about the handle. In this mode, therefore, the ball has stored kinetic energy due to spin velocity. In contrast, Scooter does not have an energy mode of operation. In this mode the angular velocity of the ball has two components, ${}^{N}\mathbf{W}^{A}$ and ${}^{A}\mathbf{W}^{S}$. The kinetic energy stored can consequently be transmitted back to the user. How this energy is re-transmitted to the user is the core of my contribution to cobotics. This re-direction of energy, for instance, allow for the implementation of haptic effects that previous cobots could not simulate. Imagine, for example, that there is a motor attached to the ball in such a way so as to spin it at a constant angular velocity ${}^{A}\mathbf{W}^{S}$. As long as the axis of rotation (AOR) is coincident with the handle, the user will not move. However, as soon as the AOR is moved away from the axis of the handle, a component of the spin velocity goes into precessing the ball, in other words the handle starts moving in **N**.

3.2 Caster Control

Any motion of the ball can be described as a rotation about an axis-of-rotation. The AOR of the ball is specified by ${}^{N}\mathbf{W}^{S}$. However, in the case of both caster and path-tracking modes of operation ${}^{N}\mathbf{W}^{S}$ has only one component, ${}^{N}\mathbf{W}^{A}$, which is always perpendicular to the handle, or $\mathbf{a}_3$.

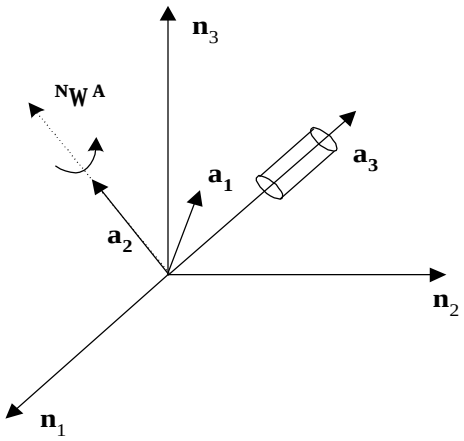


Figure 3-1. Representation of the angular velocity of the Joystick in passive mode.

The purpose of caster control is to achieve an *output behavior* at the handle equivalent to what you feel if you where handling a ball in space. For instance, if the user were to apply a force to the surface of a ball in space (i.e. which is supported at its center with a ball joint) the ball responds by accelerating in the direction of such force. This is the behavior of an ideal caster controller for the Joystick. This behavior can be attained by modeling the dynamics of such a free-moving system, hence

given that

$$b_i = {}^N\mathbf{W}^A \cdot \mathbf{a}_i \qquad i = 1, 2, 3 \tag{40}$$

$$M_i = \mathbf{M} \cdot \mathbf{a}_i \qquad i = 1, 2, 3 \tag{41}$$

$$\alpha = {}^A\mathbf{W}^S \cdot \mathbf{a}_3 \tag{42}$$

the equations of motion of the model are

$$\mathbf{M} = -F_y h\, \mathbf{a}_1 + F_x h\, \mathbf{a}_2 \tag{43}$$

$$I\dot{b}_1 = M_1 + (I - I_3) b_2 b_3 - J a b_2 \tag{44}$$

$$I\dot{b}_2 = M_2 + (I_3 - I) b_1 b_3 + J a b_1 \tag{45}$$

where h is the distance from the center of the ball to the handle, J is the axial moment of inertia of the ball and I (=I_1=I_2), I_3 are the central moments of inertia of the Joystick. Also with

$$\dot{w}_1\, \mathbf{n}_1 + \dot{w}_2\, \mathbf{n}_2 + \dot{w}_3\, \mathbf{n}_3 = (\dot{b}_1 + a b_2)\, \mathbf{a}_1 + (\dot{b}_2 - a b_1)\, \mathbf{a}_2 + \dot{a}\, \mathbf{a}_3 \tag{46}$$

and equations 37, 38, 39 I can determine the steering velocities that allow for the desired output behavior. Notice that since the ball starts from rest, $a = \dot{a} = 0$, and cannot accelerate (there is no moment applied about $\mathbf{a}_3$), the desired motion is given by

$$\dot{w}_1\, \mathbf{n}_1 + \dot{w}_2\, \mathbf{n}_2 + \dot{w}_3\, \mathbf{n}_3 = \dot{b}_1\, \mathbf{a}_1 + \dot{b}_2\, \mathbf{a}_2 \tag{47}$$

This result agrees with intuition since in caster mode the axis of rotation should lie on the plane perpendicular to the handle. It should not have an $\mathbf{a}_3$ component (it should not be spinning).

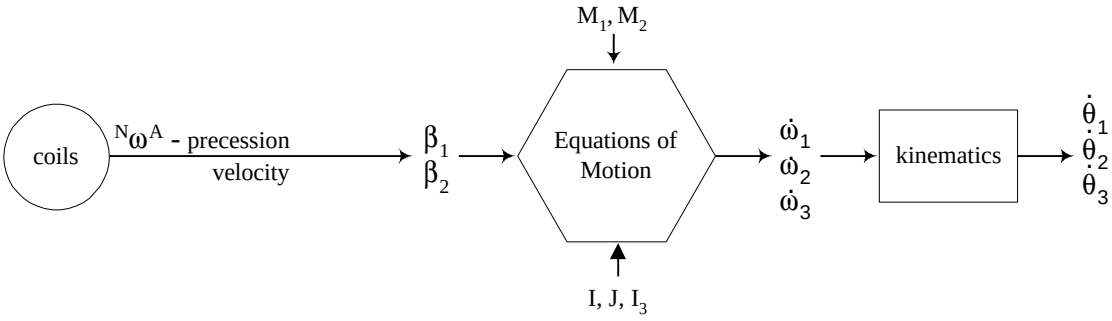


Figure 3-2. Open loop representation of passive caster control.

A second way that could be used to implement caster control is *curvature control*, as described in section §1.3.3.1. The equations presented in §1.3.3.1 can be immediately used to implement a caster controller for the Joystick by simply substituting in the Joystick kinematics. For example, C-space in the case of the Joystick in passive caster control simply includes two rotations, or R = [*f*, *g*]^T.

3.3 Path-Tracking Mode

I expect that the implementation of path-tracking mode will be very similar to the one described in §1.3.3.1 the only difference being the Joystick kinematics. Configuration space for the Joystick is described by two rotations, R = [*f*, *g*]^T. Thus,

$$r_i = L_i(R) = \mathrm{Rot}(\mathbf{n}_1, f)\,\mathrm{Rot}(\mathbf{n}_2, g)\,g_{1i} \qquad i = 1, 2, 3 \qquad (48)$$

where $g_{1i} = [l_{\mathbf{a}1i}, l_{\mathbf{a}2i}, l_{\mathbf{a}3i}, 1]$ and $r_i = [r_{\mathbf{n}1i}, r_{\mathbf{n}2i}, r_{\mathbf{n}3i}, 1]$ again describe the local position of wheel *i* and its path, respectively. g_{1i} are vectors fixed in the **S** frame (frame of the ball) which point towards the wheel and align with its steering shaft – this is true when the frame of the ball aligns with **N**. It is then obvious how *r* describes the path followed by the wheel in **N**. Other than substituting in these kinematic equations and expressing the desired acceleration as, $\mathbf{A} = [-F_y h/I, F_x h/I]$, the curvature controller described above will be implemented as was for scooter.

Combining caster and path-tracking control allows for the simulation of virtual walls. As a result, an application for the Joystick would be that of a haptic interface for video games. Other applications will be possible once energy mode is described.

3.4 Energy Mode

Unlike previous cobots, the Joystick can store energy even when the handle is not moving, and consequently re-apply that energy in interesting ways. Scooter and the Unicycle have been designed to enforce haptic effects of the 'wall' type. However, by allowing a Cobot to have a *free* degree of freedom for storing energy, it is conceivable that such energy can be used to simulate haptic effects in addition to virtual surfaces. To distinguish these new effects from those of a wall or virtual surface I refer to them as *virtual behaviors*.

To shed light on what kind of virtual behaviors can be simulated by an energy storing cobot I will describe a one dimensional analogous system. For this description, a device with a simpler configuration space is desired in order to focus attention on the control of virtual behaviors as opposed to path control (which has been the focus of cobots up to now).

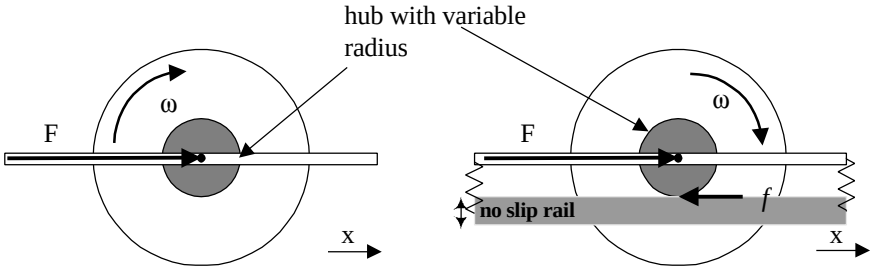


Figure 3-3. (left) Passive mode illustration of the wheel example. The user applies a force and the wheel moves in the track. (right) Illustration of the energy mode of a wheel device. By requiring the hub to be in non-slip contact with a rail the rotational velocity is directly coupled with the users translational velocity. This constraint allows for the kinetic energy, due to spin, to be redirected to the user.

A cobot which is capable of storing energy and has simpler kinematics can be constructed from a wheel. Such a device must have, as the Joystick, a passive and an energy mode. The Wheel is in passive mode when its axle is constrained to travel in a track and no rolling constraint is enforced, see Figure 3-3 (left). In this mode the user cannot impose a spin velocity and conversely the Wheel, if spinning through external means, cannot impose a velocity on the user. Energy mode requires, however, that the spin dof be coupled to the user. To enforce such coupling of velocities, a concentric hub in no-slip contact with a rail at all times (when in energy mode) is used. The hub has a variable radius (r) in order to control (vary) the ratio of spin velocity to translational velocity. The user interacts with the Wheel by pushing it by its axle.

The equation of motion of the Wheel with mass m and moment of inertia I is:

$$F - f = m\ddot{x} \tag{49}$$

$$fr = I\dot{w} \tag{50}$$

$$F = m\ddot{x} + \frac{I\dot{w}}{r} \tag{51}$$

Assigning the constraint equation (assuming no slip)

$$\dot{x} = w\,r$$

and therefore

$$\dot{w} = \frac{\ddot{x}\,\mathrm{r} - \dot{x}\,\dot{r}}{r^2} \tag{52}$$

I rewrite equation 51 as

$$F = \frac{(m r^2 + I)\,\ddot{x}}{r^2} - \frac{I\,\dot{x}\,\dot{r}}{r^3} \tag{53}$$

Linearizing about an operating point of constant velocity and no force applied ($\dot{x} = \dot{x}_0$, $r = r_0$, $\dot{r} = 0$, $\ddot{x} = 0$, $F = 0$) and assuming that the values for $\delta\dot{x}$, δr, $\delta\dot{r}$, $\delta\ddot{x}$, and δF are small, the equation of motion, valid only in the vicinity of the operating point, is

$$\delta\ddot{x} = \left(\frac{r_0^{\,2}}{m\,r_0^{\,2} + I}\right)\delta F - \left(\frac{I\,\dot{x}_0}{m\,r_0^{\,2} + I}\right)\frac{\delta\dot{r}}{r_0} \tag{54}$$

or

$$\delta\ddot{x} = B_2\,\delta F - B_1\frac{\delta\dot{r}}{r_0} \tag{55}$$

This equation yields several insights that are useful in designing a controller for the Wheel. First, the acceleration of the Wheel is dependent on the rate of change of the radius of the hub. As a result a good choice for the control variable is, $u \equiv \frac{\delta\dot{r}}{r_0}$. This is a nice result since it can be thought of as a steering velocity which relates to previous cobot controllers. Secondly, if I choose a control law such as

$$u = G_1\,\delta x + G_2\,\delta\dot{x} + G_3\,\delta F \tag{56}$$

and substitute it in equation 55

$$\delta\ddot{x} = (B_2 - G_3B_1)\,\delta F - B_1G_1\,x - B_1G_2\,\dot{x} \tag{57}$$

the result is an equation of motion that reacts to the position, and velocity of the wheel, and the force applied by the user. This expression, therefore, resembles the dynamics of a mass, spring, and damper system. Also, by controlling the magnitude and sign of the gains G_i, the wheel model can be expected to simulate any mass, spring, damper system. Bearing this in mind, I expect that the Joystick will also be able to simulate a mass, spring, damper type system.

Another interesting point that is observed from the Wheel analogy is that when in energy mode it can impose a velocity on the user as opposed to passive mode where it can only specify a direction of motion. Energy mode can impose a velocity on the user due to the ball's stored kinetic energy. For instance, if the wheel is spinning, $w \neq 0$, absent of the rolling constraint (not touching the rail), as soon as the constraint is imposed the wheel has an $\dot{x}$ component.

The Wheel device can also be used to demonstrate how energy, which is de-coupled from the user, can be stored. Imagine, for instance, that the user pushes on the wheel when it is in no-slip contact with the rail. The wheel will start spinning. If at the same time the user stops pushing the wheel the rolling constraint is removed – putting the Wheel back in passive mode – the user's translational velocity is redirected into spin velocity. The resulting spin velocity supplies the kinetic energy necessary for implementing virtual behaviors.

In order to realize virtual behaviors the Joystick must have some stored energy, in other words the ball, like the wheel, should be spinning. In the case of the Joystick, I model the dynamics of the Joystick after a free-moving ball in space and add a *virtual* moment about the handle (about $\mathbf{a}_3$) -

like *fr* in the Wheel. Recall from §3.2 that the forces applied at the handle are not enough to cause the ball to spin - there is no resulting $\mathbf{a}_3$ component. Therefore, the role of a virtual moment about $\mathbf{a}_3$ is to give the angular velocity of the ball a spin component. Going back to the wheel analogy, it is similar to bringing the rail in contact with the hub. This moment has the effect of moving the AOR in a fashion that redirects some of the energy input by the user into spinning the ball; moving it out of the $\mathbf{a}_1$, $\mathbf{a}_2$ plane. The equations of motion for such a model becomes

$$b_i = {}^N\mathbf{W}^A \cdot \mathbf{a}_i \qquad i = 1, 2, 3 \tag{58}$$

$$M_i = \mathbf{M} \cdot \mathbf{a}_i \qquad i = 1, 2, 3 \tag{59}$$

$$a = {}^A\mathbf{W}^S \cdot \mathbf{a}_3 \tag{60}$$

where $\mathbf{M}$ is now

$$\mathbf{M} = -\mathrm{F_y}\,\mathrm{h}\,\mathbf{a}_1 + \mathrm{F_x}\,\mathrm{h}\,\mathbf{a}_2 + F_f\,\mathrm{R}\,\mathbf{s}_3$$

$$\dot{b_1} = \frac{M_1 - \mathrm{J}a b_2}{\mathrm{I}} \tag{61}$$

$$\dot{b_2} = \frac{M_2 + \mathrm{J}a b_1}{\mathrm{I}} \tag{62}$$

$$\dot{a_3} = \frac{M_3}{(\mathrm{I}_3 - \mathrm{J})} \tag{63}$$

Where $F_f\,\mathrm{R}$ is a virtual moment about the handle. Notice that the encoder in the ball can only measure the relative position between the ball and the handle therefore I will always assume that any change in position is due to spin and not due to the user twisting the handle ($\therefore b_3 = 0$). The equation of motion of the Joystick when it is spinning is,

$$\dot{w_1}\,\mathbf{n}_1 + \dot{w_2}\,\mathbf{n}_2 + \dot{w_3}\,\mathbf{n}_3 = (\dot{b_1} + a b_2)\,\mathbf{a}_1 + (\dot{b_2} - a b_1)\,\mathbf{a}_2 + \dot{a}\,\mathbf{a}_3 \tag{64}$$

By introducing a virtual applied couple about $\mathbf{a}_3$ I get a desired motion that includes an acceleration about the axis of the handle. By repeating some pattern of motion the user can transmit energy to the ball, forcing it to spin up.

Once the Joystick has some stored energy it can simulate, like the Wheel, virtual behaviors. For instance, some of these behaviors include:

- variable perceived inertia,
- variable perceived resistance,
- variable perceived assistance,
- perceived gyroscopic effects.

An intuitive discussion of how these behaviors might be achieved is appropriate.

An inertial effect is dependent on the mass of the ball. When simulating a inertial behavior the user should feel as though he/she is moving a more/less massive ball. A quick glance at the wheel example illustrates how this might be done.

Assuming that the radius of the hub is constant the equation of motion of the wheel becomes

$$F = \left(m + \frac{I}{r^2} \right) \ddot{x} \tag{65}$$

It is obvious then, that different values of *r* result in different perceived masses. For instance, when the user pushes on the wheel he is not only translating the wheel but he is also suppling the energy required to spin it. The additional energy necessary to spin the wheel is perceived as a heavier mass. This explanation applies directly to the Joystick. The AOR can be equally controlled so that for any angular velocity of the ball there is always a spin component. The ratio of precession velocity to spin velocity also causes the ball to feel more massive.

An assistance or resistance haptic effect can be described when the ball has some stored energy. For example, imagine that the ball is spinning with constant velocity by means of a motor. It is apparent that if the AOR of the Joystick is the same as the axis of the handle all the kinetic energy of the motor goes into spinning the ball. However, as soon as the AOR is no longer parallel to the axis of the handle some of the energy of the motor goes into precessing the ball. (The ratio of spin velocity to precession velocity is dependent on the distance of the AOR from the handle axis.) Therefore, by controlling the AOR and knowing where the handle is, a precession velocity can be imposed. It is then possible to detect the direction of motion desired by the user – by measuring the direction of the applied force – and set the AOR so that the precession velocity resulting from the motor energy is in the direction desired by the user or opposite the direction desired by the user. Doing this will in effect feel as an assistive or resistive haptic effect. Other virtual behaviors analogous to a mass, spring, damper system can also be simulated.

On account of design similarities between the Joystick and a gyroscope it is also possible to simulate Coriolis effects. By modeling the motion of the Joystick after that of a gyrostat the user will be able to perceive a gyroscopic behavior (Coriolis acceleration). A response to Coriolis accelerations is in effect the true 'caster mode' of the Joystick when the ball is spinning.

Previous cobots had no way of controlling the desired speed of motion of the user, instead a cobot controls the direction of motion and the speed was left up to the user. Because of this only the direction component of the force applied at the handle has been necessary. Energy mode, however, allows for the control of both speed and direction of motion, velocity. Therefore, implementing virtual behaviors requires that the magnitude of the force applied by the user also be considered. The issue of how to relate the user's applied force to the his/her velocity, however, is entirely dependent on the virtual behavior being simulated.

3.5 Research Questions

The implementation of energy mode for the Joystick is essentially the core of my research. Hence, the main research questions that I will be addressing are particular to this mode. These are the following:

- How will natural gyroscopic tendencies affect the efficiency of the simulation of virtual behaviors?

Due to the gyroscopic nature of the Joystick it is reasonable to expect that physical gyroscopic effects will come into play in the efficiency of the energy storage and transmission. The way a gyroscope reacts to a force applied at the tip of its spin axis, the handle, will also be the Joystick's natural reaction. The way a gyroscope reacts to an applied force, however, is not very intuitive to a user. In implementing virtual behaviors I must assume that the direction of the user's applied force is his/her desired direction of motion. Therefore, since Coriolis accelerations move the handle in a direction normal to the balls angular velocity and the user's velocity ($\mathbf{w}\times\mathbf{v}$), they must be canceled.

Forcing the ball to move in a direction that does not comply with its physical model will limit the maximum angular velocity of the ball. Redirections of the AOR at high velocities will tend to cause the ball to jump up and lose contact with the wheels. This will undoubtedly affect the efficiency with which virtual behaviors can be simulated.

- What is the range of masses that can be simulated?

  In simulating inertial effects I will be changing the perception of the ball to be that of a more or less massive ball. There is a question as what is the smallest mass that can be simulated. These inertial effects are directly linked with the steering velocity and hence the maximum motor velocity. To explain this point imagine the simulation of a zero mass. A zero mass accelerates infinitely fast with any non-zero force applied. In the case of the ball this implies that the AOR has to accelerate infinitely fast also. Since the AOR is imposed by the steering wheels, the motors also have to be able to accelerate at that rate. Therefore, the smallest mass that can be perceived will be related to the motors' maximum acceleration.

  The largest mass that can be simulated, on the other hand, will be directly dependent on the wheel hardness. The harder the wheel the smaller the contact patch between the wheel and the ball. The smaller the contact patch the smaller the force required to cause the ball to slip. Therefore, the perceived mass is directly linked to the hardness of the wheel in that if the force required to move the simulated mass is larger than the force of friction, the ball will slip.

- Similarly, what is the range of damping and spring constants that can be simulated?

- What is the correct kinematic space to describe energy cobots?

  A configuration space such as Scooter's is probably not the most convenient way to describe the kinematics of an energy cobot. The Joystick, unlike Scooter or the Unicycle, can impose a position and a velocity. Consequently, I expect that a more adequate kinematic space for the Joystick will be one which includes some way of specifying a velocity such as state space.

- How well can AOR control be implemented and how does it affect the efficiency of virtual behaviors?

  In implementing virtual behaviors it will become imperative to have a very efficient low level AOR controller. Any misalignment of the three wheels will have the tendency to slow down the spin of the ball and thereby affect the efficiency with which virtual behaviors can be implemented.

3.6 Engineering Issues

Following is a list of some of the practical issues that will have to be dealt with when implementing the Joystick controller:

- resolution of the coil sensors,
- twisting of the handle by the user,
- steering wheel hardness,
- and how to keep the ball spinning when in energy mode?

Appendix A: *Joystick Drawings*

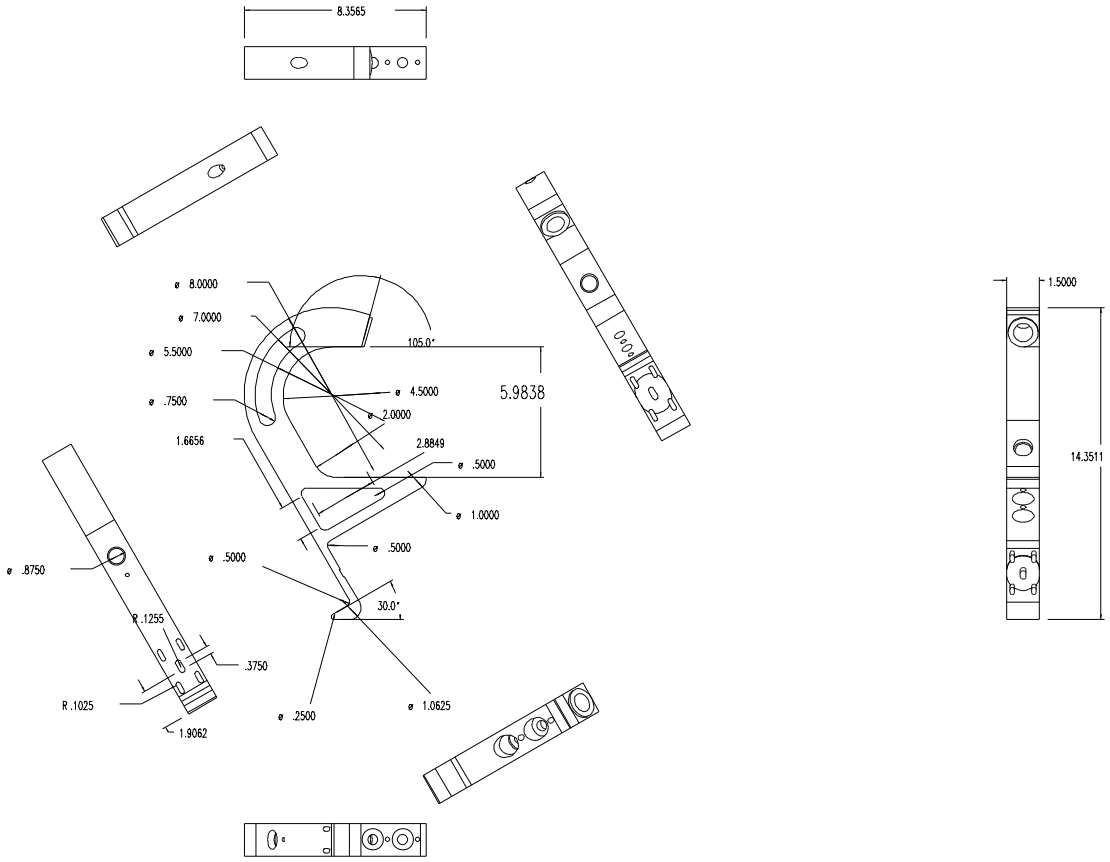


Figure 3-4. CAD drawing of the Joystick's support frame.

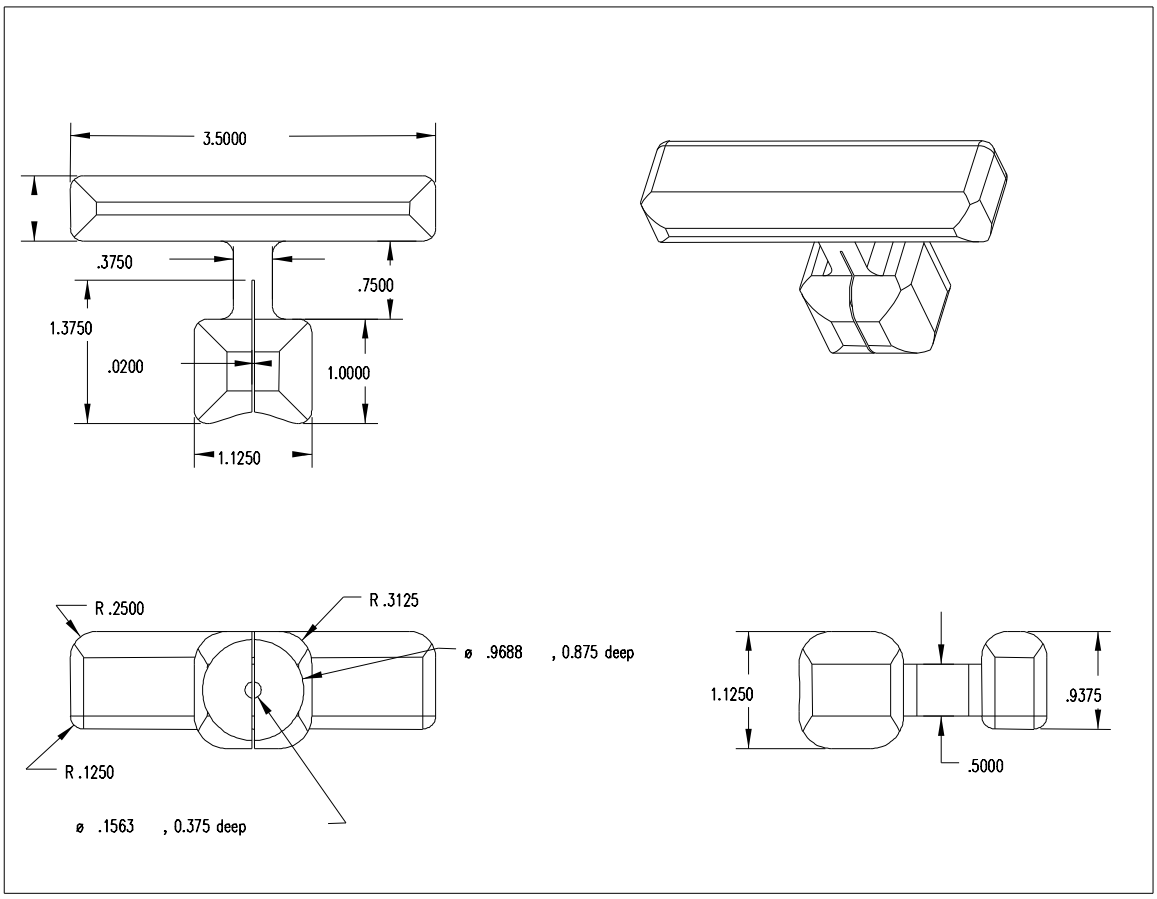


Figure 3-5. CAD drawing of the Joystick's handle.

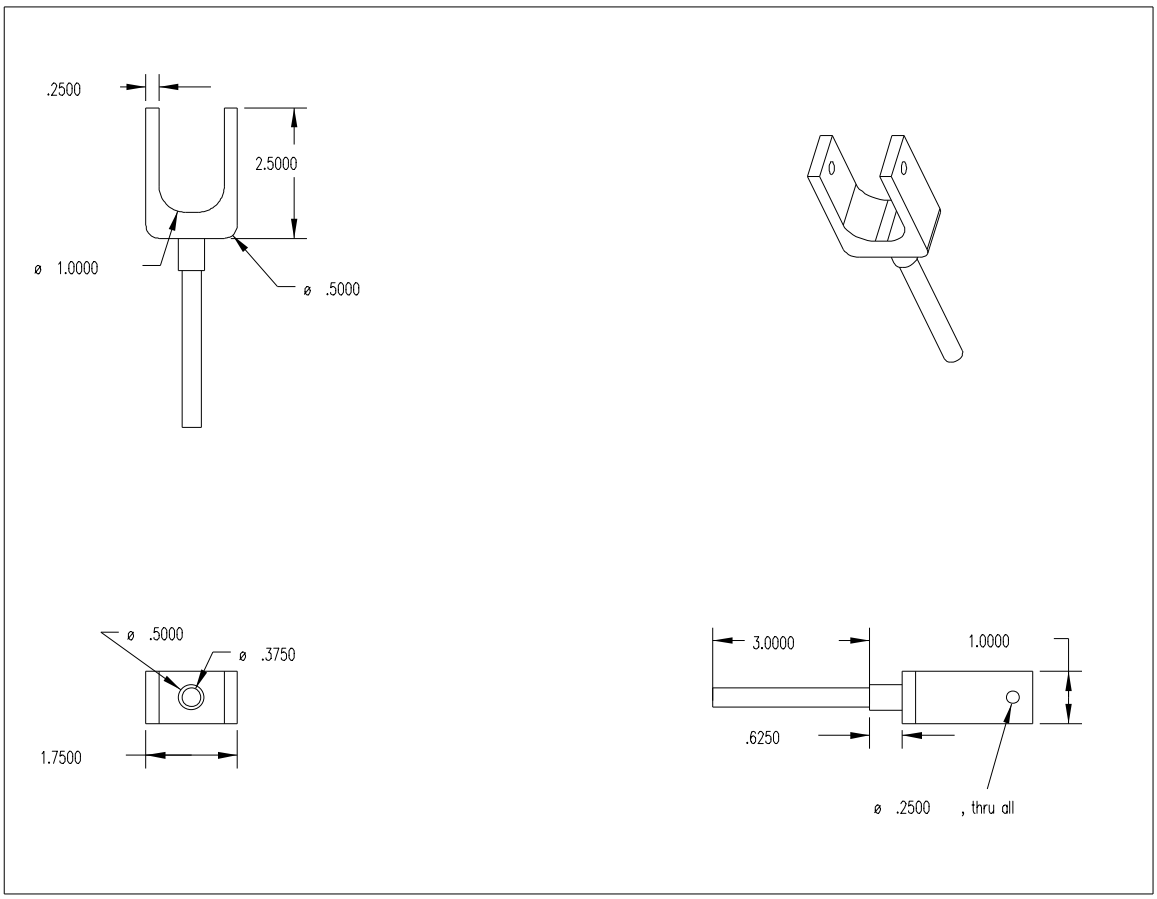


Figure 3-6. CAD drawing of the wheel shaft and fork.

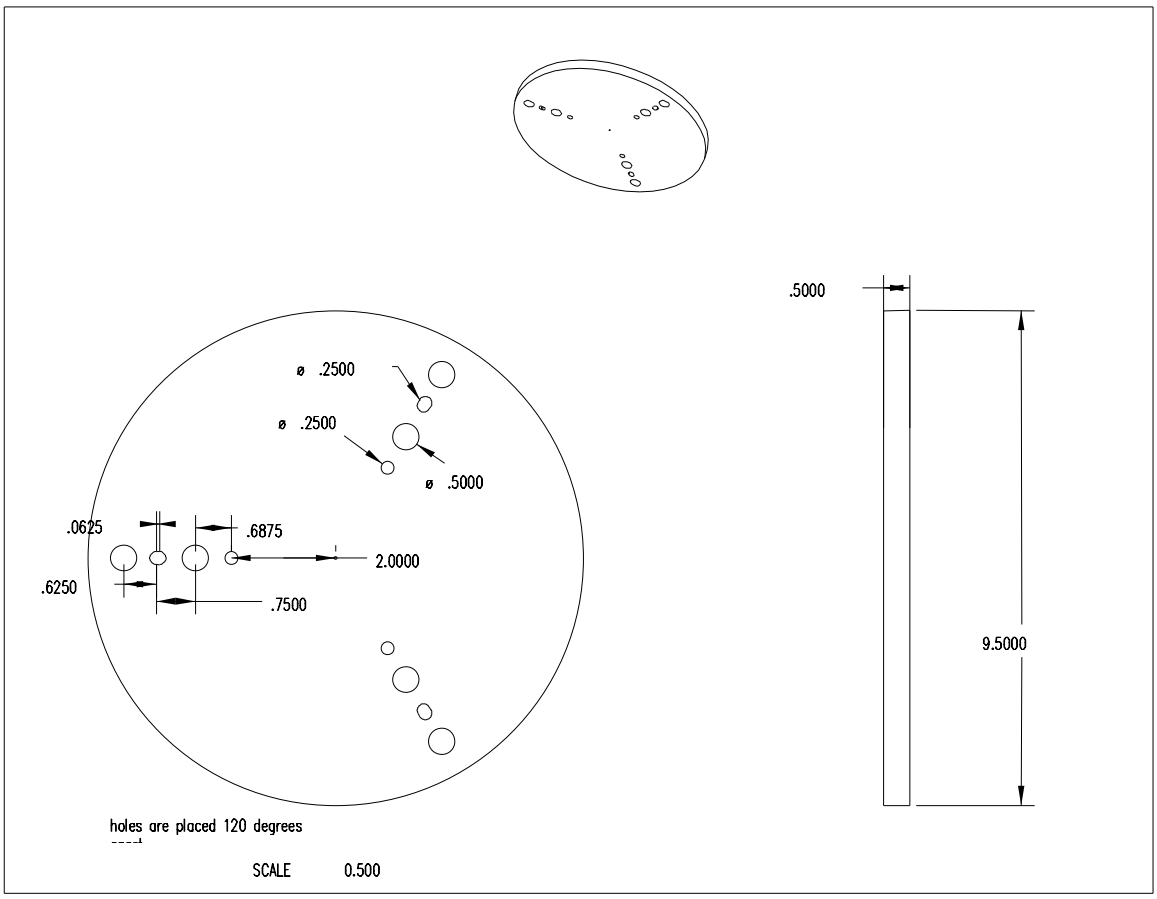


Figure 3-7. CAD Drawing of the base plate.

Index

Bibliography:

1 *Nonholonomic Haptic Display*; J.E. Colgate, M.A. Peshkin, W. Wannasuphoprasit; Proceedings IEEE International Conference on Robotics and Automation, Minneapolis, MN, 1996.

2 *Cobots: Robots for Collaboration with human operators*; J.E. Colgate, W. Wannasuphoprasit, M.A. Peshkin;

3 *Passive robots and haptic displays based on nonholonomic elements*; M. Peshkin, J.E. Colgate, C. Moore;

4 *Cobot Control*; W. Wannasuphoprasit, R.B. Gillespie, J.E. Colgate, M.A. Peshkin; Proceedings IEEE International Conference on Robotics and Automation, Albuquerque, NM, 1997.

5 *Cobots have singularities, too*; C. Moore, B. Gillespie, M.A. Peshkin, J.E. Colgate;

6 *Unicycle Cobot Control*; A. Mills, W. Wannasuphoprasit, J.E. Colgate, M.A. Peshkin;

7 Controlling Dissipative Magnetic Particle Brakes in Force Reflective Devices; M. Russo, A. Tadros; In H. Kazerooni (Ed.), ASME Winter Annual Meeting, (pp. 63-70), Anaheim, California.

8 PADyC: a Passive Arm with Dynamic Constraints. A prototype with two degrees of freedom; Y. Delnondieu, J. Troccaz; Proceedings Medical Robotics and Computer Assisted Surgery, pp. 173-180, Baltimore, MD, Nov. 1995.

9 *Factors Affecting the Z-width of a Haptic Display*; J.E. Colgate, J.M. Brown; Proceedings International Conference on Robotics and Automation, vol. 4 (pp. 3205-10), San Diego, CA, 1994.