Non-Holonomic Attitude Control

Alok Agrawal
aloka@cs.berkeley.edu

1 Introduction

The problem of manipulating satellite orientation in space has always been an important one. For the proper functioning of a satellite it is important that the satellite be properly oriented with respect to the receiving station on earth because the radio waves used for communication have to be correctly directed towards the receivers. It also has to be correctly oriented with respect to the sun to ensure that the solar panels are properly illuminated. A satellite tends to constantly drift from its position due to the solar wind pressure. One way to correct this drift is by the help of jet propulsion, where small jets attached to the frame of the satellite can rotate it back to the right orientation. However dependence on jets for maintaining orientation restricts the lifetime of the satellite as the amount of fuel that can be shipped with the satellite is limited. Since the satellite has an onboard source of energy (the solar panels) there should be some way to reorient the satellite using some internal motion. A related problem is the manipulation of objects in space by manipulators attached to the satellite's body. This kind of work requires manipulation of the some other body in space by a manipulator fixed to satellite. In free space, however, such manipulations become very difficult because the motion of the manipulator and the body to which it is attached are coupled. Any change in the manipulators motion will induce a change in the orientation of the body and this restricts arbitrary movement of the manipulator.

Both these problems can be solved if we can somehow find a way to relate the motion of the manipulators/solar-panels and the satellite. Since the angular momentum conservation works as a non-holonomic constraint, it so happens that we actually can control the orientation of the satellite as well as the manipulator/solar-panels just by the relative motion of the manipulator/solar-panels.

A number of people have studied this problem. Nakamura and Mukherjee [5] used the Lyapunov design methodology to converge the joint angles and the orientation of the satellite to the final state. Vafa and Dubowsky [1] used the notion of virtual space manipulator to control the satellite orientation while moving the manipulator attached to it arbitrarily. Longman, Lindberg and Zadd [2] have also discussed the coupling of a manipulator motion and the vehicle motion. Walsh in his thesis [4] has studied the problem of reorienting a linked rigid body (with three links) in space with internal motions. Hereafter I will restrict my discussion to a satellite with two solar panels or with a manipulator with two links, unless otherwise stated. For the project I have implemented a simulator based on Walsh's approach, which given the physical parameters of the system and the initial and final configuration decides on the requisite motions to be imparted to the joints to take the system to the final configuration. In the next few sections I briefly discuss the different approaches available and discuss the the advantages and disadvantages of each approach.

2 Planning using Lyapunov function

The method suggested by Nakamura and Mukherjee makes use of the fact that the angular momentum conservation is a nonholonomic constraint while the linear angular momentum conservation is a holonomic constraint. A satellite with a manipulator with n-degree-of-freedom manipulator has a total of $(n+6)$-degrees of freedom. Using the holonomic constraint, we can formulate the system as a nonholonomic system with $(n+3)$ variables out of which, n variables corresponding to the manipulator joints, are independently controllable and the other 3 are dependent on them. This means that we use the n independent variables to guide a set of $(n+3)$ variables. The first step of their approach is to find a system function and the control variable. This is done by making use of the principles of conservation of angular momentum and linear momentum.

If we let $\dot{\theta}_1$ describe the vehicle motion and θ_2 the manipulator motion where

$$\dot{\theta}_1 = \begin{pmatrix} I_{\dot{r}_0} \\ I_{\omega_0} \end{pmatrix}$$

$$\theta_2 = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix}$$

$I_{\dot{r}_0}$ is the linear velocity of the center of mass of the vehicle and I_{ω_0} is its angular velocity. $q_1, q_2 \ldots q_n$ are the joint angles. The linear and angular momentum conservation equation can be represented as

$$H_1\dot{\theta}_1 + H_2\dot{\theta}_2 = 0$$

In the above equation H_1 and H_2 are a function of the geometry of the system and its physical parameters. H_1 is always non-singular an so this can be simplified to

$$\dot{\theta}_1 = -H_1^{-1}H_2\dot{\theta}_2$$

The basic system equation is obtained by taking vehicle orientation and θ_2 as the state variable ($(n+3)$-variables and $\dot{\theta}_2$ as the input variable (n-variables).

$$x = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \theta_2 \end{bmatrix} \in R^{n+3} \qquad\qquad u = \dot{\theta}_2 \in R^n$$

Here α, β and γ are the Euler angles of the vehicle with respect to an inertial frame. The system equation with these definitions becomes

$$\dot{x} = Ku$$

where the matrix K is a function of H and α, β, and γ and u is the control variable.

The actual control strategy involves designing an appropriate Lyapunov function which would enable the choice of the control variable u for the system. The interested reader can see [5]. The authors have proposed a Lyapunov function based on reducing the the error of the final orientation of the satellite and the manipulator taking into account the nonlinearity of the satellite/manipulator system.

The advantage of this method is that, as the authors claim, a non linear control scheme synthesized from this can do the manipulation in real time. A big disadvantage is that it is not a guaranteed strategy and fails to achieve many configurations. The Lyapunov function can get caught in a local minima and in which case the final configuration of the system wouldn't be the desired final configuration.

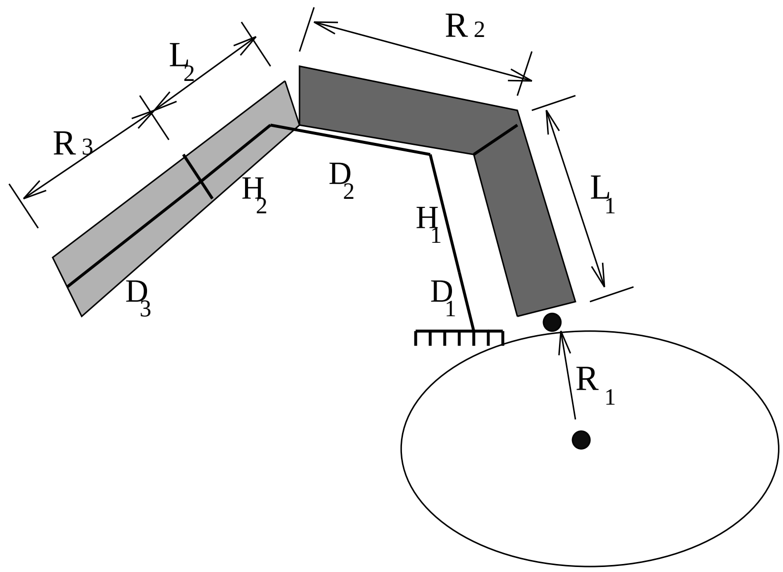

Figure 1: A Satellite with an attached manipulator and the corresponding Virtual Manipulator

3 Virtual Manipulators

In this approach by Vafa and Dubowsky [1], the satellite and the manipulator/solar-panels attached to it are transformed to a virtual manipulator and the control strategy is derived by looking at the virtual manipulator. The *Virtual Manipulator* is a massless kinematic chain. The virtual ground is an imaginary base to which the virtual manipulators base is attached. This point is assumed to be fixed, i.e it does not move when the links move. This property make it much easier to study motion of the virtual manipulator. Another important property of a virtual manipulator which comes in handy is that the axis of a virtual link is parallel to the corresponding link of the actual manipulator. As a result the rotation of the satellite now becomes the rotation of the base link attached to the fixed base. (the joint is assumed to be spherical so it allows rotation along all three directions). The transformation from a general satellite/manipulator system to the virtual manipulator can be easily done.

Figure 1 shows the actual satellite with a two link manipulator attached to it and the virtual manipulator that has similar dynamic and kinematic structure. The virtual manipulator is shown hinged to an imaginary base. As can be seen from the figure, the end effector of the virtual manipulator and the real manipulator are at the same position and also the corresponding links of the real and virtual manipulator are parallel. Since the virtual manipulators base is attached to a fixed(imaginary) ground, the analysis of it kinematic and dynamic equations is much more simpler.

The actual method used to manipulate the satellite orientation involves small motions of the manipulator joints which form a closed curve in the the *Joint Space*. These cyclic motions result in small changes in the satellite's orientation and by repeating them a number of times the desired configuration can be attained. For a general satellite/manipulator system the following equation holds.

$X = F(q, X)q$

where

$F(q)$: 3xn matrix with elements $F_{i,j}$ - determined by the geometry of the system.

X: 3x1 vector of vehicle inertial orientations.

q: nx1 vector of joint angles with elements q_i. In our case these would be the joint angles.

Let X be a set of Euler angles defining the base orientation with respect to an inertial coordinate frame. If the initial and final orientations are X_i and X_f, then the desired change is $\delta X = X_i - X_j$. The correction is applied by small motion of the joint angles such that after each step they are back to their original state. This looks like a closed curve in the joint space (Figure 2).

If we do the correction in four steps, move along W then along V then along $-W$ and $-V$, the closed path would look as shown in Figure 2. From the previous equation we get, for small W and V,

$$\delta X_k = \sum_{i=1}^{3}\sum_{j=1}^{3}\left[\sum_{m=1}^{3}\left(\frac{\delta F_{kj}}{\delta X_m}F_{mi} - \frac{\delta F_{ki}}{\delta X_m}F_{mj}\right) + \frac{\delta F_{kj}}{\delta q_i} - \frac{\delta F_{ki}}{\delta q_j}\right] W_j V_i \qquad (k = 1, 2, 3)$$

Let $T(X_j) \in SO$ be a 3x3 matrix which transforms a vector expressed in the vehicle body coordinates into inertial or Newtonian coordinates, when the body is at the j_{th} orientation. After one correction of the orientation this becomes

$$T(X_j + \delta X) = T(X_j)A$$

The matrix A is the transformation matrix and is dependent on W and V. This is a constant because the total system, the manipulators and the satellite have been subjected to a rigid body rotation in the space. So $T(X_f) = T(X_i)A^m$ (m is the number of cycles). We can solve this equation for A easily. Using this and the previous equation and the constraint equations on V and W we can solve for the V and W which then give the incremental rotation matrix A.

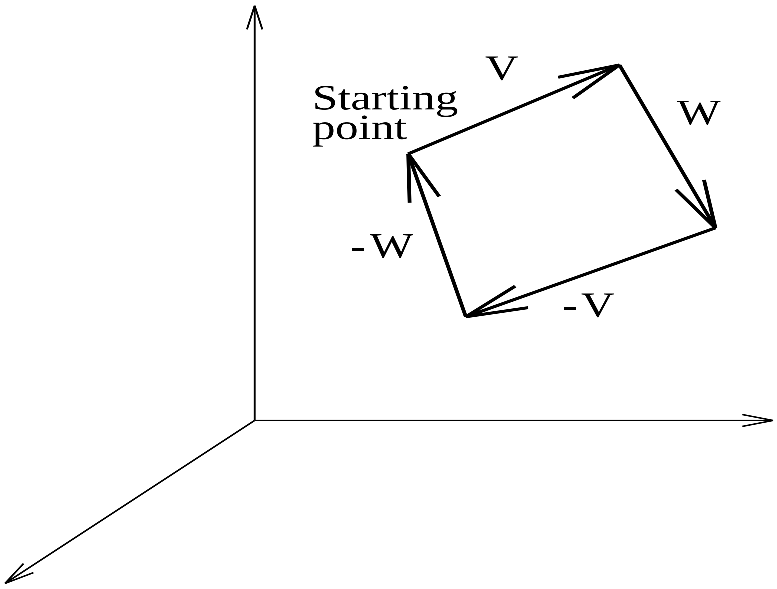

Figure 2: The cyclic correction in the configuration space

The appealing thing about his approach is its simplicity. It is a good algorithm if we need to move the manipulator to a new configuration keeping the satellite orientation unchanged. Naturally for fine tasks the manipulator has to move slowly. However the problem with using it for changing the satellite orientation can become very painful as the number of step can be very large.

4 Walsh's Algorithm

The Walsh's method for reorienting satellite takes an incremental approach. It reduces the errors in the configuration space of the system by eliminating errors in a of sequence of shape spaces (which are sub spaces of the configuration space). The method allows relatively large inputs to be given to the manipulator /solar-panels during the reorienting step and so the number of steps needed to reorient the satellite are generally very small. A brief description of the algorithm follows.

The lagrangian for the three link system (satellite and the two panels) can be written in the following form.

$$L = (\omega_1^T, \omega_2^T, \omega_3^T) J(R_{12}, R_{13}, R_{23}) \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

Here J is the Jacobian of the system and has the following form :-

$$J = \begin{bmatrix} M_{11} & M_{12}R_{12} & M_{13}R_{13} \\ R_{12}^T M_{12}^T & M_{22} & M_{23}R_{23} \\ R_{13}^T M_{13}^T & R_{23}^T M_{23}^T & M_{33} \end{bmatrix}$$

Where M_{ij}'s are constant matrices determined by the geometry of the system and the physical parameters. R_{ij}'s depend on the orientation of the three links at any moment.

The angular momentum can then be written as,

$$AM = [III]J(R_{12}, R_{13}, R_{23}) \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}$$

Define :

$A_1 = M_{11} + R_{12}^T M_{12}^T + R_{13}^T M_{13}^T$

$A_2 = M_{12}^T R_{12}^T + M_{22} + R_{23}^T M_{23}^T)$

$A_3 = M_{13}R_{13} + M_{23}R_{23} + M_{33}$

$V = [A_1 + A_2 + A_3]^{-1}$

Assuming that V is well defined, the null space vectors for this are :

$$b_1 = \begin{bmatrix} I - VA_1 \\ -VA_1 \\ -VA_1 \end{bmatrix} \qquad b_2 = \begin{bmatrix} I - VA_3 \\ -VA_3 \\ -VA_3 \end{bmatrix}$$

Given these as the input vector fields we define,

$\phi_1 = \omega_1 - \omega_2$

$\phi_2 = \omega_3 - \omega_2$

$\phi_3 = \omega_2$

Then we can write :

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ -VA_1 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ I \\ -VA_3 \end{bmatrix} u_2$$

Here, ϕ_1 and ϕ_2 are independently controllable. These are the motion given to the actuators at the joints of the manipulator. ϕ_3 represents the motion of the satellite (the middle link). The

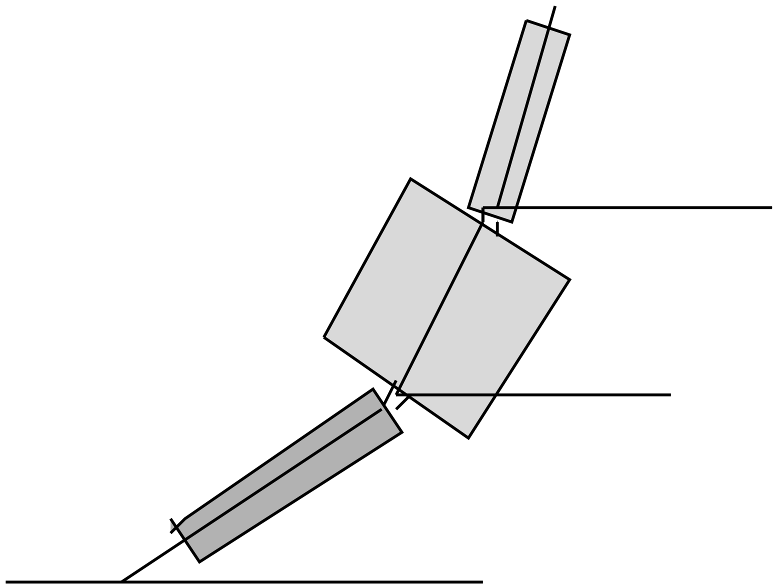

Figure 3: Projection of satellite and attached solar panels in a plane

planar version of this problem is simpler to analyze. Since I have implemented the planar version I will discuss the algorithm for the planner case only. Assuming that in the home configuration the center of mass of all the bodies lies along the x-axis we get,

$$\begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ b_1(\psi_1, \psi_2) \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ I \\ b_2(\psi_1, \psi_2) \end{bmatrix} u_2$$

With the following definitions :

$\psi_1 = \theta_1 - \theta_2$

$\psi_2 = \theta_3 - \theta_2$

$\psi_3 = \theta_2$

$a_1(\psi_1, \psi_2) = k_{11} + k_{12}\cos(\psi_1) + k_{13}\cos(\psi_1 - \psi_2)$

$a_2(\psi_1, \psi_2) = k_{12}\cos(\psi_1) + k_{22} + k_{23}\cos(\psi_2)$

$a_3(\psi_1, \psi_2) = k_{13}\cos(\psi_1 - \psi_2) + k_{23}\cos(\psi_2) + k_{33}$

$v = (a_1(\psi_1, \psi_2) + a_2(\psi_1, \psi_2) + a_3(\psi_1, \psi_2))^{-1}$

$b_1(\psi_1, \psi_2) = -a_1(\psi_1, \psi_2)v$

$b_2(\psi_1, \psi_2) = -a_3(\psi_1, \psi_2)v$

The angles θ_1, θ_2 and θ_3 parametrize the configuration. They represent the amount by which the corresponding link has to be rotated to align it with its position in the home configuration. ψ_1 and ψ_2 are the relative angles of link2 w.r.t link1 and link2 w.r.t link3, and k_{ij}'s are scalar constants determined by the physical parameters of the system..

Taking the lie bracket for the two vector fields and checking the system for controllability we get :-

$[v_1, v_2] = \xi(\psi_1, \psi_2) = 2v^2(a_1k_{23}\sin(\psi_2) - a_3k_{12}\sin(\psi_1) + a_2k_{13}\sin(\psi_1 - \psi_2))$

$det[v_1, v_2, [v_1.v_2]] = \xi(\psi_1, \psi_2)$

The determinant of the controllability matrix is zero at many points of the shape space. This superficially implies that the system lacks controllability, however, it turns out that by avoiding the singular points we can control the system. An interesting thing in this case is that the phase change induced in ψ_3 is given by

$\psi_3(1) - \psi_3(0) = \int_{area} \xi(\psi_1, \psi_2)dA$

The induced phase change is a function of the determinant of the controllability matrix. So if in some region $\xi(\psi_1, \psi_2)$ is non zero then the phase difference given by the above integral will also be non zero (for an integral) over a small region about that point (this follows from the continuity of $\xi(\psi_1, \psi_2)$). The function $\xi(\psi_1, \psi_2)$ is non zero at $(-\pi/2, -\pi/2)$ and since it is continuous, the integral would also be non zero over some small region about this point. Taking this fact into account the final algorithm looks as follows.

Step 1 : Using constant inputs u_1 and u_2 drive the system to the final values of ψ_1 and ψ_2. Measure the error in ψ_3 and make a note of it.

Step 2: Using constant inputs drive the system to $(-\pi/2+k, \pi/2)$. It is driven to $\psi_1 = -\pi/2+k$ to ensure that the two ocillations are about $(-\pi/2, -\pi/2)$. By an appropriate combinations of sinusoidal inputs $u_1 = k\cos(t)$ and $u_2 = k\sin(t)$ correct the error in ψ_3. This may require more than one cycle and different amplitudes at each cycle.

Step 3 : Using constant inputs drive the system back to the final ψ_1 and ψ_2.

5 Implementation

Each of the three algorithms discussed above have many disadvantages as well as advantages. The Nakamura-Mukherjee approach is good in the sense that it can be used for any configuration of the satellite and the manipulator. The Vafa-Dubowsky approach is also generic and can be used for any configuration of the satellite and the manipulators. Walsh's approach works well for the case when we have a simple configuration in which the system can be looked upon as three rigid bodies connected at two points. His thesis does not provide formulation for links with larger degree of freedom, however, there is no reason why the current approach cannot be extended for systems with more links. However, the problem with his approach is that it would become very complicated in a higher dimension shape space. If we restrict ourselves to three link systems, Walsh's approach is substantially better than the other two approaches. The problem with Nakamura-Mukherjee approach s that it can result in the system never reaching the final state while the Vafa-Dubowsky approach would take a very large number of steps to attain the final configuration. Because of these reasons, I decided to implement Walsh's algorithm.

The implementation has been done in MATLAB. A part of the simulator is written using SIMULINK. The part written in MATLAB is primarily to find out the optimum amplitude of the sinusoids to apply to the solar panels and the part written in SIMULINK actually calculates how the satellite rotates with the motion of the panels. The inputs to the simulator are the physical parameters of the system and the outputs are the motion that has to be imparted to the panels and the resulting motion of the satellite. The set of inputs to the simulator are listed below.

m_1 : Mass of the first solar panel (link 1)

m_2 : Mass of the satellite (link 2)

m_3 : Mass of the second solar panel (link 3)

I_{1_x} : Moment of link 1 about x axis.

I_{1_y} : Moment of link 1 about y axis.

I_{2_x} : Moment of link 2 about x axis.

I_{2_y} : Moment of link 2 about y axis.

I_{3_x} : Moment of link 3 about x axis.

I_{3_y} : Moment of link 3 about y axis.

d_1 : Vector from the joint of link 1 to its center of mass in its own reference frame.

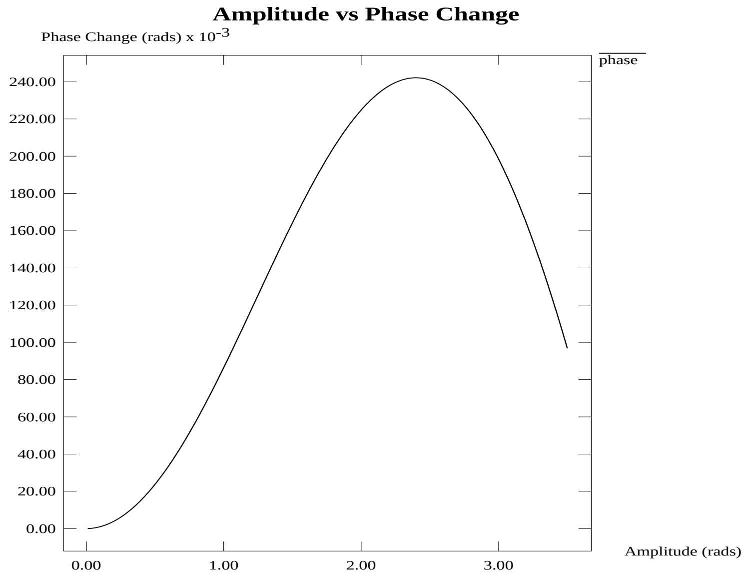

Figure 4: Phase change as a function of input amplitude

d_3 : Vector from the joint of link 2 to its center of mass in its own reference frame.

r_1 : Vector form the center of mass of link 2 to the joint with link 1 in link 2's reference frame.

r_3 : Vector form the center of mass of link 2 to the joint with link 3 in link 2's reference frame.

$\theta_{11}, \theta_{21}, \theta_{31}$: Initial configuration

$\theta_{21}, \theta_{22}, \theta_{32}$: Final configuration

6 Results

In step 2 of the reorienting algorithm we drive the system using sinusoidal inputs. Since the phase difference induced is dependent on the amplitude of the sinusoids applied, selection of the amplitude becomes very important. The function relating the amplitude with the phase difference is not a simple one and does not yield a close form solution. I tried out two different strategies for selecting the amplitude. The first strategy gives the optimal solution i.e it reorients using the minimum steps, however it tends to be computationally more expensive. The other strategy uses a greedy approach. Though it yields a solution quickly the solution is not optimal.

The first strategy which gives the optimal solution is simple. We compare the error that needs to be corrected with the largest phase shift possible (this occurs for an amplitude of about 2.4 . See

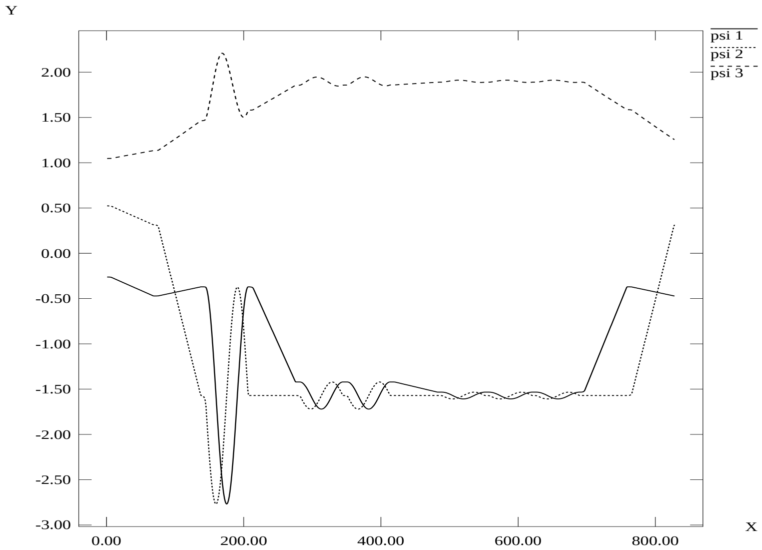

Figure 5: The orientation strategy using the greedy approach

Figure 4). If the error is more than this, we apply some cycles of inputs with this amplitude until the error has been reduced to zero or is less than the maximum phase shift possible. If there is still some error, there will be some amplitude in the range $[0, A]$ which can correct this in one cycle (this is true because the function is continuos). Find this amplitude and correct the error. The reason why computing the required amplitude generally takes a lot of time is that for middle portions of the curve the phase change induced is very sensitive to the amplitude and so the iterative solution takes a long time to converge to the correct amplitude.

The other approach is to start with an amplitude which gives the maximum phase difference. Check if this can be used to correct the error. If so apply until the error is zero or has become less than the phase shift induced. In the next step use an amplitude which is half of amplitude in this step. This process is repeated until the error reduces to zero. This approach converges quickly because the error goes down quickly and for lower portions of the curve, the phase change induced is not very sensitive to the phase difference. In an actual system, the overhead of finding the reorientation strategy is not as important as the need to reduce the number of steps in the strategy. In such a case the first approach would be a better one.

Figure 5 shows the reorientation strategy that results when we use the greedy approach. The figure shows the plots of the three variables ψ_1, ψ_2 and ψ_3. In the first step of the algorithm both

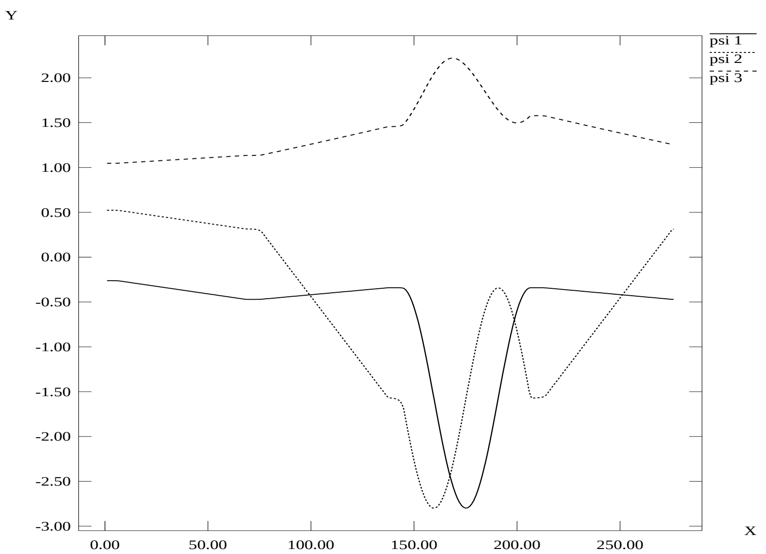

Figure 6: The optimal orientation strategy

ψ_1 and ψ_2 are driven to their final values. The the error in ψ_3 is noted and an amplitude is chosen. Sinusoidal inputs are provided to the solar panels. The phase difference in the motion of the two panels is clearly evident. After doing a couple of oscillations, the error gets reduced to a value that is less than the phase difference induced by the current amplitude so it is reduced to half. During this stage, ψ_2 remains unchanged. The same cycle is repeated until the error becomes zero. The final step is changing ψ_1 and ψ_2 to their final values. This step has to take place in two parts. In the first step ψ_1 changes while ψ_2 is kept constant. ψ_1 is driven to the point $(-\pi/2 + k)$ where k is the amplitude that the system started with. We have to do this because both the solar panels were driven at the beginning of step 2 of the algorithm such that they reached $(-\pi/2, -\pi/2 + k)$ simultaneously and so to ensure that ψ_3 takes the correct value finally they have to retrace back from exactly this state. After ψ_1 has been driven to $(-\pi/2 + k)$ we drive both ψ_1 and ψ_2 to the final configuration.

The reorientation strategy using the optimal approach is shown in Figure 6. It reorients the satellite with just one cycle of sinusoidal input to the system. This is definitely better because in general the time to actually move the solar panels by one cycle would be much larger compared to the computation time and so a little extra time in computation wouldn't matter as long the number of cycles are smaller.

7 Conclusion

In this paper I have discussed three different proposed solution to the problem of Non-Holonomic attitude control. Each of the alternative has many advantages as well as disadvantages. The algorithm to be used would depend on the system that has to be controlled. For the system that I considered, a satellite with two attached solar panels, Walsh's approach appears to be the best. With minimal cycles the system can be taken form any initial configuration to any final configuration. Since the amplitude of the sinusoids applied to the panels can be considerably large the phase shift obtained in each cycle is larger than the phase shift obtained in one cycle of the Vafa-Dubowsky algorithm. In fact their algorithm works only when the motion of the panels is very small. Walsh's algorithm for the 3-dimensional case is similar to the one for 2-dimension, however, considerably more complicated. This approach can be extended to systems with greater degrees of freedom, however the formulation would be much more complicated. In such cases, presumably, Nakamura-Mukarjee's approach would be more appropriate.

References

[1] Vafa, Z. and Dubowsky, S. "On the Dynamics of Manipulators in Space Using the Virtual Manipulator Approach". *Proceedings of 1987 IEEE International Conference on Robotics and Automation, 1987*, pp 579-585.

[2] Longman, R.W., Lindberg, R.E and Zadd, M.F. , "Satellite Mounted Robot Manipulators-New Kinematics and reaction Motion Compensation." *International Journal of Robotics Research, 1987*, Vol 6, No. 3, pp. 87-103.

[3] Kohn, W. and Healey, K., " Trajectory Planar for an Autonomous Free-Flying Robot", *Proceedings of IEEE International Conference on Robotics and Automation, 1986.*

[4] Walsh, G. "On Reorienting Linked Rigid Bodies Using Internal Motions", *M.S Thesis, U.C. Berkeley, 1990.*

[5] Nakamura, Y. and Mukharjee, R. " Nonholonomic path planning of space robots", *IEEE International Conference on Robotics and Automation, 1989*, pp. 1050-1055.

[6] Sreenath, N., Oh, Y.G., Krishnaprasad P.S. and Marsden, J.E., “The Dynamics of Coupled Planar Rigid Bodies , 1988”, *Dynamics and Stability of Systems*, vol. 3, pp. 25:49.

[7] Carmo, Manfredo P.D., “ Differential Geometry of Curves and Surfaces, 1984 ”, *Prentice-Hall Inc., Englewood Cliffs, New Jersey.*