Providing a Camera Sensor with Pointing Capabilities Independent of an Unmanned Ground Vehicle

Donald T. Shrewsbury

Report submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science (non-thesis)
in
Aerospace Engineering

Hanspeter Schaub, Chair
Christopher D. Hall
Craig A. Woolsey

November 2006
Blacksburg, Virginia

Contents

List of Figures

I. Introduction

The Autonomous Vehicle Systems (AVS) Lab at Virginia Tech is home to an emerging form of spacecraft simulator. The use of unmanned ground vehicles (UGVs) to simulate planar spacecraft movement is unique compared to other spacecraft simulators because UGVs can simulate various spacecraft orientations in addition to simulating the track that the spacecraft follows. There are some initial physical constraints which limit the simulation capabilities of using a UGV. For one, it is a two-dimensional simulator. Therefore, it cannot simulate changes in the inclination of an orbit or out of plane maneuvers. The UGV also does not have the capability to control its movement in the pitch or roll direction. The wheels of the UGV are side-mounted, thereby constricting its movement in the orbital plane. The UGV, for example, cannot perform a lateral shift to one side or another. It must first rotate 90 degrees, translate forward, and then rotate 90 degrees in the opposite direction. These limitations provide the inspiration for this project.

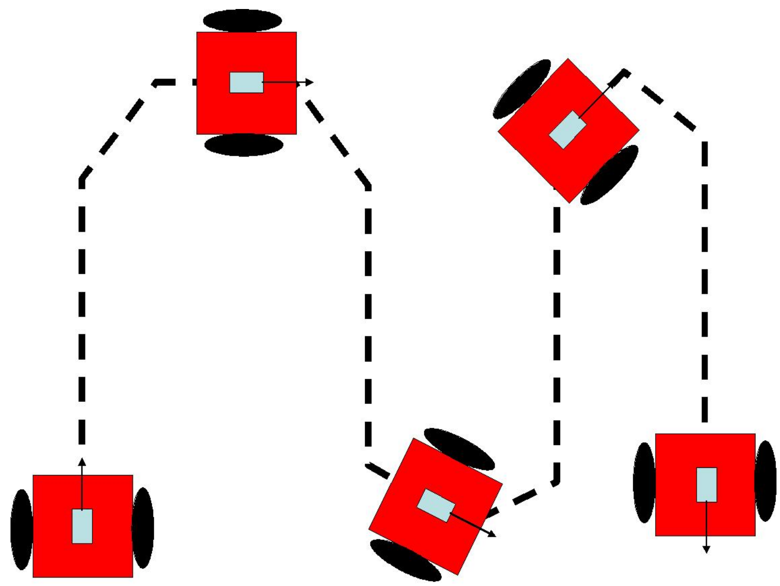

Figure 1: Fixed Camera Sensor Orientation Relative to UGV (Top View)

Figure 1 shows the change of orientation experienced by a camera sensor fixed on the UGV as the UGV maneuvers along a given track. If the camera sensor is given the capability to move independent of the UGV, a testbed consisting of both a UGV and a two-degree-of-freedom Pan-Tilt Unit (PTU) can be combined to eliminate some of the physical limitations experienced by the UGV alone. If the same lateral shift maneuver is performed with a vehicle-mounted PTU, an end-user viewing the video streaming from the camera sensor would see the image shift smoothly in the direction of the lateral movement. This would be achieved by directing the PTU to compensate for the rotational motion of the UGV. Thus, all the user sees is what an actual spacecraft sees if it is performing a similar maneuver. Coupling this motion compensation with vehicle positions and rates obtained through an attitude simulator, the combined testbed will have the ability simulate the motion history of a spacecraft regardless of the UGV movement underneath.

Figure 2 shows a planar spacecraft translating forward while performing a yaw maneuver. By combining the

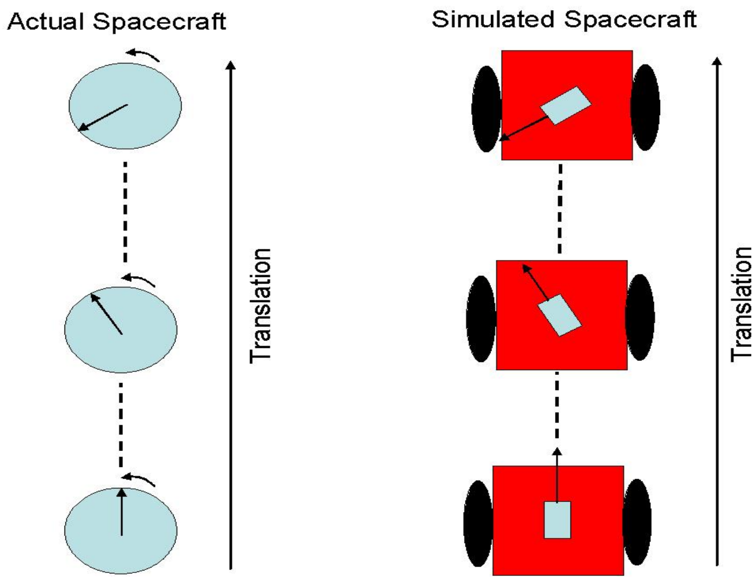

Figure 2: Illustration of Simulation Benefits from Combined Hardware Testbed (Top View)

simulation capabilities of the Pioneer UGV with the extra degrees-of-freedom offered by the PTU, the resulting testbed can be used for a larger variety of planar simulation possibilities.

The UGV employed by the AVS Lab is the ActivMedia Robotics Pioneer 3-DX, and the PTU is a Directed Perception Model PTU-D46, which gives the camera sensor the ability to move in the yaw and pitch directions relative to the vehicle. The addition of pitch control gives the testbed an out-of-plane angular degree of freedom.

To allow the camera to operate completely independent of the vehicle's movements, a control law is derived which is able to steer the PTU and add to the realism of the UGV testbed for spacecraft simulation. This control law is velocity based because the PTU accepts angular rate commands. A feedback is placed on the position error between the measured headings of both the UGV and the PTU to close the control loop.

The functionality of this control law is twofold: first, it compensates for the motion of the UGV, such that its rotational movement does not affect the heading of the camera sensor. The second functionality adds simulated spacecraft rates and orientations such that the camera is mimicking the virtual spacecraft orientation in addition to UGV motion compensation. This is accomplished by simultaneously integrating (in real time or with a fixed time step) Euler's rotational equations of motion and the spacecraft kinematic differential equations of motion. Each integration step produces positions and rates (extracted in Euler 3-2-1 angles) to be fed to the PTU control law. Together with the motion compensation terms and the rates and positions coming in from the virtual spacecraft simulator, the video from the camera sensor appears as if it is mounted on a spacecraft whose motion is prescribed by these rotational equations of motion.

II. Control Development

A. Pan-Tilt Unit Description

Because the PTU is mounted on top of the UGV, it will be rotating the camera sensor with respect to the UGV while the UGV simultaneously rotates underneath. Small errors can occur if the camera sensor experiences translation relative to the UGV while moving throughout the pan and tilt domains. The first pan-tilt unit employed by the AVS Lab is mounted on the UGV such that its pan axis is aligned with the center of rotation of the UGV. This PTU is designed in such a way that the (approximate) focal point of the camera is mounted on the pan axis of rotation.

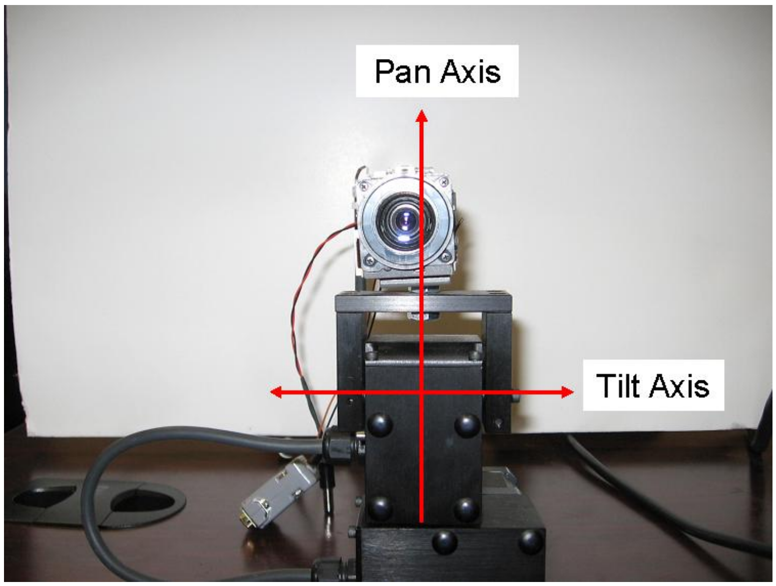

Figure 3: Original PTU Showing Intersection of Pan and Tilt Axes

The tilt axis for this PTU is not aligned with the longitudinal center of the camera, but rotates two struts supporting the plate upon which the camera is mounted. Figure 3 illustrates the physical relationship between the pan axis and the tilt axis on the PTU. The intersection of the two axes of rotation occurs inside the unit itself. For the camera sensor to experience near-zero translation, the focal point must be mounted as close to the intersection of these axes as possible. The translation associated with moving throughout the pan domain with this PTU is negligible due to the alignment of the vehicle rotation axis and PTU pan axis. The amount of translation experienced by the camera sensor in the tilt domain may exceed the sensitivity required by some simulations. Therefore the AVS Lab obtained a reconfigured PTU (Figure 4) where the intersection of the pan and tilt axes is located in a position permissible for camera sensor placement. This new configuration offsets the stock mount for the optical device away from the pan axis of the PTU, but this offset allows for custom hardware to place the focal point of the camera at the intersection of these axes.

Without custom mounting brackets, the focal point of a camera attached to the stock mount of this PTU will not lie on either the pan axis or the tilt axis. The focal point of this camera will translate while operating in both the pan and tilt domains. Due to the unwanted translation in both degrees of freedom, the initial impression is that this PTU is

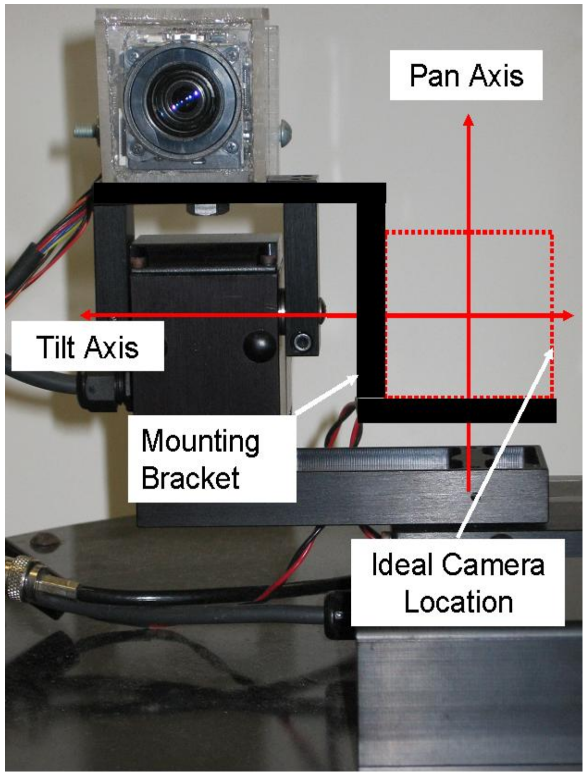

Figure 4: Modified PTU Showing Ideal Camera Location

less optimal than the first. A customized mounting bracket, shown in Figure 4, will place the focal point of the camera sensor at the intersection of the pan and tilt axes, nearly eliminating any pan or tilt induced focal point translation.

With the PTU installed on the Pioneer, a suitable control law can be developed to simulate some user-defined motion. An ocean-going vessel, airplane in flight, autonomous underwater vehicle traversing a pre-programmed track, or a spacecraft in orbit are all examples of systems that could possibly be simulated to some degree by the hardware testbed. The control law should combine UGV motion compensation with results from the spacecraft attitude simulator to allow the combined system of the PTU and UGV to simulate the desired motion regardless of the behavior of the UGV. This project simulates simple planar motion of a spacecraft with the combined movements of the PTU and UGV.

B. Coordinate Systems

In order to define the angles that are used in the control development, a coordinate system must first be defined. Figure 5 shows the Pioneer UGV and PTU with the chosen body axes orientation. Under this convention, a positive yaw is left, a positive pitch is down, and a positive roll is right. The UGV heading is measured by on board wheel encoders. The PTU employs a similar method of obtaining its heading relative to its fixed base. The UGV defines zero heading as the direction it is facing when it is powered on, but the PTU has a set zero heading despite its orientation at startup.

For this project the inertial frame is aligned with the UGV body frame at vehicle startup. Therefore, the heading

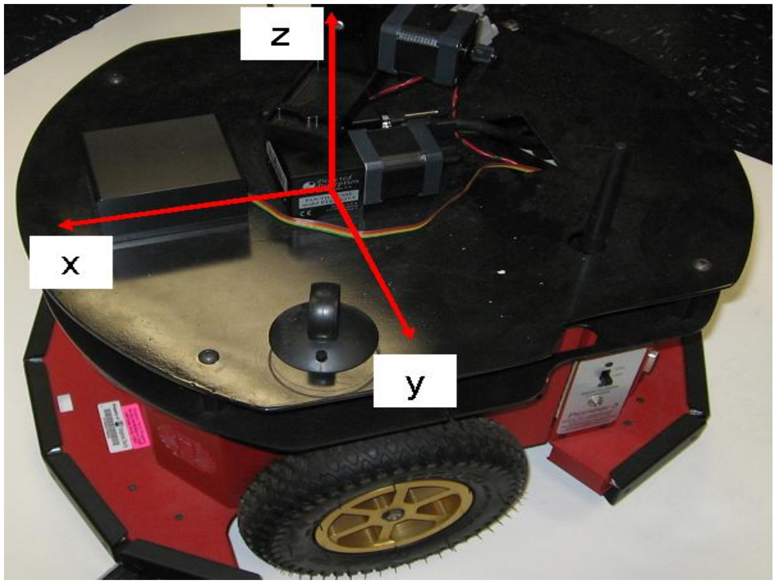

Figure 5: Body Axes of the Pioneer UGV

of the vehicle (U) with respect to the inertial frame (N) is defined as $\theta_{U/N}$.

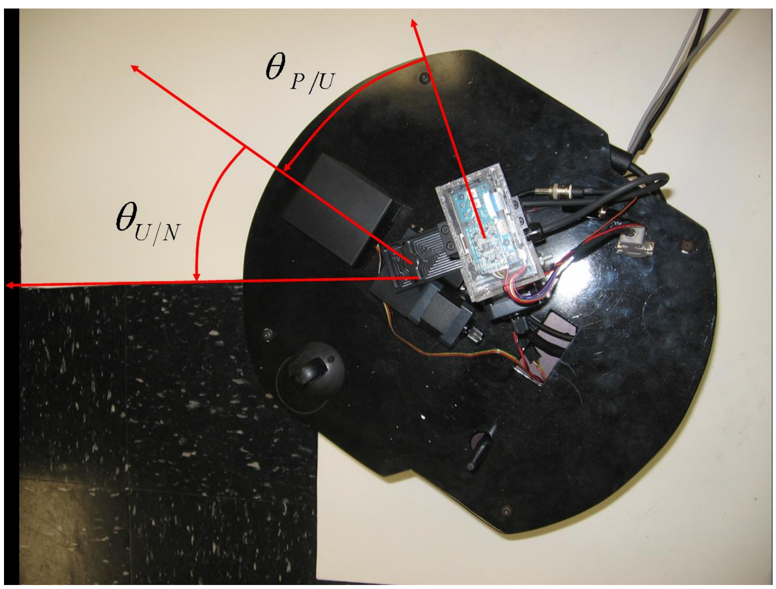

Figure 6: Definition of Angles used in Control Law (Top View)

Since the PTU is mounted to the vehicle, the internal position measurement system employed by the PTU will give the PTU heading (P) with respect to the vehicle, written as $\theta_{P/U}$. Both of these angles are shown in Figure 6. The equation relating the PTU heading to the inertial heading is then expressed as

$$\theta_{P/N} = \theta_{P/U} + \theta_{U/N} \tag{1}$$

where both $\theta_{U/N}$ and $\theta_{P/U}$ are measured quantities.

C. Pan-Tilt Feedback Control Development

The angular rates may similarly be expressed as

$$\dot{\theta}_{P/N} = \dot{\theta}_{P/U} + \dot{\theta}_{U/N} \tag{2}$$

where $\dot{\theta}_{U/N}$ is the measured rotation rate of the UGV and $\dot{\theta}_{P/U}$ is the control that is being derived. Any angle or rate with a superscript of "$*$" is defined as a commanded quantity, or a value produced by the dynamics simulator.

Now, define $\delta\theta$ as the difference between the actual heading $(\theta_{P/N})$ of the PTU with respect to the inertial frame and the commanded position $(\theta^*_{P/N})$ of the PTU with respect to the inertial frame (yaw rate input from the attitude dynamics simulator):

$$\delta\theta = \theta_{P/N} - \theta^*_{P/N} \tag{3}$$

Similarly, the corresponding rate equation $\delta\dot{\theta}$ is expressed as:

$$\delta\dot{\theta} = \dot{\theta}_{P/N} - \dot{\theta}^*_{P/N} \tag{4}$$

where $\dot{\theta}^*_{P/N}$ is defined as the yaw rate input from the attitude dynamics simulator. The stable closed-loop dynamics for a first-order system can be defined as:

$$\delta\dot{\theta} + K\delta\theta = 0 \tag{5}$$

where K is a positive user-defined gain placed on the position feedback, closing the control loop. Any errors will exponentially decay to zero. Substituting Equations 3 and 4 into Equation 5, the following is obtained:

$$\left(\dot{\theta}_{P/N} - \dot{\theta}^*_{P/N}\right) + K\left(\theta_{P/N} - \theta^*_{P/N}\right) = 0 \tag{6}$$

Expanding Equation 6 with Equations 1 and 2 yields:

$$\left(\dot{\theta}_{P/U} + \dot{\theta}_{U/N} - \dot{\theta}^*_{P/N}\right) + K\left(\theta_{P/U} + \theta_{U/N} - \theta^*_{P/N}\right) = 0 \tag{7}$$

Solving for the control input $\dot{\theta}_{P/U}$ yields the steering law of the system:

$$\dot{\theta}_{P/U} = \dot{\theta}^*_{P/N} - \dot{\theta}_{U/N} - K\left(\theta_{P/U} + \theta_{U/N} - \theta^*_{P/N}\right) \tag{8}$$

The rates that appear in the control law include: the commanded inertial rate produced by the dynamics simulator

($\dot{\theta}^*_{P/N}$) and the negative of the instantaneous rotation rate of the vehicle ($-\dot{\theta}_{U/N}$), (this is the motion compensation term). The terms in the position feedback include the measured positions of both the UGV with respect to inertial ($\theta_{U/N}$), the PTU with respect to the UGV ($\theta_{P/U}$), and the desired position produced by the dynamics simulator ($\theta^*_{P/N}$).

A similar process can be followed to derive a closed loop control law for tilt. Noting that the Pioneer has no way of measuring or controlling its pitch relative to the inertial frame, the equations developed for the control law are simplified, and the end result control law in the tilt direction comes to

$$\dot{\varphi}_{P/U} = \dot{\varphi}^*_{P/N} - K\left(\varphi_{P/U} - \varphi^*_{P/N}\right) \tag{9}$$

where each term in Equation 9 is defined like its pan domain counterpart in Equation 8.

Both closed loop control laws are linear and stable, assuming a positive gain is chosen. This statement can also be illustrated through a Lyapunov stability analysis. Because Equation 5 is of the same form as a linear spring system, a candidate Lyapunov function can be defined as:

$$V = \frac{1}{2}\delta\theta^2 \tag{10}$$

where V is a function of $\delta\theta$ rather than just position alone. Differentiating Equation 10

$$\dot{V} = \delta\theta\delta\dot{\theta} \tag{11}$$

is obtained. Substituting Equation 5 into Equation 11

$$\dot{V} = -K\delta\theta^2 \tag{12}$$

Assuming a positive gain, it is obvious that Equation 12 is negative definite over $\delta\theta$. With this condition satisfied, it is proven that this system is asymptotically stable in pan.[1] Repeating this calculation in pitch yields the same results. Therefore, this system is asymptotically stable in both the pan and tilt directions.

III. Spacecraft Attitude Simulation Development

As discussed before, the physical constraints on the UGV impose certain limitations on its simulation capacity. This project requires the development of a PTU steering law that enables the hardware testbed to expand its simulation capabilities to enable the PTU/UGV system to mimic spacecraft motion histories calculated from an attitude simulator. While a PTU motion compensation steering law enables the combined testbed to simulate certain maneuvers otherwise impossible for the UGV alone, the added degrees of freedom the PTU provides the system make it possible to further expand the capabilities of this steering law. An attitude dynamics simulator is developed in UMBRA that provides the steering law with the additional information necessary to mimic spacecraft behavior (in this case, the Euler 3-2-1 positions and rates). UMBRA is a software framework developed at Sandia National Laboratories and allows C++ code modules to be interacted with in real time. This modularization greatly simplifies the sharing of data between components and also allows for nearly flawless interactions between hardware and software simulations.[2] The use of Tcl/Tk startup scripts enable the user to link various input and output connectors between modules as well as to define values for certain variables without hard-coding the changes. The simulator uses a simple Euler integration algorithm with ties to the real-time clock in UMBRA (for the time step in the Euler integration) to integrate the equations of motion for a spacecraft. For the initial development and testing, Modified Rodrigues Parameters (MRPs) are chosen as the attitude description of choice. Of the many attitude descriptions that exist, MRPs have been singled out for their ability to avoid singularities altogether.

MRPs are related to the Euler Parameters, which are also referred to as quaternions. The quaternion vector is defined as:

$$\begin{aligned} \beta_0 &= \cos(\Phi/2) \\ \beta_1 &= e_1 \sin(\Phi/2) \\ \beta_2 &= e_2 \sin(\Phi/2) \\ \beta_3 &= e_3 \sin(\Phi/2) \end{aligned} \tag{13}$$

where Φ is equal to the principal rotation angle and $e_i \quad i = 1, 2, 3$ are the components of the principal axis.[1]

While the quaternions do not encounter a singularity, their four-parameter description requires the constant enforcement of a unit magnitude constraint. This constraint can be geometrically thought of as a four dimensional unit sphere whose four axes are orthogonal to each other.[3] Any valid attitude will fall on the surface of this hyper sphere. One very important feature of the quaternion hyper sphere is that for any attitude described by a point on the surface on the sphere, the opposite pole of that point on the sphere also describes the exact same orientation. One of these points describes the short rotation to that attitude and the "shadow set" describes the long way around to that attitude.[1]

The MRPs are a three element set of stereographic orientation parameters derived from projections from the Euler

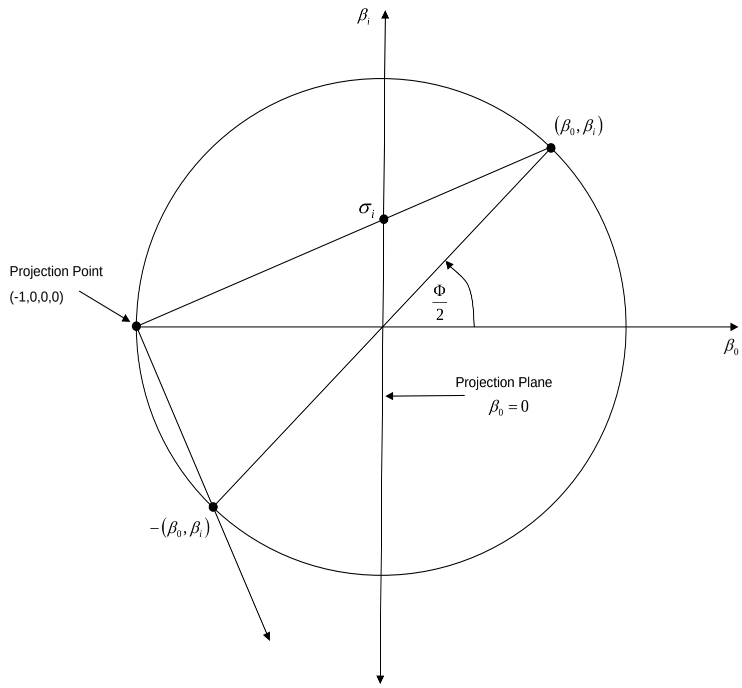

Figure 7: Modified Rodrigues Parameter Projection[1]

parameter hyper sphere onto a three-dimensional hyperplane. Figure 7 illustrates this projection. Upon inspection of the figure, it is obvious that the MRPs encounter a singularity at $\pm 360°$. The shadow set, however, encounters its singularity at $0°$. Since the original MRPs and their corresponding shadow sets describe the same orientation, the fact that their respective singularities occur at different angles is of great advantage. By switching the attitude representation between the original and shadow sets whenever the current set being used approaches a singularity, the attitude of the body will never reach a singularity.[1] The transformation between MRPs and Euler parameters is shown below.

$$\boldsymbol{\sigma}_i = \frac{\beta_i}{1+\beta_0} \quad i = 1, 2, 3 \tag{14}$$

The shadow set of MRPs is found by taking the negative of the Euler parameters.

$$\boldsymbol{\sigma}_i^S = \frac{-\beta_i}{1-\beta_0} \quad i = 1, 2, 3 \tag{15}$$

Despite all of the benefits from using MRPs as the attitude representation of the system, it should be noted that 3-2-1 Euler angles lend themselves quite well to describing the PTU hardware configuration of the testbed. With yaw representing PTU pan, pitch representing PTU tilt, and roll available for future servo upgrades to the camera system, the 3-2-1 Euler angles make the most sense with regard to a physical description of the testbed. They do, however, exhibit significant drawbacks with regard to singularities. Symmetric sets of Euler angles experience singularities when θ_2 is $0°$ or $180°$. Asymmetric sets, like the 3-2-1 set, encounter their singularities whenever θ_2 is $\pm 90°$. In order to avoid singularities, the dynamics simulator will integrate the MRP kinematic differential equation shown below:

$$\dot{\boldsymbol{\sigma}} = \frac{1}{4}\left[\left(1-\sigma^2\right)\left[I_{3\times 3}\right] + 2\left[\tilde{\boldsymbol{\sigma}}\right] + 2\boldsymbol{\sigma}\boldsymbol{\sigma}^T\right]\boldsymbol{\omega} \tag{16}$$

The Euler rotational equations of motion

$$[I]\,\dot{\boldsymbol{\omega}} = -\left[\tilde{\boldsymbol{\omega}}\right][I]\,\boldsymbol{\omega} + \boldsymbol{L}_C \tag{17}$$

are simultaneously integrated with the kinematic differential equations in order to solve for the resulting motion of the body. After each integration step, the 3-2-1 Euler angles are extracted from the rotation matrix computed as a function of the MRP vector.

$$[C] = \left[I_{3\times 3}\right] + \frac{8\left[\tilde{\boldsymbol{\sigma}}\right]^2 - 4\left(1-\sigma^2\right)\left[\tilde{\boldsymbol{\sigma}}\right]}{\left(1+\sigma^2\right)^2} \tag{18}$$

Even though the algorithm in the dynamics simulator is able to avoid singularities for all time during the simulation, the conversion to the 3-2-1 Euler angles after each integration step reintroduces the θ_2 is $\pm 90°$ singularity into the system. The physical limits on the PTU bound the camera sensor to approximately ± 30 degrees, preventing the system from physically reaching the singularity.

IV. Implementation

The UMBRA framework is used to implement the control law and the spacecraft motion simulator into the Pioneer UGV. A MATLAB script is developed to verify the results produced by the dynamics simulator. Two UMBRA modules are developed; one to simulate the dynamics of a spacecraft, the other to implement the PTU steering law.

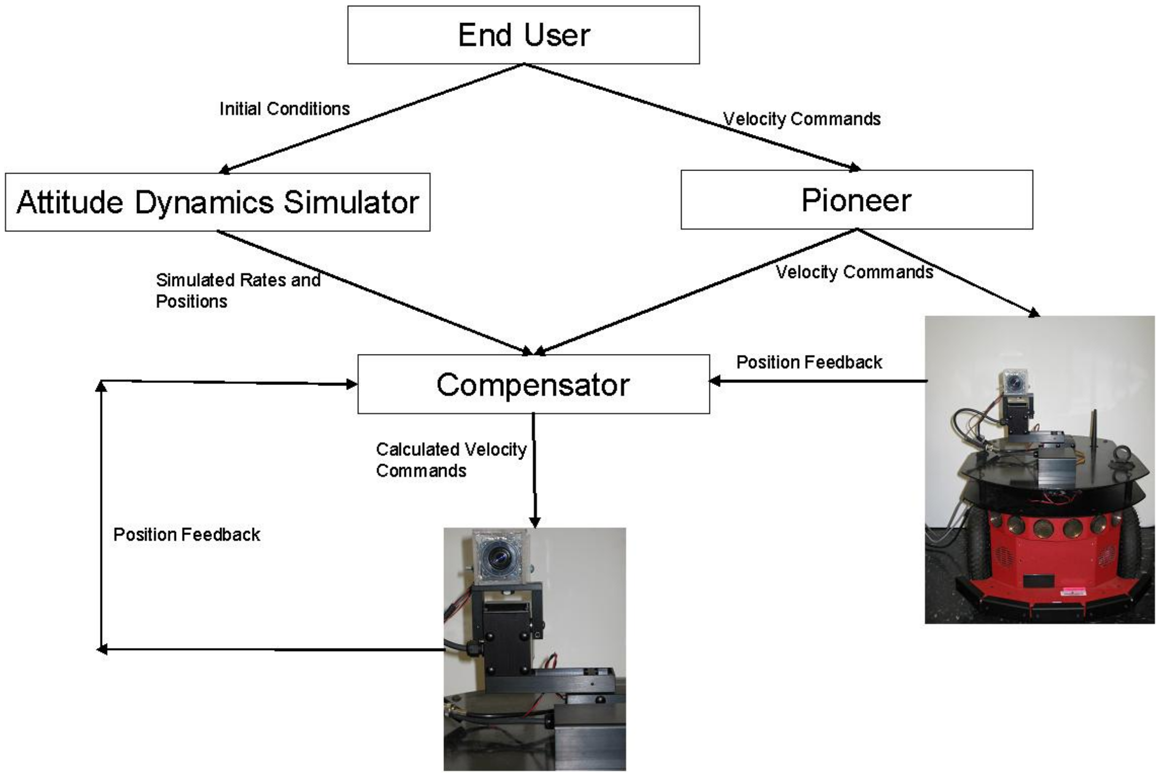

Figure 8: Block Diagram Illustrating Logic Flow

Figure 8 shows a simple flowchart to illustrate the interaction between the modules and their input/output connectors. The dynamics simulator requires the initial MRP vector, initial body angular velocity vector, and choice of attitude description to be set in the startup tcl script. The external torque vector found in Euler's rotational equations of motion is currently set as an input connector to allow for other simulations to connect their torque output directly to the dynamics simulator. The UMBRA real-time clock is also connected to the input of the dynamics simulator in order to obtain a real-time, slightly varying time step to use in the integration. At the end of each integration step, the motion compensator calculates the Euler 3-2-1 angles and their corresponding rates which are set as output connectors and sent to the compensator module. The body angular velocity vector, ω, is also available as an output connector. The Euler angle output is a default for this system, but a few extra lines of code enables the user to obtain the attitude description in whatever coordinate set they choose.

The primary function of the Pioneer module (developed by Mark Monda[2]) is to control and monitor the various

Pioneer UGV states through various modes and functions. To run the motion compensator, a manual setting in the Pioneer module is changed (the UGV is placed in *unguarded* mode) and the driving left/right velocities sent to the Pioneer are also connected to the compensator module where they are negated and used in the control law.

The motion compensator module, which houses the control law derived in Equation 8, takes the Pioneer UGV measured position, the driving velocities sent to the UGV, the positions and rates calculated by the dynamics simulator, and the measured position of the PTU as input connectors. To run a simulation in UMBRA, a startup script is written in tcl/tk (Tool Command Language) which sets initial conditions as well as initializes the modules required for the simulation. To run the steering law simulation, the tcl script sets the gain, the initial MRP vector, the initial body angular rate vector, the body inertia matrix, and the output mode. This script engages the attitude dynamics simulator module as well as the Pioneer module and the compensator module. The attitude dynamics simulator module sets output connectors of Euler 3-2-1 angles and rates, which are connected to the input connectors of the compensator module. Currently, the only output connectors employed by the compensator module are the pan and tilt rates necessary to drive the PTU to achieve the desired simulation.

V. Results

Multiple test cases are presented to demonstrate the various features of the steering law.

A. Pure Motion Compensation

First, simple motion compensation is demonstrated with no contribution from the spacecraft attitude dynamics simulator.

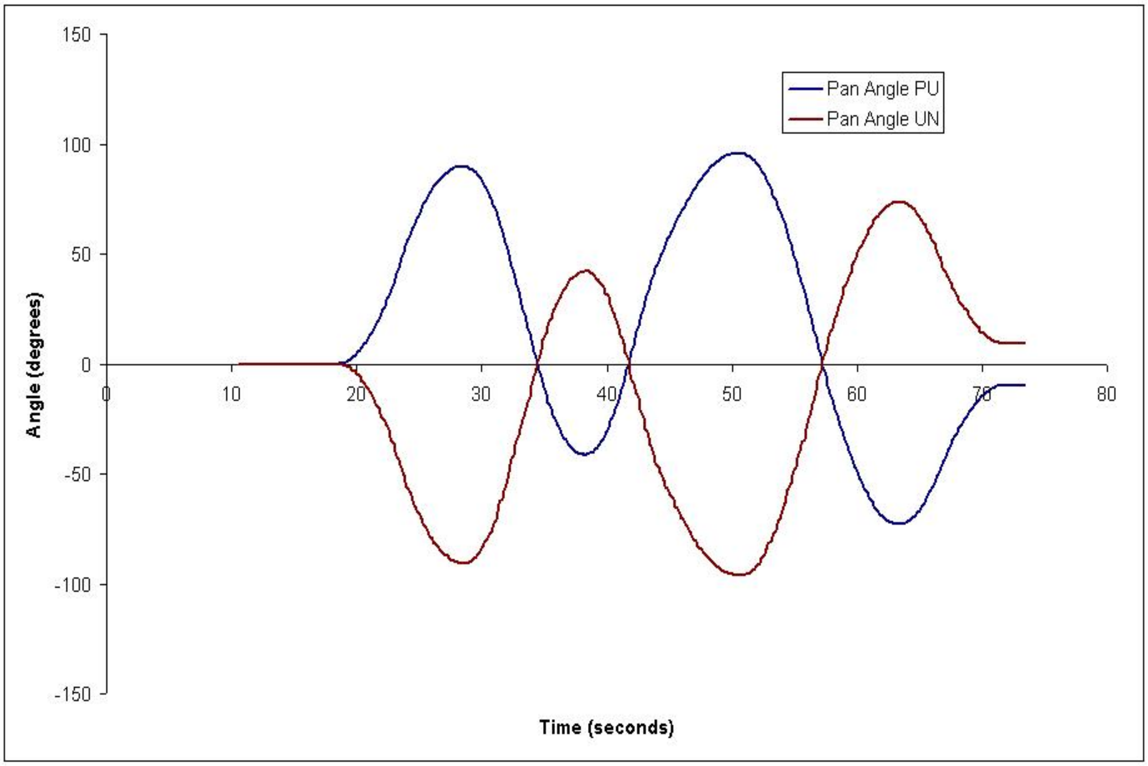

Figure 9: Motion Compensation Performance

Figure 9 illustrates the time histories of the heading angles of the UGV relative to the inertial frame and the PTU relative to the UGV. For pure motion compensation the sum of these two angles is zero, indicating no movement of the camera sensor relative to the inertial frame.

Figure 9 indicates that the PTU and UGV are moving in opposite directions simultaneously, and the error plot in Figure 10 illustrates that the sum of the two motion histories is not exactly zero. An interesting observation stemming from Figure 10 is that the error magnitude seems to be largest whenever the Pioneer and PTU are rotating the fastest (i.e., whenever the slope of the heading angles are greatest). The magnitude of the error is averaged to be 0.531 degrees.

B. Attitude Simulation in Pan with Stationary UGV

An initial condition of $\boldsymbol{\omega} = \begin{pmatrix} 0 & 0 & 0.1 \end{pmatrix}^T$ radians per second is defined in the dynamics simulator module and run with a stationary UGV to test the PTU alone at a constant angular velocity in the pan domain. The initial MRP

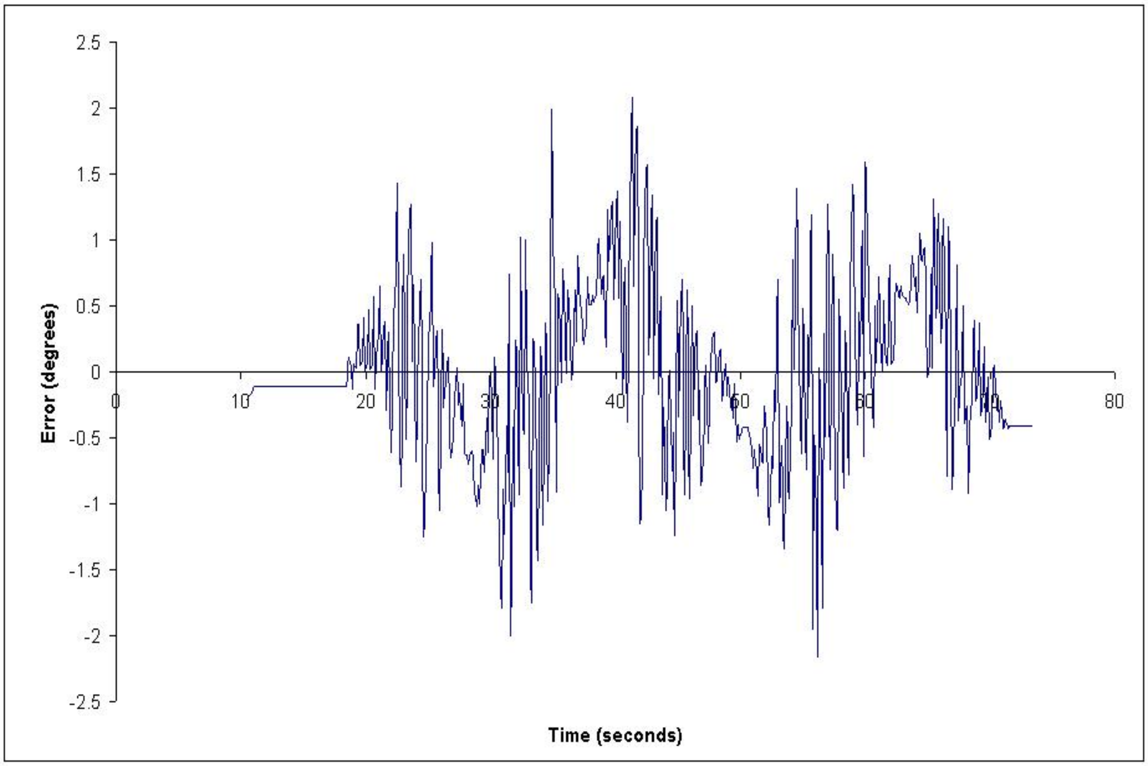

Figure 10: Error Observed During Motion Compensation

vector is defined as the origin $\boldsymbol{\sigma} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$, the constant gain is set to $K = 0.78$, and the inertia matrix of the simulated body is set to the identity matrix.

This test confirmed the close agreement of results from both the MATLAB simulation and the C++ simulation in UMBRA. In order to make this comparison as accurate as possible, the real-time time steps used in the UMBRA simulation were obtained and averaged to about 0.126 seconds, which was used as the constant time step in the MATLAB simulation. Figure 11 shows the actual pan angle measured by the PTU to be in very close agreement with the simulated values calculated by both the UMBRA attitude dynamics module and the MATLAB simulator.

The error shown in Figure 12 illustrates the precision in which the PTU tracks the simulated dynamics. The initial spike of just over -1.4 degrees stems from the fact that the servo motors on the PTU do not have an infinitely fast response time, so it takes a small amount of time for the PTU to accelerate from zero velocity to the velocity calculated by the attitude simulator. The absolute value of the error between the measured position and the expected position ($\delta\theta = \theta_{P/U} + \theta_{U/N} - \theta^*_{P/N}$) averages to 0.042 degrees.

C. Attitude Simulation in Pan with Simultaneous UGV Movement

The same test is run again with identical initial conditions, but this time a varying rotational motion is given to the UGV. Figure 13 illustrates the separate motion histories of the UGV and the PTU, as well as the tracks of the camera sensor, UMBRA simulation, and MATLAB simulation. The data show close agreement between the hardware testbed

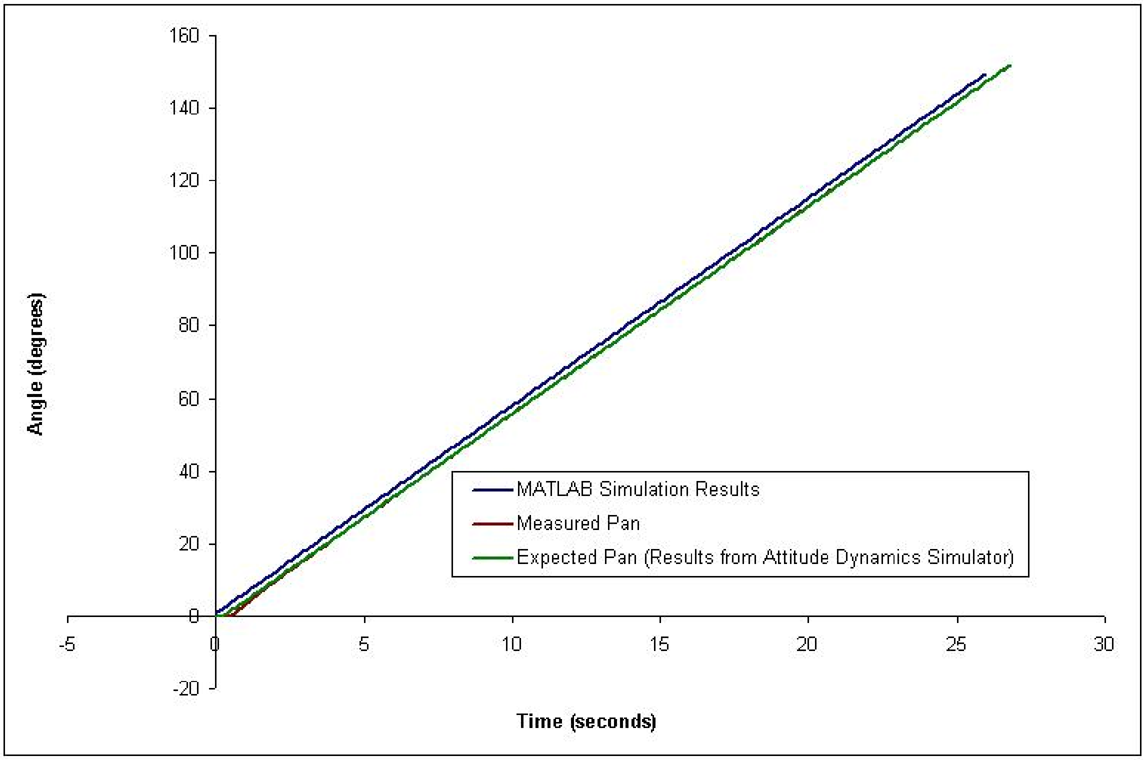

Figure 11: Simulation Performance with Stationary UGV and Constant Pan Velocity

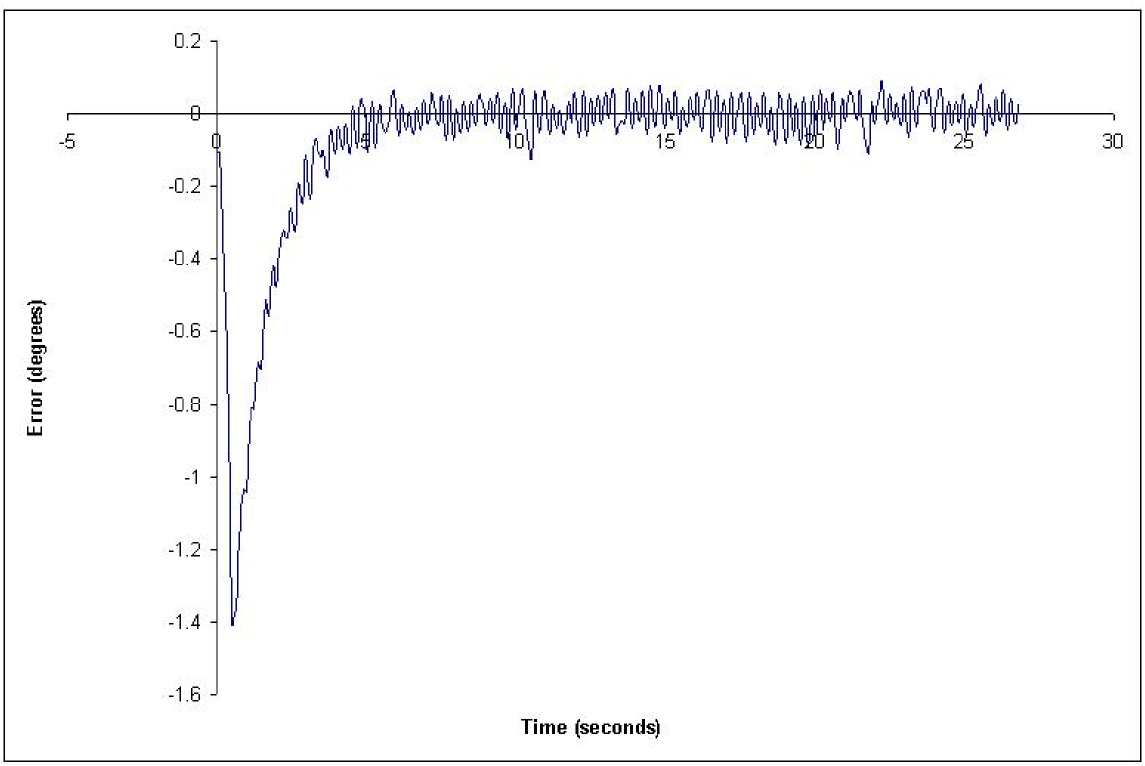

Figure 12: Position Error $\delta\theta$, UGV Stationary

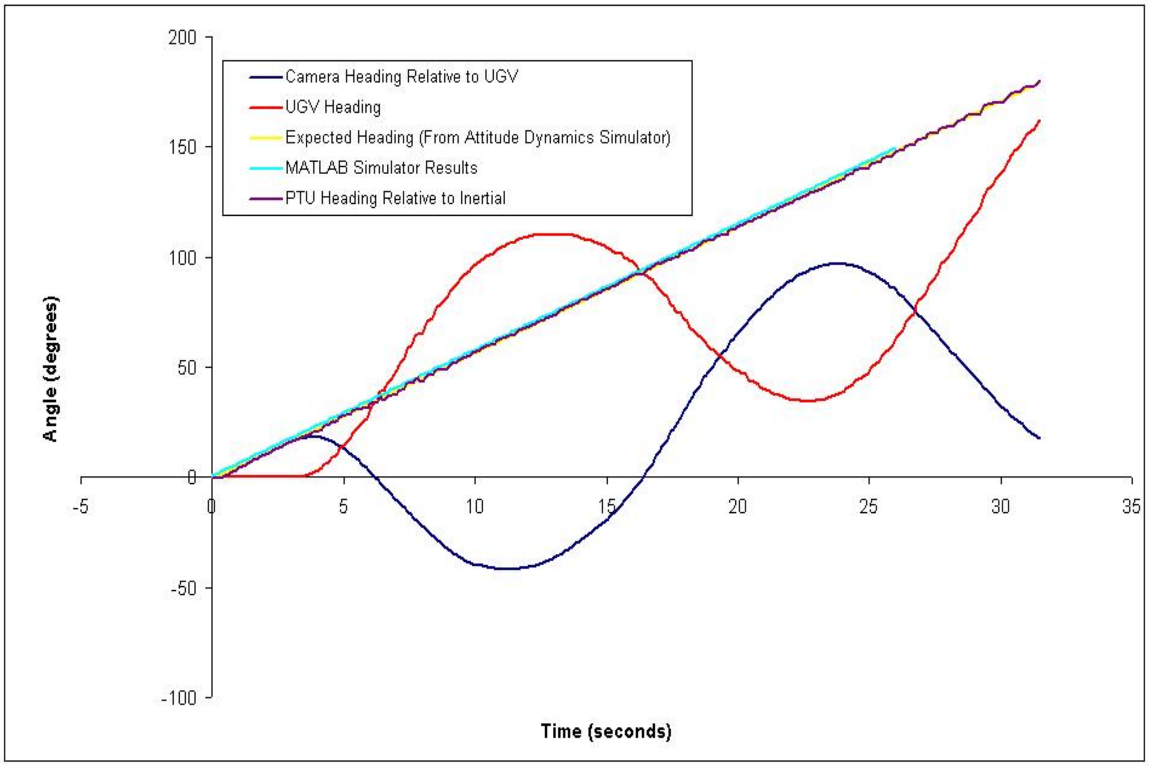

Figure 13: Simulation Performance with Rotating UGV and Constant Angular Velocity

and the numeric simulation results, even with the UGV rotating underneath.

The absolute value of the error $\delta\theta$ for the non-stationary UGV test case shown in Figure 14 averages to a value of 0.529 degrees, about an order of magnitude greater than the stationary UGV test. Again, it is of particular interest that the magnitude of the error values are greatest whenever the velocities of the PTU and UGV are highest.

D. Attitude Simulation in Tilt with Stationary UGV

An initial condition of $\boldsymbol{\omega} = \begin{pmatrix} 0 & 0.06 & 0 \end{pmatrix}^T$ radians per second is defined in the dynamics simulator module and run with a stationary UGV to test the PTU alone at a constant angular velocity in the tilt domain. Again, the initial MRP vector is defined as the origin $\boldsymbol{\sigma} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$, the constant gain is set to $K = 0.78$, and the inertia matrix of the simulated body is set to the identity matrix.

As with the constant pan rate and stationary UGV, Figure 15 illustrates the close agreement between the MATLAB and UMBRA simulations with the hardware output.

Figure 16 again illustrates the lack of perfect servo motors, but the maximum error incurred by the PTU during the initial ramp-up to the required velocity was less than one degree. The PTU can tilt about 30 degrees in either direction before reaching its physical bounds.

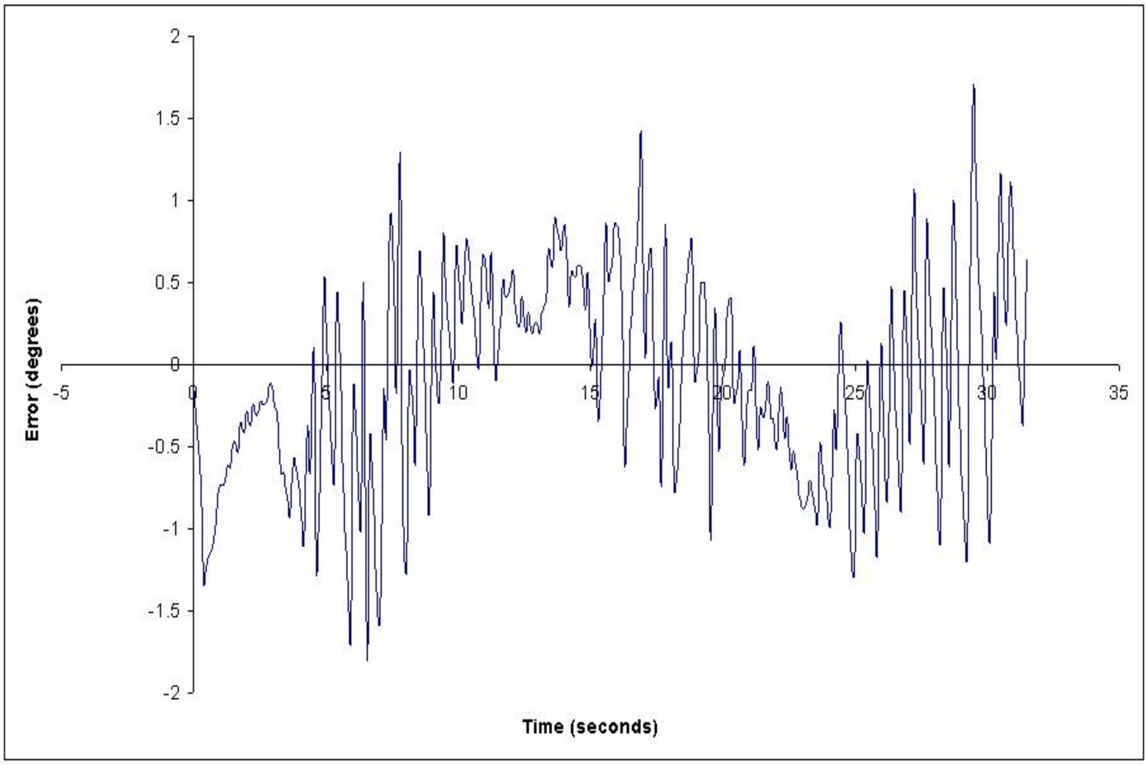

Figure 14: Position Error $\delta\theta$, UGV Rotating

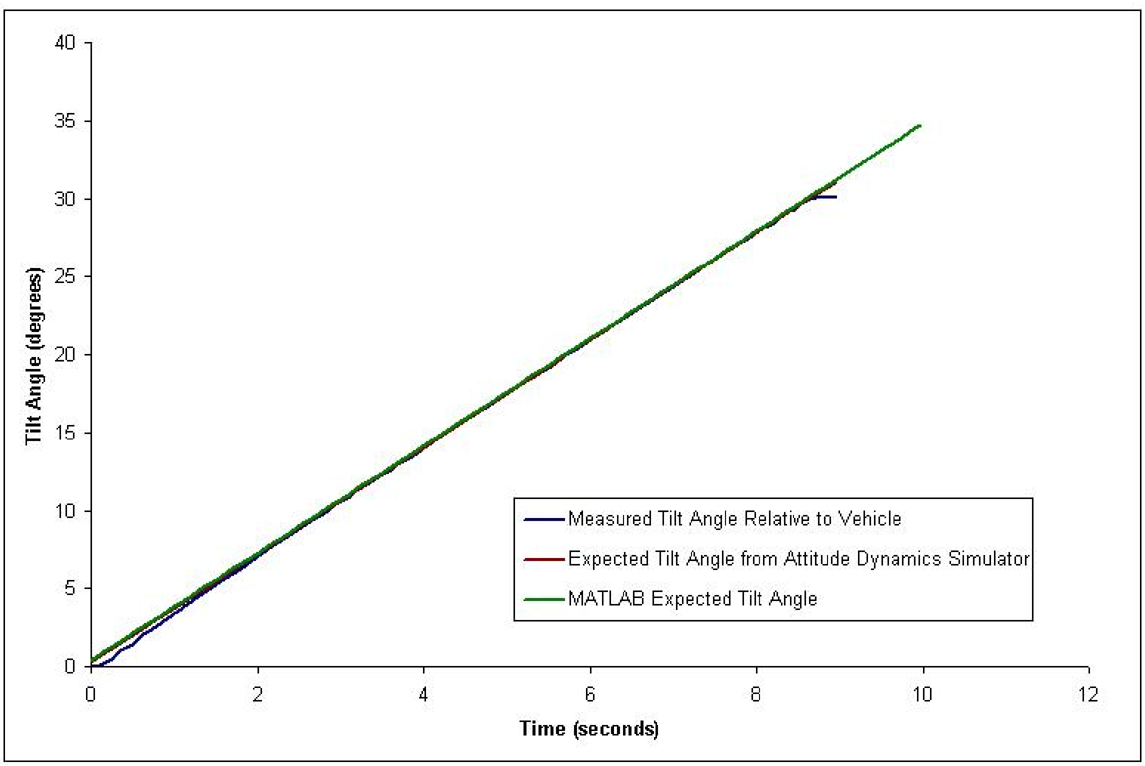

Figure 15: Simulation Performance with Stationary UGV and Constant Tilt Velocity

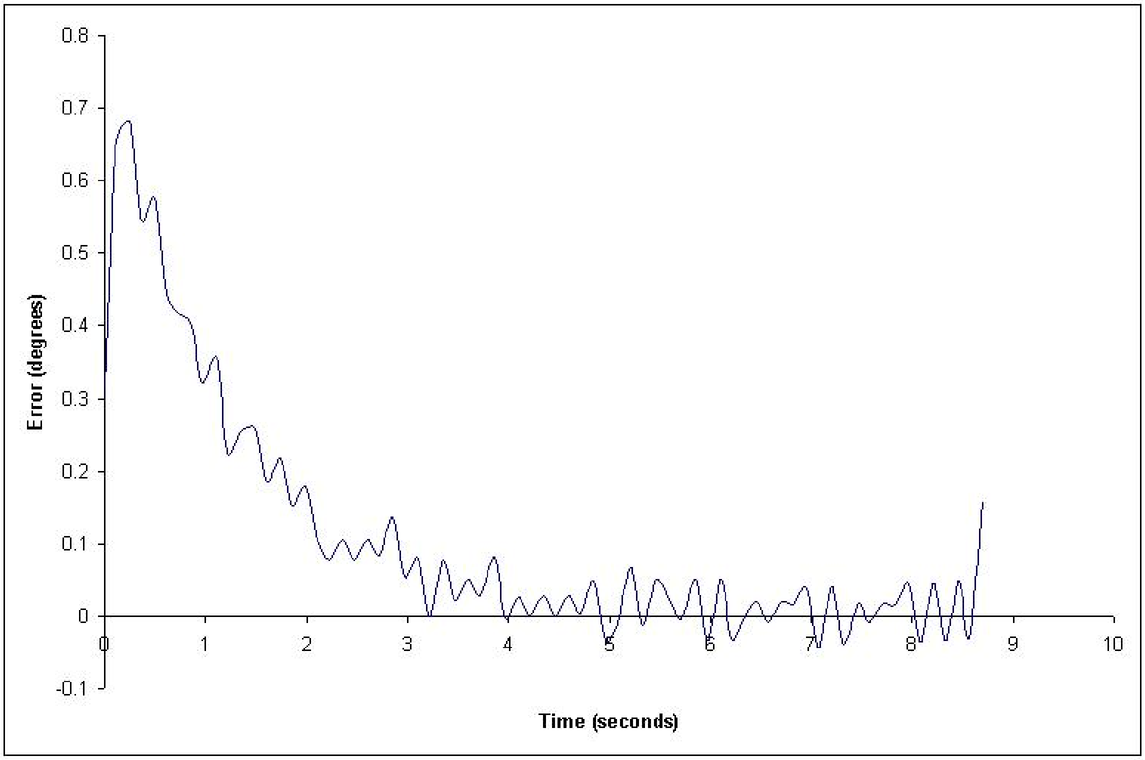

Figure 16: Position Error in Tilt Domain with Stationary UGV

E. Attitude Simulation in Pan and Tilt with Stationary UGV

Finally, a set of tests are run to gauge the performance of the testbed when initial velocities are given for both pan and tilt. An initial condition of $\boldsymbol{\omega} = \begin{pmatrix} 0 & 0.06 & 0.1 \end{pmatrix}^T$ radians per second is defined in the dynamics simulator module and run with a stationary UGV.

Figure 17 shows a very close agreement between the values calculated from the two attitude simulators and the actual positions measured by the UGV and PTU. Looking very closely at the tilt position histories, the measured tilt angle falls slightly below the simulated values around 12 seconds elapsed time. This is due to the fact that the PTU has reached its joint limit on tilt and the simulator continues to produce slightly increasing tilt angles.

The initial conditions were chosen at random, so the fact that the pitch angle increased to slightly above 30 degrees and decreased again is a coincidence. It illustrates the effect of the position feedback on the tilt control law. Figure 18 shows that right around 12 seconds elapsed time, the tilt rate sent to the PTU increased compared to the tilt rates calculated by both simulations. The gain on the position feedback term summed with the tilt rate value in the control law caused the tilt rate sent to the PTU to increase in order to account for the increasing error on position.

The error shown in Figure 19 is bounded to within a quarter of a degree (neglecting servo ramp-up times). The increase in error in the tilt domain occurring at about 12 seconds elapsed time coincides with when the tilt axis reaches its upper bound and the simulated values keep increasing. The pitch values for this simulation peak at a value slightly larger than the physical limit on the PTU, so the error on position reaches a maximum and then decreases again once

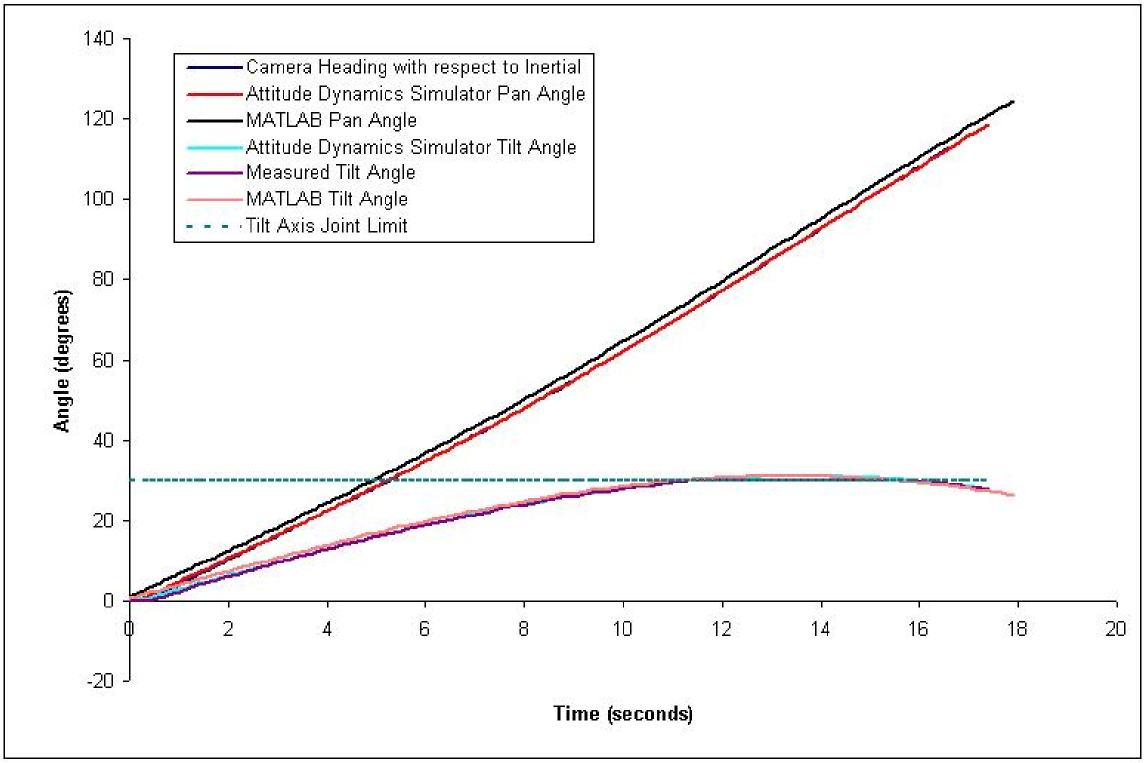

Figure 17: Simulation Performance with Initial Pan and Tilt Velocities, Stationary UGV

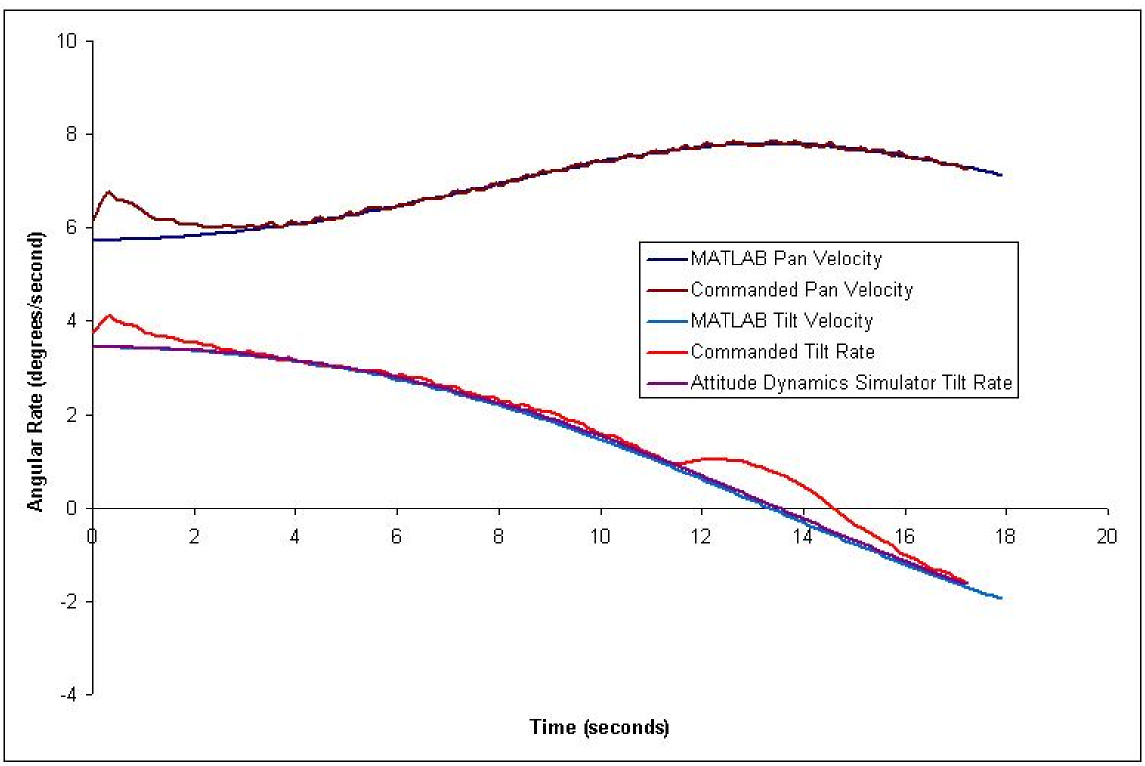

Figure 18: System Rates with Initial Pan and Tilt Velocities, Stationary UGV

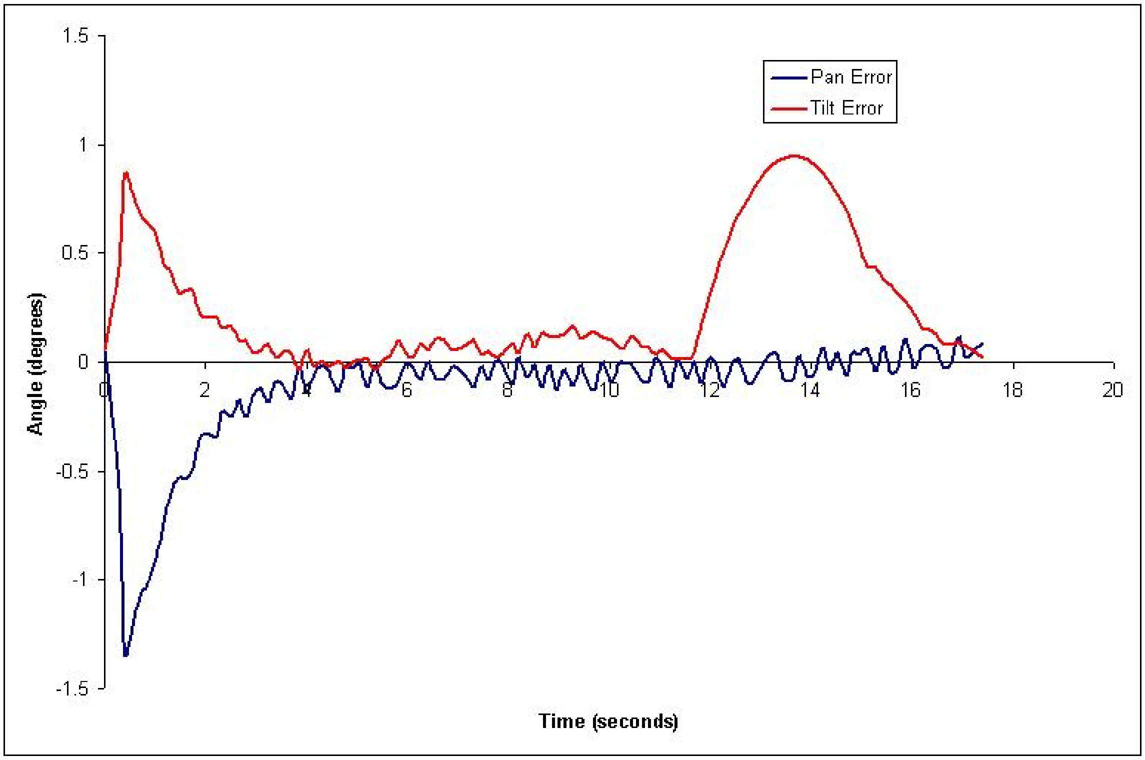

Figure 19: Error in the Pan and Tilt Domains, Stationary UGV

the simulated values re-approach the saturated position of the PTU.

F. Attitude Simulation in Pan and Tilt with Simultaneous UGV Movement

The same test is run with identical initial conditions, but this time with a rotating UGV.

Despite a randomly maneuvering UGV, Figure 20 shows a close correlation between the simulated values and measured values of the pan and tilt angles.

Figure 21 shows the same trend in tilt error as Figure 19. Again, this is due to the physical limit on the PTU tilt axis. The magnitude of the error values encountered in the pan domain has increased from the identical test case with no UGV movement. As expected, the locations where the largest errors occur are when the PTU and UGV have the fastest angular rate. Even so, the magnitude of the error calculated in pan is less than 2.5 degrees.

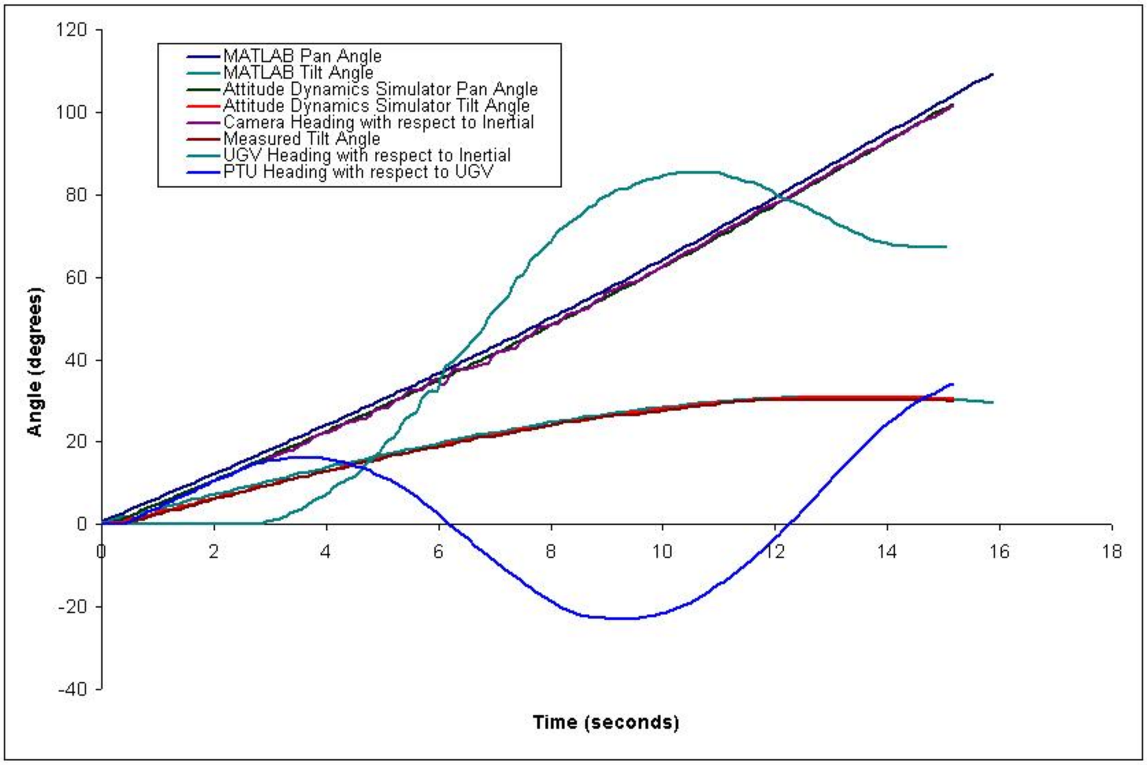

Figure 20: Simulation Performance with Initial Pan and Tilt Velocities, Simultaneous UGV Movement

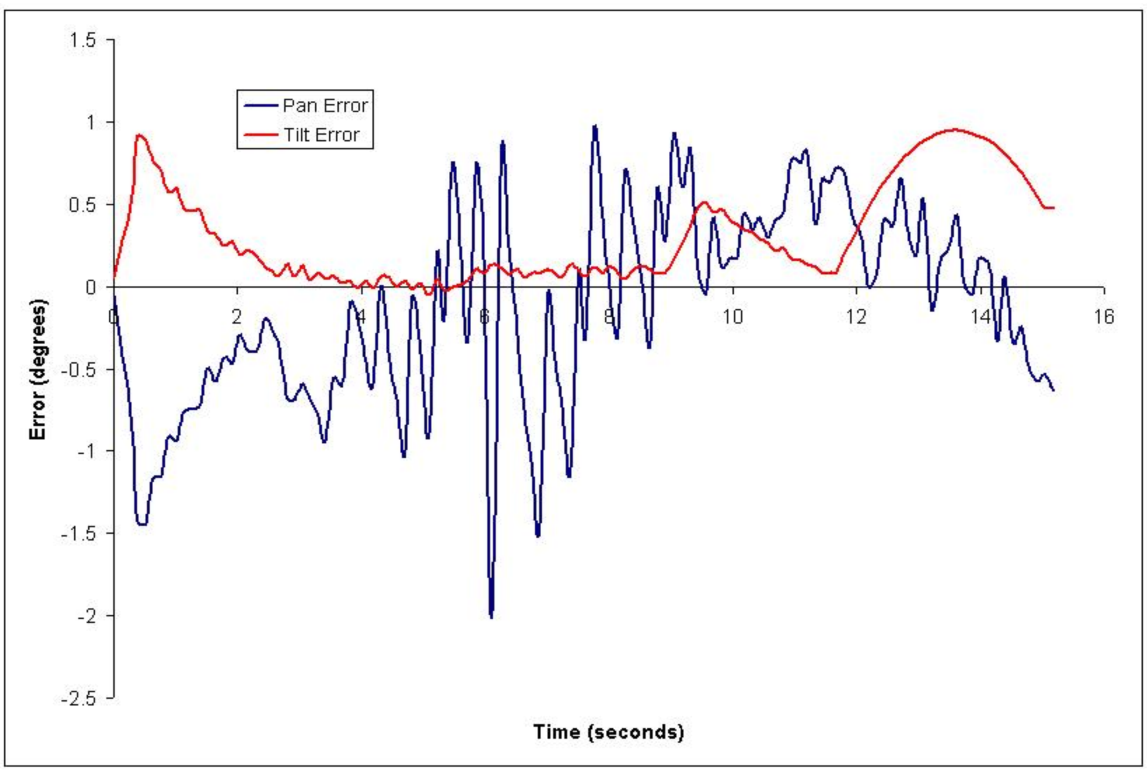

Figure 21: Error in Pan and Tilt Domains, Simultaneous UGV Movement

VI. Conclusions

A linear control law is developed to combine the simulation capabilities of an unmanned ground vehicle with the added degrees of freedom a camera sensor mounted on a pan-and-tilt unit provides. To expand the control law past having the sole functionality of a motion compensator, a spacecraft attitude dynamics simulator is developed that integrates the equations of motion for a spacecraft. These equations of motion can be replaced depending on the system to be simulated. Initial tests indicate the simulation testbed very accurately simulates the desired motion of the virtual spacecraft.

There are a number of ways to make improvements on this system. Another servo motor could be added to the camera sensor giving it the ability to move in the roll domain. Also, a bracket can be fashioned to mount the camera at the intersection of the pan and tilt axes of the PTU, eliminating translation while moving through those domains. An IMU can be integrated into the UGV to enable it to measure its pitch orientation with respect to the inertial frame. All of these improvements enable the AVS Lab simulation equipment to more accurately simulate the behavior of planar spacecraft.

References

[1] Schaub, H. and Junkins, J. L., *Analytical Mechanics of Space Systems*, AIAA Education Series, Reston, VA, 1st ed., 2003.

[2] Monda, M. J., *Hardware Testbed for Relative Navigation of Unmanned Vehicles Using Visual Servoing*, Master's thesis, Virginia Polytechnic Institute and State University, April 2006.

[3] Shrewsbury, D. T., "Specialized Attitude Coordinate Development Utilizing Symmetric Stereographic Orientation Parameters," *Spaceflight Dynamics Semester Project*, April 2006.