Control of the Gyrover

A Single-Wheel Gyroscopically Stabilized Robot

Shu-Jen Tsai Enrique D. Ferreira Christiaan J. J. Paredis

Institute for Complex Engineered Systems
Carnegie Mellon University
Pittsburgh, PA 15213-3890
{stsai,edf,cjp}@ices.cmu.edu

Abstract

The Gyrover is a single wheel gyroscopically stabilized mobile robot developed at Carnegie Mellon University. An internal pendulum serves as a counter weight for a drive motor that causes fore/aft motion, while a tilt-mechanism on a large gyroscope provides a mechanism for lateral actuation. In this paper, we develop a detailed dynamic model for the Gyrover, and use this model in an extended Kalman filter to estimate the complete state. A linearized version of the model is used to develop a state feedback controller. The design methodology is based on a semi-definite programming procedure which optimize the stability region subject to a set of Linear Matrix Inequalities that capture stability and pole placement constraints. Finally, the controller design combined with the extended Kalman filter are verified on the prototype.

1 Introduction

The concept of a single-wheel gyroscopically stabilized robot was originally proposed by Brown and Xu [2, 7]. The idea is to take advantage of the dynamic stability of a single wheel, but augment it with a gyroscope to achieve static stability. The self-stabilizing dynamics of a single wheel can be illustrated as follows. Consider a single wheel rolling down a hill. As soon as the wheel starts leaning laterally, gyroscopic precession will cause it to turn in the direction it is leaning, after which the resulting centrifugal inertial effect will right it again.

Past research on the Gyrover focussed entirely on the mechanical design. After some initial tests to verify the concept, a simplified dynamic model was developed to weigh the different design characteristics [2]: static stability vs. high speed dynamic responsiveness, slope climbing ability, etc. Based on this model, several generations of Gyrovers have been built with gradually increasing sophistication, reliability, and performance.

However, so far no attempt has been made at developing an automatic controller for the Gyrover. The Gyrover has been controlled using a remote rontrol transmitter that allowed the user to control the voltage of the drive motor and the angle of the tilt-mechanism (see Section 2 and Figure 1). Due to the coupling between the fore/aft and lateral motions and the lack of attitude sensing on the Gyrover, the user has to develop a feeling for the dynamics of the robot, estimate its current attitude by visual inspection, and provide the appropriate input commands. Because of the self-stabilizing dynamics of the Gyrover, it is relatively easy for a novice user to keep it from falling over, especially when moving at moderate and high speeds. However, it is much more challenging to track a desired trajectory, and near impossible to control the robot when it is out of sight.

To use the Gyrover for inspection tasks in which fine control in remote locations is required, we need to develop a controller that relieves the user from stability concerns and provides an intuitive control interface. This paper presents the development of such a controller.

We approach the problem in three stages. In the first stage, described in Section 2, we develop a detailed dynamic model of the Gyrover. This model lies at the basis for the subsequent derivations of the state estimator and controller described in detail in Sections 3 and 4. Simulation and experimental data to validate the model and controller are shown in Section 5.

2 Gyrover Dynamics

As shown in Figure 1, the Gyrover consists of four rigid bodies connected to each other through a 3-degree-of-freedom kinematic chain: the wheel, the pendulum, the tilt-mechanism, and the gyroscope. The wheel is the only

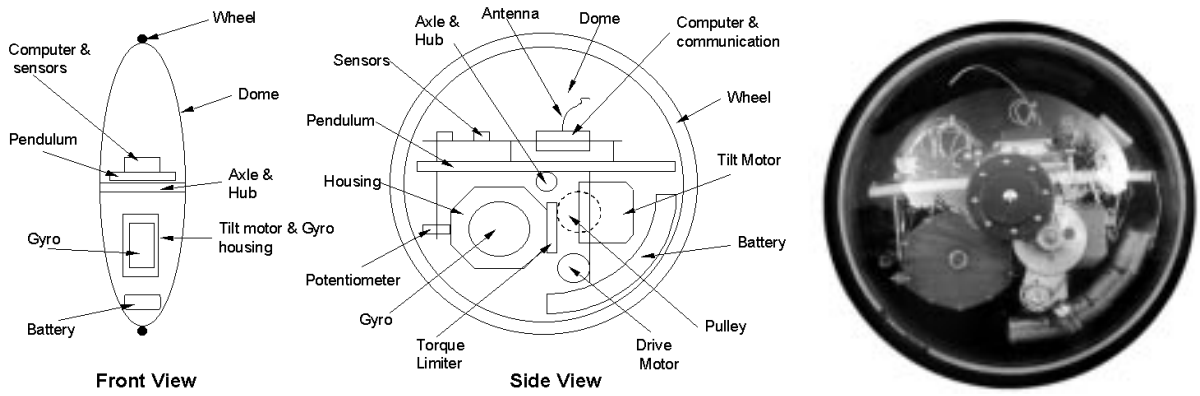


Figure 1: Ccomponent diagram (left) and a side view snapshot (right) of the Gyrover.

body that is in contact with the environment. It consists of a rim and two spherical polycarbonate domes that carry the drive shaft. Around this shaft swings the main body of the Gyrover, called pendulum. The pendulum includes a DC-motor and transmission that drive the wheel shaft. With gravity acting as reaction torque, this drive mechanism generates forward acceleration for the Gyrover. The second degree-of-freedom is formed by the tilt-servo, which articulates the rotation axis of the gyroscope with respect to the pendulum. This rotation axis is perpendicular to the the main drive shaft and is located in the middle of the Gyrover. The torque generated by the tilt-servo causes lateral motion of the wheel and at the same time induces a precessing motion given by the gyroscopic precession equation:

$$\tau \quad = \quad J\omega \times \Omega$$

where τ is the torque acting on the gyroscope, $J\omega$ is its angular momentum, and Ω is the precession rate. For example, when the forward velocity is zero, one can rotate the Gyrover to the left by leaning it slightly to the left. The gyroscopic effect stops the Gyrover from falling over and simultaneously induces a positive rotation around the vertical axis steering the robot to the left.

The final degree-of-freedom is the spin axis of the gyroscope. It is driven by a servo controlled motor that maintains a constant angular velocity of approximately $15,000$ RPM. Because the motor is too small to generate any sudden change in angular velocity, we do not use this degree-of-freedom for control purposes. In the remainder of the paper, we will therefore assume that the angular velocity of the gyroscope is kept constant.

The control of the Gyrover is achieved through the first two degrees-of-freedom: the drive motor, and the tilt servo.

The derivation of the dynamic equations for the Gyrover is based on the Newton-Euler approach [5] Previous derivations of the dynamic equations were based on a Lagrangian approach [7]. However, due to extensive simplifying assumptions, these models are inappropriate for control purposes. In our derivation we make the following assumptions:

- all the components are rigid bodies,
- the wheel rolls without slipping,
- the friction model for the contact between the wheel and the floor, and for the drive motor and transmission includes Coulomb and viscous friction,
- the angular velocity of the gyroscope is constant,
- the wheel and gyroscope are axially symmetric,
- the floor is flat and horizontal,
- the wheel remains in contact with the ground.

Unlike the Newton-Euler dynamics for fixed base manipulators, the Gyrover dynamics cannot be calculated numerically in an iterative fashion. For fixed base manipulators, the acceleration of the base is known and fixed, so that the accelerations of the distal links can be computed sequentially. Once all the accelerations are known, the reaction forces can be computed in an inward iteration from the end-effector towards the base. However, since the accelerations of the wheel of the Gyrover are not fixed but depend on the accelerations of the internal degrees-of-freedom, one cannot evaluate the Newton-Euler equations numerically. Instead, the complete dynamics need to be derived symbolically after which the contact constraints can be imposed. Kinematic and force constraints are both relevant here. Rolling without slippping imposes constraints on the wheel accelerations and torques. With the notation listed in Table 1, the acceleration constraints are:

$$\begin{aligned} \dot{v}_0 &= \dot{w}_0 \times r + \omega_{0f} \times v_0 \\ \omega_{0f} &= \left[\omega_{0x}, \quad \omega_{0y}, \quad \omega_{0y} \frac{cos(\theta_0)}{sin(\theta_0)} \right]^T \end{aligned} \tag{1}$$

Rolling without slipping also imposes a constraint on the

θ_0	Lean angle of the wheel measured between the rotation axis and the vertical.
ω_{0x}	Roll angular velocity.
ω_{0y}	Yaw angular velocity.
ω_{0z}	Pitch angular velocity.
ω_{0f}	Rotational velocity of the wheel frame (this is different from ω_0 because the frame is defined as having its X-axis horizontal; it does not rotate with the wheel)
v_0	Translation velocity of the wheel
θ_1, w_1	Angle and angular velocity of the pendulum with respect to the wheel
θ_2, w_2	Angle and angular velocity of the tilt mechanism with respect to the pendulum.
w_3	Angular velocity of the gyroscope with respect to the tilt mechanism
τ_1	Torque exerted by the drive motor
$\theta_{2,\mathrm{ref}}$	Reference position for the tilt mechanism.
N_x	Contact torque in the global X-axis.
N_y	Contact torque in the global Y-axis.
N_z	Contact torque in the global Z-axis.

Table 1: Description of kinematic and dynamic variables of the Gyrover.

torques acting on the wheel. If there is no friction, the torques exerted onto the wheel at the point of contact, (N_x, N_y, N_z), are zero [6]. In summary, the dynamics of the Gyrover take the form:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ N_x \\ N_y \\ N_z \end{bmatrix} = \mathbf{M}(\theta) \begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \dot{w}_3 \\ \dot{\omega}_{0x,rel} \\ \dot{\omega}_{0y,rel} \\ \dot{\omega}_{0z,rel} \end{bmatrix} + \begin{bmatrix} \tau_{1,nonlinear} \\ \tau_{2,nonlinear} \\ \tau_{3,nonlinear} \\ N_{x,nonlinear} \\ N_{y,nonlinear} \\ N_{z,nonlinear} \end{bmatrix} \quad (2)$$

However, due to the "rolling without slipping" constraint, some independent variables (N_x, N_y, N_z) occur on the left-hand side of Equation (2) while some dependent variables $\omega_{0x,rel}, \omega_{0y,rel}, \omega_{0z,rel}$ appear on the right-hand side. This illustrates again the need for symbolic derivation of the dynamic equations.

Even though Equation (2) captures the dynamic behavior of the Gyrover completely, it is not in state space form as is required for estimation and control purposes. The state vector X consists of $(\theta_0, \theta_1, \theta_2, w_1, w_2, \omega_{0x}, \omega_{0y}, \omega_{0z})$. The derivatives of each of these variables are obtained from Equations (1) and (2).

$$\dot{\theta}_0 = \omega_{0x} \quad (3)$$
$$\dot{\theta}_1 = w_1 + \omega_{0z} - \omega_{0y} \cot\theta_0 \quad (4)$$
$$\dot{\theta}_2 = w_2 \quad (5)$$

x, u	Plant state and control input vector.
$\hat{x}(k\|l)$	Estimated plant state at time k given measurements up to time step l.
$z(k)$	Plant measurements at time k.
$Q(k)$	Covariance matrix for process noise.
$R(k)$	Covariance matrix for output noise.
$f(x, u, k)$	Nonlinear discrete-time model function.
$\nabla f_x(k)$	Jacobian of $f(.,.,.)$ with respect to the state vector x at time k.
$h_x(x, k)$	Output function at time k.
$\nabla h_x(k)$	Jacobian of the plant output with respect to the state vector x at time k.
$P(k\|l)$	State error prediction covariance at time k given measurements up to time l.
$S(k)$	Observation error covariance matrix.
$W(k)$	Kalman gain matrix.

Table 2: Notation used in the EKF algorithm.

$$\dot{w}_1 = \dot{w}_1 \quad (6)$$
$$\dot{w}_2 = w_c^2(u_2 - \theta_2) - w_c\sqrt{2}w_2 \quad (7)$$
$$\dot{\omega}_{0x} = \dot{\omega}_{0x,rel} - \omega_{0y}\omega_{0z} + \omega_{0y}\omega_{0y}\cot\theta_0 \quad (8)$$
$$\dot{\omega}_{0y} = \dot{\omega}_{0y,rel} + \omega_{0x}\omega_{0z} - \omega_{0x}\omega_{0y}\cot\theta_0 \quad (9)$$
$$\dot{\omega}_{0z} = \dot{\omega}_{0z,rel} \quad (10)$$

where the variables $\dot{\omega}_{0x,rel}$, $\dot{\omega}_{0y,rel}$, $\dot{\omega}_{0z,rel}$, and $\dot{w}_1$ are computed by solving Equation (2). Equation (7) models the dynamics of the tilt-servo system. The inputs to the system are given by the vector $u = (\tau_1, \theta_{2,\mathrm{ref}})$.

3 State Estimator

There are five sensors mounted on the Gyrover:

- three rate gyros attached to the pendulum along the axes of the pendulum frame,
- one encoder on the drive motor, and
- one potentiometer on the tilt servo.

Since there are eight state variables but only five sensors, we need an observer to determine the full state vector. The variation of the linear Kalman Filter for nonlinear systems, called Extended Kalman Filter or EKF [3], is applied to estimate the state vector of the Gyrover. The EKF maximizes the information that is extracted from multiple sensors in a noisy environment, by taking the dynamics of the system into account. Table 2 describes the notation used in the EKF formulation of the observer problem. The following equations summarize the EKF algorithm.

Prediction step.

$$\hat{x}(k|k-1) = f(\hat{x}(k-1|k-1), u(k-1), (k-1))$$
$$P(k|k-1) = \nabla f_x(k) P(k-1|k-1) \nabla {f_x}^T(k) + Q$$

Correction step.

$$S(k) = \nabla h_x(k) P(k|k-1) \nabla {h_x}^T(k) + R(k)$$
$$W(k) = P(k|k-1) \nabla {h_x}^T(k) S^{-1}(k)$$
$$P(k|k) = P(k|k-1) - W(k) S(k) W^T(k)$$
$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)[z(k) - h(\hat{x}(k|k-1))]$$

Combining the expression for the torques in Equation (2) with the dynamic equations (3) to (10) we arrive at the expression for $f(x, u, k)$:

$$\hat{x}(k|k-1) = [\, \theta_0 \; \theta_1 \; \theta_2 \; w_1 \; w_2 \; \omega_{0x} \; \omega_{0y} \; \omega_{0z} \,]^T$$
$$= \begin{bmatrix} \theta_0 + T_s \omega_{0x} \\ \theta_1 + T_s (w_1 + \omega_{0z} - \omega_{0y} \frac{\cos\theta_0}{\sin\theta_0}) \\ \theta_2 + T_s w_2 \\ w_1 + T_s \dot{w}_1 \\ w_2 + T_s \dot{w}_2 \\ \omega_{0x} + T_s (\dot{\omega}_{0x,rel} - \omega_{0y}\omega_{0z} + \omega_{0y}\omega_{0y} \cot\theta_0) \\ \omega_{0y} + T_s (\dot{\omega}_{0y,rel} + \omega_{0x}\omega_{0z} - \omega_{0x}\omega_{0y} \cot\theta_0) \\ \omega_{0z} + T_s \dot{\omega}_{0z,rel} \end{bmatrix}$$

The output function $h(\hat{x}, k)$, to be used in the correction step of the EKF, is given by:

$$h(\hat{x}, k) = \begin{bmatrix} w_1 \\ \omega_{1x} \\ \omega_{1y} \\ \omega_{1z} \\ \theta_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ \omega_{0x}\cos\theta_1 + \omega_{0y}\sin\theta_1 \\ -\omega_{0x}\sin\theta_1 + \omega_{0y}\cos\theta_1 \\ \omega_{0z} + w_1 \\ \theta_2 \end{bmatrix}$$

Computing the Jacobians. In order to use the EKF algorithm, we need to compute the Jacobians of $f(x, u, k)$ and $h(x, k)$ with respect to the state and input vectors. While the Jacobian, $\nabla h_x(k)$ is relatively simple, finding the Jacobian of the dynamics, $\nabla f_x(k)$, is a non-trivial problem. We have implemented a computationally efficient scheme that relies on symbolic pre-computation and numeric run-time computations. Our current implementation runs at 250Hz under QNX™on a 486 Cardio™computer module located on the Gyrover pendulum.

Noise. The dynamic and measurement noise in the system is modeled in the EKF through the covariance matrices Q and R, respectively. They are assumed to be uncorrelated zero-mean Gaussian noise. The following estimates for Q and R were selected based on sensor capabilities and data from the real system:

$$Q = 10^{-6} \begin{bmatrix} .1^2\, I_3 & 0 \\ 0 & 4\, I_5 \end{bmatrix}, \quad R = 10^{-4} \begin{bmatrix} .6^2\, I_4 & 0 \\ 0 & 1 \end{bmatrix}$$

where I_n is the n-order identity matrix.

4 Controller

A controller is designed to stabilize the Gyrover around its upright position $\theta_0 = \pi/2$. Linear state feedback based on the linearized plant around the desired point is used.

4.1 Linearization Analysis

Linearizing the nonlinear dynamic equations of motion about the unstable equilibrium point

$$\theta_0 = \pi/2,\ \theta_1 = 0.0046,\ \theta_2 = 0,\ w_c = 20\pi$$
$$\omega_{0x} = \omega_{0y} = \omega_{0z} = w_1 = w_2 = 0, \quad w_3 = 15000 \text{ rpm},$$

results in the following decoupled state space representation for the system:

$$\dot{X}_i = A_i\, X_i + B_i\, u_i, \qquad Y_i = C_i\, X_i \qquad i = 1, 2.$$

Where $X_1 = \{\theta_1, w_1, \omega_{0z}\}^T$ represents the longitudinal motion, $X_2 = \{\theta_0, \theta_2, w_2, \omega_{0x}, \omega_{0y}\}$ the lateral motion of the Gyrover and $u = (\tau_1, \theta_{2,\text{ref}})$. The output vectors are $Y_1 = \{w_1, \omega_{1z}\}^T$ and $Y_2 = \{\omega_{1x}, \omega_{1y}, \theta_2\}^T$. The constant matrices are:

$$A_1 = \begin{bmatrix} 0 & 1 & 1 \\ -44.32 & 0 & 0 \\ -11.91 & 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -400\pi^2 & -88.86 & 0 & 0 \\ 51.5 & -49.3 & 6.5 & -8.7 & 23.37 \\ 19.17 & -28.3 & 102.5 & -103.7 & 8.7 \end{bmatrix}$$

$$B_1 = [\, 0, \quad 14.52, \quad -0.37 \,]^T$$

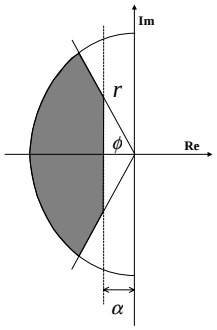
$$B_2 = [\, 0, \quad 0, \quad 400\pi^2, \quad 49.3, \quad 28.3 \,]^T$$

$$C_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0.0046 \\ 0 & 0 & 0 & -0.0046 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The system is completely controllable and observable but non-minimum phase. It has four poles at the origin, a pair of poles at $-44.4(1 \pm j)$ and another pair at $3.10^{-6} \pm 48j$. The zeros of the transfer function from the second input to ω_{1y} has three zeroes at $-8, -1.4\,10^{-4}$ and $+7$.

4.2 State Feedback Controller

To design the controller we optimize the size of the stability region subject to constraints on the inputs, states, and closed loop poles. The optimization is carried out using a semi-definite programming procedure. Stability and constrained regions are defined in terms of Linear Matrix Inequalities(LMI) [1]. Closed loop poles are constrained to a prespecified convex region $S(\alpha, r, \phi)$ [4] as shown in the figure below.



By constraining the poles to lie in a prescribed region, we can achieve a satisfactory transient response. The constraints on the states are derived from the plant mechanical design. We optimize the volume of the ellipsoid contained in the stability region using semi-definite programming(sdp) [9] and find the state feedback gain matrix.

The linearized dynamic system can be described as:

$$\dot{x} = A\,x + B\,u, \qquad y = C\,x$$

where x represents states of the system, u represents the inputs to the system, and A, B, C are constant matrices. Let the observer-based state feedback control law be

$$\dot{\hat{x}} = A\,\hat{x} + B\,u + L\,(y - C\,\hat{x}), \quad u = K\,\hat{x}$$

where $\hat{x}$ is the estimate of the state vector, L is the observer gain matrix, and K is the controller gain matrix. We define the Lyapunov function V as

$$V = x^T\,P_1\,x + (x - \hat{x})^T\,P_2\,(x - \hat{x}), \qquad (11)$$

with P_1, P_2 both symmetric and positive definite. The Lyapunov function defines the ellipsoid ε_C by $V < C$. The volume of the ellipsoid is proportional to $(\det P_1)^{-1/2} \times (\det P_2)^{-1/2}$. Maximizing the volume of ε_C is equivalent to minimizing

$$\log\,\det P_1 + \log\,\det P_2 \qquad (12)$$

Stability is guaranteed if and only if

$$Q_1 A^T + A Q_1 + Y_1^T B + B Y_1 + 2\alpha Q_1 \;<\; 0 \qquad (13)$$
$$Q_2 (A + LC)^T + (A + LC) Q_2 + 2\alpha Q_2 \;<\; 0 \qquad (14)$$

where $Q_1 = P_1^{-1}$, $Q_2 = P_2^{-1}$ and

$$Y_1 = K Q_1 \qquad (15)$$

Input constraints of the type $\|u\| < \mu$ are specified by

$$\begin{bmatrix} Q_1 & Y_1^T \\ Y_1 & \mu^2 I \end{bmatrix} \geq 0. \qquad (16)$$

State constraints of the form $|a^T x| < 1$ are handled by

$$a^T Q_1\, a < 1, \qquad (17)$$

while $S(\alpha, r, \theta)$ add the following LMIs

$$\begin{bmatrix} -rQ_1 & AQ_1 + BY_1 \\ Q_1 A^T + Y_1^T B^T & -rQ_1 \end{bmatrix} < 0 \qquad (18)$$

$$\begin{bmatrix} \sin\phi\,(Z + Z^T) & \cos\phi\,(Z - Z^T) \\ \cos\phi\,(Z^T - Z) & \sin\phi\,(Z + Z^T) \end{bmatrix} < 0 \qquad (19)$$

where $Z = AQ_1 + BY_1$.

The package sdpsol [9] was used to minimize the objective function (12). It solves the convex minimization problem using an interior-point algorithm in terms of Q_1, Q_2 and Y_1. Afterwards, K can be computed from Equation (15). For this optimization L is taken as the steady state Kalman Filter gain matrix and the parameters that define the pole placement region are selected as

$$\alpha = 0.1, \quad r = 50, \quad \phi = 45\ \deg.$$

In summary, to find the state-feedback gain, we perform a convex optimization procedure, minimizing (12) subject to the constraints specified by Equations (13) to (19). The optimization procedure described give us the following control row vectors for each decoupled subsystem:

$$K_1 \;=\; [-18.44,\ -3.31,\ 1.16\,]$$
$$K_2 \;=\; [-0.0077,\ 1.001,\ 0.0019\,0.0179,\ -0.0022\,]$$

with closed-loop poles in { $-41.3, -3.59 \pm 2.76i$, } for the longitudinal motion and { $-34,45 \pm 25.18i$, -10.34, -1.37 and -0.1 } for the lateral motion.

5 Experimental Results

Input-output data sets from the prototype were collected in several runs outdoors, on tiled-floor, and fed it to the EKF algorithm. Figure 2 shows the behavior of the EKF estimates compared to the real data from the Gyrover. Figure 3 shows a control experiment in simulation, using the nonlinear model of the system and the observer-based control scheme discussed in section 4. This simulation also allows to test the convergence of all the states of the EKF to the ones in the nonlinear model, even those that cannot be observed in a real experiment.

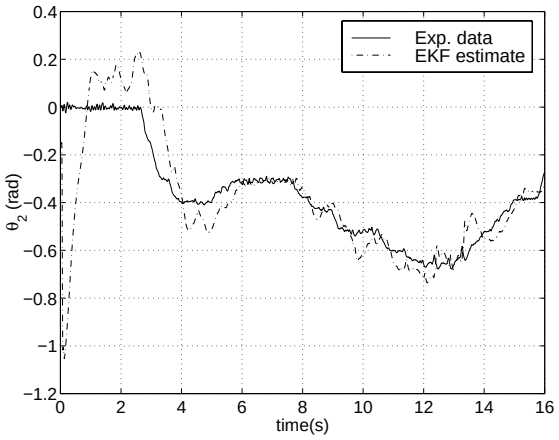


Figure 2: Observer performance from prototype data.

6 Summary

In this paper a controller for the Gyrover is developed. The Gyrover is a gyroscopically stabilized single-wheel robot. Its dynamics are described by a set of highly nonlinear coupled differential equations. However, our analysis has shown that around an operating point at which the Gyrover is upright and the gyro axis horizontal, the dynamics can be linearized into two decoupled systems (fore/aft motion and lateral motion). The decoupled system is controllable, and observable but non-minimum-phase. We have derived and implemented an Extended Kalman Filter and state feedback controller and have demonstrated accurate estimation and control in simulation and experiments.

Acknowledgements

We would like to thank Ben Brown and Arne Suppe for the development of the Gyrover and their help with the implementation of the control algorithms under QNX™. Funding for this research is provided in part by DARPA under contract DABT63-97-1-0003, and by the Institute for Complex Engineered Systems.

References

[1] S. Boyd, L. E. Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* Philadelphia: SIAM 1994.

[2] H. B. Brown, Jr. and Y. Xu, "A single-wheel, gyroscopically stabilized robot",*IEEE Robotics and Automation Magazine*, Vol. 4, No. 3, pp. 39-44, 1997.

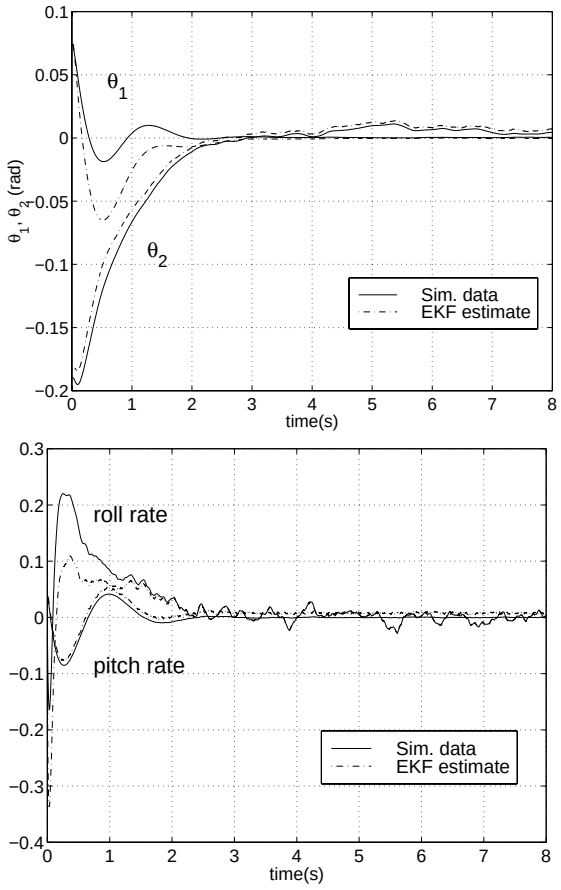


Figure 3: Observer-based controller experiment.

[3] C. K. Chui and C. Chen, *Kalman Filter* Berlin: Springer-Verlag, 1987.

[4] M. Chilali and P. Gahinet, "H_∞ Design with Pole Placement Constraints: an LMI Approach," *IEEE Trans. Aut. Contr.*, Vol. 41, No.3, 1996, pp 358-367.

[5] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.

[6] L. Meirovitch, *Methods of Analytical Dynamics.* McGraw-Hill 1988.

[7] G. C. Nandy and Y. Xu, "Dynamic model of a gyroscopic wheel", *Proc. IEEE Int. Conf. on Robotics and Automation*, 1998, pp 2683-2688.

[8] W. T. Thomson, *Introduction to Space Dynamics* New York:Wiley 1961.

[9] S.-P. Wu and S. Boyd, *A Parser/Solver for Semidefinite Programming and Determinant Maximization Problems with Matrix Structure. User's Guide, Beta Version.*, May 1996.