Proceedings of the 1999 IEEE/RSJ
International Conference on
Intelligent Robots and Systems

Modeling Human Strategy in Controlling a Dynamically Stabilized Robot

Yangsheng Xu, Wai-Kuen Yu, Kwok-Wai Au
Department of Mechanical and Automation Engineering,
The Chinese University of Hong Kong, Hong Kong

Abstract

In this paper, we present a method to model human operator's strategy in controlling a dynamically stabilized robot, Gyrover which is a single-wheel gyroscopically stablilized robot. We first select the relevant state variables for training from kinematics and dynamics equations. Then, we defined a measure of the sensitivity of each of the state variables with respect to operator's control input by a sensitivity function in order to reduce the number of the state variables required in the model. We experimentally implemented the method and demonstrated the robot can be automatically controlled using the learned modeled human control model. The work is of significance in abstracting operator's skill for controlling a dynamically stabilized system in generating an automatic control input.

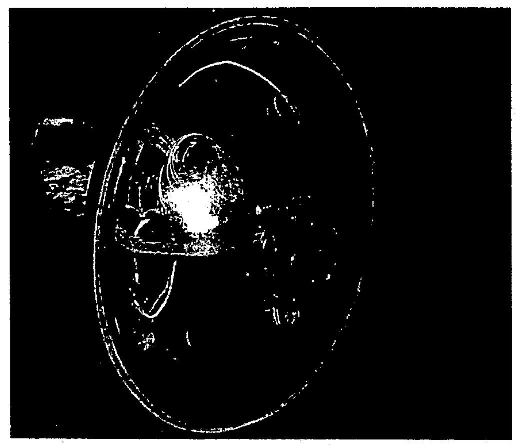

Figure 1: Photograph of Gyrover III.

1 Gyrover

Gyrover is a novel, single wheel gyroscopically stabilized robot, originally developed at Carnegie Mellon University. We have developed three prototypes of the Gyrover. Figure 1 shows the third prototype completed recently. Taking advantage of gyroscopical stability, the robot can provide fast mobility over rough terrains, and possibly on soft land.

The motion of Gyrover is based on the principle of gyroscopic precession as exhibited in the stability of a rolling wheel. Because of its angular momentum, a spinning wheel tends to precess at right angles to an applied torque, according to the fundamental equation of gyroscopic precession: $T = J \times \dot{\gamma} \times \dot{\alpha}$ where $\dot{\gamma}$ is the angular speed of the wheel, $\dot{\alpha}$ is the wheel's precession rate, normal to the spin axis, J is the wheel polar moment of inertia about the spin axis, and T is the applied torque, normal to the spin and precession axes. Therefore, when a rolling wheel leans to one side, instead of falling over, the gravitationally induced torque causes the wheel to precess so that it turns in the direction that it is leaning.

Gyrover supplements this basic concept with the addition of an internal gyroscope — the spinning flywheel — nominally aligned with the wheel and spinning in the direction of forward motion. The flywheel's angular momentum produces the lateral stability when the wheel stops or moves slowly. The dynamic model of Gyrover was derived in [1][2][3], where the system was divided into three parts: 1) single wheel, 2) internal mechanism, 3) spinning flywheel as shown in Fig 2. Only three control variables are available on the system, i.e., the drive torque (u_0), the tilt torque (u_1) and the rolling speed of the flywheel (u_2). The control of Gyrover is challenging because the nature of dynamically stabilized system and complexity of modeling. Therefore, it has been driven manually, through a radio controlled interface. We have made many sensors available in the third prototype, and it is our goal to achieve automatic control using on-board real time control system and various sensors.

0-7803-5184-3/99/$10.00

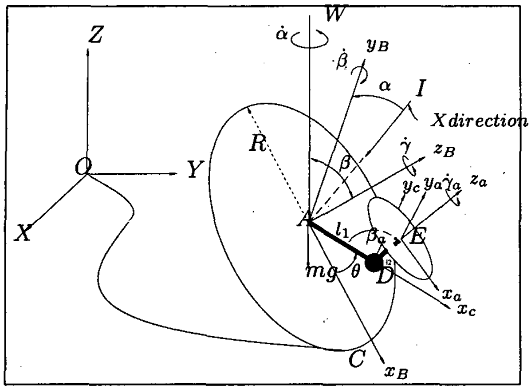

Figure 2: Coordinates assignment and parameters defined

2 Modeling Human Control Strategy

Modeling human control strategy(HCS) in response to real time input is of significance in research areas from robotics to the intelligent vehicle highway system. Because human control strategy is a dynamic, nonlinear, stochastic process, developing a valid good analytic models of human strategy tend to be difficult. Therefore, the recent work in modeling HCS has focused on learning empirical models, through, for example, neural network techniques [7]. Taking advantage of our group's work on modeling human control strategy [6], we used the flexible cascade neural network architecture with node-decoupled extended Kalman filtering (NDEFK) for the modeling the map between human operator's control command given through a radio control interface and the state variables of the robot measured from on-broad sensors in Gyrover.

We prefer the learning architecture over others for the following reasons. First, no a priori model structure is assumed; the neural network automatically adds hidden units to an initially minimal network as the training requires. Second, hidden unit activation functions are not constrained to be a particular type. Rather, for each new hidden unit, the incremental learning algorithm can select that functional form which maximally reduces the residual error over the training data. Typical alternatives to the standard sigmoidal function are sine, cosine, and the Gaussian function. Finally, the node-decoupled kalman filtering has shown to significantly improve learning speed and error convergence of the cascade learning architecture [5]. The flexible functional form which cascade learning allows is ideal for abstracting human control strategies, as shown in Fig.3, since we know very little about the underlying structure of each individual's internal controller. By making as few a priori assumptions as possible in modeling the human control data, we improve the likelihood that the learning algorithm will converge to a model of the human control data.

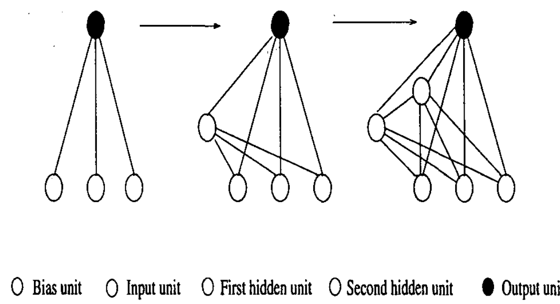

Figure 3: The cascade learning architecture adds hidden units one at a time to an initially minimal network.

The motivation for modeling human strategies in controlling a dynamically stabilized robot is as follows: First, due to the underactuated and nonholonomic nature of the system, it is very difficult to design a controller that ensures continuously trajectory tracking. Second, it is equally difficult to precisely model the system because it largely depends on various unmodeled parameters, such as friction, impact force, coriolis force and centrifugal force. These parameters are relatively more important in the dynamically stabilized system discussed here, than a conventional static or quasic-static system, like a manipulator. Third, a skilled operator can actually control a dynamically stabilized system very well. Therefore, we naturally come up the idea of abstracting human strategy in controlling such as a dynamically stabilized robot. This paper aims to develop a methodology for modeling human strategy in controlling such a system. We discuss how the state variables are selected for the model and how the model is generated for realtime control.

3 Input Variables Selection

By modeling human control strategy here we mean modeling the human operator's control action in response to the system real-time performance. That is, develop a relationship between the human control action (commands) and the system response (state variables). The human control commands refer the data generated from the radio-controlled interface manually. The system state variables are various sensor data from the on-board sensors equipped within the robot. In this study, for simplicity, we consider the drive torque u_0, tilt torque u_1, as control commands only because the flywheel speed u_2 is set to be constant. For the state variables, we have over 12 sets sensor data available combining their derivative, there are too many sets of data for training. We would like to select a set of data out of all available sensing data for more efficient training and more concise model. In selecting potentially relevant state variables, we first would like to select a set of variables which are most important for an operator to make a decision. Second, they must be measurable using the on-board sensors, and at the same time, it must be visually observable for the operator. Consider the fact that operators control the robot by not observing many state variables anyway. Therefore it is possible to minimize the set of state variables in training. To this end, we started from checking kinematic constraints and dynamics equations derived in [2]. The kinematics equations relationship is :

$$\dot{X} = R(\dot{\gamma}c\alpha + \dot{\alpha}c\alpha c\beta - \dot{\beta}s\alpha s\beta) \quad (1)$$
$$\dot{Y} = R(\dot{\gamma}s\alpha + \dot{\alpha}c\beta s\alpha + \dot{\beta}c\alpha s\beta) \quad (2)$$
$$\dot{Z} = R\dot{\beta}c\beta \quad (3)$$

The equation of gyroscopic precession yields

$$T = J \times \dot{\gamma} \times \dot{\alpha} \quad (4)$$

The torque generated from change of the angular momentum of the flywheel.

$$T = \frac{\partial H}{\partial t} \quad (5)$$

where

α, γ, β	: yaw,pitch,row angle of single wheel robot w.r.t. earth frame.
$\dot{\alpha}, \dot{\gamma}, \dot{\beta}$	: angular velocity yaw-pitch-row.
$\dot{\alpha}_a, \dot{\gamma}_a, \dot{\beta}_a$	: angular velocity of flywheel w.r.t. body frame.
$I^f_{xx}, I^f_{yy}, I^f_{zz}$	: moment of inertia of flywheel w.r.t. body frame.
R	: Radius of the Wheel
H	: $I^f_{xx}\dot{\beta}_a + I^f_{yy}\dot{\gamma}_a + I^f_{zz}\dot{\alpha}_a$

From the equations listed above, the variables $(\beta, \beta_a, \dot{\alpha}, \dot{\alpha}_a, \dot{\beta}, \dot{\beta}_a, \dot{\gamma}, \dot{\gamma}_a, \ddot{\alpha}_a, \ddot{\beta}_a, \ddot{\gamma}_a)$ can be considered as the state variables. We may further reduce the number of the state variables by the following observations. From the hardware constraints, we know $\dot{\alpha}_a = 0$ and $\ddot{\alpha}_a = 0$. We cannot measure α and $\dot{\gamma}$ because of unavailability of sensors. $\dot{\gamma}$ can be approximated by u_0, because u_0 controls $\dot{\gamma}$ and thus u_0 is directly proportional to $\dot{\gamma}$, provided that no sliding occurs. $\ddot{\gamma}_a$ must be zero because the rolling speed of flywheel is constant. Therefore, $\beta, \dot{\beta}, \beta_a, \dot{\beta}_a, \ddot{\beta}_a, \dot{\alpha}, \dot{\gamma}, \dot{\gamma}_a$ can be selected as the state variables.

Another important issue is to consider the previous state variables in forming dynamic process of human decision making. The previous state variables are significant for human operators and the current state variables of the robot. The system diagram is shown Fig. 4. Therefore, the inputs to the neural network include (1) the current and previous state variables ($\beta, \dot{\beta}, \beta_a, \dot{\beta}_a, \ddot{\beta}_a, \dot{\alpha}, \dot{\gamma}, \dot{\gamma}_a$), (2) the previous control output variables (u_0, u_1).

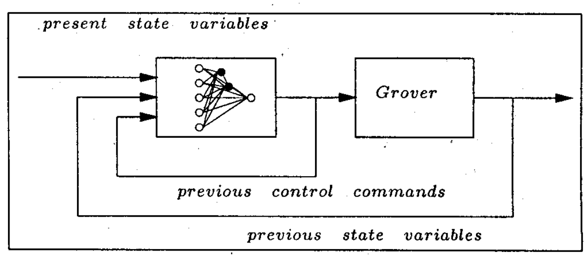

Figure 4: System diagram of modeling human control strategy.

4 Sensitivity Analysis

The above selected variables may not necessarily be significant for an operator in giving a control action. Here, we will study the sensitivity of the operations control commands with respect to the robot state variables. We defined a sensitive function of the output U with respect to each of the state variables as follow:

$$T(x^i_k) := \frac{||\Delta U_k||_2}{||\Delta x^i_k||_2} \quad (6)$$

where $X = \{x^1, x^2, \cdots\}$ is a set of the state variables, and $U = \{u^1, u^2, \cdots\}$ is a set of the control commands. Both X and U are time functions, i.e., $X_k = \{x^1_k, x^2_k, \cdots\}$ and $U_k = \{u^1_k, u^2_k, \cdots\}$ representing

average $T(x^i)$	$T(\beta_a)$	$T(\dot{\beta}_a)$	$T(\ddot{\beta}_a)$	$T(\beta)$	$T(\dot{\beta})$	$T(\dot{\alpha})$	$T(\dot{\gamma}_a)$
sample 1	7.34	1.49	0.12	1.98	16.52	0.25	∞
sample 2	4.12	0.86	∞	11.20	39.99	2.50	∞
sample 3	5.79	6.94	0.33	13.18	24.16	96.41	∞

deviation $\sigma(T(x^i))$	$\sigma(T(\beta_a))$	$\sigma(T(\dot{\beta}_a))$	$\sigma(T(\ddot{\beta}_a))$	$\sigma(T(\beta))$	$\sigma(T(\dot{\beta}))$	$\sigma(T(\dot{\alpha}))$	$\sigma(T(\dot{\gamma}_a))$
sample 1	30.4	8.06	0.54	7.43	150	0.8	nil
sample 2	157	9.0	nil	60.3	389	16.4	nil
sample 3	17.1	41.4	0.91	17.69	965.9	86.4	nil

Table 1: Sensitivity measure

X_k and U_k at time $t = k$. If $||\Delta x_k^i||_2 = ||x_k^i - x_{k-1}^i||_2$ increases, while $||\Delta U_k||_2$ changes insignificantly, i.e., $T(x^i)$ is small, we may conclude that the operator's manual control does not depend much on the state variable x^i. At the same time, when the same value is extremely large, we may conclude that, although the state variable does not change much, the control input actually change a lot, meaning their relationship is weak. In both cases, i.e. $T(x^i)$ is extremely large or extremely small, we will automatically delete x^i as the state variables for training.

We normalized X and U and computed the average value of the sensitive function by using three sets data performed by the same operator, shown in Table 1. $T(\dot{\gamma}_a)$ were extremely large because the rolling speed of the flywheel $\dot{\gamma}_a$, were constant. $T(\ddot{\beta}_a)$ were extremely small. Thus, $\ddot{\beta}_a$ and $\dot{\gamma}_a$ were given up as they do not play an important role in generating control commands. In this way, a set of relatively important variables can be selected for the operators control decision making.

5 Experiment

In our experiment, we studied the tilt-up motion of the single wheel robot, as shown in Fig 5. It is difficult to control the tilt-up motion of Gyrover as sliding sometimes occurs. Moreover, the tilt-up motion is very important for successful stabilization. While Gyrover runs with the lean angle $\beta \approx 90^o$, it is much more stable because of its gyroscopic precession. Our purpose is to make the robot stand up (that is, initially, the lean angle is $\beta \approx 18^o$ and finally, the lean angle is $\beta \approx 90^o$). In our system, there are three control variables, u_0 controlling the rolling speed of the single wheel $\dot{\gamma}$, u_1 controlling the angular position of the flywheel β_a, and u_2 controlling the rolling speed of the flywheel $\dot{\gamma}_a$. We neglect the u_2, simply because the control variable u_2 is in most cases set to be constant.

At the beginning of the task, the states of Gyrover is shown in Fig 5 (a). Then, as u_0 increases, Gyrover starts to roll. By changing the angular momentum of the flywheel as shown in Fig. 5 (b), a tilting torque is acted on the system. As a result, the robot can tilt up as shown in Fig. 5 (c). The whole tilt-up motion strategy can also be demonstrated by a simple rolling disk shown in Fig. 6.

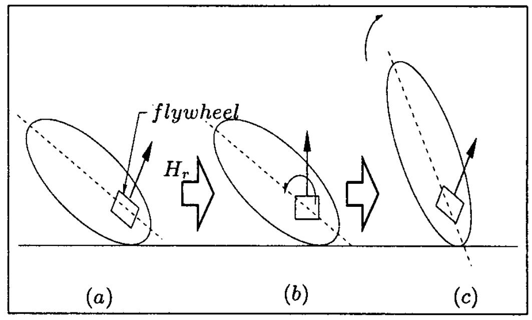

Figure 5: Grover motion in tilting up.

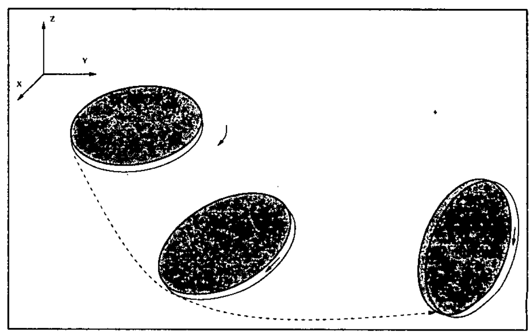

Figure 6: Tilt-up motion strategy of Grover.

An operator manually controlled the robot through the radio-controlled interface. On-board sensors measure the selected set of state variables and transmitted data to a computer through wireless communication. Both sets of data from the interface and from the robot are recorded. Figure 7 shows some examples of the operators' control commands (the bottom two figures) and the state variables measured (the upper two figures). The recorded data was used for training HCS model. The learned HCS model was used as the control input for the robot. Figures 8 to 10 show the corresponding components of neural network control inputs, and the recorded state variables. In figures, four important variables (β, β_a, u_0, u_1) were listed as examples here where β is the lean angle representing target of the mission, β_a is the lean angle of the flywheel generating the main tilt torque for stabilization, and u_0 and u_1 are the control commands.

In experiment, sometimes, it was difficult to set the initial lean angle β to be exactly same as that when human manually controlled it. However, the result seemed to be insensitive to the slightly different initial lean angles we set. By comparing the learned HCS input with the manual control input, the control strategies were shown to be similar. The perturbation of the lean angle under the learned control input is relatively small. The similar case can be found in the figures. Shown in Fig. 9 and Fig. 10, the lean angle β are overshooted over 10 degrees. It is interesting to note that u_0 becomes small while u_1 become large. When the tilt torque increases significantly, u_0 could reduce the tilt torque, based on Equation 4. It shows the learned HCS model have a high adaptability for stabilizing the system by reducing overshooting.

In each experiment, the goal was always achieved: $\beta \approx 90^o$, even if the initial angles and the execution time varied. It demonstrated that the tilt-up skill we modeled was valid. The learned control input is capable of controlling the dynamically stabilized robot by using the similar pattern of the operator's manual control input.

6 Conclusions

In this paper, we presented a method of modeling human operator 's strategy in controlling a dynamically stabilized robot, Gyrover. We first selected relevant state variables for training from kinematics and dynamics equations. Then, we defined a measure of sensitivity of the operator's control commands with respect to the state variables in order to reduce the number of the data sets for efficient training. We experimentally implemented the method and demonstrated that the robot can be automatically controlled using the learned human control input. The work is of significance in abstracting operator's skill in generating an automatic control input for controlling a dynamically stabilized system.

Acknowledgments

The research is partilly supported by RGC grant CUHK 4164/98E. The helps from J.Y. Song and W.K. Fung are much appreciated.

References

[1] H. B. Brown and Y. Xu, "A single wheel gyroscopically stabilized robot." *Proc. IEEE Int. Conf. on Robotic and Automation,* Vol. 4, pp. 3658-63, 1996.

[2] Y. Xu, K.W. Au, G.C. Nandy and H.B. Ben, "Analysis of actuation and dynamic balancing for a single Wheel Robot", *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* Vol 4, pp. 3658-63,1998.

[3] G.C. Nandy and Y. Xu, "Dynamic model of a gyroscopic wheel", *Proc. IEEE Int. Conf. on Robotic and Automation,* Vol. 3, pp. 2683-2688, 1998.

[4] Haruhiko Asada and Sheng Liu, "Transfer of human skills to neural net robot controllers", *Proc. IEEE Int. Conf. on Robotic and Automation,*,1991.

[5] S.E. Fahlman, L.D. Baker and J.A. Boyan, "The Cascade 2 learning architecture," *Techanical Report. CMU-CS-TR-96-184, Carnegie Mellon University,*1996.

[6] M.C. Nechyba and Y. Xu, "Cascade neural network with node-decoupled extended Kalman filtering", *Proc IEEE Int. Symp. on Computational Intelligence in Robotics and Automation,* Vol. 1, pp. 214-9, 1997.

[7] M.C. Nechyba and Y. Xu, "Human control strategy: abstraction, verification and replication", *IEEE Control Systems Magazine,* Vol. 17, no. 5, pp. 48-61, 1997.

[8] J.Y. Song, Y. Xu, M.C. Nechyba and Y. Yam, "Two measures for evaluating human control strategy", *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* 1998.

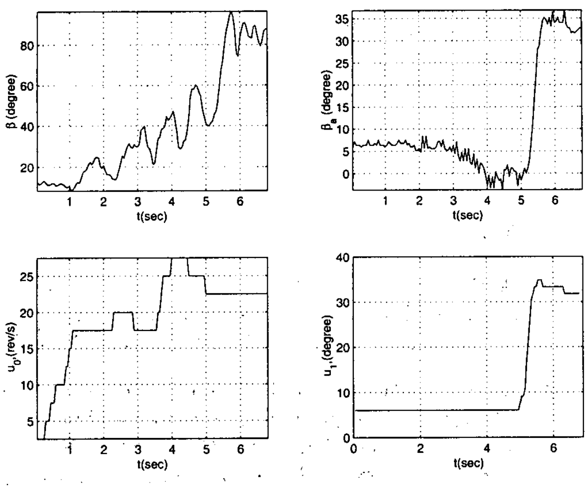

Figure 7: Data recorded for training(a)β, (b)β_a, (c)u_0, (d)u_1

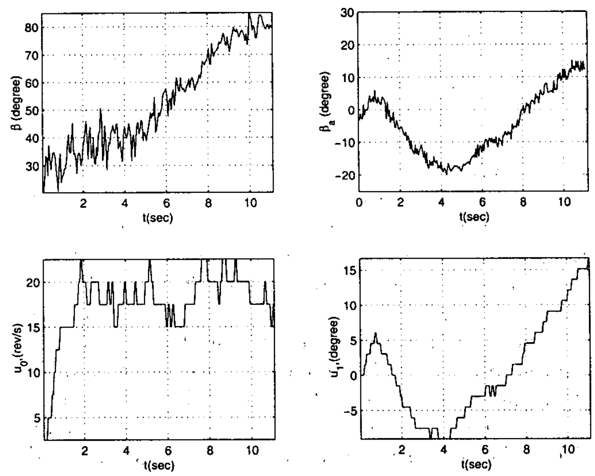

Figure 8: Data recorded when the learned model is used as control input: case I (a)β, (b)β_a, (c)u_0, (d)u_1

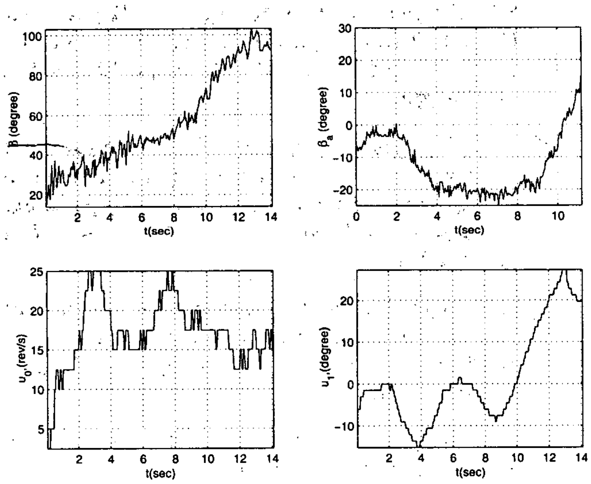

Figure 9: Data recorded when the learned model is used as control input: case II (a)β, (b)β_a, (c)u_0, (d)u_1

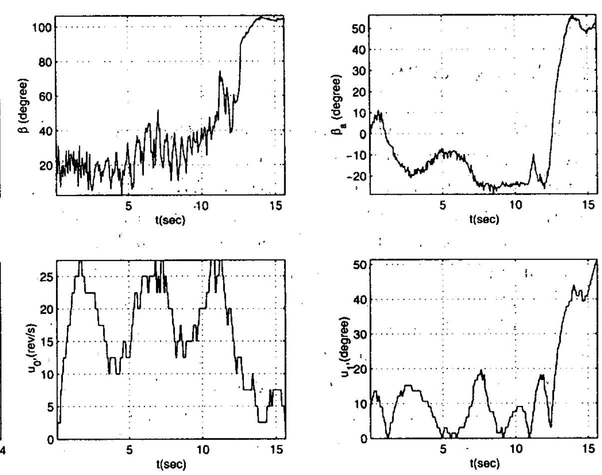

Figure 10: Data recorded when the learned model is used as control input: case III (a)β, (b)β_a, (c)u_0, (d)u_1