

On Controlling the *Segboy* using a *Gameboy Advance*

D. Block, N. Gatzke and R.S. Sreenivas
 Coordinated Science Laboratory, and
 Industrial and Enterprise Systems Engineering
 University of Illinois at Urbana-Champaign
 Urbana, IL 61801
 Corresponding Author's Email: rsree@illinois.edu

Abstract—The *Segboy* is based on a self-balancing, two-wheeled vehicle in reference [1]. This device was built at the *College of Engineering Control Systems Laboratory (COECSL)* at the University of Illinois at Urbana-Champaign, and is used in a classroom setting in many occasions. This paper describes the self-balancing control of this two-wheeled vehicle using the *Gameboy Advance* and *XPort*.

I. INTRODUCTION

The *Segboy*, built at the *College of Engineering Control Systems Laboratory (COECSL)* at the University of Illinois at Urbana-Champaign, is based on a self-balancing, two-wheeled vehicle from reference [1]. It is shown in figures 1(a), 1(b) and 1(c).

The *Gameboy* by *Nintendo*, with the *Gameboy Advance* (GBA) in the third generation, is a popular game console that can be purchased for about \$30.00-\$40.00. The *XPort* by *Charmed Labs*¹ expands the abilities of the GBA by connecting it to a field-programmable gate array (FPGA), providing 64 general purpose input/output (I/O) signals that can be programmed to communicate with practically any kind of hardware. The XPort together with the additional robot daughter board forms the *XPort Robot Controller (XRC)*. The daughter board provides in addition to the XPort other features like analog-to-digital conversion (ADC). This makes the GBA, together with the XRC, as an ideal controller for a variety of control problems. This paper is about using these common place devices as micro-controllers that balance and move the *Segboy* within a cluttered environment.

II. THE MODEL OF THE SEGBOY

The model of the *Segboy* adopted in this paper is essentially what is shown in figure 2 of reference [1], along with the simplification that the two wheels are considered as one. That is, we assume there is no rotation about the z -axis, and the movement in the xy -plane is reduced to a movement along the x -axis (i.e. Φ in the model of reference [1] is set to 90°). This is justified by the fact that the *Segboy* has a smaller mass than the vehicle of reference [1], and the effect of rotation about the z -axis is small enough that the dynamics is not influenced by it. With this assumption, the coordinate system of reference

[1] is simplified, and the model is reduced from a three-dimensional, to a two-dimensional model. Additionally, the rotational energy of the body about the z -axis is not considered when using the Lagrangian model for developing the dynamics of the system.

The vehicle's translational position, x , and angular position, Ψ , are chosen as the two independent variables to define the generalized coordinate vector q as $q = [x \ \Psi]^T$ (cf. figure 2). With reference to this figure, we observe that θ (resp. z) can be described in terms of x (resp. Ψ).

Just as with figure 2 of [1], the center-of-mass (CM) has a distance of L to the axis of the wheel with a radius of R . The translational position of the CM in x -coordinate (x_{CM}) and z -coordinate (z_{CM}) is

$$\begin{aligned} x_{CM} = x + L \sin(\Psi) &\Rightarrow \frac{d}{dt}x_{CM} = \frac{dx}{dt} + L \cos(\Psi) \frac{d\Psi}{dt} \\ z_{CM} = R + L \cos(\Psi) &\Rightarrow \frac{d}{dt}z_{CM} = -L \sin(\Psi) \frac{d\Psi}{dt} \end{aligned}$$

We require the equations for the different energies of the system in order to use the Lagrangian method, which are described following notational preliminaries. The symbol m_b denotes the mass of the body; m_w , the mass of the wheel; J_b , the moment of inertia of the body; J_w , the moment of inertia of the motor; η , the gear ratio of the motor; g the acceleration due to gravity. The translational energy in the form of kinetic energy due to the body's movement in x -direction is given by

$$E_{body,x} = \frac{1}{2}m_b \left(\frac{d}{dt}x_{CM} \right)^2 = \frac{1}{2}m_b \left(\frac{dx}{dt} + L \cos(\Psi) \frac{d\Psi}{dt} \right)^2,$$

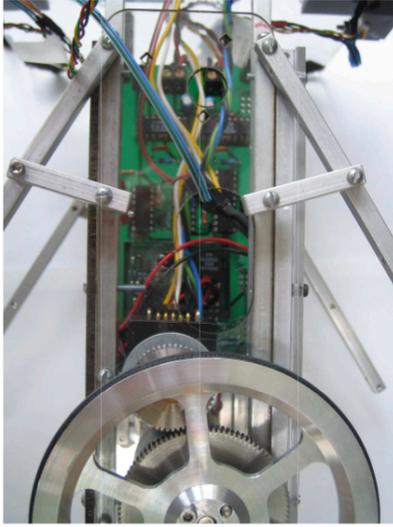
the translational energy in the form of kinetic energy due to the body's movement in the z -direction is

$$E_{body,z} = \frac{1}{2}m_b \left(\frac{d}{dt}z_{CM} \right)^2 = \frac{1}{2}m_b \left(-L \sin(\Psi) \frac{d\Psi}{dt} \right)^2,$$

the translational energy in the form of kinetic energy due to the wheel's movement in the x -direction is

$$E_{wheel,x} = \frac{1}{2}m_w \left(\frac{dx}{dt} \right)^2,$$

¹<http://www.charmedlabs.com>



(a) Side View



(b) Top View



(c) Under Balance.

Fig. 1. The *Segboy* built at the COECSL at UIUC.

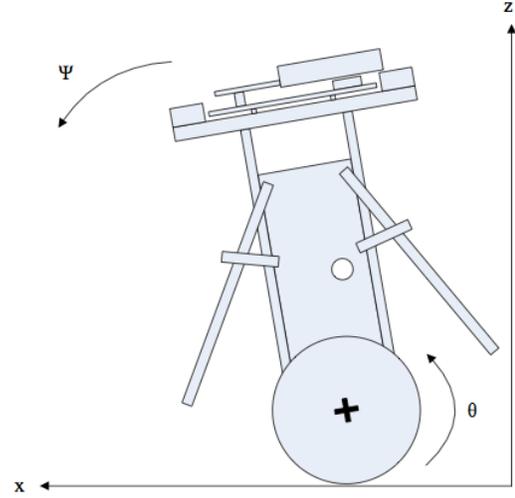


Fig. 2. 2-dimensional coordinate system of the *Segboy*. The white circle indicates (estimated) center-of-mass.

the rotational energy in the form of kinetic energy of the body's movement in the Ψ -direction is

$$E_{body,\Psi} = J_b \left(\frac{d\Psi}{dt} \right)^2,$$

the rotational energy in the form of kinetic energy of the wheel's movement in the θ -direction is

$$E_{wheel,\theta} = J_w \left(\frac{d\theta}{dt} \right)^2 = \frac{J_w}{R} \left(\frac{dx}{dt} \right)^2,$$

the rotational energy in the form of kinetic energy of the motor rotor's movement in the θ -direction is

$$E_{motor,\theta} = J_m \eta^2 \left(\frac{d\theta}{dt} - \frac{d\Psi}{dt} \right)^2 = J_m \eta^2 \left(\frac{1}{R} \frac{dx}{dt} - \frac{d\Psi}{dt} \right)^2,$$

and the potential energy of the body is given by the expression

$$E_{pot} = m_b g L \cos(\Psi).$$

The resulting Lagrangian $L (= \sum E_{kin} - \sum E_{pot})$ is

$$L = \left(\frac{m_b}{2} + m_w + \frac{J_m \eta^2}{R^2} + \frac{J_w}{R^2} \right) \left(\frac{dx}{dt} \right)^2 + \left(m_b L \cos(\Psi) - \frac{2J_m \eta^2}{R} \right) \frac{dx}{dt} \frac{d\Psi}{dt} + \left(\frac{m_b L^2}{2} + \frac{J_b}{2} + J_m \eta^2 \right) \left(\frac{d\Psi}{dt} \right)^2 - m_b g L \cos(\Psi),$$

from which the equations of motion of the system are derived according to the expression

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \left(\frac{dq}{dt} \right)} \right) - \frac{\partial L}{\partial q} = u,$$

where u denotes the control input, which in this case is the torque τ of the motor. This yields the following system of equations

$$\begin{pmatrix} m_b + 2m_w + \frac{2(J_m\eta^2 + J_w)}{R^2} & m_b L \cos(\Psi) - \frac{2J_m\eta^2}{R} \\ m_b L \cos(\Psi) - \frac{2J_m\eta^2}{R} & m_b L^2 + J_b + 2J_m\eta^2 \end{pmatrix} \begin{pmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2\Psi}{dt^2} \end{pmatrix} = \begin{pmatrix} \frac{\tau}{R} + m_b L \sin(\Psi) \left(\frac{d\Psi}{dt}\right)^2 \\ \tau + m_b g L \sin(\Psi) \end{pmatrix},$$

which is linearized about the vertical equilibrium point of $\Psi \approx 0^\circ$ and $\frac{d\Psi}{dt} \approx 0$, to yield the following equation

$$\begin{pmatrix} m_b + 2m_w + \frac{2(J_m\eta^2 + J_w)}{R^2} & m_b L \cos(\Psi) - \frac{2J_m\eta^2}{R} \\ m_b L \cos(\Psi) - \frac{2J_m\eta^2}{R} & m_b L^2 + J_b + 2J_m\eta^2 \end{pmatrix} \begin{pmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2\Psi}{dt^2} \end{pmatrix} = \begin{pmatrix} \frac{\tau}{R} \\ \tau + m_b g L \Psi \end{pmatrix}.$$

Based on this, a state-space model was developed with a state vector $\mathbf{x} = [\frac{dx}{dt} \ \Psi \ \frac{d\Psi}{dt}]^T$. The routine, yet laborious details of this procedure is skipped for brevity. The various physical parameters were estimated, or measured by straightforward methods. The uncontrolled system is unstable, which is confirmed by the presence of a pole in the left-hand-side of the s -plane. The system is controllable (and observable), which makes in amenable to pole-placement design. Using MATLAB, we obtained a set of gains for the state feedback controller that results in a stable closed-loop system.

III. SEGBOY HARDWARE

Our main objective was to decrease the size of the Segboy compared to the design of reference [1]. This was achieved by using smaller hardware components and a smaller controller (the GBA) that could be mounted to a smaller chassis.

The parts of the chassis are mainly from a remote control car kit (“*Terminator Sumo Kit*” from *Lynxmotion*, <http://www.lynxmotion.com>). This kit includes two identical ($6.37'' \times 4.5'' \times 0.125''$) laser-cut Lexan plates that form the front and back of the Segboy. A hole was drilled into one of them to access the switch used for turning the amplifiers of the motors on and off. Two identical ($6.75'' \times 2.25'' \times 0.125''$) Lexan side plates are connected via four aluminum bars. The original side plates were extended with two drilled slots used to attach the two optical encoders for the wheels. The existing side holes (about $1.7'' \times 1''$) are used to access the 5V power pins as well as the PWM pins without the need to take the construction apart. As these two side plates were designed to attach the motors of the original car, holes were already provided for that purpose.

To receive a high resolution with the optical encoders, a 2:1 gear (from *Stock Drive Products*, <http://www.stockdriveproducts.com>) was used with the motors.

Two 4'' wheels (*TTW-4000-S* from *Truturn*, <http://www.truturn.com>) were mounted to the chassis. To avoid damage to the system if there is a loss of balance, four aluminum bars were mounted to the sides, supported by four small bars also included in the kit (cf. figure 1(a) and 1(c)).

Four aluminum bars used for the main body were modified by drilling additional holes into them on top and on the sides to

carry a $7.25'' \times 10.25'' \times 0.25''$ Plexiglas plate on top, which holds the circuit board inside of the main body. This board is attached such that the accelerometer and gyroscope are as close as possible to the wheels axis of the Segboy. Former experience had showed that this configuration worked better than when they are placed further apart.

The top plate is used to carry the XRC. In the current implementation, only the ADC chip of the XRC is used, and it is powered directly by the GBA, there was no need to also attach its original battery cases used for powering the other components of the robot daughter board.

Prior experience with the original Segbot showed that a high weight on top builds a damping force against sudden accelerations of the robot, thus improving its balancing and moving abilities. Due to this fact, the top plate is also used to carry the two Lithium-Polymer batteries attached with Velcro to easily remove and change them. Six pieces of Velcro on the bottom side of the top plate are used to easily connect and replace the two infrared (IR) sensors in different positions. The relevant construction details can be seen in figure 1(b). The Segboy with all parts connected has a weight of 5.1 lbs and a height of 10''.

Two 12V DC motors (also originating from the “*Terminator Sumo Kit*” from *Lynxmotion*, <http://www.lynxmotion.com>) serve as actuators of the robot. Both have a 30:1 metal gear and a stall torque of 63.89 oz*in.

Two Li-Poly batteries (*TP2100-3S* from *Thunder Power Batteries*, <http://www.thunderpower-batteries.com>), each producing an output of 11.1V and 2100mAh, are used for the power supply of the motors and the circuit board.

A detailed description of the controller can be found in reference [2], and is skipped for brevity.

IV. SENSORS

The Segboy uses a total of seven sensors. Two optical encoders (*S1-1000-B4* from *US Digital*, <http://www.usdigital.com>) receive data from the wheels. These were mainly chosen because of their low price and availability in the COECSL. Both encoder counter values are read each control cycle, and the change to the previous value is computed to estimate the velocity; this information is also used to estimate the position of the wheels by summation of the changes. One round of an encoder is equal to a change of the absolute sensor value by 4000; however, since there is a 2:1 gear between the encoders and wheels, one round of the wheels corresponds to a change of the sensor value by an absolute value of 8000. As a user interface, a third optical encoder (*S2-1000-B* from *US Digital*, <http://www.usdigital.com>) is used, if necessary, to steer the robot while it is moving at a (pre-determined) constant velocity.

An accelerometer (*ADXL103* from *Analog Devices*, <http://www.analog.com>) and a gyroscope (*ADXRS300* from *Analog Devices*, <http://www.analog.com>) is used to sense tilt and tilt-rate of the Segboy. The accelerator is modified to estimate the

tilt angle of the Segboy². The gyroscope outputs an estimate of the tilt rate of the body. Due to the modification of the accelerator to act as a tilt sensor, the signal lags so much that the control algorithm will not work if we were to use just this input. Therefore, a sensor signal fusion is set up; this process is described in section VII.

To recognize objects in the Segboy’s environment, two infrared (IR) sensors (*Sharp GP2D02* from *Acroname*, <http://www.acroname.com>) are used. The information from these sensors are used to avoid running into objects as well as for developing algorithms like wall-following.

V. THE CIRCUIT BOARD

The board, powered by the two Thunder Power batteries, carries two up to 50V, 2A amplifiers, one for each motor. A switch is built-in to turn the amplifiers on and off. By an appropriately designed logic, we ensure that if the switch is accidentally in “on” position when the system is powered up, the amplifiers are remain turned off to avoid harm or damage by spinning motors.

An on-board DC-DC converter reduces the 12V of the power supply to 5V providing power supply for the sensors. The accelerometer and the gyroscope are placed in an additional board, which includes a DC conversion unit, changing the output signal of the sensors from a 0V to 5V range to a 0V to 3.3V range. This is necessary, as the input channels of the XRC’s ADC chip require an voltage in the range of 0V to 3.3V. The schematics for the board can be obtained from the authors of the paper.

VI. SOFTWARE

The XPort FPGA had to be programmed to interact with all sensors connected to the Segboy, store their information and coordinate it for the GBA. It is also used to produce the PWM signal to send the calculated control effort to the circuit board driving the motors. The different tasks were programmed in several modules in Verilog. The main module of the Segboy contains 6 different sub-modules. The first is the primary module, necessary for communication with the GBA. Additionally, there are three modules to read data from the ADC chip, the optical encoders and IR sensors; two modules output a PWM signal and a test signal. An overview is presented in figure 3, additional details can be found in reference [2].

The signal of the accelerometer, as well as the signal of the gyroscope, is an analog signal in the range of 0 to 3.3V. They are connected to channel 0 and 1 of the XRC’s on-board ADC chip. A module for reading the channels of the ADC had to be developed. To save resources on the FPGA, the module was written to not read all eight channels, but only the two in use (channel 0 and 1). This necessitated a transaction on three pins. A chip select signal has to be sent to the chip as well as a clock signal. Additionally, a third bidirectional connection is used as follows – first, it sends out the data

²A detailed description how to use the accelerometer as a two-axis tilt sensor can be found at www.analog.com.

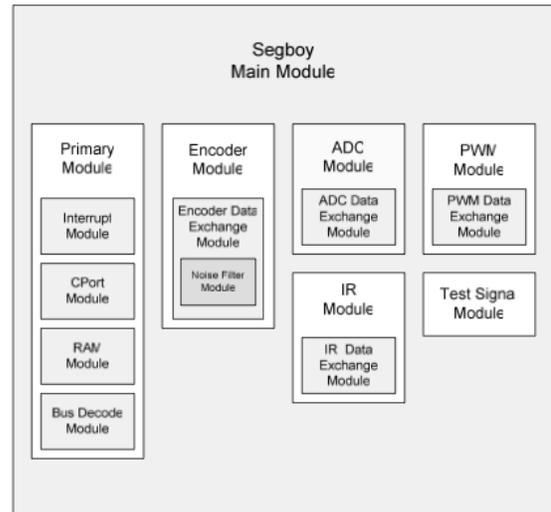


Fig. 3. The XPort FPGA modules for the Segboy.

which channel to read and starts the actual conversion. Then, it receives the 12-bit ADC signal of the chosen channel and stores it in a register that can be read by the GBA. This process was accomplished by producing a clock signal of 1MHz and a state machine triggered by it. The ADC conversion itself is started by a write operation to a certain address from the GBA (it could also be started directly on the FPGA, but to make sure the sample time can be changed quickly and easily, this was preferred). Again, a state machine is used to get the data from both channels. After the conversion is over, an interrupt is called on the GBA, starting the control algorithm.

It takes the IR sensor a minimum of about 70ms for a distance measurement. Two digital I/Os of the XRC have to be connected to each sensor, one input and one output channel. Again a module containing a state machine was programmed to implement the necessary sequence of the required signals. The resulting data of the measurement is saved by the FPGA in a register. As the IR data is updated much less often than the control algorithm on the GBA is called (about 70ms vs. 4ms), a busy signal is implemented indicating if new data is available or not, which can be checked by the GBA to avoid unnecessary read operations.

Several registers need to be defined according to the different addresses on the FPGA where the data of the sensors is stored. They are in form of vectors saved in other C-Code files. These files must be compiled for the ARM chip as well and are then uploaded with the file containing the main function on the flash memory of the XPort. Before the actual control algorithm starts, the problem of the measurement of the tilt angle described earlier needs to be solved. This is done by sensor fusion, which is described in the next section.

VII. SENSOR FUSION

As previously stated, the tilt angle of the robot Ψ and its tilt rate $\frac{d\Psi}{dt}$ are states of the systems used by the controller for

balancing the Segboy. Measuring the tilt with an accelerometer, while being accurate, is ineffective due to the large lag in sensor output. The alternate approach of integrating the tilt-rate from a gyroscope to get a fast-estimate of the tilt angle is not effective either. This is because gyroscopes have a high drift rate, which can result in very poor performance. A *complimentary filter* that combines both accelerometer and gyroscope data for a precise, fast estimate of the tilt angle was designed using the method in reference [3]. This filter essentially combines the low-pass filtered signal of the accelerometer and the high-pass filtered signal of the fast gyroscope integral as shown in figure 4. That is, when the robot is not moving or slowly moving about its wheel axis, the precise output of the tilt sensor is used. When the robot is moving fast about its wheel axis, the integrated tilt rate signal is used to estimate the tilt.

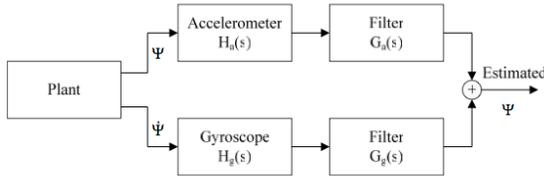


Fig. 4. Signal flow for the Complementary Filter.

Following [3], assuming ideal sensors (i.e. $H_a(s) = 1$ and $H_g(s) = 1$), the transfer functions of the complimentary filter are $G_a(s) = \frac{2\tau s + 1}{(\tau s + 1)^2}$ and $G_g(s) = \frac{\tau^2 s}{(\tau s + 1)^2}$, which was discretized using the backward rectangular rule ($s = (z - 1)/Tz$), where $\tau = 0.8$ and $T = 0.001s$. These values were selected after some iterations. The equations had to be derived in integer math, due to the GBA's inability to handle floating point math in an acceptable amount of time. This required some experimentation about the scaling factors that we would need to convert the filtering action into integer math. This also necessitated an increase in the sampling time from $T = 0.001s$ to $T = 0.004s$ before a satisfactory working filter was obtained.

VIII. THE CONTROL ALGORITHM

The Segboy is controlled by a full state feedback controller. The three states used are the translational velocity, the tilt and the tilt rate. A timer on the GBA (Timer 3) is set up such that an interrupt is called every $4ms$. The interrupt service routine it is calling starts a write operation, which prompts the FPGA to send out the proper signal to the ADC chip to start a conversion of channel 0 and 1. Once the ADC chip has completed a conversion, another interrupt is called, now by the FPGA over the external interrupt line of the GBA. This indicates the start of the control algorithm. First, the interrupt on the FPGA has to be reset, following this, the external interrupt on the GBA itself is reset.

Next, the data of both ADC channels, gyroscope data and accelerometer data, respectively, is read and scaled. The information for the tilt rate, $\frac{d\Psi}{dt}$, can be used as input without

any modification, while the signal for the tilt angle, Ψ , has to be filtered first as described in section VII. The next step involves reading the encoder values from both wheels, and, if connected, from the third encoder used as a user interface. The optical encoder counter values from the wheels are used to estimate the velocity of the robot by storing the value every cycle and subtracting it from the succeeding counter value. In case of a rollover of the register, the estimate of the velocity is simply replaced by the preceding value. The rollover is registered by checking if the calculated value has an unusually large size. At this stage, the data from the accelerometer and the gyroscope are both filtered, but the results are not added together yet, and the sensor fusion is not complete. The sensor information is collected over two cycles or $8ms$. This first part of the algorithm is illustrated in the flow graph shown in figure 5(a).

After the average of all input data is calculated, the wheel's position is obtained by finding the average of each the two wheel velocities and adding that velocity multiplied by time to the previous position. If the IR sensors are activated, a function is called to see if a measurement was completed and new data is available. This data is then used for different purposes depending on the mode the Segboy is in, which is described later.

Following the IR loop, the control effort is calculated. Parts of it are dependent on the mode the robot is in again, thus a query to determine which mode is active occurs. The mode independent control effort, that is, the effort to make the robot balance, is computed by multiplying all the three states with their gains. These three then are added together with the mode dependent effort. As an assumption of the model was that it has only one wheel, the output is simply divided into two half and send to each of the motors. To allow a turn of the robot regardless of this, another disturbance is implemented before the control effort is sent to the motors.

To improve the balancing abilities of the Segboy, a friction compensation is added. The values for it were estimated by fixing the robot on a bench with the wheels free to move, and increasing the control effort of each wheel in each direction until it starts moving. After the output to the motors is saturated, a final check for the tilt of the Segboy is made to turn the motors off if it is in a position outside its region of functionality, where it is no longer able to balance. This again prevents the robot of getting damaged. Finally, the control effort is sent to the PWM registers on the FPGA. A flow graph of this second part of the algorithm can be found in figure 5(b).

The controller gains of the Segboy was tuned to yield good results in balancing and moving. Small obstacles can be driven over without causing problems in the stability of the system.

The GBA with its multimedia capabilities and keypad offers a great way to implement a user menu. The flash memory on the XPort is big enough to provide several configurations for the Segboy to choose from as well as the opportunity for a user to change many parameter of the system over the ten buttons. The screen can display graphics as well as text. This was used

to set up a graphical user menu for an easy navigation, to show all kinds of data of the system, and to change the parameters of interest in every mode.

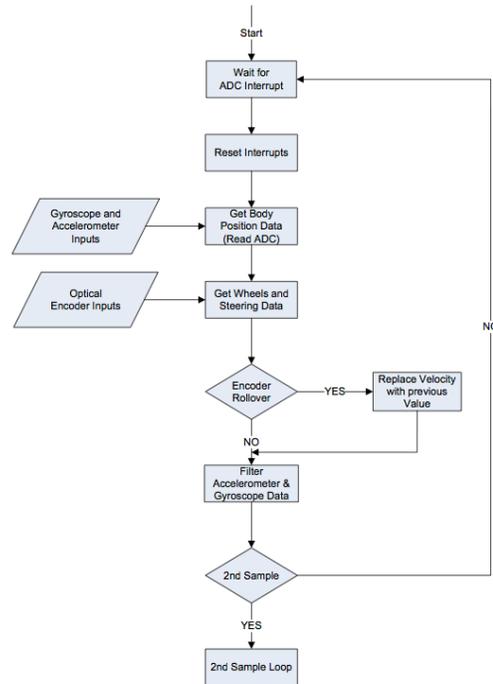
The Segboy can run in five different modes: (1) *Balance*, (2) *Right Wall Follow*, (3) *Left Wall Follow* (4) *Drive* and (5) *Setup*. The default mode the robot is started in is the *Balance* mode, where the objective of the algorithm is to make the robot balancing at its initial position. In the *Right Wall Follow* mode, the IR sensors are used to make the Segboy follow along a wall to its right in a certain distance, and turn left if it realizes an obstacle in front. Entering this mode, the robot is balanced until the IR sensors are activated, giving the user the possibility to put everything into the right position before starting. One IR sensor has to point in front direction of the robot, while the other one has to point to the right in an angle of about 45° to the front. The straight wall following is implemented by simply comparing the distance measurement from the sensor pointing to the right with a desired distance, and multiplying the error between these two values with a proportional gain for a turn command. If an obstacle appears in front, the robot is set up to turn left until the obstacle disappears. The *Left Wall Follow* mode works with the same principle as the Right Wall Following. The second IR sensor has to point left instead of right, everything else is equivalent to the procedure described earlier. In the *Drive* mode, an optical encoder is connected to the XRC. Its data is used to obtain a turn command, thus a user can use the encoder to steer the robot. Also, the IR sensors are active to avoid obstacles. The *Setup* mode allows the user to change various parameters of the system, as well as get information on all devices connected to the Controller.

IX. CONCLUSION

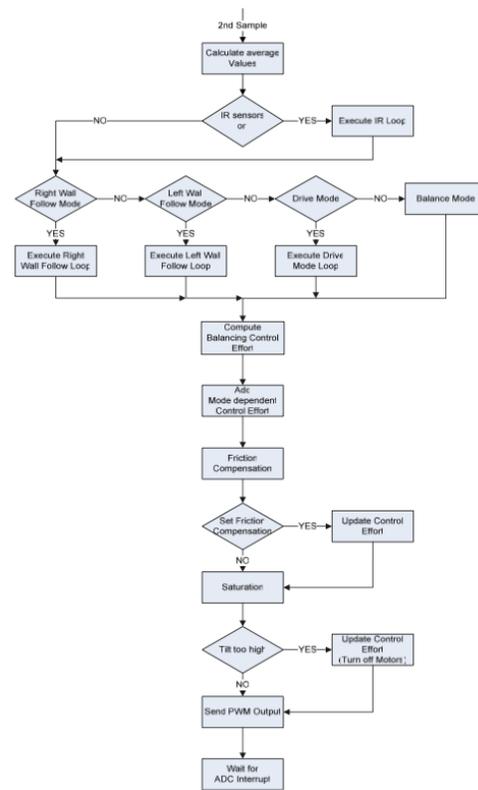
The Segboy was successfully balanced and driven by the GBA in combination with the XPort. One of the main motivation of the project was to use the results as a teaching aid in control systems classes at the University of Illinois at Urbana Champaign. Apart from the physical construction of the device, a significant portion of the over all effort involved the design and implementation of the data-fusion filter (cf. section VII). The GBA's inability to perform floating point arithmetic placed significant constraints on the achievable final design. The selection of the scaling operations to convert floating point into integer operations involved a significant amount of iterations. As a result this system has found limited use in our classroom.

REFERENCES

- [1] M. Baloh and M. Parent, "Modeling and model verification of an intelligent self-balancing two-wheeled vehicle for an autonomous urban transportation system," in *The Conference on Computational Intelligence, Robotics, and Autonomous Systems*, Singapore, December 2003.
- [2] D. Block, N. Gatzke, and R. Sreenivas, "On using the gameboy advance as a controller for inverted pendulums," Yogyakarta, November 2013, submitted.
- [3] A.-J. Baervelt and R. Klang, "A low-cost and low-weight attitude estimation system for an autonomous helicopter," center for Computer Systems Architecture, Halmstad University, Halmstad, Sweden.



(a) First Sample Loop



(b) Second Sample Loop

Fig. 5. Flow diagram of the control algorithm.