

Smartphone-Based Two-Wheeled Self-Balancing Vehicles Rider Assistant

Alexander Smirnov^{*†}, Alexey Kashevnik^{*†}, Igor Lashkov[†], Naohisa Hashimoto[‡], Ali Boyali[‡]

^{*}SPIIRAS, St.Petersburg, Russia

[†]ITMO University, St.Petersburg, Russia

[‡]National Institute of Advanced Industrial Science and Technology, Intelligent Systems Research Institute, Tsukuba, Ibaraki, Japan

smir@iias.spb.su, alexey@iias.spb.su, igor-lashkov@ya.ru, naohisa-hashimoto@aist.go.jp, boyali@gmail.com

Abstract—This paper presents an approach to a driver assistant system for a two-wheeled self-balancing mobility vehicles in particular for a Segway. The approach is aimed for the readily available mobile devices, which become a part of our daily life such as a smartphone or a tablet. If a mobile device is well-positioned on a mobility vehicle, its front and rear cameras can be utilized as sensors to capture the ride related information about the rider’s intention(s) and the interaction of the rider with the environment. In addition, attached to the handle bar of the mobility vehicle, this mobile device can be used to alert the driver using the motion and location sensor as well as cameras and gather ride characteristics. In this study, we describe a context-aware system that continuously observes both the rider and the dynamical characteristics of the ride and provides alerts to the rider anticipating the hazards, collision, the route of the other public road users, and the stability of the current ride characteristics.

I. INTRODUCTION

Rider Assistance Systems are the systems that assist the rider during the driving process. They are designed with a safe human-machine interface aiming to increase vehicle and road safety. It is common practice that such kind of systems are designed for car riders by the third party manufacturers that are specialized on them and can develop similar applications for the smartphones and tablets.

Development of the rider assistance systems for the two-wheeled self-balancing mobility devices has drawn a great deal of attention of the mobile application developers, as the use of robotic mobility devices has become in vogue in the recent years.

The authors in the study [1] propose that a paradigm shift is necessary in solving traffic issues in favor of promoting public transport instead the use of individual vehicles. However, when the elderly people are concerned, from the closest transportation hub to the final destination might arise complications for these cohort as their motor and mobility skills greatly decrease.

Two-wheeled self-balancing vehicles are the remedy for elderly people to solve the last-mile problem. However, the number of fatal accidents in Japan due to elderly riders has increased nearly three-fold in the past 17 years, while the total number of fatal accidents has decreased by nearly 30% during the same period [2].

Having been invented in 2001, the Segway Personal Transporter (PT) (trademarked by the Segway Inc. of New Hampshire, USA) is the first two-wheeled self-balancing mobility device introduced in the market. In most of the states in USA, the use of Segway is permitted under a new regulation which categorizes Segway in a newly introduced vehicle category, “Electric Personal Assistive Mobility Device (EPAMD)”.

Many countries around the world such as United Kingdom, Australia and Japan do not allow the use of Segway on the public roads except private properties and designated zones due to the fact that, they still don’t have regulations because of the vehicle category as Segway lacks of brakes and the software limited upper speed limit exceeds the limits defined for the EPAMDS.

A scant number of research papers is available in the literature that are limited to some empirically gathered data, dynamical characteristics and subjective assessments which were aimed to guide the policy-makers and traffic regulation bodies.

In the study [3, 4], the authors conducts experiments to find out the approaching distance and velocities of the Segway riders with respect to various obstacles, pedestrians and objects at the different velocity profiles.

The experiments were implemented under the controlled conditions by recruiting ten novices and ten experienced operators. A similar study [5] reports the stopping distance for the different driving maneuvers including emergency braking and response time of the riders when braking. The collected Segway riding data were also compared to the running characteristics bicycles. In the study [6], the observed reactions of the vehicle riders and passing distance

to the different types of personal mobility devices including the Segway are given in detail. The experiments in the study were performed on road crossings while the riders turning left.

Some pilot projects [7, 8] in the literature report and discuss the safety requirements for Segway on the shared public road related areas. The reports also reflect the opinion of the various stakeholders and institutions as well as the subjective assessments from the recruited riders in the experiments.

The available limited literature does not cover all aspects of the interaction of Segway or the PTs with the other public road users in a consistent and methodological way. The listed experimental studies were performed under the controlled conditions and statistical data is not sufficient to have opinion for the law and regulation authorities. On the contrary, even though a small amount of statistical data is available, Segway has been used in many countries in the designated zones for patrolling or recreational purposes. A tremendous amount of data can be taken out in these real ride conditions by mounding a common spread smart device on Segways on the grounds that each travel and riding experience induces precious statistical information that can contribute into the development of more reliable mobility robots and alleviate the limitations for the public use of these devices.

The paper proposes an approach for mobile application development that assists the Segway rider during the travel by providing visual and tactile feedback if an unsafe situation is predicted.

The front camera facing up to the rider allows the application to determine the rider's mental states, whereas the rear camera, GPS, motion and orientation sensors to detect the external environment as well as the dynamical characteristics of the vehicle.

The proposed reference system model in this paper incorporates different services and algorithms. The reference model is based on the ontological knowledge representation that allows providing for ontology-based information sharing between different services in the developed system. The ontological rider and Segway models portray them formally in terms of comprehensibility of the information systems and using these descriptions in the assisting processes. For assistive functionalities, the motion related parameters are observed and evaluated simultaneously.

The rest of the paper is structured as follows. The reference model of Segway rider assistant system is presented in Section II. Section III and Section IV present ontological models of rider and Segway. Segway motion detection using Android-based smartphone algorithm is given in Section V. The results are summarized in Conclusion.

II. REFERENCE MODEL

The reference model of the smartphone-based Segway rider assistant system is presented in Fig. 1. It consists of five main modules: mobile application, cameras, sensors, local database and cloud service. For accessing to the smartphone's sensors (accelerometer, gyroscope, magnetometer, and GPS) the Android "SensorManager" class is used. Data from the device's different sensors collected by Sensor fusion component caters for estimating the various useful quantities such as the speed, the acceleration, and the location. The Android Camera API is used to work with front-facing and rear-facing cameras. Inner components of the mobile application are context-aware camera switching algorithm, multi-core computation planner and image processing unit. Today's smartphones do not have the capability to process video streams from both of the front and the rear cameras simultaneously. In this respect, we use a context-aware algorithm that switches between the two cameras while processing the data real-time with the goal of minimizing missed events inside. The image processing unit is responsible for extracting the visual features from the images taken by the rear and front cameras. The computation planner aims to effectively leverage the multi-core architecture of modern smartphones to perform heavy calculations.

As for storage options we rely on user preferences, local database and cloud service. User preferences module allows saving and retrieving persistent key-value pairs of primitive data types. We save calibration options and other application settings accepted by user. If the Internet connection is not available, local database is responsible for storing data collected from the smartphone. As soon as the Internet connection becomes available, we are ready to synchronize a local database with the cloud service. Synchronization service is a component of the assistant system responsible for managing the information flows to/from the database located on the smartphone and to/from the cloud. Such information as smartphone characteristics, application usage statistics, and dangerous events occurred during trip is stored for using in the future. Smartphone characteristics are GPU, sensors (GPS, Accelerometer, Gyroscope, Magnetometer), cameras (front-facing / rear-facing), memory & battery capacity, and version of operation system. In addition, the cloud storage is used for keeping behavior patterns and driving style patterns. Operations that can be carried out in the cloud storage are:

- Recognition of true and false responses due to occurrence of dangerous events.
- Matching of behavior and driving style patterns.
- Analysis and classification of driver behavior and driving style for further making recommendations for safe driving.

A smartphone attached on the Segway is used for recognition of the rider behavior. For the sake of clarity, let's imagine that a rider with a several hours of trip on Segway at night, and likely to fall asleep.

We fix our smartphone on a Segway and start our application. We follow calibration instructions and the

application starts to monitor the Segway rider and the road ahead using the front-facing and rear-facing cameras respectively. The most dangerous situations while driving at night are driver drowsiness and impending road hazards. If the image-processing unit recognizes drowsiness using the image taken by the front-facing camera and computer

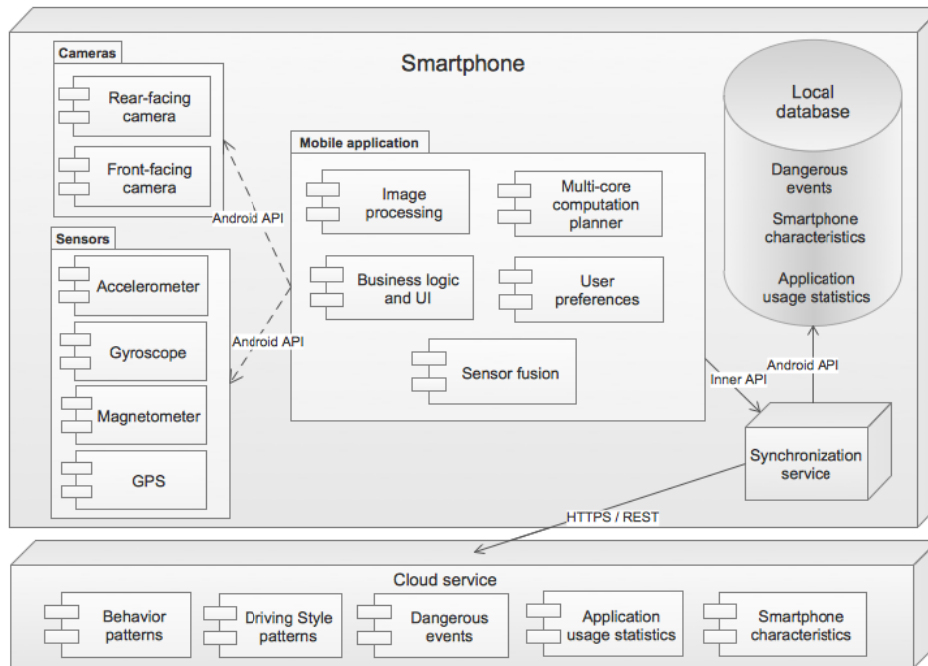


Fig. 1. Reference model of the Segway rider assistant system

vision (e.g. Speeded Up Robust Features (SURF), Haar Cascades) and machine learning (e.g. SMV (Support Vector Machine), AdaBoost) algorithms or detects any obstacles on the road ahead using the rear-facing camera. It will force the application engine to make an audio and visual alert to inform the rider to cheer up and to be more vigilant. One more unsafe frequent driver behavior event is speeding. Using the GPS and accelerometer sensors Sensor fusion module estimates the Segway speed and sends this kind of information to the application engine for decision-making process.

Driver behavior events, driver style and other extracted or estimated characteristics are stored in the local database and cloud service.

III. RIDER MODEL

Human-being perform a variety of nonverbal communication cues as well as voice, posture and the face that can be used to infer underlying mental states. The human face in particular provides one of the most powerful, versatile and natural means of communication with a wide array of mental states. We make use of the front facing camera of the smartphone attached on Segway to monitor the face of the rider. An application is to monitor the human

face the front camera of the smartphone is used. It is designed for tracking facial expressions and gestures. First of all, the system is configured initially to detect eyes closure responding with a warning signal. At the same time, it analyses the rider's eye-blink, the respiration rate and the depth of breathing to the difference between just blinking and closing of the eyes while driving.

On the focus of the system are the behavioral and physiological signals acquired from the rider to assess their mental state real-time. In the presented approach, the rider is considered as a device with a large number of internal mental states. Each of these states has its own particular control behavior and interstate transition probabilities. The canonical example of this type of model would be a bank of standard linear controllers (e.g., Kalman Filters plus a simple control law). Each controller has different dynamics and measurements, sequenced together with a Markov network of probabilistic transitions. The states of the model can be hierarchically organized to describe the short and long-term behaviors.

The scenario requires that the human's internal states have to be determined through an indirect estimation process, as they are not directly observable. To accomplish this, we can adopt the expectation-maximization methods

developed for the use with hidden Markov models. By using these methods to identify a user's current pattern of control and predict the most likely pattern of subsequent control states, it is possible to recognize human behaviors accurately and anticipate and their projections for several seconds in the future.

The people in fatigue exhibit certain type of visual behaviors that can easily be observed from the changes in their facial expressions and features from the eyes, head, and face. The typical visual characteristics observable on a face image of a person are the reduced alertness level that includes slow eyelid movement [9, 10], smaller degree of eye openness (or even closed), frequent nodding [11], yawning, gaze (narrowness in the line of sight), sluggish in facial expression, and sagging posture. To make the use of these visual cues, another increasingly popular and non-invasive approach for monitoring fatigue is to assess a rider's vigilance level through visual observation the physical conditions of the face using a camera and computer vision technologies. The techniques using computer vision are aimed for extracting visual characteristics that typically characterize a rider's vigilance level from the video images of the monitored face. The features that can be assessed are given as follows:

- PERCLOS – PERcentage of CLOSure of eyelid.
- Eye blink time.
- Eye-blinking rate.
- Eye gaze.
- Pupil movement.
- Eyelid movement.
- Postures.
- Head pose.

Visual behaviors observable from the changes in facial features listed above are:

- Eyes are opened or closed.
- Facing to the left.
- Facing to the right.
- Facing forwards.
- Gaze concentration towards the road.
- Gaze concentration not towards the road.
- Dilated pupils.
- Not dilated pupils.

The developed rider ontology (Fig. 2) includes these visual cues and visual behaviors and determines relationships between them. It consists of the five main top level classes: "AlertLevel" (warning level), "DangerousEvents" (commonly occurring dangerous driving events), "CharacteristicParameters" (visual characteristics observable from the image of a person), "RiderProfile" (a rider profile that reflects the certain

personal characteristics) and "VisualCues" (the visual cues on which we focus to detect dangerous events).

Each rider has a profile (class "RiderProfile") that reflects personal characteristics. Rider profile consists of background (class "Background") and real-time (class "AtTheMoment") context. Background context in its turn includes five elements that underpin a rider safety culture – behavior (class "Behavior"), attitude (class "Attitude"), awareness (class "Awareness"), motivation (class "Motivation") and skills (class "Skills"). These classes are associated with each other with the relationship "is_a". On the other hand we need to consider the real-time data (class "AtTheMoment"). It consists of wish for assistance (class "WishForAssistance"), driving style (class "DrivingStyle"), level of attention (class "LevelOfAttention") and driving behavior (class "DrivingBehavior"). The relationship between these classes is "is_a". System alert level (class "AlertLevel") depends on rider profile. Class "AlertLevel" is associated with the class "RiderProfile" with the relationship "depends_on".

The basic characteristic parameters (class "CharacteristicParameters") that typically characterize rider's state are PERCLOS (class "PERCLOS"), eye-blink rate (class "Eye-BlinkRate"), eye closure speed (class "EyeClosureSpeed"), eye-blinking time (class "EyeBlinkTime"), eye gaze (class "EyeGaze"), pupillary state (class "PupillaryState"), yawning (class "Yawning") and nodding level (class "HeadNodding"). The relationship between these classes is "is_a".

We infer the dangerous rider behaviors (class "DangerousEvents") such as drowsiness (class "Drowsiness"), distraction (class "Distraction") and fatigue (class "Fatigue"). In the proposed ontology, the corresponding classes ("Drowsiness", "Distraction" and "Fatigue") are associated with the class "DangerousEvents" with the relationship "is_a". At the same time, face orientation and eye gaze are used to detect distraction. (classes "FaceOrientation" and "EyeGaze" are associated with the class "Distraction" with the relationship "is_used_to_estimate"). PERCLOS, eye-blink rate, eye closure speed, eye blink time, yawning and nodding level are used to recognize drowsiness. (Classes "PERCLOS", "Eye-blinkRate", "EyeClosureSpeed", "EyeBlinkTime", "Yawning" and "HeadNodding" are associated with the class "Drowsiness" with the relationship "is_used_to_detect"). And finally, eyelid movement, face orientation, gaze movement and facial expressions reflect level of fatigue (classes "EyelidMovement", "FaceOrientation", "GazeMovement" and "FacialExpressions" are associated with the class "Fatigue" with the relationship "reflects_level_of"). Open or closed eyes are a good indicator of fatigue. (Property "EyeState" is associated with the class "Fatigue" with the relationship "is_an_indicator_of").

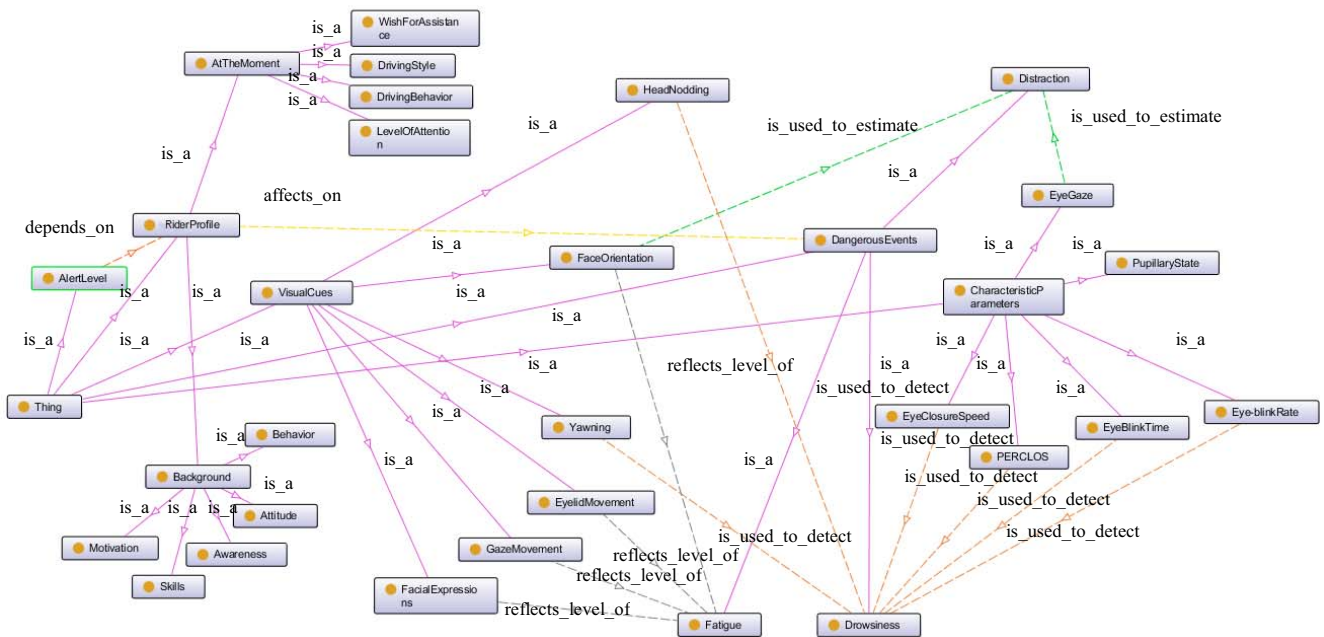


Fig. 2. The Segway rider ontology model

IV. SEGWAY MODEL

The Segway riders are faced with a multitude of road hazards and an increasing number of distractions (e.g. music, phone calls, smartphone texting and browsing, advertising information on the road, and etc.) In the presented approach the following five of the most commonly occurring dangerous driving events are addressed, such as:

- Drowsy driving.
- Vigilance decrement.
- Inattentive driving.
- Tailgating.
- Ignoring blind spots during lane changes.

The characteristics that can be observed by the images of the road and the smartphone sensors include:

- Segway speed and acceleration.
- Vehicle headway (measurement of the distance or time between vehicles).
- Lane position and road signs.
- Segway turns.

The developed Segway ontology model (Fig. 3) consists of five main top level classes: “DangerousEvents” (dangerous events that can occur during driving), “Sensors” (embedded sensors on the phone), “Cameras” (built-in front-facing and rear-facing cameras), “VehicleBehaviorParameters” (segway behavior parameters) and “RoadParameters” (parameters that

characterize the road the segway moves). At the same time, classes “VehicleBehaviorParameters”, “Sensors” and “RoadParameters” are used to recognize hazards (class “DangerousEvents”). Classes “VehicleBehaviorParameters”, “Sensors” and “RoadParameters” are associated with the class “DangerousEvents” with the relationship “is_used_to_recognize”).

The class “DangerousEvents” is classified as “Tailgating” (riders should maintain a minimum safe distance with the vehicle or moving object ahead) and “IgnoringBlindSpots” (executing lane changes safely also requires a rider to check blind spots before proceeding). In the proposed ontology, the corresponding classes (“Tailgating” and “DangerousEvents”) are associated with the class “DangerousEvents” with the relationship “is_a”.

Most Android-powered devices have built-in sensors (class “Sensors”) such as accelerometer (class “Accelerometer”), gyroscope (class “Gyroscope”), magnetometer (class “Magnetometer”) and GPS (class “GPS”). Corresponding classes (“Accelerometer”, “Magnetometer”, “Gyroscope” and “GPS”) are associated with the class “Sensors” with the relationship “is_a”.

But also Android framework includes support of cameras (class “Cameras”) and camera features available on device, allowing to capture pictures and videos in applications. We aim to work with front-facing (“FrontFacingCamera”) and rear-facing (“RearFacingCamera”) cameras. Classes “FrontFacingCamera” and “RearFacingCamera” are associated with the class “Camera” with the relationship

“is_a”. Pictures taken from rear-facing camera help us to recognize behavior parameters such as turn (class “Turn”), vehicle headway (class “VehicleHeadway”), trajectory (class “Trajectory”) and lane position (class “LanePosition”). These classes are associated with each other with the relationship “is_used_to_recognize”.

The class “VehicleBehaviorParameters” includes such parameters such as the speed (class “Speed”), acceleration (class “Acceleration”), lane position (class “LanePosition”), trajectory (class “Trajectory”), Segway turns (class “Turn”) and vehicle headway (class “VehicleHeadway”). They are associated with the class “VehicleBehaviorParameters” with the relationship “is_a”.

The GPS sensor (class “GPS”) and accelerometer (class “Accelerometer”) are used to estimate the position, acceleration (class “Acceleration”) and the speed (class “Speed”). The classes “Acceleration” and “Speed” are associated with the classes “GPS” and “Accelerometer” with the relationship “is_used_to_estimate”.

Besides, inertial sensors (classes “Accelerometer”, “Magnetometer” and “Gyroscope”) are used for trajectory detection (class “Trajectory”) and these are associated with the relationship “is_used_to_detect”. The Segway turns (class “Turn”) are detected by observing the significant changes in the direction from the time-series data of the GPS (class “GPS”) positions. The class “GPS” is associated with the class “Turn” with the relationship “is_used_to_detect”.

The last top level class is “RoadParameters”. It contains lane markers (class “LaneMarkers”), road conditions (class “RoadConditions”), obstacles (class “Obstacles”) and road signs (class “RoadSigns”). Classes “LaneMarkers”, “RoadConditions”, “Obstacles” and “RoadSigns” are associated with the class “Road” with the relationship “is_a”.

And finally, the road parameters (class “Road parameters”), sensors (class “Sensors”) and the Segway behavior parameters (“VehicleBehaviorParameters”) are used to recognize dangerous events. These classes are associated with each other with the relationship “are_used_to_recognize”.

Each driver has his own skills, motivation, attitude and qualifications that we should consider in monitoring and detecting dangerous events occurring throughout the trip. That’s why the rider and Segway models are rather closely related.

For instance, executing lane changes safely requires a driver to check blind spots before proceeding. The driver does this by looking in the side and front mirrors of the car to check for unexpected vehicles. Segway application should be capable of recognising the head position of the driver using the phone’s front camera, allowing the app to

ensure the appropriate mirror checks are performed before each lane change. Lane changing itself can be detected using the rear camera and inertial sensors, as described above.

V. SEGWAY MOTION DETECTION USING ANDROID-BASED SMARTPHONE

A standard Android-based smartphone has a built-in 3-axis accelerometer sensor and a gyroscope which measure acceleration acting on the device and the angular rates in the three axes. The accelerometer also measures the gravity component on the each axis. Thus, the gravity component must be subtracted from the measurements for the analyses.

Fig. 4 illustrates the body coordinate system of the smartphone. The rotated y-axis is the forward direction of the motion in the current Segway - smartphone configuration. For the motion classification such as that of the braking modes, the best discriminative features can be obtained from the inertial sensors are the rotation angle and the angular rate of the Segway’s handle bar around the x-axis which are the pitch angle and angular rate.

The handle bar of Segway has 2 Degree of Freedom (DOF); pitch and roll angles that are the rotations around the x and y axes.

The measurements must be aligned in respect to the body coordinate system of the Segway. To do this, the only parameters are the pitch and roll angles. Even though provided by the Android APIs the reported measurements for the orientation angles are not reliable at the higher frequencies. Thus, we compute these parameters using the trigonometric relationships between the accelerometer measurements on the three axes. The pitch and roll angles can also be computed by integrating the gyroscope measurements. However, both of the Micro Electro Mechanical System (MEMS) sensors suffer from inherent error characteristics errors such as the bias (drift), scale factor, cross-coupling, environmental conditions and random noise [12].

The accelerometers report reliable measurements at the lower frequencies and they are not discriminative for the braking mode classification. The reported measurements at the high frequencies are highly noisy and inaccurate when especially at the braking situations, whereas, those of gyroscopes are susceptible to low error levels at the higher frequencies [12]. Therefore, the rotation, roll and pitch angles derived from the measurements of the single sensor either exhibit a high signal-to-noise ratio or drift over a period of time (Fig. 5 and Fig. 6).

Both of the gyroscope and accelerometer sensors have pros and cons over each other in certain frequency ranges, therefore, the error stem from each sensor domain can be compensated by combining the measurements, namely sensor fusion.

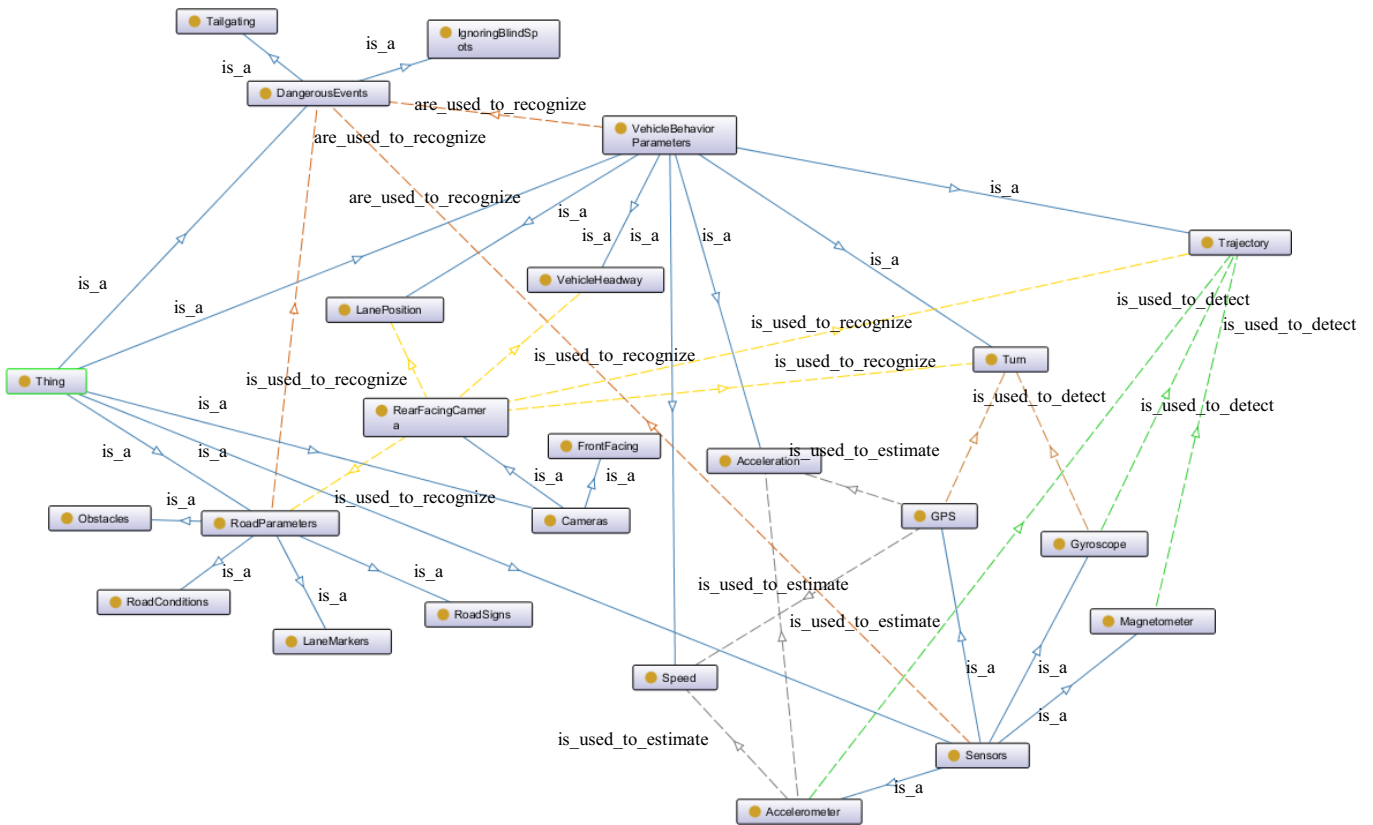


Fig. 3. Segway ontology model

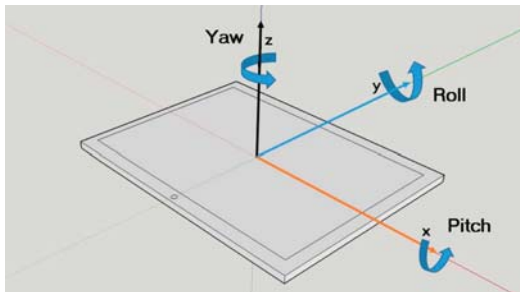


Fig. 4. Android-Based Smartphone Body Coordinate System

The Kalman or digital complementary filters can be used in that case. The use of Kalman filter incurs the system additional computational complexity, hence we opt for a complementary filter to obtain the filtered pitch (θ) and roll (ϕ).

The complementary filter is of 3rd order and easier to implement than the Kalman filter algorithms. The complementary filter exploits the low and high frequency regions of the accelerometer and gyroscope respectively which are complementary to each other (Fig. 7).

The pitch and roll angles are computed by using the accelerometer measurements of the each axis directions ($a_x; a_y; a_z$) subtracting the gravity vector appears on the each acceleration readings [12]. If no rotation is exist and

the smartphone lies down on the ground, the gravity vector is expressed in the form.

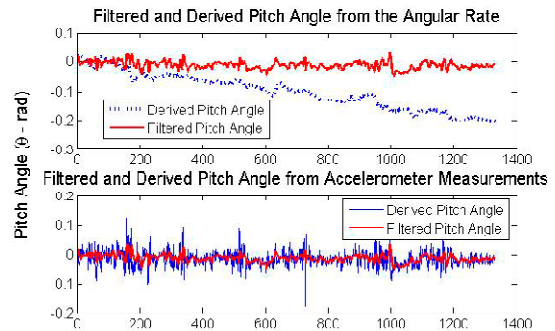


Fig. 5. Noisy and Filtered Pitch Angles Derived from the Motion Sensors

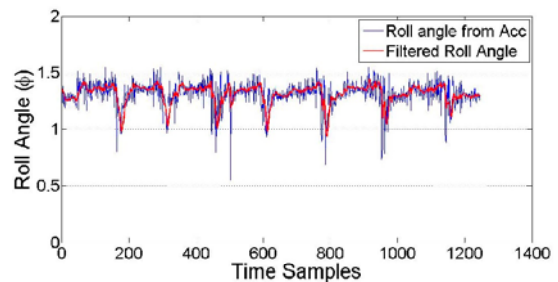


Fig. 6. The Roll Angles derived and filtered from the Accelerometer readings

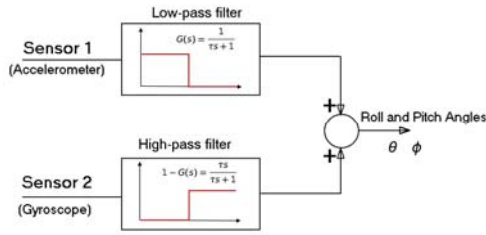


Fig. 7. Complementary Filter Block Diagram

Assuming that there are two sequential rotations around the x and y axes, and the rotation sequence is not known, two different rotation matrices can be obtained (1) and (2), as the multiplication of the rotation matrices for the pitch (3) and roll (4) are not commutative.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (1)$$

$$R_y = \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \end{bmatrix} \quad (2)$$

$$R_{xy} = \begin{bmatrix} \cos \phi & 0 & -\sin \phi \\ \sin \phi \cos \theta & \cos \theta & \cos \phi \sin \theta \\ \sin \phi \sin \theta & -\sin \theta & \cos \phi \cos \theta \end{bmatrix} \quad (3)$$

$$R_{yx} = \begin{bmatrix} \cos \phi & \sin \phi \sin \theta & -\sin \phi \cos \theta \\ 0 & \cos \theta & \sin \theta \\ \sin \phi & -\sin \theta & \cos \phi \cos \theta \end{bmatrix} \quad (4)$$

The last columns of the matrices in the (4) and (5) are the gravity vectors. In order to avoid ending up with two solutions for the pitch and roll angles, the range of either is restricted between $[-\pi, \pi]$. On the Android SDK, the roll and the pitch rotations around the y and x axes are restricted between the ranges $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and $[-\pi, \pi]$ respectively. In this case, using the trigonometric identities, the roll and pitch angles are expressed as given in the (5).

$$\begin{aligned} \phi &= \text{atan2}(a_y, a_z), \text{ and} \\ \theta &= \text{atan2}(-a_x, \sqrt{a_y^2 + a_z^2}). \end{aligned} \quad (5)$$

The computed pitch and roll rotations are given in Fig. 5 and 6. When computed in this manner, the roll and pitch rotation from the accelerometer readings exhibits a noisy characteristics in comparison to the filtered measurements as shown in the figures. We employ a Proportional-Integral (PI) controller to compute the coefficients of the complementary filter.

We first compute the rotations by integrating the gyroscope readings. The initial values are taken from the rotation angles computed by using the accelerometer readings. The difference between the rotations obtained from the gyroscope and accelerometer readings are the

inputs to the PI controller. The outputs are the filtered pitch and roll angles; the complementary filter outputs.

The transfer function for the estimated pitch and roll angles in this case becomes.

$$\phi_f = \frac{1}{s} \phi_g + \frac{K_p}{s} (\Phi_a - \Phi_f) + \frac{K_i}{s^2} (\Phi_a - \Phi_f) \quad (6)$$

After organizing (7), the transfer function for the roll and pitch angles is obtained as in (8).

$$\phi_f = \frac{s^2}{s^2 + K_p s + K_i} \left(\frac{1}{s} \phi_g \right) + \frac{K_p s + K_i}{s^2 + K_p s + K_i} \phi_a \quad (8)$$

The PI coefficients K_p and K_i must be fine-tuned. We exploit the PID tuning toolbox of Matlab to find the optimum filter coefficients. Once the transfer function and the coefficients are obtained, the difference equation for the digital filter are computed using the bilinear transformation.

The coefficients that Matlab's PID tuning toolbox gives are the optimum coefficients and the toolbox yields them as $K_p = 7.5924$ and $K_i = 20.7015$. After the bilinear transformation, the difference equation for the roll rotation (9) is obtained as follows.

In the (7), (8), and (9), ϕ_f , ϕ_g and ϕ_a represent the filtered roll rotations, the roll rotations obtained from the gyroscope and accelerometer measurements.

$$\begin{aligned} \phi_f &= \frac{(0.907z^{-1} - 1.814z^{-2} + 0.819z^{-3})\phi_g}{1 - 1.8091z^{-1} + 0.8188z^{-2}} \\ &+ \frac{(0.093z^{-1} + 0.004z^{-2} + 0.088z^{-3})\phi_a}{1 - 1.8091z^{-1} + 0.8188z^{-2}} \end{aligned} \quad (9)$$

The same difference equation is used to compute the filtered pitch angle. The resultant filtered rotation angles are given in the Fig. 5 and Fig. 6. As seen in the figures, the pitch and roll angles obtained from the trigonometric entities of gravity vector measurements of the accelerometer are highly noisy and it gives inaccurate values at the higher acceleration rates. In addition, the drift can be seen in the Fig. 5 on the integrated pitch and roll angle of the gyroscopes. The rotation matrix is obtained using these filtered pitch and roll rotations.

The accelerometer measurements are aligned with the body coordinate system using the rotation matrix, the input of which are the filtered rotation angles Fig.8 shows the aligned and un-aligned accelerometer readings. The dark curve represents the unaligned measurements in the body coordinate system in Fig. 8 while the other the aligned measurements. As seen in the figure, the gravity measured on the z -axis fluctuates around the gravity value during a sudden stop experiment.

VI. CONCLUSION

We present a reference model of a two-wheeled self-balancing vehicles rider assistant system in this paper. The model consists of five main modules: mobile application, cameras, sensors, local database and cloud service. These modules allow the system to recognize rider and vehicle behavior and producing alerts and warnings when dangerous situations are detected. We also detailed appropriate ontologies for the rider and vehicle behavior recognized as well as the Segway motion detection algorithms using an Android-based smartphone.

The smartphone-based solutions we elaborate here can be used for all types vehicles (new or old) in Segway category Manifesting itself as an affordable technology as a rider assistant system.

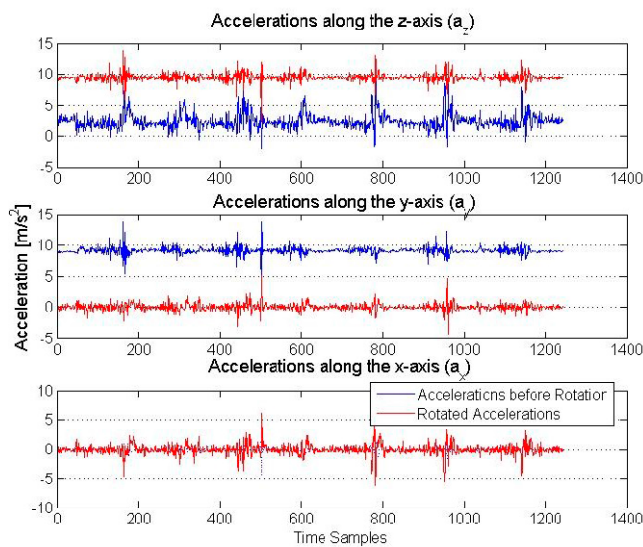


Fig. 8. Aligned and Unaligned Accelerometer Measurements during a Sudden Stop Experiment

ACKNOWLEDGMENT

The presented results are part of the research carried out within the project funded by grants # 13-07-00336, 13-07-12095, 13-01-00286 of the Russian Foundation for Basic

Research, program P40 “Actual Problems of Robotics” of the Presidium of the Russian Academy of Sciences. This work was partially financially supported by Government of Russian Federation, Grant 074-U01 and by the Japan Society for the Promotion of Science fellowship program.

REFERENCES

- [1] N. Hashimoto, K. Tomita, A. Kamimura, Y. Takinami, and O. Matsumoto, “Application for a personal mobility sharing system using two-wheeled self-balancing vehicles”, *International Conference on Mobility and Smart Cities*, 2014.
- [2] Cabinet Office, *White Paper on Traffic Safety in Japan*, 2006, p. 11, (in Japanese).
- [3] S. Miller, J. Molino, J. F. Kennedy, A. Emo, A. Do, “Segway Rider Behavior: Speed and Clearance Distance in Passing Sidewalk Objects”, *Transportation Research Record*, vol.2073, no.1, 2008, pp. 125–132.
- [4] S. Miller, J. F. Kennedy, J. Molino, A. Emo, G. Rousseau, C. Tan, A. Do, “Operating Characteristics of the Segway™ Human Transporter”, *Tech.rep.*, 2010.
- [5] H. Nishiuchi, Y. Shiomim, T. Todoroki, “Segway Running Behavior Focusing on Riders Experience Based on Image Processing Data”, in: *Proc. of the Eastern Asia Society for Transportation Studies*, Vol. 9, 2013.
- [6] N. Hashimoto, Y. Takinami, O. Matsumoto, “An Experimental Study on Vehicle Behavior to Wheel Chairs and Standing-type Vehicles at Intersection”, *Proc. 13th International Conference on ITS Telecommunications, IEEE*, 2013, pp. 350–355.
- [7] “Segway Review Report: A Review of Segway Use in the Australian Capital Territory”, *Tech. rep.*, *CT Government, Justice and Community Safety*, 2012.
- [8] “Pilot Project for Evaluating Motorized Personal Transportation Devices: Segways and Electric Scooters”, *Tech. rep.*, *Centre for Electric Vehicle Experimentation in Quebec (CEVEQ)*, 2004.
- [9] W. Wierville, “Overview of research on driver drowsiness definition and driver drowsiness detection”, *ESV*, Munich, 1994.
- [10] D. Dinges, M. Mallis, G. Maislin, J.W. Powell, “Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management”, *Department of Transportation Highway Safety Publication 808 762*, 1998.
- [11] Anon. Proximity array sensing system: head position monitor/metric. *Advanced Safety Concepts, Inc.*, Sante Fe, 1998.
- [12] A. Boyali, N. Hashimoto and O. Matsumoto, “A Signal Pattern Recognition Approach for Mobile Devices and Its Application to Braking State Classification on Robotic Mobility Devices”, *Journal of Robotics and Autonomous Systems*, [Unpublished, Under revision].