Journal of UAB ECTC
Volume 16, 2017
Department of Mechanical Engineering
The University of Alabama, Birmingham
Birmingham, Alabama USA

FINE-TUNING YOLO TO PERFORM A REAL-TIME VISUAL MULTI-OBJECT DETECTION TASK FOR AN AUTONOMOUS UNDERWATER VEHICLE

Alec Graves, Thomas Fagan, Steffen Lim, Kevin McFall PhD
Kennesaw State University
Marietta, Georgia, USA

ABSTRACT

In this paper, YOLOv2 is used to detect and classify underwater objects for use in the RoboSub competition. YOLOv2 is a Convolutional Neural Network designed for object detection, which recently achieved state-of-the-art results on standardized datasets such as VOC 2012 and MS COCO. This neural network has millions of tunable parameters and requires a significant amount of time and data to train from random initialization. To avoid this problem, the network described in this paper is initialized with the weights of a fully-trained YOLOv2 model and fine-tuned to perform detection of objects specific to the RoboSub competition; this is accomplished using only 1114 labeled images.

KEY WORDS: *Computer vision, Convolutional neural networks, Underwater object detection, Deep learning*

INTRODUCTION

The RoboSub Competition is an international competition that involves programming an autonomous underwater vehicle to perform various tasks in an obstacle course [1]. Teams earn points by successfully navigating through the course, filling containers, accurately shooting torpedoes, etc. Taking advantage of recent, major advancements in computer vision, the Kennesaw State University team is attempting to use deep learning for the competition.

Prior to using a neural network implementation, the competition team used various traditional image processing techniques. One such technique includes using OpenCV for color and shape detection [2]. Because of the dynamic nature of the underwater environment (e.g., image distortions, changing colors, reflections, etc.), the algorithm had trouble detecting objects consistently. With these effects having such dramatic consequences, a new object tracking system (resilient to the various distortions the water would create) was required.

A critical component of the competition is for the underwater robot to be able understand its relative orientation to the appropriate objects. One possible solution to this problem is using a convolutional neural network to locate these objects with a visual camera sensor. There are many such neural network designs that were able to successfully perform object detection and generate bounding boxes around them. Some of those methods include RCNN, Fast RCNN, Faster RCNN, YOLO, YOLOv2, YOLO9000 and SSD [3-8]. Though Faster RCNN and SSD have slightly higher accuracy than YOLOv2, the computational overhead is significantly higher [7]. Since real-time forward passes would be necessary for the RoboSub competition, YOLOv2, the most accurate network to meet those speed requirements, was chosen.

The best method for training the neural net to classify new objects using a limited amount of data is called fine-tuning. This method consists of taking a pre-trained model, stripping it of the last layer, and applying a new top layer with a custom number of outputs. This creates a network where the beginning consists of pre-trained weights and the end contains untrained weights. By keeping some of the pre-trained weights, the model has already learned relevant, low-level features for finding objects, and it reduces the time to train for the custom classes.

RELATED WORKS

The YOLOv2 is the current state-of-the-art CNN architecture for detection tasks [7]. The architecture has shown superior performance to other methods such as SSD, YOLO, and Faster RCNN on standardized datasets such as VOC [9] and MSCOCO [10]. Additionally, it can achieve real-time execution speeds on relatively accessible, inexpensive hardware. The network itself is based on an architecture called Darknet 19 [11], which has 19 convolutional layers. From Darknet 19, several modifications are made, including the addition of a pass-through layer, to reintegrate low-level features to the end of the network as well as an output layer with a custom loss function for learning box proposals [11].

YOLOv2 is very distinct from other approaches in its use of k bounding box priors which are gathered from k-means when estimating bounding box dimensions [7]. Instead of predicting box dimensions or coordinates directly, YOLOv2 predicts offsets from the width and height of the bounding box priors. Additionally, YOLOv2 outputs k box predictions for each spatial location on the output feature map and each output (corresponding to a spatial location of the output feature map) is trained to make predictions for a specific region on the input

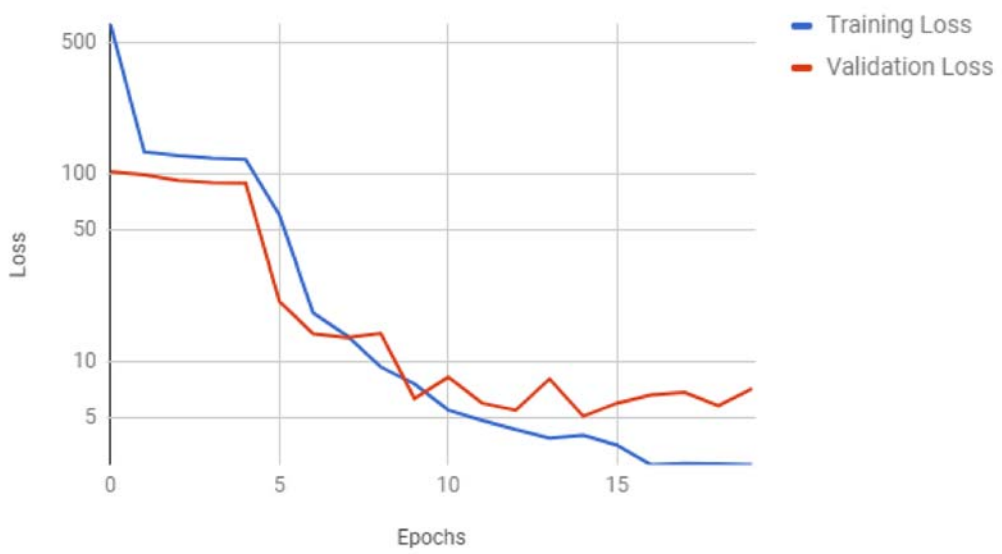

Figure 1. Validation and Training Loss

feature map. In this way, specific outputs correspond to specific input regions. Also, unlike many other architectures, each output has a receptive field the size of a large portion of the image, thereby allowing it to reason globally about what is contained in a specific region of the input image.

METHODOLOGY

Data Collection and Preprocessing

To train and fine tune the YOLOv2, new data was collected. The underwater vehicle has a few objects it needs to detect to successfully complete the RoboSub competition such as buoys, a start gate, a path marker, and a channel. To gather training images for the network, every 10th frame was taken from various videos of the competition that had been posted on the internet or donated. These videos were selected because they were taken by front-facing cameras during actual runs of the competition in previous years. Additionally, images of a red buoy, a green buoy, and a start gate were collected in a pool. An attempt to maximize different angles, varying lighting conditions, and different combinations of objects was made. Using a custom JavaScript tool, 111 validation images and 1002 training images are labeled [12].

The original RGB images are resized from 640 x 480 to 416 x 416 in order to accommodate YOLOv2's requirement for a multiple of 13, and every pixel channel value is normalized. Boxes gathered during the labeling process are rearranged into the format used to train YOLOv2 [7]. The greatest number of boxes labeled in a single training image is 6, so all-zero boxes are appended to training labels that have less than 6 boxes. This is done to assure a uniform label shape and easily allow training with larger batch sizes. This does not negatively affect the loss during training because the IOU (Intersection Over Union) score for these zero-boxes is zero.

Architecture Setup

The output of the customized YOLOv2 architecture has 5 boxes per spatial location of the output feature map, with 6 possible classes per box. Because Darknet 19 downscales the spatial dimension of the input by a factor of 32 and the images used have a shape of 416 x 416, the output feature map has a shape of 13 x 13. Additionally, each box has 5 parameters which describe its center location, its width and height offsets from the box priors, and the likelihood of the box prediction being correct. Accordingly, the custom output of the network is an array with the shape (13 x 13, 5 x (5 + 6)) = (169, 55).

Training Methodology

To begin fine-tuning, the weights from a fully trained YOLOv2 model are loaded. Next, the top layer of the network is removed and replaced with the custom output layer for the new detection task. Additionally, all parameters are frozen (made unchangeable) except for those in the last layer. After this, the network is trained for 5 epochs on the training images to allow the randomly initialized weights of the new output layer to adjust. After these epochs are complete, all layers are unfrozen and training takes place for 15-30 epochs. The validation loss is monitored and the best performing set of weights is saved and used in the final detector. A graph of the training loss versus the validation loss being minimized over epochs can be seen in Figure 1.

Testing Methodology

Several videos were created in the pool and none of the images extracted from them were used in the training processes. Out of 112 test images, only two were improperly classified, showing a success of more than 98%.

CONCLUSION

Comparing the training loss to the validation loss, it can be seen that the network has indeed learned rather than overfit to the data. Further, the test predictions, some of which are displayed in Figure 2, show that the network has potentially generalized in that environment.

The red buoy in the bottom, middle image of Figure 2 is understandably difficult to identify without context. In the other test failure, the lighting made the classification of a buoy vague.

The YOLO implementation used in this paper could evaluate images at a rate of 18.9 frames per second at 416 x 416 image resolution on a Nvidia GTX 1060 6GB GPU.

The results demonstrate that a pretrained YOLOv2 model can be fine-tuned for use as a real-time, multi-object detector on robots operating in aquatic environments. Additionally, presented data demonstrates the ability of this training technique to expand upon the network's prior knowledge and teach the network to recognize new objects, even with a limited amount of training data.

The next step is to apply the trained network in the actual RoboSub competition. It is expected to work just as well as the prototype described in this paper once labeled data from the site is used to further fine-tune the network.

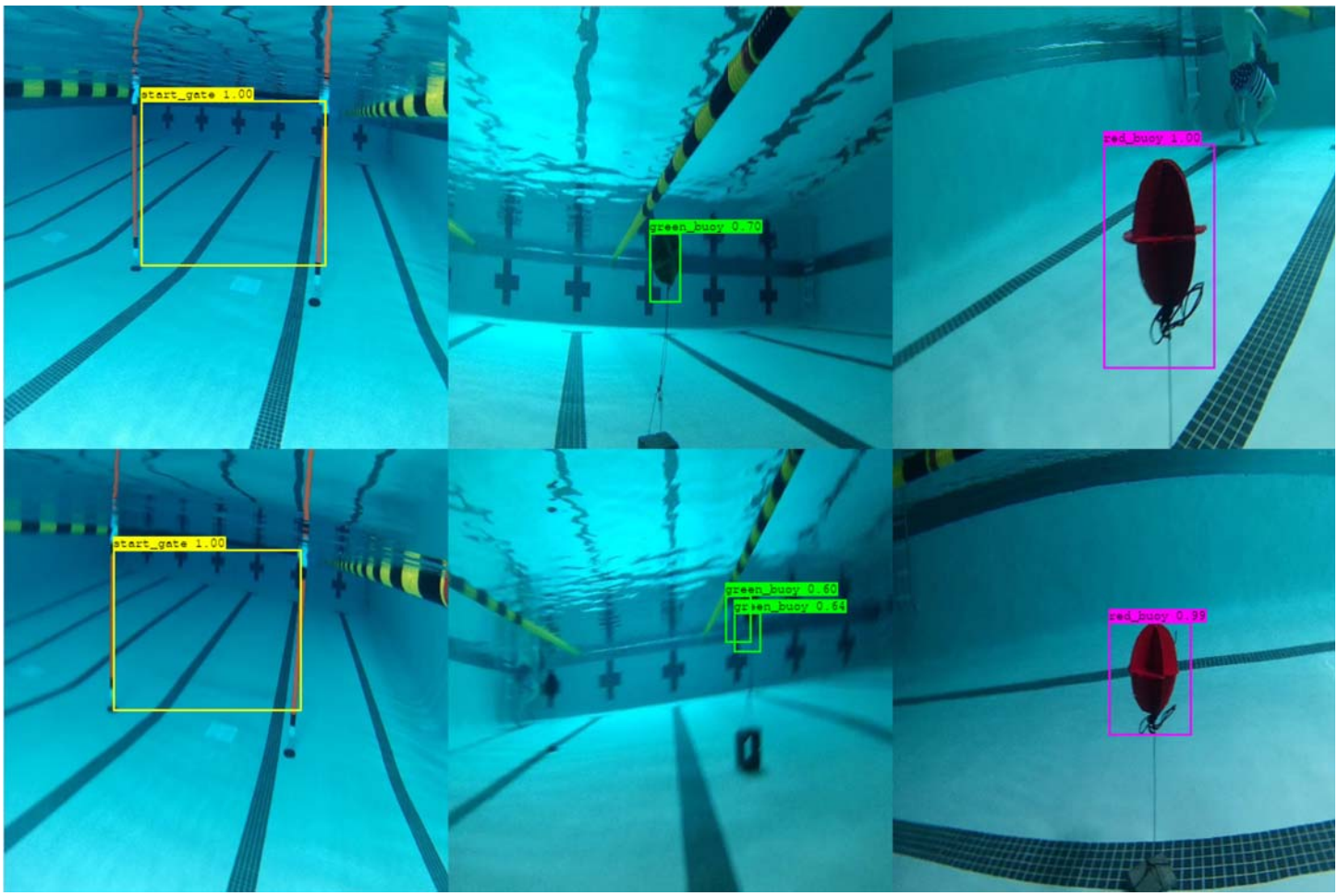

Figure 2. Test Results

ACKNOWLEDGEMENTS

This project was possible thanks to the RoboSub competition team members (Kevin Larose, NCSU, and Julio Guillén) who posted their test runs to Youtube, the Kennesaw State Autonomous Underwater Vehicle Team for helping to collect and label data, and the developers of YOLOv2 and its Keras port YAD2K (https://github.com/allanzelener/YAD2K) who made the code freely available.

REFERENCES

[1] RoboSub Final Mission and Scoring A Pirate's Life for Thee. (2016, May 9). Retrieved from http://www.robonation.org/sites/default/files/RoboSub_2016_Final_Mission%20and%20Scoring.pdf

[2] Bradski, G., 2000, "OpenCV library," Dr. Dobb's Journal of Software Tools. 2236121

[3] Girshick, R. B., and Donahue, J., and Darrell, T., and Malik, J., 2013, "Rich feature hierarchies for accurate object detection and semantic segmentation," CoRR, http://arxiv.org/abs/1311.2524

[4] Girshick, R. B., 2015, "Fast R-CNN," CoRR, http://arxiv.org/abs/1504.08083

[5] Ren, S., and Girshick, R. B., and Sun, J., 2015, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," CoRR, http://arxiv.org/abs/1506.01497

[6] Redmon, J., and Divvala, S. K., and Girshick, R. B., and Farhadi, A., 2015, "You Only Look Once: Unified, Real-Time Object Detection," CoRR, http://arxiv.org/abs/1506.02640

[7] Redmon, J., and Farhadi, A., 2016, "YOLO9000: Better, Faster, Stronger." ASME J. Tribol., 124(1), pp. 5-13.

[8] Liu, W., and Anguelov, D., and Erhan, D., and Szegedy, C., and Reed S. E., Fu, C. Y., and Berg, A. C., 2015, "SSD: Single Shot MultiBox Detector," CoRR, http://arxiv.org/abs/1512.02325

[9] Everingham, M., and Van~Gool, L., and Williams, C. K. I., and Winn, J., and Zisserman, A., 2012, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[10] Lin, T., and Maire, M., and Belongie, S. J., and Bourdev, L. D., and Girshick, R. B., and Hays, J., and Perona, P., and Ramanan, D., and Dollár, P., and Zitnick, C. L., 2014, "Microsoft COCO: Common Objects in Context," CoRR, http://arxiv.org/abs/1405.0312

[11] Redmon, J., 2013–2016, "Darknet: Open source neural networks in c." http://pjreddie.com/darknet/

[12] Underwater Data Labeler (2016, May 9). Retrieved from https://github.com/shadySource/shadysource.github.io

Journal of UAB ECTC
Volume 16, 2017
Department of Mechanical Engineering
The University of Alabama, Birmingham
Birmingham, Alabama USA

AN ANALYSIS OF AN OPEN ARCHITECTURE SYSTEM APPROACH TO THE INDIVIDUAL PROPULSION SYSTEM OF ELECTRIC VEHICLES

Bing Wang, Siyuan Zhang, Vladimir Vantsevich
School of Mechanical Engineering
University of Alabama at Birmingham
Birmingham, AL, USA

Linhui Zhao
Department of Control Science and Engineering
Harbin Institute of Technology
Harbin, Heilongjiang, China

ABSTRACT

The Intelligent wheel driving system (IWDS) is one of the subsystems of an entire electric vehicle system. In IWDS, all conventional subsystems of the vehicle are reduced into one wheel boundary. All subsystems are integrated based on the concept of an open architecture system, which considers the coupling of the subsystems. Due to the functions of all subsystems in IWDS, their merged action directly affects the vehicle's operational properties, such as handling, mobility, safety, etc. Moreover, each property is simultaneously affected by several subsystems. The propulsion system with one e-motor, as the major subsystem in IWDS, not only provides the longitudinal motion of the vehicle but also affects the other vehicle properties. In this paper, the propulsion system installed in an IWDS will be described, and the motor parameters and their influences on the motor characteristics are analyzed. This study leaves the potential to analyze the influence of the merged action of all subsystems on the entire vehicle in the future.

KEY WORDS: *Electric vehicle, open architecture system, Individual propulsion system, e-motor*

INTRODUCTION

As is commonly known, a complete system contains countable subsystems that work for different purposes. All subsystems are integrated and generate a merged action that affects the entire system. Moreover, each subsystem in the entire system might also interact with each of the others. The open architecture system (OAS) mainly focuses on considering the coupling of each of the subsystems and the interactions between them. In electric vehicle engineering, the design of the vehicle drive system employs this OAS concept while considering the coupling and interactions of each subsystem.

In conventional vehicles, the subsystems installed in a powertrain system are the internal combustion engine (ICE), transmission, driveline system, steering system, etc. All subsystems are mechanically linked. Each of these subsystems is designed and manufactured by different individual departments, and assembled together into the powertrain layout afterwards [1]. Each subsystem-design department can upgrade its subsystem in order to improve the vehicle's operational properties from different angles, such as mobility, energy efficiency, stability, turnability, etc. Moreover, because each vehicle's operational properties are affected not by only one subsystem but by several, many studies have considered the coupling of the individual subsystems. For example, the suspension system directly influences both the turnability and the stability of motion [2]. Beloousov et al. [3] proved that the suspension system also impacts the tire traction force and the normal reaction force. Another study demonstrated that the vibration of the sprung and unsprung mass in the suspension system could impact the wheel traction and braking dynamics [4, 5]. In addition, Vantsevich et al. [5] proved that the driveline system and suspension system affect each other. As a result, with considering the coupling of all subsystems and their merged actions, the OAS concept is applied to vehicle design.

However, in recent decades, with drastic development of electric vehicles (EV), a new type of intelligent wheel driving system (IWDS) is replacing the conventional drive system. All subsystems of conventional powertrain systems are replaced and integrated within one wheel boundary [6, 7]. IWDS subsystems installed in an EV are not mechanically linked with each other but are individually controlled or are controlled by a central control unit [8]. Similar to conventional vehicles, the couplings of all subsystems in IWDS are supposed to be concerned, and the propulsion subsystem, as a major component, directly affects the dynamic system of IWDS. As a result, in this study, an analysis of the propulsion system of an IWDS will be presented, in which the motor parameters and their influence on motor characteristics will be analyzed. This study leaves a potential to analyze the coupling of the subsystems in IWDS and their merged action to the entire IWDS in the future.

INDIVIDUAL PROPULSION SYSTEM OF ELECTRIC VEHICLES

The intelligent wheel driving system (IWDS) is one subsystem of an entire electric vehicle (EV) system. Figure 1 presents a typical example, Active Wheel produced by Michelin Company. It can be seen from Fig. 1, in an IWDS, all conventional subsystems of powertrain and chassis are compressed into one wheel within the wheel's boundary, but all subsystems are driven by electricity.

Figure 1 Michelin Unveils Active Wheel

In an IWDS, the propulsion system with one motor provides the wheel dynamic motion in the longitudinal direction. The required torques and angular velocities are given to the wheel by the e-motor. In a conventional vehicle, the throttle plate of the ICE is mechanically linked with the accelerator, and the driving wheel is propelled by pressing the accelerator to determine the position of the throttle plate in the engine cylinders. However, in an IWDS, an electric motor is used in place of an ICE. The corresponding function to ICE in IWDS is a by-wire system [8], which controls the angular velocity and the driving torque of the e-motor. The angular velocity and driving torque are calculated by one or several control units, depending on the driver inputs and the information obtained from different wheel sensors, such as pressure, acceleration and position sensors, etc.

Many types of e-motors are employed in EVs. Two major types are the alternating current (AC) motor and the direct current (DC) motor. Table 1 lists the characteristics of AC motors and DC motors.

Table 1 Characteristics of AC motors and DC motors

AC Motor	DC Motor
Single-speed Transmission	Multi-Speed Transmission
Light Weight	Heavier at Equivalent Power
Less expensive	More Expensive
95% efficiency at full load	85-95% efficiency at full load
More expensive controller	Simple controller
Motor/Controller/Inverter more expensive	Motor/controller less expensive

As can be seen from Table 1, AC motors are high efficiency and light weight. Thus, many EVs choose AC motors to install in their propulsion system. However, DC motors have simpler controllers, and their mathematical models are simpler. Considering this advantage of DC motors, several EVs employ DC motors. In this study, the characteristics of a DC motor will be analyzed.

MATHEMATICAL MODEL OF DC MOTORS

As mentioned above, the electric motor provides the required torques T_w and angular velocities ω_w to the wheel. The angular velocity ω_w can be determined by the mathematical model of the DC motor. The unit of a DC motor can be considered as a set of two subsystems: the electric system and the dynamic system, which are presented in Fig. 2,

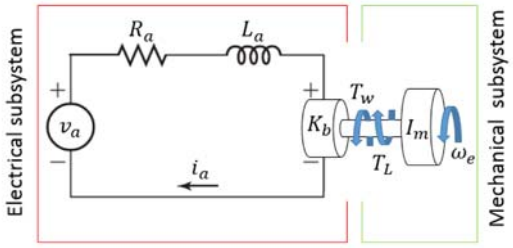

Figure 2 Subsystems of an unit of DC motor

Here, R_a is the armature resistance, L_a is the armature electric inductance, v_a is the supplied voltage, K_b is the motor's back emf, i_a is the current. I_m is the motor's inertia, ω_e is the angular velocity of the motor, T_L is the load torque. The mathematical models of both systems are described in Eqs. (1) and (2) respectively.

$$i_a R_a + L_a \frac{di_a}{dt} + K_b \omega_e = v_a \tag{1}$$

$$I_m \frac{d\omega_e}{dt} = K_b i_a - c\omega_e - T_L \tag{2}$$

Here,

$$K_b i_a = T_w \tag{3}$$

Solving the differential Eqs. (1) and (2), the angular velocity of the wheel can be determined.

When the motor is in a steady-state situation, the electric inductance L_a becomes zero. As a result, substituting Eqs. (1) and (3) into Eq. (2) yields,

$$\omega_e = \frac{v_a}{K_b} - T_w \frac{R_a}{{K_b}^2} \tag{4}$$

As can be seen from Eq. (4), the driving torque of an e-motor can be determined while knowing the motor's angular velocities. Thus, specifying the values of the motor parameters and solving the set of Eqs. (1)-(4) can obtain the required torque and angular velocity of the driving wheel while the wheel is moving in specific terrain conditions.

SIMULATION RESULTS

The previous section indicates that specifying the parameters of e-motors can yield the required driving torque and angular velocity for a wheel. As a result, in this section, several simulation results are presented. These results leave clues to readers to design the e-motors, by changing the parameters, in order to obtain the torques and velocities. It should be mentioned that the load torque T_L relates to the terrain conditions. However, this study only considers the characteristics of motors. As a result, load torque T_L is considered as zero while simulating the mathematical models. Table 2 presents a set of parameters of DC motors. Set 1 is considered as the standard motor. In sets 2-7, the value of one parameter decreases while the others are held constant. Figures 3-10 present the simulation results.

Table 2 Tested parameters

Set #	K_b	V_a	L_a	R_a	c	I_m
1	2	450	0.0012	0.25	0.3	0.3
2	1↓	-	-	-	-	-
3	-	350↓	-	-	-	-
4	-	-	0.00012↓	-	-	-
5	-	-	-	0.05↓	-	-
6	-	-	-	-	0.2↓	-
7	-	-	-	-	-	0.2↓

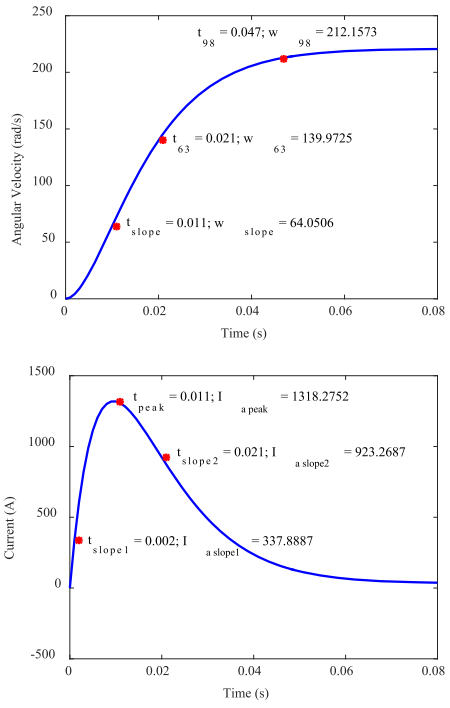

Figure 3 Simulation results of set 1

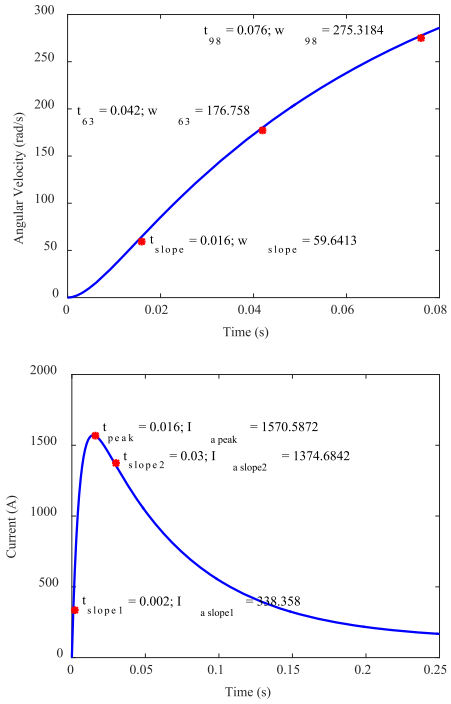

Figure 4 Set 2 $K_b = 1\ N \cdot m/A$

Figure 4 provides the simulation results when $K_b = 1\ N \cdot m/A$. Comparing these results with those in Fig. 3, it can be found that the greater the values of parameter K_b, the faster the response of the angular velocity, and the smaller the maximum value of the current and angular velocity.

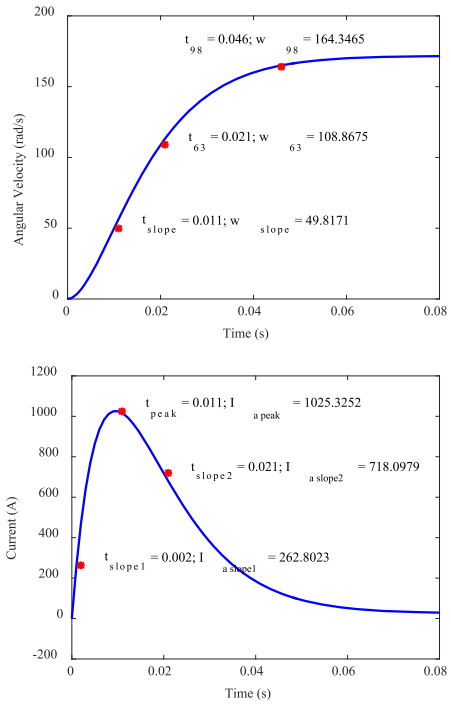

Figure 5 Set 3 $V_a =$ 350 V

Figure 5 presents the simulation results when V_a is 350 V. Comparing with these results with those in Fig. 3, it can be found that the change of the voltage has little influence on the response of the angular velocity, and the greater the values of voltage, the larger the maximum value of the current and angular velocity.

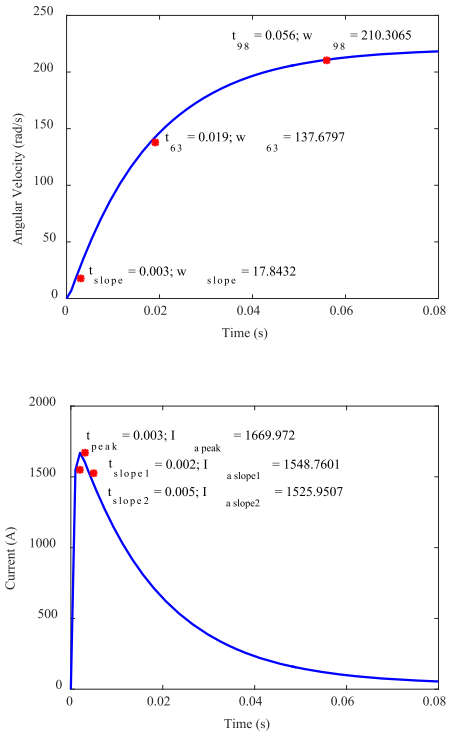

Figure 6 Set 3

Figure 6 presents the simulation results when $\boldsymbol{L_a} = 0.00012\ H$. Comparing these results with those in Fig. 3, it can be found that the greater the values of parameter L_a, the faster the response of the angular velocity, and this has little influence on the steady-state value of the angular velocity. Comparatively, the change of L_a is most significant to the maximum current of motor, and the greater the values of parameter L_a, the greater the maximum current.

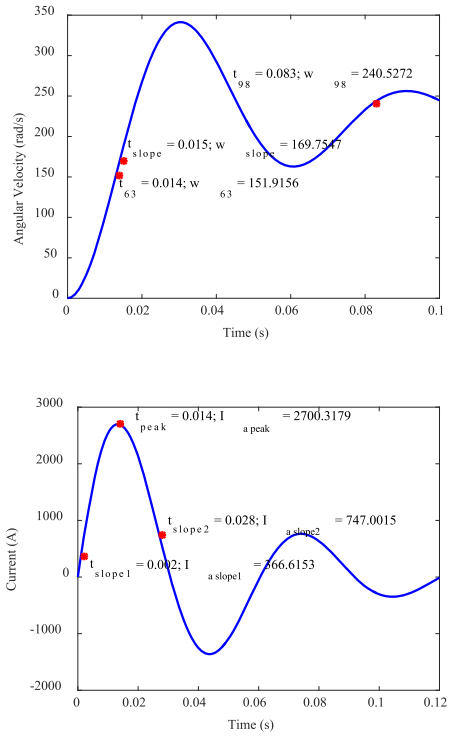

Figure 7 Set 4

Figure 7 shows the simulation results when $R_a = 0.05\ \Omega$,. Comparing these results with those in Fig. 3, it can be found that the smaller the values of parameter R_a, the faster the response of the angular velocity, and this has little influence on the steady-state value of the angular velocity. Comparatively, the change of R_a is most significant to the maximum current of motor, and the smaller the values of parameter R_a, the greater the maximum current.

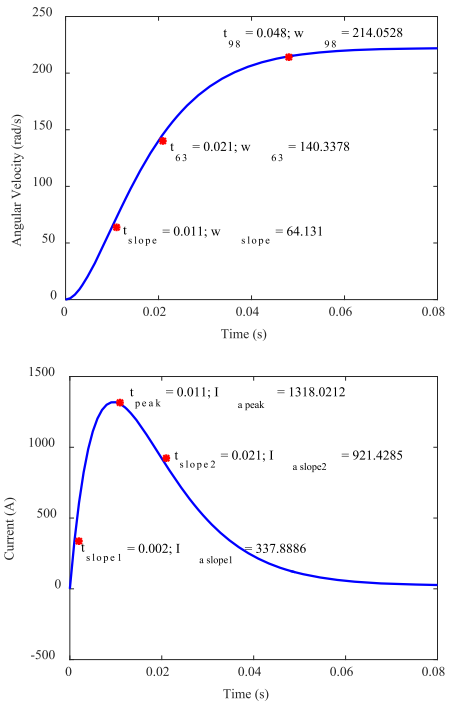

Figure 8 Set 5

Figure 8 shows the simulation results with $c = 0.2\ N \cdot m \cdot sec/rsd$. Comparing these results with those in Fig. 3, it can be found that the change of the damping has little influence on the response times and steady-state values of angular velocity and current.

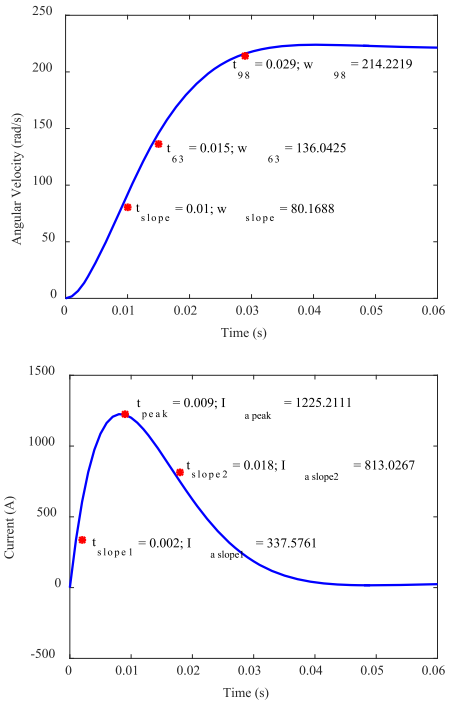

Figure 9 Set 6

Figure 9 shows the simulation results with $I_m = 0.2\ kgm^2$. Comparing these results with those in Fig. 3, it can be found that the smaller the values of parameter I_m, the faster the response of the angular velocity, and this has little influence on the steady-state value of the angular velocity. Comparatively, the change of I_m has a little more influence on the maximum current of motor, and the smaller the values of parameters I_m, the smaller the maximum current.

Table 3 presents the resultant values of the torques of different sets of motor parameters.

Table 3 Generated torques of different sets

Sets	1	2	3	4	5	6	7
Torque(Nm)	3600	1800	2800	3600	18000	3600	3600

It can be seen from Table 3 that the generated torques of a motor keep constant while the motor inductance, damping and inertia change (See sets 4, 6 and 7). Moreover, decreasing the motor's back emf and voltage directly decrease the generated torque (set 2), while decreasing the motor resistance increases the generated torque.

In summary, with the simulation results based on the mathematical model of a DC motor, the trend of generated torques and angular velocities when changing the motor's parameters can be found.

CONCLUSION

This study introduces the new type of individual propulsion system in IWDS of electric vehicles. This study indicates that the motor, as the major part of individual propulsion system, delivers torques and angular velocities to the driving wheels. This torque and angular velocity directly affect the vehicle's operational properties. As a result, the mathematical model of a DC motor is presented in the 4th section. According to the mathematical model, the torque and angular velocity generated by the motor can be obtained based on the known motor parameters.

Moreover, when EVs move on different terrain conditions, the motor should provide suitable torques and velocities to the wheel in order to meet the driving conditions. The 5th section presents the simulation results of different sets of motor parameters and provide the way that the parameters affects the motor characteristics. This study leaves a potential to design the motor for different terrain condition.

Since the motor characteristics (angular velocity and driving torques) directly impact the wheel's longitudinal dynamic motion, the next step of this study is to combine the mathematical models of motor with the differential equations of dynamic motion of an IWDS. This study could be used to analyze and consequently control the IWDS longitudinal dynamic motion. This research is ongoing.

REFERENCES

[1] A. F. Andreev, V. I. Kabanau, V. V. Vantsevich: Driveline of Ground Vehicles: Theory and Design; V. V. Vantsevich, Scientific and Engineering Editor; Taylor and Francis Group/CRC Press, 792 pages, 2010.

[2] Vladimir V. Vantsevich. 2015, “Road and off-road vehicle system dynamics. Understanding the future from the past”, Vehicle System Dynamics 53:2, pages 137-153.

[3] Beloousov, B., Ksenevich, T., and Vantsevich, V., "Load Estimation of an Open-Link Locomotion Module for Robotic and Commercial Multi-Wheel Applications," *SAE Int. J. Commer. Veh.* 6(2):301-307, 2013

[4] Soliman, A., Kaldas, M., and Mahmoud, K., “Active Suspension and Anti-lock Braking Systems for Passenger Cars,” SAE Technical Paper 2009-01-0357, 2009, doi:10.4271/2009-01-0357.

[5] Vantsevich, V. V., Stuart, Ch., “Probabilistic Interaction between Vehicle and Surroundings: Modelling for Control”, IMECE2007-44071, ASME Congress, Seattle, Washington, November 10-16, 2007

[6] Zeid A, Chang D. A modular computer model for the design of vehicle dynamics control systems. Veh Syst Dyn. 1989;18(4):201–221.

[7] J. Hag. Wheel Corner Module. Master Thesis, Royal Institute of Technology, 2011.

[8] E. Bretz, _By-wire cars turn the corner,_ in IEEE Spectrum 2001, vol. 38, no. 4, pp.68 _ 73, ISSN 0018-9235.