International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:14, No:6, 2020

Self-Organizing Map Network for Wheeled Robot Movement Optimization

Boguslaw Schreyer

Abstract**—**The paper investigates the application of the Kohonen's Self-Organizing Map (SOM) to the wheeled robot starting and braking dynamic states. In securing wheeled robot stability as well as minimum starting and braking time, it is important to ensure correct torque distribution as well as proper slope of braking and driving moments. In this paper, a correct movement distribution has been formulated, securing optimum adhesion coefficient and good transversal stability of a wheeled robot. A neural tuner has been proposed to secure the above properties, although most of the attention is attached to the SOM network application. If the delay of the torque application or torque release is not negligible, it is important to change the rising and falling slopes of the torque. The road/surface condition is also paramount in robot dynamic states control. As the road conditions may randomly change in time, application of the SOM network has been suggested in order to classify the actual road conditions.

Keywords**—**SOM network, torque distribution, torque slope, wheeled robots.

I. Introduction

THE safe, stable and efficient emergency braking of wheeled robots and vehicles is best ensured by anti-skid systems. These systems, from the point of view of control theory, are the optimal controllers that minimize the path and time of braking. At the same time, they ensure sufficient (also not perfect) transversal stability of braking and starting robots. The braking robot and wheel-road mechanics are strongly nonlinear with parameters varying in function of linear speed of the robot, adhesion coefficient, weight and geometry of the vehicle, state of tires and random road and weather conditions. For these reasons, all existing systems are in fact quasi-optimal. In most road surfaces and weather conditions, the road/wheel coefficient of adhesion has its distinct maximum in function of the wheel slip. The essence of the optimal and adaptive controller is to keep this adhesion near the maximum at all wheels, during the process of braking/starting. It should not be forgotten, that while minimizing path and time of braking/starting the robot, the transversal stability is also important. This is why in a suggested R-P anti-skid system controller an extra signal is taken into account: the alteration of transversal acceleration ΔA. This parameter is important, as the transversal instability may result with loss of minimum time, minimum path, and in critical circumstances may even cause a robot rollover. The neural tuner introduces the adaptive properties to the already existing quasi-optimal regulator. The input pattern for the neuromorphic tuner is the set of measured variables: vehicle speed and vehicle deceleration, the most important factors that affect the controller parameters. The neural network has six outputs: five of them corresponding to the parameters of the control unit and one corresponding to the braking moment modulator. The neural network training set contains the inputs and desired network outputs, being the optimal parameters of the controller, obtained earlier on the way of computer modelling.

Boguslaw Schreyer is with the Nipissing University, Canada (e-mail: bjs@nipissingu.ca).

II. An R-P Anti-Skid System Controller

In general terms, many of the existing anti-skid system controllers are of the on-off type and include two inputs: signals of torque application: R (called reselection) and torque release P (called prediction) [3]. The resulting control signal C, that turns on-off a braking moment modulator, satisfies the following Boolean relation:

$$C \equiv R \wedge (\sim P) \wedge (\sim \Delta A) \tag{1}$$

In the definitions of the existing R and P signals, the most frequently used variables are: robot longitudinal velocity, its deceleration, wheel speed, wheel deceleration/acceleration and lapse of time since P signal disappears. Among systems of reselection, three of them combined appear to be the most efficient: R_1, which produces its logic 1 for brakes application after certain constant time t1 elapsed since the moment when signal P vanishes (or appears), or since wheel acceleration becomes positive. The first solution has been applied in Lucas-Girling system WSP. Second, R2, satisfies the relation $R2 \equiv \omega \geq k_2 * \omega_h$, where, ω is the wheel rotational velocity, and ω_h is the same velocity in the moment of brakes application. Next, R3 satisfies the following relation $R_3 \equiv \omega' \geq k_3 * \omega_h$; where, ω' is the wheel deceleration, k83 > 1.

Among systems of prediction, the following two were the subject of our particular attention: $P1 \equiv -\omega\dot{\omega} \geq k1/R$; where, R is a radius of the wheels and $P_2 \equiv -\omega\dot{\omega}/\omega \geq k_4$; where, k_4 is admitted around a value of 3, sometimes affected by road conditions.

Computer modelling of all existing R and P systems in the process of braking led to the conclusion that, the above systems give the best results in terms of braking path and time as well as vehicle stability during braking. In existing anti-skid systems, the t_1, k_1, k_2 and k_3 are constant. Computer simulation has proven that the combined systems containing more than one R system and P system give the best results of braking. It appeared that, the reselection signal (equal the disjunction of component reselections) R = R1 V R2 V R3 as well as combined prediction signal (equal a conjunction of the two prediction signals): P = P1 Λ P2 secure close to minimum braking/starting path among all existing P-R systems C ≡ R Λ(~ P). It has been decided that, in order to secure transversal stability of the robot, minimum of braking path and braking time, the control signal C should include also a signal of transversal acceleration alteration ΔA: C ≡ R Λ (~ P)Λ(~ ΔA). The signals P and R depend on five constants. The numerical optimization allowed to determine the relation between those constants (t_1, k_1, k_2, k_3, k_4) explained further and robot longitudinal velocity as well as robot longitudinal deceleration. These optimal relations represent the foundation for neural tuner.

III. First Attempt: Back-Propagation Neural Network

The feedforward back-propagation neural network applied to our controller is considered as a multilayer system of interconnected simple perceptron (neuron-like) processing elements, with weighted interconnections. The input vector is applied to the network at the input layer. The input layer has as many neurons, as many inputs as needed for specific application. Next, intermediate layers follow and the last, output layer produces the output of the neural network. The intermediate output of the j_{th} neuronlike element is described by the linear function. The resulting output of the neuron is obtained as an image of a saturation nonlinear function. In our case it is a hyperbolic tangent function:

The interconnection weights represent a kind of memory. The network training can provide the way of adapting the weights to perform the given input signal processing task. The most popular technique called back propagation error learning [4] has been implemented for the neuromorphic tuner learning. As a performance criterion of learning the total error measure of network over all training set TR of training cases has been considered. An error of the network Ec for a given case c is expressed by:

$$E_c = 0.5 * \sum_{j=c}^{n} (d_{j,c} - y^0_{j,c})^2$$

where, $y^0_{j,c}$ is the actual network's output for the input from case C_c and $d_{j,c}$ is desired output for this case. The training case Cc is defined as a pair of input $x_{i,c}$ and desired output $d_{j,c}$: $C_c = (x_{i,c}, d_{j,c})$, I = 1,2,...,m; j = 1,2,...,n; c = 1,2,...,tr, where, m- number of network inputs, n- number of network outputs, tr- number of training sets. The training set TR = $\{(x_{i,c}, d_{j,c})\}$, c = 1,2,...t_r includes the t_r training cases C_c to minimize the criterion E, the gradient descent method has been adopted [7]. The momentum formula has been adopted for updating the weights values according to the error terms and signals:

$$w_{ji}(t+1) = w_{ji}(t) + \eta\delta_j y_i + \alpha(w_{ji}(t) - w_{ji}(t-1)),$$

where η is the learning procedure tuning parameter, and momentum term α is the exponential decay tuning factor from the range [0,1].

IV. Neuromorphic Tuner for Adaptive Anti-Skid Controller

The reasonable range of robot longitudinal velocity Vi: [0,50] km/h has been divided into 25 even ranges. Each range having its own number from the range [0,1], every 2 km/h. The robot deceleration $\dot{V}$ has been divided into two ranges less than or equal 0.5 g or greater than this value, and robot mass M has been divided into two even ranges from: {[0,100kg], (100,200kg]}. All range numbers create a set of tr = 25*2*2 = 100 distinct ordered triples. A multilayer feedforward neural network has been considered, with three inputs. The tr is a number of training cases for the neural network. The desired network outputs are the constants t_1, k_1, k_2, k_3, k_4, sp. The output sp affects the rising/falling slopes of the independent (front T_f or rear T_r) braking/driving moment. The dependent braking/driving moment (rear or front) is determined according a formula developed in the next chapter. The relationship between vehicle velocity and its deceleration and all the above factors has been determined by means of computer optimization. The neural network training set [1] contains the following cases:

$$TR = \{(Vic, \dot{V}ic, M),(t_{1c}, k_{1c}, k_{2c}, t_{3c}, k_{4c}, sp_c)\} \quad (2)$$

where, c = 1,2, ..., tr , is a case number and t_{1c}, k_{1c}, ... , sp_c are desired network output being optimal controller parameters for the input ordered triple (V_{ic}, $\dot{V}_{ic}$, M). The trained ANN output is the controllers constant, which in turn defines the P and R signals. The robot-controller has been numerically simulated. The results obtained are very encouraging. Robot braking path has been decreased with simultaneously good vehicle stability. The neuromorphic tuner, without changing its architecture, can be easily trained for a variety of vehicles after the optimal relationship between V, $\dot{V}$, M and P-R controller parameters have been established (Fig. 1).

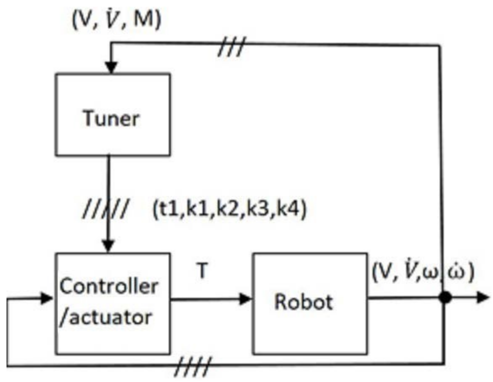

Fig. 1 Robot control loop with a tuner

Assuming identical ground adhesion coefficient under the front and rear wheels, the following braking/driving torque distribution secures the minimum braking/starting path [5]:

$$T_f = T_r - \frac{\dot{V}}{g} R (N_f - N_r) \quad (3)$$

where, T_f is a front torque, T_r - a rear torque, R is robot wheels' radius, $\dot{V}$ is robot deceleration, g is gravity acceleration, N_f is front axis vertical force, and N_r is rear axis vertical force. Assuming the same friction coefficient under front and rear wheels, the above torque distribution secures the same slip of the front and rear wheels, and therefore the possibility that all wheels have the same friction coefficient at the same time, and as a result, all wheels reach the friction coefficient maximum at the same time (Fig. 2), no matter what the road conditions are. Also, the transversal friction coefficient is high.

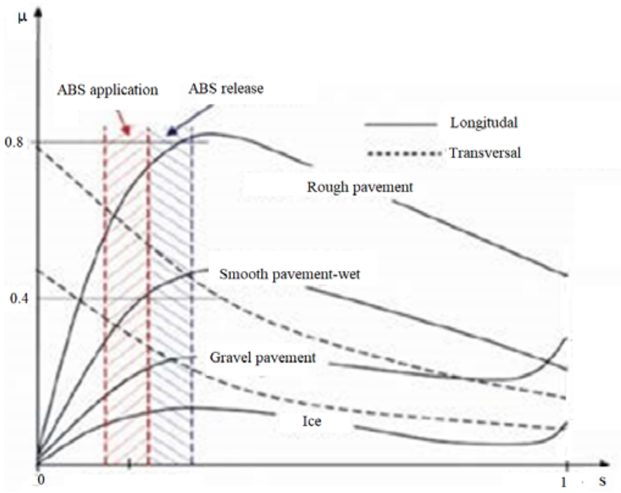

Fig. 2 μ(s) for variety of road conditions

Assuming symmetry of road conditions and robot mass distribution, and the torque distribution according to [2], the robot model can be brought to a 1-mass model, as in Fig. 3, where T is the total torque: $T=T_f+T_r$, M is total mass of the robot, $N=N_f+N_r=Mg$ - vertical force is equal to the robot weight, ω is rotational wheels' velocity, I is moment of inertia of all wheels, V is robot longitudinal velocity, and F is total friction force. This force is positive for driving state and negative for braking.

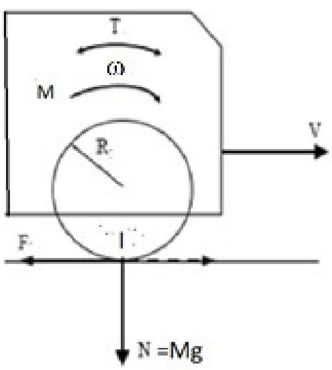

Fig. 3 Reduced, 1-mass robot model

This simplified model is correct under all symmetry assumptions. When the asymmetry of friction coefficient between left and right wheels occurs, an anti-skid system should take a decision as to which friction coefficient will be taken as a reference- the lower or the higher one. In the first case, we call it Select-Low; in the second case we call it Select-High. In any of the two cases, a transversal acceleration appears, parameter ΔA of the control system is activated, and for its critical value, the torque should be released.

IV. SOM Network for Braking/Driving Moment Slope Selection

It has been shown [2], [6] that, in case of some delays in braking/driving torque application, the torque slope affects the efficiency of robot braking/starting process. If the road friction coefficient is low, a higher torque slope may cause passing over the maximum η_m of the friction coefficient and reaching an instable area of friction in function of the slip s: μ(s). In that case, a slower torque T increase (lower positive slope value) is recommended. At the same time, faster escape from the instable μ(s) area is also advised, i.e., a lower negative torque value (higher absolute slope value). Contrary, when the friction coefficient is higher, the torque T may increase faster (i.e. bigger positive slope value), and then torque release may be slower (higher negative slope value). Therefore, assessment of the friction coefficient is important. When the friction coefficient is higher, the robot deceleration during breaking is lower. In order to assess the friction value, the deceleration has been taken into account. At every breaking torque application, i.e., during the torque increase, the 10 values of deceleration $\dot{V}$ are measured. Those 10 values may change randomly within a certain limited range at every torque application in the function of the road condition. This is why an artificial neural network has been applied, thanks to its generalization/tolerance capabilities [4]. Those values are submitted to the input of the SOM (Kohonnen Self-Organizing Maps) network in order to select a correct torque slope. The SOM weight vector for each cluster unit serves as an exemplar of the input pattern (10-tuple) associated with that cluster. The cluster units are arranged in a one-dimensional array. During the self-organization process, the cluster unit whose weight vector matches the input pattern most closely (the square of the minimum Euclidean distance), is chosen as a winner. The winning unit and its neighboring units (in terms of topology of the cluster units) update their weights. During the self-organization process, the 10-tuple of the robot deceleration are taken from the computer model of the robot. Every 10-value pattern then is classified into one of the five clusters (Fig. 4).

Every classified cluster is represented by a maximum output value. The unique maximum output represents one out of five friction coefficients, and also one out of five torque T, rising and falling slope. When the SOM network is applied to the robot control, its unique output is indicating which torque slope should be applied.

SOM algorithm:

Step 0 Initialize weights w_{ij}
Set topological neighborhood parameters

Set learning rate parameter α

Step 1 While stopping condition is false, do Steps 2-8

Step 2 For each input 10-tupple $\dot{\boldsymbol{V}}$, do Steps 3-5

Step 3 For each j, compute: $D(j)=\sum_j (w_{ij} - V_j)^2$

Step 4 Find index j such that D(j) is a minimum

Step 5 For all units j within a specified neighborhood of J, and for all i: $w_{ij}(new)=w_{ij}(old)+\alpha[V_i-w_{ij}(old)]$

Step 6 Update learning rate α

Step 7 Reduce radius of topological neighborhood at specified times

Step 8 Test stopping condition.

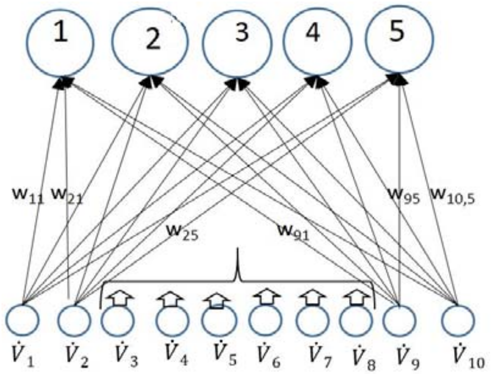

Fig. 4 SOM network

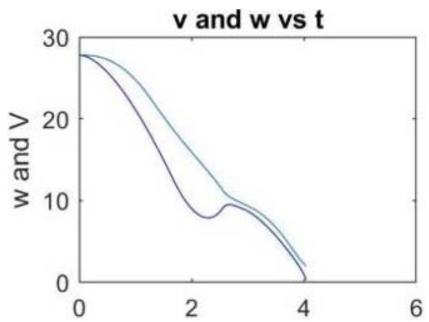

Fig. 5 Icy road, torque slope too high

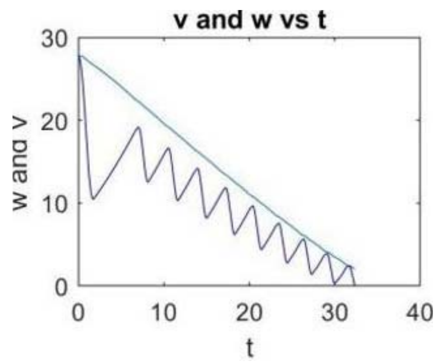

Fig. 6 Icy road, correct torque slope (500 Nm/s, 3000 Nm/s)

The clusters 1 through 5 represent the friction coefficients μ_m, equal to 0.1, 0.3, 0.5, 0.7 and 1.0. The corresponding rising/falling torque slopes in Nm/s are: (500,3000), (1000,2000), (1500,1500), (2000,1000), (3000,500).

Some effects of rising/falling slopes on robot dynamics are shown in Figs. 5-7.

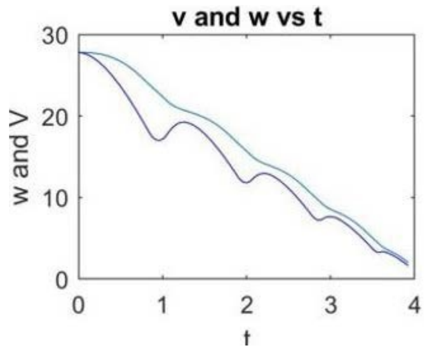

Fig. 7 Good road conditions, correct torque slope: (3000 Nm/s, 500 Nm/s)

V. Conclusion

The SOM network has been working correctly, allowing for correct torque slope choice corresponding to variable road conditions. In case of the delays in the torque modulator, adjustment of the torque slope to the existing road/wheel friction coefficient appeared to be successful. Lack of such adjustment may lead to the tire blockage and transversal instability.

References

[1] L. Acosta, J.A. Mendez, S. Torres, L. Moreno and G.N. Marichal, “On the Design and Implementation of a Neuromorphic Self-Tuning Controller”, Neural Processing Letters 9: 229–242, 1999

[2] B.J. Schreyer, “An Inquiry on 2-Mass and Wheeled Mobile Robot Dynamics”, ICMARE 2016, 18th International Conference on Mechanical, Automobile and Robotics Conference, Barcelona, Spain, August 11-12, 2016.

[3] B.J. Schreyer, “Dynamics of braking vehicle”, Internal Technical report. Military Academy of Technology. Warsaw, Poland, 1998

[4] L. Fassettt, “Fundamentals of neural networks” Prentice Hall

[5] B.J Schreyer, “2-Wheel mobile robot dynamics”, ICMARE Conference. Barcelone, Spain, August 11-12, 2016

[6] B.J Schreyer, “Efficient control of some dynamic states of wheeled robot”, MMAR Conference, Poland, Miedzyzdroje, August 28-31,2017.

[7] Yuan, Ya-xiang (1999). "Step-sizes for the gradient method" (PDF). AMS/IP Studies in Advanced Mathematics. 42 (2): 785.