ISSN 1992-6502 (Print)
Vol. 17, no. 6 (59), pp. 33-37, 2013.
Vestnik UGATU
ISSN 2225-2789 (Online)
http://journal.ugatu.ac.ru

UDC 004.7

APPLYING NEURAL NETWORKS FOR PREDICTION OF FLYING OBJECTS TRAJECTORY

K. MIRONOV[1], M. PONGRATZ[2]

[1] mironov@ict.tuwien.ac.at, [2] pongratz@ict.tuwien.ac.at

Vienna University of Technology, Austria

Submitted 2013, June 9

Abstract. The problem of flying object's trajectory prediction in material transportation by throwing is considered. Neural Network Prediction is proposed as a learning-based method for forecasting of object future position. Preliminary evaluation of the approach, using simulation software, is made. This evaluation show, that approach can satisfy requirements if equipment for throwing and measuring is precise.

Keywords: material handling; projectile; time series forecasting; neural networks.

1. INTRODUCTION

The task of an object's transportation often arises in industry (e. g. relocation of processed details from one machine tool to another). The raising demand for individual products and an increased number of product's variants over the last decades has raised the requirements for flexibility and speed of material transportation system.

The normal approach to automation of material transportation consists in application of conveyor systems (belt conveyors, screw conveyors, etc.). The lacks of this method are low speed and high power consumption.

Alternative methods for solving the object's transportation problem are connected with throwing. By the use of such transportation methods, the object is thrown from an initial point and caught in the destination point. The principle of automatized throwing and catching increases the flexibility over conveyor belts and other "traditional" ways of transportation.

An important aspect of a throwing-based transportation system is notifying the catching device about the place and the time of object interception. Object coordinates and velocity at interception time are needed to define gripper action. Trajectory prediction provides determination of these parameters, based on the information about object flight. This task belongs to well-known group of so-called time-series forecasting tasks.

2. RELATED WORK AND MOTIVATION

Transport-by-throwing as an approach for industrial transportation was proposed by Frank in [1]. Further research in this area was provided in [2, 3]. Earlier and in parallel trajectory prediction task for robotic catching was considered in [4–7]. Trajectory prediction in most of these works was based on modeling ballistic flight.

The following forces have influence on flying object [8]:

1. Gravitation is directed downwards and is constant for the flying body.

2. Pressure air drag is proportional to the velocity square for bodies, flying in the air with velocities:

$$F_d = -kv^2 . \quad (1)$$

The value of k depends on the shape of the object, its orientation with regard to motion direction, air density. For simple bodies (e. g. spheres) this value could be calculated; for more complex objects it should be defined empirically [8].

3. Viscosity, lift, Karman Vortex Street, Magnus effect, etc. is connected with air flow near the object surface. For compact objects there influence can be ignored, as its influence is low [8]. For simple bodies in special conditions they could be calculated using special formulas. Otherwise it is not possible to determine accurately influence of these effects.

4. Wind and external air flow also influence on flight.

Due to non-linearity of the flying process and the complex modeling it is logical to use learning-based methods for trajectory prediction. Such methods use a set of reference throws and their corresponding impact positions to create a prediction model. This model predicts the interception point based on the parameters of actual flight.

3. NEURAL NETWORK TRAJECTORY PREDICTION

The advantage of neural networks prediction is that neural networks are able to learn from examples only and to reveal strongly non-linear dependencies in noised training set. For time-series forecasting various time-delays neural networks (TDNN) are used. TDNN was proposed in [9] Basic architecture of such network for parameter forecasting is shown on Fig. 1.

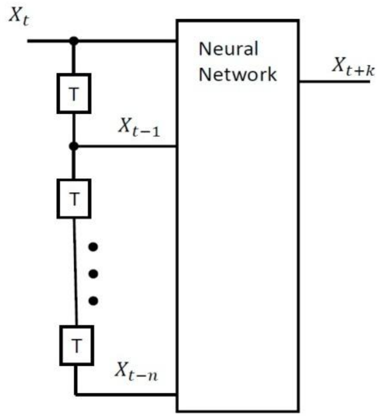


Fig. 1. Time-delay neural network prediction

To the network input a reference of objects coordinates is given. The network should predict the value of these coordinate after some fixed period. During the development of neural network trajectory predictor (NNTP), following parameters have to be defined:

1. *Network Architecture* There are two main groups of prediction networks: feedforward and recurrent. Recurrent networks have many varieties, depending on location of feedbacks in the net. Some of them (e. g. NARX) in special situations show better results than feedforward networks, but for accurate prediction they need known values of output parameters at initializing prediction. As these parameters are unknown, in current simulation feedforward networks are used.

2. *Number and values of time delays* These parameters are determined by camera frame rate.

3. *Number of neurons and layers* The optimal number of neurons depends on process and cannot be calculated. Increasing number of layers cause increased processing time, and small accuracy grow. As the application is time-sensitive, it seems better to use low number of layers.

4. *Input data* Two type of predictors are proposed. Predictor of the irst type consist of three separated networks (each with one dynamical input and one output), predicting coordinates in *x*, *y* and *z* dimension respectively. Predictor of the second type is one network with three dynamical inputs and three outputs predicting values of x, y and z.

4. SIMULATION

Evaluation of the Neural Network Trajectory Predictor (NNTP) is made using special simulation software. This software is modelling throws of the object, its flight under the influence of gravitation and air drag, and measurement of its position.

4.1. Process modeling

During the simulation of one throw three trajectories are calculated:

1. “True” trajectory is calculated on the base of initial throwing parameters, using motion model;
2. “Measured” trajectory is calculated based on “true” one, using model, that adds measurement errors;
3. “Predicted” trajectory is calculated by NNTP based on “measured” trajectory.

Measured trajectories are used to train neural network. When the accuracy of NNTP is evaluated, predicted values are being compared with the “true” trajectory. This is the advantage of the simulation in comparison with experiments on real data. In such experiments we the information about the true motion of the body is missing.

Throwing model

Potentially used throwing devices can throw object with initial velocity up to 10 m/s [3]. The distance of flight in this situation is several meters. This is the maximal value; in the majority of experiments smaller velocities are used.

In the simulation we assumed that throwing device is adjusted to throw the object with the velocity 5 m/s, towards the destination point, at an angle $\pi/4$ rad to the horizon. The differences between various trajectories are caused by the deviations in velocity and direction of throwing.

Cartesian and spherical coordinate systems are used for the calculations. The thrower and the gripper are situated in the coordinate system in the following way: throwing device is located in the origin of coordinates, *X*-axis is directed upwards, *Z*-axis is directed towards the gripper. It is assumed that the throwing parameters (velocity, azimuth, and zenith) are normally distributed near the nominal values (5 m/s, 0 and $\pi/4$ rad respectively). Mean square deviation of these parameters characterise the precision of thrower. To evaluate the influence of throw precision on prediction accuracy three virtual throwing devices were simulated:

- Low precision throw (LPT) $\sigma_v = 0.1$ m/s, $\sigma_\alpha = \sigma_\varphi = 0.1$ rad;

- High precision angle (HPA) $\sigma_v = 0.1$ m/s, $\sigma_\alpha = \sigma_\varphi = 0.01$ rad;
- High precision throw (HPT) $\sigma_v = 0.01$ m/s, $\sigma_\alpha = \sigma_\varphi = 0.01$ rad.

Motion model

Simulated object is a small sphere with the properties of tennis ball. In such situation the influence of viscosity and Karman Vortex Street is low and can be ignored. Drag doesn't depend on direction as body is spherical. The true trajectory is calculated iteratively, as: the position of the object in each moment is calculated on the base of its previous position, using formulas:

$$\vec{a}_d(t) = -\frac{k}{m}\vec{v}(t)v(t), \quad (2)$$

$$\vec{v}(t) = \vec{v}(t-dt) + (\vec{a}_d(t-dt) + \vec{g})dt, \quad (3)$$

$$\vec{x}(t) = \vec{x}(t-dt) + \frac{v(t)+v(t-dt)}{2}dt. \quad (4)$$

The step of coordinate recalculation is set to 1 ms, the time of modelling – 500 ms. An example of true trajectory set is given the Fig. 3. It consist of 10 trajectories, generated randomly, using LPT model. Fig. 3, *a* shows horizontal projections of generated trajectories, 3, *b* shows trajectory projections on *XZ*-plane.

Measurement and prediction model

It is assumed, that object coordinates are measured with normally distributed error. Within the experiment mean square error in each coordinate is set to 1 cm. Measurement are made with fixed frequency, which is lower, than frequency of true trajectory calculation. This frequency is set to 100 Hz (or one measurement for 10 coordinate's recalculations). This is close to the frame rate of the potentially used camera system (about 80 frames per second).

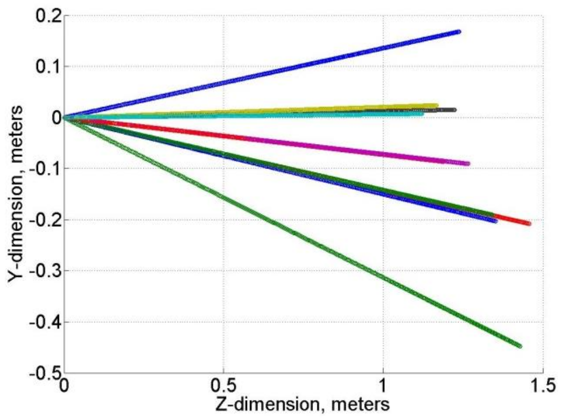
According to the task (to predict final part of trajectory, based on initial) the following prediction model is used.

$$X[n+30] = f(X[n-1], X[n-1], X[n-2], \ldots, X[n-10]) \quad (6)$$

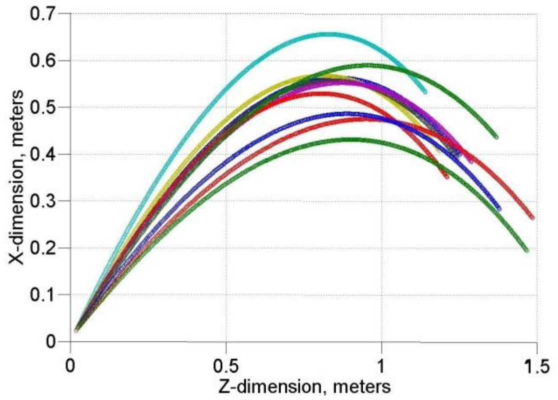
Target value show predicted coordinates in 0,3 s after current moment, based on ten previous measured values.

As each measured trajectory consists of 50 time steps, we would have 10 operations of predictor:

$$\begin{aligned} X[41] &= f(X[10], X[9], \ldots, X[1]), \\ X[42] &= f(X[11], X[10], \ldots, X[2]), \\ X[43] &= f(X[12], X[11], \ldots, X[3]), \\ &\ldots \\ X[50] &= f(X[19], X[18], \ldots, X[10]). \quad (7) \end{aligned}$$



a



b

Fig. 2. Example trajectory set for training

4.2. Simulation results

To evaluate accuracy of the prediction model 15 training sets were used (5 HPT-sets, 5 LPA-sets and 5 LPT-sets). Each set consist of 10 trajectories. Preliminary experiments shows that following numbers of neurons in hidden layer provides the best results: 6 neurons for predicting the value in x-dimension, 2 neurons for predicting the value in y-direction, 6 neurons for predicting the value in z-direction, 4 neurons for complex prediction. Feedforward networks (perceptron with one hidden layer) were used. Simulation software and prediction model was made using MATLAB. Evaluation results (mean square distance between predicted and true object coordinate) are given in the Table 1.

These results show some aspects of NNTP:

- When HPT is used, Neural Network gives results, which can be used in work of real catching system.
- More precise throwing cause more precise catching.

- Three networks, predicting coordinate in each dimension separately, usually show better accuracy than complex network.

Table 1

Mean square distance (in meters), using complex and spatial separated prediction

Test sampling	Learning sampling		
Complex prediction			
	HPT	HPA	LPT
HPT	0,024175	0,042987	0,051405
HPA	0,037465	0,043988	0,05603
LPT	0,176772	0,197149	0,089324
Spatial-separated prediction			
	HPT	HPA	LPT
HPT	0,025656	0,033006	0,056291
HPA	0,035954	0,034236	0,054745
LPT	0,197332	0,185459	0,073938

In the Table 2 accuracy values for 5 LPT sets are given.

Table 2

Mean square distance for various training sets

	σd-1, m	*σd-2*, m
LPT-1	0,059401	0,080272
LPT-2	0,058152	0,074897
LPT-3	0,085465	0,073876
LPT-4	0,081961	0,080003
LPT-5	0,084709	0,060641

Parameter *σd-1* show mean error, when given set was used for training, *σd-2* when it was used for tests. It can be seen that high values of *σd-1* match with low values of *σd-2* and vice versa. It's connected with the dispersion of trajectories in the set. High dispersion of trajectories provides better training. Hence, trajectories that provide high dispersion should be used for creating training set.

Second addition is connected with target data. Use true target values instead of measured values increased the accuracy of prediction e.g. for HPT, using true values, mean error would be 2,4 cm instead of 2,6 cm, using measured values. It means that more precise measurements of object position in catching area during the training set generation could increase accuracy.

Time of processing 100 frames by prediction algorithm is about 7 ms for common neural network and about 1.2 ms for spatial-separated model. This time is much smaller, than interframe period of measuring camera, so it these values of speed satisfy task requirements. These are values for implementation in MATLAB Neural Networks Toolbox, another implementation, especially parallelized and hardware neural networks could work much faster.

5. CONCLUSION AND FUTURE WORK

Conclusion

Two proposed NNTP architectures were evaluated using the simulation software. With precise-throwing model they show satisfying accuracy. Spatial-separated NNTP show better accuracy, than common. The speed of prediction meets requirements for the catching system.

Future work

Important step in future development of NNTP is using real-flight data instead of simulation. Using real flights prediction of complex-shaped objects trajectory could be evaluated. For complex-shaped objects not only coordinate are to be measured and predicted, but also its orientation. Another part of future work is defining of required measurement frequency. Probably it would be enough for prediction to use 3 measurements in 100 ms instead of 10.

ACKNOWLEDGMENTS

The research work of K. V. Mironov is supported by the European Erasmus-Mundus Program.

REFERENCES

1. H. Frank, N. Wellerdick-Wojtasik, B. Hagebeuker, G. Novak, and S. Mahlknecht, "Throwing Objects – A bio-inspired Approach for the Transportation of Parts," in *IEEE Int. Conf. Robotics and Biomimetics, ROBIO 2006, December 17-20*, 2006, pp. 91–96.
2. H. Frank, D. Barteit, and F. Kupzog, "Accurate prediction of interception positions for catching thrown objects in production systems," in *IEEE Int. Conf. Industrial Informatics, INDIN 2008, July 13-16*, Dejeon, Korea, 2008, pp. 893–898.
3. M. Pongratz, F. Kupzog, H. Frank, and D. Barteit, "Transport by Throwing - a bio-inspired Approach," in *IEEE Int. Conf. Industrial Informatics, INDIN 2010, July 13-16*, Dejeon, Korea, 2008, pp. 685-689.
4. A. Namiki, Y. Nakabo, I. Ishii, and M. Ishikawa, "High speed grasping using visual and force feedback," in *IEEE Int. Conf. Robotics and Automation, ICRA 1999*, vol. 4, pp. 3195-3200.
5. A. Namiki and M. Ishikawa, "Robotic catching using a direct mapping from visual information to motor command," in *IEEE Int. Conf. Robotics and Automation, ICRA 2003, Sept. 14-19*, vol. 2, pp. 2400-2405.
6. W. Hong and J.-J. E. Slotine "Experiments in hand-eye coordination using active vision," in *Experimental Robotics IV. LNCIS,* Springer, Heidelberg, 1995, vol. 223, pp. 130-139.

7. U. Frese, B. Bauml, S. Haidacher, G. Schreiber, I. Schaefer, M. Hahnle, and G. Hirzinger, "O-the-Shelf Vision for a Robotic Ball Catcher," in *IEEE Int. Conf. Intelligent Robots and Systems*, USA, 2001, pp. 1623-1629.

8. T. von Kármán, *Aerodynamics: Selected Topics in the Light of Their Historical Development*. Ithaca, New York: Cornell Univesity Press, 1954, pp. 98-108.

9. T. J. Sejnowsky and C. R. Rosenberg, "Parallel networks that learn to pronounce English text," *Complex Systems,* vol. 1, pp. 145-168, 1987.

ABOUT AUTHORS

MIRONOV, Konstantin, Doctoral Stud. (Institute of Computer Technology). Dipl.-Eng., BSc (Ufa State Aviation Tech. Univ., 2012). Researches: material transportation by throwing and catching.

PONGRATZ, Martin, Univ. Asst. (Institute of Computer Technology). Dipl.-Eng. (Higher Tech. School, Pinkafeld, 2002), MSc (Vienna Univ. of Technology, 2009). Researches: transport-by-throwing, robotic catching.