Computer Vision System to Track Moving Objects With unknown Periodic Moving Patterns – Based on DREM Algorithm

Ali Shakkouf
ITMO University
Saint Petersburg, Russia
Ali.sha8ouf.itmo@gmail.com

Vladislav Gromov
ITMO University
Saint Petersburg, Russia
gromov@itmo.ru

Abstract—Objects tracking system is a research field of great importance. The two main reasons for this: 1 - Attempting to replace humans with robots or automatic systems in every field wherever possible. 2 - The accuracy, effectiveness and speed of those systems compared to human performance. In this research we propose a new approach for tracking objects and predicting the future trajectory of these objects. the main idea is to observe the object for about 15 seconds, this gives us part of the moving pattern, then "Dynamic Regressor Extension and Mixing (DREM)" algorithm is used to estimate the main frequencies involved in the detected movement pattern. The estimated frequencies are used to predict the future trajectory of the object. The accuracy of this method was tested using 5DOF arm robot, which try to grip the object at each moment of its future trajectory. This research performed on a virtual object designed in "vredit" in MATLAB. The object moves on an LCD screen. The results were represented as the difference between the predicted trajectory and the real future trajectory. Results show that tracking error of a ball moves on the screen for different moving pattern is less than 1cm, where the screen stands 500mm far from arm base.

I. INTRODUCTION

Tracking moving objects is an extensively wide area for scientific research nowadays. Tracking is used in video surveillance, human-computer interaction, Assembly lines, and robot navigation ...etc. Tracking also could be a part of some researches, for example tracking cars on long period of time, then analyzing their trajectories to find the main reasons for cars accidents, then try to eliminate that reason in an efficient way. On one hand, there are so many developed and fast performing algorithms for tracking moving objects like [3], [11], [12]. On the other hand, there are not that much researches about predicting the future trajectory of moving objects like [4], [13], [14].

Tracking systems vary in complexity and principle they use. [2] describes a vision system to detect moving objects on a conveyor belt. stationary camera is used here. detection task is solved by subtraction of two consecutive frames. the speed is measured by dividing the passed distance on the time needed to pass it. [1] developed a detection/tracking system to be used on unmanned aircrafts. His mathematical model assumes that any acquired frame is a convolution between object image and background image. The tracking task is then performed based on trajectory graph without any initial information about the tracked objects. Both [1] and [2] don't deal with future trajectory of the tracked objects.

[5] represents a novel location prediction model based on grey theory. The negative side of such an approach is it could predict the future position for maximum five seconds after the last frame was taken. [6] carries out a study on predicting the future trajectory of people based of their past trajectory. LSTM-based model was introduced (Long Short Term Memory). This model can meet across multiple individuals to predict human paths in a scene. it also takes in consideration that people may change their motion based on the behavior of other people around them. Person may stop for a sec, or wait to accommodate a group of people towards him. [7] carries out a study on predicting future trajectory of cars for some seconds in advance. The goal of this research is to help driver assistance systems in avoiding accidents as much as possible. Chebyshev decomposition is used and the coefficients of the obtained polynomial are used as input to the probabilistic model which is used to predict future trajectory.

[8] represents a useful use of the so-called dynamic regressor extension and mixing (DREM) to track a multi-sinusoidal signal. He considers LTI system with delayed input of the form:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t-h) \\ y(t) &= Cx(t) \\ e(t) &= g(t) - y(t) \end{aligned}$$

The problem statement is to find a control law such that the error $e(t)$ goes to zero. Given that the reference signal $g(t)$ is a combination of biased sinusoidal signals with different frequencies.

Most vision systems in tracking tasks uses camera calibration techniques to extract the real 3D coordinate of the objects with respect to camera coordinate system. The most popular closed form model for camera calibration was developed by [10]. It only requires the camera to observe a calibration pattern in different orientations (at least two). This model was improved by [9] which proposed to perform calibration using an LCD screen and a software that modifies calibration pattern size to optimize the estimation of camera parameters.

The paper is organized as follows. Part 2 describes the studied problem and the new approach for tracking. Part 3 clarifies the main part. It is divided into multiple sections; A: system hardware. B: how the implemented system functions. C: mathematical model of DREM, which is used as frequency estimator. D: the conversion between screen and youBot coordinate systems. E: forward/inverse kinematic of youBot. F: real tracking experiments to test the new approach. Finally, a conclusion and ideas for future work.

II. Problem Statement

Given an object that moves with a periodic pattern of the form:

$$g(t) = \sum_{i=1}^{k} a_i . \sin(\omega_i . t) \quad (1)$$

can we predict the future trajectory of that object if it was available for observation for about 15 seconds with vision sensor?

The main idea of the proposed solution/approach is based on the theorem formulated by the French mathematician Joseph Fourier, "*any periodic function, no matter how trivial or complex, can be expressed in terms of converging series of combinations of sines and/or cosines"*, known as Fourier series.

Taking advantage of Fourier theorem, problem statement could be formulated as: Can we estimate the frequencies involved in the detected movement pattern and use that estimation to predict the future trajectory?

III. Main part

1) *Description of the implemented system*

Since it is quite difficult to have a real object in the laboratory that moves with moving patterns combined of multiple periodic signals like the ones shown in Fig. 1, we decided to create that object in Simulink. Virtual Reality Editor in MATLAB was used to create the object. From now on the tracked object is called the Target. Target is a white ball moves on an LCD screen. The vision sensor is a web camera "Logitech c170".

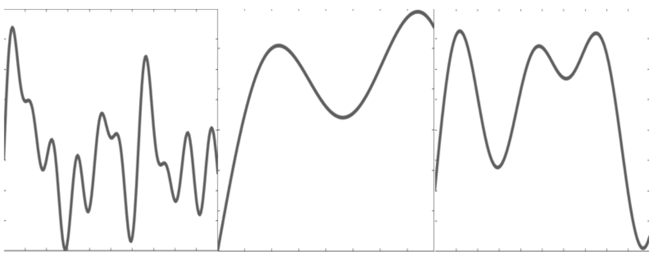

Fig. 1. examples of periodic trajectory patterns

The accuracy and efficiency of the implemented system is being tested by forcing the end effector of a manipulator to be in as small as possible distance from the target as if it is going to grip the target at each moment of the trajectory. The used manipulator is "KUKA youBot manipulator". The camera is attached to the last link of the manipulator (see Fig. 2). Distance between youBot and screen is unknown. It is estimated using camera calibration technique.

Fig. 2. System hardware

2) *System flowchart/How does the system function?*

First, we place the manipulator and the LCD screen on a flat surface (table) so they face each other. Remember that the distance between them **should be in a range, for which the manipulator can access each point of the screen**. No need to measure that distance. Also, no need to know the orientation of the screen.

Fig. 4 represents system flowchart, which could be described by the following:

- Choose a chessboard pattern with even number of squares in column and odd number of squares in row. Display that pattern on the LCD screen. Then send commands to the manipulator, so that it takes a specific configuration for each of which the camera sees the whole screen.
- Open MATLAB calibrator. Acquire frames for the calibration pattern (at least two, to optimize and obtain good results take 15 frames).
- Open the Target model in Simulink. Choose the movement function of it and run the simulation. Then a list of functions is called, each one to perform a specific task:
- *"observe()"* function. This function detects the moving pattern of the Target. It takes frame each 100 mSecond. For each frame the Target is detected. When Target is detected, his coordinates and frame time are stored as a pair <time, coordinates>. If in some frame, the object is not detectable, his coordinates are stored as NaN.
- *"math_model()"* function. This function returns the homogeneous transformation between the youBot base and screen coordinate system (see Fig. 3).

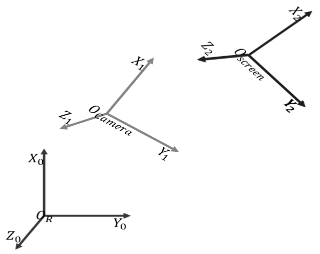

Fig. 3. System coordinate frames

- *"enhance()"* function. This function prepares the detected pattern to be a good input for DREM. It involves three operations:
  - Modifying the signal amplitude.
  - Performing interpolation to remove NANs.
  - Performing another interpolation to add extra mid-points to the pattern.
- *"DREM()"* function. This function takes, as an input, the enhanced pattern, and return the estimated frequencies existed in that pattern.
- *"tracking()"* function. This function calculates (in real time) the Target coordinates on the screen with respect to youBot base coordinate system, then start following it.

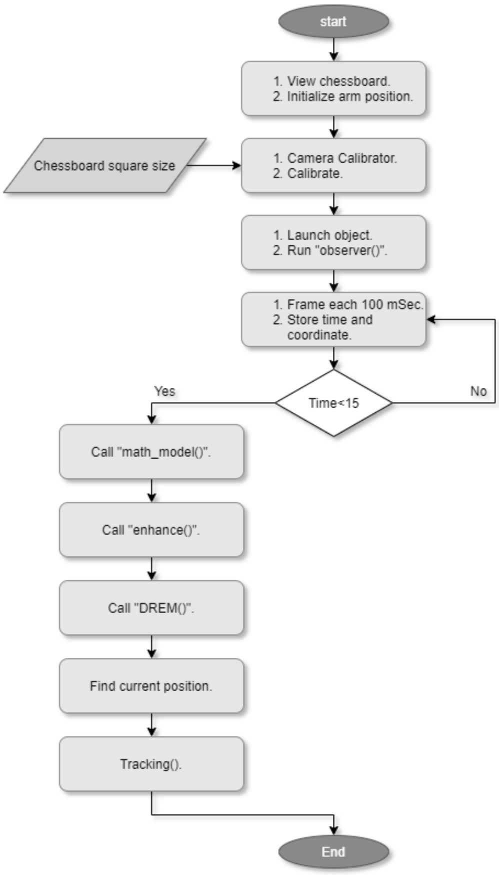

Fig. 4. system flowchart

3) *DREM algorithm*

We refer to the pattern detected by the camera as $\hat{g}(t)$ which is the input signal to DREM. Filter the signal with Hurwitz filter of the form:

$$\xi(t)=\frac{\lambda^{2l}}{(p+\lambda)^{2l}}\hat{g}(t) \tag{2}$$

Where $(p+\lambda)^{2l}$ is a Hurwitz polynomial, $p=d/dt$ is the differentiation operator, l is the number of frequencies existed in the input signal. Next, construct the following linear regression model:

$$\xi^{(2l)}(t)=\upsilon^T(t)\vartheta+\varepsilon(t) \tag{3}$$

Where $\varepsilon(t)$ is an exponentially decaying term. The regressor is composed of consecutive derivatives of the filter:

$$\upsilon^T(t)=\left[\xi^{(2l-2)}(t) \quad \cdots \quad \xi^{(2)}(t) \quad \xi(t)\right] \tag{4}$$

And the estimated vector is written as:

$$\vartheta^T=\left[\bar{\theta}_1 \quad \cdots \quad \bar{\theta}_{l-1} \quad \bar{\theta}_l\right] \tag{5}$$

Elements of ϑ are related with the following system of equations:

$$\begin{cases}\bar{\theta}_1 = & \theta_1+\theta_2+\cdots+\theta_l\\ \bar{\theta}_2 = & -\theta_1\theta_2-\theta_2\theta_3-\cdots-\theta_{l-1}\theta_l\\ \vdots & \\ \bar{\theta}_l = & (-1)^{l+1}\theta_1\theta_2\cdots\theta_l\end{cases} \tag{6}$$

Where $\theta_i = -\omega_i^2$ and ω_i is the i_{th} frequency in the input signal. At this point we can estimate frequencies using the well-known "Gradient-based frequency estimator" by the following law:

$$\dot{\hat{\vartheta}}(t)=K_\vartheta.\upsilon(t)\left(\xi^{(2l)}(t)-\upsilon^T(t)\hat{\vartheta}(t)\right) \tag{7}$$

Where $K_\vartheta \in \mathbb{R}^{1\times l}, K_\vartheta>0$. However, Gradient-based estimator is slower than DREM (need more time for estimation).

Choose $l-1$ distinct delays $\left(d_i, i\in\{1,2,...,l-1\}\right)$, then set $l-1$ filtered signals as follows:

$$\upsilon_{f_i}(t)=\upsilon(t-d_i) \tag{8}$$

$$\xi_{f_i}(t)=\xi(t-d_i) \tag{9}$$

Using the previous new elements generate two new main vectors as follows:

$$\Upsilon_e(t) := \begin{bmatrix} \upsilon^T(t) \\ \upsilon^T_{f_i}(t) \\ \vdots \\ \upsilon^T_{f_{l-1}}(t) \end{bmatrix} \tag{10}$$

$$\Xi_e(t) := \begin{bmatrix} \xi(t) \\ \xi_{f_i}(t) \\ \vdots \\ \xi_{f_i}(t) \end{bmatrix} \tag{11}$$

Where $\Upsilon_e(t) \in \mathbb{R}^{l\times l}, \Xi_e(t) \in \mathbb{R}^{l\times 1}$. Define two new variables as follows:

$$\psi_\phi(t) := \det\{\Upsilon_e(t)\} \tag{12}$$

$$\Xi(t) := adj\{\Upsilon_e(t)\}.\Xi_e(t) \tag{13}$$

The previous equations give us a set of l equations:

$$\Xi_i(t) = \psi_\phi(t).\vartheta_i \qquad :i \in \{1,2,3,...,l\} \tag{14}$$

Thus, estimation law is written as follows:

$$\dot{\hat{\vartheta}}_i(t) = \gamma_i.\psi_\phi(t)\left(\Xi_i(t) - \psi_\phi(t)\hat{\vartheta}(t)\right) \tag{15}$$

Where $\gamma_i > 0$.

4) Conversion between screen and youBot coordinate systems

As shown in Fig. 3 we have three coordinate systems; youBot, camera and screen coordinate systems. From now on we will replace "Coordinate System" with CS, and homogeneous transformation with HT. To find the representation of some point on the screen in youBot CS, we need to implement a mathematical model that connects those three said CSs. Actually, also we need to estimate HT between youBot end-effector and camera. To build this mathematical model we do the following:

- Find HT between youBot CS and end-effector CS using the forward kinematic model (described in details in subsection E).
- Estimate HT between camera and youBot end effector. The estimation is performed with the help of method and software introduced by "Visual Servoing Platform" (see [15] for details).
- Estimate HT between camera and screen CSs. Camera calibrator in MATLAB offer a helpful not that accurate estimation of that HT (sometimes error in translation vector estimation go up to 7 cm). Each estimation of HT corresponds to a different configuration of youBot while taking frames for calibration. An example of one experiment estimation is shown in Fig. 5.

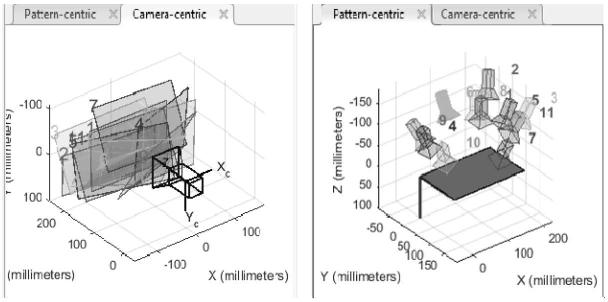

Fig. 5. camera and screen locations representation with respect to each other's

The estimation offered by camera calibrator could be improved and made more accurate. We have for each youBot configuration:

$$i_{\,th\_configuration} \Leftrightarrow \begin{Bmatrix} R_{camera\rightarrow screen} \\ T_{camera\rightarrow screen} \end{Bmatrix} \Rightarrow \begin{cases} R_{youBot\ base\rightarrow screen} \\ T_{youBot\ base\rightarrow screen} \end{cases}$$

But since screen and youBot base are fixed with respect to each other, the HT between their CS are constant. We mentioned earlier that it is preferred to take 15 frames for calibration. This gives us 15 different estimation of $R_{youBot\ base\rightarrow screen}$ and $T_{youBot\ base\rightarrow screen}$. So a simple improvement could be done by taking the average of those 15 estimations.

5) youBot manipulator forward and kinematic tasks

a. forward kinematic:

To solve the forward kinematic task, we need to assign CS to each joint of youBot as shown in Fig. 6.

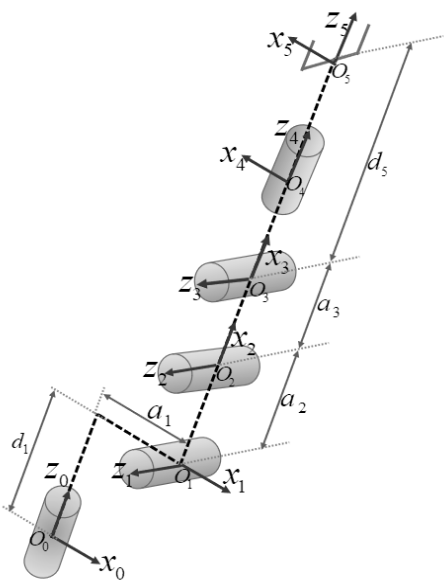

Fig. 6. joints coordinate frames in youBot

Where $a_1 = 0.033, a_2 = 0.155, a_3 = 0.135, d_1 = 0.147$, $d_5 = 0.218$ all in meter.

Following Denavit–Hartenberg rules, we fill the following TABLE I.

TABLE I. DENAVIT–HARTENBERG TABLE FOR YOUBOT

	a_i	α_i	d_i	θ_i
1	a_1	$-\pi/2$	d_1	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3
4	0	$\pi/2$	0	θ_4
5	0	0	d_5	θ_5

Substitute value from TABLE I in equation 16 to find the HT between each consecutive CSs, then by multiplying them we get the HT of forward kinematic.

$$ {}_{i-1}T^{i} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16) $$

b. Inverse kinematic:

Here we introduce a new approach to solve the IK task for our system. It is called the IK trapezoidal method. **With this approach we avoid every possible singularity.** Also we don't need to know the orientation of the end-effector. The main idea of this approach is: "slope of arm's fourth link at any time is parallel to the line that connects O_5 and O_1" in Fig. 7.

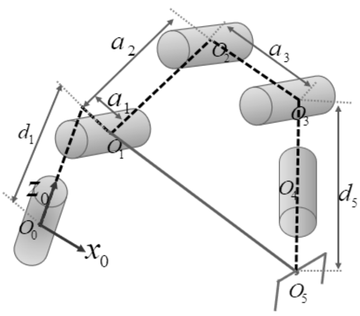

Fig. 7. a youBot configuration in trapezoidal method

2D representation of Fig. 7 is shown in Fig. 8.

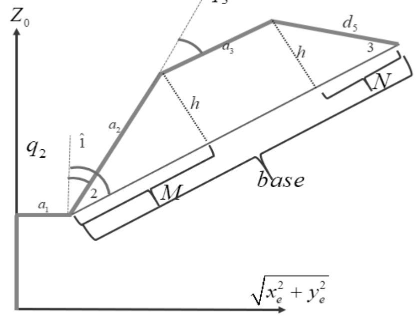

Fig. 8. 2D representation of Fig. 7

Which leads to a solution represented in 17.

$$ \left.\begin{aligned} & q_1 = a\tan\left(y_{obj}/x_{obj}\right) \\ & l_1 = \sqrt{\left(x_{obj}-a_1\cos(q_1)\right)^2+\left(y_{obj}-a_1\sin(q_1)\right)^2} \\ & base = \sqrt{l_1^2+\left(z_{obj}-d_1\right)^2} \\ & ang_1 = a\cos\left(\left(z_{obj}-d_1\right)/base\right) \\ & a = base; b = a_2; c = a_3; d = d_5 \\ & temp_1 = (a+b-c+d).(-a+b+c+d) \\ & temp_2 = (a-b-c+d).(a+b-c-d) \\ & h = \sqrt{temp_1.temp_2/\left(4.(a-c)^2\right)} \\ & m = \sqrt{b^2-h^2} \\ & ang_2 = a\cos\left(\left(b^2+m^2-h^2\right)/(2mb)\right) \\ & q_2 = ang_1 - ang_2 \\ & q_3 = ang_2, n = a-m-c \\ & ang_3 = a\cos\left(\left(d^2+n^2-h^2\right)/(2nd)\right) \\ & q_4 = ang_3, q_5 = 0 \\ & q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 \end{bmatrix} \end{aligned}\right\} \quad (17) $$

Where $x_{obj}, y_{obj}, z_{obj}$ are the 3D coordinate of some point on the screen, x_e, y_e are x and y of the end-effector origin.

6) Real experiments on the proposed approach

In this section we represent the results of two experiments were carried out in the laboratory. Each experiment relates to different movement pattern as following:

$g_1(t) = 2.66\left(\sin(0.19t)+\sin(1.69t)\right)$

$g_2(t) = 0.66\left(\sin(0.19t)+\sin(0.09t)+\sin(0.39t)+0.2\sin(1.69t)\right)$

For each experiment we present four figures (see Fig. 9 and Fig. 10):

- (a) – detected pattern: it is the trajectory through which the object moved. This pattern was detected by the camera.
- (b) – DREM estimation: it is estimation of the frequencies involved in the detected pattern.
- (c) – Real vs Predicted pattern: here two trajectories are represented; the continuous line is the real trajectory generated by the object. The dotted line is the estimated trajectory.
- (d) – Trajectory estimation error: it is the difference between the real and estimated (predicted) trajectories. We are more interested in the part right to the vertical line, cause at that part we needed trajectory estimation.

We chose to represent the error in pixels to make the approach more useful and realistic. For example, for our system of screen and a manipulator the error is less than one centimeter. Cause the real size of one pixel is too small. You can easily apply the same two experiments in different conditions. For instance, suppose an object moves on the ground and is being observed by a camera at 25 meters high from the ground, then by measuring what is the real counterpart of one pixel on the ground we figure out the error of trajectory estimation.

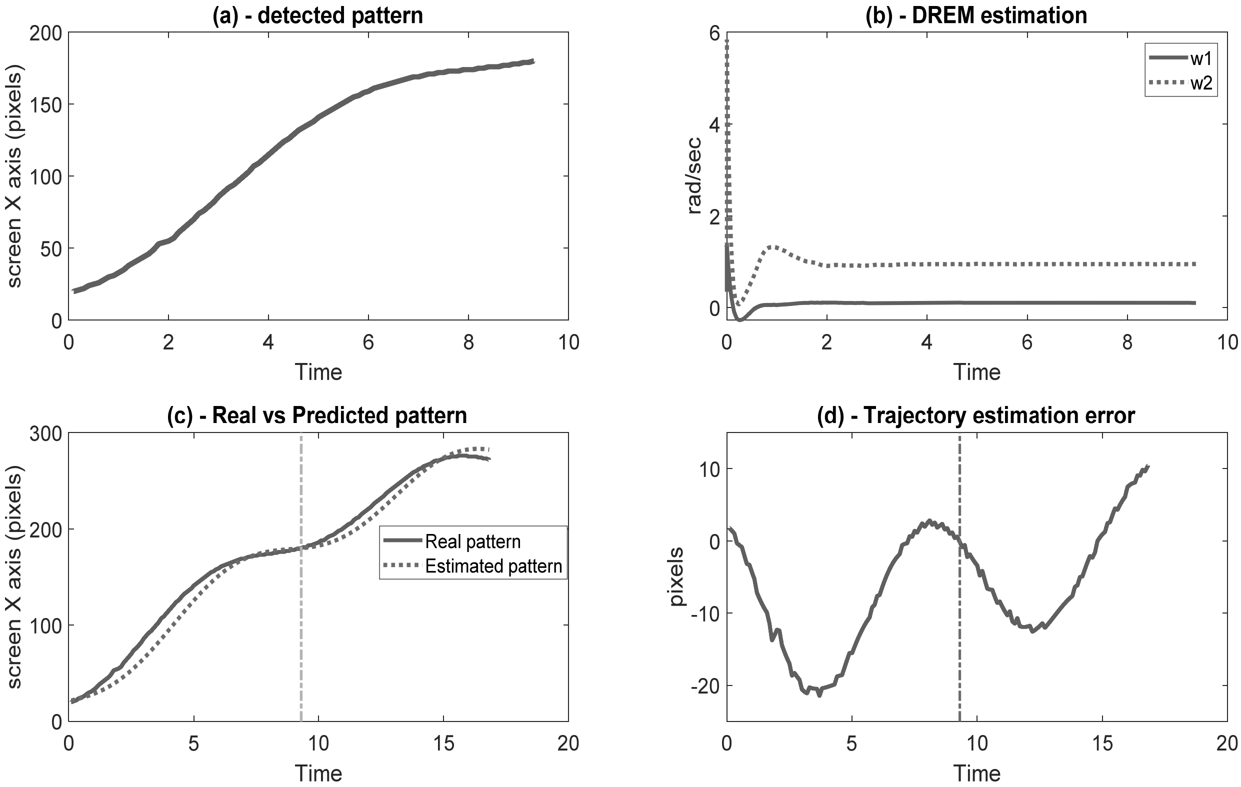

Fig. 9. Results of Experiment 1

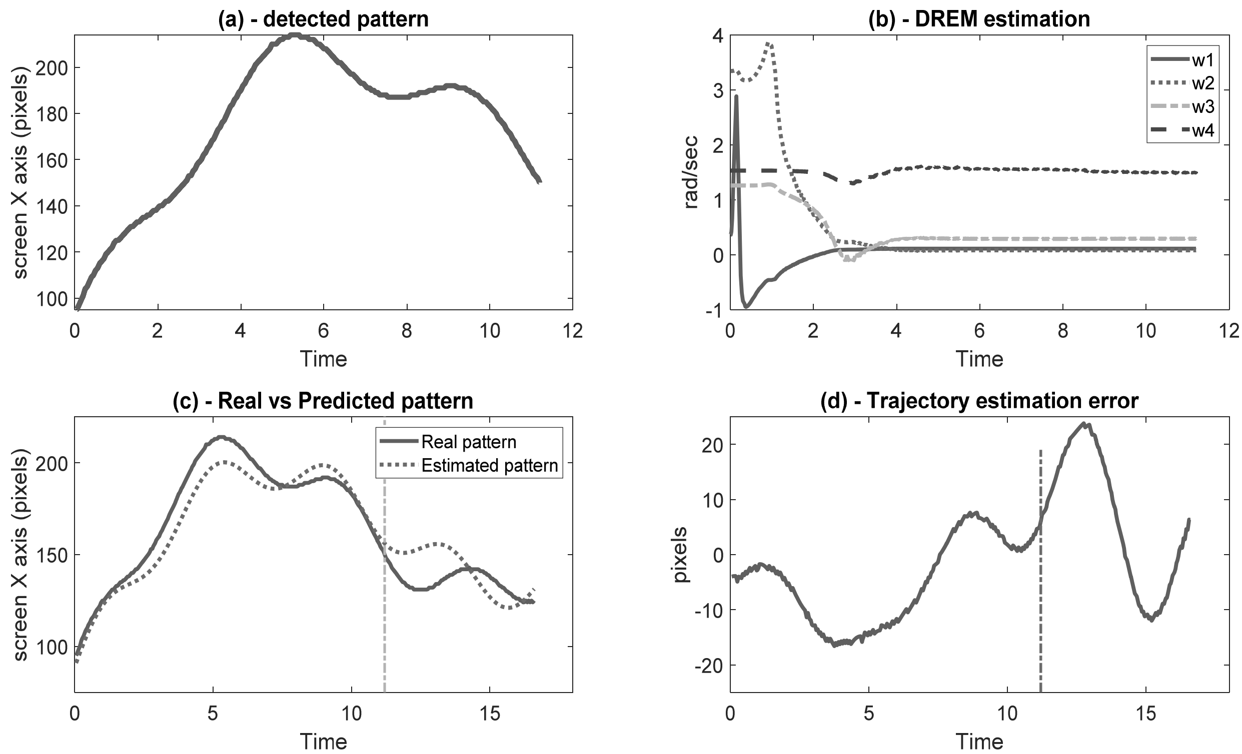

Fig. 10. Results of experiment 2

IV. CONCLUSION

In this research, a new approach to track objects and predict future trajectories has been presented and discussed. The efficiency of this method was tested in the laboratory on a target that moves with periodic moving patterns. The results and tracking errors show that the tracking error is relatively small. This research could be extended by future work (especially in the field of discrete signal processing) and used for tracking objects that move with nonperiodic moving patterns.

ACKNOWLEDGMENT

This work was financially supported by Government of Russian Federation (Grant 08-08).

REFERENCES

[1] Erokhin, Denis & Feldman, Alexander & Korepanov, S, "detection and tracking of moving objects with real-time onboard vision system", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2017, XLII-2/W4. 67-71.

[2] Sang, Ik & Nam, Sang-Hyun & Yu, Hyun & Roberts, Rodney & Moon, Seungbin, "Conveyor visual tracking using robot vision", *FCRAR*. 2006.

[3] Tsarouchi, Panagiota & Michalos, George & Makris, S. & Chryssolouris, George, "Vision System for Robotic Handling of Randomly Placed Objects", *Procedia CIRP*, 9 (2013). 61-66. 10.1016/j.procir.2013.06.169.

[4] F. Husain, A. Colomé, B. Dellen, G. Alenyà and C. Torras, "Realtime tracking and grasping of a moving object from range video", *IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 2617-2622.

[5] Y. Shen, "Location Prediction for Tracking Moving Objects", *WRI Global Congress on Intelligent Systems, Xiamen.* 2009, pp. 362-366.

[6] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. NV, 2016, pp. 961-971.

[7] A. Wiest, M. Höffken, U. Kreßel and K. Dietmayer, "Probabilistic trajectory prediction with Gaussian mixture models", *IEEE Intelligent Vehicles Symposium, Alcala de Henares*, 2012, pp. 141-146.

[8] Borisov, Oleg & Gromov, Vladislav & Vedyakov, Alexey & Pyrkin, Anton & Bobtsov, Alexey & Aranovskiy, Stanislav, "Adaptive Tracking of a Multi-Sinusoidal Signal with DREM-Based Parameters Estimation", *IFAC-PapersOnLine*. 2017, 50. 4282-4287.

[9] Doganis, Fivos (Yerres, FR). Dynamical camera calibration. 20200005491A1, Dassault Systemes (Velizy Villacoublay, FR), January 2020.

[10] Z. Zhang, "A flexible new technique for camera calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. pp. 1330-1334, Nov. 2000.

[11] Tomashevich, S.I., Borisov, O.I., Gromov, V.S., Pyrkin, A.A., Bobtsov, A.A. "Experimental study on robust output control for quadcopters", (2017) 2017 25th Mediterranean Conference on Control and Automation, MED 2017, pp. 1029-1034.

[12] Borisov, O.I., Gromov, V.S., Pyrkin, A.A., Bobtsov, A.A., Petranevsky, I.V., Klyunin, A.O. "Output robust control with anti-windup compensation for robotic boat". (2016) 2016 21st International Conference on Methods and Models in Automation and Robotics, MMAR 2016, art. no. 7575080, pp. 13-18.

[13] Borisov, O.I., Gromov, V.S., Kolyubin, S.A., Pyrkin, A.A., Bobtsov, A.A., Salikhov, V.I., Klyunin, A.O., Petranevsky, I.V. "Human-free robotic automation of industrial operations". (2016) IECON Proceedings (Industrial Electronics Conference), art. no. 7793922, pp. 6867-6872.

[14] Gromov, V.S., Borisov, O.I., Vedyakov, A.A., Pyrkin, A.A., Shavetov, S.V., Bobtsov, A.A., Salikhov, V.I., Aranovskiy, S.V. "Adaptive Multisinusoidal Signal Tracking System with Input Delay". (2016) IFAC-PapersOnLine, 49 (13), pp. 105-110.

[15] E. Marchand, F. Spindler and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills", *IEEE Robotics & Automation Magazine*. pp. 40-52, Dec. 2005.