A method for realization of nonlinear state-dependent coefficients regulators based on microcontroller memory

Semion A. A.

***Abstract* — We investigate a nonlinear stabilizing regulator with coefficients depending on the state built for nonlinear systems. Usage of a quadratic cost function allows to develop a control with coefficients that include the solution of the Riccati equation. A rather common approach is to solve this equation online which requires making high performance calculations that is not appropriate for some applications such as UAV autopilots. The method represented in this article is useful when the system state space is compact and the performance or weight of a control device is more critical than memory size. Thus, we can sample state space with some accuracy. It is offered to calculate regulator gain coefficients in advance and to keep them in memory of the control device. Assessment of the regulator gain coefficients quantity and memory size depending on accuracy of system state space sampling is provided. The algorithm of the next gain coefficient fast search is given. The numerical simulation of a UAV controlled by such a regulator is made for verification.**

***Index Terms* — quaternion algebra, state dependent coefficients, Riccati equation, nonlinear dynamic systems.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) autopilots often require fast processing of inputs from sensors and corresponding control outputs calculation. Linear regulators are not always capable to operate such objects correctly. Nonlinear differential equations of UAVs motion could be represented as nonlinear differential equations with linear structure and parameters depending on object state. Thus, State Dependent Coefficients (SDC) method could be applied to develop a regulator [1]. Usage of a quadratic cost function allows to apply the method based on the State Dependent Riccati Equation (SDRE) solutions [2].

The SDRE method usage could be complicated because of problems connected with ambiguity of equivalent transformations to SDC representation and the need of a rather fast and efficient algorithm of matrix Riccati equation online solution [3].

For correct work of a regulator it is enough to find the solution of the Riccati equation in 14 ms as it is showed in [4]. The described algorithm required a computer with CPU clock rate about 300 MHz. Solutions that could provide such performance might be not appropriative for micro UAVs.

We present an approximate method which relaxes requirements for control devices. The main disadvantage of this method is a rather small class of applicable systems. The idea is to calculate the gain coefficients of a regulator in a certain point of a possible trajectory in advance. It combines approaches of the regulator synthesis for systems with coefficients depending on states, the gain-scheduled regulators and the cache concept.

II. Problem Formulation

We will consider four rotor helicopter (quadcopter) as a controllable object. Four motors with rotors attached are located at ends of poles of length 2l. Motors are numbered clockwise starting with the front right (Fig. 1). Motor can rotate rotor only in the direction marked by an arrow.

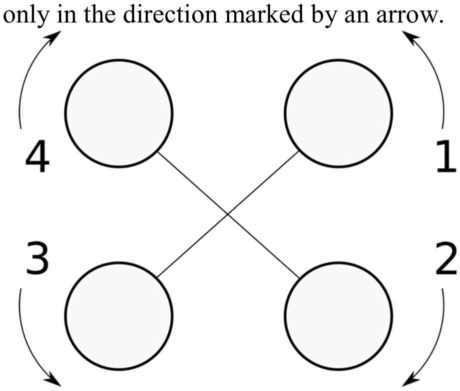

Fig. 1. Quadcopter top view. Numbers describe motor enumeration.

Rotors have the same fixed pitch. Inclination of the quadcopter is made by the differentiation of motors thrust. It is supposed that the sum of all thrust forces is able to lift a UAV. Moreover, the thrust-weight ratio is 2.

We suppose that the current state is undisturbed and is observable.

The control problem is to stabilize vehicle horizontally.

III. Equations of Motion

For further reasoning we need a mathematical model of the

This work was supported by the Russian Science Foundation (Project No. 17-11-01220).

A. A. Semion is with Institute for Systems Analysis Federal Research Center “Computer Science and Control” of Russian Academy of Sciences, Moscow, Russia. He is now a PhD student of National Research University Higher School of Economics Moscow, Russia (e-mail: alexander.semion@gmail.com).

quadcopter which is given in [5], where rotation of the UAV is described in algebra of quaternions [6].

We will use Euler's equations for rotational motion: $\tau = I\dot{\omega} + \omega \times I\omega$, here τ denotes the torque generated by the rotors, I – inertia tensor, $\omega = \begin{pmatrix} \dot{\varphi} & \dot{\theta} & \dot{\psi} \end{pmatrix}$ – angular velocity vector described in body-fixed coordinate system.

Applied torques components will be described as controls: $\tau = \begin{pmatrix} U_1 & U_2 & U_3 \end{pmatrix}^T$, where

$$U_1 = \frac{\sqrt{2}}{2} l\left(F_1 + F_2 - F_3 - F_4\right),$$

$$U_2 = \frac{\sqrt{2}}{2} l\left(-F_1 + F_2 + F_3 - F_4\right),$$

$U_3 = \alpha\left(F_1 - F_2 + F_3 - F_4\right)$ and F_i are the thrust of the corresponding rotor. The alpha coefficient is calculated empirically.

Note that for the full system description we have to add the control $U_0 = F_1 + F_2 + F_3 + F_4$, which is calculated in the height stabilisation problem and will not be provided in this work.

We also add an equation for the quaternion derivative received from the quaternion properties [6]: $\dot{\vec{\lambda}} = \frac{1}{2}\vec{\lambda} \circ \omega$.

Combining the above mentioned equations we obtain the motion equations of a quadcopter:

$$\begin{aligned}
\frac{d}{dt}\dot{\varphi} &= \frac{1}{I_x}\left[U_1 - \left(I_z - I_y\right)\dot{\theta}\dot{\psi}\right], \\
\frac{d}{dt}\dot{\theta} &= \frac{1}{I_y}\left[U_2 - \left(I_z - I_y\right)\dot{\varphi}\dot{\psi}\right], \\
\frac{d}{dt}\dot{\psi} &= \frac{1}{I_z}\left[U_3 - \left(I_z - I_y\right)\dot{\varphi}\dot{\theta}\right], \\
\frac{d}{dt}\lambda_0 &= \frac{1}{2}\left[-\lambda_1\dot{\varphi} - \lambda_2\dot{\theta} - \lambda_3\dot{\psi}\right], \quad (1) \\
\frac{d}{dt}\lambda_1 &= \frac{1}{2}\left[\lambda_0\dot{\varphi} - \lambda_3\dot{\theta} + \lambda_2\dot{\psi}\right], \\
\frac{d}{dt}\lambda_2 &= \frac{1}{2}\left[\lambda_3\dot{\varphi} + \lambda_0\dot{\theta} - \lambda_1\dot{\psi}\right], \\
\frac{d}{dt}\lambda_3 &= \frac{1}{2}\left[-\lambda_2\dot{\varphi} + \lambda_1\dot{\theta} + \lambda_0\dot{\psi}\right].
\end{aligned}$$

Here are the components of the quaternion describing the rotation of a UAV.

Note that such representation of system (1) is not controllable. However, Yaguan Yang [7] shows a method of system reduction to make this system controllable. He proves that the scalar component of the quaternion can be extracted from the system and substituted by $\lambda_0 = f\left(\vec{\lambda}\right) = \sqrt{1 - \lambda_1^2 - \lambda_2^2 - \lambda_3^2}$ with some limitations.

After such substitution the dynamic system can be represented in the standard form: $\dot{X} = A\left(X\right)X + BU$, where $X = \begin{pmatrix} \dot{\varphi} & \dot{\theta} & \dot{\psi} & \lambda_1 & \lambda_2 & \lambda_3 \end{pmatrix}^T$ – a state vector, $U = \begin{pmatrix} U_1 & U_2 & U_3 \end{pmatrix}^T$ - a control vector,

$$A = \begin{pmatrix}
0 & 0 & \frac{I_y - I_z}{I_x}\dot{\theta} & 0 & 0 & 0 \\
\frac{I_z - I_x}{I_y}\dot{\psi} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{I_x - I_y}{I_z}\dot{\varphi} & 0 & 0 & 0 & 0 \\
\frac{1}{2}f\left(\vec{\lambda}\right) & -\frac{1}{2}\lambda_3 & \frac{1}{2}\lambda_2 & 0 & 0 & 0 \\
\frac{1}{2}\lambda_3 & \frac{1}{2}f\left(\vec{\lambda}\right) & -\frac{1}{2}\lambda_1 & 0 & 0 & 0 \\
-\frac{1}{2}\lambda_2 & \frac{1}{2}\lambda_1 & \frac{1}{2}f\left(\vec{\lambda}\right) & 0 & 0 & 0
\end{pmatrix},$$

$$B = \begin{pmatrix}
\frac{1}{I_x} & 0 & 0 \\
0 & \frac{1}{I_y} & 0 \\
0 & 0 & \frac{1}{I_z} \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}.$$

IV. Control Synthesis

The regulator with variable discrete parameters is described in details in [3;5]. The functioning interval is divided into segments of equal length. The control operating in (i+1)-th interval is defined as $U_i = K\left(X_i\right)X_i = -R^{-1}B^T\left(X_i\right)S\left(X_i\right)X_i$, where $S\left(X_i\right)$ is the solution of the algebraic Riccati equation: $S\left(X_i\right)A\left(X_i\right) + A^T\left(X_i\right)S\left(X_i\right) - S\left(X_i\right)B\left(X_i\right)R^{-1}B^T\left(X_i\right)S\left(X_i\right) + Q = 0$ (2). Matrix $K\left(X_i\right)$ is the gain coefficient matrix.

The usage of the suggested control requires to solve equation (2) rather fast. However one can notice that matrix $A(X_i)$ depends only on quaternion components and angular rates. The quaternion is a unit vector and it's components lie in [-1;1] limits. Moreover, we can assume angular rates are limited by some reasonable constant so they are limited by $[-\nu;\nu]$. Considering the results of experiments with a UAV, we will assign $\nu = 20\ rad/s$.

V. Microcontroller Memory Usage Method

Let us assume that needed accuracy of quaternion components in state vector is $\varepsilon = 0.01$, that is about 1° of the UAV rotation. Then the interval [-1;1] can be divided by $2/\varepsilon + 1$ points with ε distance between each of them.

There are 201 different values for one component of a quaternion for the chosen accuracy and there are 3 quaternion components in our state vector. Thus, there are $201^3 = 8120601$ different values of these components with chosen accuracy, but we can lower this number noticing the fact that $1 - \lambda_1^2 - \lambda_2^2 - \lambda_3^2 \geq 0, \forall \lambda_i, \mathrm{i} = 1,2,3$ in accordance with the quaternion space norm.

This can be illustrated by (Fig. 2), where all the components of the quaternion lie in the nodes of the 3-dimensional grid with the size of ε and are limited by a unit sphere.

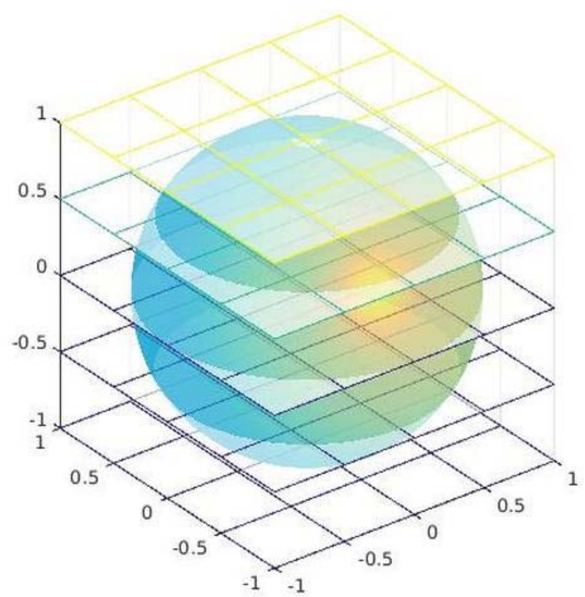

Fig. 2. Grid of $\varepsilon = 0.5$ accuracy with a unit sphere. The quaternion components are located in the intersection points of the grid lines.

Exact calculation of all possible triples was done by a bruteforce check of every dot inside of the cube with the side of 2. This was done because volume formulas do not allow achieving exact result.

The numbers of possible triplets of quaternion components for different accuracy are provided in table 1.

TABLE I
QUANTITY OF DIFFERENT TRIPLETS BY ACCURACY

Accuracy of component	Quantity of different triplets
0.1	4139
0.01	4187707
0. 001	4188780761

There are 4187707 different values for chosen accuracy. According to IEEE.754 standard describing floating point formats, there are 2 possibilities to store this value: float and double formats, sized as 4 and 8 bytes respectively. As the memory size of a control device is limited, float format is preferable.

Moreover, to lower the memory usage it is reasonable to store the $K(X_i)$ matrix. It has the size 3x6, so to store one gain matrix using the float format we have to allocate 72 bytes of memory.

The memory usage in case when the angular rates are considered is estimated using combinatorics and similar reasonings. The required memory size for all cases is presented in table 2.

TABLE 2
MEMORY SIZE REQUIRED TO STORE THE GAIN MATRICES

Accura cy	Memory size without angular rates consideration	Memory size with angular rates consideration
0.1	291 KB	113.96 MB
0.01	287.547 MB	112 GB
0.001	280 GB	110 TB

It is suggested to store the additional information about the location of the neighboring (differing by ε) gain coefficient in memory to speed up the search of the next coefficient. Obviously, there exist 2 neighbors for each of the quaternion components and we have to allocate additional 24 bytes to store the pointers. For the chosen quaternion accuracy 95.8 MB of memory is needed.

To find the next coefficient we need to sequentially move by pointers from the current matrix until the difference between the current state and the state of the candidate matrix will not be less than the chosen accuracy. This method is a simple linked list with 6 links for each of the list elements.

To analyze the speed of the provided algorithm let the state of a UAV change from $\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}^T$ to $\begin{pmatrix} 0 & 0 & 0 & -1 & -1 & -1 \end{pmatrix}^T$ with $\varepsilon = 0.01$. To find the new gain coefficient the control device has to process 603 pointers and read 43 KB. Modern SSD cards, available on the market, can achieve 30 MB/s read speed and will find the new coefficient in about 1.4 ms without taking the comparison operations timing into account.

VI. Mathematical Simulation

For the method testing the simulation in MatLab Simulink was made. The initial state is $\left(\dot{\varphi}\ \dot{\theta}\ \dot{\psi}\ \lambda_0\ \lambda_1\ \lambda_2\ \lambda_3\right)^T = \left(1\ 0\ 0\ 0.27\ 0\ 0.96\ 0\right)^T$ and $\varepsilon = 0.01$.

The figure 3 shows angular rates and quaternion trajectories of the quadcopter.

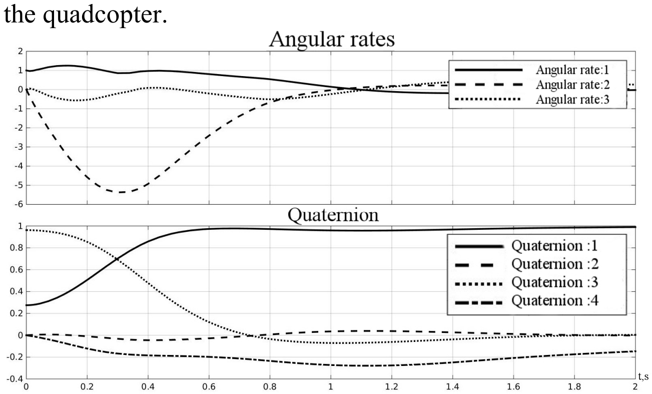

Fig. 3. Trajectories of UAV motion with suggested control law.

Perturbation of the yaw (last coordinate) appears because of the torque arising on the remaining axis while stabilization on the two other axes.

The plot for the 3 elements of $K\left(X_i\right)$ matrix change is provided in figure 4 for illustration purposes.

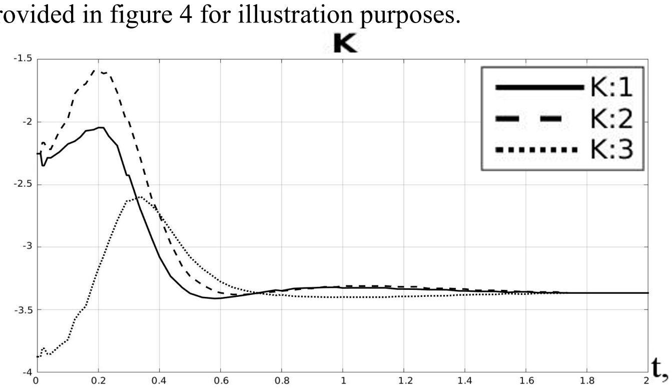

Fig. 4. Change of the first three elements of the gain coefficients matrix.

Moreover, the sequence of the corresponding gain coefficients selection is provided in figure 5.

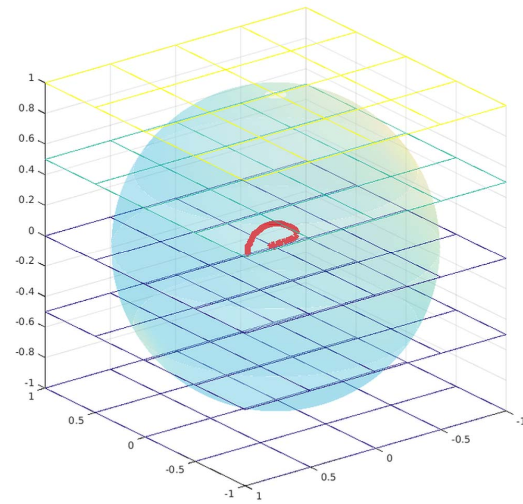

Fig. 5. Sequence of gain coefficients selection.

VII. Conclusion

The method for the realization of regulators with discrete variable parameters for nonlinear dynamic systems based on the storage of control gain coefficients in the memory of the control device was suggested in this work. Such a technique can be useful in time-critical systems control devices with low CPU performance.

Calculations of the necessary memory space depending on the accuracy were made. The described control was tested by a mathematical simulation.

The research of the accuracy change influence on the control performance is planned.

References

[1] J. D. Pearson, “Approximation methods in optimal control,” *Journal of Electronics and Control*, vol. 13, no. 1, pp. 453-469, May 1962

[2] C. P. Mracek, J. R. Cloutier, “Full envelope missile longitudinal autopilot design, using the state-dependent Riccati equation method.” in *Proc. the AIAA Guidance, Navigation, and Control Conference,* New Orleans, USA 1997, pp. 1697-1705

[3] V. N. Afanasiev, A. A. Semion, “Controller with discrete variable parameters”, *Control Sciences*, no. 5, pp. 14-19, May 2014

[4] A. Bogdanov, M. Carlson, G. Harvey, J. Hunt, D. Kieburtz, R. Merwe, E. Wan, “State-dependent Riccati equation control of a small unmanned helicopter.” in *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, Texas, USA, 2003

[5] A. A. Semion, “Quadcopter autopilot development”, *Quality. Innovations. Education*, no. 6, pp. 53-67, 2016

[6] V. N. Branetz, I. P. Shmiglevskij, “*Application of Quaterinons in Task of Rigid Body Orientation*,”, Russian Federation: Nauka, 1973.

[7] Y. Yang, “LQR Design for Spacecraft Control System Based on Quaternion Model”, *Journal of aerospace engineering*, no. 3, vol. 25, pp. 448-453, July 2012

Alexander A. Semion was born in Moscow, Russia in 1992. He received the specialist degree in applied mathematics from National Research University Higher School of Economics Moscow, Russia in 2015. Mr. Semion is currently pursuing the Ph.D. degree in system analysis at National Research University Higher

School of Economics Moscow, Russia.

He is currently working as engineer researcher in Institute for Systems Analysis Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia. His research interests include nonlinear systems analysis, optimal control and electrical engineering.