Bio-inspired Modeling of a Hovering Hummingbird with Control Policy Comparisons

Michael D. Napoli

I. Introduction

Hummingbirds represent an approachable second-order nonlinear system. At rest, hummingbirds exert only a z-axis lift force, and can control their x, y-directions using small twitches about their center of mass (COM). For this reason, the hummingbird system will be the primary focus for this report. The system will first be modeled, before being controlled by two different policies; PID and model predictive control. Then, the two control policy structures will be compared for varying disturbances from the origin.

II. Modeling

The hummingbird was modeled under the assumption that the desired configuration was when $x = 0$, which the state space was created to reflect to some extent. Figure 1 shows the free-body diagram of the hummingbird which was used to construct its governing equations.

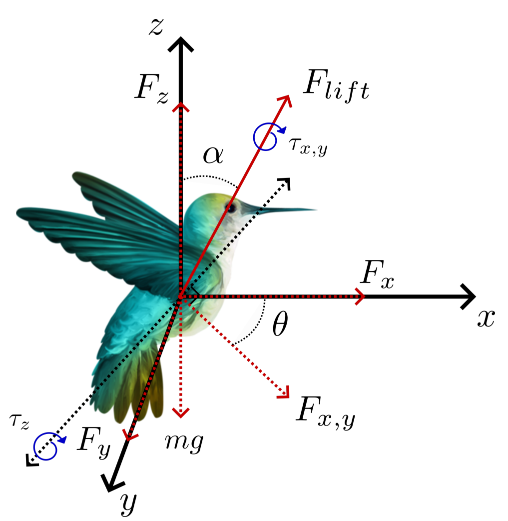


Fig. 1: Hummingbird with Free Body Diagram at the COM

Using the angle α, which describes the offset of the lift force, F_{lift}, from the z-axis, the force on the z-axis and in the x, y-plane can be defined. We also incorporate a damping term w which correlates to the loss of energy from air friction and is represented by the scalar $w > 0$.

$$\begin{aligned} F_z &= m\ddot{z} = F_{lift}\cos(\alpha) - w\dot{z} - mg \\ F_{x,y} &= F_{lift}\sin(\alpha) \end{aligned} \tag{1}$$

Then, using the offset angle from the x-axis, θ, the force along the x and y-axes can be defined along with the damping along each of the principal axes.

$$\begin{aligned} F_x &= m\ddot{x} = F_{x,y}\cos(\theta) = F_{lift}\sin(\alpha)\cos(\theta) - w\dot{x} \\ F_y &= m\ddot{y} = F_{x,y}\sin(\theta) = F_{lift}\sin(\alpha)\sin(\theta) - w\dot{y} \end{aligned} \tag{2}$$

Giving a full description of the motion of the hummingbird through the three principle axes. That said, the incorporation of α and θ into the state space means they also have shifting terms as defined by the aforementioned torques about the COM.

$$\begin{aligned} T_{x,y} &= I_z\ddot{\alpha} = \tau_{x,y} \\ T_z &= I_{x,y}\ddot{\theta} = \tau_z \end{aligned} \tag{3}$$

Where I_z and $I_{x,y}$ are the moments of inertia about the COM of the hummingbird. In the application shown here and for the sake of simple arithmetic they are taken as identities. Before stating the full state space, the force and torque terms will be replaced with generalized inputs.

$$u = \begin{bmatrix} F_{lift} \\ \tau_{x,y} \\ \tau_z \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{4}$$

Using equations (1), (2) and (3), as well as the generalized inputs, the full state space can be written in the following form.

$$\begin{bmatrix} x \\ y \\ z \\ \alpha \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}$$

$$\Rightarrow f(x,u) = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \end{bmatrix} = \begin{bmatrix} x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ \frac{1}{m}(u_1\sin(x_4)\cos(x_5) - wx_6) \\ \frac{1}{m}(u_1\sin(x_4)\cos(x_5) - wx_7) \\ \frac{1}{m}(u_1\cos(x_4) - wx_8 - mg) \\ u_2 \\ u_3 \end{bmatrix} \tag{5}$$

Giving a complete description of the state space propagation in terms of a continuous model, $x \in \mathbb{R}^{10}$ with model function $f : \mathbb{R}^{10} \to \mathbb{R}^{10}$.

III. Proportional Integral Derivative Controller

PID controllers represent a robust and fundamental control option in practical engineering. The general form of these equations are shown below.

$$u = -\alpha x - \beta \dot{x} - \gamma y \tag{6}$$

Where α, β and γ are gains on each of the PID components. Here, x is the position component, $\dot{x}$ is the velocity component and y is the error propagation through time. That is, $\dot{y} = x$ when the configuration of interest is at the origin.

The remainder of this section will be spent discussing the necessary state augmentation to incorporate integral gains into the control formula, before evaluating the stability of the formula through the Lasalle's principle.

A. State Augmentation

In order to incorporate integral gains into the control formula, the state requires the ability to evaluate error over time. Error can be described by the equation

$$\begin{aligned} e &= x \\ \Rightarrow \dot{y} &= x \end{aligned} \tag{7}$$

This is then incorporated into the state space for each of the states individually. Giving us the new augmented state space

$$\dot{y} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \quad \dot{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{8}$$

All control calculations and proofs will therefore be taken with respect to z so that the augmented state space is represented in full.

B. Evaluating Stability Through PD

Where u is the input necessary to stabilize the system, k, c and b are priority gains on each of the state-related components: x, $\dot{x}$ and y. Here, y represents the propagation of error through time. By incorporating it into the controller parameters, we can minimize steady state error as time approaches infinity. For this reason, the control stated in (5) can be replaced with

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{cases} -\alpha_1 z_3 - \beta_1 z_8 - \gamma_1 z_{13} \\ -\alpha_2 z_4 - \beta_2 z_9 - \gamma_2 z_{14} \\ -\alpha_3 z_5 - \beta_3 z_{10} - \gamma_3 z_{15} \end{cases} \tag{9}$$

So that the z-axis, α and θ are minimized as time progresses. We can observe the new behavior of this PID controller by defining an *energy* to evaluate for stability. While the full PID control method will be used in implementation, the following proof will be for when $\gamma_1, \gamma_2, \gamma_3 = 0$ and the energy function will be defined w.r.t to x alone.

$$\begin{aligned} V(x) &= \frac{1}{2} x^\intercal P x \\ \text{where } P &= \begin{bmatrix} a & 0 & \cdots & 0 \\ 0 & b & \cdots & 0 \\ \vdots, & \vdots, & \ddots, & 0 \\ 0 & 0 & 0 & j \end{bmatrix} \end{aligned} \tag{10}$$

In other words, the energy function is a summation of squares for each of the state space terms; each with the appropriate coefficient from the list $a, b, ..., \to j$. By definition,

$$V(x) = \begin{cases} 0 & \text{when } ||x|| = 0 \\ > 0 & \text{otherwise} \end{cases} \tag{11}$$

Leaving the only remaining check for Lyapunov stability being that $\dot{V}(x)$ is negative definite when the control terms are implemented into the model equations. First, the equation for $\dot{V}$ will be written as

$$\dot{V}(x) = \frac{1}{2} \dot{x}^\intercal P x + \frac{1}{2} x^\intercal P x = \dot{x}^\intercal P x. \tag{12}$$

Making the full expansion without specified control,

$$\begin{aligned} \dot{V}(x) = \; & a x_1 \dot{x}_1 + b x_2 \dot{x}_2 + \cdots + j x_{10} \dot{x}_{10} \\ = \; & a x_1 x_6 + b x_2 x_7 + c x_3 x_8 + d x_4 x_9 + e x_5 x_{10} \\ & + f x_6 (u_1 \sin(x_4) \cos(x_5) - w x_6) \\ & + g x_7 (u_1 \sin(x_4) \cos(x_5) - w x_7) \\ & + h x_8 (u_1 \cos(x_4) - w x_8) \\ & + i x_9 u_2 + j x_{10} u_3 \end{aligned} \tag{13}$$

Now plugging in the control defined by the PD controller from (9) and the negative definite terms will be moved to the top row for clarity; starting with the frictional damping, w, and the derivative gain terms β_2 and β_3.

$$\begin{aligned} \dot{V}(x) = \; & -w(f x_6^2 + g x_7^2 + h x_8^2) - i\beta_2 x_9^2 - j\beta_2 x_{10}^2 \\ & + a x_1 x_6 + b x_2 x_7 + c x_3 x_8 + d x_4 x_9 + e x_5 x_{10} \\ & - f x_6 \sin(x_4) \cos(x_5)(\alpha_1 x_3 + \beta_1 x_8) \\ & - g x_7 \sin(x_4) \cos(x_5)(\alpha_1 x_3 + \beta_1 x_8) \\ & - h x_8 \cos(x_4)(\alpha_1 x_3 + \beta_1 x_8) \\ & - i\alpha_2 x_4 x_9 - j\alpha_3 x_5 x_{10} \end{aligned} \tag{14}$$

Combining like terms leads to

$$\begin{aligned} \dot{V}(x) = \; & -w(f x_6^2 + g x_7^2 + h x_8^2) - i\beta_2 x_9^2 - j\beta_2 x_{10}^2 \\ & + x_4 x_9 (d - i\alpha_2) + x_5 x_{10}(e - j\alpha_3) \\ & + x_6 (a x_1 - f \sin(x_4) \cos(x_5)(\alpha_1 x_3 + \beta_1 x_8)) \\ & + x_7 (b x_2 - g \sin(x_4) \cos(x_5)(\alpha_1 x_3 + \beta_1 x_8)) \\ & + x_8 (c x_3 - h \cos(x_4)(\alpha_1 x_3 + \beta_1 x_8)) \end{aligned} \tag{15}$$

From here, we can make a series of assumptions to remove the nonlinearities from the equation near the origin. To start, intuition allows us to bound the domain of x_4 such that $-\frac{\pi}{2} < x_4 < \frac{\pi}{2}$; allowing us to distribute and evaluate the z-axis terms: moving the damping term to the collection of safe values.

$$\begin{aligned}\dot{V}(x) = \quad & -w(fx_6^2 + gx_7^2 + hx_8^2) - i\beta_2 x_9^2 - j\beta_2 x_{10}^2 \\ & - h\beta_1 x_8^2 \cos(x_4) \leftarrow \text{always positive} \\ & + x_4 x_9 (d - i\alpha_2) + x_5 x_{10}(e - j\alpha_3) \\ & + x_6(ax_1 - f\sin(x_4)\cos(x_5)(\alpha_1 x_3 + \beta_1 x_8)) \\ & + x_7(bx_2 - g\sin(x_4)\cos(x_5)(\alpha_1 x_3 + \beta_1 x_8)) \\ & + x_3 x_8 (c - h\alpha_1 \cos(x_4))\end{aligned} \tag{16}$$

Finally, we can make the assumption that the hummingbird does exceed some threshold of proximity around the origin so that $\cos(x_i) \approx 1$ and $\sin(x_i) \approx 0$ for $i = 4, 5$.

$$\begin{aligned}\dot{V}(x) = & \\ & -w(fx_6^2 + gx_7^2 + hx_8^2) \\ & -h\beta_1 x_8^2 - i\beta_2 x_9^2 - j\beta_2 x_{10}^2 \\ & \left\{\begin{aligned} & + x_3 x_8 (c - h\alpha_1) \\ & + x_4 x_9 (d - i\alpha_2) \\ & + x_5 x_{10}(e - j\alpha_3) \\ & + ax_1 x_6 + bx_2 x_7 \end{aligned}\right\} \qquad \text{unresolved terms}\end{aligned} \tag{17}$$

In order to make the equation $\dot{V}(x)$ negative definite, we can cancel out the sign-indefinite terms using appropriate coefficient selections. Ideally, this will lead to the ability to use the Lasalle Principle in later steps.

$$\alpha_1 = \frac{c}{h} \quad \alpha_2 = \frac{d}{i} \quad \alpha_3 = \frac{e}{j} \tag{18}$$

Allowing us to choose $\alpha_1, \alpha_2, \alpha_3 > 0$ since we defined the coefficients of the objective function arbitrarily - meaning any number can be selected in their place. Evaluating for the new proportional gains we get

$$\begin{aligned}\dot{V}(x) = \quad & -w(fx_6^2 + gx_7^2 + hx_8^2) \\ & -h\beta_1 x_8^2 - i\beta_2 x_9^2 - j\beta_2 x_{10}^2 \\ & \{+ax_1 x_6 + bx_2 x_7\} \qquad \text{unresolved terms}\end{aligned} \tag{19}$$

Unfortunately, we cannot make any claims on the remaining unresolved terms, leading to an inability to prove stability at the origin. That being said, by intuition there is an understanding that the origin w.r.t x_3, x_4 and x_5 is stable for any constant coordinates in x_1 and x_2. This conclusion can be intuitively understood from (19) if we assume

$$x_6, x_7 \to 0 \text{ as } t \to \infty \ \forall x_1, x_2. \tag{20}$$

Leading us the the final claim that the PID controller as defined in (9) is sufficient for stabilizing at the origin for x_3, x_4, x_5, but has no ability to control the x, y-axis coordinates x_1, x_2.

IV. Nonlinear Model Predictive Controller

Nonlinear model predictive control (MPC), here-to referred to as simply MPC, is a newer control strategy which was initially developed in the 1970s as a slow, but optimal prediction-oriented policy. As computation speeds in the average computer increase, the ability to implement MPC as a on-line approach becomes more realistic as well. The approach is characterized by performing an optimization problem at each time step in which the vehicle uses a predetermined model to look forward in time, characterizes the cost of its current plan, and re-plans accordingly. The state look-ahead is called the *prediction horizon* and is described by the following set with model equation $F : \mathbb{R}^n \to \mathbb{R}^n$.

$$x_{k+1} = F(x_k, u_k) \tag{21}$$

$$X = \{x_0 \wedge x_{k+1} : x_{k+1} = F(x_k, u_k) \ \forall k < P : k \in \mathbb{N}\} \tag{22}$$

Where (x_k, u_k) is the model function and is controlled by the set

$$U = \{u_k : u_k \in \mathbb{R}^m \ \forall k < P : k \in \mathbb{Z}\}. \tag{23}$$

Giving us P-states which correspond to (P-1)-controls. Using this format, a cost function can be defined which assigns a score to each of the *period* costs, that is the state and its associated input, as well as the *terminal* cost, or the final state evaluated at the end of the prediction horizon.

$$\begin{aligned} g(x_k, u_k) &= \text{period cost} \\ g_P(x_P) &= \text{terminal cost} \end{aligned} \tag{24}$$

Where the functions $g_k, g_P \in \mathbb{R}$ are defined before the start of the controller, making the set of all prediction horizon costs

$$G = \{g_k(x_k, u_k) \wedge g_P(x_P) : \forall x_k \in X, u_k \in U\}. \tag{25}$$

The optimization now becomes a minimization of the total cost over the P-prediction horizons.

$$X^*, U^* = \min_{X,U} \left(\sum_{k=0}^{P-1} g_k(x_k, u_k) + g_P(x_P) \right) \tag{26}$$

Where U^* represents the set of optimal policies as defined by the cost function solved for over the prediction horizon, and X^* is the optimal state predictions as accumulated through U^*.

In each time-step, the minimum of (26) is located and the first value is used. This iterates until the program is stopped, each time rerunning the optimization, but only implementing the first of the series of optimized inputs.

Generally speaking, finding the minimum of the MPC cost function is not a trivial process. Depending on the structure of the cost function, and the performance of the dynamics, it may even be impossible to complete in a realistic time-frame. Here, the gradient descent approach is used for a relatively small prediction horizon so as to demonstrate the policies performance.

A. Hummingbird MPC

To apply the MPC problem to the hummingbird, we must first discretize the model equation defined in (5).

$$x_{k+1} = F(x_k, u_k) = x_k + \int_{t_0}^{t_0+T} f(x_k, u_k)dt \tag{27}$$

Where T is a desired constant time-step. Since the hummingbird model as defined in (5) is time-invariant, the discrete model form becomes

$$F(x_k, u_k) = x_k + Tf(x_k, u_k) \tag{28}$$

In this implementation, the constraints associated with the physical limitations of a real hummingbird are relaxed so that the cost function is simply a penalty on the magnitude of the state space.

$$g(x_k) = (x_{ref} - x_k)^\intercal (x_{ref} - x_k) \tag{29}$$

Taking x_{ref} to be the origin (as defined in Section II), the cost function then becomes

$$g(x_k, u_k) = x_k^\intercal Q x_k \text{ where } Q = \text{diag}\left(\begin{bmatrix} 50 \\ 50 \\ 120 \\ 10 \\ 10 \\ 5 \\ 5 \\ 5 \\ 1 \\ 1 \end{bmatrix}\right) \tag{30}$$

In this form, there is no need to differentiate between the terminal and period costs as we are not incorporating the inputs into the objective function. This leads to the writing of the optimization function as

$$X^*, U^* = \min_{X,U} \sum_{k=0}^{P} x_k^\intercal Q x_k. \tag{31}$$

Where the input space U has an indirect impact on the state space through the model function. For demonstration purposes, we can rewrite the objective in terms of an initial position and the cost of the proceeding states.

$$\begin{gathered} X^*, U^* = \min_{X,U} \sum_{k=0}^{P-1} F(x_k, u_k)^\intercal Q F(x_k, u_k) \\ \text{s.t. } x_0 = x(0) \end{gathered} \tag{32}$$

This is quantitatively the same expression as (31) because of the inability to make changes to x_0 through the inputs. The minimization of this function is relatively simple when a predetermined P is applied, as it simply requires following the gradient to the function minimum. To do this, a simple form of gradient descent was implemented.

Through experimentation it will be shown that while the MPC architecture has its pitfalls (primarily computation speed and tuning issues), it gives some level of cost-minimizing control to the hummingbird's missing axes from Section III-B. That is, it cannot guarantee convergence to the global minimum of the cost space G, but shows reliable control over the states x_3, x_4, x_5 and some level of control over states x_1, x_2.

V. Discussion

Here, we will compare the two controller approaches described in the previous sections. Each of the disturbance tests will fall on a unit-cube in the strictly positive-quadrant of 3-D space (shown in figure 2.

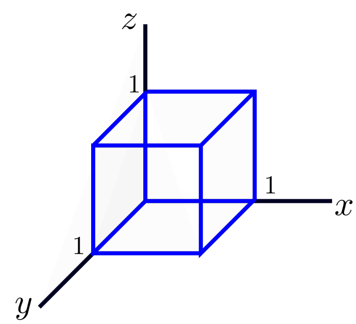


Fig. 2: Cube for Disturbance Demonstrations

Each of the disturbance tests will place the hummingbird at one of the eight available corners of the cube. The first disturbance is shown in figure 3.

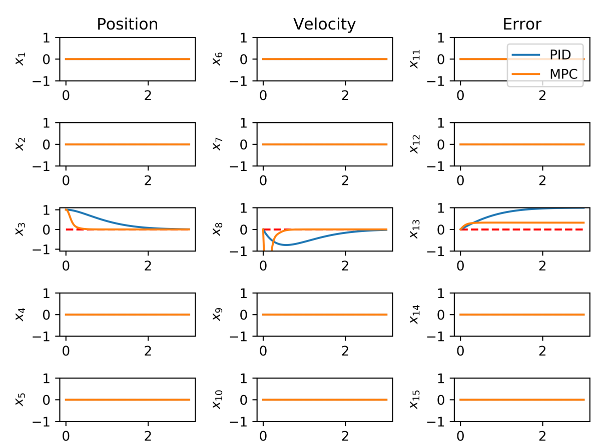
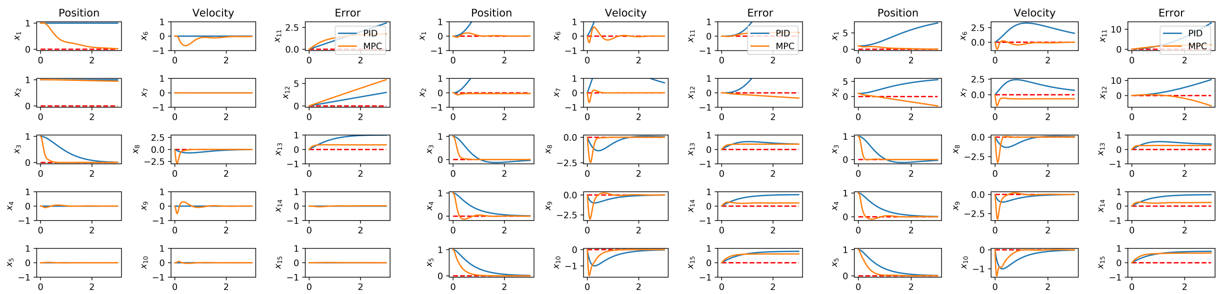


Fig. 3: $x_0 = [0, 0, 1, 0, 0]^\intercal$

Figure 3 shows a successful response by both the PID and MPC policies to stabilize the z-axis state. In the following tests, initial conditions were varied along the x, y, z axis to demonstrate the ability for the control policies to move back to the origin.

In figure 4 shows the three primary large-disturbance tests (initial positions shown in plot titles) that the control



(a) $x_0 = [1, 1, 1, 0, 0]^\intercal$ (b) $x_0 = [0, 0, 1, 1, 1]^\intercal$ (c) $x_0 = [1, 1, 1, 1, 1]^\intercal$

Fig. 4: Demonstrative Tests

polices were evaluated for. Each test will be discussed in the remainder of this section.

Figure 4a shows the performance of the policies under disturbances applied to the 3-D coordinate axes. In this demonstration, the MPC was capable of bringing the x, z-states to the origin, but was trapped in a local minima w.r.t the y-axis. The PID on the other hand was incapable of bring the x, y-states to the origin, which makes sense from our definition of the PID equations.

In figure 4b the z-axis and the angle-based states, α, θ, were perturbed. Here it is shown that the MPC is capable of bringing the entire system to the origin. Alternatively, the PID was able to bring the disturbed states back to the zero-point, but showed drift w.r.t the x, y-axes.

Finally, in figure 4c all states are disturbed. The PID showed essentially the same results as in figure 4b which is unsurprising as the addition of the x, y-disturbance does not change the reaction by the PID controller. The MPC was able to move all states to the origin, but strangely showed non-zero steady state response in $\dot{y}$. This resulted in a constant, linear drift on the y axis. I cannot explain this at present...

VI. Conclusion

Through the tests shown in this report it is demonstrated that the model predictive control form is a more intelligent alternative to PID. If a more sophisticated model was implemented (specifically with a state variable which controls center of mass pitch), the MPC would likely be capable of stabilizing the entire system for any arbitrary initial position.

Pitfalls of MPC are most commonly characterized by its relatively slow computation time and ability to fall into local minima. A demonstration of the MPC becoming trapped in a local minimum is shown in figure 4a; where the policy is incapable of seeing that the hummingbird could improve the overall cost by temporarily raising cost for a short time. Unfortunately, this can only be overcome by incorporating some parameter for direct control of the y-state variables.

In the future, focus might take the form of improving the cost function. For example, a cost barrier could be added which incorporates environment deviations and would allow the hummingbird to move towards the origin while avoiding obstacles. Soft/hard constraints may also be placed on the inputs to the system. That is, a real hummingbird may have limits to their wing-flap speed, or may have to conserve energy for prolonged flight. In such cases, constraints can be incorporated into the cost function to improve system satisfaction.

References

[1] Moritz Diehl. "Optimization Algorithms for Model Predictive Control". In: *Encyclopedia of Systems and Control*. Ed. by John Baillieul and Tariq Samad. London: Springer London, 2013, pp. 1–11. ISBN: 978-1-4471-5102-9. DOI: 10.1007/978-1-4471-5102-9_9-1. URL: http://link.springer.com/10.1007/978-1-4471-5102-9_9-1 (visited on 03/02/2022).

[2] Ruben Grandia et al. "Nonlinear Model Predictive Control of Robotic Systems with Control Lyapunov Functions". In: *Robotics: Science and Systems XVI* (July 12, 2020). DOI: 10.15607/RSS.2020.XVI.098. arXiv: 2006.01229. URL: http://arxiv.org/abs/2006.01229 (visited on 03/02/2022).

[3] B. T. Polyak. "Newton's method and its use in optimization". In: *European Journal of Operational Research* 181.3 (Sept. 16, 2007), pp. 1086–1096. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2005.06.076. URL: https://www.sciencedirect.com/science/article/pii/S0377221706001469 (visited on 05/19/2022).

[4] Jian Zhang et al. "Geometric flight control of a hovering robotic hummingbird". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017 IEEE International Conference on Robotics and Automation (ICRA). May 2017, pp. 5415–5421. DOI: 10.1109/ICRA.2017.7989638.

[5] Fan Fei et al. "Flappy Hummingbird: An Open Source Dynamic Simulation of Flapping Wing Robots and Animals". In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). ISSN: 2577-087X. May 2019, pp. 9223–9229. DOI: 10.1109/ICRA.2019.8794089.