Controlling drones

Pratyush Gupta

January 2023

1 Introduction

In the following text, we will be setting up the theoretical formulation for controlling a drone using MPC(Model Predictive Control) for a limited control problem. The problem specification and the notation to be used is as follows.

- x_D : The desired position
- v_D : The required velocity
- x : The current position
- v : The current velocity
- T : The time horizon for the MPC controller

2 MPC basics

The idea of the MPC controller is simple.

Compute a series of inputs that will minimize the error at time T, then supply the first input in this series at each sampling instant

This idea will become more clear as we continue this exposition.

3 The simpler case

Lets consider the simpler case when we do not need to control the velocity. Define

$$e_x = x_D - x$$

According to the principle of the MPC controller, we need to figure out the inputs that will get as close as possible to x_D at time T. There can be any number of such trajectories or none at all, depending on the case. In this case there can be an infinite number of such solutions. Here will choose the simplest

of these, the constant acceleration path.
The required acceleration is simple to compute

$$x_D = x + vT + \frac{1}{2}a_x T^2$$

Since $e_x = x_D - x$, we get

$$a_x = \frac{2(e_x - vT)}{T^2}$$

Going back to the MPC view, a series of inputs that causes a constant acceleration a_x will take us to x_D after the time horizon T.

The analogous case for controlling only velocity, is simple as well. The computation is left as a trivial exercise for the reader. We are stating the solution here

$$a_v = \frac{e_v}{T}$$

4 Combining the cases

When we have both the objectives, that is, controlling both the position and velocity, things get a little more interesting. At this point we will strongly recommend that the reader pause, and think of the issues that will arise and how can we deal with them. It will be more insightful to continue after doing such.

Continuing on, we hope the reader realizes that the strategy, we have adopted till now, of choosing a constant acceleration, may not work anymore. This is quite clear, since generally a_x and a_v will not be equal. At this point we need to devise a way to reconcile this apparent contradiction. Again, we request the reader pause and give this question some thought. For the greatest insight now follows.

One way to deal with the issue is to choose a more interesting strategy, one that can vary the acceleration, so that it may be able to satisfy both the velocity and the position controls. But such a strategy may be quite involved, and indeed, we have not been able to formulate such till now. *We*, of course, referring to the author. The general community is certainly much more capable.

So we wish to continue using a constant acceleration route. But now, we will formulate the problem a bit differently. Our objective is now the following

$$\underset{a}{\text{argmin}} \;\; |x_D - x_T| + |v_D - v_T|$$

At this juncture, it will serve to be very clear regarding the notation. x_D and v_D remain the same as before. x_T is the position we expect the drone to be at, if it gets an acceleration a. x_T is, obviously, a function of a. Observe that it is trivial to simplify the expression on the right, Once again, we will not show the

mechanical details of computation, which should be carried out by the reader. We ill simply write out the final result.

$$a_{desired} = \underset{a}{\text{argmin}} \ \frac{T^2}{2}|a - a_x| + T|a - a_v|$$

This expression once again calls for some careful thought. Observe that the exact coefficients of the two terms on the right depend on T. If we were only controlling either position or velocity then there is nothing to worry about. But f we are controlling both position and velocity, the we must ensure that the two coefficients are equal. If that is not the case then, we will just get a_x or a_v as the optimum. The reason is as follows:
The optimum must lie on the line joining a_x and a_v in the three dimensional vector space. This is because if there is an optimum which does not lie on the line, then we can drop a perpendicular on this line from the supposed optimum. The intersection of this perpendicular and the line is going to be atleast as good as the supposed optimum. So we need only to consider the points on the line. From here it is trivial to show the truth of the assertion.

when we equate the coefficients, we get $T = 2$. The optimum for the expression, is then going to be any point on the line joining a_x and a_v.

$$a_{desired} = \lambda a_x + (1 - \lambda)a_v$$

λ, ofcourse is [0,1]. The role of the parameter λ is quite clear to the attentive reader. If that is not the case, we encourage the student to give it some thought.

While all points on the line connecting a_x and a_v are solutions they are not identical. The parameter λ decides the weight given to the two end points. A large λ will mean, a strong position control, and a correspondingly weak velocity control. A small value of λ will have the opposite effect. Thus

$$a = \lambda \frac{2(e_x - vT)}{T^2} + (1 - \lambda)\frac{e_v}{T}$$

5 Computing the thrust, roll and pitch

As the attentive student will have noted, we are not controlling the yaw in this problem, thus simplifying the issue greatly. To compute the required thrust, roll and pitch, we only need to apply the laws of motion.

$$f\hat{z_d} - mg\hat{z} = ma$$

It is once again essential to be clear about the notation. f is the required thrust. z_d is the $z - axis$ for the drone, mg represents the weight of the drone, z is the $z - axis$ of the ground frame. a is the acceleration computed in the previous section . So

$$f\hat{z_d} = mg\hat{z} + ma$$

This allows as to compute f and z_d, f is the magnitude of the vector in the right and z_d is the direction.

Computing the roll and pitch from here is fairly straightforward. Let R be the rotation matrix that takes the ground frame to the desired drone frame.

$$\hat{z_d} = R\hat{z}$$

z is simply $[0, 0, 1]^T$. $\hat{z_d}$ is known. R has two unknowns, roll and pitch, but yaw is known. Keep in mind that we are not controlling the yaw, so we do not need to change it. On expanding the right side of the expression, we will obtain two equations and two unknown which can be easily solved for roll and pitch.

6 Amending the acceleration

Earlier we had computed the acceleration to be

$$a = \lambda \frac{2(e_x - vT)}{T^2} + (1 - \lambda)\frac{e_v}{T}$$

Now we will throw in the last bit of complexity. Suppose the desired position x_D is not constant, that is, $\dot{x_D} \neq 0$ and $\ddot{x_D} \neq 0$.

This is not hard to deal with. To deal with $\ddot{x_D}$, we will just add $\ddot{x_D}$ to a. This is essentially to cancel out the relative acceleration.
To deal with the velocity problem, we can just replace the v with $v - \dot{x_D}$. It is quite easy to see where this comes from. we will further simply it.

$$e_x = x_D - x$$

$$\dot{e_x} = \dot{x_D} - v$$

Substituting it in, we finally get

$$a = \lambda \frac{2(e_x + \dot{e_x}T)}{T^2} + (1 - \lambda)\frac{e_v}{T} + \ddot{x_D}$$

$$\boxed{a = 2\lambda \frac{e_x}{T^2} + (1 - \lambda)\frac{e_v}{T} + \ddot{x_D} + 2\lambda \frac{\dot{e_x}}{T}}$$

This is the final expression for acceleration. We have thus successfully built a controller based on the MPC principle using very simply foundational laws. Observe that there is are proportional and derivative terms, which are very elegantly produced.
The author will be glad to respond in case of any queries, corrections or contributions.