GA on optimizing the parameters of LQR tracking algorithm

Yifeng Tang

1 Background

LQR is an useful method to solve 2D and 3D trajectory tacking problems in robotics and autonomous driving etc. It‘s based on optimal control and the state space equation in modern control theory. The algorithm is essentially a general control algorithm which means it can solve a large number of problems in linear systems. However, the problem is that there are several parameter matrices in LQR that can greatly influence its performance. In fact, parameters are of vital importance in almost all of the control algorithms. Finding a proper combination of parameters is one of the main goals of designing an efficient control system. In this paper, a typical 2D trajectory tracking problem is formulated with the assumption that the kinematics model of the agent is known and there is no disturbance (friction, obstacles etc) in this environment. Most importantly, GA(Genetic Algorithm) is chosen to solve this problem, which is also a classical local search algorithm.

2 Problem Statement

Considering the following system model.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

x, y represent state vector and output vector while A is the system parameter matrix and B, D is the control value parameter matrices. u is the control value vector, C is the output parameter matrix. In this paper,

$$x = \begin{pmatrix} e \\ \dot{e} \\ \theta \\ \dot{\theta} \end{pmatrix} \quad A = \begin{pmatrix} 1 & dt & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{v}{L} \end{pmatrix}$$

t is time and e represents the error between the desired trajectory and current trajectory, θ represents the steering angel of the agent, v is agent‘s speed and L is the distance between two wheels of the agent.

After the former treatment, the tracking problem is translated into a control problem aiming at make the system stable. To solve the problem, considering LQR which is based on optimal control. Selecting quadratic performance function,

$$J = \int_0^{+\infty} \left[x^T Q x + u^T R u \right] dt$$

and the terminal goal is to minimize J which means it can be the fitness function of GA. In control theory, there is an equation called Riccati equation that can get the feedback gain matrix K.

$$PA + A^T P - PBR^{-1} B^T P + Q = 0$$

Solving the equation, and getting the final control laws.

$$K = R^{-1} B^T P \quad U = -KX = -R^{-1} B^T P X$$

Q and R are the parameter matrices to be optimized. However, to simplify the problem, R is set to be $I^{1\times 1}$ which means GA is only focusing on Q.

3 Result and Conclusion

Simulation is the main form of experiment and the programming language is Python. At the same time, vscode is the main experimental IDE.

3.1 Experimental Design

Without loss of generality, the desired trajectory is generated with some curvature. In order to evaluate the results, both of the original parameters and optimized parameters will be used to track the same trajectory. LQR algorithm is used to control the steering angle while the speed controller is simply a PD (Proportion-Differential) controller for it‘s not the factor that can greatly influence the final results.

3.2 Results

There are two conditions, tracking result of original parameters and the other one is the result of optimized parameters.

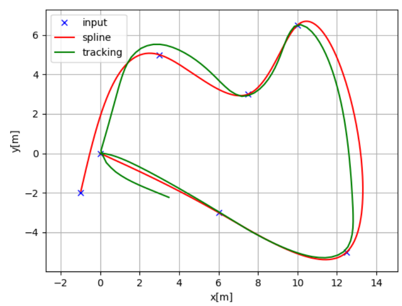

Figure 1: Original Q

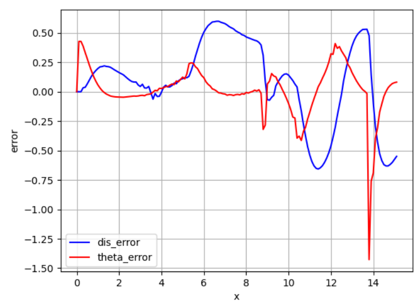

Figure 2: Errors

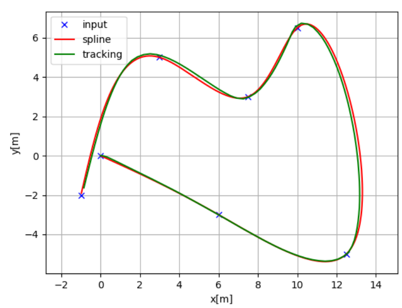

Figure 3: Optimized Q

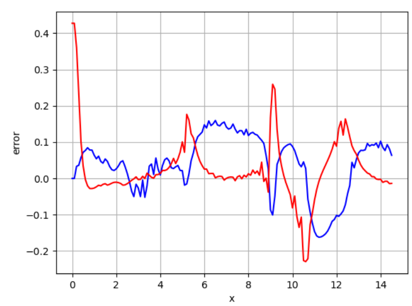

Figure 4: Errors

Fig 1 and Fig 2 present the tracking result of origin parameters, in which $Q = I^{4\times4}$.Fig 3 and Fig 4 are the results of optimized Q, and

$$Q_{optimized} = \begin{pmatrix} 10.18 & 0 & 0 & 0 \\ 0 & 9.64 & 0 & 0 \\ 0 & 0 & 6.99 & 0 \\ 0 & 0 & 0 & 4.89 \end{pmatrix}$$

.

In Fig 1 and Fig 3, the green line means the tracking trajectory and the red one means the desired trajectory. In Fig 2 and Fig 4, the red line is the errors between current steering angles and desired steering angles while the blue one is the errors between current position and desired positions. It is easy to find that the agent cannot even complete the tacking task with the original Q, and the performance is not satisfying for the maximum absolute value of steering angle can be greater than 1.20 and that of distance error can be greater than 0.5. At the same time, the performance of optimized parameters can be much better, the agent can easily complete the tacking task and the maximum absolute errors are limited in 0.4.

3.3 Conclusion

As a local search algorithm, GA can quickly find a local optimal point and this advantage is shown in the results of this paper. However, while the object model is complex and uncertain or nonlinear. GA is on longer applicable.